Flight Planning Support Tool for Airspace Design

Carolina Nascimento Alves

Thesis to obtain the degree of Master of Science in

Aerospace Engineering

Supervisors: Prof. Pedro da Graça Tavares Álvares Serrão
Eng. Jesús Conde Gomez

Examination Committee
Chairperson: Prof. José Fernando Alves da Silva
Supervisor: Prof. Pedro da Graça Tavares Álvares Serrão
Member of the Committee: Prof. Rodrigo Martins de Matos Ventura

October 2018
Acknowledgments

I would like to express my gratitude to my supervisors, without whose wisdom and constant availability this dissertation would not have been possible: to Prof. Pedro Serrão for the guidance and support given throughout this thesis, and to Eng. Jesús Conde for the opportunity to work with such a challenging subject and the readiness to help. I would also like to thank NAV, and in particular António Pinto da Silva and Vanda Cruz, for the availability, patience and kindness shown in all of the meetings during the development of this thesis.

My appreciation goes also to all my friends, without whom this journey would not have been the same. A special thanks to the group "Mas Somos Três" for the fun memories, for making long work hours go by so much easier and for the help through difficult moments.

For all the patience, help, and loving support, a very special thank you to Diogo Mafra.

Finally, I would like to thank my family, and specially my parents, for the unconditional confidence in me, for the cheering on that kept me motivated and for being the support system that allowed me to succeed.
Resumo

O crescimento do tráfego aéreo dos últimos anos colocou uma pressão acrescida nos prestadores de serviços de tráfego aéreo, tanto a nível econômico como de segurança. Esta tendência impõe uma necessidade de melhoria do espaço aéreo em termos ambientais, de capacidade e eficiência. O replaneamento da rede atual de rotas para uma maior eficiência é crucial, e para isto, o conhecimento do uso corrente do espaço aéreo existente e a comparação dos indicadores de desempenho com uma potencial rota ótima torna-se essencial.

O âmbito desta tese é o desenvolvimento de uma ferramenta que recebe como dados do projetista de espaço aéreo dois pontos, e gera ficheiros que fornecem informação com a qual é possível avaliar, das rotas atualmente disponíveis, as mais utilizadas, e contem indicadores de desempenho para uma hipotética rota direta. A ferramenta integra dados históricos para se servir de uma maior base de informação, da qual pode retirar dados estatísticos que permitam a avaliação das rotas. Consiste numa base de dados, onde é possível a manutenção e atualização da informação, e uma aplicação que calcula os indicadores de desempenho e gera os ficheiros de saída: um com os dados estatísticos, outro com a representação gráfica das rotas selecionadas.

A ferramenta desenvolvida permite a análise de rotas entre qualquer ponto de navegação e/ou aeroporto. Foi possível verificar que a rota mais curta nem sempre é a mais eficiente em termos de custos, devido às taxas de CRCO, devendo este fator ser considerado no planeamento de novas rotas.

Palavras-Chave: Gestão de Tráfego Aéreo, Design do Espaço Aéreo, Análise de Rotas, Informação de Voo, Linha Ortodrómica
Abstract

The air traffic growth of recent years has put economic and safety pressures on Air Navigation Service Providers, and this trend demands an improvement of airspace performance on capacity, efficiency and environmentally. A key part of this is the redesign of the current route network for more efficient routes. For this, the knowledge of actual use of the existing airspace and the comparison of performance indicators with a possible optimum route becomes essential.

The scope of this thesis is the development of a tool that receives as input from the airspace design user two route points, and produces outputs that provide useful information for an evaluation of the most chosen available routes between those points, as well as the calculations of the same performance indicators for the hypothetical direct route. The tool gathers historical information in order to have a larger base of information from which to retrieve the statistical data for route evaluation. It consists of a database, where the data can be updated, maintained and incremented to the most recent date, and an application that calculates the performance indicators and generates output files: a statistical data file and a visual file that allows for a graphical representation of the routes.

The developed tool enables route comparison and analysis between any navigation point and/or airport. It was possible to verify that the shortest route is not always the most cost-efficient, because of CRCO charges, and that this factor should be taken into consideration in new proposed routes.

Keywords: Air Traffic Management, Airspace Design, Route Analysis, Flight Information, Great Circle Line
Table of Contents

Acknowledgements ........................................................................................................ ii
Resumo ............................................................................................................................ iii
Abstract ............................................................................................................................... iv
List of Figures .................................................................................................................... iv
List of Tables ....................................................................................................................... vii
Abbreviations ..................................................................................................................... ix

Introduction ....................................................................................................................... 1
1.1 Motivation .................................................................................................................... 1
1.2 Objectives .................................................................................................................... 2
1.3 Thesis Outline ............................................................................................................. 3

Fundamentals ..................................................................................................................... 4
2.1 Air Navigation Concepts and Spherical Trigonometry ................................................ 4
  2.1.1 Great circle distance ............................................................................................ 4
  2.1.2 Bearing .................................................................................................................. 5
  2.1.3 Two arc cross ........................................................................................................ 5
  2.1.4 Cross-track distance ............................................................................................ 8
  2.1.5 Point in Polygon – Ray Casting Algorithm ....................................................... 8
2.2 Airline Economics ...................................................................................................... 9
  2.2.1 CASK .................................................................................................................. 10
  2.2.2 Flight Operating Costs ....................................................................................... 11
2.3 Air Traffic Management ............................................................................................ 12
  2.3.1 AIRAC cycle ........................................................................................................ 12
  2.3.2 Flight planning .................................................................................................... 12
  2.3.3 CRCO charges ..................................................................................................... 12
  2.3.4 DDR2 – Demand Data Repository .................................................................... 13

NEST Software .................................................................................................................. 14
3.1 Data Filter ................................................................................................................... 14
3.2 CRCO Charge Information ....................................................................................... 16
3.3 Absent Features ......................................................................................................... 18
# List of Figures

| Figure 1. 1: Annual predicted IFR flight movements (EUROCONTROL [1]) | 1 |
| Figure 2. 1: Representation of a great circle line (ANAC[6]) | 4 |
| Figure 2. 2: Two-arc cross on a spherical surface (T.Champougny [5]) | 6 |
| Figure 2. 3: Points P’ and P outside and inside the polygon, respectively. | 9 |
| Figure 2. 4: RASK (revenue per available seat per kilometer) and CASK (cost per available seat per kilometer) for multiple carriers [10]. | 10 |
| Figure 2. 5: Worldwide Airline Industry fuel consumption analysis[12]. | 11 |
| Figure 2. 6: Rule of Thumb for Functional Cost Distribution. | 11 |
| Figure 3. 1: NEST User Interface. | 14 |
| Figure 3. 2: Selective query result in the NEST Tool. | 15 |
| Figure 3. 3: Flight list of the selected traffic, on the selected day (NEST software). | 16 |
| Figure 3. 4: Route detail of one specific flight (NEST Software). | 16 |
| Figure 3. 5: Flight selection and export to obtain CRCO charges (NEST Software). | 17 |
| Figure 3. 6: Process to obtain intersection data files for CRCO charges calculation (NEST software). | 17 |
| Figure 3. 7: Interface to obtain the CRCO Charges (NEST Software) | 18 |
| Figure 4. 1: Flight Planning Support Tool scheme | 20 |
| Figure 4. 2: C++ application interface with some inputs already given by the user. | 25 |
| Figure 4. 3: Orthodromic route between two points and corresponding great circle line | 26 |
| Figure 4. 4: Distances per charging area. | 29 |
| Figure 4. 5: Detail image of the airspace delimited by the coordinates of the LE (Spain) and GM (Morocco) areas, as given by the .are file. | 31 |
| Figure 4. 6: The orthodromic line between waypoints VASTO and DELOG cross the LP frontier (right polyhedron of the image) several times. | 31 |
| Figure 4. 7: Green line representing the planned route from LEMD to GCLP. | 34 |
| Figure 4. 8: Route from and BAROK (lower yellow dot) to CCS (blue dot). | 34 |
| Figure 4. 9: Detail of real flight between VASTO and DELOG navigation points | 35 |
| Figure 4. 10: FIRS.kml file open in Google Earth | 36 |
| Figure 4. 11: The two first sections of the route_statistics.csv file from the LFPO-SAMAR trajectory, imported into Microsoft Excel. | 37 |
| Figure 4. 12: Planned and real flight statistics for a LEMD-GCLP flight. | 38 |
| Figure 4. 13: Extra section with routes grouped by airport, from a LFPO to SAMAR route. | 38 |
| Figure 4. 14: Route between VASTO and DELOG navigation points, with the direct route in red, the planned in green and the real in blue. | 39 |
| Figure 4. 15: Individuals flights from VASTO to DELOG represent in Google Earth. | 39 |
Figure 4. 16: Overlapping frontiers: the shaded blue zone is the common area. ........................................ 40
Figure 4. 17: Representation of a real route that re-enters the same charging zone two times, the actual distances flown (blue lines) and the distance considered (orange line). ......................... 41

Figure 5. 1: Charging zone frontiers and labels. .......................................................................................... 42
Figure 5. 2: LEMD-GCLP trajectories in Google Earth.............................................................................. 44
Figure 5. 3: Orthodromic route for LEMD-GCLP ..................................................................................... 44
Figure 5. 4: LFPO-SAMAR trajectories in Google Earth............................................................................. 47
Figure 5. 5: Real routes 2 and 9 represented in blue, orthodromic route represented in red ............. 48
Figure 5. 6: Actual individual flights, complete (airport-to-airport), represented in yellow .......... 49
Figure 5. 7: Direct and planned routes for ECKOS-EGSS in Google Earth......................................... 51
Figure 5. 8: Orthodromic route (red line), real flight routes (blue lines) and the shortest planned route (green) ........................................................................................................... 52
Figure 5. 9: Direct route for SULAM-DELOG represented in Google Earth.......................................... 54
Figure 5. 10: Planned flights vs orthodrome for SULAM-DELOG ........................................................ 55
Figure 5. 11: Real flights vs orthodrome for SULAM-DELOG ............................................................... 55
Figure 5. 12: SULAM to DELOG trajectories represented in Google Earth ......................................... 56
Figure 5. 13: Actual flights that pass through the SULAM and DELOG waypoints ....................... 58
Figure 5. 14: Comparable flights from NEST software (left) and the Flight Planning Support tool (right) for LEMD-GCLP........................................................................................................... 59
Figure 5. 15: Comparable flights from NEST software (left) and the Flight Planning Support tool (right) for LFPO-SAMAR ............................................................................................................ 59
Figure 5. 16: Comparable flights from NEST software (left) and the Flight Planning Support tool (right) for ECKOS-EGSS ............................................................................................................. 60
Figure 5. 17: Detail from a flight plan from LEMD to GCLP (NEST Software). ..................................... 61
Figure 5. 18 – Cost analysis for LEMD-GCLP on an A320 ................................................................ 61
Figure 5. 19: Cost analysis for LFPO-SAMAR on an A320 ................................................................... 63
Figure 5. 20: Cost analysis for ECKOS-EGSS on an A320 .................................................................. 63
Figure 5. 21: Cost analysis for SULAM-DELOG on an A320 ............................................................... 64
# List of Tables

Table 5. 1: Direct route information for LEMD-GCLP ................................................................. 42
Table 5. 2: Planned flight statistics for LEMD-GCLP ................................................................. 43
Table 5. 3: Real flight statistics for LEMD-GCLP ................................................................. 43
Table 5. 4: Orthodromic route detail for LEMD-GCLP .......................................................... 45
Table 5. 5: Planned flights vs. orthodrome for LEMD-GCLP .................................................. 45
Table 5. 6: Real flights vs. orthodrome for LEMD-GCLP .......................................................... 45
Table 5. 7: Direct route information for LFPO-SAMAR ............................................................ 46
Table 5. 8: Orthodromic route detail for LFPO-SAMAR .......................................................... 46
Table 5. 9: Planned flight statistics for LFPO-SAMAR ............................................................... 46
Table 5. 10: Real flight statistics for LFPO-SAMAR ................................................................. 46
Table 5. 11: Planned flights vs. orthodrome for LFPO-SAMAR ................................................ 48
Table 5. 12: Real flights vs. orthodrome for LFPO-SAMAR ....................................................... 49
Table 5. 13: Direct route information for ECKOS-EGSS ............................................................ 50
Table 5. 14: Orthodromic route detail for ECKOS-EGSS ......................................................... 50
Table 5. 15: Planned flights statistics for ECKOS-EGSS ............................................................. 50
Table 5. 16: Planned flights vs orthodrome for ECKOS-EGSS ................................................ 50
Table 5. 17: Real flights statistics ECKOS-EGSS ........................................................................ 51
Table 5. 18: Real flights vs orthodrome for ECKOS-EGSS ....................................................... 52
Table 5. 19: Direct route information for SULAM-DELOG ....................................................... 53
Table 5. 20: Orthodromic route detail for SULAM-DELOG ...................................................... 53
Table 5. 21: Planned flights statistics for SULAM-DELOG ....................................................... 54
Table 5. 22: Real flights statistics for SULAM-DELOG ............................................................. 56
Table 5. 23: Planned flights vs orthodrome for SULAM-DELOG ............................................... 57
Table 5. 24: Real flights vs orthodrome for SULAM-DELOG .................................................... 57
Table 5. 25: Error assessment for LEMD-GCLP planned flights ............................................. 59
Table 5. 26: Error assessment for LFPO-SAMAR planned flights ........................................... 60
Table 5. 27: Error assessment for ECKOS-EGSS planned flights ........................................... 60
Abbreviations

SES – Single European Sky
FRA – Free Route Airspace
ATS – Air Traffic Services
ATM – Air Traffic Management
ANSP – Air Navigation Service Providers
CRCO – Central Route Charges Office
CSV – Comma-Separated Values
FIR – Flight Information Region
ICAO – International Civil Aviation Organization
IFR – Information Flight Rules
MTOW – Maximum Take-Off Weight
CASM – Cost per available seat per mile
CASK – Cost per available seat per kilometre
AIRAC – Aeronautical Information Regulation and Control
ECAC – European Civil Aviation Conference
1. Introduction

1.1 Motivation

Over the past few years, air traffic has increased and will continue to do so according to the seven-year forecast report from EUROCONTROL [EUROCONTROL Seven-Year Forecast February 2018, EUROCONTROL, February 2018]. Even though the growth rate is predicted to slow down by comparison to the 2017 growth (4%), the information from the Monthly Network Operations Report of June 2018 [2] indicates that there was a growth of 4.1% compared with the homologous previous period. The forecast still indicates that traffic will rise through all of the predicted years, whichever the scenario (high (H), baseline (B) or low (L)):

![Figure 1. 1: Annual predicted IFR flight movements (EUROCONTROL [1])](image)

Traffic flow growth can put safety and economic pressures on aircraft operators and air navigation service providers (ANSP), along with the environmental impacts of CO2 emissions [3]. This issue started being addressed in the early 2000’s with the creation of the Single European Sky (SES) framework by the European Commission, which, motivated by the delays observed in the 1990s and 2000s, aims to reform the air traffic management (ATM) system across Europe and reduce delays, increase safety and diminish the environmental footprint.

When it comes to improvement of ATM, in 2008, EUROCONTROL, in its capacity of Network Manager, started the coordinated development and implementation of Free Route Airspace (FRA)[3], in which a user can plan the route between a defined entry and a defined exit point without the constrictions of a fixed route network. This initiative has the intent to improve airspace performance on capacity, efficiency and environment, bringing advantages for all stakeholders by granting the possibility to make flights more cost-effective and to diminish flight time – which can consequently reduce the number of conflicts.

The ATS Route Network needs to progress along with the FRA development, for an overall
airspace interconnectivity: this is particularly important in regions where there is a fixed ATS route network below the FRA, as found in the Lisbon FIR, or where there is a non-FRA area adjacent to an operational FRA area, and the definition of entry/exit points must be so that it avoids potential conflicts and won’t be damaging to capacity. Both the implementation of FRA and the improvement of the ATS Route Network are contemplated as objectives of the European Route Network Improvement Plan (ERNIP) [4].

Member states and air navigation services entities are moving towards an airspace design approach that answers the demands of the current European aviation landscape. Being that a cooperative effort towards the maximization of route efficiency is in place, to know and understand both the current traffic flows and the flight planning patterns becomes crucial in the decision making of airspace design operators.

Airspace Design requires modelling and simulation tools that make use of real historical data on air traffic flow. A potential airspace route network change needs to be based on a thorough analysis that validates the new scenario and evaluates the impacts on capacity and flight efficiency. Currently, the modelling and simulation resource used by EUROCONTROL and ANSPs is the NEST (Network Strategic Tool) Modelling Tool, a piece of software with visualization and analytical features, as well as simulation algorithms.

For the analysis of historical information, the NEST Tool displays traffic for a specific past day, either in its entirety or with filters applied – such as arriving/departing airports, or those that cross specific sectors. It also shows flight details like total length and route waypoints.

However, some features that are not available in the NEST Tool are needed for the study and redesign of the route network. There is a need to compare existing data, from both flight plans and real flight records, to the orthodromic route between the same limit points. The Central Route Charges Office (CRCO) costs that these routes imply must also be analysed. Finally, these assessments need to be made for a relevant period of time, as opposed to only for a specific day.

The tool developed in this thesis addresses these needs and its outputs allow for a better route analysis.

1.2 Objectives

The main goal of this thesis is to develop a tool that allows for a better understanding of the use of the current airspace network, through the use of statistical features and ways of displaying information that are not available with the existing software.

Making use of methods for calculating route performance measures like total distance and Air Traffic Services (ATS) costs, the objective is the creation of a new software tool that has the ability to generate an output that will answer the following needs:

- Comparison of the hypothetical orthodromic route with the corresponding planned and real flight routes between any two airspace points;
- Ability to aggregate and select historical information for any route:
- from a waypoint to another waypoint;
- from an airport to another airport;
- from waypoint to airport (and vice-versa).

- Ability to have a representative display of the gathered most-used existing routes.

Therefore, two output files are created with the new developed tool: a .csv file with the numerical and statistical information, and a .kml file that allows for the visualization of the selected routes in the Google Earth application.

1.3 Thesis Outline

The present dissertation is organised as follows:
- chapter 1 contains a presentation of the motivation for the present work and the main goals that are expected to be achieved;
- chapter 2 contains an explanation of the concepts necessary for the comprehension of this thesis, as well as the formulas, models and algorithms that are used in the development of the Flight Planning Support Tool and the validation of its results;
- chapter 3 presents the currently existing and available software, with the detail of its most important features for this context;
- chapter 4 details the implementation of the Flight Planning Support Tool, with a section for each of the software components - the database and the executable application - and a presentation of its output files;
- chapter 5 presents a case study for each of the possible types of user input combinations, validates the results comparing them with results obtained from the NEST Software, and presents a cost analysis for each case study that make use of the performance indicators of the tool (distance and charging areas costs);
- chapter 6 lists the conclusions, achievements and future work to be done in the intention of continuing the development of this tool.
2. Fundamentals

This chapter contains an explanation of all the concepts, models and algorithms that are used throughout this thesis.

2.1 Air Navigation Concepts and Spherical Trigonometry

The analysis and study of routes will imply calculating route length, assessing intersection points of routes with charging areas frontiers and gauging distances from points to trajectories. This thesis uses the same assumptions that the Central Route Charges Office (CRCO) uses for the calculus of intersections between flight paths and charging area boundaries [5]:

- the Earth will be considered a sphere, and the ellipsoidal effects of its real shape will be ignored;
- the optimal path is considered the shortest path between two points on the surface of the sphere; this means that the longest optimal path possible will be half the perimeter of the Earth.

This thesis will also consider a value of 6371.0 km for the radius of the Earth.

2.1.1 Great circle distance

The direct route of a real flight is not the length of the straight line between two sets of coordinates, but the distance measured on the surface of the Earth. The path between two points on said surface is the shortest arc of the great circle line, or orthodrome, which goes through those points. A great circle line results from the intersection of a plane that passes through the centre of a spherical surface.

![Figure 2.1: Representation of a great circle line (ANAC[6])](image)
The great circle distance can be obtained by the haversine formula [5], in which given two points on the surface of the Earth, expressed in Cartesian coordinates, the distance between those two points is given by:

\[
d = 2R \arcsin \left( \sqrt{\sin^2 \left( \frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left( \frac{\lambda_2 - \lambda_1}{2} \right)} \right)
\]  

(1)

Where \(d\) is the distance, in km; \(\varphi_1, \varphi_2\) is the latitude of the initial and ending point, respectively (in radians); \(\lambda_2, \lambda_1\) is the longitude of the initial and ending point, respectively (in radians); \(R\) is the radius of the Earth, in km.

### 2.1.2 Bearing

The initial bearing of a path is the direction or course of motion that the aircraft needs to follow along the great circle line in order to get from the initial to the final intended point. It is the angle between the Earth’s magnetic North and the plane direction. Given two sets of coordinates, the initial bearing (from Point 1 to Point 2) is given by [7]:

\[
\alpha = \arctan \left( \frac{\sin(\lambda_2 - \lambda_1) \cos(\varphi_2)}{\cos(\varphi_1) \sin(\varphi_2) - \sin(\varphi_1) \cos(\varphi_2) \cos(\lambda_2 - \lambda_1)} \right)
\]

(2)

Where \(\alpha\) is the bearing in radians, \(\varphi_1, \varphi_2\) is the latitude of the initial and ending point, respectively (in radians); \(\lambda_2, \lambda_1\) is the longitude of the initial and ending point, respectively (in radians).

### 2.1.3 Two arc cross

To determine the entry or exit point from a hypothetical orthodromic route to a charging area, the intersection between the great circle arcs that represent the trajectory and the area frontier must be calculated. The method used in this thesis is the same defined by EUROCONTROL Airspace Division (DED4) [5].

The arcs are defined by two pairs of points: (P1, P2) for the first arc and (P3, P4) for the second arc. The origin of the referential in the image below coincides with the centre of the Earth.
Each point is geographically represented by latitude and longitude. The first step is to transform the geographical coordinates – in radians – into the Cartesian coordinates (R is the radius of the sphere, in this case, the Earth):

$$\begin{align*}
    x_1 &= R \cos(lat_1) \cos(lon_1) \\
    y_1 &= R \cos(lat_1) \sin(lon_1) \\
    z_1 &= R \sin(lat_1)
\end{align*}$$

Each pair of points identifies a plane that cuts the Earth by its centre. From these defining points the vector director of the plane, which is perpendicular to the plane, can be obtained by the vectorial product of the two points:

$$V_{12} = P1 \times P2 \quad (4)$$

$$\begin{bmatrix}
    v_{12x} \\
    v_{12y} \\
    v_{12z}
\end{bmatrix} =
\begin{bmatrix}
    y_1z_2 - y_2z_1 \\
    x_2z_1 - x_1z_2 \\
    x_1y_2 - x_2y_1
\end{bmatrix} \quad (5)$$

And the unit vector ($U_{12}$) of the vector director ($V_{12}$) as well ($l$ is the length of the vector director):

$$l = \sqrt{v_{12x}^2 + v_{12y}^2 + v_{12z}^2} \quad (6)$$

$$U_{12} = \begin{bmatrix}
    u_{12x} \\
    u_{12y} \\
    u_{12z}
\end{bmatrix} = \begin{bmatrix}
    v_{12x}/l \\
    v_{12y}/l \\
    v_{12z}/l
\end{bmatrix} \quad (7)$$

The two planes that contain the great circle arcs, if not identical, will cross along a line that is described by the following set of equations:
\[
\begin{align*}
(u_{12x}x + u_{12y}y + u_{12z}z &= 0 \\
(u_{34x}x + u_{34y}y + u_{34z}z &= 0)
\end{align*}
\] (8)

Where \((u_{34x}, u_{34y}, u_{34z})\) are the coordinates for the unit vector director of the plane defined by the point \((P_3, P_4)\) with the centre of the Earth. The intersection line of the two planes has a vector director \(D\) that is the vectorial product of the vector directors of the planes \(U_{12}\) and \(U_{34}\):

\[
D = U_{12} \times U_{34} \tag{9}
\]

The intersection of the two planes and the spherical surface results in two points on the sphere, diametrically opposite. One of these two points is the intended intersection point of the path with the charging area frontier. To obtain the coordinates of these points, we need to calculate the unit vector \(S\) and its inverse \((-S)\) of the vector director \(D\), and multiply it by the Earth’s radius. \(l_D\) is the length of vector \(D\):

\[
l_D = \sqrt{d_x^2 + d_y^2 + d_z^2} \tag{11}
\]

\[
S_1 = \left(\begin{array}{c} s_{1x} \\ s_{1y} \\ s_{1z} \end{array}\right) = \frac{1}{l_D} \left(\begin{array}{c} d_x \\ d_y \\ d_z \end{array}\right) \tag{12}
\]

\[
S_2 = -S_1 = \left(\begin{array}{c} s_{2x} \\ s_{2y} \\ s_{2z} \end{array}\right) = \frac{1}{l_D} \left(\begin{array}{c} -d_x \\ -d_y \\ -d_z \end{array}\right) \tag{13}
\]

The next step converts the Cartesian coordinates back to geographical. The arc sine function is used to determine the latitude and returns a value from \(\frac{\pi}{2}\) to \(-\frac{\pi}{2}\), that correspond to the North and South Pole, respectively. The longitude is obtained with the atan2 function (2-argument arctangent), that returns an angle between \(-\pi\) and \(\pi\) that is the angle from the x axis (latitude=0, longitude=0) to the projection on the x0y plane of the ray that goes from the centre of the sphere to the point on the surface, \((R_m\) is the Earth’s radius in meters):

\[
l_{lat_1} = \arcsin(s_{1z}) \tag{15}; \quad l_{lat_2} = \arcsin(s_{2z}) \tag{16}
\]

\[
l_{lon_1} = \text{atan2}(s_{1y}, s_{1x}R_m) \tag{17}; \quad l_{lon_2} = \text{atan2}(s_{2y}, s_{2x}R_m) \tag{18}
\]

The final step implies checking which of the opposite points is the one that corresponds to the
intersection. That point will be present in both arcs. To check if the point belongs to the great circle line arcs, the length of an arc must be equal to the sum of the haversine distances of the intersection point to each extremity of the arc, that is, the following formula must be true for each arc:

$$L_A - L_1 - L_2 = 0 \quad (19)$$

Where $L_A$ is the length of the arc, $L_1$ is the length between the intersection and the initial arc point, and $L_2$ is the length between the final arc point and the intersection.

### 2.1.4 Cross-track distance

This distance corresponds to the minimum distance between a point and a great circle-path – it is the distance used to measure the GPS cross-track error. The formula for this distance, having a great circle path defined by two points, is the following [8]:

$$d = \text{asin}(\sin(\theta) \sin(\alpha_1 - \alpha_2)) \times R \quad (20)$$

Where $d$ is the distance (km), and the sign of this value will indicate on which side of the great circle line the point is; $\theta$ is the angular distance from the first great circle path point to the outer point (rad); $\alpha_1$ is the bearing between the first point of the path and the outer point, in radians; $\alpha_2$ is the initial bearing of the great circle path, in radians; $R$ is the Earth’s radius, in km.

The angular distance is obtained by the formula:

$$\theta = \frac{H_{1-2}}{R} \quad (21)$$

Where $\theta$ is the angular distance (in radians) and $H_{1-2}$ is the haversine distance from the first great circle path point to the second (km).

### 2.1.5 Point in Polygon – Ray Casting Algorithm

Besides determining the crossing points with the charging area boundaries through which the aircraft navigates, the areas that contain the first and last trajectory point also need to be found. For any of the intermediate areas we obtain the distance travelled by calculating the distance between the entry and exit points. For the extremity areas, the distance travelled is the length between the point and the first/last crossing point. These charging zones won’t have an even number of intersections.

Given that we have the coordinates of a point, and a set of ordered points that represent the area
frontier, we face a problem of point-in-polygon, a problem for which several algorithms are available that search for a solution, even for an arbitrary polygon (convex, concave, convoluted or self-intersecting), like Hormman and Agathos indicate [9].

The polygons defined by charging zone boundaries are either convex or concave, and for this type of shape the ray-casting algorithm (that’s based on the even-odd rule) provides a solution.

Given a point P and a closed polygon C, defined by an array of points \( C_0, C_1, ..., C_n = C_0 \), a line from the point is drawn to a point that’s guaranteed to lie outside the polygon and if the number of times this line crosses the edges \( e_i = \overline{C_iC_{i+1}} \) of the polygon is odd, then the point is inside the polygon; if it is even, it is outside[9].

![Figure 2.3: Points P’ and P outside and inside the polygon, respectively.](image)

In figure 2.3, the ray cast from point P to the outside of the polygon intersects the edges three times; given that it is an odd number, the point is inside the polygon. The P’ point intersects two times, and it is outside the polygon. In the tool developed in this thesis, the created algorithm is a loop that goes through each polygon edge. Each edge is a set of two sequential points of a charging zone border – and determines if there is a crossing point with the route path. The intersections are then counted to determine if the initial (and final) path point is inside the area or not.

### 2.2 Airline Economics

The improvement of route efficiency will directly impact airline costs and productivity. The cost of flying a specific trajectory can be used as a measurement of the benefit of reducing route length. In this section an explanation of airline costs is presented, with focus on costs specifically related to the operation of flying the aircraft.
2.2.1 CASK

The cost per available seat per kilometre (CASK) or per mile (CASM) is a way to measure airline efficiency and to compare expenses between companies in the industry. This measure includes costs unrelated to passengers as well, like cargo and maintenance.

![Figure 2.4: RASK (revenue per available seat per kilometer) and CASK (cost per available seat per kilometer) for multiple carriers [10].](image)

Of this total cost, according to the report performed by Oliver Wyman [10], about a third is attributed to fuel, and this cost can be directly correlated with the distance. The International Air Transport Association (IATA) estimates that fuel amounts for around 20% of the total operating expenses for the global airline industry [11].
Because the CASK depends on variables that are not related or influenced by distance flown, like aircraft seat configuration and amenities (low cost airlines have a lower CASK than traditional airlines), in order to evaluate the cost efficiency of different route lengths in this thesis we will only be looking into the Flight Operating Costs (FOC) portion that constitutes the CASK. These costs are directly related with the operation of flying the aircraft.

### 2.2.2 Flight Operating Costs

Like the ICAO Economic Development document [13] describes, the flight operating costs include fuel, pilot salaries, maintenance and aircraft ownership expenses - these are composed of depreciation, leasing costs and insurance. The same document attributes 50% of total cost of a company on direct flight operating costs as the historical “rule of thumb” [13].
2.3 Air Traffic Management

In this section a brief introduction to Air Traffic Management concepts, definitions and tools required for the understanding of this thesis is presented.

2.3.1 AIRAC cycle

The Aeronautical Information Regulation and Control (AIRAC) system is intended to guarantee that all parties involved in air navigation have access to the same information base regarding the airspace structure and design and its changes, at the same time. The AIRAC cycle is a 28-day interval cycle that was adopted by ICAO in 1964. Information provided through the AIRAC cycle has to reach its recipients (Air Traffic Controllers, Air Pilots, Air Traffic Flow Managers) 28 days before the effective change date [14].

2.3.2 Flight planning

All flights into, around and out of Europe that require air traffic services require the submission of a flight plan to EUROCONTROL Network Manager’s Operation Center (NMOC) [15]. The flight plan describes the route that the aircraft is going to follow before the actual flight. However, during the real flight some changes might be made because the expected airspace scenario may not be the same – due to, namely, bad weather, delays, airspace capacity and congested areas.

Information about the planned and real flight is kept and made available by EUROCONTROL through their data demand repository (DDR). A comparative analysis of both is useful in the scope of airspace design, allowing the uncovering of inefficiencies in the airspace structure.

The CRCO charges, explained in the next section, are calculated based on the planned flight.

2.3.3 CRCO charges

Air traffic management of the European airspace users is financed through the billing of CRCO charges. The CRCO acts in the name of the EUROCONTROL Member States, and bills and collects route charges that support air navigation services and facilities [16]. The calculation method of these charges, which was published by the CRCO and included in the tool that's the subject of this thesis, will be explained in this section.

The entry and exit points of each charging zone are determined by the route described in the last flight plan filed. This means that even though the real route might not be exactly the one filed, the charges won't be posteriorly affected by it.

The total en-route charge is a sum of the charges for each charging zone. Each of these charges is defined by the Unit Rate charge, the aircraft weight factor and the distance factor:

\[ \text{Unit Rate} \times \text{Weight Factor} \times \text{Distance Factor} = \text{Charge} \] (22)
There is a Unit Rate for each Member State, which is the charge, in euros, for each distance factor and aircraft weight factor. The rates are applicable as from 1st January of each year and can be consulted on EUROCONTROL's information circulars [17].

Weight factor
The aircraft weight factor is obtained according to the following formula \( \sqrt{\frac{MTOW}{50}} \) where MTOW is the maximum take-off weight, expressed in metric tons, rounded to the first decimal.

Distance factor
The distance factor is the number of kilometres of the orthodromic route between the entry and exit points of the overflown charging zone, divided by 100. In the case where the charging zone contains the departure (or arrival) airport, the distance considered is the orthodromic between the departure airport and the exit point (or the entry point and the arrival airport, respectively) minus 20 km (each).

2.3.4. DDR2 – Demand Data Repository
The EUROCONTROL Demand Data Repository [18] is a web portal that contains historical information about flights in the pan-European sky. The data includes flight intentions (based on the last filed flight plan) and real flight data, and can be downloaded for visualization and analysis in the NEST tool, also available on this web portal. This data can be made available in the form of an AIRAC file that contains all the historical data of the flights on the 28-day interval of that AIRAC.

Besides the planned and actual flight data, the web portal makes other necessary datasets available: airspace environment datasets (like airport locations) and route charge information (CRCO datasets, with the unit rates for the charge areas and an averaged weight for type of aircraft to be used for total charges calculus).

This repository contains all the necessary data used by the tool developed in this thesis.
3. **NEST Software**

The NEST (Network Strategic Tool) is a stand-alone desktop application that supports airspace design and flight planning, with features like simulation and modelling to help predict future demand and traffic distribution, and visualization and analysis of past data that allows for the study of traffic in the pan-European area. It includes the possibility to export data, either as is or with a selection of filters [19].

![NEST User Interface](image)

Figure 3.1: NEST User Interface.

The NEST software features the ability to import an AIRAC scenario, the downloadable file from the DDR website that contains all the flight information from the European airspace that occurred during that AIRAC period. Once imported into the tool, the user is able to explore either the planned or real data on a specific day of the AIRAC period, chosen by selecting the number on the inferior part of the interface – as shown in figure 3.1, where the 8th of June 2017 is selected. Only one type of information is displayed at one time: either the last flight plans submitted for that day (given by the “Initial” scenario filter) or the real flight data (“Actual” filter.)

### 3.1 Data Filter

The Nest Tool shows a daily record of 32 000 – 35 000 flights that translate the traffic throughout all the EUROCONTROL area. The tool allows for the addition of filters to the traffic the user wishes to visualize, which may then be exported in the format of .so6 files that contain all the traffic information for the filtered flights for the selected day. In this thesis, even though the developed tool will work with any trajectory on the European sky, the geographical focus is the Lisbon FIR, for the purpose of
having a manageable amount of data to export from the NEST software and also because it is the main area of interest of NAV Portugal, who will be the end users of the tool. Selecting traffic exclusively around this area restricts the volume of data to 3000 daily flights, which is still a large enough sample from which to build an analysis on. A query was constructed by choosing the sectors that are crossed by these flights and excluding the arrivals/departures of surrounding airports (Annex A.1).

The result of this query is demonstrated in figure 3.2.

Figure 3.2: Selective query result in the NEST Tool. Top-left: the query’s selected traffic b) top-right and c) bottom: the inverse of the query (all the flights of the day except the selected) - for every two points chosen in the "empty" area, all the trajectories that go through those two points are selected – without exception. In c) the LP sector is represented and it is completely in the empty area.

The NEST Tool also displays a list of information of either the planned or actual flights, according to the selected preference, that contains, per flight ID: arrival and departure airport, flight duration, aircraft type and total route length, like shown in figure 3.3.
Figure 3. 3: Flight list of the selected traffic, on the selected day (NEST software).

However, the detailed route information, with the specification of the waypoints crossed, is only displayed in one window per flight, like shown in figure 3.4.

3.2 CRCO Charge Information

The NEST Tool also permits the assessment of the CRCO cost for a specific flight or list of flights. The CRCO charges are displayed by flight ID, with the possibility to be displayed by charging zone as well.

In order to obtain said cost, there must be an export of the traffic data (for a chosen day). The type of file to be exported must be a .so6 file. Figure 3.5 demonstrates the export of traffic data for the 1st of July for flights from the LEMD airport to the GCLP airport:
The traffic type displayed is the Initial, because the CRCO charges are calculated based on the flight plan. Next, an intermediate step of obtaining an intersection file (.t5 file) is necessary for the charges calculation. In the “Processing” tab, the “Airspace/Traffic Intersection” opens the options box that generates this file. In the “Traffic (so6)” input we must include the file that was just exported, and in the “Sectors” input, the CRCO charges files (areas and rates) made available by the DDR. The “Run” button will create the output file with the name indicated in the “T5 output file” section.
After exporting the .so6 and .t5 files for the selected traffic, the excel file with the charges can be obtained.

In the “Analysis” menu, selecting “Route Charge”, the appropriate files should be inputted in the “Inputs” section and the name of the output excel file should be selected in the “Outputs” section. The “CRCO intersection” box must also be ticked to visualize the cost per charging zone.

![Figure 3.7: Interface to obtain the CRCO Charges (NEST Software)](image)

The final Excel file with the necessary information will be available in the application export folder: it displays the charges per flight ID and per charging zone. This type of file will be crucial for the Flight Planning Support Tool results validation.

### 3.3 Absent Features

The way the information is presented by the NEST Tool doesn’t fulfil all the requirements necessary from a re-evaluation and redesign of the route network perspective. In order to analyse existing routes and measure their efficiency, one must be able to group detailed information by a longer period of time. It is also useful to compare statistics of the planned and real flights for the same timeframe. Another important feature is to compare the available and most-used network routes to the orthodromic trajectory between the same initial and final points.

The tool developed in this thesis adds the above mentioned extra features, as well as CRCO costs for the three types of routes (planned, real and orthodromic), which is information that isn’t available in the NEST software.

The developed tool calculates the cost of the hypothetical orthodromic route because, in order to design new routes that are more efficient and less expensive to the airline companies, there’s the necessity to evaluate if, even though the distance is the shortest possible – and therefore, the fuel cost is the minimum possible – the areas through which the aircraft travels can increase or decrease the
CRCO cost. A new route design that takes into consideration a shorter flight distance would be less expensive in terms of flight operation costs, but, it could be less cost-efficient if the trajectory is through an area with a higher CRCO Unit Rate – and in the end, the flight plan may still include a detour through a less expensive area.

The real CRCO charges are calculated based on the last submitted flight plan, however, in this thesis the charges for the actual flown route are also calculated in order to be able to ascertain if the real routes would theoretically increase or decrease the charging costs, and therefore, if a difference is observed, identify that the entry and exit points are in fact different from the planned routes.
4. Flight Planning Support Tool Development

For the development of this tool, two different software applications were used: Microsoft SQL Server Express, to store, manage, and manipulate data, and Visual Studio for the development of the C++ application. Google Earth was also employed to visualize one of the output files (.kml).

Reference data from the European airspace environment and flight data are collected from the Data Demand Repository and the NEST Tool, to be imported into the database created in SQL Server. This data is imported into tables: the reference data is imported as is and a new import substitutes the previous existing table; in the case of flight information, both the raw data and a clean version - where extra unused information is removed - is stored, and new information is appended to the table, so that historical records are kept. It is in the SQL database that filters and aggregations are applied to route information.

The other piece of software used is Visual Studio Interface, where the C++ project was developed. This application is the link between the user and its inputs and the database, and it is also where calculations are performed, like route length and CRCO charges. It is also this application that creates the output files:

- route_statistics.csv file – contains the results of the calculus and the statistical information;
- trajectory.kml file – contains the graphical information, so that there is the ability to visualize the different trajectories on the Google Earth application.

![Figure 4.1: Flight Planning Support Tool scheme](image-url)
4.1 SQL SERVER

4.1.1 Database Structure

4.1.1.1 Data gathering

The data that is sourced directly through the Demand Data Repository are the following files (YYYY and MM are the year and month, respectively, which the file refers to):

1. CRCO_AircraftWeights_YYYYMM.mwc – this file contains a list of aircraft models and their average maximum take-off weight, in tons;
2. CRCO_UnitRates_YYYYMM.ur – lists the charging areas and their unit rate, in cents;
3. CRCO_ChargeAreas_YYYYMM.are – contains, per area code, all the points that define the boundary of that charging area, with one line per point coordinates. These points are sequential, so before the file is imported into the tool, an incremental number is added to the front of each line to guarantee that the order is kept when the information is transferred to the database;
4. VST_YYMM_Airports.arp – lists the ICAO code of the airports in the network and their corresponding coordinates.

The repository also contains the AIRAC files that can be imported and explored in the NEST Tool. Traffic information can then be extracted from the software. This tool allows to export the following files:

- NavPoint_YYYYYMM.npt – contains, per row, the waypoint/navigation points available in the route network and their respective coordinates;
- YYYYMMDDInitial.so6 and YYYYMMDDActual.so6 – these files contain the traffic information of the planned and real flights, respectively, for the indicated date (DD is the day). There is one file of each for each day.

All of these files should be in the same folder, from where the import function will read. The folder in this case is “C:\NEST\Import_Export\”.

4.1.1.2 Database Structure

The data is imported into a SQL database. The choice to transport and maintain the data in a relational database management system (RDBMS) was due to the possibility to easily manage, treat and rearrange a large amount of structured information that's translated into tables that correlate with one another by key entities. In this section a description of the tables, their relations and the data imported into them is presented.

Reference tables:

These tables are lookup tables that contain fixed information pertinent to the database. When the information is updated, it replaces the entire table.
- AIRPORT: lists all the airports of the EUROCONTROL area by their ICAO code. Includes the coordinates and the charging area the airports belong to;
- CRCO_AircraftWeights: lists airplane models and their average take-off weight;
- CRCO_UnitRates: lists the charging zones and the CRCO costs; it includes the reference date of the charges list;
- NAV_POINTS: lists the waypoint names for the ECAC area and their coordinates;
- MAP_DEG: lists the coordinates of frontier points of each charging area as given by the DDR2 database. Each area is delimited by the group of points identified in this table, and the points are listed in sequential order with an order number.

Core tables:
These tables contain all the base information from which the necessary data for the execution of the final tool is extracted:
- SO6_M1 and SO6_M3: these tables contain all the raw information present in the exported flight data from the NEST Tool, for the Initial.so6 and Actual.so6 files, respectively. These tables contain information for multiple days. The flights are described by ordered flight segments, and a segment is distinguished every time there is a change in flight level or the aircraft passes through a reference point. For each segment there is a length, segment id in which the beginning and ending points are identified, and a beginning and ending time, date, flight level, latitude and longitude. The call sign, flight id, aircraft model and departure and arrival airports are also identified in each row.
- SO6_FILTERED_M1(M3): this tables regroups the information from the SO6_M1(M3) table into segments from waypoint to waypoint. The segment id only has the beginning waypoint code.
- FROUTE_M1(M3): this table contains each flight id from the SO6_M1(M3) table, and the route for each id, that is, the sequence of waypoints of the flight, separated by a comma.

Staging tables:
These tables hold temporary data that was treated and prepared from the base information to be used by the final application. They are only filled while executing the final program, and their data is filtered by the inputs given by the end user.
- ROUTES_PER_ROW_M1(M3): this table gathers all the flight ids that satisfy the criteria inserted by the end user, i.e., all the flights that go through the two chosen points; it contains the arrival and departure airport for each flight, the date of flight, the AIRAC cycle to which the date corresponds to and the route between the two limit points, with waypoints separated by commas.
- RouteStatistics_M1(M3): in this table the information from the ROUTES_PER_ROW_M1(M3) is grouped by route. The table contains the beginning and ending point chosen by the user, the arrival and departure airports, and the number of distinct flight ids that flew over the two points, grouped by route.
Auxiliary tables:
The purpose of these tables is to store intermediate information while the processes that upload and treat information are running.

- **ALLFILENAMES**: lists all the files in the data folder. This folder should contain all the files that are meant to be uploaded into the tables (described on section 4.1.1.1).
- **TRAJECTORIES**: stores the coordinates of the route points of each flight id that matches the user criteria. Both the M1 (planned flights) and M3 (real flights) are gathered in this same table – a column named TRAJ makes the distinction between these flights.

### 4.1.2 Stored procedures

A stored procedure is a subroutine that contains processes that access and treat data from a database. These procedures can be called by an application.

In this thesis there are two types of stored procedures: the bulk insert procedures, that upload all the data and fill the database, and the route procedures that are called by the C++ application with the criteria the user chooses and gather and treat the information according to the request.

The bulk insert procedures are meant to run only when there is an update to the database, while the route procedures run each time the tool is executed.

#### 4.1.2.1 Information upload

For purposes of maintenance of this tool, for each of the reference tables, the SQL script rewrites the table according to the corresponding extension file present in the data folder. For this, it is important that only one file of each of these types is present in the folder. The only extension that can have more than one file is the .so6 file.

**BulkInsert_Charges:**
Based on the extension of the file found in a predefined path, it uploads data to the following tables:

1. CRCO_UnitRates;
2. CRCO_AircraftWeights.

**BulkInsert_Geo:**
Uploads information from the data folder for the following tables:

1. MAG_DEG;
2. AIRPORTS;
3. NAV_POINTS.

**spBulkInsertM1(M3):**
This procedure uploads the entire information available from the NEST Tool files from the maximum previously loaded data to the current date. For that, it checks for the most recent reference date
loaded into the tables SO6_M1 and SO6_M3 and only uploads files which are more recent than that date.

After uploading the raw flight data, it fills out two other tables:

1. SO6_FILTERED_M1(M3);
2. FROUTE_M1(M3).

4.1.2.2 Route Information

RouteInfoM1(M3):
This stores procedure collects and aggregates the information of a route between two given points.
The procedure collects all the real routes that contain both the initial and final point, and that occurred in the 5 AIRAC cycles before the given date. If no date is given, the data is relative to the 5 most recent AIRAC cycles present in the database. This is accomplished with the SQL function “NAIRAC”.
After collecting the flights, a sum of the miles of each one is made. The flight information provided by the DDR2 data site is given in flight segments, where each segment initiates at the occurrence of an event during the flight (changing altitude, passing through a waypoint), indicating the end of the previous segment. The number of flown miles is registered for each segment, so for the calculation of the distance overflown between two points, the stored procedure adds the miles of the segments between the points.
The next step writes the trajectory for each flight id, that is, it adds, in order, the waypoints of that flight between the two limiting points on a single string.
The procedure then writes the distinct routes into another table - RouteStatistics_M1(M3) - and groups them by trajectory, counting the number of flights for each different trajectory, adding the average of the distance for each route; it also groups by departure and arrival airports.

4.2. C++ Application

The main application for this tool was developed in C++, in the Visual Studio integrated development environment. This application connects all the parts of the tool, from end to end: communication with the database to extract and transform data, execution of calculations to return route statistics and creation of the output files. This means the end user only needs to execute this application in order to use the Flight Planning Support Tool.
To run the Flight Planning Support Tool, the user can click on the program executable shortcut, and the interface for this application is the command line. This interface requests inputs from the user, validates them, runs the entire program and generates the output files on the project folder – one csv file with detailed data and a kml file with visual data.

4.2.1 Requested Inputs

The first step in this application is to connect with the SQL Server database and get reference data. In the main application the functions that connect to the SQL database are defined. These functions retrieve information from the tables in the SQL Server database and insert them into vectors (numeric and string type) so that the data can be manipulated in the C++ application. A vector corresponds to a column from a table, and the order of the elements in the vector is the same as the table column’s (i.e. first row = first vector element).

The application starts by filling vectors with reference information, like airport codes, navigation points codes, aircraft models and weights, so that the input from the user can be compared and validated with the existing database.

The application then gets the beginning and ending point from the user and compares it with the list of airports and navigation points, and, in case no matches are found, indicates to the user specifically that the initial, final, or both points don’t exist in the database – this means that the user inputs must be either a waypoint name or an airport ICAO code.

In the event of a match, an optional step is presented to the user, to indicate the aircraft type. If no type is provided, the charges calculations will be shown per weight factor (weight factor of 1).

The aircraft type is also subject to validation by the program and, in case of a mismatch with the database, the user is asked for the input again.

With all the necessary inputs from the user, the program calculates the routes.

4.2.2 SQL Data Extraction

For each vector of airport and waypoint codes, there are corresponding coordinate vectors. The coordinates of the limit points of the trajectory are obtained, and the orthodromic length is calculated.
with a function that applies the haversine formula.

Then, the application calls for the stored procedures RoutInfoM1 and RoutInfoM3 mentioned in section 4.1.2, with the information given by the user. After the stored procedures are executed and the database tables are filled out, another function that connects to SQL (get_routes) inserts the route information into vectors. For the planned and real flights, the top five and top ten most frequently used routes are selected, respectively. The program checks if the vectors are empty, both for the planned and the real routes; if so, this means there are no routes between those two points in the database and the user is given the opportunity to reinsert two other points.

From this point onward, the program calculates the distance and CRCO charges, and it is divided into three sections according to the type of route:
1. orthodromic trajectory;
2. planned flight;
3. real flight.

The last two sections consist of a loop that replicates the calculus for each of the retrieved routes.

### 4.2.3 Orthodromic Distance

The following section explains the algorithm to find the intersection points of the direct trajectory with the member charging areas and the distance crossed in each area, to obtain the total distance and hypothetical CRCO Charges.

**Intersection points:**

![Figure 4. 3: Orthodromic route between two points (yellow) and corresponding great circle line (red). The dots represent the intersection of the great circle line with the charging zones – only the white dots are intersections contained in the trajectory.](image-url)
There are two loops in this process:

1. A cycle that detects all the charging areas that a great circle arch passing by both points intersects;
2. A loop through all intersected charging areas detects the ones intersected by the segment of the arch contained within the input points and finds their coordinates.

### Pseudocode 1: intersection points

#### 1. First loop:

```plaintext
while n <= size(CRCO_ZONES) do
    if area = CRCO_ZONES(n):
        point_keep = first point of area frontier
        previous_distance = 0
        for point Є area frontier
            point_next = point
            get distance from point_next to trajectory with cross-track distance formula
            if |distance+previous_distance| < |distance|+|previous_distance|:
                save point_keep and point_next into VECTOR_POINTS
                previous_distance = 0
                point_keep = point_next
            else
                point_keep = point_next
                previous_distance = distance
        area = CRCO_ZONES(n+1)
        n = n+1
end
```

#### 2. Second loop:

```plaintext
m = 0
while m <= size(VECTOR_POINTS) do
    calculates the intersection_point between the trajectory great circle line and the great circle defined by each pair of point_keep and point_next
    if trajectory segment ⊆ intersection_point:
        saves intersection_point to VECTOR_INTERSECTION
        saves intersection_point area to ZONE_INTERSECTION
    m = m+1
end
```
An explanation of the method follows:

1. **First loop**: to detect the charging zones that the great circle arc passes through, we find the intersection points between the arch and the charging zone frontier. The frontiers are given by the .are file from the DDR2 database in the form of a set of coordinates followed by the charging area codes. The coordinates are sequential (i.e. the points are in consecutive order).

   The code loops through all the charging zones, and the purpose is to find the pair of nearest points to the great circle line for each of the areas, a point of each side of the line. This pair will be bordering points of the segment of the frontier that crosses the great circle line. In order to get them, a cycle is run for each point of the frontier, and the cross-track distance from the point to the trajectory is calculated according to the formula described in section 2.1.4. The sign of this distance indicates the side of the path the point is on. Therefore, if two consecutive points have a different sign, then this means they correspond to a pair of coordinates whose segment crosses the path. These pairs are recorded during the cycle, and for each of them we have an intersection point.

2. **Second loop**: goes through each pair of crossing point and the two arc cross is calculated according to section 2.1.3 [5]– the function `two_arc_cross` is run with the input of the coordinates of the four points (2 from the frontier, 2 from the trajectory) and it returns the set of coordinates of the intersection point. In the rare occasion where the initial or ending point is contained in the frontier segment that we’re trying to intersect, the method will not be able to determine the intersection because it coincides with an input point, so the function returns a value of 1000 to indicate such a situation. Afterwards, a validation in the main function is executed to check if it is the first or last point that coincides with the frontier. For all the other calculated intersection points, the program checks if the latitude and longitude are between the latitudes and longitudes of the bordering trajectory points. With this, there is the guaranty that the intersection points are only along the trajectory, as opposed to anywhere on the great circle line that passes through the trajectory.
Distances and direction

Figure 4.4: Distances per charging area. The distances $d_1$ and $d_2$ are the distances from the charging area boundary intersection points to the first point (P1) of the trajectory. The $d_3$ distance is the length of the entire trajectory.

This part of the algorithm consists of three steps:

1. Checks the areas that have an even number of intersection points. This means that they are intermediate areas (i.e., neither the departure nor arrival charging zone).
2. Calculates the distance from the first point to each intersection point – this is intended to assess the order of the intersection points, because it is equivalent to the ascending order of distances.
3. Sorts the distances and, consequently, sorts the charging areas sequentially; the minimum distance corresponds to the first area, and the maximum to the last.

This algorithm is summarized in the pseudocode that follows.
**Pseudocode 2:** calculate the distances and their consecutive order

1. **First step:**
/* VECTOR_INTERSECTION contains the intersection points, and ZONE_INTERSECTION the charging zone for the same index */

```plaintext
n=1
intersected_area = first ZONE
while n <= size(ZONE_INTERSECTION) do
    count=0
    for intersected_area= ZONE_INTERSECTION(n)
        count=count+1
        save ZONE
        save count into count_ZONE
    intersected_area= next ZONE
    n=n+1
end
```

2. **Second step:**

```plaintext
m=1
while m<=size(VECTOR_INTERSECTION) do
    calculate haversine distance d_m between VECTOR_INTERSECTION(m) and 1st trajectory point P1
    save d_m into INTERSECTION_DISTANCE
    m=m+1
end
```

3. **Third step:**

```plaintext
p=1
for each ZONE_INTERSECTION do
    order INTERSECTION_DISTANCE
    if count_ZONE is even:
        for p<=size(INTERSECTION_DISTANCE) do
            save INTERSECTION_DISTANCE(p+1) into MAX_DISTANCE
            p=p+2
        end
    else
        q=1
        for p<=size(INTERSECTION_DISTANCE) do
            if INTERSECTION_DISTANCE(q)= min(INTERSECTION_DISTANCE):
                ZONE is the first point area
                Save distance as first distance
            if INTERSECTION_DISTANCE(q)= max(INTERSECTION_DISTANCE):
                ZONE is the last point area
                Save distance as last distance
            else
                save INTERSECTION_DISTANCE(p+2) into MAX_DISTANCE
                p=p+2
        end
    end
end
```

Explanation of the algorithm:

1. **First step:** the intersections come in pairs – one point that will be the exit from a charging area and another coincident point that’ll be the entry for the neighbour area. The
points won’t be an exact match because the neighbouring charging area borders, as given by the .are file, don’t coincide either, like figure 4.5 demonstrates.

Figure 4.5: Detail image of the airspace delimited by the coordinates of the LE (Spain) and GM (Morocco) areas, as given by the .are file. The limiting boundaries don’t exactly match – in this case, they overlap.

For each charging zone, the number of intersection points is calculated and saved on a vector. When analysing the orthodromic distance between any two points, this trajectory might intersect a specific area several times, because it is a direct line that doesn’t take into account the entry and exit points of the charging zone. This means that, if the line passes through an irregular frontier, then it will enter and exit the charging zone several times and have multiple intersection points.

Figure 4.6: The orthodromic line between waypoints VASTO and DELOG cross the LP frontier (right polyhedron of the image) several times.
Given these two conditions, a charging zone that has an even number of intersection points is an intermediate zone, and an odd number corresponds to an initial or ending area.

2. **Second step:** the distance is calculated for each point. These distances will also come in pairs. It also determines the minimum and maximum distance that should correspond to the first and last intersection points and, therefore, identify the first and last area.

3. **Third step:** because the distance comes in pairs, only one is selected for the CRCO charges calculations. The distances to the exit points is chosen – for each pair of intersection points in a charging area, ordered according the trajectory direction, the maximum distance of those points to the departure point is chosen. In the case of initial and final charging areas, if any of the bordering points is an airport, then the relevant area is found through the two first letters of the ICAO code of those airports. In any other case, in order to know which is the first or the last area, the maximum and minimum distances of these intersection points are calculated. Because the frontiers of the charging zones as given by the database occasionally overlap (fig.4.5), the maximum distance can be from the penultimate area instead of the last; therefore, a count of the number of points of intersection is kept so that it is the maximum distance of the charging zone with an odd number of points.

**CRCO charges**

The travelled distance for each charging area is the difference between the trajectory length to the exit point, and the length to the exit point of the previous area. Taking the example from figure 4.4, the distance travelled in the LF area corresponds to d2-d1.

If the first or last point is an airport, to the distance from the neighbour waypoint to the first/last point a total of 20 km is discounted in order to comply with the CRCO rule of deducting that distance in each take-off/landing charging zone.

The program selects the Unit Rates for each area and calculates the value for each distance. If the user inputted an aircraft model that exists in the database, the weight factor is calculated according to the following formula (23) described in section 2.3.3; otherwise, a weight factor of one is considered.

The total value of the charges \( C_T \) is given by the following formula:

\[
C_T = \sum_{i=2}^{M} (d_i - d_{i-1}) \times U_i \times W \tag{23}
\]

Where M is the total number of crossed charging areas, d is the distance from the starting point of the route to the exit point of the charging area i, \( U \) is the Unit Rate for that area and \( W \) is the aircraft’s weight factor.

**4.2.4 Planned and Real Flights**

The cycle that works through the top five and ten most frequently planned and real routes, respectively, is explained in this section.

The flights that contribute to the count of number of flights in the five presented planned routes
might not necessarily be the exact same for the real routes, i.e., the universe of flight IDs of the planned flights gathered by the tool won’t be the same as those from the real flights universe. This is intentional because the purpose is to collect the most effectively used route. To restrict the real routes to the planned flight universe wouldn’t be an actual portrayal of the use of the airspace. Also, real routes between the same two points differ more than planned routes because airliners tend to reuse the same flight plans for the same destinations, however during the course of the actual flight factors like weather, delays, capacity management or just convenience can deviate the aircraft slightly from the planned trajectory. Because there is a wider distribution of real routes, the tool selects the top ten instead of only the top five, so that the total number of real flights doesn’t differ a lot from the planned.

Intersections

Each route is a vector item that contains the navigation points, separated by commas. The waypoints of the route need to be extracted in order to analyse the route, so a cycle that allocates waypoints to a vector item is run for each route.

The next step consists of a cycle that goes through each of the selected routes. For each route, another inner cycle is run that goes through each route segment. Each route segment is the orthodromic route between two sequential route waypoints, so the algorithm to find the intersection points is the same as that described in the previous section (Pseudocode 1).

Pseudocode 3: selected routes intersection points

| while n <= size(selected_routes) do |
| while m <= size(route_waypoints) do |
|   first_point=route_waypoints[m] |
|   second_point=route_waypoints[m+1] |
|      run Pseudocode 1: intersection points |
|   m=m+2 |
| end |
| n=n+1 |
| end |

The problem of finding which charging zone the first and last point belong to, in order to calculate the appropriate charges, requires a solving method that’s a variation of the one used in the orthodromic route. The program still makes use of the point-in-polygon ray-casting algorithm, but a few previous steps need to be taken into consideration:

- the program only finds intersections from route segment to route segment. In order to determine if there is an even or odd number of intersections in any charging area, all the intersections for the entire route need to be assessed first, like figure 4.7 indicates:
Figure 4.7: Green line representing the planned route from LEMD to GCLP. The first route segment that contains an intersection, described by the yellow dots OBIPA-ELVAR (ELVAR is located on the frontier), detects a crossing point for both the LP (Portugal) area and the LE (Spain) area. The first point in the route is represented by the blue dot. In order for the tool to know to which of the areas the first point belongs to, there’s the need to know all the intersections, because the one represented by the red dot on the lower left of the image will prove that the LP area has an even number of intersections.

- there will be two charging zones with an odd number of crossing points. The route is analysed by consecutive order of its navigation points, so an assumption could be made that the first area for which an odd number of intersections is found is the one that contains the initial point. However, in planned and real flights, where there can be use of the Free Route Airspace, some segments will be defined only by the entry and exit points of one charging zone, like figures 4.7 and 4.8 demonstrate:

Figure 4.8: Route from and BAROK (lower yellow dot) to CCS (blue dot). The LP area is crossed and only the entry and exit points are registered (yellow dots, ELVAR and BAROK navigation points). The BAROK to ELVAR segment will detect 4 intersections: one for the LE area (right of the image) two for the LP area (left of the image) and one for GM area (bottom of the image). In this particular route, the same segment detects all the intersections.
In figure’s 4.8 specific case, the first flight segment, BAROK-ELVAR, finds 4 intersection points, (because the frontiers are either coincidental or overlap): one for the GM area, two for the LP area and one other for the LE area. In this case, the program doesn’t have a way to detect which comes first. So the orthodromic distance of each odd numbered area intersection to the initial point is calculated: this doesn’t represent the actual flown length, but the tool makes the assumption that the smallest distance corresponds to the first intersection point, and the largest to the last.

In the case of a flight between only two different charging zones, there will be only one crossing point, coincident for both of the areas, and only one segment of the entire route that crosses the frontiers. In this case, a function find_fir is used: this function receives the coordinates of the first point of the segment, and casts a ray that most likely will end outside the area by adding ten degrees to the latitude and twenty-five to the longitude. Then it runs the algorithm of detecting all the charging zones that are crossed and counts the number of intersections – the one with the odd number and the closest intersection point is the initial area. The same function is used for the ending point of the segment to know the ending charging zone.

**Distance**

To calculate the CRCO charges, the considered distance is the orthodromic length between the entry and exit point of the charging zone, according to the flight plan, like the CRCO indicates [16]. These points equate to the area intersections. It does not correspond to the sum of the route segments lengths within the same area because these segments won’t describe a direct route. For the planned flights, there will be only one entry and one exit point for each charging area.

However, this tool intends to replicate the charges calculations for the real route to be able to compare them with the planned and evaluate the efficiency of the actual flown routes. Real trajectories may have small detours, and the path registered may cross a charging zone frontier several times, like fig.4.9 exemplifies:

![Figure 4.9: Detail of real flight between VASTO and DELOG navigation points, where the aircraft overflew the frontier between the LF (left) and LE (right) area several times.](image-url)
To be able to calculate and use a distance for the CRCO charges calculation, only the first and the last of the total crossing points for a charging area are considered. This means that there will be a part of the route length that's considered twice, once for each neighbour area – the analysis of both the statistics data file and the visual Google Earth file is necessary in order to evaluate if the CRCO hypothetical cost for the real route is comparable to the planned routes.

### 4.3. Output Files

The Flight Planning Support Tool creates two main output files as the solution to the user inputs. It also outputs a .kml file with charging zone coordinates that allows for the representation of these areas in the Google Earth application, like figure 4.10 shows:

![Figure 4.10: FIRS.kml file open in Google Earth – it contains a 3D representation of the charging areas.](image)

This graphical representation mimics the one available in the NEST Tool and it is key for validating intersection points and CRCO charges calculations. It also allows for a better visualization of the trajectories chosen. This file is generated with each execution of the program, so if there is an alteration to the database of the charging zone frontiers, it is immediately replicated within the next user execution.

#### 4.3.1 route_statistics.csv

This file is defined by a header, a section with the orthodromic detailed information, and a section with the planned and real flight information grouped by route.

The header contains the user input points, the weight factor calculated based on the information of
aircraft model, and the most recent date for which the information in the document refers to - that is, the flight information used for this file dates from the previous five AIRAC cycles up to the displayed date. This date is also the most recent date present in the database.

The orthodromic section first displays the total distance of the route and the total hypothetical charges. Then a breakdown of these values is given by charging zone sector, in the order which it is crossed. The coordinates of the intersection points are also displayed – they represent the crossing of the inline charging area with the area displayed in the next line, except for the last, where the charging area displays the intersection point with the previous one – because of this, the last two points are coincident.

The distance corresponds to the length crossed in the specified sector, and well as the CRCO cost. The Unit Rate used is also displayed.

![Figure 4. 11: The two first sections of the route_statistics.csv file from the LFPO-SAMAR trajectory, imported into Microsoft Excel. Up to the 6th line there is the header, from the 7th onward is the orthodromic route information, with the summary information in line 8 and the detail below the 9th line.](image)

The next section presents the results for the planned flights. The first column is the total number of flights (“No. Flights”) planned with the specified route during the most recent five AIRAC cycles in the database. The total distance (“Average Length (km)”), based on the flight information given, is the length flown between the chosen points. In the case of an airport-to-airport route, the distance will be equal to the total flight length. If any or both of the limit points is a waypoint, the distance presented will be only for that segment of flight. The distances are retrieved directly from the database – they are not the sum the orthodromic distance between entry and exit charging zone points, like the ones used for the cost calculation, but the actual distance flown by the aircraft. Because the tool is retrieving information from several flights and grouping them, the displayed distance is the average of all the registered distances that use that corresponding route.

The next column is the total cost (“CRCO Cost”) as calculated by the cycle explained in the previous section of this thesis. The last column (“Route (waypoints)”) is the actual route, described by waypoint – each point occupies its own cell. If routes differ by one waypoint, they are considered different routes, even though the distance and the cost might be the same. If any of the different
waypoints don’t correspond to a charging area boundary framing waypoint, the intersection points will be exactly the same and so will the total cost.

The ten most used routes in real flights are presented next, with the same information as the planned flights.

<table>
<thead>
<tr>
<th>No. Flights</th>
<th>Average Length (km)</th>
<th>CRCO Cost</th>
<th>Route waypoints</th>
</tr>
</thead>
<tbody>
<tr>
<td>180</td>
<td>1895.07</td>
<td>118.75</td>
<td>SSY BOS VTB BOGS ORCA HI STL CLA MA KORKE</td>
</tr>
<tr>
<td>181</td>
<td>1875.67</td>
<td>118.15</td>
<td>SSY AVLA LONGA CCS ELVAR BAROK BENU ABIR MITLA</td>
</tr>
<tr>
<td>128</td>
<td>1875.47</td>
<td>118.15</td>
<td>SSY AVLA LONGA ORPA ELVAR BAROK BENU ABIR MITLA</td>
</tr>
<tr>
<td>148</td>
<td>1866.05</td>
<td>116.75</td>
<td>POT VTB BOGS ORCA HI STL CLA MA KORKE ABDA</td>
</tr>
<tr>
<td>135</td>
<td>1846.57</td>
<td>116.75</td>
<td>VTB BOGS ORCA HI STL CLA MA KORKE ABOBA BNLU</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No. Flights</th>
<th>Average Length (km)</th>
<th>CRCO Cost</th>
<th>Route waypoints</th>
</tr>
</thead>
<tbody>
<tr>
<td>44</td>
<td>1866.13</td>
<td>107.07</td>
<td>SSY AVLA LONGA ORPA BAROK BENU ABIR MITLA SAMAR</td>
</tr>
<tr>
<td>39</td>
<td>1882.71</td>
<td>106.13</td>
<td>SSY AVLA LONGA CCS ELVAR BAROK BENU ABIR MITLA</td>
</tr>
<tr>
<td>34</td>
<td>1878.51</td>
<td>104.15</td>
<td>SSY AVLA LONGA ORPA ELVAR BAROK BENU ABIR MITLA</td>
</tr>
<tr>
<td>31</td>
<td>1877.03</td>
<td>107.07</td>
<td>SSY AVLA LONGA CCS BAROK BENU ABIR MITLA SAMAR</td>
</tr>
<tr>
<td>19</td>
<td>1866.19</td>
<td>107.07</td>
<td>SSY AVLA LONGA CCS BAROK BENU ABIR MITLA SAMAR</td>
</tr>
<tr>
<td>17</td>
<td>1862.18</td>
<td>107.07</td>
<td>SSY CCS BAROK BENU ABIR MITLA SAMAR ISORU IDNV</td>
</tr>
<tr>
<td>17</td>
<td>1876.86</td>
<td>104.13</td>
<td>SSY AVLA LONGA CCS ELVAR BAROK BENU ABIR MITLA</td>
</tr>
<tr>
<td>17</td>
<td>1876.79</td>
<td>104.15</td>
<td>SSY AVLA LONGA CCS ELVAR BAROK BENU ABIR MITLA</td>
</tr>
<tr>
<td>14</td>
<td>1861.07</td>
<td>116.75</td>
<td>VTB BOGS ORCA HI STL CLA MA KORKE ABOBA BAULU</td>
</tr>
<tr>
<td>14</td>
<td>1862.77</td>
<td>107.07</td>
<td>SSY CCS BAROK BENU ABIR MITLA SAMAR ISORU TSORD</td>
</tr>
</tbody>
</table>

Figure 4. 12: Planned and real flight statistics for a LEMD-GCLP flight.

If the limit points chosen by the user are not both airports, another section with the same route information for the planned and real trajectories but now grouped by departing and arrival airports is presented. This allows the user to know what are the main destinations or departure sites that make use of the inputted points, and if said points are selected for several different flights, more spread out through the airspace, or are only used for the same trajectories.

<table>
<thead>
<tr>
<th>Departure</th>
<th>Arrival</th>
<th>No. Flights</th>
<th>Average Length (km)</th>
<th>CRCO Cost</th>
<th>Route waypoints</th>
</tr>
</thead>
<tbody>
<tr>
<td>LFPO</td>
<td>GDBA</td>
<td>7</td>
<td>2518.78</td>
<td>LGL</td>
<td>REDN</td>
</tr>
<tr>
<td>LFPO</td>
<td>GCLP</td>
<td>5</td>
<td>2678.36</td>
<td>LGL</td>
<td>REDN</td>
</tr>
<tr>
<td>LFPO</td>
<td>GCLP</td>
<td>5</td>
<td>2684.66</td>
<td>LGL</td>
<td>REDN</td>
</tr>
<tr>
<td>LFPO</td>
<td>GCLP</td>
<td>1</td>
<td>2700.86</td>
<td>ABORO</td>
<td>ASADA</td>
</tr>
<tr>
<td>LFPO</td>
<td>GCLP</td>
<td>3</td>
<td>2622.28</td>
<td>ASADA</td>
<td>AGOPA</td>
</tr>
<tr>
<td>LFPO</td>
<td>GCLP</td>
<td>3</td>
<td>2608.95</td>
<td>ASADA</td>
<td>AGOPA</td>
</tr>
<tr>
<td>LFPO</td>
<td>GCLP</td>
<td>2</td>
<td>2679.17</td>
<td>LGL</td>
<td>REDN</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Departure</th>
<th>Arrival</th>
<th>No. Flights</th>
<th>Average Length (km)</th>
<th>CRCO Cost</th>
<th>Route waypoints</th>
</tr>
</thead>
<tbody>
<tr>
<td>LFPO</td>
<td>GDBA</td>
<td>7</td>
<td>2518.78</td>
<td>LGL</td>
<td>REDN</td>
</tr>
<tr>
<td>LFPO</td>
<td>GCLP</td>
<td>5</td>
<td>2678.36</td>
<td>LGL</td>
<td>REDN</td>
</tr>
<tr>
<td>LFPO</td>
<td>GCLP</td>
<td>5</td>
<td>2684.66</td>
<td>LGL</td>
<td>REDN</td>
</tr>
<tr>
<td>LFPO</td>
<td>GCLP</td>
<td>1</td>
<td>2700.28</td>
<td>ABORO</td>
<td>ASADA</td>
</tr>
<tr>
<td>LFPO</td>
<td>GCLP</td>
<td>3</td>
<td>2622.28</td>
<td>ASADA</td>
<td>AGOPA</td>
</tr>
<tr>
<td>LFPO</td>
<td>GCLP</td>
<td>3</td>
<td>2608.95</td>
<td>ASADA</td>
<td>AGOPA</td>
</tr>
<tr>
<td>LFPO</td>
<td>GCLP</td>
<td>2</td>
<td>2679.17</td>
<td>LGL</td>
<td>REDN</td>
</tr>
</tbody>
</table>

Figure 4. 13: Extra section with routes grouped by airport, from a LFPO to SAMAR route. By the “Arrival” column we can identify the main destinations for flights that leave the LFPO airport and go through the SAMAR navigation point.

4.3.2 trajectory.kml

The Flight Planning Support Tool generates a .kml file where the geographical information of the analysed routes is written and can be imported into the Google Earth application.
This file contains:
- the orthodromic path, represented by a red line;
- the planned flights paths, represented in green; there are five paths, ordered from 1 to 5, and they are sorted by most frequent route, that is, they match the order in which they are presented in the route_statistics.csv file;
- the real flights, shown by blue lines, and these are also numbered by the same order in which they are presented in the .csv file;
- the individuals flight that satisfy the route conditions and over which the statistics are made, in yellow lines.

The Google Earth software allows the multiple selection of the paths, so the user can choose to see all the types at the same time or filter according to the analysis needs.

Figure 4.14: Route between VASTO and DELOG navigation points, with the direct route in red, the planned in green and the real in blue. Some of the planned paths and real paths coincide: in this case, the blue path overlaps.

Figure 4.15: Individuals flights from VASTO to DELOG represent in Google Earth. Left: the same route (VASTO-DELOG) as the previous image, but with the individual flights (yellow lines) also selected. This option allows the user to assess which actual flights make use of that part of the route. Right: part of the list of individual flights selected on Google Earth.

The individual flights are identified by a tag that contains an indication of if it is a planned or actual flight, like the right side of figure 4.15 indicates, and the flight ID.
4.4 Flight Planning Support Tool Limitations

A few aspects resulting of the Flight Planning Support Tool implementation need to be taken into account when using the tool and analysing its results.

4.4.1 Limit point in-between charging zone frontiers

For charging zone boundaries that don’t exactly coincide, and intersect originating a common area, if a limit point is inside that common space, the CRCO charges won’t be properly calculated. As exemplified in the image below, that limit point will detect an odd number of intersections for more than two charging zones.

![Diagram showing overlapping charging zones](image)

**Figure 4.16**: Overlapping frontiers: the shaded blue zone is the common area. The initial point (red dot), instead of detecting one intersection point for zone 1 and two for zone 2, will detect only one for each, and the algorithm will not be able to know which of the zones is the starting area.

If this point is the initial or ending point of the route, neither the orthodromic nor the actual flights distance and CRCO costs will be properly calculated.

If it is a waypoint contained in a planned or real route, only that route will not have the correct value for the CRCO charges. This happens because for the retrieved database routes, the path is analysed segment by segment, and each segment is treated like an orthodromic route between waypoints, so inevitably any intermediate point will be a starting and an ending point at a given iteration in the cycle. The route cycle might not be able to calculate the crossed distance in the extra odd-numbered intersection area, and it can also be unable to identify the limit charging areas.

4.4.2 Cross distance for real routes that re-enter the same charging zone

Because real routes will have deviations from the planned path, for various reasons like weather, sector capacity or delays, the actual registered real flight may re-enter the same charging zone multiple times, instead of just flying over the entry and exit planned points. In this case, the total distance computed won’t be the sum of the segments the aircraft crossed on a specific zone, but the
direct route between the crossing point that’s the closest to the initial route point, and the crossing point that is the farthest point.

Figure 4. 17: representation of a real route that re-enters the same charging zone two times, the actual distances flown (blue lines) and the distance considered (orange line).

. In figure 4.17, the real route is indicated by the light grey line, and the actual flown segments of the represented area are marked in blue. However, the program determines the minimum distance of the intersection point to the departure point (D1 – initial point to P1), and registers that point. Then, the maximum distance (D2 – initial point to P6) is determined, to find the second point for which to calculate the crossed length, and the total distance considered is the orthodromic distance between the two points, and it is represented by the orange line (P1 to P6).

This is done because since the real routes are analyzed segment by segment, if the limit waypoint of a segment describes a line that crosses the frontier multiple times, the order of those intersections is unknown. In the example of figure 4.17, for the segment that includes points P3 to P6, the cycle won’t be able to sort which of the four points is the first to occur in the flight, the second, and so forth. To solve this problem, the decision to assume that the closest to the starting point of the route is the area entry point, and the farthest, the exit point, was made. For these situations, the distances and, therefore, the CRCO charges, won’t correspond to the exact values.
5. Results

In this chapter, four case studies will be analysed: airport to airport, airport to waypoint, waypoint to airport and waypoint to waypoint. The results will be compared against real data for validation. A cost comparison between different routes for the same flight is performed in the end of this section.

A representation of the charging zones and their relevant labels for the interpretation of the case studies is shown in figure 5.1 for reference:

![Figure 5.1: Charging zone frontiers and labels.](image)

5.1 Case Study Airport-Airport: LEMD-GCLP

This case study gathers information of flights between the airport of Madrid-Barajas (LEMD) and the Gran Canaria airport (GCLP).

The airplane model considered for this flight was an Airbus A320.

The route statistics output file returns the following information for the direct route:

<table>
<thead>
<tr>
<th>Orthodromic Distance (km)</th>
<th>CRCO Cost (€)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1764.81</td>
<td>1123.21</td>
</tr>
</tbody>
</table>
For the database collected flights, the top most frequently used routes have the following characteristics:

Table 5. 2: Planned flight statistics for LEMD-GCLP

<table>
<thead>
<tr>
<th>Planned Route</th>
<th>No. Flights</th>
<th>Average Length (km)</th>
<th>CRCO Cost (€)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>189</td>
<td>1899,67</td>
<td>1168,75</td>
</tr>
<tr>
<td>2</td>
<td>181</td>
<td>1873,47</td>
<td>1084,15</td>
</tr>
<tr>
<td>3</td>
<td>156</td>
<td>1873,47</td>
<td>1084,15</td>
</tr>
<tr>
<td>4</td>
<td>148</td>
<td>1898,65</td>
<td>1168,75</td>
</tr>
<tr>
<td>5</td>
<td>135</td>
<td>1846,57</td>
<td>1168,75</td>
</tr>
</tbody>
</table>

Table 5. 3: Real flight statistics for LEMD-GCLP

<table>
<thead>
<tr>
<th>Real Route</th>
<th>No. Flights</th>
<th>Average Length (km)</th>
<th>CRCO Cost (€)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>44</td>
<td>1868,12</td>
<td>1073,07</td>
</tr>
<tr>
<td>2</td>
<td>35</td>
<td>1882,71</td>
<td>1084,15</td>
</tr>
<tr>
<td>3</td>
<td>34</td>
<td>1878,51</td>
<td>1084,15</td>
</tr>
<tr>
<td>4</td>
<td>31</td>
<td>1870,03</td>
<td>1073,07</td>
</tr>
<tr>
<td>5</td>
<td>19</td>
<td>1868,19</td>
<td>1073,07</td>
</tr>
<tr>
<td>6</td>
<td>17</td>
<td>1879,86</td>
<td>1084,15</td>
</tr>
<tr>
<td>7</td>
<td>17</td>
<td>1874,79</td>
<td>1084,15</td>
</tr>
<tr>
<td>8</td>
<td>17</td>
<td>1862,18</td>
<td>1073,07</td>
</tr>
<tr>
<td>9</td>
<td>14</td>
<td>1862,57</td>
<td>1073,07</td>
</tr>
<tr>
<td>10</td>
<td>14</td>
<td>1861,07</td>
<td>1168,75</td>
</tr>
</tbody>
</table>

By the analysis of the statistics, there are mainly two routes used in flight plans between the LEMD and GCLP airports. The CRCO Cost columns for the planned flights indicates that the orthodromic distance crossed in each charging zone is the same for routes 1, 4 and 5, and the same is true between routes 2 and 3.

To support this conclusion, the trajectories can be observed on Google Earth. As shown by figure 5.2, where all the most used trajectories are selected, only two main lines are present between the airports:
Figure 5.2: LEMD-GCLP trajectories in Google Earth. Left: the planned trajectories for the LEMD-GCLP flight. There are mainly two routes, the top line corresponds to route 2 and 3 of table 5.2, the lower to routes 1, 4 and 5. The routes overlap: the differing waypoints correspond either to different points in the take-off or approach area, or to collinear points that don’t alter the trajectory. Right: the real trajectories for the LEMD-GCLP flight. The route on the bottom right corresponds to route 10, the others above to 1-9. There is more variation in the routes, as expected, but two main tendencies call still be identified.

Figure 5.2 shows that of the two predominant trajectories there is a major difference of crossed charging zones: one crosses the LP sector, the other doesn’t. The trajectories that cross the LP area have a lesser CRCO cost – even less than the orthodromic route. This is caused by the fact that the portion of the orthodromic route that crosses the LP area is smaller, like figure 5.3 demonstrates:

Figure 5.3: Orthodromic route for LEMD-GCLP. The pins mark the intersections.

The detail of each cost per charging zone can be obtained by the orthodromic route detail displayed in the file. The sum of the distances crossed by area will not equal the total length because 20 km are removed from the landing and take-off zone in order to comply with the CRCO method of calculus for the charges.
Table 5.4: Orthodromic route detail for LEMD-GCLP

<table>
<thead>
<tr>
<th>Charging Zone</th>
<th>Distance (km)</th>
<th>CRCO Cost (€)</th>
<th>CRCO Rate (€*100)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LE</td>
<td>495,098</td>
<td>440,894</td>
<td>7176</td>
</tr>
<tr>
<td>LP</td>
<td>137,888</td>
<td>68,771</td>
<td>4019</td>
</tr>
<tr>
<td>GM</td>
<td>773,839</td>
<td>382,971</td>
<td>3988</td>
</tr>
<tr>
<td>GC</td>
<td>317,984</td>
<td>230,569</td>
<td>5843</td>
</tr>
</tbody>
</table>

A comparative analysis of the length and cost of the database routes versus the orthodromic route is presented in tables 5.5 and 5.6.

Table 5.5: Planned flights vs. orthodrome for LEMD-GCLP

<table>
<thead>
<tr>
<th>Planned Route</th>
<th>Average Length</th>
<th>CRCO Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7,64%</td>
<td>4,05%</td>
</tr>
<tr>
<td>2</td>
<td>6,16%</td>
<td>-3,48%</td>
</tr>
<tr>
<td>3</td>
<td>6,16%</td>
<td>-3,48%</td>
</tr>
<tr>
<td>4</td>
<td>7,58%</td>
<td>4,05%</td>
</tr>
<tr>
<td>5</td>
<td>4,63%</td>
<td>4,05%</td>
</tr>
</tbody>
</table>

Table 5.6: Real flights vs. orthodrome for LEMD-GCLP

<table>
<thead>
<tr>
<th>Real Route</th>
<th>Average Length</th>
<th>CRCO Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5,85%</td>
<td>-4,46%</td>
</tr>
<tr>
<td>2</td>
<td>6,68%</td>
<td>-3,48%</td>
</tr>
<tr>
<td>3</td>
<td>6,44%</td>
<td>-3,48%</td>
</tr>
<tr>
<td>4</td>
<td>5,96%</td>
<td>-4,46%</td>
</tr>
<tr>
<td>5</td>
<td>5,86%</td>
<td>-4,46%</td>
</tr>
<tr>
<td>6</td>
<td>6,52%</td>
<td>-3,48%</td>
</tr>
<tr>
<td>7</td>
<td>6,23%</td>
<td>-3,48%</td>
</tr>
<tr>
<td>8</td>
<td>5,52%</td>
<td>-4,46%</td>
</tr>
<tr>
<td>9</td>
<td>5,54%</td>
<td>-4,46%</td>
</tr>
<tr>
<td>10</td>
<td>5,45%</td>
<td>4,05%</td>
</tr>
</tbody>
</table>

From the data collected we can assess that the length for any database route is at least 4.63% higher than the orthodromic route, but for the cost the same doesn't happen, so this could indicate, depending on the cost per km of a particular flight, that the direct route may not be the most cost efficient for airliners.

### 5.2 Case Study Airport-Waypoint: LFPO-SAMAR

This case study gathers information of flights between the airport of Paris-Orly (LFPO) and an entry point for the GC area (SAMAR).

The airplane model considered was an Airbus A320.
The route statistics output file displays the following summary information for the direct route:

Table 5. 7: Direct route information for LFPO-SAMAR

<table>
<thead>
<tr>
<th>Orthodromic Distance (km)</th>
<th>CRCO Cost (€)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2434,8</td>
<td>1616,26</td>
</tr>
</tbody>
</table>

The breakdown of the distance and cost per charging zone is given by the following table:

Table 5. 8: Orthodromic route detail for LFPO-SAMAR

<table>
<thead>
<tr>
<th>Charging Zone</th>
<th>Distance (km)</th>
<th>CRCO Cost (€)</th>
<th>CRCO Rate (€*100)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LF</td>
<td>651,139</td>
<td>541,954</td>
<td>6707</td>
</tr>
<tr>
<td>LE</td>
<td>503,446</td>
<td>448,328</td>
<td>7176</td>
</tr>
<tr>
<td>LP</td>
<td>597,571</td>
<td>298,036</td>
<td>4019</td>
</tr>
<tr>
<td>GM</td>
<td>662,646</td>
<td>327,942</td>
<td>3988</td>
</tr>
</tbody>
</table>

The highest Unit Rate for this route is for the LE area, the lowest for the GM zone.

The most used routes have the following characteristics:

Table 5. 9: Planned flight statistics for LFPO-SAMAR

<table>
<thead>
<tr>
<th>Planned Route</th>
<th>No. Flights</th>
<th>Average Length (km)</th>
<th>CRCO Cost (€)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>2554,76</td>
<td>1579,07</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>2658,24</td>
<td>1573,25</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>2568</td>
<td>1579,07</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>2678,36</td>
<td>1579,06</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>2700,88</td>
<td>1686,38</td>
</tr>
</tbody>
</table>

Table 5. 10: Real flight statistics for LFPO-SAMAR

<table>
<thead>
<tr>
<th>Real Route</th>
<th>No. Flights</th>
<th>Average Length (km)</th>
<th>CRCO Cost (€)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>2602,77</td>
<td>1579,07</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>2532,35</td>
<td>1573,25</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2525,13</td>
<td>1579,06</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2557,4</td>
<td>1579,07</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2592,66</td>
<td>1573,25</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>2614,47</td>
<td>1669,4</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>2538,57</td>
<td>1579,07</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>2519,95</td>
<td>1579,07</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>2516,82</td>
<td>1686,38</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>2518,85</td>
<td>1686,37</td>
</tr>
</tbody>
</table>

For all the planned routes apart from the fifth, the CRCO charges are inferior to the direct route; when it comes to the distance the fifth – and least chosen – route is also the least cost efficient, being
at least 22.52 km longer than any other planned route.

Observing the left side of figure 5.4, where the planned routes are represented and labelled, it is visible that the fifth route has a longer route segment crossing the LE sector compared to the other routes, and a shorter segment crossing LP as a consequence, which has a lower unit rate – so the higher cost can be justified not just by the longer distance but also due to a longer path on a higher Unit Rate area.

![Figure 5.4: LFPO-SAMAR trajectories in Google Earth. Left: the planned trajectories for the LFPO-SAMAR flight; routes 1, 3 and 4 coincide. Right: the real trajectories for the LFPO-SAMAR flight. The trajectories are more diverse for the real flights.](image)

Of the real routes, the 9th and 10th have the highest CRCO cost, approximately the same of the 5th planned route, although the average distance is inferior. In figure 5.5 there is a representation of the ninth real route (in blue, labelled “9”) and the fifth planned route (in green). The routes overlap for most of the trajectory, and it is possible to verify that the crossing points of the charging zones are the same (so the total CRCO charges have to be the same), but closer to the LFPO point the planned route describes a longer path.

Figure 5.5 displays three routes, that correspond to an ascending order of CRCO cost: the second real route (labelled “2”), the orthodromic route, and the most expensive real route (ninth route). It is possible to observe that the length of the segment that crosses the LE area is also in ascending order, which is a possible explanation for the cost distribution.
The following table display a comparative analysis of the planned routes with the orthodromic.

Table 5. 11: Planned flights vs. orthodrome for LFPO-SAMAR

<table>
<thead>
<tr>
<th>Planned Route</th>
<th>Average Length</th>
<th>CRCO Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4,93%</td>
<td>-2,30%</td>
</tr>
<tr>
<td>2</td>
<td>9,18%</td>
<td>-2,66%</td>
</tr>
<tr>
<td>3</td>
<td>5,47%</td>
<td>-2,30%</td>
</tr>
<tr>
<td>4</td>
<td>10,00%</td>
<td>-2,30%</td>
</tr>
<tr>
<td>5</td>
<td>10,93%</td>
<td>4,34%</td>
</tr>
</tbody>
</table>

The most planned route is the closest to the direct route in terms of distance, but the second has minimum cost. Which of these is most cost efficient depends on the varying cost per kilometre, so it could be different for different airplane models or airliners.

The equivalent comparative analysis for the real routes is presented in the table below. The majority of real routes have an average length closer to the orthodrome, which means that in the actual flight the trajectory tends to be more direct. However, the minimum theoretical CRCO cost is not lower than the planned routes – a possible explanation for this, supported by the previous figure, is that the entry/exit points for the charging zones are the same, even though the in-sector points may not be.
Table 5.12: Real flights vs. orthodrome for LFPO-SAMAR

<table>
<thead>
<tr>
<th>Real Route</th>
<th>Average Length</th>
<th>CRCO Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6,90%</td>
<td>-2,30%</td>
</tr>
<tr>
<td>2</td>
<td>4,01%</td>
<td>-2,66%</td>
</tr>
<tr>
<td>3</td>
<td>3,71%</td>
<td>-2,30%</td>
</tr>
<tr>
<td>4</td>
<td>5,04%</td>
<td>-2,30%</td>
</tr>
<tr>
<td>5</td>
<td>6,48%</td>
<td>-2,66%</td>
</tr>
<tr>
<td>6</td>
<td>7,38%</td>
<td>3,29%</td>
</tr>
<tr>
<td>7</td>
<td>4,26%</td>
<td>-2,30%</td>
</tr>
<tr>
<td>8</td>
<td>3,50%</td>
<td>-2,30%</td>
</tr>
<tr>
<td>9</td>
<td>3,37%</td>
<td>4,34%</td>
</tr>
<tr>
<td>10</td>
<td>3,45%</td>
<td>4,34%</td>
</tr>
</tbody>
</table>

The last section of the statistics output file regroups the most used routes by airport destination. The most frequent arrival airports are GCLP (Gran Canaria airport), GVBA (Aristides Pereira airport, in Cape Verde) and GOOY (Yoff-Léopold Sédar Senghor airport, in Senegal). Figure 5.6 below displays some of the entire real flights that were the base of information of the produced file:

Figure 5.6: Actual individual flights, complete (airport-to-airport), represented in yellow.
5.3 Case Study Waypoint-Airport: ECKOS-EGSS

This case study considers routes between the ECKOS navigation point, that is close to the Gran Canaria airport, and the Stansted airport, in London (EGSS). The aircraft model considered was an A320. The orthodromic route summary and detail is displayed in the two tables below:

Table 5. 13: Direct route information for ECKOS-EGSS

<table>
<thead>
<tr>
<th>Orthodromic Distance (km)</th>
<th>CRCO Cost (€)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2950,89</td>
<td>1981,08</td>
</tr>
</tbody>
</table>

Table 5. 14: Orthodromic route detail for ECKOS-EGSS

<table>
<thead>
<tr>
<th>Charging Zone</th>
<th>Distance (km)</th>
<th>CRCO Cost (€)</th>
<th>CRCO Rate (€*100)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GC</td>
<td>324,23</td>
<td>235,099</td>
<td>5843</td>
</tr>
<tr>
<td>GM</td>
<td>633,755</td>
<td>313,644</td>
<td>3988</td>
</tr>
<tr>
<td>LP</td>
<td>721,28</td>
<td>359,735</td>
<td>4019</td>
</tr>
<tr>
<td>LE</td>
<td>349,717</td>
<td>311,429</td>
<td>7176</td>
</tr>
<tr>
<td>LF</td>
<td>676,872</td>
<td>563,372</td>
<td>6707</td>
</tr>
<tr>
<td>EG</td>
<td>225,036</td>
<td>197,801</td>
<td>7083</td>
</tr>
</tbody>
</table>

Of the planned routes, the fifth has the smallest CRCO charges cost and the shortest distance as well. It appears to be the most cost effective, however it is the least frequently chosen of the top five. This could be an indication of another type of inefficiency of this particular route: as it can be observed from the two left images of figure 5.7, the entry and exit points for the LE area are different from any other planned route, and they may be used in other routes that deplete the capacity for that segment.

Table 5. 15: Planned flights statistics for ECKOS-EGSS

<table>
<thead>
<tr>
<th>Planned Route</th>
<th>No. Flights</th>
<th>Average Length (km)</th>
<th>CRCO Cost (€)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td>3043,67</td>
<td>2098,82</td>
</tr>
<tr>
<td>2</td>
<td>17</td>
<td>3038,42</td>
<td>2033,6</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>3035,85</td>
<td>2044,72</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>3068,86</td>
<td>2068,56</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>3019,55</td>
<td>1975,98</td>
</tr>
</tbody>
</table>

Table 5. 16: Planned flights vs orthodrome for ECKOS-EGSS

<table>
<thead>
<tr>
<th>Planned Route</th>
<th>Average Length</th>
<th>CRCO Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3,14%</td>
<td>5,94%</td>
</tr>
<tr>
<td>2</td>
<td>2,97%</td>
<td>2,65%</td>
</tr>
<tr>
<td>3</td>
<td>2,88%</td>
<td>3,21%</td>
</tr>
<tr>
<td>4</td>
<td>4,00%</td>
<td>4,42%</td>
</tr>
<tr>
<td>5</td>
<td>2,33%</td>
<td>-0,26%</td>
</tr>
</tbody>
</table>
None of the most used real routes describe the same trajectory of the fifth planned flight, so this corroborates the fact that this is not an actually frequently used path, despite the apparent advantages. This type of statistics can direct the user to look for other variables made available by the NEST Tool, like flight delay.

Analyzing the table of statistics for the real routes, the ninth and tenth route stand out for their lower CRCO charges, even though both distances are at least %5 longer than any other real route. Comparing the routes in the Google Earth application, both of them contain a segment that falls out of the CRCO file charging zones, marked by the black arrow in figure 5.8. This means the cost for that segment is not being computed into the total, so these routes are not comparable to the other selected.

<table>
<thead>
<tr>
<th>Real Route</th>
<th>No. Flights</th>
<th>Average Length (km)</th>
<th>CRCO Cost (€)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>3018.47</td>
<td>2098.82</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>3006.02</td>
<td>2098.82</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3012.46</td>
<td>2115.24</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>3019.19</td>
<td>2121.57</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>3003.5</td>
<td>2135.5</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>3012.2</td>
<td>2135.5</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>2980.07</td>
<td>2098.82</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>3027.29</td>
<td>2098.82</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>3223.03</td>
<td>1788.76</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>3191.89</td>
<td>1765.51</td>
</tr>
</tbody>
</table>
Table 5. 18: Real flights vs orthodrome for ECKOS-EGSS

<table>
<thead>
<tr>
<th>Real Route</th>
<th>Average Length</th>
<th>CRCO Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2,29%</td>
<td>5,94%</td>
</tr>
<tr>
<td>2</td>
<td>1,87%</td>
<td>5,94%</td>
</tr>
<tr>
<td>3</td>
<td>2,09%</td>
<td>6,77%</td>
</tr>
<tr>
<td>4</td>
<td>2,31%</td>
<td>7,09%</td>
</tr>
<tr>
<td>5</td>
<td>1,78%</td>
<td>7,79%</td>
</tr>
<tr>
<td>6</td>
<td>2,08%</td>
<td>7,79%</td>
</tr>
<tr>
<td>7</td>
<td>0,99%</td>
<td>5,94%</td>
</tr>
<tr>
<td>8</td>
<td>2,59%</td>
<td>5,94%</td>
</tr>
<tr>
<td>9</td>
<td>9,22%</td>
<td>-9,71%</td>
</tr>
<tr>
<td>10</td>
<td>8,17%</td>
<td>-10,88%</td>
</tr>
</tbody>
</table>

Another noticeable fact in the real flight distribution is the presence of routes on the left side of the orthodromic line – for the planned flights, they are all on the right. Whether or not these left paths were planned as so, it is confirmed that the real flights describe a wider variety of routes and there are
several detours to the flight plan.
On this particular route, the departure airport is always GCLP, because ECKOS is a navigation point of the take-off area. Given so, the grouping of routes per airport is the same as the planned and real flight statistics table (tables 5.15 and 5.17).

5.4 Case Study Waypoint-Waypoint: SULAM-DELOG

This case study focuses on the routes between a waypoint in the Morocco charging zone (GM), SULAM, and an exit navigation point from the LE area, DELOG. The aircraft model considered for these routes was an Airbus A320.

The orthodromic route summary and detail is displayed in the two tables below:

<table>
<thead>
<tr>
<th>Charging Zone</th>
<th>Distance (km)</th>
<th>CRCO Cost (€)</th>
</tr>
</thead>
<tbody>
<tr>
<td>GM</td>
<td>628,785</td>
<td>311,185</td>
</tr>
<tr>
<td>LP</td>
<td>614,225</td>
<td>306,342</td>
</tr>
<tr>
<td>LE</td>
<td>62,0302</td>
<td>55,239</td>
</tr>
<tr>
<td>LP</td>
<td>8,36025</td>
<td>4,16963</td>
</tr>
<tr>
<td>LE</td>
<td>3,96227</td>
<td>3,52847</td>
</tr>
<tr>
<td>LP</td>
<td>26,2382</td>
<td>13,0862</td>
</tr>
<tr>
<td>LE</td>
<td>353,475</td>
<td>314,776</td>
</tr>
</tbody>
</table>

As one can see by table 5.20, the direct route between SULAM and DELOG crosses the LE and LP charging zones multiple times. The LP charging zone has six intersection points, that equivalent to three separate segments that cross this area. The same is true for the LE area, but only five intersection points are found because the last segment ends in the LE exit point. Figure 5.9 shows the detail of the multiple segments, with the LP distances highlighted by a parallel line of the same length; the parallel lines have the same colour of the corresponding row in table 5.20.
For the planned routes, only the first two more frequent are coincident, and they correspond to the closest route to the orthodromic line. Despite this fact, these are the longest routes in total length, because although they follow closely the orthodromic route in the LP area like it is possible to observe in figure 5.9, there is a detour in direction when crossing the frontier with the LE area that moves the line away from the orthodrome. The CRCO costs are the minimum of all the planned routes because this is the route with the longest direct crossing distance in the LP area and the shortest in the LE area.

The average distance of the planned routes however doesn’t vary much from route to route; the difference from the longest to the shortest is less than 30 km.

Table 5. 21: Planned flights statistics for SULAM-DELOG

<table>
<thead>
<tr>
<th>Planned Route</th>
<th>No. Flights</th>
<th>Average Length (km)</th>
<th>CRCO Cost (€)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>949</td>
<td>1877,79</td>
<td>1031,75</td>
</tr>
<tr>
<td>2</td>
<td>578</td>
<td>1879,19</td>
<td>1031,75</td>
</tr>
<tr>
<td>3</td>
<td>484</td>
<td>1872,9</td>
<td>1133,64</td>
</tr>
<tr>
<td>4</td>
<td>429</td>
<td>1849,46</td>
<td>1119,99</td>
</tr>
<tr>
<td>5</td>
<td>225</td>
<td>1860,85</td>
<td>1096,78</td>
</tr>
</tbody>
</table>
The real routes follow more or less the same paths as the planned ones. The range of distances is also very similar.
Table 5.22: Real flights statistics for SULAM-DELOG

<table>
<thead>
<tr>
<th>Real Route</th>
<th>No. Flights</th>
<th>Average Length (km)</th>
<th>CRCO Cost (€)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1573</td>
<td>1803.53</td>
<td>1133.64</td>
</tr>
<tr>
<td>2</td>
<td>412</td>
<td>1804.23</td>
<td>1119.99</td>
</tr>
<tr>
<td>3</td>
<td>193</td>
<td>1810.41</td>
<td>1113.4</td>
</tr>
<tr>
<td>4</td>
<td>148</td>
<td>1819.36</td>
<td>1096.78</td>
</tr>
<tr>
<td>5</td>
<td>147</td>
<td>1827.51</td>
<td>1113.39</td>
</tr>
<tr>
<td>6</td>
<td>144</td>
<td>1795.91</td>
<td>1113.39</td>
</tr>
<tr>
<td>7</td>
<td>89</td>
<td>1816.96</td>
<td>1119.99</td>
</tr>
<tr>
<td>8</td>
<td>81</td>
<td>1783.12</td>
<td>1133.64</td>
</tr>
<tr>
<td>9</td>
<td>76</td>
<td>1812.12</td>
<td>1119.99</td>
</tr>
<tr>
<td>10</td>
<td>66</td>
<td>1835.19</td>
<td>1113.4</td>
</tr>
</tbody>
</table>

The trajectory line that is closest to the orthodromic route, and consequently closer to the 1st and 2nd planned route, is the most frequent real route, route 1, that is represented in the blue line in figure 5.12. As it can be seen in the detail, the real route follows a more direct path, that shortens the total distance slightly. The closest and most distant routes from the orthodromic route are represented in figure 5.12; the farthest (number 4) is coincident with the equivalent planned route (number 5).

Figure 5.12: SULAM to DELOG trajectories represented in Google Earth. Left: orthodromic route in red, closest planned (green) and real route (blue) and farthest planned and real route (numbers 5 and 4). Right: detail of the routes.

Comparing the increase of distances of both types of information with the orthodromic route, we reach the conclusion that although the flight plans register a longer path, in reality, the pilots tend to fly over a more direct path that overall cross fewer kilometers. When it comes to the CRCO charges, it appears that the flight plans, on average, result on lesser charges; however, due to the method which the distances of real flights are calculated and considered, explained in section 4.2.4 of this thesis,
when a flight reenters a charging zone that distance may be accounted for twice, which results in the increase of the CRCO charge.

<table>
<thead>
<tr>
<th>Planned Route</th>
<th>Average Length</th>
<th>CRCO Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10,65%</td>
<td>2,32%</td>
</tr>
<tr>
<td>2</td>
<td>10,73%</td>
<td>2,32%</td>
</tr>
<tr>
<td>3</td>
<td>10,36%</td>
<td>12,43%</td>
</tr>
<tr>
<td>4</td>
<td>8,98%</td>
<td>11,07%</td>
</tr>
<tr>
<td>5</td>
<td>9,65%</td>
<td>8,77%</td>
</tr>
</tbody>
</table>

Table 5. 23: Planned flights vs orthodrome for SULAM-DELOG

<table>
<thead>
<tr>
<th>Real Route</th>
<th>Average Length</th>
<th>CRCO Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6,27%</td>
<td>12,43%</td>
</tr>
<tr>
<td>2</td>
<td>6,31%</td>
<td>11,07%</td>
</tr>
<tr>
<td>3</td>
<td>6,68%</td>
<td>10,42%</td>
</tr>
<tr>
<td>4</td>
<td>7,21%</td>
<td>8,77%</td>
</tr>
<tr>
<td>5</td>
<td>7,69%</td>
<td>10,42%</td>
</tr>
<tr>
<td>6</td>
<td>5,82%</td>
<td>10,42%</td>
</tr>
<tr>
<td>7</td>
<td>7,06%</td>
<td>11,07%</td>
</tr>
<tr>
<td>8</td>
<td>5,07%</td>
<td>12,43%</td>
</tr>
<tr>
<td>9</td>
<td>6,78%</td>
<td>11,07%</td>
</tr>
<tr>
<td>10</td>
<td>8,14%</td>
<td>10,42%</td>
</tr>
</tbody>
</table>

Table 5. 24: Real flights vs orthodrome for SULAM-DELOG

When obtaining the information of flights grouped by distinct route, arrival airport and departure airport, is it visible that the most frequent departure airports are GCLP and GCFV (Fuerteventura Airport, Spain), and the most common arrival airports are EDDL (Düsseldorf Airport, Germany), EGSS (Stansted Airport, England), and EDDV (Hannover Airport, Germany). Figure 5.13 shows some of the routes that make use of the studied segment.
5.5 Case Study Validations

The distance crossed in each charging sector will differ from planned/real routes to the orthodromic, being that the length might be larger in the direct route for some charging areas. However, the total distance given by the actual flight information must always be longer or equal to the direct route distance. This fact is observed in all case studies.

To validate the algorithm that calculates the CRCO charges, we obtain the route charges for a few example flights with equivalent trajectories from the NEST Tool, according to the method explained in the third chapter, in section 3.2. Then a comparison is made of the NEST calculated charges with the ones from the equivalent planned flight route generated by the develop Flight Planning Support Tool. An example for each of the case studies is presented.

5.5.1 LEMD-GCLP validation

Two flights on an A320 model were extracted from the NEST Tool. They equvalate to the planned routes and are represented in figure 5.14.
We obtain the total CRCO cost per flight and compare the two, calculating the error in table 5.25.

Table 5. 25: Error assessment for LEMD-GCLP planned flights

<table>
<thead>
<tr>
<th>Source</th>
<th>Flight</th>
<th>Cost (€)</th>
<th>Error, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPT</td>
<td>Planned1</td>
<td>1168.75</td>
<td>0.8305%</td>
</tr>
<tr>
<td>NEST</td>
<td>AA64993272</td>
<td>1159.12</td>
<td></td>
</tr>
<tr>
<td>FPT</td>
<td>Planned2</td>
<td>1084.15</td>
<td>0.0524%</td>
</tr>
<tr>
<td>NEST</td>
<td>AA64995158</td>
<td>1083.58</td>
<td></td>
</tr>
</tbody>
</table>

**5.5.2 LFPO-SAMAR validation**

Although in the NEST Tool we export data from a complete flight, and not just to a navigation point, the CRCO charges can be exported per charging zone. To get the total NEST calculated cost for this case study, since SAMAR is an entry point for the GC area, all the charges except for GC were summed.
The total costs obtained, as well and the difference between them, is shown in the table below.

Table 5. 26: Error assessment for LFPO-SAMAR planned flights

<table>
<thead>
<tr>
<th>Source</th>
<th>Flight</th>
<th>Cost (€)</th>
<th>Error %</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPT</td>
<td>Planned3</td>
<td>1579,07</td>
<td>0,4861%</td>
</tr>
<tr>
<td>NEST</td>
<td>AA65032724</td>
<td>1571,43</td>
<td></td>
</tr>
<tr>
<td>FPT</td>
<td>Planned5</td>
<td>1686,38</td>
<td>0,5078%</td>
</tr>
<tr>
<td>NEST</td>
<td>AA65289484</td>
<td>1677,86</td>
<td></td>
</tr>
</tbody>
</table>

5.5.3 ECKOS-EGSS validation

The chosen route from to validate this case study is a flight from GCLP to EGSS, as it equivalent to the most frequently planned route between these limit points – the GCLP airport is close to the ECKOS navigation point. For this flight an example with an A320 was not possible to obtain.

The NEST flights were performed on a B738 aircraft model, which as an average MTOW of 78,3 tons. The aircraft model used in the case study was an A320, that has a MTOW of 77 tons. For the CRCO charges to be comparable we convert them both to the weight factor of 77 tons.

The result obtained are shown in table 5.27.

Table 5. 27: Error assessment for ECKOS-EGSS planned flights.

<table>
<thead>
<tr>
<th>Source</th>
<th>Flight</th>
<th>Cost B738 (€)</th>
<th>Cost A320 (€)</th>
<th>Error %</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPT</td>
<td>Planned1</td>
<td>-</td>
<td>2098,82</td>
<td>0,3068%</td>
</tr>
<tr>
<td>NEST</td>
<td>AA65113317</td>
<td>2109,99</td>
<td>2092,40</td>
<td></td>
</tr>
<tr>
<td>FPT</td>
<td>Planned5</td>
<td>-</td>
<td>1975,98</td>
<td>0,4787%</td>
</tr>
<tr>
<td>NEST</td>
<td>AA65121131</td>
<td>1983,10</td>
<td>1966,57</td>
<td></td>
</tr>
</tbody>
</table>
5.5.4 Conclusions

Small variations of the CRCO charges are observable in all case studies – all the calculated errors are below 1%. Given that the files for the Unit Rates and the maximum take-off weights for each aircraft model is the same for both tools, the difference of total cost is attributed to a difference in the calculated crossing distance. This difference, as it will be explained in the present section, is expected, and the overall results indicate a good approximation of the developed tool results to the information provided by the EUROCONTROL NEST Software.

The NEST Tool follows the CRCO guidelines for cost assessment, but the specific method of distance calculation is not detailed in the documentation. The selected routes for validation have the same waypoints than the planned routes that they’re being compared to. However, even with a common route, the flight plans, as given by the NEST Tool by the .so6 file that is then imported into the database, don’t have the same total distance – the distance presented for each route is an average of the distances of the individual flights that have that route. So even in flight plans, although minimal, there will be some differences in total kilometers in the same route – which means that there will be a difference as well when comparing to one individual similar flight directly extracted from the tool for validation.

Differences can be due as well to the segment used to calculate the intersection point. In the Flight Planning Support Tool, the segment that intersects a certain area is defined by navigation points. The NEST Tool however registers the coordinates every time the aircraft enters a new elementary sector, so the limit points with which to calculate the intersections may be different, like the example from figure 5.17 below, with the detail of a planned flight from LEMD to GCLP indicates:

![Figure 5. 17: Detail from a flight plan from LEMD to GCLP (NEST Software).](image)

For the example in the figure above, the Flight Planning tool uses the coordinate of waypoint OBIPA and ELVAR depicted in the image above to calculate the crossing point between the LE and LP charging area, but the NEST Tool may use the entry coordinates to the elementary sector LETMARTIS.

If the NEST software uses different coordinates, unless these are colinear with the navigation points, the intersection point calculated will not be exactly equal and the crossing distances won’t be
as well.

No comparison was made for the SULAM-DELOG trajectory because the NEST Tool only returns charges for an entire flight, from airport to airport, with the possibility to see the charges per charging zone. The SULAM navigation point however is inside the GM area, and since it is not an airport there is no way to obtain from the NEST tool the charges for that distance, so the values wouldn’t be comparable.

5.6 Comparative Cost Analysis

As it can be concluded from all case studies, the shortest route possible between two points can have a superior CRCO charge cost. In this section an analysis of cost per kilometer versus charging zone costs is performed. Like explained in section 2.2, only a partial value of the CASK can be directly related to flight distance. For the purpose of this analysis, the considered portion will be 50% - this won’t be an exact portrayal of reality, because each airline has its own cost distribution, but it will be a close premise, based on the ICAO Economic Development document [13]. Two types of airliners will be analyzed: the low-cost carrier and the network carrier. Analyzing figure 2.4, we will assume for the purpose of these case studies an average CASK for the low-cost carrier of 4 cents, and 10 cents for the network airline. The airplane model for this study will be the same used in the case studies: Airbus A320, with a 180-passenger configuration for the value airline, and a 150 for the network carrier, like suggested in Oliver Wyman’s Airline Economic Analysis report [10].

The value per kilometer for each airline will be:

- $0.04 \times 0.5 \times 180 = 3.60$ euro / km for the value carrier;
- $0.10 \times 0.5 \times 150 = 7.5$ euro / km for the network carrier.

The cost analysis for the LEMD-GCLP and LFPO-SAMAR routes demonstrate that the costs, for both types of airlines, are sorted by the same order of route length. In this case the CRCO costs don’t influence the choice of route from a cost reduction perspective.

![Figure 5. 18 – Cost analysis for LEMD-GCLP on an A320.](image-url)
The planned routes with the lowest CRCO cost are the 2nd and 3rd (marked by the black frame on the graph). However, the cheapest route for both the low-cost and the network airline is the shortest, the 5th route. The most planned route, with the assumptions of this analysis, is the least cost effective. Factors like capacity limit or delays from sector on alternative routes could be an explanation for this choice.

The planned route with the lowest CRCO cost for the LFPO-SAMAR route is the 2nd most planned. The least expensive happens to be the most planned – and shortest.

For the ECKOS-EGSS segment, although the shortest distance planned route is in fact the cheapest, in the case of the 2nd and 3rd most planned route, the longest of the two is the cheapest to flight overall. The same isn’t true for the network airline, so the cost per kilometers supersedes the CRCO charges.
For the SULAM-DELOG segment, the same conclusion of the shortest route being the least expensive is reached. However, the first and second must planned route have the lowest CRCO charges, and both are longer than the third route, but are cheaper overall, for both airlines. In this case, the CRCO cost may have an impact in the route chosen.

Figure 5. 21: Cost analysis for SULAM-DELOG on an A320

For the low-cost airline, the most planned route is also less expensive than the second shortest (5th route). In this case, the second longest route is also the second cheapest.

Depending on the cost per kilometer of each airliner, the CRCO total charges can be a decision factor in the route choice for a specific flight. This cost effectiveness factor needs to be taken into account by an airspace designer when planning new routes.
6. Conclusions

In the current airspace landscape, the need to optimize and provide an adequate response to the air navigation industry's increasing demands is a pressing issue that is allocating efforts from the entire European ATM system. Our actual network system will be undergoing changes in the next few years with the widespread implementation of the Free Route Airspace concept that implicates the redesign of the airspace routes.

The importance of route efficiency in this redesign is obvious, and there needs to be not only a deep knowledge of the existing airspace structure and the network but also of the traffic and its patterns. New entry and exit points of each charging zone, or any new navigation points in the current network, need to take into consideration the total crossing distance, because not only is cost per kilometer key in the industry but also reducing CO2 emissions is one of the goals of the ATM improvement approach. Route distance reduction, however, must not impair traffic flow or block sector capacity.

With these goals in mind, the work developed in the present dissertation manages to answer some of the needs for route design that aren't completely fulfilled by the currently available software, and the additional features provided by the Flight Planning Support Tool allow for a higher comprehension of the route network and its use by the airlines.

6.1 Achievements

One of the main requirements that needed to be addressed with the Flight Planning Support Tool developed was the gathering and grouping of historical flight information, by route, and their subsequent selection according to the criteria of most frequently used. This was achieved for both types of information: planned and real trajectories.

The display of route data like the total distance, the CRCO charges and the detail of the route waypoints was also an apprehended goal.

A good route performance indicator is the orthodromic path between the same limit points, being that it is the optimum when it comes to distance; therefore, this dissertation also fulfills the requirement of having a common baseline route with which to compare the existing planned and real routes, not only in total length but in CRCO charges cost as well.

Another important feature that was accomplished was the ability to observe and retrieve the statistics for the specific route segment between the chosen limit points, as opposed to the entire airport-to-airport trajectory. This allows for the evaluation of a route segment that may serve several different flights, as the added section that presents the most frequent departure and arrival airports in the outputs statistics file demonstrates.

The ability to visualize the selected routes was also crucial for the study of the network and the validation of the tool results. The .kml files made available by the Flight Planning Support Tool allow for a clear understanding of the statistics and the ability to see a side-by-side display of the planned and
actual trajectories, which is not possible in the NEST Tool software.

Lastly, an additional advantage of the created tool is that only one executable is required to be run by the user, since the C++ application automatically makes the connection with the SQL database, and only very simple inputs are required.

6.2 Future Work

Even though the proposed goals for the Flight Planning Support Tool within the scope of this thesis were achieved, the tool may be improved, and further features that contribute for a completer and more versatile tool for route design could be added:

- **Group overlapping routes**: the current tool considers a route by the distinct sequence of its waypoints. If there are two routes that have a different set of registered waypoints that happen to be colinear, resulting in the same trajectory, the tool will still consider them to be different routes. An algorithm that checks, for routes with a high percentage of common waypoints, if the coordinates of any differing point are common to both routes could be implemented to group those routes.

- **Addition of one (or more) intermediate waypoint inputs by the user**: the currently developed tool selects the route based on its limit points (point A to point C). The possibility for the user to add an intermediate route point B could be added, and the program would select the most used planned and real routes that cross that point as well. For the direct route, the program can calculate the orthodromic trajectory from A to B, and then from B to C - this is currently possible, but it requires two executions instead of one.

- **Include flight and sector delay**: flight delays for the real flights could be gathered and a distribution of route per average flight delays could be presented, since this is also an indicator of route performance and a symptomatic delay is evidence of an inefficiency in the designed route. This average should not consider outlier values so that it won’t deteriorate the perceived route performance due to an occasional higher delay that is not related to route design.

  Sector delay can also be a determining factor in route choice. An indicator of which sector the route crosses and the average delay it will be exposed to would be of great utility for the evaluation of sector capacity in route redesign. This new feature would imply the incorporation of new data into the database structure to contain the mapping of each sector’s boundaries, so that the intersection of the routes with these sectors could be calculated in the same way they currently are for the charging zones.

- **Search of the optimum route in the available airspace**: given the two limit points, if a cost per kilometer were to be attributed to a flight, a search algorithm that would return the optimum route could be implemented. Such algorithm would have to not only take into consideration the cost input from the user but also the criteria for each FIR – whether it is a Free Route Airspace, and which are
the mandatory crossing points; and for sectors where FRA is not implemented, the maximum acceptable distance between waypoints. The total cost of the flight can be the function to minimize in the search for the optimum available route, and the direct route length from each navigation point to the destiny may be considered as a heuristic function.
Bibliography

A.1 NEST Tool Traffic Query

(VIA S LECMFI OR VIA S LECMUIR OR VIA S LPPCFIR) AND NOT ((DEP LEMD AND VIA ACC DAAACTA) OR (ARR LEMD AND VIA ACC DAAACTA)) OR (DEP LEMG AND VIA ACC DAAACTA) OR (ARR LEMG AND VIA ACC DAAACTA)

OR (ARR LEMD AND VIA S LFRRG) OR (DEP LEMD AND VIA S LFRRG) OR (ARR LEMG AND VIA S LFRRG) OR (DEP LEMG AND VIA S LFRRG) OR

(ARR LEMD AND VIA S LFBBBDX) OR (DEP LEMD AND VIA S LFBBBDX) OR (ARR LEMD AND VIA S LFBBBPZ) OR (DEP LEMD AND VIA S LFBBBPZ)

OR (ARR LEMG AND VIA S LFBBBDX) OR (DEP LEMG AND VIA S LFBBBDX) OR (DEP LEMD AND VIA ACC LECBCTAW)

OR (ARR LEMD AND VIA ACC LECBCTAW) OR (DEP LEMD AND VIA ACC LECBCTAW)

OR (ARR LEMD AND VIA S LFRRFIR) OR (DEP LEMD AND VIA S LFRRFIR) OR (ARR LEMG AND VIA S LFRRFIR) OR (DEP LEMG AND VIA S LFRRFIR))