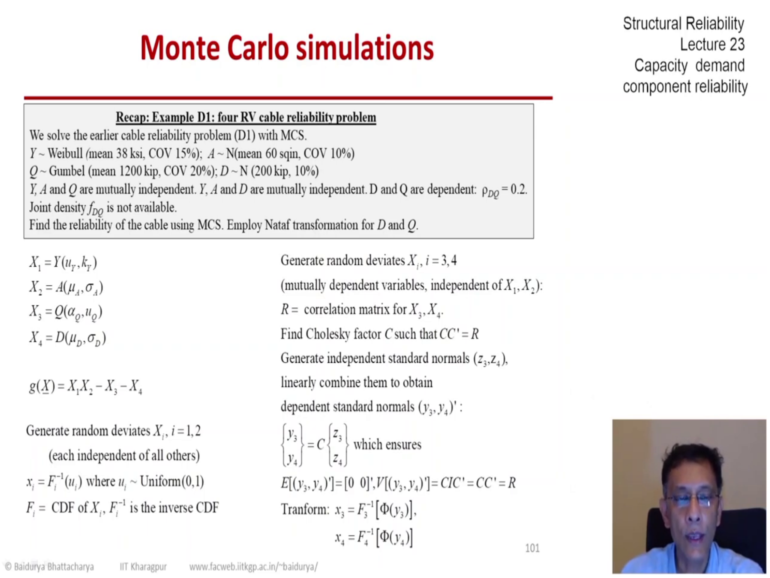
Structural Reliability
Prof. Baidurya Bhattacharya
Department of Civil Engineering
Indian Institute of Technology, Kharagpur

Lecture –187
Capacity Demand Component Reliability (Part 35)

(Refer Slide Time: 00:27)



The last problem we solve in this lecture is our old friend the cable reliability problem the four variable version of that which we called example D1. So, let us take a few seconds to recap this problem statement and then we will solve this using Monte Carlo simulations we have already solved this with FORM and SORM. So, we have four random variables X 1 X 2 X 3 X 4 out of which X 1 and X 2 are each independent of the other three and X 3 and X 4 are dependent on each other. So, our limit state function looks like X 1 X 2 minus X 3 minus X 4.

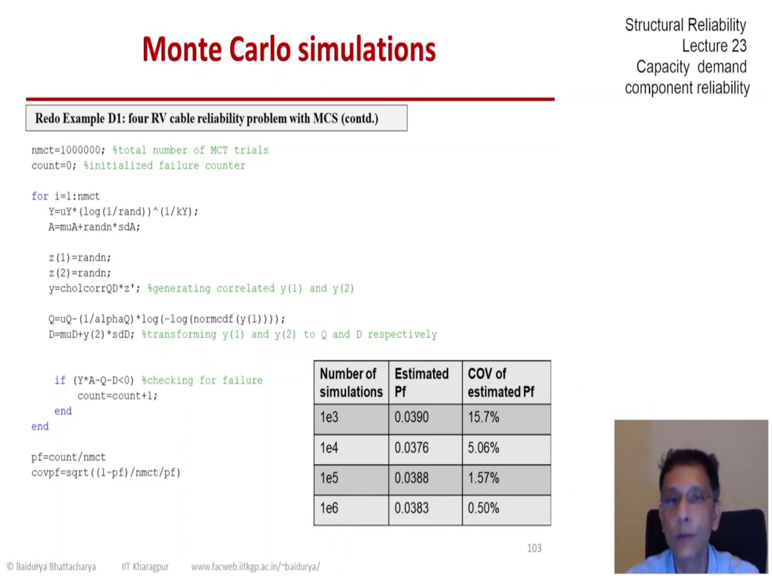
And in Monte Carlo simulations we have to generate these X 1 X 2 X 3 X 4 repeatedly. So, for the first two for X 1 and X 2 it is trivial we all we would need is to use the inversion method X 1 is the yield strength viable X 2 is the area normal. So, we have the ability to generate them from standard uniform deviates by the inversion method. So, that is straightforward we already did this in the previous example involving the save stopping distance.

But for the correlated ones we have to make sure that we ensure that the dependent structure is maintained. So, X 3 and X 4 are dependent on each other but they are independent of X 1 and X 2. So, I will let R be the correlation matrix of X 3 X 4 that is all we know about their dependent structure we do not know the joint distribution. So, we have the correlation matrix the 2 by 2 matrix and we do a cholesky factor the lower Cholesky factor C such that C C transpose is equal to R.

And then we generate independent standard normals z 3 and z 4 and then we combine them linearly to obtain dependent standard normals y 3 and y 4 with the help of c. So, y 3 y 4 are linear combinations of z 3 z 3 z 4 and just to make sure that this way we get the mean of y 3 y 4 as zero and the V of y 3 y 4 is r and then we do an inversion of y 3 to X 3 and y 4 to X 4. This way we ensure dependence between X 2 and X 4 however because of the non-linear transformation between the X's and y's we do not end up with the same R that we started with.

We have discussed this before if it is very important to ensure the exact value of row between X 3 and X 4 we should start with different values that we did at the top while finding the cholesky factor but even without that the error is very slight in most cases.

(Refer Slide Time: 04:52)



So now we are ready to present the MATLAB code and then the solution. So, we start with defining the basic variable statistics we do it for Q then we do it for A we do it for D and then we introduce the we bring in the correlation structure between Q and D. So, we define the 2 by 2 correlation matrix and then take the Cholesky factor and the last command in that line that you see is to make sure that it is the lower.

So, risky factor then we obtains the statistics of y. So, because y is viable we go through that step we have shown those before to get the shape and scale factors of the viable distribution then we are ready to start the loop we have 1 million trials once again here we are going to change that and show how the results keep getting better we initialize the count. We start the FOR loop and then we generate the random variables one by one we first generate y from the command rand which gives us the standard uniform deviate.

And we invert that to get the variable y we use the rand n command to generate the standard normal deviate and then transform that to the normal A and then for Q and D we generate two standard normals independent standard normals z 1 and z 2 and then we linearly combine them to get y using the lower choice Q factor and then once we have y one and y 2 we individually invert them to get Q and D.

So, Q comes from y 1 and d comes from y 2 using the full distribution transformation and now we are ready to check for failure. So, our limit state is y A - Q - D. So, if failure happens then we increment the failure count by 1 and that brings us to the end of the for loop and we are now we can output the estimated failure probability and the cov of that estimate we have run this for four values of NMCT the first one is just 1000 estimates and that gives about 3.9% failure probability but the cov is a bit too large about 16%.

So, let us increase the number of samples the next one is 10 000 still about 3.8% the cov comes down to 5%. So, we could stop there but let's see how the improvement goes on uh. So, 10 to the power 5 samples gives us about 3.9% and 1.6% cov and then if we do a million simulations it is still about 3.8% and it is half a percent of error. So, we can stop now we could have stopped much earlier and this result of a 3.8% we actually used when we were comparing the accuracy of FORM and SORM.