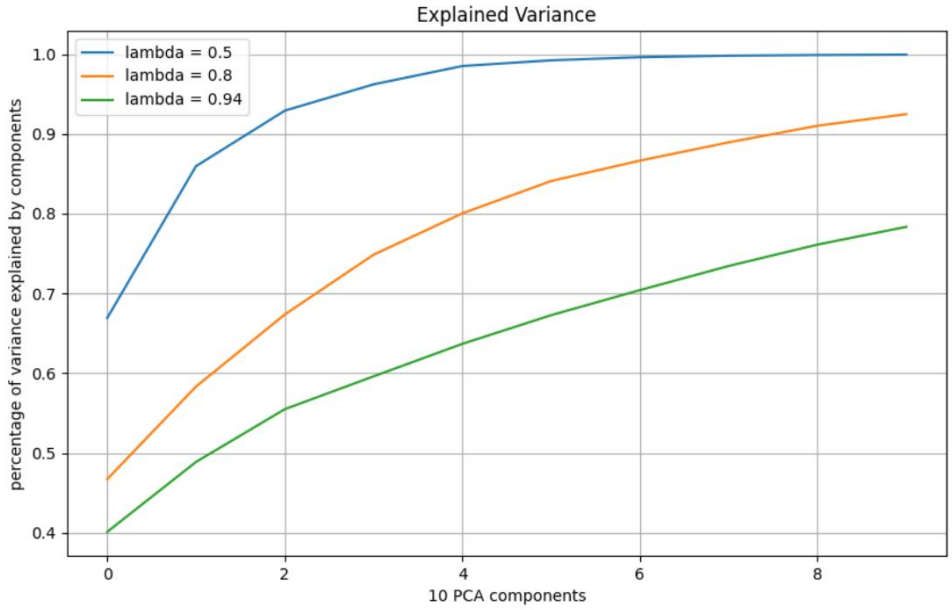
Problem 1



lambda here represents the weight in previous covariance, which means higher the lambda, higher the previous cov weight will be considered into cov calculation. Therefore, we can see from plot that lambda lower is better since PCA explained variance coverge to 1 after second PCA component, which makes sense becasue lower lambda lead to higher weight in [t] time weight (new cov) , the final covariance is more reachable by recent covariance, thus it is more likely the total variance can be explained by those few main components.

Problem 2
when n = 500:
the matrix fixed by near_PSD is PSD matrix now: True
the the runtime for near_PSD is: 0.014586
the matrix fixed by higham_PSD is PSD matrix now: True
the the runtime for higham_PSD is: 0.028744
Fro norm for higham_psd is 0.063643,while Fro norm for near_psd is 0.627523

when n = 5000:
the matrix fixed by near_PSD is PSD matrix now: True
the the runtime for near_PSD is: 11.163514
the matrix fixed by higham_PSD is PSD matrix now: True
the the runtime for higham_PSD is: 22.064304
Fro norm for higham_psd is 0.064234,while Fro norm for near_psd is 1.996614

as we can see from output of code, higham is about 2 time slower than near_psd, and it has much smaller fro norm since it has smaller distance with original sigma matrix, it aligh

with higham psd's algorithm because it should be the minimal change matrix that convert it from abnormal matrix into psd matrix. However, when considering to large n, the fro norm is not very large for near_psd, plus it's also faster than higham.

Problem 3
output:
the order of fro norm start from simulation of cov1 to cov 4 between their original cov matrix:

fro norm for four cov matrix based on direct simulation is 7.408485762389207e-08
fro norm for four cov matrix based on direct simulation is 1.7563497003180426e-07
fro norm for four cov matrix based on direct simulation is 8.424972887914081e-08
fro norm for four cov matrix based on direct simulation is 2.8824008013455094e-07

fro norm for four cov matrix based on PCA100 is 7.408485762389128e-08
fro norm for four cov matrix based on PCA100 is 1.756349700317862e-07
fro norm for four cov matrix based on PCA100 is 8.424972887914395e-08
fro norm for four cov matrix based on PCA100 is 2.8824008013454596e-07

fro norm for four cov matrix based on PCA75 is 5.825903993540617e-07
fro norm for four cov matrix based on PCA75 is 2.148762022684103e-06
fro norm for four cov matrix based on PCA75 is 7.60812666097185e-07
fro norm for four cov matrix based on PCA75 is 3.8454299299550924e-06

fro norm for four cov matrix based on PCA50 is 1.292504537527612e-06
fro norm for four cov matrix based on PCA50 is 6.788938368467905e-06
fro norm for four cov matrix based on PCA50 is 1.717058554466493e-06
fro norm for four cov matrix based on PCA50 is 1.8088370616332352e-05

As we can see from the output, the accuracy is same between PCA100 and direct simulation, which meets out expectation, and the accuracy gradually go lower (fro norm go higher) from PCA 100 to PCA 50, which means as we sacrafice more explained variances (eigenvalue), we don't have that much main component, and the re-generated cov matrix will be more different than original matrix, which also make sense.

runtime output:
PCA100:
the runtime is: 0.005857
the runtime is: 0.002834
the runtime is: 0.002702
the runtime is: 0.002869
PCA75:
the runtime is: 0.002820
the runtime is: 0.002652

the runtime is: 0.002625
the runtime is: 0.002624
PCA 50:
the runtime is: 0.002632
the runtime is: 0.002591
the runtime is: 0.002560
the runtime is: 0.002560

As we can see from output, run time gradually go faster as we reduce our explained variance requirement, which meat our expectation also, because again when there are not so many main components, the matrix operation will be a little bit easier, and the difference is not very large due to both high speed of numpy and computer.

conclusion: when we try to increase accuracy, the run time will be higher, we trade more calculation to get higher accuracy.