

## Experiment 3

This folder contains all the code necessary for running the Third experiment, exploring behaviour of a parallel Bayesian optimisation algorithm.

### Experiments and diagnosis.

#### experimentThree2d.m

This is the code for running experiment 1 for a 2-dimensional rastrigin test function. It saves the data to **experimentTwo2d.mat**

#### experimentThree4d.m

This is the code for running experiment 1 for a 2-dimensional rastrigin test function. It saves the data to **experimentTwo4d.mat**

#### experimentThree8d.m

This is the code for running experiment 1 for a 2-dimensional rastrigin test function. It saves the data to **experimentsTwo8d.mat**

#### TwoDimPlotDiff.m

This plots the 2-dimensional output of a model generated by the **cokriging.m** function and compares it with the true rastrigin function.

#### Pred.m

This gives the model output for a given x variable and is used in **TwoDimPlotDiff.m**.

### Bayesian Optimisation algorithms

#### minvarparRastd.m

This function runs a variable expected Improvement Bayesian optimisation loop for the Rastrigin test function with the number of dimensions given by the parameter dim. There are options to change the seed for reproducibility, the number of expensive points, cheap points and the total computation cost and the cost of running a high fidelity evaluation as well as the number of high-fidelity function values per iteration of the model.

### Optimisation

#### maxVarExpImp.m

This returns the maximum expected improvement for a given model, depending on whether it returns a high-fidelity point or a low fidelity point. And depending on the cost of a high-fidelity evaluation or a low-fidelity evaluation.

#### minModelPred.m

This returns the x value associated with the optimal value given by the Cokriging model.

## Test function

### `rastrigin.m`

This is the rastrigin test function with 3 different error functions for low fidelity data and a linear cost scale from [0-10000] with 1000 being the low-fidelity and 10000-being the high-fidelity.

## Modelling Code

### `Cokriging.m`

A function that takes two set of points and the corresponding low and high-fidelity function evaluations as well as an integer value indicating how many attempts at finding the best values for the Cokriging model and gives a data structure called model, which is used for computations. It contains the two likelihood functions that it calls on using the inbuilt genetic optimisation algorithm.

### `likelihoodc.m`

A function that works out the negative log likelihood for kriging regression for a given set of parameters.

### `likelihoodd.m`

A function that works out the negative log likelihood for difference Cokriging regression for a given set of parameters.

### `buildmodel.m`

A function that takes all the parameters found by the above two function and gives an output is the model structure. It also had a flag output which is set off if the covariance matrix for the Cokriging model is not positive definite.

## Initial Sampling

### `initialdesign.m`

Generates the initial sample for any function on an interval

### `srgts...`

These functions come from a toolbox on surrogate functions and are there for taking a sub sample from the initial sample.