

Numerical Simulation of Particle Deposition in the Human Lungs

Zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften

von der Fakultät für Mathematik des Karlsruher Instituts für
Technologie (KIT) genehmigte

Dissertation

von

Dipl.-Math. techn. Thomas Gengenbach
aus Pforzheim

Tag der mündlichen Prüfung: 27. Juni 2012

Referent: Prof. Dr. Vincent Heuveline

Korreferent: Prof. Dr. Marcel Filoche

Korreferent: Jun.-Prof. Dr. Jan-Philipp Weiß

Smile, breathe and go slowly.

– Thích Nhất Hạnh

Abstract

In this thesis, we model and simulate breathing and calculate particle depositions in the human lungs. We review the theory and discretization of fluid mechanics and the anatomy, physiology and measuring methods of the human lungs. We introduce a new model among others that accounts for the non-resolvable part of the human lungs and investigate it with the help of sensitivity analysis using the singular value decomposition. Particles, e.g. fine dust or medical aerosols, and their depositions are simulated in patient-specific and especially created schematized human lung geometries. We compare the results with respect to three main geometrical parameters and set them into relationship with the particle deposition in a multiplicative model of subsequent bifurcations in the lungs.

Zusammenfassung

In dieser Arbeit wird die menschliche Atmung modelliert und simuliert und Partikelablagerungen in der Lunge werden bestimmt. Sowohl die Theorie und Diskretisierung in der Strömungsmechanik, als auch die Anatomie, Physiologie und verschiedene Messverfahren für die Lunge werden behandelt. Unter anderen, wird ein neues Modell eingeführt, das den nicht auflösbaren Bereich der Lunge darstellt. Dieses Modell wird mit Hilfe der Sensitivitätsanalyse durch eine Singulärwertzerlegung näher untersucht. Partikel, wie Feinstaub oder medizinische Sprays, und ihre Verteilung werden in patientenspezifischen und schematisierten Geometrien der Lunge simuliert. Die Ergebnisse, anhand dreier geometrischer Werte parametrisiert, werden verglichen und in Zusammenhang zu einem multiplikativen Modell aufeinander folgender Verzweigungen in der Lunge gesetzt.

Acknowledgments

I am especially grateful to Prof. Dr. Vincent Heuveline who guided me already through seminars, my diploma thesis and now this work. I want to thank him for his positive encouragement, his support and helpful discussions. Also I am very grateful to my second advisor Prof. Dr. Marcel Filoche for his enduring support, Prof. Dr. Bernard Sapoval and the Karlsruhe House of Young Scientists who made it possible for me to work and live for six months in Paris. I want to thank Jun.-Prof. Dr. Jan-Phillip Weiß for his corrections and last-minute support and PD Dr. Gudrun Thäter for her encouragement and her sympathetic ears. Finally, many many thanks to everybody who supported me, my family and the working group at EMCL for the perfect working atmosphere, corrections, useful hints and all the coffee breaks.

Contents

1	Introduction	1
2	Fluid Flow Simulations	5
2.1	Modeling Biomedical Fluid Flows	5
2.2	Discretization and Implementation	10
3	The Human Lungs	21
3.1	Anatomy	22
3.2	Function	23
3.3	Major Lung Diseases	26
3.4	Measurement Methods	27
4	Models of the Human Lungs	31
4.1	Viscoelastic Model with Non-local Damping	32
4.2	Two-scale Models	32
4.3	Numerical Experiments	37
5	Sensitivity Analysis	43
5.1	Theoretical Background	44
5.2	Singular Value Decomposition	47
5.3	Eigenvalue and Eigenvector Solvers	48
5.4	Numerical Methods	51
6	Particle Deposition in the Human Lungs	63
6.1	Particle Flow	64
6.2	Patient-Specific vs. Schematic Bronchial Tree	68
6.3	A Simple Bifurcation Model	84
7	Summary and Outlook	87

A	Cluster	89
A.1	Bull Cluster Taurus	89
A.2	XServe Cluster	89
B	Eigenvalue/Eigenvector Code	91
C	Code for the ODE Solvers	93
D	Parameter Extraction Tables	95

List of Figures

1.1	Sketch of the breathing mechanism.	1
1.2	The segmented lungs.	2
1.3	Particles flowing through one schematic bifurcation.	3
2.1	Enforcing continuity by identifying degrees of freedom.	19
2.2	Mapping from reference element to physical element.	19
3.1	Detailed scheme of the complete human respiratory system.	21
3.2	Schematic human respiratory system partitioned in conducting and respiratory zones.	23
3.3	Representation of lung function values.	25
3.4	CT cast of a West-European middle-aged male.	28
3.5	The rhinomanometry method.	29
3.6	Illustration of a spirometry.	29
4.1	The scheme of the coupling between the upper geometry and the lower lung model.	33
4.2	Part of the CT cast (values inverted) of a West-European middle-aged male.	37
4.3	Unevenly branched tree pipe system.	38
4.4	Inscribed z-normal slices in lung geometry.	39
4.5	Sketch to clarify the setup for testing the different boundary conditions.	40
4.6	Resulting glyphs of the boundary condition tests I.	42
4.7	Resulting glyphs of the boundary condition tests II.	42
5.1	Geometry of a schematic two-dimensional lung with four generations. Simulations were run for the symmetric (the whole geometry) and the asymmetric case (the dark gray part of the geometry without the light gray part).	56

5.2	Scheme of the boundary conditions at the resolvable part of the geometry.	57
5.3	SVD in the symmetric case.	58
5.4	SVD in the asymmetric case.	59
5.5	Scheme of filtering after the forward and after the adjoint operator.	61
5.6	Filtering in the SVD	61
6.1	Segmented patient-specific geometry.	69
6.2	Three bifurcations with the parameters h, θ and α	70
6.3	Patient-specific bronchiole tree with centerlines and inscribed triangles.	71
6.4	Basic bifurcation without transition zone.	72
6.5	View of a single asymmetric bifurcation with the transition zone.	72
6.6	Tracheobronchial trees used for particle tracking.	73
6.7	Schematic particle trajectories.	76
6.8	Scheme of a tracheobronchial tree with inscribed planes between bifurcations.	77
6.9	Structure of the lung with segmented first generations.	79
6.10	Global escape rate in terms of Stokes number in one bifurcation for different Reynolds numbers. The scale of the Stokes number is logarithmic.	80
6.11	Global escape rate for different Reynolds numbers.	80
6.12	Standard deviation and mean of global escape rate.	81
6.13	Global escape rate in one bifurcation computed with OpenLB.	82
6.14	Global escape rate in schematic and the patient-specific 4-generation tree.	83
6.15	Scheme of the levels of bifurcations in a 4-generation tree.	83
6.16	Escape rate ratio per generation.	84
6.17	Comparison of the escape rates to the analytical model.	85
6.18	Escape rates for one and two bifurcations computed with OpenLB.	85
6.19	The Weibel "A" model tree with two bifurcations.	86
6.20	Comparison between patient-specific, schematic and multiplicative tree.	86

List of Tables

3.1	Typical values of respiration frequency, flux and ventilation rate.	26
4.1	Comparison tables for the fluxes through z-normal slices in absolute numbers.	41
4.2	Comparison tables for the fluxes through z-normal slices in relative numbers.	41
D.1	Table for the computation of the radius ratio h	95
D.2	Table for the computation of the opening angle θ	96
D.3	Table for the computation of the torque α	97

Nomenclature

(\cdot, \cdot)	Inner product.
α	Torque between bifurcations in the lungs, in Chapter 6.
\mathcal{A}	Evolution operator.
\mathcal{A}^*	Adjoint operator to the evolution operator \mathcal{A} .
$\mathcal{H}^m(\Omega)$	Hilbert space of all functions in $\mathcal{L}^2(\Omega)$ whose partial derivatives to the m^{th} order are all in $\mathcal{L}^2(\Omega)$.
$\mathcal{L}^2(\Omega)$	Hilbert space of all Lebesgue measurable functions f such that $\int_{\Omega} f^2 < \infty$.
μ	Dynamic viscosity, sometimes subscripted with F for fluid or P for particle.
$\nabla \cdot u$	Divergence of a vector field u .
∇u	Gradient of a vector field u .
ν	Kinematic viscosity.
Ω	Fluid flow domain.
∂V	Boundary of a domain V , defined also as Γ , which is then usually subscripted for better distinction.
∂_x	Partial derivative in direction x .
ρ	Density, sometimes subscripted with F for fluid or P for particle.
Σ	Matrix of singular values with entries σ_i on the diagonal.
θ	Angle between the children tubes of one bifurcation, in Chapter 6.
C^m	Space of m times continuously differentiable functions.
Df	Formal derivative of operator f .

h	Ratio of diameters of subsequent tubes, in Chapter 6.
I	Identity matrix, sometimes subscripted with the dimension.
P_i	Pressure in the i^{th} generation.
Q	Volumetric flow rate.
R_i	Resistance in the i^{th} generation.
Re	Reynolds number.
Re_P	Particle Reynolds number.
St	Stokes number for particles.

Introduction

Take a deep breath! One of the first and most natural things we do, breathing, is the subject of this thesis. However, easy as breathing may be, analyzing, modeling and simulating inhalation and exhalation from a numerical point of view is non-trivial. Respiration, which includes breathing, has been studied in many scientific fields, mainly of course in medicine and biology. Respiration is divided into cellular and external respiration. Cellular respiration happens at the cellular level and describes the metabolic reactions of cells to gain energy. We do not consider this topic and investigate external respiration instead, restricting ourselves to breathing, cf. Figure 1.1.

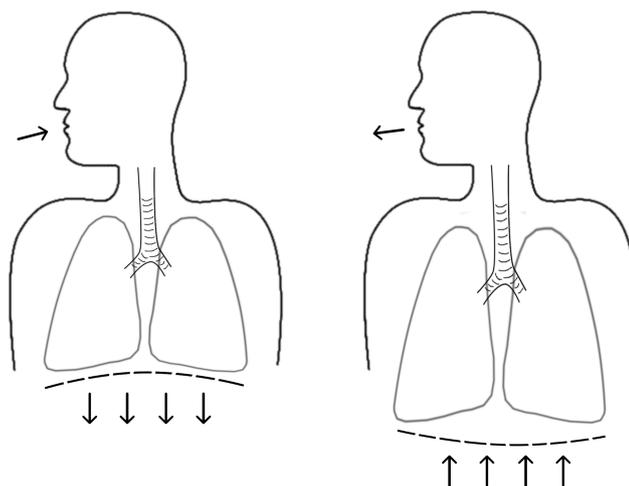


Figure 1.1: Sketch of the breathing mechanism, inspiration on the left hand side, expiration on the right hand side. Through downwards movement of the diaphragm (dashed lines) and outwards movement of the chest, the resulting low pressure in the surrounding tissue expands the lungs and inspiration occurs. Relaxation of the diaphragm and the chest walls leads to expiration. Expiration is a passive process under normal circumstances.

Breathing moves the air in and out of the lungs. The existing methods in

the field of *computational fluid dynamics* (CFD) allow to simulate the airflow in the human lungs. However, the complexity of the underlying physics in the lungs is such that open questions with respect to the modeling still need to be addressed. We provide the necessary tools to handle numerical simulations of fluid flows on high performance computers in Chapter 2 and give a brief anatomical and physiological description of the human lungs in Chapter 3.

Up to now, it is not possible to completely capture the complex structure of the lungs by medical imaging techniques. But even if we could, nowadays high performance computers would not provide enough computing power to simulate the whole human lungs in all its complexity (see Figure 1.2). We discuss an existing model and introduce two new models in Chapter 4 that are necessary to reduce the complexity and to account for effects that come from the unresolved lower lungs part. Our new models allow us to simulate the airflow in the upper part of the human lungs. The used models can also include the movement of the diaphragm and therefore the tissue to create a low pressure at alveolar level, see Figure 1.1 for illustration. Through this reduced pressure, fresh air is sucked in. The lungs bifurcate approximately 23 times so that a big surface emerges at the

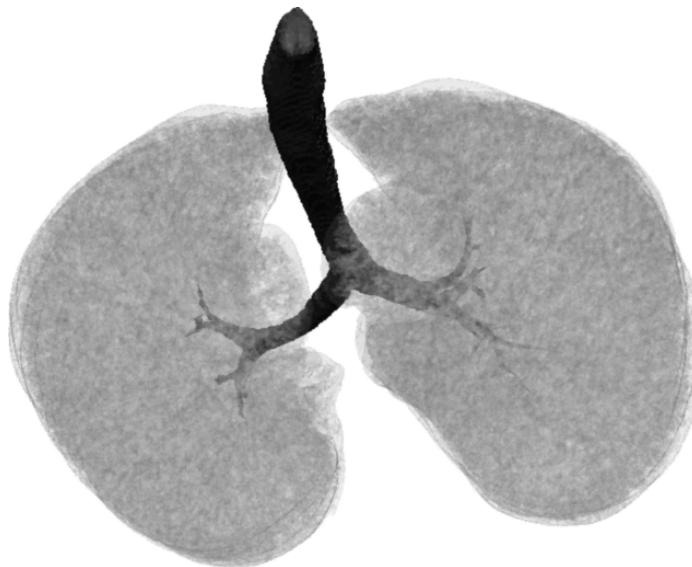


Figure 1.2: The upper part of the lungs with the trachea, the bronchi and the first bronchioles in dark color can be resolved completely. The lower part of the lungs in lighter gray color, including most of the bronchioles and the alveoles, cannot be resolved and needs to be described by models.

alveolar level. This surface, approximately the size of a soccer field, is needed for the diffusive gas exchange. The bifurcative nature of the lungs renders its

structure highly sensitive. For example a particle, once inspired from the trachea into the left side of the lungs, will never be able to get to the right side during inspiration. Sensitivity analysis, discussed in Chapter 5, analyzes the effect of small perturbations of parameters and can thus provide useful information on the sensitivity of them. We establish a framework to use sensitivity analysis directly with the fluid flow simulation to investigate the effect of models for the lower lungs part on the CFD modeled upper part. A model that is highly sensitive in the right places is able to capture for instance the influence of certain diseases.



Figure 1.3: Particles flowing through one schematic bifurcation. We show that we can plug together these building blocks, parametrized bifurcations, with respect to some patient-specific geometrical parameters and obtain significantly similar results to the full patient-specific simulation.

In the last part of this thesis, Chapter 6, we additionally model particles flowing in the human lungs, see Figure 1.3. Simulations of particles necessitate everything we addressed so far. We use a one-way coupling approach, thus we simulate the airflow first, stationary or instationary, and then inject particles. We calculate the position of the particles by following their trajectories in the fluid flow field. Knowing about particle deposition is important e.g. for pharmaceutical drug companies. They need to predict the amount of medication that crosses the blood-air barrier to judge the efficacy of drugs. As opposed to many studies about particle depositions, we model and predict particle depositions not only in patient-specific lung geometries. Instead, we also provide the framework to build a schematic model that varies and can be adjusted in

three main geometrical parameters. With our numerical particle simulations, we show a significant similarity of the particle deposition between the schematic and patient-specific simulations. Finally, we will summarize our work and give an outlook for future development of breathing and particle simulations on high performance computers.

Fluid Flow Simulations

In this chapter, we want to model the behavior of air loaded with particles flowing through the human airways. For a better understanding, we first introduce parts of the physical and mathematical background with an emphasis on fluid flows in biomedical frameworks. Then we discuss in more detail the numerical treatment of these kind of equations and the implementation in a computational fluid dynamics software.

2.1 Modeling Biomedical Fluid Flows

Airflow simulations are based on *partial differential equations (PDEs)*. These PDEs arise in turn from the principles of continuum mechanics when dealing with a fluid flowing through a domain, which are the human lungs in our case, as well as on particle mechanics when dealing with the motion of particles through the fluid. In the following, we want to introduce the basics to transfer the fluid flow into a set of equations that we can handle numerically in a fairly good way. For further details, we refer to the for us most relevant literature, e.g. William Layton, “Introduction to the numerical analysis of incompressible viscous flows”, Chapter 5 [59], Chapter 1 in “Finite Element Methods for Flow Problems” by Jean Donea and Antonio Huerta [20]. Also considered were Roger Temam and Alain Miranville, “Mathematical modeling in continuum mechanics” [84] and by George Keith Batchelor, “An Introduction to Fluid Dynamics” [5]. Of course, many more references exist that give detailed considerations of this topic.

2.1.1 Basic Conservation Equations

We assume the mechanical properties to behave like a continuum which is reasonable when we are dealing with large structures. In that case we are interested in the macroscopic behavior of the fluid. The microscopic effects enter therefore

as averages. This leads us directly to the first and very basic hypothesis that we pose with respect to space and time:

Assumption 2.1.1 (Continuum hypothesis). *The mass $M(t)$ of an arbitrary volume $V(t)$ at time t can be calculated by the integral over the density $\rho(x, t) > 0$:*

$$M(t) = \int_{V(t)} \rho(x, t) dx. \quad (2.1)$$

In the following, let $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, always denote the fluid domain that moves in space under the influence of internal and external forces. From now on, we will consider the general setup where $V(t) \subset \Omega$ is a volume always consisting of the same continuum of molecules over time. This control volume is open and bounded by a closed and sufficiently smooth control surface $\partial V(t)$. The following fundamental theorem tells us, how the rate of change of an integrated quantity is related to the time rate of change of this quantity within the control volume and the net flux with velocity $u(x, t)$ of this quantity through the control surface:

Theorem 2.1.2 (Reynolds' transport theorem). *Let $f : \Omega \times I \rightarrow \mathbb{R}$ be a differentiable function with $x \in \Omega$, $t \in I \subset \mathbb{R}$ and n the outward unit normal vector of $\partial V(t)$, then*

$$\frac{d}{dt} \int_{V(t)} f dx = \int_{V(t)} \frac{\partial f}{\partial t} dx + \int_{\partial V(t)} f u \cdot n ds. \quad (2.2)$$

For a proof of this theorem, which will be used extensively in the following, see e.g. Chapter 1.5 in [84].

Mass Equation

One of the fundamental axioms of Newtonian mechanics is the conservation of mass. It states that the change of mass M in a volume V over time t is zero:

$$\frac{dM}{dt} \stackrel{(2.1)}{=} \frac{d}{dt} \int_{V(t)} \rho dx \stackrel{!}{=} 0. \quad (2.3)$$

Expanding this equality using Reynolds' transport theorem 2.1.2 and the divergence theorem we obtain

$$0 = \frac{dM}{dt} \stackrel{(2.2)}{=} \int_{V(t)} \frac{\partial \rho}{\partial t} dx + \int_{\partial V(t)} \rho u \cdot n ds = \int_{V(t)} \left(\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) \right) dx. \quad (2.4)$$

Since the volume $V(t)$ was chosen arbitrarily and the functions are assumed to be smooth, we can get rid of the integral and deduce the *continuity equation*

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) = 0. \quad (2.5)$$

As from now, we will consider only incompressible fluids, consequently the density ρ does not change over time: $\frac{\partial \rho}{\partial t} = 0$. This turns the continuity equation (2.5) into

$$\nabla \cdot u = 0, \quad (2.6)$$

which refers to a divergence-free velocity field.

Momentum Equation

The *momentum equation* describes the motion of a fluid. It states that the change of the motion of a fluid in a volume is driven by the sum of the effects of all outer forces, which are acting on the same volume. This momentum is described by the term

$$\int_{V(t)} \rho u \, dx. \quad (2.7)$$

Using again Reynolds' transport theorem 2.1.2 (in vector form) the rate of change of the momentum can be transformed into

$$\begin{aligned} \frac{d}{dt} \int_{V(t)} \rho u \, dx &= \int_{V(t)} \frac{\partial \rho u}{\partial t} \, dx + \int_{\partial V(t)} (\rho u \otimes u) \cdot n \, ds \\ &= \int_{V(t)} \frac{\partial \rho u}{\partial t} + \nabla \cdot (\rho u \otimes u) \, dx. \end{aligned} \quad (2.8)$$

The symbol \otimes denotes the tensor product. Using the continuity equation (2.5) and the relation

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + u \cdot \nabla f, \quad (2.9)$$

(2.8) is transformed into [20]

$$\frac{d}{dt} \int_{V(t)} \rho u \, dx = \int_{V(t)} \rho \frac{du}{dt} \, dx. \quad (2.10)$$

Up to this point, we only describe the volume forces, we denote them with b , that create motion, altogether $\int_{V(t)} \rho b \, dx$, but forces acting on the surface of a volume have an influence, too. They can be expressed in terms of *Cauchy stress* σ , which is a material tensor that is symmetric, i.e. $\sigma = \sigma^\top$ under the assumption of conservation of angular momentum. Since surface forces act in normal direction on the fluid volume, we can write

$$\int_{\partial V(t)} \sigma \cdot n \, ds = \int_{V(t)} \nabla \cdot \sigma \, dx. \quad (2.11)$$

Combining volume and surface forces, we get an equation describing the conservation of the momentum:

$$\int_{V(t)} \rho \frac{du}{dt} \, dx = \int_{V(t)} \rho b \, dx + \int_{V(t)} \nabla \cdot \sigma \, dx. \quad (2.12)$$

Using the same argument as in Section 2.1.1 we conclude

$$\rho \frac{du}{dt} = \rho b + \nabla \cdot \sigma. \quad (2.13)$$

In order to get the desired form of the equation of motion, we use (2.9) again, which yields

$$\rho \frac{\partial u}{\partial t} + \rho(u \cdot \nabla)u = \rho b + \nabla \cdot \sigma. \quad (2.14)$$

We omit the energy equations because we only model the flow. Thermodynamic features like temperature and humidity would be of great interest, but add an additional complexity to the simulations and are not subject of this thesis. However, to gain a complete system with as many equations as unknowns, we need to take a closer look at the Cauchy stress tensor.

The Stress Tensor for Newtonian Fluids

Until now, we did not make any assumptions on the material properties of the fluid. We only stated that its behavior due to material constants was somehow hidden in the Cauchy stress σ , which we assumed to be symmetric. We can divide the Cauchy stress in normal and tangential forces. We interpret the normal force in an incompressible flow as (dynamic) pressure and write $p := \frac{1}{3}(\sigma_{11} + \sigma_{22} + \sigma_{33})$. The normal part of the Cauchy stress is therefore $\sigma_n = -pI$ with the identity tensor I . The tangential part describes the viscous forces. These forces are expressed by the viscous stress tensor $\sigma_v := \sigma + pI$. This yields $\sigma = \sigma_n + \sigma_v$. In general, the viscous forces depend on local changes of the velocity, which are combinations of derivatives of the velocity u . They are summarized in the deformation rate tensor D . Hence we obtain

$$\sigma_v = f(D), \quad (2.15)$$

with a material function f whose concrete identification is subject of rheology. Next, we pose some constraints on the fluid that let us derive the incompressible Navier-Stokes equations:

- We deal with *Newtonian fluids* only, this means that the stress tensor depends linearly on the deformation rate, which implies no change of viscosity with respect to velocity. Expressed as equation:

$$\sigma = -pI + f(D), \quad (2.16)$$

where f is now a continuous linear function.

- There is no viscous stress if the fluid is in rest: $f(0) = 0$.
- The fluid is isotropic, i.e. its properties stay the same in all directions:

$$f(SDS^\top) = Sf(D)S^\top \quad (2.17)$$

for every transformation S with $SS^\top = I$ and $\det(S) = 1$.

These restrictions were derived empirically on the basis of experiments and are called *Stokes' postulates* [53]. They connect the stress and the strain in the following way:

$$\sigma = (-p + \kappa \nabla \cdot u)I + 2\mu D, \quad (2.18)$$

with the bulk viscosity κ and the dynamic viscosity μ . In our case (neglecting rotational effects), the deformation rate tensor takes on the concrete form

$$D = \frac{1}{2}(\nabla u + (\nabla u)^\top). \quad (2.19)$$

Since we already decided to model incompressible fluids only, the second term in (2.18) vanishes

$$\sigma = -pI + 2\mu D. \quad (2.20)$$

Combining the momentum equation (2.14), the continuity equation (2.5) (under the assumption of incompressibility) and the just derived form of the Cauchy stress tensor (2.20), we finally end up with the *incompressible Navier-Stokes equations*

$$\begin{cases} \rho \frac{\partial u}{\partial t} + \rho(u \cdot \nabla)u - \mu \Delta u + \nabla p = \rho b \\ \nabla \cdot u = 0. \end{cases} \quad (2.21)$$

These have to be complemented with initial conditions and boundary conditions to make the system complete.

2.1.2 Reynolds Number

To compare and characterize fluid flows in different scales, we want to introduce the *Reynolds number*. It describes the dynamic similarity of flows occurring in a down-scaled domain $\tilde{\Omega} := \Omega/L$, where L is a characteristic length of the geometry, to the full-scaled domain Ω , e.g. a down-scaled model of an airplane that fits into a wind tunnel to the full-scaled airplane. For this, we have to rescale the variables in the equations as follows:

- $\tilde{x} := x/L$,

- $\tilde{u} = u/U$, with a reference speed U ,
- $\tilde{t} = Ut/L$,
- $\tilde{p} := pL/\mu U$.

Using the chain rule, i.e.

$$\frac{\partial \tilde{u}}{\partial \tilde{x}} = \frac{L}{U} \frac{\partial u}{\partial x} \quad (2.22)$$

and similar for the other variables and substituting these in the Navier-Stokes system (2.21) with $b = 0$, we obtain the incompressible Navier-Stokes equations with rescaled quantities

$$\begin{cases} \frac{\partial \tilde{u}}{\partial \tilde{t}} + (\tilde{u} \cdot \nabla) \tilde{u} - \frac{\mu}{\rho UL} \Delta \tilde{u} + \frac{\mu}{\rho UL} \nabla \tilde{p} = 0 \\ \nabla \cdot \tilde{u} = 0. \end{cases} \quad (2.23)$$

Then, the Reynolds number which represents the different flow properties is defined

$$Re := \frac{\rho L}{\mu} = \frac{UL}{\nu} \quad (2.24)$$

with the dynamic viscosity $\nu = \frac{\mu}{\rho}$. It describes the ratio between inertia and viscous forces. Now we can write (2.23) for short as

$$\begin{cases} Re \left[\frac{\partial \tilde{u}}{\partial \tilde{t}} + (\tilde{u} \cdot \nabla) \tilde{u} \right] - \Delta \tilde{u} + \nabla \tilde{p} = 0 \\ \nabla \cdot \tilde{u} = 0. \end{cases} \quad (2.25)$$

Of course, this system has to be completed with initial and boundary conditions. We discuss the numerical treatment of the boundary conditions in Section 2.2.6 and the boundary conditions that we will use in Chapter 4.

2.2 Discretization and Implementation

The *Finite Element Method (FEM)* is a commonly used method for the numerical treatment of PDEs. It is based on the variational or weak formulation of the problem. In this section, we will make use of the weak formulation and the Galerkin method to discretize and implement the incompressible, instationary Navier-Stokes equations without right-hand side

$$\begin{cases} \rho \frac{\partial u}{\partial t} + \rho(u \cdot \nabla)u - \mu \Delta u + \nabla p = 0 \text{ in } \Omega \\ \nabla \cdot u = 0 \text{ in } \Omega \end{cases} \quad (2.26)$$

and appropriate boundary conditions.

The existence of a solution $[u, p]$ in the strong sense

$$u, p \in C^\infty(\mathbb{R}^3 \times [0, \infty)) \text{ and } \int_{\mathbb{R}^3} |u(x, t)|^2 dx < C \quad \forall t \geq 0, \quad (2.27)$$

is yet to be proven or disproven. It is one of the so-called ‘‘Millennium Problems’’ [25]. However, we search in weaker spaces for a solution, which also provide a good way to make use of the Finite Element formulation for this class of problems. We will search for a weak solution in the so-called Sobolev spaces. We obtain a formulation in terms of these spaces after multiplication with test functions and integration by parts using Green’s formula. Let $\mathcal{L}^2(\Omega)$ denote the function space of all square integrable functions over the domain Ω . Together with the standard inner product

$$(u, v) = \int_{\Omega} uv \, d\Omega \quad (2.28)$$

and its induced norm $\|v\|^2 = (v, v)$, this forms a Hilbert space. $\mathcal{L}_0^2(\Omega)$ denotes the space that we will use later for the pressure in the Navier-Stokes equations. It satisfies an additional uniqueness requirement (the value of the pressure in $\mathcal{L}_0^2(\Omega)$ is only determined up to constant since only the gradient of the pressure appears in the equation) for the integral over the domain Ω :

$$\mathcal{L}_0^2(\Omega) = \{f \in \mathcal{L}^2(\Omega) : \int_{\Omega} f \, dx = 0\}. \quad (2.29)$$

The commonly used Sobolev spaces, denoted $\mathcal{H}^k(\Omega)$ with a non-negative integer k , are subspaces of $\mathcal{L}^2(\Omega)$ with a certain regularity. While \mathcal{H}^0 is actually the same space as $\mathcal{L}^2(\Omega)$, we use the multi-index notation $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_n) \in \mathbb{N}_0^n$ and $|\alpha| := \alpha_1 + \alpha_2 + \dots + \alpha_n$ to define the weak α -th derivative of $D^\alpha f = \frac{\partial^{|\alpha|} f}{\partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}}$ and hence

$$\mathcal{H}^k(\Omega) = \{u \in \mathcal{L}^2(\Omega) : D^\alpha u \in \mathcal{L}^2(\Omega) \, \forall |\alpha| \leq k\}. \quad (2.30)$$

We will often use the notation

$$\mathcal{H}_0^1(\Omega) = \{u \in \mathcal{H}^1(\Omega) : u = 0 \text{ on } \Gamma\}. \quad (2.31)$$

This is a subspace of elements in $\mathcal{H}^1(\Omega)$ with square integrable weak first derivatives that vanish on the boundary Γ of Ω . For vector valued function spaces, we write

$$[\mathcal{H}^k(\Omega)]^d = \{u_i \in \mathcal{L}^2(\Omega) : D^\alpha u_i \in \mathcal{L}^2(\Omega) \, \forall |\alpha| \leq k, i = 1, \dots, d\}, \quad (2.32)$$

where d denotes the dimension.

The choice of the spaces $\mathcal{L}^2(\Omega)$ and $[\mathcal{H}^1(\Omega)]^d$ is quite natural, since they make sense from a physical point of view. $\mathcal{L}^2(\Omega)$ only allows for finite kinetic energy, while $[\mathcal{H}^1(\Omega)]^d$ restricts the force, or the dissipation, i.e. $|\nabla u| < \infty$.

With the definition of the Sobolev spaces we can define a weak or variational form of the Navier-Stokes equations that yields an existence and uniqueness theory, at least in some cases [81].

2.2.1 Variational Form

We will introduce some additional integral forms that are usually used for the variational form of PDEs. The bilinear form

$$a(u, v) = \int_{\Omega} \nabla u : \nabla v \, dx = \int_{\Omega} \sum_{i,j=1}^d \frac{\partial u_i}{\partial x_i} \frac{\partial v_j}{\partial x_j} \, dx \quad \forall u, v \in [\mathcal{H}^1(\Omega)]^d, \quad (2.33)$$

with dimension d , the bilinear form

$$b(v, q) = - \int_{\Omega} q \nabla \cdot v \, dx \quad \forall v \in [\mathcal{H}^1(\Omega)]^d \text{ and } q \in \mathcal{L}^2(\Omega) \quad (2.34)$$

and the trilinear form

$$c(w, u, v) = \int_{\Omega} (w \cdot \nabla) u \cdot v \, dx = \int_{\Omega} \sum_{i=1}^d (w \cdot \nabla u_i) v_i \, dx \quad \forall u, v, w \in [\mathcal{H}^1(\Omega)]^d. \quad (2.35)$$

All those forms are continuous and bounded in Ω while the first introduced bilinear form is also coercive:

$$|a(v, v)| \geq \alpha \|v\|_1^2, \quad \alpha > 0 \quad \forall v \in [\mathcal{H}^1(\Omega)]^d, \quad (2.36)$$

where $\|\cdot\|_1$ denotes the norm in $[\mathcal{H}^1(\Omega)]^d$. Since we did not assume divergence-free spaces, we need to fulfill a compatibility condition: The second bilinear form needs to satisfy the inf-sup condition¹,

$$\inf_{q \in \mathcal{L}_0^2(\Omega)} \sup_{v \in [\mathcal{H}_0^1(\Omega)]^d} \frac{b(v, q)}{\|v\|_{[\mathcal{H}_0^1(\Omega)]^d} \|q\|_{\mathcal{L}_0^2(\Omega)}} \geq \beta > 0 \quad (2.37)$$

such that the (saddle-point) system we get from the Navier-Stokes equations is stable [33]. Then we can find a general weak solution pair $[u, p] \in [\mathcal{H}_0^1(\Omega)]^d \times \mathcal{L}_0^2(\Omega)$ of the Navier-Stokes equations [20].

Using this compact notation we write the Navier-Stokes equations in the variational form

$$\begin{cases} \rho(\partial_t u, \varphi) + \mu a(u, \varphi) + \rho c(u, u, \varphi) + b(\varphi, p) = 0 & \forall \varphi \in [\mathcal{H}_0^1(\Omega)]^d \\ b(u, q) = 0 & \forall q \in \mathcal{L}_0^2(\Omega). \end{cases} \quad (2.38)$$

¹The inf-sup condition is also called LBB condition after Ladyshenskaya, Babuška and Brezzi.

Note that the boundary condition $u = 0$ on $\partial\Omega$ is directly built into the Sobolev space.

2.2.2 Galerkin Method

PDEs are in general very difficult or even impossible to solve analytically. We cannot expect to be able to solve the infinite dimensional problem (2.38) in $[\mathcal{H}_0^1(\Omega)]^d \times \mathcal{L}_0^2(\Omega)$.

For the numerical solution we follow an approach known as the *conforming Galerkin method*: We replace $\mathcal{H}_0^1(\Omega)$ by some finite dimensional subspace $V_h \subset [\mathcal{H}_0^1(\Omega)]^d$, and $\mathcal{L}_0^2(\Omega)$ by some other finite dimensional subspace $S_h \subset \mathcal{L}_0^2(\Omega)$ to obtain a finite dimensional problem.

With respect to the spatial variables only, and hence discarding time derivatives for the moment, the problem reads:

$$\begin{cases} \mu a(u_h, \varphi_h) + \rho c(u_h, u_h, \varphi_h) + b(\varphi_h, p_h) = 0 & \forall \varphi_h \in V_h \\ b(u_h, q_h) = 0 & \forall q_h \in S_h. \end{cases} \quad (2.39)$$

To find solutions $[u_h, p_h] \in V_h \times S_h$ to this problem, we have to carry over the conditions from the variational form to the finite dimensional subspaces. The continuity and coercivity conditions (2.36) pose no major problem. However, a discrete version of the inf-sup conditions independent of h has to be proven for each choice of the Finite Element space individually to ensure stability [7].

2.2.3 Finite Elements

From the previous section we are able to derive a nonlinear system, once the approximation spaces V_h and S_h have been identified. Different choices of spaces will lead to different systems. In particular the choice of the basis will influence the numerical properties of the matrix. In this section, we turn to the question of how to define V_h and S_h with their corresponding bases.

There are several factors which play a role in the choice of V_h and S_h . So far, we required $V_h \subset [\mathcal{H}_0^1(\Omega)]^d$ and $S_h \subset \mathcal{L}_0^2(\Omega)$, so we cannot choose just any approximation space. In general, the larger we choose the spaces V_h and S_h , the better we can hope to approximate the real solution, however, this also increases the dimension of the nonlinear system that we have to solve. In order to be able to store and solve large systems with dimensions of the order of several millions, it is desirable for the matrix to be sparse, i.e. that almost all its entries are zero.

The Finite Element Method provides such sparse matrices by defining the space in terms of basis functions that are piecewise polynomial with a localized support. This implies that most pairs of basis functions of the correspondent spaces have zero contribution. Only those pairs whose supports are so close that they overlap, will yield a non-zero entry in the system matrix.

The supports of the basis functions are typically defined by means of a mesh, hence we have to discretize the domain Ω into Ω_h . Note that Ω_h does not have to be a subset of Ω . A mesh $\mathcal{T}_h(\Omega_h)$ is usually a partition of Ω_h into cells of a simple shape, such as triangles and quadrilaterals in 2 dimensions, tetrahedrons and hexahedrons in 3 dimensions. In this context the discretization parameter h refers to the diameter of the mesh cells. The cells in the mesh need to match certain regularity conditions so that certain error estimates hold [7].

The formal definition of a Finite Element, which Ciarlet proposed in 1978 [14] then is:

Definition 2.2.1 (Finite Element triplet). A Finite Element is a triplet (K, P, Σ) , where

- $K \subset \mathbb{R}^d$ is a polyhedral cell (geometrical dimension d),
- $P(K)$ is a space of functions defined on K ,
- $\Sigma = \{\sigma_1, \dots, \sigma_n\}$ is a basis of $\mathcal{L}(P, \mathbb{R})$; the σ_i are called degrees of freedom.

Usually, a basis of local shape functions φ_i^K for $P(K)$ is chosen such that $\sigma_i(\varphi_j) = \delta_{ij}$. With this choice, the basis is uniquely determined through the choice of the degrees of freedom. This is called the nodal basis.

Taylor-Hood Elements

In Section 2.2.1 we introduced the inf-sup condition and mentioned that it is important to make sure that it holds in the finite dimensional setting as well. Hence we need to choose pairs of Finite Element spaces V_h and S_h that satisfy this condition. In the following we will restrict ourselves to the Lagrange Finite Elements, because we gain a nodal basis this way. Other choices are possible, too, e.g. Hermite Finite Elements, which also include a directional derivative at each degree of freedom.

The order of the differential operators in the Navier-Stokes equations for the pressure and the velocity differs by one. Therefore, the natural choice of the polynomial degree of the velocity space is one order higher than for the pressure

space. However, the simplest choice of piecewise linear polynomials and piecewise constant pressure fails to fulfill the inf-sup condition. There are ways to stabilize these kinds of elements and then fulfill the inf-sup condition, see [74] for instance. Apart from this, the Taylor-Hood elements fulfill the inf-sup conditions, see [88] for a proof. The polynomial degree of the velocity space is $p = 2$, biquadratic in 2D and triquadratic in 3D, and the pressure space $p = 1$ is bilinear or trilinear, again in 2D or 3D, respectively.

2.2.4 Linearization of the Navier-Stokes Equations

If we follow the described path up to this point, we end up with a nonlinear system of algebraic equations. In order to solve this numerically, we need to apply some form of linearization to it. There are different possibilities to do that. We will use Newton's method, because the convergence rate is locally quadratic [39]. The drawback of Newton's method is that in each iterative step, the Jacobian has to be recomputed.

Let $F(x) = 0$, $x \in \mathbb{R}^N$, $F : \mathbb{R}^N \rightarrow \mathbb{R}^N$ denote a general nonlinear problem. Taylor expansion around \tilde{x} yields

$$F(x) = F(\tilde{x}) + DF(\tilde{x})(x - \tilde{x}) + \frac{1}{2}D^2F(\tilde{x})(x - \tilde{x})^2 + \dots + \frac{1}{n!}D^nF(\tilde{x})(x - \tilde{x})^n + R_n(x) \quad (2.40)$$

with the remainder term $R_n(x) = \frac{1}{(n+1)!}D^{(n+1)}F(\xi)(x - \tilde{x})$ and ξ lying in between x and \tilde{x} . We are interested in a linearization of the problem, hence we drop all terms of order two and higher and get the Newton iteration of the form

$$DF(x^k)(x^{k+1} - x^k) = -F(x^k). \quad (2.41)$$

The term $DF(x^k)(x^{k+1} - x^k)$ is the derivative of $F(x^k)$ in the direction of $(x^{k+1} - x^k)$, which is exactly the linearization. For the linear terms this is trivial. The nonlinear term gives:

$$D[c(u_h^k, u_h^k, \varphi_h)](u_h^{k+1} - u_h^k) = c(u_h^k, u_h^{k+1}, \varphi_h) + c(u_h^{k+1}, u_h^k, \varphi_h). \quad (2.42)$$

In the context of the nonlinear Navier-Stokes equations, $-F(x^k)$ can be interpreted as the residual. It is an advantage to use the residual, because it prevents rounding errors coming from non-homogeneous Dirichlet boundary conditions. The linearization with the previous Newton step solution $[u_h^k, p_h^k]$ takes on the

following form:

$$\begin{aligned}
 & D[a(u_h^k, \varphi_h) + c(u_h^k, u_h^k, \varphi_h) + b(\varphi_h, p_h^k)]((u_h^{k+1} - u_h^k), p_h^{k+1} - p_h^k) \\
 &= a(u_h^{k+1} - u_h^k, \varphi_h) + c(u_h^k, u_h^{k+1}, \varphi_h) + c(u_h^{k+1}, u_h^k, \varphi_h) + b(\varphi_h, p_h^{k+1} - p_h^k) \\
 \\
 & D[b(u_h^k, q_h)]((u_h^{k+1} - u_h^k), p_h^{k+1} - p_h^k) \\
 &= b(u_h^{k+1} - u_h^k, q_h).
 \end{aligned} \tag{2.43}$$

2.2.5 Time Discretization

Let us recapitulate the weak formulation (2.38) of the Navier-Stokes equations that we introduced previously, now assuming that we discretized them spatially but with continuous time

$$\begin{cases} \rho(\partial_t u_h, \varphi_h) + \mu a(u_h, \varphi_h) + \rho c(u_h, u_h, \varphi_h) + b(\varphi_h, p_h) = 0 & \forall \varphi_h \in V_h \subset [\mathcal{H}_0^1(\Omega)]^d, \\ b(u_h, q_h) = 0 & \forall q_h \in S_h \subset \mathcal{L}_0^2(\Omega). \end{cases} \tag{2.44}$$

We want to take care of the time discretization next. Doing this in the usual manner, we get a system of ODEs that we have to solve. This overall scheme is called the method of lines: First discretize in space, then in time. Let us introduce $F(u_h, p_h, \varphi_h)$ for the weak spatial formulation. Then the ODE system takes on the form:

$$(\partial_t u_h, \varphi_h) = F(u_h, p_h, \varphi_h) \quad \forall \varphi_h \in V_h. \tag{2.45}$$

The commonly used approaches for one-step time discretization are the explicit and implicit Euler and the Crank-Nicolson method. These methods are special cases of the θ -family of methods [20]:

$$\frac{u_h(t_{k+1}) - u_h(t_k)}{\Delta t} = \theta \partial_t u_h(t_{k+1}) + (1 - \theta) \partial_t u_h(t_k) + \mathcal{O}\left(\left(\frac{1}{2} - \theta\right)\Delta t, \Delta t^2\right), \tag{2.46}$$

where $\theta \in [0, 1]$ and $t_{k+1} - t_k =: \Delta t$. The θ -family is conditionally stable for $\theta < \frac{1}{2}$, meaning that for convergence the time-step Δt has to be chosen according to the spatial discretization, and even unconditionally stable for $\theta \geq \frac{1}{2}$. In the case of $\theta = 0$ it reflects the explicit Euler method, $\theta = 1$ is the implicit Euler method and $\theta = \frac{1}{2}$ the Crank-Nicolson method. The latter has the big advantage that it is of second order accuracy since the first part of the truncation error $\mathcal{O}\left(\left(\frac{1}{2} - \theta\right)\Delta t, \Delta t^2\right)$ vanishes for $\theta = \frac{1}{2}$.

2.2.6 Boundary conditions

At several points we mentioned boundary conditions, but we have not said anything about their treatment in the Finite Element setting, yet. In general, there are two ways:

1. either they can be incorporated directly into the space, as we did with the homogeneous Dirichlet boundary conditions, where we seek the solution in $[\mathcal{H}_0^1(\Omega)]^d$, see Equation (2.38),
2. or they can be added to the bilinear form as an extra condition in terms of a boundary integral.

In both cases it is necessary to take the boundary conditions in V_h into account in order to obtain a conforming method (i.e. $V_h \subset [\mathcal{H}_0^1(\Omega)]^d$). In the second case, it is clear how this can be done: the boundary conditions influence the bilinear form, which is directly transferred into the discretization. In the first case, however, something else has to be done to incorporate the boundary conditions into the discrete space V_h .

Returning to (2.38): If we want to impose $u = 0$ on the boundary $\partial\Omega$ of the domain, this corresponds to setting those degrees of freedom to zero that lie on the boundary. One way to do this, is to modify the linear system (2.43). For each boundary degree of freedom u_i , we need to replace one equation containing u_i by the equation $u_i = 0$. This can be done by replacing the i^{th} row of the matrix A with the identity row, with zeros everywhere except for position i , which we set to one. In position i of the right-hand-side vector b , we put a zero.

Note that we do not eliminate the column i . This is acceptable, as long as an iterative method is used to solve the system. In this case, we just need to ensure that the initial guess in the iterative method also contains the value of the boundary degree of freedom u_i , in this case zero, since this value will then be conserved throughout the iterative process.

2.2.7 Fluid Flow Solvers based on Finite Elements

Having gained an overview of the mathematical aspects of Finite Elements, we will now turn our attention to the question of how to implement solvers based on FEM. We will first look at the overall solution algorithm and then point out some details that are common to almost all implementations. For our results, we used the Finite Element Package HiFlow³.

The overall algorithm can be summarized as follows [43]:

1. Create the mesh and assign polynomial orders \rightarrow define V_h and S_h .
2. Number the degrees of freedom on all elements and enforce continuity, if necessary.
3. Start/Proceed time iteration
 - (a) Start/Proceed Newton iteration.
 - i. Assemble the linearized system taking boundary conditions into account.
 - ii. Solve the linear system.
 - (b) Update Newton iteration. If tolerance is reached, go to step 4, else, go to step 3a.
4. Update time iteration. If the maximal time is reached, go to step 5, else, go to step 3.
5. Post-process and visualize the solution.

Assembly

The classical method is the so-called cell-based assembly algorithm. For each element, the element matrix and element vector are computed in a local step that will be described later on. These local structures are then added to the global system, which has the previously mentioned saddle-point structure, using the local-to-global degrees of freedom mapping $\text{DoF}(E, i)$, which, given the element E and a local degree of freedom number $i \in \{1, \dots, n\}$, returns the corresponding global degree of freedom number.

The global assembly algorithm is rather straightforward: After having defined one element for each cell in the mesh, the global basis functions φ_i are created by combination of the local shape functions of neighboring elements. We want to obtain continuous solutions, this means that the global basis function already have to be continuous. One way to enforce this continuity for Lagrange elements can be achieved via their numbering as follows: When the same polynomial order is used on all elements, we identify the degrees of freedom that are shared by neighboring elements. Figure 2.1 illustrates this numbering of the degrees of freedom. On the left, the degrees of freedom lying at identical points on the interface between elements A and B have not yet been identified, which will make it possible to represent discontinuous functions with that basis. If one

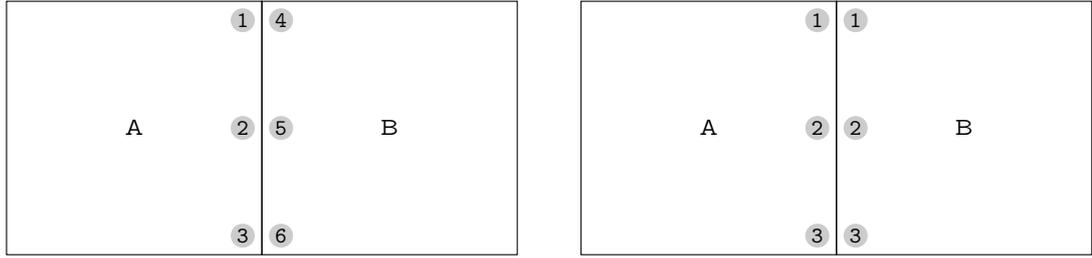


Figure 2.1: Enforcing continuity by identifying degrees of freedom.

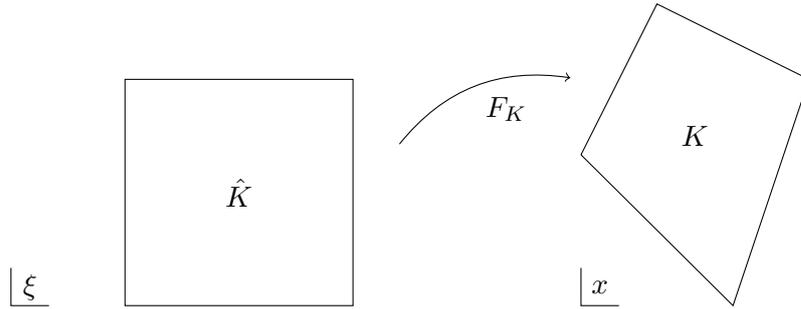


Figure 2.2: Mapping from reference element to physical element.

removes the degrees of freedom on one side of the interface, the continuity of the function will be enforced there. This can easily be understood, since the restriction of the function to the interface, in this case a quadratic polynomial of one variable, is uniquely determined by its value at the three nodal points.

The assembly is independent of the particular variational problem that is being solved. The details of the weak form are hidden in the computation of the local element matrices and vectors, which we will turn to next.

The global-local partitioning of the assembly process is motivated by the fact that the integrals for the linear and bilinear forms can be split up into a sum over the elements. If \mathcal{M} is the set of all cells in the mesh, we can partition an integral over Ω as follows:

$$\int_{\Omega} f \, dx = \sum_{K \in \mathcal{M}} \int_K f \, dx.$$

This reduces the problem of computing integrals over the whole domain, which, in general, is very complicated, to computing them over simple shapes, such as triangles or quadrilaterals. However, even integrating over a general triangle is a non-trivial task and therefore one usually introduces a reference element onto which all integral computations are transformed. Figure 2.2 illustrates the relationship between the reference element \hat{K} and an arbitrary physical element K . There is a bijective mapping F_K from \hat{K} to K for which we denote

the Jacobian matrix with $J_K = \nabla F_K$.

The following relationships make it possible to transform the integrations from the physical element to the reference element. Note that the transformation of the gradients requires the multiplication with J_K^{-T} and that the integral transformation introduces the scaling factor $|\det J_K|$:

$$\begin{aligned}\varphi(x) &= (\hat{\varphi} \circ F_K^{-1})(x), \\ \nabla\varphi(x) &= \left(J_K^{-T} (\nabla\hat{\varphi}_K \circ F_K^{-1}) \right) (x), \\ \int_K f(x) dx &= \int_{\hat{K}} (f \circ F_K)(\xi) |\det J_K(\xi)| d\xi.\end{aligned}$$

In certain situations, we are able to evaluate the integral on the reference element analytically in terms of F_K and J_K . However, numerical quadrature formulas are mostly used for the evaluation of the integral on the reference cell. A large range of quadrature rules has been developed for this purpose. Quadrature rules can be characterized by a set of n quadrature points $\xi_q \in \hat{K}$ and corresponding quadrature weights $w_q \in \mathbb{R}$. The formula for approximating an integral is then given by

$$\int_{\hat{K}} f(\xi) d\xi \approx \sum_{q=1}^n w_q f(\xi_q).$$

We use the Gauss quadrature, which is exact for polynomials up to order $2n - 1$.

Mesh Representation for Biomedical Flows

The Finite Element code HiFlow³ provides a mesh module that is especially suited for fluid flows in the context of biomedical structures. It provides functions to work with meshes of different cell types in different dimensions through a uniform abstract interface. The combination of different cell types is necessary to provide a good representation of the complex lung structure. For problems concerning fluid flows in biomedical structures, we need the ability to handle large distributed meshes for the parallelization on high performance computers. However, we want to provide local mesh refinement and coarsening for adaptive methods, too. A balance has been found between the need of high performance on the one hand and simple, maintainable code on the other. See [76] and the HiFlow³ code in the mesh module [43] for further details.

The Human Lungs

In this chapter we will give a brief overview of the human lungs starting with the anatomy, going through the function of the different structures and explaining a breathing cycle. Here, we used [27, 58, 92] as references. We continue with some of the major lung diseases and measurement methods that will help to tune the model parameters. Figure 3.1 shows a detailed scheme of the complete human respiratory system helping to get a better overview.

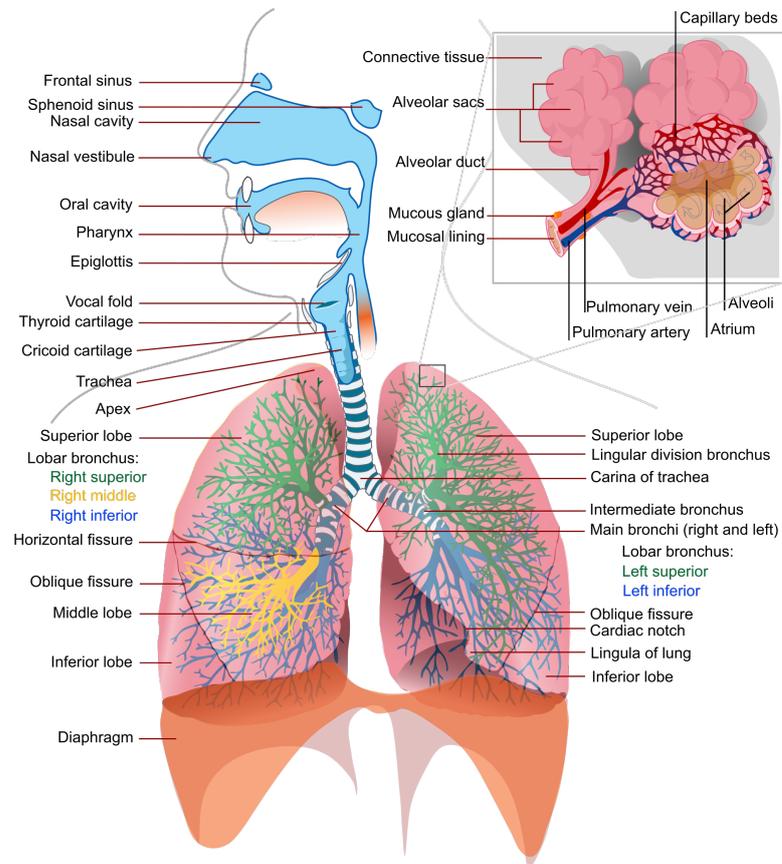


Figure 3.1: Detailed scheme of the complete human respiratory system including nose, sinuses, pharynx and lungs [93].

3.1 Anatomy

The human lungs are the organs, which are responsible for the gas exchange between the atmospheric air and the blood. As indicated by the plural, humans have two lungs that are similar but not identical. The right lung, *pulmo dexter*, is larger and consists of three *lobes*, the superior, the middle and the inferior lobe. The left lung, *pulmo sinister*, consists only of a superior and an inferior lobe, leaving space for the heart in the chest cavity. Air passes from the nose or the mouth through the *trachea*, also called windpipe, into the lobes that contain a large system of air-carrying tubes, the *respiratory tree*. The trachea is lined with 15 to 20 *cartilaginous rings*, which protect the airway reinforcing the anterior and lateral sides and keeping the upper parts of the lungs from collapsing.

The left and right main *bronchi* are branching off from the trachea. The right bronchus is wider in radius, shorter until it branches out and steeper in the angle, which yields the asymmetry of the lungs. The main bronchi bifurcate in thousands of smaller *bronchioles*.

Bronchioles are very small and shaped like tubes. They are not longer surrounded by cartilage. The bronchioles are sometimes separated into *terminal* and *respiratory bronchioles*, depending on the considered literature. The terminal bronchioles are the last part of the *conducting zone* of the lungs, while the respiratory bronchioles are connected to the *alveoli*, which together make up the *respiratory zone*. See Figure 3.2 for a schematic overview. In it, the respiratory zone is called “Acinar airways” and the conducting zone is called “Conducting airways”.

The conducting zone does not exchange gas with the blood, it merely transports the air. Due to many bifurcations in this region, the ratio of surface to volume increases drastically. The walls of the tubes have a mucus layer and very small hair, called *cilia*, to filter the air, cleaning it from dust and other particles. Through movement of the cilia, waste is transported back up to the *pharynx* or the nostrils, where it is either transported to the digestive system or blown out.

The alveoli are the smallest anatomical structure in the lungs. They form a hollow cavity with very thin walls. These walls are surrounded by blood capillaries so that oxygen and carbon dioxide can dissipate from and into the blood. The alveolar walls have cells which are producing *surfactant* that reduces the surface tension and keeps the alveoles from collapsing during the expiration phase. Alveoli contain also *macrophages* that are able to digest dust, bacteria

and carbon particles. Humans have about 300 million alveoli with a diameter of 200 to 300 μm and a total surface area of about 75 m^2 .

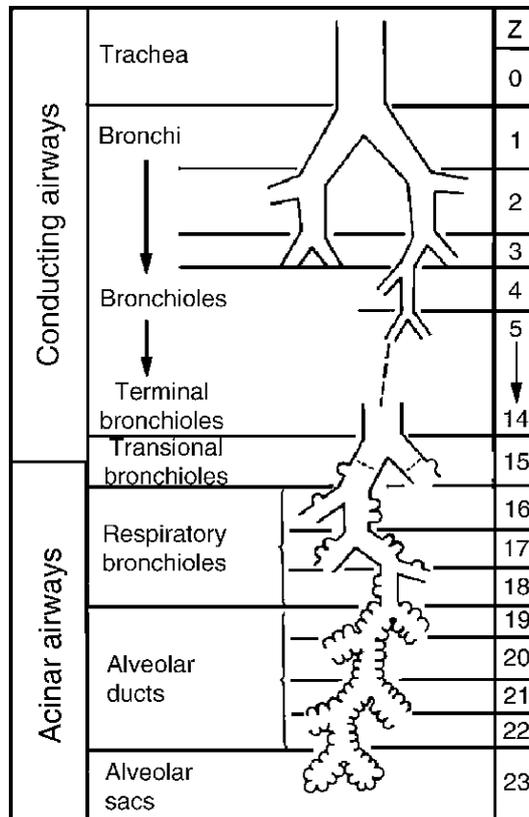


Figure 3.2: Scheme of human respiratory system partitioned in conducting and respiratory zones with the resp. anatomical structures, taken from Weibel et. al. [91]. Z stands for the generation number.

3.2 Function

The human lungs are located in the chest, which is, roughly spoken, an airtight cylinder bounded from below and above by the *diaphragm* and the neck. The ribs form a wall around the lungs and are held together by three layers of *intercostal muscles* (cf. Figure 3.1). During inspiration the volume of the chest increases by lowering the diaphragm and lifting the *sternum* creating a negative pressure. This results in a lung pressure that is below the atmospheric pressure, hence air is sucked in through the airways. Expiration on the other hand is a passive process. Due to the return of the diaphragm and the sternum to their initial positions, air is exhaled.

The gas exchange between lungs and blood happens in the alveoli. The walls

of the alveoli are extremely thin and surrounded by also very thin-walled capillaries. Oxygen depleted and carbon dioxide rich blood is transported back from the right heart chamber. Due to the difference in the partial pressure of oxygen and carbon dioxide between the blood, rich in carbon dioxide and the fresh air, oxygen diffuses in and carbon dioxide out of the blood. Oxygen binds with the hemoglobin of the red blood cells.

3.2.1 Lung Function Values

To diagnose lung diseases, different standard lung function values with healthy and pathological standard ranges exist. The upcoming list is not meant to be complete, but gives an overview and explains some used parameters and their abbreviations [89]. A schematic overview is given in Figure 3.3.

(ERV) The *expiratory reserve volume* is the difference between FRC and RV.

(FEV) The *forced expiratory volume* measures the amount of forced exhaled air after normal breathing. Three sub-parameters exist:

(FEV1) Measurement after one second.

(FEV2) Measurement after two seconds.

(FEV3) Measurement after three seconds.

(FRC) The *functional residual capacity* measures the amount of air left in the lungs after a normal exhalation.

(FVC) The *forced vital capacity* measures the amount of forced exhaled air after a deep inhalation.

(IRV) The *inspiratory reserve volume* is the difference between TLC and VC.

(MVV) The *maximum voluntary ventilation* measures the maximum amount of inhaled and exhaled air during one minute.

(PEF) The *peak expiratory flow* measures the maximum exhalation speed and is measured the same way as the FVC.

(RV) The *residual volume* measures the amount of air left in the lungs after a complete exhalation.

- (SVC) The *slow vital capacity* measures the amount of very slowly exhaled air after a maximum inhalation.
- (TV) The *tidal volume* is the amount of normal exhaled air after a normal inhalation.
- (TLC) The *total lung capacity* measures the amount of air in the lungs after a maximum inhalation.

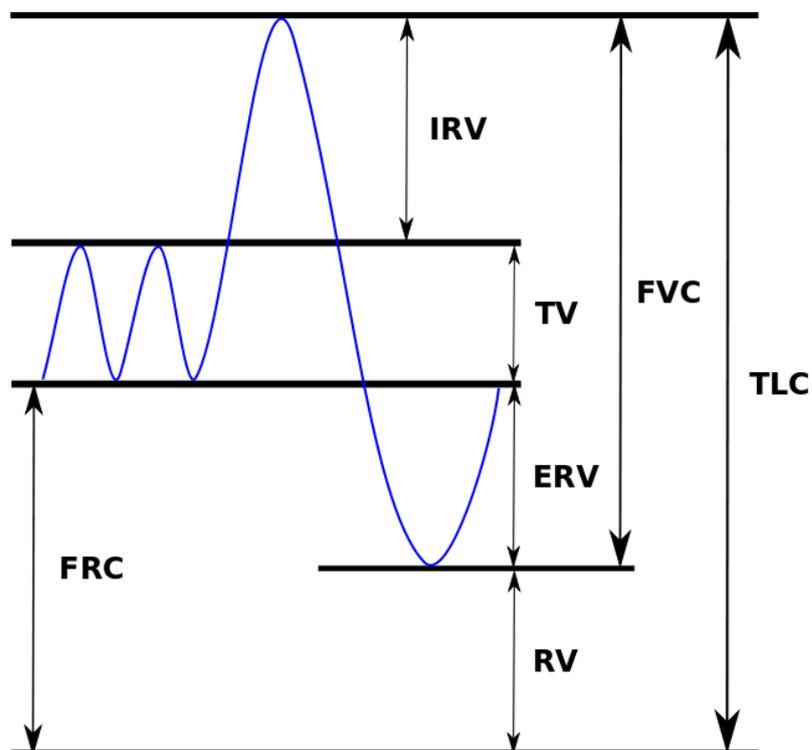


Figure 3.3: Schematic representation of some of the above described lung function values that help to diagnose lung diseases.

3.2.2 Airflow Values

Typical values of respiration frequency, flux and ventilation rate for different levels of activity for an adult man are given in Table 3.1. This table is based on the tables in the *International Commission on Radiological Protection (ICRP)* [87] and gives a good overview of different situations of concern. The mean ventilation rate is computed assuming a sine-shaped function for inhalation and exhalation.

Activity	Resting (sleeping)	Sitting awake	Light exercise	Heavy exercise
respiration frequency in min^{-1}	12	12	20	26
flux (same as TV in Fig. 3.3) in ml	630	750	1300	1900
mean ventilation rate in ml/s	126	150	433	823

Table 3.1: Typical values of respiration frequency, flux and ventilation rate.

3.3 Major Lung Diseases

Respiratory illnesses are the second most frequent diseases with respect to mortality, incidence, prevalence and cost for the health care system [8]. While the symptoms can be treated, still, very little is known about the actual cause of these symptoms.

3.3.1 Asthma

Asthma is the most common lung disease. Every 10th person in Europe suffers this disease while the numbers still increase [8].

Asthma is a disease of the bronchi. It can be caused by bronchospasms, increased generation of mucus or mucosal swelling in the bronchi. This leads to cough, gasping and shortness of breath due to narrower airways.

Infections of the respiratory system, coldness, pollen, industrial exhaust gases, cigarette smoke, allergic reactions as well as certain emotional or psychological states can cause asthmatic attacks.

3.3.2 Chronic Obstructive Pulmonary Disease

Chronic obstructive pulmonary disease (COPD) is a combination of a chronic bronchitis and an emphysema. It is far spread in the western hemisphere and affects usually strong smokers and people, who are exposed to exhaust gases on a daily basis.

A chronic bronchitis makes itself felt by daily cough, sputum and increased formation of mucus for a period longer than three months in two years. An emphysema is usually the follow-up of a long lasting chronic bronchitis. Frequent infections, destruction of lung tissue and inflation of alveoli leads to it. Alveoli get less efficient hence the FRC is increased which causes the lungs to expand.

The actual cause of COPD despite smoking is unknown. COPD is to date incurable but the symptoms can be treated with various drugs and inhalators.

3.3.3 Lung Cancer

Lung cancer is the type of cancer, that is located in the lungs. It usually starts in the cells that are along the tubes of the lungs. The two main types of lung cancer are:

- Non-small cell lung cancer (NSCLC), the most common type of lung cancer
- Small cell lung cancer, about 20% of all lung cancers.

Lung cancer is the deadliest kind of cancer. More people, regardless of gender, die of lung cancer than of any other kind of cancer combined [1, 52].

3.4 Measurement Methods

3.4.1 Computer Tomography

Computer tomography (CT) scans are the basis of almost every patient specific computation we do in the context of biomedical fluid flow simulations. The technology is based on X-rays. A rotating anode X-ray tube rotates around the object or patient we want to examine. With the inverse Radon transformation, a way to recreate functions with the integration along certain manifolds, we can recover a slice of the object. Putting several slices together, we get a 3D scan of the object. From this 3D scan, we can segment the region of interest, the lungs in our particular case, and use it for simulations in that region. For further technical as well as algorithmic details on the reconstruction of the images see [21]. The cast in Figure 3.4 is one of the casts that we used for our simulations.

3.4.2 Rhinomanometry

The active anterior rhinomanometry is a method to measure the air volume and the air pressure during respiration in the nose. The acquired data can be used as initial conditions for fluid flow simulations of the human nose and lungs.

Process: The patient sits in an upright position. Using a breathing mask, a pressure gauge and a flowmeter (cf. Figure 3.5a) one measures the pressure difference (P) and the flow as a volume per time (FL) at the inhalation (**In**) and exhalation (**Ex**) on the right (**R**) and left (**L**) nostril.

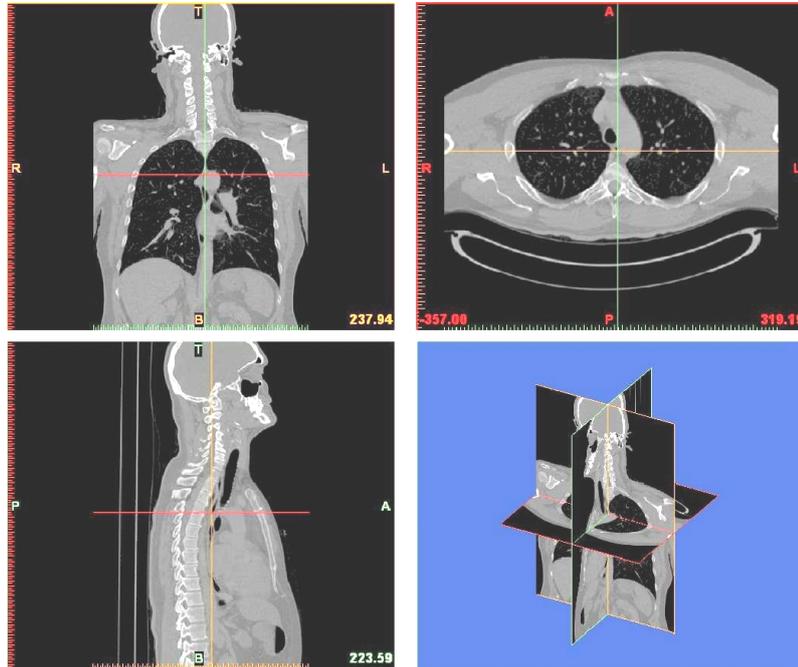


Figure 3.4: CT cast of a West-European middle-aged male. This CT-cast was acquired at the Städtisches Klinikum Karlsruhe. It is one of the casts that served as basis for the segmentation and hence simulation of the (patient-specific) lung.

The measurement must be performed for both nostrils. The patient should breathe quietly through several cycles. At least four to five cycles are measured and afterwards averaged.

As a result one obtains two curves, one for each nostril. Default maximum values are found in the literature for the pressure to be $\pm 150 Pa$ and for the flow $\pm 400 mL/s$ (cf. 3.5b) [48].

3.4.3 Spirometry

The *spirometry* is a commonly used *pulmonary function test* (PFT). It evaluates how well, specifically how much and how quickly, air is moved in and out of the lungs. The TLC can be measured as well as some other lung function values, like the ventilation of specific parts of the lungs and changes during a breathing cycle. Spirometry can be done in a silent state and during exercise. The latter is called *ergospirometry* and enables to analyze the performance capacity of the cardiopulmonary system.

Process: The patient sits in an upright position and breathes in and out through a mouthpiece into a closed container. This container swims in a fluid like water

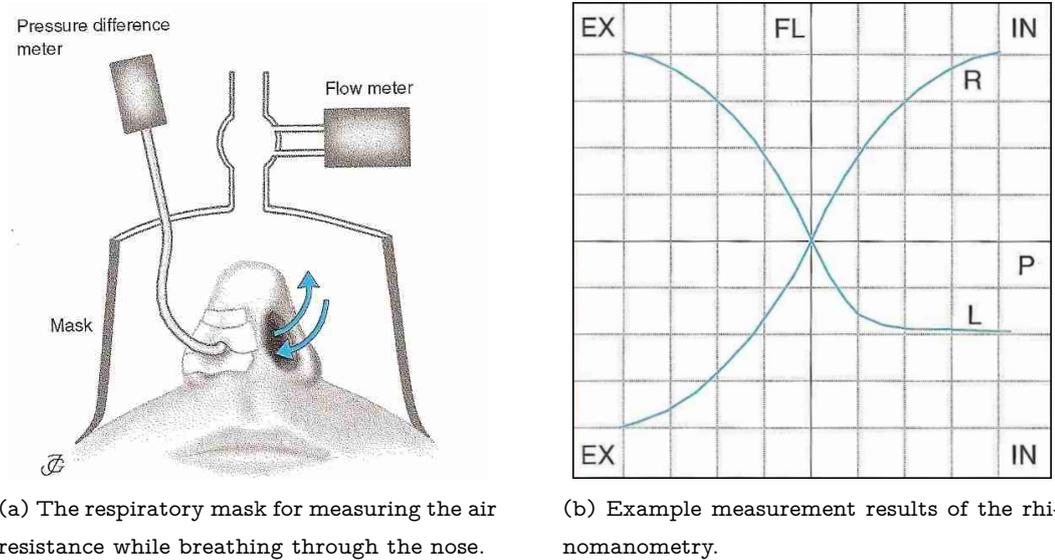


Figure 3.5: The rhinomanometry method [48].

and can move freely up and down. Due to the change of volume, the container moves up and down (cf. Figure 3.6a). The resulting curve is plotted (cf. Figure 3.6b). It is called a *spirogram*. The spirogram is used to diagnose e.g. COPD and asthma [71].

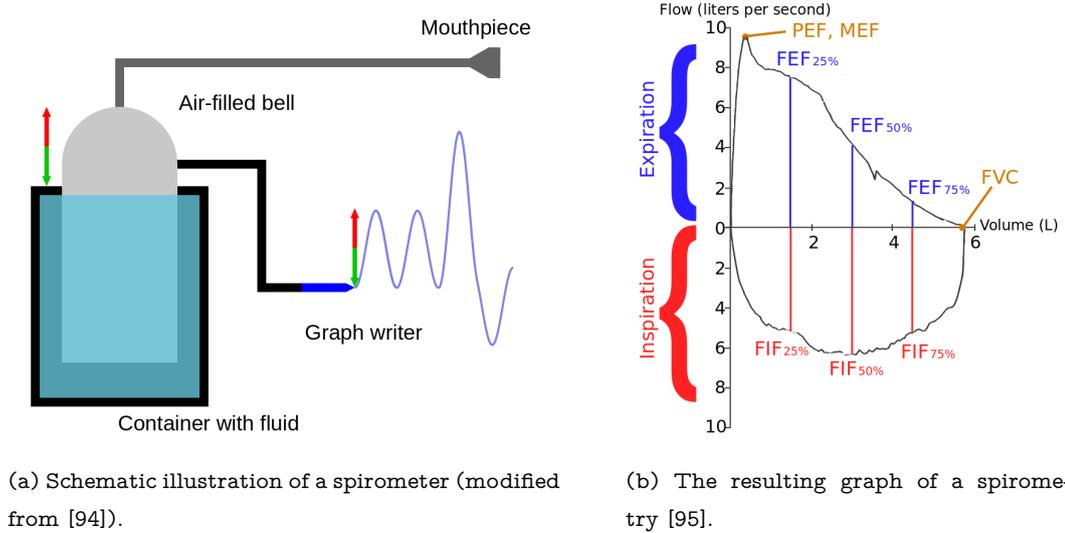


Figure 3.6: Illustration of a spirometry, the most commonly used lung function test.

3.4.4 Hyperpolarized Helium-3 MRI

Functional tomography with the help of *Magnetic Resonance Imaging* (MRI) is one of the relatively new trends in biomedical engineering. The problem with CT data is that it cannot depict the air in the lungs, but only the surrounding tissue. This has changed with the use of hyperpolarized helium-3 MRI (hyperpolarized ^3He MRI). It is possible to visualize the ventilation of the lungs, which can provide data to adjust our models. Regarding illnesses like asthma or emphysema, this is very good news. Hyperpolarized ^3He MRI, in contrast to the previously used PET (positron emission tomography) scanners that utilized radioactive gases, is harmless for the patient. With the help of polarized laser light, the spins of the ^3He atoms are aligned on one axis. After the patient has breathed in this polarized noble gas and held his breath, a normal MRI is performed. Through a new technique that allows fast polarization of the gas, it is possible to record pictures in tenths of seconds, such that a movie of the lung function can be created. Hence restrictions of the airflow through the lungs can be easily visualized. In contact with the breathed oxygen, it is possible to calculate the decay time. This gives valuable insight in the oxygen content and consumption of the lungs and allows to distinguish between healthy and pathological, usually inflated, tissue at the alveolar level [68].

Models of the Human Lungs

In this chapter, we want to give an overview of models that are used to describe the lower regions of the lungs. Several mainly anatomical features are used for these models. We briefly describe a model by Grandmont et al. [37] that we implemented and tested numerically. Then we introduce models that help to determine boundary conditions for the resolvable part of the lungs. These models are explained in detail and tested in a “worst-case scenario”, which is a tree that has artificial and unphysiological boundaries arising from the resolution of nowadays CT scanners.

Simulations of the human lungs are difficult both due to the complexity of biological structures and nonlinearities in the equations that describe fluid flows, such as the airflow in the human lungs. These aspects entail the modeling, because we need to reduce the systems, which are too big to compute even for nowadays high-performance computers, to the necessary information that we are interested in. A good compromise is needed between models that can be used generically for all lung geometries, i.e. for all human lungs, and models that represent individual characteristics of a certain patient. The goal is to identify and tune parameters so that the models can represent different kinds of lung diseases (see Section 3.3).

The first model we describe in Section 4.1 uses a spring-mass system to model the movement of the alveoles. This system is coupled to a dyadic tree mimicking the airflow in the tubes that represents the viscoelastic behavior.

The other two models can be used as lower part in a two-scale model. The upper part is modeled by 3D numerical flow simulations (3D-NFS), which are usually very unstable due to high Reynolds numbers in a complex geometry. Therefore, the lower part models that are used to specify boundary conditions for Navier-Stokes simulations of the bronchiole tree need to be very sensible with respect to specific necessities of the 3D-NFS.

A pressure-driven bronchiole model is introduced in Section 4.2. This model

is highly dependent on the number of the Weibel generation [90] where it is applied to the upper geometry. The coupling point between the model and the resolvable geometry is determined by counting the Weibel generations above, which assumes the lungs to have 23 generations. No variation in the total number of generations is allowed. This aspect is eventually taken into account and circumvented in another model. We describe this area-determined bronchiole model in Section 4.2.2. Both models try to overcome the difficulties of setting artificial boundary conditions to 3D-NFS.

4.1 Viscoelastic Model with Non-local Damping

Grandmont et al. introduced a viscoelastic model with non-local damping in [37]. It is motivated by viewing the lungs as a continuous and deformable object. The model is strongly influenced by the desire to obtain existence and uniqueness proofs. To achieve this, some physical aspects were disregarded, but it can be treated in a functional analytical framework.

The lungs are modeled as an infinite one-dimensional dyadic tree which is connected to a spring-mass system with dissipation A . It describes the movement u of the alveoles during inspiration and expiration. This discrete approach is transformed into a continuous model and studied asymptotically by letting the number of generations of the lungs and the size of the spring-mass system tend to infinity. From this, the partial differential equation

$$\partial_{tt}u - \partial_{xx}u + A(\partial_t u) = f \tag{4.1}$$

with the non-local dissipative term A is deduced.

They showed in [37] that the solution of the finite dimensional setting converges to the solution of the partial differential equation (4.1) in a weak sense.

We examined this model numerically in [28] and confirmed that the model converges in a weak sense. However, the model turned out to be substantially dependent on boundary conditions that were neglected or set to zero in the theoretical proof. This shortcoming rendered this approach inapplicable to our simulation setup.

4.2 Two-scale Models

The essential point of our two-scale model is that it couples two different models: The model of the upper, resolvable lung geometry and the model for the lower

part of the lungs, the bronchioles. The upper, resolvable lung geometry is made up of the trachea and the bronchi, cf. Figure 3.2 on page 23. At the outlets of the bronchi, we take bronchiole tree models of the lower respiratory system into account. These models yield an inflow condition at each outlet, which mainly depends on the generation number and the area of each outlet, as well as on the pressure in the alveoles and the resistances due to the radius of the bronchioles. A schematic view of the two-scale model coupling is illustrated in Figure 4.1.

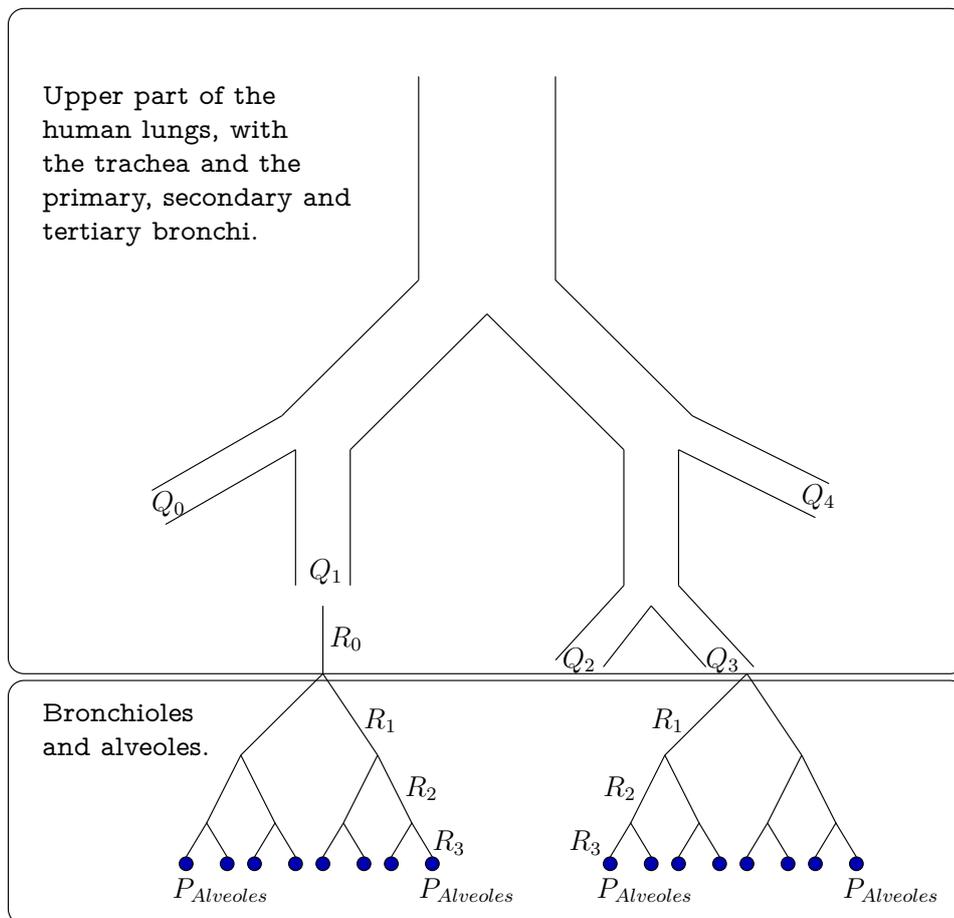


Figure 4.1: The scheme of the coupling between the upper geometry and the lower lung model.

4.2.1 Pressure-driven Bronchiole Model

Our main requirement for the model of the lower human lungs is to provide adequate boundary conditions for fluid flow simulations of the upper part of the lungs. This means that we have to extract a flow profile that reflects the complete flux in the lower lungs. To achieve this, we assume the airflow in the

bronchioles to be laminar and linear, i.e. Poiseuille's law is satisfied. To keep the model as simple as possible, we use a dyadic tree to represent the bronchioles and assume the same Weibel generation [90] to have equal resistances. The resistances decrease uniformly over the generations by a factor

$$\alpha = h^{-3} \quad (4.2)$$

that we will determine and motivate later on. This results in

$$R_i = R_0 \alpha^i \quad (4.3)$$

for the resistances R_i of the i -th generation. This is reasonable due to the fact that the radius and length of each bronchiole decrease uniformly. We further assume that the pressure is the same for each alveolus [37]. The formula of Hagen-Poiseuille combined with a parabolic flow profile leads to a relation between the flow Q and the maximum velocity v_{max} of the flow profile on each outlet of the upper airways. If we insert the parabolic flow profile

$$v_{max} = \frac{\Delta p}{4\mu l} r^2 \quad (4.4)$$

into the formula of Hagen-Poiseuille

$$\Delta p = \frac{8\mu l Q}{\pi r^4} \quad (4.5)$$

with

Δp	the pressure drop,
l	the length of pipe,
μ	the dynamic viscosity,
Q	the volumetric flow rate,
r	the radius,

we get the relation

$$v_{max} = \frac{2}{\pi r^2} Q \quad (4.6)$$

between the flow and the maximum velocity in the parabolic flow profile. We can modify this to represent the more general case that uses the surface A of the outlet:

$$v_{max} = \frac{2}{A} Q. \quad (4.7)$$

The formula of Hagen-Poiseuille also determines α , since we assume that the radii and lengths of the bronchiole change by a factor h . We have l in the

numerator and r^4 in the denominator, so we get in total the factor h^{-3} . This factor is extensively described in Section 4.2.2.

We determine Q through the tree model with the electric analogy

$$U = RI. \quad (4.8)$$

U corresponds to the pressure difference Δp , R is the resistance and I the flux Q . We consider the dyadic tree as a series of resistances. Parent to child nodes are connected in series while the child nodes are connected in parallel to each other. This yields for the total resistances of the bronchiole tree

$$R = R_j + \left(\frac{2}{R_{j+1}}\right)^{-1} + \left(\frac{4}{R_{j+2}}\right)^{-1} + \dots + \left(\frac{2^k}{R_k}\right)^{-1}, \quad k > j, \quad (4.9)$$

with

$$k = \#(\text{generations}) - \#(\text{generations of the upper connected tree}),$$

and

$$R = \sum_{i=j}^k \left(\frac{2^i}{R_i}\right)^{-1}, \quad (4.10)$$

where i denotes the current generation number and j the coupling point, i.e. the tree model is attached to the upper geometry (cf. Figure 4.1) at generation number j .

The pressure in the alveoles (level K) corresponds to the pressure at the outlets (level j)

$$P_k = P_{k-1} = \dots = P_j, \quad k > j \quad (4.11)$$

of the upper geometry. This holds, because we assumed equal resistances in each generation level.

Inserting (4.10) and (4.11) into (4.8) yields

$$Q = \frac{P}{R} = \frac{P_j}{\sum_{i=j}^k \left(\frac{2^i}{R_i}\right)^{-1}} = \frac{P_j}{\sum_{i=j}^k \frac{R_i}{2^i}}. \quad (4.12)$$

The generation number j where the flow condition couples with the upper tree and the corresponding area A are given. Therefore, the parameters that have to be set are the pressure in the alveoles

$$P_k = P_j, \quad (4.13)$$

the initial resistance R_0 of the largest bronchiole and the ratio h , which determines α . For both exist rough literature values, see for instance [72]. These values have to be fine-tuned in order to match with volume flow measurements from a respiration cycle [30].

4.2.2 Area-determined Bronchiole Model

The homothety ratio h describes the change of the radii of subsequent air pipes. It has been investigated extensively by Mauroy et al. in [64]. There, the trade-off between robustness on geometry variations in the human lungs and optimality with respect to the resistance of airways and volume uptake is analyzed. In an optimal lung structure, this ratio has been determined to be $h \approx 0.79$, whereas, due to the necessary robustness requirements of the human lungs, the actual homothety ratio is about 0.85. This matches the values in the literature [58, 75, 90]. The mean external diameter of the trachea is about $d_0 \approx 20mm$. It decreases over about 23 generations to $d_{23} \approx 0.5mm$ at the terminal bronchioles, which is again the mean literature value for the external alveolus diameter. If we take the trachea diameter d_0 , the homothety ratio $h = 0.85$ and 23 Weibel generations as input, we obtain a diameter in the terminal bronchioles of

$$d_{23} = d_0 * h^{23} \approx 0.48mm. \quad (4.14)$$

Again, we assumed an underlying dyadic tree for the bronchioles.

To determine the generation number of the interface of our two-scale model, we reverse this process: We know the diameter of the bronchi(oles) at the coupling points between the model and the resolvable upper geometry. It is therefore easy to compute the generation number where the bronchiole model is applied, see Section 4.2.2 for further details on the computation.

Summarizing, the area-determined bronchiole model is able to compute the generation number j , one of the parameters in the pressure-driven bronchiole model 4.2.1 that we had to fix there. Therefore, the area-determined bronchiole model depends strongly on the patient-specific geometry.

Generation determination

For the determination of the generation number of the bronchioles it is convenient to assume that the coupling area is shaped circularly, hence

$$A = \pi r^2 \quad (4.15)$$

holds. This is a reasonable assumption for patient-specific geometries of the upper lungs. We can validate it qualitatively by investigating CT data as shown in Figure 4.2.

For the bronchiole model introduced in Section 4.2, Eq. (4.15) is a necessary condition, since Poiseuille's law would not hold otherwise. With the homothety



Figure 4.2: Part of the CT cast (values inverted) of a West-European middle-aged male with an arrow pointing to the trachea. This CT-cast was acquired at the Städtisches Klinikum Karlsruhe.

ratio

$$h = 0.85, \quad (4.16)$$

the diameter at the trachea $d_0 \approx 20\text{mm}$ and

$$d_i = d_0 h^i, \quad (4.17)$$

this yields

$$A_i = \pi r_i^2 = \frac{\pi}{4} d_i^2 \quad \text{for } i = 0, \dots, \#\text{outlets}. \quad (4.18)$$

If we plug (4.17) in (4.18), we get an expression for the area at each bronchi

$$A_i = \frac{\pi}{4} d_0^2 h^{2i} \quad (4.19)$$

and with it the reversely determined generation number of each bronchi

$$j = \frac{1}{2} \frac{\log A_i - \log \frac{\pi}{4} d_0^2}{\log h}. \quad (4.20)$$

4.3 Numerical Experiments

In this section, we present numerical experiments of airflow simulations in an unevenly branched tree pipe system (cf. Figure 4.3). The geometry, which resembles very asymmetric human lungs, is built in order to demonstrate the weakness of conventional boundary conditions. In particular, we will see that the do-nothing boundary condition

$$\frac{\partial \tilde{u}}{\partial n} - \tilde{p}n = 0 \quad \text{on } \Gamma_{out}, \quad (4.21)$$

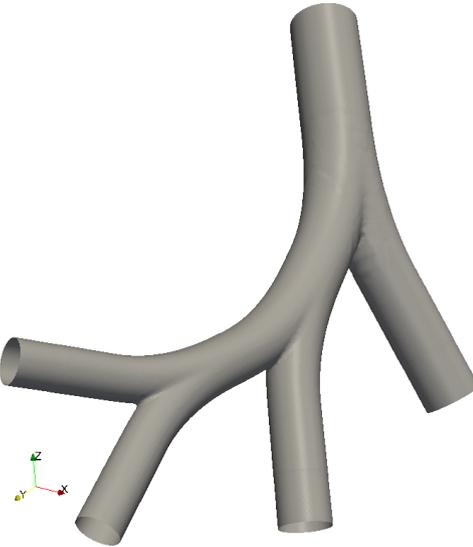


Figure 4.3: An unevenly branched tree pipe system to test different boundary condition models.

at the outlets of the lungs Γ_{out} , which seems to be a natural choice at first sight, fails to capture the physical behavior.

The tests were run sequentially on a Linux system with Intel Quad CPU9300 2.50 GHz processors and 5.7 GiB memory. We used the incompressible stationary Navier-Stokes equations that we can solve with using HiFlow³ to simulate inhalation of air. The material parameters for the air are at 20°C and normal pressure. The maximum velocity in this setup is 0.05m/s. The rather low velocity is used to make sure that no instabilities occur in the computational domain. Both, the bronchiole model and the area-determined model are used in the form of a computational library that is easily usable also in combination with other fluid flow simulation packages.

The trachea diameter at the top is 2cm, the length to diameter ratio $L/D = 3$ and the homothety ratio is $h = 0.79$. The latter quantifies the decrease of the diameter per generation and therefore the resistance of the tree. The tree geometry consists of 5,563 points and 32,343 cells in total. We used Taylor-Hood elements in the Finite Element ansatz, cf. Section 2.2.3, which yielded 130,288 DoFs.

We measure the flow going downwards through inscribed slices just below the first bifurcation (cf. Figure 4.4) to compare the different boundary condition models. The different behavior of the models can be seen easily at these points. The bulk flow is in direction of the negative z-axis, therefore all flows have a

negative sign. However, we are interested only in the qualitative behavior.

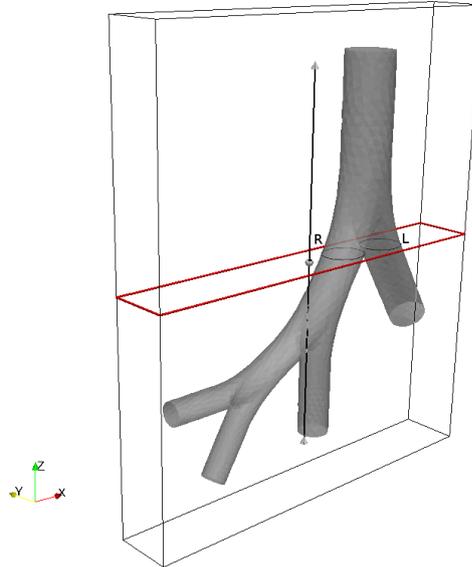


Figure 4.4: Inscribed z-normal slices just below the first bifurcation in the tree pipe system to compare the flux of the different boundary condition models. The right (R) and left (L) main bronchi are indicated.

The overall flow through both slices is set to $Q \approx -5.6 \text{ cm}^3/\text{s}$. We compare four different setups, see also Figure 4.5:

1. The do-nothing conditions at the outlet with a Dirichlet inflow condition at the top determined by the area-generation model, upper left sketch in Figure 4.5.
2. The do-nothing conditions at the outlet with a Dirichlet inflow condition at the top determined by the pressure-driven bronchiole model, upper right sketch in Figure 4.5.
3. The area-generation model set at the outlets with do-nothing conditions at the top, lower left sketch in Figure 4.5.
4. The pressure-driven bronchiole model set at the outlets with do-nothing conditions at the top, lower right sketch in Figure 4.5.

From a physiological and anatomical point of view we expect an almost equally distributed flow in all five lobes of the human lungs. This results in a flow in the left lung (the side where the heart is located) of about 40%, and in the right lung of about 60%. Due to the resolution of the CT data it is not possible to geometrically model all tubes in the lungs. The right side with three lobes is

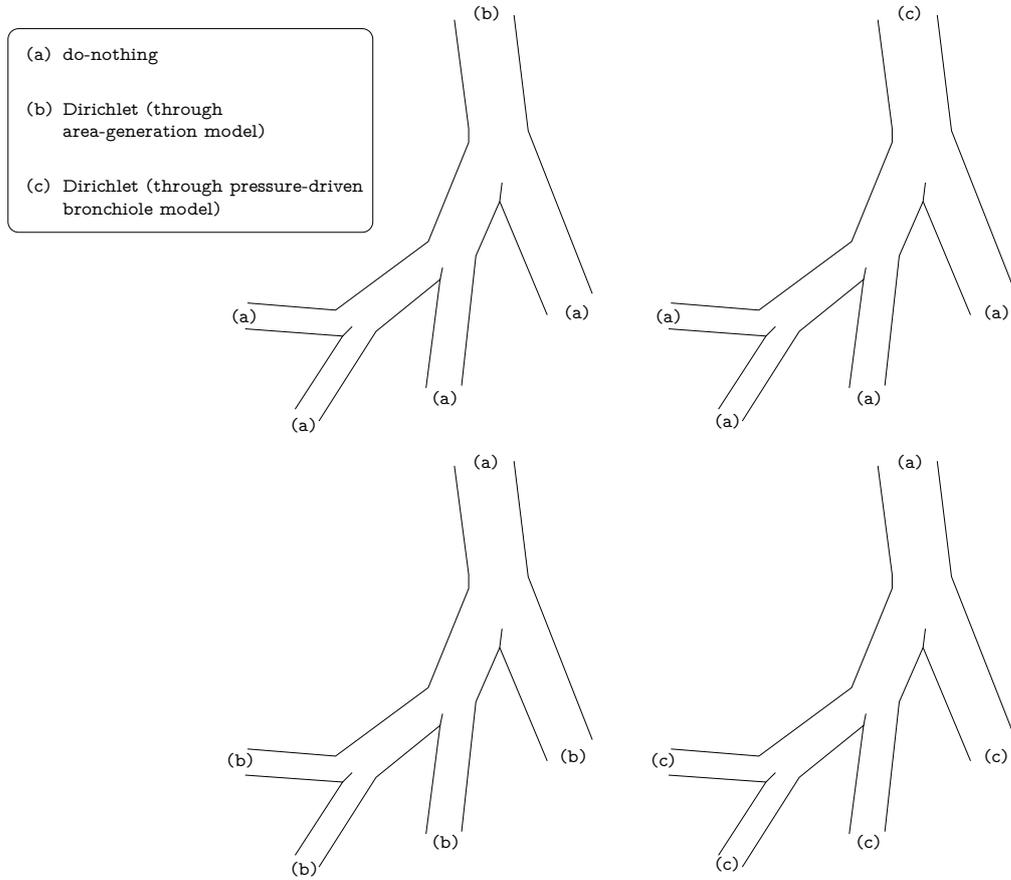


Figure 4.5: Sketch to clarify the setup for testing the different boundary conditions.

further resolvable as the left side with only two lobes, because its size is larger. This has to be considered in the model. The simulation results we will discuss in the following are summarized in Tables 4.1 and 4.2.

The do-nothing boundary condition satisfies a Poiseuille flow, which we usually assume if we know nothing particular about the region beyond the outlet. The do-nothing condition implies

$$\int_{\delta\Omega_{out}} p do = 0 \tag{4.22}$$

as hidden condition for the pressure p . This condition means that the mean pressure value is zero over all outlets, which holds not entirely true for the case of asymmetric bifurcations where the mean pressure is different from zero at the outlets. If we assume do-nothing conditions anyway, we obtain a flux that escapes primarily through the thickest and shortest way [6, Ch. 4]. However, the cuts in the geometry are artificial so this is not the desired behavior. Both tested do-nothing approaches gave the same result (cf. Figures 4.6a and 4.6b and Tables 4.1 and 4.2), because the homothety ratio $h = 0.79$ reflected in the schematic tree

Absolute flow through z-normal slices			
	Right slice	Left slice	Both slices
Do-nothing (1)	-2.158	-3.441	-5.599
Do-nothing (2)	-2.158	-3.441	-5.599
Area-generation model	-2.360	-3.253	-5.612
Bronchiole model	-3.218	-2.370	-5.589

Table 4.1: Comparison tables for the fluxes through z-normal slices in absolute numbers.

Relative flow through z-normal slices in %			
	Right slice	Left slice	Both slices
Do-nothing (1)	38.5	61.5	100
Do-nothing (2)	38.5	61.5	100
Area-generation model	42.0	58.0	100
Bronchiole model	57.6	42.4	100

Table 4.2: Comparison tables for the fluxes through z-normal slices in relative numbers.

pipe system gives the same result for the inflow as the homothety ratio $h = 0.85$ used in the model. This leads to the same (Dirichlet) input condition for both models at the top inlet. Nevertheless, both results are obviously wrong.

The area-generation model gives slightly better results, but the bulk flow is still pronounced in the wrong direction (cf. Figure 4.7a and Tables 4.1 and 4.2). This is due to the different homothety ratios. They lead to a wrong value of the generation numbers of the corresponding outlets. If the homothety ratios were the same, the results would be equal to the bronchiole model. However, this model is created for bigger differences in the sizes of the outlets, which is the designated use case. This means that the results obtained here do not reflect the strength of the model. When applied deeper in the tree it will be more sensitive than the bronchiole model and give better results.

The bronchiole model yielded the best results in this setup with respect to the expected outcome. It models the flow very closely to the correct anticipated physiological and anatomical results with a relative difference of only about 2.4% (cf. Figure 4.7b and Tables 4.1 and 4.2).

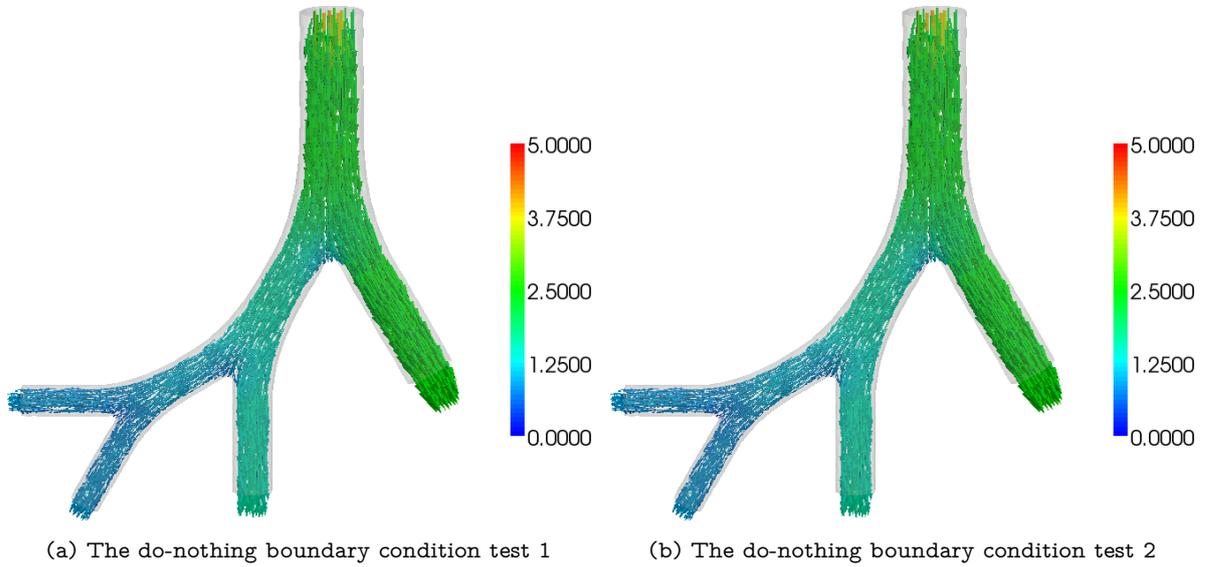


Figure 4.6: Resulting glyphs of the boundary condition tests for the setups 1 and 2 on page 39.

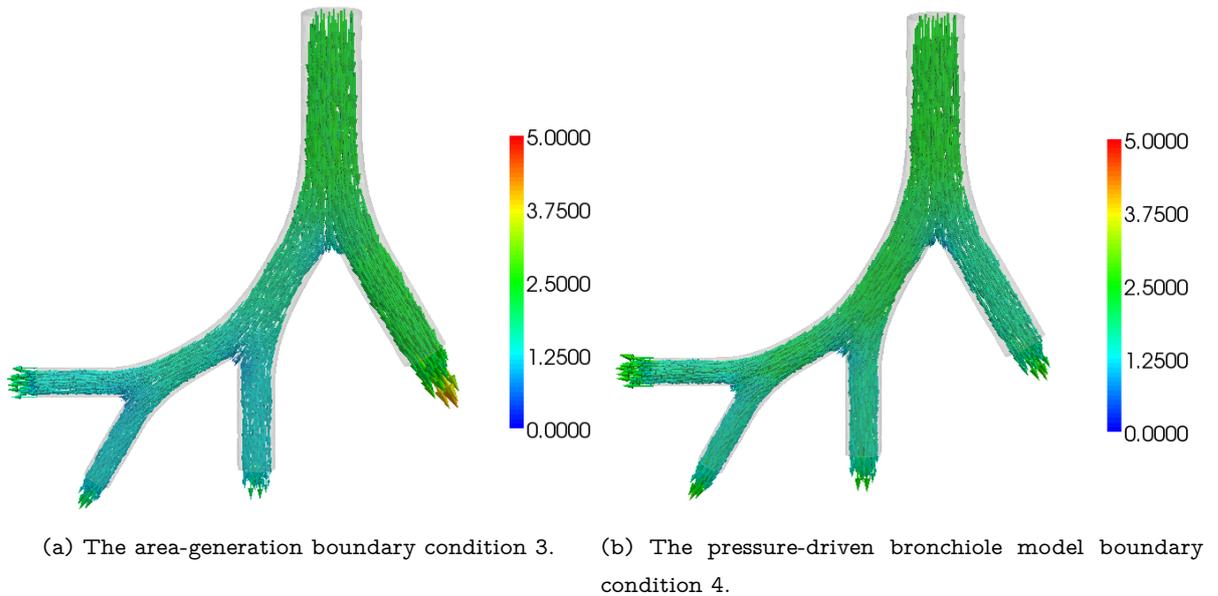


Figure 4.7: Resulting glyphs of the boundary condition tests for the setups 3 and 4 on page 39.

Sensitivity Analysis

Sensitivity Analysis is a very useful and important tool when dealing with new models and uncertain behavior of methods in a new field. Without it, we could only see the effect of *what* is happening, but we would not be able to figure out *why* it happens. One way to calculate the sensitivity would be to run simulations over and over again, every time with different initial conditions, *perturbations* to some defined states, and document the change in the solution. If we get a huge effect, we can deduct that our solution is sensitive to this specific perturbation. However, neither do we get any deeper insight into the structure of sensitive perturbations, nor do we learn, how we can manipulate the model to make it better in some way. Hence what we desire is a way to reverse this view. We want to get the optimal perturbations, let us call them $\tilde{b} = b + \Delta b$, with small Δb , that caused an observed effect J from some parameters p . The derivative

$$J' = \sum_k \frac{\partial J}{\partial b_k} \Delta b_k, \quad (5.1)$$

which is then linear, is what we had before: The effect on the model to specific perturbations Δb . To reverse this, we pick some specific observation J_l and inspect its derivative. This time however to the set of parameters p :

$$J'_l = \sum_k \frac{\partial J_l}{\partial p_k} \Delta p_k. \quad (5.2)$$

The two different derivatives have to be related since the perturbations \tilde{b} depend on the parameters p . And in fact they are related via the adjoint of the model. In a way, the adjoint interchanges the meaning of input and output to the model and therefore gives a structure of the optimal perturbations. We will see this later in this chapter, when we put this into a more stringent mathematical description [24].

In our approach, we use the *Singular Value Decomposition* (SVD). This means we have to solve the forward and the adjoint problem many times to magnify

sensitive and damp non-sensitive perturbations to the linear operator. Thus we find an optimal perturbation and get a better insight into the problem. We can also judge with the help of the SVD how and where perturbations magnify if we filter out regions of interest. The singular values σ_j that we compute quantify the effect of perturbations. In our model, denoted by the operator \mathcal{A} , they are the square roots of the eigenvalues σ_j^2 of the adjoint times the primal problem to the eigenvectors v_j :

$$\mathcal{A}^* \mathcal{A} v_j = \sigma_j^2 v_j \quad (5.3)$$

with A^* the adjoint of A . We see here that we need to build the adjoint operator to the primal problem.

5.1 Theoretical Background

In the following, we define a general optimization problem to derive the formulation of the adjoint operator and demonstrate by means of an abstract nonlinear problem the concept of sensitivity analysis.

5.1.1 A General Optimization Problem

Optimization problems usually come with four main ingredients:

- objective or cost functionals $J(\phi, g)$,
- state equations ϕ ,
- controls g , and
- constraints $F(\phi, g)$.

Then, the general setting is given by:

$$\text{Find control } g \text{ and states } \phi \text{ such that } J(\phi, g) \text{ is minimized subject to } F(\phi, g). \quad (5.4)$$

With the help of the Lagrange multiplier method [34, 97], we can combine the state equation and the cost functional, introducing an additional adjoint variable z :

$$L(\phi, g, z) = J(\phi, g) - z^* F(\phi, g). \quad (5.5)$$

We can interpret this as a new unconstrained optimization problem. We find a necessary condition for an extremum of J if we set the derivative of L with

respect to ϕ to zero, namely

$$\frac{\partial L}{\partial \phi} =: L_\phi = 0. \quad (5.6)$$

Following this approach taking formal derivatives yields the adjoint equation

$$\left(\frac{\partial F}{\partial \phi}\right)^* z = \left(\frac{\partial J}{\partial \phi}\right)^*. \quad (5.7)$$

Here, $(\cdot)^*$ denotes the adjoint. We can easily see that we need the first variation of the state equation $F(\phi, g)$ and its adjoint $F(\phi, g)^*$. The significance of that will be clearer in the aftermath of the next section. If we set the other partial derivatives $\frac{\partial L}{\partial z} =: L_z$ and $\frac{\partial L}{\partial g} =: L_g$ to zero, too, we end up with the full optimality system [38, Ch. 2.3].

5.1.2 An Abstract Nonlinear Problem

Given a parameter-dependent nonlinear PDE, i.e. the discretized Navier-Stokes equations,

$$N(U_h, g) = 0 \quad (5.8)$$

with the discrete solution U_h and some nonlinear objective function

$$J(U_h, g), \quad (5.9)$$

the sensitivity problem reads:

$$\text{Find } \frac{dJ}{dg} \text{ with some controls or parameters } g. \quad (5.10)$$

By the chain rule, this yields

$$\frac{dJ}{dg} = \frac{\partial J}{\partial U_h} \frac{\partial U_h}{\partial g} + \frac{\partial J}{\partial g}. \quad (5.11)$$

If we take the partial derivative of (5.8) with respect to g of the nonlinear problem, we obtain

$$\frac{dN}{dg} = \frac{\partial N}{\partial U_h} \frac{\partial U_h}{\partial g} + \frac{\partial N}{\partial g} = 0. \quad (5.12)$$

This is related to the sensitivity problem via

$$\frac{dJ}{dg} = -\frac{\partial J}{\partial U_h} \left(\frac{\partial N}{\partial U_h}\right)^{-1} \frac{\partial N}{\partial g} + \frac{\partial J}{\partial g} \quad (5.13)$$

under the assumption that the inverse of $\frac{\partial N}{\partial U_h}$ is bounded and with use of the implicit function theorem. Now there exist basically two possibilities to compute the sensitivity $\frac{dJ}{dg}$ with a simple vector product. One can compute

$$\frac{\partial U_h}{\partial g} = -\left(\frac{\partial N}{\partial U_h}\right)^{-1} \frac{\partial N}{\partial g} \quad (5.14)$$

first, which are the so-called forward methods. This corresponds to the solution of the sensitivity of the PDE for each design parameter g and results in evaluating the PDE for every control or parameter, which is very tedious for a big set of controls/parameters.

Or, one can compute

$$\frac{\partial J}{\partial U_h} \left(\frac{\partial N}{\partial U_h} \right)^{-1} \quad (5.15)$$

first, which is the solution of the adjoint PDE. This can be seen by observation of the Lagrange functional

$$\mathcal{L}(U_h, g, z) = J(U_h, g) - z^* N(U_h, g), \quad (5.16)$$

with the Lagrange multiplier $z \in V$. In order to minimize the objective function $J(U_h, g)$ with respect to U_h , a necessary condition is a vanishing U_h -derivative of the Lagrange functional. This results in solving the dual problem (or the adjoint PDE)

$$\frac{d\mathcal{L}}{dU_h} \varphi = \frac{\partial J}{\partial U_h} \varphi - z^* \frac{\partial N}{\partial U_h} \varphi = 0 \quad \forall \varphi \in V. \quad (5.17)$$

A solution z^* for this adjoint problem is (5.15), as described above. Note that we use the first variation of the nonlinear problem $N(U_h, g)$. Hence, if we can neglect all higher order terms, we get a linear system! For further reference see [9, 32, 38, 61, 70].

5.1.3 Continuous or Discrete Adjoint Method

In the literature, i.e. [38, 46], there exist generally two ways of deriving and computing the adjoint:

- the discrete adjoint method, and
- the continuous adjoint method.

Both methods start with the nonlinear PDE, the Navier-Stokes equations in our case. The discrete adjoint method first discretizes the nonlinear equations, linearizes them, and then forms the discrete adjoint, while the continuous method forms the adjoint to the continuous nonlinear problem and linearizes afterwards. Schematically depicted:

- Discrete approach:
 1. Nonlinear PDE
 2. Nonlinear discrete PDE

3. Linear discrete equation
 4. Discrete adjoint equation
- Continuous approach:
 1. Nonlinear PDE
 2. Linearized continuous PDE
 3. Linear continuous adjoint equation
 4. Discrete adjoint equation.

Both methods have their up- and downsides [67]. The discrete approach builds the exact gradient to the discrete objective function, hence the sensitivity converges fully. It is straightforward to implement but the code will be inefficient in the case of many parameters.

The continuous approach highlights the physical significance of adjoint variables. However, problems can exist that have inadmissible boundary conditions to the objective functions in some settings and the sensitivity might not converge fully. The continuous method is nevertheless more natural to derive than the discrete method and numerical software packages like HiFlow³ can treat it efficiently.

5.2 Singular Value Decomposition

The factorization of a matrix A in

$$A = U\Sigma V^T, \quad (5.18)$$

with $A \in \mathbb{R}^{n \times p}$, $U \in \mathbb{R}^{n \times n}$, $\Sigma \in \mathbb{R}^{n \times p}$ and $V \in \mathbb{R}^{p \times p}$, is called *singular value decomposition*. The matrices U and V are orthogonal, which means $U^T U = I_{n \times n}$, $V^T V = I_{p \times p}$, with the n -dimensional identity matrix $I_{n \times n}$. In the following, we state the details that we need here. A more complete overview of the SVD can be found in the book “Matrix Computations” of Gene H. Golub and Charles F. van Loan [36].

The columns of U are called *left singular vectors*, the rows of V^T are the *right singular vectors*. The diagonal entries σ_i of Σ are the so-called *singular values* of A . It is common, to sort the singular values in a descending order

$$\sigma_n \geq \dots \geq \sigma_i \geq \dots \geq \sigma_0 \geq 0, \quad (5.19)$$

so that Σ is uniquely determined. If the matrix A is complex, the above holds with \cdot^T replaced by \cdot^* , the *complex conjugate*.

The SVD relates to the Frobenius and the 2-norm of the matrix A in the following way:

$$\begin{aligned}\|A\|_F^2 &= \sigma_0 + \dots + \sigma_n \\ \|A\|_2 &= \sigma_n.\end{aligned}\tag{5.20}$$

In the beginning of this chapter, we already stated that the singular values are the square roots of the eigenvalues of $A^\top A$. We consider only the real case here, but with the obvious modifications this holds true for the complex case, too. From

$$\begin{aligned}A^\top A &= (U\Sigma V^\top)^\top U\Sigma V^\top \\ &= V\Sigma U^\top U\Sigma V^\top \\ &\stackrel{U \text{ orth.}}{=} V\Sigma^2 V^\top\end{aligned}\tag{5.21}$$

we see that we have to solve the eigenvalue system for the right singular vectors

$$A^\top A v_j = \sigma_j^2 v_j\tag{5.22}$$

and we see that the eigenvalues of (5.22) are exactly the squared singular values. If we reverse $A^\top A$ and use the same reasoning for AA^\top , we get the same expression for the left singular vectors. However, since the action of A and A^\top is usually hard to compute, we will recover the left singular values from the right singular values by the expression

$$U = AV\Sigma^{-1}.\tag{5.23}$$

5.3 Eigenvalue and Eigenvector Solvers

We have seen in the previous section that we need to have an eigenvalue and eigenvector solver to get hold of the singular values and singular vectors. In this section we will state two methods to compute eigenvalues of general square matrices. We have to solve the eigenvalue problem

$$Av = \lambda v,\tag{5.24}$$

with $A \in \mathbb{C}^{n \times n}$, $v \in \mathbb{C}^n \setminus \{0\}$ and $\lambda \in \mathbb{C}$. We interpret both methods in the context of projection methods. Let \mathcal{L}, \mathcal{K} be two subspaces of \mathbb{C}^n with dimension $m \ll n$. We seek an approximation $\tilde{\lambda}$ to λ and $\tilde{v} \in \mathcal{K}$ to v that fulfills the *Petrov-Galerkin condition*

$$A\tilde{v} - \tilde{\lambda}\tilde{v} \perp \mathcal{L}.\tag{5.25}$$

The cases $\mathcal{L} = \mathcal{K}$ are referred to as *orthogonal projection methods* while methods with $\mathcal{L} \neq \mathcal{K}$ are called *oblique projection methods*. We refer to the book of Y. Saad, “Numerical methods for large eigenvalue problems” [77] for further methods and convergence results.

5.3.1 The Power Method

The *Power method* is a commonly used method to approximate extremal eigenvalues and their corresponding eigenvectors. We obtain it by choosing

$$\mathcal{L} = \mathcal{K} = \text{span}\{A^k v\} \quad (5.26)$$

for $k \geq 0$, $k \in \mathbb{N}$ and a diagonalizable matrix $A \in \mathbb{C}^{n \times n}$. The implementation of the method is given in Algorithm 1, see also [36, 73]. The sequence ν_k in

Algorithm 1 Power method.

```

1: function Power( $A, k$ )
2:   Choose initial normalized vector  $v_0 \in \mathbb{C}^n$ 
3:   for  $k = 1, 2, \dots$  do
4:     Compute  $z_k = Av_{k-1}$ 
5:     Compute  $v_k = z_k / \|z_k\|_2$ 
6:     Compute Rayleigh quotient  $\nu_k = v_k^H Av_k$ 
7:   end for
8: end function

```

the Power method converges to the largest eigenvalue λ_1 under the assumption that λ_1 is a *dominant eigenvalue*, i.e. $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$ and that the initial vector v_0 has a component in the direction of the corresponding eigenvector x_1 . For further details see e.g. [36].

5.3.2 The Davidson Method

We obtain more sophisticated projection methods if we use the information given by the matrix A . We can employ this by choosing *Krylov subspaces* for \mathcal{L} and \mathcal{K} .

Definition 5.3.1 (Krylov subspace). Let $A \in \mathbb{C}^{n \times n}$ and $v \in \mathbb{C}^n$.

$$K_m(A, v) := \text{span}\{v, Av, A^2v, \dots, A^{m-1}v\}. \quad (5.27)$$

Krylov subspace methods have the nice property that they yield as projection matrix H_m an upper *Hessenberg matrix* if we choose $\mathcal{L} = \mathcal{K} = K_m(A, v)$.

The eigenvalues of H_m can be computed easily [73]. The resulting method is called *Arnoldi method*. If we further assume A to be symmetric, we get the so-called *Lanczos method*. Through the symmetry, the upper Hessenberg matrix H_m reduces to a *tridiagonal matrix*, which has non-zero entries only on the diagonal and the first upper and lower sub-diagonal. These methods have good convergence properties, if the eigenvalues are well separated [36].

Algorithm 2 Davidson method.

```
1: function Davidson( $A, l, m, \epsilon$ )
2:   Choose initial orthonormal matrix  $Q_1$ 
3:   for  $k = 1, 2, \dots$  do
4:     Compute the projection matrix  $H_k = Q_k^\top A Q_k$ 
5:     Compute  $l$  eigenpairs  $(\lambda_{k,i}, \nu_{k,i})_{0 \leq i \leq l}$  of  $H_k$ 
6:     for  $i = 1, \dots, l$  do
7:       Compute the Ritz vectors  $\psi_{k,i} = Q_k \nu_{k,i}$ 
8:       Compute the residuals  $r_{k,i} = A \psi_{k,i} - \lambda_{k,i} \psi_{k,i}$ 
9:     end for
10:    if  $\|r_k\| \leq \epsilon$  then Exit
11:    end if
12:    for  $i = 1, \dots, l$  do
13:      Compute new directions  $t_{k,i} = C_{k,i} r_{k,i}$ 
14:    end for
15:    if  $\dim(Q_k) \leq m$  then
16:       $Q_{k+1} = mGS(Q_k, t_{k,1}, \dots, t_{k,l})$ 
17:    else
18:       $Q_{k+1} = mGS(\psi_{k,1}, \dots, \psi_{k,l}, t_{k,1}, \dots, t_{k,l})$ 
19:    end if
20:  end for
21: end function
```

We can obtain better separation properties and thus better convergence of the eigenvalue solvers using preconditioners. This is done in the *Davidson method*, which was first developed by E. R. Davidson for real symmetric matrices [18]. It is a version of the Arnoldi method where the preconditions can vary in each iteration, cf. Algorithm 2 [17]. However, it has to be noted that the projection matrix H_m is not a Hessenberg matrix anymore and may even become dense. This is due to the fact that we no longer build a Krylov subspace.

A modified Gram-Schmidt procedure, abbreviated with *mGS* in Algorithm 2

is used as orthogonalization method, see e.g. [73]. The Davidson method has various advantages, it uses for instance only matrix-vector multiplications, which can be efficiently parallelized [83]. Furthermore, the preconditioning matrix $C_{k,i}$ of the form M_i^{-1} is not needed explicitly, instead one can solve iteratively

$$M_i t_{k,i} = r_{k,i}, \quad (5.28)$$

where M_i is an approximation of $(A - \lambda_i I)$. Note also that the projection matrix $H_k = Q_k^\top A Q_k$ needs not to be computed completely in each step, instead it is sufficient to update the last l rows and columns in each iteration k .

5.4 Numerical Methods

At the beginning of this chapter we discussed the difference between the forward sensitivity analysis and the adjoint sensitivity analysis: The forward sensitivity analysis shows the development of initial perturbations of input variables but needs to be differentiated and evaluated for every parameter, while the adjoint sensitivity analysis simulates the development of e.g. an anomaly backwards in time, hence finds the source of this anomaly and can be efficiently computed for a large number of parameters. In the sequel, we will focus only on the adjoint sensitivity analysis. It provides a mathematical framework that we can treat efficiently modifying only small portions of the existing implementation of the Finite Element methods for the incompressible Navier-Stokes equations. Using this, we can learn more about the effect of coupled models, like those introduced in Section 4, on the fluid domain.

5.4.1 Linearized Navier-Stokes and Adjoint Formulation

The adjoint sensitivity approach entails the linearization of the Navier-Stokes equations, because the concept of an adjoint exists only in the linear world. The adjoint to the Navier-Stokes equations is therefore always the adjoint to the linearized Navier-Stokes equations. We want to find a small optimal perturbation that has the largest possible effect on the solution, therefore we need to find the singular vectors and singular values of the Navier-Stokes system. We can compute the singular values as the square roots of the eigenvalues of the adjoint system times the forward system as discussed in Section 5.2.

As prerequisite, we need to have the base flow and pressure $[U, P]$ of the nonlinear system (2.21) precomputed. We use the boundary conditions from

the pressure-driven bronchiole model that we discussed in Section 4.2.1 at the outlets Γ_{out_i}

$$\tilde{u}_i = U_i(A_i, P_i, R_{total_i}) \text{ on } \Gamma_{out_i}, \quad (5.29)$$

because we want to estimate their sensitivity onto the fluid flow domain.

Linearized Navier-Stokes Equations

First, we derive the linearized forward equations. We want to linearize around the precomputed solution $[U, P]$ from (2.21). We obtain this system by inserting a perturbation $u = U + \epsilon \tilde{u}$ and $p = P + \epsilon \tilde{p}$ with the base flow $[U, P]$ into the nonlinear equations (2.21):

$$\begin{cases} \rho \frac{\partial(U + \epsilon \tilde{u})}{\partial t} + \rho((U + \epsilon \tilde{u}) \cdot \nabla)(U + \epsilon \tilde{u}) - \mu \Delta(U + \epsilon \tilde{u}) + \nabla(P + \epsilon \tilde{p}) = 0 & \text{in } \Omega \\ \nabla \cdot (U + \epsilon \tilde{u}) = 0 & \text{in } \Omega. \end{cases} \quad (5.30)$$

Discarding the volume forces b and keeping only the linear terms in ϵ , this yields the linearized Navier-Stokes equations, we will call it the forward system in what follows:

$$\begin{cases} \rho \frac{\partial \tilde{u}}{\partial t} + \rho \left((U \cdot \nabla) \tilde{u} + (\nabla U)^\top \cdot \tilde{u} \right) - \mu \Delta \tilde{u} + \nabla \tilde{p} = 0 & \text{in } \Omega \\ \nabla \cdot \tilde{u} = 0 & \text{in } \Omega. \end{cases} \quad (5.31)$$

with additional initial and boundary conditions [63]. In our setup, we use a random vector field as initial condition. It is however useful to ensure solenoidality already in the initial solution, so that all the computed Ritz vectors can be used [4].

We use do-nothing boundary conditions

$$\frac{\partial \tilde{u}}{\partial n} - \tilde{p} n = P_{in,out} \text{ on } \Gamma_{in,out}, \quad (5.32)$$

at the inlets Γ_{in} and the outlets Γ_{out} with a mean pressure from the nonlinear solution [6]

$$P_{in,out} = |\Gamma_{in,out}|^{-1} \int_{\Gamma_{in,out}} P do \quad (5.33)$$

and no-slip boundary conditions at the walls Γ_{wall}

$$\tilde{u} = 0 \text{ on } \Gamma_{wall}. \quad (5.34)$$

We want to write this for later use in operator form. Let us denote the action of the forward system (5.31) by the evolution operator

$$\mathcal{A} : \mathcal{W} \rightarrow \mathcal{W}, \quad \mathcal{W} := \mathcal{W}_u \times \mathcal{W}_p. \quad (5.35)$$

In the instationary case, we usually have $\mathcal{W}_u = \mathcal{L}^2(0, T; \mathcal{H}^1(\Omega))$ for the velocity part and $\mathcal{W}_p = \mathcal{L}^2(0, T; \mathcal{L}_0^2(\Omega))$ for the pressure part. This means that \mathcal{A} evolves some state, or discretized vector field in the finite dimensional case, $s := [\tilde{u}, \tilde{p}] \in \mathcal{W}$ forward in time. Hence we can write

$$\mathcal{A}(t')s(t) = s(t + t'). \quad (5.36)$$

Adjoint Navier-Stokes Equations

We derive the adjoint Navier-Stokes system, we will call it adjoint system hereinafter, either with the Lagrange multiplier method, by setting the derivative with respect to the state variables to zero, see Equation (5.17), or by the relation¹

$$(\mathcal{A}s, s^*) - (s, \mathcal{A}^*s^*) = 0, \quad (5.37)$$

where (\cdot, \cdot) denotes the inner product and \cdot^* the adjoint. The operator \mathcal{A}^* will be explicitly stated in short form later on.

The (formal) adjoint is then given by

$$\begin{cases} -\rho \frac{\partial z}{\partial t} + \rho \left((U \cdot \nabla)z - (\nabla U^\top) \cdot z \right) - \mu \Delta z + \nabla q = 0 & \text{in } \Omega \\ \nabla \cdot z = 0 & \text{in } \Omega, \end{cases} \quad (5.38)$$

where $[z, q]$ are the adjoint variables to $[\tilde{u}, \tilde{p}]$ from (5.31). The treatment of the boundary conditions for the considered problem is given in Section 5.4.3.

We write the adjoint equation also in operator form. However, the action of the adjoint system evolves the state backward in time, such that we can investigate the cause of an effect. Similar to (5.36) we write

$$\mathcal{A}^*(t')s^*(t) = s^*(t - t'), \quad (5.39)$$

with $s^* := [z, q] \in \mathcal{W}^*$, where \mathcal{W}^* is the dual space to \mathcal{W} .

We want to obtain the effect of perturbations that arise in models of the non-resolvable lower part of the lungs. Therefore, we use the boundary conditions for the adjoint that the forward system dictates. This means that we apply do-nothing conditions with the mean pressure

$$P_{in,out} = |\Gamma_{in,out}|^{-1} \int_{\Gamma_{in,out}} P do \quad (5.40)$$

from the nonlinear simulation at the inlets Γ_{in} and the outlets Γ_{out} . They are consistent with the model for the lower part, because they assume Poiseuille flow, the same assumption we have in the models for the lower part.

¹The time variable t will be omitted where it is not explicitly needed.

Recalling that we need to satisfy condition (5.37), the do-nothing conditions from the forward system translate in the adjoint system to

$$(U \cdot n)z + \frac{\partial z}{\partial n} - qn = P_{in,out} \text{ on } \Gamma_{in,out}, \quad (5.41)$$

see [4] for justification.

5.4.2 Computation of the SVD

The SVD is used to quantify the sensitivity of perturbations to the linearized Navier-Stokes system \mathcal{A} through the largest singular values. To compute the SVD, we need to find the eigenvalues of $\mathcal{A}^* \mathcal{A}$, see 5.2. Because we cannot explicitly determine the operators \mathcal{A} and \mathcal{A}^* , we have to solve the forward and adjoint system to get the resulting state v . We apply the operators subsequently: First we solve the forward system to get an intermediate state $w = \mathcal{A}u$, then we apply the adjoint operator which yields $v = \mathcal{A}^* w = \mathcal{A}^* \mathcal{A}u$ [4].

The beauty of the singular vectors in the field of fluid dynamics is that they have a physical meaning: The right singular vectors describe the structure of the “optimal” perturbations while the left singular vectors give a handle on the outcome, the “worst case disturbance” of the flow. The singular values, which relate the singular vectors to one another, are the amplification factors of the respective perturbations and determine also the maximum energy growth G . Let (u, v) denote the inner product and let us define a perturbation energy norm with

$$\|v(\tau)\|^2 = (v(\tau), v(\tau)) = (\mathcal{A}v(0), \mathcal{A}v(0)) = (v(0), \mathcal{A}^* \mathcal{A}v(0)). \quad (5.42)$$

Let us recall that we have to solve the eigenvalue problem

$$\mathcal{A}^* \mathcal{A}v_j = \sigma_j^2 v_j \quad (5.43)$$

to get the singular values σ_j . With the adjoint \mathcal{A}^* to \mathcal{A} with respect to the energy inner product and the maximum energy growth G that is obtained by maximizing the ratio of the final outcome to the initial perturbation, we obtain

$$G = \max_{v(0) \neq 0} \frac{\|v(\tau)\|}{\|v(0)\|}, \quad (5.44)$$

which turns out to be the largest singular value σ_n since

$$\frac{\|v(\tau)\|}{\|v(0)\|} \leq \frac{\|v(n)\|}{\|v(0)\|} = \frac{\sqrt{(v(0), \mathcal{A}^* \mathcal{A}v(0))}}{\sqrt{(v(0), v(0))}} = \sigma_n. \quad (5.45)$$

See [69] for further discussions.

Algorithm

We implemented different possibilities to compute the SVD. We want to present the algorithm that uses the Davidson method as eigenvalue solver, because we can determine $l > 1$ eigenvalues without a deflation step [73, Sec. 5.6.1] at the same time. We also implemented an algorithm that uses the Arnoldi method and an algorithm that uses the Power method.

Algorithm: Set $M = \mathcal{A}^* \mathcal{A}$, $Mu_k = S_k$.

1. Initialize:

Set parameters

l , the number of desired singular values,

m , the maximum basis size, whereas $m \geq l$,

ϵ , the tolerance for the convergence of singular values,

u_0 , an initial vector, either from white noise or from one run through the linearized forward and adjoint system.

Also initialize the needed structures for the computation.

2. Get first l approximate eigenvalues and eigenvectors of M . We use the Davidson method here. Note that we have to solve the forward and adjoint equations to compute the new direction $t_{k,i}$ in line 13 of Algorithm 2. The eigenvalues λ_j and Ritz vectors ψ_j of the (dense) projection matrix H_k are calculated with LAPACK [2]. The implemented code can be seen in Appendix B on page 91. Note that we project the Ritz vectors ψ_j and the eigenvalues λ_j in the Davidson method back to the full space. In case of convergence, they are the right singular vectors u_j and the singular values

$$\sigma_j = \sqrt{\lambda_j}. \quad (5.46)$$

3. Calculate left singular vectors u_j from the right singular vectors v_j and the singular values σ_j for $j \leq l$ as described at the end of Section 5.2.

5.4.3 Sensitivity Analysis with the SVD

In the remains of this chapter, we want to present and discuss the results of the SVD in a two-dimensional lung geometry with four generations, see Figure 5.1.

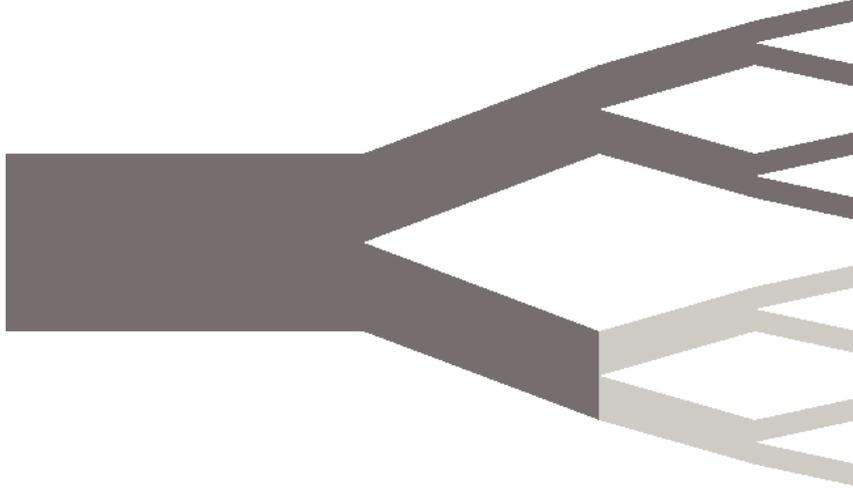


Figure 5.1: Geometry of a schematic two-dimensional lung with four generations. Simulations were run for the symmetric (the whole geometry) and the asymmetric case (the dark gray part of the geometry without the light gray part).

As governing equations for the base flow we use the instationary incompressible Navier-Stokes equations with the boundary conditions

$$U_i := U_i(A_i, P_i, R_{total_i}) = \frac{2}{A} \frac{P_i}{R_{total_i}} \quad (5.47)$$

computed by the pressure-driven bronchiole model from Section 4.2.

$$\left\{ \begin{array}{ll} \rho \frac{du}{dt} + \rho(u \cdot \nabla)u - \mu \Delta u + \nabla p = 0 & \text{in } \Omega \times (0, T), \\ \nabla \cdot u = 0 & \text{in } \Omega \times (0, T), \\ u = 0 & \text{on } \Gamma_{Wall} \times (0, T), \\ u = U_i(A_i, P_i, R_{total_i}) & \text{on } \Gamma_i \times (0, T), \\ & i = 1, \dots, N, \\ \partial_n u - pn = 0 & \text{on } \Gamma_0 \times (0, T), \\ u(\cdot, 0) = u_0 & \text{in } \Omega. \end{array} \right. \quad (5.48)$$

In the stationary case, there exists only one stationary base flow U_0 that is used throughout the computation. For the instationary case, we save the velocity field U_t of the nonlinear simulation for each time-step. Then the forward system

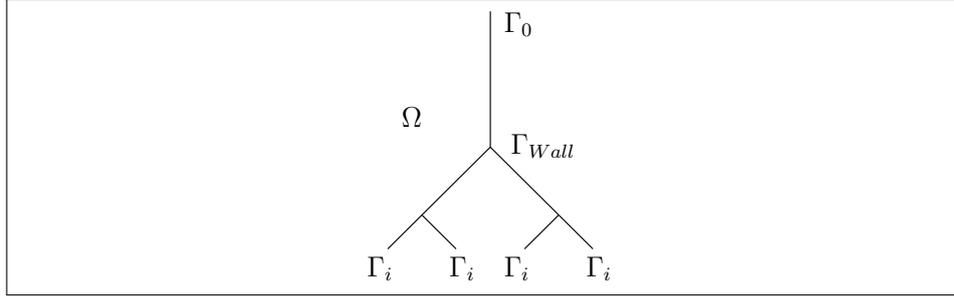


Figure 5.2: Scheme of the boundary conditions at the resolvable part of the geometry.

$$\left\{ \begin{array}{ll} \rho \frac{\partial \tilde{u}}{\partial t} + \rho \left((U_t \cdot \nabla) \tilde{u} + (\nabla U_t)^\top \cdot \tilde{u} \right) - \mu \Delta \tilde{u} + \nabla \tilde{p} = 0 & \text{in } \Omega \times (0, T), \\ \nabla \cdot \tilde{u} = 0 & \text{in } \Omega \times (0, T), \\ \tilde{u} = 0 & \text{on } \Gamma_{Wall} \times (0, T), \\ \partial_n \tilde{u} - \tilde{p} n = P_{\{0,i\}} & \text{on } \Gamma_{\{0,i\}} \times (0, T), \\ \tilde{u}(\cdot, 0) = u_0 & \text{in } \Omega, \end{array} \right. \quad (5.49)$$

with the random vector u_0 and the adjoint system

$$\left\{ \begin{array}{ll} -\rho \frac{\partial z}{\partial t} + \rho \left((U_t \cdot \nabla) z - (\nabla U_t^\top) \cdot z \right) - \mu \Delta z + \nabla q = 0 & \text{in } \Omega \times (T, 0), \\ \nabla \cdot z = 0 & \text{in } \Omega \times (T, 0), \\ z = 0 & \text{on } \Gamma_{Wall} \times (T, 0), \\ (U_t \cdot n) z + \frac{\partial z}{\partial n} - q n = P_{\{0,i\}} & \text{on } \Gamma_{\{0,i\}} \times (T, 0), \\ \tilde{z}(\cdot, T) = \tilde{u}_T & \text{in } \Omega \end{array} \right. \quad (5.50)$$

is computed. The initial condition in the adjoint system corresponds to the final solution of the forward system \tilde{u}_T .

We run this simulation for the symmetric and asymmetric case (see Figure 5.1). The largest singular value is of the order $\sigma_n \approx 10^6$, while the second largest value is only of the order 10^3 . The other singular values decay even more rapidly. They depend however on the exact boundary conditions and physical constants that we use. In the following figures 5.3 and 5.4 we have illustrated the structures of the right and left singular values corresponding to the largest singular values both in the symmetric and asymmetric case. The pressure-driven bronchiole model is applied at the right outlets, do-nothing conditions at the inlets and no-slip conditions at the wall for the nonlinear simulation. In the

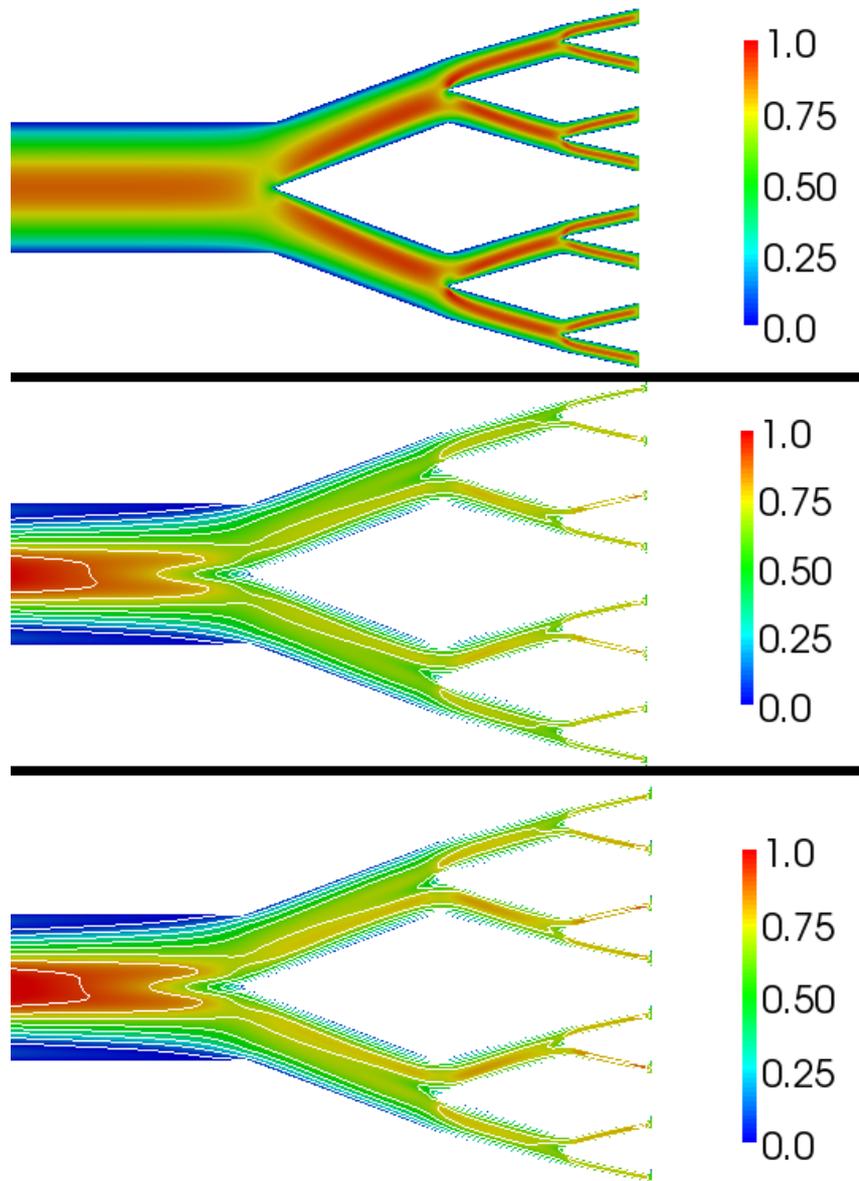


Figure 5.3: The nonlinear velocity field (top), the right (middle) and left (bottom) singular vector fields in a symmetric 2D lung with four generations.

forward and adjoint system, we apply do-nothing conditions at both ends, and no-slip conditions at the wall. The vector fields are normalized already in the SVD, but then scaled so that the values lie within the interval $[0, 1]$.

We can see that in the symmetric case, the right singular vector, which is illustrated in the middle has the same optimal perturbation structure in both branches throughout the whole geometry. Small differences are due to the non-symmetric meshing of the geometry. The left singular value which depicts the

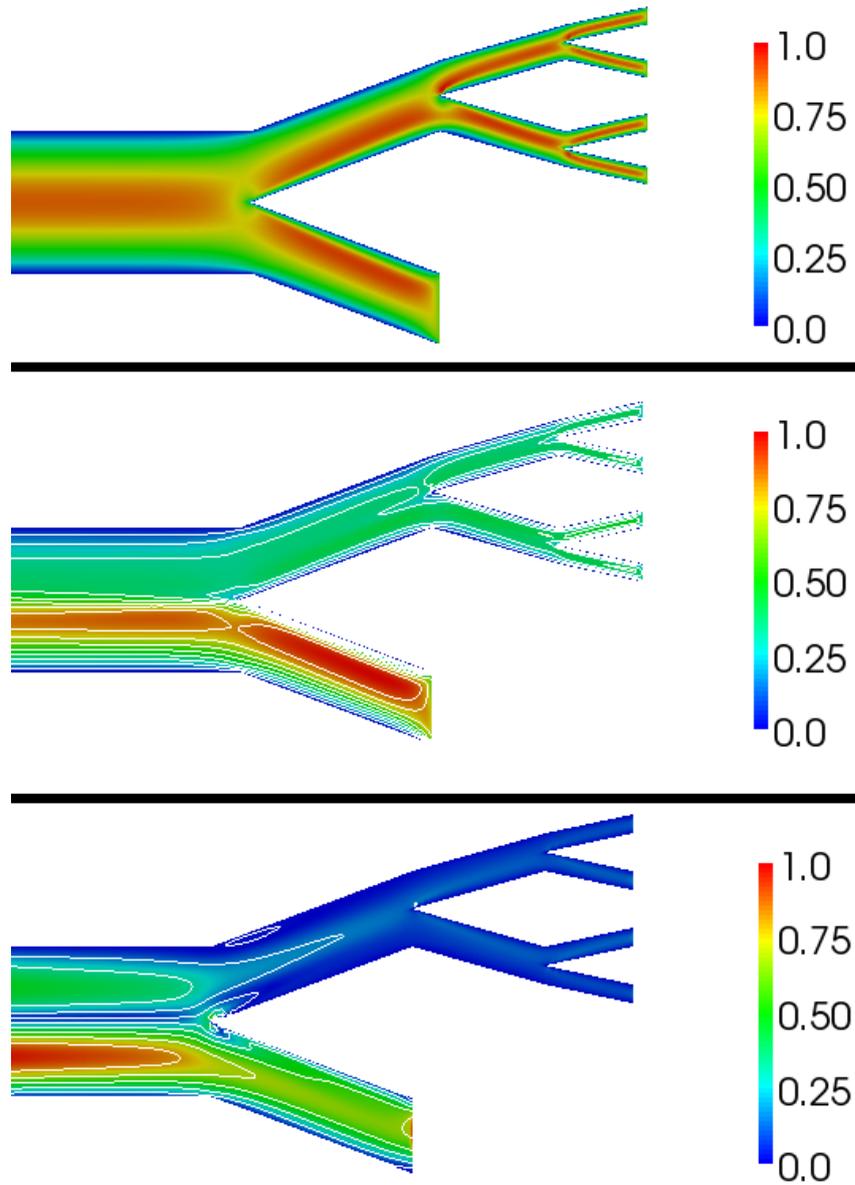


Figure 5.4: The nonlinear velocity field (top), the right (middle) and left (bottom) singular vector fields in an asymmetric 2D lung with four (two) generations.

optimal disturbance outcome has also the same structure. We conclude here that in the symmetric case, the boundary conditions, as long as symmetric themselves, play a minor role, compared to the region where the flow is divided into branches. A better mesh refinement is therefore proposed at the transition regions of the geometry.

In the asymmetric case, we receive a different impression. The transition region is still important, but not as important as the boundary condition on the

shorter of the two branches. The model that we apply on that short branch, needs to account for the asymmetry to balance the flow. In contrast to the flow distribution of 60% to 40% for the numerical experiments in Section 4.3, we want here actually an equally distributed flow, because the radii of the two branches are equal. This leads to the conclusion that the boundary condition has high sensitivity and therefore needs to be chosen carefully.

Filtering a Region of Interest

In many applications of the sensitivity analysis, one is not interested in the sensitivity of the whole computation domain rather than a specific part of it. To get the sensitivity only in the places where we want to know about it, we filter out the other parts of the domain. We do this directly in the SVD. In the forward direction (cf. Section 5.4.1) we keep only the values we are interested in and set the rest of the initial vector field to zero. This is the region, where we have perturbations whose effect is important to us. Application of the forward operator yields the perturbed system. Here, we filter again keeping only the values for which we want to know the maximum disturbance from the initial perturbation regions. Then we apply the adjoint operator (cf. Section 5.4.1) on the filtered vector field.

This algorithm is repeated throughout the SVD loop, see Scheme 5.5, each time applying the filter after application of the forward and the adjoint operator. In our case, we can therefore restrict the permitted perturbations to the lower outlets of the domain, which represents a model of the lower lungs.

The filtering is especially useful in combination with instationary fluid flow simulations. Here, we can specify one region at initial time and another region at final time and see the effect of perturbations in a specific region.

Model Reduction with the SVD

The SVD can also be used to analyze and then reduce models. The decay of the singular values determines the sensitivity of the parameters in the models. If we build a new basis with the singular vectors, we can choose only those, whose correspondent singular values are large and discard the others, whose information is not sensitive to the simulation. This is also called principal component analysis and possible to do with the framework that we introduced.

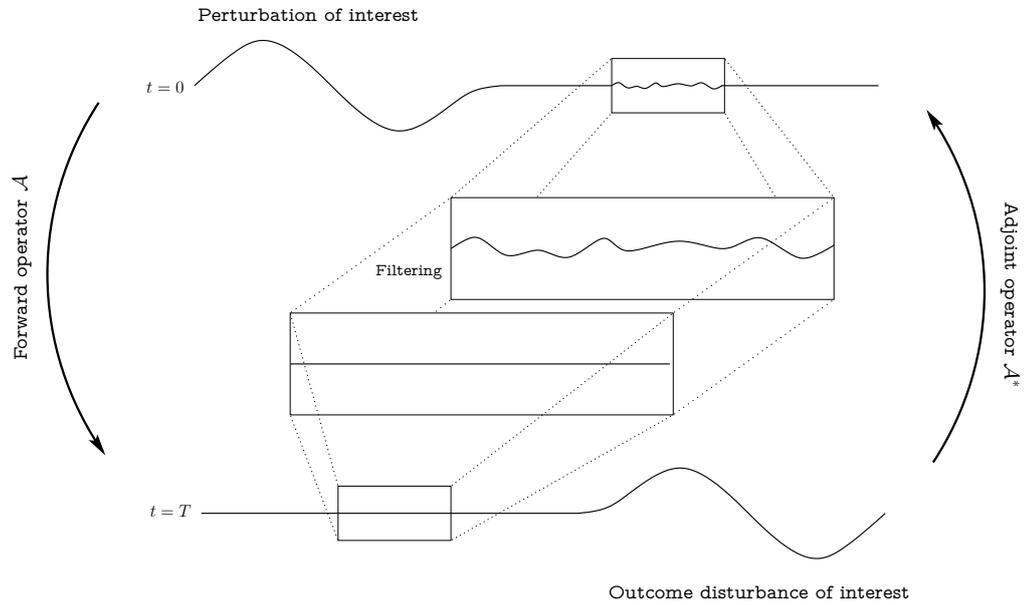


Figure 5.5: Scheme of the filtering after application of the forward and after application of the adjoint operator after each time-step t .

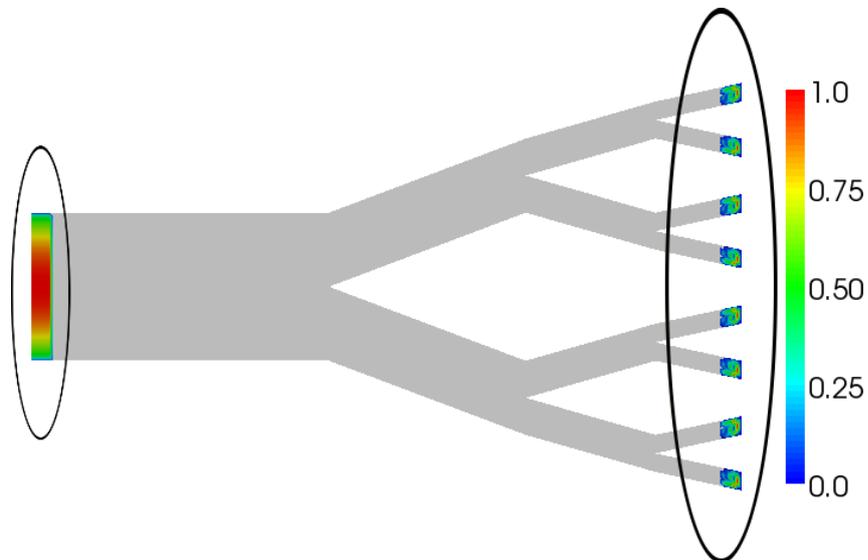


Figure 5.6: The right singular vector, the optimal perturbation, at initial time $t = 0$ encircled at the right hand side of the geometry and the left singular vector, the outcome disturbance, encircled on the left hand side at final time $t = T$. The gray part of the geometry was filtered, hence set to zero after each iteration.

Particle Deposition in the Human Lungs

The study of particle deposition patterns in the human lungs yields insight into a vast field of applications which are of great social importance. The ability to predict the impact of fine dust pollution is of high interest in environmental medicine. We can optimize the efficacy of drugs administered through the lungs by improving the design of spray nozzles. This is only possible but all the more very important for pharmaceutical companies, if the dose that actually reaches the alveolar level and can pass into the blood can be predicted accurately. Studies with inhalable insulin had to be ceased due to inexact prediction of the delivered dose, which is crucial in diabetes treatment [47, 80]. Knowledge of the type and size of particles that are trapped at the mucus layers in the lungs can help to advance insight in cancer-causing effects and help to reduce corresponding risks.

In this chapter, we want to describe patient-specific, statistical and probabilistic methods that allow statements on particle deposition in the human lungs. We are interested in

$$E = 1 - C, \tag{6.1}$$

the escape rate of particles through the lungs. The capture rate C is the ratio of the number of particles P captured at the inner surface to the total number of injected particles at the entrance:

$$C = \frac{\#(P_{\text{captured}})}{\#(P_{\text{total}})}. \tag{6.2}$$

It can be computed either in a global sense or very well-resolved in terms of deposition patterns. However, up to now, this kind of survey has always been specific to one patient and is pushed towards more detailed geometries from better CT scans of patients to get more accurate studies.

The questions: “Can we learn something from these results for other patients?” and “Is there a generality in the deposition pattern?” rise naturally. To tackle this problem, we want to estimate the impact of patient-specific features that emerge due to the geometry of the human lungs as opposed to effects that hold for all particle depositions. Therefore, we extract very few and general parameters from a CT cast of the lungs and build a schematic tracheobronchial tree (cf. Section 6.2). Afterwards, we simulate the particle flow in both geometries (patient-specific and schematic) to achieve the proposed comparison. We present an analytical model introduced by Filoche et al. in [19] (cf. Section 6.3), which enables us to predict the particle depositions in the lungs even simpler. Accordingly, we get generic results that apply to most humans and not only individuals. To quantify their accuracy, we compare these results to both patient-specific and schematic bronchial trees (cf. Section 6.2.2). We start this chapter by introducing models, equations and key features of particle motion in fluids.

6.1 Particle Flow

A wide range of approaches on the topic of particle simulation in fluids exist in literature. Depending on the nature of the particles, different ways of coupling between the fluid and the particles are required.

The one-way coupling approach only models forces in one direction, from the fluid onto the particles. This means that there is no feedback from the particles to the flow. The approach is useful when simulating spherical micron particles. The drag acting on the particles is the dominant force [55] if the particle-to-air density ratio is large and dilute particle suspensions are considered.

Two-way couplings also take the feedback of the particles onto the flow into account. This results in more accurate simulations in the case of larger, non-spherical and unevenly shaped particles. See for example Chapter 8 in “Numerical Methods for Fluids” by Roland Glowinski [35] for the numerical treatment of such particulate flows and Chapter 2 of the book “Particle Size Measurements: Fundamentals, Practice, Quality” by Henk G. Merkus [65] for the effects stemming from particle shapes. However, it is not possible to treat particles in a Lagrangian frame of reference in terms of a post-processing step, since the movement of the particles influences the fluid flow. It is almost needless to say that it requires a lot more computational power due to the coupling between particles and fluid. If the particles also interact with each other, a three-way

coupling is needed.

Forces on the Particles

Several forces, some stronger like aerodynamic and gravitational forces, some weaker, e.g. electromagnetic and diffusional forces, act on particles in a fluid. We try to handle them separately and decompose them linearly. This is a simplification and not necessarily robust since some forces depend on each other in a nonlinear fashion. Usually, these interactions have a very small influence, they are not very well understood and therefore neglected [62]. Then, each of the forces has an additive contribution so we can write them as the sum

$$F = \sum_i F_i. \quad (6.3)$$

We want to describe the Stokes drag force F_{drag} , the Basset history force F_B , the virtual mass effect F_{vm} as well as the gravitational and buoyancy force F_g . For other forces like Brownian force or surface forces due to thermal gradients we refer to [62].

Drag Force: The drag force is the essential force in our setting. It is given by

$$F_{drag} = \frac{1}{2} c \rho_F A_P (u_F - u_P)^2, \quad (6.4)$$

with the air density ρ_F , the projected particle area normal to the bulk flow A_P , the particle drag coefficient c and the velocity of the particle u_P and the fluid u_F . The particle drag coefficient is determined empirically [15, 16] to

$$c = \begin{cases} \frac{24}{Re_P} & \text{for } 0 < Re_P \leq 1.0 \\ \frac{24}{Re_P^{0.646}} & \text{for } 1.0 < Re_P \leq 400 \end{cases} \quad (6.5)$$

with

$$Re_P = \rho_F |u_F - u_P| d_P / \mu_F, \quad (6.6)$$

where μ_F denotes the fluid viscosity and d_P the diameter of the particle. However, c can also depend nonlinearly on the relative velocity $|u_F - u_P|$.

The drag is in its nature a resistance force, therefore the direction of F_{drag} is opposite to the direction of the particle motion.

Basset History Force: The Basset history force

$$F_B = \frac{3}{2} d_P^2 \sqrt{\pi \rho_F \mu_F} \int_0^t \frac{1}{\sqrt{t-t'}} \frac{d}{dt} (u_F - u_P) dt' \quad 0 < t' < t \quad (6.7)$$

with the particle diameter d_P and the fluid viscosity μ_F acts on the particle due to deviations in the flow pattern from the steady state. It can be interpreted as resistance force that has an effect due to previous changes of the fluid field. If for example shocks appear in the fluid, the Basset history force can become large [86].

Virtual Mass Effect: If a particle accelerates, the fluid surrounding this particle accelerates with it. This effect can be described in terms of an additional inertial mass m_v . Since this mass is virtual, the force is called virtual mass effect. The mass m_v is proportional to the ratio of the density of the fluid to the density of the particle

$$m_v \propto \frac{\rho_F}{\rho_P}. \quad (6.8)$$

The resulting force is

$$F_{vm} = m_P c_{vm} \frac{\rho_F}{\rho_P} (u_F - u_P) \quad (6.9)$$

with the proportional factor c_{vm} . This coefficient depends on the shape of the particles, because it contains lift and rotational effects. In the case of solid spherical particles its value is $c_{vm} = \frac{1}{2}$ [62].

Gravitational and Buoyancy Force: The gravitational and the buoyancy force oppose each other. The gravitational force accelerates a particle in direction of gravity. This results in displacement of the surrounding fluid which results in the buoyancy force. Together they are expressed as

$$F_g = m_P \left(1 - \frac{\rho_F}{\rho_P} \right) g. \quad (6.10)$$

Micron Particle Flow

We consider the case of micron particle flow through the human lungs. Micron particle flow means that the diameter d_P of the particles is larger than $1\mu m$ and typically below $100\mu m$. Here, the one-way coupling approach is sufficient, because the particles are small enough to neglect their recoil on the fluid. In our case, all forces on the particles except for the drag force are small enough so that we can neglect them, too. This is because we are dealing with gas-solid flows, where the ratio of the densities of the fluid to the particle ρ_F/ρ_P is very small and thus their contribution almost vanishes. The effect of gravity in the lungs is considered to cancel out in the mean [19] because the direction of the flow of the particles changes from bifurcation to bifurcation. Neglecting all these forces

seems to be a very rough approach, but if we recall our aim to compare a patient-specific tracheobronchial tree to a schematic tree, this is perfectly reasonable, since we are interested in the big picture, which is the overall behavior, rather than small errors.

One possibility to simulate micron particle flow is to precompute the underlying fluid flow and follow the particle traces in a separate post-step. We write down the position of the particles in terms of 3-dimensional Cartesian coordinates $(x(t), y(t), z(t))$. This yields the position vector

$$s(t) = x(t)i + y(t)j + z(t)k, \quad (6.11)$$

with the unit vectors

$$i = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad j = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \quad k = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \quad (6.12)$$

Differentiating the position vector twice with respect to the time t yields the first derivative, the velocity

$$u(t) = \frac{dx}{dt}i + \frac{dy}{dt}j + \frac{dz}{dt}k \quad (6.13)$$

and the second derivative, the acceleration

$$a(t) = \frac{d^2x}{dt^2}i + \frac{d^2y}{dt^2}j + \frac{d^2z}{dt^2}k \quad (6.14)$$

of each particle. Then we can derive a system of *ordinary differential equations* (ODEs) describing the movement of the particles.

Equation of Motion

When we combine the drag force in its vector form with Newton's second law of motion

$$F = ma \quad (6.15)$$

we obtain

$$\begin{aligned} -F_{drag} &= ma(t) \Leftrightarrow \\ -\frac{1}{2} \frac{c\rho_F A_P}{m} (u_F - \frac{dx}{dt}i + \frac{dy}{dt}j + \frac{dz}{dt}k)^2 &= \frac{d^2x}{dt^2}i + \frac{d^2y}{dt^2}j + \frac{d^2z}{dt^2}k. \end{aligned} \quad (6.16)$$

This is a system of equations for the directions x , y and z and can be solved if we insert the initial conditions

$$\begin{aligned}x &= X_0, \frac{dx}{dt} = u_{X_0}, \\y &= Y_0, \frac{dy}{dt} = u_{Y_0}, \\z &= Z_0, \frac{dz}{dt} = u_{Z_0}.\end{aligned}\tag{6.17}$$

X_0 , Y_0 and Z_0 are the initial particle positions and u_{X_0} , u_{Y_0} and u_{Z_0} the initial velocities of the underlying fluid field at the trachea inlet.

The resulting equation of motion in a shorter and more commonly used form for each particle reads

$$\begin{cases} m \frac{du_P}{dt} &= F_{drag} \\ u_P(t_0) &= u_F(s(t_0)), \end{cases}\tag{6.18}$$

where $s(t_0)$ is the initial position of the particle and u_P the velocity of the particle.

Stokes number

The Stokes number St is the key parameter of aerosols in a fluid. It is the ratio of the response time of a particle τ_P to the time characteristic of the flow system τ_f :

$$St = \frac{\tau_P}{\tau_f} = \frac{\rho_P d_P^2 U}{18 \mu_F D},\tag{6.19}$$

with the particle density ρ_P , the particle diameter d_P , the characteristic velocity of the underlying fluid field U , the viscosity μ_F of the fluid and the characteristic system length D . The particle response time describes how fast the particle responds to a change in the underlying velocity field. If the Stokes number is small (≤ 0.01), e.g. dust particles in air, the particles in the fluid have enough time to adjust to changes in the velocity field, hence the particle flow field is very similar to the fluid flow field. In contrary, if the Stokes number is large (≥ 10), e.g. falling raindrops, the particles have almost no time to respond to the variations in the velocity field. Thus the particle velocity remains approximately at its initial value.

6.2 Patient-Specific vs. Schematic Bronchial Tree

We want to compare simulations in patient-specific lung geometries acquired directly from CT scans to schematic lung geometries. This is done for different

reasons:

- The complexity of the human tracheobronchial tree (cf. Figure 6.1) is very high. Simulations with this geometry are therefore hard to compute.
- We want to obtain general, non patient-specific results.
- We want to build a hierarchy in our results that reflects the impact of the generic geometry, the impact of specific values of some main parameters and the impact of local details that are specific to the respective patient.

To achieve this, we extract relevant parameters from the specific geometry that describe the schematics. This process is accomplished in several steps that we explain in Section 6.2.1. Parameters that turned out to be of great impor-

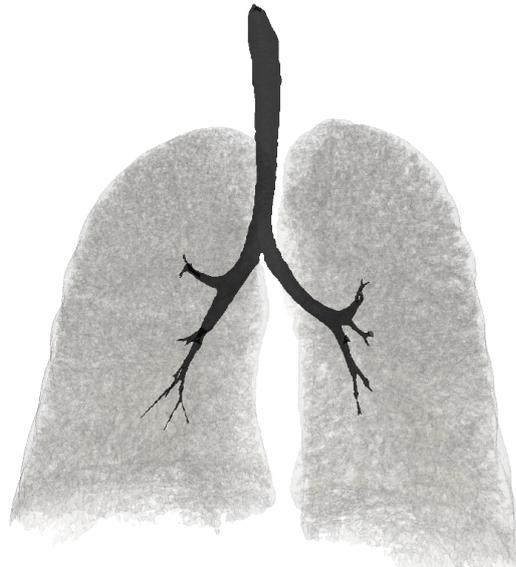


Figure 6.1: The segmented patient specific geometry (black) of the trachea up to a maximum of seven generations.

tance (cf. Figure 6.2) are

α , which is the torque between bifurcations,

θ , which is the angle between the children tubes of one bifurcation, and

h , which is the ratio of diameters of subsequent tubes [19].

If we choose $\alpha = 90^\circ$, $\theta = 60^\circ$ and $h = 2^{-1/3}$ we obtain the canonical tree, which corresponds to the tree described in Weibel's "A" model [90].

In the following, we will describe how to acquire these parameters from a scan of the human respiratory tract and build a schematic tree with these parameters

to compare the escape rates of the respective geometries with respect to the Stokes number of the particles.

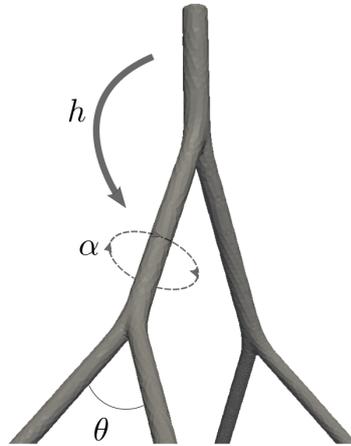


Figure 6.2: Three bifurcations with the parameters h, θ and α , whereas h is the ratio of diameters of subsequent tubes, α is the torque between subsequent bifurcations and θ is the angle between the children tubes of one bifurcation.

6.2.1 Methods

Extracting a Schematic Tree

To create a geometric bifurcation model, h, θ and α (cf. Figure 6.2) need to be extracted from the geometry segmented from a CT scan (cf. Figure 6.1) [31, 57]. The *Vascular Modeling Toolkit* (VMTK) [32], *Paraview* [40] and several scripts were used to automate some steps of this technical process.

Determine h with VMTK: VMTK is able to extract centerlines (cf. Figure 6.3) with the command

```
vmtkcenterlines -ifile filename_in -ofile filename_out
```

of a lung cast. However, this process is not perfectly stable. In this particular case it was necessary to compute the centerlines in a two-step process: First we computed the left part and afterwards the right part of the segmented tracheo-bronchial tree. In combination with the centerlines, we also use the maximum diameter of each bronchiole, which is exported by VMTK.

The centerline is the shortest weighted path c between two points p_0 and p_1 . The energy functional

$$E(c) = \int_{c^{-1}(p_0)}^{c^{-1}(p_1)} F(c(s)) ds, \quad (6.20)$$

needs to be minimized to obtain this path. Here $F(x) = \frac{1}{r(x)}$, with the radius $r(x)$ of the largest inscribed sphere. Taking the mean radius of those spheres in each tube we can determine the radius ratio h of subsequent tubes.

Determine θ and α with Paraview: We inscribed triangles (cf. Figure 6.3) in the geometry using Paraview to determine θ and α . The top points of the triangles are located at the bifurcations of the corresponding centerlines from the previous generation. The bottom points of the triangles are either at the following bifurcations or at the endpoints of the centerlines. With the inscribed

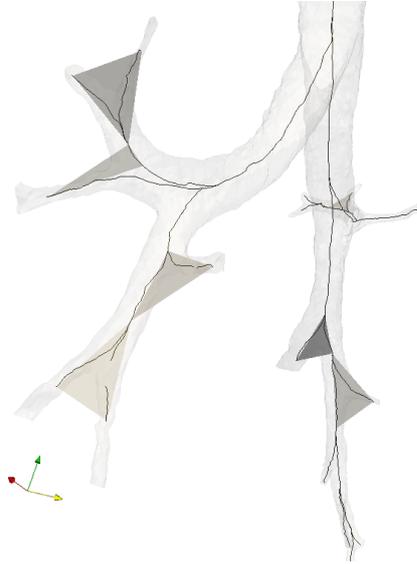


Figure 6.3: Patient-specific bronchiole tree with centerlines and inscribed triangles in the segmented patient specific bronchial tree.

triangles and the corresponding normals it is easy to compute the angle θ at the top of the triangles and the angle α . The latter is the angle between the normals of subsequent triangles. The corner points of the triangles are also used to determine the lengths of each tube and calculate the length-to-diameter ratio of each tube.

Now we can build an asymmetric schematic tree using the extracted parameters. We collected the necessary data for the CT-cast represented in Figure 3.4 in the Tables D.1, D.2 and D.3 in Appendix D on page 95.

Building Blocks: The asymmetric tree consists of basic bifurcations, each constructed by three tubes (cf. Figure 6.4) with length l and radius r . Length

and radius are determined via h and the length-to-diameter ratio, whereas as precondition, the diameter at the top of the trachea is given. Literature values of $1.5 - 2\text{cm}$ are consistent with our chosen patient-specific diameter [58]. The bifurcations can be modified geometrically, i.e. scaled and rotated (cf. Fig-

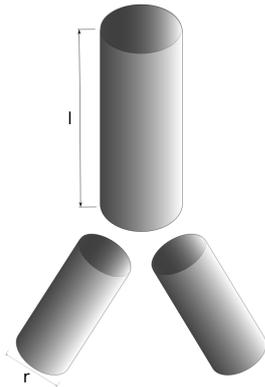


Figure 6.4: A basic bifurcation without the transition zone made up of three tubes with respective length l and radius r .

ure 6.5). The transition zone, which is the area where the tubes are connected,

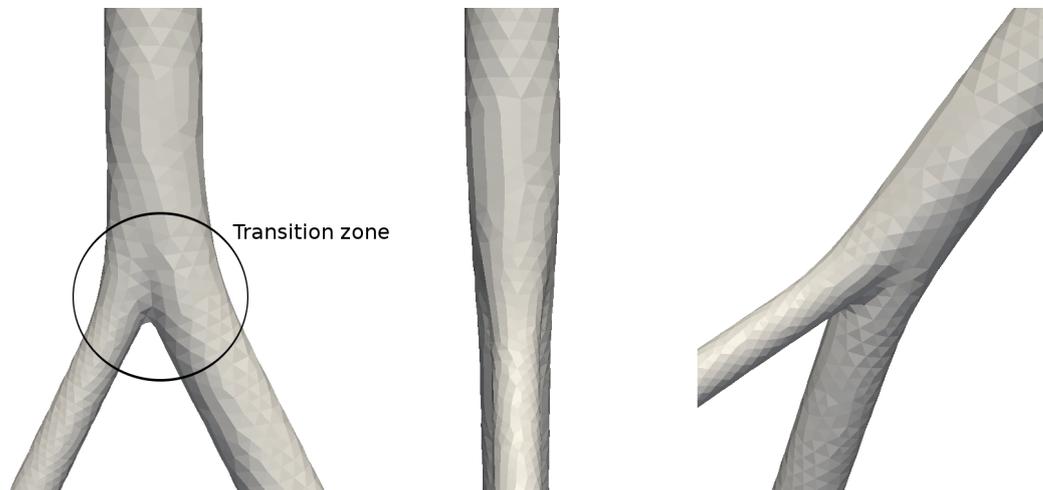


Figure 6.5: View of a single asymmetric bifurcation with the transition zone. Geometric transformations like rotations can be applied to combine consecutive bifurcations.

is constructed similar to the proposal of Lee et al. in [60]. However, we paid less attention to the specifics at the carinal ridge since we want to focus on few parameters and keep the geometry as simple as possible. Several bifurcations are connected using a script and we obtain the schematic bronchiole tree. The result resembles the patient-specific tree (cf. Figure 6.6a) very well with respect

to the extracted parameters. The schematic tree (cf. Figure 6.6b) can thus mimic the main features of the patient-specific tree [29]. The extraction process

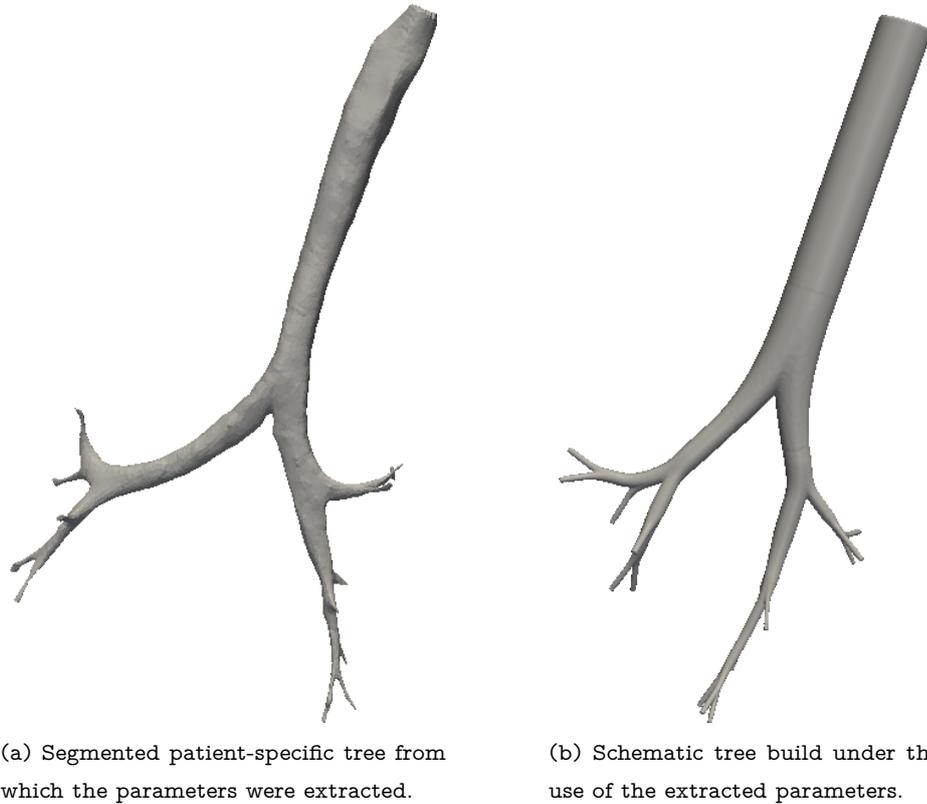


Figure 6.6: The tracheobronchial trees used for particle tracking. The main features resemble each other.

will need to be automatized in the sequel to get an advantage of the analytical model (cf. Section 6.3). Then we can predict the particle deposition up to a few percents from a patient-specific scan without tedious fluid and particle flow simulations.

The Underlying Physical Model

We perform fluid flow and particle simulations for both geometries, the schematic and the patient-specific tree. For stability reasons, we use a homotopy method for the velocity. This creates a smooth path from the solution with zero velocity at the outlet boundaries to the intended initial value. In further studies, it will help to simulate a complete breathing cycle, i.e. inhalation and exhalation. The latter is not a homotopy method per se, but the realization is very similar.

The resulting system to determine the underlying fluid field at each time-

step t are the incompressible Navier-Stokes equations:

$$\begin{cases} \rho_F \frac{\partial u}{\partial t} + \rho_F (u \cdot \nabla) u - \mu_F \Delta u + \nabla p = 0 & \text{in } \Omega \\ \nabla \cdot u = 0 & \text{in } \Omega. \end{cases} \quad (6.21)$$

This system is completed with velocity boundary conditions given by the pressure-driven bronchiole model (see Section 4.2 on page 32) at the outlets and do-nothing boundary conditions at the trachea inlet. No-slip boundary conditions are imposed at the inner walls of the bronchial tree.

Then we can determine the particle deposition. Depending on whether stationary or instationary results are desired, several time-steps or a specific stationary solution are taken for the underlying fluid field. We compute the particle trajectories by integration of Newton's second law of motion with a Stokes drag force on the right-hand side (cf. Section 6.1).

Implementation

ODE Solver: We used two different methods to solve Newton's second law of motion: The solver *LSODE* included in Octave [22] as a reference and an implicit or backward Euler method.

The Octave solver can be used in a C++ framework [66] which is linked against HiFlow³. This solver has the advantage that it is easily tune-able and extensively tested. *LSODE* takes as input parameters

- an initial solution, which we set either to the fluid velocity at the trachea inlet or the zero vector,
- the starting time t_0 and
- a function for the right-hand side, which turns out to be the drag force (see Eq. (6.4)).

Then, we perform the integration routine with a vector t that contains all the time-steps where we evaluate the trace. In the code, we need to include the necessary headers and call the appropriate Octave functions, see Listing C.1 in Appendix C.

However, to be able to parallelize this step, we implemented also another solver which uses the implicit Euler method. Let us denote a general ODE by

$$y'(t) = f(t, y(t)) \quad (6.22)$$

where $f \in [t_0, \infty) \times \mathbb{R}^d \rightarrow \mathbb{R}^d$, and the initial condition $y_0 \in \mathbb{R}^d$. By the theorem of Picard-Lindelöf there exists a unique solution to 6.22 if f is Lipschitz continuous and y is continuous [3, Th. 4.2.4]. The implicit Euler method approximates the derivative of y by

$$y'(t) \approx \frac{y(t) - y(t - \Delta t)}{\Delta t} \quad (6.23)$$

where $\Delta t := t_{n+1} - t_n$. Then we can compute the solution by the following iterative scheme:

$$y_{n+1} = y_n + \Delta t f(t_{n+1}, y_{n+1}). \quad (6.24)$$

Note that this is an implicit scheme, that is, we have to solve a linear system for each time-step, see Listing C.2 in Appendix C.

The interpolation of the values of the underlying fluid field in the lungs and the determination of the trapping point where the particle hits the wall are the critical points in this implementation approach.

Interpolation: In a first attempt, we interpolated the fluid field values via an inverse distance weighting. The interpolation method takes the velocity of the closest vertex and all its neighbors and weights it by the distance to the actual point of the particle

$$u(x) = \sum_{i=0}^N \frac{w_i(x) u_i}{\sum_{j=0}^N w_j(x)}. \quad (6.25)$$

This is called Shepard's method [79] where

$$w_i(x) = \frac{1}{d(x, x_i)^p} \quad (6.26)$$

with the distance $d(x, x_i)$ from the coordinates of the particles to the respective vertices. We used $p = 1$ for our computations. Note that this approach only uses velocity values at the points of the mesh, therefore we loose accuracy and errors can build up. We can circumvent this if we use the DoFs that we computed with the Finite Element ansatz. It is more expensive but more accurate.

For this approach, we have to transfer the full DoF and FEM information (cf. Section 2.2.3 and 2.2.7) from the simulation of the underlying fluid field to the post-processing step. Therefore, we have to setup the mesh and the Finite Element ansatz in the post-processing exactly the same way as in the underlying fluid flow simulation. The solution at each DoF is transferred to the particle simulation using the *hierarchical data format version 5 (HDF5)*. "HDF5 is a unique technology suite that makes possible the management of extremely large and complex data collections" [85]. It is implemented in HiFlow³ to write backup

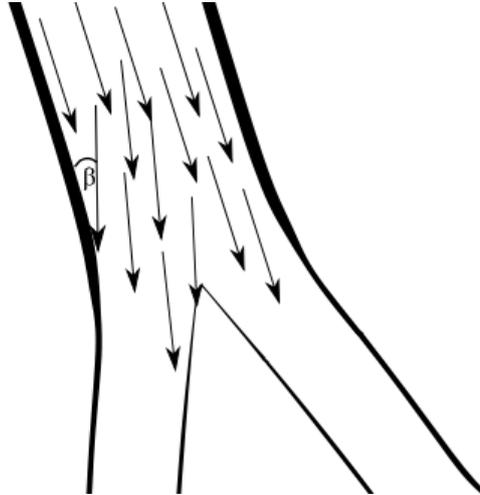


Figure 6.7: Schematic particle trajectories in a 2D bifurcation. The flat angle β causes difficulties determining the capture point of a particle.

solutions to the disk in parallel with good performance. With the information of the Finite Element ansatz and the velocity values at each DoF, we can now interpolate the velocity at the current coordinates of the particle. This yields the velocity vector which is exact in terms of the underlying fluid field simulation.

Trapping Point: The second critical point we mentioned earlier is the trapping point of the particle. A mucus layer is located at the wall of the lungs (cf. Section 3.1), which means that particles hitting the wall are captured instantly. The decision whether a particle is captured or not is very critical because of the immediate impact on the escape rate. However, it is very difficult to decide in the post-processing implementation approach if, and if so, where a particle hit the wall. This is due to numerical inaccuracies that can build up during the computation and the very flat angle β between the particle trajectory and the wall, cf. Figure 6.7. To determine the capture points as accurately as possible in a geometrical sense, we make use of the *Computational Geometry Algorithms Library (CGAL)* [10]. CGAL implements the so-called *exact computation paradigm* [96], therefore we can direct our attention to the problems arising from solving ODEs rather than geometrical inaccuracies.

Evaluation: To determine a deposition pattern, not only the global escape rate (cf. Section 6.2.2), but also the outlet where the particle escapes the simulated part of the tracheobronchial tree is important. To compute this pattern, we

define planes with an origin vector and a normal vector located at each outlet. Then we check if the particle crossed the plane or lies in the plane. If this is true, the particle escaped through the respective outlet and, if necessary, we can note its position.

This algorithm is used to compute a generation- or bifurcation-wise escape rate as well, see Section 6.2.2. For this special escape rate, planes, perpendicular to the bulk flow located between the bifurcations, are specified (cf. Figure 6.8). Then again we check whether the current particle coordinates crossed the respective plane.

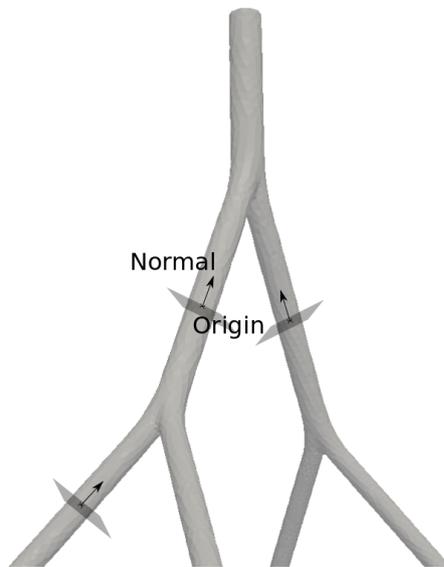


Figure 6.8: Scheme of a tracheobronchial tree with inscribed planes between bifurcations to compute a generation- or bifurcation-wise escape rate.

Setup of the Simulation

Now that we explained the necessary methods for the computation of the particle deposition, we can describe the actual setup of the simulations:

1. We mentioned before that we need to precompute an underlying fluid field which transports the particles.

This is computed with HiFlow³ [42, 43] (cf. Section 6.2.1) in parallel on the Bull cluster system named Taurus, see Appendix A.1. The computation of the underlying fluid field is based on the Navier-Stokes equations derived in Section 2 and the pressure-driven bronchiole model described in Section 4.2.

2. Unless otherwise stated, we inject 10,000 particles uniformly distributed at the entrance of the trachea right below the pharynx (see Figure 3.1 on page 21). This computation runs in parallel trivially since the particles are independent of each other and we have to compute particles of different kinds, that is, Stokes numbers. We chose to simulate water droplets because they usually serve as carriers for pharmaceutical aerosols [26]. In our simulation we used water droplets with density $\rho_P = 0.9982g/cm^3$ at $20^\circ C$. This simulation is executed on the XServe cluster (see Appendix A.2), because it has enough cores to run the simulations for 20 different Stokes numbers simultaneously. Besides, the cores are connected by an Ethernet network only. However, this relatively slow communication does not bother us, since we do not need any communication here.
3. Since the maximum velocity and the viscosity of the fluid, the diameter of the trachea and the density of the particles (see Equation (6.19)) are fixed, the only parameter that varies is the particle diameter. Therefore, the results of the simulations are ordered by the size of the resp. particles, which is between 1-40 microns. For comparison, a human hair is between 40-600 microns, particles stemming from car emission are usually between 1-150 microns [23].

6.2.2 Results

In the following we present the numerical results of the flow and particle simulations. The simulation output is parsed with Python scripts and plotted with Matplotlib [49]. We created videos (ask for access) and visualizations, e.g. Figure 6.9, using Paraview [40]. The first sections confirm that the assumptions made like independence of Reynolds number and simulation of 10,000 particles are valid. We also draw a comparison between the two different simulation codes used in our institute, OpenLB [44, 45] and HiFlow³ [42]. The latter sections confirm results in larger geometries and validate the analytical model.

Independence of Reynolds Number

Simulations in complex geometries like those arising in the field of biomedical engineering are usually difficult to handle, especially for high Reynolds numbers. It is therefore much easier and quicker if we can assume that the observed results vary only by a very small amount with respect to the Reynolds number.



Figure 6.9: The structure of the lung with the segmented first generations of the tracheobronchial tree. The sizes of the arrows depict the velocity. Micron particles float downwards towards the alveolar region.

Figure 6.10 shows our results for the escape rate of particles for one bifurcation. Similar results with even higher Reynolds numbers were gathered by Filoche et al. [19] in Figure 6.11. We can see that the change in the escape rate is negligible with respect to the change of the Reynolds numbers. Hence it is sufficient to compute the particle deposition with low Reynolds numbers, so we can reduce the computational effort.

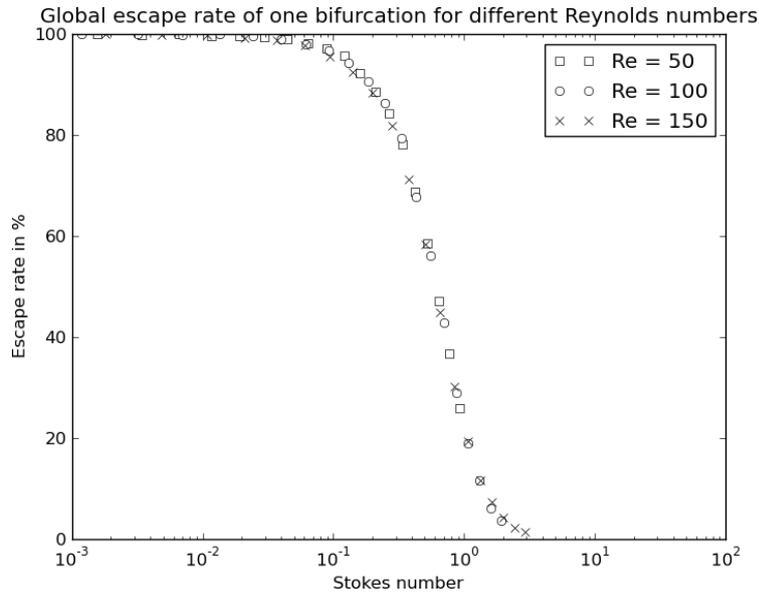


Figure 6.10: Global escape rate in terms of Stokes number in one bifurcation for different Reynolds numbers. The scale of the Stokes number is logarithmic.

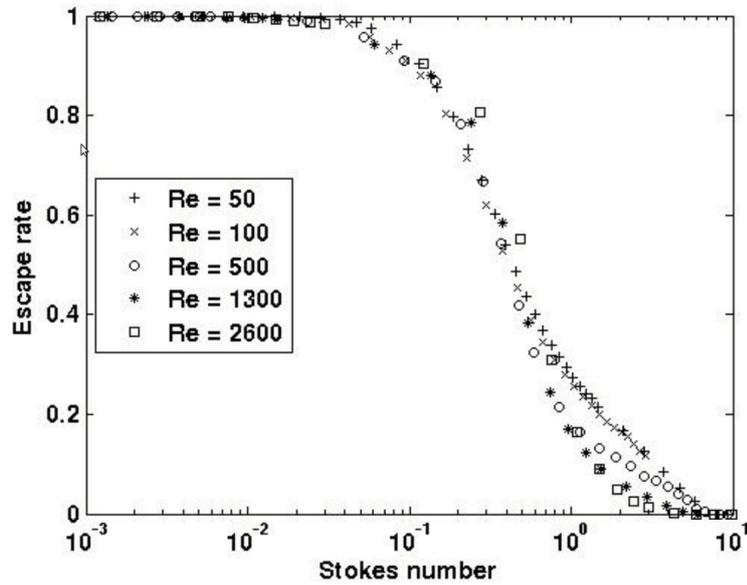


Figure 6.11: Global escape rate in terms of Stokes number in one bifurcation for different Reynolds numbers. The scale of the Stokes number is logarithmic. This plot is taken from [19].

Statistical Considerations

We stated in Section 6.2.1 that we used 10,000 particles for our simulations. As we can see here, this is reasonable from a statistical point of view since the standard deviation is small compared to the effect we want to measure. In

Figure 6.12 we plotted the arithmetic mean of the global escape rate E_i , $i = 1, \dots, n$

$$\bar{E} = \frac{1}{n} \cdot \sum_{i=1}^n E_i \quad (6.27)$$

and the standard deviation

$$\sigma_E = \sqrt{\frac{1}{n} \sum_{i=1}^n (E_i - \bar{E})^2}. \quad (6.28)$$

The bar $\bar{\cdot}$ denotes the arithmetic mean. For each number of particles (500, 1,000, 2,000, 5,000, 10,000 and 20,000) we ran the simulation $n = 10$ times always with the same Stokes number $St = 0.575$. We can see from this plot that for a statistically meaningful statement it is sufficient to compute 10,000 particles.

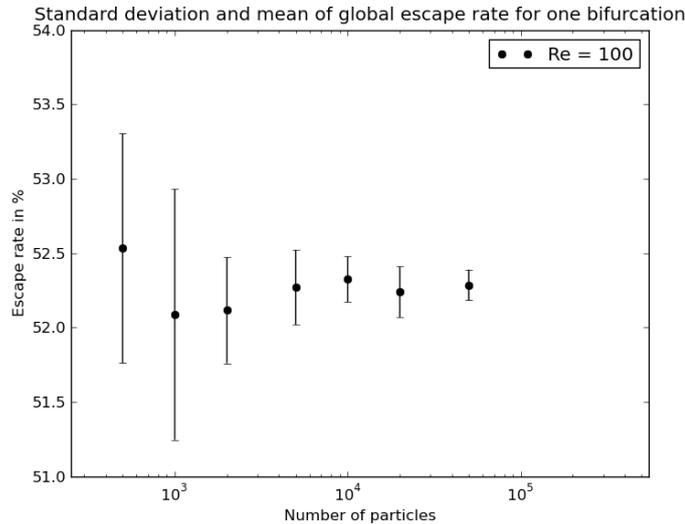


Figure 6.12: The standard deviation and mean of the global escape rate plotted against the number of particles for Reynolds number $Re = 100$ for the fluid and Stokes number $St = 0.575$ for the particles.

Comparison to OpenLB of Simulations in One Bifurcation

Similar simulations to the ones presented here were obtained in the context of the Master's thesis of T. Henn [41, Ch. 5]. The simulation setup with one bifurcation was identical to the setup presented here (cf. Section 6.2.1) except for the method. Henn implemented the particle deposition using the software package OpenLB [44, 45]. This code, co-developed in our group, uses the Lattice-Boltzmann method to simulate fluid flow. If we compare the plots 6.10 and 6.13, we see that the results coincide.

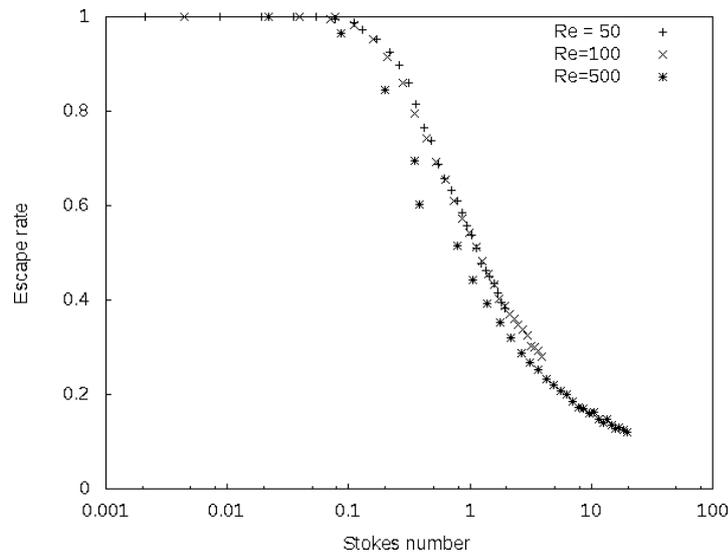


Figure 6.13: Global escape rate in terms of Stokes number in one bifurcation for different Reynolds numbers. The scale of the Stokes number is logarithmic. This plot is courtesy of T. Henn in [41].

Global Escape Rate

The global escape rate (6.1) of the schematic and patient-specific tree (in percentage %) is plotted against the Stokes number of the particles, see Figure 6.14. The overall results are very good and consistent with the expected ones. However, we can see that the values differ for higher Stokes numbers. This is because the radius of the trachea in the patient-specific geometry differs strongly and we can see in Figure 6.16 that these particles are in fact filtered directly in the first bifurcation. If we would simulate the oral and nasal region, too, these particles would have been filtered there already. This was experimentally and numerically shown in many studies, e.g. [11, 12, 13, 41, 50, 51, 54, 78].

Bifurcation-wise Escape Rate

The bifurcation-wise escape rate is calculated by the method described in Section 6.2.1. In the 4-generation tree there exist three levels of consecutive bifurcations: Two outlet planes below the first bifurcation, four outlet planes below the second level of bifurcations and 8 outlet planes below the third level of bifurcations (cf. Figure 6.15). We compute the ratio of particles escaping through each outlet plane and insert the value into the pyramid on the right in Figure 6.15 which represents the tracheobronchial tree. For both, the patient-specific and the schematic tree and each Stokes number one pyramid of numbers is created.

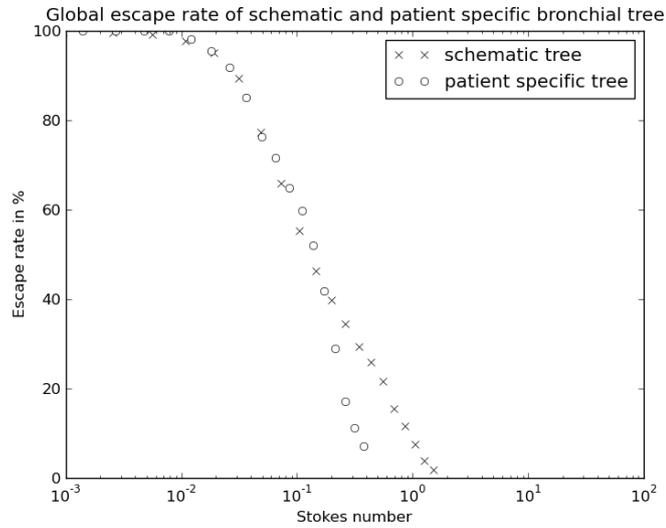


Figure 6.14: Global escape rate in terms of Stokes number of the schematic and the patient-specific 4-generation tree. The scale of the Stokes number is logarithmic.

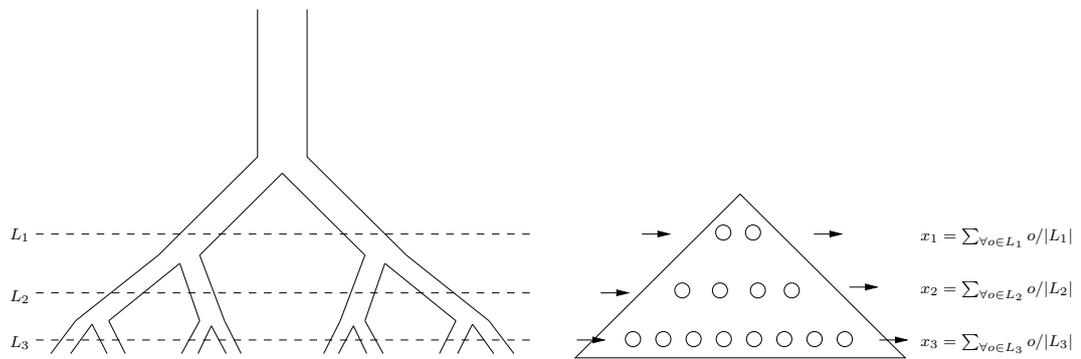


Figure 6.15: Scheme of the levels of bifurcations in a 4-generation tree. The results are inserted into the “o”’s of the pyramid and the mean value is calculated.

To compare the results, we compute the mean value of each level in the pyramid of the schematic tree and divide it by the matching mean value in the patient-specific tree. We can see in Figure 6.16 that the ratio is approximately one for small Stokes numbers at all three levels, which is also the expected result. For higher Stokes numbers, the ratio differs, primarily at the first level. However, these types of particles are usually filtered out in the nasal or oral region and not likely to be found in the lungs, as we already discussed above.

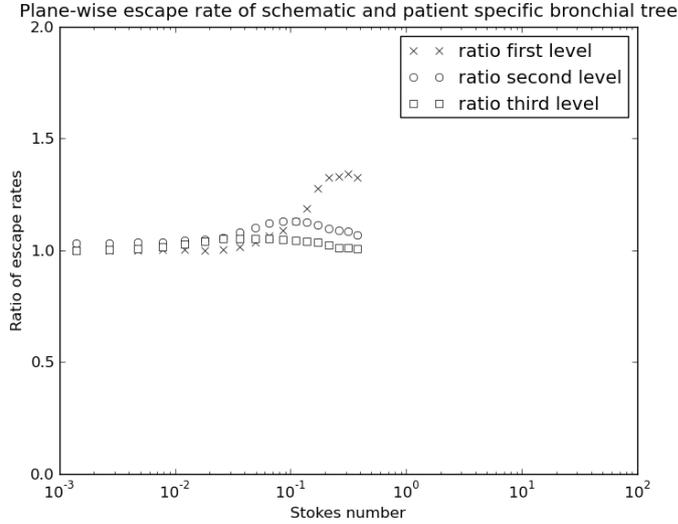


Figure 6.16: Ratio $\frac{\max(E_{sc}, 100 - C_{sc})}{\max(E_{ps}, 100 - C_{ps})}$ per generation of the schematic (sc) to the patient-specific (ps) tree. The scale of the Stokes number is logarithmic.

6.3 A Simple Bifurcation Model

In [19] Filoche et al. introduced an analytical model which predicts the escape rate of particles using three parameters (h, θ and α) and the Stokes number (see Figure 6.2 and Equation (6.19)). The escape rate was found to be multiplicative. This means for the Weibel tree that computing the n -th power of the escape rate for one generation is the same as computing the escape rate for n generations (cf. Figure 6.17). In lung geometries with a varying homothety ratio h , the global escape rate can be computed by multiplying the locally shifted escape rates of each bifurcation. The shift depends on the local Stokes number (6.19) of the particles. The local Stokes number changes because the velocity of the underlying fluid at the entrance of the respective bifurcation and the diameter, also at the entrance, changes for each bifurcation. The length L to radius R ratio in each bronchiole for the bifurcations is assumed to be $L/D = 3$ and each bifurcation is assumed to be planar (cf. Section 6.2.1).

To validate this simple bifurcation model, we want to present the simulation results 6.18 for two bifurcations with OpenLB gained in [41]. This simulation is for the easiest case of a schematic Weibel “A” tree (cf. Section 6.2). The geometry 6.19 was obtained through the method explained in Section 6.2.1.

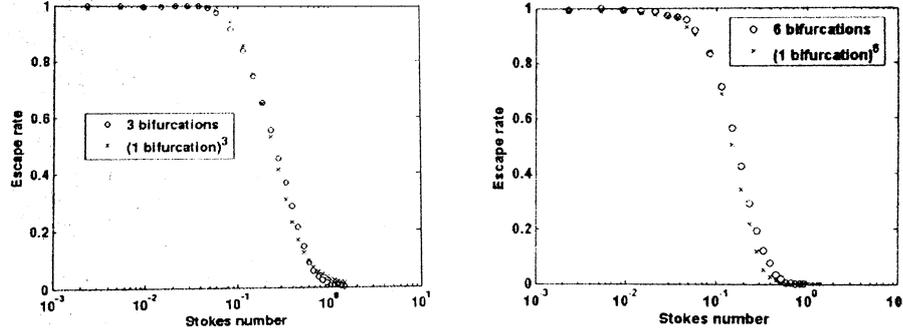


Figure 6.17: Plots comparing the results of the escape rate in one bifurcation to the third and sixth power, resp., and the results of three and six simulated generations, resp. These plots are courtesy of M. Filoche et al [19].

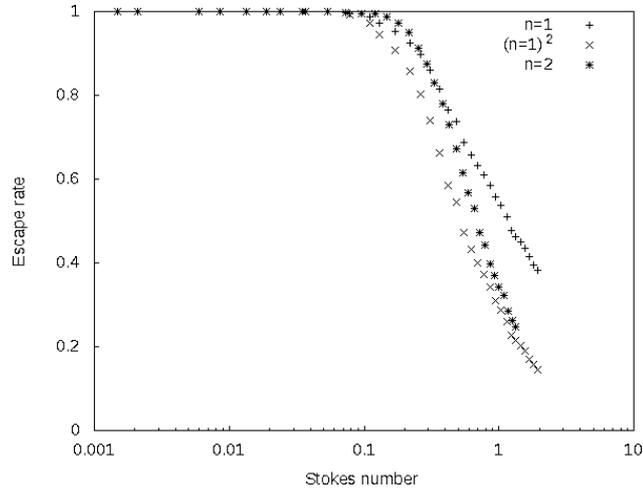


Figure 6.18: Escape rates for one and two bifurcations calculated using OpenLB for Reynolds number $Re = 50$. Additionally, the result for the multiplicative approach $(n = 1)^2$ is plotted. This plot is courtesy of T. Henn [41].

6.3.1 Multiplicative Approach in the Numerical Simulation

To justify the multiplicative approach described in [19], we launched particles from different planes in the lungs, see Figure 6.8, and calculated the capture efficiency after the next bifurcation. Then we multiply the capture rates, which yields the global escape rate

$$E(St) = \prod_{i=1}^N E_i(St_{i-1}), \quad (6.29)$$

where St_{i-1} is the local Stokes number (changed due to the different diameter of the plane, where we launched the particles). The results are depicted in Figure 6.20. We can see that the multiplicative calculations match the results of

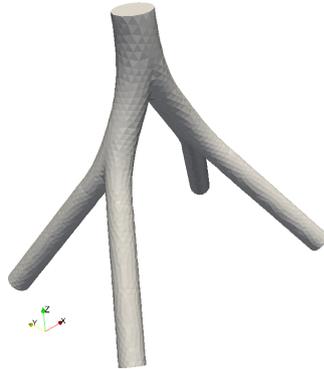


Figure 6.19: The Weibel “A” model tree with two bifurcations.

both the patient-specific and the schematic tree very well. The small deviations occur due to the fact that the positions of the particles in the inserted planes in the overall computation and the positions of the particles in the simulations of single bifurcations is highly different. We could fix this by modifying the initial positions of the particles from a uniform distribution to a shifted distribution that depends on the bifurcation above. However, this would not be consistent with the analytical multiplicative model.

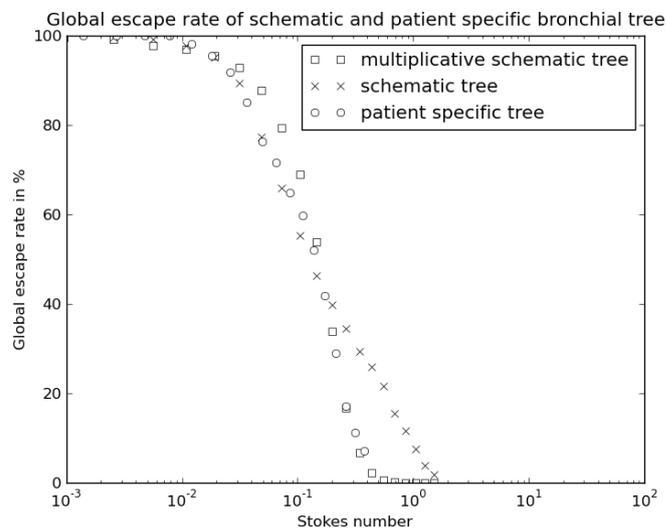


Figure 6.20: Comparison between the patient-specific (circles), the schematic (crosses) and the multiplicative tree (squares).

Summary and Outlook

We have shown an approach for the modeling and simulation of fluids and particles in the human lungs, e.g. fine dust from car emission or pharmaceutical drugs that can be delivered via the respiratory system, emphasizing the similar behavior of particle deposition in a parametrized schematic geometry and a patient-specific geometry.

For this, we introduced dedicated parts of the basic theory for the modeling of fluids. We focused especially on the discretization with a Finite Element method and used the intrinsic parallelization in the software package HiFlow³.

Then we briefly presented the anatomy and physiology of the human lungs and introduced different measuring and imaging methods. Big potential for more accurate simulations lies in the development of new imaging techniques like Hyperpolarized Helium-3 MRI and the combination of existing imaging techniques. In the future, an additional fluid flow simulation that comes directly with the scanned image can provide new possibilities for the physicians to diagnose and treat diseases. However, the structure of the human lungs is too complex and cannot be captured as a whole by nowadays medical imaging techniques. To get around this, we presented models that can account for the non-resolvable part. These models also help to reduce the necessary computational power to simulate breathing, which would not be possible otherwise. With the upcoming exascale computing, better imaging techniques and segmentation tools, we can expect to resolve and simulate more generations of the lungs with computational fluid dynamics. But at least for the next decade, the whole lung can only be captured completely using models.

We established a new pressure-driven bronchiole model that uses a binary tree to model the lower parts of the human lungs down to the alveoles. This model is coupled to the Navier-Stokes simulations of the resolvable upper part of the human lungs in a two-scale modeling approach. The model is not meant to capture the exact physical behavior of the lower lungs nor is it meant to predict

a certain behavior of them. However, it is there to provide boundary conditions for the simulations of the resolvable part that are not possible otherwise and can model diseases that have an effect on the pressure or the resistance in the human lungs, like asthma or COPD, by adapting the respective parameters.

With the help of sensitivity analysis and singular value decomposition, we want to establish the possibility to investigate the sensitivity of models for the lungs. Here, we have the prospective to examine model reduction, too. This can be done by extracting the principal components of such models by computation of the highest singular values and vectors. Analyzing the decay of the singular values, we can then judge if we can reduce the models or not. The sensitivity analysis is based on the linearization of the fluid flow model and its adjoint solution, which operates backward in time and can therefore find the source of perturbations. Results for the sensitivity are presented in a schematic 2D lung geometry. We have implemented the code that provides these tools using the software package HiFlow³.

In addition to the fluid flow in the airways, we are also interested in the deposition of particles. We followed a one-way coupling approach that enables us to simulate the particles' trajectories in a Lagrangian frame of reference. We proposed a method to extract three main geometrical parameters for the airways from segmented CT scans and provide scripts to build schematic geometries from them. With these parametrized schematic geometries, we were able to show significant similarity for particle depositions in schematic and respective patient-specific airway geometries. With these simulations, we numerically validated an analytical model of Filoche et. al. [19] that predicts multiplicative behavior of particle deposition in the human lungs. The generation of parametrized schematic tree geometries can be used in the future to get a better understanding of the fundamentals of the behavior of particles in structures like the lungs. It is possible to obtain the deposition of particles just from the parameters automatically extracted from CT scans very quickly. According to the available time and the needed accuracy for the results we can adapt the parameters for the schematic geometries so that we have them either very generic or almost exactly patient-specific. With the different complexity and genericity of the models, we want to build a hierarchy of results that allows us to gain a better understanding of the behavior of particles flowing through the lungs. We will then be able to extract generic features, characteristics of certain diseases and patient-specific attributes.



Cluster

A.1 Bull Cluster Taurus

The Taurus cluster system consists of 10 nodes, each of them with 2 hexacore Intel Xeon X5650 processors providing in total 120 cores. 6 of these nodes are equipped with 1 NVIDIA C2070 GPU allowing high performance GPU cluster computing. All the nodes have a minimum of 96GB and a maximum of 192GB memory, which allows computations that need a lot memory. 1 extra node consists of 8 octacore CPUs with 1 TB memory providing an extra 64 cores. All nodes are connected via Infiniband 4x QDR Network (theoretical 32 GBit/s p2p data transmission rate) and use a 20TB storage node.

A.2 XServe Cluster

The XServe cluster system consisted to the time of the computations of 8 nodes (now only 2 nodes), 2x4 cores with 2 threaded Intel Xeon X5550 processors. The nodes each have 24 GB of memory and are connected via an Ethernet 1Gbit network. This is very slow compared to the Bull cluster, but plays no role in our computations since this cluster was only used for the particle simulations which need no communication at all.

B

Eigenvalue/Eigenvector Code

```
1 // Use LAPACK to compute eigenvalues and eigenvectors
double h[k*k], wr[k], wi[k];
3
// transform H into Fortran vector
5 for (int i = 0; i < k; ++i) {
    for(int j = 0; j < k; ++j)
7     h[j+k*i] = H(i, j);
}
9
double vr[k*k], vl[k*k];
11
char balanc = 'N';
13 char jobvl = 'N';
char jobvr = 'V';
15 char sense = 'N';
17
int lwork = 3*k;
double work[lwork];
19 int iwork = 2*k;
21
int ilo = 1;
int ihi = k;
23
double scale[k];
25 double abnrm;
double rconde[k];
27 double rcondv[k];
int info;
29
dgeevx(&balanc, &jobvl, &jobvr, &sense, &k, h, &k, wr,
31 wi, vl, &k, vr, &k, &ilo, &ihi, scale, &abnrm,
rconde, rcondv, work, &lwork, &iwork, &info);
```

with $wr[k]$ the eigenvalues and $vr[k * k]$ the eigenvectors of H_k .

Code for the ODE Solvers

```

#include <octave/oct.h> #include <octave/config.h> #include
2 <octave/Matrix.h> #include <octave/LSODE.h>

4 ODEFunc odef(rhs(particle, t));
LSODE ls(initial_solution, t0, odef);
6 result = ls.do_integrate(t);

```

Listing C.1: Code for the Octave ODE solver.

```

void Particle::compute_time_evolution(int max_time, const double deltat) {
2  std::vector<double> underlying_vel(coords_.size(), 0.0);
  // 1. time loop
4  for (double time = 0; time < max_time; time += deltat) {
    // get underlying vel
6    underlying_velocity(underlying_vel);

    // put time in result vector
8    result_vector_.push_back(time / timestep());

    // 2. spatial loop
10   for (int i = 0; i < tdim(); ++i) {
    // compute new particle vel
12    // implicit euler
    particle_velocity_[i] = (particle_velocity_[i] + deltat * 3. * M_PI *
14     fluid_viscosity_ * diameter_ / mass_ * (underlying_vel[i])) / (1 +
    deltat * 3. * M_PI * fluid_viscosity_ * diameter_ / mass_);

16    // compute new coords
    old_coords_[i] = coords_[i];
18    coords_[i] += particle_velocity_[i] * deltat;

    // 3. put coords in result vector
20    result_vector_.push_back(coords_[i]);
22  }
24 }
}

```

Listing C.2: Code for the implicit Euler method to solve the ODE.

Parameter Extraction Tables

	Length ratio		Ratio h
	H_0	H_1	
Generation 0 -1	6.4742	5.9726	1.08
Generation 10 0	5.0544	6.4742	0.78
Generation 11 0	4.594	6.4742	0.71
Generation 20 10	2.3818	5.0544	0.47
Generation 21 10	3.9374	5.0544	0.78
Generation 22 11	3.029	4.594	0.66
Generation 23 11	3.1548	4.594	0.69
Generation 3x 20	1.16	2.3818	0.49
Generation 30 20	1.6932	2.3818	0.71
Generation 3xx 21	2.0998	3.9374	0.53
Generation 31 21	2.8063	3.9374	0.71
Generation 3xxx 22	1.4056	3.029	0.46
Generation 32 22	2.3023	3.029	0.76
Generation 3xxx 23	1.1508	3.1548	0.36
Generation 33 23	2.5744	3.1548	0.82
Generation 4x 30	0.9059	1.6932	0.54
Generation 40 30	1.343	1.6932	0.79
Generation 41 31	1.9953	2.8063	0.71
Generation 4xx 31	0.8459	2.8063	0.30
Generation 4xxx 32	1.1566	2.3023	0.50
Generation 4xxxx 32	1.1896	2.3023	0.52
Generation 4xxxxx 33	1.2758	2.5744	0.50
Generation 4xxxxxx 33	1.2336	2.5744	0.48
Generation 5x 40	0.9805	1.343	0.73
Generation 5xx 40	1.1358	1.343	0.85
Generation 5xxx 41	0.9469	1.9953	0.47
Generation 50 41	1.9138	1.9953	0.96
Generation 6x 50	1.1125	1.9138	0.58
Generation 60 50	1.3227	1.9138	0.69
Generation 7x 60	1.0148	1.3227	0.77
Generation 7xx 60	1.1771	1.3227	0.89
		Mean Ratio	0.65
	6.4742		

Table D.1: Table for the computation of the radius ratio h .

D Parameter Extraction Tables

Angle between the bifurcations						
Coordinates of the plane				Vector 1	Vector 2	theta
	x	y	z			
Generation 0						
top	214.465	175.034	-357.909	x	16.022	-38.853
left	198.443	179.433	-392.233	y	-4.399	-11.72
right	253.318	186.754	-417.805	z	34.324	59.896
						57.44
Generation 1 0						
top	198.443	179.433	-392.233	x	23.127	17.28
left	175.316	179.372	-395.552	y	0.061000	-2.557000000000002
right	181.163	181.99	-422.557	z	3.319000	30.324
						52.29
Generation 1 1						
top	253.318	186.754	-417.805	x	-14.871	-16.516
left	268.189	191.611	-435.603	y	-4.857	5.244
right	269.834	181.51	-421.963	z	17.798	4.158000000000002
						45.85
Generation 2 0						
top	175.316	179.372	-395.552	x	6.872	11.322
left	168.444	172.971	-391.285	y	6.401	-10.797
right	163.994	190.169	-385.3	z	-4.267	-10.252
						74.23
Generation 2 1						
top	181.163	181.99	-422.557	x	9.153	6.176000000000002
left	172.01	174.186	-434.248	y	7.804	-2.366999999999999
right	174.987	184.357	-430.508	z	11.691	7.950999999999997
						40.96
Generation 2 2						
top	268.189	191.611	-435.603	x	-5.116	-4.365999999999999
left	273.305	193.009	-449.522	y	-1.398	-12.164
right	272.555	203.775	-437.587	z	13.919	1.983999999999998
						69.89
Generation 2 3						
top	269.834	181.51	-421.963	x	-11.795	-3.367000000000002
left	281.629	174.093	-434.329	y	7.417	2.001
right	273.201	179.509	-419.641	z	12.366	-2.322
						72.26
Generation 3 0						
top	168.444	172.971	-391.285	x	4.922	2.369
left	163.522	170.026	-384.034	y	2.945	7.199000000000001
right	166.075	165.772	-390.357	z	-7.25100	-0.927999999999997
						55.89
Generation 3 1						
top	174.987	184.357	-430.508	x	9.257	9.108
left	165.73	194.908	-438.226	y	-10.551	-3.328
right	165.879	187.685	-447.365	z	7.718	16.857
						36.77
Generation 3 2						
top	273.305	193.009	-449.522	x	-4.34700	-9.195999999999997
left	277.652	198.228	-466.167	y	-5.21900	2.442999999999998
right	282.501	190.566	-463.721	z	16.645	14.199
						30.93
Generation 3 3						
top	273.201	179.509	-419.641	x	-8.173	-7.045999999999999
left	281.374	169.233	-411.706	y	10.276	-1.141000000000002
right	280.247	180.65	-405.076	z	-7.935	-14.565
						49.55
Generation 4 0						
top	166.075	165.772	-390.357	x	-0.55200	1.926999999999999
left	166.627	158.055	-388.353	y	7.717	8.430999999999998
right	164.148	157.341	-392.088	z	-2.00400	1.730999999999999
						30.83
Generation 4 1						
top	165.879	187.685	-447.365	x	9.299	6.068999999999999
left	156.58	185.256	-460.634	y	2.429	-3.233
right	159.81	190.918	-460.48	z	13.269	13.115
						23.44
Generation 5 0						
top	159.81	190.918	-460.48	x	13.439	1.092000000000001
left	146.371	193.133	-474.017	y	-2.215	-3.462999999999999
right	158.718	194.381	-467.282	z	13.537	6.801999999999996
						39.39
Generation 6 0						
top	158.718	194.381	-467.282	x	2.476	1.226999999999998
left	156.242	199.568	-479.865	y	-5.187	-4.935
right	157.491	199.316	-472.252	z	12.583	4.970000000000003
						22.05
of absolute values					Mean Theta	46.78
					Max	74.23
					Min	22.05

Table D.2: Table for the computation of the opening angle θ .

Normals	Normals		Alpha1 = arccos(a1 * a2)	Alpha2 = 180 - alpha1	alpha
Generation 0	Generation 1 0				
0.0596934		0.0159581			
-0.986225		-0.995533			
-0.15426		-0.0930555	4.34429678539868	175.655703214601	4.34
Generation 0	Generation 1 1				
0.0596934		0.374689			
-0.986225		0.766146			
-0.15426		0.522138	144.466010589138	35.5339894108617	-35.53
Generation 1 0	Generation 2 0				
0.0159581		0.601575			
-0.995533		-0.119196			
-0.0930555		0.789873	86.860825801911	93.139174198089	86.86
Generation 1 0	Generation 2 1				
0.0159581		-0.789002			
-0.995533		0.005005			
-0.0930555		0.61437	94.2865203672206	85.7134796327794	-85.71
Generation 1 1	Generation 2 2				
0.374689		0.910617			
0.766146		-0.276782			
0.522138		0.306869	73.1797144847637	106.820285515236	73.18
Generation 1 1	Generation 2 3				
0.374689		0.519496			
0.766146		0.854305			
0.522138		-0.0169215	32.8243363832888	147.175663616711	32.82
Generation 2 0	Generation 3 0				
0.601575		0.846361			
-0.119196		-0.215773			
0.789873		0.486945	23.1478376234793	156.852162376521	23.15
Generation 2 1	Generation 3 1				
-0.789002		0.816055			
0.005005		0.459838			
0.61437		-0.350146	148.946580665966	31.0534193340336	-31.05
Generation 2 2	Generation 3 2				
0.910617		0.726539			
-0.276782		0.578291			
0.306869		0.371109	52.017550135817	127.982449864183	52.02
Generation 2 3	Generation 3 3				
0.519496		0.838209			
0.854305		0.333394			
-0.0169215		-0.431572	43.3170008178079	136.682999182192	43.32
Generation 3 0	Generation 4 0				
0.846361		0.837541			
-0.215773		-0.080467			
0.486945		-0.540417	62.4145296129905	117.58547038701	62.41
Generation 3 1	Generation 4 1				
0.816055		0.774654			
0.459838		-0.429361			
-0.350146		-0.464284	53.3238782426821	126.676121757318	53.32
Generation 4 1	Generation 5 0				
0.774654		-0.338538			
-0.429361		0.815447			
-0.464284		0.469509	146.13537370114	33.8646262988603	-33.86
Generation 5 0	Generation 6 0				
-0.338538		-0.983689			
0.815447		-0.0848781			
0.469509		0.15859	70.2289821816641	109.771017818336	70.23
Mean alpha					22.54

Table D.3: Table for the computation of the torque α .

Bibliography

- [1] Alberg, A. J., Ford, J. G., and Samet, J. M. Epidemiology of lung cancer: ACCP evidence-based clinical practice guidelines (2nd edition). *Chest* 132, 3 Suppl (2007), 29–55.
- [2] Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D. *LAPACK Users' Guide*, third ed. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999.
- [3] Atkinson, K. E., and Han, W. *Theoretical Numerical Analysis: A Functional Analysis Framework*. Texts in applied mathematics. Springer, 2005.
- [4] Barkley, D., Blackburn, H. M., and Sherwin, S. J. Direct optimal growth analysis for timesteppers. *International Journal for Numerical Methods in Fluids* 57, 9 (2008), 1435.
- [5] Batchelor, G. K. *An Introduction to Fluid Dynamics*. Cambridge University Press, 1967.
- [6] Bönisch, S., Dunne, T., and Rannacher, R. *Hemodynamical Flows: Aspects of Modeling, Analysis and Simulation*, vol. 37 of *Oberwolfach Seminars*. Birkhäuser, 2008, ch. Numerics of Fluid-Structure Interaction, pp. 333–378.
- [7] Braess, D. *Finite Elemente*. Springer-Verlag GmbH; Auflage: 1, 2007.
- [8] British Thoracic Society. The Burden of Lung Disease.
- [9] Cao, Y. Adjoint sensitivity analysis for differential-algebraic equations: algorithms and software. *Journal of Computational and Applied Mathematics* 149, 1 (2002), 171.
- [10] CGAL. Computational Geometry Algorithms Library. <http://www.cgal.org>.
- [11] Chen, X. B., Lee, H. P., Chong, V. F. H., and Wang, D. Y. A Computational Fluid Dynamics Model for Drug Delivery in a Nasal Cavity with Inferior Turbinate Hypertrophy. *Journal of Aerosol Medicine and Pulmonary Drug Delivery* 23, 5 (2010), 329–338.
- [12] Cheng, K.-H., Cheng, Y.-S., Yeh, H.-C., Guilmette, R. A., Simpson, S. Q., Yang, Y.-H., and Swift, D. L. In vivo measurements of nasal airway dimensions and ultrafine aerosol deposition in the human nasal and oral airways. *Journal of Aerosol Science* 27, 5 (1996), 785–801.
- [13] Cheng, Y. S., Holmes, T. D., Gao, J., Guilmette, R. A., Li, S., Surakitbanharn, Y., and Rowlings, C. Characterization of Nasal Spray Pumps and Deposition Pattern in a Replica of the Human Nasal Airway. *Journal of Aerosol Medicine* 14, 2 (2001), 267–280.
- [14] Ciarlet, P. G. *The Finite Element Method for Elliptic Problems*. Studies in Mathematics and Its Applications. North-Holland Publishing Company, 1978.
- [15] Clift, R., Grace, J. R., and Weber, M. E. *Bubbles, Drops, And Particles*. Dover Books on Engineering. Dover Publications, 2005.

BIBLIOGRAPHY

- [16] Comer, J. K., Kleinstreuer, C., and Kim, C. S. Flow structures and particle deposition patterns in double-bifurcation airway models. Part 2. Aerosol transport and deposition. *Journal of Fluid Mechanics* 435 (2001).
- [17] Crouzeix, M., Philippe, B., and Sadkane, M. The Davidson Method. *SIAM Journal on Scientific Computing* 15, 1 (1994), 62–76.
- [18] Davidson, E. R. The Iterative Calculation of a Few of the Lowest Eigenvalues and Corresponding Eigenvectors of Large Real-Symmetric Matrices. *Journal of Computational Physics* 17 (1975), 87–94.
- [19] de Vasconcelos, T. F., Sapoval, B., Andrade, J. S., Grotberg, J. B., Hu, Y., and Filoche, M. Particle capture into the lung made simple? *J. Appl. Physiol.* 110, 6 (2011), 1664–73.
- [20] Donéa, J., and Huerta, A. *Finite element methods for flow problems*. Wiley, 2003.
- [21] Dössel, O. *Bildgebende Verfahren in der Medizin: von der Technik zur medizinischen Anwendung*. Springer, 1999.
- [22] Eaton, J. W. *GNU Octave Manual*. Network Theory Limited, 2002.
- [23] Engineering Toolbox. Particle Sizes. http://www.engineeringtoolbox.com/particle-sizes-d_934.html, April 2012.
- [24] Errico, R. M. What Is an Adjoint Model? *Bulletin of the American Meteorological Society* 78, 11 (1997), 2577–2591.
- [25] Fefferman, C. L. Existence and Smoothness of the Navier-Stokes Equation. In *The Millennium Prize Problems*. Clay Mathematics Institute, 2006, pp. 57–67.
- [26] Finlay, W. H. *The mechanics of inhaled pharmaceutical aerosols: an introduction*. Chemical, Petrochemical & Process. Academic Press, 2001.
- [27] Forrestal, P. *Anatomica: The Complete Home Medical Reference*. Book Sales, 2001.
- [28] Gengenbach, T. Modellierung und numerische Simulation der menschlichen Atemwege auf Hochleistungsrechnern. Master’s thesis, Universitaet Karlsruhe (TH), April 2009.
- [29] Gengenbach, T. Particle Deposition in the Lungs. In *Proceedings of the 4th EMCL-Workshop “Numerical Simulation, Optimization and High Performance Computing”* (2011), V. Heuveline and G. Thaeter, Eds., vol. 12.
- [30] Gengenbach, T., Heuveline, V., and Krause, M. Numerical Simulation of the Human Lung: A Two-scale Approach. In *BMT 2010 - Reguläre Beiträge (BMT 2010 Reguläre Beiträge)* (Rostock-Warnemünde, Germany, 2010).
- [31] Gengenbach, T., Heuveline, V., Krause, M. J., Mayer, R., and Zimny, S. A Preprocessing Approach for Innovative Patient-specific Intranasal Flow Simulations. EMCL Preprint Series, 2011. No. 7.
- [32] Giles, M. B., and Pierce, N. A. Adjoint equations in CFD: duality, boundary conditions and solution behaviour, 1997.
- [33] Girault, V., and Raviart, P. A. *Finite Element Methods for Navier-Stokes Equations: Theory and Algorithms*. Springer series in computational mathematics. Springer, 1986.
- [34] Glowinski, R. *Numerical Methods for Nonlinear Variational Problems*. Scientific Computation. Springer, 2008.

-
- [35] Glowinski, R., Ciarlet, P. G., and Lions, J. L. *Numerical Methods for Fluids*. No. Part 3 in Handbook of numerical analysis. Elsevier, 2003.
- [36] Golub, G. H., and Van Loan, C. F. *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)(3rd Edition)*, 3rd ed. The Johns Hopkins University Press, October 1996.
- [37] Grandmont, C., Maury, B., and Meunier, N. A viscoelastic model with non-local damping application to the human lungs. *Esaim-Mathematical Modelling And Numerical Analysis-Modelisation Mathematique Et Analyse Numerique* 40, 1 (January 2006), 201–224.
- [38] Gunzburger, M. D. *Perspectives in Flow Control and Optimization*. Advances in Design and Control. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.
- [39] Gunzburger, M. D., and Peterson, J. S. On conforming finite element methods for the inhomogeneous stationary Navier-Stokes equations. *Numerische Mathematik* 42, 2 (1983), 173–194. 10.1007/BF01395310.
- [40] Henderson, A. *ParaView Guide, A Parallel Visualization Application*, 2007.
- [41] Henn, T. Parallel Simulations of Particle Deposition in the Human Respiratory System with Lattice Boltzmann Methods. Master’s thesis, Karlsruhe Institute of Technology (KIT), November 2011.
- [42] Heuveline, V. HiFlow3: a flexible and hardware-aware parallel finite element package. In *Proceedings of the 9th Workshop on Parallel/High-Performance Object-Oriented Scientific Computing - POOSC '10* (New York, NY, USA, 2010), POOSC '10, ACM, pp. 1–6.
- [43] Heuveline, V., Anzt, H., Augustin, W., Baumann, M., Bockelmann, H., Gengenbach, T., Hahn, T., Ketelaer, E., Lukarski, D., Otzen, A., Ritterbusch, S., Rocker, B., Ronnas, S., Schick, M., Subramanian, C., Weiss, J. P., and Wilhelm, F. HiFlow3 – A Flexible and Hardware-Aware Parallel Finite Element Package. EMCL Preprint Series, 2010. No. 6.
- [44] Heuveline, V., and Krause, M. J. OpenLB: Towards an Efficient Parallel Open Source Library for Lattice Boltzmann Fluid Flow Simulations. vol. 9 of *PARA, International Workshop on State-of-the-Art in Scientific and Parallel Computing*. accepted.
- [45] Heuveline, V., and Latt, J. The OpenLB project: An open source and object oriented implementation of lattice Boltzmann methods. *International Journal Of Modern Physics C* 18, 4 (2007), 627–634.
- [46] Hinze, M., Pinnau, R., Ulbrich, M., and Ulbrich, S. *Optimization with PDE Constraints*. Mathematical modelling–theory and applications. Springer, 2009.
- [47] Hollander, P. A., Blonde, L., Rowe, R., Mehta, A. E., Milburn, J. L., Hershon, K. S., Chissasson, J.-L., and Levin, S. R. Efficacy and safety of inhaled insulin (exubera) compared with subcutaneous insulin therapy in patients with type 2 diabetes: results of a 6-month, randomized, comparative trial. *Diabetes Care* 27, 10 (2004), 2356–62.
- [48] Huizinga, E. H., and de Groot, J. A. M. *Functional Reconstructive Nasal Surgery*. Thieme, 2003.
- [49] Hunter, J. D. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering* 9, 3 (2007), 90–95.
- [50] Inthavong, K., Tian, Z., Li, H., Tu, J., Yang, W., Xue, C., and Li, C. A Numerical Study of Spray Particle Deposition in a Human Nasal Cavity. *Aerosol Science and Technology* 40, 11 (2006), 1034–1045.
-

BIBLIOGRAPHY

- [51] Inthavong, K., Tian, Z. F., Tu, J. Y., Yang, W., and Xue, C. Optimising nasal spray parameters for efficient drug delivery using computational fluid dynamics. *Computers in Biology and Medicine* 38, 6 (2008), 713–726.
- [52] Johnson, D. H., Blot, W. J., and Carbone, D. P. *Cancer of the lung: Non-small cell lung cancer and small cell lung cancer*, 4 ed. Clinical Oncology. Churchill Livingstone Elsevier, Philadelphia, Pa, 2008, ch. 76.
- [53] Kee, R. J., Coltrin, M. E., and Glarborg, P. *Chemically reacting flow: theory and practice*. Wiley-Interscience, 2003.
- [54] Kelly, J. T., Asgharian, B., Kimbell, J., and Wong, B. Particle Deposition in Human Nasal Airway Replicas Manufactured by Different Methods. Part I: Inertial Regime Particles. *Aerosol Science and Technology* 38, 11 (November 2004), 1063–1071.
- [55] Kleinstreuer, C., and Zhang, Z. Airflow and Particle Transport in the Human Respiratory System. *Annual Review of Fluid Mechanics* 42, 1 (2010), 301.
- [56] Krause, M., Gengenbach, T., and Heuveline, V. Hybrid Parallel Simulations of Fluid Flows in Complex Geometries: Application to the Human Lungs. In *Euro-Par 2010 Parallel Processing Workshops*, M. Guarracino, F. Vivien, J. Träff, M. Cannatoro, M. Danelutto, A. Hast, F. Perla, A. Knüpfer, B. Di Martino, and M. Alexander, Eds., vol. 6586. Springer Berlin / Heidelberg, 2011, pp. 209–216.
- [57] Krause, M. J., Gengenbach, T., Zimny, S., Mayer, R., and Heuveline, V. How to Breathe Life into CT-Data. *Computer Aided Medical Engineering*, 4 (2011), 29–33.
- [58] Kunsch, K., and Kunsch, S. *Der Mensch in Zahlen: eine Datensammlung in Tabellen mit über 20000 Einzelwerten*. 3. Spektrum, Akad. Verl., Heidelberg, Berlin, 2006.
- [59] Layton, W. J. *Introduction to the numerical analysis of incompressible viscous flows*. Computational science and engineering. Society for Industrial and Applied Mathematics, 2008.
- [60] Lee, D., Park, S. S., Ban-Weiss, G. A., Fanucchi, M. V., Plopper, C. G., and Wexler, A. S. Bifurcation Model for Characterization of Pulmonary Architecture. *The Anatomical Record: Advances in Integrative Anatomy and Evolutionary Biology* 291, 4 (2008), 379–389.
- [61] Li, S., and Petzold, L. Adjoint sensitivity analysis for time-dependent partial differential equations with adaptive mesh refinement. *Journal of Computational Physics* 198, 1 (2004), 310–325.
- [62] Loth, E. *Computational Fluid Dynamics of Bubbles, Drops and Particles*. Cambridge University Press, Cambridge (2009).
- [63] Mao, X., Blackburn, H. M., and Sherwin, S. J. Optimal inflow boundary condition perturbations in steady stenotic flows. *Journal of Fluid Mechanics* (2012).
- [64] Mauroy, B., Filoche, M., Weibel, E., and Sapoval, B. An optimal bronchial tree may be dangerous. *Nature* 427 (February 2004), 633–636.
- [65] Merkus, H. G. *Particle Size Measurements: Fundamentals, Practice, Quality*. Particle technology series. Springer, 2008.
- [66] Michel, M. *Howto Use Octave Functions in C/C++ Programs*, 2007.
- [67] Nadarajah, S. K., and Jameson, A. *A Comparison of the Continuous and Discrete Adjoint Approach to Automatic Aerodynamic Optimization*, 2000.

-
- [68] Otten, E. W. Take a breath of polarized noble gas. *Europhysics News* 35, 1 (2004), 16–20.
- [69] Palmer, T. N., Buizza, R., Molteni, F., Chen, Y.-Q., and Corti, S. Singular Vectors and the Predictability of Weather and Climate. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 348, 1688 (1994), 459.
- [70] Petzold, L., Li, S., Cao, Y., and Serban, R. Adjoint sensitivity analysis for differentialalgebraic equations: The adjoint DAE system and its numerical solution. *SIAM J. Sci. Comput* 149, 24 (2003), 1076–1089.
- [71] Pierce, R., Johns, D. P., and Campaign, N. A. *Spirometry: The Measurement and Interpretation of Ventilatory Function in Clinical Practice*. National Asthma Council Australia, 1995.
- [72] Polak, A. G., and Lutchen, K. R. Computational model for forced expiration from asymmetric normal lungs. *Ann Biomed Eng* 31, 8 (2003), 891–907.
- [73] Quarteroni, A., Sacco, R., and Saleri, F. *Numerical Mathematics*. Texts in Applied Mathematics Series. Springer, 2010.
- [74] Quarteroni, A., and Valli, A. *Numerical approximation of partial differential equations*. Springer series in computational mathematics. Springer-Verlag, 1994.
- [75] Raabe, O. G. *Tracheobronchial geometry: human, dog, rat, hamster—a compilation of selected data from the project respiratory tract deposition models*. U.S. Energy Research and Development Administration, Division of Biomedical and Environmental Research : for sale by the Supt. of Docs., U.S. Govt. Print. Off., 1976.
- [76] Ronnas, S., Gengenbach, T., Ketelaer, E., and Heuveline, V. Design and Implementation of Distributed Meshes in HiFlow3. In *Competence in High Performance Computing 2010*, C. Bischof, H.-G. Hegering, W. E. Nagel, and G. Wittum, Eds. Springer Berlin Heidelberg, 2012, pp. 61–71.
- [77] Saad, Y. *Numerical methods for large eigenvalue problems*. Algorithms and architectures for advanced scientific computing. Manchester University Press, 1992.
- [78] Se, M., Inthavong, K., and Tu, J. Unsteady particle deposition in a human nasal cavity. In *Proceedings of the Seventh International Conference on CFD in the Minerals and Process Industries* (dec 2009), CSIRO, pp. 1–6.
- [79] Shepard, D. A two-dimensional interpolation function for irregularly-spaced data. In *ACM '68 Proceedings of the 1968 23rd ACM national conference* (1968), ACM New York, NY, USA, pp. 517–524.
- [80] Siekmeier, R., and Scheuch, G. Inhaled insulin—does it become reality? *Journal of physiology and pharmacology* (2008).
- [81] Sohr, H. *The Navier-Stokes equations: An Elementary Functional Analytic Approach*. Birkhäuser advanced texts: Basler Lehrbücher. Birkhäuser Verlag, 2001.
- [82] Steinman, D., and Antiga, L. VMTK - Vascular Modeling Toolkit. Webpage, 2008.
- [83] Subramanian, C. *High Performance Computing for Stability Problems - Applications to Hydrodynamic Stability and Neutron Transport Criticality*. PhD thesis, Karlsruhe Institute of Technology (KIT), February 2011.
- [84] Temam, R., and Miranville, A. *Mathematical modeling in continuum mechanics*. Cambridge University Press, 2000.
-

BIBLIOGRAPHY

- [85] The HDF Group. Hierarchical data format version 5, 2000-2010., 2012. <http://www.hdfgroup.org/HDF5>.
- [86] Thomas, P. J. On the influence of the Basset history force on the motion of a particle through a fluid. *Physics of Fluids A: Fluid Dynamics* 4, 9 (1992), 2090.
- [87] Valentin, J. Basic Anatomical and Physiological Data for Use in Radiological Protection: Reference Values - ICRP Publication 89. *Annals of the ICRP* 32 (2002), 1–277.
- [88] Verfürth, R. Error estimates for a mixed finite element approximation of the Stokes equations. *RAIRO Anal. Numér.* 18 (1984), 175–182.
- [89] WebMD. Lung Function Tests, 2011. <http://www.webmd.com/lung/lung-function-tests>.
- [90] Weibel, E. R. *Morphometry of the Human Lung*. Springer Verlag and Academic Press, Berlin, New York, 1963.
- [91] Weibel, E. R., Sapoval, B., and Filoche, M. Design of peripheral airways for efficient gas exchange. *Respir Physiol Neurobiol* 148, 1-2 (2005), 3–21.
- [92] West, J. B. *Respiratory Physiology: The Essentials*. Point (Lippincott Williams and Wilkins) Series. Wolters Kluwer Health/Lippincott Williams & Wilkins, 2008.
- [93] Wikimedia Commons. The Respiratory System. <http://commons.wikimedia.org/wiki/File>
- [94] Wikipedia. Spirometer, 2011. <http://de.wikipedia.org/wiki/Spirometer>.
- [95] Wikipedia. Spirometry, 2011. <http://en.wikipedia.org/wiki/Spirometry>.
- [96] Yap, C., and Dubé, T. *The Exact Computation Paradigm*, 1994.
- [97] Zeidler, E. *Applied functional analysis*, vol. 2: Main principles and their applications of *Applied mathematical sciences*; 109. Springer, New York, 1995.