Towards the standardisation of obstacle-resolving model results

DISSERTATION

zur Erlangung des Doktorgrades der Naturwissenschaften

an der Fakultät für Mathematik, Informatik und Naturwissenschaften

Fachbereich Geowissenschaften der Universität Hamburg

vorgelegt von

Vivien Voss

aus Hamburg

Hamburg, 2025

Fachbereich Erdsystemwissenschaften

Datum der Disputation:	16.01.2026
Gutachter/innen der Dissertation:	Prof. Dr. Bernd Leitl
	Prof. Dr. K. Heinke Schlünzen
Zusammensetzung der Prüfungskommission:	Prof. Dr. Bernd Leitl
	Prof. Dr. K. Heinke Schlünzen
	Prof. Dr. Jürgen Böhner
	Dr. Birgit Gaye
	Dr. Angelika Heil
Vorsitzender des Fach-Promotionsausschusses Erdsystemwissenschaften:	Prof. Dr. Hermann Held
Dekan der Fakultät MIN:	Prof. Dr. Ing. Norbert Ritter

Zusammenfassung

Um eine gute Nachnutzbarkeit von veröffentlichten Forschungsdaten sicherzustellen, können Datenstandards verwendet werden. Diese legen fest, wie Datenproduzenten einheitliche Daten generieren können und damit die Nachvollziehbarkeit für den Nachnutzer erhöhen können. In der Atmosphärenforschung haben sich zur Beschreibung und Bereitstellung von Forschungsdaten das netCDF-Datenformat und die Climate and Forecast (CF)-Konventionen etabliert. Die CF-Konventionen bieten vielen Disziplinen bereits gute Möglichkeiten, ihre Daten zu beschreiben und viele numerische Modell- und Beobachtungsdaten werden in diesem Format publiziert. Aber nicht alle Nischen und ihre Besonderheiten sind durch die CF-Konventionen ausreichend beschreibbar und abgedeckt. In der Stadtklimaforschung beispielsweise, in der mit hindernisauflösenden mikroskaligen Modellen (engl. Obstacle Revolving Model – ORM) gearbeitet wird, fehlt bisher ein ORM-übergreifender und spezifischer Datenstandard. Während numerische Modelle am ehesten die netCDF und die CF-Konventionen verwenden, auch wenn wichtige Datenbeschreibungen noch nicht standardisiert sind, gibt es in der physikalischen Modellierung keinen modellübergreifenden Datenstandard. Dabei würden standardisierte Daten gerade die in der Stadtklimatologie verbreiteten Datenvergleiche zwischen numerischen Modellen, physikalischen Modellen und Beobachtungsdaten erleichtern. Um die Veröffentlichung gut beschriebener Forschungsdaten zu erhöhen, wurde der auf dem netCDF-CF-Standard basierende AtMoDat-Standard entwickelt, der Datenproduzierenden dabei helfen soll, ihre Daten besser zu beschreiben. Diese Dissertation befasst sich mit dem Prozess zur Standardisierung hindernisauflösender Modelldaten. Ziel ist es zu ermittelt, welche Ansprüche an die Standardisierung gestellt werden müssen und welche Schwierigkeiten in der Standardisierung auftreten und gelöst werden müssen. Dazu wird der AtMoDat Standard auf Datensätze des numerischen Modell MITRAS und des Windkanallabors der Universität Hamburg angewendet.

Die erste Studie dieser Arbeit, befasst sich mit der Frage, wie Daten verschiedener numerischer Modelle generiert und nachgenutzt werden und welche Ansprüche Datenproduzierende und -nutzende an einen ORM-Datenstandard haben. Mittels einer Umfrage wurden Modellierer und Datennutzende gezielt zu den von ihnen genutzten Modellen, deren Eigenschaften und Bereitstellung bzw. Verwendung von Modelldaten befragt. Daraus wurde abgeleitet, inwieweit eine Standardisierung bereits vorhanden ist und welche Anforderungen ein ORM-Standard erfüllen sollte. Die Umfrage hat ergeben, dass sich die Teilnehmenden trotz der verschiedenen Modelle der Notwendigkeit von Datenstandards bewusst sind. Sie wünschen sich trotz der verschiedenen Modellausgabeeinstellungen und -verarbeitungen die Nutzung des netCDF-Formats mit den CF-Konventionen.

Anschließend wurde die Anwendbarkeit des AtMoDat-Standard auf numerische ORM-Daten, in dieser Arbeit auf Daten des Modells MITRAS, untersucht. Zwar produziert MITRAS bereits netCDF-Daten, es fehlten jedoch einige Aspekte, die zur Erfüllung bzw. Einhaltung der Vorgaben der CF-Konventionen notwendig sind. Dabei zeigte sich, dass die Erfüllung der CF-Konventionen auf ORMs technisch machbar

ist, für einen Datenproduzenten jedoch nicht immer trivial ist. Um den Anforderungen des AtMoDat-Standards gerecht zu werden, wurde das Post-Processing des Modells um einige Aspekte ergänzt. Der modelleigene Post-Prozessor wurde erweitert und an die CF-Konventionen in Bezug auf die Datenbeschreibung angepasst. Fehlende Datenbeschreibungen wurden ergänzt und die Datenstrukturen wurden angepasst, um wichtige Informationen besser sichtbar und für Nachnutzende besser nachvollziehbar zu machen. Darüber hinaus wurde der Postprozessor nc2atmodat entwickelt, der die MITRAS-netCDF-Modellergebnisse für die Datenpublikation aufbereitet und finalisiert. nc2atmodat beinhaltet zusätzliche notwendige Anpassungen, die im modelleigenen Postprocessing nicht trivial zu lösen sind und ermöglicht so eine AtMoDat-konforme Bereitstellung von Daten. Die ersten MITRAS-Datensätze, die die Anforderungen des AtMoDat-Standards erfüllen, wurden bereits publiziert. Bei der Anpassung der Modellroutinen an die CF-Konventionen zeigte sich, dass viele Aspekte, die in ORMs geläufig sind, im CF-Standard oft nicht vorkommen oder abgedeckt sind. Dazu zählen ORM-spezifische Variablen, die in anderen ORMs leicht anders definiert sind, oder die explizite Auflösung der Gebäude im Modellgebiet, mit der sich Oberflächenvariablen abbilden lassen, die noch keine explizite Behandlung in den CF-Konventionen erfahren haben.

In der physikalischen Modellierung, die sich mit Stadtklimaprozessen befasst, besteht bisher kein übergreifender Datenstandard, der eine einheitliche Bereitstellung von Daten vorgibt. Üblich ist es, dass je nach Arbeitsgruppe oder Datenproduzent, unterschiedliche Layouts, Datenbeschreibungen oder Datenmengen verwendet werden. So werden neben den Modellmessergebnissen, die als dimensionslose Werte in ASCII- oder Exceldateien herausgegeben werden, noch Informationen zur Umrechnung, Simulationseigenschaften und Set-ups mitgeliefert. Dadurch sind diese Daten im Umfang und Handhabung komplex. Bei dem Standardisierungsversuch wurde daher ein sehr grundlegender Ansatz gewählt. Angelehnt an die Handhabung in der numerischen Modellierung und den CF-Konventionen wurde ein Konzept für die Bereitstellung von Windkanaldaten im netCDF-Format entwickelt und angewendet. Dabei zeigte sich, dass die Bereitstellung und Beschreibung von Windkanaldaten als netCDF-Daten grundsätzlich möglich zu sein scheint. Bei der Bereitstellung von relevanten Informationen zur Beschreibung der Daten sind vielfältige Möglichkeiten vorhanden. Gewisse Aspekte der CF-Konventionen lassen sich teils direkt anwenden, teils müssen vorhandene Konzepte jedoch auf den Windkanal angepasst werden, sodass sie dem aktuellen CF-Standard nicht mehr entsprechen würden. Diese Arbeit bietet daher nur erste Schritte oder Ideen zu einem möglichen Windkanalstandard, der innerhalb der Community weiter definiert, diskutiert und getestet werden muss.

Abstract

Data standards can be used to ensure that published research data can be easily reused. These standards define how data producers can generate uniform data and thereby increase the reuse for users. In atmospheric research, the netCDF data format and the Climate and Forecast (CF) conventions are commonly used to describe and provide research data. The CF conventions already offer for many disciplines good opportunities to describe their data, since numerical model and observation data are often published in this format. However, not all niches and their specific features can be adequately described and covered by the CF conventions. In urban climate research, for example, where obstacle-resolving microscale models (ORM) are used, there is currently no ORM-wide and specific data standard. While numerical models mostly use the netCDF and CF conventions, even if certain data descriptions are not yet standardised, there is no overall data standard for physical model data. Standardised data would facilitate the data comparisons between numerical models, physical models and observational data that are common in urban climatology. In order to increase the publication of well-described research data, the AtMoDat standard, which is based on the netCDF-CF standard, was developed to help data producers better describe their data. This dissertation deals with the process of standardising obstacle-resolving model data. The aim is to determine what requirements and what difficulties arise in standardisation and need to be resolved. To this end, the AtMoDat standard is applied to datasets from the numerical model MITRAS and the wind tunnel laboratory at the University of Hamburg.

The first study in this thesis deals with the question of how data from different numerical models is generated and reused, and what requirements data producers and users have for an ORM data standard. A survey was conducted to ask modellers and data users specific questions about the models they use, their properties, and the provision and use of model data. The aim is to determine the extent to which standardisation already exists and what requirements an ORM standard should meet. The survey revealed that the participants are aware of the need for data standards, and despite the different model output settings and processing methods, they would like to see the use of the netCDF format with CF conventions.

In the second part, the applicability of the AtMoDat standard to numerical ORM data, in this work to data from the MITRAS model, was investigated. Although MITRAS already produces netCDF data, some aspects necessary to comply with the requirements of the CF conventions were missing. It was found that compliance with the CF conventions on ORMs is technically feasible, but not always trivial for a data producer. In order to meet the requirements of the AtMoDat standard, several aspects were added to the post-processing of the model. The model's own post-processor was expanded and adapted to the CF conventions with regard to data description. Missing data descriptions were added, and the data structures were adapted to make important information more visible and easier to understand for subsequent users. In addition, the post-processor nc2atmodat was developed, which prepares

and finalises the MITRAS netCDF model results for publication. Furthermore, nc2atmodat includes additional necessary adjustments that are not trivial to solve in the model's own post-processing, thus enabling AtMoDat-compliant data provision. The first MITRAS datasets that meet the requirements of the AtMoDat standard have already been published. When adapting the model routines to CF conventions, it became apparent that many aspects that are common in ORMs are often not covered in the CF standard. This includes ORM-specific variables that are defined slightly differently in other ORMs, or the explicit resolution of buildings in the model area, which can be used to map surface variables that have not yet been explicitly addressed in the CF conventions.

In physical modelling dealing with urban atmospheric processes, there is currently no data standard available that ensures a uniform provision of wind tunnel model data. Furthermore, it is common practice for different layouts, data descriptions or datasets to be used depending on the working group or data producer. In addition to the model measurement results, which are published as dimensionless values in text or Excel files, information on conversion, simulation properties and set-ups is also provided. This makes the data complex in terms of scope and handling. A very basic approach was therefore chosen for the standardisation attempt. Based on the handling in numerical modelling and the CF conventions, a concept for providing wind tunnel data in netCDF format was developed and applied. This showed that the provision and description of wind tunnel data as netCDF data appears to be generally possible. But there are many options available to provide relevant information for the data descriptions. Certain aspects of the CF conventions can be applied directly. However, some existing concepts need to be adapted to fit the wind tunnel, with the possibility that they would no longer comply with the current CF standard. This work therefore only offers initial steps or ideas for a possible wind tunnel standard, which needs to be further defined, discussed and tested within the community.

List of publications

This is an overview of the published work which I was involved in, contributed to, or was based on this thesis. The list contains scientific articles, datasets and source code.

1. Ganske, A., Kraft, A., Kaiser, A., Heydebreck, D., Lammert, A., Höck, H., Thiemann, H., **Voss, V.**, Grawe, D., Leitl, B., Schlünzen, K. H., Kretzschmar, J., and Quaas, J. (2021). ATMODAT Standard (v3.0). In World Data Center for Climate (WDCC) at DKRZ. https://doi.org/https://doi.org/10.35095/WDCC/atmodat_standard_en_v3_0

2. **Voss, V.**, Schlünzen, K. H., & Grawe, D. (2021). AtMoDat (Atmospheric Model Data) — Creation of a Model Standard for Obstacle Resolving Models. Springer Proceedings in Complexity. https://doi.org/10.1007/978-3-662-63760-9_48

3. **Voss V.** (2023). Results of a Survey on Obstacle Resolving Model Data (Version 1) [Dataset]. Zenodo. https://doi.org/10.5281/zenodo.8211274

4. **Voss, V.** (2023). High resolution obstacle-resolved model results for the city centre of Hamburg, Germany, from the microscale transport and stream model MITRAS on 21 June 2000. [Dataset]. World Data Center for Climate (WDCC) at DKRZ. https://doi.org/10.26050/WDCC/MITRAS_ATMODAT

5. Cheng, G., Schlünzen, K. H., Grawe, D., **Voss, V.**, Thatcher, M., & Rayner, P. (2023). Parameterizing building effects on airflows within the urban canopy layer for high-resolution models using a nudging approach. Quarterly Journal of the Royal Meteorological Society, 149(755), 2617–2633. https://doi.org/10.1002/qj.4524

6. **Voss, V.**, Schlünzen, K. H., & Grawe, D. (2024). Survey On Output Data From Obstacle Resolving Atmospheric Models. Meteorologische Zeitschrift. https://doi.org/10.1127/metz/2024/1217

7. Ferner, K. S., Boettcher, M., Sieck, K. & **Voss, V.** (2024). Sensitivity studies on rain heterogeneity in Hamburg city centre using the obstacle-resolving model MITRAS. [Dataset]. World Data Center for Climate (WDCC) at DKRZ. https://doi.org/10.26050/WDCC/MitrasWinterHamburg

8. **Voss. V.** (2025). AtMoDat/nc2atmodat: v1.0.2 (v1.0.2) [Code]. Zenodo. https://doi.org/10.5281/zenodo.17035812

9. **Voss, V.** (2025). wt2atmodat. [Code]. Zenodo. https://doi.org/10.5281/zenodo.17176544

Contents

1 Introduction

1.1 Data Standards

Scientific research involves the generation, processing, analysis and evaluation of research data to find answers to different scientific questions. Since unusable data becomes less valuable (Korsemeyer and Thompson, 2005; Bermudez, 2017), data should be published, reusable and long-term available for users, and this must be supported by the storage system and used data formats (Eggleton and Winfield, 2020; Dickinson et al., 2002). Data storage systems and data repositories e.g. Zenodo[1] or the World Data Center for Climate (WDCC)[2], provide infrastructure for such a long-term archive to store research data and make it accessible for users.

Any type of research data (results, processed or raw data, processing scripts or code, grey literature and documentation) can be published. Nevertheless, publishing research data should adhere to certain rules. Published data needs to be prepared in a way that enables a user to work with the data. Ideally, it should follow the FAIR principles, meaning that data must be findable, accessible, interoperable and reusable (Wilkinson et al., 2016). To ensure data are identifiable and citable, datasets are assigned a persistent identifier, such as a DOI.

In atmospheric science, data can be categorised into model data and measurement data. Ideally, model data should be provided in a structured way that allows users to easily understand and access the data. This can be achieved through the selection of specific data formats, the use of particular vocabulary and processing tools. Data standards ensure a consistent preparation structure for datasets, enabling consistent generation, access, processing and understanding of the data.

The most used and well-known data standard are the Climate and Forecast (CF) conventions (Eaton et al., 2024). The CF conventions are guidelines which describe how to provide information for the sufficient description of atmospheric numerical model data. The CF conventions are mainly designed to be applied to netCDF dataset. The *network common data format* (netCDF) is a self-describing data format that stores data in an array-like, hierarchical and multi-dimensional structure within one dataset (Rew and Davis, 1990; Rew et al., 2011). NetCDF is commonly used for numerical model data, since one file can store several variables with temporal and spatial dependencies. Figure 1.1 shows an example of the data structure represented in the netCDF data format. The netCDF format is well-established in the atmospheric modelling community and is used as well for oceanic, surface model data and can be applied to observational measurement data (Eaton et al., 2024).

The emergence of so-called model intercomparison projects (MIPs) has reinforced the need for data standards. A MIP consists of a group of models that investigate certain scientific questions by simulating the same properties. The models and their results are then compared against each other to evaluate their performance. This should ensure that the models behave similarly and provide more reliable and robust insights and findings for a given research question (Gates et al., 1999; Eyring et al., 2016). The

[1] https://zenodo.org/
[2] https://www.wdc-climate.de/ui/

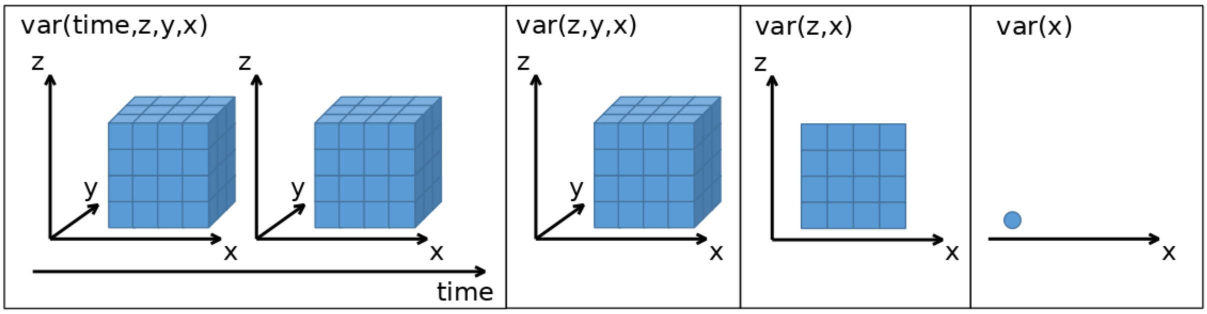

Figure 1.1: Description on the structure of variables within the netCDF datasets (Figure based on Rew and Davis (1990) and Hoyer and Hamman (2017)). Various variables and data of different dimensions can be stored in the same data file. The data values can be accessed by selecting the appropriate index.

most prominent MIP is the Coupled Model Intercomparison Project (CMIP), which is used to study the effects of climate change and plays a key role in the work of the Intergovernmental Panel on Climate Change (IPCC) (Eyring, 2016; Touzé-Pfeiffer, 2020). Further MIPs are e.g. the Atmospheric Model Intercomparison Project (AMIP) (Gates et al., 1999), the AEROCOM Project[3] (Gliß et al., 2021) and HighResMIP (Haarsma et al., 2016). With the appearance of such MIPs, the need for data standards in the climate science community has arisen with the aim to foster the intercomparison, the access, exchange and processing of data across different models (Dickinson et al., 2002).

Since the CF conventions are widely used in many research disciplines, many project-, model- or discipline-specific data standards utilise their structure and concepts, for instance, the SAMD standard for observational measurement data (Lammert et al., 2017); the model-specific data standard for the numerical model PALM-4U (Scherer et al., 2019; Maronga et al., 2019) and the netCDF-CF-OPENDAP for oceanic data (Hankin et al., 2010). Furthermore, tools and code libraries for modifying netCDF files to conform to the CF conventions have emerged (e.g. Hassell et al. (2017) or Jones et al. (2023)). Another standard, incorporating netCDF and the CF conventions, is the Atmospheric model data (AtMoDat) standard, which serves as a guideline for the publication of well-described and standardised model data, targeting non-standardised disciplines or smaller MIPs (Ganske et al., 2020, 2021). Such a non-standardised atmospheric research disciplines that produces model data is the obstacle-resolving urban climate modelling community. Within the data publication process, three main target groups are involved (Figure 1.2). A Data producer runs experiments, selects and prepares the data intended for publication and provides the required information and descriptions. A data curator of a repository checks the information and data for completeness and adds metadata for the publication. A data user is then able to access and process the data. This work will focuses on the tasks of a data producer.

[3]https://aerocom.met.no/

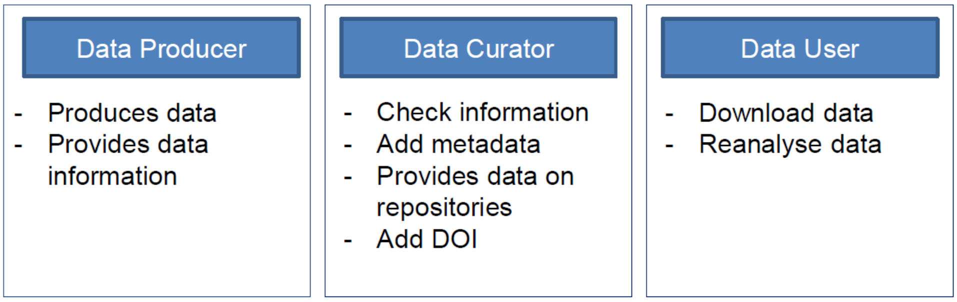

Figure 1.2: Target user groups, that are involved in the data publication process and their core functions.

1.2 Obstacle resolving modelling

According to the UN, in 2018, 55 % of the world population lived in urban areas, and they estimate that this trend will continue in the future (UN, 2018). With the increasing impact of climate change, administrations and urban developers need to ensure the well-being of the human population in densely populated areas. Artificial surfaces and obstacles, such as buildings or trees, alter the roughness length and cause turbulent atmospheric flow. This has an impact on the atmospheric conditions within urban areas, influencing the well-being and living conditions of humans, animals and vegetation. Urban atmospheric research investigates these atmospheric processes in urban areas and their effects on air quality, ventilation, wind comfort, thermal comfort, effects of vegetation, radiation and urban planning (WMO, 2023; Zhao et al., 2023; Oke et al., 2017).

To investigate and monitor the atmosphere and their processes, observational measurement networks (Oke et al., 2017; Baklanov et al., 2018), often operated by authorities or researchers, are installed. Due to the heterogeneity of urban fabric, depending on the layout and positioning of obstacles (buildings, vegetation) or anthropogenic influences, the representativeness of observational data is limited. Observational data are bound to the prevailing conditions of the atmosphere; long-term measurements have to be conducted continuously to cover those changes in atmospheric conditions. Furthermore, the running costs for such networks are high due to the required maintenance, which leads to rather sparse measuring sites operated by authorities. Nevertheless, observational data are used as validation data or boundary conditions in the model initialisation (Vieira Zezzo et al., 2023).

Another method is the use of obstacle-resolving micro-scale atmospheric models (ORMs), which can be physical models, like wind tunnels or computational numerical models. Both types replicate the real-world conditions in the atmospheric canopy layer in their respective model regime and simulate the atmospheric flow and processes over a simplified version of the study area. Model simulations are used to close the gaps within sparse measurement networks, simulate certain atmospheric conditions or the effects of new buildings on the surroundings. Nevertheless, models are limited to the capabilities, resolutions and boundaries of the respective model. To study and understand urban atmospheric processes, but also to evaluate the performance of models, data comparisons between observational measurements data, numerical and physical model data are common. These data comparisons are often used within model development to validate the performance of a model or implemented parameterisations.

1.3 Research questions

Unlike in the climate or weather forecast modelling community, no discipline-specific data standard for ORMs is available. While many numerical ORMs already provide their data partially as netCDF files, not all models incorporate the CF conventions in their data description. Furthermore, the CF convention often does not cover every aspect that typically occurs in this discipline. Typical ORM variables, e.g. variables related to building surfaces, e.g. wall temperatures, etc., are currently not standardised. In physical modelling, there is no established general data standardisation. Instead, each wind tunnel model simulation and dataset are generated and described by the data producer individually, as each simulation focuses on different aspects in the production. This hinders the production of well-described, standardised and FAIR datasets and impairs the use and handling of model data in this community. Without such standardised datasets, a potential data user will have trouble understanding or processing the data. To achieve a standardisation for ORMs, the following research questions arise:

RQ1: How do obstacle-resolving numerical models store their results so far and what do users want?

RQ2: How can existing standards be applied to obstacle-resolving numerical model data?

RQ3: How can a possible standardisation derived for obstacle resolving numerical model data be transferred to wind tunnel model data?

This work aims to explore and pave the path for the standardisation for obstacle-resolving microscale model data. The application and implementation of workflows for the standardisation of model results after the requirements of the AtMoDat Standard will be applied to datasets from the numerical Model MITRAS (Schlünzen et al., 2003; Salim et al., 2018) and wind tunnel model data from the Environmental Wind tunnel facility at the University of Hamburg. It should provide ideas on how model data must be provided to become well-described and FAIR data publications, and demonstrate the work and effort of a data producer to fulfil these steps.

1.4 Structure of the thesis

This work is divided into four main chapters. Chapter 2 provides an overview on obstacle resolving models. It explaines differences in the model types and challenges that occur in this discipline.
Chapter 3 describes the approach used to collect information from the ORM community about different properties of numerical models through an online survey. The survey mainly focuses on the experiences of different numerical model users and should provide an overview of different models used, the needs and wishes from the model user, to collect information and experiences of the users on different ORMs.

In Chapter 4, the application of the AtMoDat data standards onto the model output of MITRAS is described. It contains the steps needed to adapt the post-processing routine m2cdf for the translation of MITRAS model results into netCDF datasets to become well-described model datasets. Furthermore, a new post-processing routine called nc2atmodat was developed to adapt MITRAS model results towards the CF conventions and the AtMoDat Standard, whereas the AtMoDat Data checker was applied to MITRAS model data.

Chapter 5 deals with the application of a standardisation for wind tunnel data, by converting wind tunnel model data into the netCDF format. Examples and methods for the description of wind tunnel model data are derived from the CF conventions.

Chapter 6 summarises the work from the previous Chapters, discussing the methods used and the results and summarises the answers to the research questions. It also outlines further challenges, adaptations, and potential directions for future work.

2 Comparisons between numerical and physical models

Generally, a model is a tool that can be used to study atmospheric processes by representing the real-world conditions in a simplified way. Models are capable of maintaining constant boundary conditions, which allow constant atmospheric conditions to study certain atmospheric processes that only happen naturally occasionally or under certain circumstances. Obstacle-resolving models use grid sizes of a few meters and below and time steps of fractions of seconds, which are suited to resolve small-scale atmospheric processes. On these scales, obstacles and their effect on the atmospheric flow or turbulent fluxes are explicitly resolved in the model simulations. Processes, which cannot be directly resolved, are simplified, parametrised or neglected. Therefore, these models are used to simulate the flow over complex terrain, like urban areas, city quarters, street canyons, or single buildings. Obstacles are treated as solid structures, e.g. buildings (Ali et al., 2023; Song et al., 2018; Bohnenstengel et al., 2004) (or ships (Badeke et al., 2021; Janssen et al., 2017)) or permeable structures such as trees (Salim et al., 2015; Gromke and Ruck, 2007; Balczó et al., 2009). ORMs are classified as numerical or physical models.

2.1 Numerical models

A numerical model is a computational model that solves the governing physical equations for each grid cell and integrates those into the future (Oke et al., 2017). The capabilities of a numerical model are limited by computational resources, determining the speed and accuracy of the model simulation and the implemented processes and parameterisations. The required computational resources depend on domain size, model time step, grid size and modelled wind velocities. This dependency can be described by the Courant-Friedrich-Lewy (CFL) criterion (WMO, 2023), which describe the relation of the wind speed u in m s^{-1}, time step Δt in s and grid size Δx in m in the horizontal direction (Eq. 2.1) and determines the stability of the model simulation.

$$C = \frac{u \cdot \Delta t}{\Delta x} \leq 1 \tag{2.1}$$

Model simulations with high wind speeds are only stable with a small time step or with a large grid size, so that the ratio is below 1. If the ratio exceeds 1, the model becomes unstable, and the model results are unreliable. Flow turbulences cannot be solved analytically and need to be parametrised. To resolve the turbulent flow, three different filtering methods are available: Reynolds-averaged-Navier Stokes (RANS), Large-Eddy-Simulation (LES) and direct numerical simulation (DNS) (WMO, 2023). A DNS model resolves the whole turbulent flow and resolves all occurring turbulence directly, providing the most accurate results, but it has the highest demand on computational resources. It requires a fine grid and small time steps to resolve the whole turbulent spectrum (WMO, 2023). A RANS model provides a time-averaged turbulent flow, which has a lesser demand on computational resources. The time and spatial resolution can be coarser, because the eddies are not resolved, but it affects the ability to resolve the smallest turbulence (WMO, 2023). Mostly, such turbulences need to be parametrised. An LES model is located in the spectrum between DNS and RANS. It resolves larger or coarser turbulent structures but requires a subgrid-scheme model for smaller vortices.

In this work, the model results from the ORM MITRAS, short for microscale transport and stream model, are used. MITRAS is a non-hydrostatic, obstacle-resolving RANS model (Schlünzen et al., 2003; Salim et al., 2018; Schlünzen et al., 2018a,b), obstacles are explicitly resolved using the mask method (Briscolini and Santangelo, 1989; Salim et al., 2018), where building cells are treated as volume cells blocking the flow. The practical process to run a numerical model simulation requires three main steps (blue boxes in Figure 2.1): pre-processing, the model simulation itself and the post-processing. During

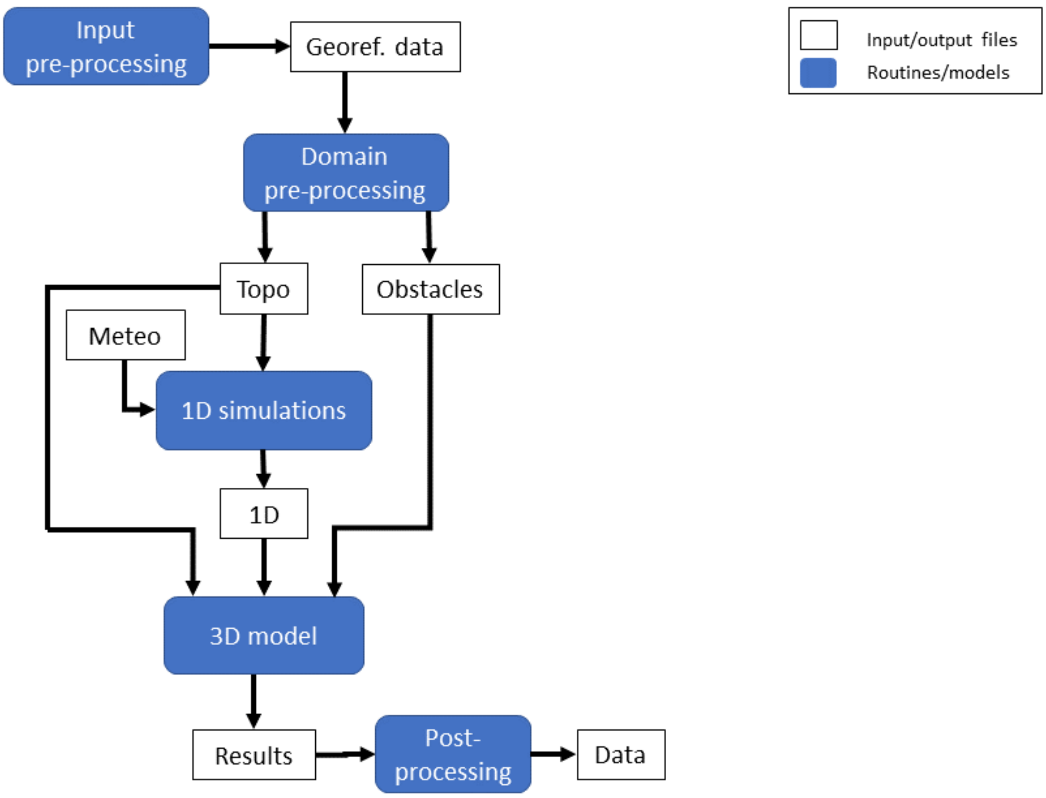

Figure 2.1: Generic workflow of a model simulation run, derived from the workflow of the model MITRAS.

this process, several input and output files in various file types and formats are generated and used (white boxes). First, in the pre-processing, the model domain needs to be created. Input files containing information on e.g. terrain height, obstacle heights and land use are used. They might need to be prepared for the actual domain pre-processor, which generates topographic and obstacles input files for the actual simulation. After the domain pre-processor, the model simulations need to be initialised (e.g. 1D simulation), requiring the model domain and meteorological input parameters to set up the initialisation. Results of the 1D initialisation can be meteorological profiles, which are then used as input for the 3D simulations. Together with the topography and the obstacle information, the 1D-simulation output is then used for the 3D simulations. With the finished 3D model simulations, the model results need to be post-processed (e.g. translated into readable data formats, like netCDF). Afterwards, the data producer analyses the results and ideally prepares the data for publication.

An overview of the standard MITRAS model input and output files is summarised in Appendix A. Not all information that occurs within the simulation process is written to or provided in the final model results. Furthermore, not all information is ultimately required by the model user or has been taken into account. This is partly for reasons of memory and model performance, as writing takes up storage space and modelling time. Moreover, the output depends on the developer or model user, which information is needed or requested in the final results. For example, the application of the AtMoDat standard requires information which was initially not provided with the overall model output. Certain aspects that are relevant for the standardisation of MITRAS data are already available to a certain extent, but have not yet been included in the final output. Finding these structures and maintaining them throughout the process must be taken into account in future model development. Adaptations to fulfil the standardisation of MITRAS model data are addressed in the Chapter 4, applied to the model-own post-processor (m2cdf) and the newly developed post-processing routine nc2atmodat.

2.2 Physical models

Within the range of artificial intelligence, computational, numerical models and atmospheric observational measurements, physical models can be used. A physical model utilises a wind tunnel or water tank to simulate the atmospheric flow over a small-scale replication of the model domain, e.g. an urban neighbourhood or a building, that is placed inside the wind tunnel facility (Oke et al., 2017; Zhao et al., 2023). Similar to numerical models, physical models are suited to investigate the flow over complex urban areas, the effect of the vegetation on the flow (Gromke and Ruck, 2007), ventilation (Ali et al., 2023), the dispersion of air pollutants or gases (Schatzmann et al., 2006; Schalau et al., 2022), thermal comfort or wind comfort (Zhao et al., 2023).

A wind tunnel model must be set up in a way that the resulting flow is similar to actual atmospheric conditions. Since the fluid in a physical model follows the same physical principles as the real-world atmospheric flow, under certain simulation conditions, wind tunnel model results can be transferred to the real world. This can be achieved if the flow in a wind tunnel simulation fulfils similarity in dynamic and geometric properties to real atmospheric conditions (Oke et al., 2017; Verein Deutscher Ingenieure, 2024). Geometric similarity is achieved if the wind tunnel obstacles are in shape and scale similar to the real world. Dynamic similarity can be achieved by achieving a similarly behaving turbulent flow. A measure for such turbulent flow is the Reynolds number (Eq. 2.2), which defines the relation between inertial and viscous fluid (Snyder, 1981).

$$Re = \frac{u \cdot L}{\nu} \tag{2.2}$$

Here u is the wind speed in ms^{-1}, L the characteristic length in m and ν the viscosity in m^2s^{-1} (Verein Deutscher Ingenieure, 2024). The more turbulent the flow, the higher the resulting Reynolds number. Above a critical Reynolds number, the turbulent flow properties do not change significantly and behave similarly. Therefore, if the Reynolds number exceeds the critical value, it can be assumed that the resulting turbulent flow in a model domain over geometrically similar obstacles is comparable to a similar turbulent realistic flow.

The implementation of wind tunnel model simulations follows a similar procedure to that used for numerical models (Figure 2.2). The workflow is also standardised by the VDI standard 3783 Blatt 12 (Verein

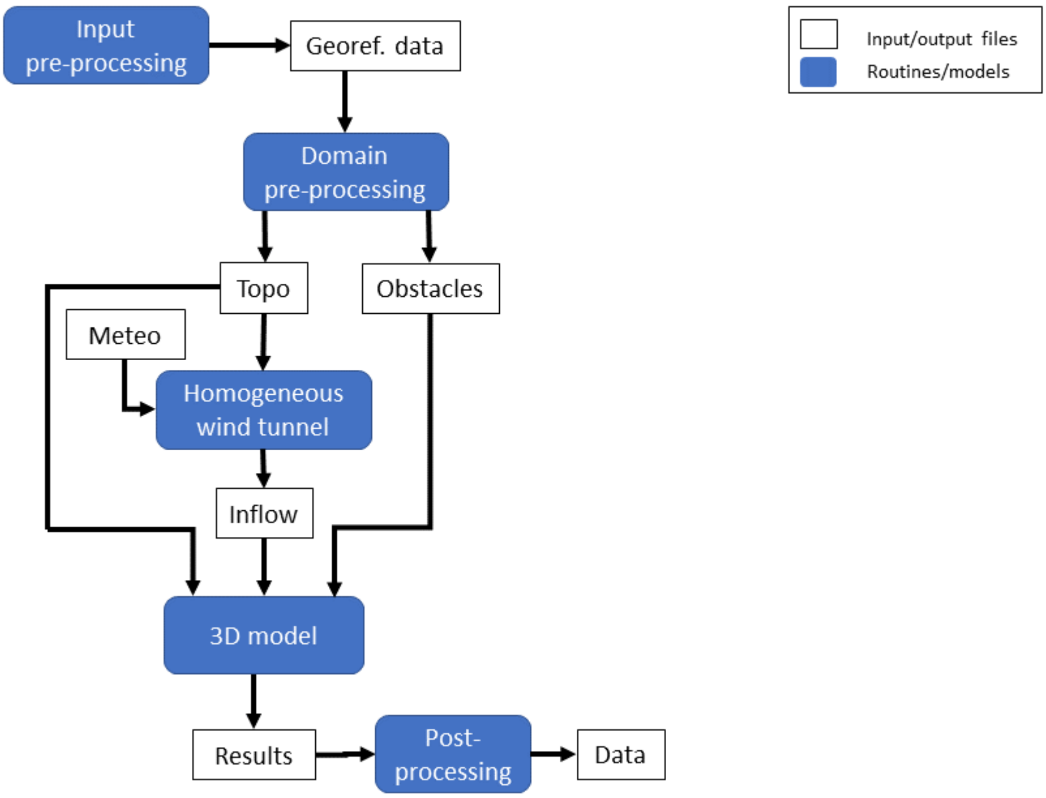

Figure 2.2: Generic workflow of a model simulation run for wind tunnel model application. The process components are similar to the numerical model in Figure 2.1.

Deutscher Ingenieure, 2024). Similar to the numerical model, three main steps are required. In the pre-processing of a wind tunnel model simulation, a model area must be constructed. The model needs to be set up to create similar conditions and wind flow to the conditions in the real world. Since a wind tunnel model is a scale model, the scales used for the simulations depend on the size of the facility. For a realistic scenario, georeferenced data containing building positions and heights can be used. These are then scaled down to the desired scale for the model.

Before the model domain can be positioned inside the wind tunnel, adequate boundary layer conditions need to be set. The development of a basic setup includes the generation of a certain atmospheric boundary layer (Zhao et al., 2023). For this, the empty wind tunnel is equipped with roughness elements that induce turbulence to create the desired wind profile for the simulations. This step is adapted as long as the desired boundary conditions are established. After initialisation of the wind tunnel, the model domain is placed inside the prepared wind tunnel, and a simulation can be conducted. At several positions, measurements (time series, repetitions) are done.

After the simulations, in the post-processing, the measured data is processed and analysed and by the wind tunnel modeller. In the end, the model data gets selected and prepared for publication and reuse processes. Furthermore, the documentation of model simulations happens for numerical models document the simulation setup in their input files within the code, wind tunnel studies need to be documented as well, to ensure, that the simulations can be repeated.

Wind tunnel model settings need to be well-documented and controlled to ensure the repeatability of the simulation. A wind tunnel modeller must ensure that the chosen measurement conditions are suited for the purpose of the simulation. Since wind tunnel model data is generated via measurements of the flow at different locations in the model domain, the measurements have to be repeated several times to reduce errors and measurement uncertainties in the results. Information about characteristic lengths, times, model scale, reference wind velocity and measurement point positions needs to be documented. In the post-processing, model results are provided as dimensionless data. The measured data are normalised by the reference wind speed, which is the overall achieved wind velocity during the wind tunnel simulation process, which is measured at a certain height and position. Therefore, typical time scales and characteristic lengths are also provided, as they are necessary information for the model simulations. With these values, the results become re-scalable to different possible real atmospheric conditions.

2.3 Limitations

Numerical and wind tunnel models provide a wide range of applicability to simulate and study atmospheric processes in urban applications. Since the model simulation process is schematically overall similar (see Figure 2.1 and Figure 2.2), differences occur in the technical aspects, possibilities and limitations, which depend on the used model (laboratory or facility or implemented code and parametrisations). Notable differences and similarities in the model properties between the numerical model and a Wind tunnel model are summarised in Table 2.1.

Generally, both models are limited by several factors, most notably by spatial and temporal resolution. Simplifications occur by using parametrisation, chosen grids and structures, model boundaries and uncertainties in the model results. E.g., the level of detail of the obstacle (e.g. buildings with complex outer shapes like balconies, canopies, roofs) determines the accuracy of the model results. A wind tunnel model can resolve and consider such obstacle detail elements rather easily, whereas a numerical model must often neglect such details due to the grid size and resulting resolution.

Numerical models are limited by the implemented approximations and parameterisations used to calculate and resolve certain processes. Therefore, a common topic in numerical modelling is the model

Table 2.1: Overview of model qualities for numerical models and physical models

Property	**Wind Tunnel**	**Numerical Model**
Variables	fluxes, concentration of gases and passive tracer, wind	cloud water/rain water, fluxes, concentration of gases, aerosols and passive tracers, humidity, precipitation, temperature, wind
Domain size	depends on the scale, typically few $100m \times 100m$ to $1km \times 1km$	$100m \times 100m$ to $1km \times 1km$
Grid	regular, irregular distribution of measurement points	Regular, irregular grid, covering the whole model domain.
Spatial resolution	measurement probe volume	depends on the grid size
Time step	measurement frequency dependence	CFL criterion (see Eq. 2.1)
Time resolution	depends on timestep and scale	Depends on parameterisation of phenomena
Scale	e.g. 1:500	1:1
Speciality	dependend on input data and model builder, high resolution, arbitrary shapes of obstacles possible (roof shapes, diagonal buildings, balconies, trees)	E.g. diurnal cycle, radiation processes, precipitation, chemical processes, parametrisation
Output formats	Excel, ASCII	binary, netCDF, ASCII
Documentation	text documents, e.g. PDF or header in output files	text documents, e.g. PDF, input files (binary, ascii)

development to improve the model performance or to include atmospheric processes and phenomena into the possibilities of modelling.

Limitations to physical modelling include the required effort to conduct simulations, since the preparations (e.g. model setup and building) and the actual simulations (boundary layer measurements, actual model measurements, repetitions) are time-consuming. The model boundaries and the size of the model domain are determined by the actual size of the wind tunnel or water tank. Wind tunnel simulations are mainly used to assess the lowest atmospheric layer, the surface layer, where such effects can be neglected more easily. If not technically equipped, physical models are not able to represent all atmospheric conditions that could happen in reality. Unless using a rotating water tank, the Coriolis force cannot be replicated in the wind tunnel. Unless a stratification model is used, atmospheric stratification, which can range from stable to unstable in reality, is not considered. Therefore, mostly neutral stratification is simulated.

The correct use of wind tunnel data also depends on the user. Since wind tunnel model data is provided as dimensionless data, the data can be scaled to different weather scenarios and conditions. To use the data, a wind tunnel data user needs certain knowledge and experience in handling this data in order to use it correctly (Leitl, 2000; Leitl and Schatzmann, 2010). Furthermore, the environmental wind tunnel modelling community has no overarching community-wide unified data standard that defines how data should be provided, named or structured. Without data standardisation, there are no uniform descriptions, data layouts or variable names that are used consistently across communities. Each working group or data producer, therefore, uses their own formats and descriptions, which can make it difficult to reuse datasets.

2.4 Data comparisons

In the past, many research projects that investigated urban atmospheric processes utilised observational measurements, numerical and physical models. Furthermore, a common practice in this field is the comparison of model or measurement results with each other. Data comparisons contribute to the understanding of the behaviour of certain atmospheric processes. Also, data comparisons are often used in the process of evaluating models; to estimate their accuracy, validate their mechanics and parametrisations, and draw conclusions about their performance Schatzmann and Leitl (2011). Also, numerical model simulations are mainly initialised after the assessment of observational data (Vieira Zezzo et al., 2023) to pick the right boundary conditions.

Examples for the investigation of atmospheric processes that incorporate data comparisons with observational datasets are the DAPPLE project (Martin et al., 2010) or VALIUM (Schatzmann et al., 2006), in which observational measurements of air pollutants in an urban street canyon (DAPPLE: London, UK; VALIUM: Hannover, Germany) were compared with wind tunnel and numerical modelling results. Further examples for model data comparisons are provided in Song et al. (2018), where the ventilation of buildings is studied, by incorporating Wind tunnel simulations, CFD-LES modelling and observational measurements (indoor and outdoor) from a full-scale building. Comparisons between wind tunnel data and observation measurements are described in Fischer (2021), to determine methods and guidelines for the evaluation of model results to determine the representativeness of Wind tunnel measurements. Examples for data comparisons between numerical model and observational data to investigate urban flow were carried out for PALM-4U (Paas et al., 2021) and for air quality aspects with MITRAS (Schatzmann et al., 2006; Grawe et al., 2013).

Data comparisons also play a role in model development, as they enable the validation of model processes to ensure that a model is suitable for the intended application and provides reliable results. For this, different guidelines for model standardisation processes are available Meroney et al. (2016). The numerical models MITRAS (Grawe et al., 2013), OpenFoam (Franke et al., 2012), MISKAM (Schatzmann and

Leitl, 2002) or PALM-4U (Gronemeier et al., 2021) were validated with Wind tunnel data.

One example for such a model evaluation guideline are the VDI standard 3783 Blatt 9 (Verein Deutscher Ingenieure, 2017), issued by the German Association of Engineers (VDI). They set out the requirements that numerical atmospheric models must fulfil to be evaluated and certified for applications. They describe the aspects that the models should calculate and represent, and provide test case setups and comparison data for evaluating the models. Additionally, the publication of documentation or model descriptions is required for approval by the VDI. However, the VDI standard do not specify the provision of the resulting model data. Another example is the guidelines created within the COST Action 732 (Franke et al., 2012). Running from 2005 to 2009, COST Action 732 developed an overview on practices, datasets and guidelines on the comparison of CFD model data with wind tunnel measurements, defining actions and methods for this approach (Franke et al., 2012; Di Sabatino et al., 2011). It included the development of improved model evaluation processes with guidelines, the provision of datasets for model evaluations, were many other objectives of projects. This project resulted in the generation of the comparison datasets. They are generated in the MUST (Mock Urban Setting Test) project and used in studies to evaluate numerical model datasets results (Sabatino et al., 2011).

Comparison datasets for the evaluation of numerical models are often generated by wind tunnels. For example, the Environmental wind tunnel laboratory facility at the Universität Hamburg [4] provides its own database of datasets, which contains datasets generated for various research projects and specifically for model evaluation (e.g. COST ES 1006, COST Action 732, CEDVAL). Further databases and data collections can be found through the European Research Community on Fluids, Turbulence and Combustion[5] (ERCOFTAC).

However, as data comparisons are common practice and regulated in urban atmospheric science and model development, the provision of datasets in this discipline is not universally standardised. Instead, each project defines its own methods and a project-specific way to provide and prepare data. In future, with the increasing impact of climate change, scientific research in urban areas and on urban atmospheric processes and their methods will further increase (WMO, 2023; Baklanov et al., 2018) and thereby increase the need of the publication of research data. Therefore, it seems timely to investigate methods for the provision of standardised model data in this discipline.

[4] https://www.mi.uni-hamburg.de/en/arbeitsgruppen/windkanallabor/data-sets.html, accessed: 09.09.2025

[5] https://www.ercoftac.org/, accessed: 19.09.2025

3 Survey on output data from obstacle resolving atmospheric models

This chapter was published as:

Voss, V., K.H. Schlünzen and D. Grawe (2024): Survey On Output Data From Obstacle Resolving Atmospheric Models. In *Meteorologische Zeitschrift* Vol. 33 No. 3 (2024), p. 243 - 253, DOI: 10.1127/metz/2024/1217

This Chapter includes the full manuscript of the published paper. Layout and numbering of the paper have been adapted to this thesis. References have been combined at the end of the thesis. Acknowledgements submitted with this manuscript are provided in Appendix B.

K.H. Schlünzen and David Grawe have contributed to the conceptualisation, survey design and discussion of the results.

Abstract

Obstacle-resolving micro-scale atmospheric models (ORMs) are important to assess atmospheric processes within urban areas. However, the data generated from such models are not standardised yet and existing data standards do not fit properly to those data. Standardised model data can help to prepare FAIR data publications, which foster the process of data reuse, sharing, comparison and distribution. Therefore, an online survey was distributed among ORM users and developers in the urban meteorology and urban climate community. The survey should help to assess, which models are currently in use and to collect suggestions and requirements from the participants on how a data standard should look like. The aim of the survey was not to test the knowledge of the participants about the model they use but rather to get an overview of how they understand, use and work with ORMs. Based on 14 finished survey entries, it shows that ORMs provide and handle their data differently. The participants are aware of the need for standardisation, preferring netCDF as data format and suggesting extending existing standards to the needs of ORM data.

3.1 Introduction

In atmospheric science, modelling is a common method to investigate and study atmospheric processes, and data generated by such models (named ”model data” hereafter) are frequently shared, compared, and reused within the modelling community. Therefore, model data has to be prepared and published in a way that the data can be easily accessed, shared, reused and processed by other researchers or data users. It is recommended to prepare data after the so-called FAIR principles, as described in Wilkinson et al. (2016). FAIR means that data must be prepared in a way, that it will be findable, accessible, interoperable and reusable. A way to achieve FAIR data publication is to apply data standards to data so that the data is prepared in a certain way, that allows a generic use of the data. Standards can define for example the usage of certain data formats and descriptions, variable names, metadata information or file naming formats.

Standardised model data are especially important in so-called model intercomparison projects (MIP), where standardised model results simplify the comparison and reuse of model results. The most prominent MIP is the Coupled Model Intercomparison Project (CMIP) (Eyring et al., 2016; Touzé-Peiffer et al., 2020), which is part of the World Climate Research Programme[6], where global climate models are used to investigate the earth's climate and the effects of climate change. This project played an important role in developing the technical requirements for standardised model data (Eyring et al., 2016), by establishing the usage of the Network Common Data Form[7] (netCDF) (Rew and Davis, 1990) together with the Climate and Forecast (CF) Convention (Eaton et al., 2022). NetCDF is a self-describing data format, that provides the possibility to store large and multi-dimensional data and metadata into one file (Rew and Davis, 1990), while the CF Conventions provide guidelines on how to describe data stored in the netCDF format so that the files are self-describing. The CF conventions provide controlled vocabulary via standardised variable names which can be found in the CF Standard Name Table[8]. netCDF and CF are well-established and widely used in atmospheric sciences.

However, the existing conventions are mainly adapted to model data from global or regional models. Model data generated outside the scope of global or regional scale models are not sufficiently covered by the existing conventions. One group are obstacle resolving atmospheric models (ORM), which are also called micro-scale models. They are with increasing computer power more and more be used to assess atmospheric processes in complex urban areas (Oke et al., 2017; WMO, 2023). ORMs are used to investigate processes in the urban atmospheric boundary layer, as they are designed to be applicable to resolve small-scale atmospheric phenomena by considering obstacles e.g. buildings and trees explicitly. These obstacle induced effects and induced phenomena are at best parameterised in larger-scale models. However, direct information is most relevant to assess the atmospheric conditions (e.g. temperature, humidity, radiation, wind) within the urban canopy at pedestrian height level.

[6]https://www.wcrp-climate.org/wgcm-cmip
[7]https://docs.unidata.ucar.edu/netcdf-c/4.9.2/index.html
[8]https://cfconventions.org/Data/cf-standard-names/current/build/cf-standard-name-table.html

Studies with ORMs often involve e.g. the thermal conditions and distributions in street canyons (Bohnenstengel et al., 2004), urban heat island effects (WMO, 2023), effects of climate change and weather in cities (Scherer et al., 2019; Ferner et al., 2023), ventilation and dispersion of air pollutants (Vardoulakis et al., 2003; Martin et al., 2010; Salim et al., 2015; Hassan et al., 2022). ORMs may be numerical or physical models, both resolve obstacles and consider their effects on the atmospheric flow in the simulation. Another important method to investigate urban atmospheric processes are measurements, collected by observational campaigns or at operational measuring sites (Scherer et al., 2019; Martin et al., 2010; Paas et al., 2021). These measurement data are often used to evaluate and assess ORMs (Paas et al., 2021).

Figure 3.1 shows an example of a simulation result of high spatial resolution, showing the wind speed at 2.5 m above ground in the Hamburg city centre, calculated with the ORM MITRAS (Schlünzen et al., 2003; Salim et al., 2018). This figure shows the position of the obstacles in the model domain, that were considered in the model run. Their influence on the flow can be seen, leading to reduced wind speeds between buildings and within the build-up areas and some relatively larger flow speeds between some obstacles. A variety of ORMs exist (Vardoulakis et al., 2003; Jänicke et al., 2021; WMO, 2023), which are different from each other in their model properties and in their ability to produce reusable model data. They differ in technical model features, e.g. in the choice of filtering methods, numerical grids, numerical solution techniques equations solved and parameterisations used in the application. In application, different temporal and spatial resolutions are used and the model output data are stored in different ways. Each model has a model-specific way of generating and writing model data, output data are stored in various data formats, and model-specific variable names are used. The usage of model-specific file formats for model results might lead to the need for specific analysis tools or data converters, which hinders the reusability of the model data if the required tools are not accessible or published. A peculiarity of ORMs is that many properties and variables commonly used in ORMs, such as the city structures, the obstacles, their surfaces and associated variables, e.g. wall temperatures, radiative fluxes on building surfaces, or precipitation on roofs, are not standardised yet. All these aspects make the reuse of model results from ORMs difficult and time-consuming. Existing ORM data are not sufficiently self-describing and additional information is needed, which makes the reuse difficult.

However, due to the high spatial resolution, the time steps needed to ensure computational stability is very small and computing resources large. Thus any model data reuse would be of help, especially since the demands on ORMs increase, using them to assess e.g. the effects of urban climate adaptation or mitigation measures on the local surroundings.

At the same time, ORMs are still in development, which means adaptation of output standards may still be relatively easy to be coded in the existing models and their connected programs. Thus, there is a need for well-described and easily reusable ORM data (WMO, 2023). An application of a model data standard to ORM data is highly preferred.

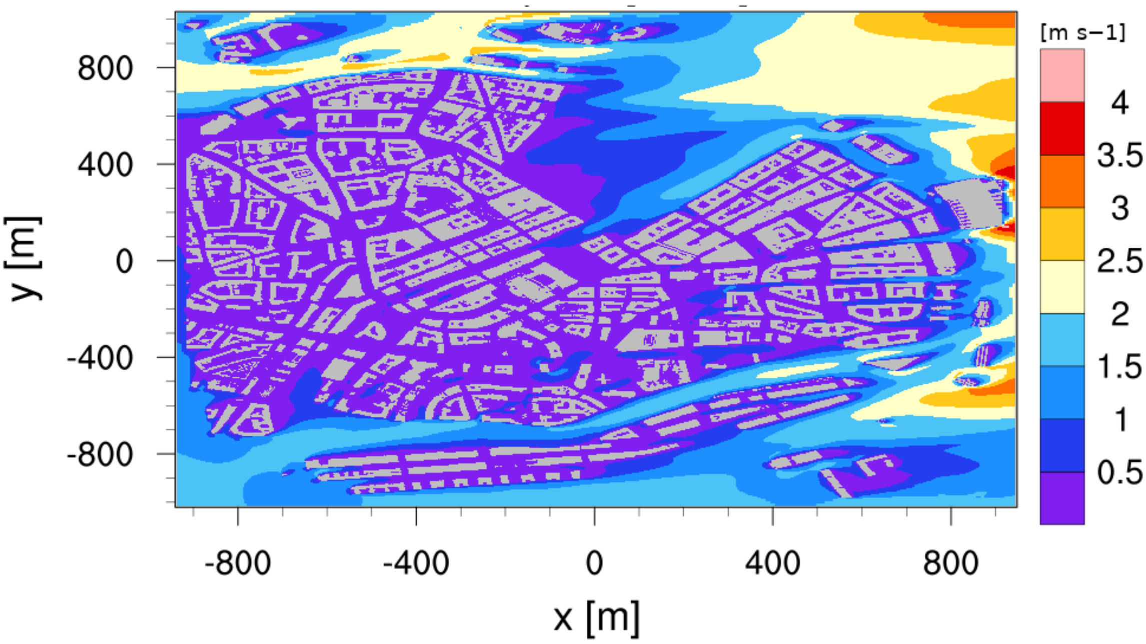

Figure 3.1: Example of the simulation results of an ORM. A horizontal grid width of 2.5 m was used and a south-westerly (232 °) inflow with a wind speed of 3 ms^{-1} was specified at an altitude of about 1 km above the ground. Data is published as Voss (2023a).

Some ORMs are adapting the CF conventions to their model output, like PALM-4U whose model data standard is based on the CF conventions (Scherer et al., 2022) and MITRAS, which has adopted the CF conventions and the AtMoDat data standard. The AtMoDat data standard, developed by the AtMoDat (Atmospheric Model Data) project[9], is a model data standard that provides guidelines for the standardisation of model data, which cannot be sufficiently described by the CF conventions alone and if no discipline-specific standards are available (Ganske et al., 2021, 2022). First model data successfully standardised according to the AtMoDat data standard are already published (Ganske et al., 2022; Voss, 2023a). To get an overview of ORMs used in urban climate and meteorology applications a survey was created to collect the model user's perspective. An additional objective was to involve the modellers' community in the development of an ORM data standard. An online survey was created and distributed at conferences and via e-mail to different (ORM) atmospheric modeller groups (Section 3.2). The survey aimed to collect information on various properties of ORMs as assessed by the users, their generated data, as well as methods used by the participants of the survey to (re-)analyse the data. The survey was not meant to test the user's knowledge of the used models, but to get an idea of what the user is experiencing during model and model result handling. Based on the results of the survey (Section 3.3), recommendations for an ORM data standard were derived (Section 3.4). The survey setup and results will be further discussed in this paper and conclusions concerning a standard drawn.

[9]www.atmodat.de

3.2 Methods

The development of a general model data standard for ORMs should consider, what different ORM properties exist, what information is stored in the model output, and how model data from different ORMs is provided. This knowledge will help to find common ground for the requirements for what is needed to standardise ORM data. To collect this knowledge and to involve many people in this process, an online survey was created and globally distributed among the numerical urban micro-scale meteorology and urban climate modelling community. The survey aimed to collect ORMs used in the meteorological community and was therefore prominently advertised in this discipline. The survey was aiming to collect the model user perspective. Thus, all answers were taken as the users provided them and no checking for completeness or correctness done.
The survey was set up with the online survey tool “LimeSurvey” and was accessible via https://uhh.de/orm-survey until June 2022. The survey was initially released at the “37. International Technical Meeting on Air Pollution Modelling and Its Application” - Conference in September 2019 (Voss et al., 2021). Furthermore, the survey was presented at the Air Quality Conference 2020, WMO GURME Meeting 2020, the European Geoscientist Union General Assembly 2020, the AGU Fall Meeting 2020, the “12. Deutsche Klimatagung 2021”, European Meteorological Society Annual Meeting 2021 and “METTOOLS XI 2021” and sent via e-mail to microscale modeller groups working in the field of urban meteorology and climate. The survey was closed in June 2022.

The survey consists of two parts. The first part covers model-related questions, where the survey participants had the option to name all models, which they used themselves and/or name the models from which they only use the model results. For each model named by the participant, information about model properties, e.g. grid, coordinate systems, initialisation and model settings can be given. The second part addresses the participants work with the model data, regarding analysis tools and data format conventions. Lastly, the participants could comment on problems they faced while handling model result data.

3.3 Survey results

The survey was accessed 40 times, with 14 completed and 26 incomplete survey entries. The majority of incomplete survey entries were cancelled after the second page. In some of the incomplete entries, at least model names were provided. In each survey entry, each participant was able to submit up to 14 models, which they worked with or used the model data from. Some completed entries might therefore contain more than one model or only model data handling information, which is why the number of total entries and amount of named models can be different. In this work, each entry is viewed as one person participating in the survey. As mentioned, the survey is divided into two parts. The first part contains seven questions covering model-related aspects and the second part contains six questions related to model use. For the model-related questions, eight entries are available, providing information for 12

ORMs. This means for each of the seven model-related questions a maximum of 12 answers is possible. Regarding the model-use-related questions, a maximum of 14 answers can be reached. Note, that even if an entry was finalised, not all questions were answered by all participants.

3.3.1 General comments on answers

Considering all 40 survey entries, including the 14 completed and 26 incomplete entries, a total of 26 model names were submitted by the participants, which can be summarised into 13 different models. Table 3.1 gives an overview of the 13 named models, that were submitted via the survey and frequencies how often they were mentioned.

By considering only the answers from the 14 completed entries, a total of 15 model names are mentioned by the participants and therefore available for the analysis. This includes duplicate mentions of the same model. From the 14 completed entries, three of them only provided information to the model-use related questions, leading to 11 entries with model information. In the 11 entries with model-related content, 15 models were mentioned but only eight of the 11 entries provided information for ORM. In these eight entries, 12 ORMs were mentioned and further analysed in this work. These 12 ORMs from the completed survey entries can be summarised into seven different ORMs: Ansys Fluent, Code_Saturne, ENVI-met, MISKAM, MITRAS, PALM-4U and PMSS.

This means, for the analyses of the model-use related part all 14 completed survey entries are considered, while for the analyses of the model-related part of the survey, only 8 entries with the information for 12 ORMs are available.

The models CALPUFF, CFD and OpenFOAM were mentioned, but the corresponding entries were incomplete and there is no information provided for these models by the contributors. Also, CFD is an abbreviation for "Computational Fluid Model", which is a model type to which most of the models mentioned here belong. This answer is therefore too generic to be counted as a model. The model LASAIR 5* is a particle measurement device and not a numerical model. The SWAT MODEL is a hydrological model and does not fit in the target group. METRAS is a mesoscale regional climate model and is not used for obstacle resolving modelling. The entries for these models are therefore neglected in the analysis.

The most named ORM in this survey is PALM-4U with seven (four completed) entries, followed by the model MITRAS with four (three completed) entries. Ansys Fluent, Code_Saturne, ENVI-met and PMSS had each two entries with only one finalised for each. MISKAM has one entry that was finalised. Note that the number of answers is not representative for the number of publications to be found for the model or its results. Regarding PALM-4U, some participants mentioned the models PALM and PALM-4U. Based on the documentation of PALM and PALM-4U, PALM-4U is linked with the model PALM.

*Correction note: LASAIR is not only the name of a particle measurement device, but also the name of a software programme of the same name used for dispersion simulations. This was not correctly stated in the published paper.

Table 3.1: Named models with all (Total) and completed (Compl.) survey entries. Only models in bold are further analysed in this work (for details see text).

Model Acronym	Full name	Total	Compl.	Comment
Ansys Fluent	Ansys Fluent	2	1	-
CALPUFF	California Puff Model	1	0	No specific given
CFD	Computational Fluid Dynamics model	1	0	No specific given
Code_Saturne	Code_Saturne	2	1	-
ENVI-met	ENVI-met	2	1	-
	PMS Lasair III 5100 Aerosol Particle Counter Models	1	1	Measurement device, not an ORM
METRAS	Mesoscale Transport and Stream model	1	1	Not an ORM
MISKAM	Microscale climatic and dispersion model	1	1	-
MITRAS	Microscale Transport and Stream model	4	3	-
OpenFOAM	Open-source Field Operation and Manipulation	1	0	No specific given
PALM-4U	Parallel Lagrange Model	7	4	-
PMSS	Parallel Micro SWIFT SPRAY model	2	1	-
SWAT MODEL	Soil & Water Assessment Tool	1	1	Ground water model, not an ORM.
Sum		**26**	**15**	**12 different ORMs compl.**

PALM turns automatically into PALM-4U if buildings are included in the model simulation and certain modules are activated[10].

Therefore, all answers assigned to these models are summarised as PALM-4U in this work. Table 3.2 provides information on the models based on literature and additional sources to get an overview of the ORMs from completed entries in this survey. The submitted answers from the participants are summarised in the answers to the corresponding question. Model information submitted by different users could lead to contradicting statements, however, this was not the case. But there is no check if the statements are true to each model. Checking with documentation provided in some cases different answers. No corrections were made since the goal was to get an overview of models based on the user's view and the assessment of the model user about certain aspects of the models they use. Missing or uncertain information in the survey is a valuable answer for setting up a standard as it shows, where model and model data users need information, do not find it easily in the output and thus might have wrong assumptions of the model or the data they use.

3.3.2 Model related questions

The first part of the survey contains seven questions about the model properties. The answers to the questions had to be provided individually for each model, that was submitted by the participant. The aim is to get an overview of the properties of the different models as the user assesses them. In the following, the answers are presented for the seven questions. If the question was not a multiple-choice question, a maximum of 12 answers is available in this part of the survey.

3.3.2.1 Meteorological and technical initialisation settings of the models The first two model-related questions are about the meteorological and technical initialisation settings for model runs, and whether these settings are stored and provided with the model output. Meteorological initialisation settings include the physical setup of the model, like initial wind speed, direction, humidity, and temperature (soil, water, atmosphere). Technical Initialisation settings should include the model setup like model domain, obstacle structure, parameterisations, boundary conditions and assumptions that are used to generate the input data for the model. That information is needed by a data reuser to better understand the model characteristics and the meteorological situation studied. To get an overview of if and how an ORM provides this data, the first two questions were: *"Are the meteorological data of the initialisation stored in the output?"* and *"Are the (technical) initialisation settings of the model simulations stored in the output?"*. If the model does store the data, the participant could also mention, how the model stores this information.

Regarding the meteorological settings, based on 12 answers from the participants, for seven models the participants selected that the model stores this information in the model output. For one model, the participant selected that the model does not store this information in the model output. In four entries

[10]https://palm.muk.uni-hannover.de/trac/wiki/palm4u

Table 3.2: Characterisation and references for obstacle resolving models with finished surveys.

Model name	Model type	Documentations and further information
Ansys Fluent	Fluid simulation software	Poole et al. (1995), https://www.ansys.com/products/fluids/ansys-fluent
Code_Saturne	CFD simulation model	https://www.code-saturne.org/cms/web/
ENVI-met	Micro-scale model for urban climate	Bruse and Fleer (1998), https://www.envi-met.com/
MISKAM	Obstacle resolving micro-scale atmospheric model	Eichhorn and Kniffka (2010), https://www.lohmeyer.de/en/about-us/work-methods/numerical-models/miskam/
MITRAS	Obstacle resolving micro-scale atmospheric model	Salim et al. (2018), https://www.mi.uni-hamburg.de/en/arbeitsgruppen/memi/modelle/mitras.html
PALM-4U	Obstacle resolving micro-scale atmospheric model	Maronga et al. (2020), https://palm.muk.uni-hannover.de/trac/wiki/palm4u
PMSS	Lagrangian particle model	Oldrini et al. (2017), https://www.aria.fr/aria_city.php

the participants selected either "uncertain" or provided "no answer" to this question. The seven "yes" answers contained further information for four models which provides their meteorological initialisation. One model user mentioned that their model provides this information with a "C++ script". Another model stores this information in so-called "Meteo files". For a third model, the model users submitted that the model writes the meteorological run settings "in the first timestep", in "report files" and in "3D arrays". For a fourth model, the participants mentioning this model wrote, that this model stores this information into "netCDF", into a "model specific output standard" and into a "model specific output standard at t=0". Thus, a variety of ways is used for storing the initial meteorological information, if the model stores this information at all. Model data handling is quite different for each model and ranges from input scripts to output files. It needs to be considered for an ORM data standard that this information should be (more easily and standardised) available.

Regarding the technical initialisation settings, from 12 submitted model related answers, five ORMs store the technical initialisation settings in the model output. The majority of the participants, seven in total, are either uncertain or did not provide an answer ("No Answer") to this question. The answer option "No" was not selected. Here it becomes apparent that the participants are rather unsure whether the models store this information. For example, some participants who use the same model say that this information is stored, while others are not sure or did not give an answer. Based on the five "yes" answers, each answer describes a different way to store the information on the technical initialisation settings. One participant writes that one model stores technical simulation settings in a "setup.log". Another model seems to store this information in "report-files" and "profiles of the 1D meteorological variables". A third model stores this information in a "Fortran namelist" and a so-called "netCDF driver file". This information is then available for the user to interpret the model data.

The handling of input and initialisation information of the model simulations shows the variety of how the different models store such information in the output variable to model users. As the ways to provide input and model set-up data are model-specific, accessing the actual information might be difficult for model users hindered by certain file format structures. Making the meteorological and technical initialisation information accessible in a standardised way in the output data can help to understand the model settings and meteorological information simulated as well as repeat the simulations with the same or another model.

3.3.2.2 Grid and coordinate system For answering the question *"What coordinate system is used?"*, concerning the horizontal and vertical coordinate system used by the model, the participant was able to choose the answer from common coordinate systems or the participant could provide an own answer. Multiple-choice answers were allowed. For the horizontal coordinate systems, the participant could pick from the following choices: "Cartesian coordinates", "Lambert conformal coordinates", "geographical coordinates (lat./lon.)" and "rotated geographical coordinates (lat./lon.)". For the vertical coordinate system, the participant could select between "z-coordinates", "surface fitted grid coordinates", "pressure coordinates" and "sigma coordinates". Additionally, it was possible to provide an own answer. Table 3.3 shows the results of the total numbers for each coordinate system, that was selected. Based on the results in Table 3.3, the most used coordinate system in the horizontal direction is the "Cartesian coordinate" system with five entries, followed by the "Lambert conformal coordinate" system, with three entries (Left two columns in Table 3.3). For the vertical direction, the majority of ORMs use "z-coordinates" (seven entries; right columns in Table 3.3). "Surface fitted grids" was selected three times. "Sigma coordinates" or "pressure grid coordinates" are not selected.

Some participants did not provide information on one of the directions, and some did not provide both. This was the case for three models, where the participants selected "unknown" in the corresponding survey entries, meaning that they did not know of the coordinate system used by the model. Thus, a misinterpretation of the model grid might occur. This underlines the relevance of coordinated information in the model results. The representation of the model data is affected by the grid the model is using. Therefore, the question *"Which grid type is used by the model?"* was asked. This is relevant for the representation of the model domain and an important factor for grid sizes and the total amount of grid cells which affects the performance of the model simulation. Therefore, it is important to see which grid types are used by the mentioned ORMs. The participant could select between "uniform grid", "non-uniform grid" and "Arakawa-A", "Arakawa-B", "Arakawa-C", "Arakawa-D" and "Arakawa-E" (types of grid staggering) which are the most common grid types. Other grid types can be mentioned by the participants as well if needed. Multiple choice was possible. The results are summarised in Figure 3.2. Most models use similar grid types. The majority of models use an "Arakawa-C" grid with a "uniform" or "non-uniform grid". Other "Arakawa" grids than "Arakawa-C" are not used. Other grids, mentioned by the participants are "unstructured grid". For one model, no grid information was provided.

3.3.2.3 Filtering method The filtering method is relevant for the spatial and temporal representativeness of a model. Three types of filtering methods are available: "Large Eddy Simulation (LES)", "Reynolds Averaged Navier Stokes (RANS)" and "Direct Numerical Simulation (DNS)". The participants were asked, *"What filtering method uses the following model"* and should either select between the

Table 3.3: Results of the survey on the horizontal and vertical coordinates as provided by the participants.

Horizontal	**Nr.**	**Vertical**	**Nr.**
Cartesian coordinates	5	Z-coordinates	7
Lambert Conformal coordinates	3	Surface fitted grid coordinates	3
Rotated geographic coordinates (lat./lon.)	1	Pressure coordinates	0
Geographical Coordinates (lat./lon.)	1	Sigma coordinates	0
Unknown	2	Unknown	2

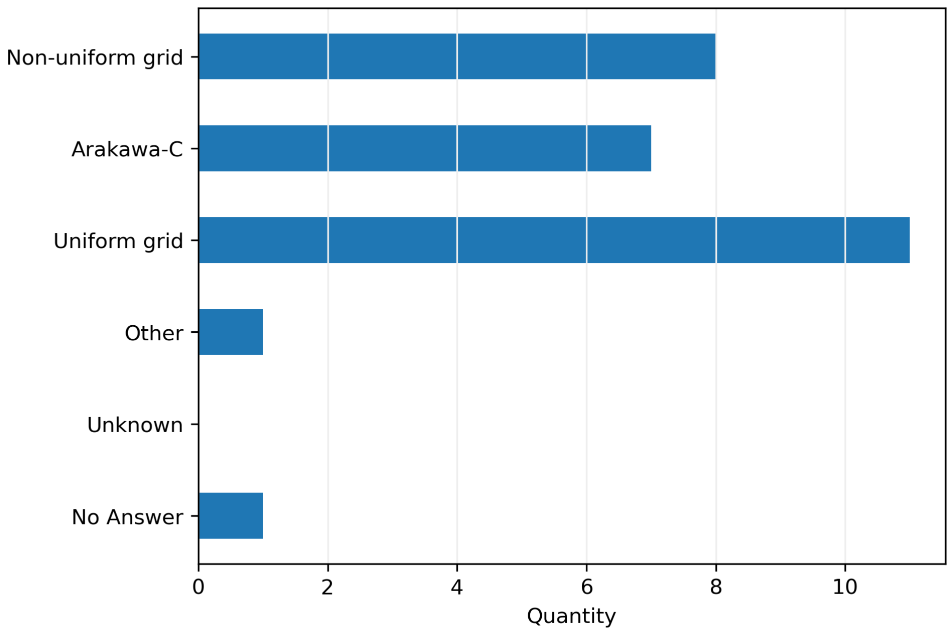

Figure 3.2: Results to the question of grid types used by the models as provided by the participants

three types or add other filtering methods, if their model uses other filtering methods. Multiple answers were possible.

Based on the answers of the participants almost all models use "RANS" filtering, three models additionally use "LES". For one model, the participants write that it allows all three filtering methods, "RANS", "LES" and "DNS". One of the models (PMSS) is not a CFD model but a Gaussian plume model and uses a "mass consistency algorithm" (Oldrini et al., 2017).

3.3.2.4 Output format and conventions The reusability of the model results is directly connected to the aspect, how the model output data is stored (Question: *"What is the format of the model output?"*). To enhance the comparison and reuse of model data, the usage of a data type, that fulfils the FAIR guidelines (Wilkinson et al., 2016) should be preferred. The participant could select between "own binary standard", "own ASCII standard", netCDF or "other". If selecting "other", the participant was able to provide other formats that are not in this list. Multiple-choice answers were allowed.

Binary files are not human-readable and might not be self-documented, which hinders their reuse. The output of the model results in model "own binary" files can lead to the need for certain converters that have to translate the files into a readable format (e.g. netCDF) or it has to be read with an analysing program specifically adjusted to the data format of the model. NetCDF is also a binary data format; it is widely used and the most common data format for atmospheric numerical model data. For this data format, many different analysis programs exist. ASCII files are text files, which are human- and machine-readable, whereas machine-readability is limited to the ability to process formatted text. Using ASCII or own binary files can lead to the problem that data must be converted to be reused or cannot be easily processed by open-source analysis programs. The majority of the participants wrote that the models provide their data in different output file formats, mainly in "own binary", "netCDF" and "own ASCII" file formats (Figure 3.3). Four models use ASCII and binary as output data type, while three also use netCDF as output format. Additional data formats mentioned by the participants were called "Ansys Ensight format"[11] and "Visualisation toolkit" (vtk)[12]. For one model no answer was given to this question. A further aspect of the model properties is the application of conventions in the model data output (Question: *"Are existing conventions applied to the model output you use or created?"*). This question is related to the question *"Are you of aware data format conventions?"*, which is part of the model-use related part (Section 3.3.3.1). Results are summarised in Table 3.4. A statement if existing conventions were applied to the model data happened only three times (twice CF (PALM-4U), once other (Code_Saturne)) while the rest remained unknown. Only PALM-4U applies the existing CF convention to their model output, Code_Saturne uses their own standard for their model data. One participant answered with "No" (ENVI-met). The majority of the participants did not answer the question or answered unknown or no. Note, that the results in Table 3.4 reflects the status as provided by the participants between September 2019 and June 2022. There might be the possibility that some adaptations happened and conventions are used now. E.g. MITRAS is nowadays adapted to a CF-based standard and publishes model data correspondingly (Voss, 2023a). However, based on the existing answers it can be assumed, that standardisation was not common and therefore now is a good time to propose a standard for model data that fulfils a FAIR data publication process.

3.3.3 Model-use related questions

The second part of the survey is about how the participants use the model data, their knowledge on data conventions, data processing tools and their suggestions and advices for the creation of a common data standard for ORM results. In total, there are five questions. In the previous, model-related part

[11]https://www.ansys.com/products/fluids/ansys-ensight
[12]https://vtk.org/

Table 3.4: Convention applied to the model data, answers provided by the participants.

Model	**Applied Conventions**
Ansys Fluent	–
Code_Saturne	Other: MED, CGNS
ENVI-met	No
MISKAM	–
MITRAS	–
PALM-4U	CF or Unknown
PMSS	Unknown

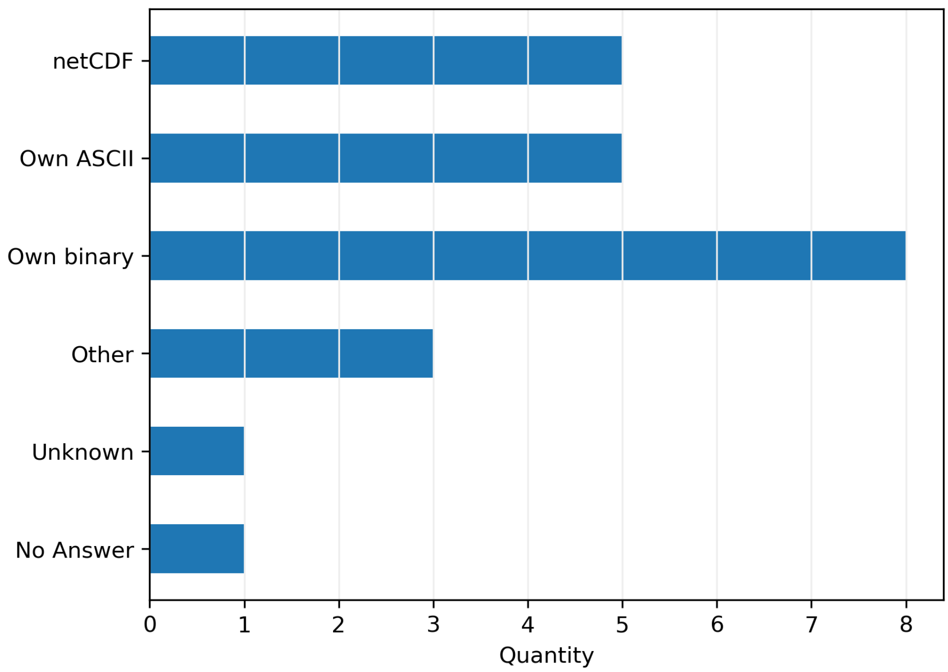

Figure 3.3: Output format of the model results as selected by the participants.

of the survey, we considered only the entries that are related to an ORM, eight in total but providing information for 12 ORMs. In this part of the survey the maximum of answers per question can be 14, because this survey was completed by 14 participants in total, containing three entries, that did not describe an ORM and three entries containing only answers to model-use questions. This is possible because the answers are not directly linked to the model they mentioned as it was in the previous section.

3.3.3.1 Model data conventions Model data conventions do already exist for model data. For netCDF datasets, the Climate and Forecast (CF) Convention (Eaton et al., 2022) is widely used. But there is currently no specific standard for obstacle resolving numerical model data, neither is the CF standard fitted to the needs of ORMs. Therefore, it is important to determine if model users are aware that data conventions do exist in general or if some existing conventions are already applied to their model data. Regarding the question *"Are you aware of data format conventions?"*, from 14 participants, seven of them answered with "yes" and stated which convention they know. Four of them selected the Climate and Forecast (CF) conventions, two submitted "unknown" as convention name. One participant named "CGNS" (CFD General Notation System)[13] and "MED"[14], which is an input format used by Code_Saturne regarding the mesh used by the model.

3.3.3.2 Data processing tools Using standardised data simplifies the analysing, post-processing and usage of graphical tools for the data user. To get an overview of the different post-processing, graphics

[13] https://cgns.github.io/cgns-modern.github.io/index.html
[14] https://docs.salome-platform.org/7/dev/MEDCoupling/index.html

and analysing tools used, we wanted to know, what tools are used by the participants to process the model data. The following tools were available for selection: "CDO", "NCO", "R", "NCL", "Python", "Fortran", "model-own post-processor" and "other", where additional processing tools could be named by the participants. Multiple choice was possible.

"CDO", "NCO" and "NCL" are tools designed to handle netCDF datasets to analyse and plot the data. "Fortran, "Python" and "R" are programming languages. The term "model-own post processor" should include any tools, that are developed for the use with the model that generated the model datasets, excluding for example commercial or open source tools.

Table 3.5 shows the results of the answers to the post-processing tools used by the participants. Based on the numbers and mentions of programmes, which were not in the initial list, it shows that the participants use a variety of tools to process their data. The tools "GIS", "Surfer", "Paraview", "Ensight" and "Matlab" were named by the participants. "Paraview" is an open-source visualisation tool. "Surfer", "Ensight" and "Matlab" are commercial tools and therefore not accessible free of charge. Depending on the GIS tool that is used, it can be an open source or a commercial program, which affects its free of charge accessibility.

A model's own post-processor requires the program together with the model data to be properly used, which can hamper the use of the model data. The usage of different tools to work with the model data may be possible as long as the published model data are at least written in a standardised way by using e.g. appropriate data formats like netCDF. Hereby, a uniform data file type and a standardised way to provide ORM data would simplify and enhance the reuse of model data.

3.3.3.3 Model data user suggestions Regarding on how to handle model data, the participants, based on their experience, could write what they miss when handling model data and what should be considered in the development of a model data standard that is applicable for obstacle resolving model data. The questions were *"What do you miss when handling model data?"* and *"What should we consider for the model standard? Do you have any suggestions?"*. One participant suggested that the model data "should be written in a unified output format" which contains "explained variables and physical units." A PALM-4U user mentioned, that PALM-4U uses a standard based on CF convention and therefore, a standard for ORM should be based on the CF convention as well. The mentioned standard that is used by PALM-4U is described in Scherer et al. (2022). Another participant mentioned that there are problems with handling data with different coordinate systems e.g. coordinates "not following the north-south and east-west direction (rotated grids)." Additionally, an easy-to-use visualisation tool is required. Furthermore, a participant has suggested, that labelling datasets with a "unique identifier" should help to recreate model run processes. Unique identifiers should be applied on files, that are necessary for the model run like input, run-script and output. Also, some of the information can be stored in the model data or be linked to the publication process of the model results. Generally, one important aspect should be that models should be able to provide their results as netCDF data, to ensure easily reusable model

Table 3.5: Data processing tools as selected by the participants.

Tools	Description	Nr.	Further information or website
CDO	Climate Data Operator	6	https://code.mpimet.mpg.de/projects/cdo
NCO	NetCDF Operator	4	https://nco.sourceforge.net/
R	Statistical programming language	6	https://www.r-project.org/
NCL	NCAR Command Language	6	https://www.ncl.ucar.edu/
Python	Programming language	6	https://www.python.org/
Fortran	Programming language	2	Several compilers available
model-own post processor	Model-specific routine to process the model data	5	–
GIS*)	Geo Information System	1	Several tools available
Surfer*)	Visualisation programm	1	https://www.goldensoftware.com/products/surfer/
Paraview*)	Visualisation programm	1	https://www.paraview.org/
Ensight*)	Visualisation programm	1	https://www.ansys.com/products/fluids/ansys-ensight
Matlab*)	Programming language	2	https://www.mathworks.com/products/matlab.html

*) Tools named by the participants.

data. Many visualisation and post-processing routines are fitted to the usage of netCDF and would enhance the reuse of model data. Using the netCDF data format together with the CF conventions can improve the reuse of model data especially for currently non-standardised model data. Therefore, an extension of the CF conventions to the needs of obstacle resolving atmospheric models will be useful. This would improve the publication of FAIR ORM data.

3.4 Discussion and conclusion

The publication of FAIR data is important for the exchange of model results in the atmospheric sciences, but existing standards are not suitable for model data outside global or climate modelling, like obstacle resolving models (ORM). Therefore, and since no standard currently exists for ORMs, the creation of a model data standard for ORMs should be helpful to ensure FAIR model data from ORMs. To get an overview of currently used ORMs and how these models handle and provide their data, an online survey was distributed between 2019 and 2022 in the ORM modelling community.

The survey collected information on how the participants work with ORMs and ORM data with the objective to analyse what similarities or differences exist in technical model features, model data handling and experiences and how to improve the model data handling. The survey should not act as a test to check the knowledge of the user on the right answers to the model they use but rather to get an impression of how they assess their models to work. It is important to note that the survey was not used to find out the exact technical characteristics of the models, as a literature study would be more useful for this. This survey is intended to determine what experiences and knowledge users have gained from using the models and the model data. Their experience plays a decisive role in determining what information should be provided with standardised datasets, as the data standard should ideally be applied in different ORMs. The model user's answers might be incomplete or wrongly interpreting model features. However, they help model developers become more aware of the user's impression of properties, features and aspects of the models. This might give ideas on how to provide and communicate these model aspects. If properly provided with the model data, such information might play an important role in future model comparison studies.

The survey was completed 14 times, containing information for 12 ORMs, that can be summarised into seven different ORMs, which are Ansys Fluent, Code_Saturne, ENVI-met, MISKAM, MITRAS, PALM-4U and PMSS. Based on the answers of the participants, it shows that the named ORMs differ in their features and their model data handling. While some technical aspects (e.g. grid, filtering method, coordinates system, etc.) might be more or less similar between the models, the handling of input and output data varies. The participants named different model data formats, like model-specific binary files, text or "ASCII files" like "name lists" and netCDF files. Several of the model data formats, and especially the model-specific formats, might require specific analysing and plotting tools, which hinders the reusability of model data and prevents ORM data from being FAIR.

The information provided by different participants for models, that were named multiple times, was sometimes partly inconsistent but never contradictory to each other. Due to this, we had the impression that some ORM users might not be completely aware of the mechanics of the model they use. This shows us, that it is even more relevant to provide self-documenting and standardised model outputs, allowing users who are not familiar with the model details to use the data in an error-free (or at least error-reduced) manner.

Overall, the feedback on the survey in the urban meteorology and climate community appears small, resulting in a rather small sample of user experiences. It still provides important information on ORMs typically used in the urban meteorology and climate community and the user experiences of working with these models. As we considered only answers for ORMs in completed surveys, further models named in incomplete surveys were not included in the summary as there were no descriptions of the user experience available. However, many other ORMs exist and have long been used by computational wind engineers. Nevertheless, the answers provided by the survey helped to consider some important aspects of what to consider to standardise ORM data, as the development of a data standard for ORMs is timely, especially since obstacle resolving modelling is still in the development stage and not operationally used by meteorological services (WMO, 2023). Thus, the number of models where output formats may need to be adjusted to follow a standard is currently limited. With Voss (2023a), a publication of standardised ORM dataset was successful. The application to other ORM data revealed necessary extensions that have to be integrated into the data preparation process. Therefore, the application of a data standard to ORM data has to be further explored, adapted and discussed within the community.

4 Towards standardising output datasets using the numerical obstacle-resolving model MITRAS as an example

---

This chapter has been prepared for publication as:

V. Voss , K.H. Schlünzen, D. Grawe and K. S. Samsel (2025): *Towards standardising output datasets using the numerical obstacle-resolving model MITRAS as an example.* In preparation for *Geoscientific Model Development.*

The full manuscript is included in this chapter. Layout and numbering within the manuscript are adopted to fit this thesis. All references are combinde in References. Supporting information, that will be submitted with this manuscript are provided in Appendix C.1 and Appendix C.2. K. Heinke Schlünzen has contributed to the conceptualising. Karolin Samsel contributed to this paper by running simulations for precipitation in Hamburg, providing the datasets for application of the described methods and wrote about the building mask treatment in MITRAS (Section 4.3.1) and the datasets in Section 4.4.3. All authors contributed to the discussion of the results.

Abstract

The publication of well-described FAIR datasets is an important part of atmospheric modelling and research. Data standards ensure that datasets are delivered in a consistent way that is easy to understand for a data user. Standards define how the data is described, i.e. which variable names, descriptions and data formats are used. However, existing model data standards such as the CF conventions are mainly adapted for global or regional scale models. For atmospheric micro-scale obstacle-resolving (urban) models (ORMs), there is no discipline-specific model data standard and the existing ones are not fully suitable to adequately describe ORM datasets. To overcome the lack of standardisation processes, the ATMODAT Standard has been developed to promote the publication of FAIR datasets when no discipline-specific standard is available. This paper describes the process of producing standardised model results. The processing for ORM MITRAS serve as an example to show possible ways for the publication of FAIR datasets. The adaptation of the model's post-processing routine *m2cdf* and the development of a new post-processing routine called nc2atmoadat are shown. The last may be applicable by other ORM modellers, its limitations, challenges and further use cases are discussed. Application of the two post-processors allows the preparation of datasets according to the requirements of the CF convention and the ATMODAT Standard. The first standardised MITRAS datasets are successfully processed and published.

4.1 Introduction

Numerical modelling is a commonly used tool to investigate complex physical processes in the atmosphere. Models can simulate processes on different scales, ranging from the global scale (weather forecasts or climate simulations) via regional scales (investigating regional effects) to the urban and neighbourhood scale, where processes in urban areas are analysed. An important aspect of atmospheric modelling is the publication and exchange of model results. Especially in the climate modelling community, model datasets have been exchanged for decades, e.g. with the Coupled Model Intercomparison Project (CMIP) (Eyring et al., 2016; Touzé-Peiffer et al., 2020) as a prominent example. The need for standardised model output data was established. The awareness and amount of publication of research data are increasing, and storing such data is becoming more and more important (Bermudez, 2017; De Vos et al., 2020; Eggleton and Winfield, 2020). Ideally, published model datasets should fulfil the FAIR principles (Wilkinson et al., 2016), meaning that the datasets should be "findable, accessible, interoperable and reusable". FAIR data ensures that the handling of model datasets is convenient and easy for data users. One way to achieve and ensure FAIR datasets is to apply data standards to datasets, as they provide structure and guidelines for the production of well-described and FAIR-compliant datasets.

In atmospheric and climate science, the *network common data form (netCDF)*[15] (Rew and Davis, 1990) is a commonly used data format and well established within the community. It is a self-describing data format for scientific data that allows a multi-dimensional structure of the data (Rew and Davis, 1990). It works well for storing time and space dependent data. NetCDF is often used alongside the Climate and Forecast (CF) conventions, which provide guidelines for the provision of (meta)data information in the netCDF data structure (Eaton et al., 2024). These conventions ensure that NetCDF files are well-described, and easily understood and usable by data users. An important feature of the CF conventions is the provision and usage of a standardised vocabulary for variable names[16]. The netCDF data format, together with the CF conventions, is widely used and frequently adapted by global and regional scale models (e.g. models within CMIP), as well as in other applications that produce large and or multi-dimensional datasets.

However, the publication of FAIR datasets is not trivial in research disciplines, where no data standardisation processes are established. Such a discipline is the high-resolution urban climate modelling, carried out with obstacle-resolving models (ORMs). The existing standards, such as the CF conventions, are not sufficient to meet the needs of these model types, since they currently do not cover discipline-specific variables and features, whereas no overarching discipline-specific standardisation is established. This hinders the publication of well-described model datasets in this discipline and impairs the reusability of such model datasets.

ORMs are obstacle-resolving atmospheric micro-scale models, focusing on processes in small domains, like urban neighbourhoods and city quarters. They use grid sizes of metres and provide results in a

[15] https://www.unidata.ucar.edu/software/netcdf/

[16] https://cfconventions.org/Data/cf-standard-names/current/build/cf-standard-name-table.html

high spatial and temporal resolution (Oke et al., 2017; Baklanov et al., 2018). The model domains are typically not much larger than a few km^2, and obstacles, like buildings or trees, are resolved. ORMs can be either numerical models or physical models, e.g. wind tunnels or water tanks (Oke et al., 2017).

Each ORM prepares and provides data in a model-specific way. Many of them use their own model-specific data formats (such as binary or text files), whereas some ORMs also provide their model data as netCDF files (Jänicke et al., 2021; Voss et al., 2024). Models such as MISKAM (Eichhorn and Kniffka, 2010; Eichhorn, 2011), OpenFoam (OpenFOAM, 2025), Ansys Fluent (Poole et al., 1995), Envi-met (Bruse and Fleer, 1998) or MITRAS (Schlünzen et al., 2003; Salim et al., 2018) provide their data mainly in their model-specific way, but also allow a conversion to netCDF with tools provided by the model developer. The ORM PALM-4U uses netCDF and applies a model-specific data standard to the netCDF files that follows the CF conventions (Scherer et al., 2019, 2022). For MITRAS, not all variables were initially adapted to the CF conventions. Therefore, the use of published data is difficult, as identifying variables without the use of standardised variable names is only possible when using the technical documentation (e.g. Schlünzen et al. (2018b) or Scherer et al. (2022)). In general, if a model provides data in their model-specific formats, the use or access of the data might be hindered and complicated, especially if special tools or programs to access or convert the data are required. Due to the lack of general standards for ORM results, the extent to which ORM datasets are FAIR varies. As ORMs continue to evolve and become more widely used, the need for standardised datasets and a common vocabulary for ORM variables will increase. In obstacle-resolving modelling, the reusability and inter-comparison of model data will become an upcoming and important aspect in this discipline (WMO, 2023). With urban areas now specifically addressed in IPCC [17], it seems even more important that ORM datasets are published FAIR and become reusable.

To overcome the lack of standardisation, e.g. in the ORM community, and to guide the creation of well-described and FAIR model datasets, the Atmospheric Model Data Standard (ATMODAT Standard) (Ganske et al., 2021) was developed. This standard is intended to be applied to model datasets from different atmospheric modelling disciplines, especially for those that do not have a discipline-specific standard. First datasets prepared according to the ATMODAT Standard were already published. The ATMODAT Standard was applied to datasets containing reanalysis data of observations to classify weather types over the Baltic and North Sea (Loewe, 2022; Loewe and Schade, 2024), to datasets for sensitivity studies of aerosol and cloud microphysics processes simulated with the ECHAM6-HAM2 model (Mülmenstädt, 2022), to datasets from urban atmospheric MITRAS simulations focusing on dispersions of particles (Voss, 2023a) and precipitation (Ferner et al., 2024; Samsel et al., 2025) and datasets of the urban climate in Hamburg for winter and summer generated with the mesoscale model METRAS (Boettcher et al., 2024a,b).

[17](https://www.ipcc.ch/report/special-report-on-climate-change-and-cities/; last acccess 17.07.2025)

In this paper, the challenges in creating a FAIR ORM data publication that fulfils the ATMODAT Standard, are discussed. For the description of a successful path to such a publication, the ORM MITRAS will serve as an example. As a first step, the application of the CF convention on MITRAS datasets are investigated. Therefore, CF standard names and attributes need to be retrieved and assigned to the model specific variables, hereby revealing issues with ORM specific variables (Section 4.2). Modifications of the post-processing of the ORM MITRAS are described, and details of the necessary steps for applying the ATMODAT Standard to ORM data are provided (Section 4.3). The application of the new standardisation processing is applied to datasets from the ORM MITRAS serves as an example (Section 4.4).

4.2 Determination of ORM variables

The CF conventions provide a collection of standardised variable names (CF Standard Name Table[18]), the so-called standard names, which make the meaning of a netCDF variable self-explanatory. Without a standardised vocabulary, each modelling group would apply their model-specific and individual naming conventions to the variables. Data user unfamiliar with the model might be unsure if those self-named variables describe the same properties they assume and that they want to use. Standardised variable names are already defined in the CF conventions for many common atmospheric variables such as wind speed, temperature, humidity, etc. Hereby, a "`standard_name`" should only be assigned to a variable if the model variable fits the definition of the corresponding entry in the CF Standard Name Table. If no matching standard name is available, a variable should be assigned with a "`long_name`" that contains a descriptive variable name to indicate the purpose of the variable (Eaton et al., 2024). This is useful if (discipline-specific) variables are not represented by existing standard names.

Many ORM-specific properties are currently neither standardised within the community nor available in the CF conventions. Such non-standardised ORM variables are e.g. related to the obstacle and its surface, such as wall temperatures, radiation fluxes from surfaces, the amount of rainwater on roofs or walls, thermal indices or obstacles themselves, if they are provided as an own variable.

To illustrate what is required to apply the CF conventions and the ATMODAT Standard, the output of ORM MITRAS is used. As a first step, all variables available in the model code that are used for output, in the post-processor code and the model documentation (Schlünzen et al., 2018b) were collected and summarised. Based on this collection, it became visible that variable names in the codes are named and defined differently, and needed to be more self-explanatory and required adjustments.

To align the variable names with those used in other ORMs and to gain an understanding of the ORM-specific variables typically used, the variable names from PALM-4U (Maronga et al., 2019, 2020) were considered. Hereby, PALM-4U was selected for comparison because it is a state-of-the-art ORM that uses netCDF as output data format, uses standard names for its variable descriptions and follows the CF conventions to describe the datasets (Scherer et al., 2019, 2022).

[18]https://cfconventions.org/Data/cf-standard-names/current/build/cf-standard-name-table.html

The comparison of variables in the ORMs MITRAS and PALM-4U was based on matching the provided "`long names`", units and descriptions in the respective model documentations. In addition, the MITRAS variables were checked against the CF Standard Name Table[19], and fitting "`standard names`" were added to the MITRAS model variables. Furthermore, typical ORM variables which are not covered by the existing standard names are identified.

Based on the comparison of variable long names between MITRAS and PALM-4U, it became clear that the mapping of variables to each other is not trivial. It is not always clear whether similar model variable names represent the same properties, which would only be conclusive if the respective model routines were checked. The meaning was predominantly unclear for variables with slightly differently constructed variable names.

The assignment of CF standard names to MITRAS variables was also not always trivial, as some aspects of the MITRAS variables do not always fully match the definitions of existing standard names. Here, it was necessary to consider whether the MITRAS variable in question needs to be adjusted in favour of the existing CF definition or whether to define a new standard name that takes the missing aspects into account. For example, MITRAS provides the variable 'water_temperature', which describes the temperature for all water bodies in the model domain. In urban modelling, it can be assumed that freshwater is more common, except if coastal cities are investigated. In the current CF conventions, only standard names for saline or seawater are available. Therefore, freshwater properties need to be introduced or the handling of this property needs to be optimised.

Without clearly defined standard names and definitions for such variables, using ORM datasets becomes difficult, as the meaning of variables might differ from model to model. Therefore, ORM variables need to be identified across different models. Standard names with corresponding definitions and units should be determined and then proposed as new standard name candidates to the CF conventions.

4.3 Adjustments to the post-processing for MITRAS

Data provided in proprietary data formats, such as model-specific output formats, may not be accessible to external users. It is therefore advisable to convert the data into more commonly used data formats. MITRAS stores simulation results in model-specific binary data files, which ensures a fast writing of the output during the run time while integrating the model (Example in Figure 4.1, MITRAS (green), direct output is binary).

The direct binary model output is transferred into netCDF with the model post-processor program *m2cdf* (Figure 4.1, step 1; details in Section 4.3.1). However, not all necessary information and metadata may be included at this point, so further information is needed to describe the data adequately. A second step is introduced, where a newly developed post-processor called *nc2atmodat* is used (Figure 4.1, step 2).

[19] https://cfconventions.org/Data/cf-standard-names/current/build/cf-standard-name-table.html

Figure 4.1: Data post-processing workflow for the FAIR publication of MITRAS model results.

This program adjusts the netCDF model data from step 1 to make it FAIR and to fulfil the requirements of the ATMODAT STANDARD.

nc2atmodat adds or corrects information missing in the netCDF datasets from step 1, adds georeferencing to the MITRAS data, allows to select the data to be published domain-wise and variable-wise and adds global metadata that are required by the ATMODAT STANDARD (details in Section 4.3.2). In step 3 these processed datasets are then checked for the fulfilment of the ATMODAT STANDARD, using the ATMODAT STANDARD CHECKER (Figure 4.1, details in Section 4.3.3). They are then ready for being published as a FAIR dataset.

4.3.1 Step 1: Adaptation of the model post-processor

To create accessible model results, the binary model output data of MITRAS is transferred to netCDF, using the model-specific post-processor *m2cdf*. The program adds descriptions and attributes to the variables as well as generic global metadata to the netCDF dataset. Based on a review of MITRAS variables (Section 4.2), it became visible that variable names and corresponding attributes needed to be added in the MITRAS post-processor *m2cdf* to become CF compliant. The post-processor *m2cdf* received an update to create CF-compliant netCDF datasets, for which the major part of the update involved the addition of variable attributes with corresponding entries following the CF conventions.

As an example, Figure 4.2 shows a comparison of the header from two MITRAS netCDF files for the variable wind speed (ff). In the original version before the update of the post-processor (left) no "`long_name`" and "`standard_name`" attributes were assigned. Instead, the variable was described by an attribute "`name`". In the update, the attribute "`name`" was therefore renamed to "`long_name`", and for each MITRAS variable a self-describing long name was then either taken from the previous entry in the attribute "`name`" or a new self-describing long name was defined. The attribute "`standard_name`" was introduced as a new attribute to *m2cdf* and, if applicable, for each variable the corresponding standard name from the CF convention was added.

According to the CF convention, the characteristics of a grid cell can be provided with the attribute "`cell_methods`" (Eaton et al., 2024). The attribute "`cell_methods`" replaces the former used attributes "`type_t`", "`type_x`", "`type_y`", and "`type_z`" (Figure 4.2) that were previously used to provide this information.

```
float ff(time, k, j, i) ;
      ff:name = "horizontal_wind_speed" ;
      ff:structure = "2301" ;
      ff:units = "m s-1" ;
      ff:coordinates = "lon lat" ;
      ff:type_t = "inst" ;
      ff:type_z = "grid_cell" ;
      ff:type_y = "grid_cell" ;
      ff:type_x = "grid_cell" ;
      ff:_FillValue = -999.f ;
```

```
float ff(time, k, j, i) ;
      ff:long_name = "horizontal wind speed" ;
      ff:standard_name = "wind_speed" ;
      ff:structure = "2301" ;
      ff:units = "m s-1" ;
      ff:coordinates = "lon lat" ;
      ff:cell_methods = "time: point k: j: i: mean" ;
      ff:grid_mapping = "crs" ;
      ff:_FillValue = -999.f ;
```

Figure 4.2: Example for the netCDF header of MITRAS model results. Left: Layout of the header before the adaptations in the post-processor *m2cdf*. Right: Header with the updated version of *m2cdf* consistent with CF conventions.

Besides the adaptation to the CF conventions, *m2cdf* received an update regarding the provision of information on obstacles (e.g. buildings) and obstacle-related variables. Obstacles are a key element and important feature of ORMs as they affect the flow within the model domain. As an ORM, MITRAS deals with explicitly resolved obstacles in the model domain. Previously, in MITRAS simulation results, the raw output did not highlight the building locations but showed the variable values continuously (Figure 4.3a), whereas the location of buildings was written as a separate variable. *m2cdf* sets a Fillvalue (-999.f) to the cells matching with the building positions to ensure that exact building locations are visible directly in the variable (Figure 4.3b). As the building mask is now included in all variables and, if needed, can be indirectly derived, it is no longer necessary to explicitly provide it. This makes it possible to reduce the size of the output. In the case that each output variable should be provided in an individual file, the building information is now guaranteed to always be provided.

Further changes in *m2cdf* concern the representation of building surface-related variables. Building surface-related variables such as the building wall surface temperature or amount of rain on roofs are given in an index-variable structure as common within the code to safe storage space for the model. However, these arrays are not intuitively understandable for users unfamiliar with MITRAS. MITRAS model equations are solved on an Arakawa-C grid (Arakawa and Lamb, 1977; Schlünzen et al., 2018b). On this grid, scalar quantities like the temperature are defined at grid cell centres, while vector variables, like the wind components, are located at grid cell boundaries. To represent a vector variable at the grid cell centre or a scalar variable at the grid cell boundary, it must be interpolated. For example, in Figure 4.4, a vertical cross-section of the building mask is shown.

Values of atmospheric variables are set to zero inside buildings (grey cells) during the calculations. As MITRAS explicitly resolves obstacles in the model domain, walls are treated as unpermeable obstacles for flow, with wall functions used at the rigid boundaries. Variables assigned to the building surfaces,

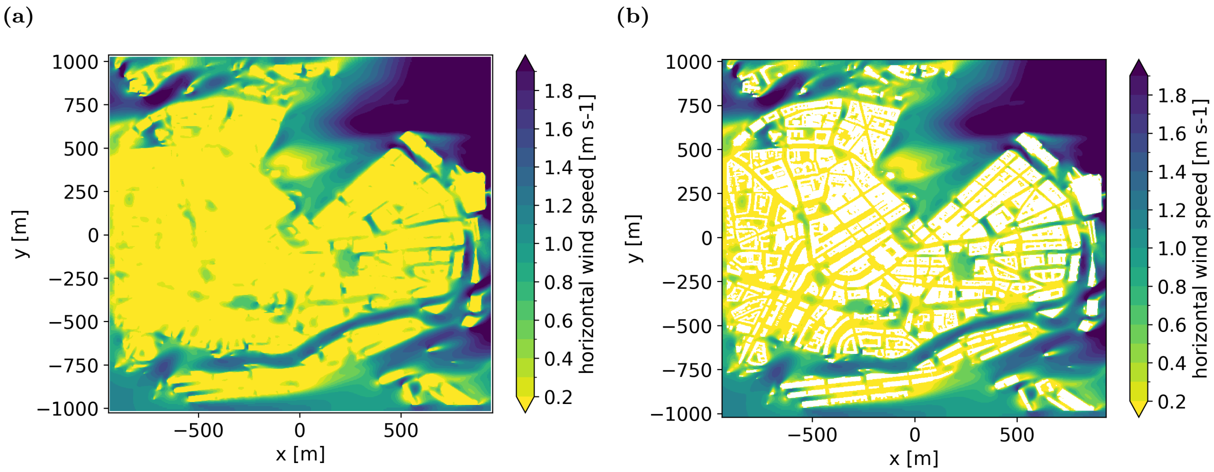

Figure 4.3: Horizontal wind speed in 2.5 m height above ground in an urban area (here: Depicting the city centre of Hamburg, Germany) without (a) and with marking the buildings with missing values (b).

e.g. wall temperatures, can be assigned or calculated solving the surface energy budget equation. The model used wall temperatures are provided in the output for further use.

The spatial dimension of a building surface-related variable is represented in an array structure, as a $N_{surfcells} \times 3$- matrix with $N_{surfcells}$ denoting the number of obstacle-adjacent grid cells. These arrays contain, based on an auxiliary grid, the grid cell index, at which a cell with an adjacent building wall is located. Further array variables provide the orientation of the building surface (Schlünzen et al., 2018a,b), representing the direction of the obstacle face (1: east/west, 2: north/south, 3:top/bottom). The structure results from how the building surface-related variable are calculated within the model code (Appendix C.2); as it improves the computational efficiency. Without further knowledge of these model properties, the grid information is not trivial to derive and therefore not automatically accessible to a data user. To better access those variables, it is advisable to represent building surface-related variables in a Cartesian grid similar to atmospheric variables like temperature or wind speed.

Building surfaces are located at the grid cell boundaries (black dots in Figure 4.4). It is preferred to provide variables on the grid cell centres, but unlike for the wind speed components. In contrast, the date would give a wrong impression of the positioning of the building boundaries. Furthermore, it is not feasible to interpolate the building surface-related variables from the grid cell centre to the boundary or vice versa. It is preferable to provide the data at the positions of the building boundaries (walls or roof), as these are located at the actual boundaries of the grid cells. Therefore, the 3D grid positioned at the cell boundaries of the model grid was introduced. This grid uses the already established vector coordinate indices iv, jv and kv. A similar approach for surface-related variables by introducing a secondary coordinate system, that describes the location of the wall properties in the model domain, is utilised by the model PALM-4U (Scherer et al., 2022).

To keep the orientation of each surface-related variable, the variables are allocated to the coordinates of the corresponding direction (east, west, north, south, top and bottom), leading to six building surface-related variables for each direction as listed for the building surface temperature (tbuisurf) in Table 4.1. An example of the application of this feature is described in Section 4.4.3.

Completing this post-processing step makes the netCDF datasets of MITRAS better described, understandable, user-friendly and amenable to open-access programs (e.g. ncview[20], panoply[21]). The drawback is a more than 6 times larger output for all surface-related variables.

4.3.2 Step 2: Development of nc2atmodat to create FAIR netCDF files for ORMs

The post-processor *m2cdf* transfers the model output to a netCDF file and adds variable names and descriptions consistent with the CF conventions to the netCDF file. Also, it provides very generic global metadata to the datasets. However, the data is not fully adapted to the AtMoDat Standard and still miss some additional features relevant to the understanding of the variables.

To fulfil the ATMODAT Standard, to modify, correct or add information to the netCDF file, and add the georeferences, the program *nc2atmodat* was developed. *nc2atmodat* provides a quick and convenient way for MITRAS data producers to add additional information, modifications or corrections to the dataset without altering the hardcoded entries in the model core or the post-processor. A re-translation of the model results into netCDF, which can be rather time-consuming, is therefore not needed. Especially, generating properly georeferenced MITRAS model datasets was added and enabled by this program. Adding global attributes required for fulfilling the ATMODAT Standard is easier compared to the provision with the existing structures in *m2cdf*. Last but not least, program *nc2atmodat* allows to select the data to be published, since in general not all data are meant to be published, but only some variables or only a part of the model domain.

[20] https://cirrus.ucsd.edu/ncview/
[21] https://www.giss.nasa.gov/tools/panoply/

Table 4.1: List of variables derived from the obstacle surface temperature.

Variable name	Description
tbuisurf_e(z,y,xv)	Obstacle surface temperature east wall
tbuisurf_w(z,y,xv)	Obstacle surface temperature west wall
tbuisurf_n(z,yv,x)	Obstacle surface temperature north wall
tbuisurf_s(z,yv,x)	Obstacle surface temperature south wall
tbuisurf_t(zv,y,x)	Obstacle surface temperature top wall (roof)
tbuisurf_b(zv,y,x)	Obstacle surface temperature bottom wall (below bridge)
tbuisurf_p(y,x)	Obstacle surface temperature on all roofs (plane view)

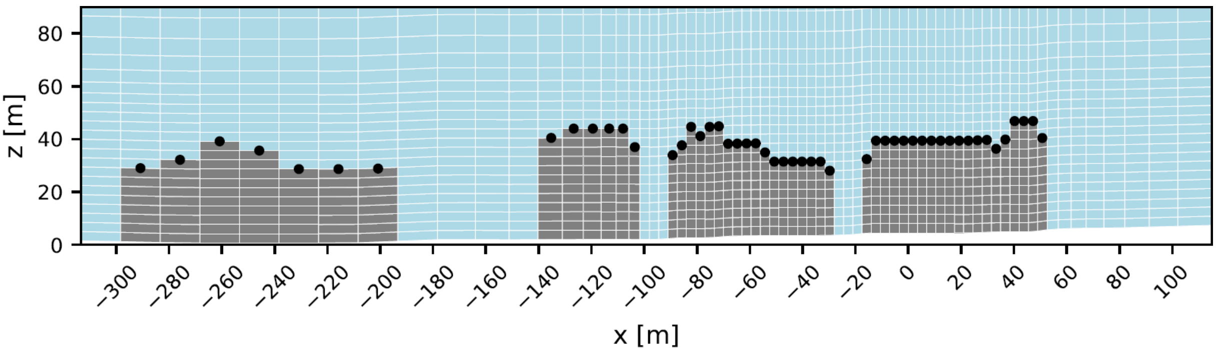

Figure 4.4: Vertical cross section of building mask at $y = -20\,\mathrm{m}$ (dashed line in Figure 4.8) with buildings in grey and atmospheric grid cells in blue. Black dots denote the location of surfaces (roofs).

Furthermore, since the program modifies netCDF files, it might be usable by other ORM modellers with a few adaptations towards their model properties. The code of *nc2atmodat* is written in Python[22] and uses the modules Xarray (Hoyer and Hamman, 2017), Pandas (pandas development team, 2020), Numpy Harris et al. (2020) and Pyproj (Whitaker et al., 2019). The program is open-source and available via GitHub (Voss, 2025a). Figure 4.5 shows the workflow of *nc2atmodat*, whereas the main functions of *nc2atmodat* are described in the following section.

4.3.2.1 User Input *nc2atmodat* includes a User Input section, where the user provides the path and name to the netCDF file and the AtMoDat metadata table. This metadata table is provided with the program and contains the global attributes entries for the final resulting datasets. The information within the metadata table has to be filled in by the program user. The User Input section provides three control options that can be activated: spatially and temporal cropping the dataset, defining a custom "georeference point" and selecting output variables.

4.3.2.2 Cropping data *nc2atmodat* crops the datasets in the temporal and spatial dimensions if requested by the user (section 4.3.2.3). Otherwise, in the spatial dimension, the outermost grid cells are removed, which is required to assign the correct grid cell bounds to each grid cell in the MITRAS case.

4.3.2.3 Spatial and temporal dimensions Spatial coordinates with corresponding attributes and information are assigned. The current version of *nc2atmodat* requires that all variables are located on a scalar grid, described by indices i and j in horizontal and k in vertical direction. An exception are building surface-related variables defined at grid cell boundaries (Section 4.3.1), which use indices iv, jv, kv, that are initially used to describe vector coordinates.

[22]https://www.python.org/

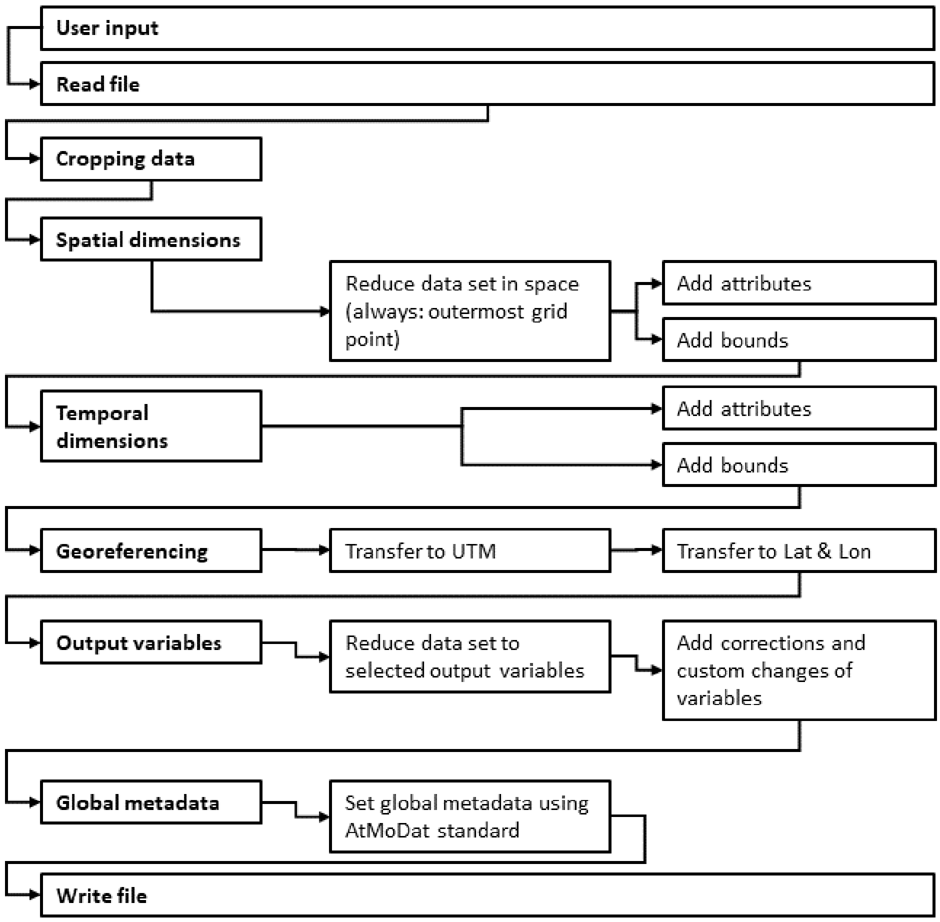

Figure 4.5: Overview of the workflow of the program nc2atmodat.

MITRAS uses a Cartesian coordinate system in the horizontal direction (Schlünzen et al., 2018b). The use of geographic coordinates is not common for ORMs, since with grid sizes of 1 to 10 m, the grid sizes in geographical coordinates will differ by only $0.03''$ to $0.3''$. While the addition of a georeferenced grid is not straightforward, it is essential for using the data in e.g. graphical systems used in meteorology (e.g. GIS applications). For further processing reasons, *nc2atmodat* introduces the dimensions x, y, z (xv, yv, zv; for surface properties), given in metres. If applicable, the indices are replaced by the newly introduced coordinates x, y, z, xv, yv and zv in the final AtMoDat conformal netCDF model results, otherwise, they remain as indices. The temporal dimension coordinate (MITRAS variable name: "time") is set and assigned with variable attributes.

For spatial and temporal coordinate variables, the attribute "`bounds`" was introduced. Bounds are used to indicate the extent of a grid cell and provide the location of the cell boundary (Eaton et al., 2024). After cropping, the spatial and temporal variables are then assigned attributes and new variable names.

4.3.2.4 Georeferencing This is a feature that is mainly implemented for the modification of MITRAS model datasets but adjustable for other ORMs that use similar projections. MITRAS provides the coordinates as distances of each grid point to the origin of the model domain provided as latitude / longitude information. To be consistent with the CF conventions and to publish georeferenced datasets, geographical coordinates have to be provided with the dataset. *nc2atmodat* calculates and adds geographic coordinates to the dataset using the projection used for the creating the grid in Cartesian coordinates. For MITRAS simulations, the model-specific pre-processor generates the model domain using Geographic coordinates and metre coordinates (specifially: Gauss-Kruger coordinates). Geographical coordinates, provided in latitude and longitude are required to calculate the solar radiation, whereas metre coordinates are used to select and construct the model domain from a set of raster-grid files that contain information on building heights, land surface cover classes and terrain heights. Currently, only the geographic coordinates, further named as "reference point coordinates", are provided with the model domain input and output files and used in the model simulations. In both files, metre coordinates are available for each grid point.

With the Python module Pyproj (Whitaker et al., 2019) and the reference point coordinates, geographic coordinates can be calculated for each grid point and are assigned to the dataset. Thus, netCDF files processed with *nc2atmodat* provide two sets of coordinates, Universal Transverse Mercator (UTM) coordinates (*x_utm*, *y_utm*) and geographical coordinates (named as *lat* and *lon*). The reference point coordinates can be either taken from the model results or provided via the User Input. With these adaptations, the dataset is then properly georeferenced.

4.3.2.5 Output variables Most likely not all variables in the model output are analysed in depth and relevant for the publication. This function of the program *nc2atmodat* allows to reduce the dataset to those variables that were picked for publication. These output variables can be either pre-defined via an additional file ("dict_OUTPUT_variables"), provided with the program or submitted via the User Input.

ORM post-processors *m2cdf* sufficiently describe the majority of variables by providing hardcoded variable attributes. The second post-processor *nc2atmodat* provides the possibility to modify or add attributes, if necessary. To give an example, MITRAS might be used to calculate the transport and concentration of air pollutants as passive tracers. However, the tracer variable in the model is not specified as a certain air pollutant in the output data, thus *m2cdf* adds a generic long name and no standard name to this variable. The CF convention provides a large list of standard names for different air pollutants, that could be applied to the variable. To retain the flexible usage of the variable in the models but to be able to add standard names to this variable, fitting long and standard names can then be added via *nc2atmodat*. Furthermore, any additional attribute can be added using *nc2atmodat*. For example, if no standard name was available, but additional information or further explanations were helpful to clarify the meaning or purpose of a variable, a "`comment`" attribute can be added, making the variable sufficiently self-explanatory. Such additional attributes and corrections can be stored in the input file ("dict_custom_variable_attributes").

4.3.2.6 Global metadata The ATMODAT Standard requires that the dataset has to be sufficiently described and contain as much information as needed to be FAIR. Therefore, the dataset shall contain the global metadata attributes summarised and explained in the ATMODAT Standard (Ganske et al., 2021). The global metadata attributes are categorised as mandatory, recommended and optional. To fulfil the requirements for the ATMODAT Standard, the data producer shall fill attributes and describe their datasets with as much detail as possible. As sometimes it might not be possible to add all attributes for each dataset, at least the mandatory attributes must be provided. To guide and help the data producer to provide global metadata to their dataset to fulfil the ATMODAT Standard, an Excel file ("Metadata_for_atmodat_standard.xlsx") is provided with the program *nc2atmodat*. With the *nc2atmodat* the global metadata can then be added to the dataset. Explanations, examples of how to fill out the form and further information for each attribute are given within the document.

4.3.2.7 Write output file Lastly, the data are saved as a new netCDF file. This is then ready for checking if it fulfils the ATMODAT standard.

4.3.3 Step 3: Application of ATMODAT Standard Checker

With the publication of the ATMODAT Standard (Ganske et al., 2021), the ATMODAT Standard Checker (Kretzschmar et al., 2022; Ganske et al., 2022) was released. It is a tool for checking if a dataset fulfils the requirements and its compliance with the ATMODAT Standard. If required, the ATMODAT Standard Checker also provides the possibility to add or correct metadata information to the dataset header. The output of the ATMODAT Standard Checker is machine-readable, which makes it possible to evaluate multiple files at a time automatically. Instructions for installing and using the checker are available on GitHub (Kretzschmar et al., 2022). The checker helps any person interested in publishing model results to determine missing information (Section 4.4.2 and Figure 4.7).

4.4 Application of suggested standardisation to ORM datasets

In this Section, the post-processing described in Section 4.3 is applied to two datasets from the ORM MITRAS: one dataset includes dispersion (Voss, 2023a), hereafter named V23, and the second dataset considers precipitation (Ferner et al., 2024) further denoted as F24. Both model outputs were converted to netCDF by using *m2cdf* (Section 4.3.1), further post-processed using *nc2atmodat*(Section 4.3.2) and are already successfully published in the World Data Center for Climate (WDCC). Both datasets are based on the same domain (Section 4.4.1). Hereby, in Section 4.4.2, the preparation of a georeferenced MITRAS dataset is shown with V23. In Section 4.4.3 the standardisation of model data with ORM-specific obstacle variables is shown using F24.

4.4.1 Model domain

Both datasets are generated by the ORM MITRAS (Schlünzen et al., 2003; Salim et al., 2018), which is part of the M-SYS model system (Trukenmüller et al., 2004; Schatzmann et al., 2006). The building data for both datasets are taken from Salim et al. (2015). They are based on the Digital Terrain Model, the data of the German geoinformation system ATKIS (official topographic-cartographic information system) (Müller and Seyfert, 2000), and on the 3D-urban model data (LoD 2) for building details. Salim et al. (2015) used a model domain representing the City Centre of Hamburg, Germany. The status of the City is from 2014, with the Hamburg City Hall (Rathaus) in the centre of the domain.

4.4.2 Dataset with dispersion

The dataset V23 depicts the domain around Hamburg City Hall as described in Section 4.4.1 without orography. It contains one simulation with constant temperature and humidity and with stable atmospheric conditions. The domain extents in the horizontal direction to $1,9 \times 2,0\,\mathrm{km}^2$ and reaches the top of the model domain at a height of $5,4\,\mathrm{km}$. It has a horizontal resolution of 2.5 m over the whole model domain. The vertical resolution begins with 5 m within the lowest model levels (positioned at 2.5 m above ground) and increases above the domain height of 152.5 m to 200 m. The dataset was mainly created to test and apply the ATMODAT Standard to an ORM dataset; therefore, no further complexity was considered in the simulation.

This model simulation includes the emission and transport of passive tracers from areas with a specific surface cover class (small green spaces within the city). These tracers were assumed to be air pollutants, with no chemical reactions, deposition or sedimentation processes occurring. V23 provides a variable

named "conc", for which no defined long or standard name is assigned in *m2cdf*, as the variable can be denoted to different air pollutants of the same qualities. In this particular dataset, the variable "conc" should be described as particulate matter with particle sizes of 10 μm (PM10). For this variable the CF standard name "mass_concentration_of_pm10_ambient_aerosol_particles_in_air" is selected and the corresponding long name is defined as "concentration of pm10". These attributes and the corresponding unit are then added to the variable with the program *nc2atmodat*.
Proper geographical coordinates had to be added to the dataset. Geographical coordinates can be calculated from the existing grid coordinates, as they are provided as distances from the model origin in metres (Section 4.3.2). In this dataset, the model origin of the coordinate system is located in the centre of the model domain at the Hamburg Rathaus (Figure 4.6). Therefore, the geographical coordinates for the model origin ($x = 0, y = 0$), the so-called reference point, are required. The reference point coordinates are mainly used during the model domain initialisation and were initially not provided with the model output. *m2cdf* includes the reference point coordinates in the model output. *nc2atmodat* then calculates the geographical coordinates from the reference point and adds them to the dataset. With these features added to both post-processing routines, the dataset can now be assigned geographical coordinates and is correctly georeferenced. This is visible in Figure 4.6, where the building positions in the model are given in latitude/longitude coordinates and align with the building positions on the underlying map.

After applying *m2cdf* and *nc2atmodat* on the model dataset, the ATMODAT Standard Checker (Section 4.3.3) was used. Figure 4.7 shows an excerpt of the ATMODAT Standard Checker output of the same dataset before and after the post-processing with *nc2atmodat*. The left-hand side shows the checker results of the dataset after the use of *m2cdf* but without the use of *nc2atmodat*. The ATMODAT Standard Checker raised several warnings, as for example, the variable "conc01" was previously missing the "`standard name`" or a "`long name`" attribute. The right-hand side of Figure 4.7 shows the checker output for the dataset, after processing with the updated version of *m2cdf* and with *nc2atmodat*. After V23 passed the ATMODAT Standard Checker, the dataset is thereby considered as fulfilling the ATMODAT Standard and thereby standardised.

4.4.3 Dataset with precipitation

F24 (Ferner et al., 2024) was created for studies on the heterogeneity of rain in an urban neighbourhood (Ferner et al., 2023). It contains precipitation quantities defined on building surfaces providing a suitable test case for the representation of variables specific to ORMs. The example given here can be applied to other building surface-related variables or results created by other ORMs.

A stretched grid is used with a highly resolved area with grid sizes of 3.5 m near Hamburg City Hall (white box in Figure 4.8). The grid sizes increase by a factor of 1.175 resulting in resolutions of 15 m horizontally and 150 m vertically at the domain boundaries. The domain extends to 1.6 $\times$ 1.8 km^2 around Hamburg City Hall reaching up to 5 km. A decrease in grid resolution was necessary to perform computationally

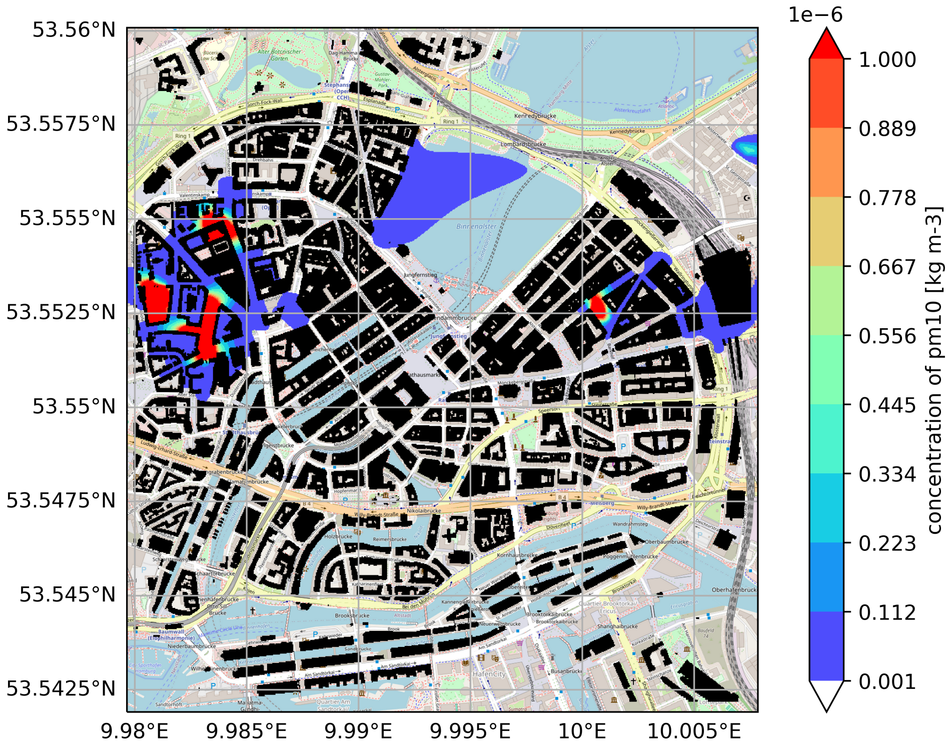

Figure 4.6: Model domain V23 of the city centre of Hamburg, Germany, with buildings in black, use of horizontally uniform grid. Coloured areas denote concentration of PM10. The underlying map was taken from openstreetmap.org. Plot was generated with Cartopy (Hrisko, 2020; Met Office, 2010 - 2015).

```
CHECKING NetCDF FILE:
=====================
WARN: Cannot determine CF version from the Conventions
attribute; checking against latest CF version: CF-1.8
Using CF Checker Version 4.1.0
Checking against CF Version CF-1.8
Using Standard Name Table Version 82 (2023-07-06T13:17:07Z)
Using Area Type Table Version 11 (06 July 2023)
Using Standardized Region Name Table Version 4 (18 December
2018)

WARN: (2.6.1): No 'Conventions' attribute present

------------------
Checking variable: conc01
------------------
WARN: (3): No standard_name or long_name attribute specified
INFO: (3.1): No units attribute set.  Please consider adding
units attribute for completeness.

------------------
Checking variable: tket
------------------
WARN: (3): No standard_name or long_name attribute specified
ERROR: (3.1): Invalid units: ?

------------------
Checking variable: time
------------------
WARN: (3): No standard_name or long_name attribute specified
WARN: (4.4.1): Use of the calendar and/or month_lengths
attributes is recommended for time coordinate variables
```

```
CHECKING NetCDF FILE:
=====================


Using CF Checker Version 4.1.0
Checking against CF Version CF-1.8
Using Standard Name Table Version 82 (2023-07-06T13:17:07Z)
Using Area Type Table Version 11 (06 July 2023)
Using Standardized Region Name Table Version 4 (18 December
2018)



------------------
Checking variable: conc01
------------------



------------------
Checking variable: tket
------------------


------------------
Checking variable: time
------------------
```

```
=== Short summary ===

ATMODAT Standard Compliance Checker Version: 1.3.2
Checking against: ATMODAT Standard 3.0, CF Version 1.8
Checked at: 2023-05-24T10:29:55

Number of checked netCDF files: 1

Mandatory ATMODAT Standard checks passed: 0/4 (4 missing, 0
error(s))
Recommended ATMODAT Standard checks passed: 1/20 (19
missing, 0 error(s))
Optional ATMODAT Standard checks passed: 0/9 (9 missing, 0
error(s))

CF checker errors: 2
CF checker warnings: 79
```

```
=== Short summary ===

ATMODAT Standard Compliance Checker Version: 1.3.2
Checking against: ATMODAT Standard 3.0, CF Version 1.8
Checked at: 2023-05-24T10:29:55

Number of checked netCDF files: 1

Mandatory ATMODAT Standard checks passed: 4/4 (0 missing, 0
error(s))
Recommended ATMODAT Standard checks passed: 16/20 (4
missing, 0 error(s))
Optional ATMODAT Standard checks passed: 7/9 (2 missing, 0
error(s))

CF checker errors: 0
CF checker warnings: 0
```

Figure 4.7: Example output of the ATMODAT Standard Checker for MITRAS model data V23 before (left) and after (right) the application of *nc2atmodat*.

expensive simulations including cloud processes within a reasonable time. Due to the irregular grid, the obstacles become coarser towards the edges of the domain. This can be seen by comparing the model domain of F24 in Figure 4.8 with the domain of V23 (Figure 4.6). The building block west of the lake Alster (9.99°E and 53.5565°N in Figure 4.6 and x=-400 m and y=700 m in Figure 4.8) is much finer resolved in the domain of V23. The stretched grid is also visible in the vertical cross-section of the building mask (Figure 4.4). In contrast to V23, in this model domain orography is considered (Figure 4.8).

As mentioned in section 4.3.1, the structure of building surface-related variables as provided initially in the MITRAS output was changed. Each surface-related variable is now written as six 4D variables for each wall with corresponding coordinates and one 3D variable holding a plane-view of all building surface roof cells. For tbuisurf on east walls (tbuisurf_e), for instance, scalar vertical coordinates (z) and y-coordinates (y) are used, but for the x-direction, the location of the grid cell boundaries (xv) is used. Figure 4.9 shows results for precipitation at roofs (left) and the different roof heights (right). While the precipitation field has a general increase towards south-east in the domain, it is also visible that high buildings might receive higher precipitation values than their surroundings, as visible at and around the highest building ($x \approx -800\,m$, $y \approx 750\,m$).

For the temperature on bottom surfaces, i.e. roofs (tbuisurf_t), the vertical coordinate equals the height of the grid cell boundaries, while the horizontal coordinates still represent the grid cell centres. With the variable tbuisurf_p a plane view of all values on roofs is provided, which is shown as an example for the rain amount on roofs in Figure 4.9a. The corresponding roof heights can be seen in Figure 4.9b.

Figure 4.10 shows a visualisation of the building surface-related variables and the location at the building surface. The building surface-related variables are located at the vector grid that overlaps with the scalar grid, indicated by the transparent colours overlapping with the building mask. Figure 4.11 shows the netCDF header for these building surface-related variables.

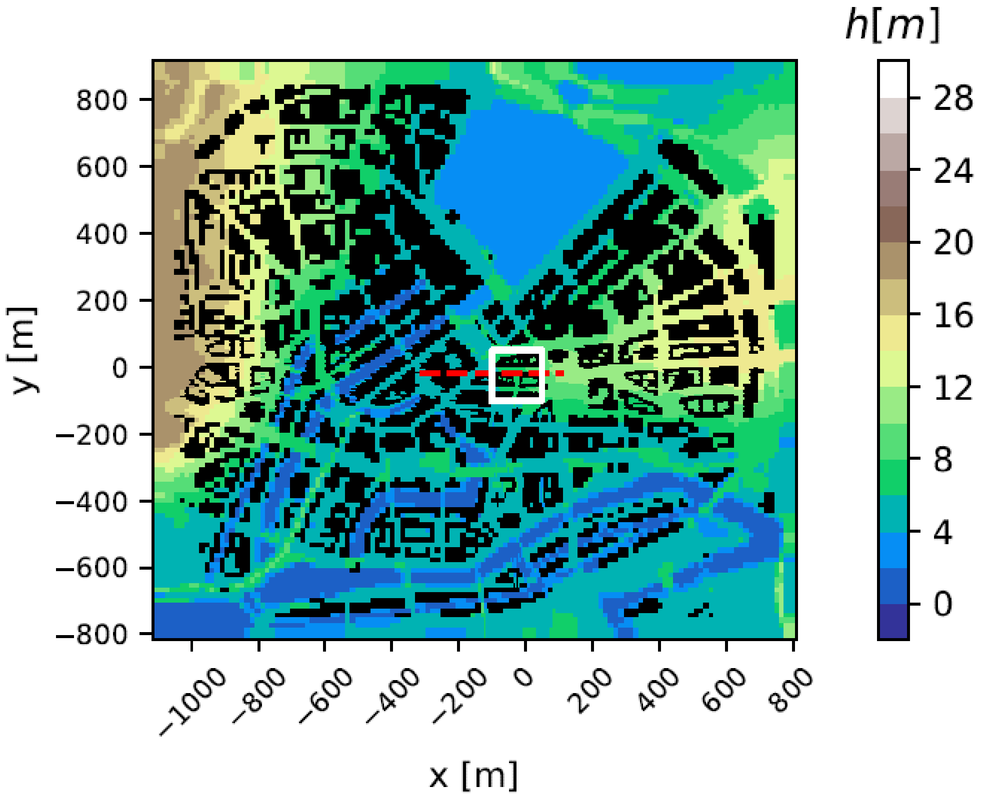

Figure 4.8: The model domain of Hamburg city centre using a horizontally non-uniform grid with buildings in black and the orography height h in m. The white box denotes the area with equally sized grid cells (equidistant area), and the red dashed line denotes the location of the cross-section in Figure 4.4.

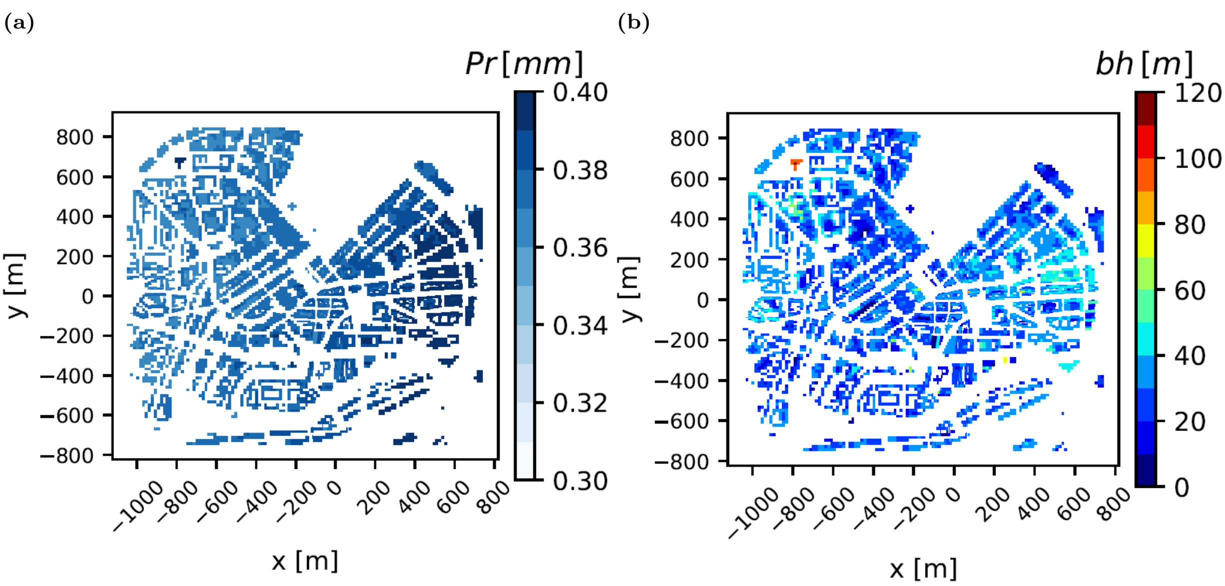

Figure 4.9: Amount of rainwater Pr on all roofs (a) and all roof heights bh (b).

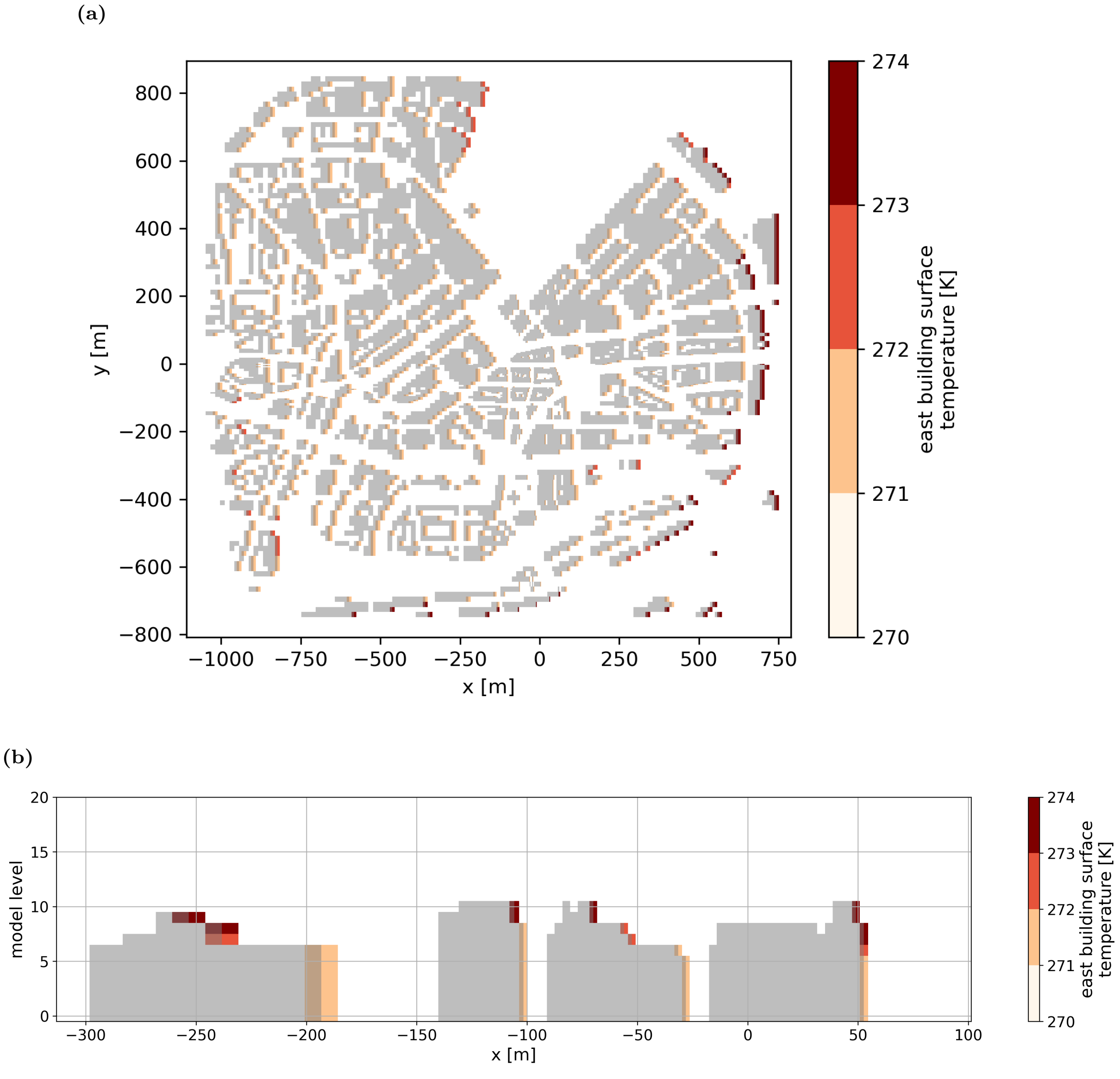

Figure 4.10: Example for the 3D representation of building surface-related variables representing the eastern building wall surface temperature as (a) horizontal cross-section view of the whole domain at a height of 10 m above ground and (b) vertical cross-section view located at the position denoted in Figure 4.8. Grey areas denote the building positions.

```
float tbuisurf_e(time, k, y, xv) ;
    tbuisurf_e::_FillValue = -999.f ;
    tbuisurf_e:long_name = "east building surface temperature" ;
    tbuisurf_e:structure = "5901 " ;
    tbuisurf_e:units = "K" ;
    tbuisurf_e:cell_methods = "time: point k: y: xv: mean" ;
    tbuisurf_e:comments = "Surface temperature at eastern building wall" ;
    tbuisurf_e:coordinates = "lonu latu" ;
float tbuisurf_n(time, k, yv, x)  ;
    tbuisurf_n::_FillValue = -999.f ;
    tbuisurf_n:long_name = "north building surface temperature" ;
    tbuisurf_n:structure = "5901 " ;
    tbuisurf_n:units = "K" ;
    tbuisurf_n:cell_methods = "time: point k: yv: x: mean" ;
    tbuisurf_n:comments = "Surface temperature at northern building wall" ;
    tbuisurf_n:coordinates = "lonv latv" ;
float tbuisurf_t(time, kv, y, x)  ;
    tbuisurf_t::_FillValue = -999.f ;
    tbuisurf_t:long_name = "top building surface temperature" ;
    tbuisurf_t:structure = "5901 " ;
    tbuisurf_t:units = "K" ;
    tbuisurf_t:cell_methods = "time: point kv: y: x: mean" ;
    tbuisurf_t:comments = "Surface temperature at building roof" ;
    tbuisurf_t:coordinates = "lon lat" ;
```

Figure 4.11: Header of F24 dataset with focus on the description for the newly added building surface-related variables.

4.5 Conclusions

To foster the publication of FAIR model data created by micro-scale obstacle-resolving atmospheric models (ORM), ORM datasets should be well-described, self-explanatory and easy to understand by model data users. This can be achieved by the application of data standards to model datasets. Existing data standards, such as the CF conventions, are widely used for numerical model data, but are mainly designed for climate model data and are therefore not fully applicable to ORMs, as they miss features that are common in ORMs. Therefore, the standardisation of ORM data is not trivial as general discipline-specific standards are not established. To overcome the lack of standardised datasets in this discipline, a standardisation routine for model data was developed using results of the ORM MITRAS as an example. The aim was to standardise ORM MITRAS results so that they comply with the ATMODAT Standard, since the the ATMODAT Standard (Ganske et al., 2021) is a generic data standard for atmospheric model datasets.

The ATMODAT Standard requires the use of the netCDF data format together with the CF convention and requires a set of global metadata that needs to be provided with the dataset. To fulfil the requirements of the ATMODAT Standard and the CF conventions, the post-processing of MITRAS received several adaptations and extensions. The description of variables and ORM features provided by the post-processor *m2cdf* was improved to be better understandable for model data users not familiar with the model. This includes the extension and addition of descriptive attributes and information to each MITRAS variable with their corresponding entries and, if applicable, standard names. Furthermore, the post-processing was extended to include a more accessible representation of building surface-related variables. Previously, building surface-related variables, (e.g. wall temperatures, radiation fluxes or rain water amounts on building walls), were stored in a vector array structure. To view these variables, a data user would requires further knowledge about the in-model representation of the variables. The new storage in 3D (time dependent 4D) array structures is more intuitive for data users and but improves the provision of building surface-related variables. It comes, however, with the drawback of needing much more storage. It will become even more relevant in future work, when further building surface-related properties will be implemented to study e.g. effects of adaptation or heat mitigation measures. This includes, for example, different facade structures, representation of bridges or balconies or green facades.

Additionally, a new post-processing program called *nc2atmodat* was introduced. This routine summarises the required steps for the preparation of MITRAS netCDF datasets. It provides the option of small changes, corrections or additions to variables and metadata, without the need to adapt the models specific post-processing routines. If required, it allows the cropping of the dataset in space and time and a variable-based selection of the data. This new post-processing has been successfully applied to two ORM MITRAS datasets, which were subsequently released as FAIR datasets in the World Data Center for Climate (WDCC). The program *nc2atmodat* is written in Python and available as open source (Voss, 2025a).

First tests on the application of *nc2atmodat* beside MITRAS model data were carried out on model data from the mesoscale sister model METRAS (Schluenzen, 1990) and the chemistry model CityChem (Karl, 2024).

The restructuring, organisation and assignment of MITRAS variables revealed that many variables, which are typically occurring in ORMs are not covered by existing standard names. For a better identification and understanding across different ORMs and to avoids confusion or misinterpretation in data handling, they need to become standardised. This would need a community approach to collect variables between different models, the definition and discussion of new standard names for typical ORM variables and the proposal of new ORM variables to the CF conventions. This would foster and increase the amount of well-described ORM datasets and improve their usability.

5 Challenges in standardising environmental physical model results and ideas for application

This chapter has been prepared for publication as:

V. Voss (2025): **Challenges in standardising environmental wind tunnel model results and ideas for standardisation**.

In preparation for *Data Science Journal*

The full manuscript is included in this chapter. Layout and numbering are adopted to fit this thesis. All references are combined in References. Supporting information and statements that will be submitted with this manuscript can be found in Appendix D.

Abstract

To investigate atmospheric processes in the lower atmospheric boundary layer, micro-scale, obstacle-resolving numerical models and environmental wind tunnel models can be used. Due to the nature of wind tunnels, their datasets are a combination of measurements and model data, often provided as dimensionless results, to be applicable for a range of different cases and environmental or meteorological conditions. Qualified boundary layer wind tunnel data can be rescaled to real-world conditions, but the reuse of such data must be approached with caution. Therefore, handling, preparing and reusing such data is not straightforward. Processing and qualifying boundary layer wind tunnel data for a safe reuse requires a number of factors and issues to be considered, and partially differs from processing results from numerical modelling.

However, there is no general data standard for environmental wind tunnel data available that defines certain rules and conventions in describing and providing reusable data. Whereas in numerical modelling and for observational measurements, the netCDF Data format is widely used, in boundary layer wind tunnel modelling, modellers often process and provide data in their own individual way. To increase the FAIRness of wind tunnel model data, a concept for how wind tunnel data could be provided was developed. This concept should be applicable in a conventional way as well as guide the translation to more standardised netCDF datasets. A post-processing routine for translating wind tunnel data into netCDF was developed and tested on exemplary datasets. Nevertheless, the application of the developed concept is not straightforward and does not resolve issues completely, due to the variety of ways to process data and the multitude of usecases of the data, making a universal application of the code difficult. Hence, the purpose of this paper is rather to discuss a first concept to transform boundary layer wind tunnel data into netCDF, and to examine the associated advantages, disadvantages, difficulties and considerations.

This work, therefore, serves as a generic template for preparing and providing such data. Furthermore, it is intended to offer a starting point for improving the standardisation of boundary layer wind tunnel data and eventually for establishing a corresponding netCDF framework. This work should provide an overview of what might be possible in terms of standardisation for environmental wind tunnel data, what to consider, implement or rethink.

5.1 Introduction

The use of models to investigate and assess atmospheric processes and the publication and exchange of model or research data with the scientific community are important aspects in atmospheric science. To publish data, it is recommended to follow the FAIR guidelines (Wilkinson et al., 2016) to ensure that published research data is well-described and usable for data users. To foster the publication of FAIR data, Data Standards can be used. In atmospheric sciences, model and observational data are often provided in the so-called netCDF data format (Rew and Davis, 1990), following the Climate and Forecast (CF) conventions (Eaton et al., 2024). The CF conventions are guidelines to generate well-described and self-explanatory netCDF datasets from the climate and weather forecast modelling community, ensuring that data is standardised, understandable and comparable. netCDF and the CF conventions are used in other earth science-related disciplines, like oceanography or for observational measurement data (Eaton et al., 2024). However, applying the netCDF-CF standard to disciplines outside of established fields is not straightforward, as the current state of the standards does not always allow for sufficient representation of discipline-specific properties. With the AtMoDat (Atmospheric Model Data) project[23], a generally applicable data standard for model data was developed, which aims to foster the publication of well-described datasets in such non-standardised disciplines. One of these disciplines is urban climate modelling, which uses so-called obstacle-resolving atmospheric models (ORM). ORMs are micro-scale models used to investigate processes in the lowest layer of the atmosphere (the boundary layer) over small-scale, complexly structured domains such as cities, urban neighbourhoods, and complex terrain (Oke et al., 2017). They use high temporal and spatial resolutions to study small-scale processes. Model domains are no larger than a few km^2. Any obstacles affecting the flow, such as buildings or trees, are resolved directly. ORMs can be either numerical or physical models. In numerical modelling, many models use netCDF as an output data format (Jänicke et al., 2021; Voss et al., 2024), whereby the CF conventions can be partially applied to their model output data, but many typical variables or properties are not standardised yet.

In physical modelling of atmospheric flow and transport phenomena, however, no standardised way to provide model data is available. Layout, design, descriptions and output format of physical model data depend on each modelling group or data producer. This is due to the rather complex conduction of physical model simulations and resulting production of model data:
Physical model data are predominantly generated in wind tunnels or water tanks, where a smaller-scale

[23]https://www.atmodat.de/

replica of the study domain is placed inside the facility and the flow and transport around the obstacles are modelled and measured inside the model domain (Oke et al., 2017).

They provide controllable conditions to study atmospheric flow and transport under fully documentable boundary conditions. Physical or fluid modelling is based on the concept of physical similarity, whereas, particularly for atmospheric flows, approximate physical similarity can be achieved (Verein Deutscher Ingenieure, 2024). Any turbulent flow above a critical turbulent Reynolds number behaves similarly and becomes Reynolds-number independent. By ensuring a sufficiently high Reynolds number of the flow and geometric similarity, the flow of a fluid in a physical model behaves similarly to the flow in the real world. Simulation results can therefore be transformed to similar real-world flow conditions. To be applicable for different situations and conditions, wind tunnel data is usually provided as dimensionless quantities, normalised by measured reference quantities and needs to be transformed to become spatial and temporal data again. Hereby, the range of re-scaling is limited, amongst others, by the range of Reynolds-number independence to be documented by a data producer. Since wind tunnel model simulations mostly cover small domains, the lowest layer of the atmosphere and, unless generated in a rotating facility, do not consider the Coriolis force. Additionally, model setup, settings, sizes, physical boundary conditions, conversions, etc., needed to be documented sufficiently and provided with the datasets (Schatzmann and Leitl, 2002). Due to the dimensionless nature of the data, the user of wind tunnel data must be able to handle such data properly to avoid misinterpretation. Sufficient information on how the data was non-dimensionalised and eventually might be rescaled to real-world conditions must be communicated with every set of experimental data.

Leitl and Schatzmann (2010) mention that the use case of wind tunnel model data is limited, as the model data is generated for certain weather and statistical conditions. Additionally, users of such data must provide knowledge and experience to handle wind tunnel data or should be in contact with the data producer to properly use the non-standardised data. Otherwise, misinterpretation and misuse of the data might affect the results and conclusions of research projects.

Environmental physical models are used to investigate atmospheric processes in urban areas, limited by the abilities of the physical modelling facility and measurement devices available (Oke et al., 2017; Zhang et al., 2016). Furthermore, physical model data are often compared with observational measurements or numerical model data. As recommended by the VDI standard (Verein Deutscher Ingenieure, 2017), wind tunnel datasets are often used to validate and evaluate numerical models, as done for example with MITRAS (Grawe et al., 2013), MISKAM (Schatzmann and Leitl, 2002) or OpenFoam (Franke et al., 2012). For example, the Environmental Wind Tunnel Facility (EWTL) of the Universität Hamburg provides a variety of comparison datasets which were generated in validation projects (e.g. CEDVAL (Schatzmann et al., 2010), COST 732 Franke et al. (2012) with MUST data (Sabatino et al., 2011) or COST ES 1006 Baumann-Stanzer et al. (2015) with Michelstadt data (Bastigkeit, 2011)).

Since physical model simulations are complex and generating physical model data requires significant resources, guidelines for conducting such simulations are available (e.g. Verein Deutscher Ingenieure (2024)). However, these guidelines do not specify or determine how the data must be provided for further use. Providing data in different formats, with different names and designs, impairs its reusability. Better and standardised data availability can foster the reuse of a large amount of experimental data and may facilitate scientific progress. As there is no standardised way to provide such data, the following questions arise: how can wind tunnel data be standardised? What options and methods are available for describing the data, and what difficulties arise in doing so?

From a numerical modeller's view, the transformation of wind tunnel model data into netCDF seems to be a rather simple task. This work tries an approach to a possible standardisation of physical model data. For this, a standardisation concept based on the CF Conventions has been developed and tested on wind tunnel datasets from the EWTL Working group. This paper can not solve all problems, but rather provides concepts, tries to raise awareness on and to initiate a discussion and progress on this topic.
Section 5.2 contains typical relevant aspects of wind tunnel model data and describes what should be considered for standardising boundary layer wind tunnel datasets, e.g. consistent descriptions and names for wind tunnel model variables. In section 5.3, an overview of attributes and formats for the description of wind tunnel datasets is compiled. Furthermore, recommendations and suggestions on how wind tunnel data can be provided are derived. A post-processing routine called *wt2atmodat* (Voss, 2025b) to translate wind tunnel data into the netCDF-CF format was developed, their routines and the application on datasets are described in Section 5.4.

5.2 Challenges for harmonisation of wind tunnel datasets

Physical model data could be considered as model simulation and measurement data at the same time, since they are measurement results under simulated, simplified atmospheric conditions. They appear as observational measurement data, but are frequently intentionally designed to be scalable and reusable to a certain extent for different scenarios and use cases. Physical models are limited to the laboratory boundaries, such as the model scale is limited by the facility size, modelled processes are restricted by the capabilities of the experimental facility or the measurements are limited by the instrumentation used. Most wind tunnels simulate neutral stratification, whereas the natural atmosphere can range between stable and unstable stratification.

Wind tunnel data are provided as dimensionless quantities; the data are therefore not directly applicable to real-world conditions and need to be "re-scaled" by a data user to become valid for real-world conditions and to be comparable to, e.g. numerical model data. Hereby, the used scales, the range of application, spatial and time dependencies has to be considered and applied before the data can be properly used. All this information needs to be provided with the model data.

The provision of data is an important topic in wind tunnel modelling (Leitl, 2000; Schatzmann and Leitl, 2002). Currently, there is no universally or generally defined way to provide and describe wind tunnel data or data standards available. Moreover, each physical model laboratory or even each model data producer decides on how their data is processed and archived. The preparation and storage of such data often depend on the use case and limitations of the datasets, as well as on the decisions of the model producer, which data should be provided and how it should be stored. Furthermore, the amount of required information that has to be provided is large, as simulation-specific properties are needed to fully document the data as well as the data quality must be ensured Schatzmann and Leitl (2011) Without data standards, reusing and accessing data is difficult and requires adaptation for each dataset. However, a data standard would ensure that model data producers have guidelines on how to prepare their data for publication and that model data users can understand and process the data more easily.

In this work, three different wind tunnel datasets, generated in the EWTL Working group at the University of Hamburg, were used (Table 5.1). The data is usually created in the following way: For the recording and first processing of the wind tunnel measurements, the working group uses a suite of software tools introduced by Fischer (2011) and continuously developed/extended to a specific Python repository. It prepares the raw measurement data into an initial dataset, containing a collection of typical variables. The data producer then applies their application- or project-specific post-processing routines to further process, analyse and evaluate the data. After this step, the data producer comprehends the data for publication. As no standardisation for post-processing of environmental wind tunnel data is specified, data producers have a large degree of freedom in the preparation of publishable datasets, concerning what data, features and information should be provided. Even individual data from one and the same wind tunnel laboratory may look different because they are prepared in different projects for various applications and are often not intended for general use although they remain of general interest and value. With individually prepared datasets, as listed in Table 5.1, different amounts, details of information, data format and file, variable names and descriptions are provided to describe the dataset. The following

Table 5.1: Dataset used in this study with description of model domain and measurement type

Dataset	**Study Case**	**Model Domain**	**File Structure**
Harms et al. (2012)	Air pollutants, Puff release measurements	Idealized City	ascii files, Excel Files, Readme-Sheet (excel) and Readme Textfile
Freitas et al. (2021)	Flow measurements	Idealized mesoscale City structure	ascii files, Readme as separate Textfile
Michel et al. (2023)	Heavy Gas dispersion	Industrial area (realistic and idealized)	ascii files, Readme as PDF

describes some aspects to consider when creating a standard.

5.2.1 Data and data formats

Wind tunnel datasets are often provided as ASCII-text files or Excel datasets. While the netCDF format is commonly used for data generated by numerical models or via observational networks, it is sparsely used for wind tunnel model results. Traces for the attempt of the use of the netCDF file format for wind tunnel datasets were proposed by ONERA (2024). A publication of wind tunnel data published via Mendeley Data, used the particle image velocimetry (PIV) as measurement device, where the flow within the wind tunnel is filmed by a camera. The results are provided as a raster field netCDF files (Wang et al., 2024), whereas wind tunnel measurement results are provided as normalised flow properties in an Excel file by Wang et al. (2024).

Another dataset that uses netCDF as data format for wind tunnel measurements is published under Zenodo by Lapo et al. (2020a), and contains wind direction measurement results with a hot optic wire (Lapo et al., 2020b).

The datasets mentioned in Table 5.1 were generated in the same wind tunnel facility and working group, but are provided differently: Freitas et al. (2021) and Michel et al. (2023) provided the data as dimensionless quantities, whereas Harms et al. (2012) provided the data in full-scale units. Furthermore, different file formats are used, either ASCII text files (Michel et al., 2023; Freitas et al., 2021; Harms et al., 2012) or Excel spreadsheets (Harms et al., 2012). Documentation or readmes are partially provided within the files that contain the data or as supplementary individual files provided as Excel spreadsheets, PDF, or ASCII text files.

In addition, the variable names differ across the datasets. Without unified variable names, there is a risk of misinterpreting them, as if similar names are used for different properties, a reuser may mistakenly assume the purpose of a variable based on previous experience or misinterpret the name. This can easily be avoided by implementing standardised names, naming structures and data formats.

Wind tunnel datasets can be provided in many different ways: as statistical aggregated model results, raw data, as dimensionless or dimensional, scaled quantities (model scale or calculated to full-scale). Overall, the amount of data and the type can be summarised into the following categories:

- Information about the atmospheric boundary layer (ABL)
- Simulation setups: e.g. implementation, instrument positions, configurations.
- Timeseries: raw data, e.g.concenctration measurements, flow measurements.
- Mean data: statistical mean data of the measurements.
- Uncertainties: either as own data series or as ranges

- Readmes: With information about the model setup, explanations of variable meanings, formulas, measurement devices, etc.

5.2.2 Model Setup and Boundary layer

Since the data are generated in laboratories under controlled conditions, settings and setup need to be documented to ensure the reproducibility of the simulations (Verein Deutscher Ingenieure, 2024). Relevant to the overall model simulation aspects are used model scale, reference velocity, particle or gas properties or emission source, etc., as well as the wind tunnel setup, e.g used roughness elements, to create the necessary boundary layer for the simulations. These information are currently mostly written in the header of the model data or provided in supplementary files.

5.2.3 Metadata

Metadata describe the overall dataset, such as information about the data producer, name of the wind tunnel facility, institutions or dataset title, descriptions, etc. In the example datasets (Table 5.1), this information is often provided in additional readme files or header sections within the dataset files. The amount and content provided in the metadata are different in each dataset. Due to the diversity of the example datasets with different treatments and features that need to be taken into account, the consideration of what is always necessary to provide seems not trivial and may differ from the description of a specific numerical code used, for example. It is not 'the wind tunnel in lab XYZ', it is about the wind tunnel facility and how it was used in a certain test setup to achieve certain results of certain quality.

Furthermore, possible inconsistencies or missing information make the data for inexperienced users of wind tunnel datasets difficult to understand. A predefined set of such global metadata information, which can also be used for wind tunnel datasets, is provided with the AtMoDat Standard (Ganske et al., 2021). Although the AtMoDat Standard is mainly defined for netCDF files, their proposed concept could also be used for conventional data formats.

5.3 Suggestions for wind tunnel variable data descriptions

A wind tunnel data standard should guide data producers in generating well-described datasets that are easy to understand and reusable. Since the CF data standard is established for numerical model data, its general structure could be an option to provide and describe wind tunnel data. In the following, examples and recommendations for describing environmental wind tunnel data are derived from the CF conventions (Eaton et al., 2024). If it is not possible to apply the netCDF format to wind tunnel data or is insufficient, the recommendations could be applied on other file formats.

To derive wind tunnel-specific data description guidelines, typical properties and aspects in wind tunnel data must be identified. The first step involves collecting wind tunnel variables to define uniform variable names. In this example, variables occurring in the three datasets mentioned in Table 5.1, are collected and compared to each other. The comparison should reveal which variables exist, how they are provided, named to derive unified variable names and definitions. A second step involves the assignment of variable attributes for a better variable description. Furthermore, handling for information such as model setup, constants, descriptions or metadata should be defined. In this section, the proposed ideas can be used as a starting point or as a base for the discussion of a potential standard.

5.3.1 Recommendation to describe wind tunnel variables

The netCDF data format provides the possibility to create self-describing datasets by adding attributes to the variables. Attributes can hold additional information or descriptions to make the meaning of a variable clear. This section describes a naming concept for wind tunnel variables, as well as suitable attributes and examples of their application. Existing attributes from the CF conventions that might be usable for wind tunnel data descriptions are listed in Table 5.2.

Besides the definition of attributes, each variable in a dataset is assigned a variable name. Using unified variable names and naming structures helps to recognise a quantity across different datasets and avoids confusion for the data user. Many variables in wind tunnel datasets can be provided in three different states: dimensionless (DL), full-scale (FS) or model-scale (MS) quantities. To indicate the status of a variable and to avoid defining three sets of variable names for the same variable, it might be advisable to add a label to the variable name:
For example, if data is provided as full-scale data, a corresponding variable should have "_FS" attached to it (e.g. "*variable_FS*"). Data provided in model scale are labelled with "_MS", ("*variable_MS*"). Data provided as dimensionless data should then be labelled with "_DL", ("*variable_DL*"). Adding these labels should help to name the variables similarly without the need to create new variable names just because the dimensionality changes. Variables in the model- or full-scale must then be provided with a unit attribute, e.g. m in full-scale or mm in model-scale. Technically, dimensionless variables do not need to be assigned a unit attribute. Nevertheless, to be consistent in the naming convention, it is advisable to add the '_DL' flag to a dimensionless variable and the unit set to "1". Furthermore, for dimensionless variables, it is then advisable to provide an attribute containing the formula that indicates how the data was calculated to become dimensionless.

Figure 5.1 shows an example header for a netCDF wind tunnel dataset with a variable describing the horizontal or longitudinal wind component called "u". The equivalent variable name in the numerical model naming convention would be "x-wind" or "wind component in direction". If it is provided as a model scale quantity, then the assigned variable name should be "u_MS", for full-scale, then "u_FS". Dimensionless variables are normalised by reference variables, e.g. U_ref, which is the reference wind velocity, that is used to normalise flow and turbulence quantities. In the literature or in dataset headers, dimensionless variables are labelled as "$\{variable\}\_Uref[-]$" e.g. "u_Uref" for the normalised longitudinal wind com-

Table 5.2: Overview and collection of CF attributes that possibly are suited for wind tunnel model variables. Further explanation of the attributes is provided in Eaton et al. (2024).

Attribute	**Description**
long_name	A self-describing variable name. Ideally, it should be constructed as a plot title or description.
units	if available, provided in SI units. Dimensionless variables should be set to 1 or not provided with an unit attribute.
comment	hold any additional information, that are required to explain a variable sufficiently and cannot be included to the `long name`. Can be used for information, that is normally provided in a README section/file.
conversion*)	contains the conversion formula to normalise or re-scale variables. Normalised quantities (_DL) should be assigned two formula entries (conversion_FS, conversion_DL). Note: The CF conventions incorporate an attribute called formula for such transformation purposes. This attribute feature might be suitable for wind tunnel model data in a similar way. (More explanation in this section).
ancillary_variables	Measurement uncertainties can be stored in an additional variable that is linked to a variable via the `ancillary variable` attribute (Eaton et al., 2024).
flags (optional)	describes e.g. status information like the functionality of measurement devices (Eaton et al., 2024).
fillValue (optional)	Indicates the missing data information value (e.g. -999, NaN)
bounds (optional)	If provided as gridded data, `bounds` can be used to describe the extent of a grid cell (Eaton et al., 2024). It might be suitable to describe the extent of the measurement volume of a device at each measurement point.

*) wind tunnel model specific attribute.

ponent. By applying the proposed convention, this quantity could be described as "$\{variable\}\_DL[-]$" or "u_DL".

Wind tunnel data can be provided in three states: dimensionless, full-scale and model-scale; hence, a wind tunnel variable can be transformed to all three states. Therefore, the formula or method for the transformation must be provided. For dimensionless variables, it is recommended to provide the information on how the data was made dimensionless. Providing the data as a netCDF file, dimensionless variables can be assigned attributes that contain the formulas for the transformation as shown in Figure 5.2. It would be recommended to provide two attributes with formulas to transform the data from model-scale to dimensionless quantity (attribute: `conversion_DL`) and the formula to re-scale the values to full-scale (attribute: `conversion_FS`). These conversion formulas are crucial for the correct use and handling of wind tunnel model data.

In CF conventions, there is an option to provide information on how to convert variables into other properties. For example, vertical coordinates can be provided as pressure or sigma coordinates or height-related z-coordinates. A conversion into each system is possible via a certain conversion formula, which is currently hardcoded in the CF conventions. Further information on this topic (Parametric Vertical Coordinates) can be found in Eaton et al. (2024). With the attribute `formula_terms`, variables required for the transformation can be mapped to each term that is required in the formula. Since the attributes `formula_term` and `formula` are already used, a new attribute `conversion` to describe the conversion process in wind tunnel model data could be established, to avoid confusion or misinterpretation with the already established attributes. For now, as wind tunnel data is not commonly provided and therefore not included in the CF conventions, the attribute `conversion` could carry the information on how to transform the data into dimensionless or dimensional (Full-scale, model-scale) quantities (Figure 5.2). In the future, it might be possible to provide the option to handle the conversion of wind tunnel variables in a similar way as `formula` is used today. With wind tunnel conversion formulas becoming hard-coded properties, the handling of such data would become more standardised.

```
dimensions:
        index = 6 ;
variables:
    int64 index(index) ;
    float x_FS(index) ;
        x_FS:long_name = "x position full scale" ;
        x_FS:units = "m";
        x_FS:comment = "full scale position of the measurement point on the longitudinal axis"
    float u_DL(index) ;
        u_DL:long_name = "dimensionless x-wind" ;
        u_DL:units = "1" ;
        u_DL:comment = "dimensionless u-wind component (longitudinal direction), normalised by the refrence wind velocity Uref";
        u_DL:conversion_DL = "u_DL = u_MS/Uref" ;
```

Figure 5.1: Example for a basic netCDF wind tunnel header. x_FS is the variable name, 'long_name', 'unit', 'comment' and 'conversion' are the used attributes to describe the variables. "index" indicates that 6 measurement points are provided in this example dataset.

```
double time_DL(index) ;
    time_DL:long_name = "dimensionless time" ;
    time_DL:_FillValue = NaN ;
    time_DL:units = "1" ;
    time_DL:conversion_DL = "time_DL=(time_MS*Uref)/Lref_MS" ;
    time_DL:conversion_FS = "time_FS=(time_DL*Lref_FS)/Uref_FS";
double conc_DL(index) ;
    conc_DL:long_name = "dimensionless concentration" ;
    conc_DL:_FillValue = NaN ;
    conc_DL:units = "1" ;
    conc_DL:conversion_DL = "conc_DL=(conc_MS*Uref_MS*L_ref_MS^2)/q_MS" ;
    conc_DL:conversion_FS = "conc_FS=(conc_DL*q_FS/(Uref_FS*L_ref_FS^2)" ;
```

Figure 5.2: Example for a netCDF wind tunnel header for dimensionless variables C* (concentration; conc_DL) and t* (time; time_DL). Both variables provide two conversion attributes, describing how to convert the variable into a dimensionless and full-scale quantity.

5.3.2 Recommendation to describe reference variables

Many wind tunnel variables are used as fixed properties in the model simulations. For example, the reference wind velocity (U_{ref}) which is measured at a certain position ($x_{ref}, y_{ref}, z_{ref}$), characteristic lengths L_{ref} and heights H_{ref}, or the model scale, which was set for the experiment. For constants that are connected to a certain position in the model domain, such as the reference wind velocity (U_{ref}), the measurement location coordinates (x_{ref}, y_{ref}, z_{ref}) have to be provided. Those values are important for normalising and rescaling the data to different atmospheric conditions. Since the reference wind speed is used to normalise or re-scale measurement quantities, the refererence coordinates z_{ref}, x_{ref}, y_{ref} have to be provided in the respective dimensionality (model-scale (z_{ref-MS}, y_{ref-MS}, x_{ref-MS}) and full-scale (z_{ref-FS}, y_{ref-FS}, x_{ref-FS})), which can be done via attributes attached to U_{ref}.

Within the netCDF file structure, there are three possible ways to provide such variables:

Method 1: Similar to measurement variables, with attributes and information assigned to them (Figure 5.3a,). It has the advantage that a value is directly assigned to each variable and, therefore, directly readable during the processing of the netCDF file. Provided as individual variables, many attributes from the previous section could be applied. However, if these pieces of information were provided as individual variables, the datasets would carry many additional variables, which would extend the dataset and might overload it.

Method 2: Within a specific reference variable that stores such information altogether within the attribute structure (Figure 5.3b). Instead of many individual variables, it would require one variable that contains this information in the variable attributes. The information would be mainly visible in the header view of the variable.

Method 3: Provided completely in the global metadata section of the netCDF file (Figure 5.3c). However, it could lead to a rather overloaded metadata section if all such constants are provided there.

A combination of the proposed methods is possible. Furthermore, reference variables can get assigned attributes that determine the validity ranges (either `valid range` or the combination of `valid min`, `valid max`) to indicate to what extent the data is use- and scalable (Figure 5.3). Further constant variables are the actual model scale, the emission source location coordinates, the emission source strength, or the inflow wind direction. Such constant variables are mostly written in the header section or provided via a supplementary file and might be suitable to be provided within the global attributes section (Example in Figure 5.3c). Measuring device information could also be stored in netCDF variables. The attributes can be used to describe the measurement devices with corresponding properties (attribute: "hardware", Figure 5.4). If necessary, links between the measuring device and the variables measured with it can also be named by an ancillary attribute in the variable entry (Figure 5.3a).

(a)

```
 float Uref ;
    Uref:long_name = "reference wind velocity" ;
    Uref:units = "m s -1" ;
    Uref:comment = "Uref is used to normalize measured quantities" ;
    Uref:valid_min = "4 m s -1" ;
    Uref:valid_max = "20 m s -1" ;
    Uref:hardware = hardware_01;
    Uref:coordinates = " Xref Yref Zref ";
float Xref ;
    Xref:long_name = "reference x position" ;
    Xref:units = "m" ;
    Xref:comment = "x coordinate position where Uref was measured" ;
```

(b)

```
float constants ;
    constants:Uref = "4 m s -1" ;
    constants:Xref = "20 m"
    constants:Yref = "100 m"
    constants:Href = " 100 m" ;
    constants:Lref = " 1 m" ;
    constants:model_scale = "1:500" ;
```

(c)

```
global_attributes :
    Uref = "4 m s -1" ;
    Href = "100 m" ;
    Lref = "1 m" ;
    model_scale = "1:500"
```

Figure 5.3: Examples and variations to provide and describe wind tunnel reference and coordinate variables within the netCDF structure. Variations include the provision as an individual variable for each constant, which allows additional explanatory attributes (a). Within an overarching variable (here named "constants") and the constant information is provided as attributes (b). Or the provision in the global metadata netCDF structure (c).

```
string Hardware_01 ;
    Hardware_01:measurement_device = "Laser-Doppler Velocimeter (LDV)" ;
    Hardware_01:spatial_resolution_MS = "x=0.05, y=0.05, z=0.8 mm" ;
    Hardware_01:spatial_resolution_FS = "x=0.37, y=0.005, z=0.015 m" ;
    Hardware_01:temporal_resolution = "6 Hz at full scale" ;
```

Figure 5.4: Example for a variable providing hardware information, such as measurement devices with corresponding attributes to describe their features.

5.3.3 Recommendation to describe wind tunnel global metadata

Global metadata are any relevant descriptive information about the overall dataset, providing potential data users a quick overview of the dataset. They carry technical features such as the model name, the names of authors or data producers, contact information, creation dates and descriptive features such as dataset title, abstracts, summaries, etc. In the example datasets, the amount and type of information provided vary depending on the purpose of the data and the decision of the data producer. To be more coherent by the provision of certain constant global metadata attributes, the Atmospheric Model Data (AtMoDat) Standard (Ganske et al., 2021) can be considered. The AtMoDat Standard is designed to standardise data from disciplines without their own data standards and guides data producers to produce well-described and FAIR dataset publications (Ganske et al., 2021, 2022). A collection of global attributes for netCDF files is defined, categorised into mandatory, recommended and optional attributes. To fulfil the minimum requirements of the standard, the mandatory global metadata attributes must be provided. Furthermore, the AtMoDat standard requires netCDF datasets that follow the CF conventions. The AtMoDat Standard could be followed for wind tunnel datasets provided as netCDF data. However, if wind tunnel data can not be prepared as netCDF datasets, it seems possible to follow the recommendations of this standard for well-described datasets.

5.4 Realisation of recommended standardisation

In the following, an updated processing routine to generate standardised wind tunnel model netCDF datasets is described (Figure 5.5): At first, the data producer conducts the model simulation measurements. After this, the data is then analysed and selected by the data producer for publication (step 1). At this stage, the data is typically provided in commonly used data formats. To standardise the data after the AtMoDat standard, two further steps are now introduced: Wind tunnel model data results are now translated into the netCDF. For this, a processing script such as *wt2atmodat* (step 2) can be used (details in Section 5.4.1). In step 3, the finalised netCDF data is checked using the AtMoDat data checker to ensure that the data fulfils all the requirements to be standardised after the AtMoDat standard (details in Section 5.4.3).

Figure 5.5: Workflow for the generation of publishable netCDF wind tunnel datasets.

5.4.1 Development of wt2atmodat

To convert the wind tunnel model datasets into the netCDF file format, the processing routine *wt2atmodat*[24] can be used. The processing routine is a prototype for a possible standardisation routine and therefore mainly adapted to the datasets listed in Table 5.1. It is written in Python[25] and uses the Xarray (Hoyer and Hamman, 2017), Pandas (pandas development team, 2020) and Numpy (Harris et al., 2020) modules. It consists of three files: wt2atmodat_main.py is the main routine and steers the processing, functions.py contains the necessary functions and routines to run the process and dict_WT_variables.py is a dictionary file that holds the variables metadata information for the data description. Furthermore, an Excel spreadsheet is used to provide global metadata attributes required after the AtMoDat standard.

The initial version of the processing routine was developed to transform the data of Freitas et al. (2021) and then afterwards adapted to the datasets Michel et al. (2023) and Harms et al. (2012). The processing script is able to read and process data from the EWTL working group, since they are more or less consistently in their layout. Nevertheless, adaptations were required for each dataset, as the header information and file format were different. The general workflow to process and translate data is shown in Figure 5.6, whereas each processing step is described in the following section.

5.4.1.1 User Input The processing script starts with an input section, where paths and file names are provided. By providing the path to a directory, all files are selected.

5.4.1.2 Read Data Then, the path or file is read, parsed and transformed into a Pandas data frame. If requested, the processing script can read and combine multiple files into one combined data frame if the header structure is consistent across the files. This could be useful, for example, if data for each measurement point is provided in a separate file. Header lines in the input text file, indicated by the character (#), are identified and skipped. Then, only the content of the last header line is used as column names of the Pandas data frame and will be assigned as the variable names within the netCDF file. The processing script *wt2atmodat* distinguishes variable name and units, if units are provided within brackets ('[ ]') or parentheses ('()'). The unit is then automatically assigned via the unit attribute to the variable. Any information stored in the comment lines (#) data header is not processed by *wt2atmodat*. Therefore, information from this section must be added manually either as a new variable or a global attribute.

5.4.1.3 Transform dataframe to Xarray After reading the input and storing the data in a Pandas dataframe structure, the Pandas dataframe is transformed via the module Xarray into the basic netCDF data file structure. If each measurement point has a distinct spatial coordinate (x, y, z), the data can be provided in a rasterised form, allowing a 3D representation of the data. Therefore, *wt2atmodat* tries to provide the data as both 1D and 3D files.

[24](https://doi.org/10.5281/zenodo.17176544

[25]https://www.python.org/

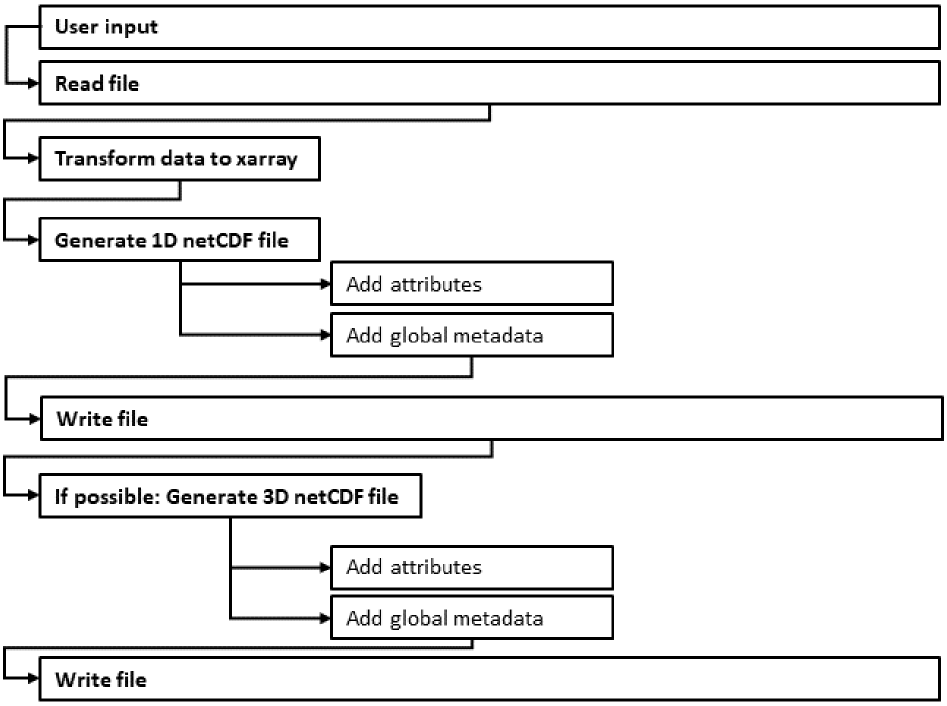

Figure 5.6: The processing script wt2atmodat is used to transform wind tunnel model data into the well-described netCDF file format. This is the general workflow of the processing script.

5.4.1.4 Generate 1D netCDF file As default, the data is translated into a 1D netCDF file that uses an index to access the data (Figure 5.7). It resembles the direct translation of the table structure into the netCDF format, where the data is accessible and iterated via an index. This method is suited for datasets in which coordinates are duplicated, e.g. due to measurement repetitions. Otherwise, Xarray cannot process duplicates in the coordinate entries, which makes the data ambiguous and prevents it from being retrieved correctly.

5.4.1.5 Generate 3D netCDF file If unique x-, y- and z-coordinates are available, a translation to a 3D dataset or rasterised grid is possible (Figure 5.8). Hereby, each variable gets three spatial coordinates assigned, which resemble the height, lateral and longitudinal (z, y, x) dimension. If the data is provided on a rasterised 3D grid, further coordinate variable attributes `cell methods` and `bounds` can be assigned. `Bounds` define the position of the grid cell boundaries, whereas `cell methods` describe the statistical representation of the variables (e.g. "means"). The attribute `bounds` might become relevant to quantify the measurement volume from a measurement device.

5.4.1.6 Add attributes When the data is translated into the netCDF file structure, variable attributes are added to the dataset. Variable attribute entries and information are set via the file "dict_WT_variables.py" for each variable in the netCDF dataset. At least, for each variable, the attributes `long name`, `units`, and `comment` with corresponding entries have to be provided.

```
netcdf example_index {
dimensions:
    index = 67 ;
variables:
    int64 index(index) ;
    float x(index) ;
        x:long_name = "x position" ;
        x:axis = " X " ;
        x:units = "m" ;
    float y(index) ;
        y:long_name = "y position" ;
        y:axis = " Y " ;
        y:units = "m" ;
    float z(index) ;
        z:long_name = "height_above_ground" ;
        z:axis = " Z " ;
        z:positive = " up " ;
        z:units = "m" ;
    float Mag_DL(index) ;
        Mag_DL:_FillValue = NaNf ;
        Mag_DL:long_name = "dimensionless 2D wind magnitude (UW plane)" ;
        Mag_DL:ancillary_variables = "dMag_DL" ;
        Mag_DL:conversion_DL = "Mag/Uref = square root[(u/Uref)^2 + (w/Uref)^2]" ;
        Mag_DL:coordinates = "z y x"
```

Figure 5.7: Example for a 1D wind tunnel netCDF file header. The 1D approach uses an index to iterate through the data.

```
netcdf example_xyz {
dimensions:
    z_FS = 7 ;
    y_FS = 3 ;
    x_FS = 6 ;
variables:
    double z_FS(z_FS) ;
        z_FS:long_name = "height" ;
        z_FS:axis = "Z" ;
        z_FS:positive = "up" ;
        z_FS:units = "m" ;
    double y_FS(y_FS) ;
        y_FS:long_name = "y_position" ;
        y_FS:axis = "Y" ;
        y_FS:units = "m" ;
    double x_FS(x_FS) ;
        x_FS:long_name = "x_position" ;
        x_FS:axis = "X" ;
        x_FS:units = "m" ;
    float Mag_DL(z_FS, y_FS, x_FS) ;
        Mag_DL:_FillValue = NaNf ;
        Mag_DL:long_name = "dimensionless 2D wind magnitude (UW plane)" ;
        Mag_DL:units = "1" ;
        Mag_DL:cell_methods = "z_FS: y_FS: x_FS: mean" ;
        Mag_DL:ancillary_variables = "dMag_Uref" ;
        Mag_DL:comment = "Mag/Uref = square root[(u/Uref)^2 + (w/Uref)^2]" ;
```

Figure 5.8: Example for a 3D wind tunnel model netCDF file. The 3D approach creates a grid or raster and the data is located at a point which is defined by the coordinates (x,y,z).

5.4.1.7 Add global metadata Lastly, global attributes with corresponding metadata are added to the file. For this, an Excel spreadsheet that contains an overview of all global metadata required by the AtMoDat Data Standard (Ganske et al., 2021) is provided with the code. Also, it contains example entries and explanations to guide the data producer to fill in the necessary information for their own datasets.

5.4.1.8 Write file The fully processed data is saved in a netCDF file. Automatically, every dataset will be written as 1D-netCDF files (default) and, if applicable, as 3D-netCDF datasets.

5.4.2 Post-processing wind tunnel datasets

The development and functionality of the *wt2atmodat* post-processor was based on the sample wind tunnel datasets used in this work, mentioned in 5.1. As they are generated by different members of the EWTL working group, carried out for different studies with different research goals and scopes, they differ in aspects such as data layout, formats, naming and headers. In the following, the application and processing of *wt2atmodat* on the example datasets is tested and described.

5.4.2.1 Dataset one The first wind tunnel model dataset, that was translated into netCDF was provided by Freitas et al. (2021). The first version of the processing script *wt2atmodat* was developed and tailored to this dataset. The purpose of the wind tunnel simulations was to create a validation dataset for an urban canopy parametrisation for the numerical model METRAS and used by Cheng (2023). Hereby, a pyramid-shaped obstacle should represent a coarse version of an urban city. The dataset consists of several files containing mean data, raw data, measurement uncertainties and boundary measurement information. The data is provided as text files or ASCII files, whereas model setup information are provided in the PDF format. Metadata and model documentation information were provided via a supplementary text file.

Measurements were done at ten measurement points around the obstacle and provide mainly wind and turbulence-related variables. The measurement results are provided for each measurement point as separate files. With wt2atmodat, the translation of the data into both 1D and 3D netCDF file configurations is possible. For the 1D approach, each measurement point files was transformed both as individual files and a combined file, whereas for the 3D approach, the ten files were merged and stored in one netCDF file.

The 1D files are accessible via the index-like structure, as described in 5.4.1. As a downside, in a simple netCDF viewer (e.g. ncview), the index points are represented in a data viewer as a ”graph” (Figure 5.9a), which could lead to misinterpretations of the data as time series. To access the data correctly, the values need to be selected by conditions (5.9b).

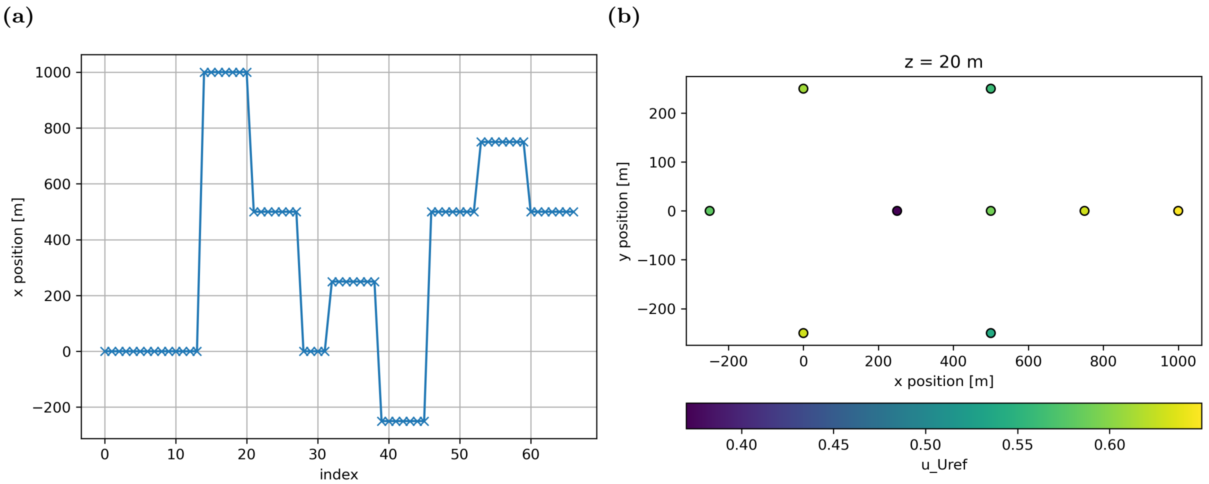

Figure 5.9: Example view of netCDF wind tunnel data as a 1D file with the index coordinate approach. Data is taken from Freitas et al. (2021), representing the available x-coordinates value per index and dimensional longitudinal wind velocity (u/U_{ref}). The index is considered a numbering element in this dataset (67 entries) and establishes a correlation between different variables and coordinates. Since the data is by default plotted as a graph in a netCDF viewer, it can be misinterpreted as a timeseries or profiles (a). With the correct handling of the data (e.g. selecting the index at the coordinate value $z = 20$ m), the data can be spatially represented at the measurement point location (b).

In comparison, Figure 5.10 shows the translated dataset in a rasterised or 3D view. The shape and size of the raster are determined by the number of measurement points in each spatial dimension and allow in a netCDF viewer a spatially easier representation of the measurement point distribution (Figure 5.10a). However, the 3D raster distorts the view of the data, as the true extents of each measurement point and the distances between each measurement point are not correctly represented. Instead, this might lead to the visual misinterpretation that the measurement value might represent a much larger area, than the actual size of the measurement point. Hereby, the correct representation of the data can be achieved by proper plotting settings (Figure 5.10b), similar to the 1D case.

The measurement point in the model domain is representative of a measurement volume, which is (upscaled to full size) only a few centimetres wide. In this example dataset, the measurements are representative of a volume of $0.5m \times 0.5m \times 0.5m$. Theoretically, such grid size information about the cell extent could be provided via "cell_bounds" as variables "x_bnds, y_bnds, and z_bnds". However, using the bounds attribute to represent the extent of the grid cells might not be the correct use as intended by the CF conventions and typical netCDF viewers do not include the positions of the bounds.

5.4.2.2 Dataset two After the initial development of *wt2atmodat* based on the dataset of Freitas et al. (2021), the processing script was next used for the data from Michel et al. (2023). This dataset includes measurements to investigate the dispersion of heavy gases in industrial areas, containing idealised

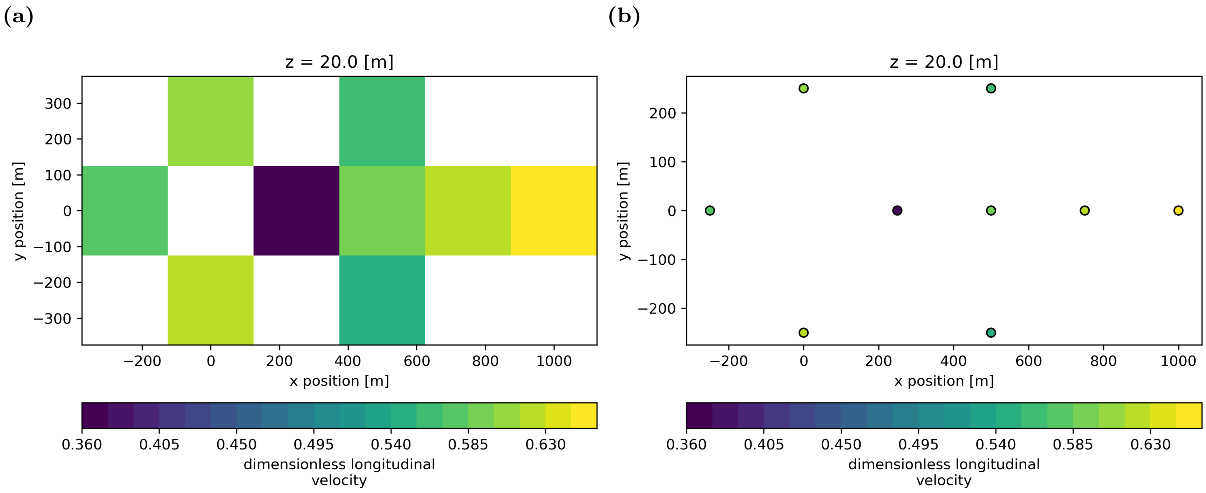

Figure 5.10: Example of netCDF wind tunnel data taken from Freitas et al. (2021). Represented is the dimensional longitudinal wind velocity (u/U_{ref}). In the 3D file, the coordinates x,y and z are used to rasterise the data. When viewing the data, spatial allocation is available through the position in the grid. The grid is distorted in the default view, which leads to a misrepresentation of the data (a). However, more accurate point plots can be created with the 3D dataset as well (b)

and realistic model domains and different source locations for the gas leakages (Michel, 2023). Compared to Freitas et al. (2021), the datasets contained more test cases, experiment setups and configurations of the model domain, leading to a larger and diverse amount of data and files. The measurement data were provided as ASCII-text files, whereas the metadata was partially provided within the header section of the measurement data (such as experiment setup information) and with a PDF file (detailed study purpose, guidelines and descriptions for accessing and handling the data). Furthermore, the model domain layouts and positions of the obstacles are provided as shapefiles. Besides wind and turbulence measurements, Michel et al. (2023) contains data and measurements of concentrations of gaseous matter and provided new variables to the variable collection. The data contains flow, dispersion mesurements and boundary layer measurements.

Processing the data with the first version of *wt2atmodat* was possible but required adaptations. The header layout of the Michel et al. (2023) dataset differs from Freitas et al. (2021), as they contain more information about the simulation setup in the file itself. The processing of the *wt2atmodat* was adapted to be able to process differently sized data headers. Furthermore, Michel et al. (2023) used slightly different names and information about the data than Freitas et al. (2021). This makes it clear that, on the one hand, there is a need for unified variable names, and on the other hand, each simulation and data producer has to consider different aspects which are relevant to their simulations.

Although the data was successfully translated into a 1D index, difficulties occurred during processing using the 3D approach. The automatic translation into the 3D dataset was hindered due to coordinate duplications. Such duplicates can occur due to repeated measurements or if measurements are taken on two planes (e.g. the x-z plane and the x-y plane; therefore, the x coordinates are used twice). Xarray is not able to translate the data into netCDF if the coordinates are ambiguous. In such cases, if the data has to be provided as 3D files, the data producer must decide beforehand whether redundant information should be removed. To avoid any loss of data, the data will only be written in the 1D index data structure by *wt2atmodat*.

Nevertheless, for some files, the translation into the 3D netCDF files was possible. Compared to the 3D netCDF in the Freitas et al. (2021) case, the rasterised grid is much finer, as more measurement points are available that define the structure of the grid. This significantly finer or denser grid makes the data less visible in the raster display 5.11.

5.4.2.3 Dataset three The third dataset was created within the COST Action ES1006 and contains measurements of so-called puff released concentration to investigate gas emissions and dispersions in urban areas (Environmental Wind Tunnel Laboratory, 2025; Harms et al., 2012). The used model domain is the "Michelstadt" case, which resembles the concept of a generic European city structure. The dataset contains data for two approach flow wind directions, 0 and 180 degrees, and is divided into a basic dataset, continuous time series, and puff time series. Different to the datasets of Freitas et al. (2021) and Michel et al. (2023), this datasets was provided as full-scale data.

In this dataset, the data is primarily provided as Excel files, with parts of the data available as ASCII files which contain the same information. In comparison, the Excel file is more clearly structured and easier for a data user to understand than the ASCII files, since the data is highlighted and a documentation sheet is included. In the ASCII file structure, the same data is split into three files: a 'readme' file, a list of filenames, and the data file. Without comparing the data to the Excel file, it was not obvious that the information about the source point used was provided differently in the ASCII files. This issue was only discovered after comparing the data with the Excel structure. Therefore, the initial processing of the ASCII data was incorrect and resulted in misrepresentation. Another issue was a partially inconsistent use of variable names across the ASCII and Excel file. For example, the variable "Duration of the measurement" is called *"t_FS[h]"* in the ASCII-file and *"Duration [h]"* in the Excel. Such inconsistencies need to be avoided in the future.

Due to the different structure in this dataset, the application of *wt2atmodat* was more difficult than in the previous cases. Generally, the data was provided in such a way that a generic application of *wt2atmodat* was partially possible for the ASCII files. To be able to process Excel files, *wt2atmodat* needed adaptations. However, post-processing the data with *wt2atmodat* reached its limit due to the large amount of files with slightly different structures and information attached. Bulk processing, as done in previous dataset cases, required additional handling. The translation of this dataset into the

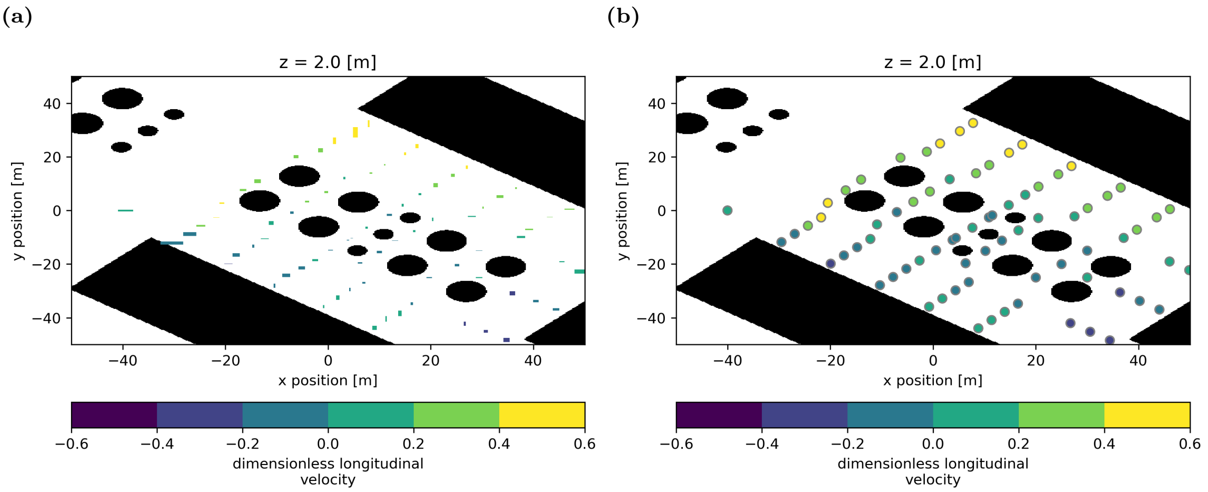

Figure 5.11: Wind tunnel measurement data provided as 3D netCDF data. Data was taken from Michel et al. (2023). Represented are the dimensional longitudinal wind velocity (u/U_{ref}) and obstacle position (black). Due to the greater number of measurement points across the entire domain, the resulting raster grid is rather fine (a), and the data is less visible than in the scatter plot view (b).

netCDF data files was therefore not successful, as the manual adaptations required too much effort to be reasonable.

Nevertheless, this dataset provided valuable insights that were not evident in previous processing. Further information about model settings, measurement data, conversion formulas and general comments from the READMEs need to be added to the netCDF data. While the netCDF structure can technically store this information, achieving a finalised, uniform approach that can be understood by any data user appears to have no clear solution and requires too much effort in handling individually at present.

5.4.3 Standardisation quality check with the AtMoDat data checker

To ensure that netCDF wind tunnel model data fulfil the requirements of the ATMODAT Data Standard (Ganske et al., 2021), the ATMODAT DATA CHECKER (Ganske et al., 2022; Kretzschmar et al., 2022) can be used. Since the AtMoDat Data Checker is only applicable to netCDF files, other file formats cannot be tested.

An example of the checker output applied to a wind tunnel netCDF file is depicted in Figure 5.12. Regarding the CF conventions, the checker output states that the tested dataset contains four errors relating to CF conventions and seven warnings concerning variable attributes. Hence, the minimum requirements of the AtMoDat Standard (all mandatory global metadata attributes are provided) are fulfilled. In conclusion, the tested dataset does not pass the checker and would require correction. Wind tunnel datasets, which are translated with *wt2atmodat*, are more likely to fulfil the AtMoDat standard, as the processing script provides the infrastructure to follow and fulfil the requirements of the AtMoDat

```
=== Short summary ===

ATMODAT Standard Compliance Checker Version: 1.3.2
Checking against: ATMODAT Standard 3.0, CF Version 1.7
Checked at: 2024-09-30T19:40:42

Number of checked netCDF files: 1

Mandatory ATMODAT Standard checks passed: 4/4 (0 missing, 0 error(s))
Recommended ATMODAT Standard checks passed: 9/20 (11 missing, 0 error(s))
Optional ATMODAT Standard checks passed: 7/9 (2 missing, 0 error(s))

CF checker errors: 4
CF checker warnings: 7
```

Figure 5.12: AtMoDat Data Checker Output information for processing one wind tunnel netCDF file. The Checker shows that the AtMoDat Standard minimal requirements are met (mandatory: 4/4 and no errors) while the CF checker spotted 4 errors. Therefore, for this dataset, adaptations are required.

standard. However, since the CF conventions are not fully applicable to wind tunnel model datasets, problems in fulfilling the requirements might occur.

5.5 Discussion

The publication of well-described and FAIR research data becomes more and more important, while the provision of well-described datasets is not always trivial. Especially in environmental wind tunnel modelling, where each modelling group has their own way to prepare model datasets. For this, data standards, such as the AtMoDat standard, can help data producers to generate well-described and FAIR publishable datasets.

Since netCDF and the CF conventions are commonly used for numerical models and observational datasets, environmental wind tunnel model data are generally not provided in such structures. This work proposes a concept to standardise wind tunnel model datasets from the Environmental Wind Tunnel Laboratory (EWTL) at the University of Hamburg. It includes the use of the netCDF file format, which is a self-describing data format and data descriptions that are heavily inspired by and based on the climate and forecast (CF) conventions.

Generally, translating wind tunnel model data into netCDF is possible, but the method for describing or providing the data within this structure requires further testing and adaptation. Establishing definitions and methods specifically for wind tunnel model datasets are required. For example, some properties that are established in the CF conventions would be suitable to describe physical model data. For wind tunnel model data, it would require adaptations of these attributes towards wind tunnel-specific aspects or needs to be different to the established use. For example, the attribute formula has a certain standardised structure and use case in the CF convention. A similar approach would be suitable for wind tunnel variables. Since it is currently not implemented or established, a different approach must be applied.

One of the main outcomes of this work is the need for standardised naming and definitions for wind tunnel variables. Without unique variable names, definitions or standardised descriptions, the ambiguous use of names or descriptions becomes possible, which could confuse potential data users. Therefore, unique variable names, definitions and features need to be defined and assigned to wind tunnel datasets, regardless of whether netCDF or other file formats are used. Further challenges besides discussing, testing, development and implementation of a data standard for wind tunnel datasets include certain aspects in the processing of the datasets.

A Python-based post-processing tool, called *wt2atmodat*, was developed to translate ASCII or Excel wind tunnel model datasets to the netCDF data format. Although the tool was primarily adapted and developed based on the datasets provided by the EWTL, applying the processing to the data was not straightforward. While the general layout of the provided datasets is more or less similar and can be processed by *wt2atmodat*, not all features provided with the datasets, e.g. in the header information, can be considered in the automatic processing and must be added individually to the processed data. Due to the different ways the datasets are prepared and provided, it is not easy to find a common ground for an overall way to standardise the provision of netCDF wind tunnel data. The processing routine is a prototype for a possible wind tunnel netCDF standardisation routine. Since it is mainly adapted to the datasets listed in Table 5.1, processing other datasets from the EWTL or other working groups will require adaptations in the code. Furthermore, the possibility of providing data shown in this paper is only one of many.

Furthermore, even though the netCDF data format produces self-descriptive data, netCDF data viewers can play a role in the misinterpretation of data. In the used example datasets, the measurement data is given as averaged values for a point at certain coordinates. By providing the spatial coordinates, the data can be rasterised. In a netCDF data viewer, however, this could cause the rasterised cells to distort the data and thus be misinterpreted to be valid for a large area instead of the actual point location. With the index variant, where each measurement point can be accessed via an index, the viewer may inadvertently assume that this point data could be understood as a continuous measurement. These might be potential issues with the netCDF data format that have to be considered. In the future, the production of wind tunnel data should aim to become unified in terms of naming, structure and data formats.

5.6 Outlook

At first glance, it seems unusual that no general data standardisation for environmental wind tunnel model data is available. However, this is not surprising given the many different uses and characteristics of the data. Data producer from the wind tunnel modelling group of the Universität Hamburg claims that they frequently interact with data users to guide them with the complex handling of the data.

Therefore, the limits, challenges and possibilities of processing and handling wind tunnel datasets with dimensionless variables as well as their supplementary files must be understood by a data user who wants to work with wind tunnel data successfully. Wind tunnel model data are often provided as dimensionless data, so that different atmospheric conditions are applicable by rescaling the data. However, the range of application has its limits, which must be communicated, provided and defined with each dataset. An inexperienced data user could overlook this aspect even in standardised datasets, and therefore misinterpret the data. Furthermore, the temporal and spatial representation of wind tunnel model data needs to be addressed, since wind tunnel datasets are not provided as georeferenced and time-dependent data and do not consider effects that are related to the Coriolis force. Furthermore, the provision of model domain information, such as layout and positions of obstacles or roughness elements, is currently not considered. Since the representation of terrain or obstacles in netCDF is technically possible, it might be inadequately represented in this data format. Only by the sufficient provision of this information could one guarantee proper data handling.

The preparation of netCDF of wind tunnel datasets seems to have potential, but still requires considerable effort in terms of proper handling. It includes further testing of methods and standardised definitions to create a wind tunnel-specific model data standard. While storing the actual data inside the netCDF format is rather simple, metadata or simulation information, stored in readmes or header sections, have to be provided as well. Such information might be too extensive for the netCDF structure and must be provided as additional information. If wind tunnel netCDF datasets are not feasible and the traditional ways are preferred by the data producers (e.g. ASCII-Text Files or Excel), the proposed concept could still be used to provide data in a standardised way. These aspects needed to be considered in the standardisation process. Model results can probably become standardised. It seems that rules can be established for this purpose that apply across datasets, while minor deviations are also possible and may be unavoidable. Nevertheless, striving for uniform structures, naming conventions and data provision is a challenge that will result in long-term and meaningful data usage. The provided examples in this work should be used as ideas for how to make standardised netCDF environmental wind tunnel datasets possible. They are neither finalised nor finished, and may not consider other aspects which are relevant in other simulation processes. Therefore, wind tunnel modelling groups and the community need to develop and discuss further methods for the realisation of such a final output data standardisation.

6 Conclusions

This thesis focuses on the production and preparation of well-described, FAIR model data from obstacle-resolving models (ORMs). ORMs are so-called microscale models, which can be numerical or physical models, that are used to investigate atmospheric processes in domains resembling urban neighbourhoods or city quarters. As obstacle-resolving models, they are capable of resolving buildings explicitly and their effect on the atmospheric flow. In atmospheric sciences, datasets are often provided as netCDF files and standardised via the CF conventions. Due to the specialities of ORMs, these established standards are not directly applicable to ORM model data, which might impair the publication and reuse of ORM datasets. By providing methods and tools for the creation of standardised datasets for the microscale model MITRAS and a generic approach for wind tunnel datasets, the first steps and concepts towards the standardisation of such data have been done. Hereby the following questions are answered:

RQ1: How do obstacle resolving numerical models store their results so far and what do users want?

For the development of a standardised way to prepare and provide data, a common ground among different models and modeller groups must be found. To collect information about different ORMs, their model properties and opinions or experiences from model users, a survey was conducted over the period from September 2019 to June 2023. The survey was targeting numerical micro-scale and obstacle resolving modelling groups to gain an information of how modellers and data users handle the data. The survey requested information about technical aspects of the models, input and output files, data formats and user knowledge on standards. This survey was spread over several conferences (Section 3) and was partially addressed directly to modelling groups, ORM modellers and ORM data users.
The major outcome of the survey was that models provide their information and results in various output data formats: some in model-specific formats, some as netCDF files, some both. Furthermore, the named models are different in their technical features (used filtering methods, coordinate systems, etc.) and their provision of information on model input or output settings and setups. Regarding the use of data standards, many participants were aware that a common and standardised output format, such as the use of the netCDF and CF conventions in the atmospheric community, has a greater benefit and should be the preferred model data format across the modelling community. Regarding the application of the CF standards, they claim that they are partially implemented in models with the netCDF output format. The participants mention that for an ORM data standard, they prefer to use the netCDF data format and to adapt ORM-specific data conventions to ORM results.

In retrospect, it appeared that some aspects of the conduction and setup of the survey might have been done differently. The total number of finished surveys and, therefore, usable replies to the survey is lower than the number of cancelled survey entries. As the survey contains a lot of technical questions, it can be assumed that participants may have been reluctant to answer incorrectly and therefore not completed the survey. In the future, if such surveys have to be conducted, questions and setup of the survey should be done differently, e.g. focusing more on getting explicit user experiences and asking less about technical aspects, since reliable information about the technical setting can be better obtained through literature and model documentations. However, answering technical model questions provides a rough estimate of each participant's model knowledge and can show, where possible misunderstandings or misconceptions occur. Such information might become interesting for a model developer, and can help to ensure that such crucial model aspect information needs to be communicated more prominently. Nevertheless, within the few responses, the need for the standardisation of ORM model data is widely acknowledged as an important and current topic in this community and will require further action within the modelling community.

RQ2: How can existing standards be applied to obstacle-resolving numerical model data?

In numerical modelling, the application of the CF conventions on model datasets is already partially common for some ORMs (Section 3), but the degree of implementation of existing standards might vary. Therefore, the AtMoDat standard can be used as a guideline to help data producers create well-described, publishable model datasets. The main requirements of the AtMoDat standard are the use of netCDF files and a data descriptions after the CF conventions. Since it should be universally applicable, the standard was applied to datasets from the ORM MITRAS, since MITRAS utilises the netCDF format as the final output format (Section 4). To fulfil the requirements of the AtMoDat standard, the model-specific post-processor m2cdf needed updates and extensions, to ensure data will be provided according to the CF conventions with uniform and sufficient information. The adaptations in the models post-processor mainly included the addition of CF attributes, the equipment of self-describing variable names and currently valid structures for describing CF features. Furthermore, MITRAS-specific properties, such as the position of buildings in the model domain or the representation of building surface-related variables, were modified to become more accessible and understandable for model data users. Additionally, a secondary post-processing routine called nc2atmodat was created to finalise the preparation of model data. These adaptations ensure now the generation of well-described, standardised and ready-to-publish MITRAS model data.

During the process of implementing the standardisation requirements, some difficulties and hurdles occurred. While many descriptive aspects for achieving CF-conform datasets were implemented rather quickly, others required more effort and workarounds to become properly assigned to the MITRAS output. Since the target format and layout of the data were known, it was still necessary to determine the right methods in the processing to achieve these desired output structures. Hereby, the biggest hurdle was the implementation of geographical coordinates to ensure georeferenced datasets, since the MITRAS model output initially did not provide the geographical information in a netCDF-typical way. The solution was found through cross-checking and comparisons with other example netCDF files, which provide georeferenced datasets and plotting with GIS applications. In the end, a MITRAS-specific and reproducible way to assign geographical coordinates was found.

Furthermore, assigning CF standard names to MITRAS variables was not always trivial. Some MITRAS variables seem not to always align with the description of presumably fitting standard name definition. For example, assigning standard names for radiation fluxes to MITRAS variables was not trivial, since they are not clearly upward or downward oriented, or include certain aspects, e.g. reflections on obstacle surfaces, into account. It became apparent that the degree of applicability of the CF standard names was not always easy to determine. During the collection of typical ORM variables and the comparison of variables from the ORM PALM-4U with MITRAS, it became visible that presumably similar variables can be slightly differently named or defined. This impairs the understanding across different models or can even be misleading if similarly named properties do not resemble the same variables. This applies in particular to variables that are not yet standardised and are therefore pure ORM variables. An overview of such "false friends" in ORM variables between PALM-4U and MITRAS, with possible similarly usable CF standard names, is provided in Appendix C.3.1.

Overall, resolving such standardisation issues required a certain understanding of the netCDF structures and model-specific knowledge, which can become hurdles in any data standardisation processes and for non-experienced users. It is possible that similar problems have been encountered by other data producers or for other models, and it is likely that an inexperienced data producer will encounter this issue afterwards as well. As the nc2atmodat post-processor is already published, it could be used as an example for other model data producers facing similar problems when processing their own data. With the updated post-processing, the preparation of well-described and AtMoDat-conform MITRAS model data is now possible, and the first standardised MITRAS datasets are published (Voss, 2023a; Ferner et al., 2024). And since m2cdf is also used by the sister model METRAS, the newly developed post-processing routines can be used to standardise METRAS model datasets as well.

RQ3: How can a possible standardisation derived for obstacle resolving numerical model data be transferred to wind tunnel model data?

In environmental wind tunnel modelling, there is no community-wide and generally applicable data standardisation for wind tunnel model data available. Instead, wind tunnel model data is prepared differently: Each simulation requires their own descriptions and depends on rules defined and applied by data producers or modelling groups. Wind tunnel model data can be provided in different shapes, formats and layouts, and the amount of information provided vary. Therefore, the possibility of creating a uniform and well-described set of wind tunnel model data after the requirements of the AtMoDat Standard was investigated. Based on three different wind tunnel datasets generated in the Environmental Wind Tunnel Facility (EWTL) at the University of Hamburg, it became apparent that, despite the different data producers, model scopes, and choices, consistent structures are available. This allows the data to be rather easily transformed into the netCDF format. Since the netCDF data format with the CF standard is commonly used for numerical modelling and observational datasets, there are many data description options and possibilities possible, which might be appropriate to use on wind tunnel data.

The general applicability of the netCDF-CF concept on wind tunnel model data was tested (Section 5). While the translation of wind tunnel data into netCDF is rather simple, deriving suited methods for variable attributes descriptions from the CF conventions is not straightforward. For the translation of wind tunnel model data, a generic processing tool named wt2atmodat was developed. Despite being generated by different data producers, the test datasets provide a certain consistent structure, making it possible to take a semi-automated approach to translating them into netCDF files.

Difficulties occurred in the application and definition of fitting descriptions and standardised wind tunnel variable names. For example, as wind tunnel variables can be available in three different states (dimensionless, full-scale or model-scale data), a naming structure is required to distinguish these variables. Further information, such as conversion formula, model setup information or used boundary conditions must be provided. Those are often provided within additional documentation files, but a concept to store such information within the netCDF structure was derived and conceptualised. Lastly, there is still a risk that a data user may misinterpret or misuse wind tunnel netCDF data. There is always a risk that they may be interpreted as natural measurements, in which case their wind tunnel-specific limitations would be ignored. Alternatively, depending on how the data is presented, viewers of netCDF data may display it in a distorted manner.

Overall, the application of the netCDF-CF-inspired standardisation for wind tunnel model data seems to be possible; however, due to the various possibilities, limitations and requirements, it is currently difficult to define an overall applicable standardised structures that cover all aspects in wind tunnel modelling. The standardisation concept derived from the netCDF-CF standard and presented in this work still requires adaptations. With a successful implementation of wind tunnel data as netCDF, an application of the AtMoDat standard would become possible.

An important step in standardisation is defining standardised variable names and descriptions. This enables wind tunnel variables to be described in a unique and unambiguous way. A collection of wind tunnel variables is provided in Appendix D.3. Once standardised names, definitions, data structures and concepts have been established, these can be used to describe data in traditional formats as well. But more importantly, the wind tunnel modelling community must discuss and agree on general structures, output data formats, layouts and naming conventions to ensure the standardised provision of wind tunnel model data.

6.1 Challenges and limitations

At first glance, the task of applying an existing data standard to ORM data appears to be a straightforward task, especially if models already utilise the desired output data formats or if many examples and references are available. But the actual application of standards is not trivial, whether certain structures are missing or the data has to be first translated into the desired data formats. Especially in disciplines without prior consistent preparation of data, the application can be a hurdle. In obstacle-resolving modelling, several issues arise.

Generally, one important factor by providing standardised data might be missing awareness, knowledge or experiences of the data producer in terms of standardisation. For each model, the right way to post-process the data or required adaptations in their current methods need to be found and tested. In the case of MITRAS, the required methods to adapt the data after the CF conventions and the AtMoDat standard need to be found. The development of the program nc2atmodat to process MITRAS model data is a consequence of this need. Since it utilises Python, it enables a rather quick and convenient way to explore the methods, processes and tools that are required to adapt the model output data into the desired final netCDF format. However, it seems more preferable to include newly developed data processing methods, implemented in nc2atmodat, into the model-specific post-processor m2cdf. It would reduce the need for a secondary post-processing step, prevent maintenance effort and minimise user application errors. For practical reasons, adapting m2cdf is only partially done, as it seemed only reasonable to implement such features, if general or required data processing steps to achieve the desired output are understood. Now that a successful method for standardising MITRAS model data has been developed, corresponding adjustments can be more easily transferred to the actual post-processor. Nevertheless, incorporating minor adjustments or adding corrections to the model results will be always quicker with nc2atmodat or similar tools, than adjusting and translating the data with m2cdf each time. Therefore, the programme will continue to serve its purpose.

Nevertheless, there is still potential to improve the current provision and post-processing of the model data. For example, the addition of georeferenced coordinates to the MITRAS model data, which is currently added via the second processor nc2atmodat, could theoretically be taken from the model input file (GA file) and pre-processing (GRIMASK) (Appendix C.3.2). This would require the information to be carried through the model run until the post-processing steps. However, this approach was not feasible for three major reasons: Firstly, it was easier to implement the provision of such information via Python from the outside than to adapt the many different FORTRAN routines to achieve the desired result. This avoided the need to revise many different model routines. However, calculating georeferenced variables within the model-specific post-processing reduces the likelihood of programming errors in any outer application. Secondly, to retain the initially used grid information, it would be necessary to make changes to the GA files, either by replacing existing coordinates or by adding two new columns to the table containing the corresponding coordinates (e.g. UTM). This change will require adaptations to several model routines and affect the sister model METRAS. For the calculation of the initial profiles for MITRAS and METRAS, the same routines are used, and METRAS already provide geographical coordinates. Thirdly, writing all this information might have an impact on the duration of the model runs, since additional information needs time and storage space to be written to each model output per time step. Thus, the actual impact of writing this information should be tested to estimate how much this will affect the model performance.

Another not yet implemented feature is the provision of information about the used surface cover classes. Since METRAS and MITRAS use the same surface cover class library, each surface cover class name can be taken from the model input and added to the model output via post-processing. This feature would improve the description of output files of both models. Similar treatment applies to the provision of model input information (e.g. used values for model initialisation) or model steering settings (switch positions, simulation settings), which likely can be provided in the netCDF as a variable or header information as well. Furthermore, many aspects and parameters that might be relevant for describing the data are often included in the modelling process (either in the input or the model run), but are not usually considered in the final model output. It would therefore be advisable to implement them in the post-processing stage. An idea for the provision of this information with Python and more details on this topic are described in Appendix C.3.2.

In wind tunnel modelling, converting wind tunnel data to netCDF is technically possible. Due to differences in layout and data descriptions in wind tunnel model datasets, the applicability of the post-processor wt2atmodat needs to be adapted to each dataset. Nevertheless, ideas for a concept to provide standardised wind tunnel data could be derived. Difficulties arise in the proper data description, which ensures that a data user can easily understand the data. Currently, since wind tunnel data is not standardised and provided in different file formats (e.g. Excel or ASCII files) a data user must assess and understand the data in advance and develop their own data-specific processing routines. By providing the data as netCDF files, automatic processing becomes possible and, thus, could decrease the awareness on the speciality of the data. Nevertheless, the netCDF data format seems to be a generally usable way of storing actual data values. Thanks to its self-describing capabilities, any required information can be theoretically assigned to the data. However, this work does not fully consider the provision of model-domain-relevant aspects. The provision of information about the model domain, building and topography layouts or wind tunnel specifics like the positions of roughness elements for the creation of the desired boundary conditions needs further investigation. Furthermore, the handling of netCDF when transformed into real-world conditions, such as scaling wind tunnel data to different atmospheric conditions, spatial and temporal scales or even the provision as georeferenced data was not tested in this work and requires further investigation. Furthermore, wt2atmodat and the provided data standard concept is only based on and applied to datasets from the EWTL. It will also require testing and evaluation on other datasets as well as by other wind tunnel model data producers and data users. As there are various possibilities for describing wind tunnel model data, a universally applicable wind tunnel data standard must be thoroughly discussed, tested, and accepted by the relevant modelling community. This concept can be used as a starting point for such further experimentations and discussions.

6.2 Outlook

Since the publication of research datasets is becoming more and more important, data standards can guide data producers to prepare well-described, uniform and re-usable, FAIR datasets. Developing such a data standard will require a community effort and involves development, testing, discussions and, most importantly, the consequent application of standardisation to ensure a widespread adoption of the practices. Communities that implement standard, also further develop on these established structure. For example, research project from communities that work with netCDF and the CF conventions, often develop their own project specific standards, which are based on already established structures (e.g. Scherer et al. (2019); Lammert et al. (2017); Haarsma et al. (2016). For the application of standards, software tools are required. Ideally, they are universally applicable to modify any netCDF files, such as CF_python (Hassell et al., 2017) or CFbuild (Jones et al., 2023), or data checkers such as the AtMoDat Data Checker (Kretzschmar et al., 2022). Those are designed to aid data producers with the creation of well-described CF conform netCDF files regardless of their discipline. Smaller tools, like nc2atmodat and wt2atmodat, which are mainly adapted to specific models, might be also used by other researchers to gain inspiration for similar solutions in other models. Such software tools are especially useful for data producers, who do not have the time to understand and acquire the concepts and requirements for data standardisation (Jones et al., 2023). The data producer does not need to reinvent this process by themself over and over any more.

With the upcoming of artificial intelligence, it might become even more easier to standardise model data. And the provision of information to become well-described data will be more important, since data must be processable and understandable by human data users and AI. Furthermore, there will be still a need for the initial development of certain solutions (scripts, workflow), which are used as templates in other applications. Therefore, defining standards and producing standardised data is crucial. Maintaining such data standardisation workflow, however, takes time and effort, and is in practice often overlooked.

As already mentioned, the next steps in the development of this topic will require the active involvement of the ORM community. A key upcoming task will be the standardisation of typical ORM variables for both numerical models and wind tunnels and will require input, discussion and collaboration across the community. Hereby, different model developers need to collect typical ORM variables from different models, and need to agree on unified definitions and descriptions to ensure an unambiguous use and understanding of ORM variables across different models. Addressing these issues can happen in communities such as GURME, WMO or ICUC to increase the awareness and international establishment of such ORM-specific standardisation efforts.

Within the wind tunnel community, the proposed concept for standardised netCDF wind tunnel data has to be discussed. Standardised variable names and attributes, as well as consistent formatting and data structures, are feasible and help data users to access and understand their model results more easily.

And if the community concludes that netCDF is not suited for the needs of wind tunnel model data, the developed description concepts could still be adapted to currently used methods. One way to involve physical modellers in the development of a wind tunnel standardisation could be to conduct a survey similar to the one in Section 3. Since a variety of physical modelling groups and datasets exist, a survey targeting wind tunnel model data specifics might be a good opportunity to find the common grounds and differences to tackle future standardisation approaches. The survey could query information about the amount and types of output files, data formats, provided information or metadata used in different wind tunnel modelling groups. Furthermore, it could collect experiences or opinions from wind tunnel model data producer and user on data handling, processing and their suggestions for generalisation and standardisation concepts. This could be a good starting point for drawing conclusions about a potential wind tunnel standard.

Appendices

A Supporting information to Chapter 2

In this work, the model results from the obstacle resolving numerical microscale model MITRAS are used. MITRAS, short for microscale transport and stream model, is a RANS model, that provides the possibility to resolve obstacles in their model domain and investigate atmospheric processes in urban areas (Schlünzen et al., 2003; Salim et al., 2018; Schlünzen et al., 2018a,b).

The process of running MITRAS and generating output data is shown in Figure A.1 and Table A.1 contains a comprehension of each input and output file.

Table A.1: Overview of a selection of the files for MITRAS model simulation and their functions

Preamble	Type	Description
PK	ascii files	used as input files for the pre-processor to create the model domain. The file is an ascii grid file. The header must contain information about the total number of rows and columns, geographical coordinates of the lower left corner either in utm or Gauss-Kruger (XLL, YLL), cell size and a nodata value.
GA	ascii file	Output file of model pre-processor GRIMASK generated from the input PK files and domain settings assigned in GRIMASK. It described the domain and contains information about the grid dimensions, grid sizes sizes, coordinates, terrain heights and surface cover classes
BM	binary	output file from the model pre-processor GRIMASK. It contains obstacles mask that is needed for 3D model run with buildings.
B1A	binary	Output from the 1D mitras model run, contains the model settings and starting conditions, e.g. the 1D profile for the 3D Model run. It is needed to run 3D model simulations.
BP	binary	Output files of the 3D model run.
.nc	binary, netCDF	Output files of the 3D model run, translated into the netCDF file format.

The application of the data standardisation in this work will mainly focus on the final product, the netCDF file and the adapataions in the post-processing routine m2cdf. Aspects that are relevant during the pre-processing (GRIMASK) or from the input file (GA file) will play a role in the provision of proper georeferencing information of the finalised dataset and need to be implemented in the future model development. To do this task, it can be expected, that the automatically provision of this information require greater adaptations in the model input and output structure, that would also affect the sister model METRAS.

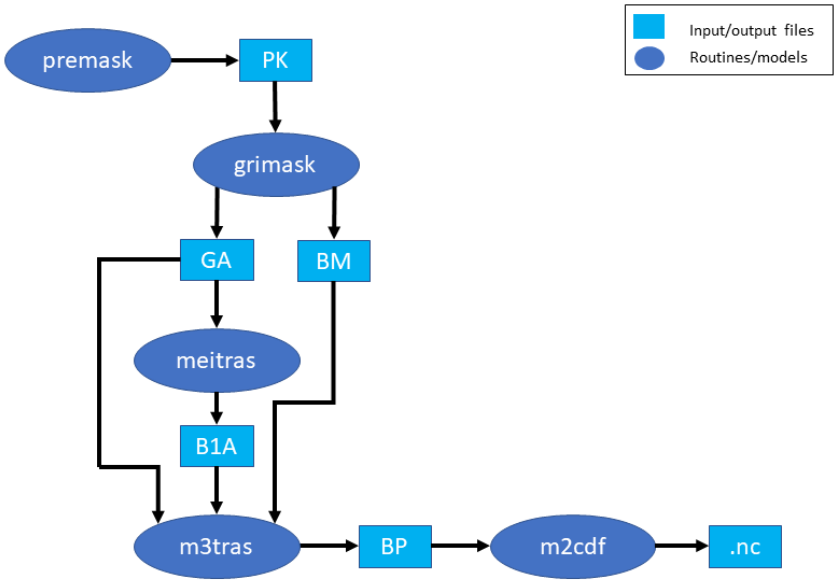

Figure A.1: Workflow of a basic MITRAS simulation run, including the components that steer the model routines (meitras (1D), m3tras (3D)), pre-processing (PREMASK, GRIMASK), post-processing (m2cdf) and input/output files.

Furthermore, further information are written into various files for control or restart purposes. In Schlünzen et al. (2018b) are such information listed and explained.

B Supporting information to Chapter 3

B.1 Statements on submitted manuscript

Data Availability

The survey results are published at Zenodo (Voss, 2023b) and available via https://doi.org/10.5281/zenodo.8211274.

Acknowledgements

This work would not have been possible without the answers to the survey by the 14 participants. Many thanks to all who answered the survey; their answers contributed to this investigation. This work contributes to and is partly funded by the Cluster of Excellence 'CLICCS - Climate, Climatic Change, and Society', and is a contribution to the Center for Earth System Research and Sustainability (CEN) of Universität Hamburg". CLICCS is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2037 'CLICCS - Climate, Climatic Change, and Society' – Project Number: 390683824, contribution to the Center for Earth System Research and Sustainability (CEN) of Universität Hamburg". This work uses results, contributes to and was partly funded by project "AtMoDat", funded by the Federal Ministry of Education and Research under the funding number 16QK02C. Responsibility for the content of this publication lies with the authors.

C Supporting information to Chapter 4

C.1 Statements on submitted manuscript

Data availability

The datasets V23 (Voss, 2023a) and F24 (Ferner et al., 2024) are published via the World Data Center for Climate (WDCC).

Code availablility

The *m2cdf* source code is distributed upon request under the terms of a user agreement with the Mesoscale and Microscale Modeling (MeMi) working group at the Meteorological Institute, University of Hamburg (metras@uni-hamburg.de, https//www.mi.uni-hamburg.de/memi). The program *nc2atmodat* is published and available via GitHub (Voss, 2025a). The ATMODAT Standard Checker is available via GitHub (Kretzschmar et al., 2022) as well.

Acknowledgement

This work contributes to and uses results of project "AtMoDat", funded by the Federal Ministry of Education and Research under the funding number 16QK02C. The creation of the dataset including precipitation was financed within the framework of the Helmholtz Institute for Climate Service Science (HICSS), a cooperation between Climate Service Center Germany (GERICS) and Universität Hamburg, Germany, and the corresponding work conducted as part of the WINTER project (Investigating climate change related impacts on the urban winter climate of Hamburg). This work contributes to and is partly funded by the Cluster of Excellence 'CLICCS - Climate, Climatic Change, and Society', and is a contribution to the Center for Earth System Research and Sustainability (CEN) of Universität Hamburg. CLICCS is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2037 'CLICCS - Climate, Climatic Change, and Society' – Project Number: 390683824, contribution to the Center for Earth System Research and Sustainability (CEN) of Universität Hamburg. Responsibility for the content of this publication lies with the authors.

C.2 Representation of a building surface variable in the model code

In MITRAS, obstacles are explicitly resolved using the mask method (Briscolini and Santangelo, 1989; Salim et al., 2018). There, a grid cell is either an atmospheric grid cell surrounded by atmospheric grid cells, an obstacle grid cell, or an atmospheric grid cell adjacent to an obstacle grid cell. The number of obstacle adjacent grid cells is given with `nsurfcells` ($N_{surfcells}$).

An example code for the calculation of an obstacle surface variable (`var_building`) is provided in Figure C.1. The first do loop runs through every obstacle adjacent grid cell. The location in the Cartesian grid is provided by `nxobst`, `nyobst`, and `nzobst`. In order to address an obstacle surface variable, it is not necessary to go through every grid cell and check whether it is obstacle adjacent. Instead, the variable is defined for every obstacle adjacent grid cell, which reduces the number of iterations in loops and increases the efficiency of the algorithm.

The number of obstacle faces at an obstacle adjacent cell is given with `nsurfcount`. Up to three obstacles faces can be represented at each obstacle adjacent grid cell. Again, the efficiency of the algorithm is increased by only including `nsurfcount` iterations instead of three regardless of surface count (second do loop in Figure C.1).

The information whether a wall is facing east, west, north, etc. is contained in `nsurftype` and `nsurfdir` as shown in Table C.1. Assuming for example an obstacle adjacent grid cell with a wall north and a wall below, `nsurfcount` would be 2, `nsurftype` and `nsurfdir` 2 and 1 for the first wall and 3 and -1 for the second wall. Note, that even though `nsurftype`, `nsurfdir`, and `var_building` have the same spatial dimensions (`nsurfcells`×3), the last dimension has a different meaning in the obstacle surface variable.

In order to calculate *var_building*, all necessary information on the type of wall and the location of the grid cell are provided. Unfortunately, there is still an if statement in the do loops, which slows down the algorithm. This statement is necessary, as different calculations are performed depending on the type of wall (`foo1`, `foo2`, `foo3` in Figure C.1). However, an additional if statement to differentiate between wall directions is avoided by utilising the fact, that `nsurfdir` is either one or minus one.

Table C.1: Representation of obstacle face directions in MITRAS.

wall	`nsurftype`	`nsurfdir`
east	1	1
west	1	-1
north	2	1
south	2	-1
top	3	1
bottom	3	-1

```fortran
DO jb=1,nsurfcells
   ji                   = nxobst(jb)
   jj                   = nyobst(jb)
   jk                   = nzobst(jb)
   DO jm=1,nsurfcount(jb)
      jr                = nsurftype(jb,jm)
      jd                = nsurfdir(jb,jm)
      IF (jr .EQ. 1_ni) THEN ! east(jd=1) / west(jd=-1)
         var_building(jb,jr) =    (1 + jd) / 2 * foo1(jk   ,jj   ,ji-1) &
                     &          + (1 - jd) / 2 * foo1(jk   ,jj   ,ji   )
      ELSEIF (jr .EQ. 2_ni) THEN ! north(jd=1) / south(jd=-1)
         var_building(jb,jr) =    (1 + jd) / 2 * foo2(jk   ,jj-1,ji   ) &
                     &          + (1 - jd) / 2 * foo2(jk   ,jj   ,ji   )
      ELSEIF (jr .EQ. 3_ni) THEN ! top(jd=1) / bottom(jd=-1)
         var_building(jb,jr) =    (1 + jd) / 2 * foo3(jk-1,jj   ,ji   ) &
                     &          + (1 - jd) / 2 * foo3(jk   ,jj   ,ji   )
```

Figure C.1: Example Fortran95 code for the calculation a building surface variable var_building.

C.3 Further suggestions and ideas to standardise MITRAS model data

C.3.1 Suggestions for ORM variable standard names

In obstacle-resolving modelling, there are discipline-specific variables that have not yet been standardised. As a result, there are no corresponding entries in the CF conventions, and the naming and definition may vary from model to model. When adding CF standard names to MITRAS variables, it became apparent that many variables could not always be clearly assigned to an existing standard name. Furthermore, a comparison between PALM-4U and MITRAS variable names was done to determine, which variables are typical for ORMs and if they are suited to become standardised (Table C.2).

One example for a typical ORM variable: both models provide a variable named "building_height', which is not available in the standard name table of the CF convention. This is likely used in other ORMs as well. Another example, the variables "land_water_temperature" (PALM-4U) and "water_temperature" (MITRAS) seem to describe the same property. it can be assumed that both variables denote the temperature of freshwater bodies like rivers or lakes. However, the current CF standard name table only provides a variety of standard names for water temperature variables in ocean or saline water bodies. None seem to be applicable for freshwater. It seems suitable to define a new standard name for freshwater bodies as"land_water_temperature", which is similarly constructed after the example "sea_water_temperature". This new name would be a candidate to be proposed as a new standard name to the CF conventions.

On the other side, some variables, which seem to have a similar counterpart in the other model variable list, it is rather unclear to derive only from the name, if they were describing the same properties. For example, PALM-4U provides a variable named "building_id" and MITRAS a variable named "building_mask". Both variables describe some building characteristics, but it is unclear whether they are really related. Another example would be the variable "anthropogenic_heat_flux" (MITRAS), whereas PALM-4U provides a variable "emission_of_anthropogenic_heat_from_buildings". These variables look like they might be connected, whereas the units do not match.

Table C.2: Examples of ORM variables used in the models PALM-4U (PA), MITRAS (MI) and (possible) related CF standard names (CF). If a variable has no matching counterpart in the CF convention, CF indicates a related standard name that is remotely related.

Variable Name	**Variable per model and possible related CF Standard Name**	**Unit**
anthropogenic_heat_flux	MI: anthropogenic_heat_flux	W m^{-2}
	PA: emission_of_anthropogenic_heat_from_buildings	K s^{-1}
	CF: surface_upward_heat_flux_due_to_anthropogenic_energy_consumption	W m^{-2}
building	MI: building_mask	1
	PA: building_id	1
	CF: –	–
building_height	MI: building_height	m
	PA: building_height	m
	CF: surface_altitude	m
building_surface_temperature	MI: building_surface_temperature	K
	PA: –	–
	CF: surface_temperature	K
cooling_rate	MI: cooling_rate_due_to_longwave_radiation	K d^{-1}
	PA: nocturnal_cooling	K h^{-1}
	CF: –	–
emission_of_anthropogenic_heat_from_buildings	MI: anthropogenic_heat_flux	W m^{-2}
	PA: emission_of_anthropogenic_heat_from_buildings	K s^{-1}
	CF: –	–
heat_conduction_through_wall	MI: heat_conduction_through_wall	W m^{-2}
	PA: –	–
	CF: –	–
indoor_temperature	MI: –	–
	PA: indoor_temperature	$^\circ$C
	CF: –	–
land_water_temperature	MI: water_temperature	K
	PA: land_water_temperature	K
	CF: sea_water_temperature	K
mean_radiant_temperature	MI: mean_radiant_temperature	$^\circ$C
	PA: mean_radiant_temperature	$^\circ$C
	CF: –	–
turbulent_mixing_lenght_in_air	MI: mixing_length	m
	PA: mixing_length	m
	CF: turbulent_mixing_length_of_sea_water	m
shortwave_heating_rate	MI: heating_rate_due_to_shortwave_radiation	K d^{-1}
	PA: shortwave_heating_rate	K h^{-1}
	CF: tendency_of_air_temperature_due_to_longwave_heating	K s^{-1}

*) Variables are included in CF standard name table.

Based on the comparison, a collection of potentially ORM standard names with corresponding descriptions and units were defined (Table C.3). This collection provides suggestions for new standard names, that can be proposed to the CF Standard Name table. Hereby, the definition of new standard names follows two approaches.

One way was to derive new standard names from an already existing standard names in the CF Standard Name table. For example, as mentioned, the CF Standard Name Table includes a standard name for water surface temperatures, that seems only applicable for sea water variables. A possible new standard name for freshwater temperature can be created similarly to the existing standard name ("sea_water_temperature") as "land_water_temperature" with the description mentioning freshwater or land water explicitly.

The second way derives new standard names on the long names provided by either PALM-4U (Scherer et al., 2022), MITRAS, or a merged name if both models use similar long names. This method was mostly applied if no related standard names were available in the CF Standard Name table. Then, these newly derived standard names were then constructed after the CF standard names guidelines[26].

[26]https://cfconventions.org/Data/cf-standard-names/docs/guidelines.html

Table C.3: Possible suggestions for "standard names" for ORM specific variables. Descriptions are partially based on existing CF convention "standard names", if similar variables exist. *) Variables are included in CF standard name table.

Standard name	**Standard name description**	**Unit**
building_height	Height of the building roof in the model domain.	m
building_surface_temperature	The obstacle surface temperature represents the temperature on the surface skin of an obstacle in the model domain at the air-obstacle interface. This quantity is subject to a large potential diurnal cycle including cool skin layer effects (especially at night under clear skies and low wind speed conditions) and warm layer effects in the daytime.	K
land_water_temperature	land water temperature is the surface temperature of the water on land. It is the temperature of water near the surface. There are standard names for sea_surface_temperature, sea_surface_skin_temperature, sea_surface_subskin_temperature and sea_surface_foundation_temperature which can be used to describe data located at the specified surfaces.	K
mean_radiant_temperature	The mean radiant temperature (MRT) is a human biometeorology parameter used to assess the linkages between the outdoor environment and human well-being. Thermal comfort indices describe how the human body experiences atmospheric conditions, specifically air temperature, humidity, wind and radiation. The mean radiant temperature ($°C$) is a numerical representation of how human beings experience radiation. The mean radiant temperature (MRT) is based on the principle that the net exchange of radiant energy between objects is approximately proportional to their temperature difference multiplied by their ability to emit and absorb heat (emissivity). In this context the MRT applies to a person in an outdoor environment and is a function of the direct, diffuse and reflected thermal and solar radiation they experience. (source: https://climate-adapt.eea.europa.eu/metadata/indicators/thermal-comfort-indices-mean-radiant-temperature-1979-2019#)	$°C$
turbulent_mixing_length_in_air	Turbulent mixing length is used in models to describe the average distance over which a fluid parcel can travel while retaining properties that allow the parcel to be distinguished from its immediate environment. "Turbulent mixing" means chaotic fluctuations of the fluid flow.	m
obstacle_surface_temperature	The obstacle surface temperature represents the temperature on the surface skin of an obstacle in the model domain at the air-obstacle interface. This quantity is subject to a large potential diurnal cycle including cool skin layer effects (especially at night under clear skies and low wind speed conditions) and warm layer effects in the daytime.	K

*) Variables are included in CF standard name table.

C.3.2 Ideas to implement the coordinates and surface cover class information

C.3.2.1 Adding geographic model coordinates to MITRAS via m2cdf One part of the MITRAS model input is the so-called GA-File (Figure C.2), that defines the model domain by providing the used surface cover classes, cell numbering, cell size or extent and coordinates. It is generated by the model pre-processor GRIMASk and written as an ASCII file.

```
1  'number of grid points:   nx3 =   60 ', nx2 =' 792 ', nx1 =' 740
2  'ref. point [dd.mmss]: latitude ='   53.3225 ',longitude'    10.0020
3  'angle  of  grid  rotation  =' 0.0000
4  'Number of Surface Cover Classes ='    5
5  'vector  Z - points  :'     -5.00    0.00    5.00   10.00   15.00
6                              20.00   25.00   30.00   35.00   40.00
7                                -       -       -       -       -
8                             235.00  252.80  273.30  296.90  324.00
9                             355.20  391.10  432.40  432.40  432.40
10
11 yxmin=   -941.880 yymin=  -1021.880
12   ii  ij    ydx    ydy   zsurf   1100  1222  2220  3100  6005       x        y
13    0   0   5.00   5.00    2.72   0.00  1.00  0.00  0.00  0.00  -941.88 -1021.88
14    1   0   5.00   5.00    2.72   0.00  1.00  0.00  0.00  0.00  -936.88 -1021.88
15    2   0   5.00   5.00    2.72   0.00  1.00  0.00  0.00  0.00  -931.88 -1021.88
16    3   0   5.00   5.00    2.72   0.00  1.00  0.00  0.00  0.00  -926.88 -1021.88
17    4   0   5.00   5.00    2.72   0.00  1.00  0.00  0.00  0.00  -921.88 -1021.88
18    -   -    -      -       -      -     -     -     -     -        -        -
19  738 793   5.00   5.00    3.83   1.00  0.00  0.00  0.00  0.00   931.88  1026.88
20  739 793   5.00   5.00    3.83   1.00  0.00  0.00  0.00  0.00   936.88  1026.88
21  740 793   5.00   5.00    3.83   1.00  0.00  0.00  0.00  0.00   941.88  1026.88
22  741 793   5.00   5.00    3.83   1.00  0.00  0.00  0.00  0.00   946.88  1026.88
```

Figure C.2: Example for a GA File (shortened). Appending two new columns after Y with XUTM and YUTM values could help to transfer the relevant geographical information through the model to the output. Requires then adaptations to process these additional columns in the pre-processor and input routines of both models METRAS and MITRAS.

In the first line, the number of total grid point in each spatial direction are defined (nx3 = z, nx2=y, nx1=x), which are associated with the indexes ii (x-direction) and ij (y-direction) (Line 12). Line 2 contains geographical coordinates to locate the reference point on the earth surface. This is provided by the model user. Line 3 contains the rotation angle of the model domain. Line 4 contains the number of surface cover classes, which are determined by the occurance of the surface cover class identification number (4-digit numbers in Line 12), provided by the input file (PK-file) for the surface cover classes. In this example, 5 surface cover classes are used (1100: water, 1222: water, 2220: concrete, 3100: vegetation, 6005: mixed landuse). Line 5 to 9 contain the position for the height of each z-vector grid point. Line 11 contains the coordinates for the lower left corner in meters. Line 12 is the header for the following block.

It contains the grid cell index (ii, for the x-direction, ij for the y-directon.), the grid size extent in metres in the respective direction (x = ydx, y= ydy), the elevation of the surface at this grid point in metres, the index for each surface cover class (4-digit number) and the metre-coordinate for each direction in relation to the reference point. The reference point is located at position x=0, y=0. After line 12 the values for each grid points are provided. Note, that the numbers in the surface-cover class digit sections are the fraction of each class per cell. The sum of each line in this block shall not exceed the value 1.

In the METRAS model GA-File, x and y contain geographical coordinates (latitude, Longitude) instead of metres. An adaptation of this file for the implementation of e.g UTM or Gauss-Kruger coordinates could be by extending the table with the respective coordinates for each direction and each grid cell, which are then carried through the simulation and included in the final model output. This extension would require changes in the writing and reading of the GA-File in both model pre-processor (MITRAS: GRIMASK and METRAS, GRITOP), to be able to be read in the routine meitras.

C.3.2.2 Adding surface cover classes information to MITRAS via m2cdf Since the information about the used surface cover class are stored in the model input and processed in the model initialisation, the actual meaning to each class is not handed down to the final output yet.

A quick solution to provide the model-specific descriptions into the netCDF file for a better understanding of each category is provided the following Listings 1 – 3.

```
# import modules for processing:
#-------------------------------------------------------
import xarray as xr
import pandas as pd
import numpy as np

# Open files
#-------------------------------------------------------
nc_raw = xr.open_dataset('/path/to/ncfile/MITRAS_result.nc')
GA = '/path/to/gafile/GA_MITRAs'

# Call function
#-------------------------------------------------------
GA_data = READ_GA(GA)

# extract ssc numbers from header and transform scc from string to float)
#-------------------------------------------------------
scc_values = list(map(float, list(GA_data.columns)[5:-2]))
# get scc meanings as string for cf
flag_meanings = ' '.join([*map(scc_class_dict.get, scc_values)])
# assign values and meanings to variables in netCDF
raw['main_index'].attrs['flag_values'] = list(np.arange(0, len(scc_values), 1, dtype='
    float64')) #ist(range(len(values)))
raw['main_index'].attrs['flag_meanings'] = flag_meanings

# Save file
#-------------------------------------------------------
output.to_netcdf('/path/to/file/MITRAS_results_out.nc')
```

Listing 1: Python code Snippet for assigning surface cover class descriptions in a MITRAS netCDF file.

```
def READ_GA(FILE):
    read_GA = open(FILE, 'r')
    GA = list(read_GA)
    # Selecting lines with header
    for i in range(len(GA)):
        temp = GA[i].strip().split()
    # locate the header
        if temp[0] == 'ii':
            caption = i
            break
    header = GA[0:caption+1]
    top = np.genfromtxt(FILE, skip_header=len(header))
    GITTER = pd.DataFrame(data=top, columns=GA[caption].strip().split())
    return(GITTER)
```

Listing 2: Python code Snippet for assigning surface cover class descriptions in a MITRAS netCDF file. Function for reading the GA-file and extraction of surface cover class ID number

```python
scc_class_dict = {
    1000. : 'water',
    1100. : 'fresh_water_stationary',
    1222. : 'fresh_water_dynamic',
    1300. : 'salt_water',
    1411. : 'mudflats',
    2106. : 'sand',
    2107. : 'gravel',
    2108. : 'rock',
    2122. : 'sanddune_with_grass',
    2123. : 'sanddune_with_spare_vegetation',
    2220. : 'asphalt',
    2230. : 'concrete',
    2240. : 'brick',
    2250. : 'steel',
    2712. : 'wet_bushes',
    2715. : 'wet_bare_ground',
    2911. : 'salt_pit',
    3100. : 'short_grass',
    3104. : 'wet_short_grass',
    3138. : 'long_grass',
    3148. : 'wet_long_grass',
    3500. : 'cropland',
    3830. : 'irrigated_cropland',
    3863. : 'cropland_on_sandy_soil',
    4102. : 'woody_savannah',
    4103. : 'savannah',
    4210. : 'heath',
    4211. : 'heath_on_sandy_soil',
    4314. : 'dry_bushes',
    4340. : 'short_bushes',
    5100. : 'deciduous_forest',
    5200. : 'coniferous_forest',
    5213. : 'wet_coniferous_forest',
    5300. : 'mixed_forest',
    5358. : 'dry_mixed_forest',
    5656. : 'wet_mixed_forest',
    5811. : 'forest_and_bushes',
    6000. : 'sealed_urban',
    6005. : 'urban_buildings_below_11m',
    6006. : 'urban_buildings_above_11m',
    7010. : 'mixed_landuse'
}
```

Listing 3: Python code Snippet for assigning surface cover class descriptions in a MITRAS netCDF file. Surface cover classes list, taken from from Schlünzen et al. (2018b). Dictionary with ID number and name.

The possible outcome of a variable is provided in Listing 4

```
int main_index(j, i) ;
    main_index:_FillValue = -999 ;
    main_index:long_name = "main_land_use_class_index" ;
    main_index:standard_name = "area_type" ;
    main_index:structure = "0000" ;
    main_index:units = "1" ;
    main_index:cell_methods = "j: i: mean" ;
    main_index:flag_values = 0., 1., 2., 3., 4., 5., 6., 7., 8., 9., 10., 11., 12. ;
    main_index:flag_meanings = "fresh_water_stationary fresh_water_dynamic asphalt
    concrete brick steel wet_bushes short_grass cropland short_bushes deciduous_forest
    sealed_urban mixed_landuse" ;
    main_index:coordinates = "lon lat" ;
```

Listing 4: Example netCDF variable header with assigned surface cover class information.

D Supporting information to Chapter 5

D.1 Statements on submitted manuscript

Acknowledgements

The author would like to thank Simon Michel, Sylvio Freitas and Frank Harms for providing the wind tunnel model datasets used in this work. Bernd Leitl and K. Heinke Schlünzen are also thanked for supervising, conceptualising and discussing the work. This work contributes to and is partly funded by the Cluster of Excellence 'CLICCS - Climate, Climatic Change, and Society', and is a contribution to the Center for Earth System Research and Sustainability (CEN) of Universität Hamburg". CLICCS is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2037 'CLICCS - Climate, Climatic Change, and Society' – Project Number: 390683824, contribution to the Center for Earth System Research and Sustainability (CEN) of Universität Hamburg". This work contributes to and uses results of project "AtMoDat", funded by the Federal Ministry of Education and Research under the funding number 16QK02C. Responsibility for the content of this publication lies with the authors.

Code Availablility

The program wt2atmodat is available via github Voss (2025b) under https://github.com/VossV/wt2atmodat/tree/main. The ATMODAT Data checker is available via github (Kretzschmar et al., 2022) under https://github.com/AtMoDat/atmodat_data_checker/tree/v1.3.1

Data Availability

Michel et al. (2023) are published and available via https://www.fdr.uni-hamburg.de/record/11779. The datasets generated by Harms et al. (2012) and Freitas et al. (2021) are available upon request.

D.2 Detailed attribute descriptions for wind tunnel netCDF variables

In the following paragraphs, detailed information on the recommended attributes for the descriptions of wind tunnel variables are provided.

Attribute: long_name A `long name` is an attribute to provide a self-descriptive and self-explaining name to a netCDf variable (Eaton et al., 2024). It allows to be more precise than the variable name itself. Ideally, a long-name should be constructed like a plot title for the variable. Besides `long names`, the CF conventions also provide so-called `standard names`, which are standardised variable names from a controlled vocabulary list with a certain fixed defined naming structure, description and unit. A `standard name` attribute should be only assigned to a variable, if the variable represents the particular quantity available in the standard name table[27]. If applicable, adding a `standard name` with the corresponding standard name entry is preferable, but due to the lack of wind tunnel variables in the CF conventions, it is only recommended if fitting names are available. New standard names can be requested and proposed to the CF convention community. For wind tunnel variables, it is therefore currently recommended to add only the `long name` attribute with a self-describing variable name. In the case that if the data is not provided as a netCDF file, but rather as e.g. text files: variable names should also be assigned with a `long name` to ensure a similar naming and meaning through different datasets.

Attribute: units Each dimensional variable (e.g. full- or model-scale variables) should be assigned with a `unit` attribute, that holds the corresponding unit. Units should be provided as SI units. For dimensionless variables, the unit entry should be either set to 1 or the attribute should not be provided (Eaton et al., 2024). In the example datasets, the units for dimensionless variables were provided by a bracket with a dash symbol ("[-]" or "(-)").

Attribute: comment The attribute `comment` can be used to provide any additional information about a variable that is not sufficiently covered by the `long name`. Such further information about variables would be otherwise provided in a README section of a dataset. It can be stored in the `comment` attribute and helps to make a variable sufficiently self-explanatory. For example, dimensionless concentration "conc_DL" is also known as "$C*$" in the literature. Such information can be added via a comment attribute to the variable.

Attribute: conversion Wind tunnel data can be provided in three states: dimensionless, full-scale and model-scale; hence, a wind tunnel variable can be transformed to all three states. Therefore, the formula or method for the transformation must be provided. For dimensionless variables, it is recommended to provide the information how the data was made dimensionless. Providing the data as netCDF file, dimensionless variables can be assigned attributes, that contain the formulas for the transformation. It

[27] https://cfconventions.org/Data/cf-standard-names/current/build/cf-standard-name-table.html

would be recommended to provide two attributes with formulas to transform the data from model-scale to dimensionless quantity (attribute: `conversion_DL`) and the formula to re-scale the values to full-scale (attribute: `conversion_FS`). These conversion formulas are crucial for the correct use and handling of wind tunnel model data.

Note: In the CF convention, an attribute named `formula` is used as a method to describe the conversion to recalculate a quantity into another pressure coordinate system. For example, for the transformation to sigma z-coordinates, CF provides with the `formula` attribute the possibility to store the variable names, that are required to calculate these coordinates from variables in the data set (Eaton et al., 2024). The actual conversion formula is hardcoded in the CF convention. To describe these variables properly, an attribute called 'formula_terms' can be provided, which are then used in the formula (Eaton et al. (2024) see. Appendix D Parametric Vertical Coordinates). `formula` is used differently to the proposed use of attribute `conversion`. However, it seem to be an option to handle this conversion for wind tunnel variables in the future. For now, as no wind tunnel conversion variables are available in the CF conventions, the attribute `conversion` carries this formula.

Attribute: ancillary_variables Measurement data contain measurement uncertainties; hence the provision of this information is necessary. In a netCDF file, information about uncertainties of a quantity or measurement can be stored in additional variables, which then can be linked to a variable with the `ancillary variable` attribute, indicating the connection between those variables (Eaton et al., 2022) (see Figure D.1). In the example dataset of Freitas et al. (2021), they provide uncertainties for each quantity in a separate file, which could be then stored together in the same netCDf file. However, the dataset of Michel (2023) provides an overall general information about the uncertainties in the measurement dataset. In this case, providing this information can be done via the global attributes for the dataset.

```
float Mag_DL(index) ;
    Mag_DL:long_name = "dimensionless 2D wind magnitude (UW plane)";
    Mag_DL:ancillary_variables = "dMag_DL";
    Mag_DL:comment = "Mag/Uref = square root[(u/Uref)^2 + (w/Uref)^2]";

float dMag_DL(index) ;
    dMag_DL:long_name = "Uncertainties of the dimensionless 2D wind magnitude (UW plane)
    ";
```

Figure D.1: Example for the use of the `ancillary_variable` in a wind tunnel netCDF file.

Attribute: flags Based on the CF Convention Chapter 3.5 (Eaton et al., 2022), the attribute `Flag` can be used to describe properties with status information like the functionality of measurement devices. Such status information occurred in a part of the dataset of Harms et al. (2012). An example how to construct the flag variable for wind tunnel is depicted in Figure D.2. Flags are connected with the `ancillary variables`.

Attibute: FillValues If the data contains missing values, e.g. marked with a certain numerical value, it can be indicated by the attribute `_Fillvalue`, as shown in Figure 5.2.

Attibute: bounds (optional) For gridded data, the attribute `bounds` is used to describe the extent of a grid cell by providing the extent and location of the boundary of a grid cell. This is used especially in numerical model data to describe the extent of each grid cell and for statistical measures (Eaton et al., 2022). For wind tunnel model data, it might be suitable to describe the extent of the measurement volume of each device at each measurement point. If wind tunnel model data should be stored as a 3D data and in a grid-like format, the use of this attribute would play an important role to precisely describe the actual boundaries of the measurement point grid cell.

```
dimensions:
    puff_number = 264 ;
variables:
    string quality_flag(puff_number) ;
        quality_flag:long_name = "quality_flag" ;
        quality_flag:valid_range = "1,8" ;
        quality_flag:flag_values = "0, 1, 2, 3, 4, 5, 6, 7, 8" ;
        quality_flag:flag_meanings = "all_puff_parameters_are_correct
                                    pc_is_exceeding_the_measurement_limit
                                    the_dosage_value_is_incorrect
                                    the_pt_value_is_incorrect
                                    the_at_is_too_short
                                    the_asct_value_is_incorrect
                                    the_dsct_value_is_incorrect
                                    the_lt_value_is_incorrect
                                    the_duration_value_is_incorrect" ;
```

Figure D.2: Example for the use of the flag attribute for wind tunnel variables.

D.3 Collection of wind tunnel variable names

Table D.1 – D.3 contains an overview of the collected variables with possible standardised variable names, units and descriptions, which can be used as entries for the comment attribute. The tables are divided into variables for flow measurements (Table D.1), variables for concentration measurements (Table D.2), and coordinate and reference variables (Table D.3). The tables contain only a fraction of possible variable names. Fixed standard names and definitions can be formed based on this collection. Furthermore, Table D.4 contains variable names collected from the datasets mentioned in Table 5.1, that are paired due to naming inconsistencies, such as alternative names or ambiguously used for different properties.

Table D.1: Suggestion for standardised wind tunnel variables. Variables are written output of the measurements and processing program described in Fischer (2011) and taken from the datasets in 5.1. DL: Dimensionless; formula for the calculation of the dimensionless quantity.

netCDF	long name	Comment	Unit
x	x-position of the measurement point	–	m
y	y-position of the measurement point	–	m
z	z-position of the measurement point	–	m
u	lateral x-wind component	DL: u/U_{ref}	$m\,s^{-1}$
v	longitudinal y-wind component	DL: V/U_{ref}	$m\,s^{-1}$
w	vertical z-wind component	DL: W/U_{ref}	$m\,s^{-1}$

netCDF	long name	Comment	Unit
mag	dimensionless uw-wind magnitude	dimensionless 2D wind magnitude (UW plane), formula: $Mag/Uref = \sqrt{(u/Uref)2 + (w/Uref)2}$	–
ustd	x-wind standard deviations	DL: $u\_std/Uref$	$m\,s^{-1}$
vstd	y-wind standard deviations	DL: $v\_std/Uref$	$m\,s^{-1}$
wstd	z-wind standard deviations	DL: $w\_std/Uref$	$m\,s^{-1}$
urms	x-wind root mean square error	DL: $u\_rms/Uref$	$m\,s^{-1}$
vrms	y-wind root mean square error	DL: $v\_rms/Uref$	$m\,s^{-1}$
wrms	z-wind root mean square error	DL: $w\_rms/Uref$	$m\,s^{-1}$
u-v-	horizontal momentum flux	Reynolds stress tensor, DL: $u'v'/Uref^2$	$m^2\,s^{-2}$
u-w-	vertical momentum flux	Reynolds stress tensor, DL: $u'w'/Uref^2$	$m^2\,s^{-2}$
v-w-	cross-flow momentum flux	Reynolds stress tensor, DL: $v'w'/Uref^2$	$m^2\,s^{-2}$
Iu	dimensionless longitudinal turbulence intensities	DL: $I_u = u_{std}/u$	–
Iv	dimensionless lateral turbulence intensities	DL: $I_v = v_{std}/u$	–
Iw	dimensionless vertical turbulence intensities	DL: $I_w = w_{std}/u$	–
lux	longitudinal integral length scale	Time-averaged longitudinal integral length scale	m

Table D.2: Suggestion for standardised wind tunnel variables describing puff release and concentration measurements. Variables are written output of the measurements and processing programme described in Fischer (2011) and taken from the datasets in 5.1, mainly from Harms et al. (2012). List might be incomplete.

netCDF	long name	Comment	Unit
conc	concentration	-	ppmV
conc_back	background concentration	backround concentration	ppmV
conc_eff	effective concentration	effective concentration, measured in the model domain without the backround concentration. Formula: $C_{eff} = c(measured) - c_{back}(backround)$	ppmV

netCDF	long name	Comment	Unit
conc_peak	peak concentatrion	–	s
time	time	–	s
time_off	offset time	time after the release of gas/tracer until the quantity was measured	–
time_arr	arrival time	time at which the gas/particles are measured	ms
time_tra	transvers time	timespan in which the gas/particles are measured	μs
time_peak	peak time	time at which the peak concentration was measured	s
time_asc	ascend time	time until the measured concentration are increasing. Formula: $asct = pt - at$	s
time_dsc	descend time	time where the measured concentration are decreasing. Formula: $dsct = lt - pt$	s
time_leave	leave time	time until concentrations are not measured.	s
duration	duration	duration of, difference between leaving time and ascent time, Formula: $duration = lt - at$	s
dosage	dosage	–	

Table D.3: Suggestion for standardised wind tunnel reference and coordinate variables. Variables are part of the readme or header information provided by the programme described in Fischer (2011) and taken from the datasets in Table 5.1. List might be imcomplete.

netCDF name	long name	Comment	Unit
source_x	x-position of the emission source	-	m
source_y	y-position of the emission source	-	m
source_z	z-position of the emission source	-	m

netCDF name	**long name**	**Comment**	**Unit**
uref	reference wind velocity	Measured wind velocity in the model domain in a lateral direction at a reference Height (Z_ref). Used to normalize measurement values	$m\,s^{-1}$
xref	reference x-position	x-position of measurement height for Uref	m
yref	reference y-position	y-position of measurement height for Uref	m
zref	reference z-position	z-position of measurement height for Uref	m
href	reference height	reference height	m
lref	reference length	reference height	m
sc	scaling factor	relation between Uref and Upr	
upr	Prantl tube wind velocity	measured wind velocity by the prantl tube	m s^{-1}
model_scale	used model scale	–	–
tracer_gas	tracer_gas	used gas	–
R	universal_gas_constant	universal gas constant: J · mol-1 · K-1	J mol^{-1} K^{-1}
tamb	air_temperature	measured ambient air temperature	°C
tgas	gas_temperature	measured gas temperature	°C
pr	air_pressure	measured ambient air pressure	hPa
dd	wind_direction	wind direction of appproach flow	°
quality_flag	quality_flag	indicator for the quality of the value	–
measurement device	measurement device	–	–

Table D.4: Collection of Wind tunnel variables from the three datasets. Variables are listed with possible false friends or alternative names, descriptions and suggestion for possible new variable names, which are more distinguishable and can be used in the future to avoid confusion.

Variable	Alternative Name/-False friend	Unit	Description	New Variable name
Uref	uref	$m\,s^{-2}$	reference wind speed	uref_FS
	wtref	$m\,s^{-2}$	basic wind speed in the wind tunnel during the measurement	wtref
	uref-model	$m\,s^{-2}$	reference wind in model dimensions	uref_MS
C*	C_star	–	dimensionless concentration	conc_DL
	$c_{effective}$	–	effective concentration	ceff_FS, ceff_MS, ceff_DL
V	Volume		volume of tracer	volume
	V	$m\,s^{-2}$	y_wind component	v
rel_c	rel_c	%	relative concentration	relc
	relative concentration	%	relative concentration	relc
	confidence	%		confidence
t	time	s, h, –	full scale time	time_FS
	temperature	°C	air temperature during laboratory measurements	tair, tamb
	Tw	s	time in wind tunnel	time_MS
	T*	–	dimensionless time	time_DL

References

Ali, N. M., Mohamad, M. F., Wang, W., Hirose, C., Yoshie, R. and Ikegaya, N. (2023), 'Quantifying natural cross-ventilation flow of a two-layered model used for terraced houses in tropical zones by particle image velocimetry', Building and Environment **244**.
URL: *https://doi.org/10.1016/j.buildenv.2023.110829*

Arakawa, A. and Lamb, V. R. (1977), 'Computational design of the basic dynamical processes of the ucla general circulation model', Method. Comput. Phys. **17**, 173–265.
URL: *https://doi.org/10.1016/B978-0-12-460817-7.50009-4*

Badeke, R., Matthias, V. and Grawe, D. (2021), 'Parameterizing the vertical downward dispersion of ship exhaust gas in the near field', Atmospheric Chemistry and Physics **21**(8), 5935–5951.
URL: *https://doi.org/10.5194/acp-21-5935-2021*

Baklanov, A., Grimmond, C. S., Carlson, D., Terblanche, D., Tang, X., Bouchet, V., Lee, B., Langendijk, G., Kolli, R. K. and Hovsepyan, A. (2018), 'From urban meteorology, climate and environment research to integrated city services', Urban Climate **23**, 330–341.
URL: *https://doi.org/10.1016/j.uclim.2017.05.004*

Balczó, M., Gromke, C. and Ruck, B. (2009), 'Numerical modeling of flow and pollutant dispersion in street canyons with tree planting', Meteorologische Zeitschrift **18**, 197–206.
URL: *http://dx.doi.org/10.1127/0941-2948/2009/0361*

Bastigkeit, I. (2011), Erzeugung von Validierungsdaten für wirbelauflösende mikroskalige Strömungs-und Ausbreitungsmodelle, PhD thesis, Universität Hamburg, Staats-und Universitätsbibliothek Hamburg Carl von Ossietzky.
URL: *https://ediss.sub.uni-hamburg.de/handle/ediss/4405*

Baumann-Stanzer, K., Andronopoulos, S., Armand, P., Berbekar, E., Efthimiou, G., Fuka, V., Gariazzo, C., Gašparac, G., Harms, F., Hellsten, A., Jurčáková, K., Petrov, A., Rákai, A., Stenzel, S., Tavares, R., Tinarelli, G. and Trini Castelli, S. (2015), 'COST ES1006 Model evaluation case studies: Approach and results'. ISBN: 987-3-9817334-2-6.

Bermudez, L. (2017), 'New frontiers on open standards for geo-spatial science', Geo-Spatial Information Science **20**, 126–133.
URL: *https://doi.org/10.1080/10095020.2017.1325613*

Boettcher, M., Flagg, D. D., Grawe, D., Hoffmann, P., Schoetter, R., Petrik, R. and Schlünzen, K. H. (2024a), 'Hamburg urban summer climate data on 250 m resolution (version 1.0)'.
URL: *https://doi.org/10.26050/WDCC/METRAS_1981-2010_summer_v1*

Boettcher, M., Flagg, D. D., Grawe, D., Hoffmann, P., Schoetter, R., Petrik, R. and Schlünzen, K. H. (2024b), 'Hamburg urban winter climate data on 250 m resolution (version 1.0)'.
URL: *https://doi.org/10.26050/WDCC/METRAS_1981-2010_winter_v1*

Bohnenstengel, S., Schlünzen, K. H. and Grawe, D. (2004), 'Influence of thermal effects on street canyon circulations', Meteorologische Zeitschrift **13**(5), 381–386.
URL: *http://dx.doi.org/10.1127/0941-2948/2004/0013-0381*

Briscolini, M. and Santangelo, P. (1989), 'Development of the mask method for incompressible unsteady flows', Journal of Computational Physics **84**, 57–75.
URL: *https://doi.org/10.1016/0021-9991(89)90181-2*

Bruse, M. and Fleer, H. (1998), 'Simulating surface–plant–air interactions inside urban environments with a three dimensional numerical model', Environmental Modelling & Software **13**, 373–384.
URL: *https://doi.org/10.1016/S1364-8152(98)00042-5*

Cheng, G. (2023), Parameterization of canopy processes for atmospheric models, PhD thesis, Universität Hamburg, Staats-und Universitätsbibliothek Hamburg Carl von Ossietzky.
URL: *https://ediss.sub.uni-hamburg.de/handle/ediss/10597*

De Vos, M. G., Hazeleger, W., Bari, D., Behrens, J., Bendoukha, S., Garcia-Marti, I., Haren, R. V., Haupt, S. E., Hut, R., Jansson, F., Mueller, A., Neilley, P., Oord, G. V. D., Pelupessy, I., Ruti, P., Schultz, M. G. and Walton, J. (2020), 'Open weather and climate science in the digital era'.
URL: *https://doi.org/10.5194/gc-3-191-2020*

Di Sabatino, S., Olesen, H. R., Berkowicz, R., Franke, J., Schatzmann, M., Leitl, B., Schlünzen, K. H., Britter, R., Martilli, A. and Carissimo, B. (2011), 'Towards a model evaluation protocol for urban scale flow and dispersion models', International Journal of Environment and Pollution **47**, 326–336.
URL: *https://doi.org/10.1504/IJEP.2011.047346*

Dickinson, R. E., Zebiak, S. E., Anderson, J. L., Blackmon, M. L., Luca, C. D., Hogan, T. F., Ji, M., Rood, R. B., Suarez, M. J. and Taylor, K. E. (2002), 'How can we advance our weather and climate models as a community?'.
URL: *https://www.jstor.org/stable/26215345*

Eaton, B., Gregory, J., Drach, B., Taylor, K., Hankin, S., Blower, J., Caron, J., Signell, R., Bentley, P., Rappa, G., Höck, H., Pamment, A., Juckes, M., Raspaud, M., Horne, R., Whiteaker, T., Blodgett, D., Zender, C., Lee, D., Hassell, D., Snow, A. D., Kölling, T., Allured, D., Jelenak, A., Soerensen, A. M., Gaultier, L. and Herlédan, S. (2022), 'Netcdf climate and forecast (cf) metadata conventions'.
URL: *https://cfconventions.org/Data/cf-conventions/cf-conventions-1.10/cf-conventions.html*

Eaton, B., Gregory, J., Drach, B., Taylor, K., Hankin, S., Caron, J., Signell, R., Bentley, P., Rappa, G., Höck, H., Pamment, A., Juckes, M., Raspaud, M., Blower, J., Horne, R., Whiteaker, T., Blodgett,

D., Zender, C., Lee, D., Hassell, D., Snow, A. D., Kölling, T., Allured, D., Jelenak, A., Soerensen, A. M., Gaultier, L., Herlédan, S., Manzano, F., Bärring, L., Barker, C. and Bartholomew, S. L. (2024), 'NetCDF Climate and Forecast (CF) Metadata Conventions'.
URL: *https://doi.org/10.5281/zenodo.14275599*

Eggleton, F. and Winfield, K. (2020), 'Open data challenges in climate science', Data Science Journal **19**.
URL: *http://dx.doi.org/10.5334/dsj-2020-052*

Eichhorn, J. (2011), 'MISKAM Manual for Version 6'. URL: https://www.lohmeyer.de/site/assets/files/4840/manual_miskam_63.pdf, accessed: 2025-06-23.

Eichhorn, J. and Kniffka, A. (2010), 'The numerical flow model MISKAM: State of development and evaluation of the basic version', Meteorologische Zeitschrift **19**(1), 81–90.
URL: *http://dx.doi.org/10.1127/0941-2948/2010/0425*

Environmental Wind Tunnel Laboratory (2025), https://www.mi.uni-hamburg.de/en/arbeitsgruppen/windkanallabor/data-sets.html. Last accessed: 01.06.2025.

Eyring, V., Bony, S., Meehl, G. A., Senior, C. A., Stevens, B., Stouffer, R. J. and Taylor, K. E. (2016), 'Overview of the coupled model intercomparison project phase 6 (cmip6) experimental design and organization', Geoscientific Model Development **9**.

Ferner, K. S., Boettcher, M. and Schlünzen, K. H. (2023), 'Modelling the heterogeneity of rain in an urban neighbourhood with an obstacle-resolving model', Meteorologische Zeitschrift **32**(1), 67–81.
URL: *http://dx.doi.org/10.1127/metz/2022/1149*

Ferner, K. S., Boettcher, M., Sieck, K. and Voss, V. (2024), 'Sensitivity studies on rain heterogeneity in hamburg city centre using the obstacle-resolving model mitras'.
URL: *https://doi.org/10.26050/WDCC/MitrasWinterHamburg*

Fischer, J. K. (2021), 'Evaluation of the spatial representativeness of meteorological measurements using a wind tunnel model', Master Thesis, Universität Hamburg.

Fischer, R. (2011), Entwicklung eines problemorientierten Software-Pakets zur automatisierten Aufbereitung, Analyse und Dokumentation von im Windkanal produzierten Daten zur LES-Validierung, PhD thesis, Universität Hamburg, Staats-und Universitätsbibliothek Hamburg Carl von Ossietzky.
URL: *https://ediss.sub.uni-hamburg.de/handle/ediss/4418*

Franke, J., Sturm, M. and Kalmbach, C. (2012), 'Validation of OpenFOAM 1.6.x with the German VDI guideline for obstacle resolving micro-scale models', Journal of Wind Engineering and Industrial Aerodynamics **104–106**, 350–359.
URL: *http://dx.doi.org/10.1016/j.jweia.2012.02.021*

Freitas, S., Harms, F. and Leitl, B. (2021), 'Near-surface characteristics of flow around mesoscale building array (single grid cell)'. Wind tunnel dataset, Environmental Wind Tunnel Laboratory (EWTL), Universität Hamburg. Personal communication, 2021.

Ganske, A., Heil, A., Lammert, A., Kretzschmar, J. and Quaas, J. (2022), 'Publication of Atmospheric Model Data using the ATMODAT Standard', Meteorologische Zeitschrift **31**(6), 493–504.
URL: *http://dx.doi.org/10.1127/metz/2022/1118*

Ganske, A., Heydebreck, D., Höck, H., Kraft, A., Quaas, J. and Kaiser, A. (2020), 'A short guide to increase FAIRness of atmospheric model data', Meteorologische Zeitschrift **29**(6), 483–491.
URL: *http://dx.doi.org/10.1127/metz/2020/1042*

Ganske, A., Kraft, A., Kaiser, A., Heydebreck, D., Lammert, A., Hoeck, H., Thiemann, H., Voss, V., Grawe, D., Leitl, B., Schlünzen, K. H., Kretzschmar, J. and Quaas, J. (2021), 'ATMODAT Standard (v3.0)', World Data Center for Climate (WDCC) at DKRZ .
URL: *https://doi.org/10.35095/WDCC/atmodat_standard_en_v3_0*

Gates, W. L., Boyle, J. S., Covey, C., Dease, C. G., Doutriaux, C. M., Drach, R. S., Fiorino, M., Gleckler, P. J., Hnilo, J. J., Marlais, S. M., Phillips, T. J., Potter, G. L., Santer, B. D., Sperber, K. R., Taylor, K. E. and Williams, D. N. (1999), 'An Overview of the Results of the Atmospheric Model Intercomparison Project (AMIP I)', Bulletin of the American Meteorological Society **80**(1), 29–55.
URL: *http://dx.doi.org/10.1175/1520-0477(1999)080⟨0029:AOOTRO⟩2.0.CO;2*

Gliß, J., Mortier, A., Schulz, M., Andrews, E., Balkanski, Y., Bauer, S. E., Benedictow, A. M. K., Bian, H., Checa-Garcia, R., Chin, M., Ginoux, P., Griesfeller, J. J., Heckel, A., Kipling, Z., Kirkevåg, A., Kokkola, H., Laj, P., Le Sager, P., Lund, M. T., Lund Myhre, C., Matsui, H., Myhre, G., Neubauer, D., van Noije, T., North, P., Olivié, D. J. L., Rémy, S., Sogacheva, L., Takemura, T., Tsigaridis, K. and Tsyro, S. G. (2021), 'Aerocom phase III multi-model evaluation of the aerosol life cycle and optical properties using ground- and space-based remote sensing as well as surface in situ observations', Atmospheric Chemistry and Physics **21**(1), 87–128.
URL: *http://dx.doi.org/10.5194/acp-21-87-2021*

Grawe, D., Schlünzen, K. H. and Pascheke, F. (2013), 'Comparison of results of an obstacle resolving microscale model with wind tunnel data', Atmospheric Environment **79**, 495–509.
URL: *http://dx.doi.org/10.1016/j.atmosenv.2013.06.039*

Gromke, C. and Ruck, B. (2007), 'Influence of trees on the dispersion of pollutants in an urban street canyon—experimental investigation of the flow and concentration field', Atmospheric Environment **41**(16), 3287–3302.
URL: *http://dx.doi.org/10.1016/J.ATMOSENV.2006.12.043*

Gronemeier, T., Surm, K., Harms, F., Leitl, B., Maronga, B. and Raasch, S. (2021), 'Evaluation of the dynamic core of the palm model system 6.0 in a neutrally stratified urban environment: comparison

between les and wind-tunnel experiments', Geoscientific Model Development **14**(6), 3317–3333.
URL: *https://gmd.copernicus.org/articles/14/3317/2021/*

Haarsma, R. J., Roberts, M. J., Vidale, P. L., Senior, C. A., Bellucci, A., Bao, Q., Chang, P., Corti, S., Fučkar, N. S., Guemas, V., von Hardenberg, J., Hazeleger, W., Kodama, C., Koenigk, T., Leung, L. R., Lu, J., Luo, J.-J., Mao, J., Mizielinski, M. S., Mizuta, R., Nobre, P., Satoh, M., Scoccimarro, E., Semmler, T., Small, J. and von Storch, J.-S. (2016), 'High Resolution Model Intercomparison Project (HighResMIP v1.0) for CMIP6', Geoscientific Model Development **9**(11), 4185–4208.
URL: *http://dx.doi.org/10.5194/gmd-9-4185-2016*

Hankin, S. C., Blower, J. D., Carval, T., Casey, K. S., Donlon, C., Lauret, O., Loubrieu, T., Srinivasan, A., Trinanes, J., Øystein Godøy, Mendelssohn, R., Signell, R., de La Beaujardiere, J., Cornillon, P., Blanc, F., Rew, R. and Harlan, J. (2010), Netcdf-cf-opendap: Standards for ocean data interoperability and object lessons for community data standards processes, in 'Proceedings of OceanObs'09: Sustained Ocean Observations and Information for Society', OceanObs09, European Space Agency, p. 450–458.
URL: *http://dx.doi.org/10.5270/OceanObs09.cwp.41*

Harms, F., Berbekar, E. and Luebcke, M. (2012), 'Michel-Stadt measurements 2012 in the frame of the COST Action ES1006'. Wind tunnel dataset, Environmental Wind Tunnel Laboratory (EWTL), Universität Hamburg. Personal communication, 2023.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C. and Oliphant, T. E. (2020), 'Array programming with NumPy', Nature **585**(7825), 357–362.
URL: *https://doi.org/10.1038/s41586-020-2649-2*

Hassan, S., Akter, U. H., Nag, P., Molla, M. M., Khan, A. and Hasan, M. F. (2022), 'Large-Eddy Simulation of Airflow and Pollutant Dispersion in a Model Street Canyon Intersection of Dhaka City', Atmosphere **13**(7), 1028.
URL: *http://dx.doi.org/10.3390/atmos13071028*

Hassell, D., Gregory, J., Blower, J., Lawrence, B. N. and Taylor, K. E. (2017), 'A data model of the Climate and Forecast metadata conventions (CF-1.6) with a software implementation (cf-python v2.1)', Geoscientific Model Development **10**, 4619–4646.
URL: *http://dx.doi.org/10.5194/gmd-10-4619-2017*

Hoyer, S. and Hamman, J. (2017), 'xarray: N-D labeled arrays and datasets in Python', Journal of Open Research Software **5**(1).
URL: *https://doi.org/10.5334/jors.148*

Hrisko, J. (2020), 'Geographic Visualizations in Python with Cartopy. Maker Portal.'. Blogpost, URL: https://makersportal.com/blog/2020/4/24/geographic-visualizations-in-python-with-cartopy, accessed: 2024-02-15.

Janssen, W. D., Blocken, B. and van Wijhe, H. J. (2017), 'Cfd simulations of wind loads on a container ship: Validation and impact of geometrical simplifications', Journal of Wind Engineering and Industrial Aerodynamics **166**, 106–116.
URL: *https://doi.org/10.1016/j.jweia.2017.03.015*

Jones, J. E., Hales, R. C., Larco, K., Nelson, E. J., Ames, D. P., Jones, N. L. and Iza, M. (2023), 'Building and validating multidimensional datasets in hydrology for data and mapping web service compliance', Water **15**(3), 411.
URL: *https://doi.org/10.3390/w15030411*

Jänicke, B., Milošević, D. and Manavvi, S. (2021), 'Review of user-friendly models to improve the urban micro-climate', Atmosphere **12**(10).
URL: *https://doi.org/10.3390/atmos12101291*

Karl, M. (2024), 'City-scale Chemistry Transport Model EPISODE-CityChem', Zenodo.
URL: *https://doi.org/10.5281/ZENODO.14361973*

Korsemeyer, D. and Thompson, C. (2005), 'Guest Editors' Introduction: Internet Access to Scientific Data', IEEE Internet Computing **9**(1), 17–19.
URL: *http://dx.doi.org/10.1109/MIC.2005.15*

Kretzschmar, J., atmodatcode, am kaiser and wachsylon (2022), 'Atmodat, atmodat_data_checker: Version 1.3.1'. Zenodo.
URL: *https://doi.org/10.5281/zenodo.6701508*

Lammert, A., Grützun, V. and Stamnas, E. (2017), 'The SAMD Product Standard (Standardized Atmospheric Measurement Data)'. URL: https://www.cen.uni-hamburg.de/en/icdc/research/docs-research/samd-product-standard-v1-2.pdf, last accessed: 2025-07-25.

Lapo, K., Freundorfer, A., Pfister, L., Schneider, J., Selker, J. and Thomas, C. (2020), 'Distributed observations of wind direction using microstructures attached to actively heated fiber-optic cables', Atmospheric Measurement Techniques **13**(3), 1563–1573.
URL: *https://amt.copernicus.org/articles/13/1563/2020/*

Lapo, K., Freundorfer, A., Pfister, L., Schneider, J. and Thomas, C. (2020), 'Wind tunnel distributed temperature sensing with actively heated fibers and microstructures for detecting wind direction'. Version 1.0, Zenodo, URL: https://doi.org/10.5281/zenodo.3724596.

Leitl, B. (2000), 'Validation Data for Microscale Dispersion Modelling', EUROTRAC newsletter **22**.

Leitl, B. and Schatzmann, M. (2010), 'Validation data for urban flow and dispersion models-are wind tunnel data qualified?', In: The Fifth 420 International Symposium on Computational Wind Engineering (CWE2010), p. 8, Chapel Hill, North Carolina, USA, 2010. .

Loewe, P. (2022), 'Lamb weather types (reduced set) and gale days over the north sea since 1948 based on ncep/ncar reanalysis 1 daily mean sea level pressure fields'.
URL: *https://doi.org/10.26050/WDCC/LambWTyRSetAndGaleDaysOverTheNo*

Loewe, P. and Schade, N. (2024), 'Lamb weather types (reduced set) and gale days over the baltic sea since 1948 based on ncep/ncar reanalysis 1 daily mean sea level pressure fields'.
URL: *https://doi.org/10.26050/WDCC/LambWTyRSetAndGaleDaysOverTheBa*

Maronga, B., Banzhaf, S., Burmeister, C., Esch, T., Forkel, R., Fröhlich, D., Fuka, V., Gehrke, K. F., Geletič, J., Giersch, S., Gronemeier, T., Groß, G., Heldens, W., Hellsten, A., Hoffmann, F., Inagaki, A., Kadasch, E., Kanani-Sühring, F., Ketelsen, K., Khan, B. A., Knigge, C., Knoop, H., Krč, P., Kurppa, M., Maamari, H., Matzarakis, A., Mauder, M., Pallasch, M., Pavlik, D., Pfafferott, J., Resler, J., Rissmann, S., Russo, E., Salim, M., Schrempf, M., Schwenkel, J., Seckmeyer, G., Schubert, S., Sühring, M., von Tils, R., Vollmer, L., Ward, S., Witha, B., Wurps, H., Zeidler, J. and Raasch, S. (2020), 'Overview of the PALM model system 6.0', Geoscientific Model Development **13**(3), 1335–1372.
URL: *https://gmd.copernicus.org/articles/13/1335/2020/*

Maronga, B., Gross, G., Raasch, S., Banzhaf, S., Forkel, R., Heldens, W., Kanani-Sühring, F., Matzarakis, A., Mauder, M., Pavlik, D., Pfafferott, J., Schubert, S., Seckmeyer, G., Sieker, H. and Winderlich, K. (2019), 'Development of a new urban climate model based on the model PALM - Project overview, planned work, and first achievements', Meteorologische Zeitschrift **28**(2), 105–119.
URL: *http://dx.doi.org/10.1127/metz/2019/0909*

Martin, D., Price, C. S., White, I. R., Nickless, G., Petersson, K. F., Britter, R. E., Robins, A. G., Belcher, S. E., Barlow, J. F., Neophytou, M., Arnold, S. J., Tomlin, A. S., Smalley, R. J. and Shallcross, D. E. (2010), 'Urban tracer dispersion experiments during the second DAPPLE field campaign in London 2004', Atmospheric Environment **44**(25), 3043–3052.
URL: *https://doi.org/10.1016/j.atmosenv.2010.05.007*

Meroney, R., Ohba, R., Leitl, B., Kondo, H., Grawe, D. and Tominaga, Y. (2016), 'Review of cfd guidelines for dispersion modeling', Fluids **1**(2), 14.
URL: *http://dx.doi.org/10.3390/fluids1020014*

Met Office (2010 - 2015), 'Cartopy: a cartographic python library with a matplotlib interface'.
URL: *https://scitools.org.uk/cartopy*

Michel, S. J. (2023), Feasibility Study of Heavy Gas Dispersion Experiments in Complex Environments in Physical Modelling, PhD thesis, Universität Hamburg, Staats-und Universitätsbibliothek Hamburg

Carl von Ossietzky.
URL: *https://ediss.sub.uni-hamburg.de/handle/ediss/10429*

Michel, S., Leitl, B., Harms, F., Schalau, S., Schalau, B., Plischka, H., Turnow, J. and Kornev, N. (2023), 'Störfallartige Ausbreitung von dichte-neutralen und schweren Gasen in komplexen Umgebungen'.
URL: *https://doi.org/10.25592/uhhfdm.11779*

Mülmenstädt, J. (2022), 'Echam6-ham2 model experiments to characterize the effect of parameterized autoconversion/warm rain on cloud lifetime'.
URL: *https://doi.org/10.26050/WDCC/Aerocom_ECHAM6-HAM2_rain*

Müller, W. and Seyfert, E. (2000), 'Atkis (R) data base revision and generation of digital topographic base maps.'. International Archives of Photogrammetry and Remote Sensing 710–717.

Oke, T. R., Mills, G., Christen, A. and Voogt, J. A. (2017), Methods. In Urban Climates, Cambridge University Press, p. 44–76.
URL: *http://dx.doi.org/10.1017/9781139016476.004*

Oldrini, O., Armand, P., Duchenne, C., Olry, C., Moussafir, J. and Tinarelli, G. (2017), 'Description and preliminary validation of the pmss fast response parallel atmospheric flow and dispersion solver in complex built-up areas', Environmental Fluid Mechanics **17**, 997–1014.

ONERA (2024), 'DATA ACQUISITION AND PROCESSING – WDF - Wind Tunnel Data Format made to Measure'. Last accessed: 23.08.2024.
URL: *https://www.onera.fr/en/windtunnel/data-acquisition-and-processing*

OpenFOAM (2025), 'Documentation'. URL: https://www.openfoam.com/, Last accessed: 06.06.2025.

Paas, B., Zimmermann, T. and Klemm, O. (2021), 'Analysis of a turbulent wind fiel in a street canyon: Good agreement between les model results and data from a mobile platform', Meteorologische Zeitschrift **30**, 45–57.
URL: *http://dx.doi.org/10.1127/metz/2020/1006*

pandas development team, T. (2020), 'pandas-dev/pandas: Pandas'.
URL: *https://doi.org/10.5281/zenodo.3509134*

Poole, E. L., Heroux, M. A., Vaidya, P. and Joshi, A. (1995), 'Performance of iterative methods in ANSYS on cray parallel/vector supercomputers', Computing Systems in Engineering **6**(3), 251–259.
URL: *http://dx.doi.org/10.1016/0956-0521(95)00016-S*

Rew, R. and Davis, G. (1990), 'NetCDF: an interface for scientific data access', IEEE Computer Graphics and Applications **10**(4), 76–82.
URL: *http://dx.doi.org/10.1109/38.56302*

Rew, R., Davis, G., Emmerson, S. and Davies, H. (2011), 'The netcdf fortran 90 interface guide'.
URL: *https://web.lmd.jussieu.fr/~lmdz/Distrib/Chgt_install/LMDZ20210529.trunk/netcdf-4.0.1/man4/netcdf-f90.pdf*

Sabatino, S. D., Buccolieri, R., Olesen, H. R., Ketzel, M., Berkowicz, R., Franke, J., Schatzmann, M., Schlunzen, K. H., Leitl, B., Britter, R., Borrego, C., Costa, A. M., Castelli, S. T., Reisin, T. G., Hellsten, A., Saloranta, J., Moussiopoulos, N., Barmpas, F., Brzozowski, K., Goricsan, I., Balczo, M., Bartzis, J. G., Efthimiou, G., Santiago, J. L., Martilli, A., Piringer, M., Stanzer, K. B., Hirtl, M., Baklanov, A. A., Nuterman, R. B. and Starchenko, A. V. (2011), 'Cost 732 in practice: the must model evaluation exercise', International Journal of Environment and Pollution **44**(1/2/3/4), 403.
URL: *http://dx.doi.org/10.1504/IJEP.2011.038442*

Salim, M. H., Schlünzen, K. H. and Grawe, D. (2015), 'Including trees in the numerical simulations of the wind flow in urban areas: Should we care?', Journal of Wind Engineering and Industrial Aerodynamics **144**, 84–95.
URL: *http://dx.doi.org/10.1016/j.jweia.2015.05.004*

Salim, M. H., Schlünzen, K. H., Grawe, D., Boettcher, M., Gierisch, A. M. U. and Fock, B. H. (2018), 'The microscale obstacle-resolving meteorological model mitras v2.0: Model theory', Geoscientific Model Development **11**, 3427–3445.
URL: *http://dx.doi.org/10.5194/gmd-11-3427-2018*

Samsel, K. S., Boettcher, M., Grawe, D., Schlünzen, K. H. and Sieck, K. (2025), 'Simulations for assessing model extensions using the obstacle-resolving model mitras'.
URL: *https://doi.org/10.26050/WDCC/WINTER_HAM_MitrasModEx*

Schalau, B., Schalau, S., Leitl, B., Michel, S., Plischka, H., Turnow, J. and Kornev, N. (2022), 'Entwicklung innovativer Softwaretools zur Simulation der Ausbreitung gasförmiger Gefahrstoffe in industrieller Umgebung'. 2.07.
URL: *https://doi.org/10.5281/zenodo.7341513*

Schatzmann, M., Bächlin, W., Emeis, S., Kühlwein, J., Leitl, B., Müller, W. J., Schäfer, K. and Schlünzen, H. (2006), 'Development and validation of tools for the implementation of european air quality policy in germany (project valium)', Atmospheric Chemistry and Physics **6**, 3077–3083.
URL: *http://dx.doi.org/10.5194/acp-6-3077-2006*

Schatzmann, M. and Leitl, B. (2002), 'Validation and application of obstacle-resolving urban dispersion models', Atmospheric Environment **36**(30), 4811–4821.
URL: *http://dx.doi.org/10.1016/S1352-2310(02)00424-7*

Schatzmann, M. and Leitl, B. (2011), 'Issues with validation of urban flow and dispersion cfd models', Journal of Wind Engineering and Industrial Aerodynamics **99**, 169–186.

Schatzmann, M., Olesen, H. and Franke, J. (2010), COST 732 MODEL EVALUATION CASE STUDIES: APPROACH AND RESULTS, University of Hamburg. ISBN: 3000183124.

Scherer, D., Ament, F., Emeis, S., Fehrenbach, U., Leitl, B., Scherber, K., Schneider, C. and Vogt, U. (2019), 'Three-Dimensional Observation of Atmospheric Processes in Cities', Meteorologische Zeitschrift **28**(2), 121–138.
URL: *http://dx.doi.org/10.1127/metz/2019/0911*

Scherer, D., Fehrenbach, U., Grassmann, T., Holtmann, A., Meier, F., Scherber, K., Pavlik, D., Höhne, T., Kanani-Sühring, F., Maronga, B., Ament, F., Banzhaf, S., Langer, I., Halbig, G., Kohler, M., Queck, R., Stratbücker, S., Winkler, M., Wegener, R. and Zeeman, M. (2022), '[uc] 2-datenstandard "stadtklima im wandel" version 1.5.2'. accessed: 20-06-2025.
URL: *https://www.uc2-program.org/sites/default/files/inline-files/uc2_datenstandard.pdf*

Schluenzen, K. (1990), 'Numerical studies on the inland penetration of sea breeze fronts at a coastline with tidally flooded mudflats', Beiträge zur Physik der Atmosphäre, 63 (3/4), 243-256. .

Schlünzen, K. H., Boettcher, M., Fock, B. H., Gierisch, A., Grawe, D. and Salim, M. (2018a), Scientific documentation of the multiscale model system m-sys, Technical report, Meteorological Institute, Unviersität Hamburg.
URL: *https://www.mi.uni-hamburg.de/en/arbeitsgruppen/memi/modelle/dokumentation/msys-scientific-documentation-20180706.pdf*

Schlünzen, K. H., Boettcher, M., Fock, B. H., Gierisch, A., Grawe, D. and Salim, M. (2018b), Technical documentation of of the multiscale model system m-sys, Technical report, Meteorological Institute, Unviersität Hamburg.
URL: *https://www.mi.uni-hamburg.de/en/arbeitsgruppen/memi/modelle/dokumentation/msys-technical-documentation-20180706.pdf*

Schlünzen, K. H., Hinneburg, D., Knoth, O., Lambrecht, M., Leitl, B., López, S., Lüpkes, C., Panskus, H., Renner, E., Schatzmann, M., Schoenemeyer, T., Trepte, S. and Wolke, R. (2003), 'Flow and transport in the obstacle layer: First results of the micro-scale model mitras', Journal of Atmospheric Chemistry **44**, 113–130.
URL: *http://dx.doi.org/10.1023/A:1022420130032*

Snyder, W. H. (1981), Guideline for fluid modeling of atmospheric diffusion. fluid modeling report no. 10, Technical report, Environmental Protection Agency, Research Triangle Park, NC (USA). Environmental Sciences Research Lab.
URL: *https://www.osti.gov/biblio/5876795*

Song, J., Fan, S., Lin, W., Mottet, L., Woodward, H., Wykes, M. D., Arcucci, R., Xiao, D., Debay, J.-E., ApSimon, H., Aristodemou, E., Birch, D., Carpentieri, M., Fang, F., Herzog, M., Hunt, G. R., Jones, R. L., Pain, C., Pavlidis, D., Robins, A. G., Short, C. A. and Linden, P. F. (2018), 'Natural ventilation

in cities: the implications of fluid mechanics', Building Research & Information **46**, 809–828.
URL: *http://dx.doi.org/10.1080/09613218.2018.1468158*

Touzé-Peiffer, L., Barberousse, A. and Treut, H. L. (2020), 'The coupled model intercomparison project: History, uses, and structural effects on climate research', Wiley Interdisciplinary Reviews: Climate Change **11**.
URL: *http://dx.doi.org/10.1002/wcc.648*

Trukenmüller, A., Grawe, D. and Schlünzen, K. H. (2004), 'A model system for the assessment of ambient air quality conforming to ec directives', Meteorologische Zeitschrift **13**, 387–394.
URL: *http://dx.doi.org/10.1127/0941-2948/2004/0013-0387*

UN (2018), 'World urbanization prospects 2018: Highlights'. URL: https://population.un.org/wup/assets/WUP2018-Highlights.pdf, accessed: 2025-08-30.

Vardoulakis, S., Fisher, B. E., Pericleous, K. and Gonzalez-Flesca, N. (2003), 'Modelling air quality in street canyons: A review', Atmospheric Environment **37**.
URL: *http://dx.doi.org/10.1016/S1352-2310(02)00857-9*

Verein Deutscher Ingenieure (2017), 'VDI 3783 Blatt 9 - Environmental meteorology - Prognostic microscale wind field models - Evaluation for flow around buildings and obstacles'.
URL: *https://www.dinmedia.de/de/technische-regel/vdi-3783-blatt-9/267500591*

Verein Deutscher Ingenieure (2024), 'VDI 3783 Blatt 12 - Environmental meteorology - Physical modelling of flow and dispersion processes in the atmospheric boundary layer - Application of wind tunnels'.
URL: *https://www.dinmedia.de/de/technische-regel/vdi-3783-blatt-12/379602361*

Vieira Zezzo, L., Pereira Coltri, P. and Dubreuil, V. (2023), 'Microscale models and urban heat island studies: a systematic review', Environmental Monitoring and Assessment **195**, 1284.
URL: *http://dx.doi.org/10.1007/s10661-023-11906-2*

Voss, V. (2023a), 'High resolution obstacle-resolved model results for the city centre of Hamburg, Germany, from the microscale transport and stream model MITRAS on 21 June 2000', World Data Center for Climate (WDCC) at DKRZ .
URL: *https://doi.org/10.26050/WDCC/MITRAS_ATMODAT*

Voss, V. (2023b), 'Results of a Survey on Obstacle Resolving Model Data'.
URL: *https://doi.org/10.5281/zenodo.8211274*

Voss, V. (2025a), 'Atmodat/nc2atmodat: v1.0.2'.
URL: *https://doi.org/10.5281/zenodo.17035812*

Voss, V. (2025b), 'Vossv/wt2atmodat: v1.0.1'.
URL: *https://doi.org/10.5281/zenodo.17176544*

Voss, V., Schlünzen, K. H. and Grawe, D. (2021), AtMoDat (Atmospheric Model Data)—Creation of a Model Standard for Obstacle Resolving Models, in C. Mensink and V. Matthias, eds, 'Air Pollution Modeling and its Application XXVII', Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 331–333.
URL: *https://doi.org/10.1007/978-3-662-63760-9_48*

Voss, V., Schlünzen, K. H. and Grawe, D. (2024), 'Survey on output data from obstacle resolving atmospheric models', Meteorologische Zeitschrift **33**(3), 243–253.
URL: *http://dx.doi.org/10.1127/metz/2024/1217*

Wang, W., Ikegaya, N., Hirose, C. and Mohamad, M. (2024), 'Indoor flow datasets of two-layered cross-ventilation models by particle image velocimetry and hot wire anemometry', Data in Brief **52**, 109856.
URL: *http://dx.doi.org/10.1016/j.dib.2023.109856*

Whitaker, J., Snow, A. D., Cochran, M., Karney, C., Ouzounoudis, G., Dearing, J., de Kloe, J., Heitor, Filipe, May, R., Itkin, M., Eubank, N., Dunphy, M., Brett, M., da Costa, M. A., de Maeyer, J., Gohlke, C., Willoughby, C., Barker, C., Wiedemann, B. M. and Couwenberg, B. (2019), 'jswhit/pyproj: version 2.0.2 release'.
URL: *https://doi.org/10.5281/zenodo.2592233*

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., Blomberg, N., Boiten, J.-W., da Silva Santos, L. B., Bourne, P. E., Bouwman, J., Brookes, A. J., Clark, T., Crosas, M., Dillo, I., Dumon, O., Edmunds, S., Evelo, C. T., Finkers, R., Gonzalez-Beltran, A., Gray, A. J., Groth, P., Goble, C., Grethe, J. S., Heringa, J., 't Hoen, P. A., Hooft, R., Kuhn, T., Kok, R., Kok, J., Lusher, S. J., Martone, M. E., Mons, A., Packer, A. L., Persson, B., Rocca-Serra, P., Roos, M., van Schaik, R., Sansone, S.-A., Schultes, E., Sengstag, T., Slater, T., Strawn, G., Swertz, M. A., Thompson, M., van der Lei, J., van Mulligen, E., Velterop, J., Waagmeester, A., Wittenburg, P., Wolstencroft, K., Zhao, J. and Mons, B. (2016), 'The FAIR Guiding Principles for scientific data management and stewardship', Scientific Data **3**, 160018.
URL: *http://dx.doi.org/10.1038/sdata.2016.18*

WMO (2023), Guidance on Measuring, Modelling and Monitoring the Canopy Layer Urban Heat Island (CL-UHI)., WMO-No. 1292, World Meteorological Organisation (WMO), CH-1211 Geneva 2, Switzerland.
URL: *https://library.wmo.int/idurl/4/58410*

Zhang, N., Du, Y. and Miao, S. (2016), 'A microscale model for air pollutant dispersion simulation in urban areas: Presentation of the model and performance over a single building', Advances in Atmospheric Sciences **33**, 184–192.
URL: *http://dx.doi.org/10.1007/s00376-015-5152-1*

Zhao, Y., Chew, L. W., Fan, Y., Gromke, C., Hang, J., Yu, Y., Ricci, A., Zhang, Y., Xue, Y., Fellini, S., Mirzaei, P. A., Gao, N., Carpentieri, M., Salizzoni, P., Niu, J. and Carmeliet, J. (2023), 'Fluid tunnel

research for challenges of urban climate', Urban Climate **51**, 101659.
URL: *http://dx.doi.org/10.1016/j.uclim.2023.101659*

List of Figures

List of Tables

List of Acronyms

AtMoDat	Atmosperic Model Data
AMIP	Atmospheric Model Intercomparison Project
AEROCOM	Aerosol Comparisons between Observations and Models
ATKIS	amtliches topographisch-kartographisches Informationssystem (official topographic-cartographic information system)
CF	Climate and Forecast
CFD	Computational fluid model
CEDVAL	Compilation of Experimental Data for Validation of Microscale Dispersion Models
CMIP	Coupled Model Intercomparison Project
COST Action	European Cooperation in Science and Technology Action
DOI	Data Object Identifier
DNS	Direct Numerical Simulation
EWTL	Environmental Wind Tunnel Laboratory
F24	Dataset Ferner et al. (2024)
LES	Large Eddy Simulation
LoD	Level of detail
MUST	Mock Urban Setting Test
MITRAS	Microscale transport and stream model
METRAS	Mesoscale transport and stream model
M-SYS	Multiscale model system
MIP	Model Intercomparison Project
netCDF	Network Common Data Format
ORM	Obstacle Resolving Model
PALM-4U	Paralell Large Eddy Simulation Model
PMSS	Parallel Micro SWIFT SPRAY model
PID	Persistent Identifier
RANS	Reynolds Averaged Navier Stokes
V23	Dataset Voss (2023a)
VDI	Verein Deutscher Ingenieure (association of German engineers)
WDCC	World Data Center for Climate
WMO	World Meteorological Organization

Danksagungen

Während ich diese Arbeit geschrieben habe, habe ich die Unterstützung vieler Menschen erfahren, die mich in dieser Zeit weitergebracht haben. Dafür möchte ich mich bei folgenden Personen bedanken: Zuerst bedanke ich mich bei K. Heinke Schlünzen und Bernd Leitl, die mich im Rahmen meiner Dissertation betreut haben. Mit beiden hatte ich sehr viele hilfreiche, kritische und erhellende Gespräche und Diskussionen geführt. Zudem haben beide immer mit ihrem Optimismus, ihrer Zuversicht und ihren Ermutigungen mich in schwierigen Phasen, in denen ich mich mit der Arbeit schwer getan und meiner Arbeit gezweifelt habe, mich wieder motivieren können, weiterzumachen. Des weiteren bedanken ich mich bei Birgit Gaye, die die Rolle des Panelchairs in der Betreuung meiner Arbeit übernommen und ausgeführt hat.

Als Mitglied der Arbeitsgruppe der Meso- und Mikroskaligen Modellierung (MeMi) am Meteorologischen Institut der Universität Hamburg gewesen, in der ich seit meiner Masterarbeit bis zum Ende meiner Doktorarbeit nahezu ununterbrochen gearbeitet habe, bedanke ich mich bei den vielen Menschen, mit denen ich während all der Zeit zusammengearbeitet habe. Besonderen Dank geht an David Grawe, der mir immer wieder kritisches Feedback zu meiner Arbeit gegeben und mich dem ich mich immer fachlich und technisch beraten lassen konnte. Desweiteren bedanke ich mich bei Marita Böttcher, Ge Cheng, Ronny Badeke und Karolin Ferner. Mit ihnen habe ich im Rahmen meiner Doktorarbeit eng zusammengearbeitet: sei es, dass sie mir Daten zur Verfügung gestellt, meine Programme getestet haben, ich mit ihnen über Schwierigkeiten reden und immer fachliche oder persönliche Ratschläge einholen konnte.

Ich bedanke mich bei Frank Harms, Sylvio Freitas, Simon Michel und Benedikt Seitzer aus der Arbeitsgruppe des Hamburger Windkanallabors (EWTL) für die Bereitstellung von Datensätze an denen ich die Standardisierung entwickeln, ausprobieren und anwenden durfte. Besonderer Dank gilt Frank Harms, der mit seiner Expertise zu Windkanalmodelldaten beratend zur Seite gestanden hat.

Ich bedanke mich bei meinen Kolleg:innen aus dem AtMoDat Projekt, in dem ich vom Zeitraum 2019 bis 2022 zusammengearbeitet habe. Besonderer Dank gilt Daniel Heydebrek und Angelika Heil, beide vom Deutschen Klimarechenzentrum (DKRZ), die mich bei der Standardisierung unserer Modelldaten unterstützt haben.

Ich bedanke mich bei meinem Arbeitgeber, der GEO-NET Umweltconsulting GmbH für die gute Arbeitsatmosphäre, die mir die Fertigstellung der Dissertation ermöglich hat.

Ich bedanke mich außerdem bei Sonja Deckwart, Christina Asmus, Helena Röhling und Lena Maßmann, die immer ein offenes Ohr für mich hatten, wenn ich einen Rat oder Ablenkung gebraucht habe. Und schließlich möchte ich meinem Partner Simon Weidmann danken, der mir all die Jahre zur Seite stand und mich immer unterstützt hat.

Eidesstattliche Versicherung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.
Sofern im Zuge der Erstellung der vorliegenden Dissertationsschrift generative Künstliche Intelligenz (gKI) basierte elektronische Hilfsmittel verwendet wurden, versichere ich, dass meine eigene Leistung im Vordergrund stand und dass eine vollständige Dokumentation aller verwendeten Hilfsmittel gemäß der Guten wissenschaftlichen Praxis vorliegt. Ich trage die Verantwortung für eventuell durch die gKI generierte fehlerhafte oder verzerrte Inhalte, fehlerhafte Referenzen, Verstöße gegen das Datenschutz- und Urheberrecht oder Plagiate.

Hamburg, den 24.09.2025 (Vivien Voss)