Note:

$(a)$ sstate() :

- Initialize lattice site and details by iterating through all lattice sites.
- Updates site occupation status and site details.
- Helps to detect overlapping of beads.

$(b)$ fmoves()

- Determines the present orientation of the first segment using the direction vector (fvec).
- If the move is valid updates the bead position vectors.
- Ensures the bead within the lattice boundaries.
- If the move is invalid (folding or out of bounds), resets the bead to its original position.

$(c)$ nmoves()

- Determines the present orientation of the last segment using the direction vector (nvec).
- If the move is valid updates the bead position vectors.
- Ensures the bead within the lattice boundaries.
- If the move is invalid (folding or out of bounds), resets the bead to its original position.

$(d)$ deltaE()

- Computes the energy difference by comparing the number of neighbouring occupied sites before and after move.
- To Prevent overlapping, introduces very large energy penalty.
- Returns the energy difference value.

$(e)$ kmoves():

- Calculates the current orientation of kth and (k-1)th segment.
- Updates the move if the dot product of adjacent segments is zero.
- Ensures the bead within lattice boundaries.
- If the move is invalid, resets the bead to its original position.

$(f)$ metrop():

- It is used as an acceptance criterion .
- Generates a random number and compare with the probability to accept move.
- If delE<=0, accepts the move or else accept move with a probability of exp(-delE).

$(g)$_accmov():

- From the move acceptance by metrop(), updates the bead's position to the new coordinates
- Updates the occupation status of the lattice