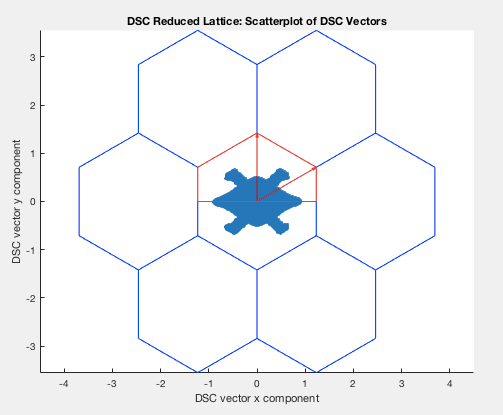
03042020 Beam width simulations

Averaging on either the reduced zone or the extended zone displacement vectors themselves is not the right way to do this calculation, since the interferometry pattern is a nonlinear function of the displacement vector. A much better way is to average the predicted blinking pattern for all the atoms over that range, and back-fit it to a displacement vector that will be plotted on the scatter plot. It seems like the atoms should contribute linearly in the interferometry space, but I don’t know for sure. We can also always do simulations to verify or deny this.

Using the Naïve averaging of the reduced-zone displacement vectors directly:



stepsize = 2;

averaging\_radius = 100;

>> tblg.cellDimXY

ans =

1.0e+02 \*

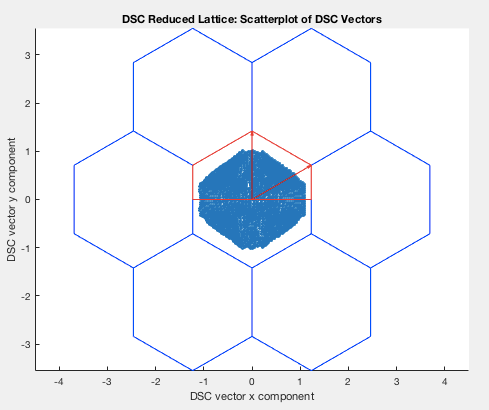
4.909741259180568 2.834640437639314

moire\_angle\_deg = 0.5;

recon\_angle = 0.5;

recon\_distance = 50;

But a more typical picture is generated something like below. Here it’s only clear that we are losing stuff around the edges.



stepsize = 5;

averaging\_radius = 50;

moire\_angle\_deg = 0.5;

recon\_angle = 0.5;

recon\_distance = 50;

For the next stage, can do the same thing with the circle, but compute the averaged blinking pattern. Then need to build a module to re-fit (multistartDiskOptimization, despite its challenges?). Then plot. Do this all as a new method in BilayerGraphene most likely.