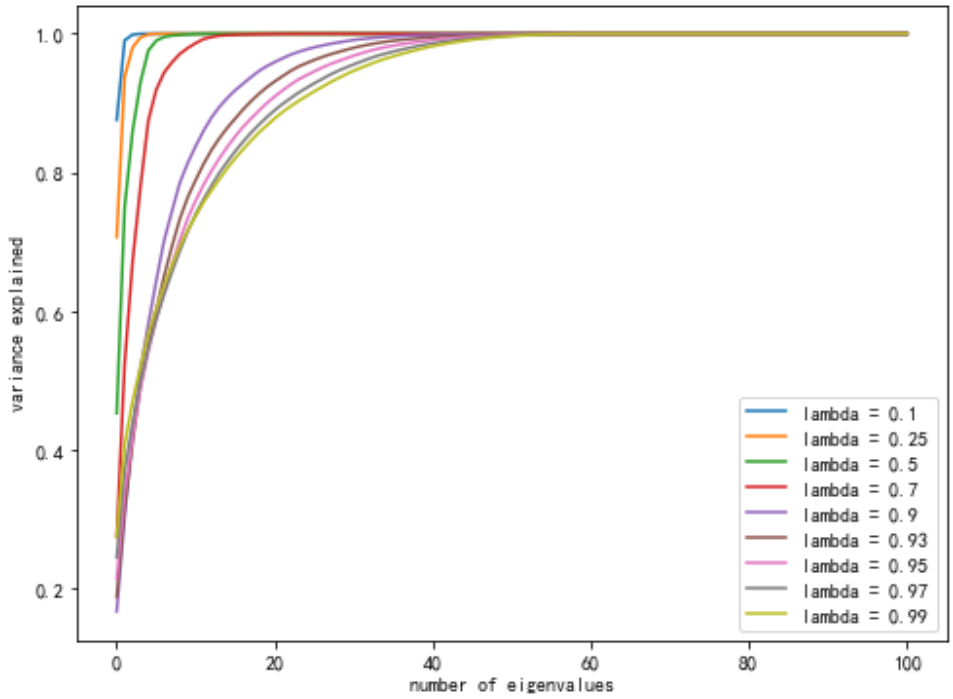
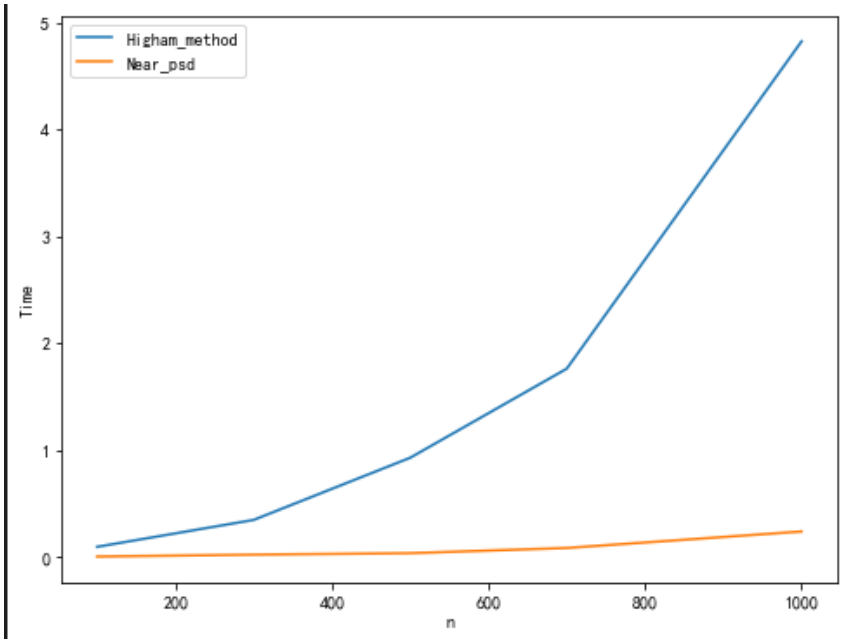
Week 03
Xiaokuan Zhao

Problem1:
What does this tell us about values of and the effect it has on the covariance matrix?

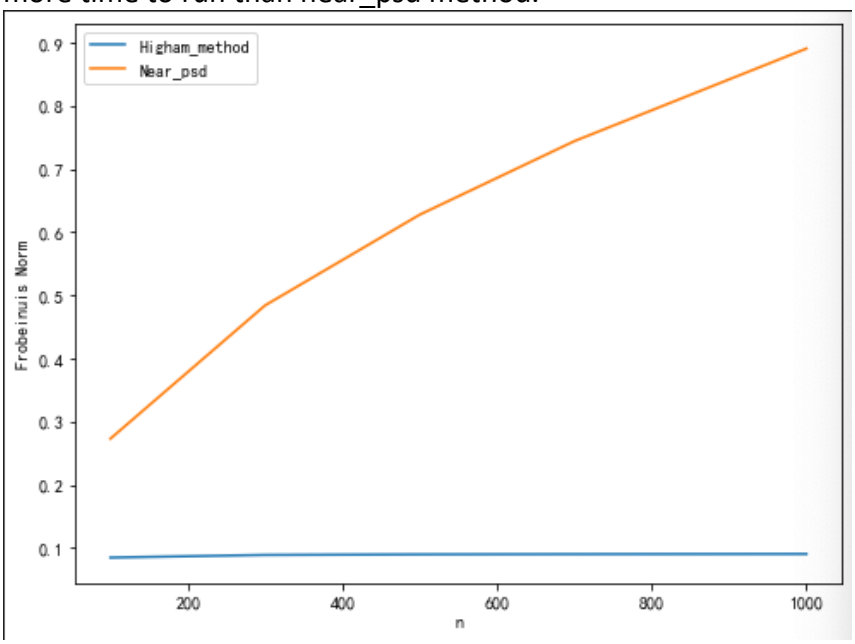


Lambda in this situation represents how much we value the recent information. The lower lambda is, the higher proportion we put on recent information because the weights are decaying fast. In our graph, we can see with a higher lambda, the weights decay slower and it is more evenly distributed.

Problem2:



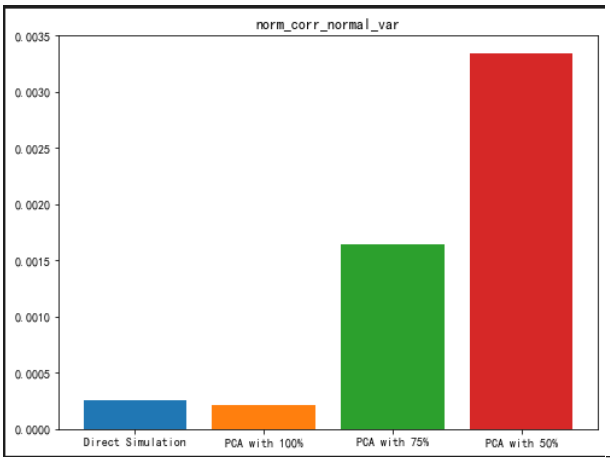
We can see from here, when the dimension of the matrix goes up, the Higham method requires more time to run than near_psd method.

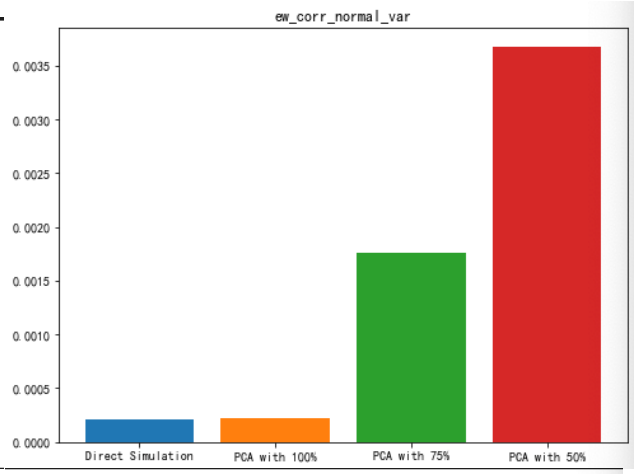


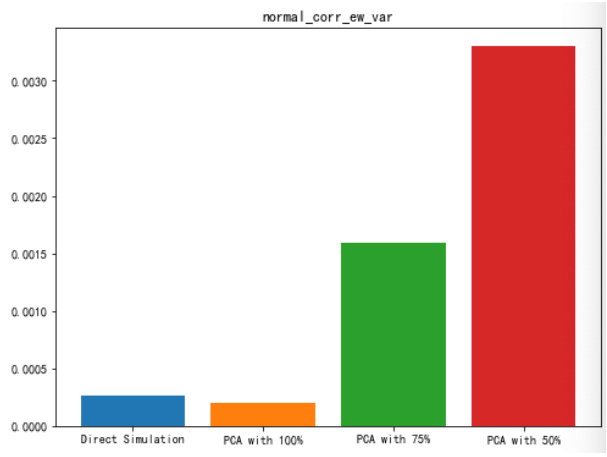
However, the error term of the Higham method is constant when dimension goes up but the error of the near_psd method goes up with the dimension.

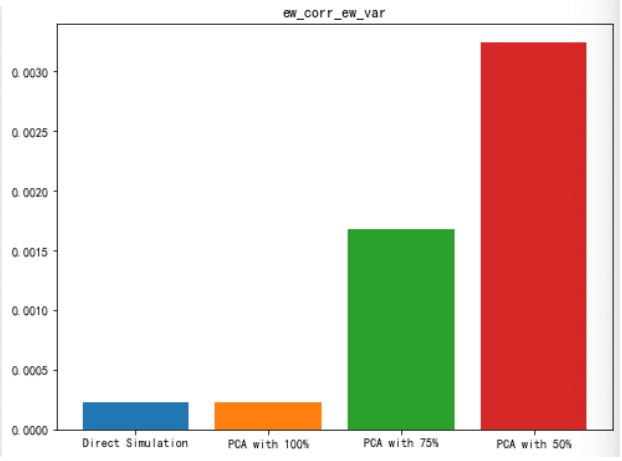
So, Higham method is more accurate but the computing cost is high and takes longer. On the other hand, the near_psd method is quicker to run, but it is not as accurate as Higham method. For me, I prefer to use Higham method when the matrix is small or the computation requires high accuracy. But when the matrix is very big (like over 5000) and the requirement of accuracy is not high, I will use near_psd method.

Problem3:

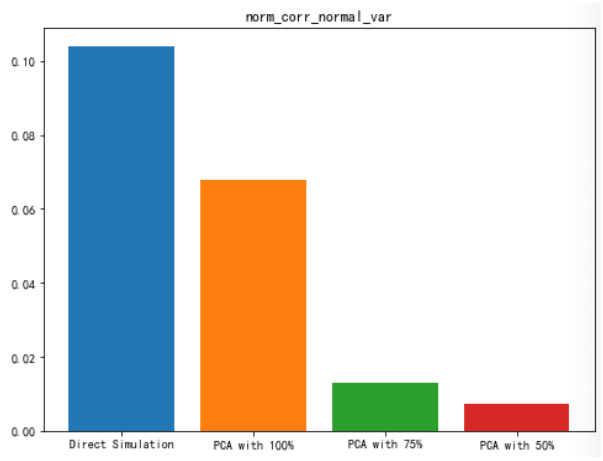


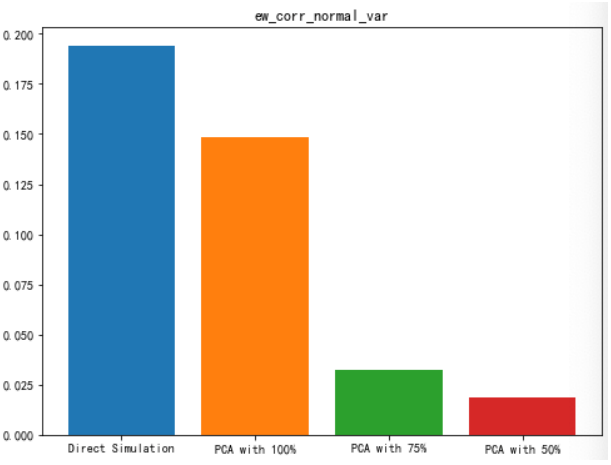


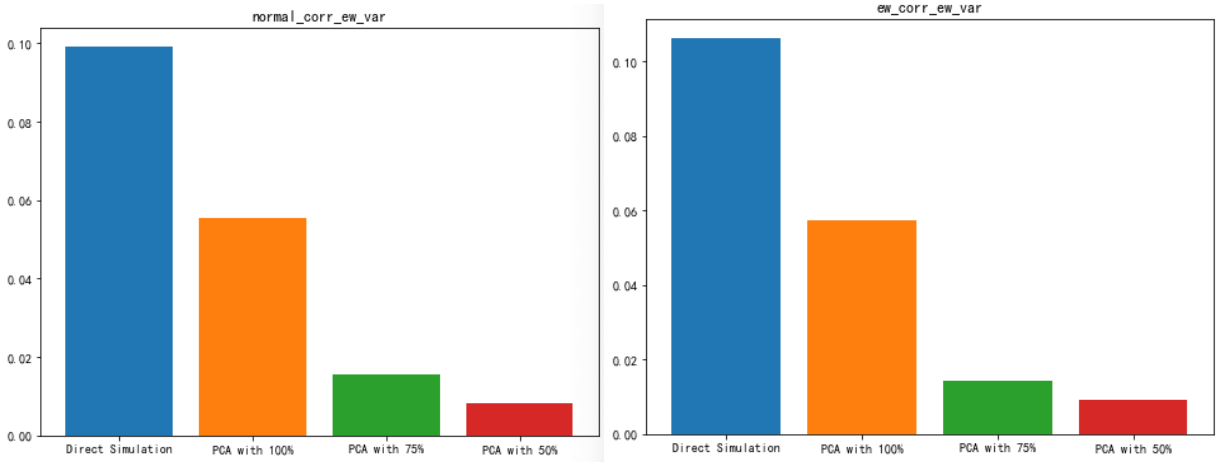




We can see from here, the error term (measured by the Frobenius norm) is low for both direct simulation and PCA with 100%, but it goes higher when the PCA percent goes lower. It is because the PCA percent goes lower, more features are left behind so the error goes higher.







But when we look at the computing time, we can see that the computing time decrease with the PAC percent decrease. The rationale is very simple because the dimension of the matrix goes down, so there is less computing time.

This is an example of the tradeoff between accuracy and computing time. The lower accuracy we want, the less computing time. Even a PCA with 100% will lower the computing time because features will an eigenvalue too small is discarded.