ESEIAAT

Bachelor's Thesis

Study of a methodology for the flight simulation of ram-air parachutes using a vortex-lattice aerodynamic model

Report

Degree: Bachelor's degree in Aerospace Vehicle Engineering

Delivery Date: 10/06/2017

Student: Pérez Sánchez, David

Director: Ortega, Enrique

Universitat Politècnica de Catalunya

Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa

“Jump, and you will find out how to unfold your wings as you fall”

Ray Bradbury

Table of contents

Acknowledgements

First, I want to acknowledge my parents, my sister and my partner for their unconditional support and affection. They have been direct observers of how I have tried to find the balance between the realization of this project and my family life. They have helped me and given me the best of themselves in different situations, and especially, in the final stretch.

My other great support has been my tutor Enrique. He has done everything necessary so that I could complete this project of barely four months, but with great sacrifice and work behind. His extensive experience, his confidence in my work, and his ability to guide my ideas have been fundamental for the project. With his patience, closeness and affection, he has made me feel better in the most difficult moments.

I also want to acknowledge CIMNE because, without their help, it wouldn't have been possible to carry out the simulations with data from real ram-air parachutes.

To finish, I want to express my most sincere gratitude to everyone that has been part of this project, for supporting me and teaching me the importance of always pushing on.

List of figures

List of tables

List of Abbreviations and Symbols

Due to the huge amount of symbols used in this thesis, it is difficult to avoid their repetition for different purposes. Consequently, they are divided in to different lists, one for chapters 1 & 2 and one for chapter 3 (flight simulator).

In Chapters 1 & 2

$\overline{\overline{\mathbf{A}}}$	Influence coefficients matrix
a	Canopy height
$\overrightarrow{\mathbf{A_k}}$ **and** $\overrightarrow{\mathbf{B_k}}$	Vectors that define the diagonals of panel "k"
$\mathbf{a_{kL}}$	Influence coefficient of vortex "L" at collocation point "k"
$\boldsymbol{\alpha}$	Angle of attack
AGARD	Advisory Group for Aerospace Research and Development
AR	Aspect Ratio
$\mathbf{A_{t1}}$, $\mathbf{A_{t2}}$	Areas of the two triangles in which a panel is divided to calculate its surface (Heron formula)
$\boldsymbol{\alpha_{ZL}}$	Zero lift line angle of attack
b	Canopy span
$\mathbf{b_f}$	Control surface span
$\boldsymbol{\beta}$	Side-slip angle
B.C.	Boundary conditions
$\boldsymbol{\gamma}$	Dihedral/arc-anhedral angle
$\boldsymbol{\Gamma}$	Vortex circulation value
$\vec{\boldsymbol{\Gamma}}$	Vortex circulation vector: $\Gamma \cdot \vec{\delta l}$
c	Canopy chord
$\mathbf{c_{root}}$	Canopy root chord
c.s	Control surface
$\mathbf{c_{tip}}$	Canopy tip chord
BRS™	Ballistic Recovery Systems
CAD	Computer-Aided Design
CIMNE	International Center for Numerical Methods in Engineering
$\mathbf{C_D}$	Ram-air wing drag force coefficient
$\mathbf{C_{D_i}}$	Ram-air wing induced drag component
$\mathbf{C_{D_l}}$	Suspension lines drag coefficient
$\mathbf{C_{D_P}}$	Profile drag coefficient integrated along the span
$\mathbf{C_{D_{PL}}}$	Payload drag coefficient
$\mathbf{C_{d_p}}$	Profile drag coefficient
$\overrightarrow{\mathbf{C_F}}$	Vector of canopy non-dimensional aerodynamic forces
$\mathbf{c_f}$	Control surface chord
CFD	Computational Fluid Dynamics
C.G.	Centre of Gravity
$\mathbf{C_L}$	Ram-air wing lift force coefficient
$\mathbf{C_{LC}}$	Ram-air wing circulation lift coefficient
$\mathbf{C_l}$	Ram-air wing section lift force coefficient
$\mathbf{C_{L_\alpha}}$	Wing lift slope coefficient

C'_{L_α}	Wing lift slope coefficient corrected for ram-air wings (low AR)
C_{l_α}	Two-dimensional lift curve slope
C'_{l_α}	Two-dimensional lift curve slope corrected for ram-air wings (low AR)
$C_{L_{\beta=0}}$	Lift force coefficient for a null dihedral angle
$\overrightarrow{C_M}$	Vector of canopy non-dimensional aerodynamic moments
C_N	Normal aerodynamic force component dimensionless coefficient
C_p	Specific heat coefficient at constant pressure
CPU	Central Processing Unit
C_T	Tangential aerodynamic force component dimensionless coefficient
D	Drag Force
d	Suspension lines diameter
ΔC_L	Ram-air wing lift coefficient increment due to the non-linearity present in low AR wings
Δt	Timestep
Δt^*	Non-dimensional timestep
δ	Efficiency parameter which relates the performance of the wing to that of an elliptical wing. It is related with the Oswald factor (e)
δ_L	Deflection of the left control surface
$\overrightarrow{\delta l}$	Vector that defines the length of a vortex segment
δ_R	Deflection of the left control surface
$\overrightarrow{I_1}$	Identity matrix
D_{ref}	Parachute reference diameter
DLM	Doublet Lattice Method
DoF	Degrees of Freedom
DSD/ SST	Deforming-spatial-domain/stabilized space-time
e	Total energy
e_o	Oswald factor
ESEIAAT	Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa. (Before ETSEIAT)
ETSEIAT	Escola Tècnica Superior d'Enginyeries Industrial i Aeronàutica de Terrassa
ϕ	Velocity potential
$\vec{F}$	Vector of canopy aerodynamic forces
$\vec{f}$	Vector of panel aerodynamic forces
$\vec{f}^e$	Electromagnetic force
FEM	Finite Element Methods
F_N	Normal aerodynamic force component
Fr	Froude Number
FSI	Fluid Structure Interaction
F_T	Tangential aerodynamic force component
$\vec{g}$	Gravity vector
GREVA	Grau en Enginyeria en Vehicles Aeroespacials
HVLM	Horseshoe Vortex Lattice Method
$\hat{i}$	O-x axis unit vector
IB	Immersed Boundary
$\hat{j}$	O-y axis unit vector
JPADS	Joint Precision Airdrop System

K	Reduction factor that relates C_{l_α} with C'_{l_α}
K_1	Factor multiplying the non-linear terms of lift and drag ram-air wing coefficients.
$\hat{\mathbf{k}}$	O-z axis unit vector
κ	Thermal conductivity
k	Acts as panel/node counter in the aerodynamic solver
K.J.	Kutta-Joukowsky
L	Lift force
Λ_{LE}	Sweep angle referenced to the leading edge
λ	Taper ratio
L1, L2, L3, L4	Length of panel sides
L5	Panel diagonal length (used for surface calculation)
LAPES	Low Altitude Parachute Extraction System
L.E.	Leading Edge
LLT	Lifting Line Theory
μ	Air dynamic viscosity
$\vec{M}$	Vector of canopy aerodynamic moments.
$\vec{m}$	Vector of panel aerodynamic moments.
M	Number of panels along the span
Ma	Mach number
MTRS	Mosquito Tactical Resupply System
N	Number of panels along the chord
n	Number of suspension lines
$\vec{n}_K$	Normal unit vector of a panel "k"
NASA	National Aeronautics and Space Administration
OE	Exterior offset: distance from the wing tip to the control surface
P_1, P_2	Points which define the extremes of a vortex segment
P_p	Point where the velocity induced by the vortex segment is calculated
p	Pressure
PADS	Precision Aerial Delivery System
PEGASYS	Precision and Extended Glide Airdrop System
$\vec{q}$	Heat conduction
R	Mean suspension line length
$\vec{r_0}$	For calculation of the vortex line induced velocity, vector from P_1 to P_2
$\vec{r_1}$	For calculation of the vortex line induced velocity, vector from P_1 to P_p
$\vec{r_2}$	For calculation of the vortex line induced velocity, vector from P_2 to P_p
ρ	Air density
ρ_0	Subscript "0" indicates that air density is ct. (incompressible flow)
Re	Reynolds Number
RFD	Rotating Flexible Decelerator
$\vec{RHS}$	Right Hand Side of the influence coefficients calculation system
S_{panel}	Panel surface
sp	Semi-perimeter of a triangle
S_{ref}	Canopy reference surface used to make forces non-dimensional
SARS	Steerable Autonomous Parachute System
T	Temperature
t	Time

$\boldsymbol{\tau}$	Small positive factor that increases the induced angle of incidence over that for the minimum case of elliptic loading
$\vec{\boldsymbol{\tau}}$	Stress tensor
$\mathbf{du}$	Internal energy differential
$\mathbf{[U(t), V(t), W(t)]_k}$	Instant aerodynamic speed vector at the collocation point "k"
$\vec{\mathbf{U}}_{\mathbf{w_k}} = \mathbf{[u_w, v_w, w_w]_k}$	Wake induced velocity vector at the collocation point "k"
$\vec{\mathbf{U}}_{\infty}$	Free stream velocity vector
$\mathbf{U}_{\infty}$	Free stream velocity
$\vec{\mathbf{U}}_{\mathbf{tr}}$	Velocity induced at the acting point of the force by all the trailing vortex lines from both the canopy and the wake
$\vec{\mathbf{U}}$	Total velocity. $\vec{\mathrm{U}} = \vec{\mathrm{U}}_{\infty} + \vec{\mathrm{U}}_{\mathrm{tr}}$
$\hat{\mathbf{U}}$	Total velocity norm
UK	United Kingdom
UPC	Universitat Politècnica de Catalunya
USA	United States of America
UVLM	Unsteady Vortex Lattice Method
$\vec{\mathbf{v}}$	Flow velocity vector
$\mathbf{V_{ref}}$	Reference aerodynamic speed used to make forces non-dimensional
VLM	Vortex Lattice Method
$\vec{\boldsymbol{\omega}}$	Flow vorticity
$(\mathbf{X_{cp}}, \mathbf{Y_{cp}}, \mathbf{Z_{cp}})$	Control point coordinates

In Chapter 3

A, B, C, P, Q, R, H	Apparent mass constants
$\mathbf{A_{panel}}$	Panel area
$\boldsymbol{\Delta A_{panel}}$	Extra panel area due to wake length correction
$\mathbf{a}$	Maximum canopy height when fully inflated
$\mathbf{AR}$	Canopy Aspect Ratio
$\mathbf{b}$	Canopy span when fully inflated
$\mathbf{c}$	Canopy chord
b2c	Body frame to canopy frame transformation matrix
C	Matrix used to store configuration values of GPSim
$\vec{\mathbf{C}}_{\mathbf{p}}$	Vector from Matlab origin to an specific control point.
$\mathbf{C_{D_{PL}}}$	Payload drag coefficient
C.G.	Centre of Gravity
DoF	Degrees of Freedom
$\vec{\mathbf{F}}_{\mathbf{A}}$	Aerodynamic force
$\vec{\mathbf{F}}_{\mathbf{AM}}$	Apparent mass force
$\vec{\mathbf{F}}_{\mathbf{PL}}$	Payload drag
$\vec{\mathbf{F}}_{\mathbf{W}}$	Weight
G	Matrix used to store geometry values of GPSim
g	Gravity
GPSim	Gliding Parachute Simulator
ifact	Increase factor of the wake length
$\bar{\bar{\mathbf{I}}}_{\mathbf{T}}$	Inertia matrix of the total system

$\bar{\bar{\mathbf{I}}}_{\mathbf{AM}}$	Apparent mass matrix
$\bar{\bar{\mathbf{I}}}_{\mathbf{AI}}$	Apparent inertia matrix
i2b	Intertial frame to body frame transformation matrix
Iner	Matrix used to store inertia values of GPSim
ISA	International Standard Atmosphere
L.E.	Leading edge
$\mathbf{m}$	Mass
$\vec{\mathbf{M}}_{\mathbf{A}}$	Aerodynamic moment
$\vec{\mathbf{M}}_{\mathbf{AM}}$	Apparent mass moment
m2c	Matlab frame to canopy frame transformation matrix
$\mathbf{N_{xw}}$	Number of wake panel rows
$\mathbf{p}, \mathbf{q}, \mathbf{r}$	System's angular velocity components expressed in the body frame
$\tilde{\mathbf{p}}, \tilde{\mathbf{q}}, \tilde{\mathbf{r}}$	Components of the system's angular velocity expressed in the canopy reference frame
$\dot{\mathbf{p}}, \dot{\mathbf{q}}, \dot{\mathbf{r}}$	System's angular acceleration components expressed in the body frame
$\vec{\mathbf{r}}$	Vector from the system's C.G. to the control point of the corresponding panel.
$\vec{\mathbf{r}}_{\mathbf{ref}}$	Vector from the root chord L.E. to the matlab frame origin.
$\mathbf{S}^{\mathbf{b}}_{\mathbf{CG,LE}}$	Auxiliary matrix used for the cross product of the vector from C.G. to root chord L.E.
$\mathbf{S}^{\mathbf{b}}_{\mathbf{CG},\mathbf{X_{am}}}$	Auxiliary matrix used for the cross product of the vector from C.G. to the apparent mass centre
$\mathbf{S}^{\mathbf{b}}_{\mathbf{CG,PL}}$	Auxiliary matrix used for the cross product of the vector from system's C.G. to payload's C.G.
$\mathbf{S_{ref}}$	Canopy reference surface
$\mathbf{S_{ref_{PL}}}$	Payload reference surface
$\mathbf{S}^{\mathbf{b}}_{\boldsymbol{\omega}}$	Auxiliary matrix used for the cross product of the angular velocity vector
$\mathbf{T}$	Kinetic energy
$\mathbf{t}$	canopy thickness
$\mathbf{u}, \mathbf{v}, \mathbf{w}$	System's C.G. linear velocity components expressed in the body frame
$\tilde{\mathbf{u}}, \tilde{\mathbf{v}}, \tilde{\mathbf{w}}$	Components of the linear velocity seen by the canopy expressed in canopy reference frame
$\dot{\mathbf{u}}, \dot{\mathbf{v}}, \dot{\mathbf{w}}$	System's C.G. linear acceleration components expressed in the body frame
$\vec{\mathbf{V}}_{\mathbf{CG}}$	Gravity centre velocity
$\mathbf{V_{PL}}$	Aerodynamic velocity seen by the payload
$\vec{\mathbf{V}}_{\mathbf{ref}}$	Reference velocity used for the non-dimensionalization
$\vec{\mathbf{V}}_{\mathbf{rot}}$	Component of velocity caused by the system rotation
$\vec{\mathbf{V}}_{\mathbf{wind}}$	Wind velocity
$\vec{\mathbf{V}}_{\infty}$	Total velocity that a control point sees.
$\dot{\mathbf{X}}, \dot{\mathbf{Y}}, \dot{\mathbf{Z}}$	System's C.G. linear velocity components expressed in the inertial frame
$\mathbf{X_b}, \mathbf{Y_b}, \mathbf{Z_b}$	Body axes/coordinates
$\mathbf{X_c}, \mathbf{Y_c}, \mathbf{Z_c}$	Canopy axes/coordinates
$\overrightarrow{X_{CG}LE}$	Vector from the system's C.G. to the rooth chord L.E.
$\mathbf{X_I}, \mathbf{Y_I}, \mathbf{Z_I}$	Inertial axes/coordinates
$\mathbf{X_M}, \mathbf{Y_M}, \mathbf{Z_M}$	Matlab axes/coordinates
$\Delta\mathbf{x}$	Panel length

Δt	Timestep
Γ	Rigging angle
Γ_{panel}	Panel circulation
Γ^{*}_{panel}	Corrected panel circulation after wake length correction
θ	Pitch angle
ρ	Density
ϕ	Roll angle
$\dot{\phi}, \dot{\theta}, \dot{\Psi}$	System's angular velocity components expressed in the inertial frame
Ψ	Yaw angle
$\vec{\omega}$	System's angular velocity vector

Aim

The aim of this work is the study and development of a methodology for the flight simulation of ram-air parachutes systems. The chosen methodology, based on an unsteady vortex lattice method and semi-empirical corrections to account for drag effects, will be designed to obtain an efficient tool allowing realistic and rapid analyses.

Scope

This study focuses on ram-air parachute-payload systems with small thickness canopies flying under nominal conditions. The model simplification and main aspects developed in this work are described below.

The flow aerodynamics will be solved only for the canopy, using a potential flow aerodynamic solver. The method chosen is the vortex lattice method (VLM) that will be developed in MATLAB® and will be implemented as a subroutine of the flight simulator. The latter will also be coded in MATLAB® and a basic verification/validation assessment of the results will be carried out using available reference data.

Regarding geometric modelling, the parachute canopy will be represented in parametric form, in terms of its taper ratio, dihedral and sweep angle. This simple model is enough for the simulation of typical canopy shapes. The methodology used (VLM) is based on a lifting surface panel method and thus, thickness effects are not accounted for. For this reason, results will be only reliable for small thickness canopies. Panel methods do not consider viscous effects (potential flow aerodynamics) and, consequently, viscous drag cannot be computed in a direct manner. In this work, these effects on the canopy, suspension lines and payload will be accounted for by means of semi-empirical corrections. It should be noted that no deformation of the model will be considered beyond the deflection of parachute control surfaces (which will be done in an approximate manner). Hence, the geometry represented will be that corresponding to nominal flight conditions.

The parachute-payload system will be studied as a whole or, in other words, it will be simulated as a rigid body. Thus, aspects such as relative deformations between payload and the canopy will be neglected. The model proposed consists of 6 degrees of freedom (DoF) including translations around the three axes (x, y and z movements) and the corresponding rotations (roll, pitch, and yaw).

Requirements

The methodology in this work will be developed under the following main requirements:

- 6 DoF trajectory simulations of ram-air parachute-payload systems flying under nominal conditions
- Potential-flow aerodynamics
- Capability of accounting for drag effects
- Modelling of parachute's control surfaces
- Low computational cost

Justification

Ram-air parachute systems have a wide range of applications in fields such as humanitarian assistance, security, merchandise delivering or military purposes. The need to plan, calculate and simulate parachute trajectory in real-time is common to all these applications and involves solving a coupled aerodynamic-structural problem. There are different alternatives to face this problem but, in general, these are complex and computationally expensive. This inherent complexity in the numerical simulation is the reason why most of the manufacturers (particularly small and mid-size companies) rely mainly on empirical results and leave apart simulation. To confirm this fact, a survey was conducted in 2009 by a student of ETSEIAT (now ESEIAAT) in which 15 different worldwide parachute manufacturers were asked about the use of numerical simulation methods in their design process. None of the 10 manufacturers that answered were using numerical methods apart from CAD software [1].

In spite of the many advances in numerical applications for parachute simulation during the last 20 years, this field offers today many possibilities for further development, which would provide great benefits to manufacturers. As a complement to empirical tests, it would be really advantageous in the design process to have additional numerical information difficult to obtain experimentally, which helps to reduce costs and time of development. To encourage manufacturers to use computer simulations, faster tools must be devised and developed. This need motivates this project.

Organisation and Scheduling

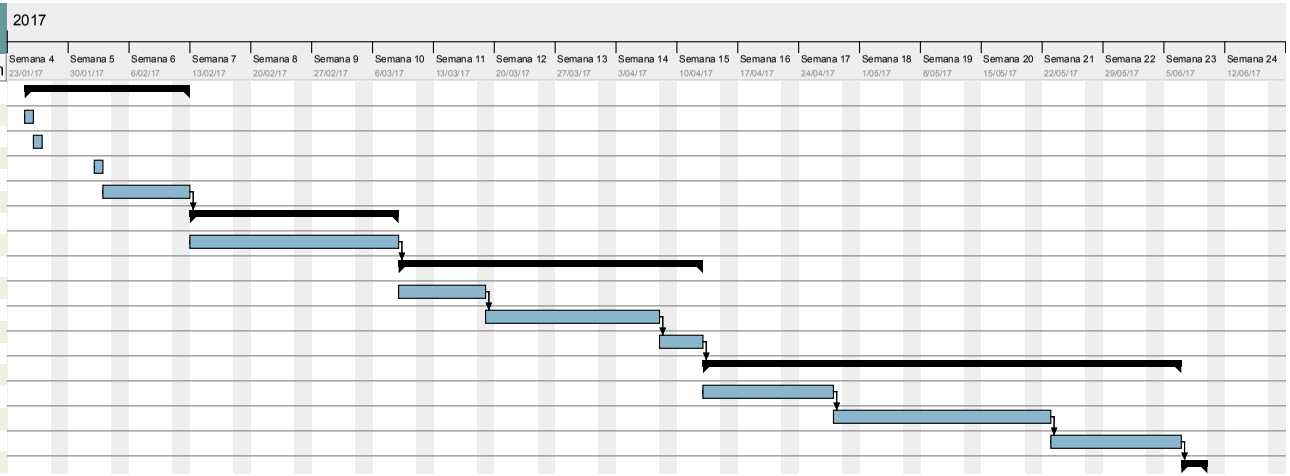
A detailed list of the tasks carried out during the development of this work are shown in the table below. Dependencies of each of the tasks are also detailed. For a better understanding, see Gantt chart in the next page.

	Task	Description	Dependencies
A. PRELIMINARY TASKS	A.1	First meeting with TFG director	
	A.2	Objectives establishment	
	A.3	TFG assignation at school's Intranet	
	A.4	Planning & Organisation	
B. LITERATURE REVIEW	B.1	Bibliographical research: parachute technology, trajectory simulation models and vortex lattice methods	
C. DEVELOPMENT OF UNSTEADY VORTEX LATTICE SOLVER WITH MATLAB	C.1	Defining input geometry and model parameters	B
	C.2	Aerodynamic calculations	C.1
	C.3	Validation/verification of the code	C.2
D. FLIGHT SIMULATOR DEVELOPMENT	D.1	Study of the simulator dynamic model	C
	D.2	Development of a 6 DoF ram-air parachute flight simulator in MATLAB	D.1
	D.3	Validation/Verification assessment	D.2
E. FINAL FORMATING OF THE PROJECT REPORT	E.1	Preparation of required additional documents / project report formatting	D
	E.2	Conclusions and recommendations of project follow-up	D

Table 1: List of tasks carried out during the development of the work

Diagrama de Gantt

Nombre	Fecha de inicio	Fecha de fin
PRELIMINARY TASKS	25/01/17	12/02/17
First meeting with TFG director	25/01/17	25/01/17
Objectives establishment	26/01/17	26/01/17
TFG assignation at school's Intranet	2/02/17	2/02/17
Planning & Organisation	3/02/17	12/02/17
LITERATURE RESEARCH	13/02/17	8/03/17
Bibliographical research	13/02/17	8/03/17
DEVELOPMENT OF AN UNSTEADY VORTEX LATTICE SOLVER WITH MATLAB	9/03/17	12/04/17
Input geometry and model parameters	9/03/17	18/03/17
Aerodynamic calculation	19/03/17	7/04/17
Validation/Verification assessment	8/04/17	12/04/17
DEVELOPMENT OF THE FLIGHT SIMULATOR (GPSim)	13/04/17	6/06/17
Study of the simulator's model implemented in CIMNE's PARASIM	13/04/17	27/04/17
Development of a 6 DoF ram-air parachute flight simulator in MATLAB	28/04/17	22/05/17
Validation/Verification assessment	23/05/17	6/06/17
FINAL FORMATING OF THE PROJECT	7/06/17	9/06/17
Extra needed documents development	7/06/17	9/06/17
Conclusions and recommendations for the project follow-up	7/06/17	9/06/17



Chapter 1

Introduction

1.1 Literature Review

This section presents an overview about parachute technology and its historical development. The focus is mainly on parachutes characteristics and applications, and also on the typical techniques used for the dynamics analysis of parachute-payload systems.

1.1.1 Historical Background

The first evidence of the existence of parachutes (or parachute-like devices), dates back to the 12th century when the Chinese used them to descent animals and humans from high towers, as well as for circus-type stunts arranged to entertain guests at court ceremonials [2, 3]. It was in 1483 when the famous inventor Leonardo Da Vinci designed in his notebook the first parachute model and wrote this note:

> *"If a man is provided with a length of gummed linen cloth with a length of 12 yards on each side and 12 yards high, he can jump from any great height whatsoever without injury."*

In 1617, Fauste Veranzio built a similar device to Da Vinci's invention and jumped successfully from a tower in Venice. In 2000, Adrian Nicholas, a British skydiver, despite having been advised by experts that Da Vinci's invention would not fly, decided to test it and jumped from a hot-air balloon 3000 meters above the ground [4]. It was a successful descent so Da Vinci's invention officially turned into the first successful parachute design. In 1783, Joseph and Jacques Montgolfier, two famous balloonists, descended animals to the ground from rooftops or balloons. It is important to note that all parachutes were built with a rigid frame at that time. It was the 22nd of October 1797 when André-Jacques Ganerin made the first jump with a flexible parachute. He jumped successfully from a balloon over Paris. In one of the many Ganerin's jumps, Lalandes, a French astronomer, realised that the parachute made several oscillations during the descend and proposed to make a cut in the centre of the parachute to reduce them. The idea worked and supposed the birth of the so-called vent. Jumping with parachutes from balloons became a way of entertainment for spectators at that time, and many exhibition jumps were made during early 19th century. According to [3] in 1808 Jordaki Koparento performed the first emergency landing, descending safely from his burning balloon over the city of Warsaw. However, it is claimed in [5] that was Jean Pierre Blanchard in 1785 who made the first emergency use of a parachute after his balloon exploded.

At the outbreak of the World War I, participants on both sides used balloons for spying the enemy. When balloons were hit by bullets, the spies used parachutes to save themselves. French army went a step further evolving their parachute systems in a way that the entire basket could be retrieved. When aircrafts made the appearance in the war, initially, the pilots didn't use parachutes. The main reason was the difficulty to be propelled out of the aircraft cockpits. The time needed to put on the harness and to pick up the parachute container was too long, so the remaining altitude was not enough for a successful descent. However, since pilots were too valuable for the army to lose them, greater efforts in the development of parachute materials and packing were done to ensure their safety. The first life of a pilot was saved in 1916 with a parachute opened by static lines attached to the aircraft, which didn't open until the pilot was completely out the cabin. At this time, there was great scepticism about the ability of human to freefall for more than some seconds without fainting. Actually, although Georgina "Tiny" Broadwick made the first freefall jump in 1914, the military still not believed on it. It was in 1919 when Leslie Irvin convinced them, jumping from 1500 feet altitude (457.2 meters), freefalling some time and then opening the parachute. This fact created a new line of development for the following 50 years and changed the way of thinking about aircraft's rescue parachutes. The end of the war was characterised by great new developments of personnel parachutes like the Heinecke aircraft escape parachute (Figure 1) in Germany. In addition, the U.S Army, whose work is well described in [6], stablished new safety rules.

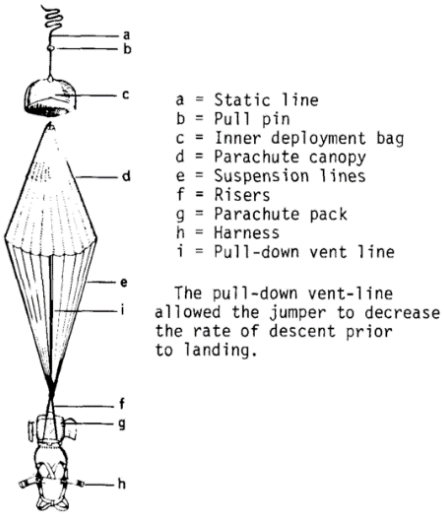


Figure 1: The Heinecke Parachute. Retrieved from [8]

In the 20's not much remarkable progress was made with the exception of the appearance of the first gliding parachute, the Hoffman Triangle Parachute (Figure 2) [7]. It is also worth to mention the first military personnel parachute standardisation in 1924 and the parachute recovery system of private airplanes (although it was not used at that time). In 1930, Soviet Army trained their soldiers in airborne operations, and this fact

forced the German to do the same. This type of operations turned to be (militarily speaking) advantageous some years later, in the early stages of second war.

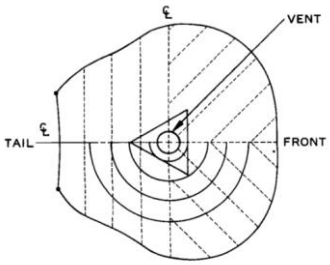


Figure 2: The Hoffman Triangle Parachute. Retrieved from [8]

The decade of 30's was characterised by a constant worldwide tension because of German rearmament and their rapid development of aircraft technology. Consequently, governments became interested in enhancing parachute technology and started sponsoring and providing support to those who worked on this sector. This reason, together with the beginning of World War II, made this period fruitful for parachute technology. Important studies about parachute aerodynamics and stability, opening and inflation technology, materials strength and porosity among others, were conducted during this decade. In addition, the family of "slotted parachutes" was born with the invention of ribbon parachute [8].

During the second war, improvements and inventions were made, especially in United Kingdom (UK), United States of America (USA) and Germany. Germans used parachutes to recover missiles, developed adjustable and retractable parachute dive breaks for their planes and designed some new parachute types, like the guided surface parachute. The Americans were pioneers in the use of airborne troops and delivery of supplies and equipment, as well as the first in using nylon for parachute construction. The British also progressed in high porosity shaped gore parachutes and the use of parachute clusters (Figure 3). At the end of the war, with the appearance of high-speed aircrafts, the manual procedure for ejecting the pilot out of the cockpit was no longer effective. This promoted the development of the ejection seat. Different approaches to the solution were made by different countries but according to [8], the original idea came from Sweden.

The period between the end of World War II and the sixties was absolutely profitable in terms of new parachute types and applications, especially in USA. High-altitude operations, supersonic speed parachutes, or massive airdrop using cluster parachutes, are some examples of the research made in this period. Extended skirt, annular, conical, ringslot, cross, ringsail and LeMoingne maneuverable parachutes are some of the types that were developed. The classification of these parachute types will be presented later in Section 1.1.2.

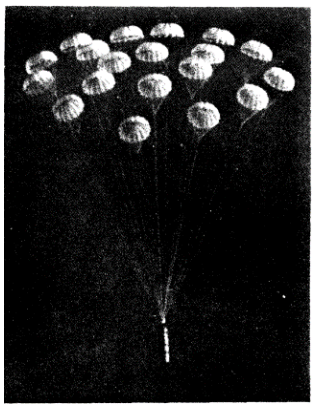


Figure 3: 19 parachutes cluster. Retrieved from [8]

In the early sixties U.S military invented the LAPES, a method used for the delivery of heavy equipment or supplies at low altitudes when the terrain is not optimum for landing (Figure 4).

Figure 4: C-130 Hercules dropping pallet with Low Altitude Parachute Extraction System on Khe Sanh runway. Retrieved from [9]

Also in the early sixties, the need of having control of the parachute trajectory in the air and to reach specific targets at landing, motivated the development of new maneuverable or gliding parachutes. These were designed with higher glide ratios than the previous ones, which gives origin to different sports applications, like skydiving.

In the line of gliding parachutes, the parawing or Rogallo wing (Figure 5) was invented by Francis and Gertrude Rogallo. Despite having some difficulties with permissions to carry out flight tests of the parachute, it was finally accepted and tested by NASA for space applications like re-entry vehicles recovery [10]. Initially, it seemed that it was going to be advantageous for some applications but high-speed performance was not satisfactory and the idea was finally discarded. Nevertheless, it was widely used in sport jumping during the 1970's and finally replaced by present parafoil or also called ram-air parachute (Figure 6).The latter was designed in the early 1960's by Domina Jalbert and is the type of parachute that will be studied in this work. A detailed description will be presented later on in section 1.1.2.

Figure 5: The Rogallo wing. Retrieved from [11]

Figure 6: Ram-air parachute example. Retrieved from [59]

During the 60's and the 70's parachutes were extensively developed and used in space missions. The Apollo spacecraft recovery in 1969, the Viking spacecraft landing on Mars in 1976 or the Space Shuttle solid boosters, recovery in 1977 were the most remarkable applications. Other example of parachute application that was studied at that time was their use for aircraft recovery when a failure occurred during the flight. The following cite was made by the founder of BRS parachutes, a company that works with airplane recovery parachute applications.

> *[12] Prior to 1975, few had attempted to implement the idea to design a parachute for an aircraft, even though it had been talked about for nearly a century. In that year, Boris Popov of Saint Paul, Minnesota, survived a 400-foot fall in a collapsed hang glider. "As I fell, I became most angry at my inability to do something," Popov explained. "I had time to throw a parachute. I knew they existed but they hadn't yet been introduced to the hang gliding community." This event led Popov to invent the whole-aircraft parachute system and to found Ballistic Recovery Systems (BRS) in 1980. 373 lives have been saved thanks to this invention (February 2017).*

In the same line of work, CIMNE (International Center for Numerical Methods in Engineering) has recently participated in the European project PARAPLANE, which aimed at improving the safety in air transport by the use of a Steerable Autonomous Parachute System (SARS) [13]. The project was led by the Spanish parachute manufacturer CIMSA Ingeniería de Sistemas and had the participation of important

European research centers and manufacturers. The first study cases in PARAPLANE were applications to small general aviation aircrafts, and also to a near-space capsule for space tourism (see Figure 7). However, they want to extend the system applicability to large aircraft or airborne systems in the future.

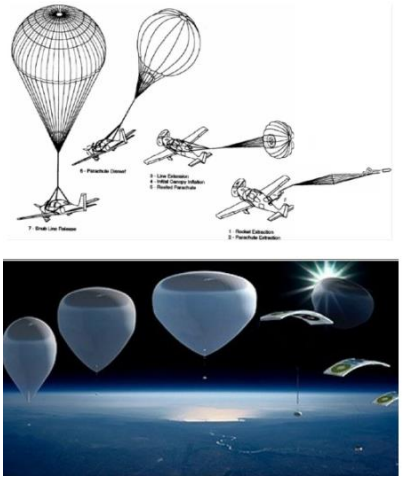


Figure 7: SARS application to general aviation aircraft (on top). Application to a near-space capsule for space tourism (on the bottom). Retrieved from [13]

Coinciding with the new popular concept of modular planes, a new possible use of parachutes for aircraft's safety has also appeared which is similar to the whole aircraft recovery parachute commented above. It consists on a plane that can detach its entire cabin in the event of an emergency and 'drop' passengers to safety using built-in parachutes. [14]

In the last three decades parachute systems have been used for an increasing number of applications and many technological advances have been made. Among them, we can highlight the use of smart materials for their construction and the development of guided systems and numerical tools for predicting their performance. An overview of these numerical methodologies is presented below in Section 1.1.3.

1.1.2 Parachute types and uses

The ram-air parachute studied in this project is just one among all the different existing types of parachutes. The aim of this section is to present a general classification of parachutes, as well as, to frame our focus of study within it.

Classifying parachutes can be a hard task because of their great number of designs and purposes. Actually, there is not a commonly accepted or standardised way of doing it and different classifications can be found in the literature. For the sake of simplicity, in this work, a division in three main groups is proposed: *conventional parachutes*, *rotational parachutes* and *maneuverable or gliding parachutes* (see Tables 2, 3 and 4). This classification is based on parachute flight characteristics.

Before presenting the classification of parachutes it is worth to mention that, although parachutes are the most commonly used decelerator recovery systems, there are other decelerator types which possess features of special advantage particularly in high temperature and high velocity performance regimes (see for instance [15] or [16]). For the current project purposes, only parachute type decelerators will be analysed in detail.

Conventional Parachutes	Solid Textile Parachutes	- Flat Circular - Hemispherical - Conical - Guide surface - Biconical - Annular - Triconical - Cross
	Slotted Textile Parachutes	- Flat Ribbon - Conical Ribbon - Ribbon (Hemisflo) - Disk-gap-band - Ringsail - Ringslot

Table 2: Conventional Parachutes

Rotational Parachutes	Rotafoil
	Vortex Ring
	Sandia RFD

Table 3: Rotational Parachutes

Maneuverable or Gliding Parachutes	TOJO, T & U slots, etc.
	LeMOIGNE
	Parawing
	Ram-air/Parafoil
	Sailwing
	Volplane

Table 4: Maneuverable or Gliding Parachutes

Conventional parachutes are designed to provide drag and no lift force and thus, its typical applications don't need a precise trajectory guidance. They are commonly subdivided in two groups depending on their construction: solid textile (Figure 8) and slotted textile (Figure 9). The first group doesn't have holes or cuts in the canopy (apart from the vent in the canopy apex) while the second one does.

Figure 8: Conventional solid textile parachute. Retrieved from [17]

Figure 9: Conventional slotted parachute. Retrieved from [18]

Rotational parachutes or RFD are constructed with a radial gap between the parachute gores that causes air loads to make one side bulge out, in some ways resembling an egg beater. As the air spills out, each gore (or rotor blade, in effect) is held at an angle of attack that causes it to move forward around the central axis of the parachute. In terms of flight characteristics, they have two main advantages: firstly, spinning provides gyroscopic stability that avoids possible pendulum-like movements; secondly, the action of centrifugal forces prevents the canopy contraction and makes the effective area bigger, thus, providing a better drag performance. This definition of RFDs is adapted from [19], where further detailed information can be found.

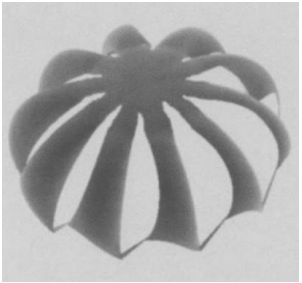


Figure 10: Rotating parachute (Sandia RFD type). Retrieved from [19]

Finally, *maneuverable or gliding parachutes* are those capable of generating a horizontal component of velocity (or drive) of the parachute and its payload [2]. Hence, it is possible for a system comprising a gliding parachute and payload to develop a resultant lift force. Ram-air parachutes, also called parafoils are included in this category. They consist of a flexible fabric wing with rectangular planform and a streamlined cross-

section which is opened at the leading edge to allow air to enter and inflate the wing to a specified shape. The payload is suspended on lines below the wing [20]. Parafoils are more expensive than other gliding parachutes or conventional parachutes but, on the other hand, they give higher gliding ratios and manoeuvrability, which allows them to be optimum for the development of precision aerial delivery systems. Many universities are carrying on research about this type of parachutes and different companies have developed commercial models, which mainly aim to the defence market and skydiving industry.

The U.S government has developed a program called Joint Precision Airdrop System (JPADS) that combines the U.S. Army's Precision and Extended Glide Airdrop System (PEGASYS) with the U.S. Air Force's Precision Airdrop System (PADS) to meet joint requirements for precision airdrop. Some examples of companies that develop products that fulfil the precision airdrop requirements under the JPADS program are [20]:

- Airborne Systems®. They have developed systems that can deliver payloads from 100 to 42,000 lbs (45 to 19050 Kg) and claim that they have "increased payload survivability due to the system's unique ability to perform a flared into-the-wind landing" (Figure 11)

- STARA Technologies Corp., which designs parachutes for medical supplies delivery. Their system is called MTRS (Mosquito Tactical Resupply System) and is capable of delivering payloads up to 150 pounds (68 Kg) within 30 meters of accuracy. (Figure 12)

Figure 11: from left ro right MicroFly II, FireFly and DragonFly models from Airborne Systems. Retrieved from [21]

Figure 12: Mosquito Tactical Resupply System (MTRS) from STARA. Retrieved from [22]

Tables containing information about parachute types' planform, profiles and common applications can be found in Appendix 4.

1.1.3 Parachute simulation and trajectory analysis

Numerical simulation of parachutes is not an easy task. It consists on a complex coupled aerodynamic-structural problem that usually involves a high computational cost. When a disturbance is introduced in the system, parachute geometry changes drastically to achieve a new equilibrium state, being the structural behaviour of parachute highly nonlinear. Not only the structural response is complex but also the simulation of the surrounding flow, which is highly unsteady. Consequently, several difficulties regarding the nonlinear behaviour of the structure, the interactions between the parachute components and flow detachment should be addressed. In addition, since the structure deforms continually under the aerodynamic loads applied, an iterative solution process is required, particularly in the presence of large unsteady wakes. This inherent complexity in numerical simulation of parachutes is the reason why most of the manufacturers rely mainly on empirical results. Despite this situation, different numerical methods have been developed to address parachute simulation problems. The most typical are described below.

1.1.3.1 Parachute simulations based on panel methods

Panel methods can be defined as a collection of techniques for the solution of general aerodynamic potential flows. In these methods, the flow is generally considered to be incompressible, inviscid and irrotational, so these are particularly well suited for flows with low-Mach and high-Reynolds numbers, where the effects of flow detachment or heat transfer are negligible. These techniques, which benefit from the linearity of the incompressible potential flow solution, model steady and unsteady complex flows as a combination or superposition of simpler basic or elementary flow solutions. Several applications have been also developed for linearized compressible flow problems. The fundamentals of panel methods will be presented in section 2.1.

In terms of computational cost, panel methods are advantageous for parachute aerodynamic simulation because they have no need of simulating the air surrounding the model, what noticeably simplifies calculations and reduces the computational cost. Successful applications of these methods have been developed for both rigid and non-rigid ram-air parachutes, see for instance [23]. Panel methods offer accurate enough results, useful for preliminary design purposes, and work particularly well when analysing gliding parachutes in nominal flight conditions with mostly attached flow. In spite of the fact that the potential flow assumptions could limit their applications, several improvements have made possible to broaden the scope of these methods. For example, panel methods don't allow the computation of viscous drag of the canopy, suspension lines and payload, but the introduction of semi-empirical corrections makes it possible. Moreover, some modifications can be introduced for the simulation of flow detachment and extensive wakes. These methods usually solve the Lagrangian form of the Navier–Stokes momentum equation written in vorticity variables. Since only the regions of concentrated vorticity must be resolved, the parachute wake is discretized by

means of point vortices while the attached (inviscid) flow areas remain mesh free. This addresses the issue of flow separation while retaining the advantages of panel methods [23].

In the line of panel methods, one of the simplest and easier way to solve three-dimensional potential flow solvers is by means of the Horseshoe Vortex Method (HVM). It is based on Prandtl's lifting line theory and allows computing lift, moment, induced drag and spanwise load distribution with satisfactory accuracy. Besides its simplicity, HVM has also some important drawbacks. Firstly, it does not model wing thickness or camber (the latter can be done with a special treatment), and secondly, the simple wake models usually employed cannot represent with accuracy the behaviour of highly unsteady wakes. Hence, for parachute aerodynamic simulation other low-order panel methods are usually used instead of HVM. The next approach (increasing accuracy but also complexity and cost) is the lifting surface method. Here, the wing has also zero thickness (only the mean camber surface of the wing is modelled), but arbitrary camber and improved wakes can be represented. Among the different types of low order panel methods, the Vortex Lattice Method (VLM) is one of the most common for the solution of thin wings (it is the choice adopted in this thesis).

In the VLM the lifting surfaces are typically represented by means of a lattice of vortex ring elements (Figure 13) lying along the mean camber surface of the wing. Each vortex ring is typically constructed by vortex lines segments generating a close polygonal panel (Figure 14), but also horseshoe vortices can be used. Since thickness is neglected, this method is only suitable for thin airfoils and lifting surfaces. In the VLM, the boundary conditions are satisfied on the actual body surface, which can have various shapes and planforms. References about this method can be found in [24]. Another approach in this line of lifting surface methods is the use of panels of doublets, which gives origin to the so-called Doublet Lattice Method (DLM). If the doublet distribution at each panel is constant, it can be demonstrated that this is similar to have a constant strength vortex ring [24].

It should be noted that VLM and DLM can account for thickness effects by complementing vortices or doublets distributions with sources. More complex (and higher-order) panel methods are also designed for the simulation of arbitrary geometries (Figure 15), see for instance [24].

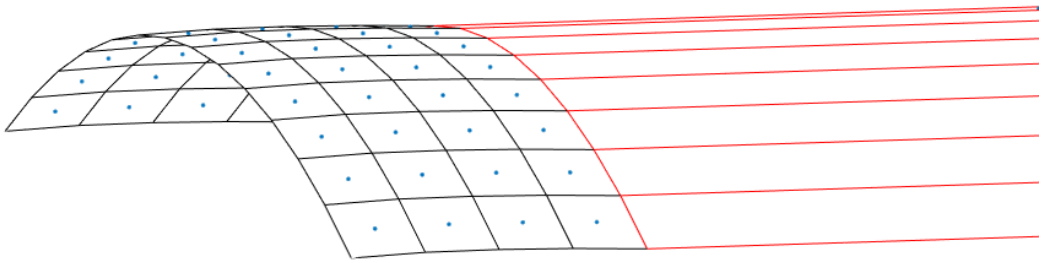


Figure 13: Lifting surface representation using a lattice of vortex ring elements. Steady case (just one row of wake panels)

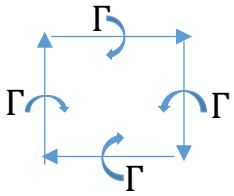


Figure 14: Vortex ring element composed by four vortex lines segments

In the case of ram-air parachutes, the canopy thickness is usually small for typical configurations (t/c<17%). Hence, its effects on the aerodynamics can be neglected and the canopy can be represented by means of its mean camber surface. This problem is one that can be solved in a very efficient manner with vortex or doublet lattice techniques where the DVL method is adopted in order to have exact match between the aerodynamic panels and the partitioning representing the wing geometry (vortex rings usually requires to be displaced in relation of the latter).

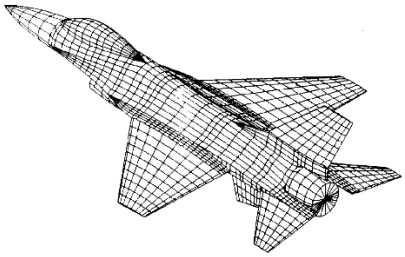


Figure 15: Discretization of a fighter plane with panel elements. Retrieved from [24]

1.1.3.2 Parachute simulations based on volume discretization methods

The aerodynamics of parachutes can also be determined by the solution of Navier-Stokes equations with a complete meshing of the surrounding flow. Different approaches can be used for the discretization of the domain but one of the most typical in parachute aerodynamics is the "Deforming-spatial-domain/stabilized space-time (DSD/ SST) methodology" [25, 26], a methodology based on Finite Elements Methods (FEM) which is particularly well-suited for the solution of problems with moving boundaries or interfaces. This formulation depends not just on time but also on space, or using the same words as those used in the literature, it is a "variational formulation

written in the space-time domain". Hence, when time advances, any spatial change that may occur in the domain would be automatically considered in this formulation. As the discretization of time and space is done in the space-time domain, if the boundary moves, the boundary nodes do the same having a new mesh after each time step. Some of the early applications of this method can be consulted on [27–30] and a recent review of its newer versions, special techniques and applications can be found in [31] and [32].

There are other methods based on FEM with entire meshing of the surrounding fluid. For example, in [33] a coupled Euler-Lagrange method which has the capability of modelling the fabric permeability and accounting for its effects is described; in [34, 35] the immersed boundary (IB) method is used to study the effects of parachute porosity in stability and [36] presents the application of large-eddy simulation of compressible flows to flexible supersonic disk-gap-band parachutes. Additionally, the software LS-DYNA® [37] has also been used to solve parachute problems [38, 39].

Despite having been proved to give reliable results in a wide range of problems of different complexity, all finite element based methods mentioned above share the characteristic of being computationally demanding. The complex behaviour of parachutes' flow field is reflexed on the computational code such that during the simulation, the mesh needs to be rebuilt almost constantly, what considerably lowers speed and robustness of the simulation. Hence, a very high computational power is required for these solutions, and thus, these methodologies are not affordable for small companies and developers. In addition, they are too expensive to be applied in practical engineering study cases.

1.1.3.3 Trajectory analysis

A parachute flight simulator integrates the equations of motion of the parachute-payload system under the effects of mass and aerodynamic forces. The parachute system is typically considered as a rigid body with different degrees of freedom, and the aerodynamic forces are obtained from simplified models based on aerodynamic derivatives. The latter are constant coefficients that express the dependence of the aerodynamic forces or moments with changes in the parachute geometry and attitude (like aerodynamic angles of attack, yaw, pitch, etc.). The number of DoFs depends mainly on the way the suspended payload is modelled. This can be treated as an independent rigid body having some kinematic links to the canopy or, as a rigid body solidary to the canopy (i.e. the whole parachute system is a single rigid body).

The computational models used to determine the trajectory of parachutes have evolved with time. In the first solution approaches, flight simulators up to 4 DoF were typically employed. No more DoF were simulated mainly due to the fact that, on the one hand, these require to adjust more aerodynamic coefficients; on the other hand, the complexity and cost of the equations of motion increase. Note that the calculation of aerodynamic derivatives, typically obtained from wind tunnel tests, is expensive, and

usually it is not possible to obtain accurate data to fit complex dynamic models. The improvement of computational capabilities in terms of memory and CPU speed made possible to deal with higher order flight simulators and to use aerodynamic prediction to complement the required wind tunnel data. These technological advances led to the appearance of new simulation tools based on low-order computational aerodynamics models and computational fluid dynamics (CFD) techniques.

In Table 5 below, there are some examples of the main variables used to describe the parachute attitude and trajectory according to the number of DoF.

Trajectory analysis	Point Mass	Planar Rigid Body	6 DoF Rigid Body	9 DoF. 2 Rigid Bodies
DoF	2	3	6	9
Variables	X,Z	X,Z,θ	X,Y,Z Ψ,θ,ϕ	X,Y,Z Ψ,θ,ϕ Ψ_P,θ_P,ϕ_P

Table 5: Examples of the major variables used in different types of parachute flight simulators. Retrieved from [40]

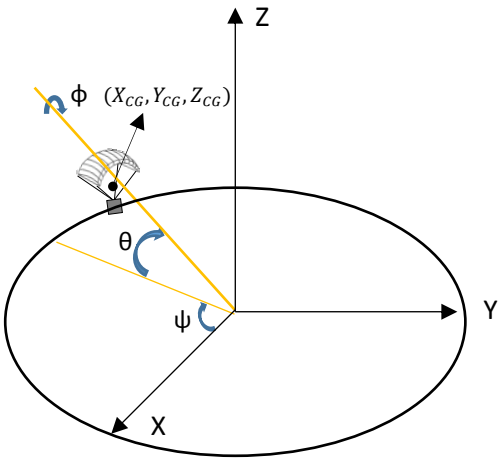


Figure 16: Graphical representation of the variables

In this thesis, a 6 d-o-f simulator model is proposed. It is based on PARASIM, a flight simulator developed at CIMNE which follows the general lines given by [41]. Experience and a multitude of published data demonstrate that 6 DoF models are effective and accurate enough for typical parachute applications (see for instance [20] and the application examples there presented). Further information about this type of simulators can be found in [42].

As shown in Table 5, the developed simulator (6 DoF) will work with x, y, z, Ψ, θ, and ϕ and their time derivatives, which are the linear and angular velocities of the center of mass of the parachute-payload system.

1.2 Basic aspects of parachute aerodynamics

The aim of this section is to present the most important concepts of the steady-state aerodynamics of parachutes, especially the ram-air type. Most of the information below has been adapted from [2, 43] which are the main references in the literature for this topic. The reader can find more detailed information in them.

1.2.1 Definition of axes and main concepts

To analyse a flight mechanics problem, first, a set of axes must be defined. For parachutes, body axes are typically chosen. These orthogonal axes represent the parachute as a rigid body and pass through an origin which is attached to the parachute and moves with it. This origin can be placed at different locations, usually those that simplify the analysis are chosen. Regarding the axes, conventional parachutes are axially symmetric. The O-x axis is contained in the axis of symmetry and points to the forward direction. O-z axis is orthogonal to O-x axis but no specific direction is given to it, leaving its definition arbitrary. Finally, O-y is also perpendicular to O-x and it satisfies the right hand rule:

$$\hat{\imath}\, x\, \hat{\jmath} = \hat{k} \quad (1)$$

For gliding parachutes, O-x axis points to the forward direction and is contained in the longitudinal plane of symmetry. O-z axis is also contained in the plane of symmetry and at the same time is orthogonal to O-x. Finally, O-y axis is satisfies the right hand rule (see Figure 17).

Note that, in the left picture, the parachute is conventional and the origin is placed at the vent, while, in other picture, the parachute is ram-air type and the origin is placed at the C.G.

Figure 17: On the left, body axes represented for a conventional parachute. On the right, body axes represented for a ram-air parachute. Retrieved from [44, 45]

Terms and symbols used for the aerodynamic analysis and flight mechanics are standardised in ISO-1151 [46]. Following these standard terms and symbols, the most important concepts are described below (Figure 18).

Note that the angle of attack (α) is defined as the angle between the projection of aerodynamic speed in the O-xz plane and the O-x axis.

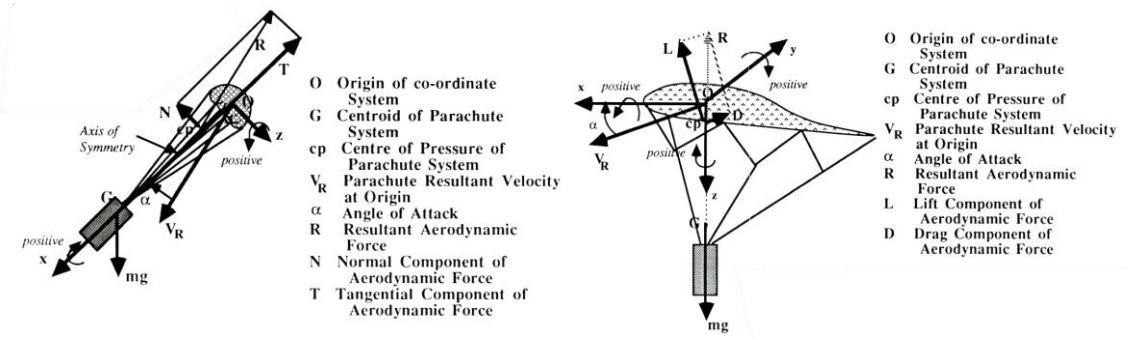


Figure 18: Symbols and forces for conventional (left) and gliding parachutes (right). Retrieved from [2]

When a parachute moves through the air, a total or resultant aerodynamic force and moment are generated due to the pressure and shear stress distributed loads caused by the relative movement between the parachute and the surrounding fluid. But, where is the resultant aerodynamic force applied? It must be placed at that point where it produces the same aerodynamic moment than the distributed loads. This point is called the centre of pressure and changes its position when the forces change. In most aerodynamic applications, approximations are usually employed to determine its location. For gliding parachutes, its position is defined as being at the intersection of the resultant aerodynamic force line of action with the chord line of the airfoil section. For conventional parachutes, it is defined to be at the intersection of the line of action of the resultant aerodynamic force with the parachute axis of symmetry [2] (see Figure 18). However, due to its changing position, it is not always a convenient concept for aerodynamics analysis. Instead, the aerodynamic centre is used, which is defined as that point where the aerodynamic moment is independent of α (this may also change with parachute deformation).

For conventional parachutes, the resultant aerodynamic force is typically decomposed into tangential and normal components. Their usual axial symmetry makes possible a 2-dimensional analysis of the problem (Figure 18 left). The tangential component is defined as that component aligned with its axis of symmetry (O-x axis) and perpendicular to O-z axis while the normal component, also contained in the Oxz plane, is aligned with O-z axis and perpendicular to O-x. These forces are made non-dimensional as follows:

$$C_T = \frac{F_T}{\frac{1}{2} \cdot \rho \cdot V_{ref}^2 \cdot S_{ref}} \quad (2) \qquad C_N = \frac{F_N}{\frac{1}{2} \cdot \rho \cdot V_{ref}^2 \cdot S_{ref}} \quad (3)$$

These coefficients are function of α, Re, and Mach (Ma). Re can be defined as:

$$Re = \frac{\rho \cdot V_{ref} \cdot D_{ref}}{\mu} = \frac{V_{ref} \cdot D_{ref}}{\nu} \quad (4)$$

where D_{ref} is:

$$D_{ref} = \sqrt{\frac{4 \cdot S_{ref}}{\pi}} \quad (5)$$

In the case of gliding parachutes the typical way to decompose the resultant aerodynamic force is onto wind axes. Those axes are defined such that its origin is contained in the longitudinal plane of symmetry (usually the CG), the O-x axis is parallel to the aerodynamic speed at each moment, the O-z axis is perpendicular to O-x and it is contained in the plane of symmetry. Finally, O-y axis follows the right hand rule. The components of the aerodynamic force projected in these axes are called lift (O-z) drag (O-x) and lateral force (O-y) (Figure 18). Lift and lateral components are perpendicular to the aerodynamic speed vector while drag is parallel. When thinking about parachutes and their main role, drag is usually considered the most important aerodynamic component, but is the lift force what makes gliding parachute different from other types. Similarly to tangential and normal force, lift and drag also can be expressed in dimensionless form:

$$C_L = \frac{L}{\frac{1}{2} \cdot \rho \cdot V_{ref}^2 \cdot S_{ref}} \quad (6) \qquad C_D = \frac{D}{\frac{1}{2} \cdot \rho \cdot V_{ref}^2 \cdot S_{ref}} \quad (7)$$

and are also function of α, Re and Ma.

In two dimensions, the lift and drag coefficients can be related with tangential and normal forces coefficients through the following geometric relations (Figure 19):

$$C_D = C_T \cdot \cos(\alpha) + C_N \cdot \sin(\alpha) \quad (8)$$
$$C_L = C_N \cdot \cos(\alpha) - C_T \cdot \sin(\alpha) \quad (9)$$
$$C_T = C_D \cdot \cos(\alpha) - C_L \cdot \sin(\alpha) \quad (10)$$
$$C_N = C_L \cdot \cos(\alpha) + C_D \cdot \sin(\alpha) \quad (11)$$

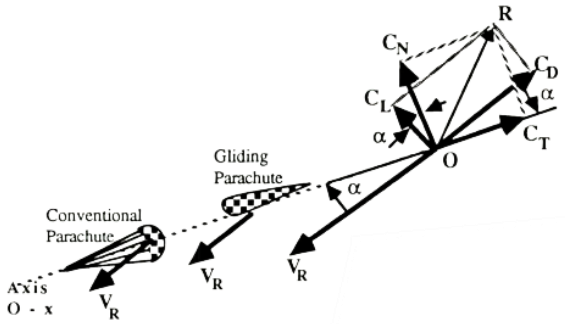


Figure 19: Relationships between the aerodynamic coefficients C_L and C_D and the coefficients C_N and C_T. Retrieved from [2]

The extension to a three-dimensional analysis of the problem is straightforward just including the lateral forces terms.

1.2.2 Models for parachute aerodynamics forces

The aim of this section is to present approximate models for the lift and drag coefficients of gliding parachutes. Similar as occurs for rigid wings, the lifting line theory (LLT) can be used here to obtain an approximate model, but some corrections are necessary to account for low aspect ratio (AR) effects (about 3-4 in typical gliding parachutes). In this case, the three dimensional flow along the chord and wake interaction become more significant and cannot be properly modelled by a single vortex line spanning the section, as occurs in LLT. To tackle this problem, Lingard [43] proposed a modification of the lifting line theory suitable for low aspect ratios. This is presented next, following the main lines given in [47].

For wings with AR > 5, the lift coefficient is linear with α and can be written as

$$C_L = C_{L_\alpha} \cdot (\alpha - \alpha_{ZL}) \quad (12)$$

where the wing's lift slope (C_{L_α}) can be obtained from the section value (C_{l_α}) as in [48]

$$C_{L_\alpha} = \frac{\pi \cdot C_{l_\alpha} \cdot AR}{\pi \cdot AR + C_{l_\alpha} \cdot (1+\tau)} \quad (13)$$

In case of gliding parachutes wing (AR < 5), these expressions are not fully applicable. The experimental results show that the wing's lift slope reduces causing the lift curve loses linearity. These effects can be accounted for by adding a non-linear term to the C_L

$$C_{L_{gliding\ parachute\ wing}} = \left(C'_{L_\alpha} \cdot (\alpha - \alpha_{ZL})\right)_{linear} + (\Delta C_L)_{non-linear} \quad (14)$$

and also a correction to the lift slope. Hoerner [47] proposes to modify the section lift slope by means of a factor K, i.e.

$$C'_{l_\alpha} = C_{l_\alpha} \cdot K \quad (15)$$

being K given by

$$K = \frac{2 \cdot \pi \cdot AR}{C_{l_\alpha}} \cdot \tanh\left(\frac{C_{l_\alpha}}{2 \cdot \pi \cdot AR}\right) \quad (16)$$

Then, Eq. (14) is still applicable, but with C_{l_α} calculated by Eq. (15)

$$C'_{L_\alpha} = \frac{\pi \cdot AR \cdot C'_{l_\alpha}}{\pi \cdot AR + C'_{l_\alpha} \cdot (1+\tau)} \quad (17)$$

Using Eq. (17) the linear part of the lift coefficient (Eq. 13) can be calculated. The nonlinear term is usually a function of AR, wing's lateral edges shape, and the component of velocity normal to the wing. In [43] it is stated that this non-linear component is still not well understood, but it appears to be caused by the drag based on the normal velocity component. An expression for this lift increment caused by the non-linear term given by [47] is

$$\Delta C_L = K_1 \cdot \sin^2(\alpha - \alpha_{ZL}) \cdot \cos(\alpha - \alpha_{ZL}) \quad (18)$$

For $1 < AR < 2.5$ experimental data shows that $\mathrm{K}_1 = 3.33 - 1.33 \cdot \mathrm{AR}$ and for $\mathrm{AR} > 2.5$, $\mathrm{K1} = 0$. Hence, the lift coefficient analytical expression can be finally expressed as

$$C_L = C'_{L_\alpha} \cdot (\alpha - \alpha_{ZL}) + K_1 \cdot \sin^2(\alpha - \alpha_{ZL}) \cdot \cos(\alpha - \alpha_{ZL}) \quad (19)$$

When designing a wing for a plane, it is known that the aerodynamic performance is not the same if it is analysed solely or attached to the fuselage. Aerodynamic interferences exist in the latter case, causing the wing to show a different behaviour. The same happens with a parachute. The previous development is valid for a ram-air wing with no interferences.

It is important to note that, apart from aerodynamic interference, when a flexible canopy is a part of a parachute system, the load distribution of the suspended mass forces the canopy to curve according to the configuration of the suspension lines. This originates the appearance of arc-anhedral (γ). According to [47], for a wing with dihedral angle, the C_L can be expressed by

$$C_L = C_{L_{\gamma=0}} \cdot \cos^2(\gamma) \quad (20)$$

where γ is defined as in Figure 20 top. It can be shown that the same equation applies for arc-anhedral wings if γ is defined as shown in figure 20 bottom.

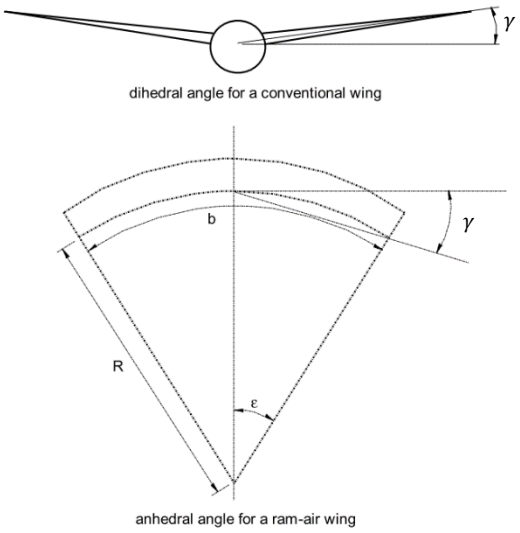


Figure 20: Definition of anhedral angle for a ram-air wing. Retrieved from [43]

From Figure 20.b it is derived that:

$$\gamma = \frac{\varepsilon}{2} = \frac{b}{4R} \quad (21)$$

It is important to note that Eq. (20) affects only the linear part of the C_L. Taking this into account, and introducing Eq. (20) into Eq. (19), the C_L can be expressed as

$$C_L = C'_{L_\alpha} \cdot (\alpha - \alpha_{ZL}) \cdot \cos^2(\gamma) + K_1 \cdot \sin^2(\alpha - \alpha_{ZL}) \cdot \cos(\alpha - \alpha_{ZL}) \quad (22)$$

Regarding the drag coefficient, it is composed of two parts, one is C_{D_P}, which is the contribution of the profile drag C_{d_p} along the span, given by

$$C_{D_P} = \frac{1}{S_{ref}} \int_{-\frac{b}{2}}^{\frac{b}{2}} C_{dp}\big(C_l(y)\big)c(y)dy \qquad (23)$$

The other part is the induced drag (C_{D_i}), also called drag due to lift, which can be calculated from LLT as

$$C_{D_i} = \frac{C_L^2}{\pi \cdot AR \cdot e} = \frac{C_L^2(1+\delta)}{\pi \cdot AR} \qquad (24)$$

Hence, the total canopy drag yields

$$C_D = C_{D_P} + C_{D_i} = \frac{1}{S_{ref}} \int_{-\frac{b}{2}}^{\frac{b}{2}} C_{dp}\big(C_l(y)\big)c(y)dy + \frac{C_L^2(1+\delta)}{\pi \cdot AR} \qquad (25)$$

being *e* and δ efficiency parameters depending mainly on the wing geometry. These relate the performance of the wing to that of an elliptical wing (planform with the lowest induced drag and higher lift slope).

As previously mentioned, ram-air wings have low aspect ratio and, consequently, an increment of lift coefficient is observed due to the introduction of the non-linear term (18). This also causes an increase of the drag coefficient (related to the induced term). In [43] the following expression is used to represent the non-linear drag increase

$$\Delta C_D = K_1 * \sin^3(\alpha - \alpha_{ZL}) \quad (26)$$

Using E1. (26), the total drag coefficient for ram-air wings results

$$C_D = C_{D_P} + \frac{C_{LC}^2}{\pi \cdot AR \cdot e} + K_1 \cdot \sin^3(\alpha - \alpha_{ZL}) \quad (27)$$

where C_{LC} represents the circulation lift, calculated by

$$C_{LC} = C'_{L_\alpha} \cdot (\alpha - \alpha_{ZL}) \quad (28)$$

Note that this coefficient does not include the non-linear lift increment given by Eq. (18) because in Eq. (27) non-linearity is accounted by the third term.

When analysing the parachute system and not only the isolated wing, other drag contributions appear. Besides the drag caused by the wing, lines and payload also generate meaningful drag that must be taken into account. Assuming that all lines have the same length and see the same transversal velocity ($V_{ref} \cdot \cos(\alpha)$, being V_{ref} the system reference velocity), reference [43] proposes the following analytical expression for the drag of the suspension lines

$$C_{D_l} = \frac{n \cdot R \cdot d \cdot \cos^3(\alpha)}{S_{ref}} \qquad (29)$$

and the payload drag coefficient ($C_{D_{PL}}$), which depends on the shape and characteristics of the payload carried, is usually obtained from experimental data. Then, the total drag coefficient of the parachute results

$$C_D = C_{D_P} + C_{D_l} + C_{D_{PL}} + \frac{C'_{L_\alpha}(\alpha - \alpha_{ZL})^2}{\pi \cdot AR \cdot e} + K_1 \cdot \sin^3(\alpha - \alpha_{ZL}) \qquad (30)$$

Chapter 2

Aerodynamic modelling

2.1 Basic aspects of potential flow solvers

Regarding the solution of potential flows, two approaches can be made: analytical or numerical. The first one is only applicable for problems having very simple geometries and boundary conditions. The second one allows the treatment of more realistic geometries and the fulfilment of the boundary conditions on the actual surface [24]. In this section, the numerical approach followed for the solution of the aerodynamic problem is described, but before that it is necessary to present some fundamentals of potential flows.

2.1.1 Potential flow equations

Potential flow is the name given to an irrotational, inviscid flow. It is usually confused with ideal flow; which is characterised by being incompressible and inviscid. Potential flows can also be divided in compressible/incompressible and steady/unsteady. The governing equation of incompressible potential flow solvers (Laplace equation) is derived next from the Navier-Stokes equations.

The Navier-Stokes equations can be written in differential form as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{v}) = 0 \quad (31)$$

$$\frac{\partial (\rho \vec{v})}{\partial t} + \nabla \cdot (\rho \vec{v}\vec{v}) + \nabla p - \nabla \cdot \vec{\tau} - \rho \vec{g} - \vec{f}^{e} = 0 \quad (32)$$

$$\frac{\partial (\rho e)}{\partial t} + \nabla \cdot (\rho \vec{v} e) + \nabla \cdot \vec{q} + \nabla \cdot (p\vec{v}) - \nabla \cdot (\vec{v} \cdot \vec{\tau}) - \rho \vec{v} \cdot \vec{g} - \vec{E} \cdot \vec{j} = 0 \quad (33)$$

where ρ is the fluid density, $\vec{v}$ is the velocity, p is the pressure, $\vec{q}$ is the heat flux and $\vec{f}^{e}$ and $\vec{E} \cdot \vec{j}$ are the electromagnetic force and energy contributions, respectively. The stress tensor $\vec{\tau}$ can be written for a Newtonian fluid as $\vec{\tau} = \mu(\nabla \vec{v} + \nabla \vec{v}^{T}) - \frac{2}{3}\mu(\nabla \cdot \vec{v})\vec{I_1}$. In addition, an equation of state must be adopted for the fluid (perfect gas is typically adopted).

Now let the electromagnetic forces terms be neglected. Then, assuming the flow to be incompressible (Mach < 0.2-0.3), these equations are simplified to:

$$\nabla \cdot \vec{v} = 0 \quad (34)$$

$$\rho_0 \frac{\partial \vec{v}}{\partial t} + \rho_0 \nabla \cdot (\vec{v}\vec{v}) + \nabla p - \mu \nabla^2 \vec{v} - \rho_0 \vec{g} = 0 \quad (35)$$

$$\rho_0 \frac{\partial e}{\partial t} + \rho_0 \nabla \cdot (\vec{v} e) + \nabla \cdot \vec{q} + \nabla \cdot (p\vec{v}) - \nabla \cdot (\vec{v} \cdot \vec{\tau}) - \rho \vec{v} \cdot \vec{g} = 0 \quad (36)$$

Under perfect gas assumption ($du = C_p dT$) it can be shown that the energy equation is finally simplified to:

$$\rho_0 C_p \left(\frac{\partial T}{\partial t} + \vec{v} \cdot \nabla T\right) = \nabla \cdot (\kappa \nabla T) + \vec{\tau} : \nabla \vec{v} \quad (37)$$

In addition, assuming that the viscous dissipation term (last term of the equation) can be neglected, energy equation converts into a heat conduction equation and is uncoupled from the mass and momentum equations. Note that mass conservation equation (34) is now a constraint over the velocity field.

For the following procedure, only mass and momentum conservation equations will be considered. Focusing on typical aerodynamic problems (Re >> 1 & Fr >> 1), it is possible to demonstrate that the effects of viscosity and gravity can be neglected in the momentum equation. Hence, the fluid equations result

$$\nabla \cdot \vec{v} = 0 \quad (38)$$

$$\frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} = -\frac{1}{\rho_0} \nabla p \quad (39)$$

Eq. (39) also can be written in terms of the fluid vorticity [49]. This leads to

$$\frac{D\vec{\omega}}{Dt} = \frac{D(\nabla \times \vec{v})}{Dt} = 0 \quad (40)$$

which can be derived as follows.

From vector algebra, it is known that $\vec{v} \cdot \nabla \vec{v} = \nabla \frac{V^2}{2} - \vec{v} \times (\nabla \times \vec{v}) = \nabla \frac{V^2}{2} - \vec{v} \times \vec{\omega}$. Hence, applying the rotational operator to both sides of Eq. (39) and using the mentioned algebra identity, the following expression is obtained

$$\frac{\partial \vec{\omega}}{\partial t} - \nabla \times (\vec{v} \times \vec{\omega}) = 0 \quad (41)$$

Expanding the second term of (41) and simplifying

$$\frac{\partial \vec{\omega}}{\partial t} - (\cancel{\vec{v} \nabla \cdot \vec{\omega}}^{\nabla \cdot (\nabla \times \vec{v}) = 0} - \vec{v} \cdot \nabla \vec{\omega} + \cancel{\vec{\omega} \cdot \nabla \vec{v}}^{\vec{\omega} \perp \nabla \vec{v} = 0} - \cancel{\vec{\omega} \nabla \cdot \vec{v}}^{\nabla \cdot \vec{v} = 0}) = 0$$

Therefore

$$\frac{\partial \vec{\omega}}{\partial t} + \vec{v} \cdot \nabla \vec{\omega} = 0 \rightarrow \frac{D\vec{\omega}}{Dt} = 0 \quad (40)$$

This result shows that the vorticity for each fluid element remains constant (what does not mean irrotational flow). Then, applying the assumption of irrotational flow, from equation (40) we obtain:

$$\vec{\omega} = \nabla \times \vec{v} = 0 \quad (42)$$

Consequently, velocity should derive from a potential function ($\vec{v} = \nabla\phi$). Note that this expression satisfies Eq. (42) ($\nabla \times \nabla\phi = 0$). Finally, introducing Eq. (42) into the mass conservation equation (38) results

$$\nabla^2\phi = 0 \quad (43)$$

Eq. (43) is the Laplace's equation, which governs the behaviour of incompressible potential flows. It consists on a second order PDE of elliptic type that can be solved with suitable boundary conditions. It is important to highlight that the solution of Eq. (43) is linear and, thus, the principle of superposition can be applied (linear combination of particular solutions is also a solution of the problem). This allows the possibility of simulating complex flow patterns by adding simpler elementary flows such as doublets, sources, vortices or any combination of them, which are solution of the Laplace's equation by construction [50].

In panel methods, the analysis domain is supposed to be enclosed by outer (freestream) and interior boundary surfaces (the body of analysis), and a thin wake extending from the latter. This delimits two regions, one corresponding to the fluid to be resolved and another is enclosed by the body and the wake. For this problem, a general solution of Eq. (43) can be constructed only in terms of surface integrals of the potential and its derivatives by means of the Green's theorem [24].

Profiting from this advantage, panel methods use distributions of these simpler flows (singularity elements) along the problem boundaries and the solution reduces to finding the strength of the singularity elements distributed. This numerical approach is more economical, from the computational point of view, than methods that solve for the entire flowfield in the whole fluid volume (e.g., finite volume methods) [24].

2.1.2 Problem definition, boundary conditions and physical considerations

As mentioned above, the solution of Laplace's equation for an incompressible potential flow can be constructed by the sum of source, vortex and doublet distributions placed on the body surfaces (domain boundaries). With source elements, body thickness effects are simulated, while vortices (or doublets) elements are necessary for modelling lifting problems. In general form we can write

$$\phi^* = \phi + \phi_\infty \quad (44)$$

where ϕ is the potential of the vortex/doublet distribution along the body surface and wake, and the source distribution on the body surface. ϕ_∞ represents the freestream potential (known from problem data). As already mentioned, this thesis focuses on low thickness parachutes. Hence, source distributions are not considered.

Since vortices, doublets and sources are designed to satisfy Laplace's equation (harmonic functions), to find a particular solution of the problem it is only necessary to

apply proper boundary conditions along the domain boundaries. These are typically the potential for freestream boundaries and a zero normal velocity along the solid body surfaces (slip condition). However, boundary conditions are not enough to find a unique (physical) solution for the potential problem unless the total circulation is fixed. This requires to introduce some physical considerations which are mainly related with experimental observations of the flow field; the so-called Kutta condition [24]. The latter says that the total value of circulation must be that which makes the flow to leave the T.E smoothly. In Section 2.3.2 it will be explained how is this condition satisfied.

Regarding the boundary conditions (B.C.), there two different types:

- *Neumann B.C.*: with this type of B.C., it is directly specified that the velocity component normal to the body surface is zero or, in other words, that the potential derivative in the normal direction is zero. Both
- *Dirichlet B.C.*: with this type of B.C. a specific potential is determined on the boundary.
- *Mixed B.*C.: The combination of the two previous is also possible and is then called mixed B.C.

2.1.3 Development of a numerical solution/potential flow solver

Potential flow solvers work with algebraic equations to obtain results. Hence, once the problem is uniquely defined with its boundary conditions and corresponding physical considerations it is reduced to a system of linear algebraic equations (based on the boundary conditions) defined by a matrix and two vectors which are explained later.

$$\begin{pmatrix} influence \\ coeficients \\ matrix \end{pmatrix} \begin{pmatrix} strenghts \\ vector \end{pmatrix} = \begin{pmatrix} RHS \\ vector \end{pmatrix} \quad (45)$$

There are some common steps towards constructing a numerical solution of this system or, in other words, developing a potential flow solver [24]:

1. ***Selection of singularity elements***: decisions such as the choice between vortex or doublet distributions, and whether source distributions are needed or not, are made in this first step.
2. ***Discretization of geometry (meshing procedure)****:* secondly, both body and wake are divided in "panels", each of them having an associated element (e.g. vortex ring) with its corresponding collocation point where the boundary condition must be applied (see Figure 21). Those elements are defined by nodes, and nodes coordinates are defined with functions or read directly from input files.

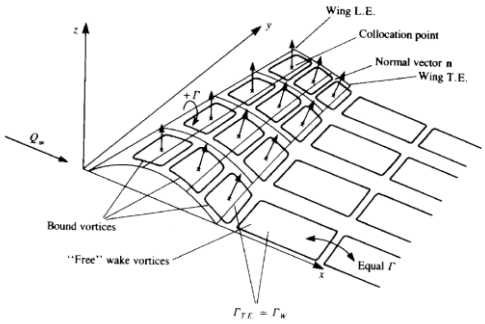


Figure 21: Vortex elements distribution on the body surface and the wake. Retrieved from [24]

The set of nodes that define the panels is called grid or mesh. An important remark is that results obtained depend on the selected grid. The more panels used, the better the results approximation but also the higher the computational cost is. For that reason, a convergence analysis is tipically done to select the optimum grid size in base to error and cost considerations.

3. ***Influence coefficients****:* At this stage the influence coefficients matrix is constructed. Assuming unitary strength singularity elements and by means of an iterative loop, a subroutine which calculates element influences (e.g. induced velocity by a vortex ring) is applied in each collocation point. This procedure automatically generates the matrix.
4. ***Establish RHS (Right-hand side)****:* The RHS vector represents the known part of the equation. For example, it includes the free stream velocity.
5. ***Solution of the system****:* Once the influence coefficients matrix and RHS vector are known it is easy to calculate the element strengths vector solving the system (47). Hence, the unknowns of the system are obtained.
6. ***Post-process****:* Finally, when the strength of each singularity element is known, secondary computations such as forces, moments, pressures, etc. can be done. It is also possible to implement the wake rollup here. This is the representation of the wake change of shape due to the induced velocities of both the body surface and the wake itself. Sometimes this requires an iterative solution procedure.

   For an unsteady case, the solution algorithm would be typically something similar to one in the next page.

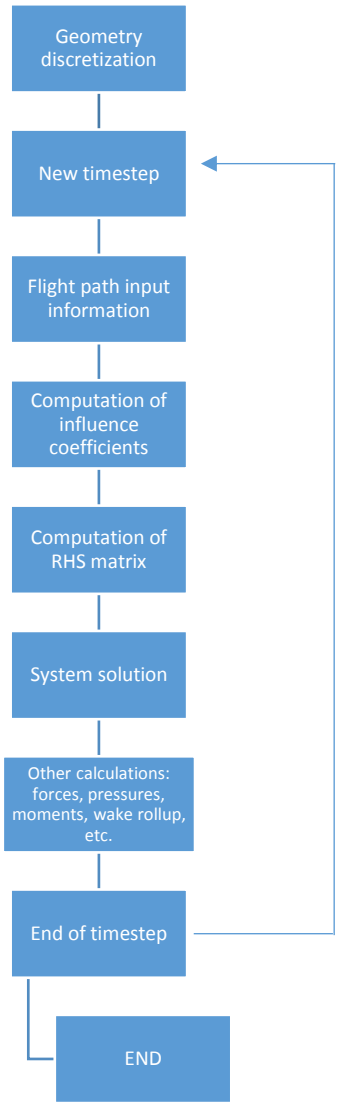


Figure 22: Flowchart of the typical algorithm of potential flow solvers. Adapted from [24]

2.2 Unsteady vortex lattice method (UVLM)

In the previous section, the common steps towards the construction of a numerical solution to the potential flow problem have been explained. The aim of the current section is to explain how this development works in the specific case of UVLM (the potential flow solver used in this thesis). The general procedure is explained here, while the detailed description of the code programming is presented in Section 2.3.

- ***Singularity elements**:* The type of singularity element used in this method is the vortex ring for both the wing and the wake. It consists on a combination of four vortex segments with the same circulation value (see Figure 23).

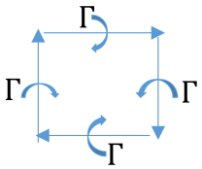


Figure 23: Vortex ring element composed by four vortex lines.

  The vortex ring top vortex line is called "leading edge bound vortex", the one in the bottom "trailing edge bound vortex", and the two lateral ones are the "trailing vortex lines".

- ***Discretization and grid generation***: This method works with the mean camber surface of the wing because it does not simulate thickness (no source distribution is needed). The lifting surface shape is usually given by $Z = \eta(X, Y)$ and is divided into M spanwise and N chordwise panels. Each panel has an associated vortex ring with the corresponding collocation point. There are different relative positions between panels and vortices (the alternative selected in this thesis is explained in the next section). For the computation of geometrical information such as vortex ring area, normal vector and the coordinates of collocation points, a subroutine is typically used. Regarding wake discretization, as already mentioned it is also modelled with vortex rings, and a time marching procedure is selected. In this way, to simulate the wake shedding (process by which the wake is developed downstream), initially just a little wake panel row is present, and then, after each time increment, a new wake panel row is shed and the wake coordinates are updated.
- ***Influence coefficients**:* these are calculated by means of a subroutine in which the velocity induced by each vortex ring at each of the collocation points is computed (assuming unitary circulation). Hence, an influence coefficients matrix of (M x N) rows and (M x N) columns is generated.
- ***RHS vector**:* The boundary condition of null velocity in the direction normal to the surface can be written for a specific collocation point "k" as:

$$a_{k1}\Gamma_1 + a_{k2}\Gamma_2 + a_{k3}\Gamma_3 + \cdots + a_{kN_p}\Gamma_{N_p} + [U(t) + u_w, V(t) + v_w, W(t) + w_w]_k \cdot \vec{n}_k = 0 \quad (46)$$

Then, moving the known terms to the RHS, we obtain:

$$a_{k1}\Gamma_1 + a_{k2}\Gamma_2 + a_{k3}\Gamma_3 + \cdots + a_{kN_p}\Gamma_{N_p} = -[U(t) + u_w, V(t) + v_w, W(t) + w_w]_k \cdot \vec{n}_k \quad (47)$$

where a_{kL} are the influence coefficients of vortex "L" at collocation point "k", Γ_L are the circulation values of each "L" vortex, and $(U(t),\ V(t),\ W(t))_k$ and $(u_w, v_w, w_w)_k$ are the aerodynamic speed vector and the wake induced velocity vector at the collocation point "k". The RHS vector then corresponds to the right hand side of the equation to solve (Eq. 47).

- ***System solution***: Now, the system is defined ($\bar{\bar{A}} \cdot \vec{\Gamma} = \overrightarrow{RHS}$) and it is only necessary to invert the influence coefficients matrix to obtain the circulation value of each vortex element:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1N_p} \\ a_{21} & a_{kL} & \cdots & a_{2N_p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N_p1} & a_{N_p2} & \cdots & a_{N_pN_p} \end{pmatrix} \begin{pmatrix} \Gamma_1 \\ \Gamma_2 \\ \Gamma_3 \\ \vdots \\ \Gamma_{N_p} \end{pmatrix} = \begin{pmatrix} RHS_1 \\ RHS_2 \\ RHS_3 \\ \vdots \\ RHS_{N_p} \end{pmatrix} \quad (48) \quad \rightarrow \quad \vec{\Gamma} = \bar{\bar{A}}^{-1} \cdot \overrightarrow{RHS}$$

With Γ known, further computations can be done.

2.3 Numerical implementation

2.3.1 Model geometry and panelling

In this section, firstly a general description of how geometrical parameters are defined is introduced. Then, an overview of the program implementation is given. This includes the definition of panel variables used to model the lifting surface, the explanation of the way of storing and filling geometry matrices and a comment on the treatment of the wake. Finally, the method used to define control surfaces is described.

In the situations analysed by the flight simulator and consequently, by the aerodynamic solver, the parachute canopy is supposed to be fully inflated and flying under nominal conditions. Moreover, the system will be considered as a rigid body and canopy aerodynamic calculations will be made using the mean camber surface. Hence, a parametric representation is enough for correctly modelling the geometry of a ram-air parachute in the considered environment of the problem. To do so, the following main variables are defined (see Figure 24).

- **Maximum canopy height (a)**
- **Canopy span (b)**
- **Canopy chord (c_{root})**
- **Taper ratio (λ)**
- **Dihedral angle (γ)**
- **Sweep angle (Λ_{LE})**

Control surfaces (c.s) parameters:

- **Control surface chord (c_f)**
- **Control surface span (b_f)**
- **Exterior offset (OE): Distance from the wing tip to the control surface.**
- **Right c.s deflection (δ_R)**
- **Left c.s deflection (δ_L)**

Note that no thickness parameter is needed because calculations are made with the mean camber surface.

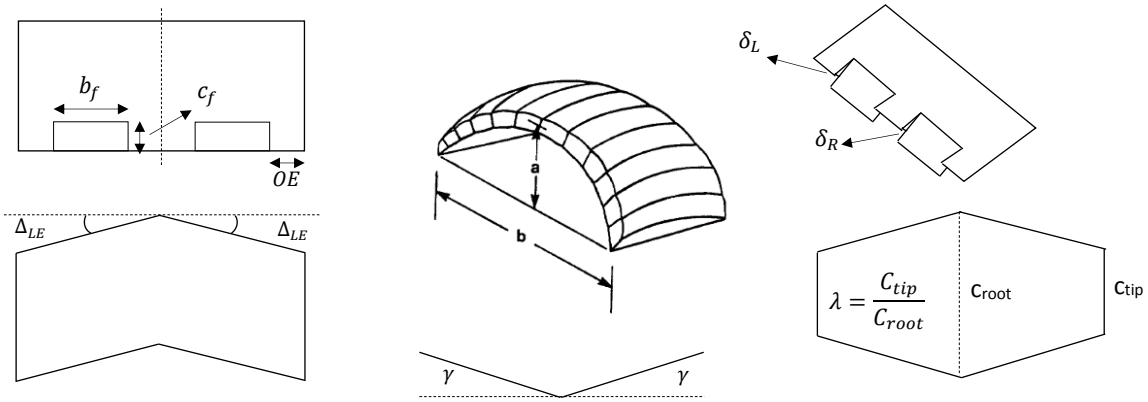


Figure 24: Parametrical representation of ram-air parachute geometry

For the generation of the mean camber surface shape, automatic modelling of NACA 4 and 5 digits airfoils is implemented in the code. Regarding reference axes, two alternatives are considered as shown in Figure 25.a and 25.b

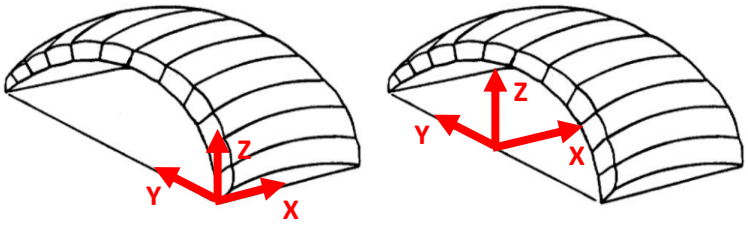


Figure 25: a) Reference axes position option 1. b) Reference axes position option 2

Both options have its advantages and disadvantages. Option 1 avoids working with negative numbers what, in some situations, may be a source of trouble. On the other hand, option 2 gives the possibility of using symmetry to simplify calculations. However, as in this thesis control surfaces are modelled, usually no symmetry will be present in the canopy because of asymmetrical deflections, so option 2 loses its advantages, and option 1 is the choice made.

Same as any panel method, the lifting surface must be panelled. There are different ways of doing it such as uniform, cosine or sine distributions. The proposed design lets the user choose between uniform or cosine distribution in both chordwise and spanwise directions. For properly doing that panelling procedure, some numerical parameters must be defined, which will be also useful for the rest of the code development (see Figure 26 and Table 6)

	Parameter	Definition
PANELS PARAMETERS	N_{xc}	Number of panels of the canopy in x direction
	N_{xw}	Number of panels of the wake in x direction
	N_x	Total number of panels in x direction ($N_x = N_{xc} + N_{xw}$)
	M	Number of panels in y direction (same for the canopy and wake)
	N_{pc}	Total number of panels in the canopy ($N_{pc} = N_{xc} \cdot M$)
	N_{pw}	Total number of panels in the wake when fully developed ($N_{pw} = N_{xw} \cdot M$)
	N_p	Total number of panels ($N_p = N_x \cdot M$)
PANELS PARAMETERS USED FOR CONTROL SURFACES DEFINITION	N_{OE}	Number of the M spanwise panels used to define the exterior offset
	N_{bf}	Number of the M spanwise panels used to define the c.s span
	N_{OI}	Number of the M spanwise panels used to define the interior offset
	N_{cf}	Number of the Nxc chordwise panels used to define the c.s chord
	N_r	The rest of the N_{xc} panels (N_{xc}-N_{cf})
POINTS/NODES PARAMETERS	n_{xc}	Number of points of the canopy in x direction ($N_{xc} + 1$)
	n_{xw}	Number of points of the wake in x direction ($N_{xw} + 1$)
	n_x	Total number of points in x direction ($n_x = N_x + 1$)
	m	Number of points in y direction ($m = M + 1$)
	n_{pc}	Total number of points in the canopy ($n_{pc} = n_{xc} \cdot m$)
	n_p	Total number of points ($n_p = n_x \cdot m$)

Table 6: Definition of the main numerical parameters related to panels and points (or nodes)

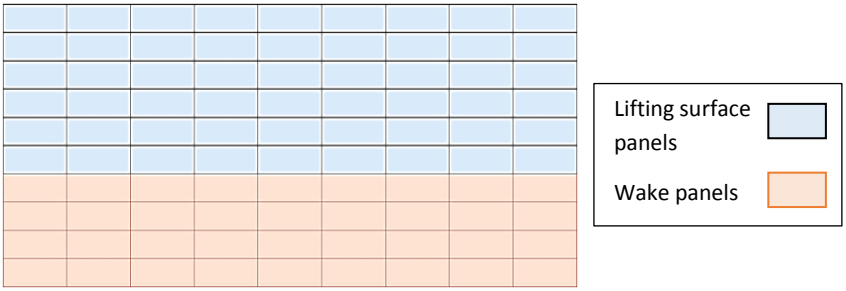


Figure 26: Planform representation of the lifting surface panelling.

Regarding the singularity elements used in UVLM, it has already been mentioned that vortex rings are typically selected. As an extension of two-dimensional methodologies used to satisfy Kutta conditions, these panels are usually placed such that the bound vortex is located at ¼ of the corresponding panel chord and the collocation point is located at the middle of the ¾ panel chord line (similar approaches are followed in lifting-line approaches) . However, for the sake of simplicity in terms of geometry, in this development it has been decided to align vortex rings with the partitioning used to model the geometry, which is the approach followed in doublet lattice methods (note that a vortex ring is equivalent to a constant strength doublet panel). That means that the four corner points/nodes of each vortex will be the same as the four corner nodes of the corresponding wing panel. In Section 2.4 there are some verification tests which have been useful to assess the suitability of this approach.

The numeration of panels and nodes has been done as proposed in [24] (See Figure 27). Through the development of the code, it is used a counter *k* to represent either the point or the panel number we are analysing or working with at that moment. Sometimes two counters may be needed to account for influences between panels. In those cases, the counter *L* is introduced.

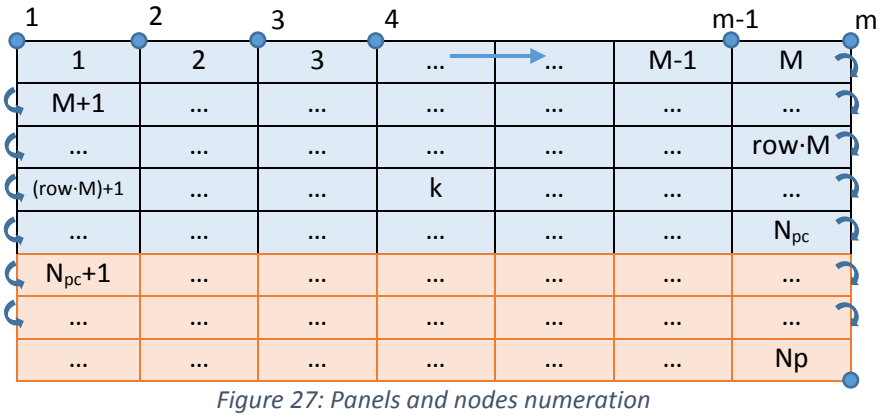


Figure 27: Panels and nodes numeration

This panel/node numeration corresponds to the case in which no control surfaces are modelled. Corresponding modifications for the case including c.s is explained later at the end of this section.

To store the coordinates of all nodes COORD is created. It is a matrix of three rows and a number of columns same as the number of points. First row stores X coordinates,

second row stores Y coordinates, third row stores Z coordinates and each column represents a different point:

$$COORD = \begin{bmatrix} X_1 & X_2 & \dots & X_{np} \\ Y_1 & Y_2 & \dots & Y_{np} \\ Z_1 & Z_2 & \dots & Z_{np} \end{bmatrix}$$

This type of matrix makes difficult to work locally with a panel. This is because if we are making calculations with an arbitrary panel 'k', it is not easy to know which is the numeration of each of the four nodes that define the panel and thus, we don't know their coordinates. To tackle that problem, an analysis of the numeration has been made to find relations between panel 'k' and the number of its four defining nodes. Once known these relations, CONNECT matrix is created. It consist on a $4xN_p$ matrix in which each column represents a panel, and the four rows represent the four corner points (let them be 1 (top right), 2 (top left), 3 (down right), 4 (down left)) (Figure 28). This matrix stablishes the relation between the global node numeration (from 1 to np) and the local node numeration (from 1 to 4). Hence, if we want to know the global number of the top right corner node of the panel "k" we have to access row 2, column "k" of the connectivity matrix. This will give us the global numeration of the sought point. Then, it is easy to obtain its coordinates by accessing the corresponding column of the COORD matrix.

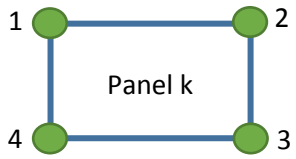


Figure 28: Panel nodes nomenclature

The relations found between the panel number and the general numeration of its four corner nodes, are shown in Figure 29:

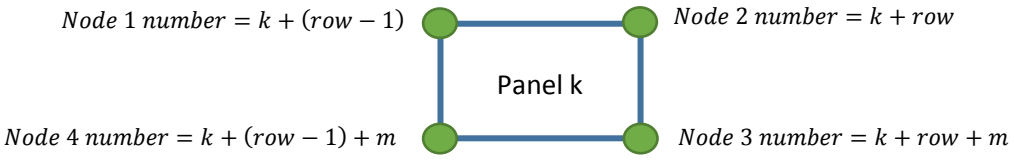


Figure 29: Relations between panel number and its four corner nodes numbers.

With these relations, CONNECT matrix can be filled up with an iterative loop.

$$CONNECT = \begin{bmatrix} A_1 & A_2 & \dots & A_{N_P} \\ B_1 & B_2 & \dots & B_{N_P} \\ C_1 & C_2 & \dots & C_{N_P} \\ D_1 & D_2 & \dots & D_{N_P} \end{bmatrix}$$

Then, COORD matrix is calculated following a specific procedure:

1. X, Y coordinates of the canopy are defined with uniform or cosine distributions introducing the corresponding taper ratio and sweep angle.
2. Z coordinates of the canopy are defined using NACA 4 or 5 digits mean camber line and a polynomial adjustment for the canopy shape. If needed, the dihedral angle is introduced at this step.
3. Wake coordinates are included.

The treatment of wake coordinates is not the same for a steady case than for an unsteady one. For steady case, the wake is modelled with a row of panels of n times the span length (Figure 30) what intends to represent that the canopy wake is fully developed. If n is not big enough (i.e. the wake is not large enough), the steady problem results will depend on it. Hence, a test is made consisting on increasing n until a value is find such that results do not change even though we keep on increasing n (see results in Section 2.4.1.2).

For the unsteady case, a function is defined to update wake coordinates after each time step. This function is called WAKESHEDDING (see related code lines in the Appendix 2.11). Each time step, it adds a new row of panels until the wake has as many rows as defined at the input data of the code. Then, the coordinates of each row are updated, but the total number of wake rows remain constant (See Figure 31). With this type of wake model, the transient aerodynamic behaviour can be studied (See Section 2.4.2).

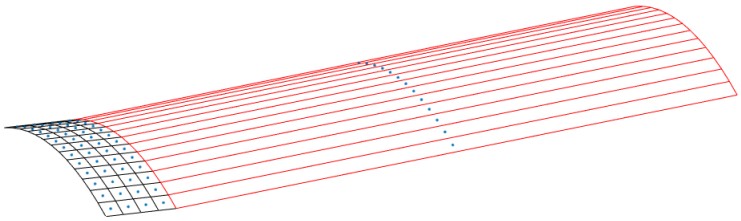


Figure 30: Steady case wake shedding. (One row of long panels without rollup)

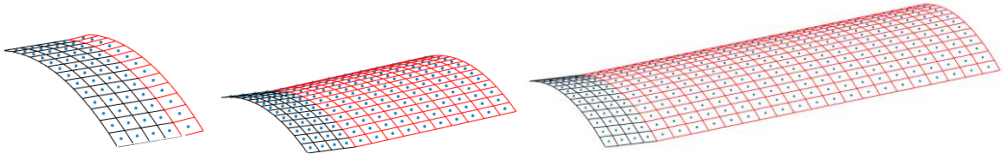


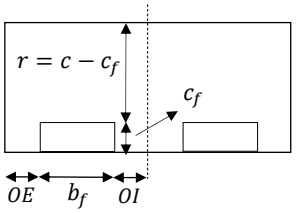
Figure 31: From left to right. Evolution of the unsteady wake without rollup

In addition, the possibility to simulate how the wake is curved (wake rollup) due to the induced velocities has been introduced. This is explained later in Section 2.3.2

Regarding vortex geometry parameters it has been already stated that its four corner nodes coordinates are shared with panel coordinates. It is also necessary to compute the collocation point, the normal vector and surface of each of those vortices. In this work that the vortex control point is located at the centre of the panel/vortex. This is

the same as saying that it is located at the panel diagonal midpoint. Hence, it is easy to obtain its coordinates just by applying the midpoint formula to the diagonal extremes. The normal vector of each panel can be easily obtained by a cross product of the two diagonal vectors oriented in a way such that normal vector comes out from the surface. Finally, the surface of each panel/vortex is computed by means of Heron formula. All this development is explained in detail in the Appendix 6. In addition, the code subroutine VORTEXGEO, which performs all this computations, is given in Appendix 1.11 and Appendix 2.9 for the steady and unsteady case, respectively.

As mentioned before, the potential aerodynamic solver also models control surfaces. To do so, a proper treatment of the numerical parameters must be done. First, the program automatically verifies if the inputs are within the logical limits by means of check_cs_input function (See related code lines in Appendix 1.3). One of these limits is that the number of panels along the span (M) is forced to be even. Hence, it is known that each semispan will have M/2 panels. Another important limit is that uniform distribution has also been enforced. Knowing those two limits, The M/2 panels of each semispan are distributed proportionally between the three main geometric distances: $b_f, OE \; and \; OI = \frac{b}{2} - (OE + b_f)$. The same is done with chordwise panels and the distances c and r (Figure 32).



- $b_f \rightarrow N_{bf}$ panels
- $OE \rightarrow N_{OE}$ panels
- $OI \rightarrow N_{OI}$ panels
- $c_f \rightarrow N_{cf}$ panels
- $r \rightarrow N_r$ panels

Figure 32: Main geometrical distances used for control surface definition and their corresponding numerical parameters.

These numerical parameters are then computed as:

$$N_{bf} = \frac{M \cdot b_f}{b} \quad (49) \qquad N_{OE} = \frac{M \cdot OE}{b} \quad (50) \qquad N_{OI} = \frac{M \cdot OI}{b} \quad (51)$$

$$N_{cf} = \frac{N_{xc} \cdot c_f}{c_{root}} \quad (52) \qquad N_r = \frac{N_{xc} \cdot r}{c_{root}} \quad (53)$$

yielding:

$$M = 2 \cdot (N_{bf} + N_{OE} + N_{OI}) \quad (54)$$

$$N_{xc} = N_{cf} + N_r \quad (55)$$

The previous calculations are done with the function "generate_control_surface" (related code can be seen in Appendix 1.6).

With the c.s numerical parameters stablished, some modifications are included to the way in which coordinates are defined. The number of total panels/vortices remain the same but some extra points have to be included to simulate correctly the deflections of control surfaces. To understand how this is managed numerically, first, the reader has

to note that there are some points which, despite being initially shared between the control surfaces and the wing (when no deflection is present), they are finally moved to another position (marked as black dots in Figures 33 and 34). The first thing that is done is to duplicate those specific points before moving them. That means that, initially, at each black marked position, there will be two nodes with same coordinates but then, one point of each pair is moved to its required position according to the input deflection. This is done by the simulate_control_surface subroutine showed in the Appendix 1.9. Note that the addition of those extra points supposes a change in the connectivity matrix (more information about this implementation is given in the Appendix 1.5(steady) and 2.5 (Unsteady)).

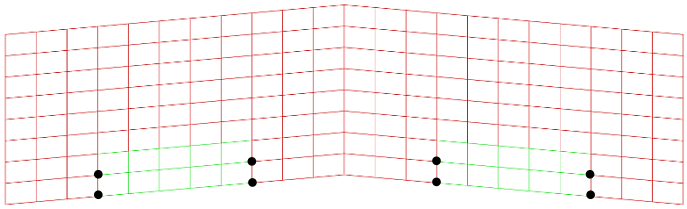


Figure 33: Planform of a wing with control surfaces. Representation of the points which need to be duplicated for proper control surfaces definition

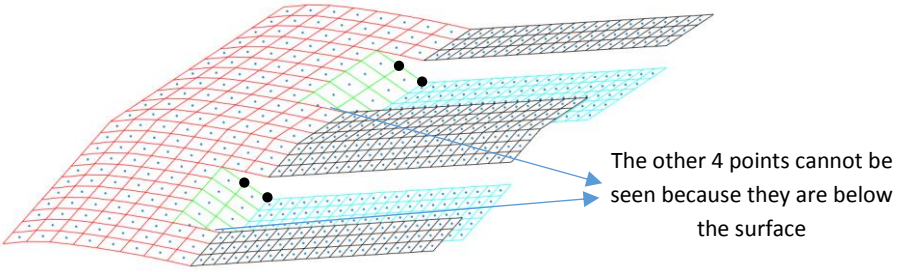


Figure 34: 3D shape of a wing with control surfaces. Representation of the points which need to be duplicated for proper control surfaces definition

2.3.2 Flow solver

When the geometry is already defined and the grid has been generated, the following step is to solve the system of equations derived from the application of the zero normal velocity boundary condition. Doing so, circulation values are obtained and thus, aerodynamic forces can be computed and wake rollup can be implemented. In this thesis, both the steady and unsteady cases of the problem have been developed so during the following explanation both variants will be stated and compared.

Let's recall the system of equations to be solved (Eq. 48):

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1N_p} \\ a_{21} & a_{kL} & \cdots & a_{2N_p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N_p1} & a_{N_p2} & \cdots & a_{N_pN_p} \end{pmatrix} \begin{pmatrix} \Gamma_1 \\ \Gamma_2 \\ \Gamma_3 \\ \vdots \\ \Gamma_{N_p} \end{pmatrix} = \begin{pmatrix} RHS_1 \\ RHS_2 \\ RHS_3 \\ \vdots \\ RHS_{N_p} \end{pmatrix}$$

For the computation of influence coefficients matrix (matrix A) and RHS vector, a subroutine called COEFCALCULATION is defined (see related code lines in Appendix 1.4 for steady and 2.4 for unsteady). As previously mentioned, matrix "A" coefficients represent the normal velocity induced by each vortex ring in each of the collocation points assuming a unit circulation. To compute them, a function that calculates the velocity induced by a vortex ring is created (VORTEXR; see Appendix 1.13). As a vortex ring is a combination of four vortex lines, its induced velocity will also be the combination of the velocity induced by the four vortex segments. Hence, also a function for the computation of the velocity induced by a vortex line is developed (VORTEXL; see Appendix 1.12). To do so, a vortex segment of circulation "Γ" defined by points P_2 and P_1 is considered as in Figure 35.

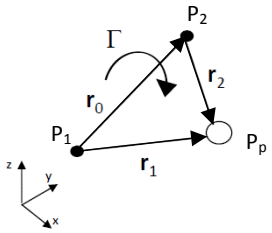


Figure 35: Geometrical parameters for the calculation of the velocity induced by a vortex line in an arbitrary point. Retrieved from [51]

By the Biot-Savart's law the velocity induced at P_p by this straight vortex line is:

$$V_{ind} = \frac{\Gamma}{4 \cdot \pi} \cdot \frac{\vec{r}_0}{\|\vec{r}_1 \times \vec{r}_2\|} \cdot \left(\frac{\vec{r}_1}{\|\vec{r}_1\|} - \frac{\vec{r}_2}{\|\vec{r}_2\|} \right) \cdot \frac{\vec{r}_1 \times \vec{r}_2}{\|\vec{r}_1 \times \vec{r}_2\|} \quad (56)$$

To prevent the result from going to infinity, the condition of $\vec{V}_{ind} = (0 \quad 0 \quad 0)$ is applied if $\|\vec{r}_1 \times \vec{r}_2\|$, $\|\vec{r}_1\|$ or $\|\vec{r}_2\|$ are lower than a small constant value. Here, the possibility to implement the "Vortex core Rankine model" (see for instance [52]) has been considered but it has finally been discarded due to lack of time.

For the computation of the velocity induced by a vortex ring in a specific collocation point, the induced velocity must be computed once per vortex line (4 times) and the contribution of each of them must be added:

$$\vec{V}_{ind} = \vec{V}_{ind_1} + \vec{V}_{ind_2} + \vec{V}_{ind_3} + \vec{V}_{ind_4} \quad (57)$$

Finally, as only the component normal to the surface is needed, the panel normal vector multiplies this induced velocity to obtain the corresponding coefficient.

$$a_{kL} = \vec{V}_{ind} \cdot \vec{n} \quad (58)$$

This is done for each canopy panel in each of the canopy panels collocation points ($N_{pc} \times N_{pc}$ coefficients). However, for the last row of panels of the canopy a special treatment is required. Initially, the circulation of the first wake row of panels (or the only wake row in the steady case) is not known, so it cannot be moved to the RHS of the equation. Nevertheless, to satisfy the Kutta condition in 3D, it must be enforced that both the last row of canopy panels and the first row of wake panels have the same values of

circulation. This makes that the circulation along the parachute trailing edge vanishes, satisfying the Kutta condition (see for instance [24]). Consequently, the velocity induced (assuming unitary circulation) in a collocation point "k" by a panel located at the wake's first row can be treated together with the "neighbour" panel from the canopy (see crossed panels in Figure 36). Then, the equation at a collocation point "k" for the unsteady case yields:

$$a_{k1}\Gamma_1 + a_{k2}\Gamma_2 + a_{k3}\Gamma_3 + \cdots + (a_{kN_p} + V_{indw(N_p+M)_k})\Gamma_{N_p} = -\left[\vec{U}_\infty + \vec{U}_{wake_{rows>1}}\right]_k \cdot \vec{n}_k \quad (59)$$

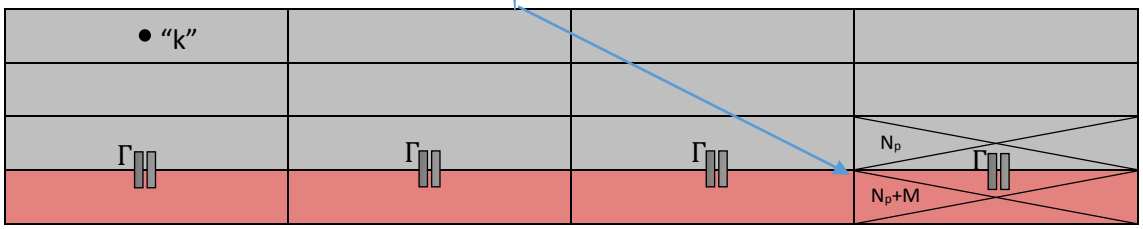


Figure 36: Graphical representation of the canopy's last row coefficients calculation

Note that on the RHS we have the freestream contributions and also the induced velocity contributions of the previously shed wake panels. That means that, conversely to the first wake row, their circulation is known (it must not change by virtue of circulation conservation statements). Therefore, after each time step, the circulation values are transferred from one row to the next one such that only the circulation of the first wake remains unknown (see Figure 37).

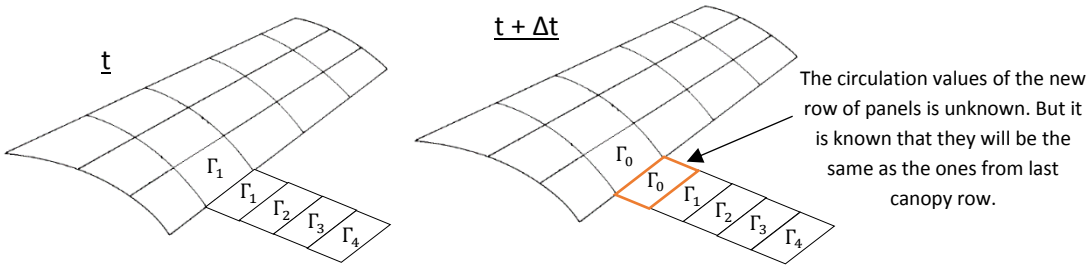


Figure 37: Evolution of wake circulation values after a timestep.

For the steady case, the components of the RHS vector are $RHS_k = -\vec{U}_{\infty_k} \cdot \vec{n}_k \quad (60)$ instead of the ones presented in Eq. (59) for the unsteady case.

At the unsteady case, special attention must be payed to the calculation of the induced velocity at the collocation point "k" by the panels of the first wake row ($V_{indw(N_p+M)_k}$ term of equation 59). In this case, instead of adding the four vortex line contributions to the induced velocity, only the L.E bound vortex and the two trailing vortices are accounted for. This was done in this way because numerical experiments demonstrated that otherwise the resulting induced drag was incorrect. Even though this requires further investigation, a hypothesis can be attempted by analogy with the steady case. In this case, the first row of panels of the wake is very long and thus, the T.E vortex is far from the canopy, what makes its contribution to the induced velocity negligible. Then, for the unsteady case to be the same, this contribution is neglected. In addition, it has

to be mentioned that for the computation of the induced velocity due to the other wake panels ($\vec{U}_{wake_{rows>1}}$ term of Eq. 59), only the free trailing vortices are considered (other contributions cancels out).

When both "A" matrix and RHS vector are computed, the circulation vector is easily obtained and further computations can be made. The first of those computations is the aerodynamic forces calculation, which is carried out with a function called AEROFORCES (see Appendix 1.2 and 2.2 for the steady and unsteady cases, respectively). This function basically follows the Kutta-Joukowsky method (explained in [53]) which calculates the aerodynamic force generated by each canopy vortex and then adds all the contributions to calculate the resultant aerodynamic force. It supposes each vortex force contribution acting at the middle point of the bound vortex and calculates it according to the Kutta-Joukowsky law, namely

$$\vec{f}_{steady} = \rho \cdot \vec{U} \times \vec{\Gamma} \quad (61)$$

where $\vec{U} = \vec{U}_\infty + \vec{U}_{tr} \rightarrow \vec{U}_\infty$ is the aerodynamic speed and $\vec{U}_{tr}$ is the velocity induced at the midpoint of the vortex line (where the force resultant force is located) by all the trailing vortex lines from both the canopy and the wake. This allows obtaining the induced drag contributions in a direct manner. The circulation vector is computed as the value of circulation multiplied by the vector defining the corresponding vortex line.

Notice that the suffix "steady" has been written in the expression because this is the vortex force contribution for the steady case. For the unsteady case, another contribution must be added to obtain the total vortex force due to the transient change in circulation values. This other contribution is expressed as [53]

$$\vec{f}_{unsteady} = \rho \frac{d\Gamma}{dt} c\left(\hat{U} \times \delta\vec{l}\right) \rightarrow \quad \vec{f}_{unsteady} = \rho \frac{\Gamma(t) - \Gamma(t - \Delta t)}{\Delta t} c\left(\hat{U} \times \delta\vec{l}\right) \quad (62)$$

Hence, the resultant aerodynamic force is:

$$\vec{F} = \sum_1^{N_{pc}} \vec{f} \quad (63)$$

where $\vec{f} = \vec{f}_{steady}$ for the steady case and $\vec{f} = \vec{f}_{steady} + \vec{f}_{unsteady}$ for the unsteady case.

An alternative to the Kutta-Joukowsy method presented (K.J.) is the formulation proposed by Katz in [24]. The advantage of the K.J. formulation is that it gives directly the three force components in body coordinates (F_X, F_Y, F_Z) while Katz formulation doesn't. This is one of the reasons why K.J. has been chosen. However, also Katz formulation was implemented and tested but it showed some problems and led to errors in the results (probably due to the lack of details given by references in the method description, which could lead to misunderstanding of some aspects such as the way of computing induced velocities).

Beside the calculations of the aerodynamic forces, also the aerodynamic moments are computed. These are calculated in a reference point which is defined by the user as an

input. This reference point (X_{ref}, Y_{ref}, Z_{ref}) can be, for example, the leading edge (L.E.) of the root chord. To compute each vortex moment contribution, the moment definition is applied.

$$\vec{m} = \vec{r} \times \vec{f} \quad (64)$$

where $\vec{r}$ is the distance vector from the reference point to the application point of the vortex force contribution (midpoint of the bound vortex). Then, the total aerodynamic moment is obtained by the sum of all the contributions:

$$\vec{M} = \sum_1^{N_{pc}} \vec{m} \quad (65)$$

These forces and moments are both expressed with dimensions (forces in N, and moments in N·m). As explained in Section 1.2, in aerodynamics it is common to work with non-dimensional variables so both force and moment vectors are made non-dimensional by means of:

$$\overrightarrow{C_F} = \frac{\vec{F}}{\frac{1}{2}\rho\|\vec{U}_\infty\|^2 S_{ref}} \quad (66) \qquad \overrightarrow{C_M} = \frac{\vec{M}}{\frac{1}{2}\rho\|\vec{U}_\infty\|^2 c_{ref} S_{ref}} \quad (67)$$

Typically, aerodynamic forces and moments are expressed in wind coordinates but all the previous expressions give forces and moments in body coordinates. Consequently, force and moment vectors are rotated to wind coordinates by means of a rotation matrix:

$$b2w(body\ to\ wind) = \begin{pmatrix} \cos(\alpha)\cos(\beta) & -\cos(\alpha)\sin(\beta) & \sin(\alpha) \\ \sin(\beta) & \cos(\beta) & 0 \\ -\sin(\alpha)\cos(\beta) & \sin(\alpha)\sin(\beta) & \cos(\alpha) \end{pmatrix} \quad (68)$$

For the unsteady case, another further computation which is done is the simulation of wake rollup. This is the calculation of how the wake is deformed due to the induced velocity field, what is very illustrative because it shows behaviours such as the starting vortex or the curl of the wingtip vortices. This is done by means of a subroutine called WAKEROLLUP (see implementation in the Appendix 2.10).

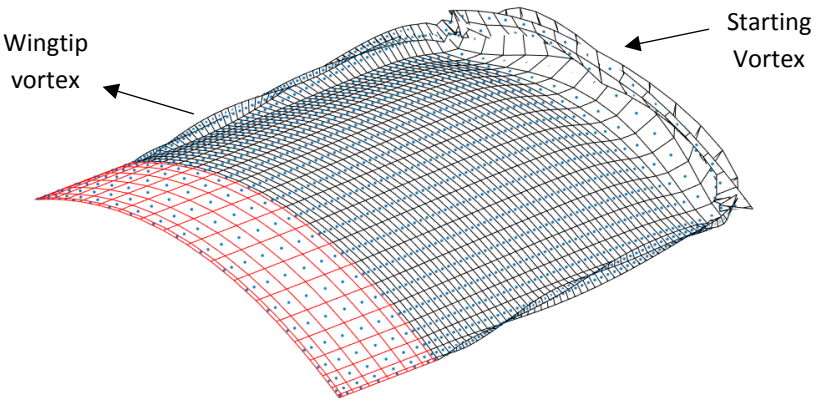


Figure 38: Representation of the wake rollup for a canopy with NACA 0010

2.3.3 Viscous corrections

As it has been explained before, potential flow solvers make the assumption of inviscid flow. Hence, the formulation described above does not account for viscous contributions. The aim of the developed flow solver is to be coupled in a parachute flight simulator so, bearing in mind that viscous drag is an important contribution to flight mechanics, it would be necessary to account for viscous contribution to some extent. The current section explains how viscous corrections are introduced in the solver.

The canopy's total drag can be expressed as

$$C_D = C_{D_p} + C_{D_i} \quad (69)$$

which is similar to Eq. (25) presented in Section 1.2.2, but C_{D_i} in this case corresponds to the induced drag computed by the solver. The contribution C_{D_p} can be obtained from the canopy profile drag C_{d_p} along the span [48], i.e. (see Section 1.2.2)

$$C_{D_p} = \frac{1}{S_{ref}} \int_0^b C_{d_p}\big(C_l(y)\big) c(y) dy \quad (70)$$

which can be obtained in terms of the local lift by means of experimental data fitting, for example $C_{d_p}\big(C_l(y)\big) = A_s C_l(y)^2 + B_s C_l(y) + C_s$ where A_s, B_s and C_s are constants coefficients (input data). The integral in Eq. (70) is done numerically by computing the local lift $C_l(y)$ and $S(y)$ in each chordwise strip (see Figure 39).

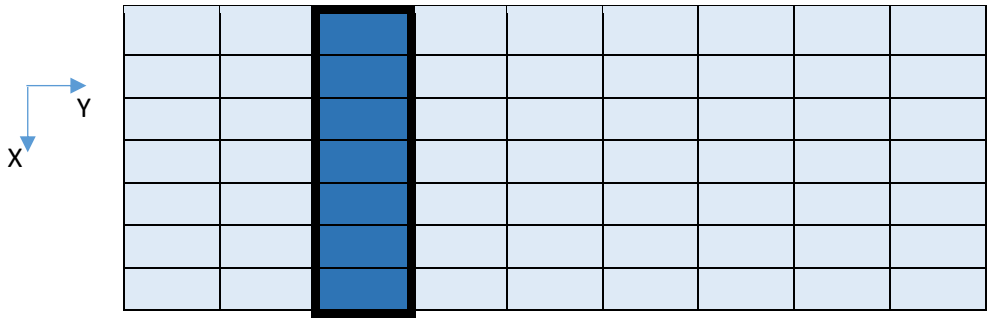


Figure 39: Graphical representation of a chordwise strip. Numerical integration of Cl along the span.

The C_l of a strip is computed as the sum of each of its panels C_l and the surface of the strip is computed as the sum of each of its panels surface. Then, the integral is solved doing a loop from the first to the last strip ("j" represents the strip/column counter):

$$C_{D_p} = \frac{1}{S_{ref}} \sum_{j=1}^{M} C_{d_p}\big(C_l(j)\big) \cdot S(j) = \frac{1}{S_{ref}} \sum_{j=1}^{M} (A \cdot C_l(j)^2 + B \cdot C_l(j) + C) \cdot S(j) \quad (71)$$

This contribution is added to the induced drag in order to obtain the total canopy drag.

2.4 Validation/verification assessment

2.4.1 Steady tests cases

2.4.1.1 C_L convergence analysis

In this type of numerical implementations, the problem discretization play a very important role in the numerical results. As mentioned before, the denser is the grid, the better approach to the real value is obtained, but the computational cost becomes higher. Hence, before performing the validation/verification tests, a convergence analysis is conducted to determine the optimum grid size. That means fast results calculation with the lowest error possible. The analysis has been made with a wing having AR = 4 and angle of attack $\alpha = 5°$. The grids studied has uniform panels distributions with M=4*Nxc, being M and Nxc the number of spanwise and chordwise panels, respectively.

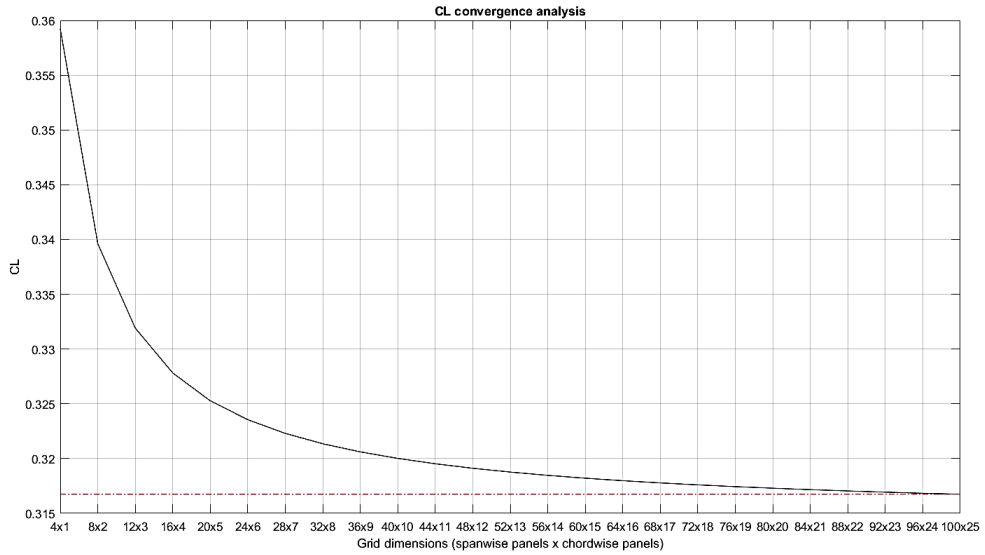


Figure 40: CL convergence analysis as a function of the grid

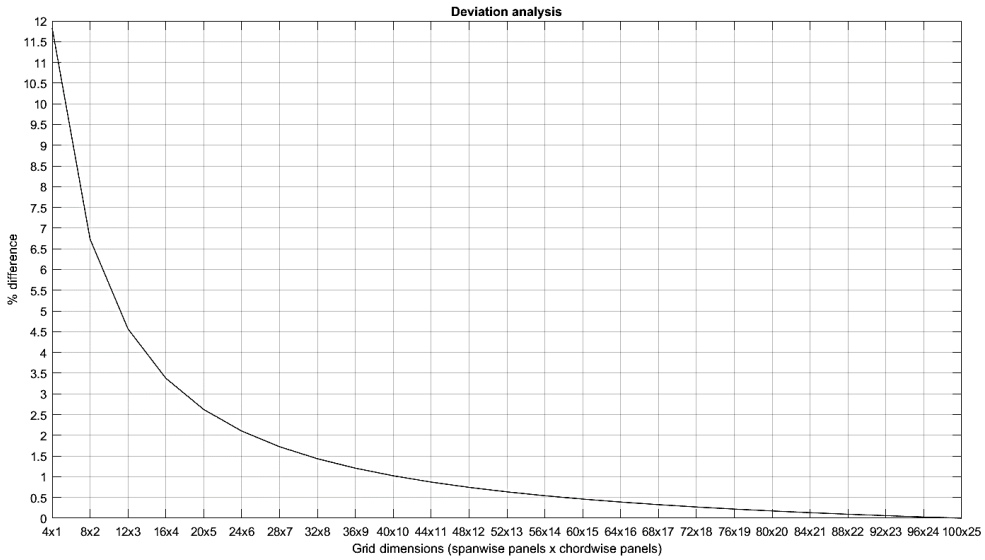


Figure 41: Evolution with grid size of the difference (%) between the calculated and final converged C_L

Analysing the results of Figures 40 and 41 it has been decided that a grid of 24x6 panels is suitable for verification/validation purposes. The difference observed with a fully converged result (finer grid) is about 2.5% and has an acceptable CPU time.

2.4.1.2 C_L convergence study as a function of wake length

Recall that for steady case, wake is modelled with a row of panels of n times the span length. If the wake length (i.e n) is not big enough, the result depends on it. The aim of this section is to find a proper value of n such that results do not change significantly, even though we keep on increasing n. The wing and selected discretization (24x6) are the same used in the previous section. The results presented in Figure 42 show that the lift does not change considerably for wake lengths greater than 10 spans, and differences are difficult to identify for a 20-spans wake length. According to these results, a length of 10 spans has been selected.

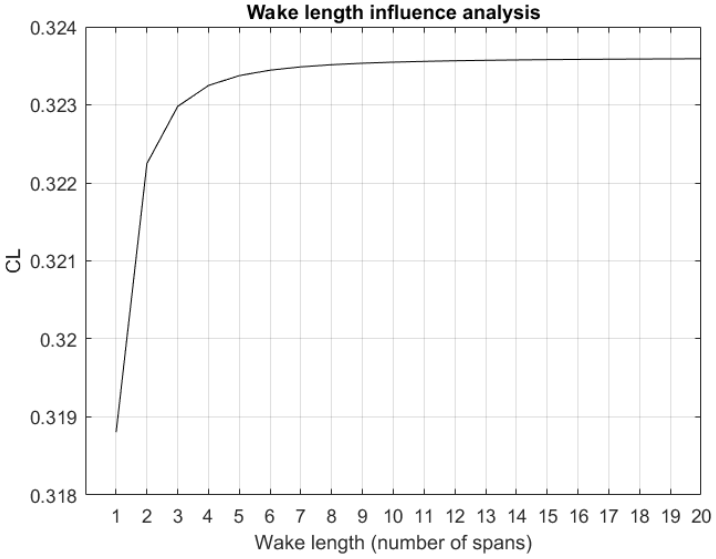


Figure 42: Wake length influence on the steady lift

2.4.1.3 Results comparison with XFLR5

The XFLR5 program is a public domain analysis tool for the solution of airfoils, wings and planes operating at low and high Reynolds Numbers [54]. It includes:

- XFoil (coupled panel-boundary layer method for airfoil analysis)
- Wing design and analysis capabilities based on: Lifiting Line Theory, Vortex Lattice Method, and on a 3D Panel Method

Since XFLR5 has been extensively validated, it will be used in this work to compare the results obtained with our aerodynamic solver. It is worth mentioning that XFLR5 uses vortex rings with the bound vortex located at ¼ of the panel's chord. Consequently, the comparison of the results will be also useful to investigate if placing vortex elements aligned with the geometric wing panels has some effect on the calculated results. The following table contains the parameters that define the simulation done in both codes.

Geometrical data	Numerical data	Aerodynamic data
• Span: 10 m • Chord: 3 m • AR: 10/3 • Airfoil: symmetric (0010) • Untappered • $\Lambda = 0°$ and $\gamma = 0°$ • Wake length = 10 spans	XFLR5 default numerical parameters: • 13 chordwise panels • 38 spanwise panels • Cosine distribution	• $V_\infty = 20\ m/s$ • Tests with: - $\alpha = 2°$ - $\alpha = 5°$ - $\alpha = 8°$

Table 7: Geometrical, numerical and aerodynamic data for the simulation carried out to compare results with XFLR5.

With the data from Table 7, the wing created by XFLR5 is shown in Figure 43. The same geometry, generated with the developed MATLAB code is presented in Figure 44. It can be observed that these are almost identical.

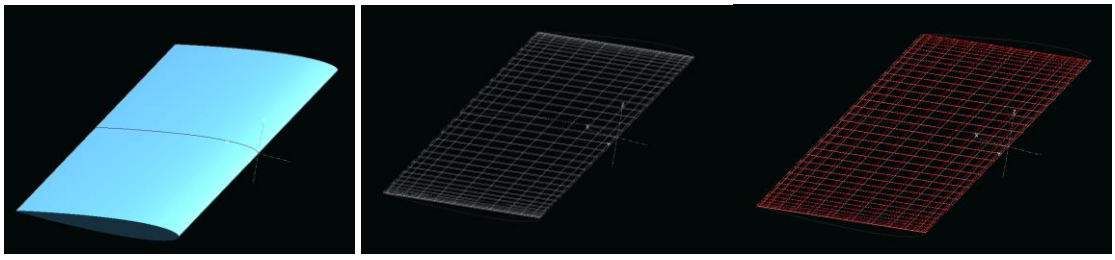


Figure 43: From left to right. Graphical representation of the simulated wing, panelled mean camber surface and vortex rings placed on the mean camber surface. From XFLR5

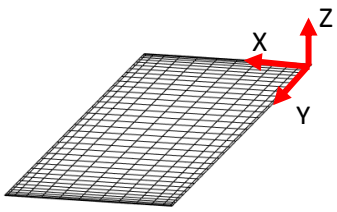


Figure 44: Graphical representation of the panelled mean camber surface (from the developed code)

The simulation is performed both in XFLR5 and the MATLAB solver and the results obtained are compared in Table 8 below. As it can be seen, a satisfactory agreement is observed between both codes (differences are less than 1.5%).

Panels	**Alpha(°)**	**C_L MATLAB**	**C_L XFLR5**	**C_{Di} MATLAB**	**C_{Di} XFLR5**	**% dif. C_L**	**% dif. C_{Di}**
13 in X 38 in Y	2	0.118125	0.117712	0.001301	0.001298	0.35	0.23
	5	0.294747	0.292780	0.008110	0.008069	0.67	0.51
	8	0.469923	0.464027	0.020648	0.020451	1.25	0.95

Table 8: Results comparison with XFLR5 code (a six decimal representation is used for C_L and C_{Di} to highlight the differences).

2.4.1.4 Further testing with taper ratio and sweep angle

The evolution of the wing's lift slope with AR ($C_{L_\alpha} = f(\mathrm{AR})$) is analyzed firstly as a function of the sweep angle (Λ). The wing planform is rectangular. The calculated results are compared with reference validation data from [24] in Figure 45. A satisfactory agreement can be observed.

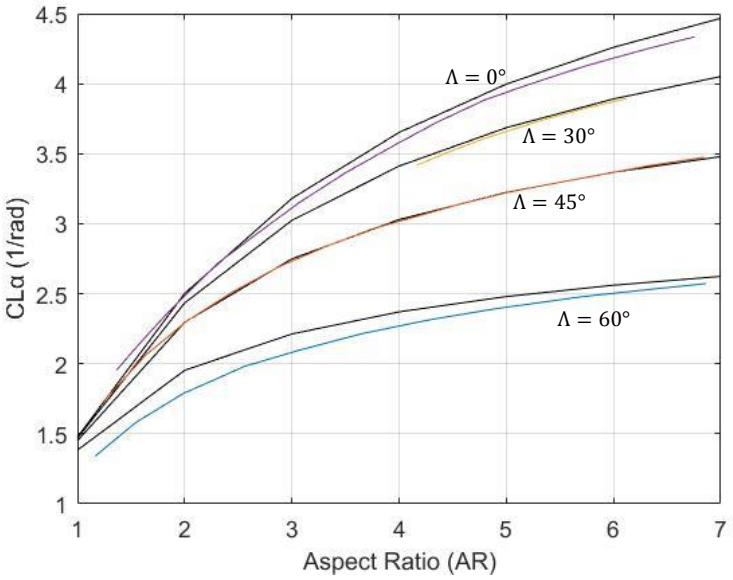


Figure 45: Comparison of the calculated $C_{L,\alpha}$ (black lines) with results from [24] *(coloured lines).*

From a physical point of view, the conclusions obtained from this graph are that, an increase in sweep angle, causes a reduction in the lift slope due to an increase of the downwash near the root and, for a given sweep angle, a lower AR means lower lift slope due to an increase in the induced downwash caused by the trailing vortices effects.

The next analysis involves the variation of the wing lift slope as a function of the sweep angle ($C_{L_\alpha} = f(\Lambda)$) for different values of aspect ratio (AR). In this case, the comparison is made for AR = 4 because there is no more available data. The agreement is also satisfactory (see Figure 46). Patterns observed in the graph are the same as those commented in previous section (the higher the AR the higher the lift slope coefficient, the higher the sweep angle, the lower the lift slope coefficient).

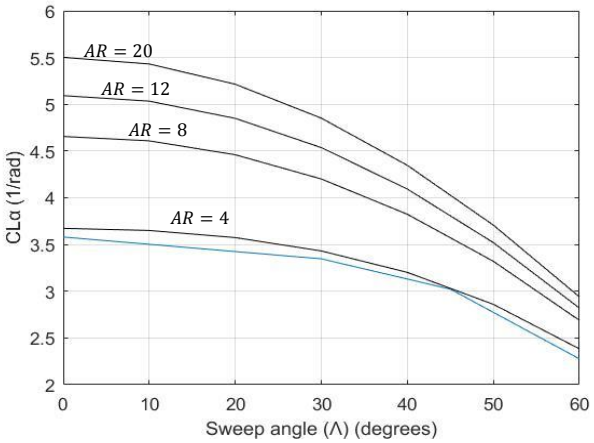


Figure 46: Comparison of the calculated $C_{L,\alpha}$ (black lines) with results from [24] *(coloured lines).*

Next, the lift distribution along the span is calculated for wings having different taper ratio. The results obtained are also compared with those presented in [24].

Analysis and comparison with reference data of the effect of taper ratio on spanwise loading.

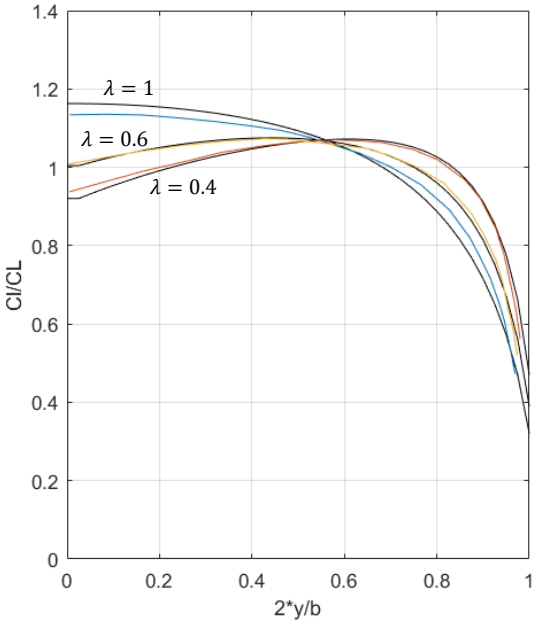


Figure 47: Effect of taper ratio on spanwise loading. Comparison between results obtained with the developed code(in black) and the ones retrieved from [24] *(in colour)*

As it can be observed in Figure 47, the spanwise load reduces towards the tip, and the values achieved depends on the taper ratio. The lower the taper ratio, the higher the value achieved. This supposes that the introduction of taper in a wing makes it worse regarding stall behaviour. Now leaving physical considerations apart, in terms of numerical results validation, it happens the same as in previous sections. Although results do not match perfectly, they show the same patterns and are similar enough to accept them as valid.

2.4.2 Unsteady tests cases

The tests carried out in previous sections are useful for the validation of steady results calculation. In addition, they are a way of verifying that special geometries (taper, sweep, etc.) are properly generated. Nevertheless, considering that, in this thesis, the aim is to develop a ram-air parachute flight simulator, it is also a need to investigate the transient (unsteady) behaviour of calculated variables is correctly simulated by the aerodynamic code. To do so, a specific case extracted from [24] has been tested. It consists of a sudden acceleration of a planar rectangular wing into a constant speed forward flight. The grid selected for the analysis has a uniform distribution of 13x4 panels. Regarding the time step used, it has been decided to adapt the one from the aforementioned reference. This used a dimensionless time increment

$$\Delta t^* = \frac{U_\infty \cdot \Delta t}{c} = \frac{1}{16} \quad (72)$$

Moreover, it has been selected an angle of attack of 5˚ for the analysis and different values of AR have been tested. This method is known as the impulsive wing case.

2.4.2.1 Impulsive wing. C_L variation with time.

First, transient C_L has been studied. The expected behaviour in this case is that the lift coefficient starts with a very high value due to the large value of $\frac{\partial \Gamma}{\partial t}$ during the first timestep. This represents the acceleration from 0 to U_∞. Then, immediately after the wing has reached its steady state speed the lift drops because of the influence of the starting vortex and most of the lift is a result of the $\frac{\partial \Gamma}{\partial t}$ term (because of the change in the downwash of the starting vortex) [24].

The results obtained by the developed aerodynamic solver are presented in Figure 48 and the equivalent results from reference [24] are presented in Figure 49. It is important to note that this comparison can be confusing because the way in which results are obtained is different for both cases. While the developed solver uses K.J. formulation, the results from Figure 49 are obtained using Katz formulation. Hence, although both models are able to capture the initial transient behaviour, there is an evident difference in the convergence speed. The developed solver converges to the solution faster and has a lower initial lift drop than the model from Katz. Regarding the convergence value, both reach a similar value but it is slightly higher (for all AR) in case of the solver from this thesis. This behaviour should be further studied. Differences in the unsteady models discussed have been also studied in the literature, see for instance [53, 55].

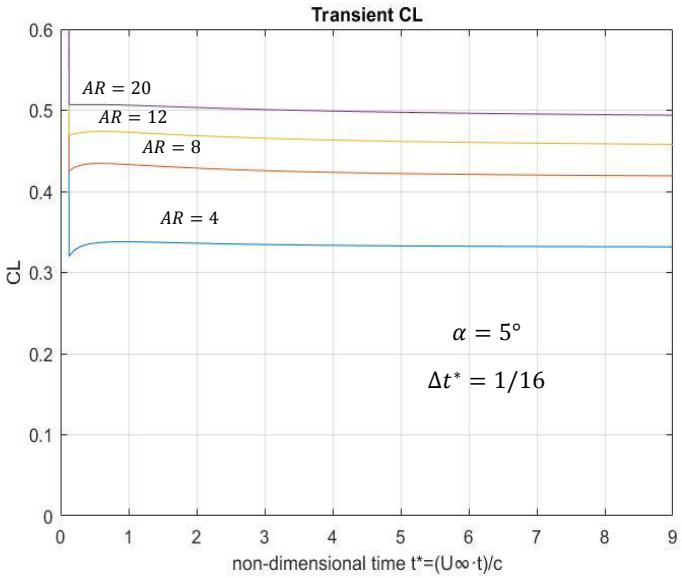


Figure 48: Transient C_L variation for a rectangular uncambered wing suddenly set into a constant speed forward flight.

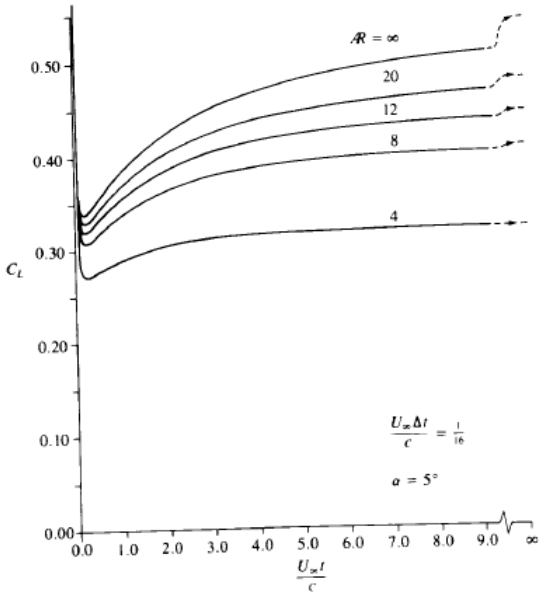


Figure 49: Transient C_L variation with time for uncambered, rectangular wings that were suddenly set into a constant speed forward flight. Retrieved from [24]

2.4.2.2 Impulsive wing. C_{Di} variation with time.

The evolution of the induced drag is investigated for the same wing studied above and the results are presented in Figure 50. It should be noticed that in spite of the fact that the stationary values obtained are slightly higher than those given in [24], these are verify to be coincident with steady results computed with the code XFLR5. As the developed aerodynamic solver calculates the induced resistance following Kutta–Joukowski formulation and reference [24] does it with a different approach, some differences are expected. This is analysed in [53] and [55], where both formulations are compared for different types of movements. The results shown that that for a small number of panels (as in the present case), the two formulations usually show different

results. Despite further investigation is required, probably a higher number of panels may improve matching between the results (convergence analysis was carried out in this work for steady test cases).

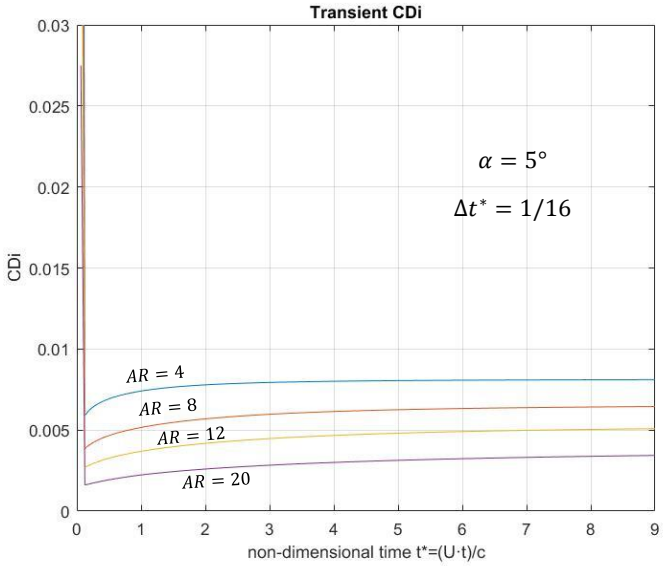


Figure 50: Transient C_{Di} variation with time for uncambered, rectangular wings that were suddenly set into a constant speed forward flight. Retrieved from [24]

Chapter 3

Trajectory simulation

This section deals with the main aspects of the development of the ram-air parachute flight simulator. Firstly, the dynamic model used to simulate the trajectory is described. Then, a description of the way in which it has been programmed is presented, and finally, verification tests are detailed. It should be noted that it has been decided to give a specific name to the simulator: *GPSim* (Gliding Parachute Simulator). This name will be used during all the section for referring to it.

3.1 Dynamic model

The dynamic model selected has six DoF and is based on that described in [20]. As mentioned before it is intended to the simulation of ram-air parachute-payload systems flying under nominal conditions and with a fully inflated rigid canopy. Therefore, the parachute-payload system is treated as a rigid body, and the movements (three displacements and three rotations) are applied to its center of mass. Next, a description of the proposed model is given, starting by axes and attitude angles definition and ending with parafoil kinematics and dynamics explanation.

3.1.1 Coordinate's frames and attitude angles definition

For the proper treatment of the problem, four different sets of axes have been defined:

- *Inertial frame*: These axes are fixed in space and are those used to determine the system position at each moment.
- *Body frame*: Its origin is placed at the centre of mass of the system (C.G.) and it is fixed and moves with it. With three successive finite rotations in a specific order, body axes are oriented with respect to inertial axes. These three rotations are the three orientation angles called Euler Angles and define the attitude of the parachute. First rotation is around Z_b axis and corresponds to yaw angle (Ψ), second rotation is done around the Y_b axis and corresponds to pitch angle (θ) and the last is around X_b and corresponds to roll angle (ϕ). See Figure 16 and Figure 51.
- *Canopy frame:* Its origin is positioned at the canopy's root chord leading edge (L.E.) and it is rotated an angle Γ (rigging angle) around Y-axis (Figure 53) relative to the body axes. It moves with the canopy.
- *Matlab frame:* It is only useful for the purposes of integration between the simulator and the aerodynamic code. This system corresponds to that used for the aerodynamic solver of previous sections (which has been developed with MATLAB). Its origin is located at the L.E. of the left wing tip; its Y-axis is coincident with Y_b and

Y_c but its X and Z axes are in the opposite direction of X_b, X_c and Z_b, Z_c respectively. See Figures 25 and 52.

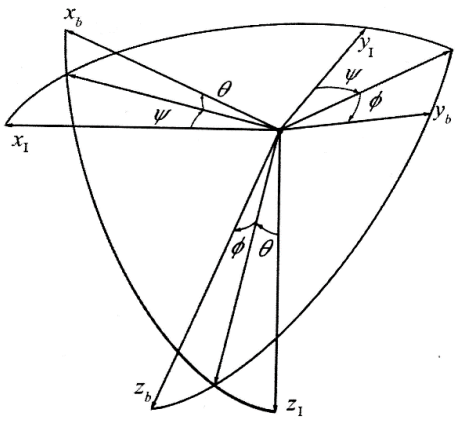


Figure 51: Body frame orientation relative to inertial frame. Euler Angles definition. Adapted from [56]

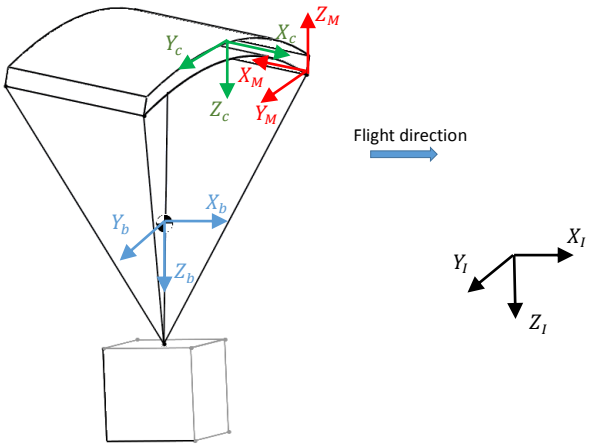


Figure 52: Drawing of a ram-air parachute system. Done with SOLIDWORKS student edition.

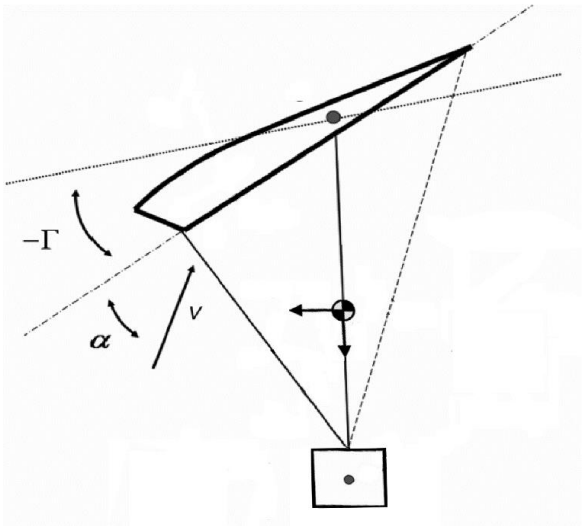


Figure 53: Lateral view of the parafoil system. Definition of the rigging angle. Adapted from [20]

In addition to the axes definition, also transformation matrices must be defined to rotate vectors from one frame to another. This is crucial for the correct performance of the flight simulator because during the code execution, it is done many times. The rotation matrices needed are:

- *i2b* → Inertial frame to body frame

$$i2b = \begin{pmatrix} \cos(\theta)\cos(\Psi) & \cos(\theta)\sin(\Psi) & -\sin(\theta) \\ \sin(\phi)\sin(\theta)\cos(\Psi) - \cos(\phi)\sin(\Psi) & \sin(\phi)\sin(\theta)\sin(\Psi) + \cos(\phi)\cos(\Psi) & \sin(\phi)\cos(\theta) \\ \cos(\phi)\sin(\theta)\cos(\Psi) + \sin(\phi)\sin(\Psi) & \cos(\phi)\sin(\theta)\sin(\Psi) - \sin(\phi)\cos(\Psi) & \cos(\phi)\cos(\theta) \end{pmatrix} \quad (73)$$

- *b2c* → Body frame to canopy frame

$$b2c = \begin{pmatrix} \cos(\Gamma) & 0 & -\sin(\Gamma) \\ 0 & 1 & 0 \\ \sin(\Gamma) & 0 & \cos(\Gamma) \end{pmatrix} \quad (74)$$

- *m2c* → Matlab frame to canopy frame

$$m2c = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \quad (75)$$

Note that rotation matrices are orthogonal and thus, their inverse is the same as their transpose.

3.1.2 Parafoil kinematics and dynamics

For the determination of the ram-air parachute trajectory, linear and angular velocities are needed. These velocities have to be expressed in the inertial reference frame but GPSim solves the dynamic equations using the body reference frame. Hence, the variables obtained in body axes are then converted to the inertial frame. This is done with the following expressions [20]

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = i2b^T \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (76)$$

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\Psi} \end{pmatrix} = \begin{pmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)/\cos(\theta) & \cos(\phi)\cos(\theta) \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix} \quad (77)$$

In order to calculate the linear and angular velocities, the system equations of motion must be solved at every time step. Those equations are obtained applying 2nd Newton's law for both forces and moments. Adding the force contributions about the C.G., these equations result [20]

$$\begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix} = \frac{1}{m}\left(\vec{F}_W + \vec{F}_A + \vec{F}_{PL} - \vec{F}_{AM}\right) - S_\omega^b \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (78)$$

$$\begin{pmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{pmatrix} = (I_T)^{-1}\left(\vec{M}_A + \vec{M}_{AM} + S_{CG,LE}^b \cdot \vec{F}_A + S_{CG,PL}^b \cdot \vec{F}_{PL} + S_{CG,X_{am}}^b \cdot \vec{F}_{AM} - S_\omega^b \cdot \left(\bar{\bar{I}}_T\right)\begin{pmatrix} p \\ q \\ r \end{pmatrix}\right) \quad (79)$$

Since body axes are used for solving Eqs. (78) and (79), all the terms must be expressed in this frame. $\vec{F}_W$, or weight force, is easily obtained in body coordinates with the following expression

$$\vec{F}_W^b = i2b \begin{pmatrix} 0 \\ 0 \\ g(h) \end{pmatrix} \quad (80)$$

where the gravity acceleration g is calculated as a function of the altitude. The aerodynamic forces and moments ($\vec{F}_A, \vec{M}_A$) are computed by the aerodynamic solver subroutine and the payload drag is obtained from Eq. (81). The moments generated by these forces are obtained by $\vec{M}_{PL} = \overrightarrow{CGPL} \times \vec{F}_{PL}$.

$$\vec{F}_{PL} = -\frac{1}{2}\rho V_{PL} S_{ref_{PL}} C_{D_{PL}} \begin{pmatrix} V_{PL_x} \\ V_{PL_y} \\ V_{PL_z} \end{pmatrix} \quad (81)$$

Note that, in Eq. (78) and (79) the terms S_x^y are matrices used to calculate cross products. Subscript "x" indicates the vector that is going to be multiplied and superscript "y" indicates the reference frame in which it is expressed. For example, to multiply $\vec{A} \times \vec{B}$ where $\vec{A}$ and $\vec{B}$ are expressed in body coordinates:

$$\vec{A} \times \vec{B} = S_A^b \cdot \vec{B} = \begin{pmatrix} 0 & -A_z & A_y \\ A_z & 0 & -A_x \\ -A_y & A_x & 0 \end{pmatrix} \begin{pmatrix} B_x \\ B_y \\ B_z \end{pmatrix} \quad (82)$$

This method highly reduces the computational cost of the cross product operation in comparison with built-in functions of MATLAB.

To finish the explanation of the forces presented in the problem, it is necessary to talk about apparent mass effect ($\vec{F}_{AM}$ and $\vec{M}_{AM}$ in Eqs. 78 and 79). It is the effect of the air mass that surrounds the parachute and moves with it. In other words, when a parachute moves, by the non-slip condition, it drags with it a certain volume of air. This volume has a certain mass and has a similar effect as if the parachute were heavier than it actually is. This is the so-called added or apparent mass. In heavy structures, this effect is very small and negligible, but in parachutes, which are very light, this air around it and its inertia, affect their behaviour. To simulate this effect, apparent mass and apparent inertia terms are added to the equations of motion of the system. Those terms are derived from the total kinetic energy of the fluid that moves with the parachute which, according to [57] can be expressed as

$$2T = A\tilde{u}^2 + B\tilde{v}^2 + C\tilde{w}^2 + P\tilde{p}^2 + Q\tilde{q}^2 + R\tilde{r}^2 + 2H(\tilde{v}\tilde{r} + \tilde{w}\tilde{q}) \quad (83)$$

where A, B, C, P, Q, R and H are constants which depend on geometric parameters. GPSim automatically computes all of them, except H, which has to be included as an input. The expressions used for their calculation are given below [58]

$$A = 0.666 \cdot \left(1 + \frac{8}{3}\left(\frac{a}{b}\right)^2\right) \cdot t^2 b \quad (84)$$

$$B = 0.267\left[t^2 + 2a^2\left(1 - \left(\frac{t}{c}\right)^2\right)\right] \cdot c \quad (85)$$

$$C = 0.785 \cdot \sqrt{1 + 2\left(\frac{a}{b}\right)^2\left(1 - \left(\frac{t}{c}\right)^2\right)} \cdot \frac{AR}{1 + AR} \cdot b \cdot c^2 \quad (86)$$

$$P = 0.055 \cdot \frac{AR}{1 + AR} \cdot b \cdot S_{ref}^2 \quad (87)$$

$$Q = 0.0308 \cdot \frac{AR}{1 + AR} \cdot \left[1 + \frac{\pi}{6}(1 + AR) \cdot AR\left(\frac{a}{b}\right)^2\left(\frac{t}{c}\right)^2 \cdot c^3 \cdot S_{ref}\right] \quad (88)$$

$$R = 0.0555 \cdot \left(1 + 8 \cdot \left(\frac{a}{b}\right)^2\right) b^3 \cdot t^2 \quad (89)$$

Note that, unlike in the case of aerodynamic solver, the thickness of the parachute cannot be neglected despite being low. Here the parachute thickness is important in order to calculate the apparent mass coefficients. With all these constants defined apparent mass force and moment can be defined. According to [58] we have

$$\vec{F}_{AM} = -(b2c)^T\left(\left[\bar{\bar{I}}_{AM}\right]\begin{pmatrix}\dot{\tilde{u}}\\ \dot{\tilde{v}}\\ \dot{\tilde{w}}\end{pmatrix} + S_\omega^c \cdot \left(\bar{\bar{I}}_{AM}\right)\begin{pmatrix}\tilde{u}\\ \tilde{v}\\ \tilde{w}\end{pmatrix}\right) \quad (90)$$

$$\vec{M}_{AM} = -(b2c)^T\left(\left(\bar{\bar{I}}_{AI}\right)\begin{pmatrix}\dot{\tilde{p}}\\ \dot{\tilde{q}}\\ \dot{\tilde{r}}\end{pmatrix} + S_\omega^c \cdot \left(\bar{\bar{I}}_{AI}\right)\right) \quad (91)$$

where:

$$\bar{\bar{I}}_{AM} = \begin{pmatrix}A & 0 & 0\\ 0 & B & 0\\ 0 & 0 & C\end{pmatrix} \quad (92)$$

$$\bar{\bar{I}}_{AI} = \begin{pmatrix}P & 0 & 0\\ 0 & Q & 0\\ 0 & 0 & R\end{pmatrix} \quad (93)$$

For the sake of simplicity, the spanwise camber effect has been neglected in this formulation.

Finally, the dynamic system is obtained by replacing all the forces and moments in Eqs. (78) and (79). This results [20]

$$\begin{pmatrix} m\vec{I}_1 + \left(\bar{\bar{I}}'_{AM}\right) & -\left(\bar{\bar{I}}'_{AM}\right)S^b_{CG,X_{am}} \\ S^b_{CG,X_{am}}\left(\bar{\bar{I}}'_{AM}\right) & \bar{\bar{I}}_T + \left(\bar{\bar{I}}'_{AI}\right) - S^b_{CG,X_{am}} \cdot \left(\bar{\bar{I}}'_{AM}\right)S^b_{CG,X_{am}} \end{pmatrix}\begin{pmatrix}\dot{u}\\ \dot{v}\\ \dot{w}\\ \dot{p}\\ \dot{q}\\ \dot{r}\end{pmatrix} = \begin{pmatrix}\Phi_x\\ \Phi_y\\ \Phi_z\\ M_x\\ M_y\\ M_z\end{pmatrix} \quad (94)$$

where:

$$\Phi = \overrightarrow{F_A} + \vec{F}_{PL} + \vec{F}_W - mS_\omega^b \begin{pmatrix} u \\ v \\ w \end{pmatrix} - (b2c)^T S_\omega^c \cdot \left(\bar{\bar{I}}_{AM} \begin{pmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{pmatrix} \right) - \left(\bar{\bar{I}}'_{AM} \right) S_\omega^b \cdot (i2b) \vec{V}_{wind} \quad (95)$$

$$M = \vec{M}_A + S_{CG,LE}^b \cdot \vec{F}_A + S_{CG,PL}^b \cdot \vec{F}_{PL} - S_\omega^b \cdot \left(\bar{\bar{I}}_T \right) \begin{pmatrix} p \\ q \\ r \end{pmatrix} - S_{CG,X_{am}}^b \cdot (b2c)^T S_\omega^c \cdot \left(\bar{\bar{I}}_{AM} \begin{pmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{pmatrix} \right)$$

$$+ (b2c)^T \left(S_\omega^c \cdot \bar{\bar{I}}_{AM} \right) \begin{pmatrix} \tilde{p} \\ \tilde{q} \\ \tilde{r} \end{pmatrix} - \left(S_{CG,X_{am}}^b \left(\bar{\bar{I}}'_{AM} \right) \right) S_\omega^b \cdot (i2b) \vec{V}_{wind} \quad (96)$$

In Eqs. (94), (95) and (96), the following notation is used

$$\bar{\bar{I}}'_X = (b2c)^T (I_X)(b2c) \quad (97)$$

The way in which system (94) is solved is explained in Section 3.2.3.

3.2 Numerical implementation

GPSim source code follows the dynamic model commented in the previous section and couples the aerodynamic solver developed in Section 2 in order to calculate the canopy aerodynamic forces and moments. In this section, an overview of the main parts in which this code is organized is presented. Firstly, the required input data is presented along with the geometry modelling; secondly, the initialization process, which prepares the code for the integration loop, is described, and, finally, some remarks about the integration of the equations of motion are given.

3.2.1 Input data and geometry

Input data is transferred to the code by means of three text files which must be defined by the user:

- *configuration.dat*: It includes data related to numerical parameters (number of panels, time step, etc.) and related to initial attitude and velocity of the system (roll angle, components of angular and linear velocities, etc.)
- *geometry.dat*: It defines geometry parameters such as span, chord, sweep, taper, etc. and others needed for the reference coordinates of the system, such as the payload C.G. or the apparent mass centre. It also includes c.s. parameters.
- *inertia.dat*: It includes the system's mass and inertia.

The data inside these txt files is copied in three matrices "C", "G" and "Iner" and thus, the code has all the data stored in memory. For more information about the input data see Appendices (3.1.1, 3.1.2 and 3.1.3).

After that pre-processing of data, the code execution starts. By means of a subroutine called CONFIGURATION, each of the values stored in matrix "C" is assigned to its corresponding variable. Then, subroutine GEOMETRY reads "G" matrix and also assigns its values to different variables. In addition, it calls COORDINATES function, which generates the canopy geometry (panelling procedure). Also in the GEOMETRY subroutine, the reference velocity for the adimensionalization of forces and moments is defined together with the aerodynamic velocity seen by each of the canopy panels. It is then convenient to explain now, how are velocities defined.

As mentioned before, it is necessary to stablish a reference velocity. For current purpose, it has been decided that the most convenient is to define it as:

$$\vec{V}_{ref} = \vec{V}_{wind} - \vec{V}_{cg} \quad (98)$$

This would be the velocity seen by all the canopy panels if no rotation was present. However, usually the system will rotate so it is important to notice that an extra linear velocity will appear and thus, each panel will see a different total velocity. This component of linear velocity caused by the rotation is:

$$\vec{V}_{rot} = \vec{\omega} \times \vec{r} \quad (99)$$

where $\vec{r}$ is the vector from the system's C.G. to the control point of the corresponding panel. $\vec{r}$ must be calculated as a sum of three other vectors because its components are not directly known in body coordinates (see Figure 54):

$$\vec{r} = \overrightarrow{X_{CG}LE} + \vec{r}_{ref} + \vec{C}_p \quad (100)$$

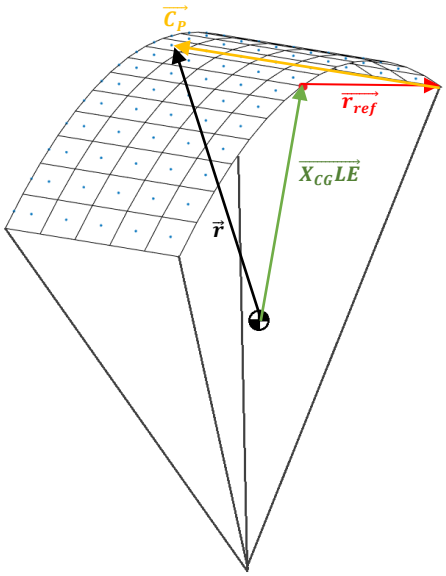


Figure 54: Calculation of the position of a specific control point

In Eq. (100), the vector $\overrightarrow{X_{CG}LE}$ is known in body frame because it is one of the required inputs. $\vec{C}_p$ and $\vec{r}_{ref}$ can also be easily obtained in Matlab frame because all the coordinates of the canopy are known in this coordinate system. Hence, applying the proper transformations $\vec{r}$ is obtained in body frame as

$$\vec{r} = \overrightarrow{X_{CG}LE} + b2c^T \cdot (m2c \cdot \vec{r}_{ref}^{matlab}) + b2c^T \cdot \left(m2c \cdot \vec{C}_p^{matlab}\right) \quad (101)$$

Finally, the total velocity seen by each of the canopy panels is

$$\vec{V}_\infty = \vec{V}_{ref} - \vec{V}_{rot} \quad (102)$$

$\vec{V}_\infty$ is given to the aerodynamic solver every time step in order to calculate the aerodynamic forces and moments.

3.2.2 Initialization

Before starting the integration of motion equations, the initial value of forces and moments is calculated. This is done together with other routines such as the generation of inertia and apparent mass matrices and the calculation of atmosphere conditions according to ISA (International Standard Atmosphere). The initialization of aerodynamic forces and moments needs a special treatment; it is not only about computing the initial loads using the values introduced by the user. To avoid problems with their calculation during the firsts steps (caused by the initial short length of the wake and the high

unsteady forces generated), the wake is forced to be at least 10 times longer than the canopy span. Moreover, this is also done later during the integration loop to ensure that the wake always maintain its length. To do so, the length of the panels is multiplied by a certain increase factor (ifact) and the next considerations apply for the initialization process.

- The number of steps used to develop the wake is the same as the number of wake panel rows introduced by the user.
- All rows of wake panels are forced to be the same length.
- The time step used is the same as the introduced by the user.
- The velocity used to develop the wake is the initial reference velocity.

As said, it is required that the length of the wake is

$$l = 10 \cdot b \quad (103)$$

Hence, the length of the wake without applying the increase factor can be expressed as the number of rows of panels in the wake multiplied by the length of a panel:

$$l_w = N_{xw} \cdot \Delta x \quad (104)$$

where the length of a panel Δx, is the distance travelled in a single time step at a speed equal to the reference speed:

$$\Delta x = V_{ref_i} \cdot \Delta t_i \quad (105)$$

Finally, since we want the length of the wake (104) to be equal to the desired one (103), introducing (105) in (104), multiplying but the increase factor and equalling to (103) the increase factor at the initialization process can be obtained:

$$10 \cdot b = N_{xw} \cdot V_{ref_i} \cdot \Delta t_i \cdot ifact_i \quad \rightarrow \quad ifact_i = \frac{10 \cdot b}{N_{xw} \cdot V_{ref_i} \cdot \Delta t_i} \quad (106)$$

Furthermore, during the integration loop

- All rows of wake panels are forced to be the same length. This is to avoid problems caused by short wakes.
- The time increment used is not constant. It changes through the integration process. (see Section 3.2.3).
- The velocity used to develop the wake is the current reference velocity.

In this case, the number of rows of panels in the wake is already the maximum and wake length is the desired one. Since the reference velocity and the timestep are not necessarily the same as the initials, the increase factor must be recalculated. In order to obtain its new value, it is forced that the new row of panels that appears in each step, maintains the same length as the rest. That way, the wake will preserve its initial length.

$$V_{ref} \cdot \Delta t \cdot ifact = V_{ref_i} \cdot \Delta t_i \cdot ifact_i \quad \rightarrow \quad ifact = \frac{V_{ref_i} \cdot \Delta t_i \cdot ifact_i}{V_{ref} \cdot \Delta t} \quad (107)$$

It is important to mention that this correction of the wake length is an approximation that is not, physically speaking, rigorous. This is because if wake panels are made longer, also their vorticity/circulation should be modified conveniently. A simple approximate approach of addressing that issue is by correcting their circulation proportionally to their variation of area, i.e.

$$\Gamma^*_{\text{panel}} = \Gamma_{\text{panel}} \cdot \frac{A_{panel}}{A_{panel} + \Delta A_{panel}} \quad (108)$$

which would be somewhat similar to reduce the vortex strength when the length of the vortex lines composing the panels increases.

3.2.3 Integration loop

In the last part of the code, equations of motion are solved numerically to simulate the trajectory of the system. GPSim uses three different explicit methodologies for time integration:

- Euler method
- Runge Kutta of 2nd order
- Runge Kutta of 4th order

The one used to carry out the verification tests is the Euler method because it is the faster one and, for current purposes, it shows a good performance. Regarding the integration of the equations of motion and their numerical solvers, an important test to be made is the determination of the maximum time increment admissible by the explicit integration method. The numerical experiments showed that it must have an order of magnitude equal to or less than 1e-4 seconds (although for some specific initial conditions 1e-3 order of magnitude have also worked properly). These very low time increments suppose that the aerodynamic solver subroutine must be called thousands of times for the simulation of just a couple of seconds, and this is not efficient in terms of computational cost.

In order to improve the efficiency of the simulator, it was decided to avoid recalculation of the aerodynamics at each time step, and two conditions to determine whether it should be recomputed or not were defined. These conditions are that, neither the angle rotated by the system in any of the axes nor the variation of the angle of attack from the last call to the aerodynamic subroutine, must be greater than a certain tolerance defined by the user. In this way, the number of calls to the aerodynamic solver is considerably reduced, allowing faster simulations with no significant impact on the accuracy of the results. Conversely to aerodynamic loads, the other forces and moments of the system are update at each time step during the integration process.

It is important to note that since the aerodynamics can be frozen by several steps, special attention must be paid when passing the time increment to the aerodynamic solver. This value will no longer be equal to the current time step used for the

integration, but will be equal to the time that has passed since the last time the subroutine was called. Otherwise, the simulation would not be consistent with the timeline.

3.3 Numerical results

3.3.1 Verification of the time integration method

The first test has been the verification of the correct integration of the equations of motion. To this end, the simulation of a free fall and a parabolic shot have been performed. These movements have a simple analytical solution, what facilitates comparison. In GPSim this has been done by only accounting for the contribution of weight force and setting all other forces to zero. Regarding the analytical solution, it has been also developed in a Matlab code. The free fall has been simulated from 2000 m height and without initial velocity, while the parabolic shot has been simulated with an initial height of 2000 m and an initial horizontal velocity of 10 m/s. Results obtained with the simulator show a deviation of 0.006% in the final position with respect to the analytical code for a time step of 2e-4 (see Figure 55).

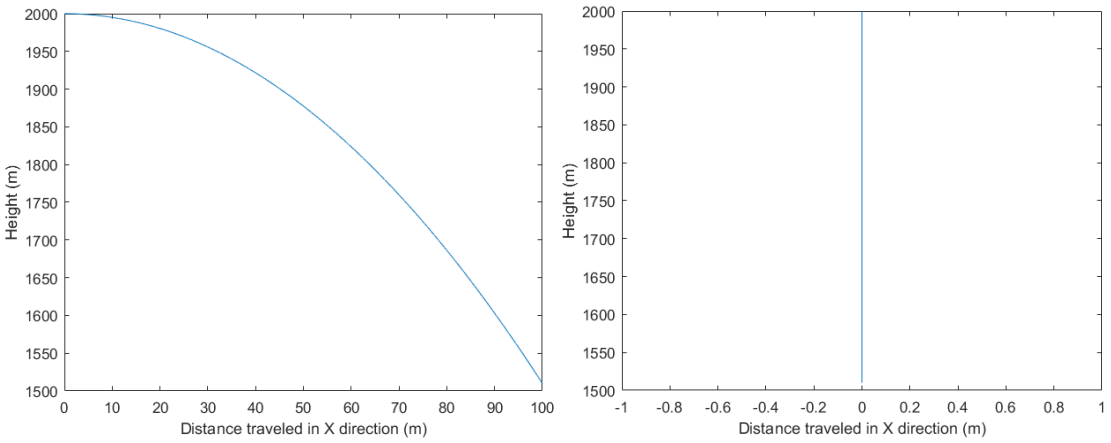


Figure 55: On the left, parabolic shot trajectory simulation after 10 s. On the right, free fall trajectory simulation after 10s. Results obtained with GPSim

3.3.2 GPSim with parametric aerodynamics (aerodynamic solver uncoupled)

The coupling between the simulator and the aerodynamic solver is a complex process and can arose many integration problems. Hence, it has been decided that first it is better to investigate the correct operation of each of the codes separately. Therefore, GPSim is tested with a simulation in which not only weight is taken into account, but also aerodynamic forces and moments. However, instead of calculating the latter with the aerodynamic solver, analytical expressions based on the derivatives have been used. Recall that derivatives are constant coefficients that express the dependence of forces or moments with changes in some variables. For this test, those coefficients have been obtained from available experimental data of a ram-air parachute provided by CIMNE. Calculating forces with that methodology reduces the numbers of possible compilation error due to its simplicity and thus, more attention can be payed to the simulator performance.

The initial conditions and the main geometrical parameters of the canopy for this simulation are given in Tables (9) and (10). The the trajectory obtained from the simulation is shown in Figure 56.

Initial velocity (u,v,w) (body coordinates)	(10,0,0) (m/s)
Initial altitude	200 m
Timestep	0.0002 s

Table 9: Initial Conditions of the simulation made with GPSim with the aerodynamic solver uncoupled

Planform	Rectangular
Chord	0.686 m
Span	1.36 m
Rigging angle	-12˚

Table 10: Main geometrical parameters of the simulation made with GPSim with the aerodynamic solver uncoupled

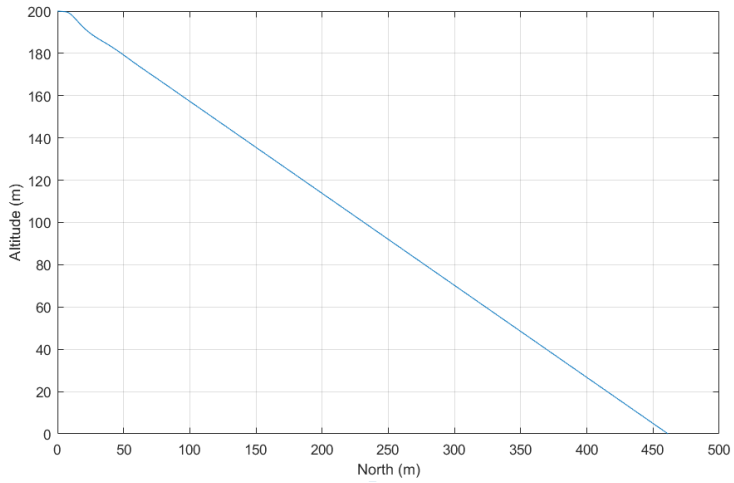


Figure 56: Trajectory of the parachute calculated with GPSim with the parametric aerodynamics.

This trajectory was compared with that obtained with PARASIM6 code of CIMNE for the same simulation conditions and showed to match perfectly. This fact verifies the proper integration of equations when also aerodynamic forces are present.

3.3.3 Aerodynamic solver with user defined trajectories

To continue with the verification of the proper performance of both the simulator and the aerodynamic solver separately, some tests have been done to the aerodynamic solver. This time it is the simulator what is uncoupled from the problem. Instead of calculating forces and moments with the attitude angles and velocities computed by the simulator, the solver now calculates forces and moments for a specific trajectory which has been previously defined by the user. Those tests are useful to detect possible errors of the aerodynamic solver when obtaining results in some particular situations. Two examples of those tests are explained below.

In the first test case (1), the trajectory shown in Figure 57 is studied. The system moves forward with a constant speed of 10 m/s with respect to the inertial system. The number of time steps analysed is ten; in the first five the parachute rotates forward and just in the fifth, there is a sudden change of direction of rotation so that the following five are analysed with rotation in the opposite direction. The time step employed is 0.5s in order to avoid problems caused by small timesteps and the grid has 28x14 panels.

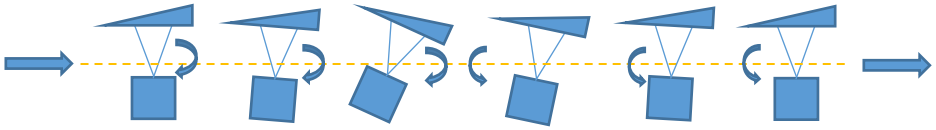


Figure 57: Test case (1): trajectory defined by the user to be analysed with the aerodynamic solver.

The expected behaviour is as follows:

- During forward rotation, the lift force should be increasingly positive (in body coordinates) because of the negative angle of attack. The pitching moment should also be increasingly positive (it tends to re-raise the system to recover the equilibrium position).
- During the back rotation to recover the initial attitude, the lift force should remain positive, but will decrease each time step to zero in the latter. In the same way, the moment should remain positive, but decreasing.

This test allowed checking if the forces and moments obtained with the aerodynamic solver have a logical magnitude and sign. Moreover, as there is a sudden change in direction of rotation, it is possible to check if there is any problem with this change from the point of view of updating the wake.

The results obtained (not reported here) have shown the expected behaviour.

The second test (2) is the same as before, but with the difference that it is not the C.G. what moves. It is the wind what impinges on the system (which remains motionless and rotating about its C.G.). The idea is to verify if the wind speed and the corresponding transformations between axes are correctly computed. The results should coincide exactly with those in test 1 because, aerodynamically speaking, it is exactly the same a system advancing at 10 m/s over a still air than a still system which receives a wind of 10 m/s. After the simulation, the results (not reported here) have been shown to match perfectly with those of test one.

In addition to the previous two tests others have been done, especially intended to check the behaviour of the wake when sudden changes in flight conditions occur. In addition, it has also been verified by doing these tests that the wake length correction mentioned in section 3.2.2 is totally necessary to avoid the wake panels being too short, what supposes numerical problems, even worse if the wake rollup is activated. In Figure

58 a lateral view of a flat plate simulation is shown. Notice that the fact of using short timesteps has supposed the generation of a chaotic wake.

Figure 58: Chaotic wake caused by its short length.

3.3.4 Simulation tests with GPSim coupled with the aerodynamic solver

The aim of this section is to investigate the performance of the fully coupled simulator. The simulation carried out consists on a rectilinear descent. The parachute starts at a height of 200 m with an initial horizontal velocity. The parachute analysed is a small ram-air parachute whose geometry was provided by CIMNE. For the sake of simplicity, the canopy has been supposed to have a symmetrical airfoil. The contribution of all forces and moments have been taken into account (i.e aerodynamic, weight, payload and apparent mass). Below, Tables 11 and 12 summarise the initial conditions and geometrical parameters. Some data is not disclosed due to confidentiality issues.

Initial configuration parameters	
Initial velocity	[25, 0, 0] m/s
Initial angular velocity	[0,0,0] rad/s
Initial parachute attitude $[\phi, \theta, \Psi]$	[0,0,0] degrees
Initial altitude	200 m
Timestep	2e-04 s
Panels along the span	10
Panels along the chord	10
Wake panel rows	10
Type of panel distribution	Uniform
Wind velocity	[0,0,0] m/s

Table 11: Initial configuration for the simulation of a rectilinear descend of a ram-air parachute.

Geometrical parameters	
Planform	Rectangular
Chord	0.686 m
Span	1.36 m
Rigging angle	-12˚
NACA Airfoil	0018
Control surfaces modelling	Not used in this simulation

Table 12: Geometrical parameters for the simulation of a rectilinear descent of a ram-air parachute.

The calculated results are shown in Figure 59 below. After an initial stage exhibiting smooth oscillations, the forces and moments acting on the parachute finally reach an equilibrium and the parachute descends with a constant sink rate. This was, in fact, the expected behaviour.

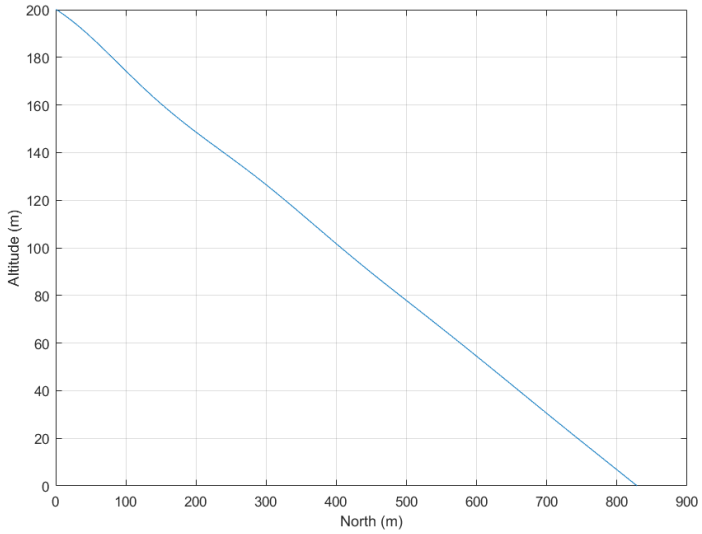


Figure 59: Parachute simulated trajectory

Something that has been checked in this first result is the trajectory angle of the parachute. By inspection of (Figure 59) this angle can be calculated as follows:

$$path\ angle = \operatorname{atan}\left(\frac{0-78}{830-500}\right) \approx -13.3° \quad (109)$$

If the simulation is correct, by analysing the transient evolution of the angle defined by the C.G. velocity of the parachute it should be noticed that the stabilisation pitch angle is similar to that calculated above. This is proved in Figure 60, where the absolute value of the path angle is represented.

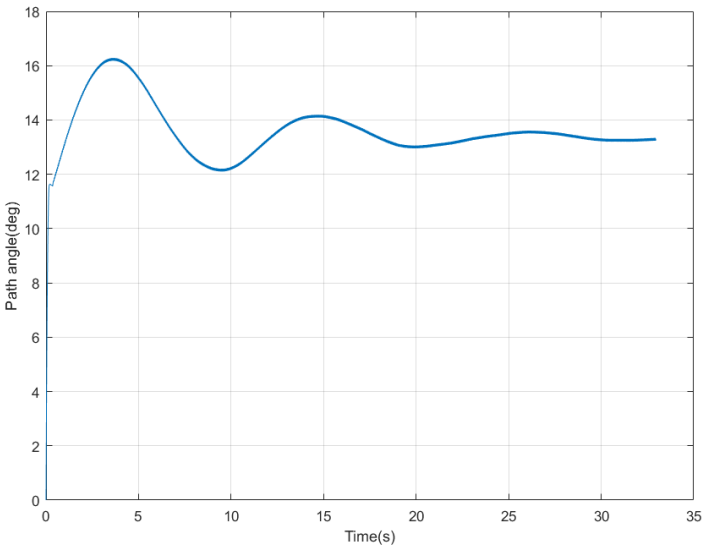


Figure 60: Evolution with time of the system's path angle (deg).

Another test made in this simulation is the influence of the initial conditions in the transient evolution of the system. It has been shown that, as it would be expected, the closer the initial condition is to that of the equilibrium, the faster the system converges and the smoother the oscillations are. In addition, it has also been proved that the convergence value is, at the end, the same. This is confirmed by the analysis of the

evolution of the pitch and angle of attack for two different initial conditions (Figures 61 and 62). The first condition is close to the equilibrium state and is, in fact, the one presented in Table 11 (orange lines in the graph). The second one changes the initial velocity from 25 to 10 m/s what forces the system to start in a condition farther from the equilibrium one.

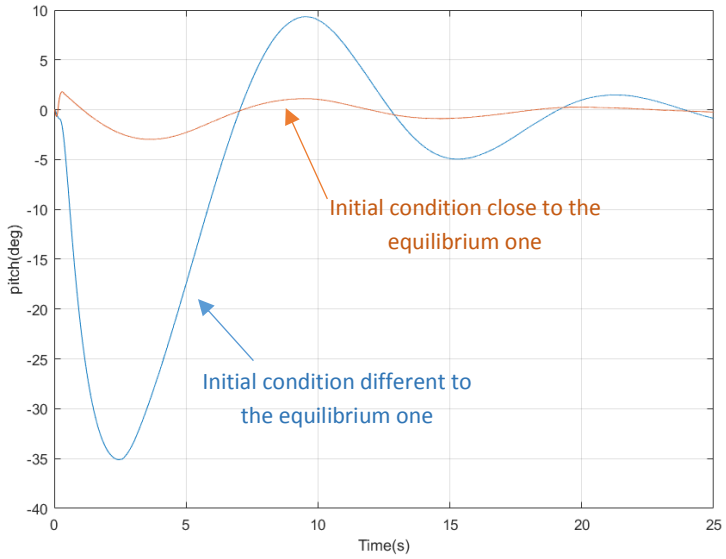


Figure 61: Evolution of the pitch angle (deg). Effect of the initial condition on the transient behaviour.

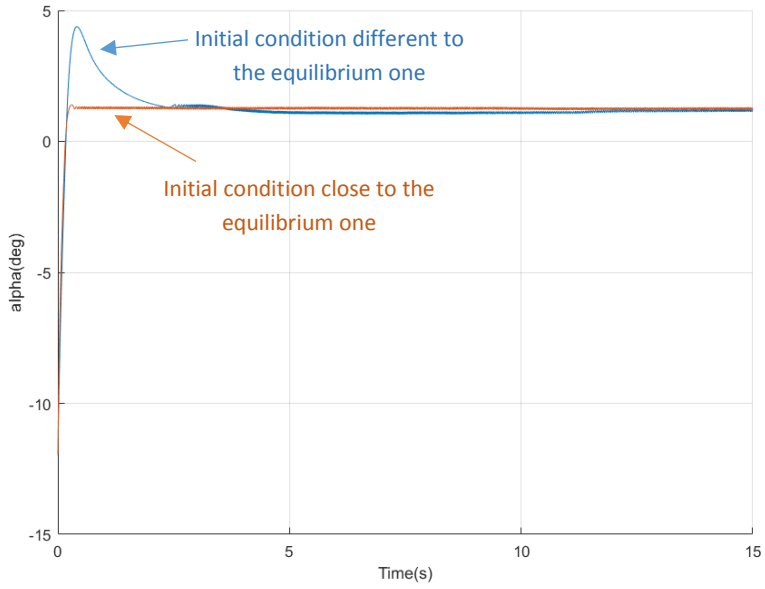


Figure 62: Evolution of the angle of attack (deg). Effect of the initial condition on the transient behaviour.

Chapter 4

Conclusions

The objective of this work was the study and development of a methodology for the flight simulation of ram-air parachutes systems. The chosen methodology, based on an unsteady vortex lattice method and semi-empirical corrections to account for drag effects, was designed to obtain an efficient tool allowing realistic and rapid analyses. The initial objectives of the work were very challenging, and we can say that, in general lines, the results of this thesis have been very satisfactory. It has been possible to develop both the aerodynamic solver and the flight simulator fulfilling all the requirements and the scope marked at the beginning of the study. The tests carried out in the corresponding verification sections confirm that.

It is worth mentioning that before and during the development of this thesis, several decisions have had to be made to achieve these objectives, starting from the title and ending with the smallest detail of the code. One of those choices was the use of UVLM as potential flow solver of the simulator, which has finally proved to be a good decision due to the good results obtained in the tests. Using Matlab programming language has also showed to be favourable because the author already had some basic notions about it, what avoided the introduction of any extra difficulties. However, the inherent difficulty in the development of both programs (which were written completely from scratch) has supposed that, considering the short available time for the developing of this project, other aspects had to be relegated. For example, the analysed cases presented have not been exhaustive. Hence, further verification and validation assessment would be conducted in future developments of the numerical tool.

Getting both codes to run without errors and providing physical results has not been an easy task, quite the contrary. The development of numerical programs is not something studied in depth in the aerospace degree curricula, so it was necessary a fast learning process about programming which added to that necessary for the theoretical aspects of the methodologies applied. The development, in general, has been slow because of the numerical errors or "bugs" that have appeared and that have meant delays in the initial planning. Moreover, these errors have not always been easy to find because, in this type of scripts, many subroutines are used and, therefore, in order to find the error, it is often necessary to perform a "debug process" or, in other words, step by step verification of the code execution. In fact, a large amount of hours have been dedicated to this purpose, especially in the case of the flight simulator. In addition, different theoretical aspects of the methods developed (which are not very clear in the literature) also had to be studied in detail, and required extensive testing.

Regarding the flight simulator, it has been verified that the initial condition greatly affects the stability of the code, causing it to diverge if the condition introduced is not close to the equilibrium. Although it may seem simple from the outside, arriving at this conclusion and getting a correct simulation of the descent of a parachute has been, in fact, one of the most time-demanding task of all the study. Regarding the aerodynamic solver performance, although finally being satisfactory, it is necessary to mention that, in some special cases, the results may diverge if the value of the unsteady force contribution is not limited. Hence, further verification and validation assessment would also be recommended here.

Finally, I would like to comment some aspects of the computational performance of the simulator, because one of the requirements of this thesis was the achievement of reliable and fast results. For this reason, during the whole development, special care has been payed to the efficiency of the calculations. All the effort made in this regard, has led to a substantial improvement of execution time. In fact, for a coarse mesh (i.e. 15x8 panels) the simulation time for a single time step was near to 120 seconds with the first code version. Now the execution time is less than 1 second. After a lot of tests done, different techniques and programming patterns that improve the code performance have been found, particularly in the case of MATLAB. Some examples are the use of functions instead of long scripts, the preallocating of variables before accessing them within loops, the use of scalar operations instead of vector operations and the minimization of calls to built-in functions (e.g. for a dot product calculation it is better to write the corresponding operations than using the dot product built-in function). Other aspects that also have an important impact are the correct storing and accessing of data (in columns in MATLAB) and the code cleaning (revision of unnecessary sections of code and creation of new variables). These considerations can be useful for any other numerical implementation made with MATLAB because, despite being a software which make thinks easy for the user, its performance is generally low, so it is necessary to take care of the way of programming.

Chapter 5

Future work

As it has been mentioned above, the time available during the development of this work was not enough to carry out a more extensive verification and validation of the numerical methodology. Hence, the first and most important thing that should be done in future works is a more exhaustive validation of results for more complex test cases. Beyond this task, some other possible improvements or extensions can be considered.

One is the implementation of an automatic flight control, which would give the possibility of simulating automatic guided flights. In addition, the developed methodology could be extended such that it can simulate other types of lifting surfaces or even include an empennage or a fuselage to the simulation. This would open a wide broad of possibilities to the code such as the simulation of UAV guided flights. This extension would also be fruitful for the simulation of planes designed by the TRENCALOS team, which is composed of students of ESEIAAT.

Another area of improvement, already mentioned, would be the efficiency of the code. Considering the code parallelization in order to improve the execution time or translating it to other programming languages such as FORTRAN, C++ or Python, which usually show better performance when compiling it, could bring significant.

Chapter 6

References

1. GUERRA, A. *Development and validation of a numerical code for analysing parachutes*. Undergraduated thesis presented at Universitat Politècnica de Catalunya. Escola Tècnica Superior d'Enginyeries Industrial i Aeronàutica de Terrassa, 2009.

2. COCKREL, D J. *The aerodynamics of parachutes*. AGARD, 1987. ISBN 9283504224.

3. STATION, Cameron and RIA, Alexan. *Performance and Design Criteria for Deployable Aerodynamic Decelerators*. 1963.

4. BBC News | SCI/TECH | Da Vinci's parachute flies. [online]. [Accessed 24 February 2017]. Available from: http://news.bbc.co.uk/2/hi/science/nature/808246.stm

5. MEYER, Jan. An Introduction to Deployable Recovery Systems. . 1985.

6. GOLD, David. Early Development of the Manually Operated Personnel Parachute, 1900-1919. In : *Aerodynamic Decelerator Systems Conference*. El Centro, 1968.

7. GOLD, David. *A Look at the Hoffman Triangle Parachute - The First Successful Glidable Parachute*. Naval Weapons Center. Aerosystems department publication.

8. KNACKE, T. W. Technical-historical development of parachutes and their applications since World War I. In : *9th Aerodynamic Decelerator and Balloon Technology Conference*. Reston, Virigina : American Institute of Aeronautics and Astronautics, 7 October 1986.

9. Low-altitude parachute-extraction system. [online]. [Accessed 27 February 2017]. Available from: https://en.wikipedia.org/wiki/Low-altitude_parachute-extraction_system

10. NASA. Rogallo Wing. The story told by NASA. [online]. [Accessed 22 March 2017]. Available from: https://history.nasa.gov/SP-4308/ch11.htm#382

11. Rogallo Wing. [online]. [Accessed 22 March 2017]. Available from: https://en.wikipedia.org/wiki/Rogallo_wing

12. BRS Parachutes. [online]. [Accessed 21 February 2017]. Available from: http://www.brsaerospace.com/BRS_History.html

13. CIMNE. PARAPLANE Project. [online]. Available from: http://www.cimne.com/paraplane/vpage/1/0/Intro

14. The plane that can detach its entire CABIN in the event of an emergency | Daily

Mail Online. [online]. [Accessed 28 February 2017]. Available from: http://www.dailymail.co.uk/sciencetech/article-3401939/The-plane-detach-entire-CABIN-event-emergency-Concept-drop-passengers-land-using-built-parachutes.html

15. KNACKE, Theo W. *Parachute recovery systems design manual*. 1st ed. ParaPublisher, 1992. ISBN 0-915516-85-3.

16. H. W. BIXBY, E. G. EWING, T. W. Knacke. *Recovery Systems Design Guide*. 1972.

17. fxc military parachutes. [online]. [Accessed 12 April 2017]. Available from: http://www.fxcguardian.com/products/personnel-parachute-systems

18. Wikipedia-Parachute. [online]. [Accessed 12 April 2017]. Available from: https://en.wikipedia.org/wiki/Parachute#/media/File:USMC_Paratrooper.jpg

19. LORENZ, Ralph. *Spinning flight : dynamics of frisbees, boomerangs, samaras, and skipping stones*. Springer Science+Business Media, 2006. ISBN 9780387472898.

20. SACCO, C, ORTEGA, E and FLORES, R. Development of a 6-DoF Simulator for Analysis and Evaluation of Autonomous Parafoil Systems. *Publication CIMNE N^{o}-356, April 2011*. 2011. No. April.

21. Airborne Systems. [online]. Available from: http://airborne-sys.com/es/

22. STARA Technologies, Corp. An ATSC Company. [online]. [Accessed 2 June 2017]. Available from: http://www.staratechnologies.com/mosquito.html

23. ORTEGA, Enrique, FLORES, Roberto and PONS-PRATS, Jordi. Ram-Air Parachute Simulation with Panel Methods and Staggered Coupling. *Journal of Aircraft*. 6 September 2016. DOI 10.2514/1.C033677.

24. KATZ, JOSEPH; PLOTKIN, Allen. *Low speed aerodynamics. From wing theory to panel methods*. 2nd Editio. INTERNATIONAL EDITION, 2005. ISBN 0-07-100876-4.

25. TEZDUYAR, T E, BEHR, M, MITTAL, S and LIOU, J. a New Strategy for Finite-Element Computations Involving Moving Boundaries and Interfaces - the Deforming-Spatial-Domain Space-Time Procedure .2. Computation of Free-Surface Flows, 2-Liquid Flows, and Flows With Drifting Cylinders. *Computer Methods in Applied Mechanics and Engineering*. 1992. Vol. 94, no. 3, p. 353–371. DOI 10.1016/0045-7825(92)90060-W.

26. TEZDUYAR, T. E., BEHR, M. and LIOU, J. New strategy for finite element computations involving moving boundaries and interfaces. The deforming-spatial-domain/space-time procedure. I. The concept and the preliminary numerical tests. *Computer Methods in Applied Mechanics and Engineering*. 1992. Vol. 94, no. 3, p. 339–351. DOI 10.1016/0045-7825(92)90059-S.

27. STEIN, Keith, BENNEY, Richard, KALRO, Vinay, JOHNSON, Andrew and TEZDUYAR, Tayfun. Parallel computation of parachute fluid-structure interactions. *14th Aerodynamic Decelerator Systems Technology Conference*. 1997. P. 277–284. DOI 10.2514/6.1997-1505.

28. TEZDUYAR, T., KALRO, V. and GARRARD, W. Parallel computational methods for

3D simulation of a parafoil with prescribed shape changes. *Parallel Computing*. 1997. Vol. 23, p. 1349–1363. DOI 10.1016/S0167-8191(97)00057-4.

29. GARRARD, William L, TEZDUYAR, Tayfun E, ALIABADI, Shahrouz K, KALRO, Vinay, LUKER, Joel, MITTAL, Sanjay, WASHINGTON, A, MINNEAPOLIS, South D and BEACH, Clearwater. Inflation analysis of ram air inflated gliding parachutes. *13th Aerodynamic Decelerator Systems Conference*. 1995. No. March. DOI doi:10.2514/6.1995-1565.

30. ALIABADI, S, GARRARD, W, KALRO, V, MITTAL, S, TEZDUYAR, T and STEIN, K. Parallel finite element computation of the dynamics of large ram air parachutes. In : *13th Aerodynamic Decelerator Systems Technology Conference*. Reston, Virigina : American Institute of Aeronautics and Astronautics, 15 May 1995.

31. TAKIZAWA, Kenji and TEZDUYAR, Tayfun E. Computational Methods for Parachute Fluid-Structure Interactions. *Archives of Computational Methods in Engineering*. 2012. Vol. 19, no. 1, p. 125–169. DOI 10.1007/s11831-012-9070-4.

32. TAKIZAWA, Kenji, TEZDUYAR, Tayfun E. and KOLESAR, Ryan. FSI modeling of the Orion spacecraft drogue parachutes. *Computational Mechanics*. 23 June 2015. Vol. 55, no. 6, p. 1167–1179. DOI 10.1007/s00466-014-1108-z.

33. AQUELET, Nicolas and TUTT, Benjamin. A new fluid structure coupling Application to parachute modelling. *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique*. 2007. Vol. 16, no. 3–4, p. 521–536. DOI 10.3166/remn.16.521-536.

34. KIM, Yongsam and PESKIN, Charles S. 3-D Parachute simulation by the immersed boundary method. *Computers and Fluids*. 2009. Vol. 38, no. 6, p. 1080–1090. DOI 10.1016/j.compfluid.2008.11.002.

35. KIM, Yongsam and PESKIN, Charles S. 2–D Parachute Simulation by the Immersed Boundary Method. *SIAM Journal on Scientific Computing*. January 2006. Vol. 28, no. 6, p. 2294–2312. DOI 10.1137/S1064827501389060.

36. KARAGIOZIS, K., KAMAKOTI, R., CIRAK, F. and PANTANO, C. A computational study of supersonic disk-gap-band parachutes using Large-Eddy Simulation coupled to a structural membrane. *Journal of Fluids and Structures*. 2011. Vol. 27, no. 2, p. 175–192. DOI 10.1016/j.jfluidstructs.2010.11.007.

37. LS-DYNA | Livermore Software Technology Corp. [online]. [Accessed 7 March 2017]. Available from: http://www.lstc.com/products/ls-dyna

38. COQUET, Yves, BORDENAVE, Pascal, CAPMAS, Guillaume and ESPINOSA, Christine. Improvements in Fluid Structure Interaction Simulations of Parachutes Using LS-Dyna®. *21st AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*. 2011. No. May, p. 1–9. DOI doi:10.2514/6.2011-2590.

39. TUTT, Benjamin, CHARLES, Richard, ROLAND, Scott and NOETSCHER, Greg. Development of Parachute Simulation Techniques in LS-DYNA ®. . No. 5, p. 25–36.

40. DOHERR, Karl-Friedrich. 10 - Parachute Flight Dynamics and Trajectory Simulation. *NASA PARACHUTE SEMINAR*. 2005.

41. SLEGERS, Nathan, BEYER, Eric and COSTELLO, Mark. Use of Variable Incidence Angle for Glide Slope Control of Autonomous Parafoils. *Journal of Guidance, Control, and Dynamics*. May 2008. Vol. 31, no. 3, p. 585–596. DOI 10.2514/1.32099.

42. MORTALONI, Paul, YAKIMENKO, Oleg, DOBROKHODOV, Vladimir and HOWARD, Richard. On the Development of a Six-Degree-of-Freedom Model of a Low-Aspect-Ratio Parafoil Delivery System. In : *17th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*. Reston, Virigina : American Institute of Aeronautics and Astronautics, 19 May 2003. p. 1–10. ISBN 978-1-62410-087-1.

43. LINGARD, John Stephen. Ram-Air Parachute Design. In : *13th Aerodynamic Decelerator Systems Technology Conference*. Reston, Virigina : American Institute of Aeronautics and Astronautics, 1995. p. 1–51.

44. www.nasa.gov. [online]. [Accessed 2 June 2017]. Available from: https://www.nasa.gov/images/content/201991main_NOC1197_330.jpg

45. www.publicdomainpictures.net. [online]. [Accessed 2 June 2017]. Available from: http://www.publicdomainpictures.net/pictures/80000/velka/parachutist-with-yellow-parachute.jpg

46. AENOR - Normas, buscador de normas. [online]. [Accessed 18 April 2017]. Available from: http://www.aenor.es/aenor/normas/buscadornormas/buscadornormas.asp#.WPXKvmnyipo

47. HOERNER, S. F. *Fluid Dynamic Lift*. 2nd. 1985. ISBN 9998831636.

48. ORTEGA, Enrique. *The Finite Wing*. 2014.

49. ORTEGA, Enrique. *The fluid flow equations*. 2014.

50. ORTEGA, Enrique. *Incompressible Irrotational Flow*. 2014.

51. ORTEGA, Enrique. Horseshoe Vortex Method. .

52. GIAIOTTI, Dario B. and STEL, Fulvio. The Rankine Vortex Model. *October*. 2006. P. 1–14.

53. SIMPSON, Robert J. S., PALACIOS, Rafael and MURUA, Joseba. Induced-Drag Calculations in the Unsteady Vortex Lattice Method. *AIAA Journal*. 2013. Vol. 51, no. 7, p. 1775–1779. DOI 10.2514/1.J052136.

54. XFLR5. [online]. [Accessed 5 June 2017]. Available from: http://www.xflr5.com/xflr5.htm

55. LAMBERT, Thomas and DIMITRIADIS, Grigorios. Induced Drag Calculations with the Unsteady Vortex Lattice Method for Cambered Wings. *AIAA Journal*. 2017. Vol. 55, no. 2, p. 668–672. DOI 10.2514/1.J055135.

56. GÓMEZ TIERNO, Miguel Ángel, PÉREZ CORTÉS, Manuel and PUENTES MÁRQUEZ, César. *Mecánica del vuelo*. 2ª ed. Garceta, 2012. ISBN 9788415452010.

57. LAMB, Horace. *Hydrodynamics*. Cambridge University Press, 1993. ISBN 9780521458689.

58. LISSAMAN, P. and BROWN, GLEN. Apparent mass effects on parafoil dynamics. *Aerospace Design Conference*. 1993. Vol. 39, no. 3, p. 445–451. DOI 10.2514/6.1993-1236.

59. JOHARI, Hamid. Aerodynamic Characteristics of Cargo Ram-Air Canopies during Maneuvering. *California State University, Northridge, California* [online]. 2010. [Accessed 28 February 2017]. Available from: https://sites.google.com/site/alieslambolchi/researches/aerodynamic-characteristics-of-cargo-ram-air-canopies-during-maneuvering