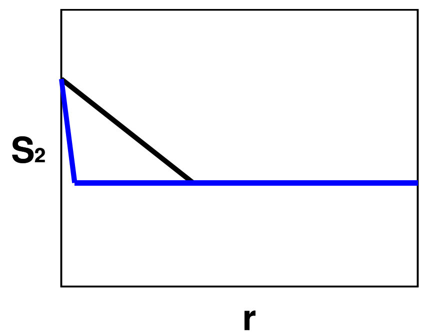
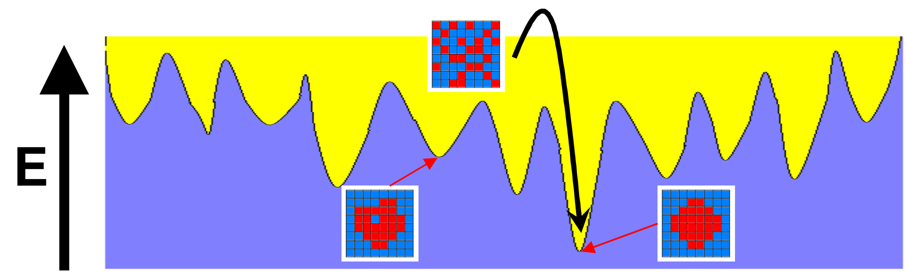
Pixel/Voxel level uncertainty quantification – microstructure reconstruction

Correlation-Function Based Microstructure Reconstruction:

$$E = \sum_{r} \left[S_2^{(i)}(r) - \hat{S}_2^{(i)}(r) \right]^2$$

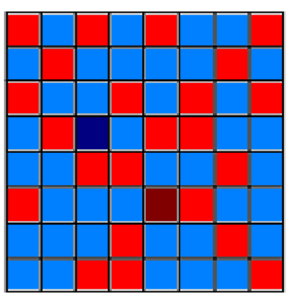


Correlation Function

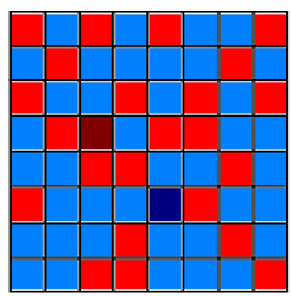


Simulated Annealing Method: Slowly Decreasing the "Temperature"

Evolution of Microstructure: Exchanging Pixels

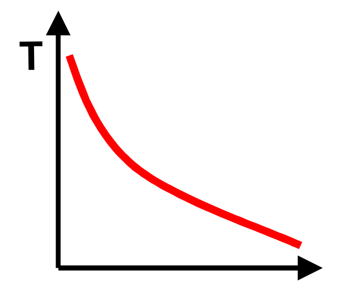


E_{old}



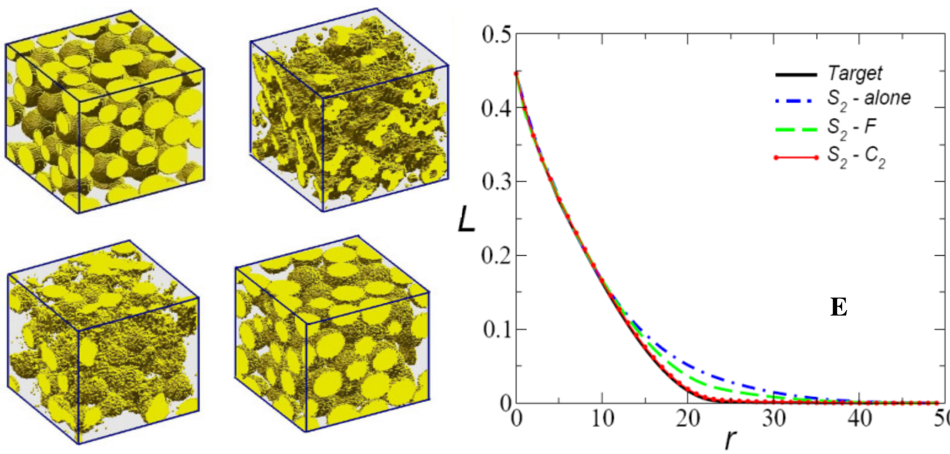
E_{new}

$$P(E_{old} \rightarrow E_{new}) = \begin{cases} 1, & \Delta E < 0 \\ \exp(-\Delta E / T), & \Delta E \geq 0 \end{cases}$$

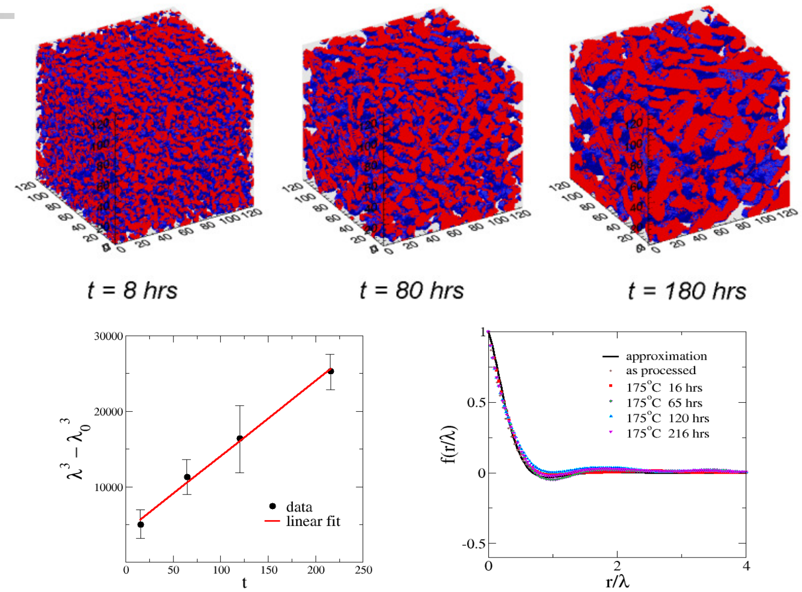


Y. Jiao, F. H. Stillinger & S. Torquato, Phys. Rev. E 76, 031110 (2007); ibid 77, 031135 (2008)

Pixel/Voxel level uncertainty quantification – bi-phase materials



- Other functions, such has cluster function can be used together with correlation function for enhanced morphology mapping
- Same volume fraction with different clustering will change some mechanical behaviors



- Microstructure evolution due to different annealing conditions
- Unified approach for reconstruction of materials with nonstationary microstructure

Pixel/Voxel level probabilistic solver – review

Challenge: "Curse of dimensionality"

- **Sampling methods**
  - Monte-Carlo simulation; MC simulation with sampling
- **Dimension reduction methods**
  - Univariate dimension-reduction; Principle component analysis, sensitivity analysis
- **Surrogate model methods**
  - High-order polynomial model; Kriging model; Support vector machine
- **Analytical methods**
  - FORM, SORM, moment matching

Pixel/Voxel level probabilistic solver – adjoint method

The potential method: FORM

Where does the computational cost come from for FORM?

The finite difference method to calculate the gradient of g($\mathbf{u}$) at $\boldsymbol{u} = (u_1, u_2, \dots, u_d)$

$$\frac{\partial g}{\partial u_i} = \frac{g(\boldsymbol{u} + h_i) - g(\boldsymbol{u})}{h_i}$$

The number of function evaluations is 1+d, which is dimension related.

Can we reduce the computational cost ?

- Adjoint method
- The response and the gradient can be obtained with 2 function evaluations. For special cases, it can be reduced to 1.
- *Independent with the dimension.*

Pixel/Voxel level probabilistic solver – Adjoint Lattice Particle Method (ALPM) 1

Assume a general limit state function,

$$g(\boldsymbol{u},\boldsymbol{p}) = u_i - \varepsilon \geq 0 \qquad (1)$$

$\boldsymbol{p}$ is the random variables. u_i is a component of state variable $\boldsymbol{u}$. $\boldsymbol{u}$ is obtained by the general equation,

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{Q} \qquad (2)$$

General FORM,

$$\begin{aligned} & find{:}\ \boldsymbol{p} \\ & minimize \quad \sqrt{\textstyle\sum_1^d p_i^2} \\ & subject\ to{:} \quad g(\boldsymbol{u},\boldsymbol{p}) = 0 \\ & \qquad\qquad \boldsymbol{K}\boldsymbol{u} = \boldsymbol{Q} \end{aligned} \qquad (3)$$

For high-dimensional problems, the most time-consuming step is to evaluate the gradients. To efficiently get the gradients of the performance function, a Lagrangian function is built first as,

$$L = g(\boldsymbol{u},\boldsymbol{p}) + \boldsymbol{\lambda}^T(\boldsymbol{Q} - \boldsymbol{K}\boldsymbol{u}) \qquad (4)$$

Pixel/Voxel level probabilistic solver – Adjoint Lattice Particle Method (ALPM) 2

The derivative of $g(\boldsymbol{u}, \boldsymbol{p})$ with respective to p_i is,

$$\begin{aligned} \frac{dg}{dp_i} = \frac{dL}{dp_i} &= \frac{\partial g}{\partial p_i} + \frac{\partial g}{\partial \boldsymbol{u}^T}\frac{d\boldsymbol{u}}{dp_i} + \frac{d\boldsymbol{\lambda}^T}{dp_i}(\boldsymbol{Q} - \boldsymbol{K}\boldsymbol{u}) + \boldsymbol{\lambda}^T\left(\frac{\partial \boldsymbol{Q}}{\partial p_i} - \boldsymbol{K}\frac{d\boldsymbol{u}}{dp_i} - \frac{\partial \boldsymbol{K}}{\partial p_i}\boldsymbol{u}\right) \\ &= \left(\frac{\partial g}{\partial \boldsymbol{u}^T} - \boldsymbol{\lambda}^T\boldsymbol{K}\right)\frac{d\boldsymbol{u}}{dp_i} + \boldsymbol{\lambda}^T\left(\frac{\partial \boldsymbol{Q}}{\partial p_i} - \frac{\partial \boldsymbol{K}}{\partial p_i}\boldsymbol{u}\right) + \frac{\partial g}{\partial p_i} \end{aligned} \quad (5)$$

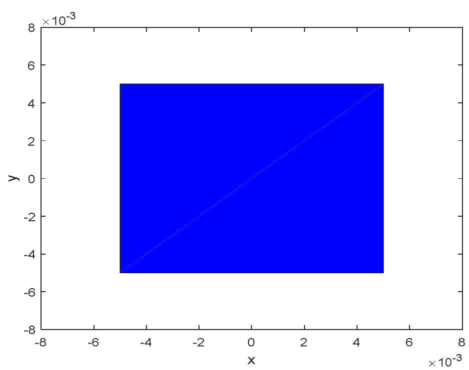
Because $(\boldsymbol{Q} - \boldsymbol{K}\boldsymbol{u}) = 0$ everywhere. $\boldsymbol{\lambda}$ is chosen so that,

$$\begin{aligned} \frac{\partial g}{\partial \boldsymbol{u}^T} - \boldsymbol{\lambda}^T\boldsymbol{K} = \boldsymbol{0} \\ \boldsymbol{\lambda} = (\boldsymbol{K}^T)^{-1}\frac{\partial g}{\partial \boldsymbol{u}} \end{aligned} \quad (6)$$

Then, the derivative becomes,

$$\frac{dg}{dp_i} = \boldsymbol{\lambda}^T\left(\frac{\partial \boldsymbol{Q}}{\partial p_i} - \frac{\partial \boldsymbol{K}}{\partial p_i}\boldsymbol{u}\right) + \frac{\partial g}{\partial p_i} \quad (7)$$

Pixel/Voxel level probabilistic solver – validation example 1



Triangular Lattice

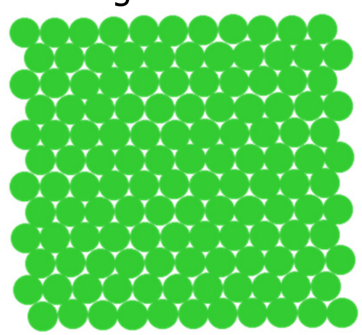


Table 1.Uncertainties of parameters

	Parameter type	value	Mean	C.O.V
Young's Modulus E_i	Normal	-	146GPa	0.02
Loads f_i	Normal	-	$-1.46 \times 10^6 N$	0.02
Possion's ratio ν	Deterministic	0.3	-	-

The designed limit state function,

$$g(\boldsymbol{u}, \boldsymbol{p}) = |u_o| - |\varepsilon| \leq 0$$

Pixel/Voxel level probabilistic solver – validation example (linear)

Table 2. Linear cases

	Case (1)	Case (2)	Case (3)	Case (4)	Case (5)	Case (6)	Case(7)
Dimension	5	7	10	13	15	17	20
Threshold /m	6.0×10^{-5}	8.2×10^{-5}	1.18×10^{-4}	1.522×10^{-4}	1.8×10^{-4}	1.99×10^{-4}	2.25×10^{-4}

Table 3. Results comparisons of linear cases

	Number of particles	p_f		Number of function evaluations			Iterations of ALPM	error
		ALPM	MC	ALPM	MC	C.O.V of MC		
Case(1)	23	0.041097	0.04105	7	20000	0.0342	6	0.1145%
Case(2)	46	0.092256	0.09200	7	10000	0.0314	6	0.2783%
Case(3)	105	0.10069	0.10020	8	10000	0.0300	7	0.4890%
Case(4)	175	0.09362	0.09420	8	10000	0.0439	7	0.6157%
Case(5)	247	0.083466	0.08460	7	10000	0.0329	6	1.340%
Case(6)	314	0.063209	0.06440	7	10000	0.0381	6	1.8494%
Case(7)	429	0.28220	0.28880	10	10000	0.0157	9	2.2853%

Pixel/Voxel level probabilistic solver – validation example (weakly nonlinear) 1

Table 4. Nonlinear cases

	Case (8)	Case (9)	Case (10)
Dimension	115	262	449
Threshold /m	1.23×10^{-4}	1.8×10^{-4}	2.269×10^{-4}

Table 5. Results comparisons of nonlinear cases

	Number of dimensions	p_f		Number of function evaluations				
		ALPM	MC	ALPM	MC	C.O.V of MC	Iterations of ALPM	error
Case(8)	115	0.065716	0.0669	10	10000	0.03735	9	1.7698%
Case(9)	262	0.090868	0.0944	11	10000	0.03097	10	3.7415%
Case(10)	449	0.088386	0.0888	14	10000	0.03203	13	0.466216%

Pixel/Voxel level probabilistic solver – validation example (weakly nonlinear) 2

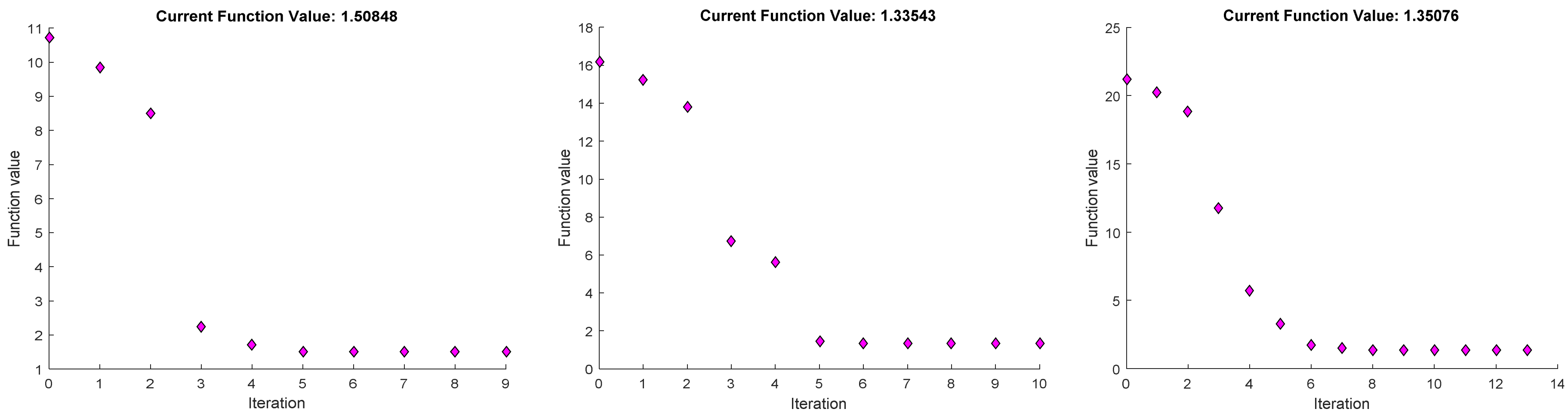


Figure 3. Reliability index convergence procedure for the nonlinear case.
Left : dimension=115 middle : dimension=269 right : dimension=452

- Very consistent convergence behavior irrespective of problem dimensions