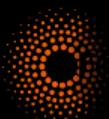




waterscales



## Finite elements and brain multiphysics

Marie E. Rognes

Chief Research Scientist  
Simula Research Laboratory  
Oslo, Norway

Fulbright Visiting Scholar  
Institute of Engineering in Medicine  
University of California San Diego

Nov 8 2022



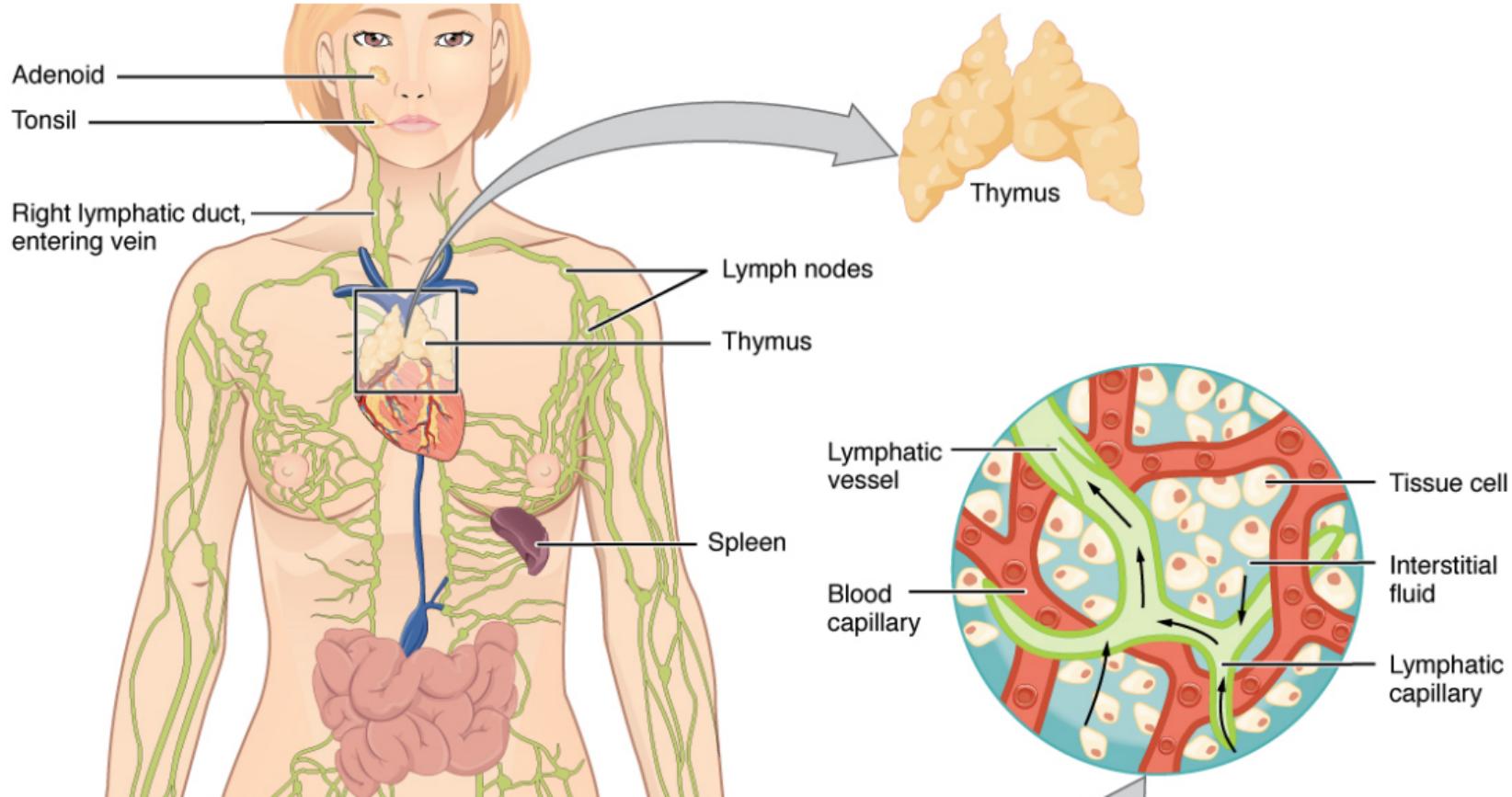
# The brain may clean out Alzheimer's plaques during sleep

If sleep deprivation puts garbage removal on the fritz, the memory-robbing disease may develop

BY LAURA BEIL 6:00AM, JULY 15, 2018

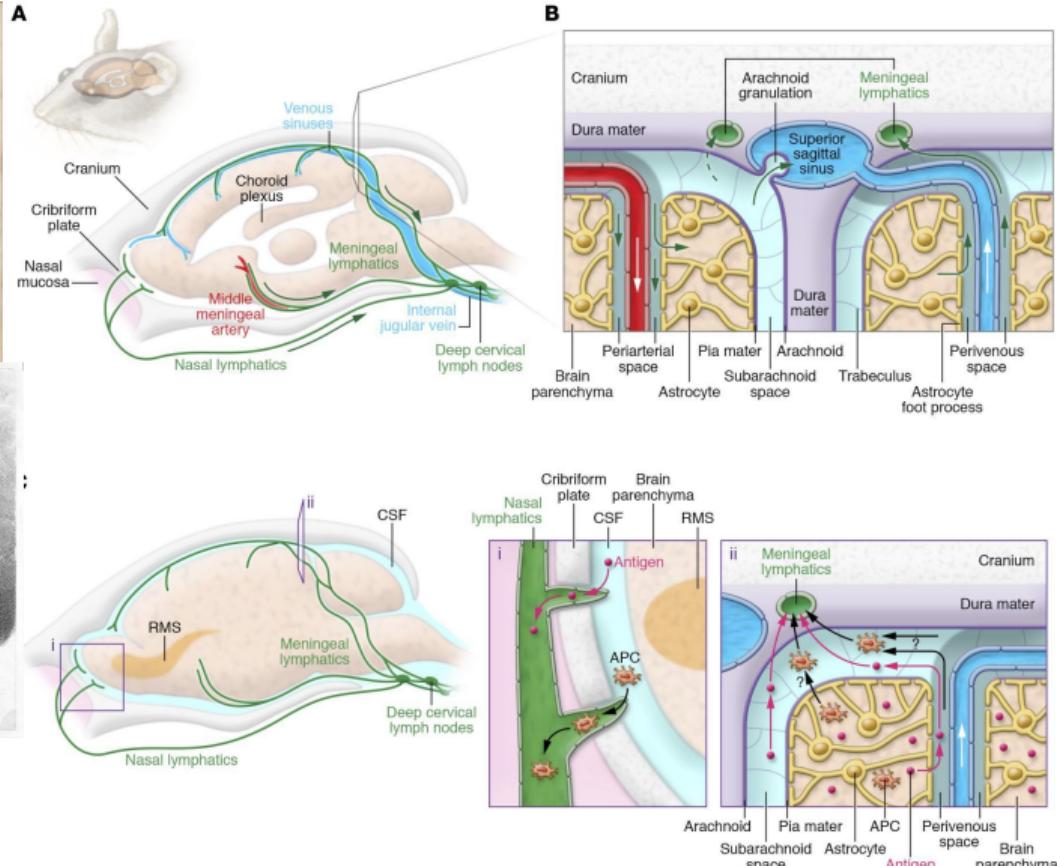
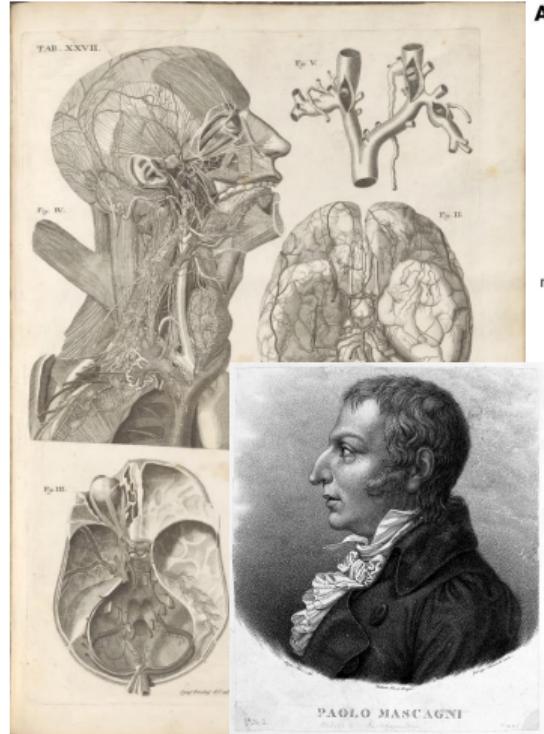


# The lymphatic system drains tissue fluid and collects metabolic waste, bacteria and cellular debris



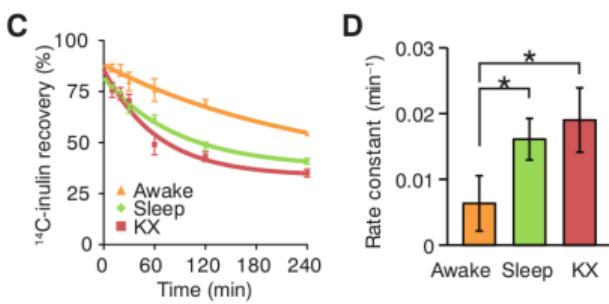
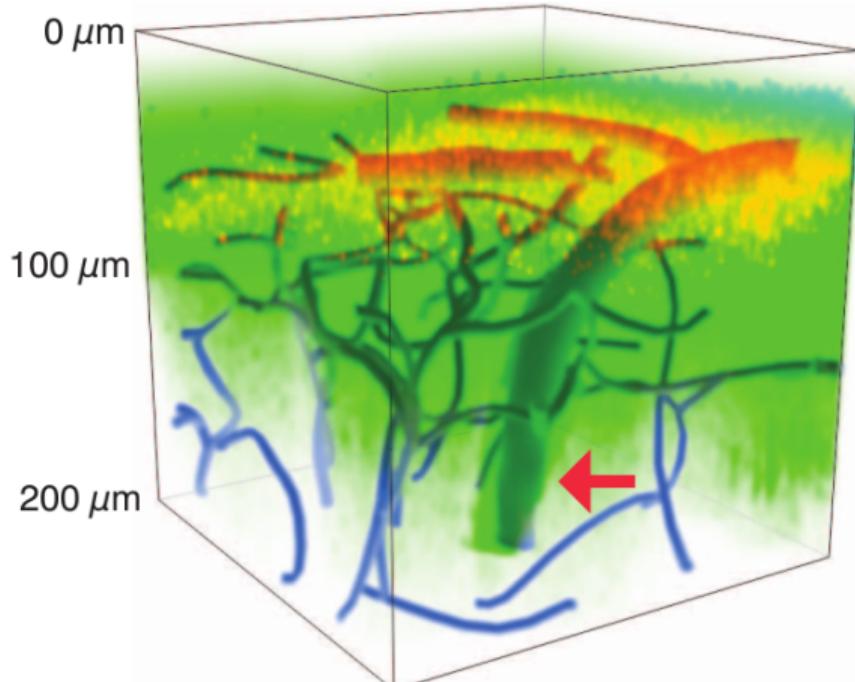
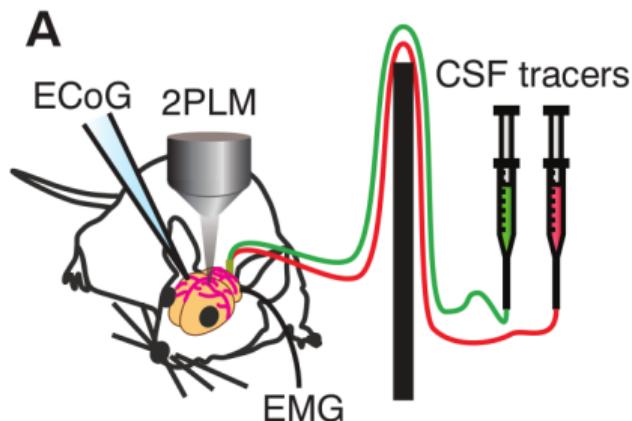
# Computational brainphatics for understanding the brain's wandscape

[Louveau et al, 2017 (Fig 2)]



[Paolo Mascagni, *Vasorum Lymphaticorum Corporis Humani Historia et Ichnographia* (1787)]

# Sleep: a fundamental driver of metabolic clearance from the brain?



[Cserr et al (1981), Rennels et al (1985), Ichimura et al (1991), Abbott (2004), Hadaczek et al (2005), Iliff et al (2012), Xie et al (2012, Fig 1A-B, Fig 2C-D), Bojarskaite et al (2020)]

## Overview

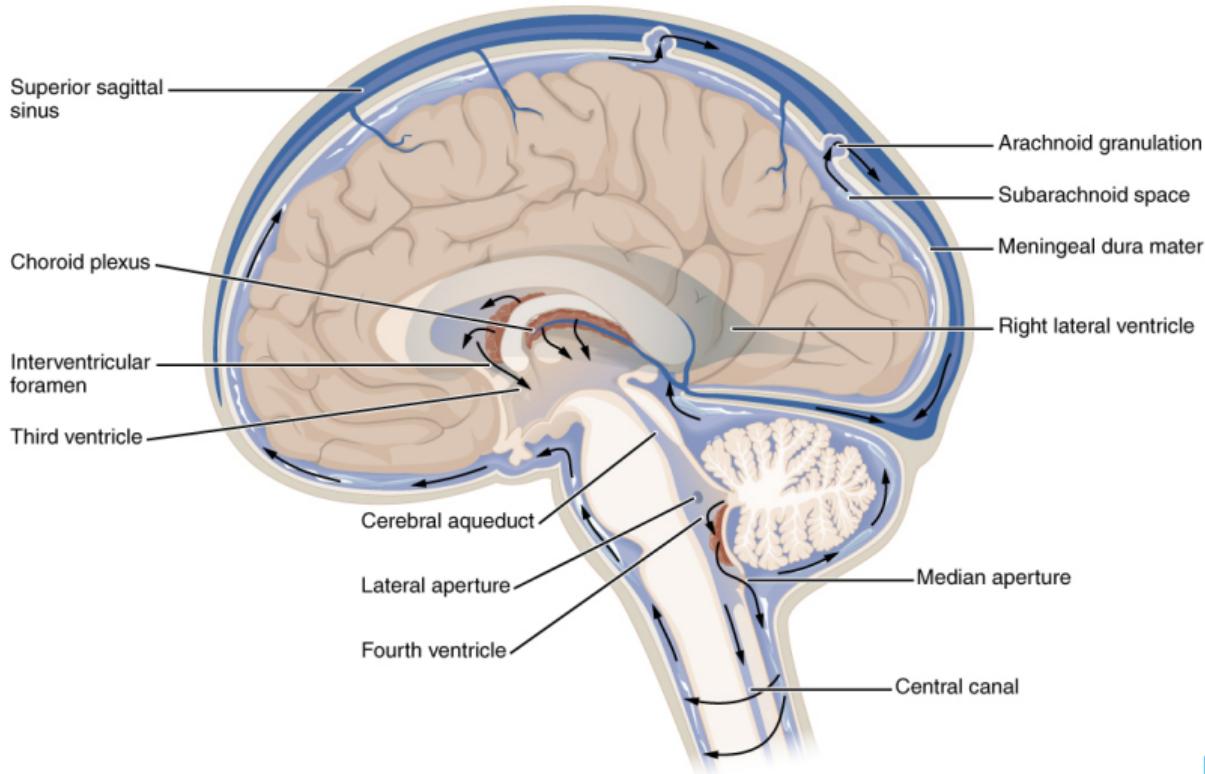
1. Motivation
2. Brain physiology and physics
3. Finite elements for brain poroelasticity  
Break
4. Simulating brain pulsatility via fluid-structure interactions
5. Simulation software and tools

## Goals

- ◊ Understand core brain physiology and mechanics
- ◊ Understand why and how the brain can be modelled as a poroelastic medium (Biot) in a viscous fluid environment (Stokes).
- ◊ Learn about finite element methods for solving Biot's equations (and coupled Stokes-Biot multiphysics problems).
- ◊ Understand how brain modelling and simulations can inform physiology and medicine
- ◊ Learn about resources for getting started with brain modelling and simulations

## Brain multiphysics

# Cerebrospinal fluid (CSF) circulates in spaces surrounding the brain



[Wikimedia Commons]

# Intracranial dynamics result from an interplay between arterial blood influx, cerebrospinal fluid flow, venous outflux, and compliances

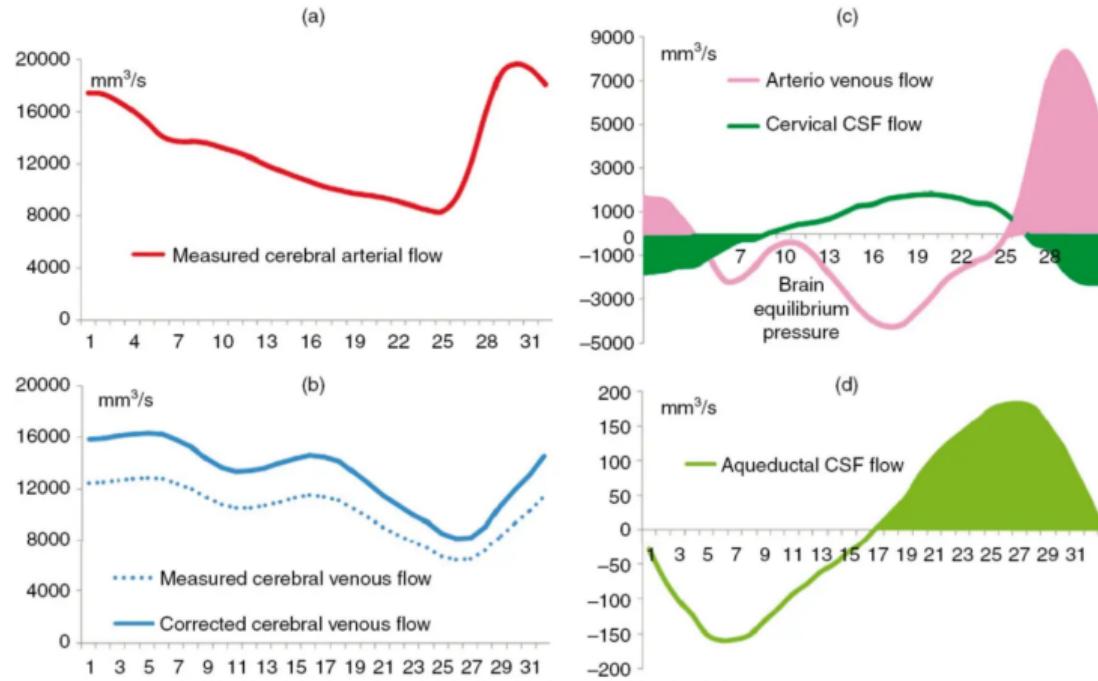
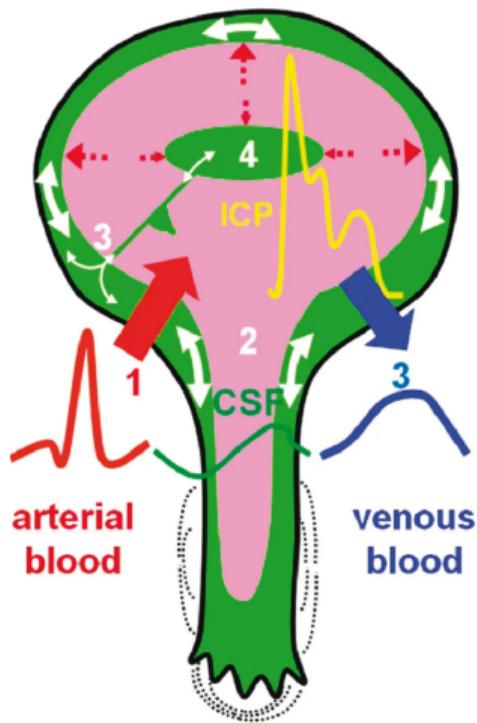
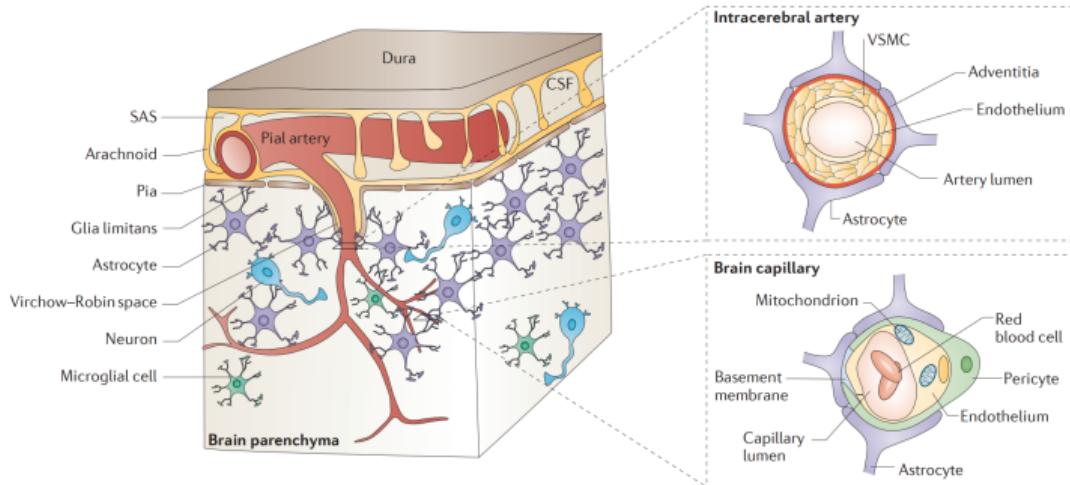


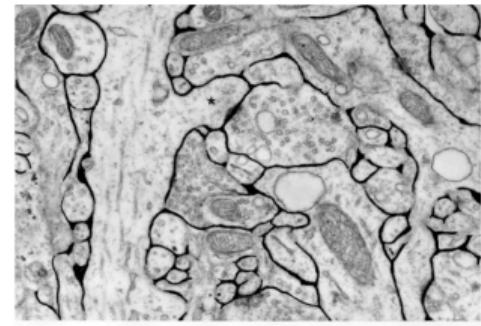
Figure 12.4 CSF and cerebral blood flow during the cardiac cycle in healthy adults (see text for full explanation).

# At the macroscale, the brain can be viewed as an elastic medium permeated by multiple fluid-filled networks



The brain parenchyma includes multiple fluid networks (extracellular spaces (ECSs), arteries, capillaries, veins, paravascular spaces (PVSs))

[Zlokovic (2011)]



Rat cerebral cortex with ECS in black  
(Scale bar:  $\approx 1\mu\text{m}$ )

[Nicholson (2001) (Fig. 2)]

**The brain is ( $\approx$ ):**  
5-10% blood  
20% ECS  
70-75% brain cells  
80% water

[Buday et al (2019)]

# Brain tissue is soft, heterogeneous and rheologically complex

## Brain tissue is

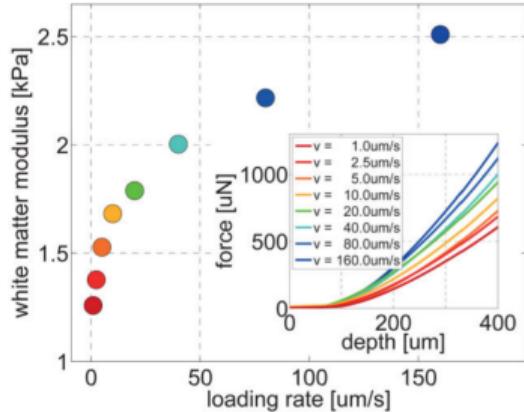
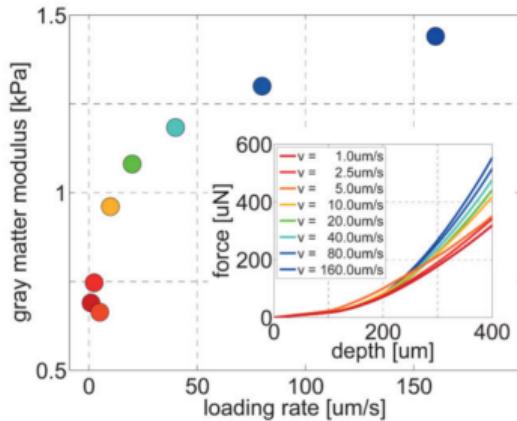
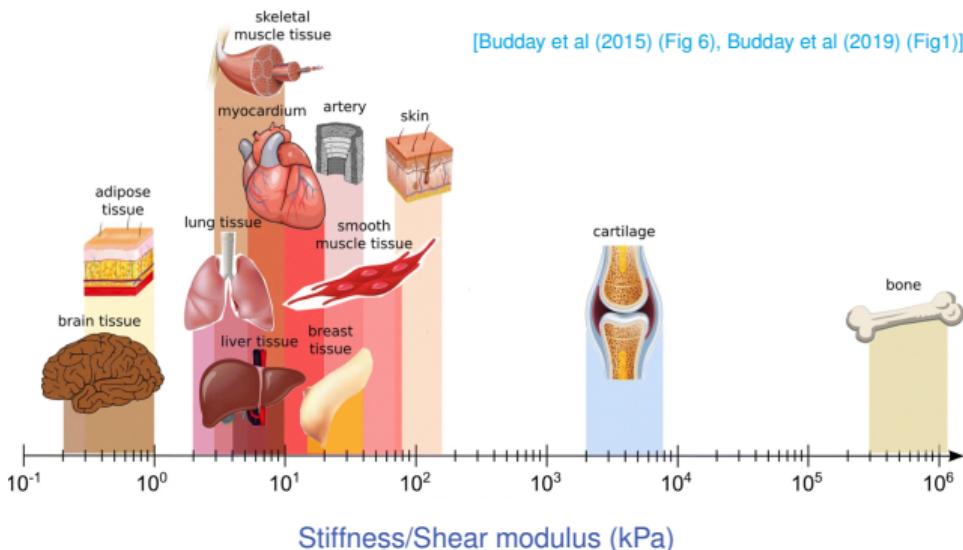
soft (shear modulus  $\approx 0.5\text{--}2.5\text{ kPa}$ )

stiffer with increasing strain/strain rates (nonlinear)

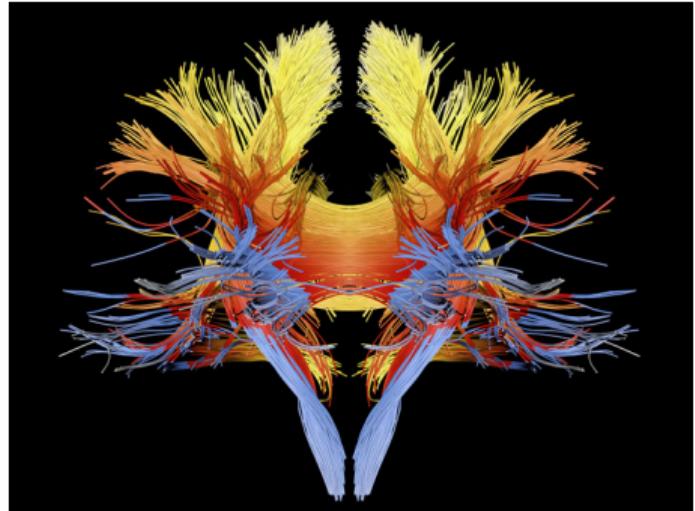
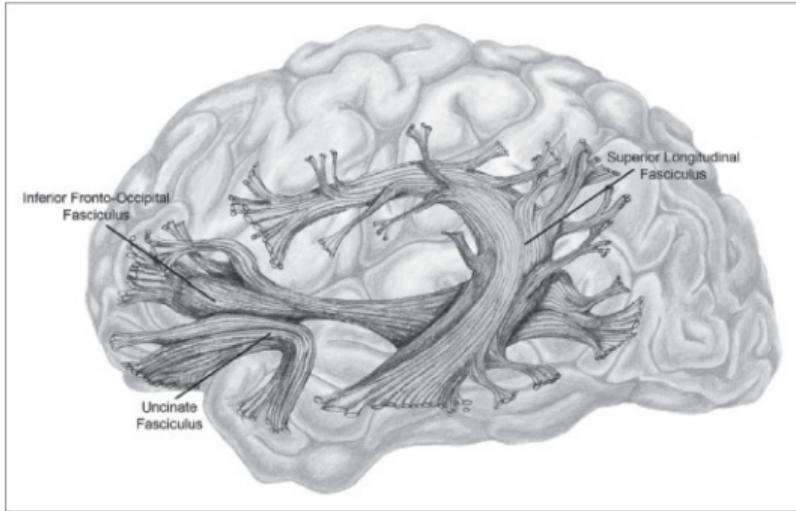
stiffer during loading than unloading (viscoelastic)

stiffer in compression than in tension (poroelastic)

stiffer in some regions than in others (heterogeneous)

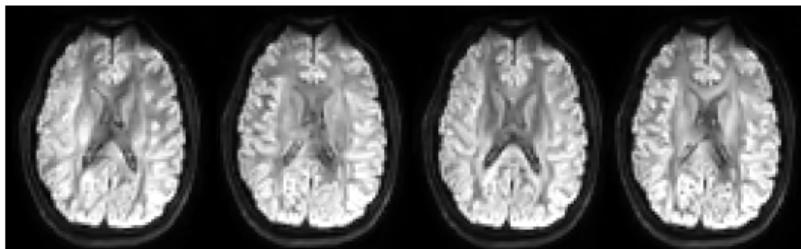


# White matter fiber tracts induce anisotropy in the brain

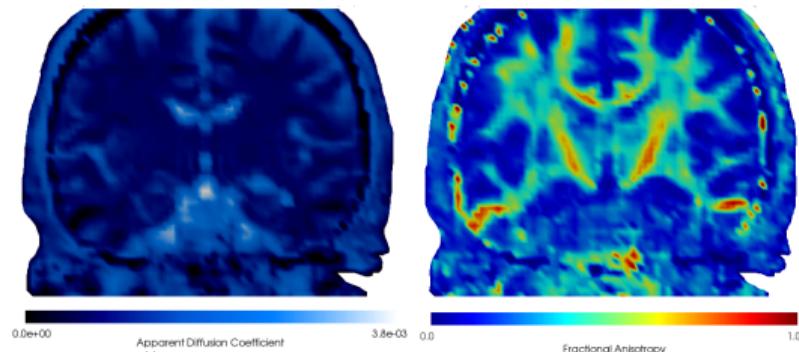


[Left: Wikimedia Commons. Right: Image credit: Alfred Pasieka]

# Brain tissue diffusion is anisotropic and can be measured with diffusion tensor imaging



Axial DTI slices measured with different b-vectors. The resolution in the diffusion tensor image is typically lower (here, 96x96x50) compared to that in the T1 images (256x256x256).



Mean diffusivity (left) and fractional anisotropy (right)

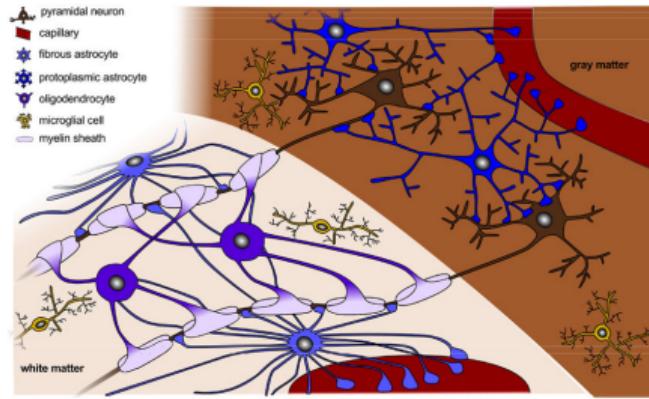
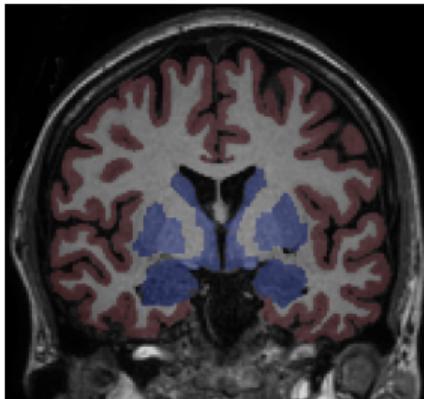
$$D = \begin{pmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{pmatrix}, \quad d_{ij} = d_{ij}(x).$$

D sym., pos. def, with eigenpairs  $(\lambda_i, v_i)$ .

$$\text{MD} = \frac{1}{3}(\lambda_1 + \lambda_2 + \lambda_3),$$

$$\text{FA}^2 = \frac{1}{2} \frac{(\lambda_1 - \lambda_2)^2 + (\lambda_2 - \lambda_3)^2 + (\lambda_3 - \lambda_1)^2}{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}.$$

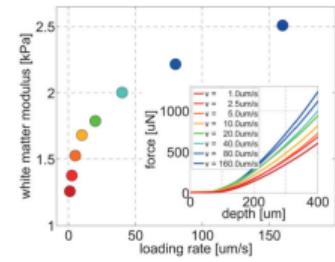
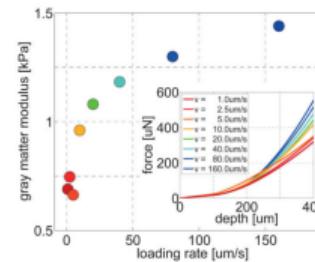
# Gray and white matter differs substantially in terms of composition and biophysical properties



## Example:

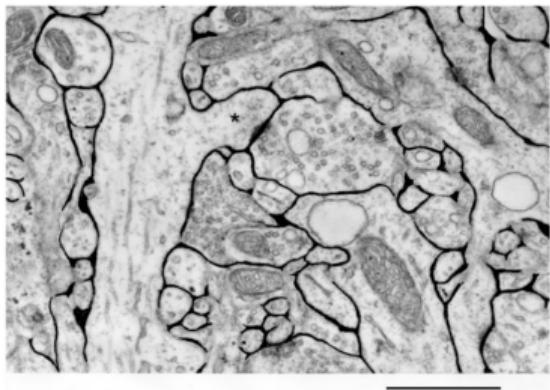
Heterogeneous stiffness (permeability, diffusion) tensor

$$K = K(x) = \begin{cases} K_g & x \in \Omega_g \text{ (in gray matter),} \\ K_w & x \in \Omega_w \text{ (in white matter).} \end{cases}$$



[Budday et al (2015) (Fig 6), Budday et al (2019) (Fig 2)]

# Solutes can diffuse in the narrow and tortuous extracellular spaces



Rat cerebral cortex with ECS in black (Scale bar:  $\approx 1\mu\text{m}$ )

[Nicholson (2001) (Fig. 2); Holter et al (2017) (Fig. 1)]

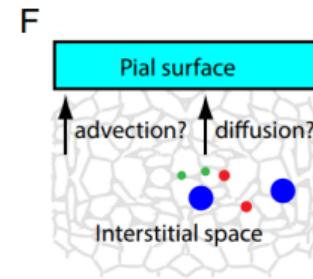
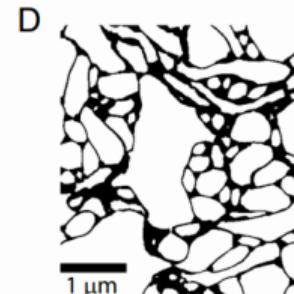
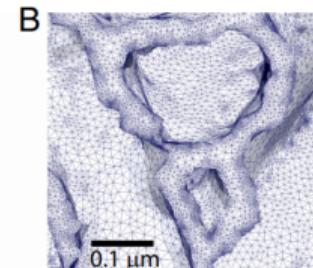
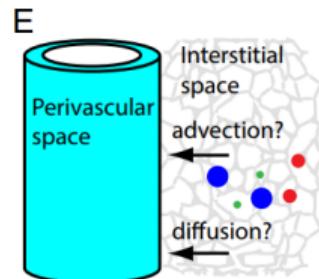
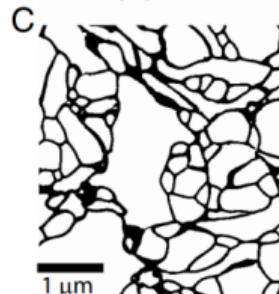
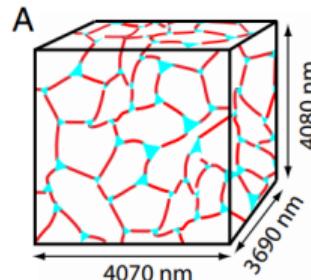
**ECS diffusion and tortuosity**  $D^* = D\lambda^{-2}$ :

Nicholson (2001):  $\lambda \approx 1.6$

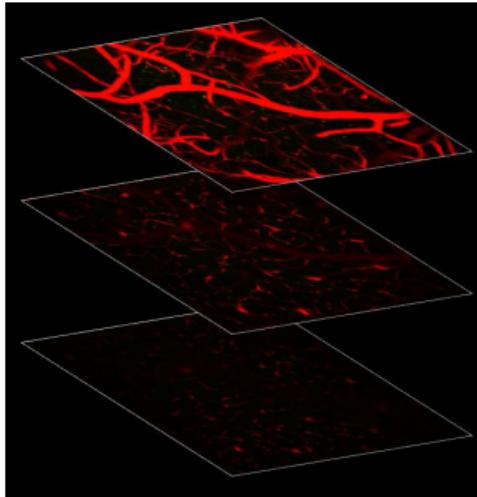
**ECS permeability** ( $\kappa$ ,  $\text{nm}^2$ ):

Holter et al (2017): 10-20

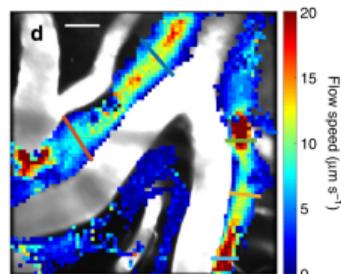
Basser (1992): 4000



# Fluid movement in perivascular spaces enhances solute transport



[Iliff et al (2012), Mestre et al (2018)]

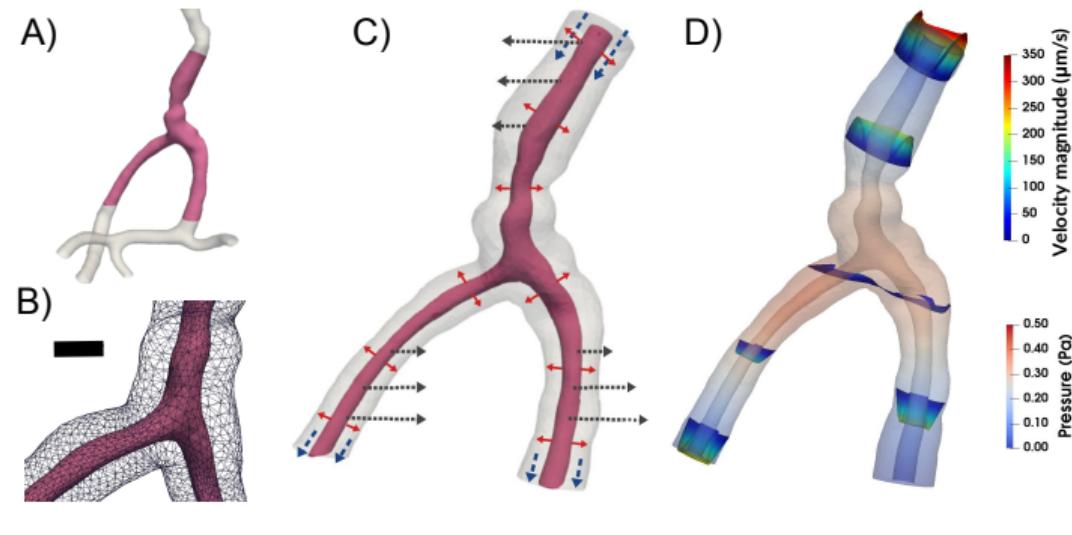


**Key mechanism** for brain solute transport:  
pulsatile flow of CSF/ISF in perivascular spaces

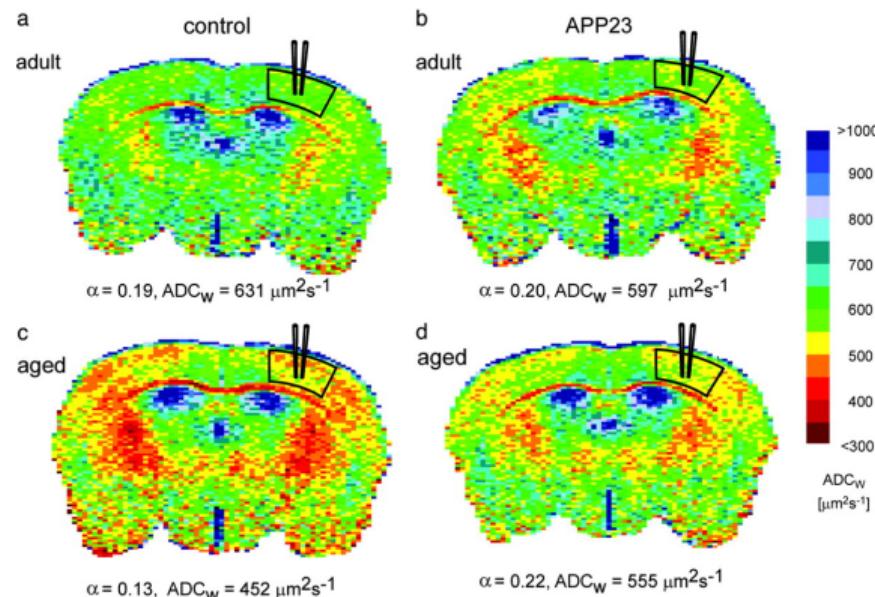
**Open questions:** Anatomical existence? Directionality?  
Magnitude? Importance?

[Rennels et al (1985), Ichimura et al (1991), Hadaczek et al (2005), Iliff et al (2012) ]

[Daversin-Catty et al (2020), Vinje et al (2021), Daversin-Catty et al (2022) ]

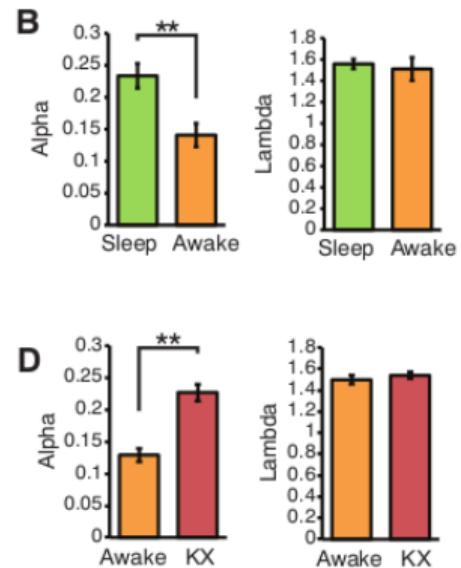


# Brain tissue is active and dynamic, and its properties change with circadian rhythm, age and pathologies



Volume fractions  $\alpha$  and apparent diffusion coefficients (ADC) in aging and Alzheimer's disease model mouse (APP23).

[Sykova et al (2005) (Fig 2)]

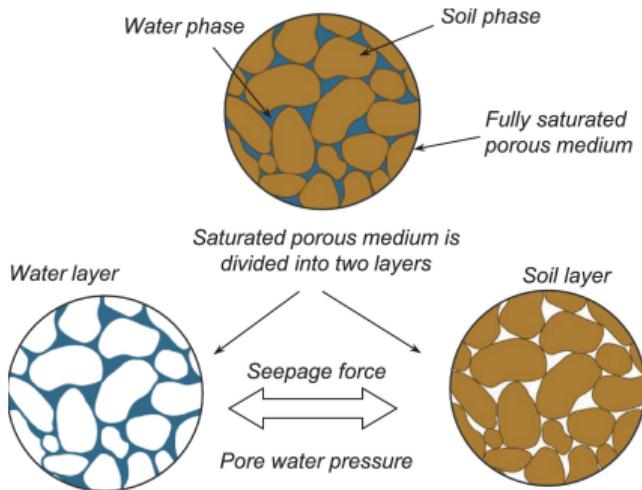


Volume fractions  $\alpha$  and tortuosity  $\lambda$  in sleeping, anaesthesized (KX) and awake mice.

[Xie et al., Science, 2013)]

The brain as a Stokes-Biot coupled problem

# Biot's equations describe displacement and fluid pressure in a porous and linearly elastic medium



[Bui and Nguyen (2017) [Fig. 1]]

Find the displacement  $u = u(x, t)$  and fluid network (or pore) pressure  $p = p(x, t)$  for  $x \in \Omega, t > 0$  such that

$$-\operatorname{div}(\sigma(u) - \alpha p \mathbf{I}) = f, \quad (1a)$$

$$\partial_t (sp + \alpha \operatorname{div} u) - \operatorname{div} \kappa \operatorname{grad} p = g. \quad (1b)$$

Linearly elastic stress-strain relationship:

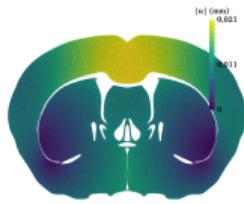
$$\sigma(u) = 2\mu \varepsilon(u) + \lambda \operatorname{div}(u)\mathbf{I}, \quad (2)$$

for Lamé parameters  $\mu > 0$ ,  $\lambda$  such that  $2\mu + d\lambda > 0$ .

$s \geq 0$  is the specific storage coefficient  
 $\alpha \in [0, 1]$  is the Biot-Willis coefficient  
 $\kappa > 0$  is the hydraulic conductivity.

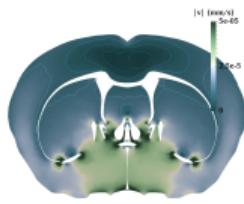
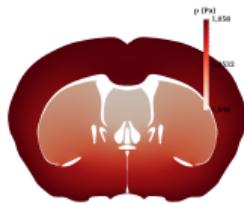
[Biot, J. Appl. Phys., 1941]

# In the limits of incompressibility (and impermeability), Biot's equations reduce to the elasticity & Darcy (and Stokes) equations

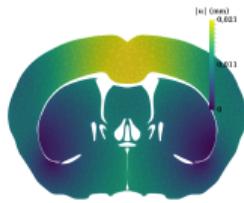


Find the displacement  $u$  and the pressure  $p$  such that :

$$\begin{aligned}-\operatorname{div}(2\mu\varepsilon(u) + \lambda \operatorname{div} u I - \alpha p I) &= f, \\ \partial_t(sp + \alpha \operatorname{div} u) - \operatorname{div} \kappa \operatorname{grad} p &= g.\end{aligned}$$



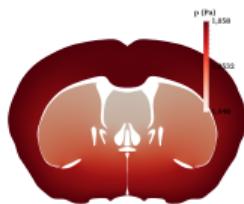
# In the limits of incompressibility (and impermeability), Biot's equations reduce to the elasticity & Darcy (and Stokes) equations



Find the displacement  $u$  and the pressure  $p$  such that :

$$-\operatorname{div}(2\mu\varepsilon(u) + \lambda \operatorname{div} u I - \alpha p I) = f,$$

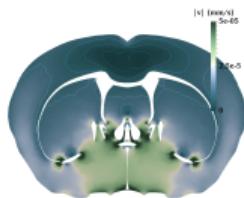
$$\partial_t(sp + \alpha \operatorname{div} u) - \operatorname{div} \kappa \operatorname{grad} p = g.$$



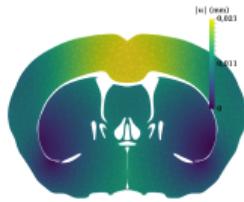
**Low-storage, incompressible regime:**  $s = 0$ ,  $\lambda \rightarrow \infty$ :  
 $\operatorname{div} u \rightarrow 0$ , the system decouples

$$-\operatorname{div} \kappa \operatorname{grad} p = g, \quad (\text{Darcy/Poisson})$$

$$-\operatorname{div}(2\mu\varepsilon(u) + \lambda \operatorname{div} u I) = f - \alpha \operatorname{grad} p. \quad (\text{Elasticity})$$

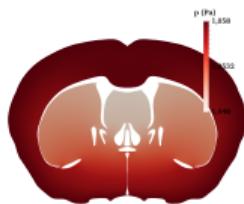


# In the limits of incompressibility (and impermeability), Biot's equations reduce to the elasticity & Darcy (and Stokes) equations



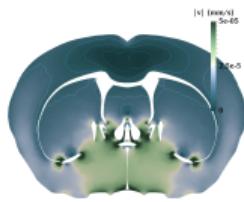
Find the displacement  $u$  and the pressure  $p$  such that :

$$\begin{aligned}-\operatorname{div}(2\mu\varepsilon(u) + \lambda \operatorname{div} u I - \alpha p I) &= f, \\ \partial_t(sp + \alpha \operatorname{div} u) - \operatorname{div} \kappa \operatorname{grad} p &= g.\end{aligned}$$



**Low-storage, incompressible regime:**  $s = 0, \lambda \rightarrow \infty$ :  
 $\operatorname{div} u \rightarrow 0$ , the system decouples

$$\begin{aligned}-\operatorname{div} \kappa \operatorname{grad} p &= g, && \text{(Darcy/Poisson)} \\ -\operatorname{div}(2\mu\varepsilon(u) + \lambda \operatorname{div} u I) &= f - \alpha \operatorname{grad} p. && \text{(Elasticity)}$$



**Low-storage, impermeable regime:**  $s = 0, \kappa \rightarrow 0$ :

$$\begin{aligned}-\operatorname{div}(2\mu\varepsilon(u) - \alpha p I) &= f, \\ \operatorname{div} u &= 0. && \text{(Stokes)}\end{aligned}$$



# Derivation of a finite difference in time / finite element discretization in space of Biot's equations

---

Consider Biot's equations : find  $u(t) : \Omega \rightarrow \mathbb{R}^d$  and  $p(t) : \Omega \rightarrow \mathbb{R}$  over a domain  $\Omega \subseteq \mathbb{R}^d$  and for  $t \in [0, T]$

$$-\operatorname{div}(\sigma(u) - \alpha p \mathbf{I}) = f \quad (\text{la})$$

$$\partial_t(s p + \alpha \operatorname{div} u) - \operatorname{div} K \operatorname{grad} p = g \quad (\text{lb})$$

$$\Omega \subseteq \mathbb{R}^d \text{ e.g } d=3$$

## Discretization in time:

Q: What would you do?

Partition our time interval  $(0, T)$  into timesteps  $\{t^n\}_{n=1}^N$



Denote the timestep  $(\Delta t)_n = t^n - t^{n-1}$  for  $n=1, \dots, N$ .

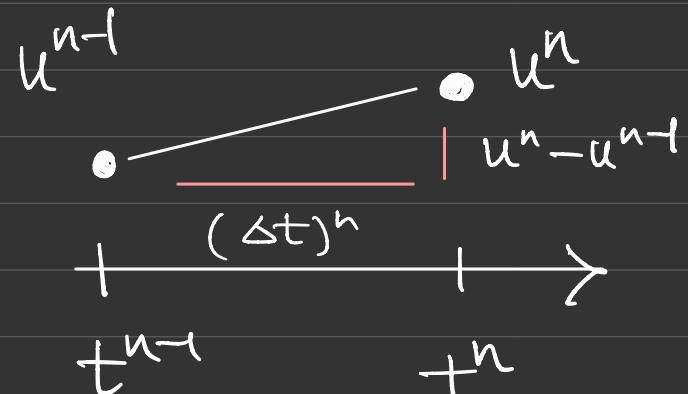
Note: Taking all time steps equal gives a uniform timestep method. Conversely, different time steps  $\rightarrow$  non-uniform or adaptive method.

## Discretization in time (cont.)

Introduce the approximations

$$u(t^n) \sim u^n \quad \text{for } n=0, \dots, N$$

$$p(t^n) \sim p^n \quad \text{for } n=0, \dots, N$$



and

$$\partial_t u(t^n) \sim \frac{u^n - u^{n-1}}{(\Delta t)_n} \quad n=1, \dots, N$$

$$\partial_t p(t^n) \sim \frac{p^n - p^{n-1}}{(\Delta t)_n}$$

Note: This is an implicit Euler discretization in time.  
Alternatives would be explicit Euler, implicit midpoint, or more elaborate splitting schemes. (3)

## Discretization in time (cont)

With these approximations, we obtain: for  $n=1, \dots, N$ ,  
find  $u^n : \Omega \rightarrow \mathbb{R}^d$  and  $\varphi^n : \Omega \rightarrow \mathbb{R}$  such that

$$-\operatorname{div} [\sigma(u^n) - \alpha \varphi^n \mathbf{I}] = f^n \equiv f(t^n) \quad (2a)$$

$$\frac{1}{(\Delta t)_n} (S p^n + \alpha \operatorname{div} u^n - S p^{n-1} - \alpha \operatorname{div} u^{n-1}) - \operatorname{div} K \operatorname{grad} \varphi^n = g^n$$

where after reordering, the 2<sup>nd</sup> equation, we obtain

$$S p^n + \alpha \operatorname{div} u^n - (\Delta t)_n \operatorname{div} K \operatorname{grad} \varphi^n = g^n + S p^{n-1} + \alpha \operatorname{div} u^{n-1} \quad (2b)$$

at each time step  $n=1, \dots, N$ .

## A note on signs

[Write u for  $u^n$  etc]

Original form (physical pressure, inwards force)

$$-\operatorname{div} \sigma(u) + \alpha \operatorname{grad} p = f$$

$$sp + \alpha \operatorname{div} u - (\Delta t) \operatorname{div} k \operatorname{grad} p = l$$

Later, we write integrate these terms by parts, want the result to have same sign. Therefore map  $p \mapsto -p$  to obtain the symmetric form:

$$-\operatorname{div} \sigma(u) - \alpha \operatorname{grad} p = f \quad (3a)$$

$$-sp + \alpha \operatorname{div} u + (\Delta t) \operatorname{div} k \operatorname{grad} p = l \quad (3b)$$

⑥

## Variational formulation of the time-discrete problem

Introduce the function spaces

$$V = H^{\frac{1}{2}}(\Omega; \mathbb{R}^d) = \{v: \Omega \rightarrow \mathbb{R}^d, v_i \in H_0^{\frac{1}{2}}(\Omega)\}$$

$$Q = H_0^{\frac{1}{2}}(\Omega) = \{q: \Omega \rightarrow \mathbb{R}, q \in L^2(\Omega),$$

" $q|_{\partial\Omega} = 0$ ", and  $\operatorname{grad} q \in L^2(\Omega)$  in the sense of weak derivatives}

Multiply (3a) by a test function  $v \in V$ , integrate,

and integrate by parts

$$\int_{\Omega} \sigma(u) : \epsilon(v) \, dx - \int_{\Omega} \alpha p \cdot \operatorname{div} v \, dx = \int_{\Omega} f \cdot v \, dx$$

where  $\epsilon(v) = \operatorname{Sym} \operatorname{grad} v = \frac{1}{2} (\operatorname{grad} v + \operatorname{grad} v^T)$

(cont.)

Similarly, multiply (3b) by a test function

$q \in Q$ , integrate and integrate last term by parts:

$$\int_{\Omega} s \cdot p \cdot q \, dx + \int_{\Omega} \alpha \operatorname{div} u \cdot q \, dx + \int_{\Omega} K \operatorname{grad} p \cdot \operatorname{grad} q \, dx \\ = \int_{\Omega} \ell \cdot q \, dx$$

A note on boundary conditions:

We have introduced

$$u|_{\partial\Omega} = 0 \quad \text{and} \quad p|_{\partial\Omega} = 0$$

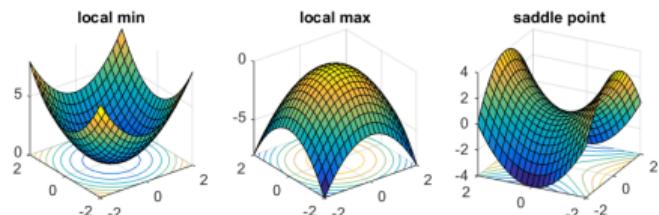
# Abstract saddle point problems and stability

## Abstract continuous form

Find  $u \in V$  and  $p \in Q$  such that

$$\begin{aligned} a(u, v) + b(v, p) &= \langle f, v \rangle \quad \forall v \in V, \\ b(u, q) - c(p, q) &= \langle g, q \rangle \quad \forall q \in Q. \end{aligned}$$

$$\begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}$$



## Example: (Time-discrete) Biot equations

Define  $V = H_0^1(\Omega; \mathbb{R}^d)$ ,  $Q = L^2(\Omega)$  and

$$\begin{aligned} a(u, v) &= \langle 2\mu\varepsilon(u), \varepsilon(v) \rangle + \langle \lambda \operatorname{div} u, \operatorname{div} v \rangle \\ b(u, p) &= \langle \alpha \operatorname{div} u, p \rangle \\ c(p, q) &= \langle sp, q \rangle + \langle \kappa \operatorname{grad} p, \operatorname{grad} q \rangle \end{aligned}$$

(after  $p \mapsto -p$  for the sake of symmetry)

[Ladyzhenskaya (1969), Babuška (1972), Brezzi (1974)]

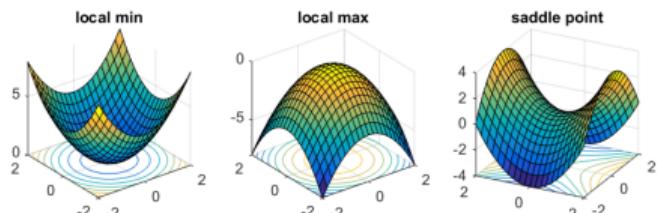
# Abstract saddle point problems and stability

## Abstract continuous form

Find  $u \in V$  and  $p \in Q$  such that

$$\begin{aligned} a(u, v) + b(v, p) &= \langle f, v \rangle \quad \forall v \in V, \\ b(u, q) - c(p, q) &= \langle g, q \rangle \quad \forall q \in Q. \end{aligned}$$

$$\begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix}$$



[Ladyzhenskaya (1969), Babuška (1972), Brezzi (1974)]

## Example: (Time-discrete) Biot equations

Define  $V = H_0^1(\Omega; \mathbb{R}^d)$ ,  $Q = L^2(\Omega)$  and

$$\begin{aligned} a(u, v) &= \langle 2\mu\varepsilon(u), \varepsilon(v) \rangle + \langle \lambda \operatorname{div} u, \operatorname{div} v \rangle \\ b(u, p) &= \langle \alpha \operatorname{div} u, p \rangle \\ c(p, q) &= \langle sp, q \rangle + \langle \kappa \operatorname{grad} p, \operatorname{grad} q \rangle \end{aligned}$$

(after  $p \mapsto -p$  for the sake of symmetry)

## Stability conditions (continuous form)

Well-posed problem if  $a, b, c$  continuous,  $c$  pos. def., and  $\exists \alpha, \beta > 0$

$$a(u, u) \geq \alpha \|u\|_V^2 \quad \forall u \in \ker B,$$

$$\inf_{q \in Q} \sup_{v \in V} |b(v, q)| \|v\|_V^{-1} \|q\|_Q^{-1} \geq \beta.$$

# Discretization of abstract saddle point problems

Introduce discrete spaces  $V_h$  and  $Q_h$ ,  
typically relative to an admissible  
triangulation  $\mathcal{T}_h$  of  $\Omega$ .

If  $V_h \subset V$  and  $Q_h \subset Q$ , the discretization is  
**conforming**.

## Abstract (conforming) discrete form

Find  $u_h \in V_h$  and  $p_h \in Q_h$  such that

$$a(u_h, v) + b(v, p_h) = \langle f, v \rangle \quad \forall v \in V_h,$$

$$b(u_h, q) - c(p_h, q) = \langle g, q \rangle \quad \forall q \in Q_h.$$

[Babuška (1972), Brezzi (1974)]

# Discretization of abstract saddle point problems

Introduce discrete spaces  $V_h$  and  $Q_h$ , typically relative to an admissible triangulation  $\mathcal{T}_h$  of  $\Omega$ .

If  $V_h \subset V$  and  $Q_h \subset Q$ , the discretization is **conforming**.

## Abstract (conforming) discrete form

Find  $u_h \in V_h$  and  $p_h \in Q_h$  such that

$$a(u_h, v) + b(v, p_h) = \langle f, v \rangle \quad \forall v \in V_h,$$

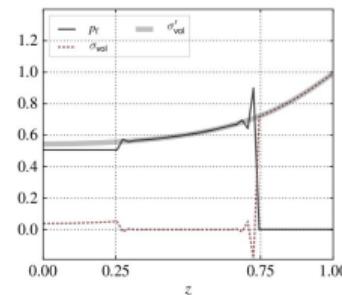
$$b(u_h, q) - c(p_h, q) = \langle g, q \rangle \quad \forall q \in Q_h.$$

[Babuška (1972), Brezzi (1974)]

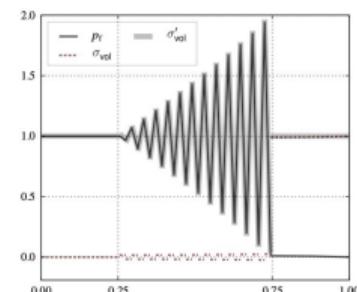
## Discrete stability conditions

The discrete problem is well-posed if  $a, b, c$  continuous,  $c$  pos. semi-def., and  $\exists \alpha, \beta > 0$  independent of  $h$  such that

$$a(u, u) \geq \alpha \|u\|_V^2 \quad \forall u \in \ker B_h,$$
$$\inf_{q \in Q_h} \sup_{v \in V_h} |b(v, q)| \|v\|_V^{-1} \|q\|_Q^{-1} \geq \beta.$$



(a) Three-layer problem using  $Q_2/Q_1$  and  $\epsilon = 10^{-8}$



(b) Oscillatory solution to the two-material problem with uniform load;  $F = 1$ ,  $\epsilon = 10^{-8}$ ,  $\Delta t = 1$ .

[Haga et al (2011)]

# FEniCS implementation of Biot discrete forms

```
from dolfin import *

# Define mesh
n = 8
mesh = UnitSquareMesh(n, n)
d = mesh.topology().dim()

# Define VxQ (mixed) element
cell = mesh.ufl_cell()
V = VectorElement("CG", cell, 2)
Q = FiniteElement("CG", cell, 1)
M = MixedElement([V, Q])

# Define mixed function space,
# test- and trial functions
W = FunctionSpace(mesh, M)
(u, p) = TrialFunctions(W)
(v, q) = TestFunctions(W)

[...]
```

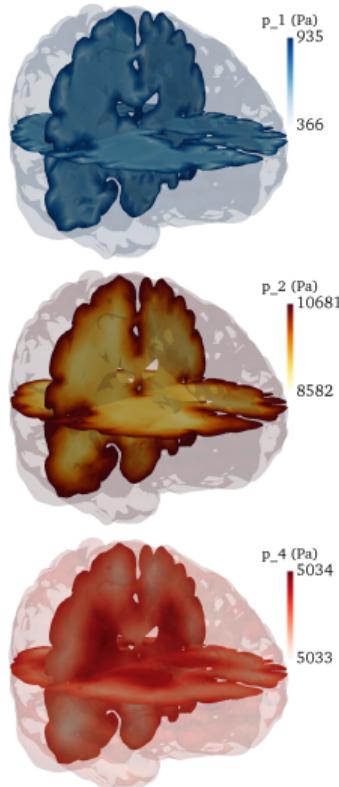
```
# Previous solutions
up_ = Function(W)
(u_, p_) = split(up_)

# Define strain and stress
eps = lambda u : sym(grad(u))
I = Identity(d)
sigma = lambda u : \
    2*mu*eps(u) + lmbda*div(u)*I

# Define variational forms
a = (inner(sigma(u), eps(v)) +
      alpha*div(v)*p +
      alpha*div(u)*q -
      s*p*q -
      inner(kappa*grad(p), grad(q)) *
      dx())
L = (dot(f, v) + g*q -
      s*p_*q + alpha*div(u_)*q)*dx()

# Assemble
A = assemble(a)
b = assemble(L)
```

# The multiple-network poroelasticity (MPET) equations describe displacement and fluid pressures in generalized poroelastic media



Find the displacement  $u = u(x, t)$  and  $J$  (network) pressures  $p_j = p_j(x, t)$  for  $j = 1, \dots, J$  such that

$$-\operatorname{div}(\sigma(u) - \sum_j \alpha_j p_j \mathbf{I}) = f, \quad (4a)$$

$$\partial_t (s_j p_j + \alpha_j \operatorname{div} u) - \operatorname{div} \kappa_j \operatorname{grad} p_j + T_j = g_j \quad j = 1, \dots, J. \quad (4b)$$

Fluid exchange between networks:

$$T_j = \sum_i T_{i \leftarrow j} = \sum_i \gamma_{ji} (p_j - p_i).$$

$J = 1$  is Biot's equations,  $J = 2$  is Barenblatt-Biot.  $\lambda \rightarrow \infty, s_j \rightarrow 0$ ,  $\gamma_{ji} = \gamma_{ij}$ ,  $\gamma_{ji} \gg 1, \gamma_{ji} \ll 1$  and  $\kappa_j \rightarrow 0$  interesting regimes.

[Biot (1941), Bai, Elsworth and Roegiers (1993)]

[Tully and Ventikos (2011)]

[Lee et al (2019), Piersanti et al (2021), Hong et al (2022), Eliseussen et al (2022)]

# The Ern-Meunier framework for coupled elliptic-parabolic problems

[Ern and Meunier (2009)]

## Coupled elliptic-parabolic problem

Find  $u \in H^1(0, T; V_a)$ ,  $p \in H^1(0, T; V_d)$  s.t.

$$a(u, v) - b(v, p) = \langle f, v \rangle, \quad (5a)$$

$$c(\dot{p}, q) + b(\dot{u}, q) + d(p, q) = \langle g, q \rangle, \quad (5b)$$

for all  $v \in V_a$  and  $q \in V_d$ .

# The Ern-Meunier framework for coupled elliptic-parabolic problems

[Ern and Meunier (2009)]

## Coupled elliptic-parabolic problem

Find  $u \in H^1(0, T; V_a)$ ,  $p \in H^1(0, T; V_d)$  s.t.

$$a(u, v) - b(v, p) = \langle f, v \rangle, \quad (5a)$$

$$c(\dot{p}, q) + b(\dot{u}, q) + d(p, q) = \langle g, q \rangle, \quad (5b)$$

for all  $v \in V_a$  and  $q \in V_d$ .

## Example: Biot

$V_a = H_0^1(\Omega; \mathbb{R}^d)$ ,  $V_d = H_0^1$ .

$$a(u, v) = \langle \sigma(u), \varepsilon(v) \rangle,$$

$$b(u, p) = \langle \alpha p, \operatorname{div} u \rangle,$$

$$c(p, q) = \langle s p, q \rangle,$$

$$d(p, q) = \langle \kappa \operatorname{grad} p, \operatorname{grad} q \rangle,$$

$L_a = L^2(\Omega; \mathbb{R}^d)$ ,  $L_d = L^2$ .

# The Ern-Meunier framework for coupled elliptic-parabolic problems

[Ern and Meunier (2009)]

## Coupled elliptic-parabolic problem

Find  $u \in H^1(0, T; V_a)$ ,  $p \in H^1(0, T; V_d)$  s.t.

$$a(u, v) - b(v, p) = \langle f, v \rangle, \quad (5a)$$

$$c(\dot{p}, q) + b(\dot{u}, q) + d(p, q) = \langle g, q \rangle, \quad (5b)$$

for all  $v \in V_a$  and  $q \in V_d$ .

## Example: Biot

$V_a = H_0^1(\Omega; \mathbb{R}^d)$ ,  $V_d = H_0^1$ .

$$a(u, v) = \langle \sigma(u), \varepsilon(v) \rangle,$$

$$b(u, p) = \langle \alpha p, \operatorname{div} u \rangle,$$

$$c(p, q) = \langle s p, q \rangle,$$

$$d(p, q) = \langle \kappa \operatorname{grad} p, \operatorname{grad} q \rangle,$$

$L_a = L^2(\Omega; \mathbb{R}^d)$ ,  $L_d = L^2$ .

## Assumptions

- A1)  $V_a, V_d$  are Hilbert spaces. There also exist Hilbert spaces  $L_a, L_d$  such that  $V_a, V_d$  are dense in  $L_a, L_d$ , and  $\|v\|_{L_a} \lesssim \|v\|_{V_a}$ ,  $\|q\|_{L_d} \lesssim \|q\|_{V_d}$ .
- A2)  $a, c : L_d \times L_d \rightarrow \mathbb{R}$ , and  $d$  are **symmetric**, **coercive** and **continuous** bilinear forms (inducing norms  $\|\cdot\|_a$  etc).
- A3)  $b : V_a \times L_d \rightarrow \mathbb{R}$  is continuous and such that  $|b(v, q)| \lesssim \|v\|_a \|q\|_c$ .

# The Ern-Meunier framework for coupled elliptic-parabolic problems

[Ern and Meunier (2009)]

## Coupled elliptic-parabolic problem

Find  $u \in H^1(0, T; V_a)$ ,  $p \in H^1(0, T; V_d)$  s.t.

$$a(u, v) - b(v, p) = \langle f, v \rangle, \quad (5a)$$

$$c(\dot{p}, q) + b(\dot{u}, q) + d(p, q) = \langle g, q \rangle, \quad (5b)$$

for all  $v \in V_a$  and  $q \in V_d$ .

## Example: Biot

$V_a = H_0^1(\Omega; \mathbb{R}^d)$ ,  $V_d = H_0^1$ .

$$a(u, v) = \langle \sigma(u), \varepsilon(v) \rangle,$$

$$b(u, p) = \langle \alpha p, \operatorname{div} u \rangle,$$

$$c(p, q) = \langle s p, q \rangle,$$

$$d(p, q) = \langle \kappa \operatorname{grad} p, \operatorname{grad} q \rangle,$$

$L_a = L^2(\Omega; \mathbb{R}^d)$ ,  $L_d = L^2$ .

## Assumptions

- A1)  $V_a, V_d$  are Hilbert spaces. There also exist Hilbert spaces  $L_a, L_d$  such that  $V_a, V_d$  are dense in  $L_a, L_d$ , and  $\|v\|_{L_a} \lesssim \|v\|_{V_a}$ ,  $\|q\|_{L_d} \lesssim \|q\|_{V_d}$ .
- A2)  $a, c : L_d \times L_d \rightarrow \mathbb{R}$ , and  $d$  are **symmetric**, **coercive** and **continuous** bilinear forms (inducing norms  $\|\cdot\|_a$  etc).
- A3)  $b : V_a \times L_d \rightarrow \mathbb{R}$  is continuous and such that  $|b(v, q)| \lesssim \|v\|_a \|q\|_c$ .

**Stability** and **uniqueness** follows

$$\begin{aligned} \|u\|_{L^\infty(0, T; V_a)}^2 + \|p\|_{L^\infty(0, T; L_d)}^2 + \|p\|_{L^2(0, T; V_d)}^2 \\ \lesssim \|(f, \dot{f}, g, u_0, p_0)\| \end{aligned}$$

## Space- and time discretization of coupled elliptic-parabolic problems

For meshes  $\mathcal{T}_h$  of  $\Omega$ , and time points

$0 < t^1 < \dots < t^N = T$  with  $\tau^n = t^n - t^{n-1}$ ,

define the time differential

$$\delta_t u_h^n = (\tau^n)^{-1} (u_h^n - u_h^{n-1})$$

Also, write  $u_{h\tau}$  as the continuous piecewise linear-in-time interpolant of  $u_h^n$ .

# Space- and time discretization of coupled elliptic-parabolic problems

For meshes  $\mathcal{T}_h$  of  $\Omega$ , and time points

$0 < t^1 < \dots < t^N = T$  with  $\tau^n = t^n - t^{n-1}$ ,

define the time differential

$$\delta_t u_h^n = (\tau^n)^{-1} (u_h^n - u_h^{n-1})$$

Also, write  $u_{h\tau}$  as the continuous piecewise linear-in-time interpolant of  $u_h^n$ .

## Discrete elliptic-parabolic problem

For  $n = 1, 2, \dots, N$ , find  $u_h^n \in V_{a,h}$  and  $p_h^n \in V_{d,h}$  such that

$$a(u_h^n, v) - b(v, p_h^n) = \langle f_h^n, v \rangle$$

$$c(\delta_t p_h^n, q) + b(\delta_t u_h^n, q) + d(p_h^n, q) = \langle g, q \rangle$$

for all  $v \in V_{a,h}$ ,  $q \in V_{d,h}$ .

# Space- and time discretization of coupled elliptic-parabolic problems

For meshes  $\mathcal{T}_h$  of  $\Omega$ , and time points  
 $0 < t^1 < \dots < t^N = T$  with  $\tau^n = t^n - t^{n-1}$ ,  
define the time differential

$$\delta_t u_h^n = (\tau^n)^{-1} (u_h^n - u_h^{n-1})$$

Also, write  $u_{h\tau}$  as the continuous piecewise linear-in-time interpolant of  $u_h^n$ .

## Discrete elliptic-parabolic problem

For  $n = 1, 2, \dots, N$ , find  $u_h^n \in V_{a,h}$  and  
 $p_h^n \in V_{d,h}$  such that

$$a(u_h^n, v) - b(v, p_h^n) = \langle f_h^n, v \rangle$$

$$c(\delta_t p_h^n, q) + b(\delta_t u_h^n, q) + d(p_h^n, q) = \langle g, q \rangle$$

for all  $v \in V_{a,h}$ ,  $q \in V_{d,h}$ .

Under a set of additional hypotheses:

- ◊ approximability of regular solutions,
- ◊ well-posed dual problem,
- ◊ finite element space compatibility,

A-priori and a-posteriori error estimates follow for the general coupled problem, and a-fortiori for examples (e.g. Biot, MPET).

# Space- and time discretization of coupled elliptic-parabolic problems

For meshes  $\mathcal{T}_h$  of  $\Omega$ , and time points

$0 < t^1 < \dots < t^N = T$  with  $\tau^n = t^n - t^{n-1}$ ,  
define the time differential

$$\delta_t u_h^n = (\tau^n)^{-1} (u_h^n - u_h^{n-1})$$

Also, write  $u_{h\tau}$  as the continuous piecewise linear-in-time interpolant of  $u_h^n$ .

## Discrete elliptic-parabolic problem

For  $n = 1, 2, \dots, N$ , find  $u_h^n \in V_{a,h}$  and  
 $p_h^n \in V_{d,h}$  such that

$$\begin{aligned} a(u_h^n, v) - b(v, p_h^n) &= \langle f_h^n, v \rangle \\ c(\delta_t p_h^n, q) + b(\delta_t u_h^n, q) + d(p_h^n, q) &= \langle g, q \rangle \end{aligned}$$

for all  $v \in V_{a,h}$ ,  $q \in V_{d,h}$ .

Under a set of additional hypotheses:

- ◊ approximability of regular solutions,
- ◊ well-posed dual problem,
- ◊ finite element space compatibility,

A-priori and a-posteriori error estimates follow for the general coupled problem, and a-fortiori for examples (e.g. Biot, MPET).

## A posteriori estimate

If  $(u, p)$  are continuous and  $(u_{h\tau}, p_{h\tau})$  discrete (Taylor–Hood-type) MPET solutions, then

$$\begin{aligned} \|u - u_{h\tau}\|_{L^\infty(0,T;H^1)} + \|p - p_{h\tau}\|_{L^\infty(0,T;L^2) \cap L^2(0,T;H^1)} \\ \lesssim \eta_1 + \eta_2 + \eta_3 + \eta_4 + \mathcal{E}_h(f, g) \end{aligned}$$

where  $\eta_1, \eta_2, \eta_3, \eta_4$  are computable.

## Simulating brain pulsatility

# Intracranial dynamics result from an interplay between arterial blood influx, cerebrospinal fluid flow, venous outflux, and compliances

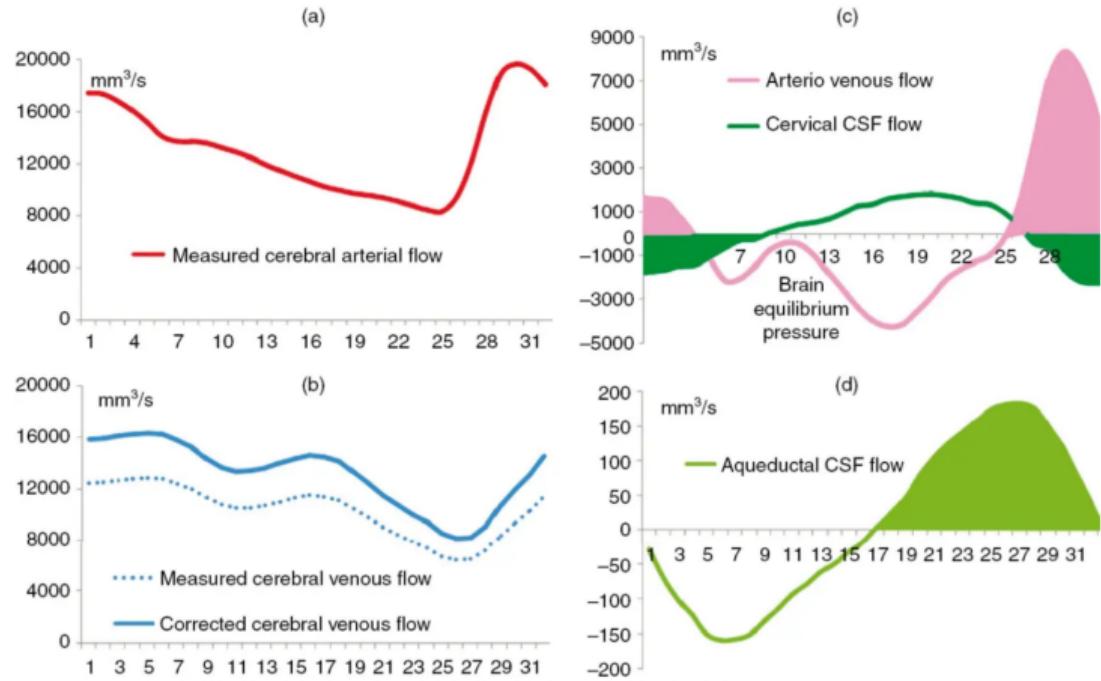
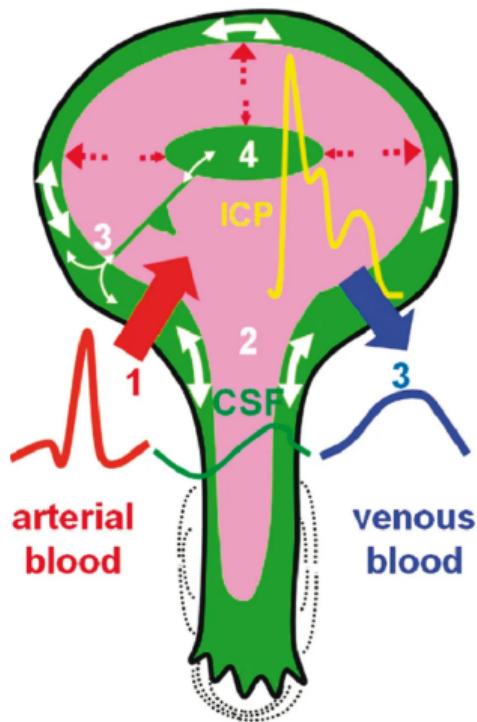
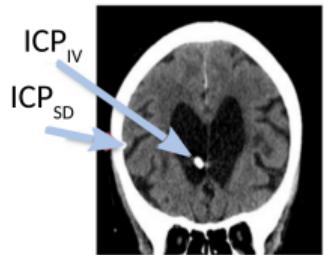
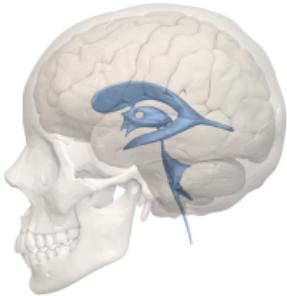


Figure 12.4 CSF and cerebral blood flow during the cardiac cycle in healthy adults (see text for full explanation).

# ICP (gradients) pulsate in sync with cardiac and respiratory cycles

[Vinje et al, Respiratory influence on cerebrospinal fluid flow..., Scientific Reports, 2019]

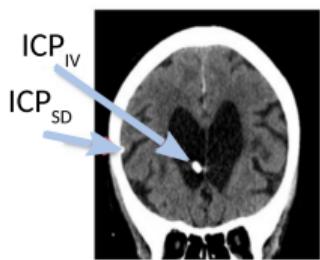
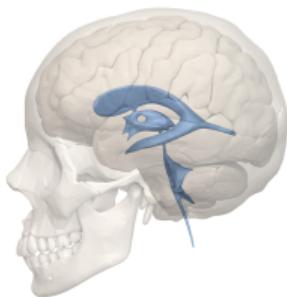


Long term ICP measurements

[Eide and Sæhle, 2010]

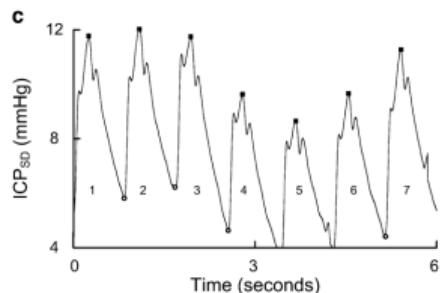
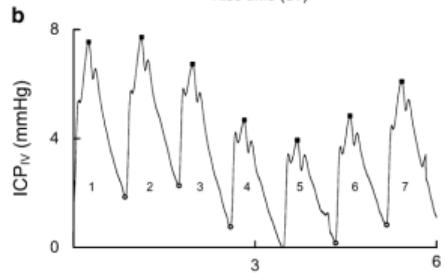
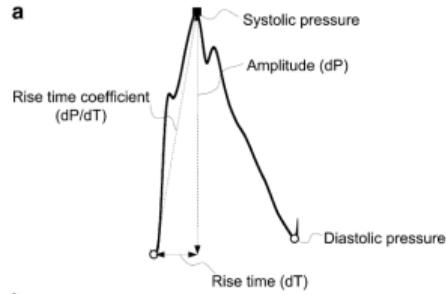
# ICP (gradients) pulsate in sync with cardiac and respiratory cycles

[Vinje et al, Respiratory influence on cerebrospinal fluid flow..., Scientific Reports, 2019]



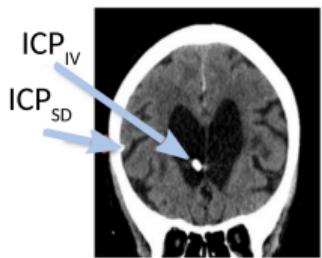
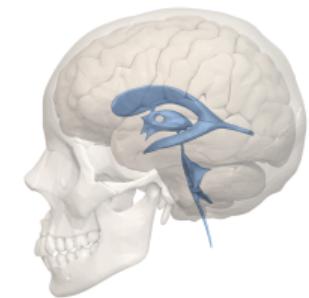
Long term ICP measurements

[Eide and Sæhle, 2010]



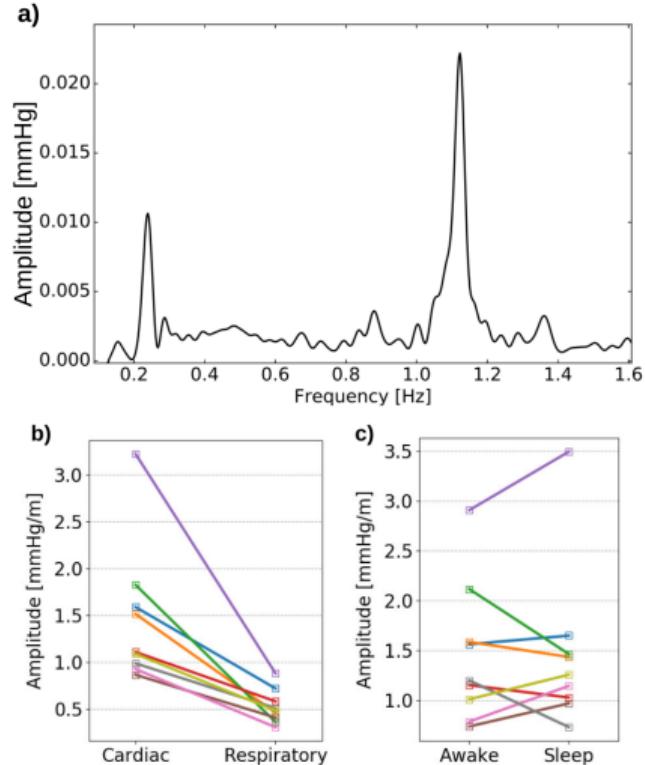
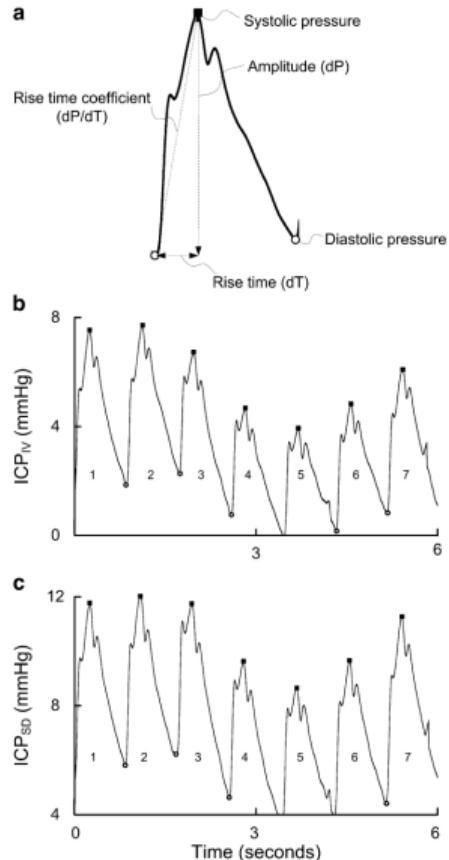
# ICP (gradients) pulsate in sync with cardiac and respiratory cycles

[Vinje et al, Respiratory influence on cerebrospinal fluid flow..., Scientific Reports, 2019]



Long term ICP measurements

[Eide and Sæhle, 2010]



$$d\text{ICP}(t) \approx a_c \sin(2\pi f_{ct}) + a_r \sin(2\pi f_{rt})$$

# Pulsating ICP gradients (mmHg/m) induce pulsating CSF flow (mL/s)

[Vinje et al, Respiratory influence on cerebrospinal fluid flow..., Scientific Reports, 2019]

## Incompressible Navier-Stokes

Velocity  $v$  and pressure  $p$  such that

$$\rho(\dot{v} + v \cdot \nabla v) - \mu\Delta v + \text{grad } p = 0$$

$$\text{div } v = 0$$

Pressure given between inlet and outlet:

$$dp(t) = a_c \sin(2\pi f_c t) + a_r \sin(2\pi f_r t)$$

# Pulsating ICP gradients (mmHg/m) induce pulsating CSF flow (mL/s)

[Vinje et al, Respiratory influence on cerebrospinal fluid flow..., Scientific Reports, 2019]

## Incompressible Navier-Stokes

Velocity  $v$  and pressure  $p$  such that

$$\rho(\dot{v} + v \cdot \nabla v) - \mu \Delta v + \text{grad } p = 0$$
$$\text{div } v = 0$$

Pressure given between inlet and outlet:

$$dp(t) = a_c \sin(2\pi f_c t) + a_r \sin(2\pi f_r t)$$

## Analytic solution(s) in axisymmetric pipe

Peak flux  $A_r, A_c$  and stroke volume  $V_r, V_c$ :

$$A = |\pi r^2 \frac{ia}{\rho\omega} \left(1 - \frac{2}{\Lambda} \frac{J_1(\Lambda)}{J_0(\Lambda)}\right)|$$

$$V = A(\pi f)^{-1}$$

where  $\omega = 2\pi f$ ,  $\Lambda = \alpha i^{3/2}$ ,  $\alpha$  Womersley...

# Pulsating ICP gradients (mmHg/m) induce pulsating CSF flow (mL/s)

[Vinje et al, Respiratory influence on cerebrospinal fluid flow..., Scientific Reports, 2019]

## Incompressible Navier-Stokes

Velocity  $v$  and pressure  $p$  such that

$$\rho(\dot{v} + v \cdot \nabla v) - \mu \Delta v + \text{grad } p = 0$$
$$\text{div } v = 0$$

Pressure given between inlet and outlet:

$$dp(t) = a_c \sin(2\pi f_c t) + a_r \sin(2\pi f_r t)$$

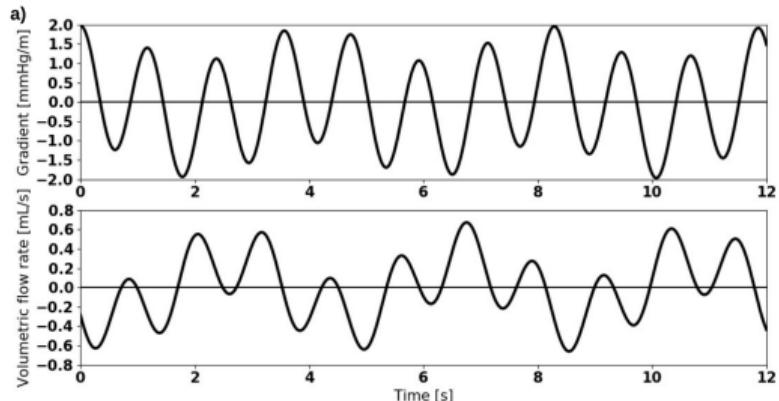
## Analytic solution(s) in axisymmetric pipe

Peak flux  $A_r, A_c$  and stroke volume  $V_r, V_c$ :

$$A = |\pi r^2 \frac{ia}{\rho\omega} \left(1 - \frac{2}{\Lambda} \frac{J_1(\Lambda)}{J_0(\Lambda)}\right)|$$

$$V = A(\pi f)^{-1}$$

where  $\omega = 2\pi f$ ,  $\Lambda = \alpha i^{3/2}$ ,  $\alpha$  Womersley...



# Pulsating ICP gradients (mmHg/m) induce pulsating CSF flow (mL/s)

[Vinje et al, Respiratory influence on cerebrospinal fluid flow..., Scientific Reports, 2019]

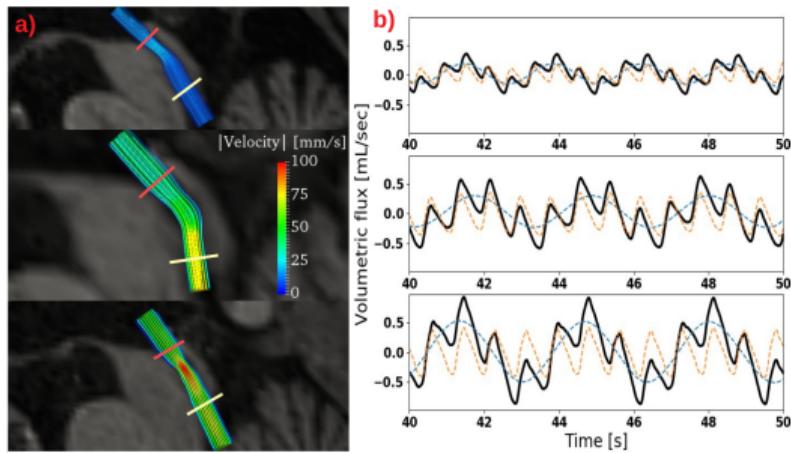
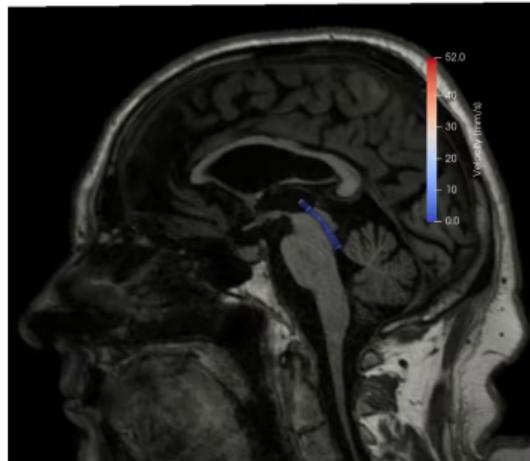
## Incompressible Navier-Stokes

Velocity  $v$  and pressure  $p$  such that

$$\rho(\dot{v} + v \cdot \nabla v) - \mu\Delta v + \text{grad } p = 0$$
$$\text{div } v = 0$$

Pressure given between inlet and outlet:

$$dp(t) = a_c \sin(2\pi f_c t) + a_r \sin(2\pi f_r t)$$



## In patients (cardiac vs respiratory)

- ◊ Average peak flow rates: 0.29 vs 0.32 mL/s
- ◊ Average stroke volumes: 70 mL vs 308 mL
- ◊ Good agreement with cardiac-gated PC-MRI
- ◊ Resolves clinical pressure vs flow mystery!

# Intracranial dynamics result from an interplay between arterial blood influx, cerebrospinal fluid flow, venous outflux, and compliances

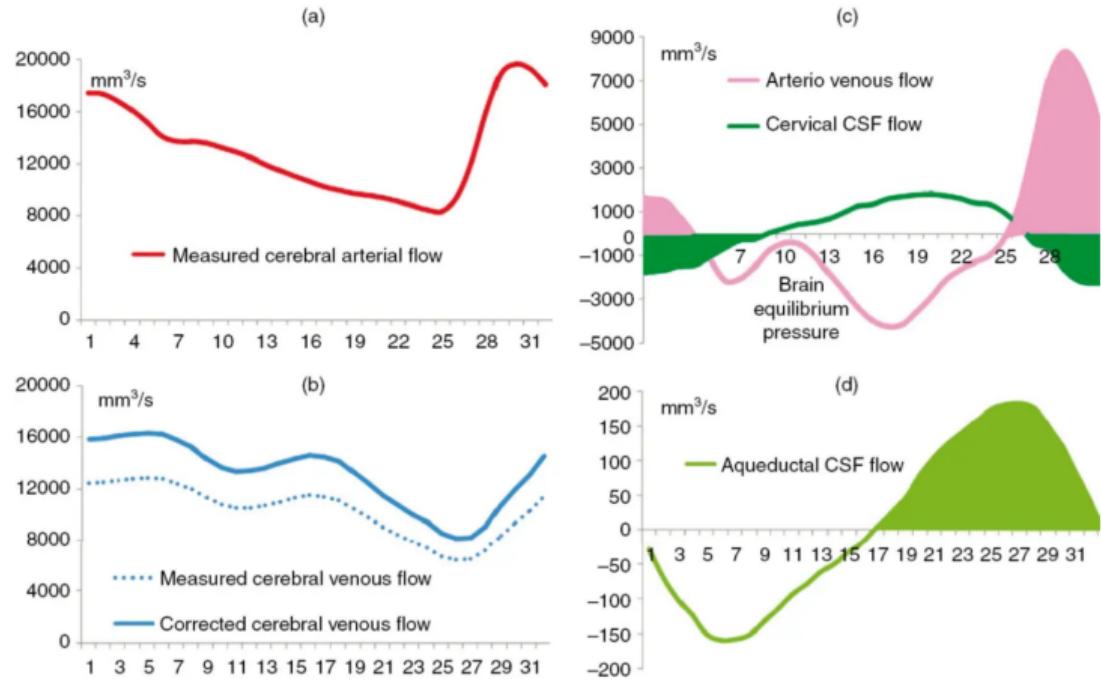
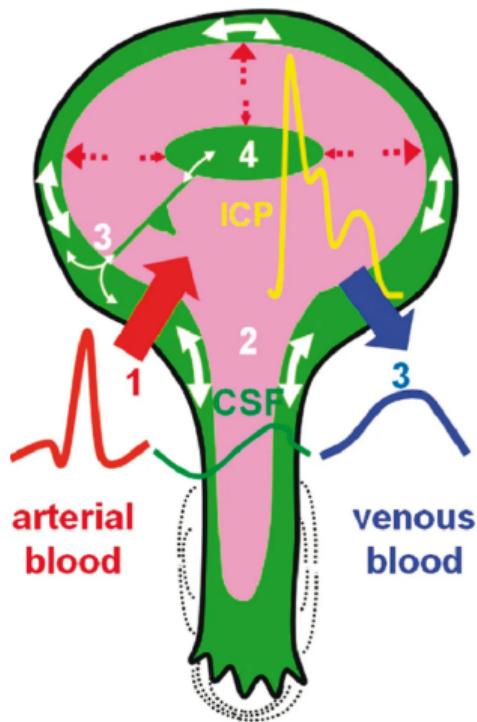


Figure 12.4 CSF and cerebral blood flow during the cardiac cycle in healthy adults (see text for full explanation).

# The brain and -environment as a coupled poroelastic-viscous system

[Causemann et al, Human intracranial pulsatility..., FBCNS, 2022]

## Biot's equations in $\Omega_P$

Find the displacement  $u = u(x, t)$  and the (fluid) pressure  $p = p(x, t)$  such that

$$-\operatorname{div}(\sigma(u) - \alpha p \mathbf{I}) = 0,$$

$$s\dot{p} + \alpha \operatorname{div} \dot{u} - \operatorname{div} \kappa \operatorname{grad} p = g$$

where  $g = g(t)$  is a given net inflow.

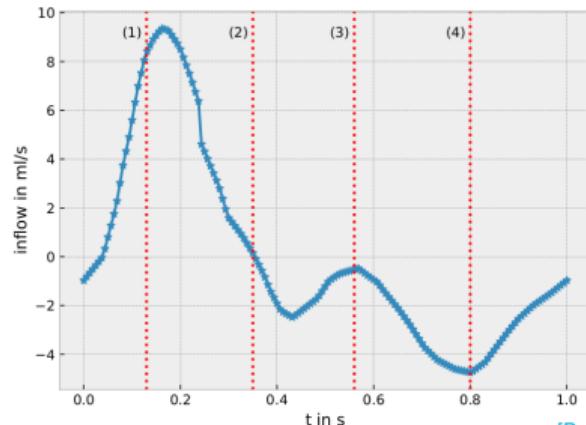
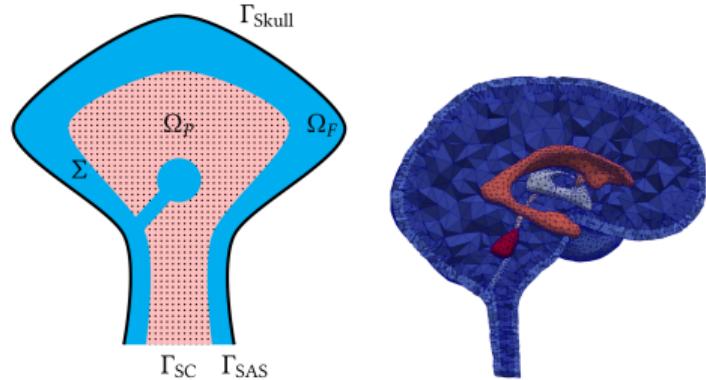
## Stokes' equations in $\Omega_F$

Find the velocity  $v = v(x, t)$  and the (fluid) pressure  $p = p(x, t)$  such that

$$\dot{v} - \operatorname{div}(\nu \varepsilon(v) - p \mathbf{I}) = 0,$$

$$\operatorname{div} v = 0$$

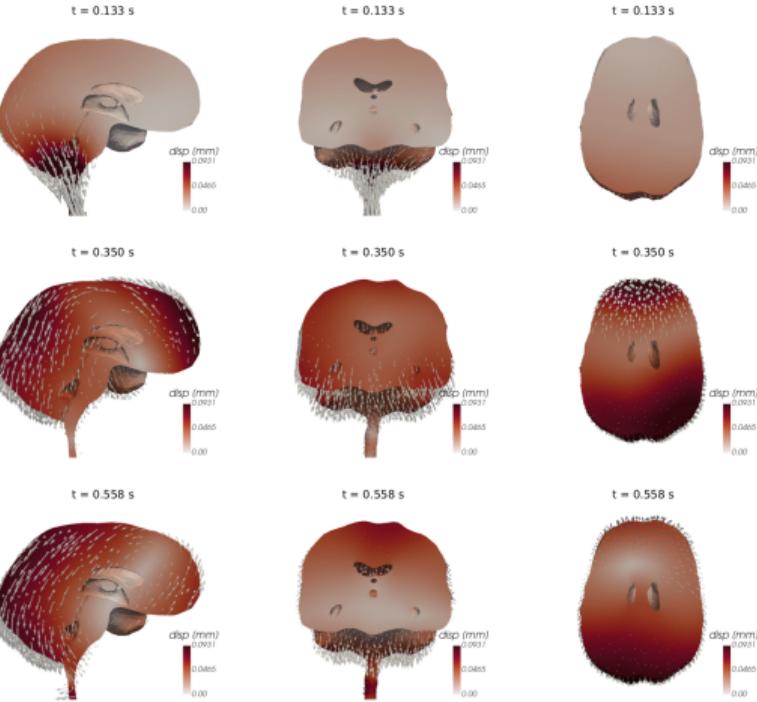
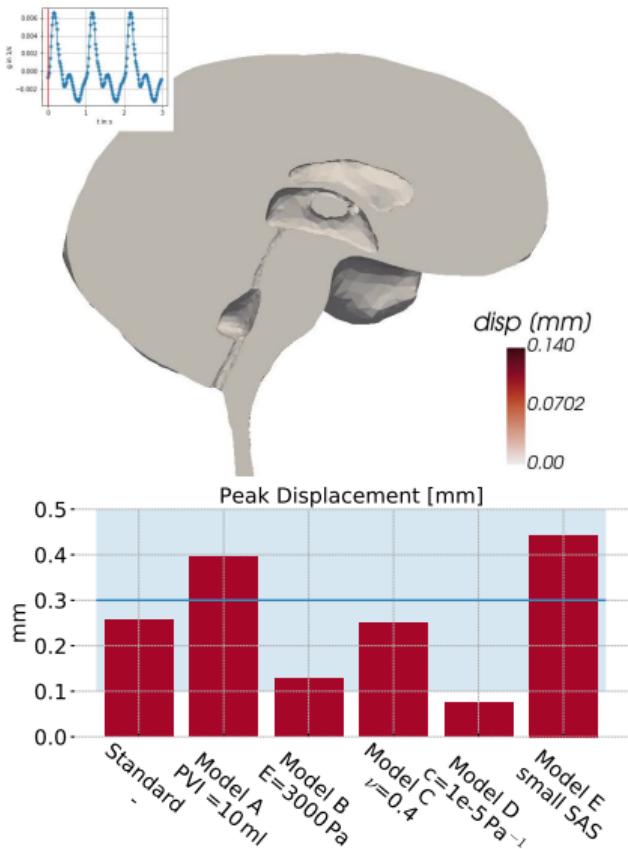
**Interface (resp. boundary) conditions at  $\Sigma$  (resp.  $\Gamma$ ).**



[Balédent (2014)]

# Pulsatile blood inflow yields heterogeneous displacement patterns

[Causemann et al. Human intracranial pulsatility... FBCNS, 2022]

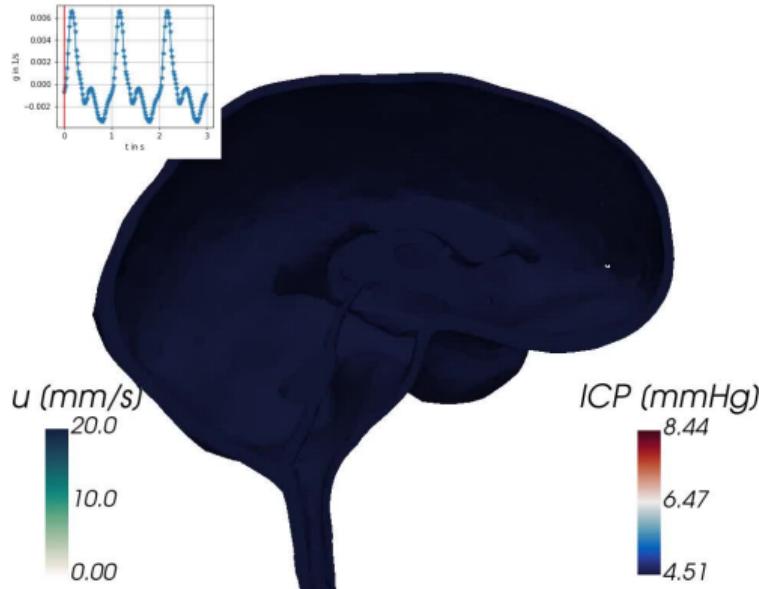


MRI studies estimate peak brain tissue displacement  $\sim 0.1\text{--}0.5 \text{ mm}$ .

[Enzmann & Pelc (1992), Greitz et al (1992), Poncelet et al (1992), Pahlavian et al (2018)]

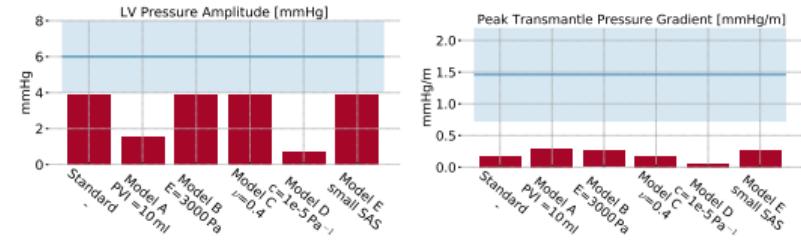
# .. nearly homogeneous pressures but complex fluid flow

[Causemann et al, Human intracranial pulsatility..., FBCNS, 2022]

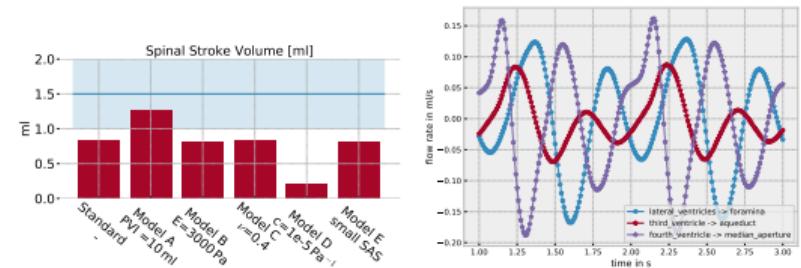


Simulations offer unprecedented detail.

Models predict pressure variations of 1–4 mmHg, transmantle gradients of 0.1–0.2 mmHg/m.



Cerebral–spinal stroke volume up to 1.3 mL.  
Multiple flow reversals in ventricles?



## **Software and workflows**

# These slides and other lecture material are available via GitHub

The screenshot shows a GitHub repository page for 'meg-simula / mri2fem-lectures'. The repository is public. The main navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. Below the navigation bar, there are buttons for Pin, Unwatch, and a dropdown menu. The main content area shows the repository's structure with a 'Code' tab selected. It displays the 'main' branch, 1 branch, and 0 tags. There are buttons for Go to file, Add file, and Code. A detailed list of files is shown, including '2022-11-08-guest-lecture-ucsd', 'slides', 'README.md', 'abstract.txt', 'lectures1-2.pdf', and 'lectures3-4.pdf'. On the right side, there is a 'Clone' section with options for HTTPS, SSH, and GitHub CLI, and a 'Download ZIP' button.

meg-simula / **mri2fem-lectures** Public

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

meg-simula Start editing software and tools part

- 2022-11-08-guest-lecture-ucsd Start editing software and tool
- slides Add PDF for lectures 5-6.
- README.md Update README.md
- abstract.txt Add beginning of Bergen shor
- lectures1-2.pdf Add PDFs of Lectures 3-4 (for Jan 22 2021)
- lectures3-4.pdf Update pdf

Clone

HTTPS SSH GitHub CLI

git@github.com:meg-simula/mri2fem-lecture

Use a password-protected SSH key.

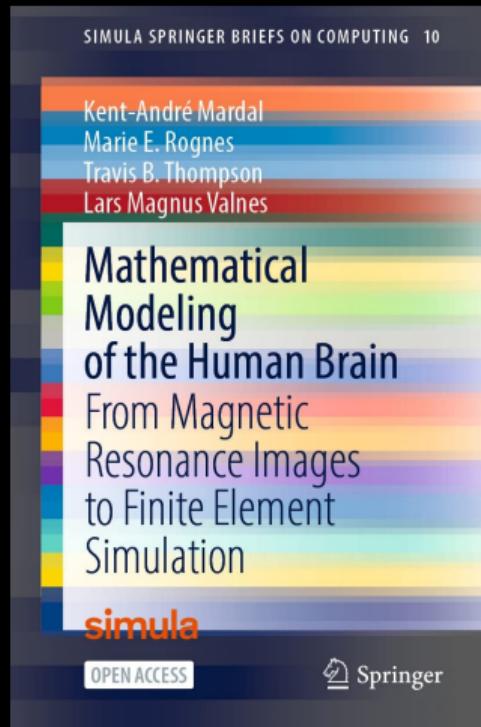
Download ZIP

[https://github.com/meg-simula/mri2fem-lectures]

2 years ago

2 years ago

# Mathematical modeling of the human brain: from MRI to FEM



A screenshot of a GitHub repository page for "kent-and / mri2fem". The repository is public and contains 1 branch and 0 tags. The most recent commit was made by meg-simula, updating the README.md file 3 minutes ago. The commit message is "Update README.md". The repository also includes a "book" folder, which contains an updated compiled PDF version of the book. The "mri2fem" folder was last modified 17 months ago. The "README.md" file was last updated 3 minutes ago. A script named "make\_archive.sh" was added 17 months ago to create a Zenodo archive. The README.md file itself describes the manuscript repository for mathematical modeling of the human brain, mentioning authors K. A. Mardal, M. E. Rognes, T. B. Thompson, and L. M. Valnes, and the year 2022. A link to the Springer book is provided: <https://link.springer.com/book/10.1007/978-3-030-95136-8>.

[<https://github.com/kent-and/mri2fem>]

# Further resources (data sets, software) are available via Zenodo

zenodo

Search  Search icon

Upload Communities

[Log in](#) [Sign up](#)

## Mathematical modeling of the human brain - from magnetic resonance images to finite element simulation

### Recent uploads

Search Mathematical modeling of the human brain - from magnetic resonance images to finite element simulation Search icon

[June 4, 2021 \(v2.0\)](#) [Software](#) [Open Access](#)

**Software for Mathematical modeling of the human brain - from magnetic resonance images to finite element simulation**

Kent-Andre Mardal, Marie E. Rognes; Travis B. Thompson; Lars Magnus Valnes;

Software collection for Mathematical modeling of the human brain From magnetic resonance images to finite element simulation by Kent-Andre Mardal, Marie E. Rognes, Travis B. Thompson, and Lars Magnus Valnes Python modules, Bash scripts, and input files organized by chapter but with otherw

Uploaded on June 4, 2021

1 more version(s) exist for this record

[View](#)

[June 4, 2021 \(v2.0\)](#) [Dataset](#) [Open Access](#)

**MRI2FEM data set**

Kent-Andre Mardal, Marie E. Rognes; Travis B. Thompson; Lars Magnus Valnes;

DICOM data and FreeSurfer recon-all generated files for Mathematical modeling of the human brain: from magnetic resonance images to finite element simulation (MRI2FEM)

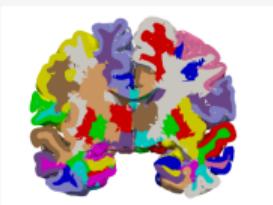
Uploaded on June 4, 2021

1 more version(s) exist for this record

[View](#)

Upload icon [New upload](#)

Community



Mathematical modeling of the human brain - from magnetic resonance images to finite element simulation

Data and software associated with "Mathematical modeling of the human brain - from magnetic resonance images to finite element simulation" by Mardal, Rognes, Thompson and Valnes (2021).

[<https://zenodo.org/communities/mri2fem/>]

# Step-by-step tutorial videos are available on YouTube

The image shows a screenshot of a YouTube search results page. The search bar at the top contains the query "mri2fem". Below the search bar, there is a list of video thumbnails and titles. On the left side of the screen, there is a sidebar with navigation links: Home, Shorts, Subscriptions, and Library. The main content area displays a video thumbnail for "From brain magnetic resonance imaging to finite..." by Marie Elisabeth Rognes, followed by a list of seven tutorial videos under the heading "Sort".

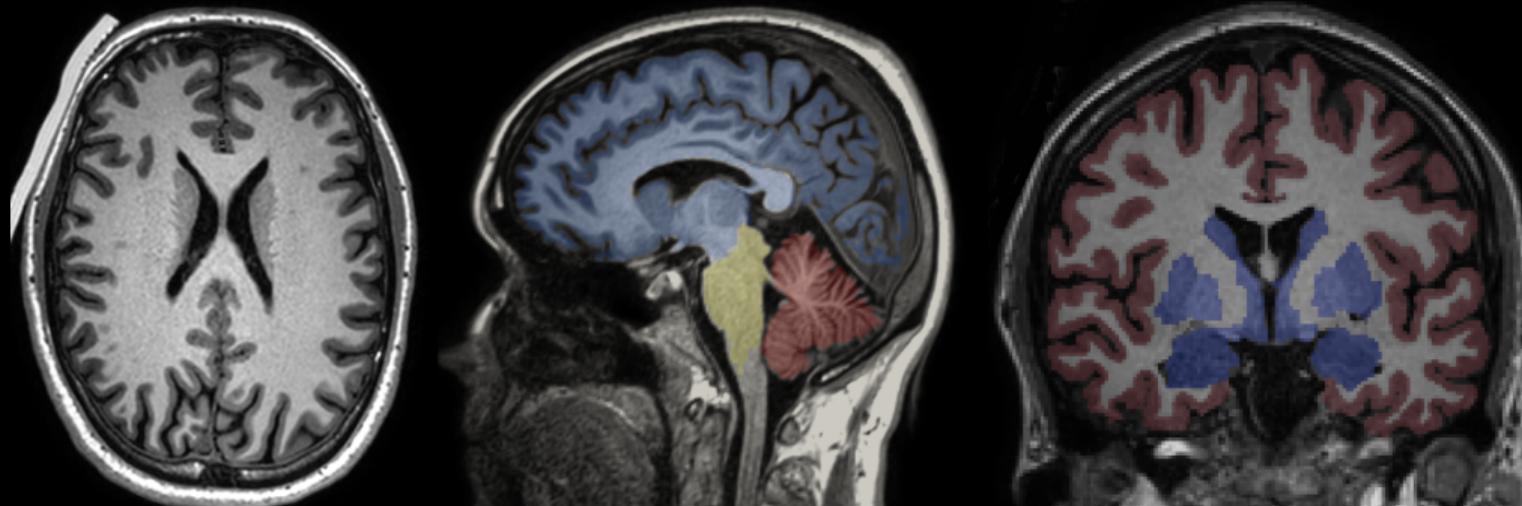
**From brain magnetic resonance imaging to finite...**  
Marie Elisabeth Rognes  
Public  
8 videos Last updated on Jan 28, 2021

**Sort**

- mri2fem: Downloading and viewing DICOMs**  
Marie Elisabeth Rognes
- mri2fem: Viewing surfaces generated by FreeSurfer using Freeview and ParaView**  
Marie Elisabeth Rognes
- mri2fem: Creating a brain mesh from surfaces using SVM-Tk**  
Marie Elisabeth Rognes
- mri2fem: Simulating diffusion in a brain hemisphere using FEniCS**  
Marie Elisabeth Rognes
- mri2fem: Creating a mesh conforming to gray and white matter using SVM-Tk and FreeSurfer surfaces**  
Marie Elisabeth Rognes
- mri2fem: extracting the ventricles and removing them from the brain mesh**  
Marie Elisabeth Rognes

[[https://www.youtube.com/playlist?list=PL2H4xO8DvXwVNy7EeIY\\_TuYUgvQ4Dw\\_Lz](https://www.youtube.com/playlist?list=PL2H4xO8DvXwVNy7EeIY_TuYUgvQ4Dw_Lz)]

# T1-weighted magnetic resonance imaging (MRI) reveals structure



Axial, sagittal and coronal cross-sections. T1-weighted (T1w) MRI: fat gives high signal intensity (light/white); fluids give low signal intensity (dark/black).

# Viewing and working with MRI data sets in DICOM format

Medical images are often stored in the DICOM file format: a collection of image files arranged in sequences. A DICOM viewer is useful for working with DICOM files.

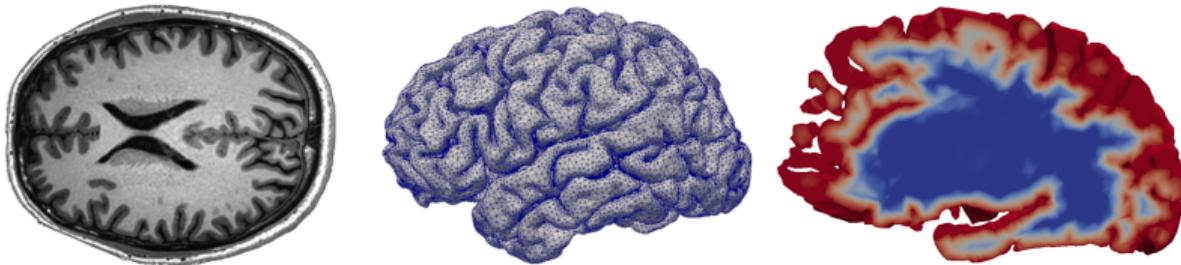
Many possibilities including the basic DicomBrowser

```
$ sudo apt-get install dicombrowser                               (Ubuntu 18.04)
$ DicomBrowser &
```

Video: Downloading and viewing the mri2fem DICOM data set

# Recommended software components

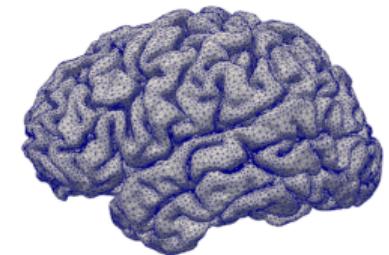
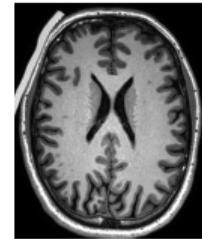
- ◊ Python3 for everything
- ◊ FreeSurfer for segmentation: <https://surfer.nmr.mgh.harvard.edu/>
- ◊ NiBabel for image manipulations: <https://nipy.org/nibabel/>
- ◊ SVM-Tk for meshing: <https://github.com/SVMTK/SVMTK>
- ◊ meshio for mesh conversions: <https://github.com/nschloe/meshio>
- ◊ FEniCS for finite elements : <https://fenicsproject.org/>
- ◊ ParaView for visualization: <https://www.paraview.org/>



# We follow these three steps to generate a mesh from T1 images

To generate a mesh from an MRI data set including T1-weighted images, we follow three main steps:

1. extract a T1-weighted image series from the MRI dataset using DicomBrowser;
2. create (boundary) surfaces from the T1-weighted images using FreeSurfer;
3. generate a volume mesh of the interior of these using SVM-Tk.



## Step 2: Using FreeSurfer to segment images and create surfaces

FreeSurfer offers the command `recon-all` to segment the brain images and reconstruct surfaces (parcellations, pial surface, white matter surface etc.)

```
$ cd mri2fem-dataset  
$ cd dicom/ernie/T13D  
$ recon-all -subjID ernie -i IM_0162 -all
```

This command is compute-intensive, with run times likely of 12–24 hours.

Key outputs (`mri2fem-dataset/freesurfer/ernie`) are

- ◊ `/stats`: contains files providing statistics derived during segmentation.
- ◊ `/mri`: contains volume files generated during segmentation
- ◊ `/surf`: contains surface files generated during segmentation

Video: Viewing the FreeSurfer output

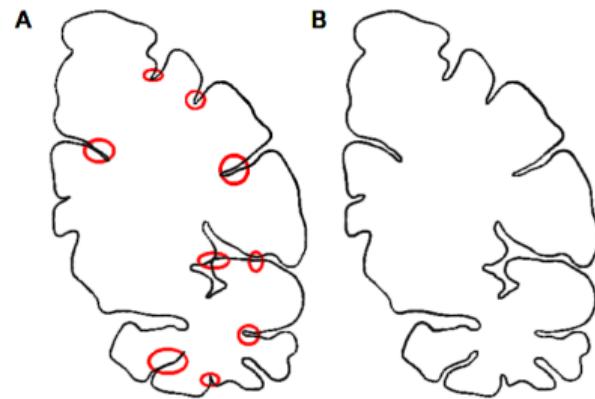
# The surfaces may need enhancement to be suitable for finite element meshes

In practice, brain surfaces generated from T1 images

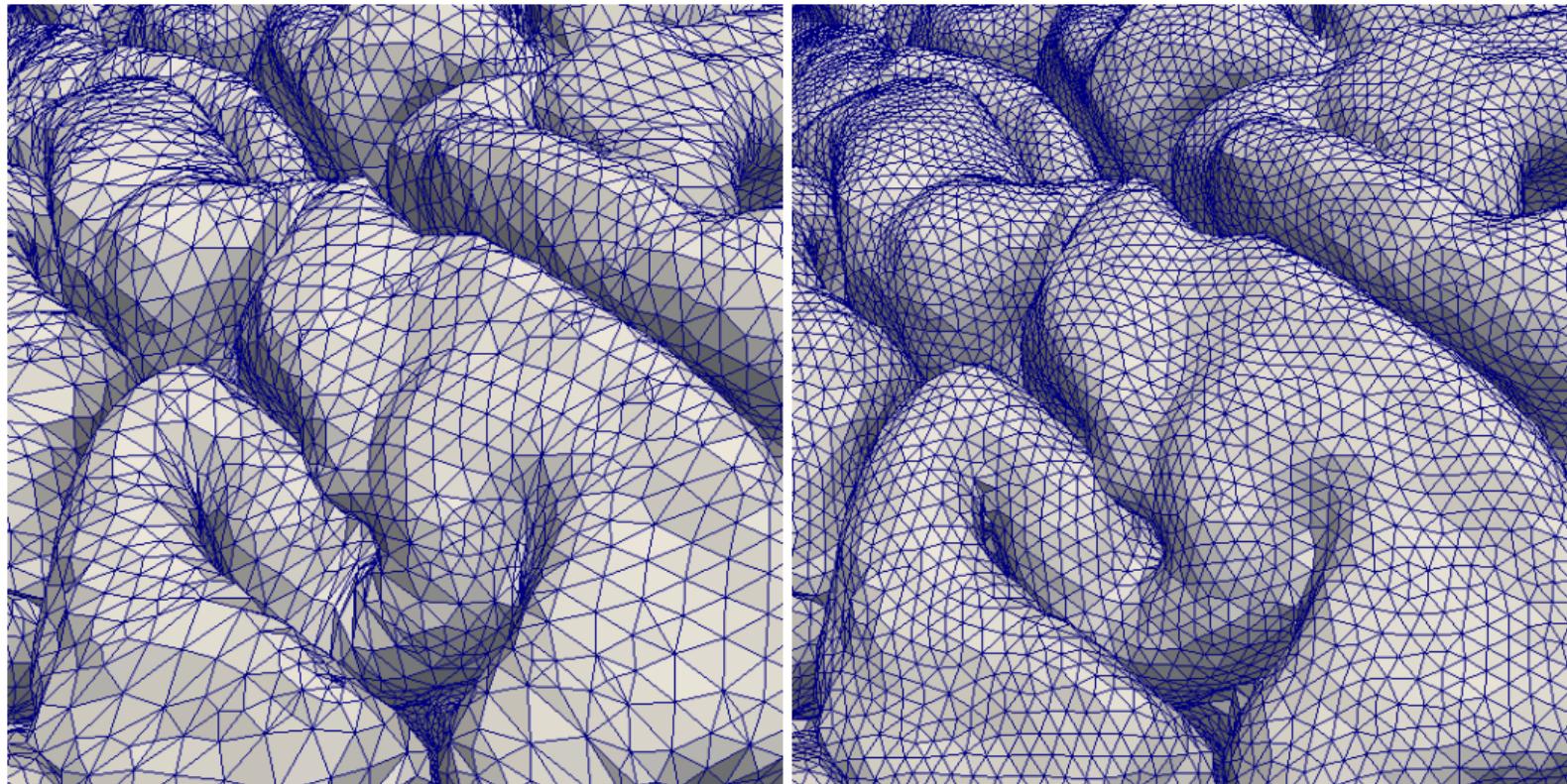
- ◊ may have unphysiologically sharp corners,
- ◊ may include triangles with very large aspect ratios,
- ◊ have topological defects such as holes, and
- ◊ may self-intersect or overlap with other surfaces.

**Result:** Low quality meshes (if any).

**Fix:** Enhance surface quality prior to meshing

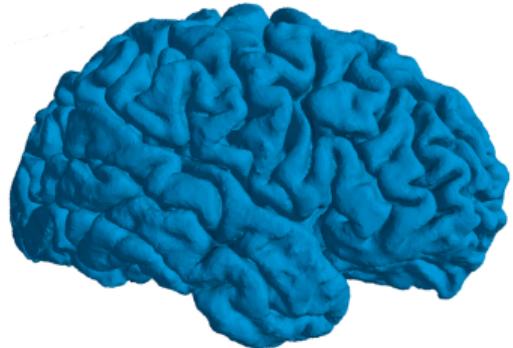


# SVM-Tk (wrapping CGAL) includes utilities for remeshing surfaces

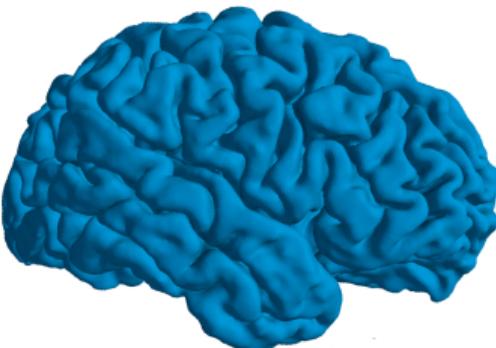


[Mardal et al (2021, Chapter 3.2.1); mri2fem/chp3/remesh\_surface.py]

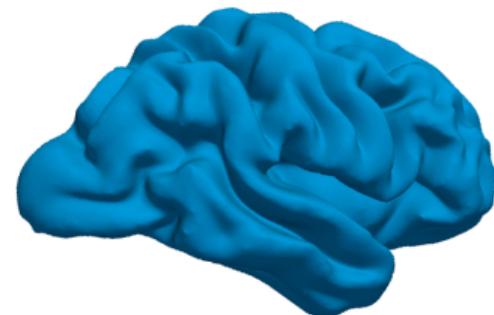
# SVM-Tk includes utilities for smoothing surfaces



Original pial surface



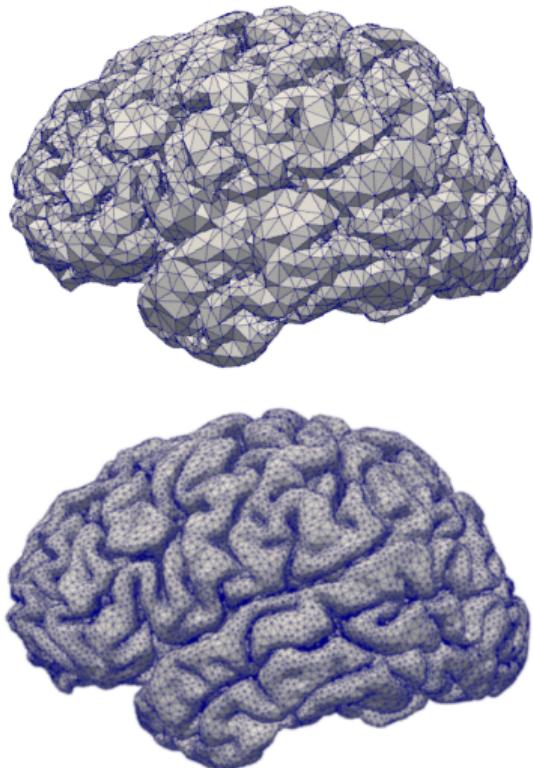
after Taubin smoothing



after over-smoothing.

[Mardal et al (2021, Chapter 3.2.2); mri2fem/chp3/smooth\_surface.py]

# SVM-Tk is designed to create brain volume meshes from surfaces



```
import SVMTK as svmtk

def create_volume_mesh(stlfile, output, n=16):
    # Load input file
    surface = svmtk.Surface(stlfile)

    # Generate the volume mesh
    domain = svmtk.Domain(surface)
    domain.create_mesh(n)

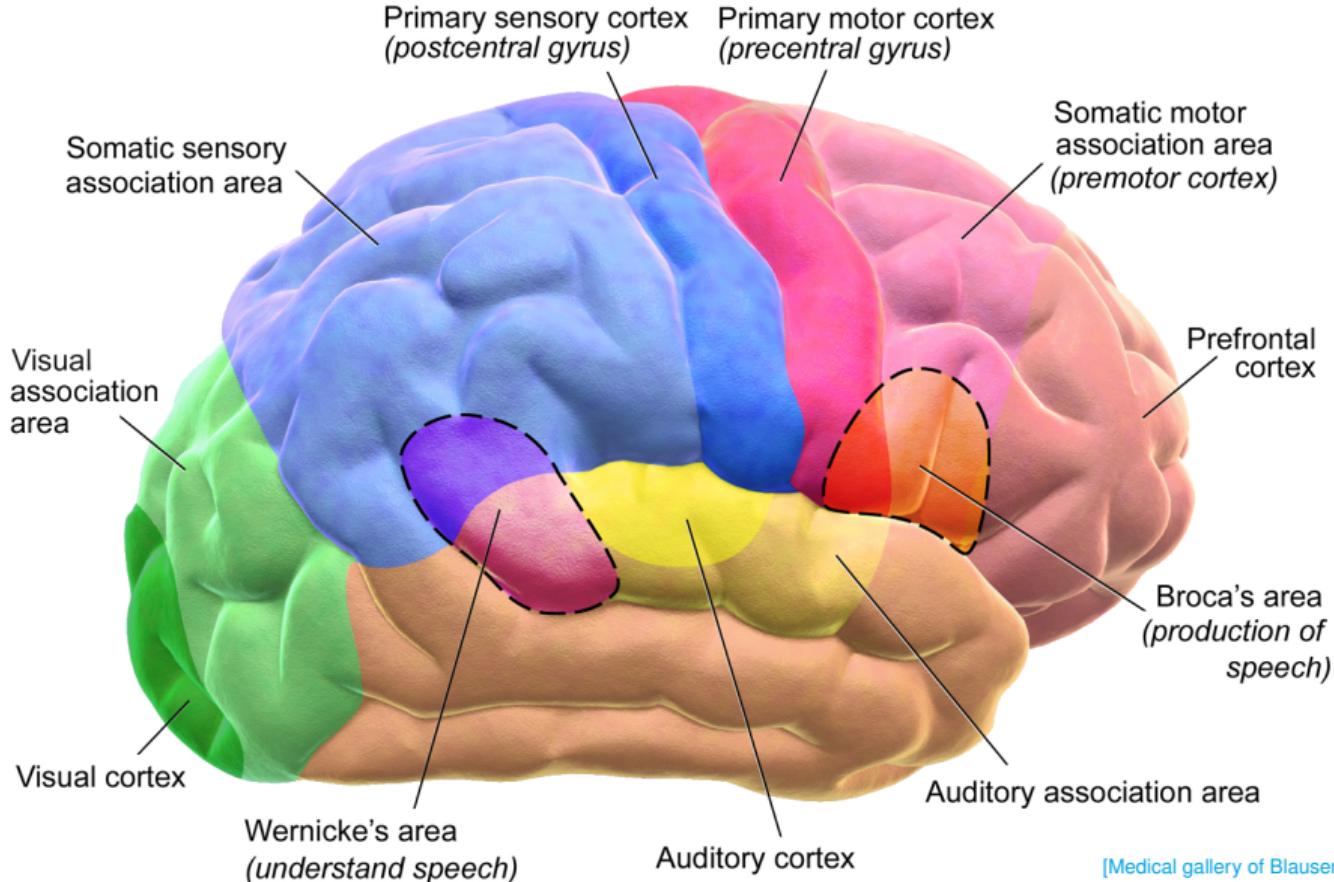
    # Write the mesh to the output file
    domain.save(output)

# Create mesh
create_volume_mesh("lh.pial.stl", "lh.mesh")
```

[Mardal et al (2021, Chapter 3.1.3); mri2fem/chp3/surface\_to\_mesh.py]

Video: Creating a mesh from brain surfaces

# Brain parcellations



[Medical gallery of Blausen Medical 2014]

# NiBabel is a useful Python module for operating on image data



Parcellation ("wmparc.mgz") in voxel space as generated by FreeSurfer.

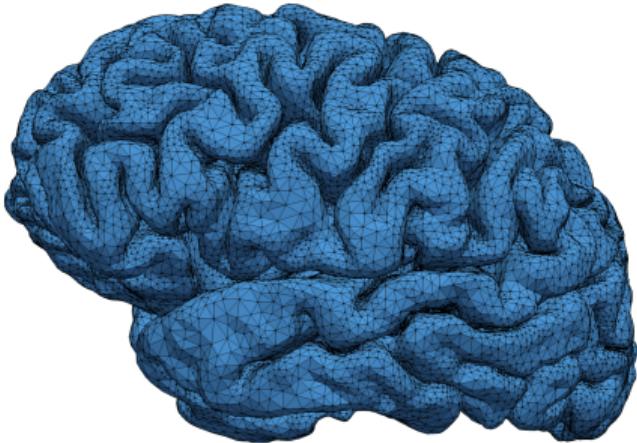
```
import numpy
import nibabel
from nibabel.affines import apply_affine

# Load image, extract its data
# and output its dimensions
image = nibabel.load("wmparc.mgz")
data = image.get_fdata()

# Examine the image dimensions,
# and print some value
print(data.shape)
print(data[100, 100, 100])
```

[<https://nipy.org/nibabel/>]

# Representing discrete mesh data in FEniCS



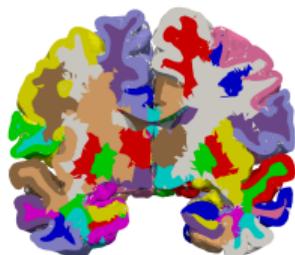
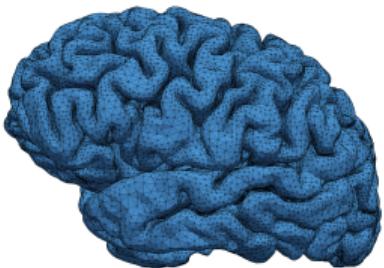
Brain mesh visualized in ParaView.

```
# Import brain mesh
mesh = Mesh()
with XDMFFFile("ernie-brain-32.xdmf") as file
    :
        file.read(mesh)
print(mesh.num_cells())

# Define cell-based region representation
n = mesh.topology().dim()
regions = MeshFunction("size_t", mesh, n, 0)
print(regions[0])
print(regions.array())
```

Video Mapping image-based data into a finite element (FEniCS) representation

# Converting brain parcellations between voxel space and RAS



```
# Find the transformation f from T1 voxel space
# to RAS space and take its inverse to get the
# map from RAS to voxel space
vox2ras = image.header.get_vox2ras_tkr()
ras2vox = numpy.linalg.inv(vox2ras)

print("Iterating over all cells...")
for cell in cells(mesh):
    c = cell.index()

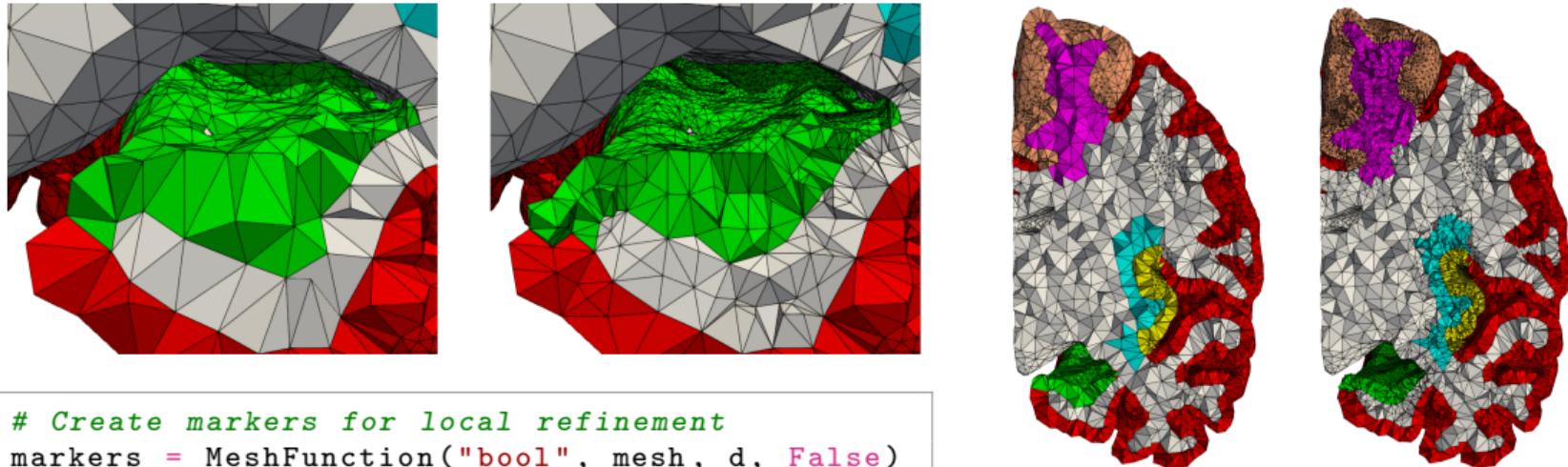
    # Extract RAS coordinates of cell midpoint
    xyz = cell.midpoint()[:]

    # Convert to voxel space
    ijk = apply_affine(ras2vox, xyz)

    # Round off to nearest integers to find voxel indices
    i, j, k = numpy.rint(ijk).astype("int")

    # Insert image data into the mesh function:
    regions.array()[c] = int(data[i, j, k])
```

# Local mesh adaptivity in regions of interest



```
# Create markers for local refinement
markers = MeshFunction("bool", mesh, d, False)

# Iterate over given tags, label all cells
# with this subdomain tag for refinement:
for tag in tags:
    markers.array()[domains.array() == tag] = True

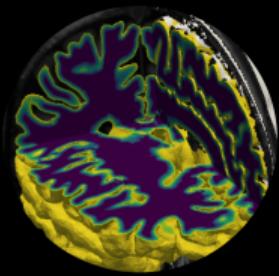
# Refine mesh according to the markers
new_mesh = adapt(mesh, markers)
```

Illustration of left hemisphere meshes with different parcellation regions at different resolutions. The meshes on the right are local refinements of the meshes on the left. Zoomed in view of the hippocampus region (green).

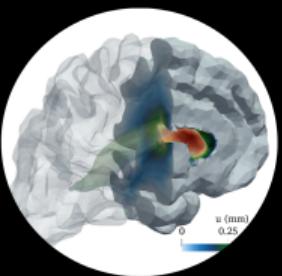
# Igor, bring me the brain! Want to give it a try? Natural first steps:

1. Download:
  - ◊ these slides: <https://github.com/meg-simula/mri2fem-lectures/2022-11-08-lecture.pdf>.
  - ◊ the mri2fem data set and software (~ 1Gb) <https://zenodo.org/communities/mri2fem/>
  - ◊ the *Mathematical modelling of the human brain: from MRI to FEM* book:  
<https://github.com/kent-and/mri2fem>.
2. Install software dependencies following the instructions in Chapter 2 of Mardal et al (2021).
3. Inspect the data set contents using the tools described (e.g. DicomBrowser, FreeSurfer)
4. Try running some of the sample code in Chapter 3 of the book.
5. Simulate something other than diffusion? Give it a try!

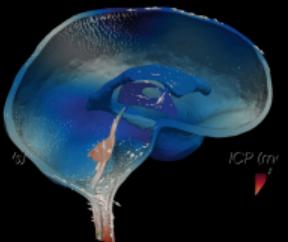
## Concluding remarks



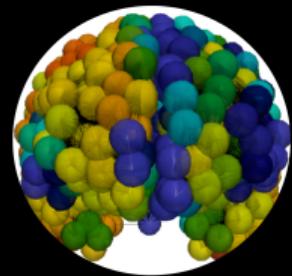
Solute transport



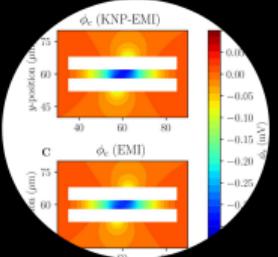
Brain mechanics



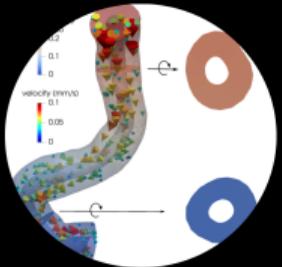
CSF flow



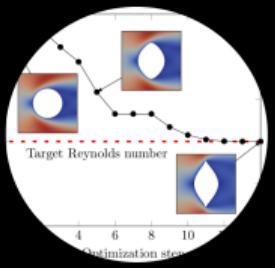
Neurodegeneration



Ions and osmosis



Model reduction



Optimal control



Software

