CUED - Engineering Tripos Part IIB 2024-2025 **Module Coursework**

Module	4F13	Title of report	Coursework 1
Date submitted: 4/11/2024		Assessment for this module is ☑ 100% / ☐ 25% coursework of which this assignment forms _____ %	

UNDERGRADUATE and POST GRADUATE STUDENTS		
Candidate number:	5549F	☑ Undergraduate ☐ Post graduate

Feedback to the student

☐ **See also comments in the text**

		Very good	**Good**	Needs improvmt
CONTENT	**Completeness, quantity of content:** Has the report covered all aspects of the lab? Has the analysis been carried out thoroughly?			
	Correctness, quality of content Is the data correct? Is the analysis of the data correct? Are the conclusions correct?			
	Depth of understanding, quality of discussion Does the report show a good technical understanding? Have all the relevant conclusions been drawn?			
	Comments:			
PRESENTATION	**Attention to detail, typesetting and typographical errors** Is the report free of typographical errors? Are the figures/tables/references presented professionally?			
	Comments:			

Marker: Date:

IIB 4F13 Coursework 1

5549F

November 2024

All central commands for parts A-E can be found in the appendix. All values are reported in their natural domain.

1 A

After minimizing negative log marginal likelihood, we obtain the following hyper-parameters:

```
hyp.cov = [0.128  0.897]
hyp.lik = 0.118
```

0.128 corresponds to the characteristic length-scale, 0.897 to the signal standard deviation and 0.118 the noise standard deviation. The data points, prediction and 95% predictive error bars can be seen in figure 1. The log marginal likelihood $logZ_{|y} = -11.9$.

The characteristic length scale here can be interpreted as the distance required for function values to become uncorrelated along the x-axis. Here the small length scale suggests that function values can change quickly and as expected, we achieve an almost exact fit to the data when making our prediction.

With 0.897, it just means this was the σ_{signal} which best represented the amplitude of the signal from the given hyper-parameters starting point.

The small σ_{noise} suggests that the model is quite confident about the best-fitting function, and as expected we see small predictive error bars around region with many training data points.

As seen in figure 1, for data-points outside $x \in [-1.5, 2]$, the predictive error bars are much larger due to sparse data points. This is also anticipated.

2 B

To determine if there are other local optima, initialize the hyper-parameters at different values. Some other local optima as seen in figure 2

In figure 2a, the large characteristic length-scale highlights the average downward trend of the data for the given input data range, x ∈ [-3,3]. However, the marginal likelihood is low because the predictive line does not go through/fit the data very much. The existence of this GP as a local optima might have come from reduced complexity penalty. Figure 2c has a similar explanation to this. In figure 2b, the predictive looks very complicated and changes very rapidly, reflected by the extremely small characteristic length-scale. The low marginal-likelihood is due to the complexity penalty. The trained σ_{noise} is also extremely small, further supporting that this local optima is leaning on a GP which overfits. Figure 2d is similar to figure 1. This could just be another local optimal which has the increase/decrease of data fit and complexity penalty offsetting each other in the marginal likelihood.

Figure 1 remains the best fit out of these 5 local optimum since it has the highest marginal likelihood. It also seems to have a reasonable predictive error bar which bounds all data points but remains uncertain in regions with sparse data.

3 C

Initializing with $log\theta$ = [-1 -0.2 0 3] we get figure 3

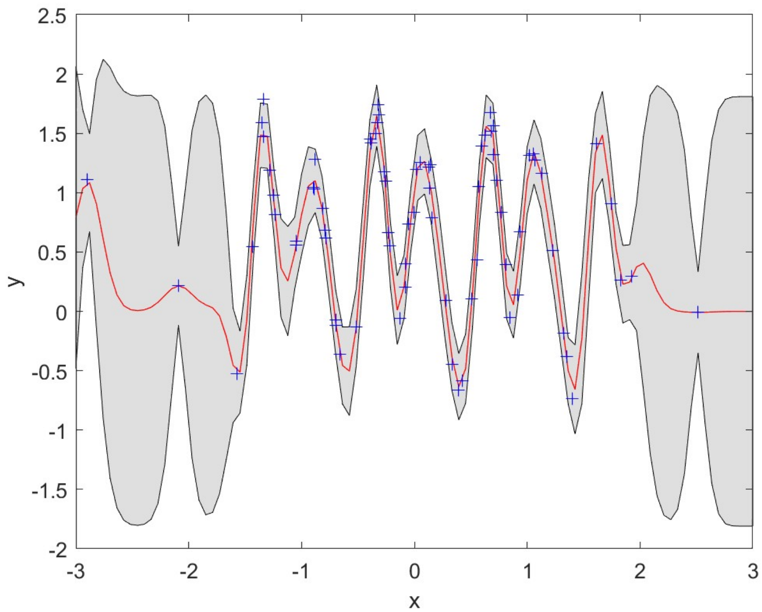
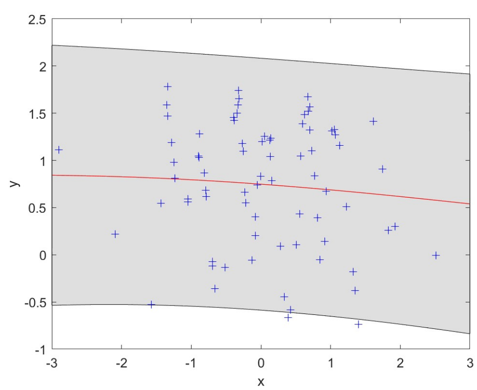
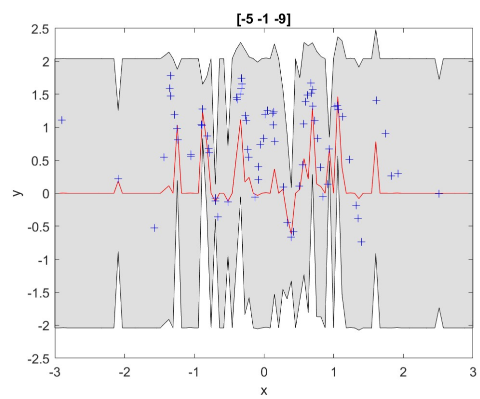


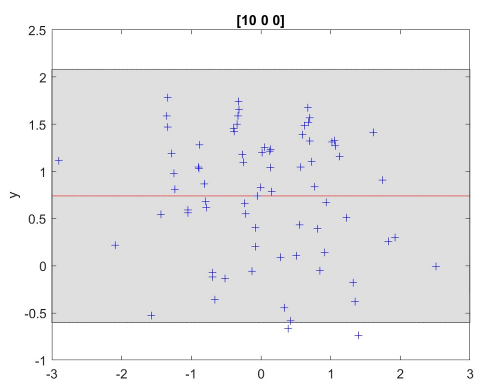
Figure 1: θ =[0.128 0.897 0.118] $logZ_{|y}$=-11.9



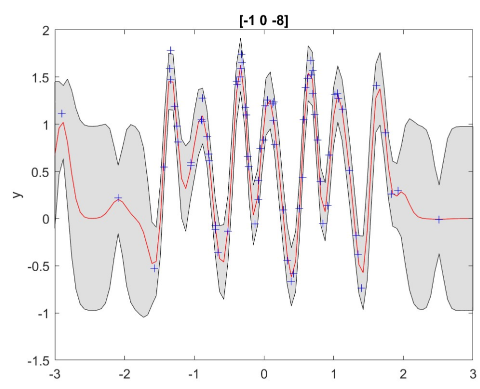
(a) θ=[8.04 0.696 0.663] $logZ_{|y}$=-78.2



(b) θ =[$5.18e^{-3}$ 1.02 $1.23e^{-4}$] $logZ_{|y}$=-99.7



(c) θ =[22030 0.744 0.667] $logZ_{|y}$=-78.3



(d) θ =[0.112 0.470 0.130] $logZ_{|y}$=-25.8

Figure 2: GP trained at various hyper-parameter local optima

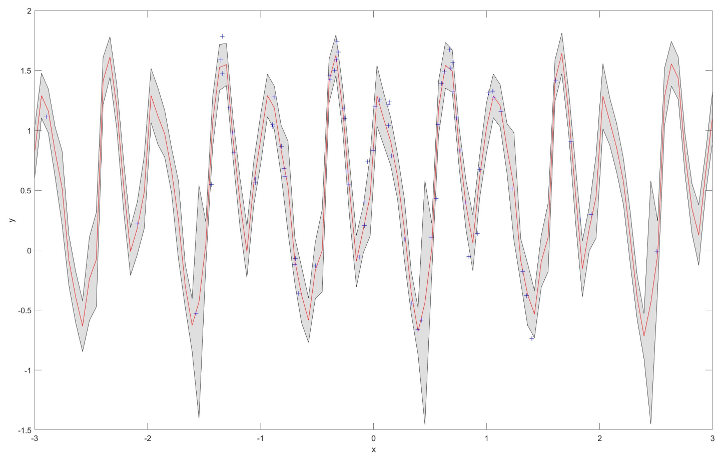


Figure 3: [0.195 0.998 1.62 0.0755] $logZ_{|y}$=-5.618

The marginal-likelihood is significantly higher with a periodic covariance function than a squared exponential covariance function, therefore, the data generating mechanism definitely has a periodic component. The 95% predictive error bars are much narrower, suggesting the model gained a lot more information about points well beyond $x \in [-1.5, 2]$, just from knowing there exist periodicity in the generating mechanism.

However, the smooth-fitting downward trend we saw in figure 2a suggest the covariance function could be a mix of both covariance functions we have trained with so far.

4 D

We let the covariance function composing of the product of periodic covariance function (covPeriodic) and squared exponential covariance function (covSEiso) to have hyper-parameters:

$[l_{periodic}\ \sigma_{periodic}\ p\ l_{SEiso}\ \sigma_{SEiso}] = [0.6065\ 1.0000\ 1.0000\ 7.3891\ 1.0000]$

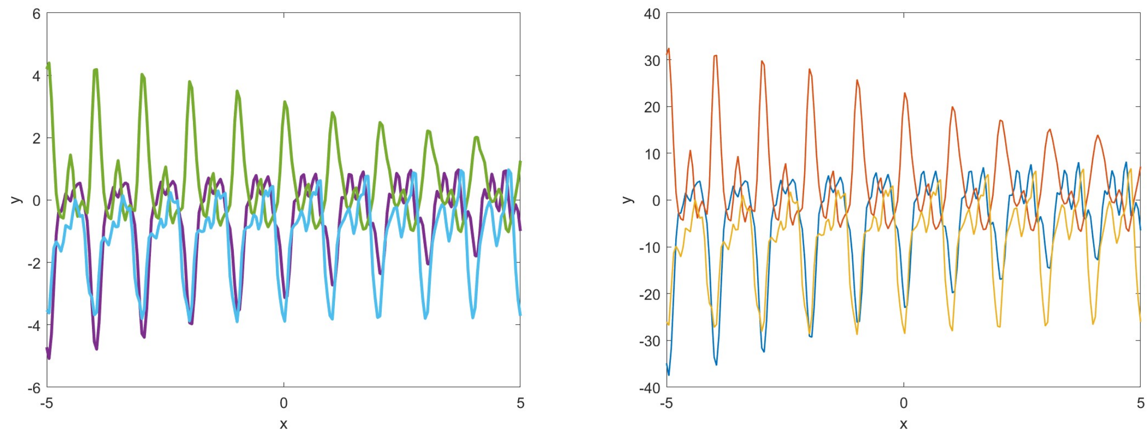
We then sample random noise free functions in figure 4a. In figure 4b we increase $\sigma_{periodic}$ and σ_{SEiso} and observe greater signal amplitude. For the periodic covariance function, we can see that the hyperparameter p gives all sample functions a period of 1. The small magnitude of $l_{periodic}$ gives rise to the rapid variation within p.

In figure 5, we can see that the function has a much larger envelope period with order of magnitude matching l_{SEiso}.

5 E

Figure 6 visualises the data from cw1e.mat.

When trained with just Squared Exponential covariance function with Automatic Relevance Determination (ARD) distance measure. The hyper-parameters obtained after minimizing the negative log marginal likelihood is
$\theta = [1.5116\ 1.2859\ 1.1073\ 0.1026]$ with $logZ_{|y} = 19.2$



(a) θ=[0.6065 1.0000 1.0000 7.3891 1.0000]

(b) θ=[0.6065 1.0000 2.7183 7.3891 2.7183]

Figure 4: 3 Sample functions from a GP with composite covariance function of covPeriodic x covSEiso

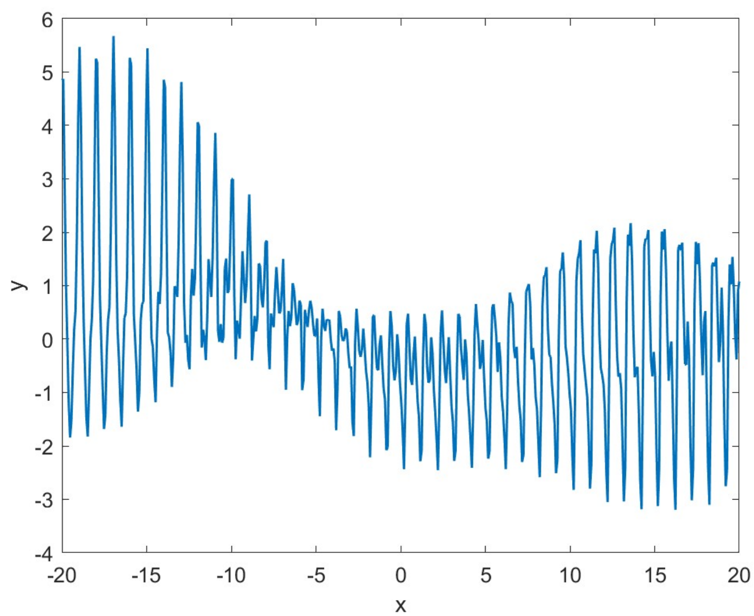


Figure 5: 1 Sample function from a GP with composite covariance function over range of x∈[-20,20]

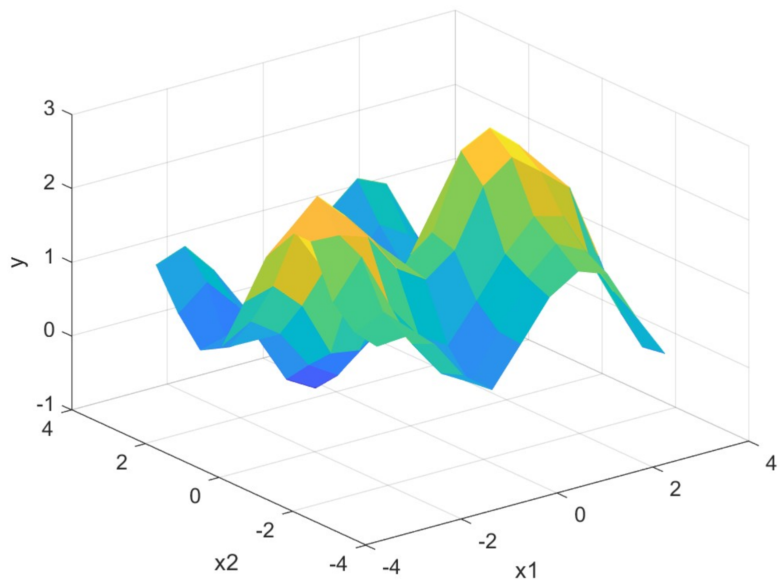
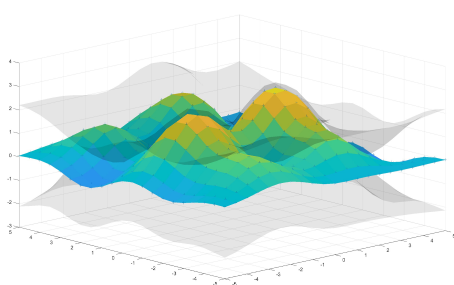
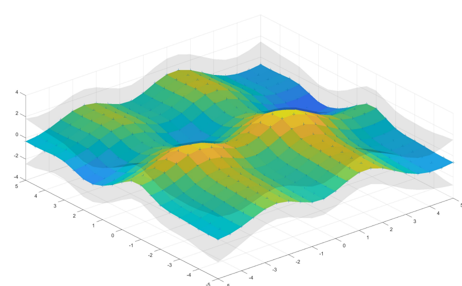


Figure 6: Data cw1e.mat



(a) GP trained with covSEard covariance



(b) GP trained with covariance sum of 2 covSEard

Training with composite covariance by summing 2 covSEard gives figure 7b
θ= [1.4322 449.7917 1.0792 312.4118 0.9889 0.7093] with $logZ_{|y} = 66.3$

Judging by the huge increase in marginal-likelihood, the GP model with composite covariance function provides a much better data-fit than the model in figure 7a. In the marginal-likelihood equation, there is a penalty term from model complexity. Despite the composite model having more parameters, marginal-likelihood still increased drastically thanks to the data-fit. Therefore, the composite model is the best in this case.

A Matlab Code: A, B, C (replace '@covSEiso' with '@covPeriodic'

```
load('cw1a.mat')
xs = linspace(-3, 3, 100)';                    % 61 test inputs

meanfunc = [];                    % empty: don't use a mean function
covfunc = @covSEiso;              % Squared Exponental covariance function
likfunc = @likGauss;              % Gaussian likelihood


hyp = struct('mean', [], 'cov', [-1 0], 'lik', 0);

hyp2 = minimize(hyp, @gp, -100, @infGaussLik, meanfunc, covfunc, likfunc, x, y);


[mu s2] = gp(hyp2, @infGaussLik, meanfunc, covfunc, likfunc, x, y, xs);

f = [mu+2*sqrt(s2); flipdim(mu-2*sqrt(s2),1)];
fill([xs; flipdim(xs,1)], f, [7 7 7]/8)
hold on; plot(xs, mu,'color','r'); plot(x, y, '+', 'color','b')
```

B Matlab Code: D

```
meanfunc = [];
hyp.mean = [];
covfunc = {@covProd, {@covPeriodic, @covSEiso}};
hyp.cov = [-0.5 0 0 2 0];
likfunc = @likGauss;
hyp.lik = 0;

x= linspace(-5,5,200)';
K = feval(covfunc{:}, hyp.cov, x);

for i = 1:3
    y = chol(K+1e-6*eye(200))'*gpml_randn(i, 200, 1, 'LineWidth', 2);
    plot(x, y)
    hold on
end
```

C Matlab Code: E

```
count=1;
xs(441,2)=0;
for x1 = -5:0.5:5
    for x2 = -5:0.5:5
        xs(count,1)=x1;
        xs(count,2)=x2;
        count = count +1;
    end
```

```
end

meanfunc = [];
covfunc = {@covSum, {@covSEard, @covSEard}};
likfunc = @likGauss;

hyp = struct('mean', [], 'cov', 0.1*randn(6,1), 'lik', 0);

hyp2 = minimize(hyp, @gp, -100, @infGaussLik, meanfunc, covfunc, likfunc, x, y);

[nlZ dnlZ] = gp(hyp2, @infGaussLik, meanfunc, covfunc, likfunc, x, y);
[mu s] = gp(hyp2, @infGaussLik, meanfunc, covfunc, likfunc, x, y, xs);

mesh(reshape(xs(:,1),21,21),reshape(xs(:,2),21,21),reshape(mu,21,21),'FaceColor','flat');
hold on;
scatter3(xs(:,1),xs(:,2),mu, '+');
hold on;
surf(reshape(xs(:,1),21,21),reshape(xs(:,2),21,21),reshape(mu+2*sqrt(s),21,21),
'FaceAlpha','0.2','EdgeColor' , 'blue','FaceColor','k');
surf(reshape(xs(:,1),21,21),reshape(xs(:,2),21,21),reshape(mu-2*sqrt(s),21,21),
'FaceAlpha','0.2','EdgeColor' , 'blue','FaceColor','k');
```