

pyBADA: Easy BADA Integration in Python for Rapid Prototyping

Ramon Dalmau, Marc Melgosa and Xavier Prats
 Department of Physics - Aerospace division
 Technical University of Catalonia
 Castelldefels, Spain 08860

Abstract—The base of aircraft data (BADA) model provides accurate modelling of aircraft performances over the complete flight envelope for flight simulation and prediction applications. BADA is based on a generic total energy model, which performance functions (e.g., drag, maximum and minimum thrust, fuel flow) are particularised for each aircraft type with coefficients included in the BADA databases. BADA has a high reputation within the academic and research world, thus it is widely used for air traffic management research applications. This poster presents the architecture and capabilities of pyBada, a Python-based software designed for the easy integration of the BADA model in trajectory simulation, prediction and optimisation applications.

Keywords—BADA, Python, aircraft performance, optimisation

I. INTRODUCTION

The current air traffic management (ATM) is facing issues regarding environmental impact, cost-efficiency, safety and capacity. Aiming to address these challenges, the US and Europe are modernising their ATM systems through the NextGen and SESAR programmes, respectively. In this context, many modelling and simulation tools are developed by the ATM community to assess the future ATM systems and concepts of operation. Aircraft trajectory simulation, prediction and optimisation are key functions of these tools. In order to obtain realistic results, these functions require an aircraft performance model (APM) to accurately represent aircraft behaviour.

The two main approaches to aircraft performance modelling for prediction, simulation and optimisation of aircraft trajectories are kinematic and kinetic. The former describes the movement (i.e., displacement, velocity) of the aircraft without taking into account the underlying physics; while the later approach models the forces acting on the aircraft, which cause its motion. The Base of Aircraft Data (BADA) is an APM based on the kinetic approach developed and maintained by Eurocontrol, with the active cooperation of aircraft manufacturers and operating airlines. BADA was designed for trajectory prediction and simulation for purposes of air traffic management (ATM) research and operations [1], [2], and has high reputation within the academic and research world.

The BADA APM has been used in many ATM research applications, ranging from agent-based air traffic simulations [3]; fast-and real-time modelling and simulation tools [4]; environmental assessment tools [5]; assessment of new concepts of operations for the future ATM paradigm [6]; trajectory optimisation algorithms [7]; and trajectory prediction tools [8].

However, parsing the BADA datasets, coding the model specification and computing the derivatives of the performance functions (e.g., for trajectory optimisation purposes) could be a tedious and time consuming task which takes resources that could be devoted to the actual goal for which BADA is used.

At present, there exist a compiled library produced in C/C++ and provided by Eurocontrol that saves the hassle of coding the BADA model specification, thus minimising the chances of coding errors and reducing the implementation time. However, the compiled code produced in C/C++ most often runs on top of the hardware architecture upon which it is written. While this typically makes the code run faster, it could have the undesirable effect of making a compiled program machine and processor dependent. Furthermore, despite allowing the user to efficiently parse the BADA datasets and evaluate the BADA performance functions (e.g., fuel flow) at various flight conditions, the existing library does not provide function derivative information, which is of utmost importance for state-of-the-art, gradient-based trajectory optimisation algorithms.

The poster will present pyBada, a multi-platform library designed for a rapid, easy and transparent integration in Python of the BADA APM for ATM research purposes. The applications of pyBada include aircraft performance modelling, trajectory prediction and optimisation, and visualisation.

II. BACKGROUND

The pyBADA library allows for easy parsing of the BADA datasets and evaluation of the BADA performance functions and their derivatives at various flight conditions. The derivatives of the BADA performance functions are computed by using the automatic differentiation algorithms provided by the CasADi framework. Section II-A presents the BADA model; while Section II-B describes the CasADi framework.

A. Base of aircraft data

The base of aircraft data (BADA) aircraft performance model is based on the following total energy model (TEM):

$$\begin{aligned} (T - D)v &= mgh + mv\dot{v} \\ \dot{m} &= -f \end{aligned} \quad (1)$$

where T is the thrust of the engines; D the aerodynamic drag; m the mass of the aircraft; v the true airspeed (TAS); g the gravity acceleration; h the altitude and f the fuel flow.

In BADA, the thrust, drag and fuel flow functions of Eq. (1) are polynomial expressions which depend on a set of coefficients. Each aircraft type has a corresponding set of coefficients, which are specified in the BADA datasets. These datasets are ASCII (American Standard Code for Information Interchange) or XML (eXtensible Markup Language) formatted files, depending on the particular BADA family.

There exist three BADA families: BADA v3, designed to model aircraft behaviour over the normal operations part of the flight envelope; BADA v4, aimed at providing modelling capabilities for the complete aircraft flight envelope; and BADA H, which was recently released for modelling of helicopters. Regardless of the BADA family, parsing the datasets could be a tedious task requiring some programming skills; implementing the model specification could lead to potential coding errors due to the complexity of the performance functions; and computing the derivatives of these functions could be a cumbersome exercise requiring mathematical background.

B. CasADi

Despite CasADi started out as a tool for algorithmic differentiation (AD) the main focus of the framework rapidly shifted towards optimisation [9]. At present, CasADi provides a set of tools that drastically decreases the effort needed to implement algorithms for optimisation, yet without sacrificing efficiency.

The core of CasADi consists of a symbolic framework that allows to construct symbolic expressions and use these to define automatically differentiable functions. Once the expressions have been created, they can be used to obtain new expressions for derivatives or be efficiently evaluated.

III. PYBADA

According to the stack overflow trends, Python has been growing rapidly in the last few years. In fact, the latest report from Forbes shows that Python showed a 456% growth in 2018. Python was also placed at 3th place in the 2018 redmonk programming rankings, which order programming languages by the number of pull requests for code repositories on GitHub and tags on questions of stack overflow. Predictions show that Python continues to climb the ranks of the most popular programming languages. This is not surprising, since Python has many positives features: Firstly, Python code is relatively easy to read and understand; secondly, it supports multiple programming paradigms; thirdly, it has a large list of libraries.

pyBADA is a Python library build on top of CasADi, which has been developed by researches of the Technical University of Catalonia (UPC) for the easy implementation of aircraft performance modelling and trajectory prediction and optimisation algorithms with BADA. Besides allowing the user to parse the BADA datasets and evaluate the BADA performance functions with minimum programming effort, pyBADA can be also used to generate symbolic expressions of these functions to easily obtain their derivatives using AD thanks to the CasADi functionalities. This allows easy implementation of many trajectory optimisation and prediction algorithms, and sensitivity analysis requiring derivative information.

Furthermore, pyBada is multi-platform (i.e., it can be used in the most popular operative systems) and is very easy to install via pip, one of the most famous and widely used package management system to install Python packages.

pyBada is composed by several modules:

- **Atmosphere module:** This module implements standard atmosphere and simplified wind models; it also provides conversions tools of aircraft speeds, e.g., Mach to TAS.
- **Performance module:** This module parses the BADA datasets of the three BADA families; evaluates the performance functions of BADA at given flight conditions; and provides efficient derivatives of these functions.
- **Optimisation module:** This module calculates optimal speeds and altitudes for given flight conditions (punctual optimisation); and generates optimal trajectories in the vertical domain (integral optimisation).
- **Trajectory prediction module:** This module provides tools for predicting aircraft trajectories given the initial conditions, the BADA datasets, a sequence of flight intents and weather conditions.
- **Visualisation module:** This module provides tools for visualisation of aircraft trajectories, aircraft envelope and performance. It allows to visualise complex expressions such as the maximum rate of climb as a function of the altitude and speed.

REFERENCES

- [1] E. Gallo, F. A. Navarro, A. Nuic, and M. Iagaru, "Advanced aircraft performance modeling for atm: Bada 4.0 results," in *IEEE/AIAA 25th Digital Avionics Systems Conference (DASC)*, Portland, OR, Oct 2006.
- [2] D. Poles, A. Nuic, and V. Mouillet, "Advanced aircraft performance modeling for atm: Analysis of bada model capabilities," in *IEEE/AIAA 29th Digital Avionics Systems Conference (DASC)*, Salt Lake City, Utah, Oct 2010.
- [3] P. Volf, D. Sislák, and M. Pechoucek, "Large-scale high-fidelity agent-based simulation in air traffic domain," *Cybernetics and Systems*, vol. 42, pp. 502–525, 2011.
- [4] "Total Airspace and Airport Modeler (TAAM) factsheet," Jeppesen, 2018.
- [5] L. Cavadini and I. de Lpinay, "Assessing trends in aviation noise and emissions in Europe using advanced modelling capabilities IMPACT & AAT," in *2nd ECATS Conference*, Athens, Greece, November 2016.
- [6] R. Dalmau and X. Prats, "Fuel and time savings by flying continuous cruise climbs. Estimating the benefit pools for maximum range operations," *Transportation Research Part D: Transport and Environment*, vol. 35, pp. 62–71, 2015.
- [7] M. Soler, A. Olivares, and E. Staffetti, "Framework for aircraft trajectory planning toward an efficient air traffic management," *Journal of Aircraft*, vol. 49, no. 1, pp. 341–348, January 2012.
- [8] E. Gallo, J. Lopez-Leones, M. A. Vilaplana, F. A. Navarro, and A. Nuic, "Trajectory computation infrastructure based on bada aircraft performance model," in *IEEE/AIAA 26th Digital Avionics Systems Conference (DASC)*, Dallas, TX, Oct 2007.
- [9] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, In Press, 2018.