PXO: Poly-XTAL operations V10.00. Free MATLAB codebase to generate and analyse complex 2D poly-crystalline grain structures

**THEORY MANUAL V1.0**

Sunil Anandatheertha1

1PhD student, Faculty of Engineering and Computing, Coventry University, Coventry CV1 5FB, UK

**Author contributions**

|  |  |
| --- | --- |
| Sunil Anandatheertha | Conceptualization, Software development and maintenance, documentation and manuscript preparation |

## Q-state Pott’s model



Figure 1: Numbering conventions used on the square lattice for theory and implementation

The simulation can be carried out on any lattice. In this work, we consider a rectangular grid having an underlying square lattice shown in Figure 1(a). The grid has K\*L lattice sites and is associated to ‘n’ state variables defining the temporal configuration of the domain, where ‘*t*’ represents simulation time. The state variables could be strongly or weakly coupled. In a weakly coupled set of state variables for example, either of and () is independent (say ) and the other is dependent. This could be written as

The partitioning of space evolves with the evolution of state variables. In this work, a single state variable is considered. It is easy to incorporate a state variable into the solver code framework in Poly-XTAL operations. At every *n*th instance of the simulation time , depends on where, form the set of 1st nearest neighbours. Temporal evolution of the value of depends on the acceptable change brought about by to a Hamiltonian (). This change is usually the reduction to brought about at in comparison to as against that was brought about at by . The objective is usually to reduce of the entire system. In a Q-state Pott’s model, there are Q number of possible unique values for every lattice site . In the modified Metropolis algorithm for the Q-state Pott’s model, calculated as below [1]:

Where, is the Kronecker delta and J is the isotropic boundary energy.

In this work, is modified as:

Where, is the anisotropic boundary energy function defined over the domain grid.

The partition function becomes then where, T is the temperature and is the Boltzmann constant. The probability, therefore, of the system to be in state is . The probability with which the new state which does not reduce the Hamiltonian can still get accepted is the transition probability and is defined as . By providing the capability to include a temperature field on the rectangular grid, spatially distributed transition probabilities can be used over the grid.

## Boundary conditions



Figure 2: Illustration of boundary condition (a) rectangular grid (b) toroidal boundary condition with torsion and (c) toroidal boundary condition with and

The present model uses self-contained uniquely mapped toroidal boundary condition on the rectangular grid. The present version uses zero-twist toroidal boundary condition shown in *Figure 2*(b). The twisted toroid boundary condition shown in *Figure 2*(b) can also be implemented and is described now. The parent rectangular domain is treated as a toroid shown in *Figure 2*(b). Edges in the parent domain are and corners are . Opposite edges interact ( interacts with adjacent neighboursand; interacts with adjacent neighbours and ), adjacent opposite and diagonally opposite points interact ( interact with adjacent neighbours, adjacent points and and diagonally opposite point . Similar interactions apply to , and ) in the calculation of Hamiltonian . This is done internally in Poly-XTAL Operations 9.04 using an appended element matrix, “ea”. A twist can be introduced into the boundary conditions if needed. If these boundary conditions only define the interaction of grid points lying on the periphery of the domain, then they form the peripheral boundary condition in Poly-XTAL operations. This could be of twisted type. This is visualized in an example in Figure 3(a) for the sampled grid point 14.



Figure 3: Boundary conditions (a) Peripheral type (b) Internal type

It is not difficult to implement custom boundary conditions in “ea” by writing user defined mapping function/scheme to map the boundaries. A new type of uniquely mapped internal boundary condition operating on the entire domain is proposed. It should be straight forward to implement. It is described now. Let , , …, , …, be the set of sparsely intersecting sub-domains of . Let be the number of grid points belonging to , then , operating over maps to and where, and . Two examples are given in Figure 3(b) for the sampled grid point 89 and 56. For the sampled member 89, interaction happens with 50, 53, 56, 92, 128, 125, 122 and 86. The twist in peripheral boundary condition applies only on the periphery and imposes no internal twists to the interaction and the interacting members are either horizontally or vertically or diagonally opposite. However, in the case of , a twist makes a member say, 56 which interacts with non-horizontal, non-vertical and non-diagonally situated members and is shown in by dotted black lines with 53, 18, 21, 35, 71, 94, 91 and 77. A fundamental requirement of these self-contained boundary conditions (both peripheral and presented above) is that no member gets under-mapped or over-mapped.

## Kernel weight functions

Custom kernel weight functions can be incorporated easily into the solver. On a square grid of size say the sampled point is In the calculation of Hamiltonian , the sampled point is taken as having 8 first order nearest neighbours ((1,1) to (3,3), exluding (2,2)), where . Here, a Kernel function is used to introduce anisotropic net non-zero moment into the sampling scheme. The kernel function used to demonstrate the capabilties in this paper is , where, the matrices contains signs, , and contain constants, and and contain spatial coordinates raised by constants. IN this expression, common subscripts do not imply summation. The calculated weights are unit normalized as , where, is the maximum of . For purposes of introducing flexibility, could be non-homogeneosuly rescaled by **R** as , where, is a scalin g matrix having real values, , and are real constants and and are 3X3 matrices having random real and integer numbers respectively.



Figure 4: Example values of Kernel weight function parameters

In MATLAB, they are input in “**CODE NAME.m**” as below:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| sign = [+1 +1 +1;  +1 +1 +1;  +1 +1 +1]; | xconst = [+1 +1 +1;  +1 +1 +1;  +1 +1 +1]; | yconst = [+1 +1 +1;  +1 +1 +1;  +1 +1 +1]; |
|  |  |  |
| xpower = [1 1 1;  1 1 1;  1 1 1]; | ypower = [1 1 1;  1 1 1;  1 1 1]; | power = [-1 -1 -1;  -1 -1 -1;  -1 -1 -1]; |
|  | | |
| MultipleOfNorm = [+1 +1 +1;  +1 +1 +1;  +1 +1 +1] + 0.0\*rand(3,3) -0.0 + 0.0\*randi(2,3); | | |

It should be noted the sub-scripts *i* and *j* used above are not the ones used to represent lattice sites in the Hamiltonian calculation.



Figure 5: Visualization of the Kernel weight matrix elements (a-i) and temporal slices of grain structure at Tsim. = 500 and 1000 evolved by using

Grain structure is simulated on a 100\*100 rectangular grid with square grid and the 500th and 1000th temporal slices are shown in *Figure 5*(i & k). Hamiltonian is calculated with using visualized in *Figure 5*(a - i) considering 1st nearest neighbour interactions. Using different parameters for the calculation of , different spatial partitioning can be generated with varying levels of symmetries. A few examples of the different parameters are provided in the appendix-C for a 32-state system. Users are urged to explore different generating parameters for and report any interesting spatial partitioning. The overall asymmetry in is represented in the visualization of in *Figure 5*(e).

## Estimation of average grain size

Three methods were used to measure the average grain size:

1. Vertical intercept method: In this method a number of verticals are drawn at lattice distances ranging from a minimum of 1 to a maximum of 10. The ratio of the average of intercept lengths to the number of intercepts on a single vertical line gives the average grain grain size measure along that vertical line. The average of these sizes obtained for all the vertical lines made represents the average grain size of the grain structure. In this method, the intercepts made on the secondary phase particles are not considered as they cannot be represented as grains.

2. Horizontal intercept method: This method is similar to Vertical intercept method except that in this case, we use horizontal intercepts.

3. Representative area method: In this method, we calculate the ratio of area covered by all the grains belonging to a particular orientation (q) to number of such q-grains. Equating this to area of a circle, we get the average grain size of all the q-grains, in terms of average q-grain area hGSq.ai, average q-grain perimeter hGSq.pi and average q-grain diameter hGSq.di. The average of these measures for all q ranging from 1 to Q grains gives, average grain area hGSa

average grain perimeter hGSpi and average grain diameter hGSdi. Thus, we

have,

hGSai = hhGSq.aiiq=1:Q

hGSpi = hhGSq.piiq=1:Q

hGSdi = hhGSq.diiq=1:Q

The first two of the above methods are the computationally least expensive. Even though we use a maximum number of intercepts when compared to the third method. But the third method is more accurate and it is performed to measure the accuracy of the first two methods and doing so, we get an idea of the deviation of the actual average grain size and that measured, if we were to use the intercept method. A comparison is made to help compensate the error in the values calculated using intercept method. We employ a varying intercept line spacing which determines the number of intercepts. When a small number of intercepts (that is max. intercept line spacing, lmax) are used, the result does not represent the statistical distribution of lattice grain size. When max. number of intercepts (that is min. intercept line spacing, lmin) are chosen, there are chences to over represent the statistical distribution of lattice grain size. Hence, an optimum number of intercepts should be chosen. This optimum number of intercepts is in turn a measure of optimum intercept line spacing lopt to be used. We assume that a the number of intercepts corresponding to intercept line spacing of the order of the average of hGSii for i = il.max to il.min should be chosen. To do this, we employ following steps.

Choose vertical intercept method (vim).

2. Choose intercept line spacings from lmax to lmin. (l.max and l.min correspond

to il.min and il.max respectively)

3. For each i ∈ [il.min, il.max] calculate hGSii

4. Find hhGSiii for values of i.

Choose intercept line spacing l such that |l − hhGSiii| is minimum.

6. Assume smallest grain size is 1

4

th and largest grain size is 7

4

th of hhGSiii.

These correspond to hhGSiiimin and hhGSiiimax respectively, and lhhGSiiimin

and lhhGSiiimax

respectively. There will be n values in this range.

7. i corresponding to each of above n values of l are iopt values, iopt.n

8. Take hGSii.opt for each of the nth optimum value iopt.n, as GSvim.n

9. Calculate hhGSvim.nini over all values of n. This represents hGSivim.

10. Choose horizontal intercept method (him) and similarly, calculate hGSihim.

11. Take average of GSvim and GShim as the average intercept grain size.

## Modelling of second phase particles in the grain structure

Figure 5.6(a) to 5.6(f) shows the different types of second phase particles considered

for study. The quantity of reinforcement present in a composite is usually specified

in terms of volume fraction (VF) or weight fraction. In our lattice we introduce

secondary phase particles of known volume fraction into the lattice representing the

lattice and perform the simulation. To accurately simulate the grain structure it

is necessary to consider the interacting phases in the most general of their form

possible. For nanotube secondary phase particles, we have considered the effects of

three nanotube parameters such as,

1. Average length of the Nano Tubes (NTs), hlnti.

2. Average thickness/radius of the NTs, hrnti.

3. Volume fraction, Vf.nt.

These parameters define each of the following two configurations of nanotube considered

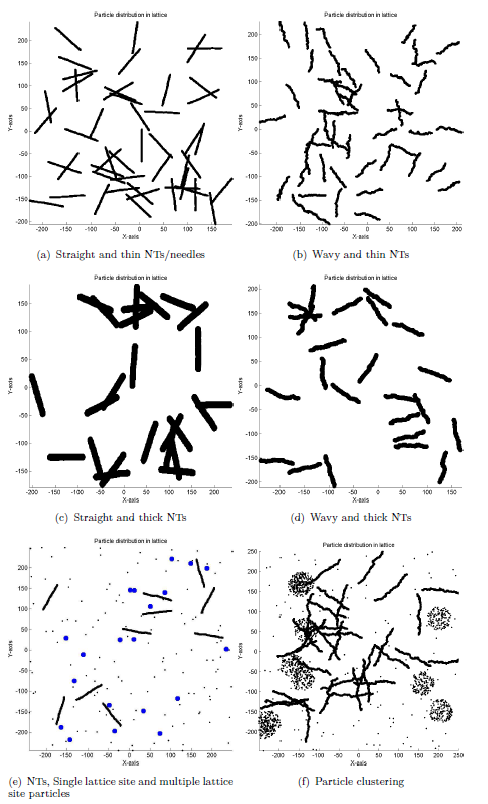


Figure 5.6: Types of secondary phase particles used in simulation: Straight and thin nanotubes/needles (a), Wavy and thin nanotubes (b), Straight and thick nan- otubes (c), Wavy and thick nanotubes (d), Nanotubes, Single lattice site particles and multiple lattice site particles (e) and Single lattice site particle clustering (the individual phases are also considered separately) (f)

Straight Nano Tubes (SNT).

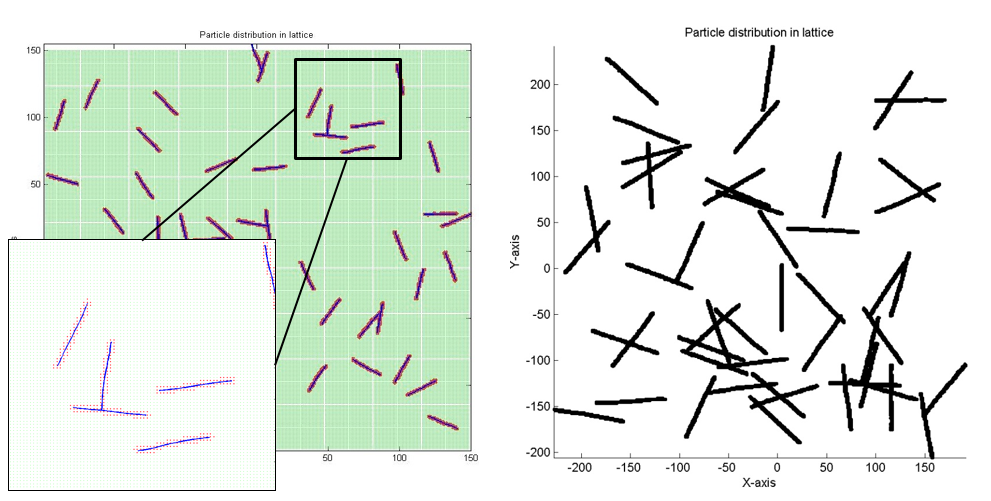
2. Wavy nanotubes (WNT).

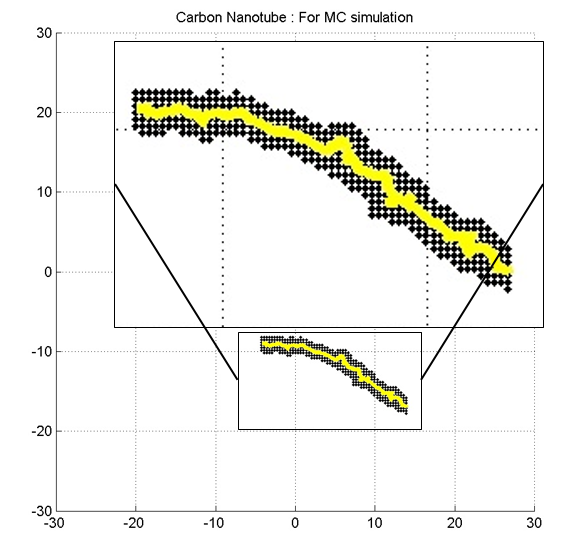
The geometry of needle type reinforcements is very much similar to that of nanotubes, in that it is mostly straight and of aspect ratios ranging from 10 to 20. Hence, in the present simulation studies, the geometry of straight nanotubes with corresponding aspect ratios also represent needle type of reinforcements. In this work, both nanotubes and needle type of reinforcements will collectively be called as High Aspect Ratio Nanoscale Reinforcement (HARNR). Single lattice points (SLPs) are defined as those secondary phase particles which occupy a single pixel. For SLPs, we consider the following two configurations.

Unclustered randomly distributed SLP (USLP), (same as SLSP).

2. Randomly distributed clusters of SLP (CSLP), (same as SLSPC).

While, volume fraction alone is used to quantify the USLP (Vf.uslp), both volume fraction and average cluster radius are used to quantify the CSLP (Vf.cslp).





CNT controlling variables included in the MATLAB code (“ccc.m”)

Thickness

Length

Orientation

Degree of waviness

Degree of randomness

A third important type of secondary phase particle which are used in the present work are the representations of spheroidal particles such as silicon carbide particles and Al2O3 precipitates in A matrix. Two parameters are used to define these particles:

1. Volume fraction Vf.splp.

2. Average radius of particles hrsplpi.

The agglomeration/clustering of NTs due to Van-der Walls attractive forces and due to improper mixing, is a very important factor in deciding the grain structure and also the mechanical behavior composite especially the fracture behaviors and it is the subject of our study in the next related research work. NT geometry is generated using controlled random walk technique and this geometry is superimposed on the numerical lattice domain to get the geometry of the NT in the form useful for the simulation.

[1] W. Blum, Y. J. Li, J. Chen, X. H. Zeng, and K. Lu (2006). On the Hall–Petch relation between flow stress and grain size. International Journal of Materials Research: Vol. 97, No. 12, pp. 1661-1666.

[2] George E. Dieter. Mechanical Metallurgy SI Metric Edition, Third Edition, Mc Graw Hill, Table 6-1, page, 193.

[3] Nicholas Metropolis and Arianna W. Rosenbluth and Marshall N. Rosenbluth and Augusta H. Teller and Edward Teller. Equation of State Calculations by Fast Computing Machines, The Journal of Chemical Physics, Vol. 21, No. 6, Year 1953, Pages. 1087-1092, doi. 10.1063/1.1699114.

[4] Azmir Harun, Elizabeth A. Holm, Mike P. Clode, Mark A. Miodownik. On computer simulation methods to simulate Zener pinning. Acta Materialia. Vol 56, Year 2006, Pages. 2361 – 3273. doi:10.1016/j.actamat.2006.03.012.

[5] J. Gao, R. G. Thompson, B.R. Patterson, Computer simulation of grain growth with second phase particle pinning, Acta Mater., 4 (1997) 3653-3658.

[6] B. K. Kad and P. M. Hazzledine. Monte Carlo simulation of grain growth and Zener pinning. Materials Science and Engineering, A238 (1997) 70-77.

[7] M. P. Hazzledine, R. D. J. Oldershaw, Computer simulation of Zener pinning, Phil. Mag., A61 (1990) 579-589.

[8] Doherty RD, Srolovitz DJ, Rollett AD, Anderson MP. Scripta Met 1987; 21:675.

[9] Smith CS. Trans AIME 1948; 175:15.

[10] Srolovitz DJ, Anderson MP, Sahni SP, Grest GS. Acta Mater1984;32:783.