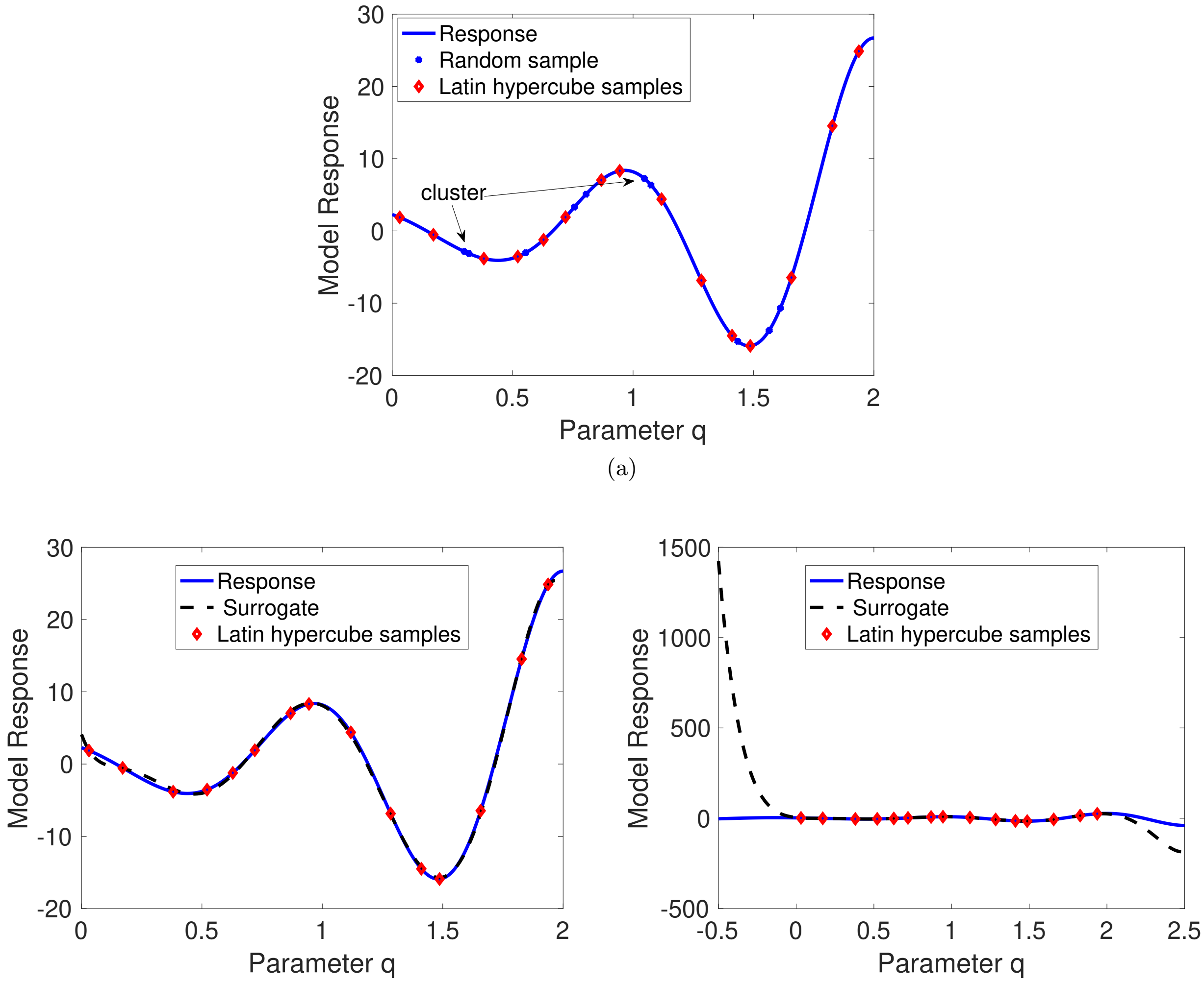
Math 540: Project 5

Tilekbek Zhoroev

1. Consider the function
$$f(q) = (6q^2 + 3)\sin(6q - 4)$$
for $q \in [0, 2]$. Plot the function and $M = 15$ random samples $q^m \sim \mathcal{U}(0, 2)$ and Latin hypercube samples generated using **lhsdesign.m**. Discuss the behavior of the two sets of samples based on the discussion in Section 15.3. For the Latin hypercube samples in **qdata.txt**, use regression to construct an $8^{th}-$order polynomial surrogate $f_s^K(q)$ and compare to $f(q)$ for $q \in [0, 2]$ and $q \in [-0.5, 2.5]$. Discuss the issues associated with using polynomial surrogates for extrapolatory or out-of-data predictions.

Solution:

We used the command **rand.m** to generate random samples from $\mathcal{U}(0, 1)$, which we then rescaled to the interval $(0, 2)$ using the transformation
$$a + (b - a)\eta, \text{ for } \eta \in (0, 1)$$
and $a = 0$, $b = 2$. Similarly, using the **lhsdesign.m** and transformation described above, we obtained Latin hypercube samples from $(0, 2)$. The plot of the given function, random samples, and Latin hypercube samples are presented in Figure 1(a). Random samples provide clusters of points, as seen in Figure 1(a), which can reduce the accuracy of surrogate models. On the other hand, Latin hypercube samples give a structure that assures that points fill the entire interval without clustering while being random in structural components, shown in Figure 1(a). Using Latin hypercube we obtain $\mathbf{y}$ where
$$\mathbf{y} = [y^1, ..., y^M]^\top, \; y^i = f(q^i) \quad i = 1, ..., M.$$
The coefficients, u_k, of the polynomial surrogate
$$f_s^K(q) = \sum_{k=0}^{K} u_k(q)^k$$
are computed by minimizing the least squares functional
$$\mathcal{J}(u) = \sum_{m=0}^{M}\left[y^m - \sum_{k=0}^{K} u_k(q^m)^k\right]^2 = ((\mathbf{y} - \mathbf{X}\mathbf{u})^\top(\mathbf{y} - \mathbf{X}\mathbf{u}))$$
and matrix
$$\mathbf{X} = \begin{bmatrix} 1 & q^1 & (q^1)^2 & \cdots & (q^1)^K \\ 1 & q^2 & (q^2)^2 & \cdots & (q^2)^K \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & q^M & (q^M)^2 & \cdots & (q^M)^K \end{bmatrix}$$
where $q^i, i = 1, ...M$ are Latin hypercube samples and $M = 15, K = 8$. Then, the optimal $\mathbf{u}$ for the least square functional is
$$\mathbf{u} = (\mathbf{X}^\top\mathbf{X})^{-\mathbf{1}}\mathbf{X}^\top\mathbf{y}$$
and we used $\mathbf{X}\backslash\mathbf{y}$ to solve on MATLAB. We plot the obtained polynomial surrogate for $q \in [0, 2]$ in Figure 1(b) and for $q \in [-.5, 2.5]$ in Figure 1(c). We observe that while the surrogate approximation is relatively accurate in the calibration zone, it is extremely wrong for extrapolatory or out-of-data predictions.



(a) (b) (c)

Figure 1: (a) Plot of the function, random samples and Latin hypercube samples; Function $f(q)$, training points (LHS), and 8^th-order polynomial surrogate for (b) $q \in [0, 2]$; (c) $q \in [-0.5, 2.5]$;.

2. Consider the spring model

$$\frac{d^2 z}{dt^2} + kz = 0$$
$$z(0) = 3, \frac{dz}{dt}(0) = 0,$$

which has the solution $z(t, k) = 3\cos(\sqrt{k} \cdot t)$. As in Example 16.15, we consider $k \sim \mathcal{U}(\bar{k} - \sigma_k, \bar{k} + \sigma_k)$ with $\bar{k} = 8.5$ and $\sigma_k = 0.001$. The model response is $f(t, k) = z(t, k)$. Construct a Legendre surrogate $f_s^K(t, \xi)$ using $K + 1 = 4$ basis functions $\{\psi_k(\xi)\}_{k=0}^K$ with with $\xi \in [-1, 1]$. Compare the mean and standard deviation

$$\mathbb{E}\left[f_s^K(t, \xi)\right] = u_0(t) \tag{1}$$

$$\sqrt{\text{var}[f_s^K(t, \xi)]} = \left[\sum_{k=1}^{K} u_k^2(t)\gamma_k\right]^{\frac{1}{2}} \tag{2}$$

with the Monte Carlo approximations

$$\mathbb{E}\left[f(t,k)\right] = \frac{1}{M}\sum_{m=1}^{M} f(t,k^m) \tag{3}$$

$$\sqrt{\mathrm{var}[f_s^K(t,\xi)]} = \left[\frac{1}{M-1}\sum_{m=1}^{M}[f(t,k^m) - \mathbb{E}\left[f(t,k)\right]]^2\right]^{\frac{1}{2}} \tag{4}$$

computed using $M = 10^5$ samples. You can use the $R = 10$ Legendre quadrature points and weights in the MATLAB code for Example 16.15 when computing the coefficients $u_k(t)$.

Solution:

Let define the function $g(\xi) : [0,1] \to [\bar{k} - \sigma_k, \bar{k} + \sigma_k]$,

$$g(\xi) = \frac{a+b}{2} + \frac{b-a}{2}\xi$$

where $a = \bar{k} - \sigma_k, b = \bar{k} + \sigma_k$. Then the Legendre polynomial surrogate of $f(t,k)$ is

$$f_s^K(t,k) = f_s^K(t,g(\xi)) = \sum_{k=0}^{K} u_k(t)\psi_k(\xi)$$

where $u_k(t)$ and $\psi_k(\xi)$ are weights and Legendre polynomials of order k. From discrete projections,

$$\begin{aligned}
u_k(t) &= \frac{1}{\gamma_k}\int_{-1}^{1} f(t,g(\xi))\psi_k(\xi)\rho(\xi)\,d\xi \\
&= \frac{1}{\gamma_k}\int_{-1}^{1} 3\cos(\sqrt{g(\xi)}\cdot t)\psi_k(\xi)\rho(\xi)\,d\xi \\
&\approx \frac{1}{\gamma_k}\left[\sum_{r=1}^{R} 3\cos(\sqrt{g(\xi_r)}\cdot t)\psi_k(\xi_r)\omega_r\right]
\end{aligned}$$

for k= 0,..,K, where ξ_r and ω_r are Legendre quadrature points and weights. The have given $K = 3$ and $R = 10$. Then using the equations (1), (2), (3), and (4) we obtained the mean and standard deviation using Legendre polynomial surrogate with discrete projection and Monte Carlo sampling, as shown in Figure 2. We observe that both computational methods give us same results.

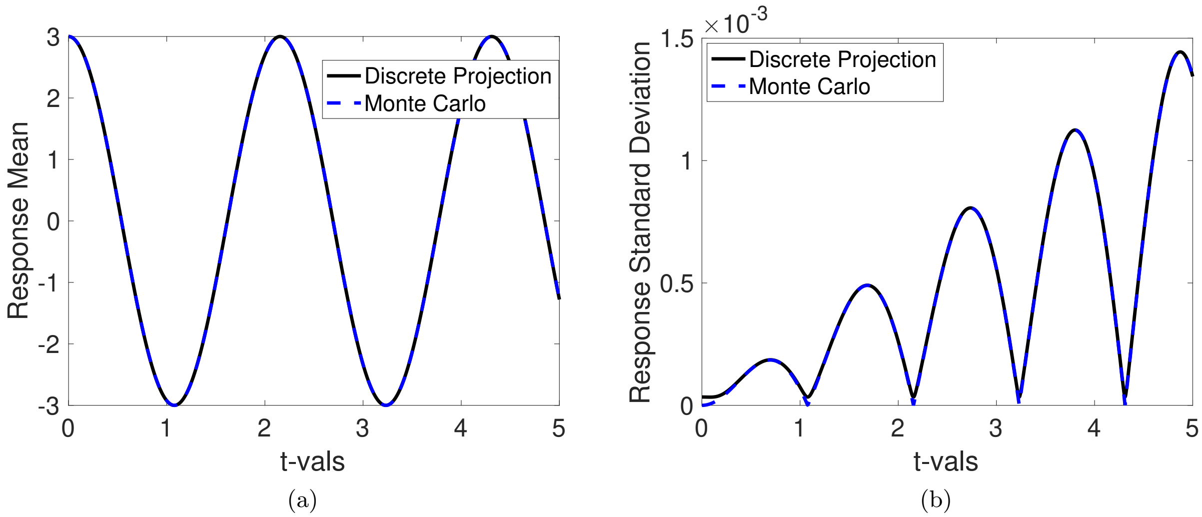


Figure 2: (a) Mean and (b) standard deviation computed using the Legendre surrogate with discrete projection and Monte Carlo sampling.

3. Consider the function

$$f(q) = (6q^2 + 3)\sin(6q - 4)$$

and the file **qdata.txt**, which contains $M = 15$ Latin hypercube samples q^m from $[0, 2]$. We employed this data to construct the polynomial surrogate in Exercise 1. Here you will employ **fitrgp.m** to construct a Gaussian process (GP) surrogate using the option for a squared exponential kernel. Plot the $M = 15$ training points, the GP mean, and GP prediction interval for $q \in [0, 2]$ and $q \in [-0.5, 2.5]$. Discuss the limitation of using a Gaussian process emulator for extrapolatory or out-of-data predictions.

Solution:

In this problem, we use squared exponential covariance kernel

$$c(q, q') = \sigma^2 e^{-(q-q')^2/2l^2}$$

for given function with constant mean function $\mu(q) = \mu_0$. Using given $M = 15$ data points and **fitrgp.m** we obtain the optimal values of the hyperparameters $\sigma = 16.9326, \mu_0 = 2.6996$ and $l = 0.3491$. Then using this optimal values we obtained the covariance function and prediction interval for $q \in [0, 2]$ and $q \in [-0.5, 2.5]$, as shown in Figure 3. We observe that Gaussian process accurate in training region and the out-of-data predictions are better than polynomial surrogate.

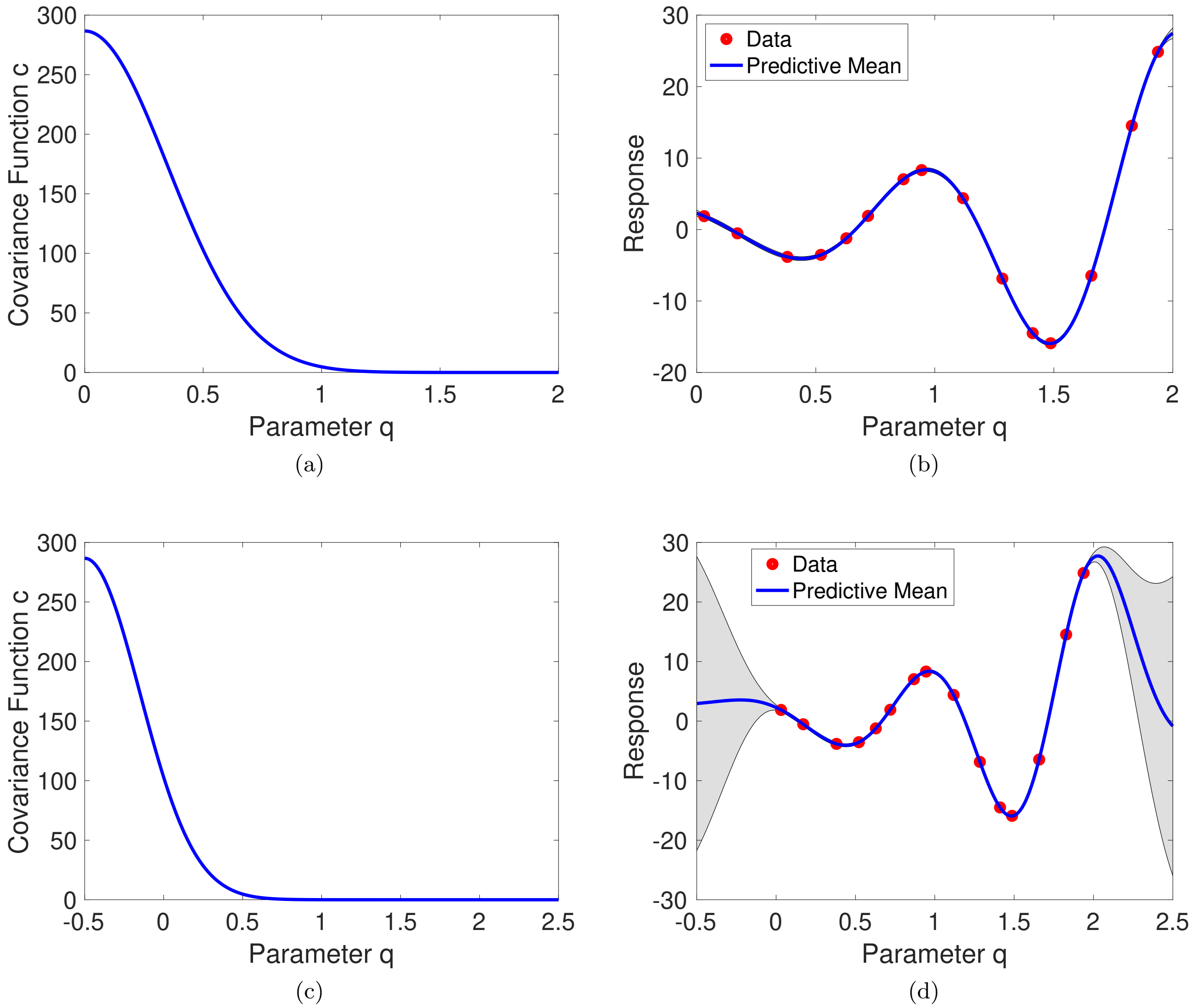


Figure 3: Covariance function for (a) $q \in [0, 2]$; (c) $q \in [-0.5, 2.5]$; Training data, mean, and 95% predictive distribution for future observations for (b) $q \in [0, 2]$; (d) $q \in [-0.5, 2.5]$;