Simulation and Optimization Methods for Reliability Analysis

M. Oberguggenberger, M. Prackwieser, M. Schwarz

University of Innsbruck, Department of Engineering Science

INTALES GmbH Engineering Solutions

Innsbruck/Natters, Austria

WOST 2013, Weimar, November 21 – 22, 2013

Weimar, November 21, 2013

Plan of Presentation

Plan of Presentation

- Description of the model
  - Brute force Monte Carlo benchmarking
  - Quality assessment of estimators by resampling and Bayes

- Description of the model
  - Brute force Monte Carlo benchmarking
  - Quality assessment of estimators by resampling and Bayes
- Cost efficient reliability analysis
  - Subset simulation
  - Asymptotic sampling
  - Sensitivity based importance sampling

Plan of Presentation

- Description of the model
  - Brute force Monte Carlo benchmarking
  - Quality assessment of estimators by resampling and Bayes
- Cost efficient reliability analysis
  - Subset simulation
  - Asymptotic sampling
  - Sensitivity based importance sampling
- Optimization and worst case scenarios
  - Genetic algorithm
  - Nelder-Mead with constraints
  - optiSLang optimization

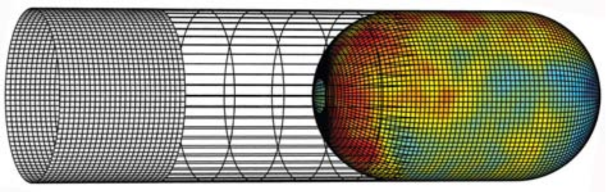
Plan of Presentation

- Description of the model
  - Brute force Monte Carlo benchmarking
  - Quality assessment of estimators by resampling and Bayes
- Cost efficient reliability analysis
  - Subset simulation
  - Asymptotic sampling
  - Sensitivity based importance sampling
- Optimization and worst case scenarios
  - Genetic algorithm
  - Nelder-Mead with constraints
  - optiSLang optimization
- Structural uncertainties – random field modelling
  - Turning the random field on
  - Observing change of output distribution

Plan of Presentation

- Description of the model
  - Brute force Monte Carlo benchmarking
  - Quality assessment of estimators by resampling and Bayes
- Cost efficient reliability analysis
  - Subset simulation
  - Asymptotic sampling
  - Sensitivity based importance sampling
- Optimization and worst case scenarios
  - Genetic algorithm
  - Nelder-Mead with constraints
  - optiSLang optimization
- Structural uncertainties – random field modelling
  - Turning the random field on
  - Observing change of output distribution

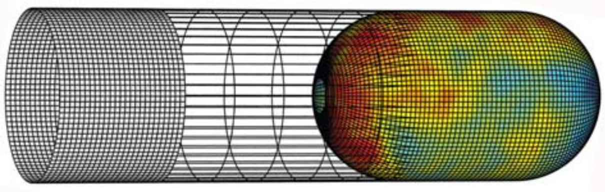
Thanks to: Dynardo GmbH, INTALES, CTU Prague, Astrium Ottobrunn



Small launcher model:

FE-model, ABAQUS, 18.000 elements, shell elements and beam elements for stiffeners. 91.000 DoF.

Description of the Model

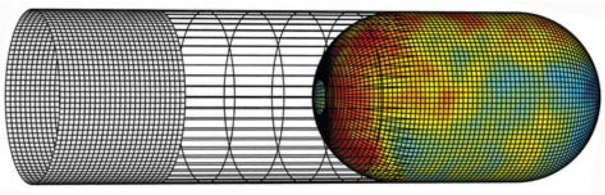


Small launcher model:

FE-model, ABAQUS, 18.000 elements, shell elements and beam elements for stiffeners. 91.000 DoF.

35 Input variables **x**: E-moduli, yield stresses; pressure-, temperature-, booster loads – applied stepwise; forces due to boundary imperfections.

Description of the Model



Small launcher model:

FE-model, ABAQUS, 18.000 elements, shell elements and beam elements for stiffeners. 91.000 DoF.

35 Input variables **x**: E-moduli, yield stresses; pressure-, temperature-, booster loads – applied stepwise; forces due to boundary imperfections.

Input statistics: Uniform distributions with spread $\pm 15\%$ around nominal value.

Failure Criterion

Output: Failure probability

$$p_f = \mathrm{P}(\Phi(\mathbf{x}) > 1)$$

defined by the the critical demand-to-capacity ratio (CDCR)

$$\Phi(\mathbf{x}) = \max\left\{\frac{PEEQ(\mathbf{x})}{0.07}, \frac{SP(\mathbf{x})}{180}, \frac{0.001}{EV(\mathbf{x})}\right\},$$

combining 3 failure criteria (plastification of metallic part, rupture of composite part, buckling).

Failure Criterion

Output: Failure probability

$$p_f = \mathrm{P}(\Phi(\mathbf{x}) > 1)$$

defined by the the critical demand-to-capacity ratio (CDCR)

$$\Phi(\mathbf{x}) = \max\left\{\frac{PEEQ(\mathbf{x})}{0.07}, \frac{SP(\mathbf{x})}{180}, \frac{0.001}{EV(\mathbf{x})}\right\},$$

combining 3 failure criteria (plastification of metallic part, rupture of composite part, buckling).

Benchmarking by brute force Monte Carlo, $N = 5000$, parallelized, performed on the HPC computer LEO III in Innsbruck.

Output: Failure probability

$$p_f = \mathrm{P}(\Phi(\mathbf{x}) > 1)$$

defined by the the critical demand-to-capacity ratio (CDCR)

$$\Phi(\mathbf{x}) = \max\left\{\frac{PEEQ(\mathbf{x})}{0.07}, \frac{SP(\mathbf{x})}{180}, \frac{0.001}{EV(\mathbf{x})}\right\},$$

combining 3 failure criteria (plastification of metallic part, rupture of composite part, buckling).

Benchmarking by brute force Monte Carlo, $N = 5000$, parallelized, performed on the HPC computer LEO III in Innsbruck.

Result of benchmark simulation:

$$p_f = 0.0116.$$

How good is the Monte Carlo estimator for p_f?

How good is the Monte Carlo estimator for $\mathbf{p_f}$?

First analysis: Bootstrap resampling. Drawing from the original sample of size $N = 5000$ with replacement, $B = 10000$ samples with the same (empirical) distribution and corresponding p_f are obtained. Result: an estimate of the statistical variation of p_f.

How good is the Monte Carlo estimator for p_f?

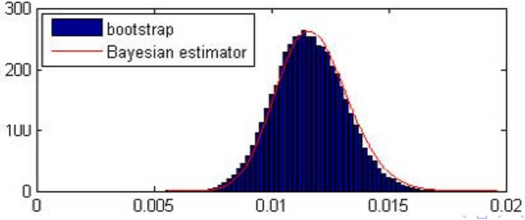
First analysis: Bootstrap resampling. Drawing from the original sample of size $N = 5000$ with replacement, $B = 10000$ samples with the same (empirical) distribution and corresponding p_f are obtained. Result: an estimate of the statistical variation of p_f.

Second analysis: Bayesian posterior density for p_f, given the data constituted by the original sample. The MC-sample can be viewed as the outcome of a Bernoulli experiment (failed/not failed). The posterior density is an N-fold product of beta distributions.

How good is the Monte Carlo estimator for $\mathbf{p_f}$?

First analysis: Bootstrap resampling. Drawing from the original sample of size $N = 5000$ with replacement, $B = 10000$ samples with the same (empirical) distribution and corresponding p_f are obtained. Result: an estimate of the statistical variation of p_f.

Second analysis: Bayesian posterior density for p_f, given the data constituted by the original sample. The MC-sample can be viewed as the outcome of a Bernoulli experiment (failed/not failed). The posterior density is an N-fold product of beta distributions.



Variation of p_f around its estimated value of 0.0116.

Reliability Analysis: Faster Methods (1)

Subset Simulation:

$$\mathrm{P}(F) = \mathrm{P}(F_m \big| F_{m-1})\mathrm{P}(F_{m-1} \big| F_{m-2}) \ldots \mathrm{P}(F_1 \big| F_0)\mathrm{P}(F_0)$$

where $F = F_m$ and F_0 is the starting region.

$$F = \{\mathbf{x} : \Phi(\mathbf{x}) > 1\}, \qquad F_i = \{\mathbf{x} : \Phi(\mathbf{x}) > \alpha_i\}$$

and α_i is chosen so that $\mathrm{P}(F_i \big| F_{i-1}) = 0.2$, say.

$\mathrm{P}(F_0)$ is estimated by brute Monte Carlo, $\mathrm{P}(F_i \big| F_{i-1})$ by starting short Markov chains at the worst 20% of obtained points.

Subset Simulation:

$$\mathrm{P}(F) = \mathrm{P}(F_m \big| F_{m-1})\mathrm{P}(F_{m-1} \big| F_{m-2}) \ldots \mathrm{P}(F_1 \big| F_0)\mathrm{P}(F_0)$$

where $F = F_m$ and F_0 is the starting region.

$$F = \{\mathbf{x} : \Phi(\mathbf{x}) > 1\}, \qquad F_i = \{\mathbf{x} : \Phi(\mathbf{x}) > \alpha_i\}$$

and α_i is chosen so that $\mathrm{P}(F_i \big| F_{i-1}) = 0.2$, say.

$\mathrm{P}(F_0)$ is estimated by brute Monte Carlo, $\mathrm{P}(F_i \big| F_{i-1})$ by starting short Markov chains at the worst 20% of obtained points.

Asymptotic sampling: $p_f = \Phi(-\beta)$, $\beta = \Phi^{-1}(1 - p_f)$. Transformation to normal probability space with $\sigma = 1$. Instead of simulating $\beta = \beta(1)$, one simulates $\beta(v)$ for smaller values of $v = 1/\sigma$, which is easy, and sets up a regression

$$\beta(v) = A + Bv + C/v + \ldots$$

Best model chosen by data analysis.

Reliability Analysis: Faster Methods (2)

Importance sampling:

$$p_f = \int \mathbb{1}_F(\mathbf{x})\rho(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int \mathbb{1}_F(\mathbf{x})\frac{\rho(\mathbf{x})}{g(\mathbf{x})}g(\mathbf{x})\,\mathrm{d}\mathbf{x}.$$

The density $g(\mathbf{x})$ is shifted into a neighborhood of the failure region.

MC simulation with a sample distributed according to $g(\mathbf{x})$.

Importance sampling:

$$p_f = \int \mathbb{1}_F(\mathbf{x})\rho(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int \mathbb{1}_F(\mathbf{x})\frac{\rho(\mathbf{x})}{g(\mathbf{x})}g(\mathbf{x})\,\mathrm{d}\mathbf{x}.$$

The density $g(\mathbf{x})$ is shifted into a neighborhood of the failure region.

MC simulation with a sample distributed according to $g(\mathbf{x})$.

How to choose $g(\mathbf{x})$? Start with cheap sensitivity analysis.

Employing all tricks of the trade (Latin hypercube sampling, correlation control), a sample size around 100 – 200 suffices.

Determine the most relevant input parameters.

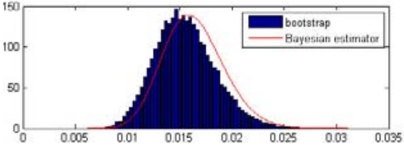
Distort their distribution according to the degree of correlation with the output (assuming monotone dependence).

Reliability Analysis: Comparison

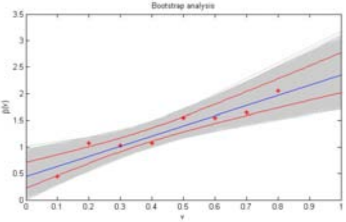
	MC	SS	AS	IS
Sample Size	5000	780	800	780
Estimated p_f	0.0116	0.0155	0.0093	0.0084
Bootstrap 95%-CI	0.0088 — 0.0146	0.0104 — 0.0217	0.0028 — 0.0219	0.0055 — 0.0118
Bayesian 95%-CI	0.0090 — 0.0150	0.0112 — 0.0225		

Reliability Analysis: Comparison

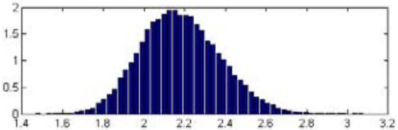
	MC	SS	AS	IS
Sample Size	5000	780	800	780
Estimated p_f	0.0116	0.0155	0.0093	0.0084
Bootstrap 95%-CI	0.0088 — 0.0146	0.0104 — 0.0217	0.0028 — 0.0219	0.0055 — 0.0118
Bayesian 95%-CI	0.0090 — 0.0150	0.0112 — 0.0225		



Subset simulation: Bootstrap and Bayesian estimate of the variability of p_f.



Asymptotic sampling: Bootstrap regression and corresponding distribution of β.



Derivative-free methods:

Derivative-free methods:

- Genetic algorithms. An initial set of points is improved (with respect to the value of the objective function) by randomly changing coordinates and interchanging components. When a local optimum has been identified, a restart is undertaken to cover other regions of the search space.

Derivative-free methods:

- Genetic algorithms. An initial set of points is improved (with respect to the value of the objective function) by randomly changing coordinates and interchanging components. When a local optimum has been identified, a restart is undertaken to cover other regions of the search space.
- Particle swarms. Similar to genetic algorithms, but the initial set of points is steered towards an optimum by means of velocity fields.

Derivative-free methods:

- Genetic algorithms. An initial set of points is improved (with respect to the value of the objective function) by randomly changing coordinates and interchanging components. When a local optimum has been identified, a restart is undertaken to cover other regions of the search space.
- Particle swarms. Similar to genetic algorithms, but the initial set of points is steered towards an optimum by means of velocity fields.
- Nelder-Mead downhill simplex algorithm. An initial simplex of points is distorted and moved by reflection, expansion, contraction, reduction. Probabilistic restart. ■

Derivative-free methods:

- Genetic algorithms. An initial set of points is improved (with respect to the value of the objective function) by randomly changing coordinates and interchanging components. When a local optimum has been identified, a restart is undertaken to cover other regions of the search space.
- Particle swarms. Similar to genetic algorithms, but the initial set of points is steered towards an optimum by means of velocity fields.
- Nelder-Mead downhill simplex algorithm. An initial simplex of points is distorted and moved by reflection, expansion, contraction, reduction. Probabilistic restart. ■

In all cases, the implementation of bounds (on input) and constraints (on output) requires additional rules.

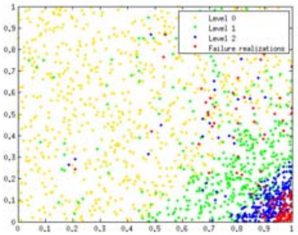
First application: In reliability analysis, the location of the failure region and the most critical points are of interest.

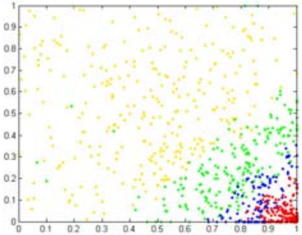
Worst Case Scenarios

First application: In reliability analysis, the location of the failure region and the most critical points are of interest.

All algorithms compute clouds of points that can be ordered according to their Φ-values and used for further analysis.

Subset Simulation (top) and genetic algorithm (bottom), pressure load sphere 2 versus yield stress cylinder 3.





Legend

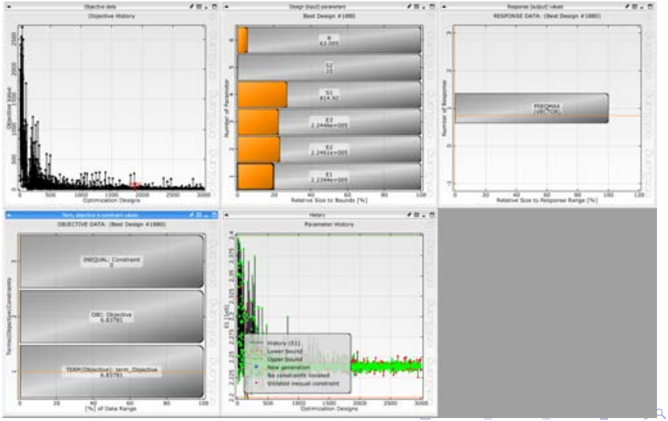
yellow	0 — 0.9543
green	0.9544 — 0.9885
blue	0.9886 — 1
red	> 1

- GRADE algorithm with CERAF restart strategy (CTU Prague)
- Nelder-Mead with probabilistic restart (in-house)
- Genetic algorithm optiSLang
- Particle swarm algorithm optiSLang

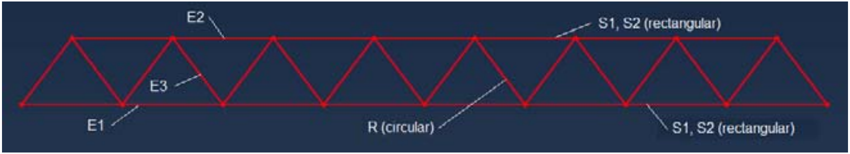
Mass Optimization under Constraint

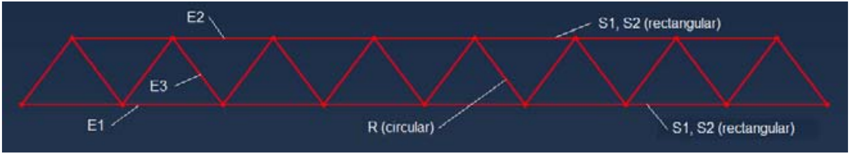
- GRADE algorithm with CERAF restart strategy (CTU Prague)
- Nelder-Mead with probabilistic restart (in-house)
- Genetic algorithm optiSLang
- Particle swarm algorithm optiSLang

optiSLAng-Gui particle swarm



Truss Structure – Comparison of Results

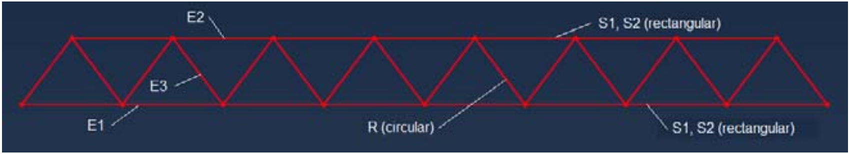




Objective value: MASS
Constraint: PEEQ = 0

Bounds:

S1	S2	R	E1	E2	E3
10	10	10	220000	220000	220000
3000	3000	1000	240001	240001	240001



Objective value: MASS

Constraint: PEEQ $= 0$

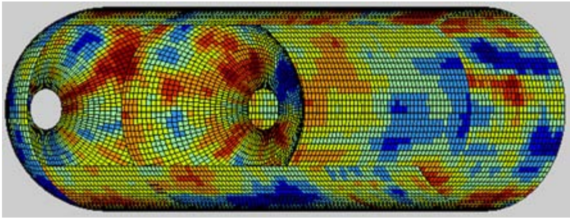
Bounds:

S1	S2	R	E1	E2	E3
10	10	10	220000	220000	220000
3000	3000	1000	240001	240001	240001

	Function calls	Mass	S1	S2	R	E1	E2	E3
GA - GRADE	1620	6.98346	828.30	10.00	63.78	240001	222901	220000
Nelder-Mead	349	6.80596	799.27	10.11	62.95	223330	238207	228351
GA - optiSLang	1528	6.85535	817.40	10.00	63.07	229650	220620	220960
PS - optiSLang	1996	6.08253	730.28	10.00	59.26	223280	229500	224290

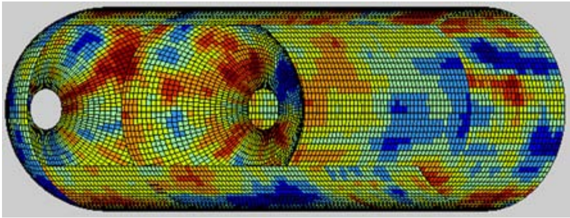
Random Field Modelling (1)

A random field on the small launcher model (material properties)



Random Field Modelling (1)

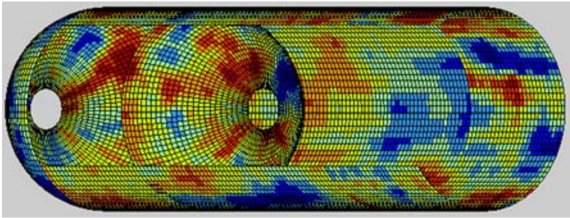
A random field on the small launcher model (material properties)



- Determination of field parameters from empirical data
- spectral decomposition of covariance matrix
- Monte Carlo simulation of the random field (Karhunen-Loève)
- repeated FE-calculation of structural response

Random Field Modelling (1)

A random field on the small launcher model (material properties)



- Determination of field parameters from empirical data
- spectral decomposition of covariance matrix
- Monte Carlo simulation of the random field (Karhunen-Loève)
- repeated FE-calculation of structural response

Typical autocovariance function

$$\mathrm{COV}(X_P, X_Q) = \sigma^2 \exp(-\mathrm{dist}_1(P,Q)/\ell_1) \exp(-\mathrm{dist}_2(P,Q)/\ell_2)$$

Random Field Modelling (2)

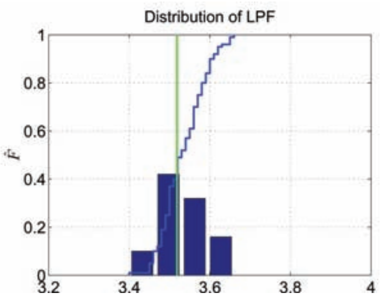
Exemplary application:

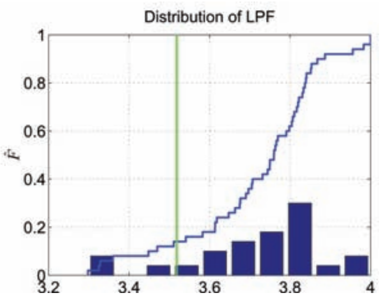
- Loads as random variables as before
- Material properties as random fields
- **Question:** Change of output with/without random field

Exemplary application:

- Loads as random variables as before
- Material properties as random fields
- **Question:** Change of output with/without random field

Example – change of distribution of load proportionality factor LPF without random field (left) and with random field (right):





Thank you for your attention!