A Convolutional Neural Network for Enhancement of Multi-Scale Localization in Granular Metallic Representative Unit Cells

(+- *my other interests*)

Peter A. Gustafson

Western Michigan University, Kalamazoo, MI, 49008, USA
NASA Glenn Research Center, Cleveland, OH, 44135, USA

Evan J. Pineda, Trenton M. Ricks, Brett A. Bednarcyk,
Brandon L. Hearley and Josh Stuckner

NASA Glenn Research Center, Cleveland, OH, 44135, USA

Mostly from AIAA SciTech 2022

Me:

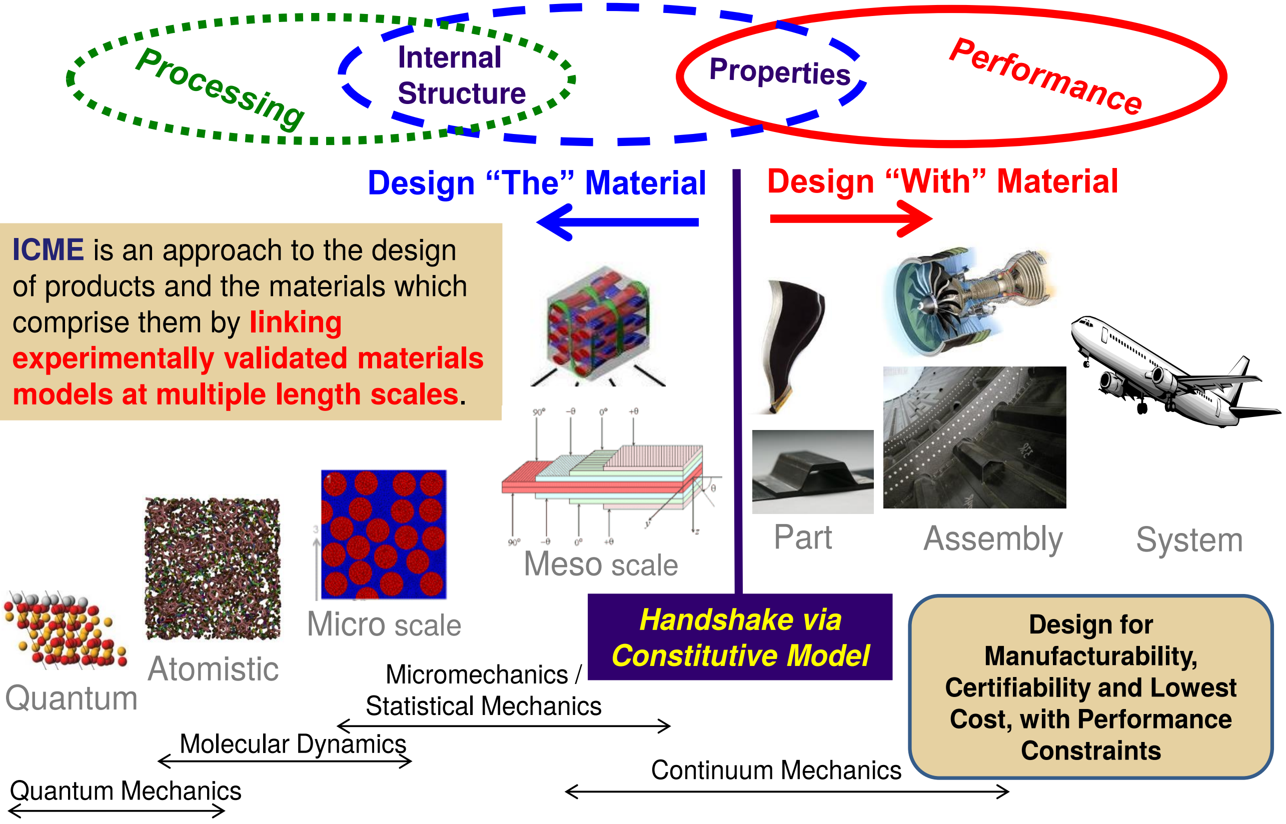
- Professor of Mechanical and Aerospace Engineering
  - On sabbatical from Western Michigan University
  - Working with Josh Stuckner, Evan Pineda
  - and others from Materials Branch (Multi-scale modeling)

- Research Interests
  - Computational Structural Mechanics (primarily FEA)
  - Aerospace Structures
  - Composite Materials
  - Biomedical Engineering (tissue modeling, devices)

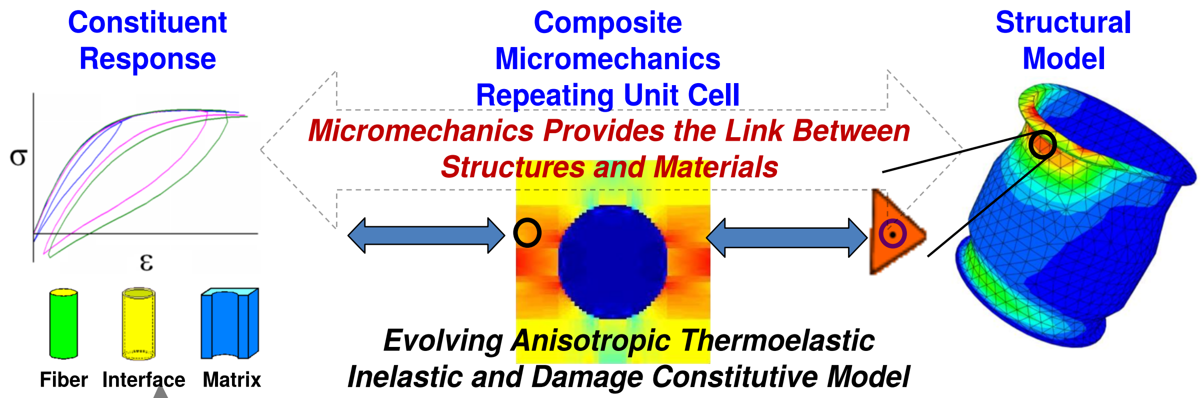
- Learning goals for the sabbatical
  - Multi-scale modeling
  - Machine learning

Relevance and Background

Integrated Computational Materials Engineering (ICME) Is The Future



Multiscale Modeling



Continuum Deformation and damage models

Macro-level Constitutive Equations

$$\bar{\sigma} = B^*\left(\bar{\varepsilon} - \bar{\varepsilon}^I - \bar{\varepsilon}^T\right)$$

$$B^* = \frac{1}{dhl}\sum_{\alpha=1}^{N_\alpha}\sum_{\beta=1}^{N_\beta}\sum_{\gamma=1}^{N_\gamma} d_\alpha h_\beta l_\gamma C^{(\alpha\beta\gamma)} A^{(\alpha\beta\gamma)}$$

$$\bar{\varepsilon}^I = \frac{-\left[B^*\right]^{-1}}{dhl}\sum_{\alpha=1}^{N_\alpha}\sum_{\beta=1}^{N_\beta}\sum_{\gamma=1}^{N_\gamma} d_\alpha h_\beta l_\gamma C^{(\alpha\beta\gamma)}\left(D^{(\alpha\beta\gamma)}\varepsilon_s^I - \varepsilon^{I(\alpha\beta\gamma)}\right)$$

$$\bar{\varepsilon}^T = \frac{-\left[B^*\right]^{-1}}{dhl}\sum_{\alpha=1}^{N_\alpha}\sum_{\beta=1}^{N_\beta}\sum_{\gamma=1}^{N_\gamma} d_\alpha h_\beta l_\gamma C^{(\alpha\beta\gamma)}\left(D^{(\alpha\beta\gamma)}\varepsilon_s^T - \varepsilon^{T(\alpha\beta\gamma)}\right)$$

Micro-level Field Equations (subcell)

$$\varepsilon^{(\alpha\beta\gamma)} = A^{(\alpha\beta\gamma)}\bar{\varepsilon} + D^{(\alpha\beta\gamma)}\left(\varepsilon_s^I + \varepsilon_s^T\right)$$

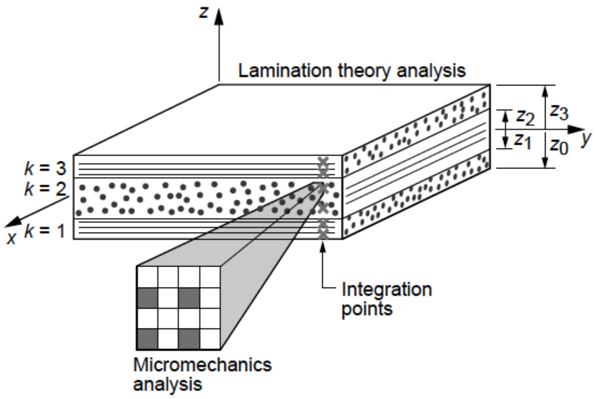
$$\sigma^{(\alpha\beta\gamma)} = C^{(\alpha\beta\gamma)}\left[A^{(\alpha\beta\gamma)}\bar{\varepsilon} + D^{(\alpha\beta\gamma)}\left(\varepsilon_s^I + \varepsilon_s^T\right) - \left(\varepsilon^{I(\alpha\beta\gamma)} + \varepsilon^{T(\alpha\beta\gamma)}\right)\right]$$

Aboudi, J., Arnold, S.M., and Bednarcyk, B.A. (2013) *Micromechanics of Composite Materials: A Generalized Multiscale Analysis Approach*, Elsevier, Oxford, UK., pp 1-984.

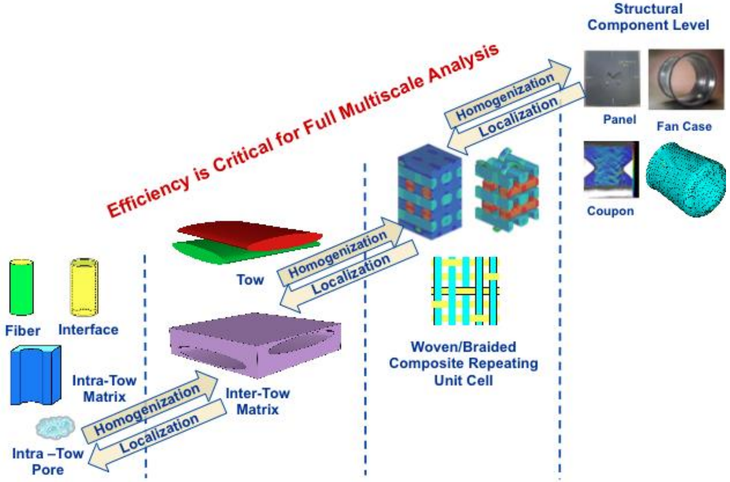
NASA tools heavily dependent on the **Generalized Method of Cells**

Robust Multiscale Modeling Framework Enables Efficient Analysis of Composite Materials and Structures

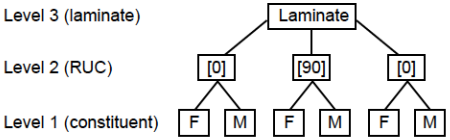
Classical Lamination Theory (CLT)



Woven/Braided Composite Systems

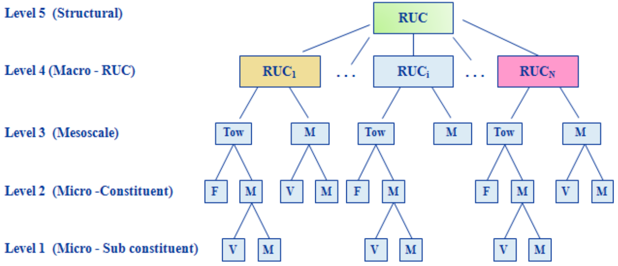


Multiscale Analysis Diagrams (MsGMC):



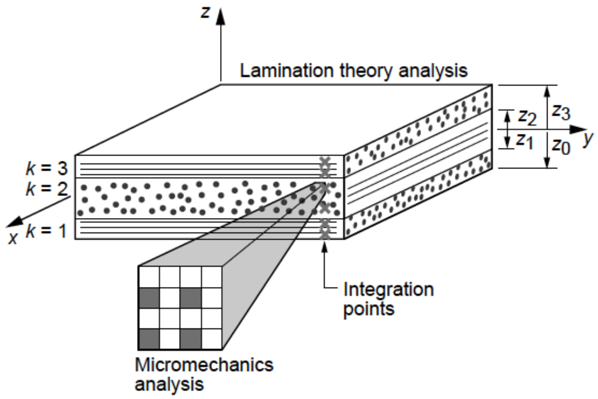
3 Scales with 2 Homogenizations/Localizations

5 Scales with 4 Homogenizations/Localizations

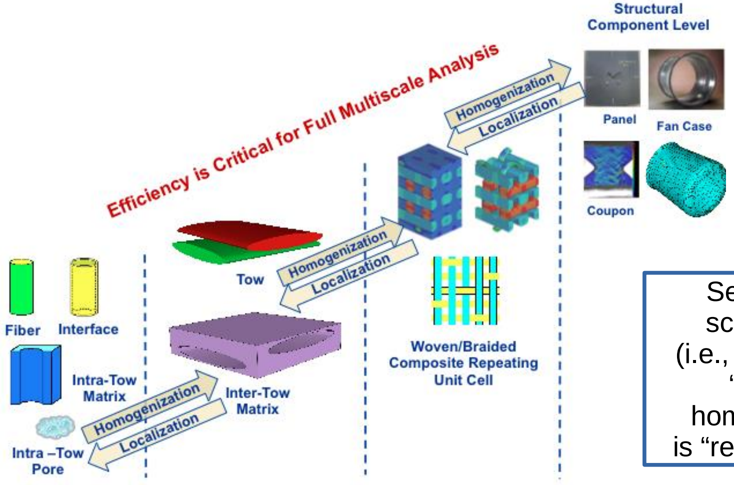


Robust Multiscale Modeling Framework Enables Efficient Analysis of Composite Materials and Structures

Classical Lamination Theory (CLT)

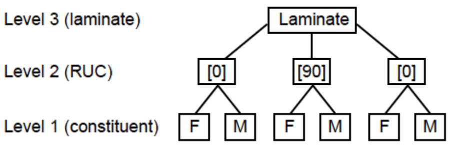


Woven/Braided Composite Systems



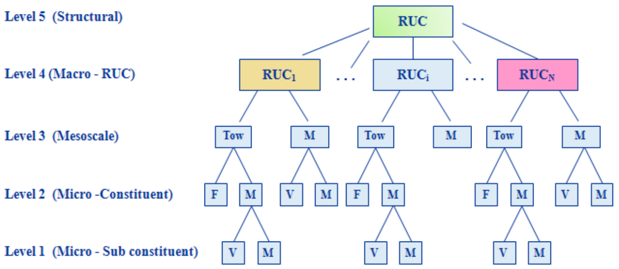
Separation of scales critical (i.e., localization is “uniform”, homogenization is “representative”)

Multiscale Analysis Diagrams (MsGMC):



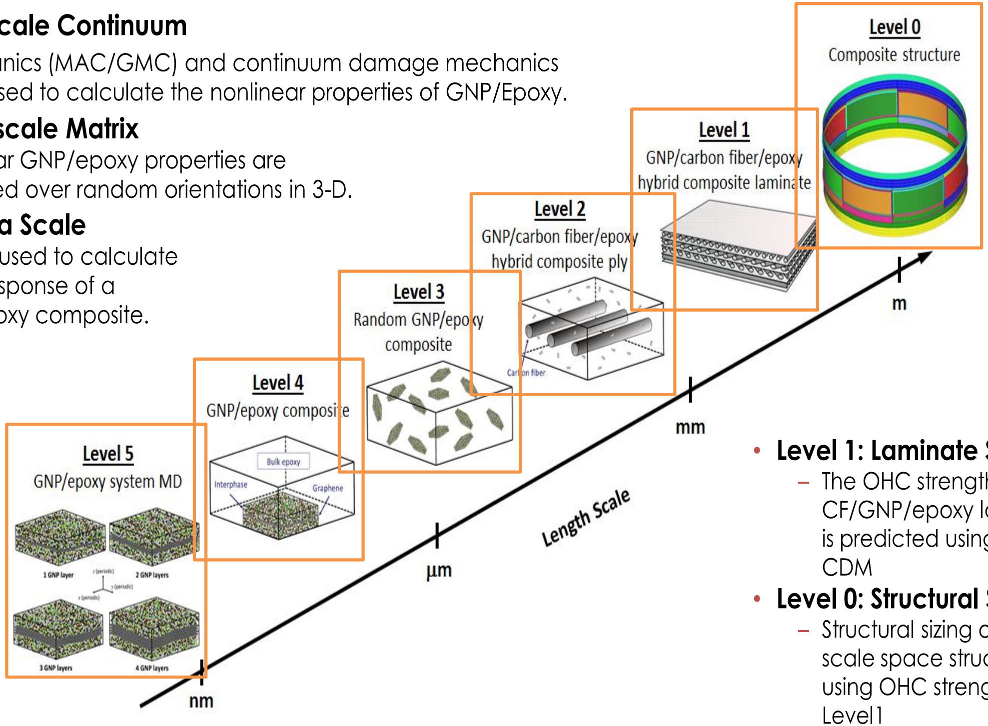
3 Scales with 2 Homogenizations/Localizations

5 Scales with 4 Homogenizations/Localizations



Thermomech. Multiscale Modeling of Large Space Structure Including GNP

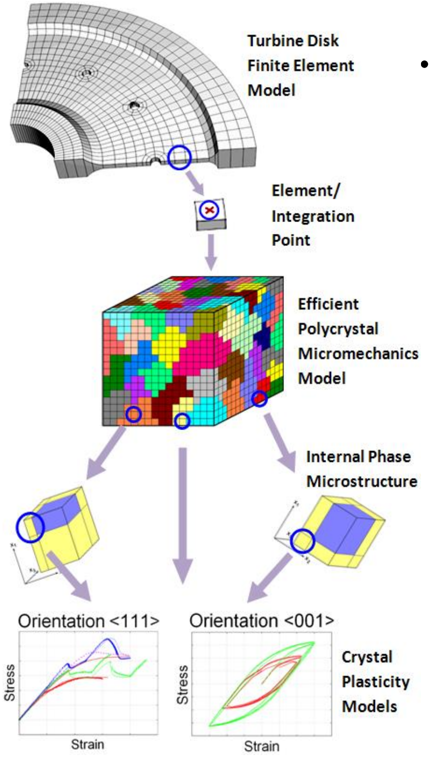
- **Level 5: Nanoscale**
  - LAMMPS Molecular Dynamics (MD) code used to predict thermoelastic properties of GNP/epoxy
- **Level 4: Nanoscale Continuum**
  - Micromechanics (MAC/GMC) and continuum damage mechanics (CDM) are used to calculate the nonlinear properties of GNP/Epoxy.
- **Level 3: Microscale Matrix**
  - The nonlinear GNP/epoxy properties are homogenized over random orientations in 3-D.
- **Level 2: Lamina Scale**
  - MAC/GMC used to calculate nonlinear response of a CF/GNP/Epoxy composite.



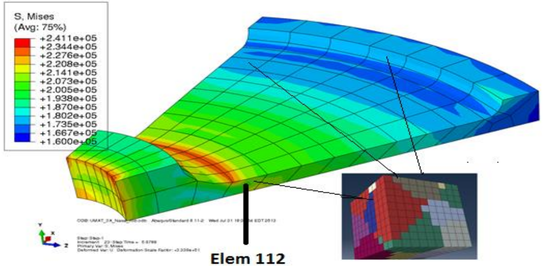
- **Level 1: Laminate Scale.**
  - The OHC strength of CF/GNP/epoxy laminates is predicted using FEA and CDM
- **Level 0: Structural Scale.**
  - Structural sizing of full-scale space structure using OHC strengths from Level1

Multiscale Model for Polycrystalline Structures
Microstructure/Property Relationship

Challenge Problem

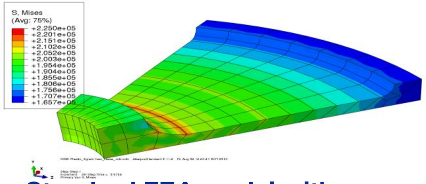


- Utilizes NASA's ultra-efficient micromechanics methods to link grain and sub-grain behavior to the performance of the structure.
- For polycrystals consisting of 10s to 100s of grains – GMC exhibited 2 to 3 orders of magnitude speed up compared to conventional FEA with minimal loss of accuracy.



FEAMAC used to link macro-scale to micro-scale (each grain crystal plasticity)

Von Misses stress distribution

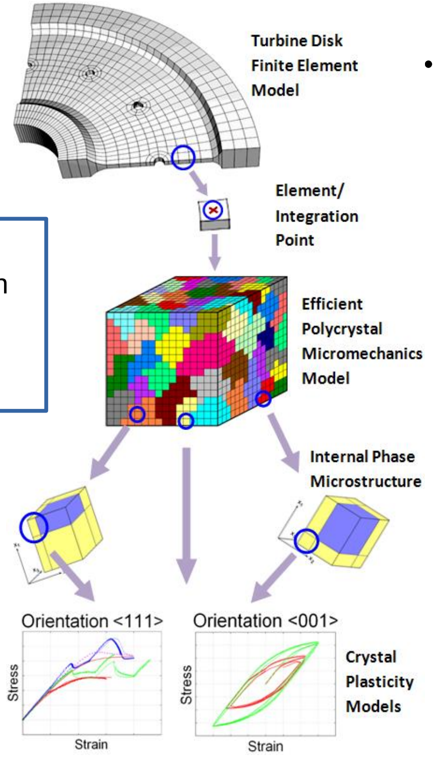


Standard FEA model with macroscale constitutive relationship

Multiscale Model for Polycrystalline Structures

Microstructure/Property Relationship

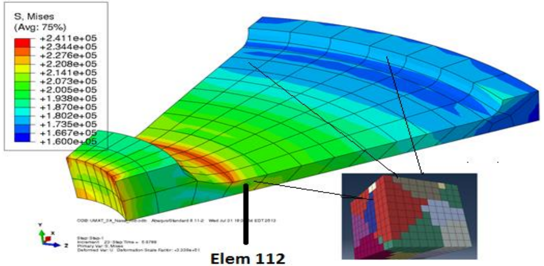
Challenge Problem



This presentation focused on this problem!!!

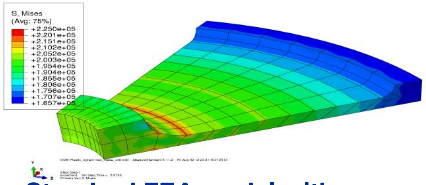
- Utilizes NASA's ultra-efficient micromechanics methods to link grain and sub-grain behavior to the performance of the structure.
- For polycrystals consisting of 10s to 100s of grains – GMC exhibited 2 to 3 orders of magnitude speed up compared to conventional FEA with minimal loss of accuracy.

For homogenization (when comparable assumptions included)



FEAMAC used to link macro-scale to micro-scale (each grain crystal plasticity)

Von Misses stress distribution



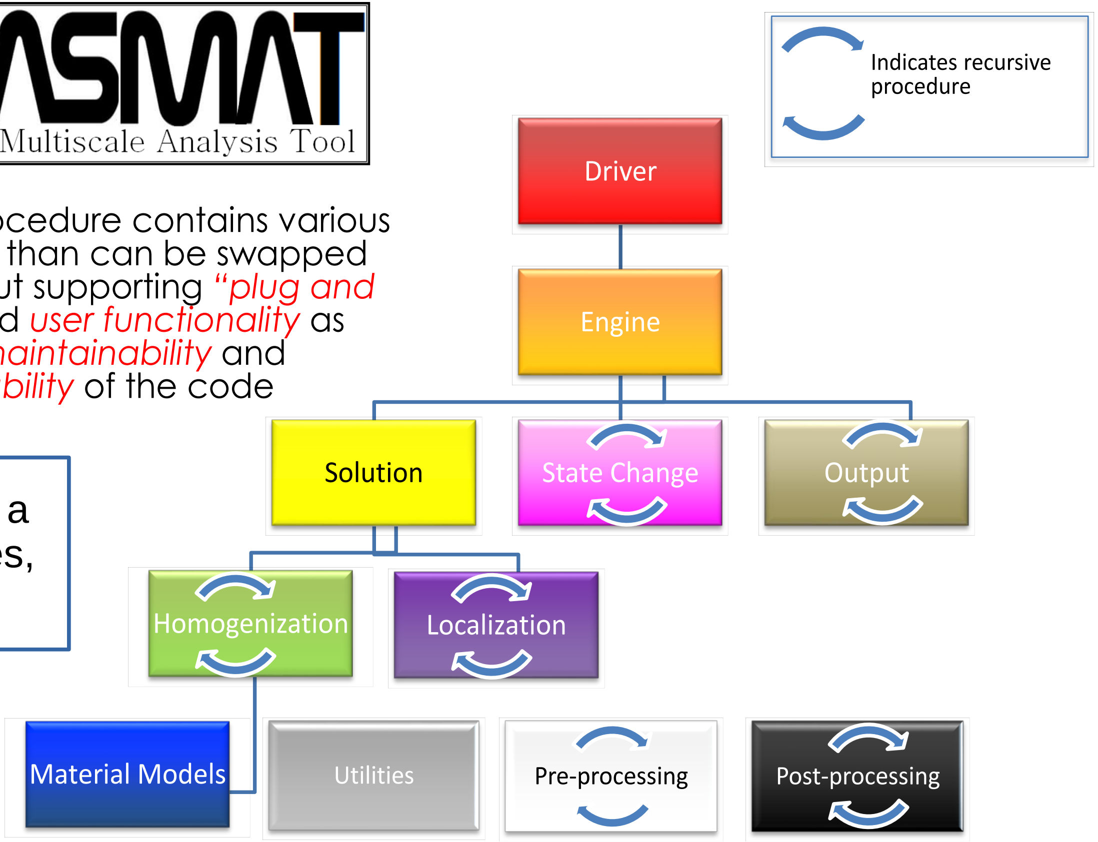
Standard FEA model with macroscale constitutive relationship

"Damage" (i.e., yield, fracture, etc) typically dependent on invariants

NASA Multiscale Analysis Tool

NASMAT
NASA Multiscale Analysis Tool

- Each procedure contains various modules than can be swapped in and out supporting *"plug and play"* and *user functionality* as well as *maintainability* and *upgradability* of the code

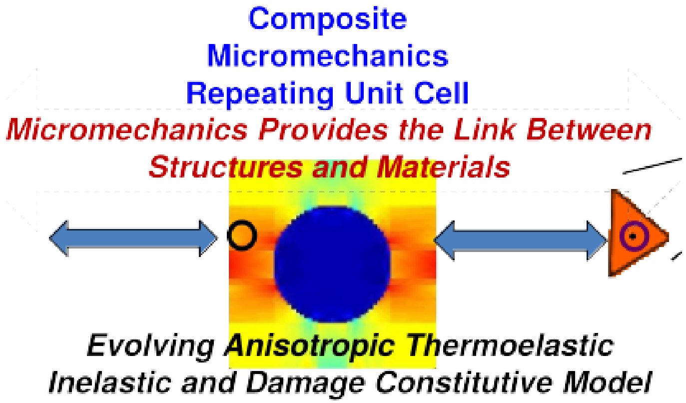


Note: parallelization is a challenge across scales, (current push)

Method of cells limitations (GMC)
(especially with respect to localization of damage mechanics)

- Normal stresses are assumed constant, consequently shear stresses are also assumed constant

$$\sigma_{xx,x} + \sigma_{yx,y} + \sigma_{zx,z} + b_x = \rho\, a_x$$
$$\sigma_{xy,x} + \sigma_{yy,y} + \sigma_{zy,z} + b_y = \rho\, a_y$$
$$\sigma_{xz,x} + \sigma_{yz,y} + \sigma_{zz,z} + b_z = \rho\, a_z$$



- There is no normal-shear stress coupling in equilibrium (i.e., no shear lag)
    - Lack of coupling draws into question localized damage
    - Especially when not invariant based
- The spatial distribution of invariants within GMC are impacted

Goals

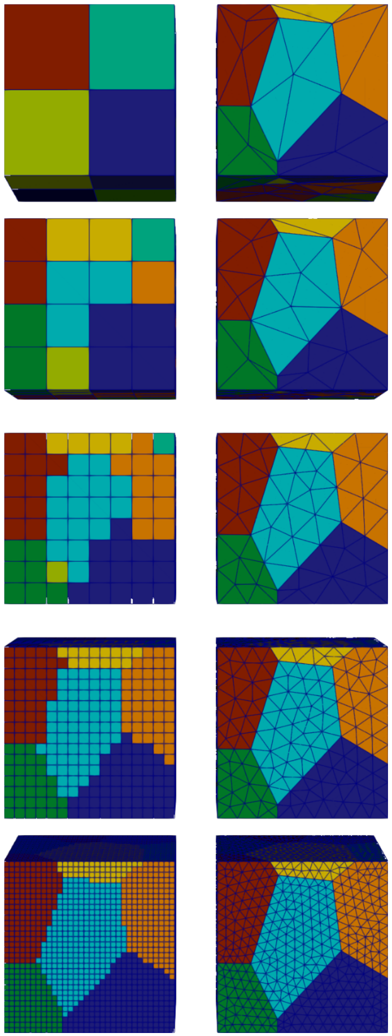
- To increase the accuracy of GMC models <u>without sacrificing speed</u>
  - Primarily focused on hierarchical multi-scale models and <u>***localization***</u>
    - (In my view, accurate damage prediction requires accurate localization whereas accurate stiffness converges quickly via homogenization)
  - Permit accurate shear-normal coupling to allow for effective failure analysis
    - (and filtering or down-selection for hierarchical models)

Side Note: My work is complementary to Josh's ML work on NASMAT

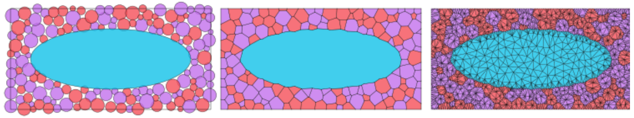
- Josh is generating ML GMC/HFGMC model predictions
  - GMC/HFGMC inputs used as ML inputs for GMC/HFGMC outputs
  - Fast, including damage mechanics
  - <u>Still subject to the limitations inherent in the generalized method of cells (or HFGMC)</u>

Method - Models

- Training data:
  - Finite element and GMC models
  - GMC RUCs with varying numbers of subcells per model.
- The left column visualizes both the GMC microstructural representation and the voxelized finite element model (C3D8)
- The right columns represent varying mesh densities of the same microstructure.
  - ML models were trained for all of these mesh densities
  - General approach less dependent of mesh density?



Process for generating ML result



- MicroStructPy
  - Create 3D grain structure
  - Based on Voronoi
- Gmsh (Initial meshing)
- CalculiX CGX (Mesh Order Change)
- Python scripts
  - Voxelized FE and NASMAT
  - BCs and Loads
- CalcuiliX CCX (FEA solver)
- NASMAT
- Hdf5 for data storage
- Paraview (postprocessing)
- Tensorflow (Machine Learning)

Gmsh

A three-dimensional finite element mesh generator with built-in pre-processing facilities

Christophe Geuzaine and Jean-François Remacle

Download | Documentation | Licensing | Screenshots | Links | References |

Gmsh is an open source 3D finite element mesh generator with a built-in CAD engine and post-processor. Its design goal is to provide a fast, l meshing tool with parametric input and advanced visualization capabilities. Gmsh is built around four modules: geometry, mesh, solver and po specification of any input to these modules is done either interactively using the graphical user interface, in ASCII text files using Gmsh's own s files), or using the C++, C, Python or Julia Application Programming Interface (API).

See this general presentation for a high-level overview of Gmsh and recent developments, the screencasts for a quick tour of Gmsh's graphic reference manual for a more thorough overview of Gmsh's capabilities, some frequently asked questions and the documentation of the C++, C

The source code repository contains many examples written using both the built-in script language and the API (see e.g. the tutorials and the

Download

CALCULIX

A Free Software Three-Dimensional Structural Finite Element Pro

Version 2.18 is available

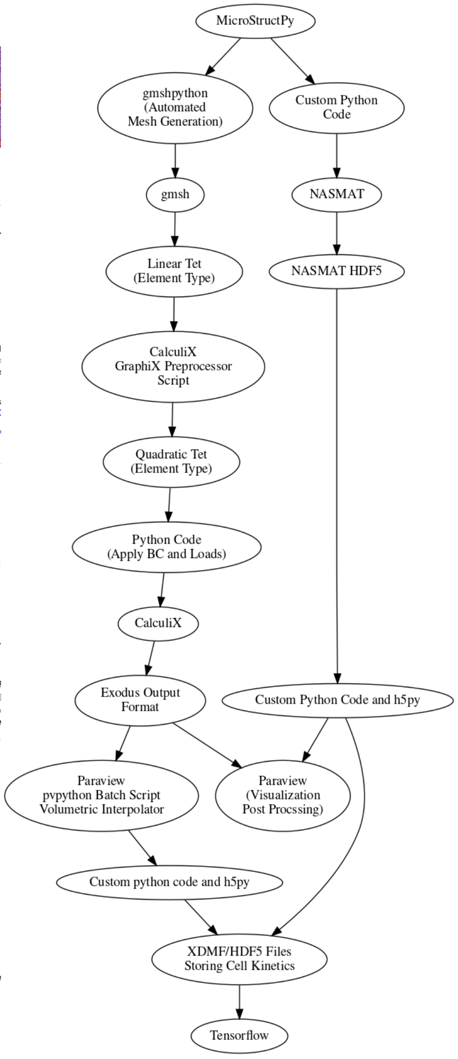
Last updated: 19. Sep 19:29:40 CEST 2021

CalculiX is a package designed to solve field problems. The method used is the finite element method.

With CalculiX Finite Element Models can be built, calculated and post-processed. The pre- and post-processor is an interactive 3 openGL API. The solver is able to do linear and non-linear calculations. Static, dynamic and thermal solutions are available. Bot used independently. Because the solver makes use of the abaqus input format it is possible to use commercial pre-processors as w pre-processor is able to write mesh related data for nastran, abaqus, ansys, code-aster and for the free-cfd codes dolfyn, duns, ISA OpenFOAM. A simple step reader is included. In addition external CAD interfaces are available. The program is designed to run platforms like Linux and Irix computers but also on MS-Windows.

ParaView

TensorFlow



Model boundary conditions

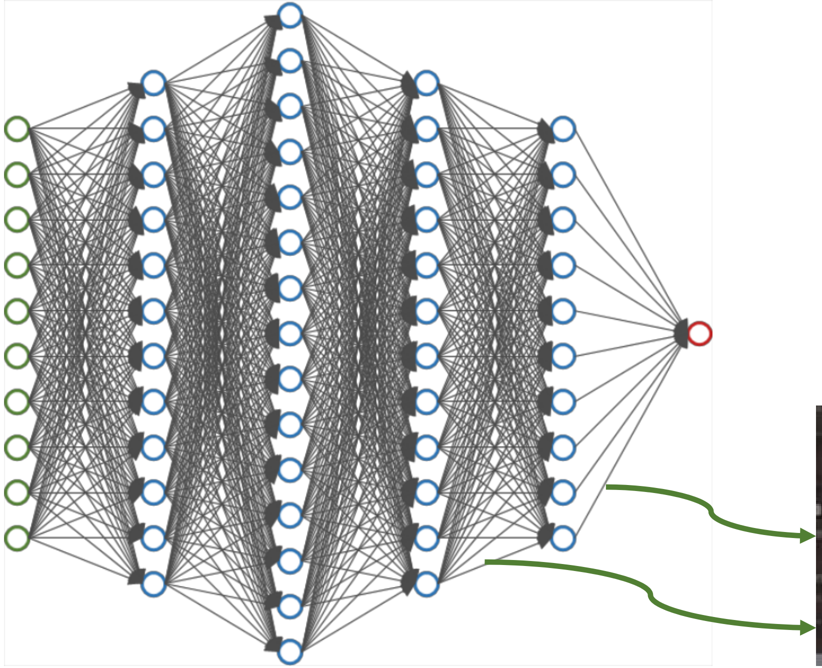
- FEA
  - Planar MPCs for face nodes, controlled by corner nodes
  - Strain control imposed on corner nodes as steps
- NASMAT
  - User strain control (one strain per model)
  - Traction periodicity on boundary subcell faces (NASMAT standard)
- Both
  - 6 isolated strain components at +- 0.001 level
  - 12 total loads (in 12 steps or 12 models)
- 1000 Randomized microstructures, 12,000 total samples

Tensorflow details

TensorFlow

- Strains were scaled by a factor of 1000
  - (thus, were approximately in the range of -1 to 1).
- Stresses were scaled by a factor of 1/1000
  - (and thus were also in the -1 to 1 order of magnitude).
- Data sets were split into training (80%), validation (15%), and testing (5%)
- The ‘relu’ activation function was used in each layer
  - **Except the final dense layer**
  - Utilized a ‘linear’ activation function to permit tensile and compressive output
- Loss function - ‘mean squared error’
- Optimizer - ‘adam’.

Training (Obligatory Representation Satisfied)

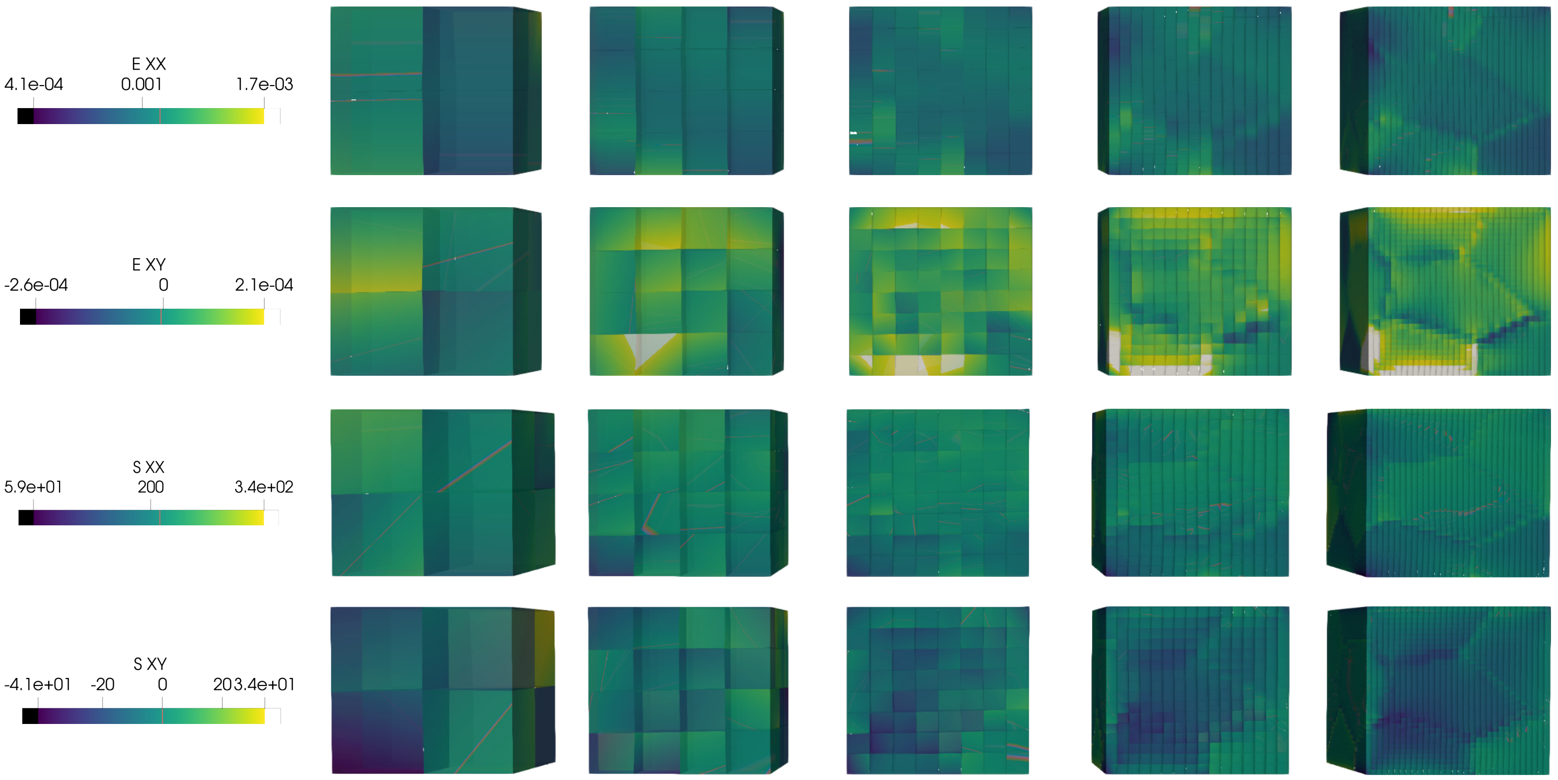


Weight update:

$$w \leftarrow w - \alpha \frac{\partial L}{dw}$$

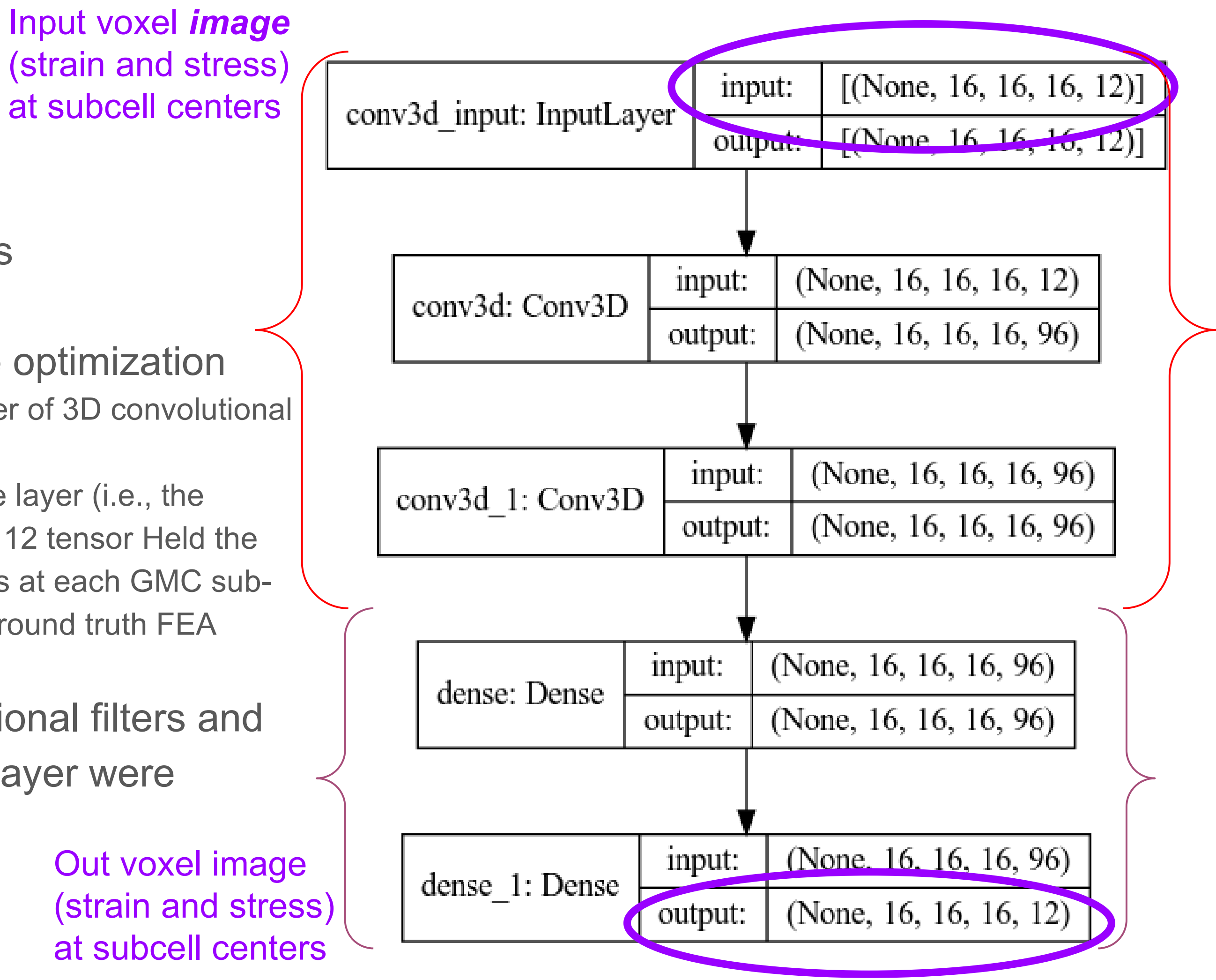
joshua.stuckner@nasa.gov

FEA Stress and strain predictions



Architecture

- # of Convolutional layers
- # of Dense layers
- Grid search architecture optimization
  - Each combining a number of 3D convolutional and dense layers.
  - The input and final dense layer (i.e., the output) was a n x n x n x 12 tensor Held the input stresses and strains at each GMC sub-cell and corresponding ground truth FEA outputs of the same.
- The number of convolutional filters and dense neurons in each layer were integer multiplier of 12.

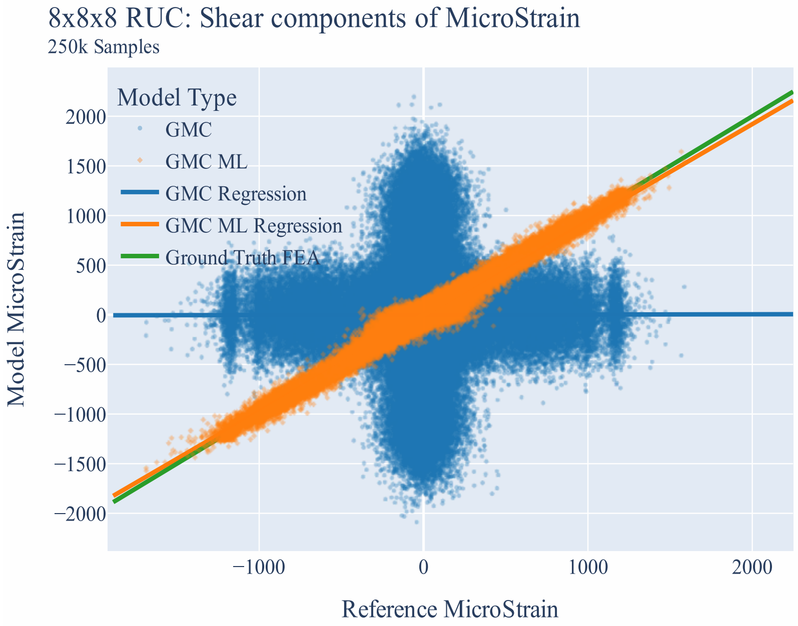


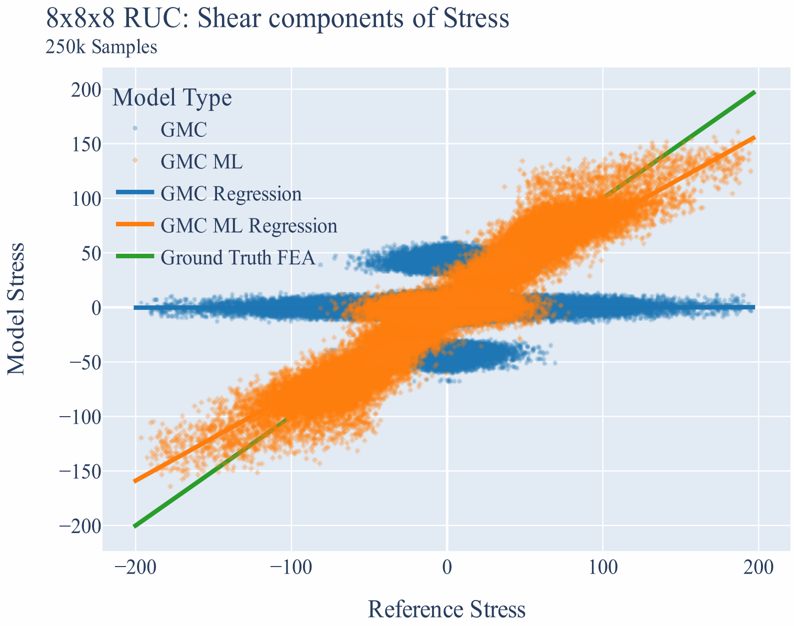
Physical measures of ML model quality?

- Training on minimization of the loss function
  - Hard (for me) to interpret
  - Especially in the context of mixed units on inputs
  - Scaling choices *<u>necessarily impact loss function</u>*
- After training, consider Strain/Stress slopes as a physically meaningful metric
  - Each plotted against the ground truth
    - for each cumulative subcell of each RUC
    - Linear slope and Pearson correlation (r^2) values were also used to assess model quality.
      - Ideal → slope of 1 and an r^2 value of 1
    - Scatter plots for each ML model, visual evaluation

What does the goal look like? What is ideal?

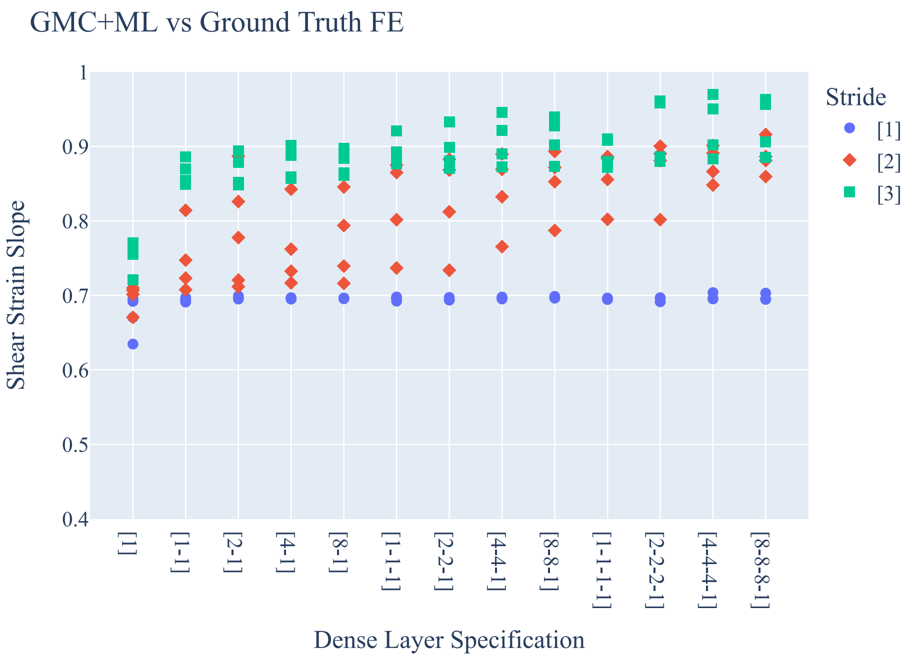
- This example, nearly ideal strain correction
- Similar stress correction (though not as good here)

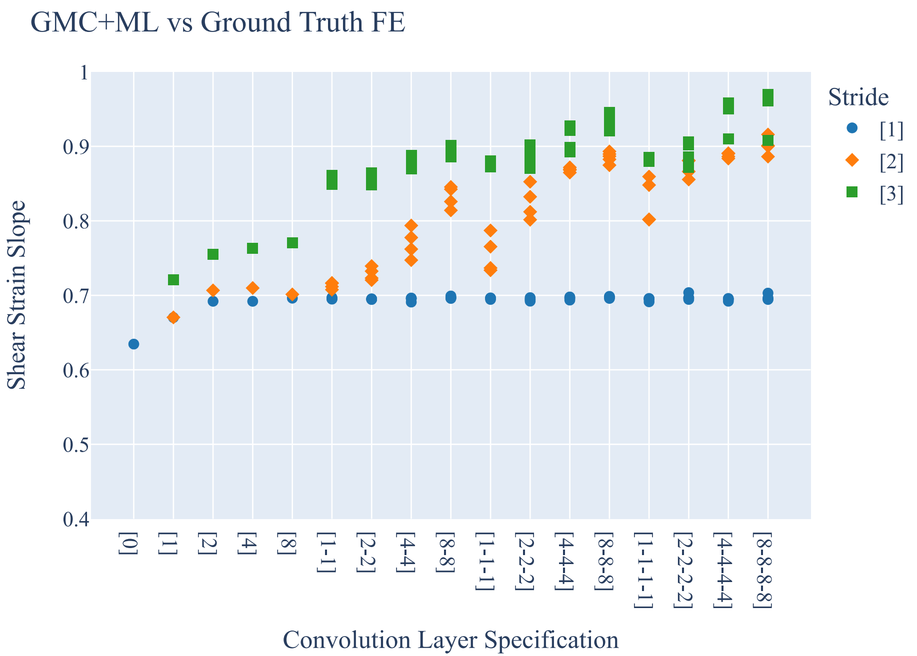




(overfitting a factor?)

ML architecture feature exploration
Layers vs Slope (**Accuracy**)
(Up is more accurate, right is more complex)

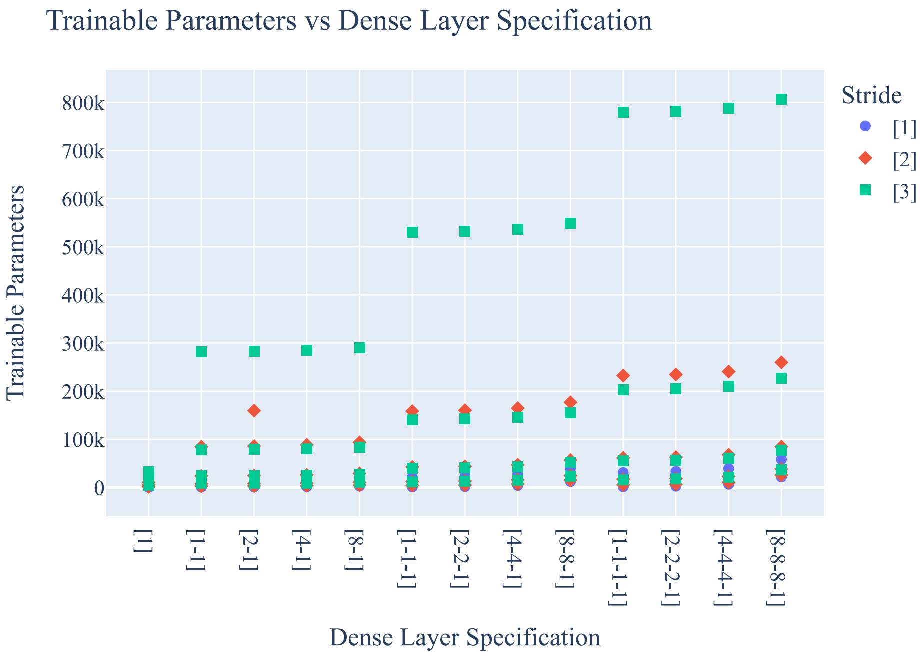


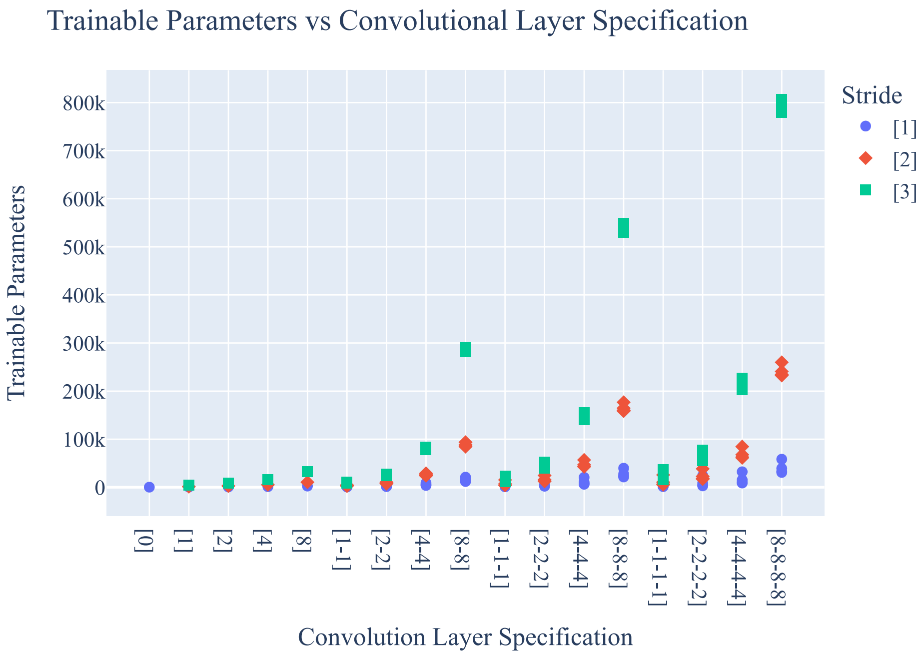


ML architecture feature exploration

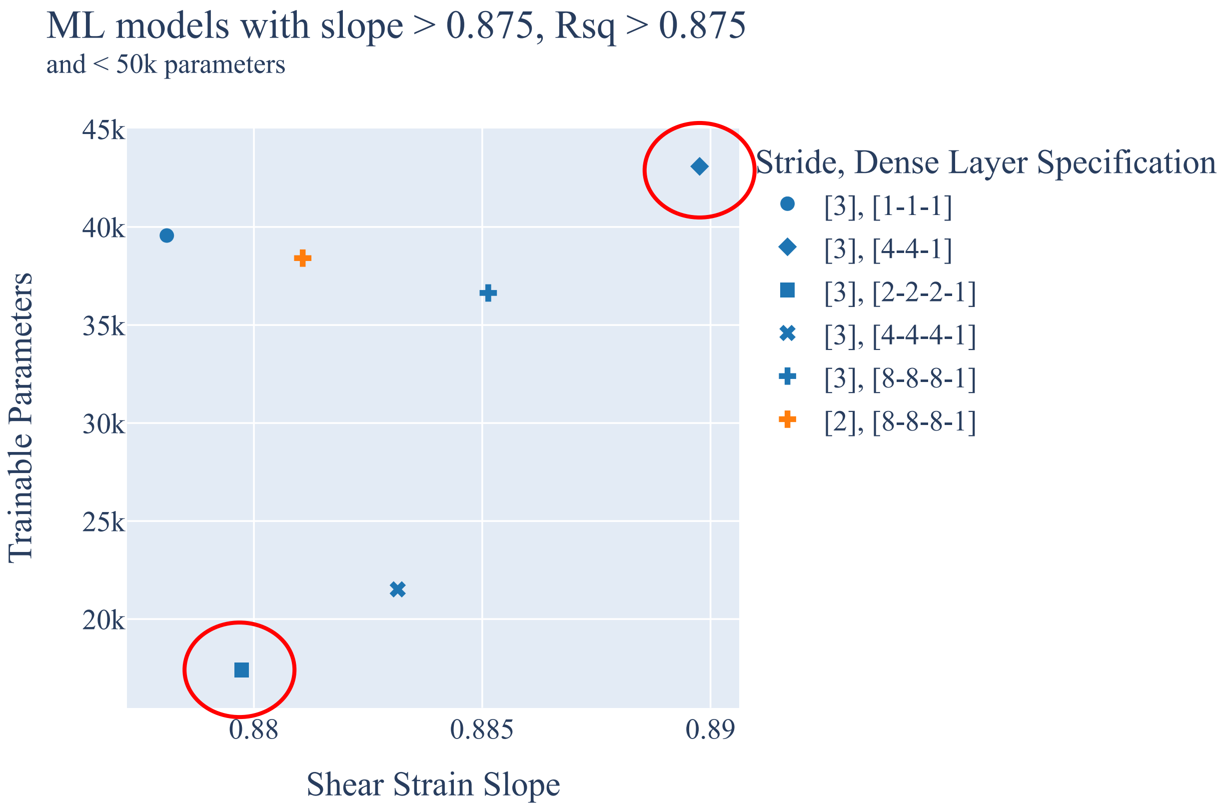
Layers vs **Cost** (Trainable Parameters)
(Up is more costly, right is more complexity of layers)

Cost of training and use





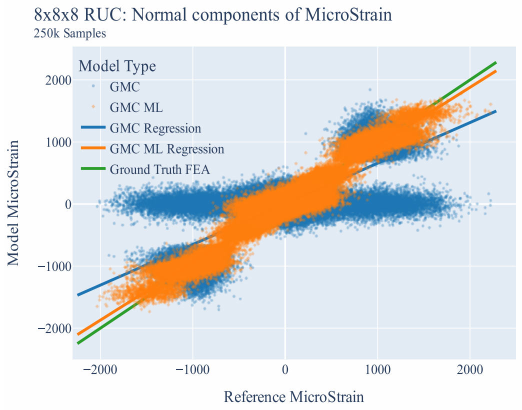
Fast and accurate models

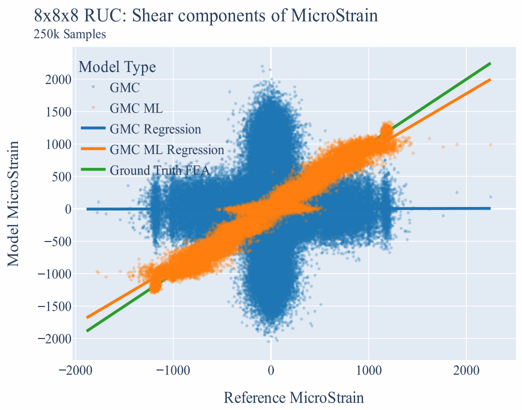


Fast Model: 4 Convolutional Layers [1-1-1-1], 4 dense layers [2-2-2-1], and stride=3 (17,412 parameter model)

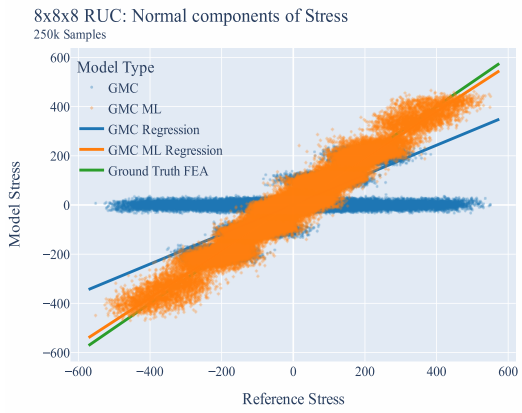
Normal | Shear

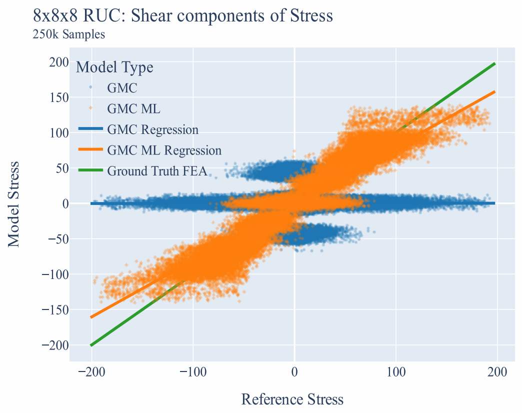
Strain





Stress



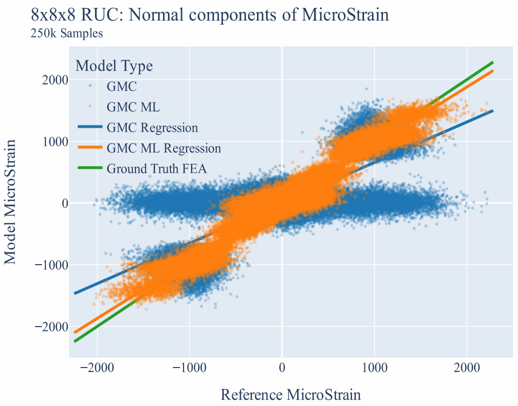


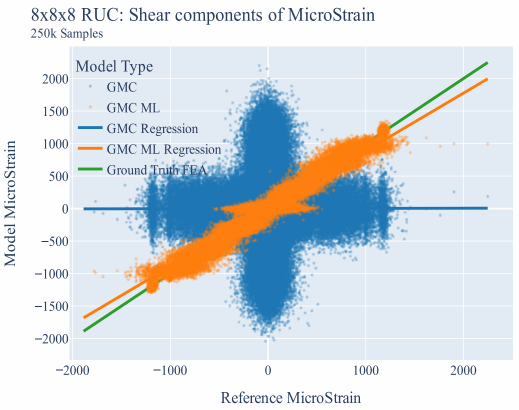
Fast Model: 3 Convolutional Layers [2-2-2], 3 dense layers [4-4-1], and stride=3 (43,092 parameter model)

Normal

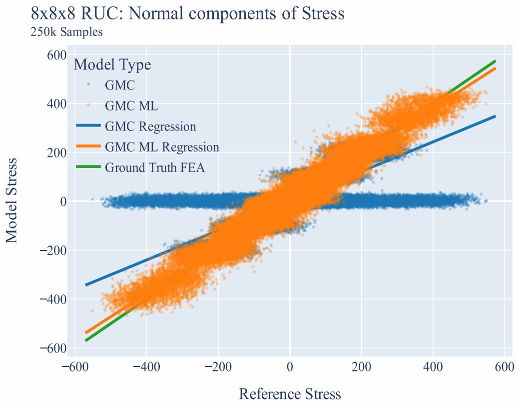
Shear

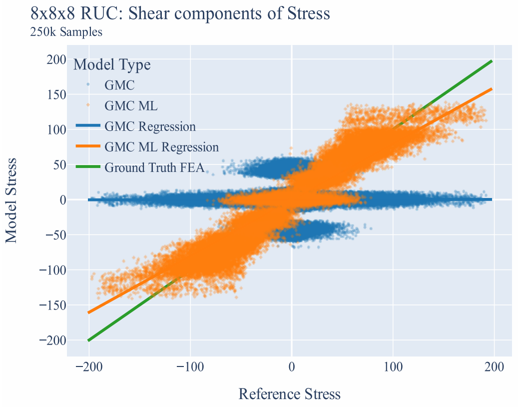
Strain





Stress



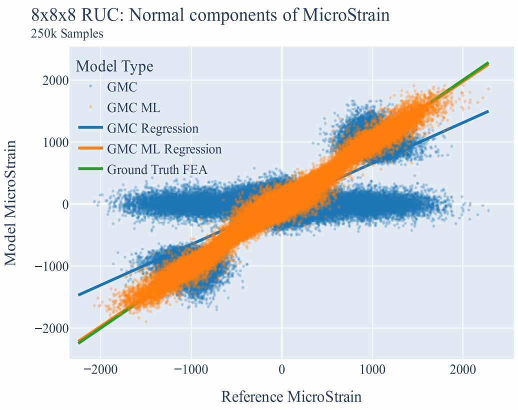


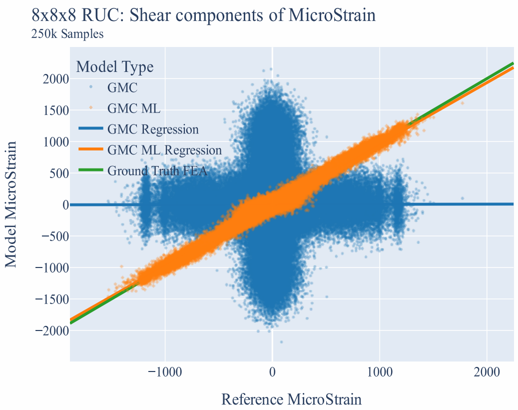
Best Shear Strain Matching: 4 Convolutional Layers [8-8-8-8], 4 dense layers [4-4-4-1], and stride=3 (787,932 parameter model)

Normal

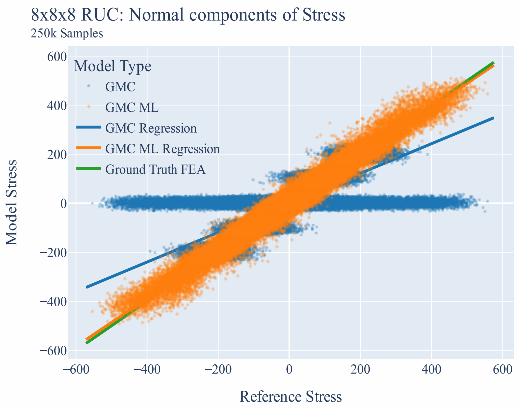
Shear

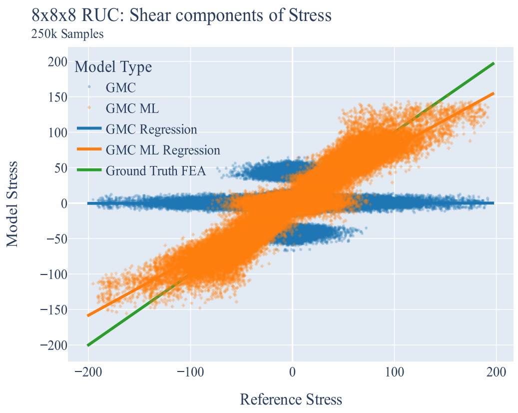
Strain





Stress



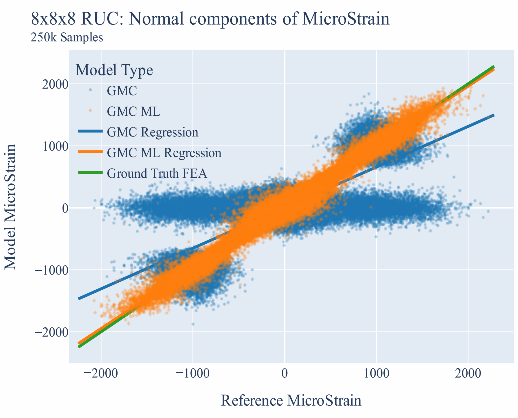


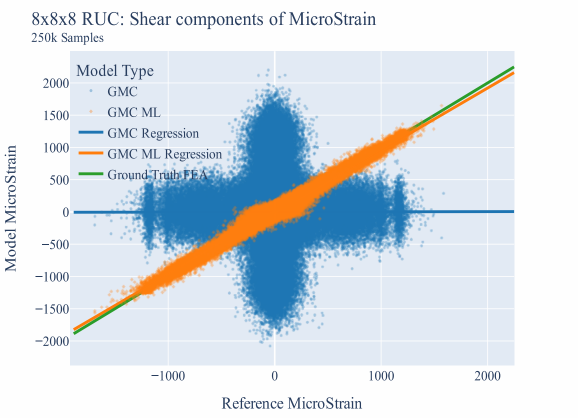
Best Pearson Correlation Coef (r2) for Shear Strain: Predictions with 4 Convolutional Layers [8-8-8-8], 4 dense layers [8-8-8-1], and stride=3 (807,084 parameter model)

Normal

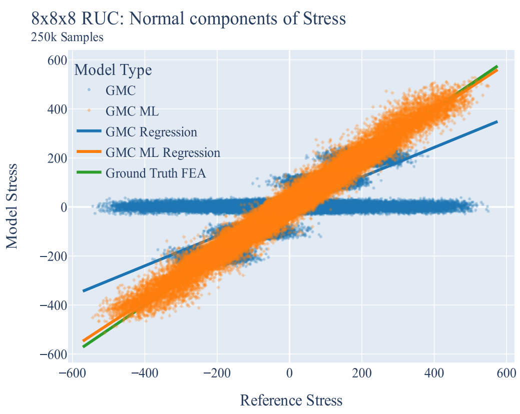
Shear

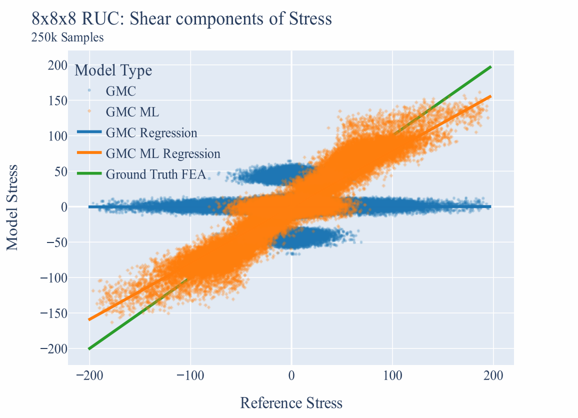
Strain





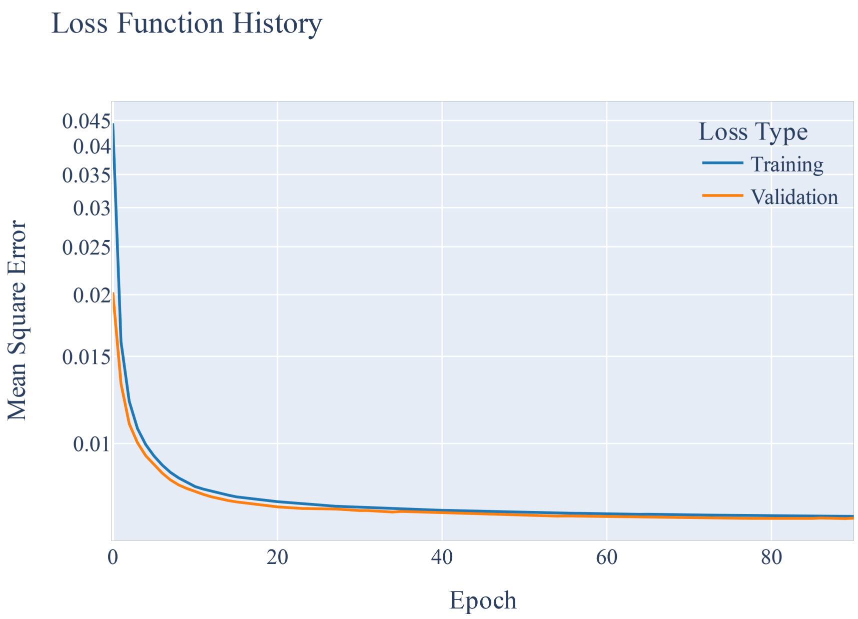
Stress

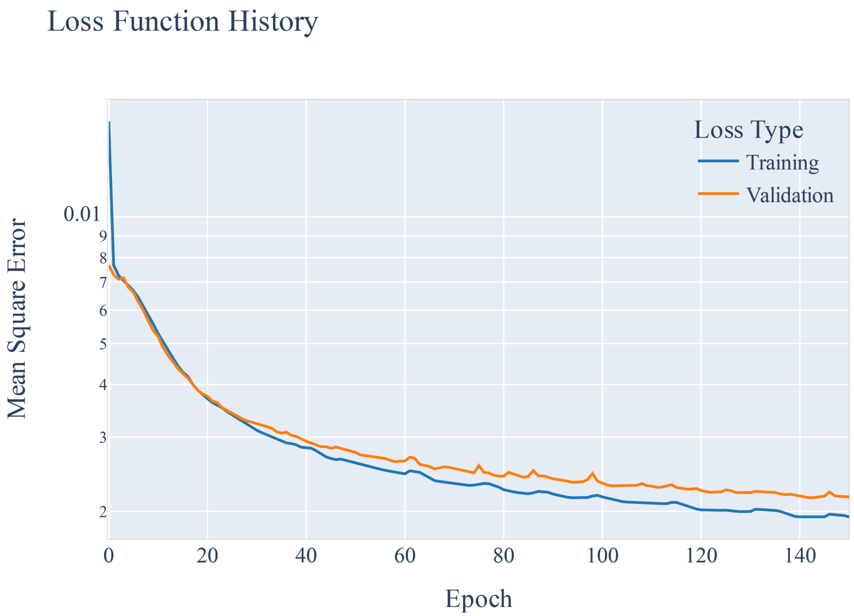




Representative Fast vs Best Loss functions

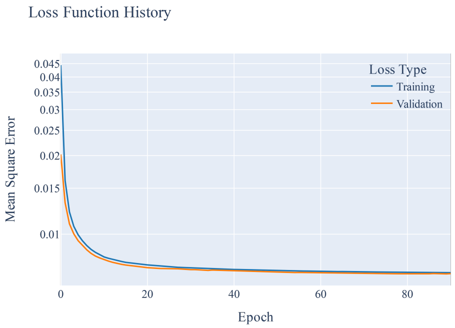
- Successful training exhibits a decreasing loss function for both training and validation loss as training proceeds.

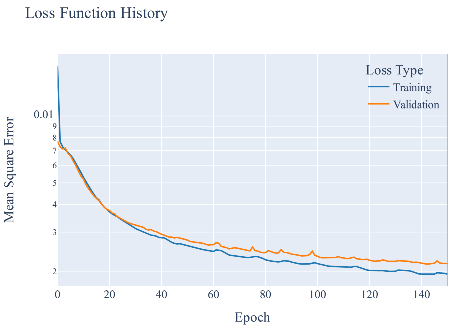




Note scale is different!

Overfit and underfit?





- The fast models:
  - Training and validation curves decrease and converge asymptotically with validation loss near training loss
  - **Thus: accurate, well behaving, and do not exhibit overfitting.**
- Large parameter models (i.e., the best slope and best r^2)
  - Validation loss is somewhat larger than the training loss after approximately 20 epochs.
    - Validation loss still decreasing
    - Suggests more training necessary
    - **Overfitting may be a concern**

Speed and computational cost

Method	Calculation Time (milliseconds)
GMC: Single CPU, Single Load (Optimized MAC/GMC)	**27.00**
GMC: Single CPU, Single Load (Unoptimized NASMAT)	603.00
CalculiX: Single CPU, Single Load (Hex Mesh)	374.24
CalculiX: Single CPU, Single Load (Tet Mesh)	708.81

Speed and computational cost

Method	Convolutional Layer Spec	Dense Layer Spec	Stride	Calculation Time (milliseconds)	Correction Cost %
ML Correction: Parallelized corrections, GPU	[1-1-1-1]	[2-2-2-1]	3	1.84	6.8%
ML Correction: Parallelized corrections, GPU	[2-2-2]	[4-4-1]	3	1.40	5.2%
ML Correction: Parallelized corrections, GPU	[8-8-8-8]	[4-4-4-1]	3	10.58	39.2%
ML Correction: Parallelized corrections, GPU	[8-8-8-8]	[8-8-8-1]	3	11.66	43.2%

Limitations

- Surrogates are trained only on randomly generated metallic microstructures
  - One randomly oriented constitutive description and one grain size
  - Likely can be generalized
- Timing is apples to oranges
  - Call overhead not well accounted
  - Parallelization of GMC in NASMAT not accounted

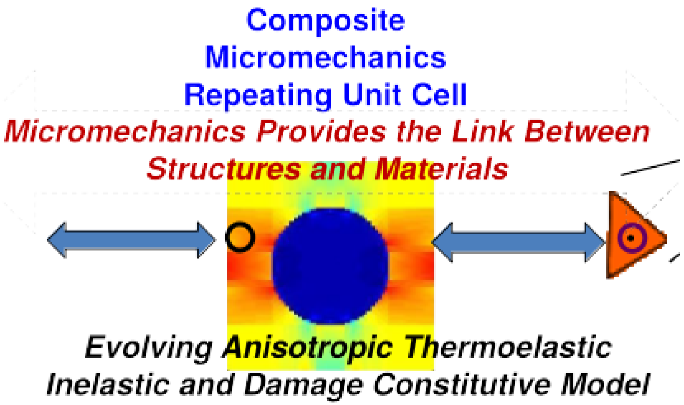
- Despite the limitations noted
  - Demonstrated **fidelity correction** using machine learning.
  - Addresses one of the *most significant criticisms* of the generalized method of cells
    - i.e., addresses lack of coupling of normal and shear through shear lag

Conclusions

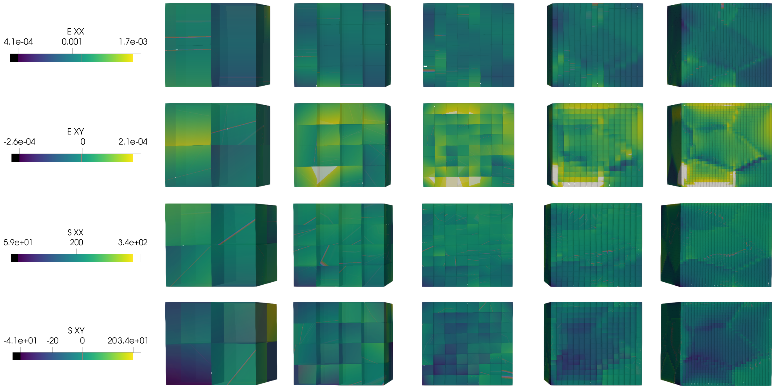
- Strains/Stress were enhanced using machine learning and FE ground truth strain/stress
- GMC+ML Shear improved from inaccurate and uncorrelated (slope=0.003, r^2=0.000) to accurate and well correlated (slope=0.890, r^2=0.882) relative to ground truth.
  - (Ideal slope =1.0, r^2 = 1.0).
- Shear stresses yielded a similar accurate and well correlated result (slope=0.782, r^2=0.800).
- Enhancement was relatively fast, estimated to be a small fraction of computational cost

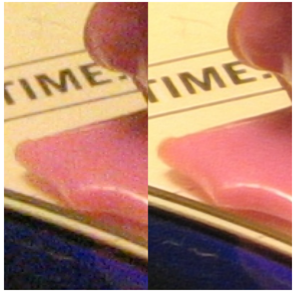
Next Steps (Part 1)

- Examine fibrous composites
  - large differences in local moduli, multiple materials
- Superesolution techniques
  - Efficient prediction of detailed models



Superresolution Enhancement

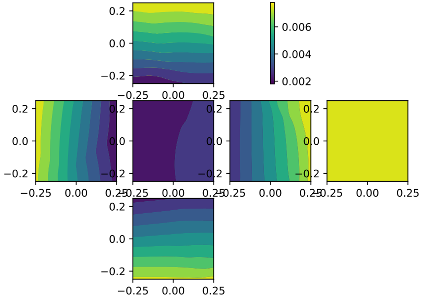




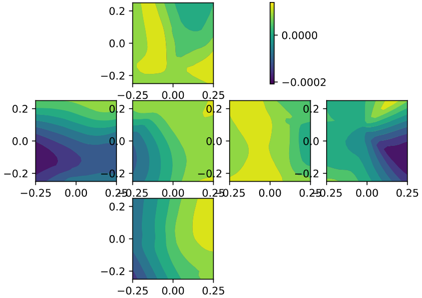
Superresolution Enhancement

Next Steps (Part 2)

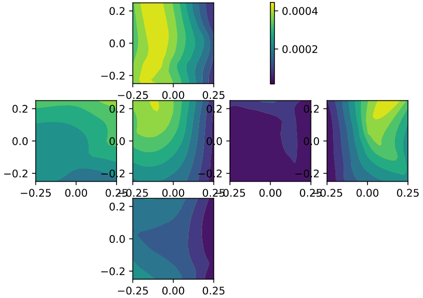
- Effect of boundary conditions
  - Assumption of periodic BCs inconsistent with assumed RUC internal asymmetries
  - Not critical for homogenization, critical for localization



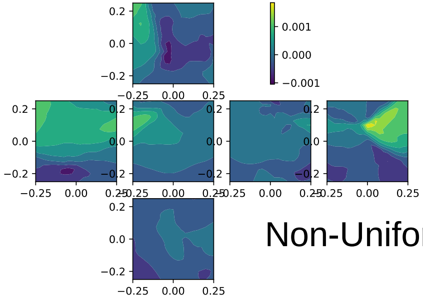
Uniform BC
Displacement (Z)



Non-Uniform BC
Displacement (X)



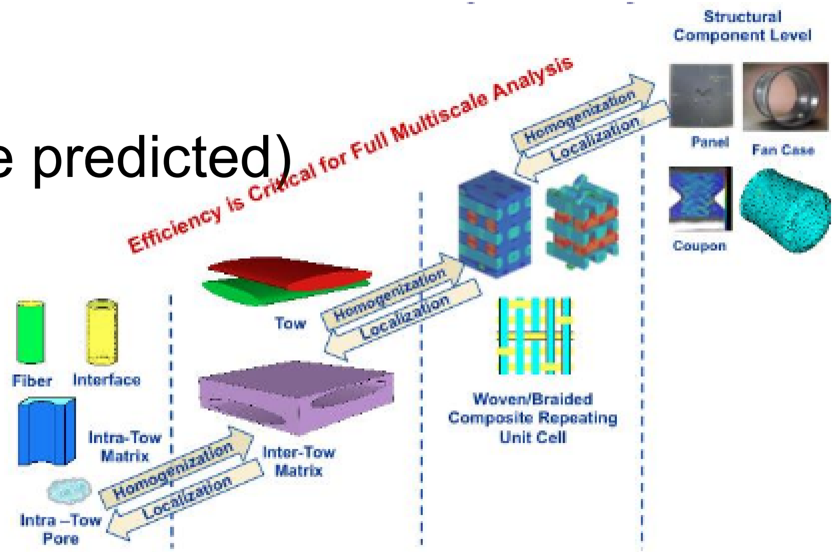
Non-Uniform BC
Displacement (Y)



Non-Uniform Strain (Exx)

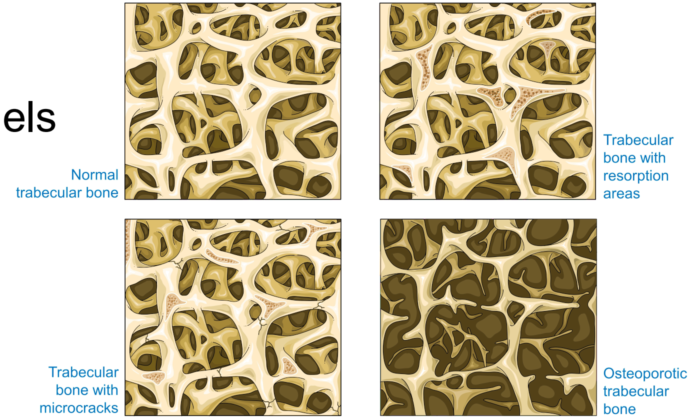
Next Steps (Part 3)

- Multi-scale outcomes
  - Quantify how improved strain/stress predictions cascade through scales
  - Localization is "uniform"
    - (but must also accurately be predicted)

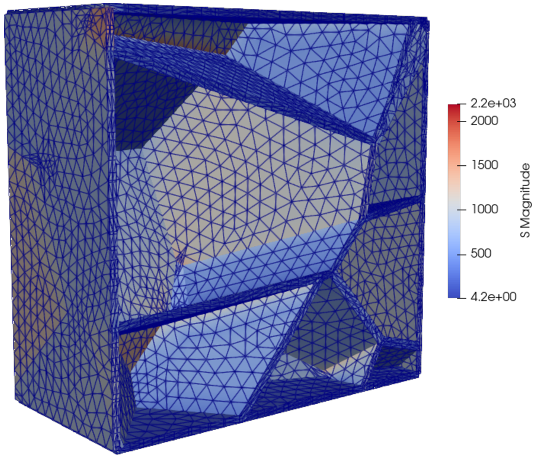


Next Steps (Part 4)

- Damage mechanics, LSTM models
- Localized MLs
  - Train on localized subscales… use for filtering
  - Only calculate lower scales where non-linear damage occurs
- Repeat
  - CMCs
  - thermomechanical models
  - Bone and soft tissue



Laboratoires Servier (Wikipedia)



Acknowledgments

- This work was funded via:
  - a NASA summer faculty fellowship
  - a NASA visiting scholar fellowship.
- **NASA funding was provided by the Transformational Tools and Technologies (TTT) project, under the Aerospace Research Mission Directorate.**
- Additional funding from:
  - The **Western Michigan University Sabbatical Program**
- These funding sources are gratefully acknowledged

- Questions?
  - peter.gustafson@wmich.edu