

Getting Started: UM Experiments Module

This manual leads you through basics of describing, running and analyzing the scanning and optimization projects using the Universal Mechanism software. It assumes that you studied the **gs_UM.pdf**¹ chapter, which is devoted to general concepts of simulation using Universal Mechanism, and you know how to fulfill simple operations: create new model, add graphical objects, bodies and joints, generate and compile equations of motion.

Three examples of using the scanning projects are considered. The first one is devoted to free and forced oscillations. The second one is devoted to scanning the railway vehicle dynamics (see **gs_UM_Loco.pdf**²). The example of using service of distributed calculations is shown in the last section.

The program tool for parametric scanning is available within **UM Experiments** module. To check if the module is in your version of UM run **UM Simulation** program, point to the **Help/About** menu command. The **About** window appears. You can see the list of the supported modules in the **Configuration** section.

¹ http://www.umlab.ru/download/50/eng/g_s_um.pdf

² http://www.umlab.ru/download/50/eng/g_s_um_loco.pdf

GETTING STARTED: UM EXPERIMENTS MODULE.....	1
1. PREFACE	3
2. SCANNING PROJECT: CREATION AND ANALYSIS.....	4
2.1. Description of scanning project	5
2.1.1. Creating new project.....	5
2.1.2. Loading a model.....	6
2.1.3. Renaming the family of alternatives.....	7
2.1.4. Hierarchy of parameters.....	7
2.1.5. Parameters of the model.....	9
2.1.6. Initial conditions	9
2.1.7. Finish conditions	9
2.1.8. Variables to save.....	10
2.1.9. Forced oscillation	11
2.2. Running the project.....	14
2.3. Analyzing obtained results	16
2.3.1. Results of separate experiments	16
2.3.2. Summary graphs	18
3. SCANNING: IMPLEMENTATION TO RAILWAY VEHICLE DYNAMICS.....	20
4. USING THE SERVICE OF DISTRIBUTED CALCULATIONS	21
4.1. Preparing for running.....	22
4.2. Loading scanning project.....	23
4.3. Running	24
5. OPTIMIZATION PROJECT: CREATING AND ANALYSIS	28
5.1. Preparing the optimization project.....	29
5.1.1. Creating new project.....	29
5.1.2. Loading a model.....	30
5.1.3. Parameters to optimize.....	31
5.1.4. Initial conditions	33
5.1.5. Stop conditions	34
5.1.6. Solver parameters	36
5.2. Objective function	37
5.3. Running the project.....	42

1. Preface

It is often required in engineering practice to carry out series of numerical experiments, for example to analyze dynamical behavior and sensitivity of mechanical system or to find out optimal parameters of a system. The built-in module **UM Experiments** includes a set of tools (*scanning, optimization, approximation*) for advanced analysis of dynamics of mechanical systems.

All tools automate fulfillment of series of numerical experiments, record course of experiments and save results of experiments on a hard disk for posterior analysis. Thus, the designer is released from monotonous execution of series of numerical experiments "manually" what saves working hours and removes errors, which people unfortunately incline to do. In other words, the researcher defines the design of experiments for scanning and approximation or parameters, its limits and precision and goal function for optimization. Then the project is started and executed automatically. Current process statistics is available during the execution: number of experiments done, time left. Series of numerical experiments are resistant to shut-down of power supply. In that case all results are saved on a hard disk and results of the latter experiments are only lost.

There is a possibility to plot a time history of any saved performance of dynamical behavior of a mechanism. Moreover, the designer can plot so-called summary graphs and surfaces. All implemented tools have no limitation in number of parameters. In other words, they all are multi-parametrical.

Dimensions of projects are set by the designer so as to solve the specify problem. But on the other hand the designer has to take into account calculating efforts that are necessary for the project. Every tool has its own merits and demerits. However they all give the designer possibilities to solve quite many problems devoted to optimization of mechanical systems.

2. Scanning project: Creation and Analysis

Describing, running and analyzing a scanning project for an oscillating system is discussed in this section. The process of creating the model is shown in details in the **gs_UM.pdf**¹ manual.

Firstly, make sure that the model of the oscillating system is in your computer: create the model according the **gs_UM.pdf** manual or check it in the `{um_root}\samples\tutorial\oscillator` directory or download it using the following link <http://www.umlab.ru/download/50/oscillator.zip>, see Fig. 2.1.

More detailed information is in the **Part 6** of the UM User's Manual, see `{um_root}\manual\06_UM_Experiments.pdf` or http://www.umlab.ru/download/50/eng/06_um_experiments.pdf.

Completely prepared scanning project, which is considered in this section, is available in the `{um_root}\samples\tutorial\scan2` or in the Internet at: <http://www.umlab.ru/download/50/scan2.zip>.

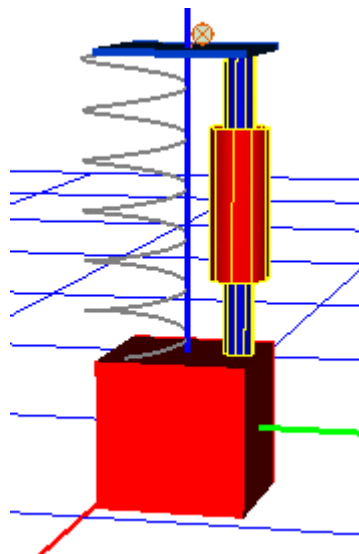


Figure. 2.1. Oscillating system

¹The **gs_UM.pdf** file you can find in the `{um_root}\manual` directory or download using the following link: http://www.umlab.ru/download/50/eng/g_s_um.pdf

2.1. Description of scanning project

Here we consider the example of scanning project for oscillating system. We will scan the dynamical property of the system depending on damping coefficient for free oscillation and frequency of exciting force for forced oscillation.

2.1.1. Creating new project

1. Run the **UM Simulation** program.
2. To create a new project from the **Advanced analysis** menu select **Scanning: new project**.
3. Input the full path to the scanning project, including project name, see Fig. 2.2.
4. Click the **Create** button and after that a window for the description of new scanning project will appear, see Fig. 2.3.

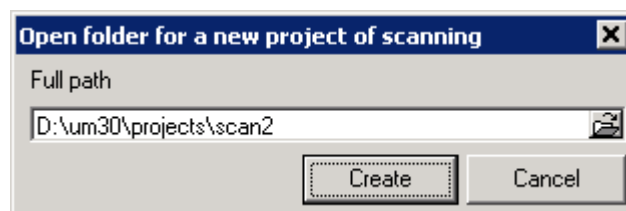


Figure 2.2. New project dialog

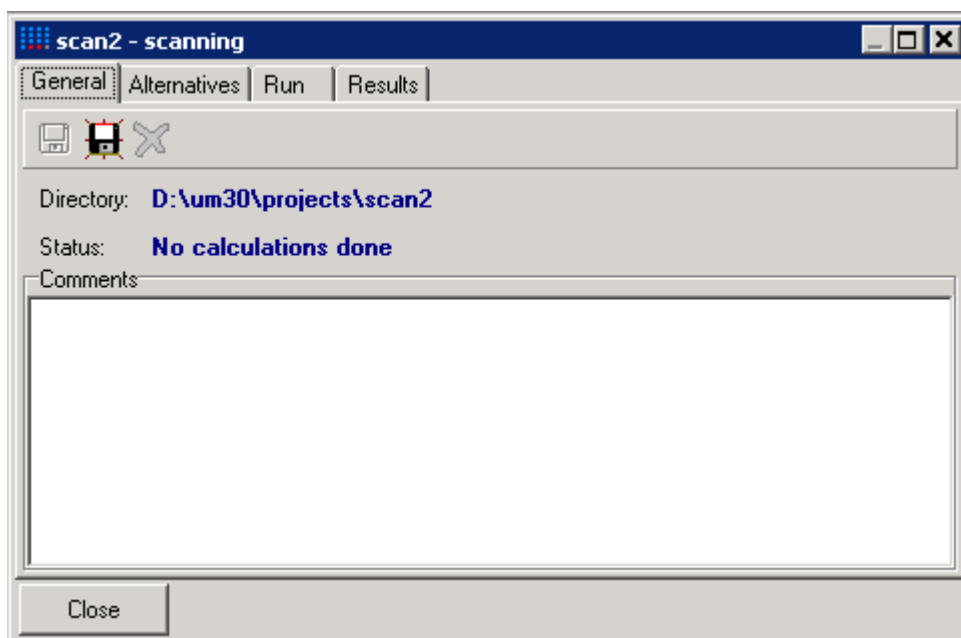


Figure 2.3. Scanning project: in the beginning

2.1.2. Loading a model

1. Select the **Alternatives** tab.
2. Click the **+** button (Add family of alternatives).
3. Select the **oscillator** model in the open dialog.

Then the **oscillator** model is loaded and added to the **Family of alternatives** list (see Fig. 2.4).

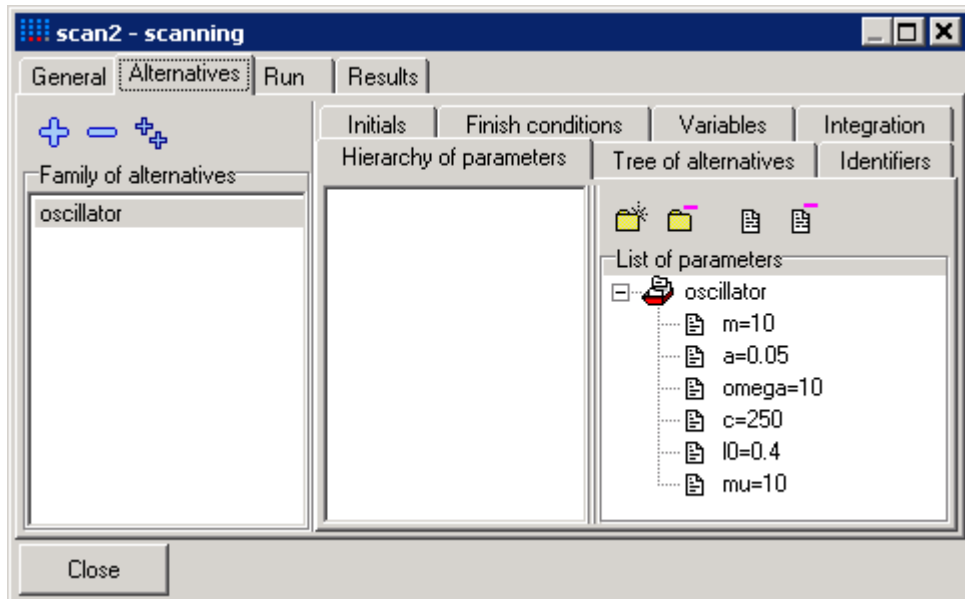
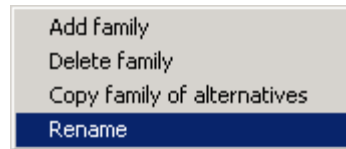


Figure 2.4. Adding new model

2.1.3. Renaming the family of alternatives

Now we will rename the first family of alternatives.

1. In the **Family of alternatives** list select the first family and from the context menu select **Rename**.



2. Set the name for the family to **Free vibrations**.

2.1.4. Hierarchy of parameters

Two cases are considered here: free and forced oscillations.

The first model (family of alternatives) is devoted to free oscillation analysis. The influence of the damping coefficient on the damped oscillation process is of interest.

1. In the **List of parameters** (in the right part of the form, see Fig. 2.4) click the **mu** (damping coefficient) parameter.
2. **Parameter values** window appears. Add values {**0, 10, 20, 30, ... , 100**}, 11 values total, see Fig. 2.5.
3. Click the **OK** button and come back to the project window. Parameter group **mu** appears in the **Hierarchy of parameters** tab, see Fig. 2.6.

So we describe eleven numerical experiments in this family of alternatives. Common settings for all of the experiments we will describe below.

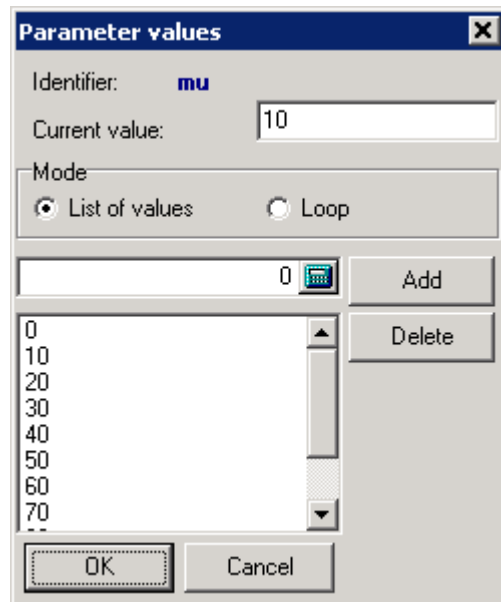


Figure 2.5. Parameter values

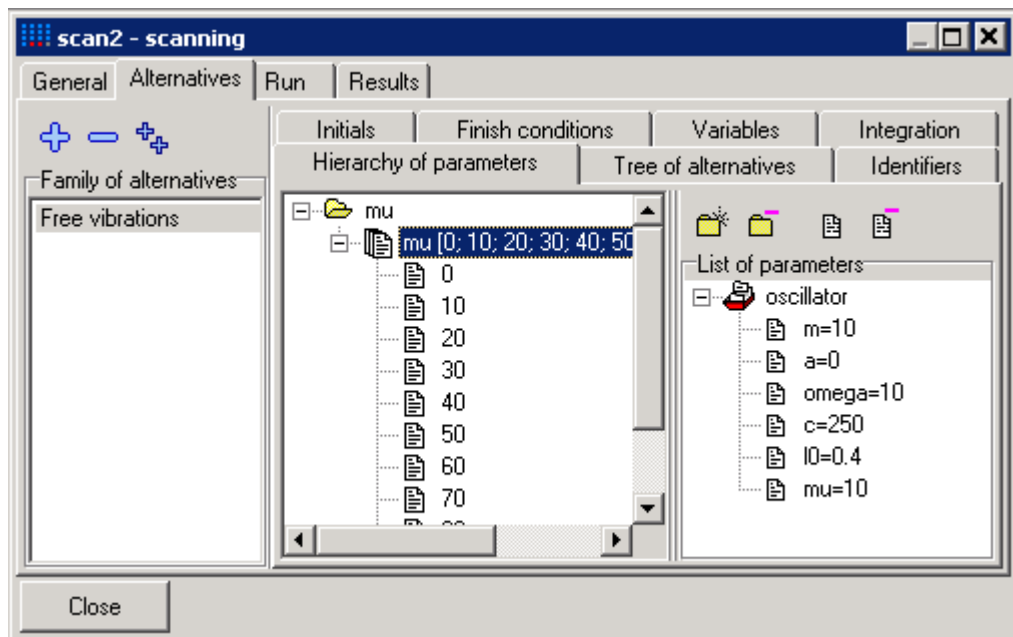


Figure 2.6. Hierarchy of parameters

2.1.5. Parameters of the model

In order to model the case of free vibrations it is necessary to fix the **Top** body – set the **a** parameter to zero.

1. Select the **Identifiers** tab.
2. Set **a** to **0**.

2.1.6. Initial conditions

1. Select the **Initials** tab.
2. Set the **Coordinate/1.1** to **0.1**.

Position of the **Brick** at zero coordinates is quite close to a steady state position. We shift the **Brick** in order to increase the amplitude of oscillation.

2.1.7. Finish conditions

Here you can describe finish conditions for each numerical experiment in the current family (see Fig. 2.7). Finish conditions are formulated in the following way: “Interrupt a numerical experiment if at least one of the conditions is satisfied”. Using scanning project you can set finish condition as

Variable [Condition] Numerical value.

You can use any variable from the **Wizard of variables** as stop criterion. By default the following finish condition is formulated:

Time >= 10 c.

It means every numerical experiment finishes when 10 seconds of simulation time expire.

1. Set simulation time to **25** seconds, see Fig. 2.7.

Note. You can set any *variable* as a criterion for finish conditions. Use just need to create a *variable* in the **Wizard of variables** and then drag it to the variable box (now you can see *Time* in that box).

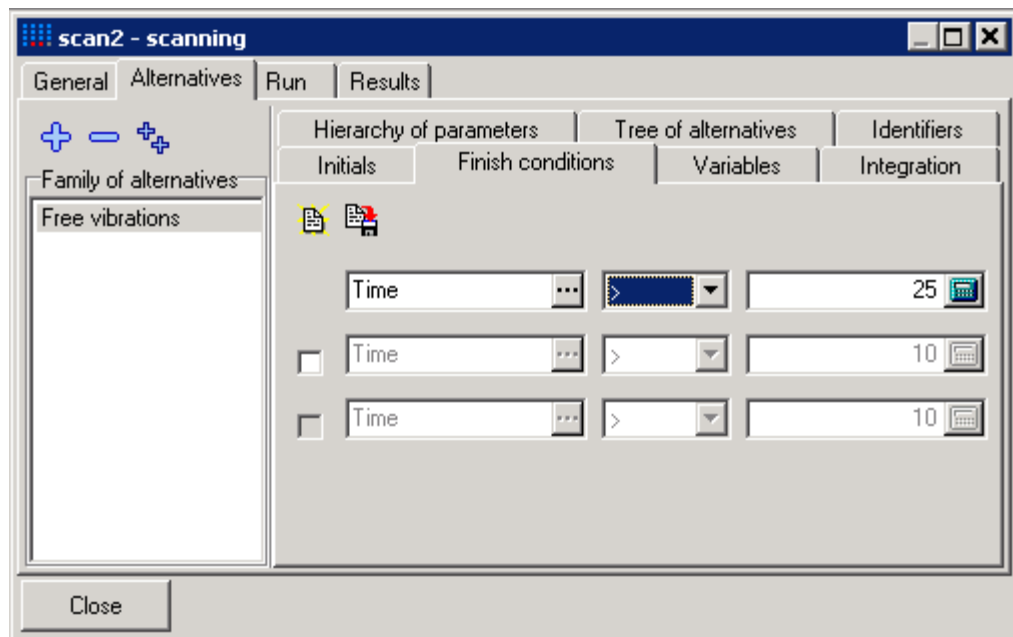



Figure 2.7. Finish conditions


2.1.8. Variables to save

1. Select the **Variables** tab.

Here you can see the **List of variables**, which will be saved for the each numerical experiment. In general you need to fill this list of variables with those variables you will analyze after the project fulfils.

2. Using the  button rename the **No name** tab to **Position**.

Now we will create the new variable – vertical position of the **Brick** – and drag it to this tab.

3. Open the **Wizard of variables (Tools/Wizard of variables)**.
4. Point to the **Linear var. (linear variables)** tab, select the **Brick** in the list of bodies in the left, set **Component** to **Z**.
5. Click the  to create new variable and then drag it to the **Position** tab.
6. Close the **Wizard of variables**.

After all your list of variable should look like in Fig. 2.8. We described the family of alternatives for the case of free vibrations and now come to the case of forced oscillation.

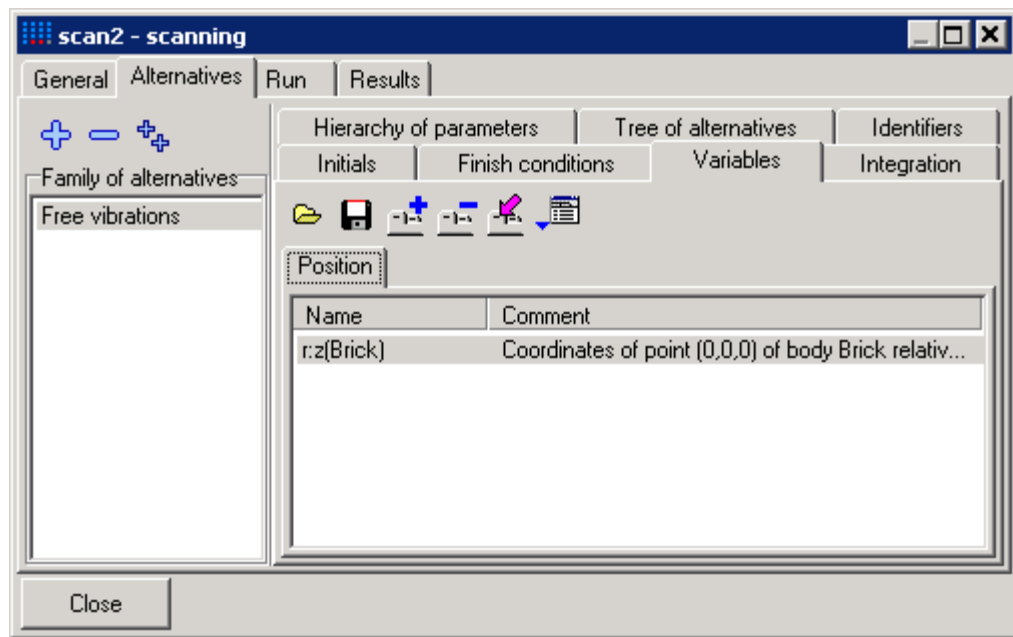


Figure 2.8. Variables to save

2.1.9. Forced oscillation

Now we will copy the first family of alternatives, set the damping coefficient to zero and then scan the dynamics of the system depending on the frequency of the exciting force. There will be the resonance case when the natural frequency of the system and the frequency of the exciting force coincide.

Copying the family of alternatives

1. In the **Family of alternatives** list select the **Free vibration** and from the context menu select the **Copy family**. The second family of alternatives **Free vibrations (1)** appears in the list.
2. Select the second family and with the help of context menu rename it to **Resonance**.

Hierarchy of parameters

1. Point to the **Hierarchy of parameters** tab (the **Resonance** family).
2. Select the **mu** group of parameters and from the context menu select **Delete group of parameters**, see Fig. 2.9.

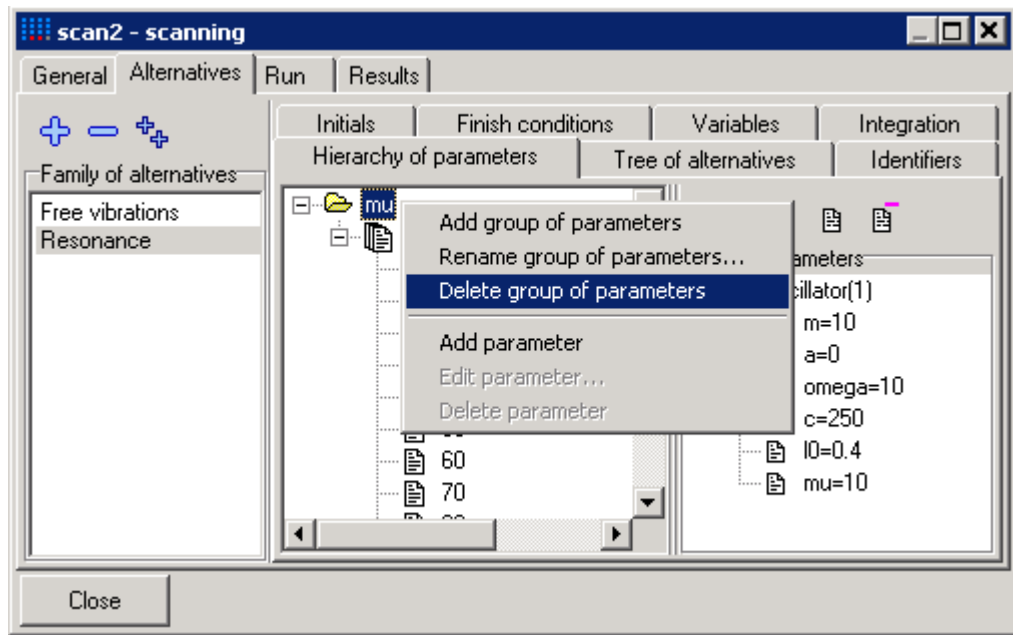


Figure 2.9. Deleting the group of parameters

3. Confirm the deleting the group of parameters.
4. Click on the **omega** parameter in the **List of parameters**. The **Parameters value** window appears.
5. Set the **Mode** to **Loop**, set the rest parameters as it is shown in the Fig. 2.10. Press **OK**.

We just described 16 iterations, from 0 up to 15 rad/s.

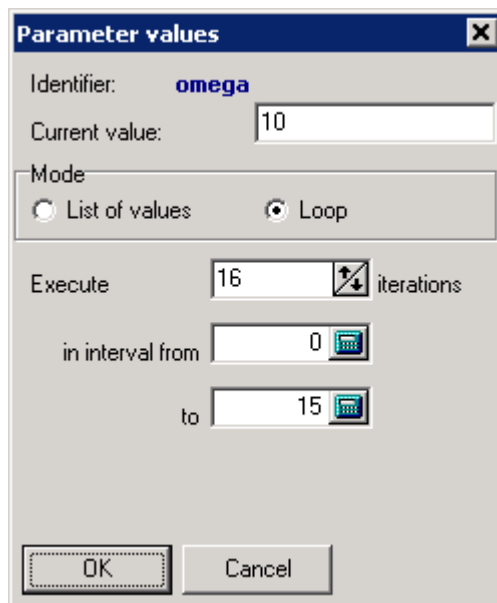


Figure 2.10. Frequency of the exciting force

Hierarchy of parameters for the **Resonance** family now should look like it is shown in the Fig. 2.11.

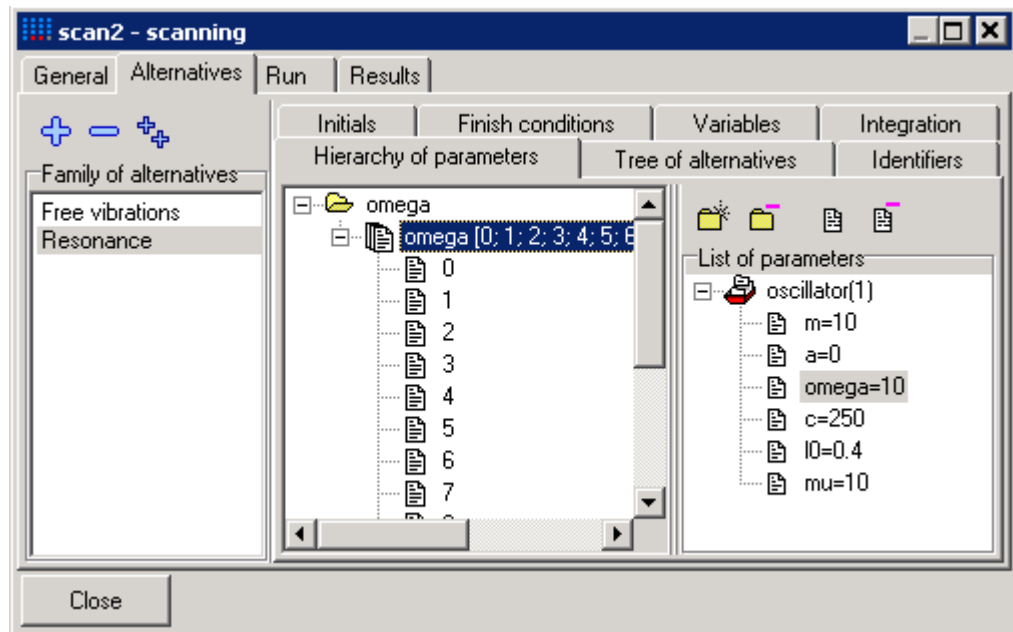


Figure 2.11. Hierarchy of parameters for the **Resonance** family

Parameters of the model

1. Select the **Identifiers** tab.
2. Set **a = 0.05** and **mu = 0**.

2.2. Running the project

1. Select the **Run** tab.
2. If you did everything in order then the **Report** contains the «**Error not found**», see Fig. 2.12.

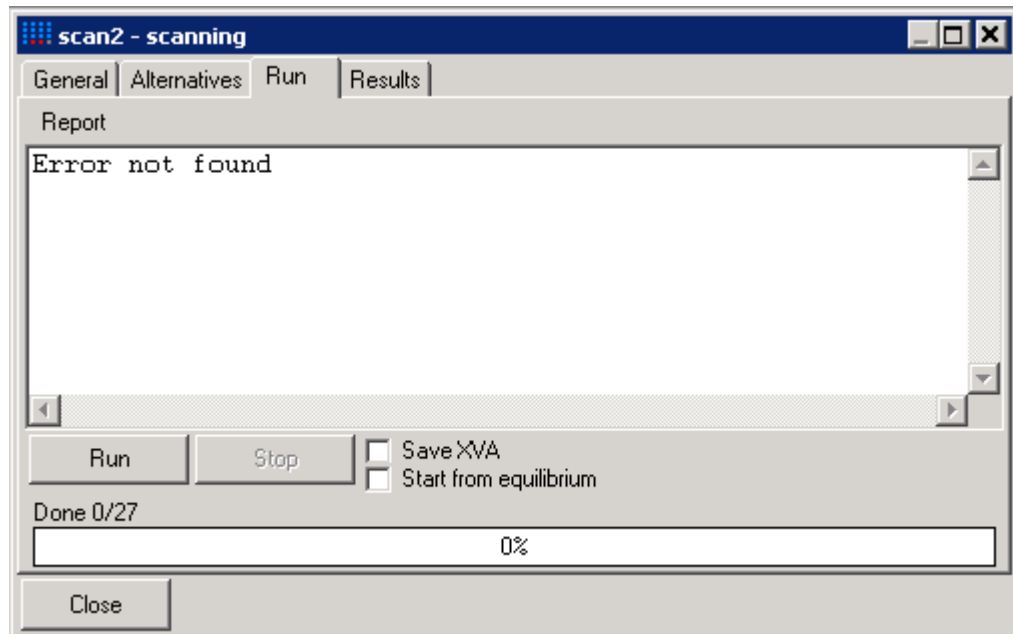


Figure 2.12.

3. Click the **Run** button.

Fulfillment of the numerical experiments starts. The **Report** shows identifier values and time efforts for every experiment, see Fig. 2.13.

The model is quite simple and it takes about a second for every numerical experiment.

4. You can see the “**Calculation of project of scanning is over**” message after project fulfillment.

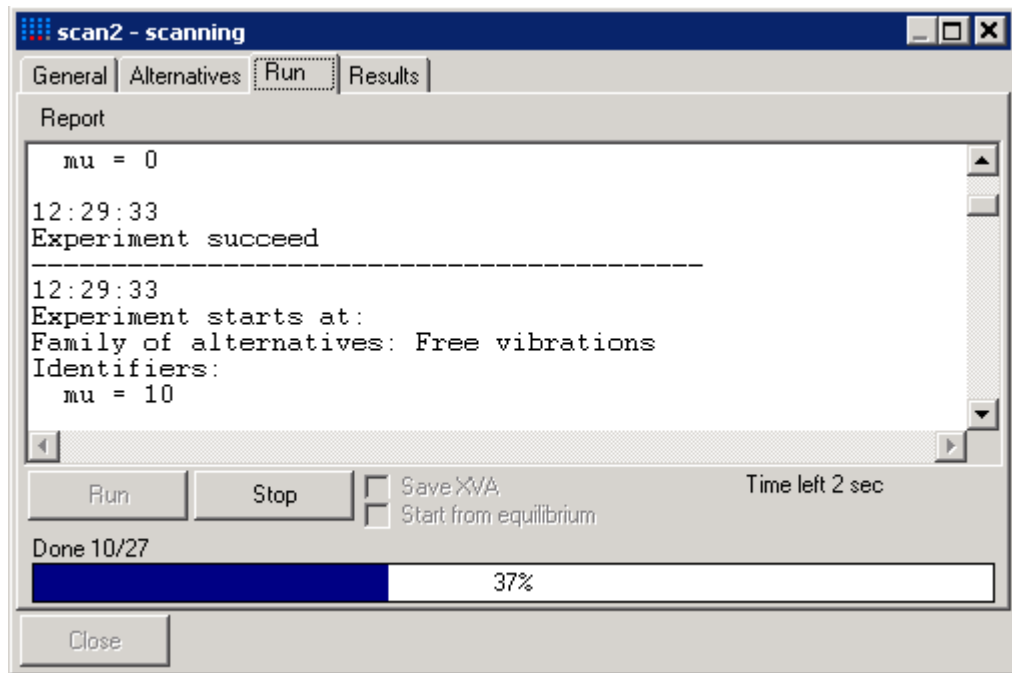


Figure 2.13. Running the project

2.3. Analyzing obtained results

2.3.1. Results of separate experiments

All numerical experiments are done now and we come to its analysis.

Free oscillation

1. Select the **Results / Free vibrations** tab.

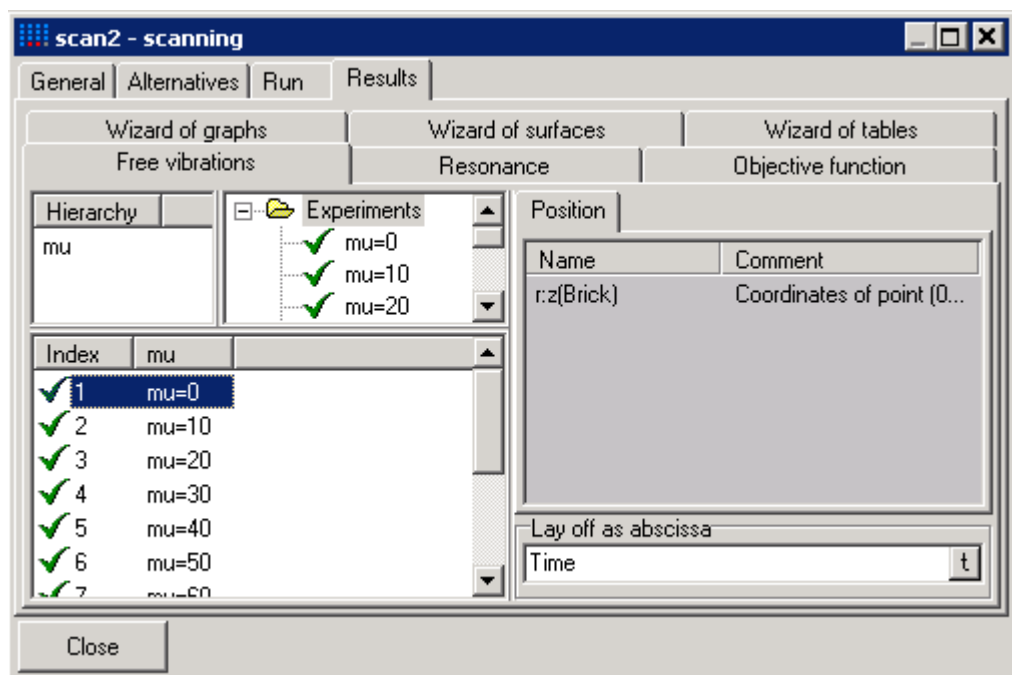


Figure 2.14. Results of the project

Let us compare the plots of the vertical position of the **Brick** at different values of damping coefficient.

2. Open new graphical window (**Tools/Graphical window**).
3. Select the project window and select all experiments in the left bottom side of the window (**Free vibrations** family), see Fig. 2.15. To select all experiments you can use the **Select all** context menu command or mouse + **Shift** or **Ctrl** buttons.

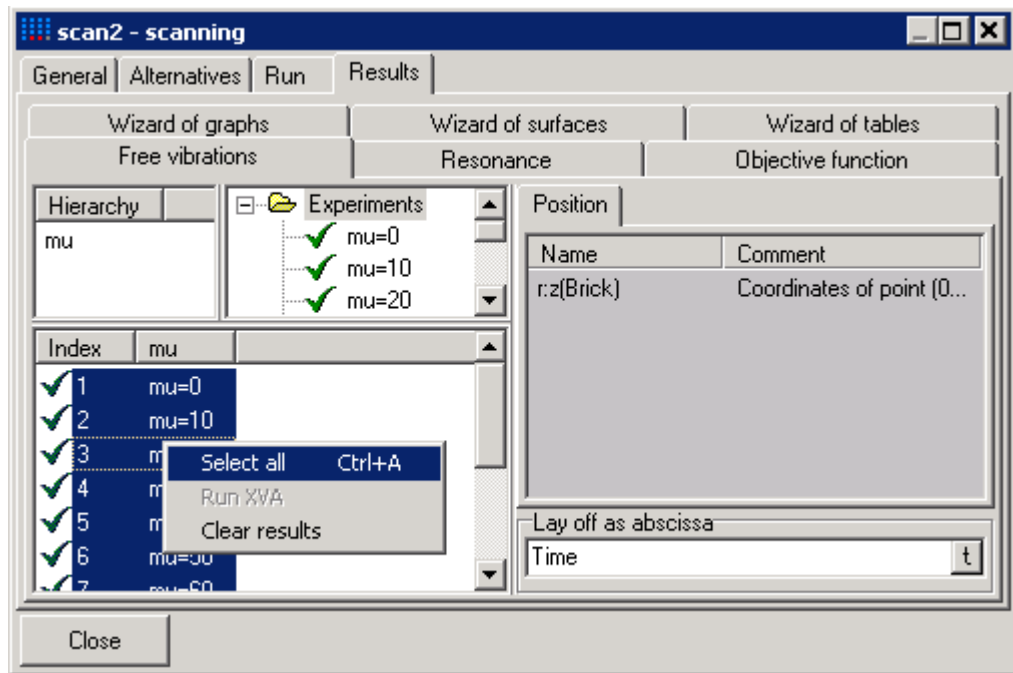


Figure 2.15.

- In the list of variables (the **Position** tab) select the **r:z(Brick)** variables and drag it to the graphical window. Plots of vertical position of the **Brick** are shown in the graphical window, see Fig. 2.16.

Note. You can show plots of calculated variables in graphical windows as well as analyze them in the **Table processor** and **Statistics** window (see the **Tools** menu).

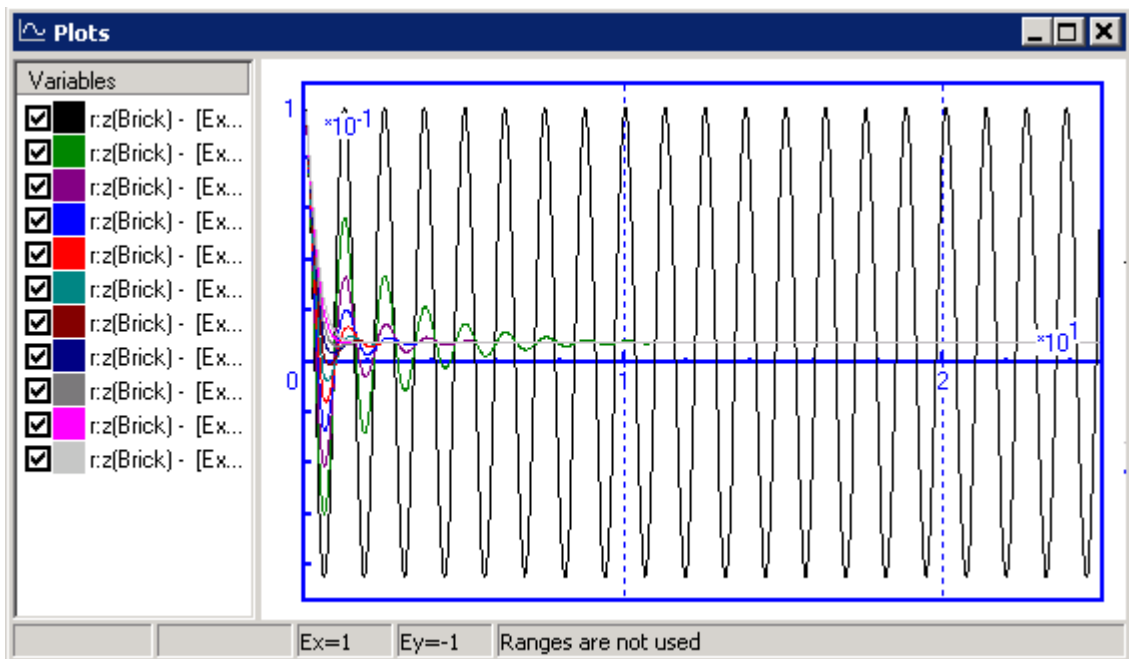


Figure 2.16. Vertical oscillation of the **Brick** at various damping coefficient

Forced oscillation

Now we come to the analysis of the case of forced oscillation.

1. Open new graphical window.
2. Point to the scanning project window and select the **Results / Resonance** tab.
3. Select all experiments and then drag the **r:z(Brick)** variable to the graphical window, see Fig. 2.17. The resonance case is clearly seen there.

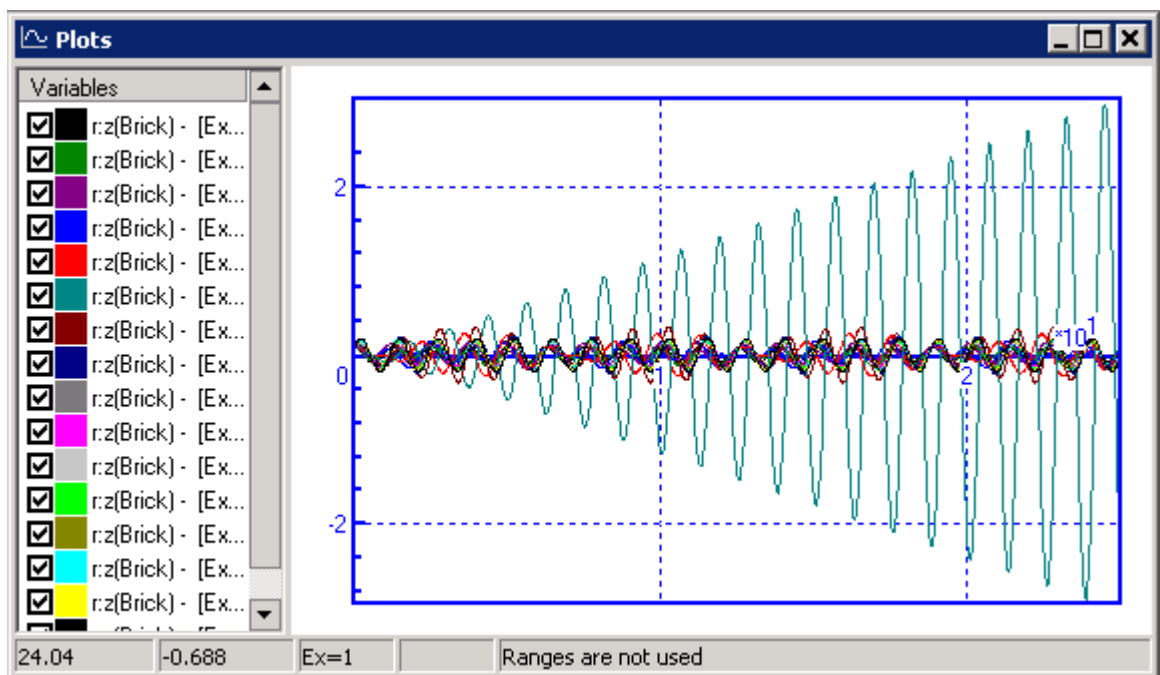


Figure 2.17. Forced oscillation

2.3.2. Summary graphs

There are several tools, which give you a possibility to get an overview of dynamics of a system without analyzing separate plots of every experiment. Here we mean wizards of graphs, surfaces and tables.

Let us create the graph of the *root mean square* (RMS) of the vertical oscillation depending on frequency of the exciting force.

1. Select the **Results / Wizard of graphs** tab.
2. In the **Family** list select **Resonance**.
3. In the list of variables select **r:z(Brick)**.
4. Set **Functional** to **RMS**.
5. Set **Parameter** to **omega**.

6. Click the  button on the top panel of the **Wizard of graphs**.

Summary graph is shown in the graphical field, see Fig. 2.18. Its analysis shows that the maximal RMS corresponds to **$\omega = 5 \text{ rad/s}$** . The reader, which is familiar with vibration theory, certainly notices that plot in the Fig. 2.18 is very similar to amplitude-frequency characteristic.

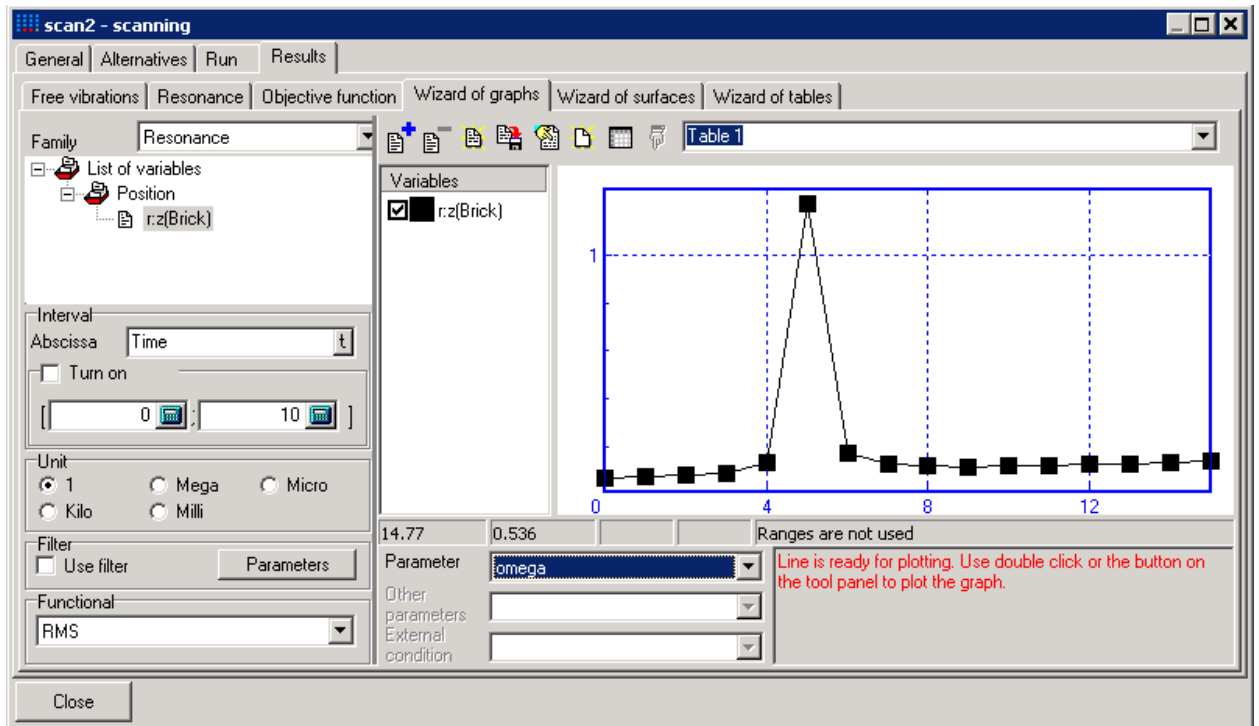


Figure 2.18. Wizard of graphs.

Abscissa: frequency of the exciting force.

Ordinate: RMS of vertical oscillation of the **Brick** body.

3. Scanning: implementation to railway vehicle dynamics

Creating, running and analyzing scanning projects in the field of railway vehicle dynamics are considered in the **gs_UM_Loco.pdf** file of UM User's Manual. Example of determining the critical velocity of AC4 railcar is shown.

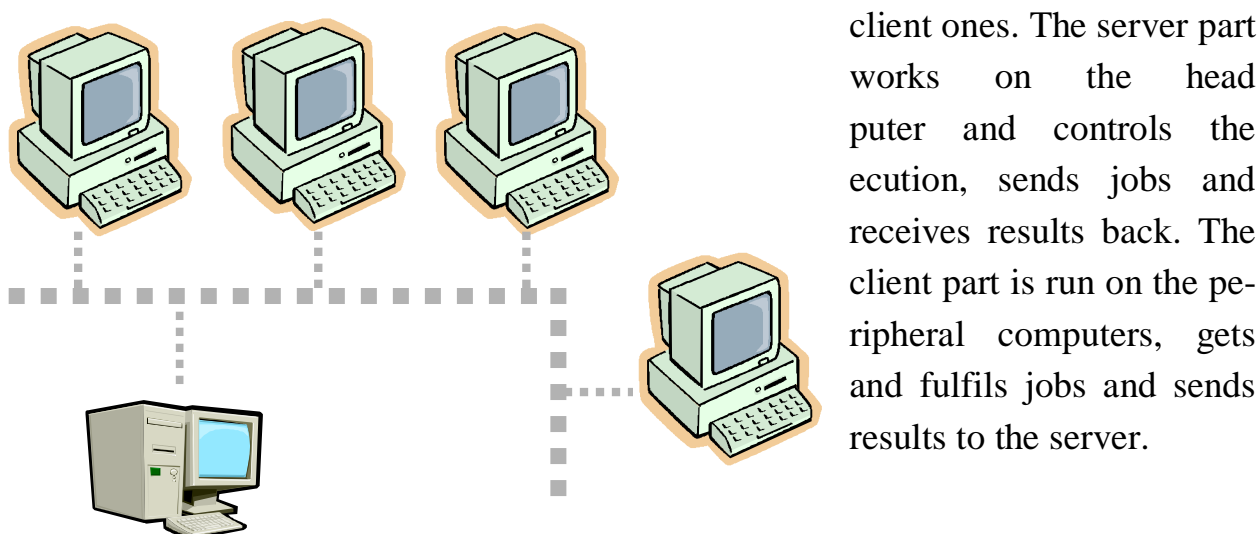
That manual you can find in the **{um_root}\manual** directory or download using the following link:

http://www.umlabor.ru/download/50/eng/gs_UM_Loco.pdf

4. Using the service of distributed calculations

The service of distributed calculations is an extra tool within **UM Experiments** module. It give the user a possibility to use all computing power of a local network for parallel running of the experiments of the scanning project. The service is used for speeding-up time-consuming projects¹.

The service is based on TCP/IP protocol and allows using any computer reachable by TCP/IP² under Windows 98 / NT / 2000 / XP for parallel numerical experiments. The service of distributed calculations consists of two parts: server and



¹ Projects might take you days, weeks and even months and years of CPU time using one computer. Service of distributed calculations helps you to finish such projects much faster.

² Any reachable computer in your local network, intranet or Internet.

4.1. Preparing for running

At first, you need to install the client part of the service to the computers, which you are going to use in your computing cluster. Installation is the **um50client.exe** file. You can find it on your CD or download from Internet using the following link: <http://www.umlab.ru/download/50/eng/um50client.exe>.

Install **um50client.exe** on the computers, which probably will be used by service of distributed calculations¹.

¹ Including your own computer – server of distributed calculations.

4.2. Loading scanning project

We will not describe a new scanning project in this section and just use the project, which was prepared in the previous section, see Sect. 2 of this manual.

If you did not prepare that project then follow instructions, which are given in the Sect. 2, or download the project using the following link:

<http://www.umlab.ru/download/50/scan2.zip>.

1. Run the **UM Simulation** program.
2. To load the **scan2** scanning project from the **Advanced analysis** menu select the **Scanning: open project**. Select the **scan2** project in the open dialog and click **OK**, see Fig. 4.1. The window of the scanning project appears.

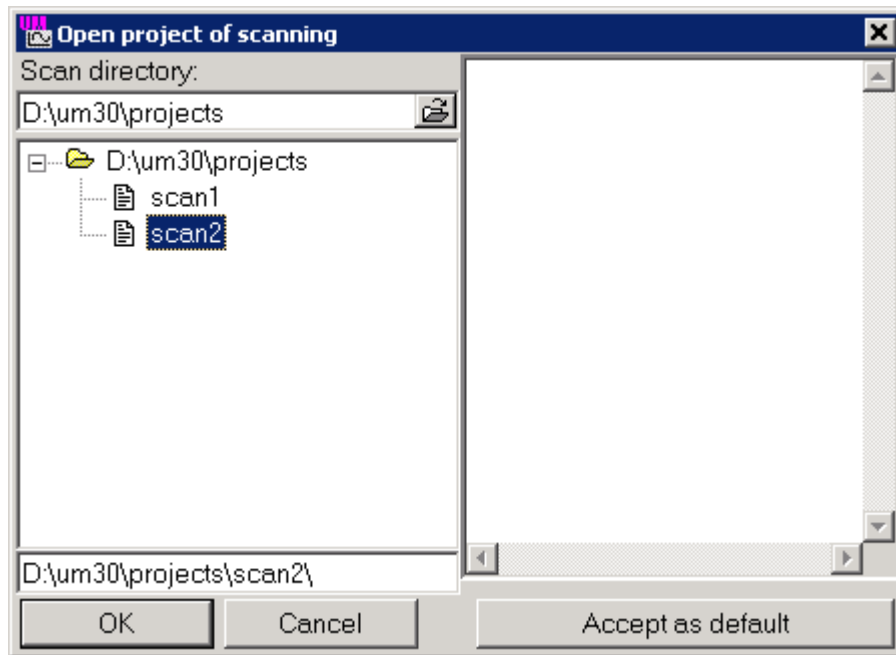


Figure 4.1. Open dialog

3. In the project's window select the **General** tab.
4. If **Status** is **Completely done** or **Partially done** then you need to delete previous results. To do that use the **X** button and then **Status** is set to **No calculations done**.

4.3. Running

1. Run the client part of the service of distributed calculations (the **cclient.exe** file) on the computers.

Note. Some software (firewalls, anti-virus and other) might consider network activity of client and server parts as potentially dangerous and block its further activity. In such a case permit the execution of UM software.

2. Come back to your computer.
3. Turn on the **Distributed calculations** check box, the **Run** tab.
4. Click the **Run** button. The server of distributed calculations appears, see Fig. 4.2.

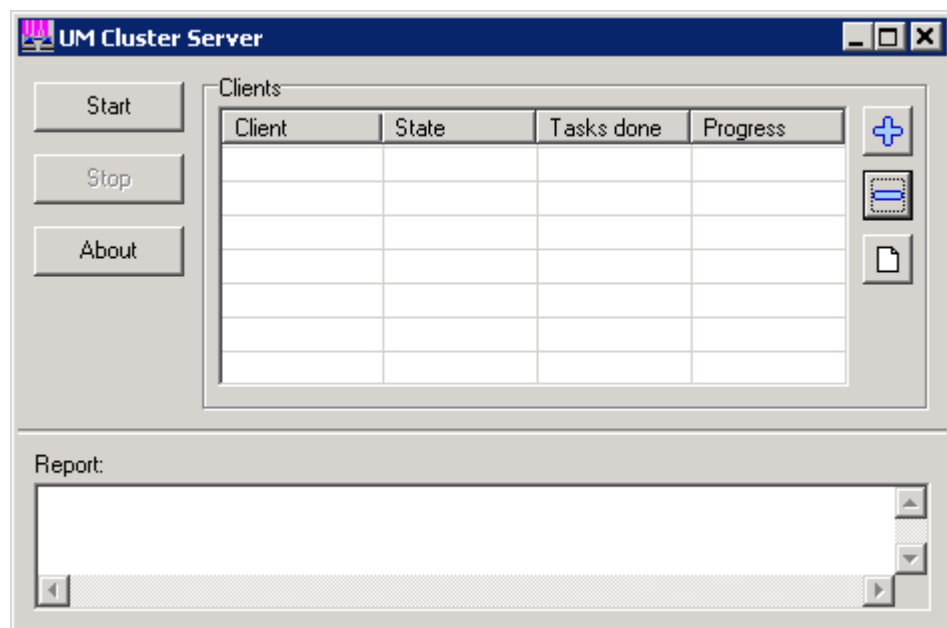
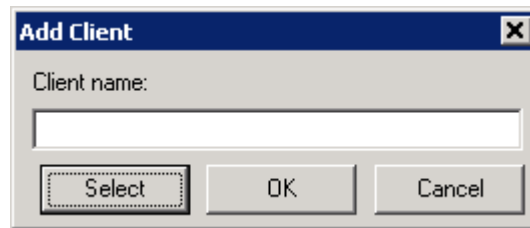


Figure 4.2. UM cluster server

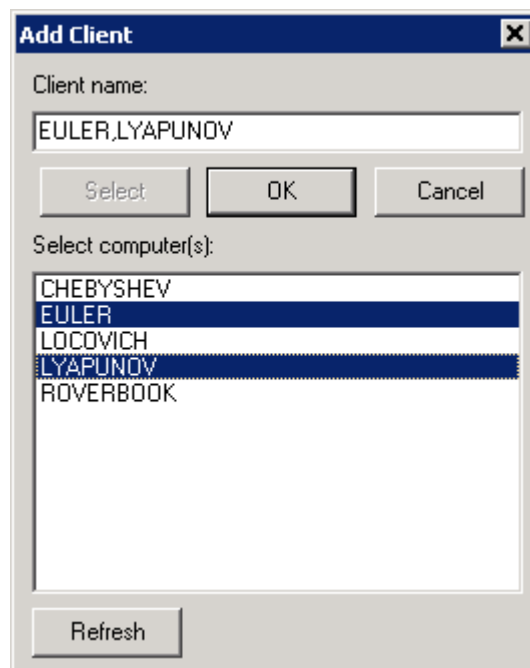
Server of distributed calculations

Firstly you need to add client computers to the **Clients** list¹.

1. Click the  button. **Add Client** dialog box appears.

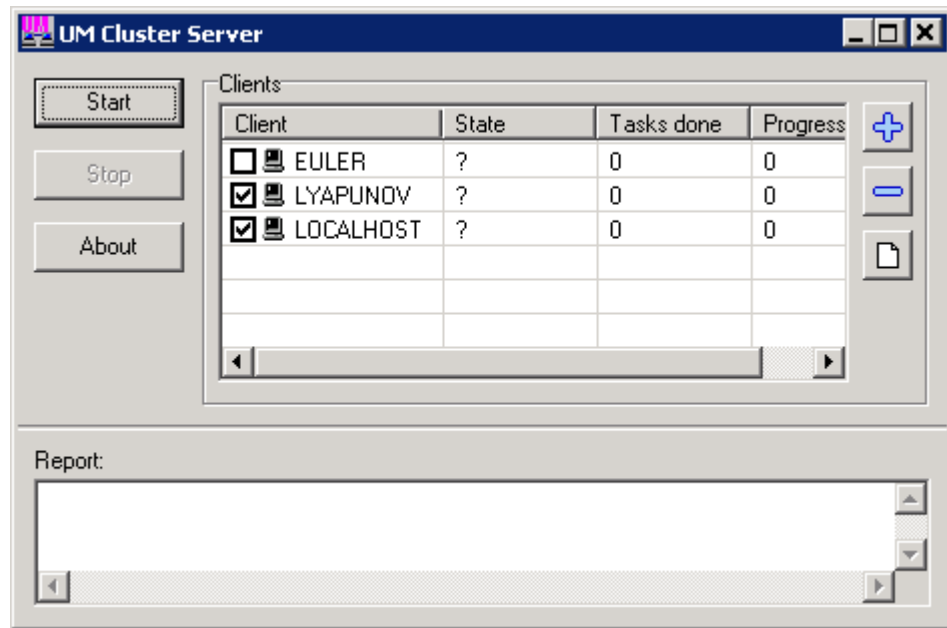


2. Click the **Select** button. The list of the computers in the local network appears in the bottom of the window. Scanning of the network might take some time.



3. Select the necessary computers with the help of **Shift** and/or **Ctrl** button and click **OK**. Selected computers appear in the list if client computers is in the UM cluster server.
4. Turn on check boxes in the **Clients** list for computers that you are going to include in the computing cluster.

¹ In the terms of parallel computing, cluster is an aggregate of computers that are used for common solution of the problem. The main computer and the program which distributes and controls common work of computers are usually named server of a computing cluster. The computer and corresponding program that works under the control of the server are called client of a computing cluster.



Note. You can include to the computing cluster any computer from intranet or Internet. In this case you need to input its IP-address in the **Client name** in the **Add Client** dialog box and click **OK**.

Note. You can use the server computer simultaneously as a client one. To add your local computer as one of the clients type **LOCALHOST** in the **Client name** box.

Running calculations

1. Click the **Start** button.

Execution of the scanning project using computing cluster starts. You can add new client computers or disconnect them. To disconnect the client after the current numerical experiment finishes use the **Disable** command from the context menu, to disconnect the client immediately with breaking the current experiment use the **Stop** command.

2. The «===== **complete** =====» message appears when all of the numerical experiments are done. Then you can close the window of the server.

Use the **Stop** button to interrupt execution of the scanning project. Results of all numerical experiments that are in progress will be lost; all already calculated data will be saved. You can continue the execution of the project with or without computing cluster.

5. Optimization project: creating and analysis

As an example of implementation of optimization methods let us consider the simple problem – balancing the rotor (100 kg) with the help of a point mass (1 kg), see Fig. 5.1.

The rotor (body 1 in Fig. 5.1) revolves on an axis that does not coincide with rotor axis of symmetry. Position of the axis is expressed via two parameters: $d_y=d_z=0.01$ m. Radius of the rotor is $R=0.2$ m. Position of the point mass (body 2 in Fig. 5.1) relative to rotor frame of reference is expressed with the help of d_{ey} and d_{ez} parameters that are Y and Z coordinates of the point mass correspondently.

Goal of optimization is minimization of dynamical reaction forces in the revolution joint that are raise by unbalanced rotor with the help of the point mass. Reaction forces in the balanced ($d_{ey}=d_{ez}=0.11$) and unbalanced ($d_{ey}=d_{ez}=0$) rotor are given in Fig. 5.2.

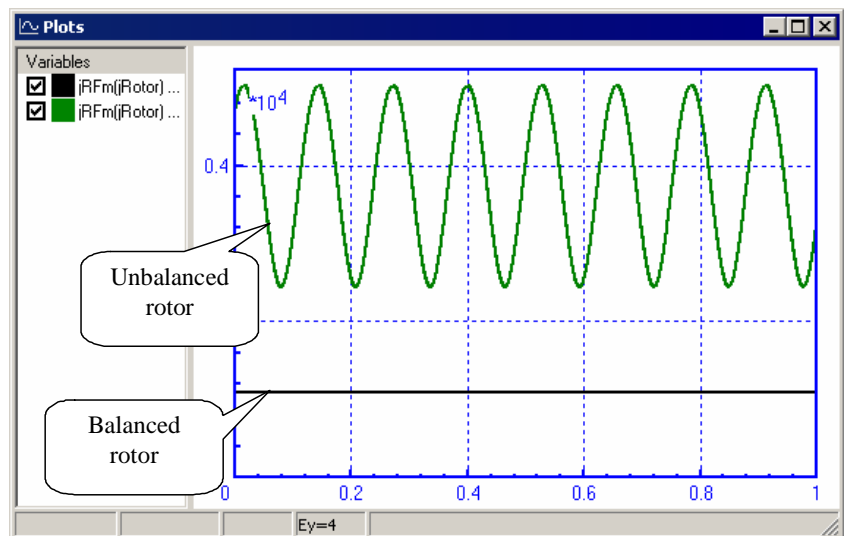
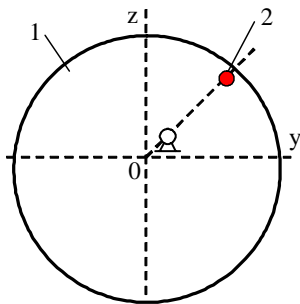


Figure 5.1. Rotor balancing

Figure 5.2. Reaction forces in the balanced and unbalanced rotor

5.1. Preparing the optimization project

5.1.1. Creating new project

1. Run **UM Simulation** program.
2. From **Advanced analysis** select **Optimization: new project** to create new optimization project. New dialog window appears.
3. Input the full path to the new project including the name of the project, see Fig. 5.3.
4. Click **Create** button. New window of optimization project appears, see Fig. 2.3.

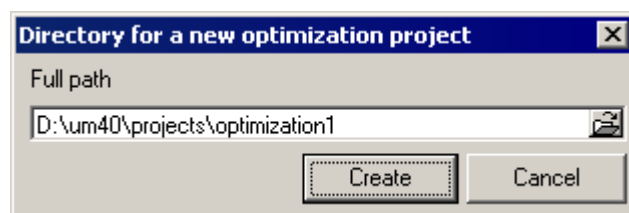


Figure 5.3. Path to the new optimization project

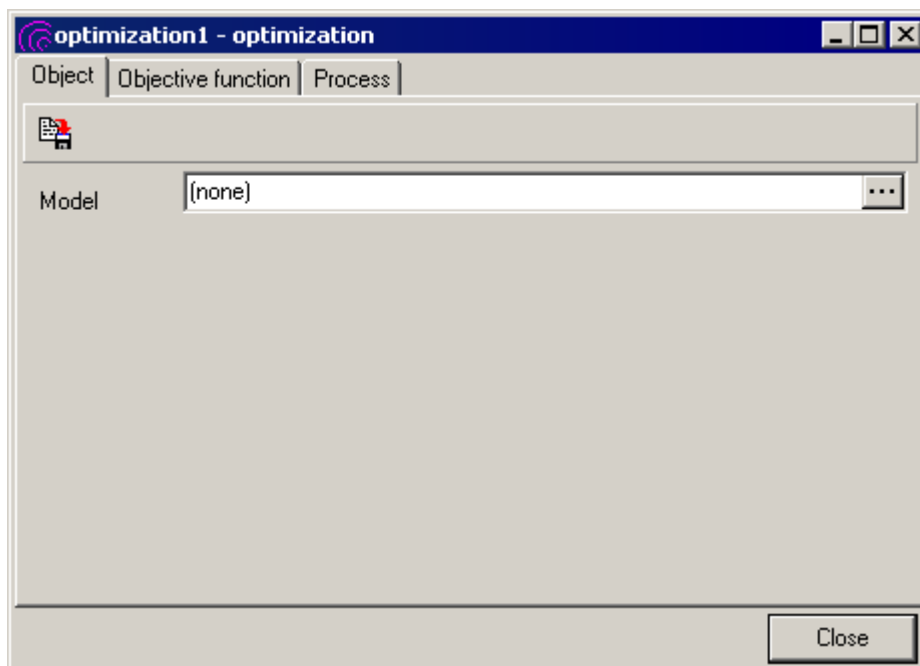



Figure 5.4. Optimization project

5.1.2. Loading a model

1. In the **Model** box click the  button to load a model to optimize. Open dialog appears.
2. Select the `{um_root}\samples\tutorial\excentric` model. Model will be loaded and its parameters are show in the right part of the window, see Fig. 5.5.

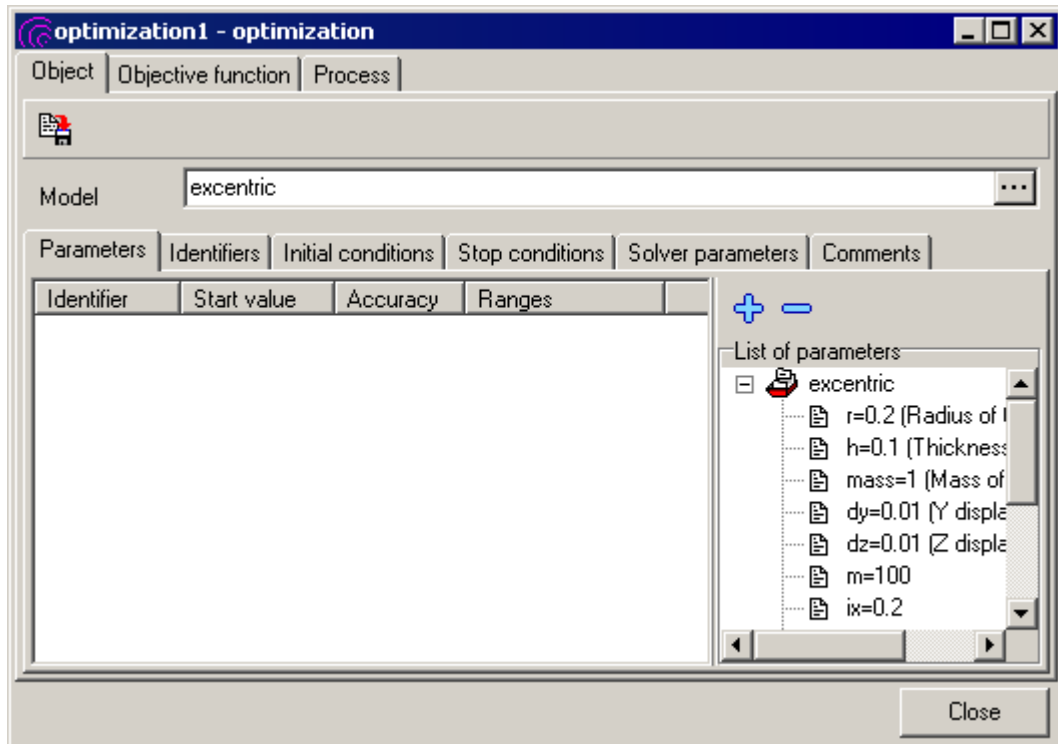


Figure 5.5. Optimization project: loading a model

5.1.3. Parameters to optimize

To solve the optimization problem we need to balance the rotor varying **dey** and **dez** parameters. Now we will add these parameters to the optimization project and set desired accuracy and boundaries for them.

1. Click the left mouse button on the **dey** parameter in the list of parameters. New dialog window appears, see Fig. 5.6.
2. Set **Accuracy** to **0.001**. Thus we will find optimum accurate to 1 mm.
3. Set **Low boundary** to **value** and input **-0.2** into numerical box.
4. Set **High boundary** to **value** and input **0.2**.
5. Click the **OK** button to close the dialog.
6. Add the **dez** parameter in the same way and input the same settings for it.

After all your optimization project looks like Fig. 5.7.

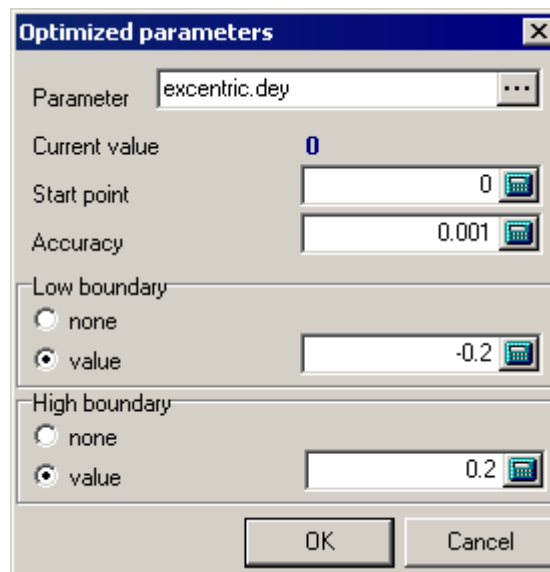


Figure 5.6. Adding new parameter

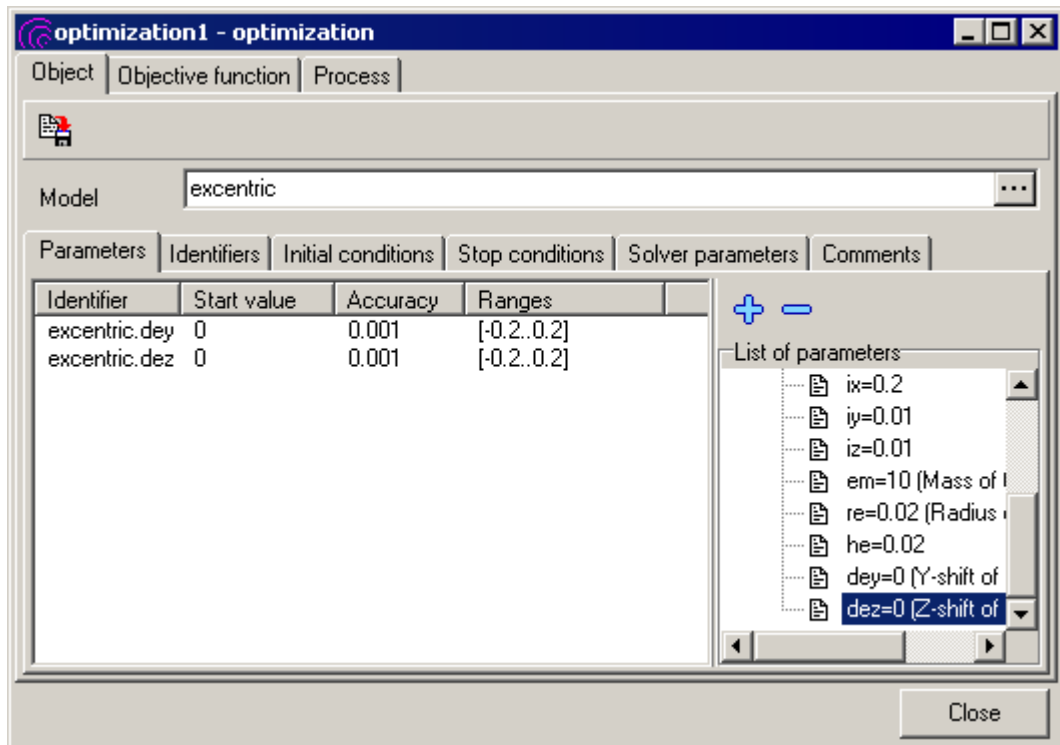


Figure 5.7. Parameters to optimize

5.1.4. Initial conditions

Now we need to set arbitrary but big enough initial velocity of the rotor with which the rotor will be able to revolve, for example 50 rad/s. Please note that having small initial velocity will lead to oscillations of the rotor like a pendulum.

1. Select the **Initial conditions** tab. Input **50** as initial velocity of the rotor, see Fig. 5.8.

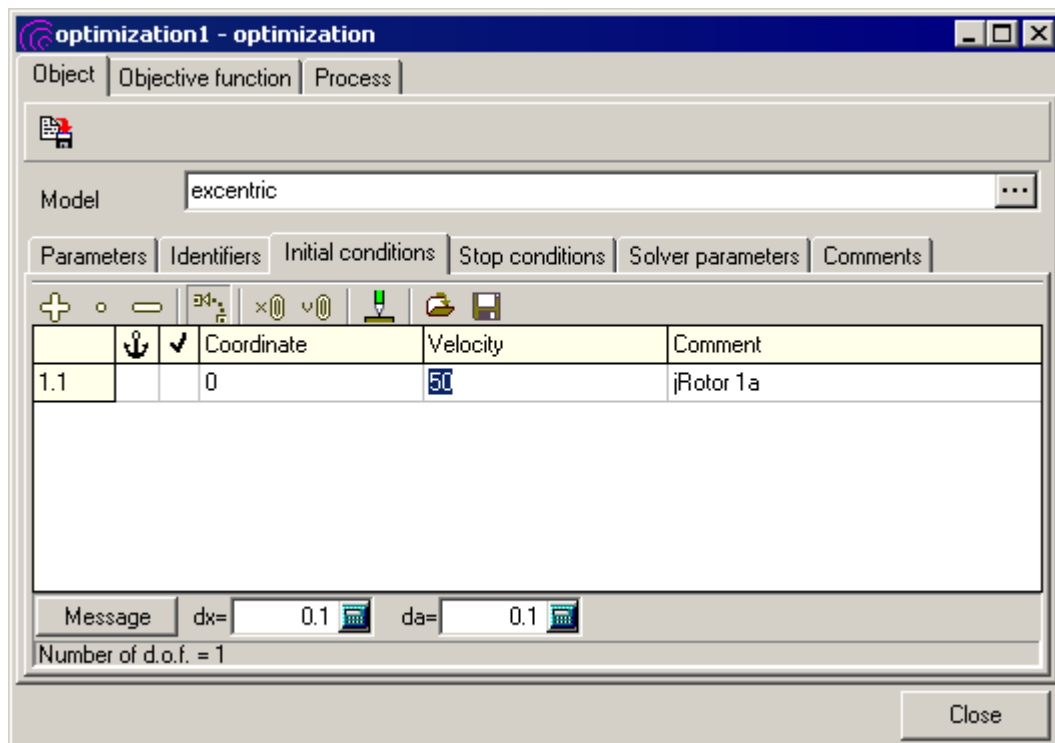


Figure 5.8. Optimization project: initial conditions

5.1.5. Stop conditions

Obviously for periodical processes it is enough to consider dynamical properties of the system during only one period. At that it is necessary to set **Step size for animation and data storage** so as it would be significantly less than one period. Otherwise it might be too few points for adequate analysis in time histories that are used for calculation of the objective function that might lead to considerable distortion of results of optimization.

Taking into account what we said above we will use the stop condition as follows:

$$\varphi > 2 * \pi,$$

where φ is turning angle of the rotor in radian.

1. Select the **Stop conditions** tab.
2. Open new **Wizard of variables** (**Tools / Wizard of variables** menu item). Create new variable corresponding to the 1.1 coordinate as it is shown in Fig. 5.9 and drag this variable to the corresponding box in the optimization project, see Fig. 5.9.
3. Input **6.283** in the numerical box.

Now we have finished describing stop conditions.

Note Turning angle of the rotor the only degree of freedom in this model. In this case we could also use time as stop condition – initial velocity is 50 rad/s gives period approximately of 0.126 s – *Time* > 0.126. But such an assumption might lead to serious errors as since angular velocity of the rotor is not a constant and is also depended on position of the point mass.

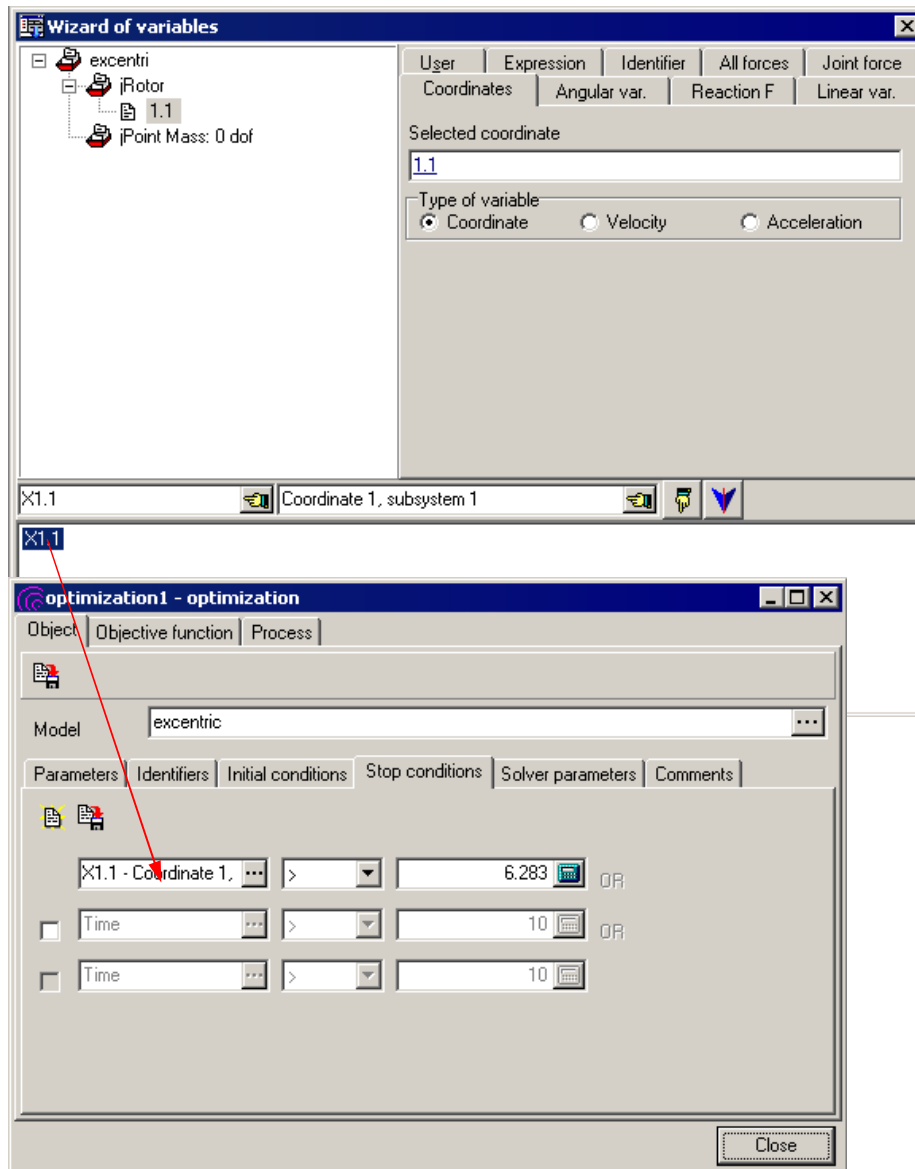


Figure 5.9. Optimization project: stop conditions

5.1.6. Solver parameters

1. Select the **Solver parameters** tab.
2. Set **Step size for animation and data storage** to **0.001** s, see Fig. 5.10. Thus we will have approximately 125 point per period that is quite enough.

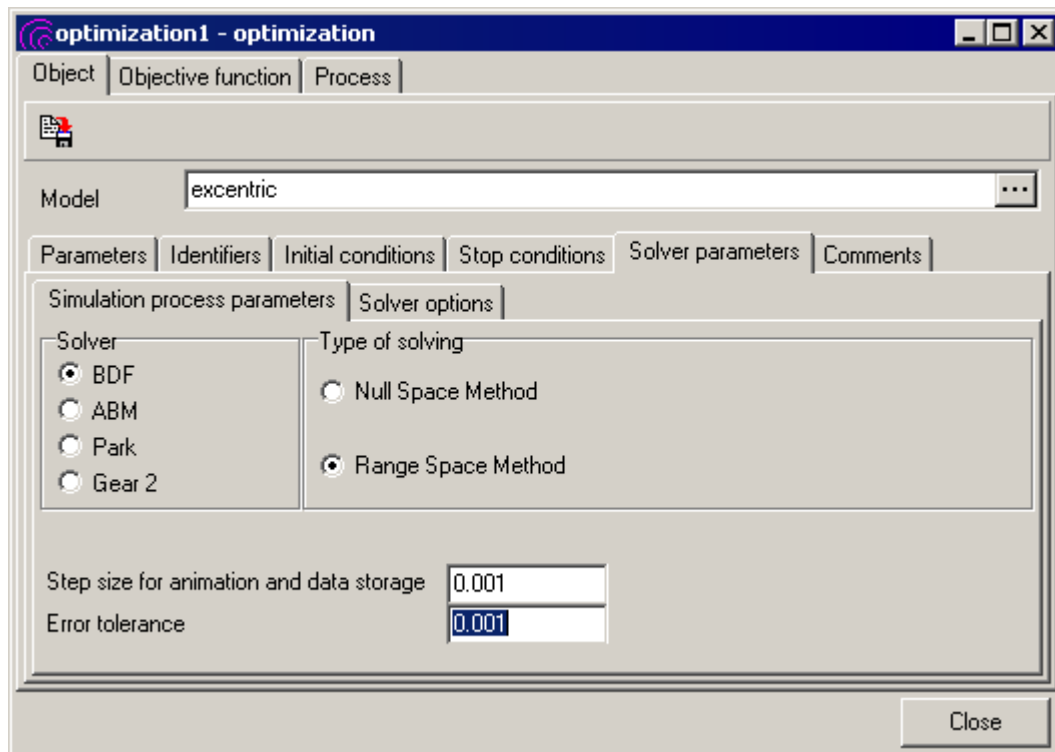


Figure 5.10. Optimization project: solver parameters

Now preparing the model is finished and we will come to describing an objective function.


5.2. Objective function

To describe the objective function within Universal Mechanism the analytic hierarchy process is used. The analytic hierarchy process was developed by Saaty¹. The method is based on principle of hierarchization, where the main, most common goal consists of several more detailed sub-goals, each sub-goal of the first level consists of the corresponding sub-goals of level two and so on. Every sub-goal has only one upper goal. Different sub-goals affect the upper goal with a different weights.

We will not create a multilevel hierarchy and will use the only criterion. Now we will describe the objective function so as to minimize maximal value of the reaction force in the joint – so called minimax problem.

1. Select the **Objective function** tab.

By default it contains the only criterion – *Optimal technical decision*, see Fig. 5.11. Let us rename it to **Reaction force**.

2. Select the **Optimal technical decision** criterion in the tree of criteria and select **Rename** from the context menu, see Fig. 5.11. Input **Reaction force** and press **Enter**.
3. Click the  button to check integrity of the objective function. Now we have 2 errors, see 5.12.

¹ Saaty, T. The Analytic Hierarchy Process, McGraw-Hill, 1980.

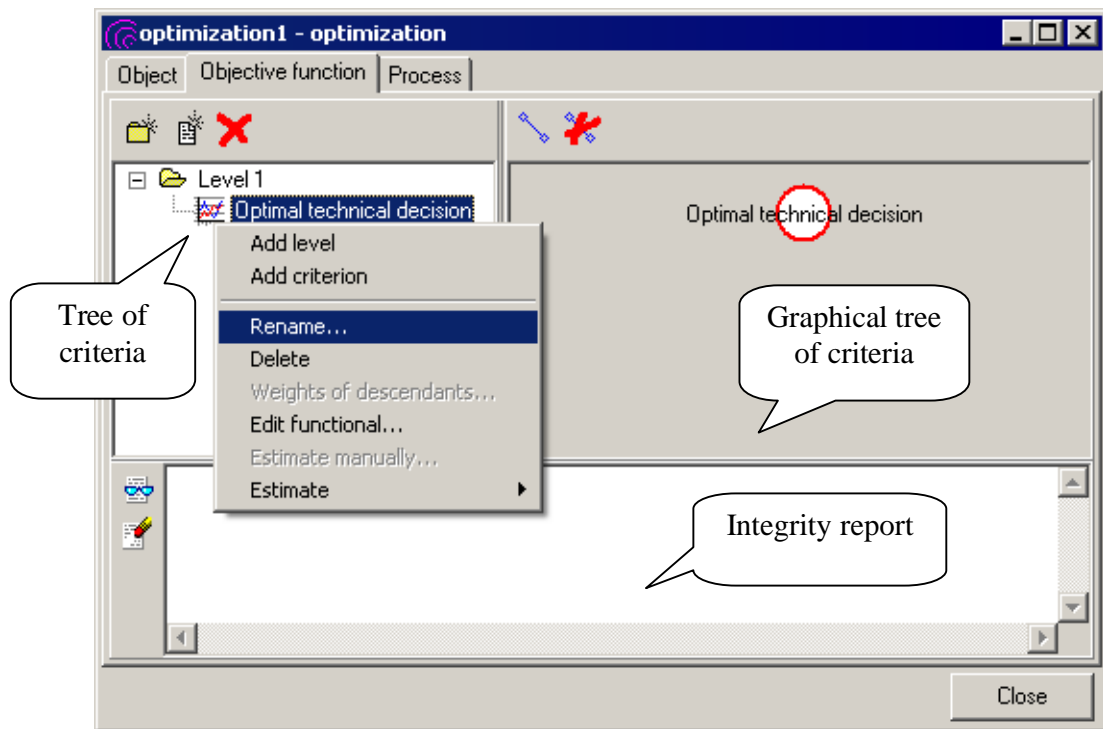


Figure 5.11. Objective function

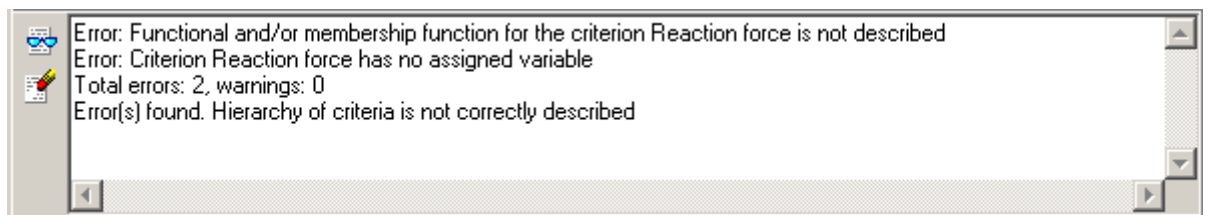


Figure 5.12. Integrity checking

Now we will create a variable corresponding to a reaction force in the joint with the help of the **Wizard of variables** and assign this variable to the criterion.

4. Open **Wizard of variables** and create the variable for the reaction force and drag it to the **Reaction force** criterion, see Fig. 5.13.

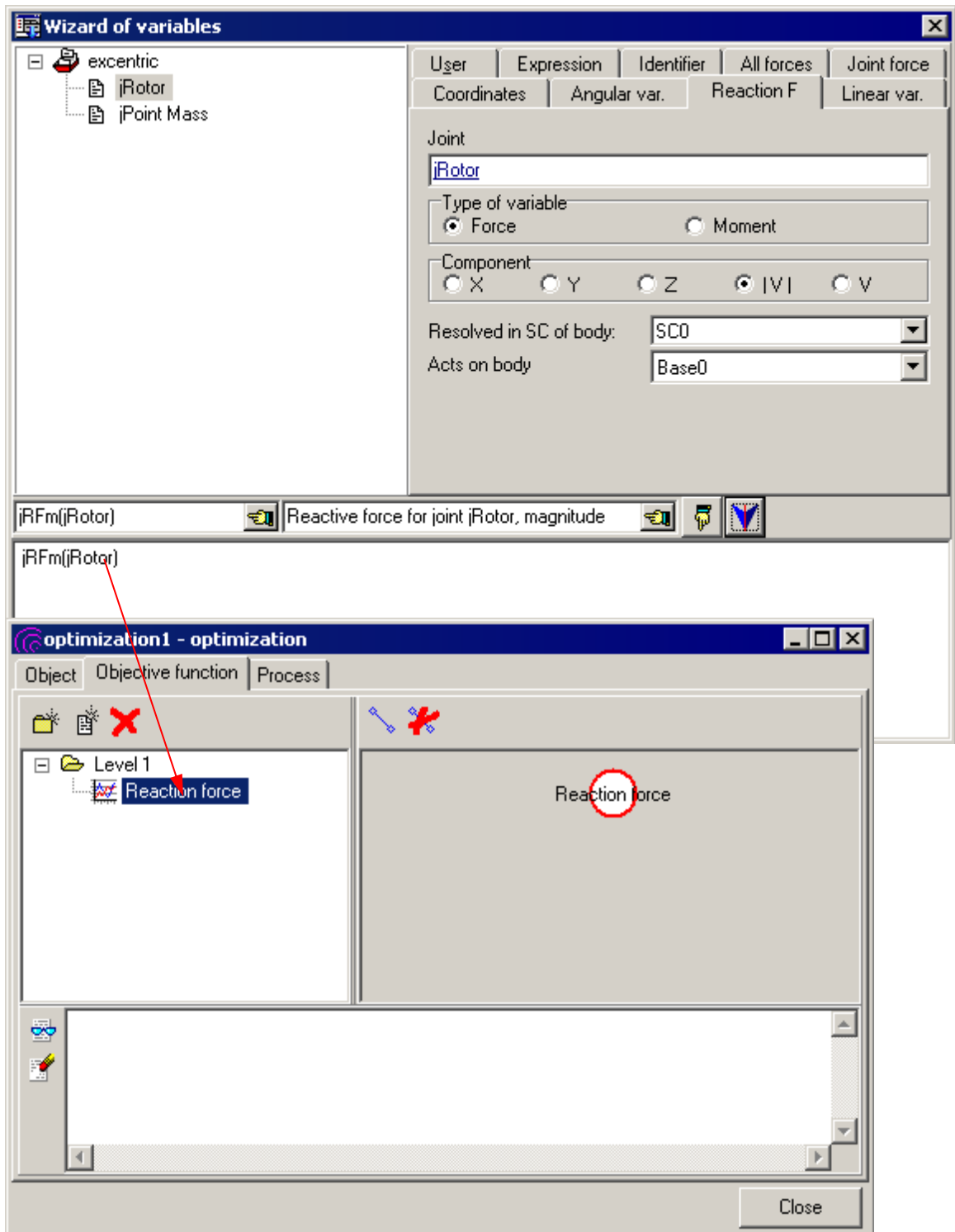


Figure 5.13. Assigning variable

We have just described the variable that will be used for calculation of the objective function, now we will describe how to calculate the objective function using this variable.

5. Double click left mouse button on the **Reaction force** criterion on the tree of criteria or select the **Edit functional** in the context menu. New dialog window appears, see Fig. 5.14.
6. Click the **+** button to add a new functional.

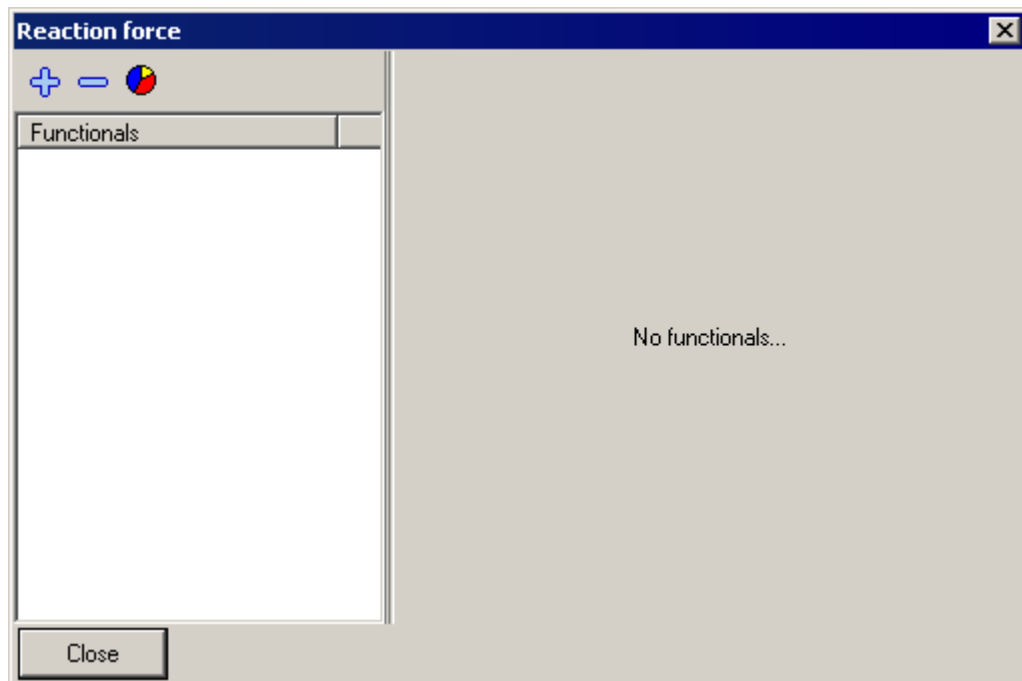


Figure 5.14. Functionals

7. Set **Functional** to **MaxAbs** (maximum in absolute value), see Fig. 5.15.
8. Set **Estimation is based on** to **value of functional**.
9. Turn off the **Use boundaries of values of functional** flag.
10. Turn on the **Minimize functional**, otherwise by default the maximization problem is solved.
11. Click **Close** to close the window.

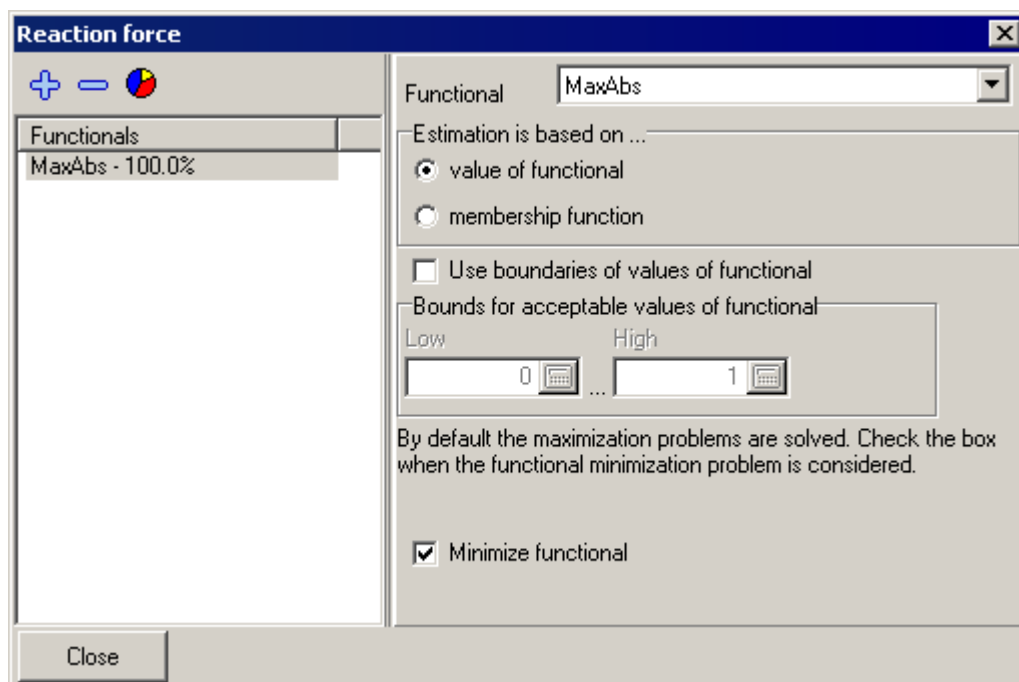



Figure 5.15. Functional

Now we have completely described the objective function, you can check integrity clicking  button and now we come to execution of the optimization project.

5.3. Running the project

1. Select the **Process** tab.
2. Set **Methods** to **Hook-Jeeves**.
3. Click the **Run** button. Execution of the optimization project starts. It might take several minutes to finish.

During the execution information about current numerical experiment is printed in the **Report** box. On completion you can see following information in the **Report**: total number of numerical experiments, optimal parameters and corresponding value of the objective function.

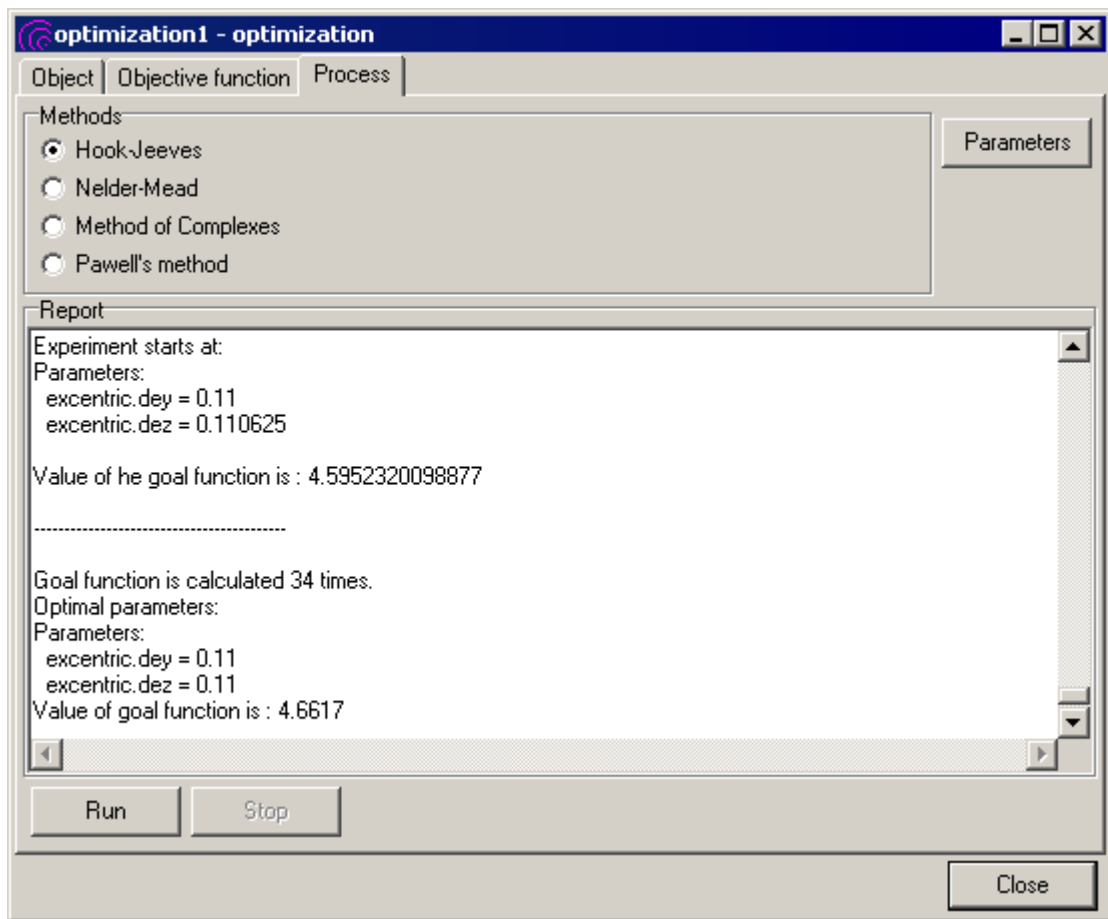


Figure 5.16. Running the project

In this test example the optimal solution was found after 34 numerical experiments. Please note that the optimal values are found with absolute accuracy. This is obviously a chance because analytical solution (0.11) divisible by accuracy (0.001).