OpenFOAM as a Model Sharing Paradigm for Macro-Microscale Biomedical Simulation

Robert F. Kunz
Head, Computational Mechanics Division, Applied Research Laboratory
Associate Professor of Aerospace Engineering
Eric Paterson
Division Scientist, Computational Mechanics Division, Applied Research Laboratory
Associate Professor of Mechanical Engineering
Brent Craven
Research Associate, Computational Mechanics Division, Applied Research Laboratory

The Pennsylvania State University

IMAG Web Seminar
5 February, 2010

Contents

- **Model sharing at the macroscale: Requirements and Challenges**
- **OpenFOAM**
  - **Architecture**
  - **How it addresses macroscale modeling challenges**
  - **Biological and biomedical examples**

Model sharing at the macroscale

- ***Lots of progress*** **in model sharing in the micro-to-molecular scale biomedical research community → numerous successful model sharing frameworks:**
  - **SBML, CellML (and offspring)**
  - **CompuCell3D**
  - **CellX**
  - **FLAME**
  - **TRND**
- **Some macroscale approaches for animal/human motion, clinical data analysis, … outside the scope of what is discussed here**

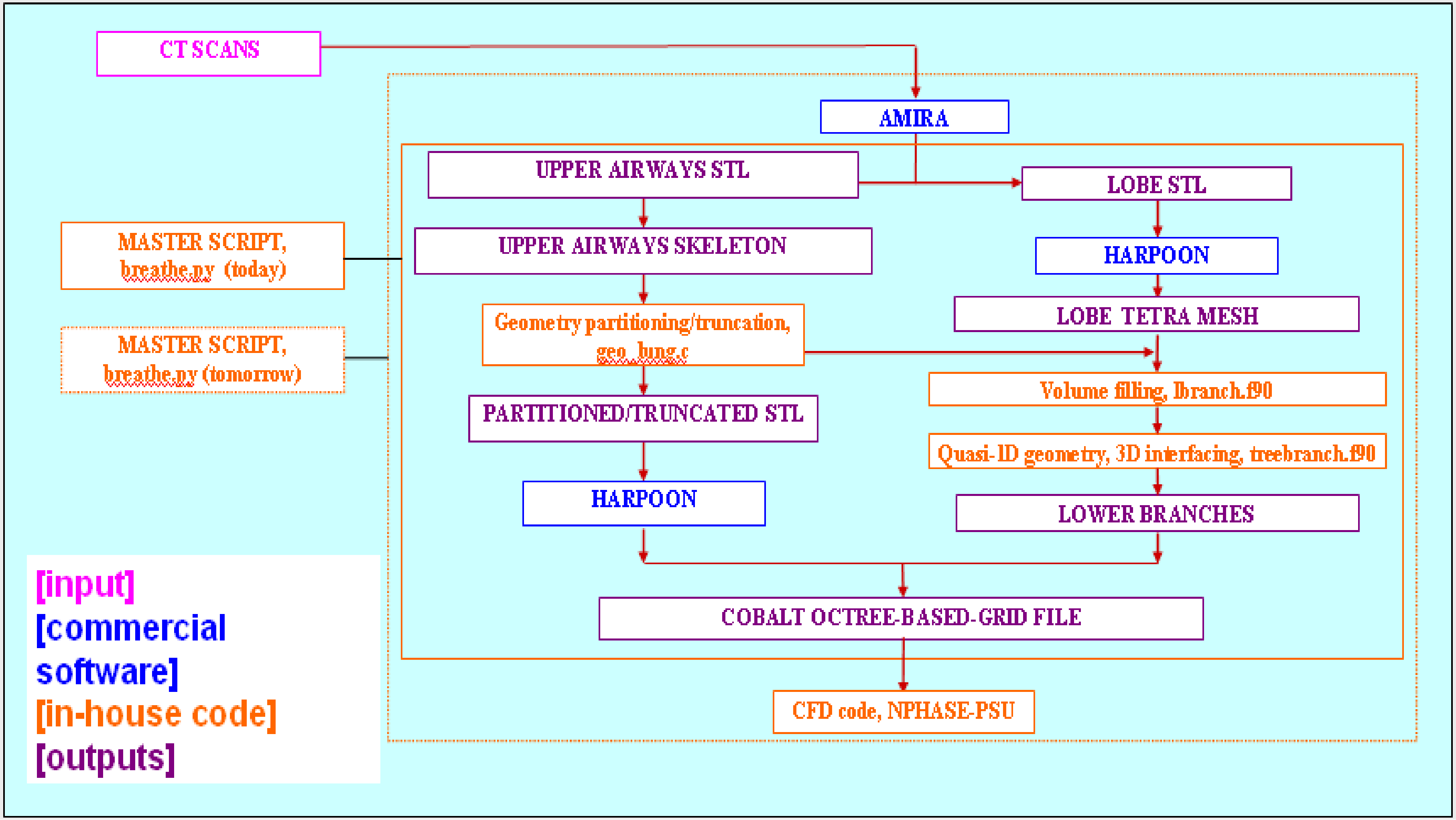
Model sharing at the macroscale

- Here we are concerned with space-time simulation of macroscale processes, i.e., those involving:
  - Medical image processing
  - Geometry processing
  - Computational Fluid Dynamics (CFD)
  - Computational Structural Mechanics (CSM)
  - Attendant meshing technology (motion, deformation, adaption)
  - Interfaces between these disciplines
  - Interfaces to microscales
- In this venue, we argue that ***comparatively little progress*** has been made in model sharing

Model sharing at the macroscale

- **"Model sharing" carries a significantly different set of implications at the micro/molecular scales than in 3D macro-scale bio-physics modeling**
- **Model sharing in this context can involve software components with *widely varying*:**
  - **Source code availability**
  - **Licensing and attendant costs (esp. for parallel)**
  - **I/O interfaces/GUIs**
  - **Data structures**
  - **Interfaces between flow-structure-mesh**
  - **Modularity (i.e., plug and play components)**
  - **Operating system requirements**
  - **Mesh topologies**
  - **Parallel execution frameworks (including domain decomposition)**
  - **Uncertain longevity**
  - **Required user training ➔ often significant (i.e., grid generation)**
  - **Other software engineering elements (e.g., scripting components)**
- **In short, they are hard to "share"**

How can I share this?



CT SCANS

AMIRA

UPPER AIRWAYS STL

UPPER AIRWAYS SKELETON

Geometry partitioning/truncation, geo_lung.c

PARTITIONED/TRUNCATED STL

HARPOON

LOBE STL

HARPOON

LOBE TETRA MESH

Volume filling, lbranch.f90

Quasi-1D geometry, 3D interfacing, treebranch.f90

LOWER BRANCHES

COBALT OCTREE-BASED-GRID FILE

CFD code, NPHASE-PSU

MASTER SCRIPT, breathe.py (today)

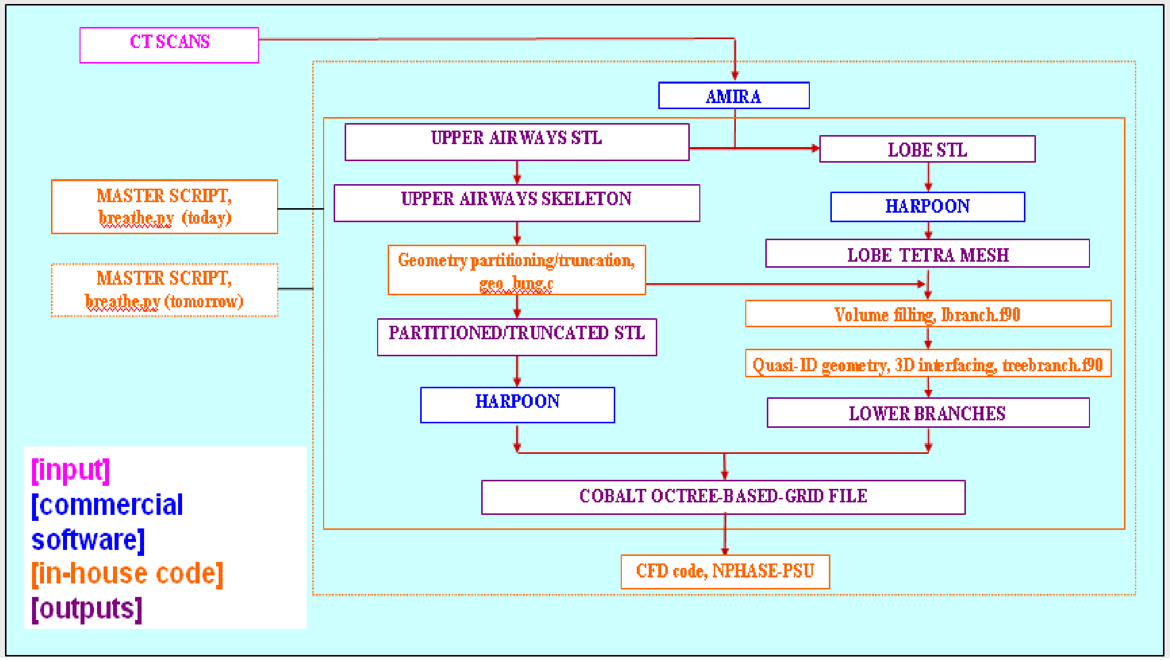
MASTER SCRIPT, breathe.py (tomorrow)

[input]
[commercial software]
[in-house code]
[outputs]

What can we share

- **In 2010, integration of the software components in complex multi-disciplinary computer modeling, is usually accomplished using scripts written in high-level languages such as PYTHON.**

  Such a master script (without its modeling components) could be usefully shared with others



CT SCANS

AMIRA

UPPER AIRWAYS STL

LOBE STL

MASTER SCRIPT, breathe.py (today)

UPPER AIRWAYS SKELETON

HARPOON

LOBE TETRA MESH

MASTER SCRIPT, breathe.py (tomorrow)

Geometry partitioning/truncation, geo_lung.c

Volume filling, lbranch.f90

PARTITIONED/TRUNCATED STL

Quasi-1D geometry, 3D interfacing, treebranch.f90

HARPOON

LOWER BRANCHES

COBALT OCTREE-BASED-GRID FILE

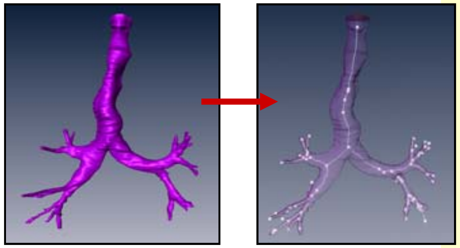
CFD code, NPHASE-PSU

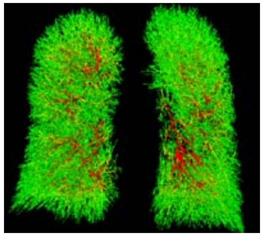
[input]
[commercial software]
[in-house code]
[outputs]

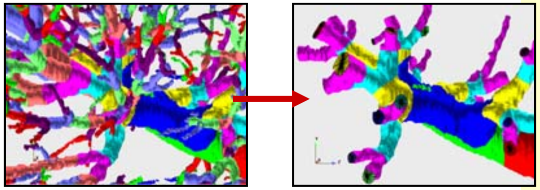
breathe.py

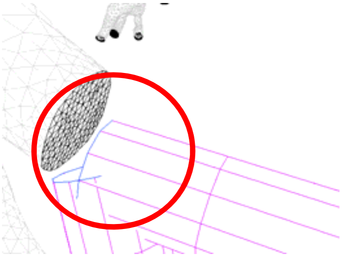
What can we share

- **Due to the complex geometries and multiple scales involved in biological systems, geometry and topology processing components are required, e.g.,**
  - **Volume filling**
  - **"thinning"**
  - **Truncation**
  - **Quasi-1D representations**
  - **Automatic attribute identification**
  - **.**
  - **.**
  - **.**









- **Often these components are small stand alone programs**

Such geometry and topology processing codes could be usefully shared with others

What is hard to share

- But for reasons discussed above the critical modeling components of CFD+CSM+meshing are hard to share.
- For example for me to "share" the respiration modeling toolkit developed under MSM would require:
  - breathe.py and several small C++ and Fortran geometry and topology processing codes (easily shared)
  - NPHASE-PSU – an open source general purpose unstructured CFD code (with limited CSM capability)
  - HARPOON – a commercial grid generation package
  - AMIRA – a commercial medical image processing tool
- So our approach was to design the toolkit such that these "hard to share" components could be swapped for other tools at the users' discretion.

What is hard to share

- **But even a very well software engineered scripting environment does not address *complexity* inherent in the analysis tools themselves. For example:**
  - **The CFD code has several respiration-simulation-specific components including:**
    - **Quasi-1D treatments for lower bronchioles**
    - **"Piston" boundary condition treatments**
    - **Data import of non-standard attributes including loss coefficients, generation number (and other topology), bronchiole length (and other geometric attributes), particle properties, mechanical properties of bronchioles**
    - **Application specific post-processing – various fluxes, pressure drops, deposition efficiency**
  - **For me to share these critical models requires that I either:**
    1) **Share the whole CFD code**
    2) **Resort to sharing only documentation of what/how these components are installed**

What is hard to share

1. **Sharing any CFD code has a subset of the issues brought forward on slide 5 (training/support, licensing, recompilation, parallel environment, attendant software [domain decomp, solver libraries, overset meshing], etc…)**

   Not a good option

2. **Sharing only documentation of what/how these components are installed requires each user to re-develop these software modules within the context of their chosen CFD platform. This • is contrary to model sharing philosophy, • requires reverification, and • may not even be possible.**

   Not a good option

What can’t we share

- The same types of issues come forward in the context of sharing CSM modeling.
- Also, very importantly, fluid-structure interaction approaches are hard to share. Specifically FSI schemes can be:
  - Loosely or strongly coupled
  - Involve single or multiple codes/modules
  - Have widely varying (and generally poor) parallel scaling
  - Have widely varying and application specific flow-structure interpolation schemes
  - Have widely varying and application specific mesh motion/adaption schemes and mesh topologies
- So again it is difficult or impossible to abstract the choices a model developer makes in the FSI context away from user-desired component codes.

Possible solutions

- **Commercial CFD-CSM-mesh motion/adaption**
  - **ANSYS multiphysics**
  - **Adina multiphysics**
  - **Other commercial solutions**
- ***In our view*, a commercial solution could only become an adequate model sharing vehicle if:**
  - **It was adopted as a community standard**
  - **Its cost was very low**
  - **It scaled well to many processors**
- ***In our view*, an open-source solution is more promising. One of these is OpenFOAM, which we discuss here. Others exist including Deal.II**

OpenFOAM

- **OpenFOAM (Open Field Operation and Manipulation) toolbox, http://www.opencfd.co.uk/openfoam/**
- **Open-source, open-development, freely extensible**
- **In our assessment, OpenFOAM has features that give it the potential for greatly expanding the ability of macro-scale biomedical modeling researchers to share their models directly.**
- **OpenFOAM is an object-oriented framework written in C++ , for customization and extension of numerical solvers for continuum mechanics problems including CFD, CSM, mesh motion/adaption and other disciplines.**

OpenFOAM

- **OpenFOAM permits the development of highly-customized solvers and utilities for numerical modeling and simulation of (among other things) coupled fluid-structural mechanics problems**
- **OpenFOAM approach: each *sub-model* (e.g., turbulence model, particle deposition model) or *high level component* (e.g., FSI solver, DNS solver, overset meshing) *can be developed and installed modularly alongside the standard and future OpenFOAM installations which are freely available.***

- **This is inherently amenable to model sharing since once users have adopted the richly capable baseline version of OpenFOAM (relevant here: CFD,CSM, mesh motion/adaption, arbitrary polyhedral unstructured, high order numerics, MPI parallel), they can easily integrate the sub-models or high level components of any other group in the community, and straightforwardly share their models with the community simply by downloading/posting the software from/to the OpenFOAM Wiki or one of numerous open forums (e.g., IMAG).**

OpenFOAM

- **This open, sharing-based, *free!* model development environment has led to OpenFOAM's explosive in growth recent years among macro-scale multi-physics simulation research communities (> 3,000 users worldwide.**
- **Beginning to build acceptance within the biomedical macro-scale physics modeling community where model sharing, dissemination and reuse could lead to more rapid research progress and efficient expenditures of sponsor support.**

OpenFOAM - Architecture

- **OpenFOAM architecture based upon Object Orientation and Generic Programming Techniques**
  - **Object-oriented approach handles complexity by splitting up the software into smaller and protected units, implemented and tested in isolation**
  - **This high-Level approach to programming critical for assuring interoperability of shared models**
  - **User developed code is outside of distribution source, libraries/applications ➔ shared to stand alongside current and future OpenFOAM distributions**
  - **Exploits standard powerful C++ contructs: classes, user defined typing, virtual functions and templates**
  - **You need to know/learn some C++!**

OpenFOAM – Baseline is richly capable

- **Parallelism:**
  - **Taken care of at a very low level: model developers don't have to worry about MPI details**
  - **Domain decomposition via multiple approaches (e.g., METIS)**
- **Multi-region modeling:**
  - **Interpolation/mapping between regions**
  - **Separate meshes for each region**
  - **Separate equations for each region; FSI, conjugate mass/heat transfer, etc.**
  - **Separate materials and properties for each region**
  - ***Ideal for macro-2-micro multi-scale modeling***

OpenFOAM – Baseline is richly capable

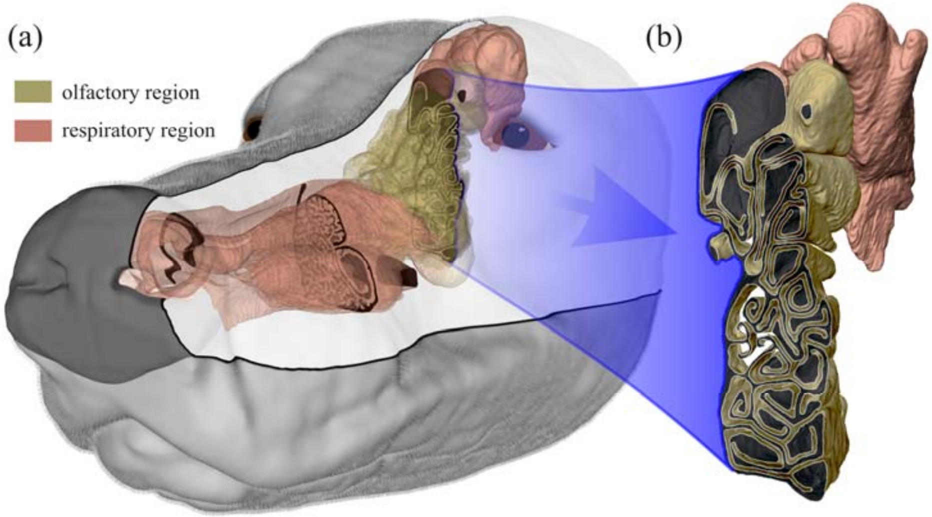
- Discretization for multi-region/multi-scale modeling
  - Finite volume method
  - Finite area method for thin layers on curved surfaces
  - Finite element method
  - Lagrangian particles/sprays/bubbles
- Arbitrary polyhedral support
  - Complex geometries
- Dynamic moving mesh
  - Prescribed and solution-dependent motions
  - Mesh motion: PDES, RBF, topology modification

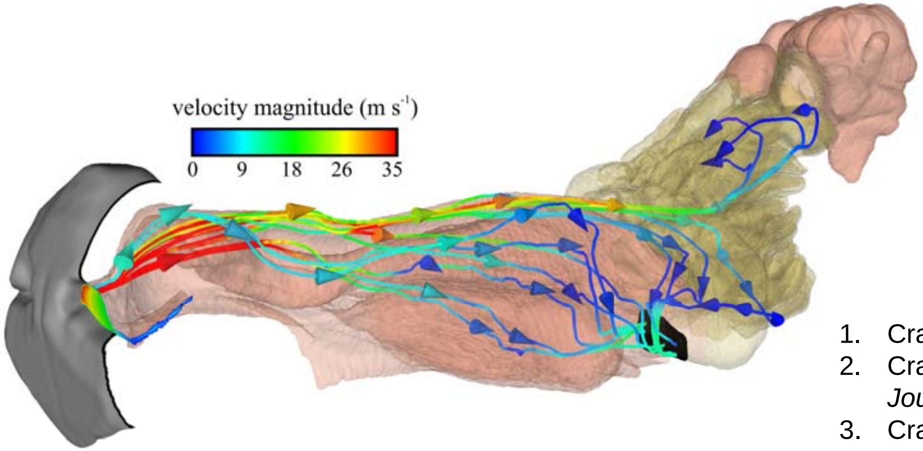
OpenFOAM – Baseline is richly capable

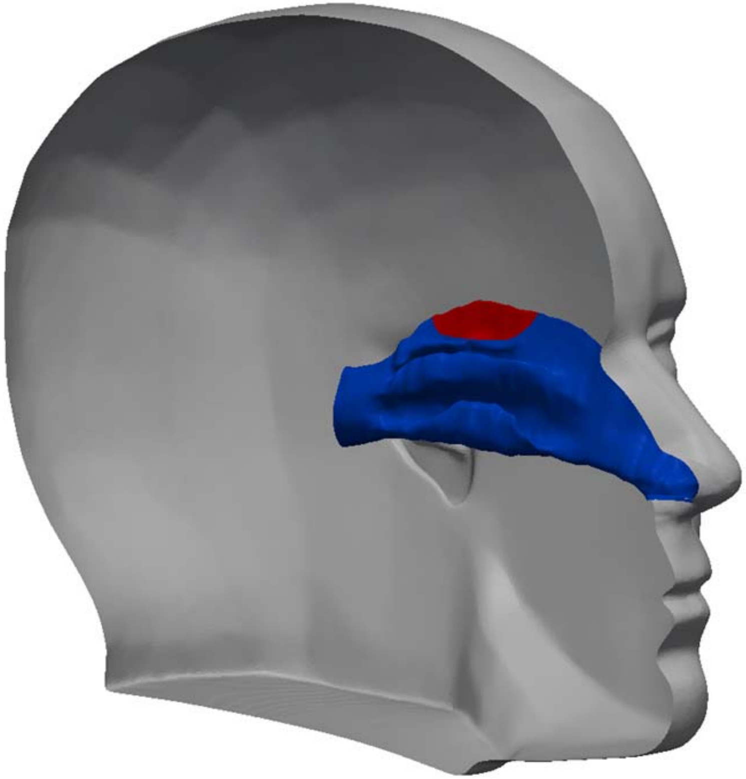
- Execution is much like any standard high-level scientific simulation package:
- Run-time selection of dynamically-loaded libraries
  - Finite-volume and other discretization schemes
  - Turbulence models: laminar, RANS, LES, DES
  - Dynamic meshing
  - Solution schemes for algebraic systems
  - Physical models: e.g., constitutive relations, equation of state, chemistry
  - Output choices
  - .
  - .
  - .

OpenFOAM - Biological and biomedical examples

Mammalian respiration and olfaction:



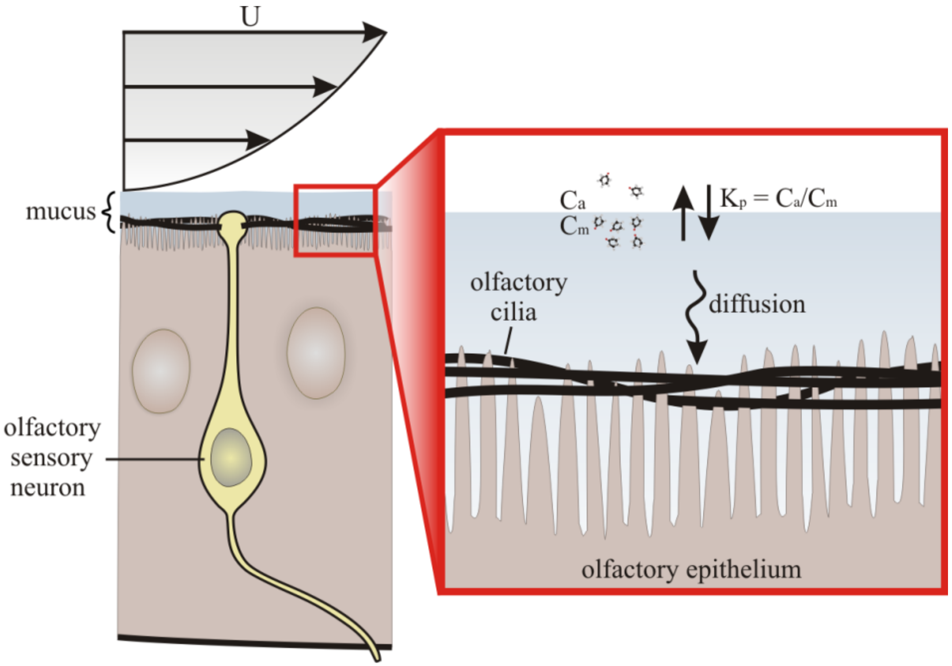


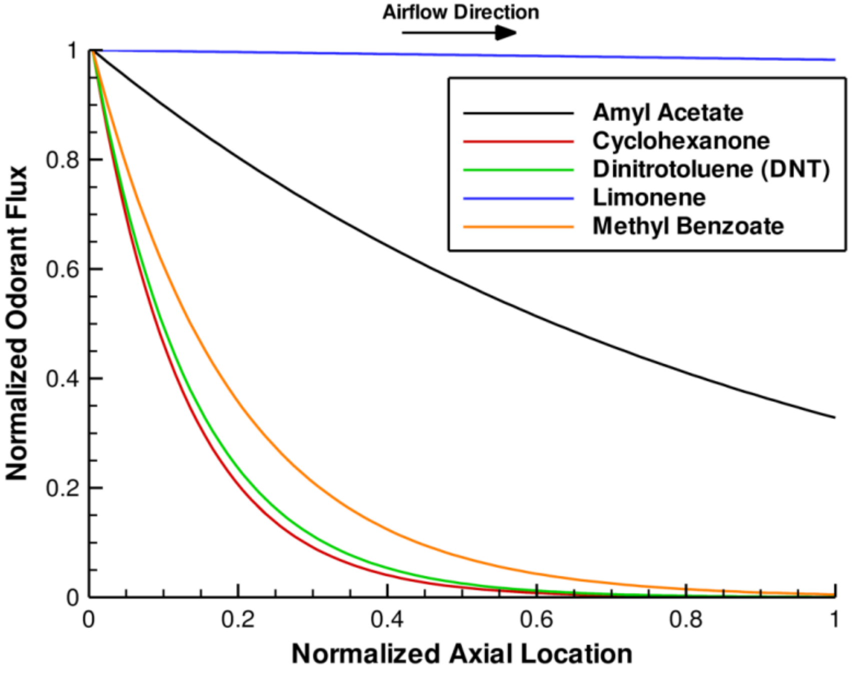


1. Craven, B. A., et al., 2007, *The Anatomical Record*, 290:1325-1340.
2. Craven, B. A., Paterson, E. G., Settles, G. S., and Lawson, M. J., 2009. *Journal of Biomechanical Engineering*, 131(091002):1-11.
3. Craven, B. A., Paterson, E. G., and Settles, G. S., 2009. *Journal of the Royal Society Interface* (published online before print Dec. 9, 2009).

OpenFOAM - Biological and biomedical examples

- **Multiphysics/multiregion modeling: vapor absorption/uptake (olfaction)**





1. Lawson, M. J., Craven, B. A., Paterson, E. G., and Settles, G. S., 2010, submitted to *Chemical Senses*.

OpenFOAM - Biological and biomedical examples

- **Fluid-structure interaction for heart pumps with polymeric deformable blades**

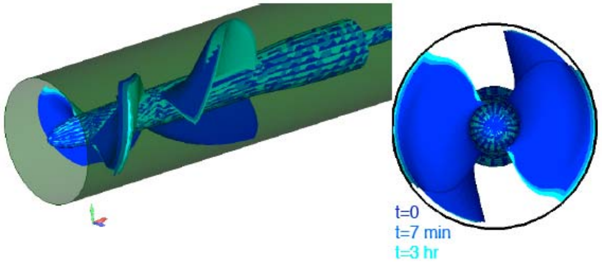


Figure 1: Notional impeller tip clearance dependency on time; obtained from a one-way fluid/solid coupling model with negative loads applied

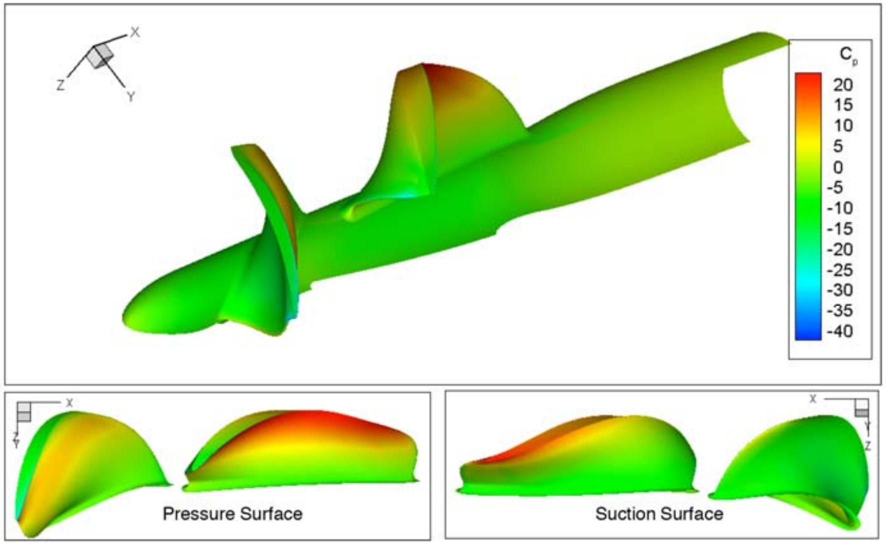


Figure 6: Impeller pressure contours from preliminary cyclic-symmetric CFD analysis

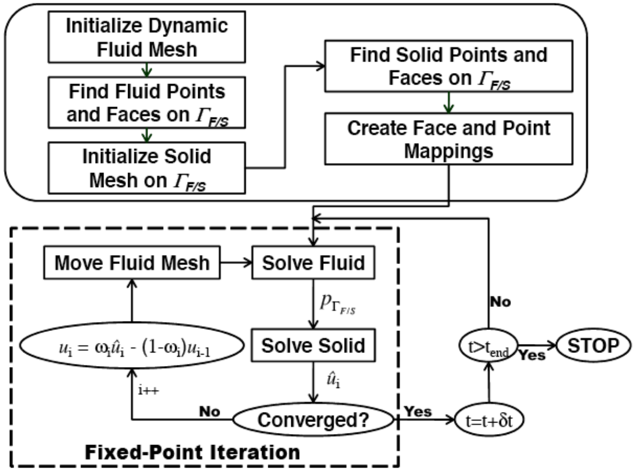


Figure 8: FSI solver implementation showing startup details and the fixed-point iteration of the partitioned approach

1. Campbell, Paterson, Reese, and Hambric, “Fluid–structure simulation of a viscoelastic hydrofoil subjected to quasi-steady flow, FSI 2009, Crete.
2. Campbell, PhD Thesis, Expected Dec. 2010.

OpenFOAM - Biological and biomedical examples

- **Fluid-structure interaction for heart pumps with polymeric deformable blades**

Validation Benchmark

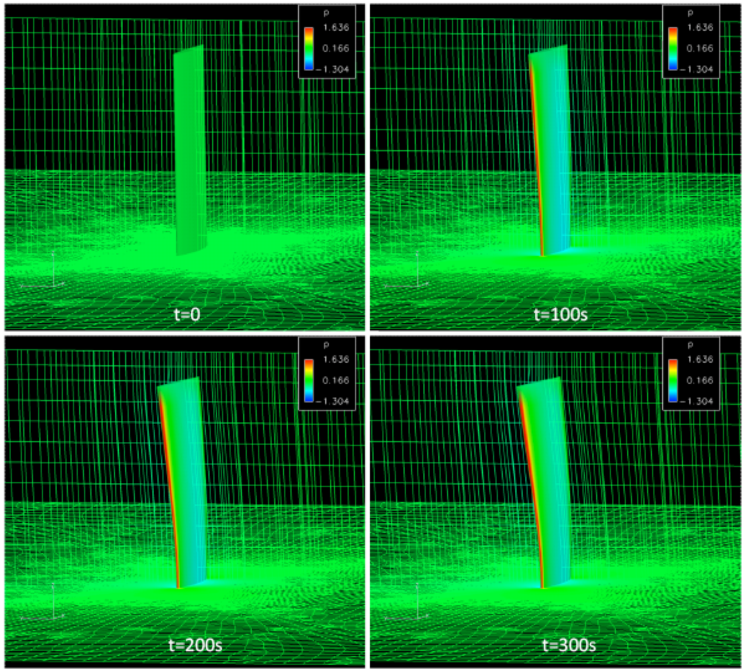


Figure 16: Mesh motion for single hydrofoil test case, front view; contours of pressure (kPa) shown on the foil surface and the cell edges; results shown for $t = 0$, 100, 200, and 300 s; images show a slice through the mesh near the mid-chord location in the y-z plane, flow is in the x-direction; the images show negligible cell deformation in the foil vicinity resulting from mesh motion with quadratic decay of the diffusion coefficient

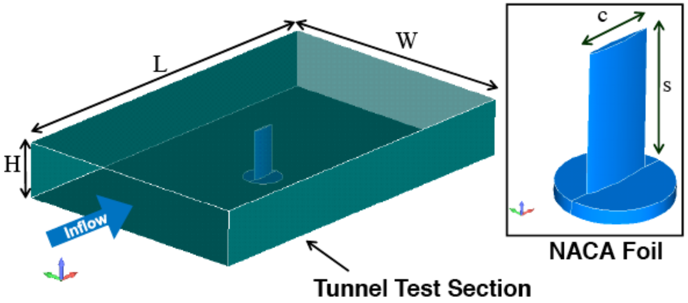
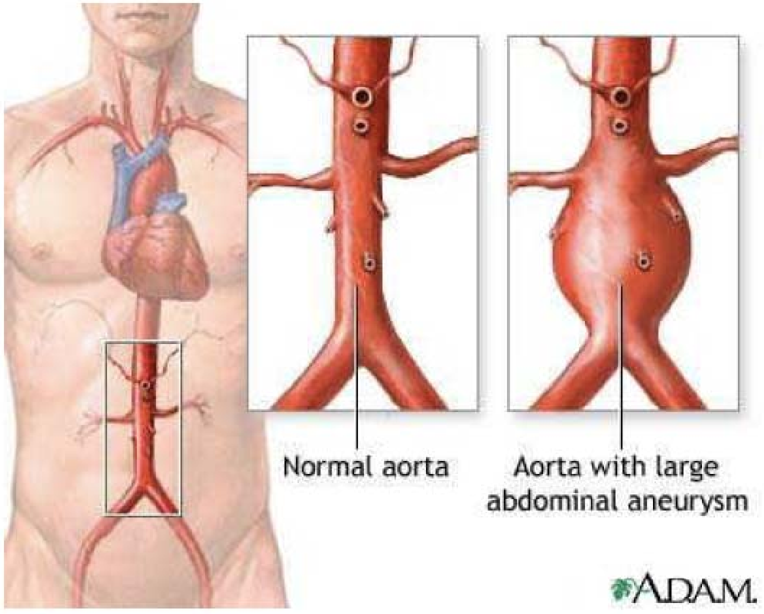


Figure 9: Flow over a cantilevered, modified NACA 66 hydrofoil with uniform inflow conditions situated in the rectangular test section of a 12-inch water tunnel; H is in the vertical direction

1. Campbell, Paterson, Reese, and Hambric, “Fluid–structure simulation of a viscoelastic hydrofoil subjected to quasi-steady flow, FSI 2009, Crete.
2. Campbell, PhD Thesis, Expected Dec. 2010.

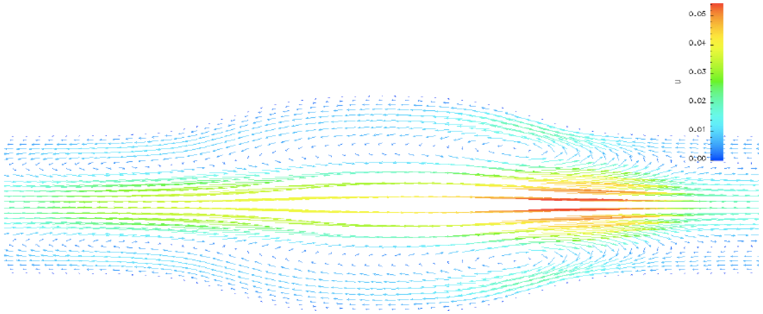
OpenFOAM - Biological and biomedical examples

- **FSI of Abdominal Aortic Aneurysms**

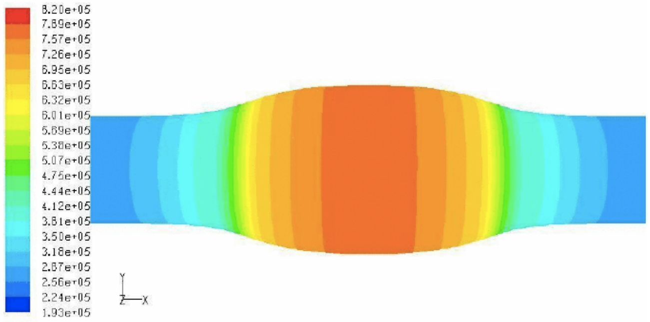


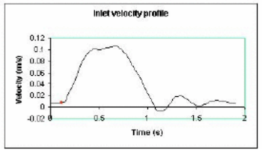
~90% mortality rate if rupture occurs

Fluid Flow



Stress – Model 2





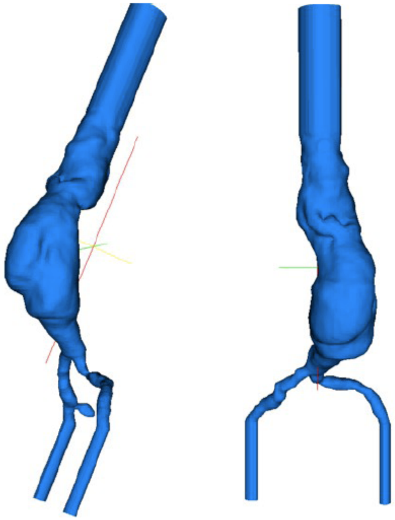
Kelly and O'Rourke, 2rd OpenFOAM Workshop, Zagreb, Croatia, 2007

OpenFOAM - Biological and biomedical examples

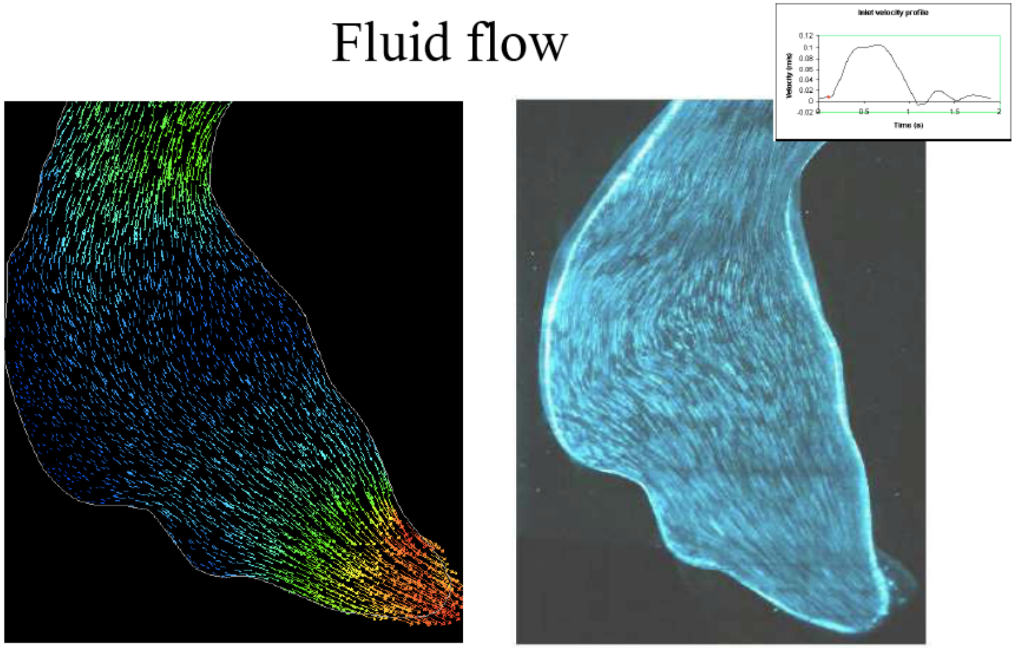
FSI of Abdominal Aortic Aneurysms

Patient based geometry

- Female patient
- 68 years old



Fluid flow



Flow visualisation results from "Haemodynamics of Abdominal Aortic Aneurysms: A Comparison between Idealised and Patient-Based Models", James McCullough, PhD thesis, UCD, 2006

Ivankovic, 2nd OpenFOAM Workshop, Zagreb, Croatia, 2007

OpenFOAM - Biological and biomedical examples

Cardiovascular fluid dynamics

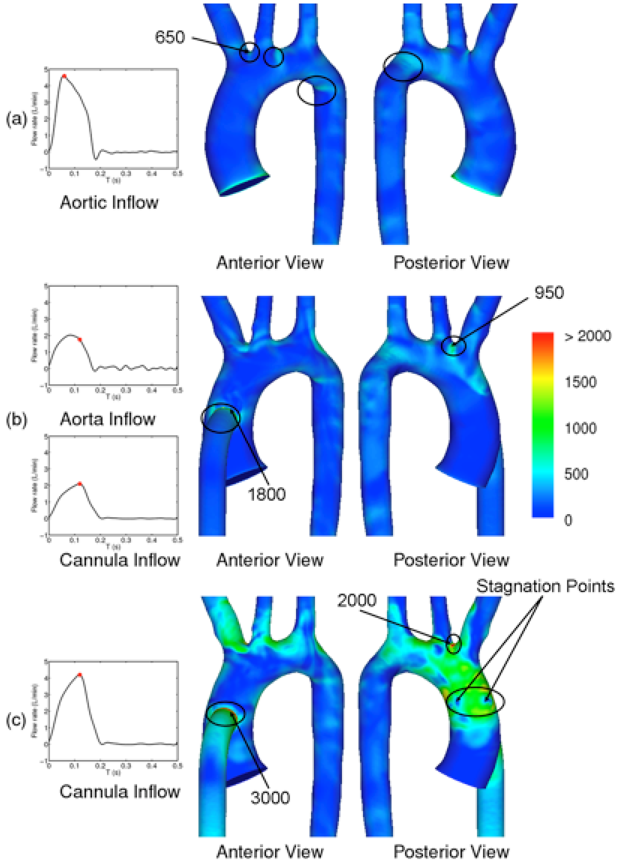


Figure 9. WSS magnitude (unit: dyn/cm2) contour at peak systole: (a) healthy pediatric aorta, (b) 50% bypass, (c) 100% bypass.

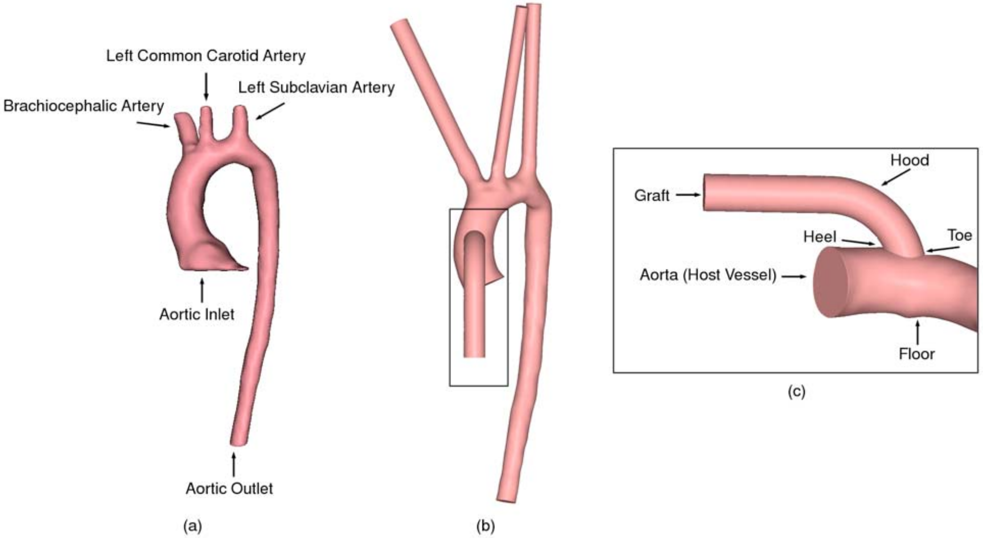
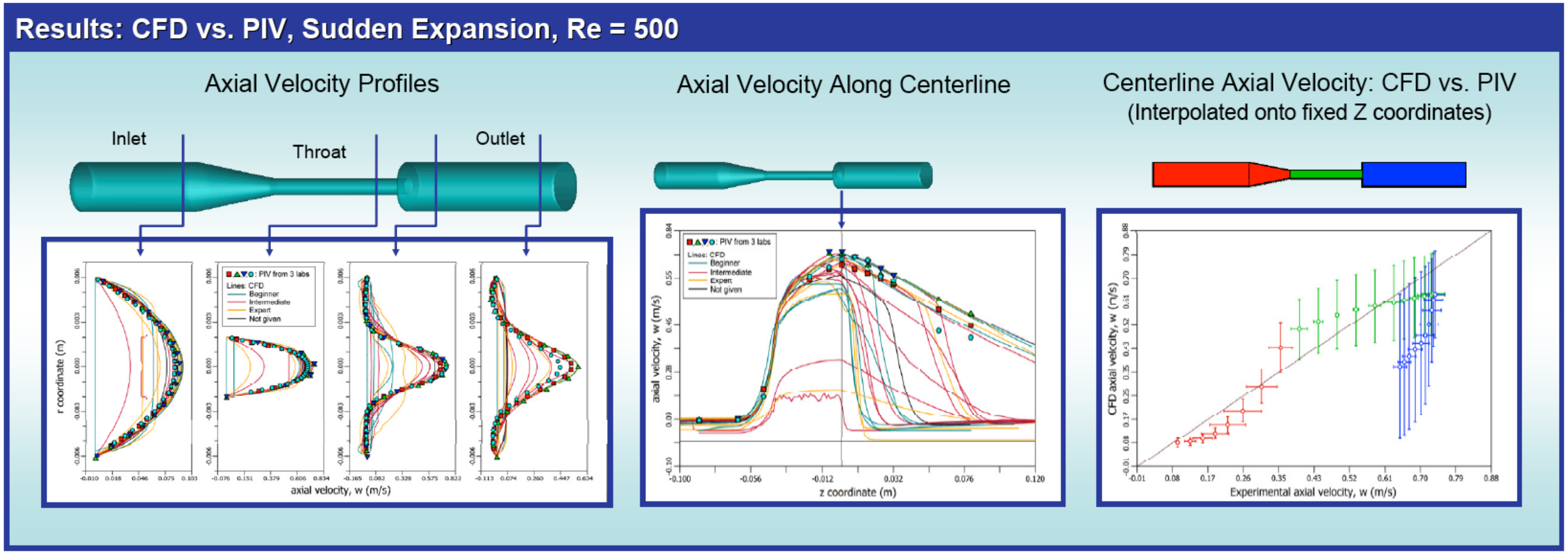


Figure 1. (a) Healthy pediatric patient specific aortic model; (b) Proximal anastomotic model; (c) Enlarged distal view of the graft junction indicated by the black square in Fig. 1 (b).

1. Yang, N., Deutsch, S., Paterson, E., and Manning, K., *Journal of Biomechanical Engineering, vol.* 131, No. 11, 9pp, Nov. 2009.
2. Yang, N., Deutsch, S., Paterson, E. and Manning, K., *Cardiovascular Engineering and Technology, 2009.*
3. Yang, N., Deutsch, S., Paterson, E. and Manning, K., *Journal of Biomechanical Engineering, 2009.*

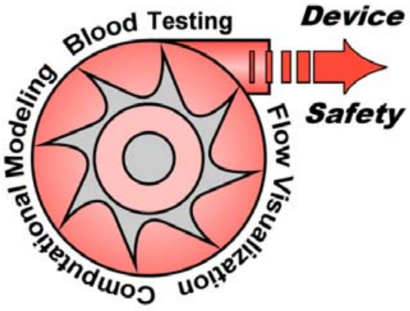
OpenFOAM - Biological and biomedical examples

Solver validation: FDA Critical Path Initiative



FDA

epi
Critical Path Initiative
Food & Drug Administration



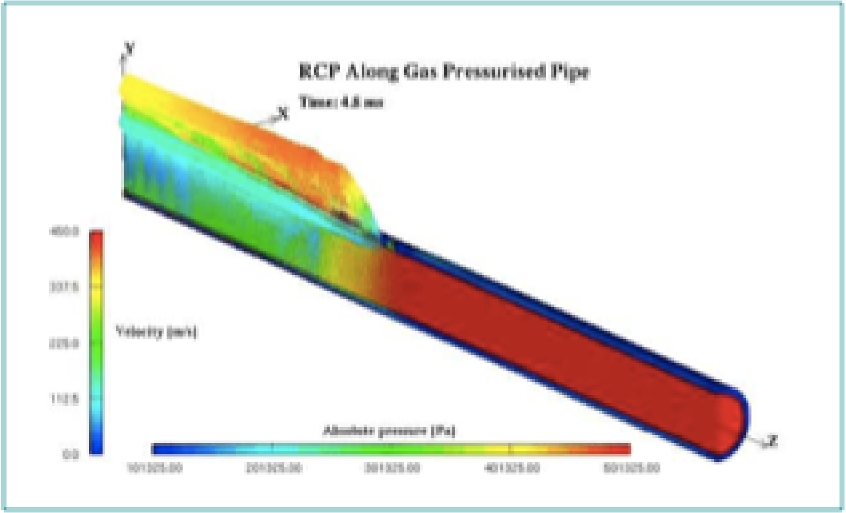
1. Stewart, S.F.C., et al. "Preliminary Results of FDA's Interlaboratory Assessment of Computational Fluid Dynamics and Hemolysis in Medical Devices," FDA/NHLBI/NSF Workshop on Computer Methods for Cardiovascular Devices, June 1-2, 2009.
2. Stewart, S.F.C., et al. "Preliminary Results of FDA's "Critical Path" Project to Validate Computational Fluid Dynamic Methods Used in Medical Device Evaluation," 55th American Society for Artificial Organs Conference, Dallas, Texas, USA, May 28-30, 2009.

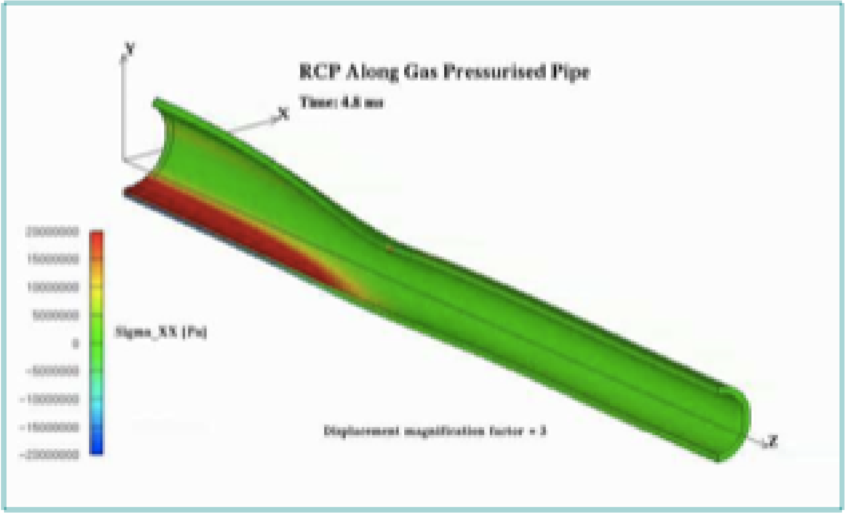
ARL
Penn State

FSI of pipeline failure and crack propagation

Plastic Pipeline Failure: FSI With Crack Propagation

- Internal pressure drives crack propagation; crack opening depressurises the pipeline, dynamically interacting with the force driving crack propagation
- Custom coupled fluid and structures solver; patch-to-patch mapping tools for data transfer
- Crack model implemented as a derived boundary condition: damage model





Jasak, Personal Communication, 2009

Summary

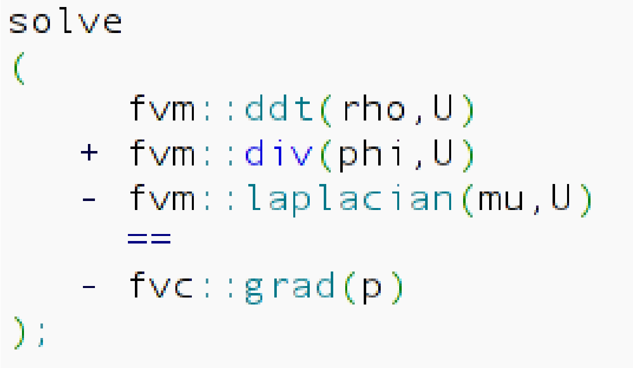
- **OpenFOAM is a rapidly-emerging open-source computational continuum mechanics toolbox with the following advantages:**
  - **Open-source = free as in freedom (freedom to share with other biomedical researchers)**
  - **OpenFOAM architecture is inherently amenable to model sharing**
  - **Custom models, utilities, etc. (e.g., FSI solver, turbulence models, particle deposition models) are developed and installed modularly alongside the standard/baseline OpenFOAM installation**
  - **Ideal for CFD, CSM, and fully-coupled modeling of multiphysics problems (e.g., FSI)**

ARL
Penn State

Summary

- **Because of its use of advanced object-oriented C++ programming concepts (e.g., operator overloading), OpenFOAM utilizes top-level mathematical descriptions of governing equations in human-readable form. Example:**

$$\frac{\partial \rho \mathbf{U}}{\partial t} + \nabla \cdot \phi \mathbf{U} - \nabla \cdot \mu \nabla \mathbf{U} = -\nabla p \quad \longrightarrow$$



```
solve
(
      fvm::ddt(rho,U)
    + fvm::div(phi,U)
    - fvm::laplacian(mu,U)
      ==
    - fvc::grad(p)
);
```

- **OpenFOAM community (currently ~3000 members) is rapidly growing world-wide**
- **Community collaboration via wikis (e.g., http://openfoamwiki.net) and discussion forums (e.g., http://www.cfd-online.com/Forums/openfoam/)**
- **Fair documentation, plenty of tutorials and example cases for most solvers and utilities**