

**Riga Technical University**  
**Scientific Laboratory for Machine and Mechanism Dynamics**

**ROBUST MULTICRITERION OPTIMIZATION OF  
COMPOSITE MATERIAL ELEMENTS**  
**KEDRO User Manual v. 1.01, 2023.**

Copyright © 2013 -2023 by Jānis Auziņš and Aleksandrs Janušovskis

Riga, RTU, 2023.

## Contents

1. General information.....	3
1.1. KEDRO licensing.....	3
1.2. Installing the program .....	3
1.3. Starting the program .....	3
1.4. Designations used .....	4
1.5. Using graphs in the KEDRO program .....	4
2. A simple example .....	5
2.1. Beginning a new project .....	5
2.2. Inputting the experimental design.....	6
2.3. Inputting responses .....	11
2.4. Conducting approximations .....	15
3. Controlling adequacy of approximations .....	16
3.1. Cross-validation.....	16
3.2. Other measurements of the approximation quality .....	19
3.3. $R^2$ -evaluation.....	19
3.4. F-criterion.....	20
3.5. Determining possible outliers .....	21
3.6. Graphic checks of residuals .....	23
4. An example of two factors and 50 points. Computer experiments. MSE LH design .....	24
4.1. Inputting experimental designs from the Internet database.....	24
4.2. Inputting responses using analytical functions .....	26
4.3. Evaluation of the significance of regression members .....	29
4.4. Locally weighted polynomial approximations .....	32
4.5. Approximation with kriging.....	34
5. Optimization.....	41
5.1. Multi-criterion optimization .....	43
6. Parametric and non-parametric approximations .....	46
6.1. Polynomial approximations .....	46
6.2. Non-parametric approximations .....	47
6.3. Mathematical foundation for locally-weighted polynomial approximations .....	47
6.4. Kriging method .....	48
Literature.....	49
Appendix 1. Entering analytical expressions in the KEDRO program .....	50

## 1. General information

The KEDRO software was created in the Laboratory of Problems of Machine and Mechanism Dynamics in the Riga Technical University. The authors of the software are the engineering science doctors: professor Jānis Auziņš, professor Aleksandrs Januševskis and associate professor Jānis Januševskis. The program implements the scientific ideas of the authors in the fields of computer experiment planning, metamodel creation and global optimization.

The main characteristics of the program KEDRO v. 1.01:

1. Number of experiment input factors – up to 12
2. Number of experimental runs – up to 500
3. The number of additional optimization constraints in the form of inequalities – up to 12

The computer must have an Internet connection in order to use the optimized experimental designs from the Internet data base.

The types of optimized designs: Latin hypercube (LH) type computer experiment plans optimized according to the Minimal Average Error, MaxiMin, Entropy, V. Eglājs' potential energy criteria, classic factorial, Central composite, Box-Behnken type plans for implementing physical experiments.

The metamodel (approximated input – response connection) creation methods: first, second, third order multivariable polynomials, locally weighted polynomial approximations, kriging.

Optimization method: modified recurrent simulated annealing of the start for optimization of one criterion, the Pareto limit set generation for multicriterion optimization (if the specific version of the program contains multicriterion optimization).

Operating systems supported by KEDRO: MS Windows XP, Vista, Windows 7, 8 and 10.

### 1.1. KEDRO licensing

KEDRO may be used only by natural and legal persons that have signed licensing agreements with the collective of authors of the software, represented by Aleksandrs Januševskis. Any other use of the software or any its component will be considered to be a violation of the Copyright Law of the Republic of Latvia (The Official Newspaper of the Saeima and the Cabinet of Ministers of the Republic of Latvia, 2000, vol. 11.; 2003, vol. 8.; 2004, vol. 10; 2007, vol. 6; 2008, vol. 3; “Latvijas Vēstnesis”, 2010, vol. 205), and the violator will be sued and punished with the full severity of the law.

### 1.2. Installing the program.

The program KEDRO along with all its folders must be copied from the data medium to the hard drive of the computer in a location convenient for the user. Afterwards the file KEDRO.exe must be started. When running the program for the first time, the license number or the user password may be requested. During the first time running KEDRO, the computer must be connected to the Internet.

### 1.3. Starting the program.

The file KEDRO.exe must be started. After the start, the license information is shown on screen. Pressing any key, after 5 seconds the license information disappears from the screen and work with the program may begin. The licensing information may also appear on the screen if the KEDRO program is not closed and the user switches to the window of some other program. Then, upon returning to KEDRO, the license window may be shown again.

## 1.4. Designations used

Designations in **bold** signify matrices and column vectors.

$n$  – the total number of runs in the experiment, including repeated runs. Also known as the number of design points. May not exceed 500.

$m$  – number of input variables (factors), may not exceed 12.

$L$  – number of responses registered in the experiment measurements, may not exceed 12.

$\mathbf{x}_i$  –  $i$ -th experiment design point in an  $m$ -dimensional space. Column vector, includes  $m$  components  $x_{ij}$   
 $j=1, \dots, m$

$\mathbf{y}_j$  –  $j$ -th response function.

$y_j(\mathbf{x}_i)$  – value of the  $j$ -th response function in the  $i$ -th experiment design point.

$x_1, \dots, x_{12}$  factor designations used in the analytical expressions in the KEDRO windows.

$y_1, \dots, y_{12}$  response designations used in the analytical expressions in the KEDRO windows.

$y_i$  – registered value of the response in the  $i$ -th experiment design point. Is used in cases where one specific response is selected from the twelve possible.

$\bar{y}$  – mean value of the response in the experiment points

$\hat{y}(\mathbf{x})$  – approximated response function

$\hat{y}(\mathbf{x}_i)$  – value of the approximated response function in the  $i$ -th experiment design point

$\hat{y}(\mathbf{x}_i) - y_i$  residual, difference between the predicted and measured response value in the  $i$ -th experimental design point.

STD – standard deviation, square root from the sum of the squares of differences divided by the number of points minus the number of degrees of freedom (see lower in the text)

## 1.5. Using graphs in the KEDRO program

All two-dimensional graphs may be zoomed. To zoom the fragment of interest, move the cursor with a pressed-down left mouse button from the left upper corner to the right lower corner. To return to the initial size, move the cursor in the opposite direction. With the right mouse button pressed, the entire graph may be moved to the right, to the left, up or down.

All three-dimensional graphs may be rotated with the left mouse button pressed. In some graphs, pressing the right mouse button may open the graph formatting window.

## 2. A simple example

Work with KEDRO usually consists of the following steps:

1. Creating the experimental design. Exporting the plan as a file for use in computer experiments.
2. Conducting physical or computer experiments.
3. Inputting the values of responses registered in the experiments.
4. Approximation of the dependences of the response – input factors. Choosing the best approximation, evaluating adequacy, filtrating gross mistakes. Evaluating the significance of the regression function members.
5. Output of the approximated polynomial models in files for use in other software
6. Implementing global optimization, using analytical expressions in criteria and boundary functions that contain both input variables and the approximated response functions.
7. Evaluating the precision of the located optimum, conducting additional experimental measurements.

Next we will review these steps in a simple example.

### 2.1. Beginning a new project

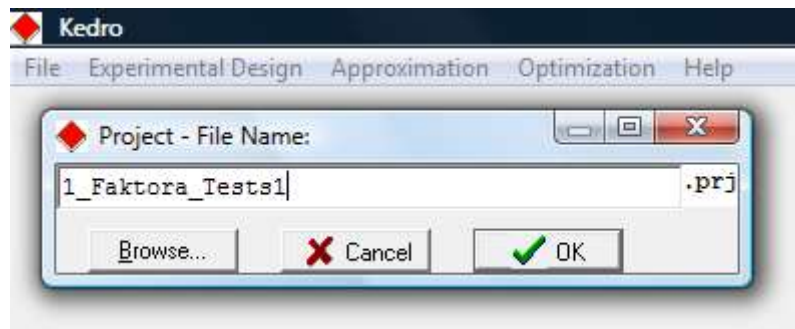


Fig.1. Inputting the project name

KEDRO project files are text files with the extension .prj. They contain information about the project: the experimental design, measured or calculated responses in the experiment points, as well as analytically entered response formulas, approximation parameters (if approximation has been conducted), comments (if they have been entered in the window Input Responses), optimization parameters (if optimization has been conducted). The program proposes a default name, adding a number to the previously saved file name. It may be replaced by a completely new name.

## 2.2. Inputting the experimental design

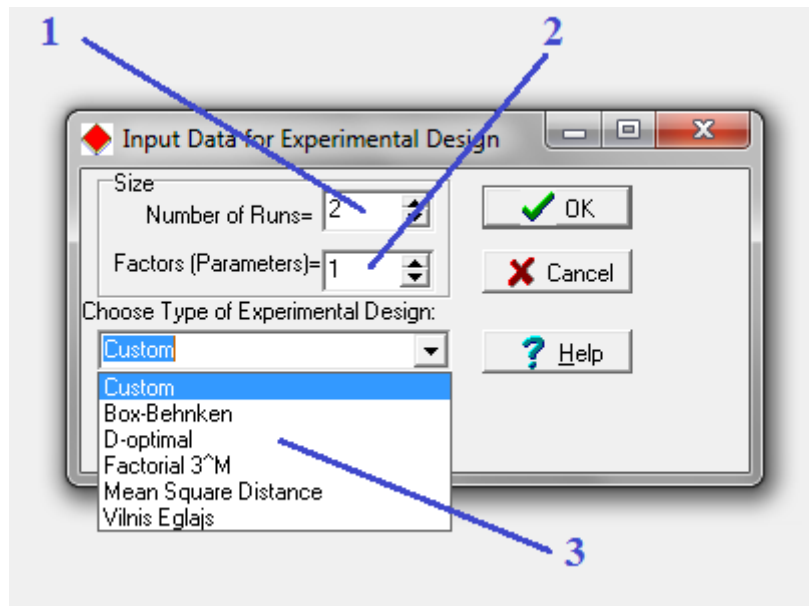


Fig. 2. Selection of the experimental design type

When entering a new experimental design, the following must be selected: 1) number of experimental runs  $n$ ; 2) number of factors  $m$ . 3) if the KEDRO package includes software for generating new designs (it is not included in the standard KEDRO package), the type of the generated design may be selected (Box-Behnken type designs for approximating physical experiment results with quadratic polynomials, D-optimal Latin hypercube (LG) and continuous D-optimal designs (mainly used for the approximation of the results of physical experiments), Mean Square Distance (also known as Minimal Squared Error, MSE designs) LH and continuous designs, LH designs optimized according to the V. Eglājs potential energy criterion (used mainly for conducting numerical experiments, less efficient than the MSE plans). The KEDRO standard package does not include software for generating new plans, however, it includes a large number of optimized designs both in the folder KEDRO\Plans, and in the Internet data base. It must be noted that the generation of new designs, if it is available, requires significant use of processor time, therefore in this menu it is recommended to choose the type Custom.

The number of factors in the standard KEDRO package may not exceed 12 and the number of runs may not exceed 500.

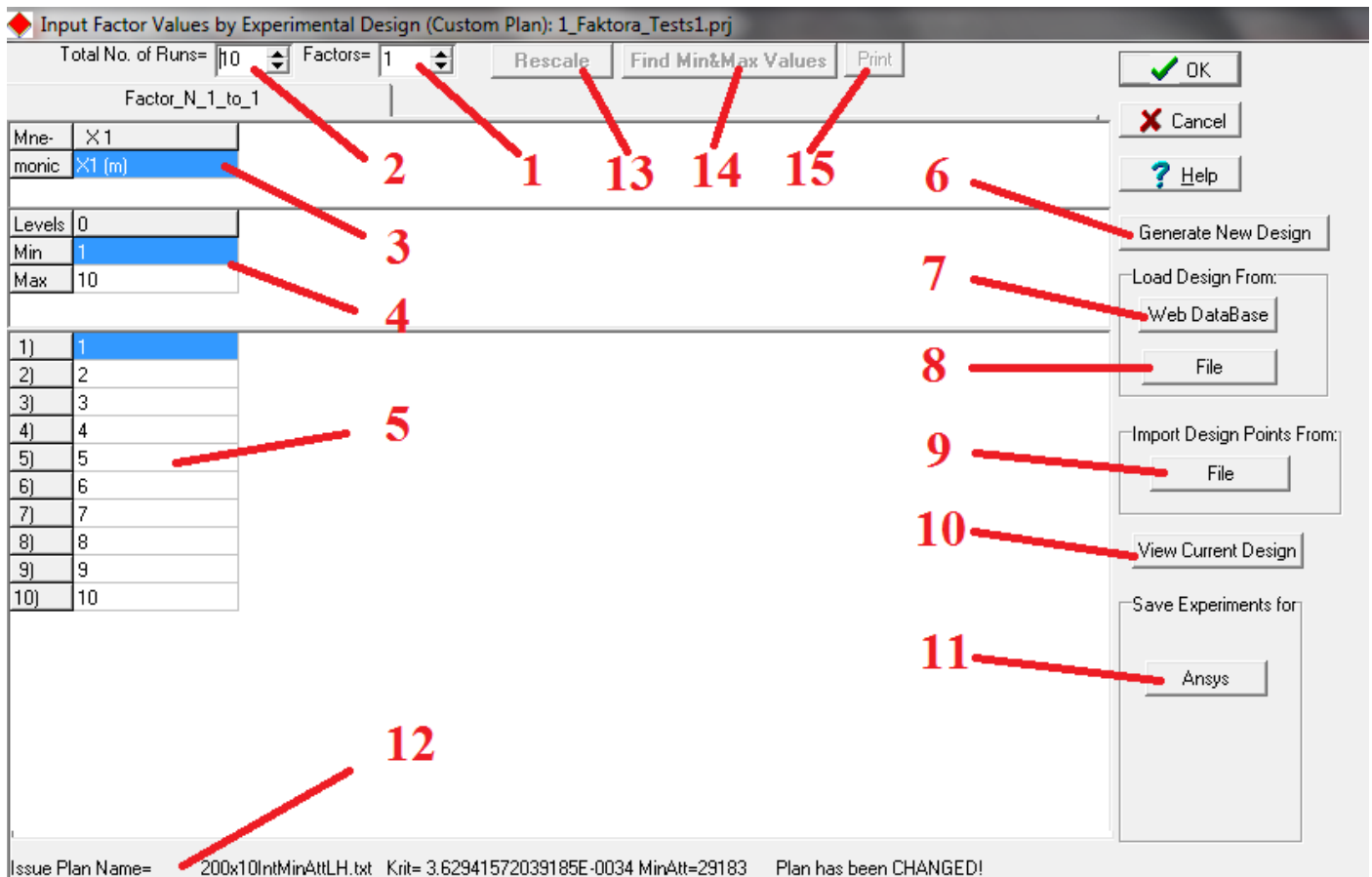


Fig. 3. The design input window

In the design input window the user may define: **1)**  $m$  – the number of factors ( $1 \leq m \leq 12$ ).

**2)**  $n$  – the number of experiment runs ( $2 \leq n \leq 500$ ). When defining  $n$ , the following recommendations should be noted.

The minimum number of experiment runs (points)  $n$  depends on the number of factors  $m$  and the anticipated degree of the approximation polynomial.

Aproksimācija	Min punktu skaits $N_{min}$	$m=1$	$m=2$	...	$m=12$
Lineāra	$m+1$	2	3		13
Kvadrātiska	$(m+1) * (m+2)/2$	3	6		91
Kubiska	$(m+1) * (m+2) * (m+3)/6$	4	10		455

In order to obtain high quality approximations, the recommended number of necessary points usually is:  $N_{rek1} > 1.5 N_{min}$  – for linear,  $N_{rek2} > 1.6 N_{min}$  – for quadratic and  $N_{rek3} > 2.0 N_{min}$  – for cubic approximations. It must be noted that the cubic approximations may not be used with Box-Behnken or factorial  $3^m$  designs. When changing  $n$  or  $m$ , the layout of the input tables dynamically changes.

**3)** The mnemonic names of the factors may not contain more than 15 symbols. The data of the table fields colored in gray may not be modified and are provided only for reference.

**4)** Minimum and maximum values of factors. Must be corrected after the manual input or reading of the design.

**5)** Table of design factor values. May be input and corrected manually, however, usually the design is obtained automatically using the methods described below. May be input and corrected manually. The Copy-Paste operation in this version may be executed only for some of the cells rather than for the entire table or its part. This applies to all KEDRO tables. (In the 1.97 Beta version full Copy-Paste is implemented in all KEDRO tables, however it is currently at the beta-testing stage.)

**6)** The button "Generate New Design" is not accessible if the software does not have an additional design calculation program. It is not included in the standard KEDRO package.



7) Reading the design from the Internet data base. The computer must possess a working Internet connection. This data base contains an enormous number of optimized experiment designs. If the data base does not include a design with a specific number of points, for example, 111, a design with a slightly lower or higher number of points (runs) should be chosen.

8) A button for reading the design from a design file. The design file is a text .txt file formatted in a special way – its first row contains information about the plan (number of points, number of factors, number of levels, comments). The remaining rows contain the design itself: the first column contains the number of the run, the others contain the values of the design factors. The design files with the corresponding names can be found in the folder KEDRO\Plans.

The names of the design files identify the type and size of the design.

The design file name begins with a number that indicates the number of experiments  $n$ , followed by  $x$ , and then the following number that identifies the number of factors  $m$ . **Int** indicates that the design contains integers. If **Int** is not included in the name, then the design contains real (floating-point) numbers that are scaled in segments  $[-1.0, 1.0]$  or  $[0, 1.0]$ . The abbreviation **BB** indicates that it is a Box-Behnken design, abbreviation **Dopt** - a D-optimal design, abbreviation **Egl** – Eglājs design, abbreviation **Fact3<sup>M</sup>** – three level factorial design. The abbreviation **LH** indicates a Latin hypercube design that is intended for the approximation of determined functions. Examples of the design file name **N x M (Int) [Dopt Egl BB Fact3<sup>M</sup>] (LH).txt**:

**5x2IntBB.txt** is a Box-Behnken design that contains 5 runs for 2 factors whose integer values are arranged in 3 levels in the range from -1 to 1;

**10x2IntDoptLH.txt** is a D-optimal Latin hypercube design that contains 10 runs for 2 factors whose integer values are arranged in 10 levels in the range from 1 to 10 so that the determinant of the corresponding information matrices (assuming that the approximated function is quadratic) would be maximally large;

**16x3EglLH.txt** is an Eglājs Latin hypercube design that contains 16 runs for 3 factors whose real (floating-point) values are arranged in 16 levels in the range from -1.0 to 1.0 so that the points would be maximally uniformly distributed in the factor space according to the Eglājs criterion, namely, that they would give the maximum amount of information on the approximated function.

**9x2IntFact3<sup>M</sup>.txt** is a factorial design  $3^m$  that contains 9 runs for 2 factors whose integer values are arranged in 3 levels for each factor in the range from -1 to 1;

After reading the design the table sizes and the assigned variable values are dynamically set.

If the design type previously selected is not "**Custom**" (for example, "**Box-Behnken**", "**D- optimal**", "**Factorial 3<sup>M</sup>**" or "**Vilnis Eglājs**"), then the program proposes to load an existing design of the specific type. If the specific design does not exist, it may be generated anew if the software package includes design calculation programs, or the design may be found in the Web data base proposed. It must be noted that the generation of "**Vilnis Eglājs**" and "**D- optimal**" designs for large values of  $m$  and  $n$  may consume significant amounts of processing time and the user must decide how many iterations should be assigned for improving the design criterion and when the process should be interrupted. In the other cases ("**Box-Behnken**", "**Factorial 3<sup>M</sup>**") the design may also be loaded if the specific design does not exist, since it will be quickly calculated before loading. When selecting "**Factorial 3<sup>M</sup>**" designs,  $m$  must be less than 6, since in the opposite case the constraint for  $n < 501$  will be violated.

For files whose names include „int" the factor values are level numbers, in other design files the factor values may be scaled in the segment  $[-1, 1]$  or  $[0, 1]$ . The user does not have to worry about this after reading the design since in any case the design must be scaled to the physical range for each factor (see. below paragraph 13 regarding design scaling).

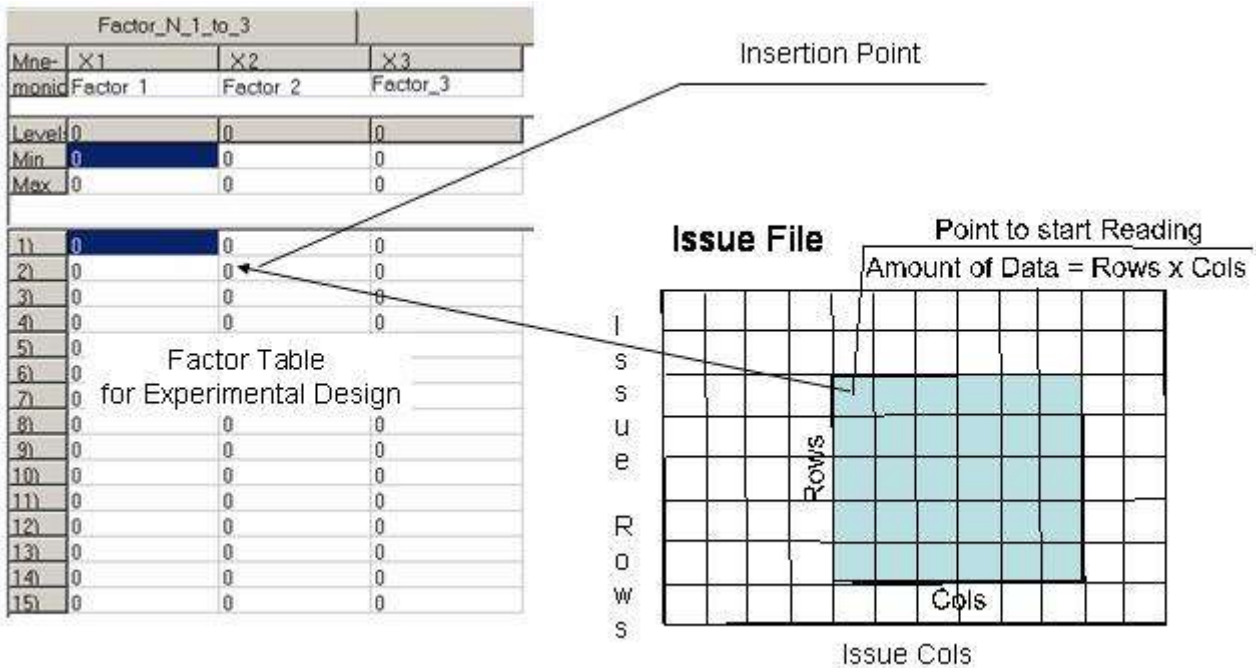
9) Inputting the design from one or more text files. They must be .txt files where the numbers are separated by space, tab or comma symbols. KEDRO does not support reading designs from MS Word or Excel files. The data reading method from the selected text file is defined in a special **Insert Data from Issue File (...) to Factor Table** dialog window. This window lets the user define the amount of data read (row number x column number). (**Amount of Data** = Rows x Cols) from the selected source – text file that must contain factor values arranged in columns.



If  $m$  factor values for  $n$  runs are read at once, then the source file must contain  $m$  columns with  $n$  rows, that is, it must be a rectangular matrix with the size  $(n \times m)$ . The amount of input data must correspond to the size of the source file.

In order to correctly (unambiguously) read the data, it is necessary to define the absolute address of the factor table elements (**Insertion Point**), indicating the row and column where the first read value should be inserted. Similarly the array element of the source file (**Point from what to start Reading**) must be indicated from which the information reading will be started.

The insertion of data in cases when the source file size is greater than the amount of readable data, may be illustrated with the following diagram.



10) To visually evaluate the actual plan loaded in the factor table, viewing its projections, the user must press the button "**View Current Design**".

11) Exporting the entered plan in various CAD, CAE formats. The current version does not support all formats, therefore some buttons are not accessible.

12) Information on the source of the previous or current experimental design (if it has been read).

13) By pressing the button „**Rescale**”, the design is obtained in the ranges of the real physical variables. The experimental designs are stored both in integer format and in floating-point format (scaled in the range from -1.0 to 1.0) (design space). Therefore, when the chosen design has been read, in order to transfer to the actual physical values of the factors, the minimal and maximal values of the range of actual changes of the factors must be entered in pos. 14 of the table (second from above) in the current window. During the entering of these values, the table of the experimental design values becomes inaccessible, however, the button "**Rescale**" is available. To obtain the specific physical values of the design – the so-called working space, the design must be recalculated (linearly transformed) by pressing the button "**Rescale**".

Input Factor Values by Experimental Design (Custom Plan): 1\_Faktora\_Tests1.prj

Total No. of Runs= 10 Factors= 1 Recalculate Find Min&Max Values

Factor\_N\_1\_to\_1

Mne-	X1
monic	X1
Levels	0
Min	0
Max	0

1)	1
2)	2
3)	3
4)	4
5)	5
6)	6
7)	7
8)	8
9)	9
10)	10

Fig. 4. Manually entered experimental design values and the factor name „X1”

14) After manual design input, if the values have been entered in physical units, the button „**Find Min&Max Values**” should be pressed.

Factor\_N\_1\_to\_1

Mne-	X1
monic	X1 (m)
Levels	0
Min	1
Max	10

1)	1
2)	2
3)	3
4)	4
5)	5
6)	6
7)	7
8)	8
9)	9
10)	10

Fig. 5. The design table after manual input and determining the range values Min and Max

15) Pressing the button „**Print**”, the entered design may be printed out.

## 2.3. Inputting responses

After inputting or correcting the design, the button OK or Cancel should be pressed (Cancel, if the changes were not successful).

After entering or changing the design the approximation button in the menu Approximation is not available since the experiment responses must be entered or corrected..

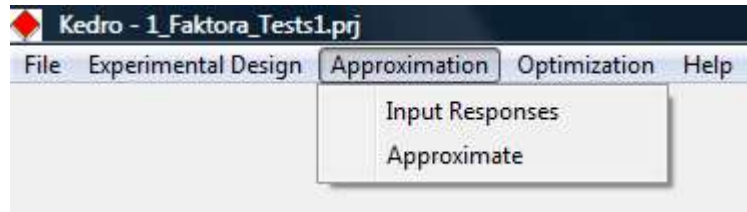


Fig. 6. The menu Approximation after correcting or changing the experimental design

The experiment run responses may be entered manually (not recommended since the user is almost certain to make mistakes, enter the data in the wrong order, enter an incorrect digit, etc.), by reading it from a text file or by entering in the form of a formula (a very useful option for understanding the properties of various designs and approximation methods).

Fig. 7 shows a filled-out response entry window:

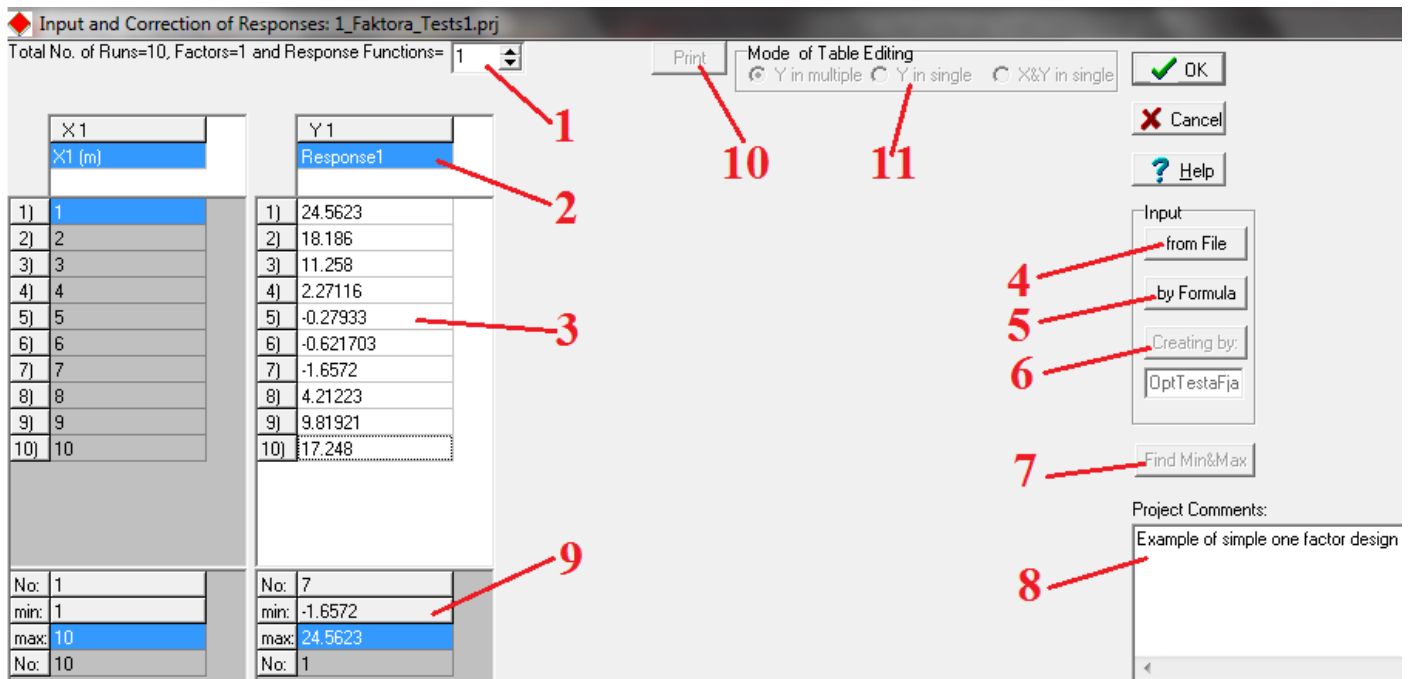


Fig. 7. Response entry window

1) Number of response functions  $L$ . When correcting  $L$ , the layout of the entry table changes dynamically. In the KEDRO version 1.01 the number of responses may not exceed 12. If the number of responses exceeds 6, then the window shows two pages with response tables – the user may switch between them to visualize them one after another.

2) Mnemonic names of the responses. Here the name of the first response is shown - Response1. **In formulas, the designations  $y_1, y_2, \dots, y_{12}$  should be used instead of the mnemonic names.**

- 3) A filled-out response table. It may be obtained, for example, by entering the corresponding values in each table cell from the keyboard or copying them from the clipboard. When manually filling in or editing the table, the user may transfer to the next cell by using the mouse or the keyboard arrow buttons.
- 4) By pressing the "**Input**" field button "from File" (this button is available only if the screen shows the input tables for response functions, that is, if the corresponding page is activated), the values by columns may be entered from a previously prepared data file in text format of arbitrary size. The rules for the file formatting are the same as with reading the design from a text file.
- 5) By pressing the "**Input**" field button "by Formula", the response values may be entered, using analytical expressions (formulas).
- 6) By pressing the "**Input**" field button "**Creating by**", the values may be input, using special programs.
- 7) The button "**Find Min&Max**" for finding the minimal and maximal values of the responses. It is intended only for control during response input, further the program will find these values automatically even if the button has not been pressed.
- 8) The field "**Project Comments**" is intended for the input of text comments that are saved in the project file.
- 9) The found minimal and maximal values of the responses after pressing the button "**Find Min&Max**" (pos.7).
- 10) The button "**Print**" allows to print out the entered response values.
- 11) To ensure comfortable data input, the mode of input should be selected depending on the specific number of factors and response functions. The data editing tables provide 3 modes: a) the factors and responses are arranged in tables in several pages (each page contains up to 6 variables); b) a single table is used for the input of response values; c) a single page is used for the input of factors and response values. The KEDRO version 1.01 implements only mode a.

For the sake of demonstration, let us read the responses from the text file Plana\_Un\_Attilzu\_teksta\_Fails.txt  
The file contains the following information:

Simple example of text file for input of experimental design or responses

Any comment, for example the date, time of experiment executing

```
1 1.0 24.5623 0.4
2 2.0 18.1860 0.5
3 3.0 11.258 0.6
4 4.0 2.27116 0.7
5 5.0 -0.279330 0.8
6 6.0 -0.621703 0.9
7 7.0 -1.65720 1.0
8 8.0 4.21223 1.1
9 9.0 9.81921 1.2
10 10.0 17.248 1.3
```

Any comment or other text

Last row of the text file

Let us read the value of the first response from this file, pressing the button **Input from File**:

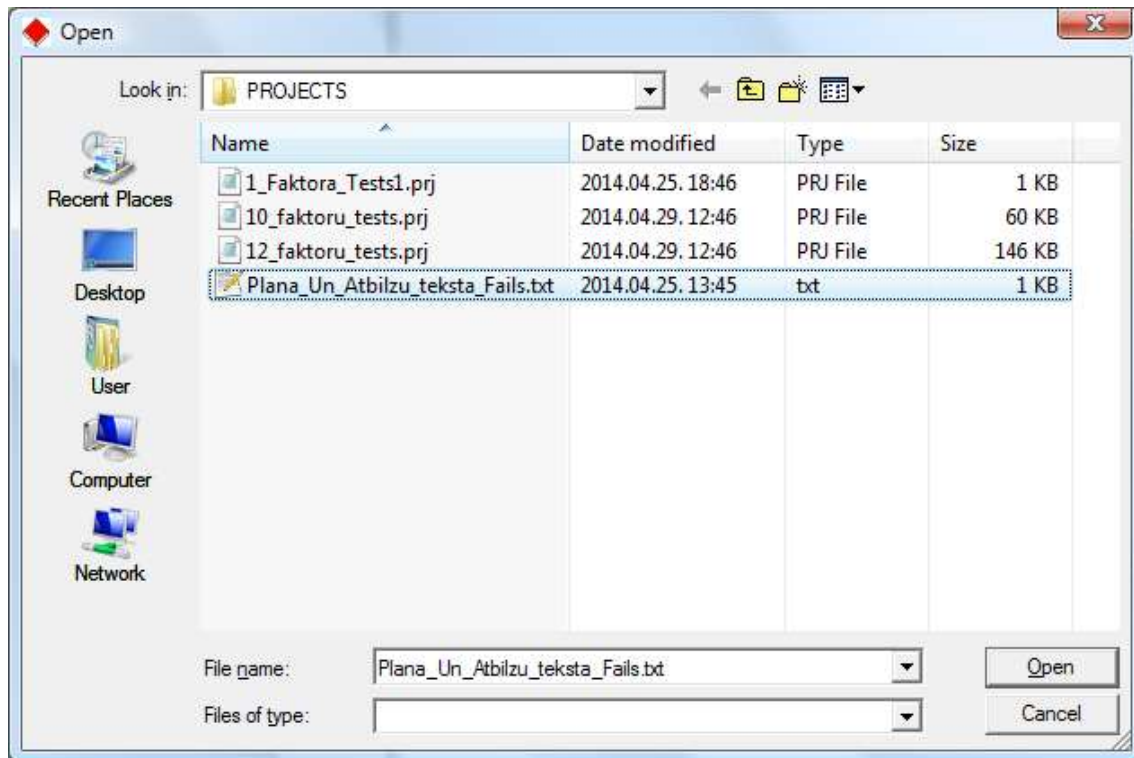


Fig. 8. Selection of the file to be read

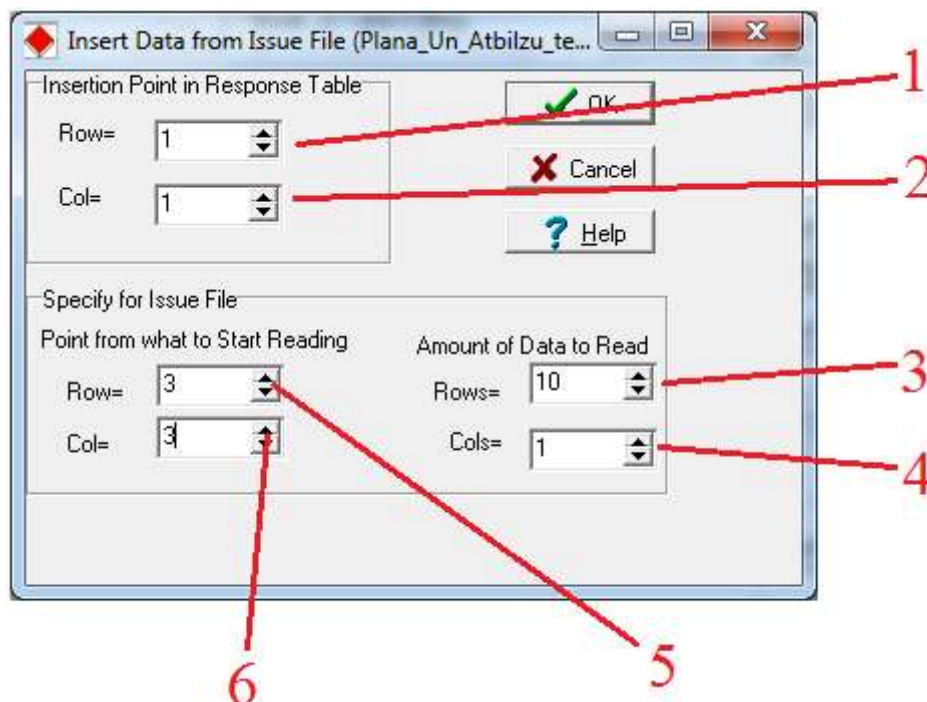


Fig. 9. Response reading window

To read responses from a file, the following must be entered:

1 – the row of the response table from which the input should be started. Usually it is the first row, but this is not compulsory.

2 – the response with which the entry should be begun. Different responses may be entered from different files.

3 – how many response rows should be entered. By default it is proposed to fill out the entire table.

4 – how many responses should be read from the given file. By default, it is proposed to read all the responses.

**Important:** If the number of planned responses L is greater than 1 (for example, 5), but it is not necessary to read all the responses, then the number of columns to be read must always be corrected – by default it will be L and a file input error may occur.

5 – from which file row the reading should be started. The first rows of the file may contain other information.

6 – from which file column the reading should be started. If it is not the first column then its left side should also contain numbers separated by spaces, tab signs or commas. After the last column to be read, the file may contain text information (separated by a space) which the program does not read.

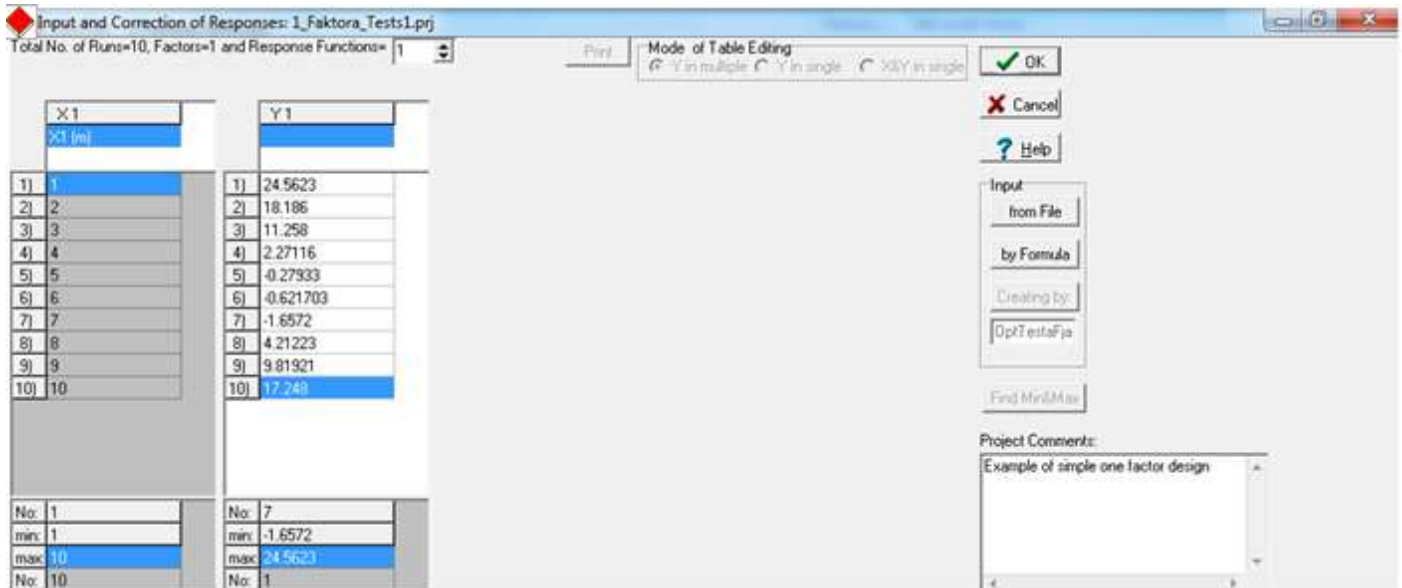


Fig. 10. Response entry window after reading from the file Plana\_Un\_Atbilzu\_teksta\_Fails.txt

**Important! KEDRO uses only the dot as a decimal separator! If the comma is set as the decimal separator in Windows, then using the Control Panel – Regional and Language Options window, button „Customize this format...”, the dot should be set as the decimal separator:**



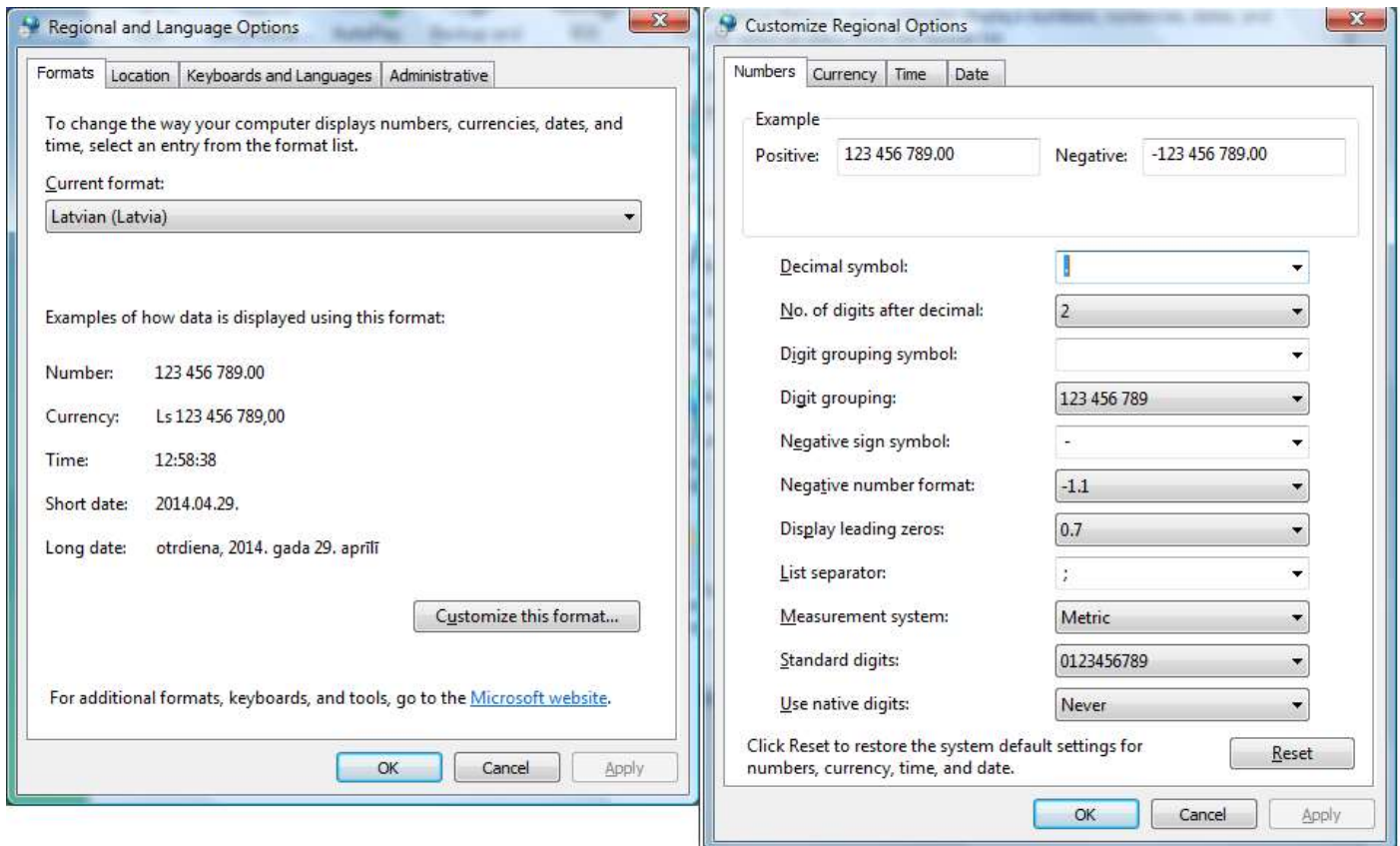


Fig. 11. Setting the dot as the decimal separator

## 2.4 Conducting approximations

After entering the responses in the main menu, the menu Approximation-**Approximate** becomes available:

Approximation of Response Surface: 1\_Faktora\_Tests1.prj

Runs Total=10

Approx Order:  Custom  Kriging  Third  Second  First-linear

Local Approx

Buttons: Calculate Point, View Coefficients, Options, Approximate, View Graphs

Functions Yi:	
Sigma Cross	11.008744
Sigma Cross%	119.930989%
R2 adjusted	0.007614
F-Crit 99%	1.069<11.259???
Sigma	9.144219
Sigma%	99.618568
MeanExpValue	8.499867
StDev of Exp	9.179232
Exp. Range	26.219500
MaxError	13.432293
Bad Point No.	10
Max Rel Error	200.00%
BadRelPointNo.	5
Max Cook Dist.	20.521559
Suspicious point	10
No.ofActualExp	10
Filtered STD	

Fig. 12. Part of the approximation window



Figure 12 shows the results of linear polynomial approximation for the entered data. For approximation, global polynomial (linear, quadratic, cubic), locally weighted polynomial or kriging methods may be chosen. The essence of the methods is explained below.

**Attention!** Sometimes after making changes in the input data, when opening the approximation window the result of the previous approximation is shown with the Approximate button inaccessible (see figure). In that case a different degree of approximation should be chosen, the approximation should not be conducted, the necessary approximation should be chosen again and then the button Approximate should be pressed. This is not an error, it is intended for cases when the approximation requires a long time.

### 3. Controlling adequacy of approximations

The left side of the table (see Fig.12) shows data that characterize the adequacy of the approximation.

#### 3.1. Cross-validation

One of the main indicators of the adequacy and quality of approximation is the cross-validation error Sigma Cross. KEDRO uses leave-one-out-cross-validation or the so-called  $n$ -fold cross-validation error:

$$\sigma_{cross} = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_{i(-i)})^2}{n}} \quad (1)$$

where  $\hat{y}_{i(-i)}$  is the value of the approximated function for input factor value  $x_i$ , if the approximation does not use the  $i$ -th experiment point. Successively discarding one of the experimental points, the program calculates the coefficients of the approximated model and the mean square difference between the approximation without the discarded point and the response measurements in the discarded point. The cross-validation error is an evaluation of the so-called model prediction error. The prediction error is essentially the most important indicator of model quality, but theoretically it should be obtained by conducting a large number of additional runs and calculating the difference between the approximated model and the response values in points that have not been used for creating the model. Cross-validation is an approximate evaluation of the prediction error and it is obtained without conducting additional runs. By itself this indicator is not very informative since the size of the cross-validation error must be evaluated against the numerical value of the response. The relative error could be calculated in relation to the mean value of the response or the response range  $y_{max}-y_{min}$ , however such relative error measurements possess certain drawbacks. The mean value could be close to zero – then the relative error will be an unreasonably large number. The response range may turn out to be very large although the majority of the experimental points give an almost constant response value – then the error evaluation will be unreasonably reduced. This situation is shown in Figure 13. In relation to the mean value of the response (about 1.0) the error of approximation is insignificant. In relation to the response change range the error is very large.

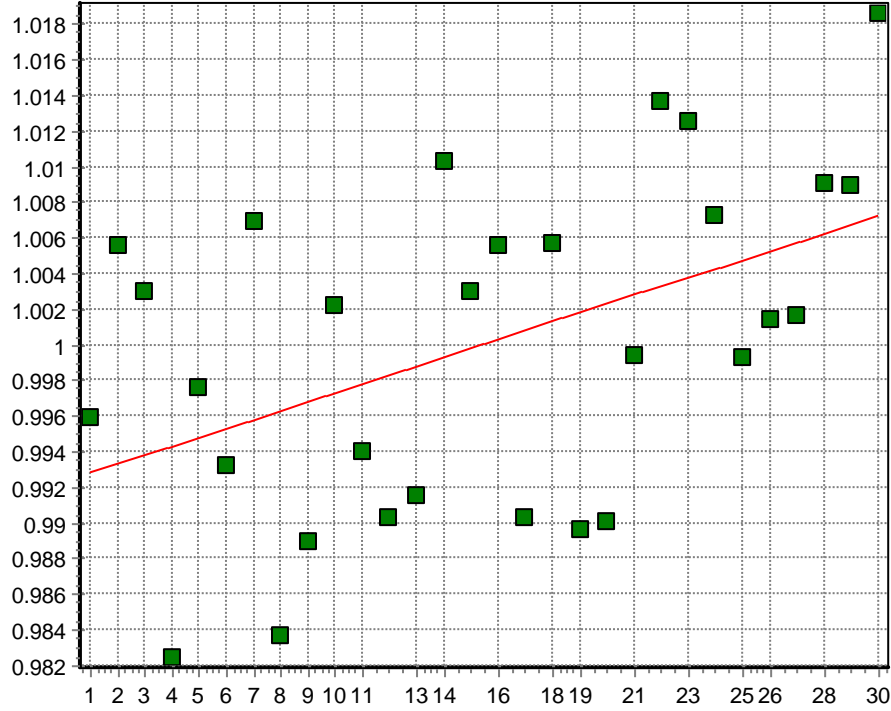


Fig. 13. Linear approximation for an almost constant function

A more objective evaluation of the relative cross-validation error is the division of the absolute error with the mean quadratic deviation of the response from the mean value  $STD$  (designated in the table as  $STDev$  of  $Exp$ )

$$STD = \sqrt{\frac{\sum_{i=1}^n (y_i - \bar{y})^2}{n-1}} \quad (2)$$

where  $\bar{y}$  is the mean value of the response in the experimental points

$$\bar{y} = \frac{\sum_{i=1}^n y_i}{n} \quad (3)$$

Relative cross-validation error (Sigma Cross%) as a percentage of  $STD$ :

$$\sigma_{cross \ rel} = \frac{\sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_{i(-i)})^2}{n}}}{STD} \times 100\% \quad (4)$$

This prediction error evaluation has the following good property: if the response is approximated with a constant, then the relative cross-validation error is 100%. Therefore, if the cross-validation error approaches 100 percent, then the approximation is useless. If the error exceeds 100%, then the approximation is absolutely wrong and the result is worse than the assumption that the response does not depend on the values of the input factors!

But, knowing this value and the values shown in the table -  $STD$ ,  $\bar{y}$  (designation  $MeanExpValue$ ) and range (designation  $Exp. Range$ ), the relative error may also be calculated in relation to the mean value and the range of the response change.

The evaluation of the cross-validation error is informative for the approximation quality of both physical and numerical (computer experiment) results. As can be seen, the cross-validation error in the example is greater than 100%, signifying that the linear approximation of these results is not acceptable.

### 3.2. Other measurements of the approximation quality

The other numbers shown in the table are classic evaluations of natural experiment result analysis that are based on the residuals analysis (difference between the measured and approximated response values) and are used mainly for the polynomial approximations of physical experiments. For computer experiments and non-parametric approximations (kriging and locally weighted polynomials) they are informative but cannot be interpreted with absolute precision.

#### Mean Squared Error

$$MSE = \hat{\sigma}^2 = \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n-p} \quad (5)$$

where  $p$  is the number of parameters in the regression function.

Correspondingly the square root from the mean square error in regression analysis is known as the **Sigma** evaluation:

$$RMSE = \hat{\sigma} = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n-p}} \quad (6)$$

In the table it is designated as Sigma. Just as the cross-validation error, the relative Sigma value is calculated in relation to the standard deviation from the mean response value:

$$\hat{\sigma}_{rel} = 100\% \frac{\hat{\sigma}}{STD} \quad (7)$$

This value is designated in the table as Sigma%

Here it should be noted that large values of residuals always indicate a bad approximation or the presence of gross errors in the input data. However, small residual Sigma values and Sigma% are no guarantee for a precise approximation.

### 3.3. $R^2$ -evaluation

In regression analysis, the so-called  $R^2$  evaluation and its more objective variant– adjusted  $R^2$  are popular:

$$R^2 = \frac{SSR}{SST} = 1 - \frac{\hat{\sigma}^2(n-p)}{STD^2(n-1)} \quad (8)$$

and

$$R_{adj}^2 = 1 - \frac{n-1}{n-p} (1 - R^2) = 1 - \frac{SSE/(n-p)}{SST/(n-1)} \quad (9)$$

Here  $SSE$  is the Sum of squares due to error with  $n-p$  degrees of freedom:

$$SSE = \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (10)$$

$$MSE = \frac{SSE}{n-p} = \hat{\sigma}^2 = \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n-p} \quad (11)$$

SSR is the Sum of squares due to regression:

$$SSR = \sum_{i=1}^n (\hat{y}_i - \bar{y}_i)^2 \quad (12)$$

with  $p-1$  degrees of freedom

SST is the Total sum of squares:

$$SST = \sum_{i=1}^n (y_i - \bar{y}_i)^2 \quad (13)$$

with  $n-1$  degrees of freedom

$$STD = \sqrt{\frac{SST}{n-1}} \quad (14)$$

It is obvious that

$$SST = SSR + SSE \quad (15)$$

Approximation is considered to be good if the adjusted  $R^2$  criterion is close to 1. In regression analysis this is often interpreted so: the value of the adjusted  $R^2$  criterion 0.99 explains 99% from the change of the response during the experiment runs. This is not always completely true, however, small  $R^2$  and adjusted  $R^2$  values always signify a bad approximation.

### 3.4. F-criterion

The  $F$ -criterion for experiments without repeated runs (designation  $F$ -Crit 99%). The classical adequacy criterion used in regression analysis is

$$F = \frac{SSR/(p-1)}{SSE/(n-p)} \quad (16)$$

In order to believe with a given confidence probability (usually 0.95 or 0.99) that the created approximation model is better than the assumption that the response of the physical experiments does not depend on the input factors, the value of the  $F$ -criterion must be greater than the value of the inverse cumulative  $F$ -distribution with  $p-1$  degrees of freedom for the numerator and  $n-p$  degrees of freedom for the denominator. KEDRO calculates the critical value for the confidence probability 0.99

In inverse  $F$ -distribution tables, Excel, MathCad programs as well as using Internet online calculators (for example, <http://www.danielsoper.com/statcalc3/calc.aspx?id=4>) the critical values for different confidence probabilities may be obtained. Usually in these tables the value of confidence level is used that is equal to 1 minus the confidence probability, therefore KEDRO uses confidence level 0.01.

For computer experiments and non-parametric approximations the  $F$ -criterion is not an absolutely safe evaluation of adequacy but it has informative character and attention should be paid to it. For example, figure 12 shows a case where the linear approximation of results does not give an adequate model: the value of relative cross-validation is greater than 100% and the value of the  $F$ -criterion is greater than the critical value for the confidence level 0.01

However, this can also be clearly seen in the graph (button **View Graphs**):

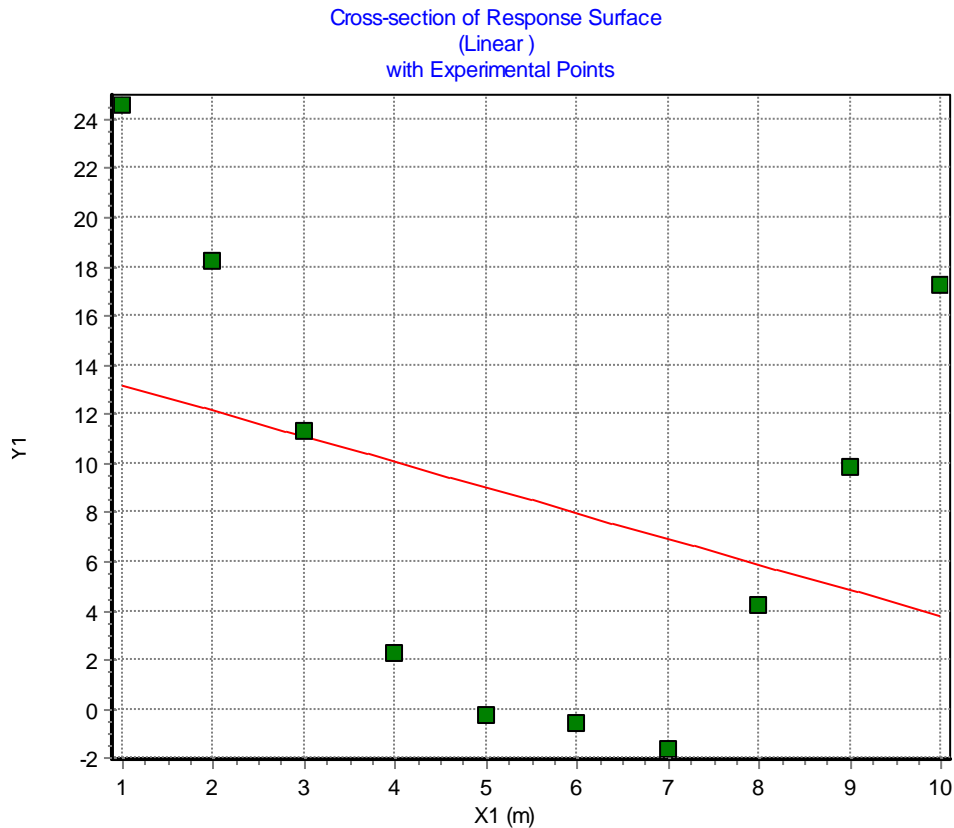


Fig. 14. A graph of an inadequate linear approximation

### 3.5. Determining possible outliers

The table data Max Error, Max Rel Error and Cook's distance and the numbers of the corresponding experimental points may allow to detect outliers in the input data. Max Error and Max Rel Error signify the maximum difference between the experimentally determined and approximated function values. The designations Bad point and Suspicious point do not guarantee that the corresponding run has a gross error. They signify candidates for additional checks. Cook's distance is the maximum difference between an approximation where the experimental point is used and an approximation where it is not used. Such a point is also a candidate for a check. One of the safest types of checks is the elimination of the suspicious point. If thereby the relative cross-validation error is significantly reduced, it signifies that either this point is in error (mistakenly entered, a mistake in measuring the response, etc.) or the program is incapable of creating an adequate model in the vicinity of this point.

Figure 15 shows the linear approximation evaluation table for the previously described example with comments

Sigma Cross	11.008744	Cross-validation error
Sigma Cross%	119.930989%	Relative cross-validation error. Exceeds 100%, therefore the approximation is useless.
R2 adjusted	0.007614	Adjusted $R^2$ criterion. Very small, therefore the approximation is bad
F-Crit 99%	1.069<11.259???	99% confidence probability F-criterion. Less than critical value
Sigma	9.144219	Adjusted evaluation of the mean approximation error.
Sigma%	99.618568	Mean relative approximation error. Very large.
MeanExpValue	8.499867	Mean response value.
StDev of Exp	9.179232	Standard deviation of the response from the mean value
Exp. Range	26.219500	Range of response change $y_{\max}-y_{\min}$
MaxError	13.432293	Maximal difference between the approximation and the measured response
Bad Point No.	10	Number of Point with maximal Absolute value of residual
Max Rel Error	200.00%	Maximal relative error (in relation to the experimental value)
BadRelPointNo.	5	Point with maximal relative error
Max Cook Dist.	20.521559	Maximal Cook's distance
Suspicious point	10	Number of point with maximal Cook distance. First candidate for checking
No.ofActualExp	10	Number of used points = $n$ - number of discarded points.
Filtered STD		The mean error of approximation for the discarded points (if such exist)

Fig. 15. Evaluation of the linear approximation quality

Let us implement the quadratic and cubic approximation of the same data.

Functions Yi:	Y1	Functions Yi:	Y1	Functions Yi:	Y1
Sigma Cross	11.008744	Sigma Cross	2.204427	Sigma Cross	2.852404
Sigma Cross%	119.930989%	Sigma Cross%	24.015375%	Sigma Cross%	31.074542%
R2 adjusted	0.007614	R2 adjusted	0.969764	R2 adjusted	0.972397
F-Crit 99%	1.069<11.259???	F-Crit 99%	145.329>9.547	F-Crit 99%	106.683>9.780
Sigma	9.144219	Sigma	1.596131	Sigma	1.525058
Sigma%	99.618568	Sigma%	17.388501	Sigma%	16.614227
MeanExpValue	8.499867	MeanExpValue	8.499867	MeanExpValue	8.499867
StDev of Exp	9.179232	StDev of Exp	9.179232	StDev of Exp	9.179232
Exp. Range	26.219500	Exp. Range	26.219500	Exp. Range	26.219500
MaxError	13.432293	MaxError	2.376763	MaxError	-1.786378
Bad Point No.	10	Bad Point No.	3	Bad Point No.	4
Max Rel Error	200.00%	Max Rel Error	200.00%	Max Rel Error	200.00%
BadRelPointNo.	5	BadRelPointNo.	5	BadRelPointNo.	5
Max Cook Dist.	20.521559	Max Cook Dist.	-5.100265	Max Cook Dist.	-5.983240
Suspicious point	10	Suspicious point	1	Suspicious point	1
No.ofActualExp	10	No.ofActualExp	10	No.ofActualExp	10
Filtered STD		Filtered STD		Filtered STD	

Fig. 16. Linear, quadratic and cubic approximation for the example results

As can be seen in Figure 16, quadratic and cubic approximations are of a significantly higher quality than linear approximation. Cubic approximation has a slightly better  $R^2$  and sigma criteria values, however, the cross-validation error clearly shows that quadratic approximation is the best.

Figure 17 shows the graph of the quadratic approximation



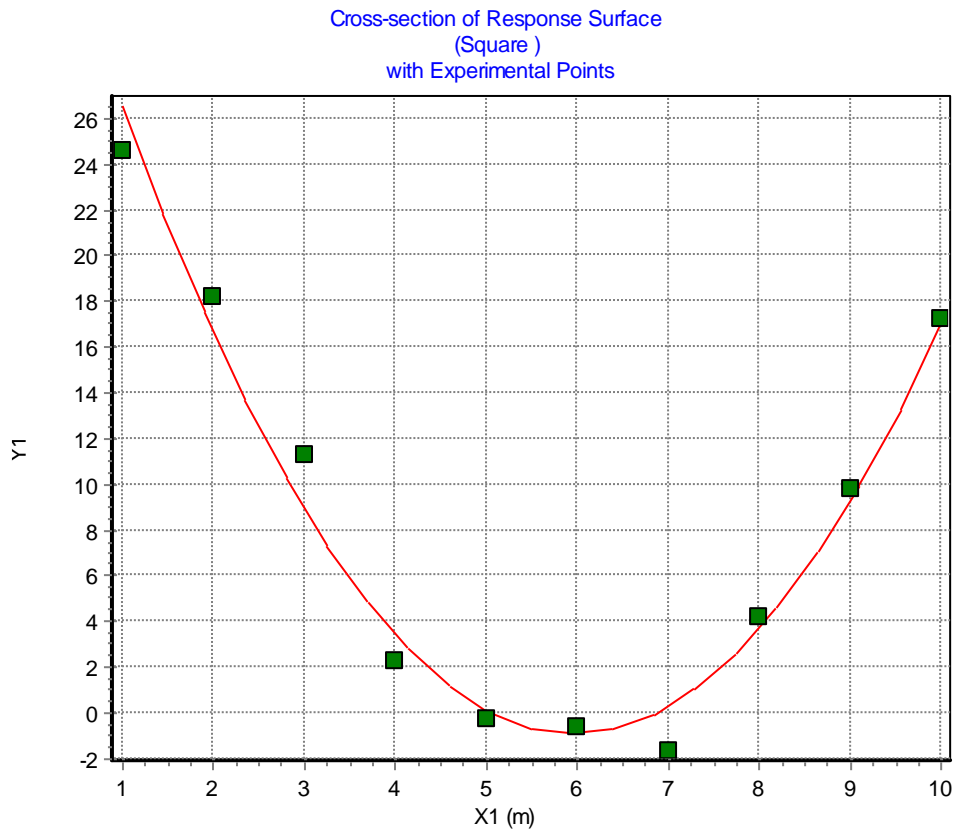


Fig. 17. Graph of the quadratic approximation

### 3.6. Graphic checks of residuals.

The button **Residuals** in the approximation window allows to visually analyze residuals (the difference between the measured and the approximated responses). One of the most important residual analysis windows is Histogram, Fig. 18.

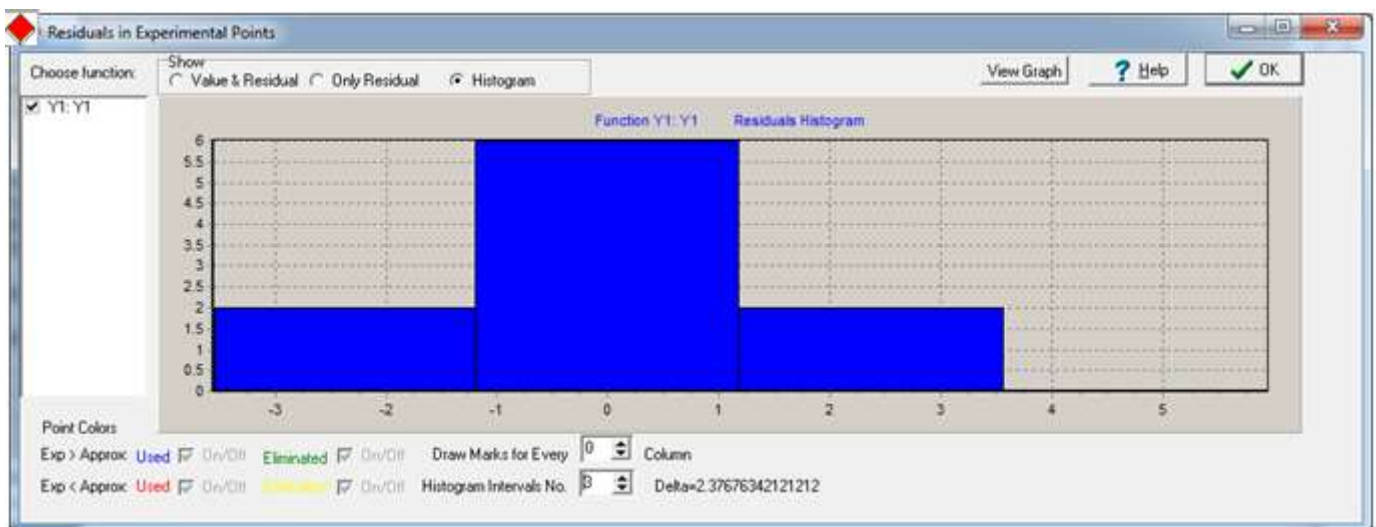


Fig. 18. The histogram window for residuals

It is particularly significant for the analysis of results of physical (natural) experiments. In that case the residuals should be approximately distributed according to the normal Gauss probability density distribution, the histogram columns should form a bell-shape, see Fig. 19.

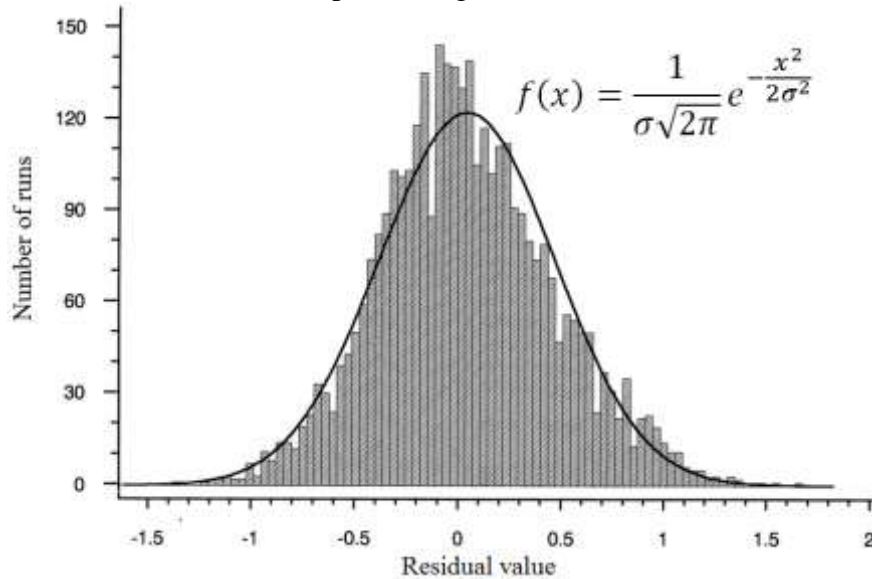


Fig. 19. A histogram that corresponds to the normal distribution.

The histogram will never correspond to the normal distribution precisely. The correspondence should be approximate. In the case of a small number of experimental points, the histogram should also have a small number of columns selected, for example, 3 columns for 10 runs, as in Fig. 18.

#### 4. An example of two factors and 50 points. Computer experiments. MSE LH design

##### 4.1. Inputting experimental designs from the Internet database

Pressing the button (pos.7) **Load Design from Web Database** in the window **Input Factor Values by Experimental Design** (see Fig. 3) we obtain:

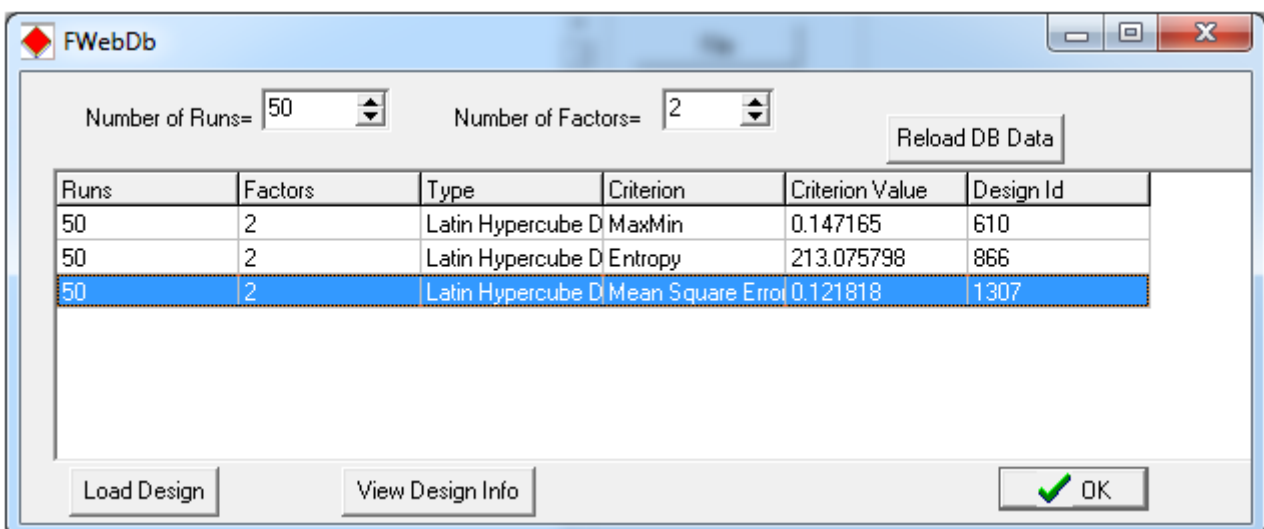


Fig. 20. Window for the database of experimental designs

Let us choose Latin Hypercube Mean Square Error and **Load Design**

After reading the design variables must be assigned names and the design must be scaled according to the minimum and maximum values of the physical variables. For example, let us scale both variables to the interval  $[-1, 1]$

Total No. of Runs= 50		Factors= 2		Recalculate	
Factor_N_1_to_2					
Mne-	X 1	X 2			
monic	X1	X2			
Levels	0	0			
Min	-1	-1			
Max	1	1			
1)	45	5			
2)	36	38			
3)	48	11			
4)	6	48			
5)	20	4			
6)	2	37			
7)	14	8			
8)	13	43			
9)	23	34			
10)	7	41			
11)	26	17			
12)	39	1			
13)	15	36			
14)	8	28			
15)	3	13			
16)	37	45			
17)	50	19			
18)	21	42			

Fig. 21. The design table prior to scaling

Pressing the button **Rescale** we obtain the design in the range of the real physical variables.

Using the button **View Current Design** we can view the design and using the button **Copy to Clipboard** and 2D projection we can copy the image of the design as a bitmap or metafile. The copied image can be copied from the clipboard to another program, for example, Microsoft Word. It is recommended to use the metafile format for copying since it provides a higher quality format of vector graphics.

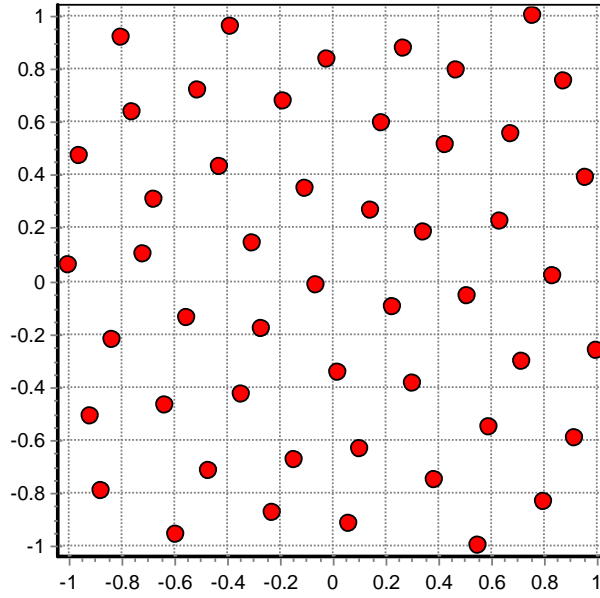


Fig. 22. The design read from the database after scaling

Using the button Save Experiments for ANSYS the design can be saved in a text file that can be read from the FEM programs ANSYS

## 4.2. Inputting responses using analytical functions

In this example we will enter an analytical function instead of real measurements. KEDRO allows entering various analytical expressions, using the designations  $x_1, x_2, \dots, x_{12}$  for variables. In the optimization window, the responses  $y_1, y_2, \dots, y_{12}$  can also be used as variables. It is compulsory to write  $x_1$  rather than the mnemonic name of the variable.

A list of KEDRO functions is given in Appendix 1. In formulas lower and upper case letters are considered to be equal.

KEDRO contains several functions intended specifically for testing. One of them is the „Standard test function“:

$$f_{test}(x) = \frac{1}{1 + \sum_{j=1}^m (x_j - 0.05)^2} - \frac{1}{1 + \sum_{j=1}^m (x_j + 0.05)^2} \quad (17)$$

This multi-argument function is difficult to approximate with polynomials, it has exactly one minimum and one maximum in the region  $[-1, 1]^m$ . The approximation prediction precision of an analytically entered function may be tested in randomly generated points - their number can be set in the window **Options for Approximations** (Fig. 31.).

Relative error in the test points TestSigma is calculated according to the formula:

$$\sigma_{test} = 100\% \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N (f_{test}(w_i) - \hat{f}_{test}(w_i))^2}}{\sqrt{\frac{1}{N} \sum_{i=1}^N (f_{test}(w_i) - \bar{f}_{test})^2}} = 100\% \sqrt{\frac{MeanSquareError}{Variance}} \quad (18)$$

where  $w_i$  – test points ( $i = 1, \dots, n$ ),  $\hat{f}_{test}(w_i)$  - the approximated test function value in the test point,  $\bar{f}_{test}$  - the mean test function value in the test points. This evaluation can be used for all analytically entered responses and may be compared to the predicted prediction evaluation Sigma Cross% , expressions (2, 3, 4).



Fig. 23. Entering an analytical function in the response input window

Noise with normal distribution and mean value zero may be added to the entered responses, where standard deviation is set as a percentage to the standard deviation of the calculated data from the mean value. Let us input noise level zero. Figure 24 shows the appearance of the Standard test function

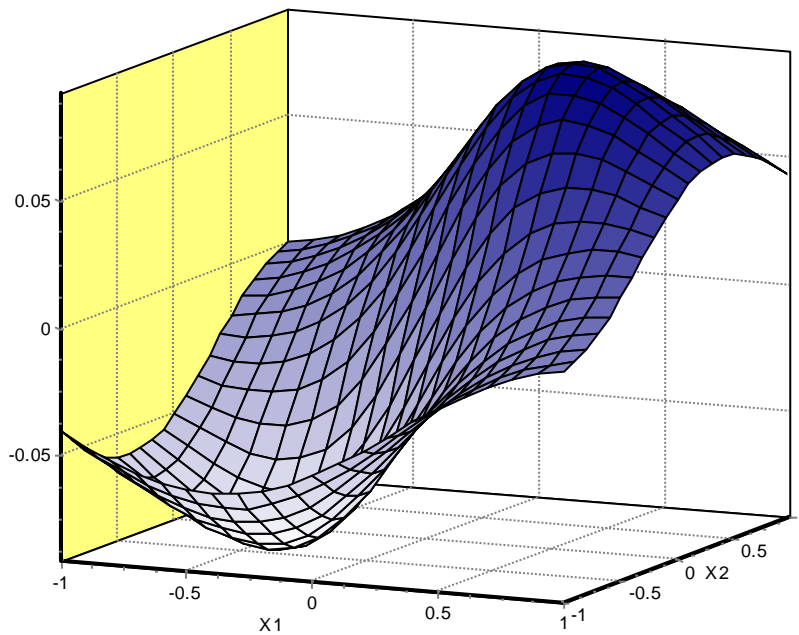


Fig. 24. Standard two argument test function

The comments window allows entering any text.  
Let us test the global polynomial approximations:

Runs		Approx Order	
Total=50	<input type="radio"/> Custom <input type="radio"/> Kriging	Total=50	<input type="radio"/> Custom <input type="radio"/> Kriging
<input type="button" value="Edit Project File"/>	<input type="radio"/> Third	<input type="button" value="Edit Project File"/>	<input checked="" type="radio"/> Second
	<input checked="" type="radio"/> First-linear		<input type="radio"/> First-linear

Functions Yi:	
Sigma Cross	0.024337
Sigma Cross%	43.872783%
R2 adjusted	0.817604
F-Crit 99%	110.823>5.087
Sigma	0.023691
Sigma%	42.707872
MeanExpValue	-0.000223
StDev of Exp	0.055472
Exp. Range	0.180635
MaxError	-0.051210
Bad Point No.	47
Max Rel Error	200.00%
BadRelPointNo.	13
Max Cook Dist.	-0.057515
Suspicious point	47
No.ofActualExp	50
Filtered STD	

Functions Yi:	
Sigma Cross	0.026488
Sigma Cross%	47.750237%
R2 adjusted	0.805823
F-Crit 99%	41.669>3.465
Sigma	0.024444
Sigma%	44.065500
MeanExpValue	-0.000223
StDev of Exp	0.055472
Exp. Range	0.180635
MaxError	-0.047448
Bad Point No.	33
Max Rel Error	200.00%
BadRelPointNo.	13
Max Cook Dist.	-0.067351
Suspicious point	47
No.ofActualExp	50
Filtered STD	

Functions Yi:	
Sigma Cross	0.010096
Sigma Cross%	18.200239%
R2 adjusted	0.975685
F-Crit 99%	219.465>2.888
Sigma	0.008650
Sigma%	15.593404
MeanExpValue	-0.000223
StDev of Exp	0.055472
Exp. Range	0.180635
MaxError	0.019737
Bad Point No.	37
Max Rel Error	200.00%
BadRelPointNo.	13
Max Cook Dist.	0.034743
Suspicious point	47
No.ofActualExp	50
Filtered STD	

Fig. 25. First, second and third order polynomial approximation tables

The best cross-validation error is given by the third order approximation. Let us verify this graphically.

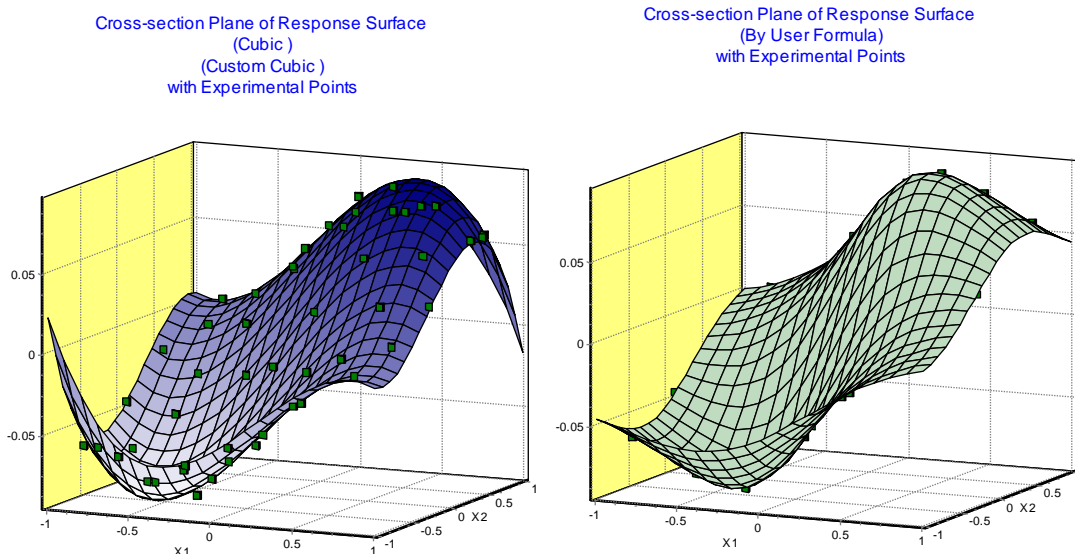


Fig. 26. The approximated function and the precise function

As can be seen, in the corner points the functions are significantly different – this is characteristic for higher order polynomial approximations. The analysis of residuals does not show a significant difference from normal distribution and does not indicate the existence of a systematic error. The actual approximation error, calculated in 10 000 test points, is approximately 0.16%, which is a rather good indicator. However, this error may be calculated only for a known response function. The adjusted  $R^2$  criterion shows that the approximation explains about 97% of the change of the response. The F-criterion with at least 99% percent probability indicated that the approximation is adequate (that is, the response is dependent on the inputs).

### 4.3. Evaluation of the significance of regression members

Pressing the button **View Coefficients** in the window **Approximation of Response Surface** it is possible to verify the polynomial approximation coefficients, evaluate the significance of the regression function members and to save in text files the coefficient values and the polynomial expressions for use in other programs.

In the window **Approximation Coefficients** – Normal, the table shows the polynomial coefficients  $\beta$ , if non-normalized physical values are used as variables.

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^L \beta_i g_i(\mathbf{x}) \quad (19)$$

Functions Y[1.. 1]:	
1) Free term	-0.000588263
2) X1: X1	0.128685
3) X2: X2	0.123848
4) X1 X1	0.00111244
5) X1 X2	0.00179867
6) X2 X2	0.00148215
7) X1 X1 X1	-0.0779118
8) X1 X1 X2	-0.0607328
9) X1 X2 X2	-0.0627106
10) X2 X2 X2	-0.0707222

Fig. 27. Two factor cubic approximation coefficients  $\beta$

These coefficients cannot be compared to each other by value, since the norm for each regression function member  $g_i(\mathbf{x})$  may be significantly different:

$$\|g_i\| = \sqrt{\frac{1}{V(\mathbf{D})} \int_{\mathbf{D}} g_i(\mathbf{x})^2 d\mathbf{x}} \quad (20)$$

where  $V(\mathbf{D})$  signifies the volume of the experimental region  $\mathbf{D}$ .

Inputting the normalized regression function members:

$$g_i^*(\mathbf{x}) = \frac{g_i(\mathbf{x})}{\|g_i\|} \quad (21)$$

we obtain

$$\hat{y} = \sum_{i=1}^L \beta_i^* g_i^*(\mathbf{x}) \quad (22)$$

The normalized coefficients  $\beta_i^* = \|g_i\| \cdot \beta_i$  may be compared to each other. A larger normalized coefficient indicates a larger contribution of the corresponding member to the regression function. This comparison is not absolutely precise since the regression function members are not orthogonal, that is, the equality

$$\int_{\mathbf{D}} g_i(\mathbf{x}) \cdot g_j(\mathbf{x}) d\mathbf{x} = 0, \text{ if } i \neq j \quad (23)$$



does not hold. Therefore the regression members are aliased and they may partially compensate one another. This means that by discarding one of the regressions function members the values of the other, non-discarded members will be obtained using the least squares method. However, despite this the information on the significance of members obtained by normalization may be used to reasonably evaluate the significance of the members. In the case of linear approximation this evaluation will be as good as the evaluation of the traditional regression analysis.

The values of normalized members are shown in the table:

View	
<input type="radio"/> Normal <input checked="" type="radio"/> Normalized	
Functions Y[1.. 1]:	
1) Free term is ignored	-0.000588263
2) X 1: X1	31.4864
3) X 2: X2	30.3548
4) X 1 X 1	0.21101
5) X 1 X 2	0.254772
6) X 2 X 2	0.281721
7) X 1 X 1 X 1	-12.4939
8) X 1 X 1 X 2	-6.66429
9) X 1 X 2 X 2	-6.88543
10) X 2 X 2 X 2	-11.3677

Fig. 28. Values of normalized coefficients

The significance of the corresponding members may also be evaluated using the so-called Pareto graphs: Normalized – Significance

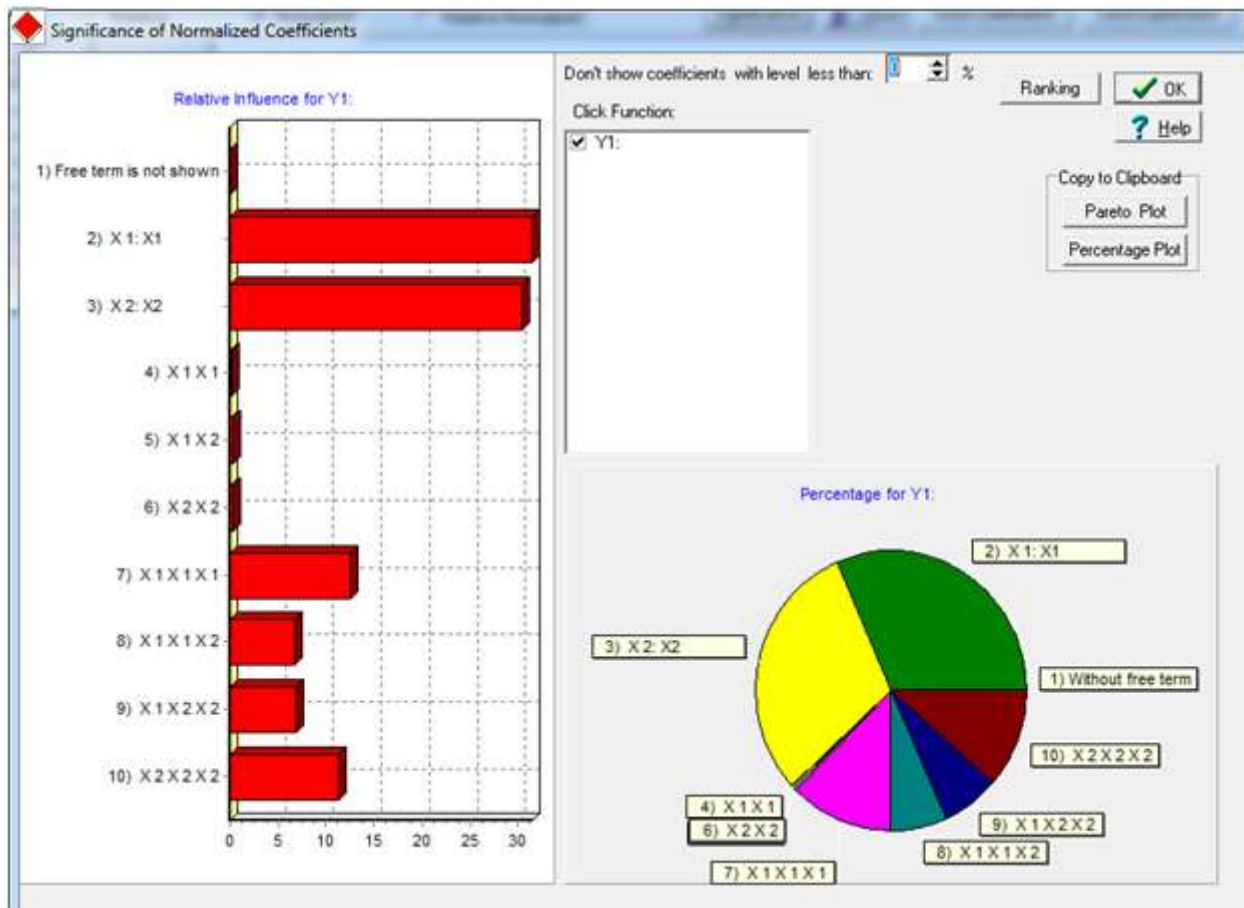


Fig. 29. Evaluation of the significance of polynomial regression coefficients

In Pareto graphs the significance of coefficients is normalized in such a way that the sum of the significances is 100%. As can be seen, in the example the linear and the cubic members are the most significant.

The option Normalized allows to mutually compare the contribution of each member to the mathematic model. For this purpose, the mean quadratic value of each regression function member is calculated and the members are normalized so that

$$\hat{y} = \sum_{i=1}^L \beta_i g_i(x) \quad (24)$$

Pressing the button Save Expressions the approximation polynomial expressions may be saved in a text file.

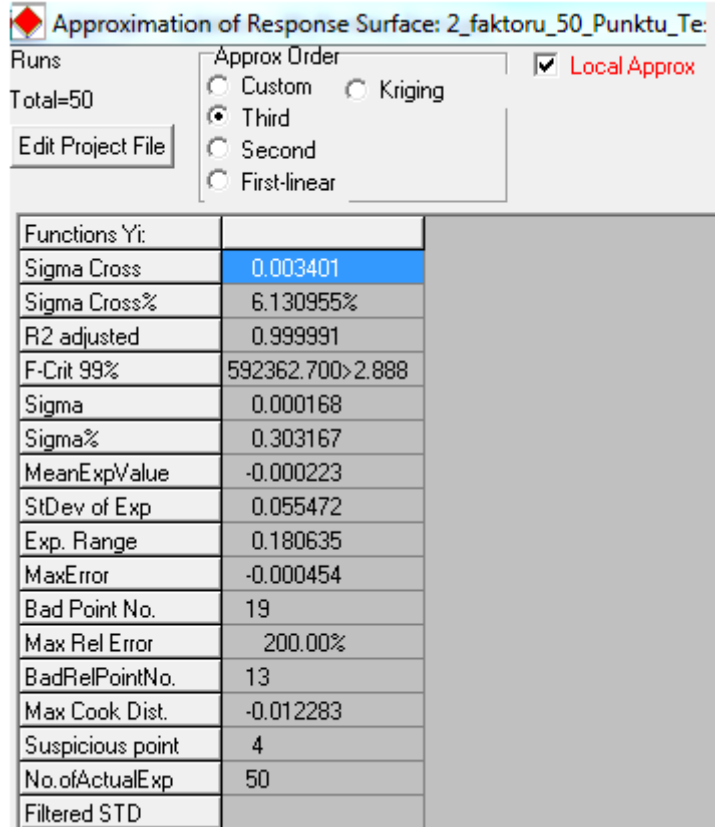
**An important note:**

The Approximation coefficient and Significance windows apply only to global parametric approximations, the information given in them cannot be applied to kriging and it can only partially be applied to locally weighted approximations!

#### 4.4. Locally weighted polynomial approximations

The theory on various approximations is given below, p. 5.

Let us implement a **locally weighted** 3<sup>rd</sup> order polynomial approximation



Functions Yi:	
Sigma Cross	0.003401
Sigma Cross%	6.130955%
R2 adjusted	0.999991
F-Crit 99%	592362.700 > 2.888
Sigma	0.000168
Sigma%	0.303167
MeanExpValue	-0.000223
StDev of Exp	0.055472
Exp. Range	0.180635
MaxError	-0.000454
Bad Point No.	19
Max Rel Error	200.00%
BadRelPointNo.	13
Max Cook Dist.	-0.012283
Suspicious point	4
No.ofActualExp	50
Filtered STD	

Fig. 30. Locally weighted cubic approximation, default bandwidth parameters.

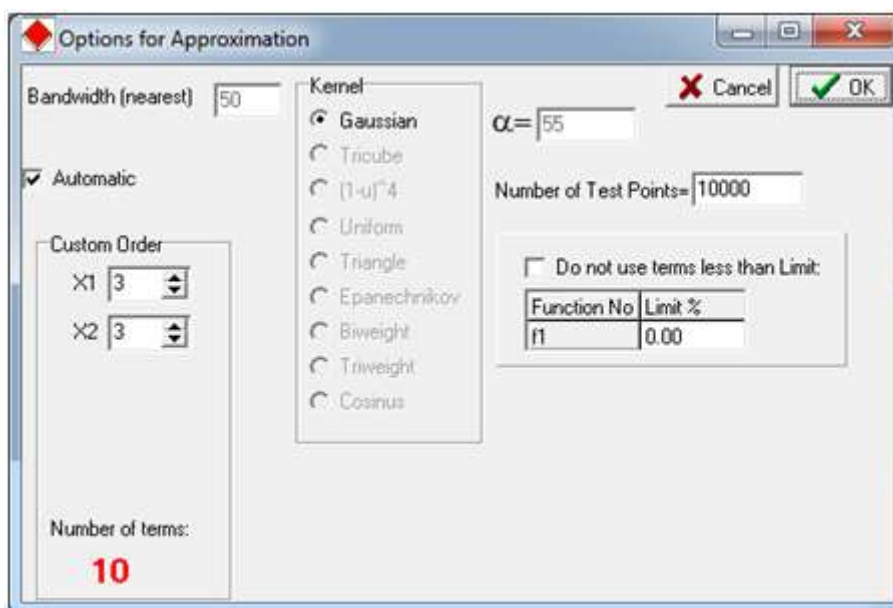


Fig. 31. The locally weighted approximation parameter window with default values

*Number of Test Points* applies only to analytically entered functions for which the approximation error may be precisely determined.

For locally weighted approximations it is possible to change the approximation parameters in the window **Options for Approximations**. Unchecking **Automatic**, it is possible to change the bandwidth and to choose other weight kernels. The result must be checked by comparing the cross-validation errors of approximations. It is recommended to use the Gauss kernel, the bandwidth equal to the number of design points and to vary the coefficient Alpha. In practice, it means to regulate the smoothness of approximation. The zero value of the Alpha coefficient signifies global polynomial approximation – all design points will have the same weight coefficient. By increasing the Alpha value, the values of the approximation function in the design points approach the registered response values. Unfortunately this does not mean that the approximation becomes more precise. It becomes less smooth and may start to adapt to noise contained in the response function. The only possibility to evaluate the precision of the approximation prediction is the cross-validation criterion. It must be noted that in the case of local approximations the cross-validation criterion value is pessimistic, that is, it is usually worse than the true evaluation.

By changing the value of the coefficient Alpha in the window **Options for Approximation** to 27 (Fig.32a), the cross-validation error is improved to the value 5.3%, but the actual error is approximately 0.021%, see Fig. 32.

**Note. In KEDRO v. 1.01 the parameters of locally weighted approximations are the same for all responses**

Functions Yi:	
Sigma Cross	0.002942
Sigma Cross%	5.303754%
R2 adjusted	0.999629
F-Crit 99%	14667.930>2.888
Sigma	0.001069
Sigma%	1.926312
MeanExpValue	-0.000223
StDev of Exp	0.055472
Exp. Range	0.180635
MaxError	-0.003202
Bad Point No.	33
Max Rel Error	200.00%
BadRelPointNo.	13
Max Cook Dist.	-0.009329
Suspicious point	4
No.ofActualExp	50
Filtered STD	

Fig. 32. Table showing the adequacy of the approximation (optimized locally-weighted approximation parameters)

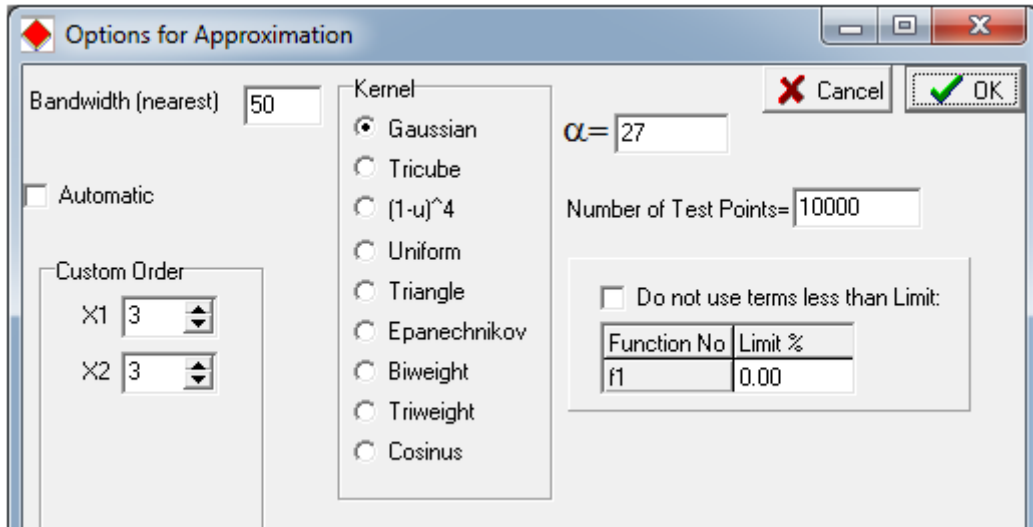


Fig.32a. Locally-weighted approximation parameters set in the window **Options for Approximation**

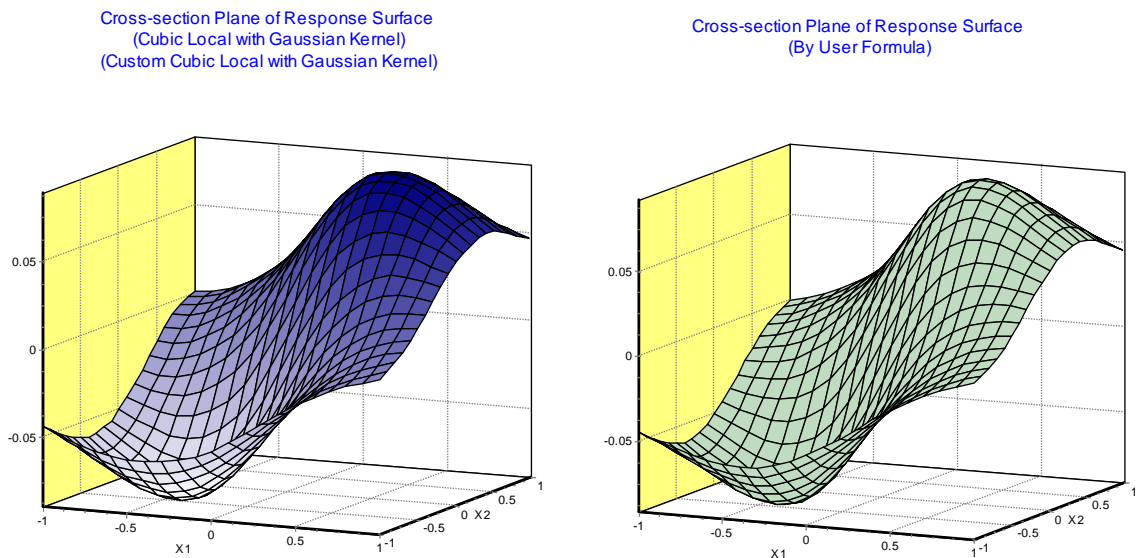


Fig. 33. Locally-weighted cubic approximation and the precise response function (to the right)

#### 4.5. Approximation with kriging

For kriging approximation, for each response  $m+1$  parameters must be defined (the so-called hyperparameter values). KEDRO also provides default values, however in practice they must always be corrected in the window **Options**.

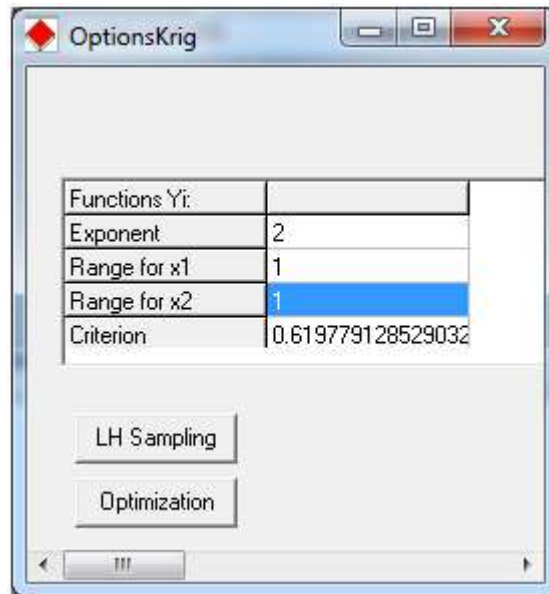


Fig. 34. Kriging Option window

In practice it is not advised to change the first parameter (Exponent), leaving it as 2. The other parameters may be changed manually, controlling the changes of the cross-validation error.

**An important remark: kriging approximations require significantly more processor time which rapidly rises depending on the number  $n$  of design runs!**

It is possible to carry out automatic hyperparameter optimization by pressing the button **LH Sampling**. In the window LH sampling the number of steps for optimization searching must be entered. This may require large amounts of processor time therefore it should first be checked by entering the value 20. To find optimal hyperparameter values, several thousand steps are required – this may take several hours of computing time if the number of design points exceeds 200.

Afterwards we press the button **Run LH sample for all functions**

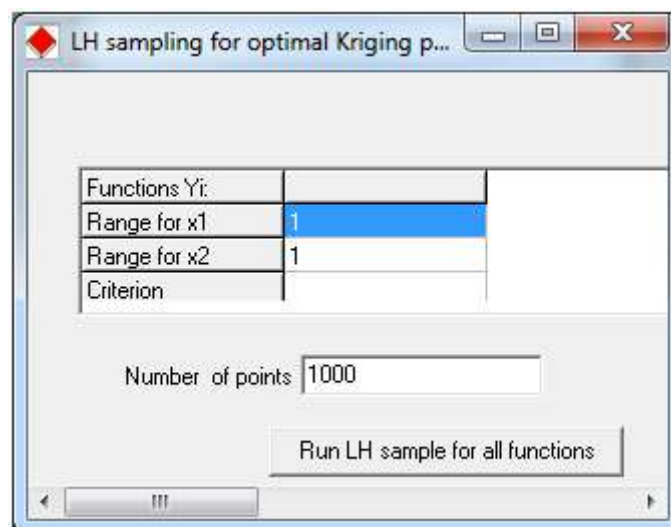


Fig. 35. Kriging options window

After the optimization of the kriging hyperparameters the adequacy of the approximation significantly improves, see Fig.36.

First-linear		0.00451452256255646%
Functions Yi:		
Sigma Cross	0.000317	
Sigma Cross%	0.571076%	
R2 adjusted		
F-Crit 99%		
Sigma	0.000000	
Sigma%	0.000000	
MeanExpValue	-0.000223	
StDev of Exp	0.055472	
Exp. Range	0.180635	
MaxError	0.000000	
Bad Point No.	37	
Max Rel Error	200.00%	
BadRelPointNo.	13	
Max Cook Dist.		
Suspicious point	4	
No.ofActualExp	50	
Filtered STD		

Fig. 36. The adequacy table for kriging approximation after parameter optimization

In one-dimensional graphs it is possible to view the cross-sections of the approximated function and the precise function in the same graph with added experimental points, see Fig. 37.



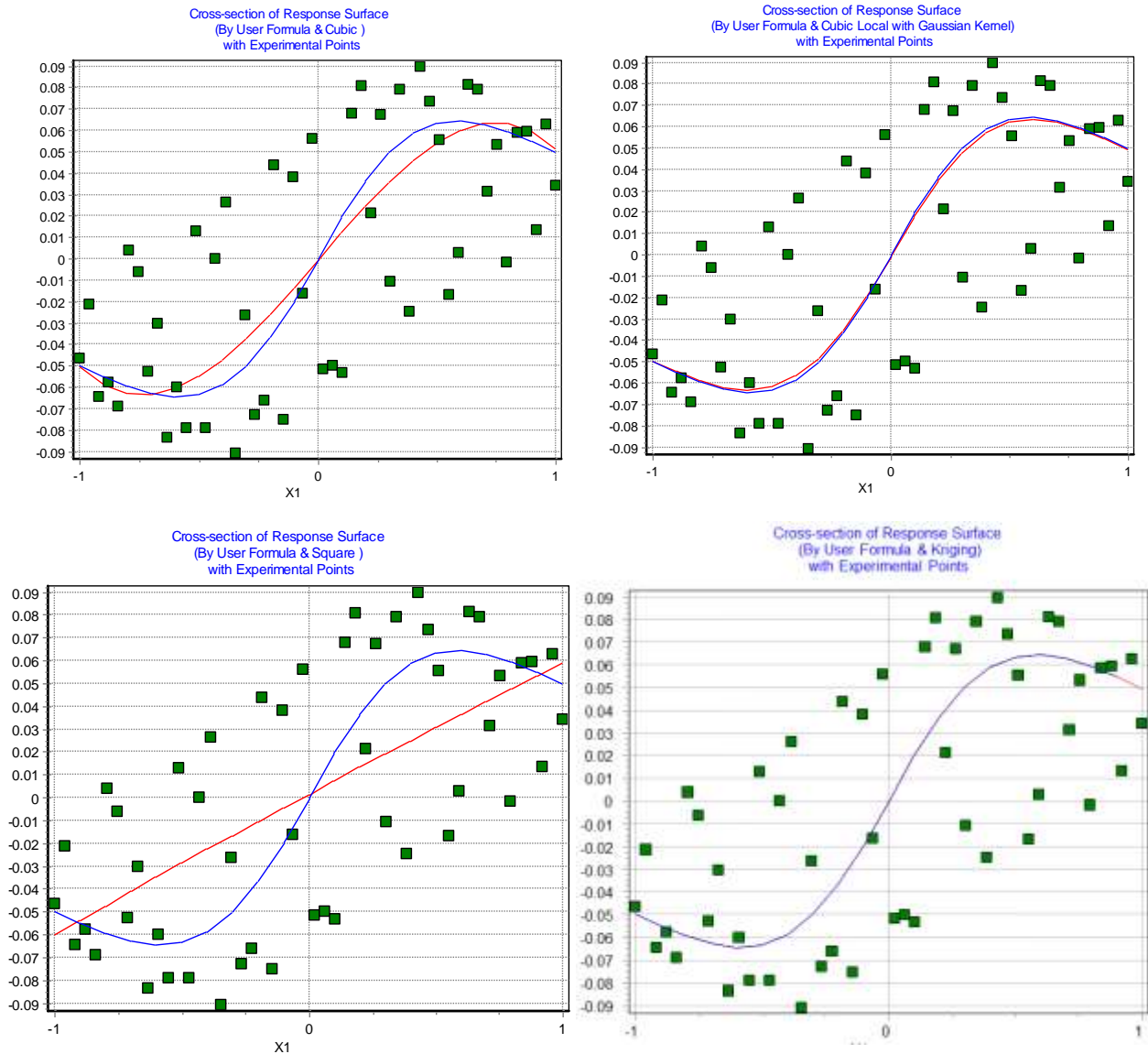


Fig. 37. Cross-section graphs for the approximated function and the precise function (top left - global cubic, top right - locally-weighted cubic, lower left - global linear, lower right - kriging)

As can be seen, in this example with a determined response, kriging is more precise than the locally-weighted polynomial approximation. The relative error of approximations for 10 000 test points (as a percentage of the standard deviation of the function values from the mean value) is 15.9% for the global cubic, 2.1% for the locally weighted cubic, 0.46% for kriging and 42% for the global linear approximation.

**Important: Experimental points in this graph should be considered to be scatterplots, since the approximation graphs are cross-sections for two argument response functions for the value  $x_2 = 0.0$ , each experimental point has its own value of the component  $x_2$  that is not visible in the graph. This applies to all graphs where the number of the graph variables is lower than the total number of the experiment input factors.**

For the approximation of numerical experiments, kriging is frequently the best method. For example, the same response function, using only an 11-point MSE LH design:

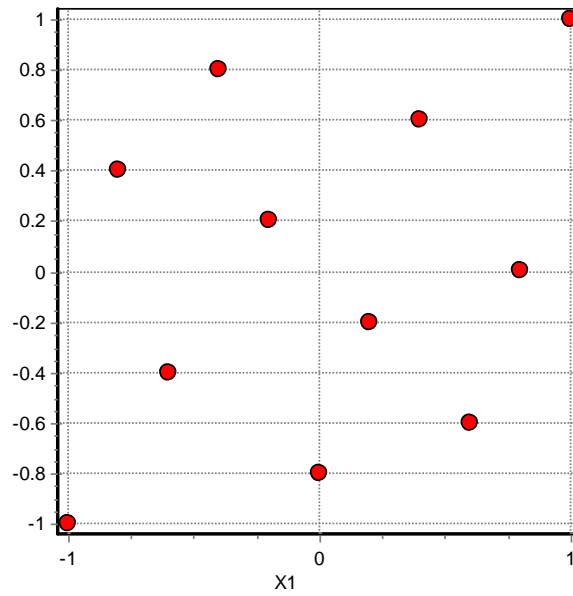


Fig. 38. 11 point 2 factor MSE-optimal LH design

and kriging hyperparameters optimized after cross-validation, an approximation is obtained with the relative error 12%, see Fig.39. If the mean relative error would be calculated relative to the range of the response change, then it would be only 4%

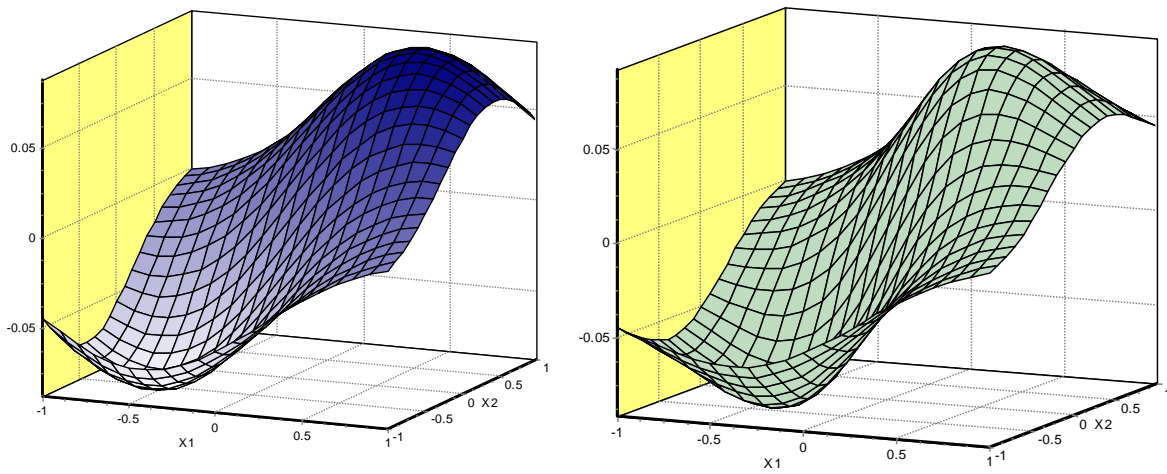


Fig. 39. Kriging approximation for an 11 point experimental design (to the left) and the precise function (to the right)

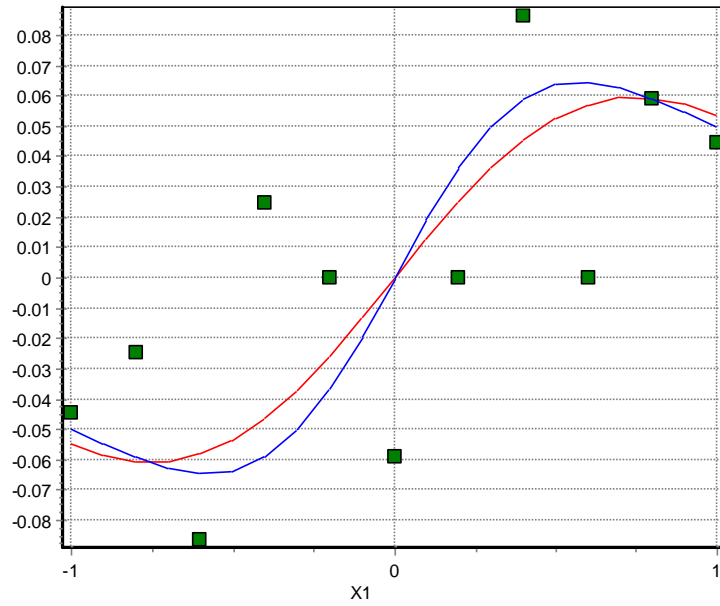


Fig. 40. The cross-section graph, the precise function is in blue, the kriging approximation is in red, 11 point 2 factor experiment

If significant noise is added even to a 50 point two factor Standard test function with 30% standard deviation, then see Fig.41.

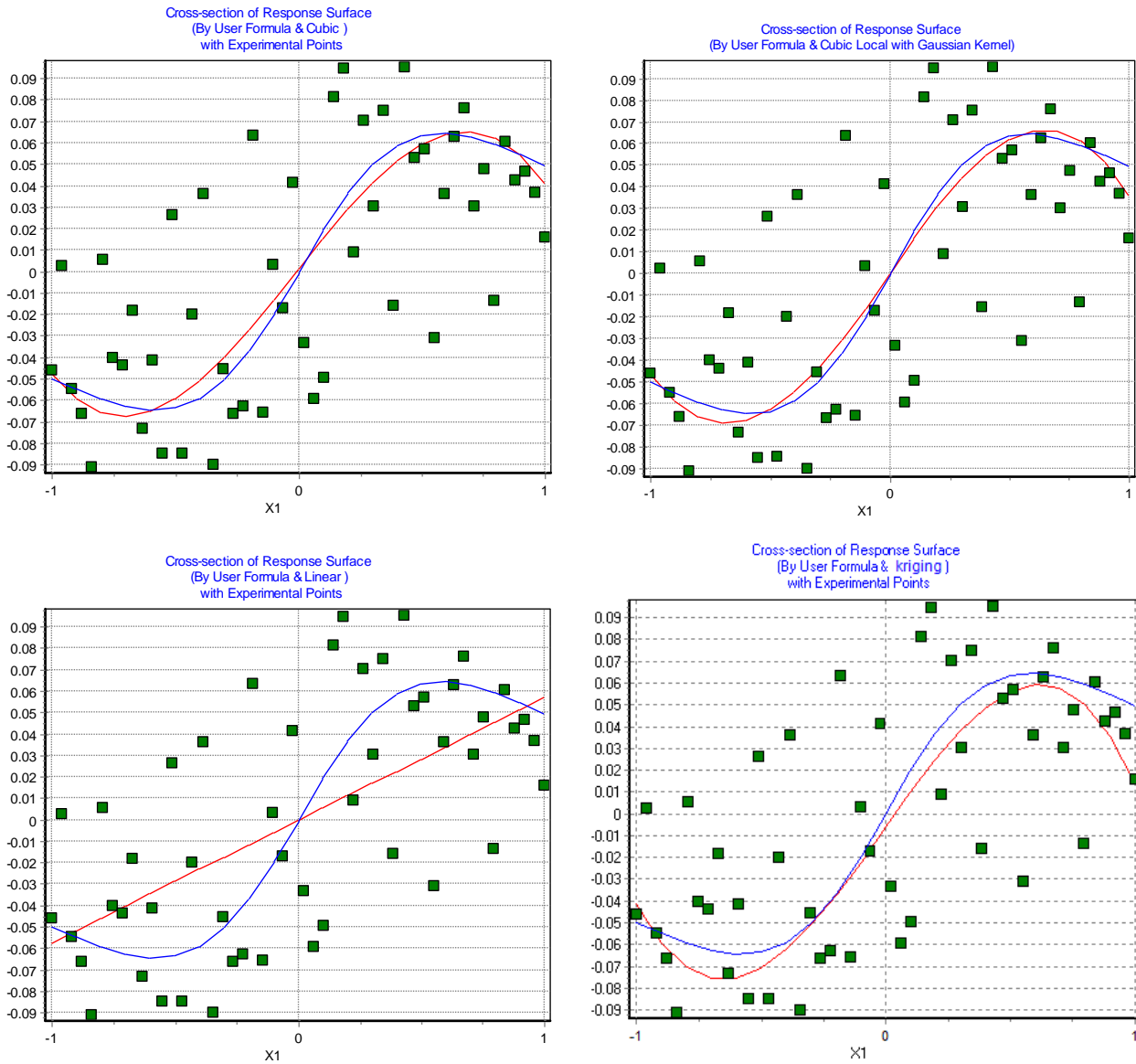


Fig. 41. The cross-section graphs for the approximated and the precise function with very noisy measurements (Top left – global cubic, top right – locally weighted cubic, lower left – global linear, lower right – kriging)

The relative error for global cubic approximation is 22%, the actual error for local cubic approximation is 21%, for global linear – 42%, for kriging – 21.5%. As can be seen, in the case of noisy responses, kriging does not give better results than locally weighted approximations

## 5. Optimization

Optimization is possible after the approximation of results. KEDRO may also be used to solve optimization tasks where approximation is not used (selecting Optimization - Direct without  $Y_i$ ), see Fig. 42.

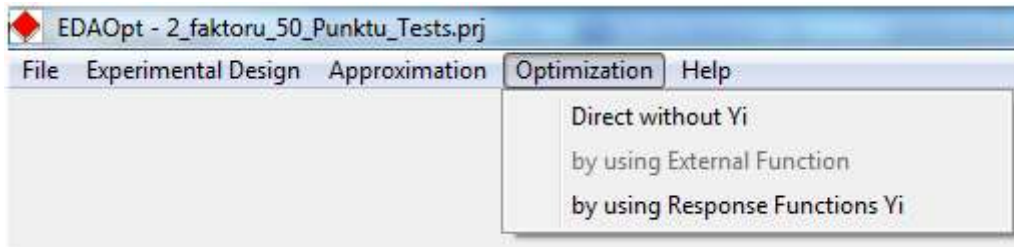


Fig. 42. Starting optimization

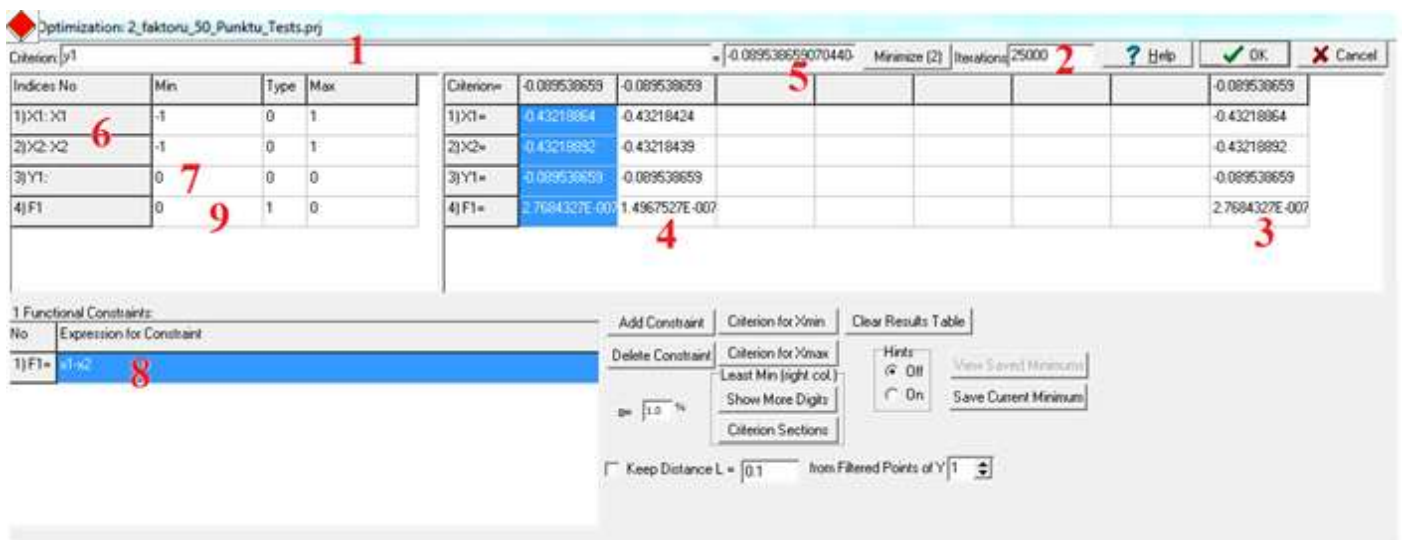


Fig. 43. The optimization window

### Information to be entered:

- The target function. The expression may contain input variables  $x_1, x_2, x_{12}$ , approximated responses  $y_1, \dots, y_{12}$ .  
The functions described in Appendix 1 may be used. Multiplication must be signified by the asterisk \*. The dot must be used as the decimal separator in numbers. KEDRO only looks for the target function minimum. To find the maximum, the sign of the target function must be reversed.
- Number of iterations. It determines the precision with which the global minimum will be found. If locally weighted polynomials and kriging are not used in the approximation, then the number of iterations may safely exceed 100 000.  
If locally weighted cubic polynomials are used, the number of iterations should be increased gradually, beginning from few thousands. Kriging requires particular caution, 100 000 kriging iterations may require several hours of processor time. The minimization runs must be repeated, gradually increasing the number of iterations and comparing the obtained results.
- The best result from repeated runs
- The result of the last run
- The value of the target function obtained in the current optimization process – changes during optimization.

6. Names of input variables (do not use names in expressions!) and the range of change. Type – zero, if the variable may have continuous values from the given range. If the variable is discrete, then Type is the discretization step, for example, Type = 1, if the variable is an integer.
7. The acceptable range of responses  $y_i$ . If Type=0, then the constraint is ignored, if Type = 1, then only the minimal value is taken into account, if Type =2, then only the upper constraint is active, if Type>2, then both the lower and the upper boundary is active.
8. Constraint expressions  $f_1, f_2, \dots, f_6$ . Notated in the same way as the target function. Can be entered after pressing the button **Add Constraint**.
9. Acceptable minimal and maximal values of the constraint functions. Type has the same meaning as in paragraph 7.

In the example, it can be seen that the constraint  $x_1 - x_2 \geq 0$  is active. Let us compare the located minimum with the precise minimum of the “Standard test function”:

Criterion: $1/(1+(0.05-x_1)^2+(0.05-x_2)^2)-1/(1+(-0.05-x_1)^2+(-0.05-x_2)^2)$			
Indices No	Min	Type	Max
1) X1: X1	-1	0	1
2) X2: X2	-1	0	1
3) Y1:	0	0	0
4) F1	0	1	0

Criterion=	-0.091513119	-0.091513119
1) X1=	-0.41028697	-0.41028794
2) X2=	-0.41028933	-0.41028851
3) Y1=	-0.089367042	-0.089367044
4) F1=	2.3643592E-008	5.6641479E-007

Fig. 44. Minimization of the analytic function

For the approximated function the minimal value -0.0895 was found in the point (-0.432, -0.432), where the precise function had the value -0.0913. For the precise function the minimum -0.0915 was in the point (-0.410, -0.410), meaning that the optimal criterion value was found with approximately 0.2% error.

#### The meaning of other elements in the optimization window

Using the button Criterion Sections the location of the optimum point and the constraint regions may be viewed visually, see Fig. 45.



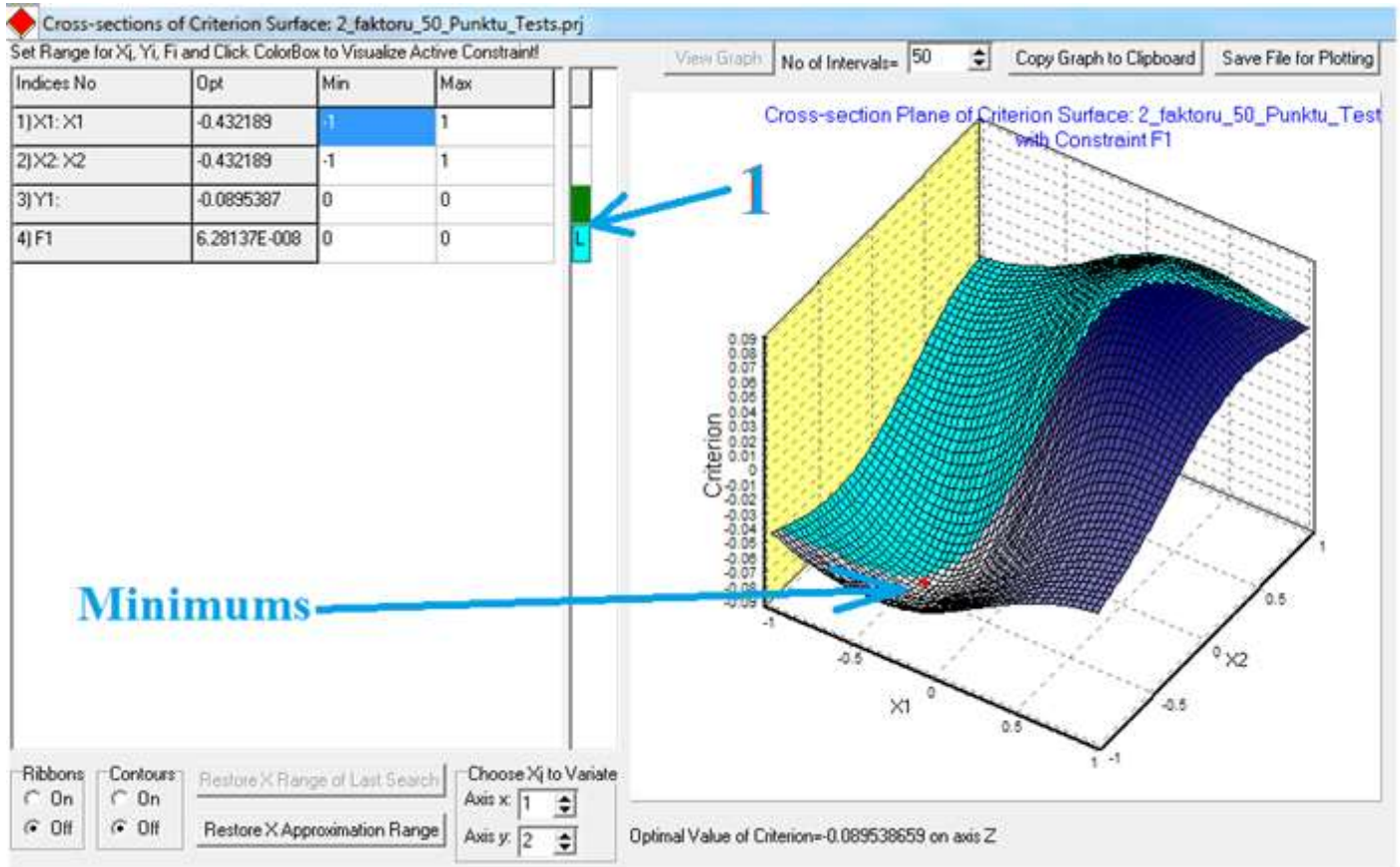


Fig. 45. Visualization of the criterion cross-sections

Pressing the colored rectangles 1, it is possible to turn on and off the graphs of the target function and constraints. The red point signifies the located minimum. The values of the invisible input parameters are the optimal values therefore the graph is called the criterion surface.

The buttons Save File for plotting allow the user to save the graph in a file as a table that may be read from visualization programs, for example, Systat SigmaPlot, MathCad or Excel.

### 5.1. Multi-criterion optimization

KEDRO v. 1.01 also allows multi-criterion optimization (multiobjective optimization). The Pareto boundary surface is found and its analysis allows the user to select the preferred point from the set of many optimal solutions.

For example, let us review the classic one variable two criteria Schaffer's problem [5]:

$$\begin{aligned} \min f_1(x) &= x^2 \\ \min f_2(x) &= (x - 2)^2 \end{aligned} \quad (25)$$

with the constraint

$$-10 \leq x \leq 10 \quad (26)$$

The precise solution of the optimization task is shown in the form of a Pareto boundary set in Fig. 46.



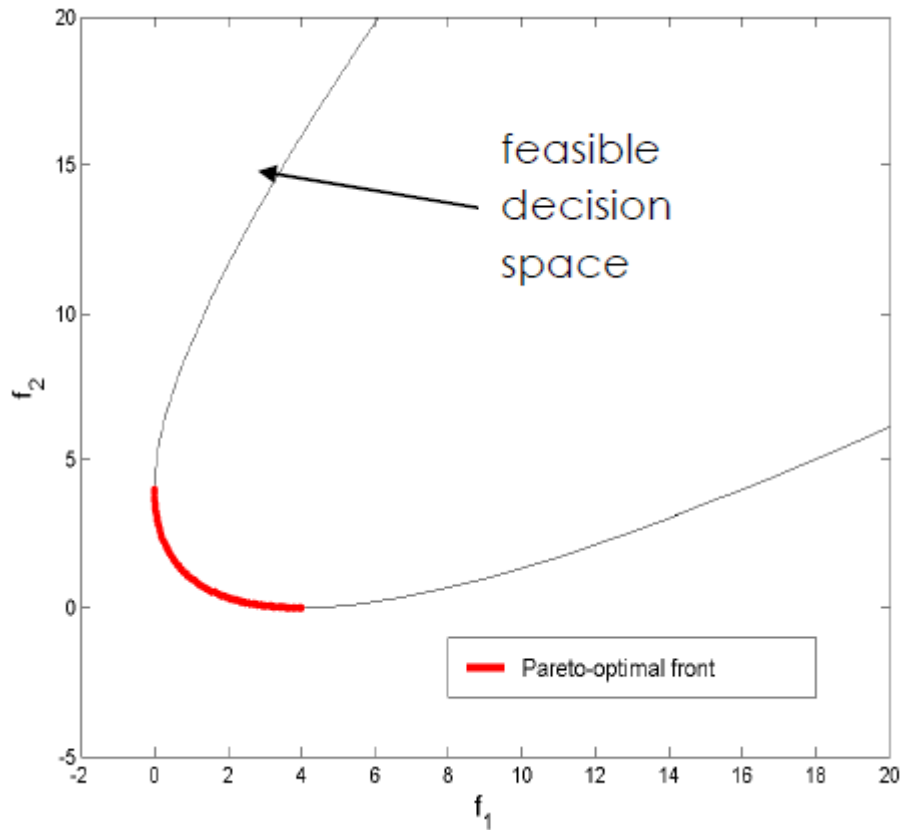


Fig. 46. Solution of the Schaffer's problem

After entering two criteria in the KEDRO multi-criterion optimization window and limiting the number of points to be found by 100, we press the button **Find Pareto Points**, see Fig. 47.



Fig. 47. Finding two criteria Pareto boundary set points

Pressing the button **Plot Pareto Points** we obtain the image:

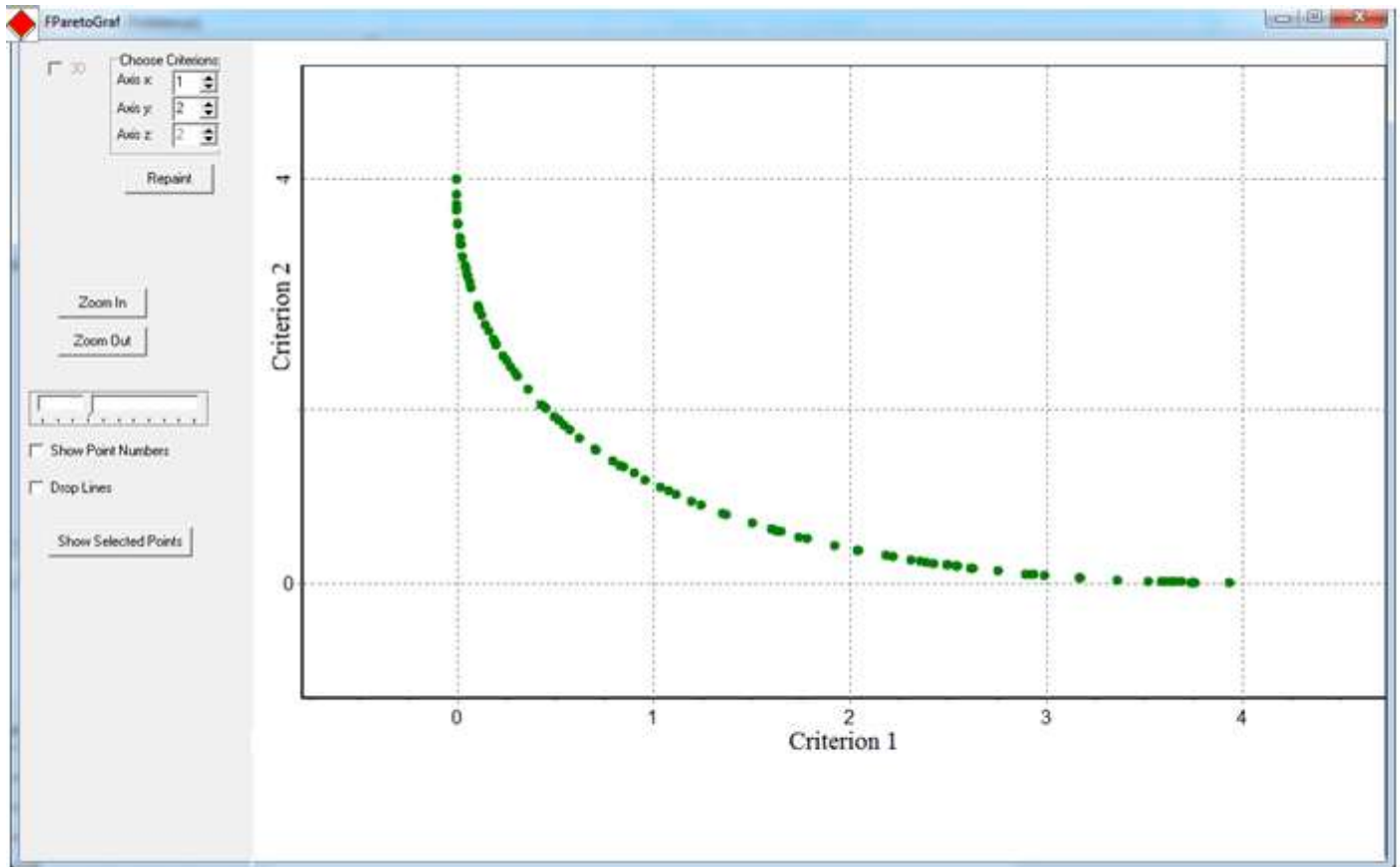


Fig. 48. Solution of the Schaffer's problem with KEDRO

By pressing the mouse button on any of the Pareto boundary set points the user can obtain full information about its: input factor, criteria and constraint function values. Of course, in criteria expressions the designations of the approximated functions  $y_1, \dots, y_{12}$  can also be used.

## 6. Parametric and non-parametric approximations.

Currently in regression analysis two different approaches are used: parametric approximation and non-parametric approximation. In parametric approximation, the functional relation between the input and response variables is known (at least the user has for some reason chosen them), for example, in the form of some analytical function that contains parameters that must be adapted to the experiment data. For example, approximation of results with a linear function is parametric regression. A significant difference of *non-parametric* regression consists in the fact that there is no *a priori* knowledge of the form of the actual function that should be found. In this case the responses are also modeled with functions that contain parameters, but the number of these parameters is very large and they do not have an easily interpretable and explainable meaning. The term “non-parametric” is not particularly successful but it has been accepted historically. The essence of the matter is that in the case of non-parametric approximation the regression function does not exist as an analytic expression that may be written in open form with unknown parameters. Sometimes parametric approximation is called “*global approximation*” and non-parametric approximation “*local approximation*”. In non-parametric approximation assumptions about the form of the regression function are not made, it is only assumed that this function is smooth (that is, it has first order derivatives – their values, however, are usually not calculated or used).

Therefore parametric approximations give as a result an approximated model that can be written down in the form of an analytical function (regression function). The coefficients of this function  $\beta_1, \beta_2, \dots, \beta_L$  are found using the least squares method.  $L$  is the number of the regression function members (coefficients). A regression function that has linear coefficients  $\beta$ , may be written as

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^L \beta_i g_i(\mathbf{x}) + \varepsilon \quad (27)$$

where  $\varepsilon$  – noise added to the response, (variance), usually it is assumed that it corresponds to the normal distribution with the mean value zero and constant variance. In the subsequent formulas  $\varepsilon$  will not be used in obvious form.

**Note. Bold symbols** everywhere signify vectors or matrices.  $\mathbf{x}$  signifies a vector column that consists of the input variables  $x_1, x_2, \dots, x_m$ .

### 6.1. Polynomial approximations

The simplest and the most frequently used parametric approximations are polynomials with  $m$  variables. KEDRO uses first, second and third order polynomials.

The number of members  $L$  for a first degree (linear function) polynomial with  $m$  variables (factors) is  $m+1$ , for a second degree (quadratic) it is  $(m+1)(m+2)/2$ , for third degree polynomial  $L = (m+1)(m+2)(m+3)/6$ .

The minimal number of experiment runs necessary to implement polynomial approximation is equal to the number of the regression function members  $L$ . Experimental designs for which the number of runs is equal to the number of the regression function coefficients, are called *saturated designs*. The minimal number of members may be used only if it is clearly known that the experiment response model really consists of polynomial functions of input variables and does not contain noise (are determined).

In practice saturated designs are almost never used, especially if the result of the response to be measured contains random or systematic error. Approximation with second degree polynomials historically is called the Response Surface Method (RSM), it is most frequently used in various type of engineering calculations. In this case practical experience shows that the number of runs for the approximation of physical experiment data must be 1.5 - 2 times greater than the minimal possible number of runs, that is,  $1.5 \times (m+1)(m+2)/2$  to  $(m+1)(m+2)$ . It must be noted that in the case when the response function is difficult to approximate with second degree polynomials, increasing the number of runs using the same regression function does not lead to significant improvement of the approximation (reduction of the approximation prediction error).

Finding regression function coefficients that give the best correspondence to the experiment results usually is carried out using the Least squares method (*LSM*). The sum of the quadratic deviation  $\Phi$

$$\Phi(\boldsymbol{\beta}) = \sum_{j=1}^n \left( y_j - \sum_{i=1}^L \beta_i g_i(\mathbf{x}_j) \right)^2 \quad (28)$$

is minimized, where  $y_1, y_2, \dots, y_n$  – registered responses  $n$  in the experiment points  $\mathbf{x}_j$ .

The minimization of the total quadratic deviation is carried out by equating the partial derivatives  $\frac{\partial \Phi}{\partial x_i}$  to zero.

Since the coefficients  $\boldsymbol{\beta}$  in the quadratic deviation function are quadratic, then after equating the derivatives to zero a system of linear equations is obtained – once it is solved, the coefficient values are obtained that give the best fit of the regression function to the experimental data.

## 6.2. Non-parametric approximations

Non-parametric approximations are approximations for which the result – the mathematical model – may not be written down in the form of an analytical function. From the class of non-parametric approximations, KEDRO uses locally-weighted polynomial approximations and kriging.

## 6.3. Mathematical foundation for locally-weighted polynomial approximations

In locally-weighted approximation, for each point of the experimental space  $\mathbf{x}^*$ , where the user wants to find out the value of the predicted response  $\hat{y}(\mathbf{x}^*)$ , the quadratic deviation function is minimized, weighing the deviations from the experimentally obtained values depending on the distance of the specific experimental point  $\mathbf{x}_i$  from the point of interest  $\mathbf{x}^*$ :

$$\Phi(\boldsymbol{\beta}) = \sum_{j=1}^n k(\|\mathbf{x}^* - \mathbf{x}_j\|) \left( y_j - \sum_{i=1}^L \beta_i g_i(\mathbf{x}_j) \right)^2 \quad (29)$$

where  $u = \|\mathbf{x}^* - \mathbf{x}_j\|$  – distance in scaled Euclid space  $[-1, 1]^m$  between points  $\mathbf{x}^*$  and  $\mathbf{x}_j$ ,  $\kappa(u)$ – weighing (that is, kernel) function. It is possible to use different weighting functions whose main property is the following: for  $\kappa(0)$  they assume maximum values,  $k(u)$  decreases as the distance  $u$  increases. In function smoothing theory weights are chosen so that  $\int_{-\infty}^{\infty} k(u) du = 1$ , for weighted polynomial approximations this condition is not significant. For classic weighing functions the parameter of distance value is also used – once it is exceeded, the weighing function returns the value zero. This is called bandwidth, usually defining it by the so-called number of nearest neighbors. Points that are not included in the set of nearest  $\mathbf{x}^*$  neighbors, are not taken into account when predicting the value  $\hat{y}(\mathbf{x}^*)$ . The most popular kernel functions are shown in the table:

Table Popular kernel functions

Kernel	Equation
Uniform	$k_0(u) = \frac{1}{2} 1 ( u  \leq 1)$
Epanechnikov	$k_1(u) = \frac{3}{4} (1 - u^2) 1 ( u  \leq 1)$
Biweight	$k_2(u) = \frac{15}{16} (1 - u^2)^2 1 ( u  \leq 1)$
Triweight	$k_3(u) = \frac{35}{32} (1 - u^2)^3 1 ( u  \leq 1)$
Gaussian	$k_\phi(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right)$

In practice, the following weighing function with adjustable degree used by KEDRO has proved to be very easy and good to use [1]:

$$k_G(u) = \exp(-\alpha u^2) \quad (30)$$

This weighing function does not require setting the bandwidth (number of nearest neighbors) – it is automatically reduced if the coefficient  $\alpha$  is increased.

#### 6.4. Kriging method

One of the non-parametric methods that is used more and more frequently, is kriging. The name of the method derives from the name of the South African mining engineer D.G. Krige, who created this method in the middle of the 20<sup>th</sup> century and used it for prediction of mineral deposits. Approximately at the same time this method was also proposed by L. S. Gandin, using it in meteorology. This high precision method views the response function as the implementation of a stochastic process. The mathematic form of the method is the following:

$$\hat{y}(\mathbf{x}) = f(\mathbf{x}) + z(\mathbf{x}), \quad (31)$$

where  $\hat{y}(\mathbf{x})$  is the evaluation of the unknown response function,  $f(\mathbf{x})$  is a known, usually polynomial function of arguments  $\mathbf{x}$  and  $z(\mathbf{x})$  is an implementation for the stochastic process with a zero mean value, variance  $\sigma^2$  and non-zero covariance. Similarly to the polynomial response surface model  $f(\mathbf{x})$  determines the “global” behavior of the model in the experimental region. Frequently instead of  $f(\mathbf{x})$  a constant is used – the mean value of the response surface in the experimental points.

Most frequently the *spatial correlation function*  $z(\mathbf{x})$  is used, for which

$$\text{cov}[Z(\mathbf{x}_i), Z(\mathbf{x}_j)] = \sigma^2 R(\mathbf{x}_i, \mathbf{x}_j), \quad (32)$$

where  $\sigma^2$  is process variance,  $\mathbf{R}$  is  $n \times n$  size correlation matrix whose elements  $R(\mathbf{x}_i, \mathbf{x}_j)$  express the correlation between the experimental design points  $\mathbf{x}_i$  and  $\mathbf{x}_j$ . The correlation function is usually chosen by the user. For this purpose different functions may be used, but KEDRO uses the most frequently used Gauss correlation function:

$$R(\mathbf{x}_i, \mathbf{x}_j) = \prod_{k=1}^m \exp(-\theta(x_{ik} - x_{jk})^p) \quad (33)$$

with  $p = 2$  and  $m$  parameters  $\theta$ , one for each dimension. Optimal parameter values for each approximation case are found, using the cross-validation method.

Kriging in many cases gives very precise approximations that are more precise than those given by other methods. This method has great advantages for strongly non-linear response functions. Another advantage of the method is the fact that according to the original assumptions it is relatively simple to calculate the possible prediction error, assuming some fixed confidence interval, for example, 95%.

However, the method also has its drawbacks. The most important – the method is quite labor intensive. Finding the parameters  $\theta$ , using the cross-validation or the maximum likelihood method, is a minimization task with  $m$  variables. For each run of this minimization calculation the inversion of a  $n \times n$  correlation matrix is necessary. This matrix in some cases may become singular, if in the experimental design some points are close to one another. Problems have been identified for some factorial designs and central composite designs. This method interpolates the results – in the experimental points the value of the prediction coincides with the registered response value (or its mean value, if the runs are repeated).

## Literature

1. J. Auzins, A. Janushevskis. Design and analysis of Experiments, Riga, Riga Technical University, 2007 (In Latvian).
2. R. H. Myers, D. C. Montgomery, C. M. Anderson-Cook. Response Surface Methodology. Process and Product Optimization Using Designed Experiments. Third edition, Wiley, 2009.
3. A. I. J. Forrester, A. Sóbester and A. J. Keane. Engineering Design via Surrogate Modelling. A Practical Guide. Wiley, 2008.
4. R. L. Mason, R. F. Gunst, J. L. Hess. Statistical Design and Analysis of Experiments. With Applications to Engineering and Science. Second Edition, Wiley, 2003.
5. J. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. 1st Int. Conf. on Genetic Algorithms, Pittsburgh, 1985.

## Appendix 1.

### Entering analytical expressions in the KEDRO program

The program KEDRO allows entering analytical expression in several places.

**Calculation of User Defined Function** allows to calculate the response function values, using analytical expressions – this is useful for testing the program and training the users.

To implement this possibility, the user must indicate the order number of the response function Y in the spinedit window and enter an analytical expression for an arbitrary function in edit window. The expression may contain factors X1, X2,...,Xm, where m is the number of factors, as well as constants and functions that are interrelated with operators that are listed below.

In order to calculate values for the indicated response function, the user must press the button "Calculate". Afterwards the expression for the next response function may be entered. When exiting the form by pressing the button "OK", all the entered response functions are assigned the calculated values.

In the optimization window the expressions may be entered in the criterion and the constraint columns. In these places in addition to the input variables X1, X2,..., Xm the expressions may also contain the approximated response function designations Y1, Y2, ..., Y12

#### Operators list

Operand	comment	example
+	addition	x1+x2
-	subtraction	x1-x2
*	multiplication	x1*x2
/	division	x1/x2
^	power	x1^x2

#### Functions list

abbreviation	comment	example
abs	absolute	abs(-10)
arccos	arccosine	arccos(1)
arcsin	arcsine	arcsin(1)
arctg	arctangent	arctg(2)
cos	cosine	cos(0.5)
cosec	cosecant	cosec(1)
ctg	cotangent	ctg(3.5)
exp	exponent	exp(2)
ln	logarithm to the base e	ln(2.718)
log10	logarithm to the base 10	log10(1000)
neg	negative	neg(-1)
pi	pi	pi()
random	random (Range: Integer) returns random number X within the range 0 <= X < Range. If Range is not specified, the result is a real-type random number within the range 0 <= X < 1.	random(); random(5)
sec	secant	sec(1)
sin	sine	sin(0.5)
sqrt	square root	sqrt(4)
sum	sum of args	sum(1,2)
tg	tangent	tg(0.4)
int	integer part of real variable	int(1.91)      result=1.0