ETH zürich

IT'IS FOUNDATION

EM-Neuro Modeling Across Scales for Bioelectronic Medicine

Lecture 4: EM Field Simulation Fundamentals

Esra Neufeld[*] and Taylor Newton[*†]

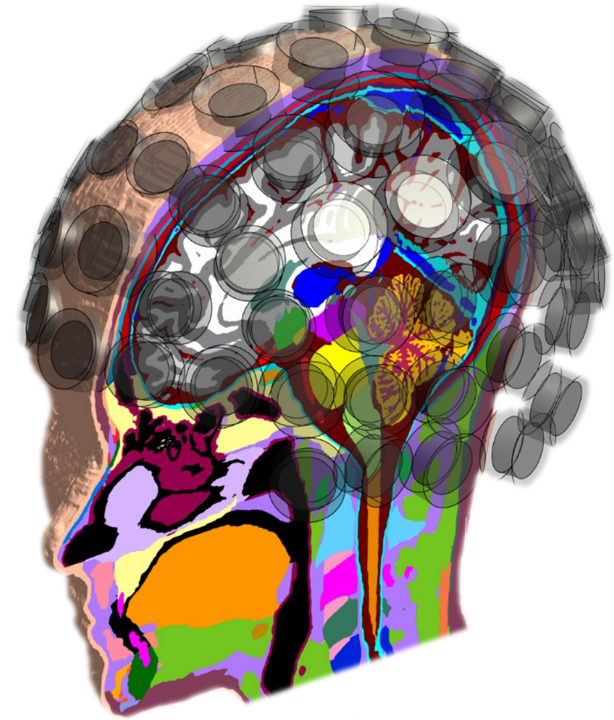
[*]IT'IS Foundation for Research on Information Technologies in Society

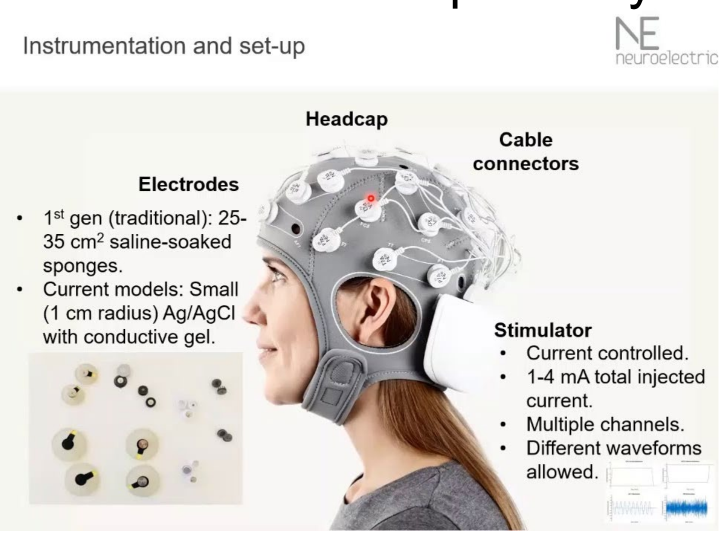
[†]Integrated Systems Laboratory, ETH Zurich

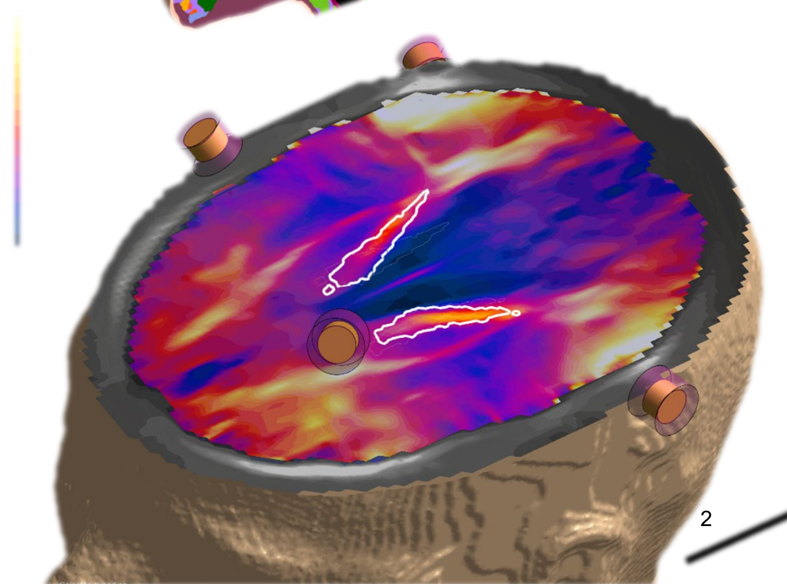
Trascranial Electric Stimulation

ETH zürich · IT'IS FOUNDATION

- non-invasive brain stimulation method using weak electrical currents through scalp electrodes to modulate neural activity
  - tDCS: direct current stimulation
  - tACS: alternating current stimulation
- affecting brain region excitability to study cognition and treat neurological / psychiatric conditions
- influences plasticity and network function







https://www.youtube.com/watch?v=zbVBfNKDGJY

Outline for today

- **Lecture Overview**
- **Maxwell's Equations & Quasistatic Approx.**
- **Discretization**
- **Finite Element Method**
- **Solver Numerics**
- **Finally: Simulating tES**
- **Summary of Today's Lecture & Outlook**

Outline for today

- **Lecture Overview**
- Maxwell's Equations & Quasistatic Approx.
- Discretization
- Finite Element Method
- Solver Numerics
- Finally: Simulating tES
- Summary of Today's Lecture & Outlook

ETHzürich

Course Outline

IT'IS FOUNDATION

DATE	LECTURE THEME
19.02	Motivation, logistics & tooling (EN, TNE)
26.02	Ion channels & membranes (EN)
05.03	Axon models, activating functions & electrical stimulation (EN)
12.03	EM field simulation fundamentals & coupled EM-neuro workflows (EN)
19.03	Peripheral nerves & interfaces for bioelectronic medicine (EN)
26.03	Spinal cord stimulation for neuroprosthetics and pain management & low-frequency exposure safety (TNE)
02.04	Morphology, synapses, microcircuits; point vs spiking networks (TNE)
09.04	No class: Easter break
16.04	Neural mass & whole brain models; hybridization (TNE)
23.04	Recording modalities, signal content & the reciprocity theorem (TNE)
30.04	Non invasive brain stimulation & temporal interference (TNE)
07.05	Image based/personalized treatment planning and optimization (EN)
14.05	No class: Ascension Day
21.05	Verification, validation, UQ, and reproducibility (EN)
28.05	Project presentations & synthesis (EN, TNE)

Room: ETZ E7
13:15-14:00 Lecture
14:00-14:15 Break
14:15-15:00 Lecture
14:00-14:15 Break
15:15-16:00 Exercise

Lecture Recordings & Slides

Provided Here
(will successively appear)

ETH zürich

Exercise Outline

IT'IS FOUNDATION

DATE	EXERCISE THEME
19.02	"Hello Neuron": integrate-and-fire in Python/NEURON
26.02	Point neuron phase portrait; basic time integration numerics
05.03	Recruitment prediction for myelinated axon using AF/GAF
12.03	EM (FEM) modeling of transcranial brain stimulation
19.03	Stimulation selectivity and signal content modeling for nerve interfaces
26.03	Guest (SCS – NeuroRestore)
02.04	Mini project work
09.04	No class: Easter break
16.04	Guest (Neuromodulation Spin-Off – Z43)
23.04	Mini project work
30.04	Guest (NIBS – Kinderspital)
07.05	Mini project work
14.05	No class: Ascension Day
21.05	Mini project work
28.05	Project presentations

Room: ETZ E7

13:15-14:00 Lecture

14:00-14:15 Break

14:15-15:00 Lecture

14:00-14:15 Break

15:15-16:00 Exercise

Course objectives

At the end of this lecture, you will have

- an understanding for the quasistatic approximations of Maxwells equations of relevance to neurostimulation
- insights into the advantages and disadvantages of structured and unstructured discretization
- learned about the finite element method and numerical aspects of solving the resulting linear systems

- The exercise will revolve around practical FEM modeling of head exposure

Outline for today

- Lecture Overview
- **Maxwell's Equations & Quasistatic Approx.**
- Discretization
- Finite Element Method
- Solver Numerics
- Finally: Simulating tES
- Summary of Today's Lecture & Outlook

Maxwell's Equations

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0}$$
$$\nabla \cdot \mathbf{B} = 0$$
$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$
$$\nabla \times \mathbf{B} = \mu_0 \left(\mathbf{J} + \varepsilon_0 \frac{\partial \mathbf{E}}{\partial t} \right)$$

macroscopic formulation:

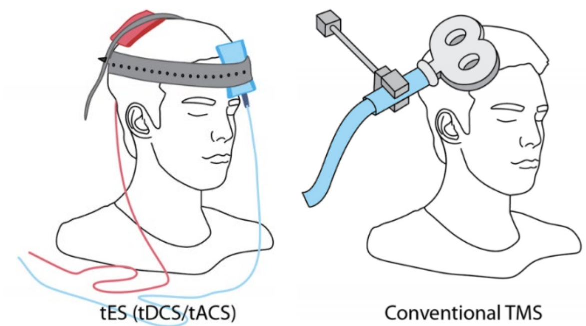
$$\nabla \cdot \mathbf{D} = \rho_f$$
$$\nabla \cdot \mathbf{B} = 0$$
$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$
$$\nabla \times \mathbf{H} = \mathbf{J}_f + \frac{\partial \mathbf{D}}{\partial t}$$
$$\mathbf{D} = \varepsilon \mathbf{E}, \quad \mathbf{H} = \frac{1}{\mu} \mathbf{B}$$

in simple words:

- Gauss' law: charges are sources of electric fields
  - E-field lines only end at charges
- Gauss' law for magnetism: there are no magnetic monopoles and thus no sources of magnetic field
  - B-field lines are closed
- Faraday's law: changing magnetic fields induce circulating E-fields
  - closed E-field lines surround changing magnetic flux
- Ampère-Maxwell law: both currents and changing E-fields produce circulating B-fields around them
  - closed B-field lines surround currents and changing E-fields
- gives rise to manifold phenomena, such as electromagnetic waves

Quasistatic Approximations

$$\nabla \cdot \mathbf{D} = \rho_f \qquad \mathbf{D} = \varepsilon \mathbf{E}, \quad \mathbf{H} = \frac{1}{\mu}\mathbf{B}$$
$$\nabla \cdot \mathbf{B} = 0 \qquad \mathbf{J} = \sigma \mathbf{E} + \mathbf{J}$$
$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$
$$\nabla \times \mathbf{H} = \mathbf{J}_f + \frac{\partial \mathbf{D}}{\partial t}$$



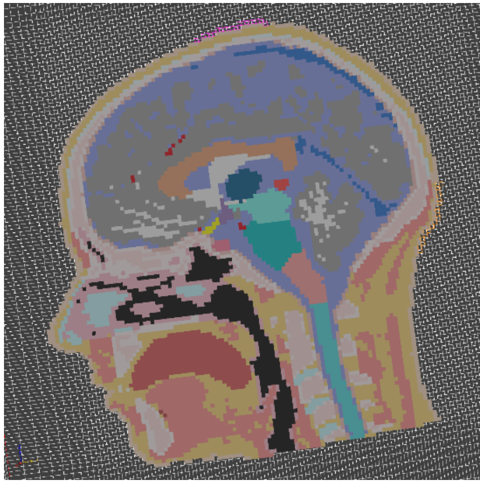
- for harmonic fields at 'low frequencies', time derivatives can become negligible and the equations decouple
  - 'low frequencies' is related to domain size; the wavelength must be must larger $|\omega^2 \tilde{\epsilon} \mu d^2| \ll 1 \iff \left(\frac{d}{\lambda}\right)^2 \ll 1$ $\tilde{\epsilon} := \epsilon_r \epsilon_0 + \frac{\sigma}{j\omega}$
- for the same reason, the E-field becomes nearly irrotational ($\nabla \times \mathbf{E} \approx 0$)
  - therefore, an electric potential can be defined with $\mathbf{E} = -\nabla \phi$
- without source current, one obtains the *electro-quasistatic equation*: $\nabla \cdot \tilde{\epsilon} \nabla \phi = 0$
  - when ohmic currents dominate over displacement currents ($\sigma \gg \omega \epsilon$), this further simplifies to the *ohmic EQS*: $\nabla \cdot \sigma \nabla \phi = 0$
    which simply says than that current is preserved and there are no internal current sources
- with source currents, one obtains the *magneto-quasistatic eq.* $\nabla \cdot \tilde{\epsilon} \nabla \phi = -j\omega \nabla \cdot (\tilde{\epsilon} \mathbf{A}_0)$
  with the vector potential $\mathbf{A}_0(\vec{r}) = \frac{\mu_0}{4\pi} \int_\Omega \frac{\mathbf{J}_0(\vec{r}\,')}{|\vec{r} - \vec{r}\,'|} d^3\vec{r}\,'$

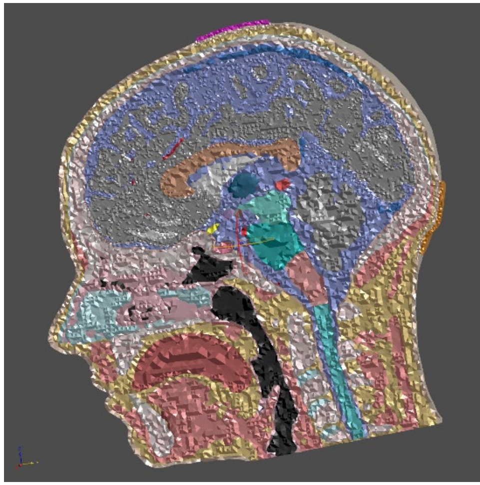
https://www.neuralengr.org/news/now-out-consensus-transcranial-electrical-and-magnetic-stimulation-tes-and-tms-for-addiction-medicine

- Lecture Overview
- Maxwell's Equations & Quasistatic Approx.
- **Discretization**
- Finite Element Method
- Solver Numerics
- Finally: Simulating tES
- Summary of Today's Lecture & Outlook

Discretization

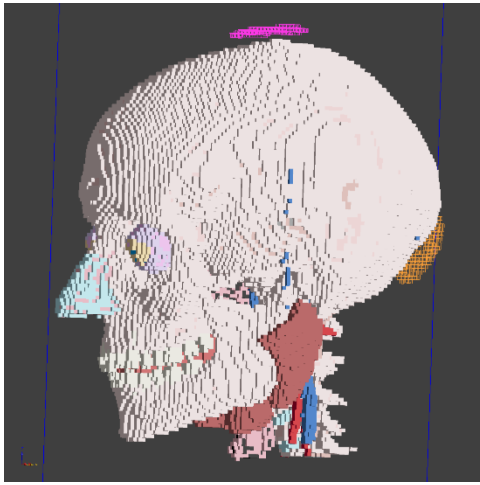
- To solve a continuums model, it must be **discretized into a finite number of degrees of freedom** (DoF)
- Advantages of **structured** grids (voxels)
  - Structure (fix neighborhood) facilitates solving (sparse matrix, conditioning, solving schemes, stencil parallelization…)
  - Robust discretization of complex geometries (anatomical)
- Advantages of **unstructured** meshes
  - Conformal (better interface/surface representation)
  - Adaptive (less DoF, good resolution where required)
- Does not determine numerical method (e.g., structured FEM)

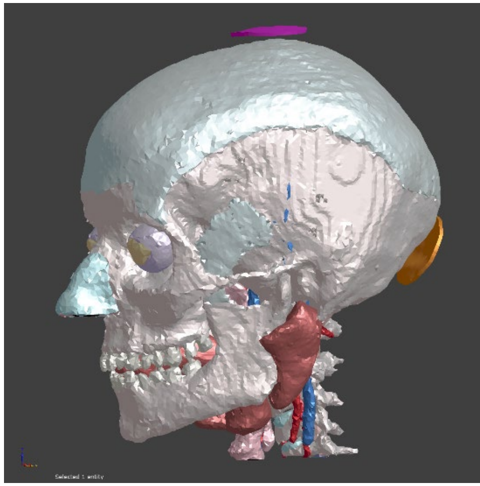




Discretization

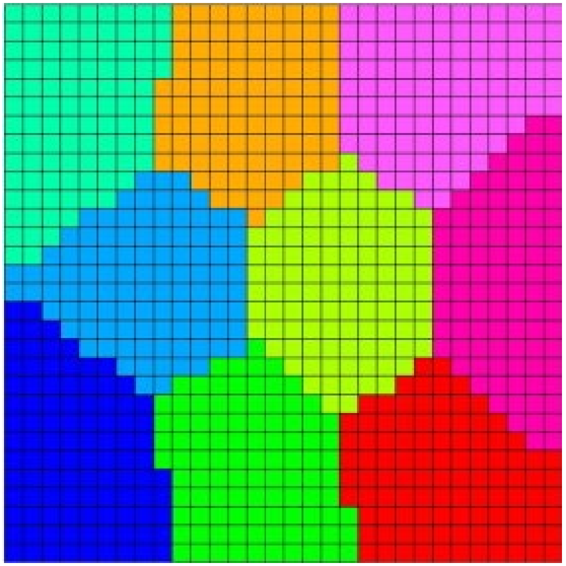
- To solve a continuums model, it must be **discretized into a finite number of degrees of freedom** (DoF)
- Advantages of **structured** grids (voxels)
  - Structure (fix neighborhood) facilitates solving (sparse matrix, conditioning, solving schemes, stencil parallelization…)
  - Robust discretization of complex geometries (anatomical)
- Advantages of **unstructured** meshes
  - Conformal (better interface/surface representation)
  - Adaptive (less DoF, good resolution where required)
- Does not determine numerical method (e.g., structured FEM)



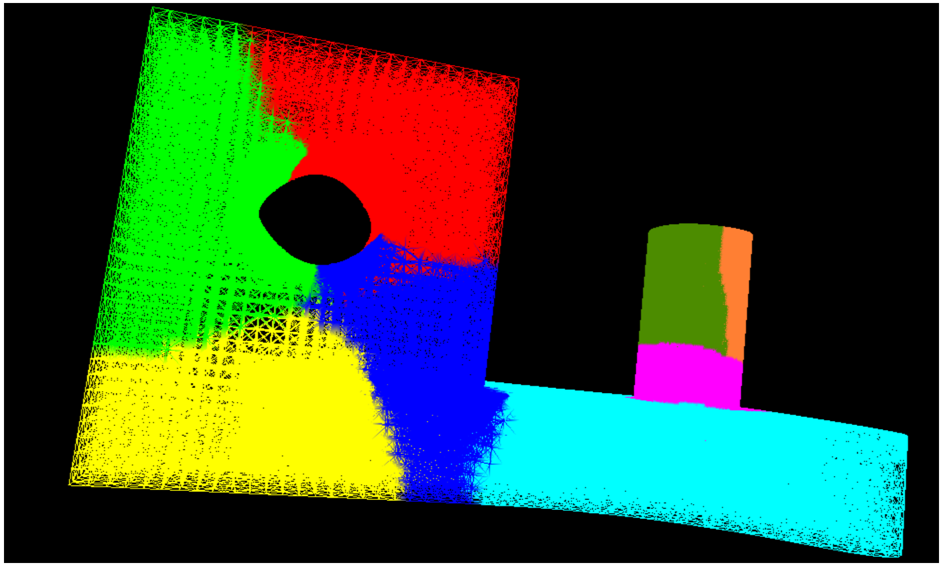


Partitioning

- Partitioning: division of the mesh into subdomains and mapping of each subdomain onto a processor



https://www.sciencedirect.com/science/article/pii/S0021999116306027



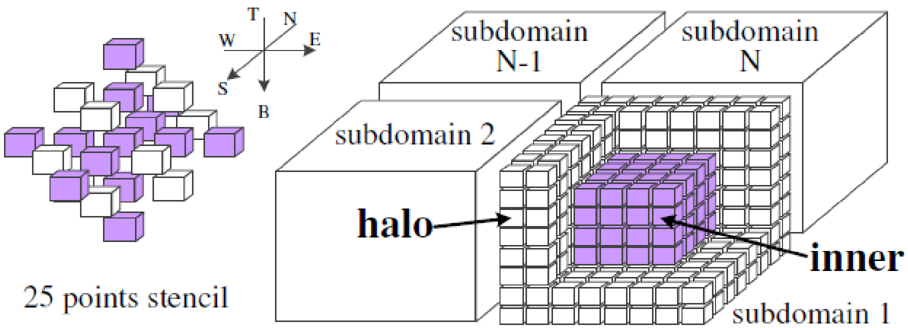
https://www.labri.fr/perso/pelegrin/scotch/

- Goal: minimize communication while maintaining load balance
- Load balance: all processors should contain roughly the same number of cells

Partitioning

Stencil Domain

- The computation on the halo region is repeated
- The various contributions are usually summed
- Load balancing trivial

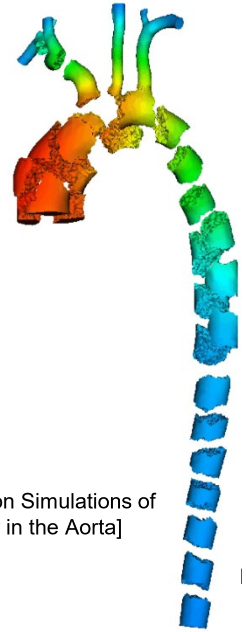


[A highly-efficient and green data flow engine for solving Euler atmospheric Equations]

- The partition can be identified by the (x,y,z) coordinates

Unstructured Grid Domain

- “map” associating degrees of freedom to a partition: “local” and “global” coordinates exist
- Load balancing not trivial
- Libraries: METIS/ParMETIS



[Fluid Structure Interaction Simulations of Physiological Blood Flow in the Aorta]

Spatial Discretization (Laplace)

$$\sigma \frac{\partial^2 \phi}{\partial x^2}$$

- **Central difference** approximation

$$\frac{\mathrm{d}^2 \phi_{(i,j,k)}}{\mathrm{d}x^2} = \frac{\phi_{(i-1,j,k)} - 2\phi_{(i,j,k)} + \phi_{(i+1,j,k)}}{\Delta x^2}$$

- Three interpretations
  - $$\frac{\mathrm{d}^2 \phi_{(i,j,k)}}{\mathrm{d}x^2} = \frac{\mathrm{d}\frac{\mathrm{d}\phi_{(i,j,k)}}{\mathrm{d}x}}{\mathrm{d}x} \approx \frac{\phi'_{(i+1/2,j,k)} - \phi'_{(i-1/2,j,k)}}{\Delta x}$$

    $$\approx \frac{(\phi_{(i-1,j,k)} - \phi_{(i,j,k)})/\Delta x - (\phi_{(i,j,k)} - \phi_{(i+1,j,k)})/\Delta x}{\Delta x} = \frac{\phi_{(i-1,j,k)} - 2\phi_{(i,j,k)} + \phi_{(i+1,j,k)}}{\Delta x^2}$$

  - $$f(x+d) = f(x) + \frac{1}{1!}f'(x)d + \frac{1}{2!}f''(x)d^2 + \frac{1}{3!}f'''(x)d^3 + \frac{1}{4!}f''''(x)d^4 \ldots$$

    $$\frac{\phi_{(i-1,j,k)} - 2\phi_{(i,j,k)} + \phi_{(i+1,j,k)}}{\Delta x^2} = f''(x) + \frac{2\Delta x^2}{4!}f''''(x)\ldots$$

$$\begin{pmatrix} f(x) & f(x) & f(x) \\ -\frac{1}{1!}f'(x)\,\Delta x & 0 & \frac{1}{1!}f'(x)\,\Delta x \\ \frac{1}{2!}f''(x)\,\Delta x^2 & 0 & \frac{1}{2!}f''(x)\,\Delta x^2 \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ f''(x) \end{pmatrix} \quad \Rightarrow \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 1/\Delta x^2 \\ -2/\Delta x^2 \\ 1/\Delta x^2 \end{pmatrix}$$

Spatial Discretization (Laplace)

$$\sigma \frac{\partial^2 \phi}{\partial x^2}$$

- **Central difference** approximation

$$\frac{\mathrm{d}^2\phi_{(i,j,k)}}{\mathrm{d}x^2} = \frac{\phi_{(i-1,j,k)} - 2\phi_{(i,j,k)} + \phi_{(i+1,j,k)}}{\Delta x^2}$$

- Three interpretations (cont.)
  - $$\frac{\oint_{\text{Voxel}} \nabla\sigma\nabla\phi \, dV}{\Delta x \cdot \Delta y \cdot \Delta z} = \frac{\oint_{\text{Voxel}} (\sigma\nabla\phi) \cdot \vec{n} \, dS}{\Delta x \cdot \Delta y \cdot \Delta z}$$

$$\approx \frac{A_x(\sigma\nabla\phi)_{x,+} - A_x(\sigma\nabla\phi)_{x,-}}{\Delta x \cdot \Delta y \cdot \Delta z} = \sigma \frac{(\nabla\phi)_{x,+} - (\nabla\phi)_{x,-}}{\Delta x}$$

$$\approx \sigma \frac{\phi_{x-1} - 2\phi_x + \phi_{x+1}}{\Delta x^2}$$

  - Fluxes! (energy conservation)

Spatial Discretization (Laplace)

$$\frac{\oint_{\text{Voxel}} \nabla\sigma\nabla\phi \, dV}{\Delta x \cdot \Delta y \cdot \Delta z} = \frac{\oint_{\text{Voxel}} (\sigma\nabla\phi)\cdot\vec{n} \, dS}{\Delta x \cdot \Delta y \cdot \Delta z}$$

$$\approx \frac{A_x(\sigma\nabla\phi)_{x,+} - A_x(\sigma\nabla\phi)_{x,-}}{\Delta x \cdot \Delta y \cdot \Delta z} = \sigma\frac{(\nabla\phi)_{x,+} - (\nabla\phi)_{x,-}}{\Delta x}$$

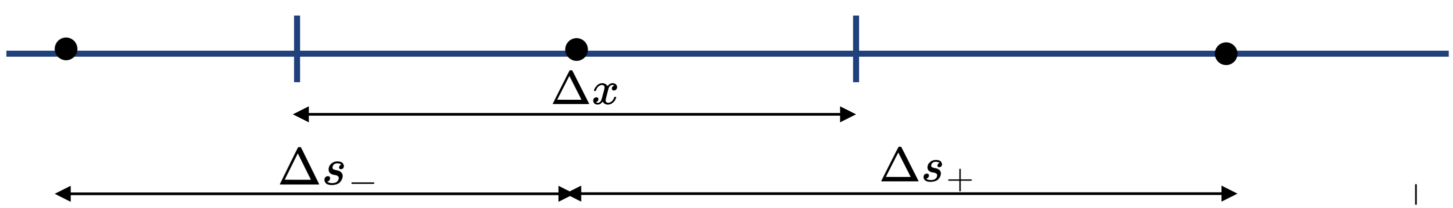
$$\approx \sigma\frac{\phi_{x-1} - 2\phi_x + \phi_{x+1}}{\Delta x^2}$$

- **Adaptive** grid steps

$$\frac{\oint_{\text{Voxel}} \nabla\sigma\nabla\phi \, dV}{\Delta x \cdot \Delta y \cdot \Delta z} = \frac{\oint_{\text{Voxel}} (\sigma\nabla\phi)\cdot\vec{n} \, dS}{\Delta x \cdot \Delta y \cdot \Delta z}$$

$$\approx \frac{A_x(\sigma\nabla\phi)_{x,+} - A_x(\sigma\nabla\phi)_{x,-}}{\Delta x \cdot \Delta y \cdot \Delta z} = \sigma\frac{(\nabla\phi)_{x,+} - (\nabla\phi)_{x,-}}{\Delta x}$$

$$\approx \sigma\frac{(\phi_{x+1} - \phi_x)/\Delta s_+ - (\phi_x - \phi_{x-1})/\Delta s_-}{\Delta x}$$



- Instead of:

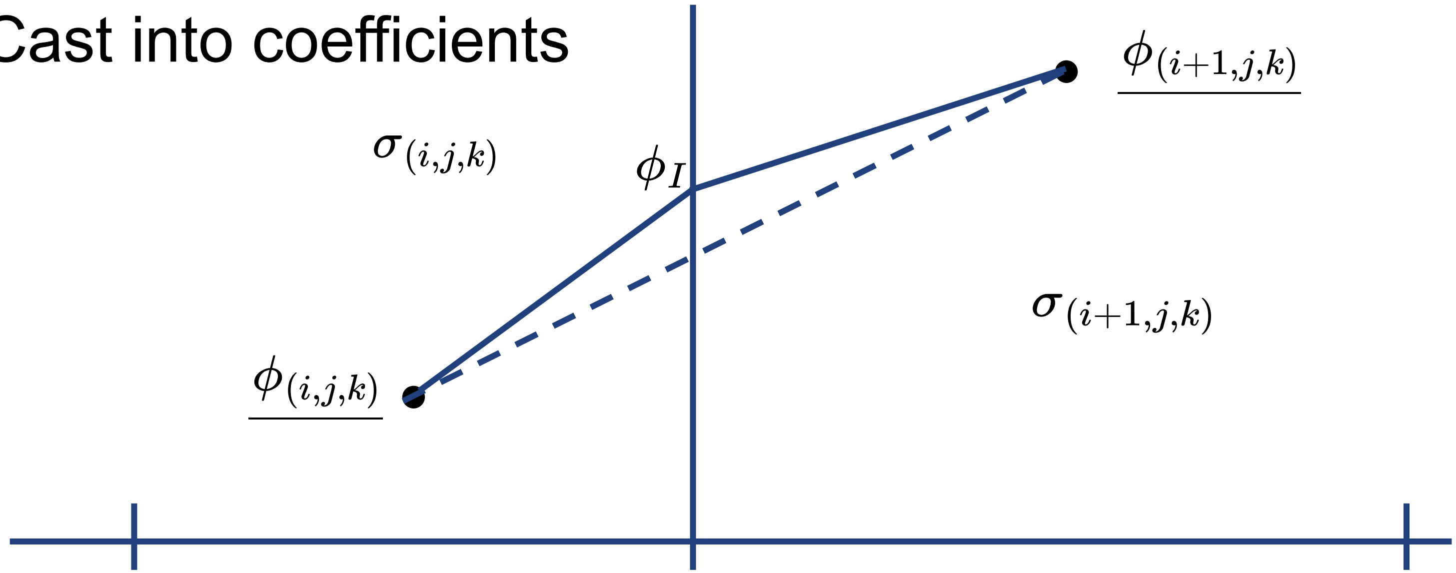
$$f = \sigma \nabla \phi_+ \approx (\phi_{x+1} - \phi_x)/\Delta s_+$$

- Flux conservation:

$$f \approx \sigma_{(i,j,k)} \frac{\phi_I - \phi_{(i,j,k)}}{\Delta x_i/2} = \sigma_{(i+1,j,k)} \frac{\phi_{(i+1,j,k)} - \phi_I}{\Delta x_{i+1}/2}$$

$$\Longrightarrow \phi_I \Longrightarrow f$$

- Cast into coefficients



Boundary Condition

- Example: convective BC

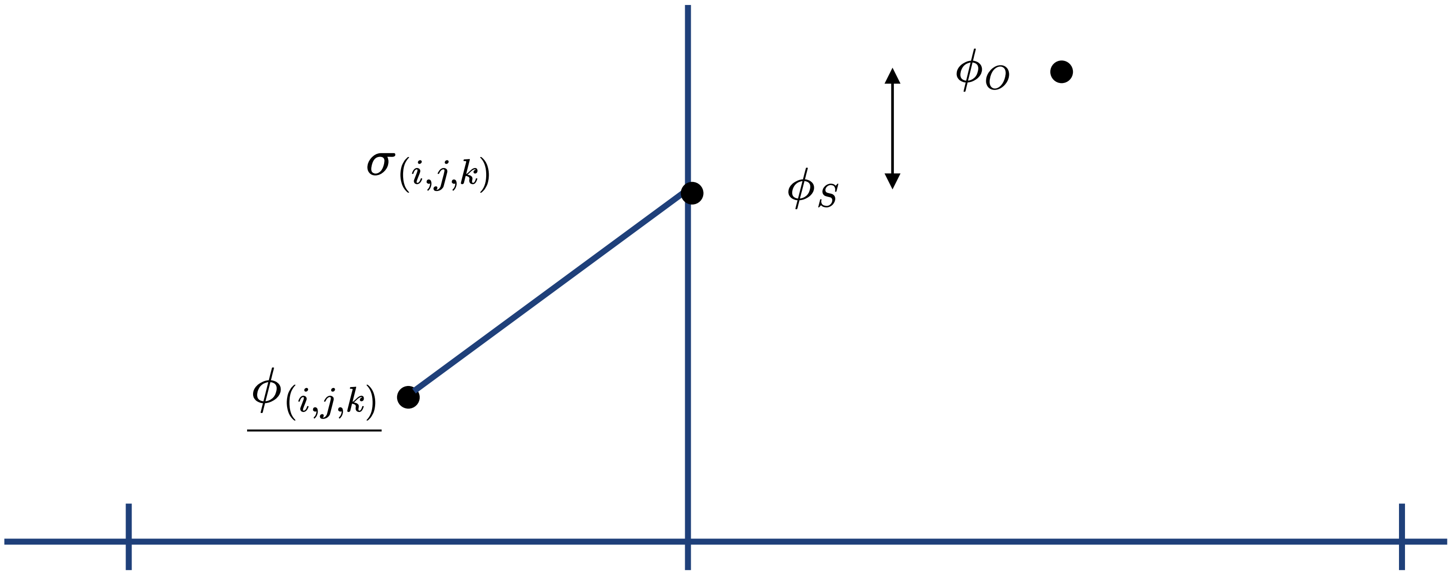
$$f = \sigma \frac{\partial \phi_S}{\partial n} = h(\phi_S - \phi_O)$$

- Flux conservation:

$$f = \sigma \frac{\phi_S - \phi}{\Delta x / 2} = h(\phi_S - \phi_O)$$

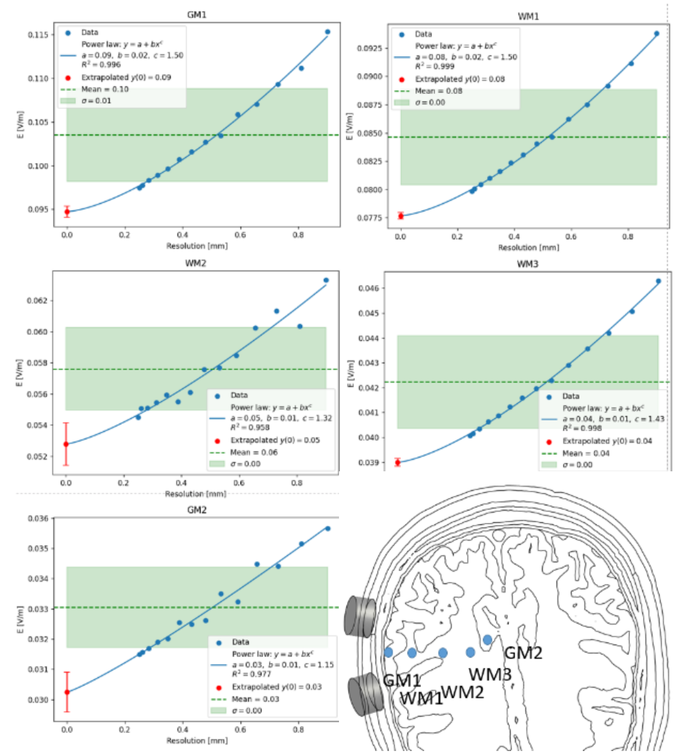
$$\Longrightarrow \phi_S \Longrightarrow f$$

- Cast into coefficients



Grid Convergence

- finite discretization introduces errors due to inability to
  - fully capture geometry
  - resolve rapidly varying fields
- geometry: typically at least two cells required
  - workarounds for special cases (e.g., virtual thin layers)
- at sufficient refinement, error typically follows power-law with exponent reflecting the order of the numerical method
- **for every serious study, perform a proper convergence analysis**
  - demonstrate that error is sufficiently small
  - part of uncertainty assessment
  - can extrapolate to continuums solution (power-law fitting)



Outline for today

- Lecture Overview
- Maxwell's Equations & Quasistatic Approx.
- Discretization
- **Finite Element Method**
- Solver Numerics
- Finally: Simulating tES
- Summary of Today's Lecture & Outlook

Finite Element Method (Galerkin)

Why reinvent the wheel? / Lazy bastard… - you decide

- https://en.wikipedia.org/wiki/Finite_element_method

Recommended for more details and practical examples

- http://jupiter.ethz.ch/~gfdteaching/femblockcourse/2015/lectures/fem-class-ETHZ2015.pdf (Chapter 2)
- https://en.wikiversity.org/wiki/Introduction_to_finite_elements/Weak_form_of_heat_equation

P.S. Advice when implementing numerical methods:

- **Avoid as a general rule to reinvent the wheel** – others have invested much more time than you in creating excellent solutions/libraries (linear algebra, partitioning, solvers…) -> be a lazy bastard

Finite Element Method (Galerkin)

Heat equation: $\rho\, C_v \dfrac{\partial T}{\partial t} - \boldsymbol{\nabla} \bullet [\boldsymbol{\kappa} \bullet \boldsymbol{\nabla} \boldsymbol{T}] = s$

Weak form: $\displaystyle\int_\Omega \left(\rho\, C_v \frac{\partial T}{\partial t}\right) w\, dV - \int_\Omega [\boldsymbol{\nabla} \bullet (\boldsymbol{\kappa} \bullet \boldsymbol{\nabla} \boldsymbol{T})]\; w\, dV = \int_\Omega s\, w\, dV$

A bit of calculus:

Variational BVP for the Heat Equation

Find a function $T(t) \in \mathcal{S}_t,\, t \in [0, \tau]$ such that for all $w \in \mathcal{V}$

$$\int_\Omega \left(\rho\, C_v \frac{\partial T}{\partial t}\right) w\, dV + \int_\Omega (\boldsymbol{\nabla} w) \bullet (\boldsymbol{\kappa} \bullet \boldsymbol{\nabla} T)\, dV = \int_\Omega s\, w\, dV - \int_{\Gamma_q} w\, \bar{q}\, dA$$

$$\int_\Omega w\, \rho\, C_v\, T(0)\, dV = \int_\Omega w\, \rho\, C_v\, T_0\, dV\,.$$

Finite Element Method (Galerkin)

Variational BVP for the Heat Equation

Find a function $T(t) \in \mathcal{S}_t, t \in [0, \tau]$ such that for all $w \in \mathcal{V}$

$$\int_\Omega \left(\rho\, C_v\, \frac{\partial T}{\partial t}\right) w\, dV + \int_\Omega (\nabla w) \bullet (\boldsymbol{\kappa} \bullet \nabla T)\, dV = \int_\Omega s\, w\, dV - \int_{\Gamma_q} w\, \bar{q}\, dA$$

$$\int_\Omega w\, \rho\, C_v\, T(0)\, dV = \int_\Omega w\, \rho\, C_v\, T_0\, dV\ .$$

Trial solution: $$\int_\Omega \left(\rho\, C_v\, \frac{\partial T_h}{\partial t}\right) w_h\, dV + \int_\Omega (\nabla w_h) \bullet (\boldsymbol{\kappa} \bullet \nabla T_h)\, dV = \int_\Omega s\, w_h\, dV - \int_{\Gamma_q} w_h\, \bar{q}\, dA$$

Base function decomposition:
(Galerkin)

$$w_h(\mathbf{x}) = \sum_{i=1}^{n} b_i N_i(\mathbf{x}) \qquad T_h = v_h + \bar{\bar{T}}_h$$

$$v_h(\mathbf{x}, t) = \sum_{i \in \eta - \eta^T} T_i(t) N_i(\mathbf{x})$$

$$\sum_j b_j \left[\sum_i \frac{\partial T_i}{\partial t} \left(\int_\Omega \rho\, C_v\, N_i\, N_j\, dV\right) + \sum_i T_i \left(\int_\Omega \nabla N_j \bullet (\boldsymbol{\kappa} \bullet \nabla N_i)\, dV\right)\right] =$$
$$\sum_j b_j \left[\int_\Omega N_j\, s\, dV - \int_{\Gamma_q} N_j\, \bar{q}\, dA - \sum_k \frac{\partial \bar{T}_k}{\partial t} \left(\int_\Omega \rho\, C_v\, N_j\, N_k\, dV\right)\right.$$
$$\left. - \sum_k \bar{T}_k \left(\int_\Omega \nabla N_j \bullet (\boldsymbol{\kappa} \bullet \nabla N_k)\, dV\right)\right]$$

Finite Element Method (Galerkin)

$$\sum_j b_j \left[\sum_i \frac{\partial T_i}{\partial t} \left(\int_\Omega \rho\, C_v\, N_i\, N_j\, dV \right) + \sum_i T_i \left(\int_\Omega \nabla N_j \bullet (\boldsymbol{\kappa} \bullet \nabla N_i)\, dV \right) \right] =$$

$$\sum_j b_j \left[\int_\Omega N_j\, s\, dV - \int_{\Gamma_q} N_j\, \overline{q}\, dA - \sum_k \frac{\partial \overline{T}_k}{\partial t} \left(\int_\Omega \rho\, C_v\, N_j\, N_k\, dV \right) - \sum_k \overline{T}_k \left(\int_\Omega \nabla N_j \bullet (\boldsymbol{\kappa} \bullet \nabla N_k)\, dV \right) \right]$$

Must be satisfied for all weighting functions *w*… therefore:

$$\sum_i \frac{\partial T_i}{\partial t} \boxed{\left(\int_\Omega \rho\, C_v\, N_i\, N_j\, dV \right)} + \sum_i T_i \boxed{\left(\int_\Omega \nabla N_j \bullet (\boldsymbol{\kappa} \bullet \nabla N_i)\, dV \right)} =$$

$$\boxed{\int_\Omega N_j\, s\, dV - \int_{\Gamma_q} N_j\, \overline{q}\, dA - \sum_k \frac{\partial \overline{T}_k}{\partial t} \left(\int_\Omega \rho\, C_v\, N_j\, N_k\, dV \right) - \sum_k \overline{T}_k \left(\int_\Omega \nabla N_j \bullet (\boldsymbol{\kappa} \bullet \nabla N_k)\, dV \right)}$$

And in matrix form: $\sum_i \boxed{M_{ji}}\, \dot{T}_i + \sum_i \boxed{K_{ji}}\, T_i = \boxed{f_j}$ $\quad \boxed{\mathbf{M}\, \dot{\mathbf{T}} + \mathbf{K}\, \mathbf{T} = \mathbf{f}}$

‘mass matrix’ ‘stiffness matrix’

Outline for today

- Lecture Overview
- Maxwell's Equations & Quasistatic Approx.
- Discretization
- Finite Element Method
- **Solver Numerics**
- Finally: Simulating tES
- Summary of Today's Lecture & Outlook

Rectilinear EQS Solver

- Hepta-diagonal system, diagonal dominant
  - Many efficient sparse solving scheme exist (we used *bicgstab(l)* in the past)

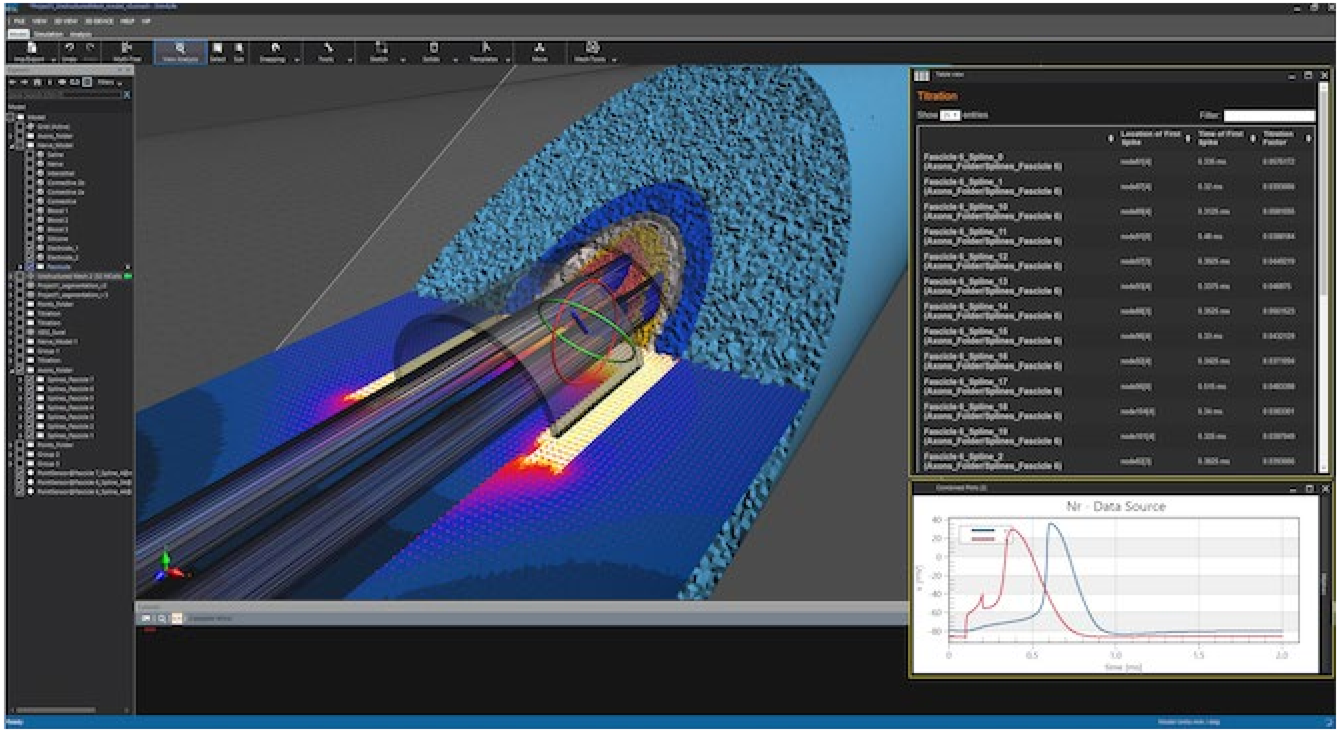
- Difficulty of generating high quality meshes of complex anatomical geometries
- Stability question is replaced by conditioning issues (e.g., highly conductive metal)

- Application of a fixed stencil, a large number of times
- (corresponds to a highly structured sparsity pattern of the update matrix)

➢ Broad range of suitable parallelization approaches

(EQS) FEM

- Requires iterative solving of large linear systems
- Typically less structured sparsity-pattern due to prevalence of unstructured meshing (conformal, adaptive; e.g., tetrahedral meshes)
  - Frequently block structure of the sparsity pattern

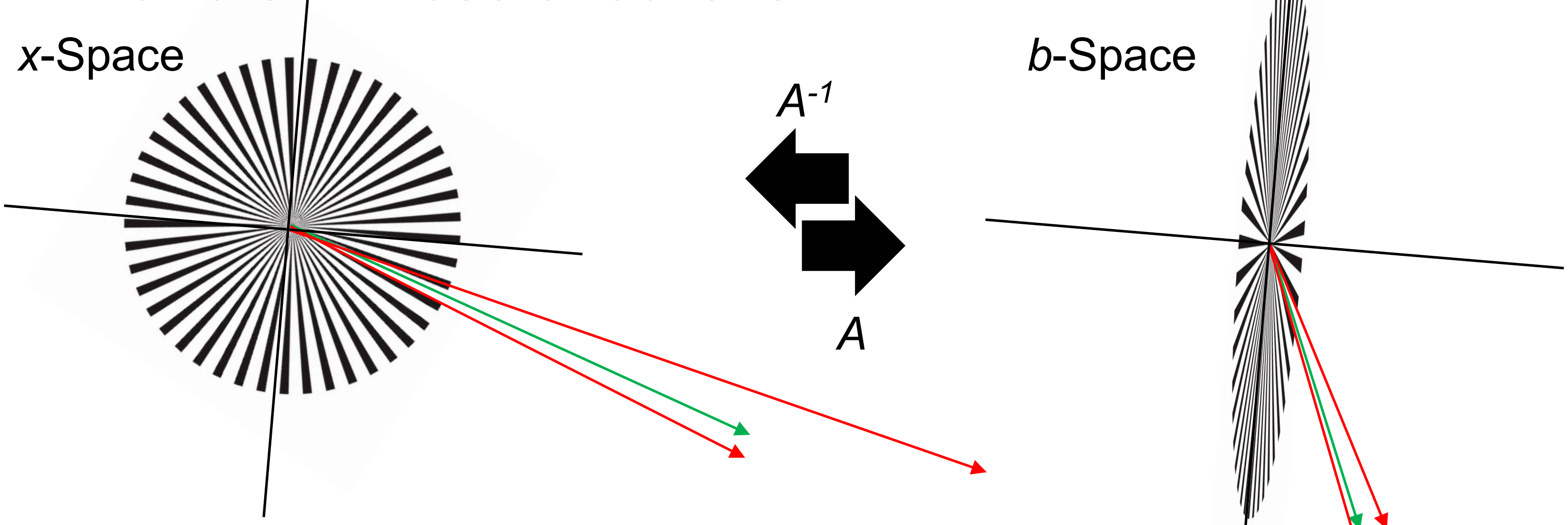


Solving is not Inverting

- Matrix inversion: compute explicitly A^{-1}, "direct method"
  - Complexity n^3, unaffordable
  - Applying the matrix is a mat/vec multiplication, $O(n^2)$
  - The inverse of a sparse matrix can be dense!
  - Hard to parallelize
- Matrix factorization: compute A=LU (lower/upper triangular) so that we solve Ly=b, Ux=y, "direct" method
  - Complexity n^2 much better than explicit computation, still usually unaffordable
  - Applying the inverse => solving 2 triangular systems, $O(n^2)$
  - LU factors of a sparse matrix can be dense!
  - Hard to parallelize
- Iterative methods (only needs action of matrix on vector)
  - Computing $A^{-1}b$ by solving the system Ax=b through an iterative method up to a defined precision
  - Complexity: depends on sparsity, cond. number, tolerance,… Sparser => better
  - Applying the inverse => solving one linear system
  - Easily parallelizable
  - We don't even need A explicitly (matrix-free methods), only its action on a vector

Condition Number

- Condition number (oversimplified): $\kappa(A) = \frac{\max\{|\lambda(A)|\}}{\min\{|\lambda(A)|\}}$
- High condition numbers complicate matrix solving
- Interpretation: certain solution space directions contribute little to the product (*b*) compared to others; hence, the residuum is less sensitive to solution (*x*) errors in those directions



- Problematic example in tES: presence of highly conductive implants

Preconditioning

- Left and/or right preconditioning:

$$({P_L}^{-1}\ J_{alpha}\ {P_R}^{-1})\ (P_R \delta T) = -{P_L}^{-1}\ R(t,T)$$

- Preconditioners should:
  - be easy to build and invert
  - should reduce the spectrum of the system (condition number)

- General preconditioners: rely only on the sparsity pattern of the Jacobian matrix
- Specialized preconditioners: consider the physics of the system

Libraries and Packages

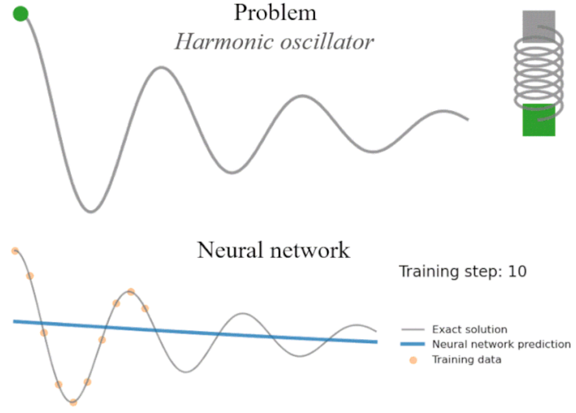
- Tons of powerful algorithms:
  - solvers: Richardson, Chebyshev, Conjugate Gradient (CG), BiCG, BiCGStab, GMRES, Flexible GMRES, Direct Methods (MUMPS, SuperLU)...
  - preconditioners: Jacobi (Diagonal), Algebraic Schwarz, Multigrid, Incomplete LU...
  - discretization, FEM system formulation...
  - with HPC support
- First consider using existing packages, such as:
  - PETSc, Trilinos, Deal II...
  - cross-system, cross-hardware...
  - there are even AIs that help pick libraries and algorithms and hyperparameters

Approaches to Parallelization

- Domain decomposition:
  - The simulation domain is partitioned into overlapping subdomains
  - Redundant computation in the overlapping region
  - Iterative process: the subdomains are coupled with the neighbors
  - Optimal data locality: usually only requires the communication of the overlapped regions
  - Load balance: usually difficult, depends on the partitioning solution (optimal partitioning is NP-complete) and on time-dependent factors
  - Usually preferred for distributed computation
- Task/Event-based parallelism
  - The processing units consume the next task in the queue
  - Best load balance: the processing units are never idle
  - Data locality is problematic
  - Requires a runtime scheduler which is not trivial and has to take into account data locality
  - Usually preferred for shared-memory computations
- Parallelization is not restricted to solving – it must cover loading, partitioning, coefficient computation…

naïve AI solution

- compute discrete solution using FEM
- apply supervised learning to fit a NN u depending on $(\vec{x}, t)$ to precomputed data points $\{(\vec{x}_i, t_i), u_i\}_{i \in I}$
- good interpolation but very bad outside of obtained points, because there is no knowledge about underlying physics

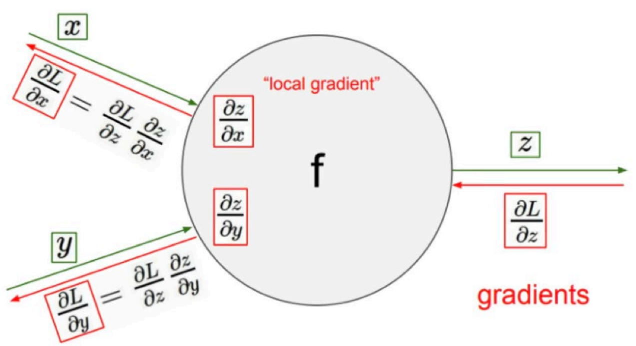


https://benmoseley.blog/my-research/so-what-is-a-physics-informed-neural-network/

observation

- AI libraries use autodiff to compute gradients during training
- no discretization required
- autodiff to compute PDE of system as loss function (Raissi 2017)

➤ Physics-Informed Neural Network (PINN)



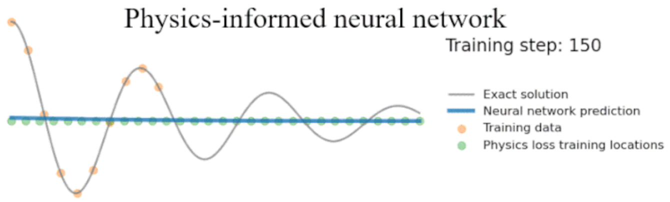
https://pantelis.github.io/cs301/docs/common/lectures/dnn/backprop-intro/

loss as weighted sum of:

- boundary conditions (Dirichlet, Neumann)
- supervised points (simulation or measurements)
- initial conditions
- PDE loss via autodiff

outcomes

- find weights of NN that minimize above loss
- solution is mesh-free and inference times are small
- quality can be evaluated in terms of PDE loss
- draw back: needs to be fitted for every new configuration of initial and boundary conditions



https://benmoseley.blog/my-research/so-what-is-a-physics-informed-neural-network/

https://www.youtube.com/watch?v=zbVBfNKDGJY

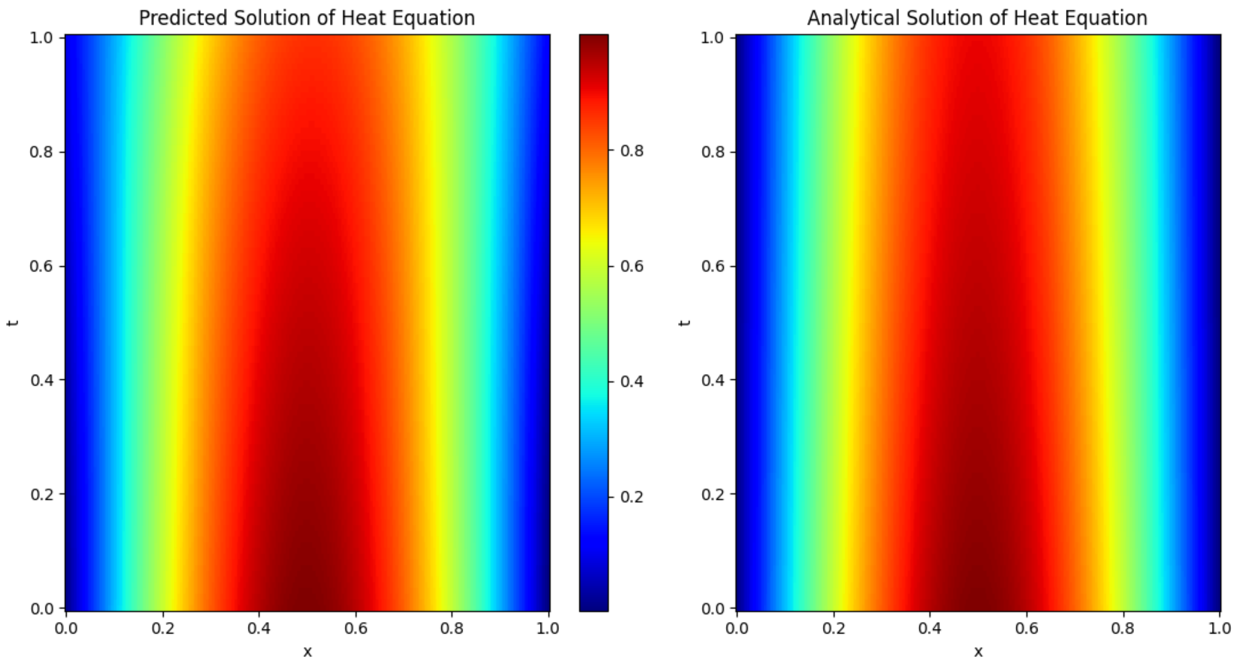
Physics-Informed Neural Networks

PINNs are observed to be empirically very successful for

- high-dimensional PDEs
- parametric PDEs
- inverse problems

Caveats:

- PINNs may not work for problems with strong gradients
- training error is a blackbox



Heat Equation: $\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$

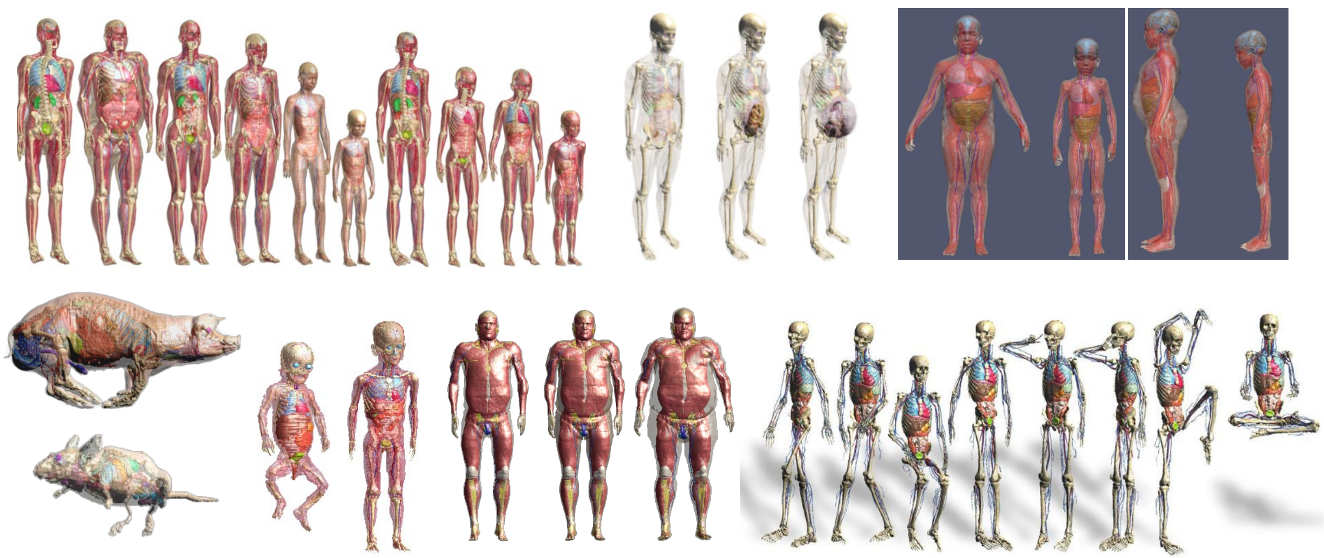
Outline for today

- Lecture Overview
- Maxwell's Equations & Quasistatic Approx.
- Discretization
- Finite Element Method
- Solver Numerics
- **Finally: Simulating tES**
- Summary of Today's Lecture & Outlook

ETH zürich | IT'IS FOUNDATION

Anatomical Models

- preexisting reference models, such as the Virtual Population, MIDA...
- image-based, personalized modeling: forthcoming lecture
- typical tissues: scalp, bone, cerebro-spinal fluid (CSF), white- and grey-matter, internal air, eyes
- often neglected, but important: cortical vs. cancellous bone, fat, muscle, galea, dura

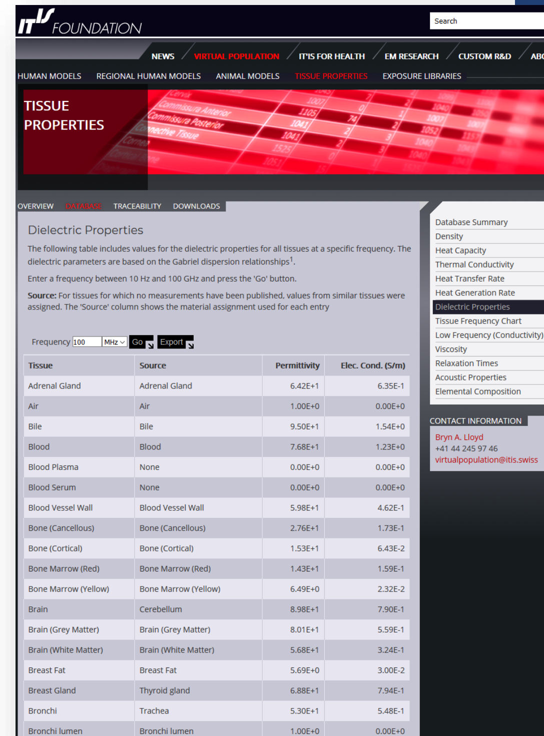


Tissue Properties

ETH zürich

IT'IS FOUNDATION

- high variability and uncertainty
  - curated reference database with QA, regular updates, and variability information: https://itis.swiss/virtual-population/tissue-properties/database/
- especially white-matter is heterogeneous and anisotropic
  - DTI-based assignment possible



IT'IS FOUNDATION

Search

NEWS / VIRTUAL POPULATION / IT'IS FOR HEALTH / EM RESEARCH / CUSTOM R&D / ABOUT

HUMAN MODELS | REGIONAL HUMAN MODELS | ANIMAL MODELS | TISSUE PROPERTIES | EXPOSURE LIBRARIES

TISSUE PROPERTIES

OVERVIEW | DATABASE | TRACEABILITY | DOWNLOADS

Dielectric Properties

The following table includes values for the dielectric properties for all tissues at a specific frequency. The dielectric parameters are based on the Gabriel dispersion relationships[1].

Enter a frequency between 10 Hz and 100 GHz and press the 'Go' button.

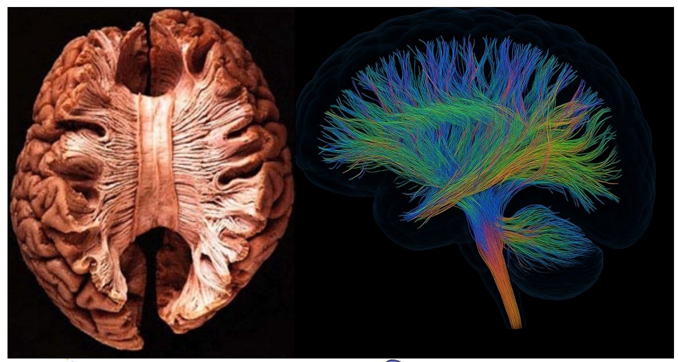
Source: For tissues for which no measurements have been published, values from similar tissues were assigned. The 'Source' column shows the material assignment used for each entry

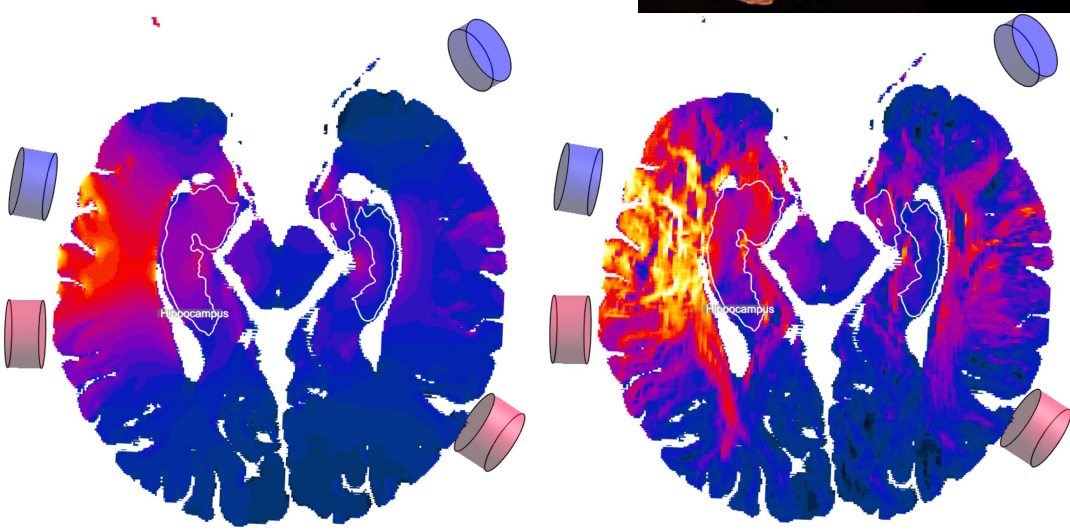
Frequency 100 MHz Go Export

Tissue	Source	Permittivity	Elec. Cond. (S/m)
Adrenal Gland	Adrenal Gland	6.42E+1	6.35E-1
Air	Air	1.00E+0	0.00E+0
Bile	Bile	9.50E+1	1.54E+0
Blood	Blood	7.68E+1	1.23E+0
Blood Plasma	None	0.00E+0	0.00E+0
Blood Serum	None	0.00E+0	0.00E+0
Blood Vessel Wall	Blood Vessel Wall	5.98E+1	4.62E-1
Bone (Cancellous)	Bone (Cancellous)	2.76E+1	1.73E-1
Bone (Cortical)	Bone (Cortical)	1.53E+1	6.43E-2
Bone Marrow (Red)	Bone Marrow (Red)	1.43E+1	1.59E-1
Bone Marrow (Yellow)	Bone Marrow (Yellow)	6.49E+0	2.32E-2
Brain	Cerebellum	8.98E+1	7.90E-1
Brain (Grey Matter)	Brain (Grey Matter)	8.01E+1	5.59E-1
Brain (White Matter)	Brain (White Matter)	5.68E+1	3.24E-1
Breast Fat	Breast Fat	5.69E+0	3.00E-2
Breast Gland	Thyroid gland	6.88E+1	7.94E-1
Bronchi	Trachea	5.30E+1	5.48E-1
Bronchi lumen	Bronchi lumen	1.00E+0	0.00E+0

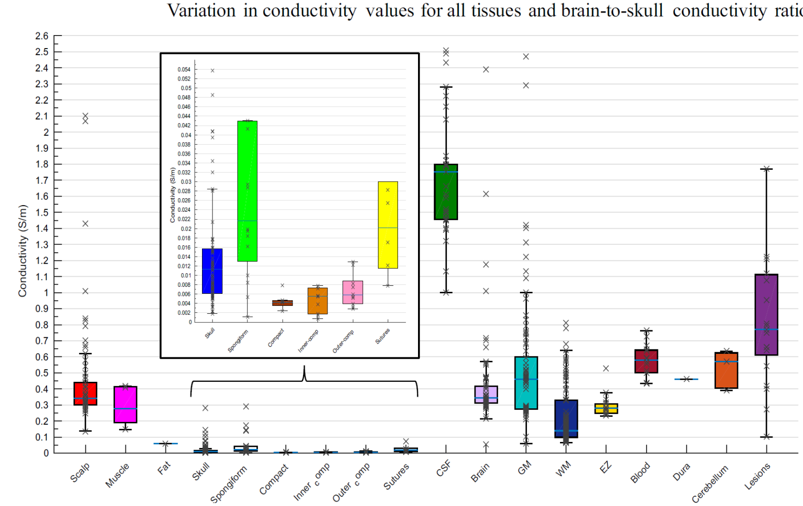
Database Summary
Density
Heat Capacity
Thermal Conductivity
Heat Transfer Rate
Heat Generation Rate
Dielectric Properties
Tissue Frequency Chart
Low Frequency (Conductivity)
Viscosity
Relaxation Times
Acoustic Properties
Elemental Composition

CONTACT INFORMATION
Bryn A. Lloyd
+41 44 245 97 46
virtualpopulation@itis.swiss





Variation in conductivity values for all tissues and brain-to-skull conductivity ratio



Boundary Conditions

- tES is typically current controlled
- often, current density BCs are used
  - but: neglects field enhancement at edges, proximity effects...
- preferable: Dirichlet (fixed voltage) BCs, followed by current normalization
  - when >2 electrodes: necessitates construction and inversion of conductance matrix (math in later lecture on stimulation optimization)

- electrode-tissue Interface
  - frequency-dependent impedance because of: imperfect contact, stratum corneum capacitance...
  - affects wave-form; multiplication in frequency space (convolution in time) required

- Lecture Overview
- Maxwell's Equations & Quasistatic Approx.
- Discretization
- Finite Element Method
- Solver Numerics
- Finally: Simulating tES
- **Summary of Today's Lecture & Outlook**

Course objectives

At the end of this lecture, you will have

- an understanding for the quasistatic approximations of Maxwells equations of relevance to neurostimulation
- insights into the advantages and disadvantages of structured and unstructured discretization
- learned about the finite element method and numerical aspects of solving the resulting linear systems

Next week: Hybrid EM-neuro modeling of nerve interfaces for bioelectronic medicine

- The exercise will revolve around practical FEM modeling of head exposure

Course Outline

ETH zürich | IT'IS FOUNDATION

DATE	LECTURE THEME
19.02	Motivation, logistics & tooling (EN, TNE)
26.02	Ion channels & membranes (EN)
05.03	Axon models, activating functions & electrical stimulation (EN)
12.03	EM field simulation fundamentals & coupled EM-neuro workflows (EN)
19.03	Peripheral nerves & interfaces for bioelectronic medicine (EN)
26.03	Spinal cord stimulation for neuroprosthetics and pain management & low-frequency exposure safety (TNE)
02.04	Morphology, synapses, microcircuits; point vs spiking networks (TNE)
09.04	No class: Easter break
16.04	Neural mass & whole brain models; hybridization (TNE)
23.04	Recording modalities, signal content & the reciprocity theorem (TNE)
30.04	Non invasive brain stimulation & temporal interference (TNE)
07.05	Image based/personalized treatment planning and optimization (EN)
14.05	No class: Ascension Day
21.05	Verification, validation, UQ, and reproducibility (EN)
28.05	Project presentations & synthesis (EN, TNE)

Room: ETZ E7
13:15-14:00 Lecture
14:00-14:15 Break
14:15-15:00 Lecture
14:00-14:15 Break
15:15-16:00 Exercise

Lecture Recordings & Slides

Provided Here
(will successively appear)

Exercise Outline

DATE	EXERCISE THEME
19.02	"Hello Neuron": integrate-and-fire in Python/NEURON
26.02	Point neuron phase portrait; basic time integration numerics
05.03	Recruitment prediction for myelinated axon using AF/GAF
12.03	EM (FEM) modeling of transcranial brain stimulation
19.03	Stimulation selectivity and signal content modeling for nerve interfaces
26.03	Guest (SCS – NeuroRestore)
02.04	Mini project work
09.04	No class: Easter break
16.04	Guest (Neuromodulation Spin-Off – Z43)
23.04	Mini project work
30.04	Guest (NIBS – Kinderspital)
07.05	Mini project work
14.05	No class: Ascension Day
21.05	Mini project work
28.05	Project presentations

Room: ETZ E7

13:15-14:00 Lecture

14:00-14:15 Break

14:15-15:00 Lecture

14:00-14:15 Break

15:15-16:00 Exercise