Open Forecast Visualization

www.open-forecast.eu

@OpenForecastEU

Co-financed by the Connecting Europe
Facility of the European Union

Project	Open Forecast
Action number	2017-DE-IA-0170
Report title	Open Forecast Visualization
Report number	OF_M08_2020
Date of publication	01.02.2020
Revision	Rev_01
Lead Partner	HLRS
Authors	Leyla Kern (HLRS)
Dissemination level	PU

Executive Summary

This report summarizes the actions performed during the Open Forecast Visualization task.

In the first part, the requirements are studied. The data format is explained along with the employed visualization software. The two factors are then combined in the definition of requirements.

The second part is focusing on the implementation. The adjustments in the software environment are illustrated. Further, also specifications on the hardware are reported in this part. The last section presents some examples of application and illustrates possible extensions of the current visualization tool.

1 Table of Contents

List of Figures

2 Introduction

The increase of size and complexity of data sets highlights the question for appropriate data analysis tools. While common methods are often not applicable or highly time-consuming, modern visualization technologies, as immersive visualization, provide improved analysis possibilities. Immersive visualization allows for interactive exploration of the data in a 3D environment. Multiple kinds of data sets can thereby be combined to a single rendering. The user can handle and compare simulation results from various sources not only on processing level but also on visualization level.

Further, visualization is particularly beneficial in combination with open data where it is crucial to offer widely understandable evaluation tools alongside. As in this project, which shall be designed to address a broad range of user, e.g. citizens, scientists, politicians, the results require a general understandable representation. This goal is reached through visualization, i.e. interactive exploration of data in an immersive environment. Post-processing and integration of simulations is facilitated in visualization software like COVISE, a Collaborative Visualization and Simulation Environment. COVISE supports large-scale data visualization along with high-level user interaction. Complex data sets can then be viewed in Virtual Reality (VR) environments and facilitate the understanding of large multidimensional data. In such cases, classical approaches like 2D visualizations can be cumbersome and might not be suitable to present the full scope. However, immersive visualization where the user can interact with the environment provide much more flexibility and therefore enable better analysis possibilities. Thus, it is reasonable to implement such an environment that can handle the here present data sets adequately and efficient.

The final application will be part of the Particulate Matter Forecast Service (PMFS)[1]. It will ease the analysis of the produced simulation results and will present them in a universally understandable representation. It is further optimized for efficient processing of large data sets which will run on the HLRS super-computing resources. However, as the software is not limited to a single predefined infrastructure it can be used on other clusters and desktop systems as well.

Besides the plain creation of such a tool, it is also given importance to the ability of extension and flexibility, such that the tool can be reused for similar further data.

This report describes the development of an immersive 3D visualization tool for the analysis of WRF-Chem simulation results, as produced in the PMFS Use Case. It reports on the design and implementation of the software, presents illustrations of applications and gives a summary on the events conducted during this action.

[1] see https://open-forecast.eu/deliverables/ for a description of the uses cases

3 Design and Implementation

3.1 Specification and Design

The design of the visualization tool first of all requires the understanding of the data to be processed. Thus, Section 3.1.1 describes the present data format. Section 3.1.2 gives an overview of the software environment that is used for visualization. In the last section of this chapter, the two previous topics will be combined in the definition of requirements.

3.1.1 Data set description

The results from the WRF-Chem simulations are stored in NetCDF [2] files. A separate file is available for each time step. Each file contains the grid used in the simulation along with the variables of interest, evaluated at the grid's vertices. In the final setup, the grid will be composed of 99 elements in the vertical direction and 600 in each of the two lateral dimensions. The data variables are of different dimension and hence require individual handling. In the following, two examples are given of a 3D and 4D variable

Terrain height:	HGT(Time, South_North, West_East)
Particulate matter:	PM10(Time, Bottom_Top, South_North, West_East)

where *Bottom_Top* is the vertical direction, *South_North* and *West_East* are the two lateral directions and *Time* has length 1, as each file only contains data of a single time step.

The WRF-Chem output files are using the 64-bit offset format. A single file, i.e. one time-step is about 20GB large. The here used data sets have a resolution of about 5 minutes, which lead to a storage size of the order of terra bytes for a complete case, where several hours were simulated. The spatial resolution of the grid is 50m [1, 2].

3.1.2 COVISE / VISTLE

COVISE is a collaborative visualization and simulation environment which is developed at HLRS. It is an extensible distributed software environment that combines post-processing of simulations and visualization. Moreover, collaborative sessions can be held which allows multiple users to work together.

COVISE covers all steps of a fully distributed visualization pipeline. Accordingly, the workflow is divided into several modules, each representing a single processing step. As they are independent processes, modules can be distributed to run on different machines. COVISE is particularly designed for the usage on high-performance infrastructures but is available for desktop computers as well [3].

VISTLE, in succession to COVISE, provides a highly scalable software environment. The whole workflow in VISTLE can be distributed across several clusters. Involving both shared and distributed memory, each processing module within VISTLE is parallelized. The parallelization is implemented through MPI and OpenMP. The transfer of data between the modules within a single node is executed through shared memory [4,5].

Both, COVISE and VISTLE, employ the OpenCOVER renderer for rendering on immersive projection systems. It is based on OpenSceneGraph and designed for the handling of large data sets. Further, OpenCOVER plugins let the user adapt parameters from within the virtual environment [6].

3.1.3 Definition of requirements

COVISE/ VISTLE form the basis for the visualization of WRF-Chem data. However, extension of the existing environment is necessary regarding the following components. First of all, a suitable reading module is needed to import the data from the files into the visualization environment. Converted to COVISE/VISTLE format, the data is then passed to post-processing modules. Here, adaption for handling the specific data structure might be necessary as well.

The reader shall be designed to handle 3D time-dependent large-scale data with numerous variables. It reads several files simultaneously, each corresponding to one time-step. It shall provide an interface for interaction through which the user can define the selection and handling of data. This includes the path to the directory of input files as well as the selection of variables of interest. The reading module should further be parallel and distribute the work to several processors.

Once the data is read from the files, post-processing can be applied. There, the data is converted to a suitable representation. Examples of post-processing modules are *Domain Surface, Cutting Surface* or *Iso Surface* modules. Although several modules already exist, they might not be designed to handle the present data structures. Thus, the task remains to extend the modules for handling other data structures as well. Moreover, the creation of new modules can be useful for instance when comparing simulation data to sensor measurements as described in Section 4.

The final product shall be a flexible and extensible tool to handle large-scale multidimensional data by using high-performance computing resources. It should be optimized, though not limited, to the here used NetCDF format as obtained from WRF-Chem simulations.

3.2 Implementation

The implementation was realized in two steps. In a first part, COVISE was used for testing small sample files. After successful creation of the required modules, the implementation was transferred to VISTLE, which facilitates the handling of larger data sets.

3.2.1 Software

The visualization of WRF-Chem data sets required extensions on several levels. In a first step, the data must be passed to the program by making use of the reading module. This module will read data from a file with NetCDF format and create corresponding VISTLE/COVISE objects. The files can reach a storage size in the order of terra bytes and therefore, efficient handling is crucial to ensure smooth rendering. The reading task is distributed to multiple processors. Each NetCDF file is split on grid level into several blocks. These blocks are then allocated to different processors. For efficiency, compact blocks shall be used to have as much data needed for calculations as possible within the block.

The number of partitions is user-defined and the variables are distributed accordingly. No distribution of the time steps is applied as this would decrease efficiency, causing some processors to be without any tasks during some time steps. The parallelization is implemented using the distributed memory concept through MPI. The use of shared memory is limited during reading procedure because only little computation is necessary in processing the acquired data. Within each block, the threads are working serial.

Each block can only access its own data. However, in further processing, it might be necessary to access data of neighboring blocks as well, e.g. when computing iso-surfaces. Thus,additional cells, so called ghost, are added to each block, representing the adjacent cells if neighboring blocks (if any). The grid partitioning is illustrated in Figure 1.

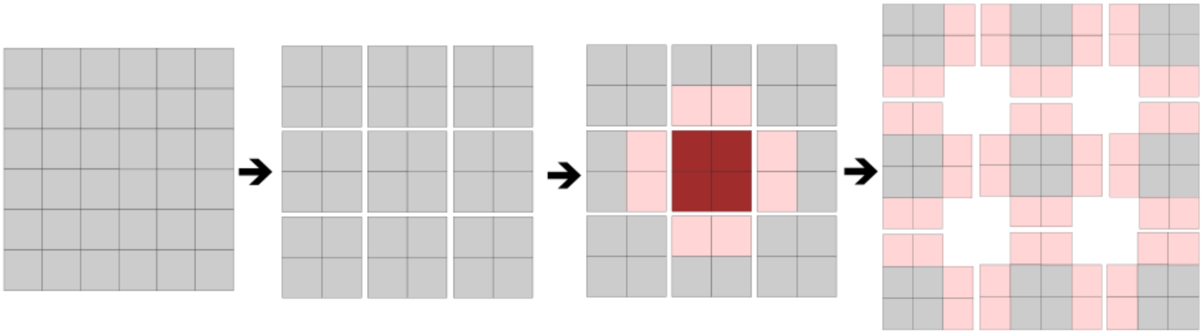

Figure 1: Partitioning of the grid for the distribution to multiple nodes along with the ghost cells highlighted in pink.

To reduce memory usage, the constant state of the grid is exploited. The grid coordinates are unchanged throughout the simulation and therefore there is no need to create individual grid objects for each time step. Instead, the same object is referenced in all time steps. The grid is stored as a structured grid whereas the variables are stored in vectors.

The reader will have several output ports where the acquired data is provided. A separate port is available for the variables to allow the rendering of multiple properties simultaneously. From the output ports, the objects will be passed on for post-processing and finally to the render. Although plenty of modules are already available in VISTLE, not all of them can handle the here present data types. For most modules however, minor adjustments are sufficient to extend them for further types of data structures. Throughout the complete pipeline the domain decomposition and data distribution are maintained.

An example of the module map is shown in Figure 2. The first module *ReadWRFChem* is responsible for reading the data from the files. Its output is then passed on to *DomainSurface.* The two connecting lines represent the grid and the property of interest, respectively, where the latter was PM10 in this example. The *Color* modules which follows on *DomainSurface*, colorizes the surface of the geometry, based on the properties (PM10) values. Color and grid are then combined in *Collect,* which forwards the data to the *Renderer.*

The resulting rendering can be found in the appendix.

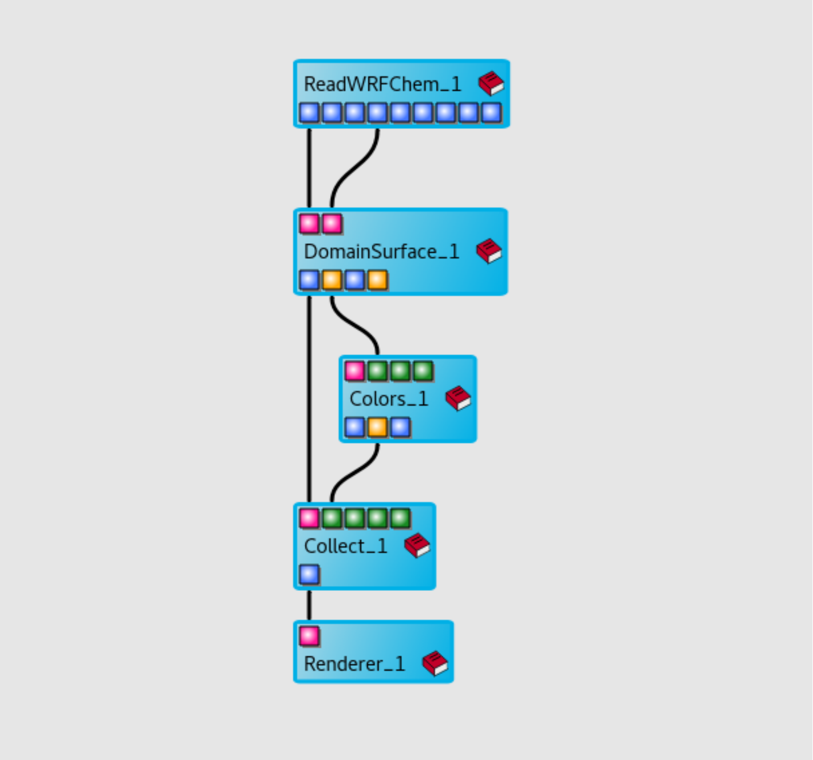

Figure 2 The COVISE module map

The visualization tool was tested with actual output files from the WRF-Chem simulations. However, slightly different dimension sizes were present in these test files compare to those that will finally be used.

The complete project of VISTLE, including the newly added modules is available on github.com [7,8].

3.2.2 Hardware

The visualization is run on HLRS resources. Smaller test files are handled by COVISE which runs on the DALCO Visualization cluster. Thus, cluster has 33 nodes, powered by two quad core SandyBridge 3.3 GHz Xeon CPU.

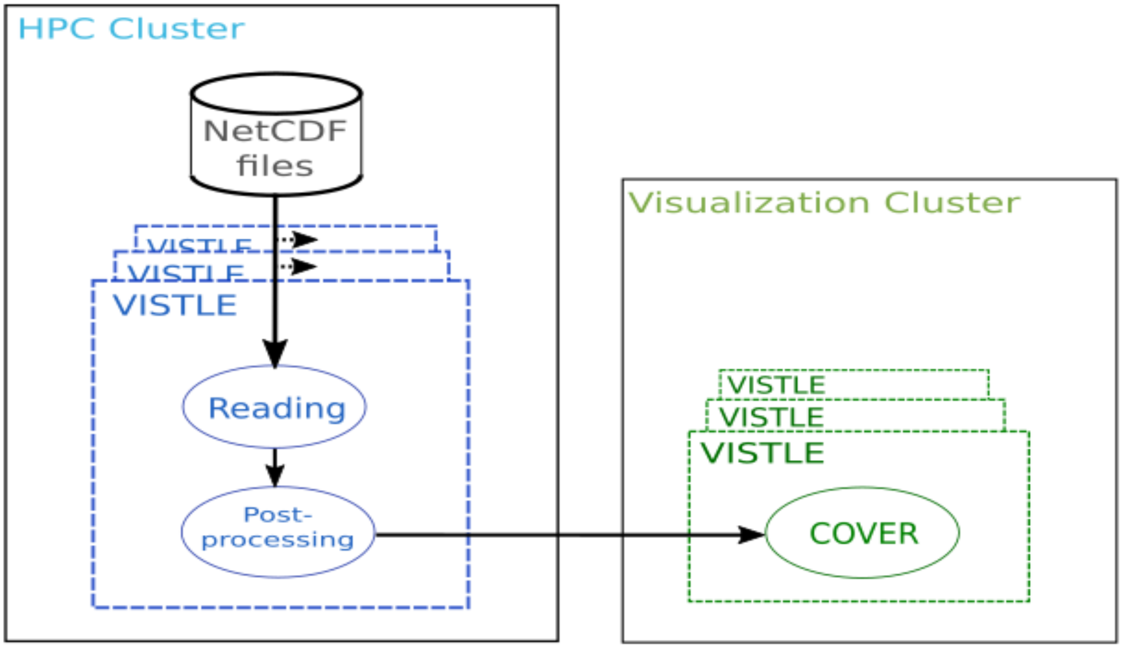

Figure 3 Workflow for the PMFS visualization

For larger files, which required parallel reading, VISTLE is used. The final set-up is shown in Figure 3. As the WRF-Chem simulations are run on HLRS resources, the resulting NetCDF files will be stored there as well. VISTLE can therefore directly access the data from there.

Along with the Visualization cluster, the NEC cluster is employed. The latter provides 751 nodes of SandyBridge, Haswell and SkyLake processors. Extensive modules, as the reading and some post-processing modules are then executed on the NEC cluster. The final module, i.e. the renderer, still runs on the Visualization cluster and is implemented in OpenCOVER. All modules run in parallel. The work is distributed to the processes through the grid partitioning as explained in the previous section.

The results can then be view in the CAVE, a walk-in cube with rear projections on five of the cube’s side walls. This enables the interactive exploration of the data by tracking the user’s head and hand [9,10].

4 Results

4.1 Applications

The implementation was tested in several scenarios. Due to the flexibility and expandability of the software, the usage in combination with other data sets was straightforward. For this reason, results from the WRF-Chem simulations were compared to measurements from various sources. These measurements include data from open data sensor networks from the project luftdaten.info [11], air quality measurement station operated by the city of Stuttgart [12] and mobile sensor devices, run by the University of Stuttgart. These sets were combined on visualization level in COVISE, to allow the comparison of simulation results and sensor data. Figure 4 illustrates the PM10 distribution as calculated in WRF-Chem simulations. Figure 5 shows measurements for the same, collected from the project luftdaten.info. The images shown are however just for demonstration of the tool and are not based on the final data sets yet.

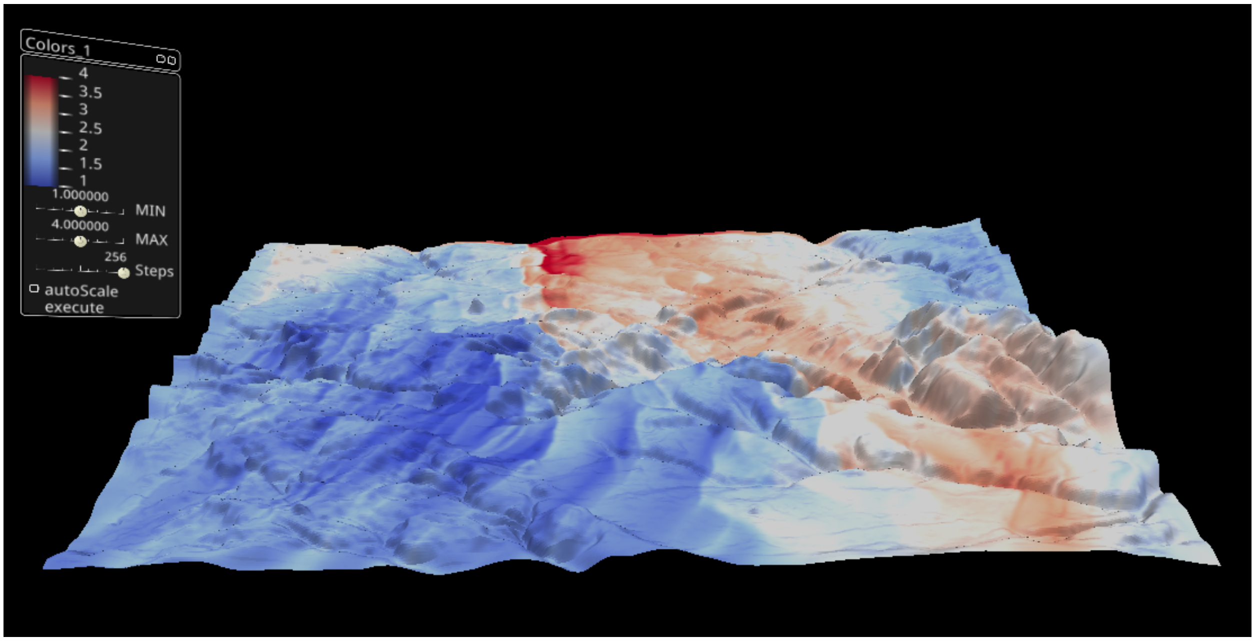

Figure 4 Particulate matter (PM10) distribution as obtained from the simulations for greater Stuttgart. Colors indicate the concentration where red corresponds to approx. 4µg/m³ and blue to 1µg/m³. The data is shown along with the ground terrain height

Several modules were added for the handling of sensor measurements. Apart from the reading module for the sensor data stored in CSV files, also two module for the combination with the simulation data were created. A first module snaps the sensor sample points onto the ground terrain that comes along with the WRF-Chem results. It was used for the generation of Figure 5. Further, a second module was implemented for sampling sensor measurements onto the grid from the simulations. This is particularly useful for the comparison of measurements and simulation.

Similar modules were also designed for the other data sets mentioned above. However, since the reading module is not limited to WRF-Chem data only, it could be also reused for other NetCDF files. With minor adjustments, further simulation data from ECMWF climate simulations could be read into COVISE and rendered using OpenCOVER.

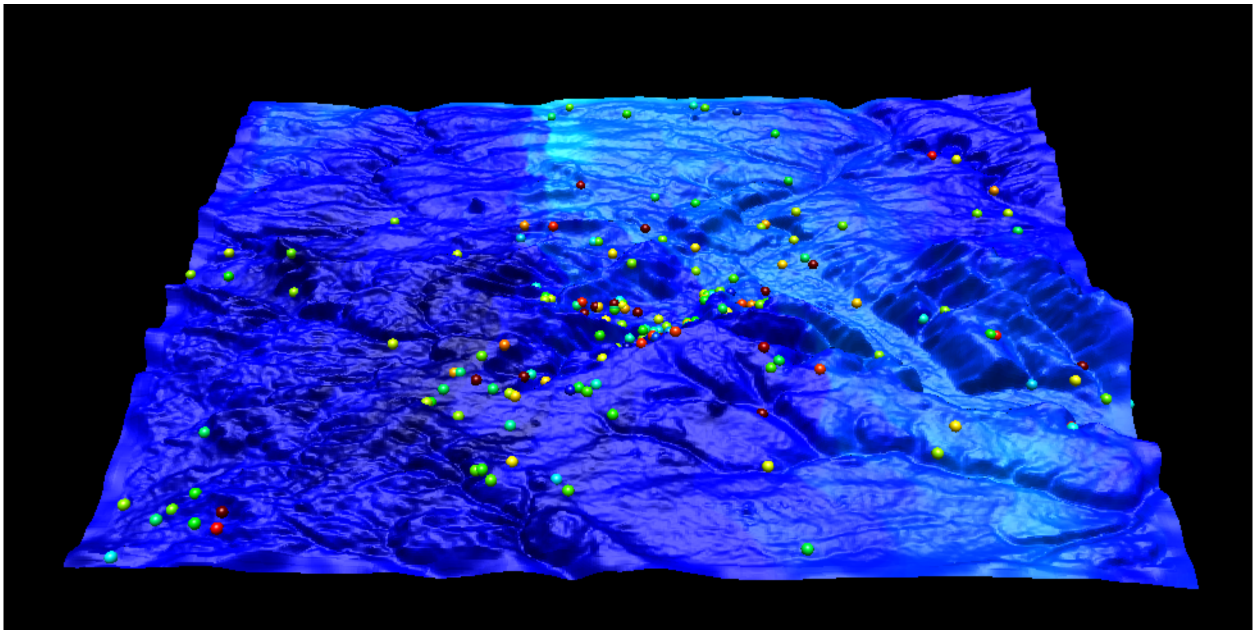

Figure 5 Visualization of sample PM10 sensor measurements from luftdaten.info for the greater area of Stuttgart. Each sphere represents a sensor device. The color indicates the PM10 concentration ranging from 0 (blue) to 25 µg/m³ (red).

4.2 Conclusions

In this action, a visualization tool was created for the rendering of WRF-Chem simulation data. It was shown, that the final product is considerably important for the analysis of simulation results, as it prepares the data in a general understandable form. This is particularly useful, in the presence of large data sets as obtained from WRF-Chem simulations. The here implemented tool is thus designed to handle large-scale data efficiently. At the same time, it provides various analysis possibilities, which can be easily managed by the user through the interface.

The flexible and extensible set-up further supports the reuse with other similar structured data. Moreover, as shown through the applications, the software environment facilitates the combination of several different data sets on visualization level. By this, the simulation results can be compared to sensor measurements.

The current state of the visualization tool for WRF-Chem simulation data provides already several modes of presentation and interaction which supports the analysis of the given data in various factors. These include, for instance, modules for the domain surface of an object and the iso-surface of a certain value of a property. For future visualizations, one can however still add further modules and use the simulation results in other combinations. Such assemblies could be for instance the integration of WRF-Chem data in 3D city models.

The visualization software was already tested with files from the WRF-Chem simulations, though it remains to load a full case with all associated time steps and final grid size.

Finally, the visualization will be presented in further public demonstrations to discuss the results. Also, the organization of workshops along with the provision of training material will then be addressed.

5 Acknowledgements

The work carried out by the Open Forecast project is co-financed by the Connecting Europe Facility of the European Union under action number 2017-DE-IA-0170.

6 References

[1] Grell, G. A., S. E. Peckham, R. Schmitz, S. A. McKeen, G. Frost, W. C. Skamarock, and B. Eder, 2005: Fully-coupled „online“ chemistry within the WRF model. Atmospheric Environment 39, 6957-6975.

[2] UCAR: *NetCDF,* https://www.unidata.ucar.edu/software/netcdf/, Acc: 08.2019

[3] HLRS: *COVISE,* https://www.hlrs.de/covise/ , Acc: 08.2019

[4] HLRS: *VISTLE,* https://www.hlrs.de/vistle/ , Acc: 08.2019

[5] Aumüller M. 2015: The Architecture of Vistle, a Scalable Distributed Visualization

System. In: Markidis S., Laure E. (eds) Solving Software Challenges for Exascale. EASC 2014. Lecture Notes in Computer Science, vol 8759. Springer, Cham,

[6] HLRS: *OpenCOVER*, https://www.hlrs.de/solutions-services/service-portfolio/visualization/covise/opencover/ , Acc: 08.2019

[7] Github: *VISTLE:* https://github.com/vistle/vistle/

[8] Github: *COVISE:* https://github.com/hlrs-vis/covise/

[9] HLRS: *NEC Cluster,* https://www.hlrs.de/systems/nec-cluster-vulcan/, Acc: 08.2019

[10] HLRS: *Visualization cluster,* https://www.hlrs.de/solutions-services/service-portfolio/visualization/virtual-reality/the-cave/ , Acc: 08.2019

[11] Luftdaten.info, https://luftdaten.info, Acc: 08.2019

[12] *Stadtklima Stuttgart,* https://www.stadtklima-stuttgart.de/index.php?luft_messdaten_station_smz , Acc: 08.2019

7 Appendix

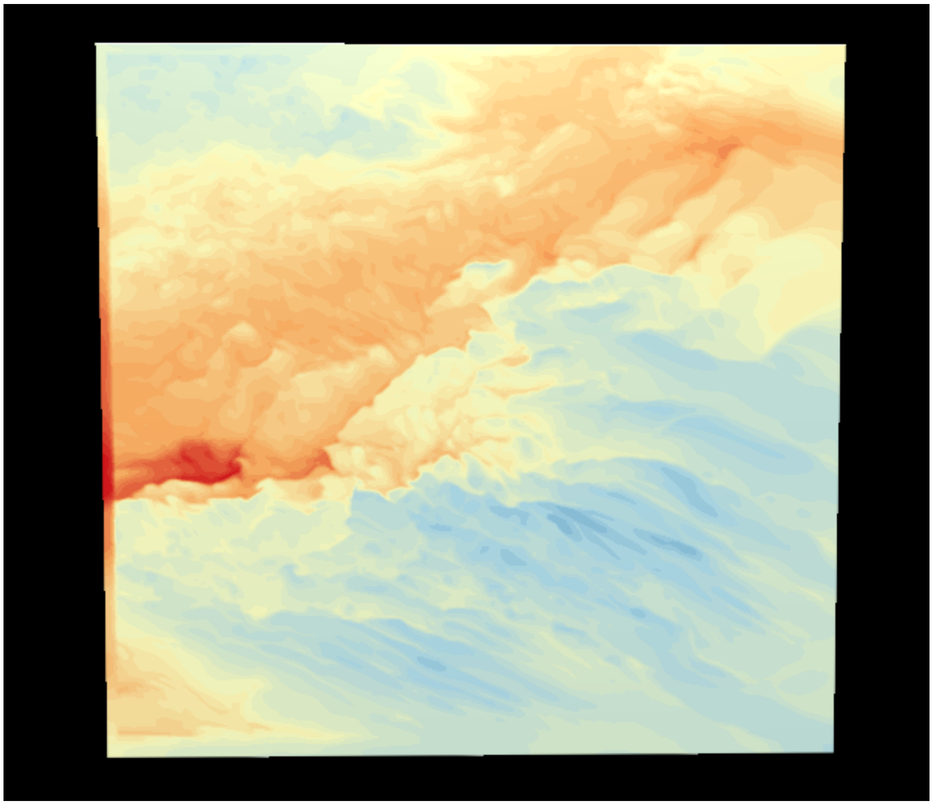

Figure 6 PM10 sample data is shown at the grid surface using the Domain Surface module. The lateral distribution is shown for the region of Stuttgart

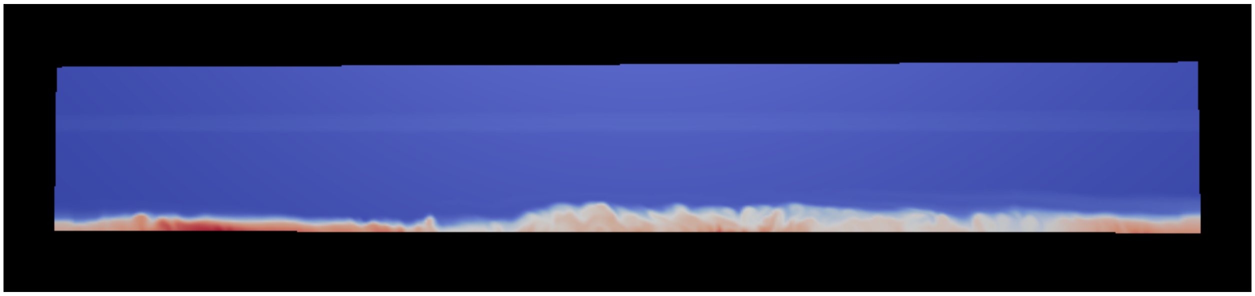

Figure 7 PM10 data from Figure 6 is shown for the cross-section of the grid, illustrating the distribution in the vertical direction. The Cutting Surface module was employed

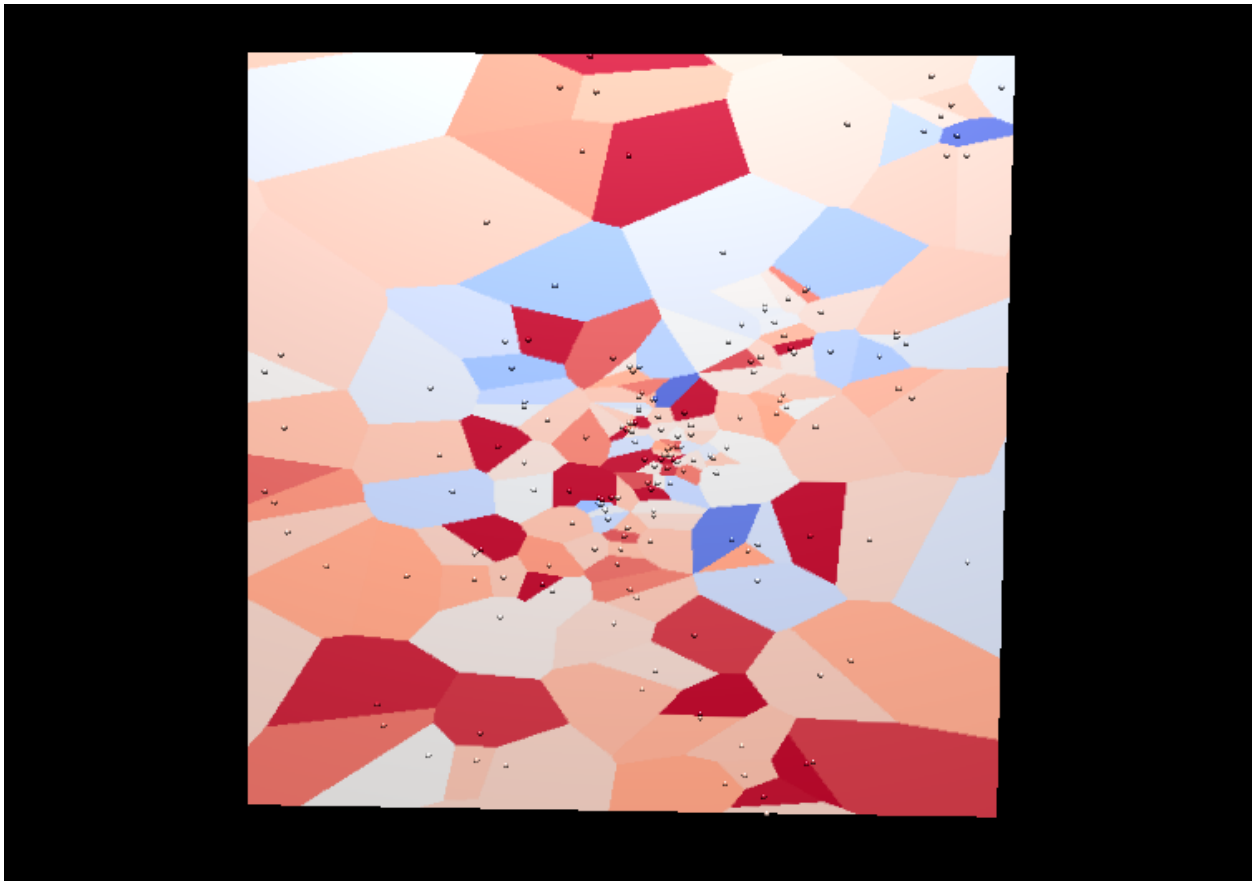

Figure 8 PM10 sensor measurements, displayed as spheres, are sampled to the grid as used in the WRF-Chem simulations. The nearest-neighbor sampling algorithm was used in the implementation