Fast contrail estimation with OpenSky data

Sun, Junzi; Roosenbrand, E.J.

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

JOAS

Fast contrail estimation with OpenSky data

Junzi Sun ●* and Esther Roosenbrand ●

Faculty of Aerospace University, Delft University of Technology, Delft, Netherlands
*Corresponding author: j.sun-1@tudelft.nl

Abstract

Contrails, formed under specific atmospheric conditions, have a noteworthy role in heat-trapping within the atmosphere. This study bridges the gap between theoretical contrail formation models and real-world data by employing flight information from OpenSky and meteorological data from the European Centre for Medium-Range Weather Forecasts. We introduce a computationally efficient contrail estimation module, leveraging a client-server architecture that allows on-demand weather data interpolation via an API, significantly reducing computational load and enhancing performance locally. The study also benchmarks the entire pipeline, from data acquisition to contrail prediction, offering a robust tool for future air traffic studies requiring interpolated weather data.

1. Introduction

1.1 Meteorological data for flight-based analysis

Aviation research has long been intertwined with meteorological data. Trajectory-based research often relies on weather data provided by meteorological institutes to ascertain the wind and temperature conditions of the flight, both in the strategic and tactical phases, as well as in post-analyses. For studies reliant on ADS-B data, where only ground speed is presented, wind becomes a crucial source of information for estimating the flight's true airspeed. This, in turn, facilitates a better approximation of aircraft performance. In an earlier study [1], the wind was found to be a significant factor in assessing aviation emissions.

Wind information is also pivotal for trajectory optimization studies. Given a specific aircraft and take-off mass, meteorological conditions are one of the primary factors causing variations in the actual optimal flight trajectory. In the study [2], the effect of wind on optimal trajectory generation was examined. One of the challenges in applying wind data in large-scale optimization has been the slow process of obtaining and processing the wind data.

1.2 Meteorology Information for Contrail Formation

The formation and persistence of contrails are highly contingent on the prevailing meteorological conditions. Understanding these atmospheric parameters is important for the analysis of contrail formation and its subsequent ramifications. These meteorological variables include temperature,

humidity, and atmospheric pressure at a given flight altitude.

The temperature and saturation levels in the surrounding atmosphere play a pivotal role in contrail formation. When aircraft engines expel hot exhaust gases into the cooler ambient atmosphere, contrails are likely to form if the atmospheric saturation is surpassed. The lower the temperature, the higher the propensity for contrail formation and persistence. Moreover, the relative humidity with respect to ice is a significant parameter [3], as contrails tend to persist in higher humidity conditions, evolving into contrail cirrus clouds under favorable conditions.

Atmospheric pressure, albeit to a lesser extent compared to temperature and humidity, also influences contrail formation. The pressure dynamics affect the rate at which exhaust gases from aircraft engines mix with the ambient atmosphere, thereby influencing the initial formation and subsequent dispersion or persistence of contrails. Additionally, wind conditions at flight altitude can impact the dispersion and longevity of contrails. Wind shear, for instance, can either elongate or dissipate contrails, affecting their visual and radiative properties.

1.3   Challenges of Integrating Meteorological Data

The primary challenge lies in the computational burden of processing large volumes of trajectory data against the four-dimensional meteorological grids, encompassing time, latitude, longitude, and altitude.

There are often two bottlenecks in utilizing the meteorological data. First and foremost, the request and download of ERA5 data from ECMWF can be time-consuming. For large datasets, the queueing for resources can take up to hours, with additional minutes required for downloading the data once it is ready. Given that only a single request can be run at a time, requesting ERA5 data for each trajectory with its bounding box is highly inefficient.

The second challenge pertains to the interpolation model. Constructing such a model necessitates the grid weather data (in GRIB format or netCDF format) to be fully loaded into the memory before the selection of a sub-region for the construction of the interpolation model. This task is often both I/O intensive and RAM intensive, posing significant hurdles to efficient data processing and analysis.

In this paper, we propose a new approach that addresses these two main challenges, significantly improving the speed of incorporating ERA5 weather data into aviation research. A new Python library, *FastMeteo*, is developed and shared as an open-source tool. We use contrail formation estimation as the primary use case to demonstrate the exceptional speed of obtaining interpolated ERA5 data for any given flight trajectory.

2. Data

2.1   Trajectory and Weather Data

This study leverages a representative sample of flight data obtained from the OpenSky Network [4], which boasts an extensive network of receivers scattered globally. The utilization of OpenSky data serves to illustrate the efficacy and expediency with which contrail estimation can be performed when dovetailed with a comprehensive meteorological dataset.

The requisite meteorological variables, namely temperature and relative humidity, are gleaned from the European Centre for Medium-Range Weather Forecasts (ECMWF) ERA5 dataset [5]. The ERA5 dataset has a high-resolution grid, sporting a resolution of 0.25 degrees horizontally. This granularity extends across 37 vertical layers, encompassing a pressure range of 1 hPa to 975 hPa, thereby providing a rich vein of atmospheric data crucial for nuanced contrail analysis. The reanalysis dataset is provided at each hour, providing accurate information for many aviation use cases.

In our use case of contrail analysis, only the ERA5 levels pertinent to en-route flights are considered, spanning between 100 hPa (approximately 53,000 ft) and 700 hPa (approximately 10,000 ft). The excluded levels either surpass the typical airliner ceiling or fall beneath the common altitude range for contrail formation.

2.2 Analysis-Ready, Cloud Optimized ERA5 Data

Crowdsourcing platforms like Pangeo Forge accelerate scientific pursuits by offering frameworks and cloud infrastructure to transform provider archives into Analysis-Ready, Cloud Optimized (ARCO) data stores, underscoring a community-driven approach towards generating analysis-ready, cloud-optimized data, including ERA5 data [6].

The Google ARCO-ERA5 dataset facilitates accessible global climate history in a cloud-optimized format. The open ARCO-ERA5 data, used in this paper, aims to make global climate history highly accessible in the cloud by the Google Cloud Public Datasets. This version of ERA5 has converted grib data to Zarr which is oriented towards common research and machine learning workflows.

Zarr, a format for chunked, compressed, N-dimensional arrays, presents several advantages over the commonly used GRIB format [7], particularly in cloud-based or parallel computing settings. It enables efficient chunking for better data access, compression for storage efficiency, concurrent read/write access beneficial in multi-threaded environments, and structured metadata storage for enhanced data interpretation and discovery. Transitioning from GRIB to Zarr within the ARCO-ERA5 significantly improves the accessibility and speed for loading a large amount of ERA5, making it a prerequisite for our approach.

3. System architecture of *FastMeteo* tool

In this research, we propose the *FastMeteo* python library to handle the acquisition and interpolation of the meteorology data for any given flight or set of locations. This architecture allows for on-demand data acquisition of ERA5 data from the Google ARCO datastore. We employ a local caching strategy to enhance computational performance.

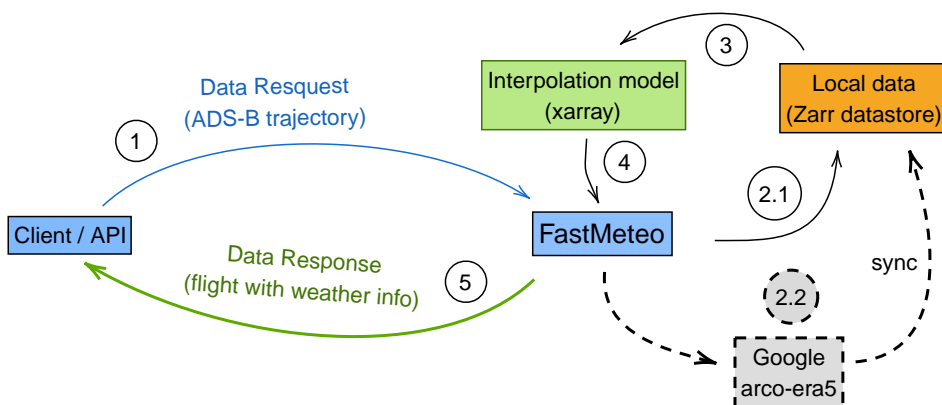**Figure 1.** The structure of the proposed *FastMeteo* architecture

Figure 1 shows the architecture of the *FastMeteo* system. It can be used as an imported common Python package if the data is stored on the local computer. It can also be deployed in a server-client model, where the ARCO ERA5 data is cached on a server, and the interpolation of weather data can be performed through API using only the client Python module.

There are several key processes involved in the design of this architecture, which are explained as follows:

1. Firstly, a request can be made either from an API (python function) or a client. The data frame is sent to the *FastMeteo* service, which is either hosted locally or on a remote server.
2. The *FastMeteo* server code checks whether the hours, based on timestamps from the data frame, are already downloaded in the local cache. If not, it will sync the hour of data with the Google ARCO-ERA5 public data store.
3. Based on the latitude and longitude boundary of the flight, we only load the region of data that is associated with the requested data frame. Since the data is stored in Zarr format, where data are stored in chunks according to the four-dimensional indices (time, latitude, longitude, and altitude), the extraction and loading of data are fast.
4. Next, the linear interpolation of relevant meteorological features is computed based using the xarray. This process is vectorized and also provides a signification improvement in computation efficiency.
5. Finally, a new data frame, containing the original information and new columns with interpolated meteorological information is returned or sent to the client.

The use of the library is extremely simple. Once the library is installed, it takes only two lines of code to get the weather information for a given flight or position. In the following code snippet (Figure 2), we first define the basic information of time, latitude, longitude, and altitude, which are required for the input. After that, the two lines of the code define the location for the local Zarr cache and then obtain the weather information through the *interpolate* function.

```python
import pandas as pd
from fastmeteo import Grid

flight = pd.DataFrame(
    {
        "timestamp": ["2021-10-12T01:10:00", "2021-10-12T01:20:00"],
        "latitude": [40.3, 42.5],
        "longitude": [4.2, 6.6],
        "altitude": [25_000, 30_000],
    }
)

# define the location for local store
fmg = Grid(local_store="/tmp/era5-zarr")

# obtain weather information
flight_new = fmg.interpolate(flight)
```

Figure 2. Example code using *FastMeteo* library to obtain meteorology data and provide interpolation of relevant parameters for the given flight

For example, the following snippet shows the input dataframe:

```
timestamp,icao24,latitude,longitude,altitude
2021-03-07 15:38:40+00:00,a56661,41.73921,-87.96647,10050.0
2021-03-07 15:38:45+00:00,a56661,41.7328,-87.97053,10275.0
2021-03-07 15:38:50+00:00,a56661,41.72649,-87.97452,10475.0
2021-03-07 15:38:55+00:00,a56661,41.72021,-87.97845,10675.0
2021-03-07 15:39:00+00:00,a56661,41.71364,-87.9826,10900.0
2021-03-07 15:39:05+00:00,a56661,41.70808,-87.98619,11125.0
...
```

while the new data frame provide by *FastMeteo* contains the following data:

```
timestamp,icao24,latitude,longitude,altitude,u_component_of_wind,v_component_of_wind,temperature,specific_humidity
2021-03-07 15:38:40+00:00,a56661,41.73921,-87.96647,10050.0,4.93797,-7.83555,266.78062,0.00019
2021-03-07 15:38:45+00:00,a56661,41.7328,-87.97053,10275.0,4.90272,-8.05699,266.43374,0.00018
2021-03-07 15:38:50+00:00,a56661,41.72649,-87.97452,10475.0,4.87196,-8.2515,266.12855,0.00018
2021-03-07 15:38:55+00:00,a56661,41.72021,-87.97845,10675.0,4.84209,-8.44564,265.82602,0.00017
2021-03-07 15:39:00+00:00,a56661,41.71364,-87.9826,10900.0,4.80977,-8.66434,265.48861,0.00017
2021-03-07 15:39:05+00:00,a56661,41.70808,-87.98619,11125.0,4.77882,-8.88355,265.15389,0.00016
...
```

When running the tool in a server-client mode. The following script can be used to start a FastAPI service on the server, which handles the flight date request, obtaining Google ARCO data if the partition is not on the server, perform the interpolation of weather data, and return the final data to the client.

```
fastmeteo-serve --local-store /tmp/era5-zarr
```

On the client side, the following code (Figure 3) can be used to submit and get the process flight with meteorology data, given that the flight data structure similar to previous Figure 2 is defined.

```
from fastmeteo import Client

# define the input data
#...

# create the client object
client = Client()

# send the flight and receive the new DataFrame
flight_new = client.submit_flight(flight)
```

Figure 3. Example code using *FastMeteo* library in a client-server mode

4. Contrail Formation Model

In this section, we explain how ERA5 parameters, including temperature and humidity, can be used for the evaluation of contrail formation based on ADS-B flight data. The models used for determining contrail formation and persistence are based on existing literature [3].

4.1 Schmidt-Appleman Criterion

Contrails primarily form under air temperatures lower than -40 °C (233 K) and high relative humidity [3]. Their formation is dictated by the Schmidt-Appleman Criterion [8], a thermodynamic model accounting for ambient pressure, humidity, and the water-to-heat ratio in exhaust plumes.

When an aircraft traverses atmospheric conditions that satisfy the Schmidt-Appleman Criterion, saturation with respect to liquid water occurs, leading to contrail formation. The specific temperature threshold for contrail formation depends on the ambient relative humidity and the slope of the isobaric mixing line (G), defined as follows:

$$G = \frac{EI_{H_2O} \; c_p \; p}{\varepsilon \; Q \; (1 - \eta)} \tag{1}$$

where constants are defined in Table 1. The isobaric mixing line, G, is influenced by the ambient pressure (p_h) at the flight level and the overall propulsion efficiency (η), which is assumed to be 0.4 according to literature [9],

The threshold temperature (T_{LC}) is the point where the saturation vapor pressure with respect to ice intersects with the mixing line. The saturation vapor pressure over water is depicted in blue in Figure 4 and is described by Murphy and Koop [10]:

$$
\begin{aligned}
\ln(e_{liq}) = {} & 54.842763 - \frac{6763.22}{T} - 4.210\ln(T) + 0.000367\,T \\
& + \tanh\left[0.0415(T - 218.8)\right]\left[53.878 - \frac{1331.22}{T} - 9.44523\ln(T) + 0.014025\,T\right]
\end{aligned}
\tag{2}
$$

where T represents the temperature, and satisfies $123K < T < 332K$. The saturation vapor pressure over ice, shown in orange in Figure 4, is given by:

$$
\log e_{ice} = 9.550426 - \frac{5723.265}{T} + 3.53068\ln(T) - 0.00728332\,T
\tag{3}
$$

where the temperature (T) is higher than 110 K. Before determining the threshold value T_{LC}, the temperature T_{LM} must first be identified, which is given by:

$$
T_{LM} = -46.46 + 9.43\ln(G - 0.053) + 0.720[\ln(G - 0.053)]^2
$$

Subsequently, the mixing line is defined as the tangent line at the T_{LM} point with the saturation water vapor pressure curve (e^*). T_{LC} represents the intersection of the mixing line and the saturation water vapor pressure curve over ice.

Table 1. Constants used in Schmidt-Appleman Criterion

Symbol	Constant	Value	Unit
EI_{H_2O}	Emission index	1.2232	$-(kg_{H_2O}/kg_{fuel})$
c_p	Specific heat capacity air	1004	$J\,kg^{-1}\,K^{-1}$
ε	Ratio molar mass of water vapour and air	0.622	-
Q	Specific combustion heat	43×10^6	$J\,kg^{-1}$
p_{st}	Pressure at steam point	101 325	Pa
p_{ice}	Pressure at ice point	611.73	Pa
R_v	Gas constant of water vapor	461.51	$J\,kg^{-1}\,K^{-1}$
R_d	Gas constant of dry air	287.05	$J\,kg^{-1}\,K^{-1}$

4.2 Ice Supersaturation

Persistent contrail formation is closely tied to the atmospheric conditions in the ice supersaturation (ISS) regions. Upon exiting aircraft engines, the hot and humid exhaust gases mix with the colder ambient air, leading to condensation and ice crystal formation if the ambient air is supersaturated with respect to ice. Ice supersaturation occurs when the relative humidity with respect to ice (RHI)

exceeds 100%, allowing for the possibility of ice crystal formation and persistence under suitable conditions. Ice supersaturation can endure for extended periods given the lack of efficient ice nucleating particles. This phenomenon is critical in understanding the life cycle of persistent contrails.

In this paper, we use the following equation to calculate RHI:

$$\text{RH}_i = \frac{e}{e_{ice}} = \frac{q\,p\,R_v}{R_d\,e_{ice}} \tag{4}$$

where e is the actual vapor pressure of water. q and p are specific humidity and pressure, which are provided by the ERA5 data. R_v and R_d are gas constants for water vapor and dry air (see Table 1).

Finally, the geometric approach of the Schmidt-Appleman Criterion to identifying T_{LC} is visualized in Figure 4.
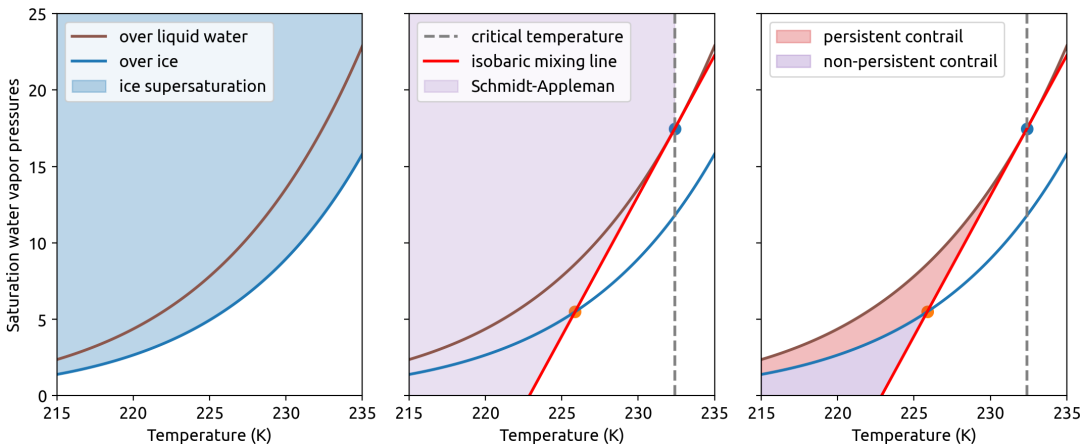

Figure 4. Contrail forming conditions

5. Results

5.1 Example trajectory

To showcase the capabilities of *FastMeteo*, we use an example flight trajectory sourced from OpenSky, depicted in Figure 5. On the left side of the figure, fundamental flight parameters from ADS-B are displayed, encompassing position, altitude, and ground speed. The right side of the figure presents the meteorological data we acquired, such as wind, temperature, and humidity.

Based on the additional temperature and humidity information obtained with *FastMeteo*, the determination of persistent contrails can be performed using the previously mentioned Schmidt-Appleman Criterion and ice supersaturation conditions. Consequently, segments of the trajectory that satisfy these conditions can be visualized alongside the original trajectory. This is depicted in Figure 6.

In the first plot, the segment of the trajectory satisfying the Schmidt-Appleman Criterion is highlighted in light purple. This segment encompasses the majority of the cruise flight due to the relatively cold temperatures. In the second plot, trajectory segments passing through ice supersaturated regions are marked in light blue. This occurs at the beginning of the trajectory, including portions of the climb and cruise. In the third plot, trajectory segments with estimated persistent contrails are marked in red, representing the overlap of the first two conditions.
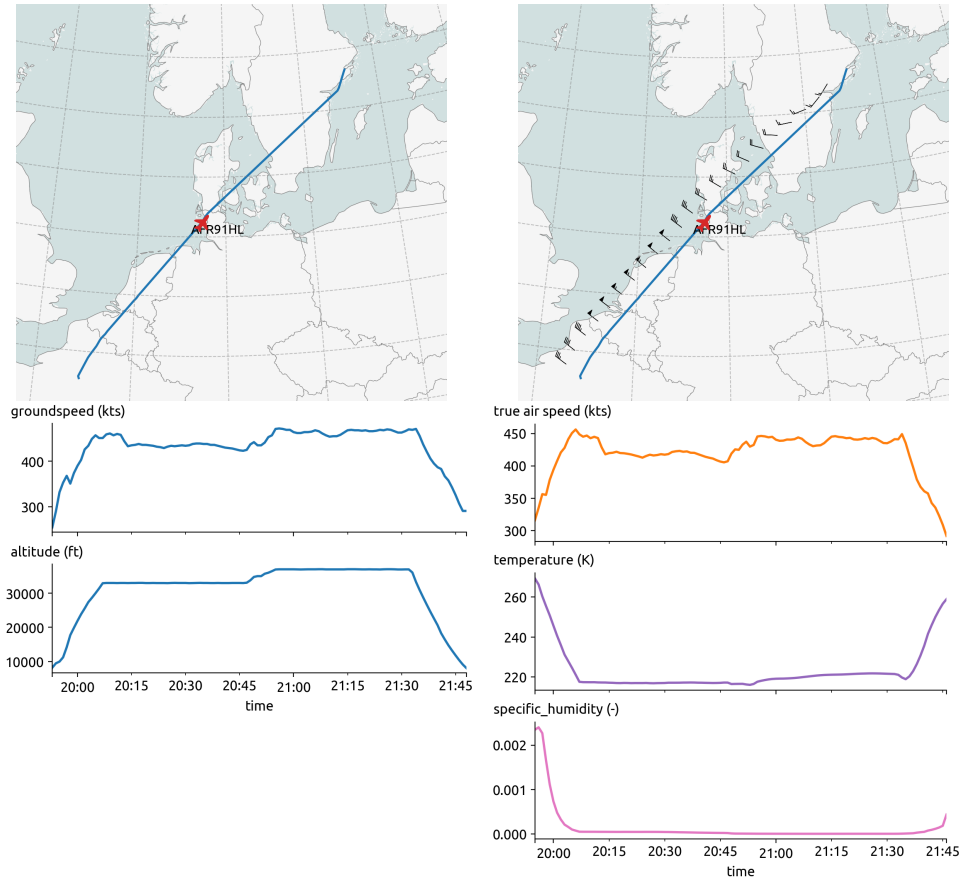
Figure 5. An example flight trajectory. On the left: the trajectory obtained from ADS-B data. On the right: enhanced trajectory data with meteorological information.
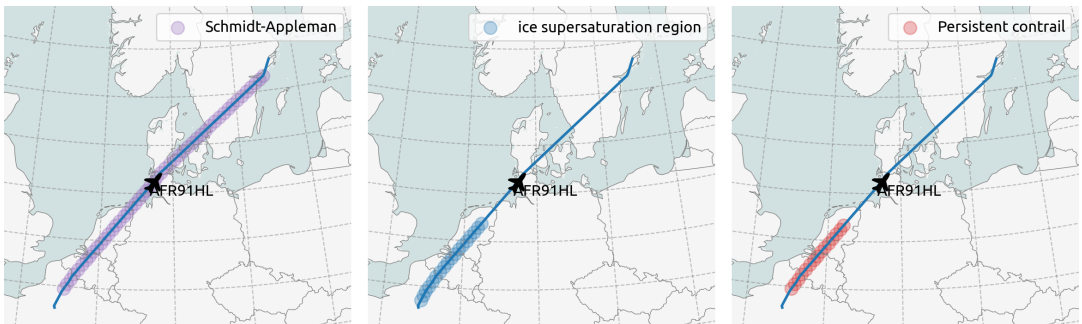
Figure 6. Persistent contrails (in red) determined from the example flight trajectory, based on the Schmidt-Appleman and ice supersaturation criteria.

5.2   Computational benchmarks

In this section, we provide a quick analysis of the computational speed of the *FastMeteo* library. For this test, we downloaded one day's worth of flights that departed from EHAM from the OpenSky network, encompassing around 850 flights. To test the influence of the number of data points, we

resampled the dataset at varying rates, ranging from 5 seconds to 600 seconds. This yielded a range of data points from approximately 860,000 to 8,000 for these scenarios. Table 2 displays the computational speeds for these different tests. Observations show that when operating on large geographic coverage, the number of data points doesn't affect performance significantly. For this dataset, computing the meteorological conditions for all 860,000 data points took less than eight seconds.

Table 2. Benchmarking the computational performance *FastMeteo*, with one data of all flights departed from EHAM, which covers a large geographic region.

sample time	data points	compute time
5 s	861,276	7.47 s
10 s	430,991	6.43 s
20 s	215,920	5.92 s
30 s	144,082	5.76 s
60 s	72,462	5.72 s
120 s	36,664	5.45 s
180 s	24,723	5.47 s
300 s	15,152	5.57 s
600 s	7,977	5.15 s

We also tested the library with individual flights in the dataset to show its performance when applied to single flights. The results are displayed in Figure 7. Most flights are processed in under 1 second.
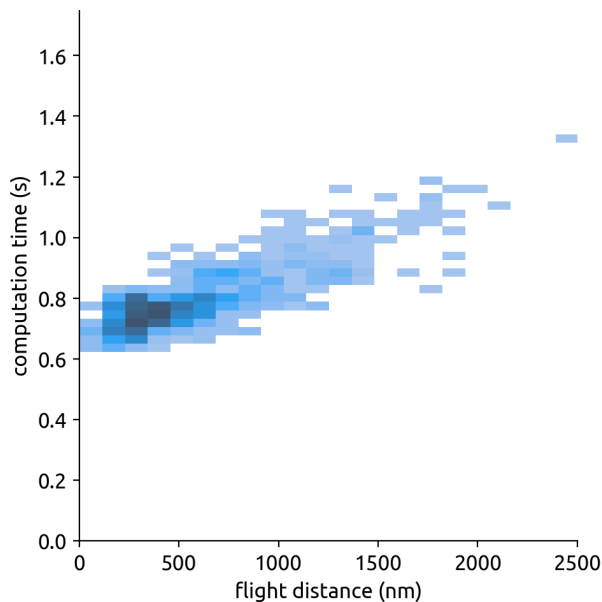
Figure 7. The relationship between computation time and flight distance. The colors indicate the varying density of the data. We can see that most of the flight in this sample is around 500 nautical miles, with a computation time of around 0.8 seconds.

Based on these two tests, we can determine that the performance is not directly proportional to the number of data points processed by the *FastMeteo* library. It is more efficient to process a large number of data points simultaneously because the interpolation of the multi-dimensional grid is vectorized, utilizing the underlying *xarray* library's interpolation functions.

6. Discussion

This research makes use of the Google ARCO-ERA5 public dataset, which is converted from the ECMWF ERA5 data. This implies that the data quality and errors remain consistent with the original ERA5 data. It is noteworthy that not all parameters from the ERA5 data are included in the ARCO dataset. However, for contrail analysis, critical information on temperature and specific humidity is available. Other commonly used meteorological data such as wind is also included, facilitating diverse aviation applications.

In addition to the standard grid presented in the original ERA5 dataset, ARCO also offers a reduced Gaussian grid [11] for its parameters. This could potentially slightly decrease the storage requirements and enhance computation speed. However, due to the complexity of the conversion process, we opt for the standard grid representation. This could be a future enhancement if such a feature becomes necessary for the aviation research community.

The efficiency of *FastMeteo* is primarily attributed to the innovative Zarr storage. It enables grid data to be stored and retrieved efficiently, even on basic personal computers. It obviates the need to load the entire dataset to extract specific data. This method is also more effective than relational or non-relational databases, as Zarr is specifically optimized for multi-dimensional data structures.

The performance bottleneck is the synchronization of local data from Google Cloud Storage. With a high-speed connection, it can take approximately 10 seconds to synchronize one hour of global data from Google ARCO-ERA5. However, once the data is available locally, subsequent interpolations are rapid.

In this paper, we focus on one use case of estimating contrails from flight data using *FastMeteo*. The application of this library can be broadened to additional aviation scenarios. For instance, with more precise wind data, we can more accurately estimate fuel consumption and emissions from flights by using a velocity closer to the true airspeed. We are also developing new programming interfaces to access a field of meteorological parameters, aiding the trajectory optimization applications.

7. Conclusion

In this study, we try to tackle the computational complexities associated with the incorporation of meteorological data into aviation studies. Two challenging tasks are identified: the process of downloading the vast ERA5 datasets from ECMWF and the computational demands associated with data interpolation. Addressing these issues, the *FastMeteo* Python library is introduced as an innovative solution. This tool, which can be seamlessly integrated into current research workflows, simplifies the acquisition and interpolation of meteorological data, offering remarkable computational efficiency.

The *FastMeteo* library can function either as a standalone Python library or as a client-server module. This adaptability ensures that users can tailor their data processing based on the dataset size and computational infrastructure available.

The OpenSky flight data is used to examine the tool's capability in contrail estimation, demonstrating the system's proficiency in fetching and assimilating complex meteorological data. The transition to ARCO-ERA5 datasets stored in the Zarr format on cloud platforms significantly optimized the access and utilization of ERA5 data.

Our benchmarks also reveal the computational efficiency of *FastMeteo*. The tool exhibits non-linear performance scaling concerning the number of data points, with a significant increase in efficiencies when processing larger datasets. This is attributed to the vectorized interpolation capabilities embedded in the underlying *xarray* library.

Author contributions

- Junzi Sun: Conceptualization, Methodology, Writing–Original draft, Data curation, Writing–Original draft, Writing – Review and Editing, Visualization
- Esther Roosenbrand: Conceptualization, Methodology, Writing–Original draft, Writing – Review and Editing, Visualization, Software

Funding statement

Open data statement

- Data used in the paper is shared at: https://github.com/junzis/fastmeteo/tree/main/paper
- It is also under the archive: https://doi.org/10.5281/zenodo.10226828.

Reproducibility statement

- Source code of *FastMeteo* is shared at: https://github.com/junzis/fastmeteo
- Code to reproduce the figures is shared at: https://github.com/junzis/fastmeteo/tree/main/paper
- All resources are archived at: https://doi.org/10.5281/zenodo.10226828

References

[1] Junzi Sun, Luis Basora, Xavier Olive, Martin Strohmeier, Matthias Schäfer, Ivan Martinovic, and Vincent Lenders. "OpenSky Report 2022: Evaluating Aviation Emissions Using Crowdsourced Open Flight Data". In: *Proceedings of the 40th Digital Avionics Systems Conference.* Sept. 2022.

[2] Junzi Sun. "OpenAP. top: Open Flight Trajectory Optimization for Air Transport and Sustainability Research". In: *Aerospace* 9.7 (July 2022), p. 383. DOI: 10.3390/aerospace9070383.

[3] Ulrich Schumann. "On conditions for contrail formation from aircraft exhausts". In: *Meteorologische Zeitschrift* 5 (1996), pp. 4–23.

[4] Matthias Schäfer, Martin Strohmeier, Vincent Lenders, Ivan Martinovic, and Matthias Wilhelm. "Bringing up OpenSky: A large-scale ADS-B sensor network for research". In: *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks.* IEEE. 2014, pp. 83–94.

[5] H Hersbach, B Bell, P Berrisford, G Biavati, A Horányi, J Muñoz Sabater, J Nicolas, C Peubey, R Radu, and I Rozum. *ERA5 hourly data on pressure levels from 1940 to present. Copernicus Climate Change Service (C3S) Climate Data Store (CDS).* 2018. DOI: 10.24381/cds.bd0915c6.

[6] Charles Stern, Ryan Abernathey, Joseph Hamman, Rachel Wegener, Chiara Lepore, Sean Harkins, and Alexander Merose. "Pangeo Forge: Crowdsourcing Analysis-Ready, Cloud Optimized Data Production". In: *Frontiers in Climate* 3 (2022), p. 782909.

[7] Taylor A Gowan, John D Horel, Alexander A Jacques, and Adair Kovac. "Using cloud computing to analyze model output archived in Zarr format". In: *Journal of Atmospheric and Oceanic Technology* 39.4 (2022), pp. 449–462.

[8] Ulrich Schumann. "Formation, properties and climatic effects of contrails". In: *Comptes Rendus Physique* 6.4-5 (2005), pp. 549–565.

[9] Rebecca Dischl, Stefan Kaufmann, and Christiane Voigt. "Regional and Seasonal Dependence of the Potential Contrail Cover and the Potential Contrail Cirrus Cover over Europe". In: *Aerospace* 9.9 (2022), p. 485.

[10]   Daniel M Murphy and Thomas Koop. "Review of the vapour pressures of ice and supercooled water for atmospheric applications". In: *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography* 131.608 (2005), pp. 1539–1565.

[11]   Mariano Hortal and AJ Simmons. "Use of reduced Gaussian grids in spectral models". In: *Monthly Weather Review* 119.4 (1991), pp. 1057–1074.