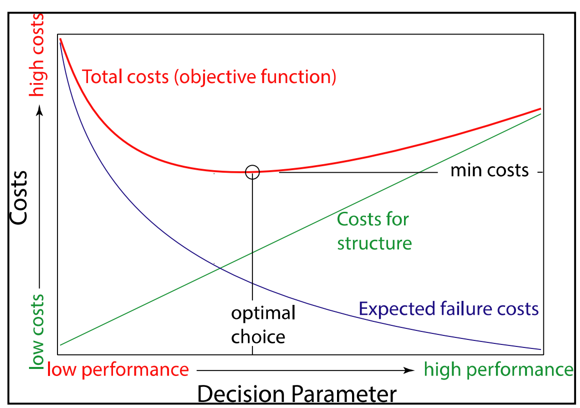
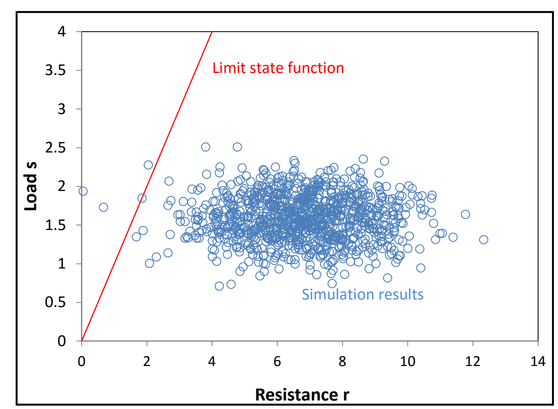
NTNU
Kunnskap for en bedre verden

Structural reliability

Jochen Köhler

7.11.2017





Limit state function

$$g(h) = f_m \frac{bh^2}{6} - \frac{Gl}{4} = 0$$

$$g(X) = R - S$$

Limit state functions are not restricted to civil engineering. The concept of limit state functions **is generic** and can be transferred to a large family of problems.

Limit state function

$$g(h) = f_m \frac{bh^2}{6} - \frac{Gl}{4} = 0$$

$$g(X) = R - S$$

The parameters considered in the limit state function are subjected to **uncertainties**.
How can we include these uncertainties for a «safe» design?

Reliability theory in civil engineering

1926 Max Mayer (German engineer) realized the significance of uncertainties on safety.

A purely empirical adjustment of the physical best practice rules with factors (global safety factors) can lead to dangerous contradictions.

Max Mayer called for an assessment of structural safety based on **probability theory**.

He recognized that the dispersion and the functional relation between the parameters is of importance for the safety.
→ But what can be deemed to be safe?

Safe enough?

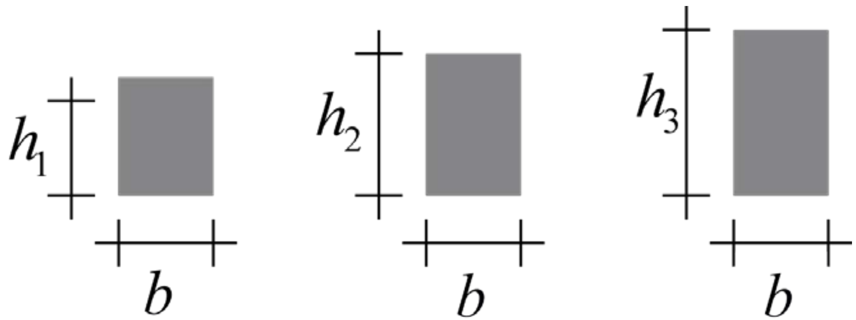
1924 Carl Forssell (Swedish Professor) proposed to assess safety by minimizing the total costs of the building.
(Ekonomi och Byggnadsvasen).

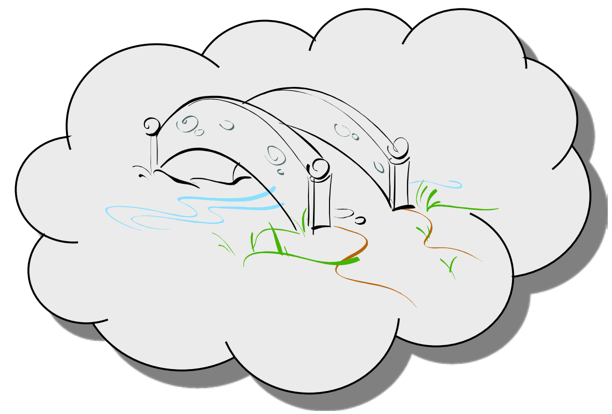
Minimize (Total Costs)=

Minimize (Costs of the structure +
Probability of failure*Costs of failure)

Risik based concept!☺

3. Risk based assessment of the cross section

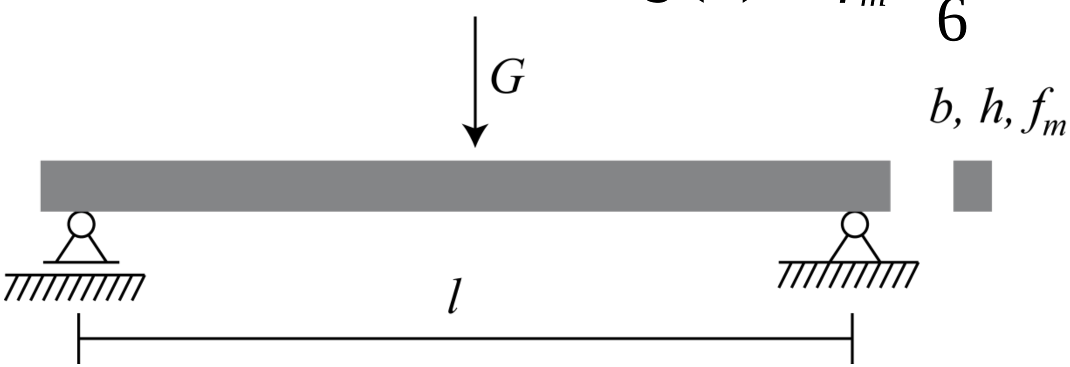




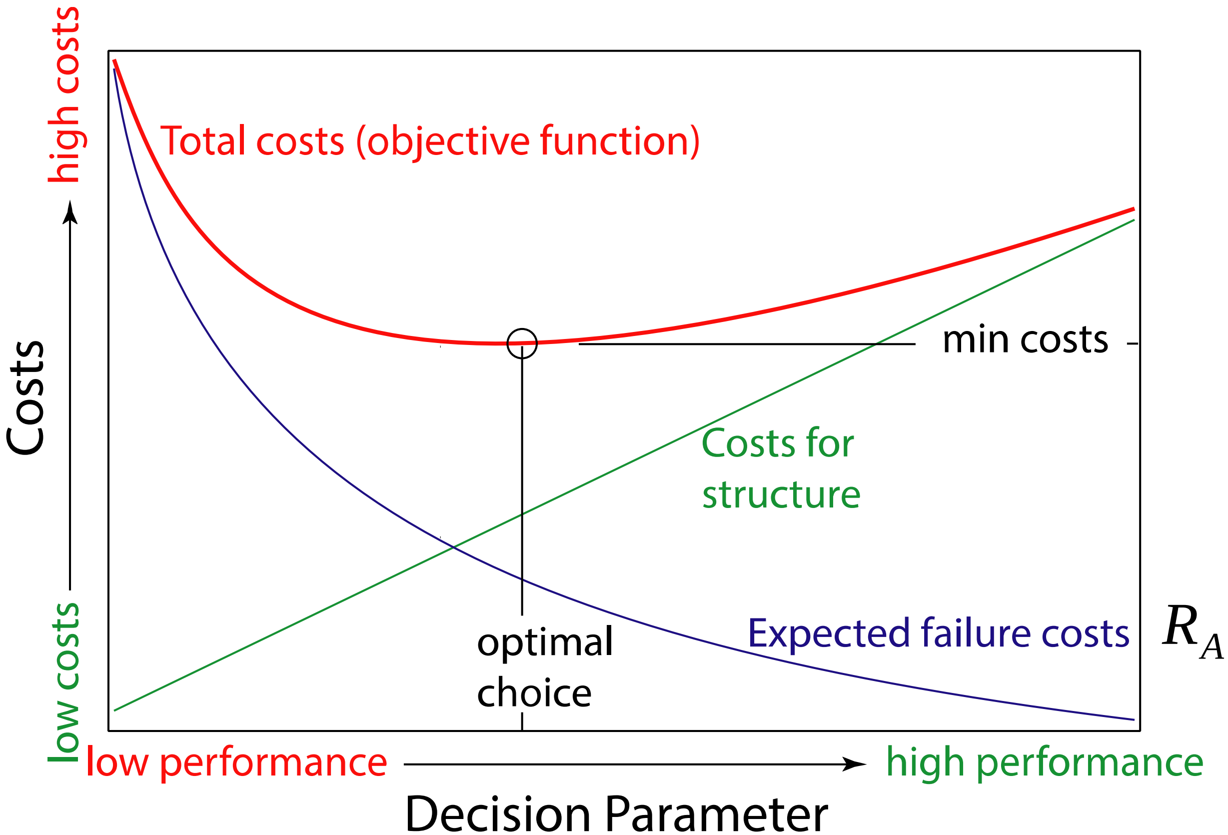
$$f_m \frac{bh^2}{6} \geq \frac{Gl}{4} \quad [kNm]$$

Limit state function

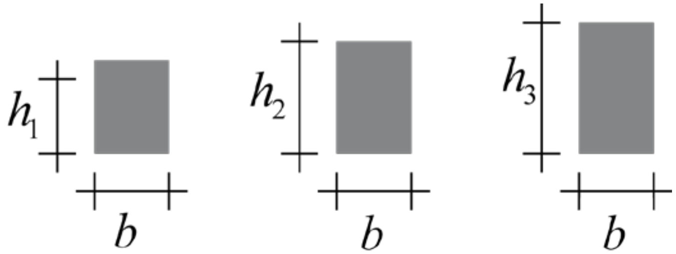
$$g(h) = f_m \frac{bh^2}{6} - \frac{Gl}{4} = 0$$



3. Risk based assessment of the cross section



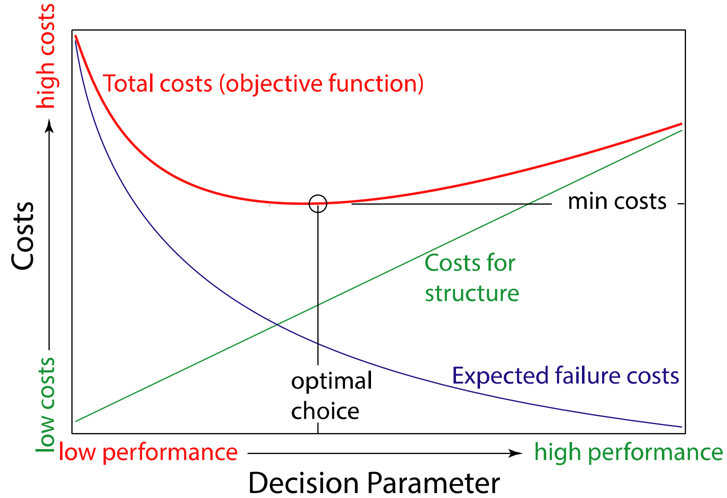
$$R_A = \sum_{i=1}^{n_E} R_{E_i} = \sum_{i=1}^{n_E} P_{E_i} \cdot C_{E_i}$$



Safe enough? – Risk based concept needed.

1. A explicit **concept** for calculating the **probability of failure** of structures is needed.

2. A **concept** for assessment of **costs** due to structural failure is needed. (Fatalities, Injuries, direct costs, indirect costs).



Concept for calculating the **probability of failure**

- Load and resistance can be represented by **random variables**

Resistance > Load

$$X_{f_m} \frac{bh^2}{6} \geq \frac{X_G l}{4} \quad [kNm]$$

$$P_F = \Pr(R \leq S) \; ; P_F \in [0,1]$$ **Probability of failure**

$$P_s = 1 - P_F$$ **Reliability**

Safe enough? – Risk based concept needed.

Reliability

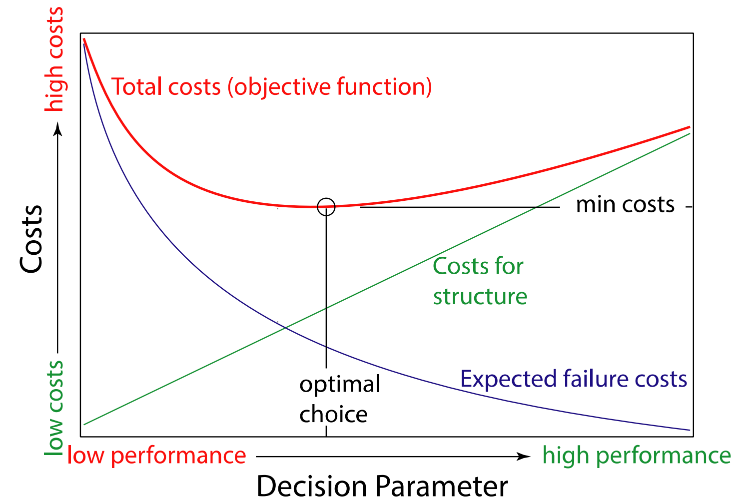
Risk

$$E[B] = I(1 - P_F) - C_D - C_F P_F$$

Benefit in use

Costs for the structure

Costs due to failure



$$\Rightarrow \frac{\partial E[B]}{\partial C_D} = 0$$

But how is the failure probability estimated ??

If *R* und *S* are normal distributed so is the safety margin ***M*** also normal distributed.

$$M = R - S$$

The probability of safety is:

$$P_F = P(R - S \leq 0) = P(M \leq 0)$$

mean value and *standard de*viation of M is:

$$\mu_M = \mu_R - \mu_S$$

$$\sigma_M = \sqrt{\sigma_R^2 + \sigma_S^2}$$

The probability of safety is thus:

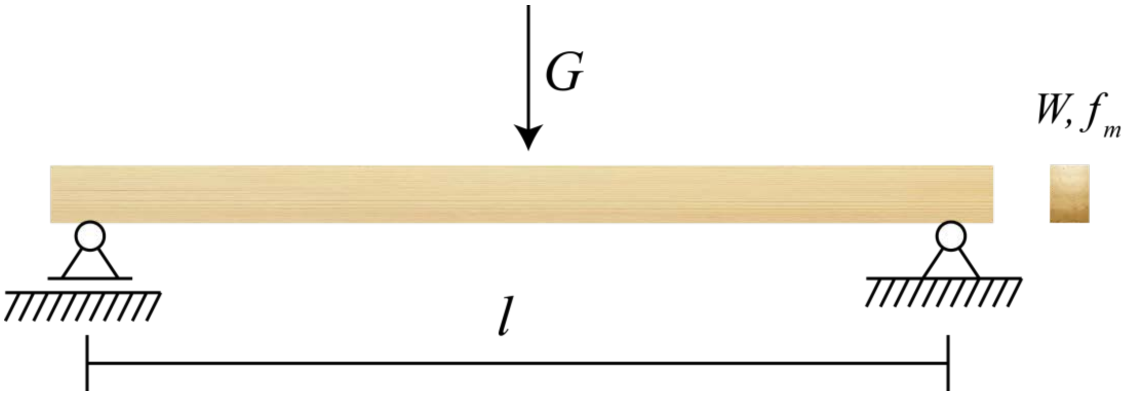
$$P_F = \Phi\left(\frac{0 - \mu_M}{\sigma_M}\right) = \Phi(-\beta)$$

and the reliability index is:

$$\beta = \mu_M / \sigma_M$$

Failure:

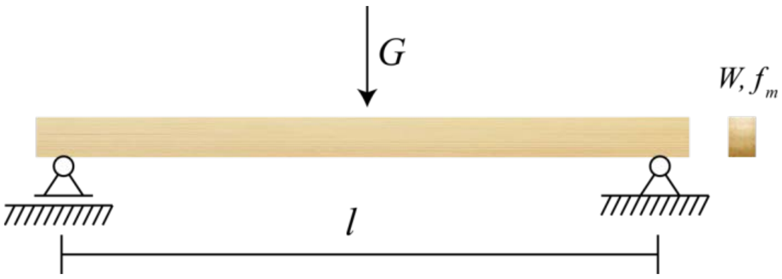
..back to the introductive example:



$$f_m W = r < s = \frac{G\,l}{4}$$

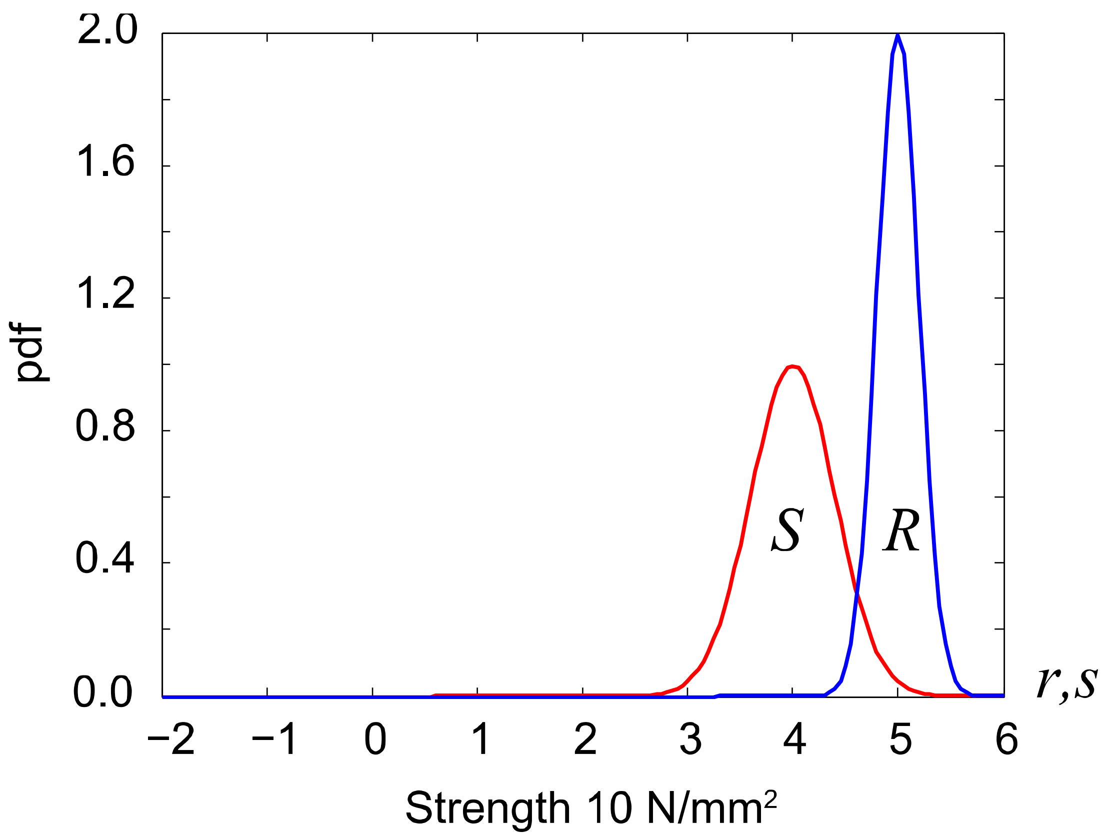
Let’s consider the load and the resistance as uncertain.

Let’s assume that *r* and s are independent normal distributed random variables with known statistical characteristics (mean and standard deviation).



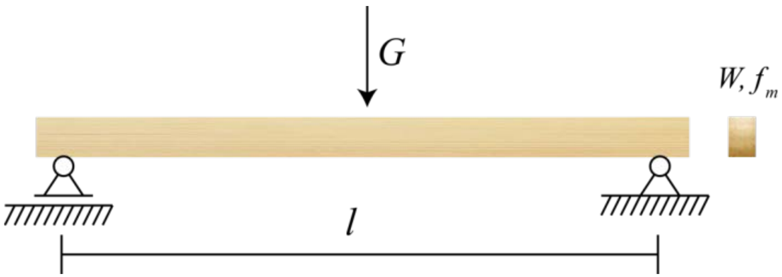
$$R = f_m W; \quad S = \frac{Gl}{4}$$

$$P_F = P(R < S) = P(R - S < 0)$$



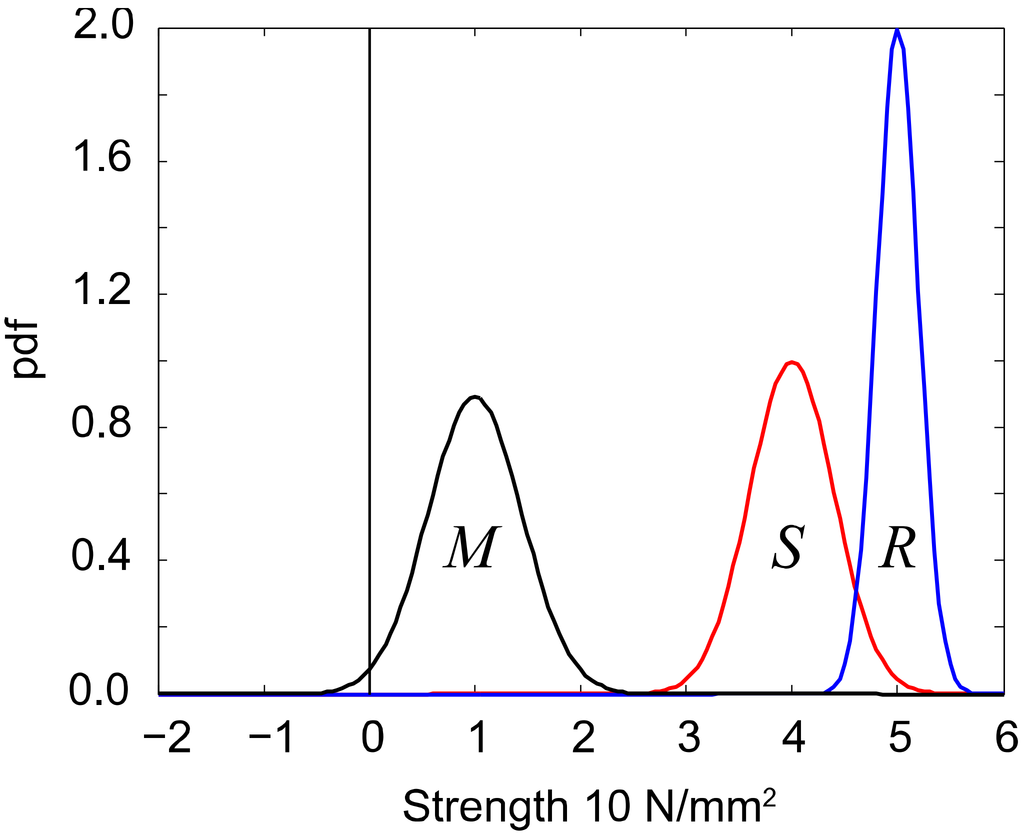
$$S \sim N(\mu_S, \sigma_S)$$

$$R \sim N(\mu_R, \sigma_R)$$



$$R = f_m W; \quad S = \frac{Gl}{4}$$

$$P_F = P(R < S) = P(R - S < 0)$$



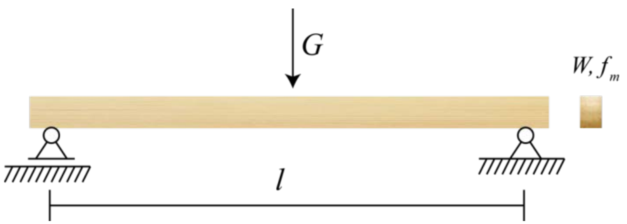
$$S \sim N(\mu_S, \sigma_S)$$

$$R \sim N(\mu_R, \sigma_R)$$

$$M \sim N(\mu_M, \sigma_M) \qquad \mu_M = \mu_R - \mu_S$$

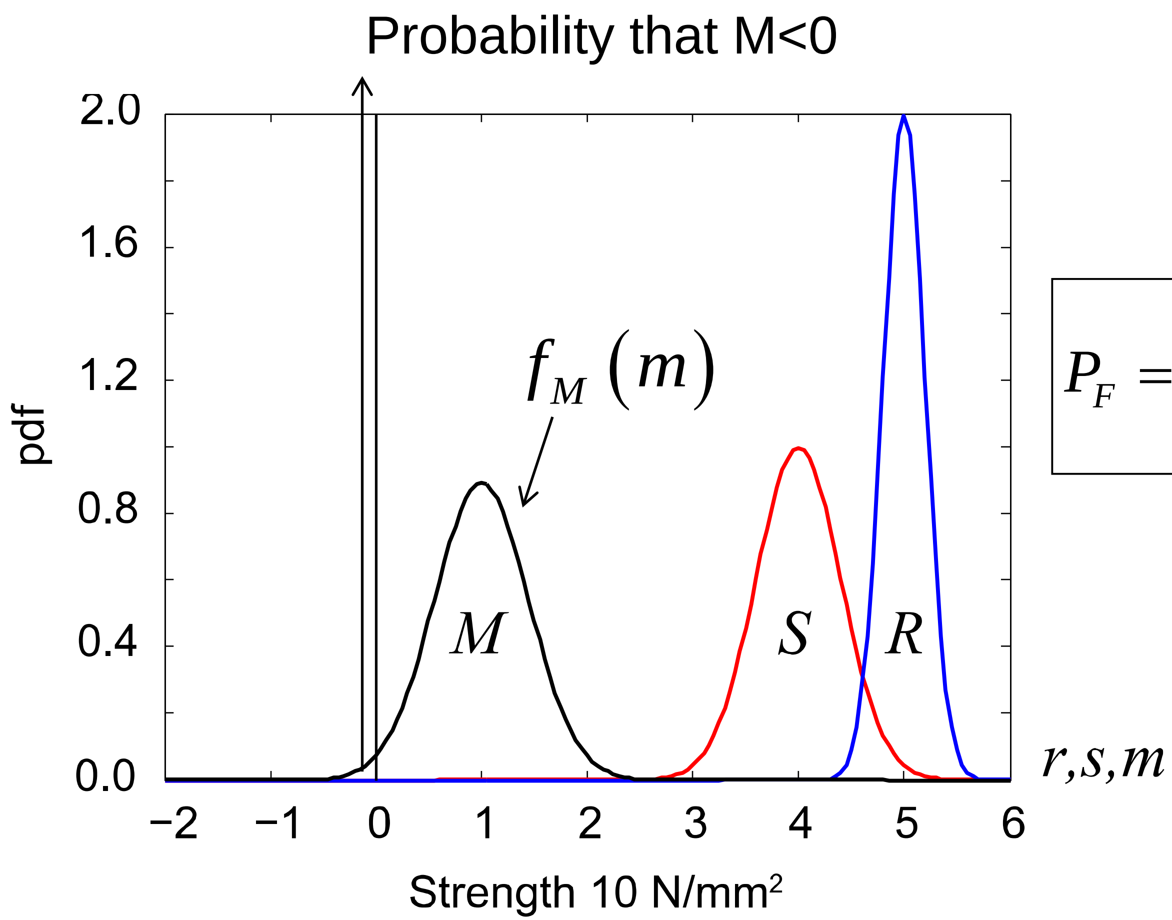
$$\sigma_M^2 = \sigma_R^2 + \sigma_S^2$$

r, s, m

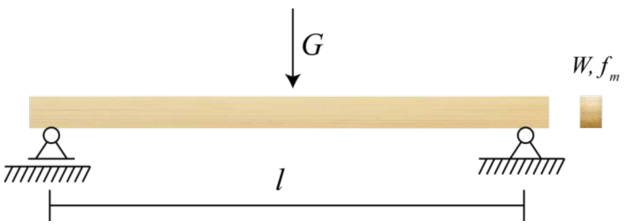


$$R = f_m W; \quad S = \frac{Gl}{4}$$

$$P_F = P(R < S) = P(R - S < 0) = P(M < 0)$$

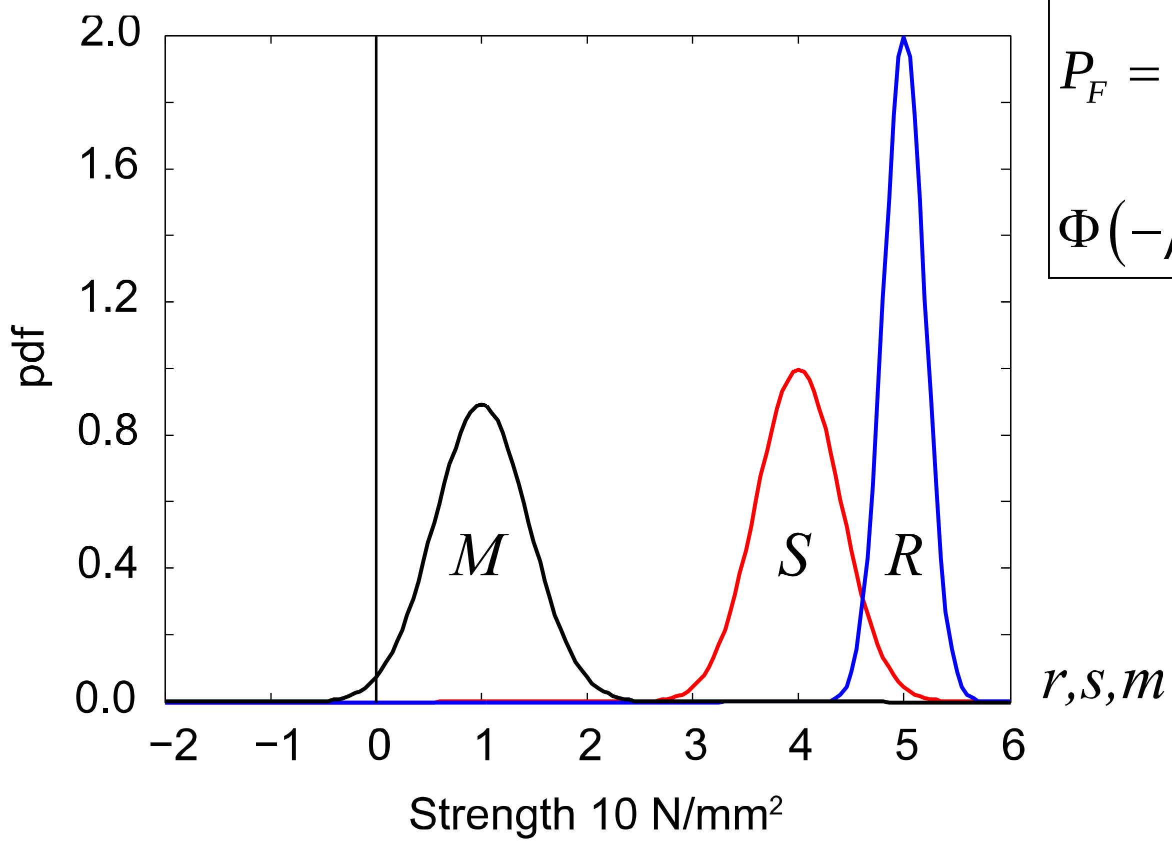


$$P_F = \int_{-\infty}^{0} f_M(m)\, dm = \Phi\left(\frac{0 - \mu_M}{\sigma_M}\right)$$



$$R = f_m W; \quad S = \frac{Gl}{4}$$

$$P_F = P(R < S) = P(R - S < 0) = P(M < 0)$$



$$P_F = \int_{-\infty}^{0} f_M(m)\, dm = \Phi\left(\frac{0 - \mu_M}{\sigma_M}\right) = \Phi(-\beta)$$

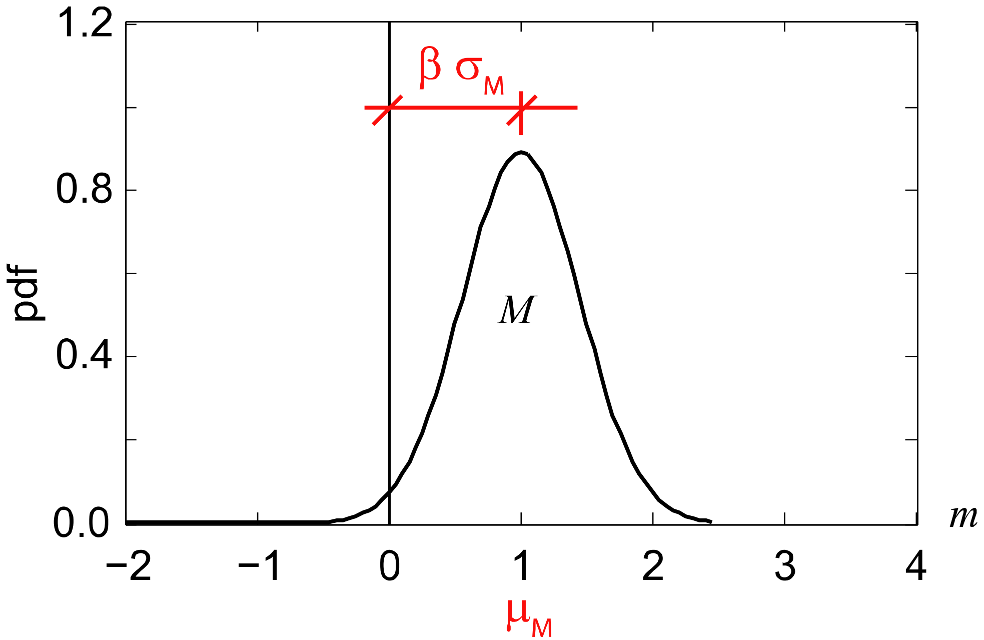
Reliability Index

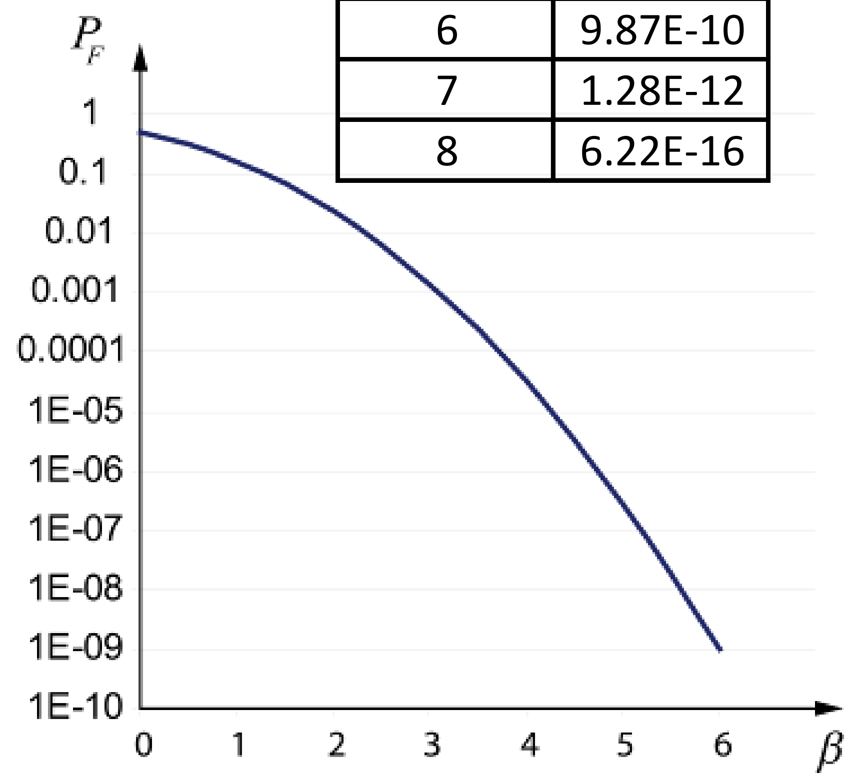
Reliability index β

Geometrical interpretation

$$P_F = \Phi\left(-\frac{\mu_M}{\sigma_M}\right) = \Phi(-\beta)$$

β	P_F
0	0.5
1	0.158655
2	0.02275
3	0.00135
4	3.17E-05
5	2.87E-07
6	9.87E-10
7	1.28E-12
8	6.22E-16





Example: Reliability Calculation

- Timber beam, length l = 8 m, cross section 10 x 10 cm.
- Load G: 1 person
- Simple probabilistic model (assumption normal distribution):

Resistance

$$\mu_{f_m} = 40.8\, N/mm^2$$

$$\sigma_{f_m} = 10.2\, N/mm^2$$

$$W = 0.1^3 m^3 / 6 = 0.00017 m^3$$

$$\mu_R = 6.8\, kNm$$

$$\sigma_R = 1.7\, kNm$$

Load

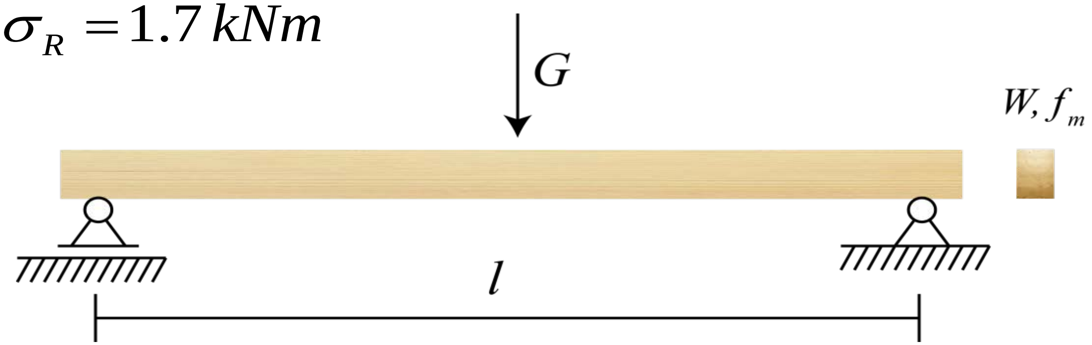
$$\mu_G = 0.8\, kN$$

$$\sigma_G = 0.15\, kN$$

$$\mu_S = 1.6\, kNm$$

$$\sigma_S = 0.3\, kNm$$

$$l = 8.00\, m$$



$$\boxed{f_m W = r < s = \frac{G\ l}{4}}$$

Example: Reliability Calculation

Resistance

$$\mu_R = 6.8\, kNm$$

$$\sigma_R = 1.7\, kNm$$

Load

$$\mu_S = 1.6\, kNm$$

$$\sigma_S = 0.3\, kNm$$

Safety margin *M*:

$$M = R - S$$

$$\mu_M = 6.8 - 1.6 = 5.2\, kNm$$

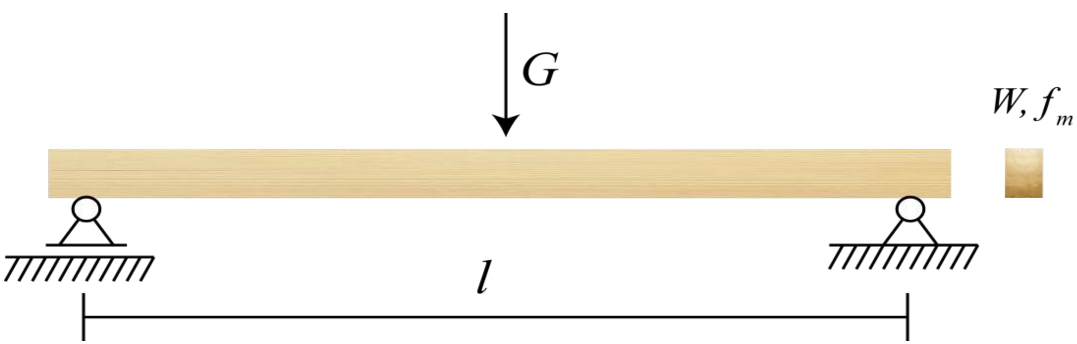
$$\sigma_M = \sqrt{1.7^2 + 0.3^2} = 1.73\, kNm$$

Reliability index β:

$$\beta = \frac{\mu_M}{\sigma_M} = 3.01$$

Probability of failure:

$$P_F = \Phi(-\beta) = 0.0013$$



$$\boxed{f_m W = r < s = \frac{G\, l}{4}}$$

Structural reliability

In general, resistance and load is described by basic random variables **X**.

$$R = f_1(\mathbf{X})$$
$$S = f_2(\mathbf{X})$$

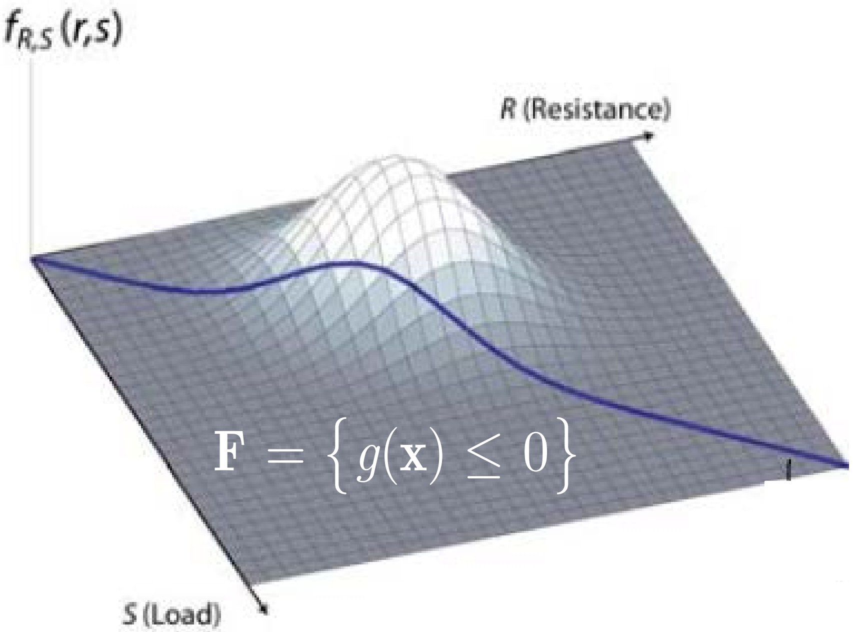
The safety margin M can be written as *g(x)*.
This is the so called limit state function.

$$M = R - S = f_1(\mathbf{X}) - f_2(\mathbf{X}) = g(\mathbf{X})$$

Failure, if *g(x)* smaller than 0:

$$g(\mathbf{x}) \leq 0$$

If g(x)=0, one obtains a (n-1) dimensional surface in n-dimensional space of the basic random variables X.



This surface separates the safe and the failure region.

The probability of failure can be written as:

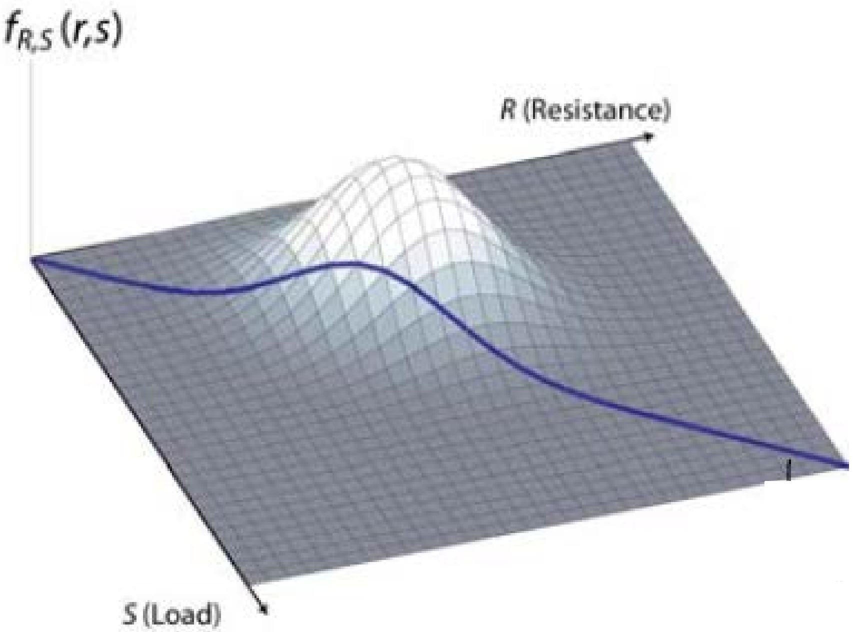
$$P_f = \int_{\Omega_f = \{g(\mathbf{x}) \leq 0\}} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}$$

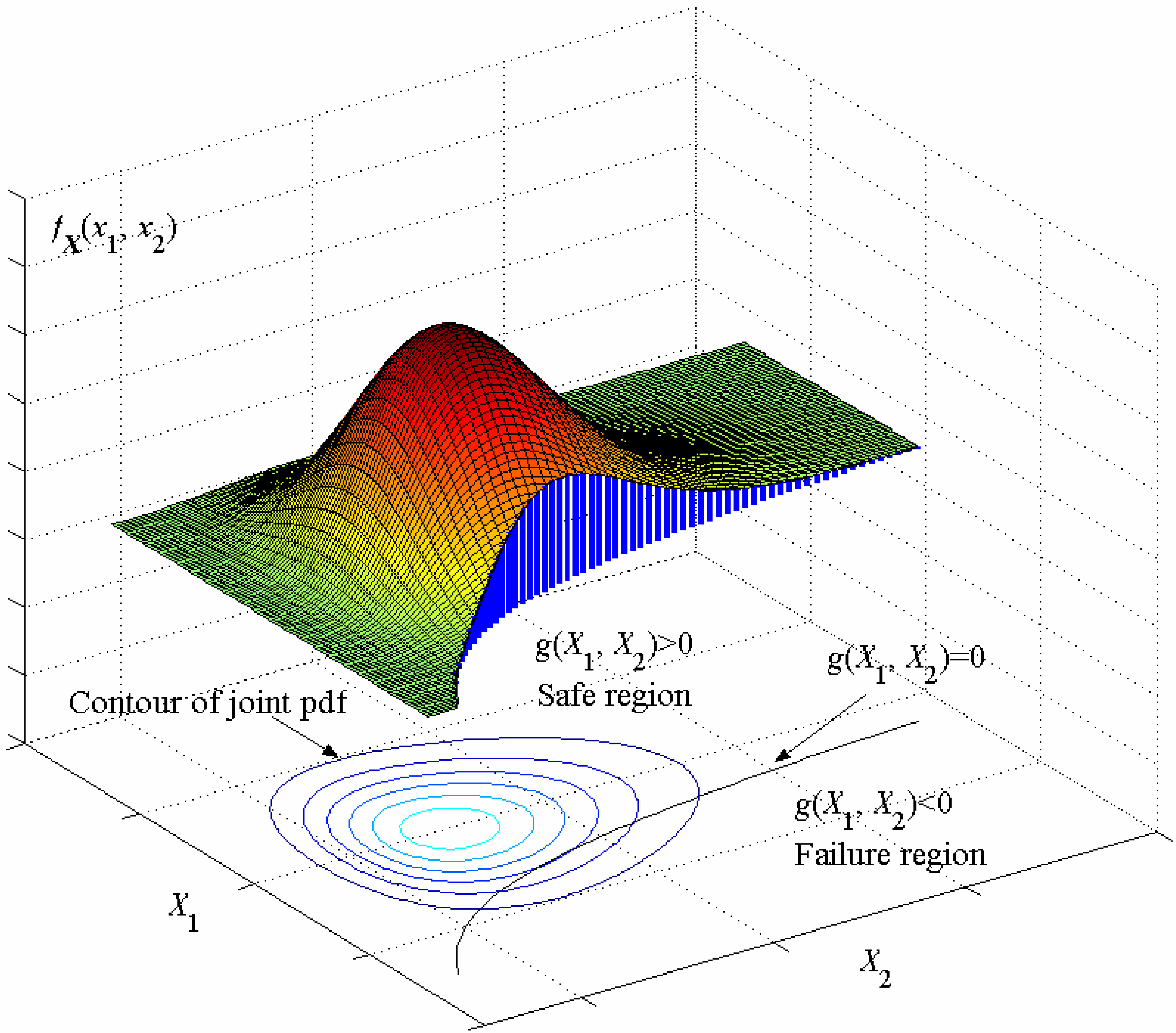
To calculate the failure probability we **only** need to solve the following integral:

$$P_f = \int_{\Omega_f = \{g(\mathbf{x}) \leq 0\}} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}$$

$f_X(\mathbf{x})$ is the joint density function of all basic random variables.

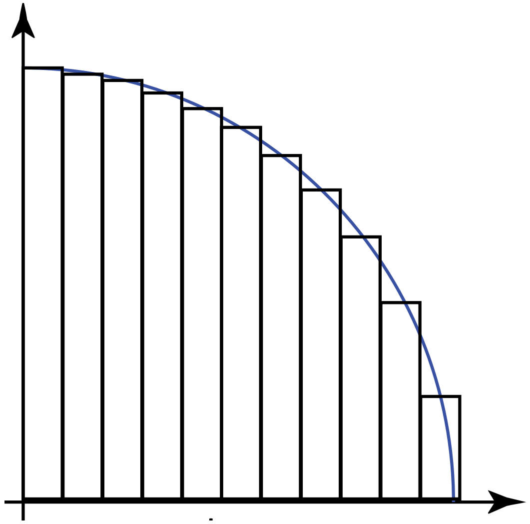
In case of 2-dimensions:




$f_X(x_1, x_2)$
$g(X_1, X_2)>0$
Safe region
$g(X_1, X_2)=0$
Contour of joint pdf
$g(X_1, X_2)<0$
Failure region
X_1
X_2

Bad news: the integral can normally not solved analytically.

The integral can be solved numerically (e.g. Gauss, Simpson, Tchebyschev).



For problems with more than 6 dimensions this integration becomes somehow cumbersome.

Methods of structural safety

- First Order Reliability Method (FORM)
- Second Order Reliability Method (SORM)
- Surrogate models (response surface, support vector machines, polynomial chaos expansion)
- Monte Carlo Simulation
  (incl. different sampling methods)

Monte Carlo Simulation

A widely used and **easy methodology** to calculate integrals.

Monte Carlo Simulations have been developed in the 1930th.

It is based on the idea of trial and error.

Monte Carlo can be called exact method since the result will converge for n-> ∞ to the exact result.

Different approaches are available, e.g.

- **Inverse Transformation Method**
- Acceptance Rejection Method
- Metropolis Hasting Algorithm
- …

Inverse Transformation Method

If we can invert the cumulative distribution function of a random variable then we can apply the inverse transformation method.

The idea is as follows:

We know that the probability is bounded between 0 and 1

We generate realizations from a uniform distribution U(0,1) and use them to calculate realizations of our CDF.

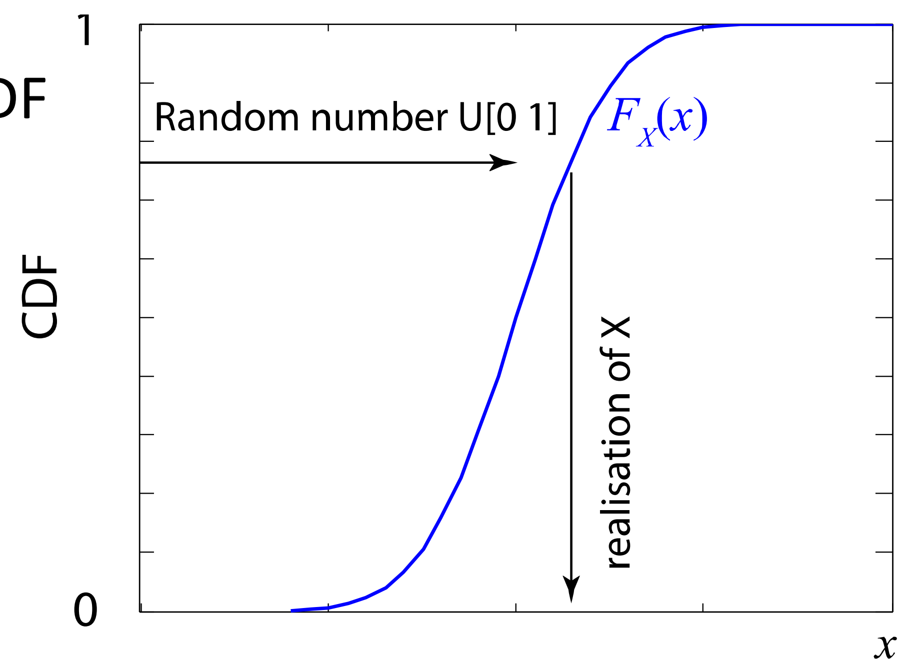
Monte Carlo Simulation – Step 1

Step 1: A set of basic random variables is generated which represents the basic random variables.

For this purpose a uniform distributed random number between zero and one generated (pseudo random number generator)

Using the inverse function of the CDF random number is converted to realization of a distribution function.

→For each basic random variable a a random realization is generated.



Random number generation in Matlab

Matlab has implemented different random number generators:

`rand`	$U(0,1)$ distributed pseudo random numbers
`randn`	$N(0,1)$ distributed pseudo random numbers
`wblrnd`	
`lognrnd`	
`exprnd`	
`gamrnd`	
`evrnd`	Extreme Value distribution

....and more ☺ - use the help im matlab for find more.

Monte Carlo Simulation – Step 2 and 3

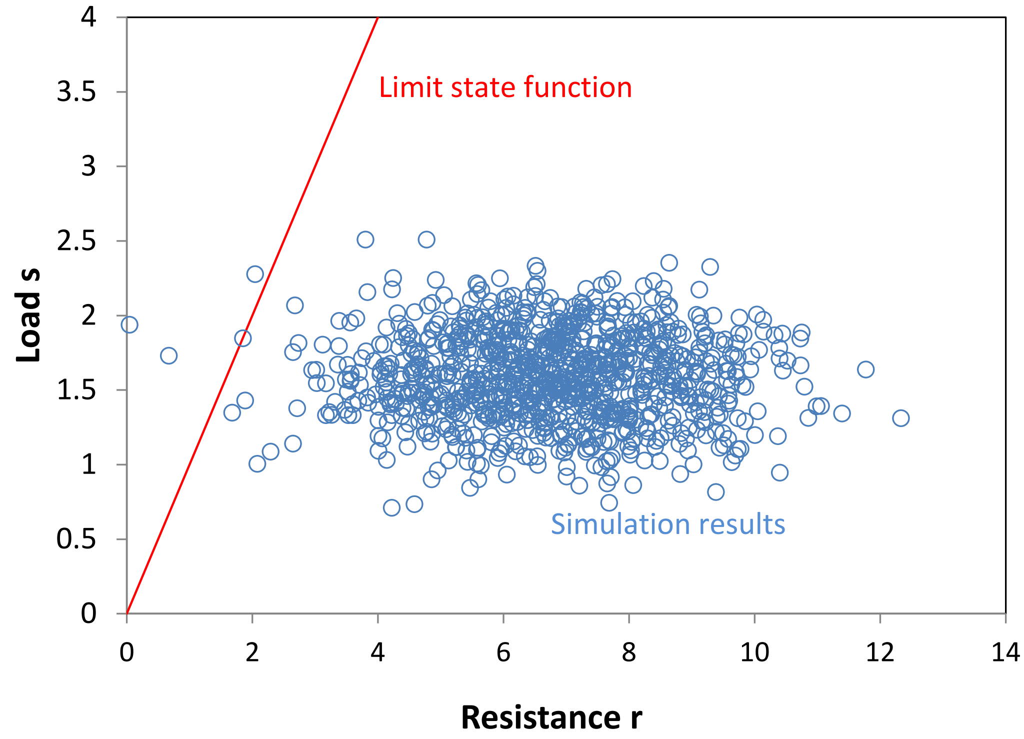
Step 2: The simulated realizations are used in the limit state function.

Step 3: The limit state function is then evaluated with these values and checked if limit state function < 0.

If yes, then this failure event is counted n_f .

Example (Timber beam)

$n_f = 1$



Monte Carlo Simulation – Step 4

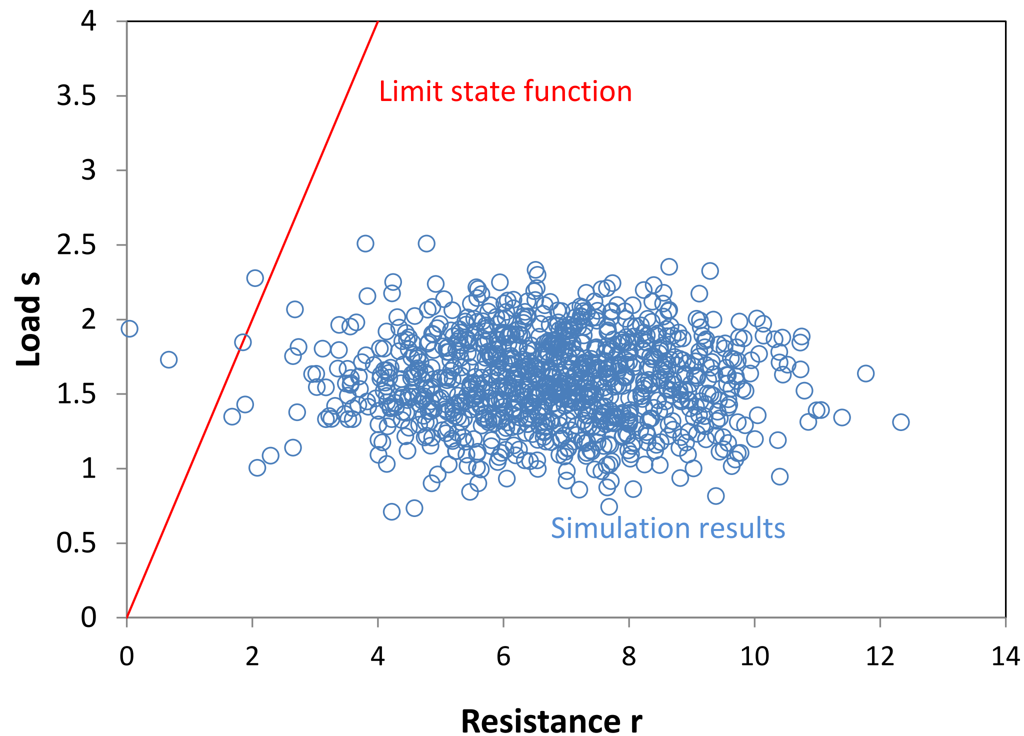
Step 4: The ratio between the number of failures and the total number of simulations is calculated.

$$P_f \simeq \frac{n_f}{m}$$

$$P_f \simeq \frac{3}{1000} = 0.003$$

For the timber beam example we calculated analytically:

$$P_f = 0.0013$$

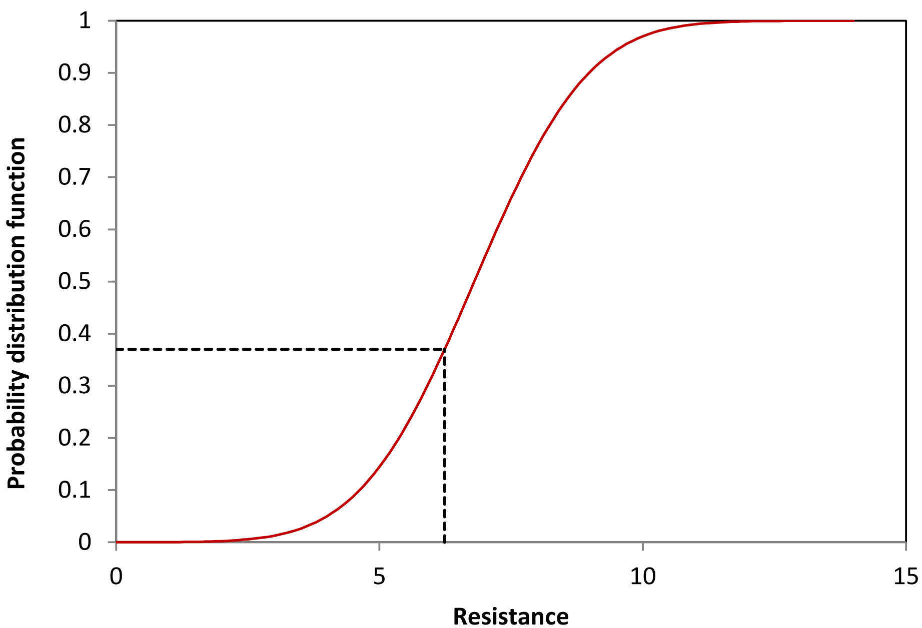


Example: Normal distribution beam

Step 1: A set of basic random variables is generated which represents the basic random variables.

Generate random number: 0.37

Excel: =Rand()



Resistance model		
$\mu f_m =$	40.8	N/mm^2
$\sigma_m =$	10.2	N/mm^2
$W =$	0.00016667	m^3
$\mu_R =$	6.8	kNm
$\sigma_R =$	1.7	kNm

Calculate realization: 6.24 kNm

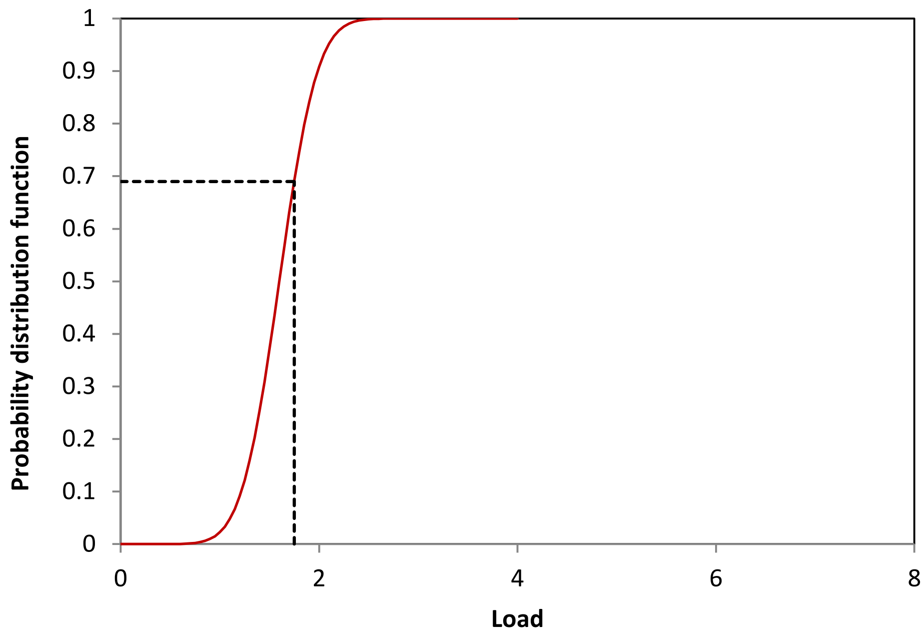
Excel: =NORM.INV(0.37;6.8;1.7)

Example: Normal distribution beam

Step 1: A set of basic random variables is generated which represents the basic random variables.

Generate random number: 0.69

Excel: =Rand()



load model	
$\mu_G =$	0.8 N/mm^2
$\sigma_G =$	0.15 N/mm^2
$l =$	8 m
$\mu_S =$	1.6 kNm
$\sigma_S =$	0.3 kNm

Calculate realization: 1.75 kNm

Excel: =NORM.INV(0.69;1.6;0.3)

Example: Normal distribution beam

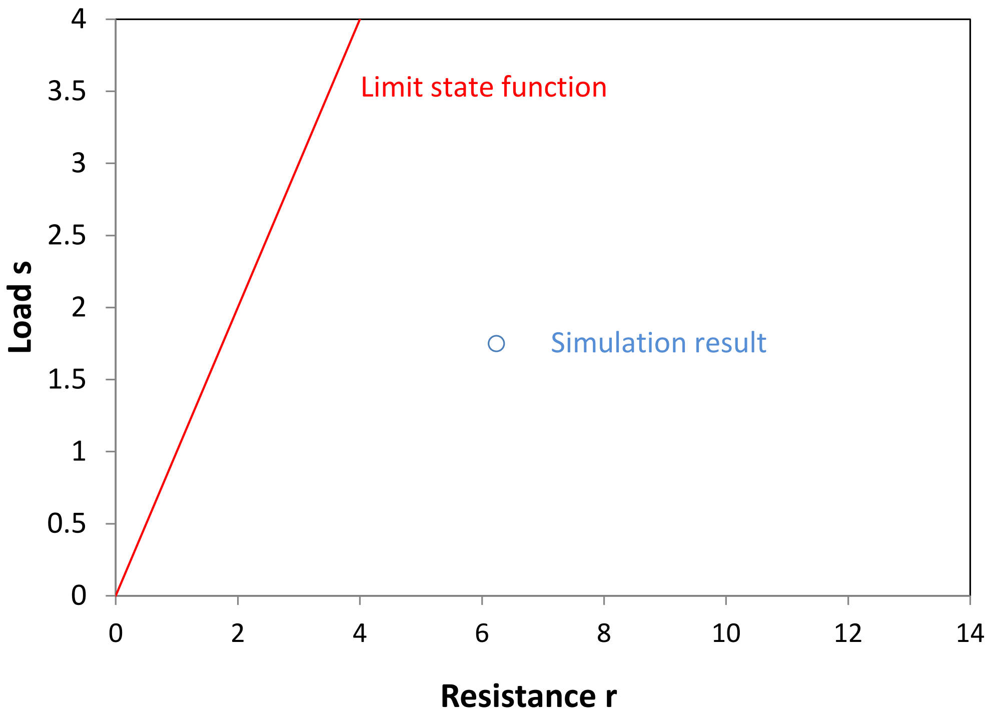
Step 2: The simulated realizations are used in the limit state function.

$$\boxed{r - s = 6.24 - 1.75 > 0}$$

Example: Normal distribution beam

Step 3: The limit state function is then evaluated with these values and checked if limit state function < 0.

$$\boxed{r - s = 6.24 - 1.75 > 0}$$



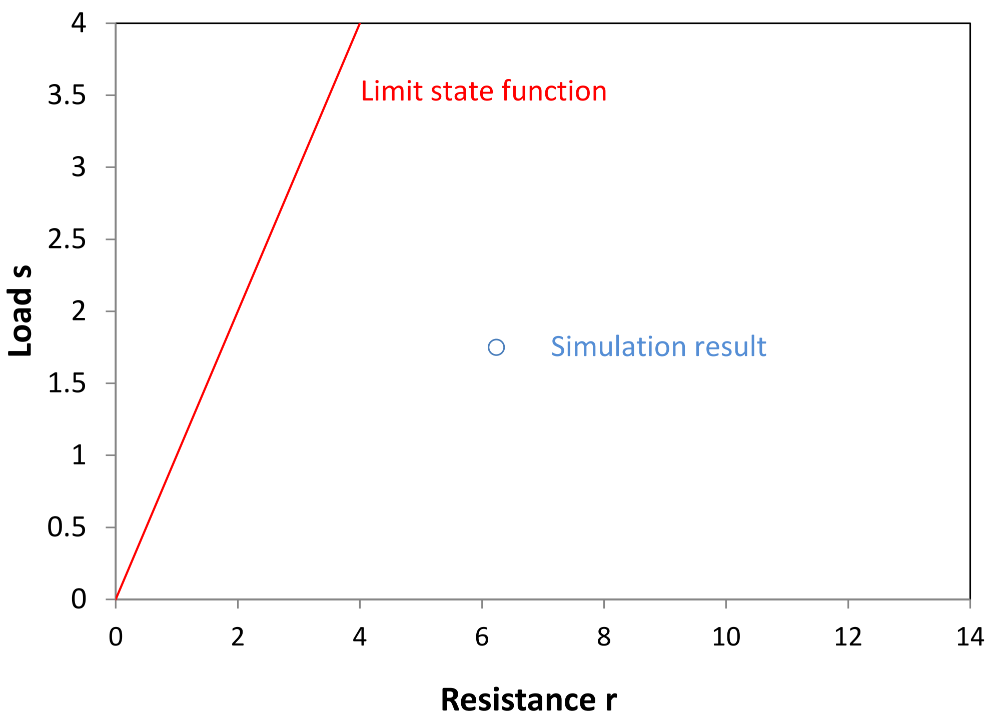
Example: Normal distribution beam

Step 4: The ratio between the number of failures and the total number of simulations is calculated.

n_f = Number g(x) < = 0

Here $n_f = 0$

$$\Pr(F) \simeq \frac{n_f}{m} = \frac{0}{1} = 0$$



Repeat Steps 1-4

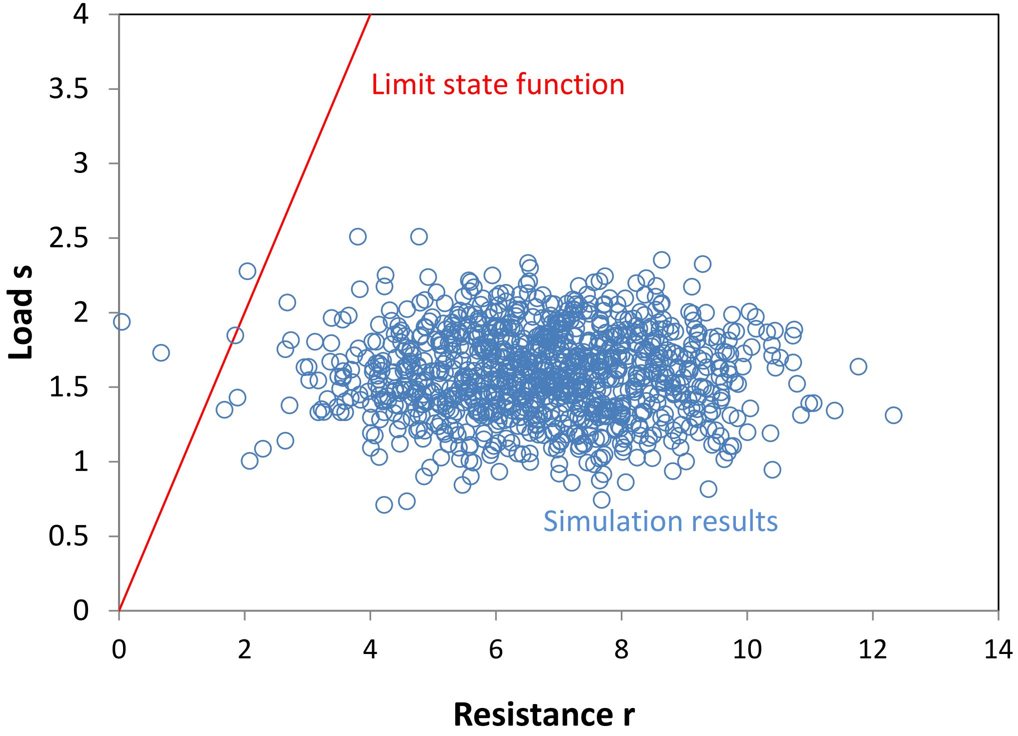
Example: Normal distribution beam

Repeat Steps 1-4

n_f =Number g(x) < = 0

Here $n_f = 3$

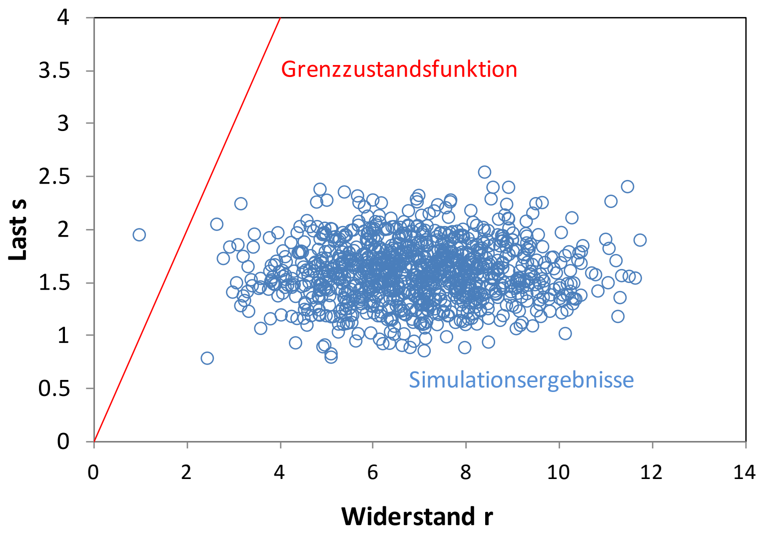
$$\Pr(F) \simeq \frac{n_f}{m} = \frac{3}{1000} = 0.003$$



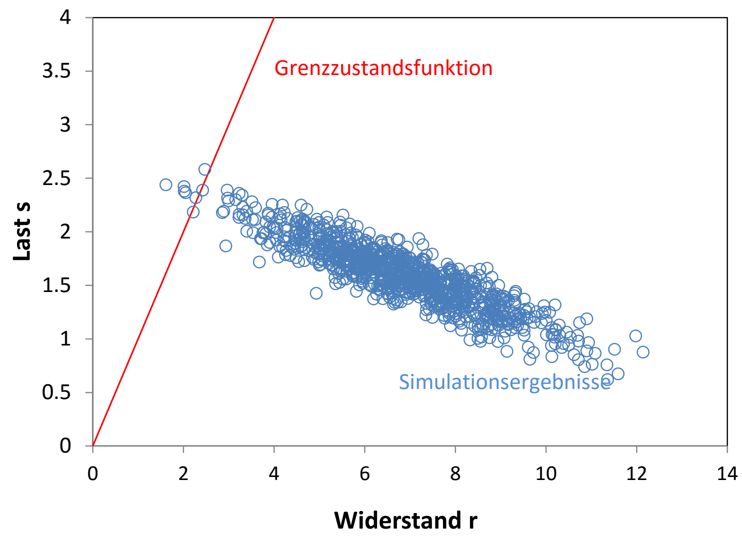
Monte Carlo Simulation

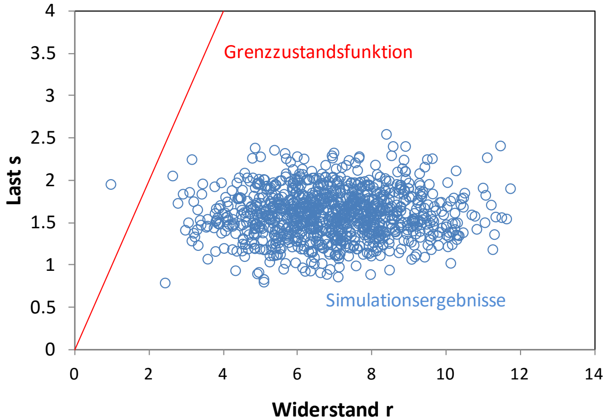
If multivariate distributions are used, correlations need to be considered (e.g. using **copulas**). Strong influence on failure probability.

Load and resistance not correlated; here: $\rho=0$

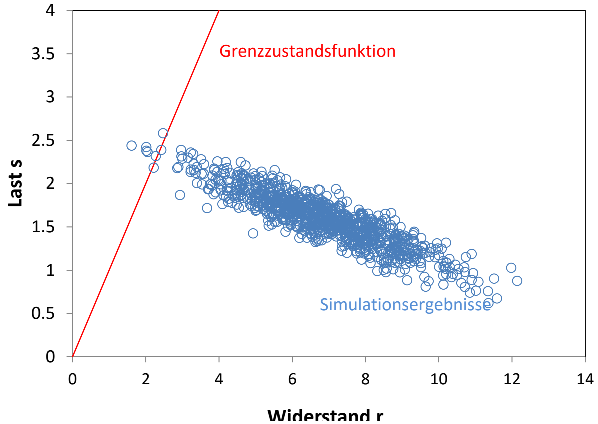


Load and resistance correlated; here: $\rho=-0.9$

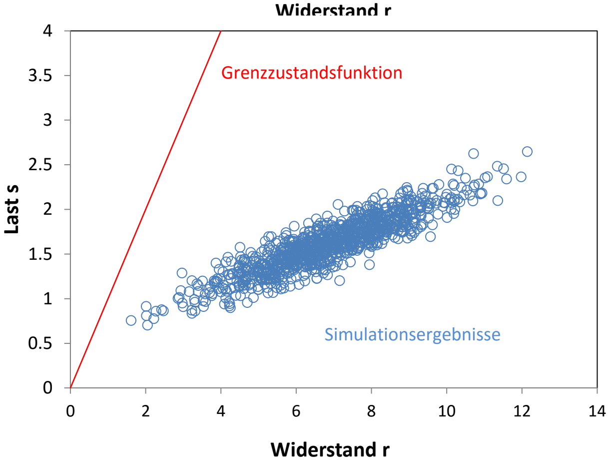




Load and resistance uncorrelated; here: ρ=0.0



Load and resistance negatively correlated; here: ρ=-0.9



Load and resistance positively correlated; here: ρ=+0.9

Multivariate case

Sometimes (actually often) random variables are correlated, so that we have to generate multivariate random variables.

In Matlab (e.g.) you can generate multivariate normal distributed random variables.

How can we then generate multivariate random numbers?

Generating correlated bivariate random variables

To generate correlated variables we generate first independent variables and then we rotate the coordinate system in a way that they become dependent.

The transformation between the coordinate system is:

$$x' = x \cdot \cos(\alpha) + y \cdot \sin(\alpha)$$
$$y' = -x \cdot \sin(\alpha) + y \cdot \cos(\alpha)$$

We set the correlation coefficient to $\rho = \cos(\alpha)$ and receive:

$$x' = x \cdot \rho + y \cdot \sqrt{1-\rho^2}$$

Generating correlated bivariate normal RV

So we can generate bivariate normal distributions by applying the following algorithm:

1) Generate X and Y (standard normal distribution)
2) Calculate

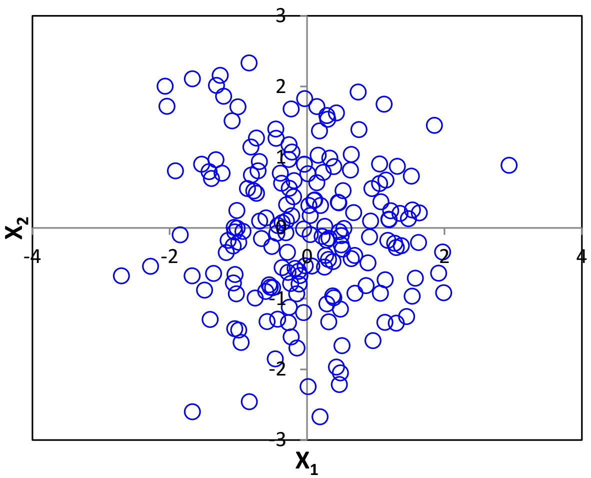
$$X' = X \cdot \rho + Y \cdot \sqrt{1-\rho^2}$$

3) Get the correlated pair of standard normal distributed random variables (X, X')

4) Transform (X_1, X_2) to general bivariate normal distribution → $\mu_i + X_i\ \sigma_i$.

Generating correlated bivariate normal RV

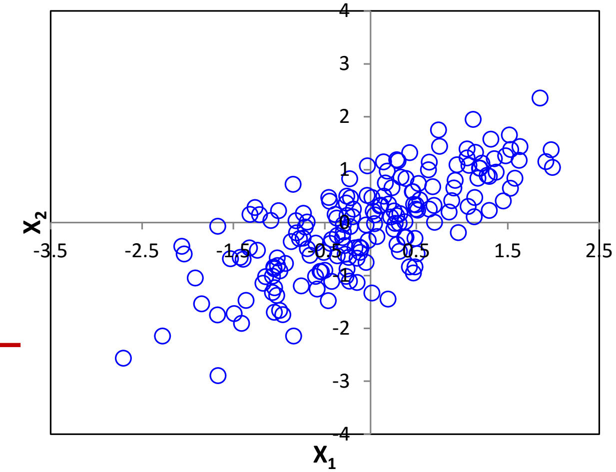
Example: Normal distribution



Uncorrelated – standard normal

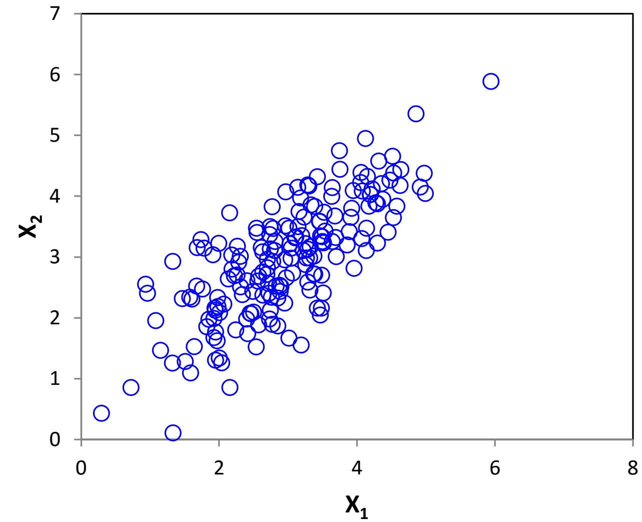
$$X' = X \cdot \rho + Y \cdot \sqrt{1-\rho^2}$$

$\rho = 0.8$



$$3 + 1\,X_i$$

$\mu_1 = \mu_2 =$	3
$\sigma_1 = \sigma_2 =$	1
$\rho =$	0.8



Generating correlated normal bivariate RV

$\mu =$	0
$\sigma =$	1

$\mu_1 = \mu_2 =$	3
$\sigma_1 = \sigma_2 =$	1
$\rho =$	0.8

$$X' = 3 + 1\left(X \cdot \rho + Y \cdot \sqrt{1-\rho^2}\right)$$

X

Y

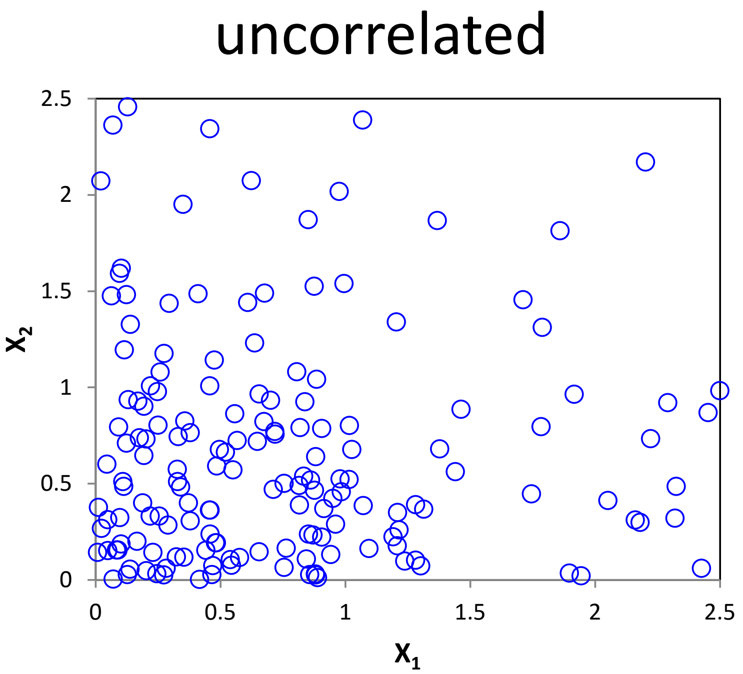
$3 + 1 \cdot X$

$N(\mu,\sigma)$	$N(\mu,\sigma)$
0.045536022	-0.095726081
-0.454720527	1.395708488
-1.91584647	0.806781165
0.30290085	1.537498226
0.322068089	-0.148771016
-0.991965403	-0.202352608
0.144022555	0.637971802
-1.263450815	2.155718699
-0.693176352	0.938555661
0.220955537	-0.12083156
-0.033617252	1.829651163
0.047658595	0.174281884
-1.04914267	-0.660055983

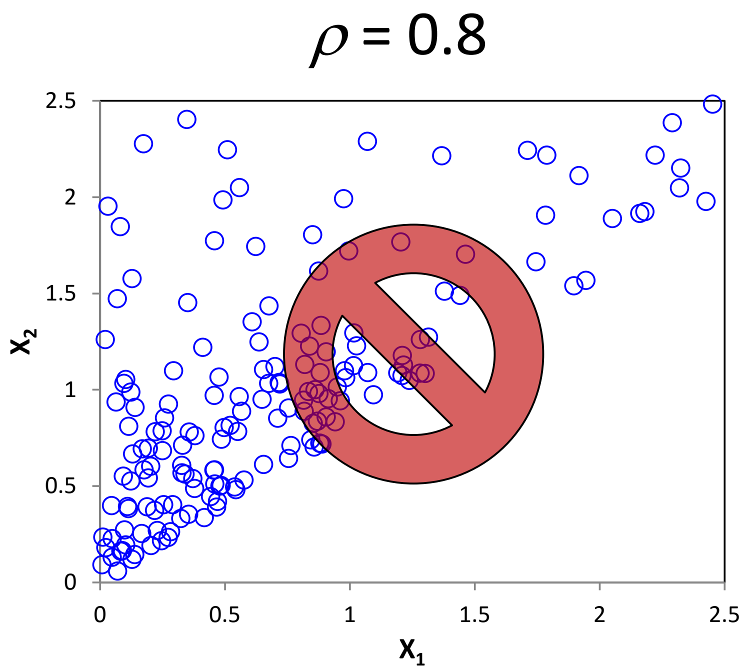
MVN (μ1,μ2 σ1,σ2)	
3.045536022	2.978993168
2.545279473	3.473648671
1.08415353	1.951391523
3.30290085	4.164819615
3.322068089	3.168391862
2.008034597	2.085016112
3.144022555	3.498001125
1.736549185	3.282670567
2.306823648	3.008592315
3.220955537	3.104265494
2.966382748	4.070896896
3.047658595	3.142696007
1.95085733	1.764652275

Generating correlated bivariate normal RV

Example: Exponential distribution



$$X' = X \cdot \rho + Y \cdot \sqrt{1-\rho^2}$$



Transformation not valid for other distributions!

Generating correlated bivariate RV

Idea:

We can generate bivariate normal random numbers.

It is easy to re-transform these multivariate normal distributed random variables into correlated uniform distributed random Numbers → using the CDF of the normal distribution.

By using these correlated uniform distributed random numbers we can generate correlated random numbers form any multivariate distribution.

Generating correlated bivariate RV: NORTA

Normal to anything

1) Generate a pair of correlated standard normal random variables (X , X')

2) Calculate $U_1 = \Phi(X)$ and $U_2 = \Phi(X')$

e.g. Excel: `Norm.dist(X,0,1)`
Matlab: `normcdf(X,0,1)`

We get correlated uniform random numbers (U_1,U_2)

3) Use U_i to generate correlated random variables – by using the before mentioned approach.

NORTA: Example

$$X = X$$
$$X' = X \cdot \rho + Y \cdot \sqrt{1-\rho^2}$$

$$U_1 = \Phi(X)$$

$$Z_i = -\frac{\ln(1-U_i)}{\lambda}$$

Nr. Sim	$N(\mu=0, \sigma=0)$	$N(\mu=0, \sigma=0)$
1	-1.721617252	0.040385677
2	1.352786938	-0.285493347
3	0.080014232	0.172940922
4	-0.166766125	0.319333788
5	0.104898079	-1.739373857
6	-0.536203406	0.200664025
7	1.097379345	0.59123515
8	-0.601325973	-0.045777285
9	-0.041260701	1.142412507
10	-0.345012549	1.062599643
11	-0.038455833	-1.428023401

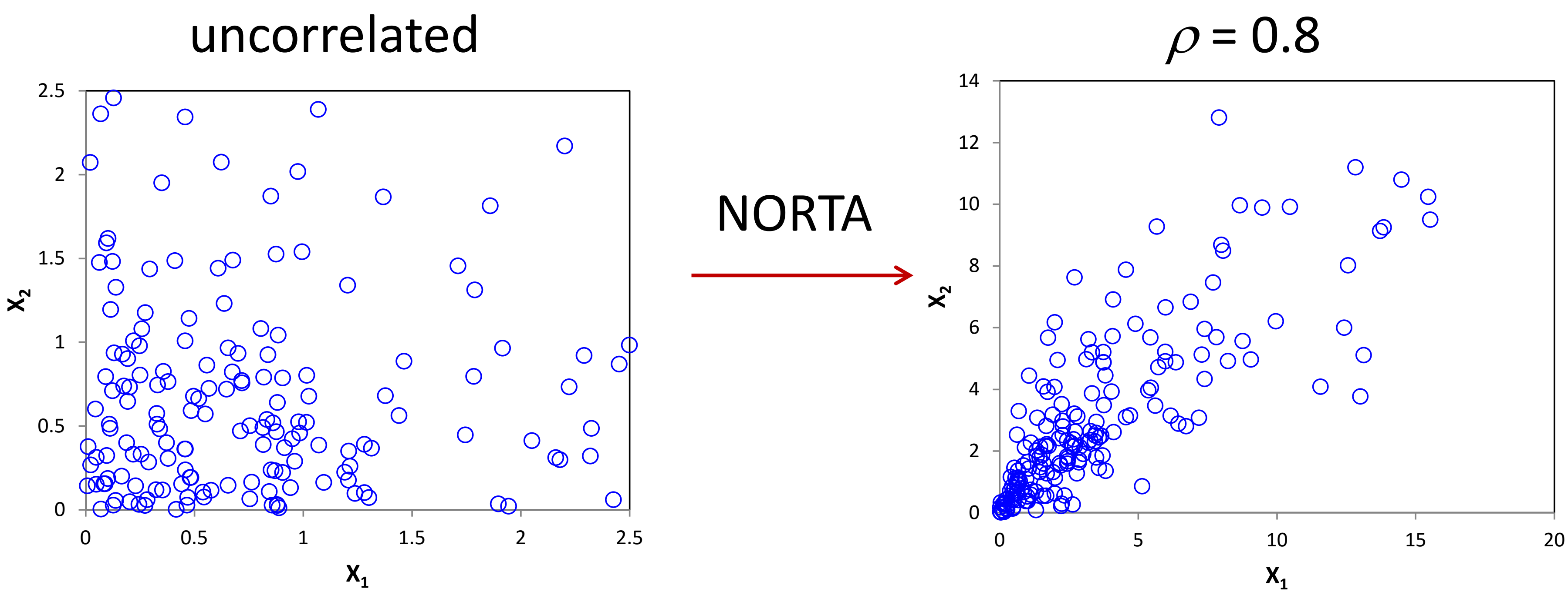
MVN (0, 0, 1, 1, ρ = 0.8)	
-1.721617252	-1.353062396
1.352786938	0.910933543
0.080014232	0.167775939
-0.166766125	0.058187373
0.104898079	-0.959705851
-0.536203406	-0.30856431
1.097379345	1.232644566
-0.601325973	-0.508527149
-0.041260701	0.652438943
-0.345012549	0.361549747
-0.038455833	-0.887578707

U (0,1)	U (0,1)
0.04256944	0.08801785
0.91193815	0.8188348
0.53188703	0.56662022
0.43377704	0.52320031
0.54177166	0.16860164
0.29590899	0.37882649
0.8637622	0.8911458
0.27381145	0.30554185
0.48354403	0.74294098
0.36504247	0.64115574
0.48466212	0.18738371

MVExp (λ_1,λ_2)	
0.13050624	0.27640458
7.2891475	5.12503795
2.27713688	2.50842256
1.70630205	2.22197644
2.34116297	0.55393867
1.05254297	1.42843448
5.98006018	6.65323769
0.95983675	1.09387014
1.98229573	4.07534866
1.36259148	3.07460039
1.98879756	0.62248873

NORTA: Example

Example: Exponential distribution

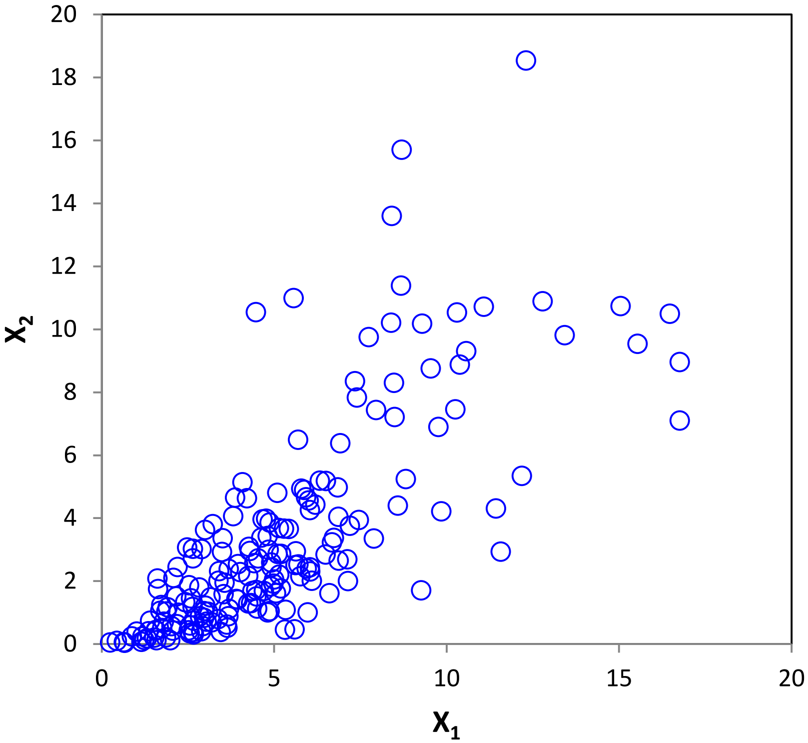


NORTA: Example

Example:

Random samples from a bivariate Gumbel-Exponential distribution

$$X_1 \sim G(\mu=5, \sigma=3)$$
$$X_2 \sim Exp(\mu=3)$$
$$\rho = 0.8$$



From bivariate to multivariate

In general the aim is to find a matrix such that the following relation holds:

$$\mathbf{R}^T\mathbf{R} = \mathbf{C}$$

Then we need to find a matrix $\mathbf{R}$ in order to transform the uncorrelated RV $\mathbf{X}$ into correlated random variables $\mathbf{Y}$

$$\mathbf{Y} = \mathbf{XR}$$

The matrix **R** can be found by using e.g. a Cholesky Decomposition of the correlation matrix:

From bivariate to multivariate standard normal

Example:

$$\mathbf{C} = \begin{bmatrix} 1 & 0.8 & 0.2 \\ 0.8 & 1 & 0.3 \\ 0.2 & 0.3 & 1 \end{bmatrix} \xrightarrow{\text{Cholesky decomposition}} R = \begin{bmatrix} 1 & 0.800 & 0.200 \\ 0 & 0.600 & 0.233 \\ 0 & 0 & 0.952 \end{bmatrix}$$

$$\overbrace{\begin{bmatrix} 1.602 & 1.536 & 0.240 \end{bmatrix}}^{\text{Correlated standard normal distributed random variables}} = \overbrace{\begin{bmatrix} 1.602 & 0.424 & -0.188 \end{bmatrix}}^{\text{Uncorrelated standard normal distributed random variables}} \begin{bmatrix} 1 & 0.800 & 0.200 \\ 0 & 0.600 & 0.233 \\ 0 & 0 & 0.952 \end{bmatrix}$$

From bivariate to multivariate standard normal

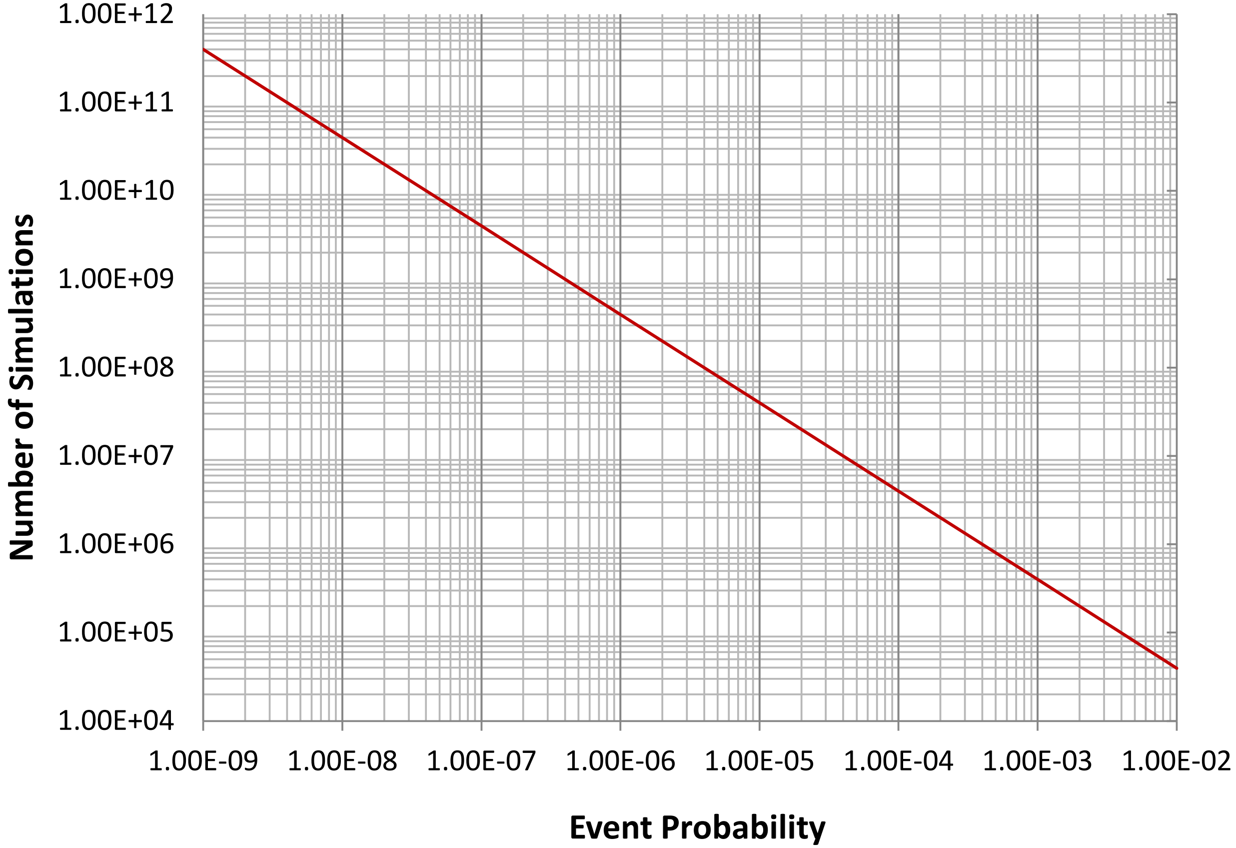
From the multivariate standard normal distributed random variables we can use the NORTA approach to calculate the multivariate uniform distribution.

By using the multivariate uniform distribution we can generate any multivariate distribution function we need.

After sampling you should check if the distribution really has the defined characteristics.

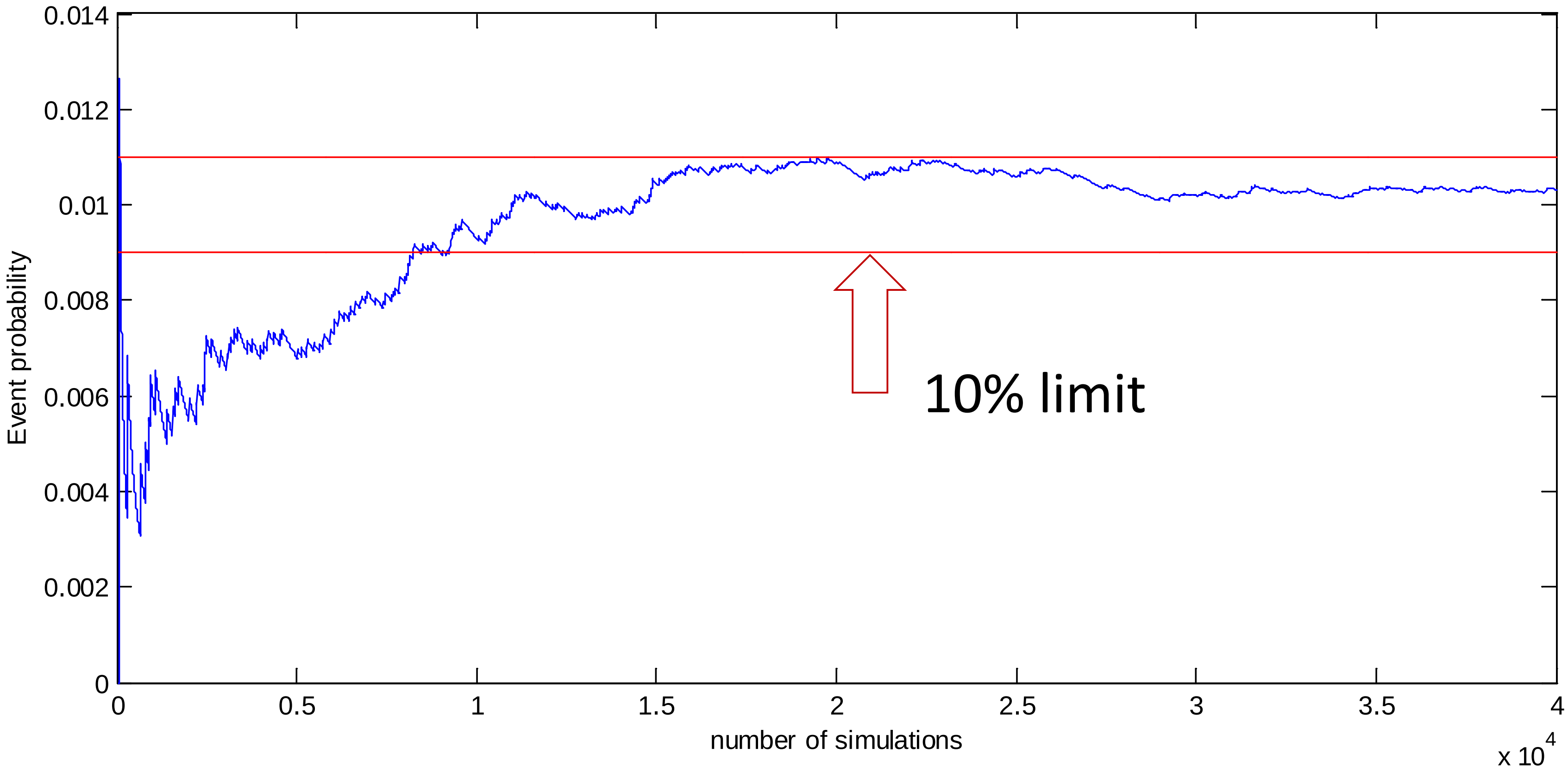
Monte Carlo Estimator

Number of simulations needed for obtaining a scatter of 10% around the mean with a probability of 90% for different event probabilities.



Monte Carlo Estimator

Example: Convergence of an estimate in dependency of the number of simulations.



Monte Carlo Simulation

Advantages

- Can be principally used for every application.
- Intuitive and simply conducted by using Matlab, Python, R C++ or even Excel VBA.
- Exact if number of simulations n→ ∞

Disadvantages

- Computational expensive if we have small probabilities of failure and a large number of variables (and dependencies among them) → sampling methods.

Monte Carlo Estimators

- Uncertainty

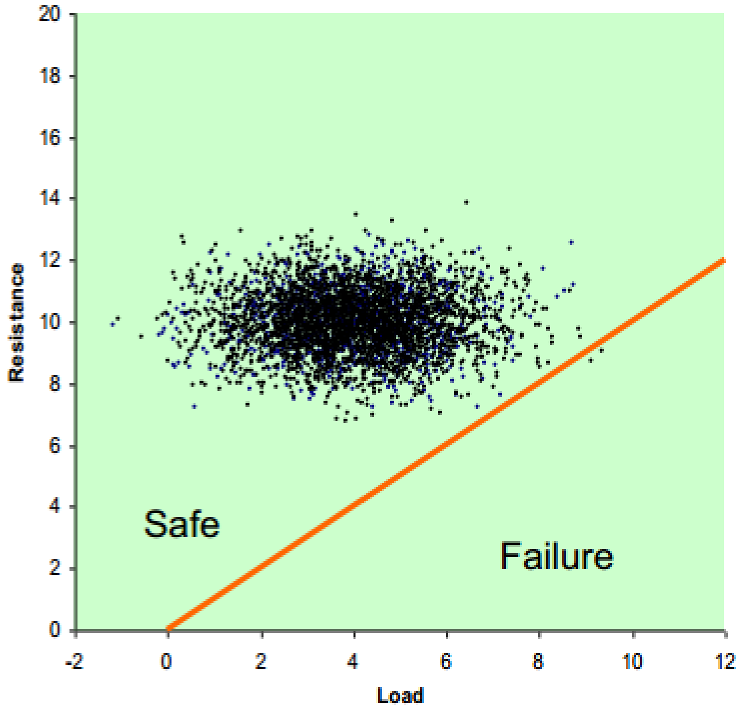
$$\sigma_{MCS} = \sqrt{(\Pr(F) - \Pr(F)^2)/n_{MCS}}$$

$$\sigma_{MCS} \approx \sqrt{(p_{MCS} - p_{MCS}^2)/n_{MCS}}$$

$$cov_{MCS} \approx \frac{1}{p_{MCS}} \sqrt{(p_{MCS} - p_{MCS}^2)/n_{MCS}}$$

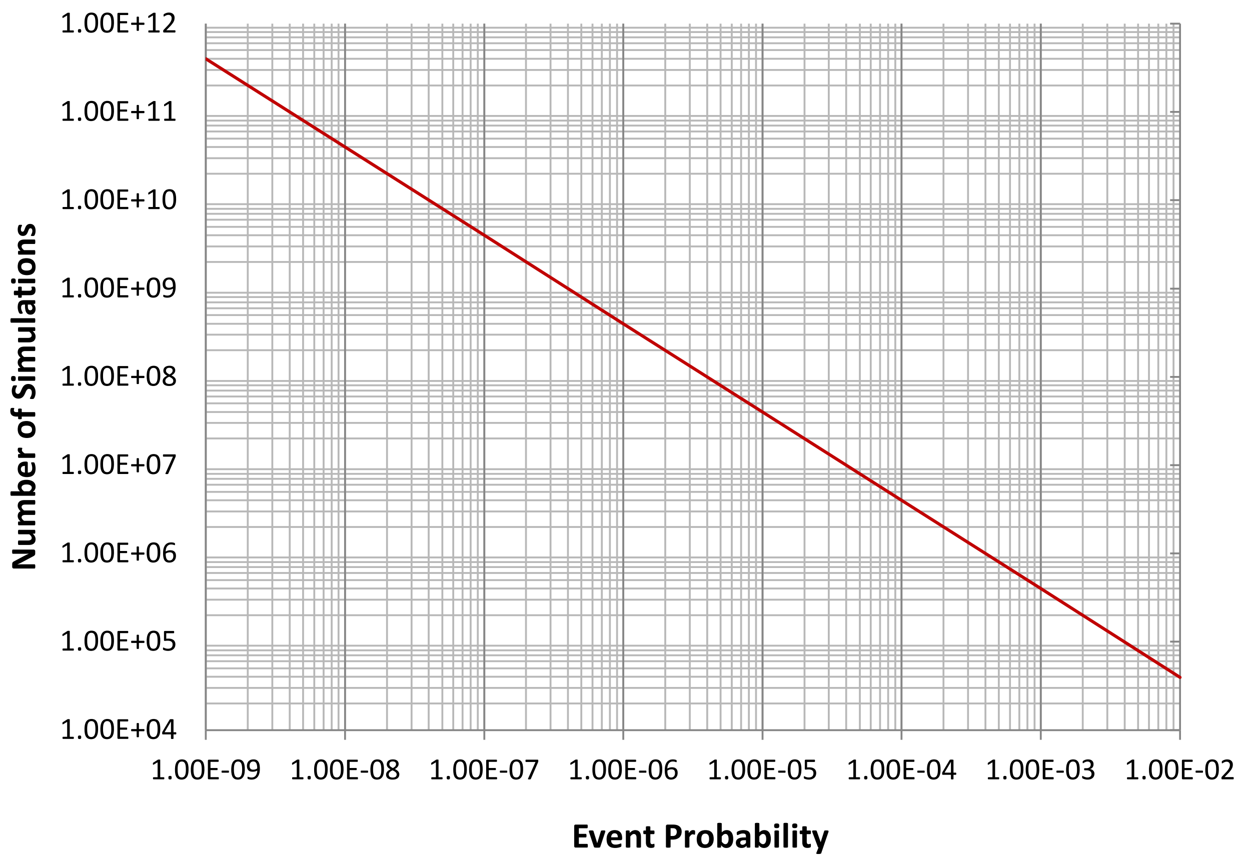
- rule of thumb:

$$n_{MCS} \geq \frac{10}{p_{MC}}$$



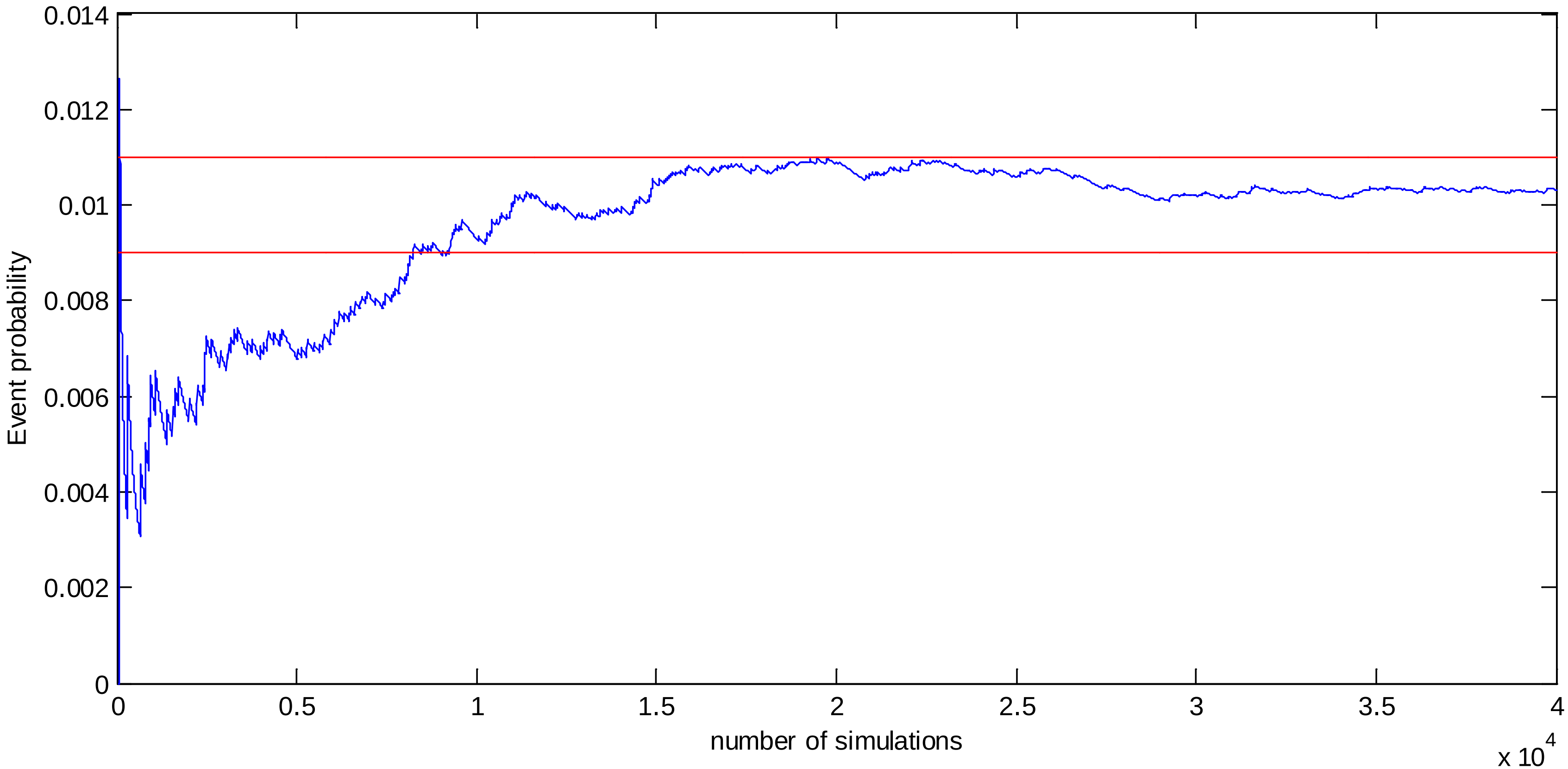
Monte Carlo Estimators

Number of simulations for receiving a 10% scatter around the mean for different event probabilities.



Monte Carlo Estimators

Conclusion: Monte Carlo is exact – but computational very expensive!



Variance reducing methodologies

Monte Carlo Simulation is computational expensive.

Variance reducing methodologies help that the result of a Monte Carlo simulation converges faster.

Several different methods have been developed in the last decades, e.g.:

Importance sampling
Stratified sampling / Latin hypercube sampling

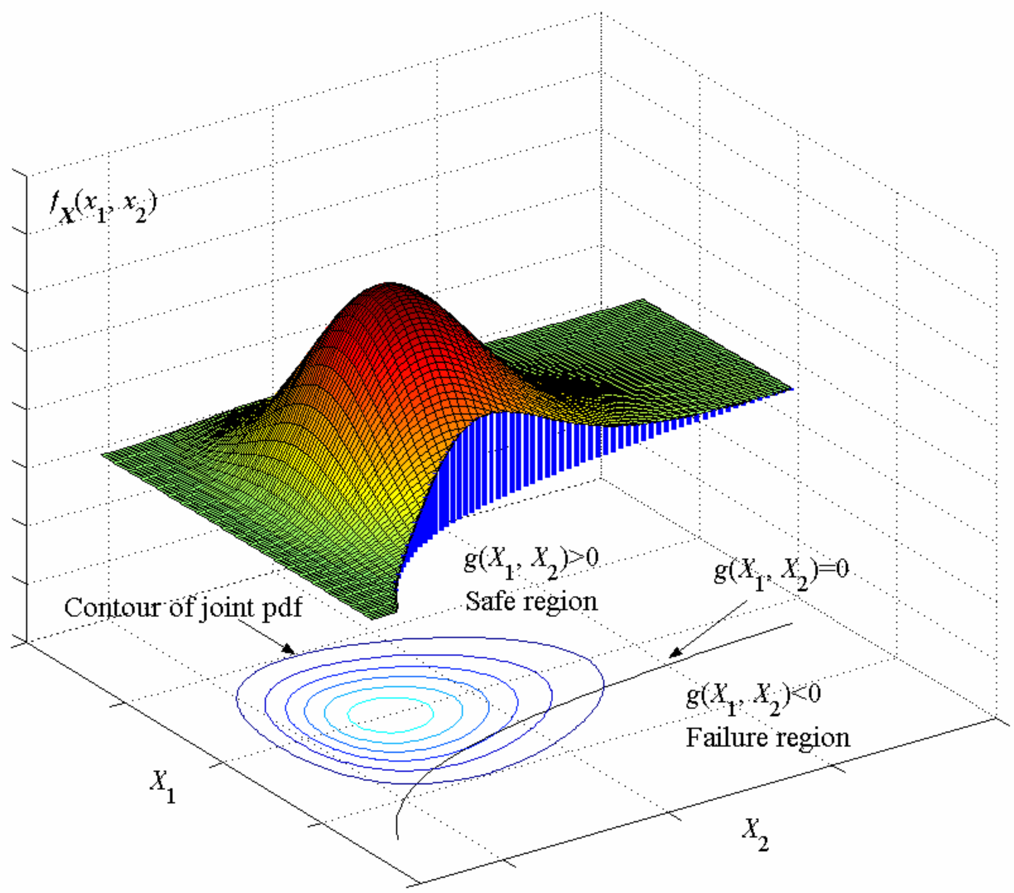
Variance reducing methodologies

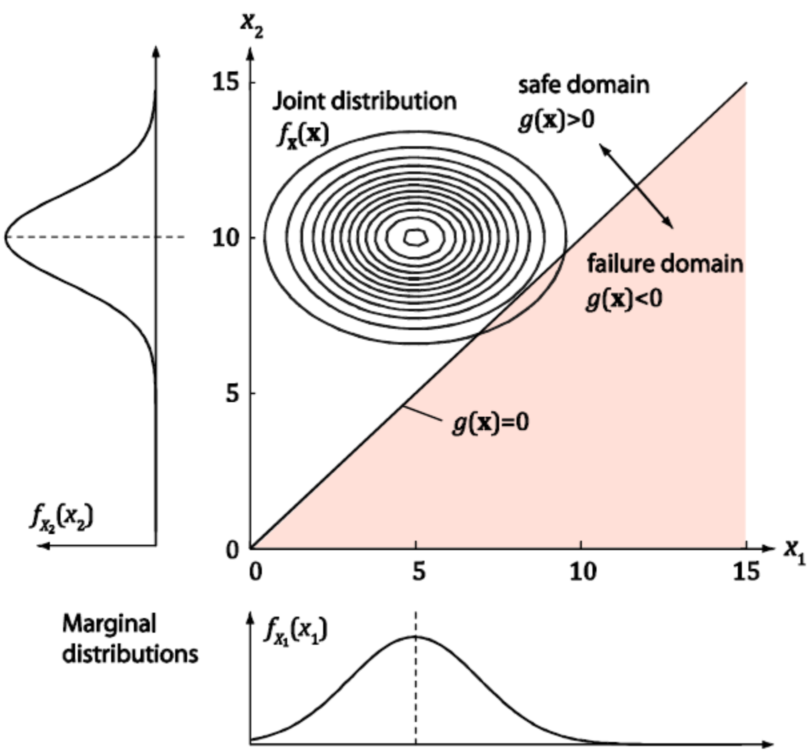
All these methods have disadvantages:

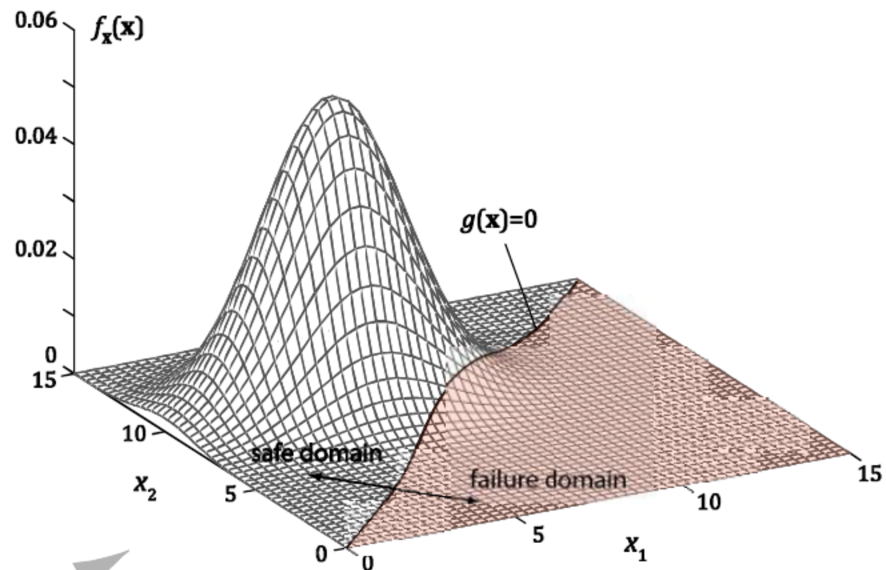
- Programming effort/modeling effort increases significantly,
- Transparency of the model and your program code decreases
- Crude Monte Carlo often still necessary to check your result.
- You buy calculation efficiency by introducing more efforts

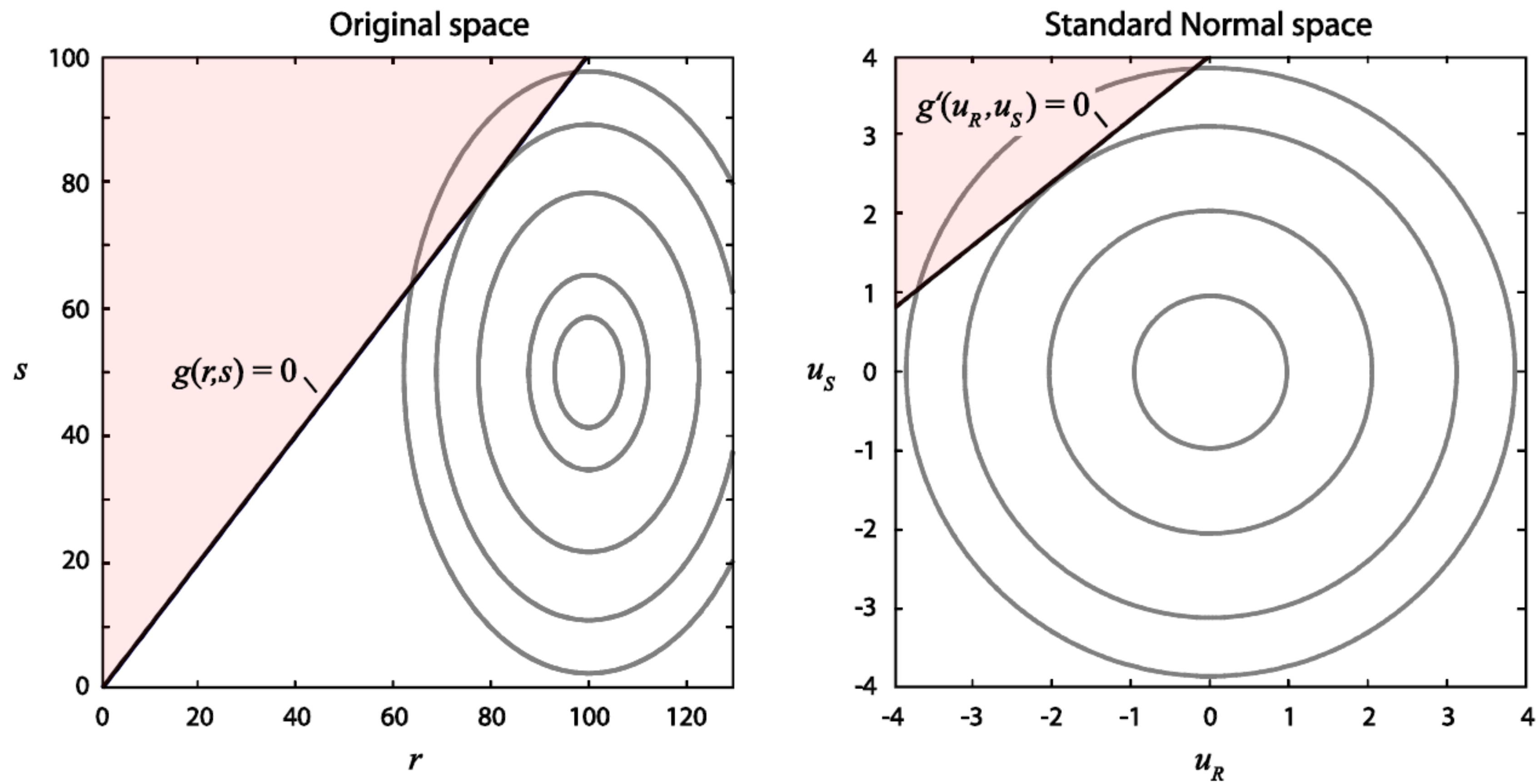
- Ask yourself: Does it make sense to spend more time on modeling/programming - How often will the model be used?

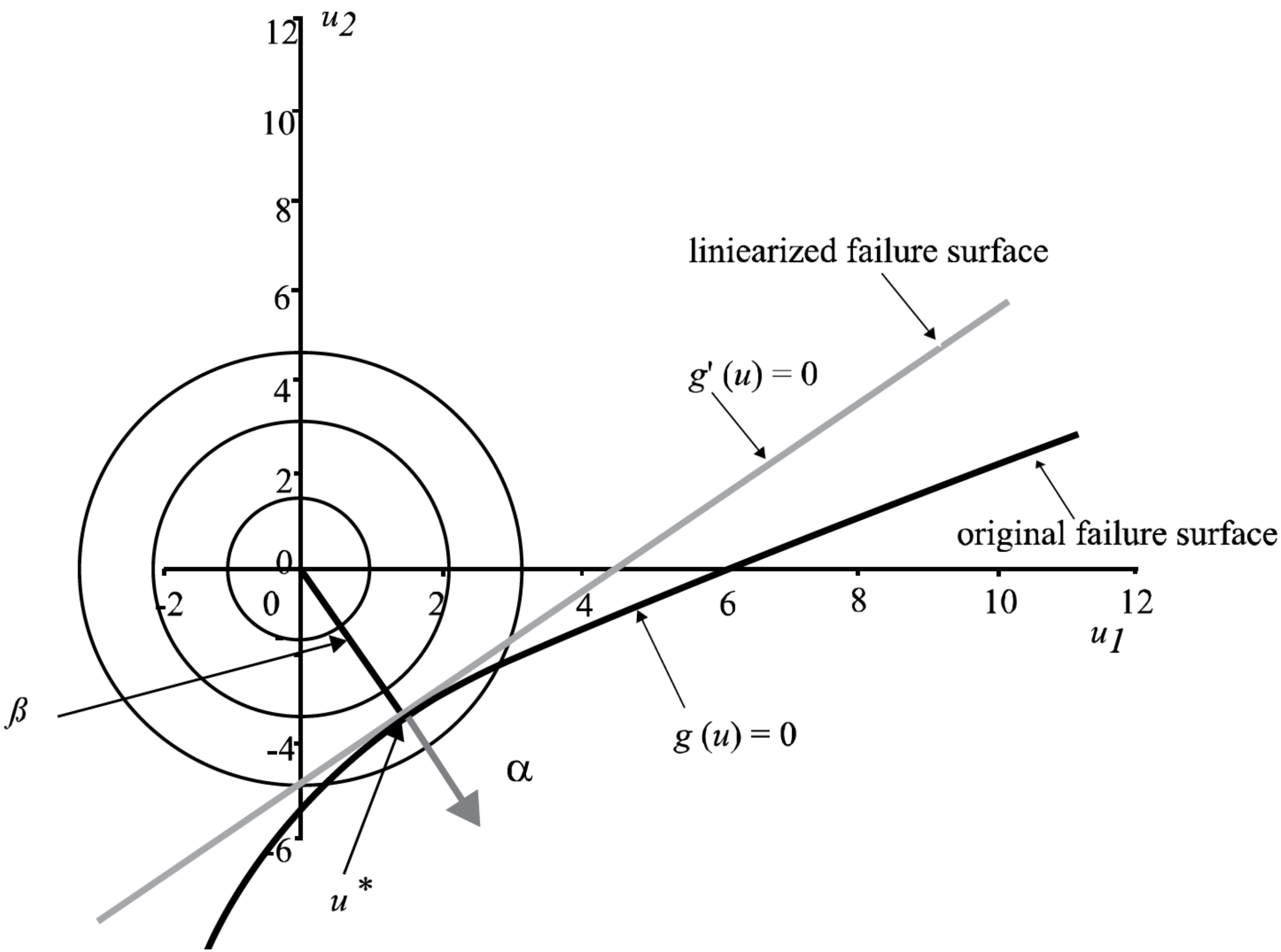
First Order Reliability Analysis - FORM



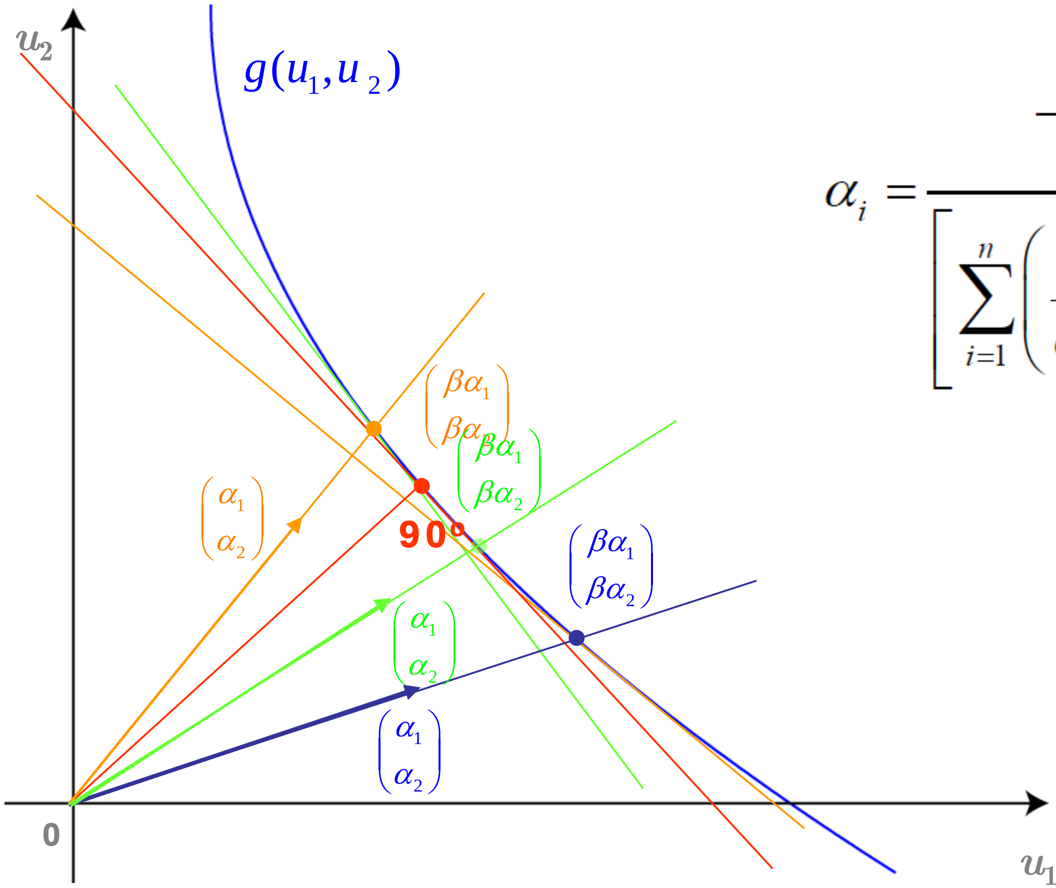

x2
15
10
5
0
Joint distribution
fX(x)
safe domain
g(x)>0
failure domain
g(x)<0
g(x)=0
fX2(x2)
x1
0
5
10
15
Marginal distributions
fX1(x1)


0.06
fX(x)
0.04
0.02
0
15
10
x2
5
0
g(x)=0
safe domain
failure domain
0
5
10
15
x1

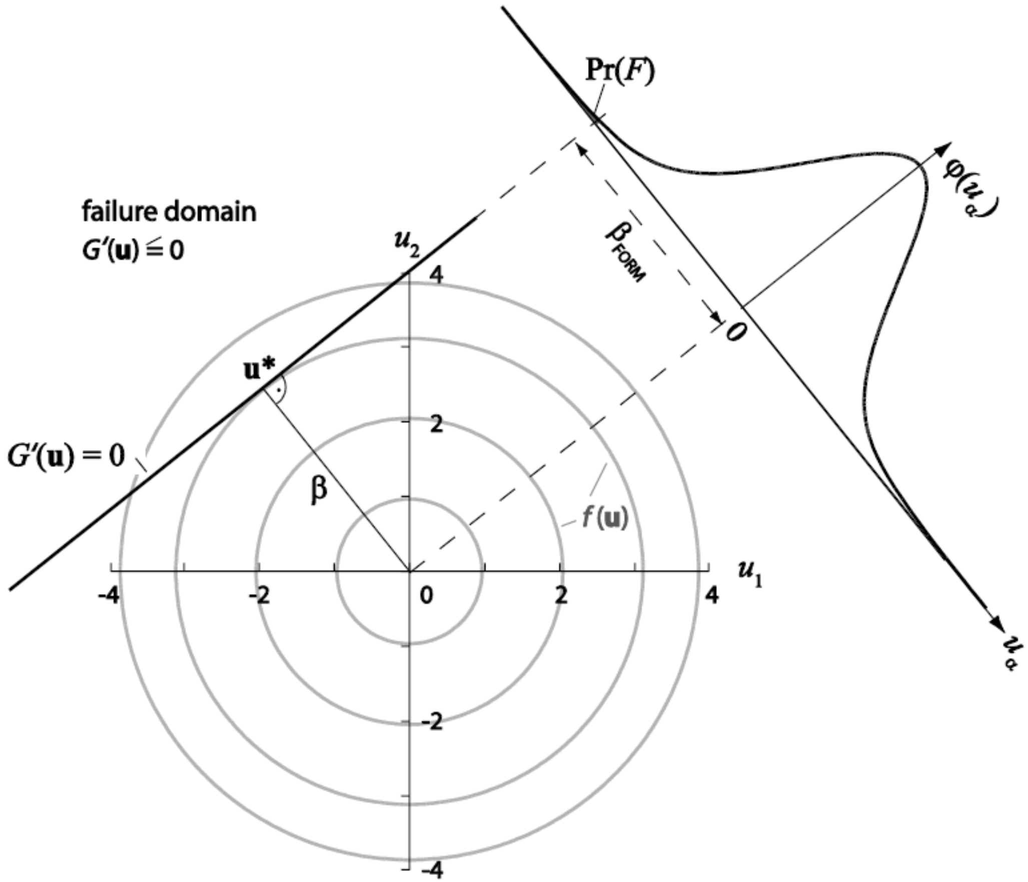

Original space
100
80
60
40
20
0
s
g(r,s) = 0
0
20
40
60
80
100
120
r
Standard Normal space
4
3
2
1
0
-1
-2
-3
-4
u_S
$g'(u_R, u_S) = 0$
-4
-3
-2
-1
0
1
2
3
4
u_R


u2
12
10
8
6
4
2
0
-4
-6
-2
0
2
4
6
8
10
12
u1
liniearized failure surface
g' (u) = 0
original failure surface
g (u) = 0
β
α
u *

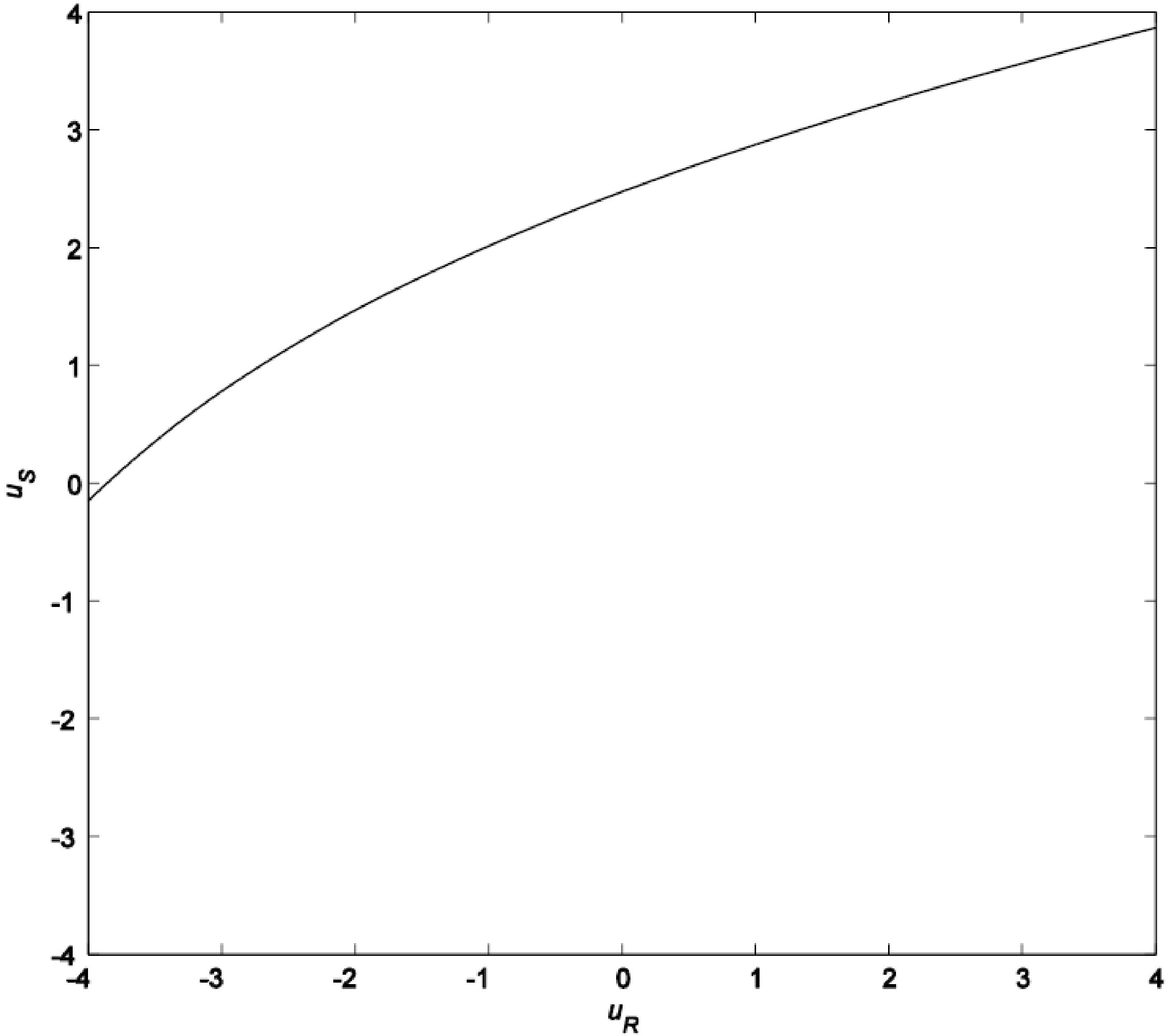
$$g(\beta \cdot \alpha_1, \beta \cdot \alpha_2, \ldots \beta \cdot \alpha_n) = 0$$



$$\alpha_i = \frac{-\dfrac{\partial g}{\partial u_i}(\beta \cdot \boldsymbol{\alpha})}{\left[\displaystyle\sum_{i=1}^{n}\left(\dfrac{\partial g}{\partial u_i}(\beta \cdot \boldsymbol{\alpha})\right)^2\right]^{1/2}}, \quad i = 1, 2, ..n$$



Pr(F)
φ(u_α)
failure domain
G′(u) ≤ 0
u₂
β_FORM
0
u*
G′(u) = 0
β
f(u)
u₁
u_α
-4
-2
0
2
4


4
3
2
1
0
-1
-2
-3
-4
u_S
-4
-3
-2
-1
0
1
2
3
4
u_R

First Order Reliability Method FORM

Advantage:

- Possibility to approximate the failure probability.
- For small probability of failures and approximate linear limit state function FORM is relative exact.
- Converges fast → computational inexpensive.

Disadvantage:

- If the number of random variables increases the performed operations increase (numerical derivations).
- Inexact if the probability of failure is large and the limit state function is highly nonlinear.