High Performance Computing with Feel++ Applications and Numerical methods

Outline

1. Cemosis and multi-disciplinary interactions
2. Some HPC Applications in Health and Physics
3. Computational Framework Feel++

The Team

IRMA Cemosis

Researchers

C. Prud'homme

M. Szopos

Research Engineers

A. Ancel V. Chabannes

V. Huber F. Der Hovsepian

Phd Students

D. Prada G. Dollé

R. Tarabay J.B. Wahl

R. Hild C. Daversin

L. Sala

Master Students

Marie Houillon

LJK LIPHY

Researchers

M. Ismail

E Maitre

Post-doctorate

T. Metivet

Phd Students

A. Sengers

IMFT

Post-doctorate

V. Doyeux

LNCMI

Research Engineers

C. Trophime

Collaborators

F. Nataf

P. Jolivet

S. Bertoluzza (CNR,It)

M. Pennacchio (CNR, It)

G. Guidoboni (IUPUI,IRMIA)

R. Sacco (Politecnico di Milano)

Cemosis

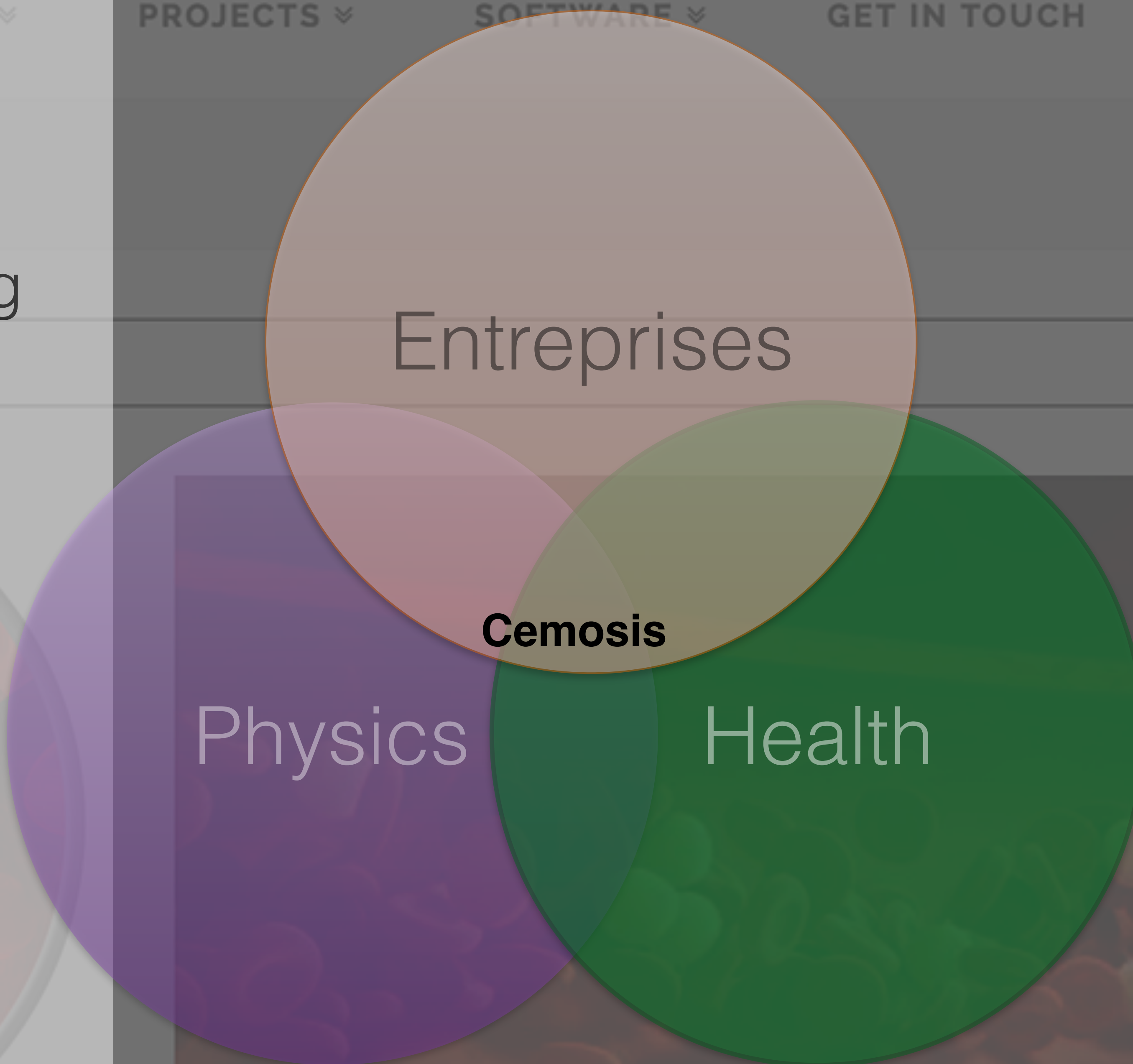
and

multi-disciplinary interactions

Cemosis

http://www.cemosis.fr

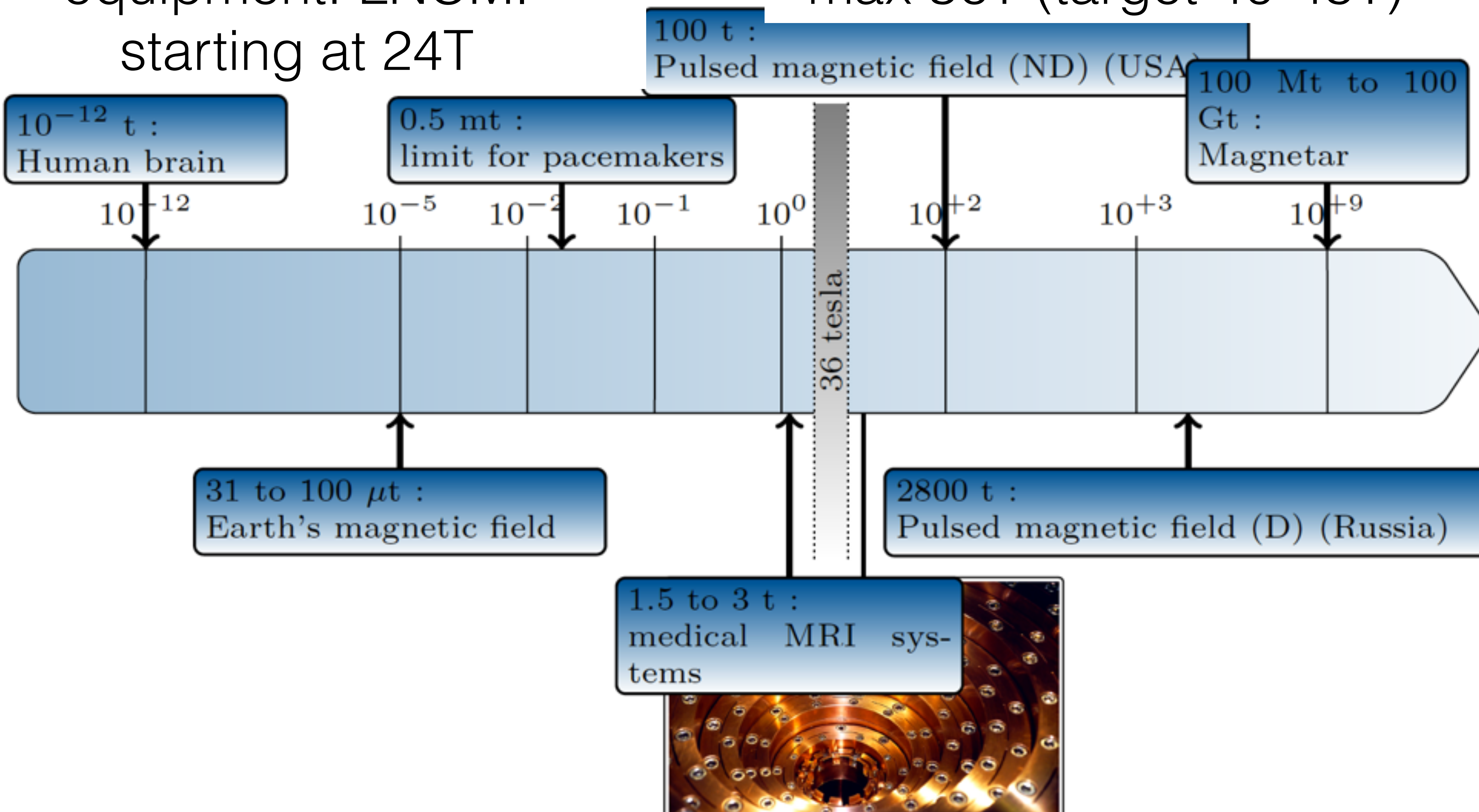
- A structure from IRMA to federate modeling and simulation in the Strasbourg region
- 3 Axes
  - Projects
  - Training
  - Software



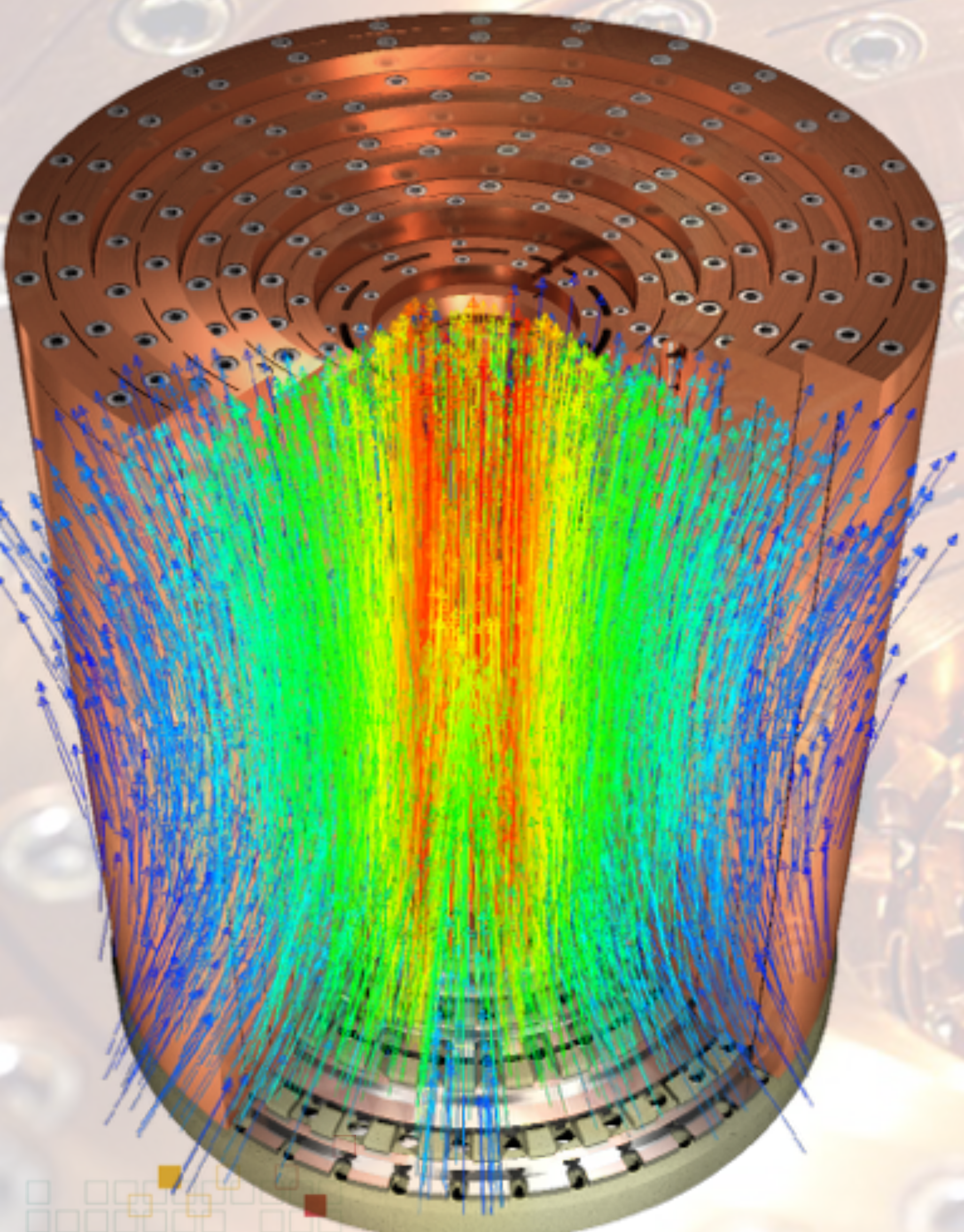
High Field Magnets

Large scale CNRS equipment: LNCMI starting at 24T

Grenoble: continuous field max 36T (target 40-45T)



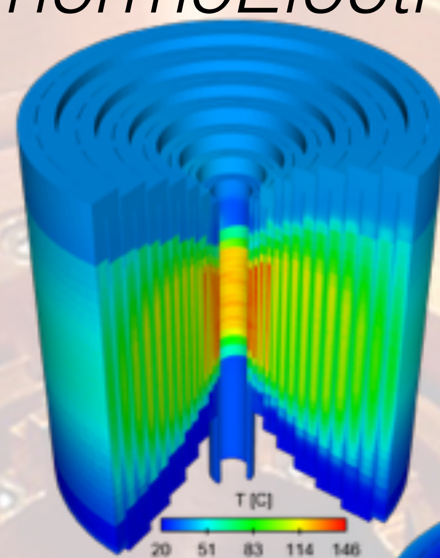
High Field Magnet
A multiphysics application with a CNRS Large Equipment (LNCMI)

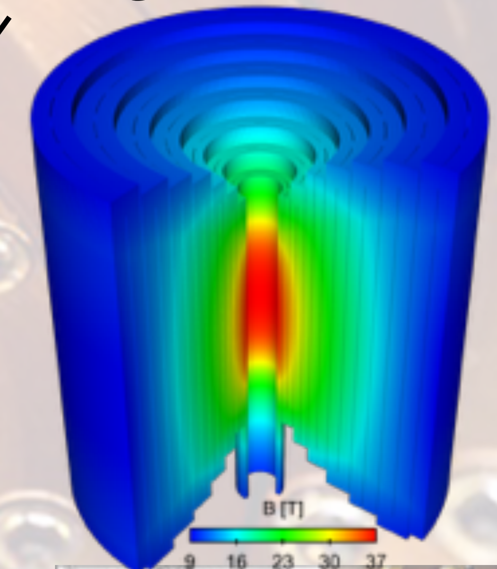


A Virtual lab for modeling and simulating high field magnets (36T and nextgen up to 43T) using Feel++

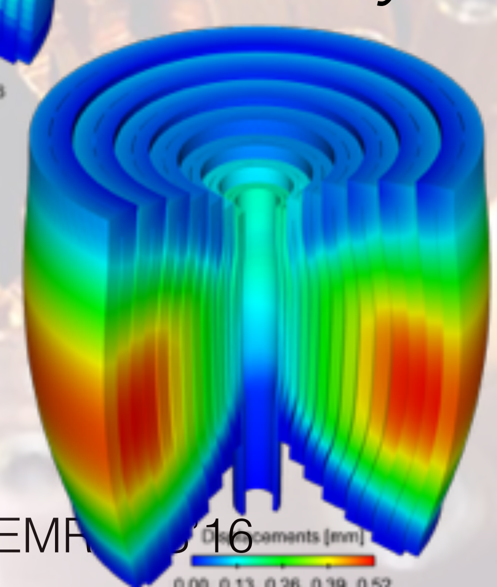
Magnetostatic

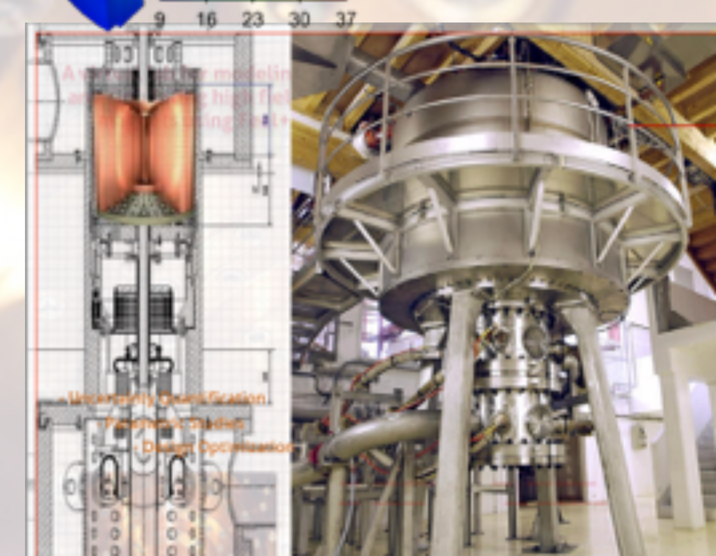
ThermoElectric



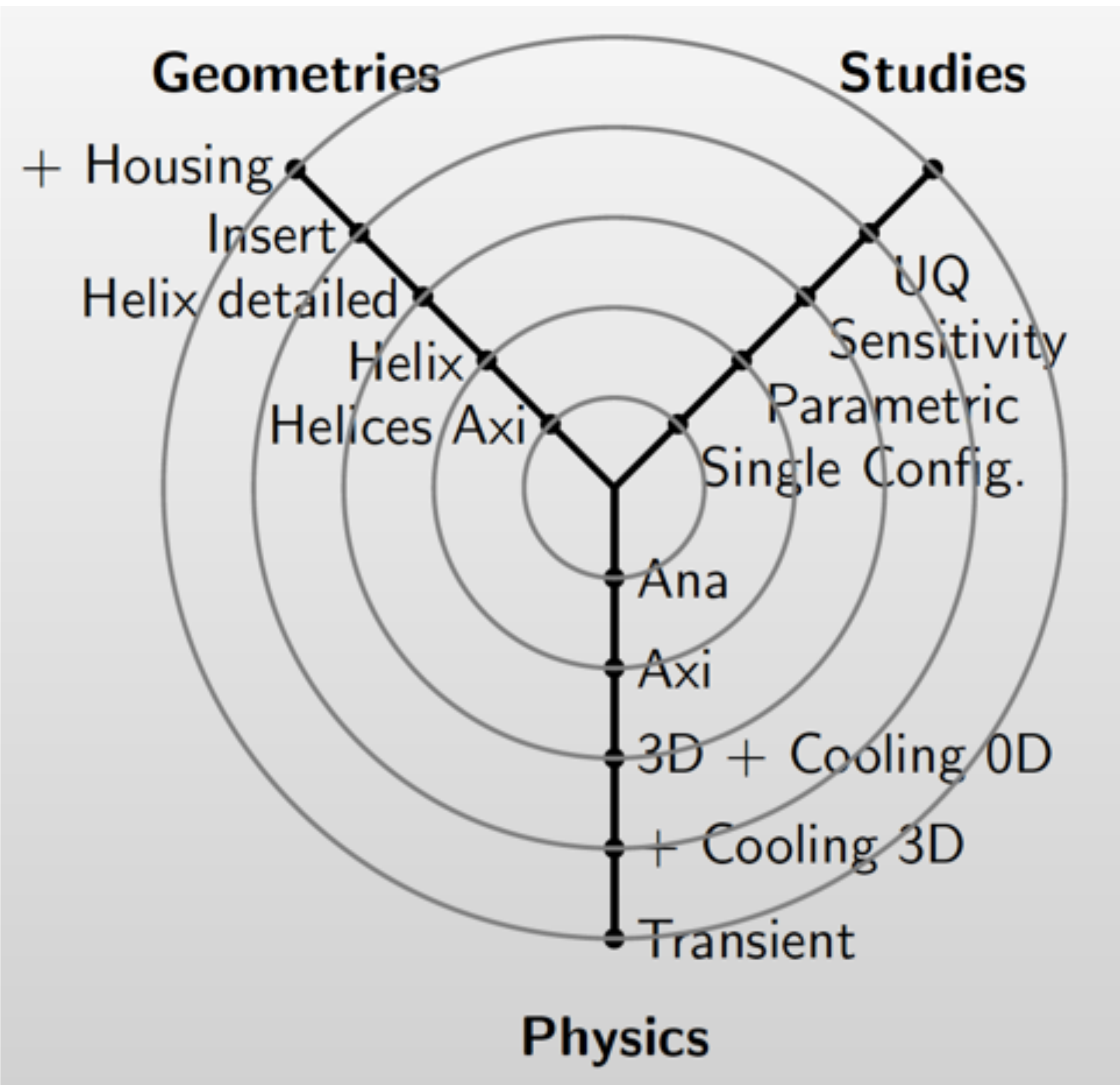


Elasticity





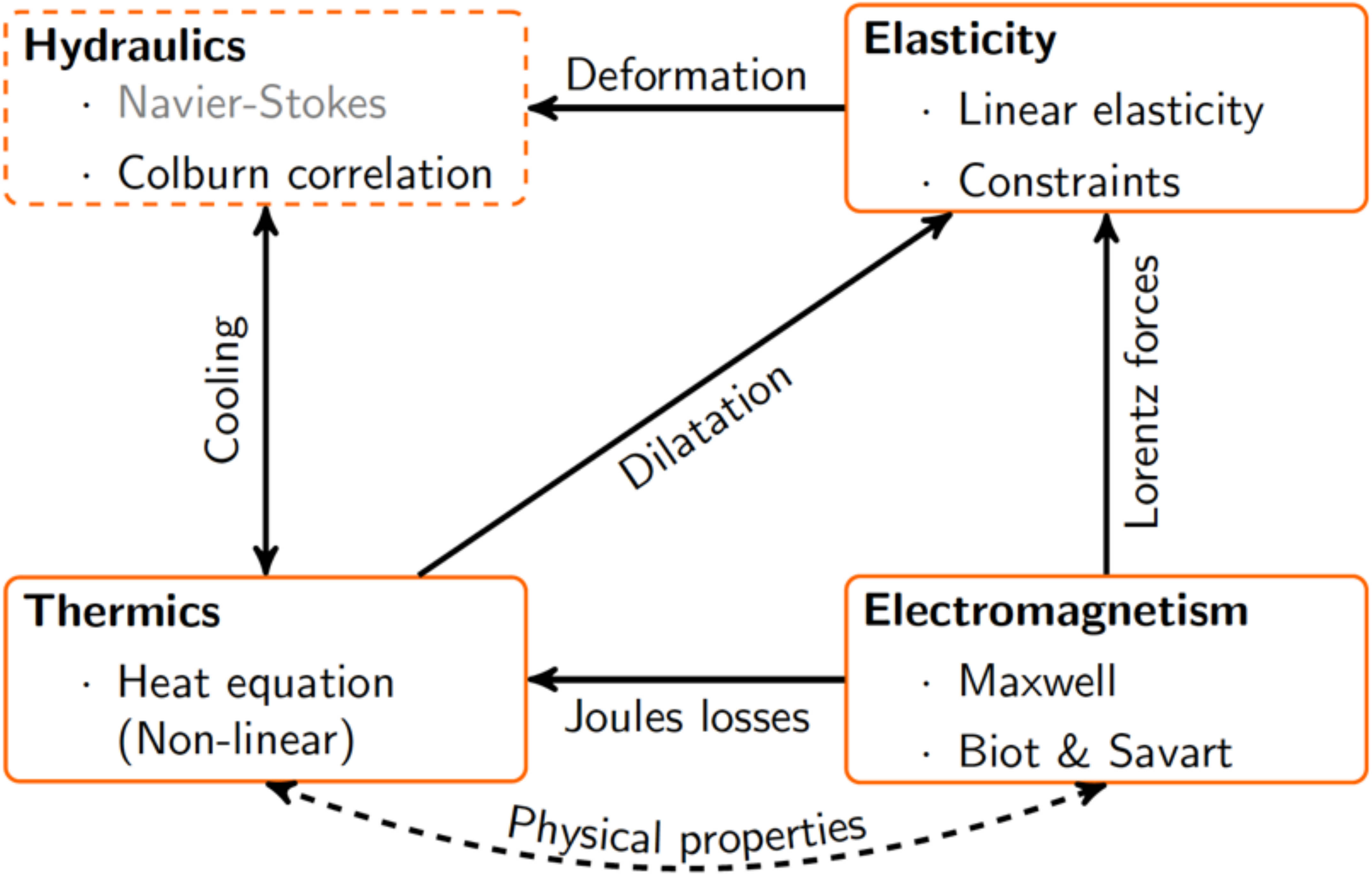
High Field Magnets



Requirements

- Generic and extensible
- ROM for Multi-Physics
- From laptop to HPC
- Use Open Source software

High Field Magnets



High Field Magnets

$$\begin{tikzcd}
\mathbb{R} \arrow[r, "id"] & H_1(\Omega) \arrow[r, "\nabla"] \arrow[d, "\pi_{H_1}"] & H_{\mathrm{curl}}(\Omega) \arrow[r, "curl"] \arrow[d, "\pi_{H_{\mathrm{curl}}}"] & H_{\mathrm{div}}(\Omega) \arrow[r, "div"] \arrow[d, "\pi_{H_{\mathrm{div}}}"] & L_2(\Omega) \arrow[d, "\pi_{L_2}"] \\
\mathbb{R} \arrow[r, "id"] & U_{\mathcal{N}} \arrow[r, "\nabla"] & V_{\mathcal{N}} \arrow[r, "curl"] & W_{\mathcal{N}} \arrow[r, "div"] & Z_{\mathcal{N}}
\end{tikzcd}$$

$$\mathbb{R} \xrightarrow{id} \mathtt{H1} \xrightarrow{\mathtt{Igrad}} \mathtt{Hcurl} \xrightarrow{\mathtt{Icurl}} \mathtt{Hdiv} \xrightarrow{\mathtt{Idiv}} \mathtt{L2}$$

```
H1 = Pch<k>(mesh);   Hcurl = Nh<k=0>(mesh);
L2 = Pdh<0>(mesh);   Hdiv = Dh<k=0>(mesh);
```

```
Igrad = Grad( _domainSpace=H1, _imageSpace=Hcurl);
Icurl = Curl( _domainSpace=Hcurl, _imageSpace=Hdiv);
Idiv = Div( _domainSpace=Hdiv, _imageSpace=L2);
```

$B = \nabla \wedge A \longrightarrow$ `B=Icurl(A);`

High Field Magnets

Electro-thermal model

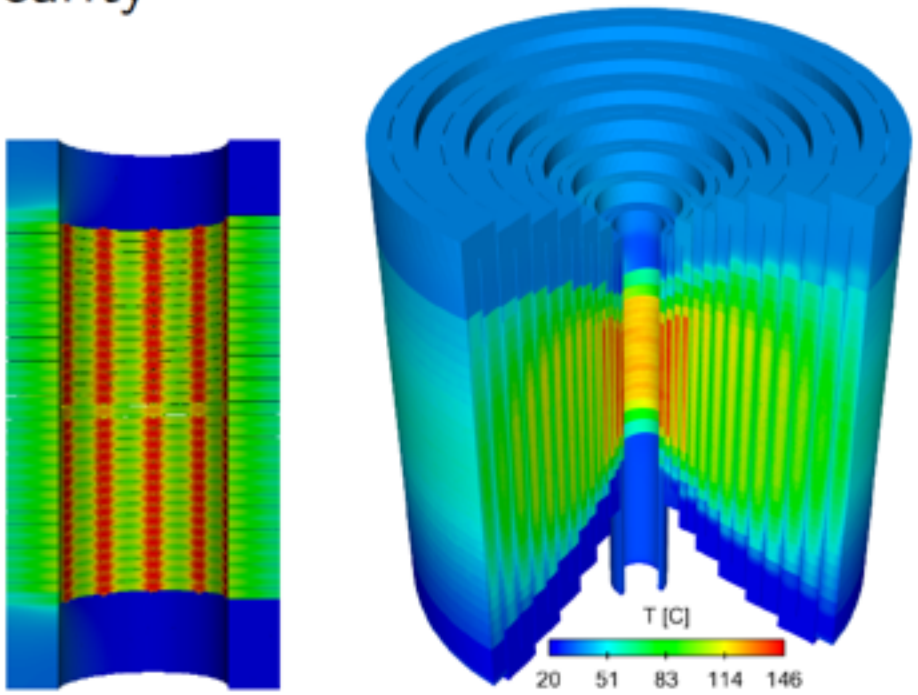
V : electric potential $[V]$

T : temperature $[K]$

$$\begin{cases} -\boldsymbol{\nabla} \cdot (\boldsymbol{\sigma}(\boldsymbol{T}) \boldsymbol{\nabla} \boldsymbol{V}) = \mathbf{0} \text{ in } \boldsymbol{\Omega} \\ -\boldsymbol{\nabla} \cdot (\boldsymbol{k}(\boldsymbol{T}) \boldsymbol{\nabla} \boldsymbol{T}) = \boldsymbol{\sigma}(\boldsymbol{T}) \boldsymbol{\nabla} \boldsymbol{V} \cdot \boldsymbol{\nabla} \boldsymbol{V} \text{ in } \boldsymbol{\Omega} \end{cases}$$

$$\sigma(T) = \frac{\sigma_0}{1 + \alpha(T - T_0)} \xleftrightarrow[\text{Non linearity}]{\text{Material properties}} k(T) = LT\sigma(T)$$

- Applied potential
  $\rightarrow \; V = 0 \text{ (in)} \quad V = V_D \text{ (out)}$
- Water / Glue electrically isolant
  $\rightarrow \; -\sigma(T)\nabla V \cdot \bar{n} = 0$
- No thermic exchange with air / glue
  $\rightarrow \; -k(T)\nabla T \cdot \bar{n} = 0$
- Thermic exchange with cooling water
  $\rightarrow \; -k(T)\nabla T \cdot \bar{n} = h(T - T_w)$



(a) Radial (b) Insert

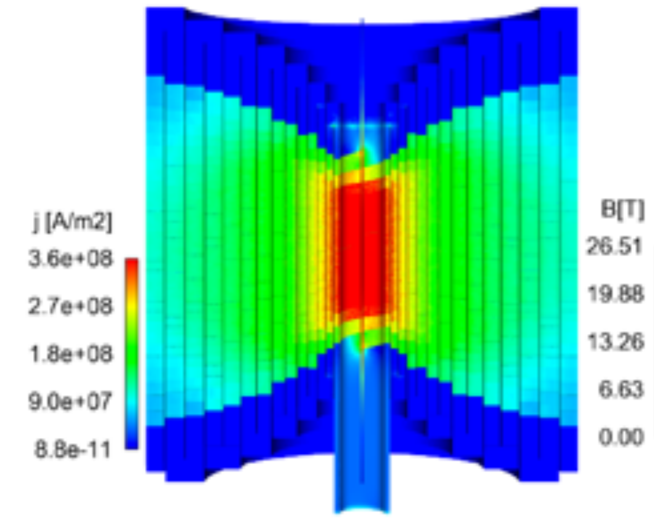
Figure: Non-Linear Temperature

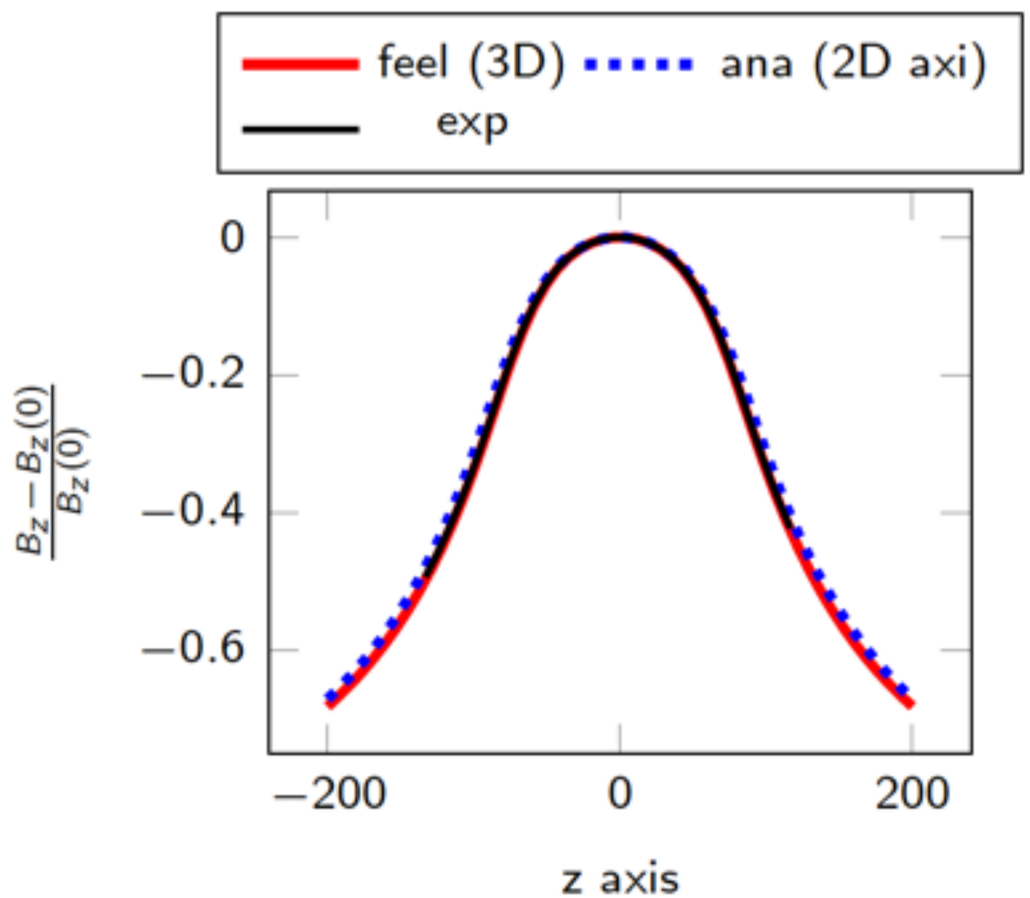
High Field Magnets

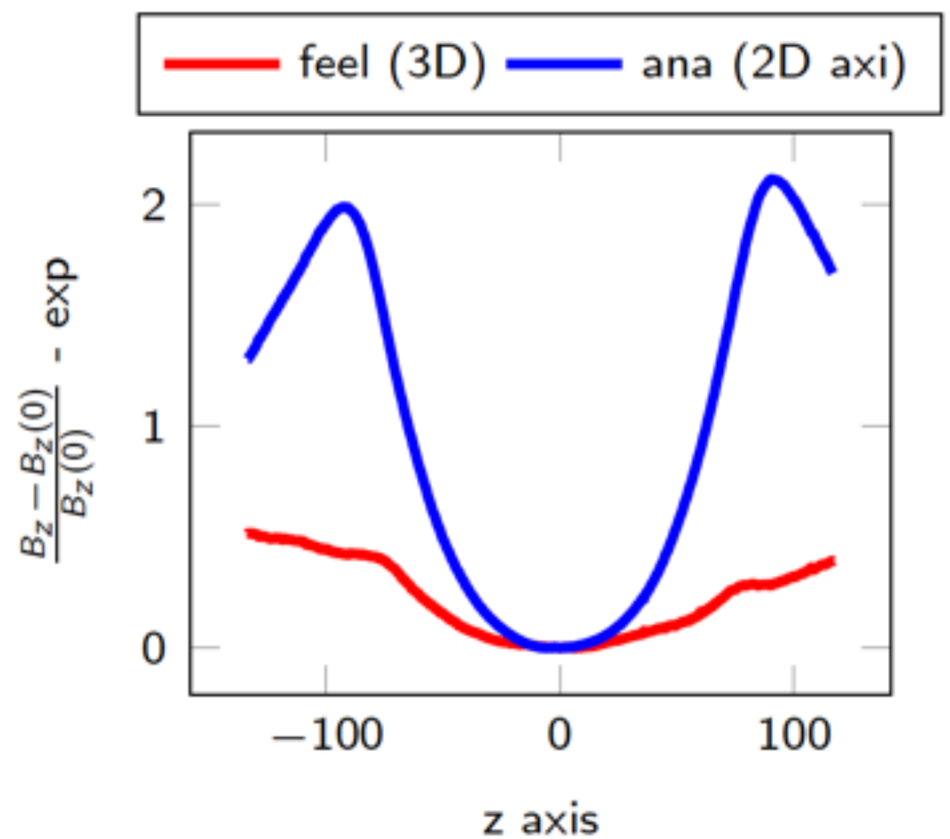
Magnetostatic : Biot & Savart

Magnetic field computation outside the conductor

$$\bar{\boldsymbol{B}}(\boldsymbol{r}) = \frac{\mu_0}{4\pi} \int_{\Omega_{cond}} \frac{\bar{\boldsymbol{j}}(\boldsymbol{r}') \wedge (\boldsymbol{r} - \boldsymbol{r}')}{|\boldsymbol{r} - \boldsymbol{r}'|^3} d\boldsymbol{r}' \text{ with } \boldsymbol{r} \text{ in } \boldsymbol{\Omega}_{\text{mgn}}$$







High Field Magnets

Magnetostatic model

$\bar{j}$: current density

$\bar{H}$: magnetic field intensity

$\bar{B}$: magnetic flux

$$\begin{cases} \nabla \cdot \bar{\boldsymbol{B}} = \mathbf{0} \\ \nabla \wedge \bar{\boldsymbol{H}} = \bar{\boldsymbol{j}} = -\boldsymbol{\sigma} \nabla \boldsymbol{V} \end{cases}$$

$$\Rightarrow \ \exists \bar{A} \ | \ \bar{B} = \nabla \wedge \bar{A}$$

$$\bar{B} = \mu \bar{H}$$

$$\nabla \wedge (\boldsymbol{\mu}^{-1} \nabla \wedge \bar{\boldsymbol{A}}) = \bar{\boldsymbol{j}}$$

Regularized formulation

$$\xrightarrow[\text{Time harmonic equation}]{\text{Fourier transform}} \nabla \wedge (\mu^{-1} \nabla \wedge \bar{A}) + (\sigma i \omega - \epsilon \omega^2) \bar{A} = \bar{j}$$

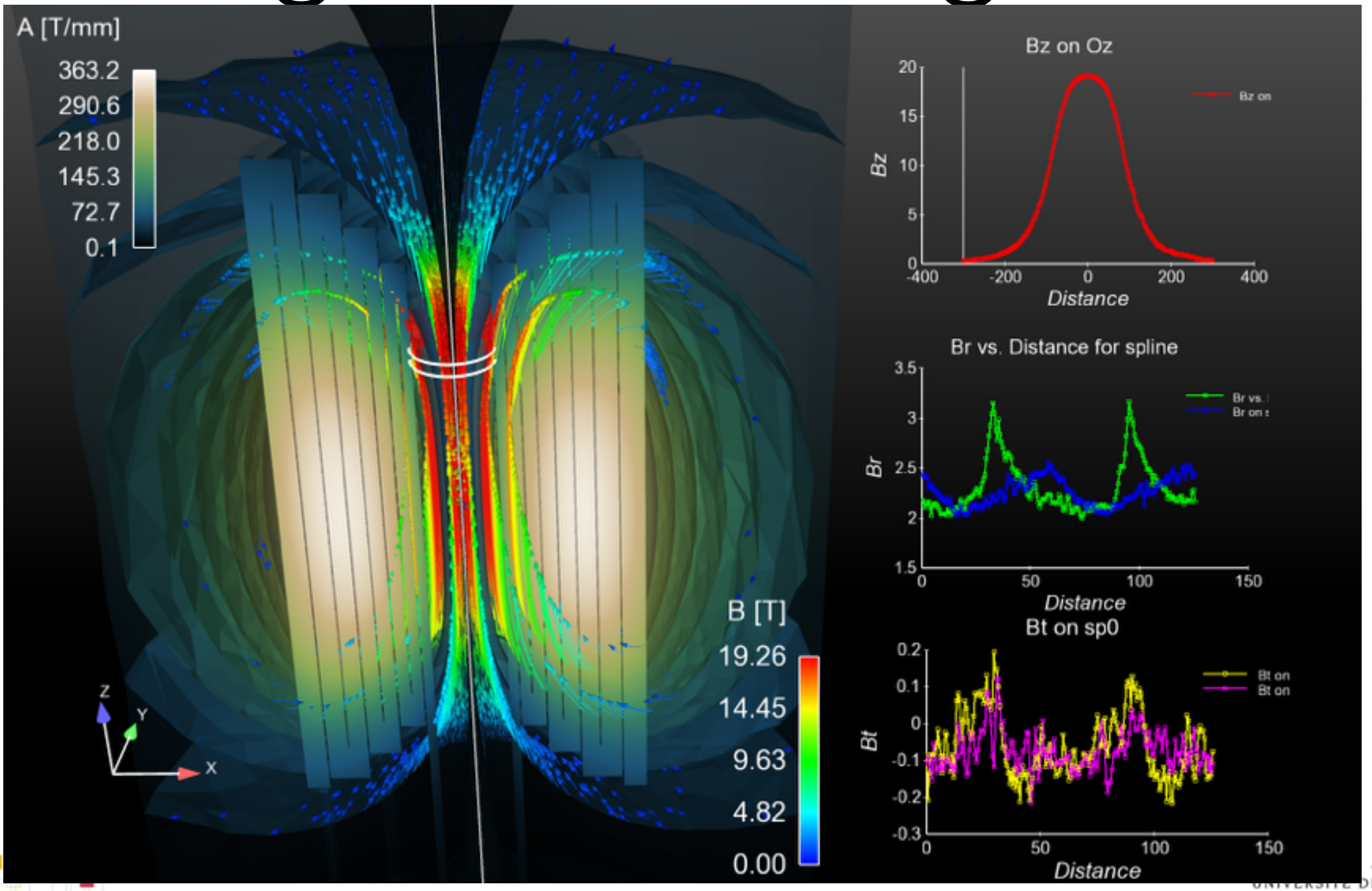
$$\nabla \wedge (\boldsymbol{\mu}^{-1} \nabla \wedge \bar{\boldsymbol{A}}_\alpha) + \alpha \bar{\boldsymbol{A}}_\alpha = \bar{\boldsymbol{j}} \text{ with } \bar{\boldsymbol{A}}_\alpha \xrightarrow[\alpha \to 0]{} \bar{\boldsymbol{A}}$$

Saddle-point formulation

$$\xrightarrow[\text{Lagrange multiplier } p]{\text{Divergence - free condition}} \begin{cases} \nabla \wedge (\mu^{-1} \nabla \wedge \bar{A}) + \nabla p = \bar{j} \\ \nabla \cdot \bar{A} = 0 \end{cases}$$

$$\nabla \wedge (\boldsymbol{\mu}^{-1} \nabla \wedge \bar{\boldsymbol{A}}) + \nabla \boldsymbol{p} = \bar{\boldsymbol{j}} \text{ with } \nabla \cdot \bar{\boldsymbol{A}} = \mathbf{0} \text{ and } \boldsymbol{p} = \mathbf{0} \text{ on boundary}$$

High Field Magnets



High Field Magnets

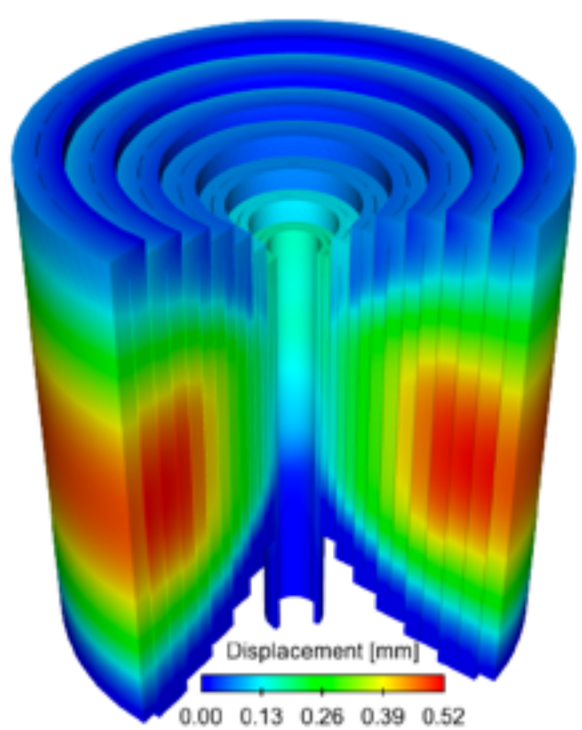
Elasticity model

$$-\underbrace{\nabla\cdot\bar{\bar{\sigma}}(\bar{\bar{\varepsilon}})}_{\text{includes thermal dilatation}} = \bar{f} \quad \text{with} \quad \bar{f} = \underbrace{\bar{j}\times\bar{B}}_{\text{Lorentz forces}} \quad \text{and} \quad \bar{\bar{\varepsilon}} = \frac{1}{2}(\nabla\bar{u}+\nabla\bar{u}^{\top})$$

$$\bar{\bar{\sigma}}(\bar{\bar{\varepsilon}}) = \frac{E}{1+\nu}\left(\bar{\bar{\varepsilon}} + \frac{\nu}{1-2\nu}Tr(\bar{\bar{\varepsilon}})I\right) - \frac{E}{1-2\nu}\delta(T-T_0)I$$

Displacement condition : $\bar{u} = \bar{u_D}$

Pressure condition : $\bar{\bar{\sigma}}\cdot\bar{n} = \bar{p}$

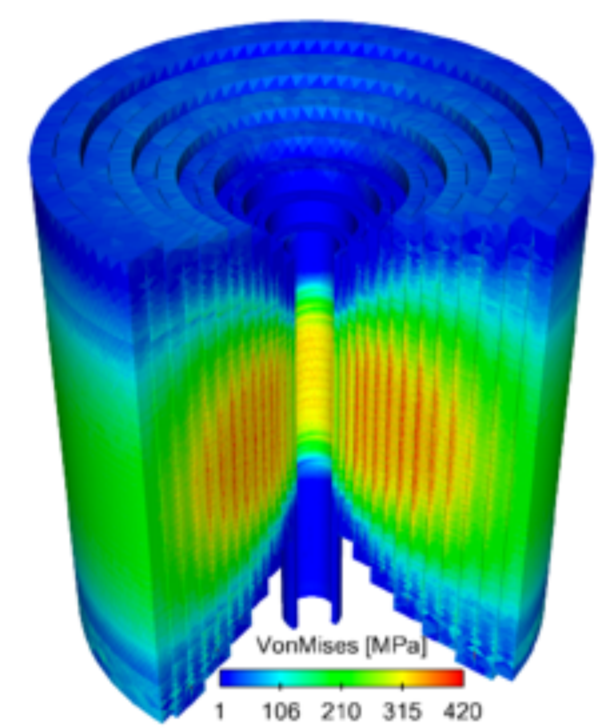


Yield strength - Tresca

$$tr = \max_{1\leqslant i<j\leqslant Dim}(|\ \bar{\bar{\sigma}}^d_{ii} - \bar{\bar{\sigma}}^d_{jj}\ |)$$

Yield strength - Von-Mises

$$vm = \sqrt{\sum_{1\leqslant i<j\leqslant Dim}\frac{1}{2}\left(\bar{\bar{\sigma}}^d_{ii} - \bar{\bar{\sigma}}^d_{jj}\right)^2}$$



ROM for High Field Magnets

A benchmark [grep et al, 2007]

$$\mu \in [0.01; 100]^2$$

$$\Omega = [0,1]^2$$

$$-\Delta u + \mu_1 \frac{e^{\mu_2 u} - 1}{\mu_2} = 100 sin(2\pi x) sin(2\pi y)$$

SER(1)

hal-01332437v1

N	M	$\max(\epsilon^{u,M}_{M,N})$	$\max(\epsilon^{s,M}_{M,N})$
4	5	7.38e-3	5.75e-3
8	10	1.01e-3	2.34e-4
12	15	1.49e-4	3.09e-5
16	20	2.21e-5	1.25e-5
20	25	5.88e-6	2.82e-6

(a) $r = M$

N	M	$\max(\epsilon^{u,5}_{M,N})$	$\max(\epsilon^{s,5}_{M,N})$
4	5	8.21e-3	6.31e-3
8	10	4.48e-3	6.18e-3
12	15	2.69e-4	2.36e-4
16	20	1.48e-4	9.31e-5
20	25	2.60e-5	1.46e-5

(b) $r = 5$

N	M	$\max(\epsilon^{u,1}_{M,N})$	$\max(\epsilon^{s,1}_{M,N})$
5	5	9.98e-3	7.77e-3
10	10	2.32e-3	1.86e-3
15	15	4.61e-4	3.75e-4
20	20	2.48e-4	2.02e-4
25	25	3.51e-5	2.33e-5

(c) $r = 1$ (W_N recomputed)

N	M	$\max(\epsilon^{u,1}_{M,N})$	$\max(\epsilon^{s,1}_{M,N})$
5	5	1.30e-2	1.02e-2
10	10	2.20e-3	1.50e-3
15	15	4.83e-4	4.05e-4
20	20	2.42e-4	1.98e-4
25	25	1.50e-5	1.24e-5

(d) $r = 1$ (W_N not recomputed)

ROM for High Field Magnets

A benchmark [grep et al, 2007]

$\mu \in [0.01; 100]^2$

$\Omega = [0,1]^2$

$$-\Delta u + \mu_1 \frac{e^{\mu_2 u} - 1}{\mu_2} = 100 sin(2\pi x) sin(2\pi y)$$

SER(l)

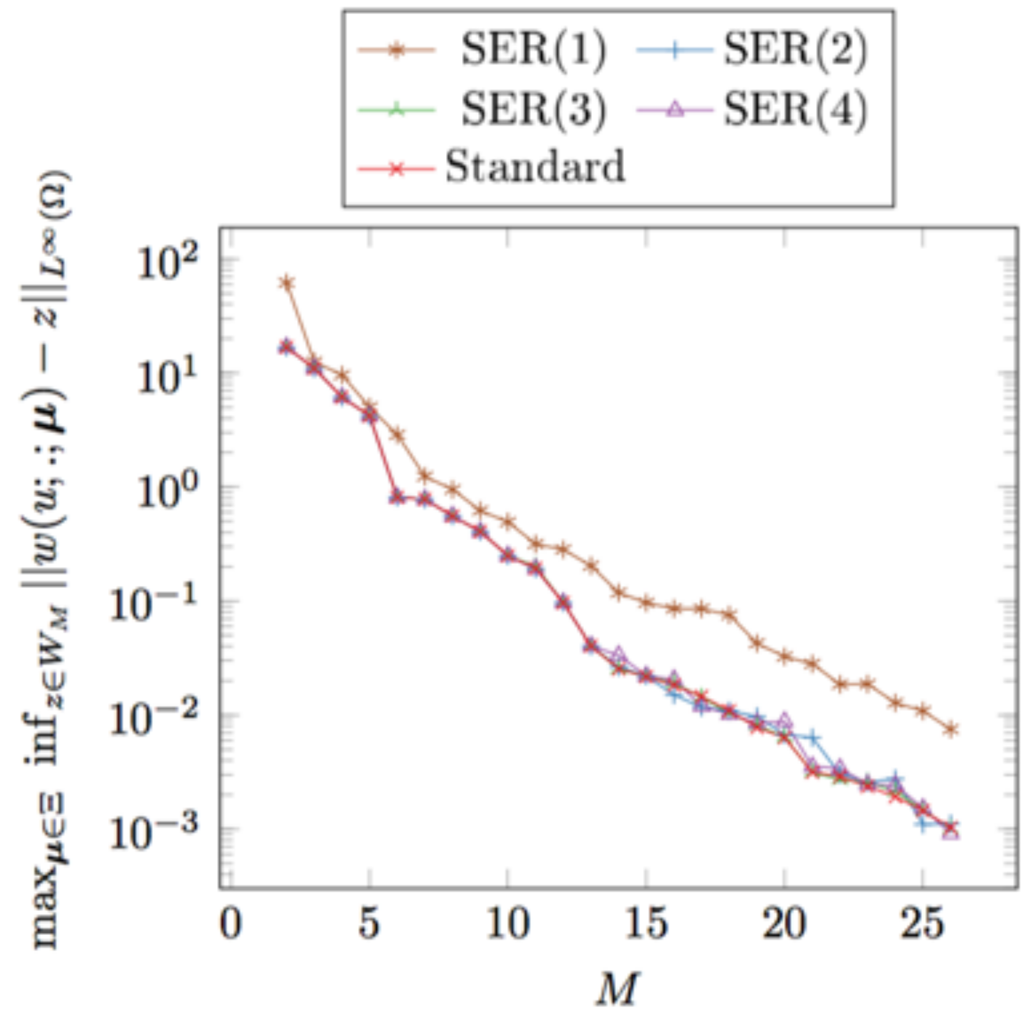
hal-01332437v1

N	M	$\max(\epsilon^u_{M,N})$	$\max(\epsilon^s_{M,N})$
5	5	9.17e-3	6.99e-3
10	10	2.89e-4	2.07e-4
15	15	4.12e-5	1.87e-5
20	20	1.44e-5	7.61e-6
25	25	2.72e-5	2.20e-5

(b) SER(2)

N	M	$\max(\epsilon^u_{M,N})$	$\max(\epsilon^s_{M,N})$
5	5	7.93e-3	6.01e-3
10	10	2.99e-4	1.80e-4
15	15	1.75e-4	1.35e-4
20	20	1.69e-5	6.02e-6
25	25	7.86e-6	5.37e-6

(c) SER(3)



ROM for High Field Magnets

SER method for high field magnet modeling

Equations

$$\begin{cases} -\nabla \cdot (\sigma(T)\nabla V) = 0 \text{ in } \Omega \\ -\nabla \cdot (k(T)\nabla T) = \sigma(T)\nabla V \cdot \nabla V \text{ in } \Omega \end{cases}$$

EIM approximations

$$\sigma_M \approx \sigma(T) = \frac{\sigma_0}{1 + \alpha(T - T_0)}$$

$$k_M \approx k(T) = LT\sigma(T)$$

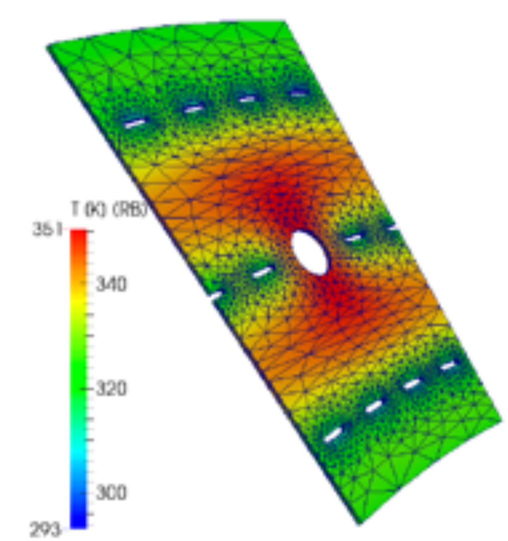
$$Q_M \approx \sigma(T)\nabla V \cdot \nabla V$$

Output : Mean temperature

$$\frac{1}{|\Omega|}\int_\Omega T$$

Inputs

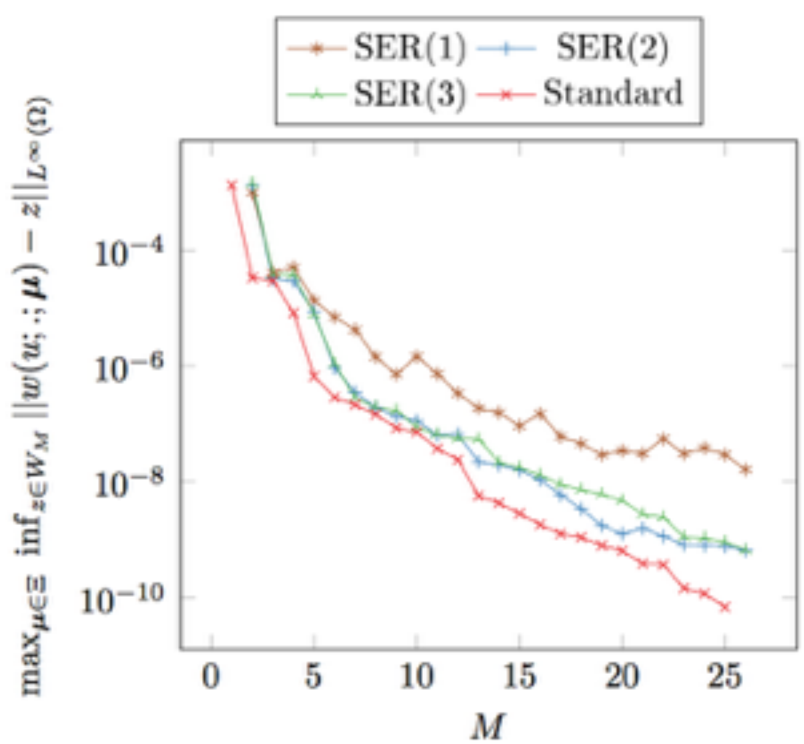
- σ_0 : elec. conductivity
- α : temperature coeff
- L : Lorentz number
- j : current density
- h : Heat transfer coeff
- T_w : Water temperature



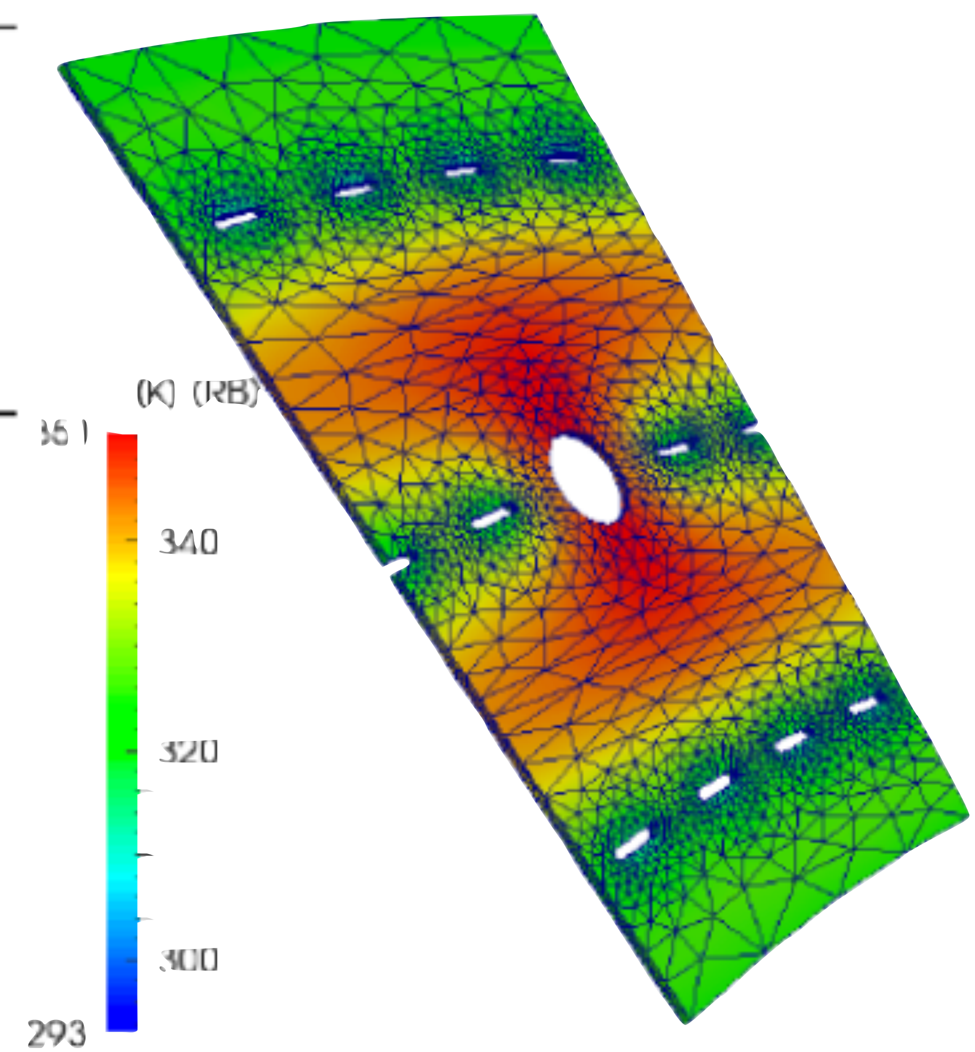
ROM for High Field Magnets

N	M	$max(\epsilon^u_{M,N})$	$max(\epsilon^s_{M,N})$	$mean(\epsilon^u_{M,N})$	$mean(\epsilon^s_{M,N})$
5	5	7.66e+0	2.71e-2	1.25e+0	7.88e-3
10	10	3.37e-1	7.82e-4	1.49e-1	1.97e-4
15	15	4.85e-2	2.64e-4	1.75e-2	5.02e-5
20	20	2.93e-2	3.40e-4	3.48e-3	2.68e-5
25	25	5.23e-3	4.59e-5	1.21e-3	3.47e-6

Table 11.4 – SER - Greedy - EIM trainset size = 100



(b) EIM $k(T)$

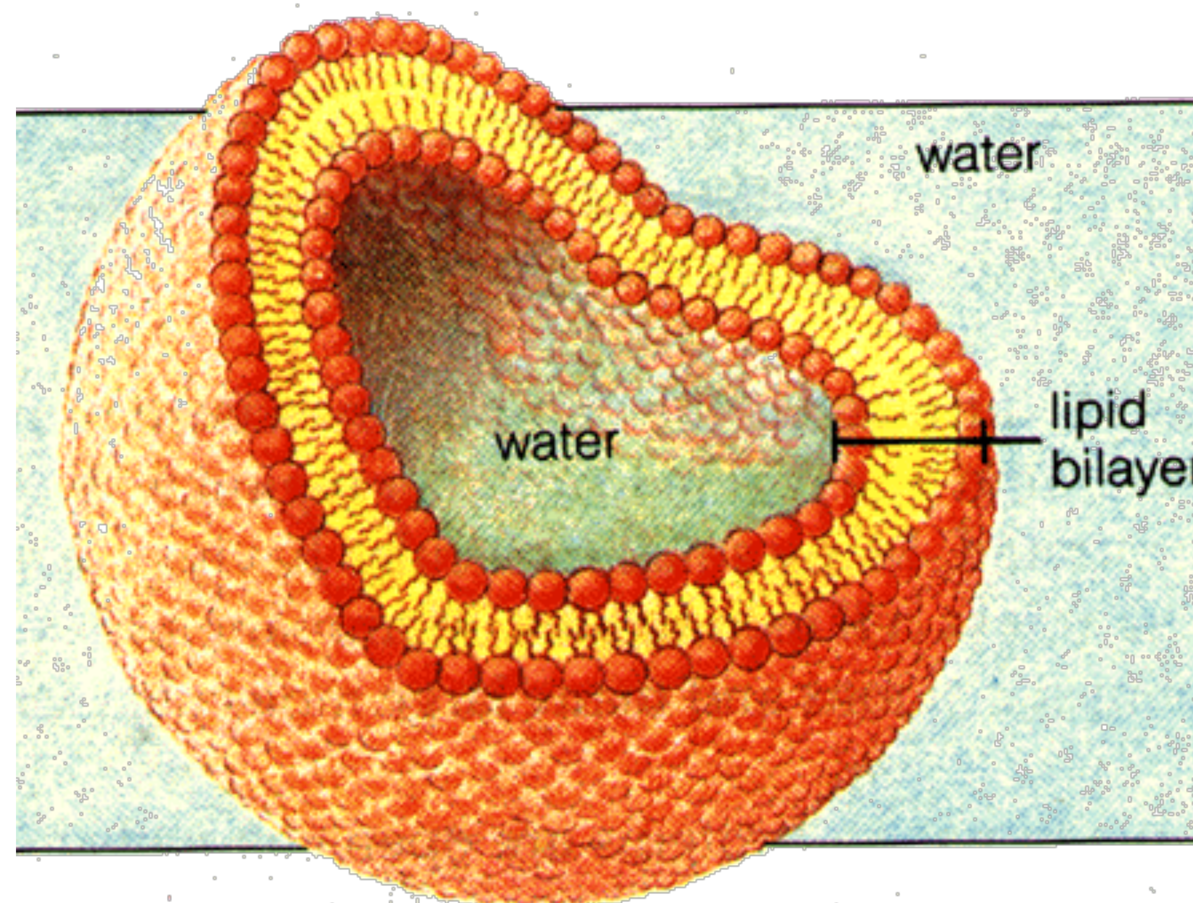


DEMO 4FASTSIM

http://www.cemosis.fr/projects/4fastsim/

Bio-Medical

A Physical Model for RBC



Vesicle

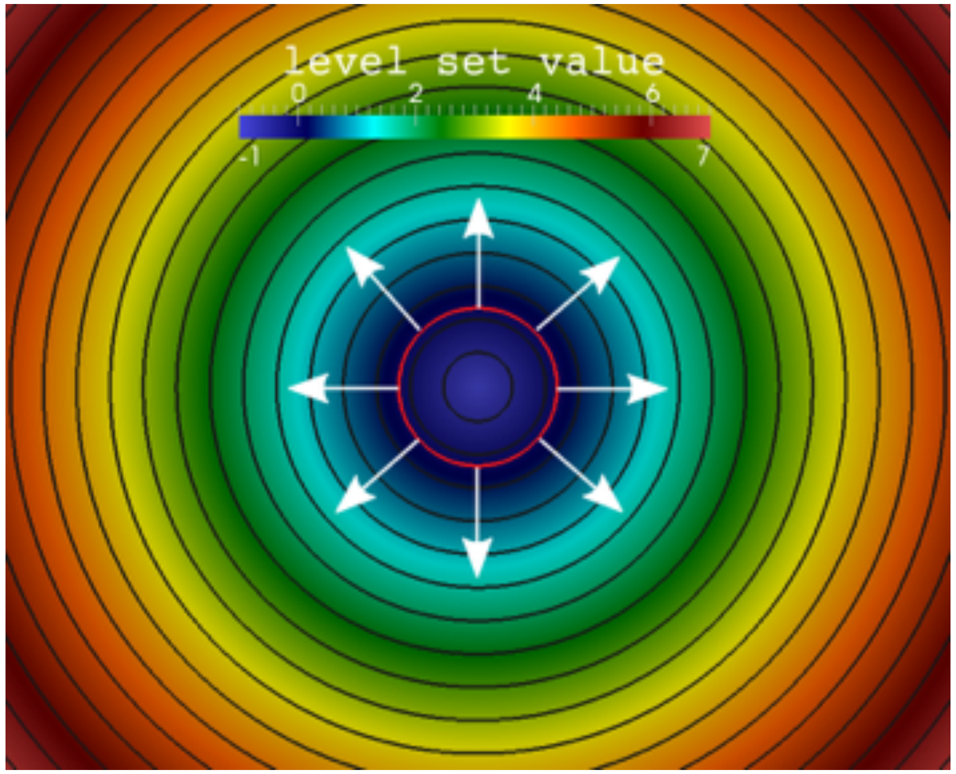
Two fluids separated by a membrane → mimics some behaviour of red blood cells (passive mechanical properties)

Membrane

- Non-porous: conservation of the volume of inner fluid
- Non-extensible: conservation of the surface
- Bending energy (Helfrich energy $E_h = \int_\Gamma \frac{k_B}{2} \kappa^2 \rightarrow$ force

A RBC Numerical Model

A function $\phi(x)$ to track the interface



$$\phi(\vec{x}) = \begin{cases} \mathrm{dist}(\vec{x}, \Gamma) & \vec{x} \in \Omega_1, \\ 0 & \vec{x} \in \Gamma, \\ -\mathrm{dist}(\vec{x}, \Gamma) & \vec{x} \in \Omega_2, \end{cases}$$

$$\mathbf{n} = \frac{\nabla \phi}{|\nabla \phi|}, \qquad \kappa = \nabla \cdot \mathbf{n}$$

An equation for the distance to the membrane

$\phi(\vec{x}) = 0$ gives the interface $\Rightarrow \frac{D\phi}{Dt} = 0$
i.e. *advection* by a divergence-free velocity $\mathbf{u}$

$$\partial_t \phi + \mathbf{u} \cdot \nabla \phi = 0$$

Incompressible Navier-Stokes Equations

$$\rho\left(\partial_t \mathbf{u} + (\mathbf{u}\cdot\nabla)\,\mathbf{u}\right) - \mu\,\mathbf{\Delta}\mathbf{u} + \nabla p = \mathbf{F}$$
$$\nabla\cdot\mathbf{u} = 0$$

Interface dependent fluid parameters

$$\rho_\phi = \rho^- + (\rho^+ - \rho^-)H_\epsilon(\phi)$$
$$\mu_\phi = \mu^- + (\mu^+ - \rho^-)H_\epsilon(\phi)$$

Surface tension, gravity, bending force, ...

$\rightarrow$ also interface-dependent

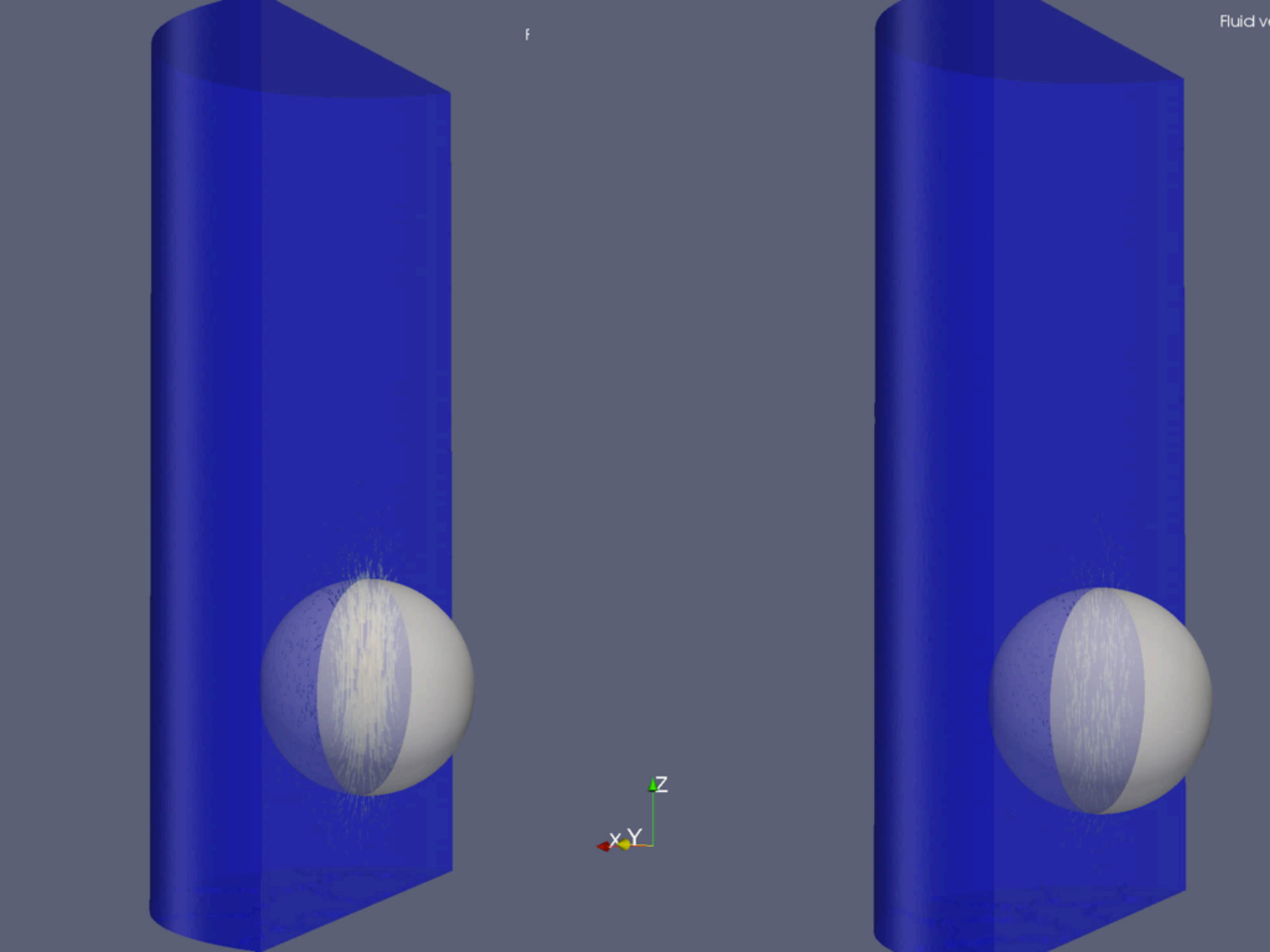
Solution Strategy

Coupling with level-set advection

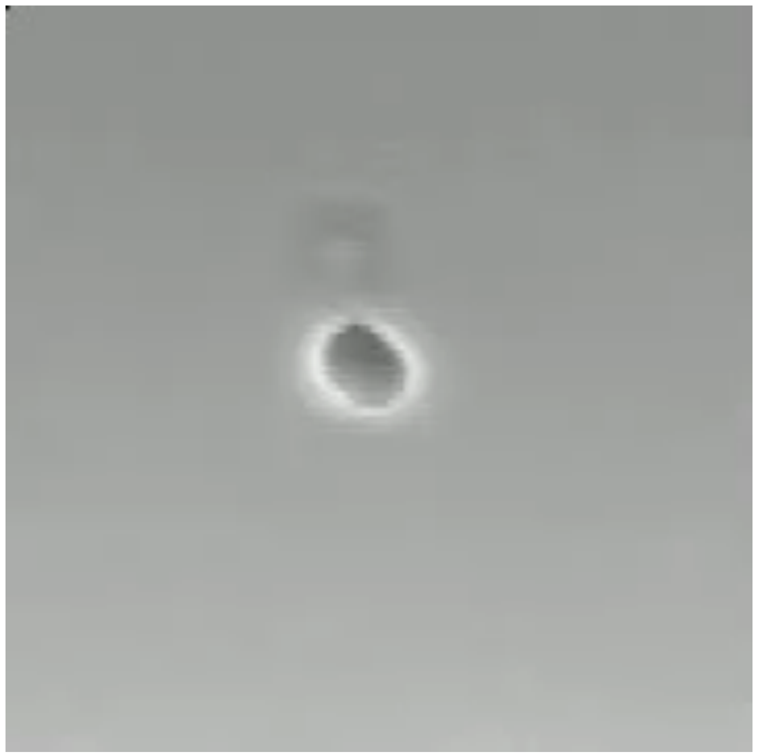
Non-monolithic approach

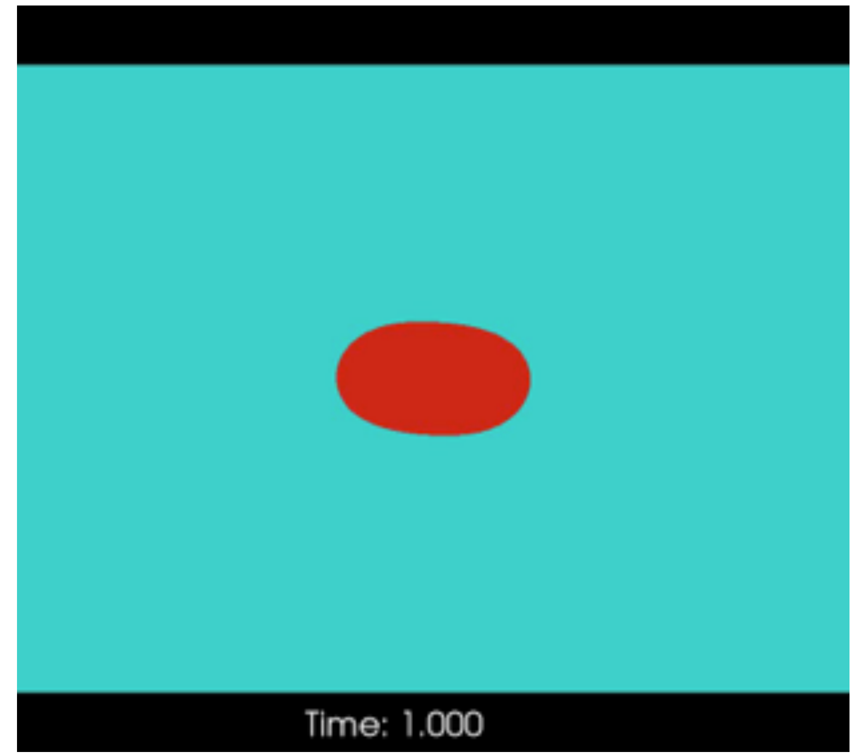
- Solve Navier-Stokes equations $\rightarrow \mathbf{u}^{(n+1)}, p^{(n+1)}$
- Advect level-set with $\mathbf{u}^{(n+1)} \rightarrow \phi^{(n+1)}$
- Update fluid parameters and forces $\rightarrow \rho_\phi^{(n+1)},\ \mu_\phi^{(n+1)},\ \mathbf{F}_\phi^{(n+1)}$
- Modular $\rightarrow$ easier development
- Allows use of optimized dedicated solvers
- Requires smaller time-steps

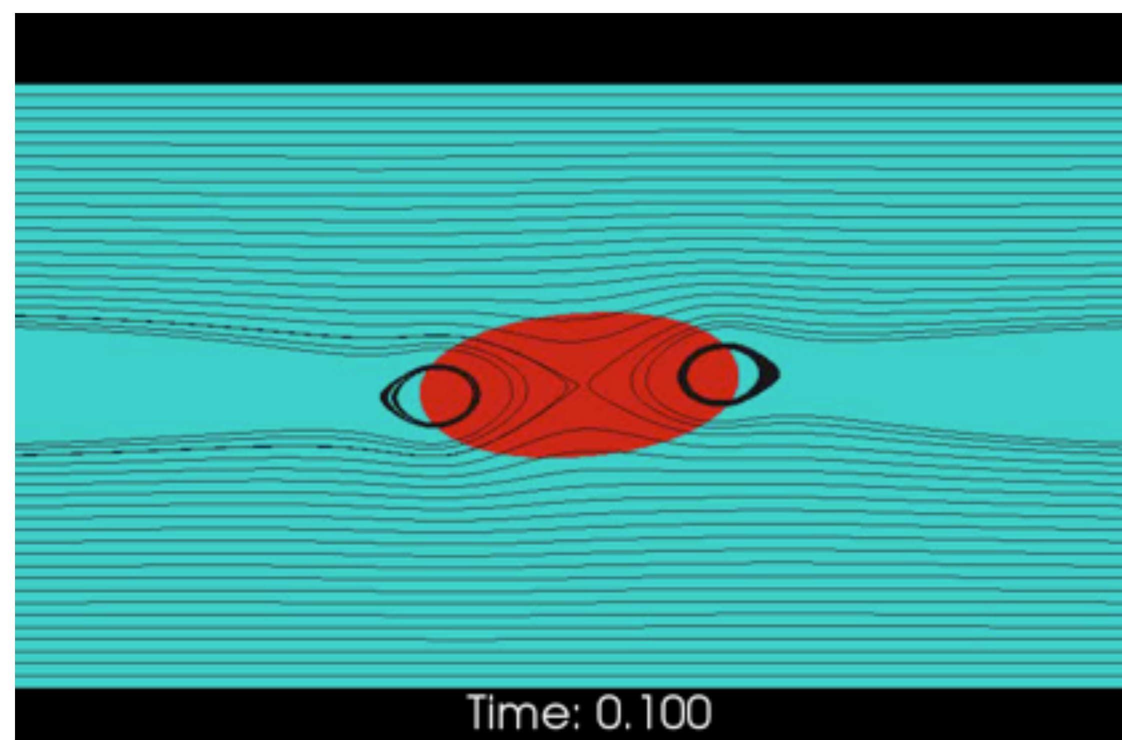
$\Rightarrow$ solved with finite-element method, using FEEL++ library


Z
X
Y

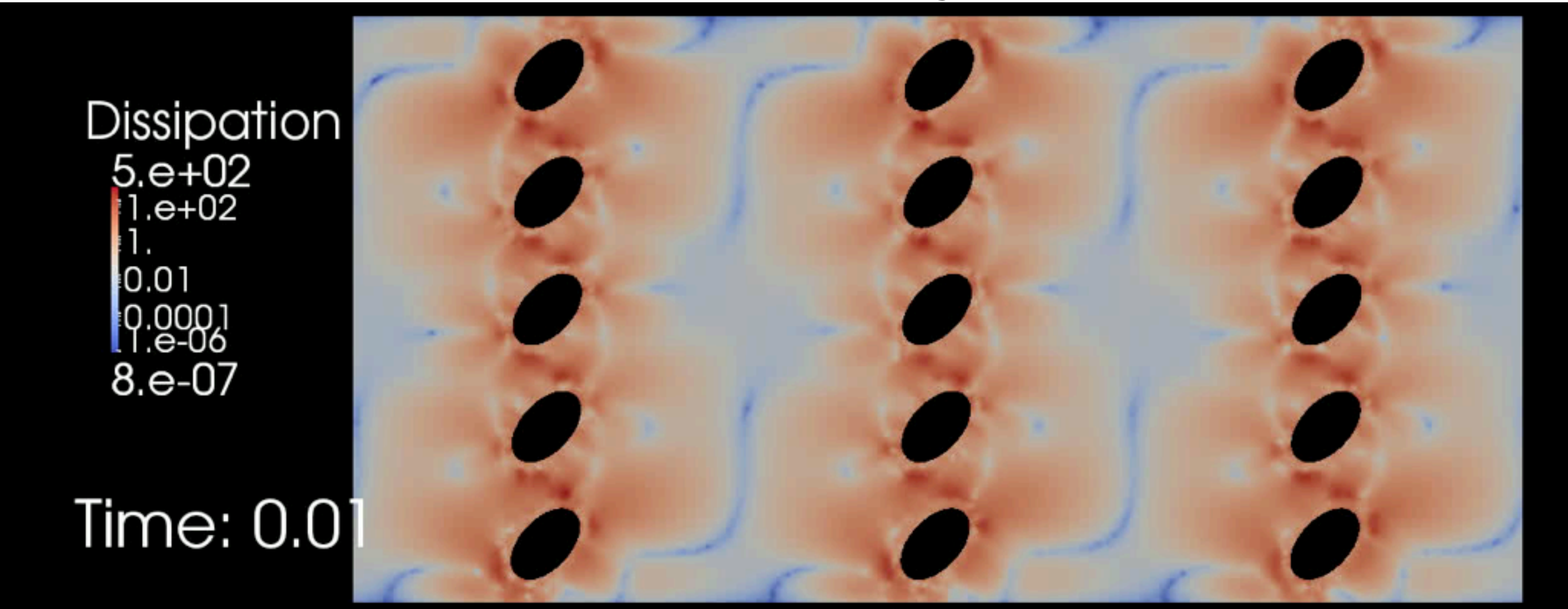
Validation

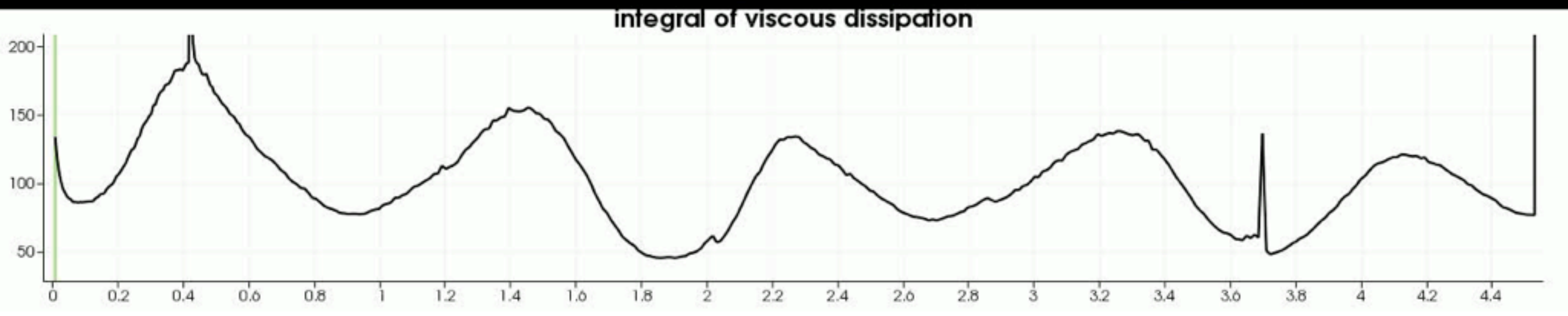




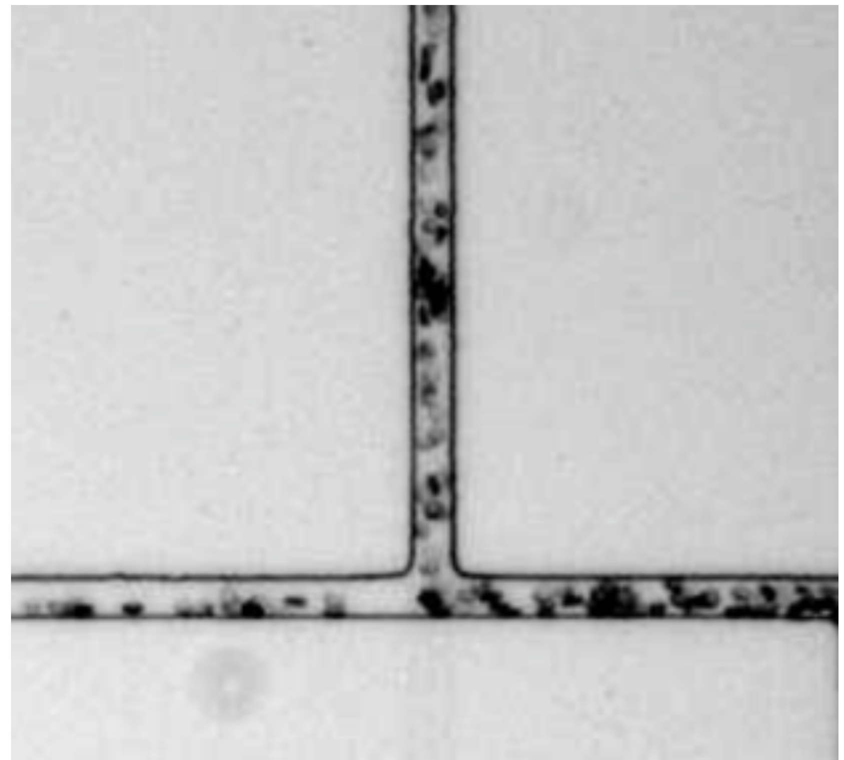


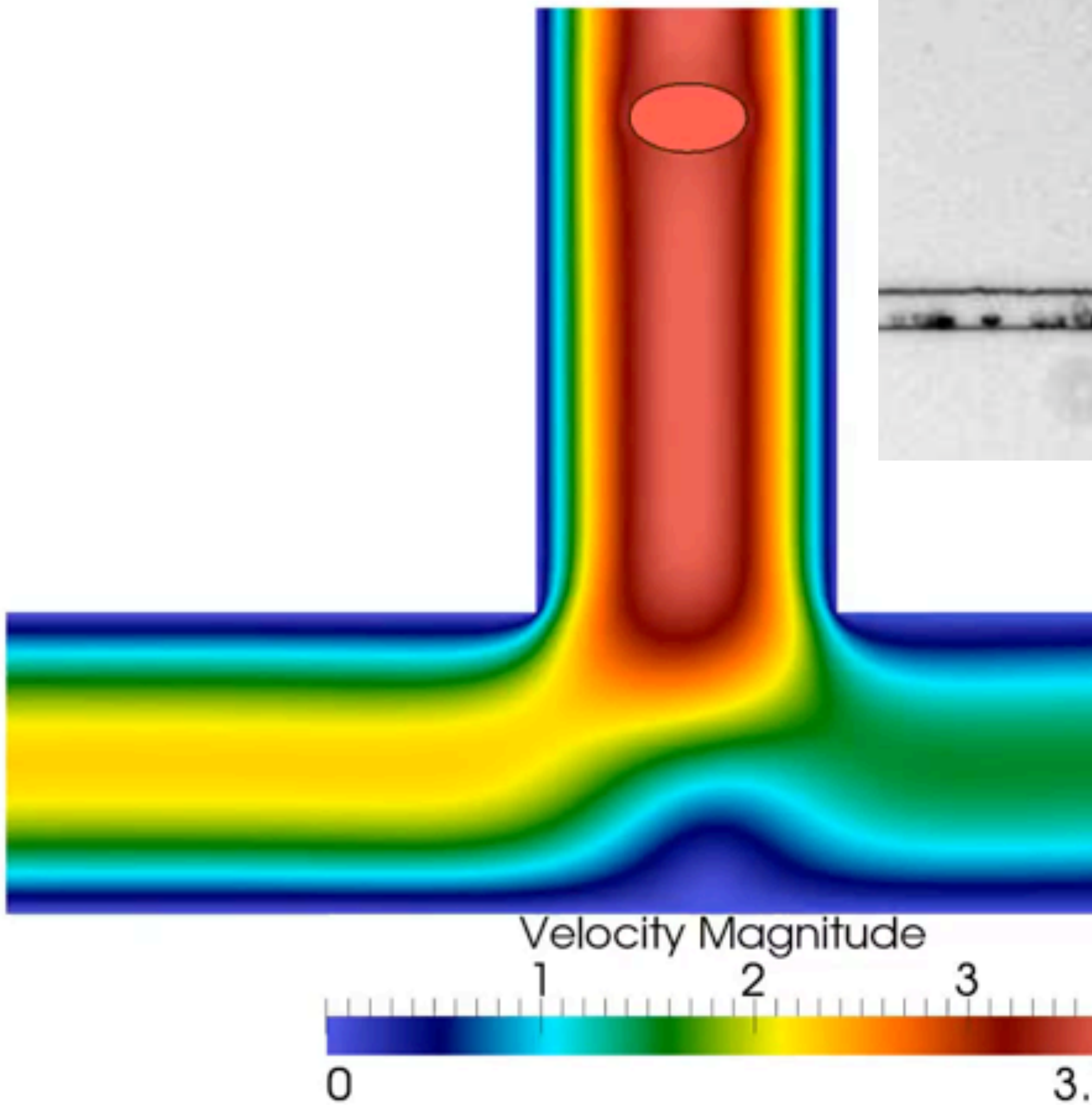
Suspensions and effective viscosity

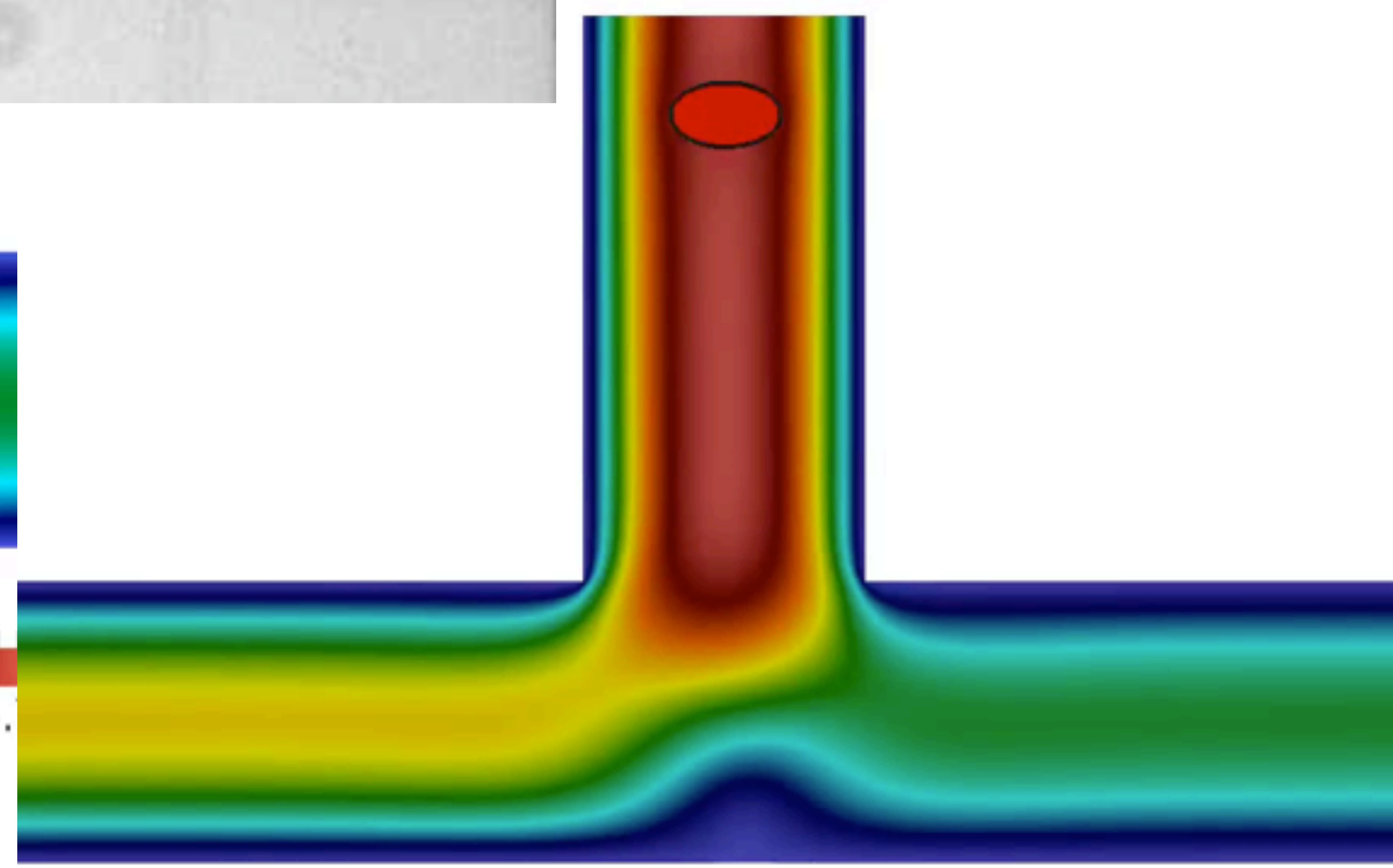




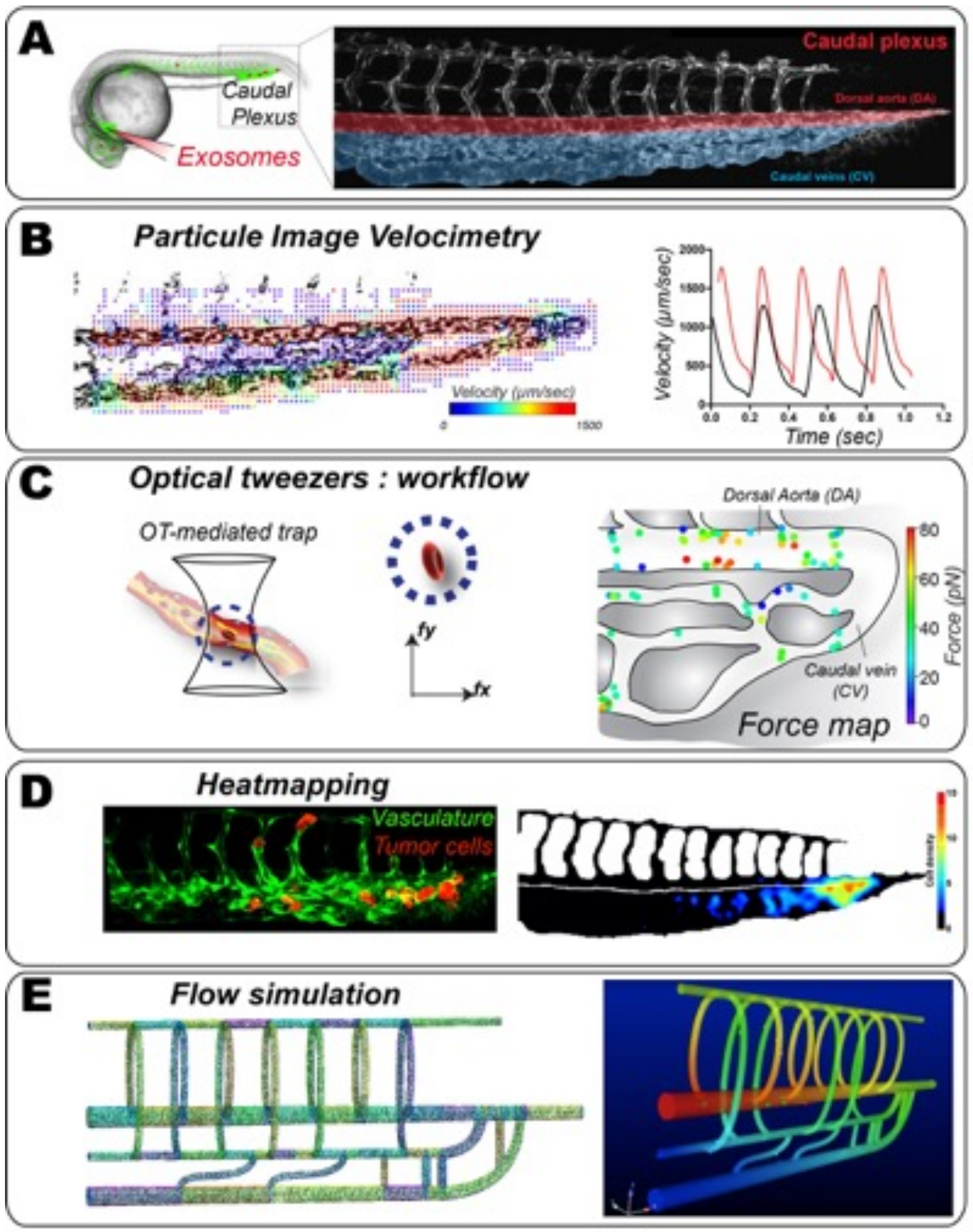
Suspensions @ Bifurcations

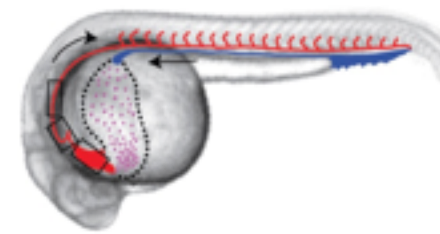


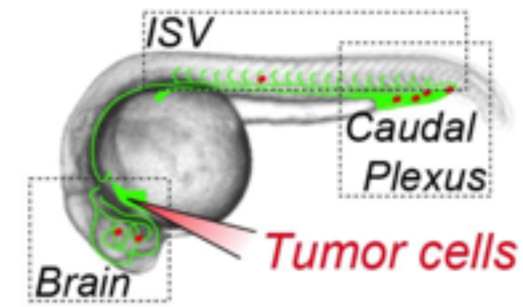


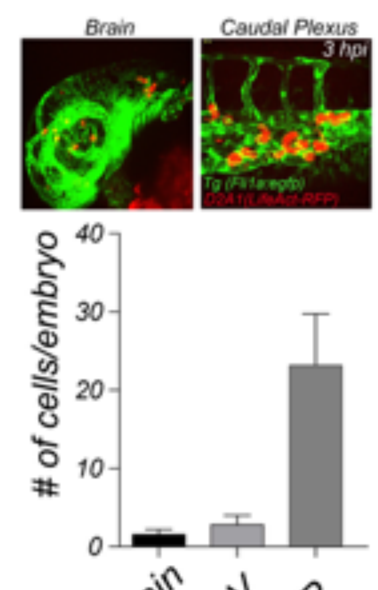


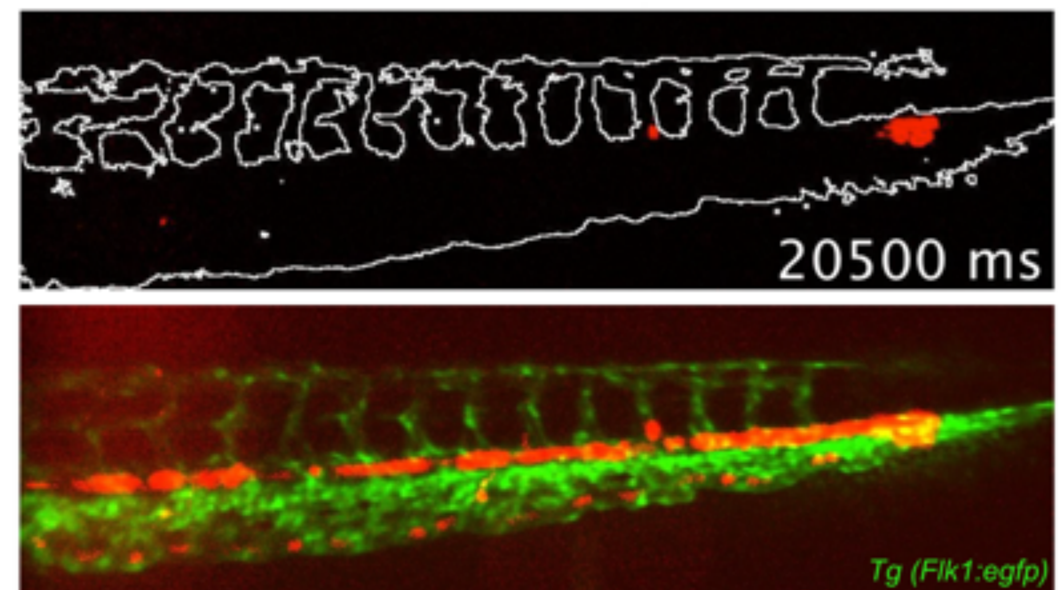
Hemotum++
Influence of hemodynamics on attachment sites of tumor cells in the vascular system [J. Goetz(inserm), S. Harlepp(ipcms)]

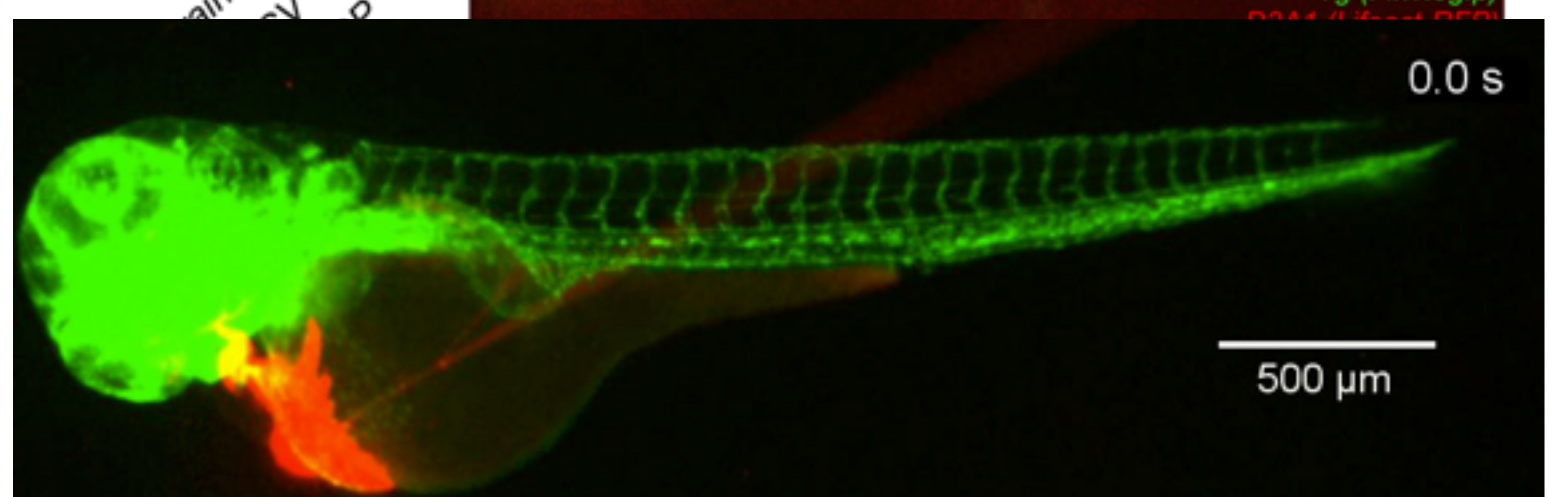






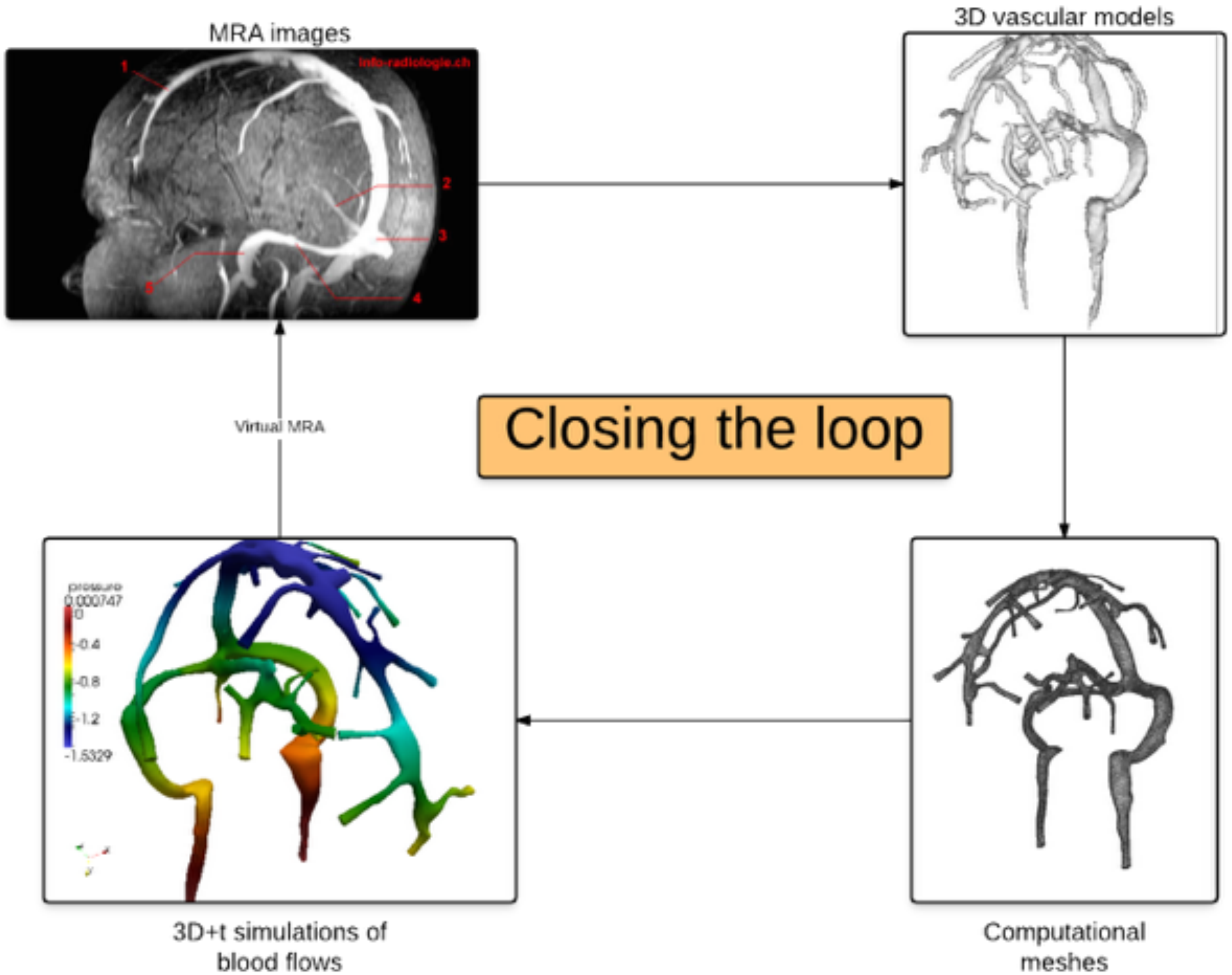






Cerebral Blood Flow

VivaBrain : A Multi-Disciplinary project



Physics

Medical imaging : MRI, MRA

Computer Science

Image processing

Model generation

Mathematics

Numerical analysis

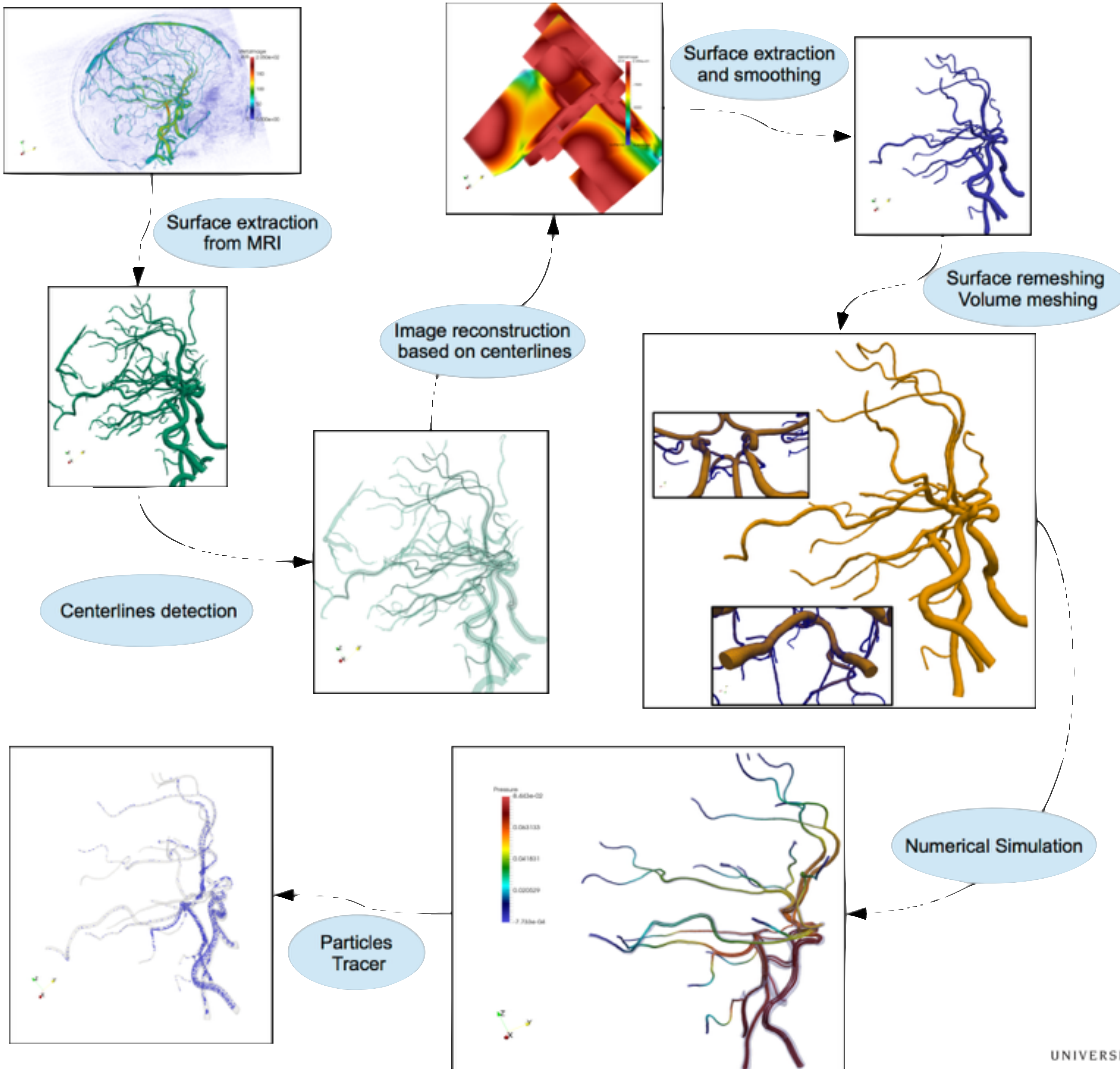
Uncertainty quantification

High performance computing

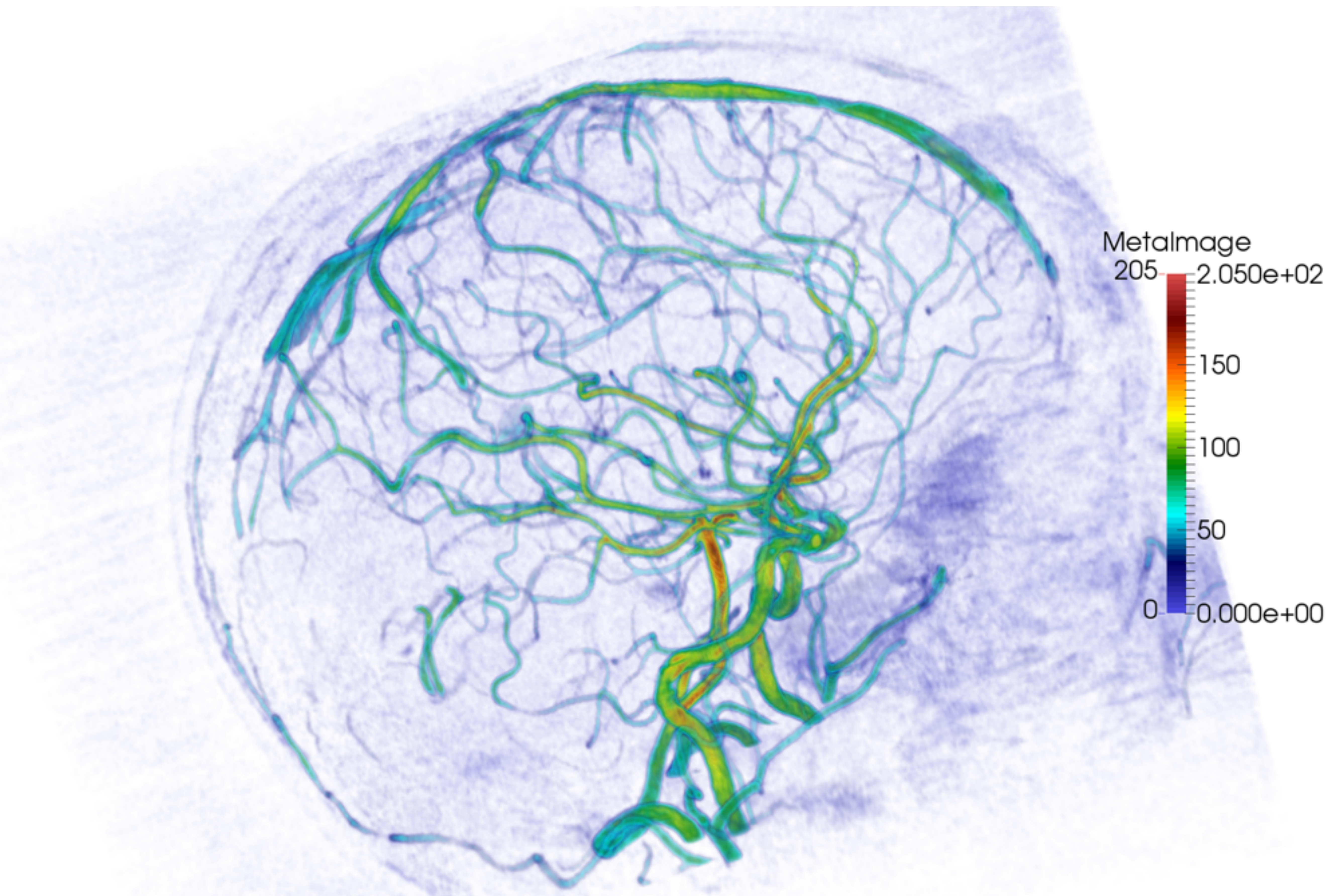
Medicine

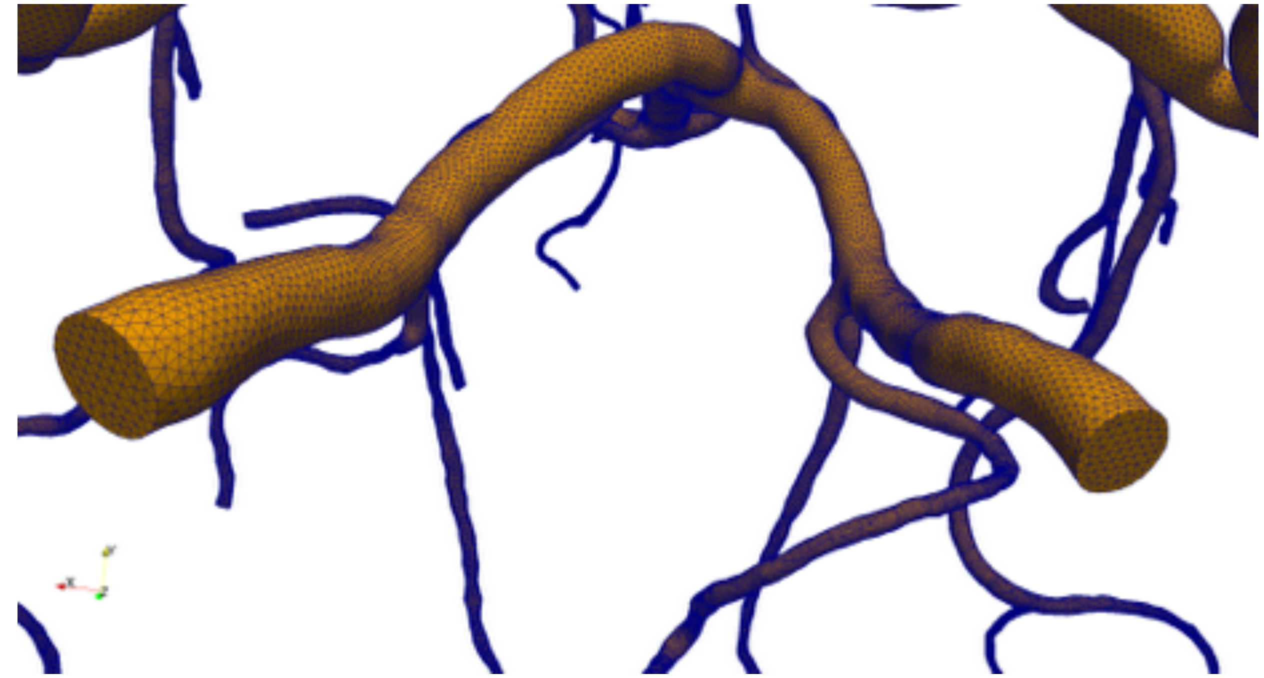
Vascular anatomy

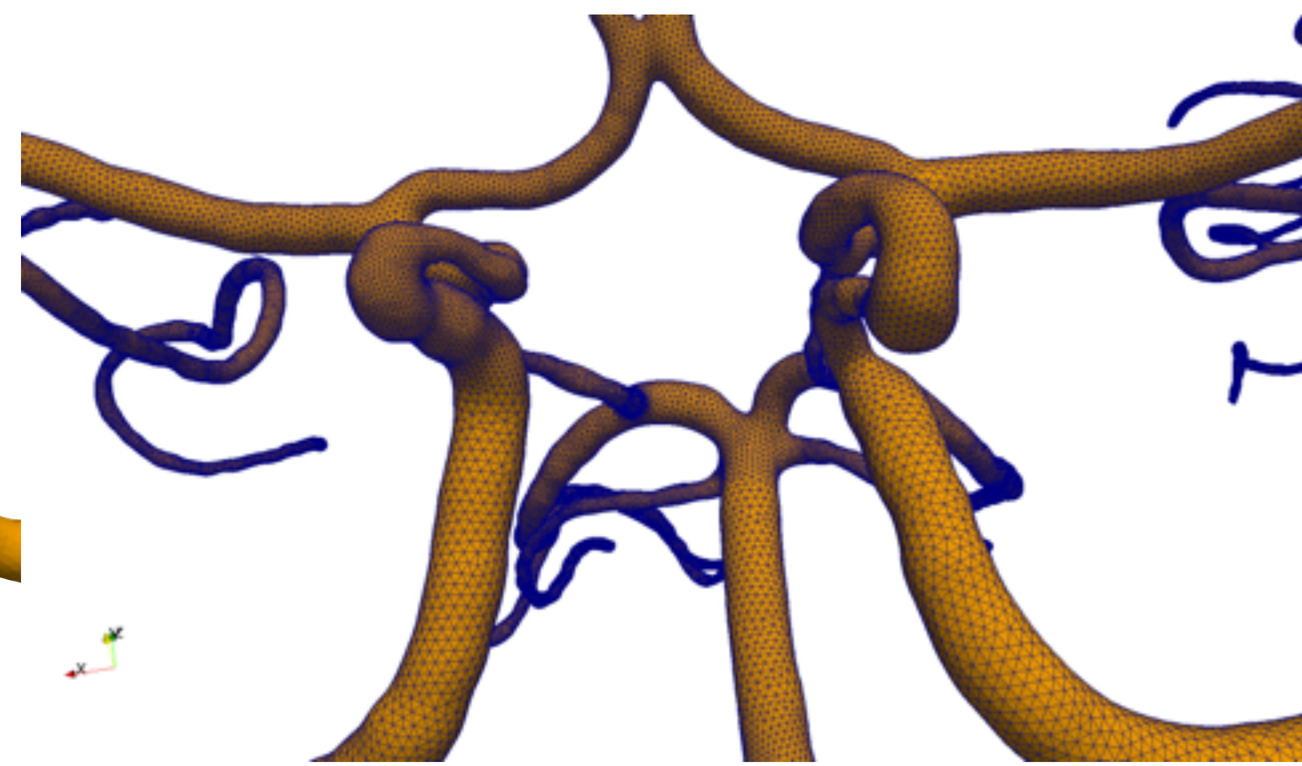
Haemodynamics

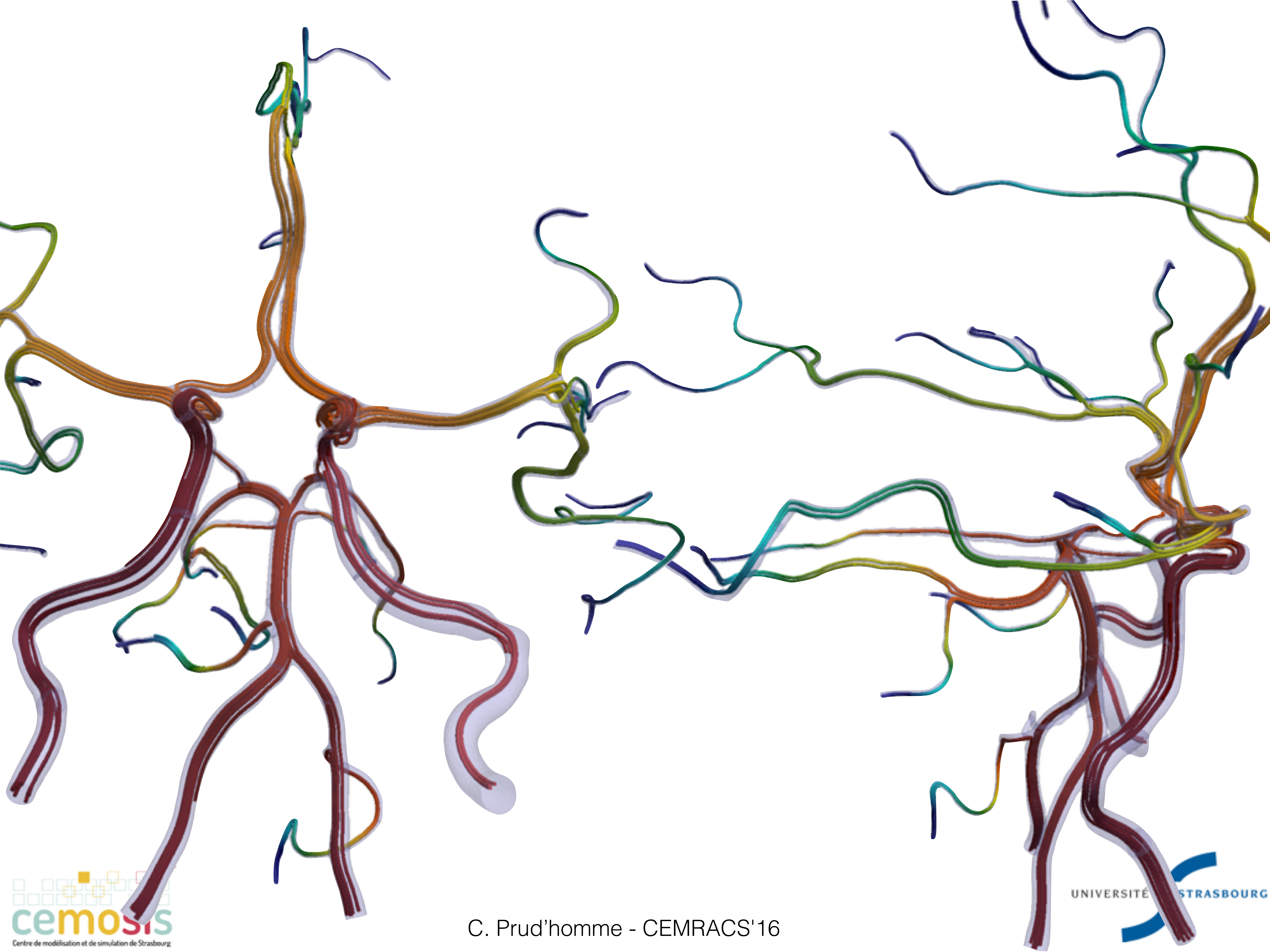


Surface extraction and smoothing
Surface extraction from MRI
Surface remeshing Volume meshing
Image reconstruction based on centerlines
Centerlines detection
Numerical Simulation
Particles Tracer


MetaImage
205
2.050e+02
150
100
50
0
0.000e+00

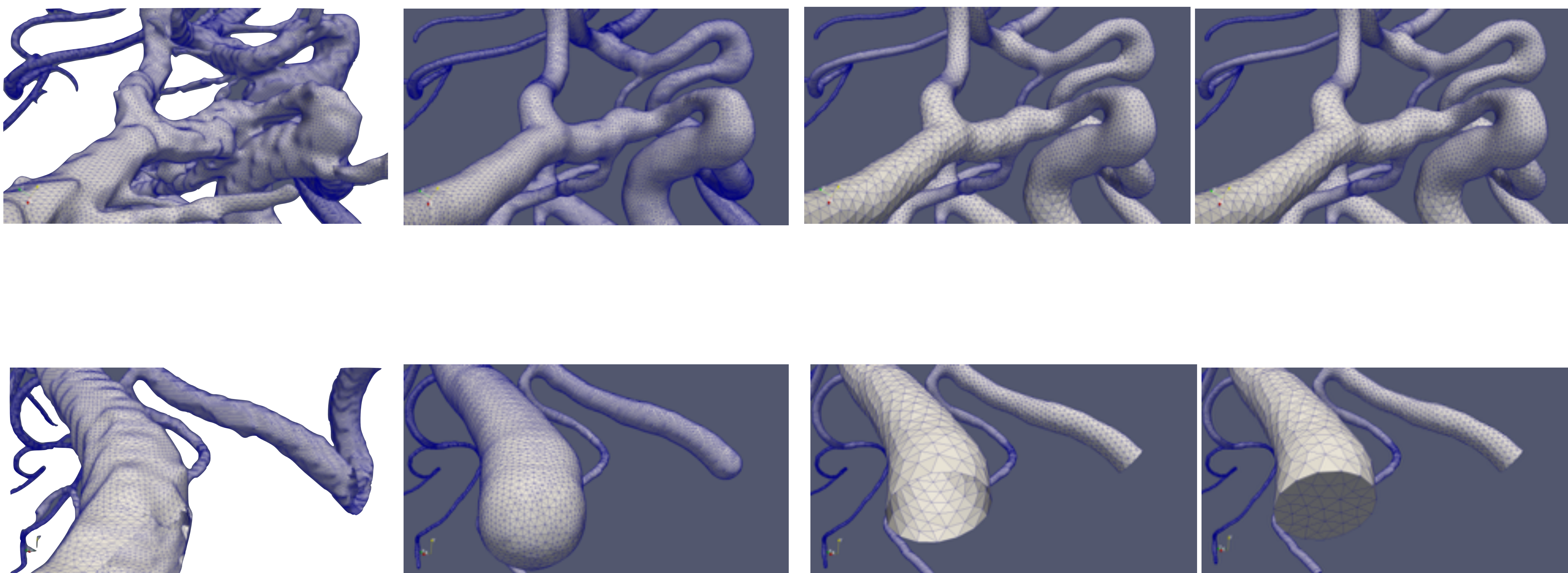






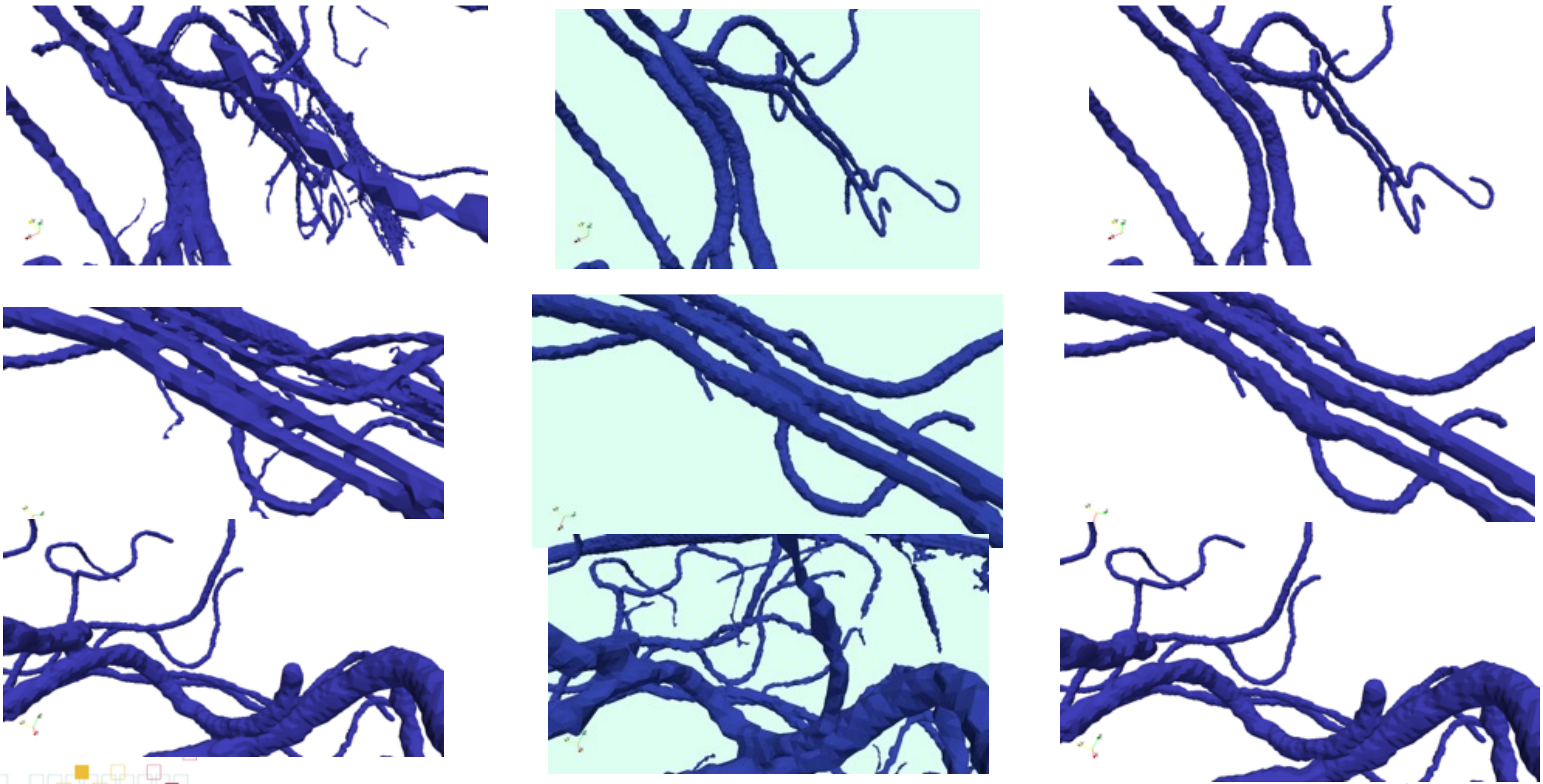
Some Challenges

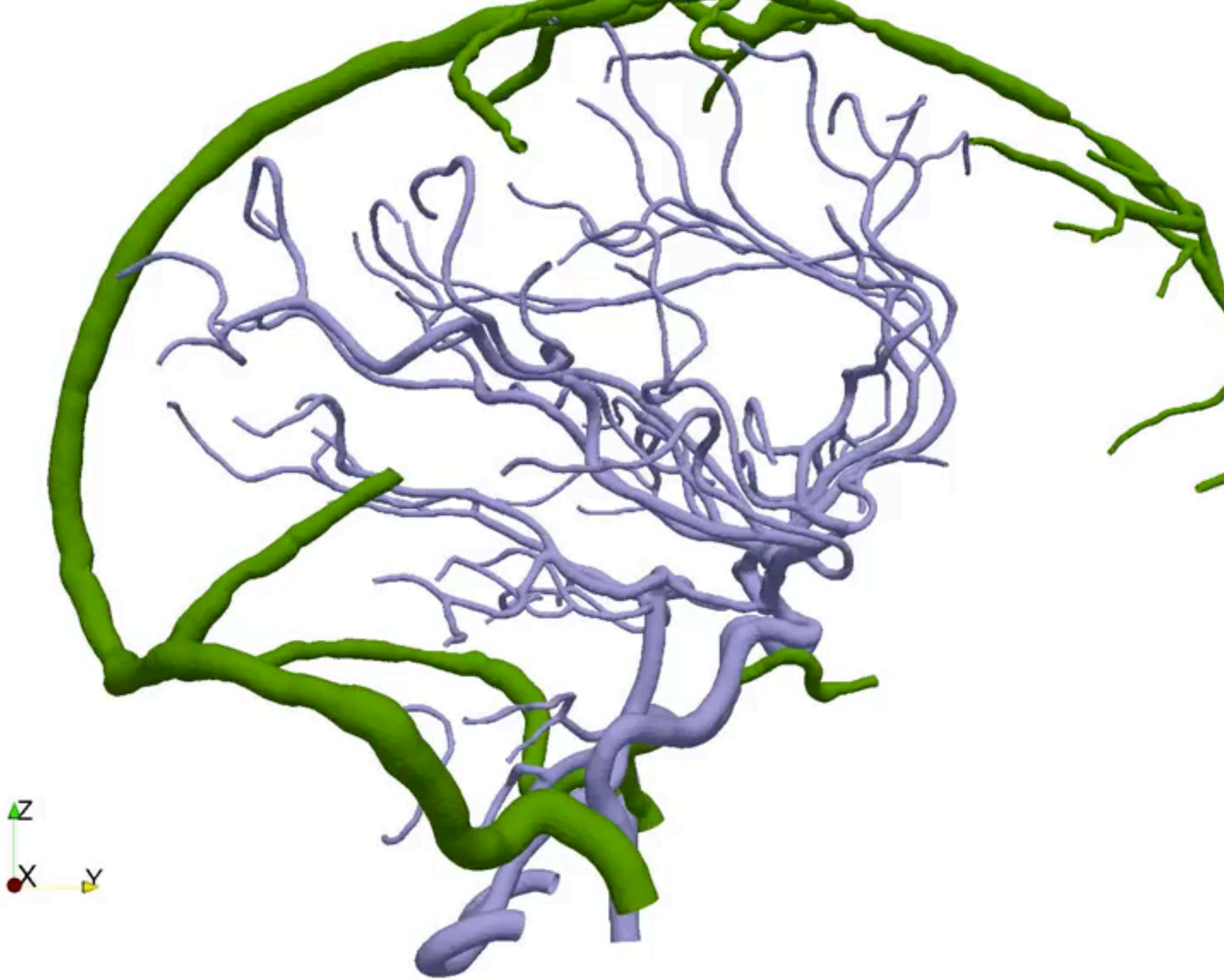
Build good computational meshes and allow extension such as building vessel wall, currently we control the mesh quality and accuracy

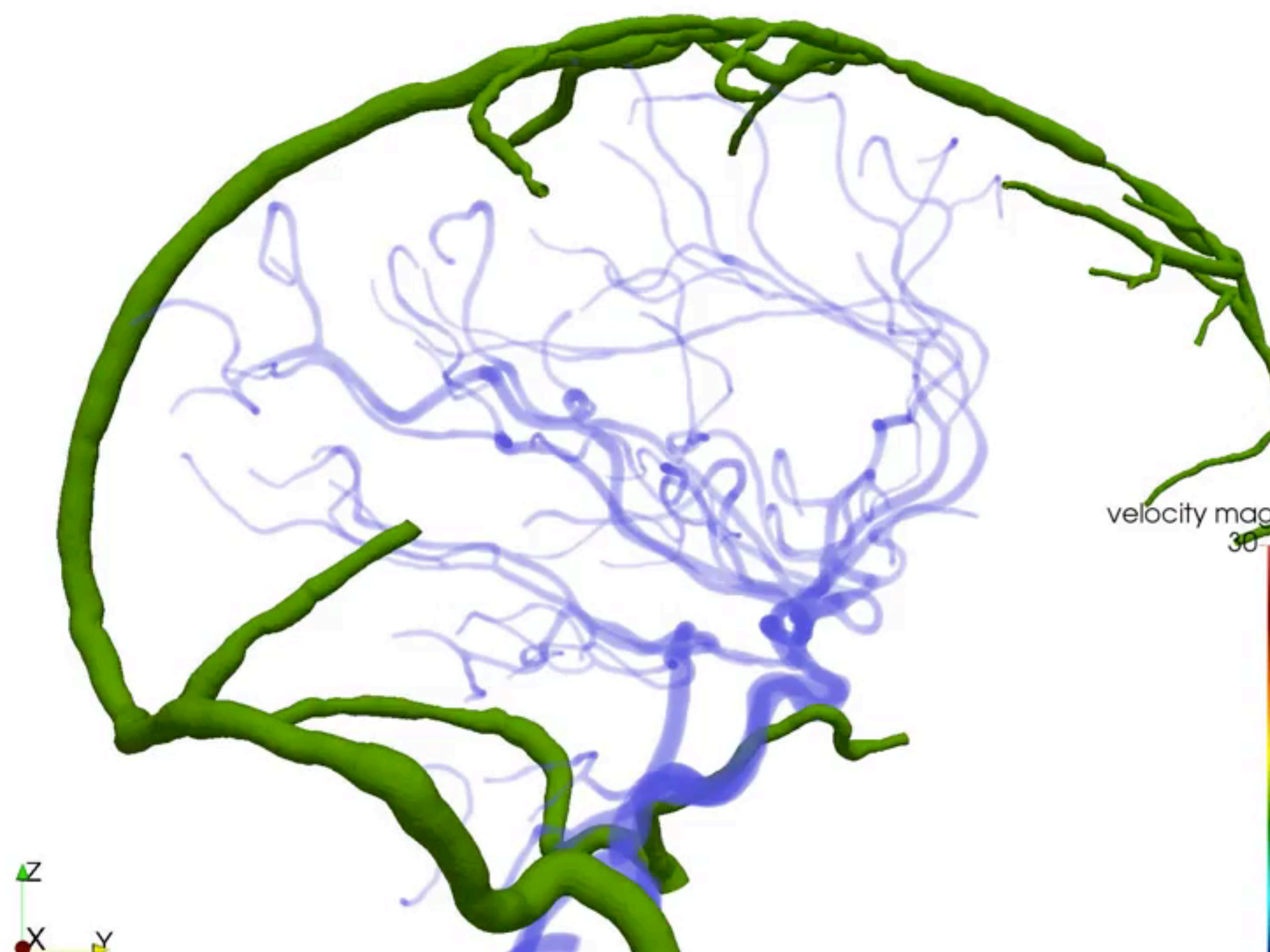


Some Challenges

Automate the process as much as possible, current difficulties centerlines and fusion of vessels




Z
X
Y



velocity mag
30
Z
X
Y

Study the influence of modeling choices in the cerebral venous network

Using a deterministic analysis framework, study the influence of

- Inflow boundary conditions
- Outflow boundary conditions
- Blood constitutive law : Newtonian vs non-Newtonian

Next Steps:

- Pressure driven flow
- Connect arterial and venous networks
- Take into account gravity
- Use statistical methods

HPC is a requirement: many simulations and very intensive post-processing computations!

Multi scale modelling of fluid-dynamical and metabolic between eye and brain

towards ocular biomarkers for neurodegenerative disorders

Joint work with

IUPUI J. Arciero, L. Carichi, S. Cassani, D. Prada, G. Guidoboni

Glick Institute A. Harris, B. Siesky

Politechnico di Milano R. Sacco

U. Strasbourg M. Szopos C. Prud'homme

AngioTK is the framework for constructing realistic geometries from medical images

Feel++ is the underlying computational framework to support detailed views in the eye and in the brain of

- blood flows
- oxygen and metabolites transport

UNIVERSITÉ DE STRASBOURG

Computational Framework
Feel++
Finite Embedded Language in C++

A Large Set of Features

A Domain Specific Language for PDEs embedded in C++ providing a syntax very close to the mathematical language to describe Galerkin methods

- Supports generalised arbitrary order Galerkin methods (cG and dG) in 1D, 2D and 3D
- Supports certified reduced basis methods
- Supports simplex, hypercube and high order meshes
- Supports seamless parallel computing
- Supports seamless interpolation between grids/function spaces
- Supports symbolic calculus thanks to GiNaC
- Supports large scale parallel linear and non-linear solvers (PETSc/SLEPc)
- Supports in-situ visualisation with Paraview

Feel++ On the Web

Website

http://www.feelpp.org

Chat room, technical discussions

http://www.gitter.im/feelpp/feelpp

Development on Github

http://github.com/feelpp/feelpp

Documentation

http://book.feelpp.org/

FeelppOrgChannel on Youtube & Google+

http://youtube.com/c/FeelppOrgChannel

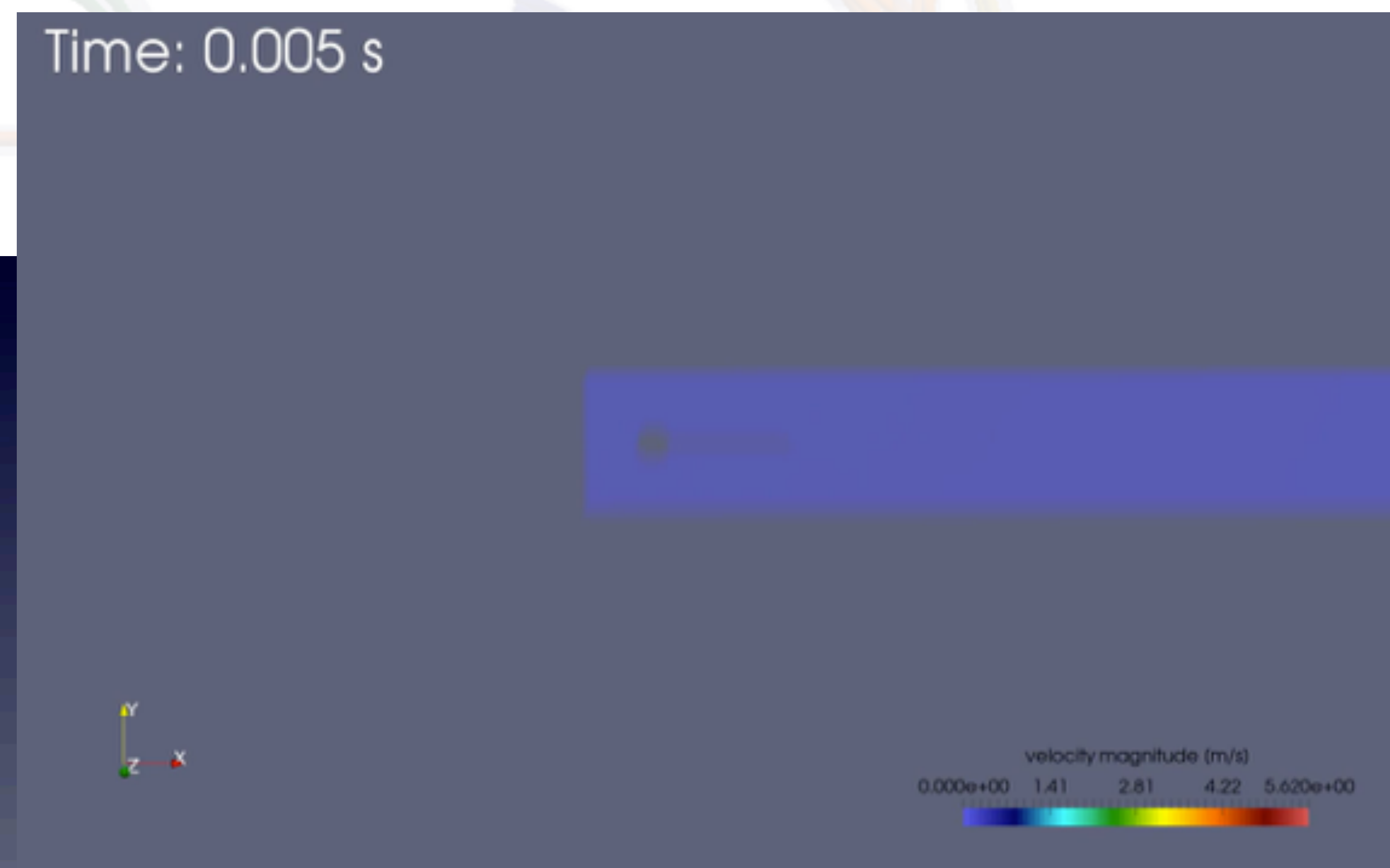
http://google.com/+FeelppOrgChannel

Hal Feel++ Collection

hal.archives-ouvertes.fr/FEEL

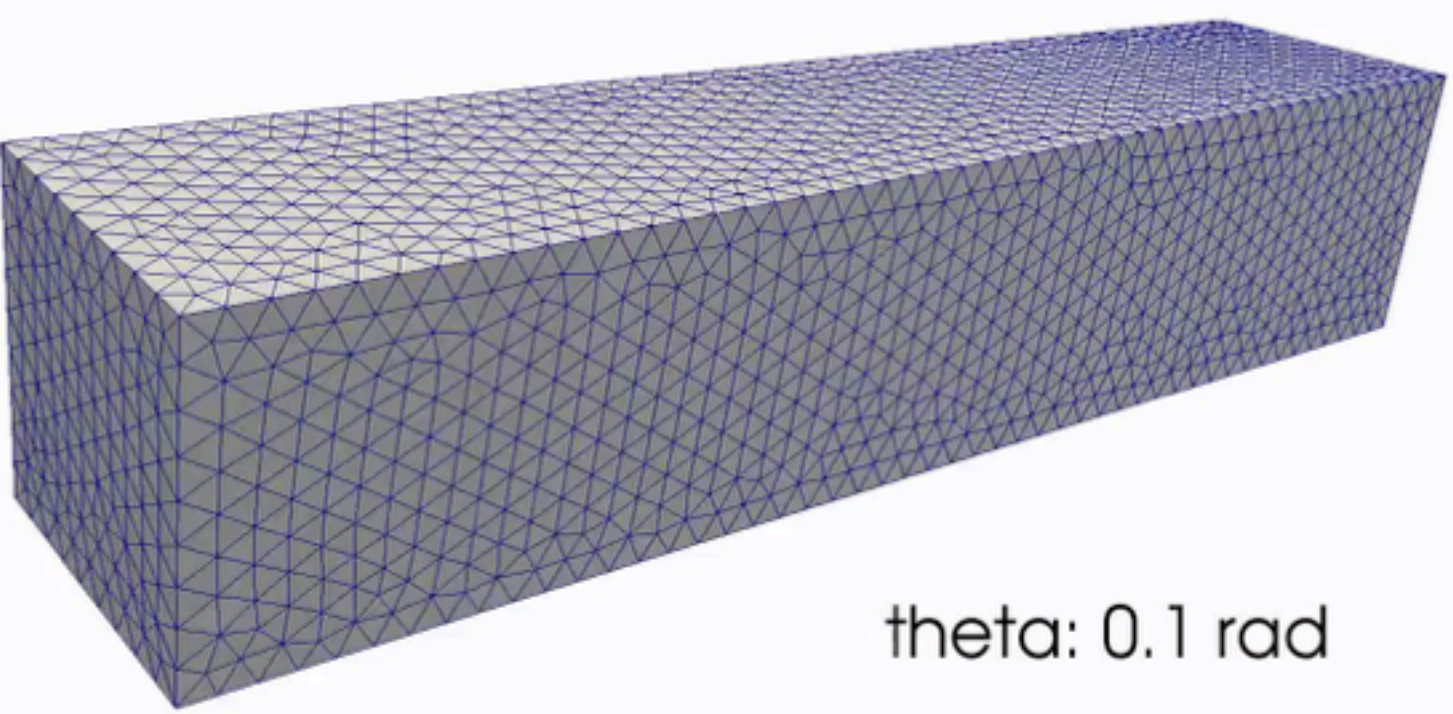
Fluid Toolbox

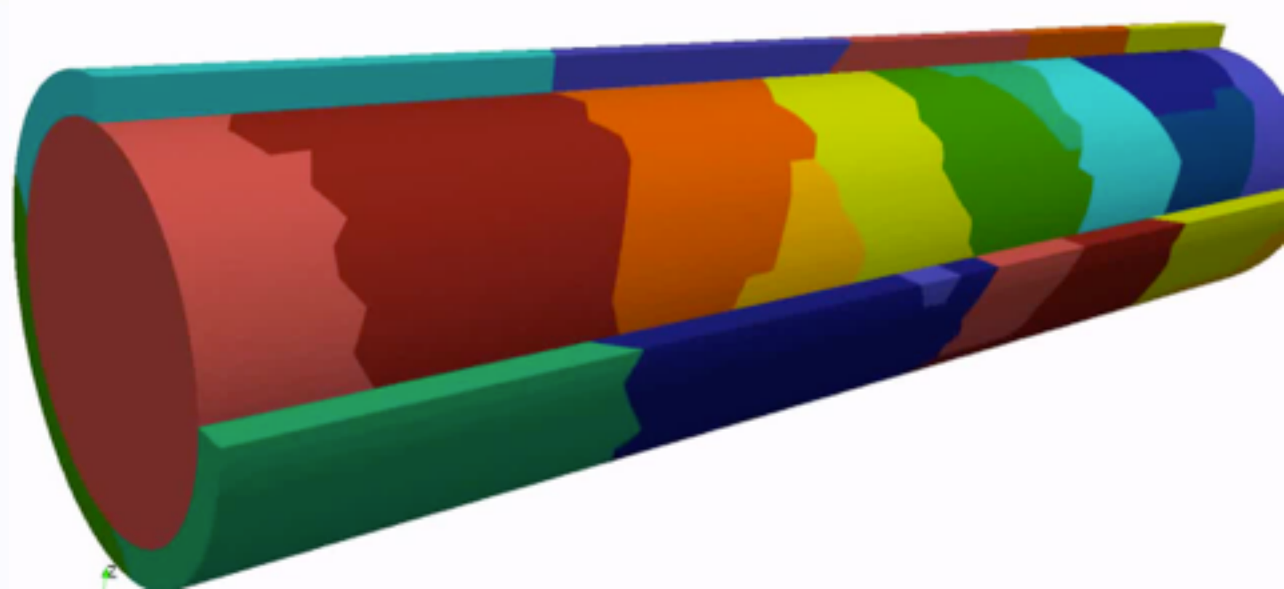
- Navier-Stokes incompressible 2D, 3D
- Newtonian and non-newtonian viscosities
- Multi-fluid support
- Moving domain support



Solid Mechanics Toolbox

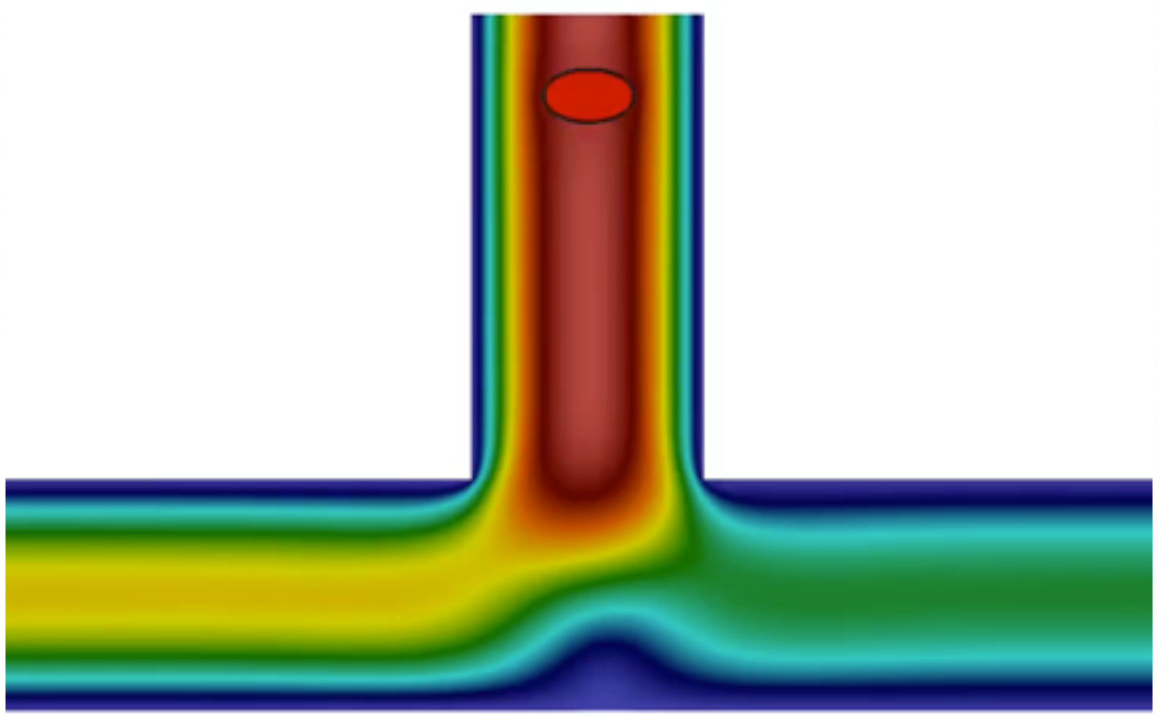
- Large deformations, large displacements
- Compressible, nearly incompressible materials
- Multi-material support





Fluid Solid Interaction Toolbox

- Partitioned methods: implicit, semi-implicit and explicit schemes
- Close fluid and structure density support
- Various methods: ALE, LevelSet, Fictitious domain



Laplace Problem

Find u such that

$$-\Delta u = f \text{ in } \Omega$$

$$u = 0 \text{ on } \partial\Omega$$

Find $u_h \in V_h$ such that

$$\int_\Omega \nabla u_h \cdot \nabla v_h = \int_\Omega f\, v_h \quad \forall v_h \in V_h$$

```
auto mesh = loadMesh(_mesh=new Mesh<Simplex<2>>);
auto Vh = Pch<2>( mesh );
auto u = Vh->element(), v = Vh->element();
auto f = expr( "2*x*y+cos(y):x:y" );
auto a = form2(_trial=Vh,_test=Vh);
a = integrate(_range=elements(mesh),
              _expr=gradt(u)*trans(grad(v)) );
auto l = form1(_test=Vh);
l = integrate(_range=elements(mesh),
              _expr=f*id(v));
a+=on(_range=boundaryfaces(mesh), _rhs=l,_element=u,
      _expr=cst(0.) );
// solve algebraic system
a.solve(_rhs=l,_solution=u);
```

Stokes Problem

Find $(\mathbf{u}, p)$ such that

$$-\Delta \mathbf{u} + \nabla p = \mathbf{f} \text{ in } \Omega$$

$$\mathbf{u} = 0 \text{ on } \partial\Omega$$

Find $(\mathbf{u}_h, p_h) \in V_h \times M_h$ such that $\forall (\mathbf{v}_h, q_h) \in V_h \times M_h$

$$\int_\Omega \nabla \mathbf{u}_h \cdot \nabla \mathbf{v}_h + p_h \nabla \cdot \mathbf{v}_h + q_h \nabla \cdot \mathbf{u}_h = \int_\Omega \mathbf{f} \cdot \mathbf{v}_h$$

```
auto mesh = loadMesh(_mesh=new Mesh<Simplex<2>>);
auto Vh = THch<1>( mesh ); // P2P1
auto U = Vh->element(), V = Vh->element();
auto u = U.element<0>(), v = V.element<0>();
auto p = U.element<1>(), q = V.element<1>();
auto a = form2(_trial=Vh, _test=Vh);
a = integrate(_range=elements( mesh ),
              _expr=inner( gradt(u),grad(v) ) );
a +=integrate(_range=elements( mesh ),
              _expr=-idt(p)*div(v) + id(q)*divt(u) );
auto l = form1(_test=Vh );
a+=on(_range=markedfaces(mesh,"wall"),_rhs=l,
       _element=u,_expr=g );
a.solve(_rhs=l,_solution=U);
```

Solvers & High Performance Computing

Scalable solvers from a few processors to thousands of processors and billions of unknowns

PETSc preconditioners

- Block-Jacobi, GASM, Multigrid (GAMG, ML),HYPRE
- Multi-Physics with fieldsplit (Gauss-Seidel, Schur)
- Mixing and tuning in Feel++ configuration file

In-House Substructuring preconditioners

- *h-p* Mortar (Elliptic)
- BDD-GenEO (P. Jolivet et al.)

Navier-Stokes block factorisation preconditioners

- SIMPLE
- LSC
- PCD
- PMM

$$P = \begin{pmatrix} F_u & B^T \\ & -\hat{S} \end{pmatrix} \quad \text{with} \quad \hat{S} = Q_p F_p^{-1} A_p$$

$A_p = B Q_p^{-1} B^T$

Q_p pressure mass matrix

F_u convection-diffusion on velocity

Scalable sub structuring preconditioners for the h-p Mortar Finite Element Method

h-p mortar finite element method

Nonconforming nonoverlapping domain decomposition method involving weak continuity constraints on space. Two approaches:

- Integrated in the approximation space
- Achieved as Lagrange multipliers

Advantages

- Different physics in different subdomains
- Heterogeneous discretizations in different subdomains
- Interesting features for parallel computing

h-p mortar finite element method

Solution strategy

- Schur Complement: decompose in terms of vertex, edge, face, volume contributions

$$Su = g$$

- Apply a change of basis allowing for block diagonal preconditioning

$$\widehat{S} = R^T S R = \begin{pmatrix} \widehat{S}_{VV} & \widehat{S}_{VE} \\ \widehat{S}_{EV} & \widehat{S}_{EE} \end{pmatrix}$$

$$\widehat{S}\widehat{u} = \widehat{g}, \qquad \widehat{u} = R^{-1}u \quad \text{and} \quad \widehat{g} = R^T g$$

h-p mortar finite element method

- Preconditioner in 2D on edges and vertices

$$P_{DG} = \begin{pmatrix} P_{DG}^V & \\ & P_E \end{pmatrix}$$

$$P_E = \begin{pmatrix} \widehat{K}_{E_1} & 0 & 0 & 0 \\ 0 & \widehat{K}_{E_2} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \widehat{K}_{E_M} \end{pmatrix}$$

$$K_E = M_E^{1/2}(M_E^{-1/2} R_E M_E^{-1/2})^{1/2} M_E^{1/2}$$

h-p mortar finite element method

- Preconditioner in 2D on edges and vertices

$$P_{DG} = \begin{pmatrix} P_{DG}^{V} & \\ & P_E \end{pmatrix}$$

$$P_V^{DG} = (1 + \log(\frac{Hp^2}{h}))(\beta P_{\text{diff}} + \gamma P_{\text{jump}})$$

$$\kappa(P_{DG}^{-1}\widehat{S}) \lesssim p^{3/2}(1 + log(\frac{Hp^2}{h}))^2$$

h-p mortar finite element method

Numerical experiments

$$-\nabla \cdot \big(\rho(\mathbf{x})\nabla u\big) = 1 \;\text{ in }\; \Omega =]0,1[^2, \qquad u = 0 \;\text{ on }\; \partial\Omega$$

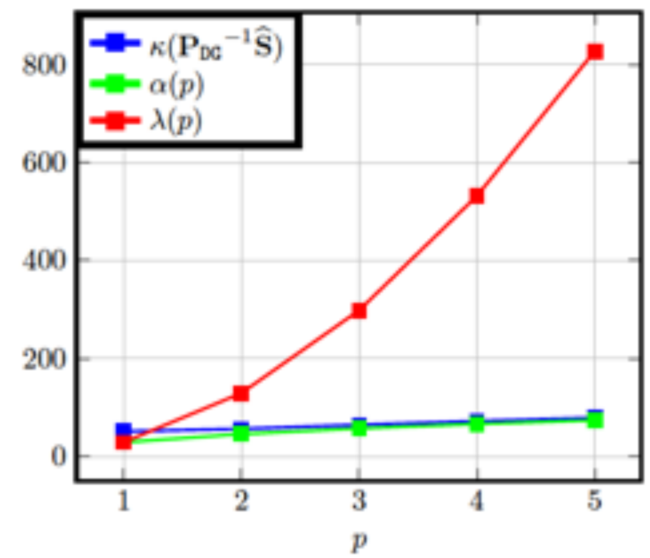
$$R = \frac{\kappa(P_{DG}^{-1}\widehat{S})}{\alpha(p)}, \qquad \alpha(p) = \left(1 + \log\left(\frac{Hp^2}{h}\right)\right)^2, \qquad \lambda(p) = p^{3/2}\alpha(p)$$

h-p mortar finite element method

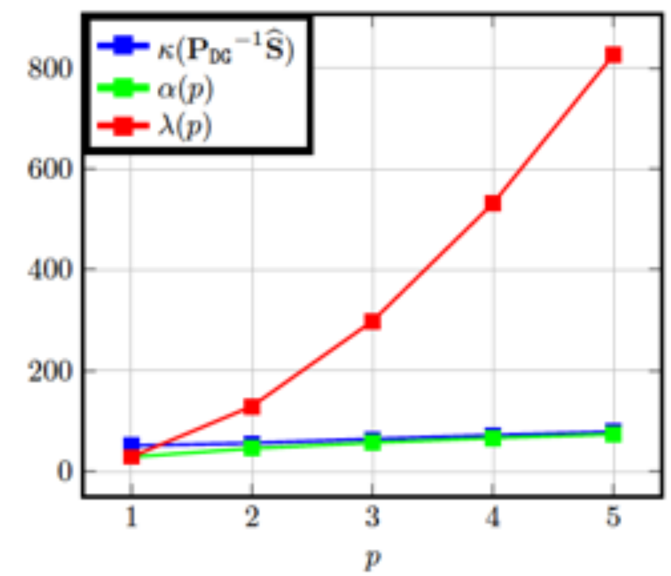
Table 5.1: Ratio R and number of iterations (between parenthesis) for $p = 5$

$N\backslash n$	5	10	20	40	80
16	1.06 (30)	1.04 (31)	1.03 (32)	1.02 (38)	1.02 (39)
64	1.11 (29)	1.09 (31)	1.08 (34)	1.07 (40)	1.07 (42)
256	1.13 (26)	1.10 (29)	1.08 (33)	1.07 (35)	1.06 (40)

- N: #of subdomains (#of processor cores)
- n: #of elements of fine mesh



#of subdomains=64

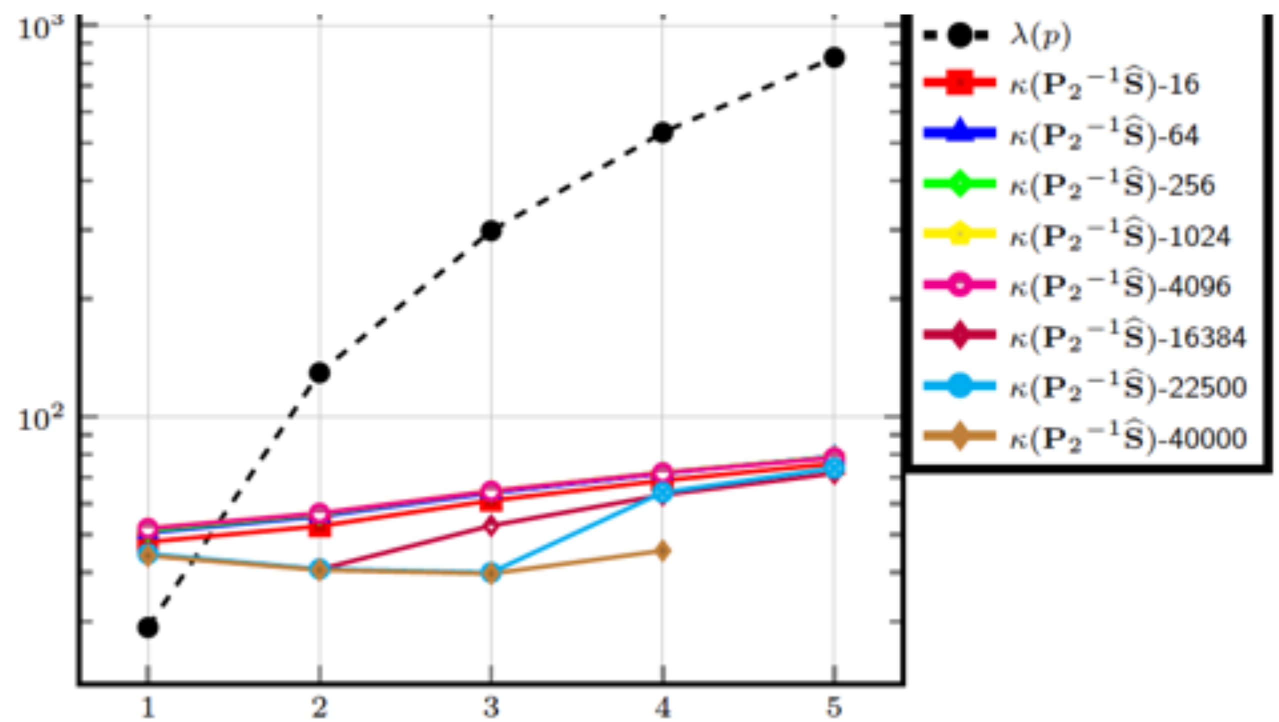


#of subdomains=256

h-p mortar finite element method

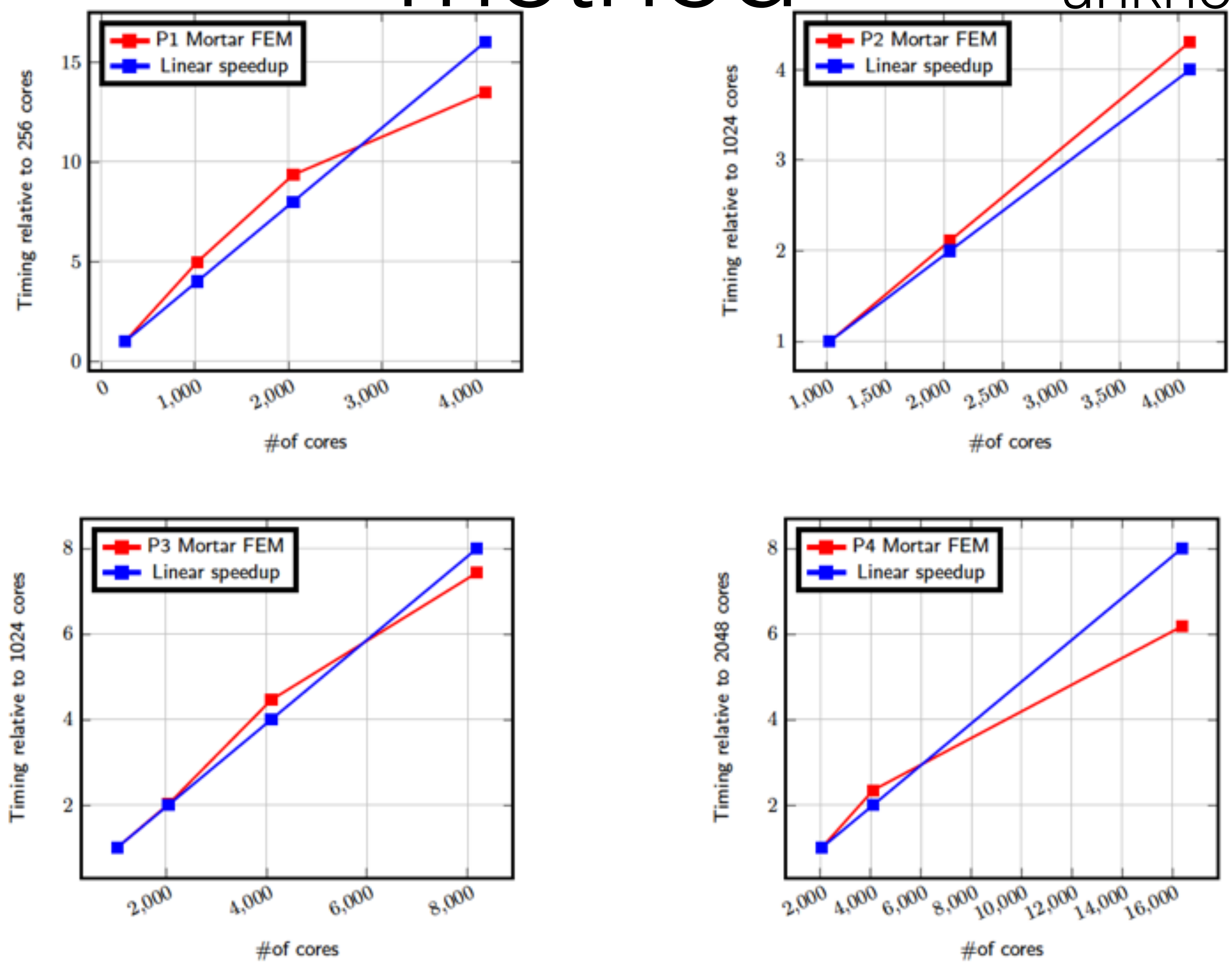
Table 5.2: Ratio R and number of iterations (between parenthesis) for $H/h = 80$

$N \backslash p$	1	2	3	4	5
16	1.65 (31)	1.14 (32)	1.06 (33)	1.03 (38)	1.02 (39)
64	1.74 (31)	1.21 (33)	1.11 (35)	1.07 (40)	1.07 (42)
256	1.76 (28)	1.23 (32)	1.12 (34)	1.08 (36)	1.06 (40)
1,024	1.78 (27)	1.23 (29)	1.12 (31)	1.08 (32)	1.06 (34)
4,096	1.79 (25)	1.23 (28)	1.12 (29)	1.08 (31)	1.06 (31)
16,384	1.52 (20)	0.88 (22)	0.91 (26)	0.94 (27)	0.96 (28)
22,500	1.52 (19)	0.88 (20)	0.69 (22)	0.95 (26)	0.99 (27)
40,000	1.52 (17)	0.88 (20)	0.69 (22)	0.68 (23)	–



h-p mortar finite element method

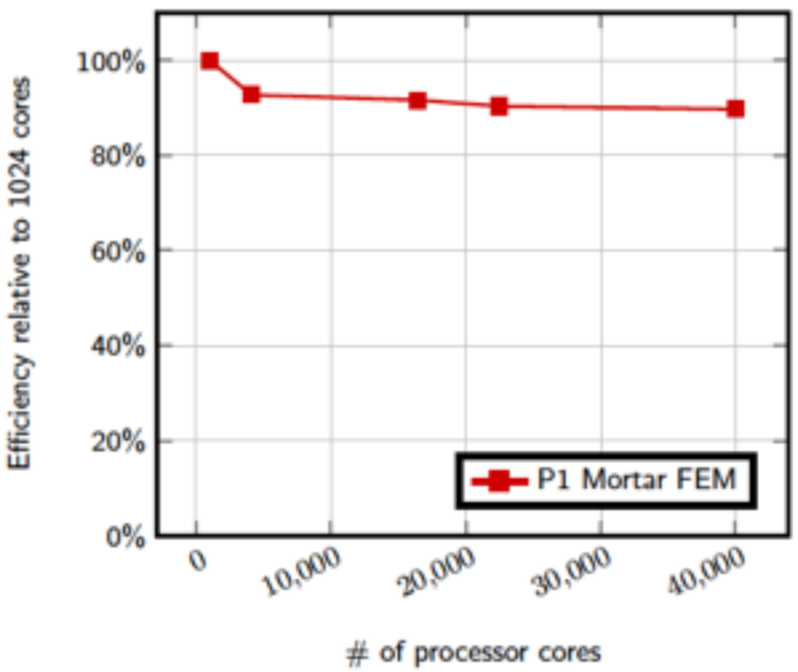
up to 500 millions unknowns

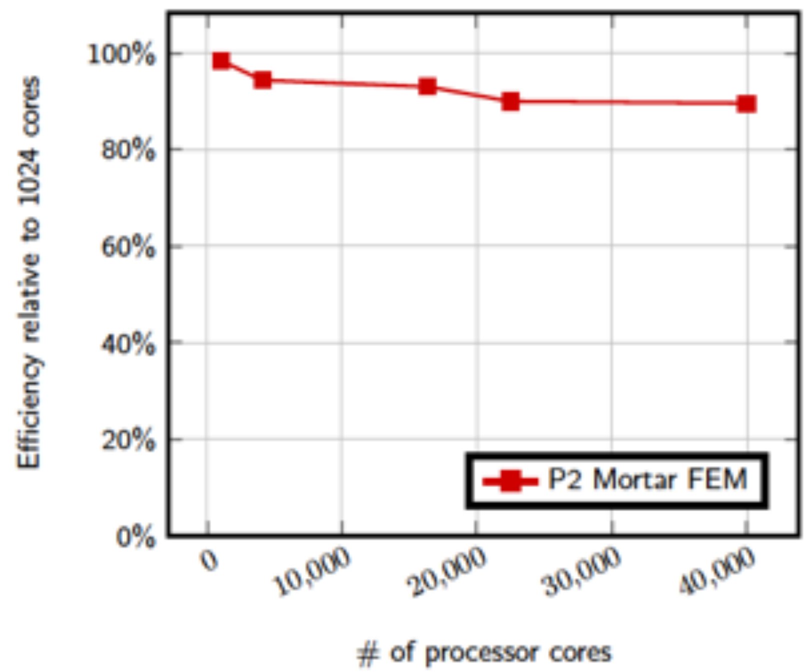


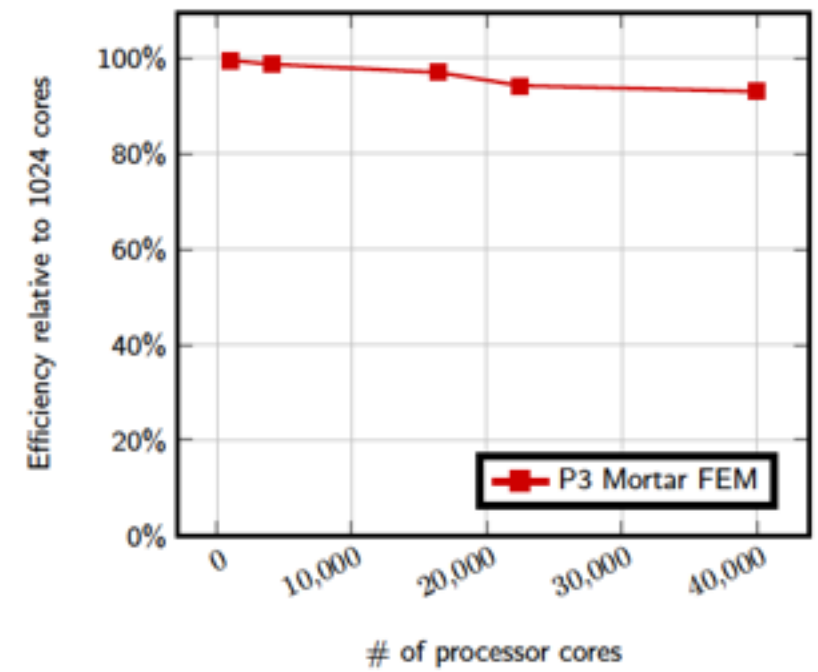
Strong scalability analysis for $p = 1, 2, 3, 4$

h-p mortar finite element method

up to 5billions unknowns







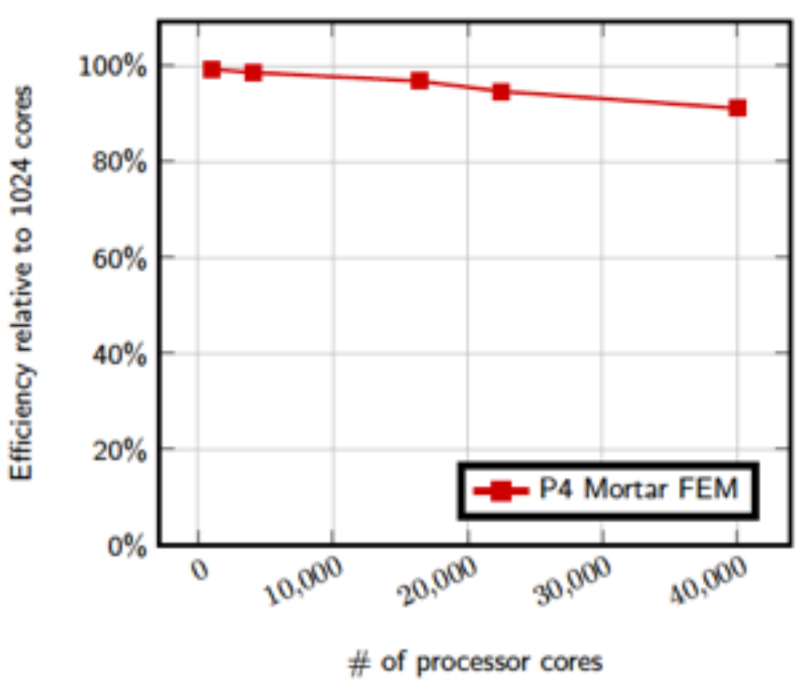


Table 5.3: Efficiency relative to 1024 processor cores

$N\backslash p$	1	2	3	4
1,024	100%	100%	100%	100%
4,096	92.73%	95.83%	99.18%	99.35%
16,384	91.67%	94.46%	97.48%	97.54%
22,500	90.37%	91.39%	94.7%	95.39%
40,000	89.73%	91.01%	93.46%	91.8%

h-p mortar finite element method

Table 5.4: Linear Elements - Preconditioned Schur Complement - R

ρ^*	ρ	$N\backslash n$	5	10	20	40	80	160	320
1e+2	1e+0	16	2.21	1.67	1.36	1.16	1.01	0.91	0.84
		64	2.2	1.66	1.35	1.15	1.02	0.92	0.86
		256	2.19	1.65	1.34	1.13	1.03	0.95	0.89
1e+3	1e+0	16	2.21	1.69	1.37	1.17	1.04	0.93	0.86
		64	2.21	1.68	1.36	1.17	1.03	0.95	0.89
		256	2.21	1.66	1.35	1.15	1.04	0.98	0.94
1e+4	1e+0	16	2.21	1.69	1.38	1.18	1.04	0.94	0.87
		64	2.21	1.69	1.37	1.17	1.04	0.95	0.91
		256	2.21	1.69	1.36	1.16	1.05	0.99	0.95
1e+9	1e+0	16	2.21	1.69	1.38	1.18	1.04	0.95	0.88
		64	2.21	1.69	1.38	1.18	1.05	0.96	0.92
		256	2.21	1.69	1.38	1.18	1.07	1.01	0.97
1e+0	1e−3	16	2.21	1.69	1.37	1.17	1.04	0.93	0.86
		64	2.21	1.68	1.36	1.17	1.03	0.95	0.89
		256	2.21	1.66	1.35	1.15	1.04	0.98	0.94
1e+0	1e−9	16	2.21	1.69	1.38	1.18	1.04	0.95	0.88
		64	2.21	1.69	1.38	1.18	1.05	0.96	0.92
		256	2.21	1.69	1.38	1.18	1.07	1.01	0.97

HDG methods

HDG Methods

Advantages of the HDG Methods

- **Optimal approximation** of both the **primal** and **flux** variables.
- **Less globally coupled degrees of freedom** than DG methods of comparable accuracy.
- **No penalization parameter** that needs **tuning** to obtain convergence.
- **Superconvergence** properties that allow for local element-by-element postprocessing.
- Suitable for devising **different methods in different parts of the computational domain** and for automatically coupling them.
- Suitable for developing **mortaring techniques**

HDG Methods

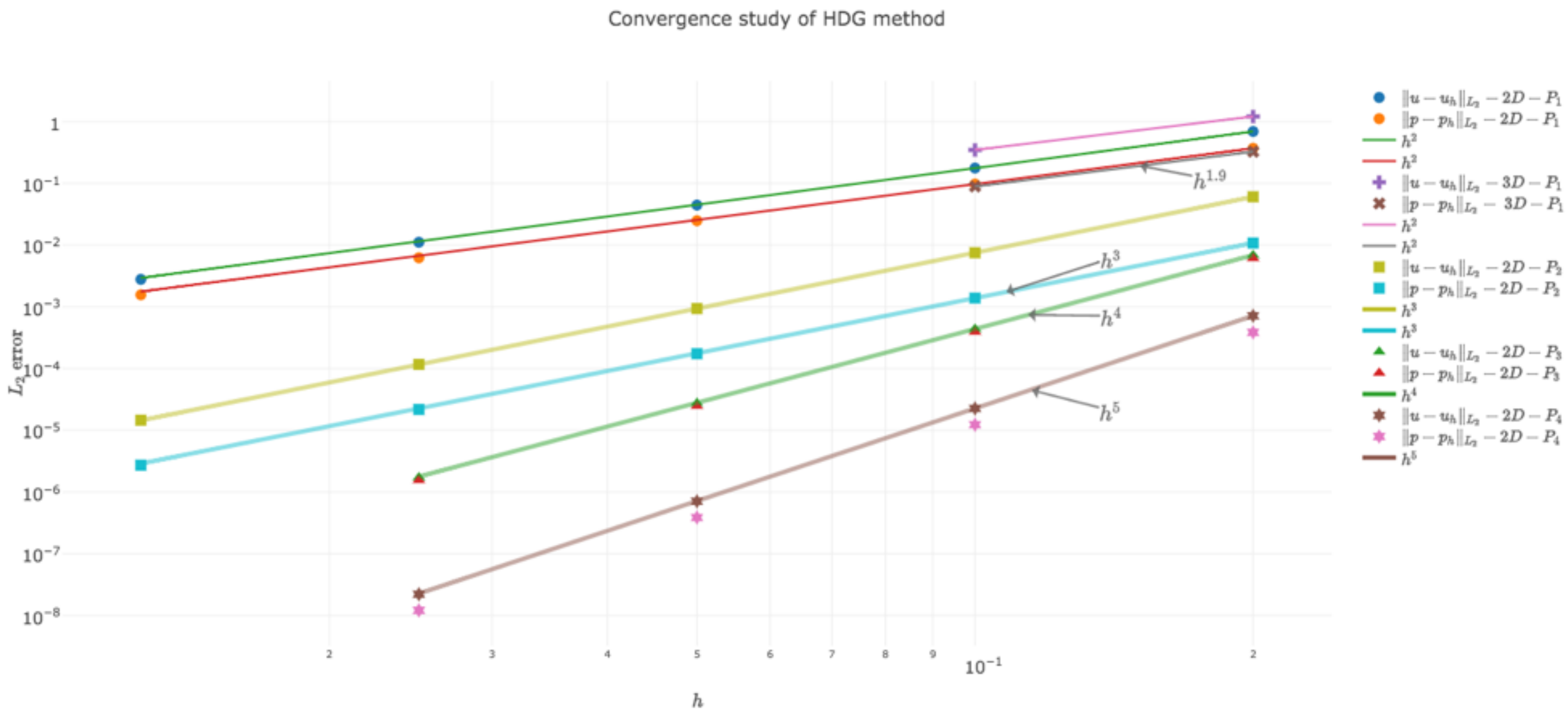
Guidelines

- Rewrite the exact problem as a collection of **local problems** and **transmission** (patching) conditions.
- Use discontinuous approximations for both the **solution** inside each element and its **trace** on the element boundary.
- Define the **local** problems by using a Galerkin method to weakly enforce the equations on each element.
- Define a **global** problem by weakly imposing the transmission conditions.

Our contribution so far

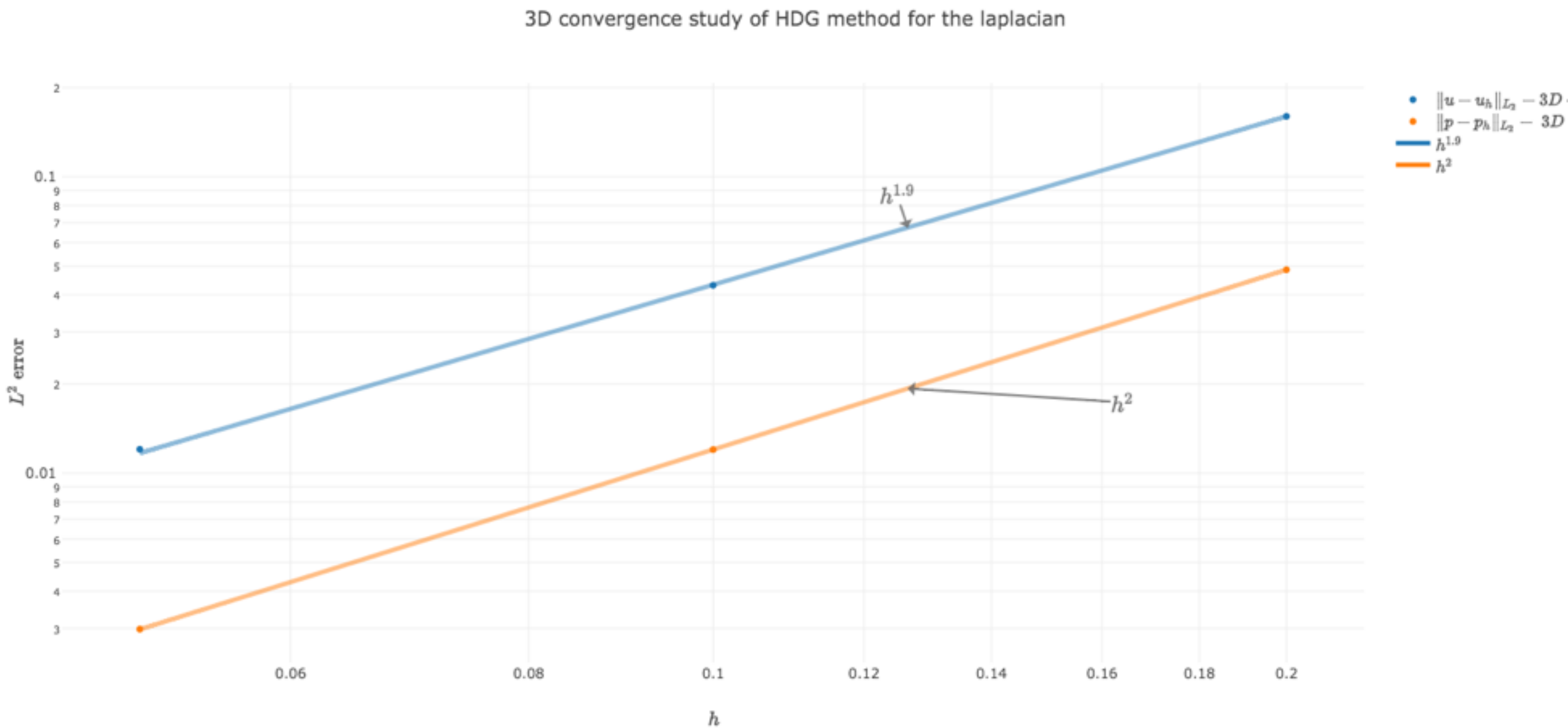
- Implementation
- Integral flux condition handling

HDG Methods



Convergence study of HDG method
1
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
L_2 error
$h^{1.9}$
h^3
h^4
h^5
2
3
4
5
6
7
8
9
10^{-1}
2
h
$\|u-u_h\|_{L_2}-2D-P_1$
$\|p-p_h\|_{L_2}-2D-P_1$
h^2
h^2
$\|u-u_h\|_{L_2}-3D-P_1$
$\|p-p_h\|_{L_2}-3D-P_1$
h^2
h^2
$\|u-u_h\|_{L_2}-2D-P_2$
$\|p-p_h\|_{L_2}-2D-P_2$
h^3
h^3
$\|u-u_h\|_{L_2}-2D-P_3$
$\|p-p_h\|_{L_2}-2D-P_3$
h^4
$\|u-u_h\|_{L_2}-2D-P_4$
$\|p-p_h\|_{L_2}-2D-P_4$
h^5

HDG Methods



Applicability requires static condensation

HDG Methods

$$
\begin{aligned}
&\underline{J} + \sigma \nabla p = 0, \quad \nabla \cdot \underline{J} = 0 && \text{in } \Omega \\
&\nabla \cdot (-k \nabla T) = J^2/\sigma && \text{in } \Omega \\
&p = V && \text{on } V_0 \\
&\int_{V_0} \underline{J} \cdot \underline{n} = I_{\text{target}} && \text{on } V_1 \\
&\underline{J} \cdot \underline{n} = 0, \quad k \nabla T|_{Ch_1 \cup Ch_2} = h_1 (T_{\text{W}} - T) && \text{on } \Gamma_C = Ch_0 \cup Ch_1 \\
&\underline{J} \cdot \underline{n} = 0, \quad k \nabla T|_{Rod \cup Hls} = h_2 (T_{\text{W}} - T) && \text{on } \Gamma_{RH} = Rod \cup Hls
\end{aligned}
$$

$\underline{J}$, current density; σ, electrical conductivity; p, electric potential

k thermal conductivity; T, temperature; h_1, h_2, heat transfer coefficients; T_{W}, water temperature.

HDG Methods

$$
\begin{aligned}
\underline{V}_h &= \{\, \underline{v} \in [L^2(\Omega_h)]^3 \mid \underline{v}|_K \in [P^k(K)]^3 \quad \forall K \in \Omega_h \,\},\\
W_h &= \{\, w \in L^2(\Omega_h) \mid w|_K \in P^k(K) \quad \forall K \in \Omega_h \,\},\\
M_h &= \{\, \mu \in L^2(\mathcal{E}_h) \mid \mu|_e \in P^k(e) \quad \forall e \in \mathcal{E}_h \,\},\\
C_h &= \{\, m \in C^0(\Omega_h) \mid m|_K \in P^0(K) \quad \forall K \in \Omega_h \,\} \equiv \mathbb{R},\\
X_h &= \{\, q \in C^0(\Omega_h) \mid q|_K \in P^k(K) \quad \forall K \in \Omega_h \,\}.
\end{aligned}
$$

HDG Methods

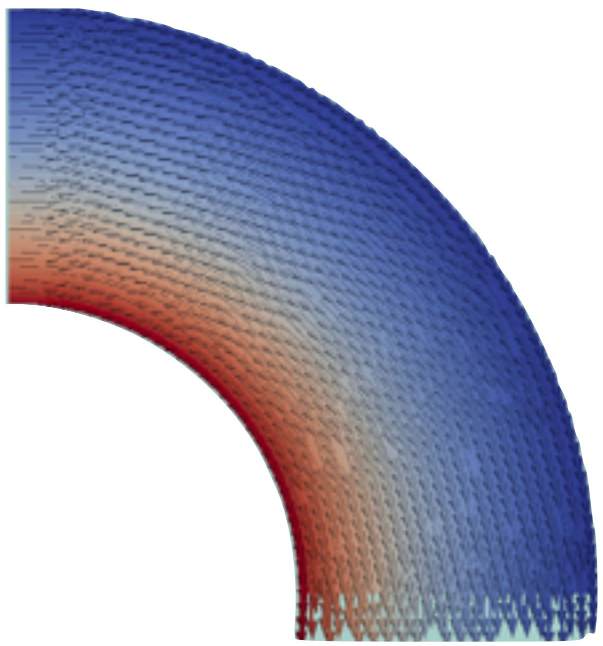
Find $(\underline{J}_h, p_h, \hat{p}_h, \lambda_h, T_h) \in \underline{V}_h \times W_h \times M_h \times C_h \times X_h$ for which:

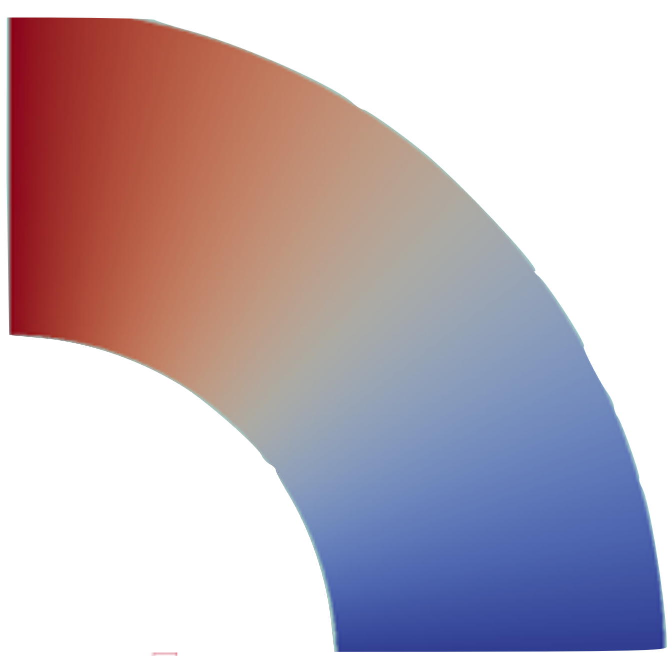
$$
\begin{aligned}
(\sigma^{-1}\underline{J}_h, \underline{v})_{\Omega_h} - (p_h, \nabla\cdot\underline{v})_{\Omega_h} + \langle \hat{p}_h, \underline{v}\cdot\underline{n}\rangle_{\partial\Omega_h} + \langle \lambda_h, \underline{v}\cdot\underline{n}\rangle_{V_1} &= 0,\\
(\nabla\cdot\underline{J}_h, w)_{\Omega_h} + \langle \tau p_h, w\rangle_{\partial\Omega_h} - \langle \tau\hat{p}_h, w\rangle_{\partial\Omega_h} - \langle \tau\lambda_h, w\rangle_{V_1} &= 0,\\
\langle \underline{u}_h\cdot\underline{n}, \mu\rangle_{\partial\Omega_h\backslash\partial\Omega} + \langle \tau p_h, \mu\rangle_{\partial\Omega_h\backslash\partial\Omega} - \langle \tau\hat{p}_h, \mu\rangle_{\partial\Omega_h\backslash\partial\Omega} &= 0,\\
-\langle \tau\lambda_h, \mu\rangle_{V_1} &= 0,\\
\langle \underline{J}_h\cdot\underline{n}, \mu\rangle_{\Gamma_C\cup\Gamma_{RH}} + \langle \tau p_h, \mu_2\rangle_{\Gamma_C\cup\Gamma_{RH}} - \langle \tau\hat{p}_h, \mu_2\rangle_{\Gamma_C\cup\Gamma_{RH}} &= 0,\\
\langle \hat{p}_h, \mu\rangle_{V_0} &= \langle V, \mu\rangle_{V_0},\\
\langle \underline{J}_h\cdot\underline{n}, m\rangle_{V_1} + \langle \tau p_h, m\rangle_{V_1} - \langle \tau\hat{p}_h, m\rangle_{V_1} &= \langle I_{target}, m\rangle_{V_1},\\
(k\nabla T_h, \nabla q) + \langle hT_h, q\rangle_{\Gamma_C\cup\Gamma_{RH}} = \langle hT_{\mathrm{W}}, q\rangle_{\Gamma_C\cup\Gamma_{RH}} &+ (\sigma^{-1}J_h^2, q)_{\Omega_h},
\end{aligned}
$$

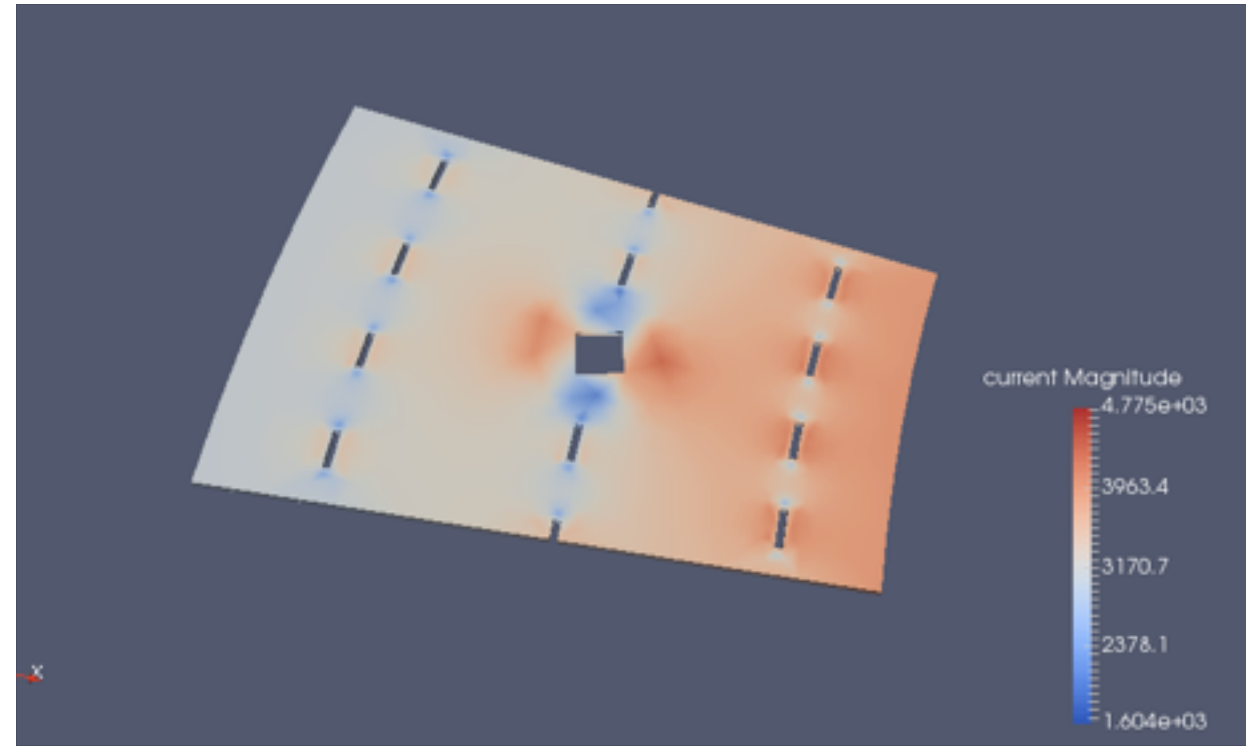
for all $(\underline{v}, w, \mu, m, q) \in \underline{V}_h \times W_h \times M_h \times C_h \times X_h$.

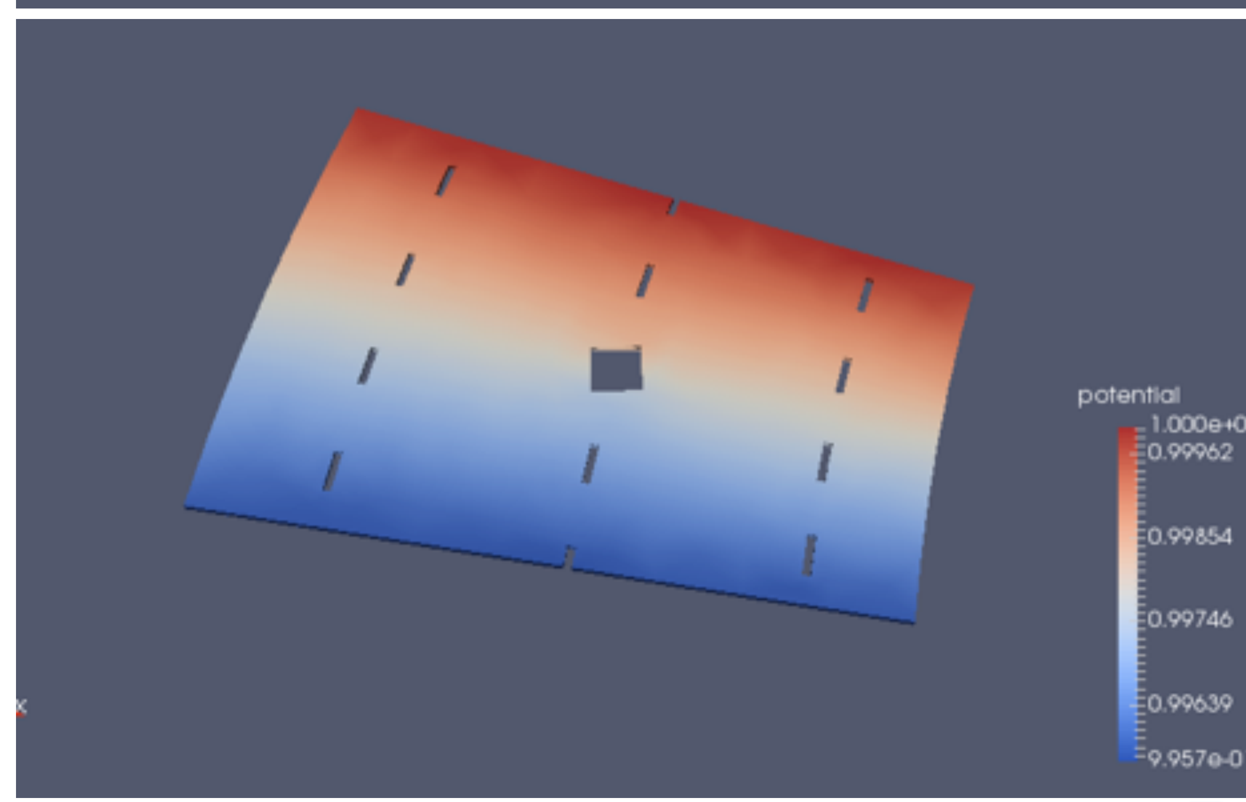
- Well-posedness: The problem is well-posed if $\tau > 0$ on ∂K, $\forall\, K \in \Omega_h$.
- Order of Convergence: If polynomials of degree $k \geq 0$ are used and τ is suitably chosen, then all the variables converge with order $k+1$. If $k \geq 1$, p_h can be postprocessed to get a new approximation p_h^* converging with order $k+2$.

HDG Methods

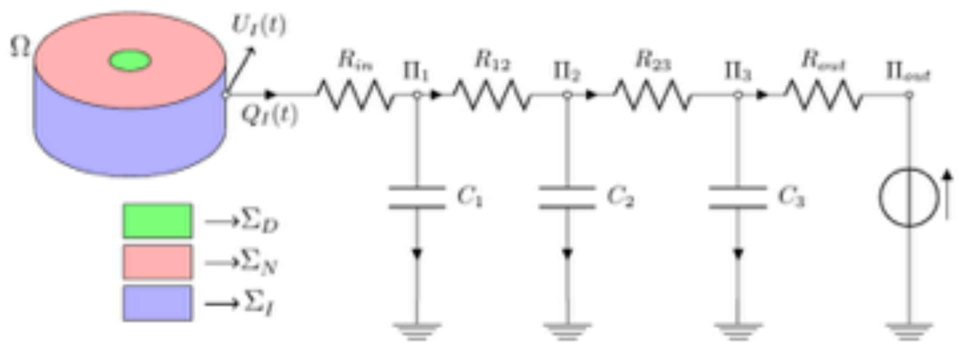


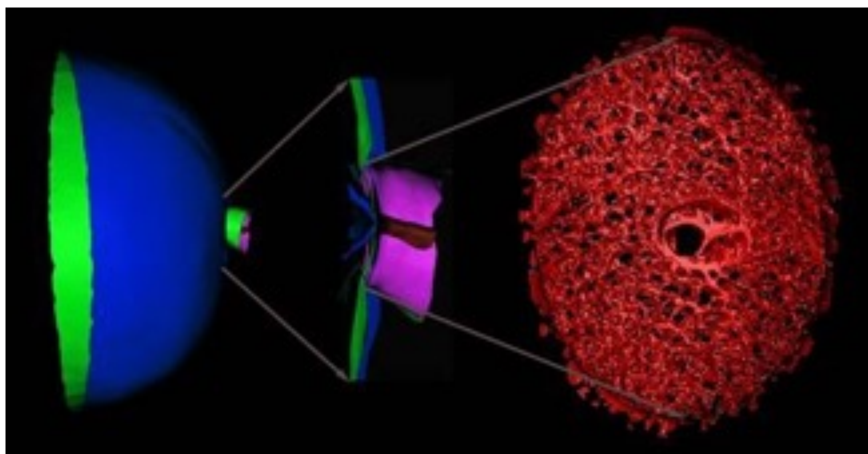


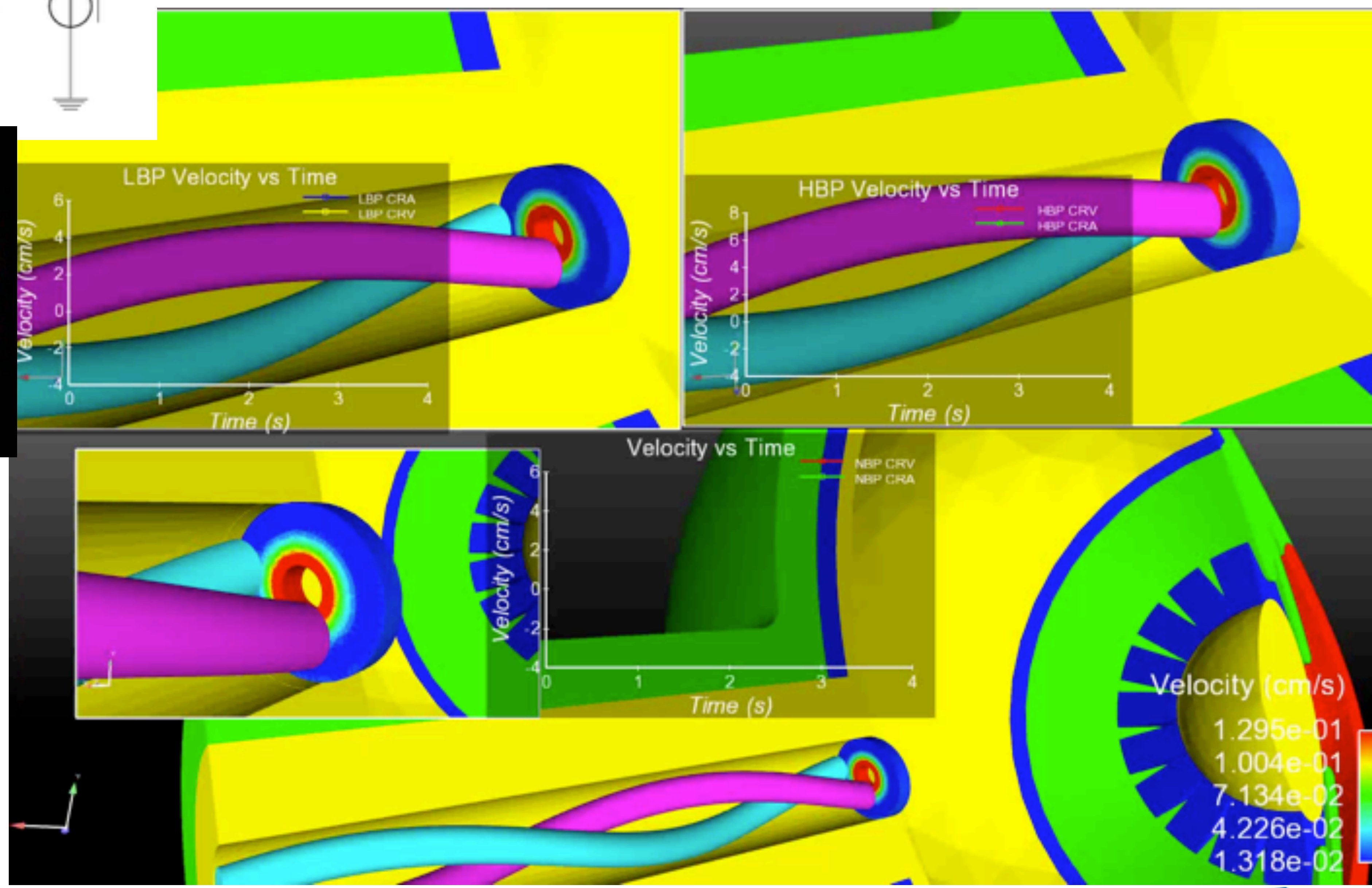

current Magnitude
4.775e+03
3963.4
3170.7
2378.1
1.604e+03


potential
1.000e+00
0.99962
0.99854
0.99746
0.99639
9.957e-01

HDG Methods for Eye2Brain







$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{j}) = 0 \text{ in } \Omega \times (0,T)$$

$$\mathbf{J} = -k \nabla u\,.$$

$$\mathbf{C}\frac{d\mathbf{\Pi}}{dt} + \mathbf{R}\,\mathbf{\Pi} = \mathbf{g}$$

$$\int_{\Sigma_I} \mathbf{J}\cdot\mathbf{n} = Q_I(t) \text{on } \Sigma_I \times (0,T)$$

$$u(\mathbf{x},t) = U_I(t) \text{on } \Sigma_I \times (0,T)$$

$$u(\mathbf{x},t) = u_D(\mathbf{x},t)\ \mathbf{x} \in \Sigma_D \times (0,T)$$

$$\mathbf{J}(\mathbf{x},t)\cdot\mathbf{n}(\mathbf{x},t) = 0\ \mathbf{x} \in \Sigma_N \times (0,T)$$

Some Conclusions & Some Perspectives

- HPC is mostly hidden to the end-user
- Applications drive partly methodology and enables to measure the effectiveness and applicability of the theoretical developments
- Shortened theory-development cycles
- HDG spreads all over our applications once static condensation is available
- Cemracs
  - RBC 3D (contact, FM),
  - Static condensation for HDG
  - Scalable preconditioners for HDG
  - HDG for other models (eye2brain)
  - FSI/ALE+FSI/LS on hemotum++

Thanks !