A machine-learning-based method for automatizing lattice-Boltzmann simulations of respiratory flows

Mario Rüttgers^{1,2,3} · Moritz Waldmann^{1,3} · Wolfgang Schröder^{1,3} · Andreas Lintermann^{2,3}

Accepted: 19 August 2021 / Published online: 1 January 2022
© The Author(s) 2022

Abstract

Many simulation workflows require to prepare the data for the simulation manually. This is time consuming and leads to a massive bottleneck when a large number of numerical simulations is requested. This bottleneck can be overcome by an automated data processing pipeline. Such a novel pipeline is developed for a medical use case from rhinology, where computer tomography recordings are used as input and flow simulation data define the results. Convolutional neural networks are applied to segment the upper airways and to detect and prepare the in- and outflow regions for accurate boundary condition prescription in the simulation. The automated process is tested on three cases which have not been used to train the networks. The accuracy of the pipeline is evaluated by comparing the network-generated output surfaces to those obtained from a semi-automated procedure performed by a medical professional. Except for minor deviations at interfaces between ethmoidal sinuses, the network-generated surface is sufficiently accurate. To further analyze the accuracy of the automated pipeline, flow simulations are conducted with a thermal lattice-Boltzmann method for both cases on a high-performance computing system. The comparison of the results of the respiratory flow simulations yield averaged errors of less than 1% for the pressure loss between the in- and outlets, and for the outlet temperature. Thus, the pipeline is shown to work accurately and the geometrical deviations at the ethmoidal sinuses to be negligible.

Keywords Nasal cavity flow · Semantic segmentation · Convolutional neural networks · Thermal lattice-Boltzmann method · High-performance computing

1 Introduction

Understanding fluid dynamics has become key to advance technologies in a variety of fields in industry and academia. Frequently, computational fluid dynamics (CFD) methods are employed to simulate and analyze complex systems involving all kinds of flows at different scales. Such simulations provide physical quantities that help to design and optimize industrial products, e.g., in transportation

drag is reduced [1] or the origin of combustion noise excited from jet flames is determined [2]. In these two examples, surfaces that are used as domain boundaries for the simulations are extracted from technical drawings. Some applications, however, require the analysis of the flow through already existing real-life objects. For instance, in the production of cement, CFD computations are used to estimate porosity and permeability of samples [3]. The geometrical information of the samples are extracted by computer tomography (CT). Their capabilities of non-destructively measuring objects makes this technique a frequently used tool to extract geometric and/or volumetric data of components made of all kinds of materials.

The integration of quality control processes into industrial workflows to find optimal product properties results in many CT recordings for which a large number of numerical simulations needs to be conducted iteratively. The computational time scales with the desired resolution of the simulation. To achieve satisfying results, simulations frequently have to run on hundreds to thousands of

✉ Mario Rüttgers
M.Ruettgers@aia.rwth-aachen.de

¹ Institute of Aerodynamics and Chair of Fluid Mechanics, RWTH Aachen University, Aachen, Germany

² Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, Jülich, Germany

³ Jülich Aachen Research Alliance - Center for Simulation and Data Sciences (JARA-CSD), RWTH Aachen University and Forschungszentrum Jülich GmbH, Germany

computational cores for hours or even days or weeks. Furthermore, preparing and setting up each simulation may consume a lot of time. Depending on the simulation framework and the simulation requirements, the geometry needs to be prepared, in- and outflow regions need to be defined, and a computational mesh needs to be generated. Many of these steps may involve manual intervention of the user and hence are not well suited for mass processing. That is, there is a need for an automated process pipeline that takes CT data as input and provides simulation results as output.

In this work, such a pipeline is developed for an example coming from the health sector, i.e., for the simulation of respiratory flows in the human nasal cavity and the pharynx. The upper respiratory airway is a complex and intricate system, see Fig. 1. It is responsible for olfaction and degustation, filtering incoming air, and moisturizing and tempering the air before guiding it to the lungs [4]. The air enters the nostrils (See Fig. 1a), and passes the inferior, middle, and superior turbinates (See Fig. 1b). The left and right channels converge in the pharynx (See Fig. 1a). The frontal, ethmoidal, and maxillary sinuses (See Fig. 1a) function as resonance bodies in the phonation process.

The structure of the nasal cavity - on a macroscopic scale - is the same for all human beings, but on a microscopic scale specific to an individual. Therefore, it is a perfect example to show the applicability and to test the performance of the developed pipeline. It should be noted that the pipeline is in principle also applicable to flows through any kind of geometry that has been measured with CT systems.

To understand how CFD methods are applied to analyze the flow in the respiratory tract, and which flow properties are relevant for ear, nose, and throat (ENT) specialists in treating patients, an introduction to this topic is subsequently given. Obstructions of the upper airway can impair the sense of smell or taste, reduce the capabilities to lung-sensitively prepare inhaled air, or increase the energy needed to breathe comfortably. Frequently, a medical intervention is the only remaining option to alleviate the patient's complaints. The corresponding treatment decisions are usually based on medical images, the patient history, and the experience of the physician in charge. Fluid mechanical properties such as the pressure loss, the heating capability, or the wall-shear stress are, in general, not considered. As

Fig. 1 Coordinate system and examples for coronal (yellow), sagittal (purple), and axial (blue) plane views on CT imaging data showing the human upper airway. A segmentation separating the airway from other matter is exemplarily illustrated in green

a consequence, patients are quite often dissatisfied by the outcome of surgical and medical treatments. For example, the study in [5] reveals that only 55.9% of patients are satisfied after a septoplasty, a surgical intervention to correct a deviated nasal septum. Conducting respiratory flow simulations is expected to play a major role in improving diagnosis and treatments in rhinology. Vanhille et al. [6] conduct a survey among surgeons and investigate the acceptance of a virtual surgery planning tool that is based on numerical simulations. On a scale of 1 (strongly disagree) to 7 (strongly agree), a mean score of 5.7 is found. This shows that numerical simulations are quite attractive for the medical community, i.e., they may become a common diagnostic method integrated into clinical environments. However, simulations are only practicable, if they can be performed automatically and without prior technical knowledge of the user.

In this study, an automated data pipeline which uses CT images as input and provides highly accurate simulation results is presented. The pipeline is based on multiple ground data filtering techniques and ML algorithms. To estimate the quality of the nasal cavity surface, the corresponding simulation results are compared to results obtained from using a surface based on a semi-manual segmentation process supported by medical professionals. The focus of this study lies on the ML-based surface generation process. The analysis of nasal cavity flows is of secondary importance. The manuscript is structured as follows. In Section 2, the individual steps of the new pipeline and related studies to these steps are presented. Section 3 describes the methods that are used to realize the individual steps. This is followed by a simulation code validation, a comparative performance study, and an analysis of the flow in a nasal cavity in Section 4. A summary and conclusions are given together with an outlook in Section 5.

2 Related works

The preparation of numerical simulations and the analysis of the results requires multiple interdisciplinary steps:

- (i) segmentation of a medical image data into airway and other matter;
- (ii) extraction of an airway surface;
- (iii) subdivision of an airway surface into inflow, outflow, and wall boundary surfaces;
- (iv) generation of a computational mesh;
- (v) numerical simulation;
- (vi) analysis of the simulation results.

An overview over related studies to individual steps and all steps together as a complete pipeline are given below.

2.1 Machine-learning-based segmentation

Machine learning techniques are discussed in the context of steps (i) and (iii) of the data processing pipeline. Both steps are related to semantic segmentation of medical images. Recently, convolutional neural networks (CNN) have shown great potential in detecting geometrical features from images. Long et al. [7] have been the first to develop a fully convolutional network to perform end-to-end image segmentation. But the sudden increase of the output at the final layer negatively affects the predictive quality of the network. Ronneberger et al. [8] overcame this challenge by applying convolutional encoder layers and deconvolutional decoder layers. They employed skip connections between encoder and decoder layers, to preserve information from features of all scales. However, concatenating information after a long skip connection leads to an increased network size. In [9, 10] the long skip connections have residual form, adding the activity of encoder layers to the output of decoder layers, allowing a faster training. Hu et al. [11] developed this approach further to successfully perform volumetric segmentation on three-dimensional medical recordings. In this study, fully convolutional encoder-decoder type networks are trained to automatically perform two- and three-dimensional segmentations on CT data. These types of networks have proven to accurately realize segmentation tasks in medical images.

2.2 Numerical analysis of respiratory flows

Numerical methods are thematized that are relevant for step (v) of the pipeline. Previous studies underline that fluid mechanical properties play a key role in analyzing nasal respiration. Lintermann et al. [12] use a lattice-Boltzmann (LB) method integrated into the simulation framework multi-physics Aerodynamisches Institut Aachen (m-AIA) (former name: Zonal Flow Solver (ZFS)) [13] to simulate the flow in the nasal cavity. They analyze the flow to classify nasal cavities into ability groups and thereby support a priori surgical intervention decision processes. For some cases, the subjective respiration evaluation is, however, not in line with the numerical findings, which emphasizes the importance of correlating the numerical results to the patient's perception. In the study of Waldmann et al. [14], the same LB method is applied to investigate effects of a partial widening of the maxilla, which is a common therapy for patients suffering from maxillary constriction, on respiration. The simulation results show that the treatment not only leads to a decrease of the respiratory resistance but also to a reduction of the heating capability. Zhang et al. [15] evaluate treatments for atrophic rhinitis, an enlarged nasal space usually caused by destructive nasal surgeries and often accompanied by a nasoseptal perforation. Narrowing the nasal cavity

and repairing the perforation reduces the wall-shear stress near the perforation, which lowers the probability of re-perforation. Furthermore, more air is guided past the olfactory organ and less along the nasal floor, resulting in an improved smell sensation. However, nasal resistance is increased by the treatments. Calmet et al. [16] study rapid inhalation with a large-eddy simulation (LES). An LES uses sub-grid scale (SGS) models such as the Smagorinsky [17] or the dynamic Smagorinsky [18] SGS approaches to model unresolved turbulent scales in the regime above the inertial subrange. They show that rapid inspiration causes a higher level of turbulence in the pharynx as compared to flow in the nasal cavity.

2.3 Automatizing simulations of respiratory flows

Various approaches for an automatic data processing pipeline covering all steps (i)–(vi) have been developed in the past. Burgos et al. [19] present an automated tool for virtual nasal surgeries that uses CT or magnetic resonance imaging (MRI) data as input data. The segmentation task is performed by thresholding those pixels in medical images that represent air. A large air volume outside of the upper airway is included into the computational domain to let flow of incoming air develop naturally. The tool allows the results to be visualized from numerical simulations of potential surgical maneuvers. For steady-state simulations of laminar flow the Navier-Stokes equations are solved in a mesh containing approx. 7 million cells in less than 6 hours on a personal computer with 4 cores [20]. Huang et al. [21] introduce a segmentation framework that is optimized to reduce the number of manual steps needed to conduct numerical simulations. Segmentation is performed using a statistical shape modeling method. Entrances and the exit of the nasal model are detected using spatial and anatomical information. They are artificially extended to clearly establish the inlet and outlet locations. The mesh generation and flow simulation are conducted with the software ANSYS by solving the Reynolds-averaged Navier-Stokes (RANS) equations with a k - ϵ turbulence model [22]. In the studies of Eitel et al. [23] and Lintermann et al. [4], nasal cavity flow is simulated using the above mentioned LB method. Nasal geometries are extracted from CT data in a semi-manual reconstruction pipeline. Segmentation is performed automatically by specifying a threshold between tissue and air, and applying a seeded region growing algorithm. A marching cubes algorithm [24] then extracts the three-dimensional representation of the air/tissue interface. Inlet and outlet regions need, however, to be detected manually by defining bounding planes that prevent the region growing algorithm to reach into the mouth, the lungs, or the outer cranium volume. Inthavong et al. [25] also perform a semi-manual segmentation. The

surface is generated by joining cloud points provided by the segmentation. It is manually subdivided into different boundary regions. Simulations are conducted with a commercial code without the usage of turbulence models. Koch et al. [26] use machine learning (ML) techniques for an automated segmentation task. The remaining steps of a simulation pipeline are, however, not reported.

The key factors for the acceptance of the process from (i) to (vi) as diagnostic method are usability and accuracy. The first factor requires to not have manual steps such as manual segmentation in the processing pipeline. Such tasks are tedious and error-prone, i.e., a perfect usability is only given if all steps can be performed automatically. Then, the user does not need to deal with questions as how to run a simulation and what to provide as input to successfully execute a simulation and analyze the output data. A tool that might be used to support treatment decisions must deliver accurate results, i.e., the second factor aims at satisfying the physician in charge and the patient.

The aforementioned pipeline approaches are either not fully automated or their simulation accuracy is not sufficient. Simulations based on solving the RANS equations and using turbulence models such as the k - ϵ , k - ω [27], or the k - ω *SST* [28] models with fixed model constants were developed for fully turbulent flow. The model constants are in general only valid for specific flow problems and need to be adapted from case to case. Furthermore, multiple studies [12, 29, 30] show that the flow in the nasal cavity for respiration at rest is rather in the laminar to transitional regime and that turbulent flow, if at all, is only a local phenomenon. Hence, artificial errors are introduced by using RANS with turbulence models, especially in laminar to transitional regions. Despite these drawbacks, RANS methods have gained popularity due to their minimal computational costs. Similar statements can be made for LES. Various studies show that LES computations are superior over RANS methods. Nevertheless, they also come at higher costs and the SGS may again introduce artificial errors. The most accurate results can be obtained by not using any turbulence model, i.e., to fully resolve all relevant flow scales by running direct numerical simulations (DNS).

For the LB simulations in this study, the DNS approach is used. Although such simulations are more expensive than RANS or LES computations and require high-performance computing (HPC) systems to efficiently execute a simulation, their possible accuracy is an inherent plus. Considering that a successful surgical pathology treatment is important for both the patient and the surgeon and that many surgical interventions fail, high accuracy as the key feature of DNS is a must in such processing pipelines. The computational mesh is generated automatically using the method from [31].

3 Computational methods

In Section 3.1, the ML-based algorithms for the segmentation of CT image data and for the generation of the three-dimensional airway model are presented. This covers steps (i) to (iii) of the pipeline introduced in Section 2. Subsequently, the mesh generation process and the simulation method are described in Section 3.2, covering steps (iv) and (v) of the pipeline.

3.1 ML-based segmentation and 3D model generation

The flow chart in Fig. 2 provides an overview over the single components of steps (i) to (iii) of the pipeline. The input to step (i) are CT image data, which provide a three-dimensional volumetric snapshot of a patient. The volume is split into $m \times m \times n$ so called voxels, where m is the number of in-plane pixels in the x- and y-direction, and n is the

Fig. 2 Flow chart illustrating steps (i)–(iii) of the pipeline and its components

Fig. 3 Hounsfield values HU from CT recordings are compared to the same data filtered with a convolutional filter Φ and with a combination of Φ and a gradient anisotropic diffusion filter Ψ , along a line illustrated in Fig. 1b. The red line is the HU threshold $\Gamma = -550$ HU separating airway and tissue

number of images in the z -direction. An example for the corresponding coordinate system, its origin, and coronal, sagittal, and axial slices centered at the data set center is shown in Fig. 1. The reconstructed surface of the nasal cavity is illustrated in Fig. 1d. The segmentation on coronal, sagittal, and axial in-plane views are shown in Fig. 1a, b and c. Each voxel contains information on the density field given in Hounsfield units $HU \in [-1024, 3071]$. Different HU values and ranges correspond to different material properties, e.g., air and water are found at $HU = -1000$ and $HU = 0$.

A straightforward approach to segment the image into non-air and air volumes is to perform seeded region growing as used, e.g., in [4, 23]. Therefore, an HU threshold Γ for voxels representing air is defined and a region growing is performed that keeps all voxels with $HU \leq \Gamma$. This approach, however, entails three challenges:

1. It is difficult to define a global threshold that guarantees a correct segmentation of the complete airway, i.e., narrow channels may lead to false segmentation results.
2. To accurately prescribe the boundary conditions for a numerical simulation, cf. Section 3.2.2, it is necessary to exclude voxels outside of the nasal cavity representing ambient air.
3. The process can only be performed semi-automatically.

These challenges are tackled in the subsequent descriptions of the steps (i), (ii), and (iii).

3.1.1 Step (i) of the automated pipeline: segmentation

The first challenge is tackled by pre-filtering the CT data. Two types of filters, see step (i) in Fig. 2, are applied to clearly separate air and non-air voxels. Similar to [23], a convolutional filter Φ is used to increase the HU gradients at air-tissue interfaces. The line plot in Fig. 3 juxtaposes the

raw CT data to the Φ -filtered data. The line is positioned parallel to the y -axis and covers 200 voxels, as illustrated in Fig. 1b. The index $\iota \in 1, \dots, 200$ represents the voxel index along this line. Obviously, Φ not only increases the HU value difference at locations with strong HU gradients, but also introduces high HU fluctuations off the interface. Applying a standard threshold of $\Gamma = -550$ HU [32] to this filtered data set leads to false segmentations. This is, e.g., the case in the vicinity of $\iota = 85$ or $\iota = 195$ in Fig. 3, where $\Gamma = -550$ HU is marked by a red line. To overcome this problem, a gradient anisotropic diffusion filter Ψ that smoothens unwanted gradients is applied. This filter employs the classic Perona-Malik, gradient magnitude-based equation to reduce image noise without removing edges [33]. The complete filtering operation for an axial slice is then given by $\Xi = \Phi \circ \Psi$.

The second and third challenges are addressed by utilizing ML techniques. A CNN, denoted as CNN-A, is trained in a supervised manner to segment all voxels that represent the human airway. The data set used for training and testing CNN-A includes 67 CT recordings.¹ All recordings have a constant number of $m = 512$, i.e., each axial slice contains 262,144 pixels. The number of axial slices of the various recordings varies with $n \in [119, 588]$. The pixel width and length δ_{xy} are equal for each image of a CT recording. It is in the range of $\delta_{xy} \in [0.3 \text{ mm}, 0.9 \text{ mm}]$ for the considered data set. The distance between the axial slices in the z -direction δ_z is given by $\delta_z \in [0.2 \text{ mm}, 2.0 \text{ mm}]$. The data set is split into 64 CT recordings for training (15,213 axial slices) and 3 recordings for testing (211, 209 and 248 axial slices), denoted as $\tau_1, \dots, \tau_3$. In the training process, CNN-A takes axial slices κ_n with index n as input in form

¹ All data have been anonymized before being processed for training or testing.

of training data I_{sl} . The input at location $q, r \in 1, \dots, m$ for κ_n is calculated by

$$I_{sl}(\kappa_n(q, r)) = \frac{\Xi(\kappa_n(q, r)) - \bar{\Xi}}{\hat{\Xi}}, \quad (1)$$

where $\bar{\Xi}$ and $\hat{\Xi}$ are the mean and the standard deviation of the filtered pixel values derived from all axial slices of the training data. The ground truth data are generated by applying Ξ to the CT recordings, keeping filtered voxels with HU values below $\Gamma = -550 HU$, and using the software 3D Slicer [34] to cut the remaining structure at the interface between ambient air and the entrance to the nasal cavity. In the numerical simulation, inlet boundary conditions, cf. Section 3.2.2, need to be imposed at these interfaces. Inlet geometries have to be chosen such that their surface is flat and that the face normal points into the streamwise direction of the flow. Cavities in regions that are not related to the airflow, e.g., ear cavities, are excluded from the ground truth. The green-colored voxels depicted in Fig. 1 are ground truth examples. During the testing process, inputs for $\tau_1, \dots, \tau_3$ are calculated according to Eq. 1.

The architecture of CNN-A is depicted in Fig. 4. It is part of step (i), see Fig. 2. It is inspired by an encoder-decoder-type U-net architecture [8], which has successfully been used for medical image segmentation. To preserve information from features on different scales, skip connections between encoder and decoder layers are employed. Skip connections allow for direct gradient flow from higher to lower layers across all hierarchy stages during the backward pass. This prevents common issues with vanishing gradients in deep architectures [9, 10]. In a convolutional block (CB), a sequence of a two-dimensional convolutional layer, batch normalization (BN), and a

rectified linear unit (ReLU) activation function is repeated twice. All convolutional layers operate with 3×3 kernels. The BN has a regularizing effect by shifting activity of layers to zero mean and unit variance. This leads to a faster and more reliable network convergence [35]. In the encoder path, downscaling is performed by 2×2 maximum pooling layers (MP). In the decoder path, upscaling is realized by two-dimensional deconvolutional (DeConv) layers. Dropout layers (DO) are employed during training with a DO probability of $P = 0.5$ to further avoid overfitting [36]. A sigmoid activation function makes sure that values in the final layer are in the range of $FL(q, r) \in [0; 1]$. To improve training, samples are randomly mirrored along the y-axis, taking advantage of the symmetry of the human head. When testing, the segmentation SG is obtained from

$$SG(q, r) = \begin{cases} 1, & FL(q, r) \geq 0.5 \\ 0, & FL(q, r) < 0.5. \end{cases} \quad (2)$$

At training, the output of the binary cross-entropy loss function

$$L = -\frac{1}{m \times m} \sum_{q=1}^m \sum_{r=1}^m GT(q, r) \cdot \log FL(q, r) + (1 - GT(q, r)) \cdot \log (1 - FL(q, r)) \quad (3)$$

is minimized, where GT stands for the ground truth data. Weights and biases are initialized from a truncated normal distribution centered around the origin with a standard deviation of $\sigma_{std} = \sqrt{2/\mathcal{C}}$, where $\mathcal{C}$ is the number of connections in a layer [37]. The weights and biases are updated by an adaptive moments (ADAM) optimizer [38]. The ADAM optimizer considers an exponentially decaying average of gradients computed in previous update steps

Fig. 4 Architectures of CNN-A, CNN-B, and CNN-C with input and output dimensions, and numbers of feature maps (FM) for each convolutional block (CB)

when adjusting the learning rate. Preliminary studies revealed that a batch size of $BS = 5$ achieves best results.

Before extracting nasal cavity surfaces, i.e., performing step (ii) of the pipeline, all predicted axial slices SG are concatenated along the z -axis, yielding the volume SG_{vol}^A . Special emphasis has to be put on the treatment of the inflow areas. The CNN-A has no information on the coherence of different regions in a three-dimensional view. This lack of information causes imperfect inflow regions when the axial slices are stacked together. Therefore, another CNN, denoted as CNN-B, that learns only to segment the airway at the inflow region is applied. The architecture of CNN-B is shown in Fig. 4. It only differs from CNN-A in terms of its dimensionality. Whereas CNN-A uses axial slices and two-dimensional convolutional or pooling layers, CNN-B is fed with $80 \times 96 \times 48$ cubes and is equipped with three-dimensional layers. At training, input cubes I_{cu} are obtained from the filtered and normalized volume data from each of the 64 CT recordings. A cube contains left and right nostrils. The corresponding ground truth cubes are extracted from the semi-manual segmentations that have been used as ground truth for training CNN-A. Since each CT recording provides only a single cube, translation and rotation data augmentation techniques are necessary to artificially enlarge the training data set. This yields a number of 37,421 training cubes. When testing CNN-B in the processing pipeline, the location of the cube for $\tau_1, \dots, \tau_3$ is specified by SG_{vol}^A . That is, the first segmented voxel in the x -direction, which is always inside of the left or right nostril, is detected. Cubes for $\tau_1, \dots, \tau_3$ are then constructed around this voxel. Next, the output of CNN-B replaces those values of SG_{vol}^A that are located inside the cube, yielding SG_{vol}^B . Alternatively to using CNN-A and CNN-B, a three-dimensional CNN with cubes distributed randomly in the CT volume data could be used. However, segmentations predicted with such an approach were found to be not smooth enough at interfaces between some of the cubes.

3.1.2 Step (ii) of the automated pipeline: surface extraction

Step (ii) of the pipeline, see Fig. 2, starts with a so called keep-largest-island (KLI) algorithm that is applied to SG_{vol}^B . The KLI step keeps only the largest coherent part of the segmentation and its output is denoted as SG_{kli} . Before extracting the nasal cavity surface, SG_{kli} is modified to yield

$$SG_{mod}(\kappa_n(q, r)) = \begin{cases} HU_{\min}, & SG_{kli}(\kappa_n(q, r)) = 0 \\ HU_{\max}, & SG_{kli}(\kappa_n(q, r)) = 1 \\ HU(q, r), & SG_{kli}(\kappa_n(q, r)) = 0 \\ \wedge SG_{kli}^N(\kappa_n((q, r)) = 1, \end{cases} \quad (4)$$

with $HU_{\min} = -1024$, $HU_{\max} = 3071$, and $SG_{kli}^N(\kappa_n((q, r))$ representing a neighboring voxel of the voxel at $SG_{kli}(\kappa_n(q, r))$. The following actions of step (ii) are inspired by the pipeline described in [4, 23]. That is, the marching cubes algorithm [24] is applied to SG_{mod} with the threshold $\Gamma = -550 HU$. The resulting surface is a three-dimensional triangulated model of the human upper airway, which is finally smoothed by a windowed sync filter (WSF) [39].

3.1.3 Step (iii) of the automated pipeline: surface subdivision

In step (iii) of the pipeline, the smoothed three-dimensional surface is subdivided into left inflow, right inflow, outflow, and wall subsurfaces. The starting point for step (iii) is a nasal cavity surface that is generated based on a segmentation done by CNN-A and CNN-B. A corresponding example is depicted in Fig. 5a. Each triangle of the nasal cavity surface needs to be distributed to one of the four subsurfaces to yield a set of surfaces as shown in Fig. 5b. However, imperfections existing at the inflow areas remaining from the smoothing process, see Fig. 5a, and open regions at

Fig. 5 Example of a nasal cavity surface

the outflow location, see Fig. 5c, make such a distribution impossible. Without planar inflow and outflow surfaces, triangles cannot be identified by a common face normal. Therefore, locations and face normals of inflow and outflow areas are detected with a third CNN, denoted as CNN-C. The architecture of CNN-C, shown in Fig. 4, differs from CNN-A in terms of the input and output data. Whereas CNN-A is only trained with I_{sl} , the input to CNN-C has two channels. The first channel is fed with I_{sl} and the second channel with axial slices of the ground truth used for training CNN-A. When testing the pipeline, input data to the first channel match with those for CNN-A, and input to the second channel are slices of the generated volume SG_{vol}^B . The ground truth data of CNN-C consists of two channels, one for each nostril. Examples for the various ground truth data are shown in Fig. 6. CNN-C learns to predict a thin layer at each nostril that provides spatial information on the corresponding inflow area. When predicting these layers, CNN-C is not fed with all axial slices of a CT recording. To accelerate the prediction process, only a certain number of slices $n_{lim} = L_{n,norm} \cdot L / \delta_z$ near the nostrils are taken. The quantity L is the model length in x -direction and $L_{n,norm} = L_n / L$ is the length that is spanned by n_{lim} normalized by L . Tests have shown that setting $L_{n,norm} = 0.4$ captures both nostrils of all three cases best. The length scales L and L_n are shown in Fig. 6. To get the positions along the z -axis of these slices, first the location of the voxel of SG_{vol}^B with the lowest x -coordinate is computed. Note that $n_{lim}/2$ slices above and below the z -coordinate of this voxel are used as input to CNN-C.

To guarantee a full cut of the nostrils, the intersection planes are moved along the nostril inside-pointing face normals by a distance L_{inl} . An example for L_{inl} at the left inlet is shown in Fig. 6c. Starting at an initial distance L_{inl}^0 , the inlets may consist of more than one closed surface due to the aforementioned smoothing imperfections. To avoid this, a region growing algorithm is embedded into a feedback

loop that subsequently increases L_{inl} by L_{inl}^{step} until both inlets consist of only a single closed surface. This process is shown in Fig. 2 as part of step (iii). The distances are set by the normalized quantities $L_{inl,norm}^0 = L_{inl}^0 / L$ and $L_{inl,norm}^{step} = L_{inl}^{step} / L$. In this study, $L_{inl,norm}^0 = 4.5 \cdot 10^{-3}$ and $L_{inl,norm}^{step} = 8 \cdot 10^{-4}$ have proven to work best.

Nostrils of different patients have similarities in their shapes, which allows a CNN to predict the inflow area by learning to detect features of the nostrils. In contrast, the pharynxes of different patients do not have clearly identifiable features, i.e., CNNs cannot be used to predict the position and face normal direction of the outflow area. The main reason for this is the spatial limit of CT recordings in the z -direction. The first axial slice at $z = 0$ can be located somewhere between the nasopharynx and the hypopharynx. To conserve as many information from CT recordings as possible, the simplest solution for defining the outflow area is to use the hole at the lower end of the pharynx. An example is shown in Fig. 5c. This, however, does not guarantee that the face normal of the outflow area points into the streamwise direction. To account for this, the concept of the geometrical centerline is used to extract the outflow surface. Therefore, the two centerlines starting at the left and right nostrils, see Fig. 7, are computed using the vascular modeling toolkit (VMTK) [40]. Both centerlines end at the lower end of the pharynx. The left and right centerline are averaged to yield a new centerline, which provides a good estimate of the main flow direction in the pharynx. The origin of the outflow area is found at the end of the averaged centerline and the face normal vector is calculated from two consecutive centerline points.

Having determined the locations and the face normal directions of the inflow and outflow areas, the nostrils and pharynx are cut and closed with flat surfaces using the pymeshfix library [41]. Subsequently, the triangles of the complete surface are distributed to one of the four subsurfaces. Depending on the z limit of the CT, the outflow

Fig. 6 Examples for the ground truth data for CNN-C (left inlet: yellow, right inlet: brown), CNN-A and CNN-B (green), and length scales L , L_n , and L_{inl}

Fig. 7 Example of centerlines starting at the left (red) and right (blue) nostrils and ending at the lower end of the pharynx

area might consist of multiple surfaces, cf. Fig. 5c. To circumvent this, the process of cutting, closing holes, and distributing triangles is embedded into a conditional loop, see step (iii) in Fig. 2. Similar to the loop used for the inlet areas, a region growing algorithm is used to check whether all voxels belonging to the outflow area are connected. If this is not satisfied, the pharynx is cut again at a higher position $z_h > 0$ of the averaged centerline with the corresponding face normal. The process is controlled by a pre-defined starting point from the bottom of the averaged centerline M_{start} , and a range of points that is skipped at each iteration M_{step} .

3.2 Flow simulation using a thermal lattice-Boltzmann method

The mesh generation approach and the simulation method are elements of the multiphysics simulation framework m-AIA. Inputs for both elements are the triangulated subsurfaces that have been described in the previous section. Details about both processes are given below.

3.2.1 Step (iv) of the automated pipeline: mesh generation

The generation of the computational mesh employs the massively parallel grid generator of m-AIA [13]. In parallel, it reads the 3D airway model and stores its triangle in an alternating digital tree (ADT) [42] for accelerated triangle searches and retrieval. On each processor, an initial cube, subsequently referred to as cell, is placed around the geometry with minimum extents. This cell on refinement level l_0 is then successively refined, i.e., each cell is subdivided into eight child cells. The parent-child relation is stored in an octree. At each refinement level, cells outside the geometry are removed. This is done by keeping track of the cells intersecting the geometry and by applying a recursive region growing algorithm. A random cell inside the fluid domain is selected as starting point of the region growing process. All cells not covered by this recursive

procedure are outside cells and are hence removed from the memory and the octree. This way, the mesh is refined to a user-defined start-level l_α , which requires to have more cells than processors used later in the simulation. At l_α , all cells on levels $l < l_\alpha$ are removed and a Hilbert space-filling curve [43] is placed into the remaining cells. The Hilbert identifier is employed to evenly distribute the set of cells among the processors. Then, each processor removes those cells that it is not responsible for and continues to refine the remaining cells with the previously described method up to a level l_β . It thereby keeps track of the cell neighborhood information across processor domains by introducing halo cells that are a copy of domain boundary cells on a neighboring processor, i.e., the so-called window cells. This results in a uniformly refined hierarchical and unstructured Cartesian mesh. Finally, the set and the hierarchy information is efficiently stored in parallel on disk using parallel NetCDF [44].

The proposed grid generation process is chosen for multiple reasons. It is part of the m-AIA simulation framework and capable of automatically and efficiently generating highly resolved meshes in parallel. An automated parallel grid generation process is a must for a pipeline that might be applied in clinical environments. In contrast to structured, body-fitted grids or unstructured tetrahedral meshes, tedious manual work is avoided while involving potentially skewed mesh elements. The orthogonal mesh lines allow for an application of high-order discretization methods with large stencils. The hierarchical nature of the mesh furthermore enables a straightforward dynamic refinement implementation and an efficient parallelization.

3.2.2 Step (v) of the automated pipeline: simulation method

To simulate the flow in the nasal cavity, the LB module of m-AIA is chosen for multiple reasons. The computations can be performed efficiently in parallel, it is straightforward to parallelize the code, and boundary conditions can

be easily applied in contrast to, e.g., cut-cell methods. Furthermore, there is no need to solve a pressure Poisson equation for quasi-incompressible flow as the pressure is an explicit result of the lattice-BGK algorithm [13].

The LB module solves the discretized form of the Boltzmann equation with the Bhatnagar-Gross-Krook approximation of the right-hand side collision process [45, 46], i.e.,

$$f_i(\mathbf{x} + \xi_i \Delta t, t + \Delta t) = f_i(\mathbf{x}, t) - \omega_f (f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)) \quad (5)$$

is solved for the particle probability distribution functions f_i at neighboring fluid cells at locations $\mathbf{x} + \xi_i \Delta t$. They are functions of the location vector $\mathbf{x} = (x_1, x_2, x_3)^T$, the molecular velocity vector $\xi_i = (\xi_1, \xi_2, \xi_3)^T$, and the time and time increment t and Δt . The subscript i is determined by the discretization model. In this study, the D3Q27 model [47] with $i \in \{1, \dots, 27\}$ in three dimensions is used, since it conserves rotational invariance in laminar flow and produces smaller errors in transitional or turbulent regimes compared to other three-dimensional lattices such as the D3Q19 model [48, 49].

The quantity

$$\omega_f = \frac{2\Delta t}{2\tau_f + \Delta t}, \quad (6)$$

with the relaxation time $\tau_f = \nu/c_s^2$, the isothermal speed of sound c_s , and the kinematic viscosity ν , determines the relaxation speed towards the thermodynamical equilibrium, given by the set of equilibrium PPDFs [46]

$$f_i^{eq}(\mathbf{x}, t) = \rho t_p \left[1 + \frac{\mathbf{v} \cdot \xi_i}{c_s^2} + \frac{\mathbf{v} \cdot \mathbf{v}}{2c_s^2} \cdot \left(\frac{\xi_i \cdot \xi_i}{c_s^2} - \delta \right) \right]. \quad (7)$$

Here, the quantity ρ represents the density, t_p is a direction-dependent weighting factor, $\mathbf{v}$ is the fluid velocity vector, and δ is the Kronecker delta.

To simulate the temperature distribution inside the nasal cavity, a thermal extension of the LB is employed (TLB). That is, a Multi-Distribution Function (MDF) approach is used to model the temperature field [50]. The quantity transported by the second set of distribution functions is the internal energy $\epsilon = DRT/2$ multiplied with the density ρ . The parameter $D = 3$ represents the number of dimensions, R is the specific gas constant, and T is the temperature. The collision and propagation step of the internal energy density distribution function (IEDDF) g_i are formulated similar to (5) by

$$g_i(\mathbf{x} + \xi_i \Delta t, t + \Delta t) = g_i(\mathbf{x}, t) - \omega_g (g_i(\mathbf{x}, t) - g_i^{eq}(\mathbf{x}, t)) \quad (8)$$

with the thermal relaxation frequency

$$\omega_g = \frac{2\Delta t}{2\tau_g + \Delta t}. \quad (9)$$

The relaxation time of the IEDDF is $\tau_g = (\chi D)/(c_s^2(D + 2))$, with the thermal diffusivity χ [51]. Hence, the relaxation time of the IEDDF is set by prescribing the PRANDTL number $Pr = \nu/\chi$ and the viscosity ν , which is initialized by the REYNOLDS number. The equilibrium distribution of the IEDDF in three dimensions is defined as [51]

$$g_i^{eq}(\mathbf{x}, t) = \rho \epsilon t_p \left[\frac{\xi_i^2}{Dc_s^2} + \left(\frac{\xi_i^2}{Dc_s^2} - \frac{2}{D} \right) \times \frac{\mathbf{v} \cdot \xi_i}{c_s^2} + \frac{\mathbf{v} \cdot \mathbf{v}}{2c_s^2} \cdot \left(\frac{\xi_i \cdot \xi_i}{c_s^2} - \delta \right) \right]. \quad (10)$$

The macroscopic fluid properties can be obtained from the moments of the PPDFs [46]:

$$\rho = \sum_i f_i(\mathbf{x}, t) \quad (11)$$

$$\rho \mathbf{v} = \sum_i \xi_i f_i(\mathbf{x}, t) \quad (12)$$

$$\rho \mathbf{v} \mathbf{v} + p_s \delta - \boldsymbol{\sigma} = \sum_i \xi_i \xi_i f_i(\mathbf{x}, t) \quad (13)$$

$$\rho \epsilon = \sum_i g_i(\mathbf{x}, t), \quad (14)$$

where $\boldsymbol{\sigma}$ and δ are the friction stress and Kronecker delta tensors. The static pressure p_s is derived from the ideal gas law by

$$p_s = \rho RT_0 = \rho c_s^2 = \rho/3 \quad (15)$$

with the reference temperature T_0 .

The REYNOLDS number of the problem is defined by $Re = \bar{v} \cdot d_p/\nu$, where $\bar{v}$ is the area-averaged velocity magnitude given by $\bar{v} = \dot{V} \cdot 10^{-6}/A_p$ with A_p being the area of the outlet geometry at the pharynx in m^2 , and the hydraulic diameter d_p in m . The latter is defined by $d_p = 4 \cdot A_p/C_p$, where C_p is the circumference of the outlet geometry in m . The volume flux $\dot{V} = 250\text{ml/s}$ is a common mean flux for respiration at rest [12]. The kinematic viscosity is set to $\nu = 1.5111 \cdot 10^{-5} \text{ m}^2/\text{s}$ for air at an ambient temperature and atmospheric pressure of $T_\infty = 293.15\text{K}$ and $p_\infty = 1013, 25\text{hPa}$.

Boundary conditions are prescribed at the various surface segments of the three-dimensional airway model, cf. Section 3.1. At the tissue wall, a second-order-accurate interpolated bounce-back no-slip wall boundary condition [52] and an isothermal wall boundary condition [53] are prescribed. The body temperature is set to $T_{Body} = 309.15\text{K}$ [12]. A von Neumann condition for the velocity is used at the outlet and the density is prescribed by an adaptive Dirichlet condition [12]. The adaptation continuously adjusts the density such that the resulting pressure gradient drives a flow at a target REYNOLDS number that matches $\dot{V}$. At the inlets, i.e., at both nostrils, the bound-

ary condition uses the well-known equation of Saint-Venant and Wantzel [54]. It is based on the method presented in [12], where the velocity is extrapolated to the boundary cells. The density is computed iteratively using the density at the previous time step and the momentum, which is obtained by applying a von Neumann condition. Compared to an artificially enlarged inlet region, as proposed by Burgos et al. [19], placing the inlet areas at the nostrils results in a smaller mesh and therefore saves computational resources.

4 Results

In the following, the thermal extension of the LB method is validated in Section 4.1. Subsequently, the automatic data processing pipeline is applied to the three test cases $\tau_1, \dots, \tau_3$. The corresponding results are presented in Section 4.2. To test the accuracy of the automatic pipeline, the pipeline-generated surface of τ_1 is compared to a surface that is based on a semi-manual segmentation performed by a medical professional in Section 4.3.

CNN-A, CNN-B, and CNN-C are trained on a single NVIDIA A100 graphics processing unit (GPU) on the GPU partition of the JURECA-DC supercomputer [55], Forschungszentrum Jülich. The pipeline is tested on a single core of a standard compute node on the central processing unit (CPU) partition of JURECA-DC. A standard compute node is equipped with two AMD EPYC 7742 CPU, each containing 64 cores clocked at 2.25 GHz. The numerical simulations are run on 3,071 cores of the CLAIX HPC system at RWTH Aachen University. Here, a compute node is equipped with two Intel Skylake CPU, each containing 24 cores clocked at 2.1 GHz. Simulations are run in two phases. In the first phase, a simulation is run for 300,000 time steps until its solution becomes independent from the initial flow field. In the second phase, statistics of the flow field are collected for another 300,000 time steps. The time-averaged flow field of the second phase is used to estimate the performance of the pipeline generated surface.

4.1 Validation of the TLB method

The LB method has already been validated in previous studies [56–58]. To validate the thermal extension of the LB method, the temperature field of a steady channel flow between two semi-infinite plates with distance $d_c = 2h$ and origin $y = 0$ in the center of the channel is simulated for an incompressible fluid. The results are compared to those of Hudson and Bankhoff [59]. The channel walls are assumed to be isothermal at temperature T_w . The upper wall of the channel is moving with a constant velocity v_w . The

resulting Couette flow is overlaid by the velocity profile of a Poiseuille flow. The velocity at position y is then given by

$$v(y) = \frac{v_w}{2} \left(1 + y' + a \left(1 - y'^2 \right) \right), \quad (16)$$

where $y' = y/h$ is the non-dimensional wall-normal coordinate and $a = (v_w/2v_{\max})$ is the ratio of the half wall velocity and the maximum velocity of a pure Poiseuille flow $v_{\max}$. For $a = 0$, a pure Couette flow is obtained. With increasing a the portion of the Poiseuille flow becomes more dominant. The computational mesh consists of $512 \times 64 \times 64$ cells. At the inlet, the velocity profile as formulated in (16) and a constant temperature T_0 are prescribed. The pressure is set to a constant value at the outlet. At the upper and lower channel wall, the interpolated bounce-back and isothermal wall boundary conditions as described in Section 3.2.2 are used. The front and back side of the channel, i.e. the domain boundaries in the z -direction, are connected using a periodic boundary condition. The REYNOLDS number is based on h and v_w and is set to $Re = 100$. The PRANDTL number is set to $Pr = 0.72$. The results are evaluated at the non-dimensional streamwise channel positions $x' = 0.2$ and $x' = 0.4$, with

$$x' = \frac{2x}{h \cdot Re \cdot Pr}, \quad (17)$$

where x is the streamwise coordinate. Figure 8 shows the non-dimensional temperature distribution $T^\dagger = (T(y') - T_w)/(T_0 - T_w)$ along y' . For all configurations, i.e., $a = 0$, $a = 1$, and $a = 5$, the simulation results are in good agreement with the results of Hudson and Bankhoff [59].

4.2 Surfaces generated by the automated data processing pipeline

Surfaces are generated for the three CT recordings $\tau_1, \dots, \tau_3$ using the automated data processing pipeline. Planar in- and outflow areas are generated with face normals pointing in streamwise direction. The results are illustrated in Fig. 9 at two stages of the pipeline, i.e., after step (ii) as single surfaces (left), and after step (iii) with inlet and outlet areas, and the remaining surface of the nasal cavity (right). Table 1 provides information on the in-plane and out-of-plane voxel resolutions δ_{xy} and δ_z , and on the number of axial slices n for each CT recording. The CT recording of case τ_1 has with $\delta_z(\tau_1) = 0.5\text{mm}$ the highest z -resolution, features, however, only $n = 211$ slices in the z -direction. From Fig. 9a it is obvious that in τ_1 only a small part of the pharynx is covered and the lower end of the CT data nearly matches the outflow area. Different observations are made for cases τ_2 and τ_3 . Their z -resolution is with $\delta_z(\tau_2) = 0.625\text{mm}$ and $\delta_z(\tau_3) = 0.7\text{mm}$ coarser than $\delta_z(\tau_1)$. Case

Fig. 8 Temperature distribution in wall-normal direction for different velocity ratios a at $x' = 0.2$ (l.) and $x' = 0.4$ (r.) calculated by Hudson and Bankhoff [59] and simulated using the TLB method

τ_3 covers the complete pharynx, see Fig. 9c. Parts of the lower end are cut to avoid multiple outflow areas, cf. Section 3.1.3.

Table 1 lists the time needed for steps (i), (ii), and (iii). For the time measurements, iterations for cutting the pharynx are controlled with $M_{start} = 5$ and $M_{step} = 5$. For all cases, step (i) takes less than three minutes on a single core of a JURECA-DC standard compute node, which is comparable to the performance of any low end computer such as a laptop, that does not need to be equipped with a high-end GPU. That is, the automated pipeline has a clear advantage over conventional methods such as semi-manual or manual segmentations, which may take hours or even days to complete. The segmentation time for τ_3 is slightly increased due to its highest n . Step (ii) is the least time-consuming step. Again, for τ_3 the process takes with roughly 90s slightly longer than for τ_1 and τ_2 . The timings for step (iii) need to be analyzed differently. Cases τ_1 and τ_2 require only a single iteration for the detection of the outflow area and hence have, in contrast to τ_3 , where

five iterations are necessary, reduced timings. Furthermore, the large pharynx, see Fig. 9, leads to a larger number of triangles n_{tr} , compared to n_{tr} of the other cases, and therefore to longer centerlines that need to be processed. In all cases, inflow areas are detected within a single iteration. Considering that manually defining inflow and outflow areas may take up to an hour, again a significant shortening of the processing time is achieved, even for τ_3 .

4.3 Comparison of a pipeline-based surface to a manually generated surface

The pipeline-generated surface of case τ_1 is compared to a surface that has been generated with the help of a medical professional. The former is denoted as τ_1^A , and the latter is named τ_1^M . Spatial deviations between the two surfaces are investigated in Section 4.3.1. The simulation results of the flow through the two nasal cavities are analyzed in Section 4.3.2. The error analysis provides details on the performance of the processing pipeline.

Table 1 The time needed for steps (i) - (iii), the resolution of the CT recordings (δ_{xy}, δ_z), the number of slices in the z -direction (n), and the number of triangles (n_{tr}) for cases $\tau_1, \dots, \tau_3$

CT	step (i) [mm:ss]	step (ii) [mm:ss]	step (iii) [mm:ss]	δ_{xy} [mm]	δ_z [mm]	n	n_{tr}
τ_1	01:45	00:47	04:57	0.338	0.5	211	568,007
τ_2	01:41	00:53	07:48	0.385	0.625	209	773,060
τ_3	02:46	01:23	23:03	0.352	0.7	248	982,678

Fig. 9 Surfaces of test cases $\tau_1, \dots, \tau_3$ after step (ii) (left) and after step (iii) (right) of the pipeline

4.3.1 Deviations between the surfaces

Deviations between surfaces of τ_1^A and τ_1^M are analyzed by the Hausdorff distance [60] given by

$$HD(\Theta, \Lambda) = \sup_{\theta \in \Theta} \inf_{\lambda \in \Lambda} \text{dist}(\theta, \lambda). \quad (18)$$

That is, the HD measures for all points in a set of base points Θ the distances to the closest point in a set of target points Λ . Here, Θ is represented by the vertices $\theta \in \Theta$ of the triangles of τ_1^M and Λ by the vertices $\lambda \in \Lambda$ of the triangles in τ_A . In Fig. 10b–f, the vertices of τ_1^M are colored by their HD values. These qualitative results are complemented by a histogram of all distances in Fig. 10a. Obviously, the majority of the HD values is below $3 \cdot 10^{-4} \text{m}$. The highest deviations are found near the inflow areas, see Fig. 10b, c, and f. They are caused by the different inflow geometries obtained from the CNNs and the semi-automatic segmentation. Since this is a rather subjective choice, a high HD value does not necessarily indicate a false surface. The only constraints the segmentation has to fulfill are that the inflow areas are located inside the nasal cavity downstream of the nostrils and the face normals are parallel to the streamwise direction. At the outflow region, see Fig. 10e, a similar observation as for the inflow regions can be made, i.e., increased HD values appear to the choice of the outflow position. Furthermore, increased distances, but still well below 1mm, appear at the bottom of the lower turbinates. They are caused by smoothing effects of the WSF, see Fig 10c. The pipeline-generated surface reproduces the nostrils, turbinates, maxillary sinuses, and sphenoidal sinuses well, see Fig. 10b, d, and f. Figure 10d reveals some minor weaknesses at partitions that separate cavities near the ethmoidal sinuses. As it is shown in the next Section 4.3.2, this does not affect the simulation results significantly. Except for the outflow area, distances at the pharynx are below $3 \cdot 10^{-4} \text{m}$, see Fig. 10b and e.

4.3.2 Deviations between results of numerical simulations

Flow simulations are conducted for τ_1^A and τ_1^M . A grid spacing of $\Delta_g = 0.07 \text{mm}$ is chosen to accurately resolve thin channels and boundary layers [12, 23]. The number of cells $\mathcal{N}$, the hydraulic diameter at the pharynx d_p , the pharynx area A_p and the corresponding circumference C_p , and the REYNOLDS number Re , cf. Section 3.2.2, are given in Table 2.

Deviations between the simulation results are analyzed based on total pressure p and temperature T distributions. The former is expressed as the sum of the static pressure p_s that has been introduced in Section 3, and the dynamic pressure $p_d = \rho v^2/2$, where v is the velocity magnitude. The pressure distribution is investigated with the area-averaged total pressure loss

$$\Delta \hat{p}(s) = \hat{p}_{in} - \hat{p}(s), \quad (19)$$

where $\hat{p}_{in}$ is the area-averaged pressure at the corresponding left or right inlet, and $\hat{p}(s)$ is the area-averaged total pressure at a downstream cross section of a location s along the left or right centerline, which is normalized by the

Fig. 10 Hausdorff distance HD between the pipeline-generated surface τ_1^A and the surface generated by a medical professional τ_1^M . The HD value of each vertice of τ_1^M to the closest vertices of τ_1^A is shown in form of a histogram and heat maps on the surface of τ_1^M

centerline arc length. For both simulations, the results are analyzed along the centerlines of τ_1^M to avoid differences caused by deviant centerline coordinates. The face normal of a cross section is the vector between two consecutive points on a centerline. At each point of the centerline a plane, in which the flow properties are averaged, is spanned using the face normal and the centerline point. To exclude cells located in the opposite cavity or the surrounding sinuses from the averaging, a region growing algorithm is

employed. That is, from all cells located in the plane, only cells directly connected to the seed point, i.e. the centerline point, are used for the averaging. Thus, a corruption of the averaging result by the opposite cavity and the surrounding sinuses is avoided. Figure 11 shows the pressure loss of cases τ_1^A and τ_1^M mapped onto the centerlines. Comparing the surfaces τ_1^A and τ_1^M reveals that the nostril region of τ_1^M is slightly extended. As explained in Section 4.3.1, this is due to the different choice of the inflow shape made by the

Table 2 The number of cells $\mathcal{N}$, the hydraulic diameter d_p , the surface area A_p , the circumference C_p , and the REYNOLDS number Re at the outlet for cases τ_1^A and τ_1^M

Case	$\mathcal{N}[\cdot 10^6]$	$d_p[\text{mm}]$	$A_p[\text{mm}^2]$	$C_p[\text{mm}]$	Re
τ_1^A	260.0	7.27	69.44	38.16	1607.69
τ_1^M	230.0	5.97	52.99	35.52	1727.39

medical professional and the CNNs. To ensure a consistent comparison of the two cases, centerline points outside the geometry of τ_1^A are excluded from the analysis and for τ_1^M the pressure loss occurring in the airway extension is subtracted.

Considering Fig. 11, the pressure loss distribution shows no visible differences between τ_1^A and τ_1^M , except for some minor deviations at the inflow regions. They are caused by the evaluation of (19) at incomplete cross sections at the initial set points of τ_1^A . Those points are additionally excluded from the further analysis. The quantitative results

of the area-averaged total pressure loss along the left and right centerlines, provided in Fig. 11a, show only small differences in the pressure distributions. The location along the centerline is normalized by the centerline arc length. The absolute averaged difference $\Delta \tilde{p}$ between the two cases along the left and right centerlines are only as small as $\Delta \tilde{p}_{left} = 0.016\text{Pa}$ and $\Delta \tilde{p}_{right} = 0.013\text{Pa}$. This yields a combined averaged error of 0.0145Pa , which is less than 1% of the mean pressure loss $\Delta \hat{p}_{mean}$ between the inlets and the outlet of both cases, see Fig. 11b. Figure 11b furthermore indicates that for the flow through the pipeline-generated

Fig. 11 Area-averaged total pressure loss $\Delta \hat{p}$

surface $\Delta \hat{p}_{left}$ and $\Delta \hat{p}_{right}$ are underpredicted by only 0.1% and 0.3%, i.e., for $\Delta \hat{p}_{mean}$ it is 0.2%.

Similar to the pressure distribution, the temperature distribution is computed as the area-averaged temperature $\hat{T}(s)$ at cross sections along locations s on the centerlines. As for the pressure distribution, the temperature distribution only shows visible differences at the inflow regions, see Fig. 12a and b. Again, for a consistent comparison, points on the centerline outside of τ_1^A are excluded from the quantitative analysis. Figure 12c shows the temperature difference along the centerlines on the left and right side. Obviously, the temperature distributions in the left channel agree well. The temperature curves in the right channels slightly deviate at normalized locations between 0.4 and 0.6. These deviations are caused by narrowed channels near the ethmoidal sinuses in τ_1^A . However, the impact is with an averaged absolute difference $\Delta \hat{T}$ along the left and right centerlines of $\Delta \hat{T}_{left} = 0.21K$ and $\Delta \hat{T}_{right} = 0.43K$, negligible. The combined averaged error of $\Delta \hat{T}_{mean} =$

0.32K is less than 1% of the temperatures reached at the outlets of each case, see Fig. 12d.

It should be noted that the illustration of flow quantities along the centerlines as part of step (vi) of the pipeline is performed automatically to fulfill the usability requirement mentioned in Section 1.

Figure 13 depicts axial cross-sections colored by the the pressure loss Δp and the temperature T in the reference and in the pipeline generated geometries τ_1^M and τ_1^A . The location of the cross-sections is highlighted in the same figure. Obviously, the flow fields of τ_1^A and τ_1^M share the same overall flow structures. With a growing distance from the inlets, the pressure loss that needs to be overcome by the diaphragm increases, see Fig. 13a. Similarly, the temperature increases in the streamwise direction as heat is continuously transferred from the walls to the fluid, see Fig. 13a. A major temperature rise is observed in the maxillary sinuses, where the fluid is almost at rest and surrounded by walls.

Fig. 12 Area-averaged temperature $\hat{T}$

Fig. 13 Pressure loss and temperature distribution of the flow field at the cross-section represented by the dashed red line

5 Summary, conclusions, and outlook

Conducting a large number of flow simulations for domains that are extracted from CT data often fails because of the time needed for manual data pre-processing. The automated data processing pipeline developed in this manuscript along a respiratory flow use case substitutes any kind of manual interaction and thus, massively reduces the data processing time. It should be noted that in principle the pipeline can be trained to adapt to any CT-based simulation case. The study focuses on the ML-based surface generation process, hence, the analysis of nasal cavity flows only plays a minor role. The pipeline filters CT data to better distinguish between voxels representing the airway and other matter. The segmentation is then performed with the help of CNNs, before the surface is generated and smoothed. The inflow areas are detected and flattened with the help of another CNN. The outflow area is defined along an averaged centerline that leads through the pharynx.

For three CT recordings, surfaces that can be used for conducting flow simulations are successfully generated. To evaluate the accuracy of the automated pipeline, the corresponding surface of an exemplary case is compared to a surface obtained by a semi-automatic procedure performed by a medical professional. HD values between the two surfaces are used to show the geometric differences. The pipeline is capable of generating an accurate surface, except for small difference at interfaces between the small ethmoidal sinuses, which, however, do not significantly affect simulation results. A TLB method, which is validated by a 2D flow in a channel, is employed for the simulations. They are conducted for both cases and show that the difference between the area-averaged total pressure at the inlets and at cross-sections along left and right centerlines are small. A marginal averaged error of around 1% of

the mean pressure loss between inlets and outlet indicates that the pipeline accurately generates surfaces from CT data. The difference between the area-averaged temperature at the inlets and at cross-sections along the centerlines are similarly small. An averaged error of less than 1% of the temperature reached at the outlet again stresses the high level of accuracy of the pipeline. A comparison to surfaces generated by other tools, e.g., the tools presented in Section 2.3, is beyond the scope of this work.

At present, integrating computationally expensive and time consuming fully resolved simulations into clinical paths, which require fast analyses, is unfeasible. To alleviate this issue and still use high-quality methods, multiple requirements need to be fulfilled. On the one hand, a fast and reliable simulation method needs to be used and on the other hand, HPC resources have to be made available to the medical community. The TLB method used in this study is especially suited for simulating complex flows in intricate geometries on HPC systems. It has shown to be highly scalable and no flow scales are modeled, i.e., it operates on highly resolved meshes that capture all relevant flow features [13]. Making such methods and computational resources available to the medical community is the next step in line.

Many HPC centers already offer services and provide computing time for customers from industry and public institutions [61–64]. Furthermore, with exascale computing just around the corner, sufficient resources will soon become available to run many simulations simultaneously in a short amount of time. Finally, ML techniques can not only help to reduce the time needed to pre-process CT data, they can also be employed to accelerate the simulations, e.g., by creating surrogates that are integrated in full loops of physics-informed deep learning workflows.

Acknowledgements The authors gratefully acknowledge the computing time granted through the Jülich Aachen Research Alliance (JARA) on the supercomputers JURECA [55] at Forschungszentrum Jülich and CLAIX at RWTH Aachen University. Furthermore, the authors would like to thank Forschungszentrum Jülich GmbH, RWTH Aachen University, and the JARA Center for Simulation and Data Science (JARA-CSD) for research funding. This work was performed as part of the Helmholtz School for Data Science in Life, Earth and Energy (HDS-LEE). Finally, the authors would like to thank Dr. med. Justus Ilgner of the Department of Otorhinolaryngology, Plastic Head and Neck Surgery, RWTH Aachen University Hospital for reviewing this manuscript.

Funding This work is funded by the Forschungszentrum Jülich GmbH, RWTH Aachen University, and the JARA Center for Simulation and Data Science (JARA-CSD).

Declarations

Conflicts of interest/Competing interests On behalf of all authors, the corresponding author states that there is no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Rüttgers M, Park J, You D (2019) Large-eddy simulation of turbulent flow over the driver fastback vehicle model. *J Wind Eng Ind Aerodyn* 186:123–138. <https://doi.org/10.1016/j.jweia.2019.01.003>
- Schlimpert S, Koh SR, Pausch K, Meinke M, Schröder W (2017) Analysis of combustion noise of a turbulent premixed slot jet flame. *Combustion and Flame* 175(C):292–306. <https://doi.org/10.1016/j.combustflame.2016.08.001>
- Yang X, Kuru E, Gingras M, Iremonger S (2019) Ct-cfd integrated investigation into porosity and permeability of neat early-age well cement at downhole condition. *Constr Build Mater* 205:73–86. <https://doi.org/10.1016/j.conbuildmat.2019.02.004>
- Lintermann A, Schröder W (2019) A hierarchical numerical journey through the nasal cavity: from nose-like models to real anatomies. *Flow Turb Combust* 102(1):89–116. <https://doi.org/10.1007/s10494-017-9876-0>
- Toyserkani N, Frisch T (2012) Are too many septal deviations operated on? A retrospective patient's satisfaction questionnaire with 11 years follow-up. *Rhinology* 50:185–190. <https://doi.org/10.4193/Rhino11.218>
- Vanhille D, Garcia GJM, Asan O (2018) Virtual surgery for the nasal airway: A preliminary report on decision support and technology acceptance. *J Amer Med Assoc Facial Plastic Surg* 20:63–69. <https://doi.org/10.1001/jamafacial.2017.1554>
- Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR), pp 3431–3440. <https://doi.org/10.1109/TPAMI.2016.2572683>
- Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells W, Frangi A (eds) Medical image computing and computer-assisted intervention – MICCAI 2015. MICCAI 2015. Lecture notes in computer science, vol 9351. Springer, Cham. https://doi.org/10.1007/978-3-319-24574-4_28
- Mao X, Shen C, Yang Y-B (2016) Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In: Lee D, Sugiyama M, Luxburg U, Guyon I, Garnett R (eds) Advances in neural information processing systems, vol 29. Curran Associates, Inc.
- Zhou Z, Rahman Siddiquee MM, Tajbakhsh N, Liang J (2019) Unet++: redesigning skip connections to exploit multiscale features in image segmentation. *IEEE Transactions on Medical Imaging*. <https://doi.org/10.1109/TMI.2019.2959609>
- Lu H, Wang H, Zhang Q, Yoon SW, Won D (2019) A 3d convolutional neural network for volumetric image semantic segmentation. *Procedia Manuf* 39:422–428. <https://doi.org/10.1016/j.promfg.2020.01.386>, <https://www.sciencedirect.com/science/article/pii/S2351978920304571>, 25th international conference on production research manufacturing innovation: cyber physical manufacturing August 9–14, 2019 — Chicago, Illinois (USA)
- Lintermann A, Meinke M, Schröder W (2013) Fluid mechanics based classification of the respiratory efficiency of several nasal cavities. *Comput Biol Med* 43(11):1833–1852. <https://doi.org/10.1016/j.combiomed.2013.09.003>
- Lintermann A, Meinke M, Schröder W (2020) Zonal Flow Solver (ZFS): a highly efficient multi-physics simulation framework. *Int J Comput Fluid Dyn*:1–28. <https://doi.org/10.1080/10618562.2020.1742328>
- Waldmann M, Lintermann A, Choi YJ, Schröder W (2020) Analysis of the effects of MARME treatment on respiratory flow Using the Lattice-Boltzmann Method. *Results Numer Exper Fluid Mech* XII:853–863. https://doi.org/10.1007/978-3-030-25253-3_80
- Zhang Y, Zhou X, Lou M, Gong M, Zhang J, Ma R, Zhang L, Huang F, Sun B, Zhu K, Tong Z, Zheng G (2019) Computational fluid dynamics (cfd) investigation of aerodynamic characters inside nasal cavity towards surgical treatments for secondary atrophic rhinitis. *Math Probl Eng* 2019:1–8. <https://doi.org/10.1155/2019/6240320>
- Calmet H, Gambaruto A, Bates A, Vázquez M, Houzeaux G, Doorly D (2015) Large-scale cfd simulations of the transitional and turbulent regime for the large human airways during rapid inhalation. *Comput Biol Med* 69. <https://doi.org/10.1016/j.combiomed.2015.12.003>
- Smagorinsky J (1963) General circulation experiments with the primitive equations. *Mon Weather Rev* 91(3):99–164. [https://doi.org/10.1175/1520-0493\(1963\)091<0099:GCEWTP>2.3.CO;2](https://doi.org/10.1175/1520-0493(1963)091<0099:GCEWTP>2.3.CO;2)
- Germano M, Piomelli U, Moin P, Cabot WH (1991) A dynamic subgrid-scale eddy viscosity model. *Phys Fluids A: Fluid Dyn* 3(7):1760–1765. <https://doi.org/10.1063/1.857955>
- Burgos Olmos M, Rojas E, Singh N, Esteban-Ortega F (2018) Digbody ® : a new 3d modeling tool for nasal virtual surgery. *Comput Biol Med* 98. <https://doi.org/10.1016/j.combiomed.2018.05.016>
- Burgos MA, Sanmiguel-Rojas E, del Pino C, Sevilla-García MA, Esteban-Ortega F (2017) New CFD tools to evaluate nasal airflow. *Eur Arch Otorhinolaryngol* 274(8):3121–3128. <https://doi.org/10.1007/s00405-017-4611-y>
- Huang R, Nedanoski A, Fletcher DF, Singh N, Schmid J, Young PM, Stow N, Bi L, Traini D, Wong E, Phillips CL, Grunstein

- RR, Kim J (2019) An automated segmentation framework for nasal computational fluid dynamics analysis in computed tomography. *Comput Biol Med* 115:103505. <https://doi.org/10.1016/j.combiomed.2019.103505>
22. Chien K-Y (1982) Predictions of channel and boundary-layer flows with a low-Reynolds-number turbulence model. *Amer Inst Aeron Astron J* 20(1):33–38. <https://doi.org/10.2514/3.51043>
23. Eitel G, Freitas RK, Lintermann A, Meinke M, Schröder W (2010) Numerical simulation of nasal cavity flow based on a lattice-Boltzmann method. In: Dillmann A, Heller G, Klaas M, Kreplin H-P, Nitsche W, Schröder W (eds) *Results in numerical and experimental fluid mechanics VII. Notes on numerical fluid mechanics and multidisciplinary design*, vol 112. Springer, Berlin, pp 513–520. https://doi.org/10.1007/978-3-642-14243-7_63
24. Lorensen WE, Cline HE (1987) Marching cubes: a high resolution 3d surface construction algorithm. *SIGGRAPH Comput Graph* 21(4):163–169. <https://doi.org/10.1145/37402.37422>
25. Inthavong K, Wen J, Tu J, Tian Z (2009) From ct scans to cfd modelling - fluid and heat transfer in a realistic human nasal cavity. *Eng Appl Comput Fluid Mech* 3(3):321–335. <https://doi.org/10.1080/19942060.2009.11015274>
26. Koch W, Lehner M, Benda O, Ortiz R, Koch G (2021) Clinical cfd applications 2. In: *Clinical and biomedical engineering in the human nose*. Springer, pp 242–249. https://doi.org/10.1007/978-981-15-6716-2_10
27. Wilcox DC (1998) *Turbulence Modeling for CFD*, 2nd edn. DCW Industries, La Canada
28. Grotjans H, Menter FR (1998) Wall functions for industrial applications. In: Papailiou KD (ed) *Computational fluid dynamics '98, ECCOMAS*. Wiley, pp 1112–1117
29. Hörschler I, Schröder W, Meinke M (2010) On the assumption of steadiness of nasal cavity flow. *J Biomech* 43(6):1081–5. <https://doi.org/10.1016/j.jbiomech.2009.12.008>
30. Li C, Jiang J, Dong H, Zhao K (2017) Computational modeling and validation of human nasal airflow under various breathing conditions. *J Biomech* 64:59–68. <https://doi.org/10.1016/j.jbiomech.2017.08.031>
31. Lintermann A, Schlimpert S, Grimm JH, Günther C, Meinke M, Schröder W (2014) Massively parallel grid generation on HPC systems. *Comput Methods Appl Mech Eng* 277:131–153. <https://doi.org/10.1016/j.cma.2014.04.009>
32. Kabuliak N, Nejati A, Loch C, Schwass D, Cater J, Jermy M (2017) Strategies for segmenting the upper airway in cone-beam computed tomography (CBCT) Data. *Open J Med Imaging* 07(4):196–219. <https://doi.org/10.4236/ojmi.2017.74019>
33. Perona P, Malik J (1990) Scale-space and edge detection using anisotropic diffusion. *IEEE Trans Pattern Anal Mach Intell* 12(7):629–639. <https://doi.org/10.1109/34.56205>
34. Kapur T, Pieper S, Fedorov A, Fillion-Robin J-C, Halle M, O'Donnell L, Lasso A, Ungi T, Pinter C, Finet J, Pujol S, Jagadeesan J, Tokuda J, Norton I, Estepar RSJ, Gering D, Aerts HJWL, Jakab M, Hata N, Ibanez L, Blezek D, Miller J, Aylward S, Grimson WEL, Fichtinger G, Wells WM, Lorensen WE, Schroeder W, Kikinis R (2016) Increasing the impact of medical image computing using community-based open-access hackathons: the na-mic and 3d slicer experience. *Med Image Anal* 33:176–180. <https://doi.org/10.1016/j.media.2016.06.035>, 20th anniversary of the Medical Image Analysis journal (MedIA)
35. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Bach F, Blei D (eds) *Proceedings of the 32nd international conference on machine Learning. Proceedings of Machine Learning Research*, vol 37. PMLR, Lille, pp 448–456
36. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: A simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15(56):1929–1958
37. He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: *Proceedings of the 2015 IEEE international conference on computer vision (ICCV)*. ICCV '15. IEEE Computer Society, USA, pp 1026–1034
38. Kingma DP, Ba J (2015) Adam: A method for stochastic optimization. *CoRR*
39. Taubin G, Zhang T, Golub G (1996) Optimal surface smoothing as filter design. <https://doi.org/10.1007/BFb0015544>
40. Antiga L, Piccinelli M, Botti L, Ene-Iordache B, Remuzzi A, Steinman D (2008) An image-based modeling framework for patient-specific computational hemodynamics. *Med Biol Eng Comput* 46:1097–1112. <https://doi.org/10.1007/s11517-008-0420-1>
41. Attene M (2010) A lightweight approach to repairing digitized polygon meshes. *Vis Comput* 26:1393–1406. <https://doi.org/10.1007/s00371-010-0416-3>
42. Bonet J, Peraire J (1991) An alternating digital tree (ADT) algorithm for 3D geometric searching and intersection problems. *Int J Numer Methods Eng* 31(1):1–17. <https://doi.org/10.1002/nme.1620310102>
43. Sagan H (1994) *Space-filling curves*, 1st edn. Universitext, Springer, New York
44. Li J, Zingale M, Liao W, Choudhary A, Ross R, Thakur R, Gropp W, Latham R, Siegel A, Gallagher B (2003) Parallel netCDF: a high-performance scientific I/O interface. In: *Proceedings of the 2003 ACM/IEEE conference on Supercomputing - SC '03*. ACM Press, New York, p 39. <https://doi.org/10.1145/1048935.1050189>
45. Bhatnagar PL, Gross EP, Krook M (1954) A Model for Collision Processes in Gases. I. Small amplitude processes in charged and neutral one-component systems. *Phys Rev* 94(3):511–525. <https://doi.org/10.1103/PhysRev.94.511>
46. Benzi R, Succi S, Vergassola M (1992) The lattice Boltzmann equation: theory and applications. *Phys Rep* 222(3):145–197. [https://doi.org/10.1016/0370-1573\(92\)90090-M](https://doi.org/10.1016/0370-1573(92)90090-M)
47. Qian YH, D'Humières D, Lallemand P (1992) Lattice BGK Models for Navier-Stokes Equation. *Europhysics Letters (EPL)* 17(6):479–484. <https://doi.org/10.1209/0295-5075/17/6/001>
48. Kuwata Y, Suga K (2015) Anomaly of the lattice boltzmann methods in three-dimensional cylindrical flows. *J Comput Phys* 280:563–569. <https://doi.org/10.1016/j.jcp.2014.10.002>
49. White AT, Chong CK (2011) Rotational invariance in the three-dimensional lattice boltzmann method is dependent on the choice of lattice. *J Comput Phys* 230(16):6367–6378. <https://doi.org/10.1016/j.jcp.2011.04.031>
50. Guo Z, Shi B, Zheng C (2002) A coupled lattice bgk model for the boussinesq equations. *Int J Numer Methods Fluids* 39(4):325–342. <https://doi.org/10.1002/flid.337>
51. He X, Chen S, Doolen GD (1998) A novel thermal model for the lattice boltzmann method in incompressible limit. *J Comput Phys* 146(1):282–300. <https://doi.org/10.1006/jcph.1998.6057>
52. Bouzidi M, Firdaouss M, Lallemand P (2001) Momentum transfer of a Boltzmann-lattice fluid with boundaries. *Phys Fluids* 13(11):3452–3459. <https://doi.org/10.1063/1.1399290>
53. Li L, Mei R, Klausner JF (2013) Boundary conditions for thermal lattice boltzmann equation method. *J Comput Phys* 237:366–395. <https://doi.org/10.1016/j.jcp.2012.11.027>
54. Saint-Venant B, Wantzel L (1839) Mémoire et expérience sur l'écoulement déterminé par des différences de pressions considérables. *J L'École Polytech H.27:85ff*

55. Krause D, Thörnig P (2018) JURECA: Modular supercomputer at Jülich Supercomputing Centre. JLSRF 4:A132. <https://doi.org/10.17815/jlsrf-4-121-1>
56. Freitas RK, Henze A, Meinke M, Schröder W (2011) Analysis of Lattice-Boltzmann methods for internal flows. *Comput Fluids* 47(1):115–121. <https://doi.org/10.1016/j.compfluid.2011.02.019>
57. Freitas R, Meinke M, Schröder W (2013) A lattice-boltzmann method with hierarchically refined meshes. *Comput Fluids* 75:127–139. <https://doi.org/10.1016/j.compfluid.2013.01.013>
58. Lintermann A, Meinke M, Schröder W (2011) Investigations of the inspiration and heating capability of the human nasal cavity based on a Lattice-Boltzmann method. In: *Proceedings of the ECCOMAS thematic international conference on simulation and modeling of biological flows (SIMBIO 2011)*, Brussels
59. Hudson JL, Bankoff SG (1965) Heat transfer to a steady couette flow with pressure gradient. *Chem Eng Sci* 20(5):415–423. [https://doi.org/10.1016/0009-2509\(65\)80054-9](https://doi.org/10.1016/0009-2509(65)80054-9)
60. Huttenlocher DP, Klanderman GA, Rucklidge WJ (1993) Comparing images using the hausdorff distance. *IEEE Trans Pattern Anal Mach Intell* 15(9):850–863. <https://doi.org/10.1109/34.232073>
61. Jülich Supercomputing Centre, Forschungszentrum Jülich GmbH, Industry Relations [cited 15.03.2021]. <http://www.fz-juelich.de/ias/jsc/industry-relations>
62. High Performance Computing Center Stuttgart, Enterprises & SMEs [cited 15.03.2021]. <https://www.hlr.de/solutions-services/enterprises-sme>
63. Board of Trustees of the University of Illinois, Crafting solutions for grand challenges in industry and government [cited 15.03.2021]. <http://www.ncsa.illinois.edu/industry>
64. RIKEN, Collaboration with Industry [cited 15.03.2021]. <https://www.riken.jp/en/collab>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Mario Rüttgers received the B.S. degree in mechanical engineering from RWTH Aachen University in 2015. He attended full-time Korean language courses at Chonnam National University (CNU) in Gwangju, South Korea, from 2015 to 2017 and received the M.S. degree in mechanical engineering from Pohang University of Science and Technology (POSTECH) in 2019. Since 2019, he has been working as a researcher at the Chair of Fluid Mechanics and

Institute of Aerodynamics of RWTH Aachen University, and at the Simulation and Data Laboratory “Highly Scalable Fluid & Solids Engineering” (SDL FSE) of the Jülich Supercomputing Centre (JSC). At the same he is a PhD student at RWTH Aachen University and associated with the Helmholtz School for Data Science in Life, Earth and Energy (HDS-LEE). His research focuses on combining computational fluid dynamics and machine learning techniques to improve diagnostics in rhinology.

Mr. Rüttgers was awarded with a scholarship from the Chinese Government Scholarship Program (CGSP) which enabled him to study at Huazhong University of Science and Technology (HUST) in Wuhan, China, from 2013 to 2014. He has been granted a scholarship by the Korean Government Scholarship Program (KGSP) from 2015 to 2019.

Moritz Waldmann is a Ph.D. student at the Chair of Fluid Mechanics and Institute of Aerodynamics of RWTH Aachen University since 2017. His research focuses on the fluid mechanical analysis of respiration using numerical methods as the lattice-Boltzmann method and the finite cell method.

Wolfgang Schröder holds the Chair of Fluid Mechanics and heads the Institute of Aerodynamics of RWTH Aachen University. He is the Editor-in-Chief of the *European Journal of Mechanics / B Fluids* and the series *Notes on Numerical Fluid Mechanics and Multi-disciplinary Design*. His main research is in the fields of computational fluid dynamics and aeroacoustics, measurement techniques, turbulence, single and multiphase flows, and biomedical flows.

Dr.-Ing. Andreas Lintermann is a postdoctoral researcher and a group leader at the Jülich Supercomputing Centre (JSC) of Forschungszentrum Jülich, Germany. He is coordinating the European Center of Excellence in Exascale Computing “Research on AI- and Simulation-Based Engineering at Exascale” (CoE RAISE) and is heading the Simulation and Data Laboratory “Highly Scalable Fluid & Solids Engineering” (SDL FSE) at JSC.

Lintermann furthermore leads the activities in the EuroCC project from Jülich's side and is involved in the Industry Relations Team of the institute. His research focuses on high-performance computing, heterogeneous hardware, modular supercomputing, artificial intelligence, bio-fluidmechanical analyses of respiratory diseases, lattice-Boltzmann methods, high-scaling meshing methods, and efficient multi-physics coupling strategies.