

Optimax USER'S MANUAL v0.6.3

Matej Borovinšek

January, 2017

1 Introduction

This is a description of the Optimax software. The Optimax software allows the user to structure the design process, explore the design space and compute optimal designs according to specified constraints and objectives.

Using Optimax a design process or a workflow is built from different building blocks. The workflow is then executed for different input parameters to obtain the results.

2 Building blocks

Optimax software is created from the following building blocks:

- *Scalar variables,*
- *Arrays,*
- *File resources,*
- *Components,*
- *Drivers* and
- *Recorders.*

2.1 Scalar variables

Scalar variable blocks (Figure 2.1) are used to store scalar values. *Scalar variables* are divided to the following categories:

- *Input variable* – a scalar variable that is modified by the *driver*;
- *Constant variable* – a scalar variable that remains constant;
- *Variable output* – a scalar variable that is read from the *file resource*;
- *Variable function* – a scalar variable that is evaluated from other *scalar variables* by an equation;
- *Variable from array* – a scalar variable that is read or evaluated from an *array*;
- *Variable from compare curves* – a scalar variable that represents the difference between two curves;
- *Variables container* – a container for *input variables*, *constant variables* and *variable functions*.

x ▾	
$x \rightarrow$	Input variable
x	Constant variable
$\rightarrow x$	Variable output
$f(x)$	Variable function
$\vec{x} \rightarrow x$	Variable from array
$\vec{x} \rightarrow \vec{x}$	Variable from compare curves
$x_i \rightarrow$	Variables container

• Figure 2.1: *Scalar variables*

2.2 Arrays

Array blocks (Figure 2.2) are used to store array values. *Arrays* are divided to the following categories:

- *Constant array* – an array that remains constant;
- *Output array* – an array that is read from the *file resource*;
- *Array function* – an array that is evaluated from other *arrays* by an equation;
- *Array operation* – an array that is the result of an operation evaluated on another *array*.

\vec{x} ▾	
\vec{x}	Constant array
$\rightarrow \vec{x}$	Output array
$f(\vec{x})$	Array function
$\emptyset(\vec{x})$	Array operation

Figure 2.2: *Arrays*

2.3 File resources

File resource blocks are used to store files. A file resource can be of one of the following types:

- *Input file* – a file that is used by a *component* block as an input file;
- *Constant file* – a file that remains unchanged;
- *Output file* – a file that is the result of the *component* block execution;
- *Copy* – a file that is a copy of another *file resource*.

2.4 Components

Component blocks (Figure 2.3) are used to represent the executable parts of the workflow. A *component* should take an input file and produce the result file after execution. Supported components are:

- *Dos component* – a component that enables the execution of any command prompt command;
- *Abaqus component* – a component that enables the execution of an Abaqus analysis;
- *LS-Dyna component* – a component that enables the execution of a LS-Dyna analysis;
- *Qsub component* – a component that enables the execution of a script file using Sun Grid Engine’s “qsub” command over SSH protocol;
- *Excel component* – a component that enables the execution of computations inside the Excel worksheet.

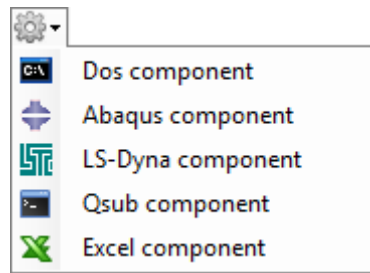


Figure 2.3: *Components*

2.5 Drivers

Driver block (Figure 2.4) determines how the workflow is executed. There can only be one driver component in a project. Supported drivers are:

- *Doe: Design of experiments* – a driver that performs the Design of experiments procedure;
- *NM: Nelder-Mead simplex method* – a driver that performs non-gradient single-objective optimization based on Nelder-Mead simplex method;
- *GA: Genetic algorithm*– a driver that performs non-gradient single-objective optimization based on Genetic algorithm;
- *AMQ: Convex approximation method* – a driver that performs constrained gradient based single-objective optimization based on Convex approximation method;
- *LM: Levenberg-Marquard method* – a driver that performs gradient based curve fitting based on Levenberg-Marquard method;
- *NSGAI* – a driver that performs non-gradient multi-objective optimization based on Non-dominated sorting genetic algorithm.

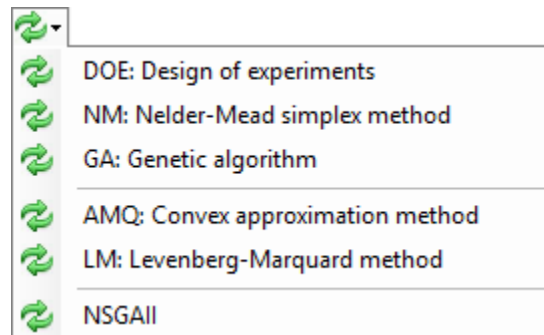


Figure 2.4: *Drivers*

2.6 Recorders

Recorder blocks (Figure 2.5) are used to record data. Supported recorders are:

- *File recorder* – a recorder that records the data as files;
- *Csv recorder* – a recorder that records the data as values and stores them as Csv (Comma Separated Values) file;
- *Memory recorder* – a recorder that records the data as values and displays them in the graphical user interface;

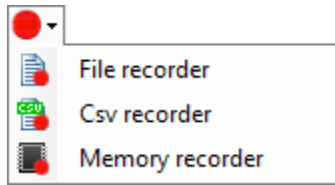


Figure 2.5: *Recorders*

3 Tutorial

3.1 Inverse determination of material properties using CalculiX FE code

Determine the material parameters of the cubical steel sample (100 x 100 x 100 mm) shown in Figure 3.1. The experimental compression test yielded the force displacement curve shown in Figure 3.2 and Table 3.1. Based on the experimental setup a simple numerical simulation will be carried out. One hexahedral finite element supported on the bottom side (red on Figure 3.3) and loaded on the top side (green on Figure 3.3) was used. The load was applied by prescribed displacement of 1 mm. A bilinear elasto-plastic material model was used with the following parameters $E = 200000 \text{ MPa}$, $\nu = 0$, $\sigma_{y,0} = 355 \text{ MPa}$, $\epsilon_{pl,0} = 0$, $\sigma_{y,1} = 2000 \text{ MPa}$ and $\epsilon_{pl,1} = 0$. The resultant reaction force in supports will be output at the following time points: 1E-6, 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.7, 1.0 and saved in the *.dat file. The described input file is shown in Listing 3.1.

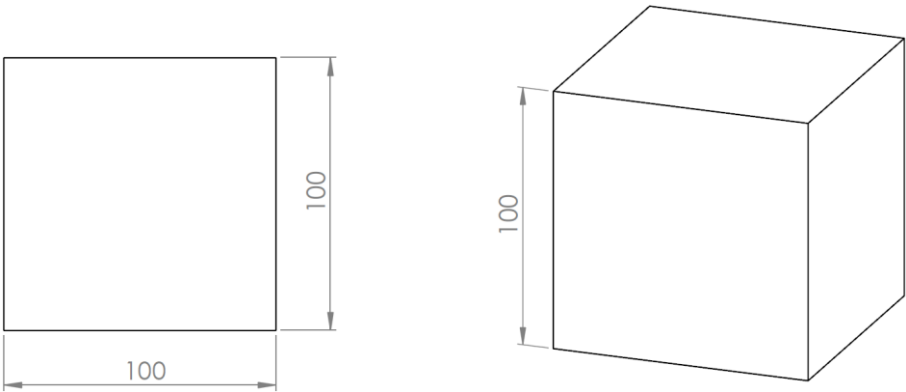


Figure 3.1: Sample model

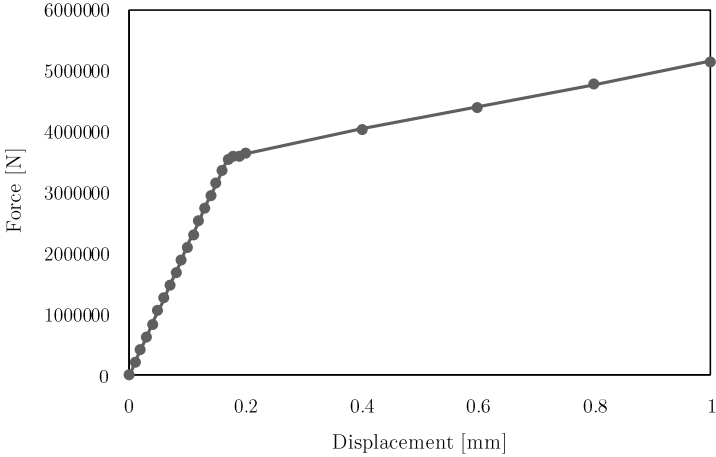


Figure 3.2: Experimental force displacement curve

Table 3.1: Experimental force displacement curve

Displacement [mm]	Force [N]
0.00	0.0
0.01	2.10E+05
0.02	4.20E+05
0.03	6.30E+05
0.04	8.40E+05
0.05	1.05E+06
0.06	1.26E+06
0.07	1.47E+06
0.08	1.68E+06
0.09	1.89E+06
0.1	2.10E+06
0.11	2.31E+06
0.12	2.52E+06
0.13	2.73E+06
0.14	2.94E+06
0.15	3.15E+06
0.16	3.36E+06
0.17	3.55E+06
0.18	3.58E+06
0.19	3.60E+06
0.2	3.63E+06
0.4	4.03E+06
0.6	4.40E+06
0.8	4.77E+06
1	5.14E+06

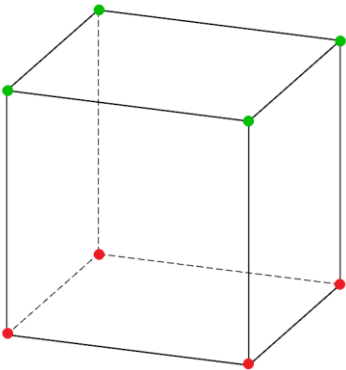


Figure 3.3: Node sets for support (red) and load (green)

Listing 3.1: CalculiX input file

```

*Heading
Model: Model-1,      Date: 1/1/2016
*Node
1, 0.0000000E+000, 0.0000000E+000, 1.0000000E+002
2, 0.0000000E+000, 0.0000000E+000, 0.0000000E+000
3, 0.0000000E+000, 1.0000000E+002, 1.0000000E+002
4, 0.0000000E+000, 1.0000000E+002, 0.0000000E+000
5, 1.0000000E+002, 0.0000000E+000, 1.0000000E+002
6, 1.0000000E+002, 0.0000000E+000, 0.0000000E+000
7, 1.0000000E+002, 1.0000000E+002, 1.0000000E+002
8, 1.0000000E+002, 1.0000000E+002, 0.0000000E+000
*ELEMENT, TYPE=C3D8
1, 1, 2, 4, 3, 5, 6, 8, 7
*Nset, nset=Load
3, 1, 5, 7
*Nset, nset=Support
2, 4, 8, 6
*Elset, elset=Sample
1
*****
** Materials
*****
*Material, name=Steel
*Elastic
200000, 0.0
*Plastic
355, 0
2000, 1
*****
*Solid section, elset=Sample, material=Steel
*TIME POINTS,NAME=T1
1E-6,0.01,0.02,0.05,0.1,0.2,0.3,0.4
0.5,0.7,1.0
*Step
*Static
1, 1
*Boundary
Support, 1, 3, 0
*Boundary
Load, 3, 3, -1
*Node print, nset=Support, time points=T1, totals=only
RF
*End step

```

3.1.1 Design of experiments

In order to find the material parameters that will give the same resultant force displacement curve as the experiment the values of E and $\sigma_{y,1}$ will be used as design parameters. First a design of experiments (DOE) procedure will be carried out and then an optimization will be performed.

The Optimax optimization workflow requires *input variables* to change the design parameter values, a *component* to perform the CalculiX analysis based on the provided input file, the *output array* to read the resulting reaction force curve from the output file, the objective function that will be optimized, the *driver* to set the optimization algorithm and its settings and some recorders to record the results.

Start the Optimax and if the **Edit project** dialog (Figure 3.4) is not open select **PROJECT** → **New** to open it. Input the **Name** of the project and **Browse** for the project directory. The **Work directory** and **Results directory** fields are automatically filled in (the work directory is used to run the components of

the workflow and the results directory is used to store the results; results are overwritten each time the workflow is run). Close the dialog with the **OK** button.

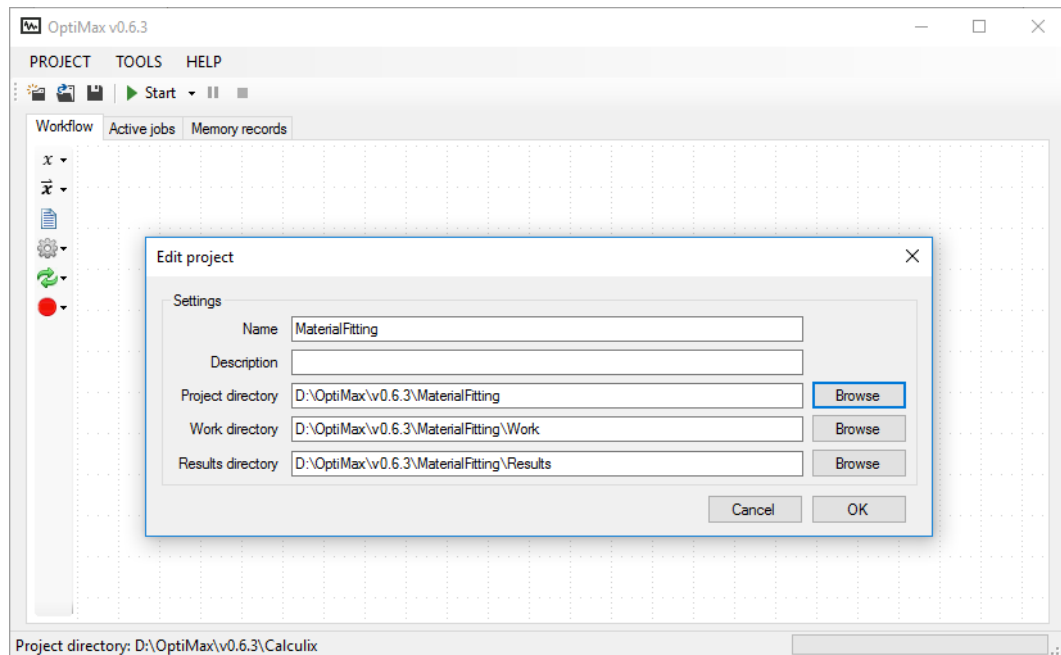


Figure 3.4: **Edit project** dialog

3.1.1.1 *Input variables*

Add the design parameters (*input variables*) named *E0* and *Sy1*, that will be changed by the *driver*. From the left toolbar select **Add variable** → **Input variable**. In the **Variable Input Editor** select the variable type as **Double** and enter the **Name** of the first parameter as *E0* (Figure 3.5). Close the dialog with **OK**. Repeat the procedure for the parameter *Sy1*.

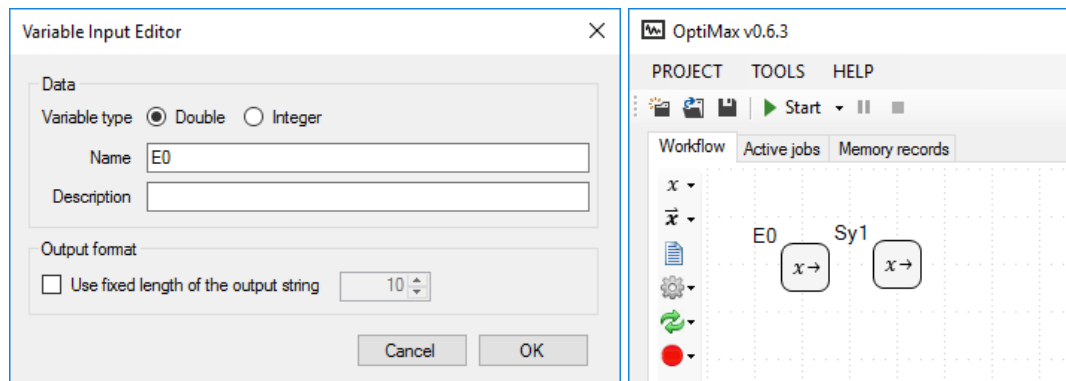


Figure 3.5: **Variable Input Editor** (left) and resulting Optimax window

3.1.1.2 *Input file*

Create a new text file, copy/paste the text shown in Listing 3.1 into it and save the file as *sample.inp*. Then add the *file resource* block from the left toolbar with **Add file resource** icon. In the **Edit File Resource** dialog select the resource type as **Input file**, enter the **Name** of the file resource as *inp* and browse for the created *sample.inp* text file (Figure 3.6). Close the dialog with the **OK** button (the file will be copied to the *Files* subdirectory of the project folder for later use).

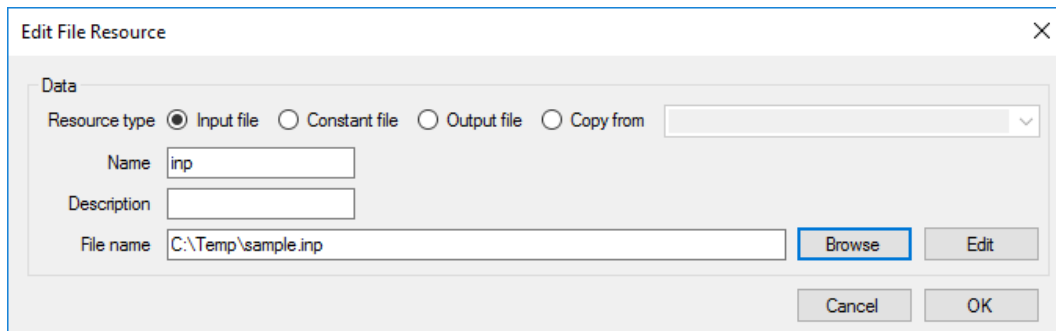


Figure 3.6: **Edit File Resource** dialog for the input file

3.1.1.3 Output file

The *inp* file resource will be used to run the CalculiX analysis and the results of the analysis will be saved to another file with the same name as the input file but with a *dat* extension: *sample.dat*. To connect this results file to Optimax, create another *file resource* block using the left toolbar and selecting the **Add file resource** icon. In the **Edit File Resource** dialog select the resource type as **Output file**, enter the **Name** of the file resource as *output* and enter the **File name** as *sample.dat* (Figure 3.7). Close the dialog with the **OK** button.

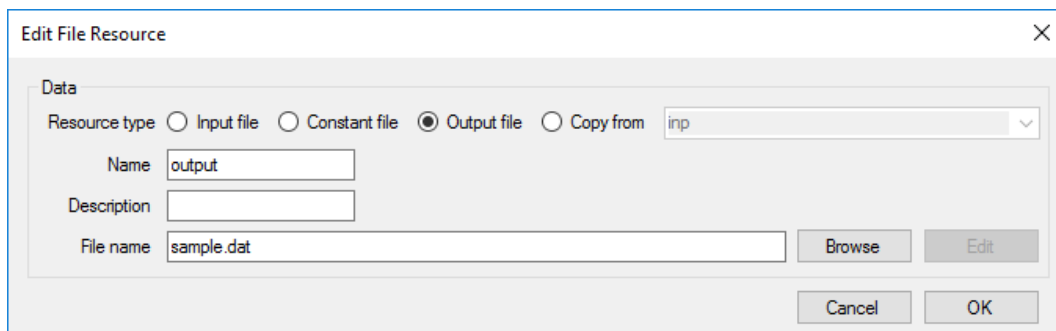


Figure 3.7: **Edit File Resource** dialog for the output file

3.1.1.4 Component

Add a *dos component* block from the left toolbar by selecting **Add component** → **Dos component** icon. Enter the **Name** of the *component* block as *solver* and browse for the CalculiX solver executable in the **Use non-local exe file** field. The *dos component* will execute the selected exe file using the arguments supplied in the **Argument** field so insert *sample* in the **Argument** field (to run the CalculiX analysis in a command prompt the name of the used input file must be supplied as an argument). **Check** the checkbox in front of the *inp* and *output* file resources (if the file resource is checked it will be copied/used/created by the component). Close the dialog with the **OK** button.

3.1.1.5 Driver

To connect and run the workflow a *driver* block must be added. From the left toolbar select the *DOE driver* block by **Add driver** → **DOE: Design of experiments** icon. **Check** the checkboxes in front of parameters *E0* and *Sy1*. For all parameters their lower and upper limits must be entered together with the number of values that will be generated between them. For the *E0* parameter enter the value of *150000* in the **Min** column and *250000* in the **Max** column. For the *Sy1* parameter enter the value of *1000* in the **Min** column and *8000* in the **Max** column (Figure 3.9). The **Num. of values** column determines how many different values will be used for each parameter in the DOE study. Set this value to *1* for both parameters (the workflow will be executed only once). In the **Workflow** box check the checkbox in front of the *solver* component in order to add it to the workflow. Close the dialog with the **OK** button.

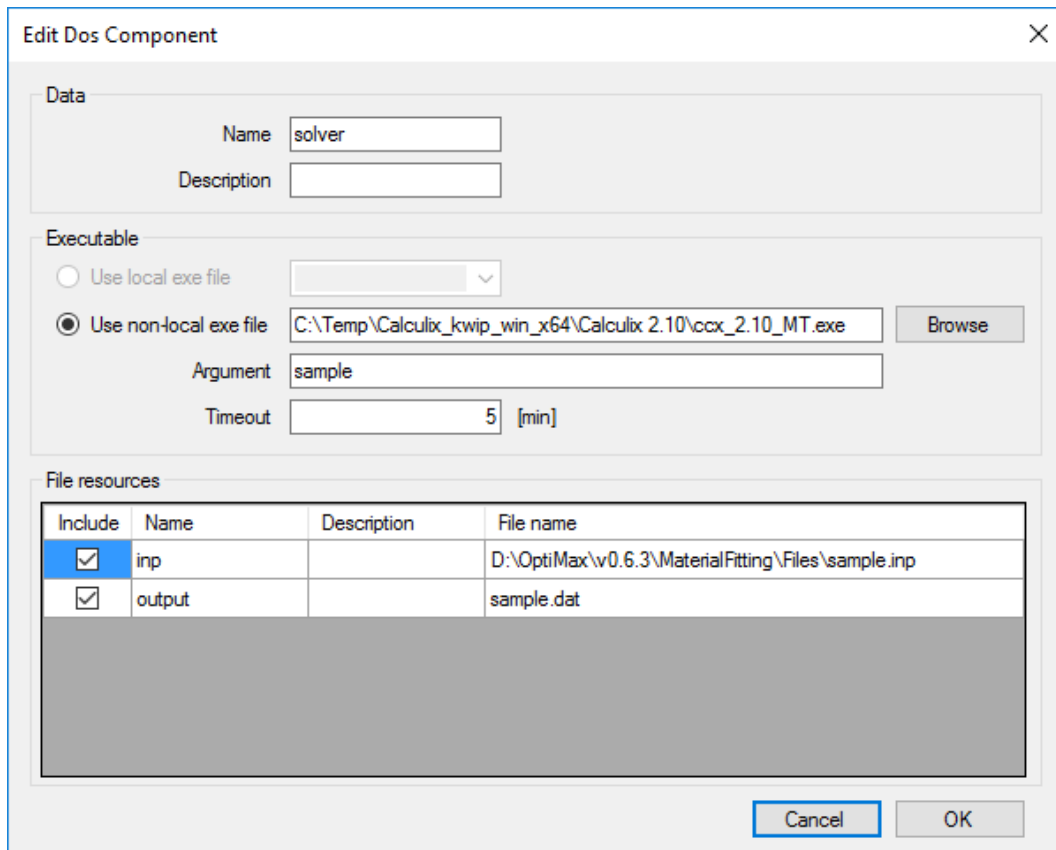


Figure 3.8: **Edit Dos Component** dialog

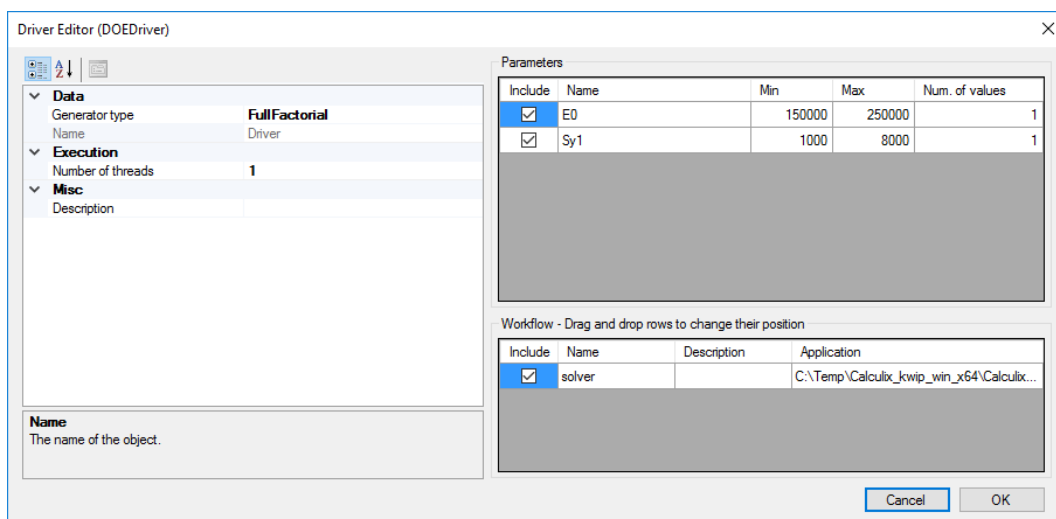


Figure 3.9: **Driver Editor** dialog

Reposition all blocks using Drag&Drop as shown in Figure 3.10 and save the project using **PROJECT** → **Save As...** Browse for the current project directory (Figure 3.4) and use *MaterialFitting.om* as the file name.

Now we can test run the workflow using the **Start** icon from the top toolbar. In the bottom output field a **Job started** line will be displayed and a couple of seconds later a **Job finished** line should appear. The resulting output files are stored in the directory “*project directory\Work\1\solver*”. All files in the Work subdirectory of the project directory are deleted before each new run of the workflow so they should be copied if needed and closed before running the workflow again (check this folder if the file

sample.dat exists; if not open the text file named “_output...” for possible errors during CalculiX execution).

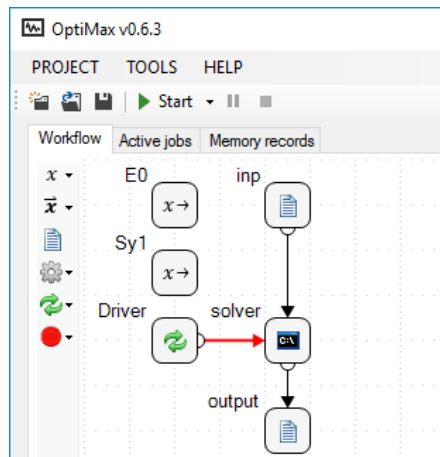


Figure 3.10: Optimax workflow

3.1.1.6 Connecting the input file with the input variables

In order to change the material properties in the finite element model the values of the *input variable* blocks *E0* and *Sy1* must be inserted into the CalculiX input file. The value of the *input variable* is inserted into text files by inserting @name@ string, where *name* is the name of the *input variable* block.

Double click the *inp file resource* block to open **Edit File Resource** dialog and select **Edit** to edit the input file using the system default text editor of *inp* files. If no program is associated with the *inp* file extension, select **Notepad** (or any other text file editor) from the displayed program list. In the opened text editor search for the *Elastic and *Plastic keywords (lines 27 and 29 in the Listing 3.1). Replace the current value of the elastic modulus (200000) with the text string @E0@ (line 28) and the current value of the second yield stress (2000) with the text string @Sy1@ (line 31) to obtain the text shown in Listing 3.2. Save the *sample.inp* file, close the text editor and close the **Edit File Resource** dialog with the **OK** button.

Listing 3.2: CalculiX input file connected with input variables *E0* and *Sy1*

```
*****
** Materials
*****
*Material, name=Steel
*Elastic
@E0@, 0.0
*Plastic
355, 0
@Sy1@, 1
```

3.1.1.7 Output array

The results of the analysis that are written in the *sample.dat* file will be read into Optimax using *Output array* block since we are not interested in only one value but in a series of values through time (*array* block instead of *variable* block). Select **Add array** → **Output array** icon in the left toolbar. In the **Output Array Editor** dialog select the **Variable type** as **Double**, enter the **Name** of the output array as *RF* and in the **File resource** field select the *output* block (Figure 3.11).

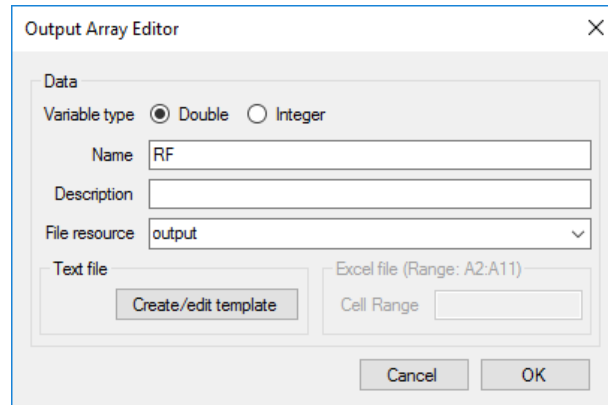


Figure 3.11: **Output Array Editor** dialog

Then a data mining template must be created to select the data that will be read from the *output file resource*. Select the **Create/edit template** button to open the file browser dialog, navigate to the previously created results file *sample.dat* in the “*project directory\Work\I\solver*” directory and open it using **Open** button (the file will be copied to the *Files* subdirectory of the project folder for later use and a *template* extension will be appended to the file name). In the opened **Data Mining** dialog, the contents of the *sample.dat* file are shown on the right. The actual data that needs to be read is written in lines 4, 8, 12 ... while all other lines contain unneeded data. To read the data only from every fourth row change the value of the **Skip lines** field to 3 and then select all data lines from 4 to 44 using mouse or keyboard selection. Then apply the mining template using **Set Data** button. The first data column in every fourth line is selected (data is colored in gray) that represents the reaction force in the x direction. Change the value of the **Data word col** to 2 in order to select the reaction force in the z direction (Figure 3.12). Apply the created data mining template with **OK** button and then close the **Data Mining** dialog with the **OK** button.

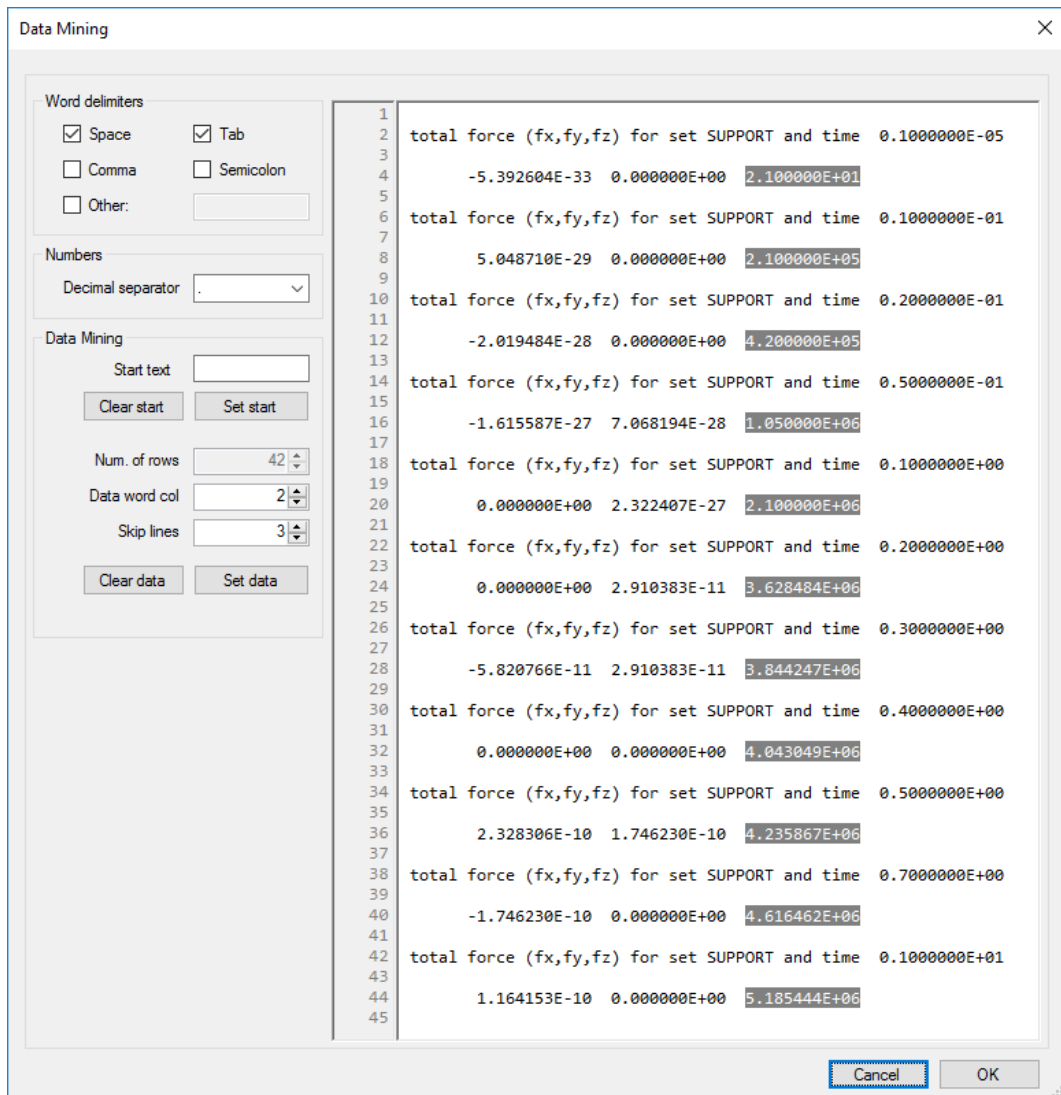


Figure 3.12: Data Mining dialog

3.1.1.8 Recorder

To follow the values of different data blocks a *recorder* block must be added. A *memory recorder* and a *csv recorder* blocks will be added. *Memory recorder* records the data inside Optimax while *csv recorder* records the data to the csv file (comma separated values) which is saved to the “*project directory*\Results\” directory. Select **Add recorder** → **Memory recorder** icon in the left toolbar. In the **Edit Memory Recorder** dialog check the checkboxes in front of all variables *E0*, *Sy1* and *RF* to record them (Figure 3.13). Close the **Edit Memory Recorder** dialog with the **OK** button.

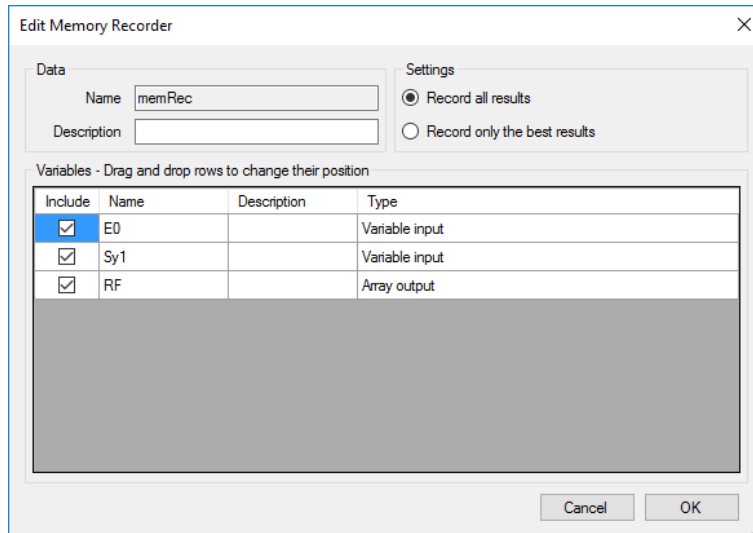


Figure 3.13: Edit Memory Recorder dialog

Repeat the procedure for the *csv recorder* (**Add recorder** → **Csv recorder**) and in the **Name** field insert *csvRec* (Figure 3.14). One workflow can contain only one *memory recorder* block while the number of other recorders is not limited. The resulting workflow is presented in Figure 3.15.

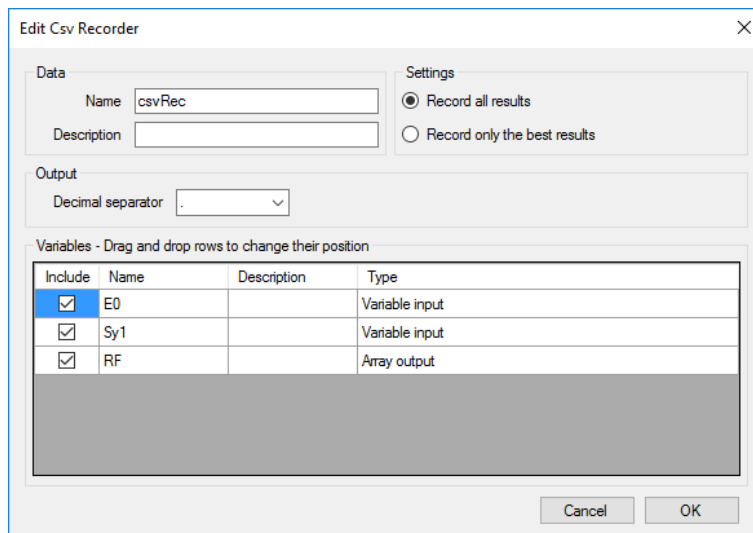


Figure 3.14: Edit Csv Recorder dialog

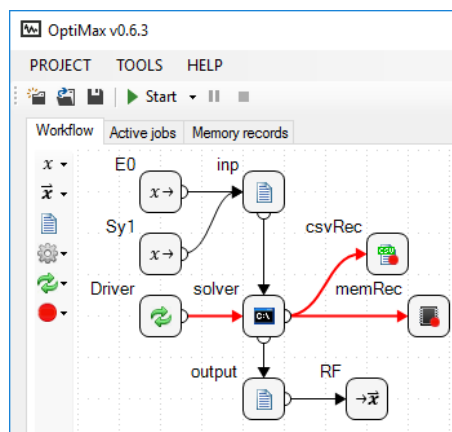


Figure 3.15: Optimax workflow

To test the last changes to the framework, we will run the workflow again but this time with different parameter values and more than once. Double click the *Driver* block (or right click it and select **Edit**) to open **Driver Editor** dialog and change the values of both parameters in the **Num. of values** column to 2 as shown in Figure 3.16 (both parameters will take 2 values and the framework will be run four times for all possible parameter value combinations).

Include	Name	Min	Max	Num. of values
<input checked="" type="checkbox"/>	E0	150000	250000	2
<input checked="" type="checkbox"/>	Sy1	1000	8000	2

Figure 3.16: Parameter values

Now run the workflow using the **Start** icon from the top toolbar. In the bottom output field, a **Job started** line will be displayed and a couple of seconds later a **Job finished** line should appear. The **Active jobs** tab displays the progress of each workflow run by its ID and the data values recorder by the *memory recorder* block are displayed in the **Memory records** tab (Figure 3.17). The data table contains the IDs and recorded values of all parameters selected in the *memory recorder* block. Each cell in column **RF** contains the reaction force of the analysis as a series of values separated by space characters. Such a series can be graphically displayed by selecting the appropriate cell with the right mouse button and selecting **Add to graph** option (in this way only y data is added to the graph and so the values on the x axis start at 0 and increase in steps of 1). Using the vertical scrollbar on the right the call that is graphically displayed can be moved up and down the table. More than one cell can be added to the graph at the same time.

The *csv recorder* data can be opened by right clicking on the *csv recorder csvRec* block and selecting **Open *.csv file** option. The file will open in the default csv editor (for the next workflow run the csv file must first be deleted so it must be closed beforehand).

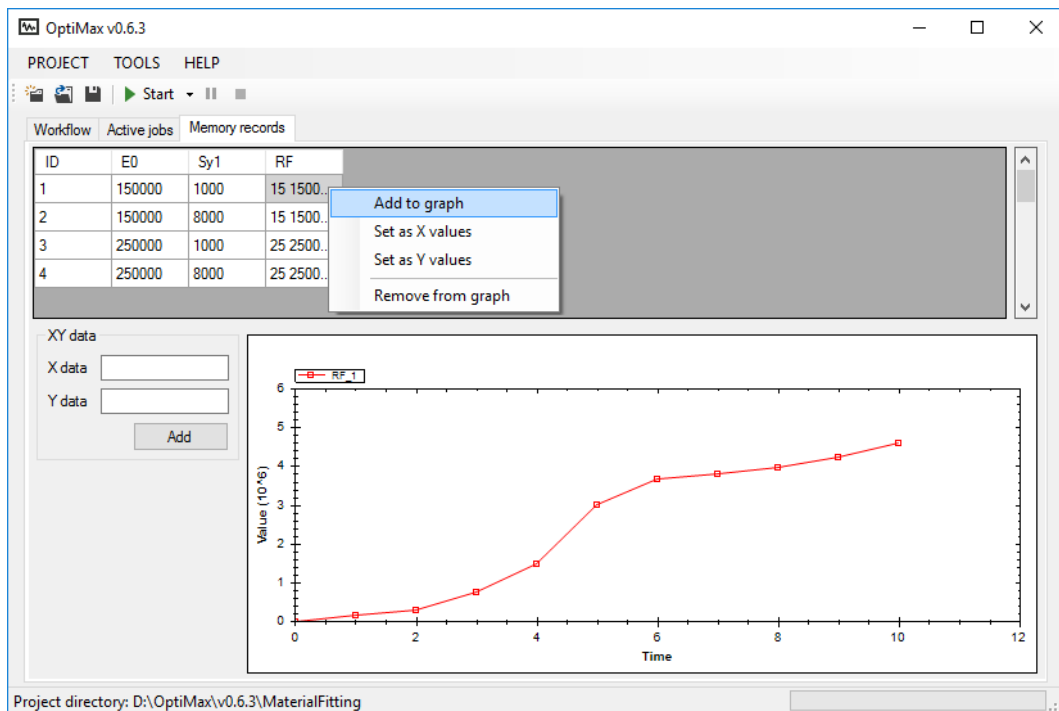


Figure 3.17: Memory records tab

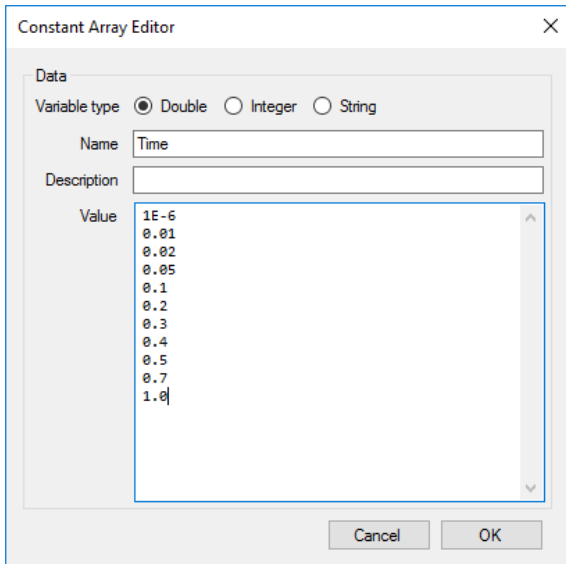
3.1.1.9 Variable from compare curve

In order to compare the computed reaction force RF with experimentally determined force displacement curve (Figure 3.2) a *variable from compare curve* block will be used, which requires the data series array for x and y axis for numerical and experimental curves. The resulting value represents the difference between the input curves. The curves are more similar if the value is low and vice versa.

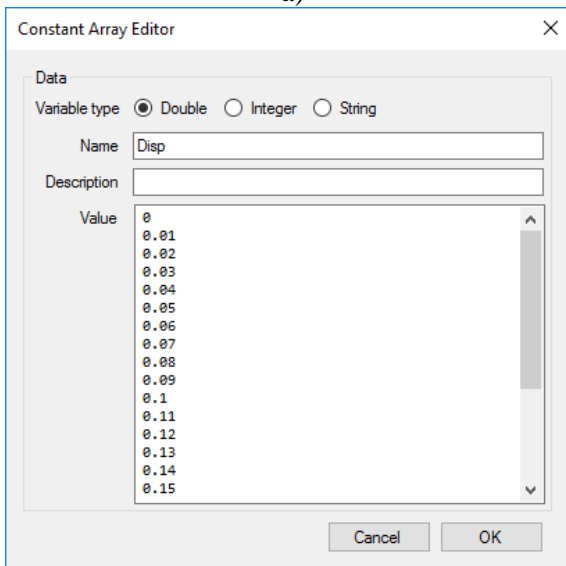
The y axis of the numerical curve is already defined by the *output array* block RF while the x axis is missing. The x axis of the numerical curve represents the time points at which the reaction force was outputted. Because these time points are constant a *constant array* block will be used. Select **Add array** → **Constant array** icon in the left toolbar. In the **Constant Array Editor** dialog select the **Variable type** as **Double**, enter the **Name** of the array as *Time* and in the **Value** field enter the values from the *TIME POINTS keyword: 1E-6, 0.01, 0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.7 and 1.0. Each value must be entered in a new empty line (Figure 3.18a). Close the dialog with the **OK** button.

Since the experimental curve (Table 3.1) is also constant it will also be input using a *constant array* block. First the values of experimental x axis will be input. Select **Add array** → **Constant array** icon in the left toolbar. In the **Constant Array Editor** dialog select the **Variable type** as **Double**, enter the **Name** of the array as *Disp* and in the **Value** field paste the first column of the Table 3.1 (3.18b). Close the dialog with the **OK** button.

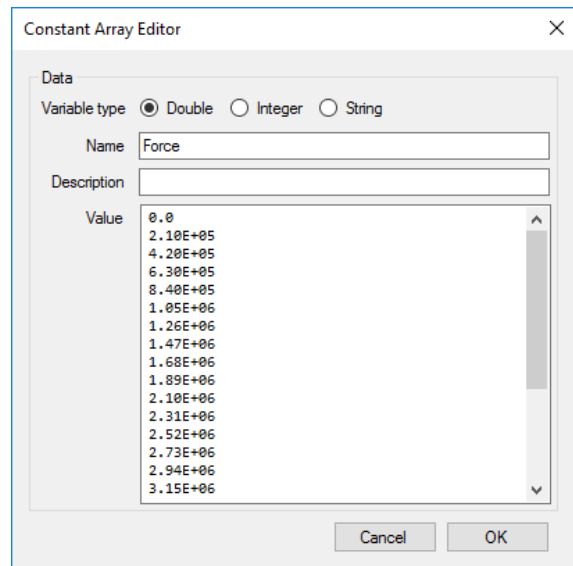
Next the values of experimental y axis will be input. Select **Add array** → **Constant array** icon in the left toolbar. In the **Constant Array Editor** dialog select the **Variable type** as **Double**, enter the **Name** of the array as *Force* and in the **Value** field paste the second column of the Table 3.1 (Figure 3.18). Close the dialog with the **OK** button.



a)



b)



c)

Figure 3.18: Constant Array Editors for a) *Time*, b) *Disp* and c) *Force* blocks

After all needed data arrays are defined the *variable from compare curves* block can be added. Select **Add variable** → **Variable from compare curves** icon in the left toolbar. In the **Variable From Compare curves** dialog select the **Variable type** as **Double**, enter the **Name** of the variable as *Err* and select the comparison **Method** as **Mean square error**. In **Curve 1** box select the *Disp* array as **X data array** and *Force* array as **Y data array**. In **Curve 2** box select the *Time* array as **X data array** and *RF* array as **Y data array** as shown in Figure 3.19 (before comparison the curves are normalized along the x axis; then for each data point in the curve with the smaller number of points the squared distance to the other curve is computed; at the end the value is averaged). Close the dialog with the **OK** button. The resulting workflow is shown in Figure 3.20.

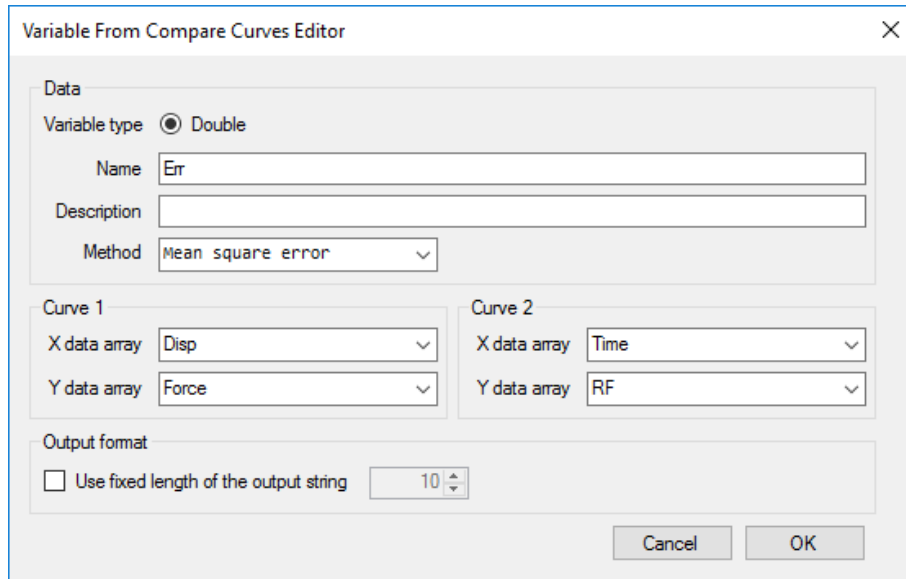


Figure 3.19: Variable From Compare curves Editor dialog

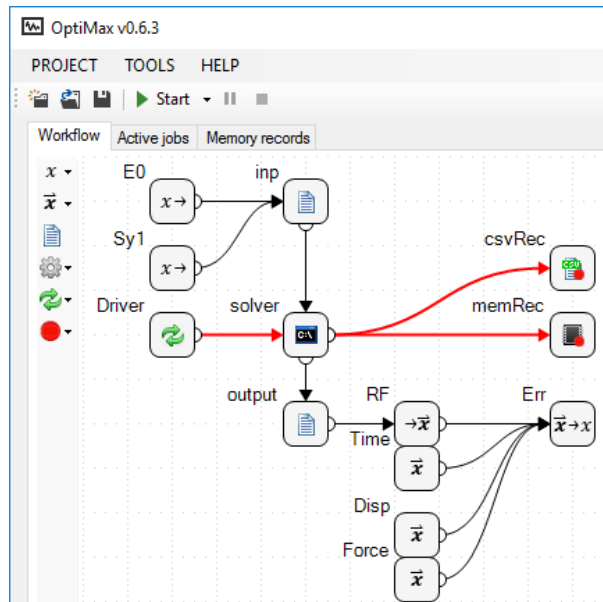


Figure 3.20: Optimax workflow

To record the value of the newly added blocks (*Err*, *Time*, *Disp* and *Force*) edit the *memory recorder* block *memRec* and the *csv recorder* block *csvRec* and check the checkboxes in front all unchecked variables. Close the editor dialogs and run the workflow using the **Start** icon from the top toolbar. After the job is finished the values of the recorded variables can be observed in **Memory records** tab. To find the result with the best resultant force curve fit (the smallest value of the *Err* variable) click the title row of the **Err** column so that the data table is sorted in ascending order. Figure 3.21 shows that the material parameters with the best curve fit are $E0 = 250000$ and $Sy1 = 1000$.

To graphically display the numerical curve with the smallest *Err* value, right click the first cell of the **Time** column and select **Set as X values** and then right click the first cell in the **RF** column and select **Set as Y values**. Then click the **Add** button in the **XY data** box. To also display the experimental curve, repeat this procedure with the columns **Disp** and **Force** (Figure 3.21).

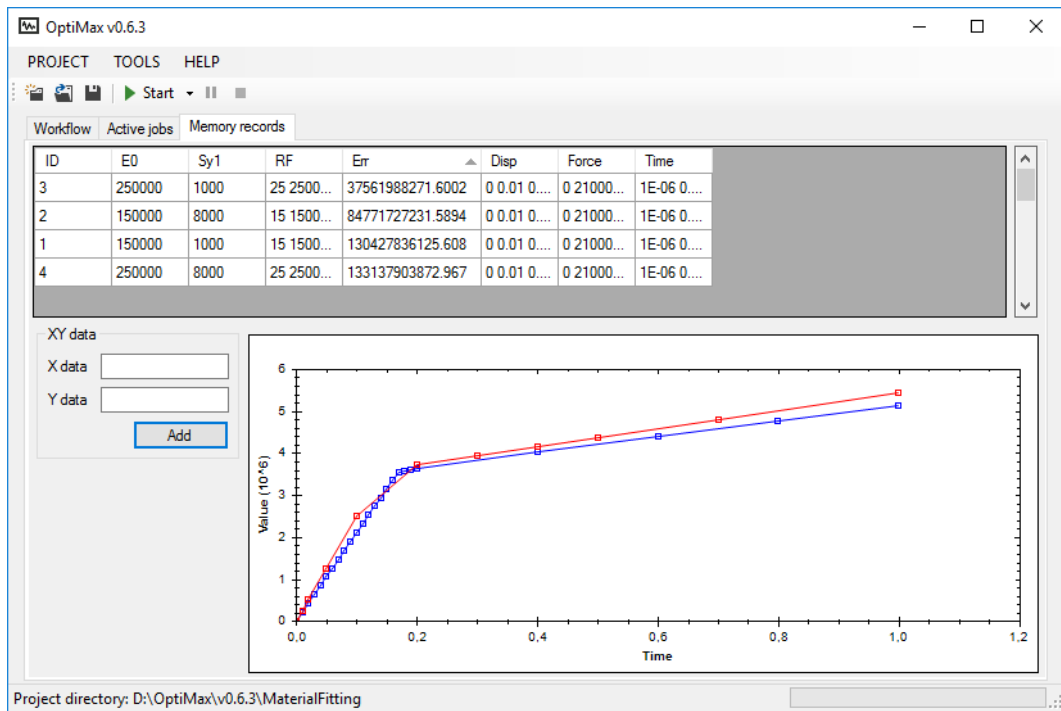


Figure 3.21: Memory records tab

3.1.2 Optimization using Nelder-Mead simplex method

The Nelder-Mead optimization algorithm requires an objective function which will be minimized during optimization. In this case the value of *Err* variable can be used as the objective function since smaller value of *Err* variable represents a better fit between numerical and experimental curve.

3.1.2.1 Driver

Optimax workflow can only have one *driver* block so first delete the existing *Driver* block. Select the *Driver* block with the right mouse button and select **Delete**. Confirm the deletion with **Yes**. Then add the *Nelder-Mead driver* block by **Add driver** → **NM: Nelder-Mead simplex method** icon. In the driver options set the value of the **Max number of increments** to 50 (if the optimization does not converge it will stop after 50 optimization increments) and the value of the **Objective variable** to *Err*. In the **Parameters** box the values of **Min** and **Max** columns are already defined but the starting value and the starting delta for the simplex optimization must be set. For the *E0* parameter enter 200000 in the **Value** column and 10000 in the **Delta** column. For the *Sy1* parameter enter 3500 in the **Value** column and 200 in the **Delta** column (Figure 3.22). Close the dialog with the **OK** button. The resulting workflow remains the same as before (Figure 3.20).

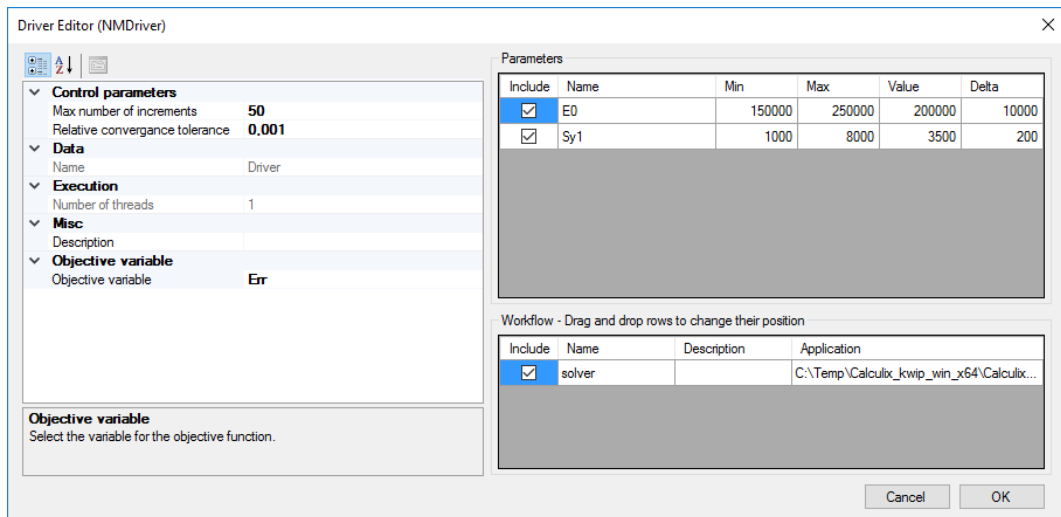


Figure 3.22: Driver Editor

Run the optimization using the **Start** icon from the top toolbar. During the optimization the values of the recorded variables can be observed in the **Memory records** tab. To find the result with the smallest value of the *Err* variable click the title row of the **Err** column so that the data table is sorted in ascending order. After 27 increments the convergence criteria is met and the optimization terminates. The results of the optimization are shown in Figure 3.23.

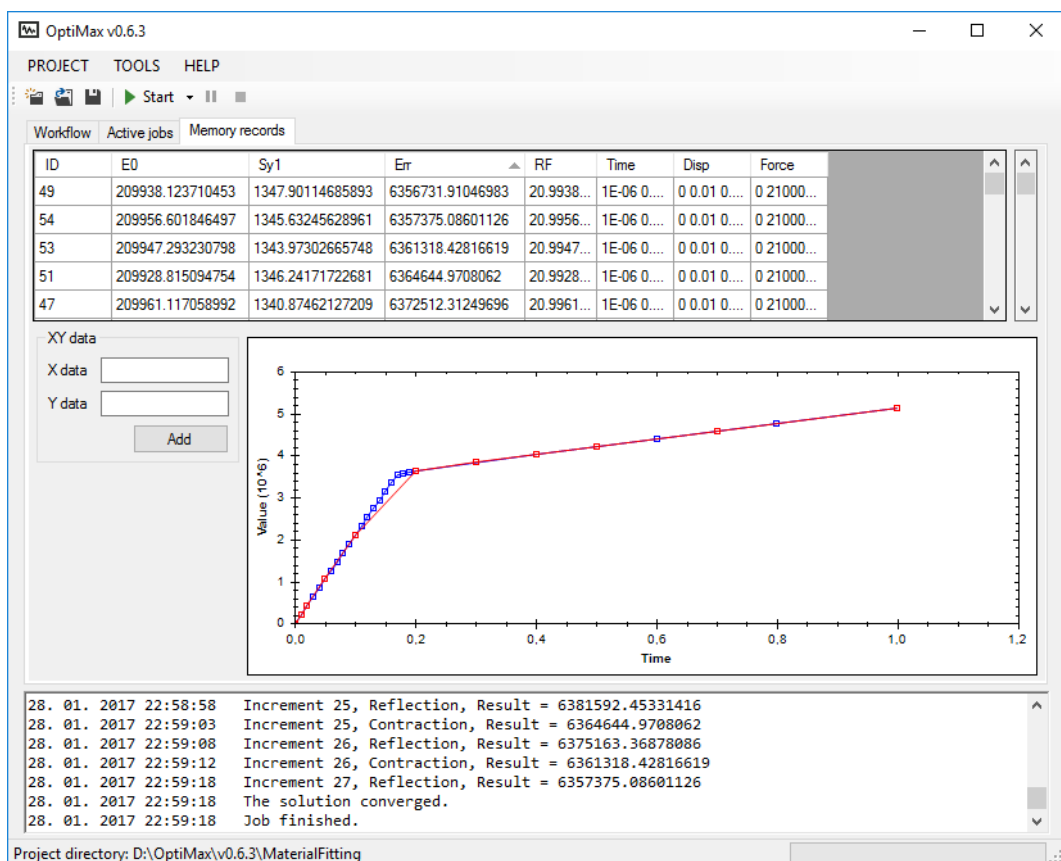


Figure 3.23: Driver Editor

The resulting numerical curve fits the experimental curve with the mean square error of $6.36E6 \text{ N}^2$ or a mean error of $\sqrt{6.36E6} = 2500 \text{ N}$. The resulting optimal material parameters are $E0 = 209938 \text{ MPa}$

and $SyI = 1347.9$ MPa. In comparison, the experimental curve for this tutorial was obtained from the numerical analysis with the following material parameters $E0 = 210000$ MPa and $SyI = 1355$ MPa.