

Bewertung von Optimierungs- und Zuverlässigkeitsalgorithmen für die virtuelle Produktauslegung

Dr.-Ing. Andreas Plotzita, PT/EST4
29. November 2007



Inhalt Bewertung Optimierungs + Zuverlässigkeitsalgorithmen für virtuelle Produktauslegung

1. Motivation: Virtuelle Produktauslegung am Beispiel eines Hammer
 - 1.1 Ziele und Zielgrößen des Marktes
 - 1.2 Aufbau und Funktionsweise eines Hammers
2. Optimierung
 - 2.1 Problemstellung für numerische Hammersimulation
 - 2.2 Optimierungsverfahren
 - 2.3 Anwendung der Verfahren und Ergebnisse
3. Zuverlässigkeitsanalyse
 - 3.1 Fertigungstoleranzen und Problemstellung
 - 3.2 Grundlagen der Zuverlässigkeitsanalyse
 - 3.3 Anwendung der Verfahren und Ergebnisse
4. Zusammenfassung
5. Ausblick



1.1 Ziele und Zielgrößen

Ziele aus dem Markt

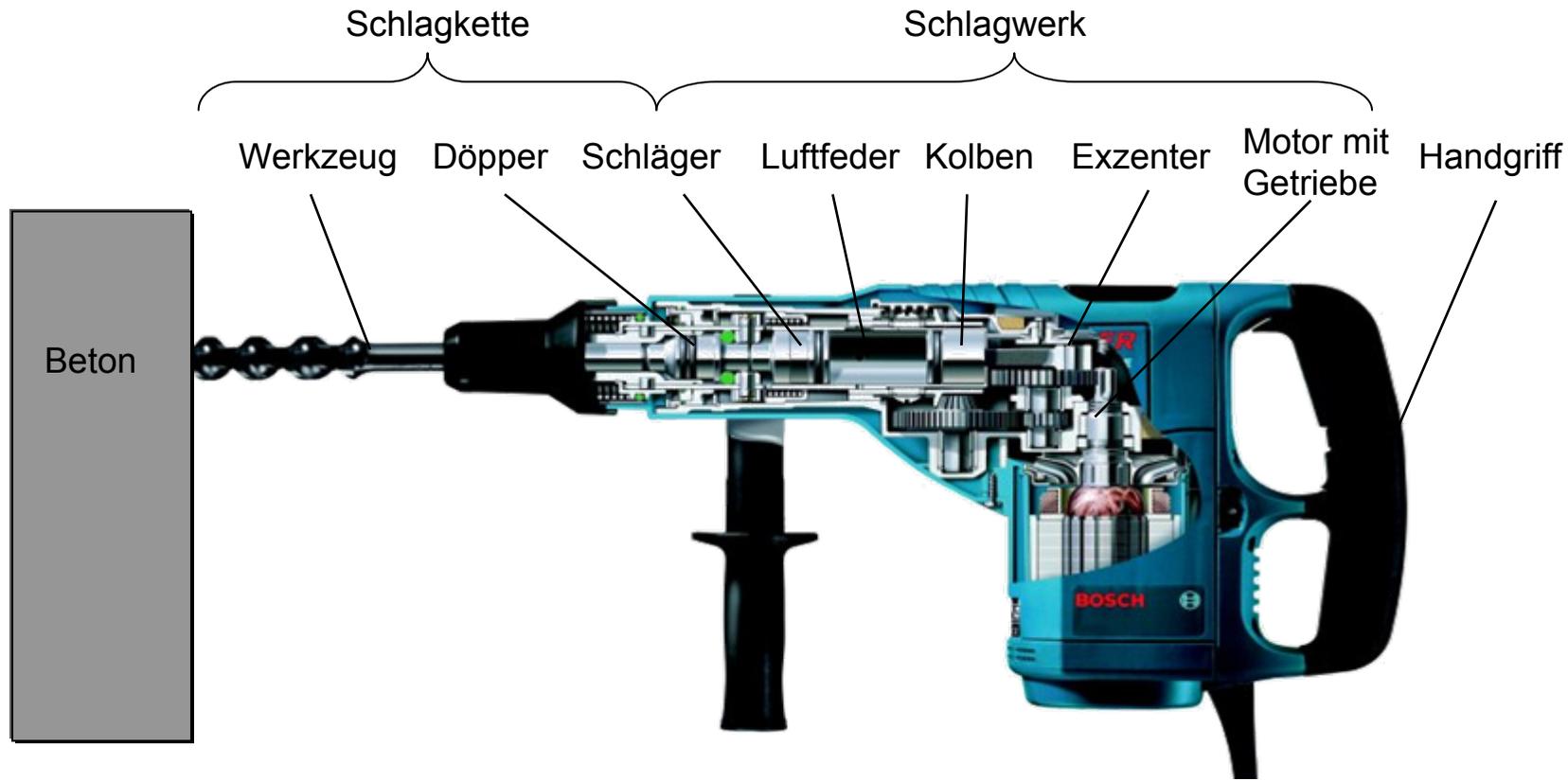
**Anpassung eines Hammers an Kundenwünsche
und
Senkung der Kosten**

Zielgrößen aus dem Markt

- Erhöhung der Performance, d.h. der Meißeleindringung
- Senkung der Vibrationsbelastung des Anwenders
- Verkürzung der Entwicklungszeit des Produkts

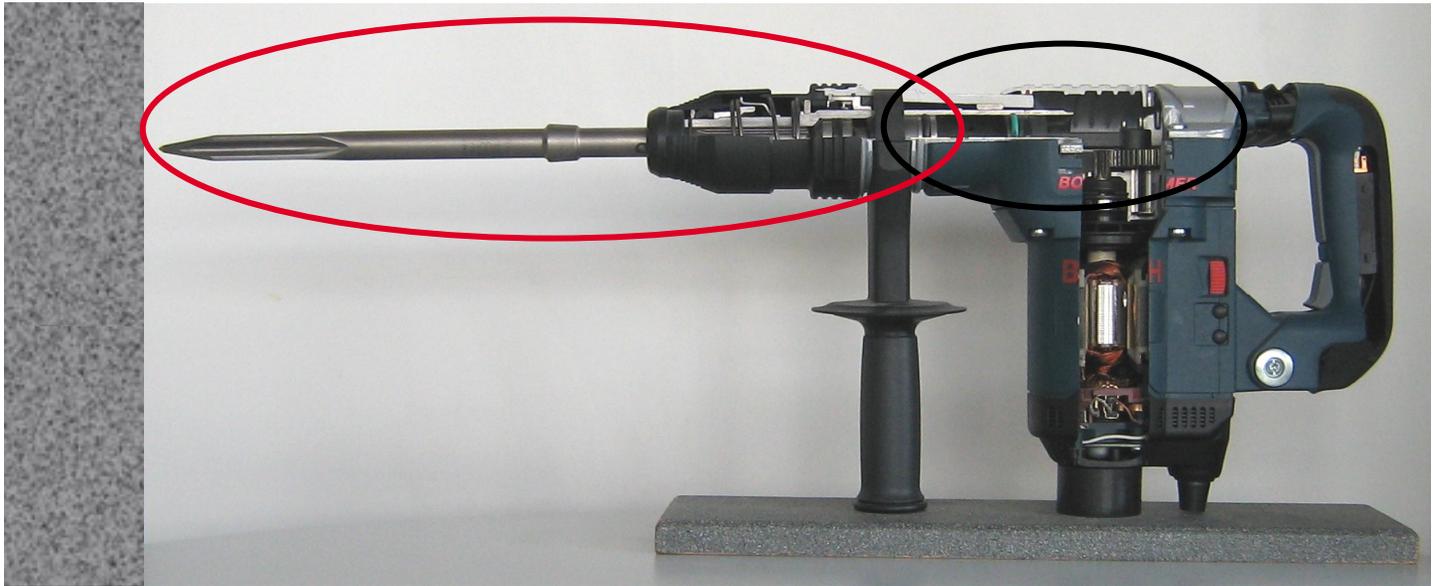
1.2 Aufbau und Funktionsweise eines Hammers

Motivation für die virtuelle Produktauslegung am Beispiel der Hämmer



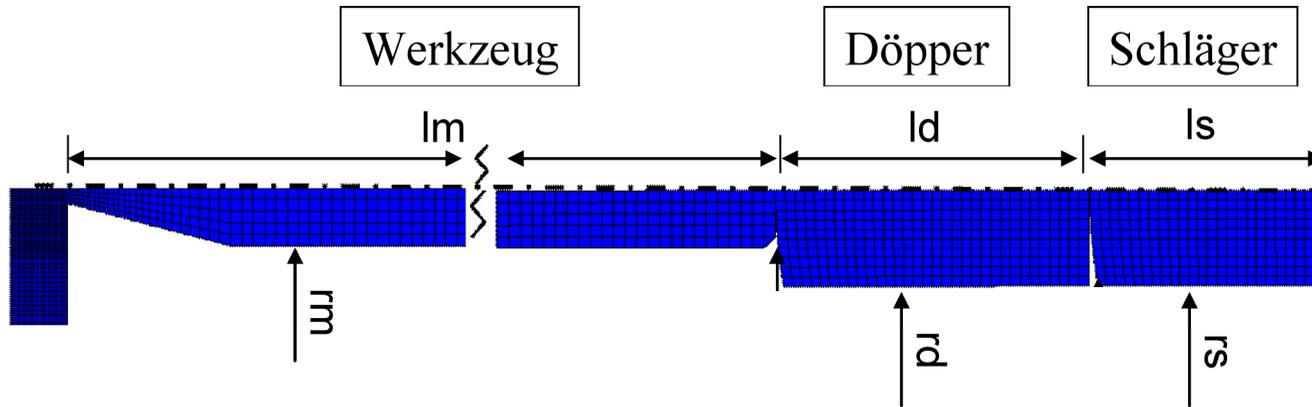
2.1 Optimierung – Problemstellung

Realer Schlaghammer



2.1 Optimierung – Problemstellung

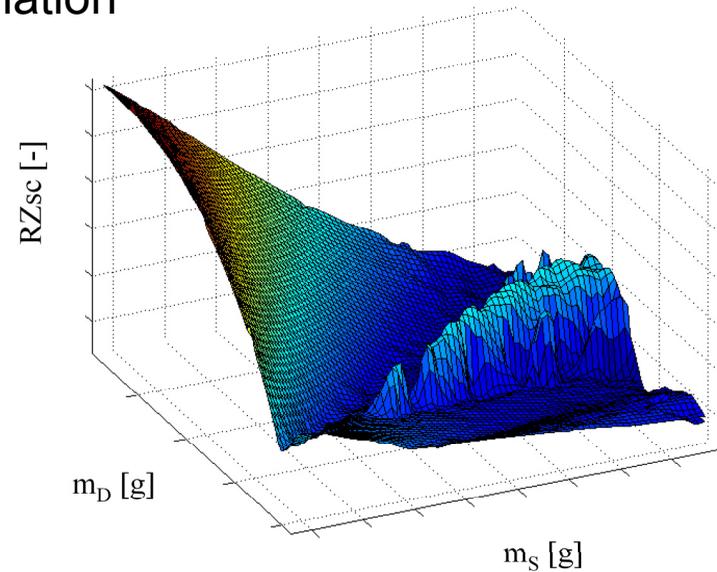
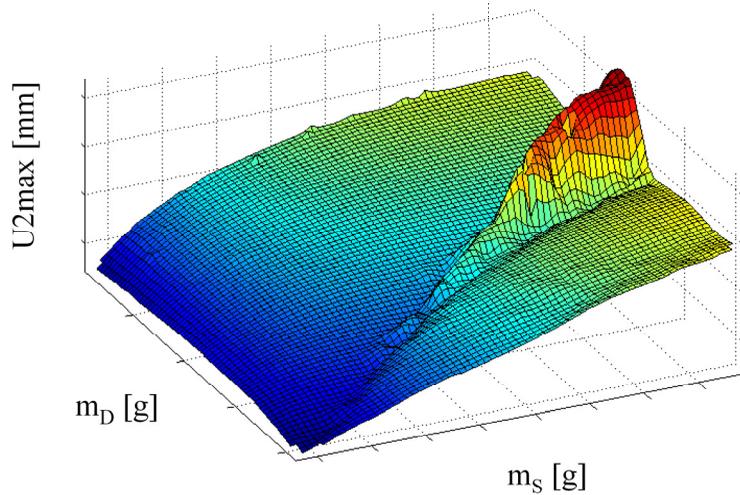
Schlagkette – numerische Hammersimulation



Zielgrößen:
$$U2 \max = f_1(m_S, m_D)$$
$$RZsc = f_2(m_S, m_D)$$
 } gilt für ein Werkzeug,
z.B. Spitzmeißel

2.1 Optimierung – Problemstellung

Zielgrößen – numerische Hammersimulation



Zielfunktion:

$$obj_1(\mathbf{m}) = -(w_1 \cdot U2_{max}(\mathbf{m}) + w_2 \cdot RZ_{sc}(\mathbf{m}))$$

2.2 Optimierung – Optimierungsverfahren

Allgemeines nichtlineares Optimierungsproblem:

$$\begin{array}{l} \min_x f(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^n \\ \text{unter} \\ c_i(\mathbf{x}) = 0 \quad \text{für } i = 1, \dots, m \\ h_j(\mathbf{x}) \leq 0 \quad \text{für } j = 1, \dots, q \end{array}$$



Bestimme n-dim. Vektor \mathbf{x}^* , so dass Zielfunktion f minimal wird, mit:
 m Gleichgewichtsbedingungen und
 q Ungleichgewichtsbedingungen

Optimierungsverfahren:

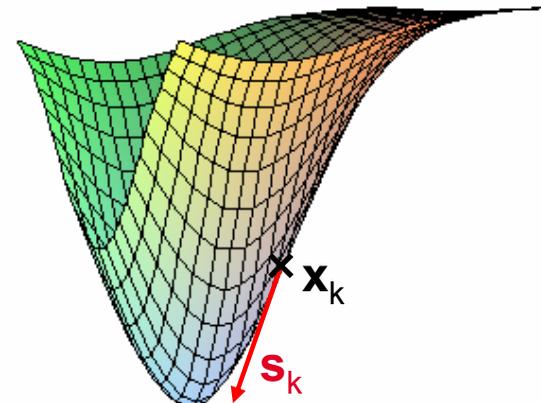
- deterministisch
Gradientenbasierte Verfahren
Approximationsverfahren
...

- stochastisch
Neuronale Netze
Evolutionäre Algorithmen
...

2.2 Optimierung – Optimierungsverfahren

Gradientenbasierte Verfahren:

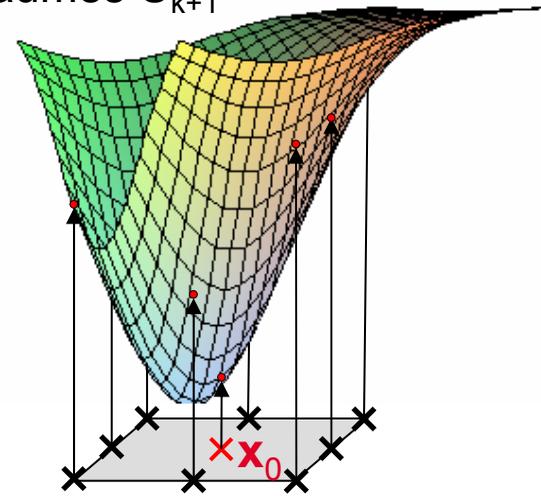
1. Wahl des Startvektors \mathbf{x}_0
2. Für $k = 0, 1, 2, \dots$
 - (i) Überprüfung der Abbruchkriterien
 - (ii) Bestimmung der Suchrichtung \mathbf{s}_k
 - (iii) Bestimmung der Schrittweite σ_k
 - (iv) Bildung von $\mathbf{x}_{k+1} = \mathbf{x}_k + \sigma_k \mathbf{s}_k$



2.2 Optimierung – Optimierungsverfahren

Adaptive Response Surface Method:

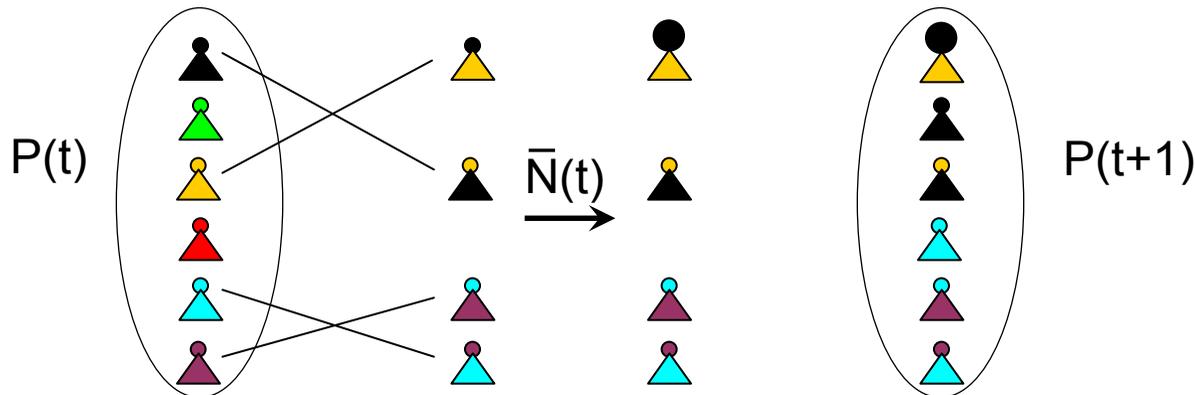
1. Wahl des Startvektors x_0 und des ersten Unterraumes U_0
2. Für $k = 0, 1, 2, \dots$
 - (i) Überprüfung der Abbruchkriterien
 - (ii) Generierung der Stützstellen in U_k
 - (iii) Berechnung der approximierten Zielfunktion $\varphi_k(\mathbf{x})$
 - (iv) Berechnung des Minimums von $\varphi_k(\mathbf{x})$
 - (v) Bestimmung des modifizierten Unterraumes U_{k+1}



2.2 Optimierung – Optimierungsverfahren

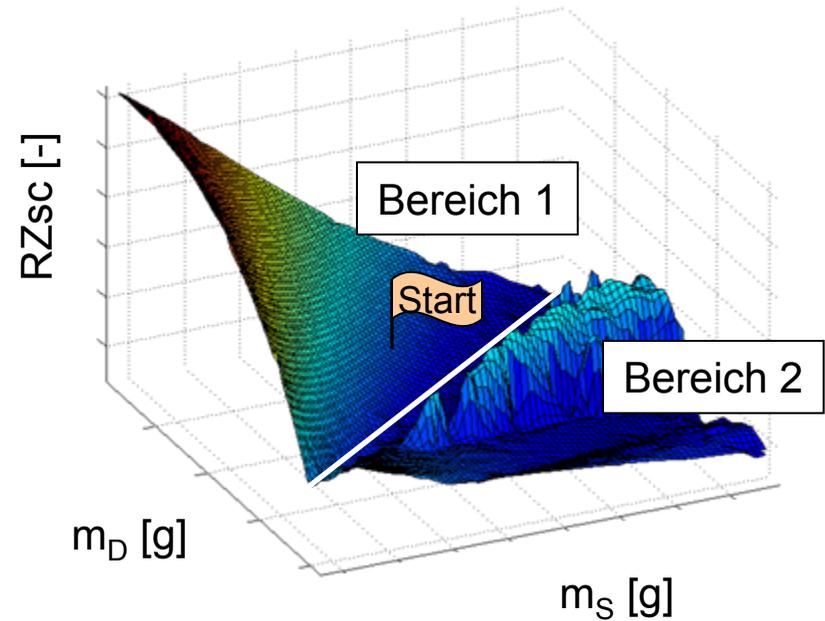
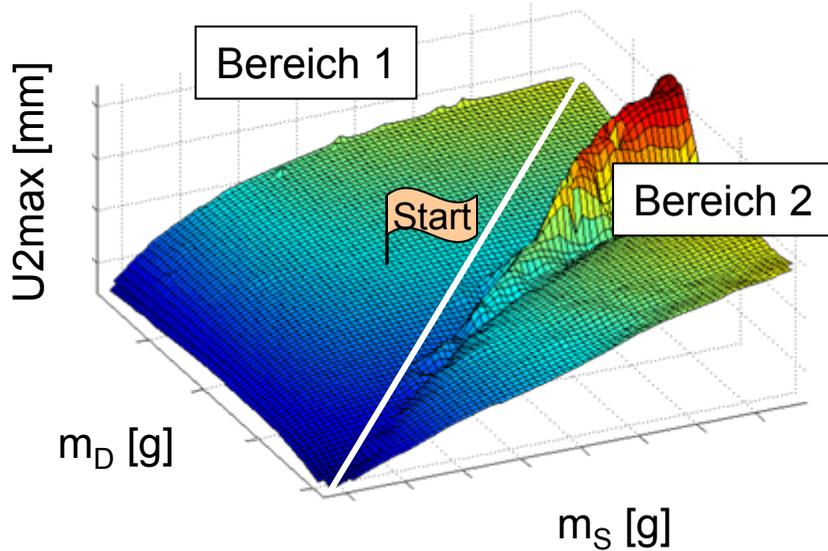
Evolutionäre Algorithmen:

1. Initialisierung der Anfangspopulation $P(0)$, $t = 0$
2. Berechnung der Zielfunktion und Bewertung der Individuen
3. Solange die Abbruchkriterien nicht erfüllt sind
 - (i) Selektion der Eltern aus der Population $P(t)$
 - (ii) Rekombination zur Erzeugung von Nachkommen $N(t)$
 - (iii) Mutation der Individuen in $N(t)$ zu $\bar{N}(t)$
 - (iv) Berechnung der Zielfunktion und Bewertung der Individuen in $\bar{N}(t)$
 - (v) Bildung der neuen Population $P(t+1)$ aus $P(t)$ und $\bar{N}(t)$, $t = t+1$
4. Bestes Individuum bildet die Lösung



2.3 Optimierung – Anwendung der Verfahren

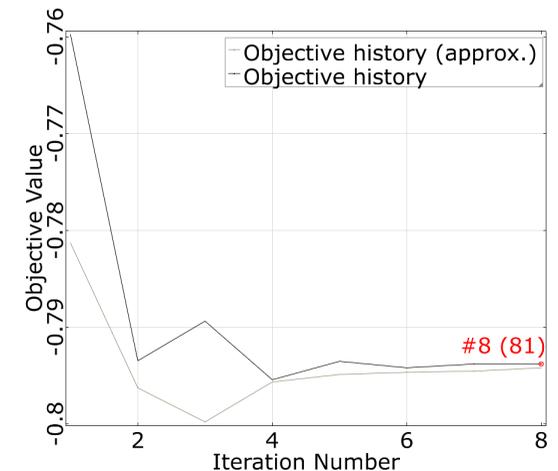
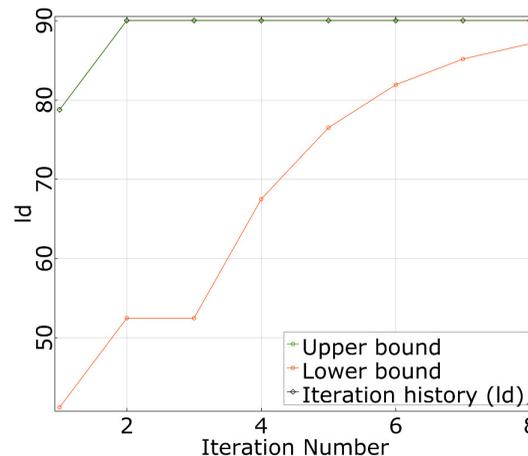
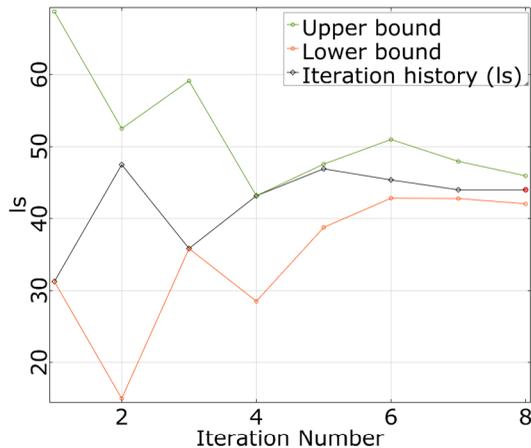
Unterteilung der Bereiche:



2.3 Optimierung – Ergebnisse (Bereich 1)

Bereich 1 – ARSM:

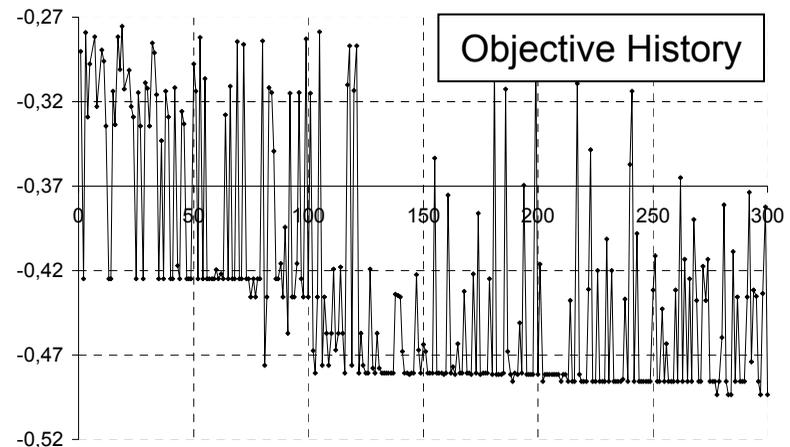
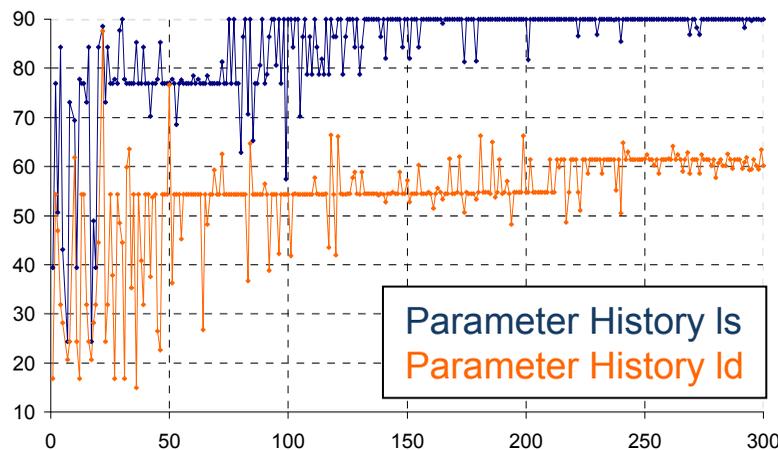
Approximationsmethode	:	quadratisch
Methode der Stützstellengenerierung	:	D-optimal
Optimierungsmethode	:	NLPQL (Gradientenverfahren)
Start Range [%]	:	50
Startpunkt \mathbf{x}_0	:	Is = 50 mm; Id = 60 mm
Aktiviere Zusatzoption	:	Wiederverwendung von Stützstellen



2.3 Optimierung – Ergebnisse (Bereich 2)

Bereich 2 – EA:

Populationsgröße	:	10
Anzahl an Generationen	:	30
Anzahl an Eltern	:	10
Rekombination / Wahrscheinlichkeit	:	uniform / 0,5
Mutationsrate	:	0,25
Standardabweichung	:	0,2 / 0,01 (Beginn / Ende)
Archiv	:	2

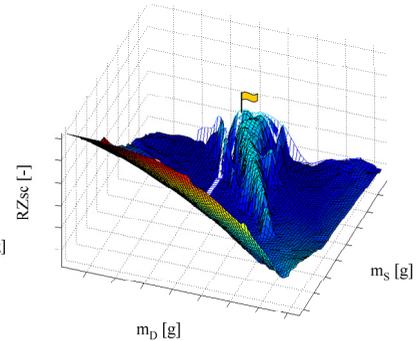
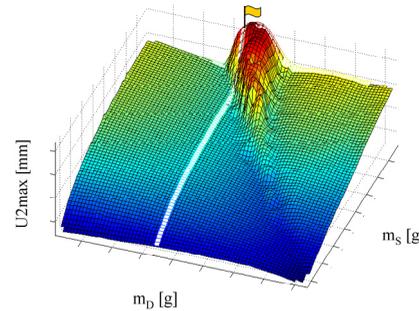
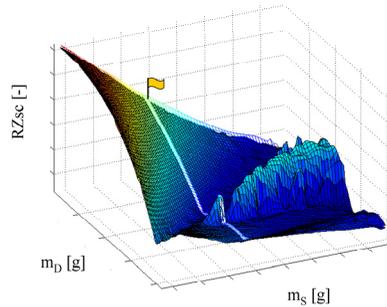
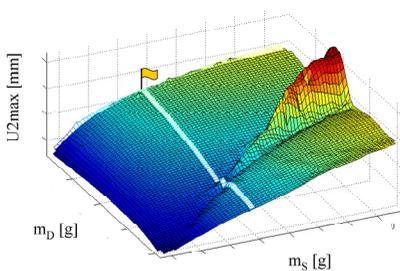
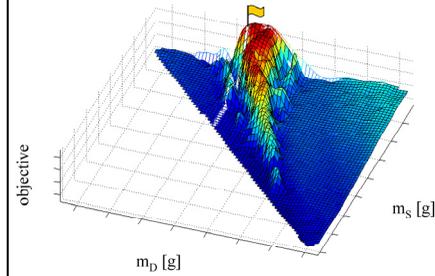
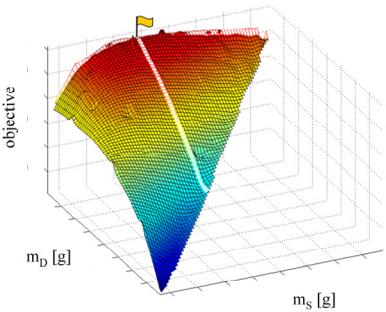


2.3 Optimierung - Ergebnisse

Bereich 1 – ARSM:

Bereich 2 – EA:

Optimum:		Optimum:	
l _s	= 43,99 mm	l _s	= 90,00 mm
l _d	= 90,00 mm	l _d	= 60,20 mm
m _S	= 239,69 g	m _S	= 494,02 g
m _D	= 494,02 g	m _D	= 329,28 g
U2max	= 0,53 mm	U2max	= 0,75 mm
RZsc	= 0,35	RZsc	= 0,23

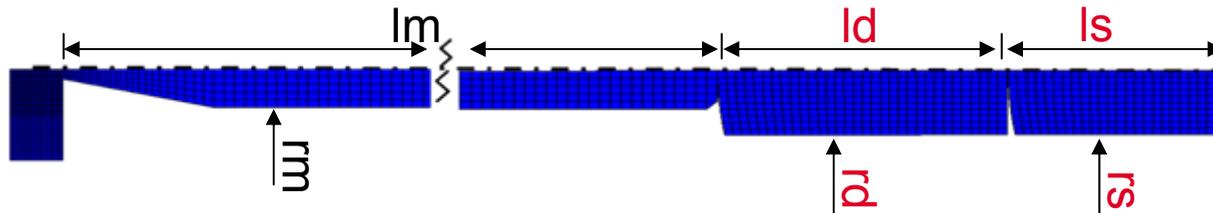


2.3 Optimierung – Vergleich der Ergebnisse

Optimierung	NLPQL	ARSM	EA
Anwendungsbereich	- mäßige Nichtlinearität	- mäßige Nichtlinearität (quadr. Approx.)	- breite Anwendbarkeit
Abweichung von Referenz-Design / Aussagegenauigkeit	26,6% - ϵ -optimale Lösung - konvergiert gegen lokale Optima	0,0% - ϵ -optimale Lösung - konvergiert bei geeigneter Approx. gegen globales Optimum	0,5% - Keine Optimalitätsgarantie - Güte hängt von investiertem Aufwand ab
CPU-Rechenzeit	85 Designs ca. 18 Stunden	70 Designs ca. 8 Stunden	117 Designs ca. 15 Stunden

3.1 Zuverlässigkeitsanalyse – Fertigungstoleranzen und Problemstellung

Fertigungstoleranzen:



Maße in mm

Bereich 1:	Parameter	ls	rs	ld	rd
	Mittelwert	43.99	15	90	15
	Abweichung	-0.1	-0.2	-0.15	-0.15
Bereich 2:	Parameter	ls	rs	ld	rd
	Mittelwert	90	15	60.2	15
	Abweichung	-0.15	-0.2	-0.1	-0.2

Nebenbedingungen des sicheren Bereiches:

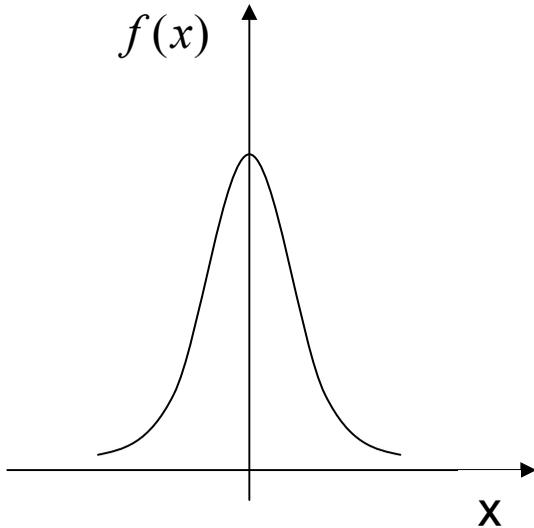
$$1,05 \cdot RZsc_{Opti} - RZsc \geq 0$$

$$U2 \max - 0,95 \cdot U2 \max_{Opti} \geq 0$$

$$RZsc - 0,95 \cdot RZsc_{Opti} \geq 0$$

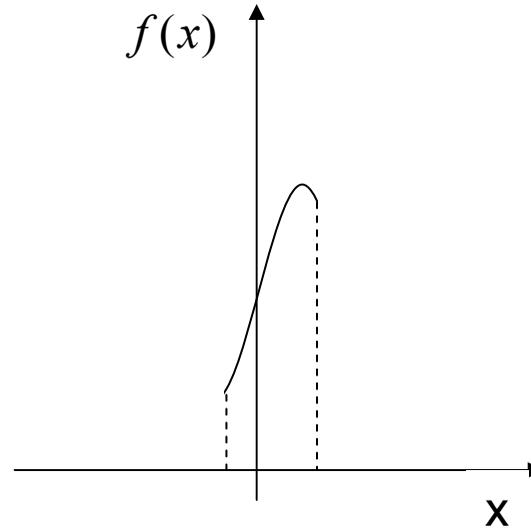
3.2 Zuverlässigkeitsanalyse - Grundlagen

Dichtefunktionen:



Normalverteilung

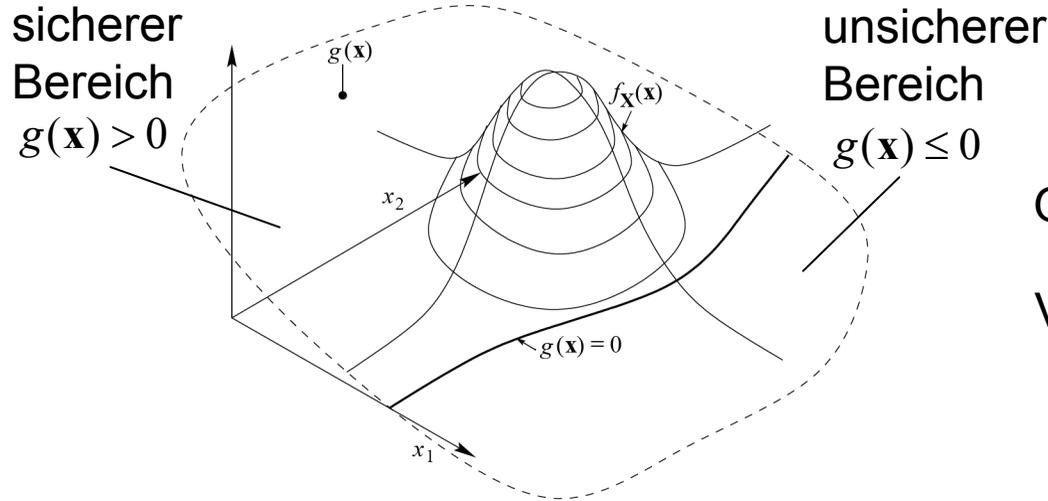
$$-\infty < x < \infty$$



Gestutzte Normalverteilung

$$x_{\min} \leq x \leq x_{\max}$$

3.2 Zuverlässigkeitsanalyse - Grundlagen



Grenzzustandsfunktion: $g(\mathbf{x})$

Versagenswahrscheinlichkeit:

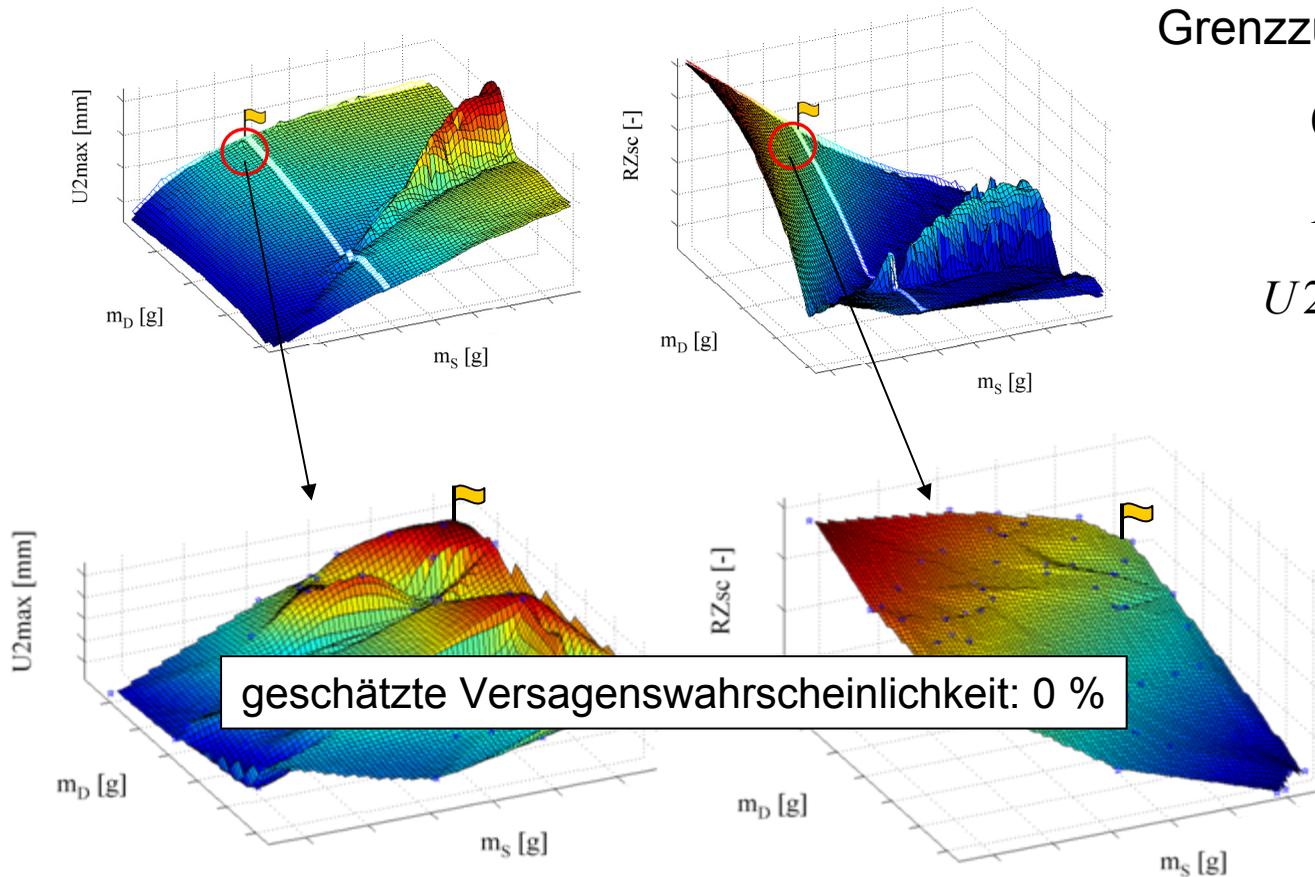
$$P_f = \int_{g(\mathbf{x}) \leq 0} \int_{\Lambda} f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}$$

Stochastische Methoden zur Ermittlung der Versagenswahrscheinlichkeit:

- Samplingmethoden
 - Monte Carlo Simulation
 - Latin Hypercube Sampling
 - Importance Sampling
 - Directional Sampling
 - ...
- Approximationsverfahren
 - First Order Reliability Method
 - Adaptive Response Surface Method
 - ...

3.3 Zuverlässigkeitsanalyse - Anwendung

Bereich 1:



Grenzzustandsfunktionen:

$$0,36 - RZ_{sc} \geq 0$$

$$RZ_{sc} - 0,33 \geq 0$$

$$U_{2max} - 0,50 \geq 0$$

3.3 Zuverlässigkeitsanalyse - Ergebnisse

Bereich 1:

Methode	Versagenswahrscheinlichkeit	Standardabweichung	Anzahl der Systemanalysen	Laufzeit
Latin Hypercube	0	0	200	17 h
Adaptive Sampling (Importance Samp.)	0	0	100	8,5 h
ARSM	ERROR	-	86	7 h
Directional Sampl.	$7,85 \cdot 10^{-9}$	$1,34 \cdot 10^{-9}$	30	10,5 h
FORM	0,5	764107	60	8 h

3.3 Zuverlässigkeitsanalyse - Anwendung

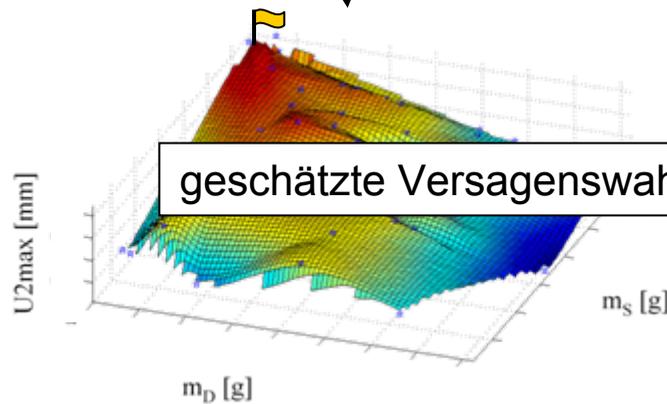
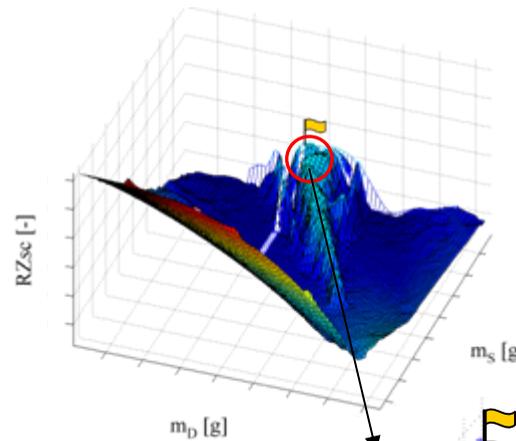
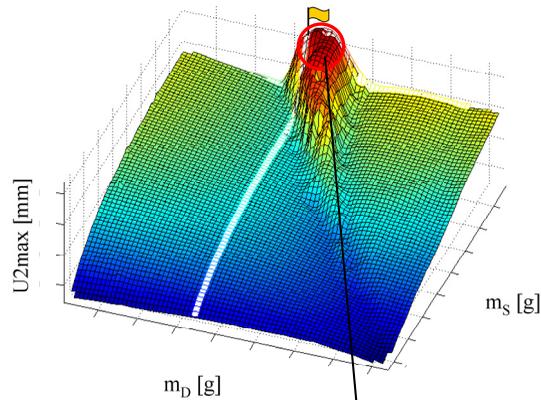
Bereich 2:

Grenzzustandsfunktionen:

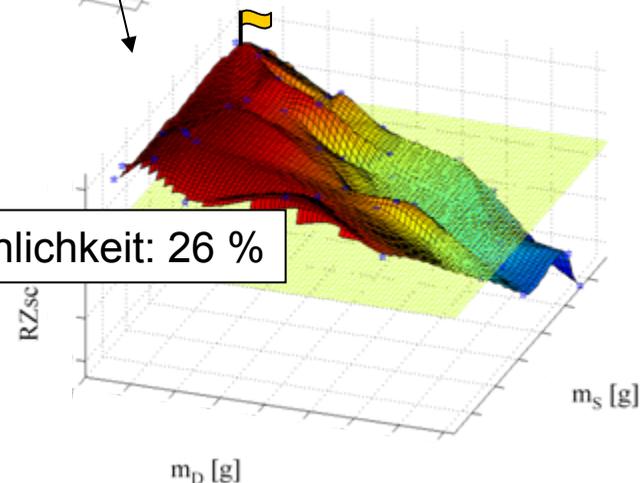
$$0,25 - RZsc \geq 0$$

$$RZsc - 0,22 \geq 0$$

$$U2max - 0,72 \geq 0$$



geschätzte Versagenswahrscheinlichkeit: 26 %



3.3 Zuverlässigkeitsanalyse - Ergebnisse

Bereich 2:

Methode	Versagenswahrscheinlichkeit	Standardabweichung	Anzahl der Systemanalysen	Laufzeit
Latin Hypercube	0,2663	0,0313	200	17 h
Latin Hypercube	0,2517	0,0177	600	55 h
Adaptive Sampling (Importance Samp.)	0,2236	0,0716	210	18 h
ARSM	0,1578	0,0681	258	22 h
Directional Sampl.	ERROR	-	186	75 h
FORM	0,5	3434	87	17 h
ISPUD	0,1834	0,0500	147	23 h

4. Zusammenfassung

Optimierung

- ARSM: - für mäßig nichtlineare Probleme wie in Bereich 1
- quadratische Approximation
- aktivierte Zusatzoption: Wiederverwendung von Stützstellen
- EA: - breite Anwendbarkeit, für Bereich 2 geeignet
- Kombination von Rekombination und Mutation

Zuverlässigkeit

- Adaptive Sampling : - für kleine erwartete Versagenswahrscheinlichkeiten wie in Bereich 1
- bei weniger als 10 Zufallsvariablen
- Latin Hypercube : - nur für hohe erwartete Versagenswahrscheinlichkeiten $P_f > 10^{-2}$ wie in Bereich 2
- unabhängig von der Anzahl an Zufallsvariablen



5. Ausblick

- Weitere Untersuchungen im Bereich 2
- Einfluss unterschiedlichen Werkzeugeinsatzes im Bereich 2
- Zuverlässigkeitsanalyse: ARSM der nächsten optiSLang-Version testen

Fazit: Optimierungs- und Zuverlässigkeitsalgorithmen sind

- für die virtuelle Produktauslegung zwingend notwendig,
- deren Kenntnis für den Anwendungsbereich des Problems erforderlich
- und dessen Ergebnisse anschließend auf Plausibilität zu überprüfen.

