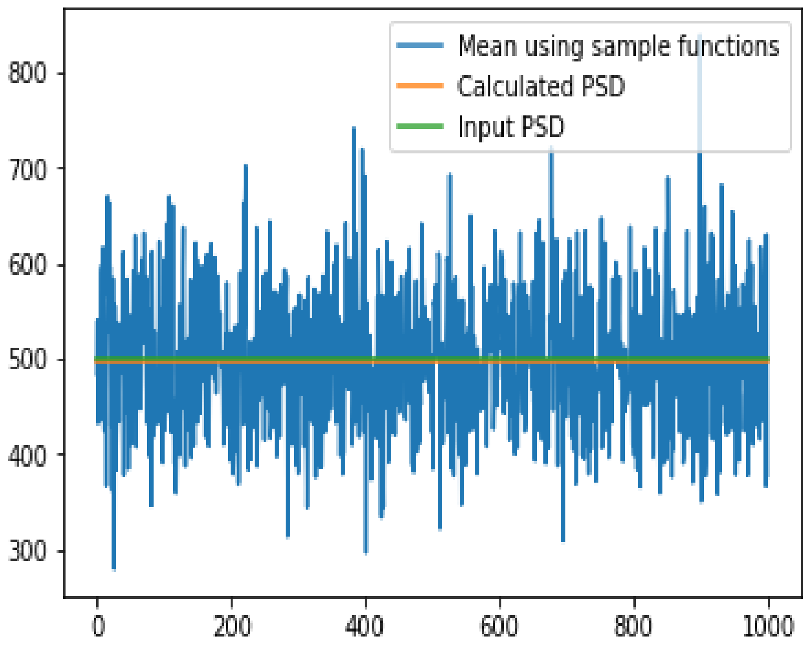
1. (a) Algorithm
   i. Calculate σ via the *myawgn* function. $\sigma = \sqrt[2]{2BK}$
   ii. For the required AWGN of length l we used random.randn(1,l) function of python.
   iii. To verify that input PSD is same as the one produced computationally , create m sample functions/instances of length n, each consisting of elements from the standard normal distribution multiplied with σ.
   iv. Take Discrete Fourier Transform with respect to every row/instance.
   v. Multiply every row with it's complex conjugate to obtain the sum of squares of the absolute values of the fourier coefficients,then find the mean and divide it with the length of the sample function and sampling frequency.
   vi. Perform this operation for every row. We shall obtain a mx1 array consisting of PSD of each sample function.
   vii. Calculate mean of all the instances in the mean vector. The value should tend to *K (PSD)*.

   (b) Observations
   i. For the given graph *PSD = 500 W/Hz , B = 100Hz , Fs= 200Hz , l = 100*
   ii. We obtain *PSD = 497.4848 W/Hz* which almost tends to the input PSD.



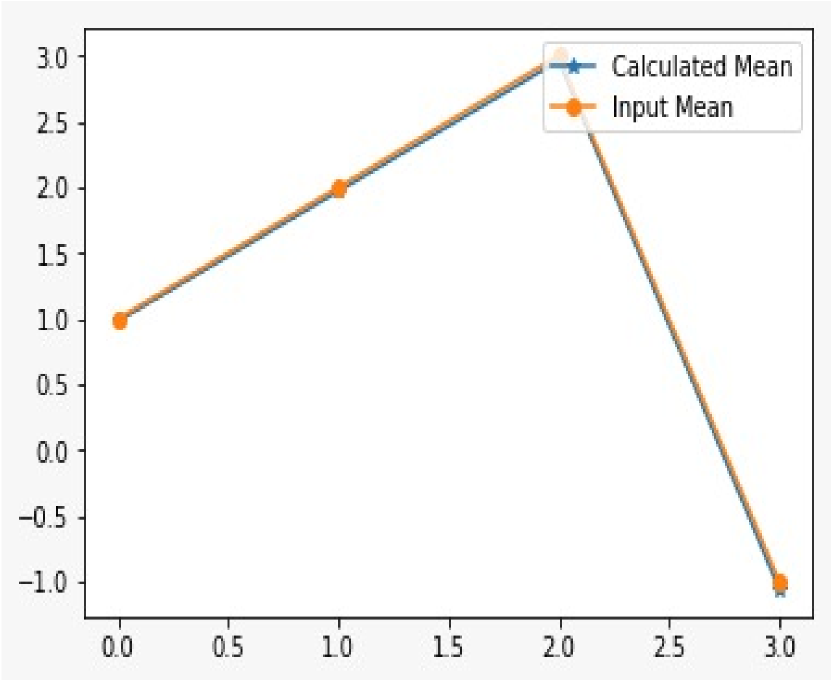
2. For a normal distribution :- $Z = \frac{X-\mu}{\sigma}$
   Where,
   Z is standard normal distribution
   μ is the mean of the required distribution
   σ is the standard deviation of the required distribution
   X comes from a Gaussian distribution with given mean and covariance
   $X = Z\sigma + \mu$

   (a) Algorithm

      i. We first find out the eigenvalue and eigenvector of the covariance matrix and call the eigenvector as P.
      ii. By performing the operation $P^{-1} * Cov * P$, we will get a diagonal version of the covariance matrix in the basis of the eigenvector i.e. we get the diagonal matrix D.
      iii. We then take the square root of the resultant diagonal matrix D, which will give us the square root of the covariance matrix in the basis of the eigen vector.
      We then perform $P * D * P^{-1}$ to convert the diagonal matrix in the basis of eigen vector to the standard basis.
      iv. We generate an array of normal random variable with mean = 0 and variance = 1, multiply it with the resulting matrix of the previous step and add the mean to it, resulting in the samples from the Gaussian distribution with given mean and covariance.
      v. To verify our result, we calculate the mean and variance of the resulting matrix.

   (b) Conclusion
   We generated 10,000 samples using the algorithm given, and the mean and covariance of these samples approximately matched with the mean and covariance given as input.

      i. From the following plot we can verify that the mean of the samples produced is the same as the mean provided in the input.Calculated mean and actual mean of every x_i coincides for every case as seen in the plot.



ii. From the following plot we can verify that the covariance of the samples produced is the same as the covariance provided in the input (here covariance matrix was reshaped into a $1 * N^2$ array for plotting).Every entry in the actual covariance matrix and calculated covariance matrix tend to the same value as seen in the plot.