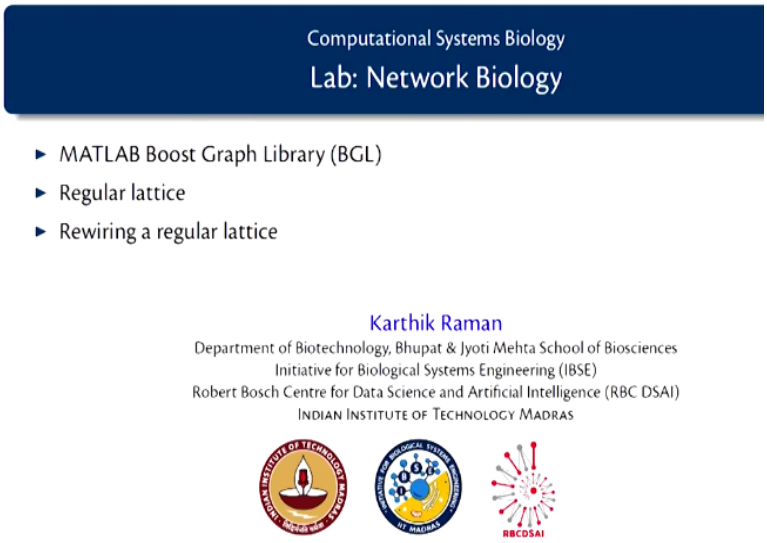
Computational Systems Biology
Karthik Raman
Department of Biotechnology
Indian Institute of Technology – Madras

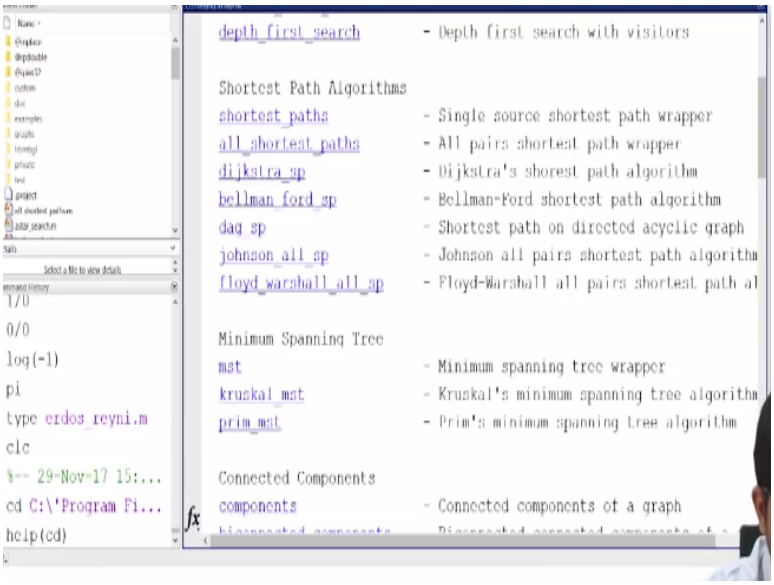
Lecture - 27
Network Biology

(Refer Slide Time: 00:10)



In today's video, we will do a lab where in we look at network biology concepts and how do you? You know build graphs in MATLAB using the boost graph library. How do you analyze for example, computing clustering coefficients or shortest parts and so on and in particular we will see how do you assemble a regular lattice? Essentially how do you build the adjacency matrix for the regular lattice and how do you rewire it? So that you can study various changes that happen to a network post rewiring.

(Refer Slide Time: 00:45)



So, today let us look at how do we work with graphs using MATLAB? There is a tool box called MATLAB BJL, which I hope most of you have already installed. So, 1 useful MATLAB command that I did not show you the other day is this help cd. It tells you, gives you, a brief help about every single file in a particular folder. So this gives you all the different pieces of code that are there in the MATLAB boost graph library.

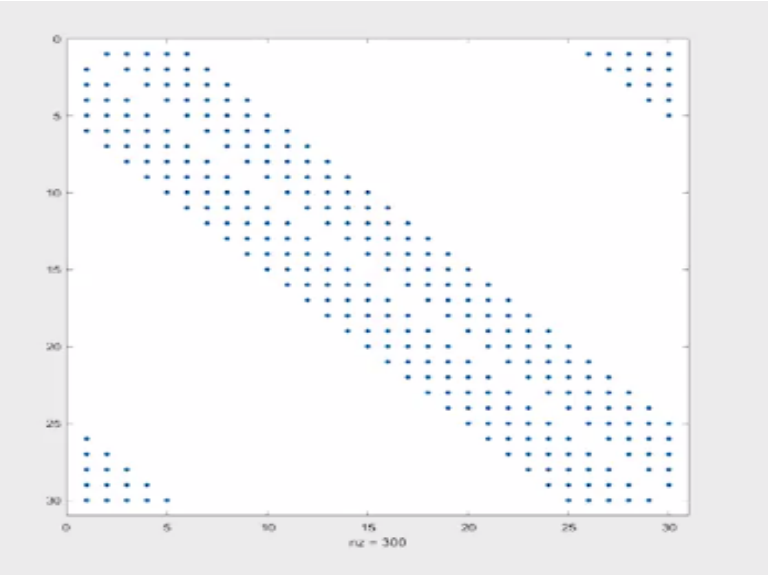
The boost graph library is actually a very powerful library for C, C++ and in this package it essentially calls the boost graph library which is C, C++ based. So we will see that the algorithm run really fast and so you have breadth first search, depth first search, star search and so on. You have shortest paths, all shortest paths. These are some of the more useful algorithms that we need to keep accessing from time to time and then finding connected components in a graph.

Then of course between a centrality clustering coefficients and so on and you can also create different kinds of graphs. This lists all the different the contents holder, the contents mfile actually gives you all this information and this as you will see is a pretty old package, but still remains a very useful and very powerful package for working with graphs and MATLAB.

MATLAB has a number of built-in tools today, but I think the boost graph library is still faster and more effective. So, today can we work on creating a regular lattice start rewiring it with

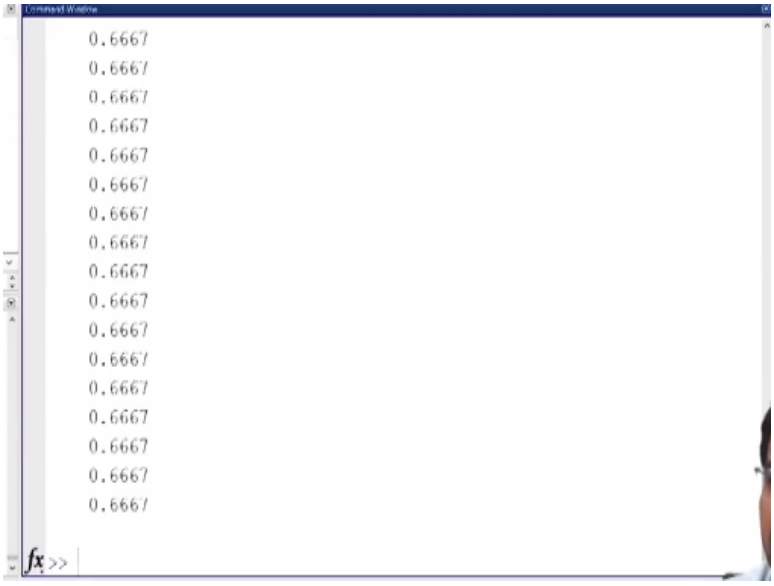
some probability and compute how different parameters change that is something we can try and work with.

(Refer Slide Time: 05:06)



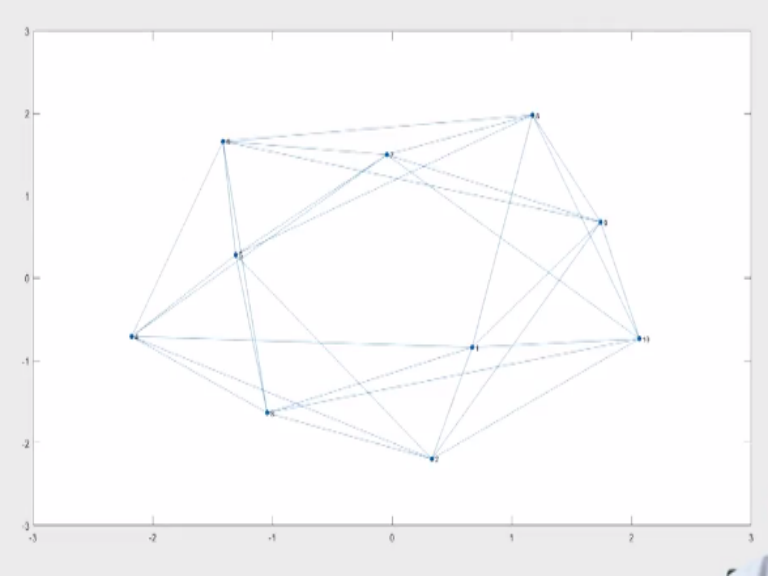
Looks right it is may be too big to see. It looks fine to me. How many neighbour should be there 30 and 5. **"Professor - student conversation starts"** What is the degree of each node? 2 * 5. K on either side 10. So 30 * 10 300 that is what you see here 300 non-zeros. No graph object, just create an adjacency matrix. There is nothing called a graph in MATLAB BJL. In MATLAB you have graph objects. In MATLAB BJL every piece of code requires a sparse adjacency matrix as input. So in fact I should be saying.

(Refer Slide Time: 06:55)



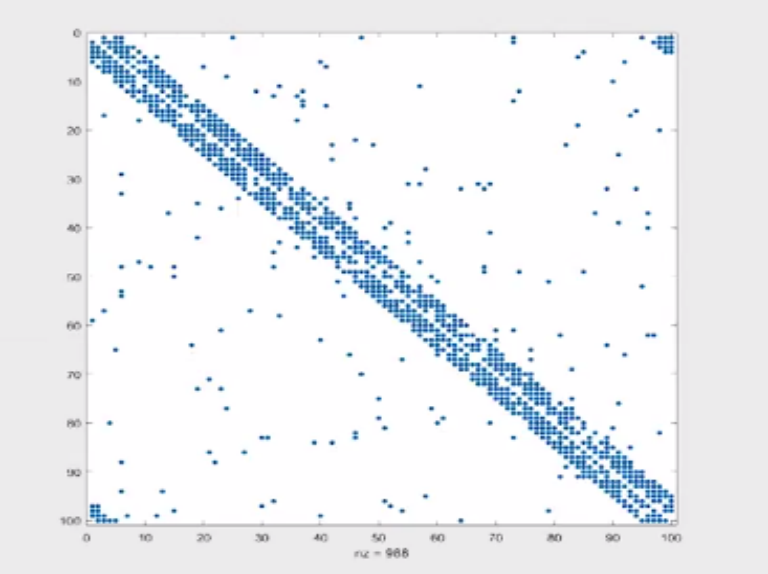
What was the clustering coefficient formula we said yesterday? 3k - 3 before k - 2. So if k is 5, 12/18. 2 x 3 or if you want how does the degree of distribution look like? Spike at 10 all the 100 nodes have a degree of 10. Sir how you got 0.6667? I use the clustering coefficients built in MATLAB BJL. Mod. I thought you saw my code as I typed it. I used mod. **"Professor - student conversation ends"** (()) (11:34) I think there is something. Let me find that.

(Refer Slide Time: 12:16)



But this actually uses MATLABs built in layout, MATLABs built in graph command, graph objects. There still seems somewhat regular and see that but if you had a nice circular layout it would have looked nicer.

(Refer Slide Time: 17:11)

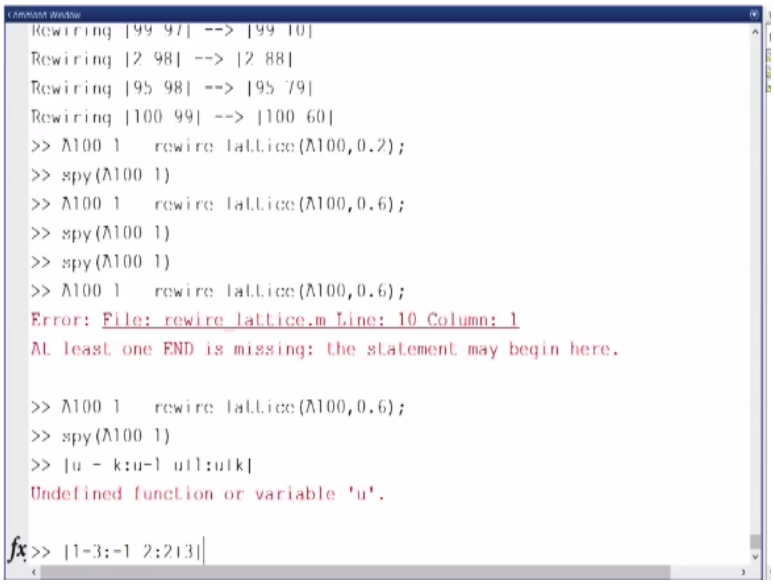


So what does this look like? **"Professor - student conversation starts"** Some rewiring. So order is mostly maintained, plus that you have some rewiring. It clearly looks like the rewiring have done has a little wrong because the number of 0s has changed. It should be some round number. It should have been like 1000 because this is 100, 5 lattice. So my first argument is all my neighbours.

So if I am 0 my neighbours will be - 1, - 2, - k, 1, 2, k, but it is not - 1 so I add k on each side. So your K is neighbors on each side. Why you have added 1 to the main? It is easier to think of it. Still it is only k on each side that is how we define regular lattice. So this 5 nearest neighbours you will take 2 on each side and the third there are 2 equally close neighbours which are going to pick, the fifth one. **"Professor - student conversation ends"**

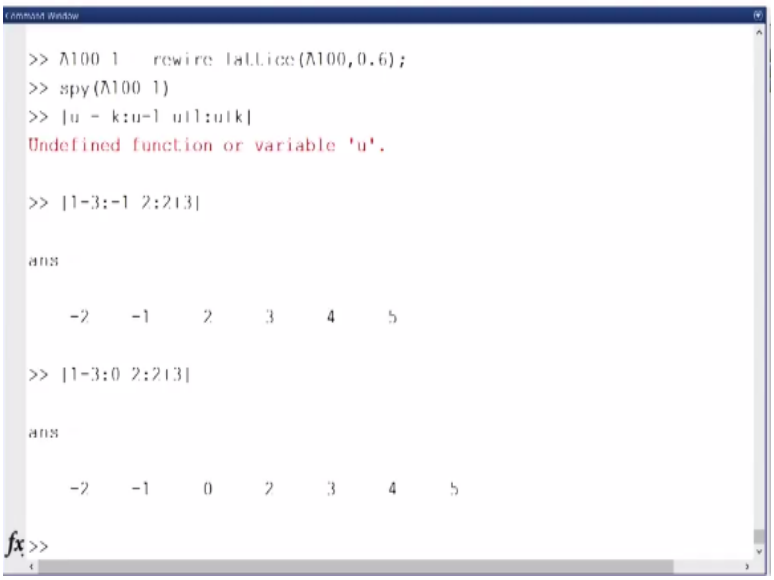
So instantly you basically take it as k on either side that is the straight forward way too. You can pass an array as argument anything in MATLAB, because in MATLAB everything is a matrix. **"Professor - student conversation starts"** Why are you adding 1 to the Mod? Because I am unnecessarily removing a 1. Why are you removing 1? Because to do the 0 offset. So let us look at this. **"Professor - student conversation ends"** So if you scroll really up you will find my.

(Refer Slide Time: 19:52)



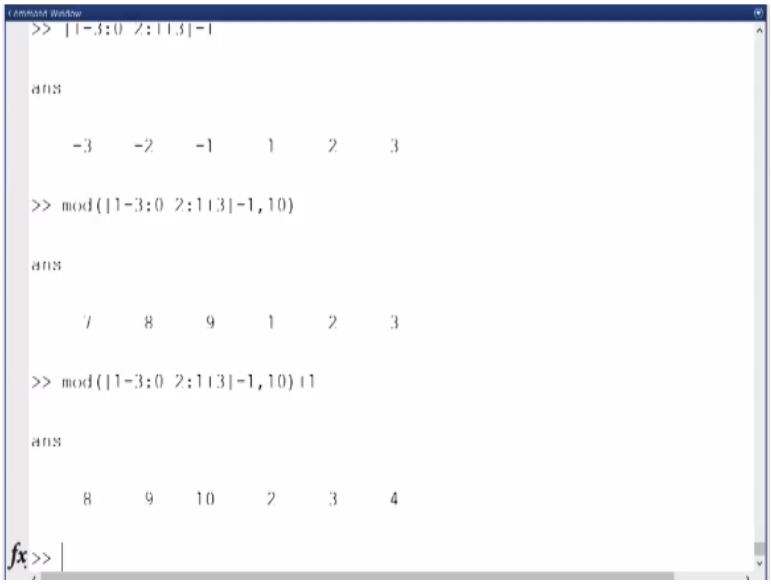
I will clear it out, but so I need to do u - k so let us say 10 - 3: Let us say u is 1, 1 - 3: - 1 and 2: 2 + 3. This should be a known number 1 these are the neighbors agreed. If I had 3 neighbors on either side.

(Refer Slide Time: 20:32)



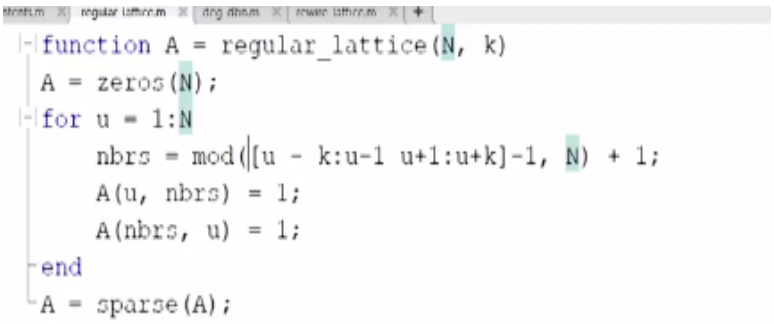
This is correct - 2, - 1, 0, 2, 3, 4, 5. It is nothing called 0 in MATLAB. 2, 3, sorry, sorry so this is 1: is that right - 2, - 1, 0, 2, 3, 4. But then if I do mod of ans, 10 I will get 8, 9, 0, 2, 3, 4 which is perfect if you were in Python because when your nodes are going to number 0 to 9 there is no 10th node. So instead what you do is you do - 1.

(Refer Slide Time: 21:52)



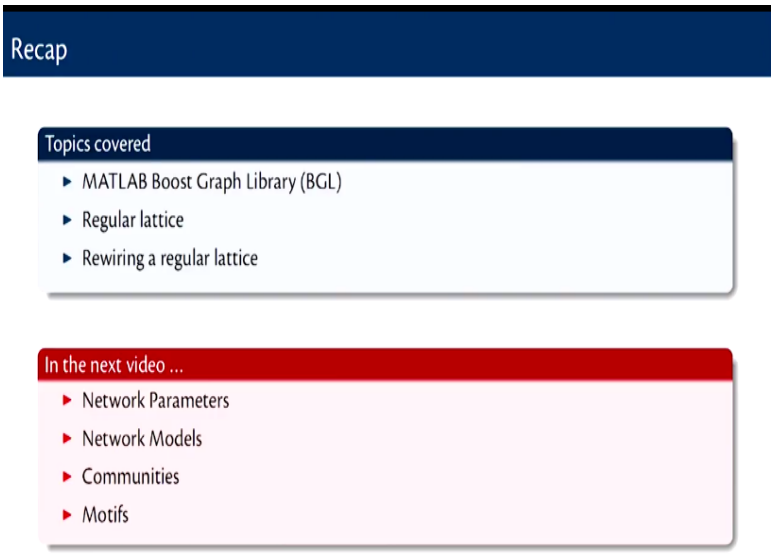
This gives you and then you do mod of this, 10 + 1. **"Professor - student conversation starts"** In Python even though I number my nodes from 1 and put an If-Else statement, and said If I + < > n and if the Mod is 0, then it will take. Exactly A boring and simpler way to do this would be to run another for loop and basically compute if abs i - absolute value of i - k < k, i - u < k then connect it. But I have commented it out.

(Refer Slide Time: 22:37)



But this will be the easiest way to do it and I think you can make it even better. I think you may be able to avoid this loop all together. **"Professor - student conversation ends"** So the mod calculation will now be correct. I am still having an ans portable bug in the rewiring. So you should try rewiring at home. Well if you finish rewiring at home, we should be able to do it because if you finish rewiring what you can do is you can compute characteristic part length, you can compute all of that and.

(Refer Slide Time: 23:03)



So I hope you had a good introduction to MATLAB boost graph library in today's video. I must emphasize here that MATLAB has since developed many graph functions you know which are inherently available in MATLAB, but I still like boost graph library because it interphases with c and it is still very fast and works well even today and I hope you, you can go back home and you can try out how to build a regular lattice, rewire them and so on at home.

In the next video, we will look network parameters, network models, communities, and motifs and essentially recap network biology.