Aeroacoustical Module Development for the NOVEMOR MDO Software

Miguel Reis Rodrigues Dias
miguel.r.dias@ist.utl.pt

Instituto Superior Técnico, Universidade de Lisboa, Portugal

July 2017

Abstract

Noise emissions are a human health hazard and one of the top sources of annoyance to populations affected by aircraft noise. Its negative effects on well-being have led to increasingly strict noise standards and projects whose goals target significant perceived noise reductions, such as Clean Sky.

The EU FP7 project NOVEMOR was created to research innovative air vehicle configurations. In the scope of this project, an MDO software tool has been developed to allow analysis and preliminary aircraft design of novel configurations and morphing solutions. This work focuses on the foundations of an acoustic module integrated into this MDO framework, which will allow predicting noise emissions in the design phase. This module uses Farassat's 1A formulation to solve the acoustic equations allowing the computation of loading and thickness noise from the airframe. Landing gear and propulsion noise are integrated by semi-empirical models and tabulated data. Lastly, the module makes post-processing analysis for confronting the results with international standards (such as those defined by ICAO). In the future, this module can be used as a basis for a variety of study scenarios and noise emission optimization inside the framework.

Keywords: aeroacoustics, airframe noise, noise emissions, preliminary aircraft design

1. Introduction

Aircraft noise is an integral problem in a world increasingly connected by air traffic. The air transportation sector, backed by political decision makers, is taking on noise emissions to support a sustainable future where the noise hazard to the population is mitigated.

Worldwide both scheduled air passenger traffic and air cargo traffic are projected to grow above 4.0% annually in the next 20 years [1]. This will accrue to a doubling of global air traffic.

As noise exposure rises, the public and political decision makers are progressively aware of its health detriments.

Action has been taken in two ways to address noise emissions.

Aviation authorities (*e.g.* European Aviation Safety Agency (EASA), Federal Aviation Administration (FAA)) release industry guidelines and enact noise emission limits on new aircraft. These rules have been made stricter over the years, keeping up with new noise reduction technologies.

Concurrently, incentives were created towards research on technologies that reduce noise emissions. As examples, we cite the EU Research and Innovation Framework Programs (FP) and the aeronautics focused public-private partnership Clean Sky.

The EU FP7 project NOVEMOR focused on researching novel air vehicle configurations. In its scope, IST developed a software tool, an MDO framework [2]. This framework is extensible and now counts with several modules which allow analysis in several areas such as aerodynamics, structures and propulsion among many others. So far it lacks the capabilities to predict acoustic emissions. The drive for this work is then to endow the MDO framework with the needed acoustic prediction tool.

In this work a software tool was developed with four main goals:

- Develop a low demanding software module that predicts noise emissions;
- Be flexible as to the scenarios it can study;
- Evaluate noise emissions and test if they are inside certification standard's limits;
- Be ready to integrate into the framework.

This tool will have the particularity of being integrated into a conceptual design MDO framework for conceptual preliminary designs, of which there are not many.

2. Background

Noise emissions from an aircraft have many sources. We divide the sources into 3 major groups which can be seen in Fig. 1.

The first group is the aircraft fuselage and lifting surfaces, which here are addressed in the same way. The second important group is the engines, a major noise contributor. Finally, we have the landing gear noise. This source is usually grouped with the rest of the airframe noise, however, it will be addressed separately here.

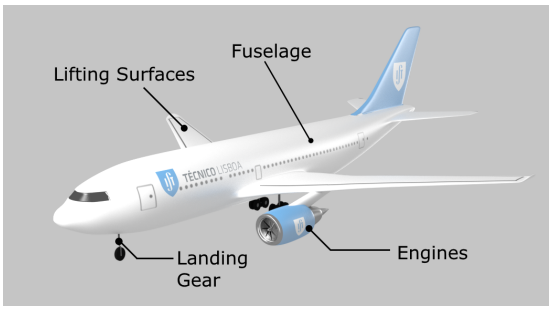


Figure 1: Noise sources in an aircraft.

2.1. Formulation 1A of Farassat

The noise from the airframe is computed using Farassat's Formulation 1A (F1A) [3]. This is a solution of the Ffowcs Williams-Hawkings equation, valid in subsonic scenarios. The total noise is given by adding two contributions: the thickness noise and the loading noise, which are given by the following equations:

$$4\pi p'_T(x,t) = \int_{f_s=0} \left[\frac{\rho_0 \dot{v}_n}{r(1-M_r)^2} + \frac{\rho_0 v_n (\hat{r}\cdot\dot{\vec{M}})}{r(1-M_r)^3} \right]_{ret} dS + \int_{f_s=0} \left[\frac{\rho_0 c v_n (M_r - M^2)}{r^2(1-M_r)^3} \right]_{ret} dS \,, \quad (1)$$

$$4\pi p'_L(x,t) = \int_{f_s=0} \left[\frac{\dot{p}\cos\theta_A}{cr(1-M_r)^3} + \frac{(\hat{r}\cdot\dot{\vec{M}})p\cos\theta_A}{cr(1-M_r)^3} \right]_{ret} dS + \int_{f_s=0} \left[\frac{p(\cos\theta_A - M_n)}{r^2(1-M_r)^2} + \frac{(M_r - M^2)p\cos\theta_A}{r^2(1-M_r)^3} \right]_{ret} dS \,. \quad (2)$$

where ρ_0 is the undisturbed air density, v_n is the normal component of the velocity to the surface, r is the distance from the source y to the observer x, $\hat{r}$ is the unit vector with the same direction as r, c is the sound speed, M is the Mach number, M_r is the Mach number in the radiation direction, p is the pressure on the surface, θ_A is the local angle between the surface normal and the radiation direction, S is the surface area, and $\dot{}$ denotes the first derivative in time.

2.2. Numerical Algorithm for Formulation 1A

Equations (1) and (2) must be solved numerically. Of the available algorithms, the Source Time-Dominant Algorithm was chosen. In this method, the computations are done for each source time when data is available, avoiding interpolations of the input data. Each source panel will contribute with a distinct unequal time spaced signal at the observer. These signals must be interpolated to a common set of times so all panel's contribution can be summed.

2.3. Source model Noise

Source models had to be implemented for the cases of the landing gear and engine noise. These would require too fine meshes to discretize in F1A, which would result in long computation times. Their pressure contributions from the source model are given by the generic function:

$$\langle p^2 \rangle = \frac{\Pi}{4\pi\,(r)^2} \frac{DF}{(1 - M_\infty \cos\theta)^4} \,. \quad (3)$$

where Π is the power function, D is the directivity function, F the spectral functions and term $(1 - M_\infty \cos\theta)$ accounts for the Doppler effect. Its specific definitions depend on the model in question. The models used for each contributor were:

- Landing Gear - Fink's model [4]
  1. Wheel
  2. Strut
- Engine
  1. Fan - Heidmann's model [5]
  2. Combustion chamber - Matta's model [6]
  3. Turbine - Matta's model [7]
  4. Jet - Stone's model [8]

2.4. Signal Processing

The noise signal that is computed with Formulation 1A is pressure in Pa as a function of time, while the noise signal computed by the engine noise models and landing gear are given as pressure in Pa as a function of one-third octave bands. They have to be converted to the same basis. As such the time signal from the airframe is first converted to the frequency domain using the Short Time Fourier Transform and separated into the one-third octave bands. At this point the signals can be combined.

Next, the signal is converted to Sound Pressure Level (SPL) by the expression:

$$SPL(dB) = 20 \log \left[\frac{p}{p_{ref}} \right] \,, \quad (4)$$

where p_{ref} is the reference pressure (2×10^{-5} Pa). From SPL the signal is converted to Effective Perceived Noise Level (EPNL). EPNL accounts for the

impact of both frequency and duration in the noise annoyance to an human observer. This parameter is relevant because it is the most used in legal and technical evaluations of aircraft noise. It is in this unit that the noise level limits are provided in the legislation.

3. Implementation

This software module was developed using Visual C++ using Visual Studio® 2013, like the rest of the software framework. The module was named Acoustic Prediction Module (APM).

3.1. Module Structure

A schematic of the program flow, separated in its core constituents, is presented in Fig. 2. Groups presented in parallel do not have precedence over each other, but all must have outputs for the next stage, which are tested by APM for their viability.

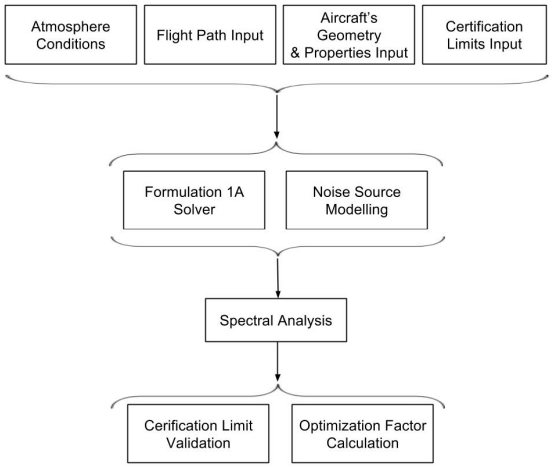


Figure 2: APM Structure.

In this diagram, only the core components for the acoustic calculations are shown, which leaves out the overhead controls, data management and validation, error handlers and the Graphical User Interface (GUI). The sections that follow will focus on the back end implementation. Details on the front end implementation are available in the developer version of APM's documentation.

3.2. Data Input and Files

There are 3 different ways to transfer data in and out of APM:

- module to module inside MDOGUI using the `SetInput()/SetOutput()` functions from `ModuleBase`, the base class for all modules in MDOGUI;
- loading/saving the data from/to a file;
- specifying the values through the GUI.

3.3. Flight Path Data

In this and the following four sections, we will cover the data structures used.

A discrete flight profile is required for APM to calculate the noise signal. Henceforth, each of these points in the flight path will be named a *case*.

For each *case* we must know:

- the time it occurs at;
- the aircraft's position and orientation;
- the aircraft's velocity;
- the ambient conditions - temperature, density and sound speed;
- the angle of attack and sideslip angle.

The angles of attack and sideslip are kept despite the fact that the aircraft's velocity and orientation are already stored. This is for convenience because they are readily available from input files and are required for the Aerodynamic module (Panel Method) call.

For a specific flight path the number of *cases* is fixed and for each *case* all terms must be known. Thus, the natural way to store the data was in matrices, where it was decided that each column represents a *case*. This matrix representation was implemented, like all other matrix members in APM, using the available matrix class in the MDOGUI software framework `CMatrixLib::FMatrix`. All quantities have SI units, apart from angles which can be either represented in degrees or radians.

Flight path data requires one additional field. In the current framework, some modules and files handle flight paths that are not continuous. In the transition between segments, there are discontinuities in all the input values. The time derivatives (Eq. (1) and (2)) in segment boundaries are meaningless and should not be computed. Thus each case in the flight path has an extra number assigned to it. This number is tested by the solver to find when a segment ends and another begins. In this manner, the input data can have segments of arbitrary length and the solver will not compute derivatives across segment boundaries.

3.4. Aircraft and surface pressure Data

Aircraft data pertains to geometry surface information and surface pressure information. The surface of the aircraft is modeled using panels. Each panel can be described by its position (x, y, z), its area and a coordinate system bound to the panel that indicates its orientation in space. Formulation 1A poses no restrictions on the panel's shape. This is the same discretization as used in other MDOGUI modules, namely the Aerodynamic module.

Surface pressure information is stored as a pressure coefficient (c_p) for each panel in the geometry. This coefficient is given by:

$$c_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty V_\infty^2} \ , \tag{5}$$

where p, ρ and V represent pressure, air density and velocity, respectively, while subscript ∞ stands for free stream conditions. These two sets of data, the geometry and the surface pressure, were grouped due to their strong connection.

A graphical representation of an aircraft discretized in this way in depicted in Fig. 3 which was obtained from the Aerodynamic module.

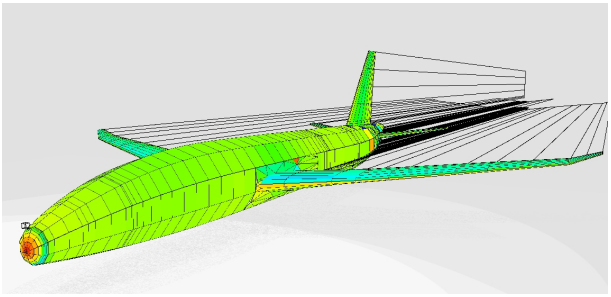


Figure 3: Aircraft mesh from the Aerodynamic Module. (Color signifies surface pressure).

3.5. Microphone Data

The microphone/observer is essentially a point in space. The point can be defined in a fixed coordinate system or relative to the aircraft's frame of reference. In this last case, one can additionally specify if it should fully track the aircraft's motion or only its translation. This tracking of the aircraft is indicated in the microphone as boolean flags.

The input of the microphone's coordinates can be done in either a Cartesian coordinate system (x, y, z) or spherical coordinate system (r, θ, ϕ). The Cartesian system will typically be more appropriate to input fixed ground observers, for example; while the spherical system is more useful in creating sets of observers placed in a sphere or hemisphere around the source, as is commonly done in acoustics to study noise directivity.

Additionally, each microphone has a tag that indicates if it is one of the points defined in the legislation limits (Approach, Flyover, Lateral).

3.6. Engine Data

The engine model is subdivided into many sub-models (*e.g.* fan, turbine, *etc.*). This leads to an intricate class to fully describe the engine. In each of these sub-models, the total noise is computed by adding the different noise components. Each component captures a fraction of the noise generated (*e.g.* the jet mixing noise, or the fan broadband noise).

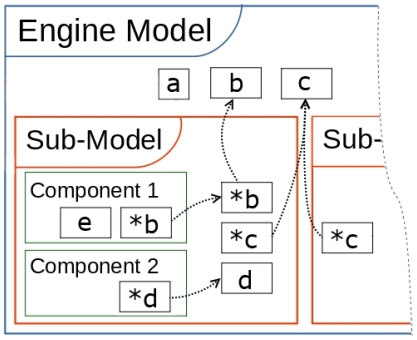


Figure 4: Engine model levels scheme.

As it is represented in Fig 4, not only each level may have its own data, but all must share information with the nested structures.

The list of all the data stored in these classes is too long to enumerate. However, this data can be subdivided into four main groups:

- **frequency data**, band frequencies from which noise emissions will be calculated
- **operational data**, data, defined by the user, portraying the engine operation during flight (*e.g.* fuel flows, shaft rotational speeds, flow temperatures, *etc.*);
- **geometry data**, user defined constants, that define the engine geometry (*e.g.* engine position relative to the aircraft, reference area, number of stator vanes, *etc.*);
- **noise model constants**, fixed constants which are characteristic of the models and should not be changed by the user.

3.7. Landing Gear Data

Landing gear noise is also obtained from a semi-empirical model. This model, however, is much simpler than that used for the engine. Moreover, the landing gear model is divided into only 2 components, the wheel and the strut, whose spectral and power functions are similar. The values required by the model are:

- position relative to the aircraft (x, y, z);
- gear geometry constants (wheel diameter, wheel number, and strut length);
- noise model constants for the computation of the power and spectrum functions.
- information on the deployment status of the landing gears.

The deployment status of the landing gear is simply given by an initial deployment flag and a time when this condition is reversed. This set of data is simple but sufficient to simulate flight paths with take off, landing or both take off and landing in the same flight profile.

3.8. Acoustic Prediction Algorithms

The noise predictions require different algorithms. Abstractly, to avoid overcomplicating the diagrams, we can represent the acoustic prediction scheme as two separate flows, one for source models (see Fig. 5) and one for Formulation 1A (see Fig. 6).

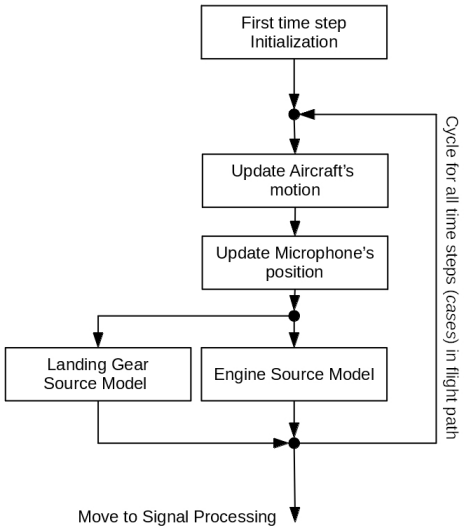


Figure 5: Source Models algorithm flow.

In practice, these two algorithms occur simultaneously in order to share the overhead of moving the aircraft and the microphones. From the perspective of the source model algorithm, the loop in path windows done in Formulation 1A is invisible. Some details of these algorithms will be covered next.

3.9. Sound Travel Time and Updating Moving Microphones' Position

When calculating the noise at an observer, it is required to compute when the signal arrives. For a fixed observer, it is easily found with the root of

$$t - \tau - |\vec{x} - \vec{y}(\tau)|/c = 0 \ , \tag{6}$$

where τ is the emission time, t the observer time and $\vec{x}$ and $\vec{y}$ the microphone's and the source's positions, respectively. The emission time, τ, is known and is constant for a specific *case*. Thus, for a fixed microphone, Eq. (6) is easy to solve algebraically.

When the microphones are in the aircraft's frame both t and $\vec{x}$ are unknown at start, thus Eq. (6) takes the form:

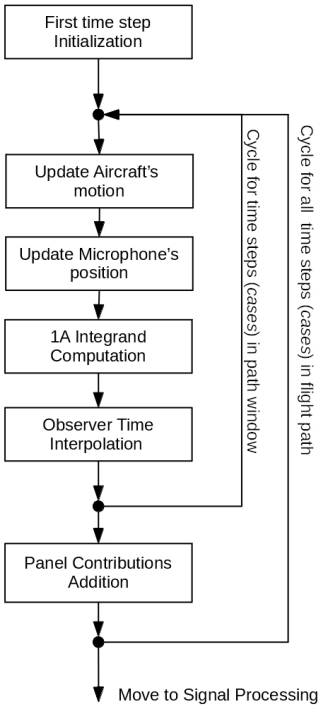


Figure 6: Formulation 1A algorithm flow.

$$t - \tau - |\vec{x}(t) - \vec{y}(\tau)|/c = 0 \ . \tag{7}$$

The root of this function must be found numerically. From the available approaches, the bisection method was selected.

3.10. Formulation 1A Solver Algorithm

The computation of Formulation 1A consists of evaluating the integrands of the thickness and loading noise equations and solving for the pressure perturbation.

The flow of the solver requires two time loops as can be seen in Fig. 6. The outer loop moves through emission time, effectively advancing the movement of the aircraft in its flight path. The need for the inside time loop, with a shorter time window, is due to the RAM intensive nature of the Source Time-Dominant algorithm.

The step representing the evaluation of the loading and thickness noise integrands, shown in Fig. 6, requires another algorithm which can be broken down. These integrands are the ones in equations (1) and (2). The parameters in these two equations can be divided into two groups: the ones that depend solely on the panel position, and those that also depend on the observer too. In the first group there are the following terms: ρ_0, c, ΔS_i, M, $\dot{\vec{M}}$, v_n, M_n, $\dot{v}_n$ and $\dot{p}$. While in the second we have: $\vec{r}$, $\hat{r}$, M_r and $\cos\theta$.

Each panel-microphone pairing is unique. As mentioned previously, there will be computations which are independent of the microphone. It follows naturally that between the two cycles, the one for all microphones and the one for all panels, the cycle for panels must be the outer cycle, as represented in Fig. 7. In this way, the number of computations is minimized.

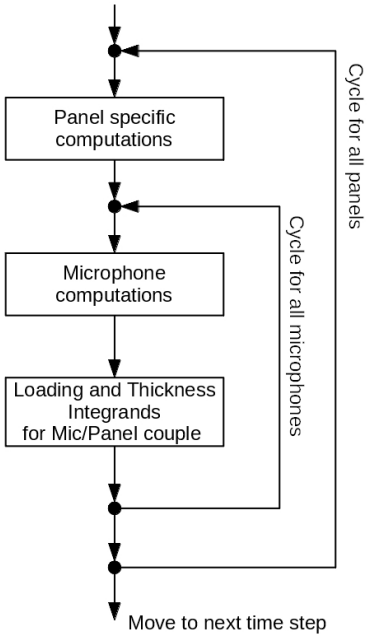


Figure 7: Algorithm flow for integrand computation for all Panel/Mic couple in one time step.

3.11. Source Models Algorithm

Each source model step (engine and landing gear) in Fig. 5 can be represented by the algorithm shown in Fig. 8. Source models, unlike Formulation 1A, already compute the noise pressure data in the frequency domain, and therefore a loop for all 1/3 octave bands' frequencies is required.

In each time step, prior to starting the loop in frequencies, the values that solely depend on the time step are computed, and the operational parameters are retrieved from the respective table for the current emission time. At the end of this procedure, the mean squared pressures for all frequency bands will be known for the current emission time.

These computations could be done for every time step, however, this is neither required nor useful. Formulation 1A computed the noise signal in time, and therefore it requires a small time step to provide a good resolution of the frequency spectrum. The source models already provide information across their frequency spectrum, and, because the results

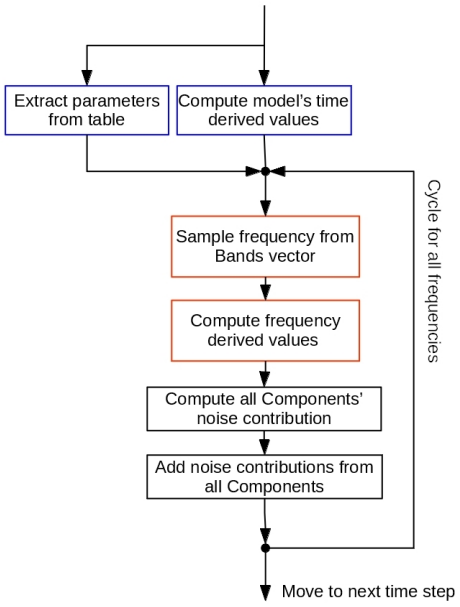


Figure 8: Source modelling algorithm flow in one time step.

are given for bands, no improved resolution would be achieved from smaller time steps.

On the other hand, the Short-time Fourier Transform must be applied to the Formulation 1A output signals in order to obtain their spectrum in time. Inevitably this transformation will reduce the resolution in the time domain. The intended resolution in time for the source models is left as a decision to the user for two reasons. First, the time resolution may not be uniform in a path and vary significantly with the flight path inputted. Second, the user may want to compute scenarios were Formulation 1A is ignored and only the noise from the models is computed, in which case the user should be free to set the resolution.

3.12. Signal Interpolation

The core algorithm to compute the interpolation of pressure results was based on that used in FAST [9]. Major differences were introduced to also allow the interpolation of source model data and replace the C-style arrays with MDOGUI's standard matrix class, `FMatrix`.

Because arrival times will differ they cannot be added directly. An interpolation must be performed to a shared set of interpolation times, t^*. To ensure that the correct interpolated values are added together, the sum is only performed at periodic intervals when there are enough points to allow a match between correspondent interpolated observer time contributions for all panels.

The source models' output signals must also be interpolated to a set of shared times. This interpo-

lation must be done, not for a single pressure value as before, but for the pressure in each 1/3 octave band. The total number of noise models sources will never be very large considering the large majority of aircraft has no more than three landing gears and four engines. This is at least two or three orders of magnitude smaller than the number of panels that discretize the geometry. As a result the source models do not pose problems of excessive memory usage and there is no need for the inner time loop.

4. Results

Next, the results from APM will be shown. The tests for the acoustic predictors are confirmed separately, starting with Formulation 1A. Lastly, the results for an example scenario with predictions from all models are shown.

4.1. Farassat Formulation Validation

For the acoustic implementation, we start with the validation of Formulation 1A for the airframe noise. No other cases of the Ffowcs Williams-Hawkings equation or a derivative were found being applied to compute full airframe noise (most often semi-empirical models are used). Thus, the validation of Formulation 1A could not be done *versus* another airframe noise predictor.

The strategy of validation was to confirm the F1A solver for a test case *versus* another tool that also uses Formulation 1A. This validated prediction software for comparison was from TU Delft and is based on the same principles. The test considered is a simple rectangular wing with the leading edge aligned with the z-axis and the chord is along the x-axis. The wing's dimensions and microphone position for this test are shown in table 1. The geometry was discretized using 33 280 panels. Velocities and pressure fluctuations were applied to the surface, matching the inputs exactly between the tools.

Table 1: Test rectangular wing and mic. position.

profile	chord (m)	span (m)
NACA 0018	0.2	0.8
mic. x (m)	mic. y (m)	mic. z (m)
0.2	2.0	0.0

All the input data was modified such that it is compatible with APM's inputs. The results for APM and TU Delft are shown in Fig. 9. In this figure, it is also presented a complementary third set of data calculated by the acoustic prediction software Farassat 1A Wind Turbines (FAWT) [10]. FAWT was developed in IST for the purposes of studying turbine noise emissions. It was validated from the same set of data provided by TU Delf and thus it is only shown here as a complement.

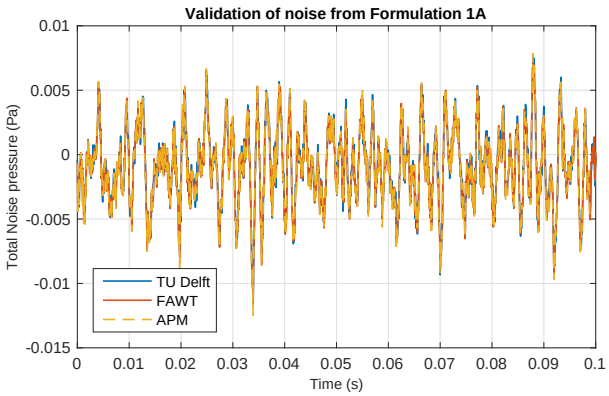


Figure 9: APM F1A results versus TU Delf and FAWT.

The root mean square error between APM's Farassat signal and TU Delft's is $5.885 \times 10^{-5} Pa$, which is two orders of magnitude smaller than the amplitude of the noise signal.

4.2. Landing Gear Model Validation

We now compare the Fink model as implemented in APM to experimental data found in [11], a work in which one of the goals was to study the behavior of the Fink model *versus* experimental data. The gear used in this case is a Boeing 777's main landing gear. Its relevant data is presented in table 2.

Table 2: Boeing 777 main landing gear data for Fink Model.

N. wheels	Wheel diam. (m)	Strut length (m)
6	1.27	3.89

The experimental data in [11] was obtained in a wind tunnel from a 6.3% scale model of the main landing gear. Because the landing gear noise depends on both polar (θ) and azimuthal (ϕ) angles, two angular positions were tested which are presented in table 3.

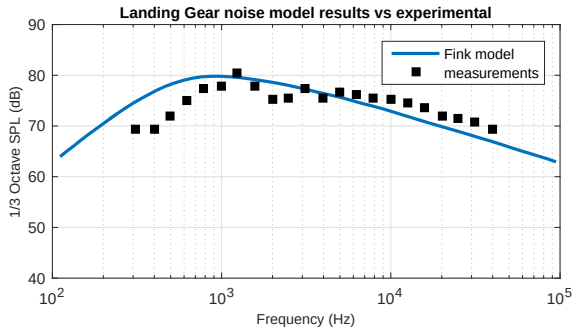
Table 3: Landing gear angular positions tested.

	θ (deg)	ϕ (deg)
condition 1	87.1	-1.0
condition 2	59.3	51.7

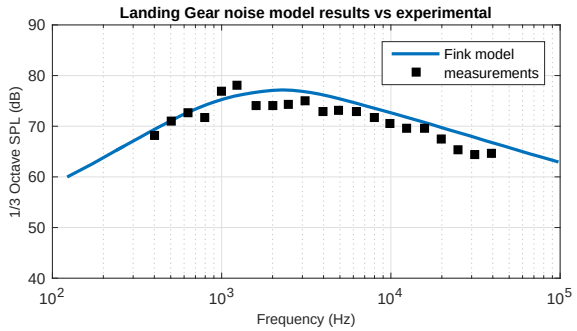
The observer is set at $24,19$ m from the landing gear for both conditions. Figure 10 presents the results from the Fink model implemented in APM and the measurements from the scaled model, showing a relatively good agreement.

4.3. Engine Model Validation

A process similar to that applied to the landing gear was also used for the engine. For the

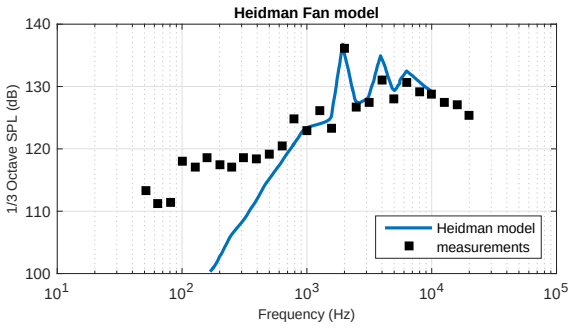


(a) $\theta = 87.1^o$, $\phi = -1.0^o$

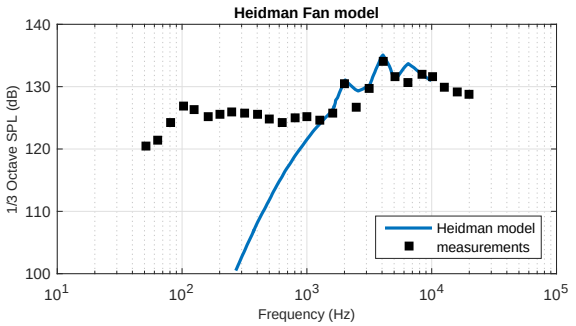


(b) $\theta = 59.3^o$, $\phi = 51.7^o$

Figure 10: APM's Fink model results.



(a) $\theta = 40^o$



(b) $\theta = 130^o$

Figure 11: APM's Heidman model results.

two most important noise sub-models, the fan and the jet, we checked the amplitude, spectral distribution, and directivity by comparing the models implemented in APM to the available model and experimental results. This was not possible for the combustor and the turbine. These have been the least studied models, and thus no references were found which had results and sufficient input data for APM. These models were verified component-wise *versus* the component graphs available in [12]. These component verifications were also performed for the fan and the jet.

4.4. Fan model validation

The Heidmann fan model implemented in APM was compared to the results presented in [5] for the fan labelled as A.

The microphones were placed 1 m from the fan in two polar angles, 40° and 130°, whose results are shown in figure 11.

There is a clear discrepancy in the lower frequency region but this was confirmed to be noise from other sources that could not be isolated in the experiments [13].

4.5. Jet model validation

A similar procedure was done to confirm Stone's jet model. In this case, the noise from a circular cold jet was predicted.

The other fluid properties, density, and temperature were taken to be equal to the ambient value as done in [8]. In the two conditions for the jet, the microphone was placed at 4.57 m from the origin and the two polar angles considered were: 120° and 145°. The results are shown in Fig. 12.

4.6. Example Scenario

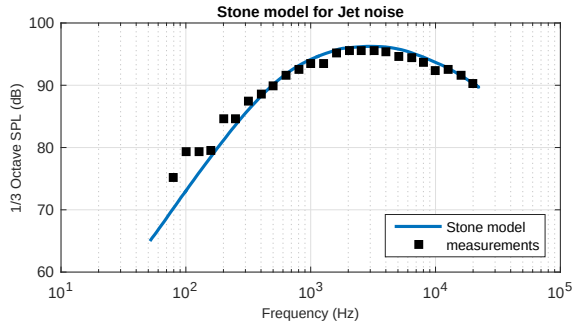
Having confirmed the noise models, in this section the results for a test scenario are presented. In this scenario, a take-off condition is considered and the noise is measured in the Flyover and Lateral points defined in the legislation. The noise was broken down into its several sources. The results are shown in Figure 13.

The geometry used for the aircraft was the one computed by the Aerodynamic module (Panel Method), which has been previously shown (Figure 3). The resulting noise emissions are identified in Fig. 13 with the label '1A' since they were computed using Formulation 1A.

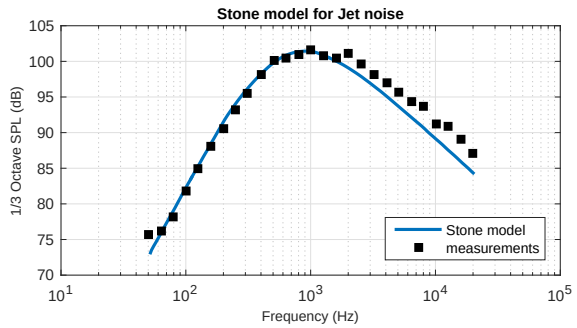
The source models were selected from the B-737. The B-737, probably due to its wide popularity, is commonly used in many acoustic studies. Therefore, it has the higher amount of reliable data for the acoustic models.

The B-737 landing gear data was extracted from [14]. Both main landing gear and the nose gear were used.

In this scenario, it was considered that the land-

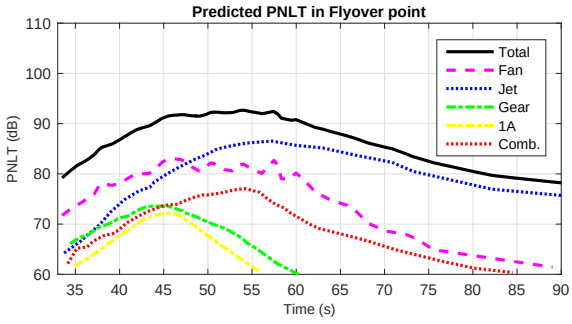


(a) $\theta = 120^o$

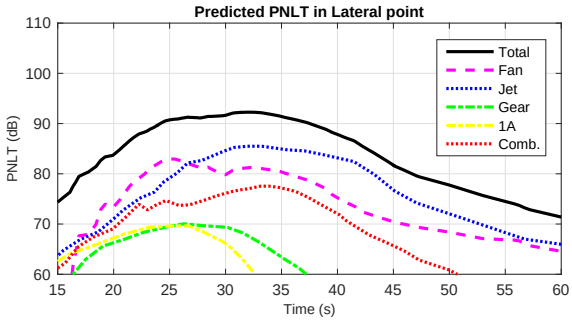


(b) $\theta = 145^o$

Figure 12: APM's Stone Jet model results.



(a) Flyover position



(b) Lateral position

Figure 13: PNLT results for example scenario.

ing gear was deployed for 45 seconds.

The engine modeled was the CFM56, which is used in many B737. Series seven is the most recent and the one for which most data was found available to the public, so it was the one used. Due to lack of geometric and operational data of this engine's turbine these models were not included, and only the fan, jet and combustion chamber contributions were computed.

From these results, one can notice that the jet and the fan are the dominant sources. Airframe noise is significantly lower than in the examples found in the literature. This is most likely due to the fact that slats and flaps were not modeled.

From the PNLT values, the EPNL values for each microphone were calculated (see table 4):

Table 4: EPNL results of test scenario.

	EPNL (EPNdB)
Lateral	104.18
Flyover	106.04

These results are above the allowable values for a single-aisle commercial jet's emissions. If we take the MTOW of the B-737-900 (79 016Kg) the limits according to Annex 16 would be 91.87 EPNdB for the flyover condition and 97.02 EPNdB for the lateral condition. The predicted values are similar to those obtained for large aircraft. Single-aisle commercial jets have present lower values. If we consider a B-737 value usually range from 80 to 90 for the flyover and 90 to 95 for the lateral position, depending on the model [15].

The largest contributor to this discrepancy is suspected to be the lack of implementation of atmosphere absorption effects. Because absorption effects are proportional to the distance covered by the noise signal, this coheres with the flyover EPNL being higher than the lateral EPNL. This behavior is contrary to real measured noise profiles. In our prediction, the Flyover point, being further away than the Lateral point, would congruently have a larger decrease in the received noise from atmosphere absorption.

4.7. Run Time

APM requires a fast turnaround to properly integrate into MDOGUI's workflow. For the example scenario presented in the previous section, the following run times were obtained:

Table 5: Computation times in example scenario.

	Time (s)
APM	682.26
Aerodynamic mod.	37 263.88

These computations were made with the release version of MDOGUI in a 64 bits system with 32,00 GB of RAM and a Intel® Xeon® E5620 processor.

From the results is can be seen that APM's computation is far less demanding than other critical modules. APM did all required computations in eleven minutes, whilst Aerodynamic module required around ten hours for all the corresponding geometry discretization and aerodynamics computations. As such, it will not be a bottleneck in the optimization pipeline.

5. Conclusions

A software module was successfully created and integrated into MDOGUI with the following implementations:

1. airframe noise prediction with Farassat Formulation 1A;
2. landing gear noise source modeling;
3. engine noise source modeling;
4. arbitrary flight path trajectories;
5. moving observers;
6. SPL, PNL, PNLT, EPNL calculation;
7. certification limits benchmarking;
8. communication with MDOGUI framework.

APM was made flexible and will allow MDOGUI to do a series of studies be it directivity of the noise emissions, legislation limit verifications and flight path optimizations.

Future work can improve APM with better algorithms for moving observers, absorption models and integration with the optimization pipeline.

Acknowledgements

The author would like to thank Professor Fernando Lau, José Vale and Frederico Reis for their support throughout work, and João Souto for the data which allowed the validation of Formulation 1A.

References

[1] Airbus S.A.S. Global Market Forecast - Mapping Demand 2016-2035. Blagnac Cedex, France, September 2016.

[2] F. Afonso, J. Vale, F. Lau, and A. Suleman. Performance based multidisciplinary design optimization of morphing aircraft. *Aerospace Science and Technology*, 67:1 – 12, 2017.

[3] F. Farassat. Derivation of Formulations 1 and 1A of Farassat. Technical Report NASA/TM-2007-214853, NASA, Langley Research Center, VA, USA, March 2007.

[4] M. Fink. Noise component method for airframe noise. *Journal of Aircraft*, 16(10):659–665, 1979.

[5] M. F. Heidmann. Interim prediction method for fan and compressor source noise. Memorandum NASA TM X-71763, Lewis Research Center, Cleveland, OH, USA, June 1979.

[6] S. B. Kazin , R. K. Matta , K. R. Bilwakesh , J. J. Emmerling , D. Latham. Supplement 1 - prediction methods. In *Core Engine Noise Control Program*, volume III. March 1976.

[7] R. K. Matta, G. T. Sandusky, V. L. Doyle. Ge core engine noise investigation - low emission engines, February 1977. FAA-RD-77-4.

[8] J. R. Stone. Interim prediction method for jet noise. Memorandum NASA-TM-X-71618, Lewis Research Center, Cleveland, OH, USA, January 1974.

[9] A. E. A. Vieira. Helicopter rotor noise, development of an acoustic software tool. Thesis to obtain master of science degree, Instituto Superior Técnico, Lisbon, November 2013.

[10] J. H. Souto. Development of an aeroacoustic prediction tool for wind turbine noise. Thesis to obtain master of science degree, Instituto Superior Técnico, Lisbon, 2017. To appear.

[11] C. Burley, T. Brooks, W. Humphreys Jr. and J. Rawls. ANOPP Landing Gear Noise Prediction Comparisons to Model-scale Data. In *13th AIAA/CEAS Aeroacoustics Conference*, Hampton, VA, USA, 21-23 May 2007. NASA Langley Research Center.

[12] W. E. Zorumski. *Aircraft Noise Prediction Program, Theoretical Manual*. NASA Langley Research Center, 1982.

[13] J. W. Hough, D. S. Weir. Aircraft Noise Prediction Program (ANOPP) Fan Noise Prediction for Small Engines. Technical Report NASA-CR-198300, NASA Langley Research Center, Marietta, GA USA, April 1996.

[14] R. A. Golub and Y.-P. Guo. Empirical prediction of aircraft landing gear noise. Technical Report NASA/CR-2005-213780, Hampton, VA, USA, July.

[15] Federal Aviation Administration. Noise levels for u.s.certificated and foreign aircraft: Advisory circular, May 2012. AEE-100.