# Parallel Computing & Health Application


***Mehrzad Tartibi**, Mechanical Engineering PhD student*

***Meriem Ben Salah**, Mechanical Engineering PhD student*

***Razvan Corneliu Carbunescu**, Computer Science PhD Student*

***Tony Keaveny**, Chancellor's Professor in Mechanical Engineering and Bioengineering*

***Panos Papadopoulos**, Professor in Mechanical Engineering*

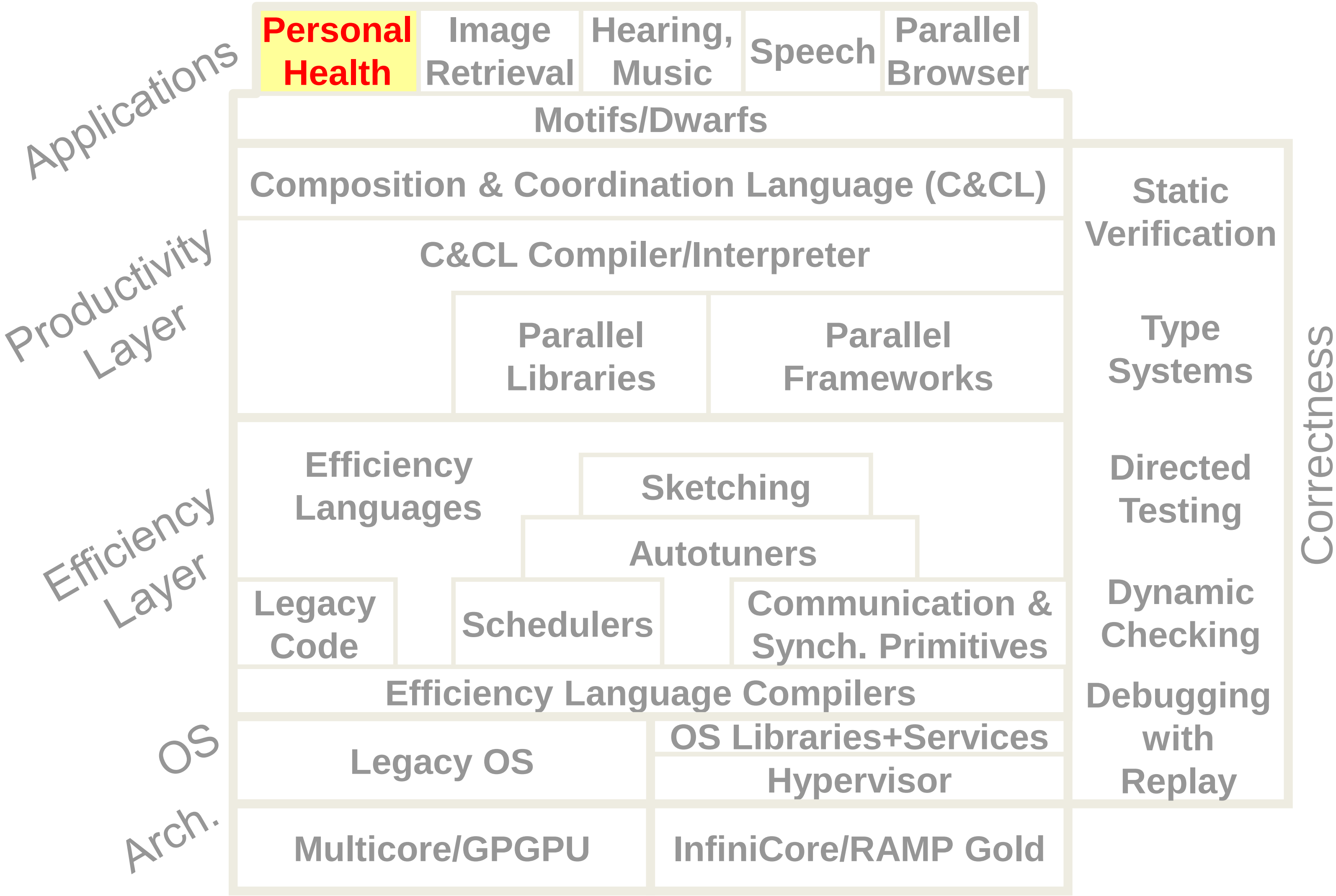
***Phil Collela**, Senior Scientist, LBNL*

***James Demmel**, Professor in Math and Computer Science*

# Outline

- Biomechanical Health Application (Mehrzad)
  - Stroke Diagnosis & Treatment
  - Types of Stroke
  - Treatment Time for Stroke
  - Brain Arterial Network
  - Motivations for Analyzing The Main Brain Arteries
  - Common Simulation Techniques
  - Our Goals
- Solid Modeling of the Arterial Wall (Meriem)
  - A Solid Mechanics Problem
  - Arterial Histology
  - A constitutive Model: an Example
  - Olympus: Finite Element Code Framework
- Fluid Blood Flow Simulation (Razvan)
  - General CFD Problem
  - CHOMBO Library
  - Embedded Boundary Library
- Solid-Fluid Interface (Razvan)

# One of the Five ParLab Applications



# Why Stroke?

- Stroke is the 3rd cause of mortality after heart attack and cancer in developed countries

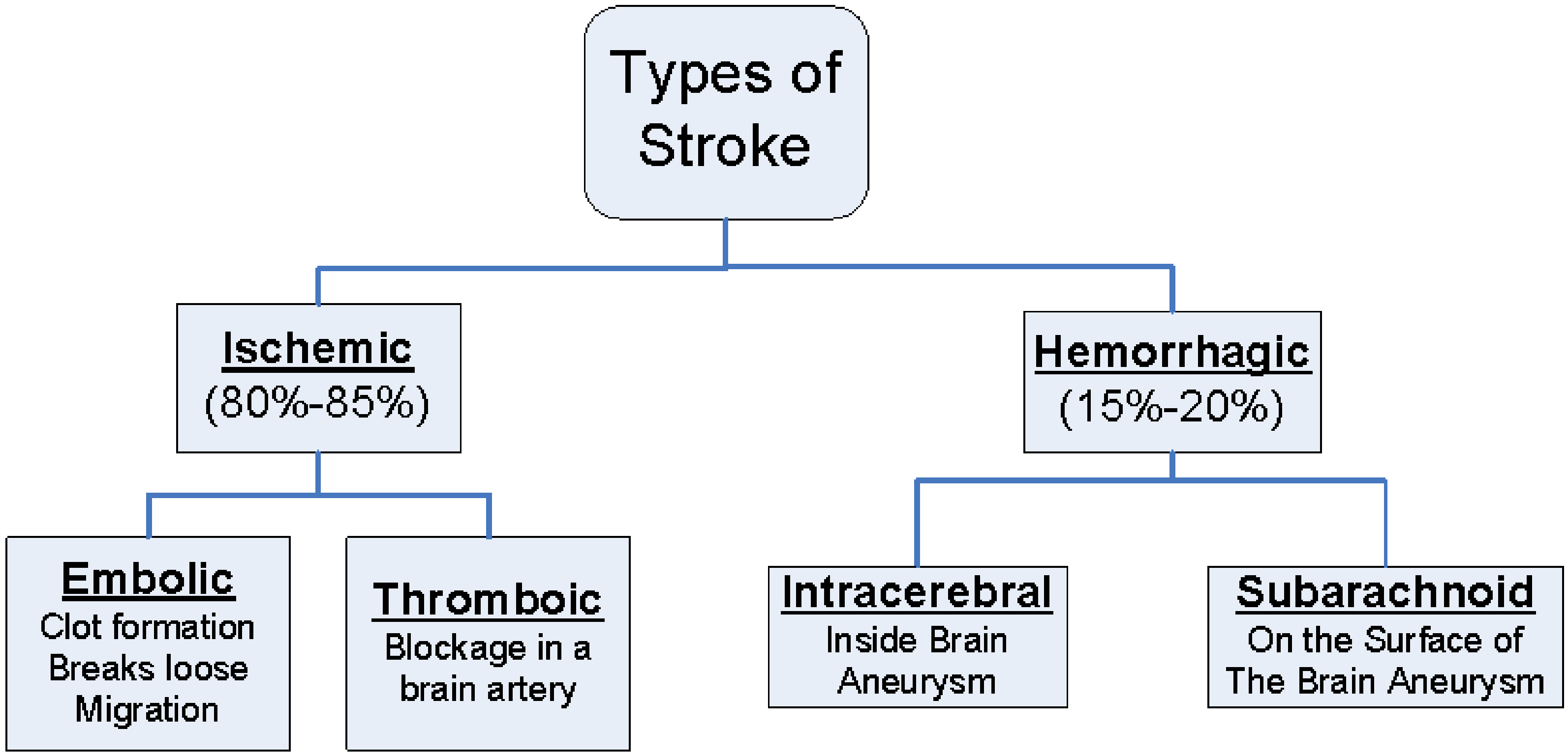
- Common diagnosis procedure:
  - Medical history,
  - Physical and neurological examination,
  - CT scan or MRI and/or Electrical Activity Test and/or blood flow test
  - Study the results

- Common treatment plan, if it is not too late! **<u>A few hours</u>**
  - Blood thinner
  - Iatrogenic

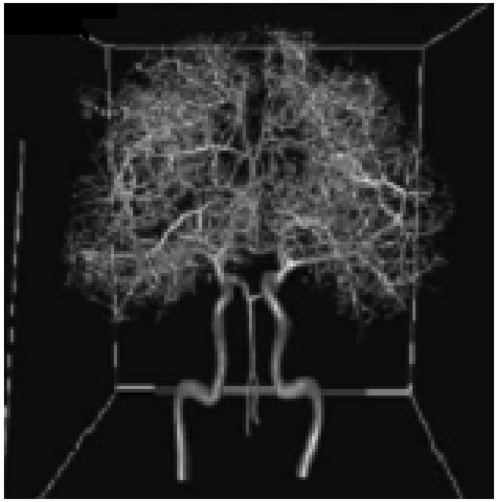
- How ParLab "Stroke Application" can help both diagnosis and treatment?

# Types of Stroke

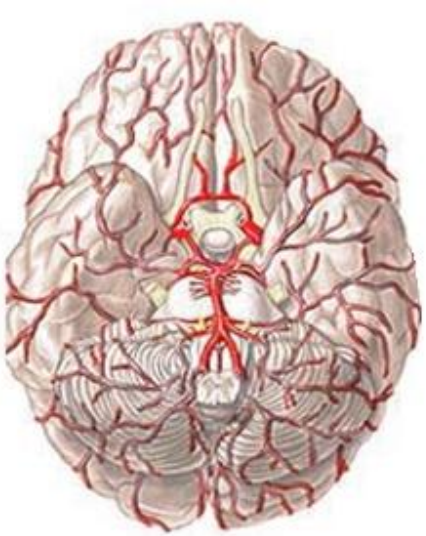
- **<u>Short treatment time for stroke patients is critical</u>**: depending on the size and the location of occlusion or restriction, neurologists have typically only a couple of hours to reduce or prevent permanent damage to brain (blood thinning and/or surgical procedure)



# Brain Arterial Network

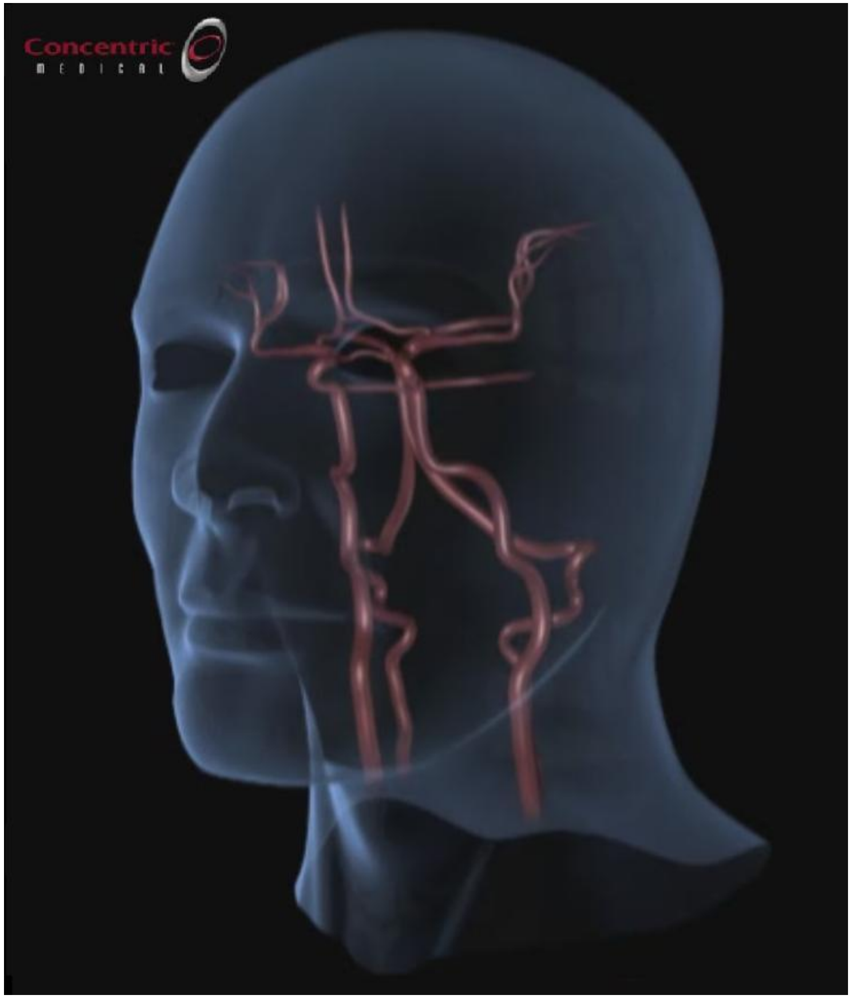


[Cebral]



Bottom view of brain

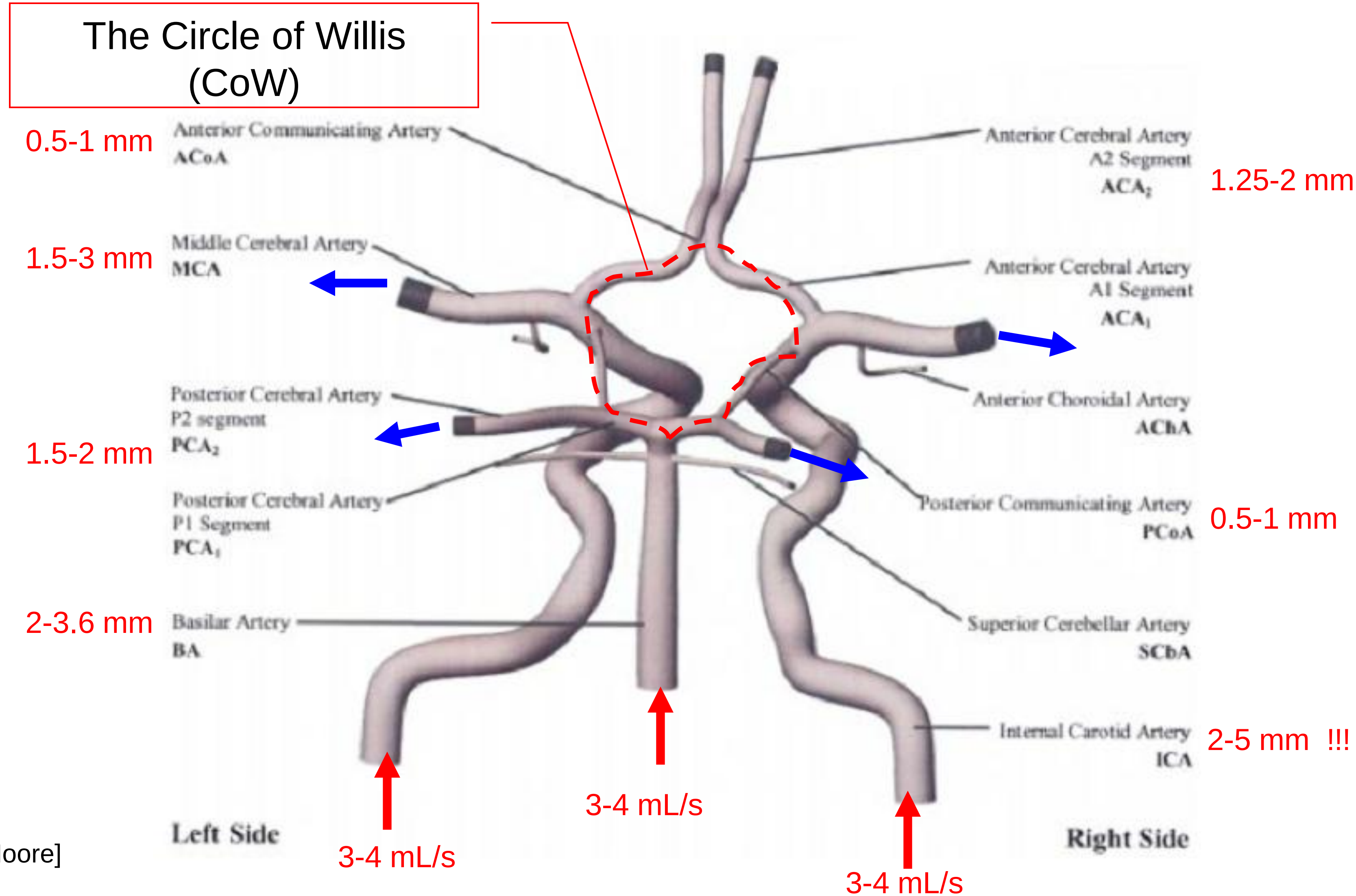
©ADAM, Inc.




- Very complex arterial tree system
- Sizes from a few mm to 7-8 micron
- Almost impractical to capture detailed images of all the arterial network system in timely fashion
- The main part of the brain network is known as The CoW = The Circle of Willis

*http://www.meddean.luc.edu/lumen/MedEd/neuro/neurovasc/navigation/cow.htm*

# A Complete Healthy Circle of Willis




[Moore]

# Motivations for Analyzing The CoW

## **Preoperative Risk Management**

- Assess the clinical tolerance of the patient w.r.t. endovascular intervention:
  - Endarterectomy - two state-of-art procedures:
    - Merci (cork screw)
    - Penumbra
  - Angioplasty (balloon expansion)
  - Stenting

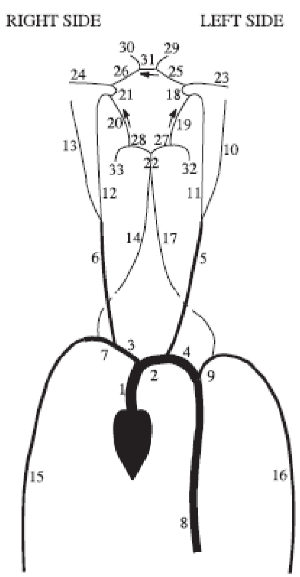
- Patient-Specific analysis of the collateral flow capability of CoW and failure risks under pre/post treatment condition

- Enormous number of possible variations
  - 40-50% of the population has complete CoW
  - 20% has incomplete CoW, but fully functional
  - Absent vessels, fused vessels, string like vessels and accessory vessels

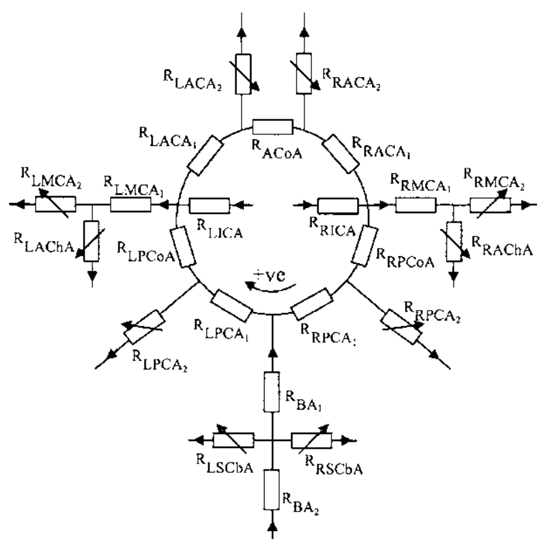
# Common Simulation Techniques

- 1D-0D models (Rigid or Elastic Vessel walls)
  - Full main arterial model
  - CoW model

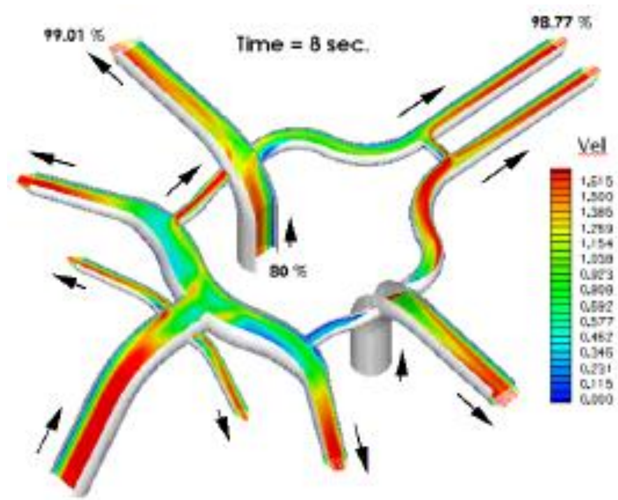
- 3D-0D models (Rigid Vessel walls)
  - Flow simulation of only main CoW vessels




[Alastruey]



[Moore/ Moorehead]




[Kim]

# Our Goal

- The first true 3D FSI (Fluid-Solid Interaction) analysis of full CoW

- Use MRI to build an accurate mathematical model of the CoW – including the actual clot/blockage shape
  - Literature search so far: researchers have been simply simulating known idealized occlusion or stenosis (details from MRI images)

- Conduct FSI (3D-0D/1D) analysis
  - Assess the arterial wall stress levels and failure under pre & post treatment conditions
  - Investigate the effect of the autoregulation (flow variation in CoW based on demand).

- 3D high resolution model of CoW leads to better understanding of the FS system and perhaps leads to new therapeutic ideas

# Our Goal and Parallel Computing

- Time is an essence in dealing with the stroke patients
- Rapid patient-specific 3D FSI analysis of CoW provides invaluable insight to the neurologist surgeons to
  - understand the problem (stroke or risk of stroke) quantitatively
  - evaluate the risk of the pre & post treatment
  - choose the treatment plan
  - evaluate the risk of the treatment plan
- Treat the stroke patients to live while minimizing the damage
- Requiring high computational power for highly-accurate simulations in near real-time.
- Many-core is needed because of:
  - Need to keep patient files and patient specific data within hospital
  - Patients can’t wait for allocation on supercomputers

# A Solid Mechanics Problem

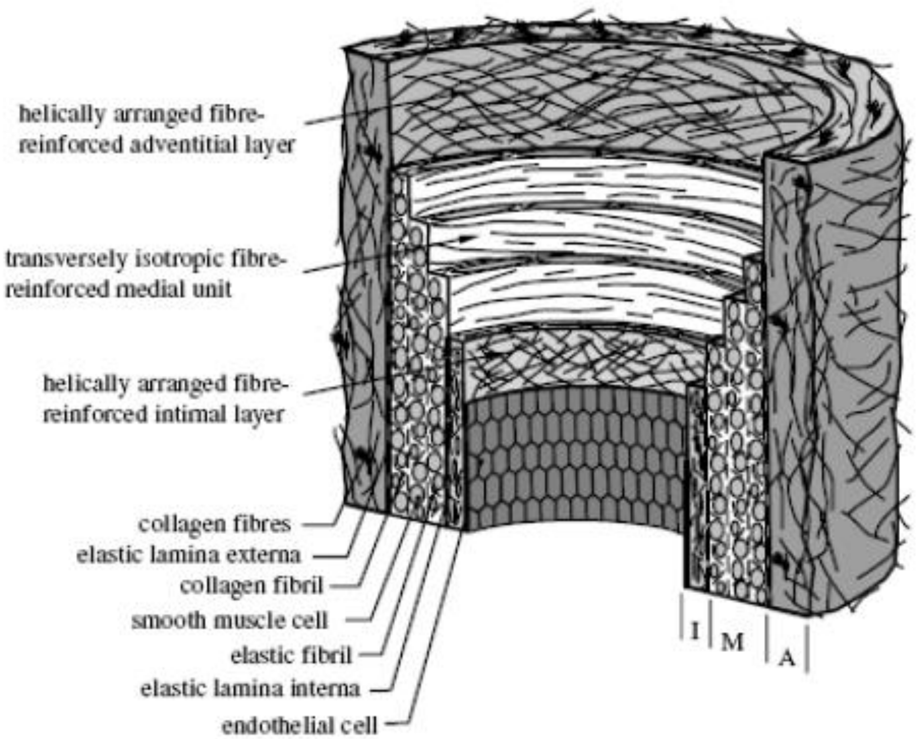
- Needed are:
  - Constitutive Model
    - Rheological, phenomological or structural description of the material behavior
  - Kinematics
    - Geometry definition
    - Deformation measures
  - Weak Form:
    - Weakening the assumptions on the solution.
    - Solved numerically
    - Boundary and intial conditions

# Arterial Histology

- Three different Layers

- Collagen fibers
  - are embedded in a soft smooth muscle cell, blood
  - are crimped in the unstressed arterial tissue
  - have no compressive strength
  - Randomly distributed: anisotropy

- Healthy arteries:
  - are highly deformable with a nonlinear strain response
  - behave as nearly incompressible solids

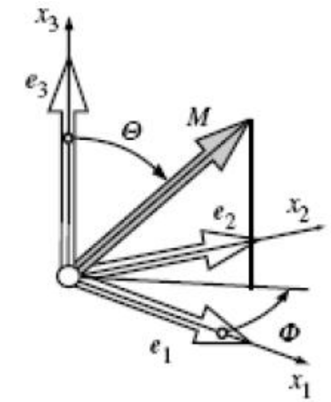


# Constitutive Model: An example

- Statistical fiber-reinforced structural models: Collagen fibers are embedded into a continuum (the groundmatrix).

- Every fiber can be characterized by a referential unit direction

$$\boldsymbol{M}, \qquad |\boldsymbol{M}| = 1$$



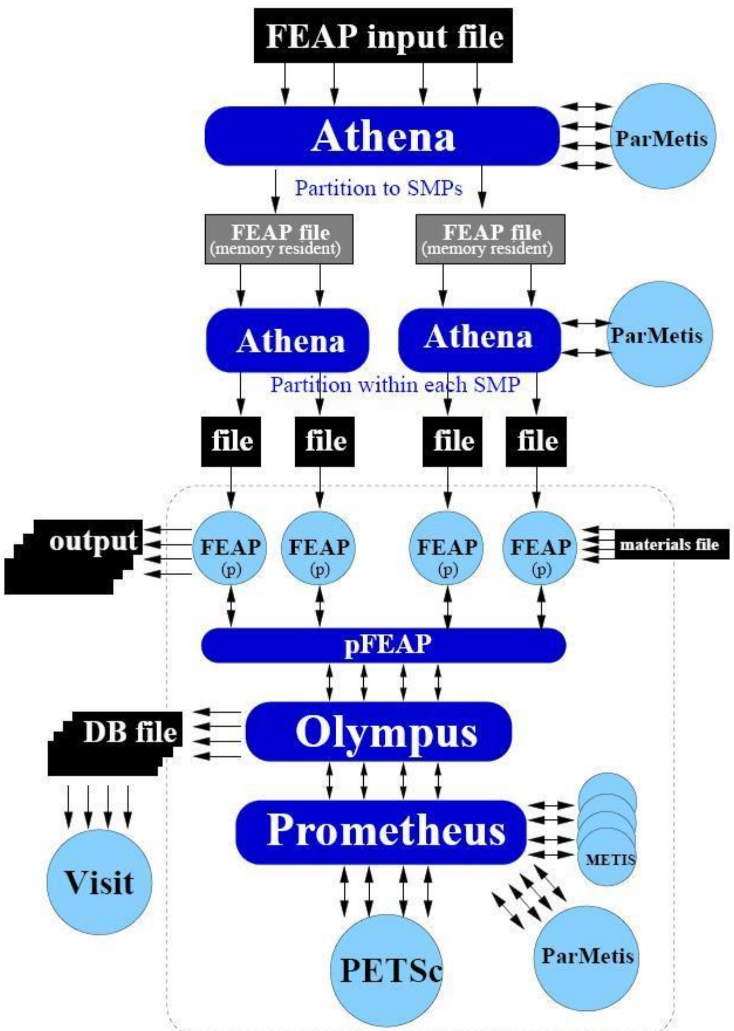

- Structure tensor to represent the fiber distribution:

$$\begin{aligned} \boldsymbol{H} &= \frac{1}{4\pi}\int_{\omega} \rho(\boldsymbol{M}(\Theta,\Phi))\boldsymbol{M}(\Theta,\Phi)) \otimes \boldsymbol{M}(\Theta,\Phi)\mathrm{d}\omega, \\ \mathrm{d}\omega &= \sin\Theta \mathrm{d}\Theta \mathrm{d}\Phi \end{aligned}$$

- Additive split of the energy stored in the groundmatrix and the collagen reinforcement:

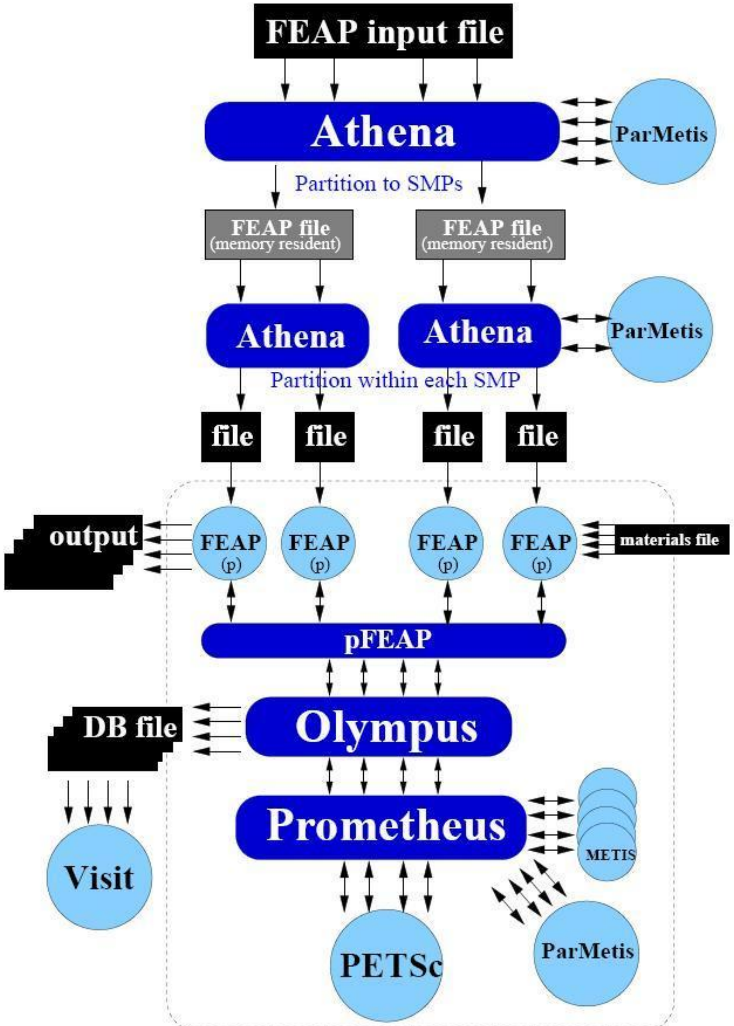
$$\overline{\Psi}(\overline{\boldsymbol{C}}, \boldsymbol{H}_i) = \overline{\Psi}_g(\overline{\boldsymbol{C}}) + \sum_{i=1,2} \overline{\Psi}_{fi}(\overline{\boldsymbol{C}}, \boldsymbol{H}_i(\boldsymbol{a}_{0i}, \kappa))$$

# Olympus: Finite Element Code Framework



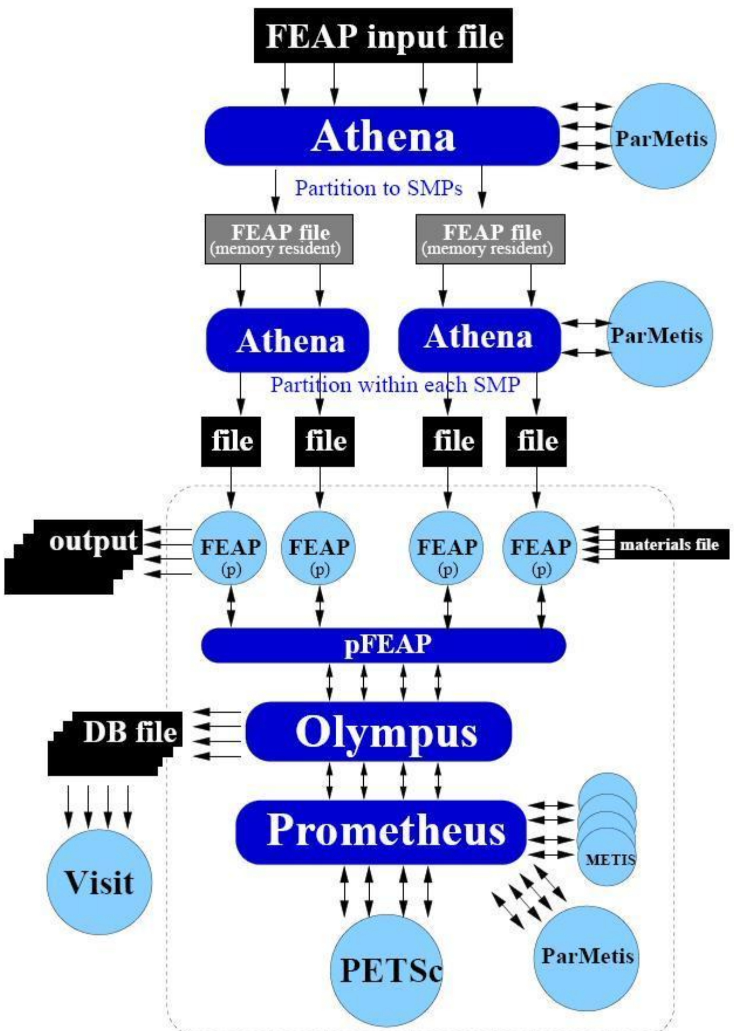

- **Olympus:** governs all the sub-modules and libraries.
- Used to simulate vertebral bone with over a billion of degrees of freedom on 4088 processors-machine
- 1.5 minutes per linear solve vs.7 mn per linear solve for 115 millions dof.
- **Input:**
  - Unstructured complex geometry
- **Athena:**
  - C Code
  - Communication between ParMetis and the processors.

# Olympus: Finite Element Code Framework




- **ParMetis:** Parallel Mesh Partitioner for the construction of finite element sub-problem on each processor.
  - converts the mesh into a graph
  - non-overlapping partition
- The creation of the mesh partitioning is done recursively
- Overlapping of the partition is enforced (ghost nodes)
- Avoid communication but redundant data on the processors
- **FEAP:** Serial general purpose finite element code.

# Olympus: Finite Element Code Framework




- **pFEAP:** A thin parallelizing layer for FEAP, that primarily maps vector and matrix quantities between the local FEAP problem and the global Olympus operator.
- Data associated to the ghost nodes is:
  - ignored when assembling
  - updated when going from local to global
- **PETSc:** used for managing the forward and backward scatter and gather information between processes
- **Prometheus:**
  - Parallel scalable Multigrid Algebraic Solver
  - Solution of the diffusion operator in unstructured large-scale elasticity problems
  - Preconditioning with CG

# Fluid BloodFlow Simulation

- Fluids have been simulated for 30 years by solving the CFD Navier-Stokes equations (continuity, momentum and energy)

- Many frameworks exist for simulating CFD problems but we wanted one that would enable an accurate scalable simulation and also be able to in the future relatively easy port the code to manycore

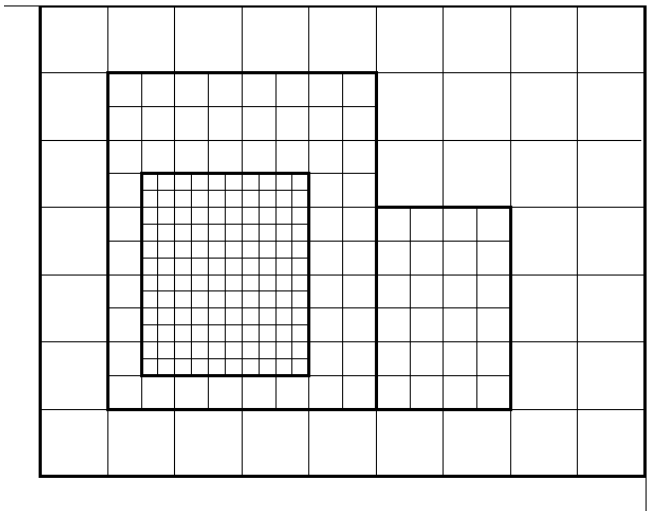
- The bloodflow in the arteries and CoW will be simulated as an incompressible fluid using two libraries developed at LBL:

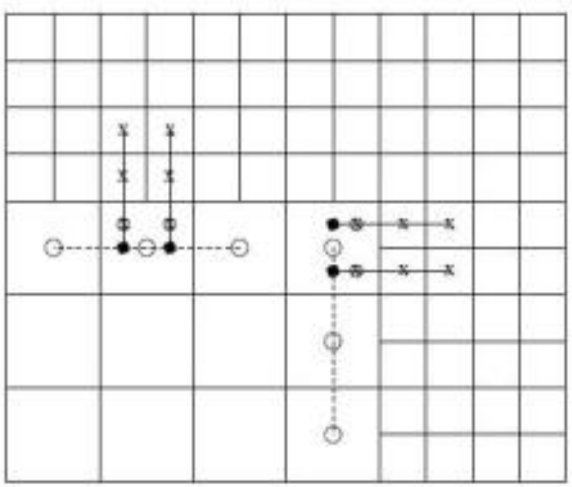
    - ***The CHOMBO Library*** which provides a set of tools for implementing finite difference methods for the solution of partial differential equations on block-structured adaptively refined rectangular grids

    - ***The Embedded Boundary Library*** will be used to model the fluid solid boundary (blood vessel wall)

- The fluid framework presented has been successfully run on more than 4000 processors

# Chombo Library





- Chombo uses a multitude of flexible/portable C++ classes and structures built on top of fast efficient legacy Fortran 77 code.

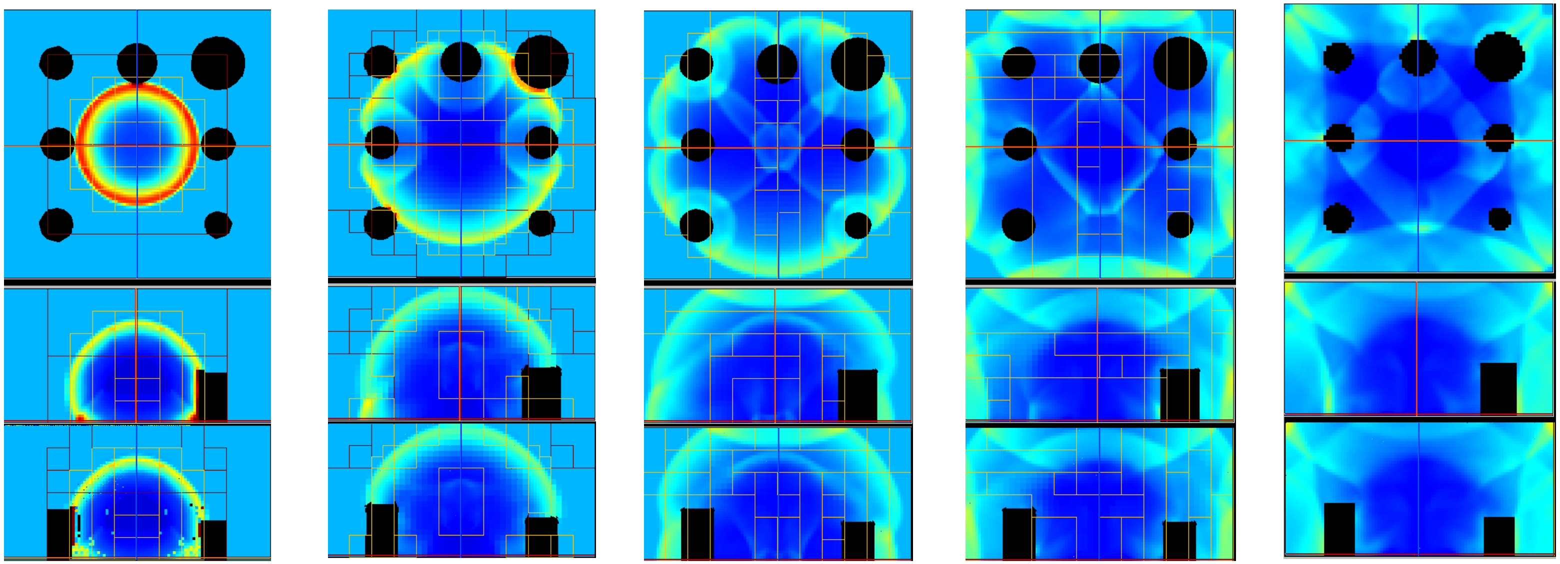
- Chombo is organized into 5 parts:

-**BoxTools**: *Provides infrastructure for calculations over unions of rectangles.*

-**AMRTools**: *Provides tools for data communication between refinement levels, including coarse-fine interpolation tools.*

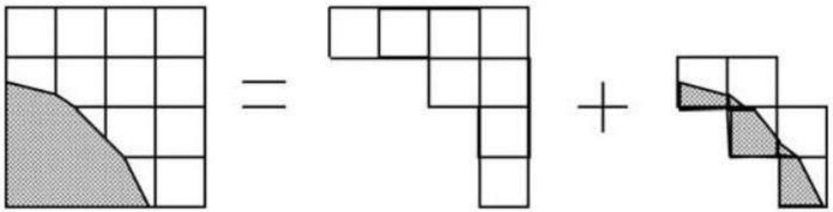
# Chombo Library

- **AMRTimeDependent**: *Manages sub-cycling in time for time-dependent adaptive calculations.*
- **AMRElliptic**: *A multigrid-based elliptic equation solver for adaptive hierarchies.*
- **ParticleTools**: *Provides the particle support for the Chombo library.*

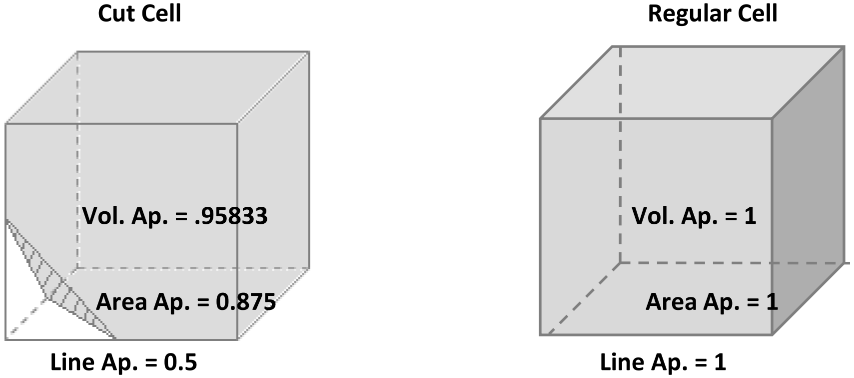


# Embedded Boundary Library

- Uses a Cartesian grid with special cells at the boundary which are called cut cells or irregular cells



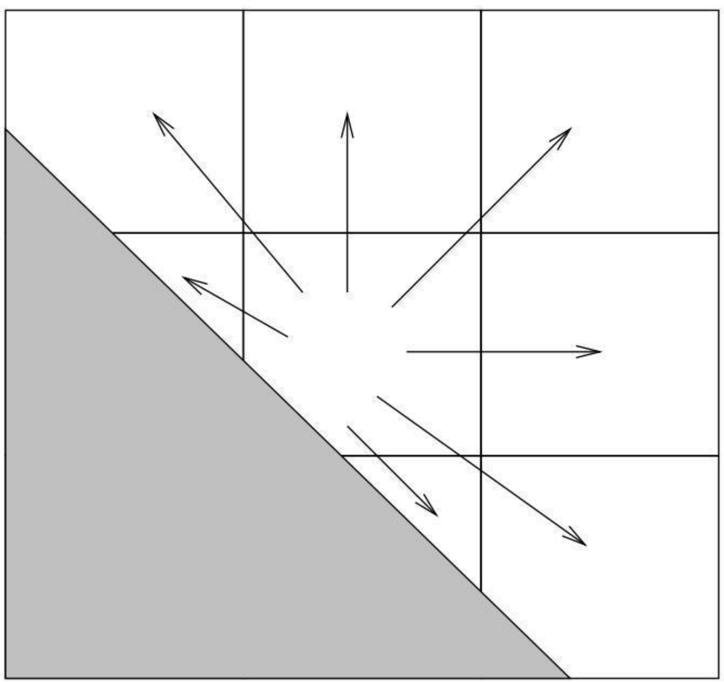
- Special nondimensional values like the apertures are kept for the irregular cells and these are used in the computation of the cell centroid, boundary normal

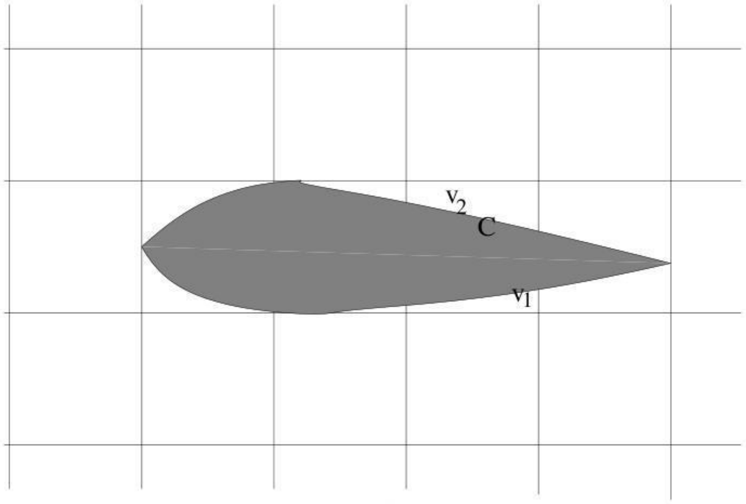


# Embedded Boundary Library

- Non-conservative cells may be used for the boundary with the mass error introduced being spread as diffusive terms on the neighboring cells

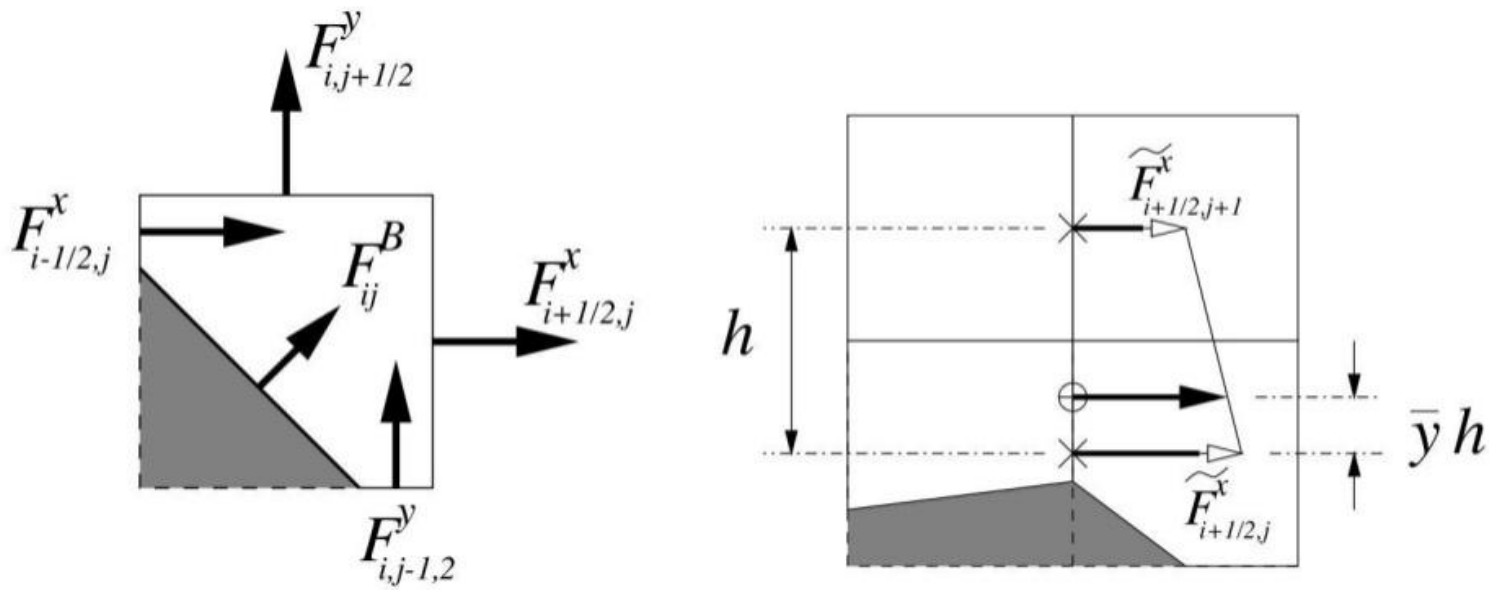
- The method also allows for easy calculation of desired values on the fluid cells regardless of the representation used to specify the boundary's location





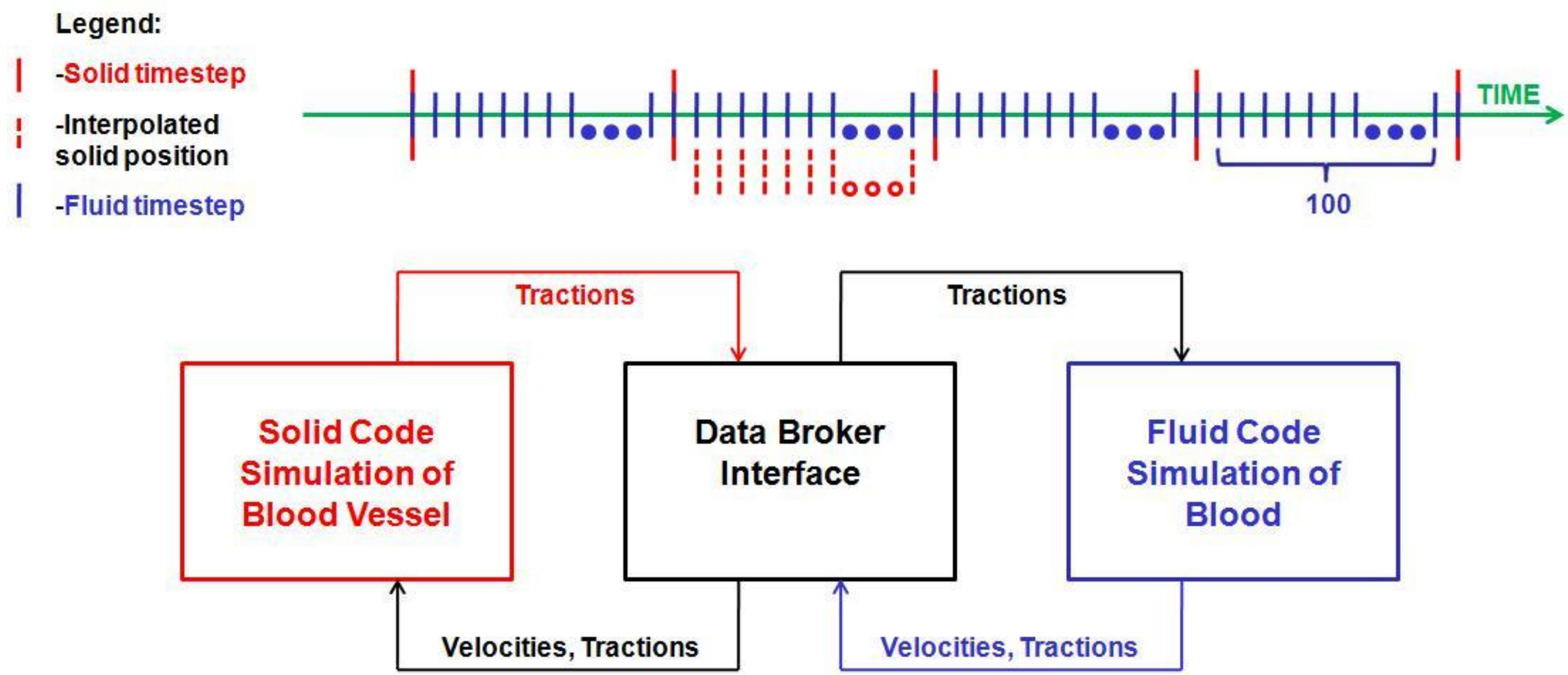
# Embedded Boundary Library

- The fluxes on the special cut cells are calculated by interpolating values on neighboring full cells for the side fluxes; and from the normal direction for the flux on the part of the boundary contained in the cell



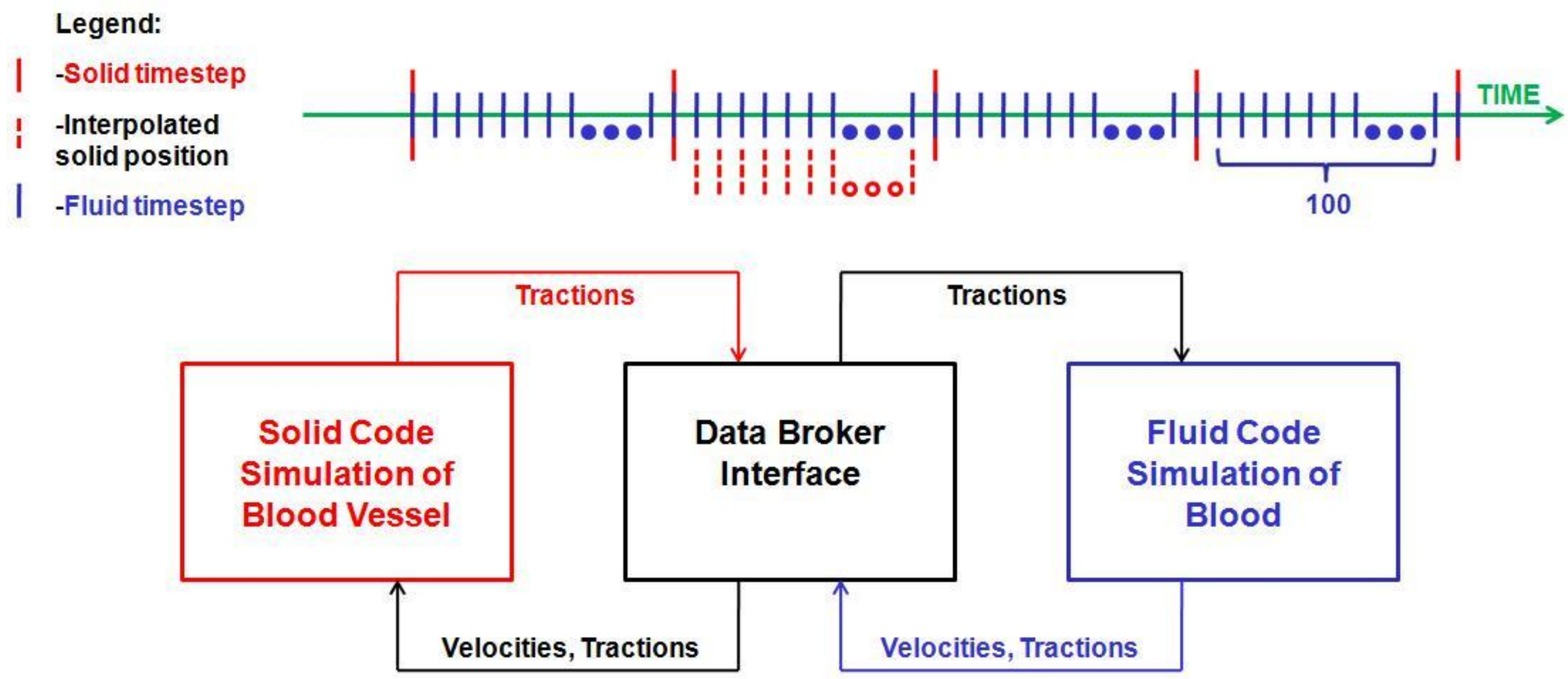

- To support the simulation of the blood vessels the Embedded boundary library will be extended to incorporate and facilitate moving embedded boundaries for the Navier-Stokes equations

# Solid-Fluid Interface




- Full monolithic coupling the two sets of PDEs would require much more computational power than is available for desired accuracy
- The problem will be solved using a fractional step method with sufficiently small time-step to allow for accurate coupling.
- Based on the physics of the flow, it is estimated that here will be roughly 100 time-steps of fluid code to every solid code time-step because of the extra requirement of the CFL stability condition on the fluid code.

# Solid-Fluid Interface




- Transfer data between the 2 different meshes
- Interpolation of necessary values
- Matching and balancing for communication reduction
- Providing an interpolation of the solid motion to restrict the movement of the boundary to within a single grid cell of the fluid time-step.

# Conclusions and Future Work

- Combine the 2 parallel codes and run the program on large clusters at LBL and UC Berkeley

- Larger runs on these types of systems at much higher resolutions will be used to verify the accuracy of lower resolution clinical runs

- Port the codes from distributed memory to manycore shared memory (taking advantage of the locality that already exists within these parallel codes)

- Try combining all the tuning options available from ParLab to try to get an acceptable run on the existing manycores or on RAMP as a proof of concept

- Collaborating with radiologists on fast, accurate image extraction from MRI data

# Questions ?

*Contact information :*

*Mehrazad Tartibi – mtartibi@berkeley.edu*

*Meriem Ben Salah – meriem.ben.salah@berkeleu.edu*

*Razvan Carbunescu – carazvan@cs.berkeley.edu*