

On the feasibility of using open source solvers for the simulation of a turbulent air flow in a dairy barn

David Janke¹ Alfonso Caiazzo², Naveed Ahmed³, Najib Alia², Oswald Knoth⁴, Baptiste Moreau²,

Ulrich Wilbrandt², Dilya Willink¹, Thomas Amon^{1,5}, Volker John^{2,6}

submitted: November 18, 2019

¹ Leibniz-Institut für
Agrartechnik und Bioökonomie e.V. (ATB)
Max-Eyth-Allee 100
14469 Potsdam
Germany
E-Mail: djanke@atb-potsdam.de
dwillink@atb-potsdam.de
tamon@atb-potsdam.de

² Weierstrass Institute
Mohrenstr. 39, 10117 Berlin
Germany
E-Mail: alfonso.caiazzo@wias-berlin.de
najib.alia@wias-berlin.de
baptiste.moreau@wias-berlin.de
ulrich.wilbrandt@wias-berlin.de
volker.john@wias-berlin.de

³ Department of Mathematics
and Natural Sciences
Gulf University for Science and Technology
Kuwait City
Kuwait
E-Mail: ahmed.n@gust.edu.kw

⁴ Leibniz-Institut für
Troposphärenforschung (TROPOS)
Permoserstraße 15
04318 Leipzig
Germany
E-Mail: knoth@tropos.de

⁵ Freie Universität Berlin
Institute for Animal Hygiene and
Environmental Health
Robert-von-Ostertag-Str. 7-13
14163 Berlin
Germany
E-Mail: tamon@atb-potsdam.de

⁶ Freie Universität Berlin
Department of Mathematics and
Computer Science
Arnimallee 6
14195 Berlin
Germany

No. 2644

Berlin 2019



Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 20372-303
E-Mail: preprint@wias-berlin.de
World Wide Web: <http://www.wias-berlin.de/>

On the feasibility of using open source solvers for the simulation of a turbulent air flow in a dairy barn

David Janke Alfonso Caiazzo, Naveed Ahmed, Najib Alia, Oswald Knoth, Baptiste Moreau, Ulrich Wilbrandt, Dilya Willink, Thomas Amon, Volker John

Abstract

Two transient open source solvers, OPENFOAM and PARMOON, are assessed with respect to the simulation of the turbulent air flow inside and around a dairy barn. For this purpose, data were obtained in an experimental campaign at a 1 : 100 scaled wind tunnel model. Both solvers used different meshes, discretization schemes, and turbulence models. The experimental data and numerical results agree well for time-averaged stream-wise and vertical-wise velocities. In particular, the air exchange was predicted with high accuracy by both solvers with relative errors less than 5 % compared to the experimental results. With respect to the turbulent quantities, good agreements at the second (downwind) half of the barn inside and especially outside the barn could be achieved, where both codes accurately predicted the flow separation and the root-mean-square velocities. Deviations between simulations and experimental results regarding turbulent quantities could be observed in the first part of the barn, due to different inlet conditions between the experimental setup and the numerical simulations. Both solvers proved to be promising tools for the accurate prediction of time-dependent phenomena in an agricultural context, e.g., like the transport of particulate matter or pathogen-laden aerosols in and around agricultural buildings.

1 Introduction

In Europe, dairy cows are mainly housed in naturally ventilated barns (NVBs). Also in pigs and poultry farming, the trend is towards naturally ventilated systems or systems with increased spout areas, mainly to increase the animal welfare and the consumers' acceptance. The NVBs act as sources of airborne pollutants, both gaseous and particle-associated. The gaseous pollutants are mainly ammonia (NH_3), methane (CH_4), carbon dioxide (CO_2), or dinitrogen monoxide (N_2O). Pollutants associated with particles are, e.g., particulate matter (PM) or droplets. These particles can act as carriers for pathogens which arise through sick animals inside the barn. Carried out of the barn, the pathogens can spread diseases that are harmful either for other animals or (in case of zoonosis) for human beings. In order to assess and possibly mitigate the risk of airborne pathogen spreading out of NVBs, it is necessary to obtain insights on the flow fields inside and around the barns and to assess the air exchange (AE). The direct coupling of the inside flow regime with the ambient, turbulent weather conditions makes it hard to measure the flow conditions and the AE. The buildings and their openings are very large, velocities and gaseous concentrations are heterogeneously distributed and vary in time and space [16, 29]. Computational fluid dynamics (CFD) is the appropriate tool for acquiring detailed insight into the complex flow fields encountered in NVBs. In particular, CFD allows to predict the main features of the flow fields also at locations where measurements are practically infeasible, or to perform virtual assessment studies with the help of computational models. The use of CFD to describe the flow characteristics in NVBs has considerably gained popularity in recent years, see Table 1 for some references. Since the flow is turbulent, the accurate modeling of the turbulence in the numerical simulations is of particular relevance. In agricultural applications, numerical flow simulations typically rely on the RANS (Reynolds averaged Navier–Stokes) approach, where the turbulence is completely parameterized. It provides a good cost-benefit ratio and it has been used in scientific and industrial applications for more than 40 years. In the RANS approach, the solutions for the velocity and pressure fields are iteratively computed, until a converged solution is reached, meaning that the flow problem is considered as a steady-state problem. In the literature, the vast majority of simulations for livestock husbandry that can be found was done with commercial flow solvers, e.g., ANSYS or StarCCM+, using the RANS approach (Table 1).

The transport and dispersion of gases and particles in a turbulent flow is by nature a dynamic (time-dependent) process, meaning that important information might be lost when transport is modeled based on a time-averaged flow field, as the one computed with the RANS approach. Consequently, the use of steady-state flow fields can lead to inaccurate results when applied to the transport of gases and particles. Transient simulations resolve

Table 1: Selection of published CFD studies in an agricultural context within the last 12 years.

Reference	Model	software	goals	turbulence model
[40]	pig barn	ANSYS	optimization AE	RANS
[30]	broiler house	OPENFOAM	validation	RANS
[31]	pig barn	CFX	comparison of turbulence models	RANS, LES
[33]	half-scale barn	STAR-CCM+	assessment AE	RANS ($k-\epsilon$)
[37]	experimental pig barn	FLUENT	velocity, tem- perature, gas concentration	RANS ($k-\epsilon$, $k-\omega$), RSM
[10]	sheep / goat barn	NN	particle flow	RANS($k-\epsilon$)

important unsteady scales and so the dynamic characteristics of a flow. Therefore, the use of transient simulations is expected to provide more accurate results for the application of gas and particle transport. However, the gained accuracy comes at the price of greater computational costs. The use of computer clusters, where several processors solve the problem in parallel, is often necessary. Depending on the license policy of the software companies, this can lead to high costs, e.g., when for every additional processor node, an extra license needs to be paid. Also, many of the computer clusters freely accessible to research institutes do not even offer the use of proprietary software. Moreover, commercial tools do not give access to the source code, thus making it difficult to understand the details of numerical methods used within the solver. Open source solvers can represent here an appealing alternative. Besides being free of charge, they provide a complete control over the numerical methods and give also the possibility of customizing the code and implementing tailored methods. In the context of research, they favor exchange of data and source codes, and naturally allow for reproducibility of numerical results by different groups. We also refer to the discussion of the benefits of open source software in the introduction of [44].

Research in the area of CFD brought in recent years the release of several open source packages for flow simulations, such as, to mention some, OPENFOAM ([34]), DEAL.II ([9]), FENICS ([3]), DUNE ([11, 12]). Further available options include the in-house research codes ASAM¹ ([25]) and PARMOON² ([17, 44]), which are currently developed in the research groups of some authors of this paper. To the best of our knowledge, the only studies using a non-commercial solver to investigate the flow inside or around agricultural buildings, are the ones from [30], [28], [27], and [23], where in all cases a steady-state RANS approach was used. No study using transient open source solvers seem to be available.

This study therefore aims to close this gap and to investigate the use of transient open source solvers for turbulent flow simulations inside and around agricultural buildings. The first goal of this paper is to demonstrate that open source codes are able to provide reliable simulation of the flow through and around a NVB. Two codes with different features are involved in the numerical studies: OPENFOAM and PARMOON. To this end, a benchmark problem with a typical naturally ventilated barn with cross flow is defined. The second goal of this paper is to experimentally investigate the benchmark problem in a wind tunnel to derive validation data for the first goal. These data sets will be published as well, which we consider to be very useful for the definition of a

¹developed at Leibniz Institute for Tropospheric Research (TROPOS) by Dr.Oswald Knoth

²developed at Weierstrass Institute for Applied Analysis and Stochastics (WIAS)



Figure 1: Left: Real scale barn. Right: 1:100 scale model in the wind tunnel.

realistic benchmark problem for turbulent flow simulations. They can be used for assessing turbulent flow solvers in the future. Both codes support different types of grids, discretizations, turbulence models, and solvers. The third goal of this paper consists in assessing these codes under these considerations and also under the terms of efficiency, user experience, and resource demands. By sharing the experiences made in this study, we hope to reduce eventually existing reservations towards the use of open source codes and promote their application in the agricultural community.

The paper is organized as follows. Section 2 describes the experimental setup to measure the benchmark dataset and introduces the corresponding mathematical models, which will be the basis for the numerical studies. Some features of the two considered open source packages will be described in detail in Section 3, while the assessment of the numerical studies is presented in Section 4. Finally, Section 5 contains the conclusions of our investigations.

2 Experimental Setup and Mathematical Model

2.1 The Studied Dairy Barn

The studied building is a naturally ventilated dairy barn located in Northern Germany, near the city of Rostock. The barn has a length of 96 m, a width of 34 m, while the roof height varies from 4.2 m at the side walls up to 10.7 m at the gable top. The side walls are completely open, while the gable walls are partially open as sketched in Figure 1. Further detailed information about the barn can be found in [20] or [29].

2.2 Wind Tunnel Setup

Experimental airflow measurements were obtained on a 1:100 scaled model (Figure 1, right) of the above described barn in the *atmospheric boundary layer wind tunnel* (ABLWT) of ATB Potsdam (Figure 2). Within the wind tunnel, a fully developed turbulent flow with a logarithmic vertical velocity profile was generated by the presence of roughness elements at the inflow section. More detailed information about the ABLWT can be found in [46] or [45]. The model was positioned with a 90° angle to the flow direction, so that the barn was under cross-flow.

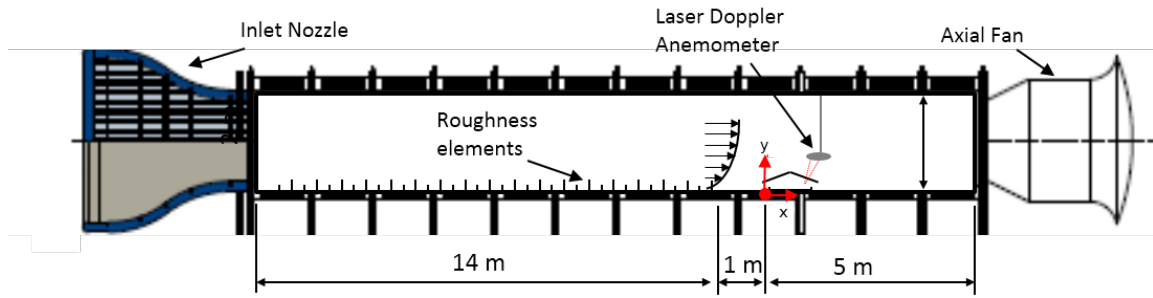
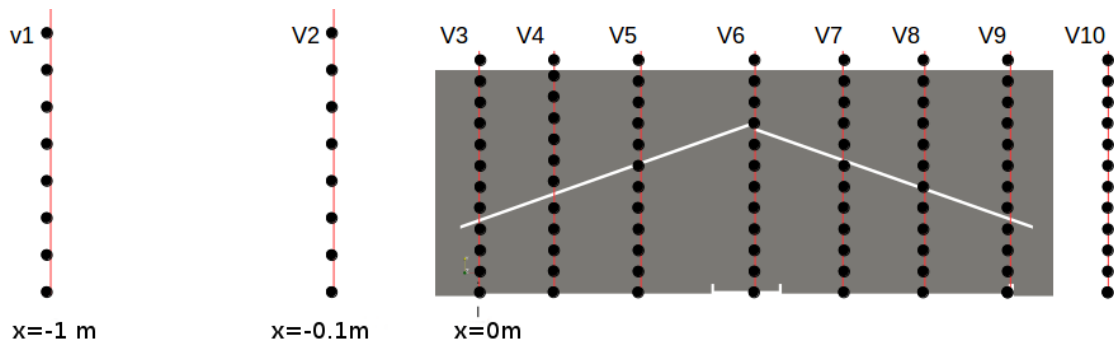


Figure 2: Sketch of the ABLWT with the scaled model.

Table 2: Horizontal coordinates and vertical extrema of the measurement sample lines (see also Figure 3).

	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
x [m]	-1	-0.1	0	0.05	0.1	0.165	0.242	0.292	0.342	0.39
y_{end} [m]	0.6	0.6	0.15	0.15	0.15	0.15	0.15	0.15	0.15	0.15

Figure 3: Visualization of the measured sample lines. All data were acquired in the middle cross-section in the z -direction.

2.3 Velocity Measurements

The velocity components along the vertical direction were measured at several vertical sample lines (v_1, \dots, v_{10}) shown in Figure 3. For each point at the sample lines, the velocity components in x -direction (stream-wise) and y -direction (vertical-wise) were measured with a two-dimensional Laser Doppler Anemometer (LDA) (Dantec Dynamics, Skovlunde, Denmark), mounted on a computer-controlled positioning traverse. The LDA measured the velocity with a sample rate between 20 and 100 Hz, depending on the density of measured seeding particles. Each point was measured for 3 min.

2.4 Mathematical Model

Air is a compressible fluid. However, since the range of velocities observed in the livestock husbandry is much lower than the speed of sound in air, the flow can be considered as incompressible, expecting only a minor impact on the numerical results.

Let $\Omega \subset \mathbb{R}^3$ be the computational flow domain and let $T > 0$ be the final time, which in the simulations was set to $T = 700$ s. Moreover, let us denote with \mathbf{u} [m/s] the air velocity (where u and v stand for the horizontal and vertical components, respectively), and with p [Pa] the air pressure. Without any external force, velocity and

Table 3: Coordinates of the points defining the model geometry (see the sketch in Figure 4).

Point	x [m]	y [m]
1	0	0
2	0	0.0048
3	0.0032	0.0048
4	0.0032	0.0016
5	0.1485	0.0016
6	0.1485	0.008
7	0.150	0.008
8	0.1503	0.0034
9	0.1917	0.0034
10	0.1917	0.0080
11	0.1935	0.0080
12	0.1935	0.0016
13	0.3388	0.0016
14	0.3388	0.0048
15	0.342	0.0048
16	0.3420	0
17	-0.0120	0.044
18	-0.01253	0.04577
19	0.354	0.044
20	0.35451	0.04562
21	0.1761	0.108
22	0.1767	0.1099
23	0.1761	0.112.0
24	0.1757	0.1138

pressure obey the incompressible Navier–Stokes equations given by

$$\begin{aligned} \partial_t \mathbf{u} - \nu \nabla \cdot \mathbb{D}(\mathbf{u}) + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p &= \mathbf{0} \quad \text{in } (0, T] \times \Omega, \\ \nabla \cdot \mathbf{u} &= 0 \quad \text{in } (0, T] \times \Omega. \end{aligned} \quad (1)$$

The velocity deformation tensor $\mathbb{D}(\mathbf{u})$ is the symmetric part of the velocity gradient, i.e., $\mathbb{D}(\mathbf{u}) := (\nabla \mathbf{u} + \nabla \mathbf{u}^T)/2$. In (1), the kinematic viscosity ν [m²/s] is the ratio of the dynamic viscosity η [kg/ms] over the density ρ [kg/m³]. For air at 15 °C, these parameters are $\eta = 1.81 \cdot 10^{-5}$ kg/ms and $\rho = 1.225$ kg/m³, and therefore $\nu = 1.48 \cdot 10^{-5}$ m²/s.

2.5 Computational domain

The computational domain for the model problem is sketched in Figure 4, see also Table 3 for additional information. It is a rectangle of 3 m length and 1 m height with the floor and roof geometry of the wind tunnel model. The coordinate system origin is placed on the bottom edge of the windward side of the barn. As already mentioned, the horizontal coordinate is denoted by x and the vertical coordinate by y .

In order to solve the Navier–Stokes equations (1) numerically, the computational domain needs to be discretized by a mesh, i.e., decomposed in a set of polyhedral mesh cells that cover the domain. On the one hand, the size, and consequently the number, of the mesh cells defines the number of degrees of freedom of the discrete problem to be solved. On the other hand, the size of the mesh cells allows for increasing the accuracy of the approximation in regions of interest. Therefore, the computational mesh plays a crucial role in CFD and it has a

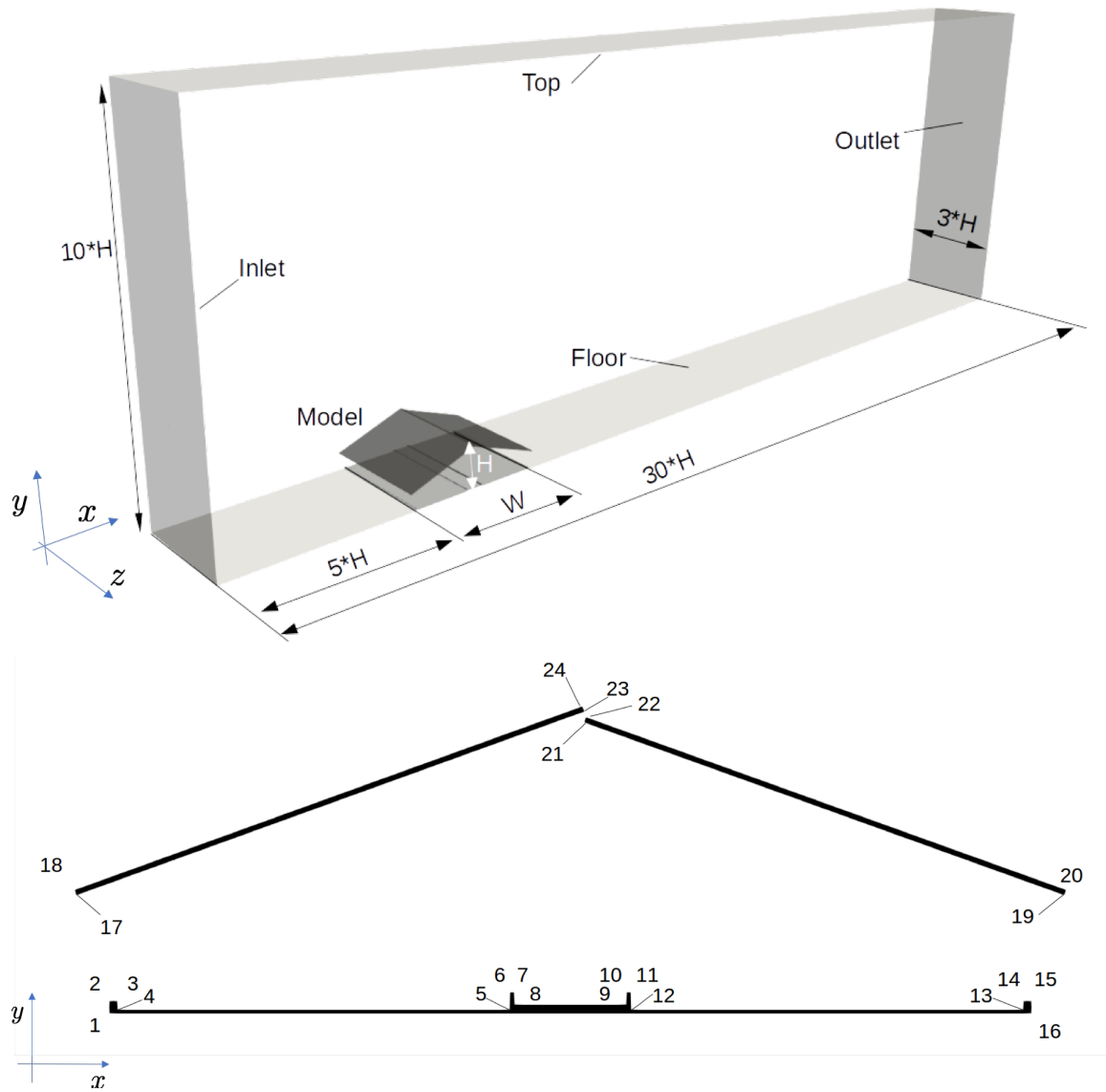


Figure 4: Top: Sketch of the computational domain. Bottom: Detailed view of the 2D projection of the barn. The numbers 1-24 refer to the edges, whose coordinates are listed in Table 3.

direct impact on the results of the numerical solvers. In the study presented in this paper, different meshes were employed, whose structure depended on the particular code. For the sake of clarity, details will be described for each code separately in Section 3.

Remark 2.1. *The geometry used for defining the computational model considered a straight roof (Figure 4). However, due to the weight of the material, the cross section of the roof in the experiment is a slightly convex curve. For this reason, the coordinates of few measurement points were in the experiment on top of the roof but in the simulations below the roof. These points were not considered for the numerical assessment presented in Section 4.*

2.6 Initial Condition and Boundary Conditions

In order to obtain a closed system, the Navier–Stokes equations (1) have to be equipped with an initial condition ($t = 0$) and with boundary conditions on the boundary of Ω .

In practice, the initial condition is not known. For this reason, one has to start with an arbitrary initial flow field,

Table 4: Experimental data from wind tunnel measurements at the inlet.

y [m]	0	0.01	0.02	0.03	0.04	0.05	0.075	0.1	0.125	0.15
u [m/s]	0	3.52	3.77	3.94	4.01	4.10	4.24	4.41	4.55	4.67
y [m]	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	
u [m/s]	4.82	5.18	5.46	5.71	5.96	6.23	6.38	6.57	6.81	

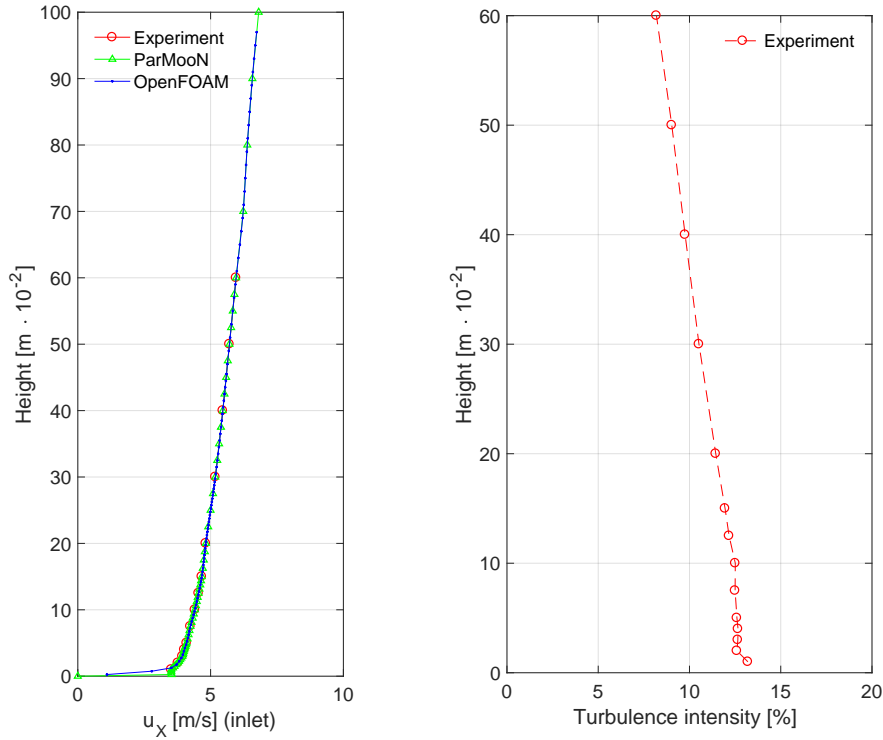


Figure 5: Left: comparison between the measured inflow profile and the interpolated data. Right: Turbulence intensity at the inflow boundary.

to run a simulation until the flow is fully developed, and then to start with monitoring the flow field. Note that the actual initial condition possesses only an impact on the time interval until a fully developed flow field is reached.

At the inlet boundary ($x = -0.5$ m), the velocity profile was prescribed based on experimental data. Concretely, the velocity was measured in the wind tunnel without the scaled barn model at different heights, see Table 4. For the simulations, some interpolation of this profile was used, see Figure 5. At the top boundary and the lateral boundaries, the free-slip condition was imposed. At the outlet boundary ($x = 2.5$ m), a stress-free condition (the so-called do-nothing condition) was applied, i.e., imposing $(2\nu\mathbb{D}(\mathbf{u}) - p\mathbb{I}) \cdot \mathbf{n} = 0$, where \mathbb{I} is the unit tensor and \mathbf{n} the outward pointing normal vector at the outlet. This condition states that the flow should leave the computational domain in the form it is arriving there. The do-nothing condition is a standard approach at outlets, in particular, in situations where no other information on the downstream domain is available. Concerning the bottom boundary, the prescribed boundary condition depended on the code. Either the no-slip condition $\mathbf{u} = \mathbf{0}$ m/s or so-called friction models (which will be detailed in Section 3) were used. The no-slip condition was imposed on the roof boundaries.

In order to characterize the flow regime, the Navier–Stokes equations can be non-dimensionalized by introducing characteristic length, velocity, and time scales. Choosing as characteristic length scale $L = 0.11$ m (approximate height of the barn) and as characteristic velocity scale $U = 5$ m/s (approximate maximal velocity

in a neighborhood of the barn), the Reynolds number of the flow is given by

$$\text{Re} = \frac{LU}{\nu} \approx 37200.$$

This number indicates that the flow is turbulent, requiring therefore suitable numerical methods for its simulation. The characteristic time scale was chosen as usual to be L/U .

2.7 Time Interval

The choice of the final time is briefly motivated in this section. The quantities of interest are time-averaged velocity profiles. Hence, one needs a sufficiently long time interval for obtaining statistically converged results. The area of interest is inside the barn, which has a maximum height of $H \approx 0.11$ m. According to the inflow profile, see Table 4, an air parcel starting at the half of this height has a velocity of about $u = 4$ m/s. Consequently, the parcel passes the whole length of the domain ($30H$) in an interval of time of approximately 0.8 s. The area of interest inside the barn has a width of $W = 0.34$ m. The given parcel passes this width around three times in one second. Based on these considerations, we assumed that a fully developed full profile can be obtained within a time interval of 1 s. Furthermore, a time interval of 6 s is assumed to be sufficient to achieve statistically converged velocity profiles. These estimated were positively validated a posteriori based on the results of our numerical simulations.

2.8 Remarks to Turbulence Modeling

There is no mathematical definition of what turbulence is. A flow is considered to be turbulent when its dynamics possess a wide spectrum of scales (eddies) – ranging from large scales to very small scales – and with the very small scales being of utmost importance for the physical character of the flow (energy dissipation). The Navier–Stokes equations (1) are a proper mathematical model for describing such flows.

Standard discretizations of the Navier–Stokes equations, like the Galerkin finite element method or central finite differences, try to resolve all important scales of the flow. However, the ability of a numerical method to approximate the flow dynamics of small scales depends on the level of resolution of the spatial discretization, i.e., the computational grid. In particular, most of the important scales in turbulent regimes are usually so small that it is not even possible to represent them on computationally affordable grids. These scales are called *unresolved* scales. Of course, scales that cannot be represented cannot be simulated. Because of these unresolved scales, standard discretizations fail for the simulation of turbulent flows, which usually results in a blow-up of the numerical simulations. The remedy consists in augmenting standard discretizations by including so-called turbulence models, which have the purpose to account for the impact of the unresolved scales onto the simulated (resolved) scales. From the numerical point of view, turbulence models introduce additional viscosity into the discrete problem.

Turbulence modeling has been an active field of research for more than forty years. Although numerous turbulence models were proposed, there is neither a standard model nor, in some sense, a best model. There are models, whose derivation is based on physical insight in turbulent flows and there are models, which were derived purely with mathematical arguments. Considering all the different motivations, assumptions, and approximations behind the derivation of turbulence models, it is therefore not surprising that often different numerical results are obtained using different turbulence models.

Commercial codes, e.g., as used for the simulations presented in [39, 40], provide in general classical two-equation turbulence models, like the k - ϵ and k - ω model. The properties of these models, in particular their shortcomings, are well described in [36, Chapters 10 and 11]. The codes used in the study presented in this paper offer the possibility to use turbulence models of different types, which will be explained briefly in the description of the individual codes.

3 The Open Source Packages

This section provides some information on the open source software packages that were used in our study as well as on specific choices in the setup of the numerical simulations. OPENFOAM is widely used by scientists and engineers for flow simulations. The other code PARMOON is a more specialized in-house research code.

3.1 OPENFOAM

OPENFOAM (Open Source Field Operation And Manipulation) is an open source software package containing different applications to model and simulate problems in fluid dynamics [43]. It is written in C++ and designed in an object-oriented fashion that allows the choice amongst many solvers for both the compressible or incompressible Navier-Stokes equations, including also RANS or Large Eddy Simulation (LES) turbulence models. Since OPENFOAM is a popular open source code and all information are readily available at [43], we decided to provide here only a brief description of this solver that concentrates on those aspects that are important for our numerical studies.

Meshing can be done in several ways, either by integrated meshing routines or by importing meshes with external (open or closed source) meshing routines like e.g. GMSH, GAMBIT or SALOME. For this study, the domain was decomposed and meshed with the SNAPPYHEXMESH utility, which generates 3-dimensional meshes containing mainly hexahedra and split-hexahedra cells automatically from triangulated geometries in .stl format (e.g. [32], [19]). The domain consisted of 4 refinement boxes (I, II, III and IV) as depicted in Figure 6. The refinement was factor 1 for each box, meaning that the edge length of a cell in box IV (inside and nearby the model) was 4 times smaller than the original edge length in box I. Table 5 provides information on the used grids.

Table 5: Grid parameters for the coarse, medium and fine grid used in OPENFOAM. 'Size' refers to the edge length in the respective refinement boxes. 'Distance wall' is the distance of the first grid cell's midpoint to the wall in the roof region. y^+ values were chosen as the maximum values from the time-averaged solution at the roof walls. The number of degrees of freedom corresponds to the number of mesh cells.

Mesh	N^o cells	Size I [mm]	Size II [mm]	Size III [mm]	Size IV [mm]	Distance wall [mm]	y^+
coarse	1,434,580	2.5	5	10	20	1	6-14
medium	4,608,675	1.25	2.5	5.0	10	0.4	4-8
fine	14,188,950	0.625	1.25	2.5	5.0	0.4	4-8

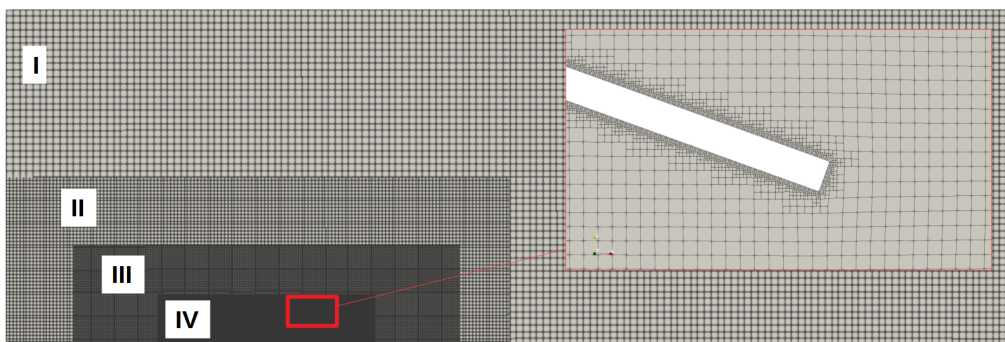


Figure 6: Domain meshed for OPENFOAM with SNAPPYHEXMESH

For the simulations presented in this paper, incompressible LES were setup using the PIMPLEFOAM solver, which merges the well-known PISO (Pressure-Implicit with Splitting of Operators [24]) algorithm with the SIMPLE (Semi-implicit Method for Pressure Linked Equations) algorithm [35], resulting in a fast convergence for transient

simulations, described e.g. by [22]. An adaptive time stepping was chosen with the constraint of a Courant number not higher than 3, resulting in time steps in average of $2 \cdot 10^{-4}$ s.

The subgrid-scale-turbulence was modeled with a one equation eddy viscosity model (`simulationType:LES`, `LESModel:kEqn`), where the not resolved scales are solved similarly to common RANS approaches with an additional equation for the turbulent kinetic energy, described in detail by [47]. The spatial discretization was done using second order `linearUpwind` schemes, the time variable was discretized with the second order `backward` scheme, both described in [43].

The experimentally derived time-averaged velocity profile, shown in Figure 5, was mapped as boundary condition onto the inlet face for the velocity, while the pressure was defined with a Dirichlet `zeroGradient` boundary condition. At the outlet, the boundary condition for velocity was set to `zeroGradient` and the pressure to `fixedValue`.

All simulations were performed at the *North-German Supercomputing Alliance* computer cluster (HLRN) on the Cray-MPI system, using Intel Xeon Haswell compute nodes with 2500 Mhz CPUs. For the simulation of the medium sized mesh, the domain was decomposed with the `scotch` algorithm and distributed with `openMPI` on 120 CPUs. The computation time for reaching 7 seconds was around 8.6 hours.

3.2 PARMOON

The software PARMOON (Parallel Mathematics and object-oriented Numerics) [17, 44] is a C++ finite element library, based on a hybrid MPI/OpenMP parallelization, developed for numerical simulations of partial differential equations from fluid dynamics.

Currently, more than 250 finite elements in two or three dimensions are implemented in PARMOON, including high order polynomials, bubble functions, and discontinuous elements. The software supports both quadrilateral/hexahedral and triangular/tetrahedral meshes. Unstructured simplicial grids in 2d and 3d can be provided using the MEDIT `.mesh` format, which is supported by several established mesh generation packages such as GMSH [18] or TETGEN [41].

Concerning the temporal discretization, PARMOON supports methods with different derivations, complexity, and accuracy. A first group of methods, so-called θ -schemes, include the explicit and the implicit Euler methods as well as the Crank–Nicolson scheme. Furthermore, the fractional-step θ -scheme and the backward differentiation formula of second order (BDF2) can be applied. Moreover, higher order variational type time stepping schemes that include the continuous Galerkin–Petrov(k) (cGP(k)), for $k \geq 2$, and the discontinuous Galerkin(k) (dG(k)), for $k \geq 1$, which are accurate of order $k + 1$, are available and have been successfully applied to several classes of problems.

Linear solvers implemented in PARMOON include iterative approaches (Krylov subspace methods) and several preconditioners, including geometric multigrid methods. Moreover, an extended choice of direct and iterative methods is available via the PETSC library [6, 7, 8]. This library offers, among others, the Boomer AMG method and the parallelized sparse direct solver MUMPS [4, 5].

PARMOON supports a number of turbulence models, in particular some Large Eddy Simulation (LES) models and variational multiscale (VMS) methods, e.g., see [2, 26].

In the following, the setup of the simulations with PARMOON is described. The domain was decomposed with unstructured tetrahedral grids that were generated with GMSH. The three-dimensional grid was obtained from an unstructured triangular grid in the (x, y) -plane, extruding it via several layers in the z -direction.

Several grids with different resolution were used, see Table 6 for information on the number of mesh cells. In all cases, a rather coarse grid was used in the bulk of the domain. In a neighborhood of the barn, the grids became gradually finer and the highest refinement was within and closely around the barn.

For the spatial discretization, the popular inf-sup stable pair of finite elements P_2/P_1 , a so-called Taylor–Hood

Table 6: Information on the grids and the degrees of freedom (dof) on these grids used with PARMOON. All grids were especially refined within and around the barn (see also Figure 7).

	cells	dof velocity	dof pressure	description
Grid 1	697 500	2 933 685	128 352	medium refinement
Grid 2	895 770	3 757 479	163 968	more refinement in front and around the barn
Grid 3	995 985	4 175 142	182 080	similar to Grid 1, but overall somewhat more refined
Grid 4	1 014 030	4 251 681	185 456	similar to Grid 2, but even more refined in front and around the barn

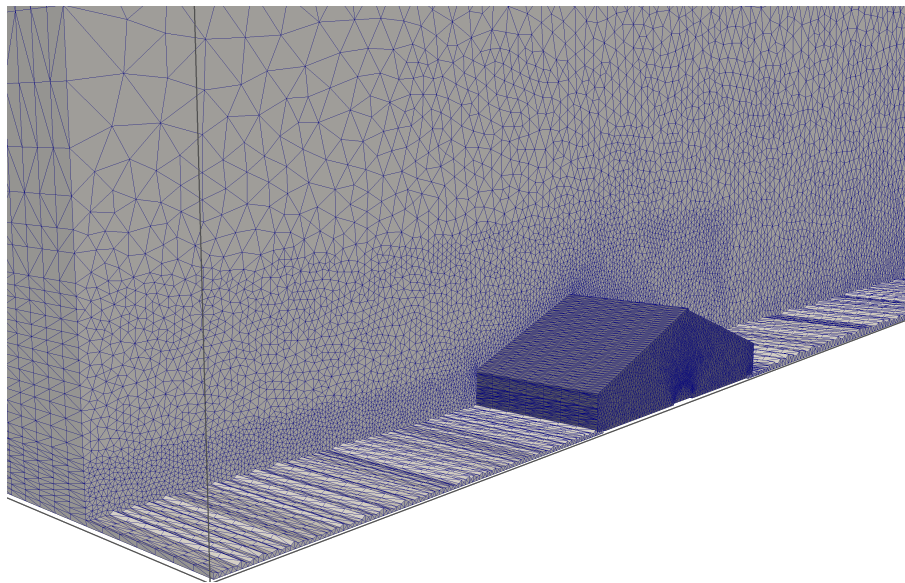


Figure 7: Cut view of Grid 4 used in PARMOON.

pair, was utilized. That means that the discrete velocity is a continuous function, piecewise quadratic on each mesh element, and the discrete pressure is a continuous function, piecewise linear on each mesh element.

As time stepping scheme, the Crank–Nicolson scheme was utilized. It turned out that the length of the time step $\Delta t = 2.5 \cdot 10^{-4}$ s was an appropriate choice in terms of computational time and sensitivity of the results. Hence, the simulation of the whole time interval of 7 seconds required 28000 time steps. A fully implicit approach was used. The stopping criterion for the solution of the nonlinear problem in each time instance was that the Euclidean norm of the residual vector was below 10^{-5} . This criterion was usually satisfied after one iteration. The arising linear problems were solved with a flexible GMRES (FGMRES) method [38] and the so-called least squares commutator (LSC) preconditioner [13] was applied. This preconditioner has been proven to be very efficient for time-dependent incompressible flow problems in the recent study [1]. Its application was similar as described in [1], i.e., the arising pressure Poisson problems were solved directly using MUMPS and the velocity subproblems were solved inexactly with an iterative method (GMRES with SSOR preconditioner, relaxation parameter $\omega = 1$).

A crucial algorithmic component for the simulations of flows through the barn is the turbulence model. For the simulations with PARMOON, a popular LES model, the Smagorinsky model [42], was used. The motivation for choosing this model consists in demonstrating that with an easy-to-implement extension of an existing solver for laminar flow problems, it is possible to perform simulations also for quite challenging applications. In fact,

for the Smagorinsky LES model, the only extension consists in replacing the viscous term of the Navier–Stokes equations (1) by $\nabla \cdot ((\nu + \nu_T)\mathbb{D}(\mathbf{u}))$ with the turbulent viscosity ν_T given by

$$\nu_T = C_{\text{Sma}} \delta^2 \|\mathbb{D}(\mathbf{u})\|_F,$$

where C_{Sma} is the user-chosen Smagorinsky constant, δ is the so-called filter width, and $\|\mathbb{A}\|_F$ is the Frobenius norm of a tensor $\mathbb{A} = (a_{ij})_{i,j=1}^3$ defined by $\|\mathbb{A}\|_F = (\sum_{i,j=1}^3 a_{ij}^2)^{1/2}$. The filter width δ is a measure of the locally smallest resolved scales of the flow. Thus, it is linked to the local mesh width. One can use different measures for the local mesh width. In our experience, e.g., [26, Example 8.128], the length of the shortest edge of a mesh cell K is an appropriate unit for the Smagorinsky model and δ should be set on K by two times the length of the shortest edge of K . In fact, we observed that the results obtained with this filter width were much more accurate than using two times the diameter of K . For brevity, studies with respect to the choice of the filter width will not be presented in this paper. Typical values for the Smagorinsky constant C_{Sma} in academic benchmark problems are of order 0.01. Smagorinsky constants of this order were also used in the simulations presented in this paper. Since the considered flow is more complex than in usual academic test problems, we could observe that smaller constants than 0.01, depending on the grid even 0.01, resulted in a blow-up of the simulations. In the considered application, the flow field in front of the barn is much less complex than in and after the barn, since there are no big vortices in front of the barn. To account for this difference and to reduce the viscous effect of the Smagorinsky model in front of the barn, the used Smagorinsky constant was scaled with 10^{-2} for $x \leq -0.05$ m.

The inlet boundary condition was constructed on the basis of the experimental data, see Table 4. In particular, inlet velocity values were interpolated between the measured points, except for the points between $y = 0$ m and $y = 0.01$ m, where a constant value, which was equal to the measured velocity at height $y = 0.01$ m, was assigned to all degrees of freedom. As usual in simulations of turbulent flows, boundary layers cannot be resolved, in particular the boundary layer at the bottom. A preliminary numerical study, whose results are omitted for brevity, showed that one gets a quite smeared boundary layer of the computed solution already in front of the barn, at sample line v2, if a linear interpolation is used in this interval. With the described approach, a notable improvement could be obtained. Then, this inlet condition was extrapolated horizontally in the domain and the resulting function was the initial condition for all simulations with PARMOON. At the time 1 s, a fully developed flow field was reached and the collection of the data was performed in the time range $[1, 7]$ (s). The comparison with different time intervals, e.g., $[1, 6]$ s or $[2, 7]$ s, showed that the obtained results can be considered to be statistically converged.

All simulations with PARMOON were performed on compute servers HP BL460c Gen9 2xXeon, Fourteen-Core 2600MHz, using 50 processors. The simulation of one time step, including of the calculation of the quantities of interest, took around 10 seconds, such that the computation for the whole time interval took between 80 h and 100 h.

4 Results and Discussion

This section presents the results of the numerical simulations and compares them with the data obtained in the experimental campaign.

4.1 Experimental Results

The measured velocities in the ABLWT experiments are qualitatively shown in Figure 8. Each sampling point is the origin of the time-averaged 2D velocity vector measured by the LDA, where the length and color of the arrows represent the magnitude of the vector. The following flow pattern attributes can be observed:

- (1) when the flow enters the barn at the inlet, it is accelerated and the vectors are focused towards the center line of the opening.

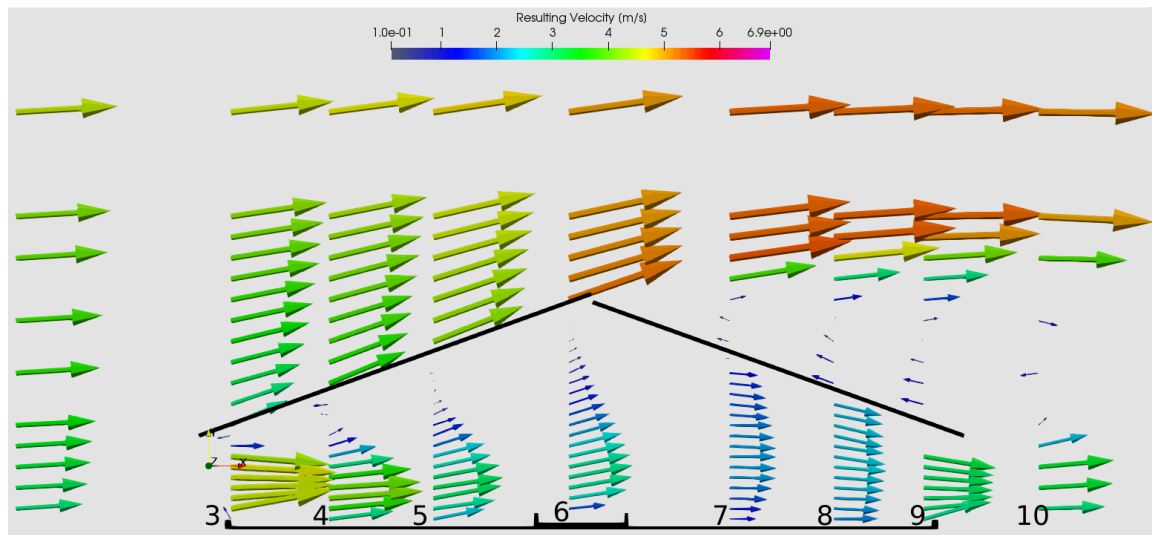


Figure 8: Experimental results for the time-averaged velocity field. The 2D vectors represent the resulting velocity from the measured vectors in x and y direction. Their origin is at the respective sampling location on the vertical sampling lines, labeled with numbers 3-10.

- (2) In the first half of the barn under the roof (at sampling lines 3 until 5), a small re-circulation zone in anti-clockwise direction can be seen.
- (3) After passing the inlet towards the middle of the barn (sampling lines 4, 5 and 6), the vertical component of the flow is positive, resulting in a drift towards the roof.
- (4) From the middle towards the outlet (sampling lines 7 and 8), the vertical component is negative, resulting in a drift towards the floor.
- (5) Outside the barn over the downwind side half of the roof, a large re-circulation zone in clockwise direction has formed.

[45] conducted wind tunnel experiments with a setup similar to this study, where the flow inside and around a NVB under cross-wind direction and different opening geometries was studied. For an opening configuration comparable to this study, attributes (1), (3), (4), and (5) could also be observed in their study. However, the re-circulation zone described as attribute (2) could not be observed. This is probably due to the different scale of the model, which was 1:40 in the study of [45] and resulted in a larger opening.

Several on-farm measurement campaigns focusing on the inside air flow pattern were conducted in the NVB that served as model for the 1:100 scale model in this study. For wind situations with an orthogonal inflow, attributes (1), (3), and (4) were observed by ([14, 15]) at the on-farm measurements. [21] conducted additional measurements under the roof and observed also the anti-clockwise re-circulation zone as described in attribute (2).

4.2 Simulation Results: Air Flow Pattern

As a first qualitative validation, Figure 9 shows the simulated time-averaged 2D air flow patterns inside and around the barn on the symmetry plane of the computational domain. The upper picture shows results for OPENFOAM and the lower picture shows results for PARMOON. Both pictures use the same color scale for the resulting velocity. The white arrows represent the respective velocity vectors with their length scaled by the resulting velocity.

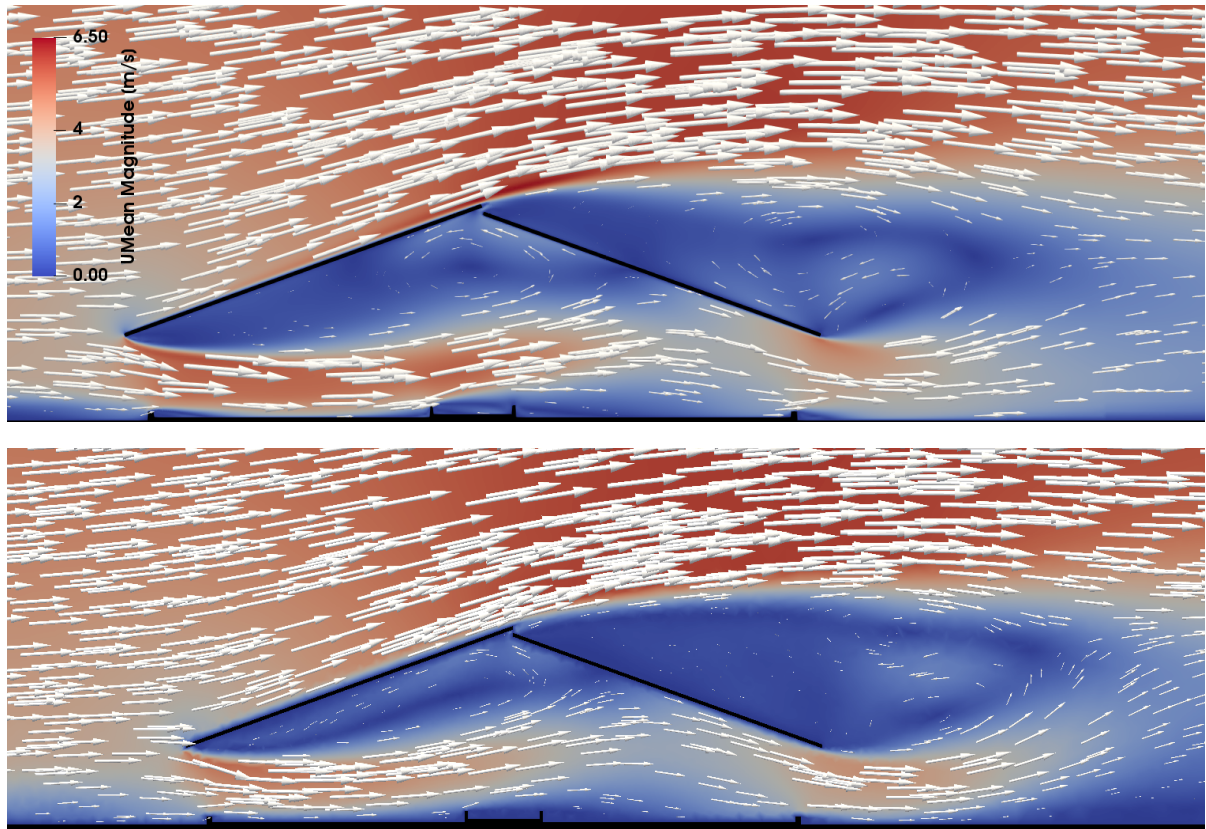


Figure 9: Time-averaged velocity fields for OPENFOAM (top) and PARMOON (bottom). Results were obtained with the finest considered discretization (fine grid for OPENFOAM and Grid 4 for PARMOON).

The previously described air flow pattern attributes (1) to (5) from the wind tunnel experiments are reproduced by both codes. In particular, the re-circulation zone under the roof (attribute (2)) and the large re-circulation zone over the downwind roof side (attribute (5)) are clearly visible. Regarding attribute (5), both codes produce two vortices forming the re-circulation zone, one larger in anti-clockwise direction, and under that, probably originating from the right roof edge, a smaller vortex in clockwise direction. The formation of these two vortices is more pronounced in the OPENFOAM simulations. Since the experimental measurements were not resolved sufficiently fine in this region to detect this feature, this is an example where numerical simulations are providing more information of the flow field than experimental data.

4.3 Simulation Results: Vertical Sampling Lines

The simulation results were compared to the experimental results at the sampling points on the vertical sampling lines sketched in Figure 3. In particular, we compared the time-averaged horizontal velocity (\bar{u}), the time-averaged vertical velocity (\bar{v}) and the corresponding root-mean-square (rms) of their turbulent fluctuations u_{rms} and v_{rms} , defined as:

$$u_{\text{rms}} = \sqrt{\frac{1}{N} \sum_{i=1}^N u' u'} \quad \text{and} \quad v_{\text{rms}} = \sqrt{\frac{1}{N} \sum_{i=1}^N v' v'},$$

with N being the number of time instances for which the velocity was monitored in the simulations. Values superscribed with an apostrophe are the fluctuating parts of the velocities, defined as $u' = u - \bar{u}$ and $v' = v - \bar{v}$.

First of all, the dependency of the numerical results on the grids will be briefly discussed. Considering sampling line v6, which is in the center of the barn (Figure 3), one can see that for OPENFOAM, the results on the medium refined and the fine grid are almost identical. Also in the case of PARMOON, the results obtained on all grids are

Table 7: Comparison (experimental vs. numerical simulations) of the volume flow through the barn.

flow volume (m^3/s)	Experimental	OPENFOAM	PARMOON
inlet	$12.96 \cdot 10^{-2}$	$13.25 \cdot 10^{-2}$ (+2.2%)	$13.11 \cdot 10^{-2}$ (+1.6%)
v9	$13.29 \cdot 10^{-2}$	$13.96 \cdot 10^{-2}$ (+5.0%)	$13.86 \cdot 10^{-2}$ (+4.3%)

quite similar.

In the following evaluation of the numerical results, deviations Δ between the simulated (res_{sim}) and the experimental results (res_{exp}) will be presented as the modulus of the relative differences in percents, with the experimental results as reference, i.e., $\Delta = |(\text{res}_{\text{exp}} - \text{res}_{\text{sim}})/\text{res}_{\text{exp}}| \cdot 100$. Figure 11 shows the comparison of velocity profiles for the u and the v -component for sampling lines v2-v10 from the two codes and the wind tunnel measurements.

The horizontal component of the velocity (u) is the most important one, since it represents the main flow direction. At sample line v2, both codes show a good agreement with the experimental results, with relative errors between 2% and 7%. Hence, it can be stated that the flow profile given at the inlet, compare Figure 5, was transported accurately through the domain. One exception can be seen for the first measurement point at the bottom, where both codes underestimate the velocity of around 27%. This is certainly due to the boundary layer at the wall, where the friction is overestimated compared with the real situation in the turbulent flow. At the inlet of the barn, at sample line v3, a very good agreement between the results of both codes and the experimental results is visible. The same very good agreement is visible at the outlet at the area inside the barn of sample line v9. Both codes show here almost identical results with a slight overprediction of the velocity compared with the experimental results. Altogether, the results indicate that the simulation can be used to estimate the total volume flow through the barn, which is an important quantity of interest with respect to the emissions of the barn.

Comparing the volume flow through the barn with the velocity profile at the barn's inlet, the differences between simulated and experimental results are about 2%, see Table 7. Considering the volume flow through the barn with the velocity profile at the barn's outlet at v9, the difference between experimental data and simulations are of the order of 5% (OPENFOAM) and 4% (PARMOON).

Comparisons for the vertical component of the velocity are presented in Figure 11 (bottom). A generally good agreement between experimental and the numerical results was obtained, with slight differences in details. As expected, the vertical velocity is considerably smaller than the stream-wise velocity.

Results for the root-mean-square velocities are depicted in Figure 12. One can observe that there are notable differences in some parts between the experimental data and the numerical results. However, it is important to observe that some modeling aspects already introduced differences between the computational and the experimental setup. For example, in the experiments roughness elements were used to induce turbulence of the flow already in front of the barn (Section 2.2), whereas in the numerical model, turbulent disturbances of the flow in front of the barn are not introduced. We think that this difference is the reason why the experimental root-mean-square velocities are larger in the first part of the barn. For the flow field in the second part of the barn, not longer the flow in front of the barn but the flow field in its first part possesses the dominating impact. Considering the results within the barn's second part, one can observe that the numerical results agree quite well with the experimental data. A slight overprediction or underprediction can be observed at some points with OPENFOAM and PARMOON, respectively.

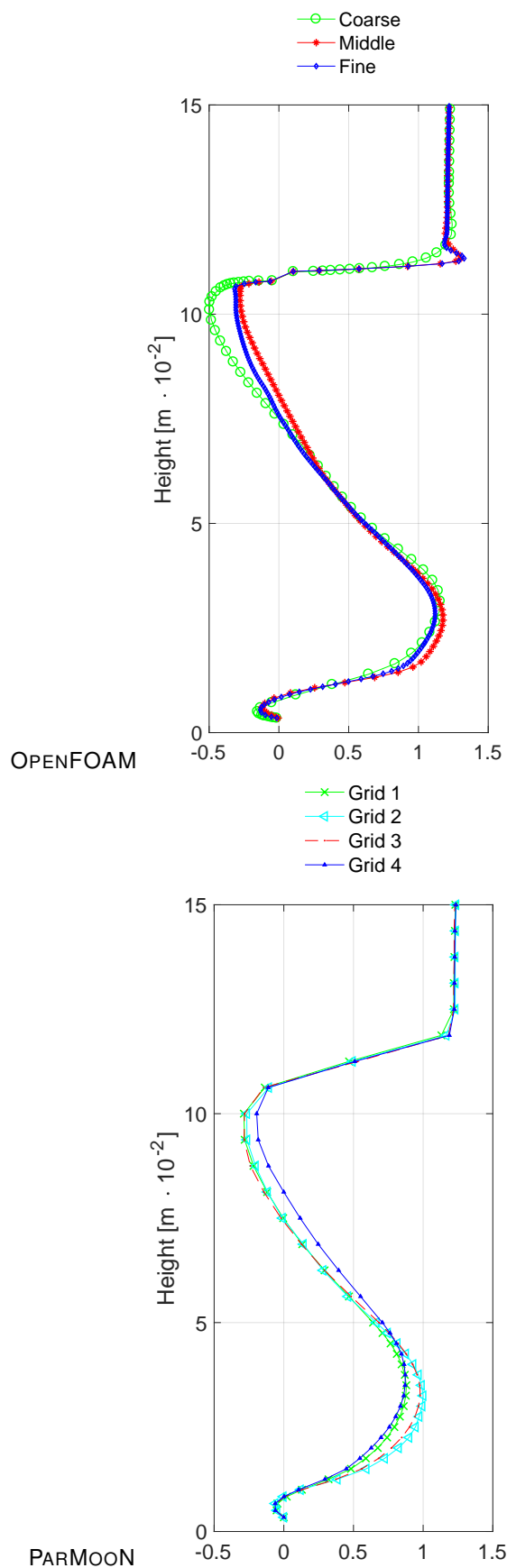


Figure 10: Grid independence study. Shown are the normalized velocities in x-direction at sample line v6, see Figure 3. Left: OPENFOAM, right: PARMOON.

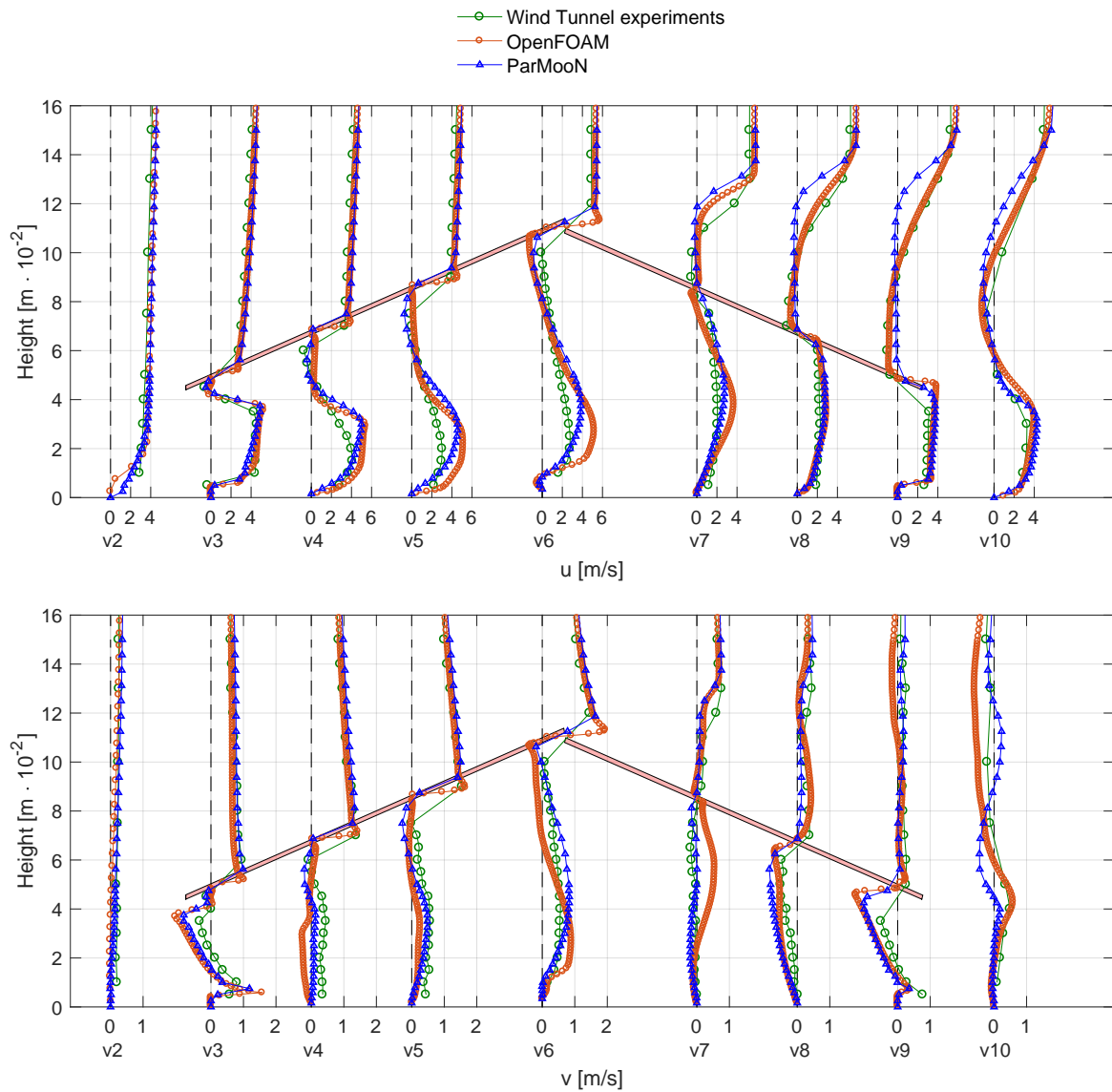


Figure 11: Comparison between experimental data and numerical velocity in the x -direction (top) and y -direction (bottom). Note the effect of the slightly different form of the roof in the experiment and the simulations explained in Remark 2.1.

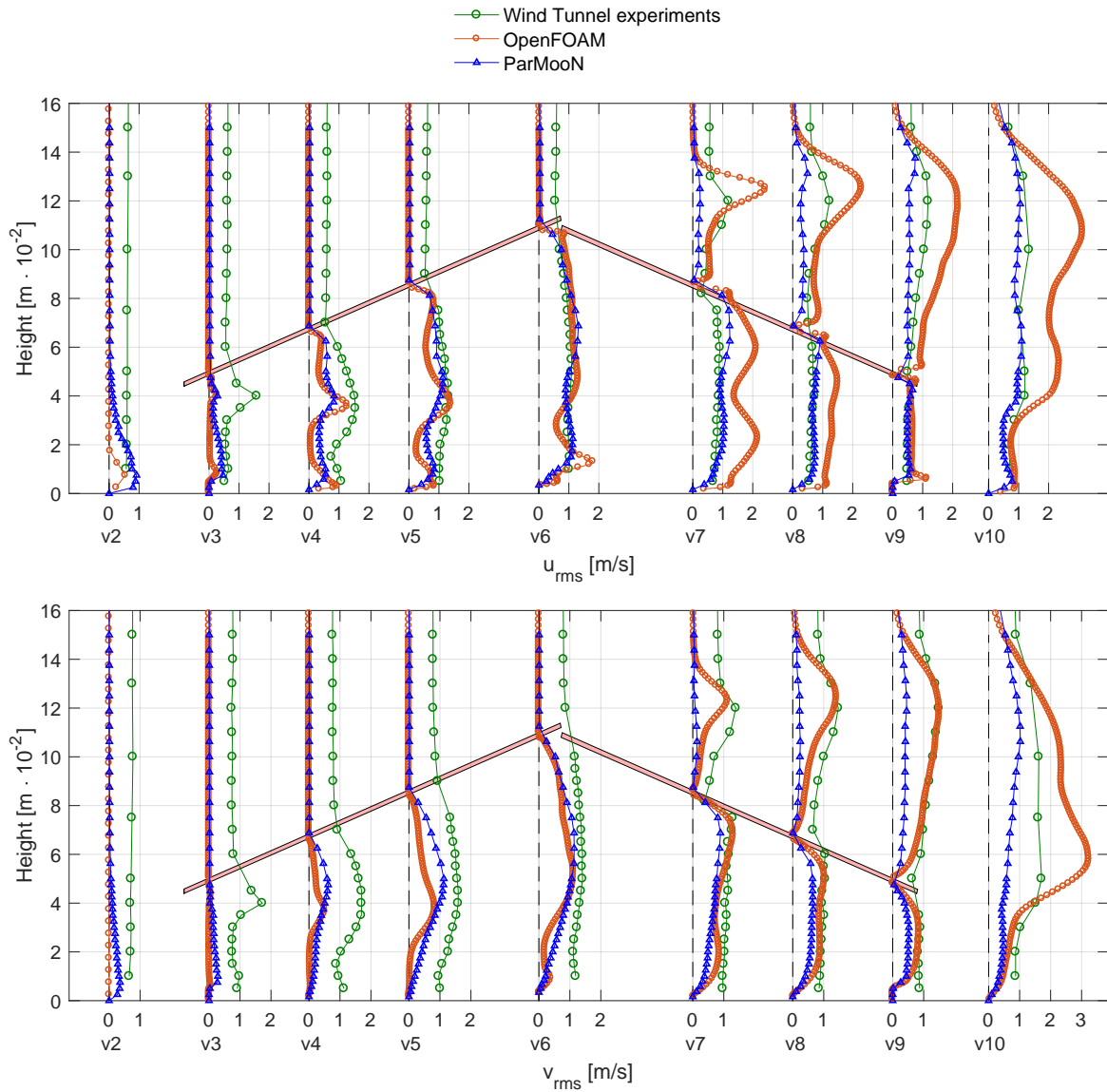


Figure 12: Comparison between experimental data and numerical root-mean-square of velocity in the x -direction (top) and y -direction (bottom). Note the effect of the slightly different form of the roof in the experiment and the simulations explained in Remark 2.1.

5 Conclusions and Outlook

This paper assessed the potential of two open source solvers (OPENFOAM and PARMOON) for simulating the turbulent air flow inside and around a naturally ventilated barn. In particular, results of numerical simulations were compared with experimental data measured in a wind tunnel. The main goal consisted in assessing the performance of the solvers in terms of their capability of simulating the transient flow with sufficient accuracy.

In our opinion, there is a clear result of our studies: both OPENFOAM and PARMOON represent competitive choices for the numerical simulation of the considered application. There was a good agreement of the time-averaged velocities at the sample lines. The differences of the root-mean-square velocities at the first sample lines are due to a difference between the experimental setup and the numerical model. At the other sample lines, there is again a good agreement within the barn.

The considered flow problem is more challenging than usual academic benchmark problems for turbulent flow simulations, e.g., as turbulent channel flows. To create a good benchmark problem from the current setup, the mathematical model should be extended in future by introducing appropriate turbulence in the flow field in front of the barn, to mimic the roughness elements of the experiment. An approach that could be explored for this issue consists in enriching the inlet boundary condition with appropriate noise.

Our experience is that the chosen turbulence model has a major impact on the numerical results. In this respect, we like to note that for OPENFOAM only one turbulence model out of a large variety of available models was chosen. PARMOON also offers more advanced (and more complicated to implement) turbulence models. In this work, a turbulence model was used that can be easily implemented in an existing CFD solver. However, in our opinion, it might be possible to improve the accuracy of the numerical results by using more sophisticated turbulence models. This topic will be subject of future research. Forthcoming studies will also include an assessment of the used open source solvers against commercial codes.

References

- [1] Naveed Ahmed, Clemens Bartsch, Volker John, and Ulrich Wilbrandt. An assessment of some solvers for saddle point problems emerging from the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 331:492–513, 2018.
- [2] Naveed Ahmed, Tomás Chacón Rebollo, Volker John, and Samuele Rubino. A review of variational multiscale methods for the simulation of turbulent incompressible flows. *Arch. Comput. Methods Eng.*, 24(1):115–164, 2017.
- [3] Martin Alnæs, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie Rognes, and Garth Wells. The fenics project version 1.5. *Archive of Numerical Software*, 3(100), 2015.
- [4] Patrick R. Amestoy, Iain S. Duff, Jean-Yves L'Excellent, and Jacko Koster. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM J. Matrix Anal. Appl.*, 23(1):15–41 (electronic), 2001.
- [5] Patrick R. Amestoy, Abdou Guermouche, Jean-Yves L'Excellent, and Stéphane Pralet. Hybrid scheduling for the parallel solution of linear systems. *Parallel Comput.*, 32(2):136–156, 2006.
- [6] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Karl Rupp, Barry F. Smith, Stefano Zampini, Hong Zhang, and Hong Zhang. PETSc Web page. <http://www.mcs.anl.gov/petsc>, 2016.
- [7] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Karl

- Rupp, Barry F. Smith, Stefano Zampini, Hong Zhang, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.7, Argonne National Laboratory, 2016.
- [8] Satish Balay, William D. Gropp, Lois Curfman McInnes, and Barry F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.
 - [9] W. Bangerth, D. Davydov, T. Heister, L. Heltai, G. Kanschat, M. Kronbichler, M. Maier, B. Turcksin, and D. Wells. The `deal.II` library, version 8.4. *Journal of Numerical Mathematics*, 24, 2016.
 - [10] T Bartzanas, C Baxevanou, D Fidaros, D Papanastasiou, C Kittas, et al. Airborne particles and microclimate distribution in a livestock building. In *International Conference on Agricultural Engineering AgEng-Toward Environmental Technologies*, pages 6–8, 2010.
 - [11] Markus Blatt, Ansgar Burchardt, Andreas Dedner, Christian Engwer, Jorrit Fahlke, Bernd Flemisch, Christoph Gersbacher, Carsten Gräser, Felix Gruber, Christoph Grüninger, Dominic Kempf, Robert Klöforn, Tobias Malkmus, Steffen Mühling, Martin Nolte, Marian Piatkowski, and Oliver Sander. The distributed and unified numerics environment, version 2.4. *Archive of Numerical Software*, 4(100):13–29, 2016.
 - [12] Andreas Dedner, Robert Klöforn, Martin Nolte, and Mario Ohlberger. A generic interface for parallel and adaptive discretization schemes: abstraction principles and the DUNE-FEM module. *Computing*, 90(3-4):165–196, 2010.
 - [13] Howard C. Elman, David J. Silvester, and Andrew J. Wathen. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford, second edition, 2014.
 - [14] Merike Fiedler, Werner Berg, Christian Ammon, Christiane Loebstin, Peter Sanftleben, Mohamed Samer, Kristina von Bobrutzki, Alaa Kiwan, and Chayan K Saha. Air velocity measurements using ultrasonic anemometers in the animal zone of a naturally ventilated dairy barn. *Biosystems engineering*, 116(3):276–285, 2013.
 - [15] Merike Fiedler, J Fischer, S Hempel, C Saha, C Loebstin, W Berg, B Amon, R Brunsch, and T Amon. Flow fields within a dairy barn-measurements, physical modelling and numerical simulation. In *Proceedings. International Conference of Agricultural Engineering AgEng*, pages 1–5, 2014.
 - [16] Merike Fiedler, Gundula Hoffmann, Christiane Loebstin, Werner Berg, Kristina von Bobrutzki, Christian Ammon, and Thomas Amon. Luftgeschwindigkeit und hitzebelastung im milchviehstall-auswirkungen auf das tierwohl. *Landtechnik*, 67(6):421–424, 2012.
 - [17] S. Ganesan, V. John, G. Matthies, R. Meesala, S. Abdus, and U. Wilbrandt. An object oriented parallel finite element scheme for computing pdes: Design and implementation. In *IEEE 23rd International Conference on High Performance Computing Workshops (HiPCW) Hyderabad*, pages 106–115. IEEE, 2016.
 - [18] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *Internat. J. Numer. Methods Engrg.*, 79(11):1309–1331, 2009.
 - [19] David Gisen. Generation of a 3d mesh using snappyhexmesh featuring anisotropic refinement and near-wall layers. In *ICHE 2014. Proceedings of the 11th International Conference on Hydrosience & Engineering*, pages 983–990, 2014.
 - [20] Sabrina Hempel, Marcel König, Christoph Menz, David Janke, Barbara Amon, Thomas M Banhazi, Fernando Estellés, and Thomas Amon. Uncertainty in the measurement of indoor temperature and humidity in naturally ventilated dairy buildings as influenced by measurement technique and data variability. *Biosystems Engineering*, 166:58–75, 2018.

- [21] Sabrina Hempel, Lina Wiedemann, Christian Ammon, Merike Fiedler, Chayan Saha, David Janke, Christiane Loebstin, Jost Fischer, Barbara Amon, Gundula Hoffmann, et al. Determine the through-flow characteristics of naturally ventilated dairy barns to optimise barn climate. In *Bau, Technik und Umwelt in der landwirtschaftlichen Nutztierhaltung 2015*, pages 346–351. Darmstadt: Kuratorium für Technik und Bauwesen in der Landwirtschaft eV, 2015.
- [22] Tobias Holzmann. *Mathematics, Numerics, Derivations and OpenFOAM®*. Holzmann CFD, 2016.
- [23] Se-Woon Hong, Vasileios Exadaktylos, In-Bok Lee, Thomas Amon, Ali Youssef, Tomas Norton, and Daniel Berckmans. Validation of an open source cfd code to simulate natural ventilation for agricultural buildings. *Computers and Electronics in Agriculture*, 138:80–91, 2017.
- [24] Raad I Issa. Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of computational physics*, 62(1):40–65, 1986.
- [25] M Jähn, O Knoth, M König, and U Vogelsberg. Asam v2. 7: a compressible atmospheric model with a cartesian cut cell approach. *Geoscientific Model Development Discussions*, 7(4), 2014.
- [26] Volker John. *Finite element methods for incompressible flow problems*, volume 51 of *Springer Series in Computational Mathematics*. Springer, Cham, 2016.
- [27] Rack-woo Kim, Se-woon Hong, In-bok Lee, and Kyeong-seok Kwon. Evaluation of wind pressure acting on multi-span greenhouses using cfd technique, part 2: Application of the cfd model. *biosystems engineering*, 164:257–280, 2017.
- [28] Rack-woo Kim, In-bok Lee, and Kyeong-seok Kwon. Evaluation of wind pressure acting on multi-span greenhouses using cfd technique, part 1: Development of the cfd model. *Biosystems engineering*, 164:235–256, 2017.
- [29] Marcel König, Sabrina Hempel, David Janke, Barbara Amon, and Thomas Amon. Variabilities in determining air exchange rates in naturally ventilated dairy buildings using the co2 production model. *Biosystems Engineering*, 174:249–259, 2018.
- [30] In-bok Lee, Sadanori Sase, and Si-heung Sung. Evaluation of cfd accuracy for the ventilation study of a naturally ventilated broiler house. *Japan Agricultural Research Quarterly: JARQ*, 41(1):53–64, 2007.
- [31] Luciano B Mendes, Gerlinde De Vogeleeer, Philippe Van Overbeke, Peter Demeyer, and Jan G Pieters. Effects of standard ke and les turbulence models on a full scale numerical cfd simulation for a naturally ventilated pig barn prototype. In *2015 ASABE Annual International Meeting*, page 1. American Society of Agricultural and Biological Engineers, 2015.
- [32] Andrea Montorfano. Mesh generation for hpc problems: the potential of snappyhexmesh, 06 2017.
- [33] Tomas Norton, Jim Grant, Richard Fallon, and Da-Wen Sun. Assessing the ventilation effectiveness of naturally ventilated livestock buildings under wind dominated conditions using computational fluid dynamics. *Biosystems Engineering*, 103(1):78–99, 2009.
- [34] OpenFOAM. OpenFOAM Web page. <http://openfoam.org/>, 2016.
- [35] Suhas V Patankar and D Brian Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. In *Numerical Prediction of Flow, Heat Transfer, Turbulence and Combustion*, pages 54–73. Elsevier, 1983.
- [36] Stephen B. Pope. *Turbulent flows*. Cambridge University Press, Cambridge, 2000.
- [37] Li Rong, Peter V Nielsen, Bjarne Bjerg, and Guoqiang Zhang. Summary of best guidelines and validation of cfd modeling in livestock buildings to ensure prediction quality. *Computers and Electronics in Agriculture*, 121:180–190, 2016.

- [38] Youcef Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Comput.*, 14(2):461–469, 1993.
- [39] Chayan Kumer Saha, Wentao Wu, Guoqiang Zhang, and Bjarne Bjerg. Assessing effect of wind tunnel sizes on air velocity and concentration boundary layers and on ammonia emission estimation using computational fluid dynamics (cfd). *Computers and Electronics in Agriculture*, 78(1):49 – 60, 2011.
- [40] Xiong Shen, Guoqiang Zhang, Wentao Wu, and Bjarne Bjerg. Model-based control of natural ventilation in dairy buildings. *Computers and Electronics in Agriculture*, 94:47 – 57, 2013.
- [41] Hang Si. Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Trans. Math. Softw.*, 41(2):11:1–11:36, February 2015.
- [42] J. Smagorinsky. General circulation experiments with the primitive equations. *Mon. Wea. Rev.*, 91:99–164, 1963.
- [43] The OpenFOAM Foundation. Openfoam user guide version 4.0, 2016.
- [44] Ulrich Wilbrandt, Clemens Bartsch, Naveed Ahmed, Najib Alia, Felix Anker, Laura Blank, Alfonso Caiazzo, Sashikumaar Ganesan, Swetlana Giere, Gunar Matthies, Raviteja Meesala, Abdus Shamim, Jagannath Venkatesan, and Volker John. ParMooN—A modernized program package based on mapped finite elements. *Comput. Math. Appl.*, 74(1):74–88, 2017.
- [45] Qianying Yi, Marcel König, David Janke, Sabrina Hempel, Guoqiang Zhang, Barbara Amon, and Thomas Amon. Wind tunnel investigations of sidewall opening effects on indoor airflows of a cross-ventilated dairy building. *Energy and Buildings*, 175:163–172, 2018.
- [46] Qianying Yi, Guoqiang Zhang, Marcel König, David Janke, Sabrina Hempel, and Thomas Amon. Investigation of discharge coefficient for wind-driven naturally ventilated dairy barns. *Energy and Buildings*, 165:132–140, 2018.
- [47] Akira Yoshizawa. Statistical theory for compressible turbulent shear flows, with the application to subgrid modeling. *Phys. Fluids*, 29:2152–2164, 1986.