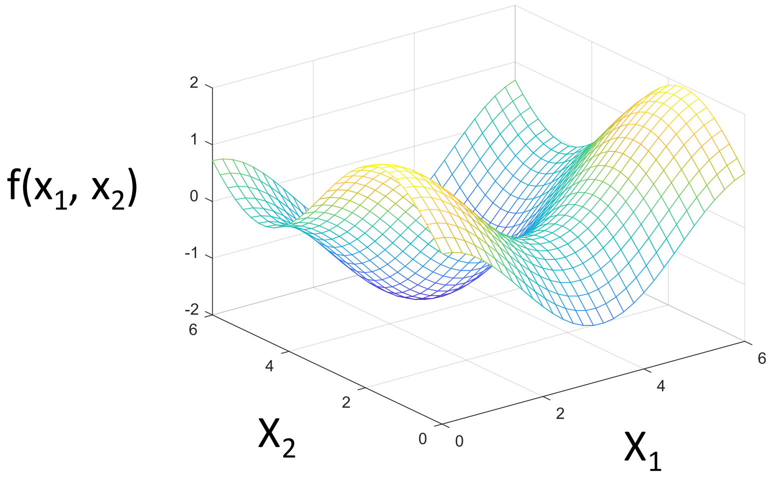
Shubh Raval

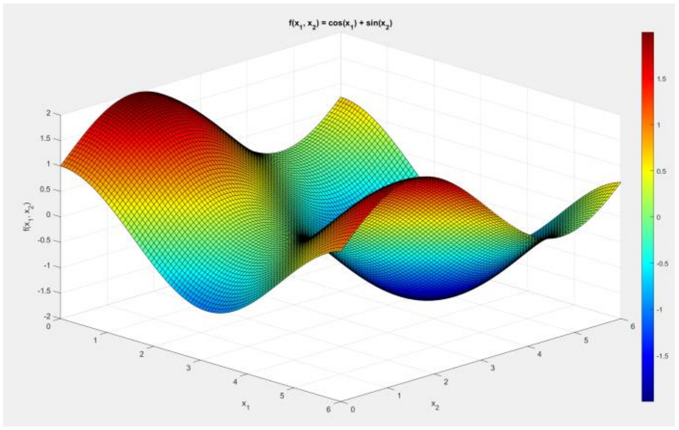
ME 6103 Engineering Optimization

Spring 2025, Dr. Carolyn Conner Seepersad

Assignment 4: Experimentation and Metamodeling

Consider the 2-dimensional function: $f(\boldsymbol{x}) = \cos(x_1) + \sin(x_2)$ for $0 \le x_n \le 6$

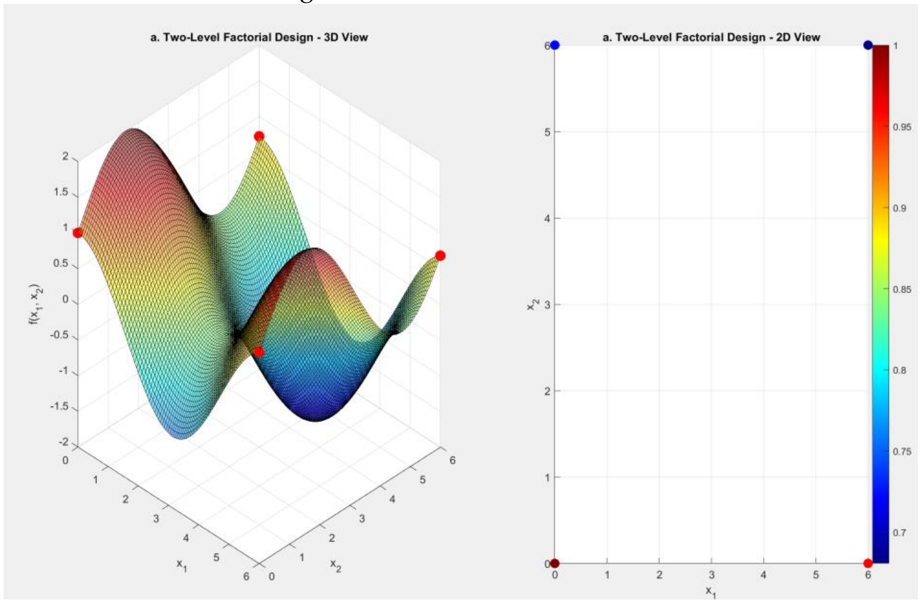




Function Recreated in Matlab

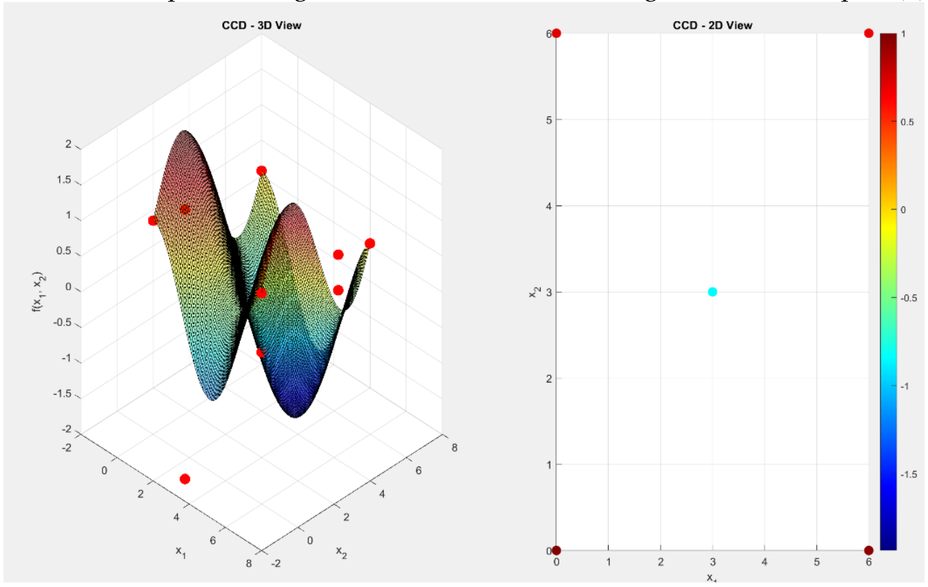
(1) Design and conduct computer experiments on the test function using the following experimental designs:

a. A two-level factorial design.



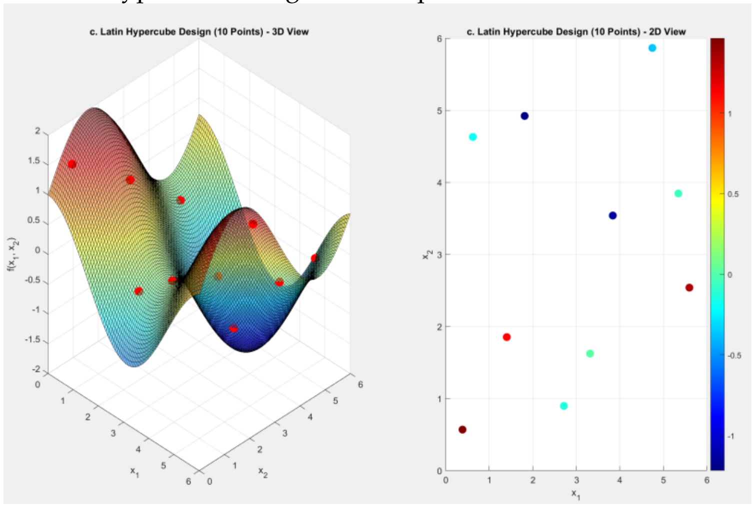
This was done using the factorial combination of the upper and lower bounds of the design variable which was then plotted both on the mesh and shown on a 2d layout

b. A central composite design, based on the factorial design conducted in part (a).



From the factorial experimentation alpha was calculated as sqrt(2) which was then scaled proportionally based on the upper and lower bounds to generate the new points used along with a center point.

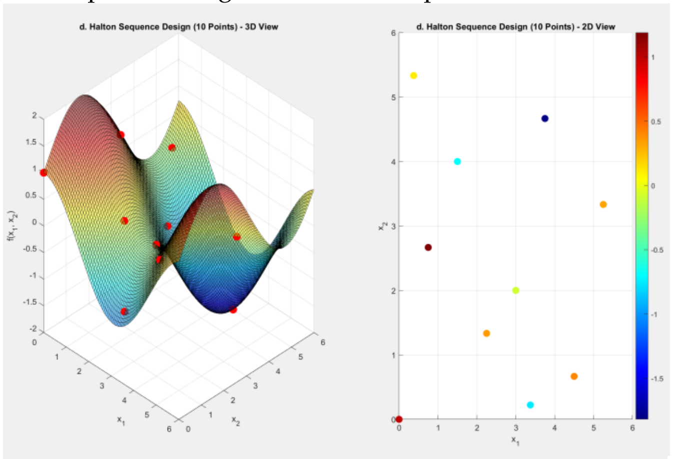
c. A Latin Hypercube design with 10 points.



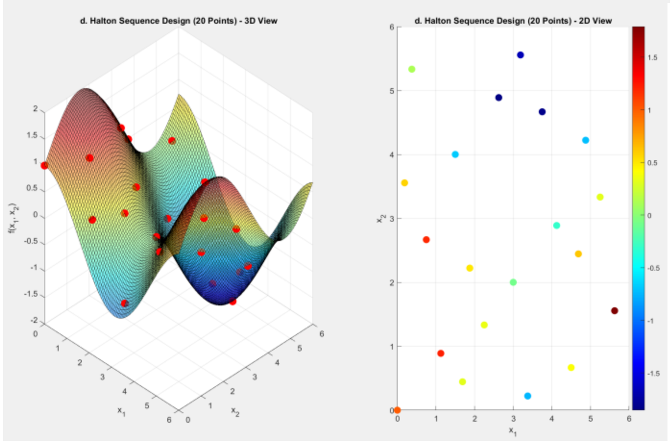
The latin hypercube was implemented with the lhsdesign toolbox from matlab for our 2d design with 10 points desired, each selected point was then evaluated and plotted.

d. Two Halton sequence designs: one with 10 points and one with 20.

i.



ii.



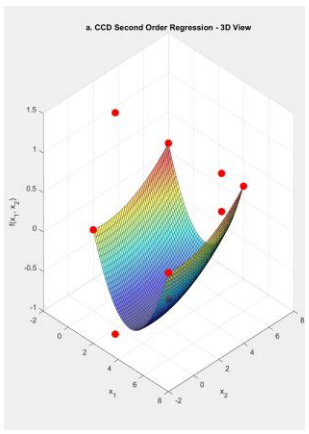
The halton sequence design was done using the haltonset toolbox, where first the set was 2 dimension object was create, then populated with the 20 points, and then these points from a unit base to one based on the upper and lower bound. An initial issue I had with the haltonset was the points not being generated after x1 =

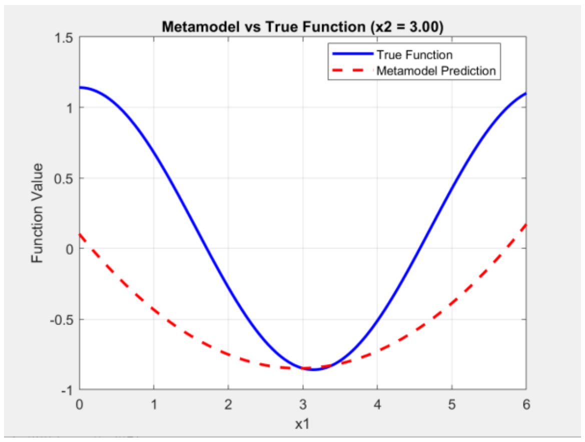
3, which I realized was due to passing in improper parameters to the haltonset object. Specifically from the matworks documentation I had originally implemented the skip and leap parameters caused points to be lost and not properly cover the full design space.

(2) Create the following metamodels based on the experiments in part (1). Plot each metamodel in a 3-d plot (like the one shown above). Quantify the accuracy of each metamodel with a uniform grid of 25 test points and the relative average and relative maximum error metrics introduced in class.

For all the following a uniform grid of the 25 points was fed into the created predictive function and were then evaluated to create the mesh. Additionally similar to lecture a 2d plot to show error was made by hold x2 = 3 was created to show the error between the true function and the prediction. Finally for each the RAAE and RMAE was calculated to determine error against the true function. This was done by creating a function that accepted the uniform gird points, the predictive function and the test responses generated by evaluating the grid. So the absolute error was found between values. Then sigma was calculated and use to the determine the relative error, then the average of these was RAAE, and the max was the RMAE.

a. A second order regression based on the data from (1b).

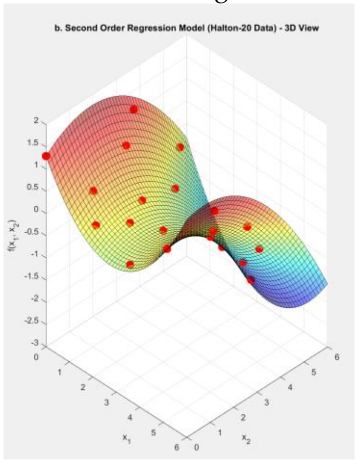


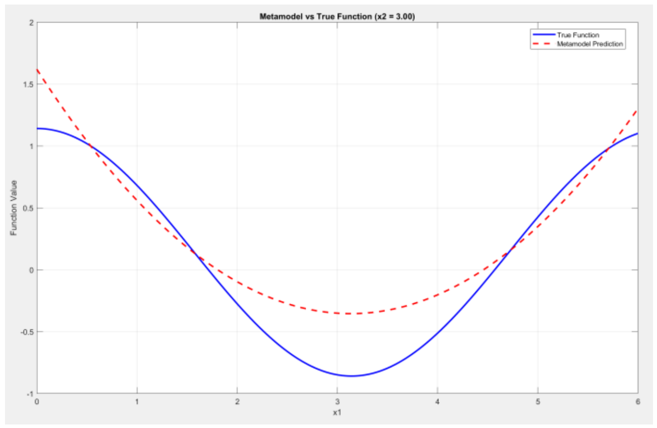


This was implemented by passing the ccd data to the fitlm function to create a function called predict that used the quadratic regression generated to evaluate the points from the uniform grid. Then using the error function the RMAE and the RAAE were calculated:

Relative Average Error (RAE): 0.7542
Relative Maximum Error (RME): 1.9613

b. A second order regression based on the data from (1d).



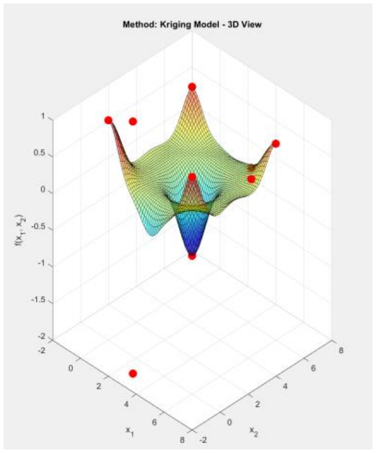


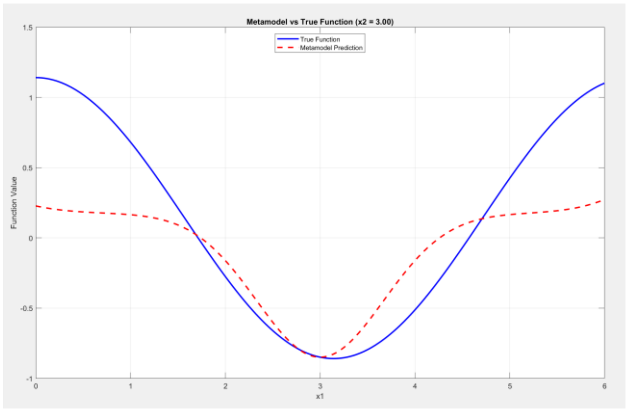
For generating this metamodel a similar implementation was used with fitlm and the RMAE and RAAE were determined to be:

Relative Average Error (RAE): 0.6431

Relative Maximum Error (RME): 2.2540

c. A kriging model based on the data from (1b).[1] Use a zero order polynomial and a Gaussian correlation function.





This was implemented via the kriging toolbox provided in class and essentially following along with the basic script give. The major change was it was turned into a function so that its easier to use in this broader scope. Finally one issue that occurred with kriging was the multiple design sites error based on duplicate results from evaluation, to handle this I simple kept the first unique instance and discarded the other duplicate instances.

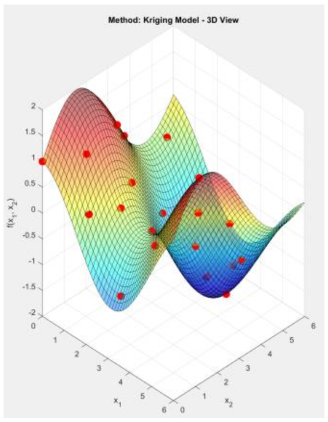
The error found was:

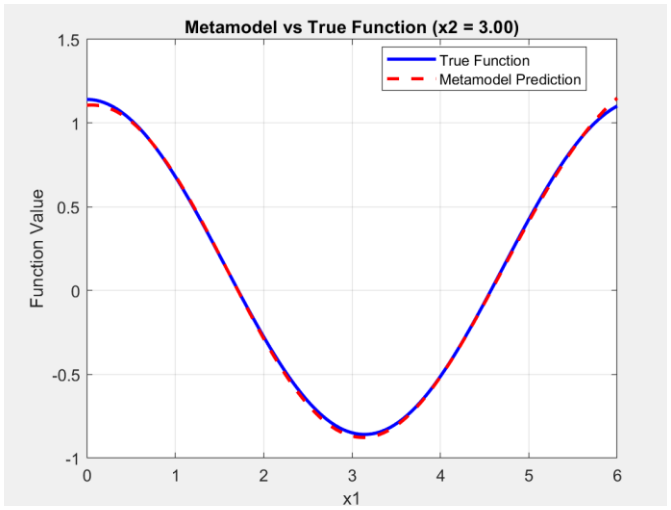
Relative Average Error (RAE): 0.6311

Relative Maximum Error (RME): 2.0466

[1] Use the DACE toolbox for Matlab, available in a zip file on Canvas in the metamodeling module. Start with the .m file labeled "SimpleKrigingforME397.m"

d. A kriging model based on the data from (1d). Use a zero order polynomial and a Gaussian correlation function.





This called the kriging function I wrote in my script and passed in the appropriate data where similarly to handle duplicate points the first unique was kept. Overall it is visually and quantitatively apparent that this combination of halton 20 and the kriging meta model has produced the most accurate result. With the error being:

Relative Average Error (RAE): 0.0630

Relative Maximum Error (RME): 0.4742

e. Compare and contrast the different metamodels and explain any trends you observe.