

### HA DANG:

Today, I am going to use Mathematica to visualize a nanoparticle polymer network and to investigate how the polymer length affects of probability that polymer would cross the nanoparticles and how the polymer length affects the nanoparticle polymer network formation. Polymers and macromolecules composed of hundred to thousand subunits. They can adopt many different configuration, depending on the identity of their subunits. Common everyday examples of polymers are PTC, polystyrene, Teflon, and so on.

When particles and polymers of are favorable interaction are mixed, it is not always apparent whether they will form a cross-linked network. If the polymer configuration makes the end to end distance of the polymer shorter than the distance between two particles, no polymer-particle network will form. This project will focus on visualizing the interaction between particles and polymers given polymer's length, or the number of subunits.

In stage 1, we make polymer by use the random walk method. We consider a particle as many lattice sites. Polymer with start a random lattice site This initiation part just ensure that the polymer will truly random walk at a certain distance away from the surface of the particle. In the random walk simulaton, the step size is the length of the subunit of the polymer, which is about 10 times smaller than the size of the particles.

The step angle tell you the direction that the poem will walk. The step tell you how much the polymer will walk in the x and the y direction. The next position of the polymer will be the current position of the polymer plus step. Each time the new polymer position is computed, it is stored inside on list which we call path. Once we have the polymer path, we can visualize it inside the particle matrix using Graphic.

In stage 2, we want to generate many polymers and visualize how they distribute and interact with particles inside a lattice of many particles. And in order to do so, we combine all the steps in state 1 into one function, `polymerRandomwalk`, which takes in the polymer length and the lattice of particles and returns a polymer path. We will generate 100 random random walk polymers of 1,000 subunit by using `Table`. We visualize the polymers and particle matrix using `Graphic`.

In stage 3, we want to calculate the probability of crossover between nanoparticle in polymers by counting how many polymers that wrap around more than two particles. In a nutshell, we

arrive a function and `crosslinkSuccess`, which test whether each subunit of the polymer would touch a neighboring particle by using `RegionMember`.

If there is at least a subunit that successfully crosses a particle, the polymer connects two or more particles and the function returns 1. If the polymer fails to cross a particle, the function will return 0. So we run this function over many polymers by using the `PR` function. And this is a resource that we got. And we really want to know the ratio of the cross-link polymer to a total number of polymers. And to do that, we just do a simple math by dividing the number of successfully cross-linked polymer to the total number of polymers. Below is a function that calculates the ratio of polymer that link two or more particles together.

Next, we calculate the ratio of cross linking for polymer of various lengths-- 100, 200, 500, 800, 1,000 and 1,500. Since the operation takes a while, I'll show just the results. The most interesting thing that we can obtain for this data is actually plotting the ratio of cross-link versus the polymer length.

What we find out is that as the polymer length increases, the probability to cross-link increases. This makes sense because as a polymer gets longer, it can walk a larger distance and touch a particle. What is even more interesting is that the graph has a shape of a lock function. We go ahead and use `Fit` to generate least square fit equation. And then we plot it with the data.

As you can see, the square fit function gives you a pretty good fit of the data. What this fit tells you is that first, there's a minimum number of subunits that cross-link will occur. In our case, the minimum polymer length will have to be between 100 and 200 subunits to be able to provide you some type of cross-linking between polymer and nanoparticles. Second, the graph plateaus at a certain polymer length. Any polymers that are longer than those would not increase the probability of cross-link.

Although the analysis of probability for polymer to cross-link is useful, it does not tell any information about the network of polymers and particles as a whole. So in stage 4, I will use cluster analysis to analyze network formation given polymer length and particle distribution. In a nutshell, the function `singlePolymerCrossLinkTest` will return all the particles' lattice sites which the input polymer can connect. Then we use a built-in function, `UndirectedEdge` and `Graph` to visualize this connection.

We want to do this simulation for all the polymers and particle to visualize the connectedness with the polymer nanoparticle network then use a cluster analysis to analyze the network formed by mixing polymers and nanoparticles together. In here, we write a little function that basically run the single polymer cross-link test over many polymer of the same length.

What I want to show you is this GraphCommunityPlot and FindGraphCommunities. In this graph, you see that there's many clusters which has a very dense cross-linking between polymer and nanoparticles. However, most of the clusters do not connect it together. This suggests that a network of polymer and nanoparticle may not be formed, even though you have many cluster of densely cross-linking between nanoparticles and polymers.

What I want to show next is a comparison between a cluster network of 100 subunit polymer and a cluster network of 1,000 subunit polymers. As you can see, as a polymer length increase, there is a higher chance that cluster can connect together. So definitely the polymer length can affect the connectedness of the system and give a higher chance that polymer can connect nanoparticle and form a cohesive network.

In a realistic system of polymers and nanoparticles which have favorable interaction, one a polymer hits a nanoparticle, it will stick to that nanoparticle. In the system that we used above, polymers are allowed to walk in such a way that does not take into account the interaction between nanoparticles and polymers. In state 5, we'll attempt to take into account that interaction by using this algorithm step.

First, we make a lattice of nanoparticle sites. One lattice is randomly selected. And a random walker will spawn from the site and can walk. If polymer walker hits a charge site, including the original one, then the polymer stick to that side. The modifiedPolymerRandomWalk function is very similar to the random walk polymer function above, except for its use RegionMember to test whether or not a subunit touched a particle. If the subunit touched a particle, the function will stop and return the current path.

Here I generate hundreds polymer of 1,000 subunit by using this modified polymer random walk function and using Graphic to visualize it. As you can look into this picture, the probability that polymers will cross-link nanoparticles decreases because you restricted it to stopping once the polymer hitting a nanoparticle.

Although this approach takes into account the interaction between nanoparticles and

polymers, it is still not a realistic representation of a real system. In this approach, we let a polymer subunit spawn on the particle surface and grow into a polymer. In a real system, polymers are thrown into a disperse solution of particles. The confirmation of polymers in a dynamic process in which the entire polymer twist, turns so that the two ends of a polymer find nanoparticle surfaces. This project has not taken into account the dynamic confirmation of polymers in a dilute solution.

For future research, I would like to improve this project by taking into account polymer confirmation when doing polymer random walk and finding how polymer concentration and nanoparticle concentration affect the nanoparticle polymer network by doing cluster analysis. I would like to thank Dr. Keane for your guidance on this project. I would like to thank Professor Carter for teaching 3016, which I found very helpful for the other project I'm working on my UROP right now. And I would like to tell you all the TA's in 3.016 say for helping me during office hour.