UQ 6310 Homework3

1. Solution: Given the autocorrelation function $R(\Delta l) = e^{-\Delta l}$, the correlation matrix is generated by calculating the value of $R(\Delta l)$ at different location difference, which are $0, 1L, 2L, 3L$ with time step L. The eigenvalue and eigenvector are further calculated. Choosing the first few eigenvector and eigenvalue, by equation (9) (To make the simulation more accurate, we use all of the eigenvectors and eigenvalues), the simulation result is as follows:

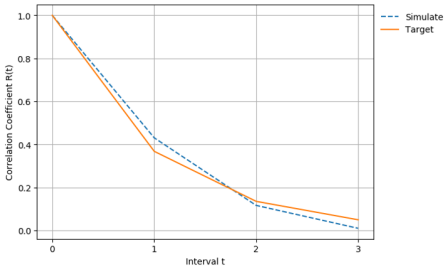
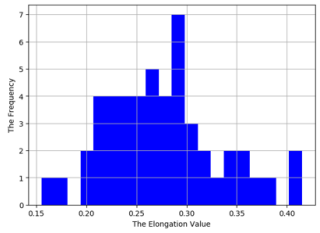


Figure 1: Simulation Gaussian process using K-L method

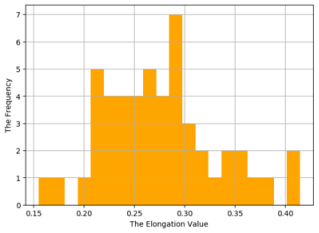
Comparing the simulated Correlation Coefficient curve with the Given Curve, the random process of E is simulated very well.

For each $E_i, i \in 1, 2, 3, 4$ and P, we sample 50 data points as our training points. By our physics model as Eq.(1), the elongation value for each training points can be calculated.

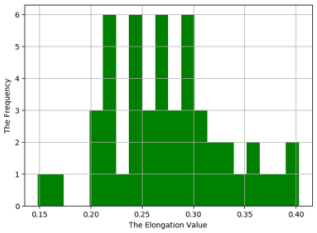
$$Y = PL/(E_1 A) + PL/(E_2 A) + PL/(E_3 A) + PL/(E_4 A) \tag{1}$$



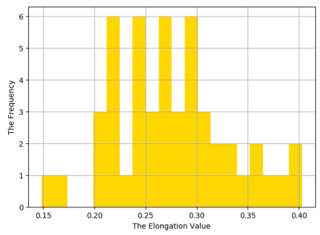
(a) By physics model



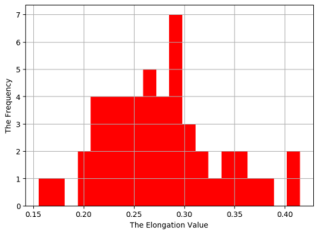
(b) By PCE model



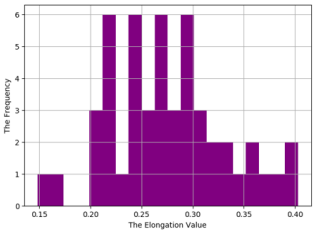
(c) By least square model



(d) By maximum likelihood



(e) By maximum likelihood and GPR



(f) By GPR package in Python

Figure 2: My complicated figure

Four surrogate models are used to predict the elongation value based on the training points, which are **Polynomial Chaos Expansion (PCE), Lease Square Model, Maximum Likelihood, Maximum Likelihood with GPR**. Specifically, to compare the performance of the GPR model programmed by ourselves with the current Python package, the package of **sklearn_gaussian_process** is also applied to the training points. The prediction results are visualized in the form of probability distribution, which can be seen as Fig.(2).

Furthermore, 10 points are sampled to validate the 5 trained models. The prediction results are shown in Fig.(3). It can be clear seen that the performance of all models are very well.

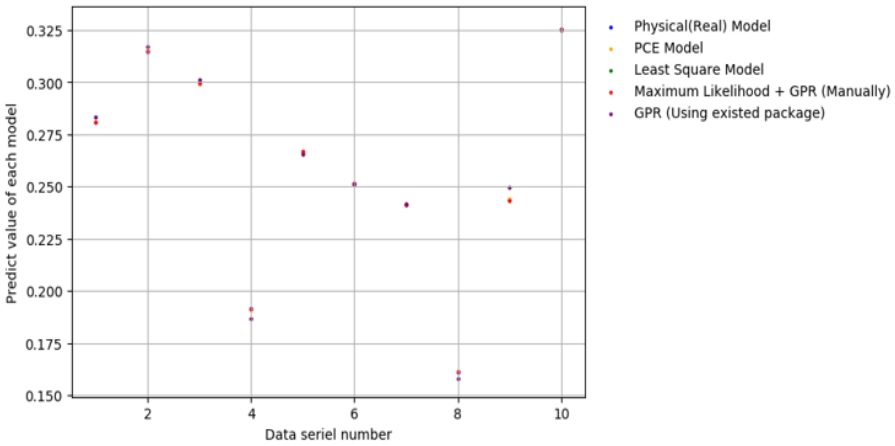


Figure 3: Prediction results for validation set

Besides, we also compare the performance of GPR model programmed ourselves with the performance of the existed GPR package in Python, the results are shown as follows:

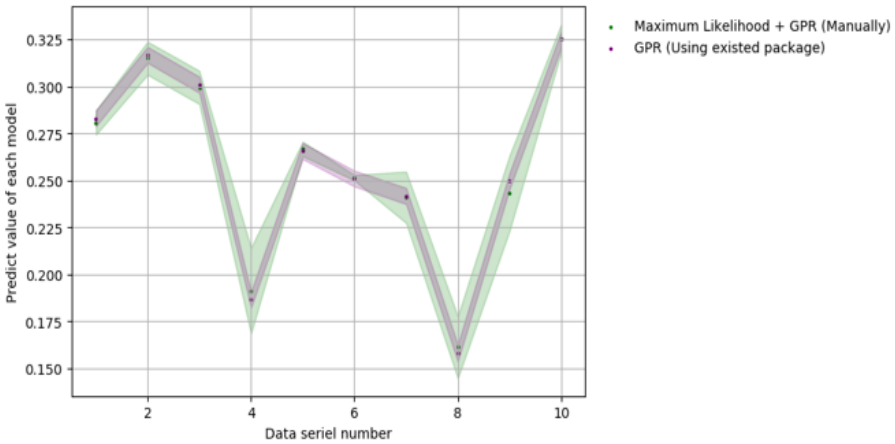


Figure 4: Prediction results for validation set