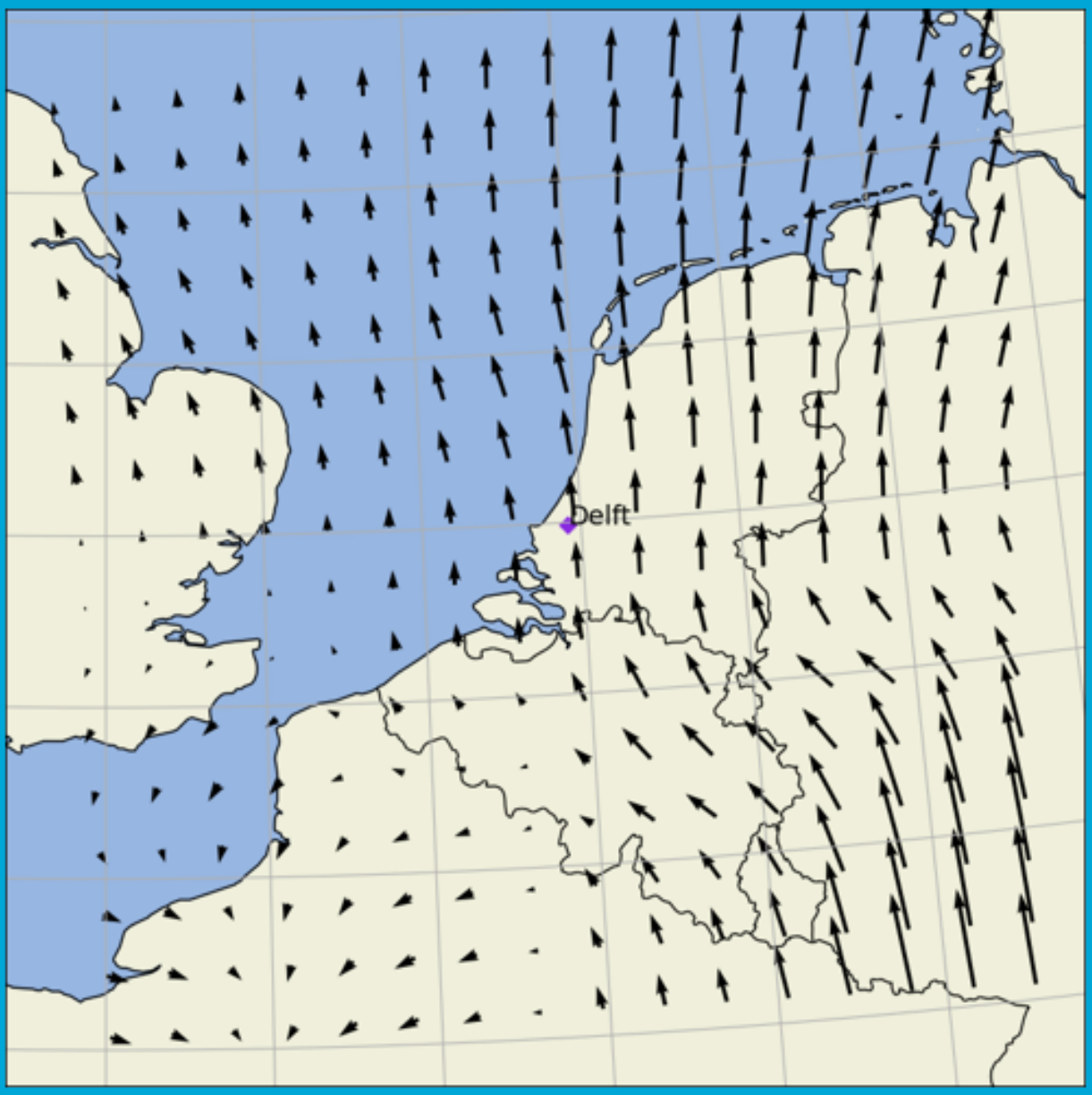
MSc Thesis

Estimating Wind Fields Using Physically Inspired Neural Networks With Aircraft Surveillance Data

AE5310: Thesis Control and Operations

J.M.L. Malfliet



Delft University of Technology

TU Delft

This page is intentionally left blank

MSc Thesis

Estimating Wind Fields Using Physically Inspired Neural Networks With Aircraft Surveillance Data

by

J.M.L. Malfliet

Thesis Defence: 26th of January 2023

Student Name	Student Number
Jari Malfliet	4668626

Thesis Committee: J.M. Hoekstra (Chair)
J. Sun (Supervisor)
A. Bombelli

Institution: Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands

TU Delft

Preface

This thesis is concerned with wind estimation derived from aircraft surveillance data. Wind knowledge can improve flight safety, flight planning and efficiency. This research continues on previous work, by exploring the possibility of a neural network approach to the wind nowcasting/forecasting task, based on aircraft surveillance inputs. The results are shared in this report.

I would like to thank Junzi Sun for his guidance during this project, and for providing the opportunity and necessary materials to conduct my research. I also would like to thank Jacco Hoekstra for his valuable feedback during the preliminary and green-light meetings. At times, the thesis project was challenging and difficult, yet they helped me in succeeding and writing this final report.

Contents

Part I

Scientific Paper

Estimating Wind Fields Using Physically Inspired Neural Networks With Aircraft Surveillance Data

Jari Malfliet
Control & Simulation, Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands

***Abstract*—Wind estimation plays an important role in many aspects of our world, both for nowcasting and forecasting. This is also true for aircraft flight routes; where wind field knowledge can contribute to flight planning and safety. Aircraft can relay flight information back to ground-based receivers, from which local wind measurements can be derived. Such measurements can be used in interpolation or a meteo-particle model to produce estimates elsewhere; yet those means do not take into account wind dynamics. A physically inspired neural network approach is applied in this work. Flight paths are derived from the aircraft measurements, and used to simulate training data from ECMWF ERA5 reanalysis data. Next, actual aircraft measurements serve as input, with the goal of predicting an entire wind field over the Netherlands and surrounding areas at cruise altitudes. The network can be leveraged by introducing physical losses, which are found to smooth the predicted flows. The network is able to predict flow fields on both simulated and real measurements, given sufficient input data. It improves over the existing methods of relying on 6-hour forecasts or using the meteo-particle model. Magnitude error of wind is reduced to 2.85 m/s, coming from 3.88 m/s for 6-hour GFS forecasts and 4.78 m/s for the Meteo-Particle model. Directional, physically inspired networks have an error of 11.2 degrees, compared to 14.4 and 17.3 degrees for 6-hour GFS forecasts and Meteo-Particle model, respectively. However, both error metrics fluctuate significantly depending on whether flow is uniform or of divergent nature at a particular day. The network is fit for nowcasting and future work can be done for longer-term forecasting.**

***Index Terms*—meteorology, wind nowcasting, deep learning, aircraft meteorological monitoring**

I. Introduction

Knowledge of wind fields can be utilized in aircraft flight planning for safety, travel speed and/or emission related reasons. Rivias et al. [5] mention the importance of local wind knowledge for conflict resolution; their conflict resolution method can take into account a probabilistic wind range and propose a solution. Conflict resolution is valid when aircraft are close, yet a conflict at increased distance becomes more difficult due to wind uncertainty [5]. Also, fuel consumption can be reduced by taking wind into account when planning aircraft trajectories, possibly allowing a reduce in cost and environmental impact, as well as improved estimation of travel and arrival times, as indicated by Legrand [16]. Typically, aircraft operators receive weather forecasts via ensemble predictions [16] to plan trajectories, while Air Traffic Services give updates [32]. An ensemble prediction is a collection of so-called ensemble members. Each member is installed with different initial conditions and boundary conditions, in order to better reflect the uncertainty of numerical weather predictions (NWP's) for a particular forecast [11]. As such, a probabilistic approach can be taken to aircraft route planning, instead of relying on one single forecast, which can deviate significantly from reality, say Steiner et al [30]. This is because the atmosphere is a nonlinear dynamic system, with discrepancies in both measured states and numerical solvers [17]. A distinction is made between forecast and analysis. A forecast predicts a future state of the atmosphere by propagating an initial state [23]. An analysis on the other hand provides a representation of the atmosphere in grid form, constructed from different measurements. A reanalysis is an analysis where the used software is fixed over a range of dates for which the reanalysis happens [23]. Analysis/reanalysis use a process called data assimilation, to combine observations with short-term forecasts in order to reach the best estimate possible of an atmospheric state [2]. Aircraft observations are incorporated in such models. The European Centre for Medium Range Weather Forecasting, ECMWF, also provides ensembles on the reanalysis to cope with uncertainties [2]. In data assimilation, two techniques are identified: 3DVAR and 4DVAR. Both techniques involve complex mathematical optimizations of cost functions [1], with 4DVAR adding a temporal dimension [20] while 3DVAR does not take observation time into account [19]. Also, such NWP models can be over-smoothed and can lack local variations [32]. To address these issues, it is possible to use real-time broadcast aircraft data to determine meteorological conditions and achieve better wind estimation in the atmosphere. Research in this topic is already established.

Techniques do exist to extract local wind measurements, which can be used to provide improved nowcasting and forecasting. Multiple ways have been developed. Sun et al. [32] identify two categories, the first one being estimation via trajectory observation from the ground, the second category being estimating wind from broadcast aircraft data. De Haan and Stoffelen [4] applied a 3DVAR High Resolution Limited Area Model (HIRLAM) to assimilate wind observations. The assimilation cycle was set to one hour, and compared to a regular 3hour cycle. An RMSE decrease of 5% was found [4]. A more recent development is the information provided by ADS-B, short for automatic dependent surveillance-broadcast. De Leege and van Paassen [15] propose using this data down-link to derive meteorological conditions.

Ensemble forecasts also allow for better nowcasting capabilities: Kikuchi et al. [11] propose a particle filter approach which selects the most likely ensemble

member/members given a set of aircraft observations. Such ensemble forecasts are provided by weather services such as the European Centre for Medium-Range Weather Forecasts, or Global Forecast System (GFS, National Weather Service). Sun et al. [32] introduced a Lagrangian transportation model, called the Meteo-Particle model (MP model), where measurements, derived from ADS-B and Mode-S, are initiated as particles and spread through the airspace, in order to estimate wind in non-measured areas [32]. The distribution is covered by a random Gaussian walk process, with a wind bias term. Every wind particle contains information on its origin and age, which allow for a confidence level to be assigned to each particle. A wind estimate for a certain point is calculated by weight averaging nearby particles, in real-time. This allows for nowcasting. Short-term forecasting is performed by Gaussian processing of particles in a particular area [32]. Prior in the papers related project, a direct data assimilation scheme was developed but found to be unsuccessful. This scheme uses the PyDDA framework [9] [24] [25] and aims to assimilate measurements directly into a GFS forecast background. Assimilation was done through directly optimizing the mixed wind field for a number of cost functions. The optimizer could not find a physically realizable solution, any improvements are attributed to smoothing of nearby regions. The output of the MP model was also combined with GFS estimates in low confidence areas. This did improve over stand-alone MP or GFS estimation, but this is to be expected, and no real wind dynamics knowledge is added this way. Another identified approach to tackle the wind nowcasting problem is deep learning.

Deep learning has become a successful technique in many scientific applications, and fluid dynamics problems are no exception. According to Kutz [14], this can be attributed to the increase in computational strength, combined with more and more access to vast data sets, on which deep architectures can be trained. In fluid modelling, other techniques exist, such as proper orthogonal decomposition (POD) and dynamic mode decomposition (DMD) to capture the systems dynamics, but they are not able to capture flows which are transient, multi-scale, or are spatially transformed, as Kutz indicates in [14]. Drawbacks do exist in the form of large computational expense, and the black box behaviour of a neural network, i.e. the way how the network captures present features or modes, cannot be directly understood [14].

When applied to physical systems, two approaches can be taken for a deep neural network: a data-driven approach, or a physics-inspired one, say Rao et al. [26]. The first approach is based on obtaining both vast input and output data from high-fidelity simulations, and learning the mapping between input-output. Neural nets can approximate such mappings, even if high-dimensional [26]. Yet, the drawback is that such amounts of high quality data must be available and trained. Physically inspired neural nets (PINN) on the other hand can include physical regularizations, based on governing PDE's. Furthermore, such a network still retains the ability to capture physics which are not included in the physical regularizations [26]. Schweri et al. [28] employed such a strategy to reconstruct wind fields using satellite observations on cloud movements via a PINN. As such, wind in non-clouded areas can be estimated. Their research builds upon the foundations of image inpainting, where an occluded area of an image is filled in by using partial convolutional layers [18]. Likewise, gaps in physical flows can be filled, although one must pay attention to the fact that a physical flow field represents a dynamical process, not a pure image in itself [28]. The architecture of concern for inpainting tasks is an encoder/dense-blocks/decoder composition, with encoder and decoder adhering to the U-net structure of Ronneberger et al. [27]. The remainder of this work is constructed as follows. In Section II, the methodology is explained, from aircraft observation to wind field prediction. Next, Section III shows relevant results, and Section IV discusses the obtained knowledge, after which a conclusion is given in Section V.

II. Methodology

The methodology in this paper is mainly inspired from Schweri et al. [28] and aircraft meteorological monitoring by [15][32].

A. Aircraft observations

For this research, aircraft observations stem from ADS-B and Mode-S (selective interrogation) messages. ADS-B broadcasts information 2 times per second, automatically without the need for interrogation [32]. However, it only contains information on the aircraft's ground speed. To estimate wind speed, aircraft air speed must be subtracted from ground speed. This air speed can be derived though from Mode-S Comm-B replies [32]. Mode-S replies come with the difficulty that the broadcasting aircraft is only known to the interrogating radar. Sun et al. [31] developed a library, called PyModeS, to process both Mode-S Comm-B replies, and ADS-B data. Mode-S information is attributed to the correct aircraft. The following aircraft states are included, among others:

- Barometric altitude
- Ground speed (GS)
- True Airspeed (TAS)
- Indicated Airspeed (IAS)
- Mach number

Because the update rate of true airspeed information is low, true airspeed is also calculated from indicated airspeed/Mach number, requiring knowledge of the temperature. The latter can be calculated from pressure and ISA constants [32]. Once ground speed and true airspeed are known, together with information on aircraft heading and track angle, the wind vector, shown in Figure 1, is given by:

$$\vec{V}_{wind} = \vec{V}_{GS} - \vec{V}_{TAS} \tag{1}$$

In [15] and [32], a more elaborate description of wind vector estimation through ADS-B and Mode-S communication may be found.

Besides wind, another important state to consider is the barometric altitude. ADS-B transmits this altitude, which is

calculated from the measured pressure and the ISA datum of 1013.25 hPa [15]. Generally speaking, the barometric altitude is not equal to the geometric altitude. Aircraft which share the same barometric altitude, are on the same isobaric (constant pressure surface).

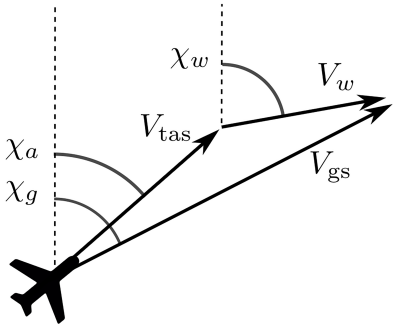


Fig. 1. **Wind vector in relation to GS, TAS vectors. Image courtesy of Sun et al. [32], https://doi.org/10.1371/journal.pone.0205029.g002**

B. Real-time Rejection

A rejection mechanism is incorporated to filter out any abnormalities in received data. Sun et al. [32] mention the probability of incorrect observations, due to either an error from measurement devices, or errors in the transmission process. Observations over $100m/s$ are rejected. Also, a real-time weighted average is kept of wind measurements per aircraft. The average is weighted with respect to time passed since the measurement. If a measurement exceeds this average by a factor of 1.5, and wind speed is above 10 m/s, it will be rejected. Additionally, duplicated measurements are filtered out as well. I.e. identical measurements which appear multiple times in the data-set.

C. Meteorological considerations

The research concerns both aeronautical and meteorological aspects. As wind is of particular concern here, some important considerations are mentioned in the following. Weather takes place as a multi-scale flow, both in a temporal and spatial sense [13]. Meteorology is a complex, widely researched scientific field. For this paper, a short timescale in the range of $< 1h$ and a scale extending to the range of the ADS-B receiver (i.e. a radius of 250NM [15]) are of interest. Also, 2D flow on flight levels is assumed. Those flight levels are isobarics, i.e. surfaces of constant pressure. Holton and Hakim [7] state that the continuity equation on such a surface equals :

$$(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y})_p + \frac{\partial \omega}{\partial p} = 0 \tag{2}$$

In which ω is the pressure change in direction of motion, $\frac{Dp}{Dt}$. This is the isobaric form of vertical motion [7]. The following assumptions hold:

- 2D flow on isobarics (thus, vertical motion ω is neglected in Equation 2)
- Flow is stable and non-turbulent at scale of minutes [32]. Also, specific weather phenomena such as convection, jet streams or thunderstorms are not taken into account.
- Flow is stable at a resolution scale of 10 km [32], i.e. the flow does not alter drastically over this distance.

D. Data Acquisition and Pre-Processing

Multiple data sets are used in the paper:

- processed ADS-B/Mode-S measurements [31] [32], from 01/01/2018 to 10/01/2018, at +/-30 minutes around 00h, 06h, 12h, and 18h for each day. [1]
- ECMWF ERA5 Reanalysis data [6]: Reanalysis fields were obtained from 2010 to 2021, the 1st and 15th of each month, at 00h and 12h, on pressure levels. The grid resolution is 0.25 degrees, both in longitudinal and latitudinal direction.
- GFS Forecast data [22]: 6-hour forecasts corresponding to the aircraft observation times mentioned earlier are collected, on pressure levels. The grid resolution is 0.25 degrees, both in longitudinal and latitudinal direction.

ECMWF and GFS data contain U-component (zonal, West-East), V-component (meridional, South-North) and geopotential height information of distinct pressure levels. The geopotential height of a pressure level usually differs from the geopotential height of that pressure level under ISA conditions. Aircraft measurements feature barometric altitude, based on pressure. To ensure valid matches, for each grid point of GFS and reanalysis data a spline is set up between isobaric pressure level data (U,V-components) and corresponding ISA height. As such, GFS and ECMWF sets are easily recalculated to any necessary barometric altitude. Panoply [29] was used to convert both GFS and ECMWF data into usable files.

E. Network Architecture

The applied architecture of the neural network is of crucial importance in addressing the wind estimation problem: Schweri et al. [28] propose a U-net like structure with physical losses and partial convolutions.

1) U-nets: The principle of U-net like architectures is the networks ability to localize features on different scales, as indicated by Ronneberger et al. [27] who apply it to biomedical image segmentation. First, an encoder is constructed, which is followed by a decoder. The encoder down-samples the input. The decoder has up-sampling operators between its layers to increase and arrive at the original resolution again [27]. The output from the encoder is directly fed to the corresponding decoder layer output. These so-called skip connections allow the extraction of features at each resolution [28]. For this reason, the U-net architecture is chosen for this work; atmospheric flow occurs at a multitude of scales [13]. Both local measurements and larger scale flows can be captured with this architecture. Schweri et al. [28] additionally implement dense blocks. Dense blocks are a collection of layers, where each layer uses the output of the preceding layer as input. According to Huang et al. [8],

[1] Sun J. Supplemental dataset for "Weather field reconstruction using aircraft surveillance data and a novel meteo-particle model"; 2018. Available from: https://doi.org/10.6084/m9.figshare.6970403.

such blocks aid the propagation and reuse of features, and help decrease the vanishing gradient problem. The features are also concatenated with each other this way. A simplified architecture is shown in Figure 2.

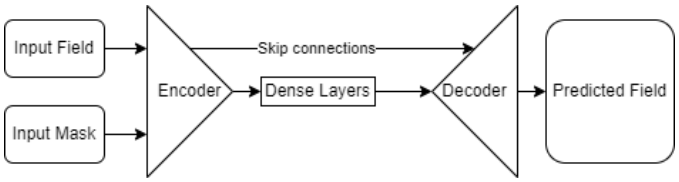


Fig. 2. **Simplified Network Architecture, a detailed architecture can be consulted in the appendix, Figure 36.**

2) Partial convolutions: A key component of inpainting networks such as used by Schweri, and adapted in this paper, is the partial convolutional layer. Usually, deep learning frameworks use convolutional layers: a kernel is moved over the input array, producing an output. A problem arises when certain values of the input array are unknown; one solution is to adapt a partial convolution from Liu et al. [18]:

$$x' = \begin{cases} C^T \cdot (X \cdot M)\frac{sum(L)}{sum(M)} + b & sum(M) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

and for the mask:

$$m' = \begin{cases} 1 & sum(M) > 0 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

X is the convolution window of the input field, while M is the accompanying mask ($= 1$ where observed). L is defined as a similar sized matrix filled with ones. The sum of the latter is divided by the amount of observed pixels (i.e. $sum(M)$), to arrive at a scaling factor [18]. C and b are the activation layers weight and bias, respectively. C is transposed to make the matrix multiplication output the required amount of channels. Should no observations be present in the convolution input (i.e. $sum(M) = 0$), the output will be zero. The mask is updated in similar fashion. If the input field has any amount of valid observed pixels, both x' and m' will eventually get filled after a sufficient amount of partial convolutions [18]. In Figure 3, this is exemplified for a mask; a partial convolution with size 2x2 and stride 2x2 is performed. This stride enables

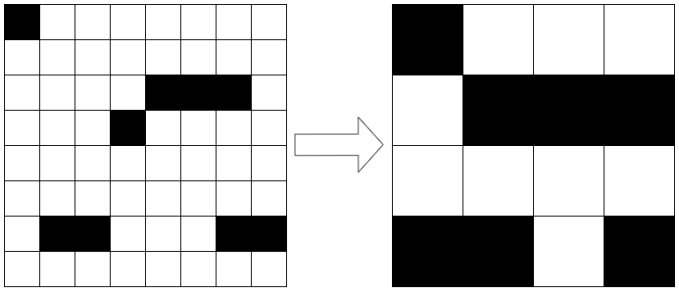


Fig. 3. **Example of a partial convolution for a mask (stride is 2x2)**

down-sampling. Up-sampling and concatenation are done in the decoder part, as shown in Figure 4. In the appendix, the full architecture is shown in Figure 36.

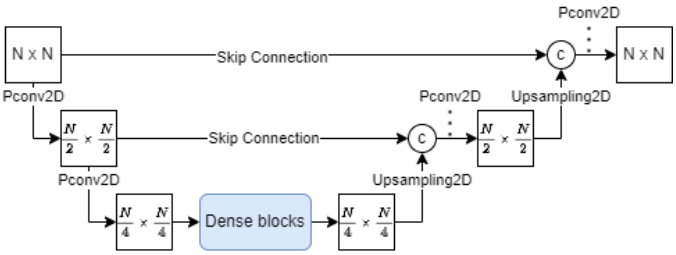


Fig. 4. **Simplified Network Architecture, a detailed architecture can be consulted in the appendix, Figure 36. Down-sampling occurs through the encoder, up-sampling operations in the decoder part bring back the original resolution. Skip connections allow for direct concatenation (c).**

3) Losses: In order to make the network aware of physical processes, the continuity requirement can be transformed into a loss function[7] [28]:

$$L_{div} = \Sigma(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}) \quad (5)$$

The network can learn vorticity as well, by adhering to an observational vorticity [28]:

$$L_{vort} = \Sigma(\nabla \times \vec{V}_{prediction} - \nabla \times \vec{V}_{truth}) \quad (6)$$

This term calculates the difference between the vorticity in the predicted field and the vorticity in the truth field. The network thus learns vorticity implicitly from reanalysis data.

Finally, the observational loss, being the difference between predicted and truth fields, is also considered:

$$L_{observational} = \Sigma(\vec{V}_{prediction} - \vec{V}_{truth})^2 \quad (7)$$

The final weighted loss function is given by:

$$L = \alpha_{vort} L_{vort} + \alpha_{div} L_{div} + L_{observational} \quad (8)$$

F. LeakyReLU Activation

A common pitfall in deep learning frameworks is the dying ReLU problem. This problem was encountered when using the regular ReLU activation function, which sets to zero for any negative output as explained by Xu et al. [33], and thus will produce zero-like outputs. Wind speeds in particular can be negative, which likely is the reason behind zero-wind outputs. A solution is implemented in the form of using a Leaky ReLU activation instead of a regular one. Leaky ReLU's allow a small gradient in their negative domain; avoiding the output getting stuck at zero during training (i.e. a 'dead neuron' [33]). The slope is only moderate, as the activation function must maintain its non-linearity. Both activation functions are illustrated in Figure 5.

G. Training and Validation setup

The networks objective is to be able to nowcast an (upper troposphere) wind field from down-linked aircraft measurements. As previously indicated, a real-time nowcasting algorithm was unsuccessful. An advantage of neural networks is to a priori train on known (labelled) data sets, resulting in a ready-to-use I/O system for real-time applications. It is

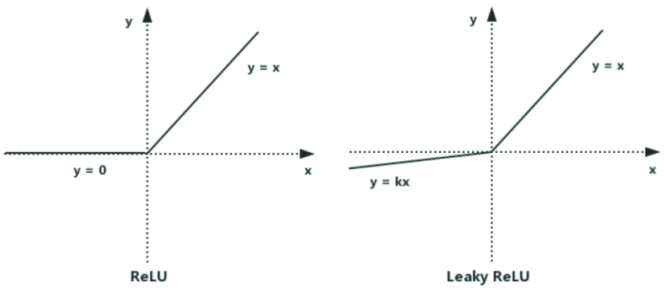


Fig. 5. **ReLU vs Leaky ReLU activation functions. Image courtesy of Xu et al. [33], 10.1109/ISCC50000.2020.9219587**

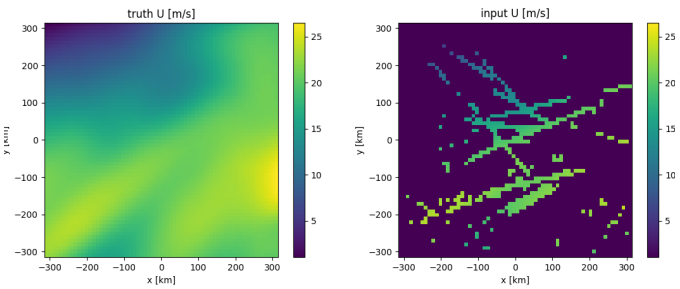


Fig. 7. **Example of mask generation, used for training. Shown left: U-component wind field, shown right: input field, derived from multiplication of truth field and observed flight routes.**

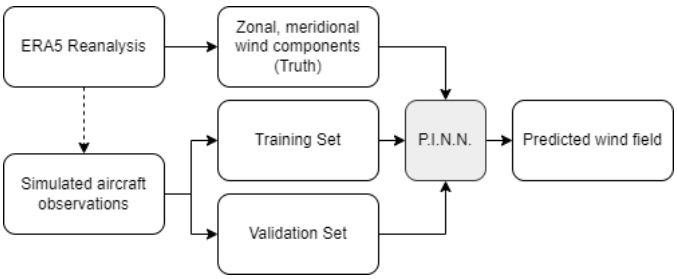


Fig. 8. **Training Setup**

decided to simulate aircraft measurements on reanalysis wind fields, via using actual flight tracks observed through Sun et al.'s ADS-B/Mode-S data [32].

The exact procedure, shown in a diagram in Figure 6, is as follows. First, a set of observations to be used is defined. Then, their precise (x,y,z) positions are reformed to a rectangular grid which serves as a mask: this 3D array equals 1 if the grid box concerned is being measured, and 0 otherwise. U (zonal) and V (meridional) components of the wind are extracted from the available ERA5 reanalysis data [6] and serve as ground truth. What follows is simple multiplication to arrive at a set of simulated measurements, serving as input fields. As the 1st of January 2018 is part of the real aircraft measurement set, this particular date is omitted from the training set. The flight mask set is augmented, by including rotations and mirroring. The masks are randomly distributed over the reanalysis truths, e.g. a reanalysis truth of e.g. 15/09/2015 can receive any mask from the 2018 flight measurements data set. An example is shown in Figure 7. Regarding the network I/O structure,

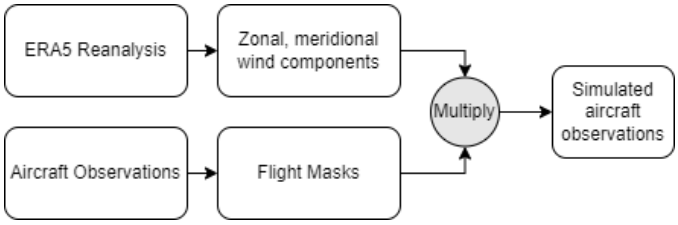


Fig. 6. **Flow chart of simulated observations data generation**

illustrated in Figure 8, simulated observations may be split into a training set and validation set. The network takes the full corresponding reanalysis fields as ground truths, to both train and validate against. The output shall be a predicted wind field (both U and V components). Two ways are considered to validate: using a portion of the simulated input as a validation sample, or second, by using the real flight data, which is the eventual objective of the project at hand. In the latter, a split is done in input data and validation data. The big difference between the two validation strategies is how network output is compared against truth. For reanalysis outputs, the entire output is compared to the entire truth field. For flight data, the output can only be compared to the chosen validation points. In addition, such validation points can also be checked against GFS forecasts and against MP model nowcasting [32]. It is important to note that input and validation data are split based on aircraft, not at random, in this project. This means that an aircraft is either in the input set, or in the validation set. This is illustrated in Figure 9.

Fig. 9. **Aircraft trajectories: training set (green) and test set (orange), as measured by an ADS-B receiver located at Delft, from 11:30 until 12:30 on 10/01/2018, FL 340-380 [32]. (via Cartopy [21])**

An exception is made when evaluating the network for

short-term forecasting. Then, data is split based on time. A last point to consider is the principle of cross-validation [14]. For each hour, multiple cases are evaluated, over which all data enters the validation set once. An input/validation split of 0.75 would imply four cross-validations. The cross-validation only applies to the second step of the setup due to computational constraints. Also, the main focus is on reconstructing flow fields from real aircraft data.

H. Adding GFS data

Although the GFS forecast is by no means exact, it provides the next best (a priori) guess, behind an actual meteorological measurement. In areas where no aircraft are present, it is therefore considered to augment the input to the physically inspired net with GFS data. Instead of combining GFS forecast and network output after prediction, the network receives a combined GFS-observation template. Two parameters are of concern: at which distance GFS augmenting starts (areas with no measurement) and how much points need to be filled.

I. Nowcasting and forecasting

The intent is to establish nowcasting and very-short term forecasting capabilities. Multiple decisions must be made on how to reach those. In terms of nowcasting, it is important to determine how far back in time training data can go, on which moments to make a prediction, and how to validate such a nowcast. It is decided to work with time windows of 10 minutes: every 10 minutes, the network will provide a prediction. To establish how the time history of observations play a role, a prediction is made based on the last 10, the last 20 and the last 30 minutes. The nowcast is tested by taking data in the 10 minutes ahead and 10 minutes past the evaluation time. The latter decision was also used by Stoffelen and De Haan [4]. Figure 10 illustrates this strategy.

For forecasting, training and validation split is based on time, not on aircraft. The same logic applies regarding history. The forecast is made from +10 minutes on-wards, to not cause data contamination as now data is split on time, not on aircraft. The strategy is shown in Figure 11.

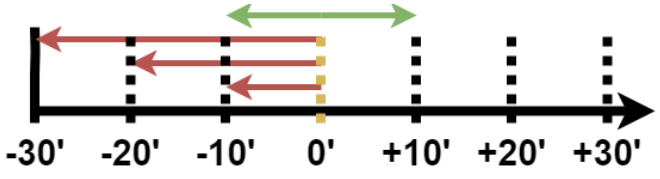


Fig. 10. **Nowcasting strategy. For a nowcast evaluation at the yellow timestamp, input field is constructed with training aircraft from red arrows. Nowcast performance is evaluated through test aircraft in green arrow range. (aircraft split)**

J. Adding GFS and extending time series

In order to find the right parameters for adding GFS and time series, a test case was performed. The case is evaluated on data from 01/01/2018, 05/01/2018 and 10/01/2018, at both

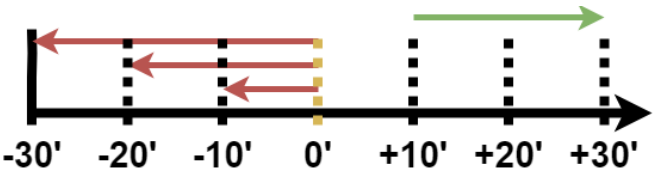


Fig. 11. **Short-term forecasting strategy. For a forecast evaluation at the yellow timestamp, input field is constructed with training data from red arrows. Forecast performance is evaluated through test data in green arrow range. (time split)**

TABLE I
EXTENSION OF TIME SERIES

	10 minutes	20 minutes	30 minutes
NC RMSE [m/s] (00h)	9.26	9.07	8.48
NC RMSE [m/s] (12h)	6.02	5.61	5.31
FC RMSE [m/s] (00h)	8.20	7.72	7.31
FC RMSE [m/s] (12h)	6.05	5.76	5.65

00h and 12h. The error metric used is RMSE, as explained in Equation 9. It is seen that taking in longer time series is beneficial towards RMSE. It is decided to evaluate nowcasts/forecasts with time series going back up to 30 minutes. At midnight, flights are very low in frequency, leading to a high RMSE. GFS can fill this gap, leading to improvements. It can be seen that a GFS fill starting at 100 km performs better than a fill starting at 50 km. This is due to the dissimilarity between GFS forecast and actual aircraft measurements; yet if no GFS is taken into account, the RMSE sharply increases as the network will produce predictions which are far off. In essence, for low-frequency flight hours, one must basically decide on a confidence region where measurements are preferred over GFS. An example of a night flight overlay is shown in Figure 12, for the 00h at 01/01/2018 at 37,000 ft. In Figure 22, in the appendix, wind predictions can be found for this input (please note U and V components use different colour scales). If no GFS infilling is used, the neural network is obviously going to drastically under-perform. If it is used, it succeeds in reconstructing the GFS field, while taking into account other wind speeds in measured areas (e.g. the southeast corner). One can question the added value of using the GFS-assisted neural network during low-frequency flying hours. During high frequency, the best scores are achieved for no GFS infilling. As such, combining GFS and measurements is concluded to be ineffective.

K. Vertical and horizontal resolutions

ADS-B/Mode-S measurements collected by Sun et al. [32] give precise information on the coordinates of the measurements; however, the neural network only accepts grid data

TABLE II
GFS INFILLING

	50km	100km	no GFS
RMSE (00h)	8.83	8.48	13.93
RMSE (12h)	5.49	5.31	5.24
FC RMSE (00h)	7.84	7.31	14.81
FC RMSE (12h)	5.76	5.65	5.61

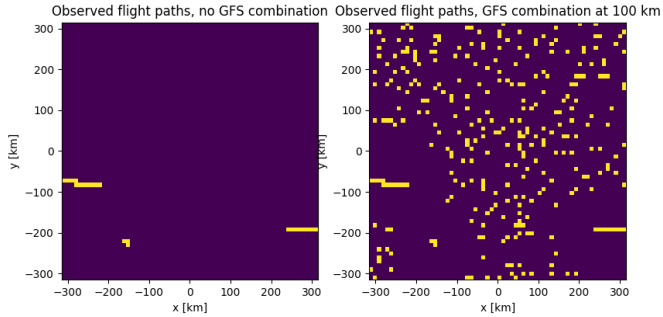


Fig. 12. **Example of GFS infilling during nighttime. 00h at 01/01/2018 at 37,000 ft.**

as input. The resolution of the grid will have an impact on the training duration of the network. It is decided to establish a rectangular grid with 10 km horizontal resolution, with borders at 320 kilometers to the North, East, South and West from the ADS-B receiver. The vertical scale is divided in 2D isobarics at multiples of 1000 ft (barometric altitudes). Aircraft routes have their cruising height at these multiples as well, as can be seen from Figure 13.

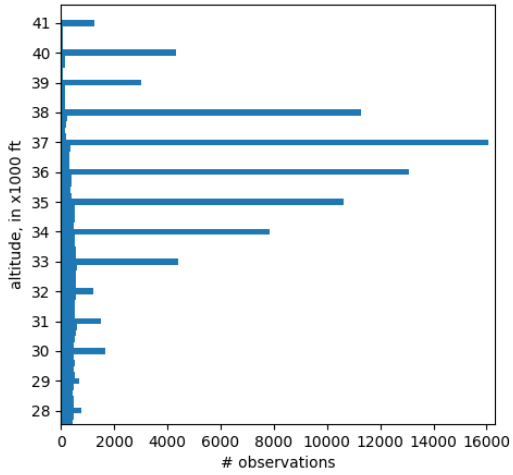


Fig. 13. **Altitude histogram, 01/01/2018, 12h**

To initialize aircraft measurements on the 3D grid (i.e. input grids to the PINN), the measurements of the preceding 10 minutes are averaged on the grid. Regarding vertical levels, measurements 500ft up and down are taken. The previous 10 minutes will fill up any open spots, and lastly, the earliest 10 minutes (30-20 minutes before nowcast) fill up any remaining open spots.

L. Error Metrics

In order to give valid answers on the projects objective, proper error metrics are defined which allow comparison between different methods. A root mean square error (RMSE) may be determined via:

$$RMSE = \sqrt{\frac{\Sigma[(u_{prediction} - u_{obs})^2 + (v_{prediction} - v_{obs})^2]}{n_{obs}}} \tag{9}$$

A difference in wind magnitude is simply computed as the absolute difference:

$$\Delta V_{magn} = |\vec{V}_{prediction} - \vec{V}_{obs}| \tag{10}$$

A difference in wind angle can be calculated by rearranging the dot product formula for two vectors:

$$\Delta\theta_{dir} = \arccos \frac{\vec{V}_{prediction} \cdot \vec{V}_{obs}}{|\vec{V}_{prediction}| \cdot |\vec{V}_{obs}|} \tag{11}$$

The RMSE error is able to express both magnitude and directional error in one metric.

M. Training and GPU

Training was performed using an Nvidia Quadro M1200 GPU. The batch size for stochastic gradient descent is fixed at 32, with the epochs consisting of n steps, n being the training set size divided by the batch size of 32. The optimizer used is Adam [12]. Learning rate scheduling is performed. In terms of training time, a batch of 32 wind field pairs takes 0.8 seconds to be trained. for 512 samples, the number of steps per epoch is thus $\frac{512}{32} = 16$, resulting in a training time of 13 seconds per epoch. When training in 100 epochs, the total time is 1300 seconds. The resulting wind fields require storage in the order of gigabytes.

III. Results

Regarding application to nowcasting and forecasting on aircraft broadcast data, only cruising altitude is considered. Most flights are located here; cruise flight is often the longest and most crucial phase, also from an environmental perspective.

A. Hyper-parameter tuning

This projects objective is not only to establish whether wind nowcasting via neural networks (NN) is possible, but also to establish if a 'PINN' can provide further improvements. The physical losses, being divergence and vorticity, can be attributed a weight in the final loss function. When zero, the network is not aware of the physical knowledge related to the loss function. The influence of the weights are determined by a hyper-parameter search, on the real measurement validation set.

The hyper-search uses the 12h measurement data-set, of each day (01/01/2018 until 10/01/2018), evaluated for nowcasting performance, at levels from 34,000 up until 38,000 ft in steps of 1,000 ft. To start, the divergence weight is increased from 0 up to 1000, with 10 and 100 in between. In Figure 14, a line is plotted showing magnitude error and directional error against the divergence weight, for all 10 days separately. In Figure 15, this procedure is repeated again for the vorticity weight. Regarding divergence weight (Figure 14), it can be seen that the relation between error and weight strongly depends on the day itself; the line with circular dots represents the average across all days. From this line, it can be seen there is a slight drop for a divergence weight of 10 for magnitude error. A low is also reached

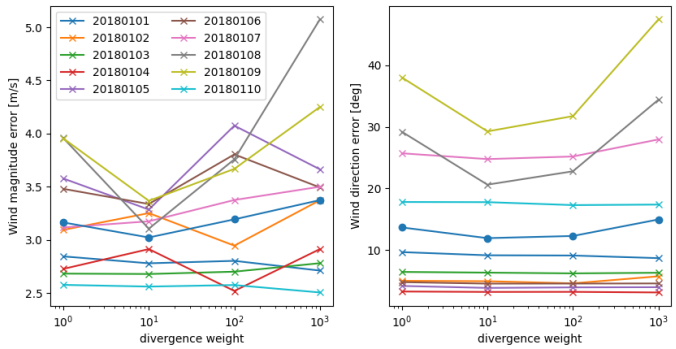


Fig. 14. **Error metrics for altering divergence weights, plotted for each day. circular dotted line is average over all days.**

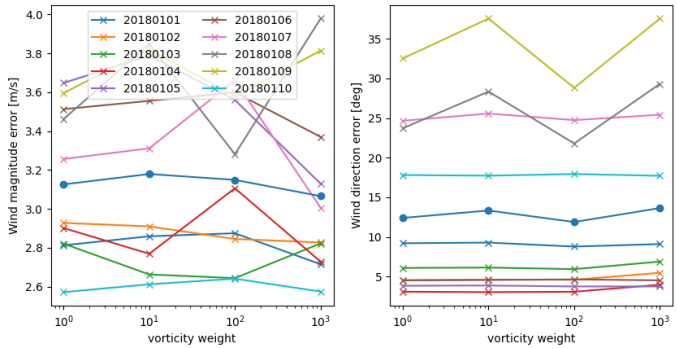


Fig. 15. **Error metrics for altering vorticity weights, plotted for each day. circular dotted line is average over all days.**

on the directional error plot. Yet, it is decided the plots are inconclusive in indicating whether divergence losses can improve performance, as behaviour differs day to day. For vorticity weight (Figure 15), the same conclusion can be drawn. The averages across all days barely change, and again behaviour depends on the evaluated day. In Figures 23 and 24 in the appendix, example velocity fields are plotted, both for zero weight cases, and for vorticity or divergence at a 1000. One can see both physical losses smooth the flow speed, yet do not offer a drastically different solution compared to the regular neural network with no losses. It is decided to continue with a divergence weight of 10, but vorticity is left out of the network, and is set to zero.

B. Nowcasting and Forecasting Windows

With the determined hyper-parameter set, the network is trained on 512 samples, for a 100 epochs, with batch size of 32. The test again consists of the 12h data collection, across all days, for 34,000 to 38,000 ft, in steps of a 1000 ft. Magnitude error and directional error are used to evaluate performance. A nowcasting comparison is made between the PINN, the 6 hour GFS forecast, and the MP model. The PINN makes use of up to half an hour of measurements, as previously explained. It is important to note that the MP model will also estimate beyond its confidence zone. The nowcasting comparison makes use of separation of training aircraft, and test aircraft, in a cross-validation set-up. In Figure 16, a box-plot is shown, collecting data over all days. Figure 17 analogously shows directional error. For magnitude error, outliers are shown until 100 m/s. All vertical scales are logarithmic. Performance for individual

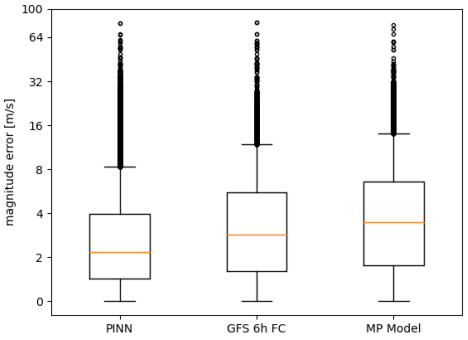


Fig. 16. **Magnitude error box-plot nowcasting, y-scale is logarithmic.**

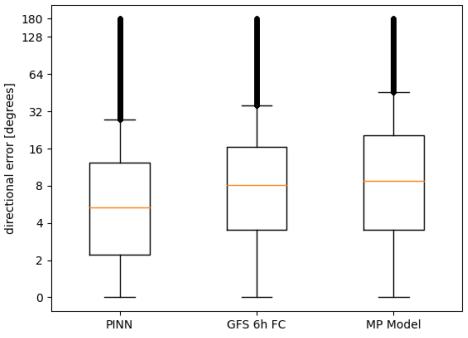


Fig. 17. **Directional error box-plot nowcasting, y-scale is logarithmic.**

days is shown in Figures 25 and 26, in the appendix. Next, Table III and IV show the means of the magnitude and directional error, respectively. A total error is also included, averaged over all days. It is clear that the PINN outperforms both the GFS forecast, and the MP model on both error metrics.

Forecasting performance is also evaluated, the PINN results can be compared against the 6 hour GFS forecasts. Figures 18 and 19 show relevant box plots, with individual days shown in Figures 27 and 28 in the appendix. Table V and VI show averaged results for each day. From the presented results, it is clear that the PINN is able to outperform the GFS forecast, and the Meteo-Particle model. In nowcasting configuration, the PINN achieves an improvement of 40% compared to the MP model, and 27% compared to the GFS forecast, for the magnitude error. For directional error, an improvement of 37 % and 22% is recorded over MP model and GFS forecast respectively. For short-term forecasting, an improvement over the 6 hour GFS forecast of 23% and 20% is noticed for magnitude and directional error, respectively.

TABLE III
NOWCASTING COMPARATIVE RESULTS: MAGNITUDE ERROR [M/S]

	01/01	02/01	03/01	04/01	05/01	06/01	07/01	08/01	09/01	10/01	Total
PINN	2.76	2.74	2.62	2.51	3.20	3.36	3.18	2.84	3.10	2.34	2.85
GFS 6h	3.79	5.30	2.60	4.10	5.18	4.25	3.85	3.59	3.29	3.10	3.88
MP Model	4.36	6.62	3.81	4.8	5.66	5.56	5.74	4.16	3.82	3.48	4.78

TABLE IV
NOWCASTING COMPARATIVE RESULTS: DIRECTIONAL ERROR [DEG]

	01/01	02/01	03/01	04/01	05/01	06/01	07/01	08/01	09/01	10/01	Total
PINN	9.2	4.3	5.7	3.0	3.6	4.6	24.5	18.9	26.0	17.0	11.2
GFS 6h	10.7	7.0	8.2	4.7	7.2	5.4	25.5	23.8	35.66	21.8	14.4
MP Model	13.2	7.0	9.7	4.3	6.6	6.3	36.2	29.7	40.3	26.3	17.3

TABLE V
FORECASTING COMPARATIVE RESULTS: MAGNITUDE ERROR [M/S]

	01/01	02/01	03/01	04/01	05/01	06/01	07/01	08/01	09/01	10/01	Total
PINN	2.95	3.35	2.62	2.54	3.92	3.18	3.16	2.87	2.96	2.36	2.97
GFS 6h	3.81	6.05	2.55	3.89	5.27	4.08	3.71	3.61	2.92	2.89	3.85

TABLE VI
FORECASTING COMPARATIVE RESULTS: DIRECTIONAL ERROR [DEG]

	01/01	02/01	03/01	04/01	05/01	06/01	07/01	08/01	09/01	10/01	Total
PINN	8.5	4.3	6.9	3.9	4.8	4.4	23.4	20.1	32.9	20.0	12.4
GFS 6h	10.3	6.5	9.2	5.2	7.6	5.6	25.9	25.1	43.5	23.1	15.5

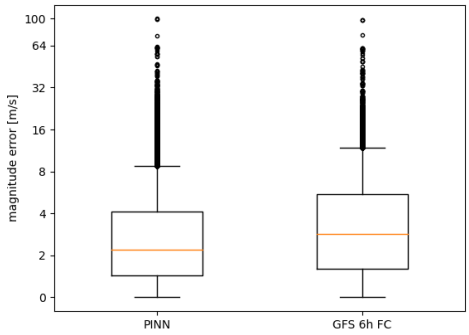


Fig. 18. **Magnitude error box-plot, forecasting, y-scale is logarithmic.**

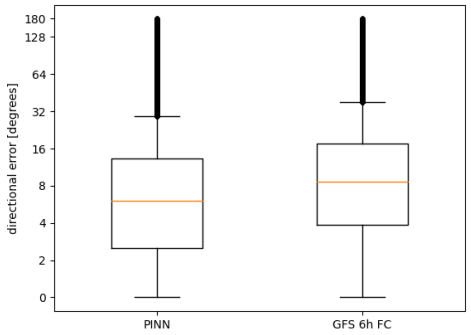


Fig. 19. **Directional error box-plot, forecasting, y-scale is logarithmic.**

Another, equally important observation to be made, is that directional errors are large for the last 4 days of the test set, namely 07/01-10/01. To investigate this result, the wind field of 01/01 (Figure 20) is compared to the wind field of 09/01 (Figure 21). Both PINN prediction, and the forecast GFS field are displayed. Note that the GFS field is a forecast, and is not a fully exact representation of the real atmospheric flow in itself. For 09/01, it can be seen clearly that both PINN and GFS field undergo major wind divergence/convergence in the southern half. On the contrary, the same fields for 01/01 are relatively uniform, with wind blowing towards the East. U and V components of the field are shown in the appendix: Figures 29, 30 show components for 01/01 at 34,000 and 36,000 ft respectively; Figures 31, 32 show components for 09/01 at 34,000 and 36,000 ft respectively.

These figures are displayed for two different barometric altitudes to display the differences between them. In the current network configuration, 2D levels of 1,000 ft are considered, and levels do not convey information to other levels. Flow will differ between levels, yet if no measurements are available for a certain level in a certain area, while a neighbouring level does have information, future work might look into conveying this information. Due to the plentiful amount of data for the 12h moments considered here, the added benefit of doing so will not have a large impact. In the figures considered, both components are in the same order of magnitude across the altitudes.

C. Reanalysis sets

The network is trained on simulated observations from reanalysis sets [6]; part of the simulated observations can be

used to test and validate the network. The network was trained on 512 samples, out of a total of 1148 available templates (not all templates are used due to computational expense). on the 636 test templates, a magnitude error of 1.92 m/s, and a directional error of 6 degrees is found. An example of reanalysis wind field reconstruction can be found in the appendix, Figures 33 and 34. The first figure, Figure 33, gives an estimate given sparse observations for the reanalysis of 01/07/2019 (12h at 38,000 ft), while the second, Figure 34, estimate is made with a denser input constructed with reanalysis data from the same day and hour, but with a different flight mask (both derived from actual ADS-B/Mode-S measurements). A denser input leads to better results.

IV. Discussion

A. Problem statement and goals

The main objective is to establish whether improvements in wind nowcasting (and short-term forecasting) are possible, by using a physics-inspired neural network. The methodology applied, described in Section II, is to train a U-net with skip connections, on simulated wind observations, potentially through physical losses. The network is then tested with real ADS-B/Mode-S measurements [32], and is able to produce nowcast fields.

B. infilling

The proposed methodology is capable of infilling the ERA5 reanalysis sets provided by ECMWF [6], as confirmed in Figures 33 and 34. The rightmost subplots display the absolute error between truth and prediction for both components. By visual inspection, it can be easily verified that areas farther away from measurements have higher errors. The network is only able to reconstruct the flow field when given a sufficiently populated input field. The southwest region in Figure 33 is not observed, and errors rise to an order of 10 m/s. However, when sufficiently populated as in Figure 34, the network is able to give a reconstruction with component errors not exceeding 5 m/s. This result validates the network.

Yet, such reanalysis fields are smooth NWP solutions. Aircraft observations are real world measurements, containing noise and errors due to sensors and transmission, and potentially being subject to turbulence. In Section II, a procedure is proposed to fill in missing gaps with GFS data; the network is then able to reconstruct the GFS field, and will draw an average between GFS data and aircraft data in the area between GFS points and aircraft points. The added value of this approach is questionable, as one simply averages out two solutions. A confidence zone could be established: within, observations are followed, outside, GFS forecasts or an interpolation between GFS and observations are followed. A GFS infilling zone of 100 km could be proposed. It is concluded that the PINN approach is best applied to regions/times of day with a high frequency in flights. Obviously, the PINN also is of most benefit in these conditions.

C. Nowcasting and forecasting performance

Both nowcasting and forecasting performance of the PINN showed significant improvements over alternatives such as the Meteo-Particle model and GFS forecast. This is true for both uniform flow fields (e.g. Figure 20) and diverging/converging flow fields (e.g. Figure 21). Full results can be reviewed in Section III. In the following, potential explanations for the recorded results are explored.

1) Neural Networks: The network architecture used, based on [28], [27] and [18], is a powerful method with already established performance in wind reconstruction by Schweri et al [28]. Deep neural networks in general are known as universal approximators [28], and can learn high-level features by breaking them down to simple representations [3]. They also perform well for interpolation tasks [26]. Skip connections are effective in allowing the network to extract feature at all scales: test runs with no skip connections do not show the same level of locality, as illustrated by Figure 35, in the appendix.

The contribution of proposed physical losses is less clear: two losses are considered, divergence and observed vorticity. For vorticity, the results were inconclusive. For divergence loss, a small drop was noticed for a weight of 10, yet no drastic changes occur. Overall, both losses smooth the flow fields, which is to be expected. Such smoothing can improve or worsen results by coincidence. That is not to say the network does not learn vorticity or divergence/convergence: neural networks are still able to discover underlying physics by themselves [26]. As the network is trained on reanalysis data-sets, it will also implicitly learn the physics of such reanalysis data. They are complex, high fidelity NWP's, with the drawback that they are smooth and averaged out in time and space [32].

2) Meteo-Particle model: A significant improvement over the Meteo-Particle model is found in Section III: magnitude error decreased by 40%, directional error by 27% for nowcasting. It is important to again repeat the MP model is asked to go beyond its confidence zone, and that aircraft are split in training and test aircraft when considering these results. However, the same conditions apply to the PINN, and GFS forecasts. In the following, a line of reasoning is put forward as to why the neural network outperforms the MP model.

The principle is to reconstruct a wind field from aircraft measurements; the MP model tries to imitate a Lagrangian transport model by introducing a Gaussian walk which is biased according to the measured wind, and spreads particles throughout the airspace [32]. When tasked with nowcasting, the MP model draws a weighted average of nearby particles. An average of the closest wind values does not necessarily constitute a correct answer: atmospheric flow is governed by nonlinear dynamics and adding up solutions averaged from other areas/timestamps is not guaranteed to produce correct

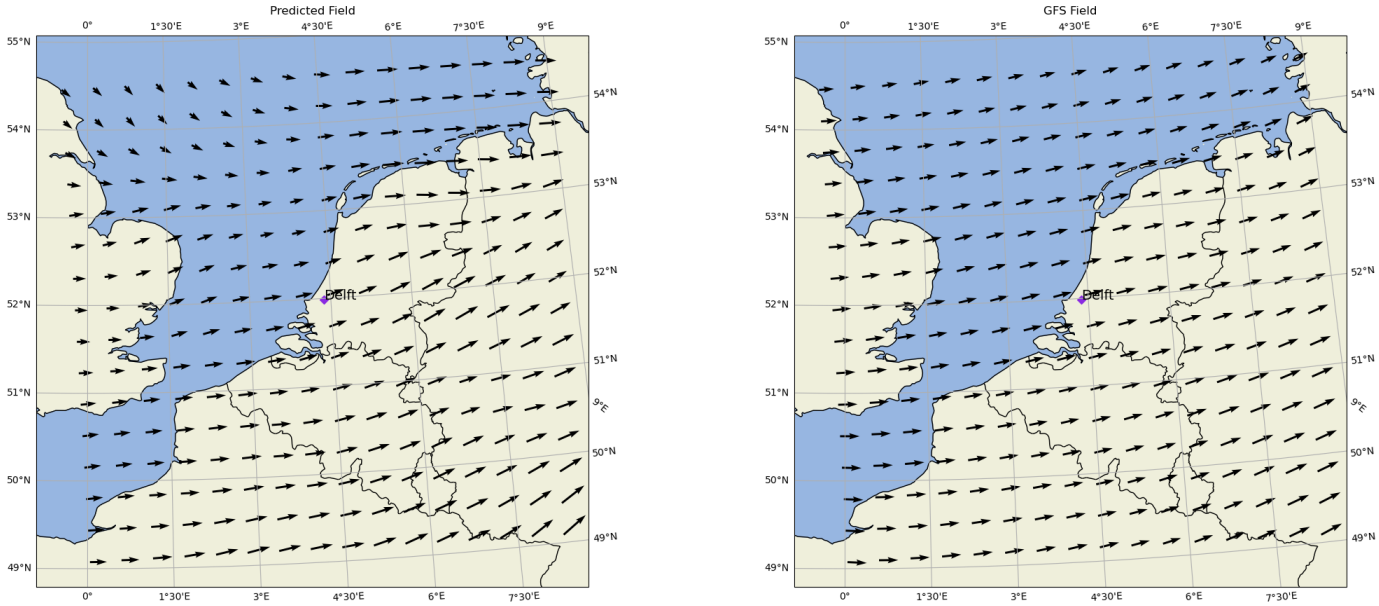


Fig. 20. **Wind fields for 01/01/2018, 12h, at 36,000 ft. Left: PINN prediction, right: GFS 6h forecast. Uniform flow is observed.**

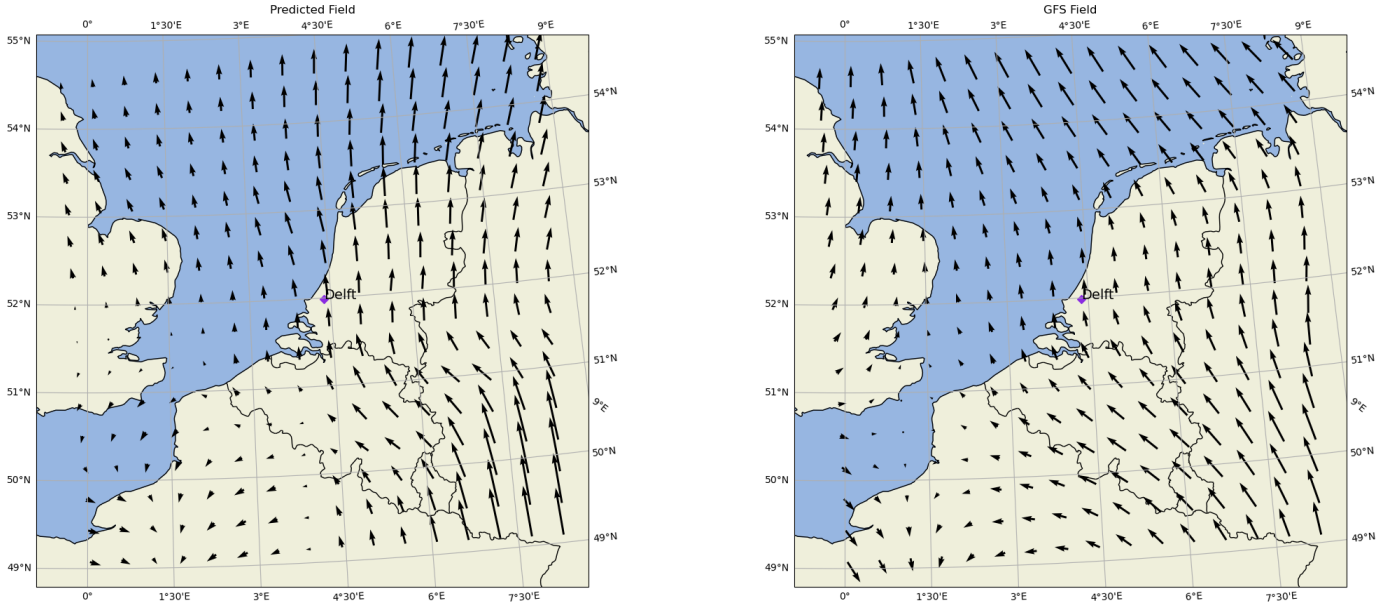


Fig. 21. **Wind fields for 09/01/2018, 12h, at 36,000 ft. Left: PINN prediction, right: GFS 6h forecast. Diverging flow is observed over northern France.**

results. In highly convergent/divergent flow conditions, the mean directional error for the MP model was found to be in the order of up to 40 degrees. In more uniform fields, the MP model is able to produce nowcasts, but also not at the level the PINN is able to do.

D. Advantages and disadvantages

A clear advantage of using the PINN architecture is that it outperforms the MP model in its current form. It can be trained on a priori data to arrive at a fast I/O system. A disadvantage is its black box behaviour [14], it is not truly known what goes on inside the net to produce its predictions. This also implies unpredictability; it might occur that the neural network outputs an entirely wrong prediction, negatively impacting flight efficiency or even safety, e.g. in case wind nowcasts are incorporated in proposing conflict resolutions. In its current form, it can only capture atmospheric dynamics partially and has little to no predicting power over jet streams, storms or strong vertical winds. This is however true also for the MP model in its blind spots. To fill these gaps, ground based monitoring, forecasts and e.g. aircraft weather radars remain essential. As Schweri et al. [28] mention, a set of observations can be explained by multiple flow fields. Large gaps thus pose a large ambiguity.

E. Assumptions

The most important assumption in this work, is 2D flow on isobarics. This ignores any vertical wind component across isobaric levels, which can occur in reality. Flow was also assumed to be stable and non-turbulent, at the scale of minutes and 10 km. Any (extreme) weather event breaks this assumption. It is observed that when flow is diverging/converging, errors rise strongly for all three methods, indicating the added difficulty in estimating wind with such complexity of such flows. Also, a Cartesian (x,y,z)-grid is used by the network, with z expressing isobaric surfaces. The curvature of the Earth is neglected. GFS and ECMWF data were interpolated to Cartesian coordinates, while originally given in latitude-longitude coordinates.

F. Recommendations and Future Work

The network could be extended with a real temporal architecture, whereas it is only spatial right now. However, training data and strategies must be available as well. The ECMWF reanalysis sets [6] are provided hourly, and interpolation methods could simulate snapshots in between. The validity of this can be questioned as the reanalysis are averaged out [32]. As for ADS-B measurements, they could be collected for a longer duration of time. The main advantage of having a temporal dimension, is the ability to produce forecasts. As of now, the forecasts remain very basic in their construction, they are essentially nowcasts which are evaluated with future test data. Scientifically speaking, a forecast is a prediction propagated from a nowcast. As for network architecture, spatio-temporal solutions are available in the form of recurrent neural networks (RNN) or long short term memory (LSTM) [10]. Should enough ADS-B/Mode-S data become available, training can also be done directly on this type of data.

The PINN approach considered 2D flow on isobaric levels, with aircraft observations put on the nearest level. Obviously, the atmosphere is a continuum where 3D flow takes place, and flows on levels possibly interact with each other, or information from one level could be inferred to another. This would require a dimensional extension of the network. The network tested in this work also limits itself to cruise altitudes in a square airspace of 320x320 km above Delft, The Netherlands. First, the network may also be extended for wind estimation in lower flight levels, or in approach areas. The network could be trained with reanalysis sets at these heights. Second, in addition to flight level influence, the global location will also likely influence networks performance. It is possibly necessary to train the neural network with local reanalysis data, instead of relying on the current trained network, which is trained for the airspace of The Netherlands and neighbouring countries. The resolution of the grid could also be increased; now it stands at 10 kilometers due to computational limitations.

V. Conclusion

The overall aim of the thesis project described in this paper, is to establish whether or not physics knowledge of wind fields can be leveraged with aircraft wind observations to obtain improved estimates of the wind field. A previous PyDDA-inspired approach failed to reproduce physically realizable flow fields; it averaged measurements with a background GFS field, and optimization of wind components via cost functions was ineffective. Another approach in the form of physically inspired neural networks was implemented on the aircraft meteorological monitoring problem, and was found to provide significant improvements over an existing particle model and over 6 hour GFS forecasts. For nowcasting, improvements of 40% were found for magnitude error, and 27% for directional error, when compared to the MP model. The magnitude error is 2.85 m/s, directional error is 11.2 degrees, when using test sets from 01/01 to 10/01 at 12h (noon) at cruising altitudes 34,000-38,000 ft. Also short-term forecasts benefit from using such a network. Such improvements can be explained by the networks ability to implicitly learn flow features on all scales, and the good approximating characteristics neural nets posses in general [26] [28]. The network has a U-net architecture with skip connections, inspired by [27] and [28], and uses partial convolutions to fill up occluded areas [18].

Training was performed via observations simulated from ECMWF ERA5 reanalysis sets [6]: reanalysis flow fields are masked with actual observed flight paths to simulate such training data. The network is able to reconstruct flow patterns, both from simulated reanalysis inputs (unseen in training), and actual aircraft observations (made via ADS-B/Mode-S [31]), given there are sufficient inputs, spread throughout the airspace. The impact of physical loss functions (divergence, observed vorticity) was found to be the smoothing of predicted wind fields, but no drastic changes in flow configuration were observed. The application of a neural network to the wind estimation problem however can still be regarded as addition of physics knowledge, as networks are able to implicitly learn physics contained in their training data (reanalysis sets in this particular work).

Wind estimation in non-observed areas remains a difficult task, possibly due to ambiguity in the range of solutions [28]. In diverging/converging conditions, all three PINN, GFS and MP models suffer from worsening accuracy, certainly in directional error, yet the PINN still outperforms the other two methods. For future work, it is recommended to extend the network in the temporal dimension, and possibly add a vertical dimension as well. A temporal dimension will mainly benefit short-term forecasting, which for now remains a nowcast projected ahead in time.

ACKNOWLEDGMENT

The author wishes to thank Junzi Sun for his guidance and related work (Meteo-Particle Model, PyModeS, and ADS-B/Mode-S data) leading up to this paper. Acknowledgement is also given to the Climate Data Store. Hersbach et al. [6] was downloaded from the Copernicus Climate Change Service (C3S) Climate Data Store. The results are generated using Copernicus Climate Change Service information (2022). Neither the European Commission nor ECMWF is responsible for any use that may be made of the Copernicus information or data it contains.

REFERENCES

[1] D. M. Barker et al. "A Three-Dimensional Variational Data Assimilation System for MM5: Implementation and Initial Results". In: *Monthly Weather Review* 132.4 (2004), pp. 897–914. DOI: 10.1175/1520-0493(2004)132⟨0897:ATVDAS⟩2.0.CO;2.

[2] ECMWF. *Earth system data assimilation*. 2020. URL: https://www.ecmwf.int/sites/default/files/medialibrary/2020-05/ecmwf-fact-sheet-data-assimilation.pdf (visited on 12/06/2022).

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[4] S. de Haan and A. Stoffelen. "Assimilation of High-Resolution Mode-S Wind and Temperature Observations in a Regional NWP Model for Nowcasting Applications". In: *Weather and Forecasting* 27.4 (2012), pp. 918–937. DOI: 10.1175/WAF-D-11-00088.1.

[5] Eulalia Hernández-Romero, Alfonso Valenzuela, and Damián Rivas. "Probabilistic multi-aircraft conflict detection and resolution considering wind forecast uncertainty". In: *Aerospace Science and Technology* 105 (2020), p. 105973. ISSN: 1270-9638. DOI: https://doi.org/10.1016/j.ast.2020.105973. URL: https://www.sciencedirect.com/science/article/pii/S1270963820306556.

[6] H. Hersbach et al. *ERA5 hourly data on pressure levels from 1959 to present*. Data collected throughout November 2022, last Accessed on 18-11-2022. 2018. URL: https://doi.org/10.24381/cds.bd0915c6.

[7] J. R. Holton and G. J. Hakim. In: *An introduction to dynamic meteorology*. 4th ed. Vol. 88. International Geophysics Series. Elsevier Academic Press, 2004.

[8] Gao Huang et al. *Densely Connected Convolutional Networks*. 2016. DOI: 10.48550/ARXIV.1608.06993. URL: https://arxiv.org/abs/1608.06993.

[9] R. Jackson et al. *PyDDA: A Pythonic Direct Data Assimilation framework for wind retrievals*. Version 0.5.2. July 2020. DOI: 10.5281/zenodo.3942686. URL: https://doi.org/10.5281/zenodo.3942686.

[10] Zahra Karevan and Johan A.K. Suykens. "Transductive LSTM for time-series prediction: An application to weather forecasting". In: *Neural Networks* 125 (2020), pp. 1–9. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2019.12.030. URL: https://www.sciencedirect.com/science/article/pii/S0893608020300010.

[11] R. Kikuchi et al. "Nowcasting algorithm for wind fields using ensemble forecasting and aircraft flight data". In: *Meteorological Applications* 25.3 (2018), pp. 365–375. DOI: https://doi.org/10.1002/met.1704.

[12] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. DOI: 10.48550/ARXIV.1412.6980. URL: https://arxiv.org/abs/1412.6980.

[13] R. Klein. "Scale-Dependent Models for Atmospheric Flows". In: *Annual Review of Fluid Mechanics* 42.1 (2010), pp. 249–274. DOI: 10.1146/annurev-fluid-121108-145537.

[14] J. Nathan Kutz. "Deep learning in fluid dynamics". In: *Journal of Fluid Mechanics* 814 (2017), pp. 1–4. DOI: 10.1017/jfm.2016.803.

[15] A. M. P. de Leege, M. M. van Paassen, and M. Mulder. "Using Automatic Dependent Surveillance-Broadcast for Meteorological Monitoring". In: *Journal of Aircraft* 50.1 (2013), pp. 249–261. DOI: 10.2514/1.C031901. eprint: https://doi.org/10.2514/1.C031901. URL: https://doi.org/10.2514/1.C031901.

[16] Karim Legrand et al. "Robust aircraft optimal trajectory in the presence of wind". In: *IEEE Aerospace and Electronic Systems Magazine* 33.11 (2018), pp. 30–38. DOI: 10.1109/MAES.2018.170050.

[17] Martin Leutbecher and T.N. Palmer. "Ensemble forecasting". In: 514 (Feb. 2007), p. 31. DOI: 10.21957/c0hq4yg78. URL: https://www.ecmwf.int/node/10729.

[18] Guilin Liu et al. *Image Inpainting for Irregular Holes Using Partial Convolutions*. 2018. DOI: 10.48550/ARXIV.1804.07723. URL: https://arxiv.org/abs/1804.07723.

[19] Andrew C. Lorenc and F. Rawlins. "Why does 4D-Var beat 3D-Var?" In: *Quarterly Journal of the Royal Meteorological Society* 131.613 (2005), pp. 3247–3257. DOI: https://doi.org/10.1256/qj.05.85. eprint: https://rmets.onlinelibrary.wiley.com/doi/pdf/10.1256/qj.05.85. URL: https://rmets.onlinelibrary.wiley.com/doi/abs/10.1256/qj.05.85.

[20] V. Mazzarella et al. "Comparison between 3D-Var and 4D-Var data assimilation methods for the simulation of a heavy rainfall case in central Italy". In: *Advances in Science and Research* 14 (Aug. 2017), pp. 271–278. DOI: 10.5194/asr-14-271-2017.

[21] Met Office. *Cartopy: a cartographic python library with a Matplotlib interface*. Exeter, Devon, 2010 - 2015. URL: https://scitools.org.uk/cartopy.

[22] National Centers for Environmental Prediction, National Weather Service, NOAA, U.S. Department of Commerce. *NCEP GFS 0.25 Degree Global Forecast Grids Historical Archive*. Data collected throughout November 2022. Boulder CO, 2015. URL: https://doi.org/10.5065/D65D8PWK.

[23] Grace Peng. *Analysis, reanalysis, forecast—what's the difference?* National Center for Atmospheric Research. 2014. URL: https://rda.ucar.edu/datasets/ds084.1/docs/Analysis.pdf (visited on 12/06/2022).

[24] C. Potvin, A. Shapiro, and J. Gao. "Use of a Vertical Vorticity Equation in Variational Dual-Doppler Wind Analysis". In: *Journal of Atmospheric and Oceanic Technology* 26.10 (2009), pp. 2089–2106. DOI: 10.1175/2009JTECHA1256.1. URL: https://journals.ametsoc.org/view/journals/atot/26/10/2009jtecha1256_1.xml.

[25] C. Potvin, A. Shapiro, and M. Xue. "Impact of a Vertical Vorticity Constraint in Variational Dual-Doppler Wind Analysis: Tests with Real and Simulated Supercell Data". In: *Journal of Atmospheric and Oceanic Technology* 29.1 (2012), pp. 32–49. DOI: 10.1175/JTECH-D-11-00019.1. URL: https://journals.ametsoc.org/view/journals/atot/29/1/jtech-d-11-00019_1.xml.

[26] Chengping Rao, Hao Sun, and Yang Liu. "Physics-informed deep learning for incompressible laminar flows". In: *Theoretical and Applied Mechanics Letters* 10.3 (2020), pp. 207–212. ISSN: 2095-0349. DOI: https://doi.org/10.1016/j.taml.2020.01.039. URL: https://www.sciencedirect.com/science/article/pii/S2095034920300350.

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. DOI: 10.48550/ARXIV.1505.04597. URL: https://arxiv.org/abs/1505.04597.

[28] L. Schweri et al. "A Physics-Aware Neural Network Approach for Flow Data Reconstruction From Satellite Observations". In: *Frontiers in Climate* 3 (2021). ISSN: 2624-9553. DOI: 10.3389/fclim.2021.656505.

[29] NASA Goddard Institute for Space Studies. *Panoply netCDF, HDF and GRIB Data Viewer*. Software package. 2022. URL: https://www.giss.nasa.gov/tools/panoply/.

[30] Matthias Steiner et al. "Translation of Ensemble Weather Forecasts into Probabilistic Air Traffic Capacity Impact". In: *Air Traffic Control Quarterly* 18 (Oct. 2010), pp. 229–254. DOI: 10.2514/atcq.18.3.229.

[31] J. Sun et al. "pyModeS: Decoding Mode-S Surveillance Data for Open Air Transportation Research". In: *IEEE Transactions on Intelligent Transportation Systems* (2019). ISSN: 1524-9050. DOI: 10.1109/TITS.2019.2914770.

[32] J. Sun et al. "Weather Field Reconstruction using aircraft surveillance data and a novel meteo-particle model". In: *PLoS ONE* 13.10 (2018), e0205029. DOI: https://doi.org/10.1371/journal.pone.0205029.

[33] Jin Xu et al. "Reluplex made more practical: Leaky ReLU". In: *2020 IEEE Symposium on Computers and Communications (ISCC)*. 2020, pp. 1–7. DOI: 10.1109/ISCC50000.2020.9219587.

APPENDIX

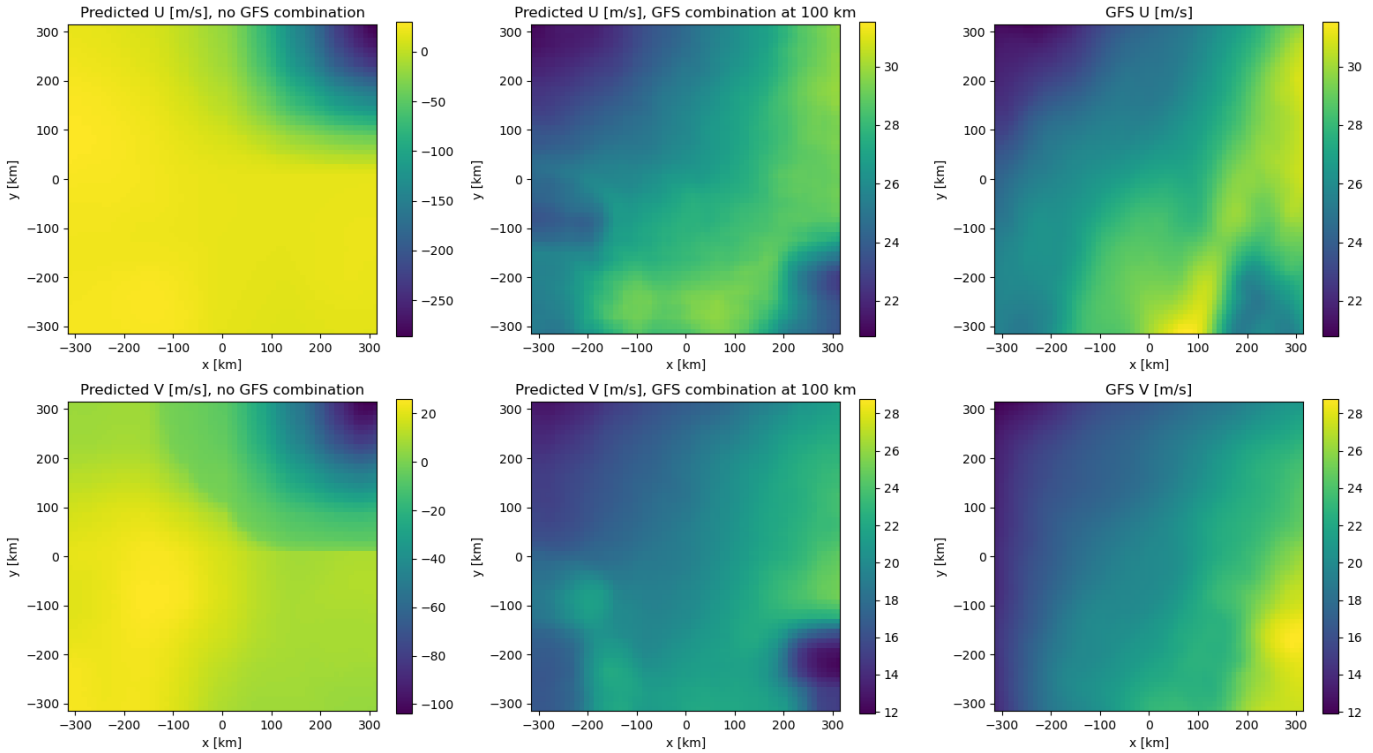


Fig. 22. **GFS infilling: wind components. 00h at 01/01/2018 at 37,000 ft. U and V-component have different colour scale. Left column: prediction on sparse nighttime data, middle column: prediction with GFS infilling. GFS grid itself is shown in third column. Infilling and GFS field use same colour scale.**

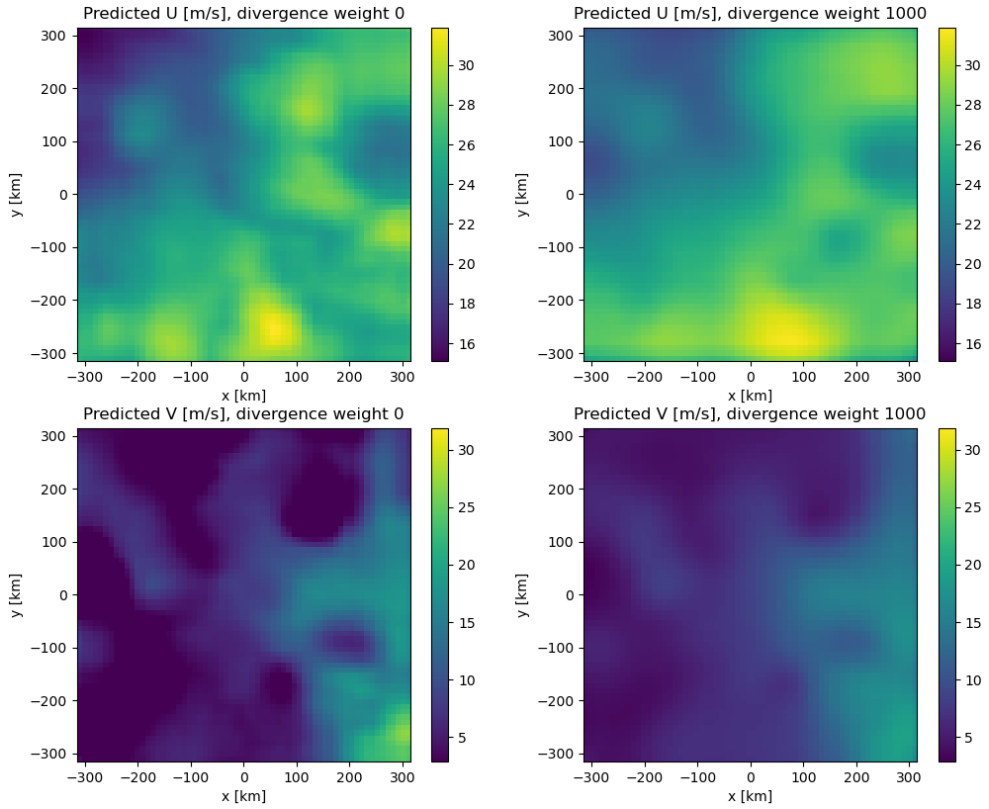


Fig. 23. **Effect of increasing divergence weight, shown for 01/01/2018, 12h nowcast, at 34,000 feet. U and V-component have different colour scale. Increasing the divergence weight leads to a smoothing of the wind field, yet it does not result in a drastically different solution. Vorticity weight is 0 in both cases.**

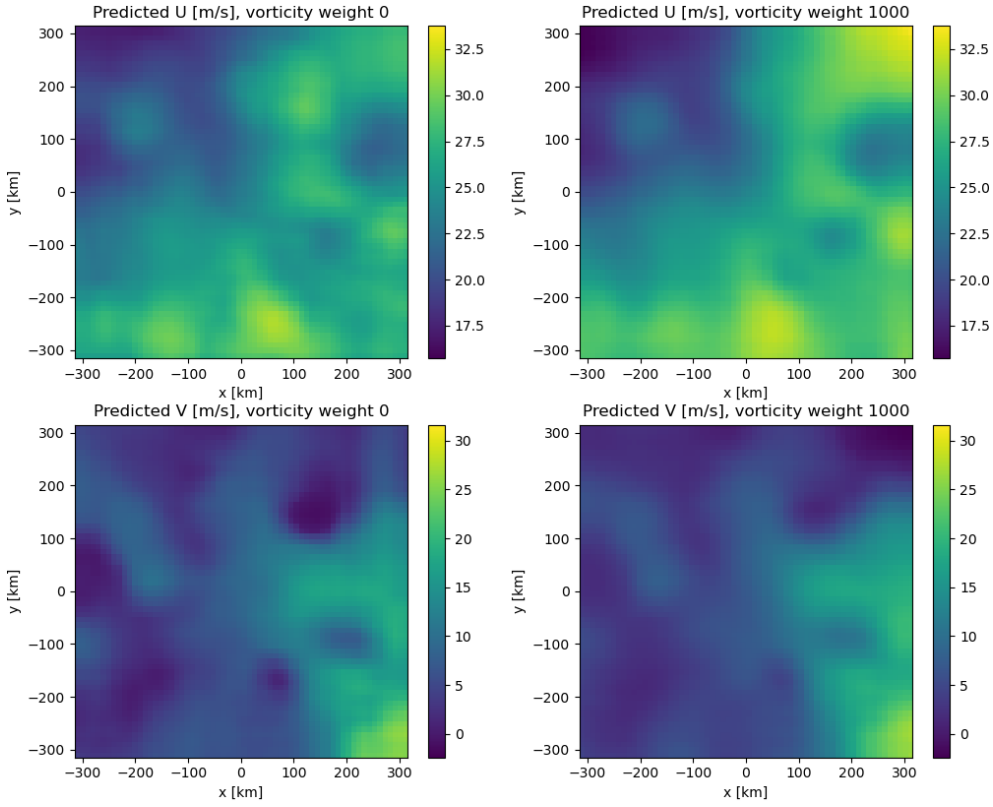


Fig. 24. **Effect of increasing vorticity weight, shown for 01/01/2018, 12h nowcast, at 34,000 feet. U and V-component have different colour scale. Increasing the divergence weight leads to a smoothing of the wind field, yet it does not result in a drastically different solution. Divergence weight is 10 in both cases.**

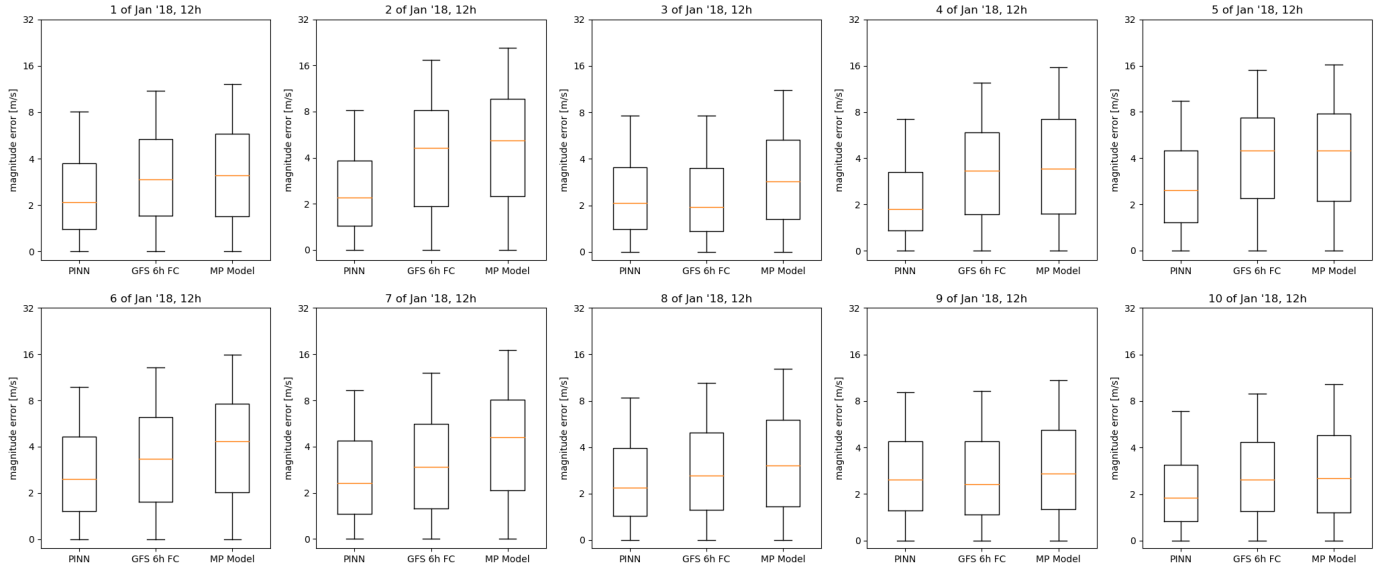


Fig. 25. **Magnitude error box-plots, nowcasting. y-scale is logarithmic. The neural network consistently outscores both GFS forecasts and the MP model. Outliers are omitted.**

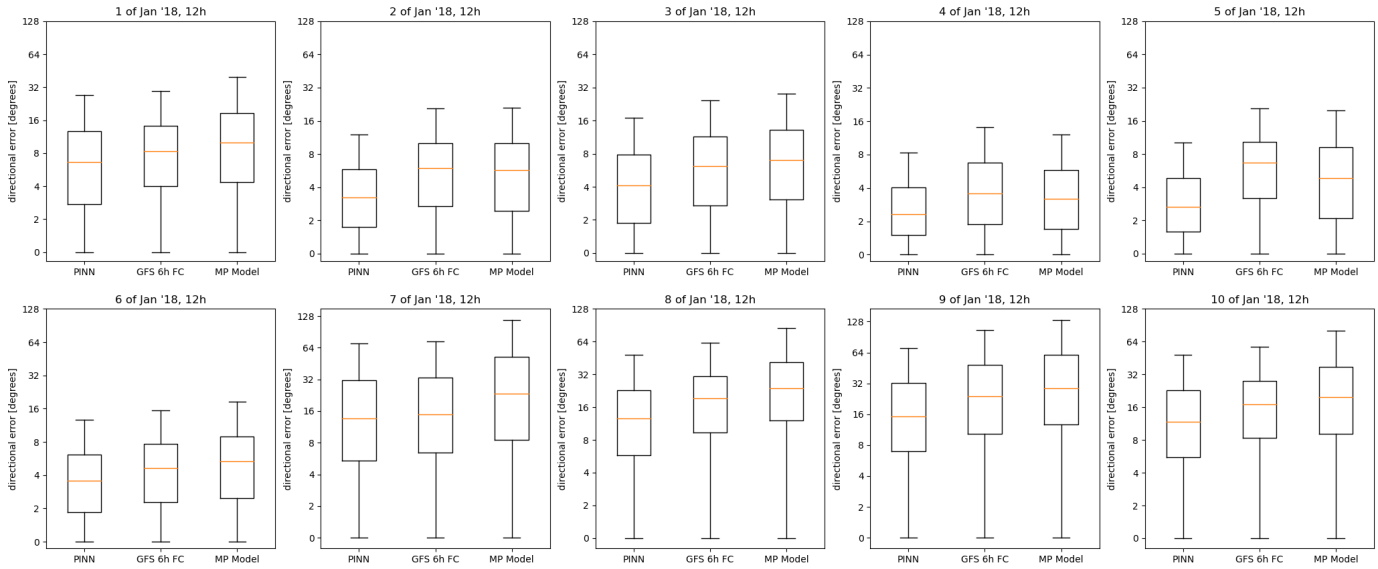


Fig. 26. **Directional error box-plots, nowcasting. y-scale is logarithmic. The neural network consistently outscores both GFS forecasts and the MP model. Outliers are omitted.**

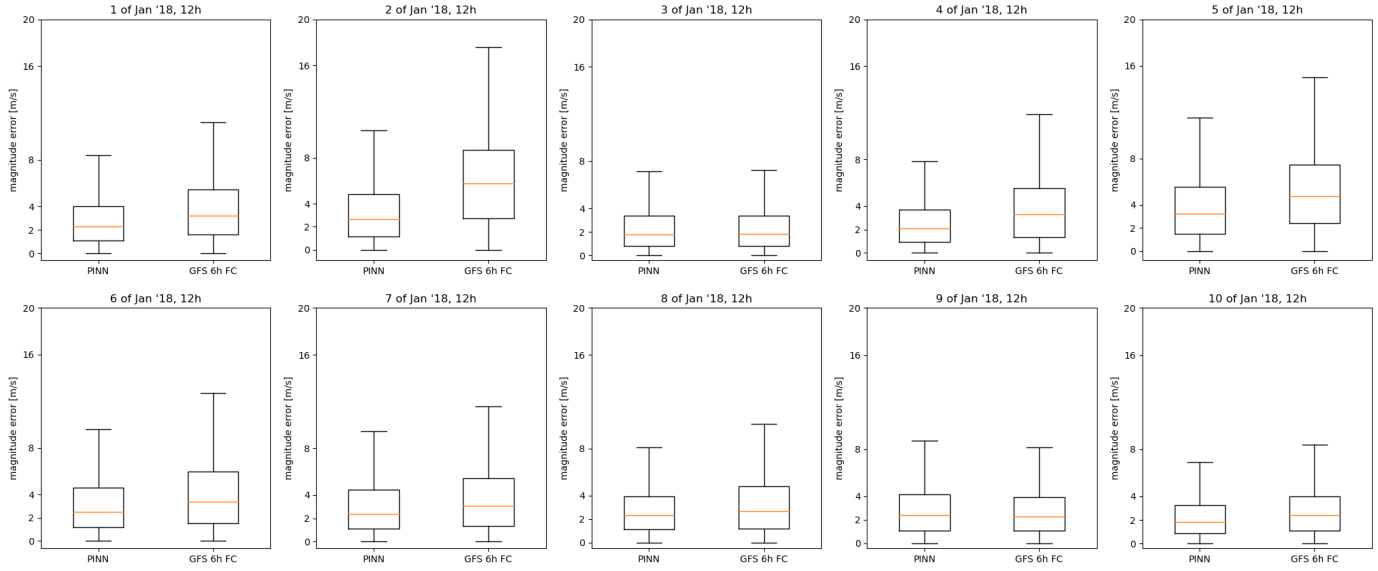


Fig. 27. **Magnitude error box-plots, forecasting. y-scale is logarithmic. The neural network consistently outscores both GFS forecasts and the MP model. Outliers are omitted.**

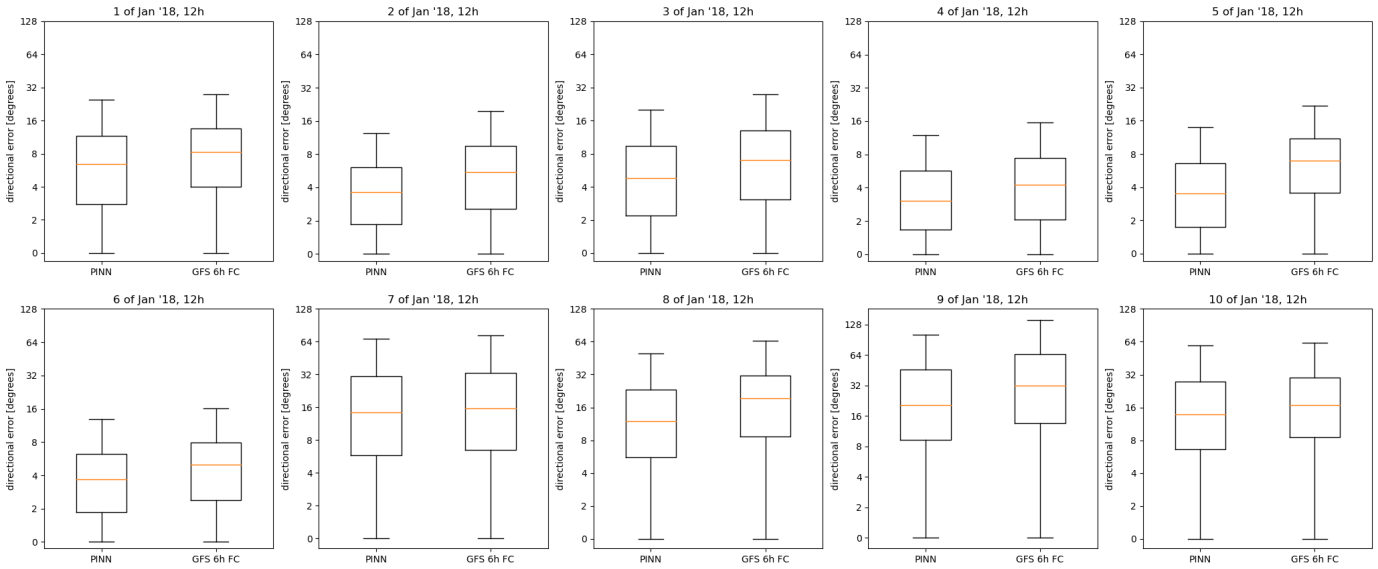


Fig. 28. **Directional error box-plots, forecasting. y-scale is logarithmic. The neural network consistently outscores both GFS forecasts and the MP model. Outliers are omitted.**

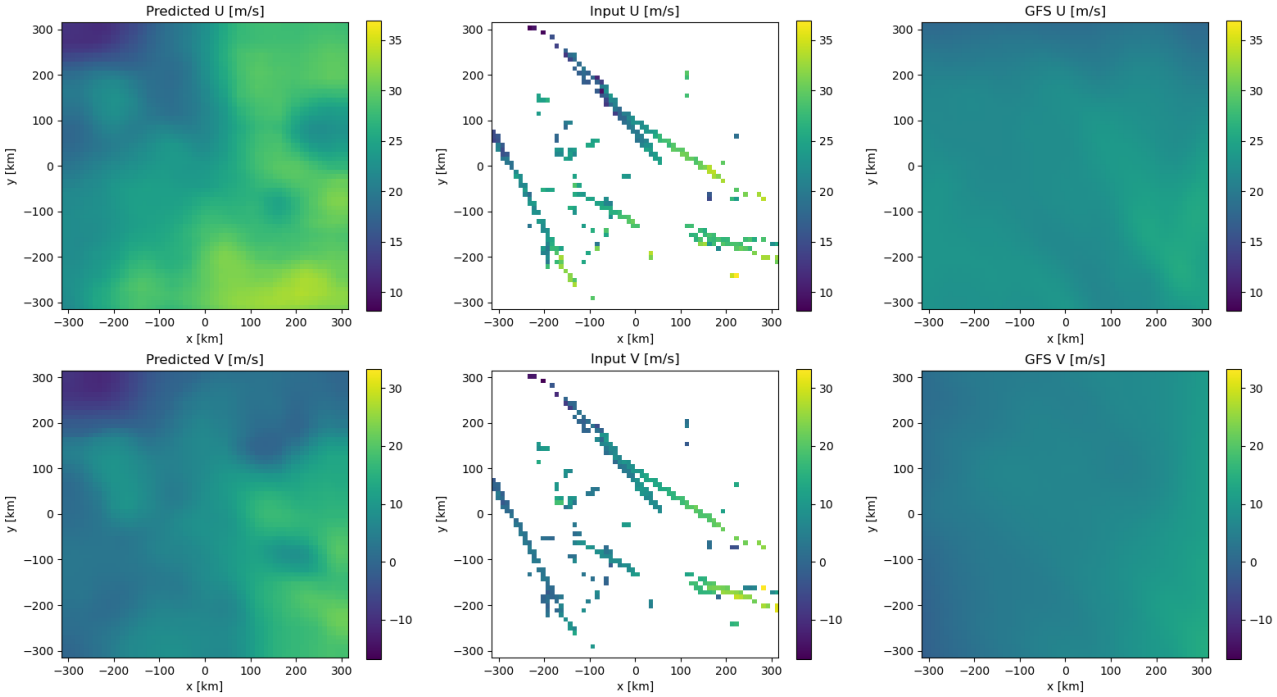


Fig. 29. **Wind field components of predicted field, input, and GFS 6h forecast, for 01/01/2018, 12h, at 34,000 ft. U and V-component have different colour scale. The GFS 6h forecast is not necessarily the true wind field.**

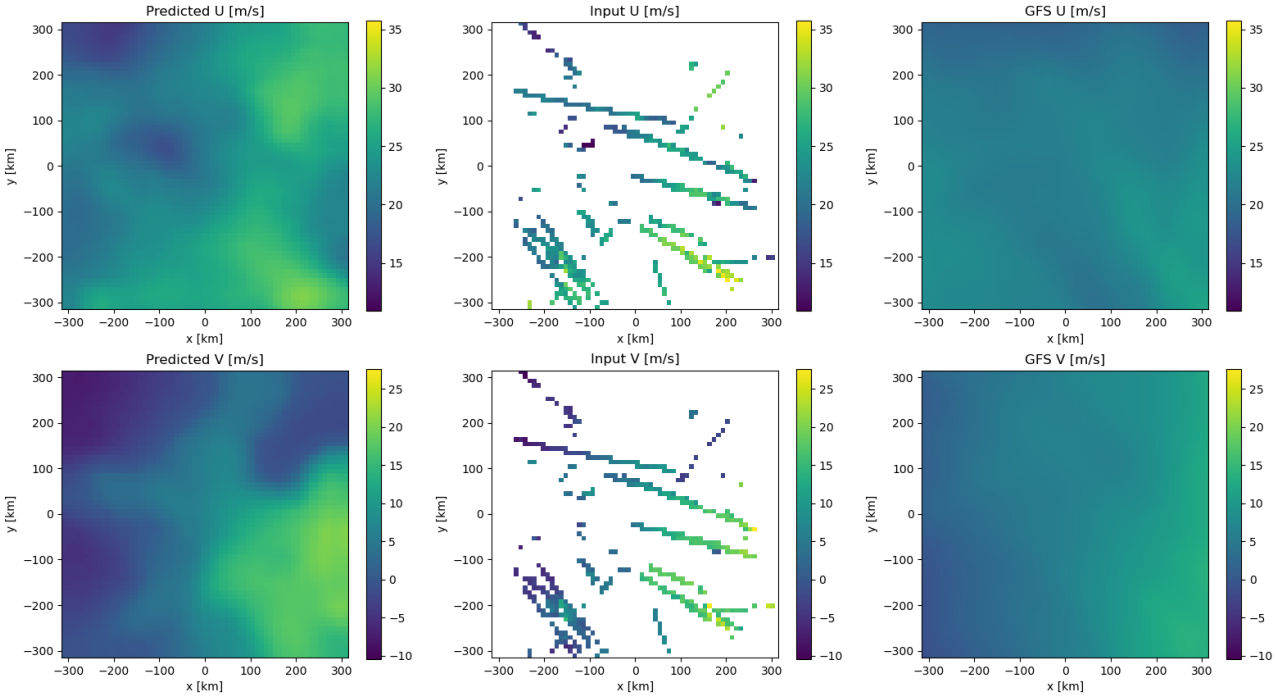


Fig. 30. **Wind field components of predicted field, input, and GFS 6h forecast, for 01/01/2018, 12h, at 36,000 ft. U and V-component have different colour scale. The GFS 6h forecast is not necessarily the true wind field.**

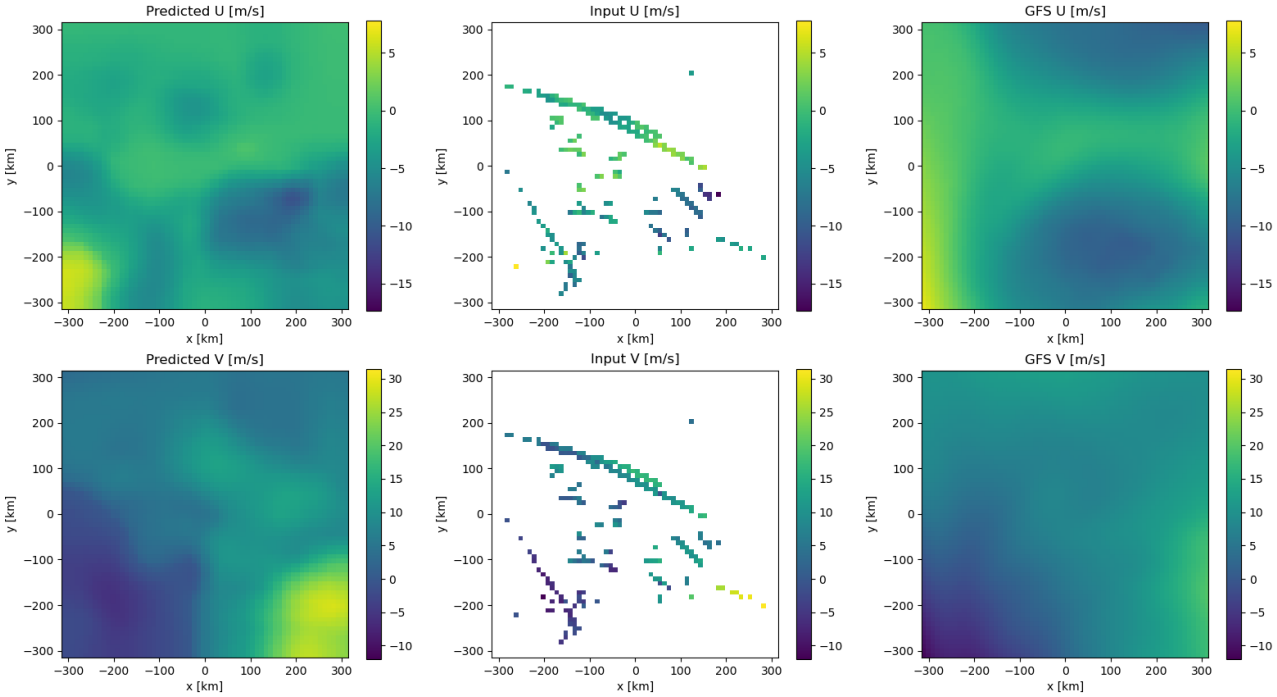


Fig. 31. **Wind field components of predicted field, input, and GFS 6h forecast, for 09/01/2018, 12h, at 34,000 ft. U and V-component have different colour scale. The GFS 6h forecast is not necessarily the true wind field.**

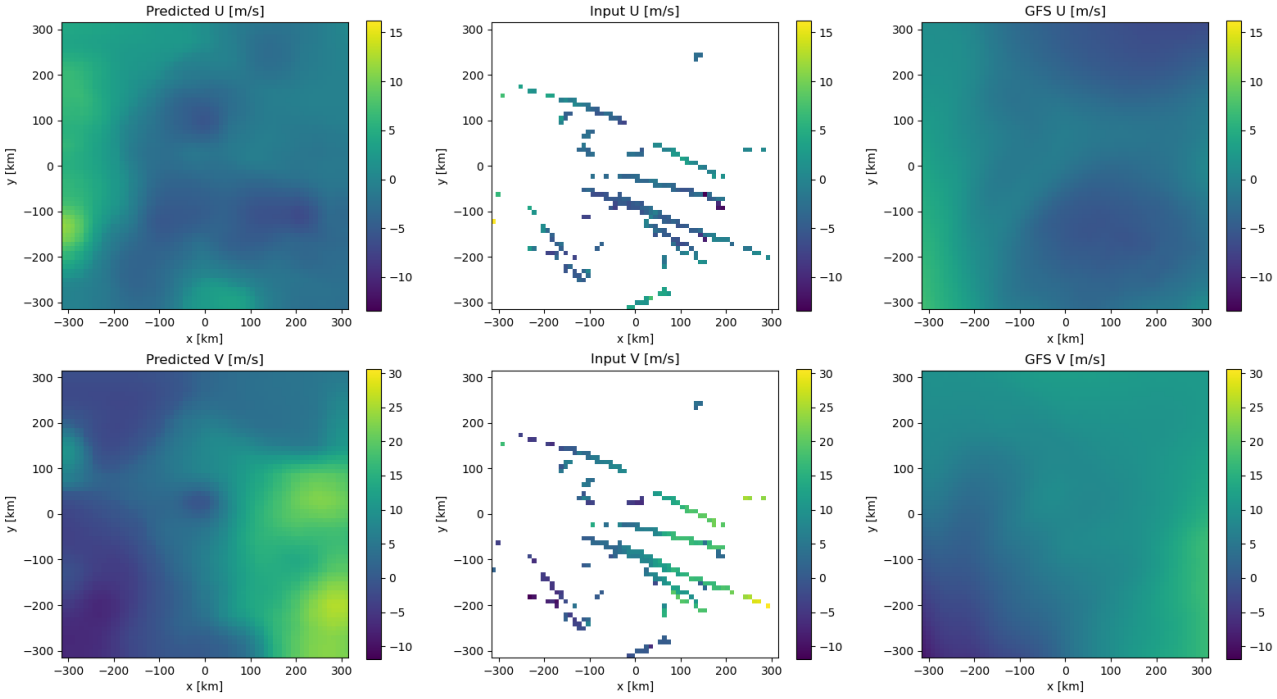


Fig. 32. **Wind field components of predicted field, input, and GFS 6h forecast, for 09/01/2018, 12h, at 36,000 ft. U and V-component have different colour scale. The GFS 6h forecast is not necessarily the true wind field.**

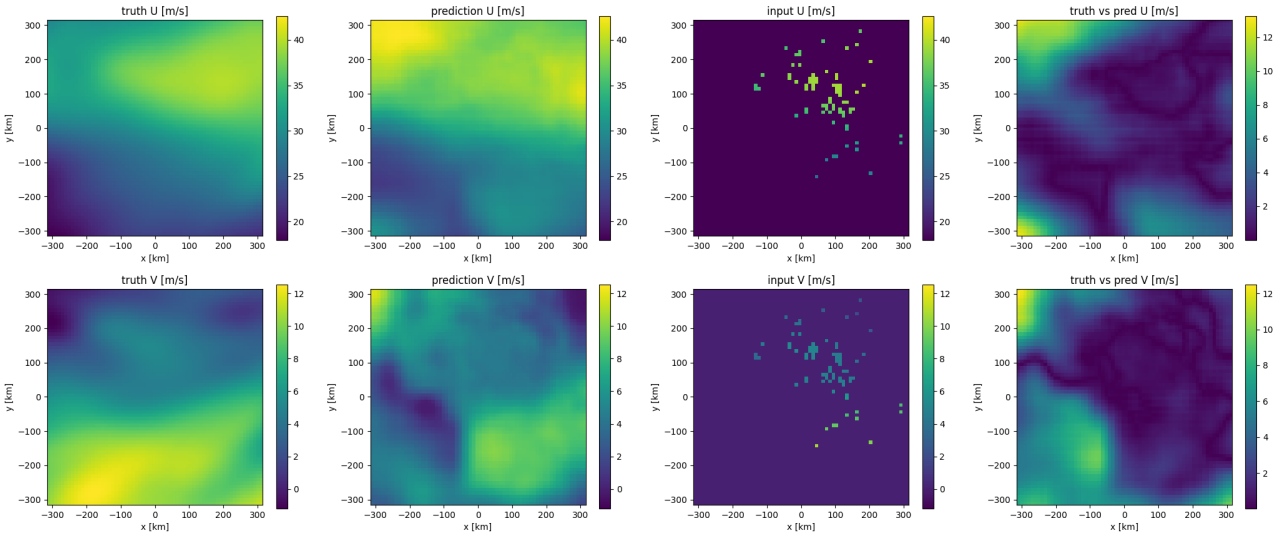


Fig. 33. **Wind field components, Reanalysis validation 01/07/2019 at 38,000 ft. Sparse input. U and V-component have different colour scale. The first column shows the reanalysis truth, with the second column showing the predicted field, based itself on the input the network received, shown in the third column. The absolute difference between truth and prediction is shown in the fourth column. For this fourth column, a different colour scale is used.**

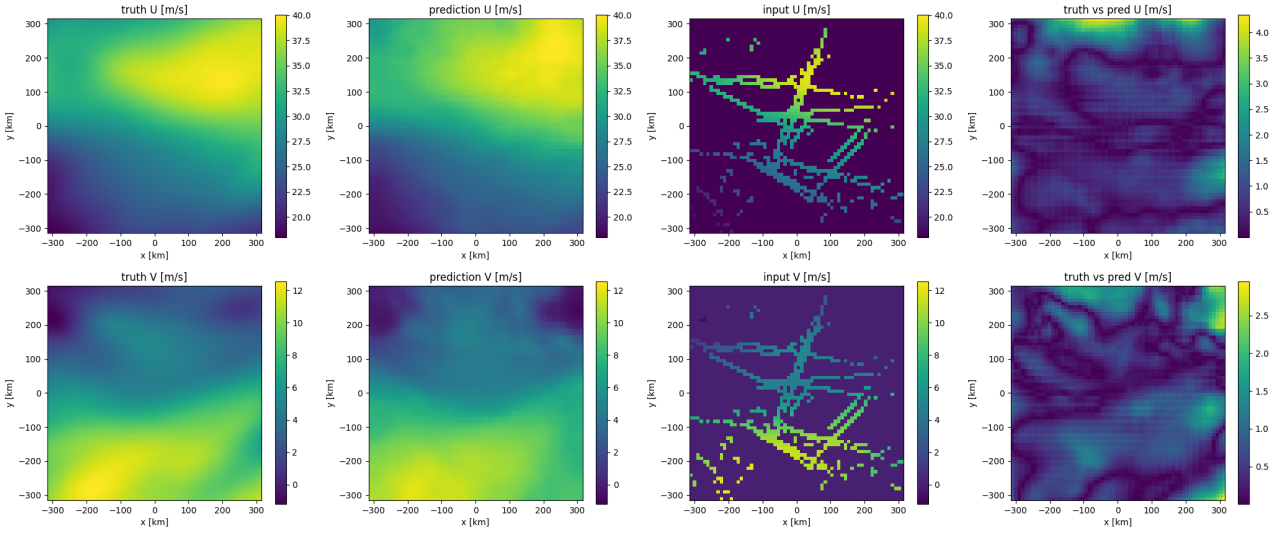


Fig. 34. **Wind field components, Reanalysis validation 01/07/2019 at 38,000 ft. Input field is less sparse (when compared with Figure 33). U and V-component have different colour scale. The first column shows the reanalysis truth, with the second column showing the predicted field, based itself on the input the network received, shown in the third column. The absolute difference between truth and prediction is shown in the fourth column. For this fourth column, a different colour scale is used.**

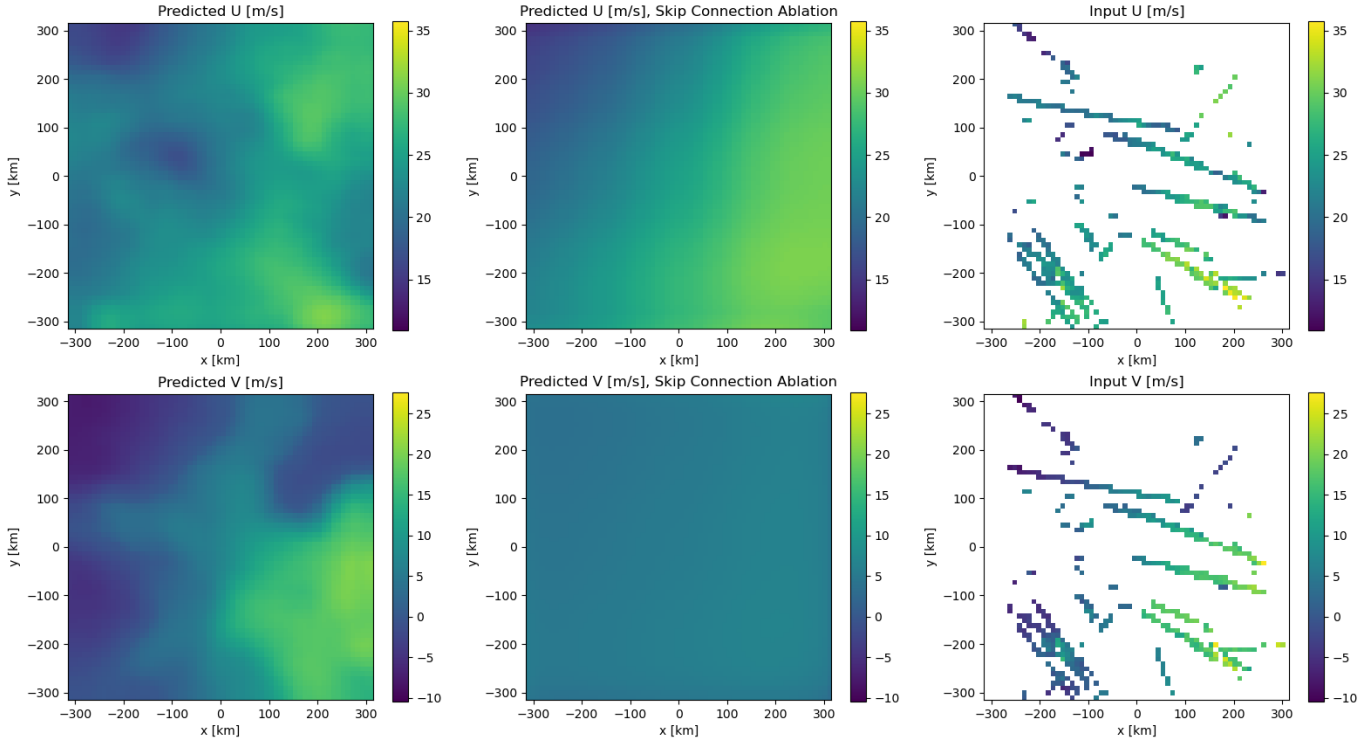


Fig. 35. **Effect of ablation of network skip connections, 01/01/2018 at 36,000 ft. U and V-component have different colour scale. First column shows predictions for a network with skip connections, local features from input (third column) are featured. Second column shows predictions for a network without skip connections, no locality can be seen. Skip connections are thus necessary to maintain locality.**

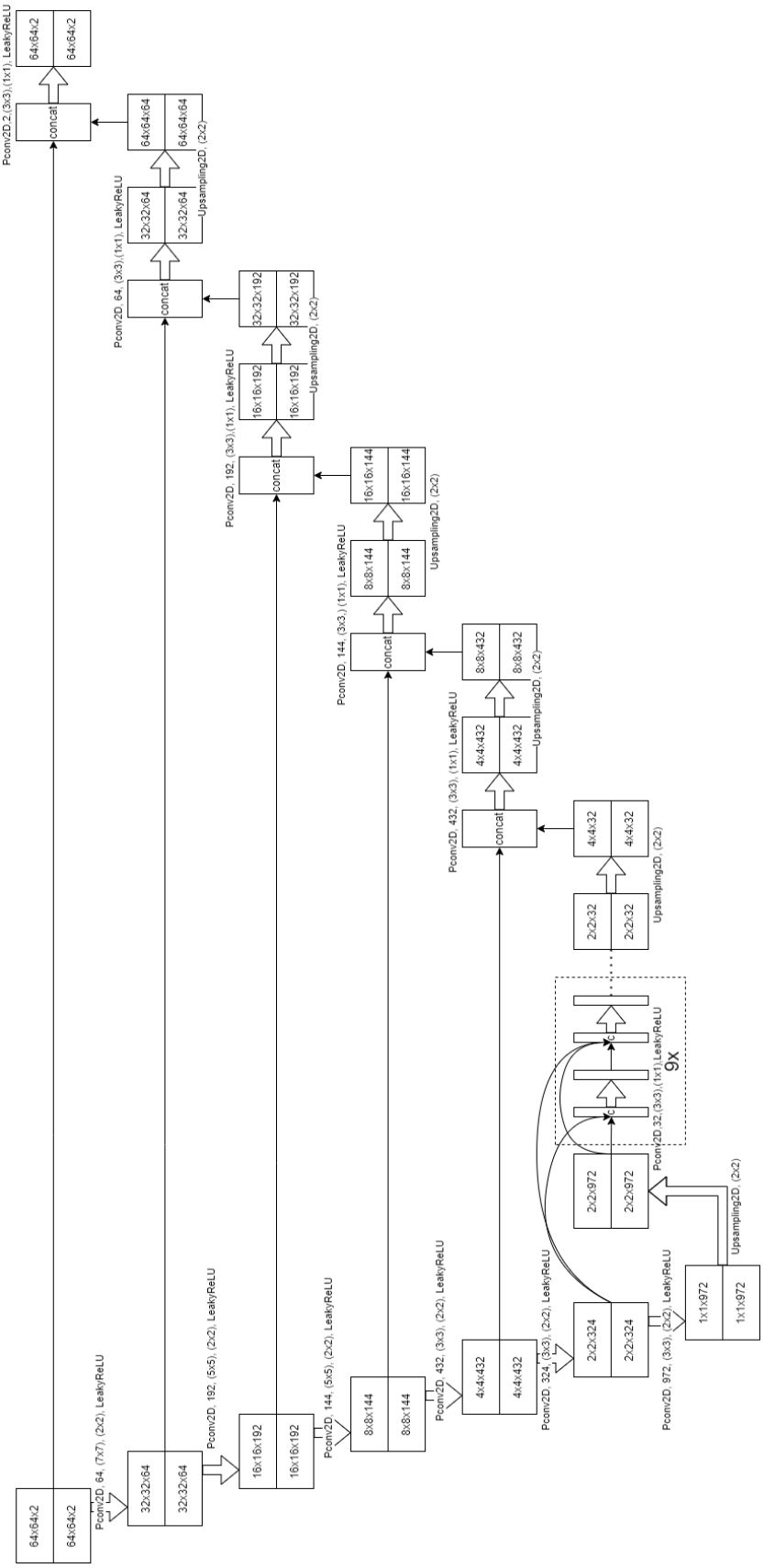


Fig. 36. **Network architecture. Convolution channels are followed by filter size and stride. Inputs are wind components and masks. Dense convolution is done 9 times.**

Part II

Preliminary Report (Graded)

1

Introduction

Wind fields are an important factor in aircraft performance studies. Both magnitude and direction of wind influence other flight factors such as trajectory, weather risk, speed and fuel consumption, as indicated by Sun et al [23]. Thus, wind nowcasting and forecasting play a crucial role within aviation industry. This can be performed by numerical weather prediction software. Forecasting relies on making a prediction based on prior weather data. Nowcasting aims to provide a weather field estimation in real-time, leveraging any data for this purpose.

In terms of aviation, wind forecasts are readily available from e.g. GFS or ECMWF services. Nowcasting can be supported by transmitted aircraft data and schemes to derive meteorological information. According to Sun et al. [23], forecast data suffer from decreased local accuracy in terms of providing real-time wind data for aircraft. Transmitted aircraft data can be used to improve the nowcast; for this purpose, Sun et al. developed the Meteo-Particle model. This algorithm considers aircraft wind/temperature observations as particles, which are spread throughout the airspace. Wind and temperature estimation can be done via interpolating surrounding particles. Short-term forecast can also be made from this model. However, the algorithm does not include wind dynamics, safe a wind bias in its particle propagation. Therefore, accuracy rapidly drops in less observed areas [23].

Problem Statement

Therefore, the aim of the thesis is to improve over the MP model, with the primary method being the incorporation of wind dynamics. The following problem statement is proposed:

Wind nowcasting and short-term forecasting are useful in aircraft operations, for which an Meteo-Particle model was developed by Sun et al.[23]. This model remains blind however to wind dynamics and suffers from low accuracy in sporadically observed areas. The research aim is to improve (upon) the MP model by including wind dynamics.

2

Literature Study

In order to gain more knowledge about the problem, and possible solutions, a literature study was done into multiple areas of interest. The starting point was the MP model itself. Then, other literature on wind nowcasting and forecasting was consulted. This work mainly includes numerical weather prediction (NWP) material and meteorology in general. Already proposed methodologies, namely neural nets and reduced order modelling, will also be discussed. This chapter will largely follow the structure and material of the previously submitted research plan for this thesis.

2.1. Meteo-Particle Model

Sun et al. [23] developed the Meteo-Particle model (abbreviated to MP model) with the purpose of generating wind field estimates based on aircraft wind measurements. Temperature estimates are also modeled this way by the MP model. The measurements are derived from Mode-S and ADS-B data [22]. The aircraft transmits both ground speed and true airspeed. Simply subtracting the true airspeed from ground speed yields the wind speed. When vectored, a direction is also derivable. The mainstay of the MP model is to regard those measurements as particles, which are multiplied, and spread throughout the airspace according to a random Gaussian walk method with wind bias [23]. Along wind and temperature information, the particle also carries information on its origin and age. The latter two can serve as an indication of the particles validity at a certain point in time and space. If one wishes to know a wind or temperature estimate at a certain location in the airspace, nearby spread particles can be weight-averaged to achieve an estimate. This nowcasting capability is joined by a short-term forecasting capability, based on Gaussian processing.

Sun et al. report a 67% error decrease in wind nowcasting compared to GFS (mean absolute error goes from 4 to 1.3 [m/s]), and 26% for temperature, 1.6 1.2 [K] MAE [23]. For short term prediction, a slight improvement to 3.7 [m/s] MAE and 1.23 [K] MAE is found [23]. The results were obtained based on a random data split in training and validation data. In the following, a summary of the MP model framework is given.

2.1.1. ADS-B and Mode-S

The aircraft wind measurements are constructed out of ADS-B and Mode-S data. ADS-B stands for Automatic Dependent Surveillance-Broadcast, aircraft fitted with this system can broadcast information, such as, but not limited to, barometric altitude, ground speed and coordinates [12] [23]. ADS-B is reported to have an update rate of 0.5 seconds, which practically means a continuous stream of information is available. However, ADS-B alone is not enough to estimate the local wind, as it only transmits ground speed, but airspeed is also required in the wind computation [23]. Sun et al. propose using Mode-S data to fill this information gap. Mode-S transponders send information when requested by a secondary surveillance radar, in the form of Comm-B messages. However, this message has no information on its type as the secondary surveillance radar already knows which aircraft it receives from and how to interpret the message [23]. Sun et al. apply a range of techniques to determine the aircrafts ICAO address, check corruptness, and check content. The ICAO address by doing a 'reverse parity

check' [23]; if such an address happens to be incorrect, it will come to light by cross-referencing with the ADS-B messages [23]. Status bits in BDS code indicate whether a register has been filled or not. Doing multiple checks result in knowing which message type is transmitted. The measurements may still contain faulty values, because of sensor or transmission errors. This must be dealt with afterwards, as the signal processing algorithm itself cannot compensate for this. Sun designed such a decoding library for Mode-S, called PyModeS [22], also used for his paper. This thesis research uses the same data [1]. This library also handles ADS-B decoding, and Comm-B.

2.1.2. Wind Speed and Temperature Models

From the transmitted ADS-B data and Mode-S data, the required variables can be constructed. Sun et al.[23] propose the next set of equations.

ADS-B transmits the following [23]:

- Barometric altitude
- Ground speed

And Mode-S provides:

- Indicated airspeed
- True airspeed
- Mach number

Assuming ISA conditions, the pressure and temperature are calculated via:

$$p = p_0 \cdot exp(-\frac{g_0}{RT_0}h) \tag{2.1}$$

$$T = \frac{V_{TAS,50} \cdot T_0}{M^2 \cdot a_0^2} \tag{2.2}$$

The final true airspeed is computed via the state equation and Mach conversion:

$$\rho = \frac{p}{RT} \tag{2.3}$$

$$V_{TAS} = M \cdot a_0 \sqrt{\frac{T}{T_0}} \tag{2.4}$$

Together with the aircraft heading χ_a, and track angle χ_g information, a wind vector may be computed as can be seen in Figure 2.1 [23]:

$$\vec{V_W} = \vec{V_G} - \vec{V_{TAS}} \tag{2.5}$$

2.1.3. MP Model Method

After proper inference of wind speed and temperature from aircraft measurements, they can be fed to the model. First, a probabilistic rejection is performed on the input data, as incorrect decoding might happen. For each new measurement, the mean and variance of particles in the same height layer are calculated. From this, a probability function is set up from this, and the new measurement will be accepted according to that probability. As said before, the new measurement will generate a number of particles, at the observation location [23]. The wind and temperature values receive a slight variance, and are propagated in the next time-steps with a slight wind bias. The particles also carry an age state, and their origin coordinates. At each time-step, probabilistic resampling is performed to delete older particles, and particles which are out-of-bounds [23].

If one wishes to gain a wind/temperature estimate at a certain location, nearby particles are weight-averaged into an estimate. The weight of the particles depends on the distance to the requested co-ordinate, and the distance to their origin, the weights are calculated by exponential functions. Finally, the estimate also receives a confidence level. This is based on the number of particles involved, the distance between the estimate and the particles, the homogeneity of the particles and their age [23].

[1] Sun J. Supplemental dataset for "Weather field reconstruction using aircraft surveillance data and a novel meteo-particle model"; 2018. Available from: https://doi.org/10.6084/m9.figshare.6970403.

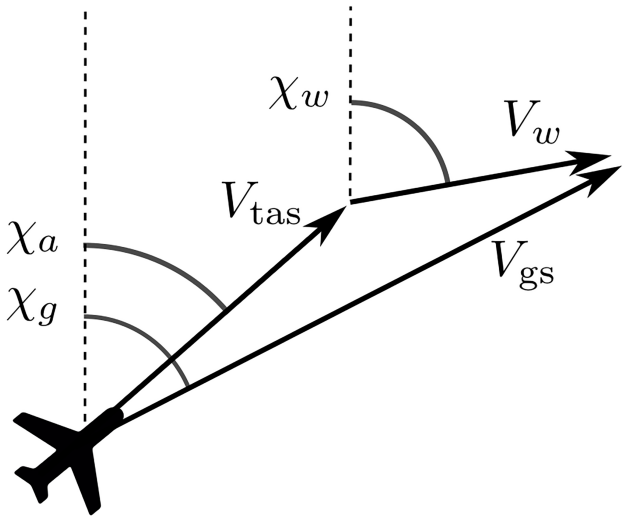


Figure 2.1: Relation of ground speed, wind speed and true air speed vectors, courtesy of Sun et al. [23]

2.1.4. Short-term prediction

Short term prediction is performed by applying Gaussian processes. A regression is built up, with the time as a dimension. A prior is formed with a set of observations, over this time dimension, based on different kernels. Then, future values are derived from this prior [23]. An RMSE of 24.7 [m/s] is reported on a 30 minute prediction.

2.2. Meteorology and Numerical Weather Prediction

The scientific discipline of meteorology and its computational branch of numerical weather prediction are widely researched topics, as understanding and predicting weather is crucial for many applications, e.g.: agriculture, risk management, economical activities, disaster response, climate change and also air traffic. Meteorology is the study field which concerns the atmosphere, both in terms of weather and of climate [6]. Weather and climate phenomena happen on different scales, both in space and time. Before delving deeper into the specifics, one must know which scale of meteorology is appropriate for this thesis research, as different equations, phenomena and assumptions hold according to Klein [10] for different scales. To further complicate things, Klein states that for general use, one must consider multiple scales at once; that is, both multiple length and time regimes.

According to Klein, motions taking place at the scale of about 150 kilometers are in the Meso-B scale. The airspace concerned in the thesis research is in the same order. The next order is the synoptic scale, at 1100 km, while the Meso-gamma scale is at 11 km [10]. Klein also provides a set of general governing equations, being the Euler or fully compressible flow equations. They are inviscid Navier-Stokes equations; the inviscid assumptions holds, as turbulence effects are much stronger than viscid forces, stated by the American Meteorological Society [20]. The equations include terms for rotation, gravity and source terms. From those Euler equations, models can be derived depending on different assumptions or simplifications. In general, the principles of meteorology are very complex, depending on many sets of governing equations, phenomena such as gravity waves and advection, and the dependence on multiple scales. A lot of research is done with vast resources; for this thesis research, the goal is to improve wind (and temperature) field estimation in an accessible, cheaper way.

2.2.1. Governing Equations

Holton and Hakim [6] give a set of basic equations, the first being the horizontal momentum equation:

$$\frac{D\vec{V}}{Dt} + f\vec{k} \times \vec{V} = -\frac{1}{\rho}\vec{\nabla}_p \tag{2.6}$$

with f being the Coriolis parameter, and the right term having the pressure gradient force in it. An isobaric form is given too [6], being:

$$\frac{D\vec{V}}{Dt} + f\vec{k} \times \vec{V} = -\vec{\nabla}_p \Phi \tag{2.7}$$

where Φ is geopotential, and the right term in its totality expresses geopotential gradient along constant pressure surfaces. The coriolis parameter, f, is expressed as [6]:

$$f = 2\Omega sin\phi \tag{2.8}$$

Ω being the Earths rotational speed, and ϕ the latitude. The Coriolis parameter incorporates the Coriolis effect into the equations. This effect is due to the conservation of angular momentum; for large scale motions, a meridional wind speed implies traversing latitudes, thus a change in distance to the rotating axis. Because of the conservation of angular momentum, the zonal wind speed must change [6]. For zonal winds, the wind is going faster than the rotation of the Earth, leading to excess centrifugal force, which is compensated by a meridional acceleration, as explained by Holton and Hakim in [6], where a more complete explanation may be found. It is also stated that Coriolis forces have limited effect on time scales compared small to Earths rotation.

Next, the well-known continuity equation is discussed, its Eulerian version being [6]:

$$\frac{\partial \rho}{\partial t} + \vec{\nabla} \cdot (\rho \vec{V}) = 0 \tag{2.9}$$

The first term is the rate of change of density for an air volume, the second term is the amount of mass flowing out of the volume. Those must add up to zero. When one assumes the density does not change, the equation reduces to:

$$\vec{\nabla} \cdot (\rho \vec{V}) = 0 \tag{2.10}$$

Holton and Hakim also derive a continuity equation on isobaric surfaces, being [6]:

$$(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y})_p + \frac{\partial \omega}{\partial p} = 0 \tag{2.11}$$

ω is defined as the pressure change in direction of the motion, as stated by Holton and Hakim [6]. Furthermore, vorticity of the flow also plays a role in meteorological dynamics. According to Potvin et al. [17], the next vorticity equation holds:

$$\frac{\partial \zeta}{\partial t} + \vec{V} \cdot \nabla \zeta + \zeta \cdot (\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}) = 0 \tag{2.12}$$

with ζ the vorticity, being $\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$ [17].

2.2.2. Data Assimilation in NWP

NWP in itself is a wide branch, which can go up all the way to planet scale climate analysis. However, for this thesis research, wind nowcasting and short term forecasting are the subject. At a later stage, temperature estimation could be added. According to De Haan and Stoffelen [5], temperature in high air fluctuates on large scales only, and therefore is of reduced concern. Nowcasting in the context of the thesis problem is closely related to the principle of meteorological data assimilation. Wind measurements are to be combined with certain physics knowledge, into an attempt to accurately estimate wind fields. In the field of meteorology, data assimilation is defined as producing a weather prediction at the so called analysis time, given earlier observations [2].

Two methods are defined, being 3DVAR, and 4DVAR. 4DVAR is a technique where an observation is instantaneously put in the model, whereas 3DVAR uses a collection of previous observations, to process at a chosen analysis time. 4DVAR is computationally more expensive, as observations are instantaneously put in to the system: this introduces an increment over the remainder of the cycle, as stated by Mazzarella et al. [13]. Barker et al. [2] report that 4DVAR therefore may not produce the required data at the required time. As the name indicates, 4DVAR is essentially 3DVAR with an added time dimension. In the following, a short description of the underlying principles of 3DVAR is given, based on Barker et al. [2] [3].

3DVAR
3DVAR addresses data assimilation for the atmosphere; an estimate of the weather must be given at the so-called analysis time, based on the next information [2]:

- Observations in a time window before the analysis time
- A background state, being from a previous forecast
- errors on both observation and background
- physics knowledge

The goal is to, at analysis time, reach an estimate of the atmosphere which is as accurate as possible. 3DVAR achieves this via minimizing a cost function.

In [3], Barker et al. constructed the following cost function:

$$J(\vec{x}) = \frac{1}{2}(\vec{x} - \vec{x_b})^T \vec{B}^{-1}(\vec{x} - \vec{x_b}) + \frac{1}{2}(\vec{y} - \vec{y_O})^T (\vec{E} + \vec{F})^{-1}(\vec{y} - \vec{y_O}) \tag{2.13}$$

$\vec{x}$ is the analysis state which has to minimize the cost function in order to arrive at the most likely state, given the background (x_b) and observations (y_B). B is the background error covariance matrix, E is the observational error covariance matrix. F is a representiveness error matrix which incorporates errors caused by mapping $\vec{x}$ to , as defined by Barker et al. in [3]. Such a function is found to be expensive to calculate, as n degrees in freedom of $\vec{x}$ would result in a number of calculations scaling with n^2, citing [2]. Therefore, a control variable is introduced instead, which is related to the increment $\vec{x'} = \vec{x} - \vec{x_b}$ via $\vec{x'} = U\vec{v}$, with U observing $B = UU^T$ [2]. As such, the cost function can be expressed as [2]:

$$J(\vec{v}) = \frac{1}{2}\vec{v}^T\vec{v} + \frac{1}{2}(\vec{y^{O'}} - \vec{H}U\vec{v})^T(\vec{E} + \vec{F})^{-1}(\vec{y^{O'}} - \vec{H}U\vec{v}) \tag{2.14}$$

with $y^{O'} = y^O - H(x^b)$, H being a linearized observation operator. In this form, the first terms computation scales with n instead of n^2. The decomposition of B is done in a horizontal part, a vertical part and also includes a change of variables into analysis variables [2].

2.2.3. Aircraft Data Assimilation
Previous research concerning the implementation of aircraft observational data into short-term weather forecasting models is discussed here.

De Haan and Stoffelen [5] used 3DVAR to assimilate wind observations; the 3DVAR system used was the High Resolution Limited Area Model (HIRLAM) NWP, with assimilation cycles of 1 hour. A reference 3 hour cycle was also run. The horizontal grid spacing was 11 kilometers, and the entire atmosphere is modelled in 60 levels. A RMSE decrease of 5 % is reported when comparing the 1 hour against the regular three hour cycle, for both wind speed and direction [5]. They also used nesting: the boundary conditions are set by a larger, less frequently updated NWP. This NWP was the ECMWF data. It is also reported that the improvement remains present for a couple of hours only (when not refreshing the cycle with new aircraft observations) [5].

2.2.4. Ensemble forecasting
In ensemble forecasting, a number of models are initialized with different (boundary) conditions and are ran in parallel. Along the way, measurements can be used to find the best fit among the different models. Kikuchi et al. [9] propose using ensemble forecasting with aircraft wind observation data. At the start of the nowcasting attempt, a number of ensemble members are available. The goal is to pick the ensemble, or the combination of ensembles, which provide the best fit to the observed wind speeds. Kikuchi et al. researched different ways; simply taking the best ensemble, based on an RMSE metric, or a particle filter strategy. The particle filter tries to establish how probable the measurements are, given the ensemble member. Based on this, the ensemble member is assigned a weight. The research found minor improvements, from 7.12 to 6.76 [m/s] RMSE when using the particle filter compared to averaging of the ensemble members [9]. The idea of generating ensemble members, and creating a best guess on a combination of those, can be of potential use in the thesis research.

2.2.5. ECMWF and GFS

Both the Global Forecasting System and the European Centre for Medium-Range Weather Forecasts provide forecasts, which can be used in the thesis research. As ECMWF data was difficult to obtain due to server migrations, GFS is used instead. Forecasts from GFS were downloaded. In chapter 4, this data acquisition is elaborated upon.

2.3. Data-driven Methods

Because of the data oriented nature of this assignment, a study of machine learning material was performed. Two methods were researched: the first one being neural networks with physical loss functions, the second one a gradient descent algorithm which attempts to directly assimilate measurements into a model, based on gradients of different cost functions.

2.3.1. Neural networks

In past research, artificial neural networks were applied to fluid applications. Thuerey et al. [24] used such a network to solve airfoil flows. An improved of 3 % was reported. Neural networks are also applied in wind field estimation itself. Schweri et al. [19] leveraged neural networks to reconstruct wind fields, aided by satellite observations. The latter include physical losses to introduces physics knowledge. Schweri derives wind speeds from observing cloud formations, however, multiple challenges arise. A non-clouded area has a severe lack of wind information, which the algorithm has to fill in. Schweri then warns of solution ambiguity, where multiple wind fields form a solution to the sparse cloud observation set [19]. Furthermore, noise is present within the observations. The research of Schweri et al. aimed to address this by so called 'data-driven regularization' [19]: a neural network learns how to recover data from occluded yet simulated flows. Thus, the actual solution is known, implying a supervised learning procedure. This is combined with physics-inspired regularization, which requires inclusion of the governing equations in one way or another. Also the network architecture must be constructed to support the wind recovery.

Network Architecture

Schweri et al. propose a network consisting of three parts: an encoder, followed by a dense block, and then a decoder, according to a U-Net structure [18] [19]. This is where the velocity field is fed in, and a completed field is the output. The encoder reduces the resolution of the input field, and the decoder reconstructs the processed output of the dense block [7] back to its original resolution [19]; this is done to extract features on all scales, as explained by Schweri et al. A separate stream function block is also present, which uses stream functions. This output is concatenated with the predicted velocity field, after which a convolution layer puts out the overall wind prediction [19].

Physical Losses

The principle of physical loss can improve machine learning models aiming to estimate physical processes. Physical losses are present in the loss function. One or more partial differential equations (PDE's) could be written as a residual, which must be minimized, and added to the loss function. The goal is to achieve a solution which is physically accurate, but the label loss must still be included nonetheless: multiple physically correct solutions might come out of the PDE, which is then itself locally satisfied, but by a different solution than what the labels would dictate, as Thuerey [25] and Schweri [19] underline. Obviously, the optimization will become more expensive. This is due to the loss function having a more complex landscape according to Krishnapriyan et al. [11].

In order to make their neural network physics-aware, Schweri et al. propose a number of physics inspired loss functions [19]:

- An L1 reconstruction loss
- An L1 Jacobian loss
- A vorticity loss
- A divergence loss
- A magnitude of the gradient loss

Differentiable Physics
Numerical solvers suffer from discretization errors . The solvers can be put into neural networks, which can leverage their autodifferentation and backpropagation to compensate for the discretization errors, as explored by Thuerey et al. in [25] and Um et al. in [26]. A more elaborate description is given in the following.

Often, one wishes to model a physical system, of which the governing PDE's are known. A numerical solver can simulate the system by discretizing those PDE's. Due to this discretization, a numerical error is introduced, which is hard to analytically quantify as explained by Um et al. [26]. They propose the usage of neural networks, which can learn the structure behind the discretization error. Um et al. give the following explanation.

The goal is to approximate a function u by discretizing a PDE P, with timestep Δt. Two discretizations are considered: an accurate one, P_r, having solutions r on a reference manifold [26]. Another approximate discretization, P_s, has solutions s on a source manifold [26]. Both are shown in Figure 2.2. Solution sequences can be constructed by:

$$r_{t+\Delta t} = P_r(r_t) \tag{2.15}$$

$$s_{t+\Delta t} = P_s(s_t) \tag{2.16}$$

Um et al. [26] introduce a mapping operator, T, which maps the reference solution to the source solution, $s_t = Tr_t$. In general, the reasoning $P_s(Tr_t) = Tr_{t+1}$ does not hold, as of the numerical errors. That is, applying the coarser P_s scheme in combination with the mapping T, does not solve the discretization problem, and the error might grow exponentially [26]. An L2 loss is defined as: $L(s_t, Tr_t) = ||s_t - Tr_t||_2$. In differential physics, the goal is to find a correction operator on the source solution, $C(s) = C(Tr_t)$ [26], which when put through the source scheme P_s, improves over the unmodified Tr_t in the L2 loss. This correction can be done through a neural network. Um identifies three cases [26].

The first way is to learn from the difference between the mapping of the reference solutions to the source manifold (Tr_t) on one side, and the source manifold itself on the other side. This is defined as purely supervised learning [26]. In Figure 2.2, this translates to learning the difference between the orange source solutions, and the blue reference solutions mapped on the source space [26]. A second way is to introduce a pre-computed correction on the source states, leading to a modified source state which should approximate the reference state. A neural network is then able to be trained by these modified source states [26]. The corrected solutions form the green line in Figure 2.2.

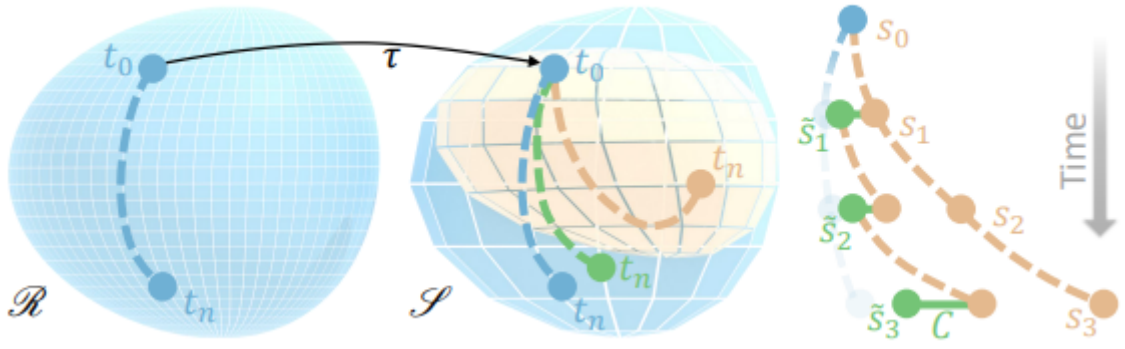


Figure 2.2: Courtesy of Um et al. [26]

The third method, solver-in-the-loop, is reached by letting corrections on the states explore the system, and receiving gradients on their performance [26]. The correction is now trained based on previous iterations, whereas a pre-computed correction cannot be trained. How to exactly implement the neural

network with the solver, is flexible. Thuerey et al. [25] mention the neural network can provide an input to the solver, and gradients then allow for backpropagation. The inverse is also possible, a PDE giving its output to a neural network, which is then modified to better resemble the true solution [25].

Um et al. report outperforming traditional solver-only methods for a number of problems, like advection and diffusion problems, and flow over objects [26]. However, implementation is non-trivial, and governing PDE's and solvers must be available.

2.3.2. Reduced Order Modelling

Another technique to simulate physical systems, is to capture its core dynamics through reducing the system to a select number of basis functions. Cao et al. used such a reduced order modelling technique in [4], namely proper orthogonal decomposition (POD), in a 4DVAR approach. 4DVAR was earlier discussed in this chapter, with the main takeaway being the expensive computational cost. Cao et al. therefore argue that reducing the control space could prove beneficial. In POD, one looks for the most energetic modes which dictate how the system behaves. Only considering these modes reduce the dimension of the system [4]. First, full order snapshots are needed, at multiple timestamps. Next, a set of basis functions is derived. They are the orthonormal eigenvectors of the spatial correlation matrix [4]. Small eigenvalues can be neglected, leading to a reduced system.

A more advanced ROM is dynamic mode decomposition (DMD). In this method, modes with spatial coherence and oscillations, decay or growth are selected, mentioned by Bai et al. in [1]. DMD is reported to be superior to POD because of the ability to both capture spatial and temporal modes.

2.3.3. PyDDA

An online Python library, Pythonic Direct Data Assimilation (PyDDA), was developed by Jackson et al. [8] to estimate wind in precipitation storms, based on Doppler radar measurements. Their library builds on the work of Potvin, Shapiro, Xue and Gao [16] [17]. Although this library thus has a different application, the methods and frameworks provided might prove useful in this thesis research. As the PyDDA framework is used as one of the methodologies, and its content is closely related to this methodology, the specifics are explained in chapter 5 instead.

3

Research Plan

In this chapter, the research framework of the thesis will be addressed. This includes research questions, objectives and potential sub-goals. This chapter is a continuation on the research plan made earlier for this thesis.

3.1. Knowledge Gap

Lots of research and literature on meteorology and forecasting is available. Aircraft observations are readily used in such schemes, to improve overall weather estimation. However, as illustrated in the work of Stoffelen and De Haan [5], the assimilation cycles are in the order of 1 hour. The knowledge gap proposed in this thesis research, is to find out whether a quicker and computationally low-effort method can be developed, with the aim of improving wind estimation for aircraft, based on incorporating aircraft wind observations with physical knowledge.

3.2. Research Questions

The main question for this thesis research is as follows:

”How can wind dynamics be modelled and combined with data gathered by aircraft, to provide accurate estimations of wind fields in an airspace?”

The following sub-questions elaborate on the main question:

Can wind estimation be improved by incorporating wind dynamics modelling into the Meteo-Particle model?

- How much improvement is noticeable? At what computational cost?
- Can the MP model be extended, or does it have to be replaced?

Which other approaches are available? Which ones seem the most promising to conduct research with?

- What advantages and disadvantages are inherent to the approaches?
- How much improvement is noticeable with the new approach and at what computation cost?

Are the improvements worth the required expense (training, simulation, design hours, etc) to be put to further work/implementation?

3.3. Research Objectives

The main research objective is to research whether improvement of wind field estimation by combining aircraft measurements with wind dynamics is achievable.

3.4. Sub-objectives

A number of intermediate objectives must be defined in order to achieve the main objective above.

In the previous research plan, the first objectives which had to be met were defining the problem, and performing a literature study to identify methods. These methods should be put through a trade-off process to select the most promising ones. Those objectives are all met as of this report.

For the thesis research itself, another list of objectives is defined:

- Acquire and process data, libraries
- Construct methods
- Test and validate methods
- Draw up conclusions and further recommendations

Aircraft observational data has to be acquired and properly processed in order to use. Also the correct software libraries must be installed. Afterwards, work on the selected methods can begin. Once completed, testing and validation should take place. As such, multiple iterations might be done, both adapting methods, and developing new ones. Results should be presented, and conclusions and recommendations must follow.

Along the way, proper progress reporting should be done on all sub-objectives. In the end, the main objective should be met: i.e. establishing whether wind field estimation can be improved by combining aircraft observations with wind dynamics.

3.5. Assumptions

A couple of assumptions were made in order to streamline the thesis research.

- Atmosphere is assumed stable, i.e. no turbulence or rapid changing events
- Pressure velocity is zero
- Incompressible flow is assumed
- Modelling of meteorological phenomena are not considered (e.g. frontogenesis, jetstreams)
- Coriolis force is not considered

3.6. Trade-off

The literature study identified multiple methods of tackling the knowledge gap. It was chosen to select two main methods. The first method should consist of already available building blocks, i.e. the MP model, GFS data, and/or PyDDA. This was later split up in two methods itself: first, the MP+GFS combination, and second, a PyDDA framework. As for the second main method, it was chosen to have a more scientifically innovative approach: for this, a neural network approach was chosen. Reduced order modelling is not considered as it was deemed too complex, and less innovative than the ANN approach.

4

Data Pre-processing

4.1. Aircraft measurements

Aircraft observational data used in this thesis research is the same data Sun et al. used for their MP model paper. The data comes in the form of a dataframe, with the following keys [23]:

- Timestamp of observation
- ICAO number of aircraft
- Latitude of observation
- Longitude of observation
- Altitude of observation [ft]
- Observed zonal wind speed [m/s]
- Observed meridional wind speed [m/s]
- Observed Temperature [K]

Zonal wind speed is the wind speed blowing from West to East. Meridional wind speed goes from South to North. Coordinates are expressed in altitude, latitude and longitude, but can be transformed to Cartesian x-y-z values by mathematical transformations, provided by Sun et al. The given altitudes are barometric altitudes, meaning they depend on pressure measurements. The measurements are collected from 01/01/2018 until 10/01/2018, around 00 hours, 06, 12 and 18 hours.

4.1.1. Error Rejection

As of the writing of this report, the error rejection mechanism used in this thesis research consists of simply omitting any wind measurements above 75 m/s. In the near future a better rejection algorithm should be constructed. Sun et al. propose probabilistic rejection.

4.1.2. Training and Validation Set Splits

Some of the observations will be used to train the models, others must be used to construct a validation set. More information on validation methods can be found in chapter 6. In terms of code, the data is simply split based on either splitting at random, or splitting the aircraft into training and validation aircraft. The former thus takes training and validation data from the same plane.

4.2. GFS Data

NWP forecasts are readily available, and can be used to augment wind field estimation. Two systems are widespread, namely the Global Forecast System (GFS), and European Centre for Medium-Range Weather Forecasts (ECMWF). Both are used in Sun et al. . At the moment, this thesis research only uses GFS, as difficulties are present in obtaining ECMWF data.

4.2.1. GFS Availability

GFS provides both 3 hour and 6 hour forecasts, done at zero hours, six, twelve and eighteen. An analysis is also done at those times. Aircraft data is available for those hours as well. It is decided to use the 6 hour forecast as GFS reference (so the forecast for eighteen hours is the one made at twelve hours). The data were obtained from the National Centers for Environmental Prediction, National Weather Service, NOAA, U.S. Department of Commerce [14].

Many parameters are retrievable. The ones currently used from the retrieved data are:

- Absolute vorticity on isobarics
- Geopotential heights on isobarics
- Temperature on isobarics
- u-component wind on isobarics
- v-component wind on isobarics

They are given on isobarics, on a longitude-longitude grid with 0.25 degrees of spacing. The download represents the atmosphere from 40 degrees to 64 degrees in latitude (North), and 10 degrees West to 20 degrees East in longitude. An example image is shown in Figure 4.1. The temperature at the 100,000 Pascal isobar is given, at midnight 01/01/2018, with help of Panoply [21].

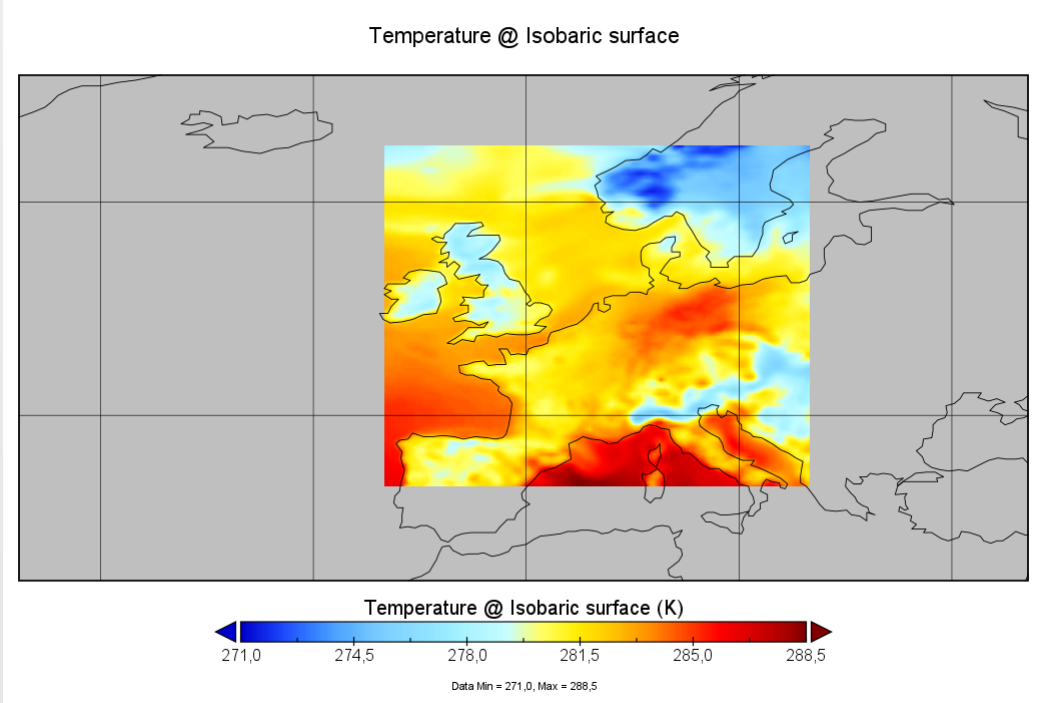


Figure 4.1: The temperature in Kelvin at the 100,000 Pascal isobar, for 01/01/18 at 00:00, as plotted by Panoply [21]

An isobaric is a surface of constant pressure. With knowledge of the geopotential height of each point of the isobaric, it is obviously clear how high that point is above the surface. Also absolute vorticity, temperature, and both wind components are available. 31 isobaric surfaces are given, with the next 31 pressures, in Pascal:

100000, 97500, 95000, 92500, 90000, 85000, 80000, 75000, 70000, 65000, 60000, 55000, 50000, 45000, 40000, 35000, 30000, 25000, 20000, 15000, 10000, 7000, 5000, 3000, 2000, 1000, 700, 500, 300, 200, 100

4.2.2. GFS Retrieval

The data comes in the format of $.grib2$ files. This data can be plotted by NASA's Panoply programme, as done above. This is useful for validating whether the Python code written can retrieve the data correctly.

4.2.3. GFS Transformation

The aircraft data observations height coordinate is barometric altitude (in [ft]). The aircraft does not measure height directly, but measures pressure instead, and calculates the height assuming ISA conditions; that is, 101325 [Pa] at mean sea level. In order to ensure the correct forecast for an observations height, the heights corresponding to the pressure levels according to ISA are considered. For each grid point, a cubic spline is calculated in the vertical; relating the isobaric pressure level data to the ISA height corresponding to that pressure. As the aircraft is also assuming ISA conditions, it is possible to calculate the conditions on any height, according to the GFS forecast.

Another spline transformation was done to convert from a longitude-latitude grid, to a cartesian grid. This was done with SciPy's $RectSphereBivariateSpline$ functionality. All in all, this implies that the algorithm is working with horizontal cartesian coordinates, while the vertical coordinate is an ISA pressure height. This pressure height does not necessarily remain flat in real geometric space, but offers a good approximation of wind flow on isobaric surfaces.

5

Methodology

In this chapter, the current methodologies developed will be put forward. As of writing this preliminary report, three main methodologies are present concerning wind estimation by aircraft measurements:

- The Meteo-Particle model by Sun et al. [23]
- Combining the Meteo-Particle model with GFS forecasts
- PyDDA model [8], helped by both measurements and GFS

Another methodology, a physics-aware neural network, inspired by the work of Schweri et al. in [19] will be researched during later stages of the thesis.

5.1. General Set-Up and Data Processing

The research goal of this thesis is to establish whether it is possible to improve (over) the Meteo-Particle model by incorporating physics knowledge; logically, newly developed methodologies must then be compared against the MP model, using the same training and the same validation set.

To ensure equal training/validation sets, a data processing function is ran first, which loads the training set, the validation set and the GFS processed data grids at the specified time. After this, different methodologies can be executed on the same data. A grid was already set up in pre-processing, with resolution in both horizontal and vertical dimensions. This grid will be used by the GFS data processing, and by PyDDA, one of the methodologies employed.

5.2. Meteo-Particle model

The methods of the MP model are fully explained by Sun et al. in [23], and a short introduction is given in chapter 2 of this report. The methodology remains unchanged, except for the fact that the training/-validation split follows a different philosophy, namely a split in training aircraft and validation aircraft, instead of a random data split. Also, the MP model was adapted to estimate beyond its confidence bounds, to offer a comparison to other methodologies. An RMSE metric for inside the confidence zone is however retained.

5.3. GFS and MP model Combination

As the MP model provides poor wind and temperature estimation in sporadically observed areas, a starting point might be to fill in those gaps with GFS data.

5.3.1. GFS interpolation

GFS information comes in the form of gridded data, which was transformed earlier. To retrieve a GFS wind/temperature estimate at a certain coordinate, an interpolation method can be applied. For this purpose, SciPy's `RegularGridInterpolator` is used, with simple linear interpolation selected. Therefore, GFS estimates are easily available.

5.3.2. Combination

The next step consists of combining this GFS estimate with what the MP model suggests. One must balance between the two estimates, based on a proper metric. The MP model gives a confidence level for the evaluated coordinate. This confidence level depends the number of involved particles, their weights, and the uniformness of particle headings [23]. The particle weights themselves depend on their distance to the point, the distance to their origin and their age, as explained by Sun et al [23].

Now, using the confidence as MP model weight, W_{conf}, the next equation determines the combined estimation:

$$X_{comb} = \frac{X_{GFS} \cdot \alpha(1 - W_{conf}) + X_{MP} \cdot W_{conf}}{\alpha(1 - W_{conf}) + W_{conf}} \tag{5.1}$$

The α term serves to shift weight more to GFS (>1) or to more to MP (<1), regardless of what the confidence level is. This can be used in hypertuning. This calculation is done for both wind components, and temperature.

5.3.3. Defining validation zones

To gain better insight into the methods performance, the idea of validation zones is implemented. The MP model has a confidence zone where estimates are regarded as valid. Outside of this zone, the MP model does not give estimates anymore. Three zones are defined: the confidence zone itself, the outside zone, and the total zone (both inside and out). The MP model was adapted to also provide estimates outside its confidence zone, for purpose of comparison. Lastly, an estimate by pure GFS interpolation will also be provided, again by SciPy's `RegularGridInterpolator`.

5.4. PyDDA

Although the GFS combination from the above section does augments the MP model with physics knowledge (the GFS forecast stems from an NWP), it is only a forecast. An even more direct approach could be the PyDDA framework by Jackson et al.[8], based on the work of Potvin and Shapiro [16] [17].

5.4.1. PyDDA Framework

The PyDDA method was originally developed by Jackson et al. [8] to estimate wind fields in storms, by using Doppler radar measurements. Thus, modification of the code was necessary in order to accommodate aircraft observations. Yet, PyDDA allows Tensorflow gradient descent optimization and therefore is chosen.

5.4.2. Cost Functions

The cost functions reflect how well the current solution answers to the cost functions underlying requirements. The lesser, the better the solution meets this requirement. PyDDA has a number of built-in cost functions, but also allows for others to be added. The ones currently used for the research are:

- mass continuity (PyDDA)
- Smoothness (PyDDA)
- Point observations (PyDDA, modified)
- Vorticity (PyDDA)
- GFS background (added)
- Aircraft measurement background (added)
- GFS boundary constraint (added)

Not all cost functions have to used necessarily during optimization. Additionally, the ability to use a mask on TensorFlow gradients was developed. This means that the optimization algorithm cannot enact changes upon masked variables; this can be applied to areas in the wind field which have been measured.

Mass continuity

Mass continuity is the physical principle which states that, for a finite volume, the inflow should equal the outflow (given their is no accumulation/decrease). PyDDA will take the divergence of the wind as cost function, joined by the anelastic term. Originally, PyDDA is developed for use with geometric coordinates. The cost function in that case is [8] [16] [17]:

$$J_c = \Sigma(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} + \frac{w}{\rho}\frac{\partial \rho}{dz}) \tag{5.2}$$

However, for this thesis research, isobaric surfaces are assumed, with no vertical wind component. Therefore, the equation reduces to:

$$J_c = \Sigma(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}) \tag{5.3}$$

In [6], Holton and Hakim give the same equation for isobaric surfaces, but with the addition of a term $\frac{\partial \omega}{\partial p}$, ω being the change in pressure following the motion, basically the equivalent of $\frac{\partial w}{\partial z}$ in geometric coordinates [6]; but this term is omitted as vertical motion is not considered as of now in the thesis research. Schweri et al. [19] also use 2D wind fields only; to confirm the validity of the assumption. The modification in PyDDA is to set all gradients of w to zero. To demonstrate the effect of the mass flow cost function, an example is shown in Figure 5.1. The aircraft measurements indicate a lower zonal speed than GFS (U0-GFS), where U0 is the initial template consisting of measurements put over GFS. A too low weight was set on the aircraft measurements, and thus the mass contuinity simply filled in the gap, as can be seen on (UF-U0), i.e. it added back zonal wind speed, resulting in UF being smoother. This example is unsatisfactory as it simply reverses the initial estimate back to the GFS estimate, but it illustrates the working of the mass continuity cost function.

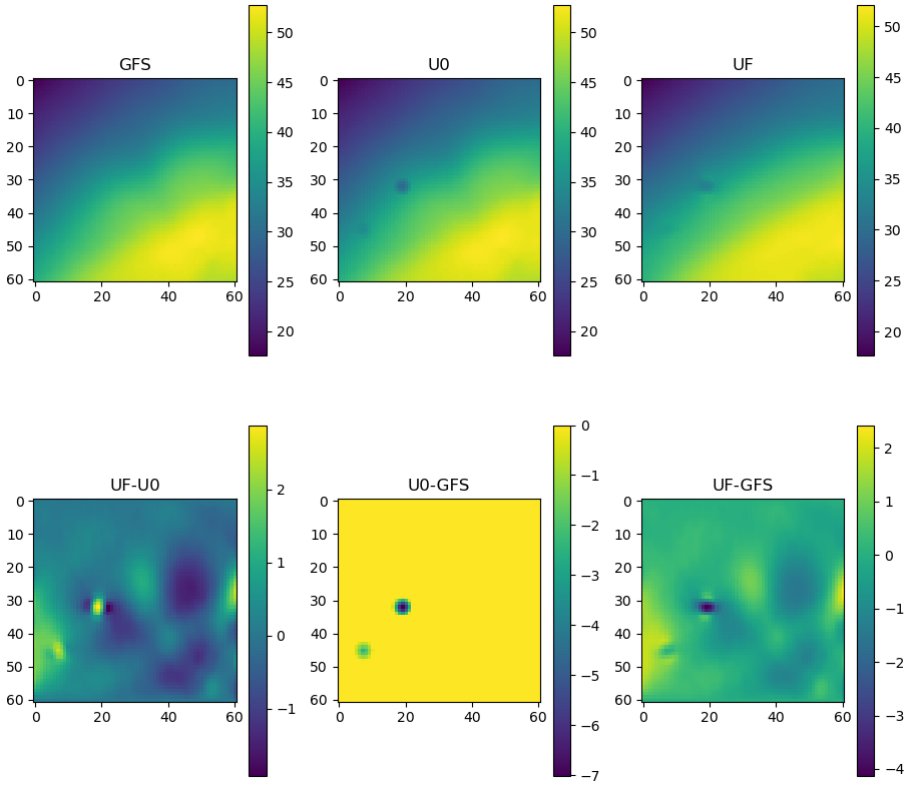


Figure 5.1: An example of a 'gap fill', illustrative purpose only

Smoothness

PyDDA includes a smoothness cost function, taking into account the smoothness of the velocity field in both horizontal and vertical directions. This is done by taking the squared Laplacian of the wind fields

[8] [16] [17]. This can be expressed as:

$$J_{smoothness} = \Sigma(\nabla^2 \vec{V})^2 \tag{5.4}$$

PyDDA allows for allocating coefficients to the directions [8], leading to an expanded version being:

$$J_{smoothness} = \Sigma(C_x(\frac{\partial^2 u}{\partial x^2})^2 + C_x(\frac{\partial^2 v}{\partial x^2})^2 + C_y(\frac{\partial^2 u}{\partial y^2})^2 + C_y(\frac{\partial^2 v}{\partial y^2})^2 + C_z(\frac{\partial^2 u}{\partial z^2})^2 + C_z(\frac{\partial^2 v}{\partial z^2})^2) \tag{5.5}$$

Note that vertical wind speed, w, is not considered in this assignment, but is still incorporated in the PyDDA framework.

Point Observations
An observation cost function is included in PyDDA with the goal of adhering to measurements at a certain location. The observation cost function was modified to simply determine the grid point closest to the observation, instead of checking the entire domain. The point observations are also modified to be done stochastically. The cost function is [8]:

$$J_{point} = \Sigma((u_{analysis} - u_{point})^2 + (v_{analysis} - v_{point})^2) \tag{5.6}$$

with the analysis component being the wind field during the PyDDA optimization.

Vorticity
Potvin and Shapiro [16] [17] also introduce vorticity into their data assimilation work. The following cost function is computed:

$$J_{vorticity} = (u - U_t)\frac{\partial \zeta}{\partial x} + (v - V_t)\frac{\partial \zeta}{\partial y} + \zeta(\frac{du}{dx} + \frac{dv}{dy}) \tag{5.7}$$

Where $\zeta = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$, and U_t and V_t are the components of the storm motion [16] [17]. They are assumed to be zero in this assignment.

Backgrounds
PyDDA also allows to calculate a cost function based on deviations from a model. This is simply the difference between the analysis field and the field of the model:

$$J_{model} = \Sigma((u_{analysis} - u_{model})^2 + (v_{analysis} - v_{model})^2) \tag{5.8}$$

This cost function is used for two models: the GFS background, and an aircraft measurement background. The latter one is a background which is extrapolated from aircraft measurements. Their implementation is discussed later in this chapter.

GFS Boundary
An additional cost function was made to make the analysis fields observe the boundary conditions imposed by GFS. This follows the exact same formula from the background cost function, albeit only in the boundary region.

5.4.3. Optimization

The PyDDA framework makes use of the L-BFGS (Limited memory Broyden-Fletcher-Goldfarb-Shanno) algorithm to optimize its solution. This algorithm is a limited memory version of the BFGS algorithm. A general explanation of BGFS is given below, based on Nocedal and Wright [15].

Suppose the goal is to minimize an objective function f, with current iterable x_k. A quadratic function can be set up, which has to be minimized:

$$m_k(p_k) = f_k + \nabla f_k^T \cdot p + \frac{1}{2} p^T \cdot B_k \cdot p \tag{5.9}$$

B_k is the approximate Hessian, p is the search direction which minimizes the quadratic function, which equals:

$$p_k = -B_k^{-1} \nabla f_k \tag{5.10}$$

Along this direction, the iterate x_k is updated. The idea of BFGS is to use an approximate Hessian instead of the real Hessian, which is not calculated every iteration, but updated according to multiple constraints. L-BFGS takes it a step further and only stores the approximate Hessian implicitly by keeping track of iterate x_k steps and gradient steps. A more elaborate explanation is given by Nocedal and Wright in their book on numerical methods [15].

TensorFlow
TensorFlow is a machine learning library compatible with Python, and is applied by PyDDA to calculate gradients of the cost functions with respect to wind field tensors. The L-BFGS optimization algorithm is provided as a function by TensorFlow, simply requiring the gradient and cost functions, the initialization, and stopping conditions.

As mentioned earlier, the ability to mask analysis fields was implemented; this means setting the gradients p to zero for those areas. This can be done for areas measured by aircraft, which then remain unchanged by the optimization.

Savitzky-Golay Filters
After the optimizing search is completed, PyDDA allows Savitzky-Golay filtering of the output wind field, in all three dimensions for all three wind components. This serves as a low-pass filter in the field, to filter out high-frequencies. The use is optional.

5.4.4. Modifications
As the PyDDA code was originally developed for another purpose, some modifications had to be made to the code. Some modifications have already been discussed, but are summarized here again for the purpose of clarity. The first change concerned the input wind field, which for PyDDA is a PyART grid object; a tailor-made object for Doppler radars. This was changed to a numpy array. Furthermore, the code was also modified to be able to include a GFS field as initial guess, for wind. Temperature is not considered at this moment. Also the cost function for point observations was modified. Earlier, the code made an influence box for every single observation, by traversing all coordinates. This was changed to simply rounding the observation coordinate to the nearest grid point.

Another modification for the point observation cost function is the introduction of stochastic gradient descent. This speeds up the process by only taking a sample of points at each optimization iteration. Two cost functions were added: one which tracks a background based on aircraft observations, and one which captures the GFS estimate. Also TensorFlow masking was implemented, giving the possibility of restricting change on certain parts of the analysis fields. Other than that, full acknowledgements are given to Potvin and Shapiro [16] [17] and the PyDDA team [8] for providing the framework and optimization strategy.

5.4.5. Implementation with Aircraft Observations
The key part of the PyDDA adoptation is combining it with observations and possibly GFS data in a successful way. Here, the methodology of implementing the modified PyDDA code will be explained.

To provide true nowcasting capabilities, it is decided to intertwine PyDDA with aircraft observations on a near-continuous basis. That is, observations are gathered over a short duration (in the order of minutes), PyDDA is applied directly afterwards, on an initial template. The final result will serve as the initialization for the next assimilation step.

Update step
Multiple fields are defined.

- The GFS fields (x and y direction winds on isobarics), vertical wind is assumed zero
- An aircraft observation weight field: assigns weight to aircraft measurements, based on frequency and age of measurements
- An aircraft observation field: What the aircraft observed values dictate, limited by distance
- A combined GFS-aircraft field

The GFS fields are readily available from earlier pre-processing steps. The aircraft observations are installed on a field as well, in an adaptable radius around the observation point. The aircraft observation weight array is accordingly updated on these observation points. This weight array includes a time decay factor: the update is carried at as such that the weight of the previous measurements diminish, in case of $C_{delay} < 1$.

$$W_{ac_t} = C_{decay} \cdot W_{ac_{t-1}} + \Sigma W_{obs_t} \tag{5.11}$$

W_{ac} is the aircraft observation weight field, and W_{obs} is the weight of a particular observation; adding the sum of all those to the previous weight field, results in the new observation weight field. To better visualize this process, an example is given in Figure 5.2. Two flights are clearly visible, over a period of 5 minutes. The value in the colorbar is the value of the weight. It can be seen that less weight is assigned to points far away from the actual aircraft.

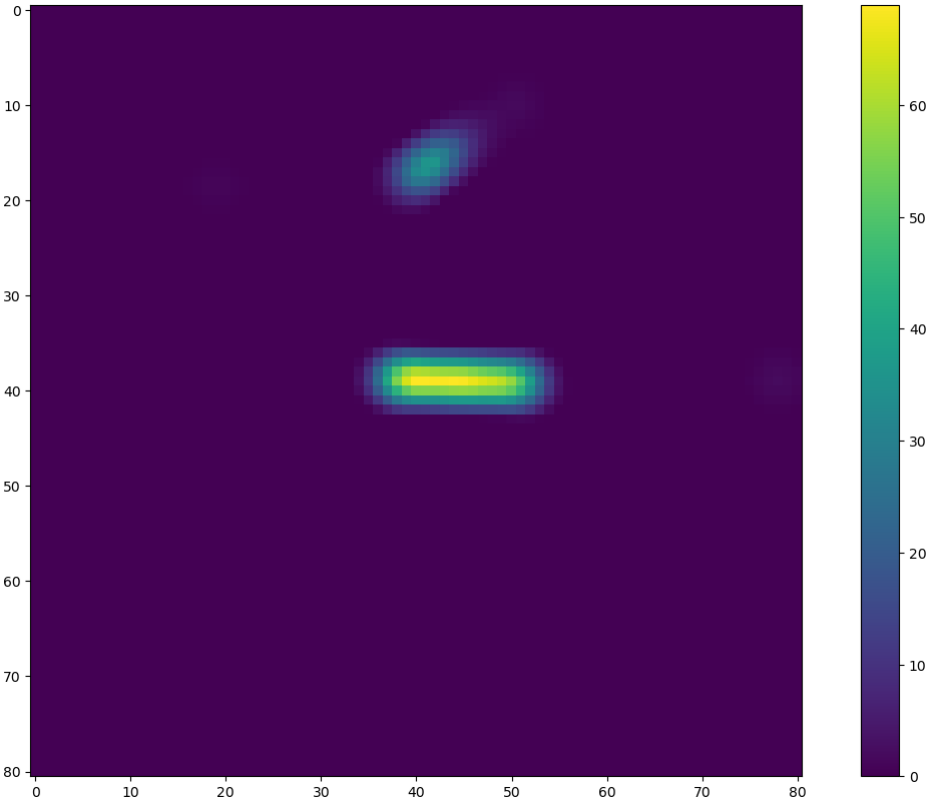


Figure 5.2: Aircraft Weight Array

Next, the meteorological field must be determined. This consists of combining the previous field with the new observation information.

$$X_{ac_t} = \frac{C_{decay} \cdot W_{ac_{t-1}} \cdot X_{ac_{t-1}} + \Sigma W_{obs_t} \cdot X_{obs_t}}{C_{decay} \cdot W_{ac_{t-1}} + \Sigma W_{obs_t}} \tag{5.12}$$

An example is illustrated in Figure 5.3, where the u-component of the wind according to aircraft measurements is shown. This corresponds to the weight locations from Figure 5.2. Two other points are shown, which are not clearly visible in the weights figure.

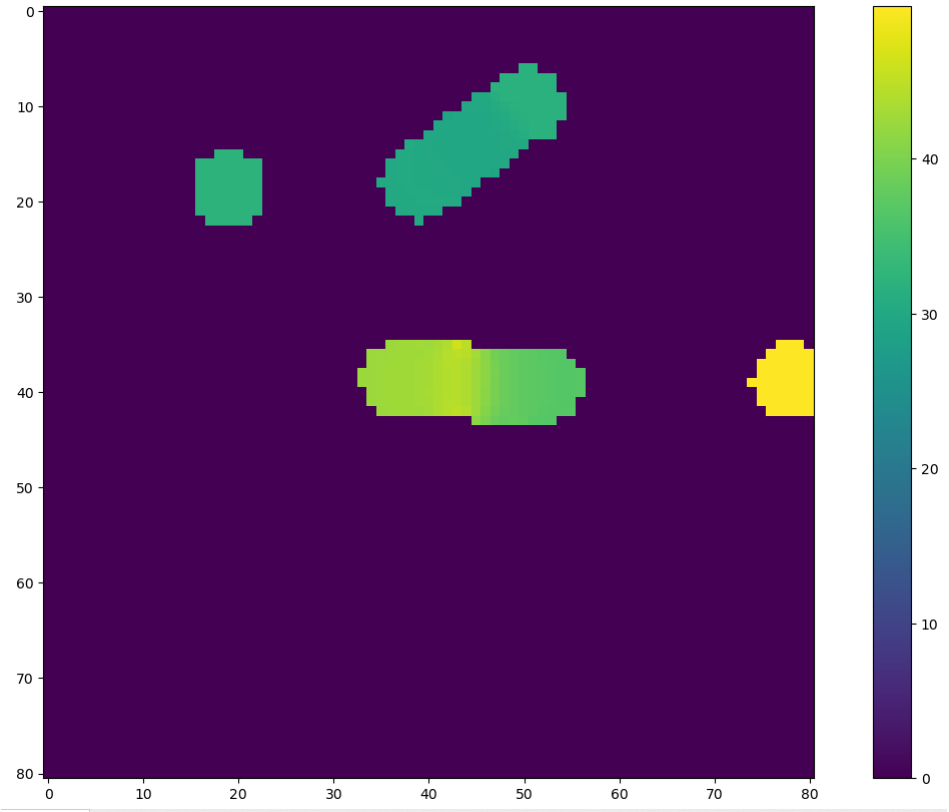


Figure 5.3: Aircraft measurement background

To combine the aircraft estimation with the last PyDDA analysis estimation, the following formula is used:

$$X_{comb_t} = \frac{W_{ac_t} \cdot X_{ac_t} + (max(0, \alpha) - W_{ac_t}) \cdot X_{an}}{W_{ac_t} + (max(0, \alpha) - W_{ac_t})} \tag{5.13}$$

α is a coefficient to express the strength of aircraft measurements compared to GFS. A higher value means a higher relative weight for the GFS field.

Initialization

As a first guess, the GFS wind field is used, combined with the first available aircraft observed values. This is done via the very same equation which combines analysis fields with the aircraft observation field.

$$X_{comb_t} = \frac{W_{ac_t} \cdot X_{ac_t} + (max(0, \alpha) - W_{ac_t}) \cdot X_{GFS}}{W_{ac_t} + (max(0, \alpha) - W_{ac_t})} \tag{5.14}$$

In Figure 5.4, the GFS prediction for the u-component is seen, together with $U0$, the initialization combining GFS and the aircraft measurement background. The edges between GFS estimate and the aircraft measurements are crisp in Figure 5.3. This can be made smoother by increasing α in Equation 5.14. Another example with a higher value for α is given in Figure 5.5.

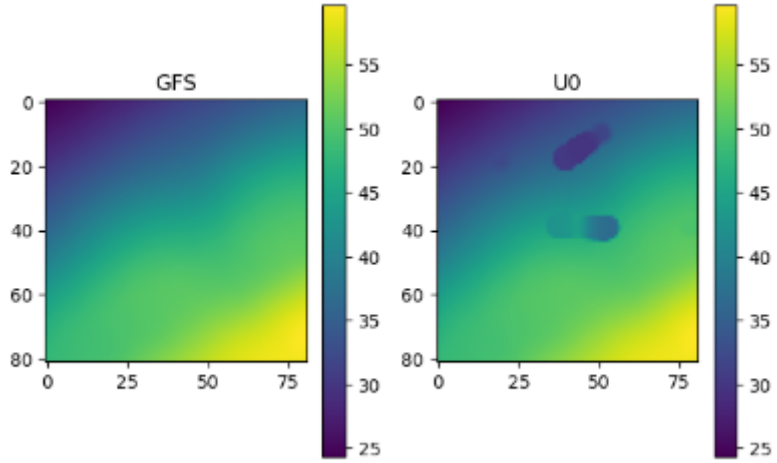


Figure 5.4: Initialization of u-component

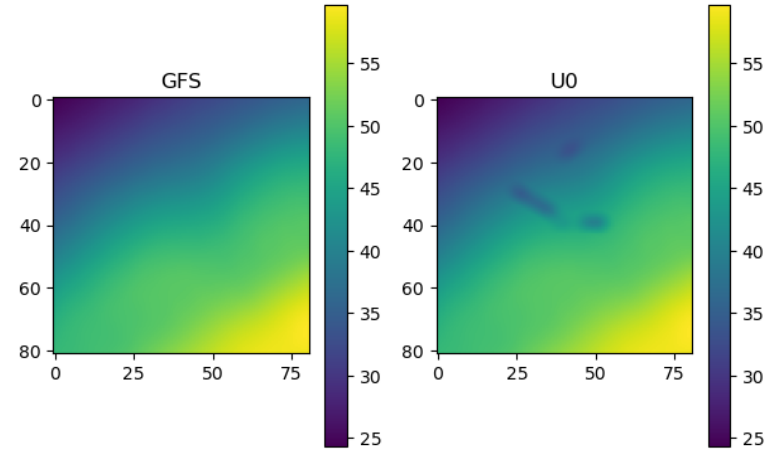


Figure 5.5: A smoother initialization. Different fligths are shown compared to Figure 5.3

Aircraft Observation Weight Determination

A certain observation at a certain coordinate must be put through to the PyDDA algorithm. Logically, the airspace in the immediate vicinity of the observation is likely to have a value similar to the observed value. This is true for both the horizontal and vertical scale. In the following, it is explained how W_{obs} is obtained.

First, the horizontal extent of the observation weight is determined. A certain radius is assumed. The observations horizontal coordinates are known: this allows for a distance calculation to each grid box. The weight is set to:

$$W_{obs_h} = max(0, \frac{R - \sqrt{(x_{obs} - x_{grid})^2 + (y_{obs} - y_{grid})^2}}{R}) \tag{5.15}$$

Next, the height difference must be included. This is done with a vertical radius of influence, R_v.

$$W_{obs_v} = max(0, \frac{R_v - abs(z_{obs} - z_{grid})}{R_v}) \tag{5.16}$$

Finally, the two weights are combined by element-wise multiplication.

Now, the total aircraft weight array can be computed, it being a 3D array. For each observation, a weight is computed, added to the total weight array, after which an updated aircraft observation wind

field is determined. Finally, this is aircraft estimate is combined with the initialization (either GFS, or a PyDDA result from a previous run).

Assimilation Time Windows
Earlier, aircraft data was split in a training set, and in a validation set. The training set forms the collection of aircraft known to the algorithms. Each observation has a timestamp, beginning at 30 minutes before the GFS hour, and ending maximally 30 minutes after. The time dimension is important to consider: the research goal is to provide nowcasting capabilities, which are very short forecasts essentially. De Haan and Stoffelen used a 1 hour assimilation cycle in [5], while the Meteo-Particle model of Sun [23] works on an actual real-time basis. The PyDDA assimilation time can set from being a couple of seconds, to in theory one hour. In the latter case, one assimilation cycle only is performed.

Also for validation, time is of importance: if one wishes to check algorithms on nowcasting capability, one can select validation set observations stemming from the same assimilation cycle the training observations were used for, or potentially a cycle later. One can also evaluate short term prediction, by taking validation set observations of e.g. 30 minutes later than the training set observations used.

Aircraft Ensembles
For a given time window, a certain number of aircraft are in the training set, and another number in the validation set. Sometimes, the aircraft observations are erroneous or contradict each other. To counter this, inspiration is taken from the idea of ensemble forecasting, where different models are installed with different initial conditions and boundary conditions, and the one with the best fit to observations is taken. For the PyDDA method considered in this thesis assignment, it could potentially prove beneficial to consider multiple ensemble members for a certain time window, built up of a different set of training aircraft. Selecting the best ensemble member could be done through a separate test set, including or not (fully) including the training set. As of writing the preliminary report, the full deployment of this technique remains an idea, but might prove interesting in future work.

6

Preliminary Results

In this chapter, preliminary results on the thesis research are presented. The results come in the form of a comparative study between the previous MP model and newly established methods. The previous MP model is taken, then compared against a pure GFS, an interpolated GFS, and a PyDDA result. Tuning of parameters in the latter is also performed.

6.1. Results Methods

A crucial part of establishing a valid comparison between MP model and other methods, is to clearly define a proper validity measure, and how it should be calculated.

6.1.1. Training and Validation Split

The way how training and validation data are split, is important as it may underline the validity of the method for a certain application. If one wishes to know local area wind fields, around aircraft, it might be enough to perform a random split of the data; that is, an aircraft is used both for training and for validation. Sun et al. applied such a data split [23].

On the other hand, splitting on an aircraft basis might prove useful in validating the performance of the models in faraway regions, or later timestamps. This means splitting aircraft themselves in a set of training aircraft, and a set of validation aircraft. The model will never see any validation aircrafts data to train with, as opposed to the random data split. In this thesis research, it is chosen to have the latter way of setting up training and validation data. It allows for effectively establishing whether better estimates can be made for far away reasons, and it is more practical for flight performance to derive validation metrics this way.

6.1.2. Error Metric

Root mean squared error (RMSE) is chosen, and is constructed by summing up the squares of all differences between prediction and validation measurements, and taking the root of this. Furthermore, visual inspection can be used on wind estimate plots to assess whether this solution is physically consistent.

6.2. Parameters

For each methodology, many parameters are tuneable and are very likely to have an effect on the methodologies overall performance.

6.2.1. Meteo-Particle Model

The foremost parameters to tune are the particle count and the particle age parameter (related to time decay). Additionally, the GFS combination has a parameter to tune how the weight should be distributed between the MP model, and the GFS estimate. This is set to 1 as a standard. This implies that the weight of the GFS is e.g. 0.3 when the confidence of the MP estimation is 0.7. A higher value than 1 favours the GFS estimate. The value is set to 1.

6.2.2. PyDDA

PyDDA has many tuneable parameters, mainly relating to cost functions of the gradient descent, and how strong and expanded aircraft measurements are, compared to GFS initialization and previous assimilation cycle results. This also includes the ability to mask certain areas to prevent them being changed by the gradient optimization. Additionally, this masking is only enforced for the mass continuity function; others such as smoothing can still change the otherwise masked values.

6.3. Results

During developing the PyDDA framework, many different distributions of weight and strategies were tried. However, none gave a significant improvement over a GFS-observation combination. That is, mass continuity could not improve in general over a simple combination of GFS estimates and observations. In the following, some attempts are demonstrated. Not all cost functions are included, as during testing, they were found to have no effect, or were of no purpose. This includes GFS background and GFS boundary, and point observations.

The height resolution is set to 1000 feet, starting at 34,000 and ending at 37,000 feet. This is chosen as the largest concentration of air traffic occurs on these heights, as can be seen in the histogram in Figure 6.1.

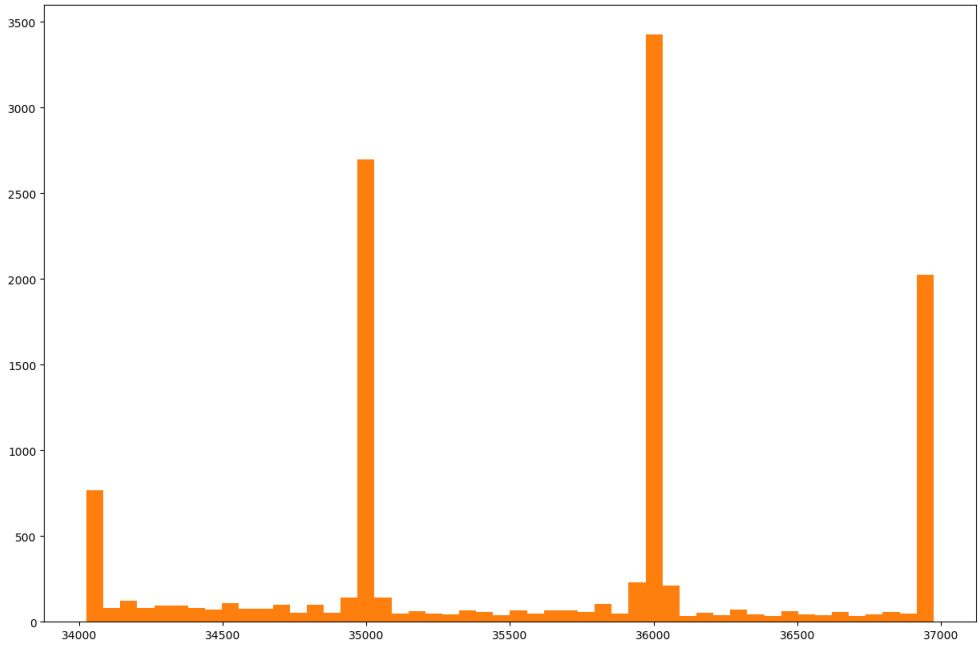


Figure 6.1: Height histogram of training observations for a 05/01 12h run, in [ft]

The horizontal resolution is set to 10 km. This is necessary to capture the smallest mesoscale effects, according to Klein [10]. Next to the regular PyDDA assimilation, an assimilation without continuity is done, and one without observations. This is to indicate whether continuity and observations actually provide any improvement. In the following, a few attempts are demonstrated, and a general discussion follows. The assimilation time is 5 minutes, leading to 12 windows over the hour. Estimates will be validated with validation points from the same 5 minute window.

Mass Flow and Smoothing
The first test considers the wind fields for 05/01 at 12h. The next weights and parameters are of concern:

- $C_{mass-continuity} = 1000^4$
- $C_{smooth} = 1000^5$
- $C_{ac-background} = 0.1$
- $R = 20000[m]$
- $mask = 0[m]$
- $\alpha = 2$

In Figure 6.2, the aircraft observational data locations can be seen. The blue lines constitute training data flights, orange is reserved for validation data flights. As can be seen the training/validation split is clearly aircraft-based. Validation data flights are also present in sporadically observed areas (the algorithms only see training data flights).

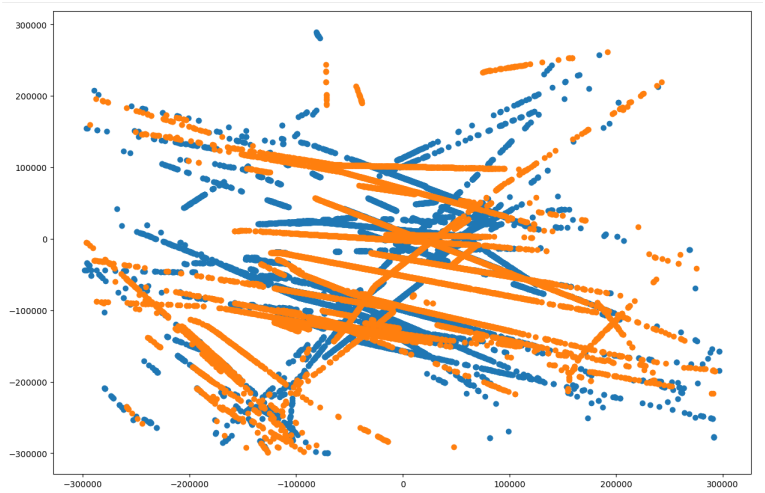


Figure 6.2: Training flights (blue) and Validation flights (orange). dimensions in [m]

First, the MP model results are considered. In Table 6.1, the RMSE's for a variety of configurations is shown. The first column addresses the MP model, the second gives the RMSE based off pure GFS, and finally the third column indicates the RMSE of the combined MP-GFS algorithm. The rows are divided as follows: the first is the confidence zone of the MP model, the second row is the out-of-confidence-zone, and the third row constitutes the total (both zones).

Mode	MP	GFS	COMBINATION
Confidence Zone RMSE	8.66 [m/s]	9.64 [m/s]	8.1 [m/s]
Out-of-zone RMSE	12.61 [m/s]	9 [m/s]	8.99 [m/s]
Total RMSE	9.76 [m/s]	9.6 [m/s]	8.36 [m/s]

Table 6.1: Meteo-Particle related results

The results reflect what has been observed many times: GFS beats the MP model outside the confidence zone, and the combination algorithm offers significant improvement over the MP model inside the confidence zone.

Next, PyDDA results are shown in Table 6.2.

Mode	Initialization	PyDDA	No continuity	no observations
RMSE	7.07 [m/s]	7.11 [m/s]	7 [m/s]	9.8 [m/s]

Table 6.2: PyDDA related results

Again, the results from this run reflect what was found overall: PyDDA with continuity cannot beat its initialization or no-continuity run. Those three however do beat MP and GFS models; this is likely due to the smoothing between measurements and GFS forecasts; it uses the same building blocks as the MP model, but seems to arrive at a better RMSE.

In the following, some additional figures are shown and commented on. The first figure, Figure 6.3, aims to provide a RMSE-distance relation. That is, the distance from a validation point to the nearest 5 training points is plotted against the RMSE of that validation point. This is done for both Meteo-Particle and PyDDA related models. The legend shows the corresponding model, and the RMSE is shown again. The x-axis is cut off at 100,000 meters (100 km).

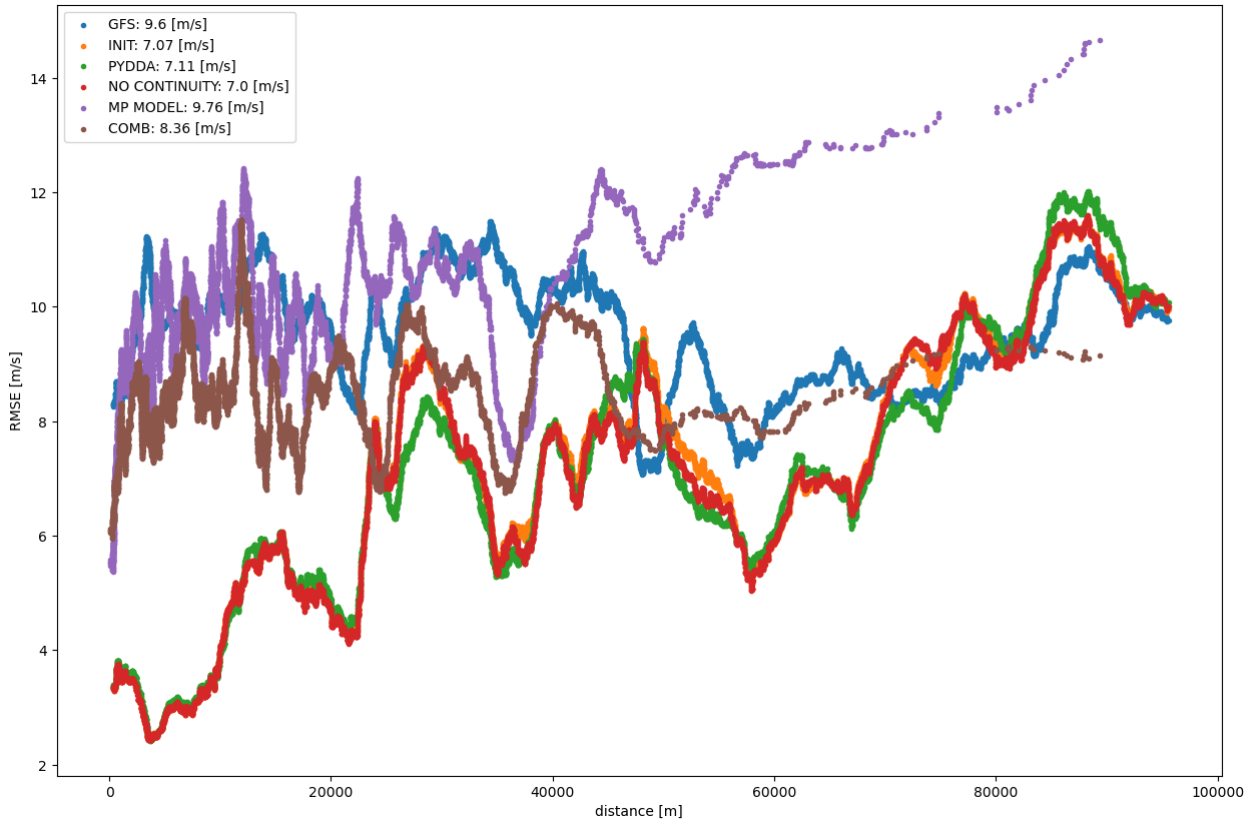


Figure 6.3: RMSE vs distance for different models

All PyDDA related models show close similarity, showing again that the continuity cost function does not result in a significant improvement. The PyDDA related models do however show improvement compared to GFS and MP related models, especially for the first 50 kilometers. Note that a moving average filter was used, with step size of 150, to smooth out the data; meaning a validation point at a certain distance is averaged with 150 points having the most similar distance. Also the combination of MP+GFS significantly improves over the MP model itself, which underperforms severely on longer distances. The combination model in itself scores better than GFS.

A second observation to be made, is that RMSE generally rises with distance, safe for GFS. This is logical, as all the models except GFS include measurement data, GFS however does obviously not depend on aircraft observations.

In Figure 6.4, the RMSE vs time of the validation point is given. Again, a smoothing average filter was applied to the data. For the PyDDA line, it can be seen error drops with time at first, but then suddenly rises from 1500 seconds onwards. This could be due to incorrect training or validation data, or validation data being further removed from training data than usual. Then, it drops again. Its RMSE

remains lower than the MP model most the time. Also the MP model plot suffers from large variation; again, this can possibly be attributed to training and validation data being incorrect, or that some validation data for some timesteps is far away from training data areas.

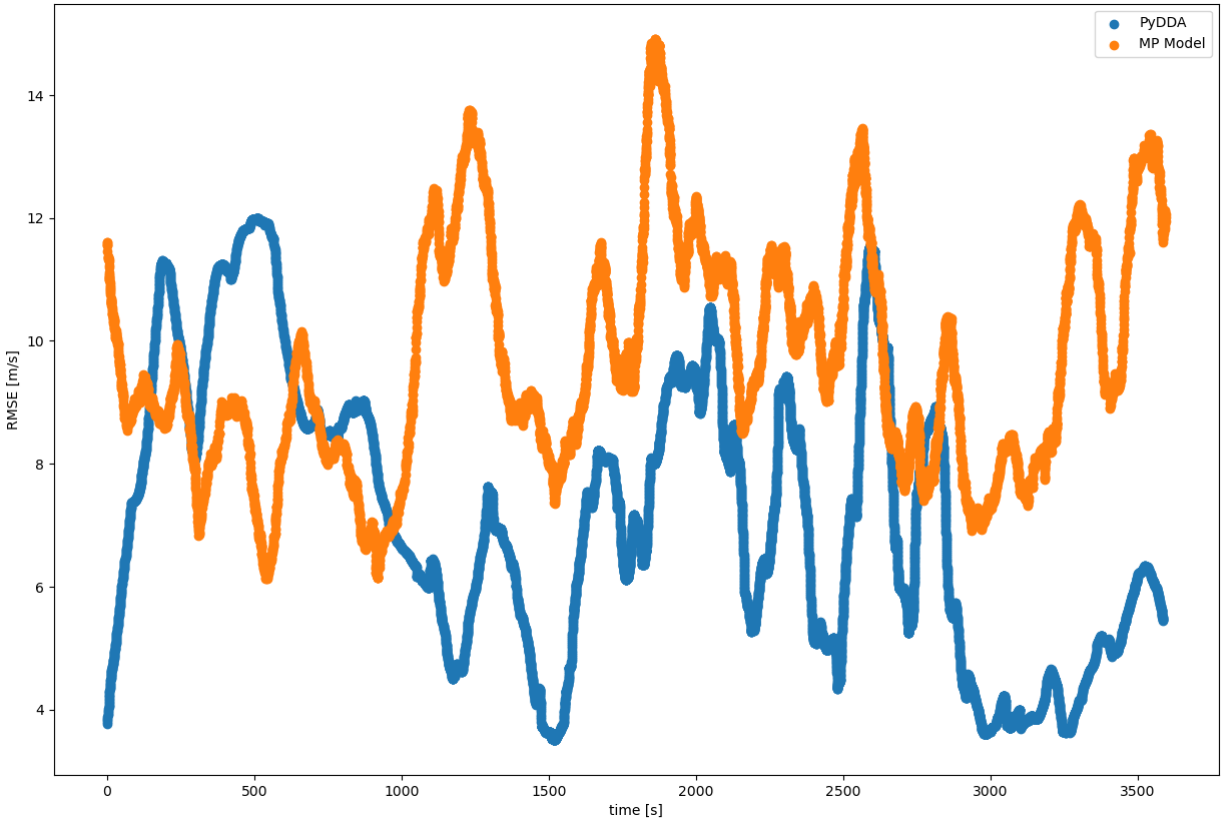


Figure 6.4: RMSE vs time

Finally, zonal wind speeds are shown for GFS, initialization $U0$ and PyDDA result UF for the first cycle (Figure 6.5), and the last assimilation cycle (Figure 6.6), at the 36,000 [ft] layer.

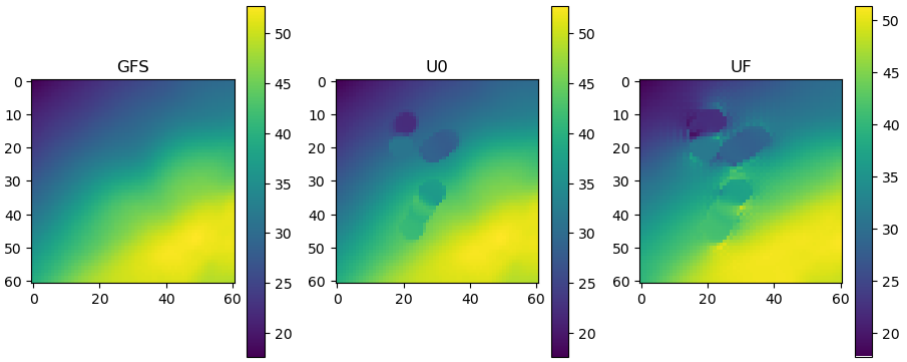


Figure 6.5: Zonal speeds for first assimilation cycle, colorbar in [m/s]. Axes show grid index.

For the first assimilation cycle, PyDDA smooths the field somewhat, but wind velocity keeps crisp boundaries, which seems physically unfeasible. In the last assimilation cycle, it can clearly be seen that the resultant wind field is a combination of all previous observations and their region of influence. This is not a smooth wind field, with sudden zones of higher or lower zonal speeds and clear distinctions between region of influences of measurements. This again looks physically infeasible. It is concluded

that any improvement in the RMSE stems from the combination of GFS with a region of influence from measurements, and not actually from a physics-based approach relating to continuity.

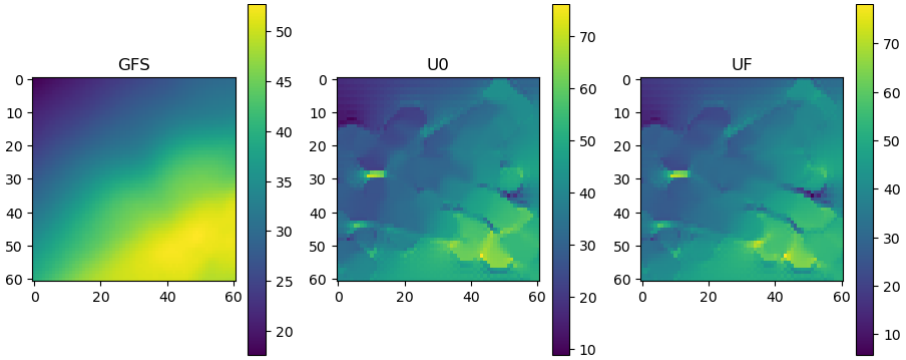


Figure 6.6: Zonal speeds for last assimilation cycle, colorbar in [m/s]. Axes show grid index.

6.3.1. TF Mask, Mass Flow and Smoothing

The next test is done on the same hour and date, but with different parameters.

- $C_{mass-continuity} = 1000^2$
- $C_{smooth} = 1000$
- $C_{ac-background} = 0.0001$
- $R = 15000[m]$
- $mask = 10000[m]$
- $\alpha = 2$

MP related results are given in Table 6.3, PyDDA related results are given in Table 6.4.

Mode	MP	GFS	COMBINATION
Total RMSE	9.46 [m/s]	8.4 [m/s]	7.82 [m/s]

Table 6.3: Meteo-Particle related results

Mode	Initialization	PyDDA	No continuity	no observations
RMSE	6.55 [m/s]	6.91 [m/s]	6.49 [m/s]	9.59 [m/s]

Table 6.4: PyDDA related results

In case for the Meteo-Particle results, GFS beats the MP model, but the combination of MP andGFS beats the GFS, proving that locality of the Meteo-Particle method can be combined with the advantage of GFS reliant estimates in sporadically measured areas. PyDDA however improves even more, albeit again not based on physics-inspired continuity, but rather the combination of GFS and aircraft measurements.

In Figure 6.7, the last assimilation window is shown for zonal wind speed fields, at 36,000 [ft]. Again, GFS, initialization and the PyDDA iteration are shown. The resultant field is less smoothed than the previous test, likely because of the mask. It also looks physically impossible.

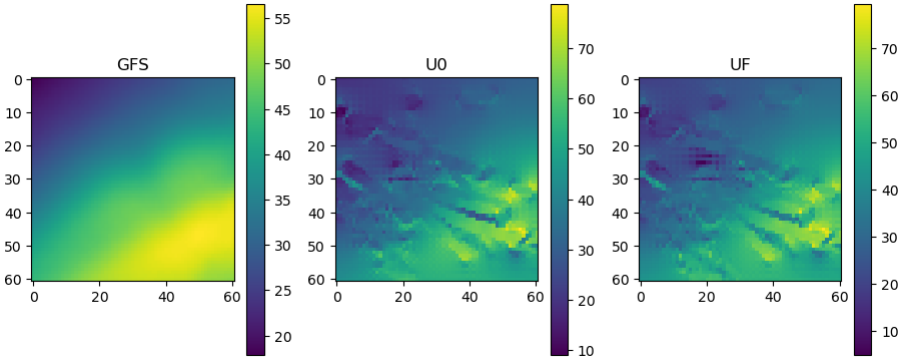


Figure 6.7: zonal speeds for last assimilation cycle, colorbar in [m/s]. Axes show grid index.

6.3.2. Vorticity

The next test is done on the same hour and date again, but the vorticity cost function is included.

- $C_{mass-continuity} = 1000^3$
- $C_{smooth} = 1000^2$
- $C_{ac-background} = 0.01$
- $R = 30000[m]$
- $mask = 0[m]$
- $\alpha = 2$
- $C_{vorticity} = 1000^3$

MP related results are given in Table 6.5, PyDDA related results are given in Table 6.6.

Mode	MP	GFS	COMBINATION
Total RMSE	9.02 [m/s]	8.35 [m/s]	7.56 [m/s]

Table 6.5: Meteo-Particle related results

Mode	Initialization	PyDDA	No continuity	no observations
RMSE	6.4 [m/s]	6.41 [m/s]	6.44 [m/s]	9.09 [m/s]

Table 6.6: PyDDA related results

Yet again, the same conclusion can be made. The combined MP+GFS method also is superior for this test. The MP related results don't depend on the PyDDA parameters, but are shown again nonetheless to illustrate its performance. Regarding the PyDDA related results, they all are in very close proximity to another, except for the no observations version, obviously. Vorticity was included, but the PyDDA run cannot improve over the initialization fields. The distance-RMSE relation is shown in Figure 6.8.

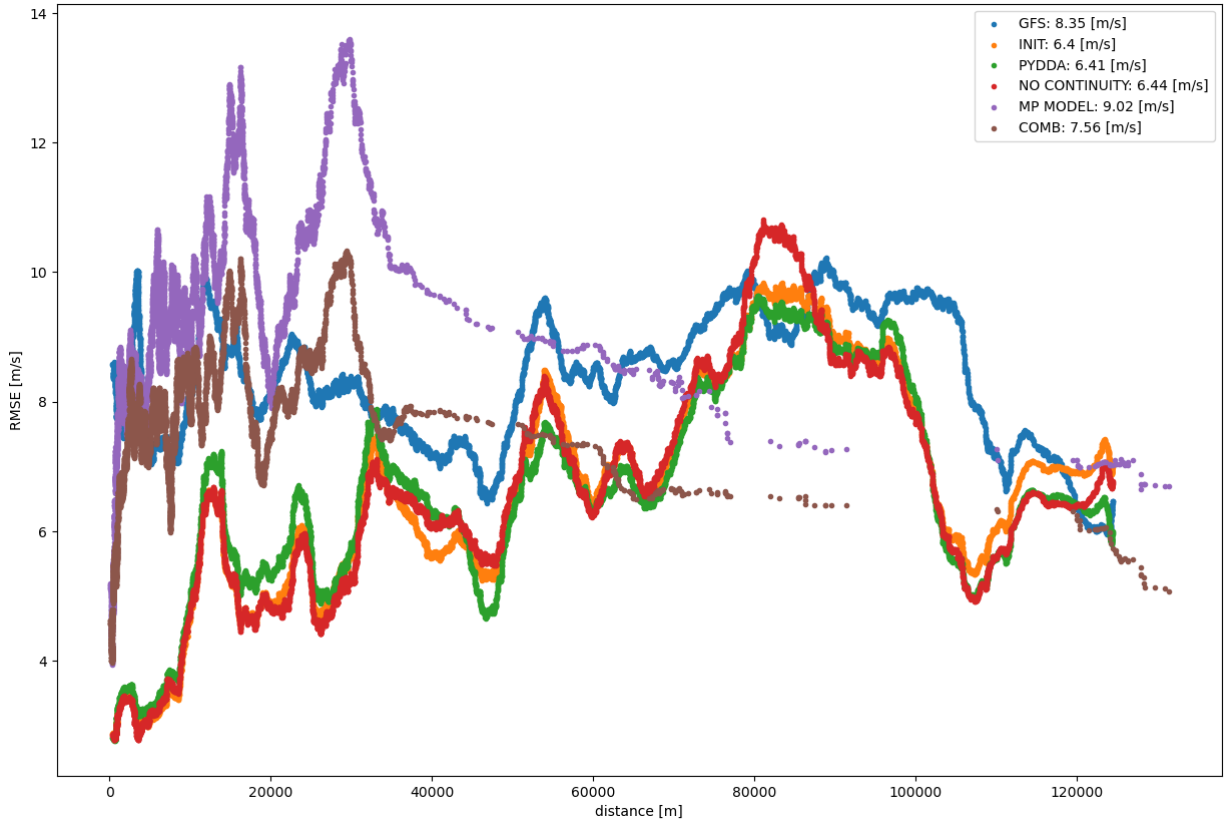


Figure 6.8: RMSE vs distance for different models

Both zonal and meridional components for the last assimilation cycle are given in Figure 6.9 and Figure 6.10, respectively. Yet again, altitude is 36,000 [ft].

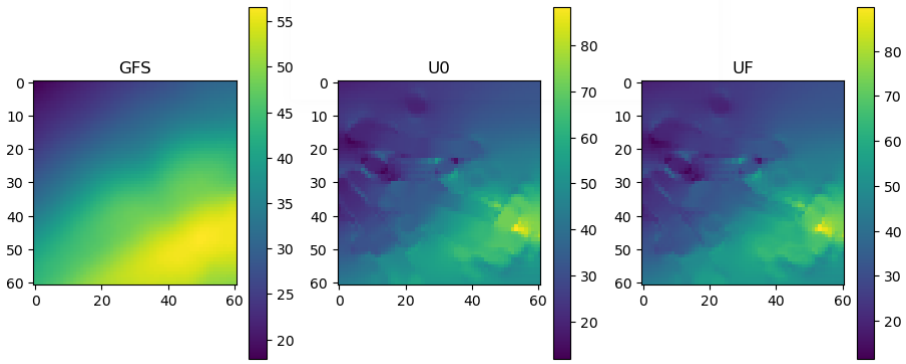


Figure 6.9: Zonal speeds for first assimilation cycle, colorbar in [m/s]. Axes show grid index.

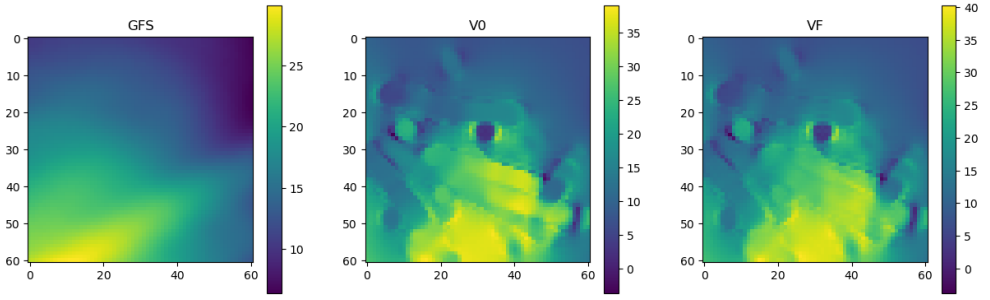


Figure 6.10: Meridional speeds for first assimilation cycle, colorbar in [m/s]. Axes show grid index.

Additionally, a quiver plot is shown in Figure 6.11.

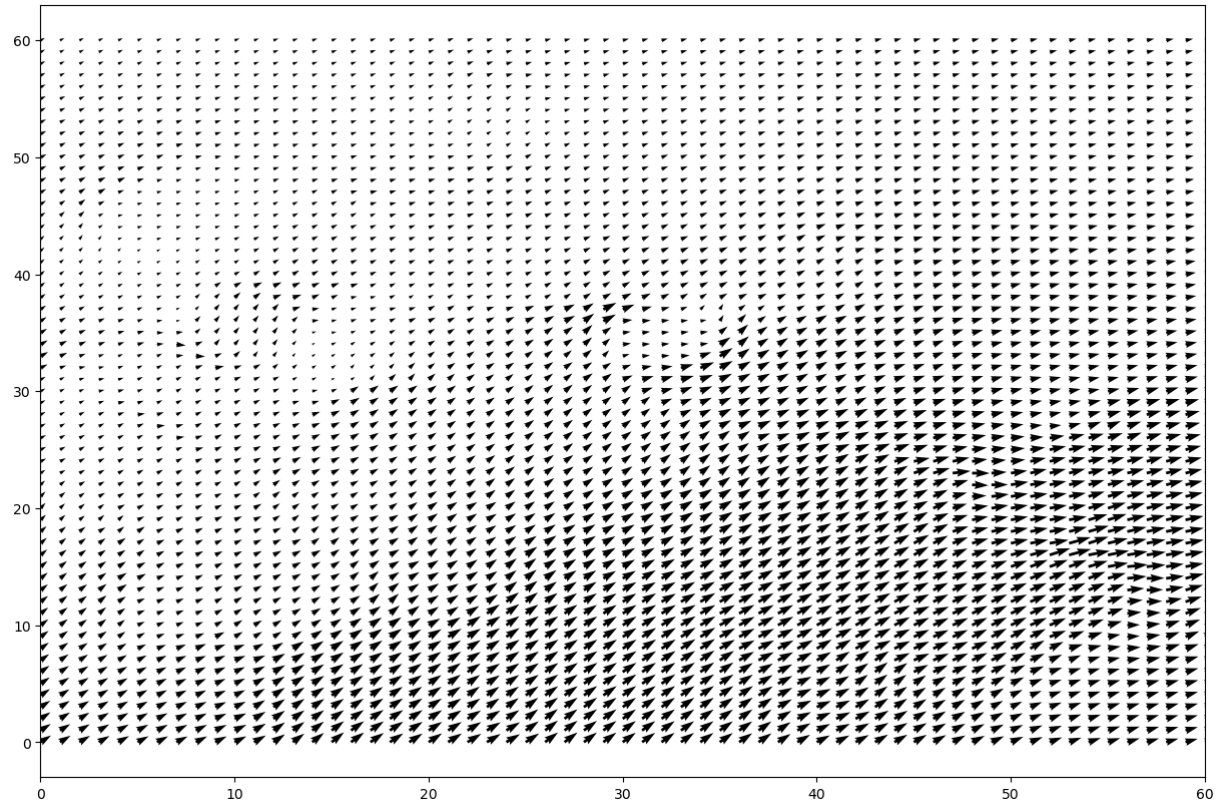


Figure 6.11: Quiver plot. Axes show grid index

6.3.3. One assimilation cycle

Lastly, a test is conducted with one assimilation cycle, i.e. 3600 seconds.

- $C_{mass-continuity} = 1000^3$
- $C_{smooth} = 1000^3$
- $C_{ac-background} = 0.01$
- $R = 30000[m]$
- $mask = 0[m]$
- $\alpha = 2$
- $C_{vorticity} = 1000^3$

MP related results are given in Table 6.7, PyDDA related results are given in Table 6.8.

Mode	MP	GFS	COMBINATION
Total RMSE	9.31 [m/s]	8.67 [m/s]	8.16 [m/s]

Table 6.7: Meteo-Particle related results

Mode	Initialization	PyDDA	No continuity	no observations
RMSE	4.98 [m/s]	5.19 [m/s]	4.94 [m/s]	8.63 [m/s]

Table 6.8: PyDDA related results

It is observed that the PyDDA runs have a lower RMSE compared to the other tests, raising the question if multiple assimilation windows leads to better performance. However, the estimates here are produced only after one hour of data collection, while for example a five minute scheme gives results every 5 minutes. The one hour scheme only gives better results because it can collect many data and therefore does suffer less from sparsity. Yet, it obviously cannot produce estimates at a high rate. The validation data also stem from the same hour, for sake of clarity. Figures Figure 6.12,Figure 6.13 and Figure 6.14 show zonal speeds, meridional speeds and a quiver plot respectively. The altitude is 36,000 [ft].

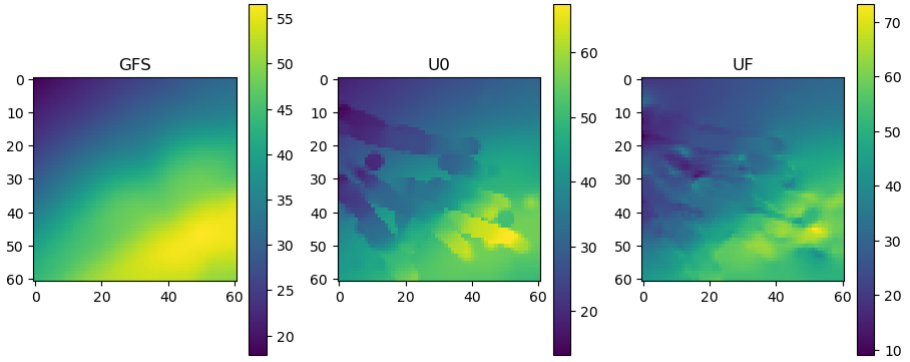


Figure 6.12: zonal speeds for last assimilation cycle, colorbar in [m/s]. Axes show grid index.

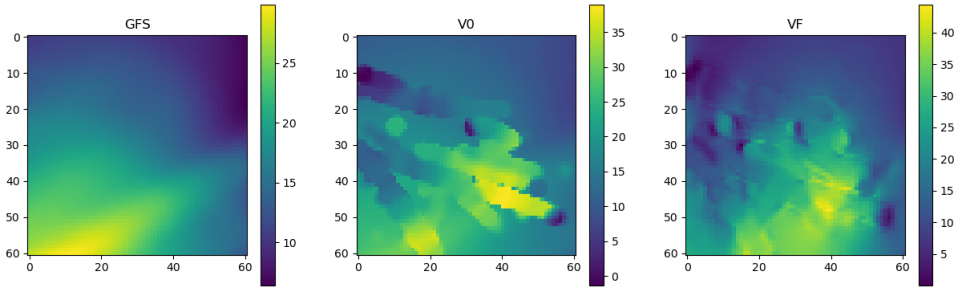


Figure 6.13: Meridional speeds for last assimilation cycle, colorbar in [m/s]. Axes show grid index.

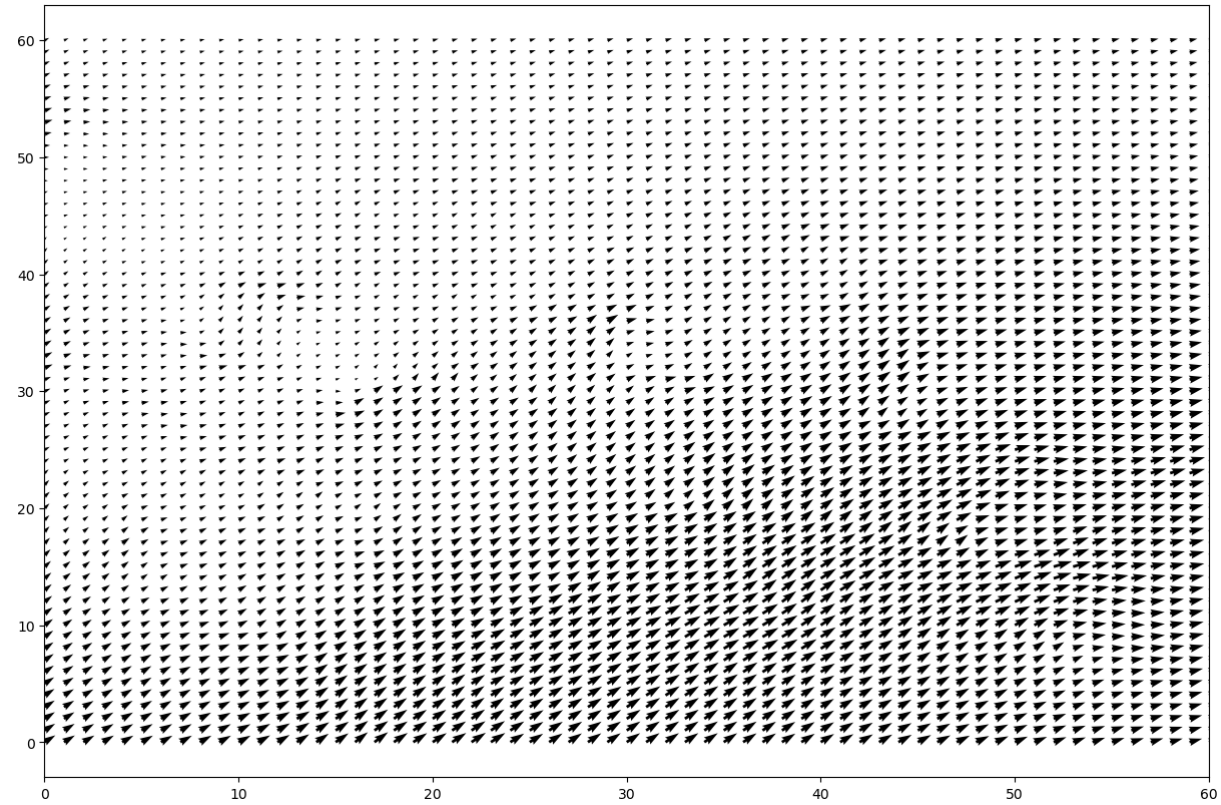


Figure 6.14: Quiver plot. Axes show grid index

6.4. Discussion

In this section, some additional material and thoughts are given on obtaining and interpreting results.

6.4.1. Overall performance of algorithms

The goal of the thesis research is to conclude whether or not wind estimation can be improved by incorporating physics knowledge with aircraft observation measurements. It is important to again mention that the aircraft data set was split into training aircraft, and validation aircraft. The previous tests illustrated that improvement over the MP model is possible when combining this model with GFS estimates outside of its confidence zone. However, this does not constitute a true incorporation of wind dynamics knowledge. The GFS forecast merely serves as a prior; it is the best guess available, in the absence of observations. Of course, it must be said that the GFS forecast was generated with lots of NWP and physics knowledge in mind, but the averaging between GFS and aircraft measurements in itself are not based on any physical process.

The PyDDA method tries to incorporate two dynamical equations, the first being continuity, the second being vorticity. However, it was concluded that those did not lead to any improvement over a baseline averaging of measurements and GFS. The exact reason behind this remains unknown, although some ideas are mentioned next.

- The optimizer (L-BGFS) cannot find a solution as the landscape is too complex; such reasoning is also given by Krishnapriyan et al.[11] for neural networks with physical losses.
- Many solutions are available to the analysis field; the optimizer cannot decide or draws an average; this phenomenon can occur in machine learning of physical models, as Thuerey et al. mention in case of neural networks [25].
- Observational data is too sparse: PyDDA was developed for Doppler measurements of storm

clouds, aircraft observations on the other hand are sparser.

- Weights are not tuned correctly, or are highly sensitive. Tuning the weights and finding the right approach (e.g. masking) is tedious, and the solutions might be highly sensitive to it.

6.4.2. Optimization effects

During the optimization, some abnormal effects were observed.

Oversmoothing

Oversmoothing of the wind field can occur when the smoothing weight is too high. Unrealistic fields are produced in that case, as can be seen in Figure 6.15. The field gets teared out in some spots, an effect which is observed to only get worse when the weight increases further.

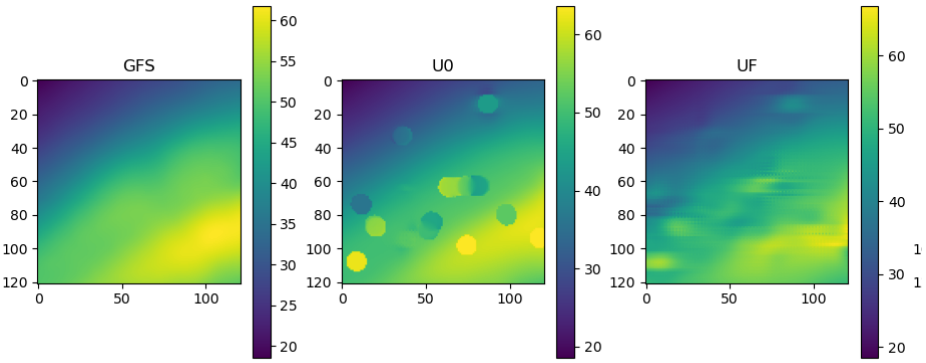


Figure 6.15: Oversmoothing effect on zonal wind speed, colorbar in [m/s]. Axes show grid index.

Checkerboard Pattern

PyDDA also suffers from checkerboard patterns in some layers, as seen in Figure 6.16. This can occur both under smoothing and non-smoothing. A valid reasoning has not been found yet.

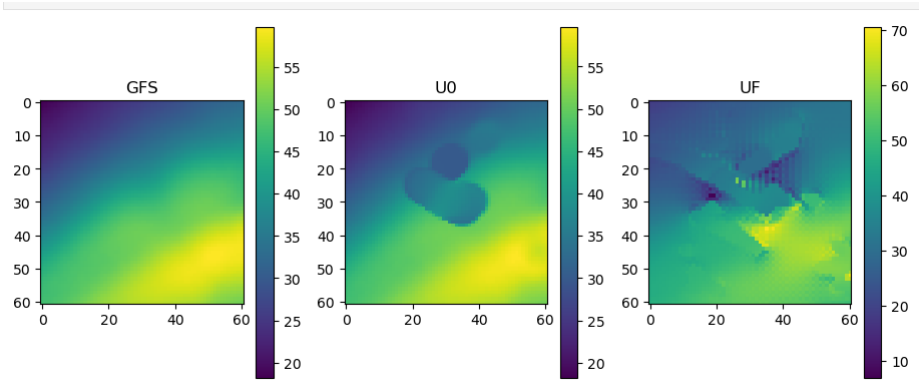


Figure 6.16: Checkerboard pattern, colorbar in [m/s]. Axes show grid index.

7

Subsequent Planning

In this chapter, an oversight of the thesis progression is given, including future work and a reflection. An updated Gantt chart was made, which can be consulted in appendix A.

7.1. Progress

The thesis starting date was 14/02/2022, the kick-off being on 18/02/2022. First, a literature study was done, in combination with the course on Research Methodologies. The research plan was completed and submitted in the beginning of April. The literature study continued and different methodologies were summarized. A small trade-off was performed, selecting two methodologies: the first being a combination of the MP model, GFS data and PyDDA. The second being a neural network architecture. This trade-off was completed at the end of April. Also in April, an entire week was spent on a lecture series on dynamical meteorology based on the book by Holton and Hakim [6] by Paul Ullrich. In hindsight, a lot of the material was out of scope and not of much use besides some chapters, included in the literature study. The main part of the literature study was concluded, after which data exploration started. This consisted of processing the GFS data. During the month of May, PyDDA was explored, as well as the MP model, and the first code was developed. In the beginning of June, 3-4 months after the thesis started, the first results were obtained based on MP-GFS combination, albeit limited. In June, the PyDDA framework was adapted and tested on the data. Small improvements were reported at first, but afterwards this was deemed untrue. After the summer holidays, work resumed and a preliminary presentation on the 23rd of August was postphoned to the 13th of September due to delays. PyDDA results are not improving based on physics losses.

7.2. Future Work and Other Methodologies

After the preliminary meeting, a more thorough analysis of aircraft data should follow, as some data could be faulty. A rejection mechanism could be constructed. Also testing different ensemble members of training aircraft could be used, and tuning could be improved. Also temperature estimation and short-term forecasting have not yet been looked into. Temperature estimation however remains a secondary objective for now; short-term forecasting only seems reasonable when nowcasting is successful.

However, as PyDDA seems unsatisfactory, another methodology may be tested in the form of neural networks which are physics-aware, modeled after the work of Schweri et al. [19]. More research into neural networks will be needed, as well as time to code. A paper and defence should be prepared as well. The neural network methodology should be completed in the months of October, November and December, with the green light meeting taking place in December too, and a defence in January.

7.2.1. Physics-informed Neural network

The methodology applied by Schweri et al. [19] is explained in chapter 2. The goal is to construct a neural network which has physics inspired loss functions, to predict a wind field based on some observations. The most difficult task will be to acquire training data. To train the parameters of the neural network in a supervised fashion, one needs a training set, itself split up in input points and training points. Also a validation set is required. The neural network has to produce a certain wind field based on the input points, and its weights must be trained by evaluating the training points. The points could be simply randomly split, or split per aircraft. Additionally, training sets could also be manufactured from GFS templates, as they provide a balanced state of the atmosphere. This would bypass the observation data.

7.2.2. Differentiable Physics

A solver-in-the-loop approach, as explained by Um et al. in [26], requires a numerical solver. This however is beyond the scope of the current assignment, and is therefore currently not considered anymore.

7.3. Reflection

The thesis assignment is challenging because of the meteorological nature; a non-trivial field to say the least. Both literature study, and coding are tedious processes, and real improvements have not occurred yet. The only improvement stems from simple averaging of measurements and forecast fields, which is not deemed a physics-inspired approach. However, a new method will be explored soon. If the results still do not improve, it should still constitute a scientific result.

8

Conclusion

The thesis work has the main objective of establishing whether it is possible to improve wind estimation by combining aircraft measurements with wind dynamics. This was coupled to a number of research questions and assumptions. Two main methodologies were developed, the first consisting of simply combining an existing nowcasting model (the Meteo-Particle model) with available GFS forecasts. The second method relates to a 3DVAR scheme which assimilates measurements into an analysis wind field, based on gradients of a number of cost functions.

Concerning the first method, improvements in the form of RMSE decreases were seen when comparing a combination of GFS and MP versus having GFS or MP as standalone models. The MP model offers strong local performance, while the GFS field fills in the gaps which are more distant from observational data. The MP model underperforms in those faraway regions, as it solely bases itself on observations which are far away from the validation point. However, this barely constitutes the inclusion of wind dynamics knowledge; only a GFS forecast is added in this approach.

The 3DVAR method, PyDDA, aimed to take it a step further with the inclusion of physics-inspired cost functions. This is a continuity loss, and a vorticity loss. It was found that this approach does not improve over results reached with ablation of the continuity/vorticity functions. It only improves over GFS or MP by being even better at averaging GFS and measurements. Also the produced analysis fiels look to be physically unrealizable, thus not being a solid solution to the assimilation problem.

Multiple ideas were put forward as why this might be, but a solid answer remains absent for now. For the near future, more thorough filtering of aircraft measurements is proposed, in combination with using ensembles. Next to that, a neural network approach, inspired by physical losses, will be tested. From other research [5] by Stoffelen and De Haan, it looks difficult however to gain any substantial improvement. They report a 5% improvement when bringing 3h assimilation cycles down to 1h. Nevertheless, the thesis research should continue with the proposed recommendations, and results should be viewed as a scientific answer to the research questions.

References

[1] Z. Bai et al. “Dynamic Mode Decomposition for Compressive System Identification”. In: *AIAA Journal* 58.2 (2020), pp. 561–574. DOI: 10.2514/1.J057870.

[2] D. M. Barker et al. *A Three-Dimensional Variational (3DVAR) Data Assimilation System For Use With MM5*. Tech. rep. No. NCAR/TN-453+STR. University Corporation for Atmospheric Research, 2003. DOI: http://dx.doi.org/10.5065/D6CF9N1J.

[3] D. M. Barker et al. “A Three-Dimensional Variational Data Assimilation System for MM5: Implementation and Initial Results”. In: *Monthly Weather Review* 132.4 (2004), pp. 897–914. DOI: 10.1175/1520-0493(2004)132<0897:ATVDAS>2.0.CO;2.

[4] Y. Cao et al. “A reduced-order approach to four-dimensional variational data assimilation using proper orthogonal decomposition”. In: *International Journal for Numerical Methods in Fluids* 53.10 (2007), pp. 1571–1583. DOI: https://doi.org/10.1002/fld.1365.

[5] S. de Haan and A. Stoffelen. “Assimilation of High-Resolution Mode-S Wind and Temperature Observations in a Regional NWP Model for Nowcasting Applications”. In: *Weather and Forecasting* 27.4 (2012), pp. 918–937. DOI: 10.1175/WAF-D-11-00088.1.

[6] J. R. Holton and G. J. Hakim. In: *An introduction to dynamic meteorology*. 4th ed. Vol. 88. International Geophysics Series. Elsevier Academic Press, 2004.

[7] Gao Huang et al. *Densely Connected Convolutional Networks*. 2016. DOI: 10.48550/ARXIV.1608.06993. URL: https://arxiv.org/abs/1608.06993.

[8] R. Jackson et al. *PyDDA: A Pythonic Direct Data Assimilation framework for wind retrievals*. Version 0.5.2. July 2020. DOI: 10.5281/zenodo.3942686. URL: https://doi.org/10.5281/zenodo.3942686.

[9] R. Kikuchi et al. “Nowcasting algorithm for wind fields using ensemble forecasting and aircraft flight data”. In: *Meteorological Applications* 25.3 (2018), pp. 365–375. DOI: https://doi.org/10.1002/met.1704.

[10] R. Klein. “Scale-Dependent Models for Atmospheric Flows”. In: *Annual Review of Fluid Mechanics* 42.1 (2010), pp. 249–274. DOI: 10.1146/annurev-fluid-121108-145537.

[11] A. S. Krishnapriyan et al. “Characterizing possible failure modes in physics-informed neural networks”. In: (2021). DOI: 10.48550/ARXIV.2109.01050.

[12] A. M. P. de Leege, M. M. van Paassen, and M. Mulder. “Using Automatic Dependent Surveillance-Broadcast for Meteorological Monitoring”. In: *Journal of Aircraft* 50.1 (2013), pp. 249–261. DOI: 10.2514/1.C031901. eprint: https://doi.org/10.2514/1.C031901. URL: https://doi.org/10.2514/1.C031901.

[13] V. Mazzarella et al. “Comparison between 3D-Var and 4D-Var data assimilation methods for the simulation of a heavy rainfall case in central Italy”. In: *Advances in Science and Research* 14 (Aug. 2017), pp. 271–278. DOI: 10.5194/asr-14-271-2017.

[14] National Centers for Environmental Prediction, National Weather Service, NOAA, U.S. Department of Commerce. *NCEP GFS 0.25 Degree Global Forecast Grids Historical Archive*. Boulder CO, 2015. URL: https://doi.org/10.5065/D65D8PWK.

[15] J. Nocedal and S. Wright. *Numerical Optimization*. Second edition. New York: Springer, 2006.

[16] C. Potvin, A. Shapiro, and J. Gao. “Use of a Vertical Vorticity Equation in Variational Dual-Doppler Wind Analysis”. In: *Journal of Atmospheric and Oceanic Technology* 26.10 (2009), pp. 2089–2106. DOI: 10.1175/2009JTECHA1256.1. URL: https://journals.ametsoc.org/view/journals/atot/26/10/2009jtecha1256_1.xml.

[17] C. Potvin, A. Shapiro, and M. Xue. “Impact of a Vertical Vorticity Constraint in Variational Dual-Doppler Wind Analysis: Tests with Real and Simulated Supercell Data”. In: *Journal of Atmospheric and Oceanic Technology* 29.1 (2012), pp. 32–49. DOI: 10.1175/JTECH-D-11-00019.1. URL: https://journals.ametsoc.org/view/journals/atot/29/1/jtech-d-11-00019_1.xml.

[18] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. DOI: 10.48550/ARXIV.1505.04597. URL: https://arxiv.org/abs/1505.04597.

[19] L. Schweri et al. “A Physics-Aware Neural Network Approach for Flow Data Reconstruction From Satellite Observations”. In: *Frontiers in Climate* 3 (2021). ISSN: 2624-9553. DOI: 10.3389/fclim.2021.656505.

[20] American Meteorological Society. *Glossary of Meteorology: Navier-Stokes Equations*. Last accessed on 06/04/2022. 2012. URL: https://glossary.ametsoc.org/wiki/Navier-stokes_equations.

[21] NASA Goddard Institute for Space Studies. *Panoply netCDF, HDF and GRIB Data Viewer*. Software package. 2022. URL: https://www.giss.nasa.gov/tools/panoply/.

[22] J. Sun et al. “pyModeS: Decoding Mode-S Surveillance Data for Open Air Transportation Research”. In: *IEEE Transactions on Intelligent Transportation Systems* (2019). ISSN: 1524-9050. DOI: 10.1109/TITS.2019.2914770.

[23] J. Sun et al. “Weather Field Reconstruction using aircraft surveillance data and a novel meteo-particle model”. In: *PLoS ONE* 13.10 (2018), e0205029. DOI: https://doi.org/10.1371/journal.pone.0205029.

[24] N. Thuerey et al. “Deep Learning Methods for Reynolds-Averaged Navier–Stokes Simulations of Airfoil Flows”. In: *AIAA Journal* 58.1 (2020), pp. 25–36. DOI: 10.2514/1.J058291.

[25] N. Thuerey et al. *Physics-based Deep Learning*. WWW, 2021. URL: https://physicsbaseddeeplearning.org.

[26] K. Um et al. *Solver-in-the-Loop: Learning from Differentiable Physics to Interact with Iterative PDE-Solvers*. 2020. DOI: 10.48550/ARXIV.2007.00016.