AE6310: Optimization for the Design of Engineered Systems

Assignment 4: MDO Architectures, Surrogate Modeling and Gradient-Free Methods

Due: April 21st, 2020

Answer the following questions. Organize your work and be careful to properly answer all parts of each question. Points will be deducted for unorganized presentation of results.

1. (80 points) This question is about the following function defined on $x_i \in [-1, 1]$:

$$f(x_1, x_2) = x_1^2 - x_2^2 - \cos(\frac{\pi}{2}x_1)\cos(\frac{\pi}{2}x_2)$$

You are given a DoE with $N = M^2$ points where each point is evenly spaced along each coordinate so that the point $i + M(j-1)$ for $i = 1, \ldots, M$, and $j = 1, \ldots, M$ is $x_1 = -1 + 2(i-1)/(M-1)$ and $x_2 = -1 + 2(j-1)/(M-1)$. This creates a DoE with evenly spaced points along each direction.

For $M = 3$, the DoE would be

Point	x_1	x_2
1	-1	-1
2	0	-1
3	1	-1
4	-1	0
5	0	0
6	1	0
7	-1	1
8	0	1
9	1	1

Part 1: Given the quadratic basis functions:

Function Index	Basis function
1	1
2	x_1
3	x_2
4	x_1^2
5	$x_1 x_2$
6	x_2^2

(a) Find the weights $w \in \mathbb{R}^6$ using $M = 3$ (with $N = 9$ points) and $M = 5$ (with $N = 25$ points)

(b) For both models plot the error $\epsilon(x) = f(x) - \hat{f}(x)$ as a contour plot

(c) Pick 100 random points in the domain and perform validation. Compute the R^2 values for each of your models.

Part 2: Given the Gaussian radial basis function $\phi(r) = e^{-r^2/2r_0^2}$ with $r_0 = 1$

(a) Using an interpolation model with $N = m = M^2$, find the radial basis interpolation with $M = 3$ (with $N = 9$ points) and $M = 5$ (with $N = 25$ points)

(b) For both models plot the error $\epsilon(x) = f(x) - \hat{f}(x)$ as a contour plot

(c) Pick 100 random points in the domain and perform validation. Compute the R^2 values for each of your models.

2. (20 points) Compare the performance of a gradient based optimizer and a gradient free optimizer on the following function:

$$f(x_1, x_2) = |x_1 x_2| + 0.1(x_1^2 + x_2^2)$$

Note: Please use `fmincon`, `scipy.optimize` or equivalent for these problems. If you're using matlab, consider using Nelder–Mead with fminsearch. If you're using python, consider using scipy optimize package with both gradient-based and gradient-free methods. Be careful to use the correct options to select either a gradient-based or gradient-free algorithm.

(a) Where is the minimizer?

(b) Does the gradient-based optimizer exit successfully, even for tight convergence tolerances?

(c) Compare the accuracy of the gradient based optimizer and the gradient-free optimizer

(d) Compare the cost in number of function evaluations of each optimizer.