Q1 (a)

Expand the log likelihood for the gaussian linear model with basis function ϕ:

$$L(w, \sigma^2) = -\frac{n}{2}\log \sigma^2 - \frac{n}{2}\log 2\pi - \frac{\sum_{i=1}^{n}(y_i - w^T\phi_i)^2}{2\sigma^2}$$

Optimisation is based on the two parameters w and σ^2. First focus on optimising w, by finding the w that minimises the negative log likelihood function L', setting the gradient to zero:

$$\frac{\partial L'}{\partial w} = 0$$

and writing ϕ, y in matrix notation ($\Phi, \boldsymbol{y}$), the optimal w^* is expressed in terms of $\Phi, \boldsymbol{y}$:

$$w^* = (\Phi^T\Phi)^{-1}\Phi^T\,\boldsymbol{y}$$

Note the gradient condition $\frac{\partial L}{\partial w} = 0$ is sufficient in the determination of a global minimum since the Hessian can be obtained as:

$$\nabla_w^2 L = \Phi^T\Phi$$

The Hessian $\nabla_w^2 L$ as a MxM matrix (since the design matrix is NxM) is positive semi-definite in this case, therefore a global minimum exists.

Now, this maximum likelihood solution is essentially the solution to a system of linear equations $Ax = b$, where $A = (\Phi^T\Phi)$, $b = \Phi^T\,\boldsymbol{y}$, x is w^*.

After w^* is obtained, the maximum likelihood value for the other parameter σ^2 can be expressed in terms of $\Phi, \boldsymbol{y}, w^*$:

$$\sigma^{2^*} = \frac{\sum_{i=1}^{n}(y_i - w^{*T}\phi_i)^2}{n}$$

Figures for (a), (b) and (c)

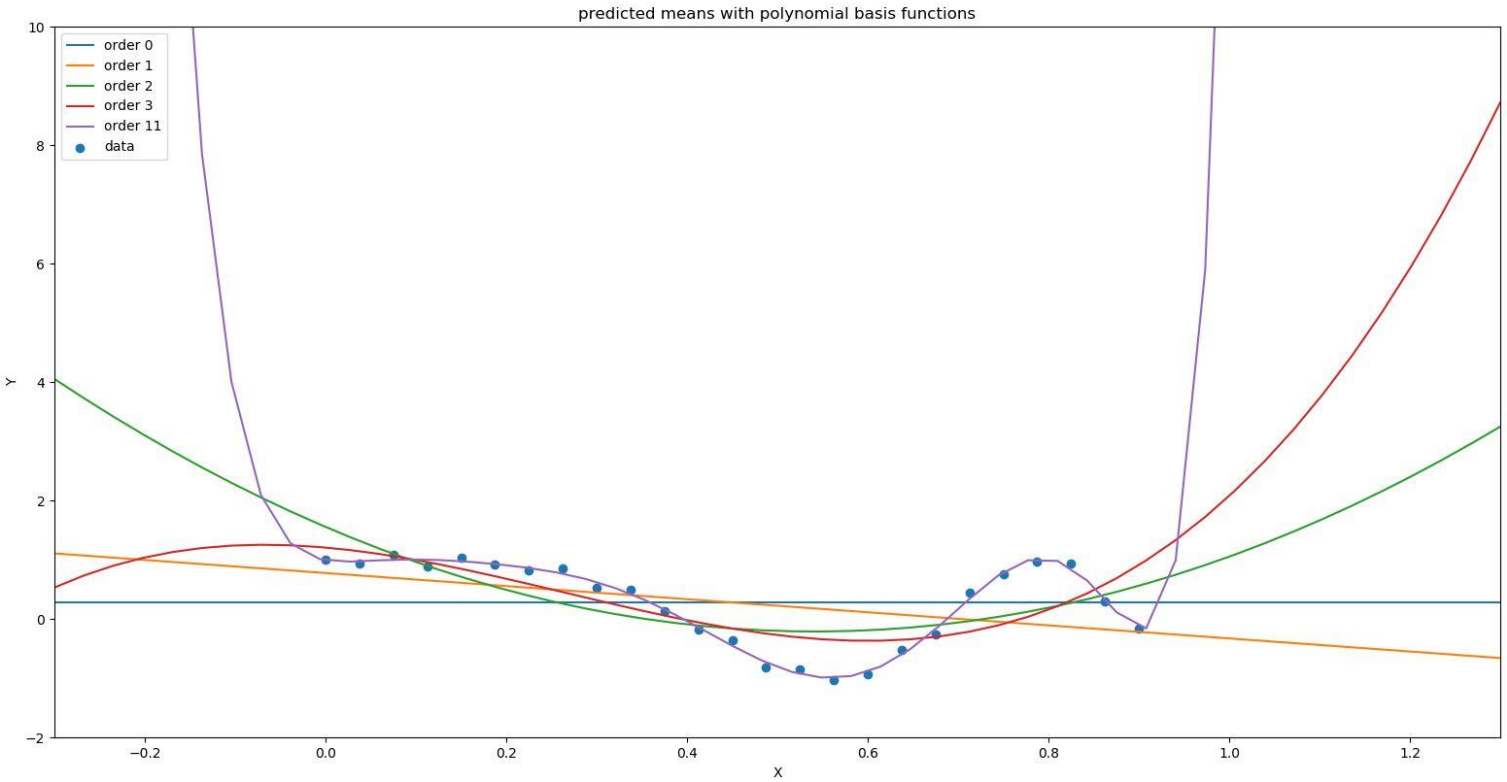

Figure 1. Plot for (a): predicted means with 5 polynomial basis function orders, from to -0.3 to 1.3, data is also shown as blue dots.

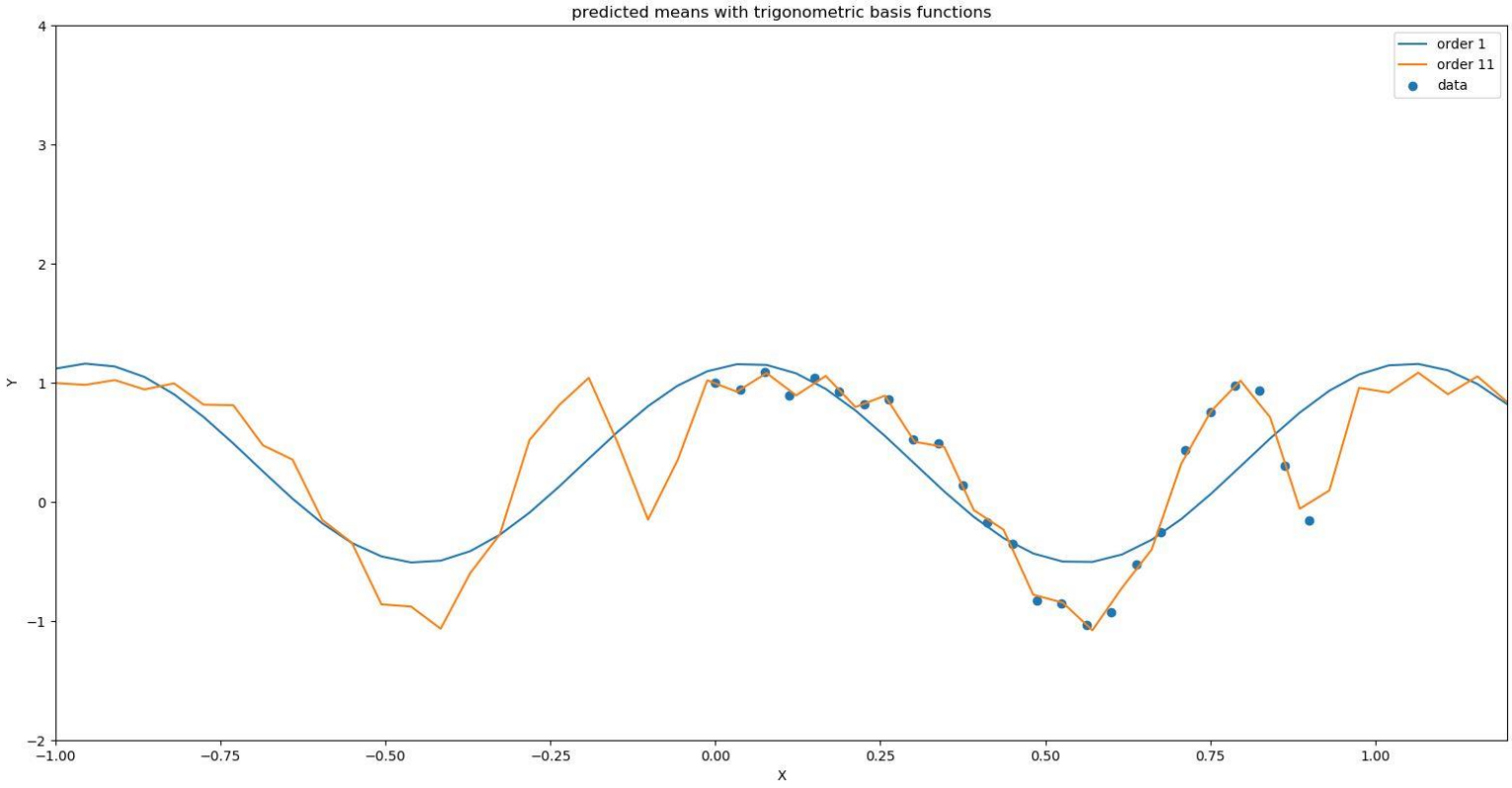

Figure 2. Plot for (b): predicted means with 2 trigonometric basis function orders, from to -1 to 1.2, data is also shown as blue dots.

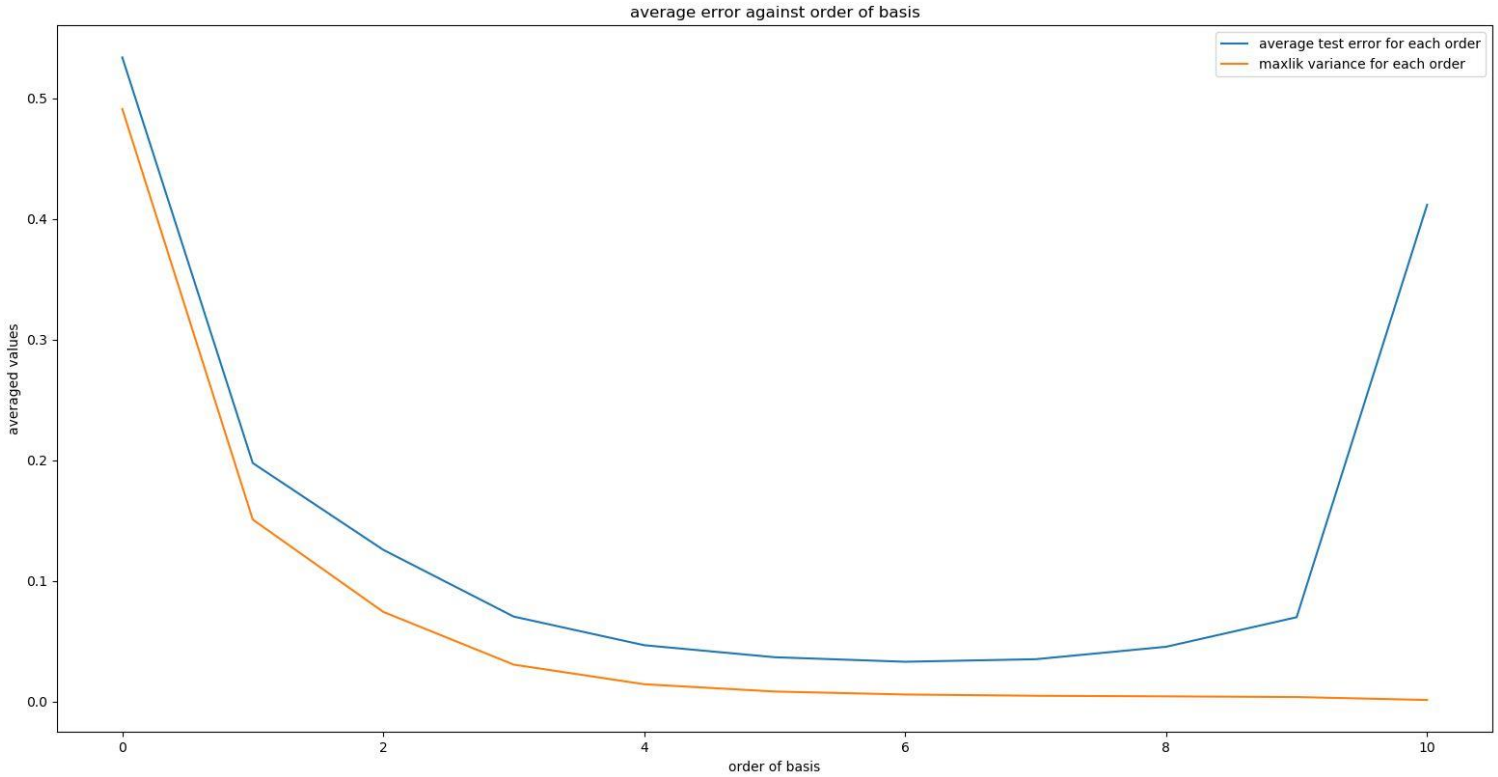

Figure 3. Plot for (c): average square errors against order of basis function, maximum likelihood variance for each order is also included.

Q1(d)

From the curves shown in both figure 1 and figure 2, it is clearly demonstrated that regardless of whichever basis function (polynomial or trigonometric), as the order of basis function increases to a sufficient level, the curve always tends to perfectly follow the scatter plot, which means it better fits the training data (in the case of first two questions the full data). An additional phenomenon observed is that when the curve "perfectly" fits the training data, dramatic trends and extreme oscillation arise between data points and outside training data range, as can be seen in figure 1 with the dramatic curve gradients when x is outside X data range (smaller than 0 and greater than 0.9), also can be seen in figure 2 with the zig-zagged profile. This implies that even though the training loss is minimal, the curve fitting may not be an optimal representation of the actual data.

In figure 3 average test error from cross validation is plotted alongside with maximum likelihood variance σ^{2^*} which has an expression:

$$\sigma^{2^*} = \frac{\sum_{i=1}^{n}(y_i - w^{*T}\phi_i)^2}{n}$$

With y_i being the data point being indexed by i from the training set, $w^{*T}\phi_i$ being the predicted mean indexed by i, the above expression suggests maximum likelihood variance is also mathematically the training loss in terms of MSE(mean squared error).

Now, from figure 3, it is seen that as order of basis increases, the training loss(variance) first rapidly drops, then steadily goes down to almost zero at an order of 10. Meanwhile test error (the blue line) initially drops rapidly, but then slowly reaches a steady level at around order=6, then begins to slowly climb back up again, and beyond order=9 a steep rise emerges in test error. This stark contrast between minimal training error and rising test error at high basis orders demonstrates an example of overfitting.

To investigate into the reasons, first consider a model defined by order M:

$$f_M(x_n; \boldsymbol{\theta}) = \sum_{m=1}^{M} x^m \theta_m$$

As the order increases, more functions can be represented by the model, since more terms with different orders are appended. The minimum(maximum likelihood) found with higher order models are guaranteed to be less than or equal to that found with lower order models, since the closed-form optimisation finds the exact function minimum, and simply more functions are available for higher order models. This explains the steady training error drop in figure 3, and the increased functional flexibility in representing data explains the more and more dramatic yet "perfect" curve fitting as order increases in figure 1 and 2 described previously. The performance on training data generally increases as order of basis function increases.

However, when evaluating model predictions with unseen test data(after obtaining model parameters with training data), the performance is significantly worse than that suggested by training loss(variance). This is the problem of overfitting.

The principle reason for overfitting in this case lies in the fact that the model is too complex to generalize well to unseen data due to excessive flexibility granted by unnecessarily raising order of basis functions in attempts to reduce training loss too much. This reason is also deeply tied to the theory of U-shaped bias-variance trade-off, which essentially suggests that a sweet spot exists in balancing underfitting and overfitting i.e. the model should be complex enough to pick out structures underpinning the data, but simple enough to avoid the danger of too much flexibility in fitting patterns. Inherent noise in training data that impairs model's ability to differentiate noise interference from signal could be another reason, although not likely for this case due to simple dataset. The lack of adequate amount of data to enable a model to be sufficiently generalizable could be a reason as well.

As the primary reason for overfitting is identified to be excessive model complexity(implying order of basis function being set too high), a primary mean to prevent overfitting can be established upon the goal of finding a balance in the selection of basis function orders. From figure 3 it can be observed that orders beyond 9 clearly overfits the data, an order in between 5 and 7 seems to be optimal as it achieves the smallest possible test error **and** training error. That is, in short, to choose a simpler model when model is too complex and overfits. To find out where such subtle balance lies, cross validation plays an vital role in spotting overfitting and providing a way to compare and determine the optimal strategy to be used. Increasing the amount of training data available can also be helpful, and various types of regularization and feature selection for data can be used to prevent overfitting.