Published in final edited form as:

J Comput Phys. 2013 July ; 244: . doi:10.1016/j.jcp.2012.10.021.

A Bidirectional Coupling Procedure Applied to Multiscale Respiratory Modeling[☆]

A.P. Kuprat^{*}, S. Kabilan, J.P. Carson, R.A. Corley, and D.R. Einstein^{**}

Fundamental and Computational Sciences Directorate, Pacific Northwest National Laboratory, Richland, WA

A.P. Kuprat: andrew.kuprat@pnnl.gov; S. Kabilan: senthil.kabilan@pnnl.gov; J.P. Carson: james.carson@pnnl.gov; R.A. Corley: rick.corley@pnnl.gov; D.R. Einstein: daniel.einstein@pnnl.gov

Abstract

In this study, we present a novel multiscale computational framework for efficiently linking multiple lower-dimensional models describing the distal lung mechanics to imaging-based 3D computational fluid dynamics (CFD) models of the upper pulmonary airways in order to incorporate physiologically appropriate outlet boundary conditions. The framework is an extension of the Modified Newton's Method with nonlinear Krylov accelerator developed by Carlson and Miller [1, 2, 3]. Our extensions include the retention of subspace information over multiple timesteps, and a special correction at the end of a timestep that allows for corrections to be accepted with verified low residual with as little as a single residual evaluation per timestep on average. In the case of a single residual evaluation per timestep, the method has zero additional computational cost compared to uncoupled or unidirectionally coupled simulations. We expect these enhancements to be generally applicable to other multiscale coupling applications where timestepping occurs. In addition we have developed a "pressure-drop" residual which allows for stable coupling of flows between a 3D incompressible CFD application and another (lower-dimensional) fluid system. We expect this residual to also be useful for coupling non-respiratory incompressible fluid applications, such as multiscale simulations involving blood flow.

The lower-dimensional models that are considered in this study are sets of simple ordinary differential equations (ODEs) representing the compliant mechanics of symmetric human pulmonary airway trees. To validate the method, we compare the predictions of hybrid CFD-ODE models against an ODE-only model of pulmonary airflow in an idealized geometry. Subsequently, we couple multiple sets of ODEs describing the distal lung to an imaging-based human lung geometry. Boundary conditions in these models consist of atmospheric pressure at the mouth and intrapleural pressure applied to the multiple sets of ODEs. In both the simplified geometry and in the imaging-based geometry, the performance of the method was comparable to that of monolithic schemes, in most cases requiring only a single CFD evaluation per time step. Thus, this new accelerator allows us to begin combining pulmonary CFD models with lower-dimensional models of pulmonary mechanics with little computational overhead. Moreover, because the CFD and lower-dimensional models are totally separate, this framework affords great flexibility in terms of the type and breadth of the adopted lower-dimensional model, allowing the biomedical researcher

[☆]This document is a collaborative effort.

© 2012 Published by Elsevier Inc.

^{*}Corresponding author. ^{**}Principal corresponding author.

Publisher's Disclaimer: This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final citable form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

to appropriately focus on model design. *Research funded by the National Heart and Blood Institute Award 1R01HL073598.*

Keywords

computational fluid dynamics; multiscale coupling; pulmonary airflows; Krylov subspace; modified Newton-Raphson

1. Introduction

Gas exchange, the primary function of the lungs, requires a functional interface between the air and vascular system. This interface is disrupted by various diseases of genetic or environmental origin, which dramatically alter the quality of life [4, 5, 6]. Computational modeling has played a critical role in understanding the functional consequences of disease as well as the potential health implications of exposure to environmental or pharmaceutical agents that lead to chronic lung diseases.

Historically, respiratory system models have utilized empirical, mass-transfer, or compartmental approaches (e.g. [7, 8, 9, 10, 11]). These lower or zero spatial-dimension models of lung physiology are helpful for understanding important phenomena, including gas-exchange, metabolism, and especially lumped tissue mechanics. However, their lack of detailed airway structures cannot account for important anatomical differences, and their inability to incorporate ventilation/perfusion heterogeneity in health and disease have limited their use for understanding the spatial character of disease and exposure.

Advancements in medical imaging, computer hardware, and computational techniques have enabled the development of spatially complex three-dimensional (3D) computational fluid dynamics (CFD) models of the respiratory system that capture detailed airway structures, and thus can address ventilation/perfusion heterogeneity (e.g. [12, 13, 14]). However, because the resolved geometry of these 3D models is limited to the resolution of the imaging data, and because they cannot account for airway or parenchymal compliance in health and in disease, their utility in understanding and predicting disease is also limited.

To ultimately understand the impact of disease states on respiratory function and dosimetry, it is desirable to construct multiscale models of respiration such that the mechanical, metabolic, and physiologic sophistication of lower-dimensional models can inform geometrically complex 3D models based on imaging data. This multiscale approach would effectively extend the resolution-limited 3D model to the entire respiratory tract. This is particularly relevant because tissue alterations due to disease are often found at the alveolar level, far below the resolution of the 3D models.

In an early effort to couple lower-dimensional models of distal lung mechanics to imaging-based 3D-CFD models of the upper airways, Ma et. al [15] employed a unidirectional coupling methodology to investigate the flow characteristics and deposition pattern in the human upper and central airways. The geometry consisted of a 3D model of the upper airway based on Magnetic Resonance (MR) images and an anatomically based 3D model of the central airways reconstructed from Computed Tomography (CT) images. A one-dimensional (1D) transmission line model was adopted to represent the distal airway tree that accounted for the impedance of the small airways. The electrical representation of the distal airway tree was coupled unidirectionally to the outlets of the 3D model, in order to supply the boundary conditions necessary for the CFD simulation. Limitations of this study were the unidirectional coupling and the assumption of homogenous ventilation based on the ratio of the number of alveoli in the attached small airway tree to the total number of alveoli

in the whole airway tree. The authors themselves suggested that for heterogeneous ventilation, a more iterative or bidirectional coupling method is required [15].

Lin et al. [16] investigated pulmonary airflow in subject-specific models of the human lung. The 3D surface of the lung was based on multidetector row computed tomography (MDCT) and the distal airways beyond the resolution of the CT were represented as volume-filling 1D tree models. Using differences between images at two points of the respiratory cycle, volume changes, and hence flows, were derived at the distal ends of the 1D models and propagated using the 1D connectivity back up to the distal ends of the 3D model, which became conventional prescribed flow boundary conditions for a 3D CFD solver. This work was pioneering in the sense that it underlined the importance of accounting for distal mechanics in airflow simulations. Due to different modeling assumptions than made in the present work, 1D pressures could simply be assumed to be equal to those computed by the 3D CFD solver at the 1D/3D interface and any necessary iteration for solution consistency occurred elsewhere.

Recently Wiechert et al. [17] adopted the Dirichlet-Neumann approach, originally introduced for blood flow [18, 19]. This multiscale model of respiration also combines an imaging based geometry with lower-dimensional models of distal lung mechanics. However, unlike the previous approaches, the interface condition was addressed bi-directionally in a numerically robust fashion by adding the degrees of freedom of the lower-dimensional model monolithically to the solution matrix. From a numerical point of view, there is little doubt that this approach is nearly optimal for those lower-dimensional systems that can be consistently discretized. The added degrees of freedom are relatively few when compared to the overall system. Thus, the added cost of coupling the lower-dimensional models is typically negligible, such that cost per timestep for these coupled simulations can be expected to be about the same as the cost of an uncoupled 3D CFD simulation.

The main limitations of this latter “monolithic” approach are two-fold. Firstly, it is inflexible since it requires the modeler, who may lack numerical sophistication, to consistently discretize the lower-dimensional model and to add the terms of that discretized model to the solution matrix. This is not an idle concern. The majority of biomedical research is conducted by those in the medical and biomedical professions. To be most effective, their focus should be on the biological system not on the underlying numerics. They should be free to introduce changes at will to their lower-dimensional models or indeed to adopt such models from others with no implementation penalty. Secondly, and perhaps more importantly, not all lower-dimensional models can be easily added to the solution matrix, and some not at all. Respiration is a complicated phenomenon in which airflow, the composition of inspired gases, the mechanical composition of tissues including smooth muscle, tissue and blood metabolism, the nervous system and cardiovascular health and mechanics are deeply intertwined. What is needed is a numerical framework that is not only efficient and robust, but that also enables the modeler the freedom to explore and understand the connection between disease, diagnosis and treatment.

In this study we present a novel partitioned multiscale computational framework for iterative coupling of multiple boundary interfaces in parallel which addresses this challenge. The method has an efficiency that is nearly equal to that of the monolithic methods, yet also offers greater flexibility in terms of enabling the coupling of existing finite element or finite volume solvers, without consistent discretization between higher-dimensional and lower-dimensional systems. Specifically, we extend the Modified Newton’s Method with nonlinear Krylov accelerator developed by Carlson and Miller [1, 2, 3]. Our extensions involve retention of subspace information over multiple timesteps, and a special correction at the end of a timestep that allows for corrections to be accepted with verified low residual with

as little as a single residual evaluation per timestep on average. In the case of a single residual evaluation per timestep, the method has zero additional computational cost compared to uncoupled or unidirectionally coupled simulations. In addition we have developed a “pressure-drop” residual which allows for stable coupling of flows between a 3D incompressible CFD application and another (lower-dimensional) fluid system.

Adopting a partitioned approach simplifies the coupling procedure from a modeling point of view. Rather than needing to specify individual components of the solution matrix, the user needs to only design and provide a residual to the accelerator which can be thought of as a ‘black box’. In addition, adoption of a partitioned approach enables the user to choose any three-dimensional solver - finite element or finite volume - for which the outer iterations are accessible, a requirement that is met by most commercial codes and all open-source codes.

2. Theory

A common characteristic for the multiscale linkage of disparate systems is that the interface conditions on the multiscale boundaries are both unknown and multiple. It is not sufficient therefore to simply impose the output of one system on the other. With the interface conditions as the unknowns, the two systems by themselves are under-determined and the condition of their common equilibrium must be arrived at by iteration.

One possible approach would be to create a monolithic system in which both systems are solved for simultaneously. This approach is generally very efficient if the two systems can be consistently discretized. However, it requires reformulation each time the system changes and it presupposes a dedicated solver. An alternative is a partitioned system, in which both systems are solved separately but iteratively, with certain common interface conditions. It is this latter approach that we formulate below.

2.1. Statement of the Problem

Formally, we seek the solution to

$$\mathbf{r}(\mathbf{x}) = \mathbf{0} \quad (1)$$

on the interface, where $\mathbf{r}(\mathbf{x})$ is the residual vector, which expresses the difference between the outputs of two time-dependent systems at a time t . Note we express $\mathbf{x}$ as a vector of length M since we have M (potentially thousands) of interface conditions between the two systems to satisfy simultaneously. Ideally as time-stepping occurs over the duration of the simulation, Equation (1) will be satisfied at all time-steps; in practice it should always be satisfied to within a numerical tolerance. The standard, though somewhat impractical, method for solving Equation (1) would be to apply *Newton's Method*

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{r}'(\mathbf{x}_n)^{-1} \mathbf{r}(\mathbf{x}_n) = \mathbf{x}_n - \mathbf{d}\mathbf{x}_{n+1} \quad (2)$$

where $\mathbf{r}'$ is the Jacobian matrix $\frac{\partial \mathbf{r}}{\partial \mathbf{x}}$. In typical practice, this approach is not strictly adhered to, since each Newton iteration requires M evaluations of the residual $\mathbf{r}(\mathbf{x})$ to evaluate $\mathbf{r}'$ by finite differences. In a multiscale setting where one of the systems is 3D with millions of unknowns, residual evaluation is expensive; moreover, potentially several Newton iterations are required per time step.

A related and considerably less costly method is the standard Modified Newton's Method which instead computes

$$\mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{K} \mathbf{r}(\mathbf{x}_n), \quad (3)$$

where $\mathbf{K}$ is a fixed approximate inverse Jacobian matrix. It may be for example $(\mathbf{r}'(\mathbf{x}_0))^{-1}$ where $\mathbf{x}_0$ is the initial guess, or the inverse of a “stale” $\mathbf{r}'$ evaluated at a previous time step, or an approximation to the inverse of such a stale Jacobian. Note the *preconditioned residual* $\mathbf{s}(\mathbf{x}) \equiv \mathbf{K} \mathbf{r}(\mathbf{x})$ has Jacobian $\mathbf{s}'(\mathbf{x}) = \mathbf{K} \mathbf{r}'(\mathbf{x})$ which is approximately $\mathbf{I}$, the identity matrix, near the desired root of (2).

2.2. Overview of the Proposed Method

A further improvement over the standard Modified Newton Method is to combine it with the nonlinear Krylov accelerator developed by Carlson and Miller [1, 2]:

$$\begin{aligned} &\text{Do} && n=0, 1, 2, \dots \\ &&& \mathbf{r}_n = \mathbf{r}(\mathbf{x}_n) \\ &&& \mathbf{s}_n = \mathbf{K} \mathbf{r}_n \\ &\text{if } (\|\mathbf{s}_n\| < \text{TOL}) && \text{then exit loop} \quad (4) \\ &&& \mathbf{dx}_{n+1} = \text{NACCEL}(\mathbf{s}_n) \\ &&& \mathbf{x}_{n+1} = \mathbf{x}_n - \mathbf{dx}_{n+1} \end{aligned}$$

The two fundamental components of the system above are 1) a *preconditioner* $\mathbf{K}$, an approximation of the inverse $(\mathbf{r}'(\mathbf{x}_n))^{-1}$ of the Jacobian matrix and 2) the NACCEL (nonlinear Krylov accelerator) procedure which modifies the iterates to enforce and accelerate convergence.

In Section 2.3 we first describe the NACCEL procedure, which is a general purpose accelerator for iteration and whose operation is largely independent of modeling considerations. In Section 2.4 we describe an enhancement of NACCEL we implemented for allowing retention of residual information across time steps. In Section 2.5 we describe a second enhancement of NACCEL which potentially decreases the number of residual evaluations required when timestepping. We also show the connection of the NACCEL routine with the Generalized Minimal Residual (GMRES) method [20] in the linear case. This is relevant, since GMRES is arguably the most popular Krylov subspace iterative method for nonsymmetric positive definite systems. In Section 2.6 we describe our particular residual function for matching incompressible CFD to distal pulmonary lung ODEs, and explain why it is somewhat more complex than simply computing the mismatch in the flows between the two regions.

2.3. Nonlinear Krylov Accelerator

The nonlinear accelerator procedure of Carlson and Miller was given in [1, 2]. This method, devised by Miller, was originally implemented by Carlson in the FORTRAN subroutine `nacel` (part of `bdf2.f`) available at [21]. It is also available as NLKAIN at <http://sourceforge.net/projects/nlkain>. We refer to the accelerator simply as ‘NACCEL’.

The NACCEL accelerator follows from several simple ideas:

- The residual $\mathbf{r}(\mathbf{x})$ is preconditioned by an approximate inverse Jacobian $\mathbf{K}$ so that the *preconditioned residual* $\mathbf{s}(\mathbf{x}) = \mathbf{K} \mathbf{r}(\mathbf{x})$ has Jacobian

$$\mathbf{s}'(\mathbf{x}) = \mathbf{K} \mathbf{r}'(\mathbf{x}) \approx \mathbf{I}, \quad (5)$$

the identity matrix *if* the preconditioner is freshly updated.

- To save computational time, we assume the preconditioner is not updated every time step, but is allowed to become “stale”. For the sake of designing a nonlinear iteration we *pretend* that the following *local linearity assumption* holds for all $\mathbf{x}$ near the desired root:

$$\mathbf{s}'(\mathbf{x}) = \mathbf{A}, \quad (6)$$

where $\mathbf{A}$ is an *unknown constant nonsingular* matrix.

- Because of (5), we make the *default assumption*

$$\mathbf{A} \, d\mathbf{x} \approx \mathbf{I} \, d\mathbf{x} \quad (7)$$

for any tiny increment $d\mathbf{x}$ for which we have no better information.

- For any approximate solution $\mathbf{x}$ with $\mathbf{s}(\mathbf{x}) \neq \mathbf{0}$, we seek to correct it to $\mathbf{x} - d\mathbf{x}$ where $\mathbf{0} = \mathbf{s}(\mathbf{x} - d\mathbf{x}) = \mathbf{s}(\mathbf{x}) - \mathbf{A} \, d\mathbf{x}$, implying the *linear correction equation*

$$\mathbf{A} \, d\mathbf{x} = \mathbf{s}(\mathbf{x}). \quad (8)$$

- As we proceed through successive iterates

$$\mathbf{x}_0, \mathbf{x}_1 \equiv \mathbf{x}_0 - d\mathbf{x}_1, \mathbf{x}_2 \equiv \mathbf{x}_1 - d\mathbf{x}_2, \dots,$$

we can evaluate

$$d\mathbf{s}_1 \equiv \mathbf{A} \, d\mathbf{x}_1 = \mathbf{s}(\mathbf{x}_0) - \mathbf{s}(\mathbf{x}_1), d\mathbf{s}_2 \equiv \mathbf{A} \, d\mathbf{x}_2 = \mathbf{s}(\mathbf{x}_1) - \mathbf{s}(\mathbf{x}_2), \dots$$

by differencing. We define the vectors

$$\delta\mathbf{s}_i \equiv d\mathbf{s}_i / \|d\mathbf{s}_i\| \text{ and } \delta\mathbf{x}_i \equiv d\mathbf{x}_i / \|d\mathbf{s}_i\|, \quad (9)$$

so that $\|\delta\mathbf{s}_i\| = 1$ and $\delta\mathbf{s}_i = \mathbf{A} \, \delta\mathbf{x}_i$. After $N + 1$ evaluations of $\mathbf{s}$, we can build up a matrix

$$\mathbf{V} = [\delta\mathbf{x}_1 \mid \delta\mathbf{x}_2 \mid \delta\mathbf{x}_3 \mid \dots \mid \delta\mathbf{x}_N]$$

for which the effect of multiplication of $\mathbf{A}$ on $\mathbf{V}$ is known and saved. Indeed we define the matrix

$$\mathbf{W} = \mathbf{A} \, \mathbf{V} = [\mathbf{A} \, \delta\mathbf{x}_1 \mid \mathbf{A} \, \delta\mathbf{x}_2 \mid \dots \mid \mathbf{A} \, \delta\mathbf{x}_N] = [\delta\mathbf{s}_1 \mid \delta\mathbf{s}_2 \mid \dots \mid \delta\mathbf{s}_N].$$

We consider the rectangular $\mathbf{V}$ to be a matrix of “causes” and the rectangular $\mathbf{W}$ to be a corresponding matrix of “effects”.

- Now given $\mathbf{V}$ and $\mathbf{W} = \mathbf{A} \, \mathbf{V}$, and an iterate $\mathbf{x}_k$ with its $\mathbf{s}_k = \mathbf{s}(\mathbf{x}_k)$ evaluated, we can approximately solve the linear correction equation (8) in a two pass procedure. First we let $\mathbf{v}$ be that element in $\text{range}(\mathbf{V}) = \text{span}\{\delta\mathbf{x}_1, \delta\mathbf{x}_2, \dots\}$ such that $\|\mathbf{s}_k - \mathbf{A} \, \mathbf{v}\|$ is minimized (a simple “linear least squares computation”). The resulting vector $\mathbf{w}^\perp \equiv \mathbf{s}_k - \mathbf{A} \, \mathbf{v}$ is orthogonal to $\text{range}(\mathbf{W}) = \text{span}\{\delta\mathbf{s}_1, \delta\mathbf{s}_2, \dots\}$. The solution to the

linear least squares problem is easily seen to be $\mathbf{v} = \mathbf{V}\mathbf{c}$ where $\mathbf{c} \in \mathbb{R}^N$ is the solution of the normal equations

$$\mathbf{W}^T \mathbf{W}\mathbf{c} = \mathbf{W}^T \mathbf{s}_k. \quad (10)$$

Second, at the corrected point $\mathbf{x}^* \equiv \mathbf{x}_k - \mathbf{v}$, we would have at zero added expense, under the local linearity assumption (6), the residual $\mathbf{s}(\mathbf{x}^*) = \mathbf{w}^\perp$ and we would want a further correction $\mathbf{z}$ satisfying

$$\mathbf{s}(\mathbf{x}^* - \mathbf{z}) = \mathbf{w}^\perp - \mathbf{A} \mathbf{z} = 0. \quad (11)$$

Now, invoking the default assumption (7) we choose $\mathbf{z} = \mathbf{w}^\perp$, thereby yielding the two pass correction

$$\mathbf{dx} = \mathbf{v} + \mathbf{w}^\perp. \quad (12)$$

- Under the *additional hypothesis* that $\mathbf{A}$ is positive definite, it is easy to see that $\text{range}(\mathbf{V}) = \text{span}\{\mathbf{s}_0, \mathbf{A} \mathbf{s}_0, \mathbf{A}^2 \mathbf{s}_0, \dots, \mathbf{A}^{N-1} \mathbf{s}_0\}$ which is $K_N(\mathbf{A}, \mathbf{s}_0)$ the N -dimensional Krylov subspace generated by $\mathbf{A}$ and $\mathbf{s}_0$. Thus NACCEL is indeed a Krylov subspace method. Further, we have that if $\mathbf{w}^\perp$ is not zero, then $\mathbf{w}^\perp$ is not in $\text{range}(\mathbf{V})$. Otherwise, $\mathbf{A} \mathbf{w}^\perp$ would be in $\text{range}(\mathbf{W}) = \mathbf{A} \cdot \text{range}(\mathbf{V})$. But $\mathbf{w}^\perp$ is orthogonal to $\text{range}(\mathbf{W}) = \mathbf{A} \cdot \text{range}(\mathbf{V})$ implying $(\mathbf{w}^\perp, \mathbf{A} \mathbf{w}^\perp) = 0$, violating the positive definiteness of $\mathbf{A}$. Thus our two pass correction $\mathbf{dx} = \mathbf{v} + \mathbf{w}^\perp$ is not in $\text{range}(\mathbf{V})$ and adjoining $\mathbf{dx}$ to $\mathbf{V}$ leads to a new matrix $\mathbf{V}$ with a larger column space. Hence the iteration does not stall.

In Algorithms 1 and 2, we present pseudo code for the original NACCEL procedure, which to our knowledge has never been formally presented in full, and which serves as necessary context for the modifications introduced in this manuscript.

Rather than inverting $\mathbf{W}^T \mathbf{W}$ in Equation (10), what is done is the standard numerical analysis technique for solving the least squares projection problem: the upper triangle of the symmetric inner production matrix $\mathbf{W}^T \mathbf{W}$ is copied into a square matrix $\mathbf{H}$ and then Cholesky factorization (Algorithm 2) is used to obtain the lower triangular factor $\mathbf{L}$ such that $\mathbf{L}\mathbf{L}^T = \mathbf{W}^T \mathbf{W}$. The factorization process writes $\mathbf{L}$ into the lower triangle of $\mathbf{H}$. Since $\mathbf{L}$ is lower triangular, the equation $\mathbf{L}\mathbf{L}^T \mathbf{c} = \mathbf{W}^T \mathbf{c}$ is then easily solved by forward substitution for $\mathbf{y} = \mathbf{L}^T \mathbf{c}$ and then $\mathbf{L}^T \mathbf{c} = \mathbf{y}$ is solved for $\mathbf{c}$ by back substitution.

The above all deals with the linear considerations of the method. A crucial feature of NACCEL is that it has logic to take into account the changing nonlinear nature of the residual $\mathbf{r}(\mathbf{x})$. In Algorithms 1 and 2, the subscripts are shuffled so that vectors $\delta_{\mathbf{s}_1}, \delta_{\mathbf{s}_2}, \dots$ are ordered from most recent to oldest (as well as corresponding vectors $\delta_{\mathbf{x}_1}, \delta_{\mathbf{x}_2}, \dots$). At each step of the Cholesky solve in Algorithm 2, the diagonal element is tested. The k^{th} diagonal element of $\mathbf{L}$, and hence the k^{th} diagonal element of $\mathbf{H}$ in Algorithm 1, gives the sine of the angle between the vector $\delta_{\mathbf{s}_k}$ and the lower-indexed (and more recent) vectors $\{\delta_{\mathbf{s}_1}, \dots, \delta_{\mathbf{s}_{k-1}}\}$.^{*} So in Algorithm 2 we test the diagonal element against a tolerance VTOL

^{*}For if $\mathbf{W} = \mathbf{Q}\mathbf{R}$ is the Gram-Schmidt decomposition of the unit column vectors in $\mathbf{W}$, where $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{R}$ is an upper triangular matrix, then $\mathbf{R} = \mathbf{L}^T$, where $\mathbf{L}$ is the Cholesky lower triangle factor $\mathbf{W}^T \mathbf{W}$ (Theorem 5.2.2 in [22]). During the Gram-Schmidt process, the k^{th} diagonal entry of $\mathbf{R}$ (and hence of $\mathbf{L}$ and thus also $\mathbf{H}$) is the component of the k^{th} vector orthogonal to the subspace spanned by the $k-1$ vectors of lower index. That is, the diagonal entry is the sine of the angle the k^{th} vector makes with the subspace spanned by the more recent lower-indexed vectors.

and if it too small the k^{th} vector is deemed to be too linearly dependent on the lower-indexed vectors. This results in the vector $\delta \mathbf{s}_k$ (and corresponding vector $\delta \mathbf{x}_k$) being dropped from the set of stored vectors, with corresponding entries in H corrected to reflect the deletion of the vector. If VTOL is small, it would take near-linear dependence for a vector to be dropped. But in the setting of nonlinear acceleration where subsequent iterations can give conflicting information when the true Jacobian $\mathbf{r}'$ is nonconstant, we use the fairly large value of VTOL=0.1 to aggressively delete vectors that are nearly linearly dependent with lower-indexed, and hence more recent, vectors. If the size of the set of retained vectors would exceed a maximum value MVEC despite possible deletions due to linear dependence, the oldest vector is then dropped.

Algorithm 1: NACCEL($\mathbf{s}, \mathbf{dx}, \mathbf{v}, \mathbf{w}^\perp, \mathbf{mvec}, \mathbf{vtol}$)

[Replace preconditioned residual vector $\mathbf{s}$ with accelerated correction vector $\mathbf{dx}$]
 [$\mathbf{v}, \mathbf{w}^\perp$ represent decomposition to $\mathbf{dx}$ according to (12)]
 [$\mathbf{mvec}$ maximum number of acceleration vectors]
 [$\mathbf{vtol}$ angular subspace rejection tolerance]
 If (this is first call to routine) then
 $N \leftarrow 0; \mathbf{v} \leftarrow \mathbf{0}; \mathbf{w}^\perp \leftarrow \mathbf{s}; \mathbf{dx} \leftarrow \mathbf{s}; \mathbf{s}_{prev} \leftarrow \mathbf{s}; \mathbf{dx}_{prev} \leftarrow \mathbf{dx};$ return
 Else
 [Shift previous vectors to right]
 For $i=N, N-1, \dots, 1$
 $\delta \mathbf{s}_{i+1} \leftarrow \delta \mathbf{s}_i$
 $\delta \mathbf{x}_{i+1} \leftarrow \delta \mathbf{x}_i$
 [Strict upper triangle of $H(i, j)$ holds inner products $\delta \mathbf{s}_i \cdot \delta \mathbf{s}_j$.]
 [Shift inner products in strict upper triangle of H .]
 For $j = 1, \dots, i-1$
 $H(j+1, i+1) \leftarrow H(j, i)$
 $N \leftarrow \max(N+1, \mathbf{mvec})$
 [Normalize new $\delta \mathbf{s}, \delta \mathbf{x}$ vector pair so that $\|\delta \mathbf{s}\| = 1$.]
 $\delta \mathbf{s}_1 \leftarrow (\mathbf{s}_{prev} - \mathbf{s}) / \|\mathbf{s}_{prev} - \mathbf{s}\|$
 $\delta \mathbf{x}_1 \leftarrow \mathbf{dx}_{prev} / \|\mathbf{s}_{prev} - \mathbf{s}\|$
 [Compute inner product of latest vector $\delta \mathbf{s}_1$ with previous vectors]
 For $i = 2, 3, \dots, N$
 $H(1, i) \leftarrow \delta \mathbf{s}_1 \cdot \delta \mathbf{s}_i$
 [Perform Cholesky factorization of inner product matrix, with rejection of nearly linearly dependent vectors]
 Call **CholeskyWithRejection**($H, \mathbf{vtol}, N, \{\delta \mathbf{s}_i\}_{i=1}^{i=N}, \{\delta \mathbf{x}_i\}_{i=1}^{i=N}$)
 [Compute projection of $\mathbf{s}$ on range of $W = [\delta \mathbf{s}_1 \mid \delta \mathbf{s}_2 \mid \dots \mid \delta \mathbf{s}_N]$ by solving $LL^T \mathbf{c} = W^T \mathbf{s}$ where L is Cholesky factor stored in lower triangle of H .]


```

do j=1,N
     $c(j) \leftarrow \mathbf{s} \cdot \delta \mathbf{s}_j$ 
    For i=1,j-1
         $c(j) \leftarrow c(j) - H(j, i) * c(i)$ 
     $c(j) \leftarrow c(j)/H(j, j)$ 
For j=N,N-1,...,2,1
    For i=j+1,N
         $c(j) \leftarrow c(j) - H(i, j) * c(i)$ 
     $c(j) \leftarrow c(j)/H(j, j)$ 
[Compute accelerated correction]
 $\mathbf{v} \leftarrow \mathbf{0}$ 
 $\mathbf{w}^\perp \leftarrow \mathbf{s}$ 
For k=1,...,N
     $\mathbf{v} \leftarrow \mathbf{v} + c(k) \delta \mathbf{x}_k$ 
     $\mathbf{w}^\perp \leftarrow \mathbf{w}^\perp - c(k) \delta \mathbf{s}_k$ 
 $\mathbf{dx} \leftarrow \mathbf{v} + \mathbf{w}^\perp$ 
[Save  $\mathbf{s}$ ,  $\mathbf{dx}$  for next call and return]
 $\mathbf{s}_{\text{prev}} \leftarrow \mathbf{s}$ ;  $\mathbf{dx}_{\text{prev}} \leftarrow \mathbf{dx}$ ; return

```

Algorithm 2: CholeskyWithRejection($H, \mathbf{vtol}, N, \{\delta \mathbf{s}_i\}_{i=1}^{i=N}, \{\delta \mathbf{x}_i\}_{i=1}^{i=N}$)

[For inner product matrix $H(i, j) = \delta \mathbf{s}_i \cdot \delta \mathbf{s}_j$ in strict upper triangle of H , find and write Cholesky factor into lower triangle of H .

That is, if $W = [\delta \mathbf{s}_1 \mid \delta \mathbf{s}_2 \mid \dots \mid \delta \mathbf{s}_N]$, strict upper triangle of H contains strict upper triangle of $W^T W$ and we compute lower triangle matrix L such that $LL^T = WW^T$ and write L into lower triangle of H .

If diagonal pivot $H(k, k)$ is too small (relative to $\mathbf{vtol}$), corresponding vector $\delta \mathbf{s}_k$ is dropped due to near linear dependence on previous vectors]

```

 $H(1,1) \leftarrow 1$ ;  $k \leftarrow 2$ 
1: Do while ( $k \leq N$ )
    For  $j = 1, k - 1$ 
         $H(k, j) \leftarrow H(j, k)$  [Copy inner product into lower triangle of H]
    For  $i = 1, j - 1$ 
         $H(k, j) \leftarrow H(k, j) - H(k, i) * H(j, i)$ 
     $H(k, j) \leftarrow H(k, j)/H(j, j)$ 
     $H(k, k) \leftarrow 1$ 
    For  $j = 1, k - 1$ 
         $H(k, k) \leftarrow H(k, k) - H(k, j)^2$ 

```

If $(H(k,k) < \mathbf{vtol}^2)$ then
 [Diagonal pivot too small due to near linear dependence.]
 For $i=k, k+1, \dots, N-1$
 [Drop $\mathbf{s}_k$ since sine of angle between $\mathbf{s}_k$ and span $\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_{k-1}\}$ is less than $\mathbf{vtol}$]
 $\delta \mathbf{s}_i \leftarrow \delta \mathbf{s}_{i+1}$
 [Drop corresponding vector $\delta \mathbf{x}_k$]
 $\delta \mathbf{x}_i \leftarrow \delta \mathbf{x}_{i+1}$
 [Compress H to reflect deletion of inner products involving $\delta \mathbf{s}_k$]
 For $j=k, N$
 For $i=1, k-1$
 $H(i, j) \leftarrow H(i, j+1)$
 For $i=k, j-1$
 $H(i, j) \leftarrow H(i+1, j+1)$

 $N \leftarrow N-1$; goto **1**
 Else [Can take square root to get Cholesky diagonal element]
 $H(k,k) \leftarrow \sqrt{H(k,k)}$; $k \leftarrow k+1$
 return

2.4. Reuse of Krylov Subspace across Time Steps

To reduce the need for a ‘good’ preconditioner $\mathbf{K}$ in (4), i.e., one that is close to and perhaps as expensive to evaluate as the exact inverse Jacobian $(\mathbf{r}')^{-1}$, extra work can be done by the Krylov accelerator. We do this by retaining vectors in the matrices $\mathbf{V}$, $\mathbf{W}$ over several time steps. That is, we make the assumption that $\mathbf{A}$ at recent previous time steps is the same as $\mathbf{A}$ at current the current time step. Thus we can keep some of the vector pairs $\delta \mathbf{x}_i$, $\delta \mathbf{s}_i$ that were obtained before time t . Thus the structure of the matrices $\mathbf{V}$, $\mathbf{W}$ that were used in computing the latest $\mathbf{dx}$ will be

$$\begin{aligned} \mathbf{W} &= [\delta \mathbf{s} \text{ vectors buildup at time } =t | \delta \mathbf{s} \text{ vectors buildup for times } < t] \text{ and} \\ \mathbf{V} &= [\delta \mathbf{x} \text{ vectors buildup at time } =t | \delta \mathbf{x} \text{ vectors buildup for times } < t]. \end{aligned}$$

Note that $\delta \mathbf{x}$, $\delta \mathbf{s}$ pairs will have been computed only at times which use at least two residual evaluations.

As mentioned previously, in order to keep the size of $\mathbf{V}$ and $\mathbf{W}$ under control, at most m =MVEC acceleration vectors are retained with the oldest vectors being dropped. In this work, we have used MVEC=10.

2.5. Partial NACCEL correction

If we use an acceptable size of the norm $\|\mathbf{s}\|$ of the preconditioned residual as a termination condition, then we can check whether $\|\mathbf{w}^\perp\| = \|\mathbf{s}(\mathbf{x}^*)\|$ in (11) is acceptably small. If so, we accept only the first pass of the NACCEL correction (i.e., we set $\mathbf{z} = \mathbf{0}$ in (11)). This allows us to confidently pass to the next timestep *without* a further evaluation of the residual at the

current timestep to verify that it is acceptably small. This can represent a crucial savings in computational cost. In the best case scenario, timestepping can occur with corrections evaluated and accepted most commonly with only a single residual evaluation per timestep, resulting in very little additional cost over a unidirectionally coupled or uncoupled simulation.

Nevertheless on those timesteps where the tested residual is not acceptably small, the full NACCEL correction must be accepted to allow for the enlargement of the $\mathbf{V}$, $\mathbf{W}$ matrices which may be necessary to ultimately reduce the residual to an acceptable size.

In short, although the *full* correction $\mathbf{dx} = \mathbf{v} + \mathbf{w}^\perp$ must be applied in order for NACCEL to work properly and enrich its input/output database, it is in fact permissible to apply the *partial* correction $\mathbf{dx} = \mathbf{v}$ if it is at the end of a timestep. Thus, we split the NACCEL correction into its $\mathbf{v}$, $\mathbf{w}^\perp$ parts and use the size of $\|\mathbf{s}(\mathbf{x}^*)\| = \|\mathbf{w}^\perp\|$ as a termination criterion. If it is too large, we apply the full NACCEL correction $\mathbf{dx} = \mathbf{v} + \mathbf{w}^\perp$. Otherwise, we apply only the partial correction $\mathbf{v}$ and pass to the next timestep.

Remark—At the “partially” corrected point $\mathbf{x}^* \equiv \mathbf{x}_k - \mathbf{v}$, we have $\mathbf{s}(\mathbf{x}^*) = \mathbf{w}^\perp$ which is orthogonal to $\text{range}(\mathbf{W}) = \mathbf{A} \cdot \text{range}(\mathbf{V})$ and $\mathbf{v}$ minimizes $\|\mathbf{A} \delta \mathbf{x} - \mathbf{s}_k\|$ over all $\delta \mathbf{x} \in \text{range}(\mathbf{V})$. Assume $\mathbf{A}$ is positive definite. The Generalized Minimal Residual (GMRES) method produces iterates that minimize the norm of the residual $\|\mathbf{A} \delta \mathbf{x} - \mathbf{s}_k\|$ over all vectors $\delta \mathbf{x}$ in the Krylov subspace $K_N(\mathbf{A}, \mathbf{s}_0)$ which is $\text{range}(\mathbf{V})$. Thus if we only apply the partial correction $\mathbf{x}^* = \mathbf{x}_k - \mathbf{v}$ we are accepting an iterate that would be generated by the popular GMRES method. This shows the close relationship between NACCEL iterates and iterates from classical GMRES in the ideal case that $\mathbf{s}(\mathbf{x}) = \mathbf{A} \mathbf{x} - \mathbf{b}$ is affine linear with $\mathbf{A}$ positive definite.

2.6. Residual Used for Determining Interfacial Pressures

In this study, we assume the pleural pressure is known, as is the atmospheric pressure at the nose or mouth. Alternatively, one could determine regional ventilation from live imaging data and apply the resulting flows to the parenchymal units that ventilate the terminal airways in an ODE system. Examples of the experimental characterization of both pleural pressure (e.g. [23, 24, 25, 26, 27, 28, 29]) and regional ventilation (e.g. [30, 31, 32, 33]) can be found in the literature. In either case, intermediary pressures are unknown and are the interface conditions that we wish to solve for. Since the conducting airways span many generations and many scales, it is computationally expensive to represent each airway explicitly in 3D. Thus the lower part of the lung, where the fluid-dynamics are less interesting are represented by a lower-dimensional model (in this manuscript as a set of ODEs), one ODE system for each boundary outlet. In this case, an obvious – though unstable – choice for the residual would be the mismatch between the CFD flow and the ODE flow at the interface:

$$r_i \equiv Q_i^{cfd} - Q_i^{ode} = 0, \quad 1 \leq i \leq M, \quad (13)$$

where $r_i = r_i(\mathbf{p})$ is the i^{th} component of the residual $\mathbf{r} = (r_1, r_2, \dots, r_M)$, which is a function of pressures $\mathbf{p} = (p_1, p_2, \dots, p_M)$ at each CFD outlet; $Q_i^{cfd}(\mathbf{p})$ is the massflux in the 3D CFD geometry for the i^{th} outlet and $Q_i^{ode}(\mathbf{p})$ is the corresponding massflux in the distal ODE system. We wish to use the iteration in (4) and in what follows we interchangeably use $\mathbf{p}$ (our vector of unknown pressures) and $\mathbf{x}$ (the vector of generic independent variables in (4)). In this section i, j subscripts denote outlet number; $n, n + 1$ is used for iteration number.

The problem with this simple scheme is illustrated in Figure 1. Due to inertia, small pressure increments from p_i to $p_i + d p_i$ result in very little change in flow. Thus, large pressure changes would need to be applied in order to get r_i to change to zero. On the next time step,

however, the inertia would keep the flow caused by the large pressure changes on the previous time step going in the same direction, and so a large and opposite pressure change would have to be applied to continue to zero out r_i (Figure 1). The result would be that the pressure changes would alternate in an unstable fashion leading to spurious pressure oscillations at the interface and the code would fail.

The solution to this problem is to modify the residual such that instead of equating the flows between CFD and ODE, we equate the pressure drops on either side of the CFD-ODE interface that occur within one diameter (see Figure 2). Assuming the interface occurs in the middle of a cylindrical branch, we note that over a length of one diameter, the pressure drop due to resistance (R) is $R_i Q_i$ and the pressure drop due to inductance (L) is $L_i \dot{Q}_i$. So we equate

$$\begin{aligned} r_i &= R_i Q_i^{ODE} + L_i \dot{Q}_i^{ODE} - (R_i Q_i^{CFD} + L_i \dot{Q}_i^{CFD}) = 0 \\ &\approx R_i Q_i^{ODE} + L_i \frac{Q_i^{ODE} - Q_i^{ODE}(t - \Delta t)}{\Delta t} - \left(R_i Q_i^{CFD} + L_i \frac{Q_i^{CFD} - Q_i^{CFD}(t - \Delta t)}{\Delta t} \right) = 0, \end{aligned} \quad (14)$$

where $Q_i^{ODE}(t - \Delta t)$ and $Q_i^{CFD}(t - \Delta t)$ are the ODE and CFD flow values from the previous timestep. That is we demand that the pressure drops on either side of the interface over a distance one diameter on either side are equal. In the absence of the inductive term (which physically represents inertia), the residual would equate $R_i Q_i$ and since R_i over a distance one diameter is equal on both sides of the interface, it would amount to equating Q_i , which reduces to the simple residual presented in (13). Thus, the introduction of the $L_i \dot{Q}_i$ term accounts for the acceleration of the flow caused by the contemplated pressure change $-dp_i$. It is clear the residual should take into account this acceleration in order to result in a stable coupling. Intuitively it is natural to select an arbitrary multiplier in front of the flow acceleration in (14) to be one such that the acceleration term is physically comparable to the flow imbalance term. That is, given a residual choice $\alpha Q + \beta \dot{Q}$, by choosing $\alpha = R$ and $\beta = L$, we obtain a residual that has both required features (flow balance and acceleration control) combined in a ratio that has physical significance (the ratio of resistive pressure drop to inductive pressure drop). This makes sense because if there is a total pressure drop available over the interface pipe, it will put a physical bound on the inductive pressure drop.

Using any time step Δt (so that $\dot{Q} = (Q(t) - Q(t - \Delta t))/\Delta t$), any pressure change $-dp$ applied to equate the flows will be limited by the inductive pressure drop that change would imply. The residual thus mimics the physical situation where any flow imbalance would be minimized in a uniform pipe (assumed to be the same on both sides of the interface) but the process would be limited by the inductive pressure drops generated by changes the fluid makes in order to equate the flows.

Although necessary for numerical stability, the Q mismatches are oscillatory in time and locally average to zero so that the simple flow mismatch residual (13) is acceptably small, as seen in a typical plot of CFD and ODE flows versus time (Figure 9). In Figure 11 we see that over a breathing cycle in a typical case that the spectrum of the Jacobian of the residual $\mathbf{r}(\mathbf{x})$ in Equation 14 is almost completely real, positive, and well-bounded from zero. In fact $\mathbf{r}'$ is nearly diagonal and slowly varying in time with its diagonal elements usually between 0.5 and 3. Since it is so well-behaved, we employ a very inexpensive preconditioner: we let $\mathbf{K}$ be the diagonal matrix whose diagonal entries are reciprocals of the corresponding diagonal entries in $\mathbf{r}'$ at a certain time, with $\mathbf{K}$ then only updated relatively infrequently. If necessary, however, $\mathbf{K}$ could more generally represent exact multiplication by the inverse of an infrequently updated Jacobian matrix by infrequently computing an LU factorization of the Jacobian [22].

3. Methods

3.1. CFD Calculation

CFD simulations were computed with *OpenFOAM*[®] library (*OpenFOAM* is an open source C++ computational continuum mechanics software and is a registered trademark of OpenCFD Ltd., Reading, UK; www.openfoam.com). The airflow predictions were based on the laminar 3D incompressible Navier-Stokes equations for fluid mass and momentum

$$\nabla \cdot \mathbf{u} = 0 \quad (15)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{u} \quad (16)$$

where ρ is the density, ν is the kinematic viscosity, $\mathbf{u}$ is the fluid velocity vector, and p is the pressure. An adaptive time-stepping algorithm was utilized for the selection of the optimum time step. Specifically, the PIMPLE (merged PISO-SIMPLE) algorithm was used (<http://www.openfoam.com/docs/>). The algorithm adjusted the time step so as to limit the Courant number to be less than 10.0 for the model presented in Section 4.1, and less than 25.0 for the models presented in Sections 4.2 & 4.3. Such large Courant numbers are permissible because the method is fully implicit.

3.2. Multiscale Coupling Algorithm

Our enhanced nonlinear Krylov accelerator NACCEL was implemented as a C++ boundary class in *OpenFOAM*[®]. The iterative algorithm at a time step is given in Algorithm 3. The preconditioner – equal to the inverse of the diagonal of the Jacobian – is computed at the initial timestep and then additionally if the current timestep Δt differs by more than a factor of 3 from the timestep $(\Delta t)_{jac}$ in force at the time the preconditioner is evaluated[†]. A preconditioner refresh will also be triggered if the timestep has not converged after 10 iterations. Whenever the preconditioner is newly refreshed, the subsequent call to NACCEL is considered a “first call” in Algorithm 1. That is $N \leftarrow 0$ in Algorithm 1 and any previously accumulated acceleration vectors are discarded. Termination occurs if the infinity norm of the preconditioned residual (estimated by the infinity norm of $\mathbf{w}^\perp$ according to the reasoning in Section 2.5) is less than a user-specified tolerance TOL.

4. Simulation Results

Here we present the results for two bi-directionally coupled cases. In Section 4.1 we compare the predictions of a hybrid CFD-ODE model against an ODE-only model of pulmonary airflow in an idealized geometry. The focus is to validate the method by comparing multiscale predictions against known results for the entire pulmonary airway tree. There are limitations to such a comparison. The ODE-only model is a simplification of the multiscale CFD-ODE model, and of reality. Nevertheless, it does provide a useful benchmark for the behavior of the system. Subsequently in Section 4.2 we couple multiple sets of ODEs describing the distal lung to an imaging-based human lung geometry. Finally in Section 4.3, to illustrate the efficiency of the method with respect to an uncoupled method, and to illustrate how uncoupled multiscale analysis misrepresents the physics, we present the results of multiple uni-directional sets of ODEs describing the distal lung

[†]If the timestep sizes are too different, the preconditioner will be poor due to the presence of inductive pressure drop contribution in

the residual which varies as $\frac{1}{\Delta t}$.

coupled to the imaging-based human lung geometry. Because in this latter case, the coupling is not bi-directional, mechanical equilibrium is not enforced on the interface. All simulations were carried out on the machine, *Navier*, a cluster composed of 14 nodes, each with dual socket six core *Intel*[®] Westmere based processors running at 2.8 Ghz. Every node on the system has a RAM of 48 GB. Fast interprocessor communication between the processors is obtained using Mellanox, ConnectX QDR Infiniband.

Algorithm 3: Time Integration Loop

[Step through time, continuously matching CFD and ODE pressure drops at interface outlets]

Obtain next timestep Δt from CFD code (i.e., determined by element Courant numbers)

Let initial pressure iterate $\mathbf{p}$ be linear extrapolation of converged pressures from last two timesteps

For $n = 1, 2, \dots$

Call CFD code to evaluate outlet flows $Q_i^{\text{CFD}}, 1 \leq i \leq M$, using outlet pressures $\{p_i\}_{i=1}^{i=M}$

Call ODE system to evaluate outlet flows $Q_i^{\text{ODE}}, 1 \leq i \leq M$, using outlet pressures $\{p_i\}_{i=1}^{i=M}$

$$r_i \leftarrow R_i Q_i^{\text{ODE}} + L_i \frac{Q_i^{\text{ODE}} - Q_i^{\text{ODE}}(t - \Delta t)}{\Delta t} - \left(R_i Q_i^{\text{CFD}} + L_i \frac{Q_i^{\text{CFD}} - Q_i^{\text{CFD}}(t - \Delta t)}{\Delta t} \right), \quad 1 \leq i \leq M$$

If (no Jacobian has been computed or $\Delta t \notin \left[\frac{1}{3}(\Delta t)_{jac}, 3(\Delta t)_{jac} \right]$ or $10 \mid n$) then

[Compute Jacobian by finite difference]

$$J_{ij} \leftarrow (r_i(p_1, p_2, \dots, p_j + h, \dots, p_M) - r_i(\mathbf{p})) / h, \quad 1 \leq i, j \leq M$$

[Compute diagonal preconditioner]

$$K_{ii} = 1/J_{ii}, \quad 1 \leq i \leq M \quad (K_{ij} \leftarrow 0 \text{ if } i \neq j)$$

$$(\Delta t)_{jac} \leftarrow \Delta t$$

$$\mathbf{s} \leftarrow \mathbf{K} \mathbf{r}$$

$(\mathbf{v}, \mathbf{w}^\perp) \leftarrow \text{NACCEL}(\mathbf{s})$ [Note: If preconditioner is new, treat call to NACCEL as a “first call”]

$$\mathbf{p} \leftarrow \mathbf{p} - \mathbf{v}$$

If $\|\mathbf{w}^\perp\| < \text{TOL}$ then

For $i = 1, 2, \dots, M$

$$\begin{aligned} p_i(t) &\leftarrow p_i \\ Q_i^{\text{ODE}}(t) &\leftarrow Q_i^{\text{ODE}} \\ Q_i^{\text{CFD}}(t) &\leftarrow Q_i^{\text{CFD}} \end{aligned}$$

break; [Proceed to next time step]

Else

$$\mathbf{p} \leftarrow \mathbf{p} - \mathbf{w}^\perp$$

4.1. Validation Case—Weibel Geometry

A number of researchers have modeled the human tracheobronchial tree up to 24 generations of airways using equivalent circuits (e.g. [7, 8, 9, 10, 11]). The main advantage of these models is that they represent the compliant mechanics of the entire lung airway tree. These models are based on systems of equations wherein each generation is represented by an ordinary differential equation expressed in terms of resistances, inductances, and compliances (RLC) that have been parameterized from experimental data. The model of [8] achieved perhaps the highest level of sophistication due to a formulation in which the resistances, inductances were made to be a function of a nonlinear airway-pressure relationship. However, it was formulated for simulating mechanical ventilation. As the boundary conditions in mechanical ventilation are somewhat more straightforward than autonomous respiration, this would not represent as stringent a test of our coupling approach. Instead, herein we adopt an approach based on a modification of the model presented in [9]. Our model differs from the original model in three ways: 1) the boundary conditions consist of the applied time derivative of a pleural pressure waveform and zero pressure at the trachea; 2) the airways are allowed to expand and contract, rendering the airway resistance and inductance a function of airway caliber; and 3) generations 17-24 are replaced by a compliant acinar unit. Other models are of course possible and perhaps even preferable. Our intent is to demonstrate the robustness and efficiency of the coupling method with as simple a model as possible that presents a like degree of nonlinearity to the true system. Importantly, by design, our method is not wed to a particular model.

The geometry (Figure 3) for this validation case was based on the well-known Weibel geometry [34]. Briefly, a four-generation symmetric human tracheobronchial tree was constructed and scaled to a volume of 2.3 liters to match the volume in the original paper [9]. The branching angle was assumed to be 60° and the azimuthal angle (angle of rotation of the plane of bifurcation originating in each daughter branch) was assumed to be 60° . To construct the Weibel model it was assumed that the daughter branch began its separation at the bifurcation point as the frustum of a cone. The large diameter of the frustum was set equal to the diameter of the parent branch and the small diameter of the frustum was set equal to the diameter of the daughter branch. The length of the frustum was assumed to be one-tenth the length of the daughter branch.

The resulting triangulated surface mesh was constrained, smoothed and adapted to the gradient-limited local feature size [35], and subsequently a hybrid prism/tetrahedral mesh was generated in BioGeom (www.simtk.org/home/biogeom). This hybrid mesh was then converted to a polyhedral mesh using the polyDualMesh utility in *OpenFOAM*[®]. The final mesh consisted of $\approx 250,000$ polyhedra.

The lower-dimensional lung model consisted of lumped RLC circuits based on the assumption of laminar Poiseuille flow through branched cylindrical tubes. Resistances, inductances and compliances, were defined as:

$$R_i = \frac{8\mu l}{\pi r_i^4}; \quad L_i = \frac{\rho}{\pi r_i^2}; \quad C_i = \frac{\Delta V_i}{\Delta P_i} \quad (17)$$

where μ and ρ are the viscosity and density of air; i is the generation number and r and l are the airway radius and length for that generation; V and P are the volume and pressure,

respectively. The density of air was assumed to be 1.3 kg/m^3 and kinematic viscosity was assumed to be $1.68 \times 10^{-5} \text{ m}^2/\text{s}$ to match the values used in [9].

The governing equations for the lower-dimensional model were as follows. For the conducting airways (generations 1 through 16),

$$\begin{aligned} P_i - P_{i-1} &= R_i Q_i + L_i \frac{dQ_i}{dt} \\ Q_{i+1} - Q_i &= C_i \cdot \frac{d}{dt} (P_i - P_{pl}). \end{aligned} \quad 1 \leq i \leq 16 \quad (18)$$

Here, P_i is the pressure at the distal end of the i^{th} generation, Q_i is the rate of airflow through the i^{th} generation, P_{pl} is the intrapleural pressure and t is the time. P_0 is the tracheal pressure; if the $i_0 + 1^{\text{th}}$ generation of the ODE model is attached to an outlet of the CFD model, then the first i_0 generations are omitted from (18). In this case, P_{i_0} represents the pressure at the interface between ODE and CFD models which is solved by iterative coupling. The respiratory zone (representing the generations of the lung beyond 16) are represented by a final acinar unit similar to the form of (18).

$$\begin{aligned} P_a - P_{16} &= R_a Q_a \\ -Q_a &= C_a \cdot \frac{d}{dt} (P_a - P_{pl}). \end{aligned} \quad (19)$$

The initial values used at $t = 0$ for R_i , L_i , C_i , $1 \leq i \leq 16$ are as in [9] and R_a , C_a values are given in [36].[‡] The linear ODE system for the variables $\{Q_1, P_1, Q_2, P_2, \dots, Q_{16}, P_{16}, Q_a, P_a\}$ is tridiagonal and thus is rapidly solved by the method in Section 2.4 of [37]. To account for the changing caliber of the airways, and thus the influence of transmural pressure on the airway resistance and inductance, at each time step Δt after system (18)-(19) is solved, the resistances and inductances given in Equation (17) are updated as follows:

$$\begin{aligned} R_i^t &= R_i^{t=0} \left(\frac{A_i^{t=0}}{A_i^t} \right)^2 \\ L_i^t &= L_i^{t=0} \left(\frac{A_i^{t=0}}{A_i^t} \right) \end{aligned} \quad 1 \leq i \leq 16 \quad (20)$$

Here A_i is the cross-sectional area of airways in the i^{th} generation updated using

$$\begin{aligned} dV_i &= (Q_{i+1} - Q_i) \Delta t \\ dA_i &= dV_i / (l_i \cdot 2^{i-i_0-1}) \\ A_i^t &= A_i^t + dA_i. \end{aligned} \quad 1 \leq i \leq 16 \quad (21)$$

Here V_i is the total volume of the i^{th} generation and l_i is the length of the airways of the i^{th} generation. The factor 2^{i-i_0-1} represents the fact that if generation i is attached to the CFD model at generation i_0 , that generation's volume is divided among 2^{i-i_0-1} airways.

To facilitate the comparison of the multiscale CFD-ODE model to the ODE solution, the compliance in the first four generations was set to zero. Boundary conditions (Figure 3) consisted of zero (atmospheric) pressure at the trachea and an applied intrapleural pressure derivative. When the inspiratory muscles relax at the moment between inspiration and expiration, the derivative of the intrapleural pressure will jump. For our example waveform, the intrapleural pressure and its derivative are shown in Figure 4A. Here we use the following piecewise smooth expression for intrapleural derivative:

[‡] $R_a = R_l$ and $C_a = \frac{1}{E_p}$ in the notation of [36].

$$\frac{dP_{pl}}{dt} \begin{cases} \frac{A\pi}{T} \cos\left(\frac{\pi t}{T} + \pi\right) & 0 \leq \hat{t} < \frac{1}{2}T \\ \frac{A\pi}{T} \cos\left(\frac{\pi t}{T} + \frac{3}{2}\pi\right) & \frac{1}{2}T \leq \hat{t} < T \\ \frac{A\pi}{T} \cos\left(\frac{\pi t}{T}\right) & T \leq \hat{t} < \frac{3}{2}T \\ \frac{A\pi}{T} \cos\left(\frac{\pi t}{T} + \frac{1}{2}\pi\right) & \frac{3}{2}T \leq \hat{t} < 2T \end{cases} \quad (22)$$

where T and A are the period and amplitude of the intrapleural pressure, and $\hat{t}$ is given by $\hat{t} = t - 2T \left\lfloor \frac{t}{2T} \right\rfloor$. The magnitude of the intrapleural pressure was set to 1000 Pa.

The results of this validation exercise can be seen in Figure 4. Overall, the two sets of results showed excellent agreement. Of particular note is the near match between interface flow (4B) over the entire simulation, indicating mechanical equilibrium. It is important to view the similarities of these two sets of model predictions within the context of the differences between the two models. Specifically, Equation (18) assumes Poiseuille laminar flow, whereas the peak Reynolds and Womersley numbers at trachea for this simulation were 4616 and 2.2, respectively. In order for the Poiseuille laminar flow assumption to be valid, the Reynolds number should be less than about 1000.0 and Womersley number (Wo) must be less than about 5.0. Moreover, Equation (18) does not account for viscous losses in the airway bifurcations during inhalation and exhalation, whereas these losses are inherent in the CFD calculation. In order for the bifurcational losses to be negligible, the Reynolds number (Re) must be less than 12.5 [38]. Given these inevitable differences, it is clear that the multiscale formulation may be considered accurate. Given the near optimal performance of the method in terms of computational cost and stability, it is also clear that it may be considered robust.

Some discussion is warranted for the difference in multiscale versus pure-ODE interface pressures seen in Figure 4B. Note on exhalation, the pressures nearly match, whereas on inhalation there is a significant difference between the two that cannot be explained away by the relatively high Reynolds number flow alone. During inhalation, for stability, *OpenFOAM*[®] automatically converts to zero total pressure (dynamic plus static pressure) at the trachea, which corresponds to zero static pressure at a quiescent far-field.[§] The Weibel model does not explicitly capture that far-field in its geometry. During inhalation at the trachea, the dynamic pressure is positive; thus to maintain zero total pressure, a negative static pressure is applied, accounting for the differences seen in the interface pressures in Figure 4B. This is a case where the CFD boundary conditions are actually more physically meaningful than those imposed in the ODE. In the human model to be presented in the next subsection (Figure 8), the far-field is explicitly represented as a cylinder beyond the face of the subject. This extension of the computational domain assures the physicality of the boundary condition as well as assuring that the orientation of the flux into the mouth is natural.

Also of note is the iteration history of the multiscale simulations (Table 1 and Figure 5). Optimal performance would be one CFD evaluation per time step for all time steps. Despite the relatively high pressure amplitudes, 99.96% of the time steps required only a single CFD evaluation in the “MOD NEWT + NACCEL” case where NACCEL was used with inverse

[§]Unlike the case for a simple system of ODEs such as (18)-(21), in 3D CFD calculations an inflow condition of prescribed static pressure p is unstable since fluid velocity $\mathbf{v}$ can rapidly increase with the standard zero Neumann condition $\Delta \mathbf{v} \cdot \mathbf{n} \approx 0$. Specification of

total pressure $p + \frac{\rho}{2} \|\mathbf{v}\|^2$ (sum of static pressure and dynamic pressure) controls $\mathbf{v}$ and is stable. For an outlet, specification of static pressure is appropriate as a boundary condition. Thus a 3D CFD code typically converts from specified total pressure to specified static pressure boundary condition if flow at an inlet reverses making it an outlet and vice versa.

diagonal Jacobian preconditioner. This performance is near-optimal and is comparable to that of the monolithic approach, but with vastly greater freedom. Thus, we can state that our bidirectional coupled runs have little more cost than an uncoupled CFD-only simulation, which would of course have un-physiological boundary conditions. In contrast, from Table 1 for the “MOD NEWT” method (i.e., Modified Newton Method with infrequently updated diagonal preconditioner and no NACCEL acceleration), significantly more residual evaluations were required.

4.2. Coupled Imaging-Based Human Multiscale Model

To illustrate the performance of the bi-directional coupling method with multiple outlets in a complex imaging-based geometry we coupled 117 different sets of ODEs based on Equations (18)-(21) to a CT based human oral cavity, larynx and tracheobronchial tree.

4.2.1. Imaging Data

The human CFD geometry was based upon multi-slice CT imaging of the head and torso of a female (84 years of age) using a GE LightSpeed VCT at 623 mm isotropic resolution across a $31.9 \times 31.9 \times 46.1$ cm field of view. The Institutional Review Boards of the University of Washington and PNNL approved all human volunteer work conducted under this study.

4.2.2. Construction of the CFD model from imaging data

Airway tissue imaging data were segmented as described in [39]. Human lung segmentations did not require pre-filtering and relied upon intensity thresholding followed by visual validation and repair as described previously [40]. To better mimic physiological breathing conditions and the influence of the mouth on the entry of air into the respiratory tract, a cylinder capturing the contours of the face was added to the segmentation. Addition of the cylinder also has the effect of representing the far-field for the imposition of a zero total pressure boundary condition.

A triangulated surface was extracted by applying a variant of the Marching Cubes algorithm [41] to the segmented CT lung image. After isosurface extraction, the centerline of the triangulated lung surface was decomposed into a topologically correct skeleton. During grid generation, the skeleton was used to automatically truncate lung geometries and introduce boundary facets at a user-specified generation or airway diameter [42, 43], resulting in 117 outlets, not including the trachea. The truncated surface of the pulmonary airways was then joined with the similarly extracted surface for the upper respiratory tract using the STL surface editor *MAGICS*[®] (*MAGICS* is a registered trademark of Materialise, Plymouth, MI, USA; www.materialise.com/Magics).

Once a final closed surface was produced, the triangulated mesh was adapted to the gradient-limited feature size [35], and subsequently a hybrid prism/tetrahedral mesh was generated in BioGeom. This hybrid mesh was then converted to a polyhedral mesh using the polyDualMesh utility in *OpenFOAM*[®]. The final mesh consisted of polyhedral, prism and hexahedral elements.

The resulting volume mesh had 117 terminal airway outlets ending in different generations, and consisted of ≈ 1.0 million cells. A set of independent ODEs based on Equation (18) was registered to the equivalent outlet generation number based on outlet diameter. The outlet diameter was automatically determined and the generation corresponding to that diameter was determined from Ginzburg et al. [9]. This adjustment was necessary (see Figure 10) because the imaging based geometry is not symmetric or simple. Relying on the nominal generation number would have resulted in unrealistic flows at the outlet interfaces. These

subtree boundary conditions were subsequently bi-directionally coupled at each outlet (Figure 6). Hence, every outlet in the 3D model had a unique distal ODE set coupled to it. As with the validation case, each set of ODEs was solved with a tridiagonal matrix solver implemented in C++.

To mimic physiological breathing conditions, atmospheric pressure was applied at the mouth and a pleural pressure derivative was applied at each generation of each ODE set, analogous to the previous case. The subtrees coupled at each outlet were registered by diameter-equivalent generation number, considering the data provided in [9]. It is important to note that we make no claim that the modeling approach taken here is perfect. For simplicity, and to focus on the method, not the model, we applied lower-dimensional models parameterized from the literature, not subject-specific imaging data. The effort made is to be consistent with what has already been presented, for the sake of clarity.

The execution time and related statistics for this human respiration model can be seen in Table 2. To speed up execution, the maximum Courant number was set to 25.0, and the model was computed on *Navier* (described above). The iteration history can be seen in Figure 7. For the method advocated in this manuscript, 99.7% of the time steps required only a single CFD evaluation. Thus, we can state that our bidirectional coupled approach, with boundary conditions governed by parenchymal dynamics is efficient.

In this multiscale model, the outlets are too numerous and the interfaces too varied to be presented exhaustively. Instead, we present the results for two at the 8th and 10th generations, with diameters of $1.50 \times 10^{-3}m$ and $1.29 \times 10^{-3}m$ respectively (Figure 8). In Figure 9 we show excellent matching of flows at the two representative outlets. In fact, this accuracy of flow match occurred at all outlets in the human oral geometry and in the previous Weibel geometry with choice of tolerance TOL=1.e-2 which was used in Algorithm 3. In the human oral coupled simulation, the system required more cycles to settle into a stable pattern than in the Weibel case. This is expected due to the interplay of a much greater number of varied interface conditions. Also note that though both interfaces are at the 8th generation, their dynamics are subtly different, illustrating that geometry and ventilation heterogeneity influence site specific flow. Representative 3D results can be seen in Figure 10. In Figure 11 we show the spectrum of the Jacobian of the unpreconditioned residual (14) as it evolves in time. The 117 eigenvalues have negligible imaginary part and mostly have real parts between 0.5 and 3 as they shift over time during a breathing cycle.

4.3. Uncoupled Imaging-Based Human Multiscale Model

To illustrate the efficiency and physiological superiority of the proposed method with respect to an uncoupled or so-called 'uni-directional' method we present the results of multiple uni-directional sets of ODEs describing the distal lung to the imaging-based human lung geometry. Zero-pressure boundary conditions were applied to identical subtrees defined in the coupled case. The flows computed from these multiple sets of ODEs were then applied as boundary conditions to the CFD outlets. The communication was one-way. Because in this case the coupling is not bi-directional, mechanical equilibrium was not enforced on the interface — flows match but pressures do not match. The wall clock time for this uncoupled case (Table 2) was 21.8 hours. The wall clock time for the coupled case with the NACCEL accelerated Jacobian evaluations was 21.7 hours. Thus, the coupled case with physiologically meaningful boundary conditions was actually slightly faster. Moreover, in Figure 12, we illustrate that the predictions of these two models are quite different. These differences explain how the uncoupled simulation could actually take longer than the coupled simulation: Slightly faster fluid velocities in the uncoupled simulation lead to a slightly smaller timestep (as seen in Table 2), since timestep is limited to a fixed

multiple of the Courant number over all elements. The decrease in timestep represented a larger computational cost than the small additional cost of coupling in the coupled simulation.

5. Discussion and Conclusion

Simulations of particle deposition, smoke/soluble gas uptake, drug delivery – to name a few applications – in the respiratory system are areas of active research (e.g. [13, 44, 18]). These models primarily depend on airflow characteristics predicted by CFD models. Therefore, it is of utmost importance for the CFD models to accurately predict regional ventilation and flow characteristics. However, the nature of the lung requires that outlet boundary conditions reflect distal as well as proximal mechanics. Furthermore, particle deposition and gas uptake often occur at a scale below that of the 3D CFD geometry. While some data are available to inform such boundary conditions (e.g. [23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33]), they typically inform the microscale and thus must be associated with the outlets of the macroscale model via the interposition of a lower-dimensional model, requiring mechanical equilibrium at the interface between scales.

To address this challenge, we have presented a novel numerical framework for the iterative computation of multiscale boundary conditions. The approach is an extension of the classical Modified Newton's Method with nonlinear Krylov accelerator NACCEL which was developed by Carlson and Miller. Our extensions include retention of subspace information over multiple timesteps, as well as a special partial NACCEL correction at the end of a timestep that allows for corrections to be accepted with verified low residual. With these extensions, a single residual evaluation was required per timestep on average for our application, implying that the method incurred zero additional computational cost compared to uncoupled or unidirectionally coupled simulations. We expect these enhancements to be generally applicable to other multiscale coupling applications where timestepping occurs.

In addition we have developed a “pressure-drop” residual which allows for stable coupling of flows between a 3D incompressible CFD application and another (lower-dimensional) fluid system. We expect this residual to also be useful for coupling non-respiratory incompressible fluid applications, such as multiscale simulations involving blood flow.

Applied to solving the interface pressures at boundary outlets to a 3D CFD simulation of pulmonary flow, our solver has near optimal performance, requiring in most cases a single CFD call per time step. This performance is comparable to the current state of the art monolithic systems that require a dedicated solver, assuming the cheaper ODE systems incur near zero additional computational overhead.

Our nonlinear Krylov accelerator was implemented as a C++ boundary class in *OpenFOAM*[®], and can be run in parallel on shared or distributed memory systems. The boundary class is generic, so it could be easily adapted to other open source codes and also commercial codes with access to outer iterations.

To validate our method, in Section 4.1 we compared the predictions of a multi-scale model on an idealized geometry to an ODE-only solution of the same geometry. For that model, 99.96% of the time steps required a single CFD evaluation and accorded well with the ODE only model, despite the limitations of the latter. Specifically, the ODE system was limited to flows with low Reynolds and Womersley numbers; moreover, unlike the CFD model, the ODEs did not capture losses at the bifurcations. In Section 4.2 we coupled 117 sets of ODEs describing the distal lung, corresponding to 117 independent asymmetric outlets, to an imaging-based human lung geometry, driven again at a physiologic flow rate by prescribed spatially-heterogeneous terminal, or ‘alveolar’, flows. In this model, 99.71% of the time

steps required a only single CFD evaluation. Finally in Section 4.3 we showed that unidirectionally coupling leads to a significant mechanical mismatch at the interfaces and does not actually save CPU time.

For both models presented, only a simple residual was communicated between scales. Otherwise the CFD model and the lower-dimensional model were totally separate. This implies that the method may be treated as a black box by the practitioner, who only need design an appropriate residual. It also implies absolute freedom in terms of the selection of the lower-dimensional model to be coupled at the interface, overcoming a limitation of monolithic methods. Thus, we conclude that this new multiscale bidirectional coupling framework is efficient, accurate and robust. This allows us to begin combining spatial (CFD/PDE) models of the pulmonary airways with non-spatial or lower-dimensional models with little overhead and with great flexibility.

5.1. Limitations and Future work

The lower-dimensional model presented in this study was simplistic and not parametrized to the subject from whom the geometry was derived. This is appropriate, as the focus of the paper is on the numerical method. However, it is clear that to be truly predictive, coupled lower-dimensional models may have to have greater mechanical sophistication. As importantly, the parameters and boundary conditions for these future models must be derived from subject-specific imaging data. Our research into the definition and characterization of such imaging-based models is ongoing. Even so, the passive tissue mechanics of the distal lung is but one part of the process of respiration. In addition, the composition of inspired gases, smooth muscle tone, tissue and blood metabolism, the nervous system, and cardiovascular mechanics all play a part. The relative importance of these features of respiration is a function of the biomedical question one wishes to investigate. We emphasize that we have presented a method that is not only efficient and robust, but that also affords the researcher the flexibility to explore the impact and importance of these features in health and disease.

Acknowledgments

We would like to thank Neil Carlson for access to the original NACCEL FORTRAN subroutine. We would like to thank Professor C. Keith Miller for the idea of applying a partial NACCEL correction at the end of a timestep. We also gratefully acknowledge Drs. Robb Glenny and Sudhaker Pipavath, UW for the human CT images. This work was financially supported by National Institutes of Health (NIH) Bioengineering Research Partnership Grant R01-HL073598 (Richard A. Corley, PI).

References

1. Carlson N, Miller K. Design and application of a gradient-weighted moving finite element code I: in one dimension. *SIAM J Sci Comput.* 1998; 19(3):728–765.
2. Miller K. Nonlinear Krylov and moving nodes in the method of lines. *Journal of Computational and Applied Mathematics.* 2005; 183(2):275–287.
3. Scott MH, Fenves GL. Krylov subspace accelerated Newton algorithm: Application to dynamic progressive collapse simulation of frames. *J Structural Eng.* 2010; 136:473–480.
4. Lebowitz MD. Populations at risk: addressing health effects due to complex mixtures with a focus on respiratory effects. *Environ Health Perspect.* 1991; 95:35–38. [PubMed: 1821375]
5. Ortega AN, Calderon JG. Pediatric asthma among minority populations. *pediatric asthma among minority populations. Current Opinion in Pediatrics.* 2000; 12(6):579–583. [PubMed: 11106279]
6. Schwartz J. Air pollution and hospital admissions for the elderly in birmingham, alabama. *Am J Epidemiol.* 1994; 139(6):589–598. [PubMed: 8172170]
7. Amin S, Majumdar A, Frey U, Suki B. Modeling the dynamics of airway constriction: effects of agonist transport and binding. *J Appl Physiol.* 2010; 109:553–563. [PubMed: 20507971]

8. Barbini P, Brighenti C, Enini G, Gnudi G. A dynamic morphometric model of the normal lung for studying expiratory flow limitation in mechanical ventilation. *Annals of Biomedical Engineering*. 2005; 33(4):518–530. [PubMed: 15909658]
9. Ginzburg I, Elad D. Dynamic model of the bronchial tree. *J Biomed Eng*. 1993; 15(4):283–288. [PubMed: 8361153]
10. Lutchen KR, Yang K, Kaczka D, Suki B. Optimal ventilation wave-forms for estimating low-frequency respiratory impedance. *J Appl Physiol*. 1993; 75:478–488. [PubMed: 8376299]
11. Renotte C, Remy M, Saucez P. Dynamic model of airway pressure drop. *Med Biol Eng Comput*. 1998; 36:101–106. [PubMed: 9614756]
12. Kabilan S, Lin CL, Hoffman EA. Characteristics of airflow in a ct-based ovine lung: a numerical study. *J Appl Physiol*. 2007; 102(4):1469–82. doi:\bibinfo{doi}{10.1152/japplphysiol.01219.2005}. [PubMed: 17110504]
13. Kleinstreuer C, Zhang Z, Li Z. Modeling airflow and particle transport/deposition in pulmonary airways. *Respir Physiol Neurobiol*. 2008; 163(1-3):128–138. doi:\bibinfo{doi}{10.1016/j.resp.2008.07.002}. [PubMed: 18674643]
14. Tawhai MH, Hunter P, Tschirren J, Reinhardt J, McLennan G, Hoffman EA. Ct-based geometry analysis and finite element models of the human and ovine bronchial tree. *J Appl Physiol*. 2004; 97(6):2310–21. doi:\bibinfo{doi}{10.1152/japplphysiol.00520.2004}. [PubMed: 15322064]
15. Ma B, Lutchen KR. An anatomically based hybrid computational model of the human lung and its application to low frequency oscillatory mechanics. *Ann Biomed Eng*. 2006; 34(11):1691–704. doi:\bibinfo{doi}{10.1007/s10439-006-9184-7}. [PubMed: 17019619]
16. Lin CL, Tawhai M, McLennan G, Hoffman E. Computational fluid dynamics. *IEEE Eng Med Biol Mag*. 2009; 28(3):25–33. [PubMed: 19457731]
17. Wiechert, L.; Comerford, A.; Rausch, S.; Wall, W. *Fundamental Medical and Engineering Investigations on Protective Artificial Respiration*. Vol. 116. Springer; Berlin / Heidelberg: 2011. Advanced multi-scale modelling of the respiratory system; p. 1-32.
18. Vignon-Clementel IE, Figueroa CA, Jansen KE, Taylor CA. Outflow boundary conditions for three-dimensional finite element modeling of blood flow and pressure in arteries. *Comput Methods Appl Mech Engrg*. 2006; 195:3776–3796.
19. Vignon-Clementel I, Figueroa C, Jansen K, Taylor C. Outflow boundary conditions for 3d simulations of non-periodic blood flow and pressure fields in deformable arteries. *Comput Methods Biomech Biomed Engin*. 2010; 4(1)
20. Saad Y, Schultz MH. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J Sci Stat Comput*. 1986; 7(3):856–869.
21. Carlson, NN. Moving finite elements software. Nacel subroutine part of bdf2.f. 2008. URL <http://mfeproject.sourceforge.net/>
22. Golub, GH.; Van Loan, CF. *Matrix Computations*. 3. The John Hopkins University Press; Baltimore MD: 1996.
23. Corona TM, Aumann M. Ventilator waveform interpretation in mechanically ventilated small animals. *J Vet Emerg Crit Care (San Antonio)*. 2011; 21(5):496–514. doi:\bibinfo{doi}{10.1111/j.1476-4431.2011.00673.x}. [PubMed: 22316197]
24. Harris RS. Pressure-volume curves of the respiratory system. *Respiratory Care*. 2005; 50(1):78–99. [PubMed: 15636647]
25. Lu Q, Rouby JJ. Measurement of pressure-volume curves in patients on mechanical ventilation: methods and significance. *Crit Care*. 2000; 4(2):91–100. [PubMed: 11094498]
26. Mead J, Turner J, Macklem P, Little J. Significance of the relationship between lung recoil and maximum expiratory flow. *J Appl Physiol*. 1967; 22(1):95–108. [PubMed: 6017658]
27. Opazo S, Du T, Wang N, Martin J. Methacholine-induced bronchoconstriction and airway smooth muscle in the guinea pig. *J Appl Physiol*. 1996; 80(2):437–444. [PubMed: 8929581]
28. Perchiazzi G, Rylander C, Vena A, Derosa S, Polieri D, Fiore T, Giuliani R, Hedenstierna G. Lung regional stress and strain as a function of posture and ventilatory mode. *J Appl Physiol*. 2011; 110(5):1374–83. doi:\bibinfo{doi}{10.1152/japplphysiol.00439.2010}. [PubMed: 21393463]

29. Sinclair SE, Molthen RC, Haworth ST, Dawson CA, Waters CM. Airway strain during mechanical ventilation in an intact animal model. *Am J Respir Crit Care Med*. 2007; 176(8):786–94. doi: \bibinfo{doi}{10.1164/rccm.200701-088OC}. [PubMed: 17626911]
30. Fouras A, Allison BJ, Kitchen MJ, Dubsky S, Nguyen J, Hourigan K, Siu KKW, Lewis RA, Wallace MJ, Hooper SB. Altered lung motion is a sensitive indicator of regional lung disease. *Ann Biomed Eng*. 2012; 40(5):1160–9. doi:\bibinfo{doi}{10.1007/s10439-011-0493-0}. [PubMed: 22189492]
31. Yin, Y. Ph D thesis; Univeristy of Iowa. 2011. Mdct-based dynamic, subject-specific lung models via image registraion for cfd-based interrogation of regional lung function.
32. Choy S, Wheatley A, McCormack DG, Parraga G. Hyperpolarized (3)he magnetic resonance imaging-derived pulmonary pressure-volume curves. *J Appl Physiol*. 2010; 109(2):574–85. doi: \bibinfo{doi}{10.1152/japplphysiol.01085.2009}. [PubMed: 20538846]
33. Castillo E, Castillo R, Martinez J, Shenoy M, Guerrero T. Four-dimensional deformable image registration using trajectory modeling. *Phys Med Biol*. 2010; 55(1):305–27. doi:\bibinfo{doi}{10.1088/0031-9155/55/1/018}. [PubMed: 20009196]
34. Weibel, ER. *Morphometry of the Human Lung*. New York: Academic; 1963.
35. Kuprat A, Einstein D. An anisotropic scale-invariant unstructured mesh generator suitable for volumetric imaging data. *J Computat Phys*. 200810.1016/j.jcp.2008.09.030
36. Barbini P, Cevenini G, Avanzolini G. Nonlinear mechanisms determining expiratory flow limitation in mechanical ventilation: A model-based interpretation. *Annals of Biomedical Engineering*. 2003; 31(8):908–916. [PubMed: 12918905]
37. Press, W.; Teukolsky, S.; Vetterling, W.; Flannery, B. *Numerical Recipes*. Third Edition. Cambridge; 2007.
38. Pedley TJ, Schroter RC, Sudlow MF. Energy losses and pressure drop in models of human airways. *Respir Physiol*. 1970; 9(3):371–86. [PubMed: 5425200]
39. Carson JP, Einstein DR, Minard KR, Fanucchi MV, Wallis CD, Corley RA. High resolution lung airway cast segmentation with proper topology suitable for computational fluid dynamic simulations. *Comput Med Imaging Graph*. 2010; 34(7):572–578. [PubMed: 20382502]
40. Corley RA, Minard KR, Kabilan S, Einstein DR, Kuprat AP, Harkema JR, Kimbell JS, Gargas ML, Kinzell JH. Magnetic resonance imaging and computational fluid dynamics (cfd) simulations of rabbit nasal airflows for the development of hybrid cfd/pbpc models. *Inhal Toxicol*. 2009; 21(6):512–8. doi:\bibinfo{doi}{10.1080/08958370802598005}. [PubMed: 19519151]
41. Lorensen W, Cline H. Marching cubes: a high resolution 3d surface construction algorithm. *Comput Graph*. 1987; 21:163–169.
42. Jiao X, Einstein D, Dyedov V, Carson J. Automatic identification and truncation of boundary outlets in complex imaging-derived biomedical geometries. *Med Biol Eng Comput*. 2009; 47(9): 989–999. [PubMed: 19526263]
43. Jiao X, Einstein D, Dyedov V. Local orthogonal cutting method for computing medial curves and its biomedical applications. *SIAM Journal on Scientific Computing*. 2010; 32:947–969. [PubMed: 20628546]
44. Tian G, Longest PW. Development of a cfd boundary condition to model transient vapor absorption in the respiratory airways. *J Biomech Eng*. 2010; 132(5):051003. doi:\bibinfo{doi}{10.1115/1.4001045}. [PubMed: 20459204]

Figure 1.

Shows the instability due to inertial effect when using the difference in the CFD and the ODE flow as the residual. Spurious pressure oscillations were noticed at the interface when using this residual.

$$\Delta P_i^{CFD} = R_i Q_i^{CFD} + L_i \dot{Q}_i^{CFD} \quad \Delta P_i^{ODE} = R_i Q_i^{ODE} + L_i \dot{Q}_i^{ODE}$$

$$R_i = \frac{128\mu l_i}{\pi D_i^4} \quad L_i = \frac{4\rho l_i}{\pi D_i^2}$$

Figure 2.

The new residual in which the pressure drop over a length of one diameter on the CFD and the ODE side were equated. μ is dynamic viscosity of air, ρ is the density, D_i is the diameter of the airway. The length l_i was assumed to be equal to D_i .

Figure 3. Schematic of the 4-generation idealized geometry. The system of equations in (18)-(21) are represented graphically at the CFD outlets. Boundary conditions for the validation case consisted of an applied time-derivative of the intrapleural pressure and zero pressure at the trachea.

Figure 4.

Results for the idealized dynamic respiration case. Since the model is symmetric, only the results for a single outlet/ODE pair are presented. Due to the relatively high Reynolds and Womersley numbers, there is no expectation that the multiscale results should agree with the ODE-only model. Nevertheless, ODE-only predictions (magenta markers) are included for reference. **A)** Boundary conditions consisted of a physiological time-varying intrapleural pressure derivative, and zero total pressure at the trachea. **B)** Four cycles of pressures at both the outlet (multiscale interface) and the terminal acinus of System (18). **C)** Interface (generation 4) flows as a function of time for both CFD-ODE and ODE only were closely matched. **D)** Note the dynamic behavior of the system. At $t = 0$, the system begins with zero flow in the terminal unit. After two cycles the system is largely stable. Showing a hysteresis loop (pressure vs. flow) in the acinus. This behavior is typical of compliant systems. Despite the high Reynolds and Womersley numbers the multiscale model and the pure ODE-only model accord very well.

Figure 5. NACCEL performance for the Weibel multiscale respiration model. For the MOD NEWT + NACCEL method, 99.96% of the time steps required a single CFD evaluation (black), meaning that the approach was near optimal. In contrast, for the MOD NEWT method (Modified Newton with diagonal Jacobian preconditioner and NACCEL accelerator not used) (gray), 66.96% of the time steps required a single CFD evaluation.

Figure 6.

A) Distribution of the 117 outlets in the imaging-based human model by generation and diameter-equivalent generation numbers. Ultimately, the morphometry and mechanics of the lower-dimensional models must come from imaging data. However, for this exercise we have used diameter-equivalent generation numbers to determine outlet subtrees based on published data [9].

Figure 7.

NACCEL performance for the human multiscale respiration model. For the *MOD NEWT* + *NACCEL* method, 99.7% of the time steps required a single CFD evaluation (black), meaning that the approach was near optimal. In contrast, for the *MOD NEWT* method, (gray), 76.2% of the time steps required a single CFD evaluation. Note that for the *MOD NEWT* + *NACCEL* method, a single time step required more than two iterations (seven iterations were required). In contrast, for the *MOD NEWT* method the maximum number of iterations was 12 (not shown).

Figure 8.

Results for the human dynamic respiration case. Data from a generation 10 outlet (#1) with a diameter of and from a generation 8 outlet (#117) are presented. **A)** Boundary conditions consisted of a physiological time-varying intrapleural pressure derivative, and zero total pressure at the trachea. **B)** Intrapleural pressure versus volume change distal to the multiscale interface for each outlet. Note the system requires one full cycle to reach a stable orbit. The distal volume is the volume contained in the unique subset of System (18) for each outlet. **C)** Interface and terminal flows as a function of time for both outlets. The difference in flows between the interface and acinar unit visible for outlet #117 (magenta) is due to the compliance in the system. **D)** Interface pressure versus interface flow for both outlets. At $t = 0$, the system begins with zero flow. After two cycles the system is largely stable, showing a hysteresis loop (pressure vs. flow) in the acinus. This behavior is typical of the lung. Note that a coupled imaging-based model requires a greater number of breathing cycles to stabilize than the idealized geometry.

Figure 9.

Interface flow plotted for two of the 117 outlets of the human multiscale model. Both CFD and ODE results on either side of the interface are shown and are seen to match very well. Instantaneously, flow rates could differ by up to 1%, but time integrated flow rate (i.e., volume of fluid passing by interface) agreed to within 0.1% at all times.

Figure 10.

CT-derived geometry of the human respiratory system with 117 terminal outlets. Unique subtrees, representing the distal lung were applied to each of the 117 outlets based on Equations (18)-(21). Boundary conditions were atmospheric pressure at the mouth and time-varying intrapleural pressure derivative in the lower-dimensional models. Geometry (**A**) and streamlines at peak inspiration (**B**).

Figure 11.

Real part of spectrum of Jacobian $\mathbf{r}'$ of the unpreconditioned residual in Equation 14 over a breathing cycle. Imaginary part was negligible.

Figure 12.

Histogram of percent difference in flow at each of the 117 different outlets, binned by percent total number of outlets. Results from the uncoupled or ‘uni-directional’ case differ by as much as 35% with respect to the correctly coupled case, wherein mechanical equilibrium is established. The inset show the time-dependent flow through the trachea for both cases.

Performance statistics for the multiscale coupling of Equations (18)-(21) to an idealized 3D Weibel geometry for the two methods (WCT stands for wall-clock time).

Table 1

Case	Processors	WCT (h)	Residual Evals	Jacobian Evals	Avg Time Step (sec)
MOD NEWT	48	9.8 hours	191504	10	3.19E-05
MOD NEWT + NACCEL	48	6.4 hours	125689	10	3.18E-05

Table 2

Performance statistics for the multiscale coupling of Equation 18 to an imaging based human geometry for the the methods investigated, *MOD NEWT* + *NACCEL*, *MOD NEWT*, and for the uncoupled case.

Case	Processors	WCT (h)	Residual Evals	Jacobian Evals	Avg δ (sec)
MOD NEWT	72	28.4 hours	243679	6	2.14×10^{-5}
MOD NEWT + NACCEL	72	21.7 hours	187702	4	2.14×10^{-5}
UNCOUPLD	72	21.8 hours	N/A	N/A	2.07×10^{-5}

Note the *MOD NEWT* + *NACCEL* method required fewer evaluations of the Jacobian than the *MOD NEWT* method. WCT stands for wall-clock time.