Series 8 - Exercise 1

Table of Contents

The objective of this exercise is to train the analysis of a design before and after the measurement. A by-product is the competence about plotting data, fitting models, producing ANOVA tables, performing a canonical analysis.

the tasks to realize are:

1. Load the data
2. Identify if it is a classical design
3. Check what type of model can be fitted
4. Check the statistical quality of the design
5. Infer the coefficients for a linear response model, a linear response model with interaction and a quadratic model. For each situation check the p-values and the LoF
6. Perform a canonical analysis and get some insight of the response function

1. Load data

The data of all the exercises are in the datafile "data_doe.xlsx". You can load them with the help of the specific tool "load data" in the "Home" menu.

Data is put in a table because it allows the presence of get some metadata and also different types of data.

```
data=readtable('data_doe.xlsx',...
    'sheet','Deposition',...
    'Range','A5:F28',...
    'ReadVariableNames',true,...
    'ReadRowNames',true);
disp(data)
```

	x1	x2	x3	x4	Y
1	20.00	13.00	20.00	150.00	51.74
2	10.00	13.00	20.00	150.00	51.46
3	30.00	13.00	20.00	150.00	51.22
4	15.00	10.40	20.00	150.00	51.72
5	25.00	15.60	20.00	150.00	46.07
6	15.00	15.60	20.00	150.00	45.89
7	25.00	10.40	20.00	150.00	51.72
8	15.00	12.13	7.75	150.00	52.33
9	25.00	13.87	32.25	150.00	45.83
10	15.00	13.87	32.25	150.00	48.83
11	20.00	11.27	32.25	150.00	49.90
12	25.00	12.13	7.75	150.00	55.76
13	20.00	14.73	7.75	150.00	49.27
14	15.00	12.13	16.94	110.47	49.48
15	25.00	13.87	23.06	189.53	46.35
16	15.00	13.87	23.06	189.53	46.19
17	20.00	11.27	23.06	189.53	48.83
18	20.00	13.00	10.81	189.53	47.95
19	25.00	12.13	16.94	110.47	48.18
20	20.00	14.73	16.94	110.47	46.04
21	20.00	13.00	29.19	110.47	44.30
22	20.00	13.00	20.00	150.00	50.56
23	20.00	13.00	20.00	150.00	49.87

There is then 23 measurements for 4 factors, and one response.

```
Nexp=size(data,1);
```

2. Identification of the design

The structure of the table and the title of the exercise give hint that the design is a Doehlert design. A quick geometrical analysis will confirm the fist impression.

Let's start by extracting the essay matrix from the data table and normalize it :

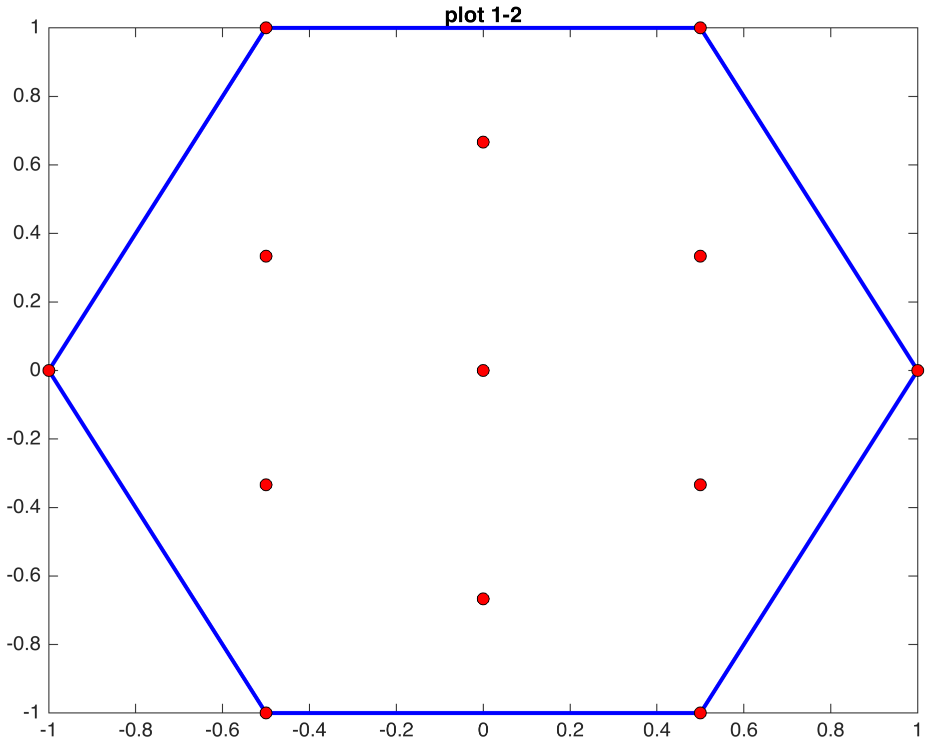
```
Essay=data{:,1:4};
Essay_norm=rescale(Essay,-1,1,"InputMax",max(Essay),"InputMin",min(Essay));
disp(Essay_norm)
```

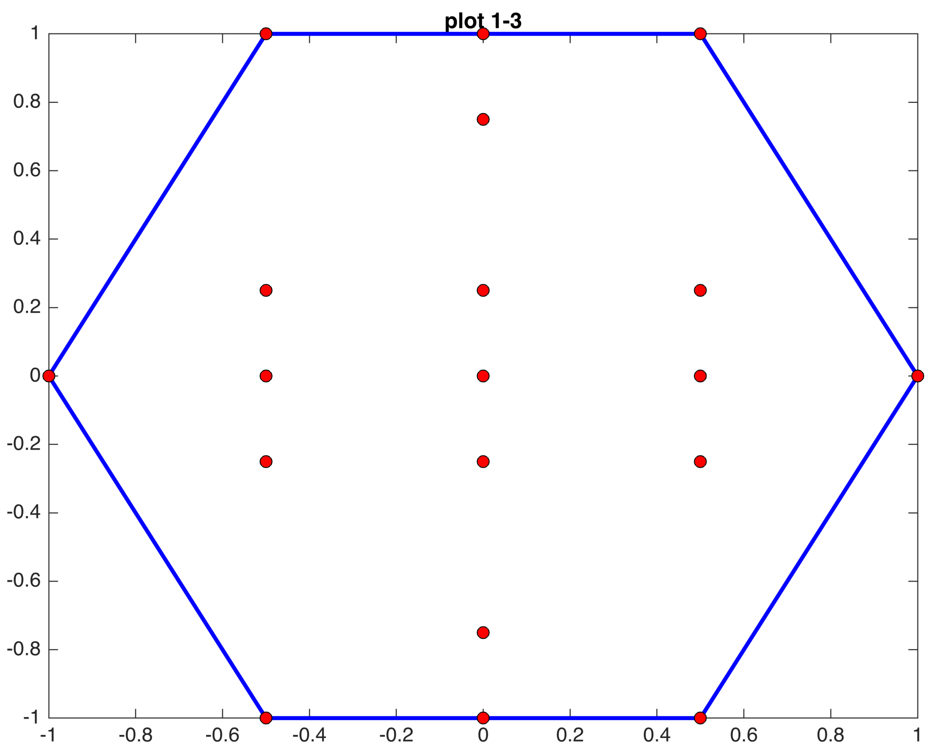
```
        0          0     -0.00          0
    -1.00          0     -0.00          0
```

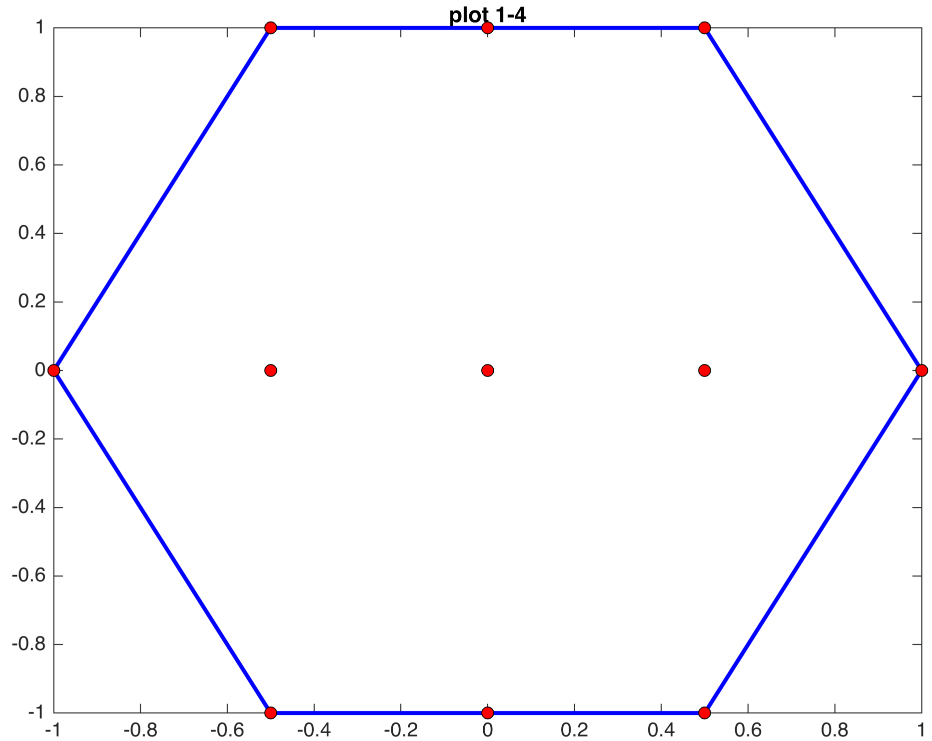
```
   1.00          0       -0.00          0
  -0.50      -1.00       -0.00          0
   0.50       1.00       -0.00          0
  -0.50       1.00       -0.00          0
   0.50      -1.00       -0.00          0
  -0.50      -0.33       -1.00          0
   0.50       0.33        1.00          0
  -0.50       0.33        1.00          0
      0      -0.67        1.00          0
   0.50      -0.33       -1.00          0
      0       0.67       -1.00          0
  -0.50      -0.33       -0.25      -1.00
   0.50       0.33        0.25       1.00
  -0.50       0.33        0.25       1.00
      0      -0.67        0.25       1.00
      0          0       -0.75       1.00
   0.50      -0.33       -0.25      -1.00
      0       0.67       -0.25      -1.00
      0          0        0.75      -1.00
      0          0       -0.00          0
      0          0       -0.00          0
```

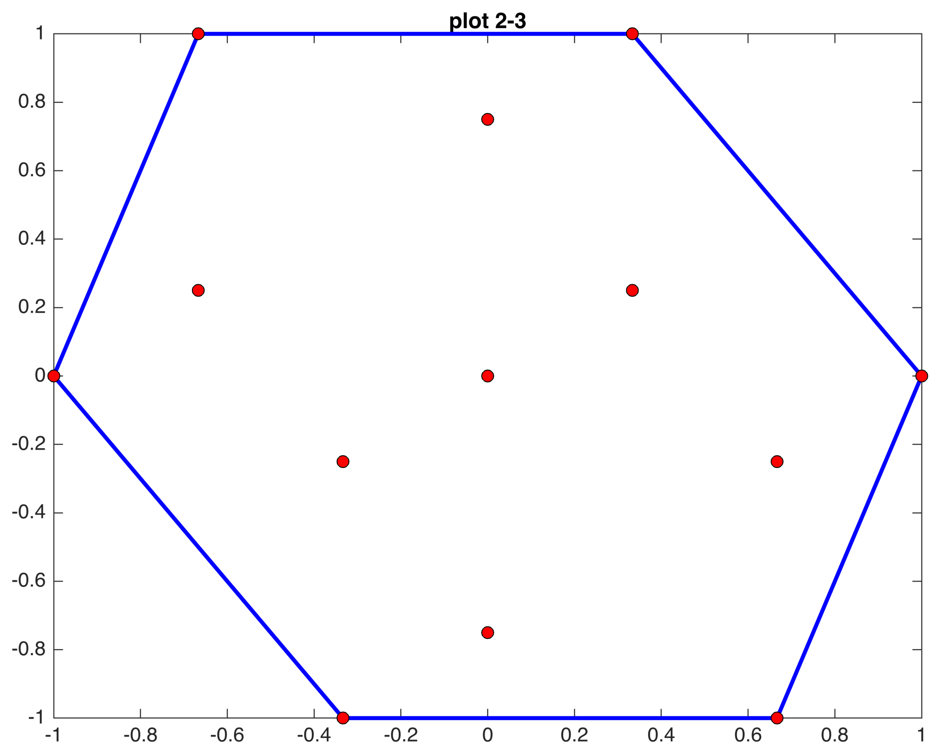
The function *convhull()* will be used to determine the data points defining a convex space. The a hull around the data points is draw in 2D selecting the factors 2 by 2 :

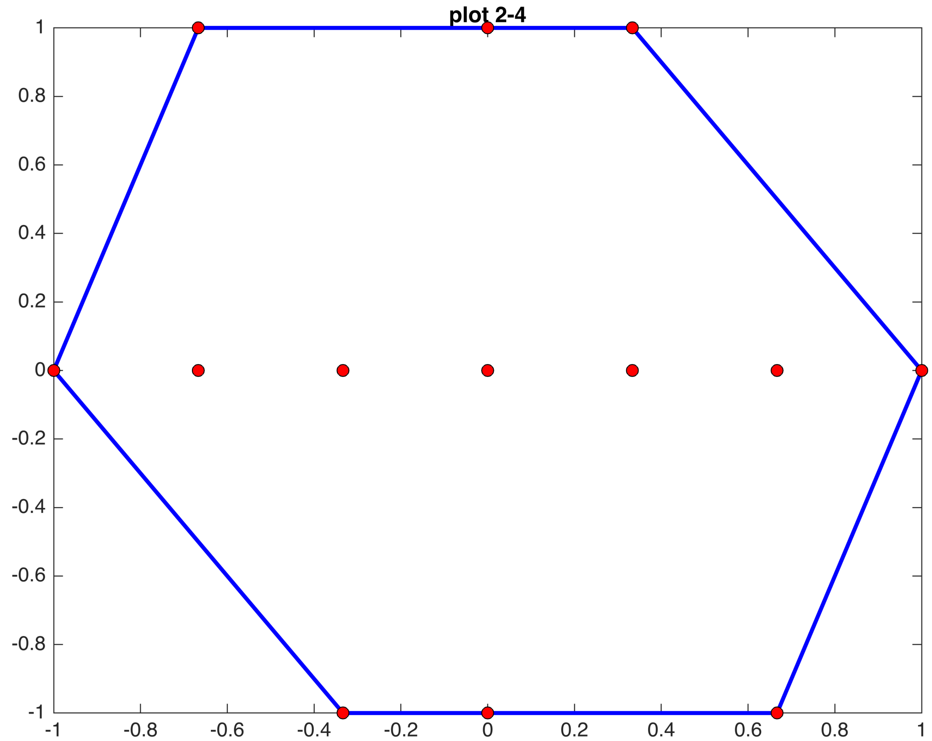
```
for k=1:3
    for l=k+1:4
        figure
        % determining the border points
        Index=convhull(Essay_norm(:,k),Essay_norm(:,l));

        % plotting the profile of the border
        plot(Essay_norm(Index,k),Essay_norm(Index,l),'b-',"LineWidth",2)

        % plotting the data points
        hold on

plot( Essay_norm(:,k),Essay_norm(:,l),'ok',"MarkerFaceColor","red")
        hold off

        title (['plot ', num2str(k), '-', num2str(l)])
    end
end
```

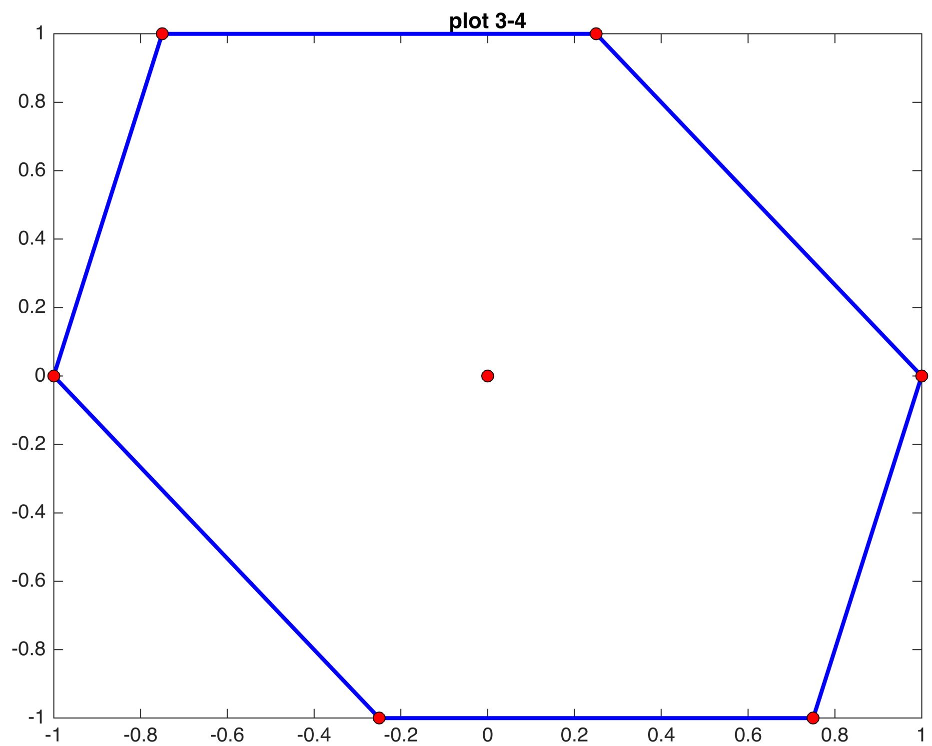

plot 1-2
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1


plot 1-3
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1


plot 1-4
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1 -0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8 1


plot 2-3
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1 -0.8 -0.6 -0.4 -0.2 0 0.2 0.4 0.6 0.8 1


plot 2-4
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1


plot 3-4
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

We may observe profils (projections on a surface) close to hexagons. We may also observe the number of levels of each factor: We are dealing with a Doehlert design for four factors. Matlab do not have a routine for generating this design. You can download the routine doehlert() from Moodle.

A Doehlert design for 4 factors counts 21 runs. Here we have 23 runs

```
% Find unique rows and their indices
[~, uniqueIdx, groupIdx] = unique(Essay_norm, 'rows');

% Count occurrences of each row
counts = accumarray(groupIdx, 1);

% Find rows that appear more than once
duplicateRows = uniqueIdx(counts > 1);

% Display the duplicate rows
disp('Duplicate rows:');
```

```
Duplicate rows:
```

```
disp(Essay_norm(duplicateRows, :));
```

```
            0            0        -0.00            0
```

```
disp('Number of duplications:');
```

```
Number of duplications:
```

```
disp(counts(counts > 1));
```

```
        3.00
```

The center experiment has been replicated 3 times.

3. Determine which model could be fitted with the data

A Doehlert matrix having been identified, the models that can be fitted are

- $y = a_o + \sum a_i x_i + \epsilon$ (linear response)
- $y = a_o + \sum a_i x_i + \sum_{i<j} a_{ij} x_i x_j + \epsilon$ (linear response and interactions)
- $y = a_o + \sum a_i x_i + \sum_{i \leq j} a_{ij} x_i x_j + \epsilon$ (quadratic)

For the 3rd degree, there is definitivelly not a sufficient number of experiments as a Taylor polynome of 3rd degree having 4+4+12+15 =35 coefficients.

Let's build a model matrix X for the quadratic model.

For 4 factors, the quadratic model counts 15 degrees of freedom:

- one constant a_o
- 4 main effects a_i
- $\frac{4!}{2!\ 2!} = 6$ interactions coefficients
- 4 quadratic coefficients

```
X_doehlert=x2fx(Essay_norm,'quadratic');

% the labels corresponding to the coefficients of the second degree model
coeflabels={'a_o','a_1','a_2','a_3','a_4',...
    'a_{12}','a_{13}','a_{14}','a_{23}','a_{24}','a_{34}',...
    'a_{11}','a_{22}','a_{33}','a_{44}'};
```

4. Statistical quality of the design

Dispersion matrix

The analysis of the quality of a design starts with the computazion of the dispersion matrix.

```
% dispersion matrix of the doehlert design
D_doehlert=inv(X_doehlert'*X_doehlert);
```

The composite design as a reference

The composite design is recognised as a good design for response surface. So let's compare the performance of the two designs.

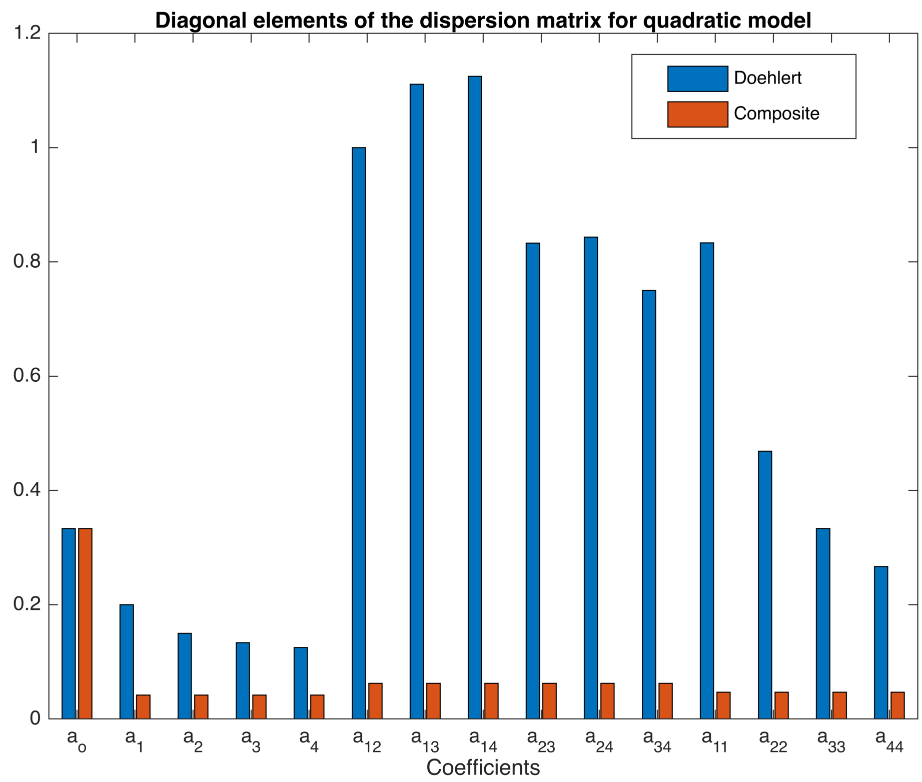
```
% Computing the composite matrix as a reference
E_composite=ccdesign(4,'center',3);
X_composite=x2fx(E_composite,'quadratic');
D_composite=inv(X_composite'*X_composite);
```

The figure below compare the diagonal elements of the two designs which are directly related to the size of the confidence intervals.

- Doehlert: 21+2 runs
- Composite: 25+2 runs

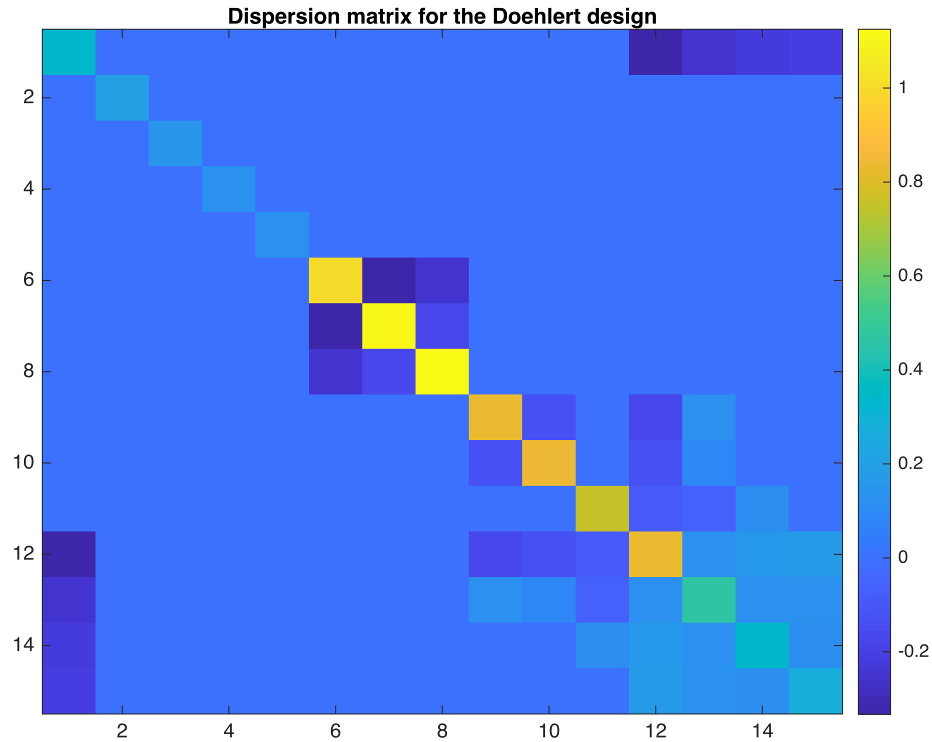
We observe that the quality of the Doehlert design is lower than the quality of the composite design. For the interaction and the quadratic coefficients the ratio is between 10 and 5 with only 4 runs of difference.

```
% A bar chart of the diagonal elements of the dispersion matrix
figure
bar([diag(D_doehlert),diag(D_composite)])
title('Diagonal elements of the dispersion matrix for quadratic model')
legend('Doehlert','Composite')
xticklabels(coeflabels)
xlabel('Coefficients')
legend("Position", [0.65,0.8,0.2,0.1])
```

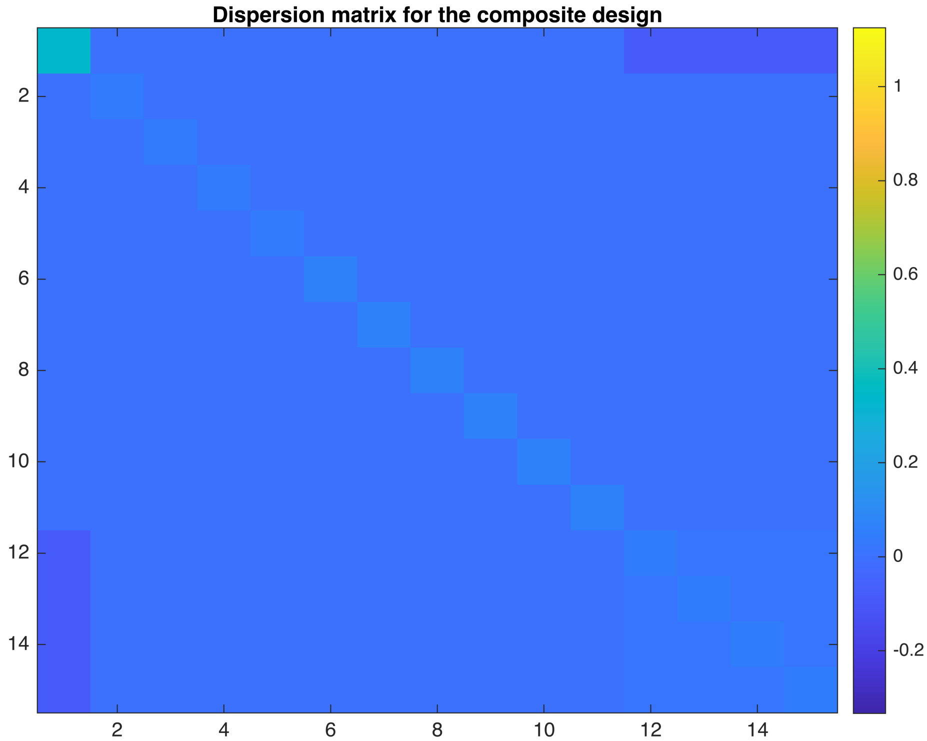



```
% Compute the common color scale limits
minValue = min([D_doehlert(:); D_composite(:)]);
maxValue = max([D_doehlert(:); D_composite(:)]);

% heatmap of the dispersion matrix
figure
imagesc(D_doehlert)
colorbar
title('Dispersion matrix for the Doehlert design')


% Set color limits to be the same for both plots
```

```
clim([minValue maxValue]);
```




```
% heatmap of the dispersion matrix
figure
imagesc(D_composite)
colorbar
title('Dispersion matrix for the composite design')

% Set color limits to be the same for both plots
clim([minValue maxValue]);
```



Let's now have a look to the whole dispersion matrix:

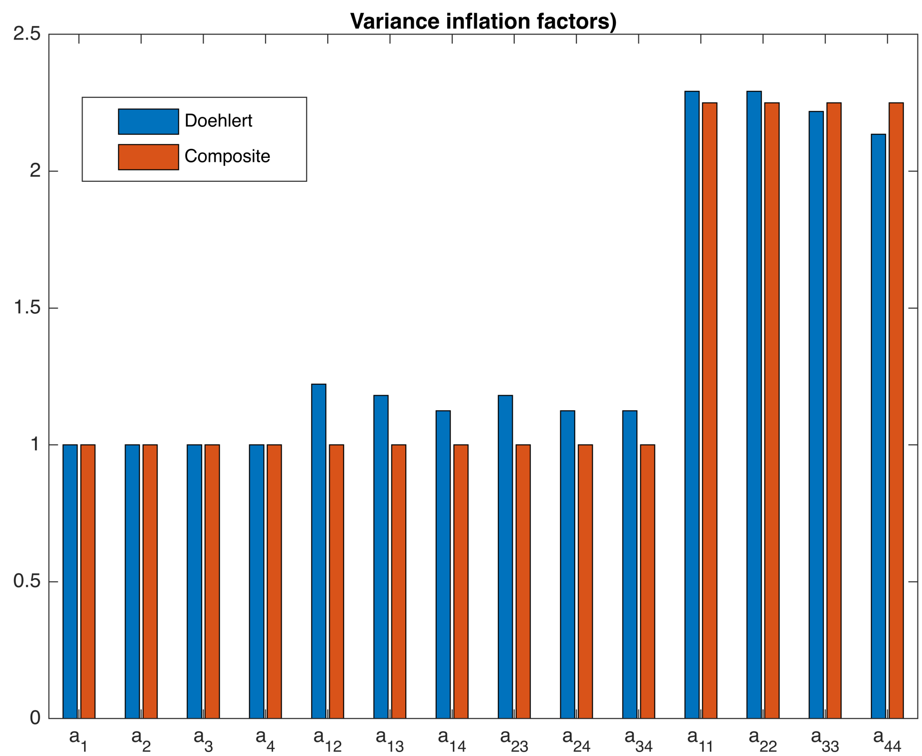
- We may observe a dispersion matrix standard for a second degree model with diagonal element of medium quality and covariances between quadratic coefficients and the constant.
- The Doehlert design is economical but its quality stay poor. It is its versatility that make it very useful.
- By comparison, the corresponding composite design with 27 experiments would have a dispersion matrix of a lot better quality as shown in the second image where covariance are only with the constant.

Variance Inflation factors

The inflation factors give an indication of the consequences of colinearities. It can be observed a few differences with the composithe design which is slightly better for the interaction coefficients and quite equivalent for the quadratic terms.

```
% correlation matrix of the Doehlert design
C_doehlert=corrcov(D_doehlert);

% VIF matrix of the Doehlert design
VIF_doehlert=diag(inv(C_doehlert));

% correlation matrix of the composite design
C_composite=corrcov(D_composite);
```

```
% VIF matrix of the composite design
VIF_composite=diag(inv(C_composite));

% comparison bar chart
figure
bar([VIF_doehlert(2:end),VIF_composite(2:end)])
title('Variance inflation factors)')
xticklabels({'a_1','a_2','a_3','a_4',...
    'a_{12}','a_{13}','a_{14}','a_{23}','a_{24}','a_{34}',...
    'a_{11}','a_{22}','a_{33}','a_{44}',})
legend('Doehlert','Composite')
legend("Position", [0.16,0.75,0.2,0.1])
```



All these elements show that the the quality of the Doehlert design is poor in comparison of the composite. When ever possible the latter schould be prefered.

5. Half effects

Let's compute now the half-effects for a quadratic model with the least square algorithm

$\alpha = (X^T X)^{-1} X^T Z Y$

the residuals

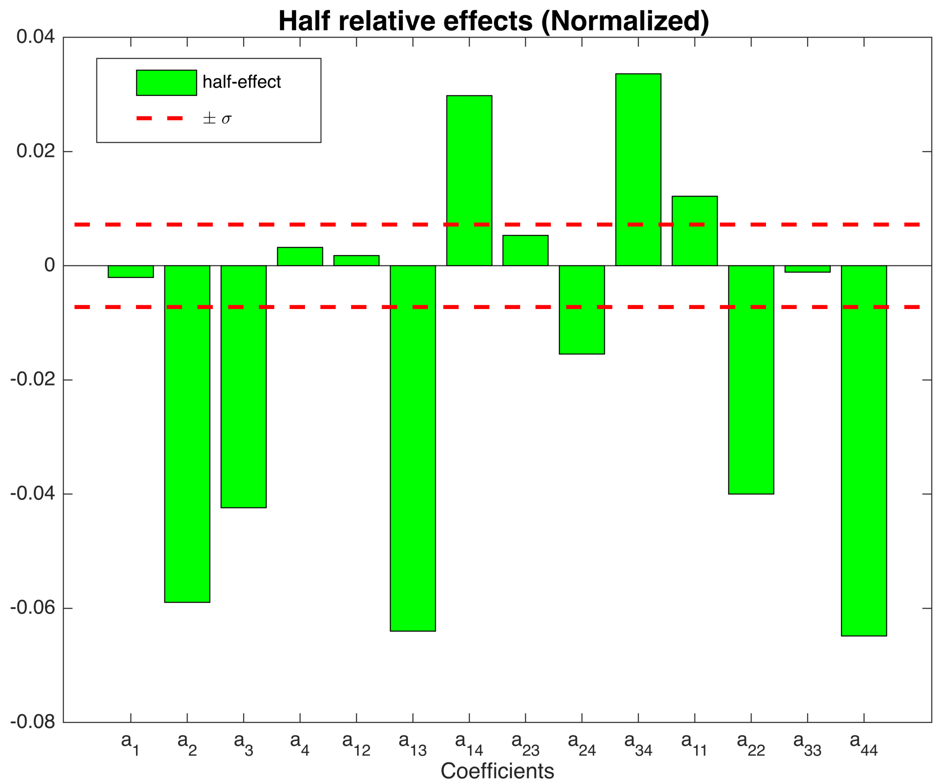
$$\epsilon = Y - \widehat{Y} = Y - X\alpha$$

and the experimental variance

$$\sigma^2 = \frac{\epsilon \cdot \epsilon}{N_{exp} - 1}$$

These calculations, for the normalised essay matrix, can be sumerize in the following bar chart:

```
% LSF estimates of the coefficients
coef=D_doehlert*X_doehlert'*data{:,5};

% Residuals
epsilon=data{:,5}-X_doehlert*coef;

% Experimental variance
s2=sumsqr(epsilon)/(Nexp-1);
s=sqrt(s2);

% Bar chart of the half-effects
figure
bar(coef(2:end)/coef(1),"green")
title('Half relative effects (Normalized) ','FontSize',14)
label={'a_1','a_2', 'a_3','a_4',...
    'a_{12}','a_{13}','a_{14}',...
    'a_{23}','a_{24}',...
    'a_{34}',...
    'a_{11}','a_{22}','a_{33}','a_{44}'};
set(gca,'XTickLabel',label)
xlabel('Coefficients')

% +/- sigma range
hold on
plot([0 15],[s/coef(1) s/coef(1)],'r--',"LineWidth",2)
plot([0 15],[-s/coef(1) -s/coef(1)],'r--',"LineWidth",2)
hold off

% legend
legend('half-effect','\pm \sigma')
legend("Position", [0.16,0.8,0.2,0.1])
```



5.2 Significant coefficients

Reference distribution

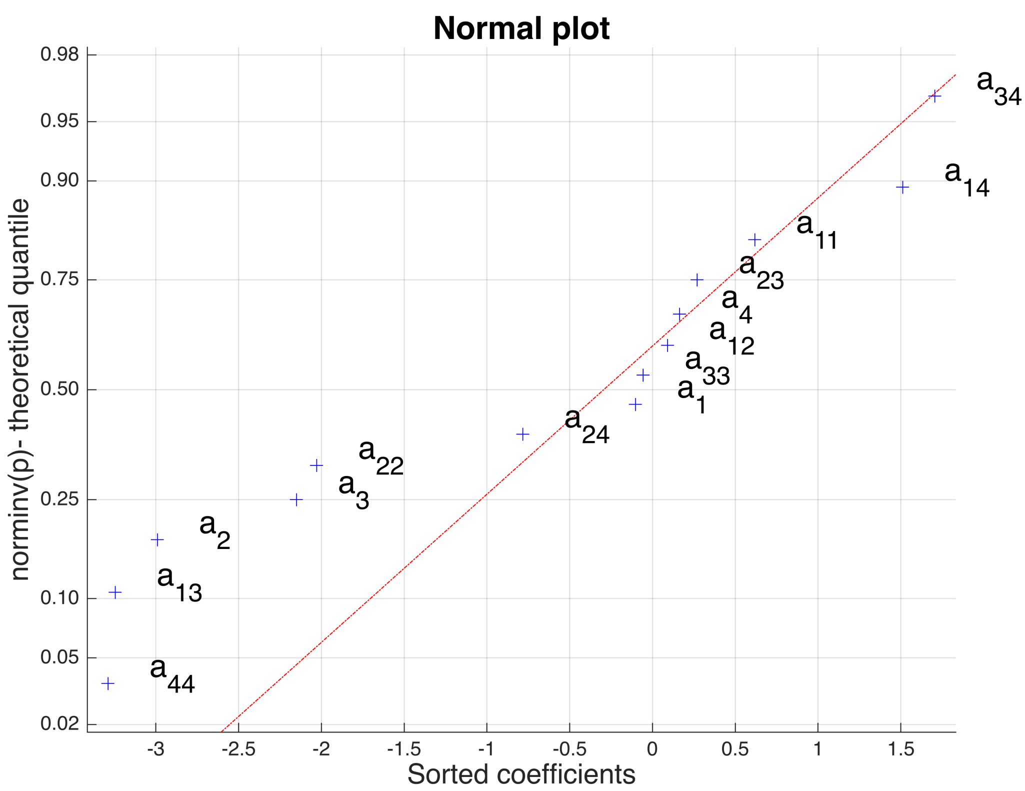
As there is sufficient degrees of freedom, an experimental variance has been computed and then if necessary a first screening can be done with the $\pm\sigma$ interval as shown on the bar chart of the half-effects with the two dashed red lines.

Normal plot

A normal plot can also be used for the screening of significant coefficients:

```
figure
h=normplot(coef(2:end));
title('Normal plot','FontWeight','bold','FontSize',16)
xlabel('Sorted coefficients','FontSize',14)
ylabel('norminv(p)- theoretical quantile','FontSize',14)


% draw again the sigmoid based on the coefficients not retained

% get the abcisses (coefficients)
x=get(h(1),'Xdata')';
```

```
% model matrix for the coefficients not retained
x=[ones(7,1),x(6:12)];

% get the ordinates (probabilities)
y=get(h(1),'Ydata')';
y=y(6:12);

% strait line regression
alpha=inv(x'*x)*x'*y;

% delete the other strait line
delete(h(2))

% draw again  the red line
set(h(3),'Xdata',[-4,2],'Ydata', [1 -4;1 2]*alpha)


[Essay,index]=sort(coef(2:end));
text(h(1).XData+.25, h(1).YData+.05, label(index),'FontSize',16)
```



The following coefficients have been retained :

$$a_o, a_2, a_3, a_{13}, a_{14}, a_{34}, a_{22}, a_{44}$$

The model is then

$$\hat{y} = a_o + a_2x_2 + a_3x_3 + a_{13}x_1x_3 + a_{14}x_1x_4 + a_{34}x_3x_4 + a_{22}x_2^2 + a_{44}x_4^2$$

Stepwise fit

An alternative is the routine stepwiselm():

```
mdlstw=stepwiselm(Essay_norm,data{:,5},'quadratic','lower','constant','Pente
r',0.03,'PRemove',0.05)
```

```
1. Removing x1:x2, FStat = 0.022395, pValue = 0.88475
2. Removing x3^2, FStat = 0.029404, pValue = 0.86764
3. Removing x2:x3, FStat = 0.29368, pValue = 0.59974
4. Removing x2:x4, FStat = 2.4083, pValue = 0.14897
5. Removing x1^2, FStat = 1.4877, pValue = 0.246
mdlstw =
Linear regression model:
    y ~ 1 + x2 + x1*x3 + x1*x4 + x3*x4 + x2^2 + x4^2

Estimated Coefficients:
```

	Estimate	SE	tStat	pValue
(Intercept)	50.87	0.19	265.90	0.00
x1	−0.10	0.25	−0.41	0.69
x2	−2.99	0.22	−13.60	0.00
x3	−2.15	0.21	−10.37	0.00
x4	0.16	0.20	0.82	0.43
x1:x3	−3.22	0.57	−5.66	0.00
x1:x4	1.53	0.59	2.62	0.02
x3:x4	1.84	0.47	3.91	0.00
x2^2	−2.03	0.35	−5.89	0.00
x4^2	−3.37	0.26	−13.10	0.00

```
Number of observations: 23, Error degrees of freedom: 13
Root Mean Squared Error: 0.568
R-squared: 0.975,  Adjusted R-Squared: 0.958
F-statistic vs. constant model: 57.2, p-value = 5.42e-09
```

The obtained model is the same as long as the input criteria are standard. We may observe an anomaly in the routine that get rid of the terms a_1 and a_4 although the high p-values élevées and the 'lower' parameter defined as 'constant'.

ANOVA (Type II)

To perform a precise ANOVA, a term matrix can be defined and then a type II ANOVA algorithm is applied

```
% define the spec matrix
spec=[0 0 0 0
    0 1 0 0
    0 0 1 0
    1 0 1 0
    1 0 0 1
    0 0 1 1
    0 2 0 0
    0 0 0 2];
```

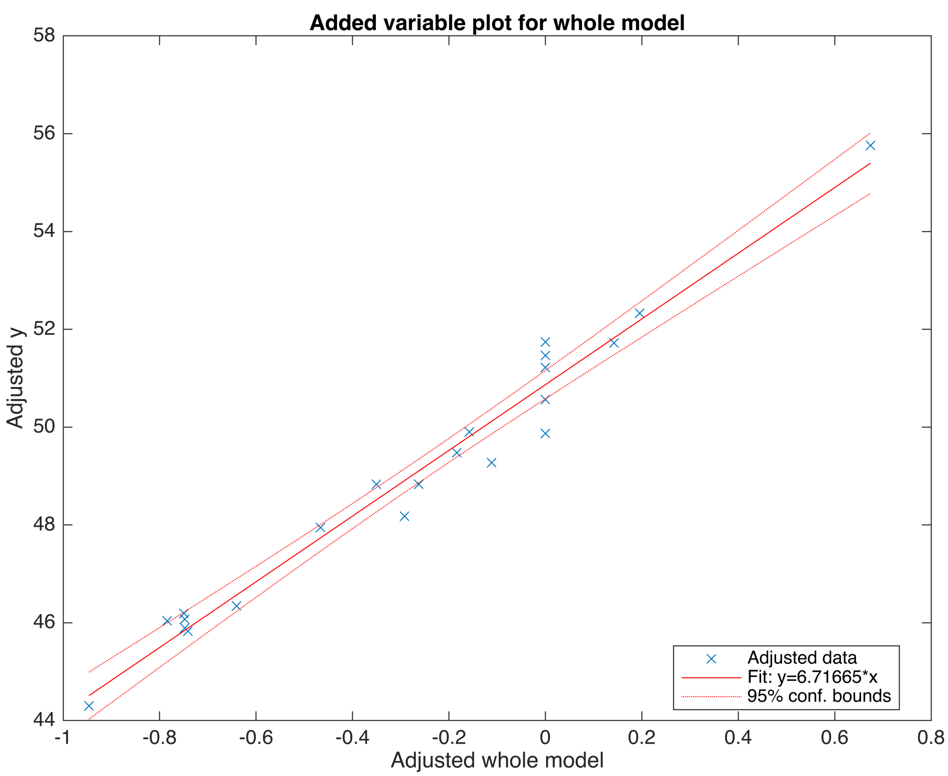
```
disp(spec)
```

```
        0          0          0          0
        0       1.00          0          0
        0          0       1.00          0
     1.00          0       1.00          0
     1.00          0          0       1.00
        0          0       1.00       1.00
        0       2.00          0          0
        0          0          0       2.00
```

```
% Fit the model with specific terms
mdl=fitlm(Essay_norm,data{:,5},spec);

% compute the ANOVA table for each coefficient of the model
A_mdl=anova(mdl,'component',2);
disp(A_mdl)
```

	SumSq	DF	MeanSq	F	pValue
x2	59.63	1.00	59.63	200.60	0.00
x3	34.67	1.00	34.67	116.62	0.00
x1:x3	10.34	1.00	10.34	34.79	0.00
x1:x4	2.22	1.00	2.22	7.45	0.02
x3:x4	4.94	1.00	4.94	16.61	0.00
x2^2	11.16	1.00	11.16	37.55	0.00
x4^2	55.30	1.00	55.30	186.03	0.00
Error	4.46	15.00	0.30		

```
plot(mdl)
```



A lack of fit can be performed with the option *'summary'* of the anova routine. A p-value of 96% indicates that there is no detectable lack of fit.

```
A_summary=anova(mdl,'summary');
disp(A_summary)
```

```
                     SumSq      DF      MeanSq       F       pValue
                     ______    _____    ______    ______    ______

    Total            170.07    22.00      7.73
    Model            165.61     7.00     23.66     79.59     0.00
    . Linear          94.30     2.00     47.15    158.61     0.00
    . Nonlinear       71.31     5.00     14.26     47.98     0.00
    Residual           4.46    15.00      0.30
    . Lack of fit      2.68    13.00      0.21      0.23     0.96
    . Pure error       1.78     2.00      0.89
```

Let's check what would have been with a simpler model without quadratic term

```
mdl_lin=fitlm(Essay_norm,data{:,5},'linear');
disp(mdl_lin)
```

```
Linear regression model:
    y ~ 1 + x1 + x2 + x3 + x4

Estimated Coefficients:
                   Estimate     SE      tStat     pValue
                   ________    ____    ______    ______

    (Intercept)      49.11     0.43    114.99    0.00
    x1               -0.10     0.92     -0.11    0.91
    x2               -2.99     0.79     -3.77    0.00
    x3               -2.15     0.75     -2.87    0.01
    x4                0.16     0.72      0.23    0.82


Number of observations: 23, Error degrees of freedom: 18
Root Mean Squared Error: 2.05
R-squared: 0.556,  Adjusted R-Squared: 0.457
F-statistic vs. constant model: 5.64, p-value = 0.00402
```

```
A_lin=anova(mdl_lin,'summary')
```

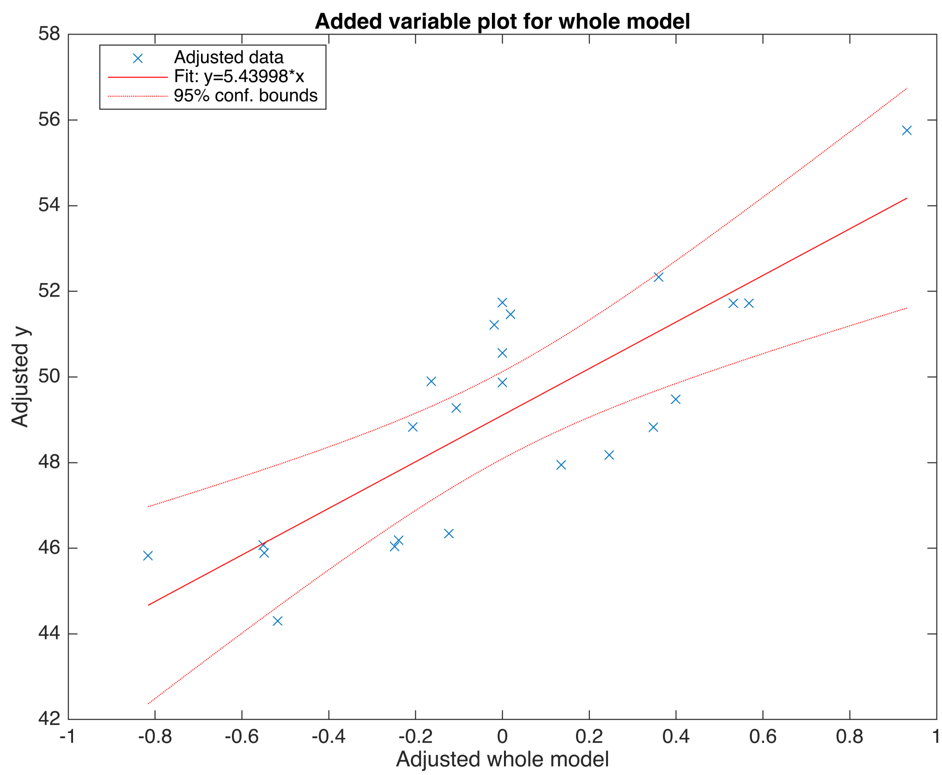
A_lin = 5×5 table

	SumSq	DF	MeanSq	F	pValue
1 Total	170.07	22	7.73	NaN	NaN
2 Model	94.57	4	23.64	5.64	0
3 Residual	75.50	18	4.19	NaN	NaN
4 . Lack of fit	73.72	16	4.61	5.17	0.17
5 . Pure error	1.78	2	0.89	NaN	NaN

```
mdl_int=fitlm(Essay_norm,data{:,5},'interactions');
disp(mdl_int)
```

```
Linear regression model:
    y ~ 1 + x1*x2 + x1*x3 + x1*x4 + x2*x3 + x2*x4 + x3*x4

Estimated Coefficients:
                   Estimate     SE      tStat      pValue
                   ________    ____    ______    ______

    (Intercept)      49.11     0.47    105.23      0.00
    x1               -0.10     1.00     -0.10      0.92
    x2               -2.99     0.87     -3.45      0.00
    x3               -2.15     0.82     -2.63      0.02
    x4                0.16     0.79      0.21      0.84
    x1:x2             0.09     2.24      0.04      0.97
    x1:x3            -3.25     2.36     -1.38      0.19
    x1:x4             1.51     2.37      0.64      0.54
    x2:x3             1.10     1.94      0.57      0.58
    x2:x4            -0.16     2.00     -0.08      0.94
    x3:x4             1.39     1.83      0.76      0.46


Number of observations: 23, Error degrees of freedom: 12
Root Mean Squared Error: 2.24
R-squared: 0.647,  Adjusted R-Squared: 0.352
F-statistic vs. constant model: 2.2, p-value = 0.099
```

```
plot(mdl_int)
```




```
A_int=anova(mdl_int,'summary');
disp(A_int)
```

	SumSq	DF	MeanSq	F	pValue
Total	170.07	22.00	7.73		
Model	109.97	10.00	11.00	2.20	0.10
. Linear	94.57	4.00	23.64	4.72	0.02
. Nonlinear	15.40	6.00	2.57	0.51	0.79
Residual	60.10	12.00	5.01		
. Lack of fit	58.32	10.00	5.83	6.55	0.14
. Pure error	1.78	2.00	0.89		

Let's summerize the situation

```
% build a table with

Tmodel=categorical({'Linear';'Interaction';'Quadratic'});
%Tsignificant={'2,3';'2,3';'2,3,13,14,34,22,44'};
Tsignificant={[2,3];[2,3];[2,3,13,14,34,22,44]};
Tpval=[0.004;0.01;5e-9];
TloF=[0.17;0.14;0.96];

Tsummary=table(Tmodel,Tsignificant,Tpval,TloF,...
    'VariableNames',{'Model','Significant Terms','Model p-value','LoF'});
format shortG
disp(Tsummary)
```

Model	Significant Terms	Model p-value	LoF
Linear	{[2 3]}	0.004	0.17
Interaction	{[2 3]}	0.01	0.14
Quadratic	{[2 3 13 14 34 22 44]}	5e-09	0.96

```
format short
```

Summary of Model Comparison

Three models were evaluated to analyze experimental data collected using a Doehlert design with four factors: a linear model, an interaction model, and a quadratic model. Each model demonstrated strong statistical validity, as indicated by the high p-values for the lack of fit test, suggesting that none of the models failed to represent the observed data systematically.

Linear Model: The linear model identified the main effects of factors 2 and 3 as significant, with p-values of 0.14% and 1%, respectively. The overall model p-value was 0.4%, indicating a good fit to the data, and the lack of fit p-value was 17%, confirming the adequacy of the linear model for describing the general trends in the data.

Interaction Model: By including interactions, the second and third factors remained significant (p-values of 0.4% and 2%), but the interactions are not significant, with p-values above conventional thresholds.

Quadratic Model: The quadratic model incorporated additional terms, including quadratic effects for factors 2 and 4 and significant interactions among several factor pairs. This model had a strikingly low p-value of 5×10^{-9}, suggesting it captured the data's variability exceptionally well. The lack of fit p-value was 96%, indicating no significant evidence of systematic deviation between the model and observed data. This suggests the quadratic model not only fits the data but does so without overfitting.

Interpretation of Lack of Fit

The lack-of-fit test yielded high p-values across all models, indicating that none of the models systematically failed to represent the data. This consistency may result from the efficient experimental design, low noise in the data, or the ability of even simpler models to capture the primary structure of the response. However, while all models are statistically acceptable, the quadratic model is preferable as it accounts for more complex relationships, including significant curvature and interactions, providing a richer and more accurate representation of the underlying response surface.

While the linear model offer valid and interpretable fit, the quadratic model provides the most comprehensive description of the data. Its ability to capture significant main effects, interactions, and curvature makes it the most appropriate choice for understanding the response and making predictions.

6. Canonical analysis

The objective of the canonical analysis is to gain insight in the geometry of the quadratic model :

$$\widehat{y} \approx 51 - 3x_2 - 2.2x_3 - 3.2x_1x_3 + 1.5x_1x_4 + 1.8x_3x_4 - 2x_2^2 - 3.4x_4^2$$

6.1 Fix point determination

The first step concists in deteremining the fix point of the function

$$\vec{x}_s = -\frac{1}{2}A^{-1}\vec{a} \quad \text{et} \quad y_s = a_o + \vec{x}_s \cdot \vec{a} + \vec{x}_s^{\,T} A \vec{x}_s$$

```
% get the coefficients of the fit
coef=mdl.Coefficients.Estimate;

% select the constant
ao=coef(1);

% select the linear terms
a= [0;coef(2);coef(3);0];

% build the curvature matrix
A= [0           0 coef(4)/2 coef(5)/2
    0           coef(7)  0  0
    coef(4)/2 0 0 coef(6)/2
    coef(5)/2 0 coef(6)/2 coef(8)];
```

```
% compute the fix point
xs= -inv(A)*a/2;

% compute the value of the model at the fix point
ys= ao + a'*xs + xs'*A * xs;
```

So the coordinates of the fix point are within the domain :

```
% Place the values within a table
T_fp=table([xs;ys], ...
    'VariableNames',{'Fix point'},...
    'RowNames',{'x_1','x_2','x_3','x_4','y_s'});
format bank
disp(T_fp)
```

	Fix point
x_1	-0.79
x_2	-0.73
x_3	-0.10
x_4	-0.21
y_s	52.07

```
%format short
```

6.2 Axes determination

The geometry of the function is determined by analysing the eigen vectors and the eigen values

```
% compute the eigen values and eigen vectors
[V,lambda]=eig(A);

% place the values in a table
T_eig=array2table([V;diag(lambda)'],...
    'VariableNames',{'X1','X2','X3','X4'},...
    'RowNames',{'x1','x2','x3','x4','lambda'});

disp(T_eig)
```

	X1	X2	X3	X4
x1	-0.31	0.00	0.64	-0.70
x2	0.00	-1.00	0.00	0.00
x3	-0.33	0.00	0.62	0.71
x4	0.89	0.00	0.45	0.02
lambda	-3.97	-2.03	-1.01	1.61

The analysis of the eigen vectors shows that

- A first eigen vector $\widetilde{X}_1$ is in the direction of x_4 with an eigen value $\lambda_1 \approx -4$
- A second eigen vector $\widetilde{X}_2$ is in the direction of x_2 with an eigen value $\lambda_2 \approx -2$
- Two eigen vectors $\widetilde{X}_3$ and $\widetilde{X}_4$ are more or less parallel to the plane x_1x_3 with an eigen value $\lambda_3 \approx -1$ and $\lambda_4 \approx 1.6$
- The canonical model is $y = 52 - 4\widetilde{X}_1^2 - 2\widetilde{X}_2^2 - \widetilde{X}_3^2 + 1.6\widetilde{X}_4^2$ that is an hyperboloïd in a 4D space
- Interaction are present only for axes x_1, x_3 and x_4 , we may then visualize the function in this 3D space $x_1x_3x_4$.
- Let's start by visualizing cuts with the function *slice*

6.3 Visualization with slice

```
% Defining the points where the function must be evaluated
[X1,X3,X4]=meshgrid([-1:.2:1]); %

% the values of X2 is fixed
X2=0.57;
```

- Compute the function at the points of the network

```
Y=ao+ a(1)*X1 + a(2)*X2 + a(3)*X3 + a(4)*X4 +...
    A(1,3)*2*X1.*X3 +A(1,4)*2*X1.*X4+A(3,4)*2*X3.*X4+...
    A(2,2)*X2.^2 + A(4,4)*X4.^2;
```

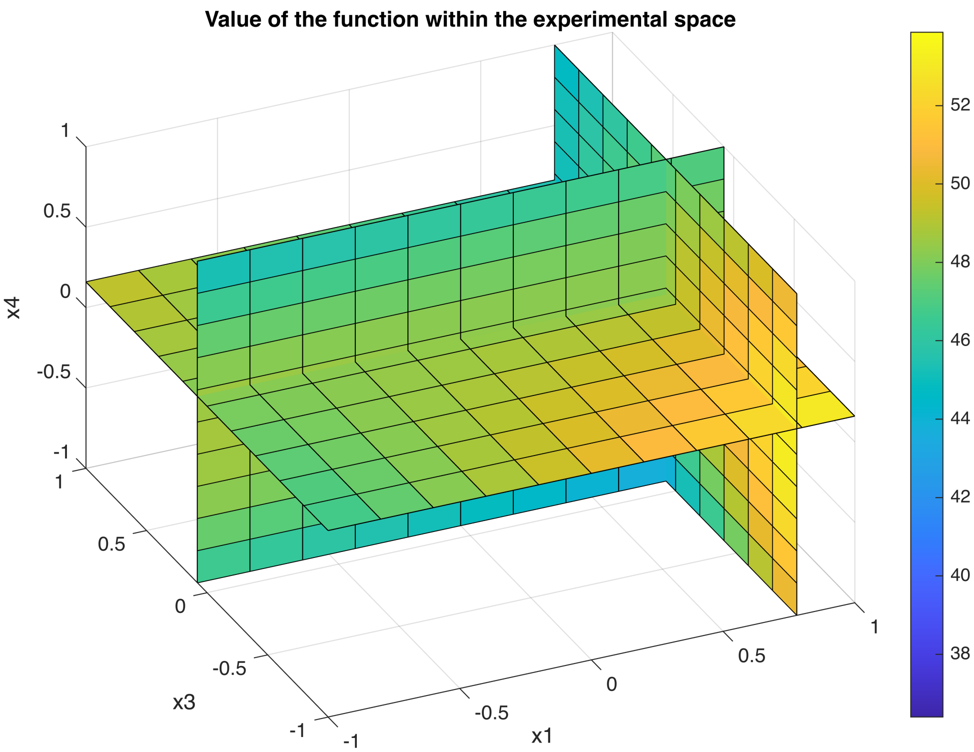
- Planes cross the fix point

```
xslice=.78;
yslice=0.08;
zslice= [.16];
```

- Figure

```
figure
slice(X1,X3,X4,Y,xslice,yslice,zslice)
title('Value of the function within the experimental space')
xlabel('x1')
ylabel('x3')
zlabel('x4')
colorbar
map=colormap;
caxis('manual')
```

```
view([-24.70 40.40])
```



The hyperbolique behaviour is easy to observe in the plane x_3x_4

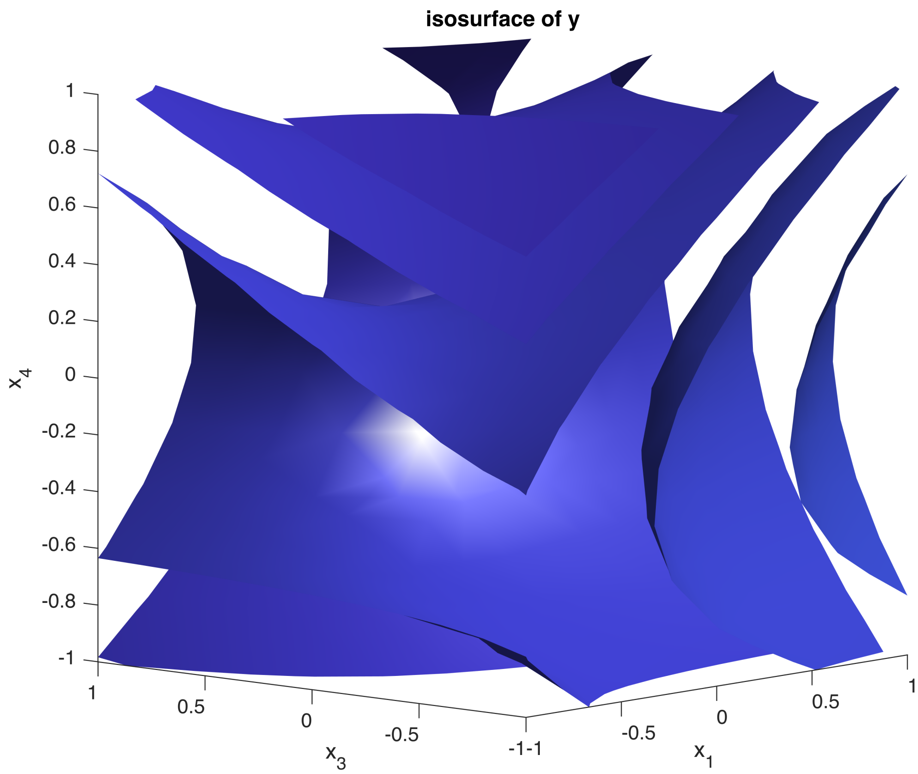
- Keep in memory the scale for the color

```
Ncouleur=size(map,1);
[Cmin,Cmax]=caxis;
```

- Draw the iso-surfaces

```
figure
valeur= [44:2:54];
for k=1:6
    p=patch(isosurface(X1,X3,X4,Y,valeur(k)));
    isonormals(X1,X3,X4,Y,p)
    p.FaceColor=map(10+k*7,1:3);
    p.EdgeColor='none';
    hold on
end
hold off
daspect([1 1 1])
view(3);
axis tight
camlight
```

```
lighting gouraud
view([-48.30 8.40])
title('isosurface of y')
xlabel('x_1')
ylabel('x_3')
zlabel('x_4')
```



- We may now observe the complex behaviour of the response which is not straitforward visible looking at the function.
- Changing the value of x_2 (the fourth factor) between -1 and $+1$ we may observe the the behaviour stays the same with maximal values at the middle of the axes $x_1 = 1$, $x_3 = -1$.
- The next step of the analysis should be with the subject matter for an scientific interpretation
- The point to keep in mind is the importance of the canonical analysis to understand and interpret the function of the second degree.