TNO Built Environment and Geosciences

Van Mourik Broekmanweg 6
P.O. Box 49
2600 AA Delft
The Netherlands

T +31 15 276 30 00
F +31 15 276 30 11

TNO report

2007-D-R0887/A

Prob2B^{TM}: variables, expressions and Excel® Installation and Getting Started

Date	23 August 2007
Author(s)	Dr. Ir. W.M.G. Courage Ir. H.M.G.M. Steenbergen
Number of pages	108
Number of appendices	4
Sponsor	
Project name	Prob2B
Project number	034.77188

Contents

Appendices

1 Introduction and installation requirements

1.1 Introduction

This document is the user and installation manual for Prob2B version 7.1 with implementations for addressing limit state functions defined by internal variables, user defined expressions and Excel® models.

Prob2B^{TM} is a probabilistic toolbox under development, capable of being coupled to and driving other software. The development has reached a stage in which it performs reliability calculations by steering stochastically obtained input to the deterministic external programs, and processing their (deterministic) output using probabilistic techniques. This loop is repeated as often as is required to obtain the end result of the reliability calculation. Prob2B^{TM} is a result of an initiative of TNO Built Environment and Geosciences, in unlocking their broad knowledge and experience with respect to reliability calculations and risk management towards a manifold of applications in civil engineering.

Excel® is part of Microsoft® Office. Copyright © 1985-2007 Microsoft Corporation.

1.2 Run-time environment

Prob2B requires the following minimum system configuration:

CPU:	Pentium® II (Pentium 4 is recommended)
Memory:	128 Mb of RAM (512 Mb is recommended)

Currently, Prob2B is available under Windows 2000 or Windows XP operating systems.

1.3 Distribution of Prob2B^{TM}

Prob2B^{TM} is either distributed from a shared folder structure or by a CD-ROM.
The shared directory of the CD-ROM has the following directory structure:

- Prob2B binary code and examples for Prob2B

2 Installing Prob2B^{TM} stand-alone

To install the Prob2B stand-alone version do the following.

Copy the directory `Prob2B` to your local disk in the 'Program Files'-directory:
C:\Program Files\Prob2B

2.1 Setting permissions to run probTB.dll

Prob2B uses the file `probTB.dll`, which is a fortran library. `probTB.dll` requires permission to be run under your PC. In order to do so, run the program **`VFRUN66BI.exe`**. You need system administrator privileges to install this program. If this program reports a conflict with items already installed on your computer DO NOT overwrite your own files (click the NO button).

2.2 File management

A predefined file management is mandatory, especially when addressing external models like for instance Mathcad, Dll's, MatLab/Femlab or stand alone programs. When restricted to internal variables or expressions, the file management will appear unnecessary strict. In that case, only input files and output files will have to be defined, as depicted schematically in Figure 1.

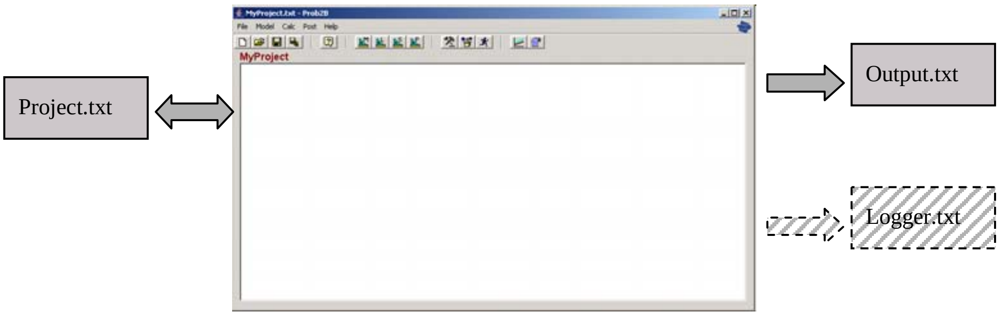


Figure 1 File management.

Project.txt:

this text file is Prob2B's project file. Once saved by Prob2B, the project file contains all project definitions set during a Prob2B session, for instance: models, parameters, distribution functions, limit states and calculation methods. Existing project files can be used to restart a saved session and continue or do recalculations. The user is free to use a user defined or project specific name (with or without any extension). Apart from the file's name, also the file's directory is used in that a log file of the session, containing informative or error messages, is written in that directory. See 'logger.txt'.

Logger.txt:

This file is automatically generated and contains a log of the session, containing informative or error messages. It is written in the same directory as the user defined project-file. The file is always named 'logger.txt'.

Output.txt

The user can save his calculation results, e.g. sample values, in a tab-delimited text file. The user is prompted for a file-name and directory.

3 Getting started with Prob2B^{TM}

After you have installed Prob2B as described in section 2, you can start Prob2B and create a new project or load an existing project.

3.1 Starting Prob2B

Run Prob2B by double clicking `Prob2B.bat` in the directory on the local disk. A dos window appears, followed by the Prob2B main screen (Figure 2)

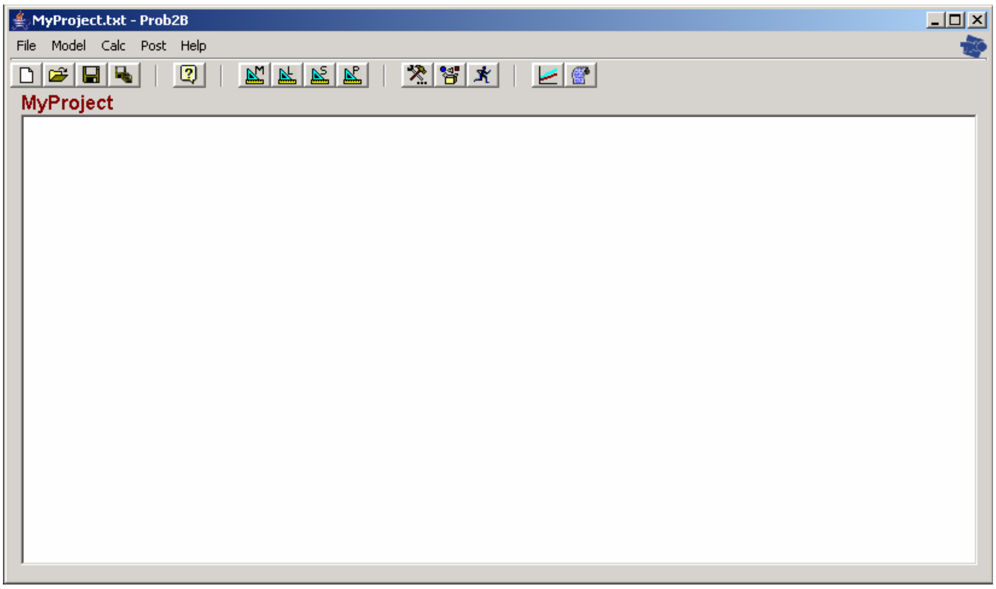


Figure 2 Main screen of the Prob2B application

3.2 General layout of Prob2B window

Looking at the top left of the main Prob2B window, we see a menu- and button layout as depicted in Figure 3. They are ordered into the following groups:

File menu / file buttons

> The file menu contains items for opening a new or an existing object. Also save and save-as actions are contained. All of these are also available as buttons below the menu bar.

Model menu / model buttons

> Through the model menu, variables, expressions and external models can be defined and loaded. Resulting variables can then be assembled into a user defined limit state function. Stochastic properties and correlations can be (re-)set in a

separate pane. Finally, also settings for parametric calculations can be defined. All of this can be done via the menu or directly via the corresponding buttons.

Calculation menu / calculation buttons

In the calculation menu or by the calculation buttons, the user can select the calculation method, select the results to be saved and give the command to run the calculations.

Post menu / post buttons

The post menu and buttons give access to graphic post processing tool and also give the possibility to save the calculation results in a results file.

Help menu / help buttons

The help options are nor operational (yet). In new versions of Prob2B, they are intended to give online help.

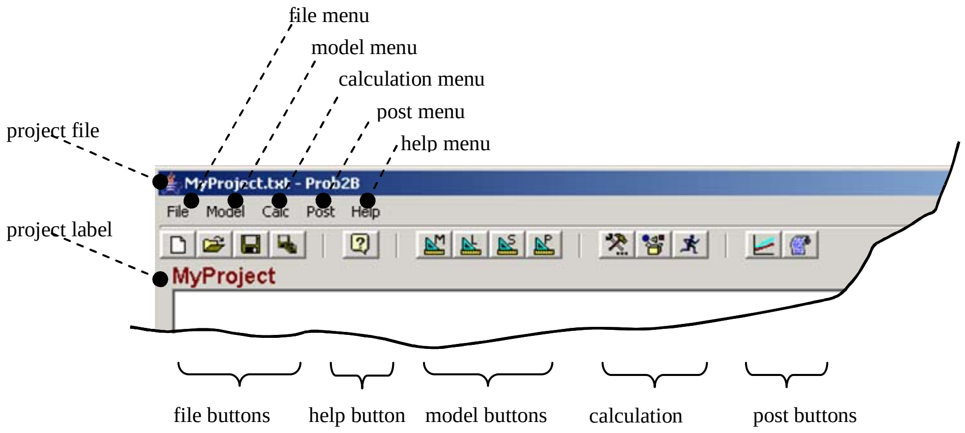


Figure 3 Global layout of Prob2B's menus and buttons

The general workflow in Prob2B would be to go from left to right through the menu's or buttons. This is shown graphically in Figure 4.

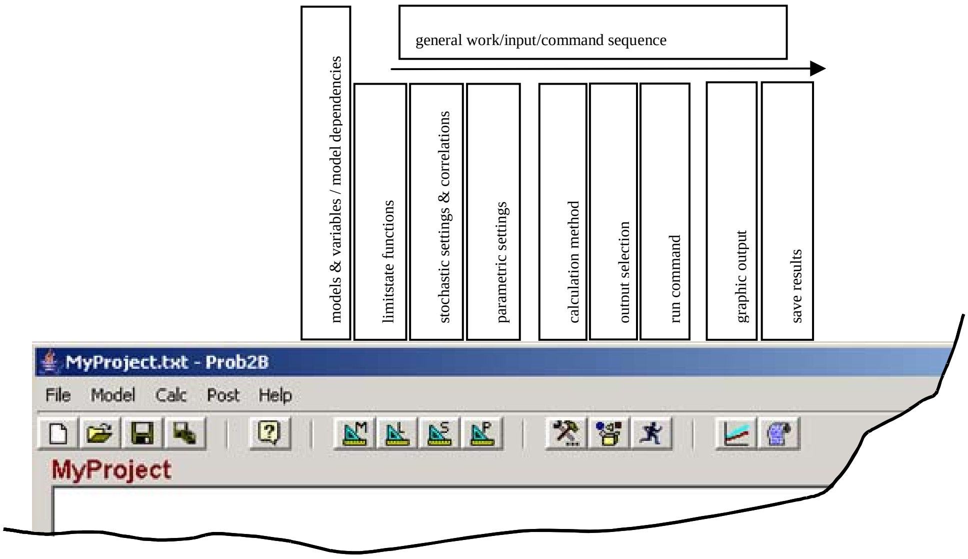


Figure 4 Buttons and general work flow

3.3 Examples presented in this manual

The following chapters will take you through a number of examples of a Prob2B calculation using internal variables and expressions and external models. The corresponding (input-) windows and buttons for Prob2B will thus be described step by step. The examples in this manual successively deal with

Variables (Chapter 4),
Expressions (Chapter 5),
External models, e.g. Excel® (Chapter 6),
Dependencies in models and variables (Chapter 7)
Stochastic parameters as functions of variables (Chapter 8)

Most general info about how to use Prob2B options and menus will be found in Chapter 4. After that, the use of expressions, external models and dependencies are more specifically dealt with in the remaining chapters.

4 Using Variables

4.1 Background on first example

In the following paragraphs the following limit state function will be modelled in Prob2B:

$$Z = \exp(J_c) - \frac{3(1-\nu^2)}{32Ed^3} R^4 (\Delta p_2 + \rho dg)^2$$

with variables as listed in Table 1.

Table 1 Input of variables for first example

Variable	**Symbol**	**Unit**	**Mean**	**St. dev**	**Distribution**
Poisson's ratio	ν	-	0,3	-	Deterministic
Radius of defect	R	m	0.3	-	Deterministic
gravity	g	m/s2	9.81	-	Deterministic
Density	ρ	kg/m^3	850	75	Normal
Change in pressure	Δp_2	N/m2	$4{,}0 \cdot 10^3$	$1{,}0 \cdot 10^3$	Normal
Variable	**Symbol**	**Unit**	**m**	**sigma**	**Distribution**
Thickness of layer	d	m	0,046	0,0052	Student (23)
Crack propagate energy	J_c	J/m^2	0,0092	0.88	Student (10)
Young's modulus	E	Pa	$2{,}0 \cdot 10^9$	$0{,}18 \cdot 10^9$	Student (8)

The variables ρ and E are assumed to have a correlation of 0.5.

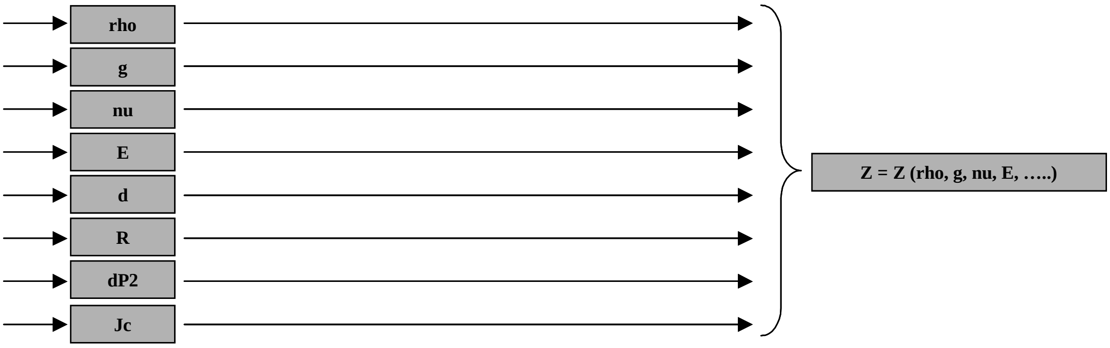


Figure 5 Variables (as models) and limitstate function.

In Prob2B, distinct variables are handled as separate models. This is schematically depicted in Figure 5. Variables are indicated by a box with on arrow for input and one arrow for output. The variable-model, does however not perform a calculation, i.e. output is identical to input. (output) variables will become available in the menus for defining a the limit state functions.

Defining the above example in Prob2B would imply the following actions:

- Create a project.
- Define the variables.
- Define the limit state function.
- (Re-)set the stochastic properties and set the correlations.
- Set the calculation method.
- Select the output desired.
- Run the calculations.
- Look at results.
- Save project and results.

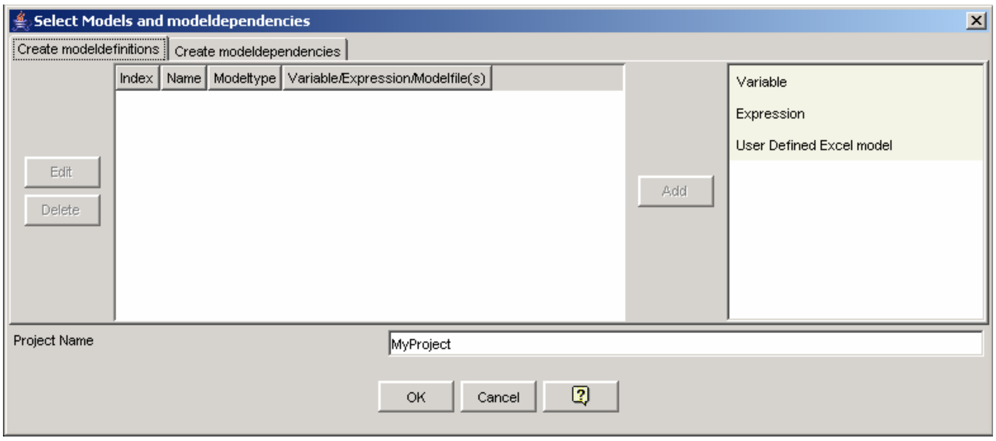
These steps, and others, are discussed in the following paragraphs.

4.2 Creating a new project

Starting up Prob2B will automatically set Prob2B to be in a mode for a new project and the user can proceed with the input.
When already within a Prob2B session, the user can create a new project by selecting the "new project " option from the file menu, or click on the new project icon. The user will be prompted if the current settings have to be saved first.

4.3 Defining models and variables

Next step would be to define the models that have to be addressed by Prob2B.
This process can be activated by clicking the Model button or by going to the menu in the menu bar.

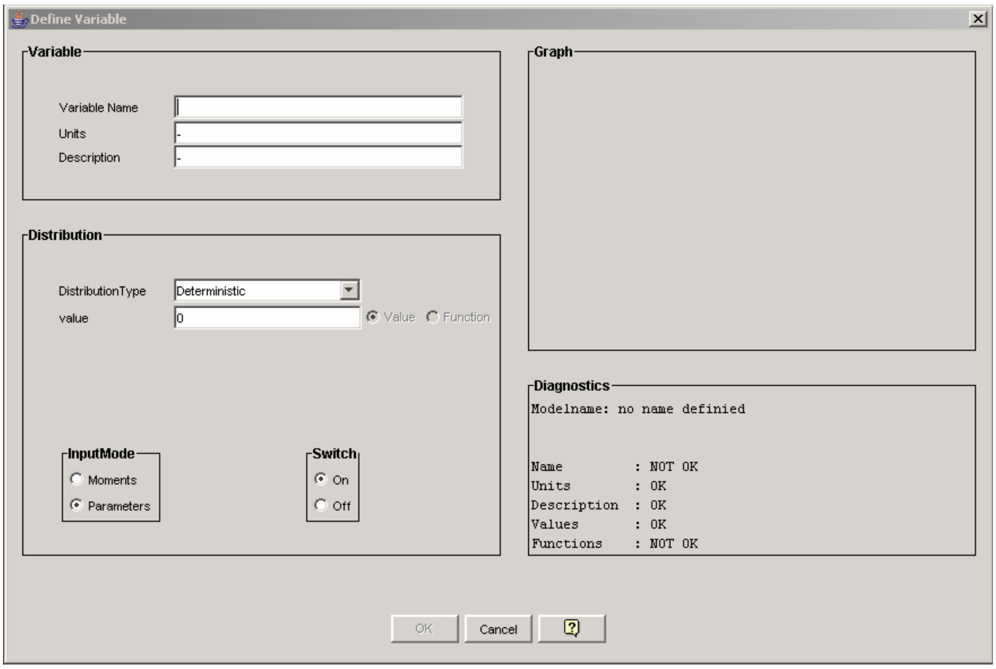


Project Name: MyProject

Figure 6 Model selection window

The model selection window appears, which consists of two tab-panes for defining the models to be analysed and for defining possible dependencies between models. It also contains a text-field in which one can define a project's label.

In defining the new project, one has to add the model (or multiple models) with the input pane shown in Figure 6.
First the user has to select the type of model one wants to add. This can be done by selecting one of the types in from the list on the right hand side. Next clicking on the add button will prompt the subsequent next input menu for the corresponding model type. (The same effect is achieved by double clicking on the desired model type).
In this example we select the 'Variable-type'. This results in the following input menu:



Variable Name:
Units: -
Description: -

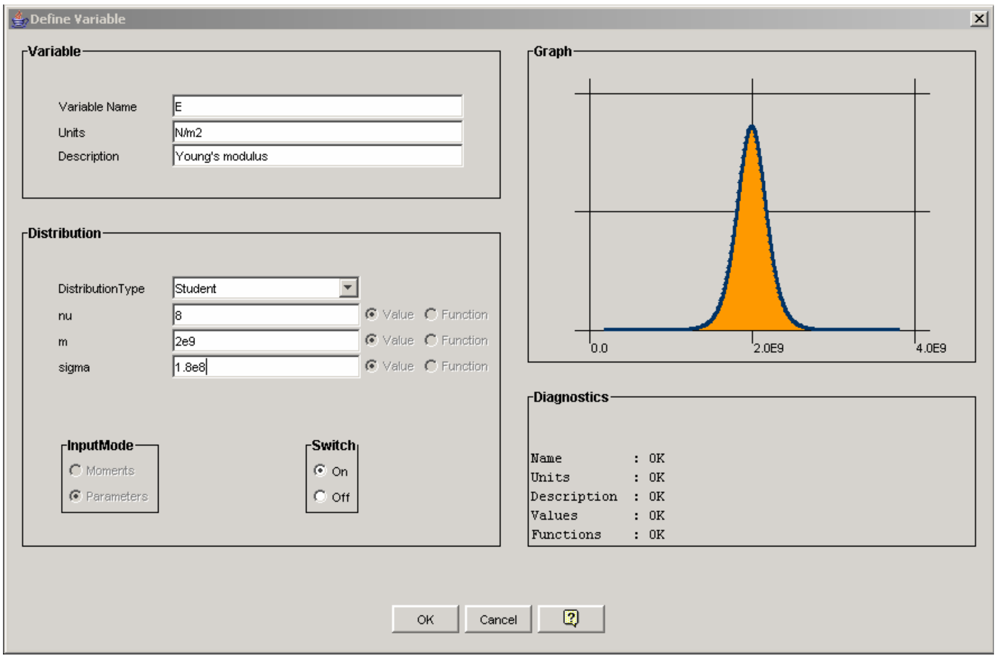
DistributionType: Deterministic
value: 0 ● Value ○ Function

InputMode: ○ Moments ● Parameters

Switch: ● On ○ Off

```
Modelname: no name definied


Name        : NOT OK
Units       : OK
Description : OK
Values      : OK
Functions   : NOT OK
```

Figure 7 Input-window for a new Variable



Variable Name: E
Units: N/m2
Description: Young's modulus

DistributionType: Student
nu: 8 ● Value ○ Function
m: 2e9 ● Value ○ Function
sigma: 1.8e8 ● Value ○ Function

InputMode: ○ Moments ● Parameters

Switch: ● On ○ Off

```
Name        : OK
Units       : OK
Description : OK
Values      : OK
Functions   : OK
```

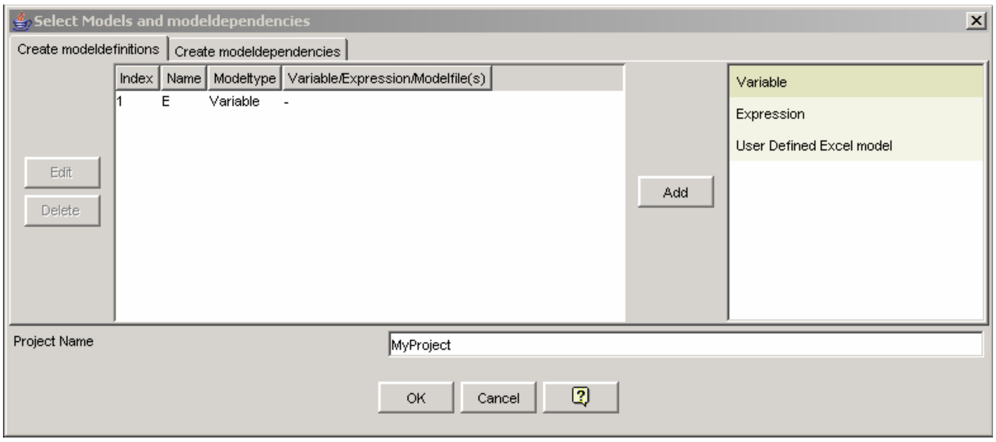
Figure 8 A new variable being defined

The input pane for variables (Figure 7 and Figure 8) is divided into four parts:

- a part for defining the variable’s name, description and units,
- a part for defining its stochastic properties,
- a part for graphic display of the distribution function and
- a diagnostic part, in which the user is given a status report of his input being valid or not.

The variable’s name has to be unique and may not contain special characters or start with a digit, in order to be accepted by Prob2B. The input for the description field and units has to contain at least 1 character to be valid.
In the second part the user can select a distribution type from the selection box and fill in the appropriate values. When allowed by the distribution type, one can choose between parameters or moments in defining the distribution. To the right, there is a check box for switching the stochastic properties on or off. When switched off, Prob2B will not alter the distribution type or values, but in calculations the variable will be treated as deterministic with a value corresponding to its mean value.

Figure 8 shows a valid input screen for a Young’s modulus with a Student distribution
Clicking ‘OK’ returns to the previous window for model selection, giving a list of models defined so far, Figure 9

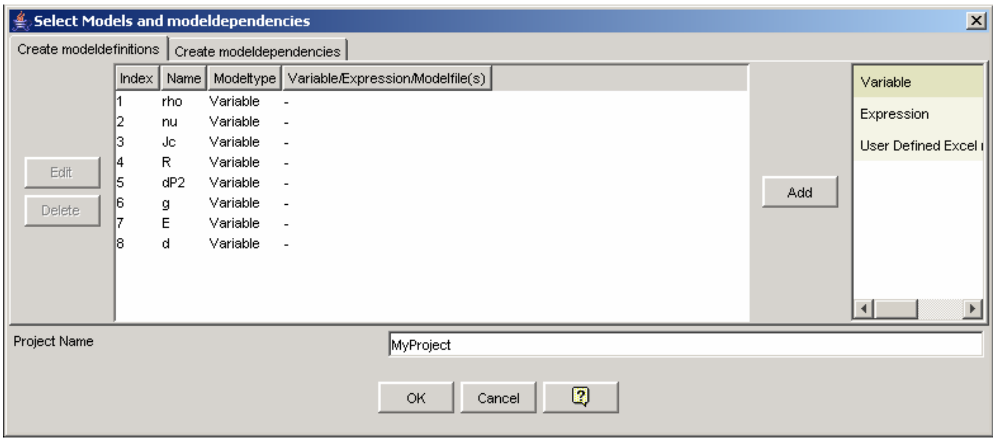


Index	Name	Modeltype	Variable/Expression/Modelfile(s)
1	E	Variable	-

Project Name: MyProject

Figure 9 First variable defined

We continue loading the other variables from Table 1. This eventually leads to the defined models as shown in Figure 10.



Index	Name	Modeltype	Variable/Expression/Modelfile(s)
1	rho	Variable	-
2	nu	Variable	-
3	Jc	Variable	-
4	R	Variable	-
5	dP2	Variable	-
6	g	Variable	-
7	E	Variable	-
8	d	Variable	-

Project Name MyProject

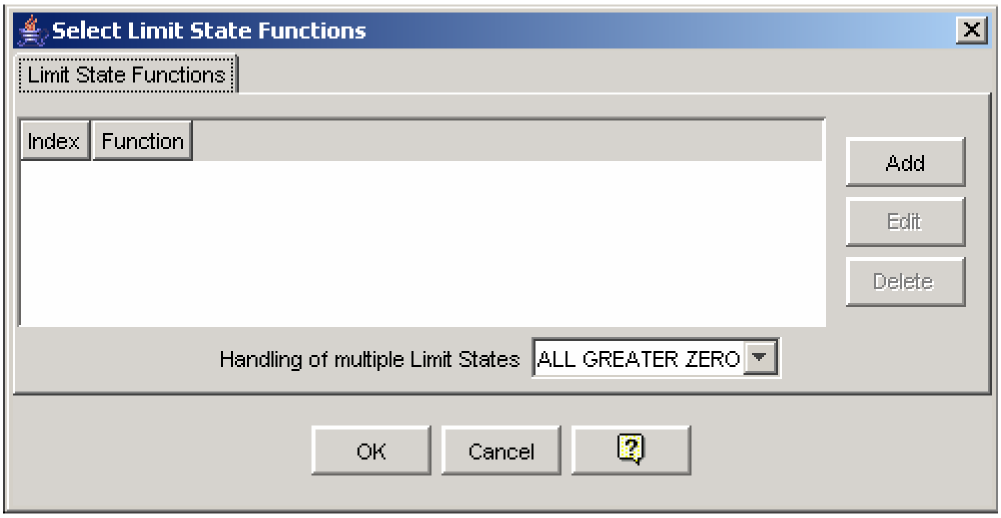
Figure 10 Defined variables.

Clicking ‘OK’ returns to the main window of Prob2B.
Going back through the Model selection menu or clicking the Model button retrieves the model selection windows for adding or editing.

4.4 Setting the limit state function

After specifying the model(s) to use, we will set the limit state function definition and set the relevant parameters.

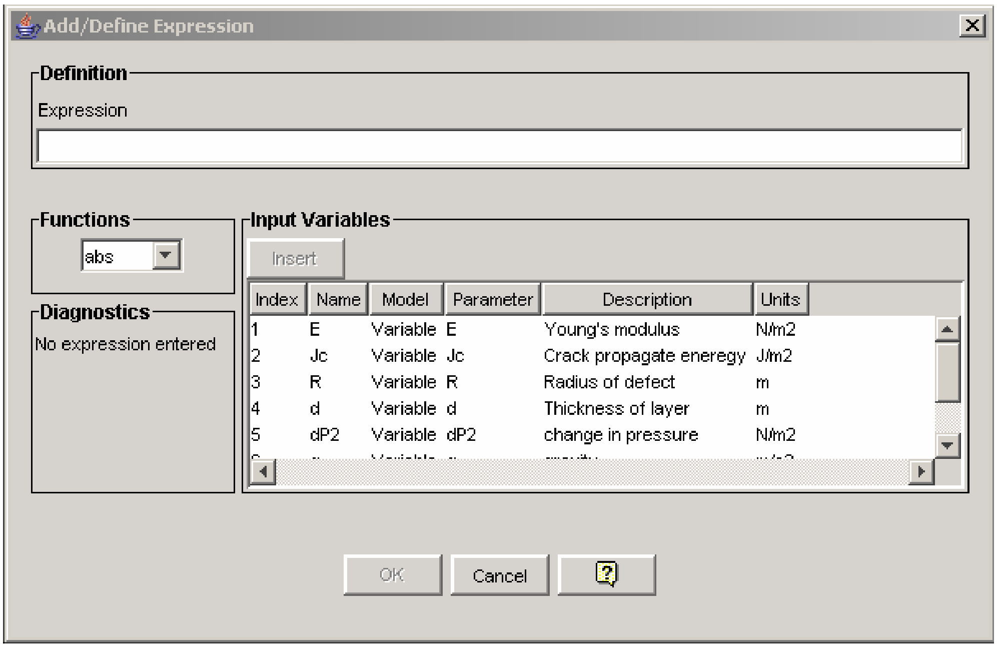
In the model menu one can select “select limit states”. Alternatively one can click directly on the Limit button. Again a tabbed pane appears, shown in Figure 11.



Handling of multiple Limit States: ALL GREATER ZERO

Figure 11 Selecting Limit State Parameters and functions

Clicking ‘Add’ on the Limit State function pane gives the possibility to define a new limit state function

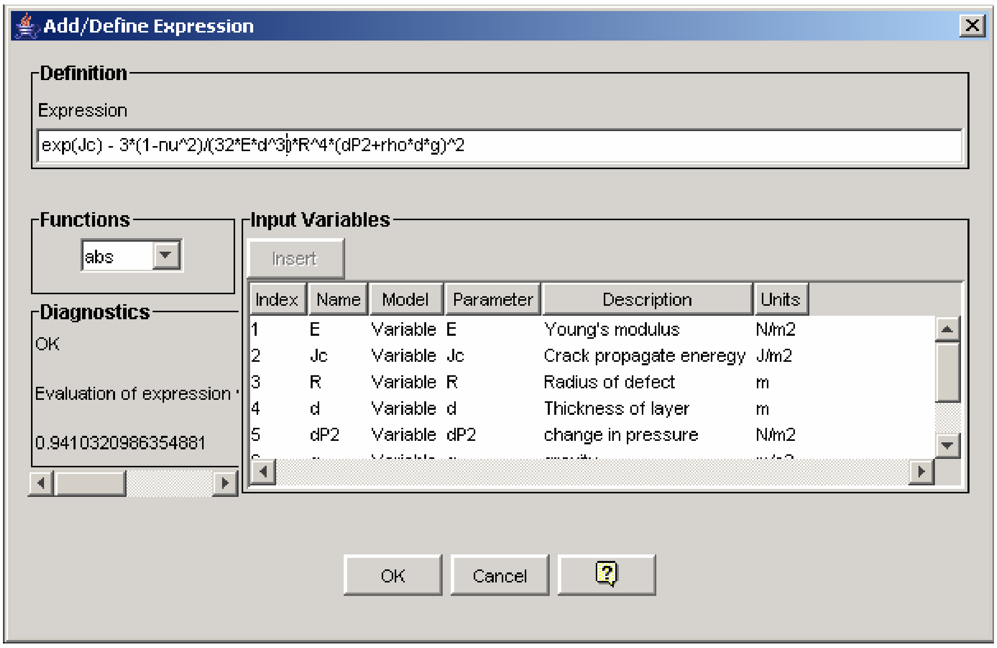


No expression entered

Index	Name	Model	Parameter	Description	Units
1	E	Variable	E	Young's modulus	N/m2
2	Jc	Variable	Jc	Crack propagate eneregy	J/m2
3	R	Variable	R	Radius of defect	m
4	d	Variable	d	Thickness of layer	m
5	dP2	Variable	dP2	change in pressure	N/m2

Figure 12 Defining a limit state

Limit State Functions will be defined by using Limit State parameters. These Limit State Parameters can be seen as candidate parameters for a Limit State function and form a subset of model output results. Prob2B will prompt with a candidate list (named ‘Input Variables’) in accordance with the available models.
When a parameter from the list is selected, by a mouse click, the parameter can be transferred into the expression by pressing the insert button.
Thus, an expression for a limit state function can be created. The expression can be edited and one may select some functions to be used in the expression.
In Figure 13, the limit state function from paragraph 4.1 is added as expression.



Expression: exp(Jc) - 3*(1-nu^2)/(32*E*d^3)*R^4*(dP2+rho*d*g)^2

Functions: abs

Diagnostics:
OK

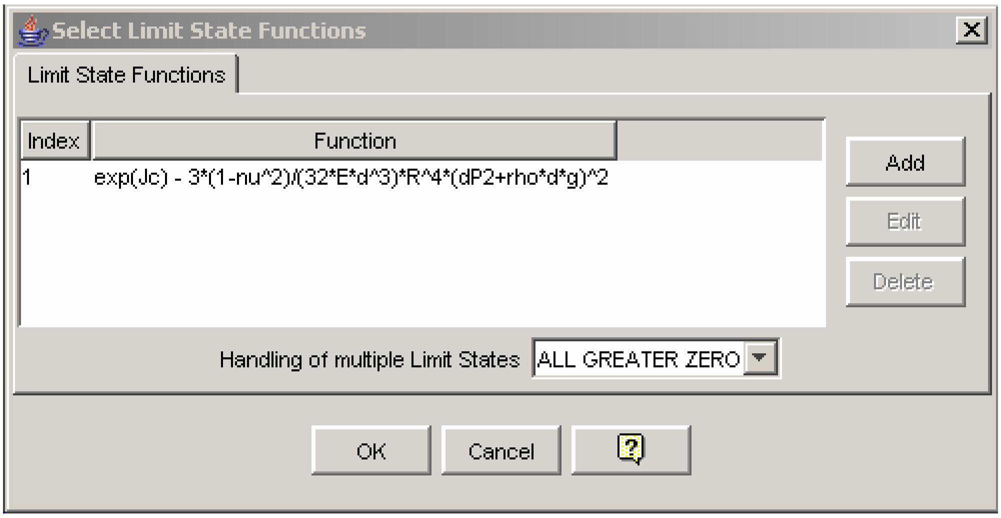
Evaluation of expression

0.9410320986354881

Index	Name	Model	Parameter	Description	Units
1	E	Variable	E	Young's modulus	N/m2
2	Jc	Variable	Jc	Crack propagate eneregy	J/m2
3	R	Variable	R	Radius of defect	m
4	d	Variable	d	Thickness of layer	m
5	dP2	Variable	dP2	change in pressure	N/m2

Figure 13 Adding the limit state function

Clicking OK returns to the tabbed pane of the limit state window.



Index	Function
1	exp(Jc) - 3*(1-nu^2)/(32*E*d^3)*R^4*(dP2+rho*d*g)^2

Handling of multiple Limit States: ALL GREATER ZERO

Figure 14 Limit State window with limit state function defined

Multiple Limit States are allowed to be evaluated simultaneously. The handling of these Limit States will be such that the ‘failure’ is reached when one or more of the Limit State functions become less than zero. Hence, care has to be taken when multiple limit states are used in combination with FORM or SORM methods.

Clicking ‘OK’ returns to the main window of Prob2B.
Going back through the Limit selection menu or directly clicking on the Limit button retrieves the Limit selection windows for adding or editing.

4.5 Setting stochastic properties to variables

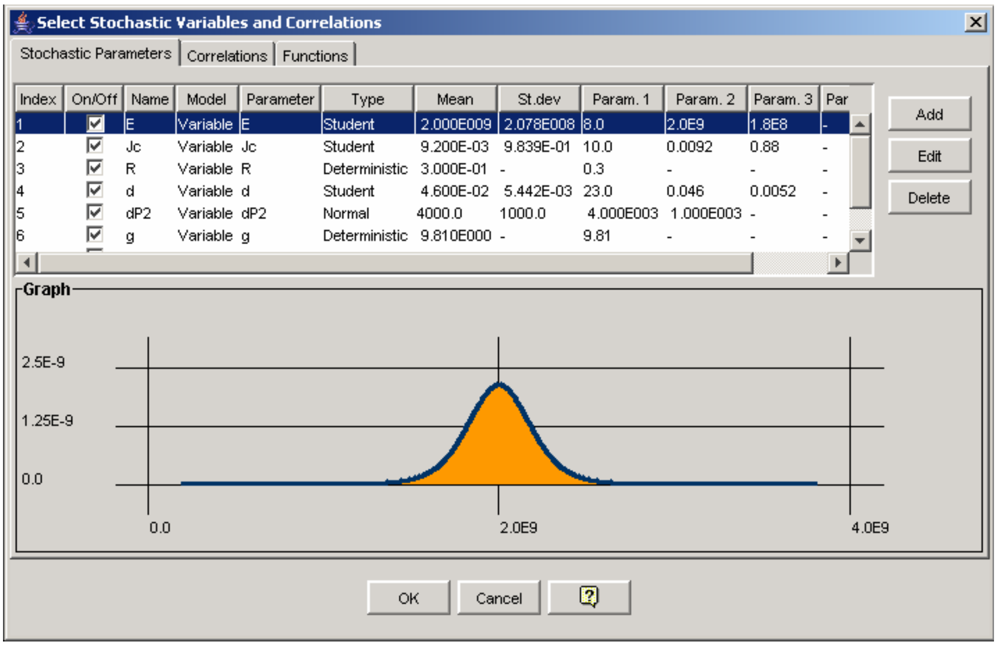
Next we will go through the menus for defining or editing the stochastic properties of parameters.

A list of possible distribution functions is presented in Table 2. Besides selecting a function, one can also read stochastic properties from a file. Formats for such files are presented in Appendix C. A more mathematical description of the distribution functions can be found in Appendix B.

Table 2 Available Distribution functions in Prob2B

Deterministic
Uniform
Triangle
Normal
Lognormal
Exponentional
Gamma
Beta
Gumbel (max. type I)
Frechet (max. type II)
Rayleigh (max.type I)
Student
Weibull (min. type III)
x-u table
x-q table
x-p table
Discrete realisations

In the model menu one can select “Define Stochasts” or one can click directly on the Stochast button. Again, a tabbed pane appears as shown in Figure 15.

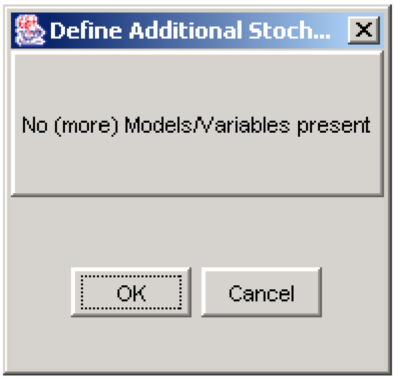


Index	On/Off	Name	Model	Parameter	Type	Mean	St.dev	Param. 1	Param. 2	Param. 3	Par
1	☑	E	Variable	E	Student	2.000E009	2.078E008	8.0	2.0E9	1.8E8	-
2	☑	Jc	Variable	Jc	Student	9.200E-03	9.839E-01	10.0	0.0092	0.88	-
3	☑	R	Variable	R	Deterministic	3.000E-01	-	0.3	-	-	-
4	☑	d	Variable	d	Student	4.600E-02	5.442E-03	23.0	0.046	0.0052	-
5	☑	dP2	Variable	dP2	Normal	4000.0	1000.0	4.000E003	1.000E003	-	-
6	☑	g	Variable	g	Deterministic	9.810E000	-	9.81	-	-	-

Figure 15 Defining stochastic properties

In this example we only use internal variables. For these parameters, the stochastic properties were already prompted for in the ‘Model’-menus.
As a result, we immediately see a list with stochastic parameters when going into stochastic menu.
(if no variables were defined in the model-menu, but expressions or external models instead, then the list would be empty. Stochastic properties of parameters from expressions or external models would then have to be explicitly defined in this menu).

Adding stochastic properties to a parameter would be achieved by pressing the Add button from the “stochastic parameters” tab pane. Doing so in this example leads to the message that there is nothing more to add, as all variables are already defined, Figure 16.

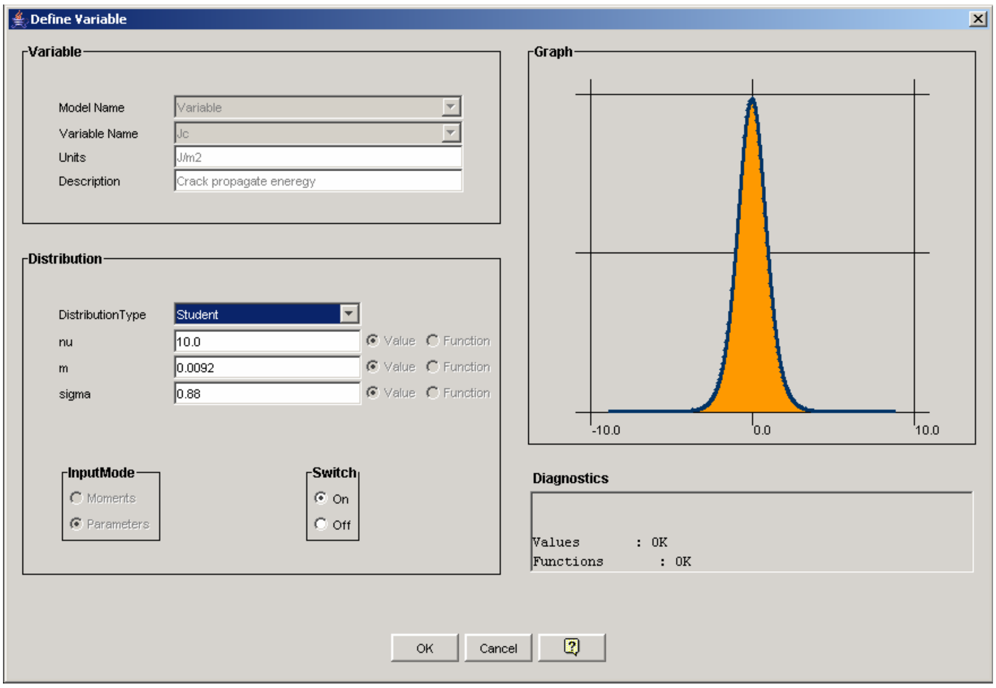


No (more) Models/Variables present

Figure 16 no more stochastic properties to be defined

Editing (or deleting) a stochastic parameter from the table is achieved by selecting the corresponding row and clicking the ‘Edit’ (or ‘Delete’) button. The edit option is also

activated by double clicking a row from the table. Single clicking a row will activate the corresponding graphical display of the selected stochastic parameter.
When selecting the variable Jc and clicking the edit button, the window in Figure 17 will appear.



Model Name: Variable
Variable Name: Jc
Units: J/m2
Description: Crack propagate eneregy

DistributionType: Student
nu: 10.0 ● Value ○ Function
m: 0.0092 ● Value ○ Function
sigma: 0.88 ● Value ○ Function

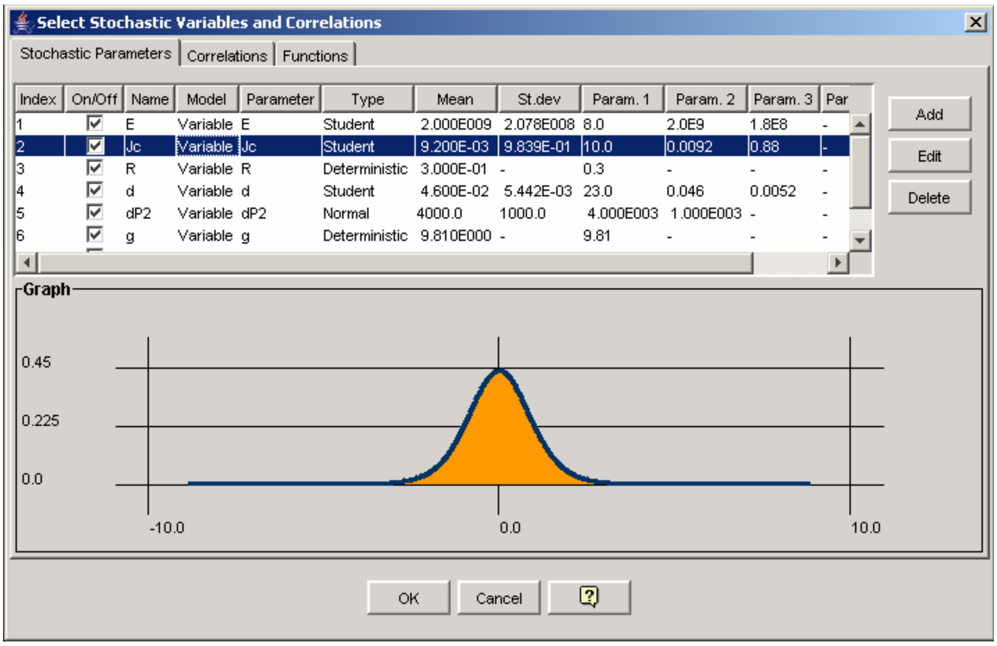
InputMode: ○ Moments ● Parameters
Switch: ● On ○ Off

```
Values      : OK
Functions      : OK
```

Figure 17 Editing Stochastic Properties for parameters

We see that we cannot alter the variables name. Only its stochastic properties and whether it is active or not. When it is set ‘not-active’ (Off), Prob2B will make calculations with the parameter’s value **fixed** to its mean value. This option is only available for parameters whose stochastic data is not defined in a file.
(When one wants the calculations to be performed with the parameter value fixed to its initial value, e.g. the default value of a parameter from an external model, one has to delete the stochastic parameter from the table. Another possibility would be to explicitly set its value through a deterministic distribution type).

Click ‘OK’ to return to the stochastic tabbed pane.

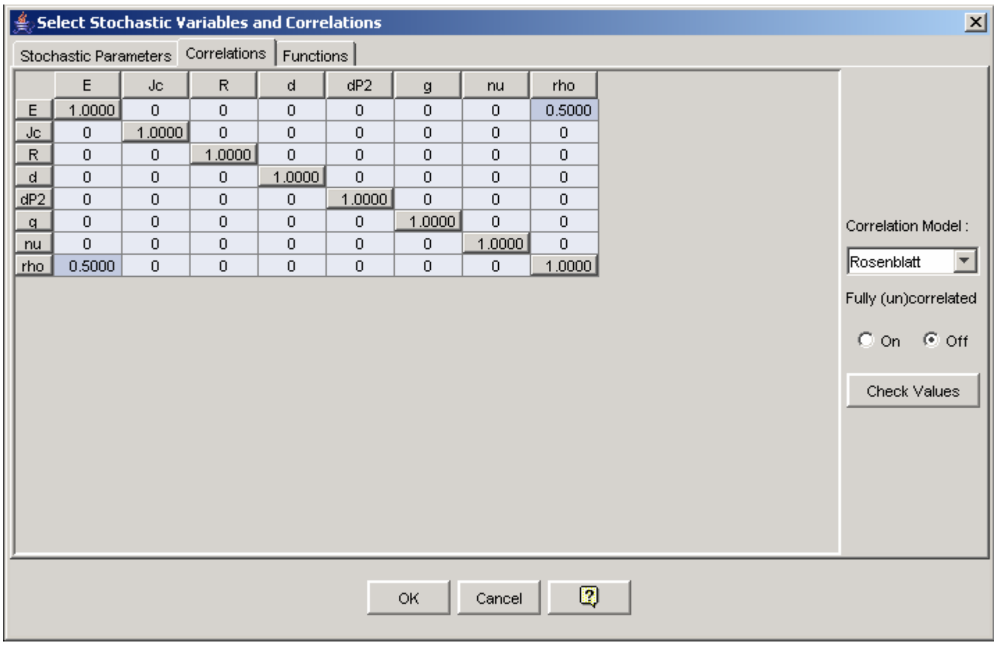


Index	On/Off	Name	Model	Parameter	Type	Mean	St.dev	Param. 1	Param. 2	Param. 3	Par
1	☑	E	Variable	E	Student	2.000E009	2.078E008	8.0	2.0E9	1.8E8	-
2	☑	Jc	Variable	Jc	Student	9.200E-03	9.839E-01	10.0	0.0092	0.88	-
3	☑	R	Variable	R	Deterministic	3.000E-01	-	0.3	-	-	-
4	☑	d	Variable	d	Student	4.600E-02	5.442E-03	23.0	0.046	0.0052	-
5	☑	dP2	Variable	dP2	Normal	4000.0	1000.0	4.000E003	1.000E003	-	-
6	☑	g	Variable	g	Deterministic	9.810E000	-	9.81	-	-	-

Figure 18 Main stochastic tabbed pane

In the second column one also sees the possibility of setting a stochastic parameter ‘on’ or ‘off’.

The second tab-pane gives access to the input window for defining correlations:



	E	Jc	R	d	dP2	g	nu	rho
E	1.0000	0	0	0	0	0	0	0.5000
Jc	0	1.0000	0	0	0	0	0	0
R	0	0	1.0000	0	0	0	0	0
d	0	0	0	1.0000	0	0	0	0
dP2	0	0	0	0	1.0000	0	0	0
g	0	0	0	0	0	1.0000	0	0
nu	0	0	0	0	0	0	1.0000	0
rho	0.5000	0	0	0	0	0	0	1.0000

Correlation Model : Rosenblatt

Fully (un)correlated ○ On ● Off

Figure 19 Input of correlations

In Prob2B, the Rosenblatt correlations are implemented. Values can be edited directly in the corresponding input fields. Symmetry is taken care of by Prob2B: only upper or lower diagonal terms need to be put in. The user can check the correlation matrix on its positive definiteness by clicking the ‘Check values’ button. When a matrix is not

positive definite, Prob2B will make adjustments so as to make it positive definite. In that case the user is prompted whether to accept the suggested values or not.
Finally, there is a button group (on/off) ‘Fully (un)correlated’. When switched to ‘on’, only {-1, 0, 1} is allowed as input value. However, more assistance is obtained in filling dependent matrix positions based on a newly entered value.

In Figure 19, we defined a correlation coefficient of 0.5 between the variables E and rho.
Clicking ‘OK’ returns to the main window of Prob2B. Before doing so, Prob2B will automatically check if alterations were performed on a correlation matrix and check its positive definiteness.
Going back through the Model selection menu ‘Define Stochasts’ or directly clicking on the Limit button retrieves the Stochast selection windows for adding or editing.

4.6 Parametric calculation settings

Prob2B provides a means to perform parametric calculations. A parameter can be told to successively take a different value between user defined boundaries. For each - temporarily fixed - parameter value, a probabilistic calculation can be performed conform the user’s settings.
Parametric calculations can (so far) only be performed in up to 2 dimensions, i.e. a maximum of two parameters can be parametrically varied.

In the model menu one can select “Define Parameters”. Alternatively one can click directly on the Parametric button. Initially, the following window appears:

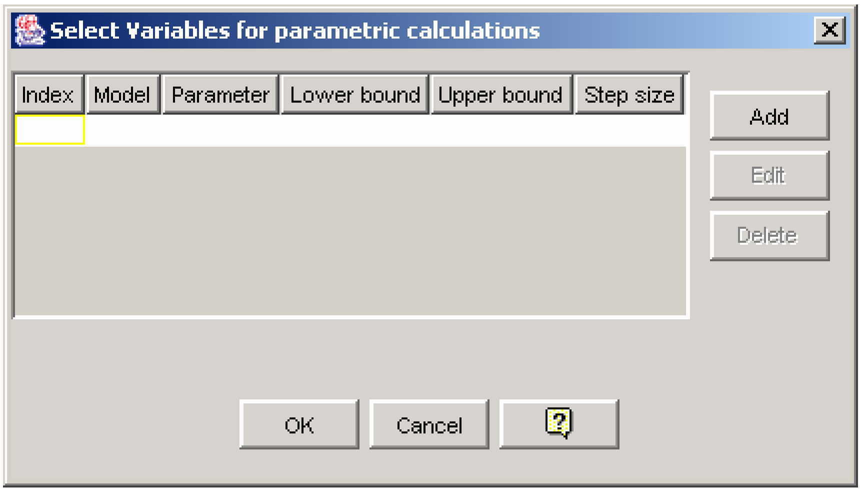
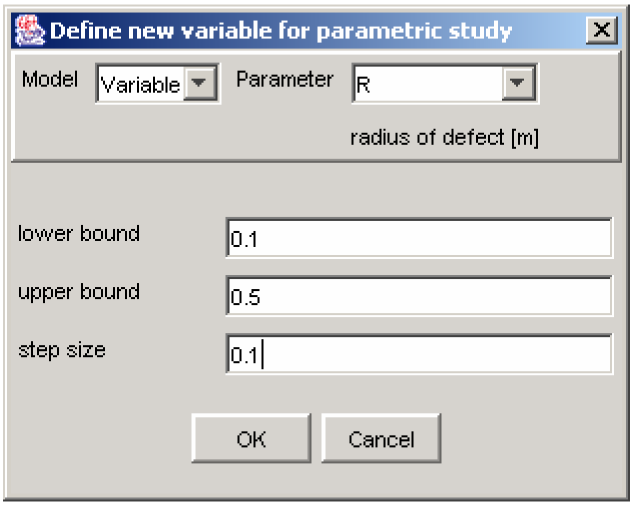


Figure 20 Parametric window.

By clicking the ‘Add’ button the user can (start to) define his parametric variables. The definition window looks like:



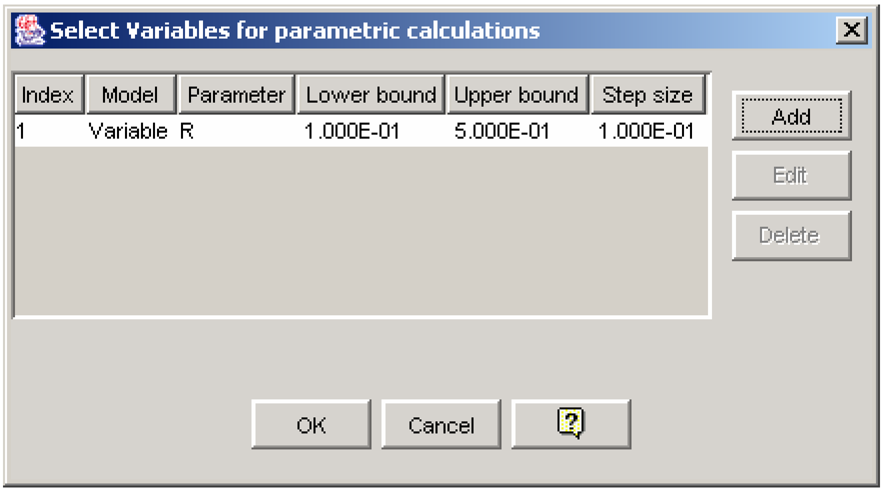
Model: Variable
Parameter: R
radius of defect [m]

lower bound: 0.1
upper bound: 0.5
step size: 0.1

Figure 21 Input window for parametric values.

In the example shown in Figure 21, first the model is selected combined with the parameter that has to be varied parametrically in its value. A lower and upper bound and a stepsize define its values. Hence, for the example, probabilistic calculations will be performed with 'R' from the set of variables set to 0.1, 0.2, 0.3, 0.4 and 0.5. in value.

Clicking OK results in



Index	Model	Parameter	Lower bound	Upper bound	Step size
1	Variable	R	1.000E-01	5.000E-01	1.000E-01

Figure 22 Overview of Parametric settings

Selecting a row in the table, gives access to the 'Delete' and 'Edit' menus for the corresponding parameter.

Clicking 'OK' again returns to the main Prob2B window.

Going back through the Model selection menu ‘Define Parameters’ or directly clicking on the Parameter button retrieves the Parametric selection windows for adding or editing.

4.7 Hierarchy in variable settings

In the preceding paragraphs the variable R was set as deterministic in the models menu. Next, in the stochastic menu, the settings for R were left unaltered, although it could have been set to a(nother) distribution type. Finally, in paragraph 4.6, R was set parametric.

Not dealt with yet, is the possibility to put dependencies in models and/or variables (this will be described via a fourth example in chapter 0).

Giving all these possibilities, a parameter can be defined as:

- Set to parametric values
- Dependent on another variable
- Stochastic (including deterministic).

Input for these settings are such that they do not exclude each other in the input menus. This means for example, that a parameter can be defined by its distribution parameters but still can be selected for parametric settings (and model dependencies) and vice versa.

In calculations, Prob2B will handle these settings conform a hierarchy depicted in Figure 23:

1. If, for a variable, parametric values were entered, these will be used in calculations. Parametric values overrule model dependencies and stochastic settings.
2. If, for a variable, model dependencies are defined (but no parametric values) the dependencies will be taken into account. Model dependencies overrule stochastic settings, but are themselves overruled by parametric settings.
3. Finally, if no parametric settings or dependencies are defined, the stochastic (or deterministic) properties are used in calculations.

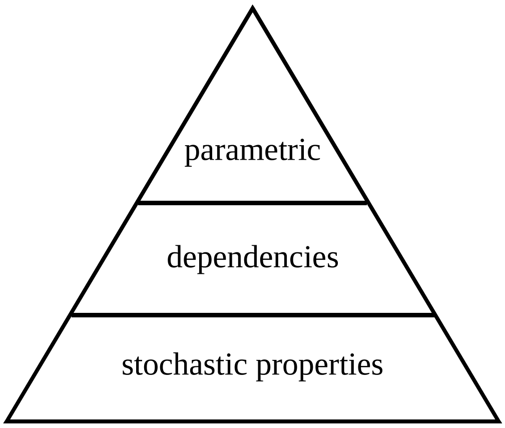
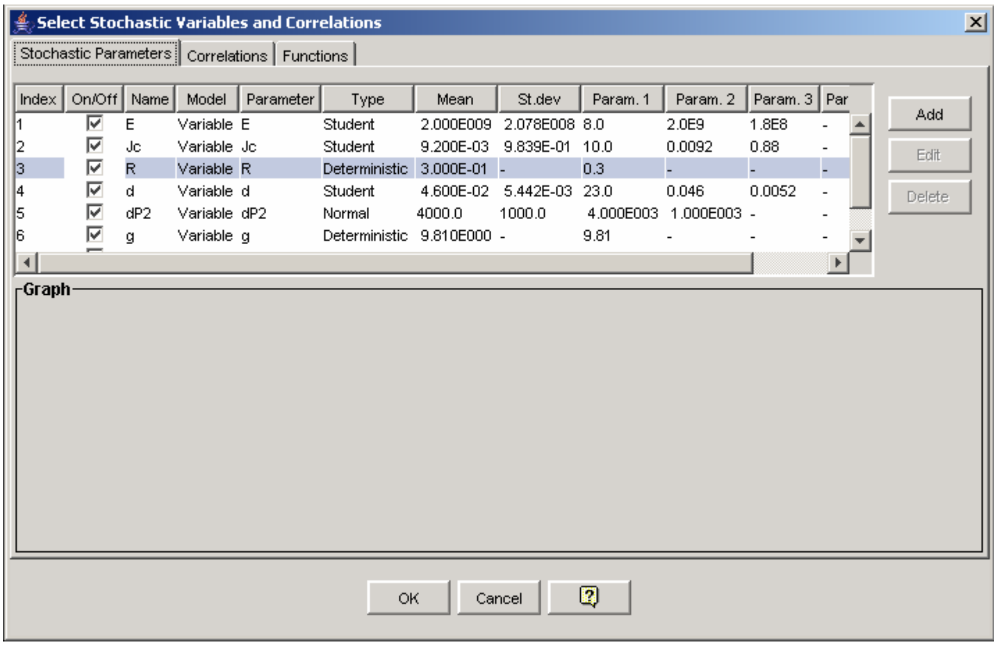


Figure 23 Hierarchy in parameter settings

Indications whether a variable is overruled by other settings can be seen in the appropriate menus in that such a variable is listed against a light blue background. For instance in the example at hand we overruled the stochastic values for R by parametric values. Going back into the stochastic menu will show a table as in Figure 24.



Index	On/Off	Name	Model	Parameter	Type	Mean	St.dev	Param. 1	Param. 2	Param. 3	Par
1	☑	E	Variable	E	Student	2.000E009	2.078E008	8.0	2.0E9	1.8E8	-
2	☑	Jc	Variable	Jc	Student	9.200E-03	9.839E-01	10.0	0.0092	0.88	-
3	☑	R	Variable	R	Deterministic	3.000E-01	-	0.3	-	-	-
4	☑	d	Variable	d	Student	4.600E-02	5.442E-03	23.0	0.046	0.0052	-
5	☑	dP2	Variable	dP2	Normal	4000.0	1000.0	4.000E003	1.000E003	-	-
6	☑	g	Variable	g	Deterministic	9.810E000	-	9.81	-	-	-

Figure 24 Stochastic settings overruled

Here we see that the corresponding row in the table is coloured in a light blue tint.

4.8 Correlations in parametric calculations

In situations where variables with stochastic properties and correlations are defined and then overruled by parametric settings for the variables, the preset correlations are no longer accounted for.
In the present implementation, the stochastic properties including the correlations are (temporarily) removed.

4.9 Setting the reliability calculation method

The settings for the reliability calculations are set through the “Calc” menu, by selecting ‘Select Reliability Method’. Alternatively one can click the ‘tools’ button .
A method selection window appears, in which the user may select one of the methods from Table 3. (the blue coloured methods in Figure 25 are not yet implemented).

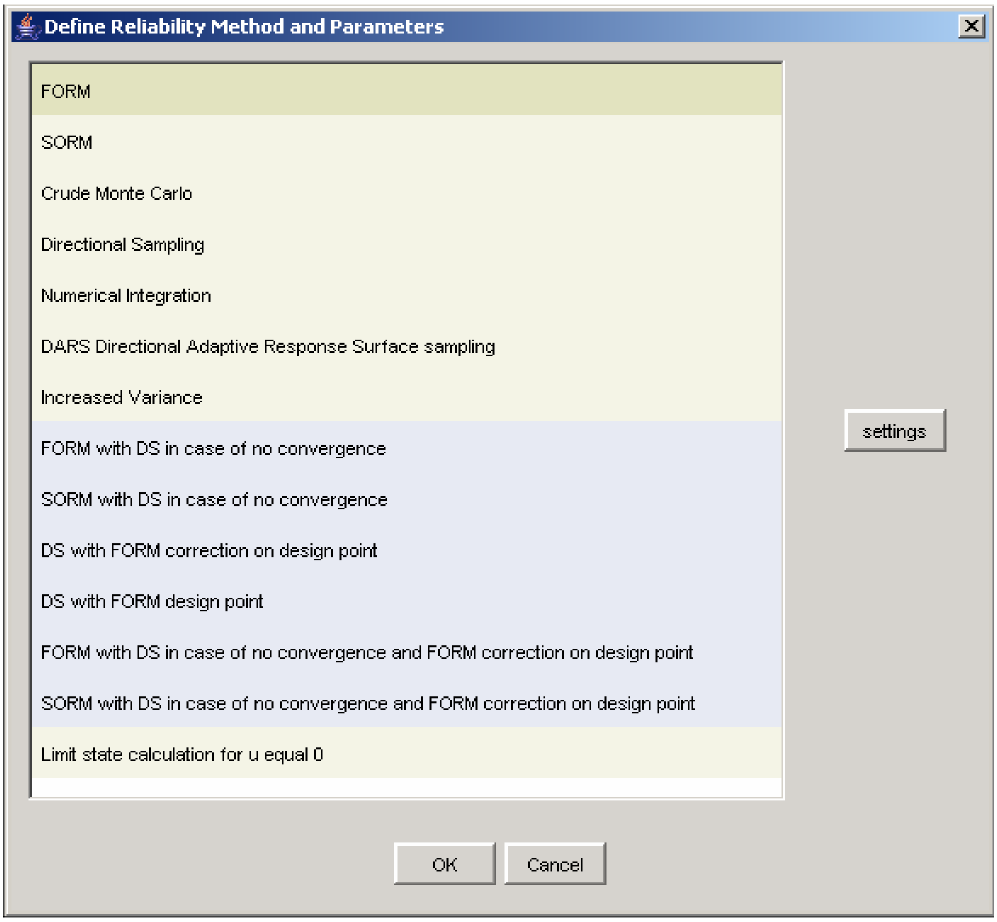
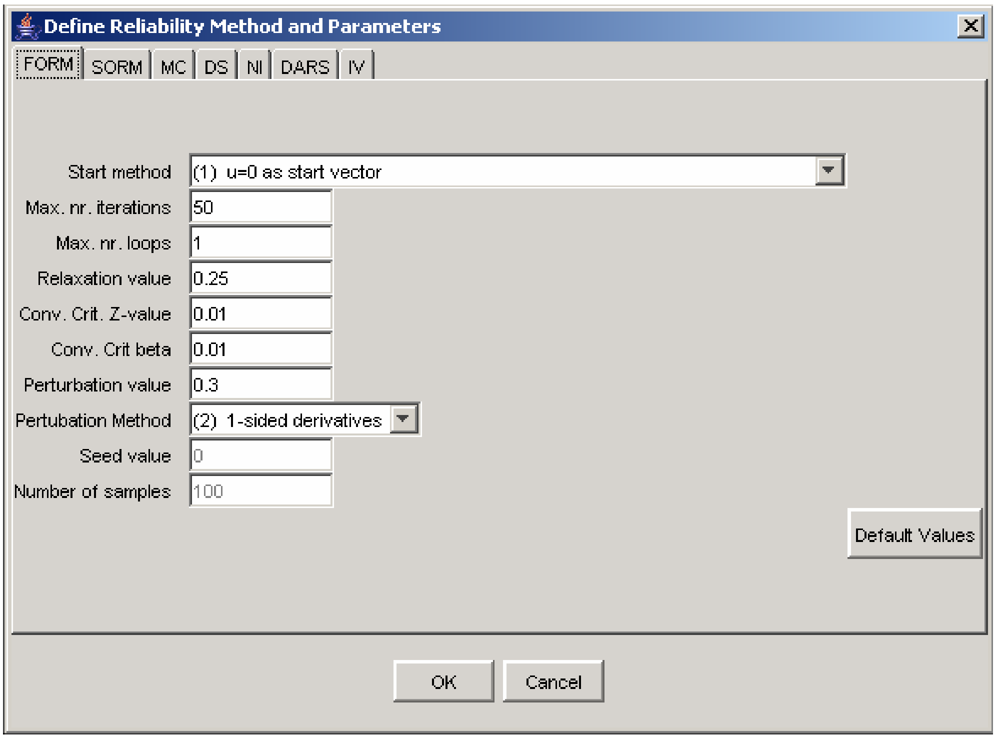

Define Reliability Method and Parameters
FORM
SORM
Crude Monte Carlo
Directional Sampling
Numerical Integration
DARS Directional Adaptive Response Surface sampling
Increased Variance
FORM with DS in case of no convergence
SORM with DS in case of no convergence
DS with FORM correction on design point
DS with FORM design point
FORM with DS in case of no convergence and FORM correction on design point
SORM with DS in case of no convergence and FORM correction on design point
Limit state calculation for u equal 0
settings
OK
Cancel

Figure 25 Selecting the reliability method

Table 3 Available methods in Prob2B

FORM	First Order Reliability Method
SORM	Second Order Reliability Method
CMC	Crude Monte Carlo
DS	Directional Sampling
NI	Numerical Integration
DARS	Directional Adaptive Response Surface Sampling
IV	Increased Variance
SC	Single calculation for u=0

Some background theory on these methods can be found in Appendix A.
Each method will require some settings to be assigned. Prob2B will prompt the user with default values. The user may alter thes settings by clicking the settings menu. The corresponding input windows for these settings are presented in Figure 26 to Figure 31, together with a brief description of the input values in Table 4 to Table 9.

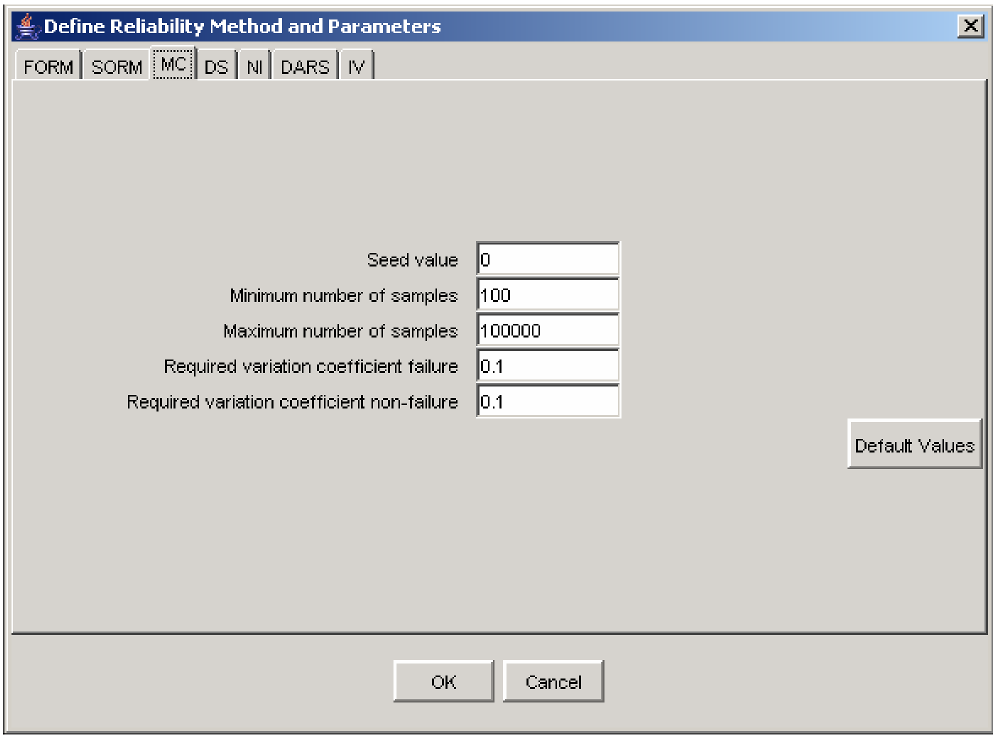


Start method: (1) u=0 as start vector
Max. nr. iterations: 50
Max. nr. loops: 1
Relaxation value: 0.25
Conv. Crit. Z-value: 0.01
Conv. Crit beta: 0.01
Perturbation value: 0.3
Pertubation Method: (2) 1-sided derivatives
Seed value: 0
Number of samples: 100

Figure 26 Input panel for FORM and SORM (SORM will prompt for comparable settings)

Table 4 Settings for FORM and SORM

Item	**Description**	**Input values**	**Default values**
Start method	Indication for selecting the first point (in U-space) to start the FORM or SORM iterations from.	1: $\underline{u} = \underline{0}$ 2: $\underline{u} = \underline{1}$	1
Max. nr. iterations	Maximum number of iterations for a FORM/SORM calculation	>0	50
Max. nr. loops	Maximum number of loops in search for convergence	>0	1
Relaxation value	Adaptation of the step size during FORM/SORM iterations. E.g. 0.3 means that 0.3 times the corrective step is applied. The iterations can thus be set to be 'less energetic' and 'more cautious'. Although smaller steps are taken, it can make the method more robust for irregular limit state functions.	>0 <=1	0.25
Conv. crit. Z-value	A convergence criterion based on the limit state functions being near enough to 0	>0	0.01
Conv. crit. beta	A convergence criterion based on having found a minimum for beta. (Updates for beta becoming small)	>0	0.01
Perturbation value	Perturbation value for calculating numerical derivatives of the limit state function per stochastic parameter. Values in units standard deviation.	>0	0.3
Perturbation method	Numerical derivatives of the Z functions per stochastic parameter are calculated either one or two-sided (central).	1: central 2: one-sided	2
Seed value	Seed value for initialising the random generator, start method option 4 is selected. Values >0 are directly used as seed value and can be used to regenerate a sequence of random numbers. A value of 0 will 'randomise' the initialisation itself, by retrieving the seed value from the current clock settings. Each run will hence generate a new sequence of random numbers	>=0	0
Number. Of Samples	Number of sampling the Z-function (=number of calculations) when sart method option 4 is selected. Resulting sample with value closest to zero will be used as start vector.	>0	100



Seed value: 0
Minimum number of samples: 100
Maximum number of samples: 100000
Required variation coefficient failure: 0.1
Required variation coefficient non-failure: 0.1

Figure 27 Input panel for Crude Monte Carlo

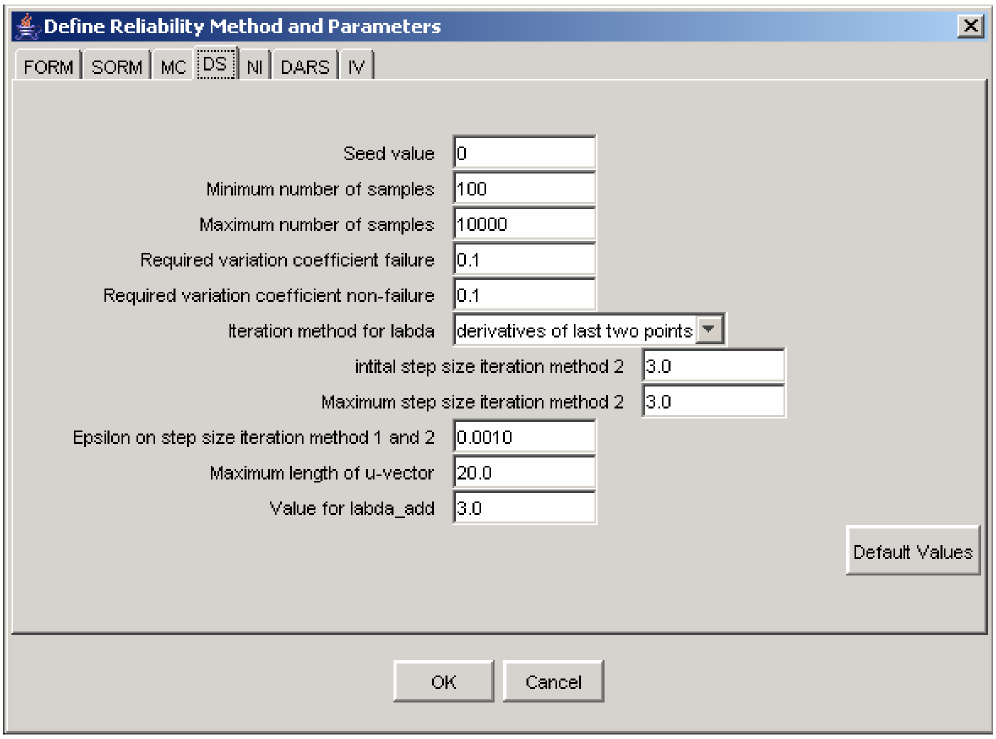
Table 5 Settings for Crude Monte Carlo

Item	Description	Input values	Default value
Seed value	Seed value for initialising the random generator. Values >0 are directly used as seed value and can be used to regenerate a sequence of random numbers. A value of 0 will 'randomise' the initialisation itself, by retrieving the seed value from the current clock settings. Each run will hence generate a new sequence of random numbers	>=0	0
Min. nr. Of Samples	Minimum number of sampling the Z-function (=number of calculations) for Crude Monte Carlo.	>0	100
Max. nr. Of Samples	Maximum number sampling the Z-function (=number of calculations) for Crude Monte Carlo.	>0	10000
Variance. coefficient failure	Required Variance coefficient for probability of failure	>0	0.1
Variance. coefficient non failure	Required Variance coefficient for probability of non-failure	>0	0.1

The Crude Monte Carlo Calculations will stop when

The convergence criteria are met, provided the minimum number of calculations are obtained, or

The maximum number of calculations is exceeded.

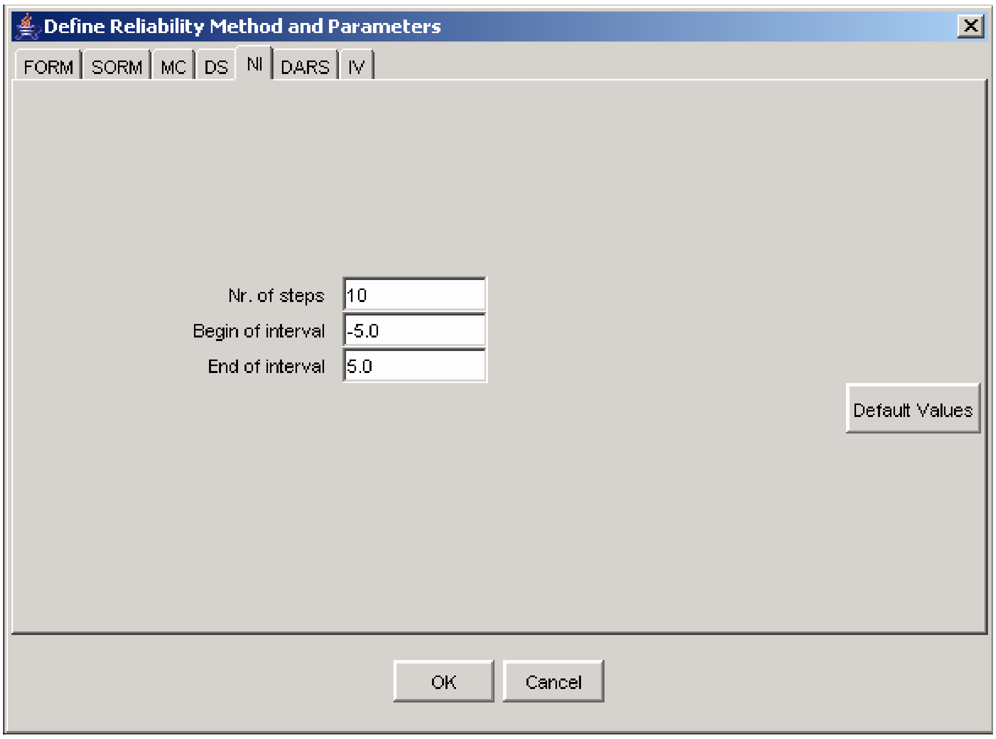


Seed value	0
Minimum number of samples	100
Maximum number of samples	10000
Required variation coefficient failure	0.1
Required variation coefficient non-failure	0.1
Iteration method for labda	derivatives of last two points
intital step size iteration method 2	3.0
Maximum step size iteration method 2	3.0
Epsilon on step size iteration method 1 and 2	0.0010
Maximum length of u-vector	20.0
Value for labda_add	3.0

Figure 28 Input panel for Directional Sampling

Table 6 Settings for Directional Sampling

Item	**Description**	**Input values**	**Default value**
Seed value	Seed value for initialising the random generator. Values >0 are directly used as seed value and can be used to regenerate a sequence of random numbers. A value of 0 will 'randomise' the initialisation itself, by retrieving the seed value from the current clock settings. Each run will hence generate a new sequence of random numbers	>=0	0
Min. nr. Of Samples	Maximum number of search directions in case of Directional Sampling (<= number of calculations).	>0	100
Max. nr. Of Samples	Maximum number of search directions in case of Directional Sampling (<= number of calculations).	>0	10000
Variance. coefficient failure	Required Variance coefficient for probability of failure	>0	0.1
Variance. coefficient non failure	Required Variance coefficient for probability of non-failure	>0	0.1
	T.B.D.		

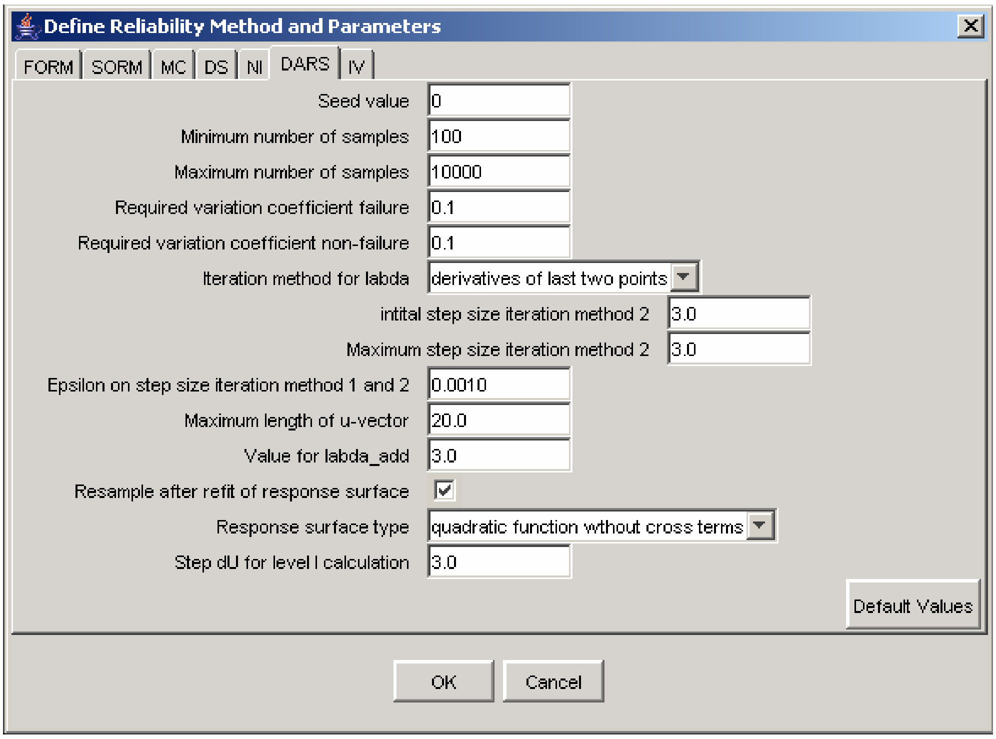


Nr. of steps 10
Begin of interval -5.0
End of interval 5.0

Figure 29 Input panel for numerical Integration

Table 7 Settings for Numerical Integration

Item	**Description**	**Input values**
Number of steps	Number of (equidistant) steps in sampling the interval	>0
Begin of interval	Starting value for the interval. Values are in U -space.	Less than end of interval
End of interval	Upper bound value for the interval. Values are in U -space.	Greater than begin of interval

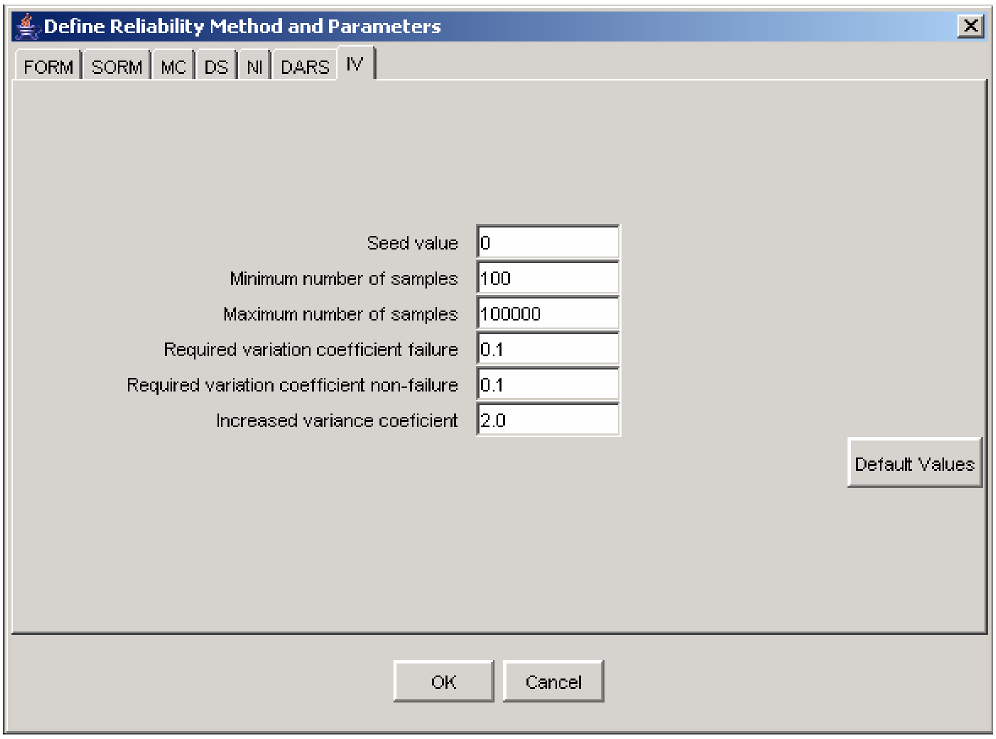


Seed value 0
Minimum number of samples 100
Maximum number of samples 10000
Required variation coefficient failure 0.1
Required variation coefficient non-failure 0.1
Iteration method for labda derivatives of last two points
intital step size iteration method 2 3.0
Maximum step size iteration method 2 3.0
Epsilon on step size iteration method 1 and 2 0.0010
Maximum length of u-vector 20.0
Value for labda_add 3.0
Resample after refit of response surface ☑
Response surface type quadratic function wthout cross terms
Step dU for level I calculation 3.0

Figure 30 Input panel for DARS

Table 8 Settings for DARS

Item	**Description**	**Input values**	**Default value**
Seed value	Seed value for initialising the random generator. Values >0 are directly used as seed value and can be used to regenerate a sequence of random numbers. A value of 0 will 'randomise' the initialisation itself, by retrieving the seed value from the current clock settings. Each run will hence generate a new sequence of random numbers	>=0	0
Min. nr. Of Samples	Maximum number of search directions in case of Directional Sampling (<= number of calculations).	>0	100
Max. nr. Of Samples	Maximum number of search directions in case of Directional Sampling (<= number of calculations).	>0	10000
Variance. coefficient failure	Required Variance coefficient for probability of failure	>0	0.1
Variance. coefficient non failure	Required Variance coefficient for probability of non-failure	>0	0.1
	T.B.D.		



Seed value 0
Minimum number of samples 100
Maximum number of samples 100000
Required variation coefficient failure 0.1
Required variation coefficient non-failure 0.1
Increased variance coeficient 2.0

Figure 31 Input panel for Increased Variance

Table 9 Settings for Increased Variance

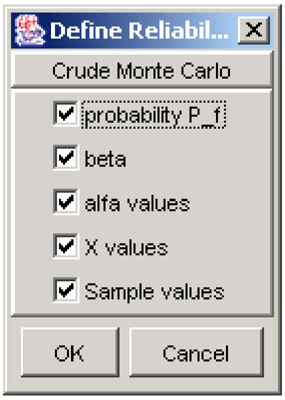
Item	**Description**	**Input values**	**Default value**
Seed value	Seed value for initialising the random generator. (see Crude Monte Carlo)	>=0	0
Min. nr. Of Samples	Minimum number of sampling the Z-function (=number of calculations) for Crude Monte Carlo.	>0	100
Max. nr. Of Samples	Maximum number sampling the Z-function (=number of calculations) for Crude Monte Carlo.	>0	10000
Variance. coefficient failure	Required Variance coefficient for probability of failure	>0	0.1
Variance. coefficient non failure	Required Variance coefficient for probability of non-failure	>0	0.1
Increased variance coefficient	Multiplication factor for parameter variances	>0	2

After selecting and setting the calculation method, one returns to the main window by clicking the ‘OK’ button. Clicking ‘Cancel’ will leave the calculation method unaltered, being the settings set previously or the default settings for FORM.

4.10 Setting the output options

The output options for the calculations need to be set prior to calculation. You access these options trough the calc menu, option “output selection menu”, or by clicking the ‘results’ button

Doing so makes the following window appear:



☑ probability P_f
☑ beta
☑ alfa values
☑ X values
☑ Sample values

Figure 32 Defining reliability output results

Items can be marked in order to get them calculated and shown as output. These items are described in Table 10.

Table 10 Reliability Output Results

Item	**Description**	**Remarks**
probability P_f	Probability of failure (limit state Z less than 0)	All methods
Beta	Reliability index	All methods
alfa values	Influence factors for the stochastic parameters upon the Z function in the design point.	**FORM**: direct calculation result **SORM**: idem **Monte Carlo**: alfa values are based on the design point in U-space (approximation) **Directional Sampling**: idem **Numerical Integration**: idem
X values	The values of the stochastic parameters in the design point	**FORM**: direct calculation result **SORM**: design point is replaced (rescaled) such that its length in U-space corresponds with the reliability index (i.e. presented as FORM result) **Monte Carlo**: failure point closest to the origin in U-space is rescaled such that its length corresponds with the reliability index (approximation) **Directional Sampling**: idem **Numerical Integration**: idem
Sample values	The sample values for the stochastic parameters and the	All methods

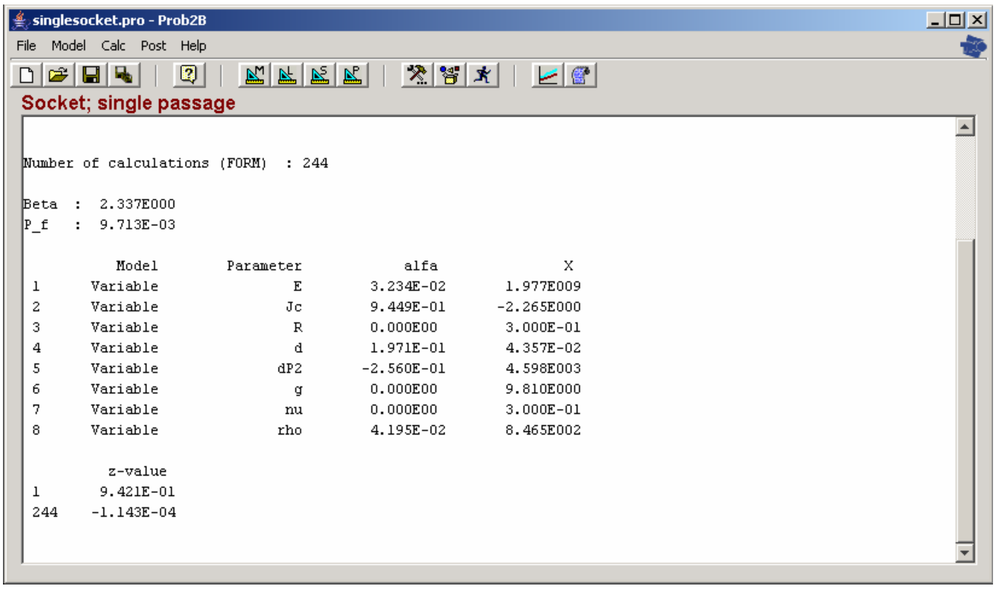
	corresponding limit state function values are kept in memory, for post processing purposes	When parametric calculations are performed, only the sample values and limit state values belonging to the last parameter set are kept track of.

The user should be aware of the fact that with no samples saved during the calculations, graphical post processing (see paragraph 4.14) will not be accessible.

4.11 Running the calculations

The calculations are run through pressing the Run button or selecting the ‘Run’ option from the ‘Calc’ menu.

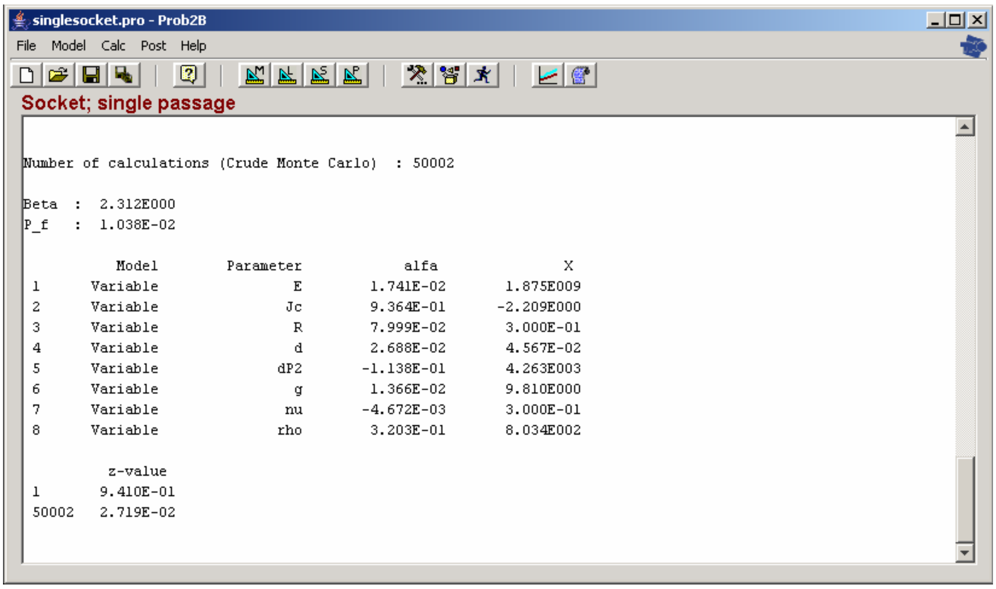
When the calculations are finished, the results are summarized in the output pane of the main Prob2B window. (For this output pane a copy/paste functionality is operational for transferring this data to other applications like text editors).
Figure 33 shows an example of such output (the parametric calculations are not performed here, the calculation results are for a radius R of 0.3 m).



```
Number of calculations (FORM)  : 244

Beta  :  2.337E000
P_f   :  9.713E-03

         Model        Parameter         alfa            X
 1      Variable              E       3.234E-02      1.977E009
 2      Variable             Jc       9.449E-01     -2.265E000
 3      Variable              R       0.000E00       3.000E-01
 4      Variable              d       1.971E-01      4.357E-02
 5      Variable            dP2      -2.560E-01      4.598E003
 6      Variable              g       0.000E00       9.810E000
 7      Variable             nu       0.000E00       3.000E-01
 8      Variable            rho       4.195E-02      8.465E002

        z-value
 1       9.421E-01
 244    -1.143E-04
```

Figure 33 Output after FORM calculation

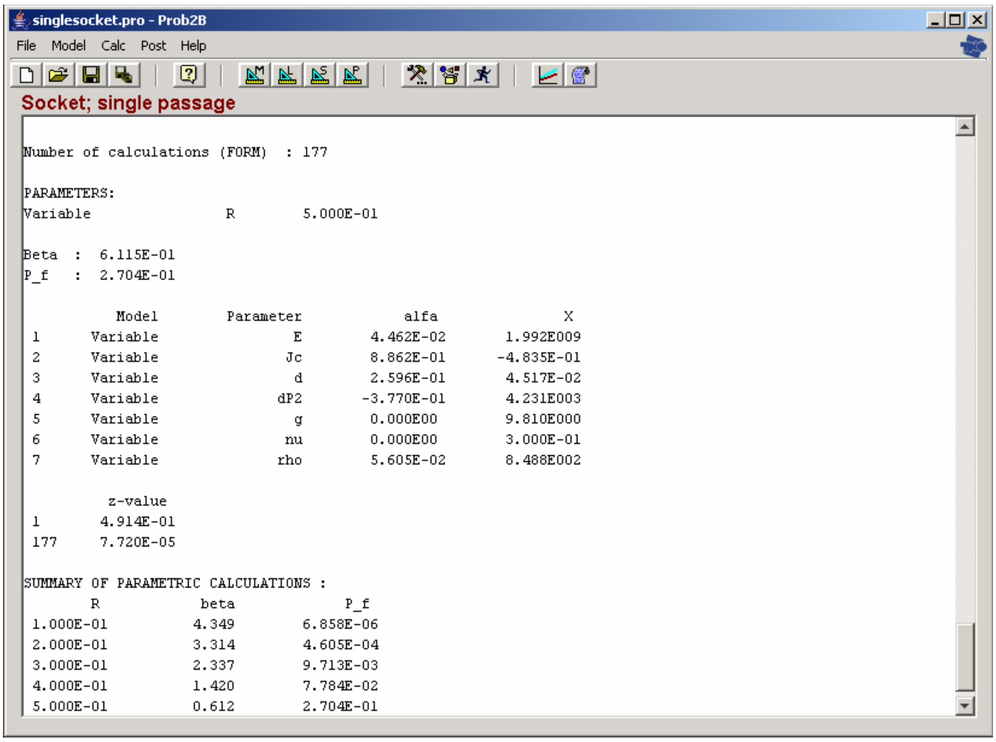
The design point is dominated by the values for Jc as indicated by its high valued influence factor alfa.
A probability of failure is calculated of 0.0097 with a corresponding reliability index of 2.34.
With respect to the Z-values listed in the output pane, only the first and last calculated values are presented. This is only done to give an impression of the resulting Z-value being small compared to the initially calculated value.



```
Number of calculations (Crude Monte Carlo)  : 50002

Beta  :  2.312E000
P_f   :  1.038E-02

            Model         Parameter           alfa            X
 1      Variable                  E       1.741E-02      1.875E009
 2      Variable                 Jc       9.364E-01     -2.209E000
 3      Variable                  R       7.999E-02      3.000E-01
 4      Variable                  d       2.688E-02      4.567E-02
 5      Variable                dP2      -1.138E-01      4.263E003
 6      Variable                  g       1.366E-02      9.810E000
 7      Variable                 nu      -4.672E-03      3.000E-01
 8      Variable                rho       3.203E-01      8.034E002

           z-value
 1       9.410E-01
 50002   2.719E-02
```

Figure 34 Output after 50000 Monte Carlo simulations

Figure 34shows the results when the reliability is calculated with Monte Carlo. Fifty thousand model calculations are performed. One can see that the probability of failure and the reliability index compare well with the results of FORM (Figure 33). With respect to the design point and the influence factors, the results are to be interpreted with more care due to the approximations made.
The listings for the first and last values for the limit state Z have to be seen as such, without any interpretation as for FORM.



```
Number of calculations (FORM)  : 177

PARAMETERS:
Variable                R        5.000E-01

Beta  :  6.115E-01
P_f   :  2.704E-01

            Model         Parameter           alfa            X
 1      Variable                  E       4.462E-02      1.992E009
 2      Variable                 Jc       8.862E-01     -4.835E-01
 3      Variable                  d       2.596E-01      4.517E-02
 4      Variable                dP2      -3.770E-01      4.231E003
 5      Variable                  g       0.000E00       9.810E000
 6      Variable                 nu       0.000E00       3.000E-01
 7      Variable                rho       5.605E-02      8.488E002

           z-value
 1       4.914E-01
 177     7.720E-05

SUMMARY OF PARAMETRIC CALCULATIONS :
        R             beta            P_f
 1.000E-01          4.349         6.858E-06
 2.000E-01          3.314         4.605E-04
 3.000E-01          2.337         9.713E-03
 4.000E-01          1.420         7.784E-02
 5.000E-01          0.612         2.704E-01
```

Figure 35 Output after Parametric Calculations (5 FORM reliability calculations)

When parametric calculations are activated, the output pane shows a sequence of summaries for each reliability calculation per parameter set, Figure 35. The layout per calculation is comparable to the ones presented in Figure 33 and Figure 34, with the corresponding parametric values added.
At the end of the calculations a table is presented showing a list of the reliability factors and the probabilities of failure for the parametric values.

4.12 Saving the project

The project can be saved using the save project option in the file menu. Saving the project results in an ASCII file containing all project settings, except for the output.

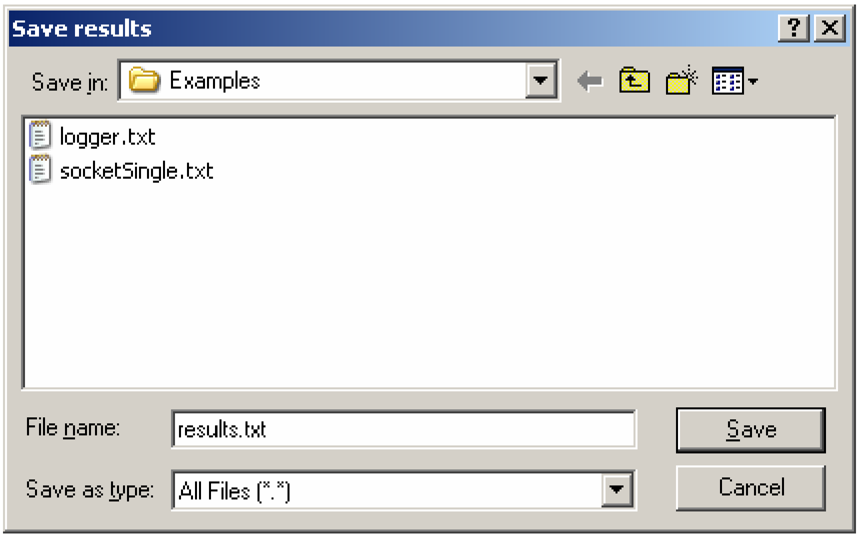
The project is saved by pressing the save button or selecting the ‘Save’ option from the ‘File’ menu. In doing so the project will be saved under the name that was defined while opening the new project. In this example it would be ‘socketSingle.txt’.

Alternatively one can save the project under a different name, using the ‘Save as’ button or option from the ‘File’ menu. Future save actions during the session will then automatically use the newly defined file name, leaving the original one unaltered (since his last save).

Saved project files can be reloaded into Prob2B, see section 4.15.

4.13 Saving the calculation results

When one wants to save the calculation results into a text-file, one can select the ‘Report’ button. Alternatively the ‘Report Results’ option can be chosen under the ‘Post’ menu.
Prob2B will then prompt with a file dialog as in Figure 36.



Save in: Examples

logger.txt
socketSingle.txt

File name: results.txt
Save as type: All Files (*.*)

Figure 36 File selection dialog for saving calculation results

The user can select or type its filename (other than the project file) and save his calculation results.
The results are then saved in a tab-delimited text file.

At the top of the file project specific data is echoed, such that the results file can also be interpreted as a project file. The results file could thus be used for (re-)loading as an existing project. Results listed in the file will however **not** be reloaded into a new Prob2B Session.

The results file will give the calculation summaries that also appeared in the output pane of Prob2B.
If keeping track of samples was selected (paragraph 4.10), then they will also be dumped into the file. However, if this was done in combination with parametric calculations, only the samples for the **last** calculated parameter set will be saved.

4.14 Graphical views of results

Depending on the kind of calculation (parametric or not) and the selected results (saved samples), a number of graphical representations can be generated within Prob2B.

To activate this option the user can click the 'Graph' button 📈 or select the corresponding item under the 'post' menu. The following window then appears:

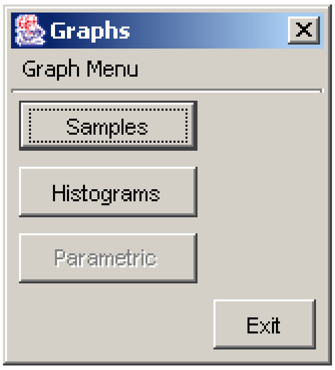


Figure 37 Starting window with respect to Graphs

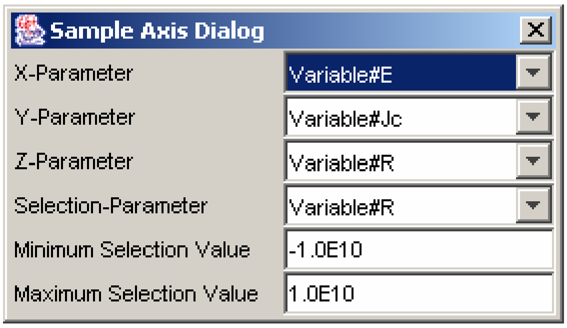
The Graph menu will depend on the results saved. Before Figure 35 a non-parametric calculation was made, with samples saved. Hence the first two buttons, being 'Samples' and 'Histograms' are enabled, whereas the 'Parametric' button is disabled.

The user should be aware of the fact that with no samples saved during the calculations, the first two options are inaccessible. (See paragraph 4.10).

4.14.1 Samples

When selecting the 'Samples' button from the main Graphs window (Figure 37), a 2D graph with samples presented dots will appear. The Graph is combined with a table showing the sample values. The example in this manual was run with 50000 Monte Carlo samples. The initial graphic pane then looks like the one shown in Figure 40.

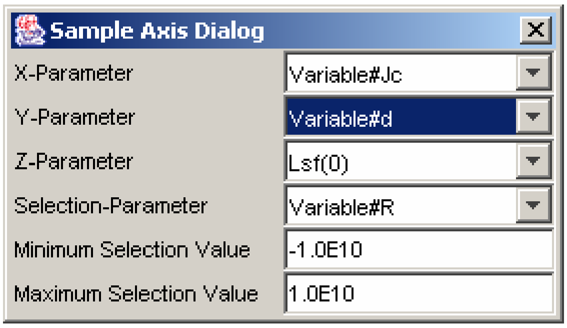
Giving a right mouse click in the data table pane activates an Axis Dialog, enabling the selection of the parameters to be viewed in the graph. The parameter on the horizontal axis is selected as X-parameter. On the vertical axis the selected Y-parameter is presented. The Z-parameter is shown in the graph by its colouring.
The data presented in the graph can furthermore be confined to a subset of the total data set. This is done by defining a Selection parameter and giving appropriate boundaries.



X-Parameter	Variable#E
Y-Parameter	Variable#Jc
Z-Parameter	Variable#R
Selection-Parameter	Variable#R
Minimum Selection Value	-1.0E10
Maximum Selection Value	1.0E10

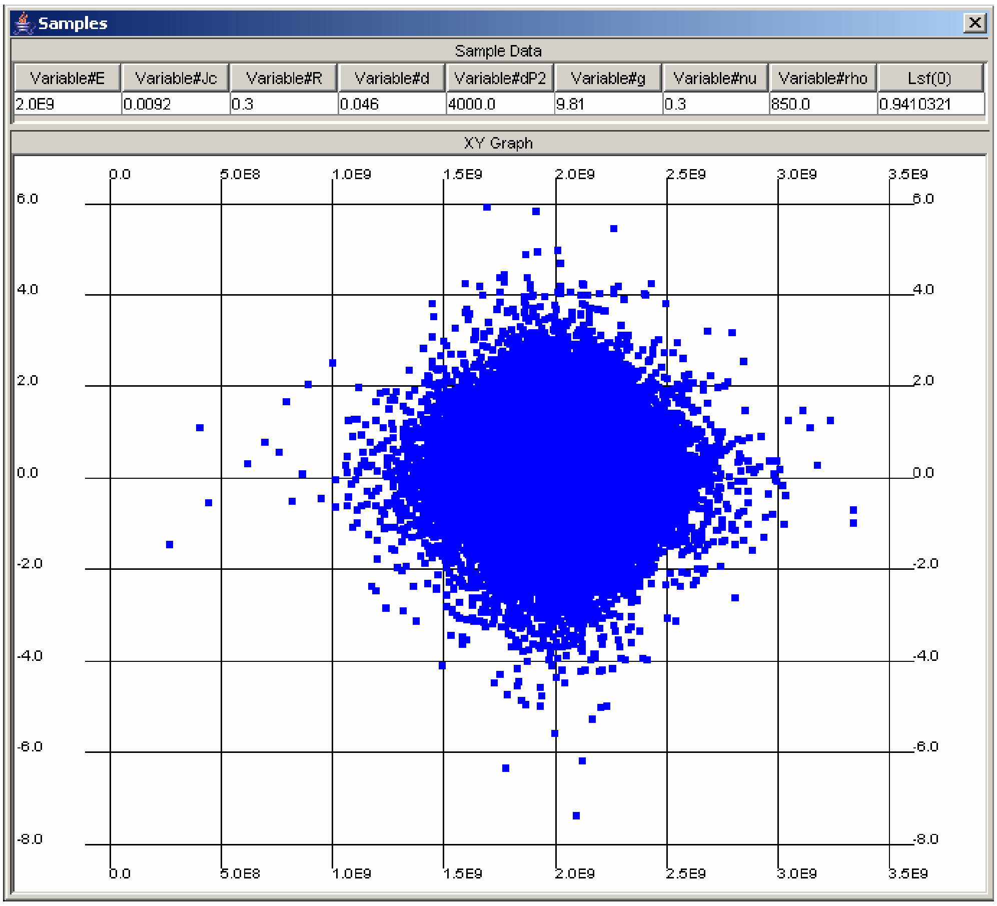
Figure 38 Axis dialog

Changing the selection of Figure 38 into the ones of Figure 39 will change the graph in Figure 40into the one shown in Figure 41.



X-Parameter	Variable#Jc
Y-Parameter	Variable#d
Z-Parameter	Lsf(0)
Selection-Parameter	Variable#R
Minimum Selection Value	-1.0E10
Maximum Selection Value	1.0E10

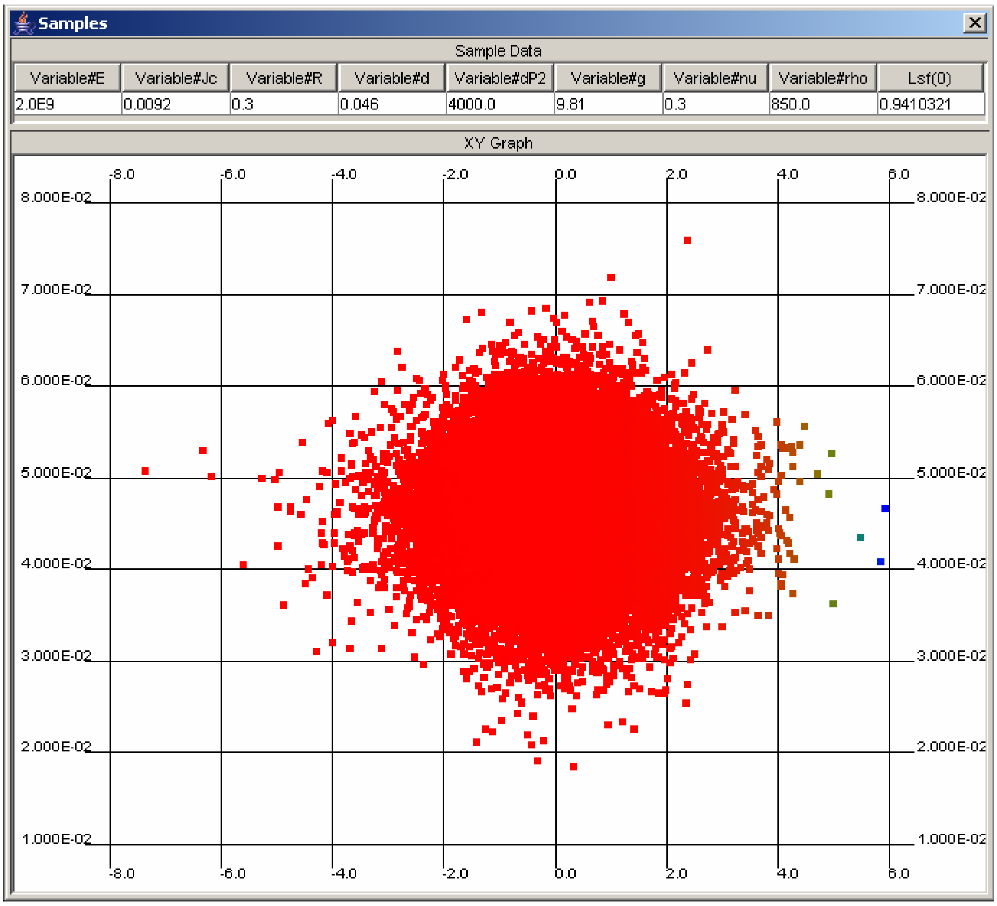
Figure 39 limit State Function selected as Z-Parameter



Sample Data

Variable#E	Variable#Jc	Variable#R	Variable#d	Variable#dP2	Variable#g	Variable#nu	Variable#rho	Lsf(0)
2.0E9	0.0092	0.3	0.046	4000.0	9.81	0.3	850.0	0.9410321

Figure 40 Initial Graph for samples.



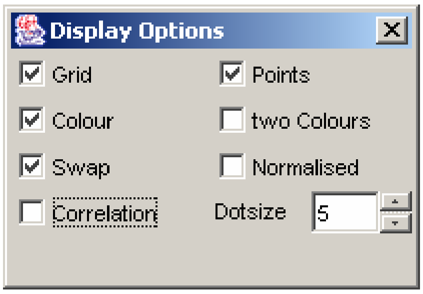
Sample Data

Variable#E	Variable#Jc	Variable#R	Variable#d	Variable#dP2	Variable#g	Variable#nu	Variable#rho	Lsf(0)
2.0E9	0.0092	0.3	0.046	4000.0	9.81	0.3	850.0	0.9410321

Figure 41 Samples plot with limit state as Z-parameter (coloured values)

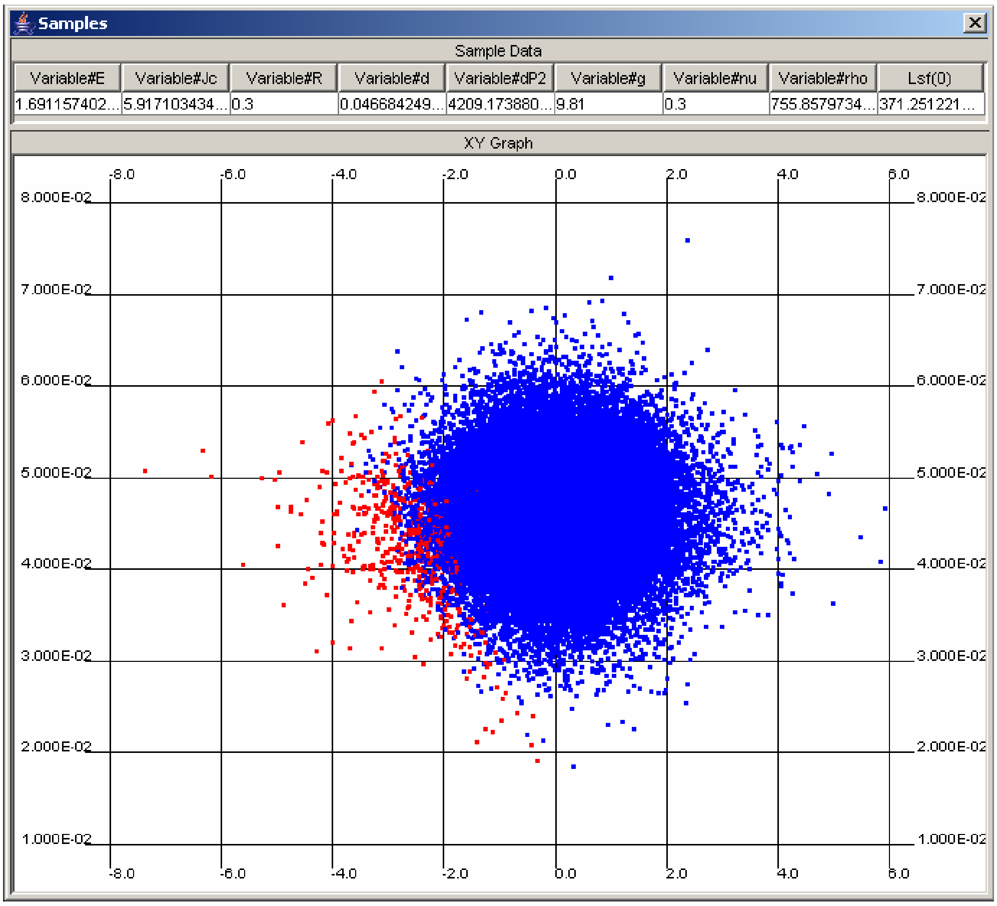
The coloured values for the limit state go from low (red) to high (blue)

Giving a right mouse click in the graphic pane activates a Display Options Dialog, with which the appearance of the graph can be altered. Figure 45 shows this dialog window. Most options are supposed to be self-explaining, their effect can easily be found by clicking on and of. Only two options will be shown in this manual: ‘two Colours’ and ‘Correlation’. Checking the item ‘two Colours’ to ‘on’ and reducing the dotsize to 3, results in Figure 43.



☑ Grid ☑ Points
☑ Colour ☐ two Colours
☑ Swap ☐ Normalised
☐ Correlation Dotsize 5

Figure 42 Display options



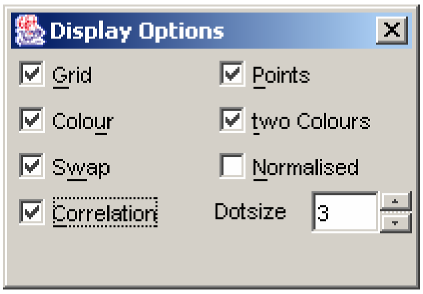
Sample Data

Variable#E	Variable#Jc	Variable#R	Variable#d	Variable#dP2	Variable#g	Variable#nu	Variable#rho	Lsf(0)
1.691157402...	5.917103434...	0.3	0.046684249...	4209.173880...	9.81	0.3	755.8579734...	371.251221...

Figure 43 Displaying a two-coloured graph

In Figure 43, we see that the values for the Z-parameter (being selected to be the limit state function) are divided into two groups: those with values less then 0 (red coloured) and those with positive values (blue coloured). It facilitates the interpretation of the limit state values, giving an impression of the ratio between failure and non-failure points. It also shows how the failure plane cuts through the 2 dimensional plane defined by the 2 selected input parameters. (Sharp/fuzzy, straight/curved etc.)

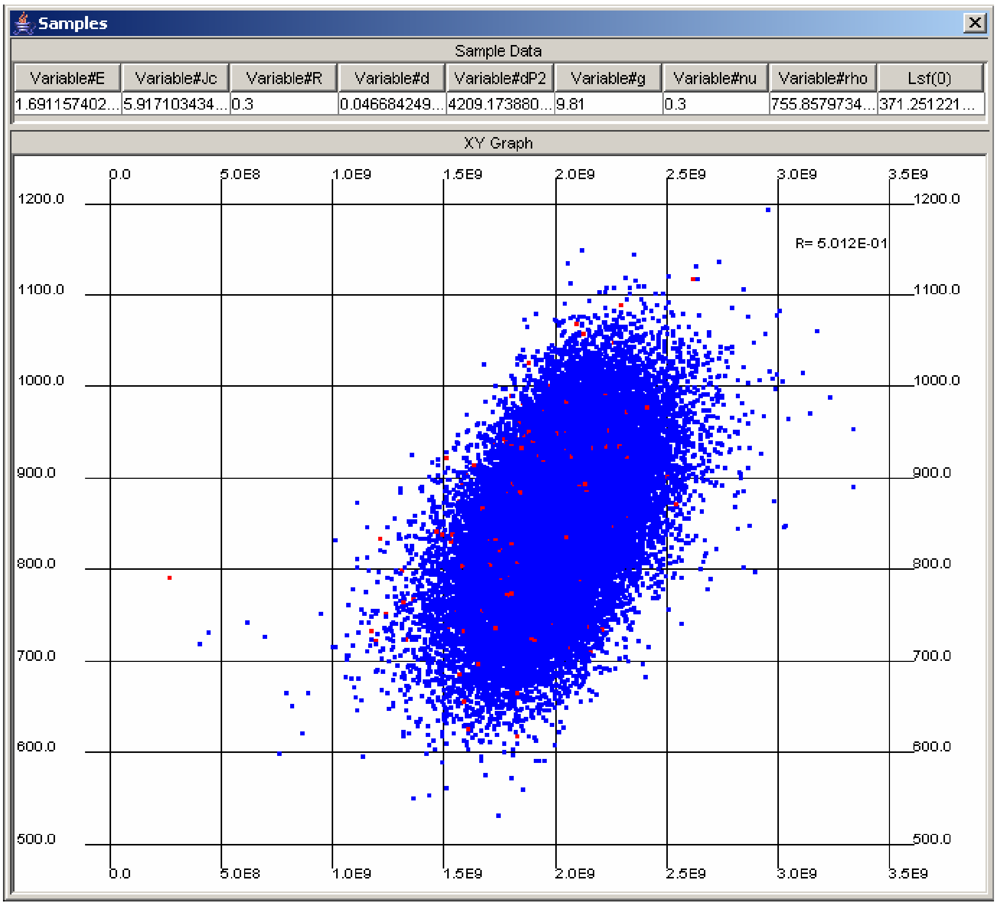
Figure 43 also demonstrates another feature: by left-mouse, double-clicking on a data point in the graphic pane, the corresponding row in the table pane will automatically be highlighted. In Figure 43, the most right data point in the graph was selected.



☑ Grid
☑ Points
☑ Colour
☑ two Colours
☑ Swap
☐ Normalised
☑ Correlation
Dotsize 3

Figure 44 Selection of options to obtain Figure 45

Checking the correlation to ‘on’, see Figure 44, will calculate the correlation value for the parameters on the X- and Y-axis and show the result in the upper right corner of the displayed graph. Figure 45 shows the result of this action when this is done with the variables ‘E’ and ‘rho’ on the two axes.



Variable#E	Variable#Jc	Variable#R	Variable#d	Variable#dP2	Variable#g	Variable#nu	Variable#rho	Lsf(0)
1.691157402...	5.917103434...	0.3	0.046684249...	4209.173880...	9.81	0.3	755.8579734...	371.251221...

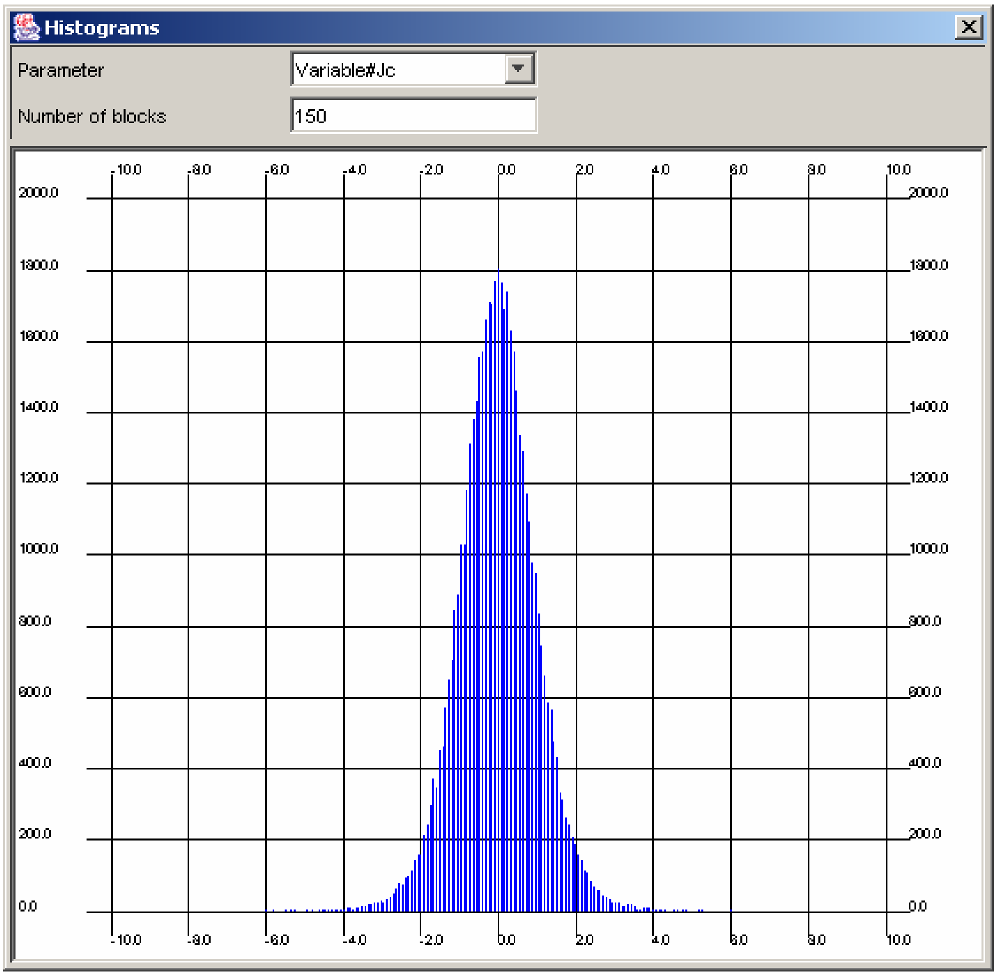
R= 5.012E-01

Figure 45 Presentation including correlation value

A value of 0.4965 is calculated from the sample data, which coincides well with the given input value of 0.5 in the correlation matrix.

4.14.2 Histograms

When selecting the ‘Histograms’ button from the main Graphs window (Figure 37), a graphic pane will appear. The user can select the stochastic input parameter or the limit state function he wants to present in a histogram, as well as the number of bars with which the graph is constructed. Figure 46 shows such a histogram for values of the variable ‘Jc’.



Parameter Variable#Jc

Number of blocks 150

Figure 46 Histogram for Jc values based on 10000 Monte Carlo calculations

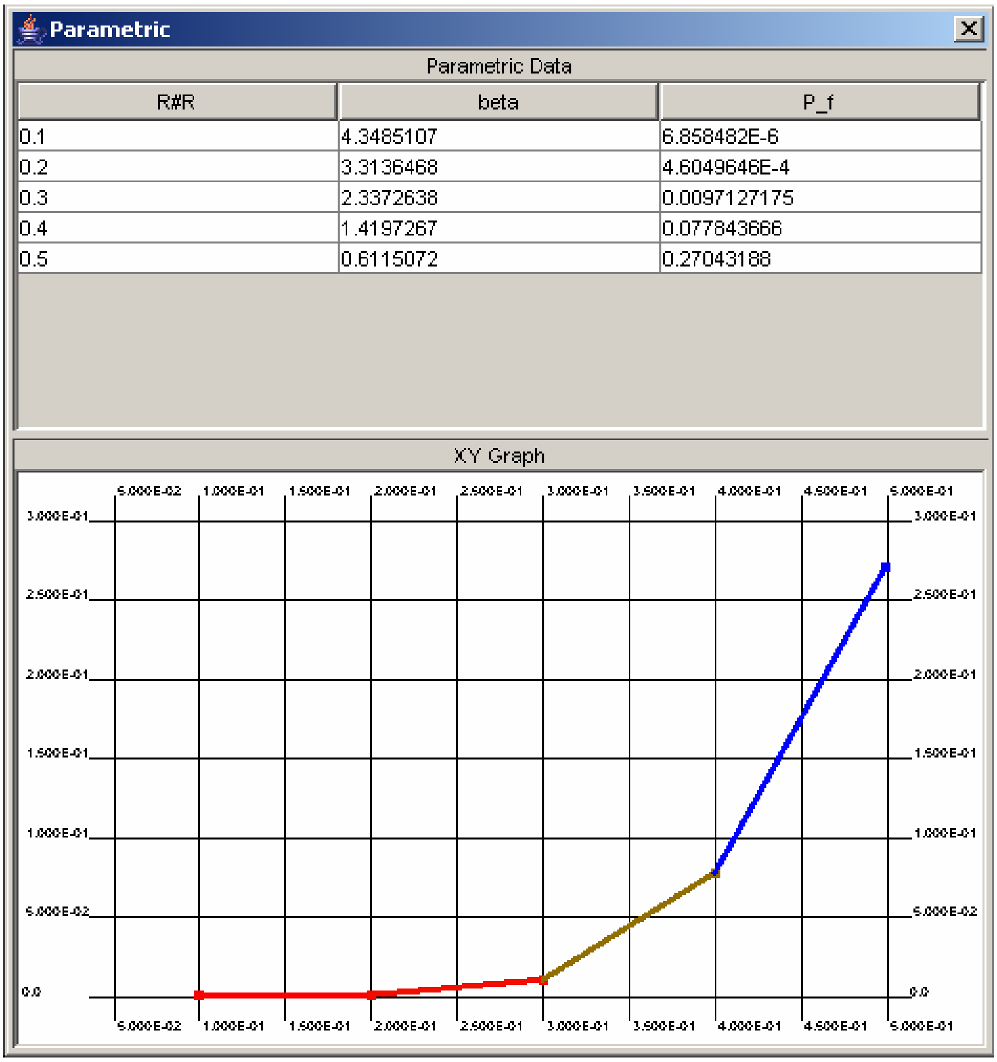
4.14.3 Parametric

In order to give an example for the graphs for parametric calculations, the FORM results with parametric values for ‘R’ are recalculated, see Figure 35.

As a result also the third option ‘Parametric’ in the main graph window Figure 37, will be enabled.

Selecting this button for this example shows the probability of failure as a function of the parametric values.

Comparable ‘Axis Dialog’ and ‘Display Options’ windows can be obtained, as was the case for displaying the samples (section 4.14.1). Again, by right mouse clicks in either the table or the graphic pane.



R#R	beta	P_f
0.1	4.3485107	6.858482E-6
0.2	3.3136468	4.6049646E-4
0.3	2.3372638	0.0097127175
0.4	1.4197267	0.077843666
0.5	0.6115072	0.27043188

Figure 47 Probability of failure as function of parametric values for R.

4.15 Loading an existing project

After starting Prob2B one can load a saved session and continue input or do recalculations.
During a session one can also load another project file. The existing session will then be overruled by the loaded settings. Hence, the user will be prompted first to save the current settings.

To activate this option the user can click the ‘open’ button 📂 or select the corresponding item under the ‘File’ menu.

Loading a file with saved results, will only load the project settings **not** the saved results

5 Using model expressions

The use of expressions as model type will be discussed using a second example.

5.1 Background on second example

The limit state function from the previous sections is modified into the following:

$$Z = C_f \exp\left(J_c\right) - \frac{3(1-\nu^2)}{32Ed^3} R^4 \left(\Delta p_F\right)^2 \tag{2}$$

with

$$\Delta p_F = \frac{1}{2}\left(-p_0 + \Delta p_2 - \frac{V_0}{k} + \sqrt{(p_0 - \Delta p_2 + \frac{V_0}{k})^2 + 4\frac{V_0}{k}\Delta p_2} \right) \tag{3}$$

in which

$$\frac{V_0}{k} = \frac{16Ed^3}{(1-\nu^2)}\left(\frac{\rho g(1-\nu^2)}{16Ed^2} + \frac{V}{\pi R^6} \right) \tag{4}$$

The resulting variables are listed in Table 11.

Table 11 Input of variables for second example

Variable	**Symbol**	**Unit**	**Mean**	**St. dev**	**Distribution**
Poisson’s ratio	ν	-	0,3	-	Deterministic
Radius of defect	R	m	0.3	-	Deterministic
Volume of hole	V	m^3	0.004	-	Deterministic
pressure	p0	N/m2	10^5	-	Deterministic
gravity	g	m/s2	9.81	-	Deterministic
Density	ρ	kg/m^3	850	75	Normal
Fatigue factor	C_f	-	0.5	0.15	Normal
Change in pressure	Δp_2	N/m2	$4.0 \cdot 10^3$	$1.0 \cdot 10^3$	Normal
Variable	**Symbol**	**Unit**	**M**	**sigma**	**Distribution**
Thickness of layer	d	m	0.046	0.0052	Student (23)
Crack propagate energy	J_c	J/m^2	0,0092	0.88	Student (10)
Young’s modulus	E	Pa	$2.0 \cdot 10^9$	$0.18 \cdot 10^9$	Student (8)

The variables ρ and E will now be taken uncorrelated.

Equations (2) to (4) form a nested set of equations. First the result of equation 4 has to be calculated and substituted in equation (3). Finally the result of equation (3) is used in the limit state function (2).

Prob2B is able to model this situation by means of two features:

1) equations can be defined as models
2) models can be but in a sequence, in which model outputs can be used as input for other models.

The situation at hand is schematically depicted in Figure 48.

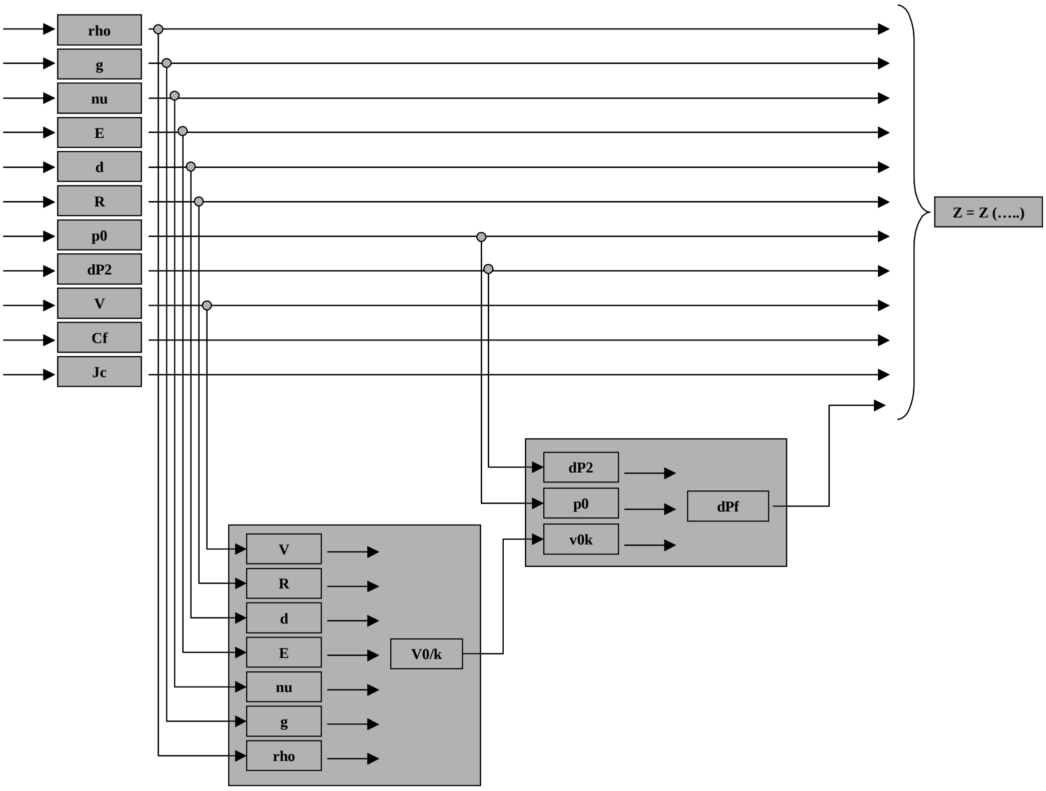


Figure 48 Variables and expressions (as models) and limitstate function.

In Prob2B, distinct equations can be handled as separate models called expressions. These models consist of a set of input variables, a definition for the equation and a definition for the resulting output variable. The input variables and their names are to be defined before hand by defining variables, previous expressions or external models. Once the necessary set of variables is available, an expression can be built as a new model. Names for the expression output variables are defined within the model. The name of the model will be equal to the name of the output variable.
Equation (3), for example, could be defined with local variables ‘dP2’, ‘p0’, ‘Vok’ as input and ‘dPf’ as output. The model name would then automatically be ‘dPf’.

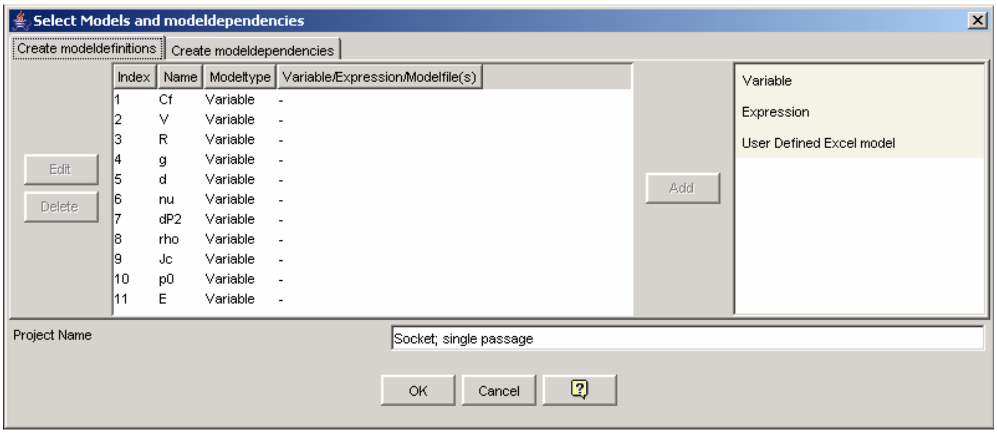
Defining the above example in Prob2B would imply the following actions:

- Create a project.
- Define the variables.
- Define the two models for equation (3) and (4)
- Define the limit state function.
- (Re-)set the stochastic properties and set the correlations.
- Set the calculation method.
- Select the output desired.
- Run the calculations.
- Look at results.
- Save project and results.

As far as these steps differ from the ones already discussed, they will be described in the following subsections. Prob2B will check the use of variables and expressions substituted in other models and expressions on consistency. When consistency is found, Prob2B will automatically put the different models in a correct calculation order.

5.2 Defining equations as models

We follow the steps from paragraphs 4.2 and 4.3 for creating a project and defining the variables. As a result we have the following models available:

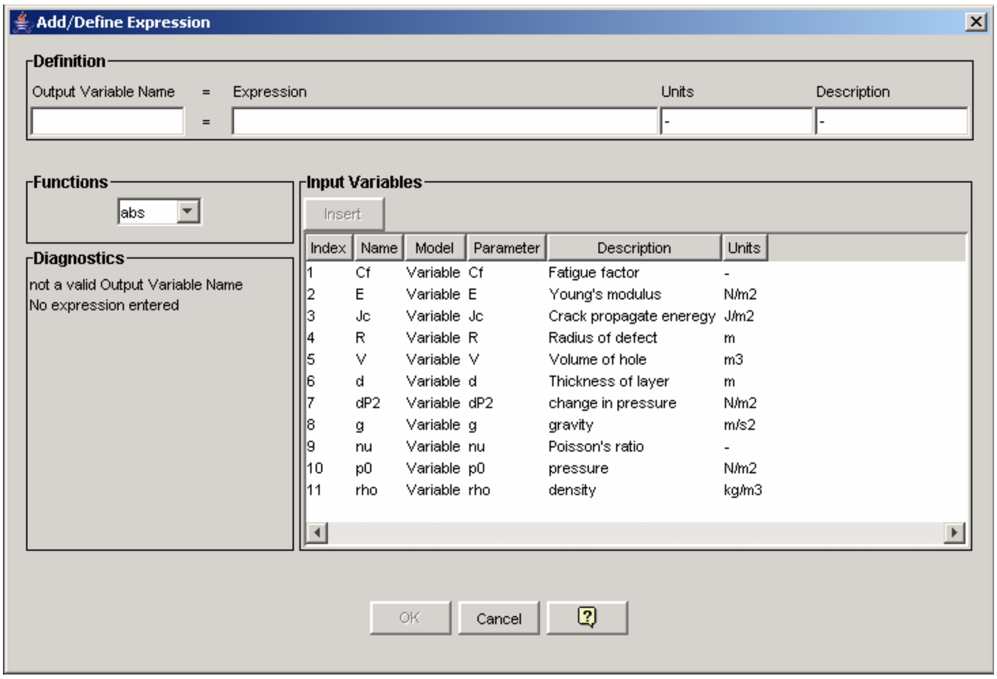


Index	Name	Modeltype	Variable/Expression/Modelfile(s)
1	Cf	Variable	-
2	V	Variable	-
3	R	Variable	-
4	g	Variable	-
5	d	Variable	-
6	nu	Variable	-
7	dP2	Variable	-
8	rho	Variable	-
9	Jc	Variable	-
10	p0	Variable	-
11	E	Variable	-

Project Name: Socket; single passage

Figure 49 List of parameters of model type ‘Variable’ for modified limit state.

Next two models have to be added of type ‘expression’. Selecting ‘Expression’ in the list on the right hand side and clicking on Add (or double click on ‘Expression’), results in the input pane as shown in Figure 50.



Units: -

Description: -

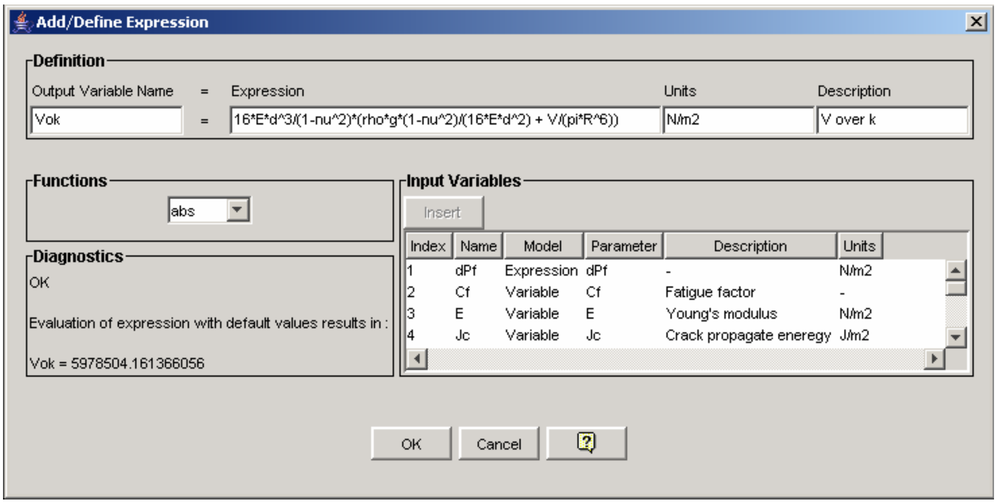
not a valid Output Variable Name
No expression entered

Index	Name	Model	Parameter	Description	Units
1	Cf	Variable	Cf	Fatigue factor	-
2	E	Variable	E	Young's modulus	N/m2
3	Jc	Variable	Jc	Crack propagate eneregy	J/m2
4	R	Variable	R	Radius of defect	m
5	V	Variable	V	Volume of hole	m3
6	d	Variable	d	Thickness of layer	m
7	dP2	Variable	dP2	change in pressure	N/m2
8	g	Variable	g	gravity	m/s2
9	nu	Variable	nu	Poisson's ratio	-
10	p0	Variable	p0	pressure	N/m2
11	rho	Variable	rho	density	kg/m3

Figure 50 Input window for adding an expression as model.

At the top this window contains input fields for the model name (= output variable name), the expression, the units and the description.
Below that, a selection box is present with predefined functions and a table for the input variables. These input variables are global variables and already exist.
The input variables can be selected and inserted in the expression using the insert button.
At the left of the window is a message box in which diagnostic messages will be echoed concerning the completeness and correctness of the input.

A valid input for equation (4) would for instance look like Figure 51.
Pressing OK returns to the model input window.
Next equation (3) can be added the input window of which is shown in Figure 52. Equation (3) can only be defined after equation (4) is defined, as we need the variable ‘Vok’ to be available for equation (3).
After returning to the model input window we see the list of available models, i.e. two models of type ‘expression’, and eleven models of type ‘variable’ (Figure 53).



Output Variable Name	=	Expression	Units	Description
Vok	=	16*E*d^3/(1-nu^2)*(rho*g*(1-nu^2)/(16*E*d^2) + V/(pi*R^6))	N/m2	V over k

Diagnostics

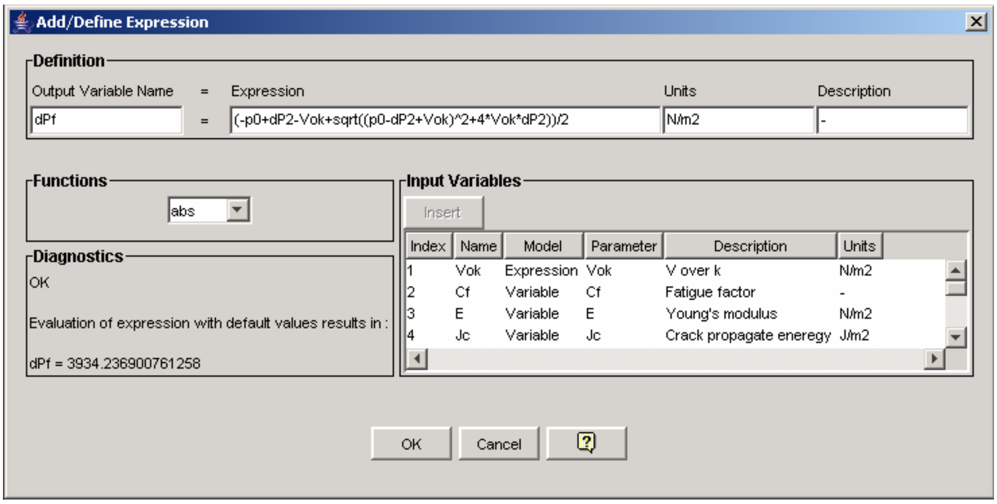
OK

Evaluation of expression with default values results in :

Vok = 5978504.161366056

Index	Name	Model	Parameter	Description	Units
1	dPf	Expression	dPf	-	N/m2
2	Cf	Variable	Cf	Fatigue factor	-
3	E	Variable	E	Young's modulus	N/m2
4	Jc	Variable	Jc	Crack propagate eneregy	J/m2

Figure 51 Input window for adding expression (4).



Output Variable Name	=	Expression	Units	Description
dPf	=	(-p0+dP2-Vok+sqrt((p0-dP2+Vok)^2+4*Vok*dP2))/2	N/m2	-

Diagnostics

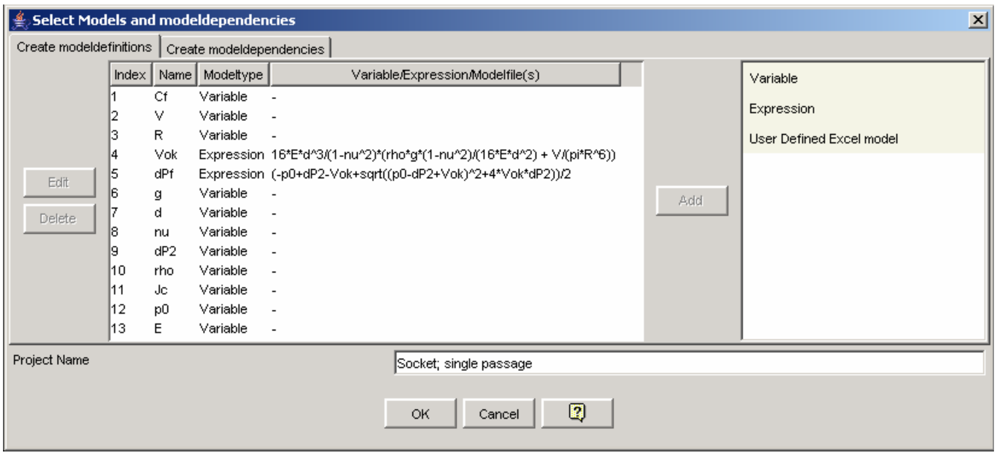
OK

Evaluation of expression with default values results in :

dPf = 3934.236900761258

Index	Name	Model	Parameter	Description	Units
1	Vok	Expression	Vok	V over k	N/m2
2	Cf	Variable	Cf	Fatigue factor	-
3	E	Variable	E	Young's modulus	N/m2
4	Jc	Variable	Jc	Crack propagate eneregy	J/m2

Figure 52 Input window for adding expression (3).



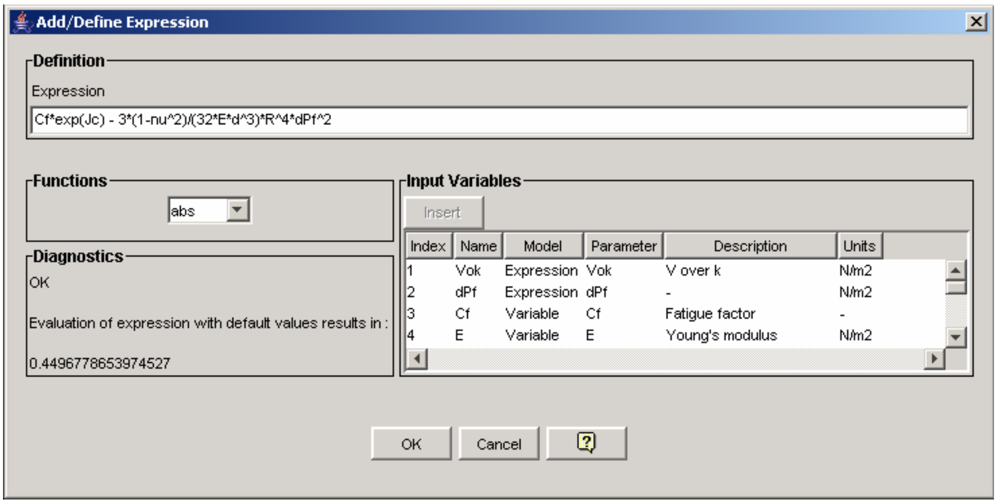
Index	Name	Modeltype	Variable/Expression/Modelfile(s)
1	Cf	Variable	-
2	V	Variable	-
3	R	Variable	-
4	Vok	Expression	16*E*d^3/(1-nu^2)*(rho*g*(1-nu^2)/(16*E*d^2) + V/(pi*R^6))
5	dPf	Expression	(-p0+dP2-Vok+sqrt((p0-dP2+Vok)^2+4*Vok*dP2))/2
6	g	Variable	-
7	d	Variable	-
8	nu	Variable	-
9	dP2	Variable	-
10	rho	Variable	-
11	Jc	Variable	-
12	p0	Variable	-
13	E	Variable	-

Project Name: Socket; single passage

Figure 53 Listing of models of type ‘variable’ and ‘expression’.

5.3 Completing the input for the second example and calculation results

We can now go into the Limit state menu and define the limit state according to Figure 54



Expression

Cf*exp(Jc) - 3*(1-nu^2)/(32*E*d^3)*R^4*dPf^2

Diagnostics

OK

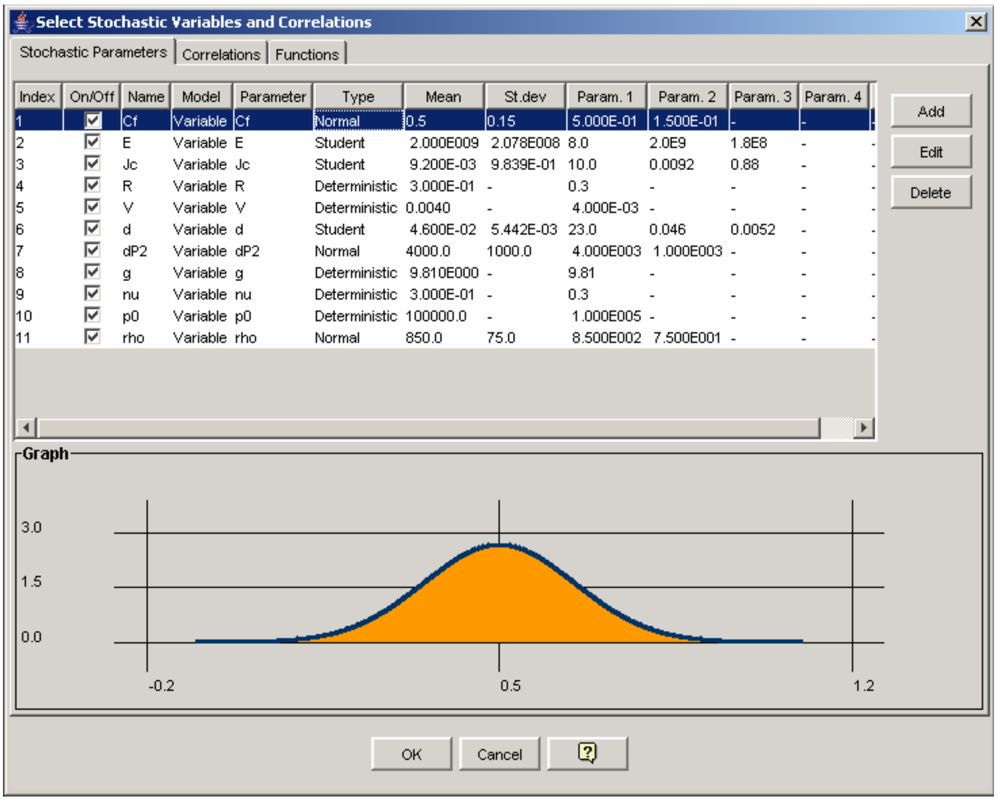
Evaluation of expression with default values results in :

0.4496778653974527

Index	Name	Model	Parameter	Description	Units
1	Vok	Expression	Vok	V over k	N/m2
2	dPf	Expression	dPf	-	N/m2
3	Cf	Variable	Cf	Fatigue factor	-
4	E	Variable	E	Young's modulus	N/m2

Figure 54 Limit state function for second example

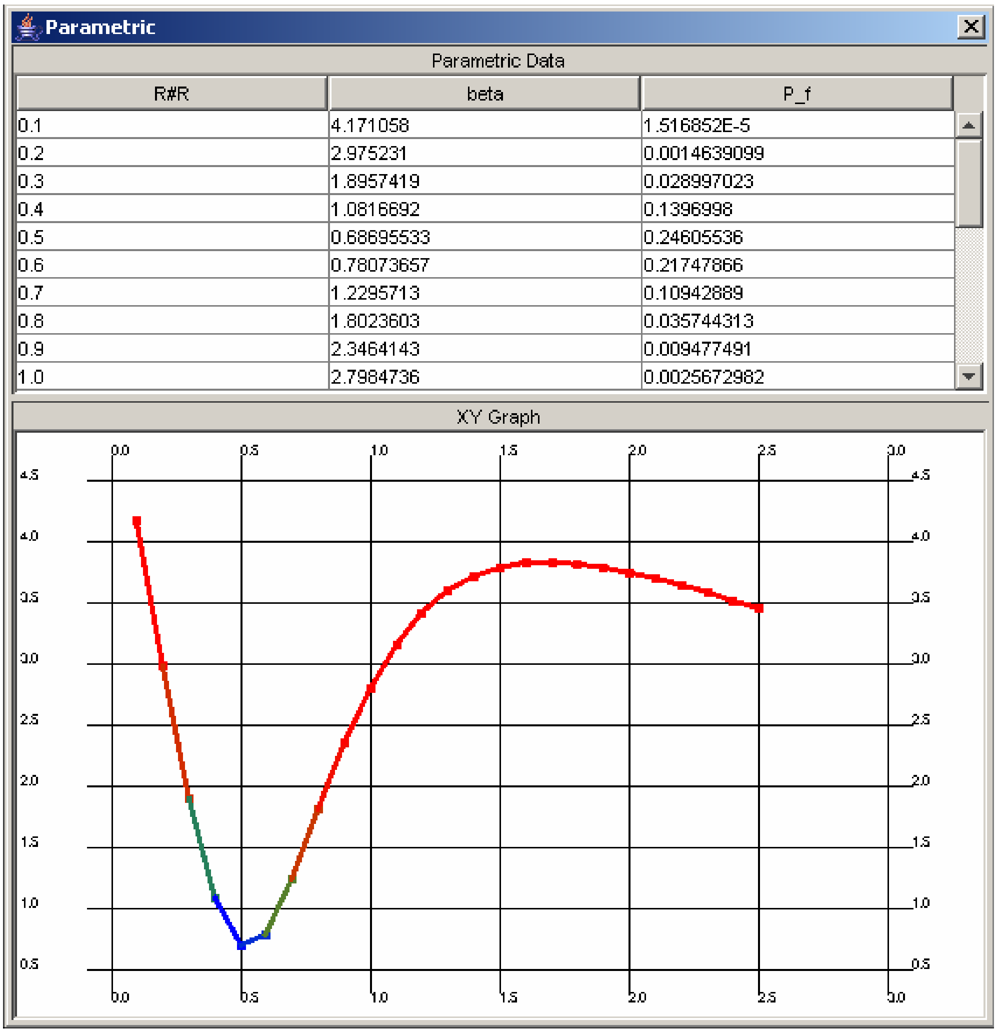
The stochastic properties are listed in the stochastic menu and shown in Figure 55. No correlations are taken into account.



Index	On/Off	Name	Model	Parameter	Type	Mean	St.dev	Param. 1	Param. 2	Param. 3	Param. 4
1	☑	Cf	Variable	Cf	Normal	0.5	0.15	5.000E-01	1.500E-01	-	-
2	☑	E	Variable	E	Student	2.000E009	2.078E008	8.0	2.0E9	1.8E8	-
3	☑	Jc	Variable	Jc	Student	9.200E-03	9.839E-01	10.0	0.0092	0.88	-
4	☑	R	Variable	R	Deterministic	3.000E-01	-	0.3	-	-	-
5	☑	V	Variable	V	Deterministic	0.0040	-	4.000E-03	-	-	-
6	☑	d	Variable	d	Student	4.600E-02	5.442E-03	23.0	0.046	0.0052	-
7	☑	dP2	Variable	dP2	Normal	4000.0	1000.0	4.000E003	1.000E003	-	-
8	☑	g	Variable	g	Deterministic	9.810E000	-	9.81	-	-	-
9	☑	nu	Variable	nu	Deterministic	3.000E-01	-	0.3	-	-	-
10	☑	p0	Variable	p0	Deterministic	100000.0	-	1.000E005	-	-	-
11	☑	rho	Variable	rho	Normal	850.0	75.0	8.500E002	7.500E001	-	-

Figure 55 Stochastic properties for second example

We overrule the properties for ‘R’ in setting them parametric in the range from 0.1 to 2.5 m with steps of 0.1 m. Calculations are performed using FORM and calculation results for β as a function of ‘R’ are presented in Figure 56.



R#R	beta	P_f
0.1	4.171058	1.516852E-5
0.2	2.975231	0.0014639099
0.3	1.8957419	0.028997023
0.4	1.0816692	0.1396998
0.5	0.68695533	0.24605536
0.6	0.78073657	0.21747866
0.7	1.2295713	0.10942889
0.8	1.8023603	0.035744313
0.9	2.3464143	0.009477491
1.0	2.7984736	0.0025672982

Figure 56 Graphical results for second example.

6 Use of external models

The use of loading an external model into Prob2B will be discussed using a third example. This external model involves an Excel® model. The description on how to make Excel® accessible for Prob2B can be found in appendix D. Please, take note of section D.2.4 and section D.3 before proceeding with the example in this chapter.

6.1 Background on third example

In this example we elaborate on the example in chapter 5, by assuming an Excel® model being present that describes the equation for Δp_F:

$$\Delta p_F = \frac{1}{2}\left(-p_0 + \Delta p_2 - Vok + \sqrt{(p_0 - \Delta p_2 + Vok)^2 + 4(Vok)\Delta p_2} \right) \tag{5}$$

The resulting variables are listed in Table 12.

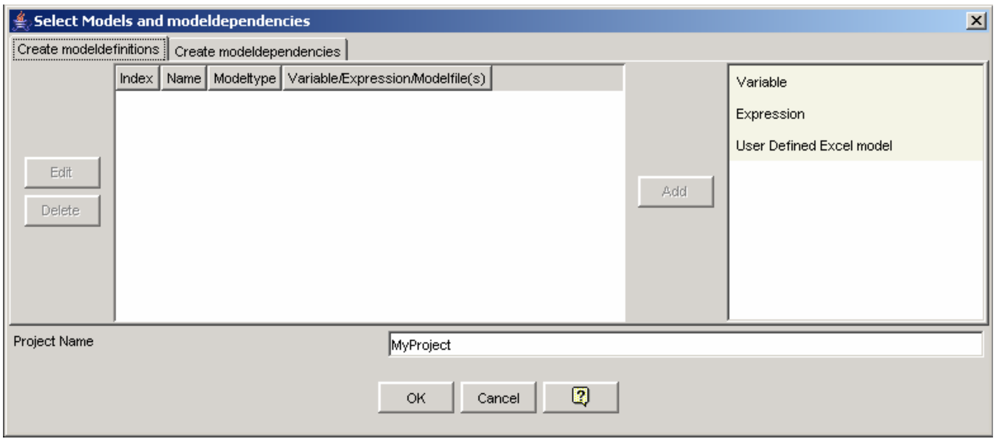
Table 12 Input of variables for third example

Variable	**Symbol**	**Unit**	**Mean**	**St. dev**	**Distribution**
V over k	Vok	N/m2	$6.3 \cdot 10^6$	$2.3 \cdot 10^6$	Lognormal
pressure	p0	N/m2	10^5	-	Deterministic
Change in pressure	Δp_2	N/m2	$4.0 \cdot 10^3$	$1.0 \cdot 10^3$	Normal

The descriptions in the following paragraphs start after having created a new project.

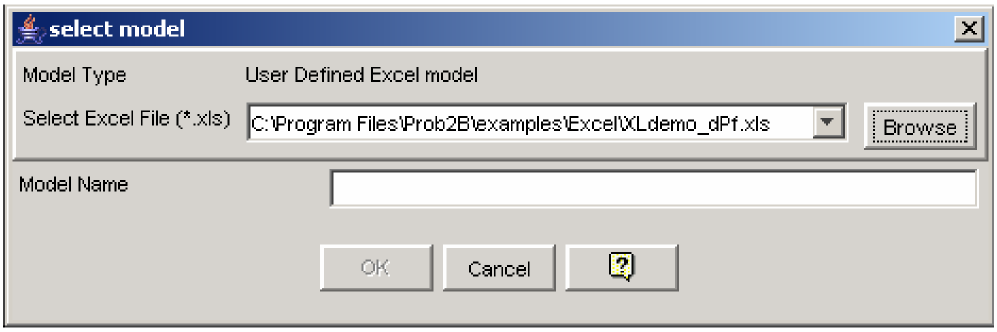
6.2 Loading Excel® as an external model

We go into the model definition menu, Figure 57, and select ‘User Defined Excel® model’ in the list at the right hand side. Next, clicking on ‘Add’, brings us into a new input menu in which we are asked for the Excel® document. With the browser one can select the appropriate document on from disk, see Figure 58.
Finally, as in Figure 59, we are asked for a model name that will be used within Prob2B for referencing to the Excel®-model.



Project Name MyProject

Figure 57 Model input menu

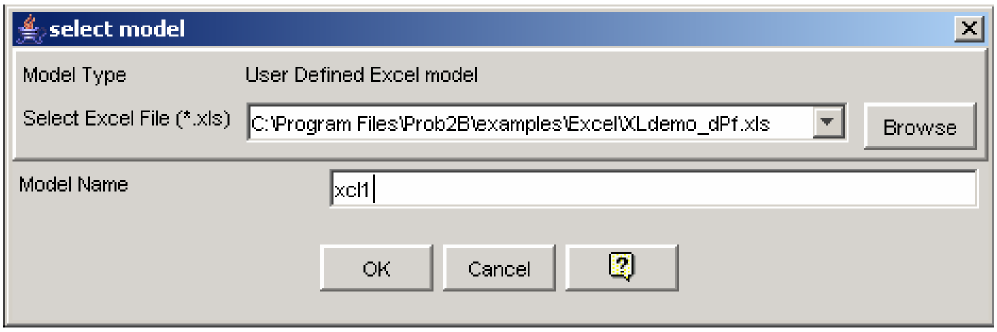


Model Type User Defined Excel model

Select Excel File (*.xls) C:\Program Files\Prob2B\examples\Excel\XLdemo_dPf.xls

Model Name

Figure 58 Selecting external model, model browser



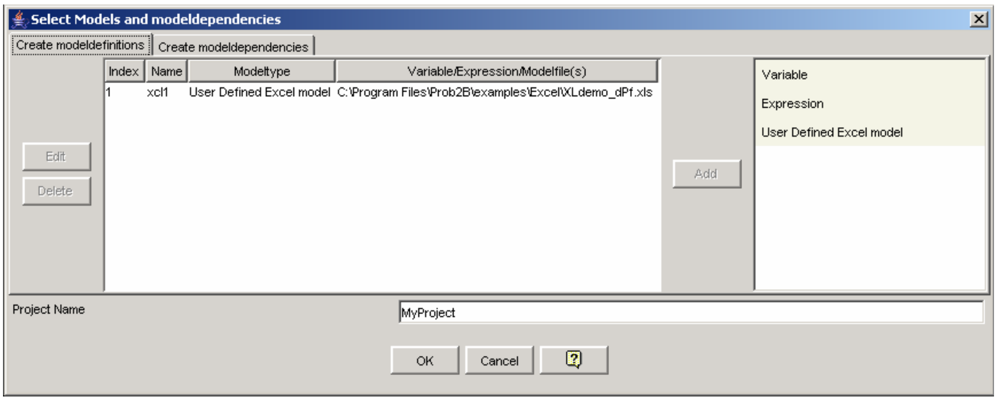
Model Type User Defined Excel model

Select Excel File (*.xls) C:\Program Files\Prob2B\examples\Excel\XLdemo_dPf.xls

Model Name xcl1

Figure 59 Selecting external model, defining model name

Pressing ‘OK’ makes Prob2B return to the model selection menu in which we now see the loaded Excel®-model listed.



Index	Name	Modeltype	Variable/Expression/Modelfile(s)
1	xcl1	User Defined Excel model	C:\Program Files\Prob2B\examples\Excel\XLdemo_dPf.xls

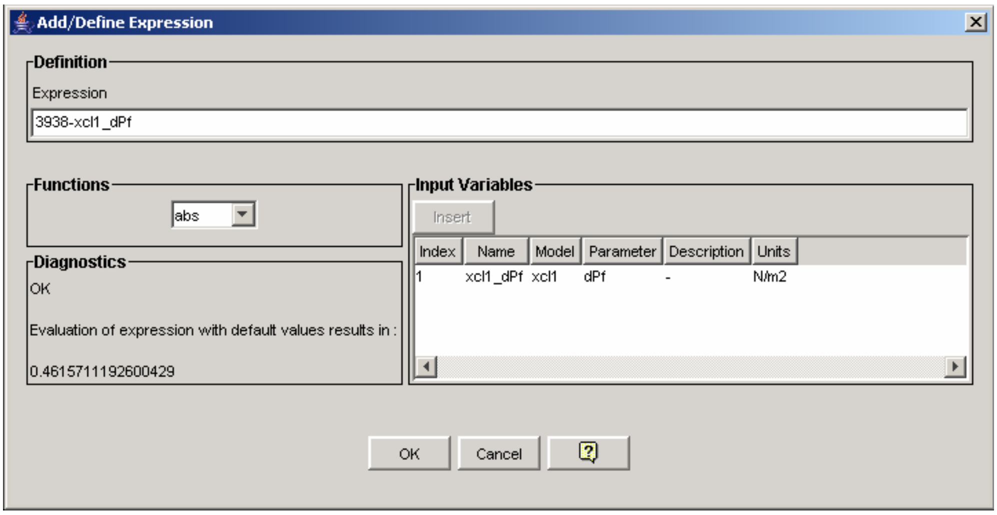
Project Name: MyProject

Figure 60 Overview of model definitions

The model is now loaded, i.e. the model as well as the variables are known to Prob2B. The latter will become clear when we are going into the menus for defining the limit state function and the menus for the stochastic properties for the variables.

6.3 Defining a limit state

Click ‘OK’ to return to the main Prob2B window and next go into the limit state menu with the button. Click the ‘Add’ button next.
We see that this menu now prompts with the output variables present in model ‘xcl1’, being just one in this case namely ‘dPf’. See Figure 61.
In this example we suffice to look at values that can be obtained for $\Delta p_{F.}$ Hence from the variables we just select ‘xcl1_dPf’. We will look at the variation of its values relative to 3938, so the function is formulated as ‘3938-dPf’.



Expression: 3938-xcl1_dPf

Index	Name	Model	Parameter	Description	Units
1	xcl1_dPf	xcl1	dPf	-	N/m2

OK

Evaluation of expression with default values results in :

0.4615711192600429

Figure 61 Creating the limit state

6.4 Defining distributions for variables in external models

Clicking ‘OK’ twice lets us return to the main Prob2B window
When going into the stochastic properties menu, button , we get an empty list as in Figure 62. This differs from paragraph 4.5, where properties were already defined for internal variables. With an external model, stochastic properties explicitly have to be added.
Click on the ‘Add’ button to get to the input menu shown in Figure 63.
We now see that in the top left the model name and the Variable name consist of combo boxes that can be used to first select the external model and next select the variable within this model.
Doing so for variable ‘Vok’ of model ‘xcl1’ allows us to define the stochastic properties for this variable, Figure 64. After finishing, we press ‘OK’ and again on ‘Add’ to proceed with the next variable. Finally, this leads to the list in Figure 65.

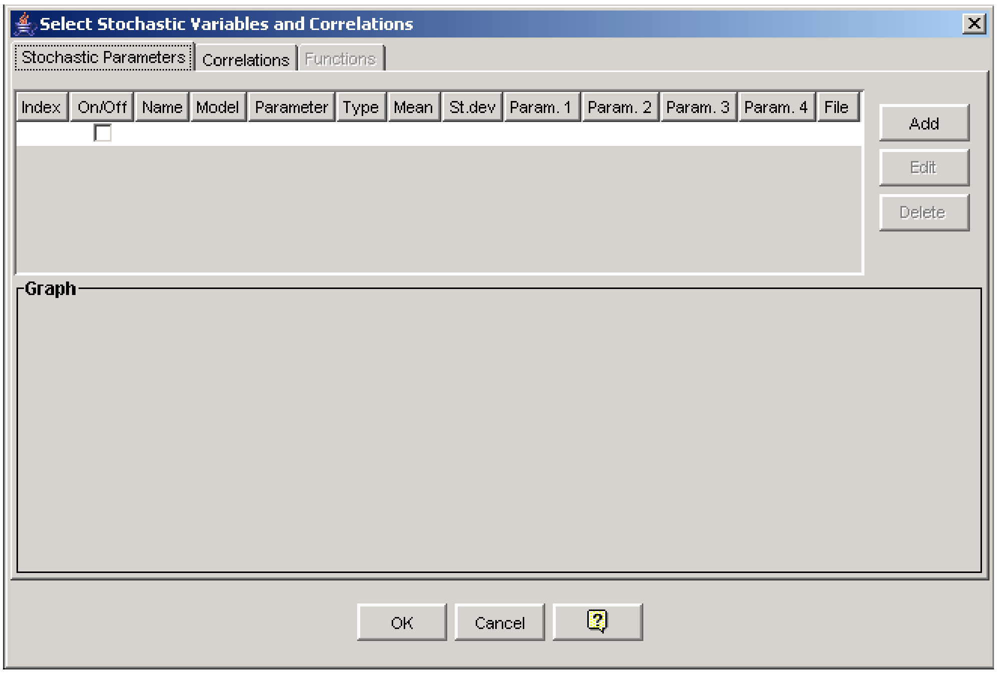
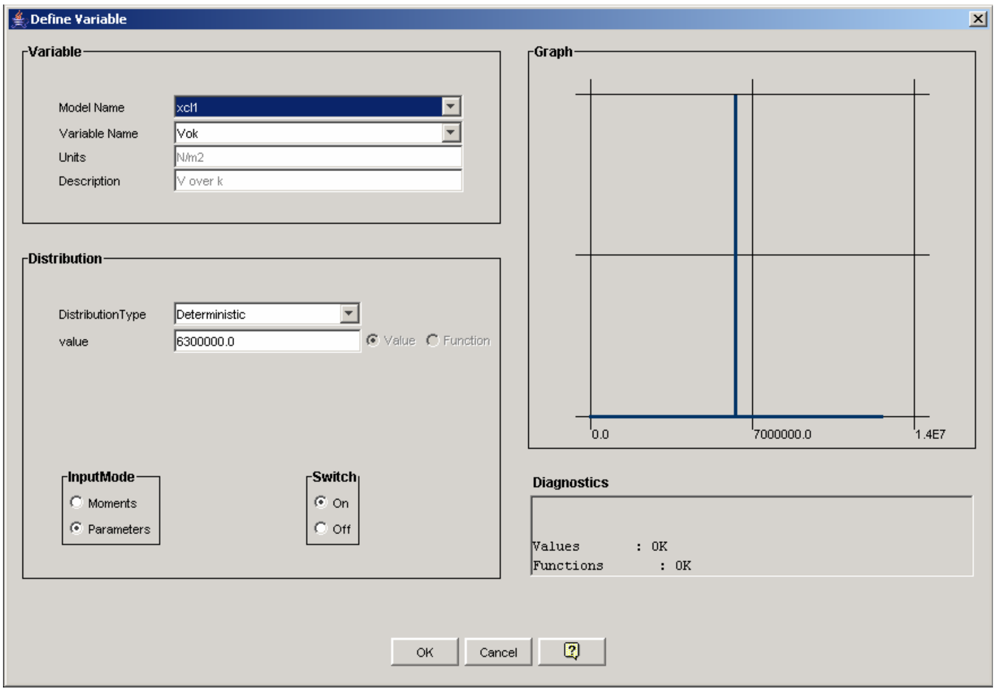


Figure 62 Stochastic properties, initial view



Model Name: xcl1

Variable Name: Vok

Units: N/m2

Description: V over k

DistributionType: Deterministic

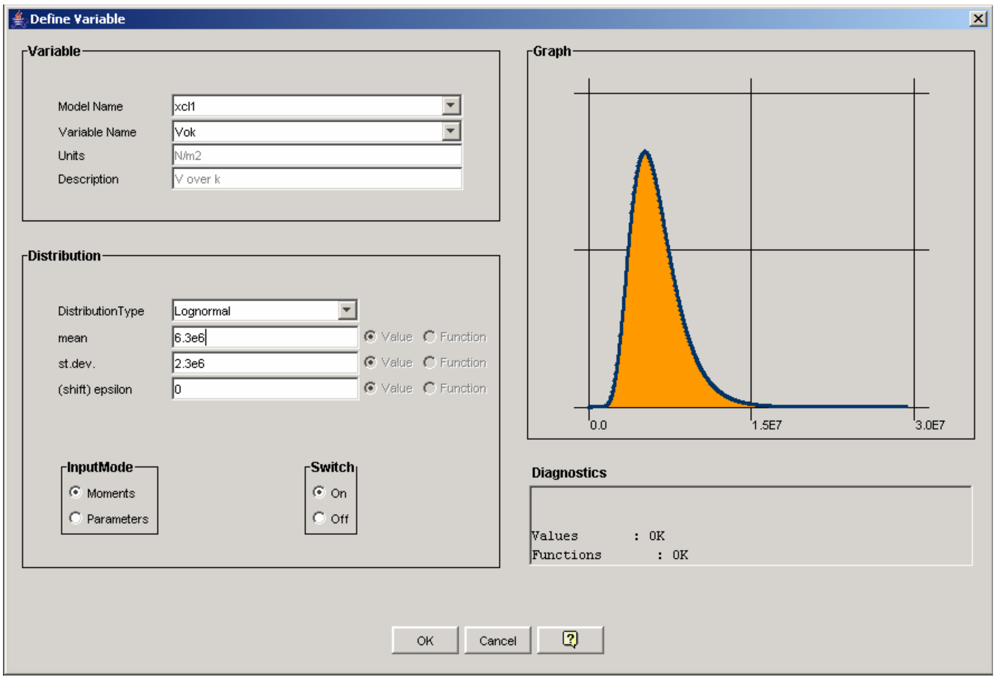
value: 6300000.0 ● Value ○ Function

InputMode: ○ Moments ● Parameters

Switch: ● On ○ Off

```
Values     : OK
Functions    : OK
```

Figure 63 Setting stochastic properties, initial view



Model Name: xcl1

Variable Name: Vok

Units: N/m2

Description: V over k

DistributionType: Lognormal

mean: 6.3e6 ● Value ○ Function

st.dev.: 2.3e6 ● Value ○ Function

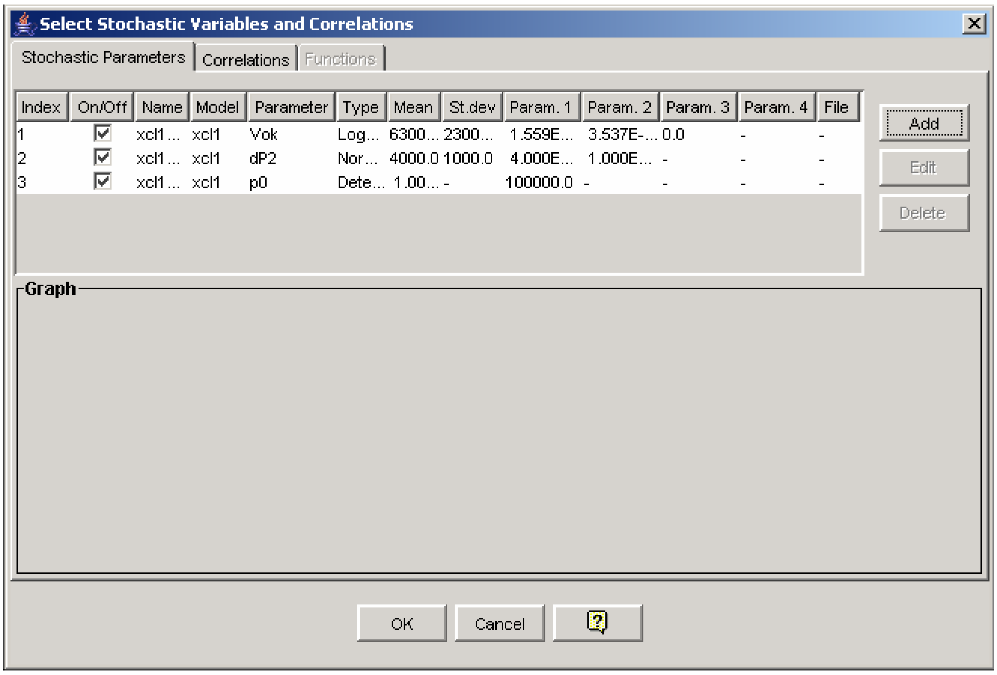
(shift) epsilon: 0 ● Value ○ Function

InputMode: ● Moments ○ Parameters

Switch: ● On ○ Off

```
Values     : OK
Functions    : OK
```

Figure 64 Setting stochastic properties, selected values



Index	On/Off	Name	Model	Parameter	Type	Mean	St.dev	Param. 1	Param. 2	Param. 3	Param. 4	File
1	☑	xcl1 ...	xcl1	Vok	Log...	6300...	2300...	1.559E...	3.537E-...	0.0	-	-
2	☑	xcl1 ...	xcl1	dP2	Nor...	4000.0	1000.0	4.000E...	1.000E...	-	-	-
3	☑	xcl1 ...	xcl1	p0	Dete...	1.00...	-	100000.0	-	-	-	-

Figure 65 Stochastic properties, final view

6.5 Example of Calculation results

We performed 100 MC calculations.
This is achieved by selecting the MC method and set the minimum and maximum number of calculations to 100.
Results are shown in Figure 66

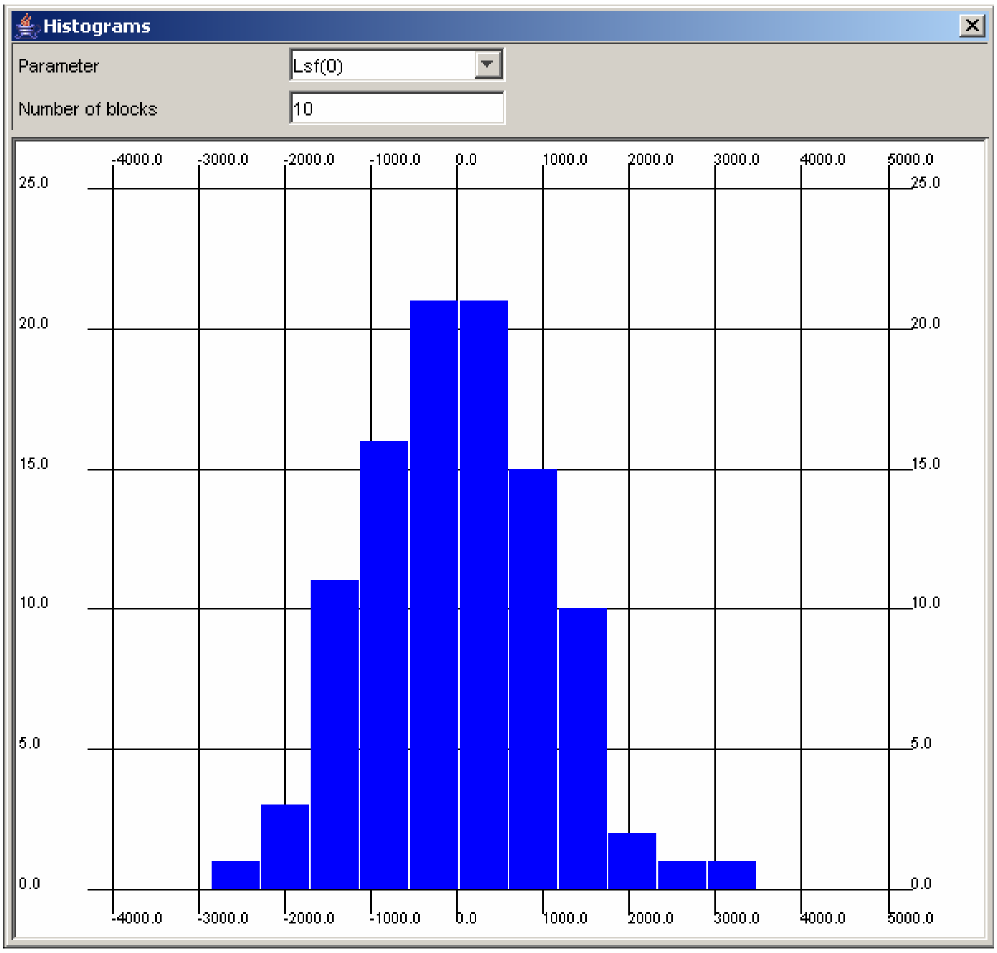


Figure 66 Values for (3938-dPf).

7 Use of dependencies

Using multiple models, e.g. variables, expressions and external models, makes the need apparent to put them in a calculation order and/or to make some models dependent upon other models. With the latter it is meant that input variables of a model become a function of output variables of other models. The use of dependencies will be discussed in this chapter using a fourth example.

7.1 Background on fourth example

The fourth example will be a combination of the Excel® model of Chapter 6, and the variables and expression models of Chapter5.
We again try to accomplish a model flow as in Figure 67. However instead of the expression for dPf we will now make us of the Excel® model.

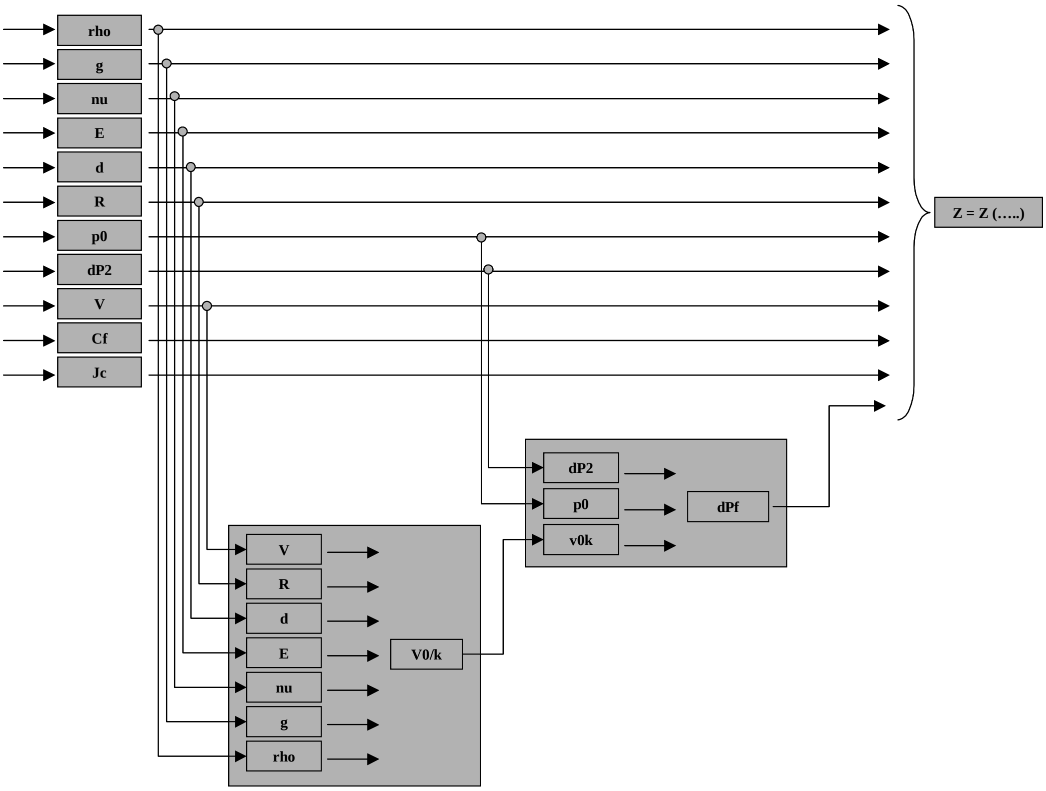
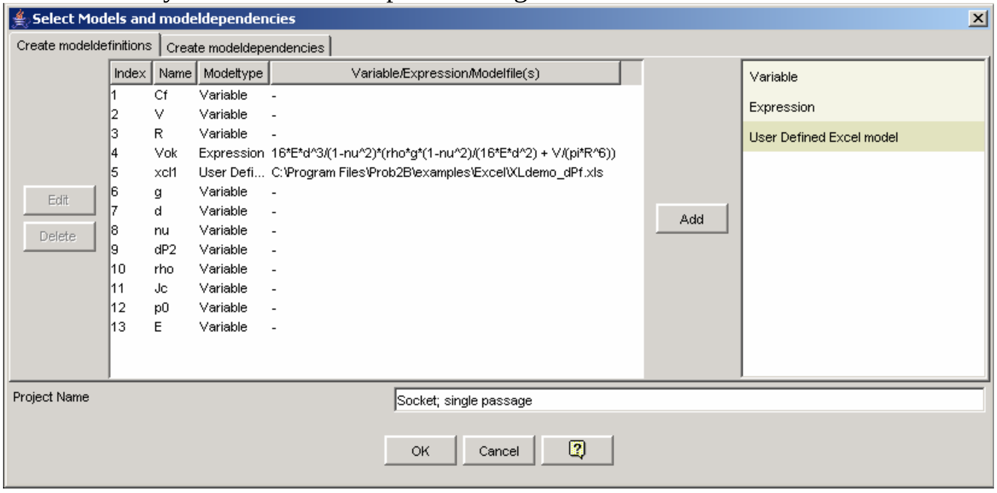


Figure 67 Models and variables

7.2 Loading the models

We start with defining the necessary variables from Chapter 5

Next we define the expression for Vok and finally we load the Excel® model for dPf. This eventually leads to the list depicted in Figure 68.



Index	Name	Modeltype	Variable/Expression/Modelfile(s)
1	Cf	Variable	-
2	V	Variable	-
3	R	Variable	-
4	Vok	Expression	16*E*d^3/(1-nu^2)*(rho*g*(1-nu^2)/(16*E*d^2) + V/(pi*R^6))
5	xcl1	User Defi...	C:\Program Files\Prob2B\examples\Excel\XLdemo_dPf.xls
6	g	Variable	-
7	d	Variable	-
8	nu	Variable	-
9	dP2	Variable	-
10	rho	Variable	-
11	Jc	Variable	-
12	p0	Variable	-
13	E	Variable	-

Project Name: Socket; single passage

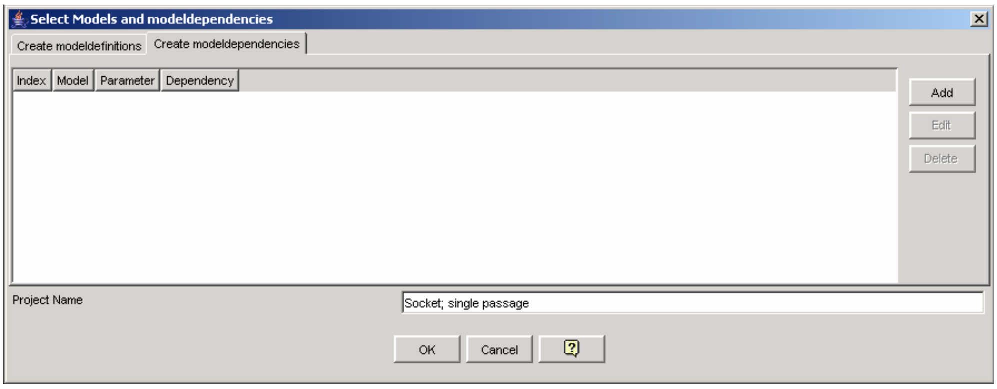
Figure 68 List of models and variables

We now have our stochastic variables ‘p0’ and ‘dP2’ and our results of the expression for ‘Vok’ that have to be used as input for the input variables of our Excel® sheet.

7.3 Creating dependencies

With the use of the model dependencies, we can put our models in a desired calculation order and make inputs of models dependent output results from other models.

From the model definition menu select the second tabbed pane: ‘Create model dependencies’. The resulting window then looks like Figure 69

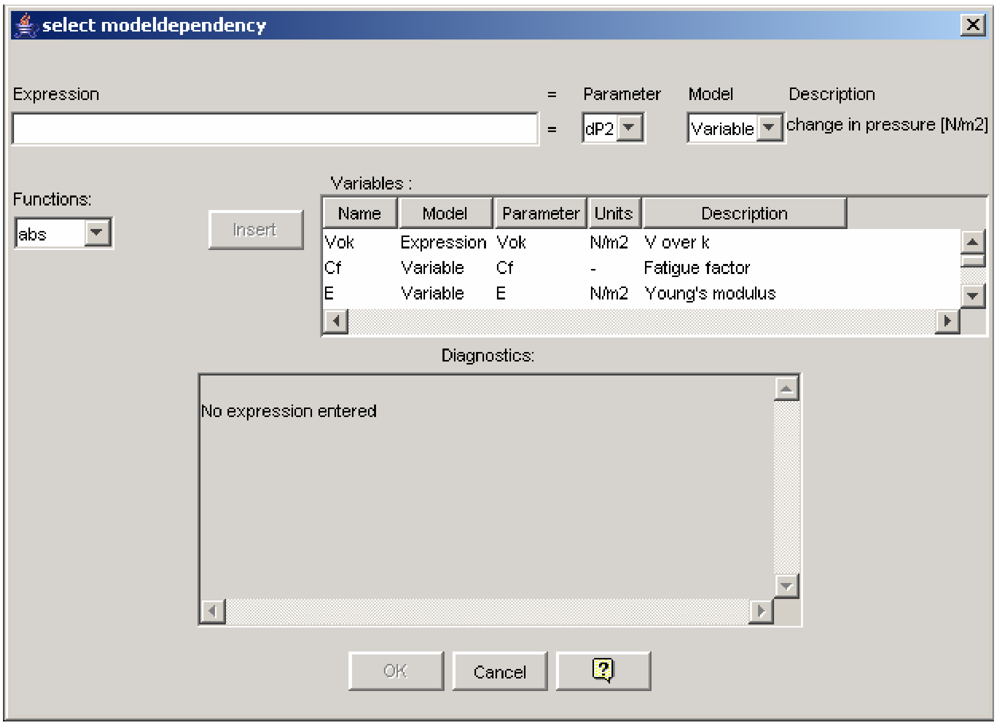


Index	Model	Parameter	Dependency

Project Name: Socket; single passage

Figure 69 Create model dependencies list, initial view.

Clicking the ‘Add’ button gives the window as shown in Figure 70. With the combo-boxes on the top right side we first select the model we want to make dependent of other models. Next the parameter is selected that is to be made dependent. Finally, the variable it is dependent upon is selected from the list. The independent variable can be used as is, or it can be combined with functions and/or other independent variables into an expression, see Figure 71 and Figure 72.



Expression:

= Parameter: dP2

Model: Variable

Description: change in pressure [N/m2]

Functions: abs

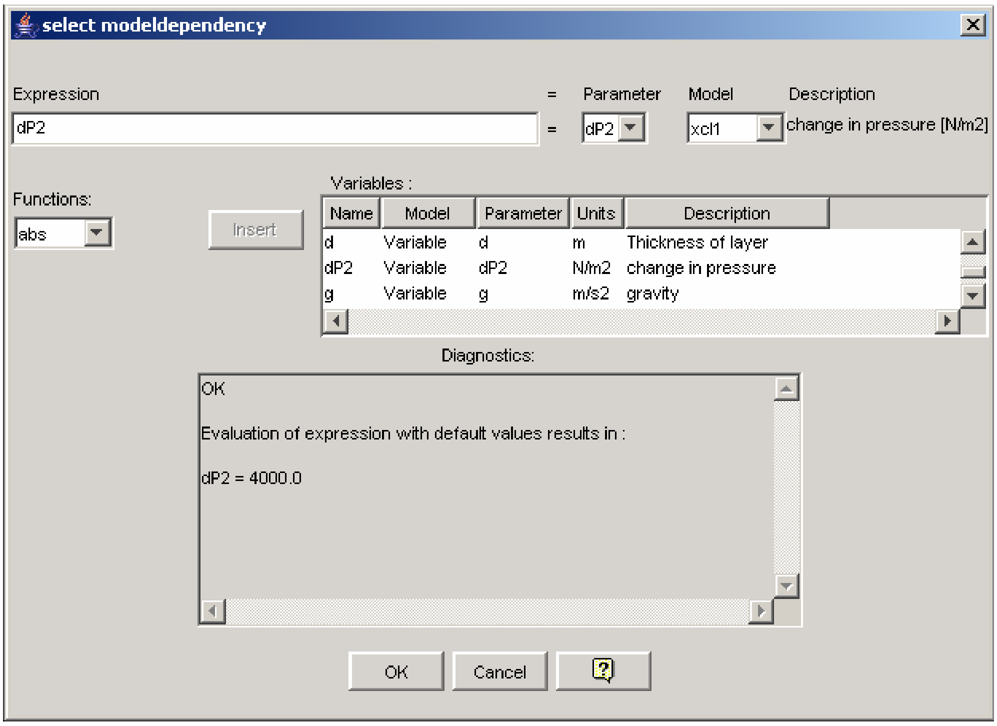
Variables :

Name	Model	Parameter	Units	Description
Vok	Expression	Vok	N/m2	V over k
Cf	Variable	Cf	-	Fatigue factor
E	Variable	E	N/m2	Young's modulus

Diagnostics:

No expression entered

Figure 70 Setting model dependencies, initial view.



Expression: dP2

= Parameter: dP2

Model: xcl1

Description: change in pressure [N/m2]

Functions: abs

Variables :

Name	Model	Parameter	Units	Description
d	Variable	d	m	Thickness of layer
dP2	Variable	dP2	N/m2	change in pressure
g	Variable	g	m/s2	gravity

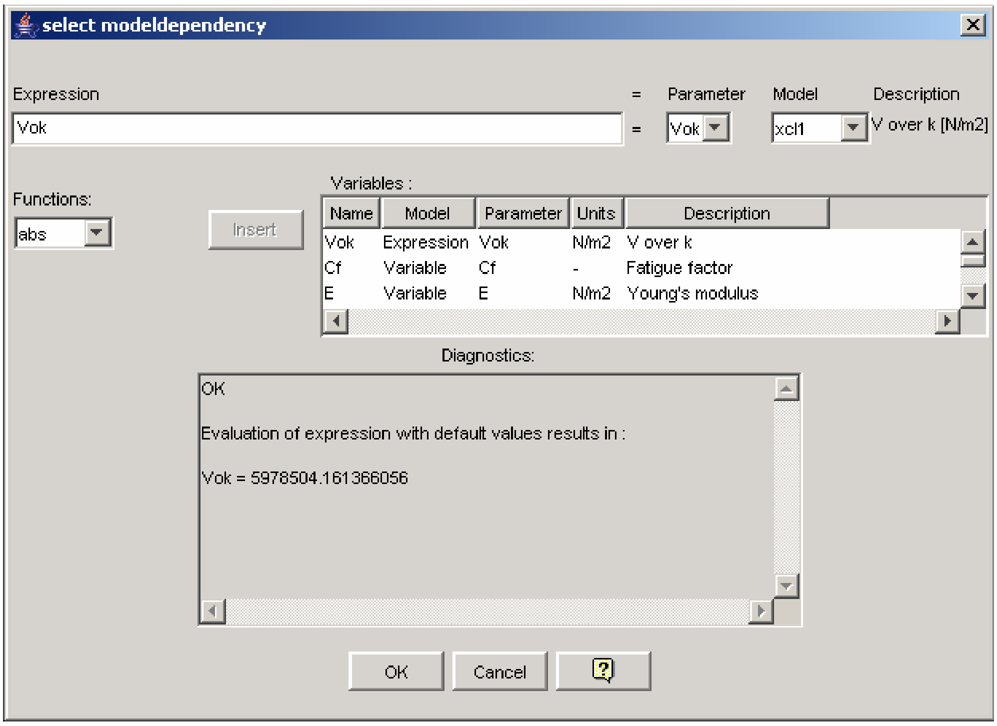
Diagnostics:

OK

Evaluation of expression with default values results in :

dP2 = 4000.0

Figure 71 Setting model dependencies for variable ‘dP2’ of model ‘xcl1’.



Expression: Vok = Parameter: Vok Model: xcl1 Description: V over k [N/m2]

Functions: abs

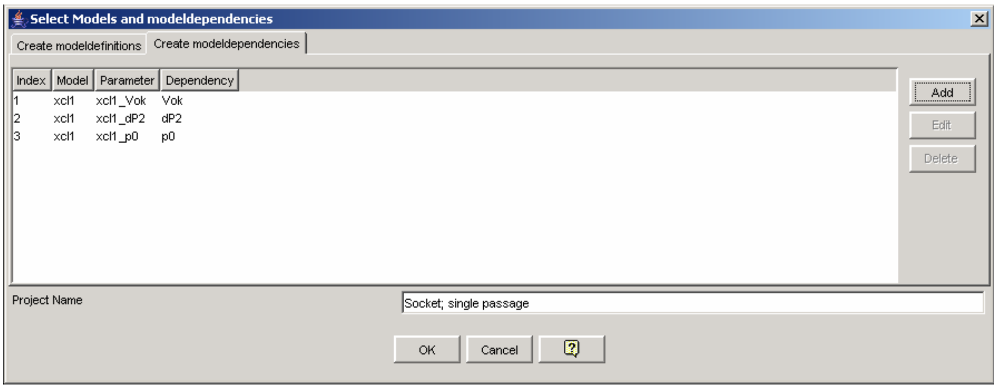
Variables :

Name	Model	Parameter	Units	Description
Vok	Expression	Vok	N/m2	V over k
Cf	Variable	Cf	-	Fatigue factor
E	Variable	E	N/m2	Young's modulus

Diagnostics:

```
OK

Evaluation of expression with default values results in :

Vok = 5978504.161366056
```

Figure 72 Setting model dependencies for variable ‘Vok’ of model ‘xcl1’..



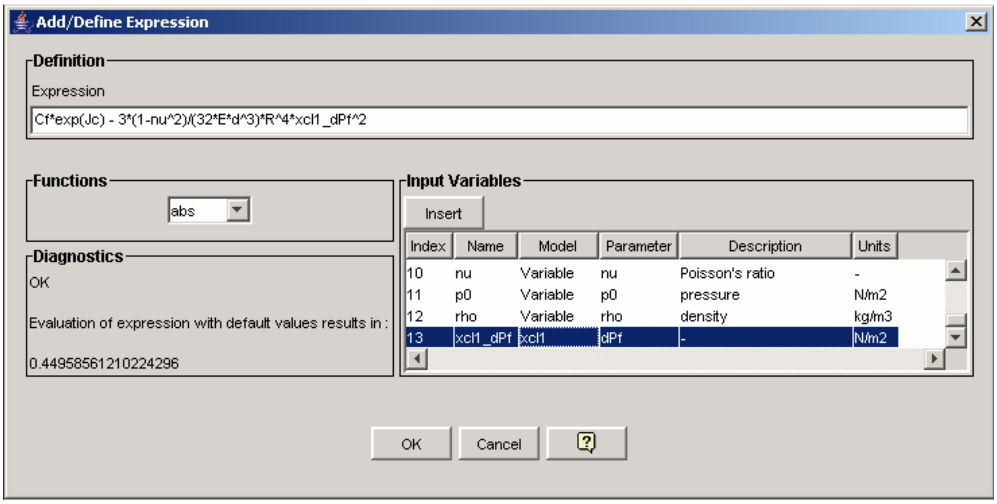
Index	Model	Parameter	Dependency
1	xcl1	xcl1_Vok	Vok
2	xcl1	xcl1_dP2	dP2
3	xcl1	xcl1_p0	p0

Project Name: Socket; single passage

Figure 73 Create model dependencies list, final view.

Having done this for all input variables of the Excel® model, we get to the list depicted in Figure 73.

7.4 Limit state definition



Definition

Expression

Cf*exp(Jc) - 3*(1-nu^2)/(32*E*d^3)*R^4*xcl1_dPf^2

Diagnostics

OK

Evaluation of expression with default values results in :

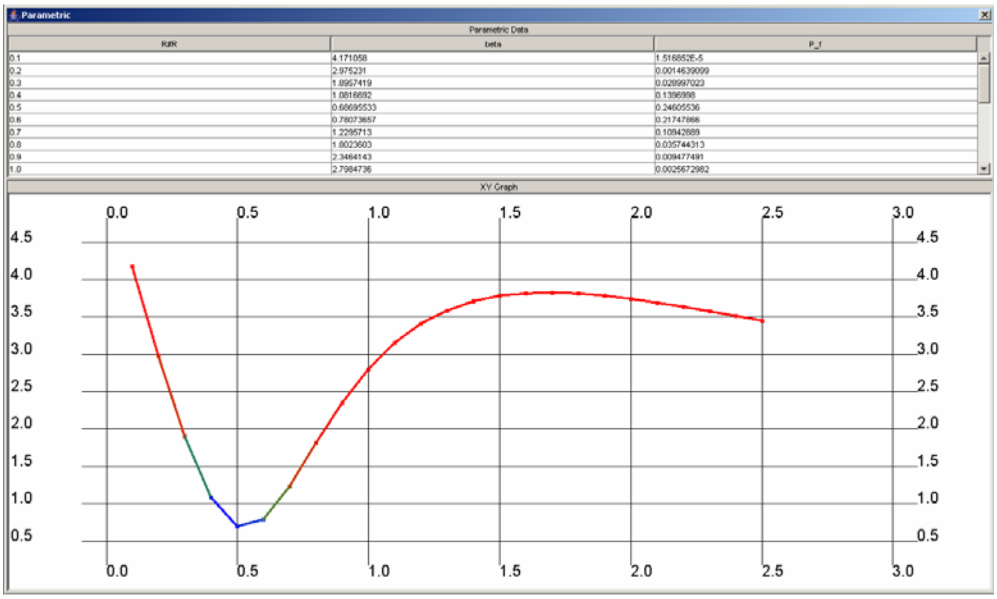
0.44958561210224296

Input Variables

Index	Name	Model	Parameter	Description	Units
10	nu	Variable	nu	Poisson's ratio	-
11	p0	Variable	p0	pressure	N/m2
12	rho	Variable	rho	density	kg/m3
13	xcl1_dPf	xcl1	dPf	-	N/m2

Figure 74 Defining limit state with variable ‘xcl1_dPf’, i.e. variable ‘dPf’ of model ‘xcl1’.

7.5 Calculation results



Parametric Data

R/R	beta	P_f
0.1	4.171058	1.516852E-5
0.2	2.975231	0.0014639099
0.3	1.8957419	0.028997023
0.4	1.0816692	0.1396998
0.5	0.68695533	0.24605536
0.6	0.78073657	0.21747866
0.7	1.2295713	0.10942889
0.8	1.8023603	0.035744313
0.9	2.3464143	0.009477491
1.0	2.7984736	0.0025672982

Figure 75 Calculation results.

7.6 Correlations and model dependencies

When a input variable for a model has stochastic properties including correlations, then these properties will be overruled when the parameter is set dependent on other parameters through a model dependency definition. In fact, same considerations apply as for parametric calculations. Reference is made to paragraph 4.7 and 4.8.

8 Using stochastic parameters as functions of variables

Not dealt with yet in the previous chapters is the option to define distribution functions whose parameters, e.g. mean or standard deviation are functions of other (stochastic) variables. This option will be discussed in the following.

8.1 Background of fifth example

We take a look at the level achieved by a water level (h) plus a wave height (Hs)

$H = h + Hs$

In which the water level has an exponential distribution, see Table 13, and the wave height has a normal distribution with its mean value dependent on the water level:

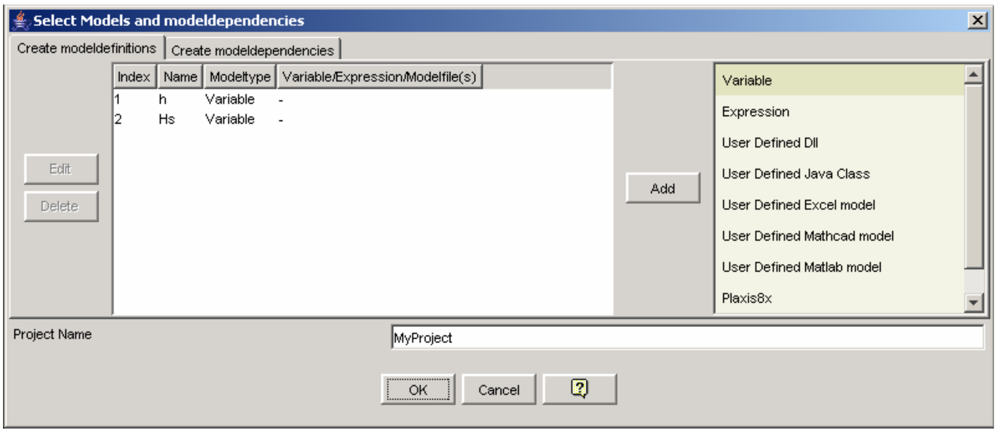
$\mu = 4.82 + 0.6h - 0.0063(7.0 - h)^{3.13}$
$\sigma = 0.6$

Table 13 Input of variables for fifth example

Variable	Symbol	Unit	Mean	St. dev	Distribution
Water level	h	m	2.52	0.33	Exponential
Wave height	Hs	m	f(h)	0.6	Normal

8.2 Defining equations and models

We start with defining 2 variables, like in Figure 76

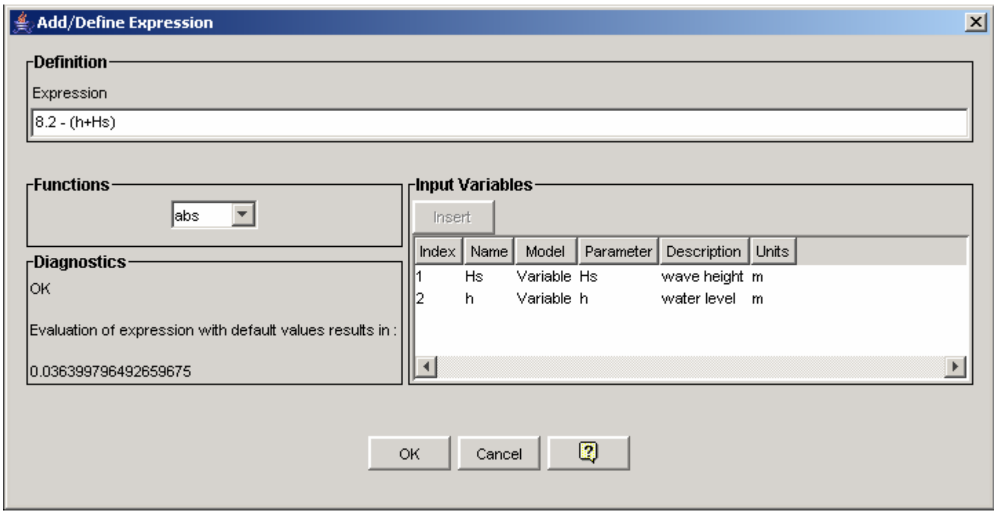


Index	Name	Modeltype	Variable/Expression/Modelfile(s)
1	h	Variable	-
2	Hs	Variable	-

Project Name: MyProject

Figure 76 Variables in fifth example.

Next the limit state function is defined according to Figure 77.



Expression

8.2 - (h+Hs)

abs

OK

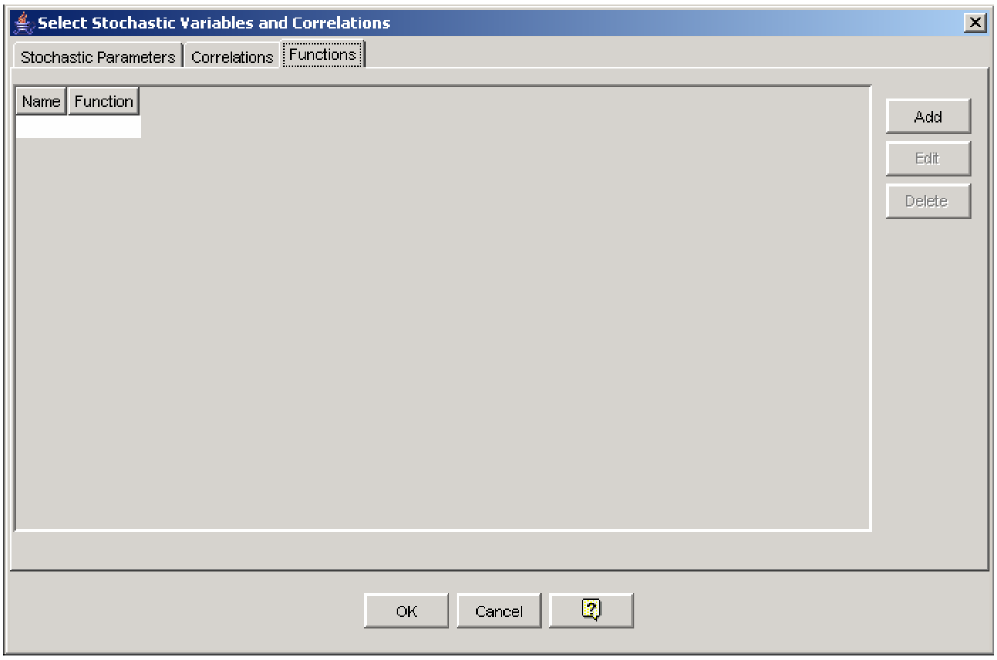
Evaluation of expression with default values results in :

0.036399796492659675

Index	Name	Model	Parameter	Description	Units
1	Hs	Variable	Hs	wave height	m
2	h	Variable	h	water level	m

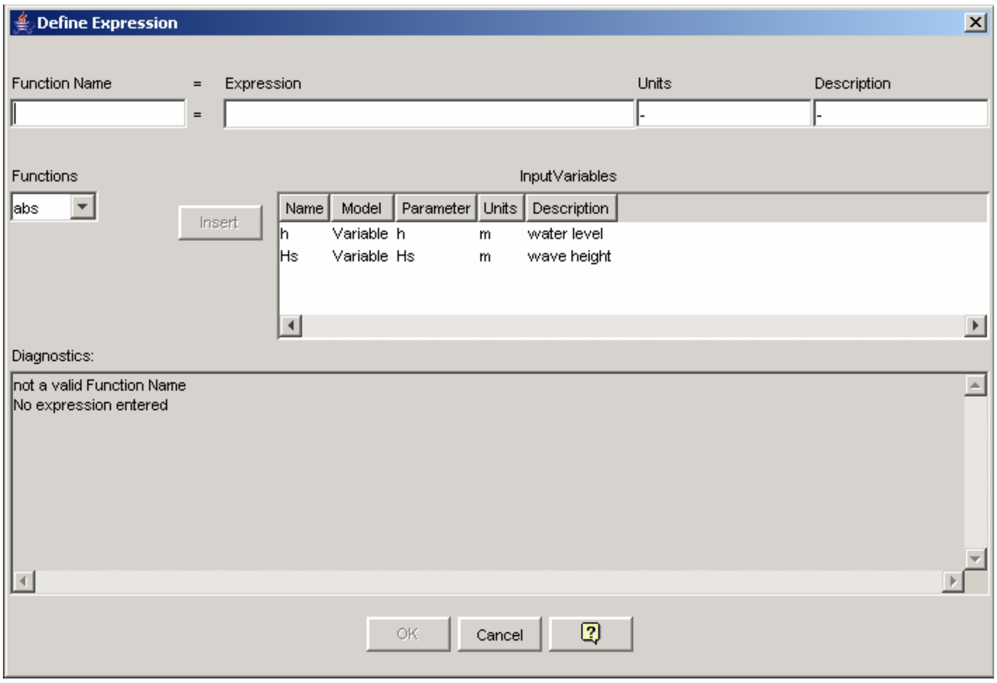
Figure 77 Limit state definition.

Finally we get to the definitions of the dependency for μ(Hs) in h. We go into the stochastic menu, and select the function tab, like in Figure 78. Clicking the add button brings us in the input menu for a stochastic function, see Figure 79.



Name	Function

Figure 78 Functions list in stochastic menu, initial view.



Define Expression

Function Name = Expression | Units | Description

Function Name:
Expression:
Units: -
Description: -

Functions: abs

InputVariables

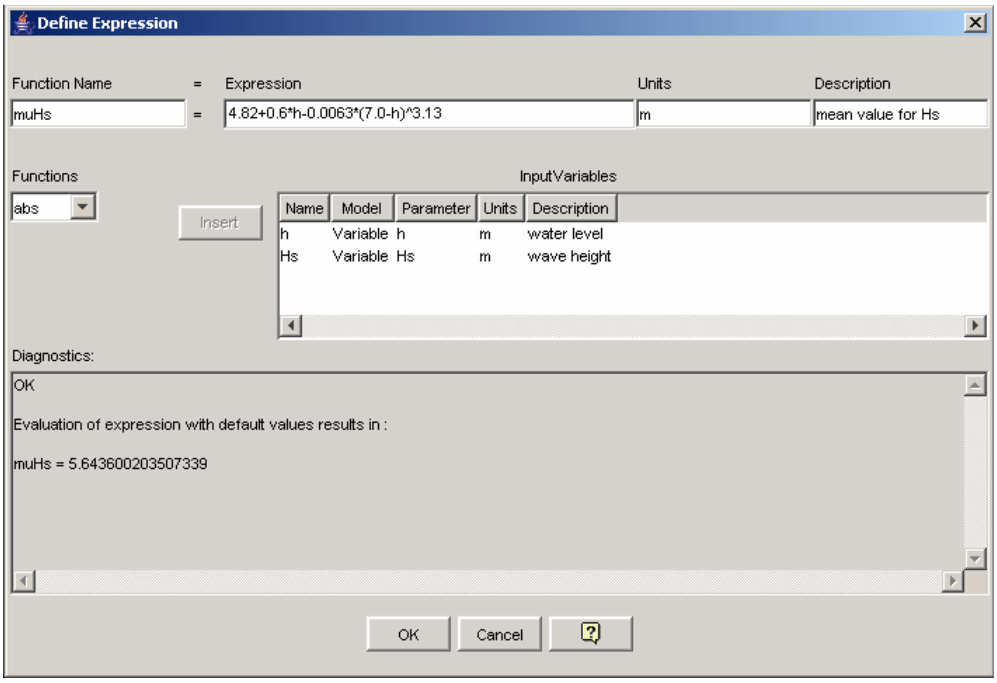
Name	Model	Parameter	Units	Description
h	Variable	h	m	water level
Hs	Variable	Hs	m	wave height

Diagnostics:

```
not a valid Function Name
No expression entered
```

Figure 79 Function definition, initial view.

We now can create a function for the mean value of Hs. Lets call this function ‘muHs’. The function is defined as shown in Figure 80.



Define Expression

Function Name: muHs
Expression: 4.82+0.6*h-0.0063*(7.0-h)^3.13
Units: m
Description: mean value for Hs

Functions: abs

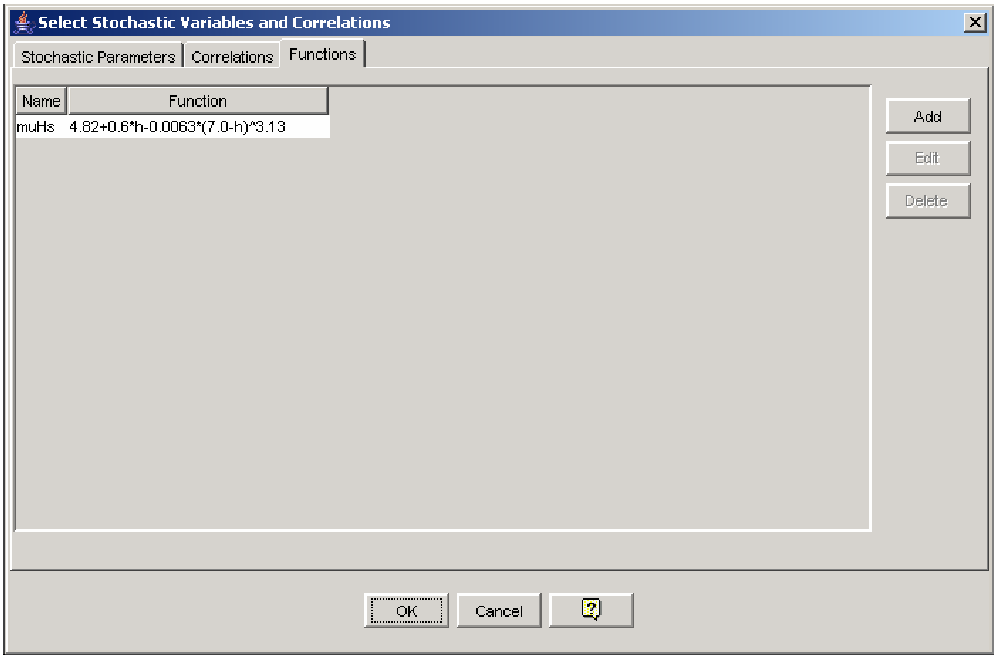
InputVariables

Name	Model	Parameter	Units	Description
h	Variable	h	m	water level
Hs	Variable	Hs	m	wave height

Diagnostics:

```
OK

Evaluation of expression with default values results in :

muHs = 5.643600203507339
```

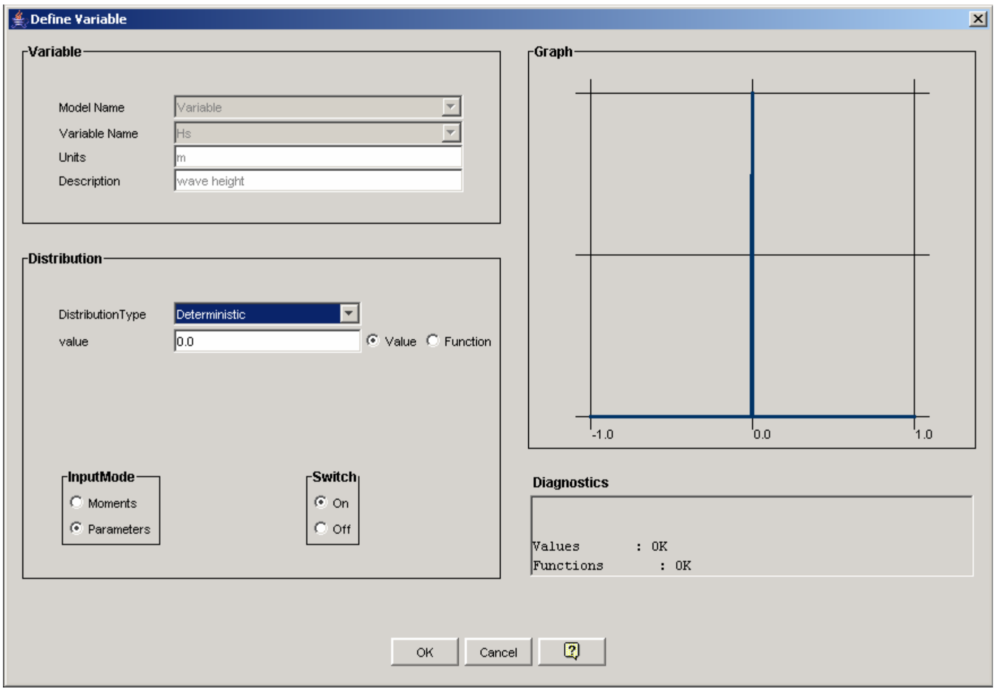
Figure 80 Function definition, final view.



Name	Function
muHs	4.82+0.6*h-0.0063*(7.0-h)^3.13

Figure 81 Functions list in stochastic menu, final view.

Clocking ‘OK’ brings us back one level, see Figure 81. Next we will be able to redefine the stochastic properties for Hs. Edit the stochastic properties for Hs. E.g. initially it might look like Figure 82. We see that with a stochastic function now available, we are able to switch between ‘function’ and ‘value’ through the checkboxes in the distribution window.



Variable

Model Name: Variable
Variable Name: Hs
Units: m
Description: wave height

Distribution

DistributionType: Deterministic
value: 0.0 ● Value ○ Function

InputMode
○ Moments
● Parameters

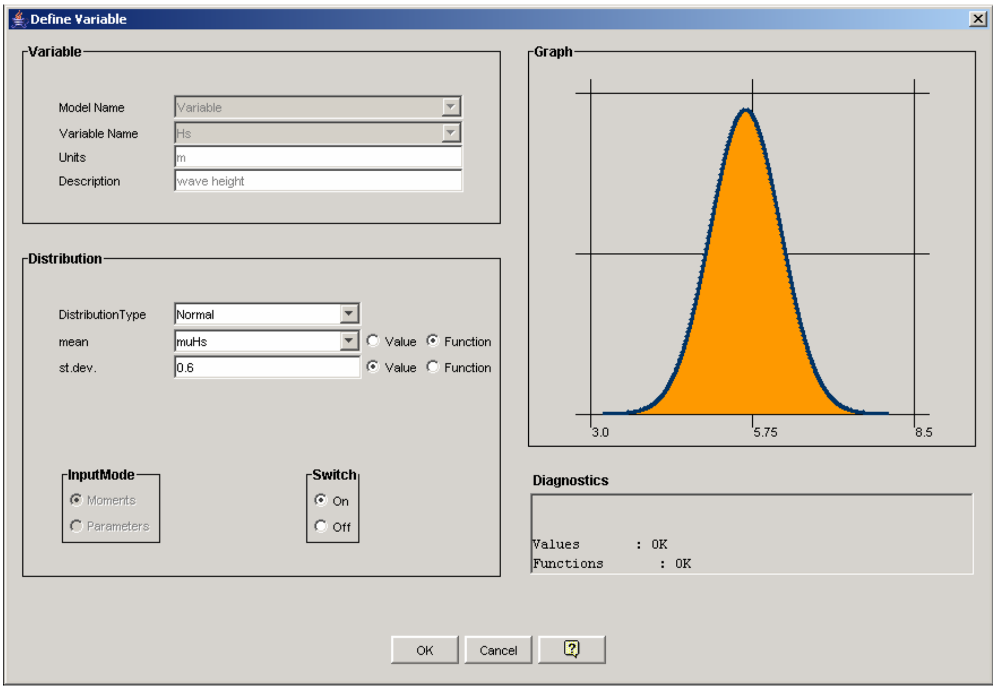
Switch
● On
○ Off

Diagnostics

```
Values       : OK
Functions       : OK
```

Figure 82 Editing stochastic properties for Hs, initial view.

But first we choose the normal distribution. For the mean value we switch the function option on. As a result the input field changes into a combo box with the predefined functions. With only one function available in this example, we select ‘muHs’. The stochastic settings are completed as in Figure 83. Going back one level (clicking OK) we see the stochastic function ‘muHs’ also present in the property list for variable’ Hs’, see Figure 84.



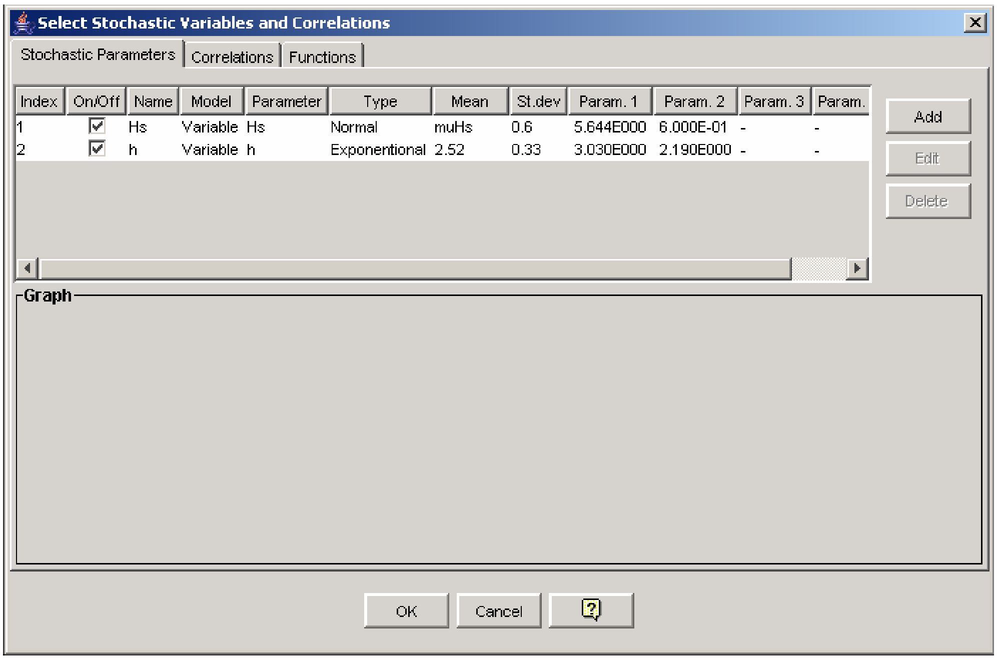
Model Name: Variable
Variable Name: Hs
Units: m
Description: wave height

DistributionType: Normal
mean: muHs ○ Value ● Function
st.dev.: 0.6 ● Value ○ Function

InputMode: ● Moments ○ Parameters
Switch: ● On ○ Off

```
Values       : OK
Functions       : OK
```

Figure 83 Editing stochastic properties for Hs, final view

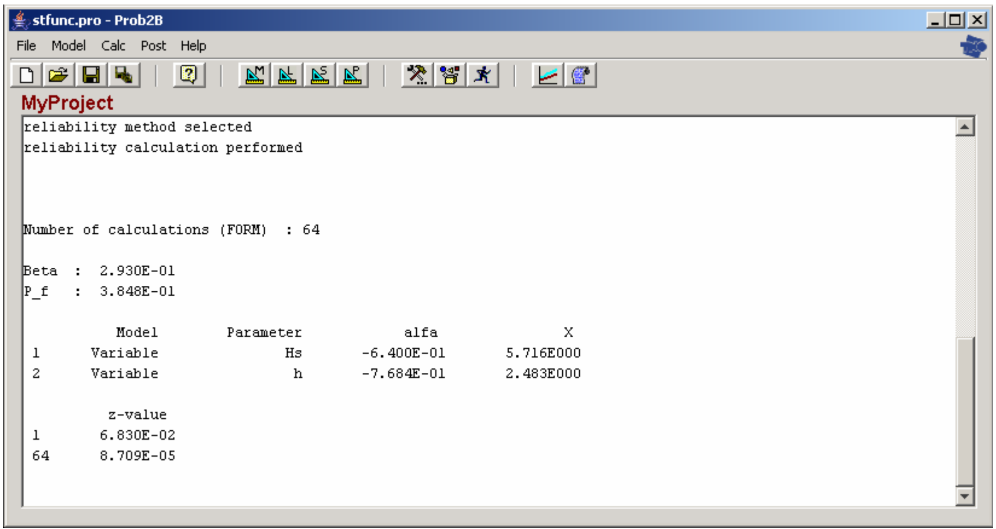


Index	On/Off	Name	Model	Parameter	Type	Mean	St.dev	Param. 1	Param. 2	Param. 3	Param.
1	☑	Hs	Variable	Hs	Normal	muHs	0.6	5.644E000	6.000E-01	-	-
2	☑	h	Variable	h	Exponentional	2.52	0.33	3.030E000	2.190E000	-	-

Figure 84 List of stochastic variables, including function ‘muHs’ as mean for Hs

8.3 Example of calculation results

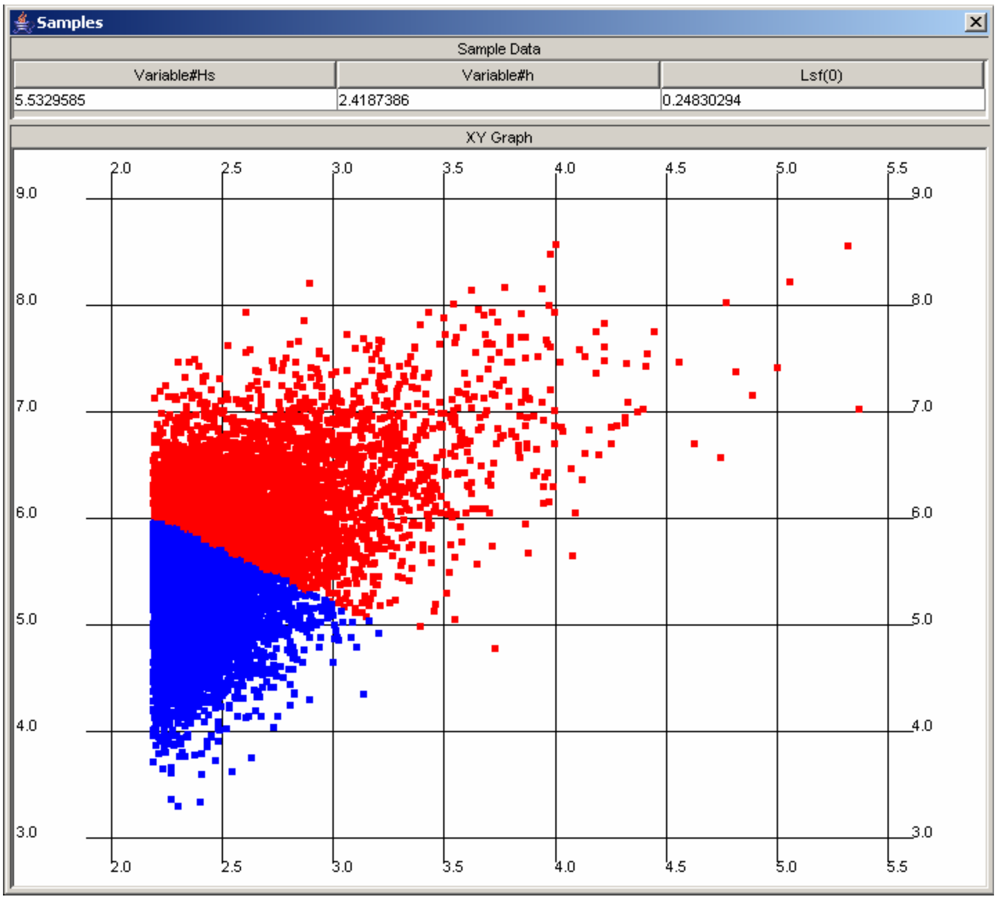
A FORM calculation is performed on the limit state.
The results are presented in Figure 85.



```
reliability method selected
reliability calculation performed



Number of calculations (FORM)  : 64

Beta  :  2.930E-01
P_f   :  3.848E-01

           Model        Parameter           alfa            X
 1       Variable              Hs      -6.400E-01       5.716E000
 2       Variable               h      -7.684E-01       2.483E000

          z-value
 1       6.830E-02
 64      8.709E-05
```

Figure 85 FORM results of 5th example

Next a MC calculation is also performed (10^4 samples), with the results presented as a scatter diagram in Figure 86. In this figure values are plotted for h (horizontal axle) and Hs (vertical axle). A dependency of Hs on h can be seen in the scatter plot. The colours indicate the limit state function values: blue for values >= 0, red for values <0.



Variable#Hs	Variable#h	Lsf(0)
5.5329585	2.4187386	0.24830294

Figure 86 MC results of 5th example

9 References

[1] R.L. Iman and J.C. Helton, 1985, *A comparison of uncertainty and sensitivity analysis techniques for computer models*, Report NUREG/CR-3904,Sandia Nat. Lab., Alberqueque, New Mexico.

[2] A. Vrouwenvelder, *Monte Carlo Importance Sampling - Application to Structural Reliability Analysis*, TNO Report B-83-529, TNO Building and Construction Research, Rijswijk, The Netherlands, 1983.

[3] O. Ditlevsen & P. Bjerager, *Plastic Reliability Analysis by Directional Simulation*, Report No. 353, June 1987.

[4] Ditlevsen, O. en H.O. Madsen, *Structural Reliability Methods*, Chichester: John Wiley & Sons, 1996.

[5] C.G. Bucher, *Adaptive Sampling - An Iterative Fast Monte-Carlo Procedure*, W. Ouypornprasert, Adaptive Numerical Integration for Reliability Analysis, Report 12-87, Insbruck, 1987.

[6] C.F. de Valk, *Estimation of the frequencies of failure of coastal structures from offshore data of environmental loads*, Report H2240, Delft Hydraulics, February 1996.

[7] A.M. Hasofer & N.C. Lind, *An exact and invariant first-order reliability format*, ASCE, 100 (EM1), 1974, pages 111-121.

[8] K. Breitung & M. Hohenbichler, *Some Asymptotic Results in Second Order Reliability*, 1984 Berichte zur Zuverlässigkeitstheorie der Bauwerke, München.

[9] M. Hohenbichler & R. Rackwitz, *First-order Concepts in System Reliability*, Structural Safety, 1, 1983, pages 177-188.

[10] O. Ditlevsen, *Narrow reliability bounds for structural systems*, Journal of Structural Mechanics 7 (4), 1979, pages 453-472.

[11] J. Stevenson & F. Moses, *Reliability Analysis of Frame Structures*, Journal of Structural Division, ASCE, Vol. 96, ST11, November 1970, pages 2409-2427.

[12] A. Vrouwenvelder, *Rekenmethoden voor een probabilistische dijkringanalyse*,TNO Bouw rapport 93-CON-R1332, Rijswijk, 30 november 1993.

[13] M. Rosenblatt, *Remarks on a Multivariate Transformation*, The annals of Mathematical Statistics, Vol. 23, pages 470-472, 1952.

[14] K. Stange, *Angewandte Statistik - Zweiter Teil, Mehrdimensionale Probleme*, Springler-Verlag, 1971 (290-295).

A Theoretical Background (in Dutch)

A.1 Faalkans van een enkel element

A.1.1 Inleiding

De kans op overschrijding van de faalgrens bij een enkel element kan formeel worden geschreven als:

$$P(F) = P(\ Z(X) < 0\ ) \qquad \text{(A.1.1.1)}$$

waarin:
P is de kans;
F is de gebeurtenis Falen;
Z is de grenstoestandsfunctie;
X is de vector met stochastische variabelen.

De grenstoestand is derhalve zodanig gedefinieerd dat $Z < 0$ duidt op een overschrijding van de beschouwde grenstoestand en daarmee op falen. Het oplossen van (A.1.1.1) vereist een rekentechniek. In tabel A.1 zijn een aantal van deze technieken samengevat.

Elke genoemde methode heeft voor- en nadelen. Zo is een analytisch resultaat meestal ideaal, maar lang niet altijd mogelijk. Numerieke Integratie is nauwkeurig en betrouwbaar maar zeer tijdrovend bij meer dan een gering aantal variabelen. Crude Monte Carlo is eenvoudig van opzet, maar tijdrovend bij een ingewikkeld rekenmodel en een kleine faalkans. De Monte Carlo varianten als Importance sampling, Directional Sampling en de extreme-waardenschaling (Methode De Haan) brengen daar verbetering in, maar kennen ook hun beperkingen.

Tabel A.1 Berekeningstechnieken

Methode	**Variant**	**Lit**
Analytisch		
Numerieke Integratie	Full	[1]
	Hypercube sampling	
Monte Carlo	Crude	
	Importance sampling	[2]
	Directional sampling	[3][4]
	Adaptive sampling	[5]
	Extreme waardenschaling	[6]
FORM First Order Reliability Method		[7]
SORM Second Order Reliability Method		[8]
SYSTEEM analyse: $P(Z_1<0 \text{ EN } Z_2<0 \text{ EN } ..)$	Hohenbichler/Rackwitz	[9]
	Ditlevsen Bounds	[10]
	Stevenson Moses	[11]
	Oprolmethode	[12]

FORM is een bijzonder snelle methode; de nadelen zijn echter:

1. er treedt soms geen convergentie op;
2. het resultaat is onnauwkeurig als de Z-functie sterk gekromd is;
3. men vindt in de minimaliseringsprocedure soms een lokaal i.p.v. een globaal minimum.

De eerste twee nadelen zijn meestal wel op te heffen (via zorg voor continue en continu differentieerbare Z-functies, respectievelijk via het maken van een aanvullende SORM berekening), het derde nadeel is echter moeilijker te omzeilen. Men kan hier iets aan doen door een aantal berekeningen te doen met verschillende (random gekozen) startwaarden of gebruik te maken van voorkennis. Tenslotte: FORM kan in beginsel geen sommen aan van het type $P(Z_1 < 0 \text{ EN } Z_2 < 0)$. Dit betekent dat hiervoor aparte zogenaamde “SYSTEEM-analyse-routines” nodig zijn. Ook daarvoor zijn enkele varianten opgenomen in tabel A.1.

Het is overigens niet nodig dat men zich hoeft te beperken tot één enkele methode. Integendeel, men kan vaak met succes verschillende methoden combineren. Zo kan men bijvoorbeeld een eenvoudige methode als FORM gebruiken voor ”productie-runs”, maar deze een paar maal controleren met Monte Carlo of Numerieke Integratie. Ook kan men binnen een berekening een deel met de ene methode en een deel met de andere methode uitvoeren.

De routines voor het bepalen van de faalkans van één enkel element worden hierna in meer detail besproken.

A.1.2 FORM

FORM staat voor First Order Reliability Method [7]. De term “First Order” verwijst naar de linearisering van de grenstoestandsfunctie waarvan de methode gebruik maakt. De linearisering wordt uitgevoerd in een punt dat meestal als “Design Point” of "ontwerppunt" wordt aangeduid. Dit punt is gedefinieerd als het punt op de faalgrens $Z = 0$ waar de kansdichtheid maximaal is. Dit punt is van te voren niet bekend en het moet via een iteratieprocedure worden gezocht.

De FORM procedure heeft verder een set van standaard normaal verdeelde variabelen u als werkruimte. De variabelen u_i hebben per definitie een gemiddelde 0 en een standaardafwijking 1 en zijn onderling onafhankelijk:

$$\mu(u) = 0 \quad \sigma(u) = 1 \quad \rho(u_i\, u_j) = 0$$

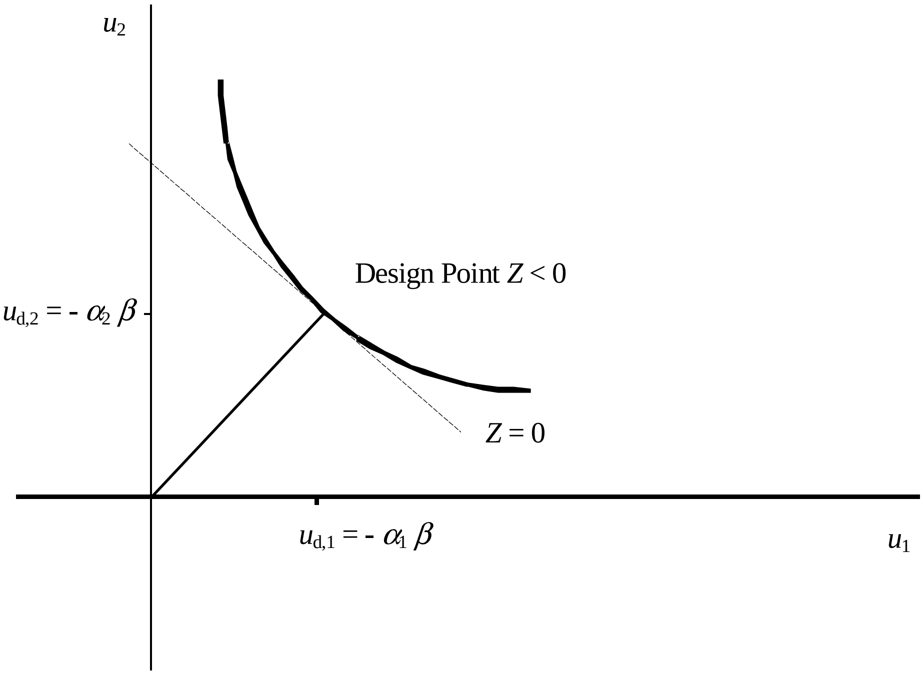
De fysische probleemvariabelen X van vergelijking (A.1.1.1) volgen daarbij via een transformatie van u naar X. Het voordeel van deze werkwijze is dat er een heldere interpretatie van het “Design Point” ontstaat: dit is namelijk het punt op de grens $Z = 0$ met de kortste afstand van de oorsprong tot de kromme $Z(u) = 0$ (zie figuur A.1). Voor een normaal verdeelde variabele X is de transformatie van u naar X eenvoudig te schrijven als:

$$X = \mu_X + u\ \sigma_X \qquad \text{(A.1.2.1a)}$$

Voor een variabele X met een willekeurige verdeling $F_X(x)$ volgt de transformatie via gelijkstelling van de onderschrijdingskansen:

$$F_X(x) = \Phi(u) \tag{A.1.2.1b}$$

met $\Phi(..)$ de verdelingsfunctie van de standaard normale verdeling.



Figuur A.1 - Definitie van Design Point in de u*-ruimte*

Veronderstel nu dat de grenstoestandsfunctie $Z(X)$ in het Design Point via een of andere procedure is gelineariseerd. Het proces op zich om tot het design point te komen, laten we hier verder buiten bespreking. De gelineariseerde Z-functie Z_L is in zijn algemeenheid te schrijven als:

$$Z_L = B + A_1\,u_1 + A_2\,u_2 + ... \tag{A.1.2.2}$$

De betrouwbaarheidsindex $\beta = \mu(Z_L) \;/\; \sigma(Z_L)$ (zie [7]) kan men dan wegens de statistische eigenschappen van de u-variabelen eenvoudig vinden via:

$$\mu(Z_L) = B$$

$$\sigma(Z_L) = \sqrt{}\,\Sigma\,A_i^2$$

$$\beta = B\,/\sqrt{}\,\Sigma\,A_i^2 \tag{A.1.2.3}$$

De sommatie is van $i = 1$ tot n, met n het aantal stochasten. De faalkans volgt uit:

$$P(Z < 0) \approx P(Z_L < 0) = \Phi(-\beta) \tag{A.1.2.4}$$

Met $\Phi(..)$ wederom de verdelingsfunctie van de standaard normale verdeling.

Gegeven de eigenschap van het Design Point dat dit het punt is op de lijn $Z = 0$ dat het dichtst bij de oorsprong ligt, zijn de coördinaten van het Design Point eenvoudig met behulp van een vlakke meetkunde beschouwing te vinden als:

$$u_{d,i} = - A_i \, B \,/\, \Sigma \, A_j^2 \tag{A.1.2.5}$$

Aangezien een grenstoestandsfunctie zonder bezwaar kan worden vermenigvuldigd met of worden gedeeld door een willekeurig positief getal, is het handig de coëfficiënten B en A_i in de grenstoestand-functie (A.1.2.2) te delen door $\sqrt{\Sigma \, A_j^2}$. Er ontstaat dan de volgende gestandaardiseerde vorm:

$$Z_L = \beta + \alpha_1 \, u_1 + \alpha_2 \, u_2 \; .... \tag{A.1.2.6}$$

met:

$$\alpha_i = A_i \,/\, \sqrt{\Sigma \, A_j^2} \tag{A.1.2.7}$$

De sommatie $\Sigma \, A_j^2$ is nog steeds van $j = 1$ tot n. Het Design Point u_d is nu te schrijven als (zie figuur A.1):

$$u_{d,i} = - \alpha_i \, \beta \tag{A.1.2.8}$$

De betrouwbaarheidsindex β blijkt dan te interpreteren als de afstand van het Design Point tot de oorsprong, immers:

$$||0 - \mathrm{DP}||^2 = \Sigma \, (u_{d,i})^2 = \Sigma(\alpha_i \, \beta)^2 = \beta^2 \, \Sigma \, \alpha_i^2 = \beta^2$$

De Design Point waarden X_d volgen dan uit:

$$X_{d,i} = F^{-1} \, \Phi(u_{d,i}) = F^{-1} \, \Phi(-\alpha_i \, \beta) \tag{A.1.2..9}$$

Samenvattend kan worden gesteld dat een FORM-berekening tot de volgende resultaten leidt:

- een betrouwbaarheidsindex β
- een set invloedscoëfficiënten α_i

Hieruit vallen dan verder af te leiden:

· de faalkans $P(F)$	via (A.1.2.4)
· een standaard uitdrukking voor de betrouwbaarheidsfunctie	via (A.1.2.6)
· het Design Point u_d	via (A.1.2.8)
· het Design Point X_d	via (A.1.2.9)

Met name de gestandaardiseerde formulering van de Z-functie speelt een belangrijke rol bij de faalkans van een systeem.

A.1.3 SORM

De procedure SORM bestaat uit een FORM-berekening gevolgd door een tweede orde correctie. Ook in deze beschrijving van SORM wordt het probleem niet in de basisvariabelen X gesteld, maar in de u-werkruimte.

De aan SORM voorafgaand uitgevoerde FORM berekening leidt tot (zie paragraaf A.1.2):

- een betrouwbaarheidsindex β
- een set invloedscoëfficiënten α_i
- het Design Point $u_{d,i} = -\alpha_i \beta$

en de grenstoestandsfunctie Z die, gelineariseerd in het Design Point, wordt gegeven door:

$$Z_L = Z(\mathbf{u}_d) + \sum \alpha_i (u_i - u_{d,i}) = 0 + \sum \alpha_i u_i - \sum -\alpha_i^2 \beta = \beta + \sum \alpha_i u_i \quad \text{(A.1.3.1)}$$

Door de tweede afgeleiden te bepalen, kan voor Z een tweede orde ontwikkeling worden uitgevoerd:

$$Z_Q = \beta + \sum \alpha_i u_i + \frac{1}{2} \sum\sum \frac{\partial^2 Z}{\partial u_i \partial u_j} (u_i - u_{d,i})(u_j - u_{d,j}) \quad \text{(A.1.3.2)}$$

De tweede afgeleiden worden uiteraard ontwikkeld in het Design Point. De sommaties lopen van 1 tot n, waarbij n het aantal stochasten voorstelt.

Vervolgens voeren we een rotatie-transformatie uit van u_i naar v_i. De v_1-as wordt daarbij door het Design Point gekozen (zie figuur A.2). Hierbij wordt:

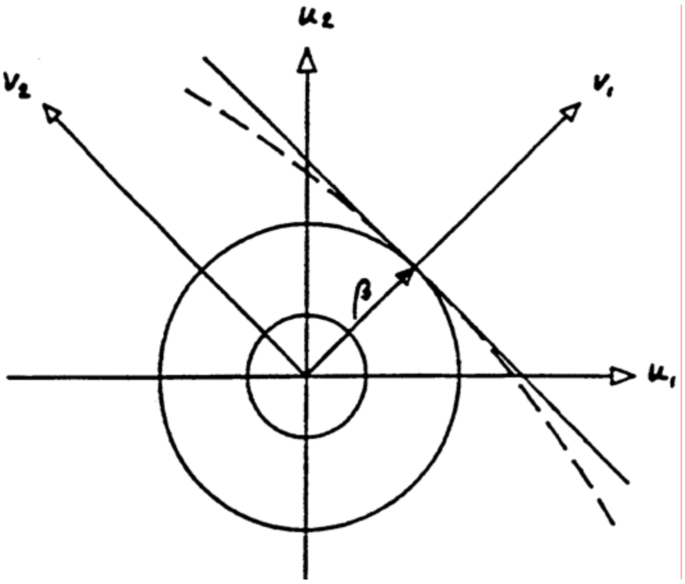
$$v_{d,1} = \beta \ en \ v_{d,2} = v_{d,3} = ... = 0,$$

$$\alpha_1 = -1 \ en \ \alpha_2 = \alpha_3 = ... = 0$$

De grenstoestandsfunctie wordt daarmee:

$$Z_Q = \beta - v_1 + \frac{1}{2} \sum \sum \frac{\partial^2 Z}{\partial v_i \partial v_j} (v_i - v_{d,i})(v_j - v_{d,j}) \quad \text{(A.1.3.3)}$$

De sommaties lopen van 1 tot n.



Figuur A.2 - Transformatie naar (v1, v2)

Uitwerking voor 2 variablen

Beschouw nu eerst het probleem in 2-dimensies. Voor de tweede afgeleiden geldt:

$$\frac{\partial^2 Z}{\partial v_1^2}, \frac{\partial^2 Z}{\partial v_1 \partial v_2} = \frac{\partial^2 Z}{\partial v_2 \partial v_1} en \frac{\partial^2 Z}{\partial v_2^2} \tag{A.1.3.4}$$

Van deze drie tweede afgeleiden is met name de laatste van belang. Deze immers bepaalt het verloop van $Z = 0$ in het v_1-v_2-vlak. De andere twee termen bepalen respectievelijk het verloop in het Z-v_1-vlak en de wringing (zie figuur A.3).

We herschrijven daarom de betrouwbaarheidsfunctie (A.1.3.3) als (bedenk dat $v_{d,2}= 0$):

$$Z_Q = \beta - v_1 + \frac{1}{2} v_2^2 \frac{\partial^2 Z}{\partial v_2^2} \tag{A.1.3.5}$$

Voor de tweede afgeleide geldt:

$$\frac{\partial^2 Z}{\partial v_2^2} = -\kappa_{22} = -\frac{1}{R_{22}} \tag{A.1.3.6}$$

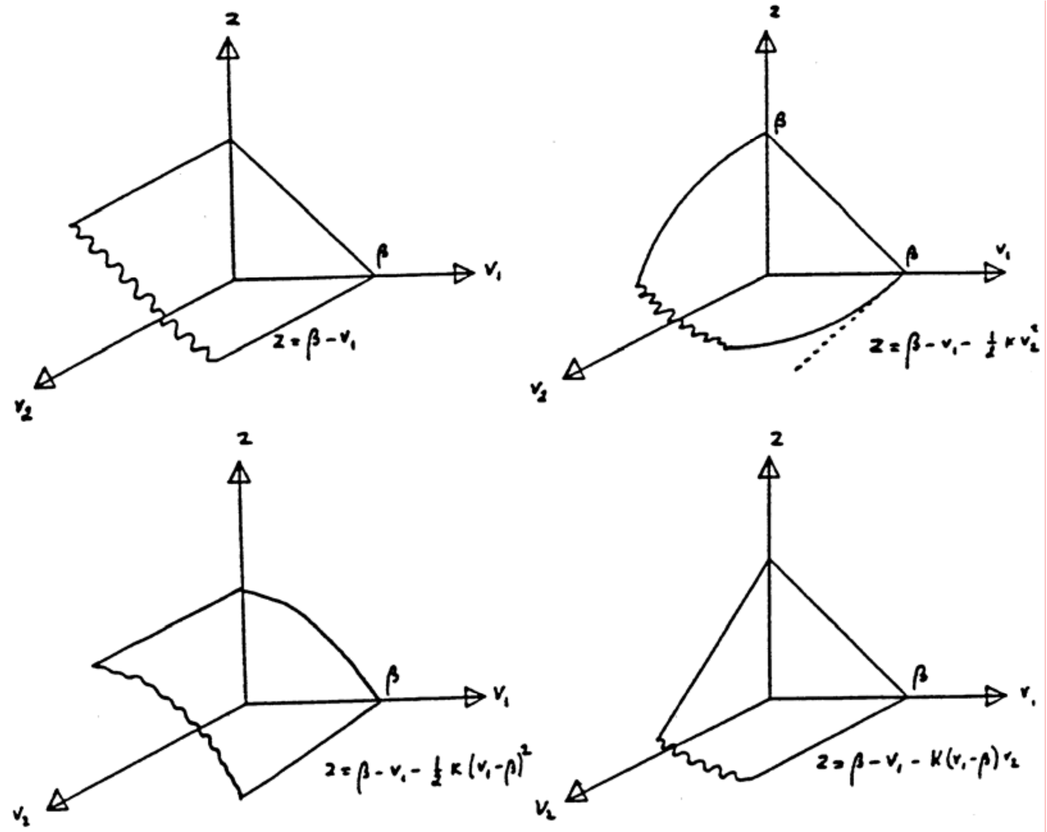
Waarin k_{22} de kromming en R_{22} de kromtestraal voorstelt.

Herschrijven van (A.1.3.5) in de kromming geeft:

$$Z_Q = \beta - v_1 - \frac{1}{2} \kappa_{22} v_2^2 \tag{A.1.3.7}$$

Voor de kans $P\{Z_Q < 0\}$ heeft Breitung [8] de volgende benadering afgeleid:

$$P\{Z_Q < 0\} = \Phi(-\beta)\{1-\beta\kappa_{22}\}^{-\frac{1}{2}} \qquad \text{(A.1.3.8)}$$



Figuur A.3 - Invloed van de verschillende kwadratische termen

Voorbeeld:

Beschouw ter controle op de nauwkeurigheid van de methode het probleem:

$$Z = \beta - v_1 - \kappa_{22} * v_2^2 / 2$$

Voor β = 3,0 kan berekend worden:

κ_{22}	exact	$\Phi(-\beta)*(1-\beta\kappa_{22})^{-1/2}$
0,0	0,00135	0,00135
0,1	0,00163	0,00161
0,2	0,00212	0,00213
0,25	0,00245	0,00270
0,3	0,00297	0,00426
0,32	0,00320	0,04676
0,4	0,00441	oneindig

Voor een goede benadering dient $\beta\ \kappa_{22} < 0{,}75$.

Uitwerking voor *n* variabelen

Bij meer dan twee dimensies wordt de matrix $\partial^2 Z / \partial v_i \partial v_j$ ontdaan van de eerste kolom en de eerste rij. Dit levert de matrix G op met:

$$G_{ij} = \partial^2 Z / \partial v_{i+1} \partial v_{j+1} \tag{A.1.3.9}$$

$i, j = 1 \ldots n - 1$

Daarna worden in de $\{v_2 \ldots v_n\}$ - ruimte de hoofdkrommingen opgezocht via het oplossen van:

$$\text{Det} \, \| G - \kappa I \| = 0 \tag{A.1.3.10}$$

waarin:
I is de eenheidsmatrix;
κ is de vector met (n - 1) hoofdkrommingen.

De faalkans wordt gegeven door:

$$P\{F\} = \Phi(-\beta) \prod_{i=1}^{n-1} (1 - \beta\kappa_i)^{-1/2} \tag{A.1.3.11}$$

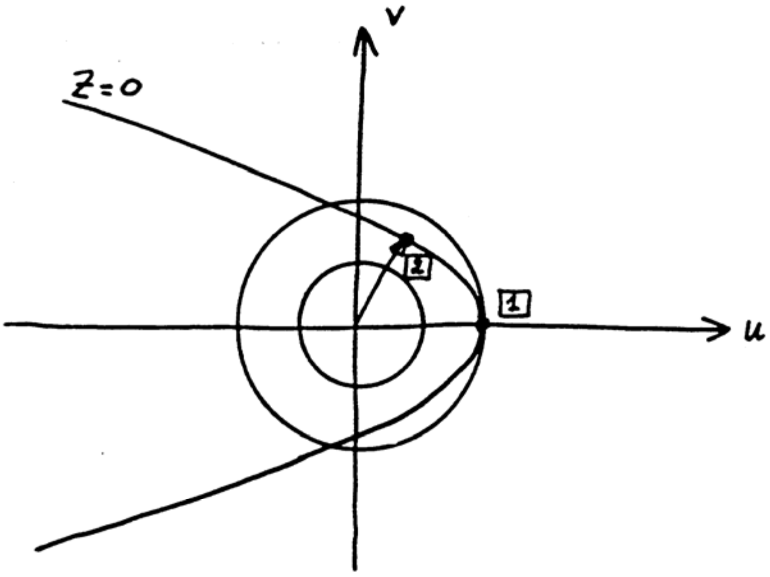
Teruggerekend uit de faalkans kan een equivalente β worden gevonden. Het α_i-equivalent wordt gelijk gehouden aan de eerste orde α_i, al bestaan er methoden om hierop een correctie aan te brengen onder de voorwaarde dat κ_{22} heel klein is.

Opmerkingen:

1. Het feit dat het punt $v_d = (\beta, 0, \ldots 0)^T$ een Design Point is zoals volgt uit een FORM-procedure heeft tot gevolg dat de krommingen aan bepaalde voorwaarden voldoen. Immers als de krommingen te sterk zijn, dan is er in de omgeving van het punt v_d een punt op de kromme $Z = 0$ dat dichter bij de oorsprong ligt. Het punt v_d is dan geen design point meer (zie figuur A.4).
2. Vat de betrouwbaarheidsfunctie = $Z(v_1, v_2)$ op als $v_1 = v_1(v_2)$. De kromming die in feite van belang is voor de tweede orde correctie is d^2v_1/dv_2^2. Hiervoor valt, uitgaande van (A.1.3.3), af te leiden:

$$\frac{d^2 v_1}{dv_2^2} = \frac{\kappa_{22}}{1 - \beta\kappa_{11}}$$

Conclusie: κ_{11} is eigenlijk ook van belang en (A.1.3.5) is alleen geldig bij kleine κ_{11}. Hiermee wordt in alle huidige SORM programma’s geen rekening gehouden. Mogelijk moet dit punt nog eens nader worden uitgezocht.



Figuur A.4 - Bij te sterke kromming is niet 1 maar 2 het Design Point

A.1.4 Crude Monte Carlo

Crude Monte Carlo is de basismethode uit de Monte-Carlo-familie. Bij deze methode wordt net als bij de FORM-berekening uitgegaan van de formulering van de Z-functie in de u-variabelen.

Bij een standaard Monte-Carlo-Sampling bestaat elke trekking uit een set u_i-waarden. De faalkans kan dan worden geschat door de frequentie te bepalen waarmee deze trekking zich in het faalgebied bevinden.

Deze trekking wordt N keer herhaald en vervolgens wordt de faalkans berekend uit:

$$P(F) = N_f / N = (1 / N) \sum_{i=1}^{N} I(Z(u)) \qquad \text{(A.1.4.1)}$$

waarin:

N_f	is het aantal trekkingen waarbij falen optrad,
N	is het totaal aantal trekkingen,
i	is het nummer van de trekking,
$I(Z(u) = 1$	indien $Z(u) \leq 0$ en
$I(Z(u) = 0$	indien $Z(u) > 0$

A.1.5 Directional Sampling

Directional Sampling is een methode uit de Monte-Carlo-familie. Bij deze methode wordt net als bij de FORM-berekening uitgegaan van de formulering van de Z-functie in de u-variabelen.

Bij een standaard Monte-Carlo-Sampling bestaat elke trekking uit een set u_i-waarden. De faalkans kan dan worden geschat door de frequentie te bepalen waarmee deze trekking zich in het faalgebied bevinden.

Bij Directional Sampling wordt een willekeurige trekking $u = (u_1,...,u_n)$ gedeeld door de lengte van de vector u en vermenigvuldigd met λ, zodanig dat $Z(\lambda\ u\ /\ |u|) = 0$. De waarde van λ is dus altijd groter dan 0 en meestal groter dan 1,0.

De som van de kwadraten $R^2 = \sum_{i=1}^{n} u_i^2$ heeft een chi-kwadraat-verdeling met n vrijheidsgraden. De kans op falen is dus, als λ in elke richting constant zou zijn, gelijk aan:

$$P(F) = - \chi^2 (\lambda, n) \qquad \text{(A.1.5.1)}$$

waarin:
$\chi^2(..)$ is de chi-kwadraatverdeling;
n is het aantal vrijheidsgraden (aantal stochastische grootheden).

Deze trekking wordt N keer herhaald en vervolgens wordt de faalkans geschat uit:

$$P(F) = (1 / N) \sum_{i=1}^{N} P_\mathrm{i} \text{ met } P_\mathrm{i} = 1 - \chi^2 (\lambda_\mathrm{i}, n) \qquad \text{(A.1.5.2)}$$

waarin:
N is het aantal trekkingen,
i is het nummer van de trekking,
λ_i is de λ-waarde horend bij trekking i.

De bijbehorende variantie is:

$$\sigma_{P(F)}^2 = \frac{1}{N(N-1)} \sum_{i=1}^{N} \{P_i - P(F)\}^2 \qquad \text{(A.1.5.3)}$$

Voorbeeld:

Neem als voorbeeld de Z-functie (zie figuur C.5):

$$Z = 3 - (u_1 + u_2)$$

u_1 en u_2 zijn daarbij onafhankelijke normaalverdeelde stochasten met gemiddelde 0 en standaardafwijking 1. In onderstaande tabel is hiervoor de Directional-Sampling procedure uitgewerkt, waarbij is uitgegaan van 20 samples.

sample	u1	u2	lambda	p
1	1.111	-1.698	-10.38	.
2	.971	.464	2.25	.0796
3	-.36	-1.379	-2.459	.
4	-.449	-.033	-2.801	.
5	.495	.492	2.121	.1054
6	.841	1.	2.129	.1036
7	-.031	.222	3.53	.002
8	.752	-.58	16.607	.
9	-.53	.234	-5.882	.
10	.791	.759	2.122	.1053
11	.57	1.586	2.345	.0639
12	.022	.375	2.837	.0179
13	.262	1.954	2.669	.0284
14	-1.522	.606	-5.367	.
15	-.174	-.122	-2.155	.
16	.944	.819	2.127	.1042
17	1.249	.297	2.491	.0449
18	-.429	-1.073	-2.308	.
19	1.061	.9	2.128	.1038
20	-.28	.041	-3.554	.

De tweede kolom zijn de trekkingen voor u_1 en de derde voor u_2. In de vierde kolom staat de waarde voor λ die volgt uit de betrekking:

$$3-\lambda\frac{u_1+u_2}{\sqrt{u_1^2+u_2^2}}=0$$

waarmee:

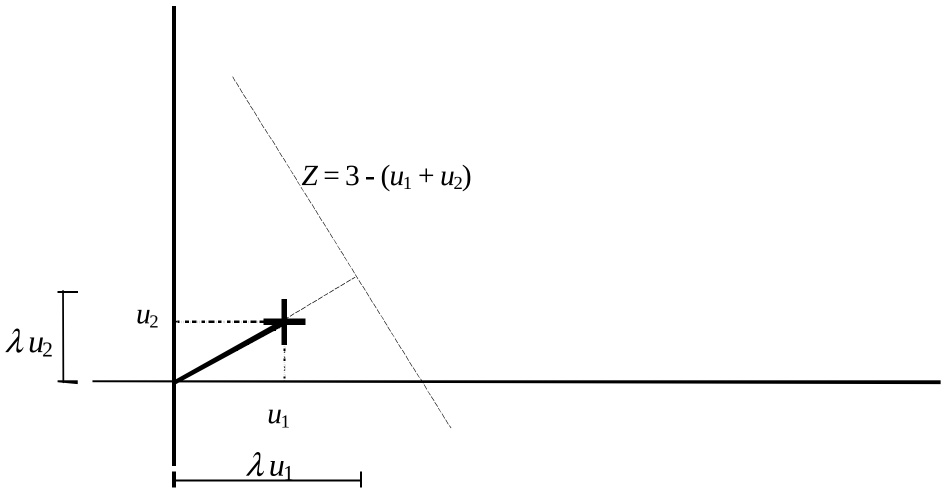
$$\lambda=\frac{3\sqrt{u_1^2+u_2^2}}{u_1+u_2}$$

In de vijfde kolom staat de overschrijdingskans volgens de chi-kwadraatverdeling met 2 vrijheidsgraden. Bij 2 variabelen is de chi-kwadraat-verdeling gelijk aan een Rayleigh-verdeling:

$$P = \exp\{-\lambda^2 / 2\}$$

Voor negatieve waarden van λ is de kans gelijk aan nul omdat dan de ‘$Z = 0$’-grens in het geheel niet gesneden wordt. De faalkans volgt ten slotte door de kansen P op te tellen en door 20 te delen:

$P\{Z < 0\} = 0{,}76 / 20 = 0{,}038$



Figuur A.5 - Z-functie voor het Directional-Samplingvoorbeeld

De exacte waarde is in dit geval natuurlijk eenvoudig te bepalen:

$$\beta = \frac{3}{\sqrt{\sigma^2(u_1) + \sigma^2(u_2)}} = \frac{3}{1{,}4} = 2{,}14$$

De faalkans die daarbij hoort is:

$P\{Z < 0\} = \Phi(-2{,}14) = 0{,}016$

De fout bedraagt dus een factor 2, maar dat mag gezien het zeer gering aantal samples ook geen verwondering wekken. De standaardafwijking volgens (A.1.5.3) van $P(Z < 0)$ is gelijk aan 0,010.

A.1.6 Numerieke Integratie

Bij numerieke integratie kan net als bij de FORM-berekening worden uitgegaan van de formulering van de *Z*-functie in de *u*-variabelen.

$$P(F) = \sum_{i_1=-\infty}^{\infty} ,... \sum_{i_n=-\infty}^{\infty} I(Z(u))\, f(u_1,..,u_n) \Delta u_1,...\Delta u_n \qquad \text{(A.1.6.1)}$$

waarin:
$I(Z(u) = 1$ indien $Z(u) \leq 0$ en
$I(Z(u) = 0$ indien $Z(u) > 0$

A.1.7 Correlaties

Bij de hiervoor genoemde rekentechnieken wordt uitgegaan van een set van standaard normaal verdeelde variabelen *u* als werkruimte. De variabelen u_i zijn onderling onafhankelijk (ongecorreleerd).

Vaak zijn stochasten onderling gecorreleerd. In dergelijke gevallen kunnen gecorreleerde variabelen worden getransformeerd in ongecorreleerde variabelen. Hiervoor zijn verschillende methoden beschikbaar. Het meest bekent is de zogenaamde Rosenblatt transformatie [13]. Andere methoden zijn Nataf, Hermite en Johnson.

Rosenblatt transformatie
In het geval van een Rosenblatt transformatie worden de afhankelijke variabelen achtereenvolgens getransformeerd in onafhankelijke variabelen volgens:

$$\begin{aligned}\Phi(u_1) &= F_{x_1}(\xi_1)\\ \Phi(u_2) &= F_{x_2|x_1}(\xi_2 \mid \xi_1)\\ \Phi(u_3) &= F_{x_3|x_2,x_1}(\xi_3 \mid \xi_2, \xi_1)\end{aligned}$$

Deze transformatie kan voor normaal verdeelde gecorreleerde variabelen eenvoudig worden uitgevoerd met de eigenwaarden en eigenvectoren van de covariantie matrix Γ [14].

$$\Gamma = W^T \Lambda W$$

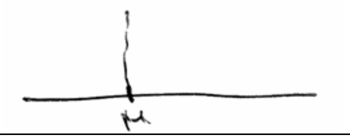
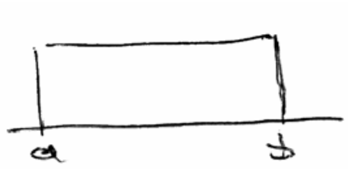
waarin:
Λ is de diagonaal matrix met eigenwaarden van de covariantie matrix Γ
W is de matrix met eigenvectoren van de covariantie matrix Γ

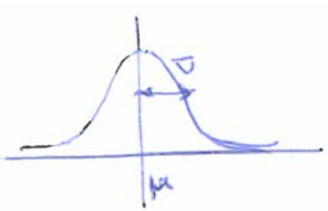
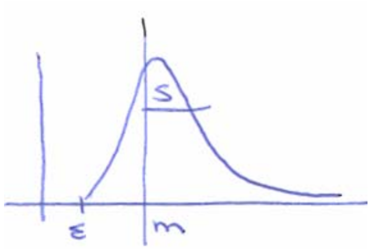
De vector $\underline{u}$ met ongecorreleerde variabelen kan nu worden getransformeerd naar een vector $\underline{u}^*$ met gecorreleerde variabelen met vergelijking:

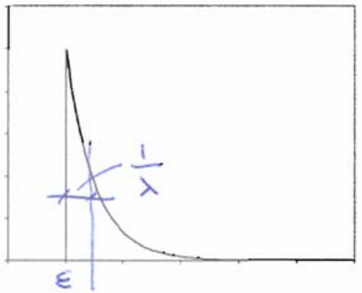
$$\underline{u}^* = \mathrm{W}\ \sqrt{\Lambda}\ \underline{u}$$

Opmerkingen:
- In principe zijn er n! verschillende transformaties mogelijke, afhankelijk van de volgorde van de eigenvectoren. In het geval van niet normaal verdeelde variabelen kan iedere volgorde aanleiding geven tot verschillende schattingen van de faalkans.
- Door de specifieke eigenschappen van eigenwaarden en eigenvectoren kan bij berekeningen het teken en de volgorde van de invloedscoëfficiënten zijn verwisseld.
- De covariantie matrix betreft de covariantie matrix voor de standaard normaal verdeelde variabelen u, waarnaar andere verdelingstypen worden getransformeerd.

B Distribution types

Type	Distribution	Probability density function and Cumulative distribution	Parameters	Moments
0	Deterministic 	$x < \mu$: $f_x(x) = 0$ $x > \mu$: $f_x(x) = 0$ $x < \mu$: $F_x(x) = 0$ $x > \mu$: $F_x(x) = 1$	$1 = \mu$	$m = \mu$ $s = 0$
1	Uniform 	$a \leq x \leq b$: $f_x(x) = \dfrac{1}{b-a}$ $F_x(x) = \dfrac{x-a}{b-a}$	$1 = a$ $2 = b$ $a < b$	$m = \dfrac{a+b}{2}$ $s = \dfrac{b-a}{\sqrt{12}}$

Type	Distribution	Probability density function and Cumulative distribution	Parameters	Moments
2	Normal/Gaussian  	$-\infty < x < \infty$: $f_x(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$ $F_x(x) = \int_{-\infty}^{x} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} dx$	1 = m 2 = s $\sigma > 0$	$m = \mu$ $s = \sigma$
3	Shifted lognormal  	$\varepsilon < x < \infty$: $f_x(x) = \frac{1}{(x-\varepsilon)\zeta\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\ln(x-\varepsilon)-\lambda}{\zeta}\right)^2}$ $F_x(x) = \int_{-\infty}^{x} \frac{1}{(x-\varepsilon)\zeta\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\ln(x-\varepsilon)-\lambda}{\zeta}\right)^2} dx$	$1 = \lambda$ $2 = \zeta$ $3 = \varepsilon$ $m > \varepsilon$ $s > 0$	$m = \varepsilon + e^{\left(\lambda + \frac{\zeta^2}{2}\right)}$ $s = e^{\left(\lambda + \frac{\zeta^2}{2}\right)} \sqrt{e^{\zeta^2} - 1}$

Type	Distribution	Probability density function and Cumulative distribution	Parameters	Moments
4	Shifted exponential  	ε < x < ∞: $f_x(x) = \lambda\, e^{-\lambda(x-\varepsilon)}$ $F_x(x) = 1 - e^{-\lambda(x-\varepsilon)}$	1 = λ 2 = ε λ > 0	$m = \varepsilon + \frac{1}{\lambda}$ $s = \frac{1}{\lambda}$
5	Shifted gamma	ε ≤ x < ∞: $f_x(x) = \frac{b^p}{\Gamma(p)}(x-\varepsilon)^{p-1} e^{-b(x-\varepsilon)}$ $F_x(x) = \int_{\varepsilon}^{x} \frac{b^p}{\Gamma(p)}(x-\varepsilon)^{p-1} e^{-b(x-\varepsilon)} dx$ $F_x(x) = \frac{\Gamma(b(x-\varepsilon), p)}{\Gamma(p)}$	1 = p 2 = b 3 = ε p > 0: b > 0	$m = \varepsilon + \frac{p}{b}$ $s = \frac{\sqrt{p}}{b}$

Type	Distribution	Probability density function and Cumulative distribution	Parameters	Moments

6	Beta	$a \le x \le b$:
		$f_x(x) = \dfrac{\left(\dfrac{x-a}{b-a}\right)^{r-1}\left(1-\dfrac{x-a}{b-a}\right)^{t-1}}{(b-a)\,B(r,t)}$
		$f_x(x) = \dfrac{(x-a)^{r-1}(b-x)^{t-1}}{(b-a)^{r+t-1}B(r,t)}$
		$F_x(x) = \dfrac{B\left(\dfrac{x-a}{b-a},r,t\right)}{B(r,t)}$
			1 = a
			2 = b
			3 = r
			4 = t
			$a < b$
			$r > 0$
			$t > 0$
				m = $a+(b-a)\dfrac{r}{r+t}$
				s = $(b-a)\sqrt{\dfrac{r\,t}{(r+t)^2\,(r+t+1)}}$

7	Gumbel
	(max. type I)
		$-\infty < x < \infty$:
		$f_x(x) = \alpha\, e^{-\alpha(x-u)-e^{-\alpha(x-u)}}$
		$F_x(x) = e^{-e^{-\alpha(x-u)}}$
			1 = u
			2 = α
			$\alpha > 0$
				m = $u+\dfrac{0.577216}{\alpha}$
				s = $\dfrac{\pi}{\alpha\sqrt{6}}$

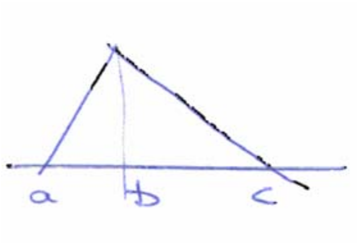
Type	Distribution	Probability density function and Cumulative distribution	Parameters	Moments

8	Fréchet
	(max. type II)
		$\varepsilon < x < \infty$:
		$f_x(x) = \dfrac{k}{x-\varepsilon}\left(\dfrac{u-\varepsilon}{x-\varepsilon}\right)^k e^{-\left(\frac{u-\varepsilon}{x-\varepsilon}\right)^k}$
		$F_x(x) = e^{-\left(\frac{u-\varepsilon}{x-\varepsilon}\right)^k}$
			1 = u
			2 = k
			3 = ε
			u > ε
			k > 0
				k > 1:
				m = $\varepsilon + (u-\varepsilon)\Gamma\left(1-\dfrac{1}{k}\right)$
				k > 2:
				s = $(u-\varepsilon)\sqrt{\Gamma\left(1-\dfrac{2}{k}\right)-\Gamma^2\left(1-\dfrac{1}{k}\right)}$

9	Weibull
	(min. type III)
		$\varepsilon \le x < \infty$:
		$f_x(x) = \dfrac{k}{u-\varepsilon}\left(\dfrac{x-\varepsilon}{u-\varepsilon}\right)^{k-1} e^{-\left(\frac{x-\varepsilon}{u-\varepsilon}\right)^k}$
		$F_x(x) = 1-e^{-\left(\frac{x-\varepsilon}{u-\varepsilon}\right)^k}$
			1 = u
			2 = k
			3 = ε
			u > ε
			k > 0
				m = $\varepsilon + (u-\varepsilon)\Gamma\left(1+\dfrac{1}{k}\right)$
				s = $(u-\varepsilon)\sqrt{\Gamma\left(1+\dfrac{2}{k}\right)-\Gamma^2\left(1+\dfrac{1}{k}\right)}$

10	Rayleigh	$\varepsilon \le x < \infty$:
		$f_x(x) = \dfrac{x-\varepsilon}{\alpha^2} e^{-\frac{(x-\varepsilon)^2}{2\alpha^2}}$
		$F_x(x) = 1-e^{-\frac{(x-\varepsilon)^2}{2\alpha^2}}$
			1 = α
			2 = ε
			α > 0
				m = $\varepsilon + \alpha\sqrt{\dfrac{\pi}{2}}$
				s = $\alpha\sqrt{2-\dfrac{\pi}{2}}$

Type	Distribution	Probability density function and Cumulative distribution	Parameters	Moments
11	Pareto	$u \leq x < \infty$: $f_x(x) = \frac{0.5}{\sigma}\left(1+\gamma\frac{(x-u)}{\sigma}\right)^{-\frac{1}{\gamma}-1}$ $F_x(x) = 1-0.5\left(1+\gamma\frac{(x-u)}{\sigma}\right)^{-\frac{1}{\gamma}}$	1 = u 2 = σ 3 = γ	
12	Student	$-\infty < x < \infty$: $f_x(x) = \frac{1}{\sqrt{\nu\pi}}\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sigma\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{\left(\frac{x-\mu}{\sigma}\right)^2}{\nu}\right)^{-\frac{\nu+1}{2}}$ $F_x(x) = \int_{-\infty}^{x}\frac{1}{\sqrt{\nu\pi}}\frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sigma\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{\left(\frac{x-\mu}{\sigma}\right)^2}{\nu}\right)^{-\frac{\nu+1}{2}} dx$	1 = ν 2 = μ 3 = σ $\nu \geq 1$ $\sigma > 0$	m = μ $\nu \geq 2$: $s = \sigma\sqrt{\frac{\nu}{\nu-2}}$

Type	Distribution	Probability density function and Cumulative distribution	Parameters	Moments
13	Triangle 	a ≤ x ≤ c: a ≤ x < b: $f_x(x) = \frac{2(x-a)}{(b-a)(c-a)}$ b ≤ x ≤ c: $f_x(x) = \frac{2(c-x)}{(c-b)(c-a)}$ a ≤ x < b: $F_x(x) = \frac{(x-a)^2}{(b-a)(c-a)}$ b ≤ x ≤ c: $F_x(x) = \frac{b-a}{c-a} + \frac{x(2c-x)+b(b-2c)}{(c-b)(c-a)}$	1 = a 2 = b 3 = c a ≤ b ≤ c	$m = \frac{a+b+c}{3}$ $s^2 = \frac{a^2+b^2+c^2-ab-ac-bc}{18}$
20	x-u table	Tabulated values of realisations with corresponding values for the realisations in u space (standard normal). The table is ordered from small u-values to larger u-values		
21	x-q table	Tabulated values of realisations with corresponding complementary cumulative distribution function values (1-p) The table is ordered from large (1-p)-values to small (1-p) -values		
22	x-p table	Tabulated values of realisations with corresponding cumulative distribution function values (p). The table is ordered from small p-values to larger p-values		

Type	Distribution	Probability density function and Cumulative distribution	Parameters	Moments
30	Discrete realisations	Table with discrete probabilities for listed events. Only allowed in combination with Crude Monte Carlo or Numerical Integration method. Influence factors α are no longer representative for the design point.		

C Files containing tabulated stochastic properties for input parameters

Besides selecting a distribution function, Prob2B can also read stochastic properties for a parameter from a user-supplied file. Four types of tabulated values with corresponding files are accepted.

x-p table:
Tabulated values of realisations with corresponding cumulative distribution function values (p). Files need to have the extension ‘.xpt’

x-q table:
Tabulated values of realisations with corresponding complementary cumulative distribution function values (1-p). Files need to have the extension ‘.xqt’

x-u table:
Tabulated values of realisations with corresponding values for the realisations in u space (standard normal). Files need to have the extension ‘.xut’

discrete table:
Tabulated values of realisations with corresponding probability of occurrence. Files need to have the extension ‘.dis’

C.1 ‘.xpt’ file

Example:

```
%       two columns per line
%       tab and or space -delimited
%       empty and comment lines are allowed
%       comment lines start with '%' (in position 1)
%       both columns are ordered and non-decreasing
%       X                       P
4.45000000E-06    1.90363988E-08
4.47500000E-06    7.62013579E-08
4.50000000E-06    2.87105000E-07
4.52500000E-06    1.01832850E-06
4.55000000E-06    3.40080306E-06
4.57500000E-06    1.06956857E-05
4.60000000E-06    3.16860346E-05
..
 (lines deliberately omitted in this listing)
..
5.42500000E-06    9.99989304E-01
5.45000000E-06    9.99996599E-01
5.47500000E-06    9.99998982E-01
5.50000000E-06    9.99999713E-01
5.52500000E-06    9.99999924E-01
5.55000000E-06    9.99999981E-01
```

C.2 ‘.xqt’ file

Example:

```
%	two columns per line
%	tab and or space -delimited
%	empty and comment lines are allowed
%	comment lines start with '%' (at position 1)
%	first column is ordered and non-decreasing
%	second column is non-increasing
%	X			1-P
4.45000000E-06	9.99999981E-01
4.47500000E-06	9.99999924E-01
4.50000000E-06	9.99999713E-01
4.52500000E-06	9.99998982E-01
4.55000000E-06	9.99996599E-01
4.57500000E-06	9.99989304E-01
4.60000000E-06	9.99968314E-01
..
 (lines deliberately omitted in this listing)
..
5.42500000E-06	1.06956857E-05
5.45000000E-06	3.40080306E-06
5.47500000E-06	1.01832850E-06
5.50000000E-06	2.87105000E-07
5.52500000E-06	7.62013579E-08
5.55000000E-06	1.90363988E-08
```

C.3 ‘.xut’ file

Example:

```
%       two columns per line
%       tab and or space -delimited
%       empty and comment lines are allowed
%       comment lines start with '%' (at position 1)
%       both columns ordered and non-decreasing
%       X                     U
4.45000000E-06     -5.50000000E+00
4.47500000E-06     -5.25000000E+00
4.50000000E-06     -5.00000000E+00
4.52500000E-06     -4.75000000E+00
4.55000000E-06     -4.50000000E+00
4.57500000E-06     -4.25000000E+00
4.60000000E-06     -4.00000000E+00
..
 (lines deliberately omitted in this listing)
..
5.42500000E-06     4.25000000E+00
5.45000000E-06     4.50000000E+00
5.47500000E-06     4.75000000E+00
5.50000000E-06     5.00000000E+00
5.52500000E-06     5.25000000E+00
5.55000000E-06     5.50000000E+00
```

C.4 ‘.dis’ file

Example:

```
%       two columns per line
%       tab and or space -delimited
%       empty and comment lines are allowed
%       comment lines start with '%' (at position 1)
%       no prescribed ordering
%       X                       f_discr.
4.45000000E-06     2.90758287E-08
4.47500000E-06     1.10657417E-07
4.50000000E-06     3.95767933E-07
4.52500000E-06     1.33018700E-06
4.55000000E-06     4.20142707E-06
4.57500000E-06     1.24707513E-05
4.60000000E-06     3.47855829E-05
..
 (lines deliberately omitted in this listing)
..
5.42500000E-06     1.24707513E-05
5.45000000E-06     4.20142707E-06
5.47500000E-06     1.33018700E-06
5.50000000E-06     3.95767933E-07
5.52500000E-06     1.10657417E-07
5.55000000E-06     2.90758287E-08
```

D Creating an Excel® model to be addressed by Prob2B

In this Appendix it is described how an Excel® spreadsheet can be adapted such that it can be addressed by Prob2B. The procedure would normally be as follows:

- There is a stand alone work-book that makes a model calculation,
- With this workbook as starting point, adaptations are made for interfacing it with Prob2B.

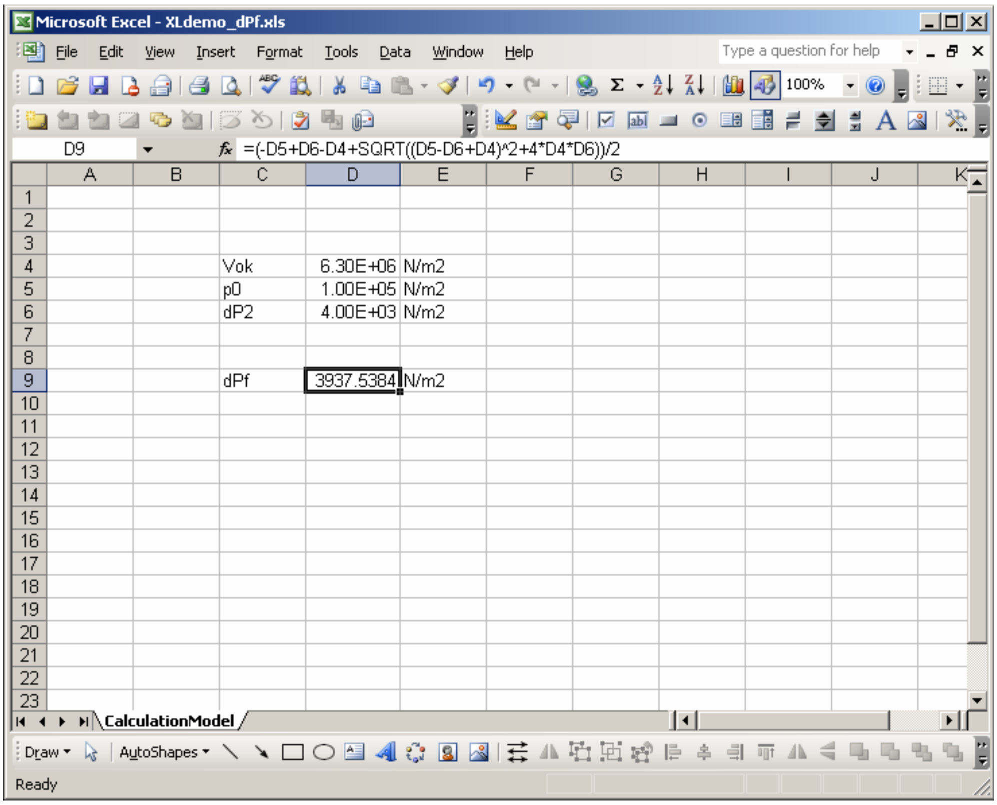
The example used in this appendix is the one that is used in Chapter 6 en Chapter 7 of the main text. Please, take note of section D.3.

D.1 The stand-alone workbook

We create a workbook containing a sheet that is able to calculate the following function:

$$\Delta p_F = \frac{1}{2}\left(-p_0 + \Delta p_2 - Vok + \sqrt{(p_0 - \Delta p_2 + Vok)^2 + 4(Vok)\Delta p_2}\right)$$

This can be done straightforward by putting it as a cell dependent equation. The example is presented in Figure 87


Microsoft Excel - XLdemo_dPf.xls
File Edit View Insert Format Tools Data Window Help
Type a question for help
100%
CalculationModel
Draw AutoShapes
Ready

D9 =(-D5+D6-D4+SQRT((D5-D6+D4)^2+4*D4*D6))/2

	C	D	E
4	Vok	6.30E+06	N/m2
5	p0	1.00E+05	N/m2
6	dP2	4.00E+03	N/m2
9	dPf	3937.5384	N/m2

Figure 87 Excel® sheet, starting point

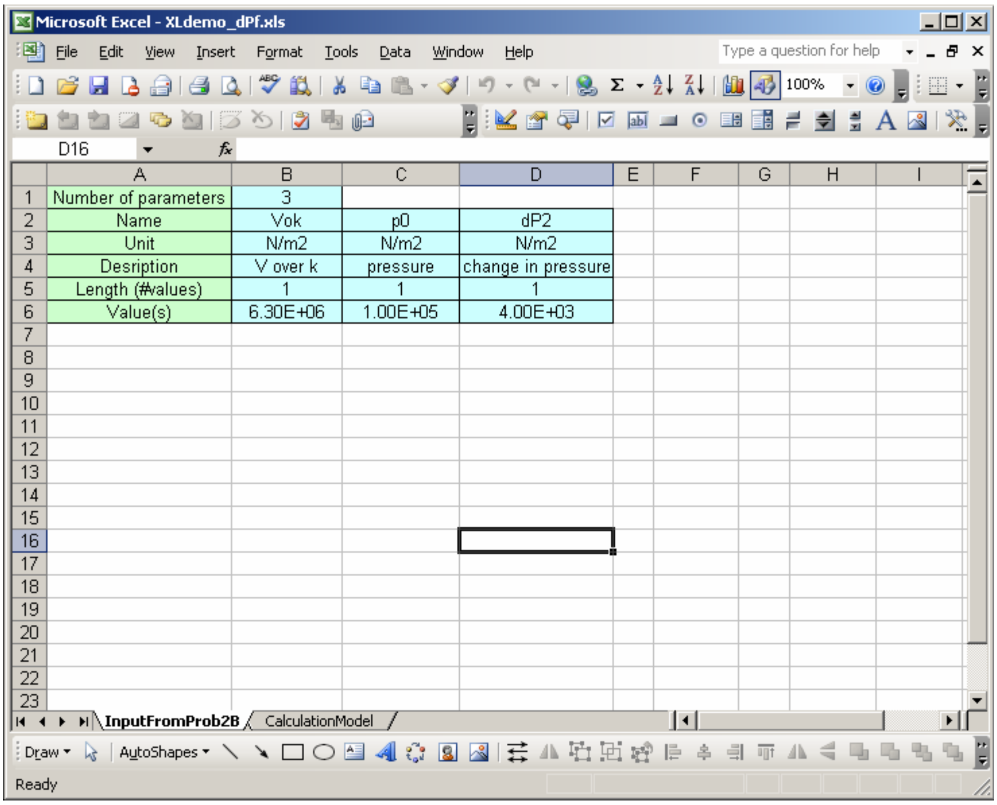
We see the equation implemented in cell D9, dependent on the cells D4 to D6 which have just given a value.

D.2 Interfacing with Prob2B

For interfacing with Prob2B the following actions are taken:

- A worksheet is added for setting the input and made interactive with the sheet(s) that make the calculation.
- A worksheet is added for reading the output and made interactive with the sheet(s) that make the calculation.
- A macro is added, that will be called by Prob2B to make a forward calculation.
- The security levels are adjusted in order to let Excel® workbook be assessed

D.2.1 Adding a worksheet for input



	A	B	C	D
1	Number of parameters	3		
2	Name	Vok	p0	dP2
3	Unit	N/m2	N/m2	N/m2
4	Desription	V over k	pressure	change in pressure
5	Length (#values)	1	1	1
6	Value(s)	6.30E+06	1.00E+05	4.00E+03

Figure 88 Excel® sheet for interfacing input values added

The worksheet shown in Figure 88 is added. A number of properties for this sheet are mandatory.

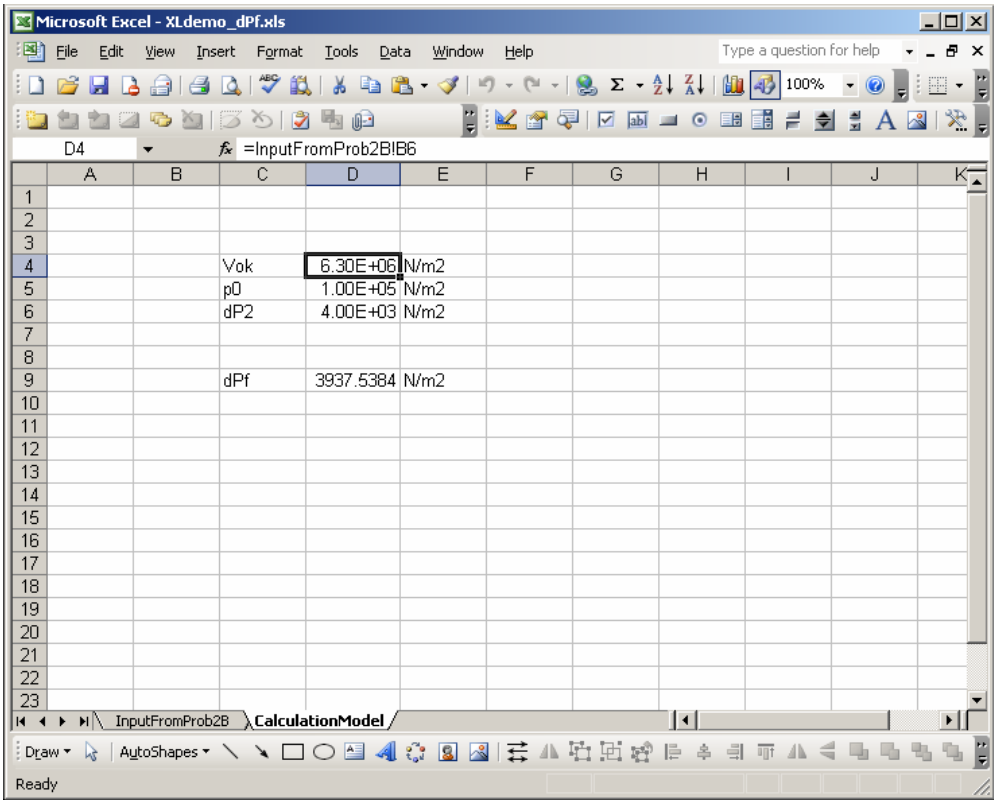
First of all the sheets name has to be “**InputFromProb2B**”. Prob2B will be looking for this sheet and when it can not be found, an error will occur.

Column A will not be used by Prob2B, it is however now filled with some comments about the meaning of the values in the corresponding rows.

First item to be interpreted by Prob2B will be the value in cell ‘B1’. It refers to the number of input variables that play a role in the dialog between the Excel® model and Prob2B. In this example the value is 3. As a consequence Prob2B will scan the columns B to D for properties of the input variables. The cell values B2 to D6 are now expected to be non-empty. For each parameter the first 3 rows (row 2 to 4) contain respectively the name, unit and description that will be prompted within Prob2B. The name is of course essential. Unit and description can be filled with for instance ‘-’ if the user doesn’t bother much about it.

Row 5 has to be filled with the dimension size of the variable. A value 1 refers to a scalar, a value N refers to a vector of length N. Finally, on row 6 the actual value of the variable is given. The first time Prob2b interprets the sheet the value is red as default value for the variable. (During calculation runs Prob2B will update these cells with sample values before giving the command to calculate).

Making the input sheet “InputFromProb2B” interact with the calculation sheet(s) is achieved by cell reference as shown in Figure 89. Here we see that the value in cell D4 for ‘Vok’ is updated with the value from the input sheet. The same is done for ‘p0’ and ‘dP2’.

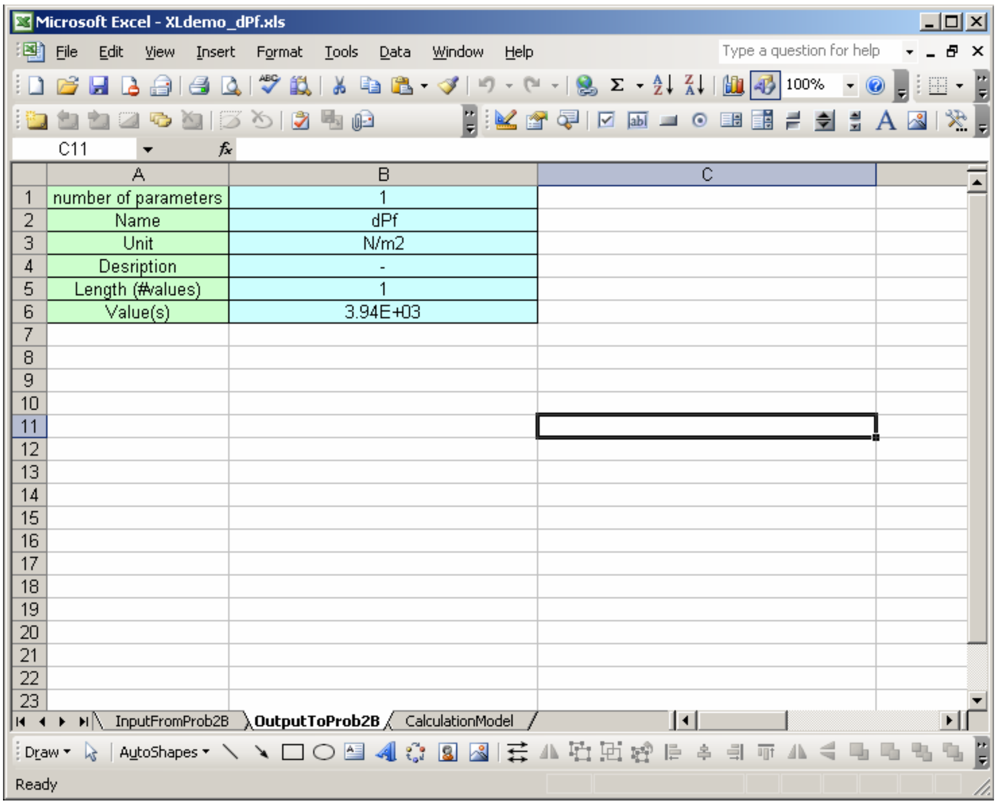


D4 =InputFromProb2B!B6

	C	D	E
4	Vok	6.30E+06	N/m2
5	p0	1.00E+05	N/m2
6	dP2	4.00E+03	N/m2
9	dPf	3937.5384	N/m2

Figure 89 Interacting of model sheet(s) with input sheet by cell reference

D.2.2 Adding a worksheet for output

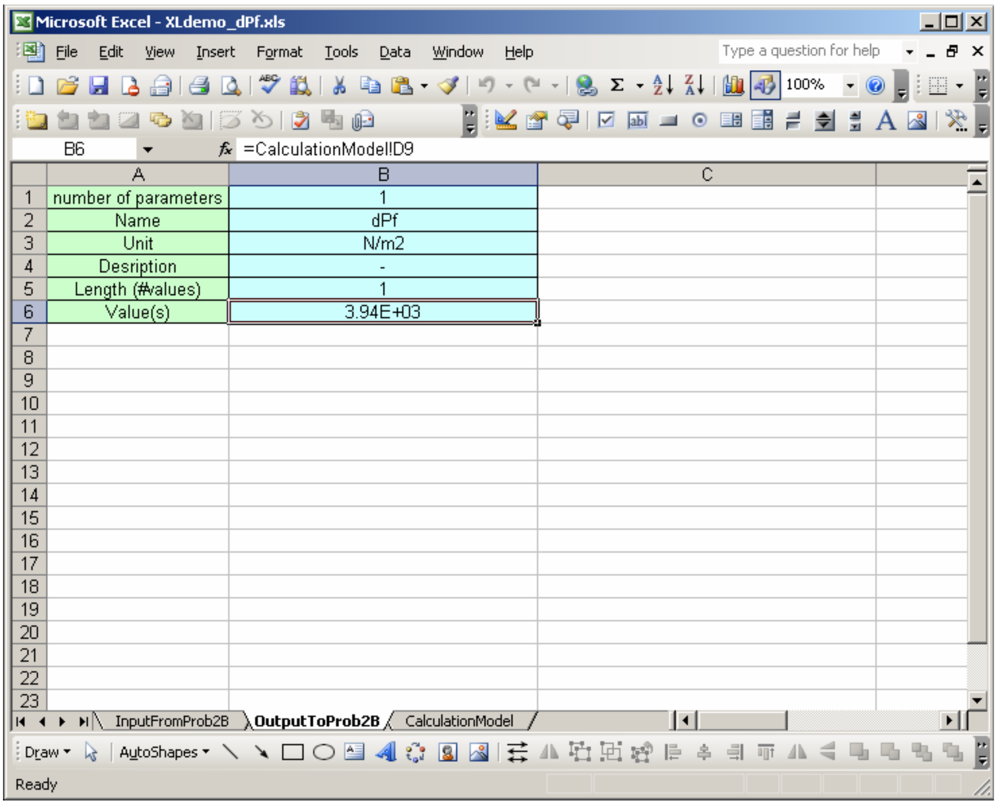


	A	B
1	number of parameters	1
2	Name	dPf
3	Unit	N/m2
4	Desription	-
5	Length (#values)	1
6	Value(s)	3.94E+03

Figure 90 Excel® sheet for interfacing output values

For output variables a comparable procedure is followed as in D.2.1.
Now, the worksheet shown in Figure 90 is added. Again, a number of properties for this sheet are mandatory.
First of all the sheets name has to be “**OutPutToProb2B**”. Prob2B will be looking for this sheet and when it can not be found, an error will occur.
Column A will not be used by Prob2B. It is however now filled with some comments about the meaning of the values in the corresponding rows.
First item to be interpreted by Prob2B will be the value in cell ‘B1’. It refers to the number of output variables that play a role in the dialog between the Excel® model and Prob2B. In this example the value is 1. As a consequence Prob2B will scan column B for properties of the output variables. The cell values B2 to B6 are now expected to be non-empty. For each parameter the first 3 rows (row 2 to 4) contain respectively the name, unit and description that will be prompted within Prob2B. The name is of course essential. Unit and description can be filled with for instance ‘-’ if the user doesn’t bother much about it.
Row 5 has to be filled with the dimension size of the variable. A value 1 refers to a scalar, a value N refers to a vector of length N. Finally, on row 6 the actual value of the variable will be red by Prob2B

Making the output sheet “OutputToProb2B” interact with the calculation sheet(s) is achieved by cell reference as shown in Figure 91. The actual value for dPf is updated by the calculated value in the calculation sheet(s).



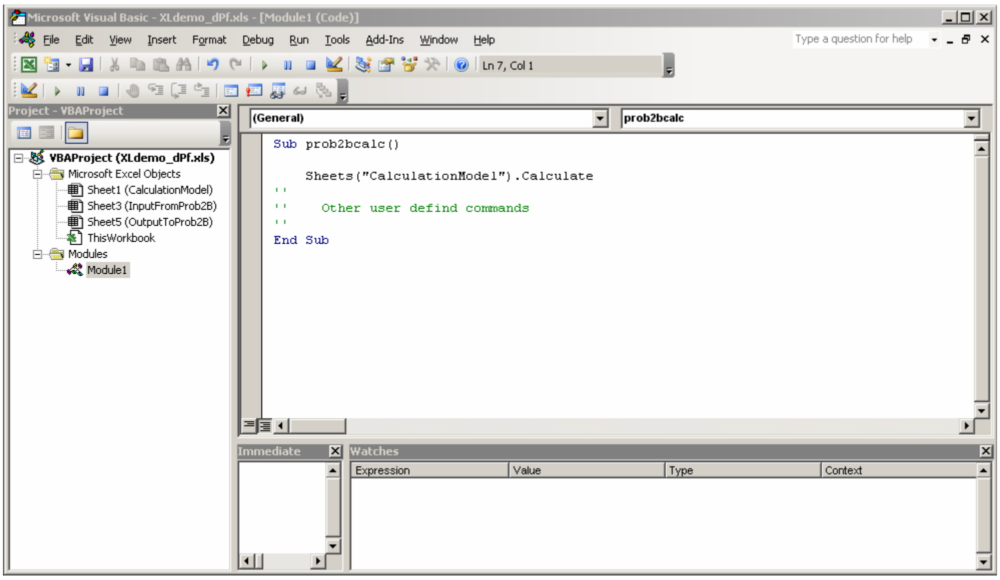
B6 =CalculationModel!D9

	A	B
1	number of parameters	1
2	Name	dPf
3	Unit	N/m2
4	Desription	-
5	Length (#values)	1
6	Value(s)	3.94E+03

Figure 91 Interacting of output sheet with model sheet(s) by cell reference

D.2.3 Adding the macro ‘prob2bcalc’

When addressing the Excel® model for calculations, Prob2B will first put the sample values for the calculation in the corresponding cells of the sheet “InputFromProb2B”.
One could then rely on the automatic updates of the calculation sheet(s), but in order to make the calculation possibilities more versatile, use will be made of running a macro to force a calculation.
This macro has to be called “**prob2bcalc**” (no capital characters) and the user is free to write his content with commands to perform the calculations. This could be just an update of the calculation sheet(s), but also calls to other functions and subroutines are possible from within this macro.
Besides the mandatory name “prob2bcalc”, also the macro has to be present in “**Module1**” of the VB modules.
Figure 92shows the (simple) coding for the macro used in our example.

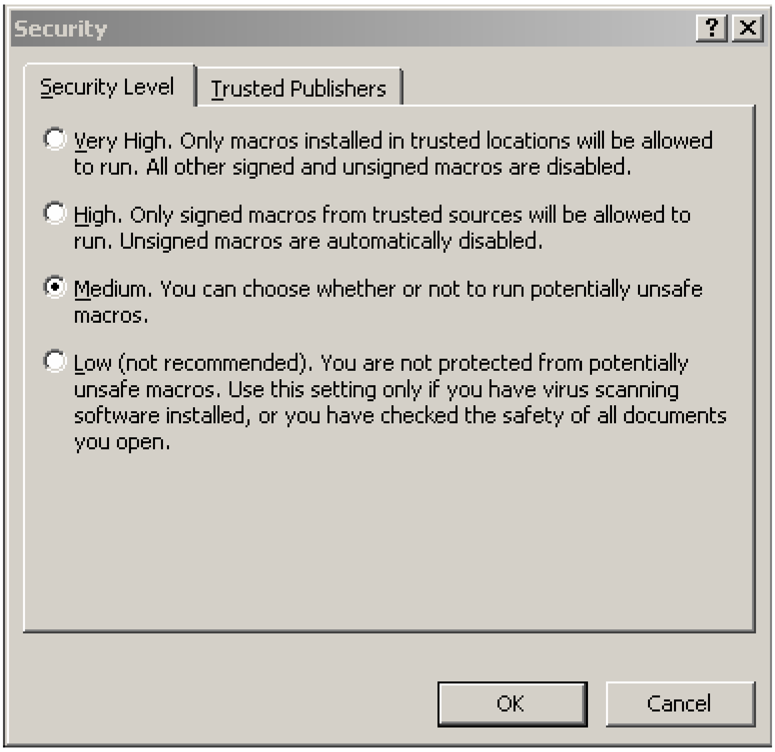


```
Sub prob2bcalc()

    Sheets("CalculationModel").Calculate
''
''    Other user defind commands
''
End Sub
```

Figure 92 Coding example of the macro ‘prob2bcalc’

D.2.4 *Setting the security levels*

To make the Excel® workbook accessible to Prob2B, security levels have to be set. These security settings can be found under Tools→ Macro→ Security.
Set them to the following (see remark under footnote 1):



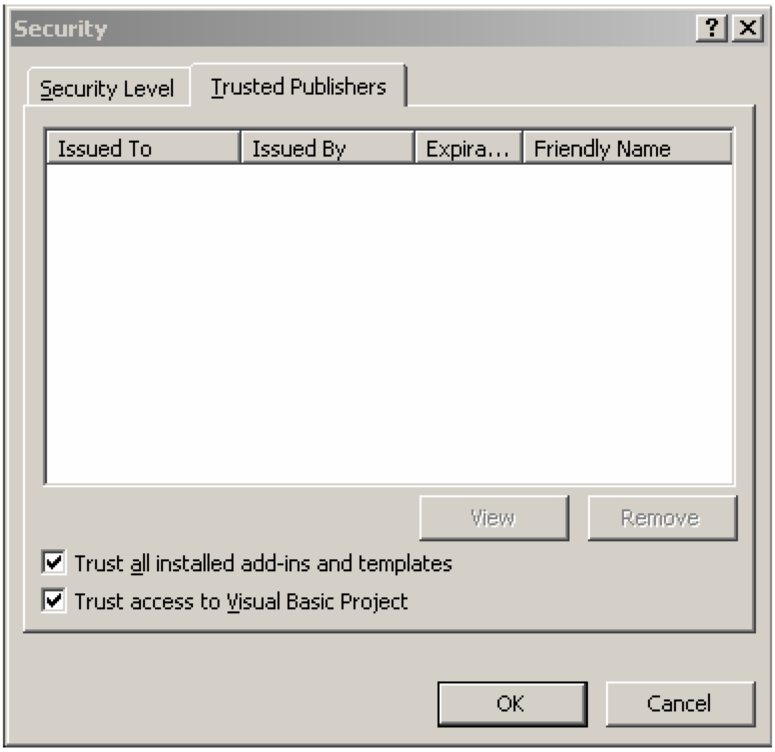
○ Very High. Only macros installed in trusted locations will be allowed to run. All other signed and unsigned macros are disabled.

○ High. Only signed macros from trusted sources will be allowed to run. Unsigned macros are automatically disabled.

● Medium. You can choose whether or not to run potentially unsafe macros.

○ Low (not recommended). You are not protected from potentially unsafe macros. Use this setting only if you have virus scanning software installed, or you have checked the safety of all documents you open.

Figure 93 Setting for Security level, see remark under footnote 1



☑ Trust all installed add-ins and templates

☑ Trust access to Visual Basic Project

Figure 94 Settings for trusted publishers, see remark under footnote 1.

Footnote 1

It is stressed that it is the users own responsibility whether he accepts the risks involved in lowering the security settings and increasing the accessibility.

D.3 Installing the Prob2B interface to Excel®

Prob2B uses an interface to communicate with Excel®. In order to be used it has to be installed (once). This can be done by the following:
Go to the directory:

C:\Program Files\Prob2B\reglib\Excel®

And run the executable ‘setup.exe’.

D.4 Running Prob2B with Excel®

Please close all Excel® sessions while you are running Prob2B in combination with Excel®.