CHALMERS | UNIVERSITY OF GOTHENBURG

MASTER'S THESIS

Optimization of the Number of Evaluations in Optimization

TURGAY GÛL

Department of Mathematical Sciences
Division of Mathematics
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2012

Thesis for the Degree of Master of Science

Optimization of the Number of Evaluations in Optimization

Turgay Gûl

Department of Mathematical Sciences
Division of Mathematics
Chalmers University of Technology and University of Gothenburg
SE – 412 96 Gothenburg, Sweden
Gothenburg, December 2012

Matematiska vetenskaper
Göteborg 2012

THESIS FOR THE DEGREE OF MASTER OF SCIENCE

OPTIMIZATION OF THE NUMBER OF EVALUATIONS IN OPTIMIZATION

Supervisors:

SVEN AHLINDER

Volvo Technology Corporation

IVAR GUSTAFSSON

Chalmers University of Technology

Author:

TURGAY GÛL

Department of Mathematical Sciences

Chalmers University of Technology

Abstract

Any automobile existing today constitutes of thousands of parts. Each part when under analysis can be characterized by different parameters. The numbers of parameters associated to each part depend on the complexity of that particular part e.g. body or the engine of the vehicle which needs millions of parameters to be clearly defined to be reproducibly manufactured. The engine can be build up by a system, which we can say is a black-box, which for instance can describe the gasoline consumption of the engine of a car. In practice it is impossible to work with an unknown system while investigating the optimization methods, therefore we assume that we have a quadratic function instead.

In the optimization we fit the quadratic function with linear models of four different kinds. These are the gradient method, the design of experiments, the spherical method and the lean optimization method.

Supersaturated design approach (which we did use in the spherical and the lean optimization method) is a way to make the number of experiments less than the number parameters of a system. The spherical method was the best one in the optimization of the functions. It gives the smallest standard deviation among these four methods. From the optimization of functions it was possible to see that the spherical method gives very good optimum values (close to zero). The methods are possible to be used for optimization but not for prediction. In the prediction we are removing some values and using some model-values instead of these values. The standard deviation of the original function value was better than the standard deviation of the difference of the four different optimization functions and the original function.

Acknowledgements

I would like to thank Sven Ahlinder, my supervisor from Volvo Technology Corporation, for helping me during this project. I also would like to thank Ivar Gustafsson, my examiner from Department of Mathematical Sciences, Chalmers University of Technology, for his comments on my work.

Contents

1 Introduction

Experiments are going to be performed on a black-box. These are experiments for calculating the optimum value (minimum) for a system. But these are expensive and time-consuming and should be very carefully planned. Our aim is to minimize the function describing the system. For simplicity we assume a quadratic function $f(x)= x^T A^T Ax$ where A is a normally distributed random matrix. Observe that $A^T A$ is a positive semidefinite matrix since

$x^T A^T Ax \geq 0$ for all vectors x.

For a present x we are calculating different descent directions of the function $f(x)= x^T A^T Ax$ and then using the line search method to calculate an x-vector, and a corresponding function value, which is smaller than the function value at the x-vector we started at. We did use four kinds of methods and we have one design matrix for each of these optimization methods. In this report besides the gradient method, we will also consider the Design of Experiments method, the lean optimization method and the spherical method.

For calculating the descent direction we use different so called design matrices and one technique is called Design of Experiments (DoE.)

The main goal of Design of Experiments, which is a very cost effective approach and is used in the early part of the problem before the screening of the system, is to plan a process in an optimal way with single or multiple underlying objectives [2]. In the Design of Experiments method we have a design matrix which represents a number of experiments to be performed. Each variable is represented by a column and each row represents one set of values for each variable. We call each row an experiment. When we have 2 –level case, we evaluate each variable only at its lowest and its highest value (in the matrix design it is represented by -1 and +1). This will be explained in more details in section 2.

We have also studied the prediction ability of the methods. We have 30000 corners of a 15 dimensional cube from which 17 corners are selected. In the prediction you are extracting the predicted response from some corners. We did estimate the standard deviation of $y= x^T A^T Ax$ (original spread) and the difference of the four different optimization functions and the original function. We are predicting the function value of the corners. We are using and comparing four methods, the gradient, the DoE, the lean and the spherical method as mentioned before. In the prediction the corners are tested by its prediction ability.

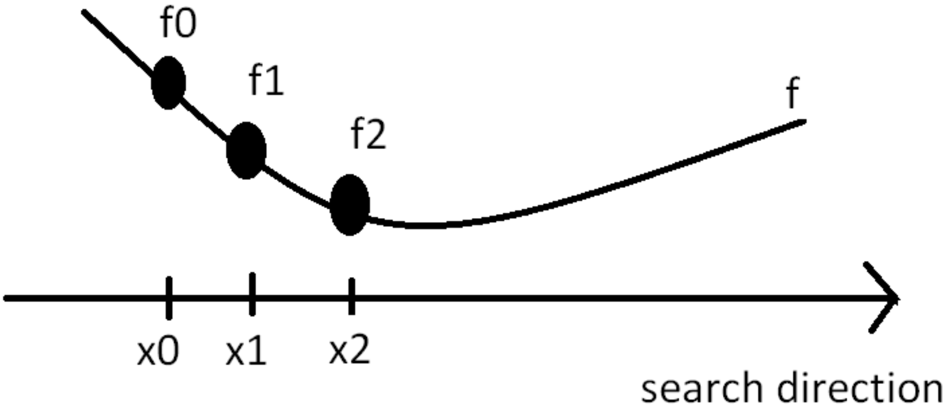
We are checking the difference between function value and prediction: We are using the original function $y=x^T A^T Ax$. Here x are all the corners. The other function we have to calculate with linear approximation. We will get the difference between these two function values in order to calculate the standard deviation.

2 Design matrix

2.1 Line search method

Assume that we have calculated an approximation to the gradient of the function $f(x)=x^T A^T Ax$ at a point x0. How this is done is explained in section 2.2. Line search means that we are minimizing the function along the search direction which is opposite the gradient direction.

Since we have a quadratic model function $f(x)=x^T A^T Ax$, line search is quite simple. We may compute the minimum value of f(x) along a search direction by taking three equidistant points in this direction and compute the f-values in these points. By these values the quadratic function is uniquely determined and it is an easy matter to compute the minimum.



2.2 Design matrix in the line search method

The design matrix is used to approximate the gradient of the function. The following MATLAB-algorithm will be used to define a linear system, the solution of which gives the approximate gradient: G=D+ones(m,1)*x0. Here G is the matrix of the linear system and D is a design matrix, see 2.3. At the very beginning x0 is a vector for initialization, but this vector will be updated with new x0 at the end of the line search method.

Here follows the principle for the procedure of approximating the gradient in two dimensions for m=3 and x0=(x_1,x_2).

For the design matrix D= $\begin{pmatrix} \delta & 0 \\ 0 & \delta \\ \delta & \delta \end{pmatrix}$ we get G= $\begin{pmatrix} x_1+\delta & x_2 \\ x_1 & x_{2+}\delta \\ x_1+\delta & x_2+\delta \end{pmatrix}$

And we solve the following system to get an approximation of the gradient:

$$\begin{bmatrix} 1 & x_1+\delta & x_2 \\ 1 & x_1 & x_2+\delta \\ 1 & x_1+\delta & x_2+\delta \end{bmatrix} \begin{bmatrix} \bullet \\ g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

Observe that we have added a first column of ones to the matrix G and the dot character is a notation for a value that we do not need in the solution. The system is equivalent to

$$\begin{cases} \bullet + (x_1+\delta)g_1 + x_2 g_2 = f_1 \\ \bullet + x_1 g_1 + (x_2+\delta)g_2 = f_2 \\ \bullet + (x_1+\delta)g_1 + (x_2+\delta)g_2 = f_3 \end{cases}$$

where f_1, f_2 and f_3 are the function values at the points (x_1+δ, x_2), (x_1, x_2+ δ) and (x_1+δ, x_2+ δ) respectrively. The unique solution to this system is

$$\begin{cases} g_1 = \dfrac{(f_1 - f)}{\delta} \\ g_2 = \dfrac{(f_2 - f)}{\delta} \end{cases}$$

Here $\bullet + x_1g_1+x_2g_2=f$

is the function value at the point (x_1, x_2). It becomes clear that $\begin{pmatrix} g_1 \\ g_2 \end{pmatrix}$ is the divided difference approximation to the gradient of f at the point (x_1, x_2).

The different design matrices D in section 2.3 sometimes result in an overdetermined system of equations and sometimes in an underdetermined system. In case of an overdetermined system we use least squares and in case of an underdetermined system we use the pseudoinverse. The pseudoinverse is based on the singular value decomposition of the matrix, see [1].

For instance, in the lean and spherical method we are using the pseudoinverse since the system of equations appearing are underdetermined. The line search method is done in same fashion for all considered designs.

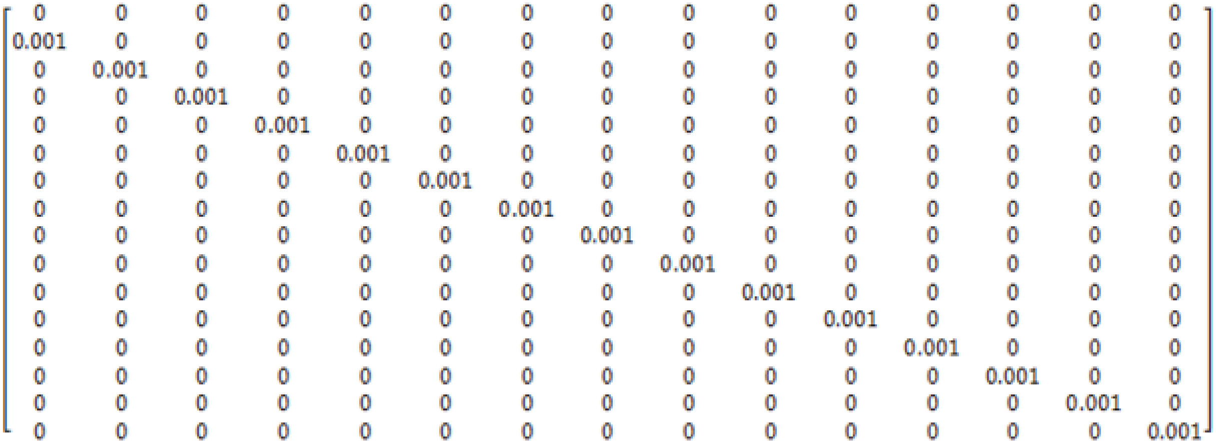
The optimization methods (gradient, DoE, lean optimization and spherical method) are done with and without permutations [3]. By permutation we mean that we are permuting the columns with each other. When the permutation in the columns is done we will get new experiments.

2.3 Experimental designs

2.3.1 Gradient matrix

In this case the design matrix takes the form:

D$\mathbf{D_G}$=

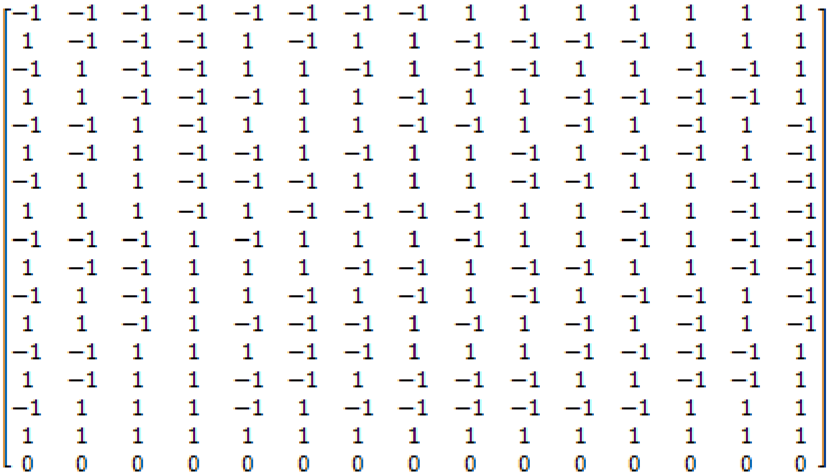


$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.001 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.001 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.001 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.001 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.001 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.001 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.001
\end{bmatrix}$$

In our experimental design we have 15 columns and 16 rows which means that we have an over determined system. The step size is $\delta = 0.001$ (as shows in the diagonal). The columns are independent.

2.3.2 DoE (Design of Experiment)

Here the design matrix takes the form: $\mathbf{D_E}$=



$$\begin{bmatrix}
-1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & -1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 \\
-1 & 1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 \\
-1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 \\
-1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 \\
-1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & -1 \\
1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
-1 & 1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 & -1 \\
1 & 1 & -1 & 1 & -1 & -1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
-1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 \\
1 & -1 & 1 & 1 & -1 & -1 & 1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
-1 & 1 & 1 & 1 & -1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

In this our data is all spread on the corners of the cube. In DoE method cubes are being used with two level centrum point, consisting of upper level (+1) and lower level (-1). The last row, which consists of only zeros, means that you are in the centrum point of the cube, so you do not have any upper nor lower level. The row which consists of only ones means that you have upper level in all the corners of the cube.

DoE is involved in making set of experiments. It is used in many industrial sectors e.g. in the development and in the optimization of manufacturing process. It is also used in the laboratory pilot plot and full scale production.

The main advantage of DoE is full organized approach with which it deals with simple and tricky problems. We will use classical DoE, which is used for estimating the unknown parameter in linear regression model.

2.3.3 Lean optimization

Here the design matrix takes the form::

D_L=

$$\begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

In the lean optimization we are dealing with a number of experiments very less than the number of the variables. In the lean method we will consider the AG design, [2]. We used two different kinds of matrices. The first one is the lean-matrix, given above. The other one is the spherical matrix, see 2.3.4.

In the design matrix D_L above the data is well spread on the corners of the cube. It is a supersaturated design where 1 indicates the higher level and -1 indicates the lower level of the cube. In this method we have less number of rows then columns. In this case you can solve the equation system for calculating the gradient by using the pseudoinverse.

2.3.4 Spherical optimization

Here the design matrix takes the form: $\mathbf{D_S}$=

$$\begin{bmatrix} -0.34 & -0.34 & -0.34 & -0.37 & -0.01 & -0.01 & -0.10 & 0.10 & -0.11 & -0.48 & -0.48 & -0.10 & -0.05 & -0.09 & -0.04 \\ 0.05 & 0.05 & 0.06 & 0.07 & -0.24 & -0.24 & -0.24 & -0.21 & -0.03 & 0.29 & 0.29 & -0.46 & -0.53 & -0.23 & -0.22 \\ 0.18 & 0.18 & 0.17 & 0.17 & -0.12 & -0.12 & 0.06 & -0.55 & -0.38 & -0.07 & -0.07 & 0.32 & 0.52 & 0.04 & -0.06 \\ 0.25 & 0.25 & 0.28 & 0.16 & 0.37 & 0.37 & 0.14 & 0.30 & 0.17 & -0.08 & -0.08 & -0.39 & 0.05 & 0.19 & 0.39 \\ -0.14 & -0.14 & -0.17 & -0.03 & 0.00 & 0.00 & 0.14 & 0.37 & 0.36 & 0.34 & 0.35 & 0.63 & 0.00 & 0.09 & -0.07 \end{bmatrix}$$

This design is called spherical design, there are as many dimensions as the number of variables. Moreover there are as many rows as the number of experiments. All the data are spread out and we are maximizing the minimal distance of the data. Spherical design means that the distance is the same between all the data in the system, the data is well-spread. This means that all data is well spread on the sphere with equal distance from the origin.

3 Prediction of corners

3.1 Idea

We did test the model prediction by considering a 15 dimensional cube with 2^{15} (around 30 000) corners out of which we did extract the predicted response from some corners. By this we mean that we are checking the prediction of the function values of the corners. We did also check the standard deviation of the original function y= $x^T A^T Ax$ (original spread) and the difference of the four different optimization functions and the original function. By this we mean that we are simply calculating the standard deviation of the difference generated by the four methods and the original function and also the deviation of only the original function.

We have 30000 corners from which 17 corners are selected. In DoE we did work with different methods that are classical Design of Experiments, Partial least square method [4] and variable selection. Partial least square method (PLS) is a regression which works to connect the information in two blocks of variables, X and Y in each other.

In its simple form a linear model shows the linear relationship between the dependent variable (Y) and a set of predicted variables (X).

$$\mathrm{Y} = \mathrm{b}_0 + \mathrm{b}_1\mathrm{X}_1 + \mathrm{b}_2\mathrm{X}_2 + \cdots .. + \mathrm{b}_\mathrm{p}\mathrm{X}_\mathrm{p}$$

X_i are the independent variables. From equation, b_0 is the registered regression coefficient and b_i (where, i = 1, 2, ...p) are the regression coefficient computed from the data [5]. Here X_p is the last predicted variable and p is the dimension of the space.

Note that the original function value, y= $x^T A^T Ax$, is calculated in different way and it is not same as big y. The big y is used for calculating the function value of the four different optimization functions. Moreover the small x:s are same as the big X:s because we are using same x-values in the calculations of both the original function value and the function value of the four optimization functions.

Note that the x-coordinates (X_1, …, X_p) will be given. Calculations of the gradient, g0, can be done for each element step by step. Therefore it is possible to say that g0 corresponds to b ($b_0, \dots, b_p$) and conclude that there is a linear relationship between y and x (y= $g0 \cdot x$). This is explained by a simple example in section 2.2. Note that x corresponds to the big x:s while y corresponds to big y. The other y-value which we can get from the original function, y= $x^T A^T Ax$, is different and is used for computing g0.

In the prediction of the corners we did consider two things. One is histograms and other is the standard deviation. While dealing with the corners we are working with the fit of the model and on the prediction.

Here we use standard measures from statistics, denoted Q^2 and R^2, see [5]

Q^2 is the predictive measure corresponding to the measure of fit. By fit here we mean how well a mathematical function will fit to a series of data points. R^2 is the percent variation of the response explained by the model (prediction). R^2 varies between 0 and 1. If we have 1 then it means that we have perfect fit. Otherwise if we have 0 then it means that we do not have any model. A Q^2 larger than zero indicates that the model is significant and predictive.

$$\text{Prediction } Q^2 = \frac{\text{ss} - \text{press}}{\text{ss}}$$

$$\text{fit } R^2 = \frac{\text{ss} - \text{ss}_{\text{resid}}}{\text{ss}}$$

Prediction Residual Sum of Squares, PRESS, is a measure of how well the model will predict the responses for new experimental conditions. SS means sum of squares of Y corrected for the mean. SS_{resid} is often referred to as a measure of unexplained variation—the amount of variation in Y that cannot be attributed to the linear relationship between X (independent variables) and Y (dependent variable).

3.2 Methods

The DoE method uses partial least square, PLS, and variable selection. In variable selection we are selecting the random relevant features of the model, by removing the irrelevant data work and keeping the relevant data that helps in improvement of the performance of the model.

The work carried out by classical DoE in MODDE (Modeling and Design), a windows program for the generation and evaluation of statistical experimental designs, checking the system fit and the senses of the prediction of the system.

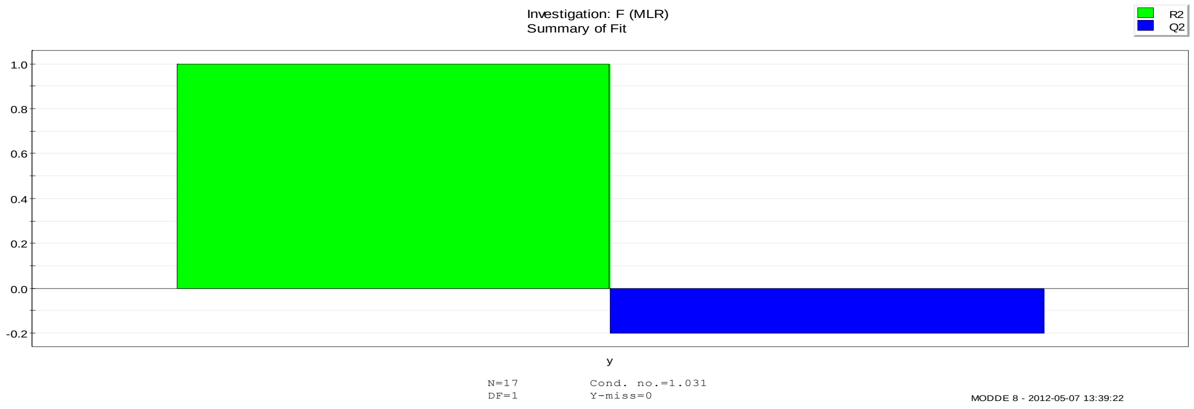


Figure 1: illustrating the fit and the prediction of the system. Here we did use PLS in DoE.

You are considering a linear fit to your design. If the area to the left, which shows the fit, is fulfilled, then we have perfect fit. If the area to the right, which shows the prediction, is fulfilled, then we have perfect prediction.

R^2 (to the left) shows us the fit of the system, which is perfect. Q^2 (to the right) shows us the prediction of the system, which is negative. This implies that the system does not have any sense for the prediction.

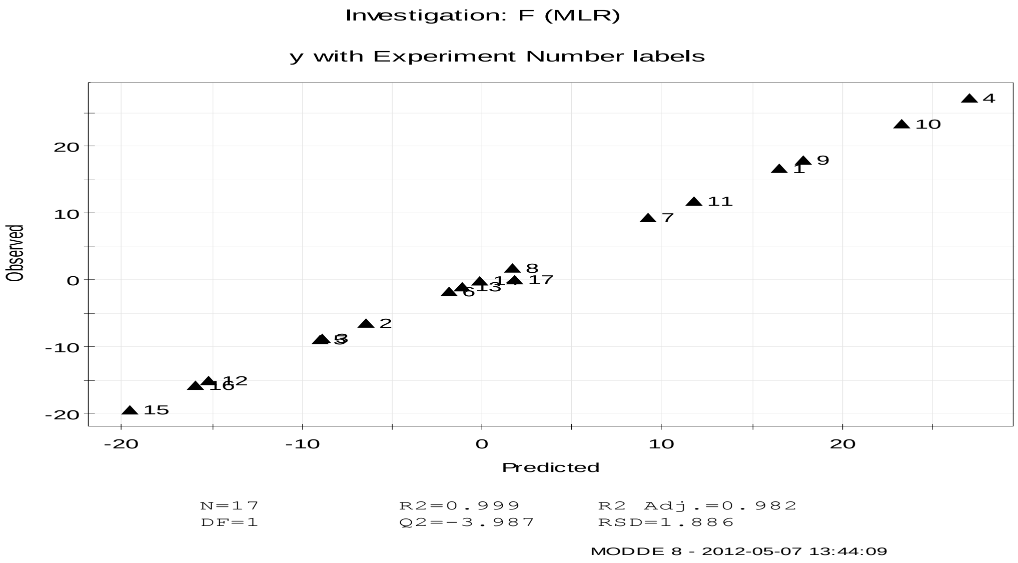


Figure 2: shows that there is good fit of the model (which consists of the function values of the 17 corners) because it seems to be linear. But we do not have any sense of prediction, which can be seen from figure 1. Here we did use the PLS method.

Now consider the work carried out by the partial least square method in SIMCA (Soft Independent Modeling of Class Analogy), a classification method.

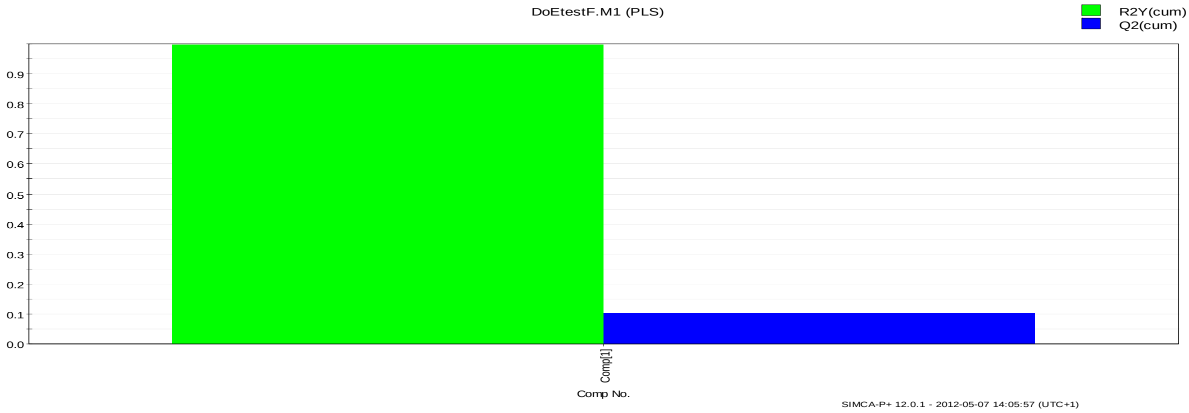


Figure 3: illustrating the fit and the prediction of system by PLS in DoE

In figure 3 R^2 (to the left) shows us the system fit which is perfect and Q^2(to the right) shows the prediction. The system has some prediction ability. The fit is the difference between the measured values and the model-values.

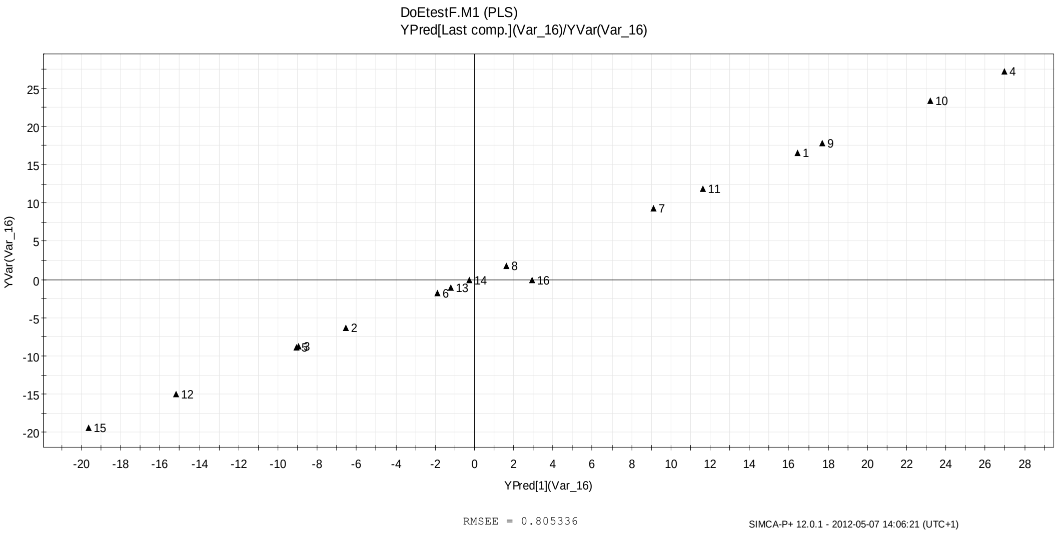
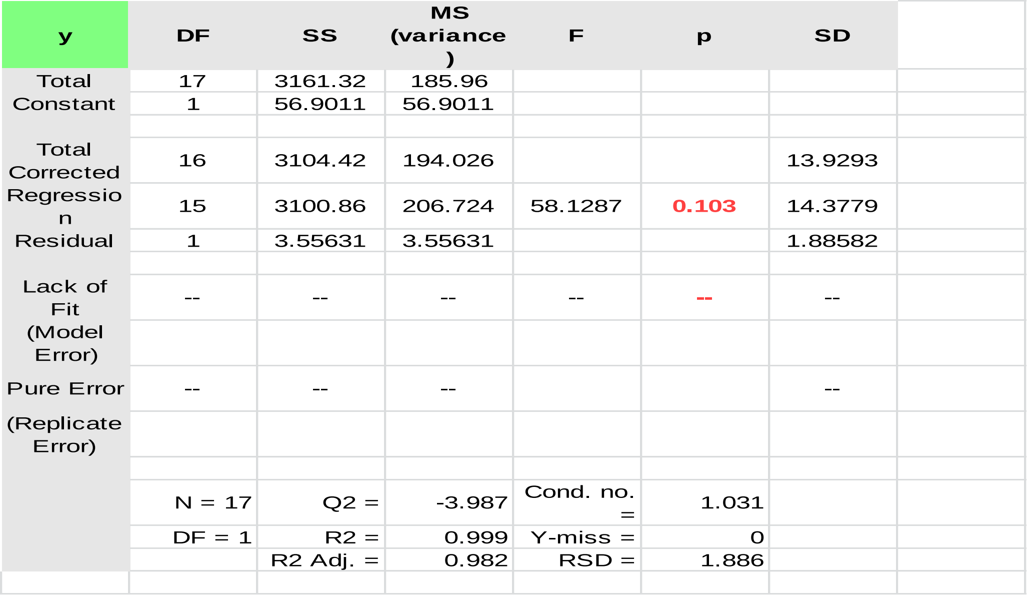


Figure 4: shows slightly worse fit than before because the model is not linear. Here we have the prediction ability (which is not good because only a very small area is filled), which can be seen from figure 3. We did use the PLS method.

3.3 Fit and prediction



y	DF	SS	MS (variance)	F	p	SD	
Total	17	3161.32	185.96				
Constant	1	56.9011	56.9011				
Total Corrected	16	3104.42	194.026			13.9293	
Regression	15	3100.86	206.724	58.1287	0.103	14.3779	
Residual	1	3.55631	3.55631			1.88582	
Lack of Fit	--	--	--	--	--	--	
(Model Error)							
Pure Error	--	--	--			--	
(Replicate Error)							
	N = 17	Q2 =	-3.987	Cond. no. =	1.031		
	DF = 1	R2 =	0.999	Y-miss =	0		
		R2 Adj. =	0.982	RSD =	1.886		

Figure 5: In our test we have 17 observations. Here the method we did use is called the ANOVA-method.

ANOVA table stands for the analysis of variance. It is a statistical technique. In the ANOVA-table from figure 5 we are calculating the mean square (MS) with this formula: MS=SS/df. Here the probability is 10.3 percent. The ANOVA table [6] is significant if the P-value is bigger than alpha level of significance (reject if alpha is bigger than 0.103). If we assume that the significance level is 0.95, then it means that this table is not significant, therefore we should reject this model, which means that we cannot use the ANOVA table for prediction.

In the ANOVA-table (figure 5) R^2 shows the fit of the model. This varies between 0 and 1. It is used in model validation. In our case this value corresponds to 0.999. The goodness of fit is shown as an R2-value. A value of R2=1.0 indicates a perfect fit, whereas R2=0.0 indicates that the regression model might be unsuitable for this type of data.

Q^2 is the prediction variation which calculates the prediction power of the model. It estimates the goodness of the prediction.

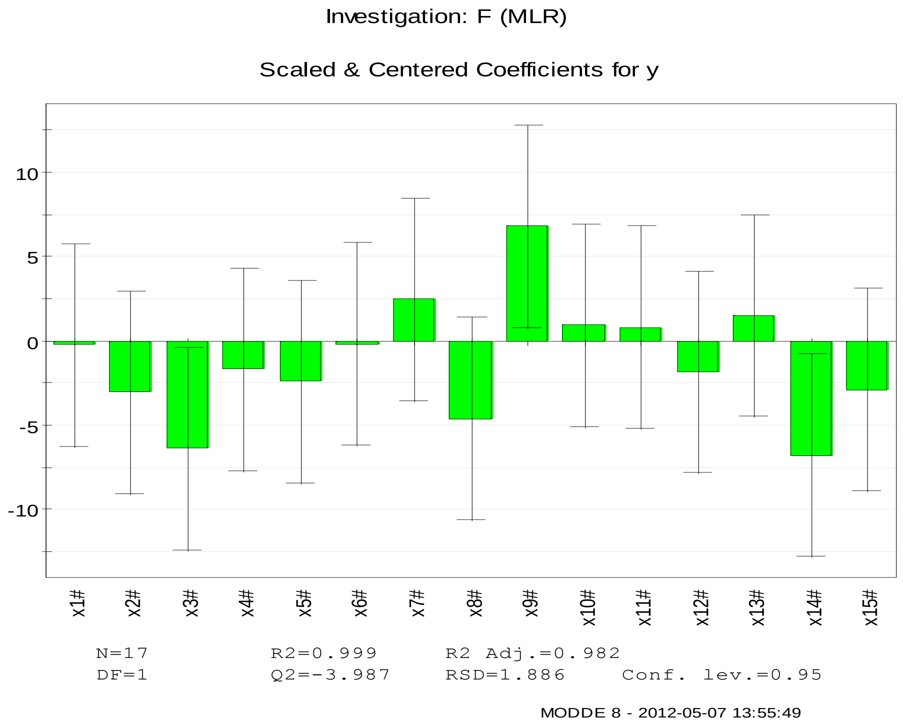


Figure 6: interval value of the coefficient from classical DoE. x_3, x_9, x_{14} are significantly different from 0. As we can see from the figure those whiskers are not crossing the line. If the whiskers, the interval of the value of the coefficients, are separated from zero, then it means that these are good because they are not crossing the line (x-axis).

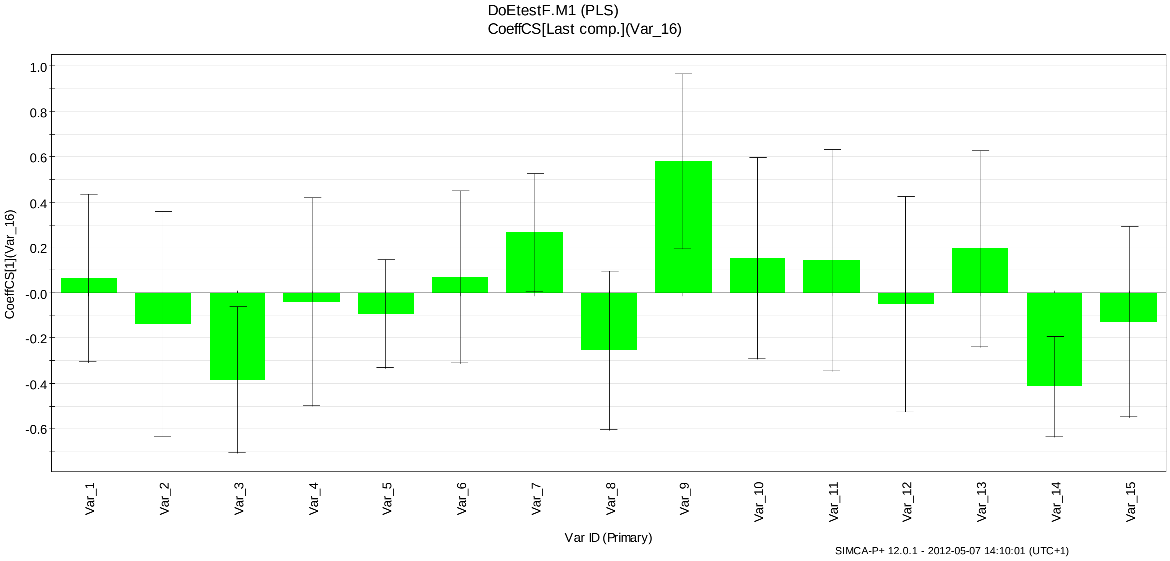


Figure 7: Interval value of coefficient from PLS. We can see from the figure that 3, 9, 14 are significantly different from 0. We have improvement in 7. If we compare with figure 6 we can realize that this coefficient this time was separated from zero. PLS seems to be better than classical DoE.

In the histograms it is possible to distinguish how many times of each number (of the difference between the function value generated by respective method and the original function value) we can find (by looking at the x-axis).

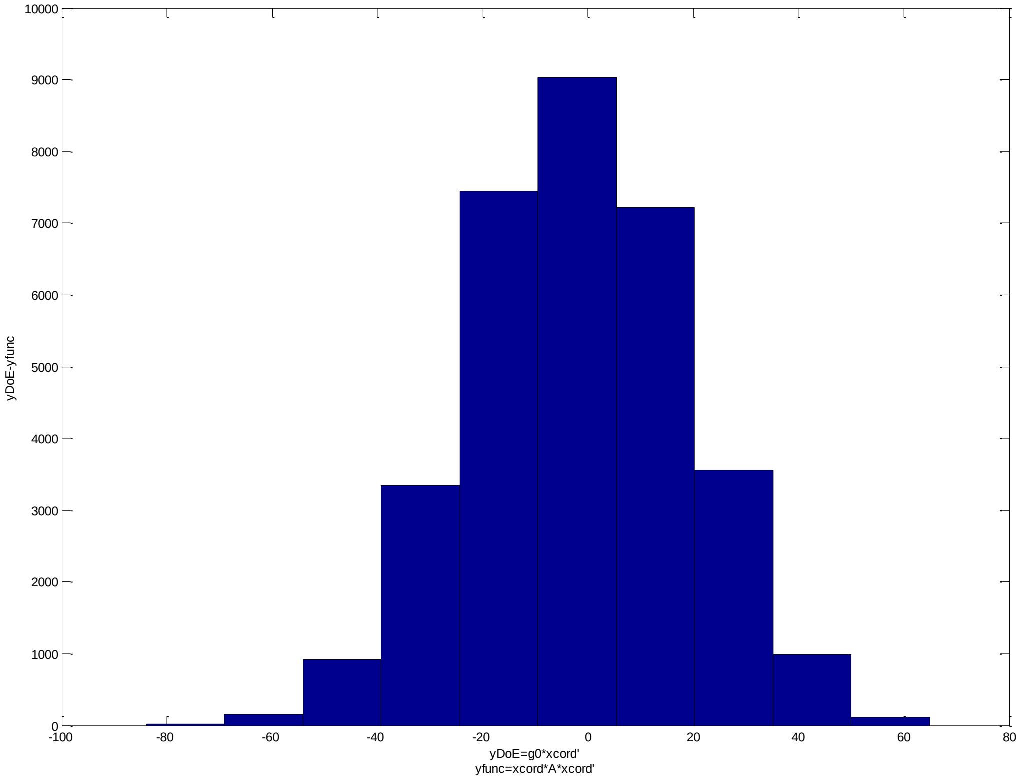


Figure 8: DoE histogram plot of the difference between the function value generated by the DoE method and the original function value, which we get from y= $x^T A^T Ax$.

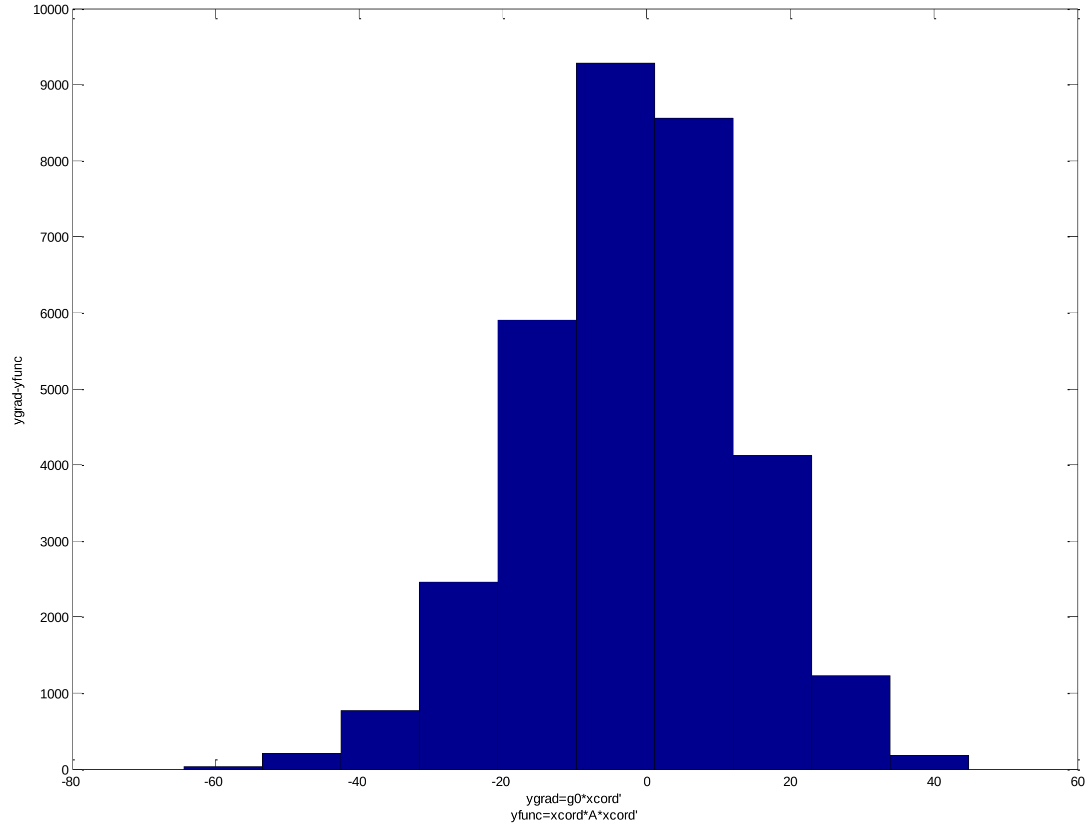


Figure 9: gradient histogram plot of the difference between the function value generated by the gradient method and the original function value.

From the figure 8 and figure 9 we can see that we have more narrow spread in the second histogram, gradient method, which means that this method is better than DoE according to the histograms. This is also possible to realize by looking at the standard deviations (the gradient method has smaller deviation than DoE).

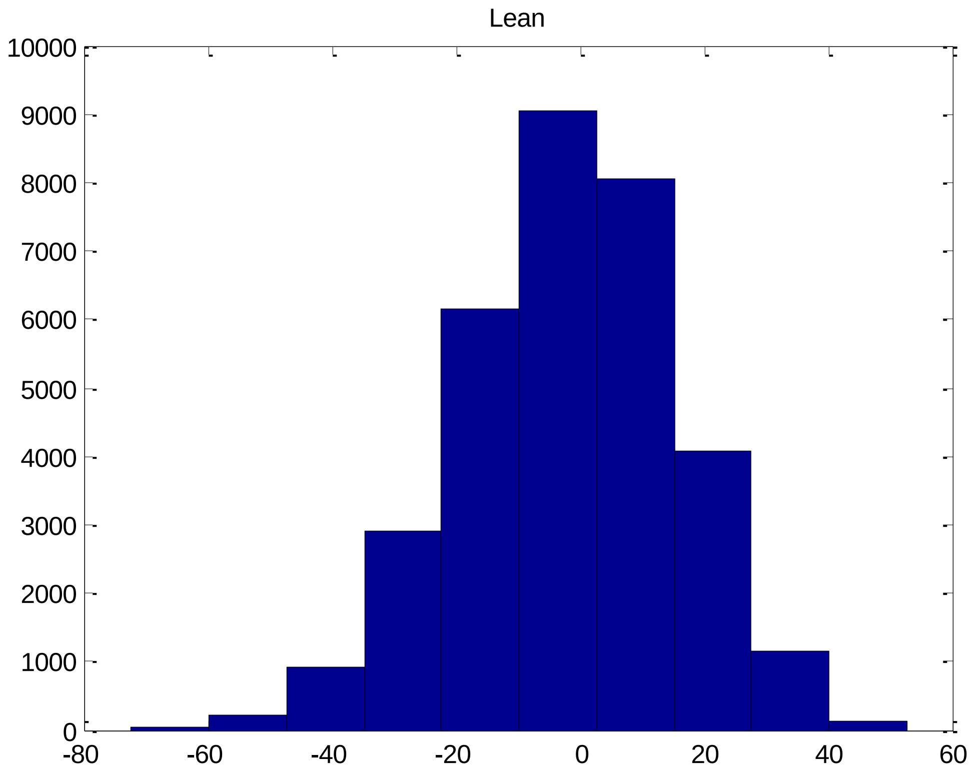


Figure 10: Lean L histogram plot of the difference between the function value generated by the lean method and the original function value.

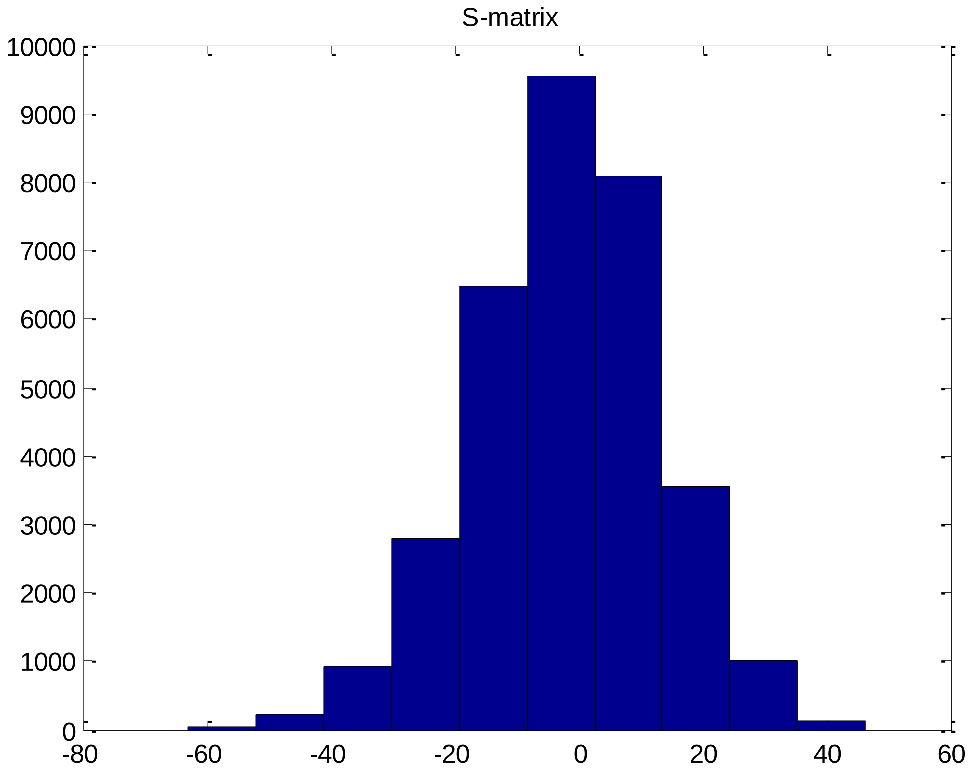


Figure 11: Lean S (spherical matrix) histogram plot of the difference between the function value generated by the spherical method and the original function value.

If the standard deviation for any of these methods are less than the standard deviation for the original spread, than it means that the actual method is better than a guess. Otherwise if the original spread is better (the standard deviation is less than the deviation of the optimization methods), than it is better to guess what the mean value is (which we can say is zero).

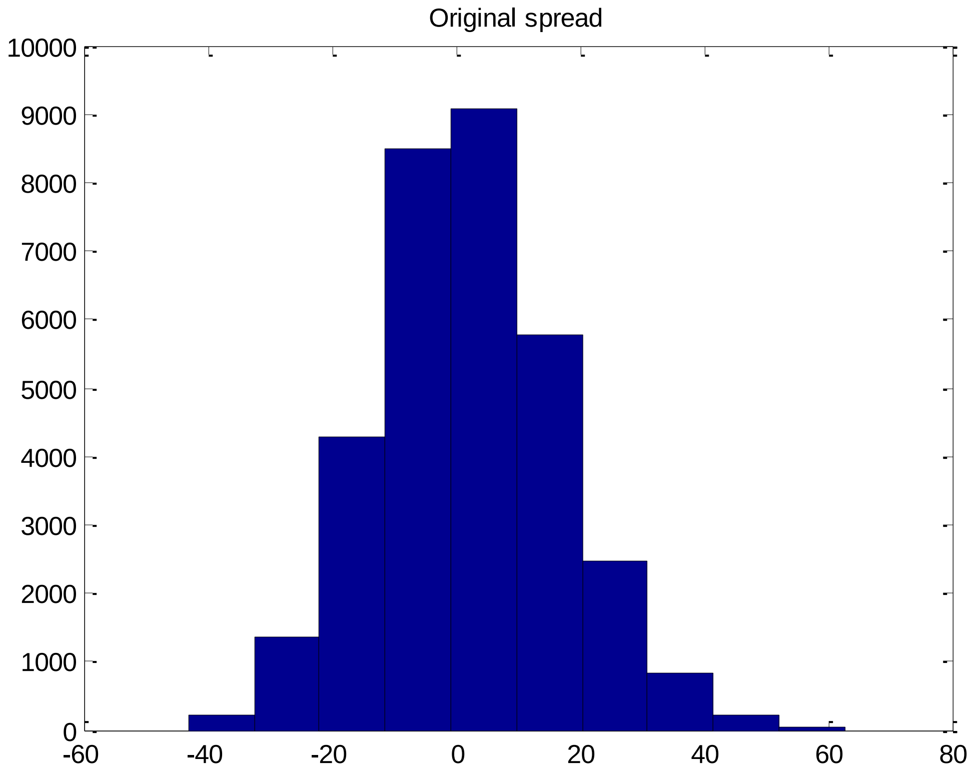


Figure 12: Original spread histogram plot showing the distribution of the original function $y=x^TAx$.

functions	Standard deviation
Original spread	14,8589
Gradient	15,0027
Design of Experiment	20,3617
Lean S	14,9129
Lean L	17,5651

Table 1: illustrating the different standard deviations corresponding to the original function $y=x^TA^TAx$ (original spread) and the difference of the four different optimization functions and the original function. If we look at the standard deviations from this table, it is possible to see that the lowest deviation comes from the original spread. This implies that it is better to guess what the mean value is of how many times the original function value does exist.

4 Optimization of functions

4.1 Gradient method

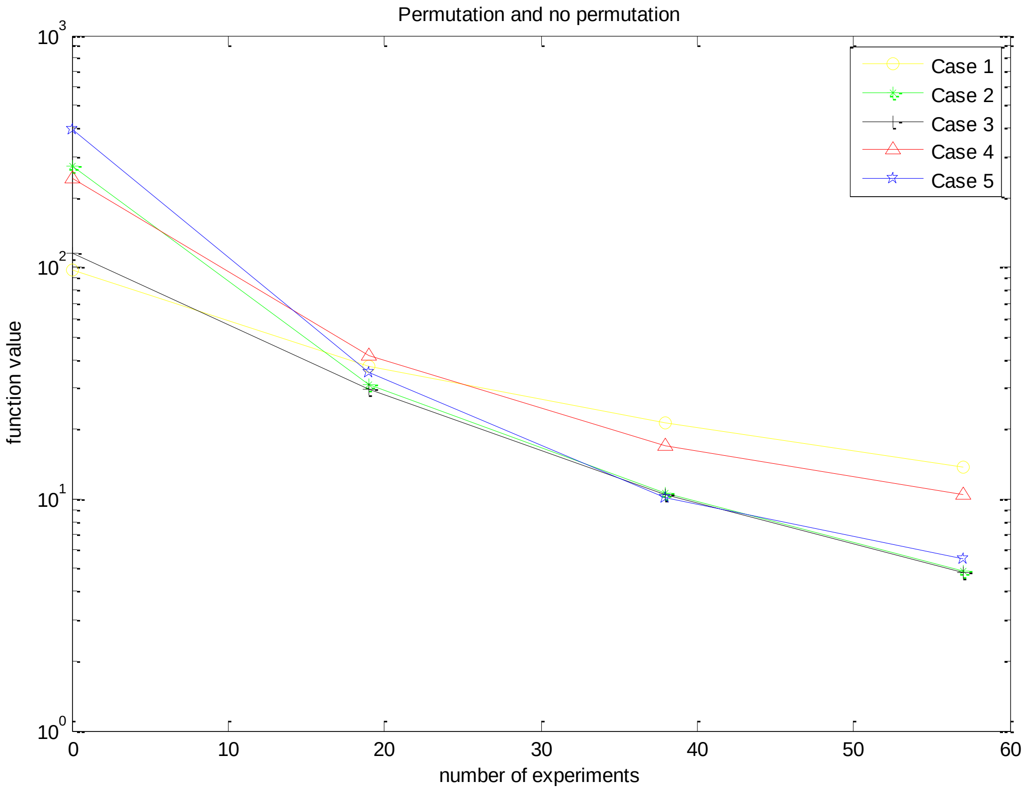


Figure 13: Plot of the gradient method with and without permutation.

In these graphs we are using different matrices A^TA for defining the function $f(x)=x^TA^TAx$, that is what we mean with different cases. In figure 13 we present the result for the gradient method. We are doing the graphs for the other three optimization methods in same fashion. From figure 13 it is possible to distinguish that case 2 and case 3 gives the best optimum value, which should be as close to zero as possible. Note that in the gradient method it does not matter if the elements in the columns are permuted or not, that is why we have only one figure illustrating this optimization method.

As we can see from this figure all the lines are starting from different positions, which means that we are using different matrices for each of the five cases as mentioned before. In all the optimization methods we did totally around 60 experiments. For instance in the gradient method in figure 13 we are taking three steps. Each step corresponds to 19 experiments (number of rows in the design matrix), so totally there will be 57 experiments.

4.2 Design of Experiment

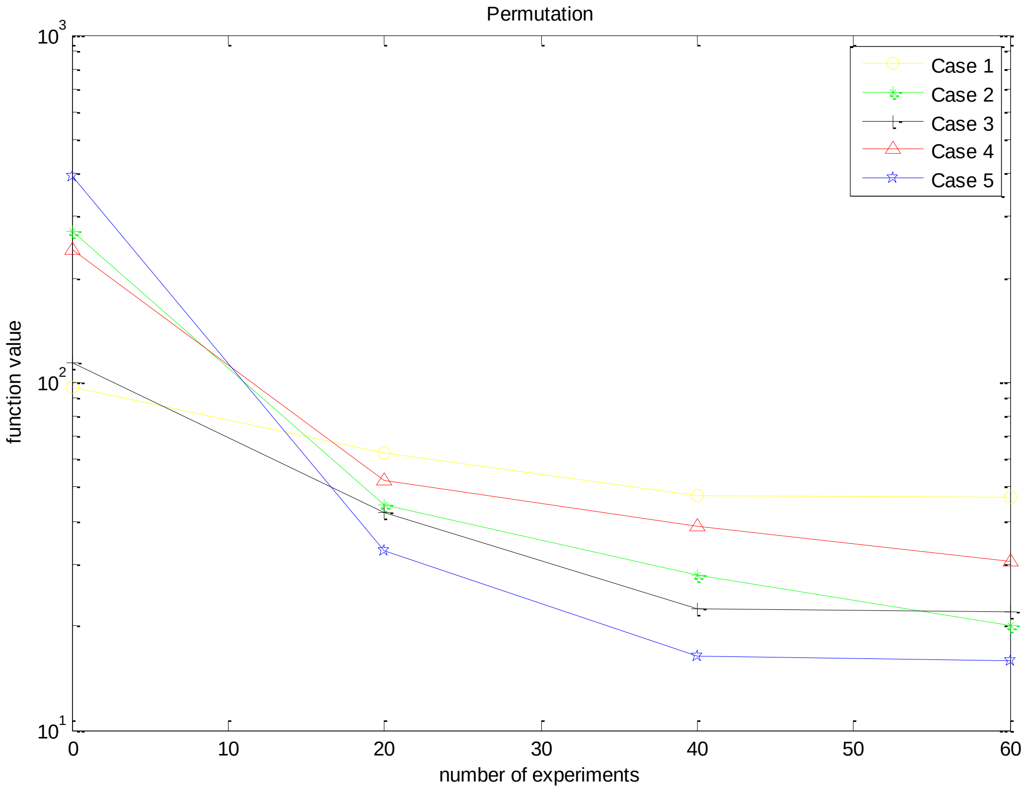


Figure 14: Plot of the Design of Experiment method with permutation (we are changing the positions of columns with each other to make search in new direction)

In the figure 14 it is obvious that case 5 is the best one because it gives us the lowest minimum value. Also case 2 and case 3 seems to be quite good. If we compare this optimization method with the first one, the gradient method, we can see that more lines in the previous figure are below the value 10, which implies that the gradient method seems to give us better optimum values than Design of experiments according to the graphs illustrating the optimization of the functions.

Furthermore we will see that these graphs are better than the following graphs from figure 15, which are not permuted. We are using random permutation.

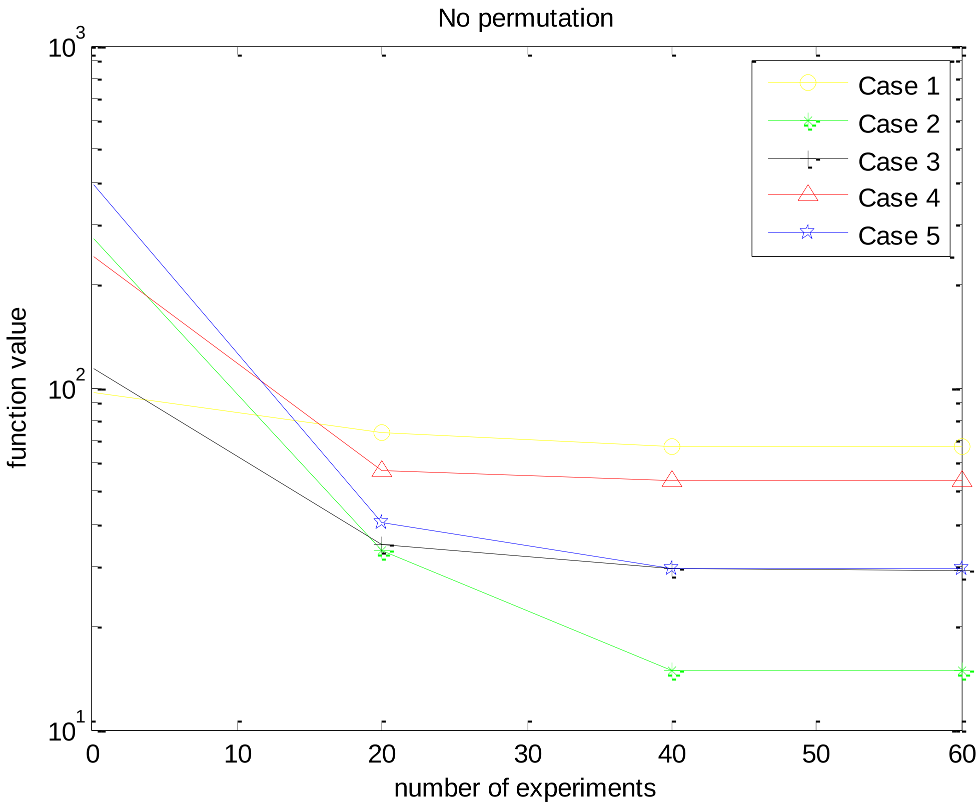


Figure 15: Plot of the Design of Experiment method without permutation**.**

These lines are not permuted, and from the figure we can see that all these lines, except case 2, are higher than the graphs from figure 14.

4.3 Lean L

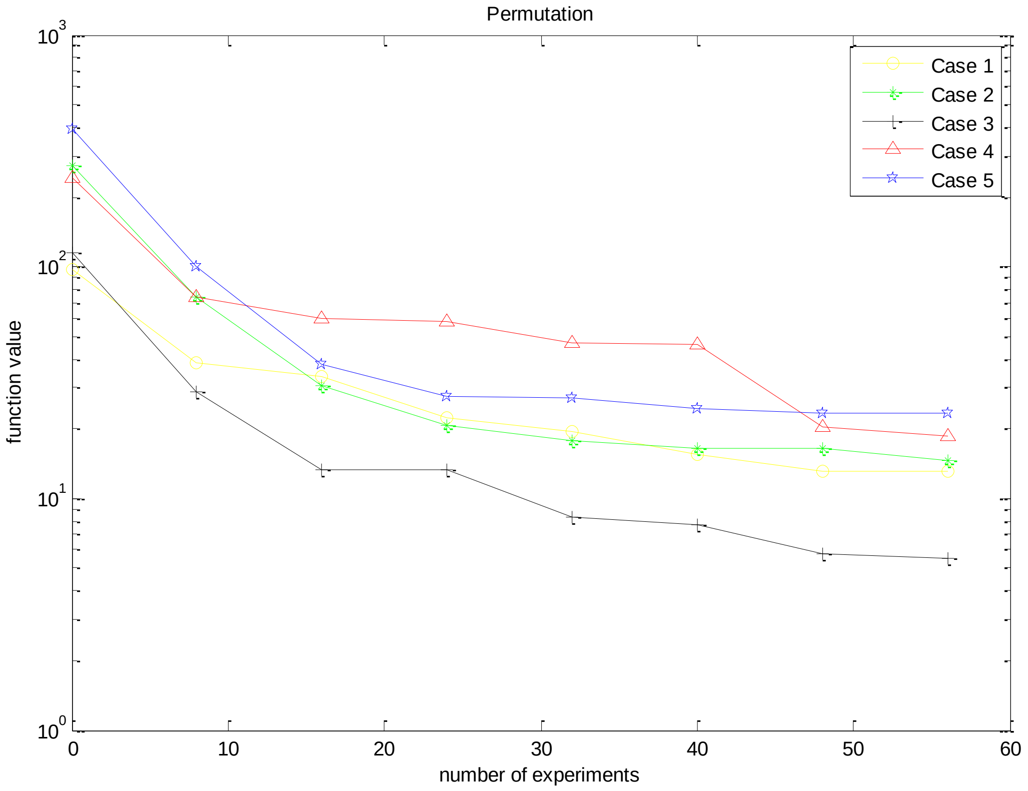


Figure 16: Plot of the Lean method with permutation.

In the lean optimization we can see that only one line is below 10, case 3. The other lines are close to each other and higher than 10.

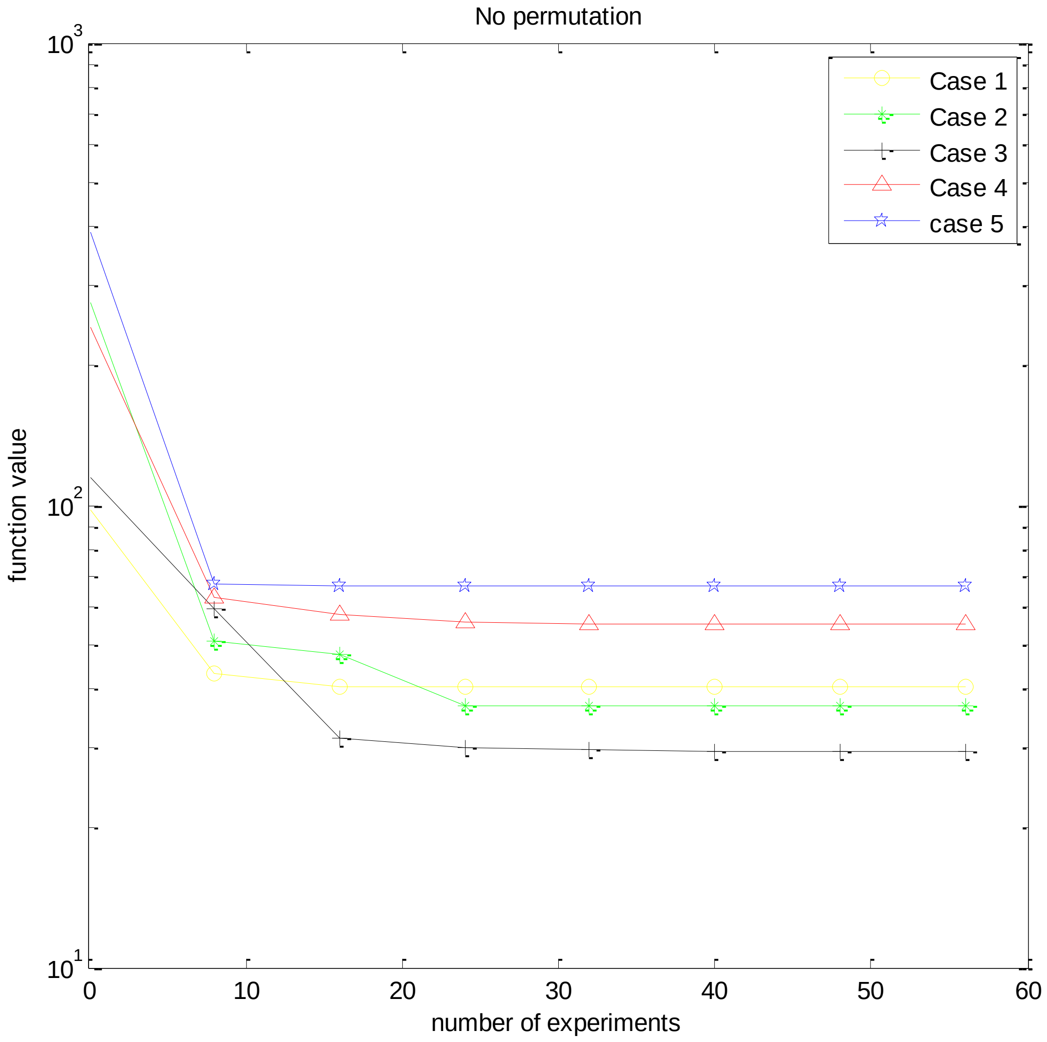


Figure 17: Plot of the Lean method without permutation.

Also in this optimization method we can see that permutation is improving the result. If we compare this figure with figure 16, which was permuted, we will see that this figure is worse than that one because all the lines seems to be higher. As you can see we have better result with the permutation because the lines are decreasing in that case.

4.4 Spherical

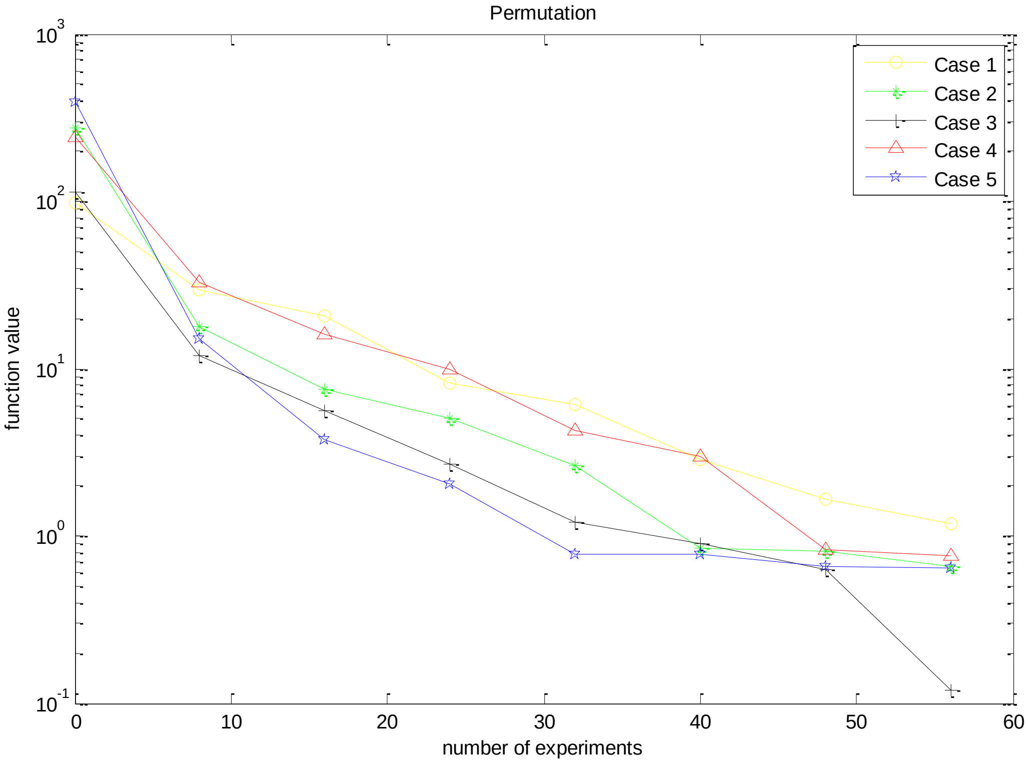


Figure 18: Plot of the spherical method with permutation.

The most important result which we can see from this figure is that all the lines are more close to zero. Especially case 3, which we can see decreases to 0.1, seems to be very good. But also the other lines are better if we compare with the previous three methods, they are all around 1. Therefore we can say that the spheric method is the best method among these four optimization methods.

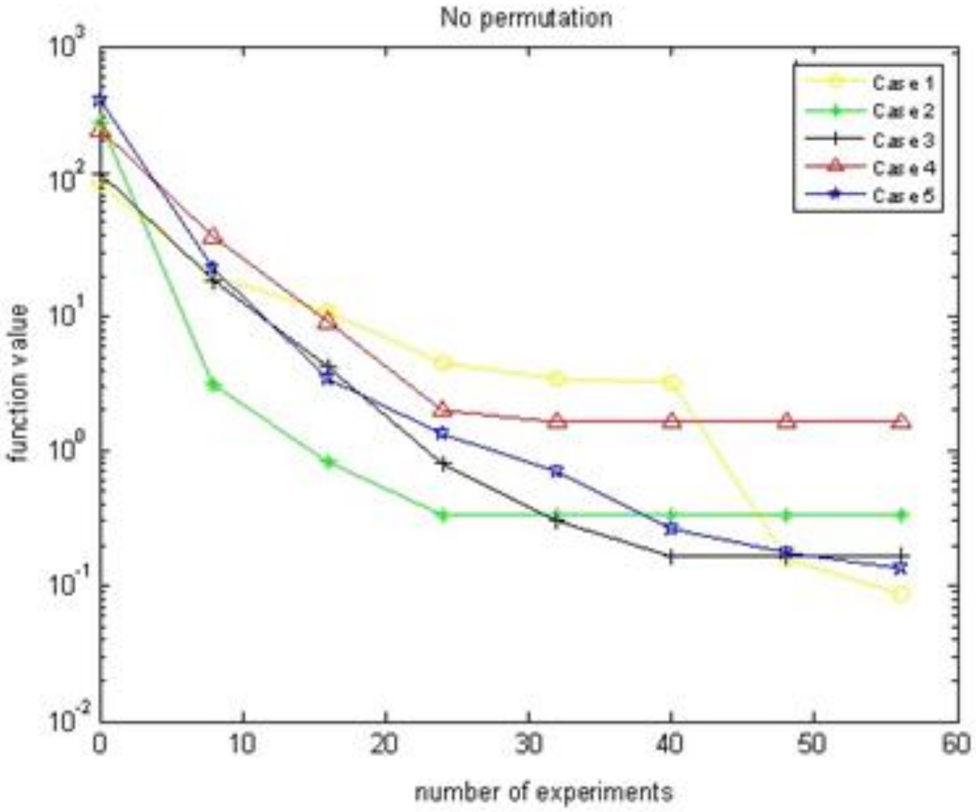


Figure 19: Plot of the spherical method without permutation.

From this figure we can see that all the lines, except case 4, are lower than 1. Case 1 is going to 0.1, which means that this one is best among all these five lines.

Moreover, from these results, it is not possible to see if permutation is improving the result very much. But still we can say that it is not bad to use permutation when doing these optimization methods.

4.5 Gradient, DoE, Lean and Sphere methods with the same positive semi definite matrix

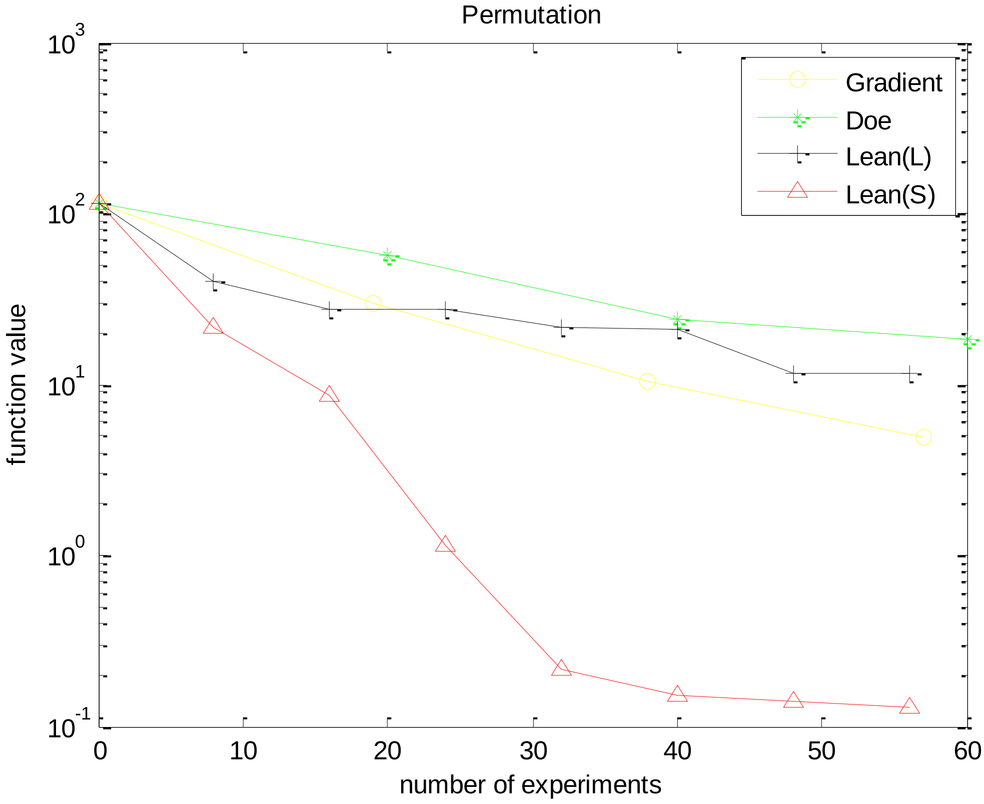


Figure 20: Plot of the four optimization methods (with permutation) with the same positive semi definite matrix.

From figure 20 we can see that the spheric method gives the best result. The gradient method and the DoE method seem to give similar results while the lean method is worse than the other methods in this logarithmic plot.

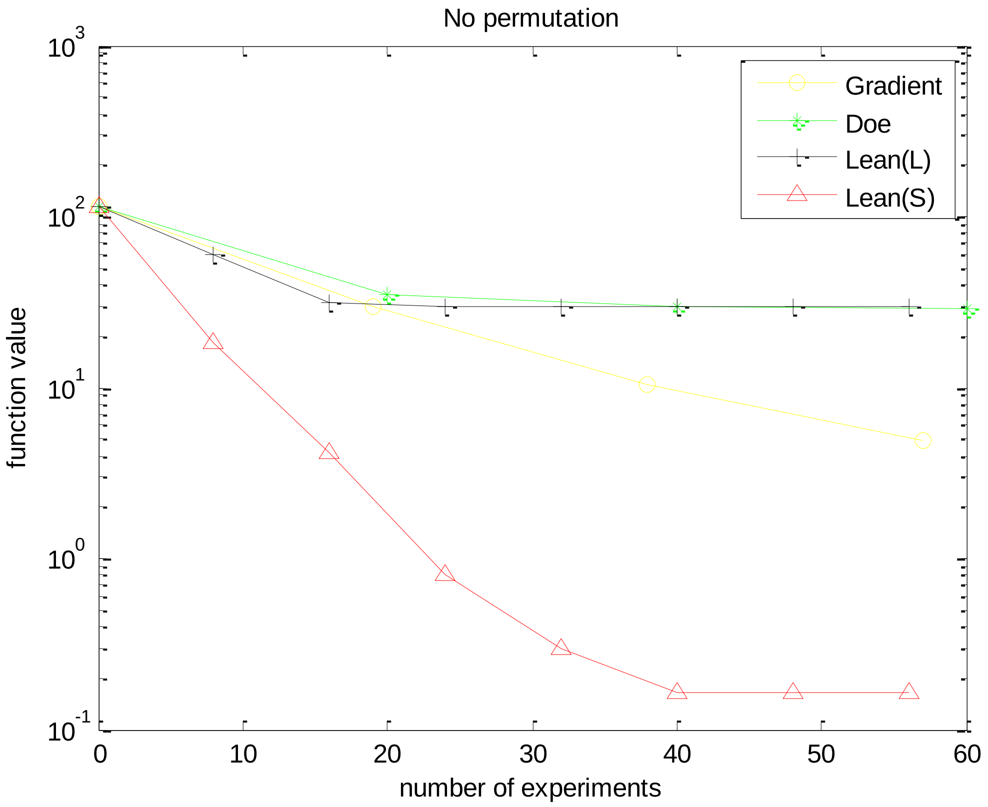


Figure 21: Plot of the four optimization methods (without permutation) with the same positive semi definite matrix.

If we are doing these four optimization methods without permutation it is easier to distinguish that the spherical method is much better, the optimum values goes down to 0.1 which is very close to zero. This result agrees with what we said about the spherical method when we looked at the four optimization methods separately and did a comparison between them.

Moreover from this figure we can see that the gradient method and DoE method also in this case seems to give better function values than the lean method.

5 Conclusion

We did calculate the standard deviation of the difference between prediction and the correct value. According to the histograms gradient method is better than the DoE method, it gives better prediction ability. The standard deviation from the original spread gives the smallest value, which means that it is better to guess what the mean value is of the number of the function value.

From MODDE we could see that the classical DoE has almost perfect fit, which means that the fit goes from zero to one, but does not have any sense for prediction. From SIMCA we could observe that the partial least square method also has perfect fit and some prediction ability as well.

Moreover we can also conclude that the ANOVA-table of the DoE model is not statistically significant, which means that the p-value is less than the significance level. It is probably not good for prediction.

We did fit the quadratic function to the linear model but it did not work for the prediction of corners.

When we did investigate the interval value of the coefficient from DoE respective PLS, we could see that same three whiskers are separated from zero in both methods and that we have improvement in one of the whiskers for PLS.

As can be seen from figure 19 the spherical method did improve the result and we got good optimum value, while the other methods did gave us bad values which was far from zero. For a comparison, see figures 20 and 21.

Permutations, which did not work for the gradient method, are necessary to make search in new direction. In the spherical method, to permute columns means that we are searching in 5 new directions.

6 Discussion

In this report we did see that it is not possible to use the optimization methods for the prediction of corners when we did fit the quadratic function to the linear model. But we got very good optimum value which was very close to zero when we did use the spherical method. That means that it is good to use spherical method in further investigation of optimization methods. Our goal was to reach the minimum value zero, which we almost did with the spherical method.

References

[1] James W. Demmel, Applied Numerical Linear Algebra, SIAM (1997)

[2] I.Siomina, S.Ahlinder, Efficiently Improving a Black Box Model characterized by a Million Parameters, Department of Mathematical Sciences (2009) 2-21

[3] http://mathworld.wolfram.com/Permutation.html

[4] L.Eriksson, E.Johansson, N.Kettaneh-Wold, C.Wikström, S.Wold, Design of Experiments Principles and Applications, Umetrics Academy (2008) 1-25

[5] L.Eriksson, E.Johansson, N.Kettaneh-Wold, S.Wold, Multi- and Megavariate Data Analysis, Umetrics Academy (2001) 71-72, 248-249

[6] A.Rice, Mathematical Statistics and Data Analysis,

University of California (2007) 477-489

```
%linesearchgradient


clear all
clc
load G  %download the G matrix
load A
n=15;
x0=ones(4,n);
X=[0 0 1; 1 1 1; 4 2 1];   %polynomial p(t)=at^2+bt+c
fend=zeros(1,4);
fend(1)=x0(s,:)*A'*A*x0(s,:)';
gg=[];
for s=1:3
        %G(:,:,s+1)=G(:,:,1)+ones(16,1)*x0(s,:);
        G(:,:,s+1)=G(:,randperm(15),1)+ones(16,1)*x0(s,:);    %
        %G(:,:,s+1)=G(:,randperm(15),s+1)
        diag(G(:,:,s+1)*A'*A*G(:,:,s+1)');%grad
        gg(:,s)=diag(G(:,:,s+1)*A'*A*G(:,:,s+1)');%grad
       % sg(:,s)=pinv([ones(size(S(:,:,s+1),1),1),S(:,:,s+1)])*gg(:,s);  %ascent direction
       sg(:,s)=[ones(size(G(:,:,s+1),1),1),G(:,:,s+1)]\gg(:,s);
        s0=-sg(2:end,s);    %descent direction

    u=x0(s,:)+s0'*0;  %move in the gradient direction
    v=x0(s,:)+s0'*1;  %distance from x0(s,:), use coefficients
    w=x0(s,:)+s0'*2;

    fu=u*A'*A*u';  %calculate corresponding function value
    fv=v*A'*A*v';
    fw=w*A'*A*w';
    b(:,s)=[fu  fv fw]';             %three initial functions

    k123=X\b(:,s);                   %coefficients, a, b and c

    t(s)=-k123(2)./(2*k123(1));  %formula: t=-b/(2*a)
   x0(s+1,:)=  x0(s,:)  + t(s) * s0';
  fend(s+1)=x0(s+1,:)*A'*A*x0(s+1,:)';%
end

semilogy([0  19 38 57], fend,'-*')

legend('gradient (G)')
fend
```

```
%gradient  DoE

%linesearchgradient

%x0(s,:)=ones(1,15);
clear all
clc
load F   %download  the G matrix
load A
n=15;
x0=ones(4,n);
X=[0 0 1; 1 1 1; 4 2 1];    %polynomial  p(t)=at^2+bt+c
fend=zeros(1,4);
fend(1)=x0(s,:)*A'*A*x0(s,:)';
gg=[];
for s=1:3
       F(:,:,s+1)=F(:,randperm(15),1)+ones(17,1)*x0(s,:);     %
       %F(:,:,s+1)=F(:,:,1)+ones(17,1)*x0(s,:);     %
       %F(:,:,s+1)=F(:,randperm(15),s+1)
       diag(F(:,:,s+1)*A'*A*F(:,:,s+1)');%grad
       gg(:,s)=diag(F(:,:,s+1)*A'*A*F(:,:,s+1)');%grad
      % sg(:,s)=pinv([ones(size(S(:,:,s+1),1),1),S(:,:,s+1)])*gg(:,s);  %ascent direction
      sg(:,s)=[ones(size(F(:,:,s+1),1),1),F(:,:,s+1)]\gg(:,s);
       s0=-sg(2:end,s);     %descent direction

   u=x0(s,:)+s0'*0;  %move  in the  gradient  direction
   v=x0(s,:)+s0'*1;  %distance  from  x0(s,:),  use  coefficients
   w=x0(s,:)+s0'*2;

   fu=u*A'*A*u';  %calculate  corresponding  function  value
   fv=v*A'*A*v';
   fw=w*A'*A*w';
   b(:,s)=[fu  fv fw]';              %three  initial  functions

   k123=X\b(:,s);                    %coefficients,  a, b and c

   t(s)=-k123(2)./(2*k123(1));  %formula:  t=-b/(2*a)
  x0(s+1,:)=  x0(s,:)  + t(s) * s0';
 fend(s+1)=x0(s+1,:)*A'*A*x0(s+1,:)';%
end
plot([0  20 40 60], fend)
legend('DoE  (F)')
fend
```

```
%lean (L)

%x0(s,:)=ones(1,15);
clear all
%clc
load L  %download the G matrix
load A5
n=15;
x0=ones(4,n);
X=[0 0 1; 1 1 1; 4 2 1];   %polynomial p(t)=at^2+bt+c
fend=zeros(1,4);
s=1;
fend(1)=x0(s,:)*A'*A*x0(s,:)';
gg=[];
for s=1:7
      L(:,:,s+1)=L(:,:,1)+ones(5,1)*x0(s,:);    %
      %L(:,:,s+1)=L(:,randperm(15),1)+ones(5,1)*x0(s,:);    %
      %L(:,:,s+1)=L(:,randperm(15),s+1)
      diag(L(:,:,s+1)*A'*A*L(:,:,s+1)');%grad
      gg(:,s)=diag(L(:,:,s+1)*A'*A*L(:,:,s+1)');%grad
      sg(:,s)=pinv([ones(size(L(:,:,s+1),1),1),L(:,:,s+1)])*gg(:,s);  %ascent direction

      s0=-sg(2:end,s);    %descent direction

   u=x0(s,:)+s0'*0;  %move in the gradient direction
   v=x0(s,:)+s0'*1;  %distance from x0(s,:), use coefficients
   w=x0(s,:)+s0'*2;

   fu=u*A'*A*u';  %calculate corresponding function value
   fv=v*A'*A*v';
   fw=w*A'*A*w';
   b(:,s)=[fu fv fw]';            %three initial functions

   k123=X\b(:,s);                 %coefficients, a, b and c

   t(s)=-k123(2)./(2*k123(1));  %formula: t=-b/(2*a)
  x0(s+1,:)= x0(s,:) + t(s) * s0';
 fend(s+1)=x0(s+1,:)*A'*A*x0(s+1,:)';%
end
plot([0 8 16 24 32 40 48 56], fend)
legend('lean (L)')
fend
```

```
clear all, clc
%n=3;
%n=15;
n=15;
%function [y0,y] =conjugate_DoE(n)
%clc, clear all
%n=3;
%for s=1:100
load('A.mat');


load('M.mat')
xcord=M;
 k=1;
 index=15;
x0=zeros(1,n);
%A=randn(n);
%AA(:,:,s)=A;


% initalization
y0=x0*A*x0';


    for i=1:n

        x(k,:)=x0;
        x(k,i)=x(k,i)+1;
        y(k,i)=x(k,:)*A*x(k,:)';
        g0(k,i)=(y(k,i)-y0)/(x(k,i)-x0(i));    %first element in the gradient

    end






ygrad=g0*xcord';



    x0=zeros(1,index);
    %DoE in this case F=17*15

    %x0=zeros(1,n);
   % x0=zeros(1,index);
    %A=AA(:,:,s);
    y0=x0*A*x0';

    %F=[-1 -1 1; 1 -1 -1; -1 1 -1; 1 1 1];
    F=importdata('data_DoE.csv ');
    %y=[];

    y=[];

    for i=1:17

        x(i,:)=x0;                   %vector with 1 row and n columns
            x(i,:)=x(i,:)+F(i,:);       %above vector + the first row in experimental design

        %x(k,:)=x(k,:)+1;
       % y(k,i)=x(k,:)*A*x(k,:)';
        y(i)=x(i,:)*A*x(i,:)';
         %  y(i)=x(i,:)*F*x(i,:)';
        %y=diag(x*A*x');             %obs! not necessery

        %y=diag(y(k));
        %g0(k,i)=(y(k,i)-y0)/(x(k,i)-x0(i));    %first element in the gradient
        %g0=A.\y;
        %g0=A\y';
%          K=k+1;
    end
    ysave=y;
        %g0=(y-mean(y))
         %size(y)
         %y=y';
         y=y-y0;
         y=y';

         g0(k,:)=(F(1:16,:)\y(1:16))';

yDoE=g0*xcord';
%A=load('A.mat')

for z=1:32768
    yfunc(z)=(xcord(z,:))*A*((xcord(z,:))');
end
close all
 hist((yDoE-yfunc))
 figure
hist((ygrad-yfunc))
```

Printed and Bound at
Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg
2012