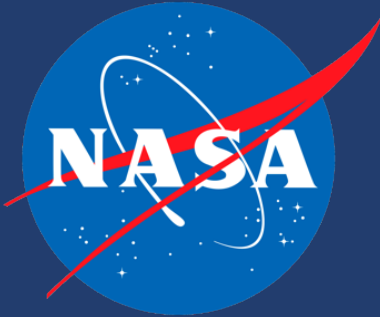


# PuMA and Multiscale Modeling

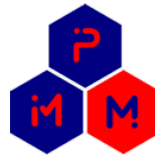
Federico Semeraro

Thermal Protection Materials Branch (ARC-TSM)



NASA Ames Research Center Moffett Field, CA

Predictive Material Modeling (PMM) group



Entry System Modeling (ESM) project



# Content of presentation

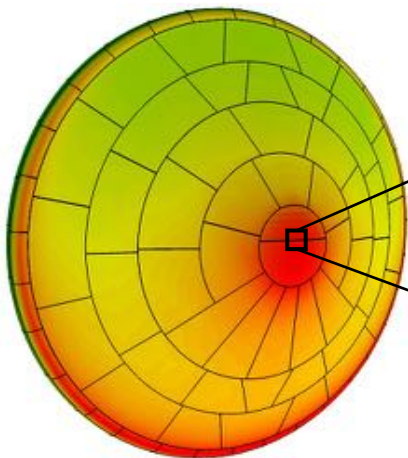
- Motivation
- Overview of PuMA
  - Functionalities
  - Advanced domain generation
  - Material response examples
- Effective properties for fibrous media
  - Fiber orientation
  - Thermal conductivity
  - Stress analysis



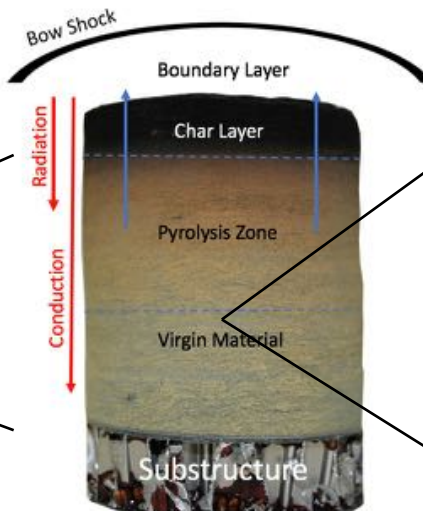
# Modeling TPS

## Macroscale

Full-scale material response solvers, using homogenized properties



Simulation of surface temperature for MSL heatshield



## Microscale

Homogenization of material properties to be used in macroscale models





# X-ray microtomography

Collect X-ray images of the sample as it rotates through  $180^\circ$



Use the series of images to reconstruct the 3D object



Courtesy of D. Parkinson  
(ALS at Berkeley National Lab)

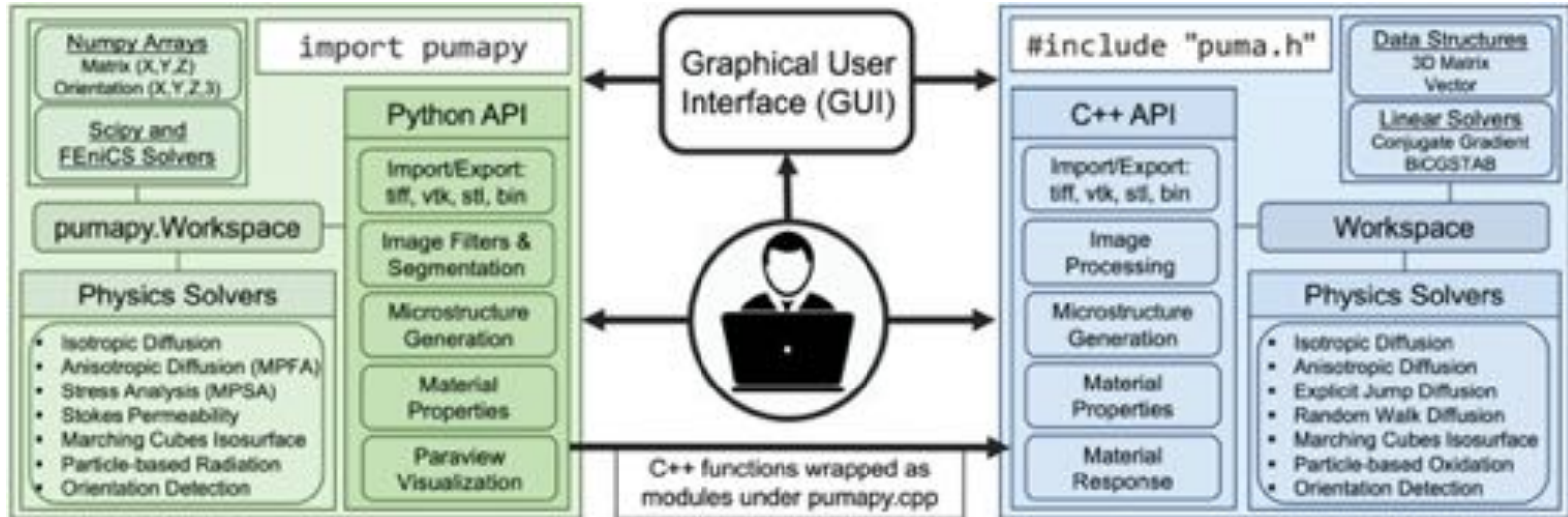


# Porous Microstructure Analysis (PuMA)



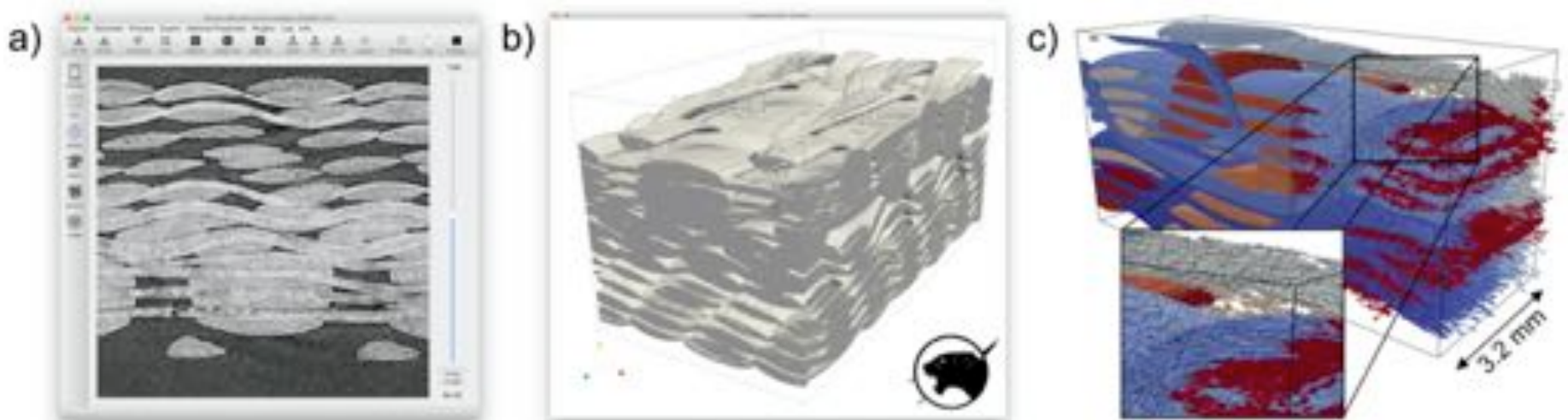
Git repository: <https://github.com/nasa/puma>

Documentation and tutorials: <https://puma-nasa.readthedocs.io>



# Improved processing of micro-CT weaves

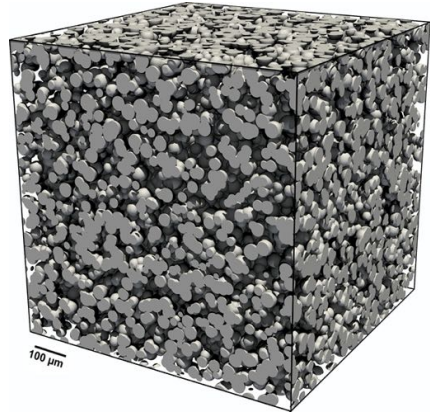
- a) Adept 12 ply imported into PuMA GUI
- b) 3D rendering of microstructure
- c) Post-processed weave after segmentation and orientation detection



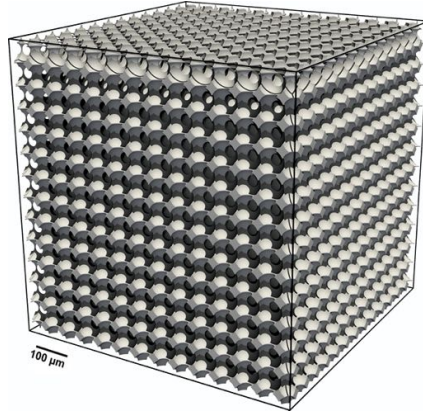


# Advanced domain generation

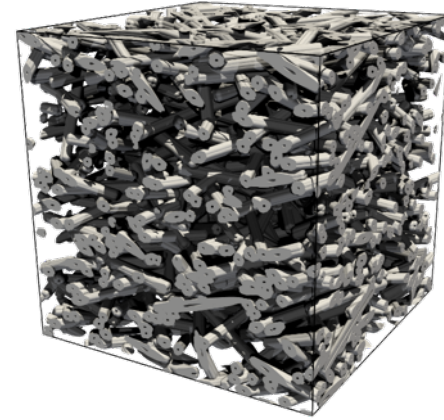
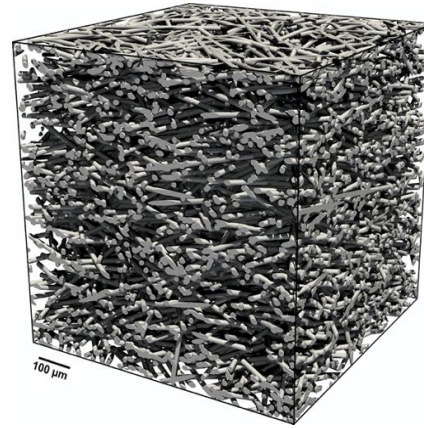
Packed Sphere Beds



Periodic Foams



Fiber Structures



In this tutorial we demonstrate the creation of weaves using TexGen and how to import them into PuMApy. TexGen python scripts can be simply obtained by either following TexGen tutorials <https://github.com/foalupts/TexGenScriptingGuide> or by recording the actions in the TexGen GUI (of which there are already bundled executables for Windows at <https://sourceforge.net/projects/texgen/files/>).

Let's start by importing PuMApy, Numpy and TexGen and creating a simple LayerToLayer weave using a TexGen python script.

```

1: import os
2: import sys
3: import numpy as np
4: from TexGen.core import *

5: # Number of layers = 2
6: numLayers = 2
7: numFibers = 5
8: numFibers = 4
9: KSpacing = 5.0
10: KSpacing = 5.0
11: KSpacing = 5.0
12: height = 0.2
13: height = 0.2
14: height = 0.2
15: weave = CreateLayerToLayer(numFibers, numFibers, KSpacing, KSpacing)

16: # Set number of binder / warp yarns
17: numBinderYarns = 2
18: numWarpYarns = numFibers - numBinderYarns
19: weave.SetWarpSpatial(numWarpYarns)
20: weave.SetBinderSpatial(numBinderYarns)

21: # Setup layers: 2 warp, 4 weft
22: weave.SetLayers(2, 4, numBinderYarns)

23: # Set yarn diameters: width / height
24: weave.SetFiberSpatial(0.8, 0.8)
25: weave.SetFiberSpatial(0.8, 0.8)
26: weave.SetBinderSpatial(0.8, 0.8)
27: weave.SetBinderSpatial(0.8, 0.8)

28: # Define offsets for the two binder yarns
29: P = [(0, 1, 2, 3), (0, 1, 2, 3)]

30: # Assign the z-positions to the binder yarns
31: for y in range(numWarpYarns, numFibers): # Loop through number of binder
32:     offset = 0
33:     for x in range(numFibers): # Loop through the node positions
34:         weave.SetBinderPosition(x, y, P[y-numWarpYarns][offset])
35:         offset = offset+1

```

Next we create the domain, i.e. the lengths of the box containing the weave. In this case we will let TexGen figure it out automatically using the AssignDefaultDomain function:

```

1: weave.AssignDefaultDomain()
2: domain = weave.GetDefaultDomain()

```

Then, we pass the weave and the domain to the PuMApy exporter, which creates the voxel grid and exports it to a directory (expensive step ~5-2 min):

```

1: puma.export_weave_voxelgrid = "tests/out/newweave", weave, domain

```

This can be then read back into a pumapyWorkspace:

```

1: w = puma.Import_weave_voxelgrid = "tests/out/newweave"

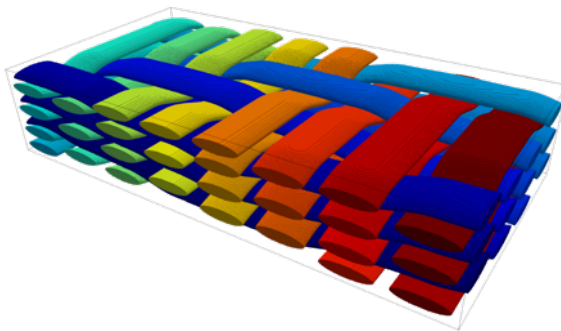
```

# Artificial weave generator

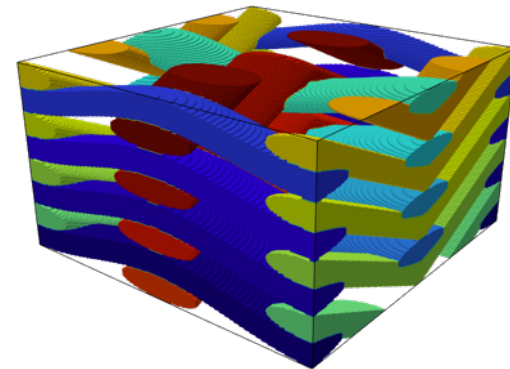


Pumapy was coupled with TexGen's Python API to provide flexible synthetic weave generation.

Wrappers to convert from a TexGen geometry to a pumapy.Workspace were also implemented



Layer-to-layer weave

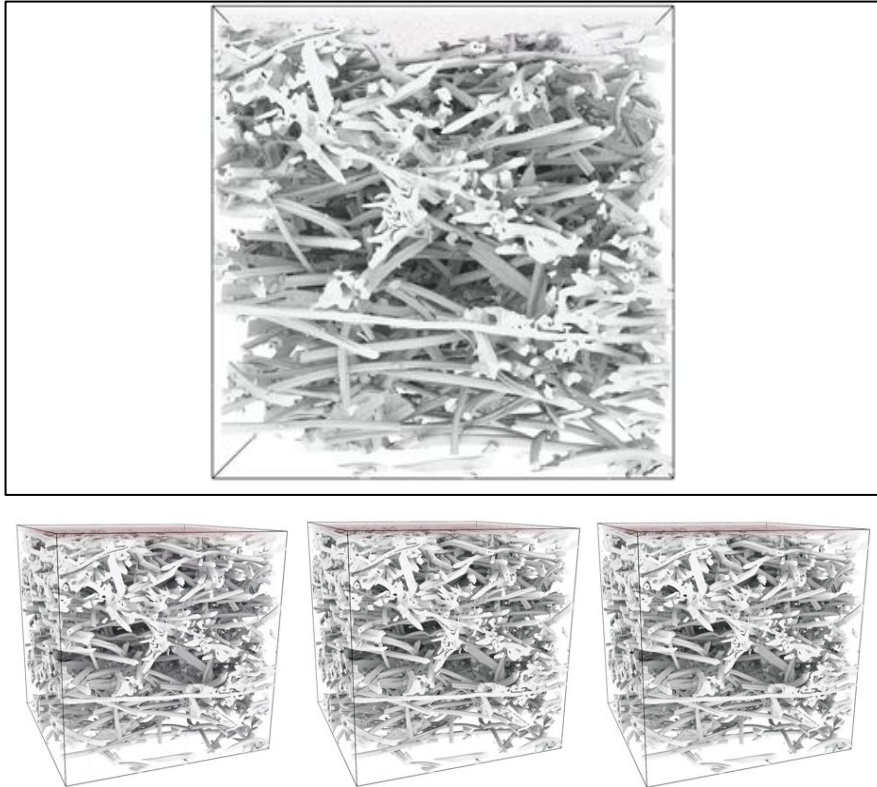


3MDCP geometry

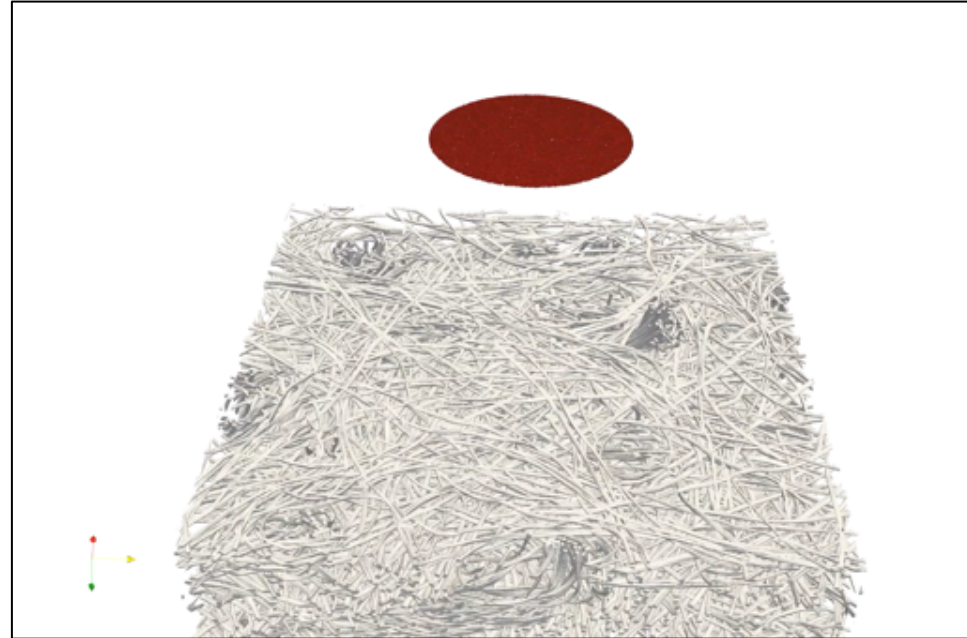


# Material response examples

Oxidation simulation

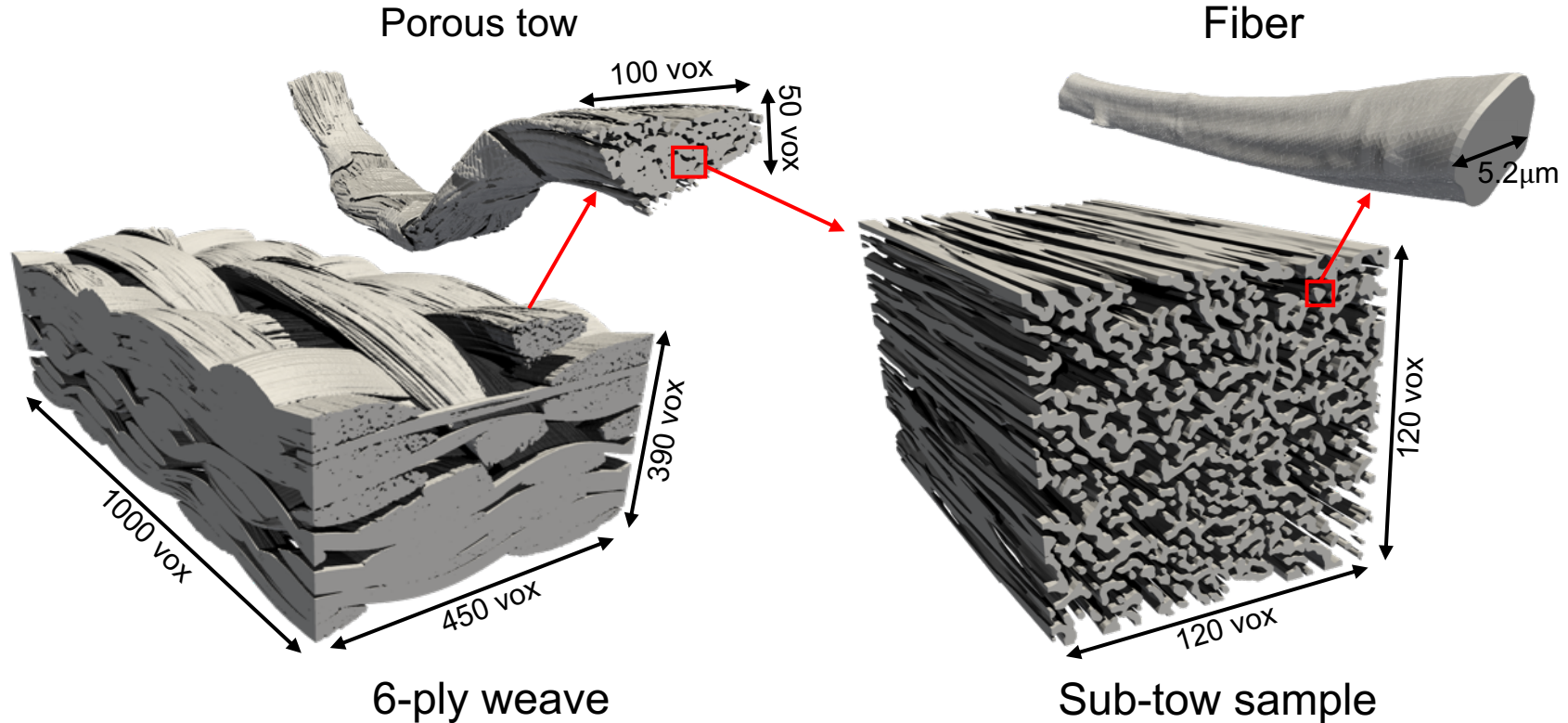


Hyperthermal beam



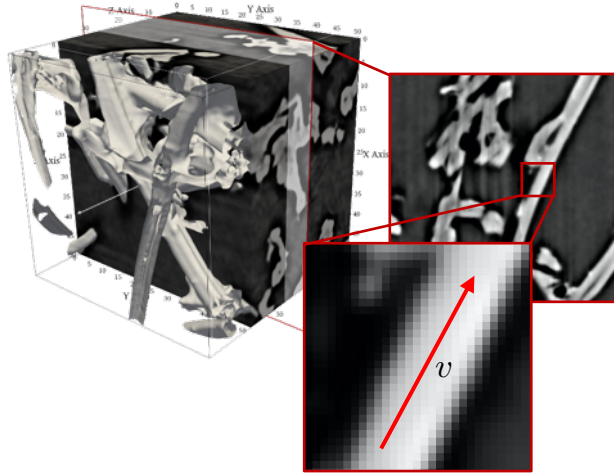


# An anisotropic and multiscale problem



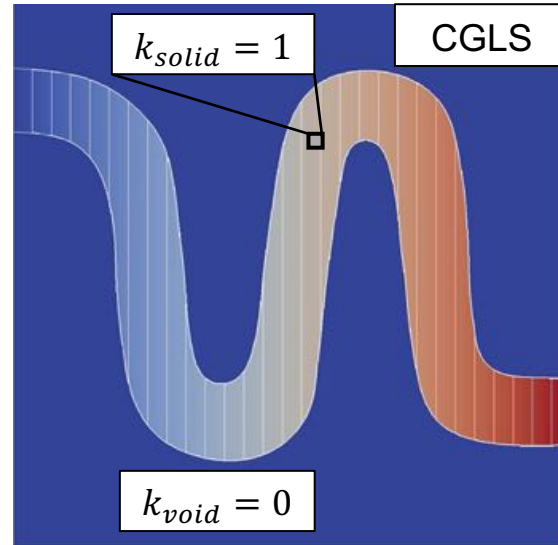
# Fiber orientation methods

Structure tensor

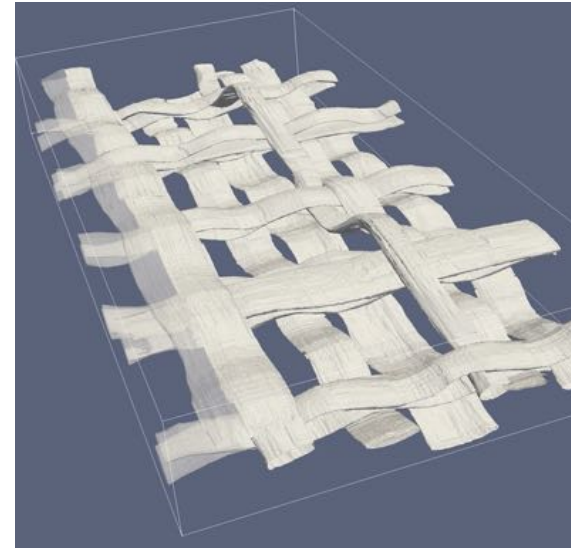


$$(I(x + v) - I(x))^2 \approx 0$$

Artificial flux



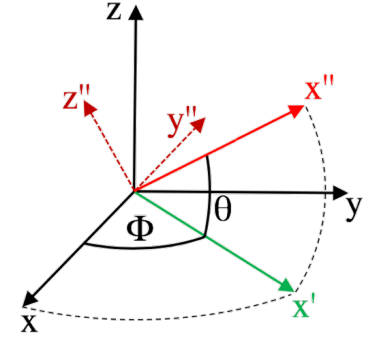
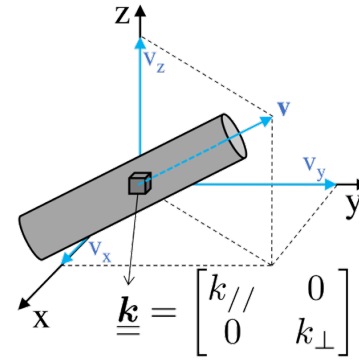
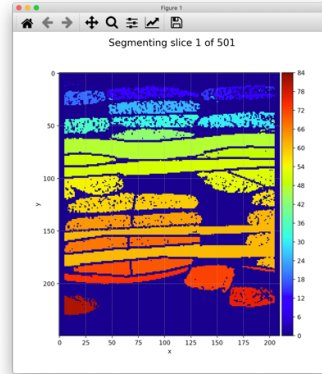
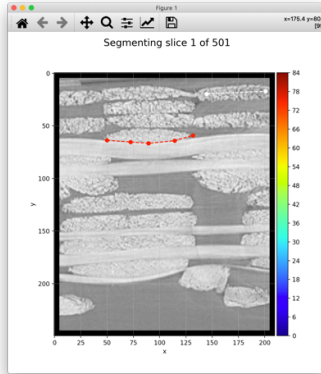
Ray casting





# Micro-CT weave post-processing

Individual fiber/tow panoptic segmentation and tracking

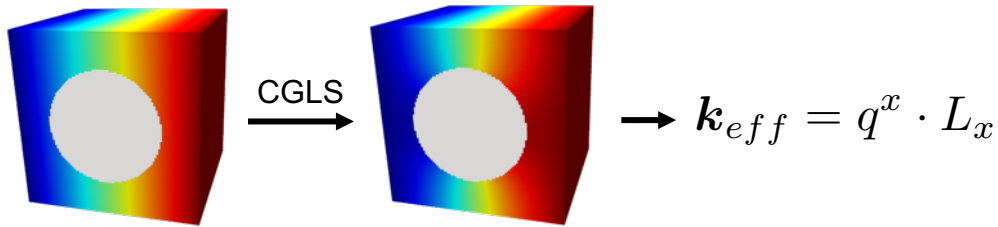


$$\mathbf{q} = \underbrace{[\mathbf{R}^{-1} \mathbf{k}'' \mathbf{R}]}_{\mathbf{k}} \nabla T$$

$$\mathbf{R} = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



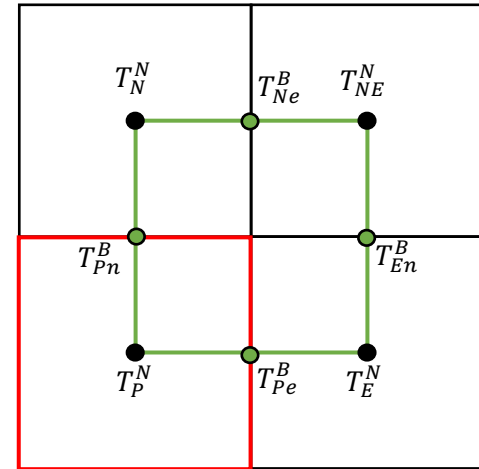
# Numerical method for solid conduction



$$\nabla \cdot \mathbf{q} = 0 \rightarrow \mathbf{q}(x, t) = \mathbf{E}(x) \mathbf{T}^N(x, t)$$

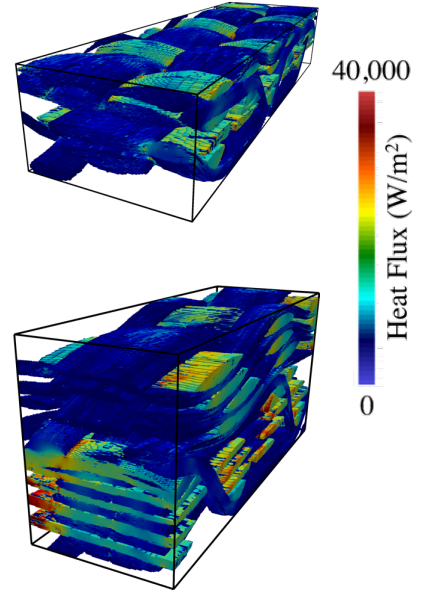
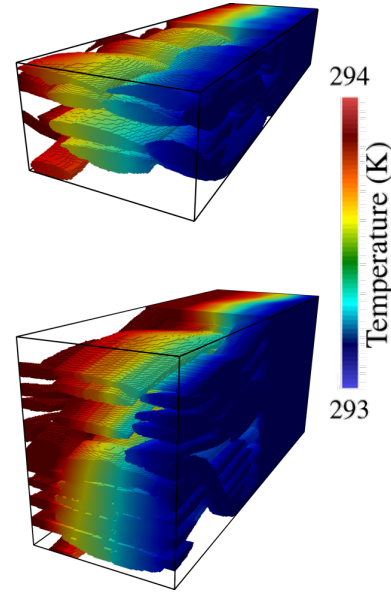
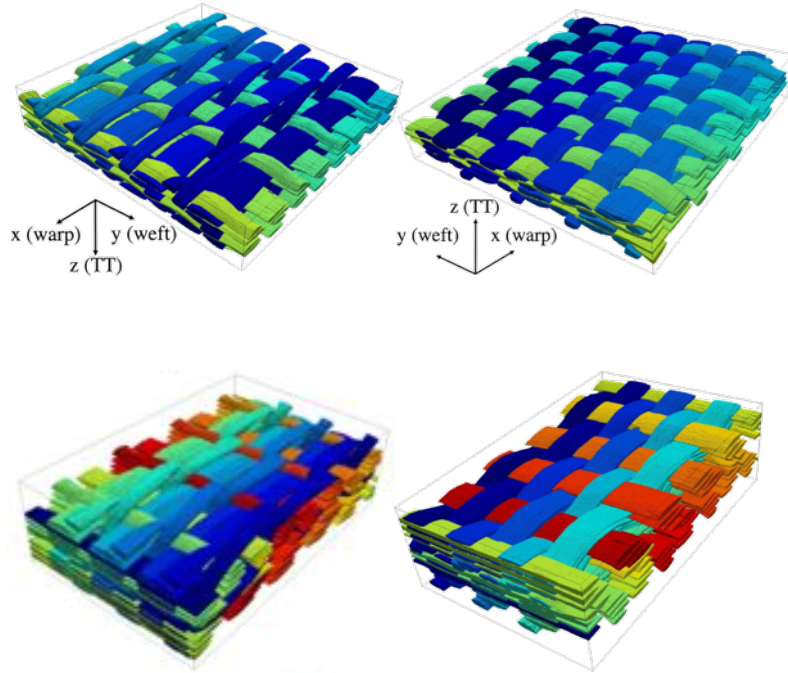
Multi-Point Flux Approximation (MPFA-O)\*  
i.e. continuity of fluxes and temperatures at Boundary points:

$$\begin{aligned} \mathbf{q}_P^{xx} &= \mathbf{q}_E^{xx} \\ \mathbf{q}_P^{yy} &= \mathbf{q}_N^{yy} \\ &\vdots \end{aligned}$$





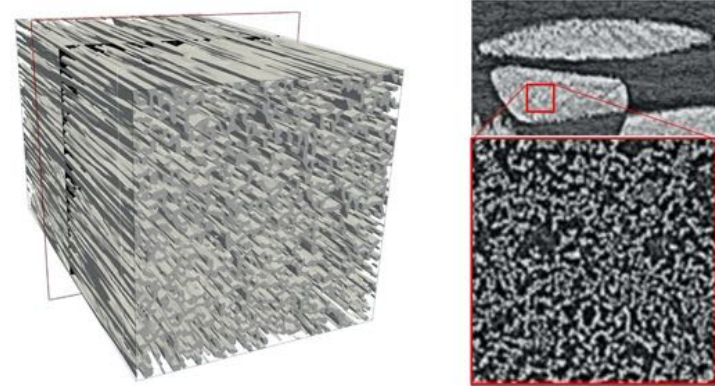
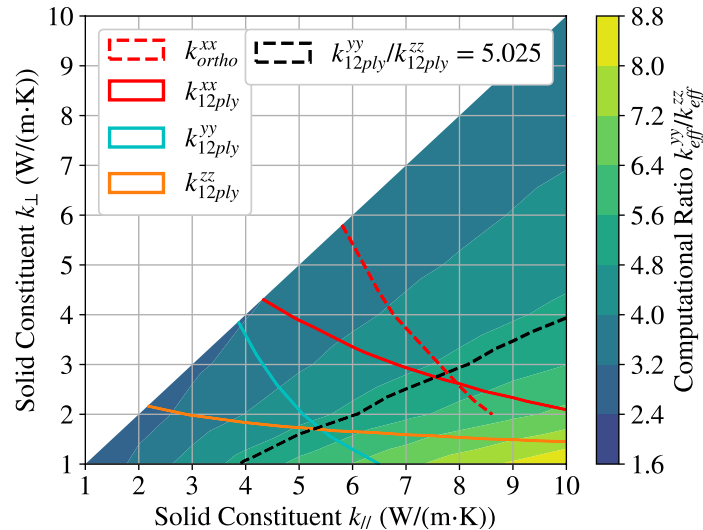
# Conductivity solver validation: ADEPT



# Single fiber conductivity estimation

Experimental value at room temperature:

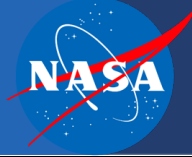
$$\mathbf{k}_{exp}^{12ply} = \begin{bmatrix} 2.184 & - & - \\ - & 1.980 & - \\ - & - & 0.394 \end{bmatrix}$$



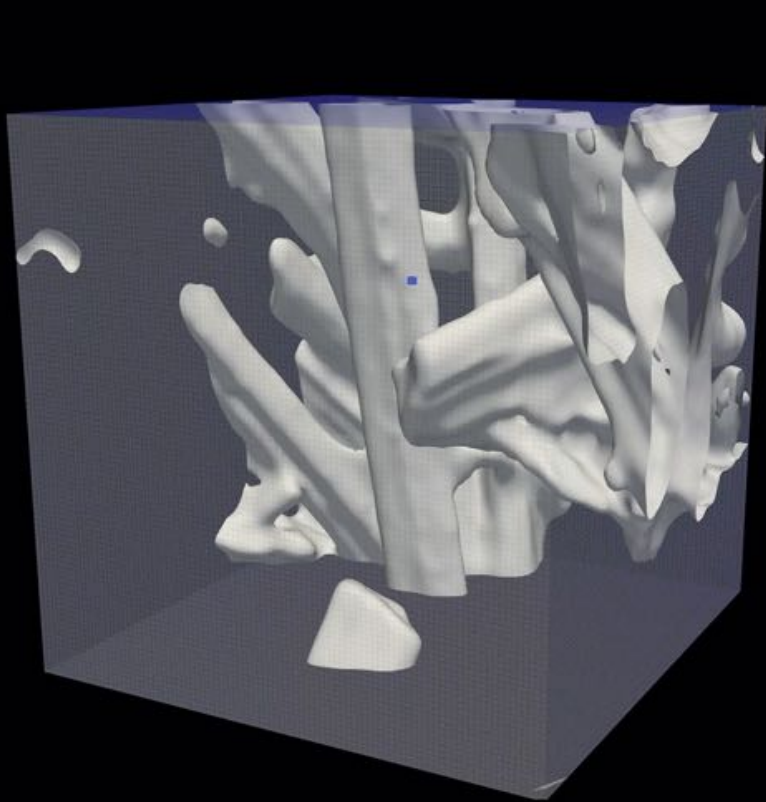
Single fiber thermal conductivity

$$[k_{//}, k_{\perp}] = [9.7, 5.5] \frac{\text{W}}{\text{mK}}$$

$$\mathbf{k}_{num}^{12ply} = \begin{bmatrix} 2.310 & -0.414 & 0.000 \\ -0.524 & 2.030 & 0.071 \\ 0.007 & 0.050 & 0.504 \end{bmatrix}$$

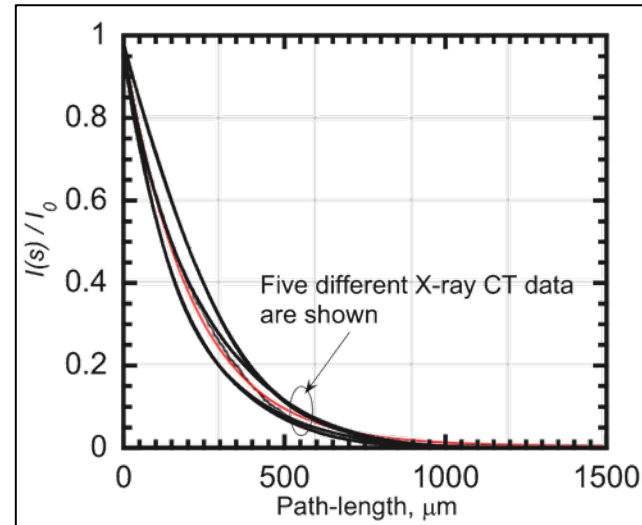


# Radiative conduction



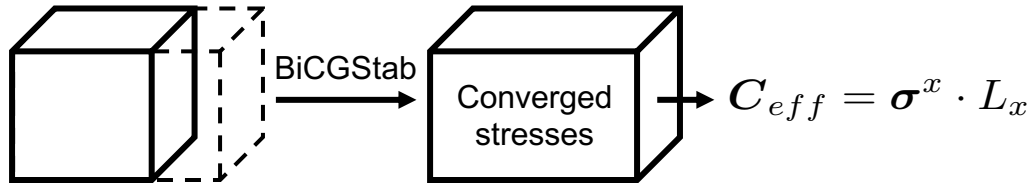
$$k_{\text{eff}} = k_{\text{sol}}(T) + k_{\text{rad}}(T) = k_{\text{sol}}(T) + \frac{16\sigma T^3}{3\beta}$$

$$\frac{I(s)}{I_0} = \exp(-\beta s) = 1 - \int_0^s F(s)ds.$$





# Numerical method for stress analysis

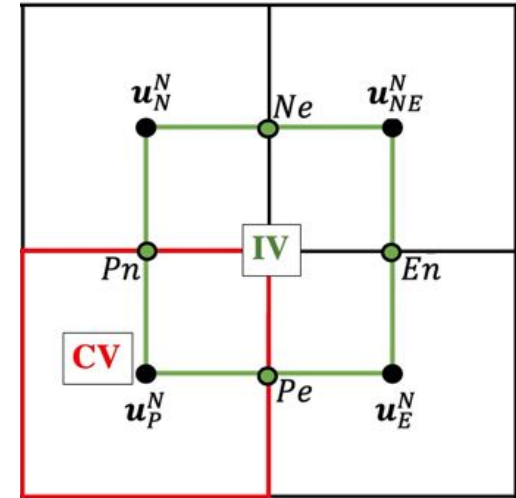


$$\nabla \cdot \sigma = \nabla \cdot (C \varepsilon) = 0 \rightarrow \sigma(x, t) = E(x) D^N(x, t)$$

Multi-Point Stress Approximation (MPSA-W)\*

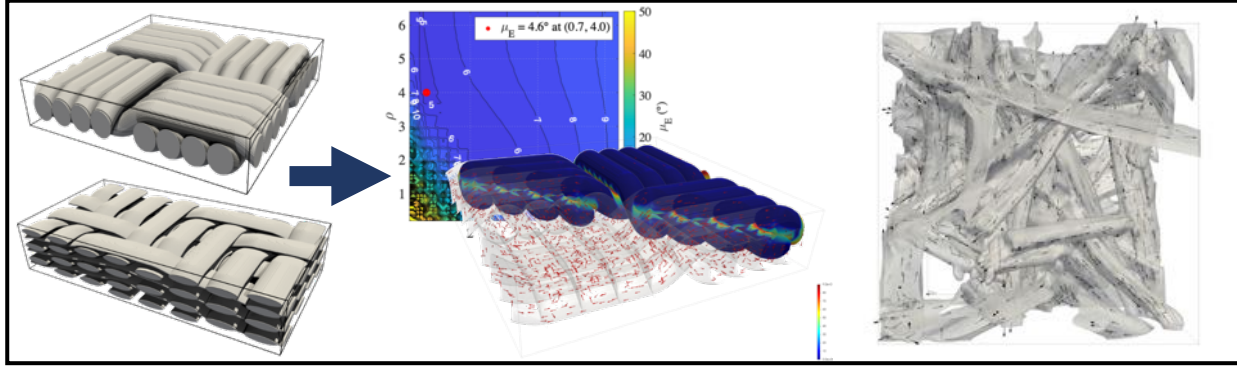
Continuity of Stresses

$$\begin{aligned} \boxed{\sigma_P^{xx}} &= \sigma_E^{xx} \\ \sigma_P^{yx} &= \sigma_E^{yx} \\ &\vdots \end{aligned}$$

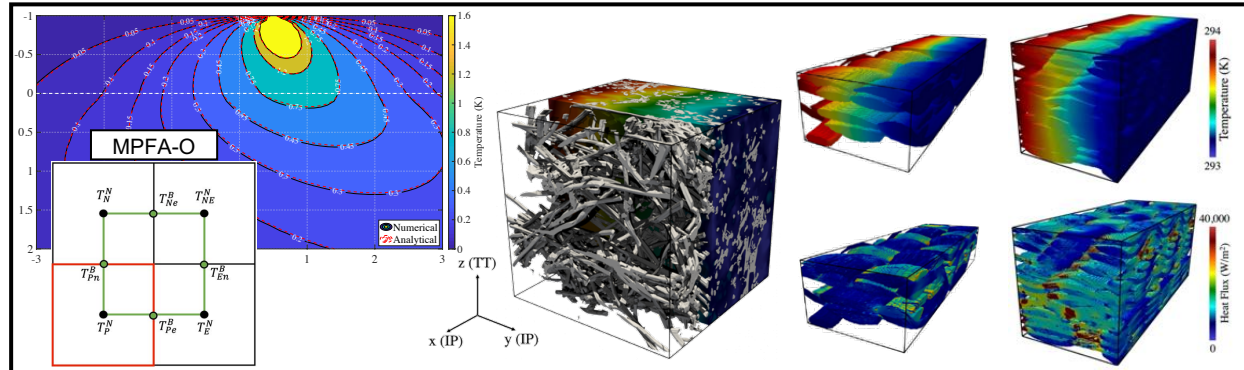


# Conclusion

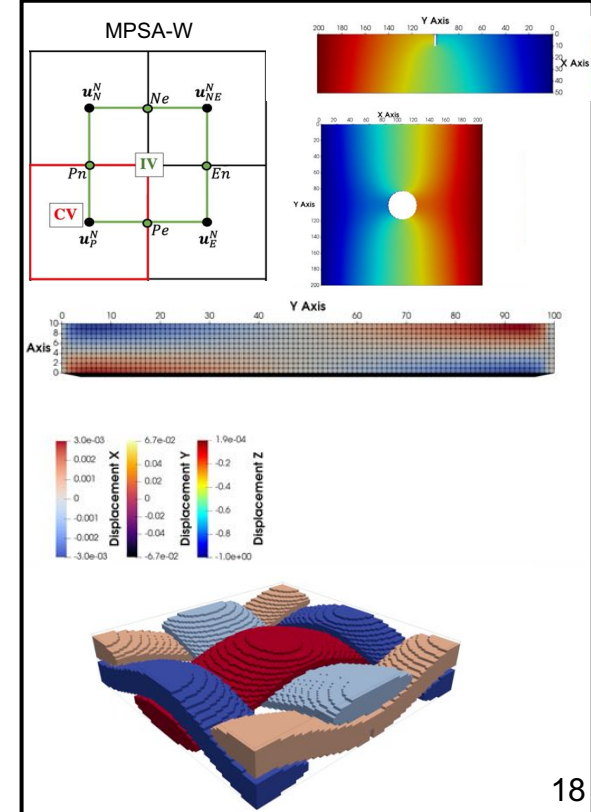
**Part 1: Estimation of local orientation.** *Computational Materials Science*, 178, p.109631. (2020)



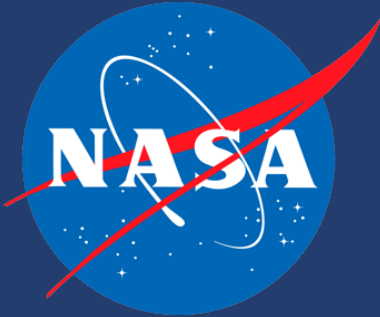
**Part 2: Computation of effective conductivity.** *Computational Materials Science*, 186, p.109956. (2021)



**Part 3: Computation of effective elasticity.**  
(work in progress)



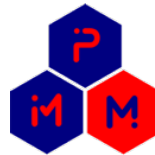
# Thank you. Questions?



NASA Ames Research Center Moffett Field, CA



Predictive Material Modeling (PMM) group



Entry System Modeling (ESM) project