Applied Mathematics

Helmut Schmidt Universität
Universität der Bundeswehr Hamburg

Angewandte Mathematik und Optimierung Schriftenreihe
Applied Mathematics and Optimization Series
AMOS # 58(2017)

Liana Amaya Moreno, Armin Fügenschuh, Anton Kaier,
Swen Schlobach

A Nonlinear Model for Vertical Free-flight Trajectory Planning

Herausgegeben von der
Professur für Angewandte Mathematik
Professor Dr. rer. nat. Armin Fügenschuh

Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg
Fachbereich Maschinenbau
Holstenhofweg 85
D-22043 Hamburg

Telefon: +49 (0)40 6541 3540
Fax: +49 (0)40 6541 3672

e-mail: appliedmath@hsu-hh.de
URL: http://www.hsu-hh.de/am

Angewandte Mathematik und Optimierung Schriftenreihe (AMOS), ISSN-Print 2199-1928
Angewandte Mathematik und Optimierung Schriftenreihe (AMOS), ISSN-Internet 2199-1936

A Nonlinear Model for Vertical Free-flight Trajectory Planning

Liana Amaya Moreno[1], Armin Fügenschuh[1], Anton Kaier[2], and Swen Schlobach[2]

[1] Professorship of Applied Mathematics, Department of Mechanical Engineering, Helmut Schmidt University/University of the Federal Armed Forces, Hamburg, Germany.
{lamayamo, fuegeschuh}@hsu-hh.de
[2] Lufthansa Systems AG, Kelsterbach, Germany.
{anton.kaier, swen.schlobach}@lhsystems.com

Abstract. The traditional approach of flight trajectory planning for commercial airplanes aligns the routes to a finite air travel network (ATN) graph. A new alternative is the free-flight trajectory planning, where routes can use the entire 4D space (3D+time) for more fuel-efficient trajectories that minimize the travel costs. In this work, we focus on the vertical optimization part of such trajectories for a fixed horizontal trajectory, computed or manually derived beforehand. The idea is to assign to each of the trajectory's segments an optimal altitude and speed for the cruise phase of the flight. We formulate this problem as a nonlinear programming (NLP) problem. As for the input of the model, information about the airplane's fuel consumption is provided for discrete levels of speed and weight values. Thus a continuous formulation of this input data is required, to meet the NLP requirements. We implement different interpolation and approximation techniques for this. Using AMPL as modeling language, along with nonlinear commercial solvers such as SNOPT, CONOPT, KNITRO, and MINOS, we present numerical results on test instances for real-world instance data and compare the resulting trajectories in terms of the fuel consumption and the computation times.

Keywords: Nonlinear Programming, Free Flight Trajectory Optimization, Modeling.

1 Introduction

In recent years free-flight trajectory planning came into the focus for the commercial airline industry. It provides a new way to deal with the rapid growth of the air traffic in Europe [3] and the resulting difficulties that this entails for the air traffic management (ATM). Although the priority of the ATM is to ensure the safety of the flight operations, other factors such as CO_2 emissions and, directly related to this, fuel costs, could benefit if all three goals are considered simultaneously by an integrating approach. This translates into computing fuel optimal trajectories that reduce the environmental degradation due to carbon fuel combustion and might further lead to a reduction in costs given the ever growing prices of fuel in the last years. From a computational point of view, the challenge is to find trajectories, composed of adjacent segments connecting two points (on the earth's surface), that avoid head-winds and benefit from tail-winds. Moreover, a time constraint is always enforced in order not to incur extra costs due to early or late arrival. This 4-dimensional problem (3 space dimensions plus time) is computationally difficult, and it is solved in practice typically in two subsequent stages: a horizontal phase, in which the segments of a 2-dimensional trajectory are computed, and then a vertical phase, in which different altitudes are assigned to the segments. Moreover, fuel consumption data is needed to optimally assign speed and altitude in order to minimize the amount of fuel used during the flight. Fuel information is given by the aircrafts manufactures, as a black box function which provides data only for a grid of points depending on speed, altitude and weight levels. In general during the optimization process, fuel consumption data is required for values that do not coincide with the given grid points, hence some techniques must be applied to obtain the required intermediate fuel consumption values. To come up with this continuous formulation of the data, different interpolation and approximation techniques are used. It is important to note that these drastically affect the computation times. In this study we concentrate on the vertical flight planning of commercial aircrafts. We propose an NLP model in which we integrate local and global interpolation and approximation techniques as continuous formulations of the problem's input data. We discuss briefly the characteristics of these formulations. Moreover, we compare different available commercial solvers for nonlinear programming for our test instances.

2 Mathematical Model

Our work is based on a model for vertical flight planning [6, 2], where speed and altitude are assigned to each of the segments that compose the trajectory, and the wind is assumed to be equal in all altitudes over one segment (but can vary from segment to segment). The fuel consumption is a bivariate function that depends on the current weight of the airplane (which is decreasing during the flight, since fuel is consumed) and the selected speed, see Figure 1. The fuel also depends on the flown altitude, which is de-

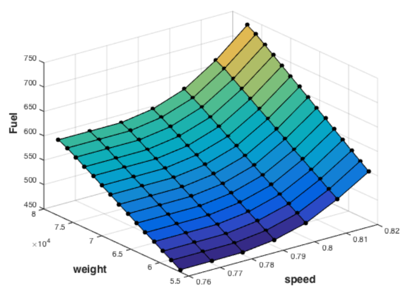


Fig. 1. Unit fuel consumption (kg per nautical mile) for the Airbus 320. The horizontal axis is the aircrafts speed (Mach number from optimal speed to maximal speed), and the vertical axis is the weight (kg).

termined in a post-processing step, once the optimal speed and weight are computed. Hence we do not need to consider altitude as a variable in our model. The objective of the model is to assign to each of the segments that compose the trajectory a speed and a weight value. Let n be total number of segments, and let $S = \{1,\dots,n\}$ denote the segment indexes. The nodes that link the segments then have the indexes $N = S \cup \{0\}$. Besides the fuel consumption data, the instance is further specified by the following data: L_i is the length of segment $i \in S$. The minimum and maximum duration of the entire trip are given by $\underline{T}$ and $\overline{T}$, respectively. The dry weight of the loaded airplane including the contingency fuel is W^{dry}.

We introduce the following variables: For each segment $i \in S$ the variable $v_i \in \mathbb{R}_+$ models the velocity of the airplane in this segment (the velocity can only be set once for the entire segment). The weight of the airplane at node $i \in N$ is denoted by $w_i \in \mathbb{R}_+$, and $w_i^{\text{mid}} \in \mathbb{R}_+$ is the "middle weight" of the airplane within segment $i \in S$, which is an auxiliary variable that is used in the computation of the fuel consumption $f_i \in \mathbb{R}_+$. The mathematical model reads as follows:

$$\begin{aligned}
\min \quad & w_0 - w_n && (1)\\
\text{s. t.} \quad & t_0 = 0, \qquad \underline{T} \le t_n \le \overline{T} && (2)\\
\forall i \in S: \quad & \Delta t_i = t_i - t_{i-1} && (3)\\
\forall i \in S: \quad & L_i = v_i \cdot \Delta t_i && (4)\\
& w_n = W^{dry} && (5)\\
\forall i \in S: \quad & w_{i-1} = w_i + f_i && (6)\\
\forall i \in S: \quad & w_{i-1} + w_i = 2 \cdot w_i^{mid} && (7)\\
\forall i \in S: \quad & f_i = L_i \cdot \widehat{F}(v_i, w_i^{mid}) && (8)
\end{aligned}$$

The objective function (1) minimizes the fuel consumed during the trip. It is computed as the difference between the start and arrival weight. In equation (2) the starting time t_0 is set to zero, and the final time t_n is forced to be within the arrival time window. Equation (3) enforces the time consistency, and the equation of motion is given by (4). With equation (5) all the fuel is consumed during the flight. The weight consistency is enforce by equation (6). The middle weight is computed in (7) which is required to calculate the fuel consumption in each segment in (8), where $\widehat{F}(v_i, w_i^{mid})$ is the continuous approximation or interpolation of the discrete fuel consumption data. This function offers intermediate data points within the corresponding ranges. Both interpolation and approximation techniques accomplish this purpose, however, the choice between one or the other depends on user. Nonlinear solvers require information about the first and sometimes the second order partial derivatives of all functions used in the problem formulation. In our model, equation (1)-(7) are either linear or quadratic equations, therefore its derivatives are easy to compute and this is done automatically by the solver. For equation (8), we need to explicitly compute first and second derivatives so they can be passed on to the solver. In the following we briefly describe the interpolation and approximation techniques used in this work. Each of them yields a polynomial function in two dimensions for the fuel consumption function. The first and second derivatives of the fuel function are then approximated by taking the first and second derivatives of these approximations.

Bilinear: Bilinear interpolation is a local technique, where intermediate values computed based on the four neighboring points. The interpolant is obtained by performing a linear interpolation along each of the dimensions of the table, which leads to a second degree polynomial. Further details can be found in [1].

Bicubic: Bicubic interpolation is a local technique where intermediate values are computed based on the four neighboring points, the first and second derivatives of these points (which are approximated). This leads to a linear system of 16 equations where the variables are the coefficients of a 6th order polynomial in two dimensions. In this work we have approximated the derivatives at the points of the table by two different methods, using finite differences and using cubic splines. For more details we refer the reader to [5].

Cubic Splines: Intermediate values are computed based on the information of the whole table. Therefore we refer to it as a global technique. The idea is to construct one-dimensional cubic splines along all the rows of the table and evaluate them at one of the first coordinate of the intermediate point. With these new values, another one-dimensional spline is constructed and finally evaluated at the second coordinate of the intermediate point. If smoothing is desired, a smoothing parameter is used for the construction of the cubic splines (approximation method). This results in a new set of points that best approximates the surface using cubic splines. For further details we refer the reader to [1, 4].

3 Numerical Results

The models were written using AMPL as modeling language and solved by the NLP solvers SNOPT 7.2-5, CONOPT 3.5C, KNITRO 8.1.1, and MINOS 5.51. We have used similar test instances as in [2], that is, the airplanes Airbus 320, 380, Boeing 737 and 772. For each airplane several travel distances were tested ranging form 800 Nautical Miles (NM) for the B737. to 7500 NM for A380 and B772. Two different time windows were used for each distance, for a total of 42 instances. Each flight is divided into equidistant segments of 100 NM. Table 1 summarizes the features of the test instances. All instances were solved using a 6-core Intel Xeon E5 at 3.5 GHz and 16

Table 1. Maximal speed (in Mach number), dry weight and maximal weight in (kg), maximal distance (in NM) and number of segments $|S|$ for each instance.

Type	Max. Speed	Dry Weight	Max. Weight	Max. Distance	$\|S\|$
A320	0.82	56614	76990	3500	15, 20, 30, 35
A380	0.89	349750	569000	7500	30, 40, 50, 60, 70 ,75
B737	0.76	43190	54000	1800	8, 12, 15, 18
B772	0.89	183240	294835	7500	25, 35, 45, 55 ,65,75

GB RAM computing machine. In Table 2, we give the percentage of the instances that were actually solved within a 10% error of the global optimal values reported on [2] by each solver using the different methods. We have used the following abbreviations: Splines1 refers to cubic splines interpolation (no smoothing of the data); *Splines2* refers to the method of smoothing cubic splines; *Bicubic1* refers to bicubic interpolation using finite differences approximations for the value of the derivatives and finally, *Bicubic2* refers to bicubic interpolation using cubic splines to approximate the value of the derivatives. The results in Table 2 indicate that the most successful methods are

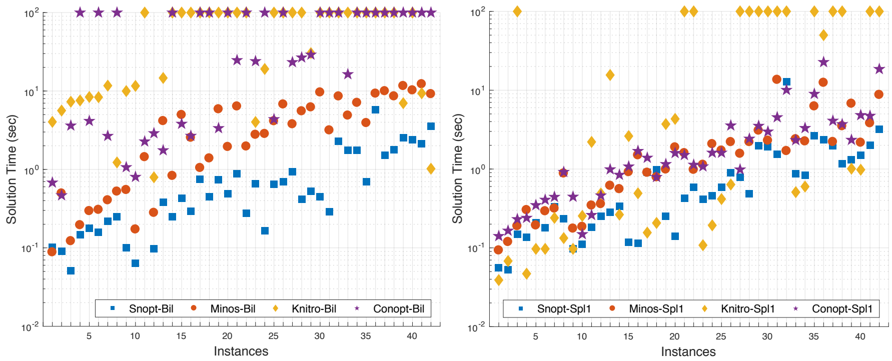
Table 2. Percentage of solved instances with each solver and each method.

	SNOPT	MINOS	KNITRO	CONOPT
Bilinear	100	100	43	48
Splines1	100	100	67	100
Splines2	17	31	5	5
Bicubic1	0	12	0	0
Bicubic2	0	10	0	0

Splines1 followed by *bilinear*, both interpolating techniques. *Splines1* is consistently, among all the solvers, the one that allows to solve the greatest number of instances. In order to compare these two methods, and the solvers as well, we give a graphical evaluation of the solution times in figure 2. On the x-axis of these plots, the instances are listed in ascending order according to their size, i.e., according the number of segments used for the trip. The data points, whose solution time are 100 seconds, represent the instances that were not solved, within a 10% gap from the global optimum. For both methods, the solution times of most instances are below 12 seconds. For the bilinear method, the solver Snopt outperforms the others. Note that the squared-shaped data points are consistently below all other data points. Most of the instances are solved within one second; the rest, within five. Minos is also very successful using bilinear interpolation, as the solution of all instances requires at most 10 seconds. The solution times of our instances using cubic splines interpolation are below 25 seconds. In this case, there is no straightforward outperformance of one solver over the others. On the contrary, the solvers take similar time to compute the (same) optimal solution. Knitro fails sometimes this purpose.

In conclusion, these two methods provide suitable continuous formulations of the input data, that can efficiently be integrated into our NLP models. Bilinear interpolation is very simple to implement and the number of computations needed is very low in comparison to the cubic splines method. The latter one requires the solution of many systems of linear equations (in the order of rows or columns in the input data table). Thus one can expect that cubic splines takes a longer solving time. Nevertheless, the derivatives provided with this last method, are smoother than the ones obtained with the bilinear interpolation, which can explain its faster convergence to a local optimum.

What is important to note here, is that the approximations of the derivatives with bilinear technique are still good enough for the solvers to search in good directions for local minima. As for the bicubic methods, it is obvious, they are not successful. A reason behind this, might be the inaccurate approximations of the required first and second derivatives at the points of the table.



a) *Bilinear* **b)** *Splines1*

Fig. 2. Solution times of all test instances using **a)** bilinear and **b)** cubic splines interpolation with each solver.

In our ongoing work we extend our methods to a full 4-dimensional trajectory planning. That is, to include a vertical optimization phase where a more realistic wind field (which can deviate also in altitude) is taken into account. This adds one more dimension to the fuel consumption data. On the other hand, introducing dynamic wind also increases one dimension to the wind data, therefore we need to study if the interpolation and approximation techniques, here presented, extend efficiently to more dimensions.

References

1. L. Amaya Moreno, A. Fügenschuh, A. Kaier, and S. Schlobach. An NLP Model for Horizontal Free-flight Trajectory Planning. 2017.
2. Liana Amaya Moreno, Zhi Yuan, Armin Fügenschuh, Anton Kaier, and Swen Schlobach. *Combining NLP and MILP in Vertical Flight Planning*, pages 273–278. Springer International Publishing, Cham, 2017.
3. EUROCONTROL. Challenges of Growth 2013: Task 4: European Air Traffic in 2035, June 2013.
4. D. Knott. *Interpolating Cubic Splines*, volume 18 of *Progress in Computer Science and Applied Logic*. Springer-Science+Business Media, LLC, 2000.
5. W.H. Press, S.A. Teukolsky, W.T. Vetterling, and Flannery. B.P. *Numerical Recepies in C. The Art of Scientific Computing*. Cambridge University Press, 2 edition, 1992.
6. Zhi Yuan, Armin Fügenschuh, Anton Kaier, and Swen Schlobach. *Variable Speed in Vertical Flight Planning*, pages 635–641. Springer International Publishing, Cham, 2016.