Franke's Function Emulation

Scott Phillips

Contents

1 Introduction

This report will use emulation to analyse Franke's function. Franke's function has been chosen because it is comprised of two Gaussian peaks with different heights and therefore has rich properties and provides interesting emulation for the various aims to be discussed. The methods implemented in this report will be to construct a 1 and 2 dimensional emulator, for which various parameter sets are chosen to inspect how emulators are affected by the prior parameters. All subsequent analysis will be conducted on a 2 dimensional emulator. This subsequent analysis will be comprised of history matching and Bayesian optimisation based on a history matching framework. Unfortunately, there are no practical implications to discuss here as there are in other physical models. This is because after some initial exploration of other functions, the Franke function was the most interesting despite not being a physical function due to it being non-linear and consisting of Gaussian functions. Franke's function has some important properties, namely it has 2 inputs and their respective domains are the interval [0,1]. It takes the form:

$$\begin{aligned} f(\boldsymbol{x}) = {} & 0.75\exp(-\frac{(9x_1-2)^2}{4} - \frac{(9x_2-2)^2}{4}) + 0.75\exp(-\frac{(9x_1+1)^2}{49} - \frac{(9x_2+1)}{10}) \\ & +0.5\exp(-\frac{(9x_1-7)^2}{4} - \frac{(9x_2-3)^2}{4}) - 0.2\exp(-(9x_1-4)^2 - (9x_2-7)^2) \end{aligned}$$

2 1-Dimensional Emulator

2.1 Methodology

In this section a 1-Dimensional Emulator is constructed, this is done via keeping the x_2 variable fixed. The fixed value is chosen to give the most interesting emulator, as such $x_2 = 0.3$. Now within the emulator, in order to showcase how an emulator works and is effected, various parameter sets are chosen to illustrate this. A 1-Dimensional emulator is dependent on 3 prior parameters, σ, θ, and the expectation of the function. θ represents correlation and takes values in the domain [0,1], σ represents variance and also takes values in the domain [0,1]. The expected value can take any rational number, but if there is little prior knowledge, a safe bet is 0. It is also insightful to look at how the emulator is affected by prior knowledge on how many values the actual function is known for or can be simulated for, all of this is explored below.

2.2 Implementation and Results

Now, implementing the methods stated in the previous subsection, starting with a 5 run emulator and a 7 run emulator over the points $x_1 = (0, 0.25, 0.5, 0.75, 1)$ and $x_1 = (0, 0.125, 0.25, 0.5, 0.75, 0.875, 1)$ respectively. These can be seen in Figure 1 where it is clear the uncertainty of the emulator is vastly reduced with increased runs.

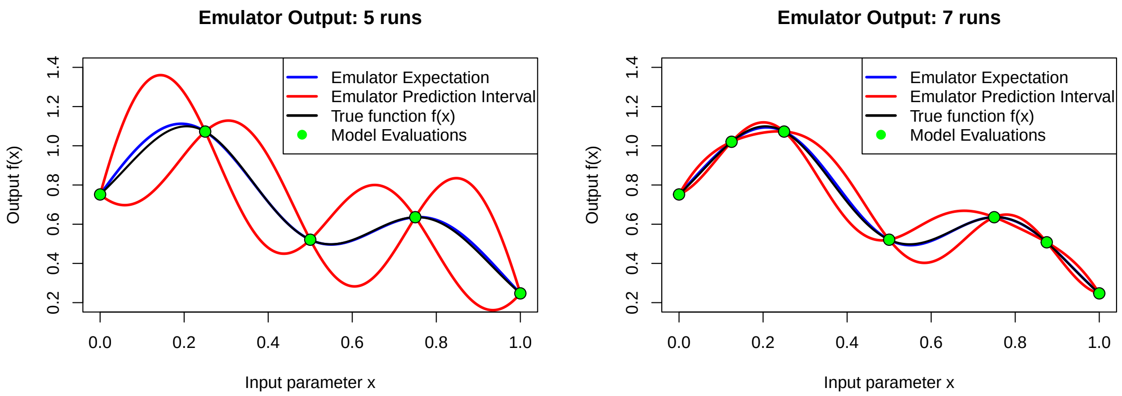


Figure 1: Emulator simulation for 5 and 7 known values

Implementing the change in parameter θ through the range of values $\theta = (0.01, 0.25, 0.5, 0.9)$ can be seen in Figure 2. It is immediately apparent that as theta increases, the correlation between data points increases as the uncertainty in the emulators expectation decreases.

Finally, implementing the change in parameter σ through the values $\sigma = (0.25, 0.75)$. Figure 2 shows that there is a clear linear relationship between uncertainty and variance as you would expect based on the uncertainty intervals being derived from.

$$E_D f(x) \pm 3\sqrt{Var_D f(x)}$$

Less values for σ are used here because it is not as interesting a result as varying θ. This concludes our 1-Dimensional emulator.

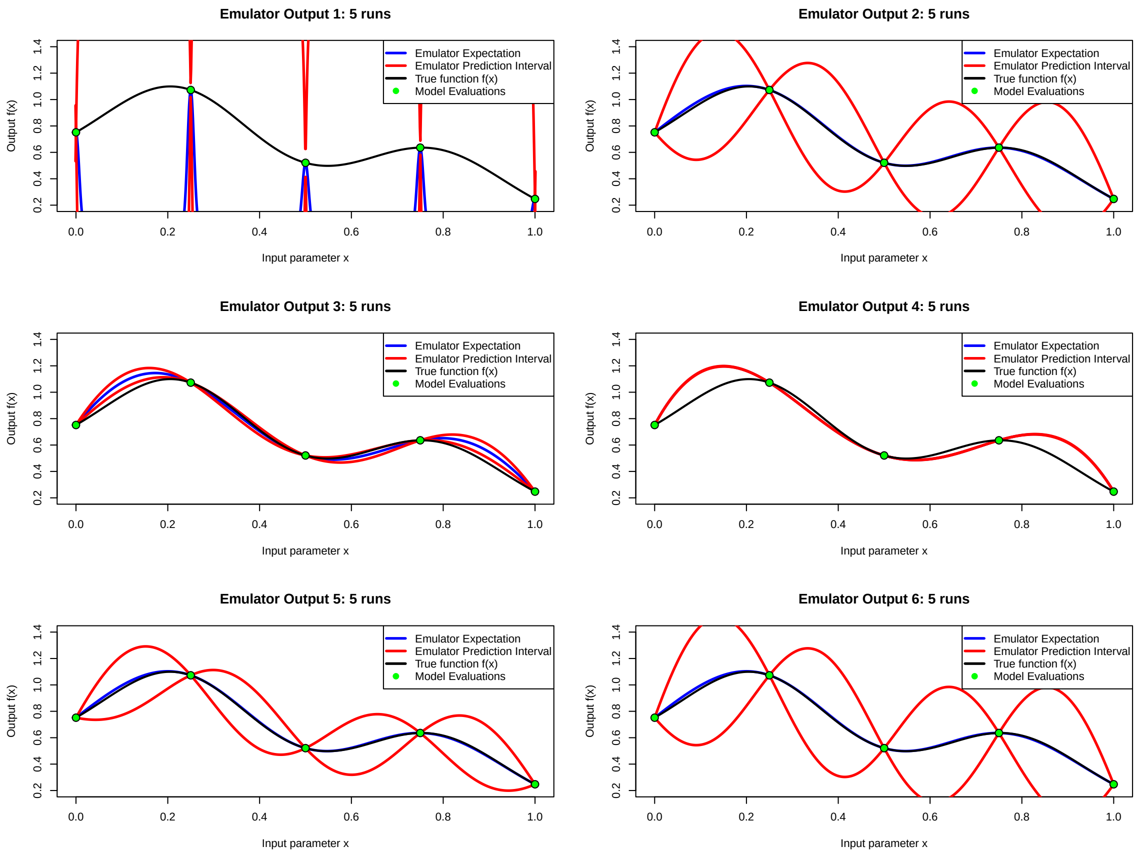


Figure 2: Emulator simulation for plots with prior values $\theta = 0.25$, $\sigma = 0.5$, $E_f = 0$, first 2 rows represent θ change, last row represents σ change

3 2-Dimensional Emulator

3.1 Methodology

The 2-Dimensional application of the emulator is very similar to the 1-Dimensional application. This section will explore ways of choosing prior runs for the emulator function to display the best results and the corresponding emulator expectation, variance, and diagnostics. Emulator diagnostics are introduced in this report here due to the rest of the analysis being done from the 2-Dimensional viewpoint. Emulator diagnostics are defined by:

$$S_D f(x) = \frac{f(x) - E_D[f(x)]}{\sqrt{Var_D[f(x)]}}$$

Where it can be shown that any values beyond 3 and -3 indicate a large amount of uncertainty due to the 3-sigma rule and will show up as white space on contour plots, suggesting the emulator is not viable. It is useful to show emulator diagnostics as it aids in the identification of any unexpected observations.

Moving onto the way of defining a grid of prior runs for the emulator to base itself off, the Latin Hypercube Design with a maxi-min approach is employed to give the best spread of data points so that the exploration of the emulator is done as efficiently as possible and to minimise uncertainty. Efficiency is essential for emulators as their purpose is to reduce computation time.

3.2 Implementation and Results

In Figure 3, the emulator is shown for a symmetric grid of 16 prior runs with plots representing expectation, variance, diagnostics, and the true model based off Franke's function. It is clear to see that the emulator's expectation does a poor job of simulating the true model, missing the second Gaussian peak and the dip.

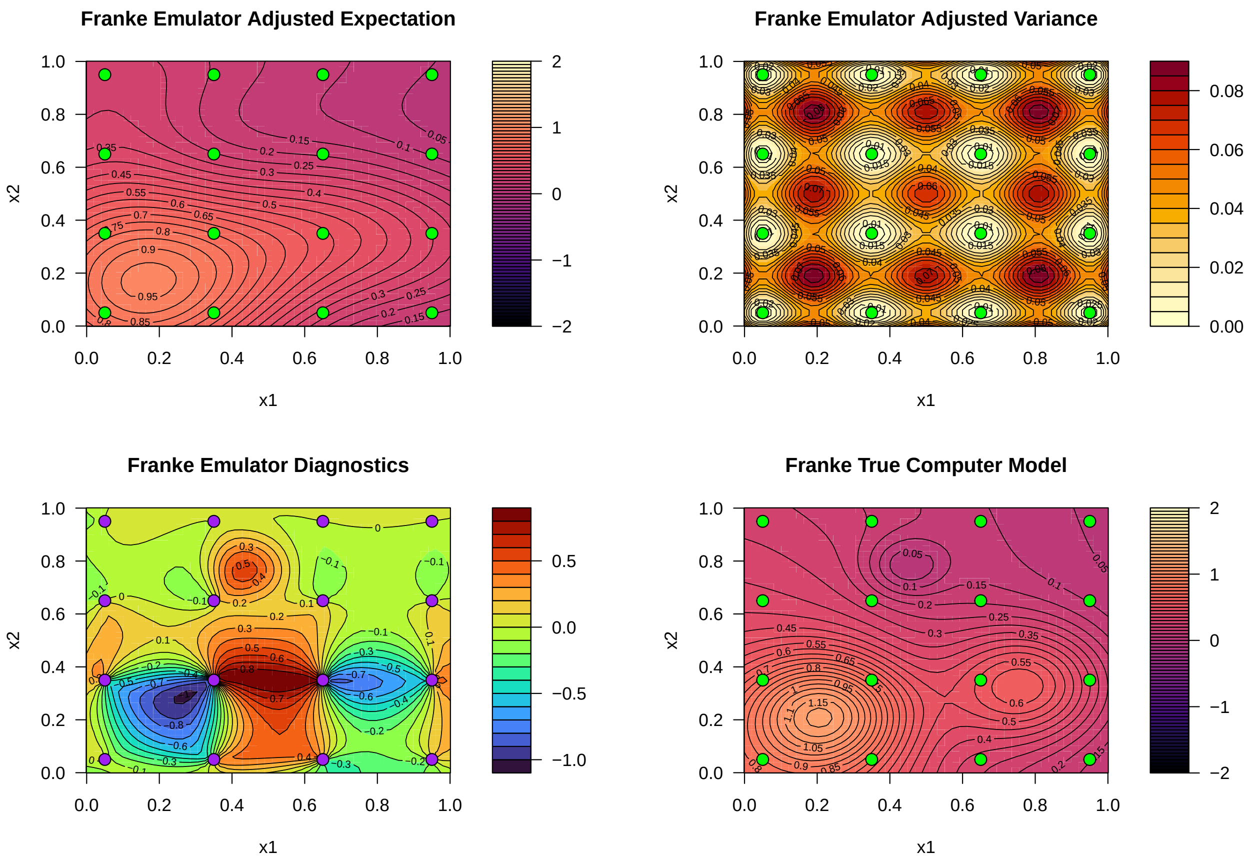


Figure 3: Emulator plots for symmetric grid design, $\theta = 0.35$, $\sigma = 1$, $E_f = 0$

In Figure 4, a LHD has been employed with the points shown in yellow and the same plots repeated, notice the increased similarity between the LHD's expectation and the true model, it acknowledges the second peak and is starting to form contours around the dip. The main downside is that the perimeter of the plot has increased variance, however, the variance in the centre of the model has vastly decreased. The diagnostic plots do not clearly show which model is better, but this combined with the variance plots indicates that the LHD design does provide better exploration of the functions domain.

To finish off the introduction to 2-dimensional emulators, an inspection of how the emulator's expectation is effected by the correlation parameter is performed. Figures 11-16 intuitively show how the points have a radius of effect on the points around them, dependent on the correlation parameter, and as the parameter increases, the radius of effect increases. It is important to not let the parameter be too large as this can lead to over-fitting. This concludes the introductory discussion of 2-D emulators.

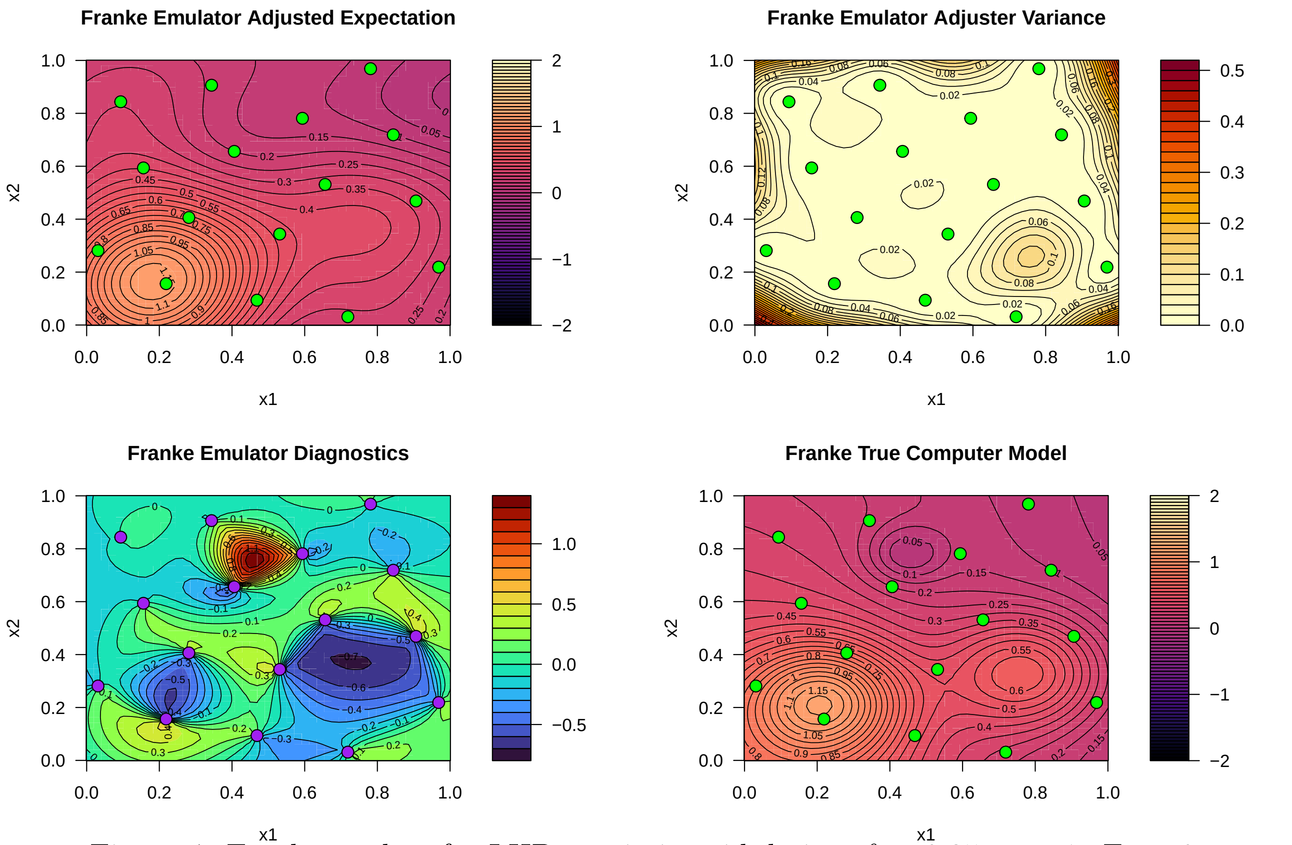
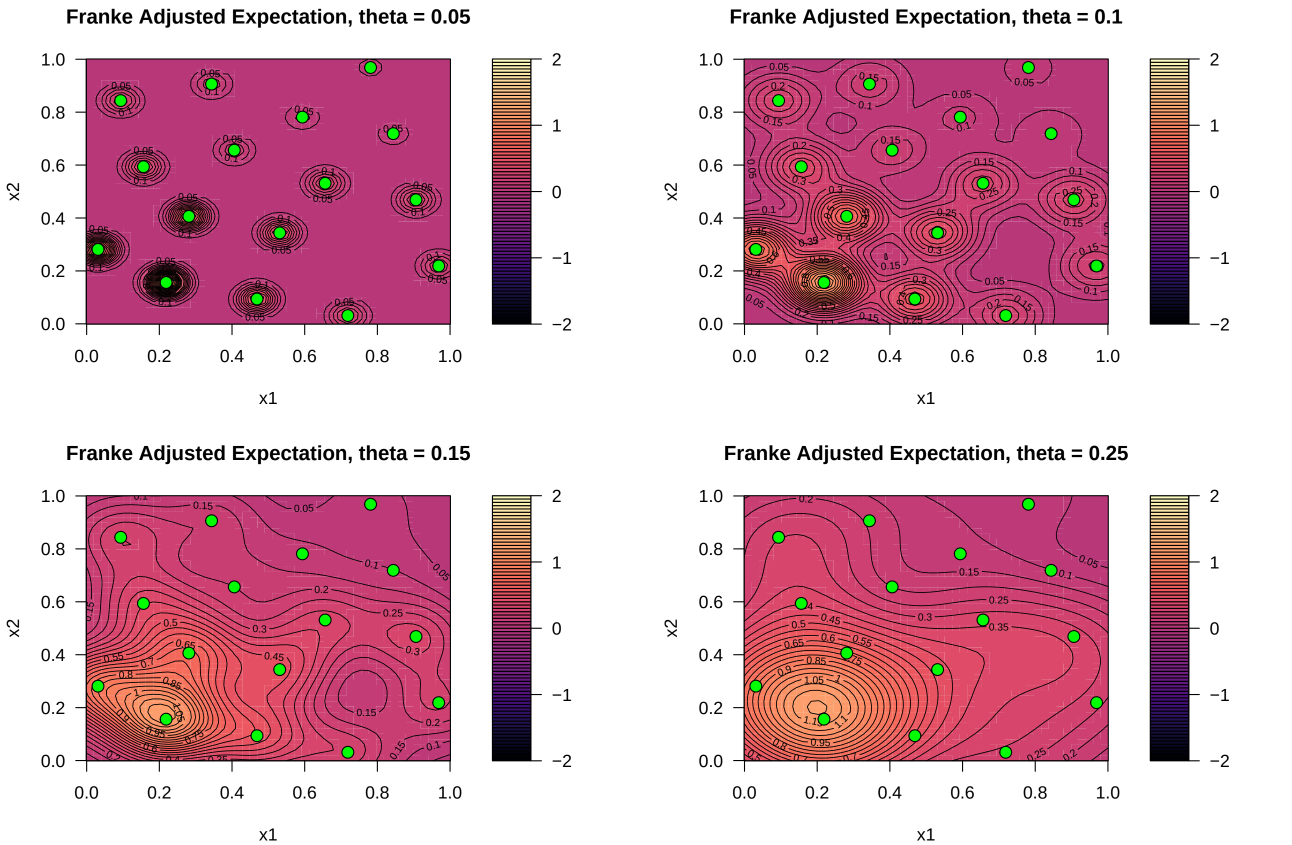


Figure 4: Emulator plots for LHD maximin grid design, $\theta = 0.35$, $\sigma = 1$, $E_f = 0$



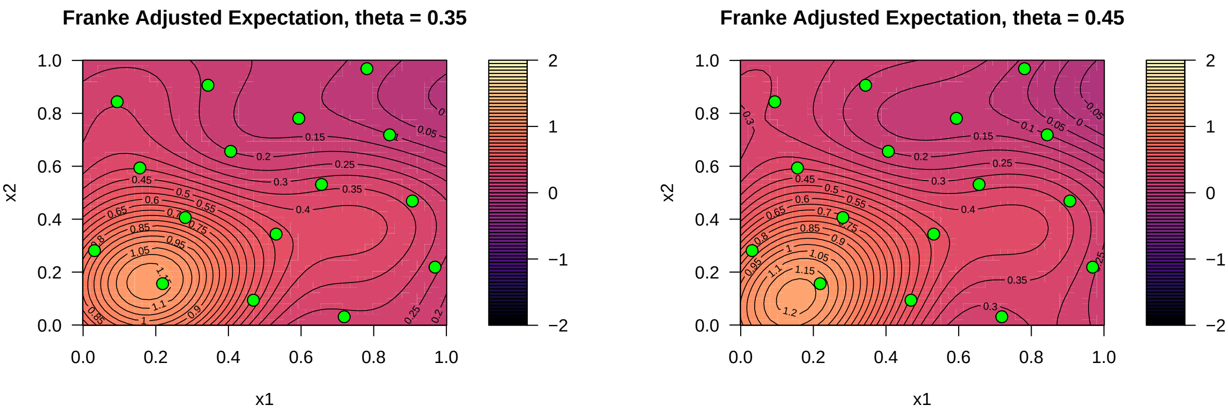


Figure 5: Plots for LHD maximin grid design with $\sigma = 1$, $E_f = 0$

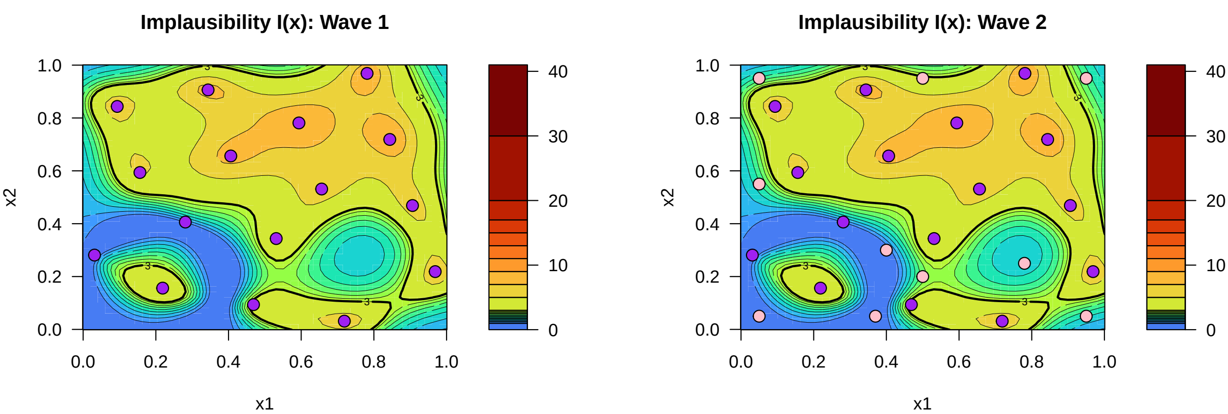
4 History Matching

4.1 Methodology

In this section, 2-D emulation is furthered and history matching is used to solve the task of given an observed value, where does the emulator state its corresponding inputs would be. This is a very applicable part of emulation and involves a multi-wave approach. History matching utilises implausibility to show the region of the domain for which the inputs correspond to the observed value. After finding these regions, the emulator must undergo a second wave of runs to ensure the output is accurate within the emulators capabilities. It is also important to note that, due to dealing with physical observations, there are now errors to account for within the observation of the data and the model itself which will be implemented into the emulator when finding the non-implausible regions. The main concern with multi-wave methods is the computational burden, it is important to find a balance between computational cost and accuracy of results.

4.2 Implementation and Results

To start the analysis, an initial run is performed using a LHD of 16 points at an observed value of 0.8 with error values for observation and model uncertainty given by 0.05 each. This is shown in Figure 6 with the corresponding implausibility regions that show which values within the emulator are non-implausible and as shown, a second wave of runs is done. The new points are represented by pink dots in this region to decrease the uncertainty of the history matching output and a final plot of the true function is applied to show how accurate the emulator is.



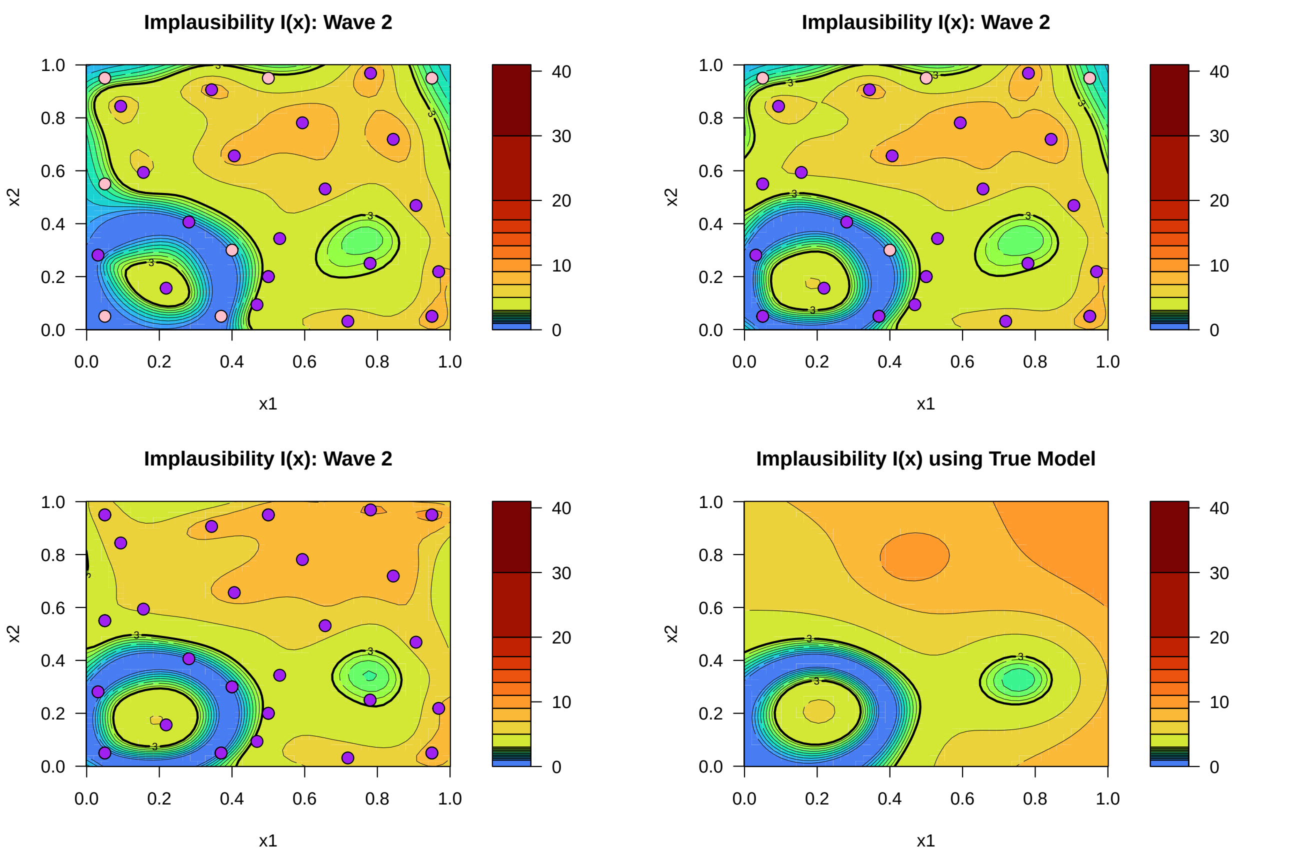
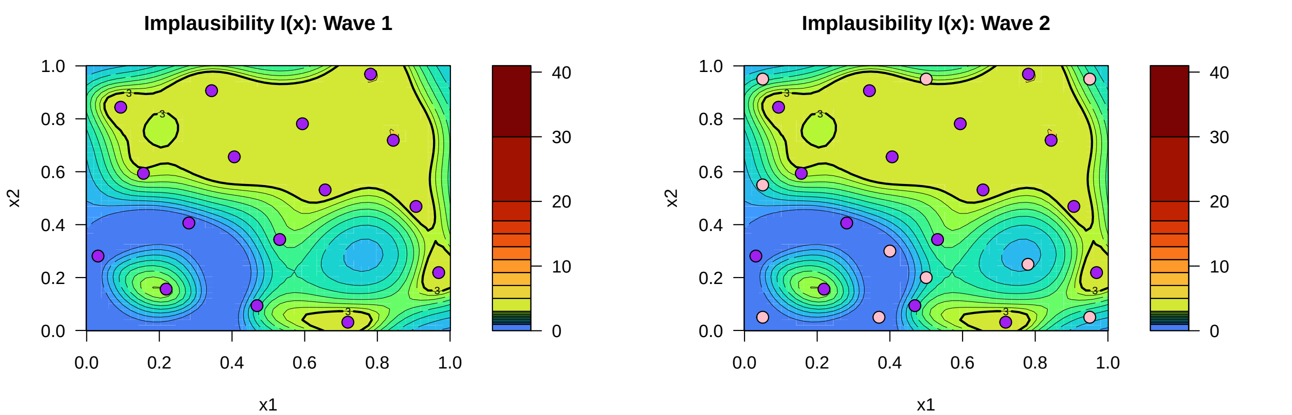


Figure 6: History Matching with initial LHD maximin grid design, errors = 0.05

Now, repeating this analysis with observation and model uncertainty equivalent to 0.1 to showcase that as uncertainty increases, the region of not implausible inputs increases in both the emulator and the true model to accommodate the increased uncertainty. This is shown in Figure 7, where new runs are introduced again by pink dots and this concludes the discussion of history matching.



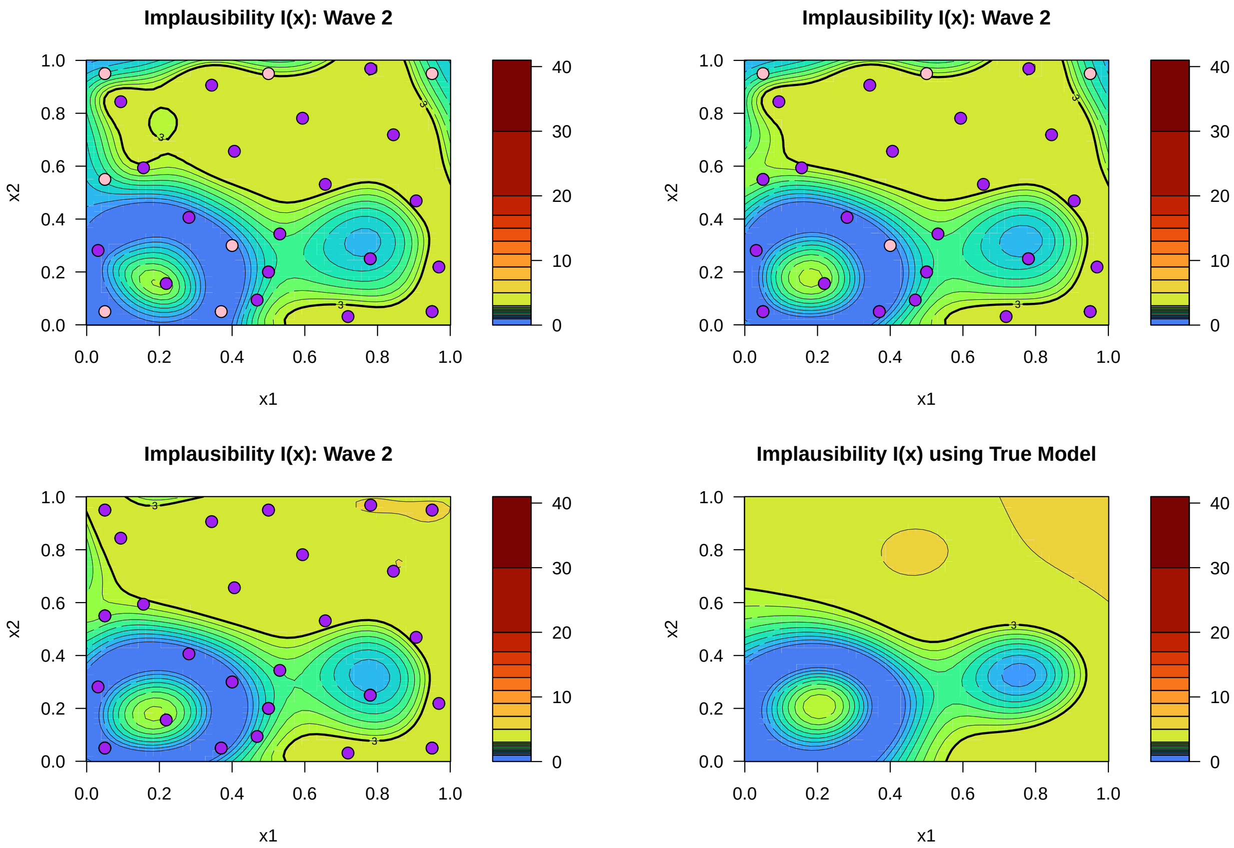


Figure 7: History Matching with initial LHD maximin grid design, errors = 0.1

5 Bayesian Optimisation

5.1 Methodology

The final part of this report's analysis is to further the application of history matching to Bayesian optimisation with the aim to find the maxima of the function, via the emulator corresponding to this function. This is utilised because of the nature of history matching, it is more robust and allows for more uncertainty in the maximum, ensuring the model accurately explores the area to find the true maximum.

5.2 Implementation and Results

Implementing this is fairly straightforward and merely requires altering the code for history matching so that, instead of matching to an observation, it matches to the input corresponding to the largest output found so far.

$$x^+ = argmax_{x^{(i)} \in x^{(1)},\dots,x^{(n)}} f(x^{(i)}))$$
$$I(x) = \frac{f(x^+) - E_D f(x)}{\sqrt{Var_D f(x)}}$$

The implausibility now calculates the possible inputs for which the maximum occurs. This is shown in Figure 8 where it can now be seen that in the first wave, the emulator has found the region for which the maximum is contained within the model and observation uncertainties equal to 0.05, comparing this to the true models non-implausibility regions shows it is very similar. The second wave investigates this non-implausibility region further to ensure its accuracy. This concludes our analysis of 2-D emulators and their applications.

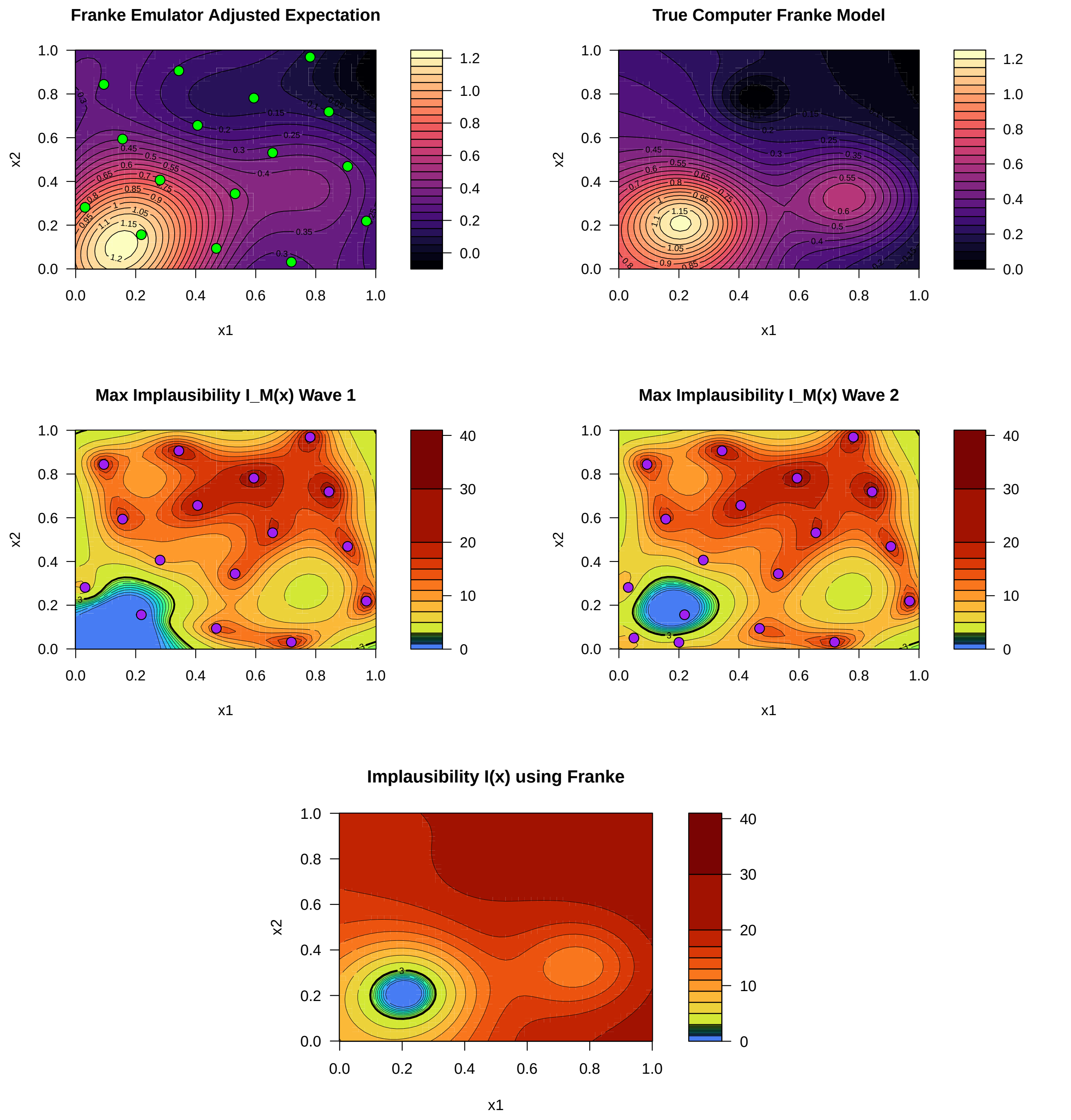


Figure 8: Optimisation with initial LHD maximin grid design, errors = 0.05

6 Conclusion

To conclude the discussion of emulation of the Franke function, it is important to acknowledge the success of the emulator and how well behaved it has been. Diagnostic plots have been within the 3-sigma interval, suggesting that the observations match well with the emulator's output. The variance and expectation plots have shown the output of the emulator and the accuracy of the emulator in the input domain. Variance plots are crucial to understanding how reliable estimates based off the emulator are and in this report it has been seen that the emulator gives very accurate results. Additionally, a comparison of the emulator expectation to the true model shows that the emulator is successful in replicating Franke's function. Important considerations to be made with emulators is the computational cost of fitting the emulator to be accurate over different grid designs. For higher dimensions the number of runs required will increase to maintain the same accuracy and the complexity of the solutions will also increase in time, this is where the benefit of maxi-min LHD becomes much more apparent than what is shown for the 2-D emulation shown in Figure 4.

The aim of this report was to perform emulation on a known function to see how it performs, unfortunately Franke's function has few practical considerations, but it can clearly be seen that emulation has performed very well within the aims of the report. History matching methods were successful in both the optimisation and standard applications with minimal computational cost and a comparison was drawn for different model and observation uncertainty's, clearly showing the effect uncertainty has on matching outputs to inputs and then a comparison to the true implausibility showcased it clearly was an accurate emulation for each parameter set. All of this showcases the success of the emulation of the Franke function.