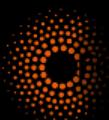




waterscales



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



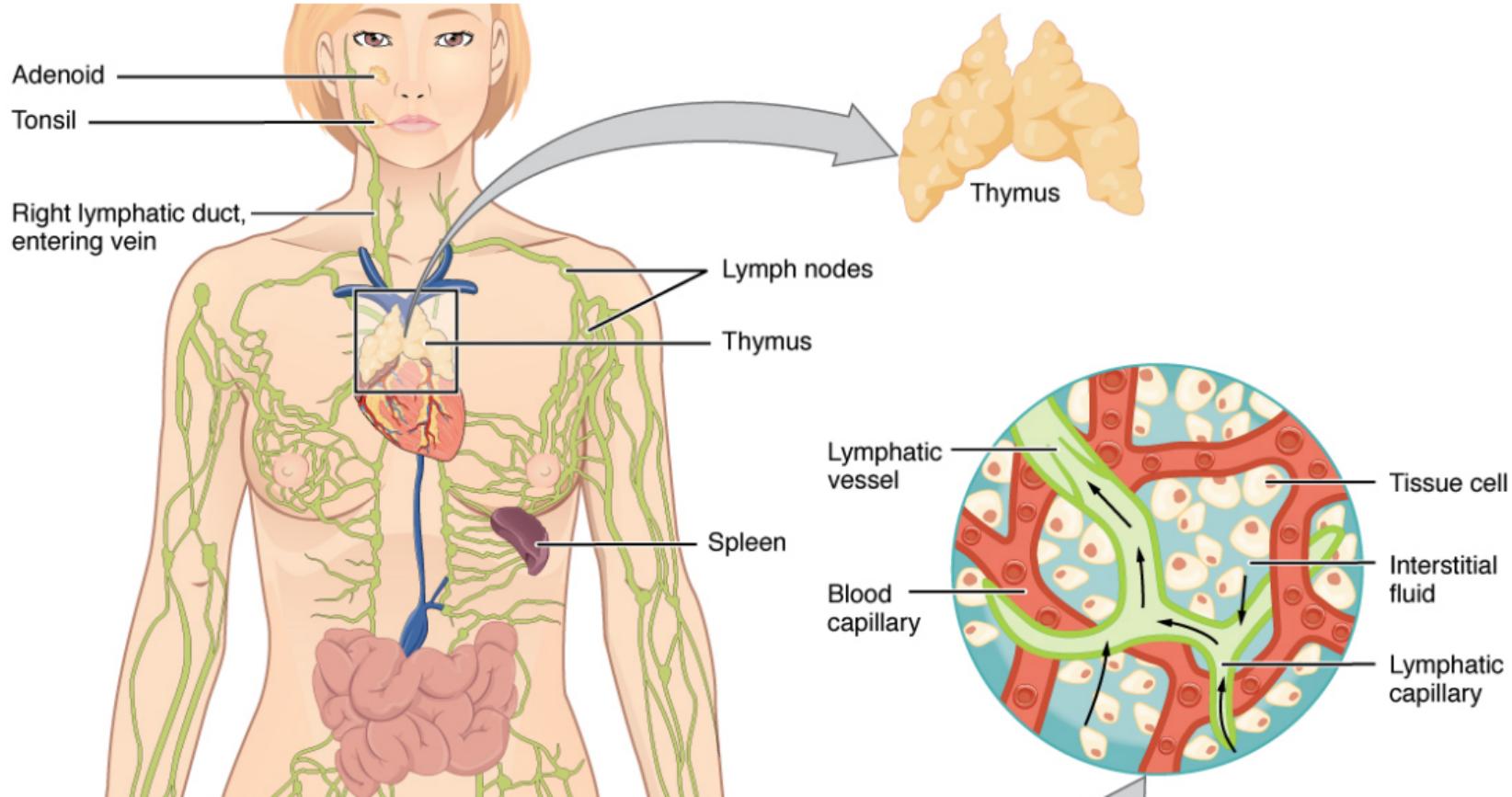
The brain may clean out Alzheimer's plaques during sleep

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

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

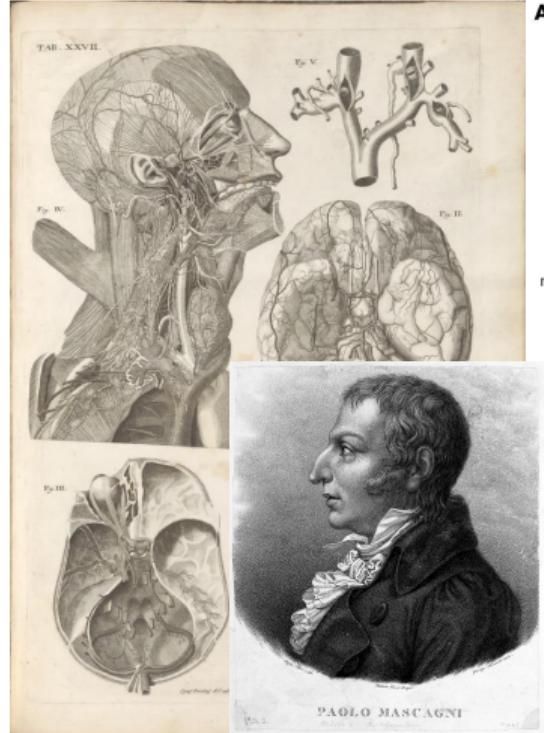


The lymphatic system drains tissue fluid and collects metabolic waste, bacteria and cellular debris

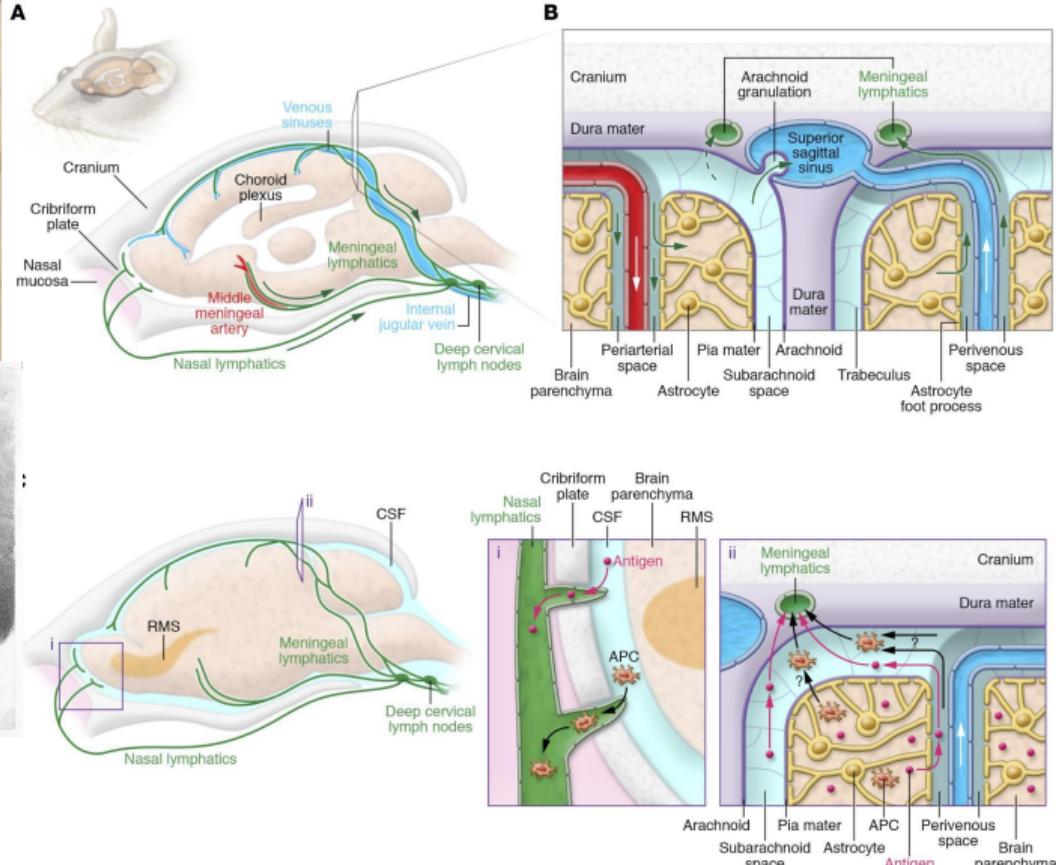


Computational brainphatics for understanding the brain's wandscape

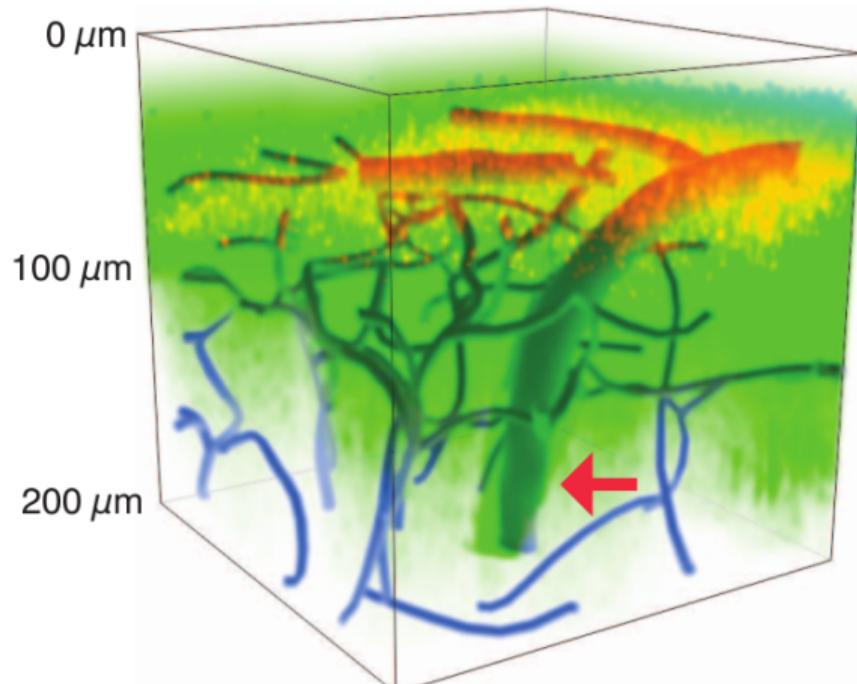
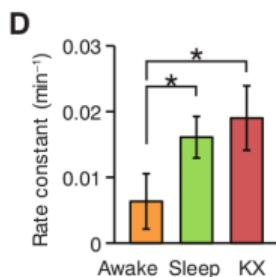
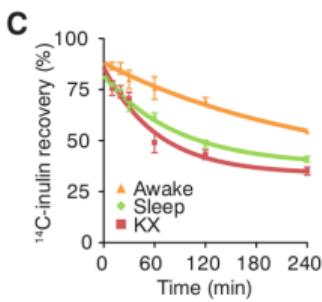
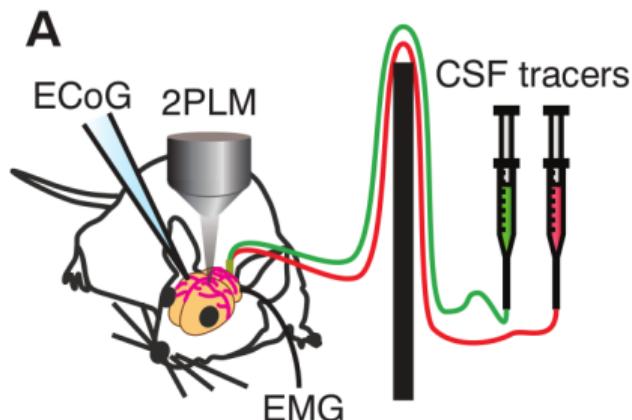
[Louveau et al, 2017 (Fig 2)]



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



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



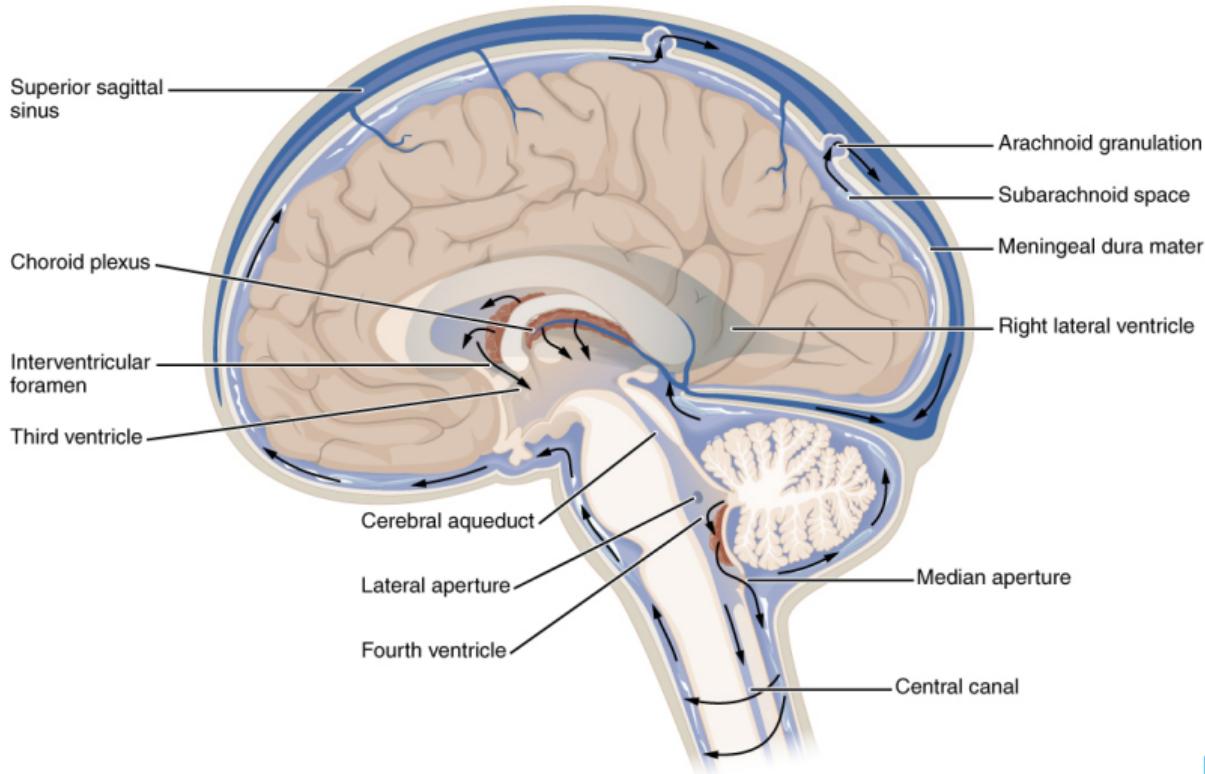
[Cserr et al (1981), Rennels et al (1985), Ichimura et al (1991), Abbott (2004), Hadaczek et al (2005), Iliff et al (2012), Xie et al (2012, Fig 1A-B, Fig 2C-D), Bojarskaite et al (2020)]

Interdisciplinary overview and goals of lecture

1. Motivation (5 min)
2. Brain physiology and physics (15 min)
3. The brain as a Stokes-Biot coupled problem (30 min)
4. Break (5 min)
5. Simulating brain pulsatility (15 min)
6. Software and getting started (15 min)
 1. Understand core aspects of brain physiology and mechanics
 2. Understand how the brain can be modelled as a poroelastic medium (Biot) in a viscous fluid environment (Stokes).
 3. Learn about finite element methods for solving Biot's equations and coupled Stokes-Biot multiphysics problems.
 4. Understand how brain modelling and simulations can inform physiology and medicine
 5. 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

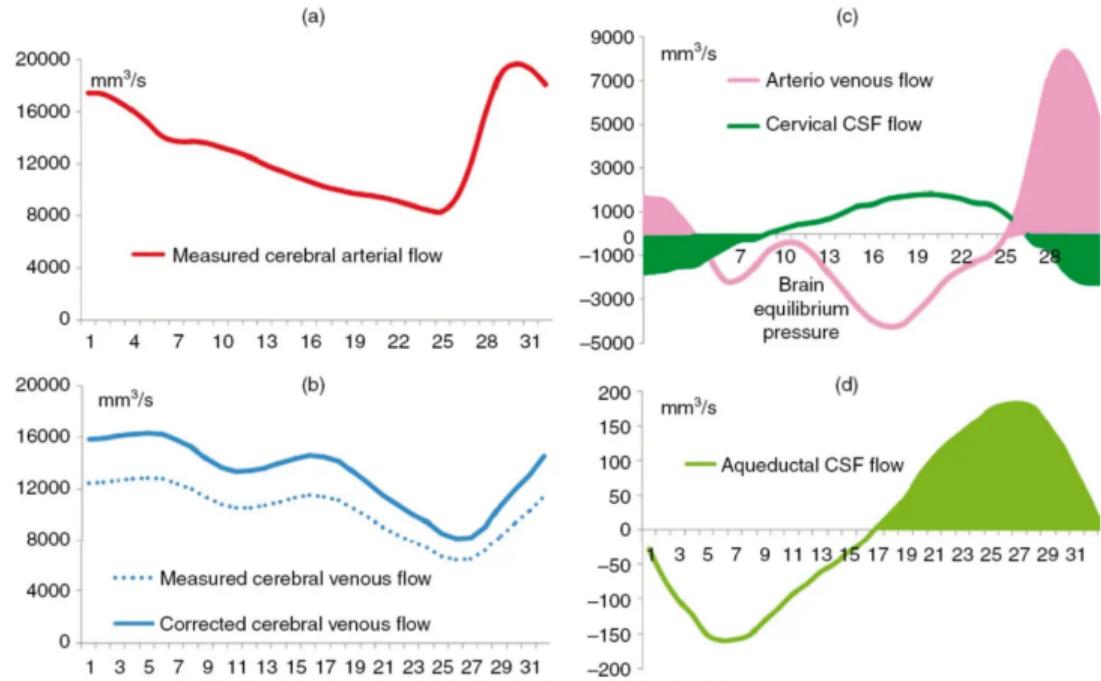
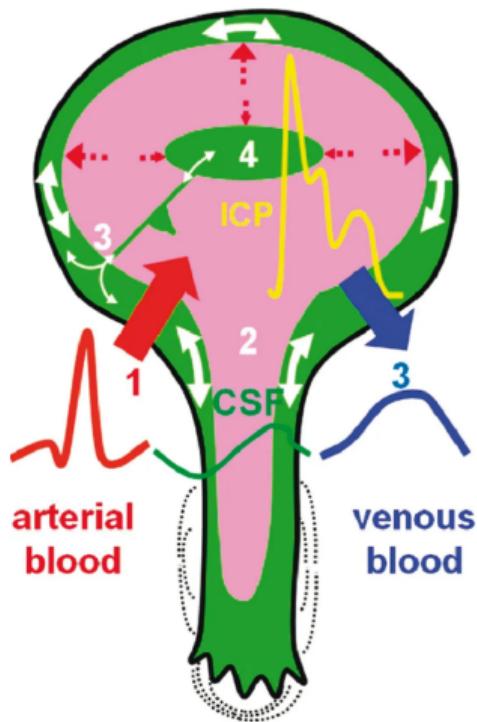
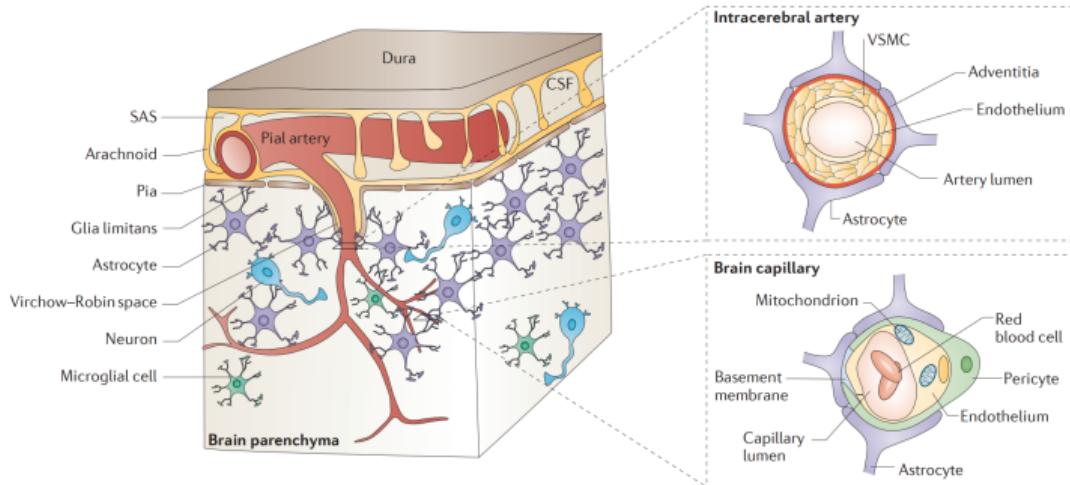


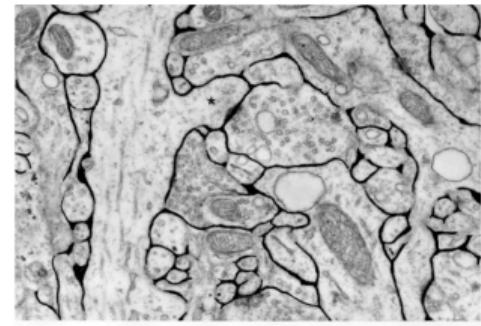
Figure 12.4 CSF and cerebral blood flow during the cardiac cycle in healthy adults (see text for full explanation).

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

[Budday et al (2019)]

Brain tissue is soft, heterogeneous and rheologically complex

Brain tissue is

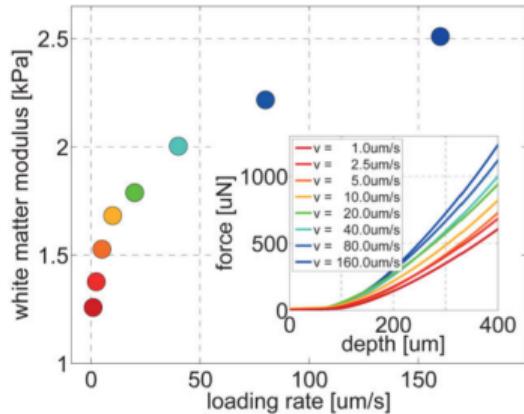
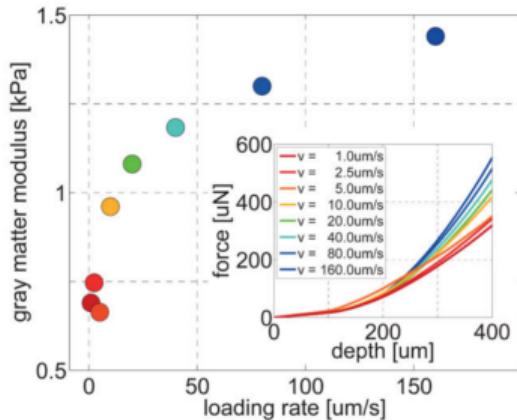
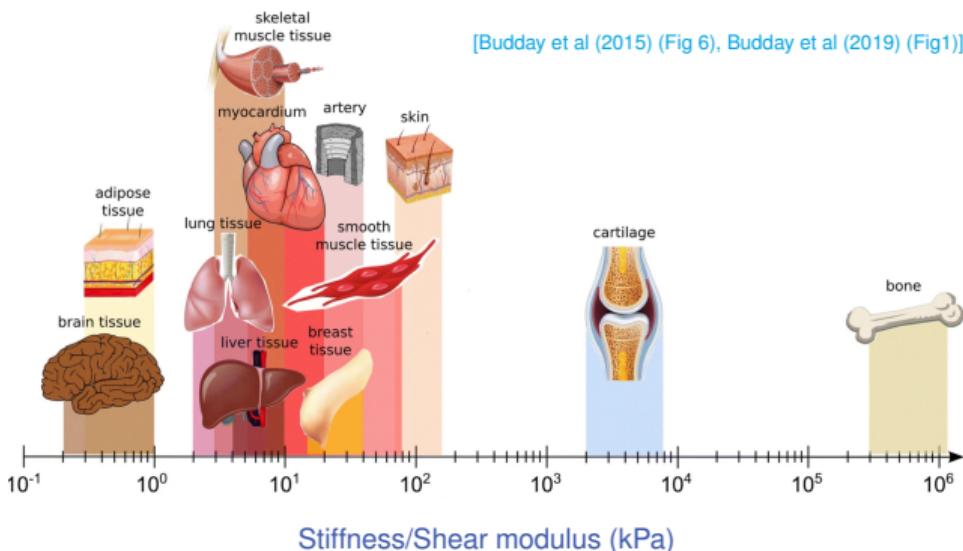
soft (shear modulus $\approx 0.5\text{--}2.5\text{ kPa}$)

stiffer with increasing strain/strain rates (nonlinear)

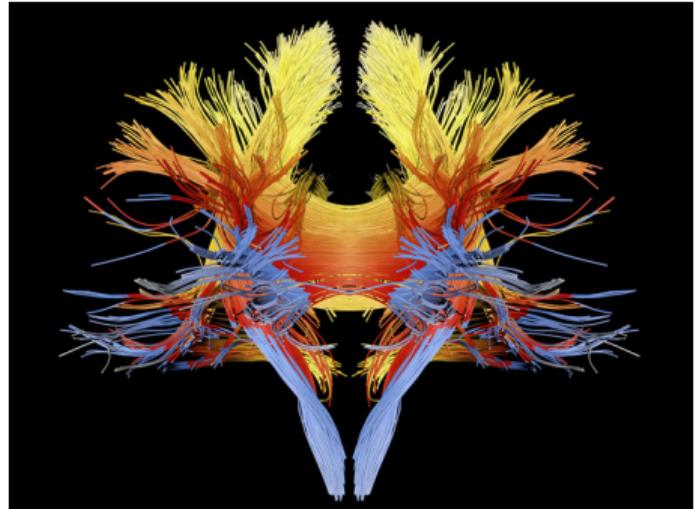
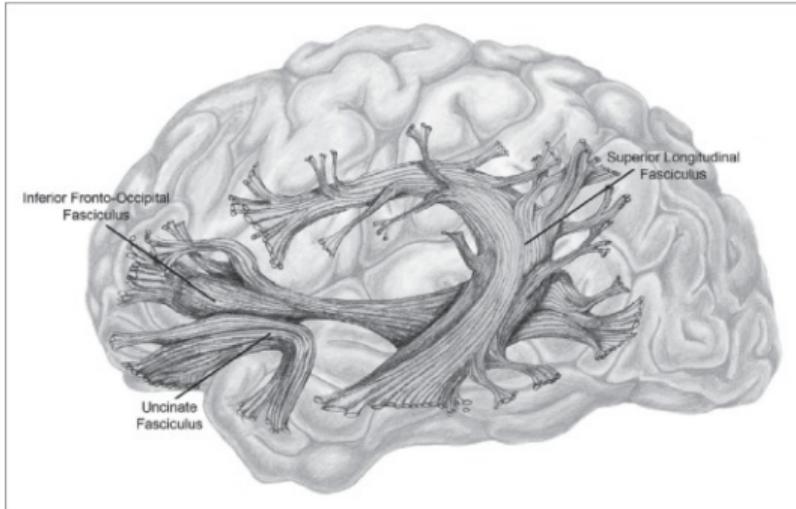
stiffer during loading than unloading (viscoelastic)

stiffer in compression than in tension (poroelastic)

stiffer in some regions than in others (heterogeneous)

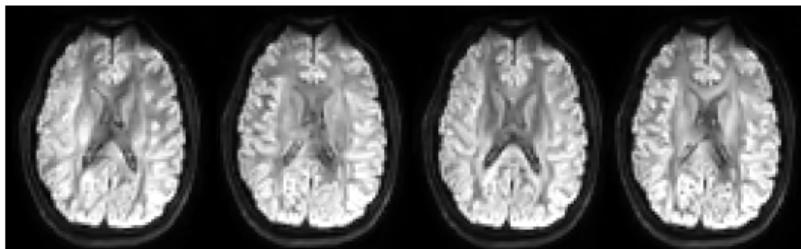


White matter fiber tracts induce anisotropy in the brain

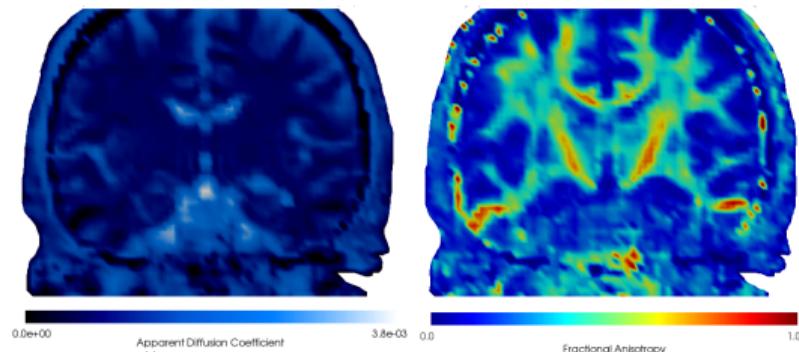


[Left: Wikimedia Commons. Right: Image credit: Alfred Pasieka]

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)

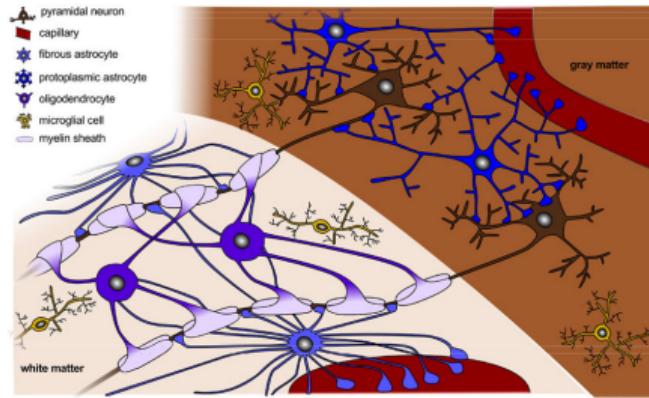
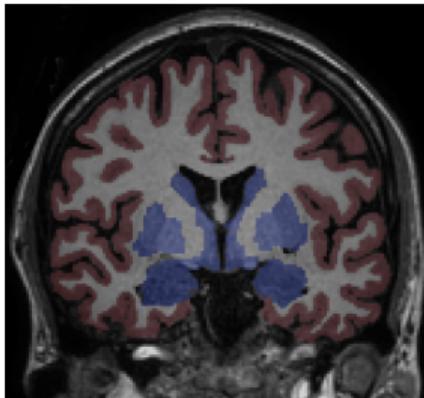
$$D = \begin{pmatrix} d_{11} & d_{12} & d_{13} \\ d_{21} & d_{22} & d_{23} \\ d_{31} & d_{32} & d_{33} \end{pmatrix}, \quad d_{ij} = d_{ij}(x).$$

D sym., pos. def, with eigenpairs (λ_i, v_i) .

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

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

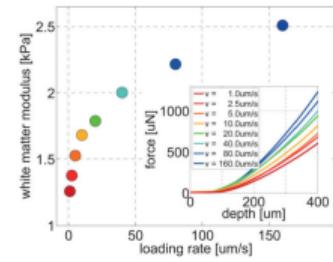
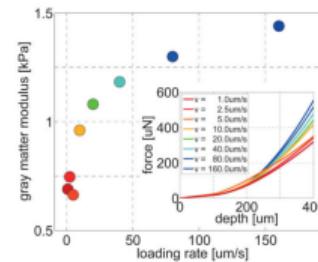
Gray and white matter differs substantially in terms of composition and biophysical properties



Example:

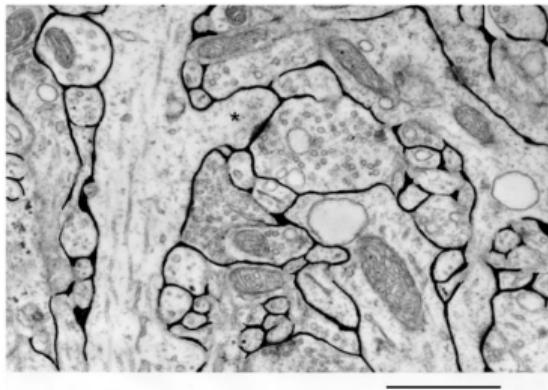
Heterogeneous stiffness (permeability, diffusion) tensor

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



[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)]

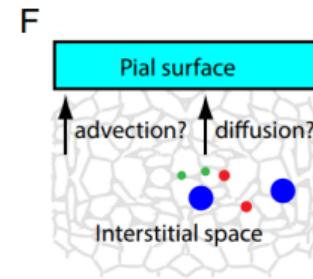
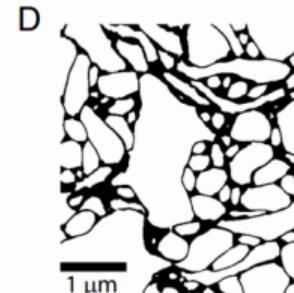
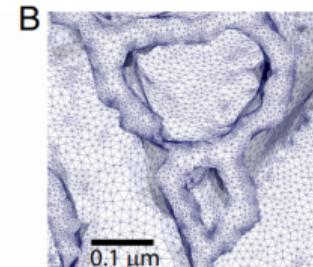
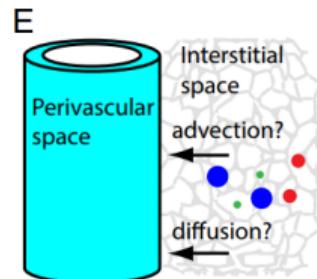
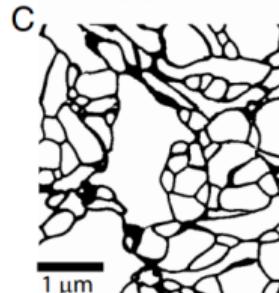
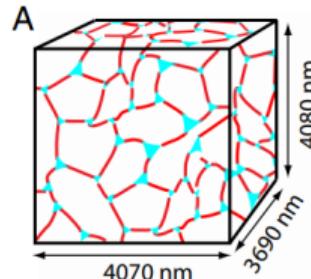
ECS diffusion and tortuosity $D^* = D\lambda^{-2}$:

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

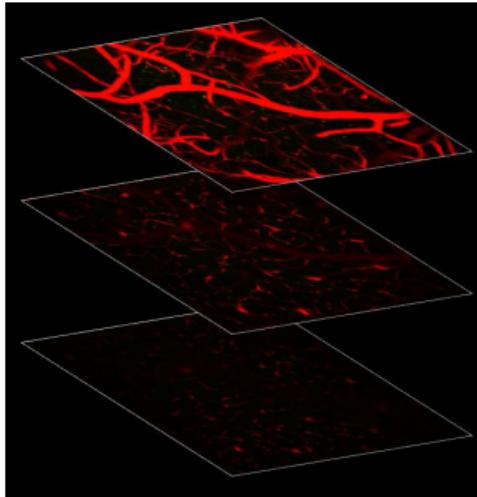
ECS permeability (κ , nm^2):

Holter et al (2017): 10-20

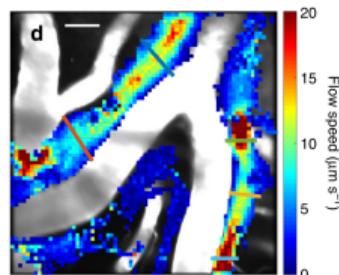
Basser (1992): 4000



Fluid movement in perivascular spaces enhances solute transport



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

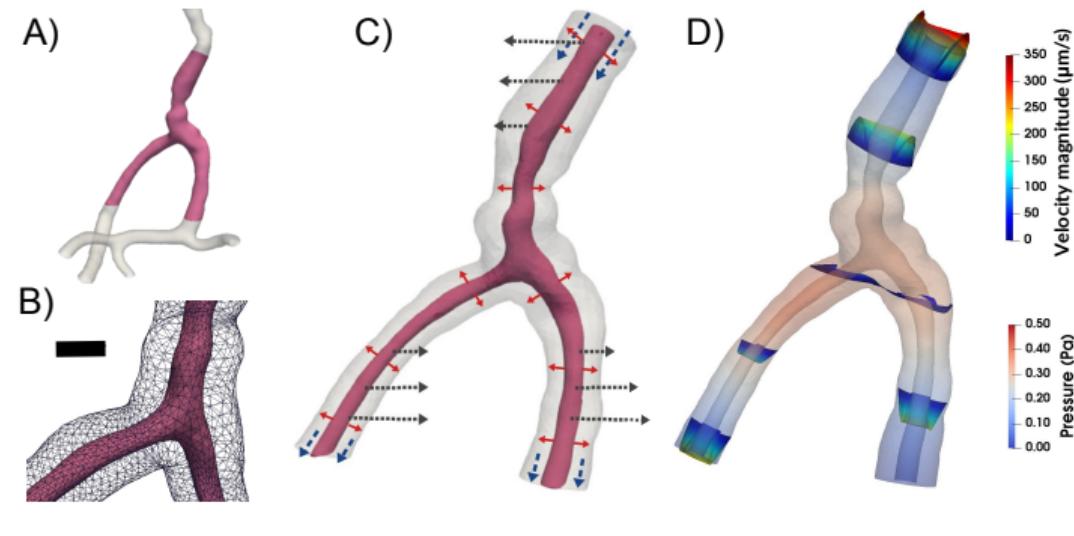


Key mechanism for brain solute transport:
pulsatile flow of CSF/ISF in perivascular spaces

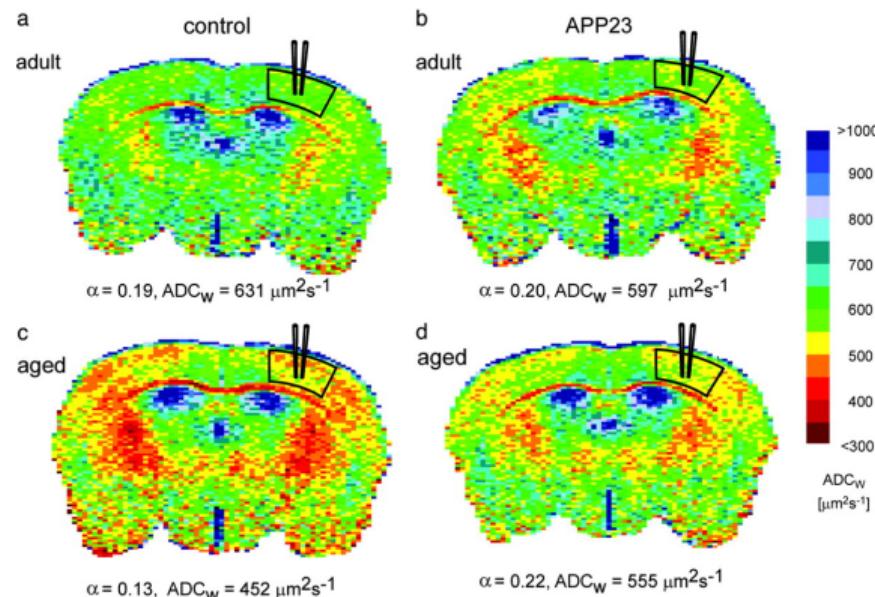
Open questions: Anatomical existence? Directionality?
Magnitude? Importance?

[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)]

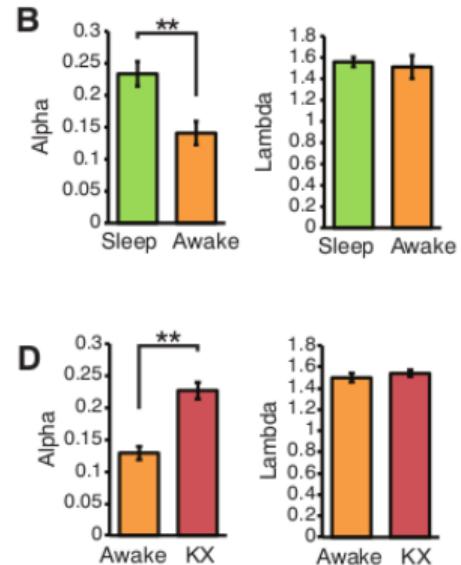


Brain tissue is active and dynamic, and its properties change with circadian rhythm, age and pathologies



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

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

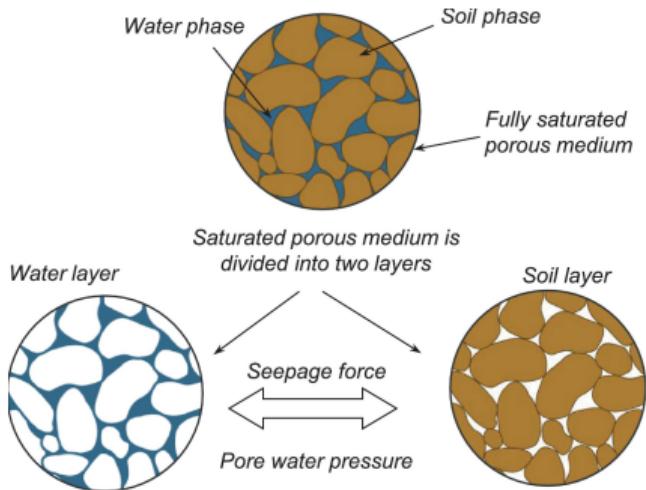


Volume fractions α and tortuosity λ in sleeping, anaesthesized (KX) and awake mice.

[Xie et al., Science, 2013)]

The brain as a poroelastic medium (in a fluid environment)

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

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

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

Linearly elastic stress-strain relationship:

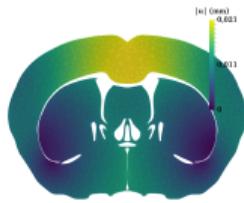
$$\sigma(u) = 2\mu \varepsilon(u) + \lambda \operatorname{div}(u)I. \quad (2)$$

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.

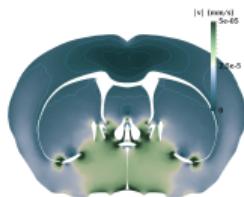
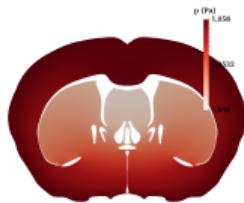
[Biot, J. Appl. Phys., 1941]

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



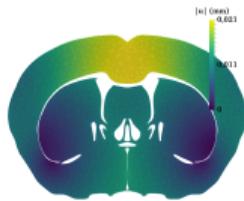
Find the displacement u and the pressure p such that :

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



[Biot (1941), Murad, Thomée and Loula (1992-1996), Phillips and Wheeler (2007-2008), and many others]

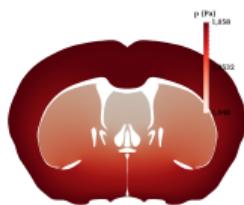
In the limits of incompressibility (and impermeability), Biot's equations reduce to the elasticity & Darcy (and Stokes) equations



Find the displacement u and the pressure p such that :

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

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

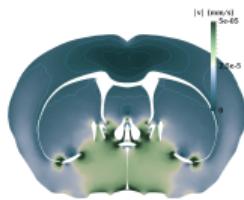


Low-storage, incompressible regime: $s = 0$, $\lambda \rightarrow \infty$:
 $\operatorname{div} u \rightarrow 0$, the system decouples

(Darcy)

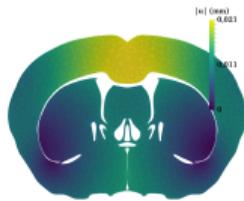
$$-\operatorname{div} \kappa \operatorname{grad} p = g,$$

(Elasticity)



[Biot (1941), Murad, Thomée and Loula (1992-1996), Phillips and Wheeler (2007-2008), and many others]

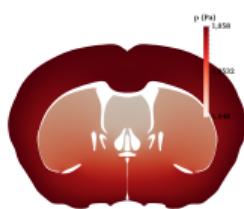
In the limits of incompressibility (and impermeability), Biot's equations reduce to the elasticity & Darcy (and Stokes) equations



Find the displacement u and the pressure p such that :

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

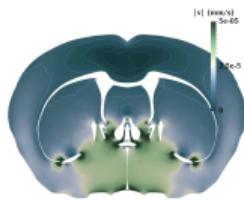
$$\dot{s}p + \alpha \operatorname{div} \dot{u} - \operatorname{div} \kappa \operatorname{grad} p = g.$$



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

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

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



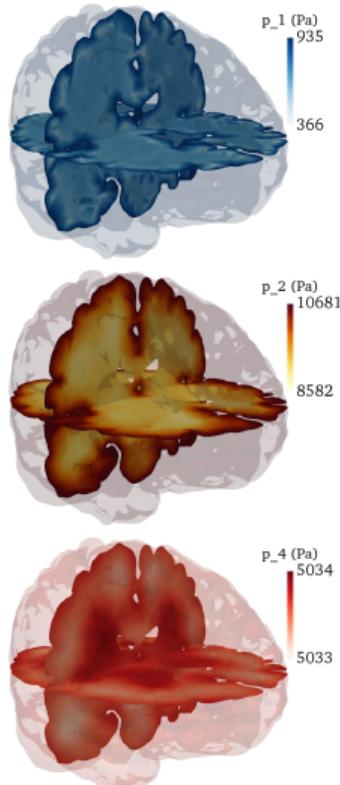
Low-storage, impermeable regime: $s = 0$, $\kappa \rightarrow 0$:

$$-\operatorname{div}(2\mu\varepsilon(u) - \alpha p I) = f, \quad (\text{Stokes})$$

$$\operatorname{div} u = 0.$$



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

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

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

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)]

Finite element methods for Stokes-Biot coupled problems

Simulating brain pulsatility

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

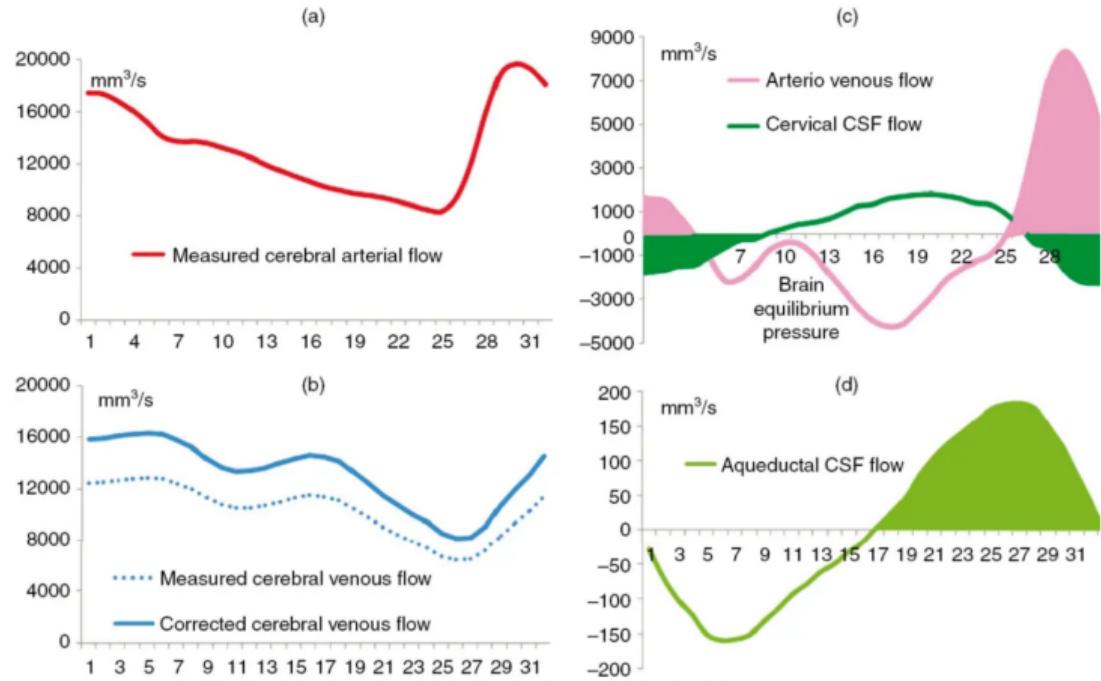
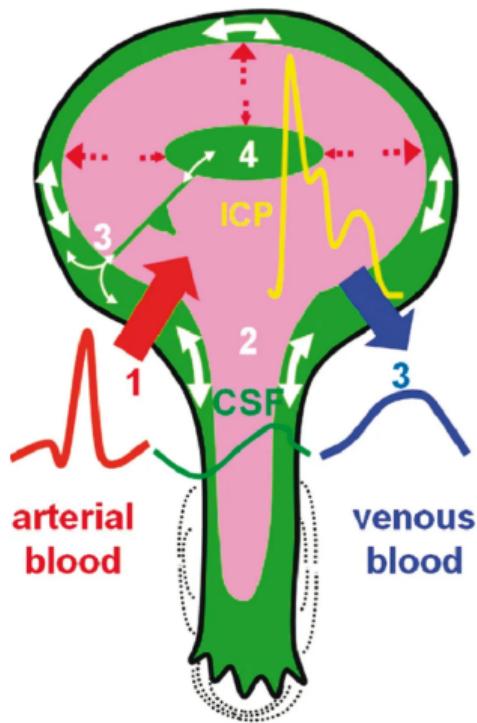
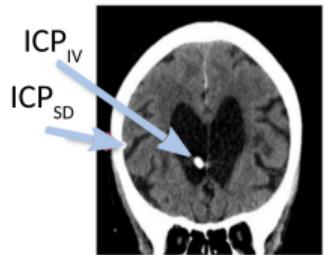
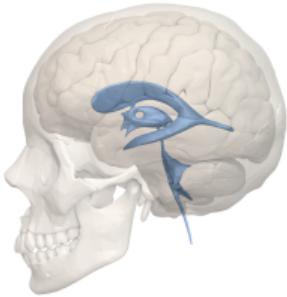


Figure 12.4 CSF and cerebral blood flow during the cardiac cycle in healthy adults (see text for full explanation).

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

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

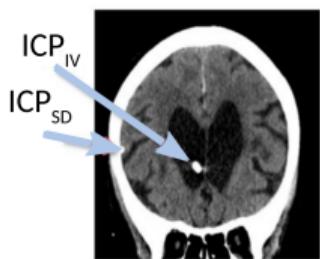
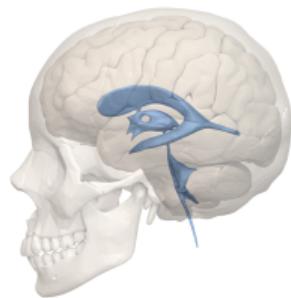


Long term ICP measurements

[Eide and Sæhle, 2010]

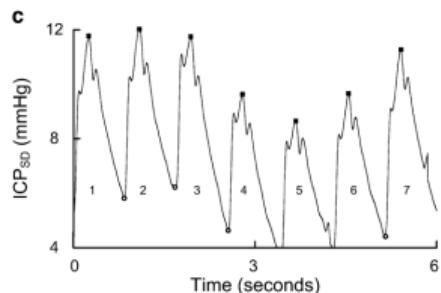
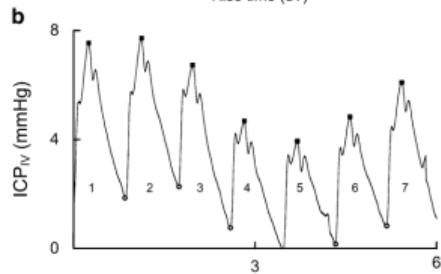
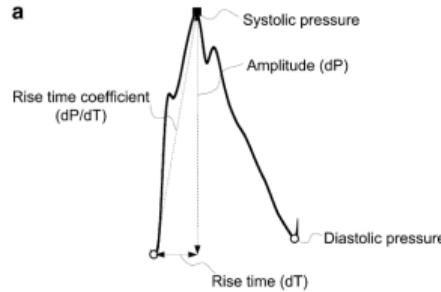
ICP (gradients) pulsate in sync with cardiac and respiratory cycles

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



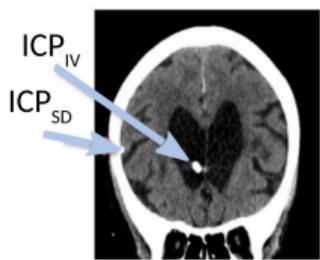
Long term ICP measurements

[Eide and Sæhle, 2010]



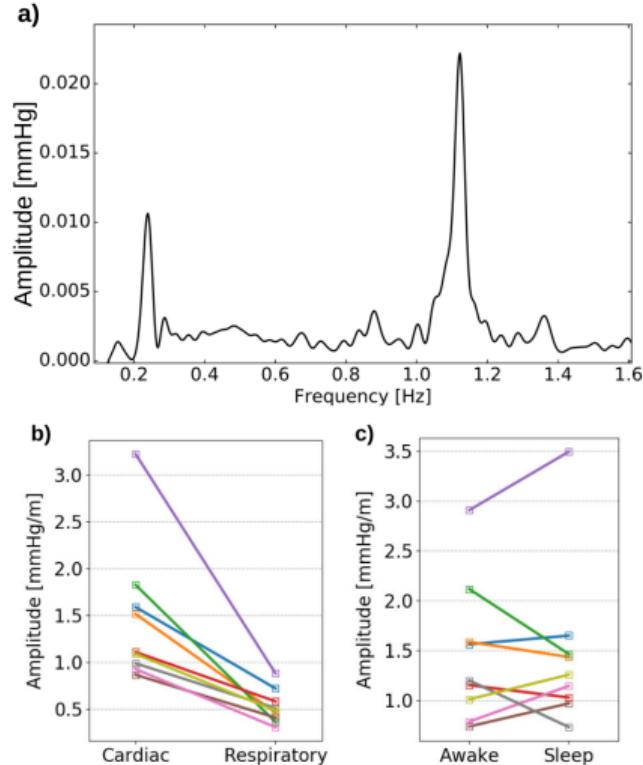
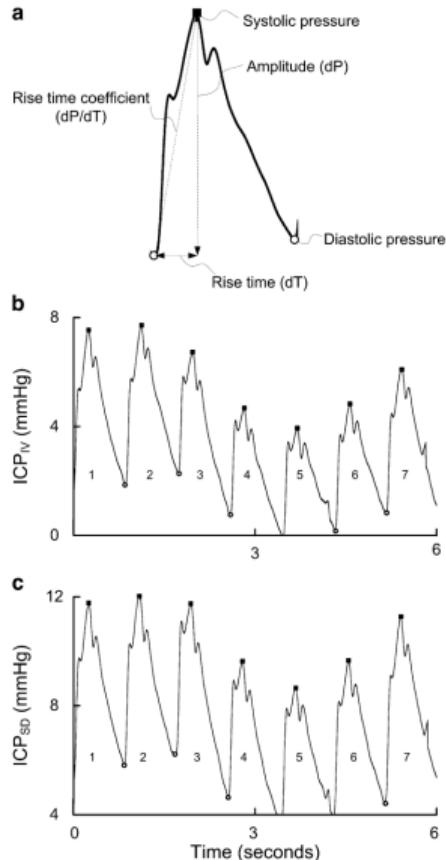
ICP (gradients) pulsate in sync with cardiac and respiratory cycles

[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 induce pulsating CSF flow

[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 induce pulsating CSF flow

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

Pulsating ICP gradients induce pulsating CSF flow

[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:

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

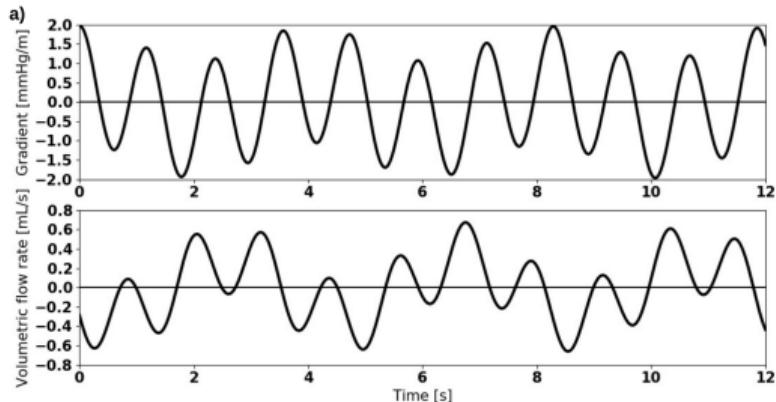
Analytic solution(s) in axisymmetric pipe

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



Pulsating ICP gradients induce pulsating CSF flow

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

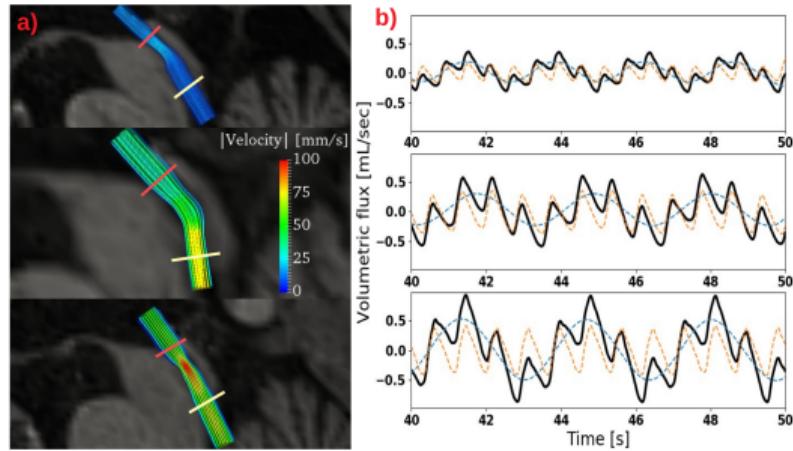
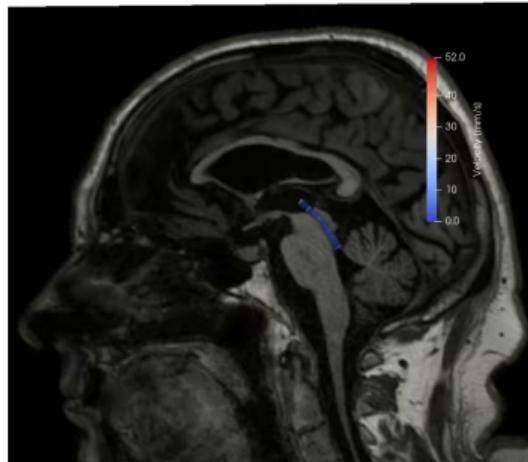
Incompressible Navier-Stokes

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

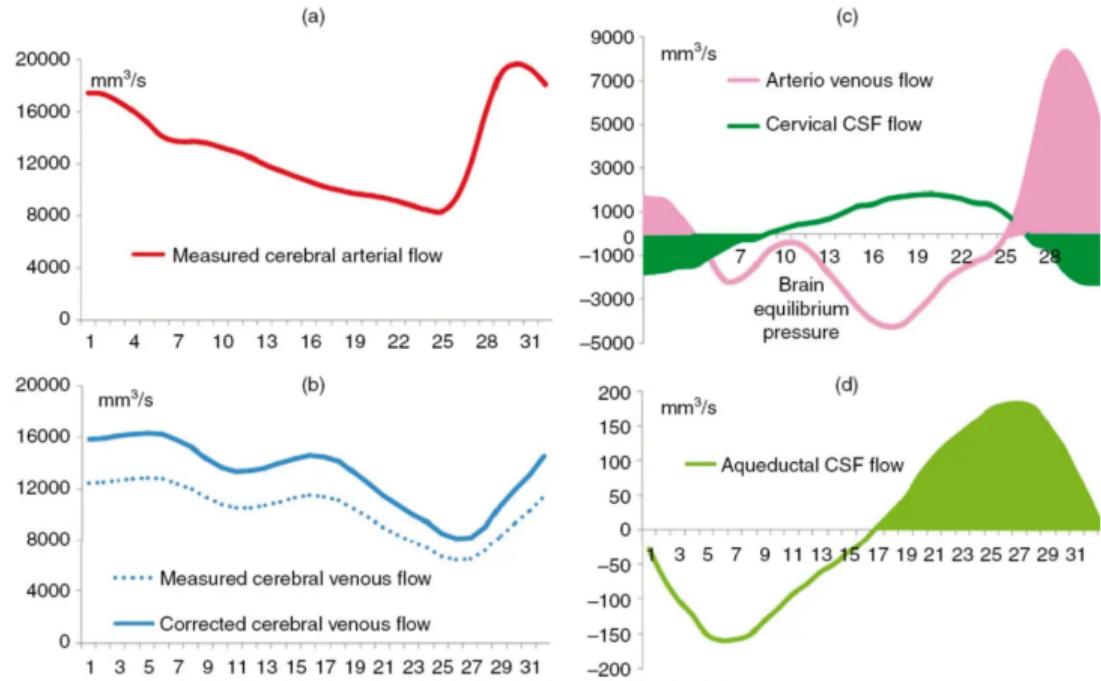
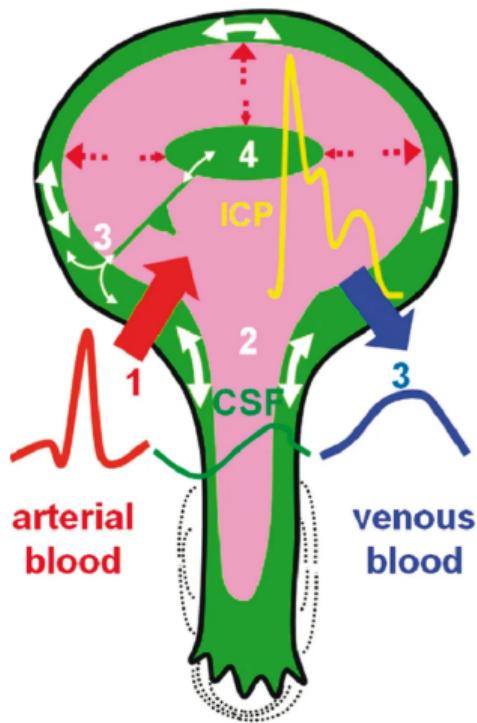


Figure 12.4 CSF and cerebral blood flow during the cardiac cycle in healthy adults (see text for full explanation).

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

[Causemann, 2021]

Biot's equations in Ω_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.

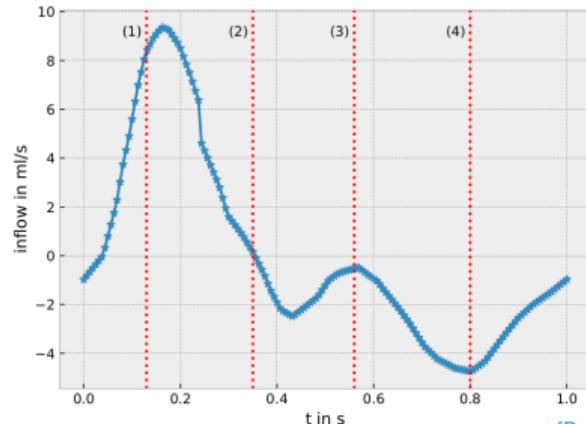
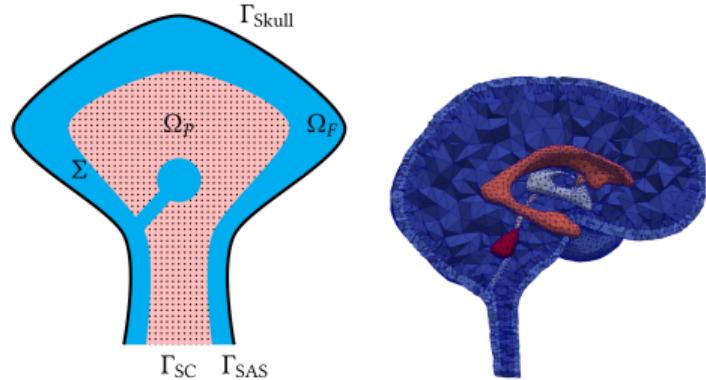
Stokes' equations in Ω_F

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

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

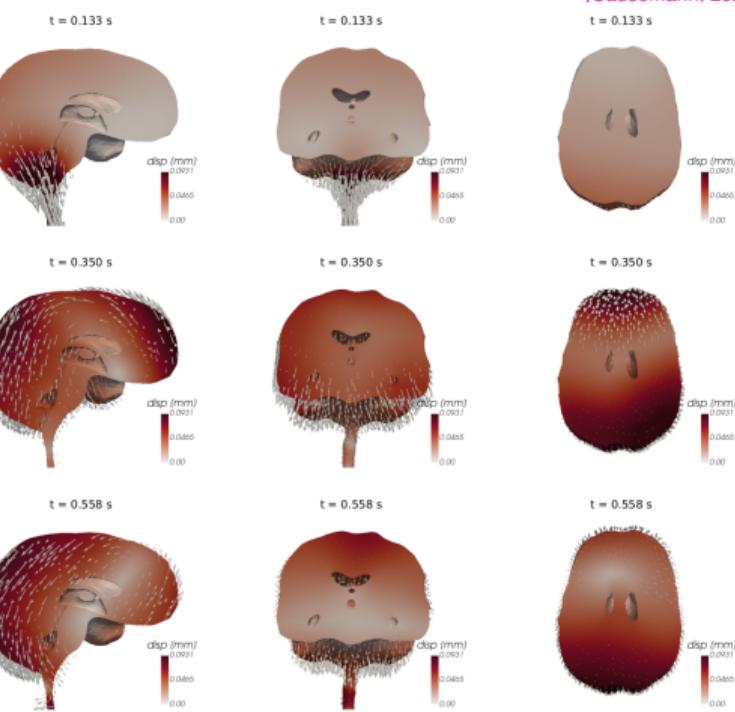
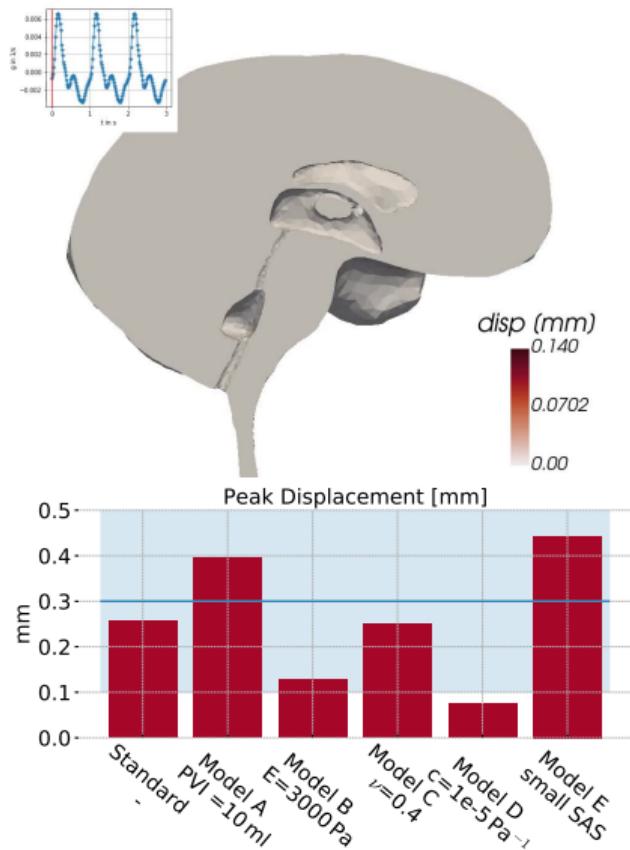
$$\operatorname{div} v = 0$$

Interface (resp. boundary) conditions at Σ (resp. Γ).



[Balédent (2014)]

Pulsatile blood inflow yields heterogeneous displacement patterns

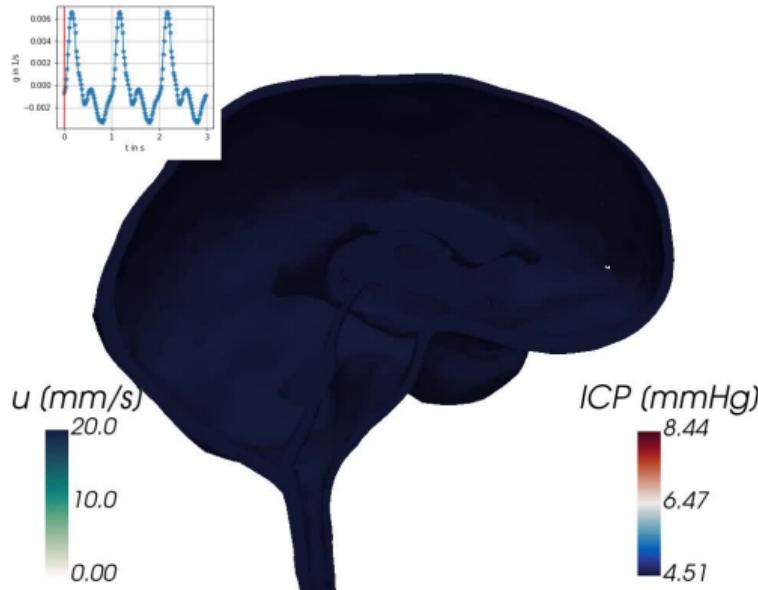


MRI studies estimate peak brain tissue displacement $\sim 0.1\text{--}0.5 \text{ mm}$.

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

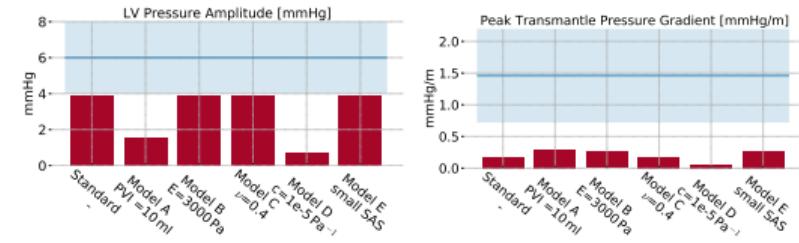
.. nearly homogeneous pressures but complex fluid flow

[Causemann, 2021]

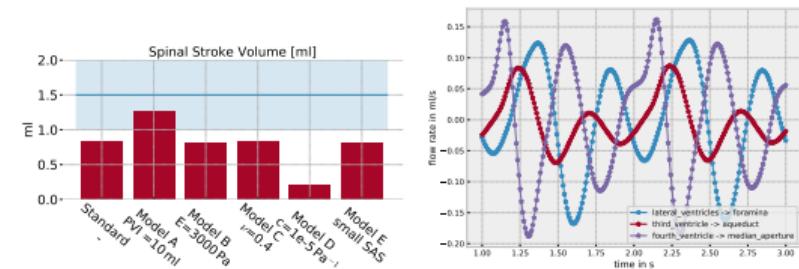


Simulations offer unprecedented detail.

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 openly available via GitHub

The screenshot shows a GitHub repository page for the user 'meg-simula' with the repository name 'mri2fem-lectures'. The repository is private. The main navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. Below the navigation bar, there are tabs for Code, Issues, Pull requests, Actions, Projects, Security, Insights, and Settings. The 'Code' tab is currently selected. The repository summary shows the 'main' branch, 1 branch, and 0 tags. On the right side, there are buttons for Go to file, Add file, and Code (with a dropdown menu). A modal window is open over the repository content, showing the 'Clone' section with options for HTTPS, SSH, and GitHub CLI. It also displays the command 'gh repo clone meg-simula/mri2fem-lectures' and a note about working fast with the official CLI. Other repository files listed include 'slides', 'README.md', and 'abstract.txt'. The 'README.md' file is currently selected.

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

Mathematical modeling of the human brain – book available on GitHub

by KA Mardal, ME Rognes, TB Thompson and LM Valnes; Simula SpringerBrief on Computing (2021)

The screenshot shows the GitHub repository page for `kent-and/mri2fem`. The repository has 1 branch and 0 tags. The last commit was made by `kent-and` 5 days ago, adding Marie's latest modifications. The repository has 2 forks and 1 star. It includes sections for About, Releases, and Packages.

About
mri2fem

Releases
No releases published

Packages
No packages published

Branch	Commits	Last Commit
master	4	5 days ago

File	Commit Message	Time
book	added Marie's latest modifications	5 days ago
mri2fem	added Marie's latest modifications	5 days ago

[Mardal et al (2021): <https://github.com/kent-and/mri2fem>]

Resources (data, software) are openly available with our Zenodo community

The screenshot shows the Zenodo search interface. At the top, there is a blue header bar with the Zenodo logo, a search bar, an upload button, and a communities link. On the right side of the header are 'Log in' and 'Sign up' buttons. Below the header, the main content area has a title 'Mathematical modeling of the human brain - from magnetic resonance images to finite element simulation'. This title is also present in a search bar below it.

Recent uploads

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

December 22, 2020 (v0.7) 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 December 23, 2020

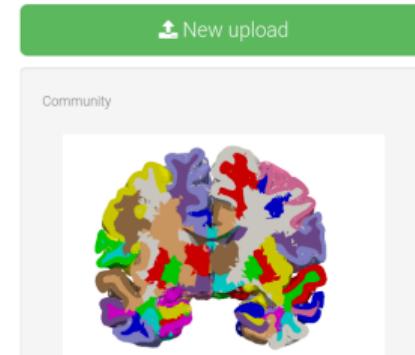
December 22, 2020 (v1.0) Dataset Open Access

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)

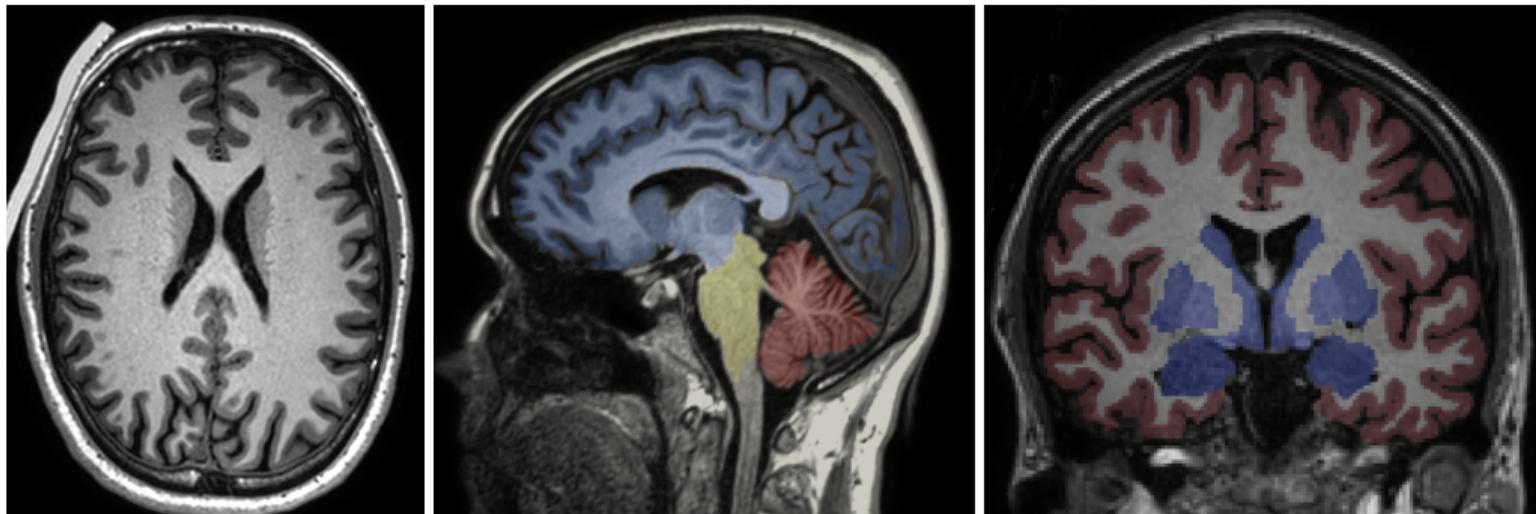
Uploaded on December 23, 2020



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

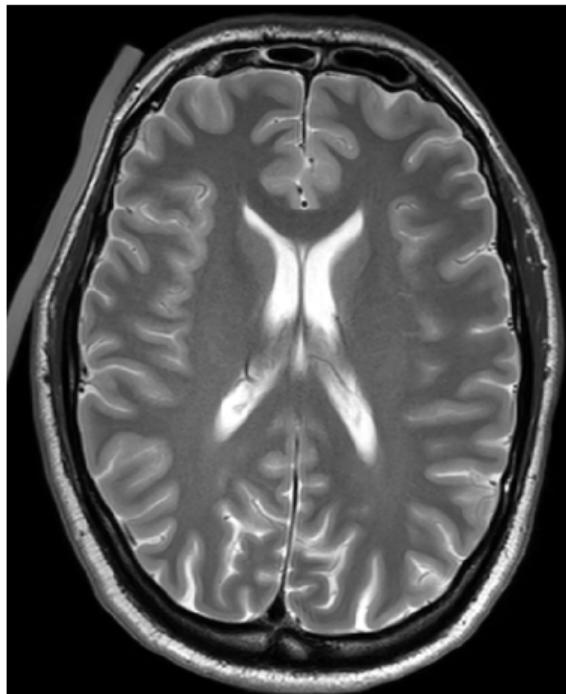
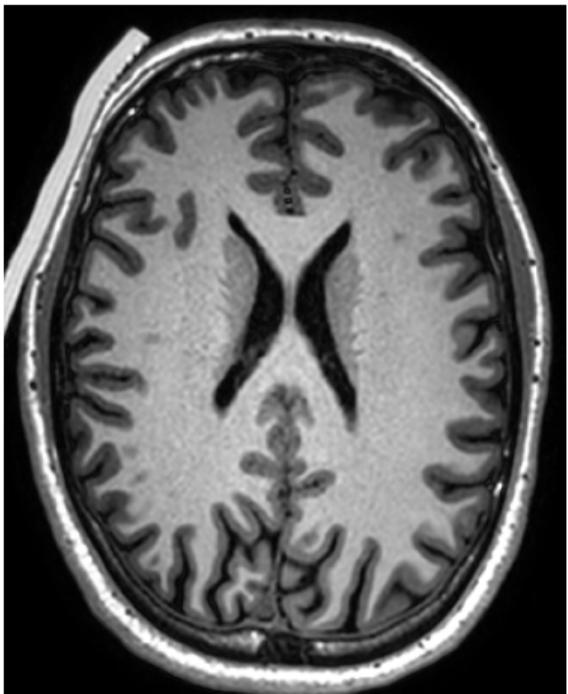
Data and software associated with "Mathematical modeling of the human brain - from magnetic resonance images to finite element simulation" by Mardal, Rognes, Thompson and Valnes (2021).

T1-weighted MRI reveals brain structure



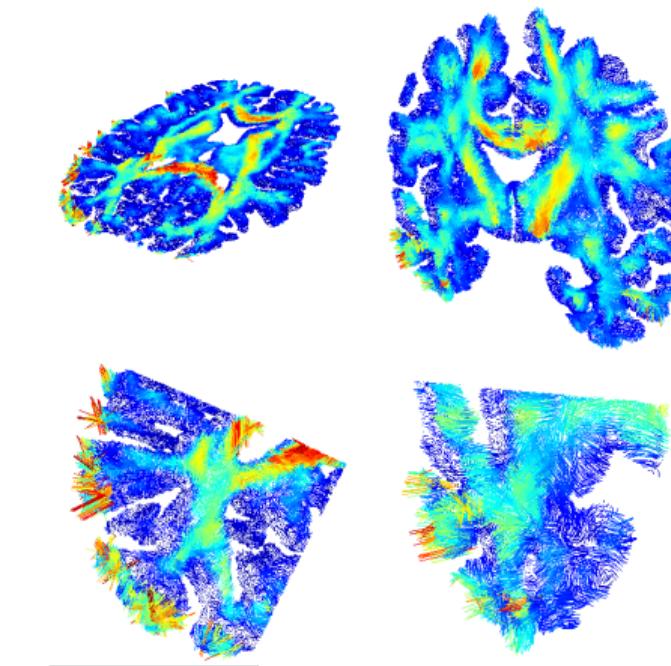
Axial, sagittal and coronal cross-sections. T1w MRI: fat gives high signal intensity (light/white); fluids give low signal intensity (dark/black).

T2-weighted MRI reveals brain structure and fluids



T1w (left) versus T2w (right). In T2w MRI: fluids gives high signal intensity (light/white)

Diffusion-tensor MRI reveals water movement and directionality



In DTI: high signal intensity indicates high degree of anisotropy

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.

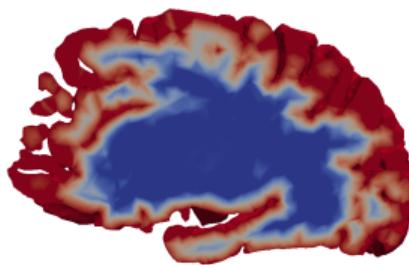
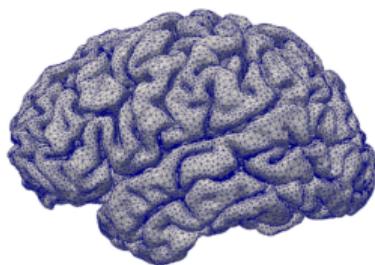
Many possibilities including the basic DicomBrowser

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

Video example: 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 example: viewing the FreeSurfer output

```
$ cd mri2fem-dataset/freesurfer/ernie/surf  
$ freeview # Open lh.pial  
$  
$ mris_convert ./lh.pial pial.stl  
$ paraview # Open lh.pial.stl
```

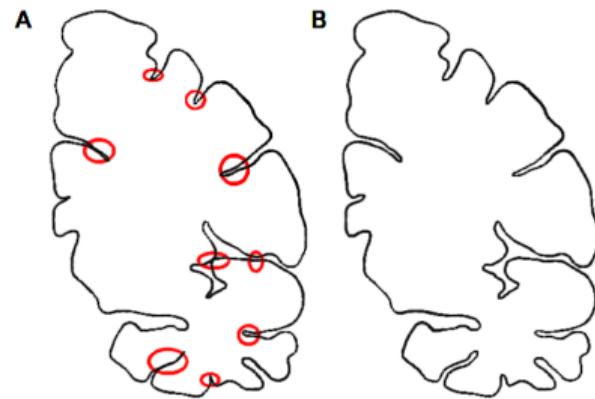
The surfaces may need enhancement to be suitable for finite element meshes

In practice, brain surfaces generated from T1 images

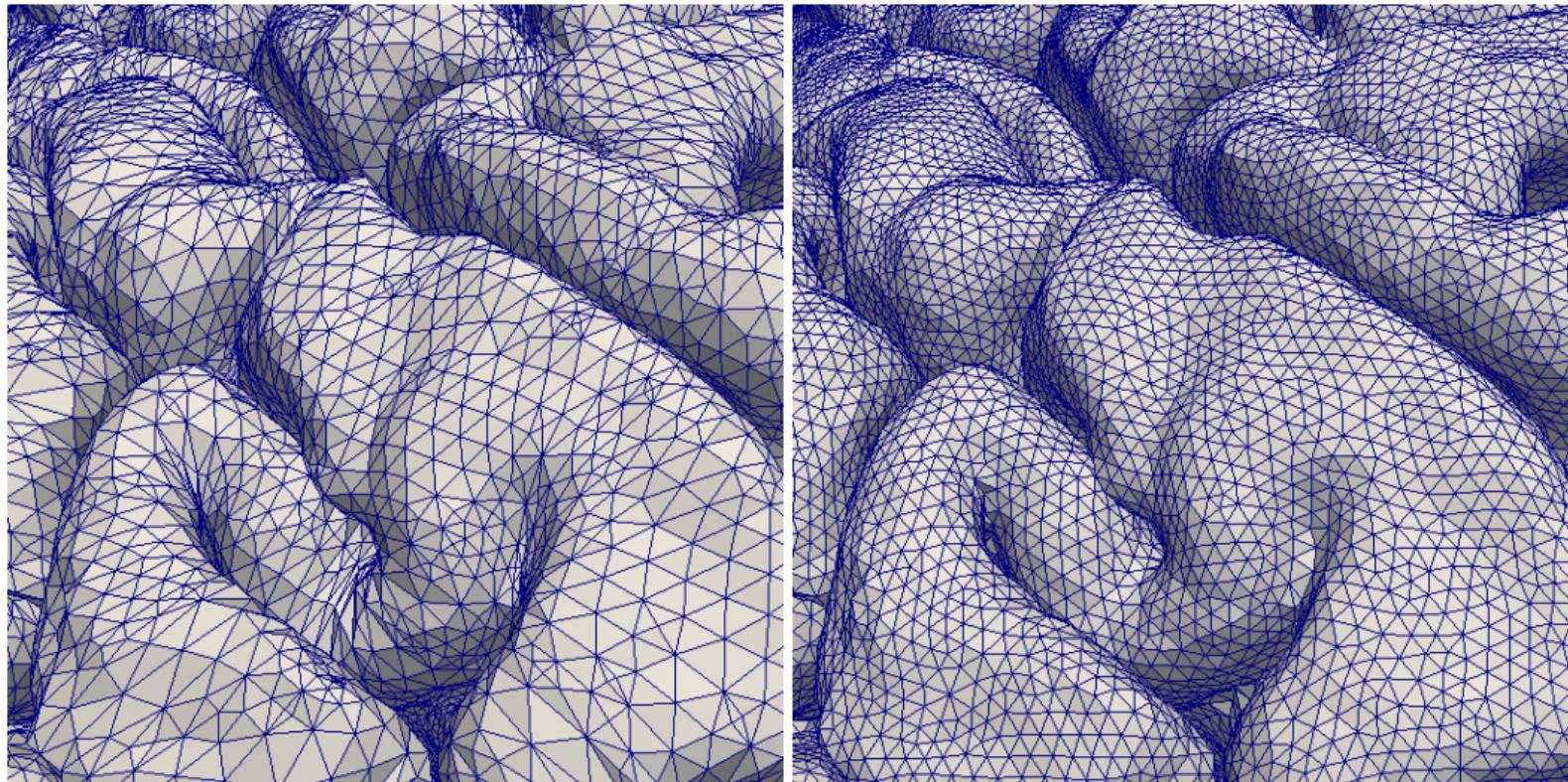
- ▶ may have unphysiologically sharp corners,
- ▶ may include triangles with very large aspect ratios,
- ▶ have topological defects such as holes, and
- ▶ may self-intersect or overlap with other surfaces.

Result: Low quality meshes (if any).

Fix: Enhance surface quality prior to meshing

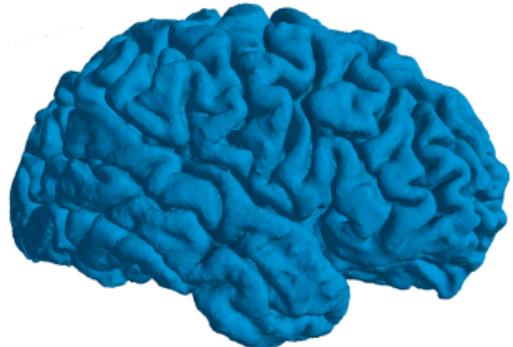


SVM-Tk (wrapping CGAL) includes utilities for remeshing surfaces

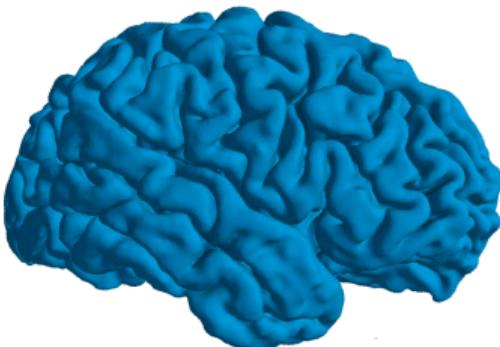


[Mardal et al (2021, Chapter 3.2.1); mri2fem/chp3/remesh_surface.py]

SVM-Tk includes utilities for smoothing surfaces



Original pial surface



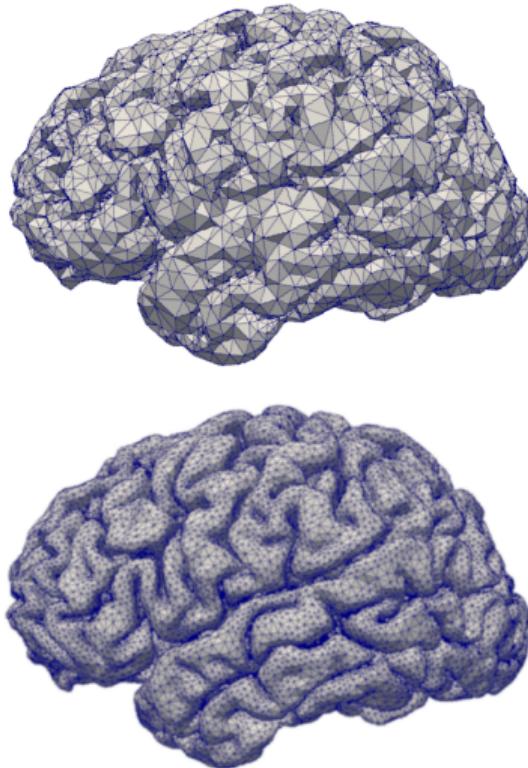
after Taubin smoothing



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)

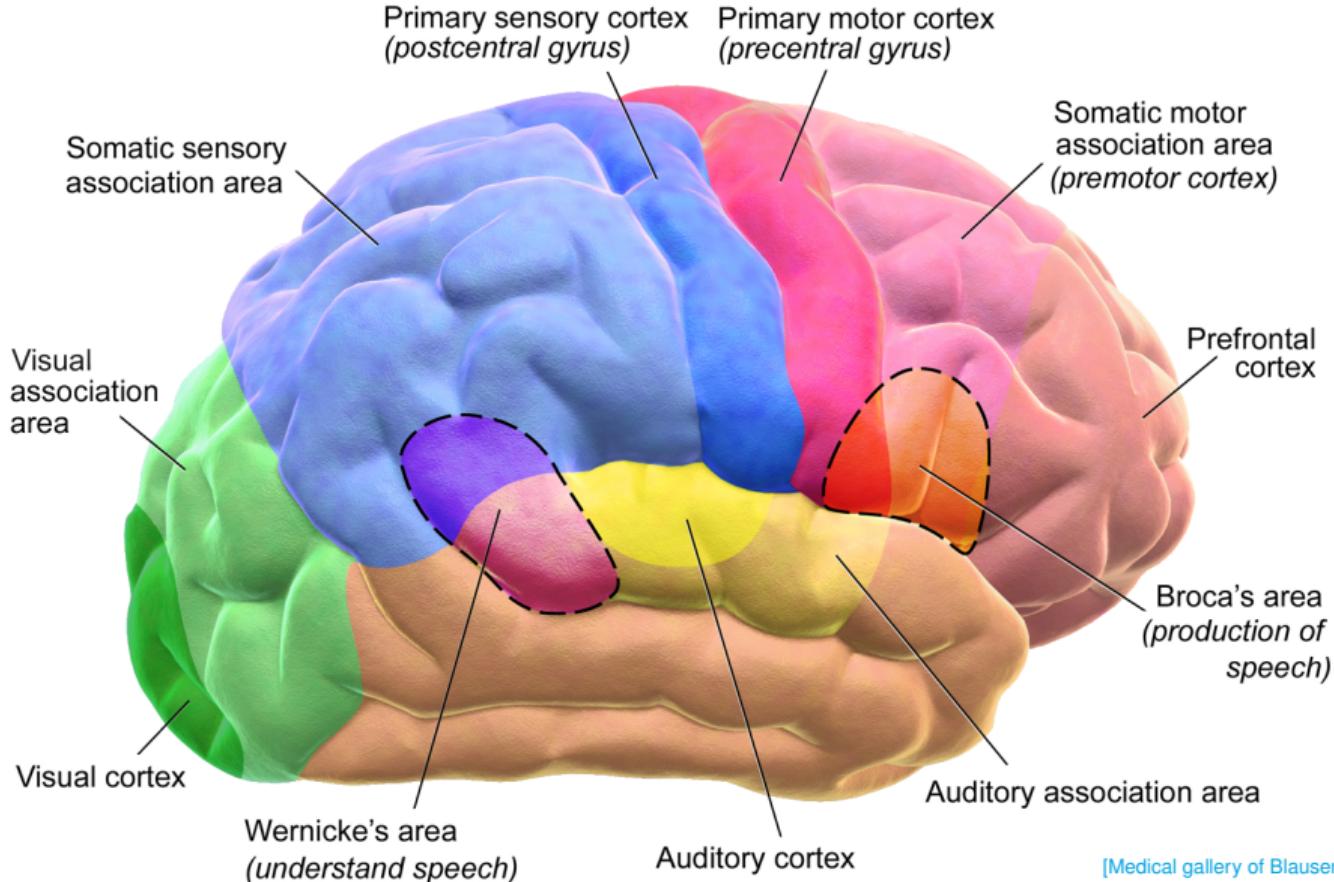
    # Write the mesh to the output file
    domain.save(output)

# 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 example: creating a mesh of the left hemisphere

Brain parcellations

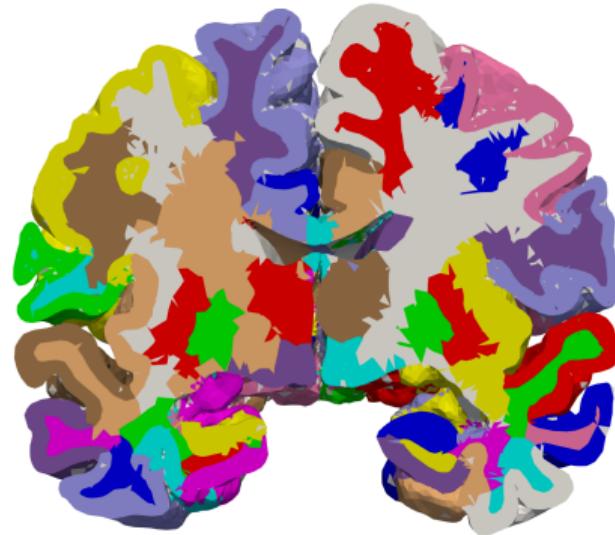


[Medical gallery of Blausen Medical 2014]

The FreeSurfer segmentation process (recon_all) generates volumetric brain parcellations



Parcellation as generated by FreeSurfer and visualized using Freeview.



Same parcellation transferred onto the FEniCS brain mesh and visualized using Paraview.

Converting between voxel space and mesh coordinates

NiBabel is a useful Python module for operating on image data



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

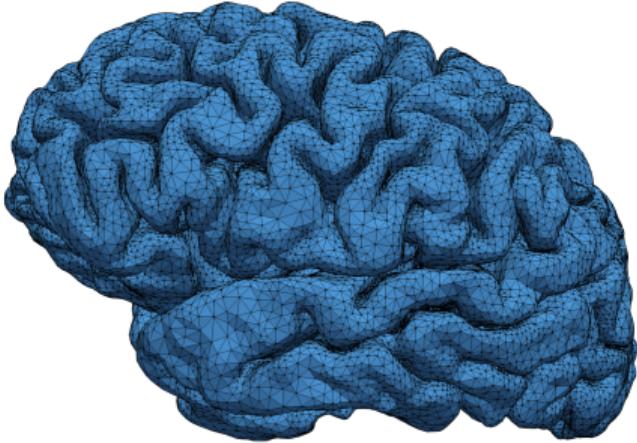
```
import numpy
import nibabel
from nibabel.affines import apply_affine

# 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

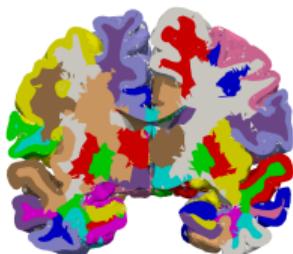
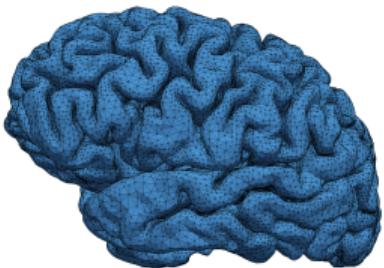


Brain mesh visualized in ParaView.

```
# 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())
```

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()[:]

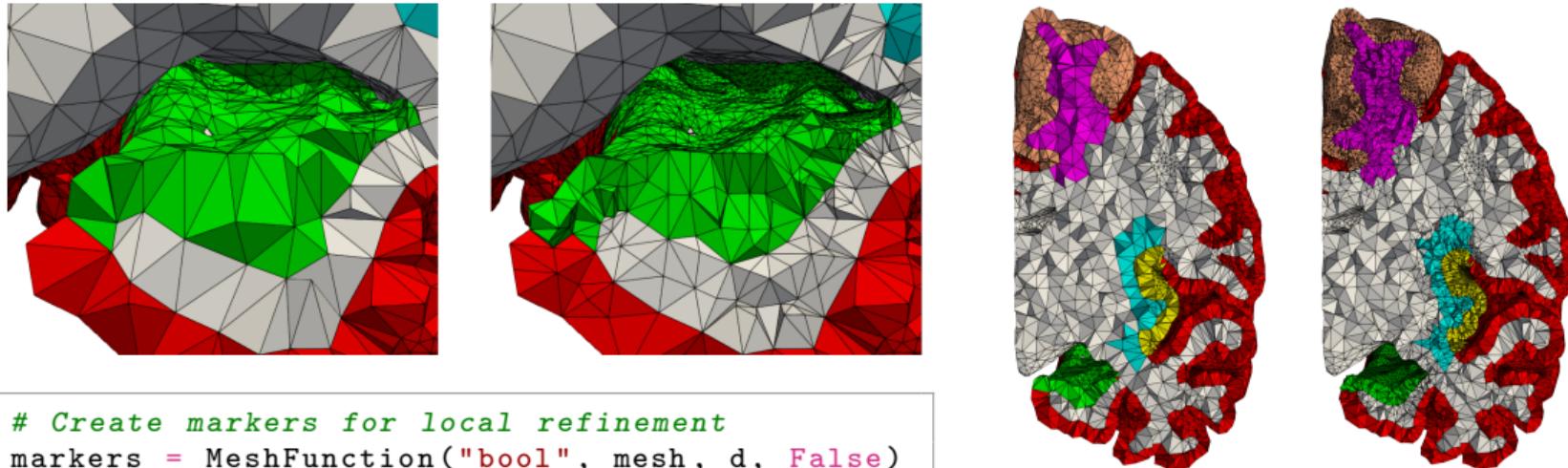
    # Convert to voxel space
    ijk = apply_affine(ras2vox, xyz)

    # 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])
```

Video example 1: Brain parcellations

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)
```

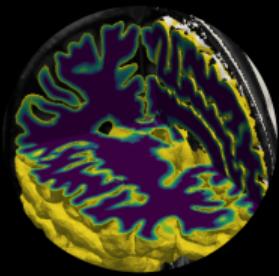
Illustration of left hemisphere meshes with different parcellation regions at different resolutions. The meshes on the right are local refinements of the meshes on the left. Zoomed in view of the hippocampus region (green).

Video example: simulating diffusion using FEniCS

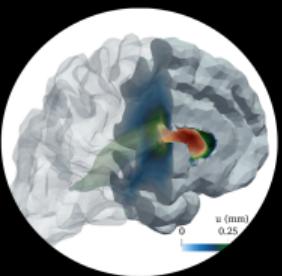
Igor, bring me the brain! Want to give it a try? Natural first steps:

1. Download the slides from this lecture:
<https://github.com/meg-simula/mri2fem-lectures>.
2. Download the mri2fem data set and software from Zenodo and inspect the contents using the tools discussed.
3. Download the book: Mardal et al (2021): <https://github.com/kent-and/mri2fem>.
4. Install software dependencies following the instructions in Chapter 2 of Mardal et al (2021).
5. Try running some of the sample code in Chapter 3 of Mardal et al (2021).
6. Simulate something other than diffusion? Give it a try!

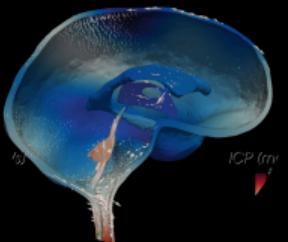
Concluding remarks



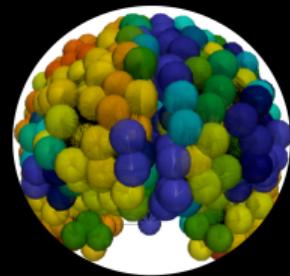
Solute transport



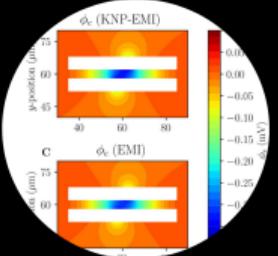
Brain mechanics



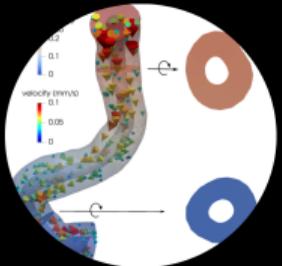
CSF flow



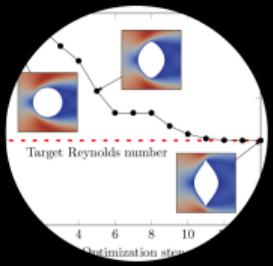
Neurodegeneration



Ions and osmosis



Model reduction



Optimal control



Software

