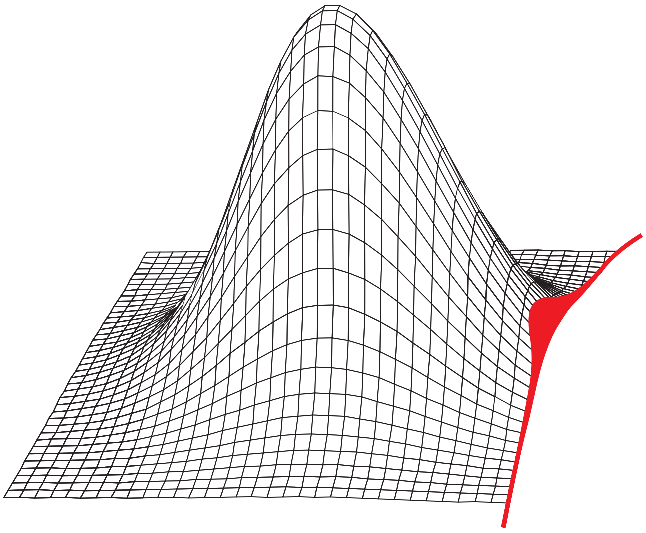
ProSi



MANUAL
Version 2.3

Technische Universität Dresden
Institute for Fluid Mechanics
Helmholtzstr. 10
D01069 Dresden

Authors: Matthias Voigt (matthias.voigt@tu-dresden.de)
Gunter Lang (gunter.lang@mailbox.tu-dresden.de)
Thomas Bischoff (thomas.bischoff@.tu-dresden.de)
Thorsten van Lil (thorsten.van_lil@tu-dresden.de)

Dresden, September 2011

This book was typeset
using LaTeX 2_ε

Contents

Introduction

ProSi (**Pro**babilistic **Si**mulation) is a software package for probabilistic analysis of generic physical systems. It consists of the ProSi core module and a user interface which provides full functionality for the Pre- and Postprocessing. Furthermore the user interface manages the deterministic simulation runs, including the possibility of distributed computing in a local area network.
ProSi has the functionality to perform Monte Carlo Simulations (MCS) and Response Surface Method (RSM) analysis, as well as a combination of both methods. The following sampling methods are supported by the MCS-modul included in ProSi:

- Simple Random Sampling
- Latin Hypercube Sampling
- Descriptive Sampling

For all three sampling methods the generation of user defined correlation structures in the input data is supported.

The RSM-modul supports linear, quadratic and cubic response surfaces. Following designs are supported:

- 2^{n_b} Design
- Central Composite Design
- Box Behnken Design
- Simplex Design

Furthermore it is possible to approximate response surfaces using the results of a MCS and also to perform a MCS using a response surface instead of a deterministic solver.
The ProSi core module is controlled by an ASCII-file. The structure of this file is described in the sections 4.2.1 and 4.2.2. Furthermore the command *./prosi –help* (on Linux) resp. *prosi.exe -help* (on Microsoft Windows command line) shows a short description of the input file. To set up this file and to use all of ProSi's features, the graphical user interfaces ProSi Pre, ProSi Exe and ProSi Post have been created. They are described in sections 3.1, 3.2 and 3.4. Further information about the probabilistic theories used in ProSi are given in chapter 2 and, in German language, in [**?**] [**?**].

1. Installation

1.1. Before installing ProSi

1.1.1. on Linux/Unix

Before ProSi and its auxiliary tools can be installed and used the following conditions have to be fulfilled.

- A C++ compiler has to be available on the system. Here the GNU GCC compiler is the default one, but others should also work.

- Tclkit has to be available on the system. Tclkit is a command interpreter which is needed by the graphical user interface and the ProSi Server which are written in Tclkit script language. A Tclkit interpreter can be downloaded from the internet as described at 6 below.

- rcp, rsh commands have to be useable without password protection for distributed computing across the LAN. If this commands are not available the calculations can only be performed on the local PC.

- csh (C-Shell) or bash (Bash-Shell) has to be available on the PC. The user interface can use either of the two shells to control the single deterministic calculations. If both shell types are not available on the system and can not be installed please contact the developers as it is also possible to use other script formats (korn shell for example).

- convert command has to be available if the user wants to print *.gif pictures out of ProSi Post. Convert is part of the freeware package ImageMagic. If this tool is not available only *.ps graphics can be plotted.

1.1.2. on Microsoft Windows

Before ProSi and its auxiliary tools can be installed and used the following conditions have to be fulfilled. System Requirements for vcredist_x86.exe:

- Supported Operating Systems: Windows 2000 Service Pack 4; Windows Server 2003; Windows Server 2008; Windows Vista; Windows XP

- Minimum: 6 MB (x86)

- Required Software: Windows Installer 3.0 Windows Installer 3.1 or later is recommended.

1.2. Installing ProSi

1.2.1. on Linux/Unix

For the installation of ProSi, the user needs the permission to write in a directory and a C++ compiler installed on the system.
For the installation follow the steps described below:

1. Create an empty directory, for example:

   /mkdir /home/“name of the user”/prosi_2.3

2. Copy the file prosi_2.3.tar.gz into this directory. For example:

   cp prosi_2.3.tar.gz /home/“name of the user”/prosi_2.3

3. Change to this directory with:

   cd /home/“name of the user”/prosi_2.3

4. Unpack the file prosi_2.3.tar.gz with:

   gunzip prosi_2.3.tar.gz

5. Untar the file prosi_2.3.tar with:

   tar xvf prosi_2.3.tar

   The following directories should be created in the directory /home/“name of the user”/prosi_2.3:

prosi_2.3	the main directory of ProSi and his utility programs
documentation	the directory for the ProSi documentation
input_data	the directory for input files
scripts	the directory for user scripts

6. Download of the tcl-interpreter for your system from the internet page:

   http://www.equi4.com/pub/tk/downloads.html

7. Save the tcl-interpreter in the directory: /home/“name of the user”/prosi_2.3/prosi_2.3 using the name tclkit.

8. Check the compiler definition in the file: /home/“name of the user”/prosi_2.3/prosi_2.3/Makefile.in

   If you do not have the GNU C++ compiler, please change the compiler settings in the file: /home/“name of the user”/prosi_2.3/prosi_2.3/Makefile.in to your preferred compiler.

9. Compile ProSi and all the other programs by calling the install-script:

   ./INSTALL.csh

   This command will create the executable ProSi file and the utility program files in the directory: /home/“name of the user”/prosi_2.3/prosi_2.3

10. The directory /home/“name of the user”/prosi_2.3/prosi_2.3 should now contain the following files:

prosi
corre_matrix_calc
calculate_correlations
prosi_exe.kit
prosi_post.kit
prosi_pre.kit
prosi_server.kit
prosi_server_exe.kit
tclkit

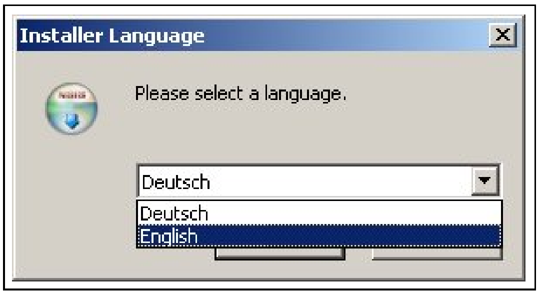
Further applications

In order to use the image export function in ProSi-Post, it's necessary to also install ImageMagick and Ghostscript. On most Linux distributions both should be installed by default. Otherwise it can easily be installed from within the package management. Contact the administrator to install the packages.

1.2.2. on Microsoft Windows

For Installation of ProSi is needed to have administrator permission. There is only one file, which starts an installation wizard, so you can follow the steps described.
During installation the file vcredist_x86.exe will be extracted and registered. This is a package of Microsoft libraries for the MSVC++ Compiler and necessary for executing ProSi on Windows-platforms.

1. Language Selection



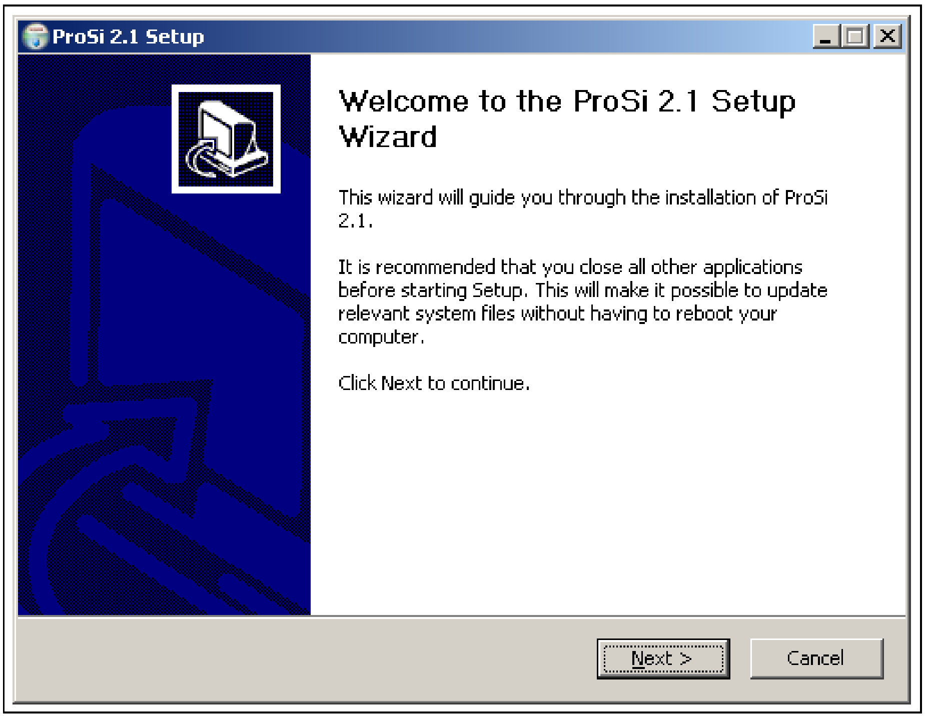
Installer Language

Please select a language.

Deutsch

Figure 1.1. "Installation Wizard - Language Selection"-Window

2. Confirm Advices



ProSi 2.1 Setup

Welcome to the ProSi 2.1 Setup Wizard

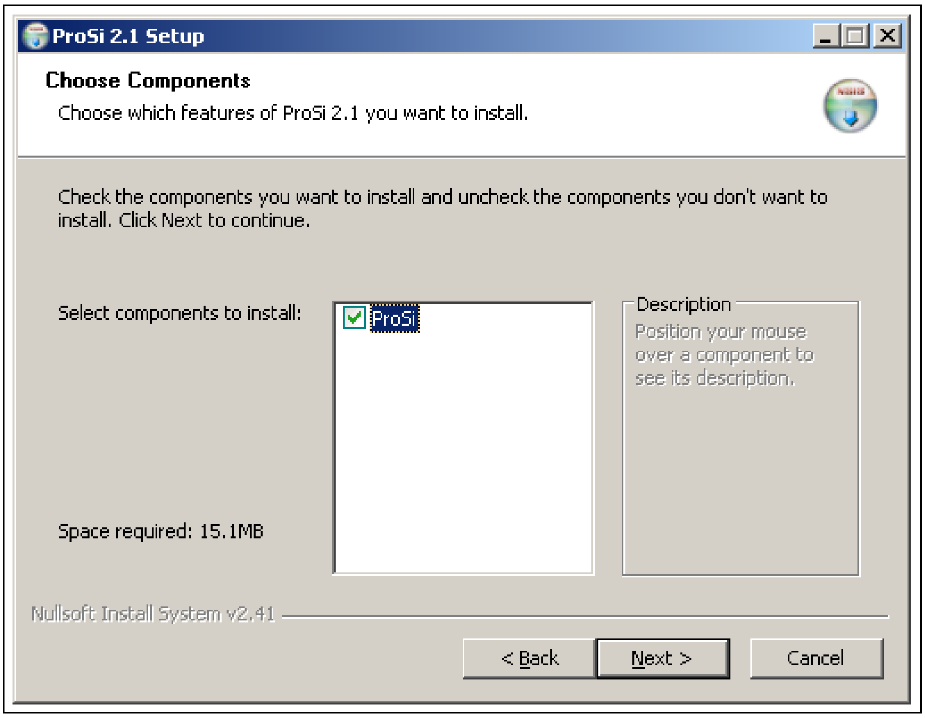
This wizard will guide you through the installation of ProSi 2.1.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Figure 1.2. "Installation Wizard - Confirm Advices"-Window

3. Package Selection



ProSi 2.1 Setup

Choose Components

Choose which features of ProSi 2.1 you want to install.

Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install: ☑ ProSi

Description

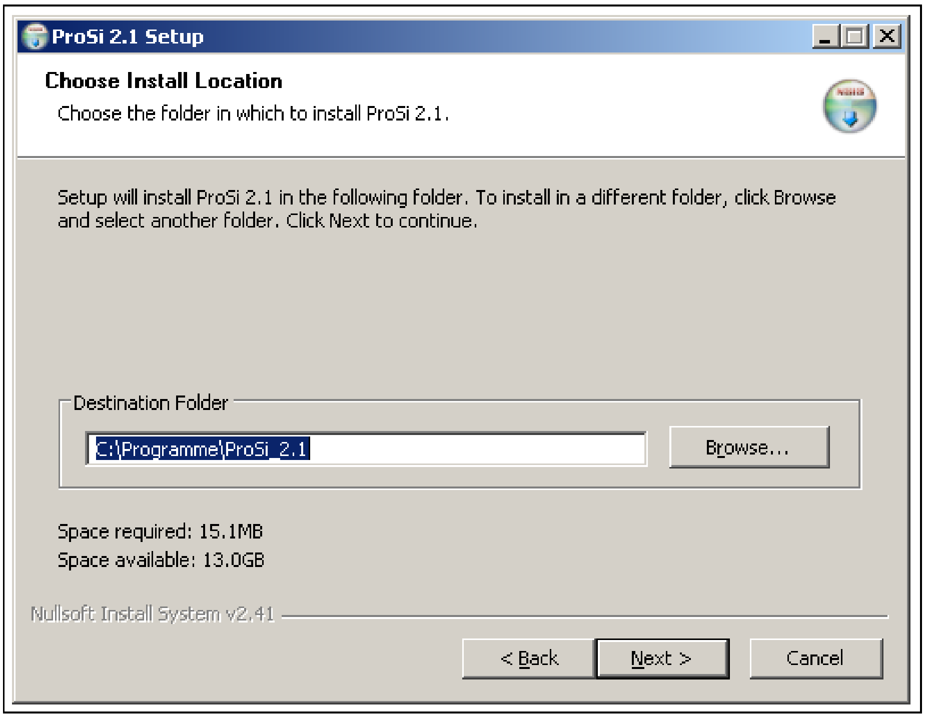
Position your mouse over a component to see its description.

Space required: 15.1MB

Nullsoft Install System v2.41

Figure 1.3. "Installation Wizard - the package"-Window

4. Choose Install Directory



Choose Install Location

Choose the folder in which to install ProSi 2.1.

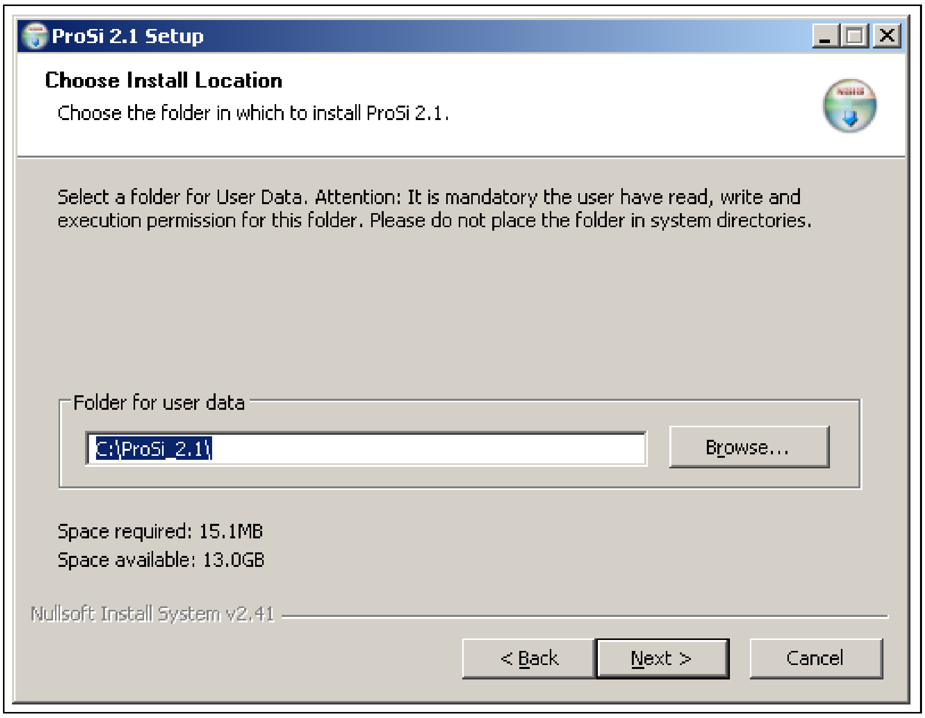
Setup will install ProSi 2.1 in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.

Destination Folder: C:\Programme\ProSi_2.1

Space required: 15.1MB
Space available: 13.0GB

Figure 1.4. “Installation Wizard - Installation Directory”-Window

5. Choose Install Directory for User Data



Choose Install Location

Choose the folder in which to install ProSi 2.1.

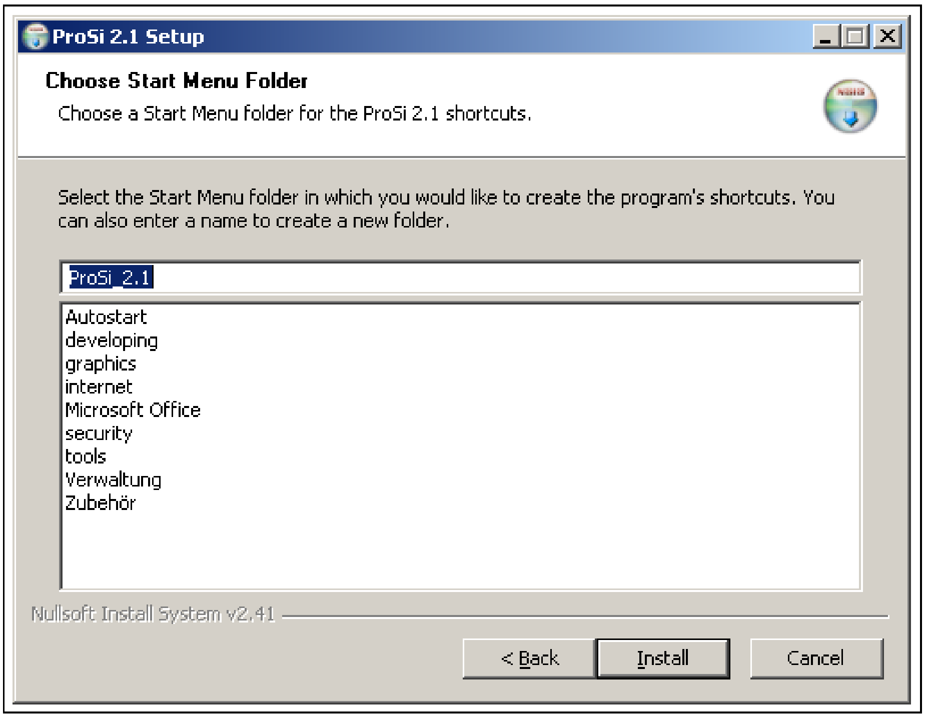
Select a folder for User Data. Attention: It is mandatory the user have read, write and execution permission for this folder. Please do not place the folder in system directories.

Folder for user data: C:\ProSi_2.1\

Space required: 15.1MB
Space available: 13.0GB

Figure 1.5. “Installation Wizard - Installation Directory for User Data”-Window

6. Start Menu Group Definition



Choose Start Menu Folder

Choose a Start Menu folder for the ProSi 2.1 shortcuts.

Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.

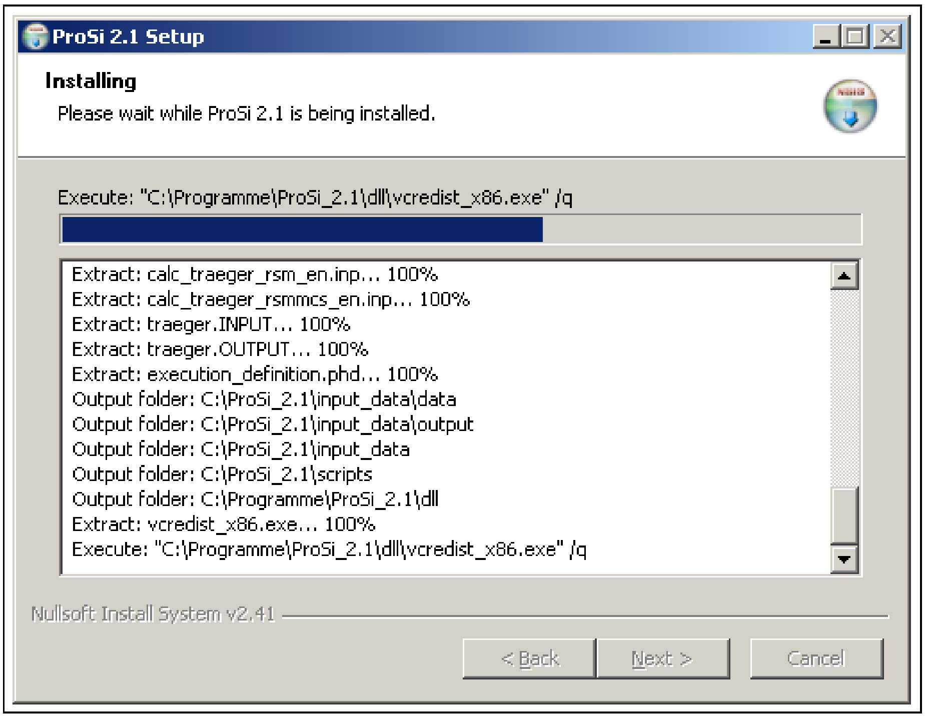
ProSi_2.1

- Autostart
- developing
- graphics
- internet
- Microsoft Office
- security
- tools
- Verwaltung
- Zubehör

Nullsoft Install System v2.41

Figure 1.6. "Installation Wizard - Start Menu Group"-Window

7. Files will be installed



Installing

Please wait while ProSi 2.1 is being installed.

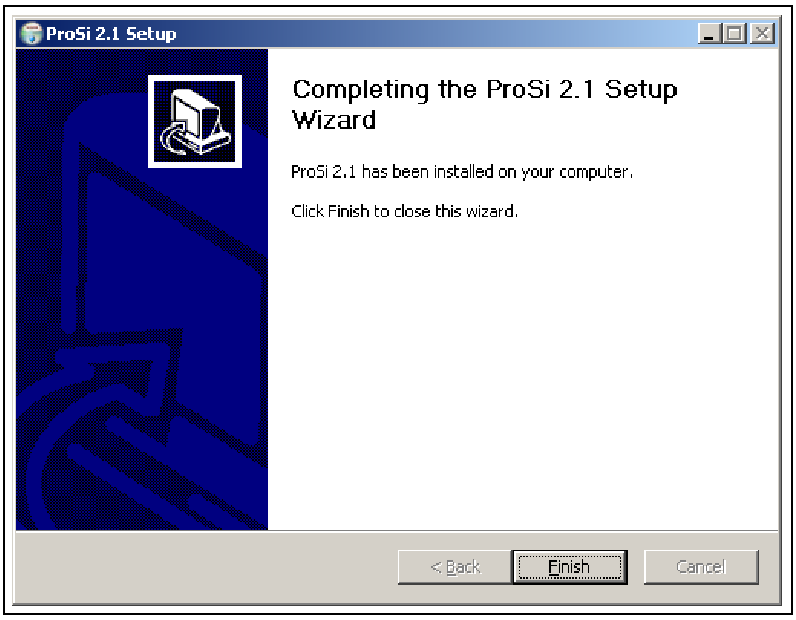
Execute: "C:\Programme\ProSi_2.1\dll\vcredist_x86.exe" /q

```
Extract: calc_traeger_rsm_en.inp... 100%
Extract: calc_traeger_rsmmcs_en.inp... 100%
Extract: traeger.INPUT... 100%
Extract: traeger.OUTPUT... 100%
Extract: execution_definition.phd... 100%
Output folder: C:\ProSi_2.1\input_data\data
Output folder: C:\ProSi_2.1\input_data\output
Output folder: C:\ProSi_2.1\input_data
Output folder: C:\ProSi_2.1\scripts
Output folder: C:\Programme\ProSi_2.1\dll
Extract: vcredist_x86.exe... 100%
Execute: "C:\Programme\ProSi_2.1\dll\vcredist_x86.exe" /q
```

Nullsoft Install System v2.41

Figure 1.7. "Installation Wizard - Installation"-Window

8. Finish Installation



Completing the ProSi 2.1 Setup Wizard

ProSi 2.1 has been installed on your computer.

Click Finish to close this wizard.

Figure 1.8. “Installation Wizard - Finishing”-Window

After successful installation a symbolic link can be found on the desktop. Also your Start Menu will consist of links for ProSi, the manual and the uninstall link.

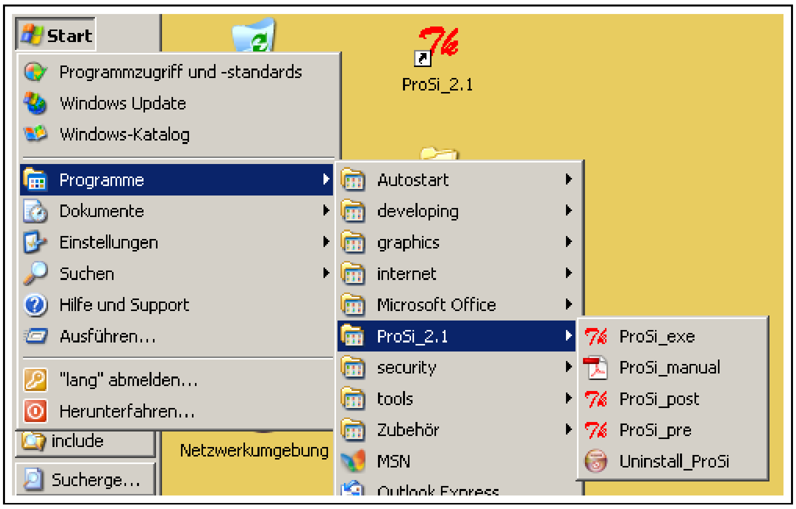


Figure 1.9. “Installation Wizard - Finishing”-Window

Known problems:

- firewall: If you have a firewall, allow ProSi to operate and acting as server. (files: tclkit.exe, tclkitsh.exe)

Further applications

In order to use the image export function in ProSi-Post, it’s necessary to also install ImageMagick and Ghostscript.

Be aware, that due to a bug in the interaction of ImageMagick and Ghostscript both must be installed as 32-bit versions, although a 64-bit system is installed.

- The installation process requires administration permissions.
- Visit the ImageMagick Homepage and download the latest Win32-static version.
- Visit the Ghostscript Homepage and download the latest w32-version.
- Install ImageMagick
- Install GhostScript.

2. Basic theory

2.1. Deterministic model

A deterministic analysis model can be described by the general relation

$$\mathbf{y} = \Phi(\mathbf{x}) \tag{2.1}$$

where $\mathbf{x} = (x_1, x_2, ...)^T$ are deterministic parameters to describe the structural model whereas $\mathbf{y} = (y_1, y_2, ...)^T$ denotes the deterministic structural response. A closed analytical function Φ that relates $\mathbf{x}$ to $\mathbf{y}$ can only be established for simple geometry and load conditions. Instead, numerical methods (e.g. Finite Element Methods) are applied to assess the structural behavior for complex engineering structures.

For all deterministic models which are analysed using ProSi, the deterministic input parameters have to be defined in ASCII-files.

2.2. Probabilistic model

All structural systems operate in random environments. Moreover, the structural systems themselves imply a certain degree of uncertainty. As a consequence, the real-live response of the structure scatters and, dependent on the robustness of the system, this scatter may have significant influence on functionality and reliability. Contrary to this, standard mechanical analysis is based on deterministic models, which relate a specific loading and a specific realization of the structural system (boundary conditions, material properties, geometry) to the structural response. Clearly, the deterministic approach does not take into account the inherent uncertainty, which is characteristic for each real engineering system and therefore an important part of physics is not considered in deterministic simulation models. This behavior might be described by probabilistic design approaches, Figure 2.1.

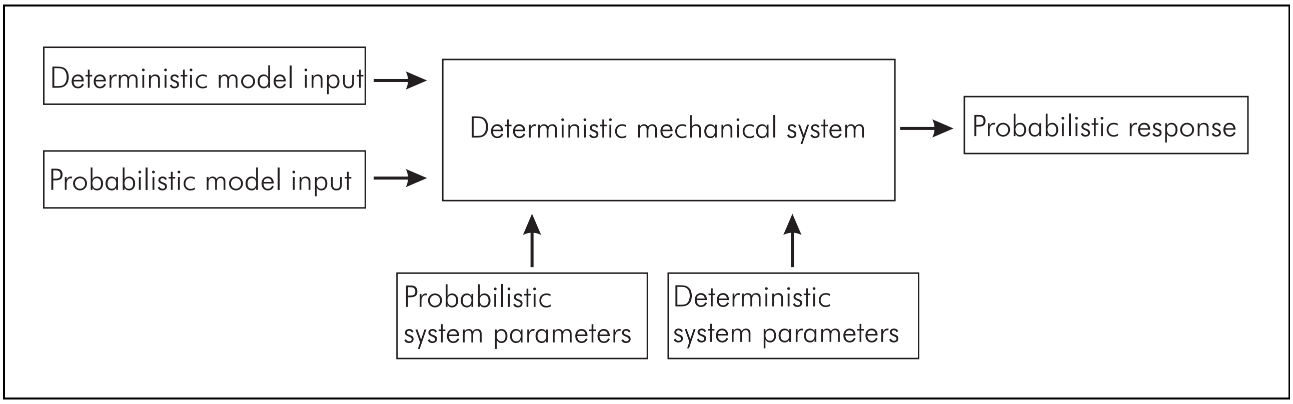

Deterministic model input
Probabilistic model input
Deterministic mechanical system
Probabilistic response
Probabilistic system parameters
Deterministic system parameters

Figure 2.1. Probabilistic model description

2.2.1. Probabilistic variables

In probabilistic analysis the most dominant deterministic parameters are replaced by probabilistic variables

$$\mathbf{b} = (b_1, b_2, ..., b_{n_b})^T \tag{2.2}$$

which are described by appropriate probability distribution functions. Each realization $\mathbf{b}_r$ of the probabilistic variables corresponds to a different set of "deterministic" model input parameters, whereas the entirety of realizations characterizes all possible variations of input data. For the probabilistic procedures used in ProSi, the stochastic variables can be of Gaussian or non-Gaussian type, and may be independent of each other or correlated.

2.2.2. Supported distribution

Normal (Gaussian)

The normal distribution, also called the Gaussian distribution, is a continuous distribution within the co domain of the real numbers. The density function of a normal distribution has a global maximum at $b_M = \mu$ and two inflection points at $b_{W1} = \mu + \sigma$ and $b_{W2} = \mu - \sigma$. The density function

$$f(b) = \frac{1}{\sigma\sqrt{2\pi}} \exp\{-\frac{(b-\mu)^2}{2\,\sigma^2}\} \tag{2.3}$$

and the distribution function

$$F(b) = \frac{1}{\sigma\sqrt{2\pi}} \int\limits_{-\infty}^{b} \exp\{-\frac{(\widetilde{b}-\mu)^2}{2\,\sigma^2}\}\, d\widetilde{b} \tag{2.4}$$

are shown in figure 2.2. The scatters of many material specific variables (e.g.: Young's modulus, expansion coefficient) can be described using the normal distribution.

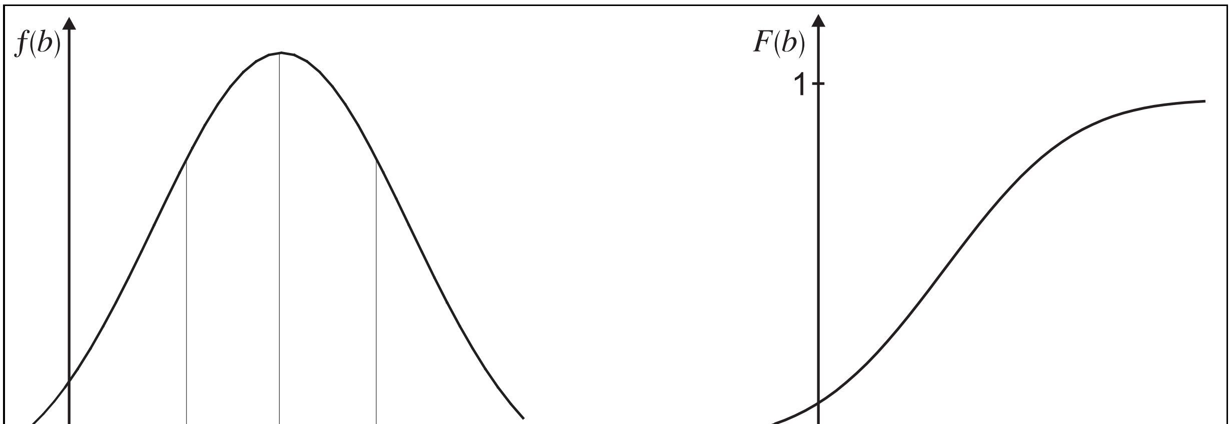


Figure 2.2. Density and distribution function of a normal distribution

The standardized normal distribution represents a special case of the normal distribution according to Gauss with the mean value $\mu = 0$ and the standard deviation $\sigma = 1$. Based on (2.3) the density function of this distribution can be described as

$$f(b) = \frac{1}{\sqrt{2\pi}} \exp\{-\frac{b^2}{2}\}. \tag{2.5}$$

Its distribution function follows from (2.4) as

$$F(b) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{b} \exp\{-\frac{\tilde{b}^2}{2}\} \, d\tilde{b} \; . \tag{2.6}$$

Lognormal

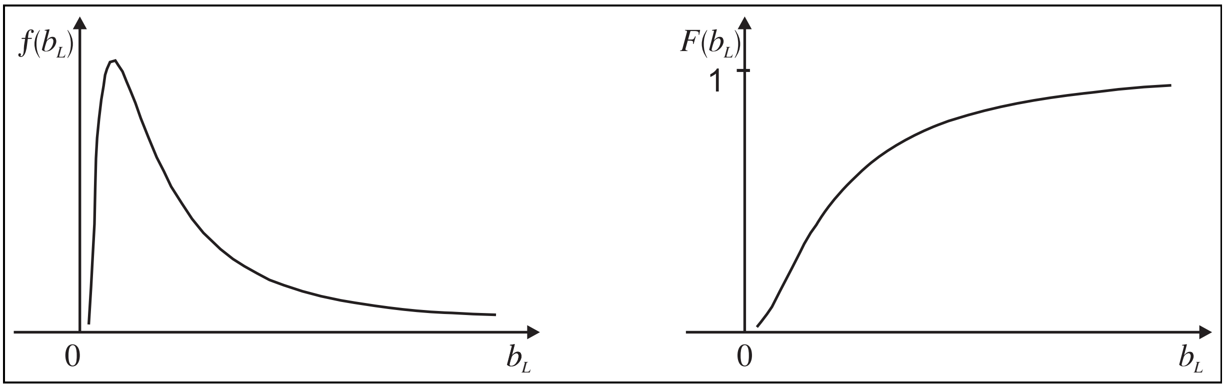


Figure 2.3. Density and distribution function of a lognormal distribution

A stochastic variable b_L is deemed to be distributed lognormal, if a normal distributed variable b can be calculated using the logarithm onto b_L.

$$b = \ln(b_L) \tag{2.7}$$

The density function

$$f(b_L) = \begin{cases} \dfrac{1}{\zeta\sqrt{2\pi}\, b_L} \exp\{-\dfrac{(\ln b_L - \lambda)^2}{2\,\zeta^2}\} & \text{for } b_L > 0, \\ 0 & \text{others} \end{cases} \tag{2.8}$$

and the distribution function

$$F(b_L) = \begin{cases} \dfrac{1}{\zeta\,\sqrt{2\,\pi}} \displaystyle\int_0^{b_L} \dfrac{1}{\tilde{b}_L} \exp\{-\dfrac{(\ln \tilde{b}_L - \lambda)^2}{2\,\zeta^2}\} \, d\tilde{b}_L & \text{for } \tilde{b}_L > 0, \\ 0 & \text{others} \end{cases} \tag{2.9}$$

are shown in Figure 2.3. In similarity to the normal distribution, the value of the distribution function can only be calculated at a defined point b_L by integrating the density function from zero up to the point b_L. An indefinite integration is also not possible here. The scatters of the low-cycle-fatigue-material properties, for example, can be described using a lognormal distribution.

Uniform

The uniform distribution describes a random variable b, which realisations all show the same probability at the interval $[b_l, b_r]$. Beyond this interval the probability density is zero. As the density function is a constant function between b_l and b_r, the distribution function runs linear. The run of both functions is diagrammed in Figure 2.4. For the uniform distribution the following functional relations are applied for the density function

$$f(b) = \begin{cases} \dfrac{1}{b_r - b_l} & b_l \le b \le b_r, \\ 0 & \text{others} \end{cases} \tag{2.10}$$

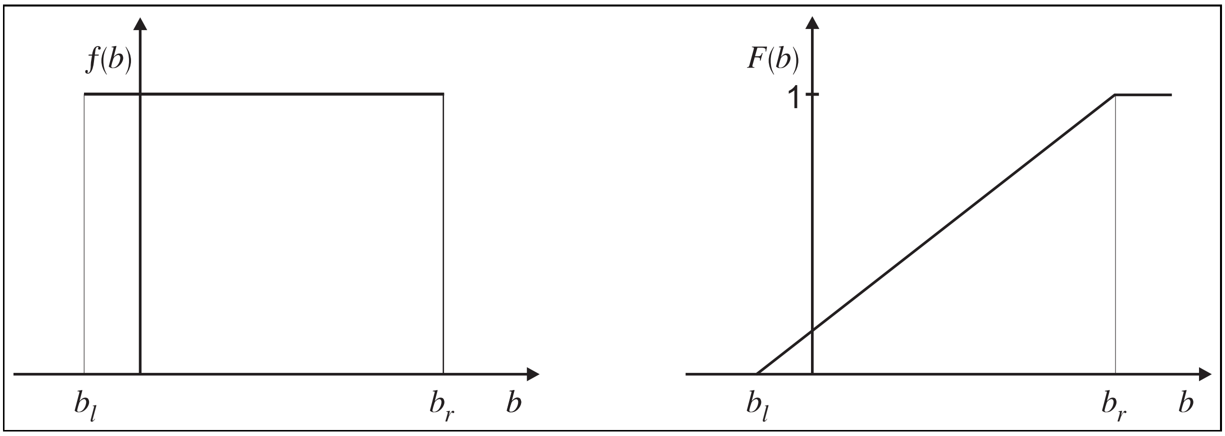


Figure 2.4. Density and distribution function of an uniform distribution

and

$$F(b) = \begin{cases} 0 & b < b_l \\ \dfrac{b - b_l}{b_r - b_l} & b_l \leq b \leq b_r \\ 1 & b > b_r \end{cases} \tag{2.11}$$

for the distribution function. The uniform distribution can be used, for example, to describe scatters of the geometry, resulting out of manufacturing tolerances.

Triangular

The triangular distribution describes a random variable which values represent a linearly distributed probability density within the interval $[b_l, b_r]$. The point c represents a maximum and a discontinuity. Beyond the interval the probability density is zero. As the density function is a linear function, the distribution function has a quadratic characteristic. The two functions are shown in Figure 2.5. Due to the discontinuity at c, two functions are needed for the description, which are referred to as f_l and F_l (on the left side of c) and f_r and F_r (on the right side of c) here. Therefore the density function

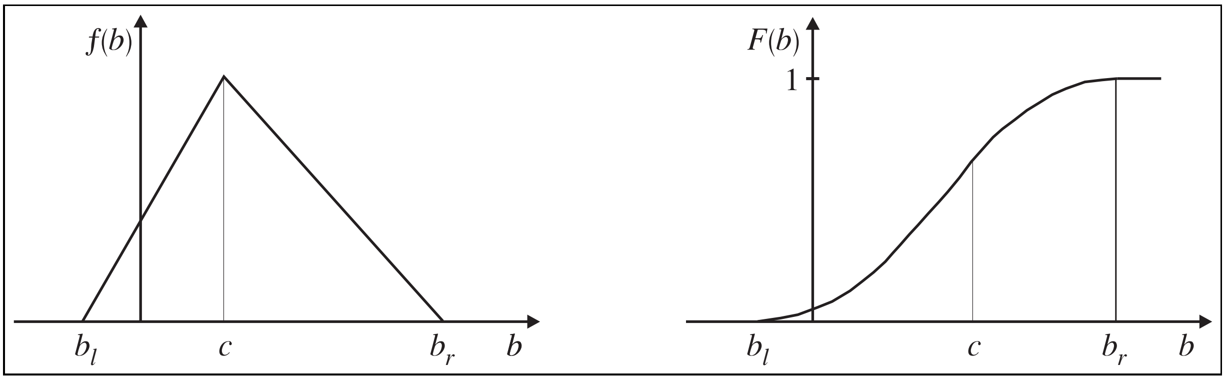


Figure 2.5. Density and distribution function of a triangular distribution

can be described as

$$f(b) = \begin{cases} f_l(b) = \dfrac{2\,(b-b_l)}{(b_r-b_l)\,(c-b_l)} & b_l \leq b < c, \\ f_r(b) = \dfrac{2\,(b_r-b)}{(b_r-b_l)\,(b_r-c)} & c \leq b \leq b_r, \\ 0 & \text{others} \end{cases} \tag{2.12}$$

and the distribution function as

$$F(b) = \begin{cases} 0 & b < b_l \\ F_l(b) = \dfrac{b^2 + b_l^2 - 2\,b_l\,b}{(b_l-b_r)\,(b_l-c)} & b_l \leq x < c \\ F_r(b) = \dfrac{b^2 - 2\,b_r\,b + b_r\,c + b_l\,(b-c)}{(b_l-b_r)\,(b_r-c)} & c \leq b \leq b_r \\ 1 & b > b_r \end{cases} \tag{2.13}$$

A triangular distribution arises if a stochastic input parameter is the sum of two uniform distributed input parameters.

Exponential

The exponential distribution is a family of density functions, which is given by an exponential function. It can be used to describe the random duration between two events, like the lifetime of atoms or components.

The density function

$$f(b) = \begin{cases} \lambda \exp\left(-\lambda x\right) & \text{for } b \geq 0, \\ 0 & \text{others} \end{cases} \tag{2.14}$$

and the distribution function

$$F(b) = \begin{cases} 1 - \exp\left(-\lambda x\right) & \text{for } b \geq 0, \\ 0 & \text{others} \end{cases} \tag{2.15}$$

are shown in Figure 2.6. The rate parameter λ can alternatively be replaced by the scale parameter $\mu = 1/\lambda$, which is the mean value too.

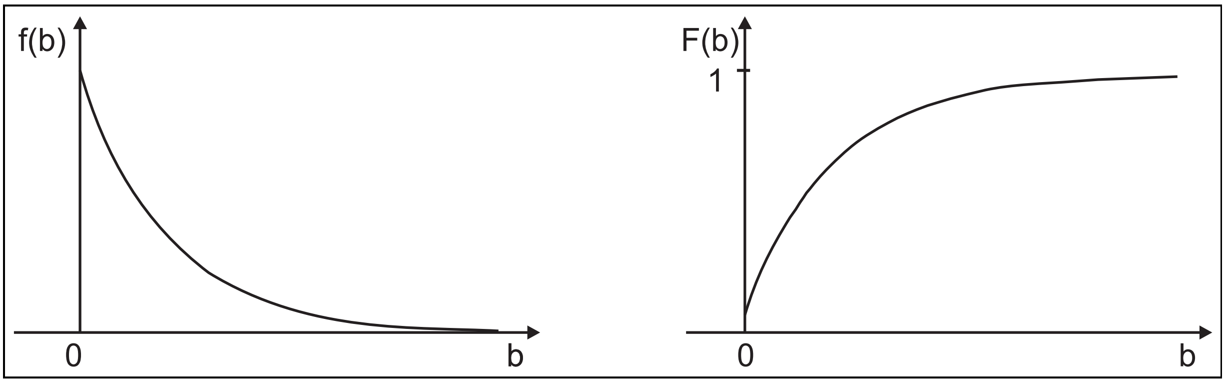


Figure 2.6. Density and distribution function of an exponential distribution

Weibull

The Weibull distribution is a function, with the scale parameter α and the shape parameter β. The density function is

$$f(b) = \begin{cases} \frac{\beta}{\alpha}\left(\frac{b}{\alpha}\right)^{\beta-1} \exp\{-\left(\frac{b}{\alpha}\right)^{\beta}\} & \text{for } b \geq 0, \\ 0 & \text{others} \end{cases} \tag{2.16}$$

and the distribution function is

$$F(b) = \begin{cases} 1 - \exp\{-(b/\alpha)^{\beta}\} & \text{for } b \geq 0, \\ 0 & \text{others} \end{cases} \tag{2.17}$$

In contrary to the exponential distribution, the Weibull distribution has a variable rate parameter, due to the extra parameter β. Figure 2.7 shows both functions.

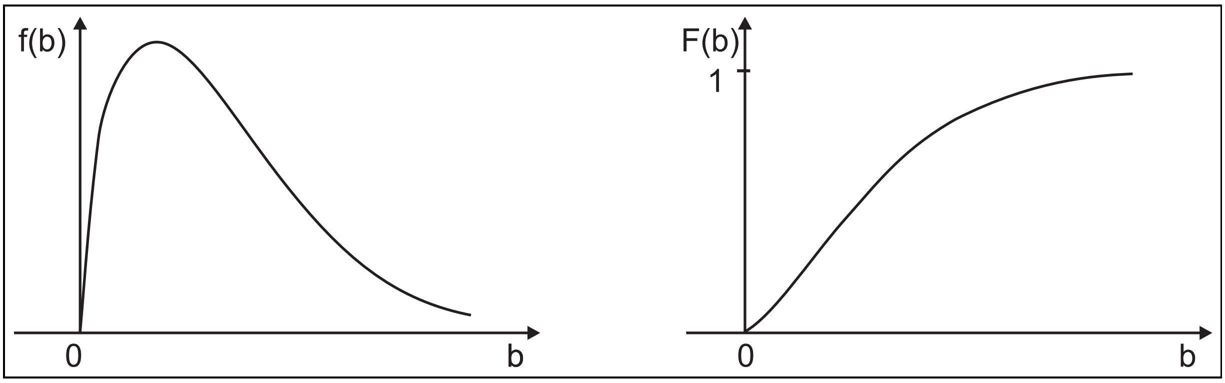


Figure 2.7. Density and distribution function of a Weibull distribution

2.3. Probability of failure

Probabilistic methods for component lifetime assessment require a mathematical definition of a failure criterion. To this end, a failure function $g(\mathbf{b}, \mathbf{y})$ is introduced to give

$$g(\mathbf{b}, \mathbf{y}) \leq 0 \tag{2.18}$$

for a failing design and

$$g(\mathbf{b}, \mathbf{y}) > 0 \tag{2.19}$$

for a safe design. The probability of failure is then defined as

$$P_f = P[g(\mathbf{b}, \mathbf{y}) \leq 0] = \int\limits_{\underline{\mathbf{b}} \;\;|\;\; g(\mathbf{b},\mathbf{y}) \leq 0} f(\mathbf{b}) db_1 db_2 ... db_{n_b} \tag{2.20}$$

where P_f denotes the probability that the failure criterion $g(\mathbf{b}, \mathbf{y}) \leq 0$ is violated and $\underline{\mathbf{b}}$ describes the n_b-dimensional space of the input variables $b_1, b_2, ..., b_{n_b}$.

The specific form of the failure function depends on the problem to be solved. For lifetime assessment of turbine parts, the failure functions are typically formulated in terms of stress, temperature, creep strain, cycles to crack initiation or eigenfrequencies.

There are different approaches to evaluate the integral in equation (2.20). Amongst them are the direct Monte Carlo Simulation (MCS) and the RSM procedures. These two methods are briefly described in the following two sections.

2.4. Direct Monte Carlo Simulation

Using the direct Monte Carlo Simulation method (see Figure 2.8), the probability of failure is determined by the following steps:

1. A random number generator produces realizations $\mathbf{b}_r$ of the stochastic variables $\mathbf{b}$ which corresponds to an user-defined probability distribution function.

2. The structural analysis program evaluates the failure function $g(\mathbf{b}, \mathbf{y})$ for each set of realizations.

3. An indicator function

$$I(g(\mathbf{b}_r, \mathbf{y})) = \begin{cases} 1, & \text{for} \quad g(\mathbf{b}_r, \mathbf{y}) \leq 0 \\ 0, & \text{for} \quad g(\mathbf{b}_r, \mathbf{y}) > 0 \end{cases} \tag{2.21}$$

   is applied to differ between failing and safe realizations.

4. The probability of failure is approximated by

$$P_f = \frac{1}{n_{sim}} \sum_{r=1}^{n_{sim}} I(g(\mathbf{b}_r, \mathbf{y}) \leq 0) = \frac{n_f}{n_{sim}} \tag{2.22}$$

   where n_{sim} denote the total number of simulations and n_f the number of simulations which violated the failure criterium respectively.

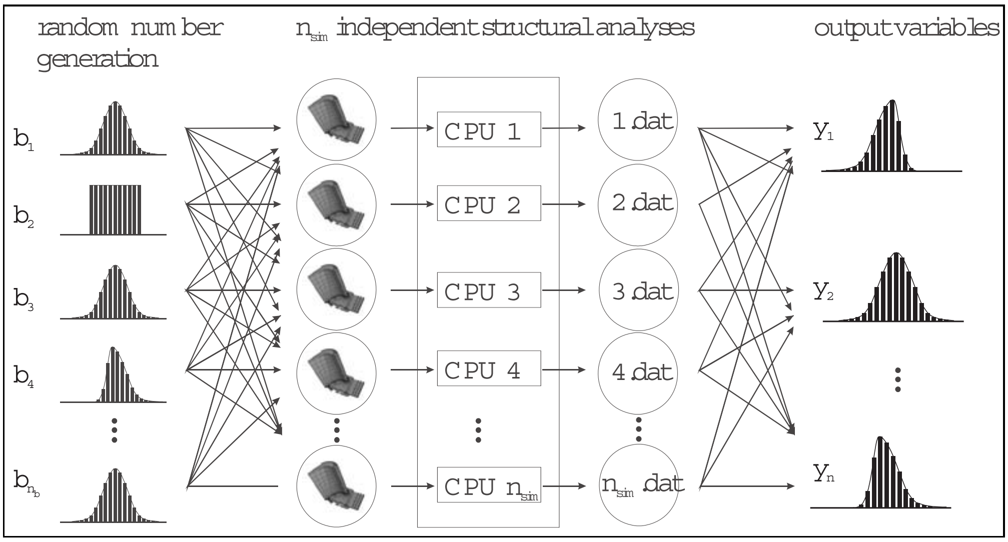


Figure 2.8. Direct Monte Carlo Simulation scheme

2.5. Sampling methods

There are different methods for generating the realisations of the random variables. The three methods Simple Random Sampling (SRS), Latin Hypercube Sampling (LHS) and Descriptive Sampling (DS), implemented in ProSi will be described briefly in the following sections.

2.5.1. Congruential generator

A method widely used for the numerical generation of random numbers are random number generators based on the multiplicative congruence equation of the number theory. This congruence method is based on the equation

$$B_{n+1} = (a * B_n + c) \bmod M \quad n = 1, ..., n_{sim} \ , \tag{2.23}$$

where M should be a large, nonnegative integer. Furthermore a is a number between 1 and $M-1$, coprime to M and c is a nonnegative integer which has to be smaller than M. As initial value B_0 (the so called seed) a number between zero and $M-1$ has to be chosen. By normalizing the random numbers using

$$b_{r,n} = \frac{B_n}{M} \tag{2.24}$$

a uniform distribution containing n_{sim} random numbers within the interval $[0, 1[$ can be generated. In ProSi the congruential generator is implemented using the parameters $a = 1103515245$, $c = 12345$ and $M = 2147483648$. These parameters align with the ones used by recent C-compilers. With these parameters a repetition of the generated random numbers occurs after 2147483648 numbers. This behavior is uncritical as such a large amount of random numbers is not needed for the analysis of technical problems in general. Further information on these and other types of random number generators can be found in [?], for example.

2.5.2. Simple Random Sampling

The Simple Random Sampling is the classic method of generating random numbers. The random numbers generated by the congruential generator are transformed into the needed random distribution here. The realisations of uniform distributed random variables can be calculated directly by transforming the random numbers delivered by the generator to match the interval needed. Thus it is not possible to generate others than uniformly distributed random numbers using the congruential generator, the acceptance-rejection method is used to transform the random numbers into other distributions. A further description of this method can be found in [?] or [?] for example. The number of deterministic simulations n_{sim} needed for the description of a certain probability of failure P_f can be estimated as

$$n_{sim} \approx \frac{100}{\widetilde{P}_f} \tag{2.25}$$

while using the Simple Random Sampling.

2.5.3. Latin Hypercube Sampling

The Latin Hypercube Sampling uses the fact, that for a discrete probability density function all realisations represent the same probability. Figure 2.9 shows the principal approach of the LHS-method. The procedure of the LHS is as following:

- Definition of the number of samples n_{sim}
- Dividing the distribution function in strips of the width $1/n_{sim} = \Delta P$ on the $F(\mathbf{b})$-axes
- Determination of the width of the strips on the $\mathbf{b}$-axes
- Generation of on sample for each strip using a uniform distribution

Due to the splitting of the distribution function and the assignment of the samples to the different strips, the distribution function can be described well across its domain by a small number of samples. Caused by the method the order of the samples is defined by their value and not randomly. Therefor a

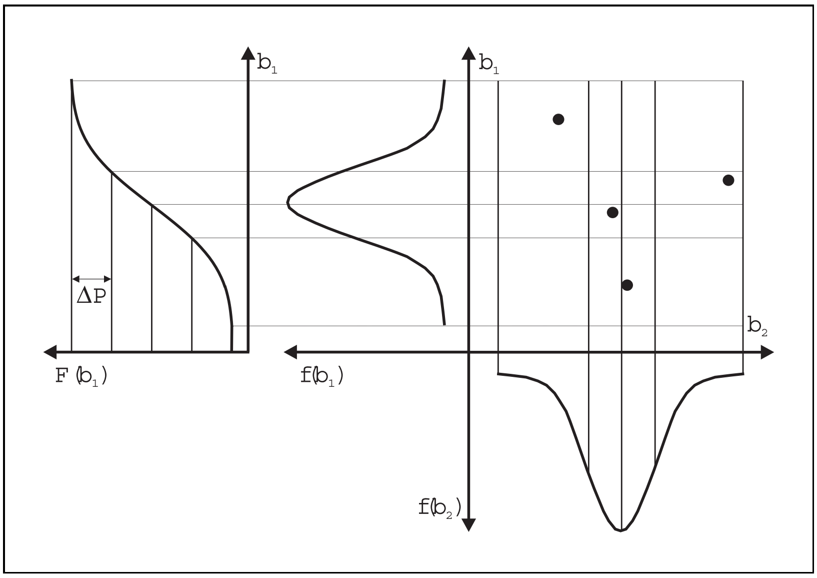


Figure 2.9. Latin Hypercube Sampling

rearrangement of the samples has to be done in an additional step in a way which guarantees that the correlation between the single distributions generated does not exceed the correlation allowed. As the density function of the random variable is described more exactly while using the LHS, the number of deterministic shots n_{sim} needed to calculate a certain probability of failure P_f is smaller than the one needed while using the SRS, and can be estimated as

$$n_{sim} \approx \frac{10}{\widetilde{P}_f} \tag{2.26}$$

for the Latin Hypercube Sampling.

2.5.4. Descriptive Sampling

The Descriptive Sampling is a sampling method which exactly calculates the position of each sample. The DS is based on the same procedure as the LHS. The only difference is, that the realisations for every strip are not generated using a random distribution, but are positioned in the middle of the each strip. ProSi even goes one step further and places the samples in the centre of area of the strips, which results in an even better approximation quality. The samples generated by the DS are completely reproducible due to the assignment of the sample points. A rearrangement of the single realisations is done similar to the LHS in order to suppress correlations between the generated distributions. The number of realisations and deterministic calculations can be estimated using equation (2.26), in the same way as for the LHS.

2.5.5. Evaluation and field of application of the sampling methods

The advantages of the SRS are its high working speed and the simple feasibility in programming. The main drawback of this method is the, compared to the other methods, bad quality of approximation, which causes the number of deterministic simulations needed to increase. Figure 2.10 shows three discrete realisations of a random variable with a deviation of one and a mean value of zero, containing of 200 samples each. It shows, that the realisations generated using the SRS do not reflect the density function well. The realisations generated using the LHS and the DS in contrast show a god alignment to the density function all-over the domain. This is the reason why the LHS and the DS allow it to describe all possible conditions of the system with few samples.

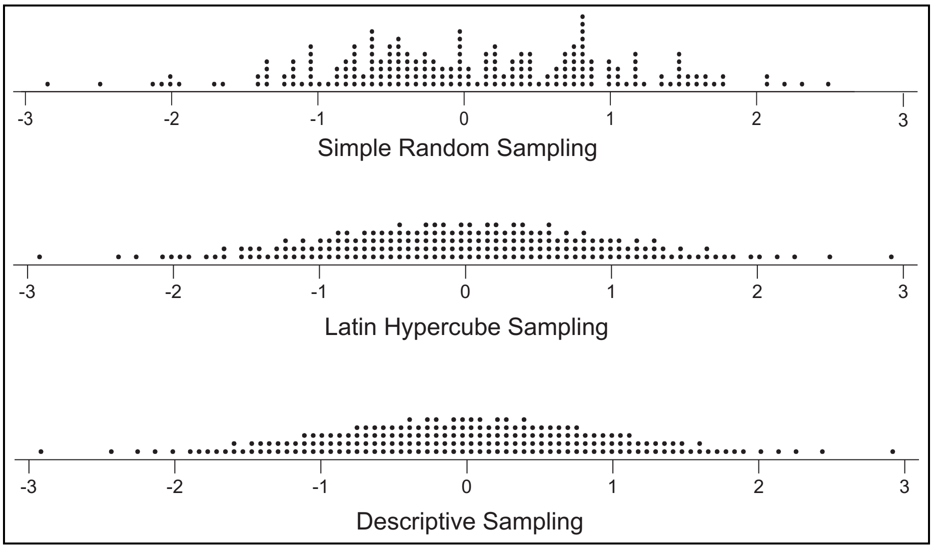


Figure 2.10. Sampling method comparison

2.6. Response Surface Method

The RSM consists of deterministic calculations, which are systematically carried out according to a “numerical” test plan. The test plan can be determined by an appropriate Design of Experiment (DoE) approach (see 2.6.1). The results of the deterministic analyses are used to evaluate an approximate failure function $\overline{g}(\mathbf{b}, \mathbf{y})$. To achieve a high quality of approximation, the center of the test plan should be located nearby the failure surface $\overline{g}(\mathbf{b}, \mathbf{y}) = 0$, Figure 2.11. Since the failure surface follows from the results of the analysis, an iteration is required to shift the center of the test plan at the appropriate position.

Different shape functions can be used to approximate the failure surface. In most cases a quadratic polynomial function

$$\overline{g} = c_0 + \sum_{i=1}^{n_b} c_i b_i + \sum_{i=1}^{n_b} \sum_{j \geq i}^{n_b} c_{ij} b_i b_j + \varepsilon \tag{2.27}$$

is applied, which can be represented in matrix notation as

$$\mathbf{y} = \mathbf{BC} + \boldsymbol{\epsilon}. \tag{2.28}$$

For two stochastic variables the matrix is

$$\mathbf{B} = \begin{bmatrix} 1 & b_{11} & b_{21} & b_{11}^2 & b_{21}^2 & b_{11}b_{21} \\ 1 & b_{12} & b_{22} & b_{12}^2 & b_{22}^2 & b_{12}b_{22} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & b_{1r} & b_{2r} & b_{1r}^2 & b_{2r}^2 & b_{1r}b_{2r} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & b_{1n_{sim}} & b_{2n_{sim}} & b_{1n_{sim}}^2 & b_{2n_{sim}}^2 & b_{1n_{sim}}b_{2n_{sim}} \end{bmatrix}$$

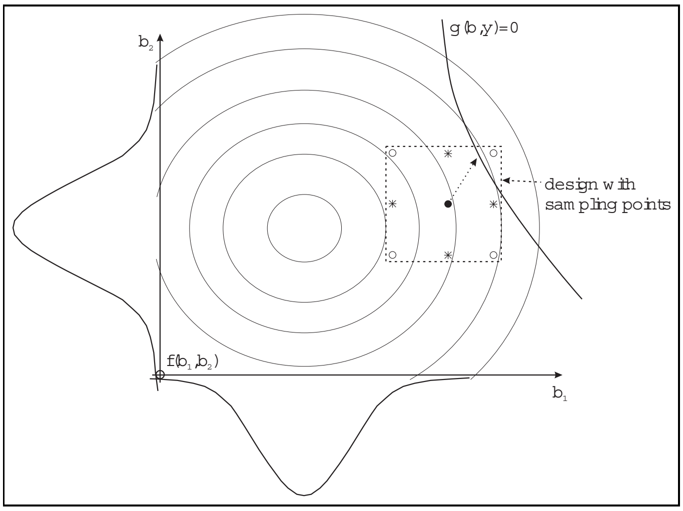


Figure 2.11. Response Surface for two stochastic variables

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n_{sim}} \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_{11} \\ c_{22} \\ c_{12} \end{bmatrix} \quad \boldsymbol{\epsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_{n_{sim}} \end{bmatrix}$$

where index r, ranging from 1 to n_{sim}, denotes specific values of the stochastic variables.
At least $n_{sim} = 1 + 2n_b + n_b(n_b - 1)/2$ structural analyses are necessary to determine the unknown coefficients in $\mathbf{C}$. If the number of structure simulations is higher, a least square fitting procedure

$$S = (\mathbf{y} - \mathbf{BC})^T(\mathbf{y} - \mathbf{BC}) = \boldsymbol{\epsilon}^T\boldsymbol{\epsilon} \Rightarrow \min \tag{2.29}$$

may be applied to determine the parameters of the failure surface. From the condition of minimum

$$\frac{\partial S}{\partial \mathbf{C}} = 2\mathbf{B}^T(\mathbf{y} - \mathbf{BC}) = 0 \tag{2.30}$$

the unknown coefficients can be determined by

$$\mathbf{C} = \left(\mathbf{B}^T\mathbf{B}\right)^{-1}\mathbf{B}^T\mathbf{y}. \tag{2.31}$$

The quality of the response surface may be controlled by the significance test. Thereby, the coefficient of determination

$$R^2 = \frac{S_R}{S_y} \tag{2.32}$$

is evaluated from the regression sum of squares

$$S_R = \mathbf{C}^T\mathbf{B}^T\mathbf{y} - \frac{\left(\sum_{i=1}^{n_{sim}} y_i\right)^2}{n_{sim}} \tag{2.33}$$

and the total sum of squares

$$S_y = S_R + S_E \tag{2.34}$$

with the sum of error or residual

$$S_E = \mathbf{y}^T\mathbf{y} - \mathbf{C}^T\mathbf{B}^T\mathbf{y}. \tag{2.35}$$

The value of R^2 ranges from 0 to 1. A value of 1 indicates that the approximate response surface passes through all results of simulations.
Using the approximate failure function of equation (2.27), the probability of failure is estimated by

$$\begin{aligned} P_f &= P[g(\mathbf{b},\mathbf{y}) \leq 0] \approx P[\overline{g}(\mathbf{b},\mathbf{y}) \leq 0] \\ P[\overline{g}(\mathbf{b},\mathbf{y}) \leq 0] &= \int\limits_{\underline{\mathbf{b}} \ | \ \overline{g}(\mathbf{b},\mathbf{y}) \leq 0} f(\mathbf{b}) db_1 db_2 ... db_{n_b}. \end{aligned} \tag{2.36}$$

Note that no further FEA is required to obtain the failure probability P_f. Instead, the integral (2.36) can be estimated by using the analytical approximation of the failure criterion together with a Monte Carlo based integration scheme.

2.6.1. Designs

The deterministic calculations necessary for the rsm run are calculated systematically following a design. There are different requirements for these designs. These requirements arise from the deterministic system and the failure surface. Furthermore the design has to provide enough sampling points (see table 2.1).

2^{n_b} Design

The simplest design, which is supported by the RSM modul included in ProSi, is the 2^{n_b} Design. With this design 2^{n_b} sampling points are generated which represent all possible combinations of the n_b variables. Using such a design, all main and correlation effects can be differentiated and determined. The primary drawback of this design is its inefficiency. The number of sampling points needed increases exponentially with 2^{n_b}.

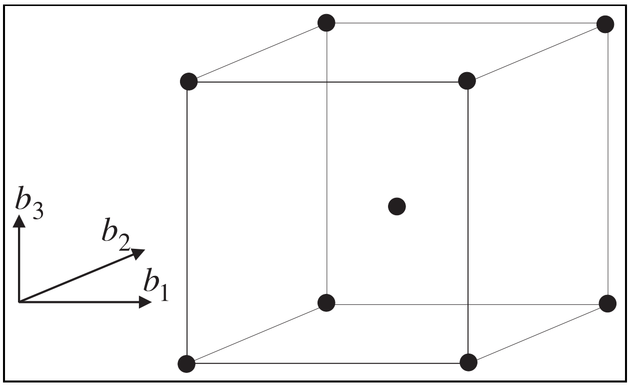


Figure 2.12. 2^3 **Design**

Central Composite Design

An alternative design to the 2^{n_b} design is the Central Composite Design (CCD). The CCD is an efficient design for fitting second-order response surfaces. As shown in Figure 2.13, the CCD is composed of a full factorial and a part factorial design. This part factorial design consists of so called star-points, which are arranged around the central point with the distance α. Using $\alpha = \sqrt[4]{n_b}$ it can be guaranteed that the design is rotatable.

Box Behnken Design

Another design for the analysis of quadratic coherences has been developed by Box and Behnken. In this design, which is shown in Figure 2.14, the sampling points are situated in the mid of the edge. Thus extreme combinations of variables are excluded. The advantage of the Box-Behnken-design compared to the CCD is the lower number of sampling points needed. This advantage is purchased at the cost of the orthogonality of the design. An overview of the number of sampling points needed for polynomials of different order and of the number of sampling points provided by the single designs is given in Table 2.1.

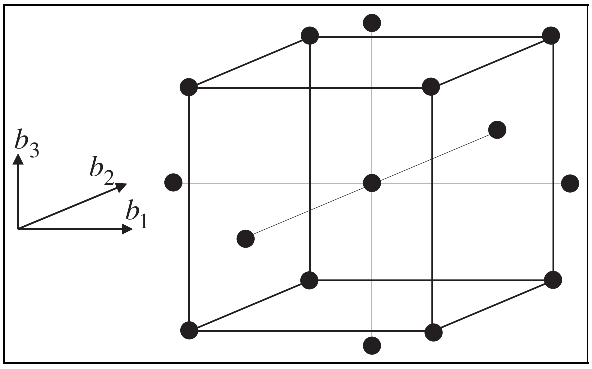


Figure 2.13. Central Composite Design

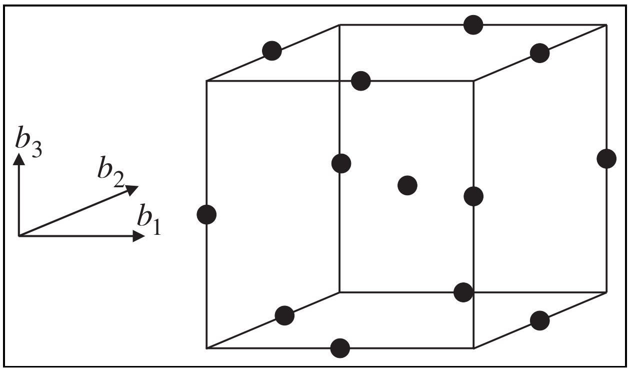


Figure 2.14. Box Behnken Design

Variables	Polynom			Simplex	2^{n_b} Design	CCD	BBD
	1. order	2. order	3. order				
1	2	3	4				
2	3	6	10	3	5	9	
3	4	10	20	4	9	15	13
4	5	15	35	5	17	25	25
5	6	21	56	6	33	43	41
6	7	28	84	7	65	77	49
7	8	36	120	8	129	143	57
8	9	45	165	9	257	273	113
9	10	55	220	10	513	531	145
10	11	66	286	11	1025	1045	161
11	12	78	364	12	2049	2071	177
12	13	91	455	13	4097	4121	193
13	14	105	560	14	8193	8219	209
14	15	120	680	15	16385	16413	225
15	16	136	816	16	32769	32799	

Table 2.1. Comparison of the number of sampling points needed and the number of sampling points provided by the designs

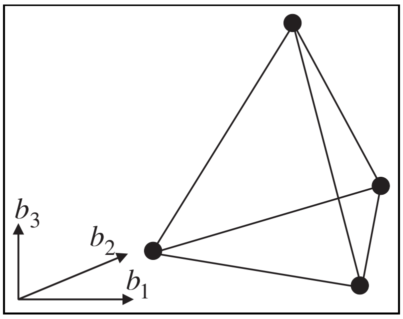


Figure 2.15. Simplex Design

Simplex Design

The last design implemented is the Simplex Design. It is shown in Figure 2.15. This design can only be used to capture the reaction of linear systems. Using this design, at least $n_{sim} = (n_b + 1)$ deterministic analyses are necessary to determine the unknown coefficients in C.

2.6.2. Adaptive RSM

The least square method minimizes the error of approximation $\boldsymbol{\epsilon}$ based on a function containing polynomials. The structure of this function (e.g. order, number of polynomials...) has to be previously provided by the user. In traditional RSM, this structure remains constant. The error of approximation can only be minimized through the specific coefficients $\mathbf{C}$.
Now the adaptive RSM executes additional calculations to further minimize the error of approximation. It is trying to fit the approximate function closer to the system which is to be approximated. The approximate function is altered using two methods.
First, the $\mathbf{y}$-values of the specific sampling points are being manipulated with the help of the Box-Cox-transformation [?]. This can be done by exponentiating the approximate function

$$y^x \approx \tilde{g}^x = \left(c_0 + \sum_{i=1}^{n_b} c_i b_i + \sum_{i=1}^{n_b} \sum_{j=1}^{n_b} c_{ij} b_i b_j + \epsilon \right)^x \tag{2.37}$$

with a variable x. Afterwards the error of approximation $\boldsymbol{\epsilon}$ is minimized through variation of x and through applying of the least square method. This method is called "Automatic improvement" of the RSM within ProSi. An examples of this method is shown in figures 2.16 and 2.17. The quadratic function $y = x^2$ results in a linear function after exponentiating the y-values using 1/2. This linear function can be approximated more accurately using the least square method.

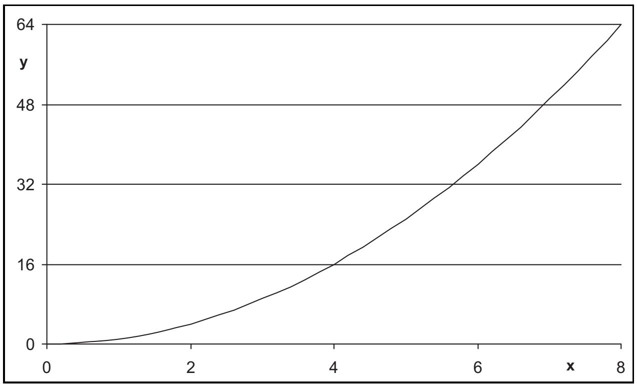


Figure 2.16. Original y-Axis

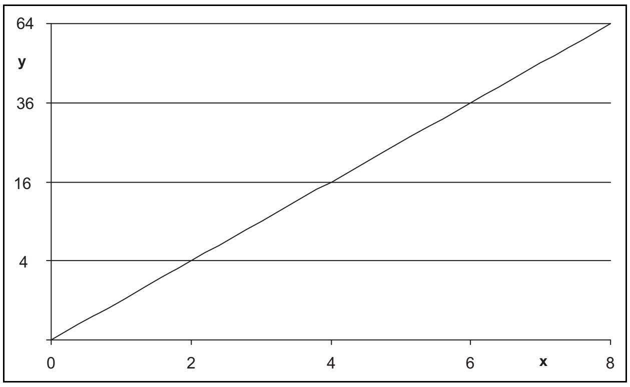


Figure 2.17. y-Axis scaled

The second method to improve the response surface is not done through minimizing the error of approximation, but through the coefficients of the polynomials, which have no or only little influence on the system response y. Those coefficients are identified and excluded from the process of minimizing the error of approximation $\boldsymbol{\epsilon}$ using the least square method. By doing this, it is possible to reduce the numeric error in the application of the RSM and also to simplify the response surface equation. This method is called "Automatic reduction" of the RSM within ProSi.

3. Graphical user interface

3.1. ProSi Pre

3.1.1. Introduction

ProSi Pre has been developed as a part of the ProSi product family. It is used as the graphical interface to create valid input files for ProSi. In that it generates the probabilistic model and directs ProSi in modifying the deterministic model as needed.

3.1.2. Requirements

To use ProSi Pre, one needs data files for the deterministic model of the object of interest. This manual will use a simple deflection of a beam under load (Figure 3.1) as an example to go through the entire process of using the program. To actually calculate the deflection, a separate program is used. In this case it is a simple routine that will use the following data for input:

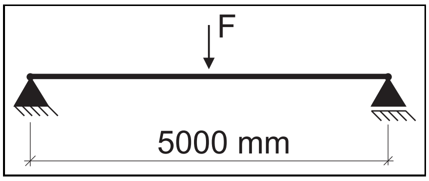


Figure 3.1. Beam Under Load

- the acting force
- Young's Modulus
- the height of the beam
- the width of the beam
- the length of the beam

The goal of ProSi Pre is to create an input-file for ProSi so that any of the above inputs can be changed by ProSi.

3.1.3. Starting ProSi Pre

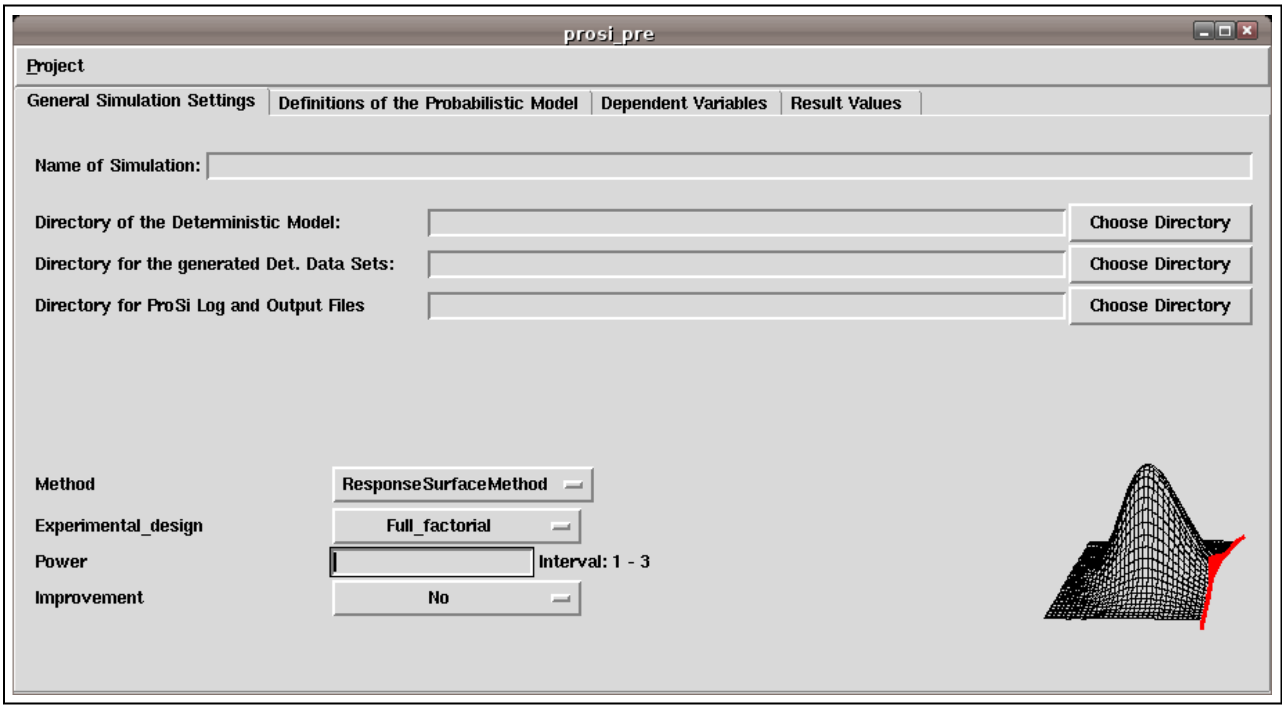
on Linux

To start ProSi Pre change into the directory where ProSi Pre is located. Now type ./tclkit prosi_pre.kit and ProSi Pre will open. If an existing *.inp-file is to be altered one may directly open it during the startup process by appending the file name to the startup syntax, for example "./tclkit prosi_pre.kit example.inp".

on WINDOWS

To start ProSi Pre click on START - PROGRAMS - ProSi_2.3 - ProSi_pre or use the icon on you desktop with a doubleclick.

3.1.4. General Simulation Settings



Name of Simulation:

Directory of the Deterministic Model:

Directory for the generated Det. Data Sets:

Directory for ProSi Log and Output Files

Method ResponseSurfaceMethod

Experimental_design Full_factorial

Power Interval: 1 - 3

Improvement No

Figure 3.2. "General Simulation Settings"-Tab at start

When starting ProSi Pre for the first time, the program opens the "General Simulation Settings"-Tab by default (Figure 3.2). The first thing to do is to give the simulation a name. Enter it into the respective field. The next step is to define the directories ProSi will use. Two types of directory-definition are available. They may be defined as relative-directories or absolute-directories.

Absolute directories work as follows: ProSi will always look for the specified path beginning in the root-directory of the system. The advantage of this is that the ProSi files may be copied to a different directory or even computer and ProSi will still use the same structure of directories. To use a definition like this, merely type the path into the respective field beginning with a "/". Note that if using the "Choose Directory"-Button, the same procedure will be applied. ProSi will be given an absolute directory. Using this method, one can be sure that the .inp-file may be located anywhere on the computer and ProSi will still use one and the same directory-structure.

A relative directory simply means, that ProSi will create the specified directories within the directory of the ProSi *.inp-file. This is useful if an absolute directory-structure could not be maintained over different computers, for example if it contained usernames. The easiest form of a relative directory is to simply type a dot into the field. This will order ProSi to stay within the directory the original by ProSi Pre created .inp-file is located in. To create a relative directory that uses names, just type the desired name into the field without any prefixed symbols. In any case, the location of the original .inp-file will define the complete structure of any relative directory. Be aware that if a "/" is put in front of the directory to be created, ProSi will read it as an absolute directory.

Now to the actual definition of the directories.

The first one is the deterministic model. The here defined directory will contain all the necessary files that ProSi will be given to work with. The directory may be defined as a specific path using either one of the above described procedures.
Now define the directory for the probabilistic model (Directory for the gen. Det. Data Sets). ProSi will use this directory later on, to save the changed model data in it. Again, to define the directory, use one of the above methods. In the example all data files will be saved to the subdirectory "data/.".
The last directory is being used by ProSi to save output-data like merely random numbers and logfiles. No actual interactions with the deterministic model have taken place in the files within this directory. Again, choose a specific path or enter a directory within the ProSi Pre-file directory. In the example "output" has been entered here.
Note that ProSi will create the two directories for "data" and "output" automatically, if they are not present yet. They don't have to be created beforehand.

Next is to define the method of simulation. There are 3 possible choices.

- Monte Carlo Simulation
- Response Surface Method
- RSM+MCS combined

In order to give a compact, but complete view of the program, the method of using both, Monte Carlo Simulation (MCS) and Response Surface Method (RSM), has been chosen for the example in this manual. Using this method, ProSi will first run a RSM in a specific area of interest. After this a Monte Carlo Simulation will be applied to this specific area. MCS will now not use the original deterministic model, but will use the approximated response-surface to calculate the system-answers for the single samples of the MCS.
If just using either one of the mentioned methods, the options to define the other method will not appear. The procedures will remain the same. The mathematical background of these methods is described in chapter 2.
The first two input fields deal with the MCS. Simply enter the amount of shots into the appropriate field. This defines how many altered clones of the deterministic model ProSi will generate. Now pick the Sampling-Method of choice. There are 3 possible Sampling-Methods:

- Latin-Hypercube-Sampling LHS
- Simple-Random-Sampling SRS
- Descriptive Sampling DS

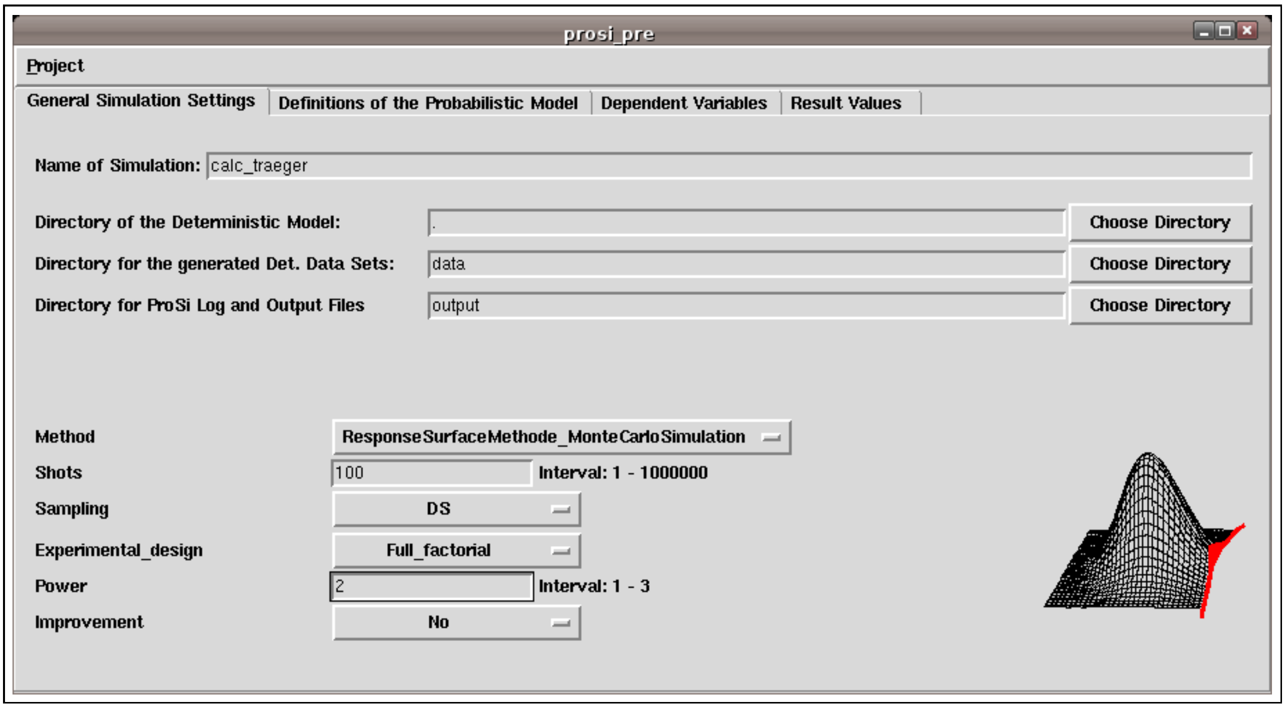
The actual differences of the Sampling-Methods are described in section 2.5 of this manual.
The next 3 input-fields deal with the Response Surface Method. First is the choice of the Experimental Design:

- Full Factorial
- Central Composite Design CCD
- Box Behnken
- Simplex

The description of these design-variants can be found in section 2.6.1 of this manual.

The second input for the RSM is to which power ProSi is supposed to calculate. This defines the order of the polynom that will be calculated. It may range from 1 to 3, providing a maximum of a third order polynom.

Finally set whether ProSi should automatically reduce the Response Surface. By activating this feature, ProSi will check how strong the influence of individual coefficients on the Response Surface is and then automatically set weak coefficients to zero, thus simplifying the entire formula for the Response Surface. The "General Simulation Settings"-Tab is now completed (see Figure 3.3) and one may move on to the next tab.



Name of Simulation: calc_traeger

Directory of the Deterministic Model: .

Directory for the generated Det. Data Sets: data

Directory for ProSi Log and Output Files output

Method ResponseSurfaceMethode_MonteCarloSimulation

Shots 100 Interval: 1 - 1000000

Sampling DS

Experimental_design Full_factorial

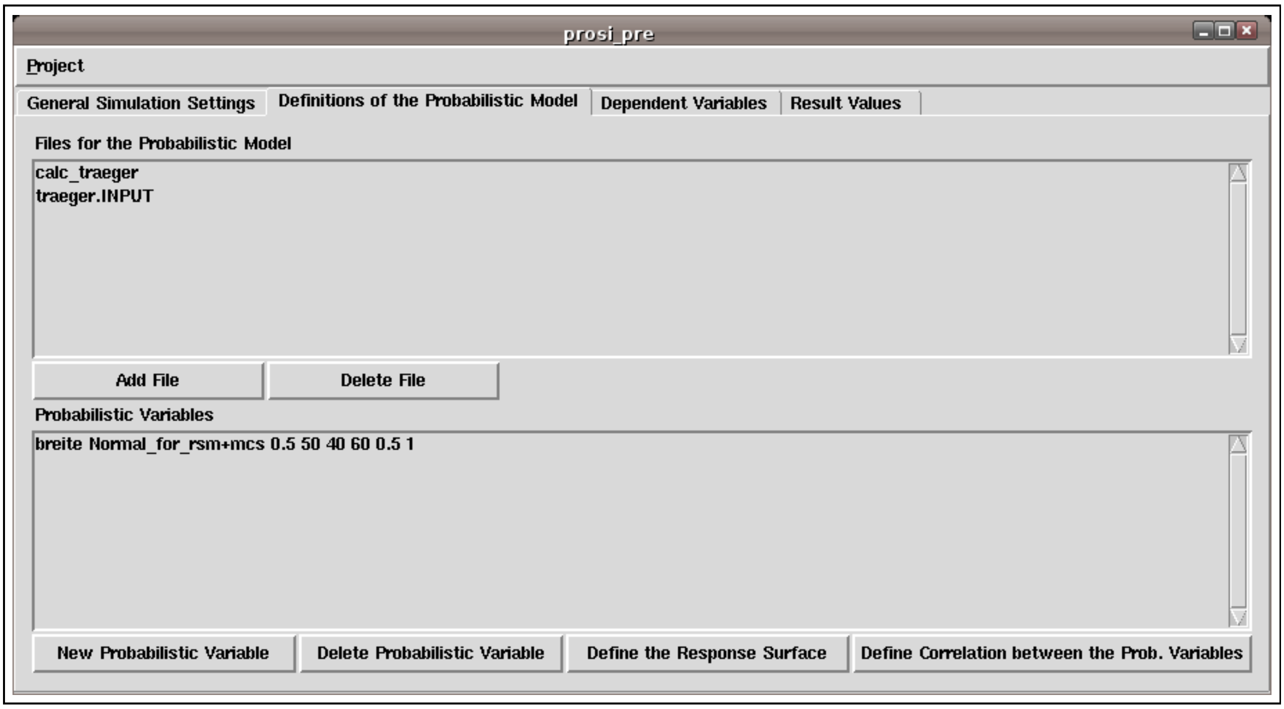
Power 2 Interval: 1 - 3

Improvement No

Figure 3.3. "General Simulation Settings"-Tab completed

3.1.5. Definitions of the Probabilistic Model

The definition of the probabilistic model is very important for ProSi to work right. This tab (see Figure 3.4) is divided into two screens. The upper one defines the actual files and the lower one defines new probabilistic variables. First use the "Add File"-Button to add all the files needed for the model. In the example case this is merely the .INPUT-file which contains all the input-data for the beam that is to be put under load and the small program that will calculate the actual deflection with the data included in the .INPUT file. To delete a file from the list, simply highlight it and press the "Delete File"-Button. The file will be removed from the project, but will remain on the harddrive for later use. Later on ProSi will use the defined simulation method to alter the data inside the .INPUT-file and will create a new directory for each clone of the deterministic model. Inside these directories, which will all be placed inside the "data"-directory as defined on the "General Simulation Settings"-Tab, ProSi will copy all the files included in this list, so that each and every shot can easily be analysed by itself. This feature is also used to distribute the entire workload onto several computers if large amounts of data are to be processed. A more detailed instruction on how to do this will be given in the section 3.2.

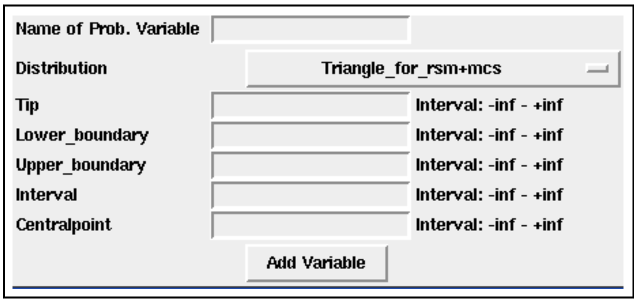

prosi_pre
Project
General Simulation Settings
Definitions of the Probabilistic Model
Dependent Variables
Result Values
Files for the Probabilistic Model
Add File
Delete File
Probabilistic Variables
New Probabilistic Variable
Delete Probabilistic Variable
Define the Response Surface
Define Correlation between the Prob. Variables

calc_traeger
traeger.INPUT

breite Normal_for_rsm+mcs 0.5 50 40 60 0.5 1

Figure 3.4. "Definitions of the Probabilistic Model"-Tab

Now that the list of needed files is defined, move on to the definition of the actual probabilistic variable, which ProSi will generate. Click on "New Probabilistic Variable". An input screen will open which will look like this: First give the variable a unique, recognizable name. The name may not contain spaces or arithmetic operators. Then choose the type of random distribution that is to be used. The given options depend on which Simulation-Method has been chosen. In case of the example, where both were chosen, the options are as follows:



Distribution Triangle_for_rsm+mcs

Figure 3.5. Definition of a Probabilistic Variable

uniform for rsm+mcs a uniform distribution will be created within the defined lower and upper boundary

normal for rsm+mcs a normal distribution will be created within the defined lower and upper boundary, using the defined mean-value and the scatter σ

lognormal for rsm+mcs a lognormal distribution will be created using the same type of input as a normal distribution

triangle for rsm+mcs a triangle distribution will be created within the lower and upper boundary, using a tip-value inside these boundaries

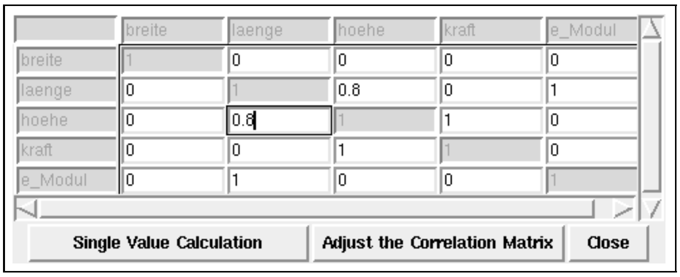
weibull for rsm+mcs a weibull distribution will be created within the lower and upper boundary, using alpha as the scale- and beta as the shape-parameter.

exponential for rsm+mcs an exponential distribution will be created within the lower and upper boundary, using lamba as the parameter.

In each case, there are two more input-fields on the bottom called “Interval” and “Centralpoint”. The interval defines the size of the test plan and the centralpoint obviously defines the center of the test plan. These are the definitions needed for the RSM-calculations. Simply enter the desired values into the appropriate fields.

If using just the RSM then none of the 4 above distributions will be available, but just a definition for the RSM-Variable(s) will be requested asking for an interval and a centralpoint.
If using only MCS, the 4 distributions will remain the same, except that there is no possibility to enter values for interval and centralpoint.
If the probabilistic variables have been set and one later on changes the Method of Simulation on the “General Simulation Settings”-Tab, then the probabilistic variables have to be deleted and newly defined.

To delete a certain probabilistic variable, simply highlight it and click “Delete Probabilistic Variable”. It is also possible to alter a previously defined probabilistic variable by highlighting it and then clicking on it in the list with the Right Mouse-Button.
If desired, a certain wanted correlation between any of the defined probabilistic variables may be set using the “Define the Correlation between the Prob. Variables”-Button on the lower right. A screen will open with two lists. On the left all of the defined variables are listed and may now be selected to be included in the correlation matrix by simply clicking on them. Alternatively “Select All Variables” will obviously select all variables into the matrix. Now click on “View the Correlation Matrix” to open the following screen: Now the desired correlation between any of the probabilistic variables can be entered. There are two ways of entering correlations into the matrix. By default the full correlation matrix calculation will pop up. In this screen one may enter desired correlations between the previously selected variables. When done, click on “Adjust the Correlation Matrix” and ProSi Pre will now check whether the entered correlation coefficients are valid and if necessary will adjust the matrix to the nearest possible values. By doing this, one accepts that every correlation coefficient is given the same importance within the entire matrix.



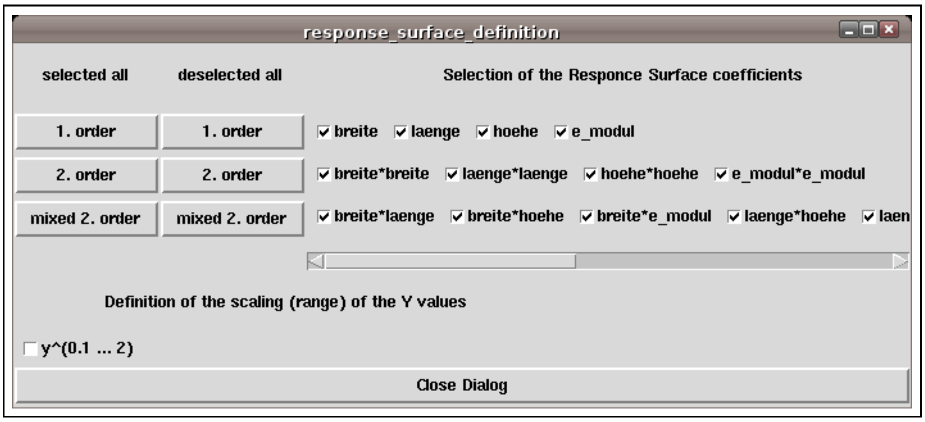
	breite	laenge	hoehe	kraft	e_Modul
breite	1	0	0	0	0
laenge	0	1	0.8	0	1
hoehe	0	0.8	1	1	0
kraft	0	0	1	1	0
e_Modul	0	1	0	0	1

Figure 3.6. Definition of a Correlation Matrix

If there are pairs of variables that need more specific correlations and the “Adjust the Correlation Matrix”-Button alters the correlation in an unacceptable way, then use the “Single Value Calculations”-Method by clicking on the respective button. Now start with the most important pair of variables and enter their correlation. Move on to the next most important pair and use the right mouse-button

to click into the specific field. A popup-window will appear with a range of by ProSi Pre calculated possible correlation coefficients. Choose a correlation coefficient within these boundaries and enter it into the input-field. Move down the list of pairings by their importance. One should notice that the possible range of correlation becomes smaller with increasing number of defined correlations.
When done, simply click close to return to the previous screen.

Back in the main window of the definition of the probabilistic model a forth button is available if the project being set up contains a rsm-calculation - the “Define the Response Surface”-Button. Upon clicking it the screen as shown in figure 3.7 is seen. It is now possible to determine which probabilistic


response_surface_definition
selected all
deselected all
Selection of the Responce Surface coefficients
1. order
1. order
2. order
2. order
mixed 2. order
mixed 2. order
Definition of the scaling (range) of the Y values
Close Dialog

☑ breite ☑ laenge ☑ hoehe ☑ e_modul
☑ breite*breite ☑ laenge*laenge ☑ hoehe*hoehe ☑ e_modul*e_modul
☑ breite*laenge ☑ breite*hoehe ☑ breite*e_modul ☑ laenge*hoehe ☑ laen
□ y^(0.1 ... 2)

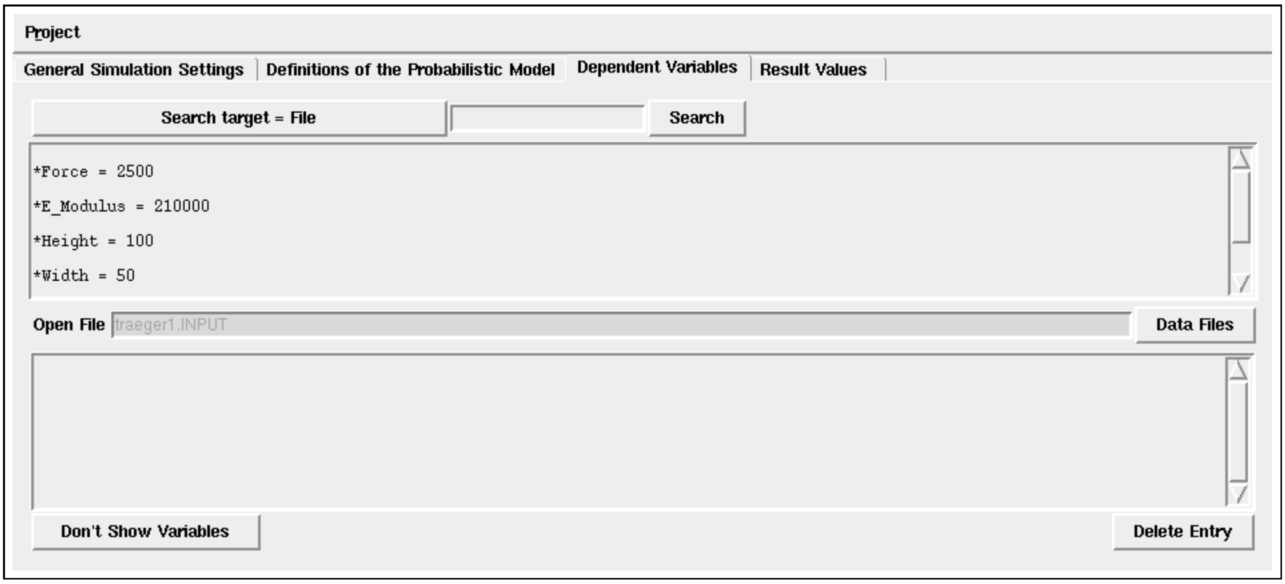
Figure 3.7. “Definition of the Response Surface”-Tab

variables are to be included into the calculation of the response surface. It is possible to select or deselect any combination of variables that would be included into a full response surface. This feature is especially useful for users that know which variables in which combination will have no or only very little effect on the response surface. By deselecting those combinations the coefficient for this combination of variables within the response surface calculation will be set to zero. Please note that this can cause serious errors since those combinations of variables will not be put into any calculation of the response surface. Only deselect them when absolutely sure. Otherwise use the “Automatic Reduction”-feature that automatically determines which combination has no or very little influence on the system.
Additionally it is possible to activate an “Automatic Improvement” of the response surface using exponentiation of the y-values. To do this, check the box at the bottom of the window. This will cause ProSi to exponentiate the values of y using the given range of exponents.
When done, simply click close to return to the previous screen.

Now ProSi Pre for the first time has enough information to save a .inp-file that ProSi can work with. Do this by selecting “Save ProSi project file” from the “Project”-Menu on the top left.

3.1.6. Dependent Variables

This section deals with the definition of the dependent variables ProSi will calculate from the previously defined probabilistic variables and the deterministic model (Figure 3.8).
So first select the data file by clicking on “Data Files”. A screen will pop up that will prompt the user to select one of the files of the deterministic model selected in the previous tab. In the example case this merely prompts the .INPUT-file and the actual program to calculate the deflection. The .INPUT file has been chosen as choosing the program-file would not make any sense, since a program should not be changed using probabilistic methods. The upper part of the window will now simply display

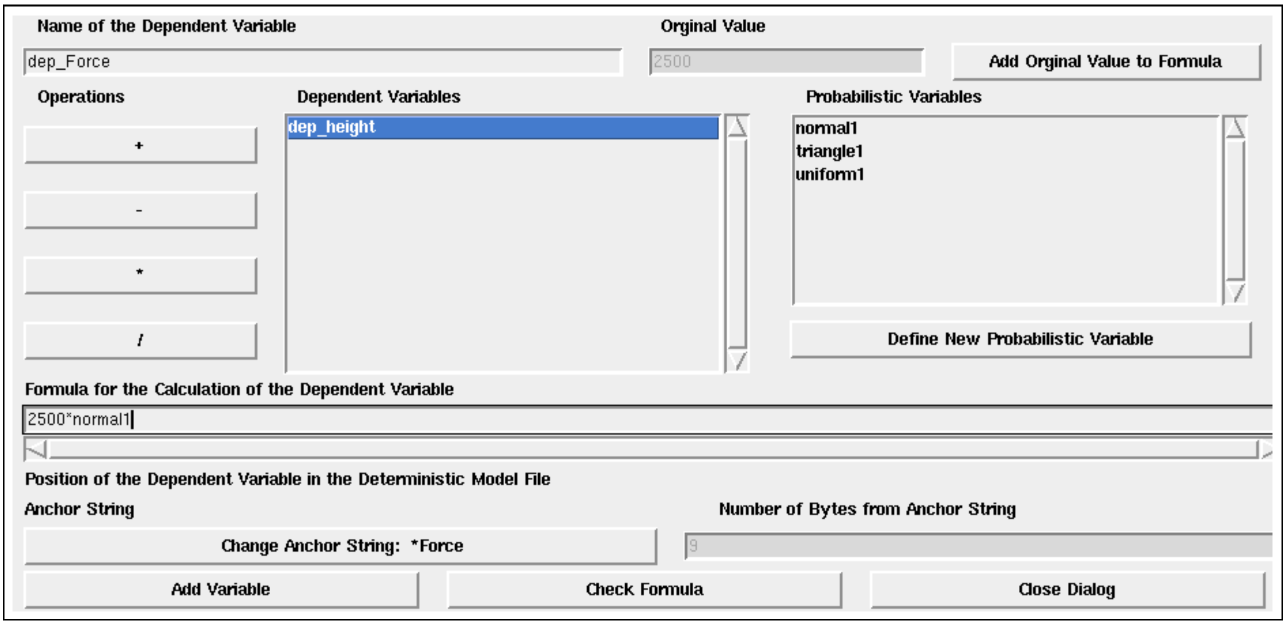


```
*Force = 2500
*E_Modulus = 210000
*Height = 100
*Width = 50
```

Open File: traeger1.INPUT

Figure 3.8. "Definitions of the Dependent Variable Positions"-Tab

the given file. To define a dependent variable using any of the values included in the selected file, simply highlighting it in the upper list and then right-click on that highlighted value. To highlight a certain value either click and drag the mouse-pointer over the value, or simply double-click the value of interest. It is also possible to select more than one value within the same column. How to do this, is discussed later within this chapter. If the data file is very long, one may need to search for a specific value by entering its name into the search field and click search. After right-clicking the appropriate value, ProSi Pre will prompt a new window in which the dependent variable can be defined (Figure 3.9). First of all a summary over all the information contained in this window. On the top left is the


Add Orginal Value to Formula
Operations
+
-
*
/
Define New Probabilistic Variable
Add Variable
Check Formula
Close Dialog

Name of the Dependent Variable: dep_Force

Orginal Value: 2500

Dependent Variables: dep_height

Probabilistic Variables: normal1, triangle1, uniform1

Formula for the Calculation of the Dependent Variable: 2500*normal1

Position of the Dependent Variable in the Deterministic Model File

Anchor String: Change Anchor String: *Force

Number of Bytes from Anchor String: 9

Figure 3.9. "Definitions of a Dependent Variable"-Window

field for defining the name for the dependent variable. Do this at first and use a unique, recognizable name. The name may contain up to 15 characters, but again no spaces or arithmetic operators. Also note, that a dependent variable must not be named like any of the previously defined probabilistic variables. Moving down on the left, there is a list of all the available mathematical operations ProSi can do when given this input-file. To the right of this list is a field that contains any previously defined dependent variable that may be used again within the current dependent variable. On the very right is a field with all of the previously defined probabilistic variables.

In case a wrong value had been used to create this dependent variable, it is possible to change it. To do this one needs to go back to the list where the wrong value originally had been highlighted. Be aware that the definition screen needs to stay opened. Do not close that dialog yet. In the list, highlight the right value. Now hold down the Ctrl-Key and click on it. Now check the definition dialog and the new value will be inserted there.

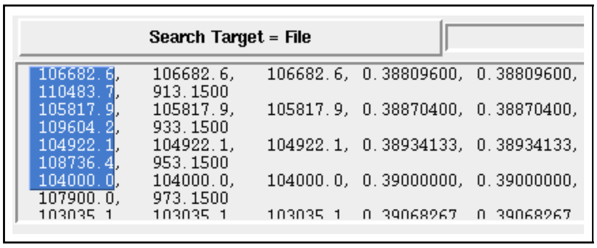
So after a fitting name has been given, put the cursor into the field that is labeled "Formula for the Calculation of the Dependent Variable". One may now proceed and actually define this variable. A useful feature is to click on "Add Original Value to Formula" as this will obviously copy the value which had been highlighted to create this dependent variable into the calculation-field. It is now processed as a part within the calculation of the dependent variable. Also any of the probabilistic variables or dependent variables shown in the above lists may be used within the formula of the new dependent variable. Simply add the full name of each variable and a fitting operator to the calculation-line by either typing the name or by highlighting the desired variable with the left mouse-button and then clicking the middle mouse-button. Please note that ProSi can only process the operators given in the list. x^2 for example cannot be entered as xˆ2, but needs to be x*x. Once the entire formula is entered, click the "Check Formula"-Button and ProSi Pre will check whether the given formula-string can be processed by ProSi. Edit the formula if necessary.

A very useful feature is the "Change Anchor String"-Button. To understand this feature, a little background may be required. By default this button contains the information "_Beginning_of_File_" and a number in the grey field behind the button, for example 26. This simply means that ProSi will look for the value the user had previously highlighted to create this dependent variable after the 26th character of the input-file. As long as the input-file is fairly short and unlikely to be changed in any way this may appear sufficient. But if the input file is changed in any way, even if it is just an added space in the very first line, the entire process of reading values will be corrupted, since the expected value will now be after character-position 27. To avoid this, one may change the anchor string to something closer to the value, for example the value-name. First highlight the value-name within the "Dependent Variables"-Tab within the ProSi Pre main-program without closing the window for the newly created dependent variable. Drag the window aside, if needed. Now switch back into the window for the dependent variable definition and click "Change Anchor String". The button should now contain the previously highlighted string of characters and a number behind it. Now ProSi will not count the characters from the beginning of the file to pick the value, but it will first search for that new anchor-string and then count from the first character of the anchor string.

Now click on "Add Variable" to add the newly defined variable into the list of dependent variables. If need be, one may change any variable by simply right-clicking on it in the list, which will again prompt the definition screen for that specific variable. To totally delete a variable, highlight it and click the "Delete Entry"-Button.

As mentioned earlier, several values of one column may be selected at once to be changed within one single step. To visualize this approach, note that a different example-file has been used for the next few steps. Such a selection should look like shown in Figure 3.10. ProSi Pre will now prompt a slightly different definition screen for the dependent variable.

But first, how is such a block-selection done? Press and hold down the Ctrl-Key of the keyboard and click in front of the first value of the desired block. Now release Ctrl. Nothing visible happens yet. Now just click behind the last digit of the block that is to be selected. Now the blue selection should appear as seen above. Note that the Ctrl-Key defines the beginning of a block-selection. The second



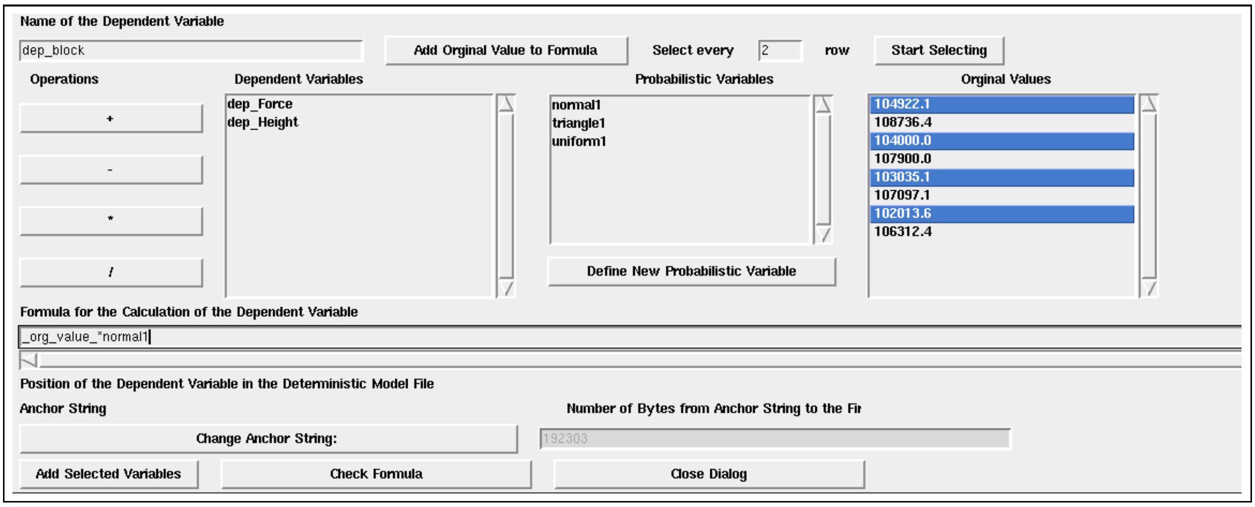
```
106682.6,    106682.6,    106682.6,  0.38809600,  0.38809600,
110483.7,    913.1500
105817.9,    105817.9,    105817.9,  0.38870400,  0.38870400,
109604.2,    933.1500
104922.1,    104922.1,    104922.1,  0.38934133,  0.38934133,
108736.4,    953.1500
104000.0,    104000.0,    104000.0,  0.39000000,  0.39000000,
107900.0,    973.1500
103035.1     103035.1     103035.1   0.39068267   0.39068267
```

Figure 3.10. Block-Selection

click is therefor done without Ctrl.

Important Note: Be sure to check whether the entire value has been selected! ProSi Pre cannot recognize whether the highlighted digits are what is actually needed! So it might happen, that one makes the first click a little too far right and thus in the example the selection would not start with 106682.6, but with 06682.6 or even 6682.6. The same applies to the second click that defines the right end of the block. Make sure all of the wanted digits are included or the number of the first line might end in 106682. or even less. In any case, ProSi Pre will continue to use these wrong values as it is not possible to check this within the program. In case either of this happened, discard the newly popped up definition window by clicking “Close Dialog” and redo the selection.

Back to the definition screen:



Name of the Dependent Variable: dep_block

Select every 2 row

Dependent Variables: dep_Force, dep_Height

Probabilistic Variables: normal1, triangle1, uniform1

Orginal Values:
104922.1
108736.4
104000.0
107900.0
103035.1
107097.1
102013.6
106312.4

Formula for the Calculation of the Dependent Variable: _org_value_*normal1

Position of the Dependent Variable in the Deterministic Model File

Anchor String

Number of Bytes from Anchor String to the Fir: 192303

Figure 3.11. Block-Selection Definition Screen

The basic layout remains similar to the one described earlier, with the difference of the new list of selected variables on the right side. The selected block of values is now visible in that list. As with the single-value definition, enter the formula the dependent variable is supposed to use in the respective line. When done, check the formula by clicking on “Check Formula”. Now define the original values to be used. As ProSi Pre selects every line within the “drawn” box, it might become necessary to actually redefine which variables of that box are to be used. Enter a “2” into the “Select every”-Field and only every second line will become highlighted. If necessary to just read every third or forth line, simply enter a “3” or “4” or any other desired value. Now click “Start Selecting” and the values to be processed become highlighted. It is also possible to completely manually choose which variables are to be used, by right-clicking all the wanted variables. Finally click “Add Selected Variables” to return to the previous screen.

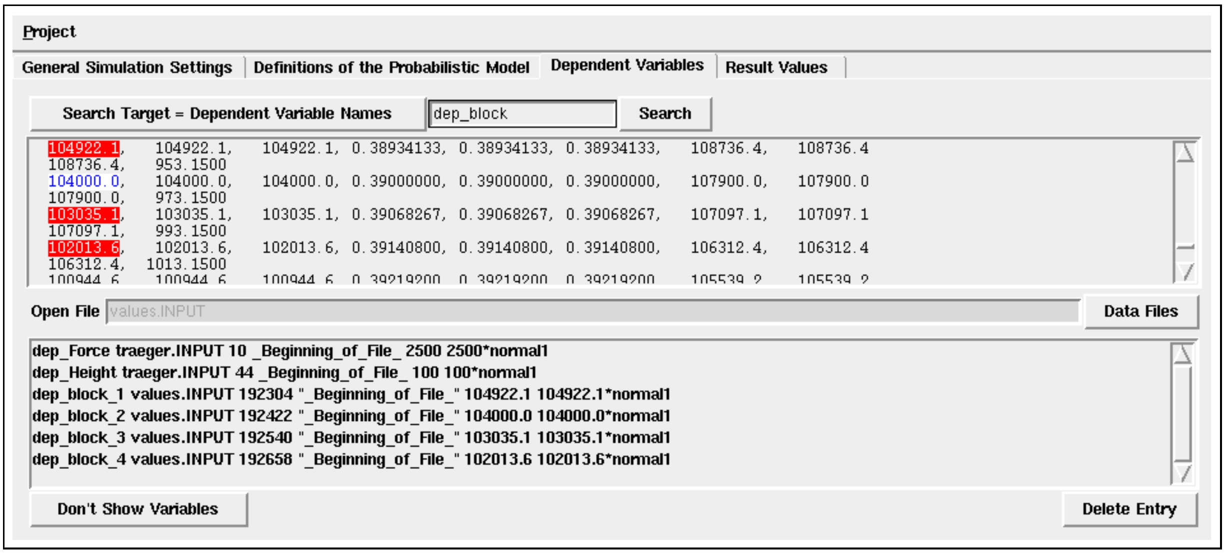
ProSi Pre is now ready to give ProSi a fully defined input-file. Save the input-file and continue with ProSi as described in the ProSi documentation.

Before moving on to the last tab, there are some more features to be described within this window.

As stated earlier, this window contains a search feature to search for specific information within the lists. The default search is to look for specific strings within the source-file as it is shown in the upper part of the window. So if looking for Young’s Modulus within a large input-file, simply type an appropriate searchtext into the field. The search-function may as well be used to search for specific dependent variables. This can be rather useful if a large amount of them has been created. To do this, press the “Search Target = File”-Button and it changes to “Search Target = Dependent Variables Names”. Enter the name of the variable to be searched for and click “Search”.

Now the bottom-list may be used for some more interesting features. By right-clicking on any of the variables, the respective variable can be altered through the definition screen.

By holding down the Ctrl-Key and left-clicking on any of the dependent variables the upper list will display the actual value highlighted blue within the source-file (see Figure 3.12). All other values that are bound to any defined variable are highlighted red within the original file. If different defined variables were created using different files, ProSi Pre will automatically open the needed file. On the bottom left is a button labeled “Don’t Show Variables”. A click on it will cause the highlighting of the variables in the upper list to be toggled off. This is useful if large blocks had been defined. Highlighting these large amounts of data could severely slow down the computer the program is being run on. The last option is the “Delete Entry”-Button which will cause the selected dependent variable to be removed from the list.


Project
General Simulation Settings
Definitions of the Probabilistic Model
Dependent Variables
Result Values
Search Target = Dependent Variable Names
Search
Data Files
Don't Show Variables
Delete Entry

dep_block

```
104922.1,    104922.1,    104922.1, 0.38934133, 0.38934133, 0.38934133,    108736.4,    108736.4
108736.4,    953.1500
104000.0,    104000.0,    104000.0, 0.39000000, 0.39000000, 0.39000000,    107900.0,    107900.0
107900.0,    973.1500
103035.1,    103035.1,    103035.1, 0.39068267, 0.39068267, 0.39068267,    107097.1,    107097.1
107097.1,    993.1500
102013.6,    102013.6,    102013.6, 0.39140800, 0.39140800, 0.39140800,    106312.4,    106312.4
106312.4,   1013.1500
100944.6,    100944.6,    100944.6, 0.39219200, 0.39219200, 0.39219200,    105539.2,    105539.2
```

Open File values.INPUT

```
dep_Force traeger.INPUT 10 _Beginning_of_File_ 2500 2500*normal1
dep_Height traeger.INPUT 44 _Beginning_of_File_ 100 100*normal1
dep_block_1 values.INPUT 192304 "_Beginning_of_File_" 104922.1 104922.1*normal1
dep_block_2 values.INPUT 192422 "_Beginning_of_File_" 104000.0 104000.0*normal1
dep_block_3 values.INPUT 192540 "_Beginning_of_File_" 103035.1 103035.1*normal1
dep_block_4 values.INPUT 192658 "_Beginning_of_File_" 102013.6 102013.6*normal1
```

Figure 3.12. “Definitions of the Dependent Variable Position”-Tab completed

3.1.7. Result Values

The last tab named “Result Values” is used for post-processing purposes (see Figure 3.13). It basically works the same as the “Dependent Variables”-Tab. In terms of the beam under load example this means that after ProSi’s first run to generate the changed clone models of the original deterministic model the actual deflection calculation has to take place by starting the calculation-routine within every clone.

The following describes how the .inp-file has to be set up in order for ProSi to be able to do a result-scan.
First of all, one needs a complete result-file of the deterministic model. This file needs to be part of the original deterministic model and should therefor be located in the directory for the deterministic model. Remember, all of this is done before ProSi is started the first time. That means that for the example, the deflection calculation has to be done using the unaltered, original deterministic model. Using the now acquired .OUTPUT-file, ProSi Pre may now be instructed in how to analyse the .OUPUT-files that will be gained from the generated clones.
Click on “Add Result Data File”. A selection screen will pop up and one may choose the .OUPUT-file just created from the deterministic model. The file will be opened in the upper part.

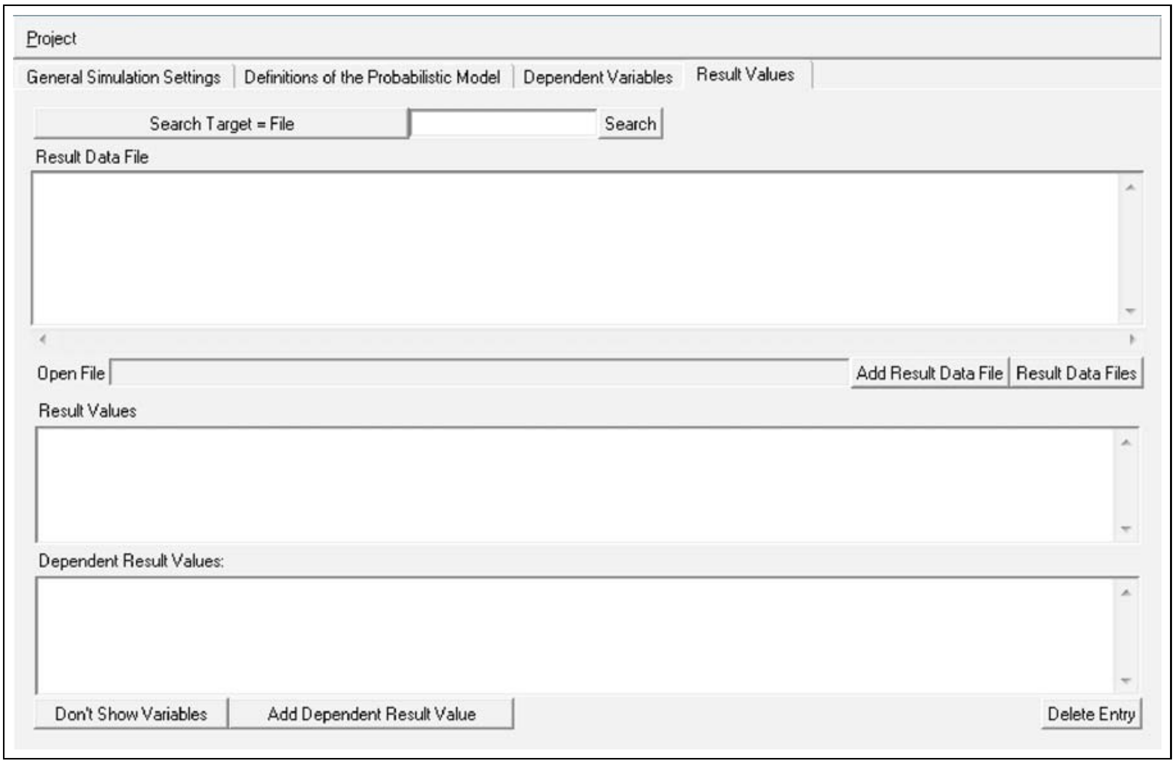
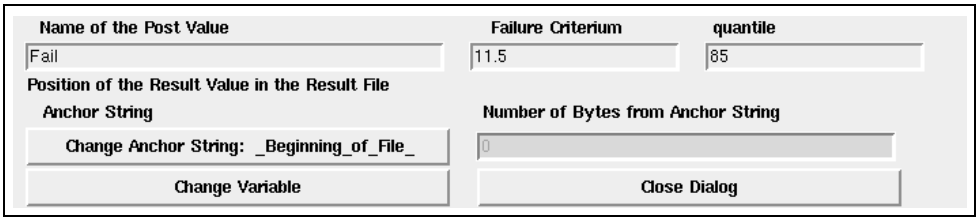


Figure 3.13. “Definitions of the Result Values”-Tab

It is seen that the .OUTPUT-file merely consists of the deflection-value. Other more sophisticated output-files may include a lot more information. Again, there is a search-function that will work as described within 3.1.6. To create the evaluation-variable highlight the value that is to be analysed and right-click it. Note, that as the length of the string of the result value may vary, ProSi will automatically use all digits following the selection until the next non-digit character. In addition ProSi includes a mechanism to prevent reading incomplete values or missing a leading minus sign. This works in the way that if ProSi finds a number or a minus directly left of the selected position, the reading position will be adjusted automatically to the left until a non number sign is reached. If this automatic engages a warning is written to the log file. The read in values should be checked manually in this case. A screen like Figure 3.15 will pop up. ProSi will use the now defined variable(s) within its postprocessing routines to write an *.out file that may be analysed using appropriate programs like ProSi Post or other spreadsheet analysis programs.
As usual, define a recognizable name. There again is a function to change the Anchor String. This will work as described within 3.1.6. Reasons for doing this are also given in that subsection. In case a wrong value had been used to create this dependent variable, it is possible to change it. To do this



Name of the Post Value: Fail

Failure Criterium: 11.5

quantile: 85

Position of the Result Value in the Result File

Anchor String: Change Anchor String: _Beginning_of_File_

Number of Bytes from Anchor String: 0

Figure 3.14. "Definitions of a Result Value"-Window

one needs to go back to the list where the wrong value originally had been highlighted. Be aware that the definition screen needs to stay opened. Do not close that dialog yet. In the list, highlight the right value. Now hold down the Ctrl-Key and click on it. Now check the definition dialog and the new value will be inserted there.

To get a first overview over the resulting data, two options are given. Either search for a "Failure Criterion" or analyse a specific Quantile of shots.

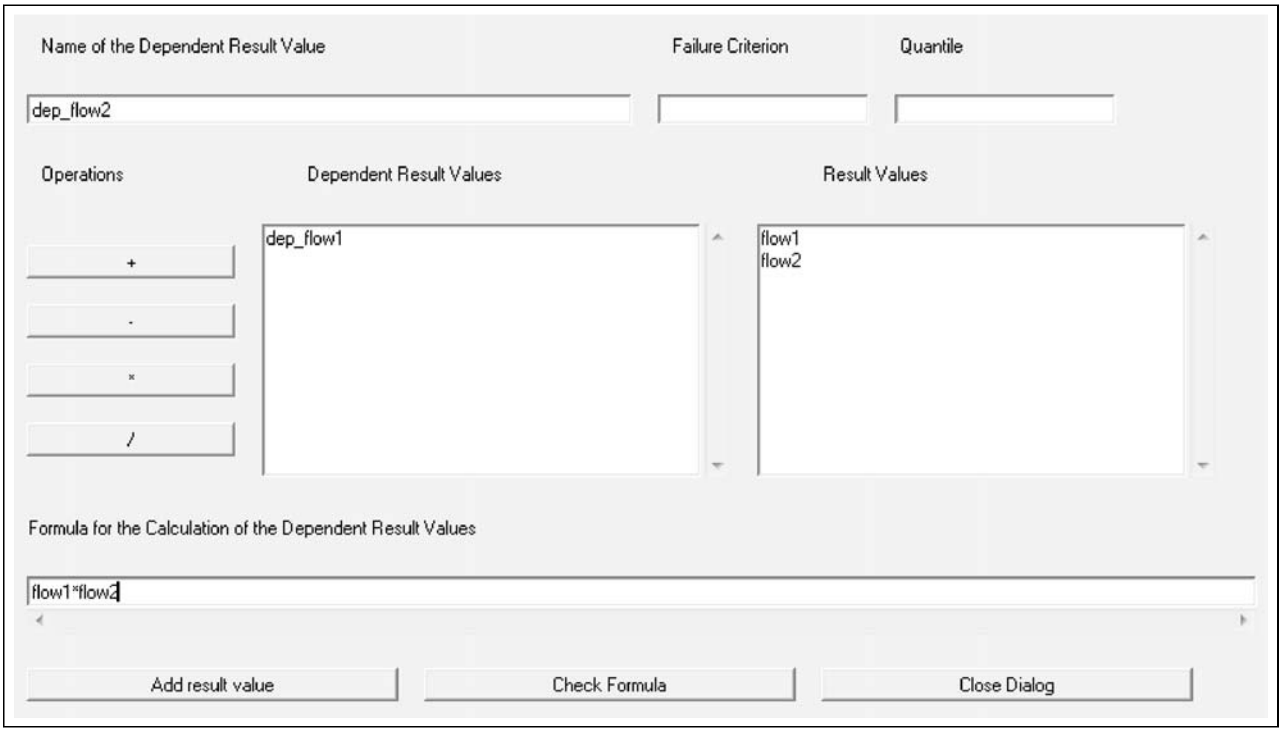
The first is rather simple. ProSi will search for any result-value within the clones that exceed this "Failure Criterion". It will return the percentage of all shots that exceed this value.

The option to search by Quantile simply means that one specifies a certain Quantile, 80% for example, and ProSi will return the value that Quantile represents, so the user now knows, below which value 80% of all shots are located.

Please note that ProSi will always integrate from the defined boundary to positive infinity.

After the variable has been fully defined, simply click "Add Variable". Again, a previously defined variable may be altered by right-clicking it within the list of variables.

The button "Don't Show Variables" has been described in 3.1.6. Reasons for this button are also given within that subsection.



Name of the Dependent Result Value: dep_flow2

Failure Criterion:

Quantile:

Dependent Result Values: dep_flow1

Result Values: flow1, flow2

Formula for the Calculation of the Dependent Result Values: flow1*flow2

Figure 3.15. "Definitions of a Dependent Result Value"-Window

In the lower section, depended result values can be created and listed. Like the dependent variables, ProSi will calculate the dependent result values with the previous defined result values. Press ”Add Dependent Result Value” to open the dialogue for new dependent result values (see Figure 3.15). Enter a name for the new result value, in the appropriate input box. In the lower input box, which is named ”Formula for the Calculation of the Dependent Result Values”, one can enter the formula. Use the names of the existing result value (which can be seen in the box ”Result Values”) and the operator +, -, *, and / as listed as the buttons on the left side of the dialogue. Please be aware, that ProSi can only process the operators given in the list. x^2 for example cannot be entered as xˆ2, but needs to be x*x. Once the entire formula is entered, click the “Check Formula”-Button and ProSi Pre will check whether the given formula-string can be processed by ProSi. Edit the formula if necessary. One can also define failure criterion and quantile as described before for result values.

3.1.8. Saving the Project and further steps

After setting up the project with ProSi Pre there are two possibilities to go on. The first is to simply save the project by clicking the respective button entry in the “Project”-menu. ProSi Pre will now prompt an appropriate filename using the name of the simulation defined within the “General Simulation Settings”-Tab. This given name may be altered as seen fit. After saving the project ProSi Pre may be closed and all further steps have to be taken manually.

The second possibility to go on is to directly send the project to the module ProSi Exe which manages further work with ProSi. This may be done by clicking on the “Start ProSi execution program” entry within the “Project”-menu. ProSi Pre will again prompt the “Save” screen and after the project has been saved ProSi Pre will exit and instantly start the ProSi Exe program. For further documentation on ProSi Exe see the next chapter.

3.2. ProSi Execute

3.2.1. Introduction

ProSi Exe is the central module of the ProSi product family. It is the graphic user interface (see Figure 3.16) from which all features can be accessed and managed.

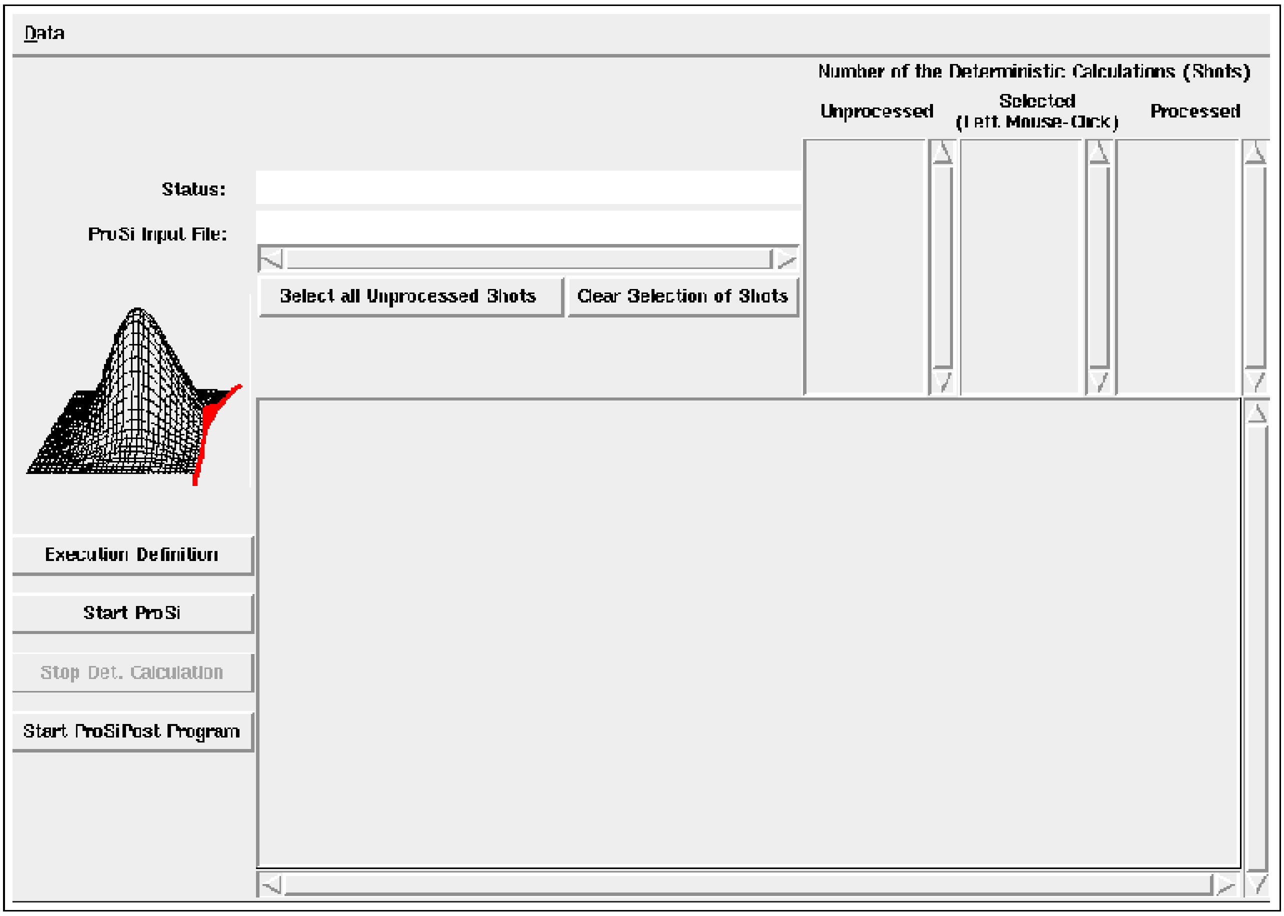


Figure 3.16. "ProSi Execute"-Window

on Linux

To start ProSi Exe change into the directory where it is located. Now type *./tclkit prosi_exe.kit* and the above main window will be displayed. It may also be started out of ProSi Pre after creation of a project file. For further documentation about ProSi Pre and its functions, please refer to section 3.1.

on Microsoft Windows

To start ProSi Exe click on START - PROGRAMS - ProSi_2.3 - ProSi_exe.

3.2.2. Getting Started

To begin working with ProSi Exe a project file is needed. This project file is created using ProSi Pre. There are different ways to acquire the project file from ProSi Pre.

- ProSi Pre has been used separately and a project file has been saved

- ProSi Exe has been started out of ProSi Pre after creation of a project

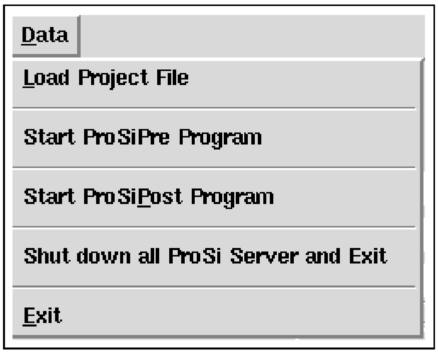


Figure 3.17. “The ProSi Exe Data”-menu

If using a previously created project simply select “Load Project File” from the “Data”-menu or ProSi Exe (Figure 3.17) may also be directly started with a valid file by appending the file name to the syntax of starting ProSi Exe (Linux only), for example “*./tclkit prosi_exe.kit example.inp*”. If ProSi Exe has been started out of ProSi Pre then there is no need to open a project file, as the project file last used within ProSi Pre is automatically opened. In case the third possibility applies and there is no project ready at all, it has to be created using ProSi Pre. For this either manually start ProSi Pre as described in its manual or use the “Start ProSi Pre program” entry within the “Data”-menu. This will cause ProSi Pre to open and a project may be defined as described in its manual.

Upon loading a project file ProSi Exe will check whether this project has been processed before and in case it has, a window will open where to choose either a previous sub-project for recalculation or to create a new sub-project. This is useful if large amounts of shots are to be processed so for example the first half of all shots is calculated in a first session and later on ProSi Exe can continue calculating the remaining shots by choosing this project from the list.

Another option is given within the “Data”-menu called “Start ProSi Post program” which will start the tool ProSi Post described in section 3.4. This menu-entry has the same function as the button on the main ProSi Exe window.

The last option besides exiting ProSi Exe is to first shut down all open ProSi Servers running on the local machine and on all slaves as previously defined within the “Execution Definition”. After all servers are shut down, ProSi Exe exits. Please note, that this procedure shuts down all servers regardless of possible other running calculations. This may occur when more then one instance of ProSi Exe have been started and are conducting calculations. The reason for this is, that per user only one ProSi Server is started, but several ProSi Exe may run on it. So the use of this feature is only recommended after all ProSi work is done.

Once ProSi Exe has acquired a valid project file move on to the “Execution Definition”.

3.2.3. Execution Definition

The right definition of the execution parameters for the deterministic model is vital to the correct operation of ProSi. Click the button to open the “Execution Definition” window (see Figure 3.18) . To have the calculation be distributed over several computers (Linux only) it is possible to enter slaves within the top part of this window.

To create a new slave computer press the “New Slave”-Button. This will prompt the window in Figure 3.19. First a “Host Name” has to be defined. This name is merely used to help the user identify the slave later on, so a recognizable name is recommended.

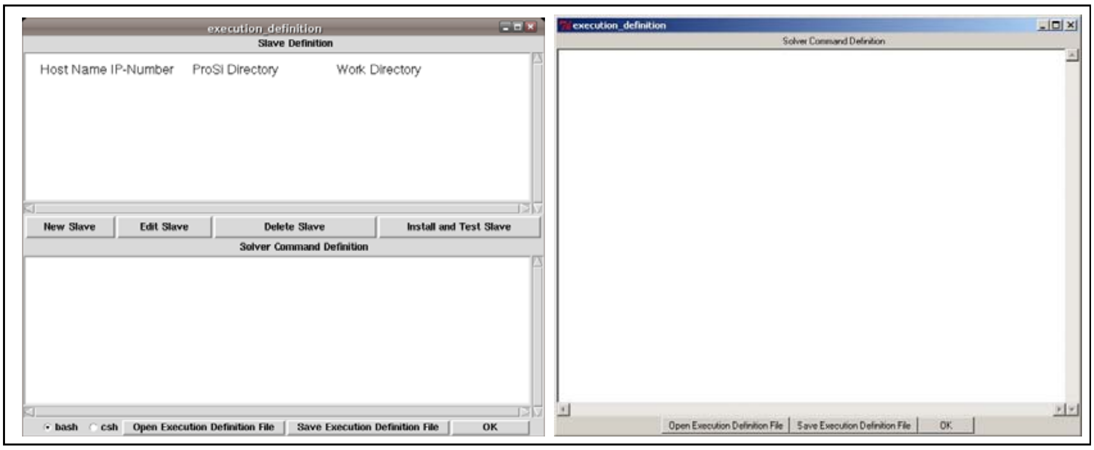

execution_definition
Slave Definition
Host Name IP-Number
ProSi Directory
Work Directory
New Slave
Edit Slave
Delete Slave
Install and Test Slave
Solver Command Definition
bash
csh
Open Execution Definition File
Save Execution Definition File
OK
execution_definition
Solver Command Definition
Open Execution Definition File
Save Execution Definition File
OK

Figure 3.18. "Execution Definition"-Window (left: Linux; right: Microsoft Windows)

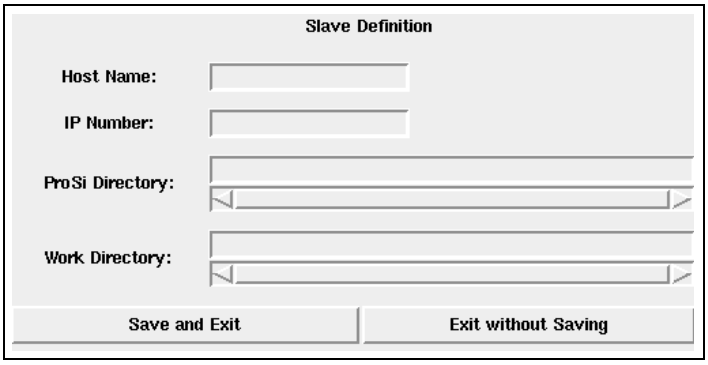


Figure 3.19. "Slave Definition"-Window

More important is the "IP Number". Enter the correct number into the field to ensure that the right host is being used.
The "ProSi Directory" defines where ProSi is completely installed on the slave computer. The path given here needs to be the absolute path on the host.
The same applies to the "Work Directory". It defines where the calculations of ProSi will take place and where the work will be stored. Within this directory ProSi will create the directories that have been defined within the *.inp file by using ProSi Pre.
After defining all slaves ProSi Exe will start the needed components on the slaves and will then test whether all the slaves work properly. If an error occurred during testing, please recheck all the slave settings and also check whether the slaves themselves all work properly.
If no information about slave-computers is given, then ProSi Exe will run entirely on the local machine where it is situated. A message will confirm this upon clicking the "OK"-Button of this window.
In the bottom part of the window the user has to enter the command line for the solver ProSi will start for each deterministic shot. Later ProSi Exe will run this command line for every shot that has been created by ProSi. This command line may be entered using regular Linux/Microsoft Windows expressions.
For example: you have a C-program "calc_traeger" which needs an input file "traeger.INPUT" for calculation. The command inserted in the window must be:

./calc_traeger traeger.INPUT	(in one line) for use on Linux
calc_traeger.exe traeger.INPUT	(in one line) for use on Microsoft Windows

If you have more than one command in a sequence write each command with all arguments in a separate line in order to there execution.

An important feature at this point is to define whether the above defined command line script / batch-file is to be run in either a C-Shell (csh) or a Bourne-Again-Shell (bash) / command line for Microsoft Windows. To do so, simply activate the dot corresponding to the desired shell.

For later use of the entire definition it may be saved to a separate file using the “Save”-Button on the bottom of the screen. Such a saved file may later be activated by using the “Open”-Button.

After entering all the required data, press “OK” to return to the main screen. ProSi Exe is now ready to run ProSi.

3.2.4. Start ProSi

After completing the Execution Definition the program is ready to run ProSi. Press the “Start ProSi”-Button on the left and the center display will show what ProSi is currently doing. In the status line above the output display a message is given what ProSi Exe is currently doing.

In the first run ProSi creates the random distributions needed to build the single deterministic shots. This take usually a few seconds, but according to the used settings for distributions, correlations and number of shots, this maybe take several minutes. In the center display a estimated time for the creation is given. If one is not comfortable with the given time, the creation can always be stopped by using the Button “Stop ProSi”. When the first run is finished, ProSi is ready to create and calculate the single shots. To do this, press the “Select all Unprocessed Shots”-Button. Now all shots will be listed within the middle list. To remove all of them, press the respective “Clear Selection of Shots”-Button. It is also possible to select the shots that are to be calculated manually in the left of the three lists by clicking on them using the left mouse-button. The shots that have been selected are now listed in the middle list. False selections may be unselected by simply clicking on the false shot within the middle list. This is especially useful if not the entire amount of shots is to be calculated at once.

Now press the “Start Det. Calculation”-Button to have ProSi Exe create the needed shots and run the solver as it has been defined within the “Execution Definition” earlier. The right list will now display the shots that have already been calculated and the middle list shows the remaining shots.

In case of one or more faulty shots, ProSi will ignore them and continue with the next shot that is free from error. ProSi Exe will report the faulty shot within the center screen. It will also not remove those shots from the list of selected shots, as they have not been processed.

“Start Det. Calculation” also triggers the start of a sub-program called “prosi_server_exe” which monitors the deterministic calculation. This program requires no user-intervention whatsoever and is being shut down after the deterministic calculation is done.

To interrupt the current processing of the deterministic calculation press the “Stop Det. Calculation”-Button. ProSi will finish it’s current calculation and halt afterwards.

If the current calculation was a Monte-Carlo-Simulation ProSi Exe is now finished and one can move on to evaluating the data as described in 3.2.5.

If the current calculation is an RSM-calculation a new button label appears after the deterministic calculation. The button “Start Det. Calculation” switches to “Start RS Calculation” and upon clicking it, ProSi will calculate the response surface using the definitions defined in the input file. After completing the calculation one may again move on to evaluating the data as described in 3.2.5.

If the current calculation is a combined RSM and MCS-calculation yet another new button will appear after the steps for an RSM-calculation have been completed. “Start RS+MCS Calculation”. Clicking it will cause ProSi to run its MCS based on the the definitions of the input files. After completing the calculation one may again move on to evaluating the data as described in 3.2.5.

3.2.5. Evaluating the work

After pressing the “Start ProSi”-Button the second time, even while the solver is still running, it is now possible to directly access the tool ProSi Post to evaluate the calculations. To access ProSi Post press the “Start ProSi Post”-Button in the main window of ProSi Exe or use the entry within the “Data”-Menu. For further documentation on ProSi Post, please refer to the ProSi Post manual. Please note that in case an RSM+MCS-calculation is running, there are two options to access ProSi Post. One to evaluate the response-surface and one to evaluate the Monte-Carlo-Simulation. Simply click the corresponding button to access either data.

3.2.6. Using ProSi Exe with more than one user

The distribution of work onto several slaves using ProSi Exe starts a subprogram, a server, on every single slave in use. It is needed to perform the communication between the slaves and the master. As soon as ProSi Exe communicates with a slave, this server is started. ProSi Exe now uses a user-specific port to communicate with the server on the slave. If a second user is going to use ProSi on the same slave, he could not use the same server, because of user-rights management. Thus a second server is started using another port. Up to 10 users can use the same slave in this manner and thus up to 10 ports will be assigned. By default the ports used are 9001 through 9010.

3.3. ProSi Server

The ProSi Server is the interface between the graphical tool ProSi Exe and the ProSi core module. Furthermore it is responsible for controlling the single calculations and for all network traffic on distributed simulations. The Server is automatically launched with ProSi Exe. Furthermore it is launched on all slave computers during the slave definition in ProSi Exe. The Server does NOT shut down automatically! It stays active after finishing ProSi Exe. If ProSi Exe detects an active Server on startup no additional Server will be started.

The ProSi Server can be started manually using the command *./tclkit prosi_server.kit “port” “name of the user”* (on Linux) or *tclkit.exe prosi_server.kit “port” “name of the user”* (on WINDOWS in the command line).

To shut down all open ProSi Servers running on the local machine and on all slaves as previously defined within the “Execution Definition”, use the “Shut down all ProSi Servers and Exit”-entry in the “Data”-menu of ProSi Exe. Please note, that this procedure shuts down all servers regardless of possible other running calculations. This may occur when more then one instance of ProSi Exe have been started and are conducting calculations. The reason for this is, that per user only one ProSi Server is started, but several ProSi Exe may run on it. So the use of this feature is only recommended after all ProSi work is done.

3.4. ProSi Post

3.4.1. Introduction

ProSi Post (Figure 3.20) has been developed as part of the ProSi product family and can be used for statistical and graphical evaluation of output files created by ProSi. ProSi Post is not restricted to usage with ProSi, as it can also evaluate other databases. The Features included are:

- Values - display entire dataset
- Basic Statistics - simple statistics, minimum, maximum, mean value, standard deviation, ...
- Cumulative Density - graphical display of the cumulative density of a certain variable
- Histogram - graphical display of the distribution of a certain variable
- Ant-Hill-Plot - plot any variable against another
- Pie-Chart - correlation between one variable and a set of others
- Correlation Matrix - correlation between several specified variables
- CoI Matrix - coefficient of importance of the input variables
- Response Surface - display and evaluate the response surface
- Refresh Data - refresh the current data-file

The function of each will be discussed in the respective chapters.

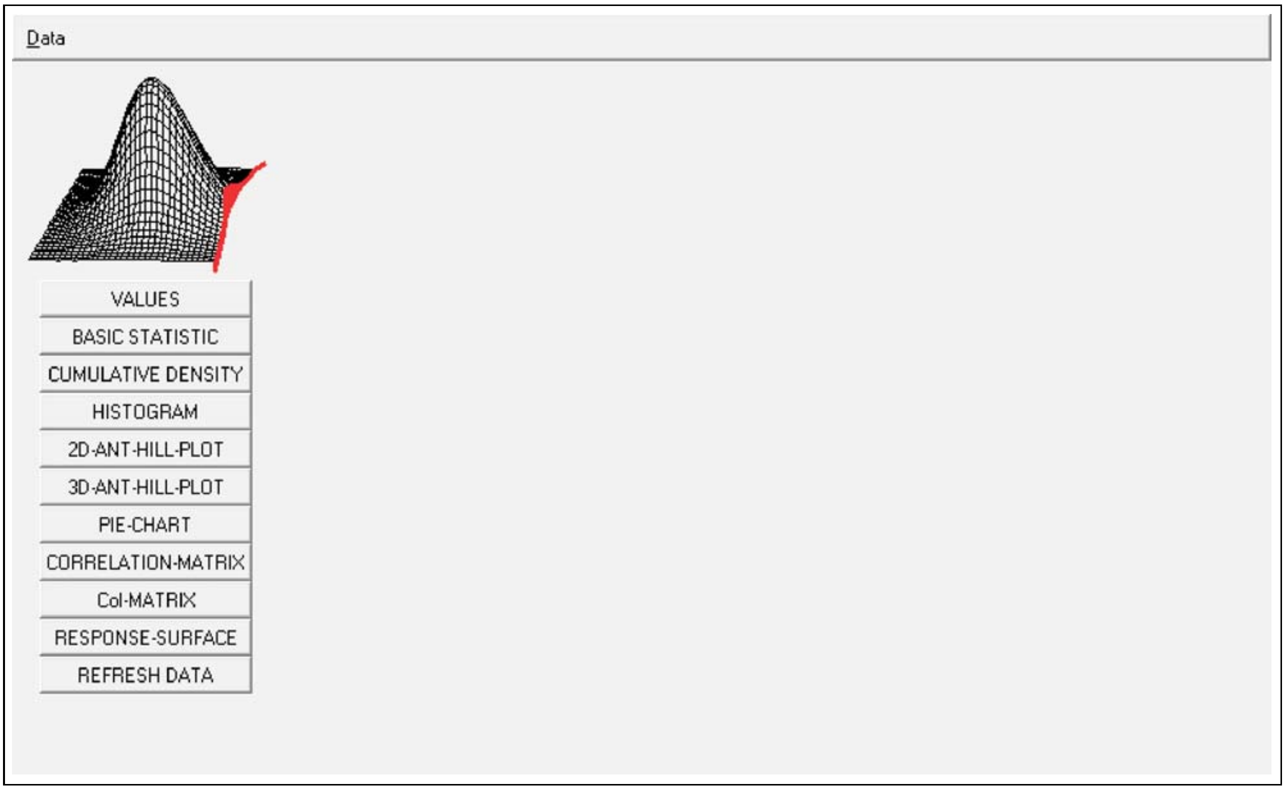


Figure 3.20. ProSi Post Main Window

3.4.2. Starting ProSi Post

on Linux

To open ProSi Post either click on ProSi Post within ProSi Exe or change into the directory where it is located and type *./tclkit prosi_post.kit* and ProSi Post will open.

on Microsoft Windows

To open ProSi Post either click on ProSi Post within ProSi Exe or click on START - PROGRAMS - ProSi_2.3 - ProSi_post and ProSi Post will open.

Now there are three possibilities of getting data into ProSi Post (see Figure 3.21). The first is a fully automated reading of any *.out file created by ProSi, the second is for importing *.INPUT file created by ProSi, the third one is used for reading other formats of databases.

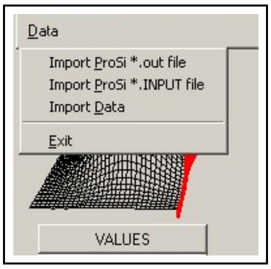


Figure 3.21. ProSi Post Data Menu

3.4.3. Automated Data Import

ProSi Post is able to automatically read a specified *.out file during the process of starting ProSi Post. To do this simply append the name of the file to the command line to open the program (Linux only), for example "./tclkit prosi_post.kit example.out". In case ProSi Post is already running, use the either one of the following methods to open a data file.
After clicking on "Import ProSi *.out file" from the Data menu, simply choose the file to be analysed from the correct directory in the following dialog. Click "OK" and ProSi Post will read all the necessary data. Depending on the amount of data and the speed of the computer ProSi Post is being used on, this process may take from a few seconds to several minutes.
Following this procedure, ProSi Post will read all data included in the .out-file. If desired, for example in cases with large amounts of data, ProSi Post can also read only a certain, specified amount of data. To do this, please proceed using the method of "Manual Data Import" as described in 3.4.4.

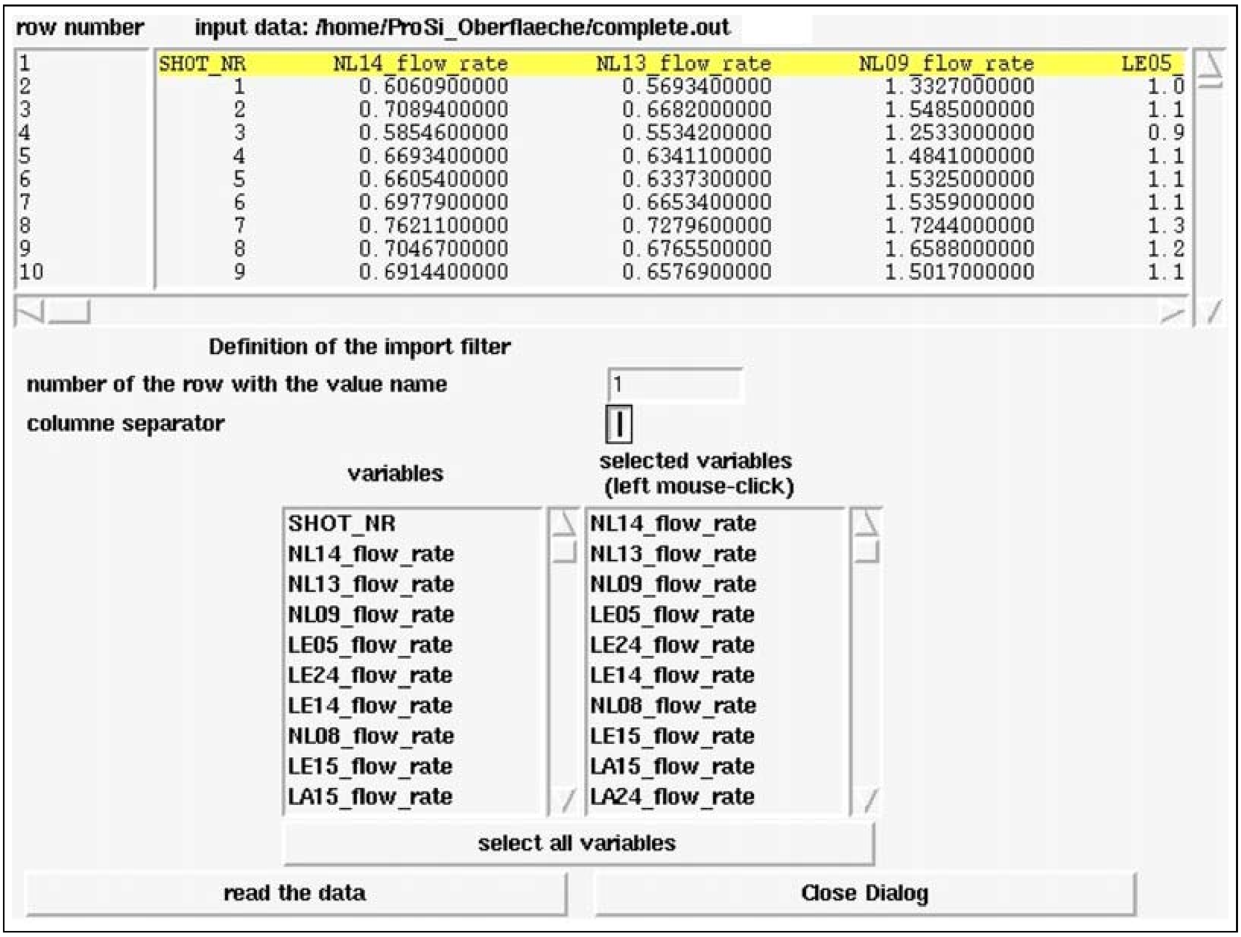
3.4.4. Manual Data Import

If a different type of dataset is to be analysed, the manual Import feature has to be used. Make sure the file fits the following premises:

- the file has to be written in ASCII-Code
- the relevant data is below the row with the value names, any data above the name-row will be ignored
- every row of data starts on a new line, continuous data is not supported
- every value on each row is separated with the same separator (Space, Semicolon, Underscore...)

- the decimal-separator needs to be a dot, not a comma, as ProSi Post is not able to interpret commas as decimal-separator
- the data has to be in one block - any data below an empty line will be ignored so this can be used to store additional information that will not be processed during the reading of the data

Upon clicking "Import Data" from the Data menu, one is prompted to choose the database. Locate the file and click "OK". The following input screen should look similar to Figure 3.22.



row number input data: /home/ProSi_Oberflaeche/complete.out

row number					
1	SHOT_NR	NL14_flow_rate	NL13_flow_rate	NL09_flow_rate	LE05_
2	1	0.6060900000	0.5693400000	1.3327000000	1.0
3	2	0.7089400000	0.6682000000	1.5485000000	1.1
4	3	0.5854600000	0.5534200000	1.2533000000	0.9
5	4	0.6693400000	0.6341100000	1.4841000000	1.1
6	5	0.6605400000	0.6337300000	1.5325000000	1.1
7	6	0.6977900000	0.6653400000	1.5359000000	1.1
8	7	0.7621100000	0.7279600000	1.7244000000	1.3
9	8	0.7046700000	0.6765500000	1.6588000000	1.2
10	9	0.6914400000	0.6576900000	1.5017000000	1.1

Definition of the import filter

number of the row with the value name: 1

columne separator: |

variables	selected variables (left mouse-click)
SHOT_NR	NL14_flow_rate
NL14_flow_rate	NL13_flow_rate
NL13_flow_rate	NL09_flow_rate
NL09_flow_rate	LE05_flow_rate
LE05_flow_rate	LE24_flow_rate
LE24_flow_rate	LE14_flow_rate
LE14_flow_rate	NL08_flow_rate
NL08_flow_rate	LE15_flow_rate
LE15_flow_rate	LA15_flow_rate
LA15_flow_rate	LA24_flow_rate

Figure 3.22. Import Data Window

If an error is received while trying to open the file, please check whether the input file fulfills all of the above premises.
Back to the Input-screen. In the upper part one can scroll through the database. By doing this, it can be confirmed that the right file has been opened.
Now the number of the row which has the value names in it has to be specified. In the example this is done by merely typing a "1" into the appropriate input field. ProSi Post will only read value-rows below this row. If there is no row containing value names, just enter "0" into the input field. ProSi Post will now begin reading data beginning in the first row and will assign its own value names, for example "1, 2, 3, ...".
Next is to define the separator symbol that separates the single columns in one row. In the example case there is no actual sign but just empty spaces, specifically Tabulators, between the values. So the input in the field "Column Separator" in that case was to press the space-bar once. If unsure what the actual separator is that the program used to create the database, just check the dataset in the upper

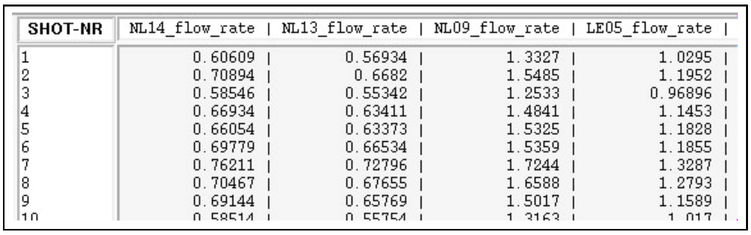
part of the screen. As mentioned in the premises to use this feature before, a separator can be about anything. Most commonly used are Spaces, Semicolon, Underscores ... But even numbers or letters are possible. Though it is highly recommended *not* to use numbers or letters as they obviously could interfere with the actual values or variable names.
After entering the column separator one should now find a list of all the available input-variables in the lower left list, just as seen in the example. From there either specific variables for ProSi Post to analyse can be picked by just clicking on them or simply click on "Select All Variables" to copy the entire variable list from the left to the right. In case a deletion of certain selected variables is required, simply click on it in the right list. Now everything is set to give ProSi Post it's input to work with. Click on "Read the Data".
Another note on using manual input: In case the data-file contains a column named "Shot Number" or anything similar to this and it is selected during for the read-process, then ProSi Post will treat this column as a regular data-column. So please make sure, that it is not included on the right list of read-data before clicking "Read the Data". When using the automated read-method as described in section 3.4.3, ProSi Post recognizes the number-column by itself.
ProSi Post now prompts the main window. If anything went wrong during the reading process one will be prompted with an appropriate error-message and should recheck whether the file fulfills all of the premises. A common error received at this point is that of "expected floating-point number but got ...". In that case a decimal-point has not been used in the input file for decimal separation. Please create a new input file that uses points.
One is now set to use the tools ProSi Post provides to analyse your data.

3.4.5. Refresh Data

The "Refresh Data"-Button is helpful when using ProSi Post while the *.out file is still in use by ProSi, for example when calculating a large amount of shots and an early analysis of the already calculated data is of interest. By clicking the "Refresh Data"-Button ProSi Post will reread the data file and include any new data into it's current displays.

3.4.6. Values

Now that the data is read into ProSi Post, it can start analysing the file. The first choice is to click on "Values". All this will do is to show the file as it has been read by ProSi Post (Figure 3.23). No calculation are done here.

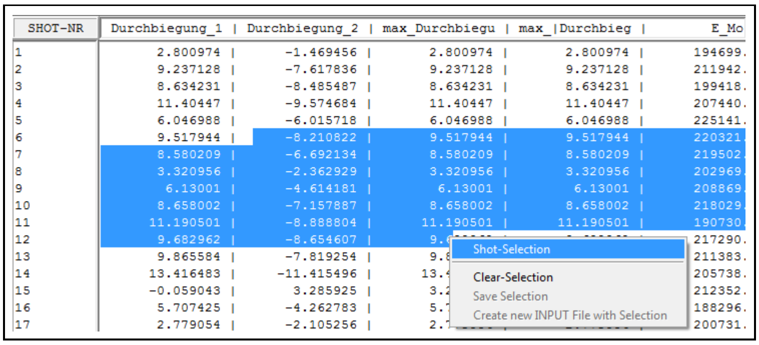


SHOT-NR	NL14_flow_rate	NL13_flow_rate	NL09_flow_rate	LE05_flow_rate
1	0.60609	0.56934	1.3327	1.0295
2	0.70894	0.6682	1.5485	1.1952
3	0.58546	0.55342	1.2533	0.96896
4	0.66934	0.63411	1.4841	1.1453
5	0.66054	0.63373	1.5325	1.1828
6	0.69779	0.66534	1.5359	1.1855
7	0.76211	0.72796	1.7244	1.3287
8	0.70467	0.67655	1.6588	1.2793
9	0.69144	0.65769	1.5017	1.1589
10	0.58514	0.55754	1.3163	1.017

Figure 3.23. ProSi Post Values Window

The view of the values allows to select specific shots. To do this, use the mouse to mark one or more shots and press the right mouse button. As shown in figure 3.24, a context menu will appear. By choosing "Shot-Selection", the marked shot will get a flag in the first column and will appear as selected shots in the 2D- and 3D-Ant-Hill-Plot (which are described later). The option "Clear-Selection" is used to deselect all shots. ProSi-Post is also able to export all selected shots to a new out-file or an INPUT-file. But be aware that it's not possible to edit the name of the new INPUT-file.

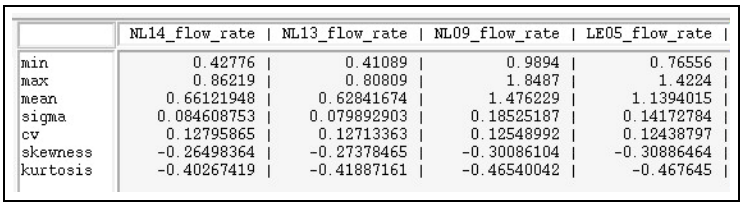
The original INPUT-File (if there is one in the destination directory) will be renamed with the suffix ".old".



SHOT-NR	Durchbiegung_1	Durchbiegung_2	max_Durchbiegu	max_|Durchbieg	E_Mo
1	2.800974	-1.469456	2.800974	2.800974	194699.
2	9.237128	-7.617836	9.237128	9.237128	211942.
3	8.634231	-8.485487	8.634231	8.634231	199418.
4	11.40447	-9.574684	11.40447	11.40447	207440.
5	6.046988	-6.015718	6.046988	6.046988	225141.
6	9.517944	-8.210822	9.517944	9.517944	220321.
7	8.580209	-6.692134	8.580209	8.580209	219502.
8	3.320956	-2.362929	3.320956	3.320956	202969.
9	6.13001	-4.614181	6.13001	6.13001	208869.
10	8.658002	-7.157887	8.658002	8.658002	218029.
11	11.190501	-8.888804	11.190501	11.190501	190730.
12	9.682962	-8.654607	9.6		217290.
13	9.865584	-7.819254	9.8		211383.
14	13.416483	-11.415496	13.4		205738.
15	-0.059043	3.285925	3.2		212352.
16	5.707425	-4.262783	5.7		188296.
17	2.779054	-2.105256	2.7		200731.

Figure 3.24. ProSi Post Values Window with selections

3.4.7. Basic Statistic



	NL14_flow_rate	NL13_flow_rate	NL09_flow_rate	LE05_flow_rate
min	0.42776	0.41089	0.9894	0.76556
max	0.86219	0.80809	1.8487	1.4224
mean	0.66121948	0.62841674	1.476229	1.1394015
sigma	0.084608753	0.079892903	0.18525187	0.14172784
cv	0.12795865	0.12713363	0.12548992	0.12438797
skewness	-0.26498364	-0.27378465	-0.30086104	-0.30886464
kurtosis	-0.40267419	-0.41887161	-0.46540042	-0.467645

Figure 3.25. Basic Statistic Window

The first time ProSi Post calculates anything is when "Basic Statistic" is clicked (Figure 3.25). At first it will display the minimum, the maximum and the mean values of each of the variables. Furthermore listed are the Sigma-value, called the Standard Deviation of a distribution, the cv-value, which is the coefficient of variation, the Skewness and the Kurtosis of each variable.
For detailed information about the mathematical background, please see to the Appendix A.

3.4.8. Cumulative Density

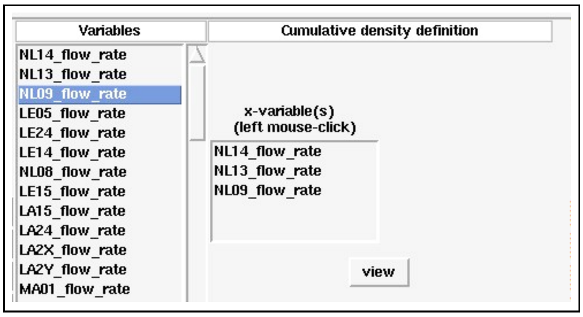
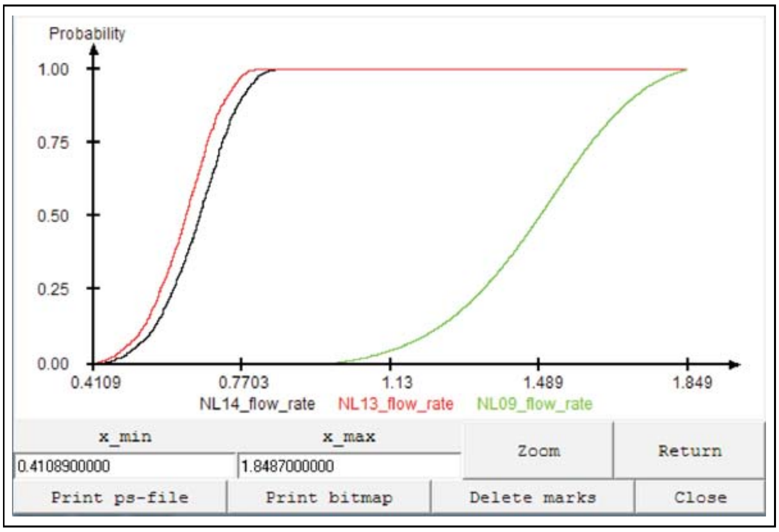


Figure 3.26. Cumulative Density Definition Window

The "Cumulative Density" feature of ProSi Post will plot graphs for the cumulative density of each input variable (Figure 3.26). First of all a choice has to be made which variable the cumulative density function needs to be drawn for. Multiple variables can be chosen, but be careful, if their value-range differs to much, ProSi Post will not be able to output a useful graph.
To choose a variable to show the cumulative density for, simply click on it and it will be listed in the right list. To remove a variable from the right list, simply click on it in the right list. For the next step just click "View" and the graph or graphs will be plotted on screen (see Figure 3.27). Now the graph can either be saved to a separate file using the post script format by clicking on "Print ps-file" or as a picture using the gif format by clicking "Print gif-file". The



x_min: 0.4108900000

x_max: 1.8487000000

Figure 3.27. Cumulative Density Function Plot Window

window is discarded by clicking “Close”. ProSi Post will then return to the previous screen, where either different variables can be picked or one may continue to the next part of ProSi Post-calculations.

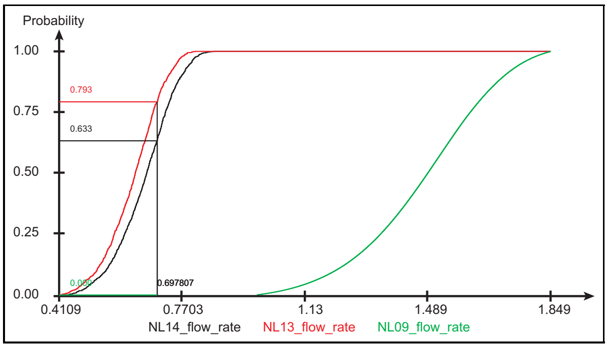


Figure 3.28. Cumulative Density with ruler

If one hover the mouse over the cumulative density plot, a ruler will appear and show the specific probability of the value on the X-Axis, as seen in figure 3.28. A right-click on the plot will fix the ruler at the current position, which allows to set several marks on the plot. To delete the marks, use the button “Delete marks”. Furthermore one can zoom into the plot by pressing the left mouse-button and raising a rectangle (with the mouse-button still pressed). Moreover one can use the entries on the bottom of the window to enter the left and right border of the X-Axis. Use the “Zoom” Button to apply the input.

3.4.9. Histogram

The “Histogram” will show the distribution of the values of each variable. The “Histogram” screen (Figure 3.29) works similar to the “Cumulative Density” with the difference of another input-field, namely the “Number of Buckets”. The “Number of Buckets” simply means the amount of bars included within the histogram. The default value is calculated by $\sqrt{Number\,of\,Shots}$, rounded to the next integer. Changing this value can greatly change the appearance of the histogram and can used to emphasize certain values.

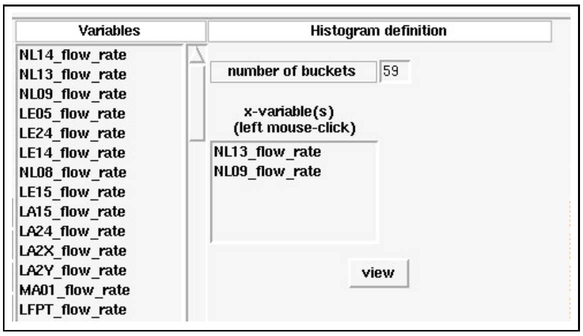


Figure 3.29. Histogram Definition Window

Now, choose the variables to be shown from the left list by clicking on them. To remove a variable from the list, simply click on it in the right list. The maximum number of variables that can be depicted in one plot is 5. Now the number of buckets may be changed. More buckets will obviously result in a more detailed histogram, but will also take more time to compute, depending on the amount of data and the computer ProSi Post runs on.

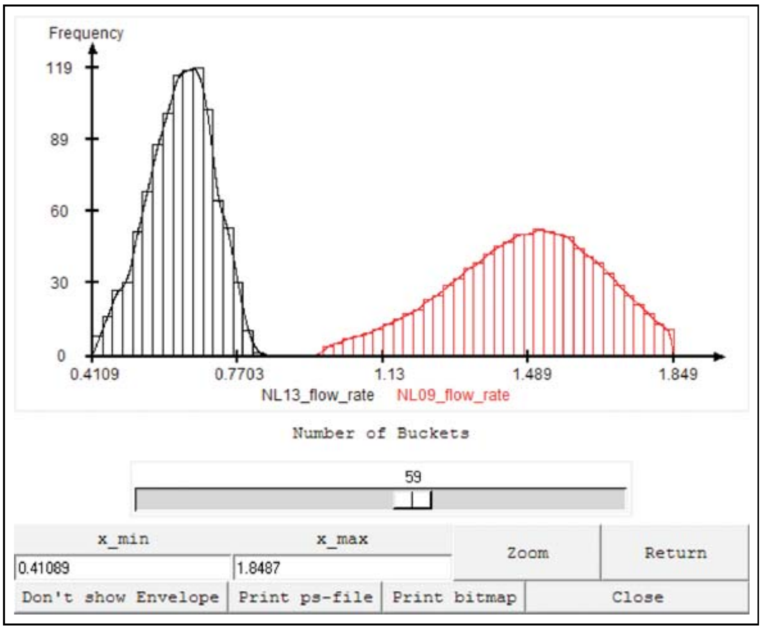


Figure 3.30. Histogram Plot Window

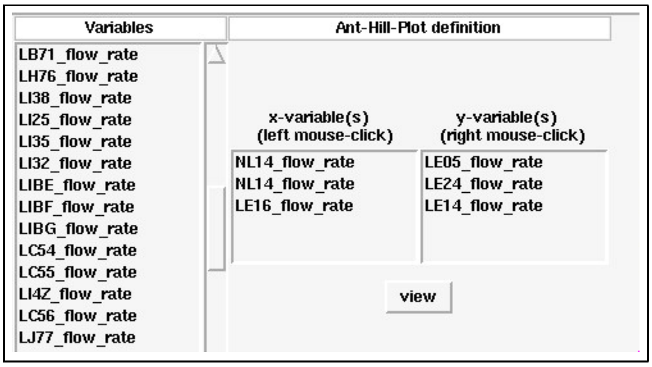
Followed by a click on "View" the actual histogram is plotted on screen. As shown in the Figure 3.30, there are two options to manipulate the graph shown in the graph-window.
With the slide-bar the number of buckets can be adjusted as fitting. The maximum number of buckets is limited to 100. Be aware, that adjusting the number of buckets with the slider may take a while to compute if using large amounts of shots for each variable. By clicking on "Show Envelope" the line connecting each bucket will be toggled on, as seen in the example. By default this is turned off when first showing the graph.

It's also possible to manipulate the limits of the plot. Which offers the chance to create plots which can be compared with other plots, as all plots can have the same limits. To do so enter the new limits as "x_min" and "x_max" values and press "Zoom" to confirm the input. Be aware, that the new limits must be outlining the bars of the plot. Thus, it's not possible to enter an x_max value which is below the original x_max value or to enter an x_min value which is above the original x_min value.

Finally it is again possible to save the current graph as a ps-file or gif-file for later viewing.

3.4.10. 2D-Ant-Hill-Plot

The "2D-Ant-Hill-Plot" feature allows to plot any variable of the database against any other. The procedure to do this is as follows (see Figure 3.31). Pick the variables to be placed on the axis of abscissae with the left mouse-button. One should now see those variables in the respective list. With the right mouse-button, the variables that are to be put on the axis of ordinates are chosen. To remove any variable from the left list, simply click on it with the left mouse-button within the left list. If a wrong variable was entered into the right list, use the right mouse-button to remove it. Up to 5 combinations may be entered, but be sure to have the same amount of variables in each list as one value obviously cannot be plotted against nothing. The combination of which variable will be plotted against which other variable depends on the order they are being clicked into the lists. In the example all the first two combinations to be plotted will be NL14 vs. LE05 and NL14 vs. LE24, the third plot will depict LE16 vs. LE14. By clicking on "View" the Ant-Hill-Plot will be displayed (see Figure 3.32 (left)). ProSi Post now chooses an ideal window that will picture all of the data points. The scaling of the axis may be changed to zoom in or better fit specific data. For once that can be done by clicking and holding the left mouse-button anywhere inside the graph and then dragging it to create a box that the graph will zoom into. The other option to change the viewsize is to manually insert numbers into the x_min, x_max, y_min and y_max fields. After entering the numbers into those fields, simply click "Zoom" and ProSi Post will resize the graph-window with the new border-parameters. To get back to the original view, simply click on "Return".



LB71_flow_rate
LH76_flow_rate
LI38_flow_rate
LI25_flow_rate
LI35_flow_rate
LI32_flow_rate
LIBE_flow_rate
LIBF_flow_rate
LIBG_flow_rate
LC54_flow_rate
LC55_flow_rate
LI4Z_flow_rate
LC56_flow_rate
LJ77_flow_rate

x-variable(s) (left mouse-click)	y-variable(s) (right mouse-click)
NL14_flow_rate	LE05_flow_rate
NL14_flow_rate	LE24_flow_rate
LE16_flow_rate	LE14_flow_rate

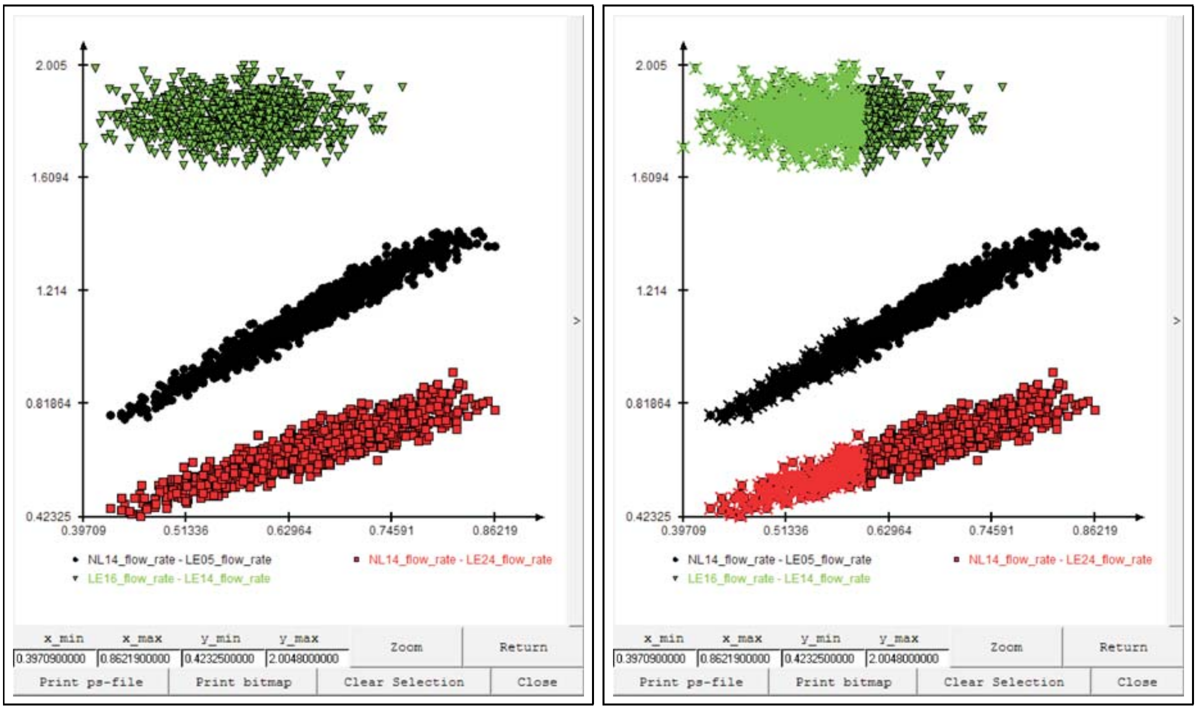
Figure 3.31. Ant-Hill-Plot Definition Window

A very useful feature is the selection of groups of specific points. To do this, drag a selection box across the area in which the plots of interest are located with the right mouse-button. The graph-window will now look something like the right-side of Figure 3.32.

Notice how the selected dots have turned into crosses. This step may be repeated several times in different areas of the graph window to select different groups of plots at once. To de-select all selected points, click on "Clear Selection". Be careful, this will reset the entire selection.

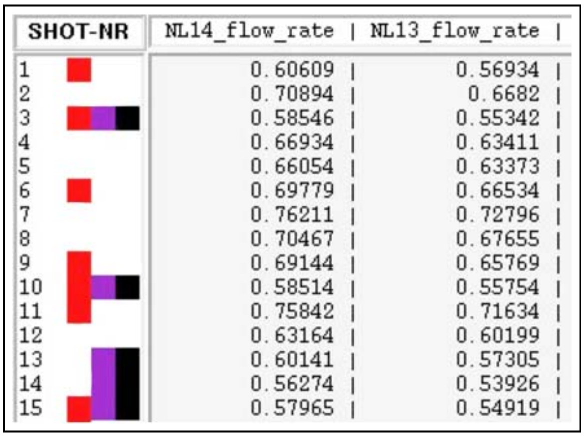
Also notice, that the main ProSi Post window running in the background changed to the "Values"-screen. If one scrolls down the list of shots, the within in the graph window selected shots will be shown in their respective colour.

On the right side of the window is a narrow but high button with an ">" on it. If the button is pressed an extra menu with two lists will deploy. In each list one can select an extra variable. The selected



x_min 0.3970900000 x_max 0.8621900000 y_min 0.4232500000 y_max 2.0048000000

Figure 3.32. left: Ant-Hill-Plot, right: Ant-Hill-Plot with Selection



SHOT-NR		NL14_flow_rate	NL13_flow_rate
1	🟥	0.60609	0.56934
2		0.70894	0.6682
3	🟥🟪⬛	0.58546	0.55342
4		0.66934	0.63411
5		0.66054	0.63373
6	🟥	0.69779	0.66534
7		0.76211	0.72796
8		0.70467	0.67655
9	🟥	0.69144	0.65769
10	🟥🟪⬛	0.58514	0.55754
11	🟥	0.75842	0.71634
12		0.63164	0.60199
13	🟪⬛	0.60141	0.57305
14	🟪⬛	0.56274	0.53926
15	🟥🟪⬛	0.57965	0.54919

Figure 3.33. Values screen with selected shots

variable of the upper list will be illustrated via rainbow color as third information. So, high values are colored red and low values blue etc.. The selected variable of the lower list is displayed as forth information via size variation in the same way. Right beside the plot a legend has been added to help to evaluate the extra information.
As usual, it is again possible to save the current view of the graph into a ps-file or a gif-file, by simply clicking on the respective button.

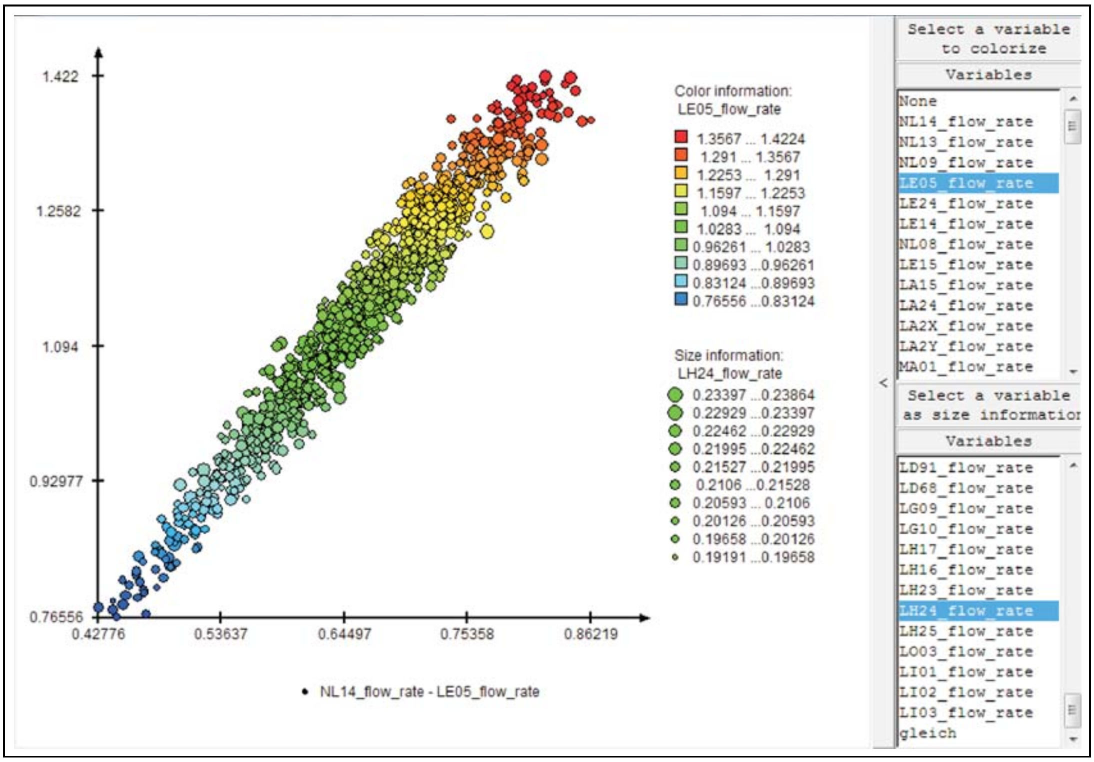
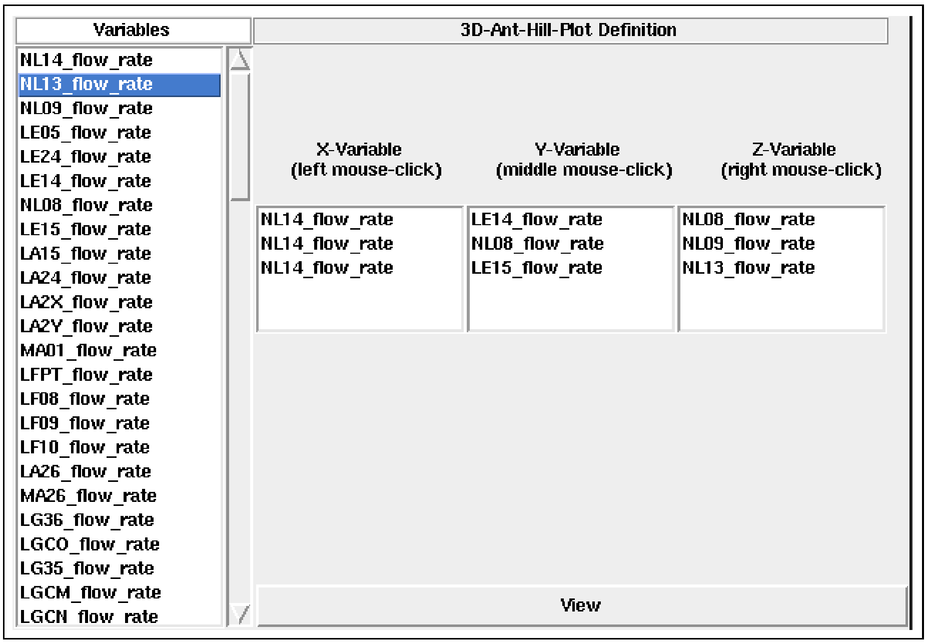


Figure 3.34. Ant-Hill-Plot with color and size information

3.4.11. 3D-Ant-Hill-Plot

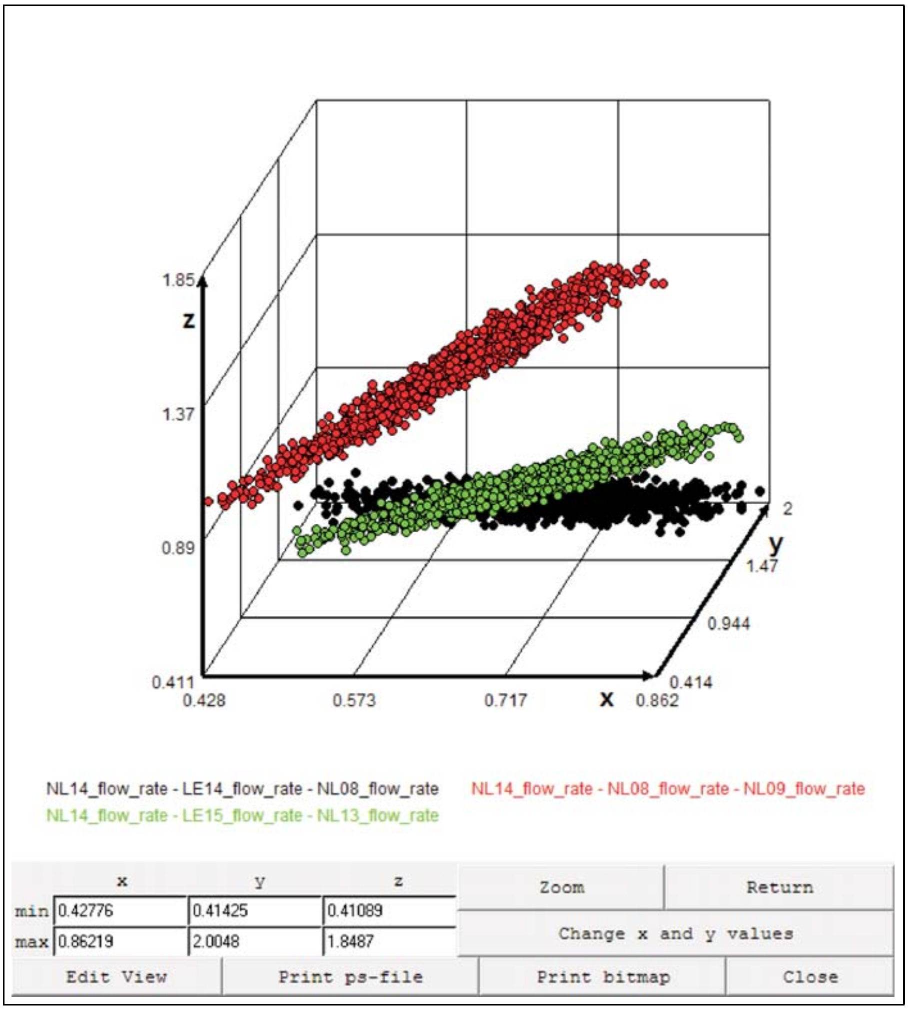
The "3D-Ant-Hill-Plot" works similar to the "2D-Ant-Hill-Plot" as described in 3.4.10. Upon selecting the feature "3D-Ant-Hill-Plot" the user is prompted with the selection screen (see Figure 3.35). To



X-Variable (left mouse-click)	Y-Variable (middle mouse-click)	Z-Variable (right mouse-click)
NL14_flow_rate	LE14_flow_rate	NL08_flow_rate
NL14_flow_rate	NL08_flow_rate	NL09_flow_rate
NL14_flow_rate	LE15_flow_rate	NL13_flow_rate

Figure 3.35. 3D-Ant-Hill-Plot Definition Window

select the desired variables into the respective columns use the 3 mouse buttons. By clicking the left mouse button on a variable in the left list this variable is selected into the list for the X-axis. The middle mouse button selects the Y-axis and the right mouse button respectively the Z-axis. It is possible to select up to 5 combinations of variables in any desired order. To de-select a variable in either one of the lists, click the corresponding mouse button on the variable that is to be deleted inside the axis-list of which it is to be removed from.
After all selections have been made, press the “View” button. A 3D-Ant-Hill-Plot (see Figure 3.36) will be displayed.


1.85
z
1.37
0.89
0.411
0.428
0.573
0.717
x
0.862
0.414
0.944
1.47
y
2
NL14_flow_rate - LE14_flow_rate - NL08_flow_rate
NL14_flow_rate - NL08_flow_rate - NL09_flow_rate
NL14_flow_rate - LE15_flow_rate - NL13_flow_rate
Zoom
Return
Change x and y values
Edit View
Print ps-file
Print bitmap
Close

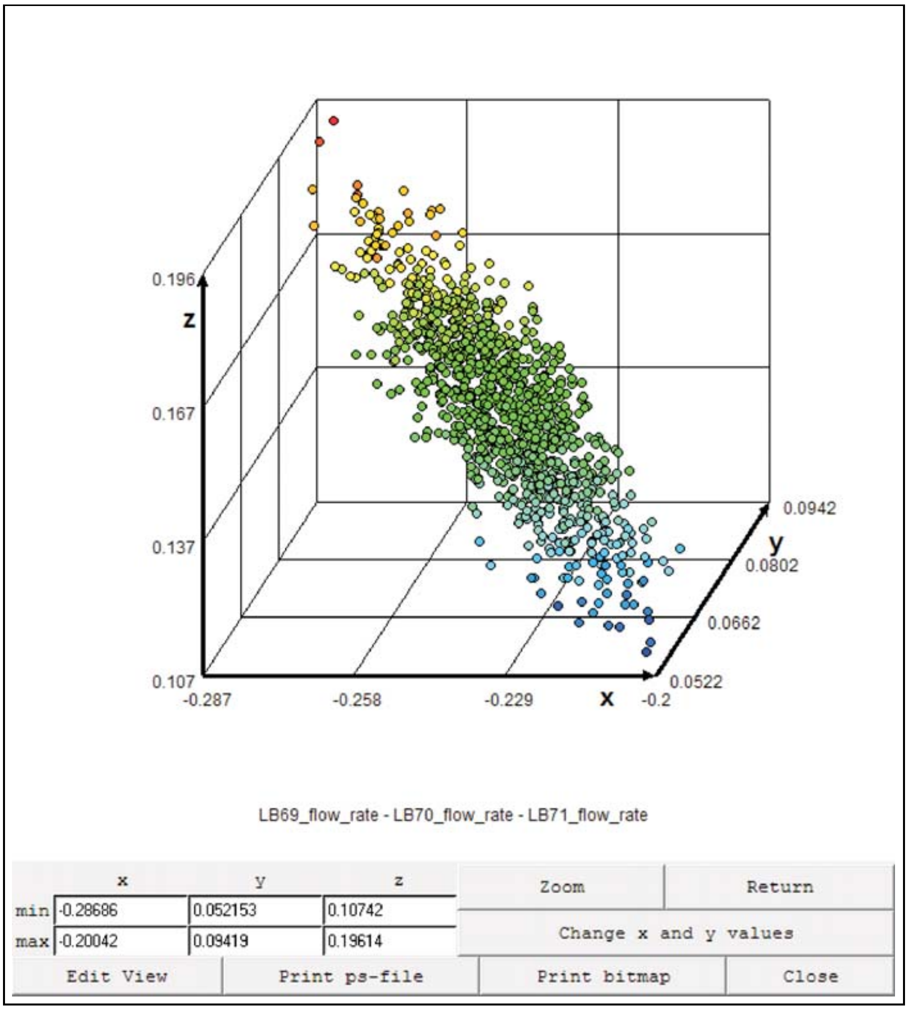
	x	y	z
min	0.42776	0.41425	0.41089
max	0.86219	2.0048	1.8487

Figure 3.36. 3D-Ant-Hill-Plot

ProSi Post now chooses an ideal window that will picture all of the data points. The scaling of the axis may be changed to zoom in or better fit specific data. This is done by typing the desired values into the x_min, x_max, y_min, y_max, z_min and z_max fields. After entering the numbers into those fields, simply click “Zoom” and ProSi Post will resize the graph-window with the new border-parameters. To get back to the original view, simply click on “Return”.
To rotate the angle of view, use the left mouse button, keep it pressed and move the mouse around to rotate the plot into the desired position. To return to the original view, click the “Return”-button.

With the mouse movement the plot can only rotate about two axes. Therefore one can use the button “Edit View” which will show an extra window with several methods to edit the view. There are 8 icons for some default views. Just press the icon to change to the given view. Below these icons there are 3 sliders for each of the three axes. It will always turn around the actual position of the specific axis. To move the plot, use the right mouse button. Keep it pressed and pan the plot into the desired position.

ProSi Post chooses a different colour for every combination of variables. If just a single set of variables is selected, ProSi Post uses different colors within one single plot representing the height on the z-axis for a better 3-dimensional view. (See Figure 3.37)



	x	y	z
min	-0.28686	0.052153	0.10742
max	-0.20042	0.09419	0.19614

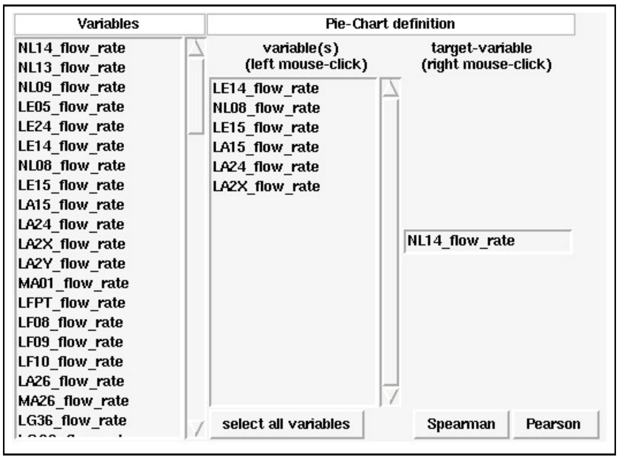
Figure 3.37. 3D-Ant-Hill-Plot

The “Change x and y axes” button does what its name implies, it swaps the orientation of the x and y axis. This is useful as one can now look at the 3D-plot from a different angle.
As in all other parts of ProSi Post that incorporate graphics, it is possible to export the current view into a file. For doing this, click either the button “Print ps-file” or “Print gif-file” and a selection window for saving the file in the corresponding file format will be opened.
Note that in 3D-Ant-Hill-Plots it is not possible to select shots by using a drag box as it is possible in a 2D-Ant-Hill-Plot as described in 3.4.10. The 3D-Ant-Hill-Plot is merely intended to give a possibility to compare 3 different variables to each other. In some cases it can be very hard to actually get a 3-dimensional view, thus for actual evaluation of data, the 2D-Ant-Hill-Plot as described in 3.4.10

should be preferred.

3.4.12. Pie-Chart

Using the “Pie-Chart” feature, ProSi Post will show a pie-chart of how a selected amount of variables correlates with a target-variable. It is possible to show either the Spearman-Rank-Correlation or Pearson’s Correlation-Coefficient between the variables. Within the different parts of the pie-chart, the actual values of the coefficient of correlation will be written.
First of all choose the variable which all other chosen variables will be related to. This is done by clicking on the specific variable on the left with the right mouse-button. It should now be listed in the appropriate field on the right. To choose the variables of which the correlation to the target-variable is to be placed inside the pie-chart, simply click on them on the list with the left mouse-button. The selection screen will look similar to this now: It is possible to insert all variables at once by clicking “Select all variables”. By doing this all variables, except the selected target-variable will be put into the selection list. If no target-variable has been selected at this point, one can select it from the left list by clicking the right mouse-button on it. This variable will then be removed from the selected variables list and is put into the field on the right.



Variables:
NL14_flow_rate
NL13_flow_rate
NL09_flow_rate
LE05_flow_rate
LE24_flow_rate
LE14_flow_rate
NL08_flow_rate
LE15_flow_rate
LA15_flow_rate
LA24_flow_rate
LA2X_flow_rate
LA2Y_flow_rate
MA01_flow_rate
LFPT_flow_rate
LF08_flow_rate
LF09_flow_rate
LF10_flow_rate
LA26_flow_rate
MA26_flow_rate
LG36_flow_rate

variable(s) (left mouse-click):
LE14_flow_rate
NL08_flow_rate
LE15_flow_rate
LA15_flow_rate
LA24_flow_rate
LA2X_flow_rate

target-variable (right mouse-click):
NL14_flow_rate

Figure 3.38. Pie-Chart Definition

To remove any of the variables, proceed as before. Simply click on the variable to be deleted within the lists on the right. To remove the target-variable, click on it with the right mouse-button.
Now choose between Spearman or Pearson correlation by pressing the respective button. The output for the example will look like this. Inside the pie-chart the correlation of the selected variables towards the target-variable is written into the parts of the pie-chart. With large amounts of variables one may end up with a lot of very small pie-sections that will not be given a specific colour anymore, since the correlation is too small.
The difference between the two types of correlation coefficients can be seen immediately. The example pictures Spearman on the right and Pearson on the left. The actual difference between the two types of correlation is not subject of this manual and shall not be discussed any further. It is possible to save the pie-charts into a ps-file or a gif-file via the respective button below the pie chart.

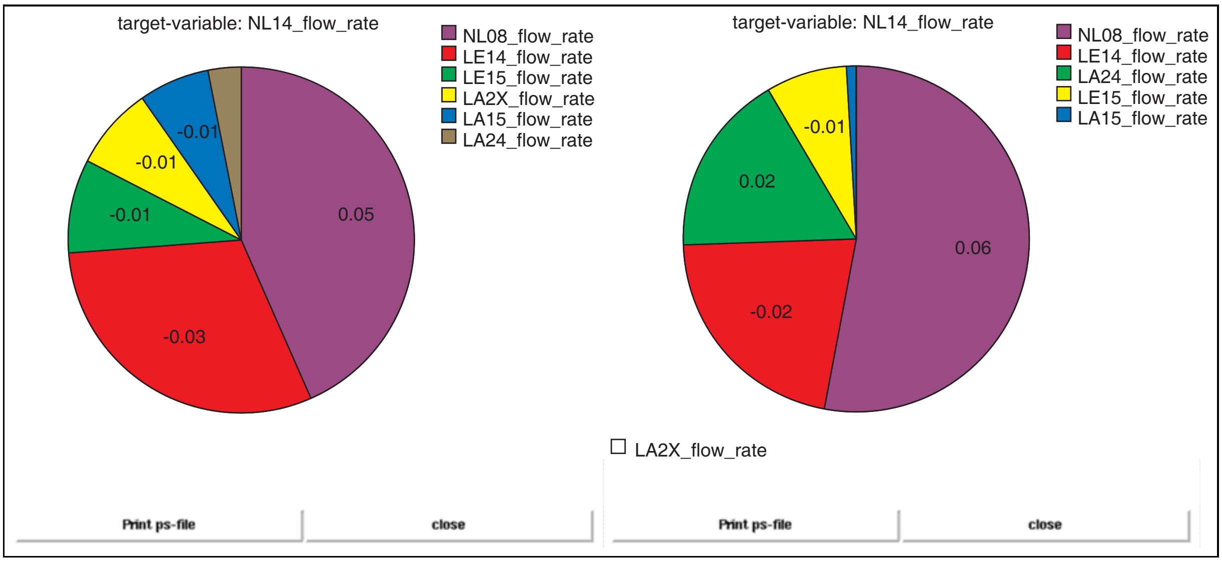


Figure 3.39. Pie-Charts (Pearson on the left, Spearman on the right side)

3.4.13. The Correlation Matrix

In this part of ProSi Post, the program will create a matrix that will show the correlation between any variable of the entire database. One can choose specific variables to be put into the matrix or simply click "Select all variables". This will produce a clearly arranged symmetrical matrix. The example database used in this manual has quite a large amount of variables and thus produces a rather big matrix when all variables are selected. The elements of the matrix describe the correlation between its components. Grey-shades depict nonlinear correlation coefficients and colours stand for the Spearman Rank Correlation. The exact definition of every shade of colour is shown on the right side.

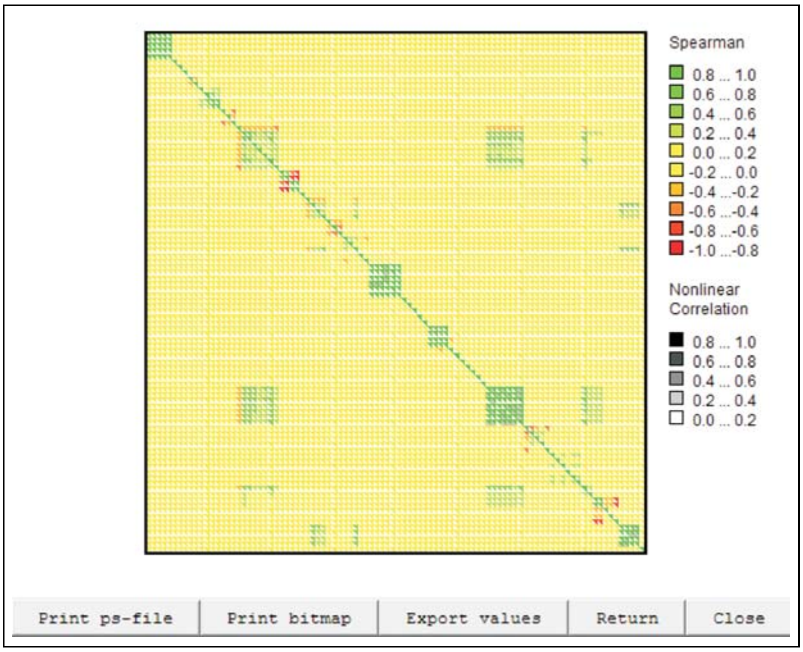
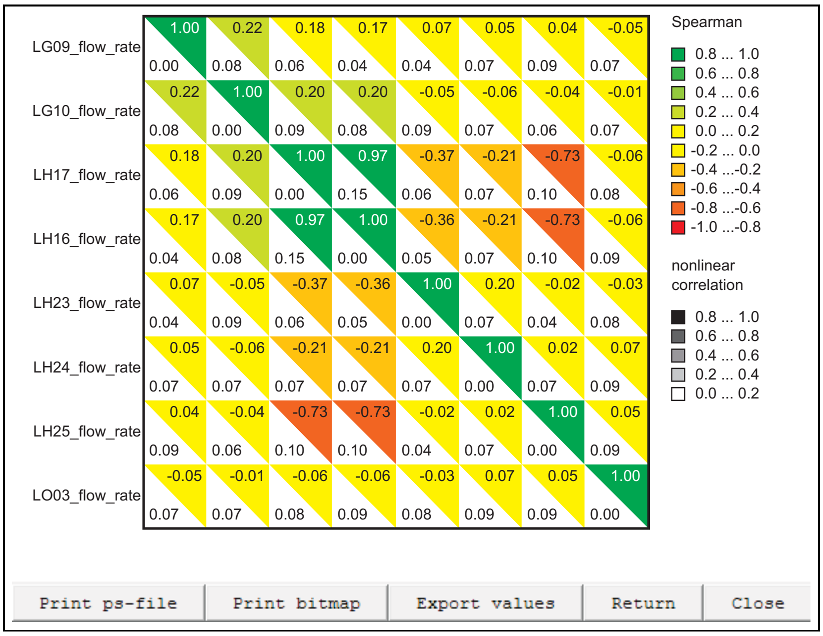


Figure 3.40. Correlation Matrix

Please note that nonlinear correlation coefficients can only be calculated with datasets that contain at least 500 shots. If using data consisting of less than 500 shots, the nonlinear correlation section on the right will not be visible and within the matrix, the nonlinear correlation will be displayed as "-1". The correlation matrix for both linear and nonlinear correlations is a symmetric matrix.

It is now possible to get a more detailed view of one part of the matrix by clicking and dragging a box with the left mouse-button across an area of interest as was done with the Ant-Hill-Plot. The graph-window now changes to the selected elements of the matrix. Click on "Return" to reset the zoom-window. After a selection the resulting matrix-view may look like this:



LG09_flow_rate	1.00 0.00	0.22 0.08	0.18 0.06	0.17 0.04	0.07 0.04	0.05 0.07	0.04 0.09	-0.05 0.07
LG10_flow_rate	0.22 0.08	1.00 0.00	0.20 0.09	0.20 0.08	-0.05 0.09	-0.06 0.07	-0.04 0.06	-0.01 0.07
LH17_flow_rate	0.18 0.06	0.20 0.09	1.00 0.00	0.97 0.15	-0.37 0.06	-0.21 0.07	-0.73 0.10	-0.06 0.08
LH16_flow_rate	0.17 0.04	0.20 0.08	0.97 0.15	1.00 0.00	-0.36 0.05	-0.21 0.07	-0.73 0.10	-0.06 0.09
LH23_flow_rate	0.07 0.04	-0.05 0.09	-0.37 0.06	-0.36 0.05	1.00 0.00	0.20 0.07	-0.02 0.04	-0.03 0.08
LH24_flow_rate	0.05 0.07	-0.06 0.07	-0.21 0.07	-0.21 0.07	0.20 0.07	1.00 0.00	0.02 0.07	0.07 0.09
LH25_flow_rate	0.04 0.09	-0.04 0.06	-0.73 0.10	-0.73 0.10	-0.02 0.04	0.02 0.07	1.00 0.00	0.05 0.09
LO03_flow_rate	-0.05 0.07	-0.01 0.07	-0.06 0.08	-0.06 0.09	-0.03 0.08	0.07 0.09	0.05 0.09	1.00 0.00

Spearman

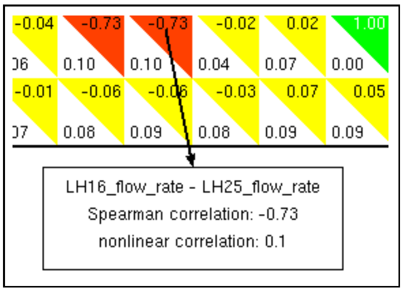
🟩 0.8 ... 1.0
🟩 0.6 ... 0.8
🟩 0.4 ... 0.6
🟨 0.2 ... 0.4
🟨 0.0 ... 0.2
🟨 -0.2 ... 0.0
🟧 -0.4 ...-0.2
🟧 -0.6 ...-0.4
🟥 -0.8 ...-0.6
🟥 -1.0 ...-0.8

nonlinear correlation

■ 0.8 ... 1.0
■ 0.6 ... 0.8
■ 0.4 ... 0.6
□ 0.2 ... 0.4
□ 0.0 ... 0.2

Figure 3.41. Zoom-Windows, Correlation Matrix

As shown above, there is now enough room to show the labels of the ordinate-variables. To find out what each element correlates to, simply click on one and a box with specific information for this element will pop up.



-0.04)6	-0.73 0.10	-0.73 0.10	-0.02 0.04	0.02 0.07	1.00 0.00
-0.01)7	-0.06 0.08	-0.06 0.09	-0.03 0.08	0.07 0.09	0.05 0.09

LH16_flow_rate - LH25_flow_rate
Spearman correlation: -0.73
nonlinear correlation: 0.1

Figure 3.42. Element Information, Correlation Matrix

Now click on an element of interest with the right mouse-button. A selection menu will pop up and prompts the user with several options: By clicking on Ant-Hill-Plot one will immediately be given an Ant-Hill-Diagram of the two components of that specific element plotted against each other.

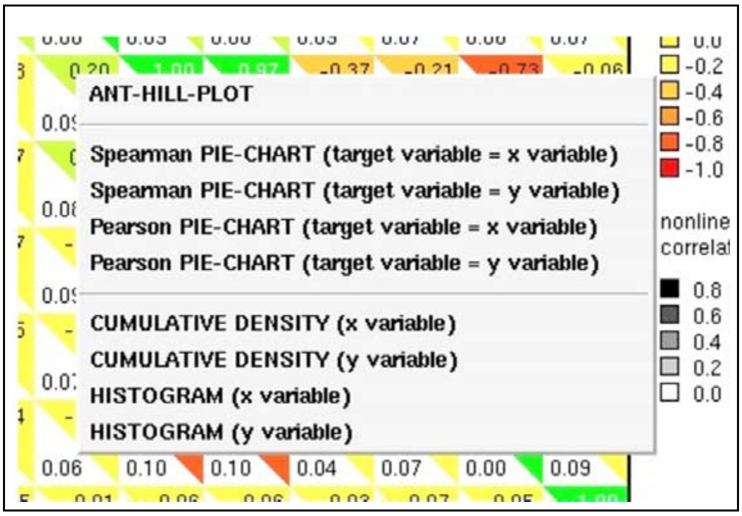

ANT-HILL-PLOT
Spearman PIE-CHART (target variable = x variable)
Spearman PIE-CHART (target variable = y variable)
Pearson PIE-CHART (target variable = x variable)
Pearson PIE-CHART (target variable = y variable)
CUMULATIVE DENSITY (x variable)
CUMULATIVE DENSITY (y variable)
HISTOGRAM (x variable)
HISTOGRAM (y variable)
0.20
1.00
0.97
-0.37
-0.21
-0.73
-0.06
0.06
0.10
0.10
0.04
0.07
0.00
0.09
-0.2
-0.4
-0.6
-0.8
-1.0
nonline
correlat
0.8
0.6
0.4
0.2
0.0

Figure 3.43. Pop-up Menu, Correlation Matrix

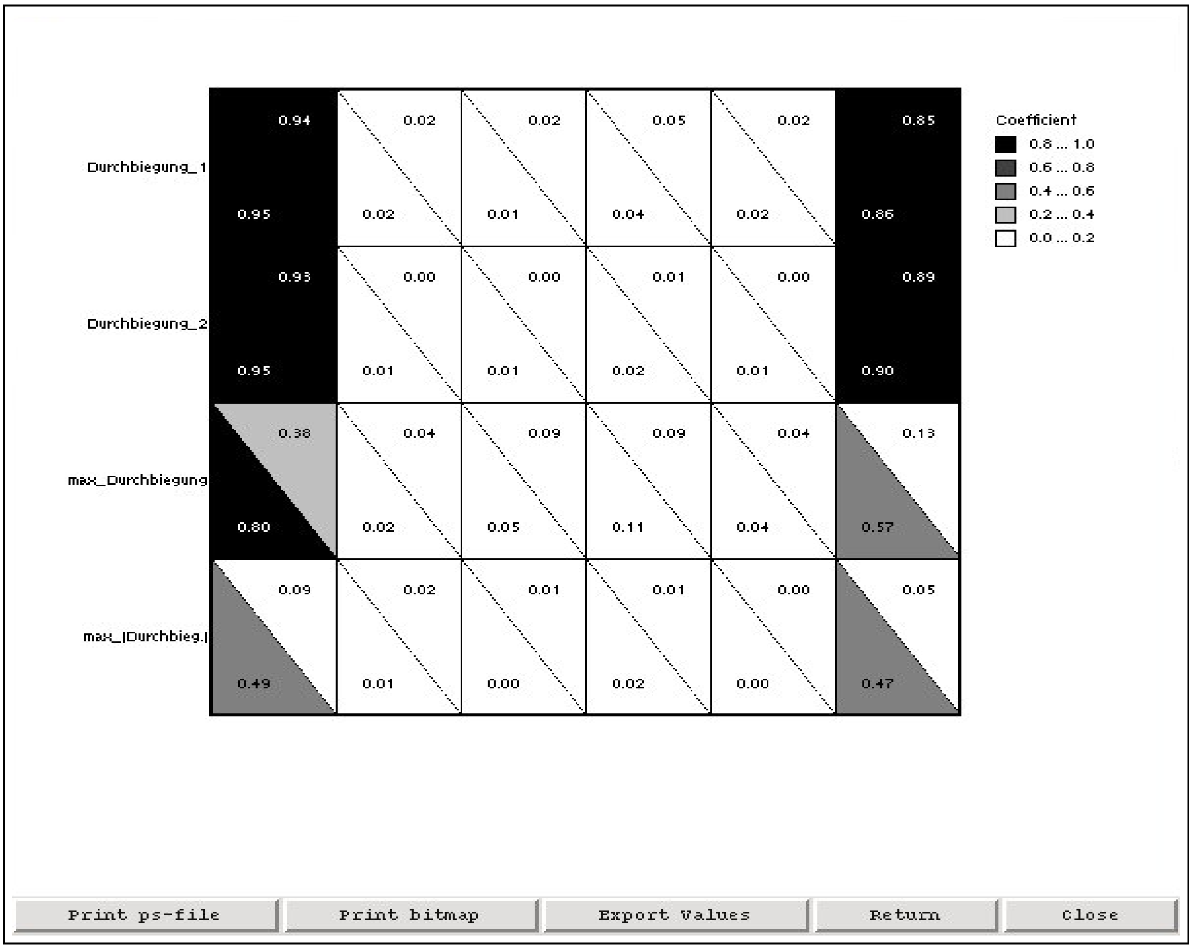
By clicking on one of the Pie-Chart options either the x or the y variable of the matrix-element are selected as the target variable for a pie-chart. One is then prompted with the input screen for a pie-chart. By default, the rest of the values of the current view of the correlation matrix is inserted as input-variables. This pre-selection can be changed manually as seen fit. This may be done as described in 3.4.12.

The options “Cumulative Density” and “Histogram” are fairly self-explanatory. For further details about the feature “Cumulative Density” please refer to 3.4.8. The feature “Histogram” is described in 3.4.9.

3.4.14. The CoI Matrix

CoI is the abbreviation for **C**oefficient **o**f **I**mportance. It evaluates the influence of input variables on an output variable. ProSi-Post finds an response surface of an output variable as a function of all input variables and the related coefficient of determination (CoD) to analyse the fitness of the response surface, which are shown as R^2-values in the first column of the matrix as seen in figure 3.44. The CoIs are shown in the columns behind the R^2 values and will be within 0 and 1, where 0 means no influence detected and 1 means that the variable has an big influence on the result variable. But be aware, that the CoD of the original response surface needs a good fit ($R^2 \geq 0.8$) to get reliable values for the CoI. Always start looking at the first column if the CoD is good enough.

In the section "CoI-Matrix" of ProSi-Post are the lists of input and result variables, which can not be altered. Right beside the lists, one can select the order of the response surface. The matrix can display two values for each element. Therefore it is possible to define two different orders and options of the response surface. The "improvement" option is an Box-Cox-Transformation (subsection 2.6.2), which may lead to better fitness of the original response surface. To calculate and show the CoI matrix, press the "View" button on the bottom of the window. The calculation may take a while.



Durchbiegung_1	0.94 0.95	0.02 0.02	0.02 0.01	0.05 0.04	0.02 0.02	0.85 0.86
Durchbiegung_2	0.93 0.95	0.00 0.01	0.00 0.01	0.01 0.02	0.00 0.01	0.89 0.90
max_Durchbiegung	0.38 0.80	0.04 0.02	0.09 0.05	0.09 0.11	0.04 0.04	0.13 0.57
max_|Durchbieg.|	0.09 0.49	0.02 0.01	0.01 0.00	0.01 0.02	0.00 0.00	0.05 0.47

Figure 3.44. CoI-Matrix

In the CoI matrix, as shown in figure 3.44, are the result variables in the rows and the input variables in the columns, where the first column represents the CoD value of the original response surface (with all input variables). Check them first ($R^2 \geq 0.8$) before evaluating the CoI values. The upper triangle of the squares always refers to the options (order and the improvement option) selected for the first value (in the setup window), where the lower triangle refers to the options selected for the second value.

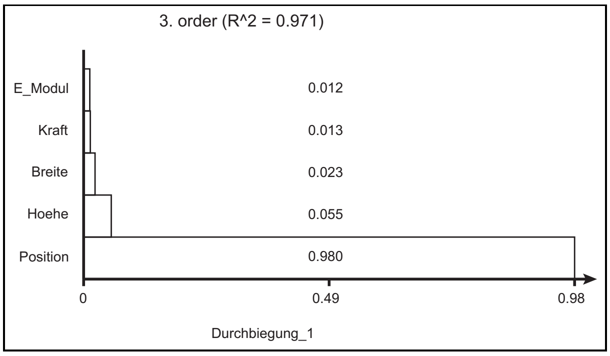


Figure 3.45. CoI-bar plot

To get the exact CoI value click with the left mouse button on one square. Additional options will be shown, if one right-clicks on the square. It is possible to access the already known functions as pie-charts, 2D Ant-Hill-Plots, histograms and cumulative densities. In addition the response surface can be displayed in a 3D Ant-Hill-Plot (which will be described later) and a CoI-bar plot. The bar-plot lists up to 15 input variables with there CoI values sorted by their values. The CoD value is shown as an R^2-value in the top. If there are more than 15 input variables only the 15 highest values are displayed and a writing on the top shows the last shown value.

Once, the CoI-Matrix has been calculated, one can find two new files (*.rs and *.rs.csv) in the directory of the loaded *.out-file. The *.rs-file contains the formula of the response surface as well as some statistical information of the selected result variables. Furthermore, at the end of the file there are listed the values of all variables in the same way as in the *.out-file itself. In the *.rs.csv-file one can find the same information, which can be imported in Microsoft Excel or an other spreadsheet. Instead of the name of the input variables, there are relative cell references in the formula of the response surface. That makes it possible to calculate the approximated value of the result variables for all realisations in an semi-automated way.

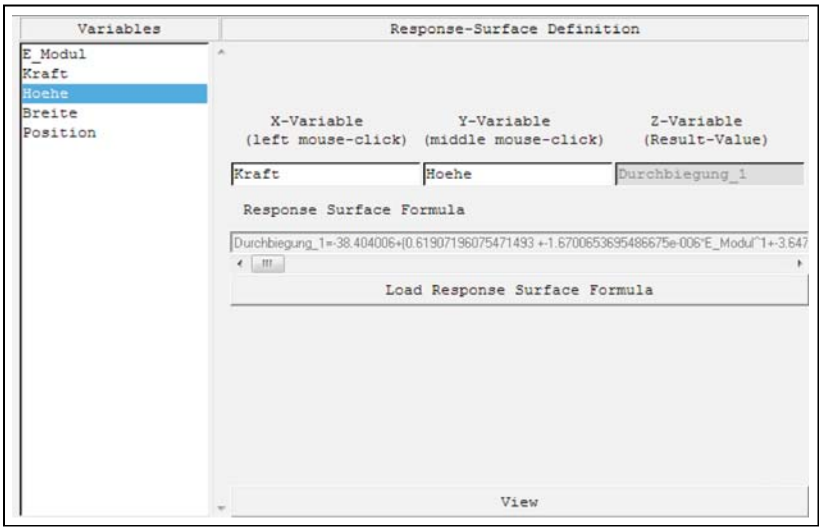
The “Export values” button allows to export the displayed matrix to an text file. As usual, it is again possible to save the current view of the graph into a ps-file or a gif-file, by simply clicking on the respective button.

3.4.15. Response Surface

The section “Response Surface” allows the user to evaluate RSM-calculations. Upon clicking “Response Surface” the user is asked to define the variables that are to be included within the display of the RSM as shown in 3.46. Select two appropriate variables for the x and y-axis. The z-axis will always be used for the result variable of the RSM. To include the result of the RSM calculation click the “Load Response Surface Formula” button. Now locate the appropriate subproject *.rs file that corresponds to the current subproject *.out file. After loading the file, the formula of the RSM is displayed in grey and the z-variable is set to the result value. Now click “View” to show the response surface.

The display window as shown in 3.47 of the response surface works the same way as the 3D-Ant-Hill-Plot. It is possible to rotate, pan and zoom the response surface to better understand the system. Another feature that could be useful for slower machines is the ability to lower the amount of colors displayed within the response surface. To decrease the amount of colors, press the middle mouse button and drag the mouse to the left. This will drastically reduce calculation time for rotating and panning the graphics window. After the desired view has been achieved, press the middle mouse button again and drag the mouse to the right to increase the color scale up to a maximum of 50 different colors.

The response surface is a multidimensional surface, which is reduced to a 3D surface to display it in



X-Variable (left mouse-click): Kraft

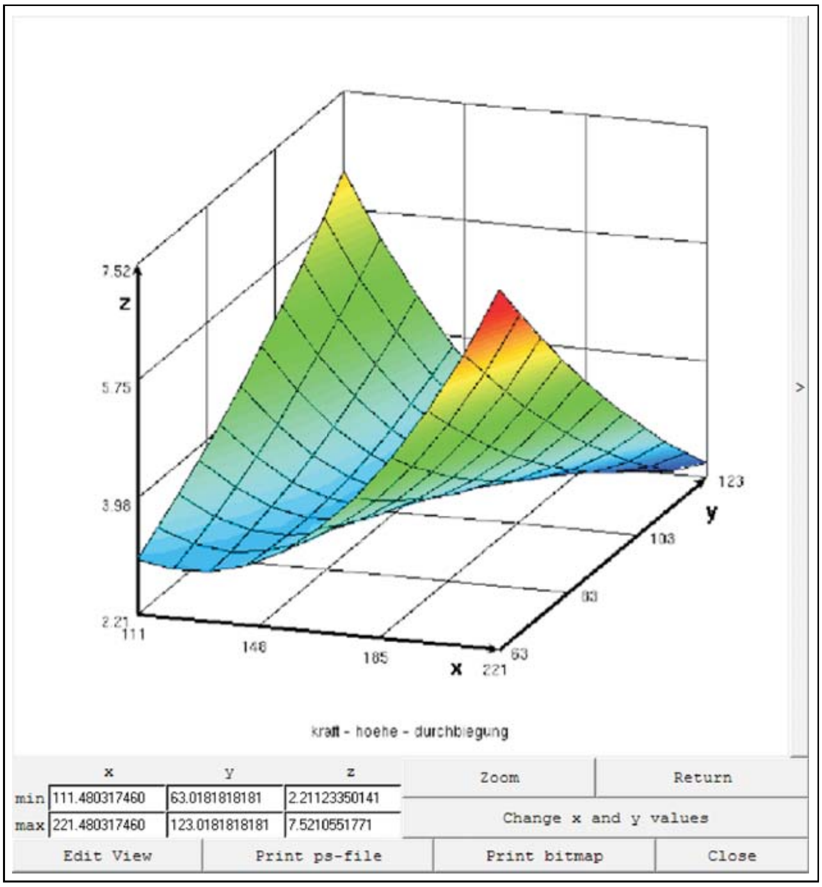
Y-Variable (middle mouse-click): Hoehe

Z-Variable (Result-Value): Durchbiegung_1

Response Surface Formula

Durchbiegung_1=-38.404006+(0.61907196075471493 +-1.6700653695486675e-006*E_Modul^1+-3.647

Figure 3.46. RSM Definition



	x	y	z
min	111.480317460	63.0181818181	2.21123350141
max	221.480317460	123.0181818181	7.5210551771

Figure 3.47. RS Example

the plot. As a consequence, the difference of one dot and the displayed response surface is not the difference of the dot with the real (multidimensional) response surface. To get an adequate information of the deviation one can use the color of the dots to show the error of the response surface. To do so, use the ">" button to deploy an extra menu. Press the "Show error" button, to get the error plot.

With the option “Signed error” one can switch to a view with signed error, as the default error plot reflects the absolute error. As shown in figure 3.48, the response surface itself is now printed in grey scales.

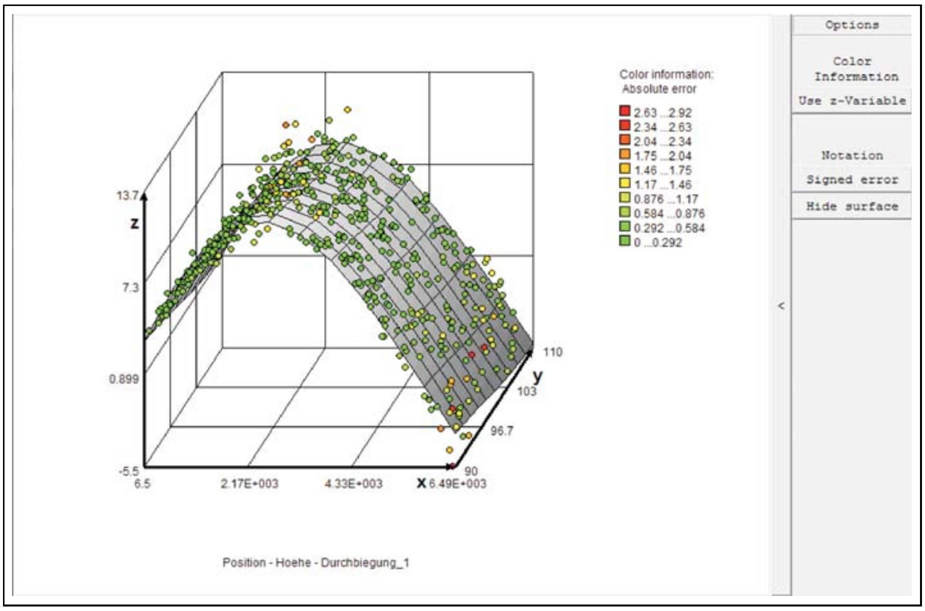
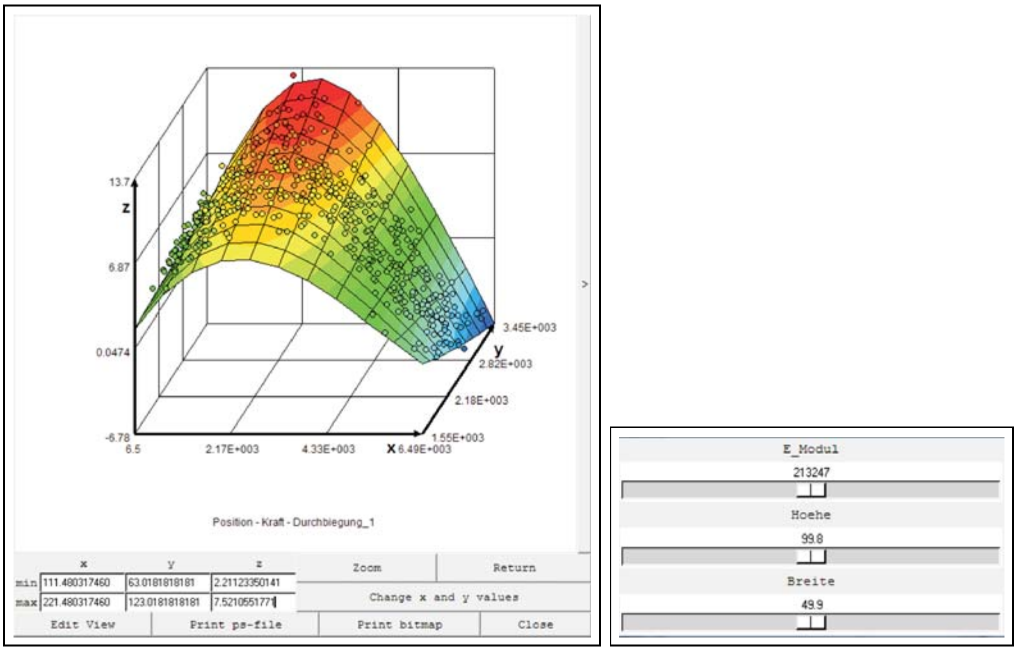


Figure 3.48. RS with error

The regular view of the response surface shows the response surface using the result variable and the two selected variables. To include further variables and study their effect onto the system use the drag bars as shown in 3.49 on the right. The Sampling Points of the response surface are shown as individual dots as seen in on the left. Again, use the zoom, rotate, pan and dragbars to better understand the response surface and to identify the most important variables of the system.



	x	y	z
min	111.480317460	63.0181818181	2.21123350141
max	221.480317460	123.0181818181	7.5210551771

E_Modul
213247

Hoehe
99.8

Breite
49.9

Figure 3.49. left: RS Sampling Points, right: RSM Dragbars

4. ProSi for advanced users

4.1. The ProSi directory and file structure

This chapter is dedicated to providing the user with an insight view of how the directory structure works within the ProSi product family and which files are written by ProSi.
As with any program, the most important directory is where ProSi and all of its components are installed. For information about the installation process, please refer to the installation instructions in chapter 1.
To get started with ProSi the tool ProSi Pre is used. Within which three directories are defined (see also Figure 4.1).

- directory for the deterministic model
- directory for the generated deterministic data sets
- directory for log and output files written by ProSi

The first directory labeled "deterministic model" contains all the needed files that ProSi will be working with. The content of this folder has to be created before anything is done with any ProSi-tool. Later on, when ProSi creates a directory for every shot, the deterministic model defined here will be copied into that directory. The second directory defined contains the generated data and can thus become rather large. For this example, it will just be called "/data". For every project that has been run on that computer, ProSi will create a subdirectory using the project-name within this data directory, e.g. "/project1". Within this project directory more subdirectories are being created. Every time ProSi runs a project, a random number containing of 8 digits is assigned to this specific project-run. This number is part of the directory name created within the project directory inside the data directory. The other part is the date on which the data was generated. In this case the subdirectory is called "/73996527_project1_24_12_06". Within this random number directory the last subdirectories are being created. For every single shot that ProSi creates, a new directory will be created. The name of this directory will be made up of the random number previously assigned, the date and the shot number. For the example: "/73996527_24_12_06_XXX", whereas XXX is the placeholder for the shot number. It will contain the new deterministic model created by ProSi for every shot. This is the full example: /data/project1/73996527_project1_24_12_06/73996527_24_12_06_XXX
The third directory defined within ProSi Pre is the log and output files directory. The directory structure within this directory is similar to the data directory. ProSi creates a subdirectory made up of the project-name and within that subdirectory it will store all the log and data files created by ProSi. The same numbering as used for the directories applies to the ProSi files. For every sub-project ProSi creates specific files which, following the example project number 73996527 would be

- the log file called "73996527_project1.LOG"
- the subproject specific input file called "73996527_project1.inp"
- the restart file called "73996527_project1.INPUT"
- the result file called "73996527_project1.out"
- the statistic file called "73996527_project1.STATISTIC"

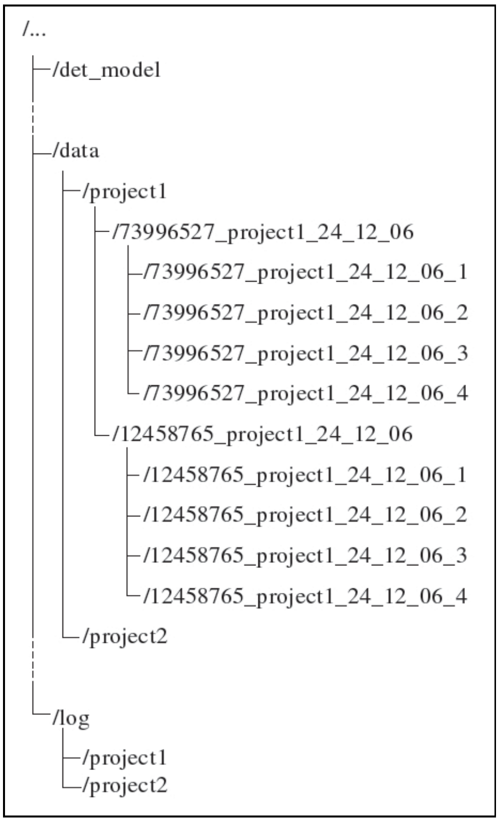

/...
/det_model
/data
/project1
/73996527_project1_24_12_06
/73996527_project1_24_12_06_1
/73996527_project1_24_12_06_2
/73996527_project1_24_12_06_3
/73996527_project1_24_12_06_4
/12458765_project1_24_12_06
/12458765_project1_24_12_06_1
/12458765_project1_24_12_06_2
/12458765_project1_24_12_06_3
/12458765_project1_24_12_06_4
/project2
/log
/project1
/project2

Figure 4.1. Directory Structure of ProSi

- in case of a RSM: the RSM output file 73996527_project1.rs
- in case of a combined RSM and MCS: combined output file “73996527_project1_RSM+MCS.out”

These files should not be altered in any external way and are only used for this specific sub-project.
The log file contains basic information about the project like the date of creation, the probabilistic method used and some more. Furthermore all messages and warnings that occur during a ProSi run are stored in this file. If ProSi encounters an error during the run, the log file is renamed from “73996527_project1.LOG” to “73996527_project1.ERROR” and the error message is written into the file.
The subproject input file is almost identical to the project input file, only the paths mentioned above are absolute paths including the subproject number, and at the end of each definition of a dependant parameter a format string determining the output format for this dependant parameter is written. This subproject specific input file can only be used to perform a restart of ProSi, not for a new start.
The restart file “73996527_project1.INPUT” contains the generated values for all probabilistic parameters. The data in this file is read in by ProSi and used to generate the dependant parameters.
The output file “73996527_project1.out” is updated after each finished deterministic calculation. It contains the result values scanned, sorted by shot numbers. Furthermore it contains the probabilistic input values for each shot (the same values that can be found in the “73996527_project1.INPUT” file) to allow sensitivity analysis and plots of result parameters against input values. This is also the file the graphical postprocessing tool ProSi Post reads in.
The file “73996527_project1.STATISTIC” contains some basic statistic analysis of the results as well as the input parameters. If failure criteria or fractil values have been provided for the results it also contains this information. This file can be used for a quick overview of the results and also to control that the distributions of the input parameters align with the distributions wanted.

The response surface output file “73996527_project1.rs” is only created if a response surface analysis is performed. It contains the results of the response surface analysis which are from top to bottom: the coefficients of the surface, statistics of the surface to judge the fitting and the sampling points used to calculate the surface. If the feature for improving the response surface has been activated, all response surfaces will be included within this file - the originally calculated ones as well as the improved ones. This enables the user to evaluate the improvements done.
The file “73996527_project1_RSM+MCS.out” contains all data belonging to the RSM calculation and the MCS that ones done based on this RSM.

4.2. The ProSi input file structure

This section explains in which way the information describing the probabilistic model is arranged in the ProSi input file. The information concerning the probabilistic model are saved in different paragraphs of the input file used by ProSi. This paragraphs do always start with a * followed by a keyword representative for the paragraph. In one or more lines following each keyword the parameters describing the model for ProSi are written. This “describing parameters” are separated by semicolons. The single paragraphs can be situated in arbitrary order within the ProSi input file. In the following each paragraph of the input file will be first introduced briefly in form of a table. This illustrates the number and the position of the “describing parameters”. Afterwards the paragraph will be described more in extensive.

4.2.1. Paragraphs of the input file needed for the preprocessing

For the preprocessing mode of ProSi the following paragraphs have to be defined in the ProSi input file:

- *method
- *doe
- *rsm/ *mcs/ (only one, depending on the method defined under *method)
- *directories
- *files
- *variables
- *dep_variables
- *correlation_matrix (not required if all correlations are to be set to zero)

*method “???”	Switch for the probabilistic method used. Options are “rsm”, “mcs”, “doe” or “rsm+mcs”

Table 4.1. Method Definition

Within the paragraph “*method” the particular probabilistic method is defined. Available options are “mcs”, “rsm”, “doe” or “rsm+mcs” respectively. In case ProSi is started without a method parameter as argument this paragraph is used to determine the ProSi run mode. Any method parameter provided as argument will cause ProSi to ignore this section. This means if for example ProSi is called using "prosi input_file -rsm" ProSi will switch to response surface calculation mode even if under method in the input file e.g. "mcs" is defined.

*doe "1"	"Design of Experiments", determines the design plan used to generate the deterministic calculation for a Response Surface Analysis. 1: This switch sets the design plan used. Options are: 1 - Full Factorial design, 2 - Box Benken Design, 3 - Central Composite Design, 4 - Simplex Design

Table 4.2. Design of Experiment Definition

If RSM has been chosen for the probabilistic method under the keyword "*method", the additional keyword "*doe" followed by the parameter for this method has to be written in the input file (see Table 4.2). The parameter determines the type of experimental design, which will be used for the RSM. At this time the supported types of experimental designs are the 2^{n_b} Design (1), the Box Benken Design (2) and the Central Composite Design (3) and the Simplex Design (4, only suitable for linear system responses). A description of these three types of experimental design can be found in section 2.6.1. The maximum allowed number of probabilistic variables is limited to 11 for the Full Factorial and the Central Composite Design and to 15 for a Box Benken Design.

*rsm "1" ;"2"	General parameters for a RSM (only needed if RSM has been chosen above) The second line has to contain the parameters: 1: Power of the system of equations for the RSM (1, 2 or 3 allowed) 2: Switch for automatic reduction of the Response Surface Options are: 0 - deactivated, 1 - activated

Table 4.3. RSM Definition

If RSM has been chosen for the probabilistic method under the keyword "*method", the additional keyword "*rsm" followed by the two general parameters for this method has to be written in the input file (see Table 4.4). The power of the response surface is determined by the first switch in this line. It is possible to calculate response surfaces of first to third order. The usage of third order response surfaces although possible is not recommended nor necessary for most deterministic models.

It must be pointed out, that for the calculation of the response surface a sufficient number of sampling points has to be available. The number of sampling points needed in dependency of the number of probabilistic variables can be seen in Table 2.1.

At the second position of the line, the user can decide if ProSi should automatically reduce the number of coefficients of the Response Surface. By activating this feature, ProSi will check how strong the influence of individual coefficients on the Response Surface is and then automatically set weak coefficients to zero, thus simplifying the entire formula for the Response Surface.

*rs_y_improvement y^(0.1 to 2)	Automatic improvement of the Response Surface using exponentiation of the result (y) value.

Table 4.4. Automatic Response Surface improvement by exponentiating the result value

Using the switch “*rs_y_improvement” a range for the exponents used during the automatic improvement is defined. ProSi calculates a number of different response surfaces using exponentiated result values y^a to achieve a better approximation of the system response. The range for the exponent a is defined here.

*mcs “1” ;“2”	General parameters for a MCS (only needed if MCS has been chosen above) 1: Number of deterministic shots for the MCS 2: Sampling method for the MCS (1 - SRS, 2 - LHS, 3 - DS)

Table 4.5. MCS Definition

If MCS has been chosen for the probabilistic method under the keyword “*method”, the additional keyword “*mcs” followed by the two general parameters for this method has to be written in the input file. The first value determines how many deterministic data sets (shots) have to be generated. The second value determines the method of sampling (see section 2.5.3 and 2.5.4).

*directories “1” ;“2” ;“3”	Directories for prosi to read out of and to write to. 1: Path to the original data (deterministic model) 2: Path for the generated input data sets (probabilistic model) 3: Path for ProSi files (input parameters, log-file, etc.)

Table 4.6. Directories Definition

The paragraph *directories defines the working directories used by ProSi. The first path points to the original data representing the deterministic model, which has to be analysed probabilistically. The manipulated deterministic models are written to the second path by ProSi. Thereby each model is written to a separate subdirectory. The names of these subdirectories include the date of the generation of the model and the number of the deterministic model which is saved in the subdirectory. In the third directory ProSi writes its own output files and the log-file. For more information on the ProSi data structure see section 4.1

*files “?” ;“?” ;...	Names of all files which are needed for the simulation. Filenames without the path, separated by “;”, all in one line

Table 4.7. Files Definition

All files needed for the deterministic model have to be written in the line following the keyword “*files”. ProSi manipulates this files and copies them into the subdirectories in which the manipulated deterministic models will be calculated.

In the paragraph “*variables” the single probabilistic variables of the probabilistic model are defined (see Table 4.8). Thereby every line describes one variable. At the first position the name of the variable as to be specified. The name is arbitrary, but must not contain any semicolons or arithmetic operators (+ - * /...).
At the second position the type of distribution is defined. The options are 1 for a uniform distributed, 2 for a normal distributed, 3 for a lognormal distributed and 4 for a triangular distributed variable. A description of these four types of distributions can be found in section 2.2.2.
At the third position the standard deviation (σ) has to be written if a normal or lognormal distribution is used. For the two other distributions no value is needed at the this position.
The mean value of the particular distribution is defined at the fourth position of the line. If a uniform

distribution is used, no value is needed at this position.
At the fifth position the left side or lower boundary of the distribution is defined. For uniform or triangular distributions this boundary is recommended.

*variables	Definition of the probabilistic input variables.
“1” ;“2” ;“3” ;“4” ;“5” ;“6” ;“7” ;“8”	Each line describes one distribution, the single values are: 1: Name of the input variable 2: Type of distribution (1 - uniform, 2- normal, 3 - lognormal, 4 - triangular) 3: Standard deviation, only needed for normal and lognormal distributions 4: Mean value, not needed for uniform distributions 5: Lower boundary of the distribution as absolute value. Necessary for uniform and triangular distributions, optional for normal and lognormal distributions. 6: Upper boundary of the distribution as absolute value. Necessary for uniform and triangular distributions, optional for normal and lognormal distributions. 7: Delta value of the experimental design (only needed for RSM) 8: Center point of the experimental design (only needed for RSM)

Table 4.8. Probabilistic Variables Definition

For normal and lognormal distributions it is not recommended, but a value x may still be entered to define the lower (left side) boundary of the distribution.

At the sixth position the right side or upper boundary of the distribution is defined. In the same way as for the lower boundary, this value x is recommended for uniform and triangular distributions and is optional for normal and lognormal distributions, where it limits the distribution at the right side.
The value of the delta-value (see subsection 2.6.1) and of the central point of the experimental design are defined at the seventh and the eighth position of the particular line, respectively. These values are only needed for a RSM analysis if the data sets of the deterministic models at the sampling points have to be created by ProSi.

At the paragraph “*dep_variables” the dependent variables in the files of the deterministic model are defined. Thereby every line describes one dependent variable.
The name of the dependent variable stands at the first position of the line. It is arbitrary, but must not include any semicolons and has to be unique.
At the second position the name of the file in which the dependent parameter can be found has to be written. This filename has to be specified also at the paragraph *files.
The third position specifies the number of characters which are between the beginning of the original file (or the beginning of the anchor string if one is defined) and the first character of the value which has to be replaced. This value is used by ProSi to find the variable at the file of the deterministic model.
The fourth parameter is the anchor string in the file. If an anchor string is given ProSi will interpret the position given under 3 as a relative position from the first character of the anchor string. The anchor string can be any string in the file but it has to be situated above the position of the dependent variable in the file, has to include only one line and may contain spaces.
At the fifth position, the original value of the dependent variable out of the file of the deterministic

*dep_variables “1” ;“2” ;“3” ;“4” ;“5” ;“6”	Describes the positions of the dependent input values in the original files of the deterministic model and how the dependent values are to be calculated. Each line describes one dependent input value. The single positions are: 1: Name of the dependent input parameter 2: Name of the file in which the parameter can be found 3: Position of the dependent parameter in the original file as number of bytes from beginning of the file or, if defined, from the beginning of the anchor string 4: Anchor string in the file (not imperative), A unique string situated above the dependent parameter in the file 5: Value of the parameter in the original file (exactly as written in the original file) 6: Formula for the calculation of the value for the dependent parameter.

Table 4.9. Dependent Variables Definition

model has to be written exactly as it appears in the file. This is used by ProSi as a check if the right position has been found. If ProSi does not find exactly this value at the given position it will exit with an error message.

The sixth position specifies a formula from which the value of the dependent parameter is calculated. The formula may contain the arithmetic operators “+ - * /”, brackets, numbers and names of probabilistic (independent) variables.

Note: Version 2.0 of the ProSi core is able to interpret exponents in the formulas. These have to be written as x ˆ y. However the use of exponents is not recommended as the graphical user interface is currently not capable of reading them. Therefore a file using exponents can be used by the ProSi core but it will not be possible the reimport this file into the graphical user interface.

At the paragraph “*correlation_matrix” the lower triangular part of the correlation matrix (including the 1’s in the main diagonal) is defined. Only variables which are to be correlated have to be written here. All variables not found in the correlation matrix are assumed to be independent. The first column has to contain the name of the variable described in this line. The following columns have to contain the desired correlations in order of the names. The line directly below the correlations has to contain the keyword “variable names” in the first column followed by all variable names in the same order as they appeared in the first column above.

4.2.2. Paragraphs of the input file needed for the postprocessing

During the postprocessing the directory containing the single subdirectories with the deterministic results is scanned by ProSi. The results of the single shots are collected and written to the ProSi *.out output file.

The realisations of the probabilistic input parameters (for MCS) or the input values at the sampling points of the experimental design (for RSM) are also written to this file. Furthermore a basic statistic analysis of the input and the result values is performed by ProSi and the results of this statistic analysis are written to the *.STATISTIC file. For the postprocessing, the same input file may be used as for the preprocessing. Thereby the postprocessing is controlled by the following paragraphs of the input file:

- *directories

- *scan_positions

The paragraph *directories has been described in the section preprocessing 4.2.1. The path written at the second position of the line following *directories is the path to the directory which will be scanned for the result parameters. This directory has to contain the results of the single deterministic calculations in the way that the results of each deterministic simulation are in a separate subdirectory. The results of the postprocessing will be written to the path which is specified at the third position in the line following *directories.

The paragraph *scan_positions defines the result parameters which have to be scanned by ProSi out of the result files of the deterministic simulations. Each line following the keyword *scan_positions defines one result parameter. The first position of each line contains the name of the according result parameter. This name is arbitrary, but must not contain any semicolons. The second position specifies the name of the result file (only the name of the file without a path) from which ProSi has to read this parameter. The third position is the location of the result value in the file specified at position two. This location has to be given as the number of characters from the first character of the reference string (specified under position four) to the first character of the result value which has to be read. The anchor string specified at position four of each line has to be a string of arbitrary length, which is unique in the result file and therefore can be found by ProSi. The string has to be written between semicolons, without additional spaces at the beginning or the end of the string. ProSi will consider all characters between the semicolons as belonging to the anchor string.

If a value for failure criterion is provided, ProSi will calculate the probability for values below this criteria.

If a value for quantile is provided (a percentage number) ProSi will calculate the absolute value which represents the percentage probability given.

*scan_positions “1”; “2”; “3”; “4”	Definition of the result parameters of the simulation. Only needed for the postprocessing. Each line defines one result parameter. The data needed is: 1: Name of the result value 2: File from which the result value should be read 3: Position of the result value within the file as number of characters from the anchor string specified under 4 4: Anchor string, A string which is unique in the result file. If none is defined, the beginning of the result result file will used as reference point.

Table 4.10. Definition of the Result Position

4.3. Using ProSi in command line mode

The usual way to start ProSi is by using the execution user interface ProSi Exe. However, it is still possible to start ProSi without a graphical user interface directly from the command line. If started from the command line ProSi can only be used at the local computer, no distributed computing is possible.

To start ProSi from the command line, the syntax is: prosi “input_file_name” “arguments”.

4.3.1. ProSi command line arguments

The first argument following the program call “prosi” is always the name of the input file. The only exception for this is the call “prosi –help” which will display an overview of the ProSi command line

arguments and the structure of the input file.
After the name of the input file usually one argument follows, which controls the run mode of ProSi. Available arguments depend whether the input file specified is a project input file or a subproject input file. Project input files are the ones created directly by the user or by ProSi Pre. Subproject input files are always created by ProSi (ProSi Exe in graphical mode, or the ProSi main program in non graphical mode). Subproject input files do always start with a string of eight numbers followed by an underscore and the name of the project input file they belong to.
Possible command line arguments for calling ProSi in connection with a project input file are:

- “no argument”
- -silent
- -doe
- -mcs

A call of ProSi with only the input file and no further arguments starts the first part of the preprocessing. ProSi will create a subproject including the directory structure needed and the subproject input file and generate the probabilistic variables which are then stored in the restart file *.INPUT. The method used for the generation of the variables depends on the definition under “*method” in the input file in this case. If either “-mcs” or “-rsm” is used ProSi also generates the probabilistic variables but ignores the “*method” switch in the input file. In all cases no generation of deterministic samples is done.
Using the argument “-silent” ProSi will run without giving any output of what ProSi does into the console, except for error-messages and very important warnings. All messages will still be written into the *.LOG file.
Possible command line arguments in connection with a subproject input file are:

- -write=x
- -scan=shot_directory
- -sort
- -rsm
- -rsm+mcs
- -silent

The first argument “-write=x” will cause ProSi to generate the shot number x, if only “-write” is provided ProSi generates all shots. It thus prepares the given shot for deterministic calculation.
The second possible argument “-scan=shot_directory” means, the ProSi scans the results of the shot in the directory specified. If only “-scan” is provided the results for all available shots are scanned. Ultimately ProSi will store the results of the deterministic calculation of the selected shot into the *.out file.
The next argument called “-sort” is used for postprocessing. ProSi only sorts an existing output file by shot numbers and performs a statistical analysis.
Continuing in the list of arguments the next possibility is “-rsm”. ProSi uses the values it finds in the *.out file for the project as sampling points and calculates a response surface. In case that more than one result parameters are defined in the input file, the RS will be calculated for the first one. If using a project which was an MCS an additional method definition (keyword “*rsm”) for the RSM has to be added to the input file.

The argument “-rsm+mcs” is used to make ProSi use the values it finds in the *.out file to calculate a response surface. Afterwards an MCS is performed using the response surface generated to calculate the shot results. Both method definitions (*rms and *mcs) have to be available in the input file. Again, using the argument “-silent” ProSi will run without giving any output of what ProSi does into the console, except for error-messages. All other messages are still written into the *.LOG file.

A. Appendix

A.1. Mathematical Appendix

mean value:

$$\mu = \frac{1}{n_{sim}} \sum_{i=1}^{n_{sim}} b_i^2$$

sigma:

$$\sigma = \frac{1}{n_{sim} - 1} \sum_{i=1}^{n_{sim}} (b_i - \mu)^2$$

coefficient of variation:

$$CV = \frac{\sigma}{\mu}$$

skewness

$$S = \frac{\sum_{i=1}^{n_{sim}} (b_i - \mu)^3}{n_{sim} * \sigma^3}$$

kurtosis

$$K = \frac{\sum_{i=1}^{n_{sim}} (b_i - \mu)^4}{n_{sim} * \sigma^4} - 3$$

A.2. Examples Appendix

You will find some examples for each method ProSi can operate in the directory:

%PROSI%/input_data

respectively

%PROSI%\input_data

For each method one example will be shown. The basics for the examples are descibed in this manual in chapter 3. Graphical user interface.

A.2.1. Example - MCS

Filename: calc_traeger_mcs_en.inp

Method:	Monte Carlo Simulation
Shots:	100
Sampling:	Descriptive Sampling (DS)

Directories:

deterministic model:	.	the same where the configuration file is located
generated Deterministic Data Sets:	data/.	subdirectory ”data” from the position of the configuration file
ProSi Log and Output Files:	output/.	subdirectory ”output” from the position of the configuration file

Files used for the Probabilistic Model:

- calc_traeger
- traeger.INPUT

Files used for the result values:

- traeger.OUTPUT

probabilistic Variables:

Variables Name	distribution	sigma	mean	lower bound	upper bound
force	normal	300	2500	0	5000
emodul	normal	10000	210000	100000	320000
height	uniform			90	110
width	uniform			40	60
length	uniform			4900	5100

dependent Variables:

dependent Variables	correspond to
dep_force	force
dep_emodul	emodul
dep_height	height
dep_width	width
dep_length	length

Result Values
deflexion (with failure criterion F<12)

Result-files:
/data/. there is a project subdirectory, which contains

- subdirectory for every processed within
- the modified files: calc_traeger.INPUT calc_traeger.OUTPUT
- a script/batch-file
- *.out_ShotNr-file
- the program file calc_traeger

/output/. there is a project subdirectory, which contains

- the configurations file for the project (.inp)
- the experimental design (.INPUT)
- the results from experimental design (.out)
- the logfile (LOG)
- the statistic file (.STATISTIC)
- the execution logfile (execution.LOG)

A.2.2. Example - RSM

Filename: calc_traeger_rsm_en.inp

Method:	Response Surface Method
Experimental_Design:	BoxBehnken
Power:	2
Improvement:	yes

Directories:

deterministic model:	.	the same where the configuration file is located
generated Deterministic Data Sets:	data/.	subdirectory ”data” from the position of the configuration file
ProSi Log and Output Files:	output/.	subdirectory ”output” from the position of the configuration file

Files used for the Probabilistic Model:

- calc_traeger
- traeger.INPUT

Files used for the result values:

- traeger.OUTPUT

probabilistic Variables:

VariablesName	description	Interval	Centerpoint
force	the force on the beam	300	2500
emodul	the modules of elasticity of the beam	10000	210000
height	the height of the beam	5	100
width	the width of the beam	5	50
length	the force position on the beam	100	5000

dependent Variables:

dependent Variables	correspond to
dep_force	force
dep_emodul	emodul
dep_height	height
dep_width	width
dep_length	length

Result Values

deflexion

Result-files:

/data/. there is a project subdirectory, which contains

- subdirectory for every processed within
- the modified files: calc_traeger.INPUT calc_traeger.OUTPUT
- a script/batch-file
- *.out_ShotNr-file

- the program file calc_traeger

/output/. there is a project subdirectory, which contains

- the configurations file for the project (.inp)
- the experimental design (.INPUT)
- the results from experimental design (.out)
- the logfile (LOG)
- the statistic file (.STATISTIC)
- the execution logfile (execution.LOG)
- the results of RSM (.rs)

A.2.3. Example - RSM + MCS

Filename: calc_traeger_rsmmcs_en.inp

Method:	Response Surface Method + Monte Carlo Simulation
Shots:	100
Sampling:	Latin Hypercube Sampling (LHS)
Experimental Design:	Box Behnken
Power:	2
Improvement:	no

Directories:

deterministic model:	.	the same where the configuration file is located
generated Deterministic Data Sets:	data/.	subdirectory ”data” from the position of the configuration file
ProSi Log and Output Files:	output/.	subdirectory ”output” from the position of the configuration file

Files used for the Probabilistic Model:

- calc_traeger
- traeger.INPUT

Files used for the result values:

- traeger.OUTPUT

probabilistic Variables:

VariablesName	distribution	mean	sigma	lower bound	upper bound
force	normal	2500	300	0	5000
emodul	normal	210000	10000	100000	320000
height	uniform			95	105
width	uniform			45	55
length	uniform			4900	5100

VariablesName	distribution	Interval	Centralpoint
force	normal	300	2500
emodul	normal	10000	210000
height	uniform	5	100
width	uniform	5	50
length	uniform	100	5000

dependent Variables:

dependent Variables	correspond to
dep_force	force
dep_emodul	emodul
dep_height	height
dep_width	width
dep_length	length

Result Values

deflexion

Result-files:

/data/. there is a project subdirectory, which contains

- subdirectory for every processed within
- the modified files: calc_traeger.INPUT calc_traeger.OUTPUT
- a script/batch-file
- *.out_ShotNr-file
- the program file calc_traeger

/output/. there is a project subdirectory, which contains

- the configurations file for the project (.inp)
- the experimental design (.INPUT)
- the results from experimental design (.out)
- the logfile (LOG)
- the statistic file (.STATISTIC)
- the execution logfile (execution.LOG)
- the results of RSM (.rs)
- the results of the MCS (_RSM_MCS.out)