Fractal Lung

In-silico experiments of single and multiple-breath gas washout

Version 1.0.0

David Hasler

Zürich
November 2018

Contents

Nomenclature

Symbols and indices used in the text are listed in the first and their description in the second column. The optional third column specify the units (meter → m, second → s, kilogram → kg, Newton → N = kg m s^{-2}, Pascal → Pa = N m^{-2}, Joule → J = N m, Watt → W = J/s, pixel → px). Vectors and matrices are typed in bold letters (e.g. vector $\mathbf{v}$, matrix $\mathbf{M}$). Variables are only listed with their indices if the index is required for a clear distinction.

Roman upper case

D	diffusion coefficent	$m^2\ s^{-1}$
Q	flow rate in airway	$m^3\ s^{-1}$
Q_t	flow rate in terminal duct-like airway	$m^3\ s^{-1}$
R	flow resistance in	$m^3\ s^{-1}$
S_t	cross-section in terminal duct-like airway	m^2
$S_{lb}(x,t)$	cross-section in trumpet lobule	m^2
T	flow transmissibility in airway $1/R$	$m^3\ s^{-1}$
$V_{lb}(t)$	trumpet lobule volume	m^3
V_{lb}^0	trumpet lobule residual volume	m^3
$\tilde{V}_{lb}(t)$	trumpet lobule dynamic volume	m^3

Roman lower case

$c(x,t)$	normalized gas concentration	[–]
d	airway diameter	m
d_{Lim}	limit diameter for duct-like airways	m
l	airway length	m
n_1, n_2	lobule geometry power-law exponents	
$p_i(t)$	pressure in airway node i	Pa
p_1, p_2	lobule geometry power-law coefficients	
$p_{pl}(t)$	pleural pressure	Pa
r	asymmetry parameter (airway scheme)	[–]
t	time	[s]
u	flow velocity (average over airway cross-section)	[–]
x	airway streamwise coordinate	[–]
z	airway generation	[–]

Greek upper case

Greek lower case

β	shape parameter for lobule constitutive law
γ	non-linearity parameter for lobule constitutive law
ϕ	modification parameter for lobule compliance
η	reduction rate (airway scheme)
κ	airway dimension ratio
τ	modification parameter for lobule resistance
θ	modification parameter for lobule residual size

Dimensionless variables

Re	Reynolds number
Wo	Womersley number
Pe	Peclet number

Abbreviations

DTG-SBW	double tracer gas single breath washout
FRC	functional residual capacity
LCI	Lung clearance index
MBW	multiple-breath washout
SBW	single breath washout
TSR	time-step refinement

Introduction

The lung model introduced here serves the purpose of simulating multiple-breath washout (MBW) and single breath washout (SBW), two lung function test commonly performed in pediatric pneumology (Robinson et al., 2013; Singer et al., 2013). Inert gas washouts with one tracer gas (e.g. N_2-MBW) or two tracer gases (SF_6-He double tracer gas SBW) can be simulated.

At the core of the lung model are two solvers: A lumped parameter model (also referred to as 0D model) for the ventilation within the lung, and a 1D network model for gas transport. The morphology of the model is defined by a tree-like network of bifurcating airways (mainly for the conducting airways), ending in trumpet-like compliant airway units. In mathematical terms, the ventilation solves a linear system for the pressure p in each bifurcation node and in the pleural gap, and in subsequent step solves for the flow rates Q between two nodes, at discrete time steps. The 1D model solves an advection-diffusion transport equation for one (or optional two) scalar (normalized) quantity c on 1D grid along all airways units. The two solvers are coupled through the flow-rate Q resulting from the ventilation lumped parameter model and which is used to derive the advection velocity u used in the transport equation. A set of model parameters, defining the material and structural properties of the lung, are at hand, most of which can be modified through dedicated tables and input files.

The current model serves well to put into relation, structural and mechanical properties and variations/modification of these, with features of the gas washout profile. In that sense, effects of certain types of disease and conditions on the lung function can be simulated and analysed in a qualitative manner. However, with the current state of the model, it is not possible to simulate all types of airway disease and analyse the their outcomes in an absolute and quantitative manner.

Chapter 1

Model Description

1.1 Morphology

The airway morphology is represented by a generic network of straight branching ducts, which terminate in trumpet-like compartments. In that sense, straight ducts are used to represent the bigger, non-compliant airways, where convective transport is dominant. The dimensions of the trachea and the two main bronchi are defined by anatomical data described by Weibel (1963). In addition, the functional residual capacity (FRC) that is, the air volume that remains in the lung after a tidal expiration, serves as scaling factor for the trachea and therefore accounts for teh scaling of the entire airway network. The length of the scaled trachea is defined as

$$l_0 = l_{0W} \left(\frac{\mathrm{FRC}}{\mathrm{FRC_W}} \right)^{1/3}, \tag{1.1}$$

Where the subscript W indicates a reference quantity from the data by Weibel (1963). For airways past the main bronchi, the scheme for a regular branching asymmetry introduced by Majumdar et al. (2005) was applied. In this scheme, each duct-like airway bifurcates in a major and a minor daughter. In the airway network every parent, minor and major daughter share a common node. The dimensions of the daughter airways, namely their diameter and length, are different fractions of the dimension of their common parent duct,

$$\begin{aligned} d_{z+1_{\mathrm{maj}}} &= d_z \kappa_{\mathrm{maj}} \quad \text{with} \quad \kappa_{\mathrm{maj}} = (1-r)^{1/\eta} \\ d_{z+1_{\mathrm{min}}} &= d_z \kappa_{\mathrm{min}} \quad \text{with} \quad \kappa_{\mathrm{min}} = r^{1/\eta}. \end{aligned} \tag{1.2}$$

Here, d_z is the diameter of a duct at generation z, and r and η denote the asymmetry parameter and reduction rate, respectively. The same scheme was used to define the length of airways. In their study, Majumdar et al. (2005) presented values $r = 0.326$ and $\eta = 2.97$ for which the resulting structure best represents the human lung on a statistical basis. In the present model this bifurcation scheme is applied until the diameter of a duct is smaller than a limit diameter $d_{\mathrm{lim}} = 1.8$ mm, typically chosen to represent bronchi at the 8th-12th generation. The last generation of duct-like airways depends on the limit diameter and may vary within the model, due to the asymmetric bifurcation scheme. The unit distal to an ending duct constitutes a lobule and is modelled using a trumpet-like compartment (trumpet lobule). The lobule as defined here contains, apart from the acini and the respiratory bronchioles, the last generations of conductive airways. Compared to the duct-like, rigid airway, the trumpet lobule can be interpreted as a compartment with diverging, time-variable cross-section (see Figure 1.1). The number of these trumpet lobules is equal to the number of terminal ducts, which again is determined by the limit diameter and the FRC. In general, each trumpet can have distinct geometrical and mechanical properties. An important constraint, however, is that the total

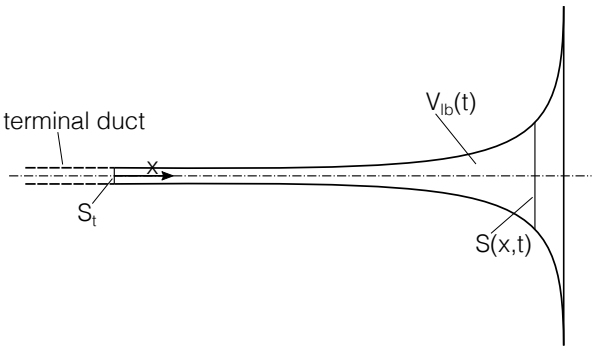

Fig. 1.1: *Schematic representation of a trumpet-lobule with inlet cross-section* S_t*, dynamic cross-section* $S_{lb}(x,t)$ *and volume* $V_{lb}(t)$

volume of the model (duct-like airways and trumpet lobules) under static conditions equals a predefined FRC. In order to simulate the effects of lung structural inhomogeneity on the MBW test, properties of both the stiff and the compliant parts of the model can be modified (see Section 1.3 for more informations on possible model modifications). Figure illustrates a sketch of a model lung showing the relative scales of a network of airways defined by the bifurcation rule (Equation (1.2)). A detailed mathematical description of the trumpet, its geometrical and dynamical properties, can be found in Section 1.2.2.

1.2 Mathematical models

1.2.1 Ventilation

Lung ventilation is simulated by means of a lumped parameter (0-dimensional) model illustrated in Figure 1.2. For the duct-like airways purely resistive elements were used. During normal breathing, airflow in the lung is unsteady and inertia plays a considerable role regarding the flow resistance in bigger airways until the tenth generation (Kaczka et al., 2011). The theory by Womersley (1957) for pulsatile flow in tubes was therefore applied. In general, the pressure difference between two subsequent nodes with node index i and j in the network reads.

$$p_i - p_j = R_{ij} Q_{ij}, \quad \text{with} \quad R_{ij} = \Re \left\{ \frac{i\omega\rho l_{ij}}{r_{ij}^2 \pi} \left[1 - \frac{2J_1(i^{3/2}\alpha)}{i^{3/2}\alpha J_2(i^{3/2}\alpha)} \right]^{-1} \right\}. \tag{1.3}$$

Here Q_{ij} is the flow rate in direction of the negative pressure gradient, and the hydrodynamic resistance R_{ij} depends on the radius r_{ij} and the length l_{ij} of the conducting airway between two nodes, and on the breath period T_B.

Pressure volume relation in trumpet lobules

The trumpet model, which mainly represents compliant, peripheral airways (lobules) is composed of a nonlinear compliance (elastic pressure, p_{el}) element and a resistance (pressure loss, p_{diss}) element, acting in series between a node with index i, corresponding to a terminal duct, and the pleural gap. The corresponding pressure difference is defined as

$$p_i - p_{pl} = \underbrace{\beta e^{\gamma V_{lb}^0} \left(e^{\gamma \tilde{V}_{lb}} - 1 \right)}_{\text{elastic pressure, } p_{el}} + \underbrace{R_{lb} Q_{lb}}_{\text{dissipation}}, \tag{1.4}$$

where R_{lb}, $V_{lb}(t) = V_{lb}^0 + \tilde{V}_{lb}$ and Q_{lb} denote the flow resistance and the volume of, respectively the flow rate into, a trumpet lobule past a terminal duct-like airway, V_{lb}^0 is the lobule volume at

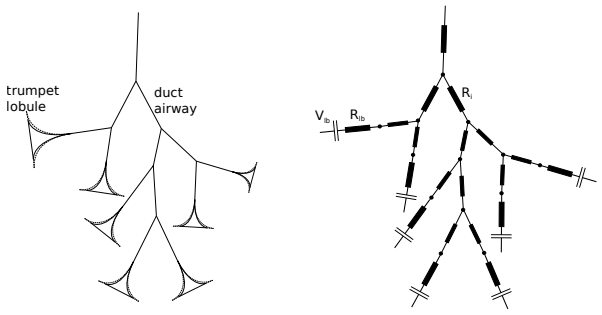

Fig. 1.2: *Left, Network of duct and trumpet-like elements used for the upper and lower airways, respectively. Right, corresponding LPM composed of resistances and compliances.*

FRC and $\tilde{V}_{lb}$ is the dynamic volume during breathing. The shape parameters β and γ are used to define and modify the compliance of the trumpet lobule. More information on the mechanical properties of the model can be found in the Section 1.3.

Mass conservation has to be satisfied in the network. Thus, in each node, the flow rates had to be balanced $\sum Q = 0$, and for the lobule model the flow rate of the terminal duct Q_t equals the change of volume of the lobule, thus

$$Q_t = \frac{dV_{lb}}{dt} \tag{1.5}$$

From these relations, a system of differential algebraic equations (DAE) results. Together with appropriate boundary conditions, the DAE system governs the flow distribution in the ventilation LPM.

From a physiological point of view, during inspiration, airflow from the mouth and the nose moves to the alveolar membrane as a result of the motion of the diaphragm and the thoracic cavity, causing a volume increase and a pressure decrease in the pleural cap. This negative (relative) pressure in the pleural gap causes a pressure gradient across the peripheral lung tissue and along the airways to the mouth and nose. In that sense, obvious boundary conditions for the LPM would be the pleural pressure and the pressure at the mouth. However, pleural pressure data is in general not obtained during clinical routine and therefore not accessible. Instead, flow rate data at the mouth can be easily measured by flowmeters (e.g. in the multiple breath washout test). Imposing boundary conditions for pressure and flow at the mouth allows solving the DAE with unknown pleural pressure. To this end, a spatially uniform pleural pressure is assumed, which varies over time according to the prescribed volume change and the material law of the lung tissue. A detailed description of the numerical implementation of the LPM is provided in Section 1.4.1.

Trumpet lobule geometry

For the trumpet model representing the lobules and their peripheral airways, a model for the total cross-section of the trumpet lobule $S_{lb} = S_{lb}(x, t)$, as well as for the mean advection velocity $u_{lb}(x, t)$ had to be derived. From the LPM the flow rate at the inlet of each trumpet $Q_t(t)$ and the total volume of the trumpet $V_{lb}(t) = \int_0^t Q_t dt + {V_{lb}}^0$ was known. The initial volume of the trumpet lobule, $V_{lb}(t = 0) = {V_{lb}}^0$, followed from FRC-based scaling of the lung model. Assuming a uniform homothety ratio of $\kappa = 0.85$ Weibel (1963) for airways lumped in

a trumpet lobule, the total change of cross-section along the streamwise coordinate x can be described as

$$S(x,0) = S_t \hat{\kappa}^{z(x)}, \quad \text{with} \quad \hat{\kappa} = 2\kappa^2 \tag{1.6}$$

where S_t is the cross-section of the terminal duct and $z(x)$ is the generation at absolute position x with respect to the inlet of the trumpet lobule, both with respect to the terminal duct where $z = x = 0$. Considering l_t to be the length of the terminal duct, the cumulative length at generation z (with respect to the inlet of the lobule) would be $\sum_{k=1}^{z} l_t \kappa^k$. The limes of this sum for $z \to \infty$ is

$$\lim_{z\to\infty} \sum_{k=1}^{z} l_t \kappa^k = l_t \kappa/(\kappa - 1)) =: L. \tag{1.7}$$

From these relations, an expression for the the generation in function of the distance to the inlet of the trumpet can be computed,

$$z(x) = \frac{\log\left[x\,\frac{\kappa-1}{\kappa\, l_t} + 1\right]}{\log(\kappa)}. \tag{1.8}$$

Using equation (1.6) together with (1.8) for further treatment of the lobule model becomes a rather cumbersome task. We therefore sought a model $S_{lb}(x,t)$, which approximates Equation (1.6) but allows to derive an analytical expression for $u_{lb}(x,t)$. To this end, we used a power law of the form

$$S_{lb}(x,t) = p_1 x^{n_1} + p_2 x^{n_2} + S_t, \tag{1.9}$$

with $n_1, n_2 = 16, 2$, where the coefficients p_1 and p_2 were defined such that the S_{lb} intersects with Equation (1.6) at a chosen generation $z^\star$, and the prescribed initial volume V_{lb}^0 of the trumpet lobule is obtained for a given length l_{lb} of the trumpet lobule. The coefficients were thus defined by the following

$$\begin{bmatrix} x^{\star n_1} & x^{\star n_2} \\ \frac{l_{lb}^{n_1+1}}{n_1+1} & \frac{l_{lb}^{n_1+1}}{n_1+1} \end{bmatrix} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} S^\star - S_t \\ V_{lb}^0 - S_t\, l_{lb} \end{pmatrix} \tag{1.10}$$

with $S^\star = S_{lb}(x^\star, 0)$ and $z^\star = z(x^\star)$.

In Figure 1.3, a comparison between equations (1.6) and (1.9) shows a good agreement. The actual length of the trumpet model was defined based on the cumulative length

$$l_{lb} = \sum_{k=1}^{z_{max}} l_t \kappa^k \tag{1.11}$$

where $l_{lb} =\approx 0.98L$ follows for a lobule with $z_{max} = 16 - 18$ generation. The system (1.10) determined the initial (or end-expiratory) trumpet shape $S_{lb}(x,0)$. During breathing the cross-section widens. Therefore the first coefficient of the power law was set to be a time-depending parameter $p_1 = p_1(t)$. After the initial shape of the trumpet lobule was defined (using Equations (1.10)), $p_1(t)$ was updated according to the lobular volume $V_{lb}(t)$

$$V_{lb}(t) = \int_0^{l_{lb}} S_{lb}(x,t)\, dx \tag{1.12}$$

$$\Rightarrow p_1(t) = \frac{n_1+1}{l_{lb}^{n_1+1}} \left[V_{lb}(t) - \frac{l_{lb}^{n_2+1}}{n_2+1} p_2 - S_t l_{lb} \right] \tag{1.13}$$

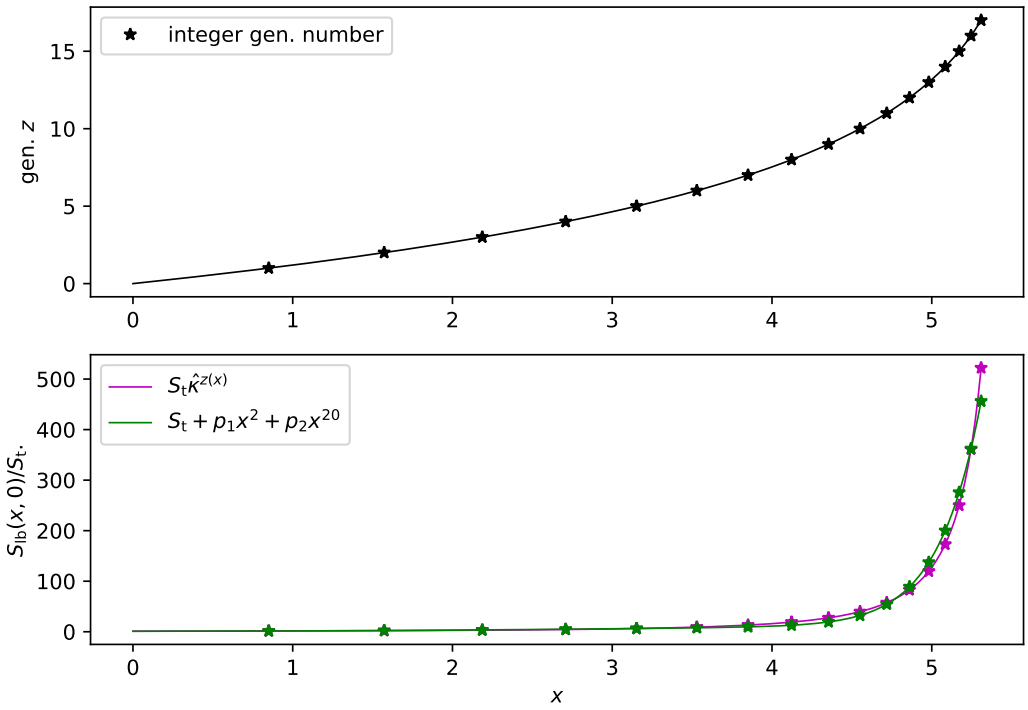

Fig. 1.3: *Above: the generation z plotted as function of the cumulative length or distance x with $l_t = d_t = \mathrm{unity}$. The solid line indicates the function described by equation (1.8). Below: The cross-section as it follows from equation (1.6) (magenta) and the model (1.9) (green) where p_1 and p_2 were defined to yield the same integral as the exponential law and intersects at generation $z^\star = 5$ ($z = 0$ at trumpet inlet).*

This determined the shape of the trumpet acinus at any time. An important feature of the model (1.9) is the major contribution of peripheral airway (where x is close to l_{lb}) to the overall expansion of the lung.

1.2.2 Gas transport

To model the time-variable distribution of tracer gases in the lung, typically nitrogen (N_2) or sulfur-hexafluoride (SF_6), a one-dimensional transport equation for a scalar variable representing the concentration $c = c(x, t) \in [0, 1]$ (normalised with the maximum value), is solved with a finite difference method (see Section 1.4.2). Simulation of MBW involves the computation of gas concentration in all model airways over multiple breath periods. Moreover, only inert gases were considered. For this case, the dominant transport mechanisms are advection by the carrier gas (ambient air) and diffusion. The molecular mass of N_2 as the tracer gas is much smaller than that of the carrier gas. Therefore, the assumption of a diluted gas is made, where molecular diffusion of a tracer gas in a carrier gas depends only on the physical properties of the two gases involved Cussler (2009). As a further assumption, these properties (temperature, molecular velocity, etc.) and therefore the diffusivity of a tracer gas are considered to be constant in the lung and throughout the breath cycle. The coordinate (here denoted with x), along which the transport process takes place, is the center line of a model airway, hence along each duct-like airway and along each lobule trumpet. The 1D approximation entails consideration of averaged (lumped) quantities within a cross-section, or in case of the trumpet model, within the total

cross-section of all airways at a given position x. Advection-diffusion processes, taking place in a pipe-like geometry are subject to high radial velocity gradients. As a consequence, high-concentration gas is transported (advected) along low-concentration gas increasing the interface of high concentration gradients and therefore increasing the average diffusivity within a cross-section. In the human lung this phenomena can take place on different scales: In the upper airway, where the Reynolds number $Re = udp/\mu$ is in the order of $10'000$, turbulent flow strongly enhances mixing. In smaller airways ($Re < 2'000$), Poiseuille flow with a parabolic velocity profile can be assumed. Enhanced diffusion due to high velocity gradients is usually modelled based on the concept of Taylor dispersion, where the local diffusion coefficient is a function of the local Peclet number Cussler (2009),

$$\hat{D} = D\left(1 + \frac{1}{192}Pe^2\right). \tag{1.14}$$

Here, $\hat{D}$ and D are the effective and molecular diffusion coefficient, respectively, and Pe is the Peclet number defined with the (local) mean velocity u and the (local) airway diameter d.

In the lung model, the gas concentration evolves in a network of one-dimensional domains. To this end, the transport equation is solved in each of these sub-domains separately (domain decomposition approach), applying a coupling of concentration values from different domains in bifurcation nodes. A transport equation of the following general form is considered

$$\frac{\partial(Sc)}{\partial t} + \frac{\partial F}{\partial x} = 0, \quad \text{with the flux} \quad F = S_{ad}uc - S\hat{D}\frac{\partial c}{\partial x}. \tag{1.15}$$

Here, we distinguished between the total cross-section $S(x,t)$ and the cross-section S_{ad} where advection with the carrier gas velocity $u(x,t)$ takes place. At bifurcations the concentration flux has to be conserved $\sum F = 0$. The geometry of the trumpet lobule $S_{lb}(x,t)$ (derived in Section 1.2.1) allows a more specific form of Equation (1.15) to be stated for the gas transport within trumpet lobules with non-constant cross-section (see upcoming section).

Modified advection-diffusion equation

The derivation of the transport equation (Equation 1.15) is different for duct-like and trumpet-like airways in the lung model. In the first case, the advection velocity directly results from the solution of the ventilation LPM (solving for node pressures p_i and p_j, and the flow rate Q_{ij} between the nodes i and j) and is constant within a duct, $u_{ij} = Q_{ij}/S_{adij}$. Furthermore, ducts are stiff and hence $S = S_{ad} = \text{const}$. Omitting the subscripts referring to nodes for briefity, the transport equation for a species concentration $c = c(x,t)$ in a duct-like airway can be written as an advection-diffusion-equation

$$\frac{\partial c}{\partial t} + u\frac{\partial c}{\partial x} - \hat{D}\frac{\partial^2 c}{\partial x^2} = 0. \tag{1.16}$$

For a trumpet lobule with cross-section $S_{lb}(x,t)$, an expression for the advection velocity in the trumpet model can be found introducing the 1D continuity equation for a non-constant cross-section,

$$\frac{\partial S_{lb}}{\partial t} + \frac{\partial Q_{lb}}{\partial x} = 0. \tag{1.17}$$

Using the model $S_{lb}(x,t)$ introduced above, the first term becomes

$$\frac{\partial S_{lb}}{\partial t} = x^{n_1}\frac{\partial p_1}{\partial t} = x^{n_1}\frac{\partial p_1}{\partial V_{lb}}\frac{\partial V_{lb}}{\partial t} = x^{n_1}\frac{n_1+1}{l_{lb}^{n_1+1}}\,Q_t \tag{1.18}$$

Integrating the continuity equation then yields an equation for the flow rate

$$\begin{aligned} Q_{lb}(x,t) &= -\int \frac{\partial S_{lb}}{\partial t} dx \\ &= Q_t(t)\left[1 - \frac{x^{n_1+1}}{l_{lb}^{n_1+1}}\right] \end{aligned} \tag{1.19}$$

The term in brackets is 1 for $x = 0$ and 0 for $x = l_{lb}$, which is in agreement with the condition of zero outflow at the end of the trumpet and follows from the constraint $Q_t = dV_{lb}/dt$. The advection velocity in the trumpet is related to the flow rate, $u_{lb}(x,t) = Q_{lb}(x,t)/S_{ad}$, where for the advection cross-section the same model as for S_{lb} is used but half the initial lobular initial volume is used for the determination of the coefficients p_1 and p_2.

The transport equation (Equ. 6 and 7 in the main article) together with the continuity equation (1.17) can be rearranged for non constant $S_{lb}(x,t)$ and $u_{lb}(x,t)$ yielding an advection-diffusion equation with modified advection velocity

$$\frac{\partial c}{\partial t} + \underbrace{\left[\frac{S_{ad}}{S_{lb}} u_{lb} - \frac{\hat{D}}{S_{lb}}\frac{\partial S_{lb}}{\partial x}\right]}_{=\hat{u}} \frac{\partial c}{\partial x} - \hat{D}\frac{\partial^2 c}{\partial x^2} = 0 \tag{1.20}$$

In summary, equations (1.16) and (1.20) describe the gas transport in duct, respectively trumpet-type airway models and equations (1.9) and (1.19) together with (1.13) describe the dynamics of the trumpet model. The system is coupled with the ventilation LPM through the average flow velocity in each duct and into each trumpet. The numerical solution method for the LPM and for the advection-diffusion equation are presented in Section 1.4.

1.3 Model modifications

An important application of the lung model is to simulate the effects of different functional and structural inhomogeneities on the gas transport process within the lung. I.e. non-uniform tissue compliance, non-uniform size of airway units (e.g. lobules, acini), and non-uniform flow resistance in small airways, affecting different parts of the lung.

Lobule Compliance: Tissue compliance can be tuned with the modification parameter ϕ, which is used to define the shape parameter β and γ in Equation (1.4). To this end, a mean lobular tidal volume $V_{lb}^{TV} = V_T/N_{lb}$ is considered, where V_T is the tidal volume, and N_{lb} is the total number of trumpet lobules. Then the nominal lobule volume range is defined $[V_{lb}^0, V_{lb}^0 + V_{lb}^{TV}]$. Furthermore, a reference pressure amplitude $p_{lb}^{TV} = 1500$ Pa is considered, which corresponds to a typical pressure range in the pleural gap (pleural pressure). For a given pressure load p_{lb}^{TV}, the elastic component of the lobule mechanics can be considered stiff (reduced compliance) if its maximum dynamic volume is smaller than V_{lb}^{TV}, and soft (increased compliance) otherwise. In Figure 1.4 (a), the elastic pressure curve (see Equation 1.4) for a lobule is shown assuming different values of ϕ, which effectively modify the curve via the shape parameter β as

$$\beta = \frac{p_{lb}^{TV}}{e^{\gamma V_{lb}^0}(e^{\gamma\phi V_{lb}^{TV}} - 1)}, \tag{1.21}$$

The remaining shape parameter γ accounts for the non-linearity of the elastic pressure curve (see Figure 1.4 (b) and was computed such that the elastic pressure curve intersects with the point $(V_{lb}^0 + 3/4\phi V_{lb}^{TV},\ 1/4 p_{lb}^{TV})$ in the volume-pressure plane. For this point, the resulting curve qualitatively lies between a linear and an almost right-angled curve.

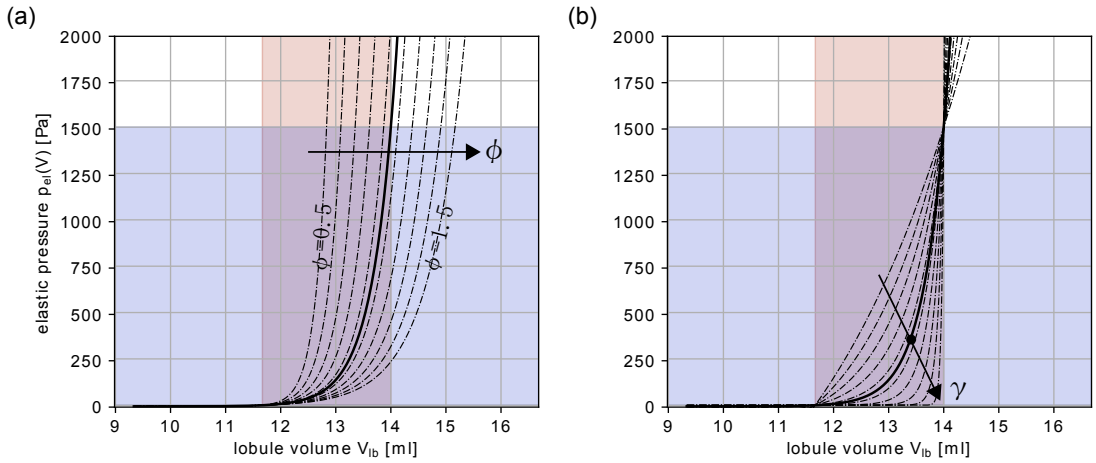

Fig. 1.4: *Elastic pressure* p_{el} *as defined in Equation 1.4. The purple area indicates the reference pleural pressure range* $[0, p_{lb}^{TV}]$ *and the peach area indicates the nominal lobule volume range* $[V_{lb}^0, V_{lb}^0 + V_{lb}^{TV}]$. *In panel (a) the influence of the compliance modification parameter* $\phi \in [0.5, 1.5]$ *is shown. For* $\phi = 1$ *(solid line) the elastic pressure curve intersects in the point* $(V_{lb}^0 + V_{lb}^{TV}, p_{lb}^{TV})$. *In panel (b) the effect of the non-linearity parameter* γ *is shown. The parameter is chosen such that the elastic pressure curve intersects with the point* $[V_{lb}^0 + 3/4\phi V_{lb}, 1/4 p_{lb}^{TV}]$ *(black dot). Note that for all graphs in panel (b)* $\phi = 1$ *was used.*

Lobule Volume: The modification parameter θ is used to modify the residual volume of a trumpet lobule as $V_{lb_{mod}}^0 = \theta V_{lb}^0$.

Lobule Resistance: The hydrodynamic flow resistance within a specific trumpet-lobule, i.e. R_{lb}, can be changed via the modification parameter τ, causing a modified pressure loss as $p_{diss_{mod}}$. Here, the $\tau > 1$ acts as a multiplier of the lobular flow resistance, hence modelling the obstruction of airways in a lobule, e.g. due to mucus formation.

1.4 Numerical implementation

1.4.1 Ventilaton lumped parameter model

Duct-like airways

Regarding airflow in duct-like airways, we applied Womersley's theory for pulsatile flow in tubes (Womersley, 1957). For this case, the pressure difference between two subsequent nodes with index i and j can be computed from the analytical expression given in Equation 1.3. To save computation time in the simulations, the resistance R_{ij} is interpolated from a previously computed table for different diameters and breath periods.

Compliant trumpet lobules

For the constitutive law of the trumpet lobules, we used used an exponential relation between the elastic pressure and the (tidal) volume of the lobule. Additionally a pressure loss was introduced due to the hydrodynamic flow resistance in the lobule. Referring to Figure 1.5 for notations, the total pressure drop of a trumpet lobule can be stated as

$$p_6 - p_{pl} = R_{lb} Q_t + \beta e^{\gamma V_{lb}^0} \left(e^{\gamma \tilde{V}_{lb}} - 1 \right), \tag{1.22}$$

where p_6 is the pressure at the end of a duct-like airway feeding into a trumpet-lobule, p_{pl} is the pressure in the pleural cap, R_{lb} is the total hydrodynamic resistance within the trumpet lobule, $Q_t = dV_{lb}/dt$ is the flow rate at the inlet of the lobule, and V_{lb} is the lobule volume. The parameters β and γ were determined by the choice of a specific material law with its characteristic compliance and non-linearity (Equ. 3 in the main article). The lobule volume can be decomposed into a static part, V_{lb}^0 determined by the number of trumpet lobules and the functional residual capacity (FRC), and a dynamic part which varies during tidal breathing

$$V_{lb} = V_{lb}^0 + \tilde{V}_{lb}. \tag{1.23}$$

Differentiation of equation (1.22) with respect to time and rearranging yields

$$\frac{dQ_t}{dt} = \frac{1}{R_{lb}} \left[-\gamma\beta e^{\gamma V_{lb}} Q_t + \left(\frac{dp_6}{dt} - \frac{dp_{pl}}{dt} \right) \right]. \tag{1.24}$$

Replacing the flow rate in the terminal duct with $Q_t = T_{56}(p_5 - p_6)$, with T_{56} the transmissibility (inverse of resistance) in the terminal conducting airway between nodes 5 and 6, the differential equation (1.24) can be written in the form

$$\frac{dp_6}{dt} - \frac{dp_{pl}}{dt} - R_{lb}T_{56}\left(\frac{dp_5}{dt} - \frac{dp_6}{dt} \right) = T_{56}\gamma\beta e^{\gamma V_{lb}} (p_5 - p_6) \tag{1.25}$$

Numerical solution

We applied the trapezoidal rule for the numerical integration of Equation (1.25).

$$\begin{aligned} (p_6^n - p_6^{n-1}) - (p_{pl}^n - p_{pl}^{n-1}) - R_{lb}T_{56}\left[(p_5^n - p_5^{n-1}) - (p_6^n - p_6^{n-1}) \right] \\ = \frac{\Delta t}{2} T_{56}\gamma\beta e^{\gamma V_{lb}^\star} \left[(p_5^n + p_5^{n-1}) - (p_6^n + p_6^{n-1}) \right] \end{aligned} \tag{1.26}$$

Here, $(.)^n$ denotes variables at the new, updated, discrete time $t = n\Delta t$, with Δt the time step. The lobular volume at timestep n is estimated using an explicite Euler formulation $V_{lb}^\star = V_{lb}^{n-1} + \Delta t Q_t^{n-1}$. For implementation purposes, Equation (1.26) is reformulated as

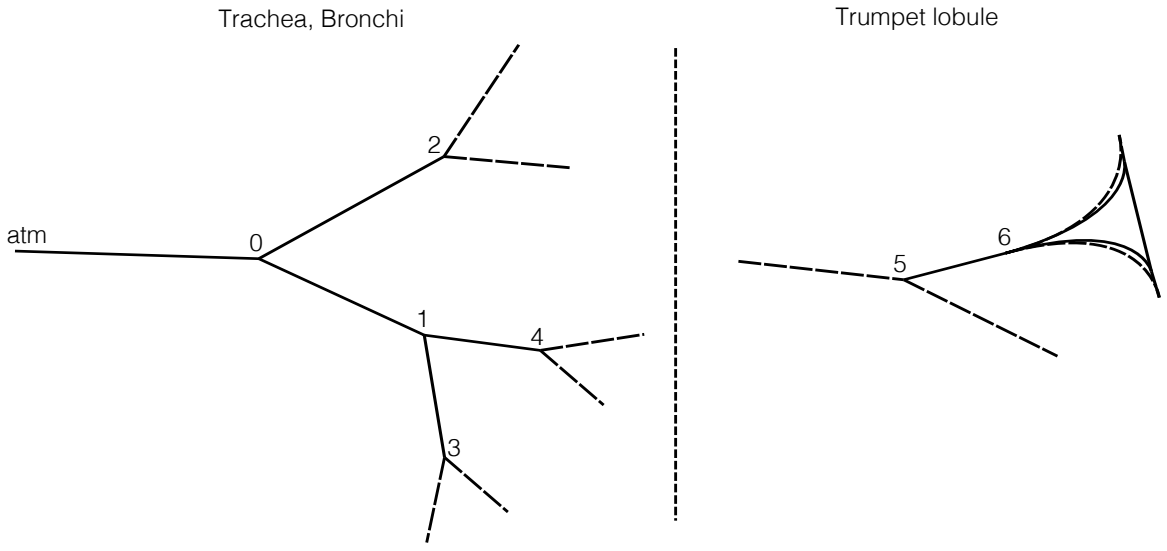

Fig. 1.5: *Reduced size network of duct-like airways and trumpet lobule at the periphery*

$$-K_1 p_5^n + (1 + K_1) p_6^n - p_{pl}^n = K_2 \tag{1.27}$$

where the two coefficients K_1 and K_2 are computed based on available information from the former time step.

$$K_1 = R_{lb} T_{56} + \frac{\Delta t}{2} T_{56} \gamma \beta e^{\gamma V_{lb}^\star} \tag{1.28}$$

$$K_2 = -R_{lb} T_{56} p_5^{n-1} + (1 + R_{lb} T_{56})\, p_6^{n-1} - p_{pl}^{n-1} + \frac{\Delta t}{2} T_{56} \gamma \beta e^{\gamma V_{lb}^\star} \left(p_5^{n-1} - p_6^{n-1} \right) \tag{1.29}$$

At bifurcations of duct-like airways, the flow balance has to be satisfied in each time step. Referring to node 1 in Figure 1.5 this reads

$$\begin{aligned} Q_{01}^n - Q_{13}^n - Q_{14}^n &= 0 \\ T_{01}(p_0^n - p_1^n) - T_{13}(p_1^n - p_3^n) - T_{14}(p_1^n - p_4^n) &= 0 \end{aligned} \tag{1.30}$$

subject to boundary conditions imposed at the inlet of the airway tree,

$$T_{tr}(p_{atm} - p_0) = Q_{in}(t) \tag{1.31}$$

The flow rate profile $Q_{in}(t)$ was pre-defined and constant atmospheric pressure was assumed at the inlet. T_{tr} is the transmissibility of the trachea and p_0 the pressure in the first bifurcation. Equations (1.27) and (1.30) can be stated for all bifurcation nodes and lobules. Together with the boundary conditions this yields a system of equations for the pressure values in the bifurcation nodes, p_i, and the pleural pressure, p_{pl}, which had to be solved in each time step. Thereby, for each discrete time-step, a uniform pleural pressure was assumed. To solve the resulting system of equations, we applied an iterative method (bi conjugate gradient stabilized solver) from the EIGEN linear algebra library (eigen.tuxfamily.org).

1.4.2 Gas transport equation

For the transport of an inert gas in the lung two transport mechanism were considered: Advection (in this context also called convection) and molecular diffusion. More information about the underlying assumptions are given in the method section of the main article. The 1D-advection-diffusion equation governs the evolution of the gas concentration $c = c(x, t)$ (normalized with the maximum concentration, $c \in [0, 1]$) in time and along the airway streamwise coordinate x:

$$\frac{\partial c}{\partial t} = - \underbrace{u \frac{\partial c}{\partial x}}_{\text{advection}} + \underbrace{D \frac{\partial^2 c}{\partial x^2}}_{\text{diffusion}} \tag{1.32}$$

In the lung model the velocity u was computed by means of a lumped parameter model (LPM) for the ventilation of all model airway, and the diffusion coefficient D, given the properties of both the carrier gas and the inert tracer gas, was computed using Chapman-Enskog equation. In addition, D was modified to account for Taylor's dispersion (see Cussler (2009)). For the discretization of Equation (1.32) the advection velocity u was assumed to be constant within a duct-like airway. Transport phenomena related to spacial velocity gradients at the bifurcations are not considered. In trumpet lobules, the non-constant cross-section lead to a modified advection velocity $\hat{u}$ (see Section 1.2.2). Otherwise the numerical treatment was the same for duct-like airways and trumpet lobules. A duct or trumpet lobule was discretized into N elements and hence Equation (1.32) is evaluated on $N + 1$ nodes, where N can be different

for each airway. The avection-diffuison equation (1.32) was discretized applying a second order upwind scheme and a second order central scheme for the spatial discretization of the advection term and the diffusion term, respectively, and using a generalized Crank-Nicolsen scheme for the temporal discretization.

$$\frac{\underline{c}^{n+1} - \underline{c}^{n}}{\Delta t} = \theta(-u\underline{\underline{\mathcal{D}_1}} + D\underline{\underline{\mathcal{D}_2}})\underline{c}^{n+1} + (1-\theta)(-u\underline{\underline{\mathcal{D}_1}} + D\underline{\underline{\mathcal{D}_2}})\underline{c}^{n} + \underline{b}^{n} \tag{1.33}$$

Here, $\underline{c}$ is the vector containing the $N+1$ concentration values within the given airway model, $\underline{\underline{\mathcal{D}_1}}$ and $\underline{\underline{\mathcal{D}_2}}$ are finite difference operators for the first, respectively the second spatial derivatives, and $\underline{b}$ are boundary conditions containing information from the neighboring ducts. The superscript n refers to the time step $t_n = n\Delta t$. Equation (1.33) is partly implicite due to the concentration vector considered at the new (updated) time step t_{n+1} appearing on the right-hand side. The coefficient $\theta \in [0,1]$ controls the weight of the implicite information (at t_{n+1}). For all simulations $\theta = 0.7$ was used. Rearranging the discretized advection-diffusion equation yields a linear system for the concentration values at the new time-step $\underline{c}^{n+1}$.

$$\underbrace{\left[1 - \Delta t\theta(-u\underline{\underline{\mathcal{D}_1}} + D\underline{\underline{\mathcal{D}_2}}))\right]}_{=:\underline{\underline{\mathcal{L}}}}\underline{c}^{n+1} = \underbrace{\left[1 + \Delta t(1-\theta)(-u\underline{\underline{\mathcal{D}_1}} + D\underline{\underline{\mathcal{D}_2}}))\right]}_{=:\underline{\underline{\mathcal{R}}}}\underline{c}^{n} + \Delta t\underline{b}^{n} \tag{1.34}$$

Considering the gridsize h of the 1D mesh in a duct or trumpet-like structure, the upwind operator $\underline{\underline{\mathcal{D}_1}}$ was chosen according to the flow direction during inspiration and expiration:

$$\text{if } u \geq 0\text{:}\quad \underline{\underline{\mathcal{D}_1}} = \frac{1}{2h}\begin{bmatrix} 3 & 0 & \dots & \dots & \dots & \dots & 0 \\ -4 & 3 & 0 & \dots & \dots & \dots & \vdots \\ 1 & -4 & 3 & 0 & \dots & \dots & \vdots \\ 0 & 1 & -4 & 3 & 0 & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \dots & 0 & 1 & -4 & 3 & 0 \\ 0 & \dots & \dots & 0 & 1 & -4 & 3 \end{bmatrix}$$

$$\text{if } u < 0\text{:}\quad \underline{\underline{\mathcal{D}_1}} = \frac{1}{2h}\begin{bmatrix} -3 & 4 & -1 & 0 & \dots & \dots & 0 \\ 0 & -3 & 4 & -1 & 0 & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \dots & 0 & -3 & 4 & -1 & 0 \\ \vdots & \dots & \dots & 0 & -3 & 4 & -1 \\ \vdots & \dots & \dots & \dots & 0 & -3 & 4 \\ 0 & \dots & \dots & \dots & \dots & 0 & -3 \end{bmatrix} \tag{1.35}$$

The finite difference operator for the second spatial derivative is given by

$$\underline{\underline{\mathcal{D}_2}} = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & 0 & \dots & \dots & \dots & 0 \\ 1 & -2 & 1 & 0 & \dots & \dots & \vdots \\ 0 & 1 & -2 & 1 & 0 & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \dots & 0 & 1 & -2 & 1 & 0 \\ \vdots & \dots & \dots & 0 & 1 & -2 & 1 \\ 0 & \dots & \dots & \dots & 0 & 1 & -2 \end{bmatrix} \tag{1.36}$$

Boundary conditions

Bifurcation boundary conditions: Referring to the schematic in Figure 1.6, boundary conditions for the airway A can be stated based on the concentrations in the last nodes of airways B & C or of the first nodes in airways D & E, depending on the advection direction. The contributions of each airway B & C or D & E were weighted by the specific flow rates or cross-sections in the corresponding neighboring airways. In the following, we state the bifurcation boundary conditions for A for positive flow ($u_A > 0$), where airways B & C feed into airway A, which feeds into airways D & E, and for negative flow ($u_A < 0$), where airways D & E feed into airway A, which feeds into airways B & C. The bifurcation boundary conditions were implemented by means of ghost-nodes up-stream and down-stream of a given airway. For the considered case the ghost cell concentration values read

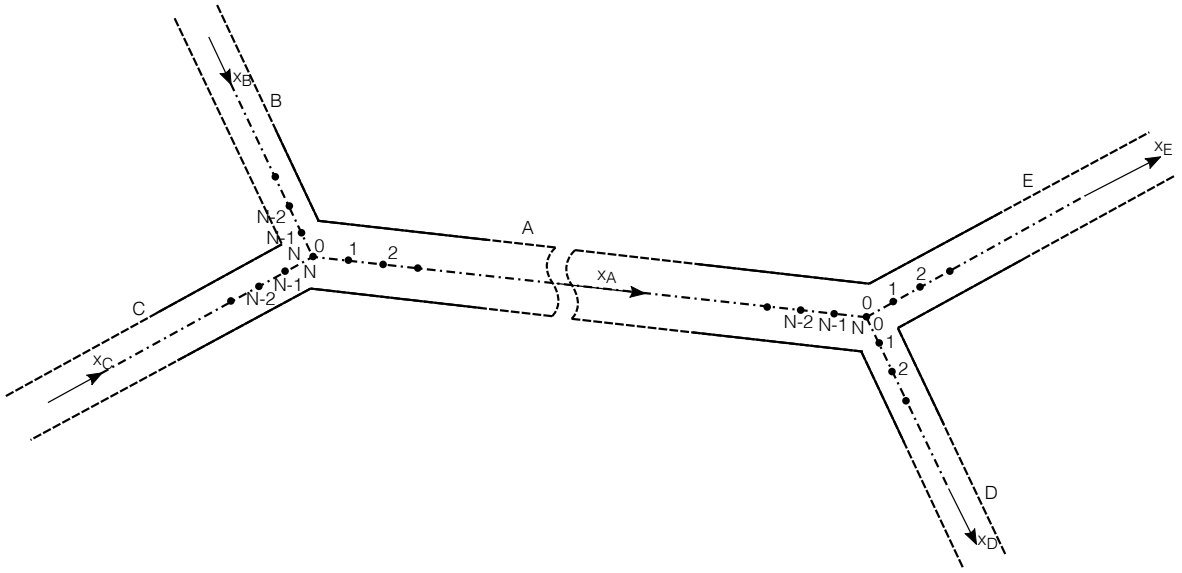

Fig. 1.6: *Bifurcation of duct-like airways with one-dimensional grid for gas transport.*

$$
\begin{aligned}
c^{adv}_{A_{-1}} &= \frac{|Q_B|}{|Q_B|+|Q_C|}c^{\star}_{B_N} + \frac{|Q_C|}{|Q_B|+|Q_C|}c^{\star}_{C_N} \\
c^{adv}_{A_{-2}} &= \frac{|Q_B|}{|Q_B|+|Q_C|}c^{\star}_{B_{N-1}} + \frac{|Q_C|}{|Q_B|+|Q_C|}c^{\star}_{C_{N-1}} \\
c^{adv}_{A_{N+1}} &= \frac{|Q_D|}{|Q_D|+|Q_E|}c^{\star}_{D_1} + \frac{|Q_E|}{|Q_D|+|Q_E|}c^{\star}_{E_1} \\
c^{adv}_{A_{N+2}} &= \frac{|Q_D|}{|Q_D|+|Q_E|}c^{\star}_{D_2} + \frac{|Q_E|}{|Q_D|+|Q_E|}c^{\star}_{E_2}
\end{aligned}
\tag{1.37}
$$

$$
\begin{aligned}
c^{dif}_{A_{-1}} &= \frac{|S_B|}{|S_B|+|S_C|}c^{\star}_{B_N} + \frac{|S_C|}{|S_B|+|S_C|}c^{\star}_{C_N} \\
c^{dif}_{A_{N+1}} &= \frac{|S_D|}{|S_D|+|S_E|}c^{\star}_{D_1} + \frac{|S_E|}{|S_D|+|S_E|}c^{\star}_{E_1}
\end{aligned}
$$

where Q_B, Q_C, Q_D, Q_E are the flow rates in the neighboring airways of A, and $c^{\star}$ denotes a concentration value in neighboring duct interpolated onto a grid with the same grid-size as the considered airway A. Using these values, the bifurcation boundary conditions in equation (1.34) were introduced with the vector $\underline{b}$ as follows

$$
\text{if } u \geq 0\text{:} \quad \underline{b} = \begin{pmatrix} \frac{-u_A}{2h}\left(-4c^{adv}_{A_{-1}} + c^{adv}_{A_{-2}}\right) + \frac{D}{h^2}c^{dif}_{A_{-1}} \\ \frac{-u_A}{2h}c^{adv}_{A_{-2}} \\ 0 \\ \vdots \\ +\frac{D}{h^2}c^{dif}_{A_{N+1}} \end{pmatrix}
$$

$$
\text{if } u < 0\text{:} \quad \underline{b} = \begin{pmatrix} \frac{D}{h^2}c^{dif}_{A_{-1}} \\ \vdots \\ 0 \\ \frac{-u_A}{2h}(-c_{A_{N+1}}) \\ \frac{-u_A}{2h}\left(4c_{A_{N+1}} - c_{A_{N+2}}\right) + \frac{D}{h^2}c_{A_{N+1}} \end{pmatrix}
\tag{1.38}
$$

Inlet boundary conditions: At the inlet (mouth), boundary conditions of Dirichlet type, $c_0 = 1$, for inspiration, and Neumann type, $-4c_0 + 3c_1 - 1c_2 = 0$, for expiration were chosen. The left and right-hand operator, $\mathcal{L}$ and $\mathcal{R}$ respectively, in Equation (1.34) were modified accordingly.

Time-step and grid size

The time-step size Δ was linked to the grid-size h by means of the well known CFL condition.

$$
h = \frac{u\Delta t}{CFL} \tag{1.39}
$$

Considering a given time step Δt, the number of elements of a 1D mesh of length $l = Nh$ (in either duct-like airways or a trumpet module) was fixed as

$$
N = \texttt{floor}\left\{\frac{l\ CFL}{u\Delta t}\right\} \tag{1.40}
$$

where $\mathsf{CFL} = 1.0$ was used for all simulations.

If faster advection velocities (e.g. during peak inspiration or expiration) causes the number of elements to drop bellow a defined minimal number $\mathsf{N}_{\min}$, a time-step refinement (TSR) $\Delta t/n_{\text{tsr}}$ was activated, which was computed as

$$n_{\text{tsr}} = \frac{u \Delta t \mathsf{N}_{\min}}{l \, \mathsf{CFL}} \tag{1.41}$$

The refined time-step $\Delta t/n_{\text{tsr}}$ was then used for the numerical solution.

1.5 Code structure

The computer code is implemented in C++ and consists of four objects (classes) (`lung.cpp`, `duct.cpp`, `lobule.cpp`, `gas.cpp`) which constitute the lung model. The remaining source files relate to the procedural aspects of the simulation and data handling. An overview of the whole environment of the model is explained in more detail in Section 2.1 in the upcoming chapter. Here, the main procedural structure of the solver, wich are defined in `main.cpp` and `flDriver.cpp`, are outlined, and the architecture of the lung model is shown schematically (see Figure 1.7).

Input / Output

Reading and writing of input, respectively output data is defined in `main.cpp`, which also contains the simulation start command via the call of a driver. The structure is as follows:

- Reading system (model) parameters and simulation control parameters from input files.
- Dynamically allocate and fill (define) containers for these parameters.
- Dynamically allocate (no filling yet) containers for simulation outputs / results.
- Call of driver for simulation.
- Write output containers to local environment.

Model construction & simulation

The driver `flDriver.cpp` called in `main.cpp` first constructs (builds) the model according to a parametrized fractal morphology and initializes its member variables. After this the simulation time loop is started in which the different (per-time-step) numerical commands are called. In the following the procedure of a simulation run is outlined. Thereby $\mathbf{p}(t)$ and $\mathbf{Q}(t)$ refer to vectors containing pressure data at all bifurcation nodes, respectively flow data in all airway units, $\mathbf{c}(t)$ refers to a vector containing all concentration values in all airway units, and Γ decribes a list with all model parameters

- Construct lung object (see model architecture in Figure 1.7).
- Generate fractal tree of duct-like airways.
- Generate / Add trumpet-lobules at the end of each terminal duct.
- Pass input data (e.g. flow rate profile, breath period) to lung model.
- Read and apply lung modification instructions (e.g. for lobular compliance modification via parameter ϕ)

- Allocate / Initialize vectors and matrices for ventilation LPM (use of pressure initial conditions).
- Define gridsize.
- Setup gas object and initialize (use of concentration initial conditions).
- Start time loop: $t^n = t^0, t^1, t^2, ..., t^{end}$
  - Re-define breath period if necessary.
  - Compute TSR.
  - Start inner loop (according to TSR): $t^{n,k} = t^{n,0}, t^{n,1}, t^{n,2}, ..., t^{n,n_{tsr}}$
    - Write entries in system matrix and right-hand side vector for ventilation LPM.
    - Solve for pressure and distribute to nodes in LPM: $\mathbf{p}(t^{n,k+1}) = f(\mathbf{p}^{n,k}, \mathbf{Q}^{n,k}, \Gamma)$.
    - Compute flow rate in each duct-like airway: $\mathbf{Q}(t^{n,k+1}) = f(\mathbf{p}^{n,k+1}, \Gamma)$
    - Update trumpet-lobule dynamics.
    - Successively update concentration in 1D domains (domain decomposition approach): $\mathbf{c}(t^{n,k+1}) = f(\mathbf{c}^{n,k}, \mathbf{Q}^{n,k+1}, \Gamma)$
  - Write results of current time step to output data containers.

Model architecture

The numerical model of the lung consists of four different objects: The lung-object (`lung.cpp`) governs the construction of the model and all calls of different operations (e.g. numerical computations) within the different lung units, i.e. ducts and lobules. For computational or data transfer operations that are applied to all (or certain) lung units, a recursive implementation is chosen. E.g. the rules to grow (or stop growing) daughter ducts according to Equation (1.2) are the same for all ducts. Therefore it is handy to implement only the rule and then apply it via recursion to all ducts and lobules within the tree-like morphology. This concept is applied for many operational steps of the model. The strict implementation of recursive functions for this type of domain morphology also assures that the domains are scanned / traveled in a predefined manner, which eases the right allocations of different units or nodes. The lung-object points (via pointer *pTrachea*) on a fractal tree of duct-objets (`duct.cpp`). A duct-like airways is connected (via pointers *pParent*, *pMinoDaughter*, *pMajorDaughter*) to the neighoring branches (ducts). Terminal ducts (defined by d_{Lim}) are connected to a lobule-object (`lobule.cpp`, via pointer *pLobule*) Each duct and lobule is associated with a proper gas object via *pSpecies*. A schematics of this architecture is shown in Figure 1.7.

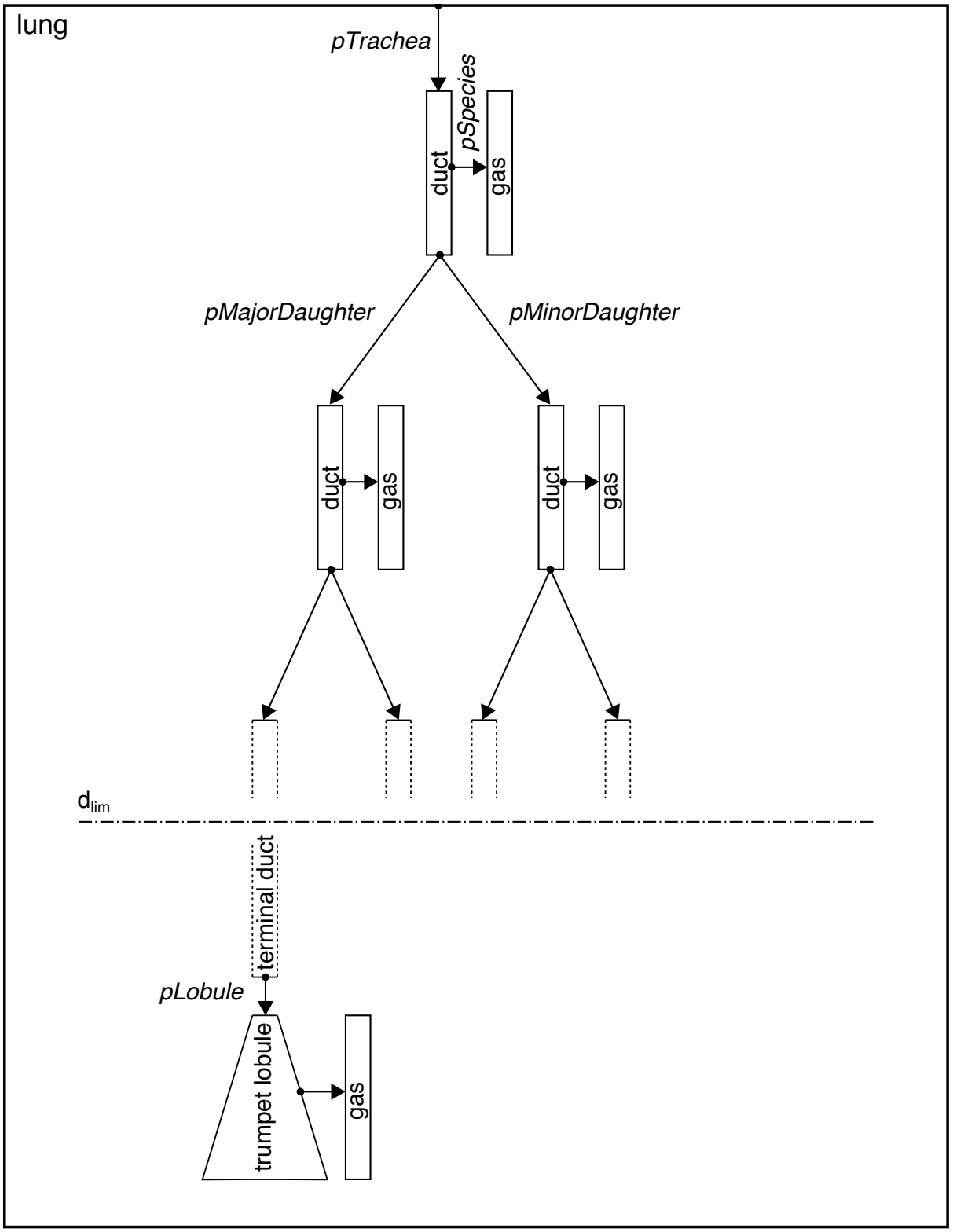

Fig. 1.7: *Architecture of the lung model. Each object (lung, duct, lobule, gas) is connected by pointers (e.g pMajorDaughter) with neighboring objects, i.e. lung units within the model. Operations are executed in a recursive manner starting at the duct representing the trachea. In addition each element in the airways tree (ducts and lobules) are asociated with a gas object (via pSpecies). Note: For sake of a better visibility the connection from a daugher duct to its parent duct, i.e. pParent, is not shown.*

Chapter 2

Application

2.1 Environment

The model is hosted, both for development and application, in an environment with a specific folder structure, which must not be changed. It contains the following directories

Dir. name	Content
constant	Constant input data/files with constant model and simulation parameters parameters used for initialization and construction. Files are usually defined/set only once for a specific campaign (several simulations)
data	Variable input data/files which such as the inlet flow profile and the parameter modification table. These are files which are usually altered for each single simulation. Output files (results). Other files wich results from pre-processing steps.
scripts	Several Python scripts used for pre and post processing
src	C++ source files of the model

The constant directory

The following files are stored in this directory:

File name	Content
`controlDict`	Process settings (washout type, gas species, boundary conditions type) Settings for the numerical solution method Output format settings
`dataWeibel`	Data for airways geometry from Weibel (1963) (currently not dynamically linked to / read by the model)
`diffCoeff`	look-up table with molecular diffusion coefficients for different species pairing (O2, CO2, He, SF6, N2)
`systemProperties`	All static (constant) model parameters
`transFact`	Lookup-table for transmissibility factors from pulsatile flow (Womersley) theory. This table is precomputed using the script `pulsatile_transmissibility.py` in the *scripts* directory (see bellow)

The data directory

The following files are stored in this directory (I: simulation /model input, O: simulation output):

File name	Content
`inletFlow` (I)	Flow rate data (one column only) in m^3/s. The data is computed using the sripts `generateFLinputSine.py` or `generateFLinputSplines.py` (generic), or copied from a actual measurement using the scripts `generateFLinputFromAFile.py` (SBW data) or `generateFLinputFromBFile.py` (MBW data).
`modifyLung` (I)	Table containing data for structural and mechanical lung model modifications The columns refer to ... first column: duct or lobule absolute index. second column: transmissibility modification parameter ξ for ducts. third column: lobular residual volume modification parameter θ fourth column: lobular compliance modification parameter ϕ fifth column: lobular flow resistance modification parameter τ If the file is empty, no modifications will be applied.
`nbfs` (I)	Contains two numbers: The number of breaths and the sampling frequency corresponding to `inletFlow`. The sampling frequency f_s also determines the time-step size $\Delta t = 1/f_s$ used in the simulation.
`pleuralPressure` (I)	Generic data (from `generateFLinputSplines.py`) for the pleural pressure (one column). This data is only used when pressure boundary conditions at the periphery are used in the ventilation LPM (via the flag in `controlDict`) Important notice: In the current version 1.0 of the model this option has been implemented but not tested yet.
`primary_results` (O)	Primary (high-level) simulation results that are always outputted. The columns refer to ... first column: time in s second column: normalized concentration of first tracer gas third column: normalized concentration of second tracer gas fourth column: computed (relative) pleural pressure (solution of ventilation LPM)
`TBTVN` (I)	Table containing breath period T_B [s] (first column), tidal volume V_T m^3, and the number of samples for each breath. The data corresponds with `inletFlow` and is automatically computed with the same script (e.g. `generateFLinputSine.py`).

The following sub-directories are hosted in this directory (I: simulation /model input, O: simulation output):

Dir. name	Content
duct (O)	Contains secondary output data (full / total lung model output), i.e. nodal data for several quantities within the discretized network of duct-like airways The following quantities are outputted ... Duct index number; velocity, pressure, gas concentrations, duct (airway) radius. The data is written / outputted only if the corresponding flag in the `controlDict` file is set. The output frequency, i.e. for which time-step the full output is stored is also defined in `controlDict`. The file format is (VTK). One file per time-step is stored.
lobule (O)	Contains secondary output data (full / total lung model output), i.e. nodal data for several quantities for each trumpet lobule The following quantities are outputted ... Lobule index number; velocity, gas concentrations, trumpet lobule radius, relative dilatation, pleural pressure The data is written / outputted only if the corresponding flag in the `controlDict` file is set. The output frequency, i.e. for which time-step the full output is stored is also defined in `controlDict`. The file format is (VTK). One file per time-step is stored.
lufu (I)	contains the A-Files (SBW) and B-Files (MBW) from which the flow rate data is copied if no generic flow rate profile is used.

The scripts directory

The following scripts are hosted in this directory (PRE: for pre-processing, PST: for post-processing):

Script name	Function
`dilatation_map.py` (PST)	Script to visualize the dilatation of the trumpet lobules as a function of time.
`generateFLinputFromAFile.py` (PRE)	Extracts the flow rate data from actual SBW measurements (i.e. an A-File), generates the input data `inletFlow`, `nbfs`, and `TBTVN`, and stores them in the *data* directory. Here, the sampling frequency f_s, i.e. the inverse of the time-step Δt is defined.
`generateFLinputFromBFile.py` (PRE)	Extracts the flow rate data from actual MBW measurements (i.e. an B-File), generates the input data `inletFlow`, `nbfs`, and `TBTVN`, and stores them in the *data* directory. Here, the sampling frequency f_s, i.e. the inverse of the time-step Δt is defined.
`generateFLinputSine.py` (PRE)	Computes generic flow rate data, generates the input data `inletFlow`, `nbfs`, and `TBTVN`, and stores them in the *data* directory. Here, the sampling frequency f_s, i.e. the inverse of the time-step Δt, the tidal volume and the number of breaths are defined.
`generateFLinputSpline.py` (PRE)	Computes generic flow rate and pleural pressure data, generates the input data `inletFlow`, `nbfs`, and `TBTVN`, and stores them in the *data* directory. Here, the sampling frequency f_s, i.e. the inverse of the time-step Δt, the tidal volume, the number of breaths, and the breath profile asymmetry (different durations of inspiration and expiration) are defined.
`genModifyLung.py` (PRE)	Definition of lung model modification parameters, i.e. which ducts or lobules are modified and how they are modified.
`pulsatile_transmissibility.py` (PRE)	Computes a table with factors $\in [0, 1]$ used to change the transmissibility to account for inertial effects in pulsatile flow. The table is stored as `transFact` in the *constant* directory.
`results.py` (PST)	Visualization of the primary results, i.e. normalized gas concentration (washout profile) and pleural pressure evolution.

The src directory

The following source files are hosted in this directory:

File name	Description
`main.cpp`	Dynamic input / output handling; storage allocation; driver call.
`flDriver.cpp`	Procedural calls for model construction, model modification, simulation.
`lung.cpp`	Lung-object with all lung-level member variables and instances.
`duct.cpp`	Duct-object with all duct-level member variables and instances.
`lobule.cpp`	Lobule-object with all lobule-level member variables and instances.
`gas.cpp`	Gas-object (species-object) with all gas-level member variables and instances.
`IOdict.cpp`	Object for data input / output handling.
`visit_writer.cpp`	Function for writing vtk-data format (from VisIt LLNL source code)

2.2 Pre-processing

Besides the various parameter input files in *constant*, the main inputs for the model are the flow profile (`inletFlow`) at the mouth and a table (`modifyLung`) containing the values of the modification parameters applied to a specific airway unit (duct-like airway or trumpet lobule). Alongside with the flow profile the duration of the simulation (number of simulated breaths), and the tidal volume V_T and the breath period of each breath are defined. Moreover, the sampling frequency f_s of the flow signal directly defines the time-step size ($\Delta t = 1/f_s$) used in the model simulation.

Flow profile and related quantities

The flow profile can either be read from an actual A-File or B-File, and prepared (cropped and resampled wit f_s), or it can be generated based on a sine-function or using splines. For each different approach, a separate Python-script in the sub-folder *sripts* is prepared and when run handles all the needed computations. For instance, the script `generateFLinputSine.py` defines a generic sine-shaped flow rate profile. At the beginning of the script, the needed parameters have to be defined (i.e. number of breaths, time-step size / sampling frequency, tidal volume, breath period). The profile (i.e the file `inletFlow`) is then stored in the *data* sub-folder. In case the flow rate profile is taken from an A/B-File, this File has to be stored previously in the corresponding *data/lufu* sub-folder. Besides the flow profile, separat files, one containing a table where the breath period the tidal volume and the number of data points are listed, another containing only two values for the total number of breaths and for the sampling frequency, are stored in the *data* sub-folder, when one of the `generateFLinput*.py` scripts is executed. Note: The spline-based generic flow profile is used for asymmetric (but regular) flow profiles, meaning that duration of inspiration and expiration are different. For this option, an additional signal for the pleural pressure is generate, which can be used in conjunction with different types of model boundary conditions. Although the pre-processing (i.e. the corresponding script `generateFLinputSplines.py`) works, the use of asymmetric flow profiles or pleural pressure boundary conditions has not yet been tested with the model.

Tab. 2.1: *Modification parameter, their meaning and typical numerical ranges*

Param.	Description	Typ. range
ξ	Reduces the transmissibility (inverse of flow resistance $T = 1/R$) to a certain fraction. Transmissibility is proportional to the diameter to the power of four ($T \sim d^4$). This means that a parameter value of 0.5 corresponds to airway obstruction of approximately 16%. Note: to small parameter values applied to numerous airways might cause numerical issues during simulation run	$[0.5, 1]$
θ	Scales the residual volume of a trumpet lobuel. A parameter value of $\phi = 0.5$ and $\phi = 1.5$ causes the lobule volume to be reduced, respectively increase to 50% and 150% of the nominal residual volume resulting from the FRC and the total number of lobules. (For more details see Section 1.3)	$[0.5, 1.5]$
ϕ	Reduces or increases the lobular compliance. (For more details see Section 1.3)	$[0.5, 1.5]$
τ	Increases the total flow resistance within a trumpet lobule. (For more details see Section 1.3)	$[1, 10]$

Modification parameters

The modification parameter table is computed with the script `genModifyLung.py`. The different modification parameters and their meanings are listed in table 2.1. The total number of airway units which are modified, has to be defined in the script. This number should not be higher than the total number of ducts or lobules generated in the model, which on the other hand depends on the values set for d_{lim} in the `systemProperties` file. For a specific set of modifications, i.e. which unit is affected (via airway ID), which value is set for the modification parameters (for duct transmissibility ξ, for lobule residual volume θ, for lobule compliance ϕ, or for lobular flow resistance τ), has to be implemented by the user. There exists the possibility to define a subset of units which are modified in the same manner, or to define the parameter values using statistical distributions functions. After the script has been executed, the table (`modifyLung`) is stored in the *data* sub-folder.

The `genModifyLung.py` script is used to automatically create a modification parameter table according to specific rules and/or distribution function. However, it is also possible to generate or adapted a table by hand, for instance in case only airway units ought to be modified. It is in this case important to preserve the structure of the table, i.e. the order of the parameters, and although an unmodified airway unit does not have to be listed, as soon as one of it's parameters is modified (e.g. not equal 1) the other parameters have to be listed, whether they are modified or not.

Pulsatile transmissibility

The script (pulsatile_transmissibility.py) computes the look-up table for the pulsatile transmissibility factors based on Womersley's (Womersley, 1957) theory, which is stored in the *constant* sub-folder. The data in this table remains the same for practically all simulations. It is therefore usually not needed to run the script nor to make any changes to it.

2.2.1 Code compilation

The compilation of the C++ source files has to be done on each OS and Computer separately. The application which is built during compilation is called `flPROG`.

Mac & Linux

In a UNIX environments, the make file provided in the *src* sub-folder can be used for comilation To this end, simply open a terminal (in the directory containing the source :), and use the command *make*. Then use *make moveup* to move the compiled code to its natural habitat (one directory up). Optionally, you can clean up the directory with *make clean*. Note: To get the needed compiler on Mac, donload Xcode. On Linux, get GCC 4.8.5 or higher.

Windows

On Windows it is recommended to use the Code::Blocks IDE (www.codebloks.org) and integrate the C++ source files in a dedicated project. Using Code::Blocks, compiler options can be set via the toolbar of the IDE.

2.2.2 System and control parameter settings

Most of the time, the majority of the system and control parameters hosted in `systemProperties` and `controlDict` (*constant* sub-folder) remain unchanged within a run of multiple simulations, while typically the modification parameters (`modifyLung`) and/or the flow profile (`lung`) are altered. A possible exception could be the prescribed FRC (in `systemProperties`), which might be changed more frequently. In general, however, system and control parameters should only be changed, if the user is awar of their meaning and implication to the simulation. In the corresponding files, a brief description is provided together with the parameter name and the set value.

2.3 Simulation run

In order to access the files and tables in the various sub-folders, the model (application `flPROG`) has to be stored in the same directory as the *constant*, and the *data* sub-folders. It is however recommended to use the same folder structure as described in Section 2.1. This way, prepocessing, simulation run, and post-processing can be performed without copy-pasting files from one place to another.

Single simulation run

The application is executed in Mac/Linux using the `./` command in the terminal, and in Windows simply by double clicking on the Executable (*.exe* file), i.e. the application. Or of course appropriate commands in a Shell or Python script. When the simulation starts running, a prompt displays the system and the control variable first (under Windows a console will open). Later, at specific time-steps, the absolute and relative errors of the linear system numerical solution (see Section 1.4.1) is displayed. The values of the errors should be $\ll 1$ (i.e. $< 1e-07$). The corresponding time is also displayed, showing the progress of the simulation. Successful simulation end will be indicated with the word *done* and the total execution time. The model output, will be stored as a plain text file called `primary_results` (described above in Section 2.1) in the *data* sub-folder. Optionally, more detailed simulaton results will be stored in the *data/lobule* or *data/duct* sub-folder.

Batch simulation run

For running multiple simulations automatially, e.g. each with different modifications (as defined in each `modifyLung` file), a short Python-script called `batch_run.py` is prepared and stored in the *script* sub-folder. This is the only script, which must not be executed from *script*. Instead, for a batch simulation, a new folder structure has to be created which contains a separate folder (e.g. called *sim_1*) for each simulation run and the `batch_run.py` is then placed in the. Each of this simulation-folders corresponds with the model folder (i.e. the environment) but with different content e.g. in the *data* or *constant* sub-folders. Running the `batch_run.py` script will then pass throught the simulation folders and successifely executed each of the model applications stored therin. This will run the simulations in series. For running parallel simulations, e.g. by running several `batch_run.py` scripts at the same time, it should be checked how manx CPUs are available, because for each newly executed application, a new CPU will be allocated.

2.4 Post-processing

Possible ways of post-processing the raw simulation data are quit open and depend on the type of analysis which is planed. Two types of output data are at hand: The time series for the simulated washout profile and the pleural pressure (`primary_results` in *data*), and the unstructured mesh data for different simulation variables, e.g. $\mathtt{c}(\mathtt{x},\mathtt{t}$, $\mathtt{u}(\mathtt{x},\mathtt{t})$, $\mathtt{p}(\mathtt{x},\mathtt{t})$, throughout the model (full output, `VTK` files in *data/duct* and data/lobule sub-folders). Certain basic means for visualization and further processing are provided in the *scripts* sub-folder. For instance it is possible to plot the simulated washout profile and pleural pressure, and store these time-series in a B-File (*make_BFile.py*), which can be further used for regular type MBW data analysis such as calculation of FRC, LCI, S_{cond}, S_{acin}. Regarding the full output, a very powerful tool for visualization and inspection of the mesh-data is the free software `VisIt` (Lawrence Livermoore National Laboratory). Example for Visit scence and plots here.

xxx

List of Figures

List of Tables

Bibliography

Cussler, E. L. (2009). *Diffusion*. Cambridge University Press, 1st edition.

Kaczka, D. W., Colletti, A. A., Tawhai, M. H., and Simon, B. A. (2011). Computational Analyses of Airway Flow and Lung Tissue Dynamics. In *Image-Based Computational Modeling of the Human Circulatory and Respiratory System*, chapter 10. Springer US.

Majumdar, A., Alencar, A. M., Buldyrev, S. V., Hantos, Z., Lutchen, K. R., Stanley, E. H., and Suki, B. (2005). Relating Airway Diameter Distributions to Regular Branching Asymmetry in the Lung. *Physical Review Letter*, 95(168101).

Robinson, P., Latzin, P., Verbanck, S., Hall, G., Horsley, A., Gappa, M., Thamrin, C., Arets, H., Aurora, P., Fuchs, S., King, G., Lum, S., Macleod, K., Paiva, M., Pillow, J., Ranganathan, S., Ratjen, F., Singer, F., Sonnappa, S., Stocks, J., Subbarao, P., Thompson, B., and Gustafsson, P. (2013). Consensus statement for inert gas washout measurement using multiple- and single-breath tests. *European Respiratory Journal*, 41(3):507–22.

Singer, F., Stern, G., Thamrin, C., Abbas, C., Casaulta, C., Frey, U., and Latzin, P. (2013). A new double-tracer gas single-breath washout to assess early cystic fibrosis lung disease. *European Respiratory Journal*, 41:339–345.

Weibel, E. R. (1963). *Morphometry of the Human Lung*. Springer-Verlag Berlin Heidelberg, xy edition.

Womersley, J. R. (1957). Oscillatory flow in arteries: the constrained elastic tube as a model of arterial flow and pulse transmission. *Physics in Medicine and Biology*, 2(2).