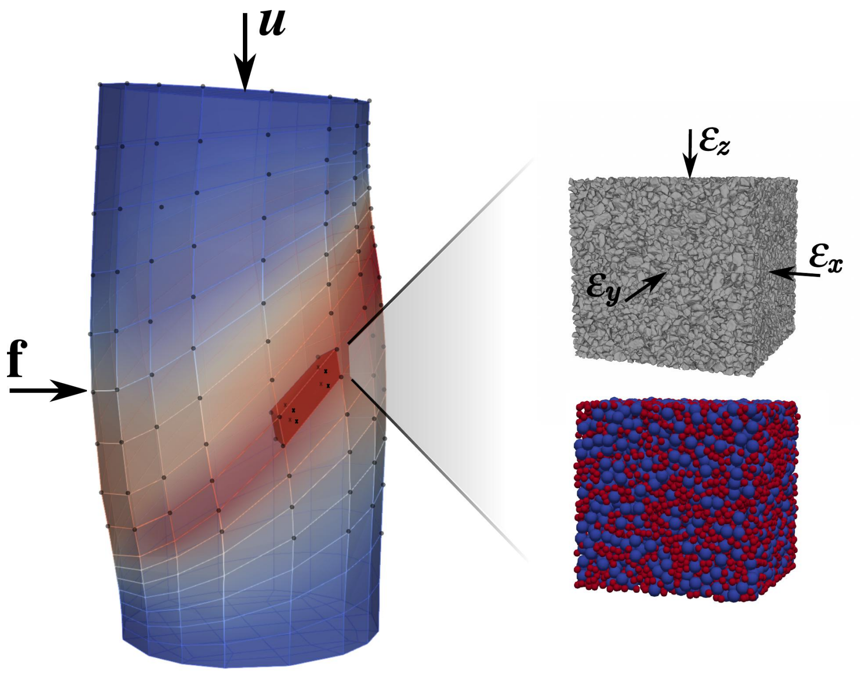
EPFL

K.Karapiperis



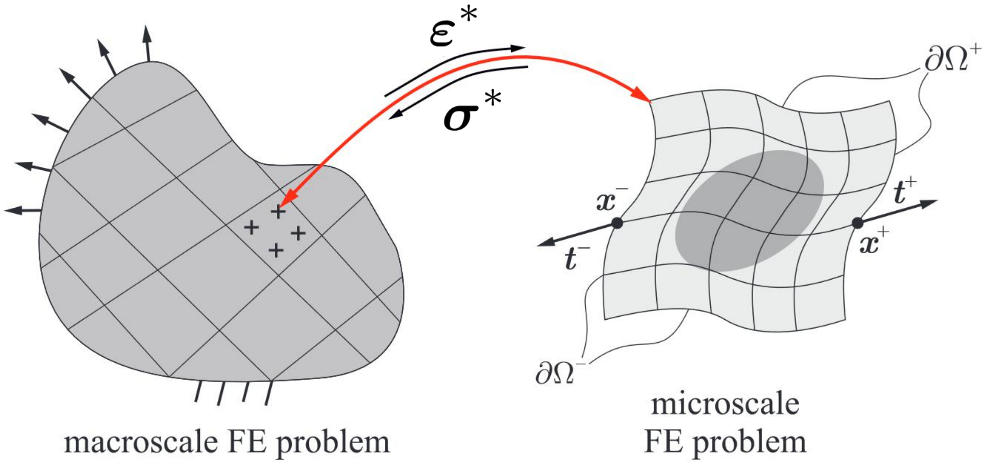
CIVIL-408 Multiscale Modeling in Mechanics

Prof. Kostas Karapiperis

Week 10

Conventional hierarchical multiscale methods

Conventional vertical scale bridging relies on repeatedly ("on-the-fly") solving a boundary value problem at the microscale.
This serves as an effective material model.



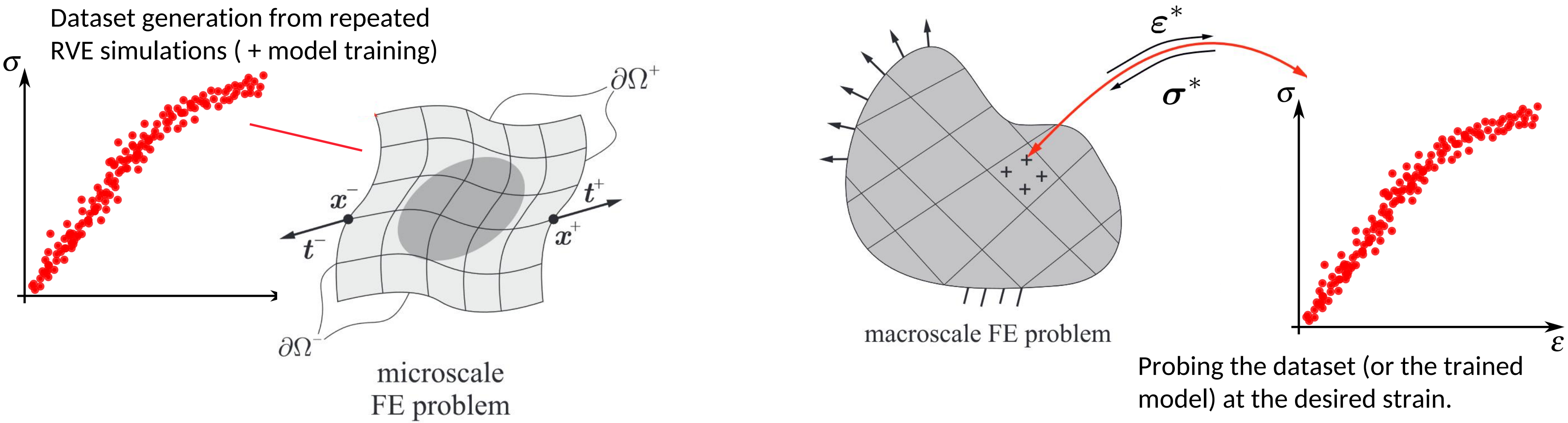
$$\langle \varepsilon \rangle = \varepsilon$$

$$\sigma = \langle \sigma \rangle$$

Though versatile, this can incur **significant computational expenses**!

A useful alternative: Data-driven methods

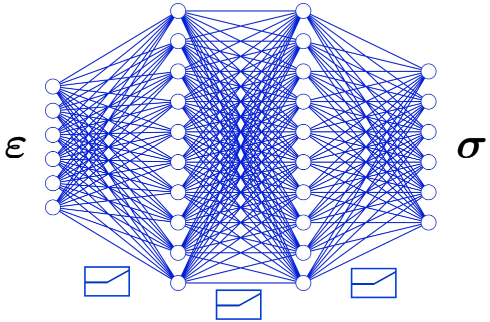
What if we introduced an **“offline” phase** where we evaluate this effective material model over a variety of paths, and then use that data inside a data-driven multiscale framework?



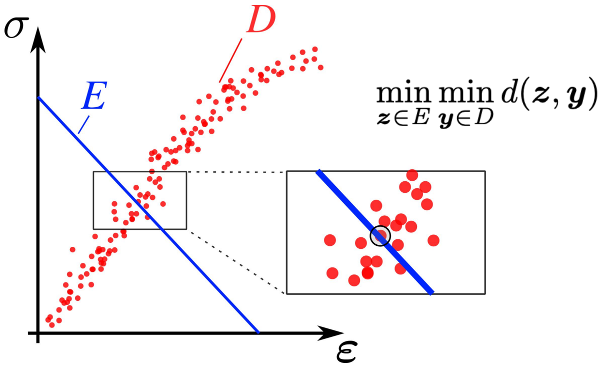
Such approaches hold potential for **significant acceleration**!

An (incomplete) taxonomy of data-driven methods

Machine learning-based models



Data-driven computational mechanics



"To learn or not to learn?"

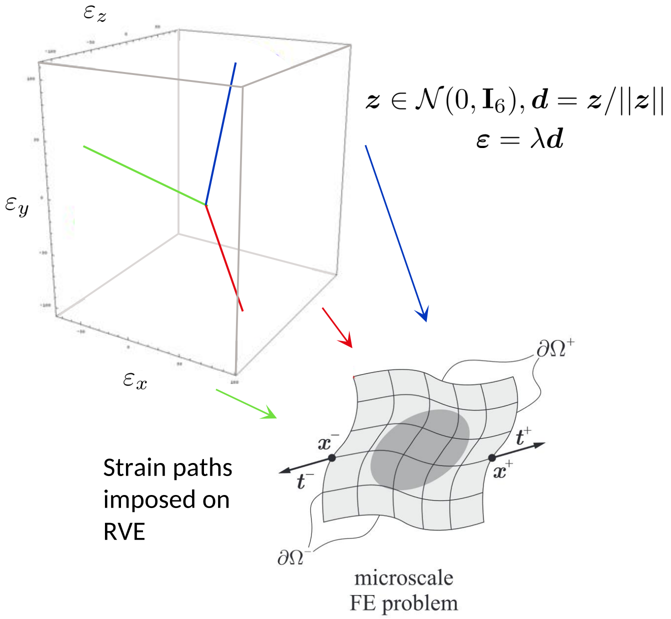
Relies on a **machine learning model trained on the stress-strain data.**

Relies only on the **raw stress-strain data.** No approximations. No loss of information.

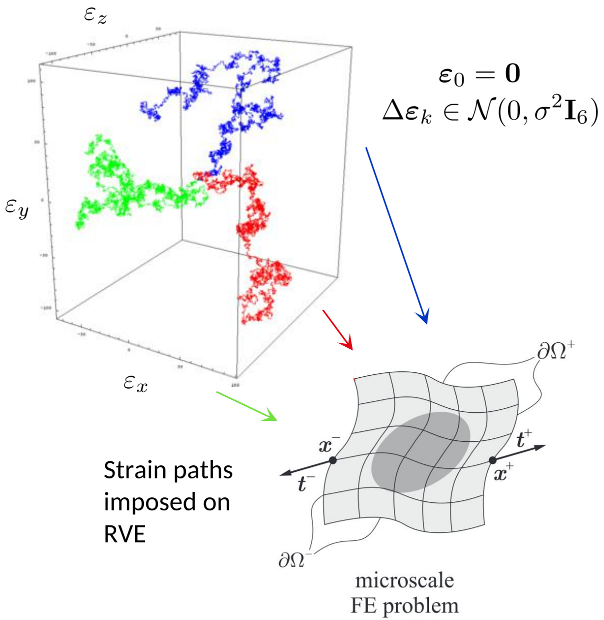
Machine learning-based multiscale modeling

NN-based surrogate constitutive laws

Dataset generation
via proportional loading



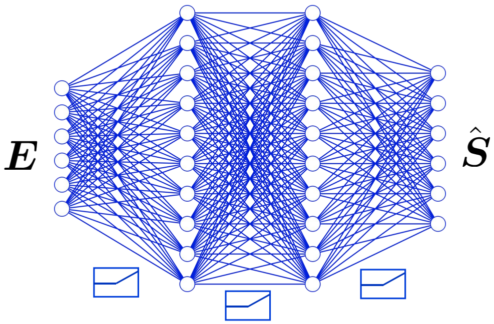
via random walks



Machine learning-based multiscale modeling

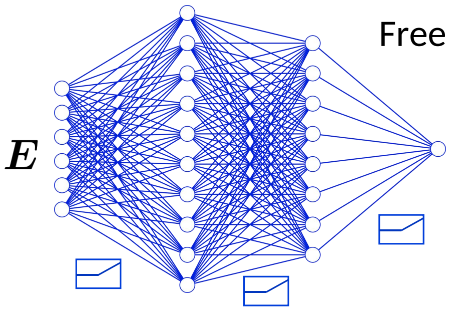
NN-based surrogate constitutive laws (elasticity)

Choice of architecture:
Simple Feed-forward Neural Network



+ Simplicity
- Does not respect thermodynamics, objectivity

Thermodynamics-informed Neural Network



+ Built-in thermodynamic consistency assuming the mapping is convex (e.g. via Input Convex Neural Networks)

Training:

$$\min_{\theta} ||\hat{\boldsymbol{S}} - \boldsymbol{S}||$$

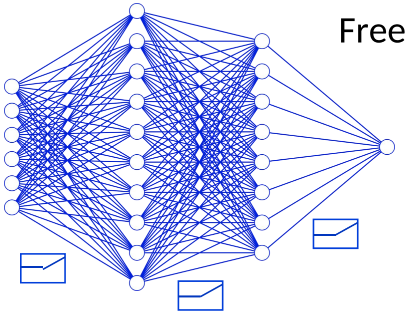
$$\min_{\theta} ||\hat{\psi} - \psi|| + w_S||\hat{\boldsymbol{S}} - \boldsymbol{S}||$$

Machine learning-based multiscale modeling

NN-based surrogate constitutive laws (elasticity)

Choice of architecture:
Invariant-based "Constitutive" Neural Network

$$\boldsymbol{E} \rightarrow I_1, I_2, I_3, \vdots$$



Free energy

$$\hat{\psi}(I_i(\boldsymbol{E})) \rightarrow \hat{\boldsymbol{S}} = \frac{\partial \hat{\psi}}{\partial \boldsymbol{E}}$$

+ Built-in objectivity/frame indifference

Note:
There are also other ways to satisfy frame indifference, e.g. via Tensor-basis Neural Networks

Training:

$$\min_{\theta} ||\hat{\psi} - \psi|| + w_S||\hat{\boldsymbol{S}} - \boldsymbol{S}||$$

Machine learning-based multiscale modeling

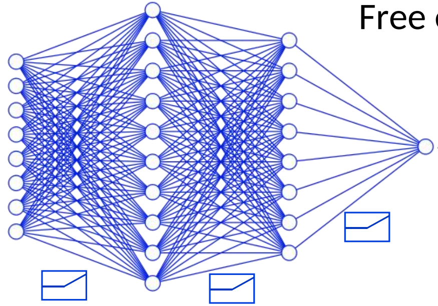
NN-based surrogate constitutive laws (inelasticity)

K.Karapiperis

Approach:
Generalized Standard Material Networks (GSMN)

$\boldsymbol{E}$

ISV $\longrightarrow$ $\boldsymbol{z}$



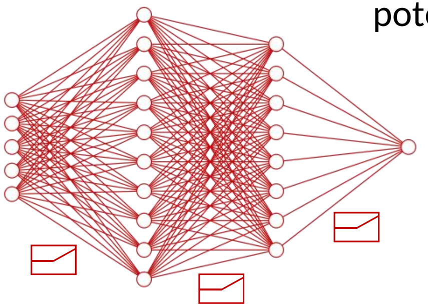
Free energy

$$\hat{\psi}(\boldsymbol{E}, \boldsymbol{z}) \longrightarrow \hat{\boldsymbol{S}} = \frac{\partial \hat{\psi}}{\partial \boldsymbol{E}}$$

Dissipation potential

$\dot{\boldsymbol{z}}$

$\boldsymbol{z}$



$$\hat{\phi}(\dot{\boldsymbol{z}}, \boldsymbol{z})$$

+ Built-in thermodynamic constistency incl. positivity of the dissipation
- Does not encompass all types of materials

Evolution of ISVs governed by Biot equation:

$$\frac{\partial \hat{\phi}}{\partial \dot{\boldsymbol{z}}} + \frac{\partial \hat{\psi}}{\partial \boldsymbol{z}} = \boldsymbol{0}$$

Training:

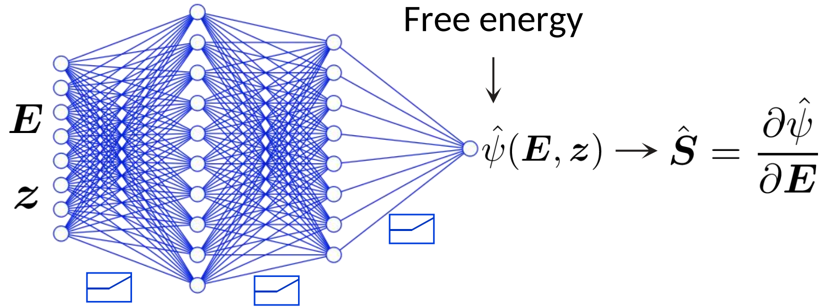
$$\min_{\theta} ||\hat{\psi} - \psi|| + w_S ||\hat{\boldsymbol{S}} - \boldsymbol{S}|| + w_{Biot} \left|\left| \frac{\partial \hat{\phi}}{\partial \dot{\boldsymbol{z}}} + \frac{\partial \hat{\psi}}{\partial \boldsymbol{z}} \right|\right|$$

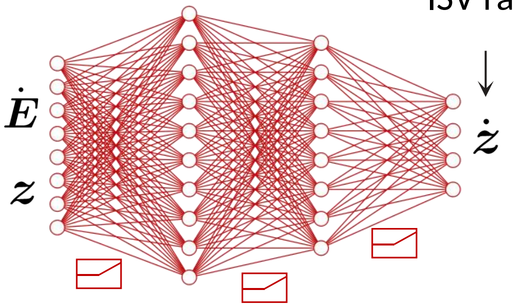
Machine learning-based multiscale modeling

NN-based surrogate constitutive laws (inelasticity)

K.Karapiperis

Approach:
Neural ODE/Recurrent Neural Networks





+ Versatile framework for wider class of materials
- Weakly-enforced positive dissipation

Alternatively, the ISV at the next time step may be predicted by an RNN/LSTM/..

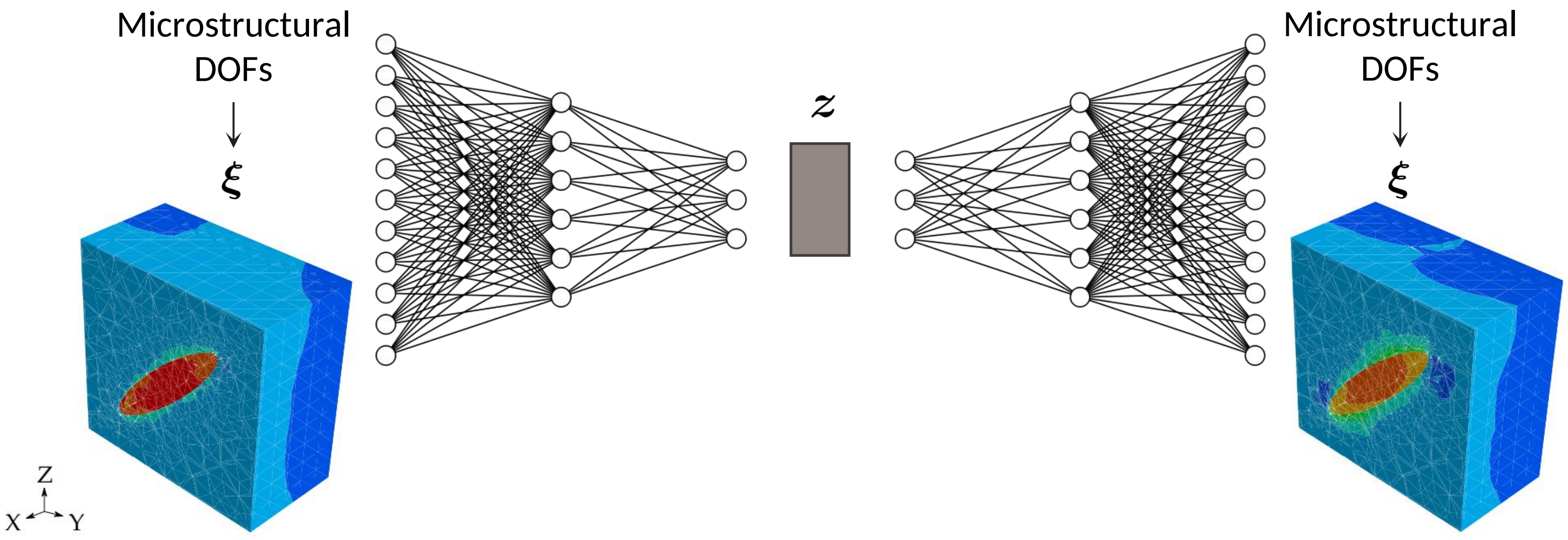
Training:

$$\min_{\theta} ||\hat{\psi} - \psi|| + w_S||\hat{\boldsymbol{S}} - \boldsymbol{S}|| + w_z||\hat{\boldsymbol{z}} - \boldsymbol{z}|| + w_D \mathrm{ReLU}\left(\frac{\partial \psi}{\partial \boldsymbol{z}} : \dot{\boldsymbol{z}}\right)$$

Machine learning-based multiscale modeling

NN-based surrogate constitutive laws (inelasticity)

Both the GSMN and the Neural ODE-based approach need to be supplemented by a model that compresses all microstructural RVE information ξ to an appropriate set of ISVs z, e.g. via an Autoencoder or a Proper Orthogonal Decomposition.

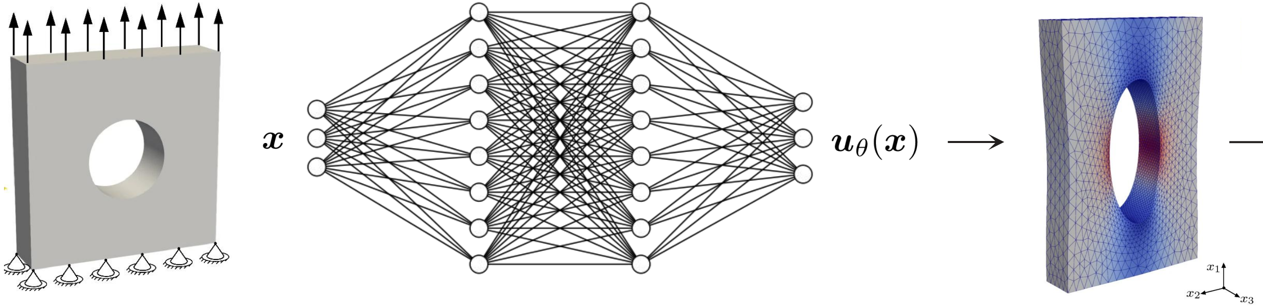


Physics-informed ML-based multiscale modeling

K.Karapiperis

Physics-informed *neural networks*

Instead of directly approximating the effective energy/stress, we can also use ML to efficiently solve the microscale boundary value problem while respecting the physics (momentum balance, etc) at that scale and the boundary conditions.



Homogenization

$\langle \boldsymbol{\sigma} \rangle, \langle \mathbb{C} \rangle$

Training:

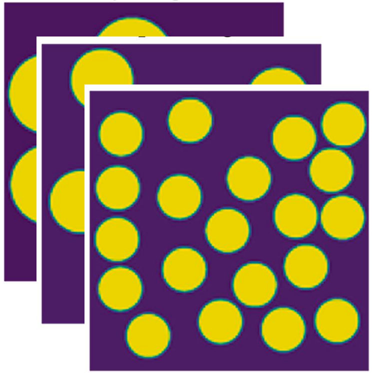
PDE residual ↓ Boundary conditions ↓

$$\min_\theta w_{\mathrm{PDE}} \mathcal{L}_{\mathrm{PDE}} + w_{\mathrm{BC}} \mathcal{L}_{\mathrm{BC}}$$

Physics-informed ML-based multiscale modeling

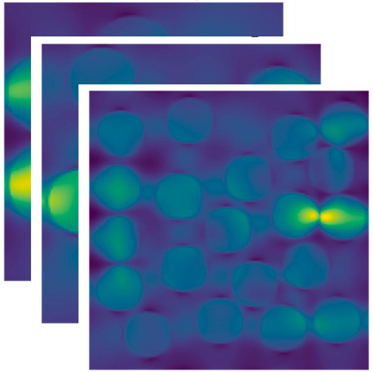
Physics-informed *neural operators*

The previous approach can only furnish the microscopic solution field for a given RVE. Neural operators instead learn the mapping between any RVE (or the same RVE with different loading conditions) and the corresponding microscopic solution field.



Material field

$$\mathcal{F} : \mathcal{A}(\Omega; \mathbb{R}^{d_M}) \to \mathcal{U}(\Omega; \mathbb{R}^{d_u})$$



Stress field

Training:

$$\mathcal{L}_{\text{data}} = \frac{1}{n_{\text{data}}} \sum_{a=1}^{n_{\text{data}}} \sum_{b=1}^{n_{\text{discr}}} ||\hat{\boldsymbol{S}}_{ab} - \boldsymbol{S}_{ab}|| \qquad \mathcal{L}_{\text{PDE}} = \frac{1}{n_{\text{data}}} \sum_{a=1}^{n_{\text{data}}} ||\text{div}\hat{\boldsymbol{S}}_{a}||$$

Data ↓ PDE residual ↓

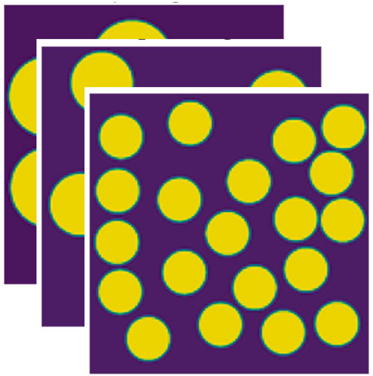
$$\min_{\theta} w_{\text{data}} \mathcal{L}_{\text{data}} + w_{\text{PDE}} \mathcal{L}_{\text{PDE}}$$

Physics-informed ML-based multiscale modeling

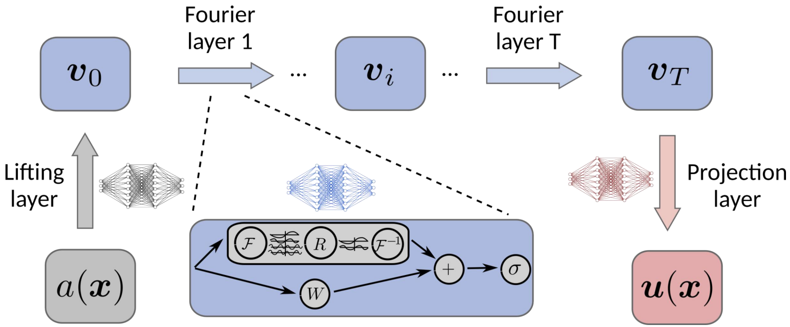
K.Karapiperis

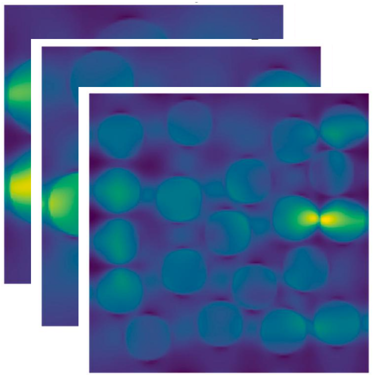
Physics-informed *neural operators*

The previous approach can only furnish the microscopic solution field for a given RVE. Neural operators instead learn the mapping between any RVE (or the same RVE with different loading conditions) and the corresponding microscopic solution field.



Material field





Stress field

Training:

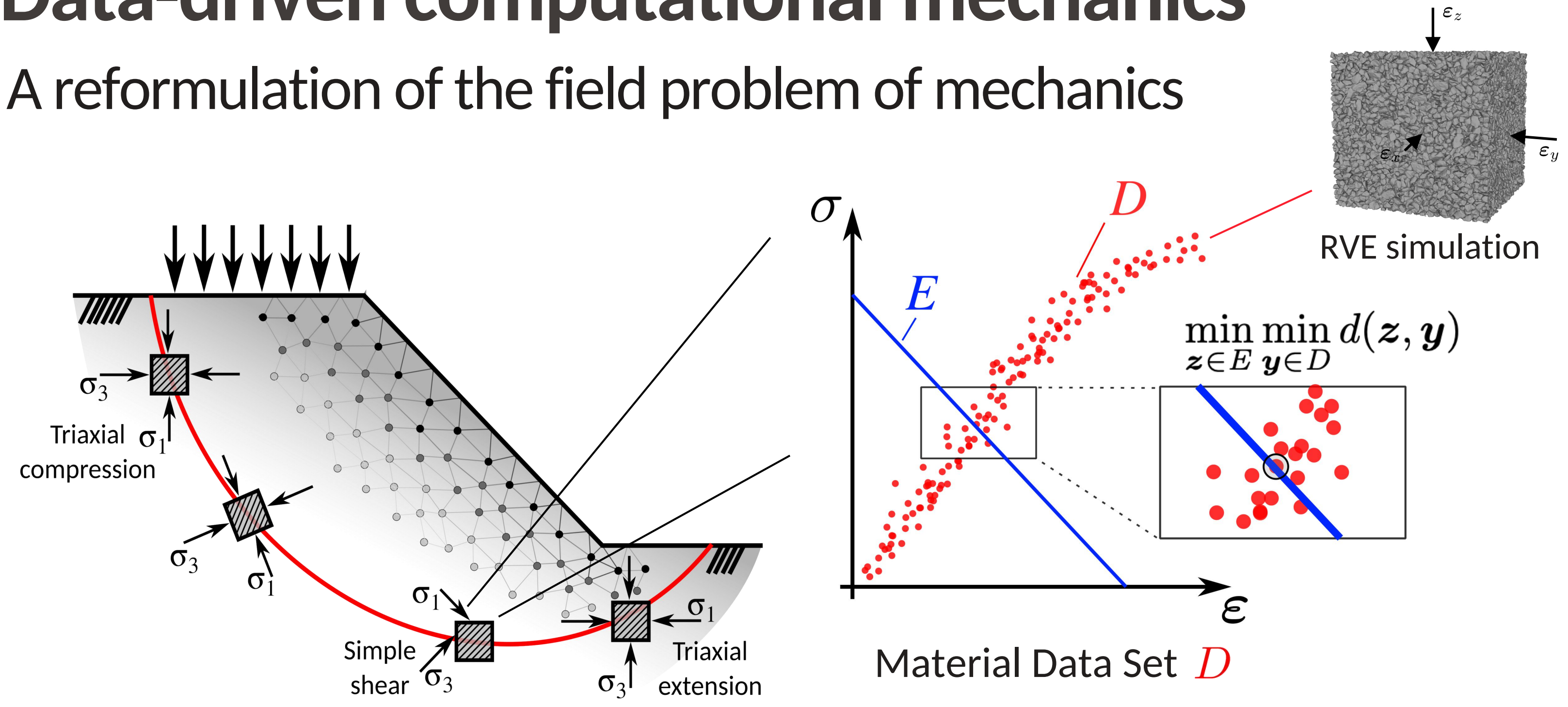
$$\mathcal{L}_{\text{data}} = \frac{1}{n_{\text{data}}} \sum_{a=1}^{n_{\text{data}}} \sum_{b=1}^{n_{\text{discr}}} ||\hat{\boldsymbol{S}}_{ab} - \boldsymbol{S}_{ab}|| \qquad \mathcal{L}_{\text{PDE}} = \frac{1}{n_{\text{data}}} \sum_{a=1}^{n_{\text{data}}} ||\text{div}\hat{\boldsymbol{S}}_{a}||$$

Data ↓ PDE residual ↓

$$\min_{\theta} w_{\text{data}} \mathcal{L}_{\text{data}} + w_{\text{PDE}} \mathcal{L}_{\text{PDE}}$$

Data-driven computational mechanics

A reformulation of the field problem of mechanics

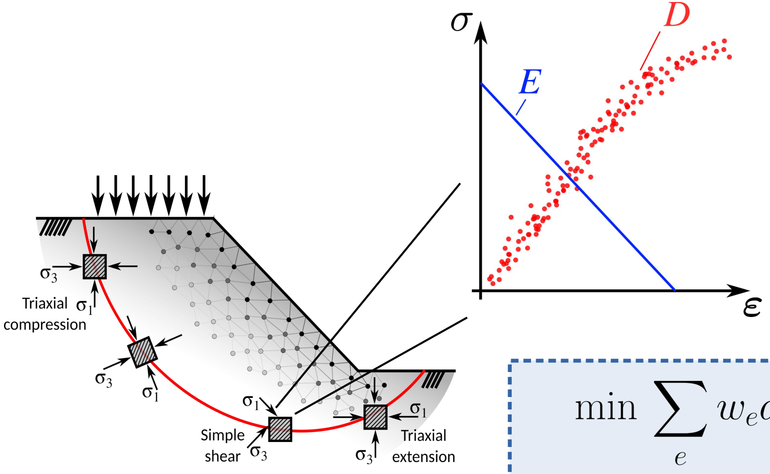


- No material model, no loss of information
- Direct use of data and physical principles, without any interpolation
- Eminent physical meaning (distance)

K.Karapiperis

Data-driven computational mechanics

A reformulation of the field problem of mechanics



Perform double minimization in staggered fashion

$$\min_{\boldsymbol{z}\in E}\min_{\boldsymbol{y}\in D} d(\boldsymbol{z},\boldsymbol{y})$$

$$\min \sum_e w_e d_e(\boldsymbol{\varepsilon}_e,\boldsymbol{\sigma}_e)$$

$$\text{s.t.}\quad \boldsymbol{\varepsilon}_e = \sum_i \boldsymbol{B}_{ei} u_i,$$

$$\sum_e w_e \boldsymbol{B}_{ei}^T \boldsymbol{\sigma}_e = F_i$$

→ Mechanical states

$$\mathbf{z} = \{(\boldsymbol{\varepsilon}^e,\boldsymbol{\sigma}^e)\}_{e=1,\dots,N}$$

$$d_e(\boldsymbol{\varepsilon}_e,\boldsymbol{\sigma}_e) = \min_{\boldsymbol{\varepsilon}_e^*,\boldsymbol{\sigma}_e^*\in D_e} (\boldsymbol{\varepsilon}_e-\boldsymbol{\varepsilon}_e^*)^T\mathbb{C}_e(\boldsymbol{\varepsilon}_e-\boldsymbol{\varepsilon}_e^*)+ (\boldsymbol{\sigma}_e-\boldsymbol{\sigma}_e^*)^T\mathbb{C}_e^{-1}(\boldsymbol{\sigma}_e-\boldsymbol{\sigma}_e^*)$$

→ Material states

$$\mathbf{y} = \{(\boldsymbol{\varepsilon}^{e*},\boldsymbol{\sigma}^{e*})\}_{e=1,\dots,N}$$

The stiffness $\mathbb{C}_e$ is simply a distance-inducing numerical parameter

Data-driven computational mechanics

Fixed-point algorithm

Iterative scheme, involving:

i) Solution of two modified ‘elasticity’ problems

ii) Database search

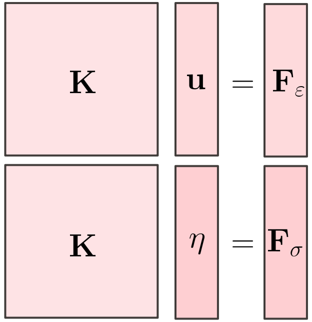
$$\Big(\sum_{e=1}^{M} w_e \mathbf{B}_e^T \mathbb{C}_e \mathbf{B}_e\Big)\mathbf{u}^{(i)} = \sum_{e=1}^{M} w_e \mathbf{B}_e^T \mathbb{C}_e \varepsilon_e^{*(i)} \quad (1)$$

$$\Big(\sum_{e=1}^{M} w_e \mathbf{B}_e^T \mathbb{C}_e \mathbf{B}_e\Big)\boldsymbol{\eta}^{(i)} = \mathbf{f} - \sum_{e=1}^{M} w_e \mathbf{B}_e^T \sigma_e^{*(i)} \quad (2)$$

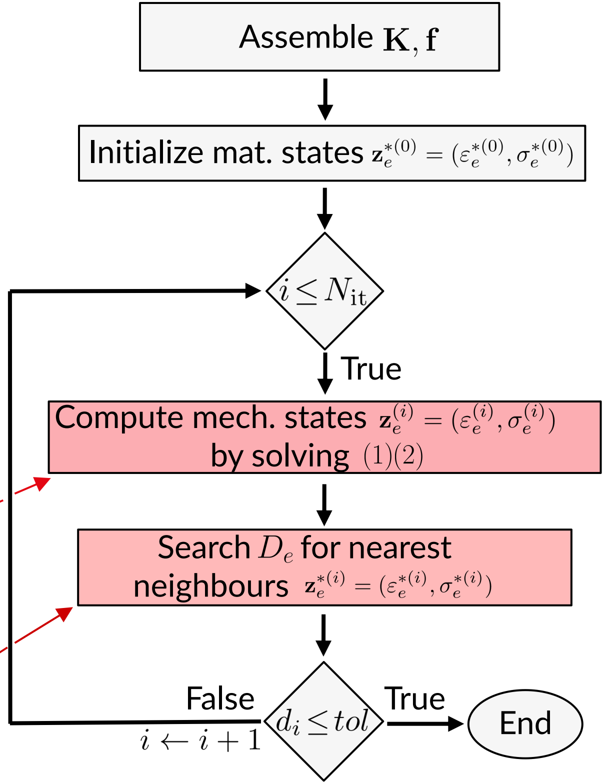
$$\sigma_e^{(i)} = \sigma_e^{*(i)} + \mathbb{C}_e \sum_{\alpha=1}^{N} \mathbf{B}_{e\alpha} \eta_\alpha^{(i)}$$

Positive-definite distance-inducing tensor

$$|\boldsymbol{z}^e| = \mathbb{C}^e \boldsymbol{\varepsilon}^e \cdot \boldsymbol{\varepsilon}^e + \mathbb{C}^{e^{-1}} \boldsymbol{\sigma}^e \cdot \boldsymbol{\sigma}^e$$



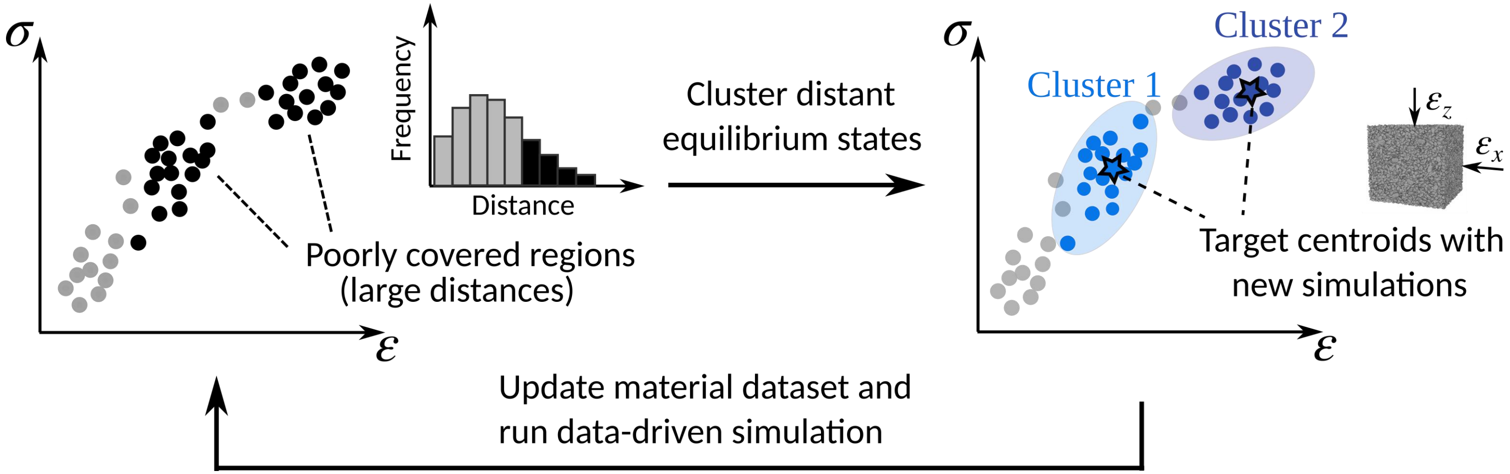
Major computational burden (tree-search)



Data-driven computational mechanics

What if data are scarce?

Iteratively sample poorly populated regions of the phase space



Adaptive sampling via *unsupervised learning*
Repeat until distance is small enough

K.Karapiperis

Data-driven computational mechanics

History-dependent (inelastic) formulation

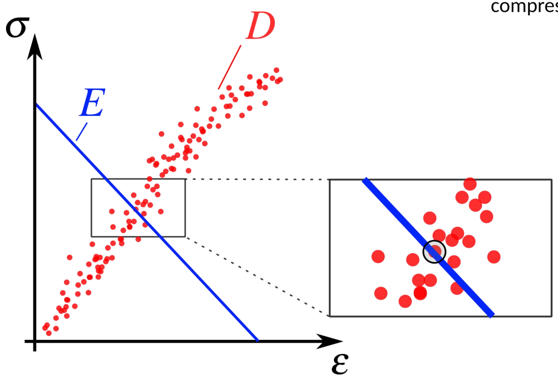
Original formulation

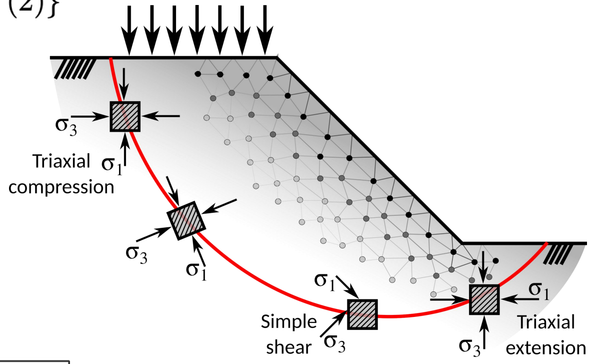
Phase Space: $Z = \{(\boldsymbol{\epsilon}, \boldsymbol{\sigma})\}$

Equilibrium Set: $E = \{\boldsymbol{z} \in Z \mid (1), (2)\}$

Material Data Set: $D \subset Z$

$$\min_{\boldsymbol{z} \in E} \min_{\boldsymbol{y} \in D} d(\boldsymbol{z}, \boldsymbol{y})$$





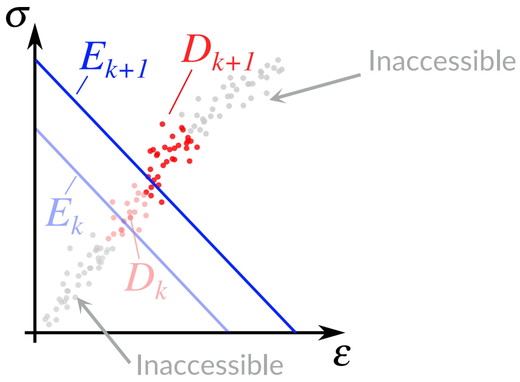
History-dependent extension

History-dependence ⟶ Notion of time

Time-dependent Equilibrium Set: $E_k \subset Z$

Time-dependent Material Data Set: $D_k \subset Z$

$$\min_{\mathbf{z}_k \in E_k} \min_{\mathbf{y}_k \in D_k} d(\mathbf{z}_k, \mathbf{y}_k)$$



Data-driven computational mechanics

Modified solution algorithm

Iterative scheme, involving:

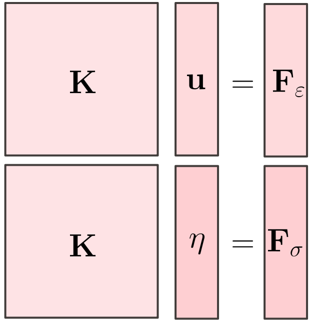
i) Solution of two modified ‘elasticity’ problems

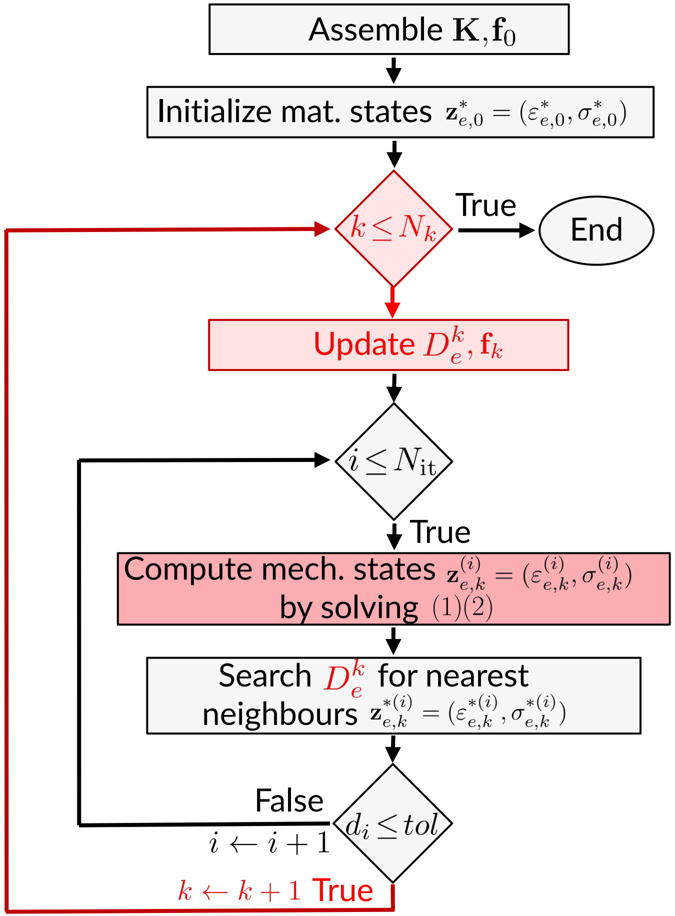
ii) Database search

$$\Big(\sum_{e=1}^{M} w_e \mathbf{B}_e^T \mathbb{C}_e \mathbf{B}_e\Big)\mathbf{u}_k^{(i)} = \sum_{e=1}^{M} w_e \mathbf{B}_e^T \mathbb{C}_e \varepsilon_{e,k}^{*(i)}$$

$$\Big(\sum_{e=1}^{M} w_e \mathbf{B}_e^T \mathbb{C}_e \mathbf{B}_e\Big)\boldsymbol{\eta}_k^{(i)} = \mathbf{f}_k - \sum_{e=1}^{M} w_e \mathbf{B}_e^T \sigma_{e,k}^{*(i)}$$

$$\sigma_{e,k}^{(i)} = \sigma_{e,k}^{*(i)} + \mathbb{C}_e \sum_{\alpha=1}^{N} \mathbf{B}_{e\alpha} \eta_{\alpha,k}^{(i)}$$





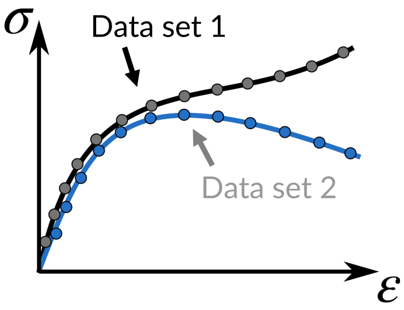
K.Karapiperis

Data-driven computational mechanics

History parametrization

History-matching

$$D_{k+1} = \{(\boldsymbol{\epsilon}_{k+1}, \boldsymbol{\sigma}_{k+1}) \,|\, \{\boldsymbol{\epsilon}_l\}_{l \leq k}\}$$

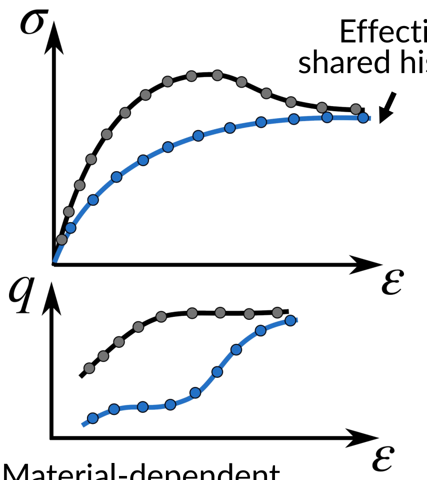


Material-independent
Inefficient

Internal variable

$$D_{k+1} = \{(\boldsymbol{\epsilon}_{k+1}, \boldsymbol{\sigma}_{k+1}) \,|\, (\boldsymbol{\epsilon}_k, \boldsymbol{\sigma}_k, \mathbf{q}_k)\}$$

Internal variable: $\mathbf{q} = \{\mathbf{F}, \ldots\}$



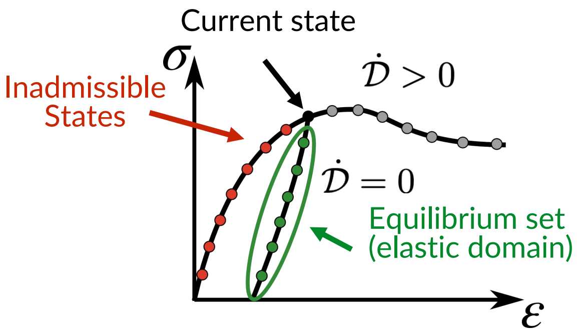
Material-dependent
Requires access to micromechanics
Augments space by set of int. variables

Thermodynamics

$$D_{k+1} = \{(\boldsymbol{\epsilon}_{k+1}, \boldsymbol{\sigma}_{k+1}) \,|\, (\boldsymbol{\epsilon}_k, \boldsymbol{\sigma}_k), (1)\}$$

$$\mathcal{D}_{k+1} - \mathcal{D}_k = \frac{\boldsymbol{\sigma}_k + \boldsymbol{\sigma}_{k+1}}{2} : (\boldsymbol{\epsilon}_{k+1} - \boldsymbol{\epsilon}_k) - (\mathcal{A}_{k+1} - \mathcal{A}_k) \geq 0 \quad (1)$$

Dissipation ($\mathcal{D}$)
Free energy ($\mathcal{A}$)



Material-independent

Data-driven computational mechanics

Application to granular materials

Free energy/Dissipation

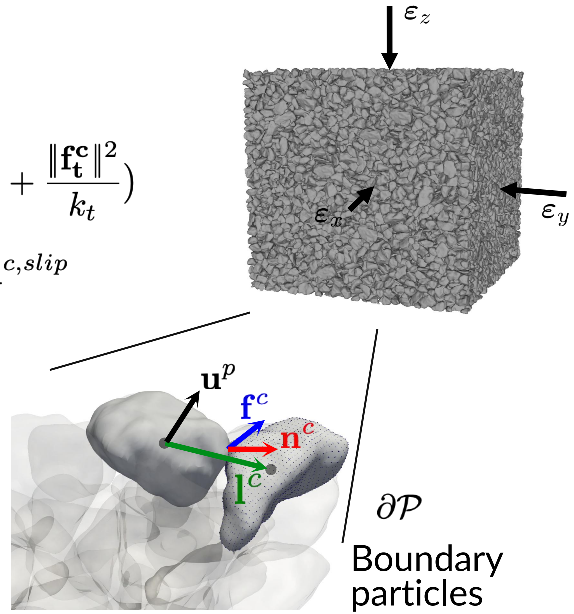
$$\boldsymbol{\varepsilon} = \frac{1}{2V}\mathrm{sym}\Big(\sum_{p\in\partial\mathcal{P}} \mathbf{u}^p \otimes \mathbf{n}^p\Big)$$

$$\boldsymbol{\sigma} = \frac{1}{V}\mathrm{sym}\Big(\sum_{c\in\mathcal{C}} \mathbf{f}^c \otimes \mathbf{l}^c\Big)$$

$$\mathcal{A} = \sum_c \mathcal{A}^c = \frac{1}{2V}\sum_c \Big(\frac{\|\mathbf{f_n^c}\|^2}{k_n} + \frac{\|\mathbf{f_t^c}\|^2}{k_t}\Big)$$

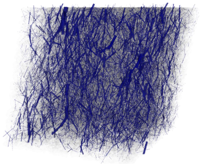
$$d\mathcal{D} = \sum_c d\mathcal{D}^c = \frac{1}{V}\sum_c \mathbf{f}_t^c \cdot d\mathbf{u}^{c,slip}$$

Granular RVE



Internal variables

$$\mathbf{q} = P_{\mathbf{nfl}}(\mathbf{n},\mathbf{f},\mathbf{l})$$
$$= \underbrace{P_{\mathbf{n}}(\mathbf{n})}_{\text{Contact structure}} P_{\mathbf{f}|\mathbf{n}}(\mathbf{f}|\mathbf{n}) P_{\mathbf{l}|\mathbf{f},\mathbf{n}}(\mathbf{l}|\mathbf{f},\mathbf{n})$$



Dimensionality reduction (Second-order statistics)

$$P_{\mathbf{n}}(\mathbf{n}) = \frac{1}{4\pi}(1 + \mathbf{n}\cdot\mathbf{F}\mathbf{n})$$

Collect variables

$$\mathbf{q} = \{\mathbf{F}, \ldots\}$$

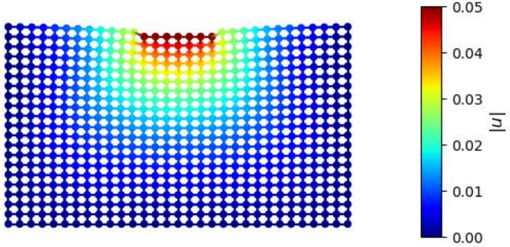
Fabric tensor

Fabric

Outlook

- ML-based multiscale modeling: Tomorrow's exercise

- DDCM-based multiscale modeling: Next week's exercise



Discussion

That's what I prepared for you today.

What would you like to discuss?

Reading for next class:

Collection of research articles

(Week 11 - Reading Assignments)