Software: FEM - Tutorial - Magnetfeld

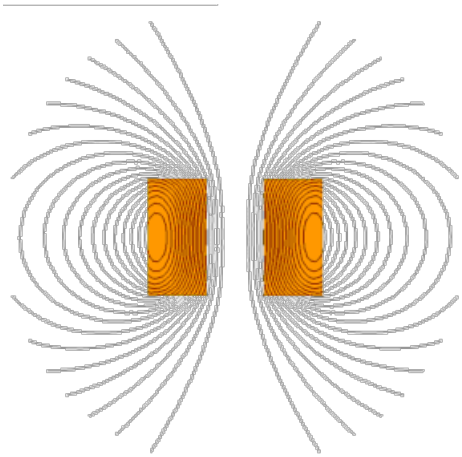
Aus OptiYummy

↑

← →

5. Komplex im FEM-Tutorial
Magnetfeld (statisch)
- Wirbelfeld-Modell mit FEMM -
Autor: Dr.-Ing. Alfred Kamusella

Ein Computer kann alles was du willst,
solange du nur willst, was er kann!
- Autor unbekannt -



1. Grundlagen

- Physik des Magnetfeldes
- Potentialfeld-Analogie
- Wirbelfeld-Ansatz
- Einführungsbeispiel

2. Modell eines Elektro-Magneten

- Problemdefinition
- Material und Stromkreis
- Geometrie
- Randbedingungen
- Vernetzung und Berechnung

3. Kraft- und Koppelfluss-Kennfelder

- Kraft und Koppelfluss (Einführung)
- Kraft und Koppelfluss (Lua-Script)
- Kennfeld-Berechnung (OptiY):
  - Einbindung in OptiY-Workflow
  - Datenschnittstellen (In-/Outputfiles)
  - Rastersuche
  - Parallelisierung

4. Einfluss von Toleranzen auf die Haltekraft

- Wirkung kleiner Luftspalte
- Probabilistische Simulation (OptiY):
  - Experiment-Konfiguration
  - Numerische Stabilität des Modells
  - Verfahren mit Zufallszahlen (Monte Carlo)
  - Momenten-Methode

5. Kraft- und Koppelfluss-Kennfelder (Response Surface)

- Identifikation von Ersatzmodellen

- Adaptiver Gauss-Prozess
- Kennfeld-Export als C-Code (***Kein Bestandteil der Lehrveranstaltung FEM!***)
  - Kennfeld-Testmodell (***Kein Bestandteil der Lehrveranstaltung FEM!***)
  - Differentierbarkeit von Antwortflächen (***Kein Bestandteil der Lehrveranstaltung FEM!***)

6. System-Simulation mit Kennfeldern (***Kein Bestandteil der Lehrveranstaltung FEM!***)

- Elektromagneto-mechanischer Wandler (C-Code)
- Kennfeld-Export als Modelica-Code
- Kennfeld als Modelica-Implantat
- Elektromagneto-mechanischer Wandler (Modelica-Implantat)

Einzusendende Ergebnisse:

- Teilnehmer der Lehrveranstaltung **"Finite Elemente Methode"** laden ihre Ergebnisse bei Opal hoch.
- Für den Upload sind mit (**xx**=Teilnehmer-Nummer 01...99) alle erforderlichen vollständig konfigurierten Dateien **Magnet_xx.***, **Eingabe_xx.txt** und **Ergebnis_xx.txt** sowie die Antwort-PDF in **einem** Archiv-File **FEM5_xx** (.zip) zu senden.
- Die Experimente der OptiY-Projektdatei müssen für die durchgeführten Analyse alle konfigurierten Ergebnis-Fenster enthalten.
- In der bereitgestellten PDF zu beantwortende Fragen zur Haltekraft des Magneten bei einem Strom **i=0.2xx A**:
  1. Wie groß ist der Mittelwert der Haltekraft?
  2. Wie groß ist die Streuung der Haltekraft infolge der Luftspalt-Toleranzen?
  3. Welche Luftspalt-Toleranz besitzt den größten Einfluss auf die Streuung der Haltekraft? Diese Antwort ist zu begründen!

Einsendeschluss ist die Nacht vor dem nächsten Übungskomplex. Die Nacht endet um 10:00 Uhr.

← →

Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld&oldid=26081“

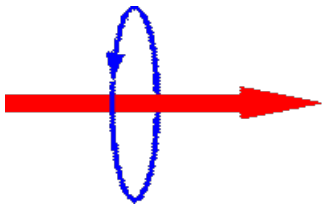
Software: FEM - Tutorial - Magnetfeld - Physik

Aus OptiYummy

↑

← →

Physik des Magnetfeldes



Ein Stromdurchflossener Leiter ist immer mit einem Magnetfeld verknüpft. Die grundlegenden physikalischen Zusammenhänge werden im Folgenden kurz dargestellt.

Wirbelfeld

Die Feldlinien des magnetischen Feldes bilden geschlossene Linien um den stromdurchflossenen Leiter.

Magnetischer Kreis

Das ist der gesamte vom Magnetfeld erfüllte Raum um den stromdurchflossenen Leiter.

Magnetischer Fluss $[\Phi]=V\cdot s$ (=Wb)

Das ist die gesamte in sich geschlossene Erscheinung, welche den Strom "umwirbelt" - die Richtung des Flusses ist mit der positiven Richtung des Stromes über eine "Rechtsschraube" verbunden.
In Wirklichkeit "strömt" bzw. "wirbelt" nichts, sondern es handelt sich nur um die Charakterisierung eines Raumzustandes!

Magn. Flussdichte (Induktion) $[B=d\Phi/dA\perp] = V\cdot s/m^2$ (=T)

Man spricht von "Quellenfreiheit", weil keine "magnetischen Ladungen" existieren und die Fluss(dichte)-Linien immer in sich geschlossen sind:

- $\boldsymbol{\Sigma\Phi_{in}=\Sigma\Phi_{out}}$ - der gesamte in ein Volumen-Element "hineinströmende" Fluss ist gleich der Menge des "herausströmenden" Flusses.
- $\int_{\text{Hülle}} \mathbf{B\cdot dA=0}$ - das Hüllenintegral der Induktion über die Oberfläche eines Volumen-Elements ist Null.

Magnetomotorische Kraft (Durchflutung) $[\Theta] = A$

Die MMK ist gleich der Summe der umschlossenen Ströme, wenn man einen beliebigen geschlossenen Weg **s** im Magnetkreis betrachtet:

- $\boldsymbol{\Theta = \Sigma i_x}$ - die Ströme werden als "antreibende Kraft" für den Fluss entlang des betrachteten Weges im magnetischen Kreis interpretiert.
- $\int_{\text{Umlauf}} \mathbf{B\cdot ds=\mu_0\cdot\Sigma i_x}$ - das Umlaufintegral der Flussdichte entlang dieses Weges ist in Vakuum immer proportional zur Summe der umschlossenen Ströme (zur MMK).

Induktionskonstante $\boldsymbol{\mu_o} = 4\cdot\pi\cdot10^{-7}\ V\cdot s/(A\cdot m)$

- $\int_{\text{Umlauf}} \mathbf{B\cdot ds=\mu_0\cdot\Theta}$

μ_0 ist die Naturkonstante, welche das Umlaufintegral der Flussdichte längs eines Weges mit dem Wert der umschlossenen MMK verknüpft.

- $c^{-2} = \mu_0 \cdot \varepsilon_0$

  Die enge Verknüpfung zwischen magnetischem und elektrischem Feld wird deutlich über den Zusammenhang zwischen Vakuum-Lichtgeschwindigkeit, Induktions- und Influenzkonstante.

Permeabilität

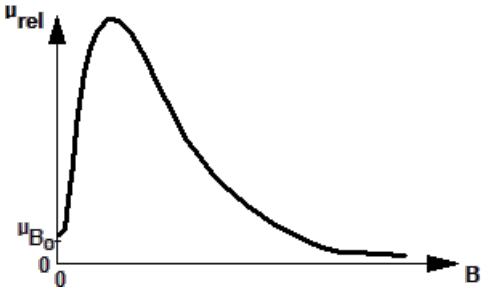
Legt man den Weg für das Umlaufintegral nicht durch Vakuum, sondern durch Ferromagnetika (Eisen, Nickel, Kobalt, Kupfermangan-Legierungen), so ist der Wert des Umlaufintegrals von **B** wesentlich größer als $\mu_0 \cdot \Sigma i_x$:

- d.h. bei gleichem el. Strom sind der magn. Fluss bzw. die Flussdichte wesentlich größer als in Vakuum (oder in nicht ferromagnetischen Stoffen).
- **relative Permeabilität**: dimensionsloser Faktor μ_r, welcher die magnetische Eigenschaft des jeweiligen Stoffes im Vergleich zum Vakuum charakterisiert.
- **absolute Permeabilität**: $\mu = \mu_r \cdot \mu_0$ mit der Einheit $V \cdot s/(A \cdot m)$

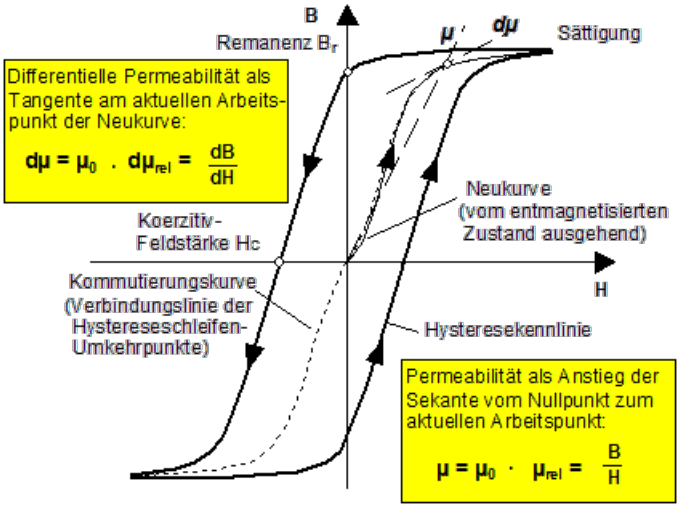
Magn. Feldstärke $[H=B/\mu] = A/m$

- $\int_{Umlauf} \mathbf{H \cdot ds = \Sigma i_x = \Theta}$ - Die Einführung der magnetischen Feldstärke hat den Vorteil, dass mit dieser Größe das Umlaufintegral entlang eines Weges unabhängig von den Stoffeigenschaften immer gleich dem Wert der umschlossenen Ströme ist (MMK=Durchflutung).
- Dieses "Durchflutungsgesetz" ist ein Grundgesetz des Magnetfeldes. Indem der elektrische Strom verallgemeinert als Summe von Konvektions- und Verschiebungsstrom betrachtet wird, entspricht es der 1. Maxwellschen Gleichung in der Integralform.

Magnetisierungskennlinie B=f(H)



- Beim Auf- bzw. Abmagnetisieren von Eisen entstehen Schleifen der Magnetisierungskurve, deren Breite von der Aussteuerung in den Umkehrpunkten abhängt. Bei großen H-Werten tritt keine zusätzliche, durch das Material bedingte B-Erhöhung mehr ein (Sättigung).
- Für die Modellierung ist die Nachbildung der Magnetisierungshysterese meist zu aufwändig. Für viele Anwendungsfälle genügt die Nachbildung der Neukurve, welche praktisch mit dem Verlauf der Kommutierungskurve übereinstimmt. Es besteht dann ein eindeutiger Zusammenhang zwischen der Flussdichte B und der Feldstärke H, der über eine Funktion $\mu_{rel}=f(B)$ dargestellt werden kann:



Magn. Spannung [V] = A

- $\int_{Umlauf} \mathbf{H \cdot ds = \Theta = \Sigma V_x}$ - unterteilt man den Umlaufweg in einem magnetischen Kreis in einzelne Abschnitte, so ist die Summe aller magn. Spannungen der einzelnen Abschnitte gleich der MMK dieses Umlaufweges.
- $\mathbf{V_{AB} = \int_{A-B} H \cdot ds}$ - die magn. Spannung zwischen zwei Punkten des magn. Kreises ergibt sich aus dem Weg-Integral der Feldstärke zwischen diesen beiden Punkten

Magn. Widerstand [$Rm_{AB}=V_{AB}/\Phi$] = A/(V·s)

- In Analogie zum elektrischen Widerstand kann man einen magnetischen Widerstand zwischen zwei Punkten eines magnetischen Kreises definieren.
- Ein Abschnitt des Raumes wird somit auf ein idealisiertes Element mit konzentrierten Parametern reduziert.
- Mit Hilfe der Netzwerkanalogien kann man damit das Verhalten des magnetischen Kreises simulieren.
- **Rm=L/(μ·A)** - im homogenen Feld gilt für prismatische Raumabschnitte der Länge **L** und dem Querschnitt **A** (z.B. Zylinder, Quader) die Dimensionierungsgleichung.
- Für komplexere Teil-Geometrien eines magnetischen Kreises existieren teilweise zugeschnittene Dimensionierungsgleichungen (z.B. für Kugel- und Zylinderschalen).
- Der "exakte" magnetische Widerstand von komplexen Magnetkreis-Abschnitten lässt sich oft nur mit der Methode der Finiten Elemente ermitteln.
- **Gm=1/Rm** - der magnetische Leitwert besitzt die Einheit H=V·s/A.
- **$L=w^2 \cdot Gm$** - Der gesamte magnetische Leitwert eines magnetischen Kreises ist proportional der elektrischen Induktivität **L** der Spule mit der Windungszahl **w**.

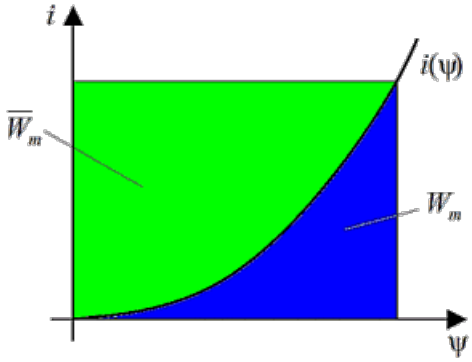
Kraftwirkung des Elektromagneten

In Abhängigkeit von möglichen Idealisierungen des magnetischen Feldes kann man unterschiedliche Verfahren nutzen, um die Kraftwirkung auf Grenzflächen innerhalb des magnetischen Kreises hinreichend genau zu berechnen:

a) Energieänderung (Induktivität)

- **$F=0{,}5 \cdot i^2 \cdot (dL/ds)$** - betrachtet man die Energieänderung in der elektrischen (Ersatz-)Induktivität bei Änderung **ds** des Arbeitsluftspalts **s**, so erhält man eine Formel, nach der die Kraft proportional der Ableitung der Induktivität nach dem Luftspalt ist.
- Der Proportionalitätsfaktor ist das Quadrat des elektrischen Stroms **i**.
- Ist **dL/ds** für einen vorgegeben Luftspalt bekannt, so kann man für "beliebige" Ströme die

Kraft berechnen (unter der Annahme, dass **dL/di=0**, d.h. die Induktivität ist "hinreichend" unabhängig vom Spulenstrom.



Maxwellsche Zugkraftformel

- Unter der Annahme, dass an der Grenzfläche **A** sehr große **μ**-Unterschiede existieren und ein homogenes Feld der Flussdichte **B** hindurchgeht, gilt die Maxwellsche Zugkraft-Formel $F=A\cdot B^2/(2\cdot\mu_0)$

Maxwellsches Stress Tensor Volumenintegral

- Für den allgemeinen Fall des inhomogenen Feldes durch einen Körper mit anderem **μ** als seine Umgebung, kann man die resultierende Kraft durch Aufintegration aller differentiell kleinen Kraftanteile über seine Oberfläche gewinnen.
- Grundlage der Kraftberechnung ist die differentielle Änderung der magnetischen Coenergie [math]\displaystyle{ \overline{W} }[/math]$_m$ bei differentieller Verrückung der Oberfläche (***Ψ***=*Verkettungsfluss*).

← →

Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld_-_Physik&oldid=22469“

Software: FEM - Tutorial - Magnetfeld - Potentialfeld-Analogie

Aus OptiYummy

↑

← →

Potentialfeld-Analogie

In den bisherigen Übungen haben wir zur FEM-Simulation elektrischer Felder in *Autodesk Fusion* die Analogien zum Temperatur-Feld benutzt. Nur so konnte man mit den Mitteln dieses FEM-System die elektrische Domäne behandeln. Zur Erinnerung werden diese Analogien hier noch einmal dargestellt:

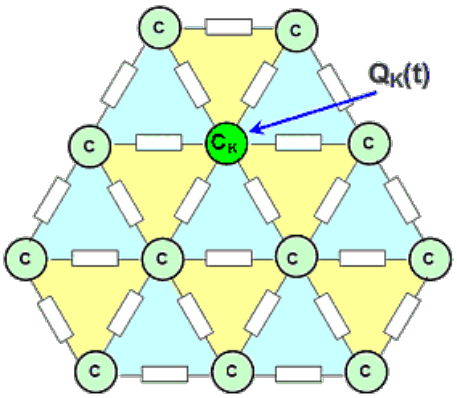
	Temperatur-Feld	**Elektr. (Fluss-)Feld**	**Elektrostatisches Feld**
Potential	Temperatur	elektrische Spannung	el.statisches Potential
.	[K]	[V]	[V]
Pot.Gradient	Temperatur-Gradient	el. Spannungsabfall	Feldstärke
.	[K/m]	[V/m]	[V/m]
Material-eigenschaft	Wärmeleitfähigkeit	spez. el. Leitfähigk.	Dielektrizitätskonst.
	[W/(K·m)]	[1/(Ohm·m)]	[(A·s)/(V·m)]
Flussgröße	Wärmestrom	Elektrischer Strom	Verschiebung (Ladung)
.	[W]	[A]	[A·s]
Flussdichte	Wärmestromdichte	Stromdichte	Ladungsdichte
	"Flächenlast" [W/m²]	[A/m²]	[(A·s)/m²]

Auch für die magnetische Domäne kann man solche Analogien definieren, wobei sich je nach Zielstellung unterschiedliche Varianten von Analogiebeziehungen bewährt haben. Problematisch bei der Analogie zum Temperaturfeld ist der Wirbelfeld-Charakter des Magnetfeldes (Quellenfreiheit)!

Für Wärmefelder gilt: $\{C\} \cdot \{\dot{T}\} + \{K\} \cdot \{T\} = \{Q(t)\}$

```
{C}     Wärmekapazitätsmatrix
{T'}    Vektor der Temperaturänderungsgeschwindigkeit
{K}     Leitfähigkeitsmatrix
{T}     Temperaturvektor
{Q(t)}  Wärmestromvektor
```

Die Wärmegleichung beschreibt die Leistungsbilanz für jeden Knoten:



Q(t) Energiefluss zur Netzumgebung (Last)
C·T' Energiefluss in Wärmespeicher des Knoten
K·T Energiefluss über die Wärmeübergangswiderstände zu den Nachbarknoten

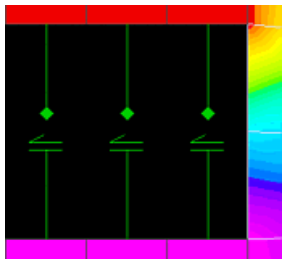
Analogien zum Magnetfeld lassen sich nur für den statischen Fall aufstellen, d.h., wenn Speicherelemente nicht berücksichtigt werden (**{C}={0}**). Dann gelten die bekannten Analogien:

```
{K}     Permeabiltätsmatrix {µ}
{T}     Vektor der magnetischen Potentiale {Vm}
{Q(t)}  Vektor der magnetischen Flüsse {Φ}
```

Diese Analogien zwischen dem Temperatur- und dem Magnetfeld nun vollständig in tabellarischer Form:

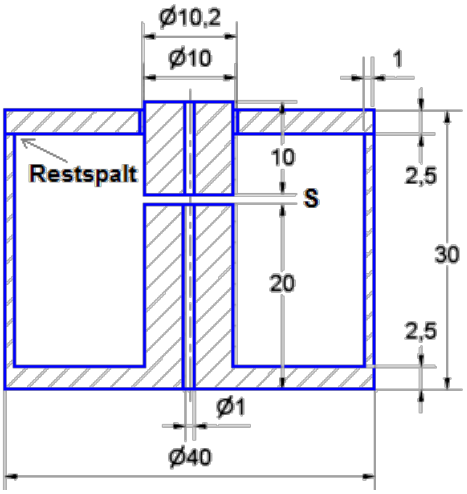
	Temperatur-Feld	**Magnetfeld**
Potential	Temperatur	magnetische Spannung
.	[K]	[A]
Pot.Gradient	Temperatur-Gradient	magnetische Feldstärke
.	[K/m]	[A/m]
Material-eigenschaft	Wärmeleitfähigkeit	Permeabilität
	[J/(K·m)]	[V·s/(A·m)]
Flussgröße	Wärmestrom	magnetischer Fluss
.	[W]	[V·s=Wb]
Flussdichte	Wärmestromdichte	magn. Flussdichte (Induktion)
"Flächenlast"	[W/m²]	[(V·s)/m²=T]

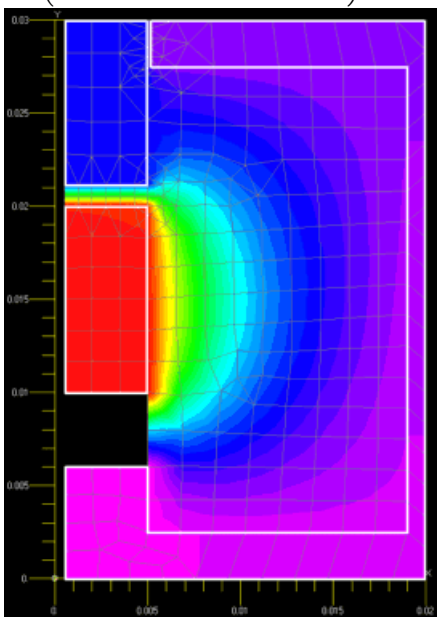
Damit diese Analogien funktionieren, muss man im räumlichen Bereich der MMK (=Spule) einen magnetischen "Isolator" (magn. Leitwert=0 H) vorsehen:



- Es genügt ein hinreichend großes "Loch" an der Spulen-Position im Netz.
- Wenn man an einer Seite des MMK-Loches eine Flussdichte in den magnetischen Kreis speist und die andere Seite mittels "Konvektion" auf Nullpotential zieht, stellt sich eine magn. Spannung über der MMK ein.

- Aus dem eingespeisten magnetischen Fluss und dem berechneten MMK-Wert kann man den magnetischen Widerstand des gesamten Magnetkreises (aus "Sicht" der MMK) berechnen.





Die Anwendung dieser Potentialfeld-Analogie für magnetische Kreise weist Schwächen auf:

- Magnetische Durchflutungen (Spulen) können nur eingeschränkt berücksichtigt werden, da es nicht möglich ist, den elektrischen Strom als Ursache des Magnetfeldes im Modell direkt zu berücksichtigen.
- Komplette Magnetkreise sind nur eingeschränkt modellierbar, da das Wirbelfeld "aufgeschnitten" und als Potentialfeld zwischen 2 magnetischen Polen betrachtet werden muss.
- Deshalb werden wir im Folgenden nur das FEMM-Programm benutzen, welches den Wirbelfeld-Charakter des Magnetfeldes behandeln kann!

← →

Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld_-_Potentialfeld-Analogie&oldid=24142“

Software: FEM - Tutorial - Magnetfeld - Wirbelfeld-Ansatz

Aus OptiYummy

↑

← →

Wirbelfeld-Ansatz

Am Beispiel einer 2D-Magnetfeld-Berechnung soll verdeutlicht werden, was sich hinter der Begriffs- und Formelwelt verbirgt, mit welcher man bei der FEM-Simulation konfrontiert wird.

Gewünschte Simulationsergebnisse

In einem abgegrenzten Gebiet der x,y-Ebene sollen die magnetische Induktion $\overline{\mathbf{B}}$**(x,y)** und die elektrische Feldstärke $\overline{\mathbf{E}}$**(x,y)** als grundlegende Größen berechnet werden:

- Die magnetische Induktion $\overline{\mathbf{B}}$**(x,y)** ist ein Vektor in der x,y-Ebene
- Die elektrische Feldstärke $\overline{\mathbf{E}}$**(x,y)** ist ein Vektor in z-Richtung (Senkrecht zur Induktion!)
- Alle anderen elektrischen und magnetischen Größen lassen sich über Abhängigkeiten aus $\overline{\mathbf{B}}$ und $\overline{\mathbf{E}}$ berechnen.

Maxwellsche Gleichungen

Der Wirbelfeld-Ansatz berücksichtigt die Verkopplung zwischen dem elektrischen und dem magnetischen Feld über die ersten beiden **Maxwellschen Gleichungen** sowie die Quelleneigenschaften von magn. und elektrischem Feld:

1. **Durchflutungsgesetz** (rot $\overline{\mathbf{H}} = \overline{\mathbf{J}} + \delta\overline{\mathbf{D}} / \delta t$)
   Das Umlaufintegral der magn. Feldstärke (MMK) ist gleich der Summe der von diesem Umlauf umfassten Ströme, dies verdeutlicht die Integralform (Gesetz von Stokes).
2. **Induktionsgesetz** (rot $\overline{\mathbf{E}} = -\delta\overline{\mathbf{B}} / \delta t$)
   Das Umlaufintegral der elektrischen Feldstärke ist gleich der von diesem Umlauf umfassten Flussänderungsgeschwindigkeit (EMK)
3. **Quellenfreiheit des Magnetfeldes** (div $\overline{\mathbf{B}}=0$)
   Der in eine Hülle "hineinströmende" Fluss ist gleich dem aus dieser Hülle "herausströmenden" Fluss.
4. **Elektrische Ladungsquelle** (div $\overline{\mathbf{D}}=\varrho$)
   Die Differenz aus der in eine Hülle hineinströmenden Ladung und der herausströmenden Ladung entspricht der Ladungsänderung des umhüllten Volumens.

Magnetisches Vektorpotential

Das Berechnungsproblem lässt sich numerisch vereinfachen, wenn man es auf die Berechnung des so genannten **magnetischen Vektorpotentials** reduziert:

- Für beliebige Wirbelfelder (die immer quellenfrei sind!) lässt sich ein "allgemeines" Vektorpotential $\overline{\mathbf{A}}$ definieren (div rot $\overline{\mathbf{A}}=0$).
- Für magnetische Felder gilt wegen der Quellenfreiheit div $\overline{\mathbf{B}}=0$.
- Daraus schlussfolgerte Maxwell $\overline{\mathbf{B}}=\text{rot}\,\overline{\mathbf{A}}$, womit der Zusammenhang zwischen "konkreter" Induktion und "abstraktem" Vektorpotential hergestellt ist.

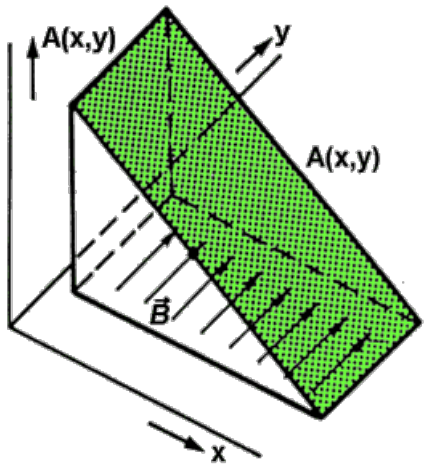
- Da $\overline{\mathbf{A}}$ als zu berechnende Unbekannte in unserem 2D-Beispiel nur eine Komponente in z-Richtung besitzt, reduziert sich der Rechenaufwand für das Gleichungssystem gewaltig.
- $\overline{\mathbf{A}}$(x,y) kann man sich als Lösungsfläche über der betrachteten x,y-Ebene vorstellen, wenn man die Länge **z** des Vektors $\overline{\mathbf{A}}$(x,y) als Höhenwert über dem Punkt (x,y) aufträgt.
- Die Neigung dieser Fläche entspricht in jedem Punkt (x,y) der dort vorhandenen Induktion $\overline{\mathbf{B}}$(x,y):

$$\mathbf{B_x} = \delta\overline{\mathbf{A}} \,/\, \delta y \text{ und } \mathbf{B_y} = -\delta\overline{\mathbf{A}} \,/\, \delta x$$

- Das ergibt sich aus der vollständigen Koordinaten-Darstellung des Differential-Operators **rot**. Weil $\overline{\mathbf{A}}$(x,y,z) nur ein Vektor in z-Richtung ist, sind die partiellen Ableitungen von A_x und A_y in alle 3 Koordinatenrichtungen gleich Null:

$$\overline{\mathbf{B}}(x,y,z) = \text{rot } \overline{\mathbf{A}}(x,y,z)$$

$$=\left(\frac{\partial A_z}{\partial y} - \frac{\partial A_y}{\partial z}\right)\vec{e}_x + \left(\frac{\partial A_x}{\partial z} - \frac{\partial A_z}{\partial x}\right)\vec{e}_y + \left(\frac{\partial A_y}{\partial x} - \frac{\partial A_x}{\partial y}\right)\vec{e}_z$$



- Das Bild soll den Zusammenhang zwischen dem Vektorpotential $\overline{\mathbf{A}}$ und der Induktion $\overline{\mathbf{B}}$ am Beispiel eines homogenen Magnetfeldes verdeutlichen. Die grüne Fläche repräsentiert das berechnete Vektorpotential. Der negative Anstieg an der Fläche entspricht im Beispiel dem Wert der Induktion in y-Richtung.
- Stellt man das Vektorfeld durch Äquipotentiallinien dar, indem man Punkte gleichen Vektorpotentials (gleicher Höhe) verbindet, so entsteht das Feldlinienbild des Magnetfeldes:
  - Zwischen benachbarten Äquipotentiallinien existieren gleiche Potentialdifferenzen.
  - Starke Anstiege des Vektorfeldes entsprechen großen Induktionswerten und widerspiegeln sich in dicht beieinander liegenden Feldlinien.

← →

Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld_-_Wirbelfeld-

Ansatz&oldid=22468“

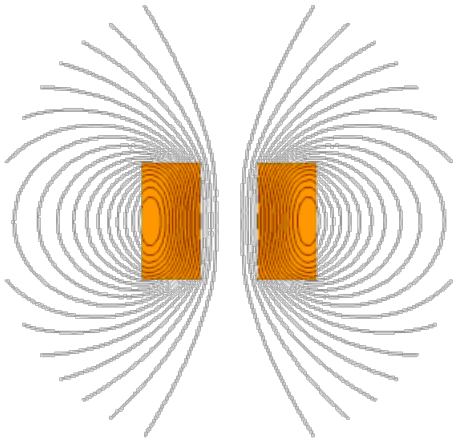
Software: FEM - Tutorial - Magnetfeld - Einfuehrungsbeispiel

Aus OptiYummy

↑

← →

Einführungsbeispiel

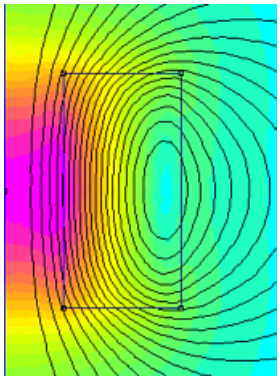


Die magnetische Domäne wird als "Magnetics Problem" im Programmsystem FEMM behandelt. Der zugehörige Programm-Modul ermöglicht die Berechnung von 2D- und axialsymmetrischen Problemen für niederfrequente magnetische Felder basierend auf dem Wirbelfeld-Ansatz. "Niederfrequent" bedeutet, dass die Stromverdrängung im elektrischen Leiter infolge hochfrequenter magnetischer Felder nicht im Gleichungssystem berücksichtigt wird.

In Hinblick auf die Einarbeitung in die Magnetfeld-Simulation sollte man unbedingt zuerst das Beispiel "Luftspule" bearbeiten. Die Arbeit mit der grafischen Oberfläche wird Schritt für Schritt in der Datei **tutorial-magnetic.pdf** erläutert, welche man zusätzlich zum **manual.pdf** auch im FEMM-Programmordner findet. Folgt man den Anweisungen im Tutorial, so kann man kaum etwas falsch machen.

An dieser Stelle erfolgen deshalb nur einige Hinweise in Bezug auf die Interpretation der Simulationsergebnisse.

Vektorpotential (Contour Plot)



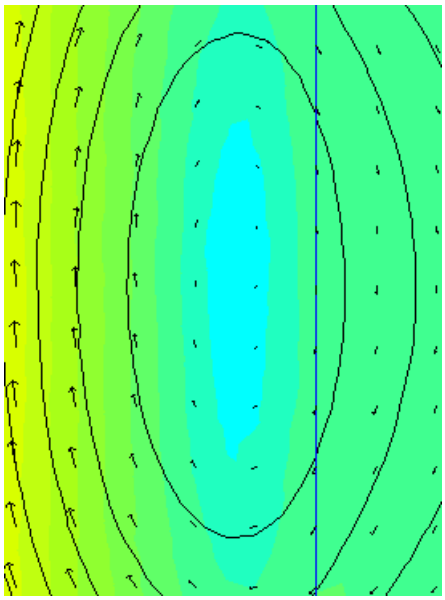
- Das nach der Berechnung im Postprocessing standardmäßig generierte Feldlinienbild repräsentiert Iso-Potentiallinien des Vektorpotentials
- Die Anzahl der dargestellten Linien sowie den Darstellbereich kann man über den Dialog des Contour-Plot-Buttons ändern.
- Die Symmetrieachse von axialsymmetrischen Problemen besitzt immer das Vektorpotential 0 V·s/m.
- Das höchste Vektorpotential existiert in der Nähe des äußeren Spulenrandes ("Bergkuppe").
- ***Beachte:***
  Nur für planare 2D-Probleme handelt es sich bei den dargestellten Linien um Äquipotentiallinien des Vektorpotentialfeldes $\overline{\mathbf{A}}$ (d.h. gleiche Potentialdifferenzen zwischen benachbarten Linien). Bei axialsymmetrischen 2D-Problemen wird stattdessen die Funktion $\mathbf{2 \cdot \pi \cdot r \cdot \overline{A}}$ mit Äquipotentiallinien

abgebildet!

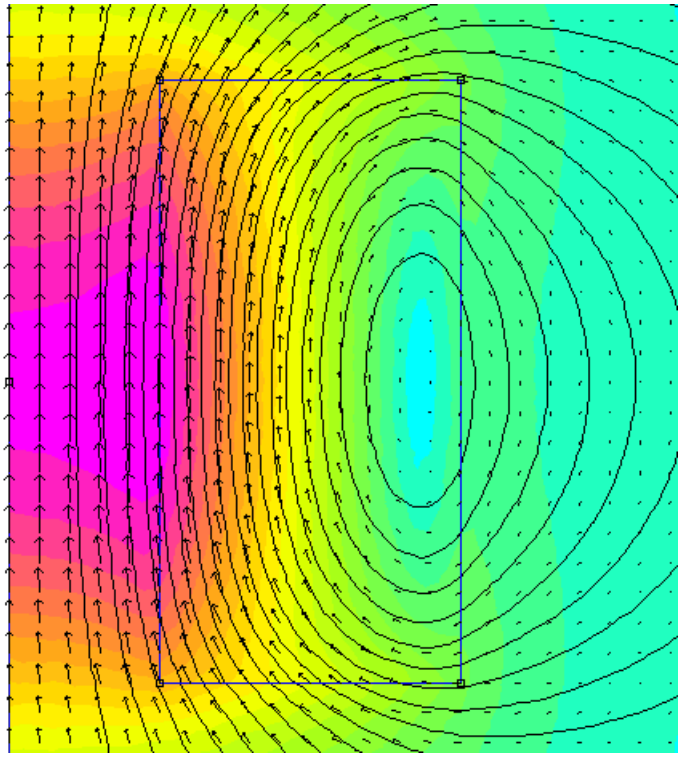
Flussdichte (Density Plot)

- Der Dialog des Density Plot bietet die Möglichkeit, die magn. Flussdichte im Sinne einer farbigen Contour-Darstellung abzubilden (cyan=kleine Werte ... magenta=große Werte).
- Im oberen Bild erkennt man deutlich, dass es sich bei den dargestellten Feldlinien nicht um Äquipotentiallinien von $\overline{\mathbf{A}}$ handelt, denn trotz größerer Linienabstände ist die Induktion **B** im Spulenzentrum (magenta) größer als in den unteren Wicklungslagen (rot bzw. gelb).

Feldstärke/Induktion (Vector Plot)



- In Form von skalierten Vektoren kann man entweder die magnetische Induktion oder die Feldstärke darstellen lassen.
- Wählt man die Induktion **B** (Bildausschnitt der rechten Spulenseite), so erkennt man, dass der Betrag von $\overline{\mathbf{B}}$ vom Anstieg des Vektorpotentialfeldes abhängt. Die Richtung von $\overline{\mathbf{B}}$ steht in jedem Punkt senkrecht auf dem Gradientenvektor des steilsten Abstiegs.
- Wählt man die Feldstärke **H** (unteres Bild), so ist der Betrag der Feldstärke der Induktion proportional ($H=B/\mu_0$) und besitzt die gleiche Richtung. Die Wicklung der Spule wirkt magnetisch wie Luft.



← →

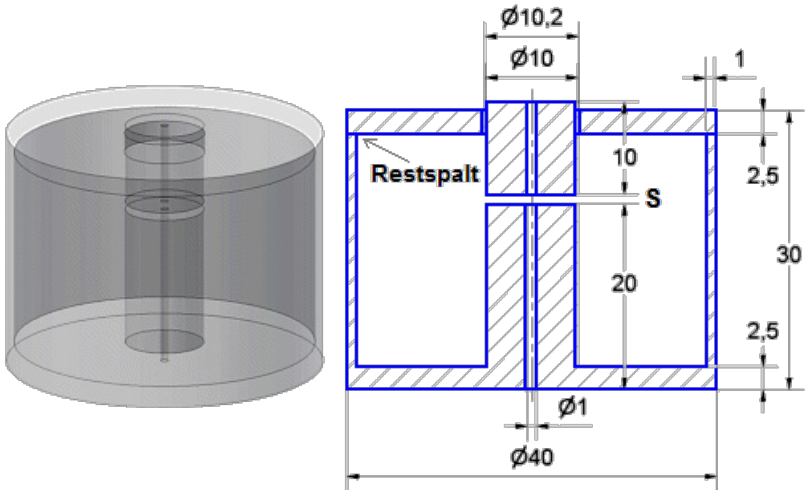
Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld_-_Einfuehrungsbeispiel&oldid=18050“

Software: FEM - Tutorial - Magnetfeld - Problemdefinition

Aus OptiYummy

↑

← →
Problemdefinition



Inhaltsverzeichnis

- 1 Bauform des Elektro-Magneten
- 2 Experiment-Ziele
- 3 Definition des Problem-Typs
- 4 Parameter

Bauform des Elektro-Magneten

Untersucht werden soll in diesem Übungskomplex der im Simulationspraktikum zur Lehrveranstaltung **"Optimierung"** intensiv behandelte Topfmagnet für den Antrieb einer Präge-Nadel:

- Axial im Eisenkreis befindet sich eine Bohrung für den Stift der Präge-Nadel.
- Der bewegliche ***Anker*** erfordert im ***Deckel*** des Eisentopfes einen Führungsspalt von 0,1 mm.
- Zwischen ***Deckel*** und ***Topf*** existiert infolge unvermeidlicher Toleranzen und Oberflächen-Rauhigkeiten ein luftgefüllter Restspalt ***sDeckel*** im µm-Bereich
- Die Kraft im Arbeitsluftspalt ***s*** soll für einen Wertebereich von ***sMin*≈0 mm** bis ***sMax*=4 mm** untersucht werden.
- Die Spule besitzt eine Windungszahl ***wSpule*=400**.
- Die im Bild nicht dargestellte Spule füllt den Wickelraum fast vollständig aus. Zwischen den

Drähten und dem Eisenkreis ist ein ***Spalt*=0,5 mm**.
- Die Gesamtdicke der Lackbeschichtung des Spulendrahtes wird zu ***dLack*=60 µm** angenommen.
- Es wird die nichtlineare Magnetisierungskurve des verwendeten Eisenwerkstoffes benutzt. Die Hysterese der BH-Kurve wird vernachlässigt.
- Die Wirbelströme im Eisenkreis werden nicht berücksichtigt

Experiment-Ziele

- Mittels der FEM-Simulation soll die Kraft **F** im Arbeitsluftspalt in Abhängigkeit von der Ankerposition **s** und dem aktuellen Strom **i** berechnet werden.
- Die Einflüsse von wesentlichen Fertigungstoleranzen auf die Haltekraft des Magneten sind zu untersuchen.

Wir entwickeln das erforderliche Modell sofort als LUA-Script "**Magnet_xx.LUA**" unter Nutzung z.B. des Crimson-Editors. Die Verwendung der LUA-Konsole von FEMM ist auf Grund des großen Script-Umfangs und unserer bereits vorhandenen Erfahrungen für die Script-Entwicklung nicht mehr empfehlenswert!

Definition des Problem-Typs

Wichtig:
Wir werden im Folgenden das standardmäßig aktivierte *Smart Meshing* abschalten. Durch die geplante Einbindung des FEMM-Modells in eine Analyse-/Optimierungsumgebung ergeben sich sehr viele Modellberechnungen. Dazu ist im Sinne der Berechnungszeit ein Kompromiss zwischen Vernetzungsdichte und Berechnungsgenauigkeit erforderlich. Diesen erreicht man nur durch eigene Intelligenz bei der Vernetzung!

Ein FEM-Script-Modell sollte grundsätzlich mit der Problem-Definition beginnen. Damit werden die Weichen für die eigentliche Modellbildung gestellt:

- Das ***Magnetic Problem*** konfigurieren wir als ***statisch*** und ***axialsymmetrisch*** mit der Längeneinheit ***Millimeter***.
- Im FEMM-Programm werden zusätzlich grundlegende Eigenschaften der späteren Vernetzung und Berechnung festgelegt:

```
-----------------------------------------------------------------
--------------
-- Definition des Problem-Typs
--
-----------------------------------------------------------------
--------------
   create(0);         --
0=magnetic|1=electrostatic|2=heatflow|3=currentflow
-- mi_probdef(frequency,units,type,precision,(depth),(minangle))
   mi_probdef(0,"millimeters","axi",1e-8,0,30);
                      -- 1e-8=Max.Fehler für Solver
                      -- 30° =Min.Winkel für Netz
```

Hinweis: Eine Frequenz=0 Hz beschreibt ein statisches Problem. Bei einem Wert>0 Hz würde eine harmonische Berechnung mit komplexen Zahlen durchgeführt, bei der alle Feldgrößen mit der gewählten Frequenz schwingen.

Parameter

Die geplanten Simulationsexperimente sind die Grundlage für die Definition einer Liste von Modellparametern. Durch Änderungen an diesen Modell-Parametern gewinnt der Modell-Anwender Erkenntnisse für die Lösung seines Entwurfsproblems. Man unterscheidet zwei Klassen von Modellparametern:

1. Entwurfsparameter (Objektbezogen)

- Konstruktive Parameter (Abmessungen und geometrische Form, Material und Zustandsgrößen: z.B. Vorspannung, Magnetisierung)
- Technologische Parameter (Streuungen und Grenzwerte konstruktiver Parameter)
- Betriebsparameter (aktuelle Werte veränderlicher Zustandsgrößen: z.B. Strom, Positionen)

2. Numerische Parameter (Modellbezogen)

- Vernetzungsvorgaben (Elementgröße und -form)
- Solver-Parameter (Verfahren, Genauigkeit)

Da wir Millimeter als günstige Längeneinheit bei der Problem-Definition wählten, müssen Abmessungen konsequent in der Maßeinheit mm angeben werden:

```
-------------------------------------------------------------------
--------------
-- Konstruktive, technologische und Betriebs-Parameter
--
-------------------------------------------------------------------
--------------
   dMagnet=40;   -- Topf-Durchmesser [mm]
   hTopf=27.5;   -- Topf-Höhe [mm]
   dAnker=10;    -- Anker-Durchmesser [mm]
   lAnker=10;    -- Anker-Länge [mm]
   hDeckel=2.5;  -- Dicke des Deckels bzw. Topfbodens [mm]
   dStift=1;     -- Durchmesser der Bohrung [mm]
   dWand=1;      -- Wandstärke des Topfes [mm]
   wSpule=400;   -- Windungen in der Spule
   dLack=0.06;   -- Gesamtdicke der Drahtbeschichtung [mm]
   sWickel=0.5;  -- Spalt zw. Spulendraht und Eisen [mm]
   sAnker=1;     -- Arbeitsluftspalt [mm]
   sDeckel=0.01; -- Restspalt zw. Deckel und Topf [mm]
   sGleit=0.1;   -- Führungsluftspalt für Anker [mm]
   iSpule=1;     -- Spulenstrom [A]
```

← →

Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld_-_Problemdefinition&oldid=25738“

Software: FEM - Tutorial - Magnetfeld - Material und Stromkreis

Aus OptiYummy

↑

← →
Material und Stromkreis

Inhaltsverzeichnis

1 Material
- 1.1 Draht
- 1.2 Stahl
- 1.3 Luft

2 Stromkreis

Material

Es ist sinnvoll, die Geometrie nicht losgelöst von den stofflich-physikalischen Eigenschaften zu beschreiben. Deshalb sollte man vor der Geometrie alle benötigten Modell-Materialien definieren. Als zusätzliche physikalische Eigenschaften sind z.B. in Magnetkreisen die Eigenschaften der Wicklungsstromkreise zu beschreiben.

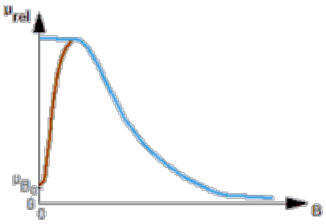
Die benötigten Materialien (Luft, Kupferdraht, Eisen) müssen innerhalb des Modells als "Model Materials" definiert werden.

Draht

- Für den Kupfer-Draht wurde die spezifische Leitfähigkeit von **58 MS/m** benutzt.
- Zur richtigen Modellierung des Wickelraums ist die benutzte Cu-Drahtstärke anzugeben:
  - Dafür gilt näherungsweise: $\mathbf{d=\sqrt{(h \cdot b)/w - dLack}}$.
  - Je nach vorgegebener Windungszahl und Magnet-Geometrie wird eine andere Drahtstärke gewählt. Dabei müsste man eigentlich berücksichtigen, dass nur eine bestimmte Reihe von Drahtdurchmessern verfügbar ist. Aus Aufwandsgründen nutzen wir hier nur obige Näherungsformel!

```
-----------------------------------------------------------------
--------------
-- Materialien (Luft, Kupfer, Eisen)
--
-----------------------------------------------------------------
--------------
-- mux, muy     : rel. Permeabilitaet in x- bzw. y-Richtung (hier
r bzw. z)
-- Hc           : Koerzitivfeldstärke [A/m] -> hier Null
-- J            : aktuelle Stromdichte [A/mm²] -> hier Null
-- Cduct        : spez. Leitfähigkeit [MS/m] -> nur bei
Kupferdraht
-- LamD         : Schichtdicke [mm] -> falls Laminat-Type<>0
-- PhiHmax      : Hysterese-Verlustwinkel [deg] -> genutzt für
BH-Kurve
-- LamFill      : Material-Fuellgrad des Volumens -> 1=komplett
gefüllt
-- LamTyp       : Laminierungstype, z.B. 0=massiv oder in Ebene /
3=Magnetspule
-- PhiHx, PhiHy : Hysterese-Verlustwinkel [deg] -> genutzt fuer
µ=konst.
-- nStr         : Zahl der Einzeldrähte in Spule -> 1 in der
Magnetspule
-- dWire        : Durchmesser Cu im Spulendraht [mm]
   bSpule       = 0.5*(dMagnet-dAnker)-dWand-2*sWickel; -- Breite
Wickelfenster
   hSpule       = hTopf-hDeckel-2*sWickel; -- Höhe Wickelfenster
   dWire        = sqrt(hSpule*bSpule/wSpule)-dLack;
   mue          = 400; -- nicht mehr wirksam, wenn B-H-Kennlinie
definiert!
-- mi_addmaterial('name'
,mux,muy,Hc,J,Cduct,LamD,PhiHmax,LamFill,LamTyp,PhiHx,PhiHy,nStr,
dWire);
   mi_addmaterial("Luft" ,  1,  1, 0,0,    0,   0,      0,
1,     0,    0,    0,   0,  0  );
   mi_addmaterial("Draht",  1,  1, 0,0,   58,   0,      0,
1,     3,    0,    0,   1,dWire);
   mi_addmaterial("Stahl",mue,mue, 0,0,    0,   0,      0,
1,     0,    0,    0,   0,  0  );
```

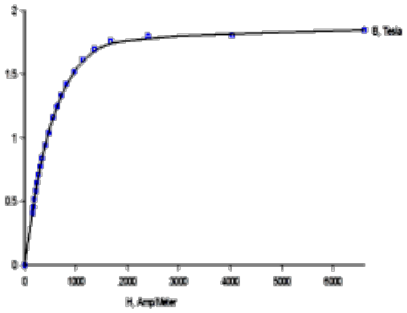
```
-- B-H-Kennlinie Stahl (Vielzahl der Ziffern aus Umrechnung der
μ(B)-Funktion)
   mi_addbhpoint("Stahl", 0       ,      0      );
   mi_addbhpoint("Stahl", 0.2004  ,    318.31   );
   mi_addbhpoint("Stahl", 0.6008  ,    636.62   );
   mi_addbhpoint("Stahl", 1.102   ,   1591.55   );
   mi_addbhpoint("Stahl", 1.386   ,   4774.65   );
   mi_addbhpoint("Stahl", 1.465   ,   7957.75   );
   mi_addbhpoint("Stahl", 1.667548,  32902.23251);
   mi_addbhpoint("Stahl", 1.789509,  92585.3973 );
   mi_addbhpoint("Stahl", 1.878209, 160685.3549 );
   mi_addbhpoint("Stahl", 2.08    , 318310      );
```



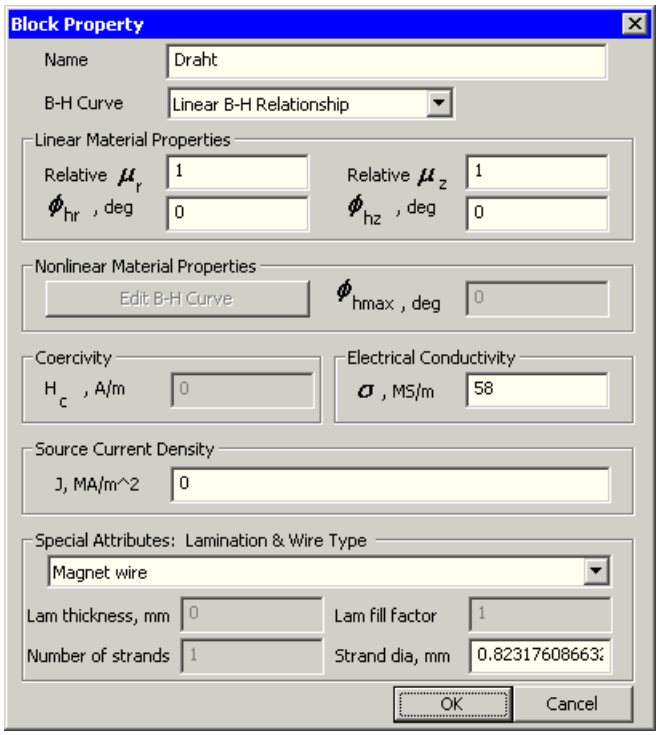
Die BH-Kurve muss in FE-Modellen so gestaltet werden, dass die daraus resultierende μ(B)-Funktion monoton fällt. Ansonsten entstehen numerische Probleme mit dem Gleichungslöser infolge von Mehrdeutigkeiten.

Die originale μ(B)-Kurve wird deshalb für kleine Flussdichten nicht nachgebildet, sondern die μ(B)-Kurve beginnt mit einem μ, welches etwas größer als der eigentliche Maximalwert von μ ist!

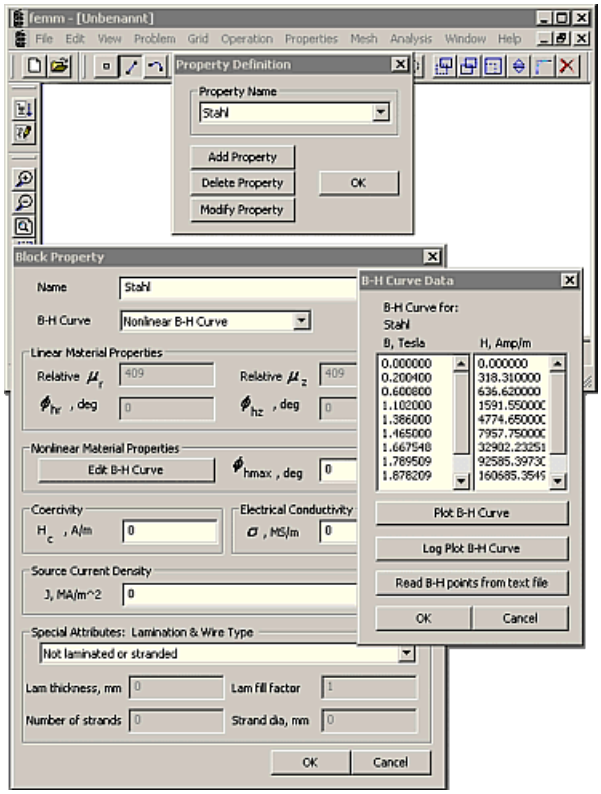
Das FEMM-System benötigt die zugehörige BH-Kennlinie, die in Form von Stützstellen einzugeben ist. Über diese Stützstellen interpoliert der Solver mittels einer Spline-Funktion. Da für die Berechnung der Haltekraft die Sättigung des Eisens bei hohen Flussdichten entscheidend ist, muss viel Sorgfalt auf die Nachbildung des μ-Wertes für hohe Flussdichten aufgewandt werden. Dieser Wert entspricht dem Anstieg der BH-Kennlinie am Ende. Hier muss (wie im Beispiel) meist noch eine "künstliche" Stützstelle für sehr hohe Feldstärken ergänzt werden.



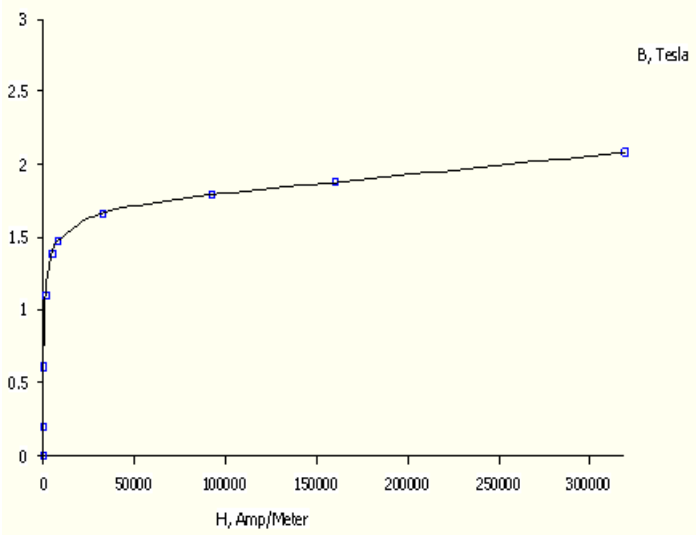
Nachdem im FEMM obige LUA-Script-Anweisungen abgearbeitet wurden, kann man die daraus resultierenden Material-Definitionen überprüfen (Menü-Punkt ***Properties - Material***):



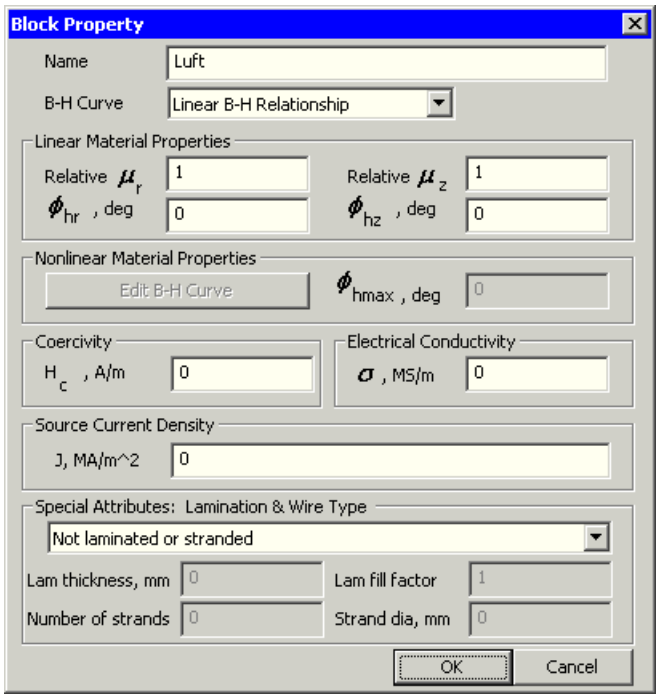
Stahl



- Nach dem Selektieren von Stahl gelangt man über den Button **Modify Property** zu den Stahl-Parametern.
- Die elektrische Leitfähigkeit des Eisenmaterials wurde Null gesetzt, weil sie für statische Berechnungen keine Rolle spielt (keine Wirbelströme)!
- Der Button **Edit B-H-Curve** führt zur Liste der Kennlinien-Punkte.
- Mittels **Plot B-H-Curve** erhält man die Grafik des zugehörigen Funktionsverlaufs:



Luft



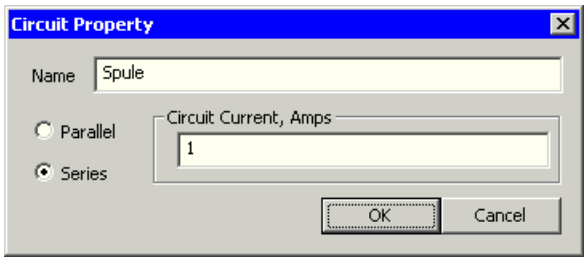
Stromkreis

- Der Bereich der Spulenwicklung muss noch näher spezifiziert werden. Es handelt sich um einen elektrischen Kreis mit eingeprägtem Strom.
- Eigenschaften solcher elektrischen Kreise werden im FEMM als "Circuit Property" beschrieben:

```
-----------------------------------------------------------------
--------------
-- Circuit Property für die Wicklung definieren (Strom und
Wicklungstyp      --
-----------------------------------------------------------------
--------------
-- mi_addcircprop("circuitname", i, type) i: Strom / type:
0=parallel 1=seriell
   mi_addcircprop("Spule", iSpule, 1);
```

- Nach Abarbeitung des LUA-Scripts kann man die erfolgreiche Definition dieses Spulen-

Stromkreises auf der grafischen Oberfläche unter ***Properties - Circuits - Modify Property*** überprüfen:



← →

Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld_-_Material_und_Stromkreis&oldid=15331“

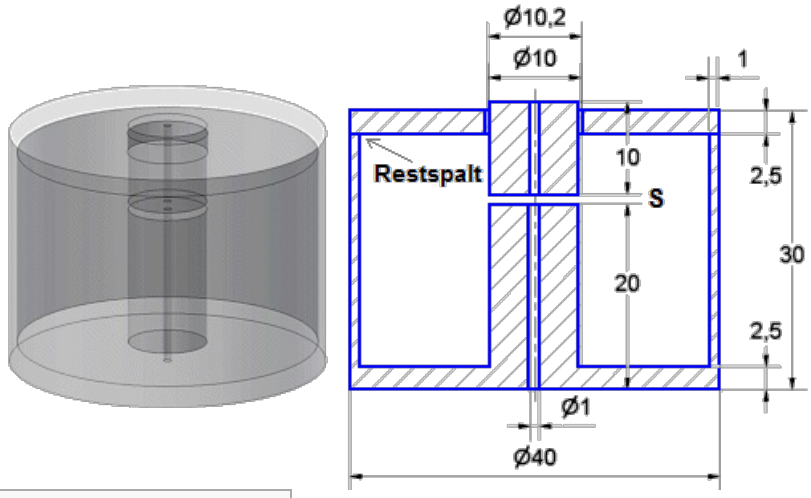
Software: FEM - Tutorial - Magnetfeld - Geometrie

Aus OptiYummy

↑

← →

Geometrie



Inhaltsverzeichnis

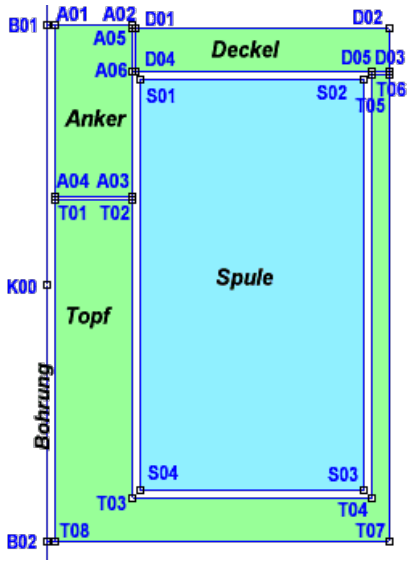
- 1 Hinweise zur Geometrie-Definition
- 2 Unendlicher Luftraum
- 3 Wickelraum der Spule
- 4 Eisenkreis (Topf)
- 5 Eisenkreis (Anker)
- 6 Eisenkreis (Deckel)
- 7 Arbeitsluftspalt (Topf/Anker)
- 8 Restspalt (Deckel/Topf)
- 9 Führungsspalt (Deckel/Anker)
- 10 Bohrung
- 11 Wickelspalt (Spule/Eisenkreis)

Hinweise zur Geometrie-Definition

Ein FEM-Modell setzt sich im Normalfall aus mehreren geometrischen Teilbereichen zusammen. Die Strukturierung in ´Teilbereiche wird durch zwei Aspekte bestimmt:

1. Abgrenzung nach stofflich-physikalischen Eigenschaften (Material, Zustandsgrößen, Isotropie)
2. Berücksichtigung numerischer Erfordernisse in Abhängigkeit von den zu erwartenden Feldgradienten (feinere Vernetzung enger Spalten oder von Bereichen mit starken Feldgradienten).

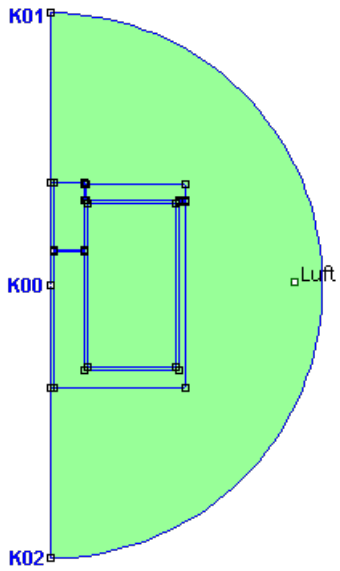
Alle markanten Punkte der abgegrenzten Bereiche muss man als *Nodes* definieren. Um bei der Vielzahl der zu definierenden Koordinaten die Übersicht nicht zu verlieren, sollte man sich bei den *Node*-Bezeichnern an den Namen der Komponenten orientieren (z.B. Anfangsbuchstaben):



- Das magnetische Feld einer Spule erstreckt sich induziert durch den Stromfluss in den unendlichen Luftraum ($\mu_{rel}=1$). Dieses Feld im Luftraum nennt man auch Streufeld der Spule.
- Vereinfacht kann man sich vorstellen, dass die Komponenten des Eisenkreises dem Streuleitwert des Luftraumes parallel geschalten sind. Auf Grund der viel besseren magnetischen Leitfähigkeit fließt der größte Teil des magnetischen Flusses durch das Eisen, anstatt durch die benachbarten Luftbereiche.
- Ausgehend von dieser vereinfachten Vorstellung sollte man auch das Modell eines Magneten aufbauen:

  1. Beschreibung des unendlichen Raumes ($\mu_{rel}=1$).
  2. Definition der magnetischen MMK (Spule, Permanentmagnet)
  3. Beschreibung der Bereiche mit $\mu_{rel}\neq 1$ (z.B. Eisen) und Definition von Luftbereichen mit besonderen Forderungen an die Feinheit der Vernetzung (z.B. kleine Spalte zwischen den Eisenkreis-Komponenten).

- Bei der Reihenfolge der Beschreibung muss man darauf achten, dass Zusammenbau-Abhängigkeiten zwischen den Komponenten des Eisenkreises existieren. Das äußert sich in der Abhängigkeit der Eckpunkt-Koordinaten. So orientieren sich z.B. die Platzierungen von Anker und Deckel an der Größe und Form des Topfes.



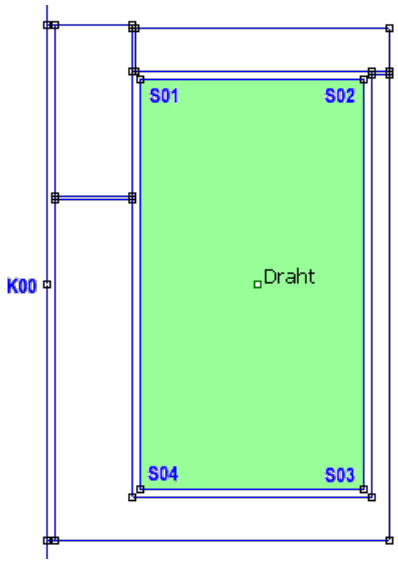
Unendlicher Luftraum

- In unserem Beispiel hat infolge des geschlossenen Eisenkreises der "unendliche" Luftraum um den Magneten wahrscheinlich nur einen geringen Einfluss. Trotzdem wollen wir die Eigenschaften dieses "unendlichen" Raumes wie im Beispiel der elektrischen Kapazität mit Hilfe einer Kugelsphäre nachbilden, die man mit geeigneten Randbedingungen versieht.

- Der Radius der Kugelsphäre sollte in jeder Richtung wenigstens doppelt so groß sein, wie die Ausdehnung der Magnetbaugruppe. Wenn wir als Radius der Kugelsphäre die Summe aus Durchmesser und Höhe des Magneten benutzen, sollte das für beliebig proportionierte Magneten reichen.
- Das Zentrum **K00** der Kugel legen wir in den Koordinaten-Ursprung (0,0), die Geometrie des Magneten ist in der Skizze bereits angedeutet.
- ***Achtung:*** Es handelt sich nun um Zylinderkoordinaten mit Radius **r** und Höhe **z**!

```
-----------------------------------------------------------------
--------------
-- Geometrie (Zylinder-Koordinaten mit r,z: alle Abmessungen in
mm!)         --
-- Blocklabels in alle Geometriebereiche und Zuweisen der
Materialien        --
-----------------------------------------------------------------
--------------
                                --
*********************************************
   mi_seteditmode ("nodes");   -- K=Kugelsphäre
******************************
                                --
*********************************************
   rKugel=dMagnet+hTopf+hDeckel;
   rK00= 0;           zK00= 0;           mi_addnode (rK00,zK00);
--Ursprung 0,0
   rK01= 0;           zK01= rKugel;      mi_addnode (rK01,zK01);
--Oben
   rK02= 0;           zK02=-rKugel;      mi_addnode (rK02,zK02);
--Unten
   mi_zoomnatural(); mi_zoomout();            -- Zoom incl. Achse
im Bild
   mi_seteditmode ("arcsegments");
-- mi_addarc(  r1,z1   ,   r2,z2   ,angle, maxseg deg.);
   mi_addarc(rK02,zK02 , rK01,zK01 , 180 , 5) -- Halbkreis
entgegen Uhrzeiger
   mi_seteditmode ("segments");               -- Symmetrieachse
im Halbkreis
   mi_addsegment  ( rK01,zK01 , rK02,zK02 );  ------------------
--------------
   mi_seteditmode ("blocks");                 -- Block Luft mit
Automesh=1   --
   r0_K=0.9*rKugel;              z0_K=0;      -- 10% vom rechten
Rand        --
   mi_addblocklabel(r0_K, z0_K); mi_selectlabel(r0_K,z0_K);
   mi_setblockprop("Luft",1,0,"<None>",0,0,0);
mi_clearselected();
--
mi_setblockprop("name",automesh,meshsize,"incircuit",magdir,group
,turns);
```

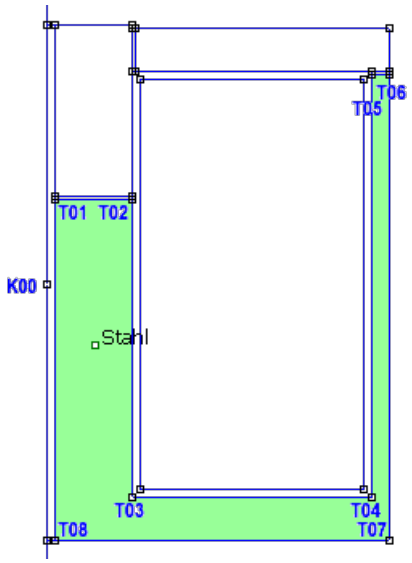
Wickelraum der Spule



- Falls es wie im Beispiel auf Grund der konkreten Form möglich ist, sollte man die Spule zentrisch in der "unendlichen" Kugelsphäre anordnen.
- Wir zeichnen die Spule als Rechteck (**hSpule** x **bSpule**) auf der Basis von Punkten und Linien.
- Die Abmessungen des Spulen-Wickelfensters berechnen wir vor der Definition der Eck-Knoten:

```
--                                --
*********************************************
   mi_seteditmode  ("nodes");  -- S=Spulen-Wickelfenster
*********************
--                                --
*********************************************
   rS01= dAnker/2+sWickel;zS01= hSpule/2;
mi_addnode (rS01,zS01);
   rS02= rS01+bSpule;     zS02= zS01;
mi_addnode (rS02,zS02);
   rS03= rS02;            zS03=-hSpule/2;
mi_addnode (rS03,zS03);
   rS04= rS01;            zS04= zS03;
mi_addnode (rS04,zS04);
   mi_seteditmode  ("segments");
   mi_addsegment   ( rS01,zS01 , rS02,zS02 );
   mi_addsegment   ( rS02,zS02 , rS03,zS03 );
   mi_addsegment   ( rS03,zS03 , rS04,zS04 );
   mi_addsegment   ( rS04,zS04 , rS01,zS01 );    ----------------
--------------
   mi_seteditmode  ("blocks");                   -- Block "Draht"
als "Spule"--
   r0_S=rS01+(rS02-rS01)/2; z0_S=zS04+(zS01-zS04)/2;-- in Mitte
Wickelfenster--
   mi_addblocklabel(r0_S, z0_S);
mi_selectlabel(r0_S,z0_S);
   mi_setblockprop ("Draht",1,0,"Spule",0,0,wSpule);
mi_clearselected();
```

Eisenkreis (Topf)

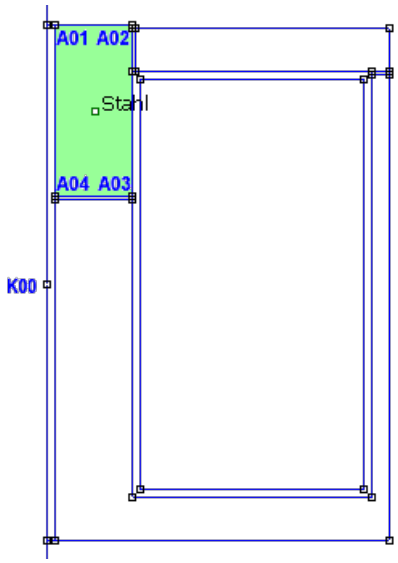


- Wenn der Arbeitsluftspalt **sAnker** und der Restspalt **sDeckel** gleich Null sind, schließt der Anker oben mit dem Deckel fluchtend ab.

Dies wird bei der Berechnung der Eckpunkte des Topf-Schnittes berücksichtigt:

```
--                              --
*********************************************
   mi_seteditmode  ("nodes"); -- T=Eisenkreis (Topf)
************************
--                              --
*********************************************
   rT01= dStift/2;       zT01= (hTopf+hDeckel)/2-lAnker;
mi_addnode(rT01,zT01);
   rT02= dAnker/2;       zT02= zT01;
mi_addnode(rT02,zT02);
   rT03= rT02;           zT03=-(hTopf-hDeckel)/2;
mi_addnode(rT03,zT03);
   rT04= dMagnet/2-dWand;zT04= zT03;
mi_addnode(rT04,zT04);
   rT05= rT04;           zT05=-zT04;
mi_addnode(rT05,zT05);
   rT06= dMagnet/2;      zT06= zT05;
mi_addnode(rT06,zT06);
   rT07= rT06;           zT07=-(hTopf+hDeckel)/2;
mi_addnode(rT07,zT07);
   rT08= rT01;           zT08= zT07;
mi_addnode(rT08,zT08);
   mi_seteditmode  ("segments");
   mi_addsegment   ( rT01,zT01 , rT02,zT02 );
   mi_addsegment   ( rT02,zT02 , rT03,zT03 );
   mi_addsegment   ( rT03,zT03 , rT04,zT04 );
   mi_addsegment   ( rT04,zT04 , rT05,zT05 );
   mi_addsegment   ( rT05,zT05 , rT06,zT06 );
   mi_addsegment   ( rT06,zT06 , rT07,zT07 );
   mi_addsegment   ( rT07,zT07 , rT08,zT08 );
   mi_addsegment   ( rT08,zT08 , rT01,zT01 ); -------------------
--------------
   mi_seteditmode  ("blocks");                -- Block "Stahl" in
Topfmitte  --
   r0_T=rT01+(rT02-rT01)/2; z0_T=zT03+(zT02-zT03)/2;
   mi_addblocklabel(r0_T, z0_T);
mi_selectlabel(r0_T,z0_T);
   mi_setblockprop ("Stahl",1,0,"<None>",0,0,0);
mi_clearselected();
```

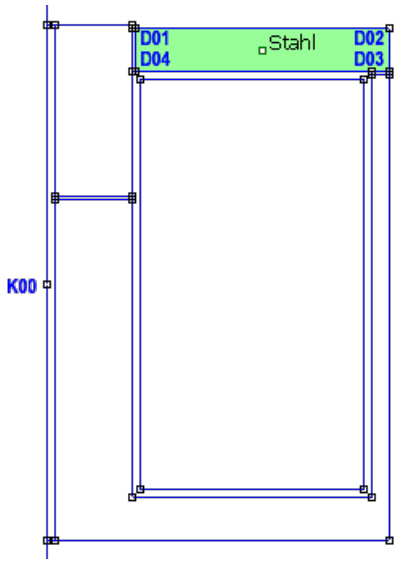
Eisenkreis (Anker)



- Die Position des Ankers ist durch den Topf und den Arbeitsluftspalt eindeutig bestimmt.
- Bei Arbeitsluftspalt **sAnker=0** soll der Anker fluchtend mit der Oberseite des Deckels abschließen.

```
--                                --
*********************************************
   mi_seteditmode  ("nodes"); -- A=Eisenkreis (Anker)
************************
--                                --
*********************************************
   rA01= dStift/2;        zA01= zT01+sAnker+lAnker;
mi_addnode(rA01,zA01);
   rA02= dAnker/2;        zA02= zA01;
mi_addnode(rA02,zA02);
   rA03= rA02;            zA03= zT01+sAnker;
mi_addnode(rA03,zA03);
   rA04= rA01;            zA04= zA03;
mi_addnode(rA04,zA04);
   mi_seteditmode  ("segments");
   mi_addsegment   ( rA01,zA01 , rA02,zA02 );
   mi_addsegment   ( rA02,zA02 , rA03,zA03 );
   mi_addsegment   ( rA03,zA03 , rA04,zA04 );
   mi_addsegment   ( rA04,zA04 , rA01,zA01 ); -------------------
--------------
   mi_seteditmode  ("blocks");                -- Block "Stahl" in
Ankermitte --
   r0_A=rA01+(rA02-rA01)/2; z0_A=zA03+(zA02-zA03)/2;
   mi_addblocklabel(r0_A, z0_A);
mi_selectlabel(r0_A,z0_A);
   mi_setblockprop ("Stahl",1,0,"<None>",0,0,0);
mi_clearselected();
```

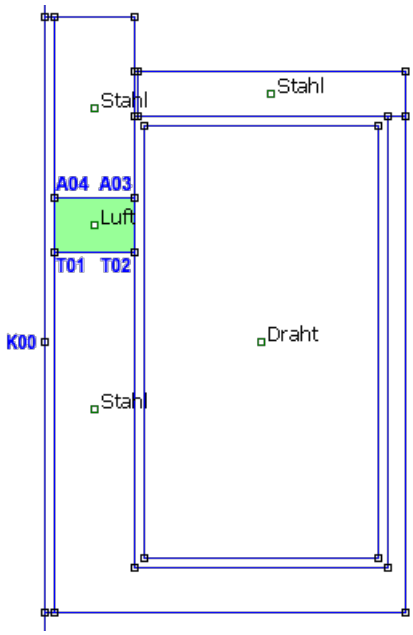
Eisenkreis (Deckel)



- Da der Spalt **sDeckel=10 µm** zwischen Topf und Deckel praktisch unsichtbar ist, sollte man zur Überprüfung der aus dem Script resultierenden Geometrie diesen Spalt temporär vergrößern.

```
--                                --
*********************************************
   mi_seteditmode  ("nodes"); -- D=Eisenkreis (Deckel)
**********************
--                                --
*********************************************
   rD01= dAnker/2+sGleit; zD01= zT06+sDeckel+hDeckel;
mi_addnode(rD01,zD01);
   rD02= dMagnet/2;       zD02= zD01;
mi_addnode(rD02,zD02);
   rD03= rD02;            zD03= zD01-hDeckel;
mi_addnode(rD03,zD03);
   rD04= rD01;            zD04= zD03;
mi_addnode(rD04,zD04);
   mi_seteditmode  ("segments");
   mi_addsegment   ( rD01,zD01 , rD02,zD02 );
   mi_addsegment   ( rD02,zD02 , rD03,zD03 );
   mi_addsegment   ( rD03,zD03 , rD04,zD04 );
   mi_addsegment   ( rD04,zD04 , rD01,zD01 ); -------------------
--------------
   mi_seteditmode  ("blocks");                -- Block "Stahl" in
Deckelmitte--
   r0_D=rD01+(rD02-rD01)/2; z0_D=zD03+(zD02-zD03)/2;
   mi_addblocklabel(r0_D, z0_D);
mi_selectlabel(r0_D,z0_D);
   mi_setblockprop ("Stahl",1,0,"<None>",0,0,0);
mi_clearselected();
```

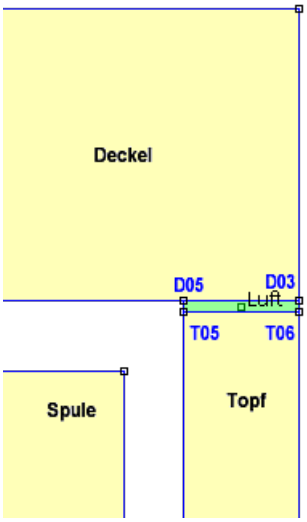
Arbeitsluftspalt (Topf/Anker)



Es genügen zwei zusätzliche senkrechte Linien-Segmente, um den Arbeitsluftspalt als Bereich abzugrenzen. Dafür kann man bereits definierte Knoten von Topf und Anker nutzen:

```
--                                --
*********************************************
   mi_seteditmode("segments");-- L=Arbeitsluftspalt (Topf/Anker)
*************
--                                --
*********************************************
   mi_addsegment   ( rA04,zA04 , rT01,zT01 );
   mi_addsegment   ( rA03,zA03 , rT02,zT02 ); -------------------
-------------
   mi_seteditmode  ("blocks");                -- Block "Luft" in
Spaltmitte --
   r0_L=rA04+(rA03-rA04)/2; z0_L=zT01+(zA04-zT01)/2;
   mi_addblocklabel(r0_L, z0_L);
mi_selectlabel(r0_L,z0_L);
   mi_setblockprop ("Luft",1,0,"<None>",0,0,0);
mi_clearselected();
```

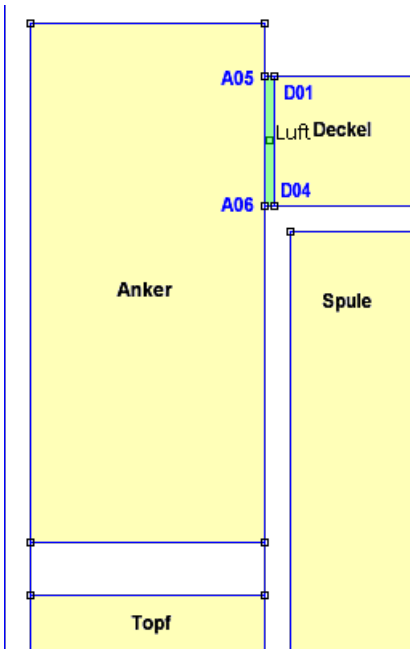
Restspalt (Deckel/Topf)



- Zur Verdeutlichung wurde der Restspalt **sDeckel** temporär auf 0.1 mm vergrößert.
- Für die Abgrenzung des Bereiches benötigt man einen zusätzlichen Knoten **D05** am Deckel und zwei senkrechte Segmente:

```
--                                --
*********************************************
   mi_seteditmode  ("nodes"); -- R=Restspalt (Deckel/Topf)
*******************
--                                --
*********************************************
   rD05= rT05;            zD05= zD03;
mi_addnode(rD05,zD05);
   mi_seteditmode  ("segments");
   mi_addsegment   ( rD05,zD05 , rT05,zT05 );
   mi_addsegment   ( rD03,zD03 , rT06,zT06 );  ------------------
--------------
   mi_seteditmode  ("blocks");                 -- Block "Luft" in
Spaltmitte --
   r0_R=rD05+(rD03-rD05)/2; z0_R=zT05+(zD05-zT05)/2;
   mi_addblocklabel(r0_R, z0_R);
mi_selectlabel(r0_R,z0_R);
   mi_setblockprop ("Luft",1,0,"<None>",0,0,0);
mi_clearselected();
```

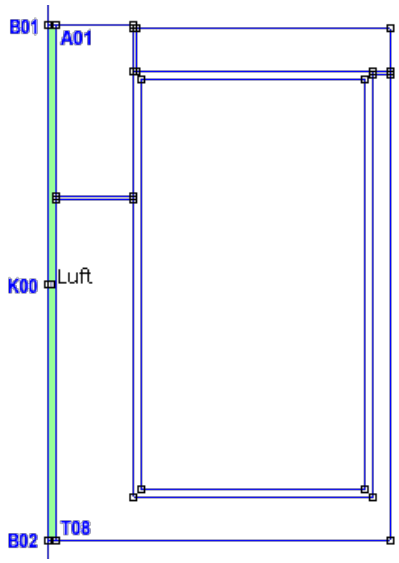
Führungsspalt (Deckel/Anker)



- Zur Verdeutlichung wurde der Führungsspalt **sGleit** temporär auf 0.2 mm vergrößert.
- Für die Abgrenzung des Bereiches benötigt man zwei zusätzliche Knoten **A05** und **A06** am Anker und zwei waagerechte Segmente:

```
--                                --
*********************************************
   mi_seteditmode  ("nodes"); -- F=Führungsspalt (Deckel/Anker)
**************
--                                --
*********************************************
   rA05= rD01-sGleit;      zA05= zD01;
mi_addnode(rA05,zA05);
   rA06= rA05;             zA06= zD04;
mi_addnode(rA06,zA06);
   mi_seteditmode  ("segments");
   mi_addsegment   ( rA05,zA05 , rD01,zD01 );
   mi_addsegment   ( rA06,zA06 , rD04,zD04 );  ------------------
--------------
   mi_seteditmode  ("blocks");                 -- Block "Luft" in
Spaltmitte --
   r0_F=rA05+(rD01-rA05)/2; z0_F=zA06+(zA05-zA06)/2;
   mi_addblocklabel(r0_F, z0_F);
mi_selectlabel(r0_F,z0_F);
   mi_setblockprop ("Luft",1,0,"<None>",0,0,0);
mi_clearselected();
```

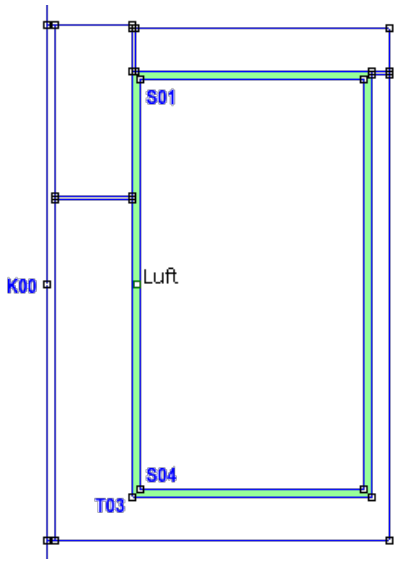
Bohrung



Um die Vernetzung in der Bohrung unabhängig von der Kugelsphäre steuern zu können, sollte man den Bereich der Bohrung abgrenzen. Man benötigt dafür zwei zusätzliche Knoten und zwei waagerechte Segmente:

```
--                                --
*********************************************
   mi_seteditmode  ("nodes"); -- B=Bohrung
*********************************
--                                --
*********************************************
   rB01= 0;                zB01= zA01;
mi_addnode(rB01,zB01);
   rB02= 0;                zB02= zT08;
mi_addnode(rB02,zB02);
   mi_seteditmode  ("segments");
   mi_addsegment   ( rB01,zB01 , rA01,zA01 );
   mi_addsegment   ( rB02,zB02 , rT08,zT08 );--------------------
--------------
   mi_seteditmode  ("blocks");               -- Block "Luft" in
Bereichsmitte-
   r0_B=rA01/2; z0_B=0;
   mi_addblocklabel(r0_B, z0_B);
mi_selectlabel(r0_B,z0_B);
   mi_setblockprop ("Luft",1,0,"<None>",0,0,0);
mi_clearselected();
```

Wickelspalt (Spule/Eisenkreis)



Der Luftraum um die Spulenwicklung ist bereits komplett als Bereich abgegrenzt. Es fehlt nur das Blocklabel:

```
--                                --
*********************************************
   mi_seteditmode ("blocks"); -- W=Wickelspalt
******************************
--                                --
*********************************************
   r0_W=rT03+(rS04-rT03)/2; z0_W=zS04+(zS01-zS04)/2; --> in Mitte
innerer Spalt
   mi_addblocklabel(r0_W, z0_W);
mi_selectlabel(r0_W,z0_W);
   mi_setblockprop("Luft",1,0,"<None>",0,0,0);
mi_clearselected();
```

← →

Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld_-_Geometrie&oldid=13339“

Software: FEM - Tutorial - Magnetfeld - Randbedingungen

Aus OptiYummy

↑

← →

Randbedingungen

Überblick

- Über Zwangsbedingungen kann man definieren, wie sich das Feld über den Rand des Modells hinaus ausbreiten soll. Die richtigen Randbedingungen sind entscheidend für die Modell-Genauigkeit.
- Das ist abhängig von der Art des Materials, welches an den Modellrand grenzt und der Form des Modellrandes. Die Formulierung der Randbedingungen ist oft kein triviales Problem!
- Im Beispiel des magnetischen Wirbelfeldes muss man den unendlichen Luftraum nachbilden, in den sich das Streufeld des Magneten erstreckt ("Open Boundary Problem").

Für das elektrostatische Problem der Leiter-Kapazität hatten wir solch ein "Open Boundary Problem" bereits definiert. An dieser Stelle sollen die verschiedenen Ansätze für diese Art von Randbedingungen insbesondere unter dem Aspekt des Magnetfeldes näher betrachtet werden. Für "Open Boundary Problems" kann man drei grundsätzliche Ansätze unterscheiden, welche nach Mathematikern benannt sind:

1. **Dirichlet** (*Mathematiker 1805-1859*)
   Wird die abhängige Variable auf dem Rand direkt vorgegeben, liegt eine "*Dirichlet Randbedingung*" vor. Im Wirbelfeld erhält das Vektorpotential am Rand des Modells einen festen Wert. Für magnetische Probleme wird dabei das Vektorpotential entlang der Grenzlinie meist $\boldsymbol{A=0}$ gesetzt. Damit verhindert man den magnetischen Fluss über diese Grenzlinie hinweg.
   Diese Art der Randbedingung setzt voraus, dass um den betrachteten Magnetkreis sich möglichst wenig Streufeld in den Luftraum erstreckt. Bei dem betrachteten Topfmagneten wäre dies der Fall. Wenn man hier einen hinreichend großen Luftraum modelliert (ca. 5x so groß wie der Magnet selbst), so könnte man den äußeren Rand dieses Bereiches mit dem Vektorpotential $\boldsymbol{A=0}$ belegen.
   Hinweis: Für axialsymmetrische Probleme wird die Symmetrieachse (**r=0**) automatisch mit $\boldsymbol{A=0}$ belegt.
2. **Neumann** (*Mathematiker 1832-1925*)
   Wird nicht der Wert selbst, sondern seine örtliche Ableitung senkrecht zur Grenzlinie vorgegeben, liegt eine "*Neumann Randbedingung*" vor. Für magnetische Probleme (mit $\boldsymbol{n}$=Normalenvektor zur Grenze) setzt man $\boldsymbol{\delta A/\delta n = 0}$.
   Diese Randbedingung zwingt den magnetischen Fluss, exakt im Winkel von 90° die Grenze zu durchdringen. Dies beschreibt einen Übergang zwischen einem Material geringer Permeabilität im Modell (z.B. μ_0 für Luft) und einem Material hoher Permeabilität außerhalb des Modells (z.B. Eisen).
   Hinweise:
   1. Wird im FEMM-Modell keine Randbedingung definiert, so wird standardmäßig die "*Neumann Randbedingung*" verwendet. Das widerspiegelt aber selten die Realität (Luft um den Magneten)!
   2. Zusätzlich zur "*Neumann Randbedingung*" muss mindestens eine ableitungsfreie Randbedingung existieren, um eine eindeutige Lösung zu erhalten! Für axialsymmetrische Magnetprobleme ist dies durch die Belegung der Symmetrieachse gesichert. Ansonsten muss für mindestens einen Punkt (Knoten des Netzes) der Wert des Vektorpotentials vorgegeben werden.
3. **Robin** (*Mathematiker 1855-1897*)

Bei der "*Robin Randbedingung*" handelt es sich um einen Mix aus obigen beiden Formen. Beschrieben wird eine Abhängigkeit zwischen dem Wert von ***A*** und seiner "Normalen-Ableitung" an der Grenze. Ein Beispiel für solch eine Randbedingung ist $\delta A/\delta n + c \cdot A = 0$.
Durch diese Randbedingung ist es möglich, mit einem begrenzten Modellbereich das Verhalten einer unbegrenzten (unendlich großen) Region nachzubilden.
Hinweis: Auch hier ist zusätzlich eine ableitungsfreie Randbedingung für eine eindeutige Lösung erforderlich!

Modellierung

In unserem Beispiel soll der unendliche Luftraum durch eine Kugelgrenzfläche mit dem Radius R=***rKugel*** [mm] nachgebildet werden. Dafür ist eine geeignete Robin-Randbedingung zu definieren. Die "asymptotische Randbedingung" beschreibt den Verlauf der Feldlinien an der betrachteten Grenze. Die folgende Gleichung setzt voraus, dass auf beiden Seiten der Grenze die gleiche relative Permeabilität existiert und die Grenze selbst eine Kugelfläche darstellt:

$$\frac{1}{\mu_r \mu_o}\frac{\partial A}{\partial n} + c_o A + c_1 = 0 \text{ mit } c_o = \frac{1}{\mu_r \mu_o R} \text{ und } c_1 = 0$$

Da unsere Grenze in Luft verläuft, ist μ_r**=1**. Der Radius **R** ist in Meter anzugeben, unabhängig von der gewählten Maßeinheit des Problems! ***A*** ist das magnetische Vektorpotential und ***n*** repräsentiert den Normalenvektor zur Grenzfläche, in unserem Beispiel der Kugeloberfläche.

Hinweis: Der Zählerwert=**1** im Bruch für die Berechnung von c_0 ist der problemabhängige Wert für ***n*** und gilt nur für den speziellen Fall des magnetischen Dipols im achsensymmetrischen Problem! Falls man z.B. statt einer Dipolspule mehrpolige Permanentmagnet-Anordnungen berechnen möchte, muss man für ***n*** einen anderen Wert einsetzen.

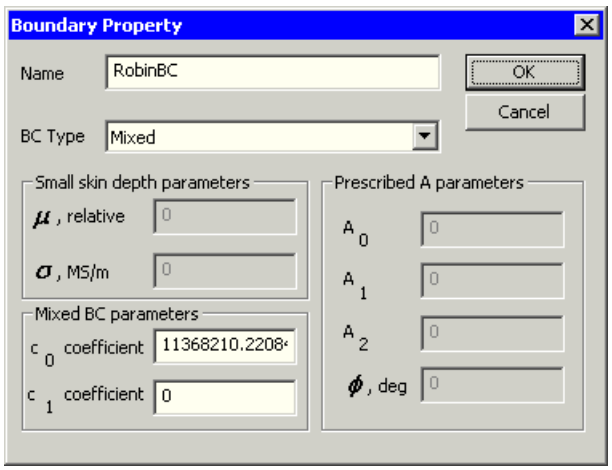
Robin-Randbedingung (Mixed-Type):

```
-----------------------------------------------------------------
--------------
-- Randbedingungen für die Kugelsphäre als "Open Boundary
Conditions"        --
-----------------------------------------------------------------
--------------
-- mi_addboundprop("name", A0, A1, A2, Phi, Mu, Sig, c0, c1,
BdryFormat);
-- BC-Type Mixed: DdryFormat=2, c0 und c1 wie berechnet, alle
anderen Parm=0
   c0=1/(uo*rKugel*mm); c1=0;
   mi_addboundprop("RobinBC", 0, 0, 0, 0, 0, 0, c0, c1, 2);
```

- c_0 wird abhängig von der aktuellen Magnetgröße berechnet: **1/(uo*rKugel*mm)**.
- In der Lua Scriptsprache von FEMM ist **uo=**μ_0 . Das trotz Angabe von ***rKugel*** in [mm] der Radius in Metern benutzt wird, ist Aufgabe des Faktors **mm=0.001**.

Den Erfolg der Definition sollte man wieder überprüfen:

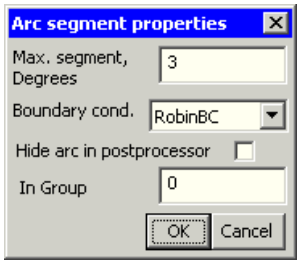
- Über ***Properties - Boundary - Modify Property*** erscheint der Parameter-Dialog für die Randbedingung:



- Diese Randbedingung muss nun der Kugeloberfläche zugewiesen werden:

```
   mi_seteditmode ("arcsegments");
-- mi_selectarcsegment (  r   ,z); -- Select the arc segment
closest to (r,z)
   mi_selectarcsegment (rKugel,0); --> Mittelpunkt des Bogens
-- mi_setarcsegmentprop(maxsegdeg,"propname",hide,group);
   mi_setarcsegmentprop(   3     ,"RobinBC" ,  0 , 0   );
   mi_clearselected();
```

Den Erfolg der Zuweisung sollte man nach Abarbeitung des Lua-Scripts wieder über die grafische Oberfläche überprüfen:



- Im Kreissegment-Modus und wählt man den Halbkreis durch Anklicken mit der rechten Maustaste.
- Nach Betätigen der Leertaste oder gelangt man in den Dialog, der die Zuordnung der Randbedingung anzeigt.

← →

Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld_-_Randbedingungen&oldid=22471“

Software: FEM - Tutorial - Magnetfeld - Vernetzung und Berechnung

Aus OptiYummy

↑

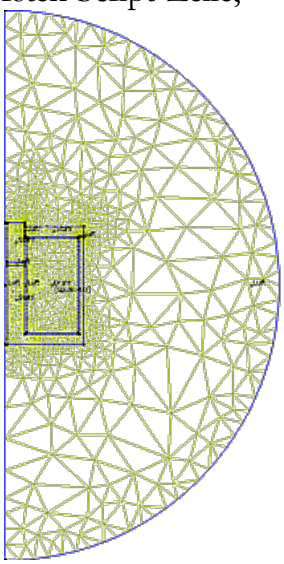
← →
Vernetzung und Berechnung

Inhaltsverzeichnis

- 1 Globale Pfad-Variable
- 2 Automatische Vernetzung
- 3 Vernetzungssteuerung
  - 3.1 Führungsspalt
  - 3.2 Wickelspalt
  - 3.3 Bohrung
  - 3.4 Arbeitsluftspalt
  - 3.5 Restspalt
- 4 Feldberechnung und Darstellung

Globale Pfad-Variable

- Spätestens jetzt muss man das Speichern des FEM-Modells veranlassen, da erst danach die Vernetzung durchgeführt werden kann.
- Im Sinne der Portabilität definieren wir dafür eine globale Pfad-Variable in der ersten Script-Zeile, bevor wir das Script fortschreiben:



```
   mydir="";           -- es wird der aktuelle Pfad der
geöffneten LUA-Datei benutzt
-- mydir="E:\\femm\\"; -- Beispiel für eine absolute Pfadangabe
   :
-----------------------------------------------------------------
--------------
-- Vernetzung und Feld-Berechnung  (nach Speichern des Modells)
--
-----------------------------------------------------------------
--------------
   mi_saveas(mydir .. "Magnet_xx.fem"); -- Speichern in
vorhandenen Ordner
   :
-----------------------------------------------------------------
--------------
-- Loeschen nicht mehr benoetigter Dateien am Ende des Lua-
Scripts           --
-----------------------------------------------------------------
--------------
   remove(mydir .. "Magnet_xx.fem"); -- Loeschen der Modelldatei
   remove(mydir .. "Magnet_xx.ans"); -- Loeschen der Grafikdatei
```

- Leider kann es manchmal zu ganz gemeinen Blockierungsproblemen mit derart erzeugten **.fem**-Dateien kommen (*Siehe:* ***Workaround***).
- Wir müssen deshalb darauf achten, dass wir diese Datei am Ende des Lua-Scripts wieder löschen! Das gleiche gilt für die grafische **.ans**-Ergebnisdatei, welche wir im Folgenden noch erzeugen

Automatische Vernetzung

Wir hatten für alle Bereiche des Modells eingestellt, dass der Vernetzungsgenerator "Triangle" selbst die erforderliche Maschengröße ermitteln soll:

```
   mi_createmesh(); -- Erzeugen des Netzes
```

- Das Ergebnis der automatischen Vernetzung sieht auf den ersten Blick nicht schlecht aus. So werden z.B. in den kleineren Bereichen feinere Netze genutzt (weil größere Maschen nicht hineinpassen!).
- Schaut man sich die Vernetzung in den Luftspalten jedoch genauer an, erkennt man, dass grundsätzlich nur eine Element-Schicht benutzt wird. Das betrifft:

  1. Führungsspalt zwischen Anker und Deckel
  2. Wicklungsspalt zwischen Spule und Eisenkreis
  3. Bohrungsbereich
  4. Arbeitsluftspalt zwischen Anker und Topf
  5. Restspalt zwischen Topf und Deckel

- Insbesondere für die Ermittlung der Magnetkraft (z.B. auf den beweglichen Anker) wird im FEMM-Manual eine mehrlagige Vernetzung der Luft um den untersuchten Eisenbereich gefordert. Nur dadurch wird gewährleistet, dass die Feldgradienten hinreichend genau berechnet werden.
- Vergleichsrechnungen zeigten zwar, dass die dadurch bedingten Abweichungen bei der Magnetkraft-

Berechnung kleiner als 1% sind. Trotzdem werden wir in Hinblick auf eine Minimierung numerischer Ungenauigkeiten die Luftspalte mit mehreren Lagen vernetzen.

Vernetzungssteuerung

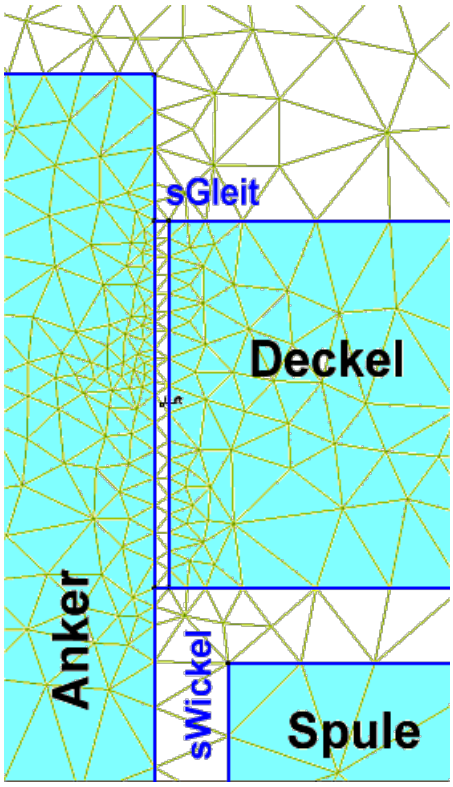
- Um den Vernetzungsgrad beeinflussen zu können, definieren wir dafür eine globale Variable ***nLuft*** im Parameterabschnitt des Lua-Scripts.
- Dazu notieren wir einen separaten Abschnitt für numerische Parameter vor den Entwurfsparametern:

```
-------------------------------------------------------------------
--------------
-- Numerische Parameter zur Steuerung der Simulationsgenauigkeit
--
-------------------------------------------------------------------
--------------
   nLuft=3;                      -- Zahl von Elementlagen in
kleinen Luftspalten
```

- Da eine feinere Vernetzung sehr schnell zu sehr langen Rechenzeiten führt, können wir über den Parameter ***nLuft*** sehr einfach einen günstigen Kompromiss zwischen Genauigkeit und Rechenzeit finden.
- Für jeden Luftspalt existiert ein Parameterwert (z.B. ***sGleit***). So kann man problemlos beim Setzen der Blockproperties die automatische Vernetzung durch die Angabe einer spaltabhängigen Maschengröße ersetzen (z.B. ***sGleit/nLuft***).

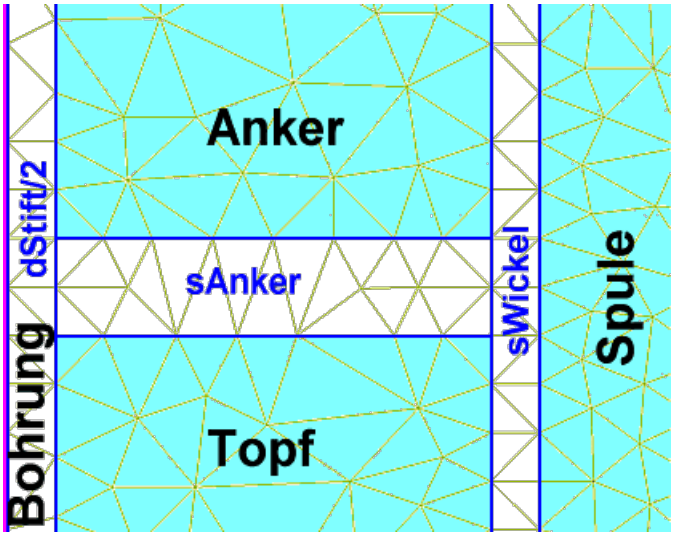
Führungsspalt

```
-- mi_setblockprop ("Luft",1,      0     ,"<None>",0,0,0);
ersetzen durch:
   mi_setblockprop ("Luft",0,sGleit/nLuft,"<None>",0,0,0);
```



Wickelspalt

```
-- mi_setblockprop ("Luft",1,        0      ,"<None>",0,0,0);
ersetzen durch:
   mi_setblockprop ("Luft",0,sWickel/nLuft,"<None>",0,0,0);
```



Bohrung

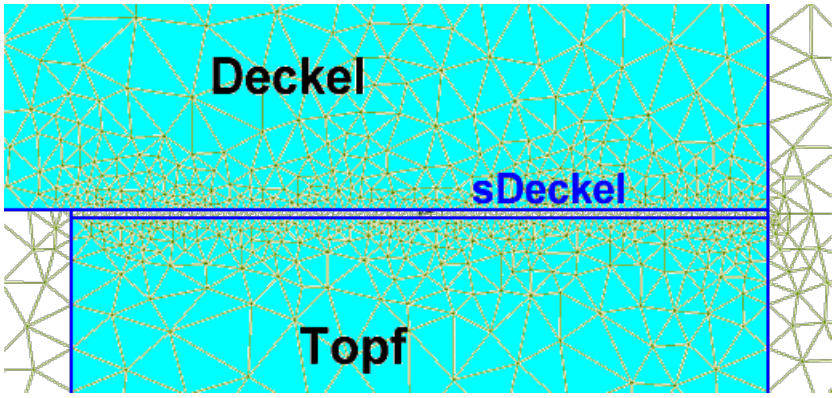
```
-- mi_setblockprop ("Luft",1,        0       ,"<None>",0,0,0);
ersetzen durch:
   mi_setblockprop ("Luft",0,dStift/2/nLuft,"<None>",0,0,0);
```

Arbeitsluftspalt

```
-- mi_setblockprop ("Luft",1,      0       ,"<None>",0,0,0);
ersetzen durch:
   mi_setblockprop ("Luft",0,sAnker/nLuft,"<None>",0,0,0);
```

Restspalt

```
-- mi_setblockprop ("Luft",1,       0       ,"<None>",0,0,0);
ersetzen durch:
   mi_setblockprop ("Luft",0,sDeckel/nLuft,"<None>",0,0,0);
```

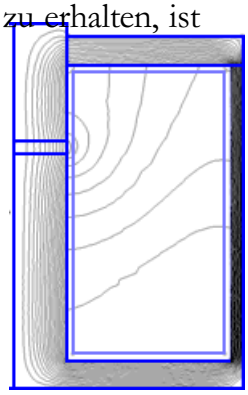


Feldberechnung und Darstellung

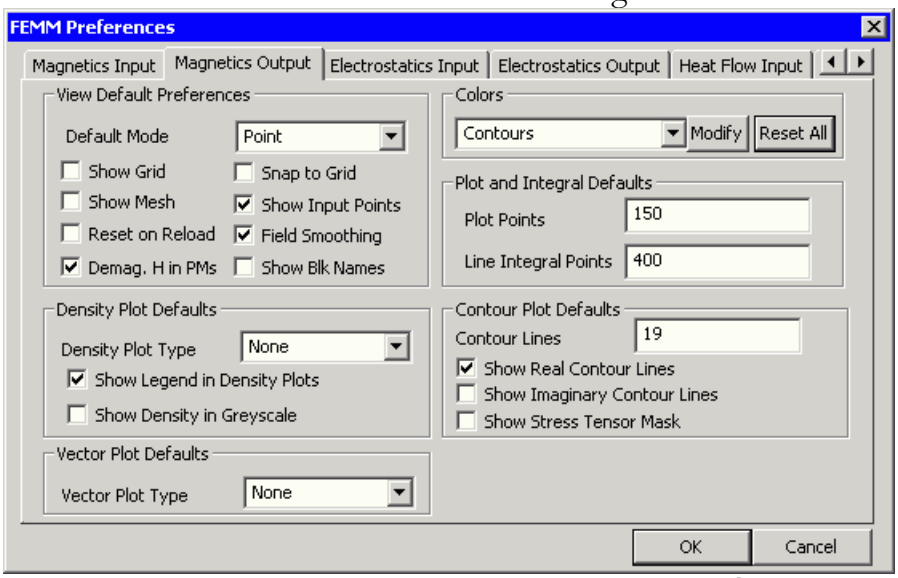
Nachdem die Vernetzung in den Luftspalten wunschgemäß über den Parameter ***nLuft*** gesteuert werden kann, können wir unser Script fortschreiben. Dazu ergänzen wir die Anweisungen für die Berechnung und das Öffnen des Ergebnisfensters:

```
-----------------------------------------------------------------
--------------
-- Vernetzung und Feld-Berechnung  (nach Speichern des Modells)
--
-----------------------------------------------------------------
--------------
   :
   mi_analyze(1);     -- Solverfenster 1=minimiert / 0=sichtbar
   mi_loadsolution(); -- öffnet Ergebnisfenster für Postprozess
-----------------------------------------------------------------
--------------
-- Beginn des Postprozesses
--
-----------------------------------------------------------------
--------------
   mo_zoom(0,-rKugel/3,rKugel/3,rKugel/3); -- Rechteckausschnitt
in der Kugel
   mo_showcontourplot (-1); -- Feldlinien normiert auf Bereich
(Preferences)
-- mo_showdensityplot(legend,gscale,upper_B,lower_B,type);
   mo_showdensityplot(   1   ,   0   ,   1   ,   0   ,"bmag");
```

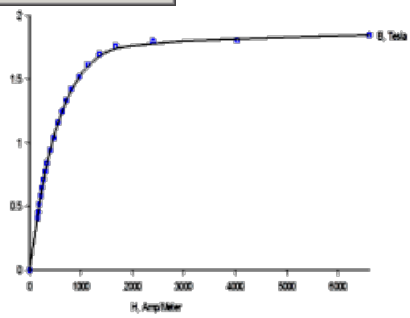
- Die gebräuchlichste Methode, um ein Gefühl für das berechnete magnetische Feld zu erhalten, ist die Darstellung der "Flusslinien". Diese repräsentieren die Flusswege, auf denen der magnetische Fluss innerhalb der abgebildeten Geometrie fließt. Je dichter diese Flusslinien liegen, desto größer ist die Flussdichte. Ausgehend vom berechneten Vektor-Potential ***A***, sind die Flusslinien einfach Isolinien von ***A***. Das gilt jedoch nur für den planaren Fall. In axial-symmetrischen Problemen sind es die Isolinien von **2·π·r·*A***.

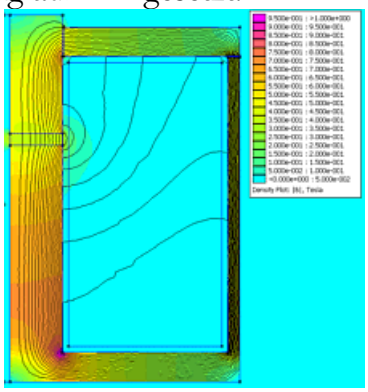


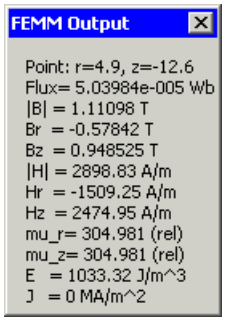
- Die Feldlinien-Darstellung könnte man im Lua-Script mit ***mo_showcontourplot*** individuell konfigurieren. Wir nutzen im Beispiel mit dem Parameter (-1) die unter ***Edit - Preferences - Magnetics Output*** eingestellte Default-Konfiguration. Die Default-Einstellung für die Feldlinien hat hier den Vorteil, dass die Grenzen automatisch auf den aktuellen Min- und Max-Wert des Feldes gesetzt werden:



- Der Wert der Flussdichte ***B*** zeigt, in welchem Maße das Eisenmaterial auf der BH-Kurve ausgesteuert wird. Standardmäßig ist im FEMM-Programm der Density-Plot ausgeschalten.
- Man könnte die Darstellung von **|*B*|** in den Default-Einstellungen für den Density Plot aktivieren. Wir werden aber die farbige Darstellung des B-Feldes über den Script-Befehl ***mo_showdensityplot*** steuern. Die Wahl geeigneter Grenz-Werte für die Darstellung ist unproblematisch, da der Wert von *B* auf Grund der BH-Kennlinie auf ca. **2 T** begrenzt ist. Im Script-Beispiel wurde der obere Grenzwert vorläufig auf **1 T** gesetzt:







- Man erkennt nach der Berechnung und Ergebnisdarstellung, dass der Raum außerhalb des Magnetkreises weitestgehend feldfrei ist.
- Von jedem Punkt des Modells erhält man nach Klick mit der linken Maustaste die wesentlichen magnetischen Größen in einem Ausgabefenster aufgelistet. Im Beispiel ist dies der Punkt im Bereich der höchsten Aussteuerung des Eisenmaterials. Die Größen werden jeweils mit ihrem Betrag und den Komponenten in die entsprechende Koordinaten-Richtung dargestellt.

← →

Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld_-_Vernetzung_und_Berechnung&oldid=25261“

Software: FEM - Tutorial - Magnetfeld - Kraft und Koppelfluss

Aus OptiYummy

↑

← →
Kraft und Koppelfluss (Einführung)

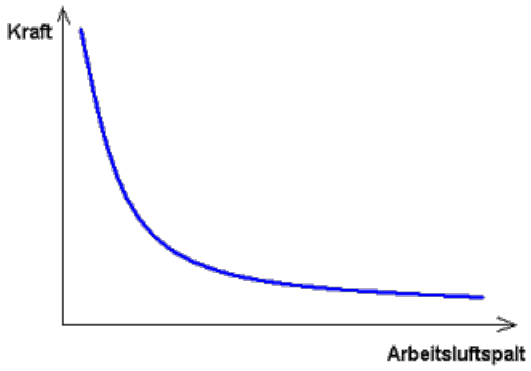
Inhaltsverzeichnis

1 Magnetkraft
- 1.1 Kraft auf Trennflächen
- 1.2 Gewichtetes "Stress-Tensor-Volumenintegral"
- 1.3 Maxwellscher Stress Tensor

2 Koppelfluss

Magnetkraft

Die im Arbeitsluftspalt ***s*** eines Elektromagneten erzeugte Kraft ist abhängig von der Stärke des Spulenstroms ***i***. Aber noch stärker hängt diese Kraft von der Größe des Arbeitsluftspalts ab. Bei konstantem Spulenstrom steigt die Kraft mit kleiner werdendem Luftspalt sehr stark an:



Eine statische FEM-Simulation ergibt bei einem vorgegebenen Spulenstrom nur den Kraftwert für den aktuell eingestellten Arbeitsluftspalt. Interessiert man sich für das Kennfeld ***F=f(s,i)***, so kann man dieses nur über eine Abfolge von FEM-Simulationen mit systematisch geänderten Werten für ***s*** und ***i*** ermitteln:

- Für diese systematische Variation der Parameterwerte ***s*** und ***i*** werden wir die Rastersuche das OptiY-Programms nutzen. Dieses Programm bietet uns auch die Möglichkeit, das gewonnene Kennfeld als 3D-Diagramm zu visualisieren. Wir können uns im Lua-Script also auf die Berechnung der Magnetkraft beschränken.
- Bevor wir das Lua-Script um die Berechnung der Magnetkraft erweitern, wollen wir das Prinzip der Kraftberechnung auf dem Niveau der grafischen Oberfläche erkunden. Voraussetzung ist die vorherige Abarbeitung unseres bisherigen Lua-Scripts.
- Wir betrachten hier 3 unterschiedliche Methoden für die Ermittlung der Magnetkraft.

Kraft auf Trennflächen

- Eine Abschätzung der Magnetkraft ermöglicht bereits die mittlere Flussdichte im Luftspalt zusammen mit der wirksamen Luftspaltfläche über die Maxwellsche Zugkraft-Formel:

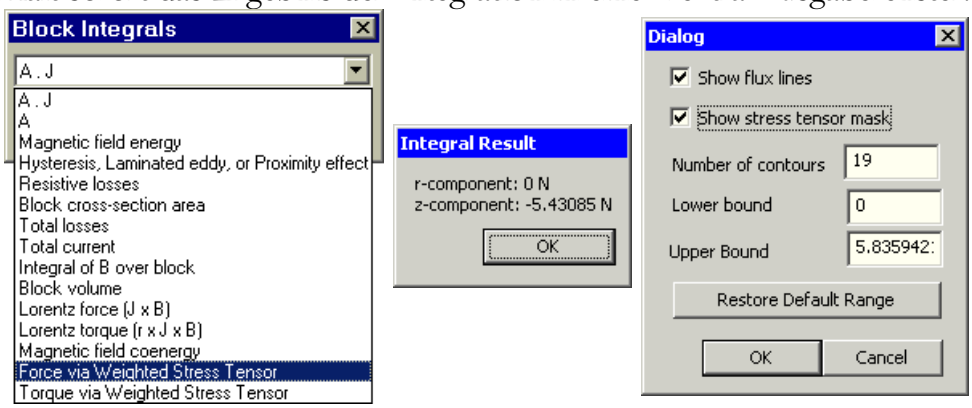
$$\mathbf{F = A \cdot B^2 / (2 \cdot \mu_0)}$$

- Bei einem Luftspalt **sAnker = 1 mm** ergibt sich mit dem Ankerquerschnitt **A = 77,7 mm²** und der Flussdichte **B ≈ 0,4 T** eine Kraft **F ≈ 5 N**.

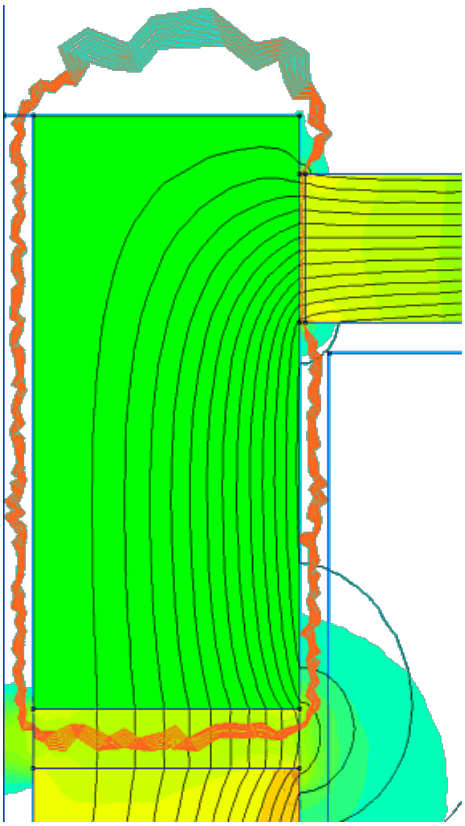
Gewichtetes "Stress-Tensor-Volumenintegral"

Im Vergleich zur fehlerträchtigen Berechnung nach obiger Formel, ist die im FEMM-Postprocessor implementierte Methode ein Kinderspiel. Unter der Voraussetzung, dass der Magnet-Anker überall an Luft grenzt, braucht man das zugehörige Geometrie-Objekt des Ankers nur markieren und die resultierenden Kräfte aufintegrieren lassen:

- Die Analysen werden in der geöffneten **.ans**-Datei vorgenommen.
- Man aktiviert den Areas-Operation-Modus .
- Man wählt die Region des Ankers durch Anklicken mit der linken Maustaste (Anker wird "grün").
- Nach Aufruf der Integration wählt man die Ermittlung der Kraft mittels "Gewichtetem Stress-Tensor" und erhält sofort das Ergebnis der Integration in einem extra Ausgabefenster:



- Der erhaltene Wert ist unter gleichen Bedingungen ca. 10% größer als der vorherige "Schätzwert", der nur die Kraftwirkung im Luftspalt selbst berücksichtigte.
- FEMM legt um die gewählte Region automatisch eine Stress-Tensor-Maske. Diese beschreibt den Weg für die Integration. Um die Maske darzustellen, muss man über den Contour-Plot-Dialog diese Anzeige aktivieren. Die roten Linien repräsentieren die Pfade für die Integration:



- Obiges Ergebnis für die Magnetkraft wurde mit ***nLuft=3*** zur Steuerung der Netzdichte in den Luftspalten ermittelt. Das Gesamtnetz umfasst damit ca. 12500 Knoten und erfordert schon eine merkliche Berechnungszeit.
- Für die systematische Abtastung des Kennfeldes müssen wir sehr viele Modellberechnungen durchführen. Deshalb ist es wichtig, eine hinreichend genaue Kraftberechnung mit einem möglichst groben Netz zu erhalten. Wir testen deshalb im Folgenden für ***sAnker=1 mm*** und ***iSpule=1 A*** den Einfluss der Vernetzungsfeinheit für ***nLuft=1..5***, indem wir ***nLuft*** im Lua-Script manuell ändern und das Script erneut abarbeiten lassen. Dabei schätzen wir auch die Abarbeitungszeit ***t*** für das Script. Dafür ergeben sich beispielhaft folgende Werte:

nLuft	F/N	t/s
1	5,30	1
2	5,37	3
3	5,43	12
4	5,44	21
5	5,44	30

Man kann aus diesen Ergebnissen zwei wesentliche Erkenntnisse ableiten:

1. Die Kraftberechnung reagiert relativ unempfindlich auf die Vernetzung in den Luftspalten (Fehler max. 2%). Da das Modell auch bei exakter Berechnung infolge der Unbestimmtheit vieler Parameter (insbesondere Material und Restluftspalte) um einige Prozent ungenauer ist, spielt dieser Vernetzungsfehler nur eine untergeordnete Rolle.
2. Die Berechnungszeit für das Modell steigt dramatisch. Man benötigt mindestens die 10-fache Zeit, wenn man Berechnungsungenauigkeiten im Promille-Bereich anstrebt (z.B. *nLuft=3*). Für die einzelne Modell-Berechnung ist das im Beispiel noch unkritisch. Da für das systematische Abrastern des gewünschten Kennfeldes mindestens 100 bis 400 Modell-Berechnungen erforderlich sind, ist der Unterschied jedoch gewaltig!

Wir setzen deshalb für die weitere Modell-Nutzung im Lua-Script ***nLuft=1***.

Maxwellscher Stress Tensor

Leider ist die soeben beschriebene Methode der Magnetkraft-Bestimmung nicht in allen FEM-Systemen implementiert. Deshalb ist man häufig darauf angewiesen, manuell den Integrationsweg für die Kraftberechnung vorzugeben. Darin liegt jedoch gerade für unerfahrene Nutzer eine nicht zu unterschätzende Fehlerquelle:

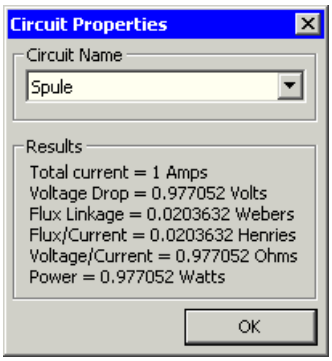
- Ursächlich sind numerische Probleme bei der im Hintergrund erforderlichen Ermittlung von ***B*** und ***H*** mittels numerischer Differentiation aus dem Primärergebnis "Vektorpotential ***A***".
- Besonders falsch werden ***B*** und ***H*** an der Grenze zwischen Materialien unterschiedlicher Permeabilität berechnet (z.B. Luft / Eisen).

Daraus resultieren folgende Regeln:

1. Der Integrationsweg darf nie direkt an der Grenze zwischen verschiedenen Materialien verlaufen.
2. Es sollten mindestens 2 Elemente zwischen dem Integrationspfad und der Grenze des interessierenden Objekts (hier der Anker) liegen.
3. An den kritischen Stellen rings um das interessierende Objekt sollte man eine möglichst feine Vernetzung verwenden, um "stochastische" Sprünge bei der numerischen Differentiation des Vektorpotentials zu minimieren.

- Wenn man unser Beispiel betrachtet, so sieht man, dass mit der aktuellen Vernetzung obige Regeln nicht befolgt werden können.
- Auf Grund des Aufwandes soll diese allgemeine Methode in dieser Übung nicht praktisch erprobt werden!

Koppelfluss



- Für die Berechnung des Koppelflusses **Ψ** (auch *Verkettungsfluss* genannt) werden die magnetischen Flüsse über alle Windungen **w** aufsummiert. Handelt es sich um eine einlagige Zylinderspule im homogenen Feld, so gilt für diesen Spezialfall **Ψ=w·Φ**.
- Man benötigt den Koppelfluss **Ψ** für die Berechnung der in der Spule infolge von Flussänderungen induzierten Spannung **v_ind=dΨ/dt**.
- Daraus resultiert auch die Bezeichnung ***Koppelfluss***, weil er als integrale Größe die Wechselwirkung (die Kopplung mit dem Magnetfeld) auf der elektrischen Seite des elektro-mechanischen Wandlers repräsentiert.
- Der Wert des Koppelflusses (Flux Linkage) einer Spule wird im magnetischen Postprozessor des FEMM-Programms als Bestandteil der ***Circuit Properties*** berechnet. Der Einfluss der groben Vernetzung mit ***nLuft=1*** beträgt wie bei der Kraftberechnung ca. 2%.

← →

Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld_-_Kraft_und_Koppelfluss&oldid=25263“

Software: FEM - Tutorial - Magnetfeld - Kraft und Koppelfluss - Script

Aus OptiYummy

↑

← →

Kraft und Koppelfluss (Lua-Script)

Nachdem wir nun das relativ einfache Verfahren (gewichtetes "Stress-Tensor-Volumenintegral") zur Ermittlung der Magnetkraft kennen gelernt haben, werden wir es in unser Lua-Script einfügen:

Magnetkraft

Wir öffnen zu Beginn des Postprozesses die Lua-Konsole mittels "showconsole();":

```
-------------------------------------------------------------------
--------------
-- Beginn des Postprozesses
--
-------------------------------------------------------------------
--------------
-- Kontrollausgaben auf der Lua-Konsole:
   showconsole();                 -- Lua-Konsole öffnen, falls
nicht offen
   mo_zoom(0,-rKugel/3,rKugel/3,rKugel/3); -- Rechteck um Magnet
in Kugelsphäre
   mo_showcontourplot (-1); -- Feldlinien normiert auf Bereich
(lt.Preferences)
-- mo_showdensityplot(legend,gscale,upper_B,lower_B,type);
   mo_showdensityplot(   1  ,   0  ,   1   ,   0   ,"bmag");
   mo_seteditmode("areas");
   mo_groupselectblock(1);        -- Auswahl Anker-Block ->
Group=1 bei Def.!
   Fz_Anker=mo_blockintegral(19); -- (19): z-Komponente der
statischen Kraft
   print ("F=", Fz_Anker, "[N]")  -- Kraft in Outputfeld der LUA-
Konsole
```

Die Berechnung des Blockintegrals bezieht sich immer auf den aktuell ausgewählten Block:

- Einen Block wählt man am einfachsten mit ***groupselectblock(Gruppennummer)***.
- Bereiche des Modells, mit welchen später gemeinsame Operationen ausgeführt werden, sollten deshalb bereits bei der Definition einer gemeinsamen Gruppen-Nummer zugeordnet werden.
- Neben der Berechnung des Blockintegrals sind solche Operationen z.B. auch geometrische Transformation (Verschieben, Drehen).

- Im Beispiel muss deshalb im LUA-Script bei der Definition des Ankers in den Block-Properties statt der Gruppe **0** die Gruppe **1** verwendet werden:

```
-----------------------------------------------------------------
--------------
                                  --
*******************************************
   mi_seteditmode ("nodes");   -- A=Eisenkreis (Anker)
************************
                                  --
*******************************************
   :
   mi_setblockprop ("Stahl",1,0,"<None>",0,1,0);
mi_clearselected();
```

Anscheinend gibt es kein LUA-Kommando, um die Linien der Stress-Tensor-Maske sichtbar zu machen. Man muss das Einblenden auf der grafischen Oberfläche veranlassen:

- Man kann unter ***View > Contour Plot*** die Anzeige der **Stress Tensor Mask** aktivieren!
- Das Einblenden des Integrationspfades erfolgt erst mit einiger Verzögerung bei einer Aktualisierung des Grafikfensters.

Koppelfluss

Zu Testzwecken sollte man die berechneten Ergebnisse auf der LUA-Konsole darstellen:

```
-----------------------------------------------------------------
--------------
-- Beginn des Postprozesses
--
-----------------------------------------------------------------
--------------
   :
   iSpule, vOhm, Psi = mo_getcircuitproperties("Spule"); --
statische Größen
   print ("i=", iSpule   , "[A]");-- Strom
   print ("v=", vOhm     , "[V]");-- Spannungsabfall
   print ("R=",vOhm/iSpule,"[Ohm]");-- Drahtwiderstand
   print ("L=",Psi/iSpule ,"[H]");-- Induktivität
   print ("Psi=", Psi    ,"[Wb]");-- Koppelfluss
```

Den Koppelfluss Ψ der Spule erhält man als Bestandteil der Circuit-Properties nur gemeinsam mit dem statischen Spannungsabfall am ohmschen Widerstand des Spulendrahtes und dem aktuellen Strom:

- Als Variable für den Strom wurde der Parameter ***iSpule*** verwendet, da im statischen Fall der Strom konstant bleibt.
- ***vOhm*** und ***Psi*** werden im LUA-Script automatisch als Variablen angelegt (bei erstmaliger Benutzung).

- Aus Strom und Spannung kann man den ohmschen Widerstand ***R*** des Spulendrahtes berechnen.
- Analog erhält man aus dem Koppelfluss ***Ψ=f(s,i)*** und dem Strom ***ispule*** die aktuelle Induktivität ***L=f(s,i)*** .

← →

Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld_-_Kraft_und_Koppelfluss_-_Script&oldid=15335“

Software: FEM - Tutorial - Magnetfeld - Einbindung in OptiY-Workflow

Aus OptiYummy

↑

← →

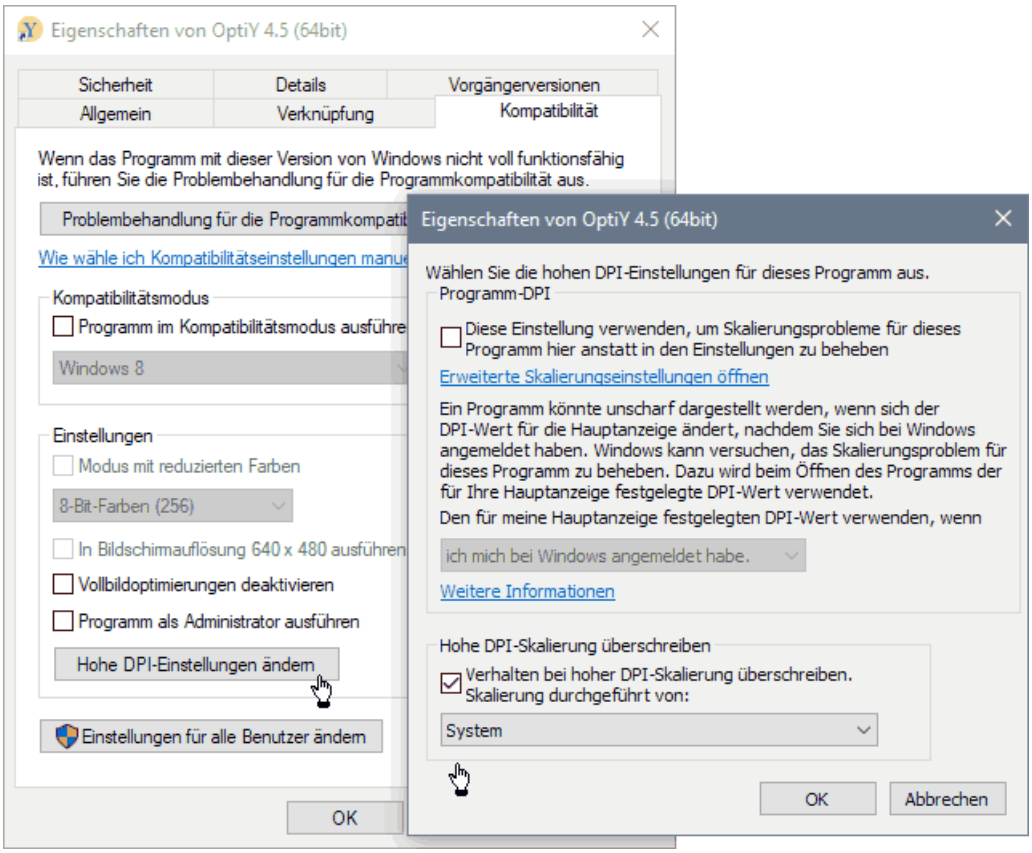
Einbindung in OptiY-Workflow

Die Ermittlung einer Kraft-Weg-Kennlinie wird im FEMM-Beispiel **https://www.femm.info/wiki/CoilGun** behandelt. Dieses **Beispiel** sollte man sich einmal anschauen. Die zugehörige **Modell-Datei** und **Lua-Scriptdatei** kann man von dort laden und erspart sich damit das Eintippen:

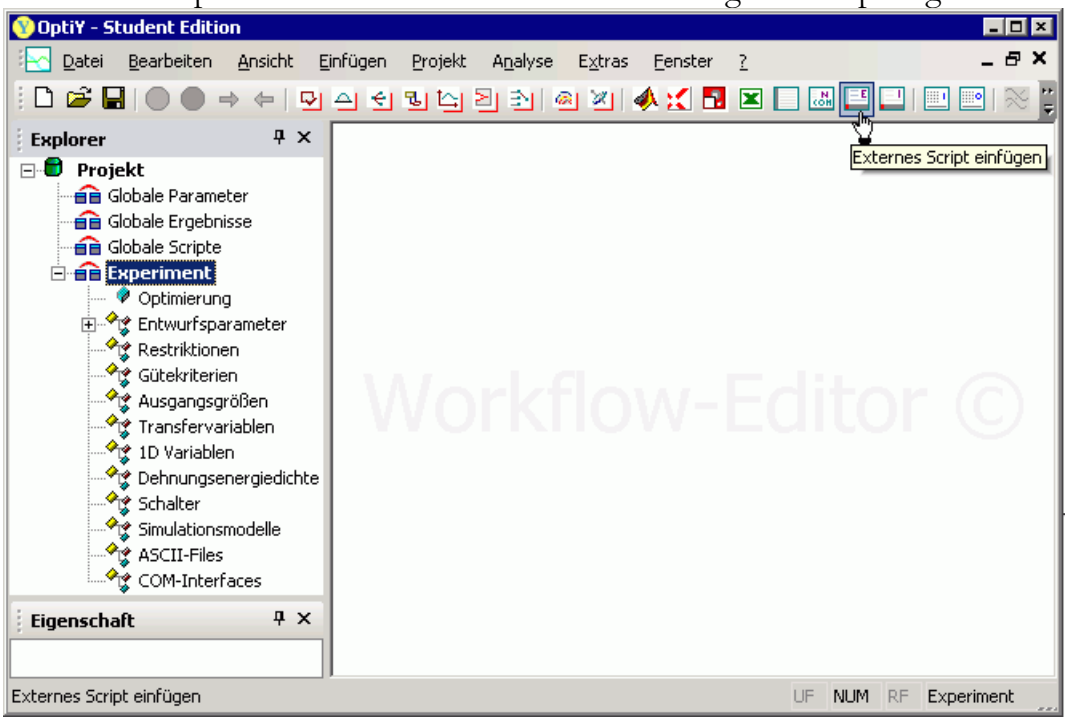
- Kennlinien oder auch höherdimensionale Kennfelder werden durch systematisches Abtasten der abzubildenden Funktion gebildet.
- Jeder Abtastpunkt entspricht einer vollständigen FEM-Berechnung, meist einschließlich des kompletten Pre-Prozesses.
- Innerhalb von Script-Dateien realisiert man das systematische Abtasten, indem man die Funktionsvariablen in (geschachtelten) Schleifen-Anweisungen zwischen vorgegebenen Grenzen variiert.
- Im "coilgun"-Beispiel wird ein Block des Modells Schritt für Schritt verschoben.
- Nach der Berechnung steht man dann noch vor dem Problem, die Ergebnisse in eine Form zu konvertieren, welche für die weitere Verarbeitung (z.B. zur Dynamik-Simulation) nutzbar ist.

Wir gehen in dieser Übung einen einfacheren Weg, indem wir die Möglichkeiten eines Optimierungstools zur Rastersuche verwenden. Die erhaltenen Ergebnisse stellen wir anschaulich als 3D-Diagramme dar. Als Optimierungstool nutzen wir **OptiY**. Die *Trial Edition* ist ausreichend, welche unter **www.optiy.eu/Download.html** bereitgestellt wird. Im Rahmen der Übung wird die Konfiguration der erforderlichen Optimierungsexperimente für den Einsteiger beschrieben. Vorkenntnisse zum Programm OptiY sind also nicht erforderlich:

- ***Hinweis zur Nutzung hochauflösender Displays*:**
  - Viele Programme haben Probleme mit extrem hochauflösenden Bildschirmen, wie sie verstärkt in modernen Tablett-Computern zum Einsatz kommen. In OptiY äußert sich dies in teilweise unlesbar kleinen Diagramm-Legenden oder sich überlappenden Listen-Elementen.
  - Abhilfe schaffen hier die Kompatibilitätseinstellungen (Kontextmenü des OptiY-Icon auf dem Desktop) "Hohe DPI-Einstellungen ändern":



- Es ist der Modus "Verhalten bei hoher DPI-Skalierung überschreiben" zu aktivieren und "Skalierung durchgeführt von System" einzustellen. Danach sollten alle Programm-Komponente in aufeinander abgestimmter Skalierung dargestellt werden.
- Nach der wahrscheinlich problemlosen Installation kann das Programm OptiY gestartet werden:



Mit dem ***Workflow-Editor*** (Menü ***Einfügen***) von OptiY fügen wir unser Lua-Script des FEMM-Modells als ***Externes Script-Modell*** an zentraler Stelle in den Workflow des Experiments ein:

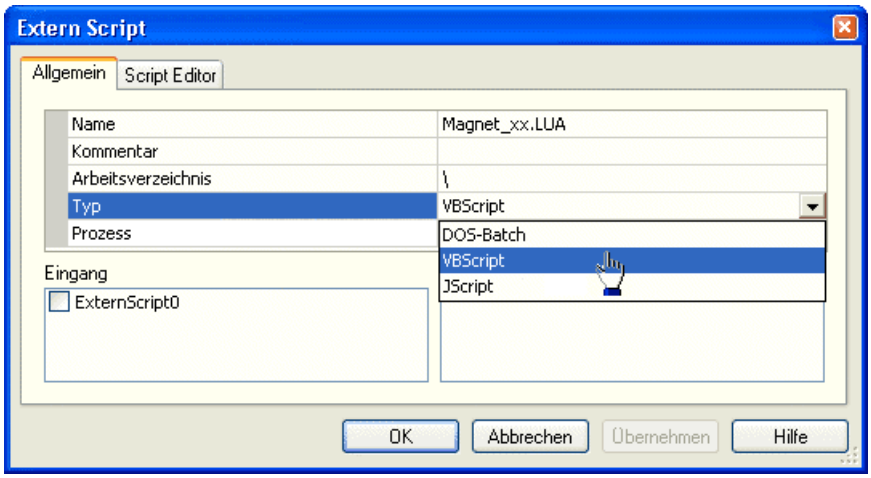
- Über dieses Script muss das FEMM-Programm gestartet werden, welches dann seinerseits das LUA-Script ausführt.
- Die Ausführung des Lua-Scripts im FEMM kann man über das vom FEMM-Programm bereitgestellte ActiveX-Interface veranlassen.
- Ein Doppelklick auf das platzierte Symbol öffnet den erforderlichen Konfigurationsdialog.

Im OptiY werden mehrere Script-Sprachen unterstützt. Beispielhaft soll dies im Folgenden für zwei Script-Typen demonstriert werden:

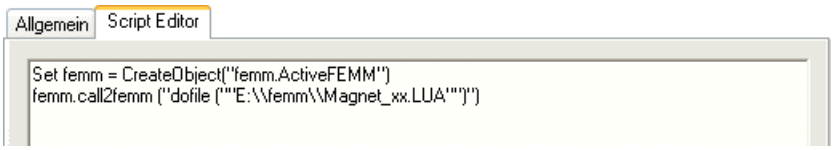
---

1. Visual Basic Script *(nur zur Information - in dieser Übung nicht anwenden!)*:

Zur Aktivierung der Schnittstelle und Aufruf des FEMM-Programms werden wir im Rahmen dieser Übung *VBScript* **nicht** als Scriptsprache benutzen, weil über die COM-Schnittstelle keine Parallelisierung der Berechnung möglich ist.



- Im Script-Editor muss der komplette Pfad der Lua-Datei angegeben werden, wobei unbedingt doppelte Backslash \\ zu verwenden sind:

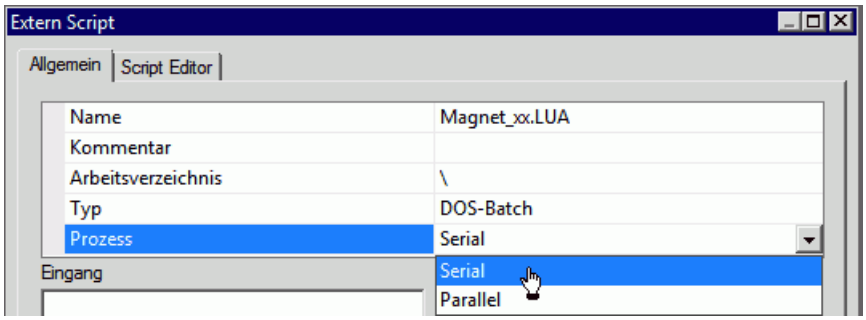



```
Set femm = CreateObject("femm.ActiveFEMM")
femm.call2femm ("dofile (""E:\\femm\\Magnet_xx.LUA"")")
```

- Für dieses im Script-Editor von OptiY eingetragene Script generiert OptiY bei der Experiment-Durchführung eine Script-Datei und startet sie.
- ***Wichtig***: Damit der FEMM-Aufruf mittels VBScript funktioniert, muss das FEMM-Programm richtig als COM-Objekt registriert sein. Dazu muss FEMM zuvor mindestens einmal mit Administrator-Rechten gestartet werden.

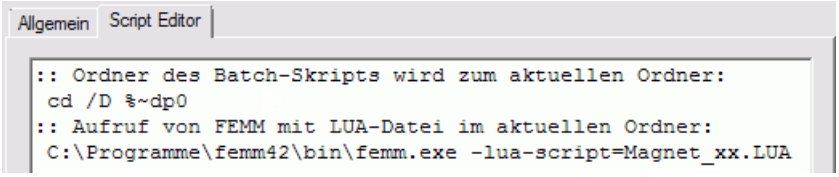
2. DOS-Batch Script:

Im fast in Vergessenheit geratenem Betriebssystem **MS-DOS** konnte man mit **BAT-Dateien** automatisiert eine Folge von **MS-DOS-Kommandos** ausführen. Das funktioniert auch noch unter Microsoft Windows und ermöglicht die parallele Abarbeitung der Modellberechnungen:



Wichtig: vorläufig konfigurieren wir den Prozess als seriell, weil zur Realisierung einer parallelen Abarbeitung der Batch-Scripte noch weitere Maßnahmen erforderlich sind.

- Im DOS-Batch-Script nur einfache Backslash \ im Pfadbezeichner zu verwenden.
- Ist das FEMM-Programmverzeichnis nicht Bestandteil des Suchpfades (PATH), so kann die *femm.exe* nur mit der vollständigen Pfad-Angabe aufgerufen werden.
- Der komplette Pfad der Lua-Datei muss nicht angegeben werden, wenn sich die Lua-Datei im aktuellen Ordner befindet. Dies realisieren wir mit **cd**-Kommando (Change Directory):




```
:: Ordner des Batch-Skripts wird zum aktuellen Ordner:
cd /D %~dp0
:: Aufruf von FEMM mit Lua-Datei im aktuellen Ordner:
C:\Programme\femm42\bin\femm.exe -lua-script=Magnet_xx.LUA -
windowhide
```

- ***Hinweis:*** Im PC-Pool findet man FEMM im Standardpfad der Installation: C:\femm42 , es **entfällt** also der Unterordner **Programme** im Pfad.
- ***Achtung***:
  Enthält ein Dateiname Leer- oder Sonderzeichen, so muss er in Anführungsstrichen notiert werden, z.B.:

```
"c:\Program Files (x86)\femm42\bin\femm.exe" -lua-
script=Magnet_xx.LUA -windowhide
```

- Der Parameter "**-windowhide**" vermeidet während der durch OptiY veranlassten FEMM-Berechnungen das periodische "Fenster-Geflacker" durch eine Unterdrückung des FEMM-Programmfensters.

Hinweise zum LUA-Script-Anpassung:

- Wird die Abarbeitung des LUA-Scripts über *DOS-Batch* veranlasst, so endet das FEMM-Programm nicht selbstständig. Das LUA-Script muss deshalb abschließend um den *exit*-Befehl erweitert werden:

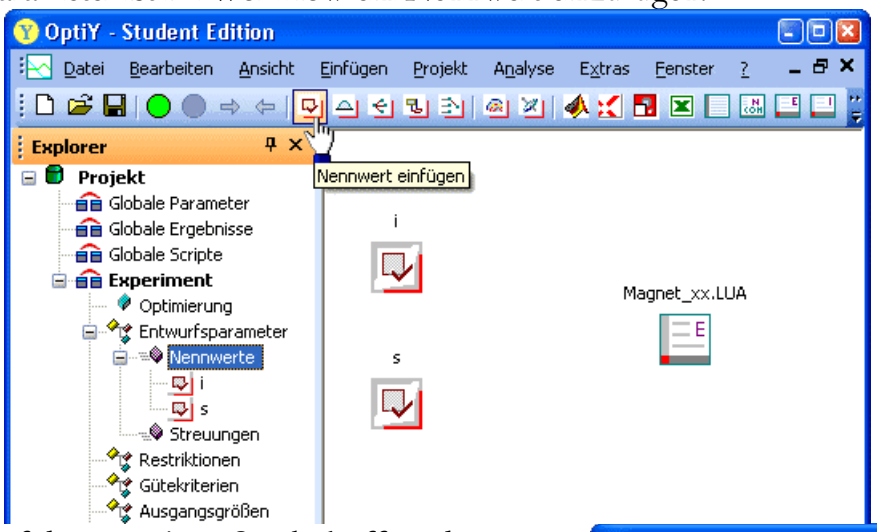
```
   exit();
```

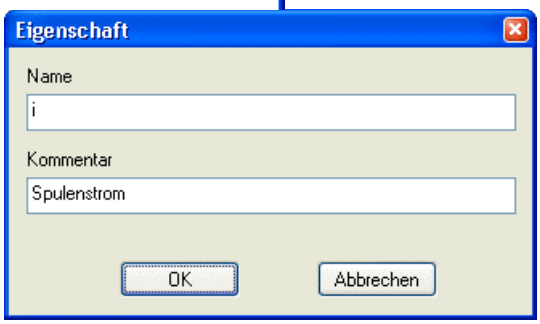
- Möchte man das gleiche LUA-Script ohne OptiY autonom im FEMM benutzen, so muss man natürlich diesen *exit*-Befehl "auskommentieren", damit das FEMM nach Script-Abarbeitung nicht sofort endet!
- Um die FEMM-Berechnung während der Optimierung wirklich ohne Fenster-Geflacker auszuführen, muss man zusätzlich noch das Öffnen der LUA-Console abschalten:

```
-- showconsole(); -- LUA-Console öffnen, falls Script als Datei
ausgeführt
   hideconsole(); -- LUA-Console verstecken
```

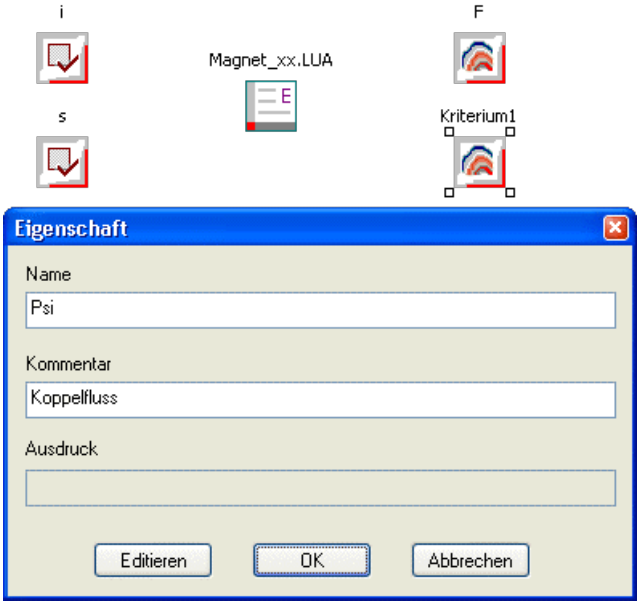
Es ist möglich, im OptiY-Workflow die Verbindung mit allen Eingabe-Parametern des Modells herzustellen. Für die Ermittlung der Kennfelder beschränken wir uns auf die zwei zu variierenden Parameter ***Spulenstrom i*** und ***Arbeitsluftspalt s***.

- Für jeden dieser Parameter ist im Workflow ein Nennwert einzufügen:

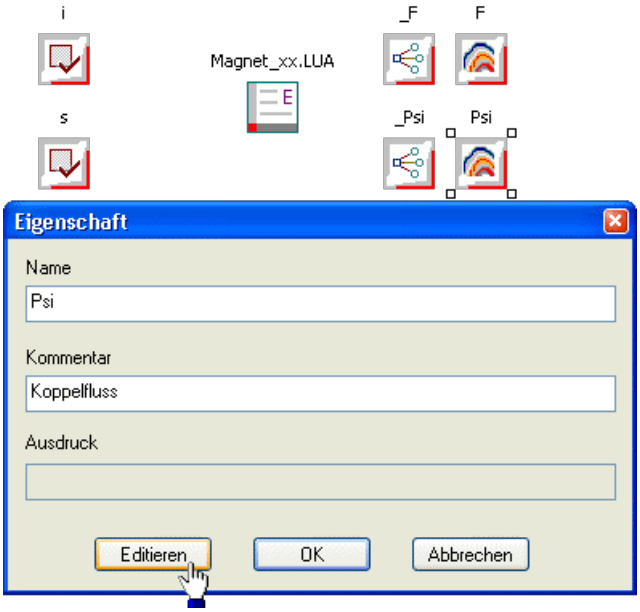




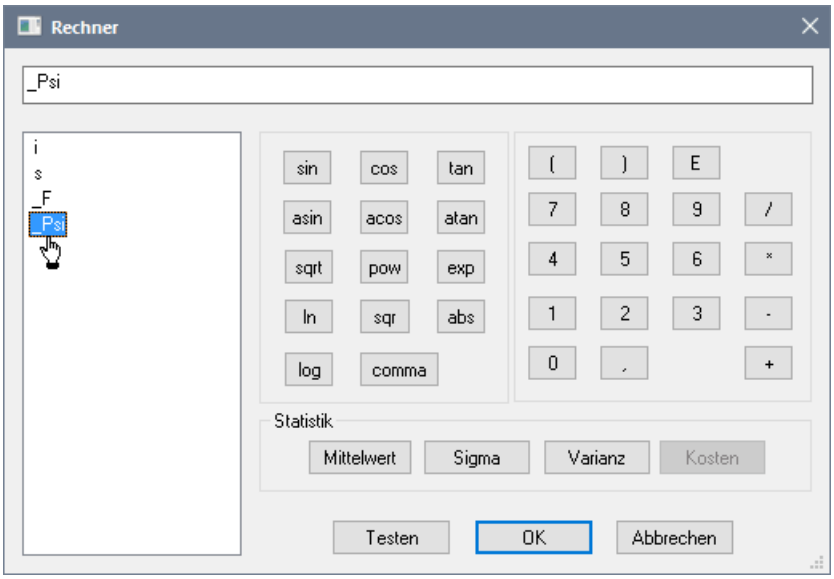
- Ein Doppelklick auf das generierte Symbol öffnet den Konfigurationsdialog für den Nennwert. Hier sollte man einen sinnvollen Namen und Kommentar vergeben. Der Name muss nicht, sollte aber mit dem Parameter-Namen im Modell übereinstimmen.
- Mit den noch fehlenden Verbindungen zwischen dem FEMM-Modell und den beiden im Workflow definierten Nennwerten beschäftigen wir uns später!
- Im OptiY-Workflow müssen die berechneten Werte für die ***Magnetkraft F*** und den ***Koppelfluss Psi*** eingefügt werden. Nur dann kann man die beiden Kennfelder als 3D-Diagramm darstellen. Wir definieren beide Größen als Gütekriterien (***Einfügen > Gütekriterien***):



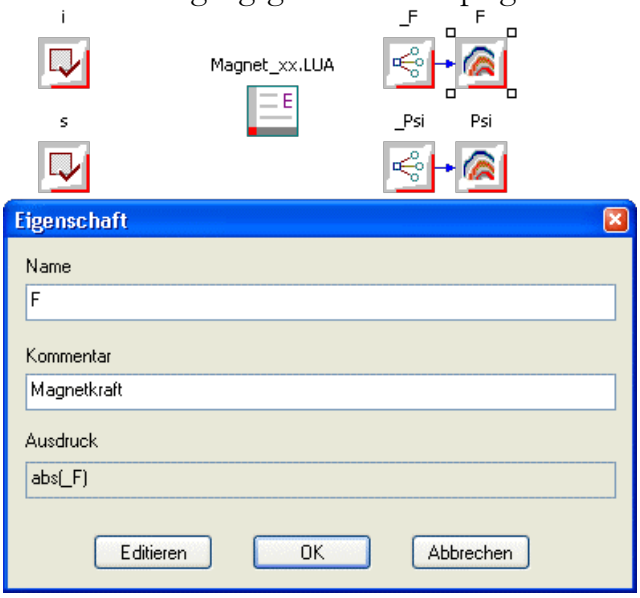
- Den Gütekriterien kann man nicht direkt die Ergebnisgrößen eines Modells zuweisen. Deshalb benötigt man zusätzliche Ausgangsgrößen, welche die Ergebnisse des Modells repräsentieren (***Einfügen > Ausgangsgrößen***). Da gleiche Namen für unterschiedliche Variablen im Workflow nicht zulässig sind, stellen wir bei den Ausgangsgrößen einen Unterstrich voran, welche den gleichen Wert besitzen, wie das zugeordnete Gütekriterium:



- Die Verknüpfung zwischen Ausgangsgröße und Güterkriterium stellt man her, indem man den "Ausdruck" des Gütekriteriums mit dem Wert der Ausgangsgröße belegt. Nach Doppelklick auf das Kriterium aktiviert man mit **Editieren** eine Art "Taschenrechner". Mit diesem klickt man die erforderliche Formel zusammen (Doppelklick zum Einfügen einer der aufgelisteten Workflow-Größen):



- Für die Kraft ***F*** ist es günstig, aus Gründen der Anschaulichkeit den Absolutwert ***abs(_F)*** zu verwenden, da im FEMM-Modell negative Kraftwerte berechnet werden. Die Verknüpfung zwischen den Gütekriterien und den Ausgangsgrößen widerspiegelt sich danach im Workflow:



Achtung: Den aktuellen Zustand des OptiY-Versuchsstandes speichern wir als Datei **Magnet_xx.opy** mit **xx**=Teilnehmer-Nummer

Nun fehlt nur noch die Ankopplung der Nennwerte und der Ausgangsgrößen an das FEMM-Modell. Diesem Problem widmen wir uns auf der nächsten Seite. Dazu verlassen wir OptiY und arbeiten wieder mit dem FEMM-Programm samt zugehörigem Script-Editor.

← →

Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld_-_Einbindung_in_OptiY-Workflow&oldid=25739“

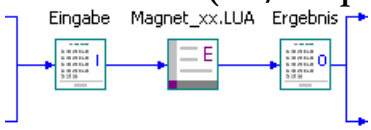
Software: FEM - Tutorial - Magnetfeld - OptiY-Datenschnittstellen

Aus OptiYummy

↑

← →

Datenschnittstellen (In-/Outputfiles)



Inhaltsverzeichnis

- 1 Datenfluss über Input-/Output-Dateien
- 2 Input-Datei für Vorgabe-Parameter
- 3 Output-Datei für Simulationsergebnisse
- 4 Verifizierung des Modells
- 5 Einbindung in den OptiY-Workflow

Datenfluss über Input-/Output-Dateien

Es existieren unterschiedliche Varianten, wie ein Optimierungstool die Parameterwerte (Input) eines Modells modifizieren und auf die damit berechneten Ergebniswerte (Output) zugreifen kann. Eine sehr allgemeine und übersichtliche Methode ist die Benutzung von ASCII-Textdateien für den Input/Output eines Modells:

- Das Optimierungstool schreibt die aktuellen Parameterwerte an die dafür vorgesehenen Stellen in eine Textdatei.
- Aus dieser Textdatei liest das Modell seine Werte für eine Simulationsrechnung.
- Die berechneten Ergebnisgrößen werden vom Modell nach der Simulationsrechnung wieder an definierte Stellen in eine andere Textdatei geschrieben.
- Das Optimierungstool liest diese Ergebniswerte aus der Textdatei und verarbeitet sie je nach Experimentierziel. In unserem Beispiel der Raster-Suche würden die Ergebnisgrößen nur als Funktionsflächen in 3D-Diagrammen dargestellt.

Es ist meist besser, eine strikte Trennung zwischen Parametern (Input-File), Modell (LUA-Script) und Ergebnisgrößen (Output-File) vorzunehmen. In diesem Sinne soll das LUA-Script nun umgestaltet werden.

Input-Datei für Vorgabe-Parameter

Bisher sind die Input-Parameter Bestandteil des Modell-Scripts. Die Input-Parameter sollen als separate

ASCII-Textdatei **Eingabe_xx.txt** (mit **xx**=Teilnehmer-Nummer) gestaltet werden. Die Datenwerte in dieser Textdatei müssen so strukturiert werden, dass sowohl das Optimierungsprogramm als auch das Modellscript problemlos darauf zugreifen können:

Wichtung:
Jeder Teilnehmer der FEM-Lehrveranstaltung benutzt ab sofort einen individuellen Strom von **i=0,2*xx* A** (mit **xx**=Teilnehmer-Nummer):

```
dMagnet
40
hTopf
27.5
dAnker
10
lAnker
10
hDeckel
2.5
dStift
1
dWand
1
wSpule
400
dLack
0.06
sWickel
0.5
sAnker
1
sDeckel
0.01
sGleit
0.1
iSpule
0.2xx
```

```
:
-------------------------------------------------------------------
--------------
-- Konstruktive, technologische und Betriebs-Parameter
(Standardwerte)       --
-------------------------------------------------------------------
--------------
   xx=00;                        -- Teilnehmer-Nr. 00 bis 30
-- dMagnet=40;                   -- Topf-Durchmesser [mm]
-- hTopf=27.5;                   -- Topf-Höhe [mm]
-- dAnker=10;                    -- Anker-Durchmesser [mm]
-- lAnker=10;                    -- Anker-Länge [mm]
-- hDeckel=2.5;                  -- Höhe des Deckels bzw.
```

```
Topfbodens [mm]
-- dStift=1;                        -- Durchmesser der Bohrung [mm]
-- dWand=1;                         -- Wandstärke des Topfes [mm]
-- wSpule=400;                      -- Windungen in der Spule
-- dLack=0.06;                      -- Gesamtdicke der
Drahtbeschichtung [mm]
-- sWickel=0.5;                     -- Spalt zw. Spulendraht und
Eisen [mm]
-- sAnker=1;                        -- Arbeitsluftspalt [mm]
-- sDeckel=0.01;                    -- Restspalt zw. Deckel und Topf
[mm]
-- sGleit=0.1;                      -- Führungsluftspalt für Anker
[mm]
-- iSpule=1;                        -- Spulenstrom [A]
-- Parameter aus Textdatei einlesen mit Fehlertest
   in = openfile(mydir .. "Eingabe_xx.txt","r"); -- Datei öffnen
zum Lesen
   if in then -- wenn Datei geöffnet, dann Werte einlesen
   read(in, "*l");    dMagnet = read(in, "*l", "*n");
   read(in, "*l");    hTopf = read(in, "*l", "*n");
   read(in, "*l");    dAnker = read(in, "*l", "*n");
   read(in, "*l");    lAnker = read(in, "*l", "*n");
   read(in, "*l");    hDeckel = read(in, "*l", "*n");
   read(in, "*l");    dStift = read(in, "*l", "*n");
   read(in, "*l");    dWand = read(in, "*l", "*n");
   read(in, "*l");    wSpule = read(in, "*l", "*n");
   read(in, "*l");    dLack = read(in, "*l", "*n");
   read(in, "*l");    sWickel = read(in, "*l", "*n");
   read(in, "*l");    sAnker = read(in, "*l", "*n");
   read(in, "*l");    sDeckel = read(in, "*l", "*n");
   read(in, "*l");    sGleit = read(in, "*l", "*n");
   read(in, "*l");    iSpule = read(in, "*l", "*n");
   else print("Could not open Eingabe_xx.txt"); end;
   closefile(in);
-------------------------------------------------------------------
--------------
-- Materialien (Luft, Kupfer, Eisen) --
-------------------------------------------------------------------
--------------
   :
```

Die Input-Zeilen mit den Parameter-Namen werden jeweils überlesen. Der auf der nächsten Zeile folgende Wert wird dem im Script definierten Parameter-Bezeichner zugewiesen. Die ursprünglichen Parameterwerte sollte man mit Standardwerten versehen und auskommentieren. Damit weiß man auch später noch, um welche Parameter es sich handelt.

Achtung: Es ist unbedingt zu überprüfen, ob nach der Verwendung einer Input-Datei mit den gleichen Parameter-Werten wirklich die gleichen Ergebnisse berechnet werden. Dazu muss man den Stromwert kurzzeitig wieder auf 1 A setzen!

Output-Datei für Simulationsergebnisse

Die für die Bewertung der Simulation relevanten Ergebnisgrößen müssen ebenfalls in eine Textdatei geschrieben werden. In unserem Beispiel sind das die Magnetkraft und der Koppelfluss. Die erforderlichen File-Operationen werden an das Ende des LUA-Scripts platziert:

```
    :
-- Output-File mit Ergebniswerten erzeugen:
   out = openfile(mydir .. "Ergebnis_xx.txt", "w");
   write (out,"F="  , Fz_Anker, "\n");
   write (out,"Psi=", Psi     , "\n");
   closefile(out);
-- Dateien loeschen:
   remove (mydir .. "Magnet_xx.fem"); -- Loeschen der Modelldatei
   remove (mydir .. "Magnet_xx.ans"); -- Loeschen der Grafikdatei
-- exit();  -- FEMM-Programm beenden
```

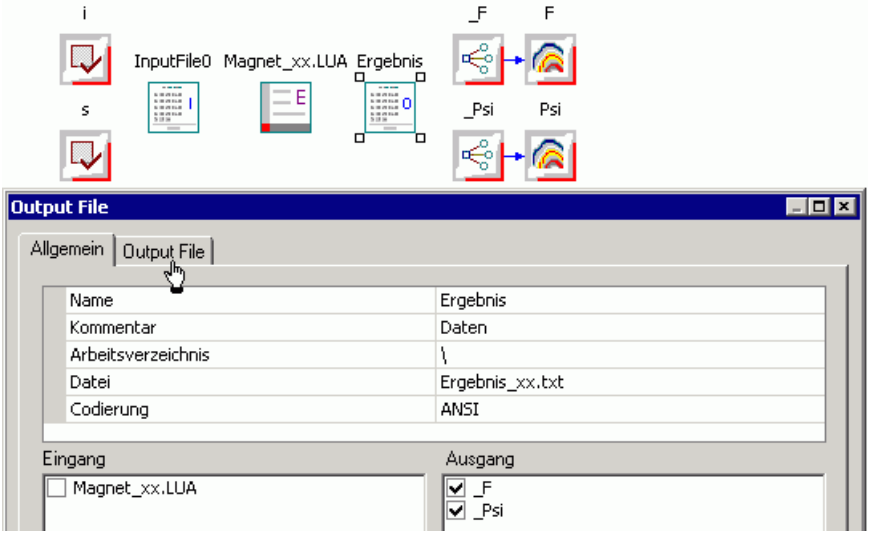
Verifizierung des Modells

Nach diesen Änderungen müssen nach Ausführung des kompletten LUA-Scripts im FEMM in der Datei **Ergebnis_xx.txt** die richtigen Werte für Magnetkraft und Koppelfluss stehen..

Einbindung in den OptiY-Workflow

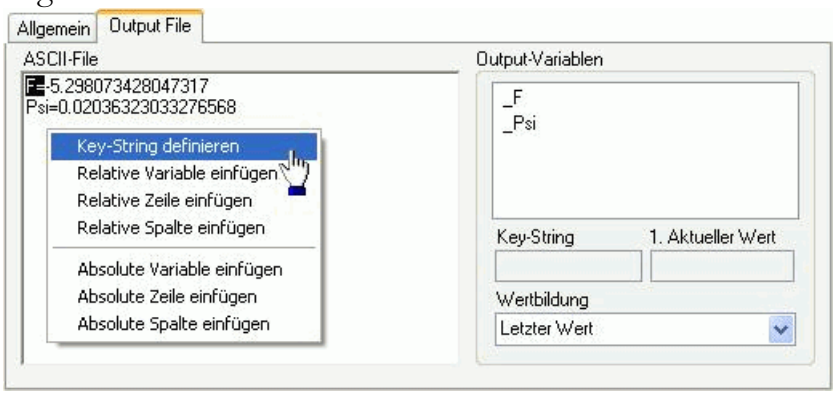
Falls es nicht mehr geöffnet ist, so öffnen wir erneut OptiY mit der Datei Magnet_xx.opy und ergänzen nun die fehlende Einbindung des Modells.

- Einfügen der Symbole von Input- und Output-File auf dem Workflow, Benennung und Zuordnung der konkreten ASCII-Textdatei:

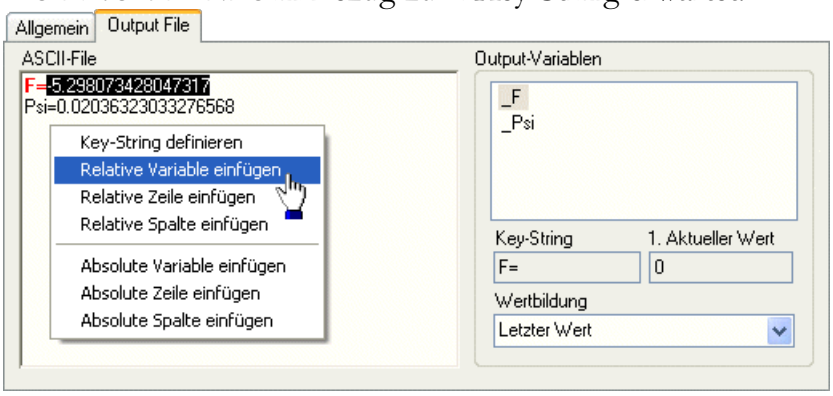


- ***Workaround***:
  In der OptiY-Version 4.5 (Stand 24.01.2019) existiert ein Problem mit dem Auslesen von **Keystring**-basierten **Relativen Variablen** aus einer Output-Datei. Anstatt nach dem im folgenden beschriebenen Vorgehen muss man deshalb die Werte der Variablen aus **Absoluten Zeilen einfügen** und das LUA-Script entsprechend anpassen (Siehe: **Output-File in Übung 4**). Mit OptiY-Version 4.6 ist das Problem behoben und die Anleitung kann wieder regulär genutzt werden.
- Bevor wir die Nennwerte des Workflows mit der Input-Datei verbinden, beginnen wir mit dem Herstellen der beiden Output-Verbindungen zum Output-File. Wir markieren dazu im allgemeinen Dialog des Outputfiles die zu verbindenden Ausgangsgrößen ***_F*** und ***_Psi***. Dann wechseln wir in

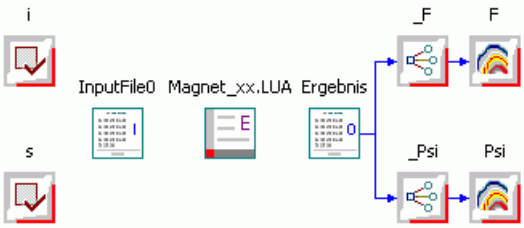
den Output File-Dialog:



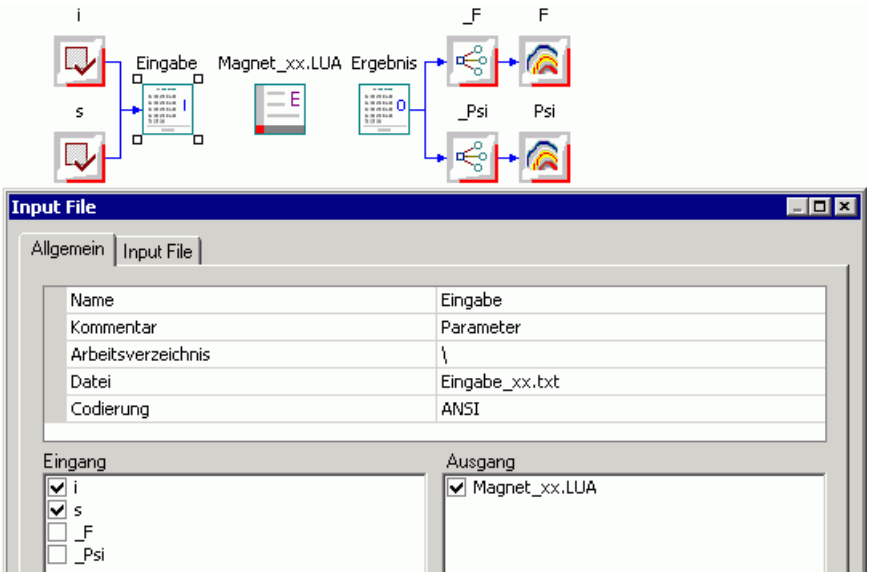
- Nach Wahl der zuzuordnenden Output-Variablen ***_F*** definieren wir **"F="** als **Key-String**, welcher vor dem eigentlichen Wert steht:
  - Nach dem Markieren des Strings führt der Klick mit der rechten Maustaste zum Definitionsmenü.
  - Wird ein Key-String definiert, so wird der Zahlenwert erst nach diesem Schlüsselwort erwartet.
- Die zuzuordnende Output-Variable sollte immer noch ausgewählt sein. Wir definieren nun die Position des Zahlenwertes relativ zum Key-String:
  - Wir markieren dazu die gesamte Zahl ohne Key-String. Der Klick mit der rechten Maustaste führt wieder zum Definitionsmenü.
  - Die Zahl wird als **Relative Variable** in Bezug zum Key-String erwartet:



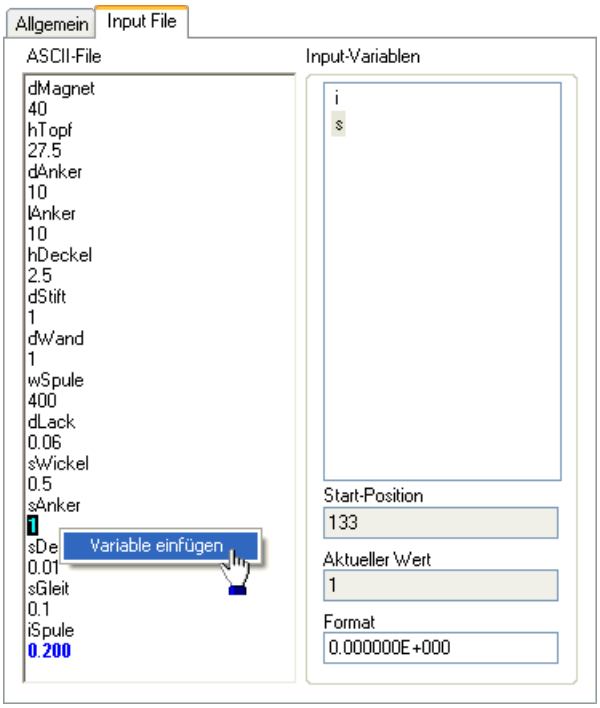
- Mit dieser Art der Definition findet OptiY den Wert für ***_F*** auch, wenn sich durch weitere Ergebnisgrößen die Struktur der Output-Datei ändert. Allerdings muss vor dem Zahlenwert der Key-String "F=" stehen!
- Nach dem gleichen Prinzip ordnen wir die Output-Variable ***_Psi*** dem Datei-Eintrag zu.
- Die Zuordnung der Workflow-Ausgangsgrößen ***_F*** und ***_Psi*** zur Output-Datei widerspiegelt sich im Workflow:



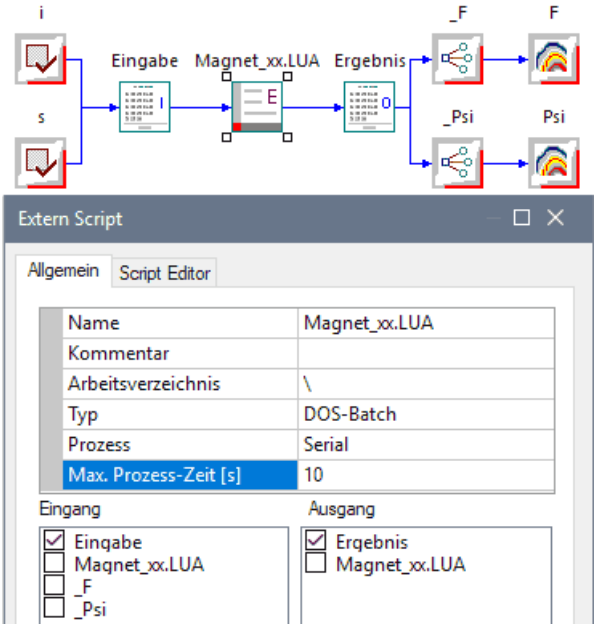
- Es fehlt noch die Zuordnung des Spulestroms ***i*** und des Arbeitsluftspalts ***s*** zur Input-Datei. Nach Doppelklick auf die **Eingabe**-Datei markiert man diese Input-Verbindungen:



- Über die Registerkarte des **Input File** legt man für das OptiY fest, wo die beiden Parameter-Werte in die Input-Datei einzutragen sind. Die Zuordnung innerhalb der Input-Datei funktioniert nach dem gleichen Schema, wie in der Output-Datei (Input-Variable wählen und zugehörigen Wert markieren). Jedoch existiert im Menü der rechten Maustaste nur die Funktion "Variable einfügen":



- Die im Workflow noch fehlende Anbindung der In- und Outputdateien an das Modell wollen wir nun nachholen. Dazu öffnen wir durch Doppelklick den Script-Dialog zu *Magnet_xx.LUA* und markieren die entsprechenden Input-/Output-Verbindungen:



Damit ist der Workflow für das Experiment fertig. Im nächsten Schritt werden wir die Rastersuche als Experiment konfigurieren.

Hinweis:
Ein **Wert > Null** für die **"Max. Prozess-Zeit [s]"** verhindert ein Verklemmen der folgenden Experimente infolge einer aus kleinen Luftspalten manchmal resultierenden fehlerhaften Vernetzung (extreme Elementezahl führt zu extremen Berechnungszeiten!)

← →

Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld_-_OptiY-Datenschnittstellen&oldid=25740“

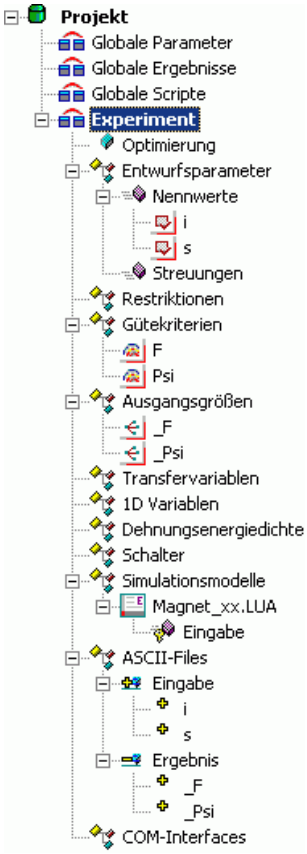
Software: FEM - Tutorial - Magnetfeld - Kennfelder mit OptiY-Rastersuche

Aus OptiYummy

↑

← →

Kennfeldberechnung mit OptiY-Rastersuche



Parallel zur Erstellung der Workflow-Grafik entstand in Form einer Baumstruktur die Grundlage für die Konfiguration eines Experiments innerhalb des OptiY-Versuchsstandes. Nach Wahl einer Komponente in der Baumstruktur erscheinen die zugehörigen Parameter im Eigenschaft-Fenster:

Inhaltsverzeichnis

- 1 Optimierungsverfahren
- 2 Entwurfsparameter (Nennwerte)
- 3 Gütekriterien
- 4 Visualisierung der Experiment-Ergebnisse
- 5 Start des Experiments
- 6 Auswertung des Experiments

Optimierungsverfahren

- Wir nutzen das globale Suchverfahren der Rastersuche, um die Kennfelder durch Abtastung zu berechnen.
- Wir wählen vorläufig 121 Simulationsläufe (11 Gleichverteilte Abtastpunkte in jeder Richtung ergeben eine Teilung in 10 Intervalle). Bei einer Rechenzeit von ca. einer Sekunde für eine Stützstelle bleibt man dabei im Bereich von wenigen Minuten:

Eigenschaft

Optimierung	
Auto-Stop	Manueller Stop
Optimierungsschritte	121
Startschrittweite	Standard
Verfahren	Raster-Suche

Entwurfsparameter (Nennwerte)

- Wir nutzen die Entwurfsparameter als Variablen der Kennfeldfunktionen ***F = f (i,s)*** und ***Psi = f (i,s)***:

Eigenschaft	
Nennwert Daten	
Name	s
Einheit	mm
Kommentar	Arbeitsluftspalt
Werte	
Wert	0
Untergrenze	0.03
Obergrenze	4
Genauigkeit	0
Typ	Variable

Eigenschaft	
Nennwert Daten	
Name	i
Einheit	A
Kommentar	Spulenstrom
Werte	
Wert	0
Untergrenze	0.001
Obergrenze	10
Genauigkeit	0
Typ	Variable

Achtung:

- Der ***Arbeitsluftspalt s*** sollte im Bereich von **0 bis 4 mm** variiert werden. Man muss jedoch beachten, dass die entstehende Geometrie im FE-Modell vernetzbar bleiben muss! Das ist bei ***s*=0 mm** nicht mehr der Fall. Zum Glück existiert auch in der Praxis auf Grund von Oberflächen-Rauigkeit oder Fremdschichten immer ein Restluftspalt von mehr als **30 µm**. Diesen wollen wir als unteren Grenzwert angeben.
- Der ***Spulenstrom i*** sollte im Bereich von **0 bis 10 A** berücksichtigt werden, was für die FEM-Simulation problemlos möglich ist (bei I=0 A ergeben sich alle Output-Variablen ebenfalls zu Null). Allerdings werden vom *OptiY* parallelisierte Simulationen mit gleichen Ergebnissen (hier Null) als "failed" behandelt, weshalb man einen sehr kleinen Wert als Strom-Untergrenze verwenden muss (z.B. **1 mA**).
- Die Maßeinheit ***A*** sollte man als Einheit eintragen, auch wenn OptiY Maßeinheiten bei der Berechnung nirgendwo berücksichtigt. Das gleiche gilt für einen sinnvollen Kommentar. Analog dazu ergänzen wir für den Arbeitsluftspalt ***s*** die Maßeinheit **mm** und einen Kommentar.

Gütekriterien

- Die Gütekriterien ***F*** und ***Psi*** verwenden wir als Funktionswerte für die zu berechnenden Kennfelder. Auch hier sollte man die Maßeinheiten ergänzen und auf sinnvolle Kommentierung achten:

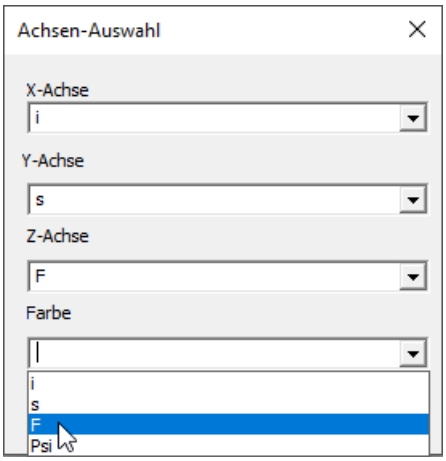
Eigenschaft	
Kriterium Daten	
Name	Psi
Einheit	Wb
Kommentar	Koppelfluss
Werte	
Gewichtsfaktor [0..1]	1
Optimierungsziel	Minimieren
Letzter Wert	0

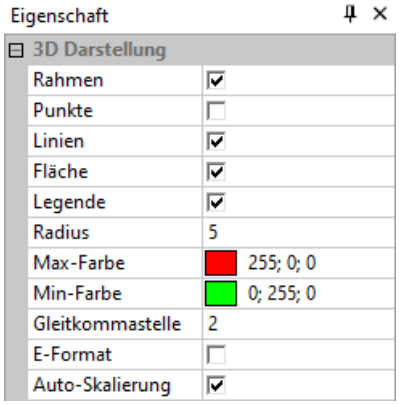
Eigenschaft	
Kriterium Daten	
Name	F
Einheit	N
Kommentar	Magnetkraft
Werte	
Gewichtsfaktor [0..1]	1
Optimierungsziel	Minimieren
Letzter Wert	0

- Der Gewichtsfaktor=1 hat hier keine Bedeutung. Auch wird kein Optimierungsziel verfolgt, wir nutzen die Rastersuche nur zur Berechnung der Kennfelder.

Visualisierung der Experiment-Ergebnisse

- Bisher wird der Grafikbereich von OptiY durch das Fenster des Workflows belegt. Dieses kann man nun schließen (***Ansicht > Workflow***).
- Wir wollen die beiden Kennfelder als 3D-Diagramme visualisieren (***Analyse > Darstellung > 3D-Darstellung***).
- Im rechten Bild gezeigt für ***F = f (i,s)*** erfolgt die Zuweisung der Achsen bei ***Psi = f (i,s)*** analog.
- ***Fenster > Nebeneinander*** führt meist zu einer übersichtlichen Anordnung.
- Nach Auswahl eines 3D-Diagramm-Fensters durch Anklicken mit der Maus, kann man die Darstellung im Eigenschaftsfeld konfigurieren:



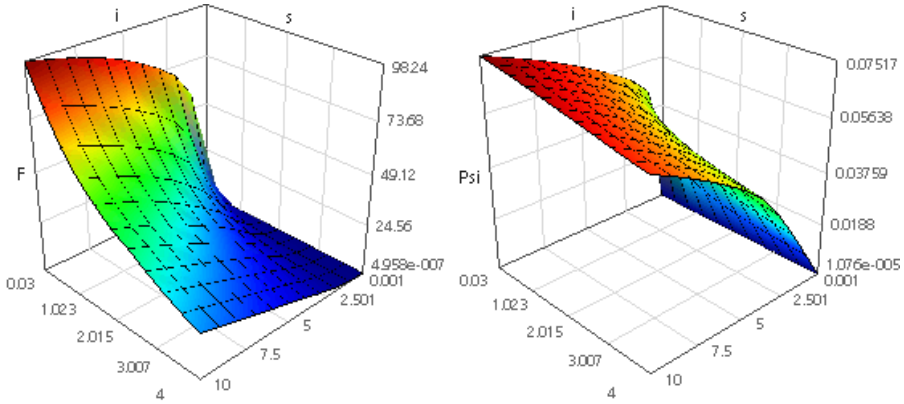


- Im Beispiel sollte man die Auto-Skalierung der Achsen nutzen.

Start des Experiments

Achtung: Spätestens vor Beginn des Experiments sollte man die aktuelle Konfiguration speichern!

- Die Berechnung des durch die Nennwerte aufgespannten Lösungsraums beginnt nach dem Starten der Optimierung (***Projekt - Start*** **<F5>**).
- Falls die Verbindungen zu den Input-/Output-Dateien richtig hergestellt wurden, sollten sich die Kennfelder zeilenweise abbilden:

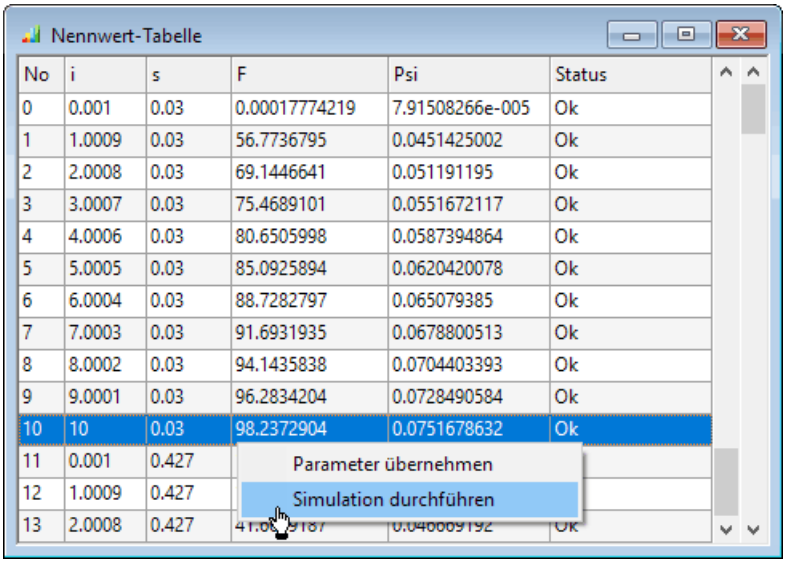


Hinweise:

- Mit gedrückter linker Maustaste kann man die 3D-Darstellungen frei drehen.
- Eventuelle fehlerhafte Modellberechnungen erscheinen als "Zacken" in der ansonsten stetigen Fläche der Kennfelder. Bei FE-Simulationen ist die Gefahr z.B. von Vernetzungsfehlern in engen Bereichen relativ groß, wenn man eine freie Variation von Geometrieparametern gestattet. In der nachfolgend beschriebenen Nennwert-Tabelle sind fehlerhafte Simulationsläufe mit **Status=False** gekennzeichnet. Wie bei der Experiment-Auswertung beschrieben, kann man die Ursache der fehlerhaften Simulationsläufe durch Aufruf des Simulationsprogramms analysieren.

Auswertung des Experiments

Alle Abtaststellen der Rastersuche sind in der Nennwert-Tabelle aufgelistet (***Analyse > Nennwert-Optimierung > Nennwert-Tabelle***):



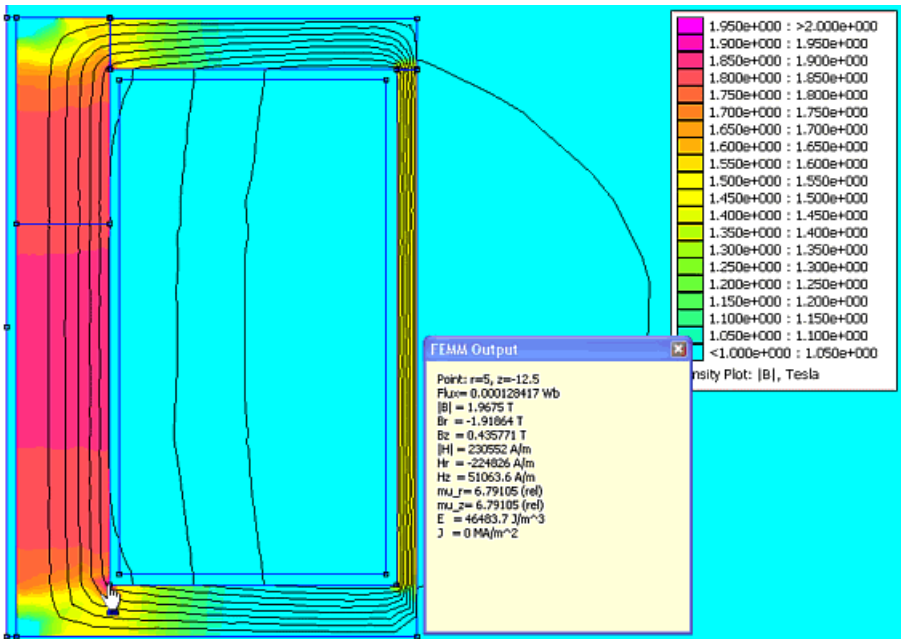
Nennwert-Tabelle

No	i	s	F	Psi	Status
0	0.001	0.03	0.00017774219	7.91508266e-005	Ok
1	1.0009	0.03	56.7736795	0.0451425002	Ok
2	2.0008	0.03	69.1446641	0.051191195	Ok
3	3.0007	0.03	75.4689101	0.0551672117	Ok
4	4.0006	0.03	80.6505998	0.0587394864	Ok
5	5.0005	0.03	85.0925894	0.0620420078	Ok
6	6.0004	0.03	88.7282797	0.065079385	Ok
7	7.0003	0.03	91.6931935	0.0678800513	Ok
8	8.0002	0.03	94.1435838	0.0704403393	Ok
9	9.0001	0.03	96.2834204	0.0728490584	Ok
10	10	0.03	98.2372904	0.0751678632	Ok
11	0.001	0.427			
12	1.0009	0.427			
13	2.0008	0.427			Ok

Parameter übernehmen
Simulation durchführen

Mit diesen Werten kann man jeweils einen einzelnen Aufruf des Simulationsprogramms starten (**Kontextmenü rechte Maus** > ***Simulation durchführen***). Damit könnte man z.B. das Magnetfeld für den Worstcase (kleinster Luftspalt und größter Strom) analysieren. Leider wird in unserem speziellen Fall das FEMM-Programm nach der Abarbeitung des Scripts sofort wieder beendet:

- Die Werte des Optimums stehen jetzt aber in der Datei ***Eingabe_xx.txt***.
- Nach dem Auskommentieren des exit-Befehls im LUA-Script starten wir danach das FEMM-Programm im Windows-Betriebssystem und öffnen darin das Lua-Script zur Ausführung.
- Es stehen uns nun die aktuellen Parameter der letzten Simulation im FEMM-Programm zur Verfügung:



- Für das obige Bild wurde die Zahl der Flusslinien auf 9 reduziert und die Grenzen für den Density-Plot der Flussdichte wurden auf den Bereich von 1 bis 2 T gesetzt. Man erkennt nun deutlich die Bereiche der größten Flussdichte, welche an der markierten Stelle ca. 2 T erreicht (links unten an der Innenkante).
- Damit der Klick mit der linken Maustaste die zur Cursorposition gehörenden Ergebnisgrößen im Ausgabefenster anzeigt, muss man dieses zuvor mit ***View > Output Window*** aktivieren und

Operation > Point properties wählen.

← →

Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld_-_Kennfelder_mit_OptiY-Rastersuche&oldid=22511“

Software: FEM - Tutorial - Magnetfeld - Kennfelder - Parallelisierung

Aus OptiYummy

↑

← →

Kennfeldberechnung mit parallelisierter Rastersuche

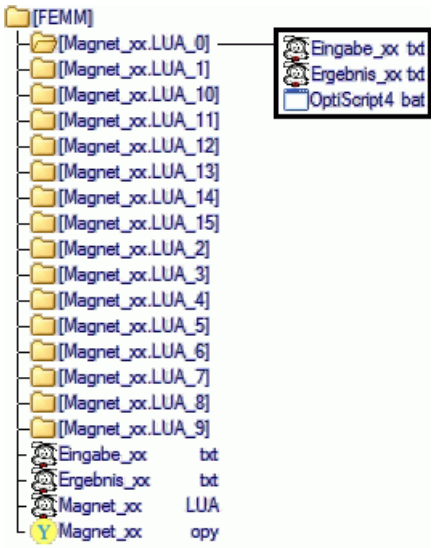
Bisher haben wir für die Durchführung der Rastersuche den **seriellen Prozess** gewählt:

- Die nächste Modellberechnung für die Rastersuche wird erst gestartet, wenn die vorherige Modellberechnung beendet ist.
- Damit kann es zu keinen Ressourcenkonflikten zwischen den einzelnen Modellberechnungen kommen. Deshalb sind auch keine besonderen Maßnahmen zur Vermeidung solcher Konflikte erforderlich. Alle Dateien, welche während einer Modellberechnung nötig sind, können im aktuellen Arbeitsverzeichnis des Experiments verwaltet werden.

Bei der Rastersuche können alle Stützstellen völlig unabhängig voneinander berechnet werden. Deshalb kann nach Wahl von **paralleler Prozess** der gleichzeitige Start aller erforderlichen Modellberechnungen erfolgen. Dazu sind zuvor jedoch Maßnahmen zur Vermeidung von Ressourcenkonflikten zwischen den Modellberechnungen erforderlich.

Im Folgenden wird eine Lösung für die Parallelisierung der FEMM-Aufrufe im OptiY-Workflow entwickelt, welche auch für den seriellen Prozess funktioniert. Es wird dabei auf die Vermeidung von absoluten Pfadangaben geachtet, damit die Modelle und Prozesse portierbar bleiben.

1. Individuelle Script-Ordner pro Modell-Aufruf:



- Für jeden einzelnen Script-Aufruf innerhalb des parallelen Prozesses wird vom OptiY im aktuellen Script-Arbeitsverzeichnis ein individueller Script-Ordner angelegt (Ordner-Name = Script-Name + Nummer).
- In jedem dieser Ordner wird die Input-Datei mit den zum Aufruf gehörenden Modell-Parametern gespeichert.
- Eine Kopie des Scriptes (hier .BAT) wird ebenfalls in jedem Ordner gespeichert und aufgerufen.
- Nach Ausführen dieses Scriptes erwartet OptiY die Output-Datei mit den Berechnungsergebnissen im zugehörigen Ordner.
- Die Ordnerstruktur für eine 4x4-Rastersuche (16 Schritte) wird im nebenstehenden Bild gezeigt.
- Für den Ordner "0" sind beispielhaft die enthaltenen Dateien angedeutet.
- "OptiScript4" enthält die Befehle des BAT-Files.
- Nach erfolgreicher Abarbeitung aller parallel gestarteten Script-Aufrufe werden die generierten Ordner vom OptiY wieder gelöscht.

2. Zuordnung des Lua-Scripts zu den Modell-Aufrufen

- Das Modell in Form des Lua-Scripts existiert in einem Exemplar nur im Script-Arbeitsverzeichnis. Es wird vom OptiY keine Kopie in den generierten Unterordnern erzeugt.
- **Die serielle Abarbeitung** ist relativ einfach, weil man die Lua-Datei aus dem aktuellen Ordner des BAT-Scriptes ohne Angabe eines Pfad-Namens verwenden kann:

```
cd /D %~dp0
C:\Programme\femm42\bin\femm.exe -lua-script=Magnet_xx.LUA -
windowhide
```

- **Die parallele Abarbeitung** muss die Lua-Datei aus dem übergeordneten Ordner des BAT-Scriptes benutzen. Für jedes BAT-Script wird ja ein individueller Unterordner generiert, der damit zum aktuellen Ordner wird:

```
cd /D %~dp0
C:\Programme\femm42\bin\femm.exe -lua-script=..\Magnet_xx.LUA -
windowhide
```

- **Wahlfreie serielle oder parallele Abarbeitung** eines Scripts kann man nur durch Testen und Umschalten der Aufruf-Methode gewährleisten:

```
:: Ordner des Batch-Skripts wird zum aktuellen Ordner:
cd /D %~dp0
:: Test, ob Lua-Datei im aktuellen Ordner existiert:
if exist Magnet_xx.LUA goto seriell
:parallel
C:\Programme\femm42\bin\femm.exe -lua-script=..\Magnet_xx.LUA -
windowhide
goto Ende
:seriell
C:\Programme\femm42\bin\femm.exe -lua-script=Magnet_xx.LUA -
windowhide
:Ende
```

- Die Möglichkeit der wahlfreien seriellen oder parallelen Abarbeitung sollte man anstreben. Dabei wird man zum Testen des Prozesses immer mit der Realisierung des seriellen Modellaufrufes beginnen. Die erforderlichen Anpassungen an den parallelen Prozess erfolgen dann abschließend.
- **Hinweis:** Der FEMM-Parameter **-windowhide** unterdrückt die FEMM-Fenster. Dies erhöht die Abarbeitungsgeschwindigkeit der parallelen Prozesse sehr stark!

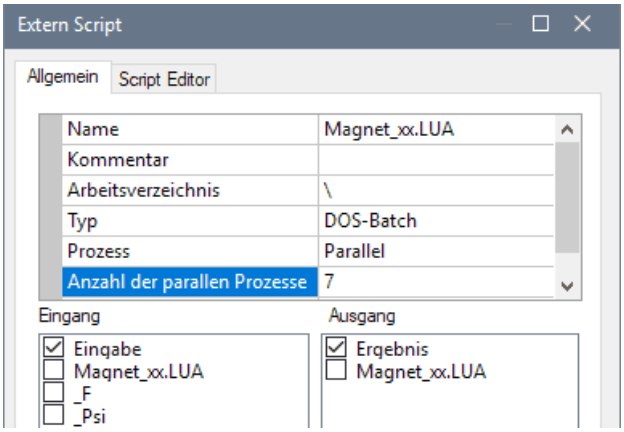
3. Zuordnung der Input-/Outputdateien zum Lua-Script

- Das Lua-Script benutzt für seine Daten immer den aktuellen Script-Ordner (gewährleistet durch cd-Befehl im BAT-Script).
- Damit kann im Lua-Script die Angabe eines Absolutpfades entfallen. Die entsprechende **mydir**-Zuweisung wird dafür durch eine Zeichenkette der Länge 0 ersetzt:

```
mydir="";
```

Hinweise:

- Die Anzahl der gleichzeitig möglichen FEMM-Aufrufe wird neben der Hardware-Ausstattung des Computers überwiegend durch das verwendete Betriebssystem bestimmt.
- Unter Windows 7/8/10 - 64 Bit funktioniert meist noch eine Rastersuche mit 81 Schritten (9x9). Abhängig von der konkreten System-Konfiguration kann jedoch auch bereits bei 64 Schritten (8x8) Schluss sein!
- Unter Windows XP - 32 Bit konnte man z.B. nur das Prinzip der parallelisierten Rastersuche mit 16 Schritten (4x4) testen. Inwieweit dies für andere 32-Bit-Systeme gilt, wurde nicht getestet.
- Ein Überschreiten der möglichen Anzahl führt zu Fehlermeldungen und Abstürzen der "überzähligen" FEMM-Aufrufe, welche man manuell quittieren muss. Außerdem hat man dann das Problem der Bereinigung des Arbeitsverzeichnisses von überflüssigen temporären Dateien und Ordner.
- Gibt man für die "**Anzahl der parallelen Prozesse**" keinen Wert > 0 an, so werden immer alle parallelisierbaren Modellrechnungen gleichzeitig gestartet. Bei einer Rastersuche sind das alle angegebenen Optimierungsschritte, welche jeweils durch Start des FEMM-Programms repräsentiert werden. Danach ist der Computer nur noch mit der Verwaltung dieser vielen Prozesse beschäftigt, was den Wirkungsgrad für die eigentlichen Berechnungen stark senkt und den Computer meist in dieser Zeit unbedienbar macht:



- Den größten Berechnungswirkungsgrad erzielt man, wenn man sich an der Anzahl der verfügbaren CPU-Kerne orientiert (bei *Hyper-Threading* gleich Anzahl logischer CPU-Kerne):
  - Auf älteren Dual-Core-Systemen sollte man sich auf zwei parallele Prozesse beschränken.
  - Bei N-Core-Systemen sollte man sich auf (N-1) parallele Prozesse beschränken, um den letzten CPU-Kern für die System-Bedienung zu erhalten.
- Geschickt eingesetzt führt die Prozess-Parallelisierung zu eine enormen Steigerung der Berechnungsgeschwindigkeit. Die Grenzen der Parallelisierung unter Windows überwindet man durch Verlagerung der Modellaufrufe in eine Netzwerk-Umgebung.

← →

Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld_-_Kennfelder_-_Parallelisierung&oldid=22516“

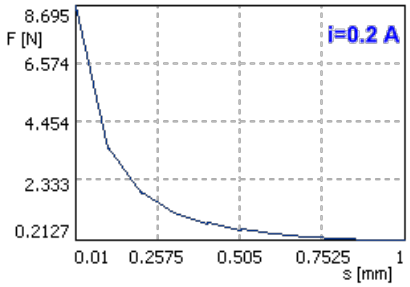
Software: FEM - Tutorial - Magnetfeld - Wirkung kleiner Luftspalte

Aus OptiYummy

↑

← →

Wirkung kleiner Luftspalte



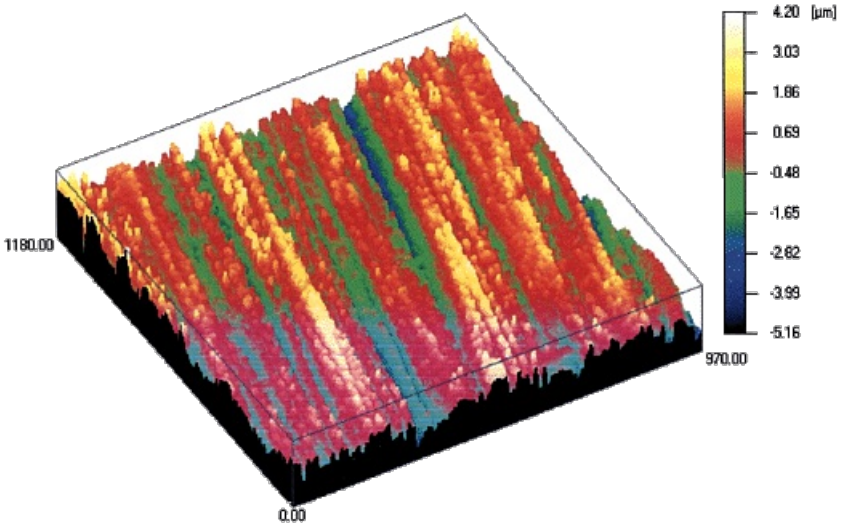
Im angezogenen Zustand eines E-Magneten besteht häufig die Aufgabe, mit möglichst kleinem Strom eine ausreichende Haltekraft aufzubringen. Das wird unterstützt durch die stark nichtlineare Kennlinie ***F = f (s)***, welche zu einem steilen Anstieg der Magnetkraft für kleine Arbeitsluftspalte führt. Für obiges Diagramm wurde der Kraftverlauf bei **i=0.2 A** im Bereich des Arbeitsluftspalts von **10 µm ... 1 mm** berechnet.

Bei angezogenem Anker existieren innerhalb unseres Magnetkreises insgesamt 3 kleine Luftspalte:

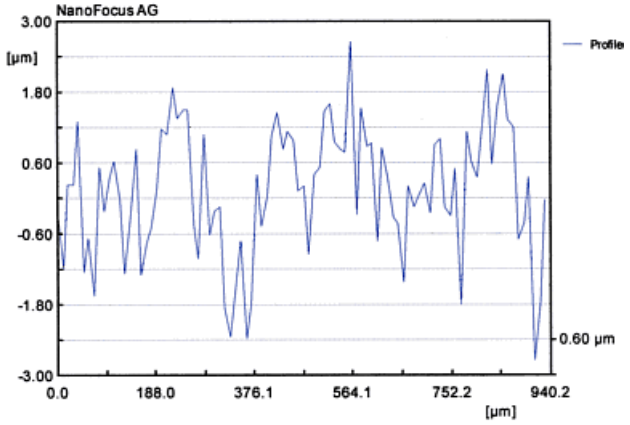
- Restspalt des Arbeitsluftspalts
- Führungsluftspalt zwischen Anker und Deckel;
- Luftspalt zwischen Magnet-Topf und Deckel.

Leider besitzen kleine Luftspalte innerhalb eines Magnetkreises die Eigenschaft, in ihrem magnetisch wirksamen Wert in Bezug auf ihren Nennwert eine große prozentuale Streuung aufzuweisen:

- Auch geschliffene Flächen besitzen noch Oberflächen-Profile mit Höhenunterschieden im µm-Bereich. Das folgende Bild zeigt den überhöhten 2½D-Scan einer geschliffenen, laminierten Polfläche:



- Aus diesem Gesamt-Scan zeigt das nächste Diagramm einen Schnitt:



- Zusätzlich bilden sich im Verlauf der Zeit auf den Oberflächen Korrosions- und Verschmutzungsschichten im µm-Bereich.
- Weiterhin sind die Oberflächen nicht exakt eben und orthogonal.

Es soll mit dem folgenden Experiment untersucht werden, in welchem Maße die Haltekraft des E-Magneten bei einem Strom **i=0.2xx A** von den Streuungen der Luftspaltwerte abhängt (**xx**=01..99 - Teilnehmer-Nummer). Dazu werden folgende Normalverteilten Streuungen angenommen:

- **sAnker** = 50 µm ±20 µm
- **sDeckel** = 20 µm ±10 µm
- **sGleit** = 80 µm ±20 µm

← →

Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld_-_Wirkung_kleiner_Luftspalte&oldid=13294“

Software: FEM - Tutorial - Magnetfeld - Probabilistik - Konfiguration

Aus OptiYummy

↑

← →

Probabilistische Simulation - Experimentkonfiguration

Nennwert-Simulation:
Modelle, mit denen der Konstrukteur heutzutage arbeitet, benutzen fast ausschließlich konkrete Werte als Parameter. Die berechneten Simulationsergebnisse entsprechen demzufolge einem Ist-Zustand des modellierten Objektes. Man spricht auch von einer Nennwert-Simulation.

Die "konkreten" Werte für die Modellparameter können dabei unterschiedlichster Natur sein (ganze Zahlen, reelle Zahlen, komplexe Zahlen, Vektoren, Matrizen, Funktionsverläufe).

Probabilistische Simulation:
Leider sind alle physikalisch-technischen Größen, welche durch die Modellparameter repräsentiert werden, in der Realität mit Streuungen behaftet. Ihre Werte streuen bedingt durch die Fertigung und die Betriebsbedingungen von Exemplar zu Exemplar, aber auch bei mehrmaliger Nutzung eines Exemplars des modellierten Objektes.

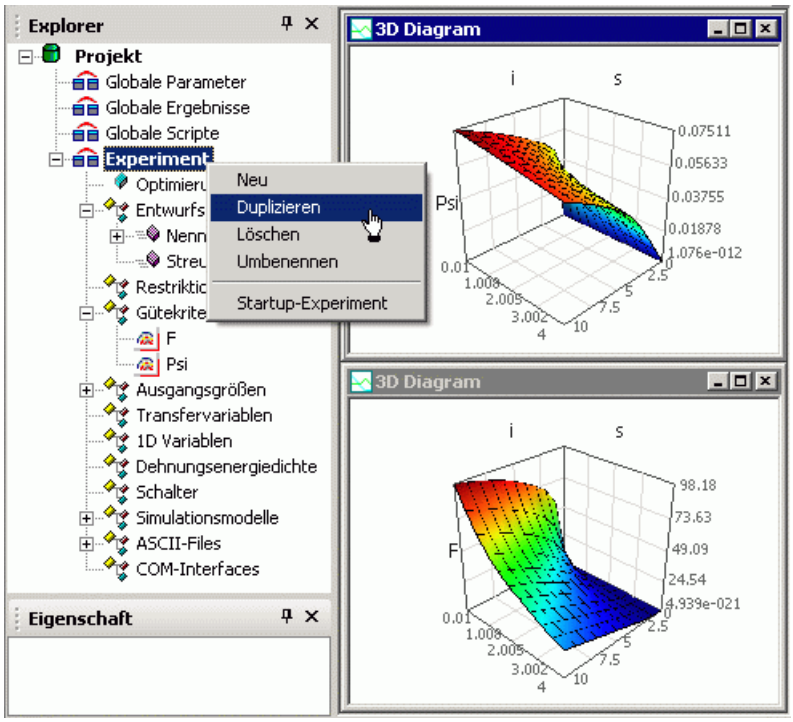
Auch gibt es Unsicherheiten in Hinblick auf die Simulationsergebnisse durch die Idealisierung bei der Modellbildung und numerische Ungenauigkeiten bei der Modellberechnung.

Die probabilistische Simulation bietet die Möglichkeit, diese Streuungen physikalisch-technischer Größen und des Objektverhaltens in Form von Verteilungsdichtefunktionen zu berücksichtigen. Die Simulation erfolgt nicht mehr mit "konkreten" Werten, sondern berücksichtigt die Streuung der Werte. Wir betrachten damit nicht nur ein konkretes Exemplar des modellierten Objekts unter konkreten Betriebsbedingungen.

Es wird praktisch eine Stichprobe von allen möglichen Exemplaren und von allen möglichen Betriebsbedingungen simuliert. Die Ergebnisse dieser Simulation erlauben auf Basis der statistischen Versuchsplanung Aussagen zu statistischen Eigenschaften des modellierten Objekts.

Experiment-Konfiguration

OptiY bietet die Möglichkeit, innerhalb eines Projektes mehrere Experimente zu verwalten. Ein vorhandenes Experiment mit seiner aktuellen Konfiguration kann man nach Anwahl im OptiY-Explorer mittels der rechten Maustaste einfach Duplizieren:



- Nach dem ***Duplizieren*** des Experiments erscheint im OptiY-Explorer die Kopie als ***Experiment2***.
- Dieses weitere Experiment besitzt am Anfang die gleiche Konfiguration wie das Original. Nur die Anzeigefenster muss man neu definieren.
- Verwaltet man mehrere Experimente in einem Projekt, so sollte man mittels ***Umbenennen*** dafür sinnvolle Namen vergeben:
  - Das bisherige Experiment könnte man z.B. ***Kennfeld-Berechnung*** nennen.
  - Das neue Experiment soll mittels Monte-Carlo-Methode die Streuung simulieren: ***Monte Carlo***.

- Für jedes Experiment wird von OptiY ein separater Workflow verwaltet, so dass darin unabhängig voneinander Änderungen vorgenommen werden können.
- Die Befehle des Projekt-Menüs (**Start / Stopp / Weiter / Rücksetzen**) wirken nur auf das **Startup-Experiment**. Deshalb sollte man das aktuell zu bearbeitende Experiment als Startup-Experiment auswählen, damit man nicht aus Versehen z.B. durch Rücksetzen die Ergebnisse eines anderen Experiments zurücksetzt.

Beachte:

- Die *OptiY Trial Edition* ist beschränkt auf die Verwendung von jeweils 3 Nennwerten und Streuungen. Um Platz zu schaffen für die 3 Nennwerte der Luftspalt-Größen, löschen wir im Workflow des Monte-Carlo-Experiments den Nennwert **i** des Stroms.
- Der geforderte Stromwert für **iSpule=0.2xx A** (mit **Teilnehmer-Nr. xx=01..99**) muss nun manuell in die Input-Datei eingetragen werden!

Wir ergänzen im Workflow die Streugrößen für die drei Luftspalte (***Einfügen > Entwurfsparameter > Streuungen***) und vergeben dafür eindeutige Namen und sinnvolle Kommentare. Bei der Wahl der Namen müssen Syntax-Anforderungen von Programmiersprachen (z.B. *C* und *Modelica*) berücksichtigt werden:

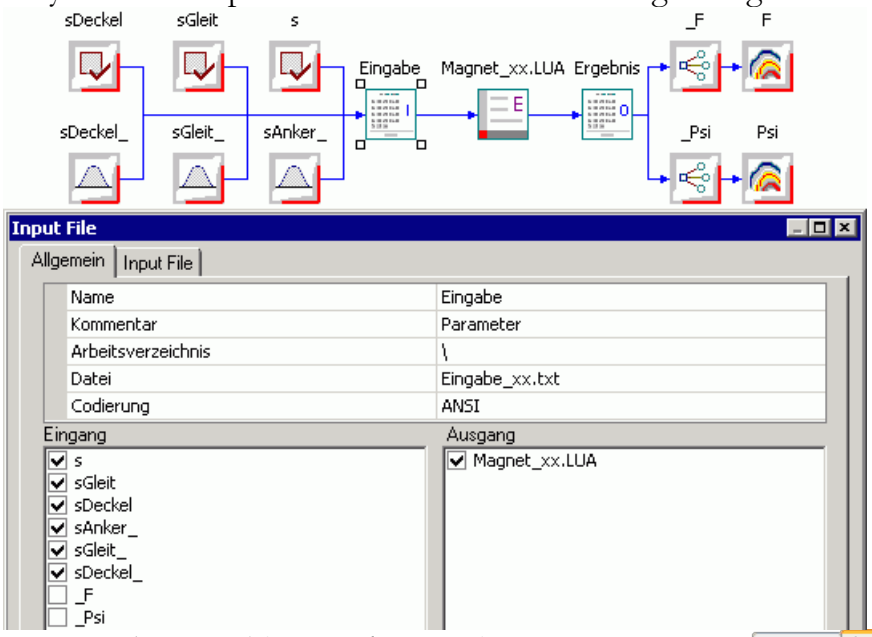
- ***sAnker_*** = Streuung Anker-Restspalt
- ***sGleit_*** = Streuung Gleitspalt
- ***sDeckel_*** = Streuung Deckelspalt

Für ***sAnker*** wurde bereit der Nennwert ***s*** definiert. Wir ergänzen die zwei zusätzlichen Luftspalte:

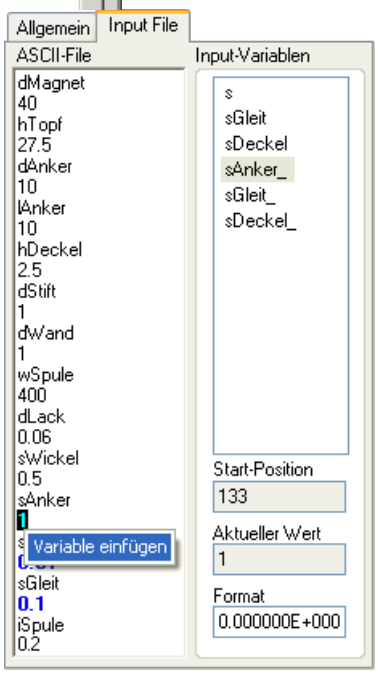
- ***sGleit*** = Nennwert Gleitspalt
- ***sDeckel*** = Nennwert Deckelspalt

Die Zuordnung der Streuungen zum Input-File ***Eingabe.txt*** erfolgt analog zu den Nennwerten:

- Doppelklick auf das Symbol des Input-Files öffnet den Verbindungsdialog:



- In der **Input File** - Registerkarte wählt man die jeweilige Input-Variable, markiert im ASCII-File den zugehörigen Zahlenwert und veranlasst über die rechte Maustaste ***Variable einfügen***.



Achtung:

- Bereits einem Nennwert zugeordnete Eingabe-Werte muss man zusätzlich der zugehörigen Streugröße zuordnen, wie dies am Beispiel des Arbeitsluftspalts ***sAnker*** deutlich wird. Damit "erkennt" OptiY, dass Nennwert und Streuung zusammengehören.
- Die Verbindungen im Workflow werden auch gezeichnet, wenn man die Zuordnungen im *Input File* mittels ***Variable einfügen*** nicht korrekt durchgeführt hat. Deshalb sollte man die Zuordnung aller Input-Variablen in dieser Registerkarte sorgfältig überprüfen!

Die Streu-Breiten beziehen sich immer auf die zugehörigen Nennwerte (=Toleranzmittenwerte), welche für die zu untersuchende Lösung einen aktuellen Wert besitzen:

- sAnker = 50 µm ±20 µm
- sDeckel = 20 µm ±10 µm
- sGleit = 80 µm ±20 µm

Im Beispiel weichen diese Nennwerte von den ursprünglichen Werten des Input-Files ab:

- Da sich die Nennwerte der streuenden Parameter während dieser Untersuchung nicht ändern sollen, wird ihr ***Typ=Konstante*** gesetzt.
- Wir setzen die Nennwerte ***s=50 µm***, ***sDeckel=20 µm*** und ***sGleit=80 µm***:

Eigenschaft	
Nennwert Daten	
Name	s
Einheit	mm
Kommentar	Arbeitsluftspalt
Werte	
Wert	0.05
Typ	Konstante

Eigenschaft	
Nennwert Daten	
Name	sDeckel
Einheit	mm
Kommentar	Restspalt am Deckel
Werte	
Wert	0.02
Typ	Konstante

Eigenschaft	
Nennwert Daten	
Name	sGleit
Einheit	mm
Kommentar	Führungsspalt
Werte	
Wert	0.08
Typ	Konstante

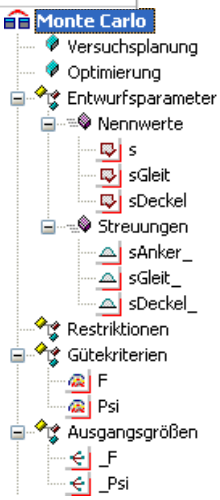
Nach dem Einstellen der erforderlichen Nennwerte konfigurieren wir die zugehörigen Streuungen:

- Auch hier achten wir auf die korrekten Einheiten und auf sinnvolle Kommentare.
- Obwohl sich die Toleranzen (=Streubreiten) der Luftspalte während dieser Untersuchung nicht ändern sollen, wird hier ***Typ=Zufall*** gesetzt. Damit bleiben die Streubreiten konstant, der Parameter bewegt sich aber zufällig im Toleranzbereich. Die Einstellung "Konstante" würde den Parameter vollständig fixieren.
- Wir nehmen als Form die **Normalverteilung** für alle drei Streu-Größen an.
- Der Wert ***T*** von ***Toleranz*** beschreibt die Breite des Variationsbereiches um den ***aktuellen Wert***, der standardmäßig dem zugehörigen Nennwert ***N*** entspricht: ***(N-T/2)*** bis ***(N+T/2)***.
- Für die Normalverteilung ergibt sich ***T*** aus der Standardabweichung σ ($T = 6 \cdot \sigma$).
- Für die Normalverteilung umfasst die Toleranzbreite ***T*** 99,73 % aller "Messwerte" einer Stichprobe:

Eigenschaft	
Streuung Daten	
Name	sAnker_
Einheit	mm
Kommentar	Arbeitsluftspalt
Versuchsplanung	
Nennwert	0
Toleranz	0.04
Genauigkeit	0
Verteilung	Normalverteilung
Typ	Zufall
Virtueller Entwurf	
Entwurfsparameter	☐
Nennwert	0.05
Toleranz	0.04
Verteilung	Normalverteilung
Typ	Zufall

Eigenschaft	
Streuung Daten	
Name	sDeckel_
Einheit	mm
Kommentar	Restspalt am Deckel
Versuchsplanung	
Nennwert	0
Toleranz	0.02
Genauigkeit	0
Verteilung	Normalverteilung
Typ	Zufall
Virtueller Entwurf	
Entwurfsparameter	☐
Nennwert	0.02
Toleranz	0.02
Verteilung	Normalverteilung
Typ	Zufall

Eigenschaft	
Streuung Daten	
Name	sGleit_
Einheit	mm
Kommentar	Führungsspalt
Versuchsplanung	
Nennwert	0
Toleranz	0.04
Genauigkeit	0
Verteilung	Normalverteilung
Typ	Zufall
Virtueller Entwurf	
Entwurfsparameter	☐
Nennwert	0.08
Toleranz	0.04
Verteilung	Normalverteilung
Typ	Zufall



"Nennwert" einer Streuung:

Der Begriff *Nennwert* als Parameter der Streuung ist ungünstig gewählt. In der OptiY-Hilfe steht dazu:

> "Das ist der Nennwert der Streuung. Wenn noch ein Nennwert des Experiments zusammen mit dieser Streuung demselben Modellparameter zugeordnet wird, ist der gesamte Nennwert für den Modellparameter die Summe aus den beiden Nennwerten."

Es handelt sich hier also um den *Toleranzmittenabstand* als algebraische Differenz zwischen dem *Toleranzmittenwert* und dem *Nennwert*.
Da unsere Streuung symmetrisch zum eigentlichen Nennmaß liegt, müssen wir diesen **Toleranzmittenabstand=0** setzen!

Werte des virtuellen Entwurfs:

- Auf Basis der laut Versuchsplanung durchgeführten Modellberechnungen wird im OptiY immer ein Ersatzmodell in Form mathematischer Funktionen gewonnen.
- Auf Grundlage dieses Ersatzmodells können ohne erneute Modelldurchrechnungen Analysen und Optimierungen durchgeführt werden (*virtueller Entwurf*).
- **Entwurfsparameter=False** bedeutet, dass beim "virtuellen Entwurf" die Toleranzbreite und Nennwert einer Streugröße als gegeben angenommen werden. Der Nennwert dieser Streuung wird bei einer "virtuellen Optimierung" also nicht verändert.
- **Nennwert** beschreibt den aktuellen Arbeitspunkt auf dem Ersatzmodell. Wir benutzen vorläufig den "echten" Toleranzmittenwert.
- **Toleranz** beschreibt die Toleranzbreite für die Simulation einer virtuellen Stichprobe. Wir benutzen dafür den "echten" Toleranzwert. Damit sollten die Eigenschaften der virtuellen Stichprobe denen einer realen Stichprobe mit dem "echten" Modell entsprechen.
- Die Bedeutung dieser Werte wird später bei der Durchführung der Analyse-Experimente noch erläutert.

Wenn man Streuungen als Entwurfsparameter im Experiment verwendet, so ist das ein Kennzeichen dafür, dass man das Niveau der Nennwert-Simulation verlässt. Im Rahmen der statistischen Versuchsplanung muss man nun spezifizieren, mit welchen Methoden man das Verhalten einer Stichprobe nachbilden möchte. Probabilistische Simulation kann nach verschiedenen Verfahren durchgeführt werden:

1. Streuung der Eingangsgrößen durch Generierung von Zufallszahlen.
2. Ermittlung von Übertragungsfunktionen zwischen den streuenden Inputgrößen und den Outputgrößen. Analytische Berechnung der Output-Streuung anhand der Eingangsstreuungen und der ermittelten Übertragungsfunktionen.

Die Anwendung dieser beiden grundlegenden Verfahren der probabilistischen Simulation wird in den folgenden Abschnitten am Beispiel des E-Magneten behandelt.

← →

Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld_-_Probabilistik_-_Konfiguration&oldid=27896“

Software: FEM - Tutorial - Magnetfeld - Probabilistik - Modell-Robustheit

Aus OptiYummy

↑

← →

Probabilistische Simulation - Numerische Stabilität des Modells

Unser FEM-Modell des Magneten funktioniert stabil nur in einem bestimmten Parameterbereich:

- Es dürfen sich keine negativen Bereichsabmessungen (Luftspalte, Luft- bzw. Eisenbereiche) ergeben, sonst existiert keine Lösung.
- Sehr kleine Luftspalte führen im Beispiel zu sehr großen Netzen mit mehreren Millionen Elementen. Eine Berechnung im Rahmen einer Stichproben-Simulation ist aus Zeitgründen praktisch nicht mehr möglich.

Man kann für die Streuungen die Verteilungsfunktion mit Bereichsgrenzen festlegen:

- Leider besitzt aber z.B. eine Normalverteilung keine echten Grenzen, sondern es gibt immer Ausreißer. Diese werden vom OptiY bei der probabilistischen Simulation ebenfalls berücksichtigt.
- Man muss davon ausgehen, dass es bei der Nutzung eines Modells innerhalb eines Tools wie OptiY immer zu "unzulässigen" Parameter-Kombinationen kommen kann.
- Die Kunst besteht darin, innerhalb der Modelle solche unzulässigen Parameter-Kombinationen so zu behandeln, dass für die berechneten Ergebnisgrößen im Sinne einer Funktionsbewertung sinnvolle Werte entstehen.

Unter dem Aspekt unzulässiger Parameter müssen wir unser Modell im Folgenden untersuchen und die erforderliche numerische Stabilität realisieren. Wir beschränken uns dabei auf die Parameter, welche zurzeit vom OptiY in den aktuellen Experimenten verändert werden können. Die Fehlerbehandlung fügen wir unmittelbar nach dem Einlesen der Parameter aus dem Input-File in das Lua-Script ein:

```
-----------------------------------------------------------------
--------------
-- Unzulässige Eingabe-Parameter von OptiY begrenzen
--
-----------------------------------------------------------------
--------------
   if 1*sAnker  < 0.03  then sAnker=0.03;  end;
   if 1*sAnker  > 5     then sAnker=5;     end;
   if 1*sGleit  < 0.05  then sGleit=0.05;  end;
   if 1*sGleit  > 1     then sGleit=1;     end;
   if 1*sDeckel < 0.01  then sDeckel=0.01; end;
   if 1*sDeckel > 1     then sDeckel=1;    end;
   if 1*iSpule <= 1E-10 then iSpule=1E-10; end; -- damit R und L
berechenbar!
```

Dazu einige Erläuterungen:

- Die zu testenden Parameter mussten mit dem Faktor 1 multipliziert werden, da es ansonsten zu einer Fehlermeldung kam. Wahrscheinlich entsteht erst nach Ausführen der Rechenoperation der Variablen-Typ "number", der in einer Vergleichsoperation benutzt werden kann.
- Der Ankerluftspalt musste auf 30 µm begrenzt werden, obwohl im vorherigen Experiment die Kennfelder im Prinzip bis zu einem Wert von 10 µm berechenbar waren.
  Leider kommt es in Kombination mit der Änderung der anderen Luftspalte jedoch zu einem eigenartigen Effekt. Die Kugelsphäre wird mit extrem kleinen Elementen vernetzt. Wahrscheinlich kann FEMM durch Rundungsfehler die Blocklabel nicht mehr exakt in der Mitte schmaler Bereich platzieren. Sie rutschen damit unter Umständen z.B. in die Kugelsphäre und führten damit zu riesigen Netzen.
- Es wurde vorsichtshalber eine obere Begrenzung für die Luftspalte eingeführt, damit in jedem Fall eine gültige Geometrie erhalten bleibt.
- Der Spulenstrom wurde auf einen extrem kleinen Wert größer Null begrenzt, damit die Werte des Widerstands und der Induktivität berechnet werden können.

Hinweis: Bei der Fehlerbehandlung sollte man sich auf das aktuell Notwendige beschränken. Nur offensichtliche und leicht zu behebende Probleme (z.B. Divisionen durch Null, unzulässige negative Zahlen) sollte man zusätzlich zum aktuell Notwendigen behandeln:

- Fehlerbehandlung kostet Zeit und damit Geld und
- führt mit großer Wahrscheinlichkeit zu neuen Fehlern!

← →

Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld_-_Probabilistik_-_Modell-Robustheit&oldid=22523“

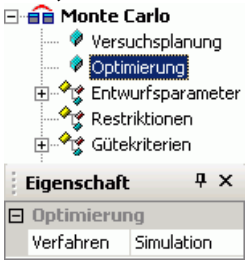
Software: FEM - Tutorial - Magnetfeld - Probabilistik - Monte-Carlo

Aus OptiYummy

↑

← →

Probabilistische Simulation - Verfahren mit Zufallszahlen (Monte Carlo)



Es existieren verschiedene Verfahren, wie man durch "Würfeln" Verteilungsdichten über die Streubreite der Parameter nachbilden kann. Man spricht hierbei auch von ***Sampling-Verfahren*** oder ***Monte-Carlo-Verfahren***. (Siehe "OptiY-Hilfe: Theoretische Grundlagen > Statistische Versuchsplanung > Sampling Verfahren). Wir werden uns hier auf das Rechenzeit-optimale Verfahren des ***Latin Hypercube Sampling*** beschränken.

Achtung:
Falls es noch nicht geschehen ist - man muss "Simulation" als Optimierungsverfahren wählen! Damit wird nur **eine** Stichprobe simuliert.

Inhaltsverzeichnis

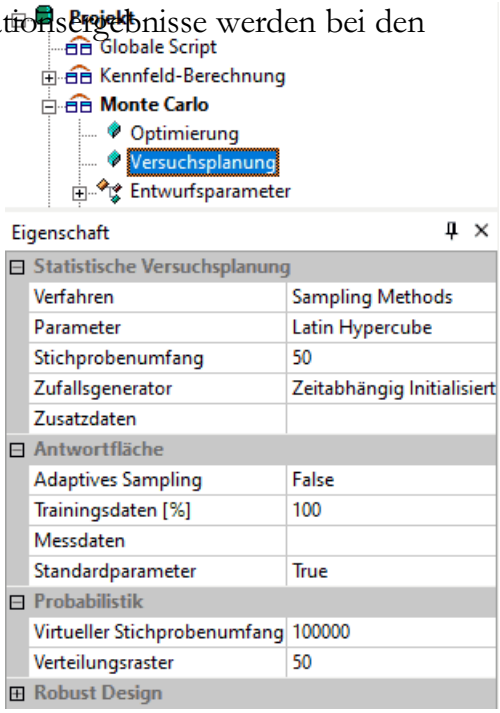
- 1 Versuchsplanung
- 2 Analyse
  - 2.1 Reale Stichprobe
    - 2.1.1 DOE-Tabelle
    - 2.1.2 Histogramme
    - 2.1.3 Korrelationsmatrix
    - 2.1.4 Anthill-Plot
  - 2.2 Virtuelle Stichprobe
    - 2.2.1 Verteilungen
    - 2.2.2 Sensitivitäten
  - 2.3 Antwortflächen (Response Surface)
    - 2.3.1 Residuum Plot
    - 2.3.2 Schnittdiagramm
    - 2.3.3 Neu Berechnen
    - 2.3.4 3D-Antwortfläche
    - 2.3.5 Koeffizienten der Ersatzfunktionen
    - 2.3.6 Modell-Export
- 3 Zusammenfassung

Versuchsplanung

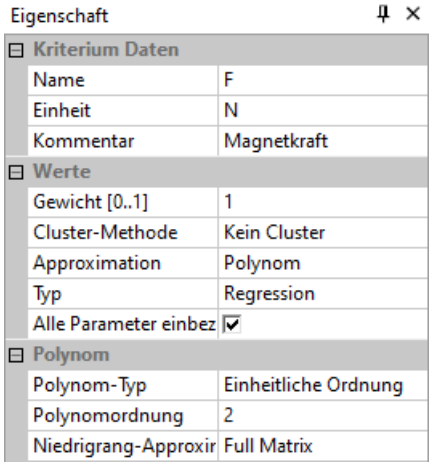
Das *Latin Hypercube Sampling* ist eine geeignete Sampling-Methode, um in unserem Beispiel mit akzeptablem Berechnungsaufwand hinreichend genaue und anschauliche Ergebnisse zu erhalten. Bei diesem Verfahren wird der gesamte Streubereich jedes Parameters in Intervalle unterteilt. Innerhalb dieser Intervalle werden entsprechend der Verteilungsdichte Werte "erwürfelt":

- Der gewählte Stichprobenumfang von 50 ist ein guter Kompromiss um einen Eindruck von dieser Methode zu erhalten.
- Der Zufallsgenerator soll im Beispiel Zufallszahlen in Abhängigkeit von der aktuellen Computer-Zeit liefern, d.h. bei jedem Experiment werden, wie in der Realität, etwas andere Ergebnisse entstehen!

Die Analyse-Diagramme zur Bewertung der probabilistischen Simulationsergebnisse werden bei den Sampling-Methoden nur teilweise auf Basis der wirklich berechneten Stichprobe generiert:



- Aus den mit dem Stichprobenumfang (im Beispiel =50) "erwürfelten" Werten der streuenden Parametern (hier die Luftspalte) und jeder daraus berechneten Bewertungsgröße des Modells (hier Magnetkraft und Koppelfluss) wird nach der Methode der kleinsten Fehlerquadrate eine Übertragungsfunktion approximiert.
- Für jede Bewertungsgröße (im Beispiel die Gütekriterien **F** und **Psi**) kann man unabhängig voneinander eine geeignete Funktion für die Approximation des Übertragungsverhaltens wählen. Wir verwenden hier "Polynom 2. Ordnung" für beide Größen.
- "Einheitliche Ordnung" bedeutet hierbei, dass die angegebene Ordnung für die Abhängigkeit von jeder Streugröße zu verwenden ist:



- Eine virtuelle Stichprobe (im Beispiel *Stichprobenumfang=100000*) wird auf Grundlage der zuvor approximierten Übertragungsfunktionen (= Ersatzmodell für das originale Simulationsmodell) generiert.
- Welche Punkte dieses Ersatzmodells für eine virtuelle Stichprobe benutzt werden, wird gleichfalls mit der für die reale Stichprobe gewählten Sampling-Methode "erwürfelt" (im Beispiel *Latin Hypercube*).
- Mit virtuellen Stichproben erhält man anschaulichere, "geglättete" Darstellungen auch bei einem kleinen realen Stichprobenumfang. Die Anzahl der *Verteilungspunkte* entspricht der Anzahl der Intervalle, in welche jeder Streubereich gleichmäßig geteilt wird, um stetige Verteilungsdichtefunktionen aus den diskreten Werten zu ermitteln.

Analyse

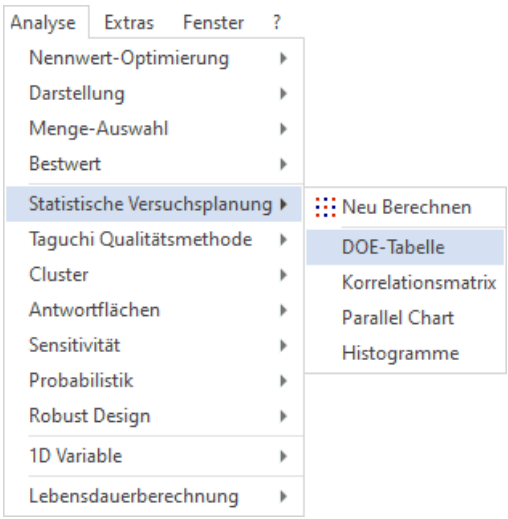
Reale Stichprobe

Alle Exemplare der realen Stichprobe werden auf Grundlage des Experiment-Workflows berechnet:

- Jedes Exemplar der Stichprobe ist gekennzeichnet durch seine konkreten Parameter-Werte (Nennwerte und Istwerte der Streugrößen) und die daraus resultierenden konkreten Bewertungsgrößen (Gütekriterien / Restriktionen).
- Bei der Abarbeitung des Workflows werden die in den Workflow eingebundenen Simulationsmodelle berechnet, was sehr zeitaufwändig sein kann.

- Die reale Stichprobe wird nur berechnet nach einem Start des Experiments bzw. dessen Weiter-Führung nach Stopp .

Bei der Nutzung von Sampling-Verfahren kann man bereits während der Simulation den Verlauf des Experiments beobachten:



- Wie in der Realität wird aus der gesamten Stichprobe ein Modell-Exemplar nach dem nächsten untersucht.
- Die generierten Werte der streuenden Parameter und die Ergebnisgrößen (Gütekriterien und Restriktionen) können für jede einzelne Modellrechnung mittels der bereitgestellten *Analyse*-Funktionen dargestellt werden.
- Die Darstellmöglichkeiten, welche sich nur auf die Ergebnisse der realen Stichprobe beziehen, findet man im Menü ***Analyse > Statistische Versuchsplanung***.
- Wir werden im Folgenden die einzelnen Darstellungen einzeln aktivieren.

DOE-Tabelle

Wir beginnen mit der DOE-Tabelle. In dieser werden alle Werte der realen Stichproben-Exemplare in der Reihenfolge ihrer Berechnung aufgelistet:

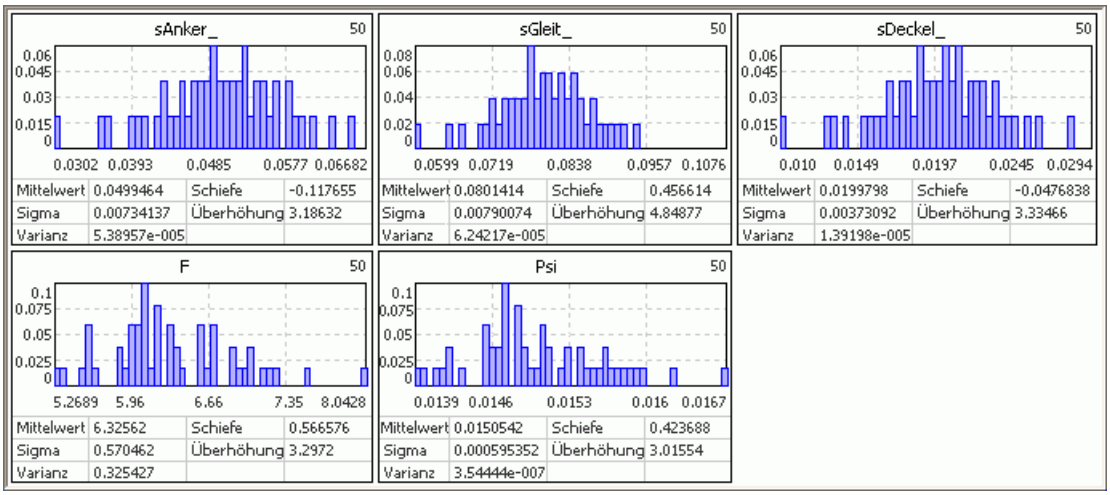
- *DOE="Design of Experiments" (Versuchsplanung).*
- Listet für jede Modellrechnung der echten Stichprobe (=1 Zeile) in den Spalten eine Auswahl von allen im Workflow definierten Größen auf.
- Über die Menü-Funktion ***Datei - Daten Export*** kann man diese umfangreichen Datensätze der DOE-Tabelle bei Bedarf in einer Excel-Datei weiter verarbeiten:

DOE-Table

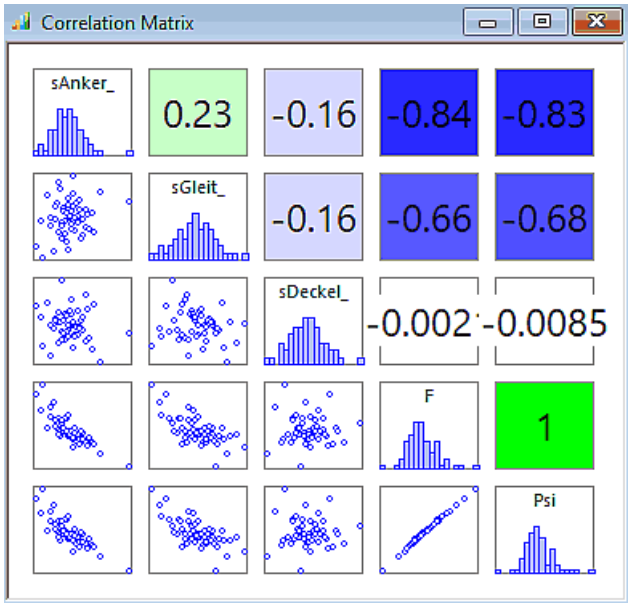
No	sAnker_	sGleit_	sDeckel_	F	Psi	Status
0	0.0585991935	0.0915750071	0.0197304332	5.44643878	0.014151526	Ok
1	0.0418084464	0.0744871012	0.0188495681	7.17113568	0.01596772	Ok
2	0.0425504579	0.0862392527	0.0202076247	6.52555026	0.0152851404	Ok
3	0.0538119828	0.0811795066	0.0210870205	6.04108803	0.0148145562	Ok
4	0.0471873924	0.0881050471	0.01545417	6.31105153	0.0150820033	Ok
5	0.0529441694	0.0796797223	0.0255016414	6.0330195	0.0147938569	Ok
6	0.0380659511	0.082222908	0.0143596672	7.21543958	0.0159833589	Ok
7	0.072201828	0.0829341284	0.0191339683	5.10144173	0.0138073193	Ok

Histogramme

Histogramme stellen die Häufigkeitsverteilung der abgebildeten Größen innerhalb des Streubereiches dar. Die Streugrößen der Luftspalte, die Kraft **F** und den Koppelfluss **Psi** ziehen wir mit *Drag & Drop* in das anfangs leere Histogramm-Fenster:



Korrelationsmatrix



Es wird für die reale Stichprobe die *Korrelation* zwischen allen Parameter-Streuungen und Bewertungsgrößen in Form von linearen Spearman Korrelationskoeffizienten dargestellt. Der Koeffizient **K** mit einem Bereich von -1 bis +1 ist auch farblich gekennzeichnet:

- |K|=0 → keine Korrelation mit der Toleranzgröße (weiß)
- |K|=1 → starke Korrelation mit der Toleranzgröße (dunkelblau).

Interpretation:

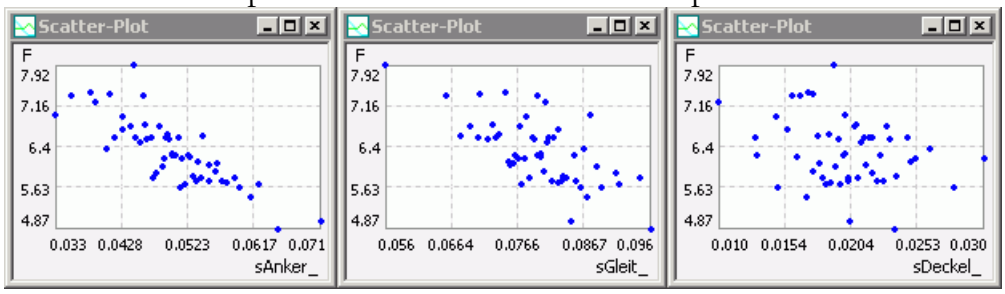
- Entlang der Diagonalen sind die einzelnen Streuungen als Histogramme eingetragen.
- Die 2D-Anthill-Plots unterhalb der Diagonalen stellen den Zusammenhang zwischen jeweils zwei Streuungen dar.
- Welche zwei Streuungen dies jeweils sind, ergibt sich durch Verfolgen der Spalte und Zeile bis zur Diagonalen.
- Spiegelbildlich zu den 2D-Anthill-Plots befinden sich oberhalb die zugehörigen Korrelationskoeffizienten.
- Falls die Bildung der Zufallszahlen gut funktioniert, darf keine Korrelation zwischen unterschiedlichen Parameter-Streuungen existieren (K=0). Auf Grund der kleinen Stichprobe ist ca. K<0.2. Die Korrelation zwischen streuenden Parametern hat insbesondere Bedeutung bei der Benutzung von Messwerten.
- Die Korrelation zwischen streuenden Parametern und Bewertungsgrößen ist abhängig vom Übertragungsverhalten des Modells.
- ***Achtung:*** Korrelation bedeutet nicht "kausale Abhängigkeit"! In technischen Anwendungen verbirgt sich dahinter aber meist eine Ursache-Wirkungs-Beziehung.

Anthill-Plot

Der "Ameisenhaufen" stand Pate für die Bezeichnung dieser Darstellform (Punktdiagramm), welche auch als *Streudiagramm* (engl. Scatterplot) bekannt ist. In OptiY existieren zwei Formen von Anthill-Plots (***Analyse > Cluster > Anthill-Plot***):

2D-Anthill-Plot:

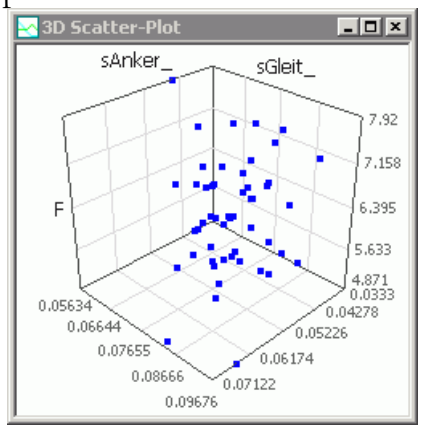
- Die X- und Y-Achse sind frei belegbar mit den im Workflow definierten Größen.
- Jedes Exemplar der realen Stichprobe wird durch einen Punkt repräsentiert:



- Je mehr die Punktwolke in einem dieser Scatter-Plots sich der Form einer Geraden annähert, desto stärker korrelieren die Werte der beiden dargestellten Größen. Im Beispiel korreliert die Magnetkraft **F** am stärksten mit dem Arbeitsluftspalt **sAnker**.
- Wenn man innerhalb der DOE-Tabelle eine Zeile mit Doppelklick auswählt (= Exemplar der Stichprobe), so wird der zugehörige Punkt in allen Anthill-Plots hervorgehoben und es werden daneben die "Koordinatenwerte" eingeblendet.

3D-Anthill-Plot:

- Es besteht auch die Möglichkeit, die Abhängigkeit einer Ergebnis-Größe (z.B. der Magnetkraft) von zwei Streu-Größen darzustellen.
- Jedes Exemplar der virtuellen Stichprobe wird ebenfalls durch einen Punkt repräsentiert:



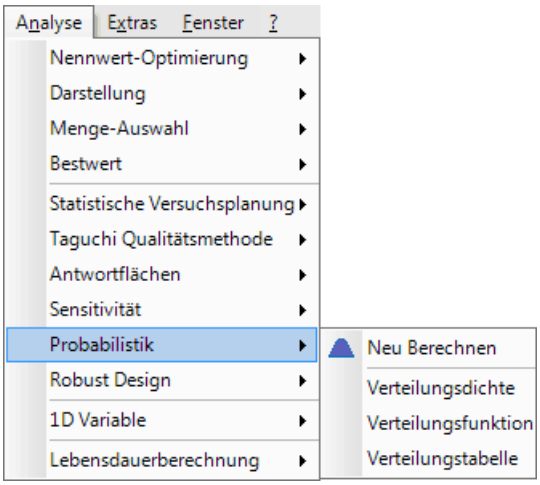
- Die X-, Y- und Z-Achsen dieses 3D-Scatter-Plot sind frei belegbar mit den im Workflow definierten Größen.
- Im obigen Beispiel wurde die Kraft als Funktion des Ankerluftspalts und des Führungsspalts dargestellt.

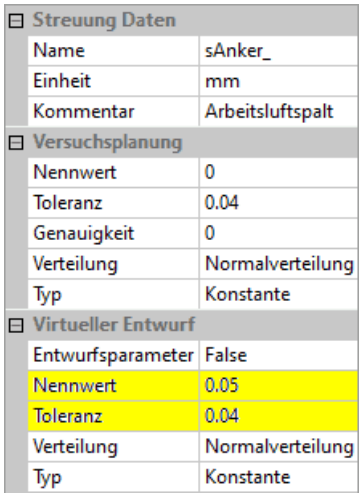
Virtuelle Stichprobe

- Für die Simulation einer virtuellen Stichprobe wird das Ersatzmodell genutzt, welches auf Basis der realen Stichprobe gebildet wird. Die Details der Ersatzmodell-Bildung betrachten wir später.
- Der Umfang einer virtuellen Stichprobe kann sehr groß gewählt werden, da die Ersatzmodelle um Größenordnungen schneller rechnen als die Originalmodelle.
- Damit können statistischen Aussagen zu den Ersatzfunktionen praktisch mit beliebiger Genauigkeit gewonnen werden. Die statistischen Ergebnisse zu den **Verteilungen** und **Sensitivitäten** können über ***Analyse > Probabilistik*** bzw. ***Analyse > Sensitivität*** abgerufen werden.
- Die Anzahl der virtuellen Exemplare wird vorgegeben durch den **"Virtuellen Stichprobenumfang"** in der Konfiguration der **Versuchsplanung**.
- Der virtuelle Nennwert (=Toleranzmittenwert) und die virtuelle Toleranz können unabhängig vom Toleranzmittenwert und Streubreite der realen Stichprobe gewählt werden. Da uns das Verhalten für

die aktuelle Magnetgeometrie interessiert, benutzen wir die gleichen Werte, wie in der realen Stichprobe.

- Mit dem Ersatzmodell kann man auch eine Optimierung der Streugrößen durchführen, dann müsste man *Entwurfsparameter=True* setzen. Wir untersuchen jedoch nur den Einfluss der Streuung bei konstantem virtuellen Nennwert:
  - **(virtueller) Nennwert:** Ist der aktuelle Toleranzmittenwert, der unabhängig vom Nennwert und dem Toleranzmittenabstand des realen Modells geändert werden kann.
  - **(virtuelle) Toleranz:** Bestimmt die wirksame Streubreite bei der Berechnung der virtuellen Stichprobe.
- Unabhängig von den Werten der realen Stichprobe kann man mit virtuellen Stichproben die Auswirkungen von Toleranz- und Mittenwert-Änderungen auf das Verhalten analysieren, ohne erneut aufwändige Modellberechnungen durchführen zu müssen.





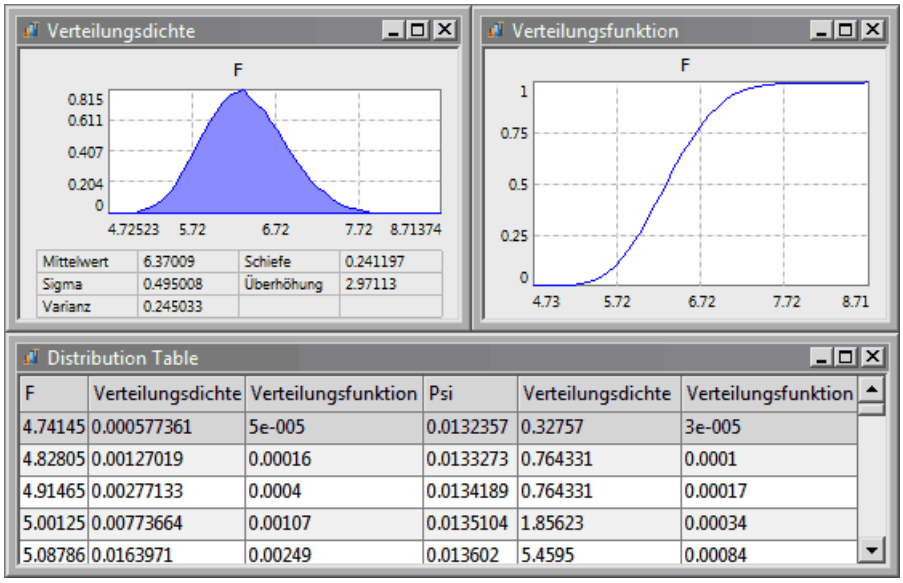
Streuung Daten	
Name	sAnker_
Einheit	mm
Kommentar	Arbeitsluftspalt
Versuchsplanung	
Nennwert	0
Toleranz	0.04
Genauigkeit	0
Verteilung	Normalverteilung
Typ	Konstante
Virtueller Entwurf	
Entwurfsparameter	False
Nennwert	0.05
Toleranz	0.04
Verteilung	Normalverteilung
Typ	Konstante

Wichtig:
Der Streubereich der virtuellen Stichprobe sollte den Streubereich der realen Stichprobe nicht verlassen. Da das Ersatzmodell nur für den Streubereich der realen Stichprobe ermittelt wurde, käme das einer (meist unzulässigen) Extrapolation des Modellverhaltens gleich!

Die Berechnung der virtuellen Stichprobe erfolgt automatisch nach Abschluss der Berechnung der realen Stichprobe oder nach (***Analyse > Probabilistik > Neu Berechnen***):

Verteilungen

Die Verläufe der Verteilungsdichten und der Verteilungsfunktionen der Ergebnisgrößen werden auf Basis der virtuellen Stichprobe ermittelt. Dabei wird der betrachtete Streu-Bereich der jeweiligen Ergebnisgröße standardmäßig in 50 Stützstellen unterteilt:

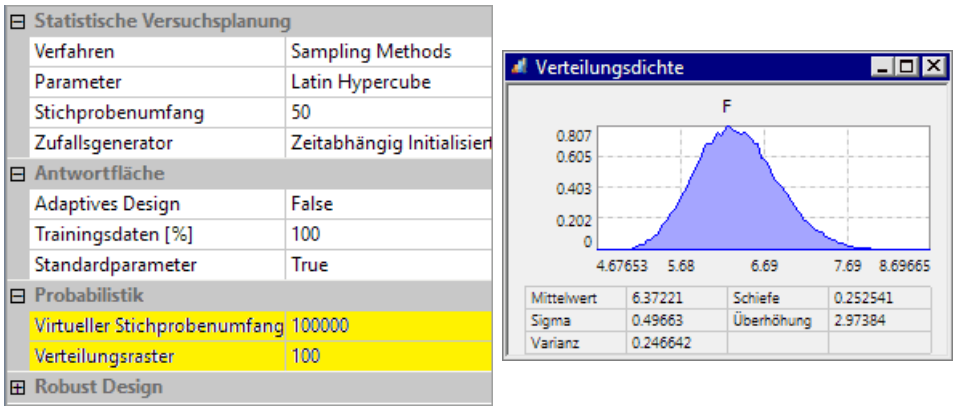


F	Verteilungsdichte	Verteilungsfunktion	Psi	Verteilungsdichte	Verteilungsfunktion
4.74145	0.000577361	5e-005	0.0132357	0.32757	3e-005
4.82805	0.00127019	0.00016	0.0133273	0.764331	0.0001
4.91465	0.00277133	0.0004	0.0134189	0.764331	0.00017
5.00125	0.00773664	0.00107	0.0135104	1.85623	0.00034
5.08786	0.0163971	0.00249	0.013602	5.4595	0.00084

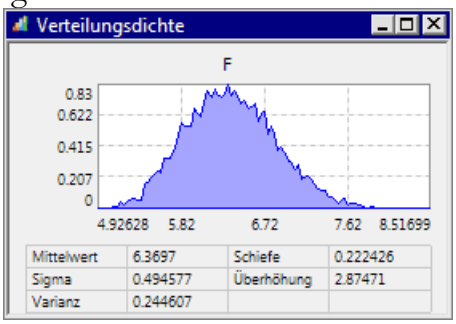
Die Verteilungstabelle enthält für alle Ergebnisgrößen die Werte der Verteilungsdichte und -funktion für alle Intervalle entsprechend der Zahl der Verteilungspunkte in der Versuchsplanung. Sie dient vor allem für den Datenexport, um diese Daten mit anderen Programmen weiter verarbeiten zu können.

Der Verlauf der Verteilungsdichte ist im Beispiel etwas unstetig. Dafür gibt es zwei Ursachen:

1. Mit steigender Anzahl der Verteilungspunkte (=Intervalle) wird dir Kurve unstetiger. Ändert man im Beispiel die Anzahl auf 100 und berechnet die Probabilistik neu, so ergibt sich der folgende Verlauf:



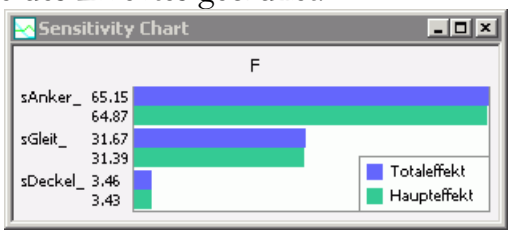
2. Mit steigendem Umfang der virtuellen Stichprobe wird die vom Ersatzmodell bewirkte Verteilungsdichte exakter berechnet. Verringert man den Umfang der virtuellen Stichprobe temporär um den Faktor 10 auf 10000, so ergibt sich mit 100 Intervallen der folgende Verlauf:



Sensitivitäten

Sensitivität-Chart:

- Es erscheint zuerst ein leeres Fenster, in das man die gewünschten Ausgangsgrößen (Restriktionen bzw. Gütekriterien) per *Drag&Drop* aus dem Explorer hineinziehen kann. Für jede dieser Ausgangsgrößen wird dann ein Pareto-Chart in Bezug auf alle Streuungen generiert:
- Unter *Pareto-Chart* versteht man ein Balkendiagramm (Histogramm), das anzeigt, in welchem Maße ein bestimmtes Ergebnis (Effekt) durch eine bestimmte Ursache (Streuung) hervorgerufen wurde. Die Balken sind nach der Größe des Effektes geordnet:



Den Pareto-Charts kann man zwei wesentliche Informationen entnehmen:

1. **Welche Streu-Größen haben einen vernachlässigbaren Einfluss auf die betrachteten Bewertungsgrößen?**
   Im Beispiel hat der Restluftspalt des Deckels nur einen sehr geringen Einfluss auf die Magnetkraft.

Damit könnte man z.B. für die Optimierung die Streuung dieses Luftspalts unberücksichtigt lassen. Das spart Rechenzeit!

2. **Existieren merkliche Interaktionen zwischen den Toleranzgrößen?**
   Wenn die aktuellen Ist-Werte der anderen Streuungen den Einfluss der zu betrachtenden Streugröße auf das Verhalten der Bewertungsgrößen merklich verändern, so gibt es Wechselwirkungen zwischen den Streugrößen.
   In den Pareto-Charts erkennt man das an dem Unterschied zwischen den Werten von *Total- und Haupteffekt*.

Existieren (wie im Beispiel) keine merklichen Wechselwirkungen zwischen den Streugrößen, so kann man bei Verwendung der im nächsten Abschnitt beschriebenen Momenten-Methode die probabilistische Simulation mit vereinfachten Ansätzen durchführen. Das spart Rechenzeit!

- **Haupteffekt:**
  Er repräsentiert den Haupteinfluss der betrachteten Streugröße Xi auf die Ausgangsgröße Y. Definiert ist er als Quotient aus der Varianz der durch Xi verursachten Streuung der Ausgangsgröße Var(Y|Xi) und der Varianz der durch alle Streugrößen **X** verursachten Streuung Var(Y|**X**)
  $$SH = Var(Y|Xi) / Var(Y|\mathbf{X})$$
- **Totaleffekt:**
  Er setzt sich zusammen aus dem Haupteffekt und den Interaktionen zwischen den einzelnen Streugrößen (Xi, Xj)
  $$ST = Var(Y|Xi) / Var(Y|\mathbf{X}) + Var(Y|Xi,Xj)/Var(Y|\mathbf{X})$$

  In OptiY wird die Interaktion durch paarweise Kombination aller Streugrößen berücksichtigt, da die gleichzeitige Berücksichtigung sämtlicher Streugrößen zu einem nicht beherrschbaren Berechnungsaufwand führt. Jedes Paar (Xi, Xj) wird als ein Glied dieser Summenformel berücksichtigt. Der Wert dieses Gliedes ist jeweils Null, wenn es keine Interaktion innerhalb des Streugrößen-Paares gibt.

Sensitivität-Tabelle:

- Diese Tabelle zeigt einen kompletten Überblick über die Werte von Haupt- und Totaleffekt der Streuungen auf alle Ergebnisgrößen.
- Über den Kopf der Tabelle kann man die Zeilen nach den unterschiedlichsten Kriterien sortieren:

Sensitivitäten-Tabelle

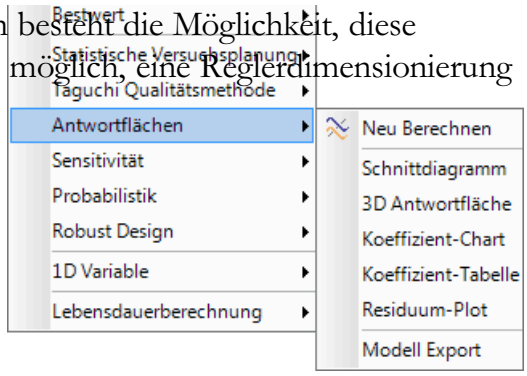
Streuungen	F-Haupteffekt	F-Totaleffekt	Psi-Haupteffekt	Psi-Totaleffekt
sAnker_	64.8711	65.1519	59.3685	59.4509
sGleit_	31.3978	31.6793	36.057	36.1419
sDeckel_	3.43891	3.461	4.48051	4.50126
sAnker_*sGleit_	0.270092	0.270092	0.0732494	0.0732494
sAnker_*sDeckel_	0.0107373	0.0107373	0.0091642	0.0091642
sGleit_*sDeckel_	0.0113589	0.0113589	0.01158	0.01158

- Zusätzlich zu den Effekten der Streuungen auf die Ausgangsgrößen wird in der Tabelle auch der Effekt der Interaktionen zwischen den Streugrößen auf die Ausgangsgrößen dargestellt.

Antwortflächen (Response Surface)

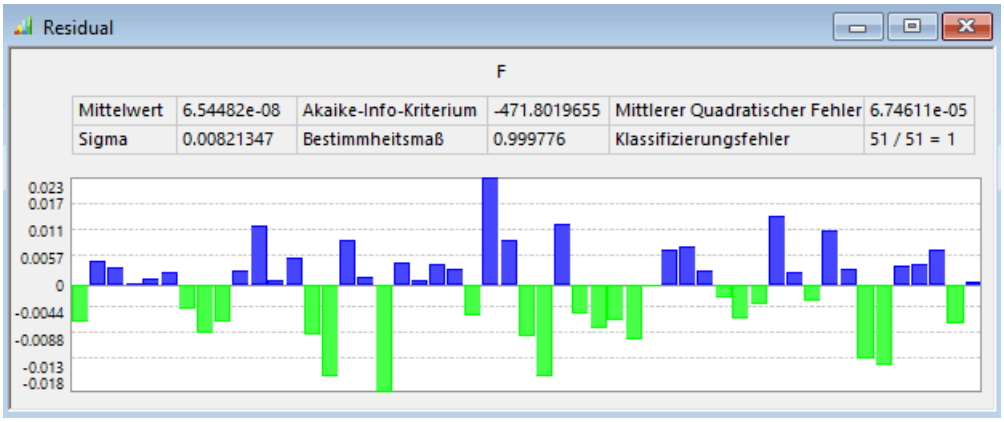
In OptiY finden die probabilistischen Analysen auf Grundlage der approximierten Ersatzmodelle (Übertragungsfunktionen der Ausgangsgrößen) statt. Anhand dieser Übertragungsfunktionen, welche praktisch Flächen im n-Dimensionalen Parameter-Raum darstellen, wird die System-Antwort auf eingespeiste Parameter berechnet. Daraus resultiert die Bezeichnung ***Response Surface Method*** (**RSM**):

Neben verschiedenen Darstellungen dieser Übertragungsfunktionen besteht die Möglichkeit, diese Funktionen z.B. als C-Quelltext zu exportieren. Damit wäre es z.B. möglich, eine Reglerdimensionierung mit einem vereinfachten, schnell rechnenden Ersatzmodell vorzunehmen.



Residuum Plot

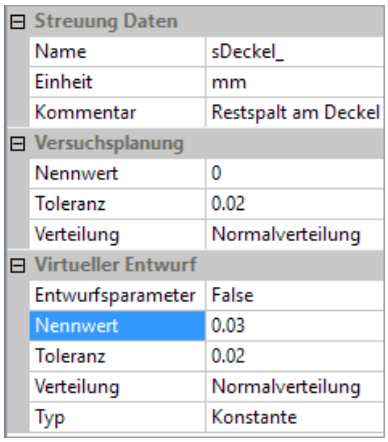
- Eigentlich interessieren den Anwender nicht die Aussagen zur Ersatzfunktion, sondern die Eigenschaften des zu untersuchenden Originals. Die Genauigkeit der statistischen Aussagen in Bezug auf das Originalmodell wird überwiegend durch die Genauigkeit der Approximationsfunktionen bestimmt.
- Die Residuen der Ausgleichsrechnung für eine vorliegende reale Stichprobe zeigen nur, wie genau die Ausgleichsfläche in die vorhandene "Punktwolke" berechneten realen Exemplare passt (***Analyse > Antwortflächen > Residuum-Plot*** mit *Drag&Drop* von Kriterium/Restriktion).
- ***Achtung***: Nach einer Überarbeitung der Residuen-Darstellung scheint es ein Problem in der aktuellen OptiY-Version zu geben - vor der Neuberechnung der Ergebnisgrößen erscheinen dann zwar die leeren Diagramme für die darzustellenden Restriktionen/Gütekriterien, aber nach Berechnung der Antwortfläche verschwinden diese wieder und das Fenster ist komplett leer!



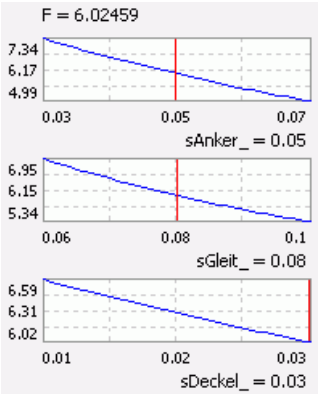
- Residuen sind absolute Differenzen zwischen den "echten" Modellberechnungen und den aus dem Ersatzmodell berechneten Punkten. Im Diagramm kann man den Betrag der maximalen Differenz erkennen. Die Residuen geben damit die Qualität der Approximation an die berechnete Punktwolke wieder. Bei einem Mittelwert der Magnetkraft von 6.3 N sind im Beispiel Abweichungen von max. ca. 0.02 N wahrscheinlich vernachlässigbar!
- Informationen zur Genauigkeit bzw. Sinnfälligkeit der Ausgleichsfläche in den Zwischenräumen der Punktwolke sind damit nicht zu gewinnen.

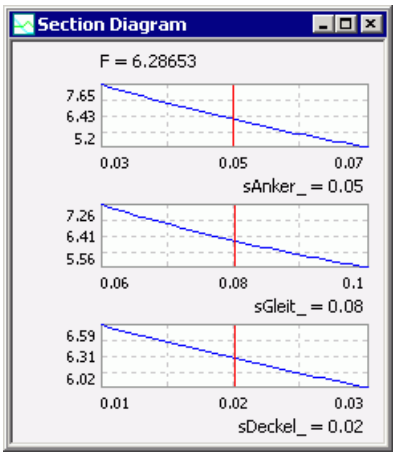
Schnittdiagramm

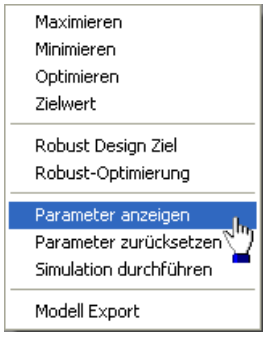
- In einem Schnittdiagramm kann man für die ausgewählten Ergebnisgrößen (im Beispiel die Kraft) den Einfluss ausgewählter Streugrößen (hier der Luftspalte) analysieren.
- Betrachtet man die Abhängigkeit der Magnetkraft ***F(sAnker_)***, so gilt die eingezeichnete Kurve nur für die aktuellen Werte aller anderen Streugrößen (roter Strich).
- Ändert man im OptiY-Explorer z.B. den aktuellen Wert des Deckelspalts auf 30 µm, so ändern sich die Kurvenverläufe in den anderen Schnittdiagrammen zu kleineren Kraftwerten:



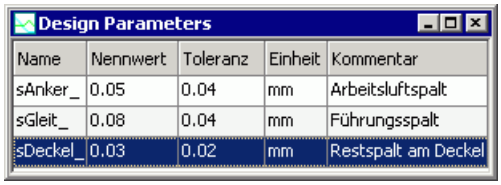
Streuung Daten	
Name	sDeckel_
Einheit	mm
Kommentar	Restspalt am Deckel
Versuchsplanung	
Nennwert	0
Toleranz	0.02
Verteilung	Normalverteilung
Virtueller Entwurf	
Entwurfsparameter	False
Nennwert	0.03
Toleranz	0.02
Verteilung	Normalverteilung
Typ	Konstante







- Das Kontext-Menü des Schnittdiagramms (Rechtsklick auf ein Schnittdiagramm) bietet zahlreiche Möglichkeiten zur Arbeit mit diesen approximierten Ersatzfunktionen:
  - Damit man nicht den Überblick verliert, welche Istwerte aktuell für alle Streugrößen eingestellt sind, kann man diese z.B. in einer Tabelle als ***Parameter anzeigen*** lassen:



Design Parameters

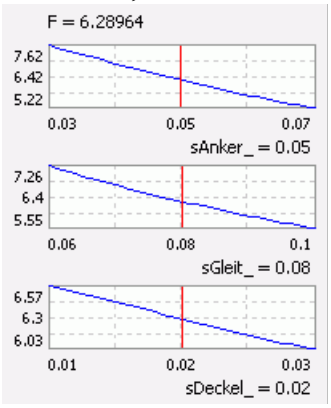
Name	Nennwert	Toleranz	Einheit	Kommentar
sAnker_	0.05	0.04	mm	Arbeitsluftspalt
sGleit_	0.08	0.04	mm	Führungsspalt
sDeckel_	0.03	0.02	mm	Restspalt am Deckel

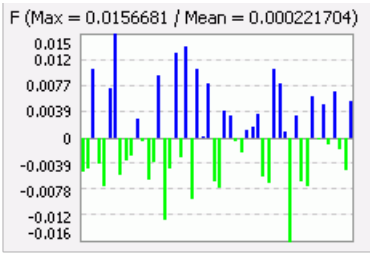
  - Sinnvoll ist auch das ***Parameter zurücksetzen*** auf ihre ursprünglichen Werte.
  - Für die aktuellen Werte kann man eine ***Simulation durchführen*** mit dem "echten" Workflow-Modell.

Neu Berechnen

Falls man nicht sicher ist, ob der für die Ergebnisgrößen gewählte Funktionsansatz bzw. die Ordnung der Approximation hinreichend sind, kann man z.B. eine höhere Polynom-Ordnung wählen und die Antwortfläche neu berechnen :

- Dabei wird ohne erneute Modell-Berechnung wieder die echte Stichprobe benutzt.
- Im Beispiel verbessert für die Magnetkraft **F** eine **Erhöhung der Polynom-Ordnung von 2 auf 3** die Anpassung der Ersatzfunktion, was man auch an kleineren Werten im *Residuum Plot* erkennt:



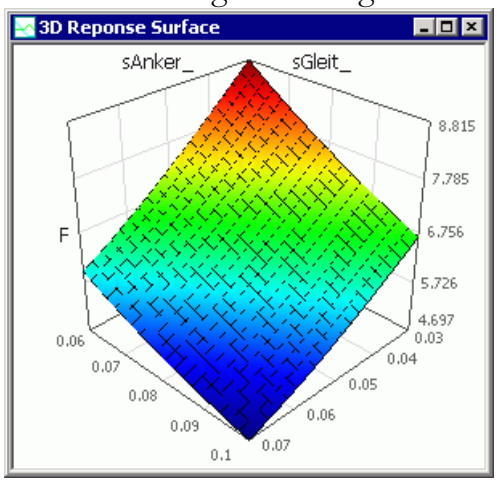


- Nachträglich muss man dann die Neuberechnung der virtuellen Stichprobe veranlassen. Erst danach werden die Verteilungen und Sensitivitäten in den Probabilistik-Diagrammen aktualisiert!

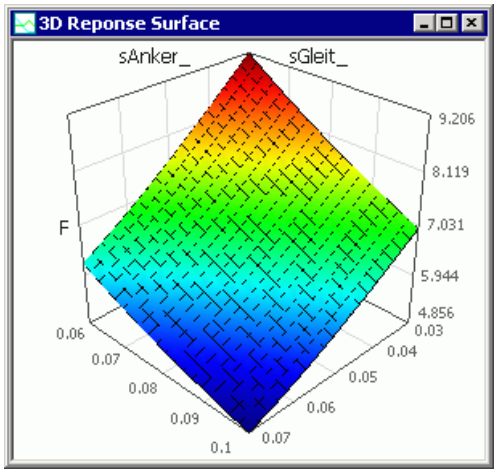
3D-Antwortfläche

- Hier handelt es sich praktisch auch um ein Schnitt-Diagramm. Allerdings wird die ausgewählte

Ergebnisgröße hier in Abhängigkeit von 2 Streugrößen dargestellt:

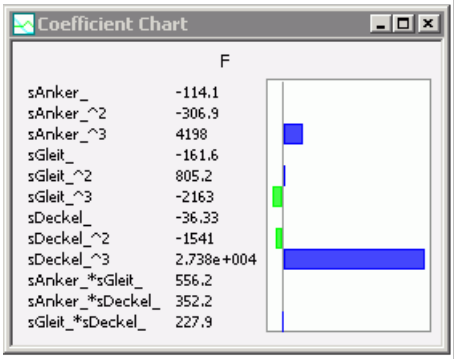


- Die dargestellte Übertragungsfunktion ist nur gültig für die aktuellen Werte aller Streu-Größen. Ändert man die aktuellen Werte, so wird die 3D-Antwortfläche aktualisiert. Verringert man z.B. den aktuellen Wert des Luftspalts **sDeckel_** von **20 µm** auf **10 µm**, so entsteht insgesamt eine etwas höhere Magnetkraft:



Koeffizienten der Ersatzfunktionen

- Für jede Ergebnisgröße wird eine individuelle Ersatzfunktion approximiert (z.B. Polynom), welche die Abhängigkeit von allen Streugrößen beschreibt. Auf die Koeffizienten jeder dieser Ersatzfunktionen kann man zugreifen (***Koeffizient-Chart*** und ***Koeffizient-Tabelle***). Damit könnte man sich bei Bedarf das approximierte Ersatzmodell in einer anderen Umgebung aufbauen:



Coefficient

Coefficient	F	Coefficient	Psi
1	19.3062	1	0.0267533
sAnker_	-114.121	sAnker_	-0.117526
sAnker_^2	-306.877	sAnker_^2	0.26233
sAnker_^3	4197.6	sGleit_	-0.107229
sGleit_	-161.637	sGleit_^2	0.240626
sGleit_^2	805.155	sDeckel_	-0.0739978
sGleit_^3	-2163.11	sDeckel_^2	0.21558
sDeckel_	-36.334	sAnker_*sGleit_	0.321694
sDeckel_^2	-1541.17	sAnker_*sDeckel_	0.227572
sDeckel_^3	27381.3	sGleit_*sDeckel_	0.255815
sAnker_*sGleit_	556.171		
sAnker_*sDeckel_	352.198		
sGleit_*sDeckel_	227.938		

- Die Koeffizient-Tabelle kann man, wie jede andere Tabelle auch, bei Bedarf z.B. als Excel-Tabelle exportieren (***Datei > Daten Export***).
- Im Beispiel erkennt man die unterschiedliche Polynom-Ordnung für die Ergebnisgrößen **F** und **Psi**.

Modell-Export

Einfacher als mit dem Daten-Export der Koeffizienten-Tabelle erhält man durch ***Modell-Export*** sofort einen C-, Modelica-, Visual Basic- oder Matlab-Quelltext des approximierten Ersatzmodells:

- Diese Funktion steht nur in der **OptiY Professional Edition** zur Verfügung!

Zusammenfassung

Der Vorteil der Sampling-Methode besteht darin, dass mit hinreichend großem Stichprobenumfang beliebige nichtlineare Zusammenhänge zwischen den Streu- und Ergebnisgrößen statistisch erfasst werden können. Von Nachteil ist, dass ein "hinreichend" großer Stichprobenumfang eine sehr große Anzahl vom Exemplaren umfasst (>>1000). Diesen Nachteil kann man durch die *Response Surface Methode* mildern:

- Im OptiY können z.B. **Polynome** beliebiger Ordnung als Approximationsfunktion für jede Ergebnisgröße verwendet werden:
  [math]\displaystyle{ P(x) = \sum_{i=0}^n a_ix^i = a_nx^n + a_{n-1}x^{n-1} + \cdots + a_2x^2 + a_1x + a_0 }[/math]
- Durch Anwendung der **Methode der kleinsten Fehlerquadrate** werden die Parameter der Polynome so bestimmt, dass die Approximationsfunktionen möglichst gut in die Punktwolke der "gesampelten" Stichprobe passen.
- Die minimal erforderliche Anzahl der "echten" Modellberechnungen **M** (=Stichprobengröße) für die Bildung der Ersatzfunktionen ergibt sich aus der Anzahl **n** der stochastischen Variablen und der gewählten Ordnung **O** der Polynom-Funktion zu **$M=(n^2-n)/2+O*n+1$**.
  Im Beispiel ist M=10 bei n=3 und O=2, was auch mit umfangreichen Modellen noch akzeptable Rechenzeiten ergeben kann.
- Die minimale Stichprobengröße führt jedoch auf Grund der geringen Stützstellendichte nur dann zu einer hinreichend genauen Ersatzfunktion zwischen den Stützstellen, wenn mit dem gewählten Funktionsansatz überhaupt eine sinnvolle Nachbildung des Modellverhaltens möglich ist.
- Die eigentliche statistische Analyse wird mit einer virtuellen Stichprobe sehr großen Umfangs auf Basis des zuvor approximierten Ersatzmodells durchgeführt. Die erreichbare Genauigkeit ist demzufolge nur noch von der Genauigkeit des Ersatzmodelles abhängig!

← →

Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld_-_Probabilistik_-_Monte-Carlo&oldid=25268“

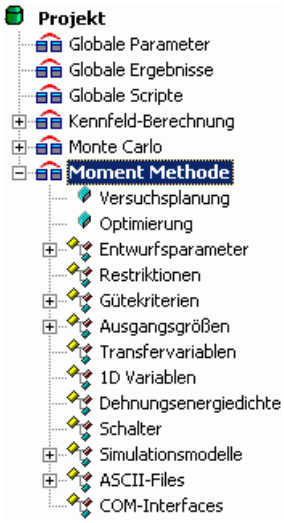
Software: FEM - Tutorial - Magnetfeld - Probabilistik - Momenten-Methode

Aus OptiYummy

↑

← →

Probabilistische Simulation - Momenten-Methode



Vergleichend zur Toleranzsimulation mit der Sampling-Methode Latin ***Hypercube Sampling*** soll nun der analytische Ansatz der ***Moment-Methode*** benutzt werden:

- Durch Duplizieren gewinnen wir aus dem Zufallszahlen-Experiment die Grundlage für die Konfiguration eines neuen Experiments und wählen dieses als Startup-Experiment.
- Auch für das neue Experiment vergeben wir einen sinnvollen Namen.
- In dem neuen Experiment müssen wir die Versuchsplanung entsprechend umkonfigurieren und die Darstellung von Ergebnissen neu organisieren.

Moment-Methode

Der erste Schritt der Moment-Methode verläuft ähnlich wie bei den Sampling-Methoden - es wird ein Ersatzmodell gebildet:

- Für jede Ergebnis-Größe (Gütekriterien und Restriktionen) des Workflows wird eine Funktion ***f*** gebildet, mit welcher sich der Wert der Ergebnis-Größe aus dem Variablenvektor ***x*** der **n** Streu-Größen berechnen lässt.
- Für die Approximation jeder dieser Funktionen ***f*** wird im Unterschied zur Sampling-Methode eine Taylor-Reihe benutzt. Ihre Approximationsordnung ist dabei auf den Maximalwert 2 beschränkt (*First Order=1 / Second Order=2*).
- ***Hinweis:*** Bei Wahl der Momenten-Methode zur statistischen Versuchsplanung sind die Einstellungen der Gütekriterien in Hinblick auf die Approximation wirkungslos!
- Wir nutzen zuerst die ***Second Order Analysis*** (Ordnung=2) mit Berücksichtigung von Interaktionen (Wechselwirkungen) zwischen den Streu-Größen. Die vollständige Funktion dafür lautet:

$$f = f_0 + \sum_{j=1}^{n} \frac{\partial f}{\partial x_j}(x_j - x_0) + \frac{1}{2}\sum_{j=1}^{n} \frac{\partial^2 f}{\partial x_j^2}(x_j - x_0)^2 + \sum_{j=1}^{n-1}\sum_{k=j+1}^{n} \frac{\partial^2 f}{\partial x_j \partial x_k}(x_j - x_0)(x_k - x_0)$$

Die erforderlichen Ersatz-Übertragungsfunktionen werden durch die Berechnung von Stützstellen gewonnen:

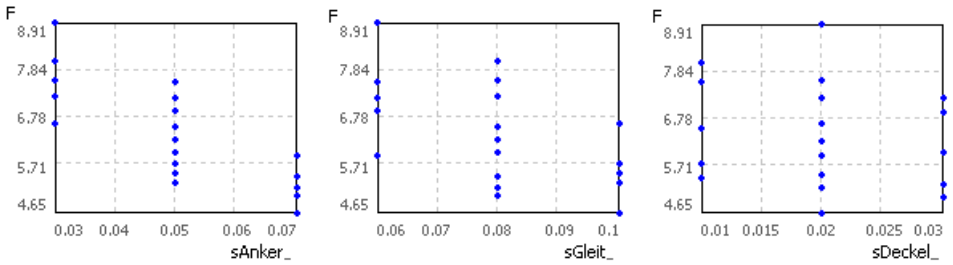
- Pro Streu-Größe $\mathbf{S_i}$ werden nur drei Stützstellen genutzt (Grenzwerte und Mittelwert)
- Bedingt durch die kombinatorische Abtastung des Modells steigt die benötigte Anzahl der Modell-Läufe quadratisch mit der Anzahl der Streu-Größen **n** (Simulationsläufe = $\mathbf{2 \cdot n^2 + 1}$).
- Diese Art der Gewinnung eines Ersatz-Modells führt nur zu einem hinreichend genauen Ergebnis, wenn für das zu approximierende reale Modellverhalten Taylor-Reihen 2. Ordnung ausreichend sind. Zum Glück ist dies für die Mehrzahl der Anwendungsfälle zutreffend.

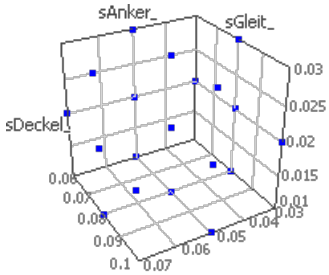
Nach der Berechnung des Ersatz-Modells (Menge von Taylor-Reihen) erfolgt die eigentliche Analyse wieder auf Basis dieses Ersatz-Modells. Hier existiert nun der entscheidende Unterschied zu den Sampling Methoden:

- Die statistischen *Momente* der Ausgangsgrößen werden analytisch näherungsweise aus den statistischen Momenten der Eingangsgrößen berechnet. Aus den berechneten Momenten werden anschließend die Verteilungen der Ausgangsgrößen approximiert.
- Das Verfahren der *Second Order Analysis* arbeitet sehr genau, wenn das Verhalten der Ausgangsgrößen im Streu-Bereich höchstens quadratische Abhängigkeiten zu den Streu-Größen aufweist. Die Ergebnisse mit 4 Streu-Größen sind dann vergleichbar mit einer Monte-Carlo-Simulation bei einer Stichprobengröße von 100.000. Da das Verfahren ohne Zufallszahlen arbeitet, ist es numerisch sehr stabil und erlaubt auch eine schnelle Optimierung unter Berücksichtigung von Streuungen.
- Der Nachteil der *Second Order Analyse* liegt in dem hohen Rechenaufwand bei einer großer Anzahl von Streu-Größen (Simulationsläufe = $\mathbf{2 \cdot n^2+1}$).

Analyse

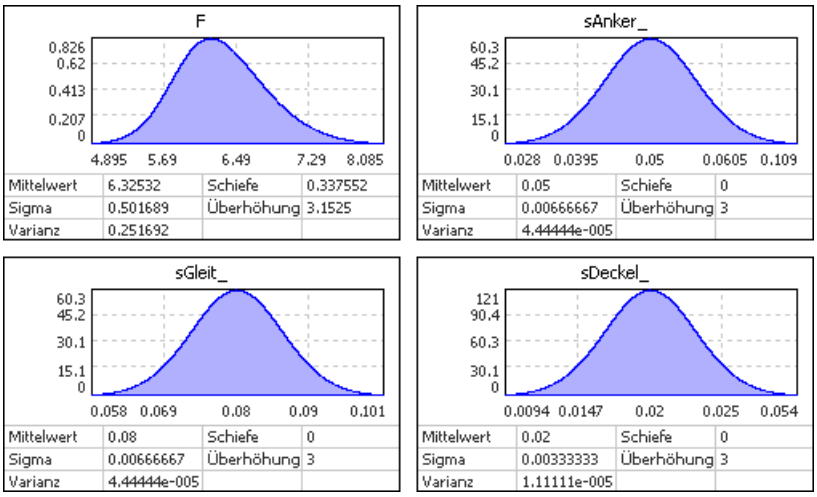
- Nach Start der Simulation erfolgen zuerst die Modellrechnungen für die Gewinnung des Ersatz-Modells. Erst danach werden die Verteilungen der Ausgangsgrößen berechnet. Die Ergebnisse der Probabilistik stehen also erst nach allen Modellberechnungen zur Verfügung.
- Man kann dies sehr gut beobachten, indem man vor dem Start der Simulation zusätzlich zu den Verteilungsdichten der Streuungen auch die 2D-Anthill-Plots *F(sAnker_)*, *F(sGleit_)* und *F(sDeckel_)* öffnet:





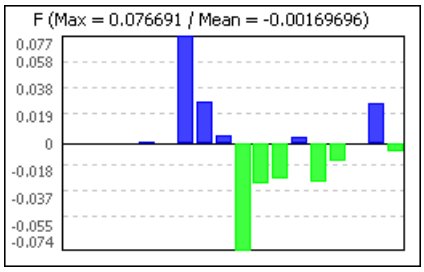
Während der 19 Abtastberechnungen werden schrittweise die Punkte in den Anthill-Plots abgebildet. Anhand der senkrechten Linien erkennt man deutlich die Abtastungen in der Mitte und an den Rändern jeder Streu-Größe.
Damit wird der gesamte Streu-Raum sehr gleichmäßig abgetastet, wie man sehr gut im 3D-Anthill-Plot erkennt.

Erst nach der kompletten Abtastung erscheinen die Verteilungen:



Hinweis: Histogramme stehen bei der Moment-Methode nicht zur Verfügung!

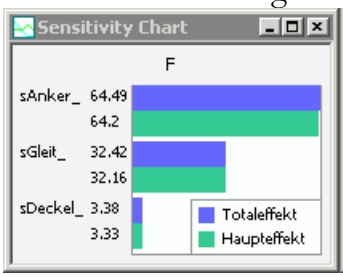
Die Genauigkeit des approximierten Modells an den Abtastpunkten kann man wie bei der Sampling-Methode über das Residual-Diagramm beurteilen (***Analyse > Antwortflächen > Residuum-Plot***):



- An zwei Abtaststellen ist die Abweichung im Beispiel um ca. den Faktor 4-5 höher, als die Maximalabweichung bei der Sampling-Methode. Da diese Abweichung des Ersatzmodells aber nur bei ca. 1% im Vergleich zur realen Modellberechnung liegt, kann man das akzeptieren.
- ***Achtung:***
  Die Genauigkeit an den Abtaststellen der Moment-Methode sagt nichts über die Genauigkeit des Ersatz-Modells zwischen den Abtaststellen aus! Ob die gewählte Ordnung für die Taylor-Reihen ausreichend ist, kann man nur mit einer hinreichend großen Stichprobe in der Sampling-Methode überprüfen.

Nun ist sicher interessant, ob man in Hinblick auf den Einfluss der Streuungen zum gleichen Ergebnis kommt, wie bei der Sampling-Methode:

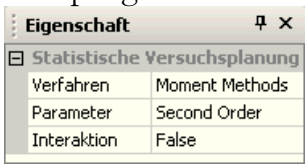
- Korrelationsmatrix und -tabelle stehen bei der Momenten-Methode nicht zur Verfügung.
- Sensitivität-Chart und -Tabelle zeigen den Effekt der Streuungen auf die Ausgangsgrößen:



Sensitivitäten-Tabelle

Streuungen	F-Haupteffekt	F-Totaleffekt	Psi-Haupteffekt	Psi-Totaleffekt
sAnker_	64.203	64.492	59.9519	60.0585
sGleit_	32.164	32.4289	35.7053	35.8036
sDeckel_	3.33085	3.38118	4.23053	4.25015
sAnker_*sGleit_	0.251781	0.251781	0.0926757	0.0926757
sAnker_*sDeckel_	0.0372335	0.0372335	0.0139374	0.0139374
sGleit_*sDeckel_	0.013093	0.013093	0.00568478	0.00568478

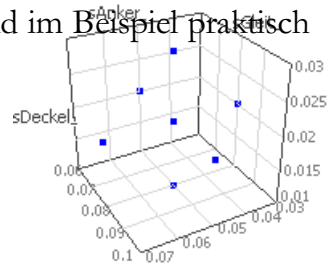
- Bis auf winzige Abweichungen erhält man die gleichen Ergebnisse, wie bei der Sampling-Methode.



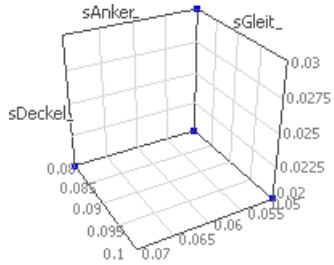
Die praktische Gleichheit von Haupt- und Totaleffekt bedeutet, dass es keine Interaktionen (Wechselwirkungen) zwischen den Streugrößen gibt. Mit dieser Erkenntnis sollte man die Versuchsplanung entsprechend ändern:

- Die Auswirkung auf die Berechnungszeit einer Stichprobe ist enorm, da der Großteil der Abtastschritte für die Berücksichtigung der Interaktionen benötigt wird. Die benötigte Anzahl der Modell-Läufe steigt nun nur noch linear mit der Anzahl der Streu-Größen **n** (Simulationsläufe = **2·n+1**).
- Im Beispiel verringert sich die erforderliche Zahl von Modellrechnungen von **19** auf **7**, was man im 3D-Anthill-Plot gut beobachten kann.

- Die mit dieser "reduzierten" Second-Order-Methode erzielten Ergebnisse sind im Beispiel praktisch identisch, da es keine Wechselwirkungen zwischen den Streu-Größen gibt.



Hinweis: Die First-Order-Approximation ohne Interaktion erfordert nur **n+1** Modellabtastungen. Allerdings werden damit nur Hyperebenen (ohne Krümmung) zwischen den abgetasteten Ecken des Streu-Raumes interpoliert:



- Das ist im Beispiel zwar nicht ganz exakt, aber die damit ermittelten Ergebnisse sind nicht wesentlich andere.
- Besitzt man nur ein FEM-Modell mit sehr langen Berechnungszeiten, so sollte man in diesem Fall die Stichproben-Simulation innerhalb von Optimierungsexperimenten mit diesem First-Order-Ansatz durchführen!

← →

Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld_-_Probabilistik_-_Momenten-Methode&oldid=22530“

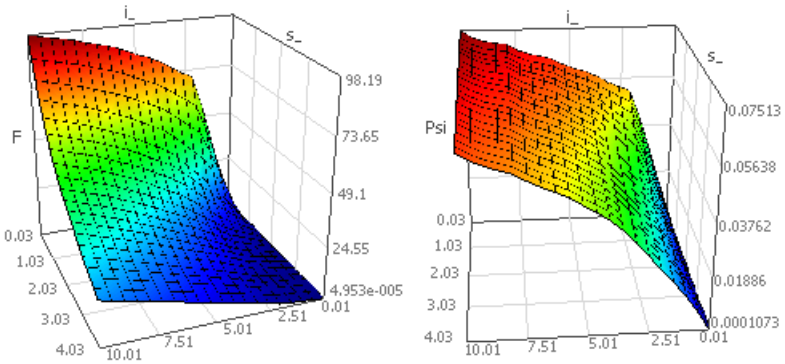
Software: FEM - Tutorial - Magnetfeld - Probabilistik - Kennfeld-Identifikation

Aus OptiYummy

↑

← →

Identifikation von Ersatzmodellen



Die Kraft- und Koppelfluss-Kennfelder eines E-Magneten kann man in Modellen der System-Simulation als Ersatzmodell für eine konkrete Wandler-Geometrie benutzen:

1. Die Ergebnisse einer FEM-Rastersuche kann man als Daten exportieren und die berechneten Abtaststellen des FEM-Modells als Stützstellen von 3D-Funktionsflächen in der System-Simulation verwenden. Aus diesen Stützstellen werden in der System-Simulation dann die Funktionswerte der Zwischenräume interpoliert.
2. Die für die Durchführung der virtuellen Stichprobe von OptiY generierten mathematischen Funktionen der Antwortflächen kann man aber auch direkt als Modell-Code in Modelle der System-Simulation implementieren. Die Gewinnung von Ersatzmodellen nach diesem Prinzip wollen wir abschließend in diesem Übungskomplex durchführen.

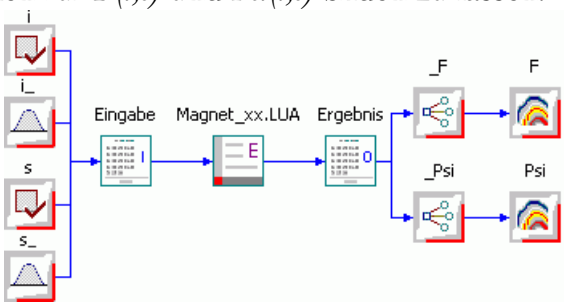
Ausgehend vom Raster-Experiment **Kennfeld-Berechnung** konfigurieren wir nach dem Duplizieren ein Experiment **Kennfeld-Identifikation**.

Inhaltsverzeichnis

- 1 Workflow
- 2 Versuchsplanung
- 3 Approximation mit Polynom-Ansatz
- 4 Approximation mit Gauss-Prozess
- 5 Hinweise

Workflow

- Wir werden die probabilistische Simulation benutzen, um über den gesamten Arbeitsbereich unseres E-Magneten die Antwortflächen für *F(i,s)* und *Psi(i,s)* bilden zu lassen:



- Die **Nennwerte** von Strom und Arbeitsluftspalt sind nun konstante Größen. Diese **Streuungsmittenwerte** sind so zu wählen, dass über den gesamten Streubereich numerisch instabile Werte nahe Null vermieden werden:
  - i.Wert=5.01 A (i≥0.01 A)
  - s.Wert=2.03 mm (s≥30 µm)
- Die **Streuungen** von Strom und Arbeitsluftspalt umfassen den gesamten Arbeitsbereich des E-Magneten. Die **Gleichverteilung** innerhalb der Streu-Bereiche gewährleistet eine gleichmäßige Abtastung, falls man eine Sample Methode verwendet:
  - i_.Toleranz=10 A
  - s_.Toleranz= 4 mm
  - i_.Nennwert und s_.Nennwert Null (Toleranzmittenabstand=0)

Streuung Daten	
Name	i_
Einheit	A
Kommentar	Breite Strom
Versuchsplanung	
Nennwert	0
Toleranz	10
Genauigkeit	0
Verteilung	Gleichverteilung
Typ	Zufall
Virtueller Entwurf	
Entwurfsparameter	☐
Nennwert	5.01
Toleranz	10
Verteilung	Gleichverteilung
Typ	Zufall

- Die beiden anderen Luftspalte setzen wir manuell in der Input-Datei auf ihre Nennwerte:
  - **sDeckel** = 20 µm
  - **sGleit** = 80 µm

Versuchsplanung

Wir werden dieses Simulationsexperiment (**Optimierungsverfahren=Simulation**) so konfigurieren, dass wir möglichst anschaulich erkennen, wie genau die approximierten Antwortflächen die wirklichen Übertragungsfunktionen des Modells abbilden:

Statistische Versuchsplanung	
Verfahren	Factorial Design
Parameter	Full Factorial
Stufen	6
Antwortfläche	
Adaptives Design	False
Trainingsdaten [%]	100
Standardparameter	True
Probabilistik	
Virtueller Stichprobenumfang	0
Verteilungsraster	50
Robust Design	

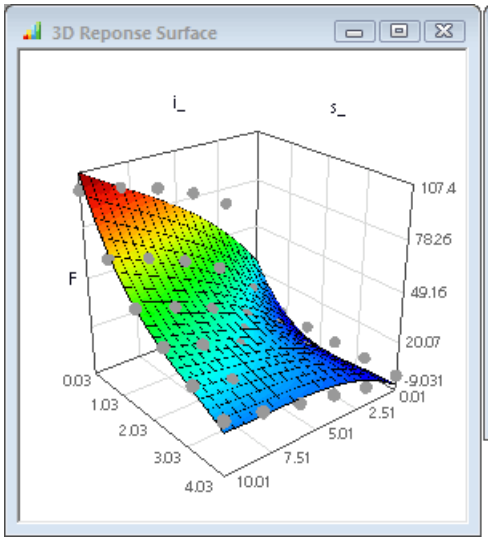
- Grundlage für die Berechnung der Antwortfläche soll das **Full Factorial Design** sein, welches praktisch einer Rastersuche entspricht. Wir wählen dafür **6 Stufen**, um die Anzahl der Modellberechnungen mit 36 gering zu halten. Die Anzahl 36 ergibt sich aus dem Raster 6x6.
- Wir setzen den **virtuellen Stichprobenumfang=0**, da wir in diesem Experiment nur die Antwortflächen gewinnen.

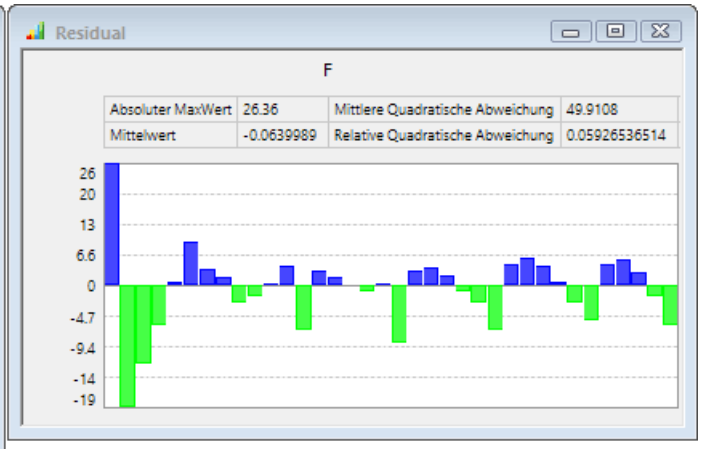
Approximation mit Polynom-Ansatz

- Wir verwenden zuerst die bereits für die Sample Methode genutzte **polynomiale Approximation** für die Gütekriterien und beginnen mit der einheitlichen **Polynomordnung=1**.
- Nach Abschluss der Simulation liegen die Antwortflächen als Funktionen im (i,s)-Raum vor.
- In die Diagramme der 3D-Antwortflächen blenden wir die Punkte der berechneten Stützstellen ein.
- **Schrittweises Erhöhen der Polynomordnung** mit anschließendem Neuberechnen der Antwortflächen führt zu einer optimalen Anpassung der Antwortflächen. Aus einem zu großem Wert für die Polynomordnung (z.B. ab Ordnung=6) resultiert dann wieder eine Verschlechterung der Approximation durch extrem starke Welligkeit der Fläche. Für unsere beiden Kennfelder führt

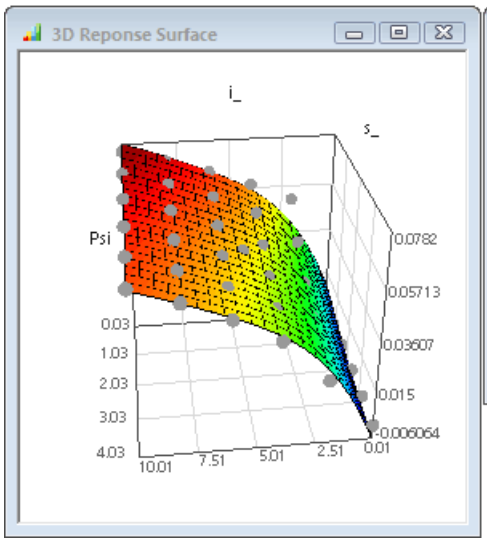
eine manuelle, individuelle Anpassung der Polynomordnung an die Abszissen-Richtungen (Strom und Luftspalt) zu keiner weiteren Verbesserung.
Achtung: Ein Rücksetzen des Experiments mit erneuter Berechnung der realen Stichprobe ist beim Ermitteln der optimalen Polynomordnung nicht erforderlich!

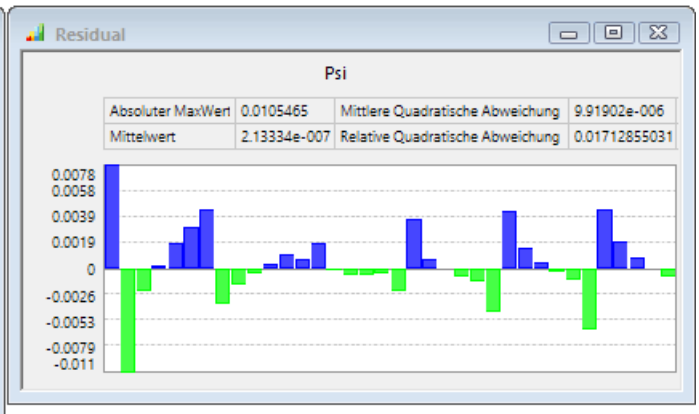
Kriterium Daten	
Name	F
Einheit	N
Kommentar	Magnetkraft
Werte	
Gewichtsfaktor [0..1]	1
Typ	Regression
Approximation	Polynomial
Cluster	
Cluster-Methode	Non-Cluster
Polynom	
Polynom-Typ	Einheitliche Ordnung
Polynomordnung	1
Niedrigrang-Approx	Full Matrix
Optimierungsziel	Minimieren
Letzter Wert	33.509





- Ein Drehen des Diagramms der 3D-Antwortfläche zeigt deutlich die Lage der realen Modellwerte abseits der Antwortfläche. Besonders stark sind die Abweichungen bei der Kombination von minimalem Luftspalt mit minimalem Strom.
- Das Residuum-Plot zeigt quantitativ die Abweichungen der realen Modellberechnungen von den identifizierten Antwortflächen. Bei der Magnetkraft liegt diese Abweichung überwiegend im Bereich von ca. 10%, was für ein Ersatzmodell häufig noch akzeptiert werden kann.
- Nicht akzeptabel ist jedoch, dass die Ersatzfunktion im Bereich kleiner Ströme zu negativen Kraftwerten führt, was physikalisch nicht korrekt ist!
- Ähnlich ungenau sehen die Ergebnisse für das Psi-Kennfeld aus:





Schlussfolgerung:
Polynom-Ansätze sind für die Bildung von globalen Ersatzmodellen bei nichtlinearem Übertragungsverhalten nicht besonders gut geeignet.

Approximation mit Gauss-Prozess

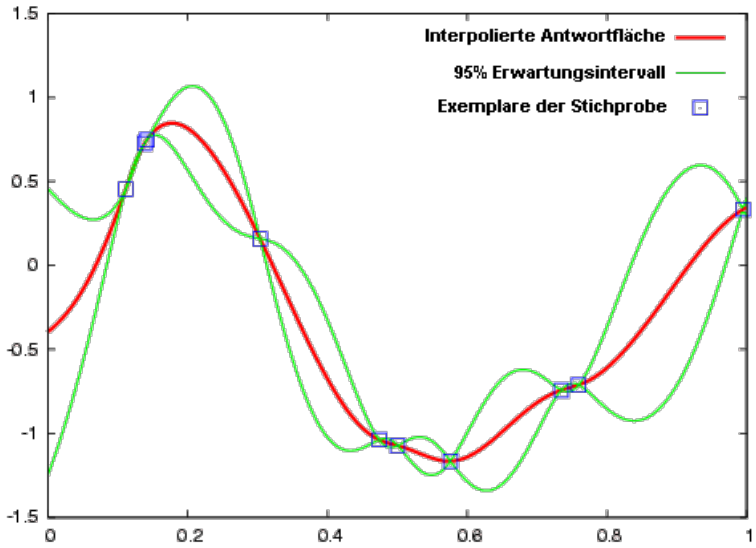
Der **Gauß-Prozess**, angewandt in der Geostatistik auch als **Kriging** bekannt, ist ein statistisches Verfahren, mit dem man Werte an Orten, für die keine Probe vorliegt, durch umliegende Messwerte interpolieren oder auch annähern kann.

Der Gauß-Prozess besteht aus einem globalen Modell **f(x)** und einem stochastischen Prozess **Z(x)**, welcher die mögliche Abweichung von dem globalen Modell beschreibt:

$$Y(\mathbf{x}) \quad = \quad \sum_{i=1}^{p} \beta_i \cdot f_i(\mathbf{x}) + Z(\mathbf{x})$$

- **x** ist ein m-dimensionaler Parametervektor.
- **Y(x)** ist der Ergebnisvektor für den Punkt **x** im Parameterraum.
- **f(x)** sind Polynome beliebiger Ordnung, welche zusammen mit den unbekannten Regressionskoeffizienten β_i die **Regressionsfunktion** bilden.
- **Z(x)** ist ein stationärer **stochastischer Prozess** mit dem *Mittelwert* Null, der *Varianz* σ und der *Covarianz* ***R***. Dieser Anteil des Gauß-Prozesses beschreibt das 95% Erwartungsintervall für jeden Punkt **x** des Parameterraumes.

In dem englischen Wikipedia-Artikel zum **Kriging** wird das für die Interpolation einer eindimensionalen Funktion sehr anschaulich dargestellt:



Die berechneten Grenzverläufe des Erwartungsintervalls werden wesentlich bestimmt durch das Erfahrungswissen in Hinblick auf den erwartenden Verlauf der zu interpolierenden Funktion zwischen den bekannten Werten der Stichproben-Exemplare. Diese Erwartung wird durch die Wahl einer geeigneten **Covarianz-Funktion** ***R*** beschrieben.

- In OptiY sind unterschiedlichste Covarianz-Funktionen ***R*** implementiert. Diese beschreiben den Verlauf des 95% Erwartungsintervalls zwischen den Stützstellen in Abhängigkeit von der Stützstellendichte. Die folgende Notation bezieht sich auf zwei Stützstellen x_1 und x_2 im Abstand $(x_1 - x_2)$:

1. **Square Exponential**
   $$R(x_1 - x_2) \quad = \quad \exp(-w^2 \cdot (x_1 - x_2)^2)$$
2. **Exponential**
   $$R(x_1 - x_2) \quad = \quad \exp(-w \cdot (x_1 - x_2))$$
3. **Gamma-Exponential**
   $$R(x_1 - x_2) \quad = \quad \exp(-w^\gamma \cdot (x_1 - x_2)^\gamma)$$
4. **Matern Class 3/2**
   $$R(x_1 - x_2) \quad = \quad (1 + w \cdot (x_1 - x_2)) \exp(-w \cdot (x_1 - x_2))$$
5. **Matern Class 5/2**
   $$R(x_1 - x_2) \quad = \quad (1 + w \cdot (x_1 - x_2) + w^2 \cdot (x_1 - x_2)^2 / 3) \exp(-w \cdot (x_1 - x_2))$$
6. **Periodic**
   $$R(x_1 - x_2) = \exp(-w^2 \cdot \sin^2(\pi \frac{|x_1 - x_2|}{\lambda}))$$
7. **Rational Quadratic**
   $$R(x_1 - x_2) \quad = \quad \left(1 + \frac{w^2 \cdot (x_1 - x_2)^2}{\alpha}\right)^{-\alpha}$$

- Der allgemeine Fall für Exponential-Funktionen ist *Gamma-Exponential*, die anderen beiden Exponentialfunktionen sind die Spezialfälle für $\gamma=1$ bzw. $\gamma=2$.
- Die *Matern Class* Funktionen sind Erweiterungen der Exponential-Funktion.
- Die Hyper-Parameter **w**, **γ** , **λ** und **α** werden mittels der Maximierung der **Likelihood-Funktion** der **multivariaten Normalverteilung** ermittelt.

Die Ermittlung dieser Hyper-Parameter des Gauß-Prozesses erfolgt in der statistischen Versuchsplanung mittels eines implementierten Optimierungsverfahrens, in dessen Konfiguration wir die Standardparameter nutzen. Die in der Versuchsplanung zu wählende Optimierungsmethode für das "Robust Design" ist davon völlig unabhängig:

⊟ Statistische Versuchsplanung	
Verfahren	Factorial Design
Parameter	Full Factorial
Stufen	6
⊟ Antwortfläche	
Adaptives Sampling	☐
Trainingsdaten [%]	100
Messdaten	
⊞ Trainingsdaten	
⊞ Kernel-Methode	
⊞ Nichtlineare Methode	
⊞ 1D-Variablen	
⊞ Hierachische Matrix	
⊟ Probabilistik	
Virtueller Stichprobenumfang	0
Verteilungsraster	50
⊟ Robust Design	
Optimierungsmethode	Evolutionsstrateg
Max. Schritte	500
Anzahl der Eltern	4
Anzahl der Kinder	20
Pareto-Filter	☐

Über die Polynomordnung des globalen Modells kann man einen Kompromiss finden zwischen bester Anpassung der interpolierten Regressionsfunktion an vorhandene Stützstellen und optimalem Verlauf zwischen diesen Stützstellen:

- Wie im vorherigen Abschnitt "Approximation mit Polynom-Ansatz" beschrieben, bestimmt die Polynomordnung des globalen Modells **f(x)** die allgemeine Richtung (**globale Anpassung**) der Regressionsfunktion. Die dabei verbleibenden Residuen sind noch sehr groß.
- Der stochastische Prozess **Z(x)** hat dann die Aufgabe, diese verbleibenden Residuen mittels der Covarianz-Funktion (Normal-Verteilung) zu eliminieren (**lokale Anpassung**).
- Wenn die Anzahl der Stützstellen bzw. der Daten hinreichend groß ist und man die verbleibenden Residuen statistisch auswertet, entsteht eine **Normal-Verteilung** (auch Gauss-Verteilung genannt) mit dem Mittelwert=0. Das ist die ursprüngliche Idee des Gauß-Prozesses.
- Wenn man die globale Anpassung mit einer zu hohen Ordnungen der Polynome durchführt, besitzt die Kurve mehr Freiheitsgrade als nötig. Das führt dann zu Welligkeiten zwischen den Stützstellen, weil dies durch keine Zwangsbedingungen verhindert wird.

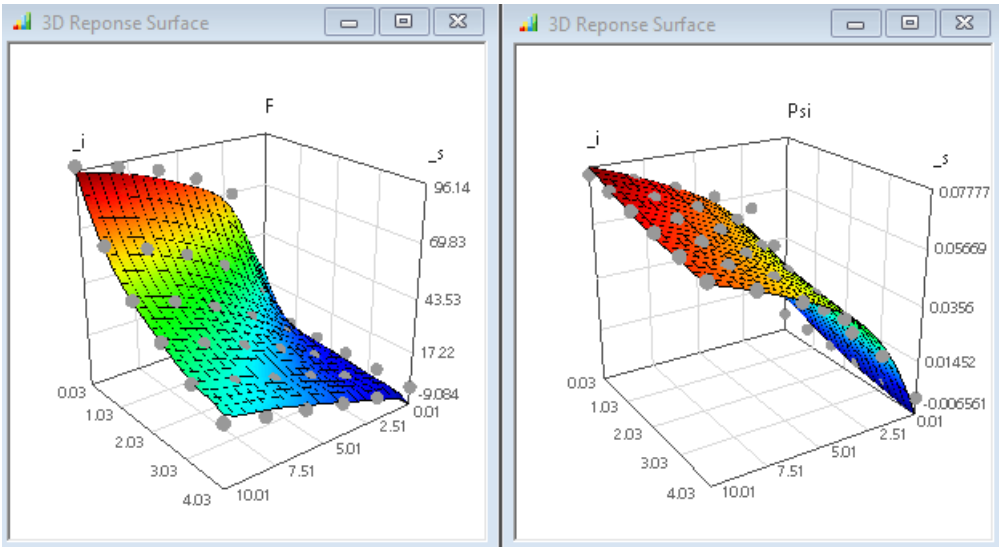
Dieses Prinzip der Bildung von Ersatzmodellen (Antwortflächen) werden wir nun auf unser Kennfeld-Problem anwenden. Wir nutzen dafür das bereits konfigurierte Experiment *Kennfeld-Identifikation* ohne erneute Berechnung der Stichprobe:

- Wir wählen für die beiden Ergebnisgrößen **F** und **Psi** zur Approximation den **Gauss Prozess** und überlassen dem Programm für den **Polynom-Typ** die Ermittlung der **Beste Ordnung** für die Polynome in jede Streuungsrichtung.
- Mit der Option **Best Covariance** versucht *OptiY* automatisch die beste **Covarianz-Funktion** zu ermitteln, was inzwischen recht gute Ergebnisse liefert:

Kriterium Daten	
Name	F
Einheit	N
Kommentar	Magnetkraft
Werte	
Gewichtsfaktor [0..1]	1
Typ	Regression
Approximation	Gaussian Process
Cluster	
Cluster-Methode	Non-Cluster
Gauß-Prozess	
Covarianz-Funktion	Best Covariance
Niedrigrang-Approxim	Full Matrix
Polynom	
Polynom-Typ	Beste Ordnung
Niedrigrang-Approxim	Full Matrix
Optimierungsziel	Minimieren
Letzter Wert	21.2869

Kriterium Daten	
Name	Psi
Einheit	Wb
Kommentar	Koppelfluss
Werte	
Gewichtsfaktor [0..1]	1
Typ	Regression
Approximation	Gaussian Process
Cluster	
Cluster-Methode	Non-Cluster
Gauß-Prozess	
Covarianz-Funktion	Best Covariance
Niedrigrang-Approxim	Full Matrix
Polynom	
Polynom-Typ	Beste Ordnung
Niedrigrang-Approxim	Full Matrix
Optimierungsziel	Minimieren
Letzter Wert	0.0634976

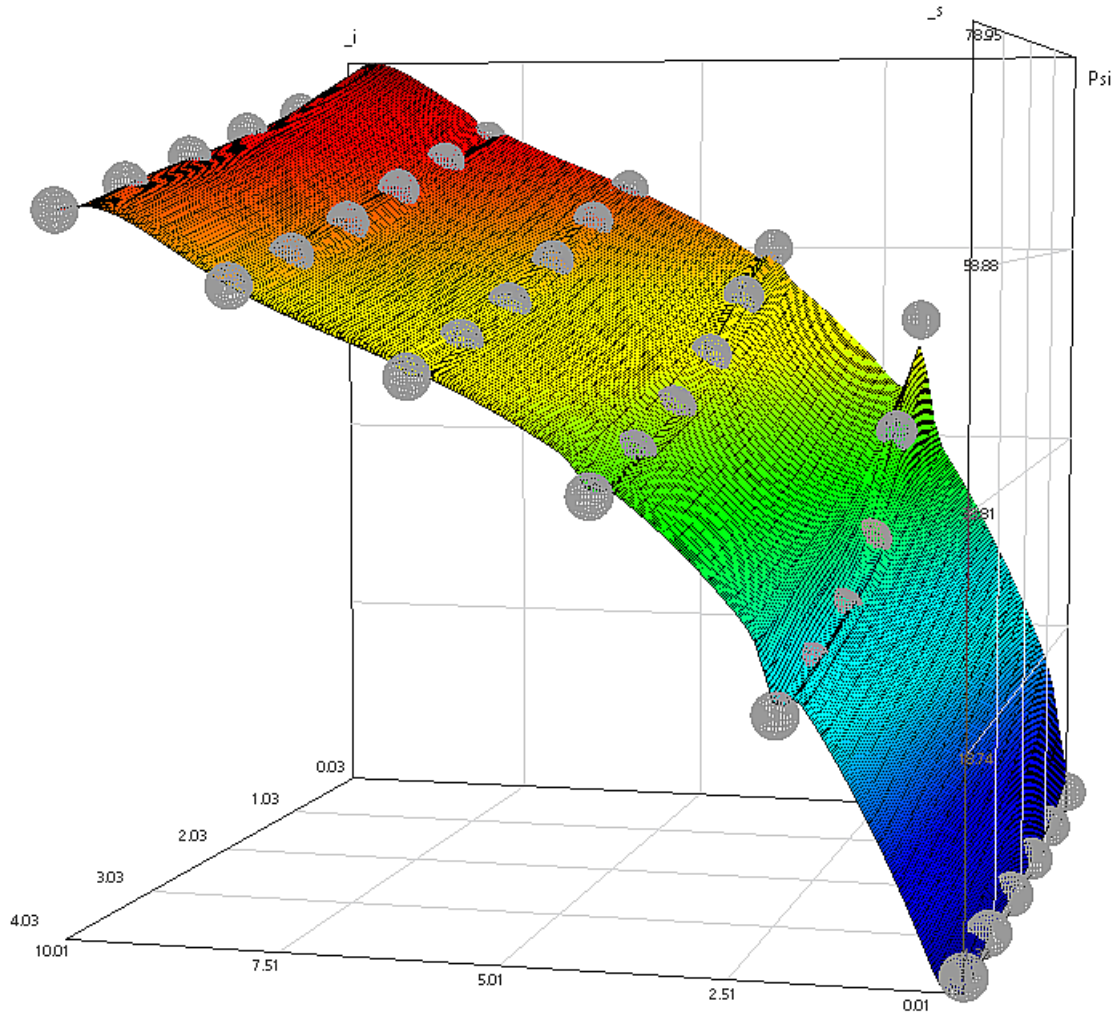
- Nach dem Neuberechnen der Antwortflächen ergeben sich folgende Approximationen für das F-Kennfeld und das Psi-Kennfeld:



- Die Residuen zeigen, dass die Punkte der echten Stichprobe etwas genauer in die identifizierte Ersatzfunktionen **F=f(s,i)** bzw. **Psi=f(s,i)** eingebettet sind, als beim reinen Polynom-Ansatz:

- ***Hinweis:*** In Versionen vor OptiY 4.6 kam es zu Problemen bei der Residuenberechnung, welche eine höhere Genauigkeit darstellten, als mit den Antwortflächen eigentlich erreicht wurde.
- Die jetzt korrekt berechneten Residuen decken sich in etwa mit der 3D-Darstellung der Antwortflächen. Trotzdem ist zu beachten::
  1. Die dargestellten Flächen beruhen auf einer Approximation der wirklichen 3D-Funktionen.
  2. Die Genauigkeit der Darstellung wird über die Zahl der Rasterpunkte gesteuert (Standardwert=20). Dabei kann jedoch die Genauigkeit der Darstellung nicht besser werden, als die zugrundeliegende Genauigkeit der Approximation.

- **Nur zur Info - muss in der Übung nicht ausprobiert werden:**
  - Um den Effekt hier in dieser Anleitung zu verdeutlichen, wurde innerhalb des Workflows der Psi-Wert mit dem Faktor 1000 multipliziert (Psi-Werte also in mWb).
  - Nach erneuter Berechnung der Stichprobe ergibt sich dann eine wesentlich bessere Approximation der darzustellenden Fläche (hier dargestellt mit dem Raster 100x100).
  - Dabei wird deutlich, dass die Funktionsfläche des Gauss-Prozesses so "zurecht gezupft" wurde, dass die Stützstellen wirklich exakt auf dieser Fläche liegen.
  - Dadurch entstehen "Unstetigkeiten" (hier in Form schmaler Rinnen bzw. Grate), welche in der Realität nicht vorhanden sind:

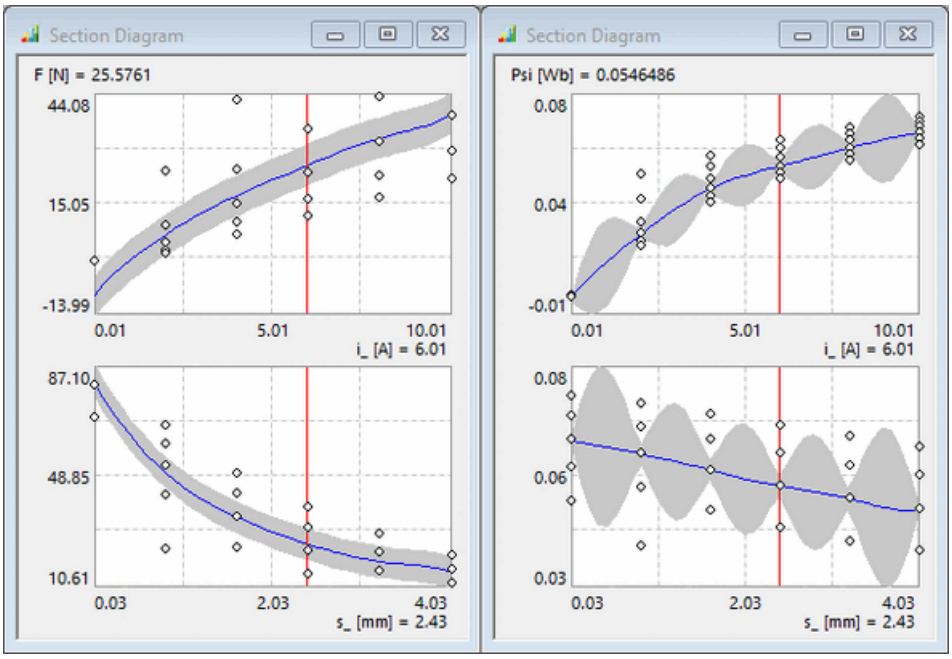


***Schlussfolgerung*:**

- Man muss die mittels automatisiertem Gauss-Prozess erzeugten Antwortflächen einer gründlichen Analyse in Hinblick auf unzulässige Welligkeiten und Unstetigkeiten untersuchen.
- Bei Bedarf muss man den Gauss-Prozess dann doch manuell optimieren:
  1. Verwendung einer möglichst kleinen Polynomordnung.
  2. Wahl einer geeigneten Covarianzfunktion zur Erzeugung realistischer Übergänge zwischen den Stützstellen der realen Stichprobe.

Zusätzlich zu den Residuen und den 3D-Antwortflächen öffnen wir für jede Ergebnisgröße auch die Schnittdiagramme. Innerhalb dieser Schnittdiagramme aktivieren wir die Darstellung des Vertrauensintervalls:

- Die automatisch identifizierten Kennfelder besitzen Welligkeiten zwischen den Stützstellen. Hier würde man physikalisch monotone Verläufe der Kennfelder erwarten.
- Die 95% Vertrauensintervalle zwischen den Stützstellen sind recht eng:



Insbesondere den Aspekt des Vertrauensintervalls sollten wir näher betrachten:

- Um die obige Darstellung der Vertrauensintervalle zu erhalten, müssen wir:

1. die Anzeige der Stützpunkte für die Schnittdiagramme aktivieren.
2. die aktuellen Werte der Streugrößen (*Virtueller Entwurf - Nennwert*) auf eine Stützstellen-Reihe setzen (z.B. _i=6.01 A und _s=2.43 mm).

***Wichtig*:**

- An den Stützstellen selbst ist die Breite des Vertrauensintervalls immer Null, da der Wert bekannt ist.
- Die berechneten Vertrauensintervalle resultieren nur aus den benutzten mathematischen Ansätzen. Entscheidend für uns sind jedoch die interpolierten Funktionsverläufe von Kraft und Koppelfluss! Wir wissen (im Unterschied zum mathematischen Formalismus), dass sich diese physikalischen Größen zwischen den Stützstellen monoton ändern (ohne "Welligkeit").
- Die Vertrauensintervalle zeigen den *Worst Case* der Unsicherheit bei fehlenden Kenntnissen zu den abgebildeten physikalischen Zusammenhängen.

Hinweise

Die beschriebene Vorgehensweise für die Identifikation von Ersatzmodellen als Antwortflächen mittels Gauß-Prozess ist nur unter den folgenden Randbedingungen durchführbar:

1. Es existieren bereits hinreichend viele Stützstellen für das zu identifizierende Modell innerhalb des vorgesehenen Wertebereiches der freien Parameter.
2. Die Anzahl der freien Parameter ist überschaubar (z.B. maximal drei).
3. Man besitzt bereits qualitative Vorstellungen zu den Eigenschaften der Antwortflächen.

Die Identifikation von Kennfeldern für einen elektro-magnetischen Wandler ist nicht ganz einfach:

- Entscheidend für die Genauigkeit des damit aufzubauenden Dynamik-Modells ist die hinreichend genaue zeitliche Ableitung des Psi-Kennfeldes. Die damit berechnete Induktionsspannung bestimmt die zeitliche Entwicklung des Stromes in der Spule.
- Der Gauss-Prozess berücksichtigt nur, wie exakt die berechneten Stützstellen in die Antwortfläche passen und wie breit das Vertrauensintervall der identifizierten Antwortfläche zwischen den berechneten Stützstellen ist. Knicke und Welligkeiten der Antwortfläche fließen nicht in die

Bewertung der Antwortflächen-Güte ein, sind aber entscheidend für die Güte der benötigten Ableitung!
- Die Güte der Ableitung eines Kennfeldes kann man mittels OptiY zurzeit nur iterativ durch die Wahl einer geeigneten Covarianz-Funktion in Kombination mit einer günstigen Polynomordnung beeinflussen.

Qualität der Ableitung:

- Benötigt man innerhalb der Systemsimulation partielle Ableitungen der identifizierten Funktionen, so muss man dies als zusätzliches Qualitätsmerkmal berücksichtigen. Im Beispiel trifft dies für **Psi(i,s)** zu, denn die zeitliche Ableitung des Koppelflusses entspricht im Netzwerkmodell der induzierten Spannung in der Spule.
- Leider kann man die Qualität partieller Ableitungen der identifizierten Antwortflächen innerhalb des OptiY nur sehr grob beurteilen:
  1. Knicke bewirken Sprünge der Ableitung
  2. Welligkeit kann zu Vorzeichenwechsel der Ableitung führen
- Um die physikalische Sinnfälligkeit der Ableitung zu beurteilen, muss man zurzeit diese Ableitung auf Basis des exportierten Modell-Codes berechnen und darstellen!

← →

Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld_-_Probabilistik_-_Kennfeld-Identifikation&oldid=27899“

Software: FEM - Tutorial - Magnetfeld - C-Kennfeld Adaptiver Gauss-Prozess

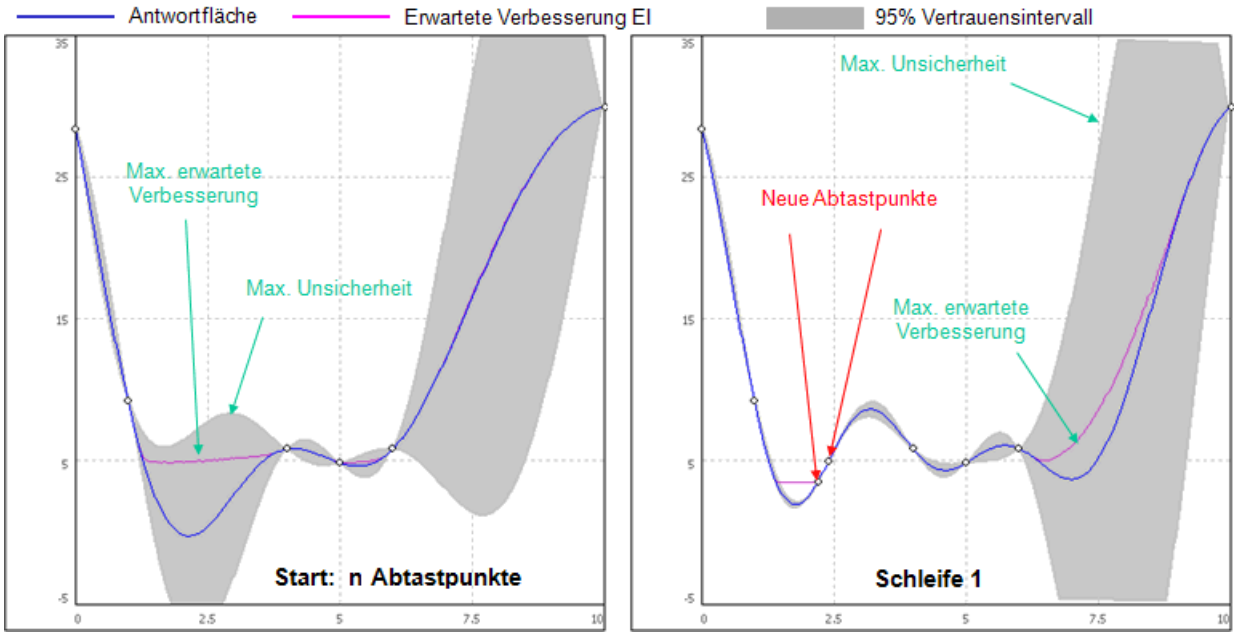
Aus OptiYummy

↑

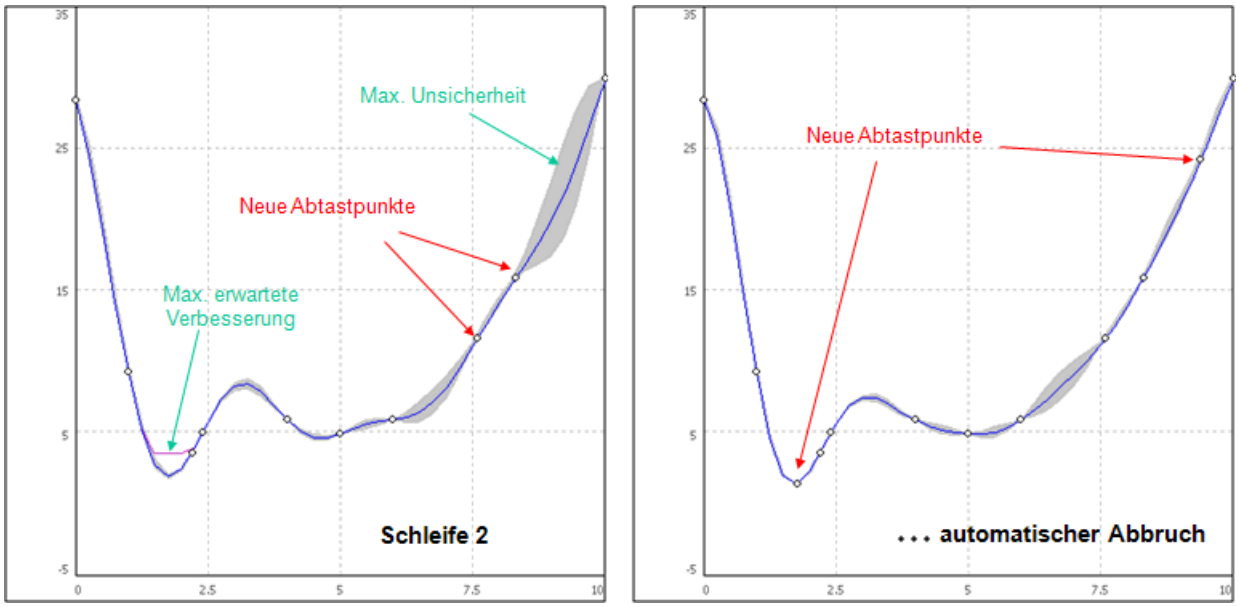
← →

Adaptiver Gauß-Prozess

Der adaptive Gauss-Prozess soll genutzt werden, um ausgehend von einer vorhandenen Stichprobe mit möglichst wenigen zusätzlichen Stützstellen ein hochwertiges Ersatzmodell zu gewinnen. Dabei werden ausgehend von den vorhandenen Abtastpunkten nach Identifikation der Antwortflächen die Positionen der maximalen Unsicherheit und der maximal zu erwartenden Verbesserung ermittelt. Dort werden mit dem Modell zwei neue Stützstellen berechnet:



Dieser Prozess wird iterativ ausgeführt (bis max. Stützstellen-Zahl oder geforderte Genauigkeit erreicht):



Adaptiver Gauß-Prozess:

- Die erwartete Verbesserung ***EI*** entspricht dem Potenzial zur Verbesserung der Approximationsgüte. Sie ist definiert mit

$$EI = \sigma\left\{\frac{y_{\min}-\hat{Y}}{\sigma}\Phi\left(\frac{y_{\min}-\hat{Y}}{\sigma}\right)+\Psi\left(\frac{y_{\min}-\hat{Y}}{\sigma}\right)\right\}$$

  wobei Φ und ψ die Verteilungsfunktion und die Dichtefunktion der normalisierten Normalverteilung sind.
- Der adaptive Gauß-Prozess schlägt für die nächste Schleife der Modellberechnungen weitere Punkte im Entwurfsraum vor, welches folgende Bedingungen erfüllen:
  1. Maximierung der erwarteten Verbesserung, um den optimalen Entwurf in Hinblick auf die geforderte Genauigkeit zu validieren.
  2. Maximierung der Varianz σ, um die Unsicherheiten der Vorhersage zu minimieren.
- Diese nächsten Punkte werden anhand des aktuellen Gauß-Prozess-Ergebnisses durch Optimierung mittels einer Evolutionsstrategie berechnet.

Konfiguration

Durch **Duplizieren** des Experiments "**Kennfeld-Identifikation**" gewinnen wir ein neues Experiment "**Kennfeld-adaptiv**":

- Basis ist wieder ein 6-stufiges *Full Factorial Design*, dessen Ergebnis-Antwortfläche durch zusätzliche Abtastungen des Modells verbessert werden soll.
- Für die Polynom-Ordnung und für die Covarianz-Funktion nutzen wir diesmal jedoch nicht die Automatismen, sondern legen auf Basis der gesammelten Erfahrungen eine niedrige Polynom-Ordnung und eine "gutmütige" Covarianz-Funktion fest. Als günstig für den Gauß-Prozess der beiden Antwortflächen (**F** und **Psi**) haben sich die einheitliche **Polynomordnung=1** mit der **Co-Varianzfunktion=Exponential** bewährt und diese sollten deshalb hier verwendet werden:

⊟ Kriterium Daten	
Name	F
Einheit	N
Kommentar	Magnetkraft
⊟ Werte	
Gewichtsfaktor [0..1]	1
Typ	Regression
Approximation	Gaussian Process
⊟ Cluster	
Cluster-Methode	Non-Cluster
⊟ Gauß-Prozess	
Covarianz-Funktion	Exponential
⊟ Polynom	
Polynom-Typ	Einheitliche Ordnung
Polynomordnung	1
Niedrigrang-Approximation	Full Matrix
Optimierungsziel	Minimieren
Letzter Wert	49.9756

⊟ Statistische Versuchsplanung	
Verfahren	Factorial Design
Parameter	Full Factorial
Stufen	6
⊟ Antwortfläche	
Adaptives Sampling	☑
Messdaten	
⊟ Adaptiver Gauss-Prozess	
Max. Vertrauensintervall [%]	1
Max. Punkte	50
Anzahl der Eltern	10
Anzahl der Kinder	50
⊟ Kernel-Methode	
Max. Ordnung	10
Rausch-Optimierung	☑
Gewichtsoptimierung	☐
Optimierungsmethode	Gradient-Verfahren
Max. Iterationen	50
⊞ Nichtlineare Methode	
⊞ 1D-Variablen	
⊟ Hierachische Matrix	
Große Daten	Automatisch
Max. Matrixgröße	32
⊟ Probabilistik	
Virtueller Stichprobenumfang	0
Verteilungsraster	50
⊞ Robust Design	

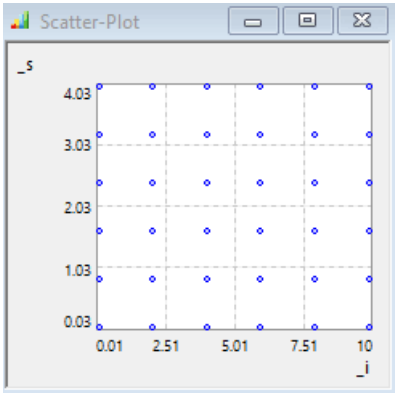
- Wir aktivieren in der Versuchsplanung des adaptive Sampling und blenden mittels "Standardparameter=False" die Standardeinstellungen ein.
- Immer ausgehend vom Ergebnis des aktuellen Gauß-Prozesses sollen dann durch den adaptiven Gauß-Prozess die nächsten erforderlichen Stützstellen berechnet werden.

- Sowohl für den Gauß-Prozess als auch für den sich anschließenden adaptiven Gauß-Prozess verwenden wir eine Evolutionsstrategie mit einheitlichen Vorgabewerten von 10 Eltern und 50 Kindern.
- Die angestrebte Genauigkeit der Approximation wird durch das max. Vertrauensintervall (im Beispiel 1%) beschrieben.
- Die Anzahl der **max. Punkte=50** dient als vorzeitiges Abbruch-Kriterium, falls die geforderte Genauigkeit noch nicht erreicht wurde. Diese Anzahl umfasst nur die neu vorgeschlagenen Punkte nach dem ersten Gauß-Prozess und ist in der Trial-Version auf 100 Punkte begrenzt.

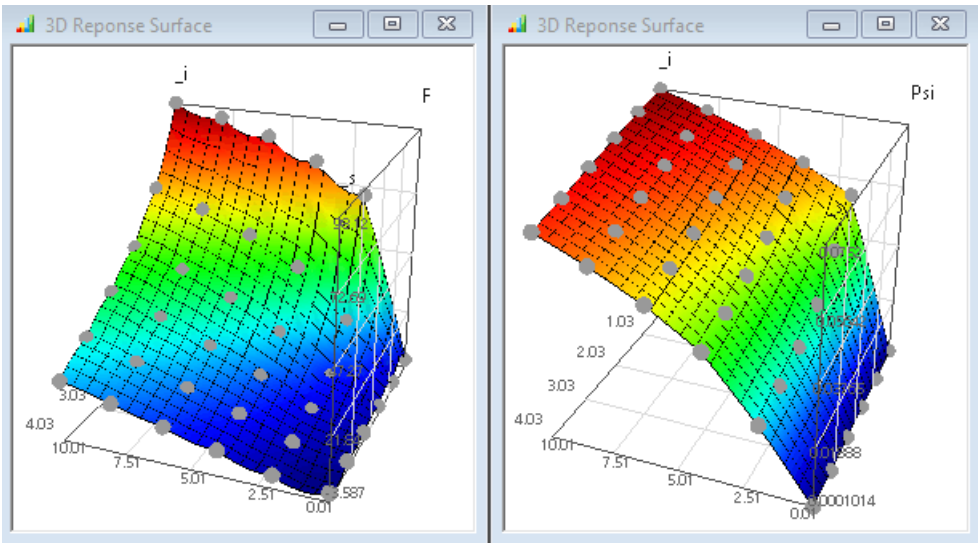
Fortschritt

- Um den Fortschritt der adaptiven Kennfeld-Approximation Live verfolgen zu können, schalten wir vorläufig "**Adaptives Sampling=False**".
- Damit findet nach Start des Experiments nur für die 36 Punkte des *Full Factorial Design* eine Approximation mittels "normalem" Gauß-Prozess statt.
- Hierfür konfigurieren wir die folgenden grafischen Darstellungen:

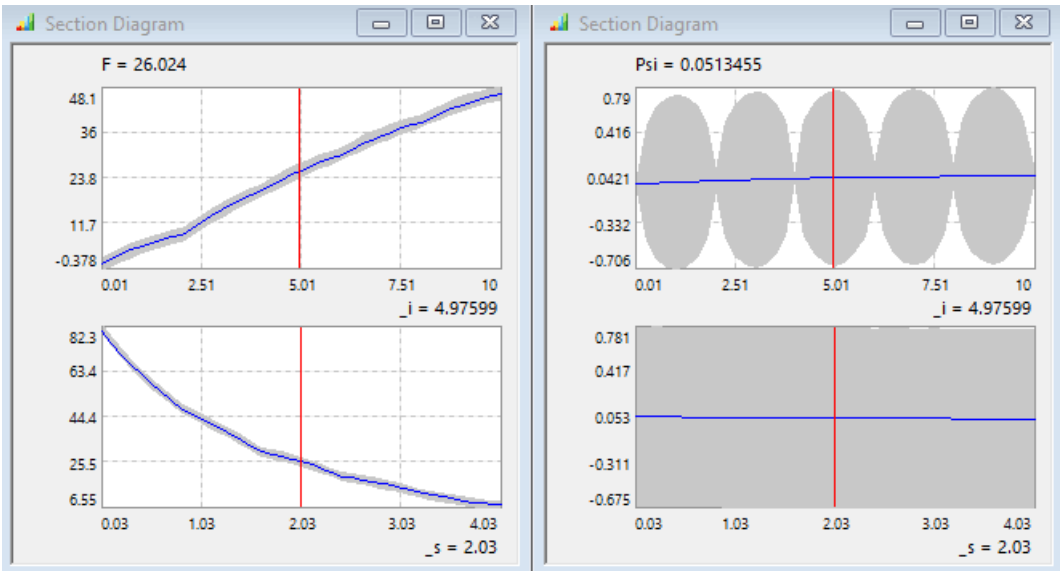
2D Anthill-Plot (_i,_s):



3D-Antwortflächen:

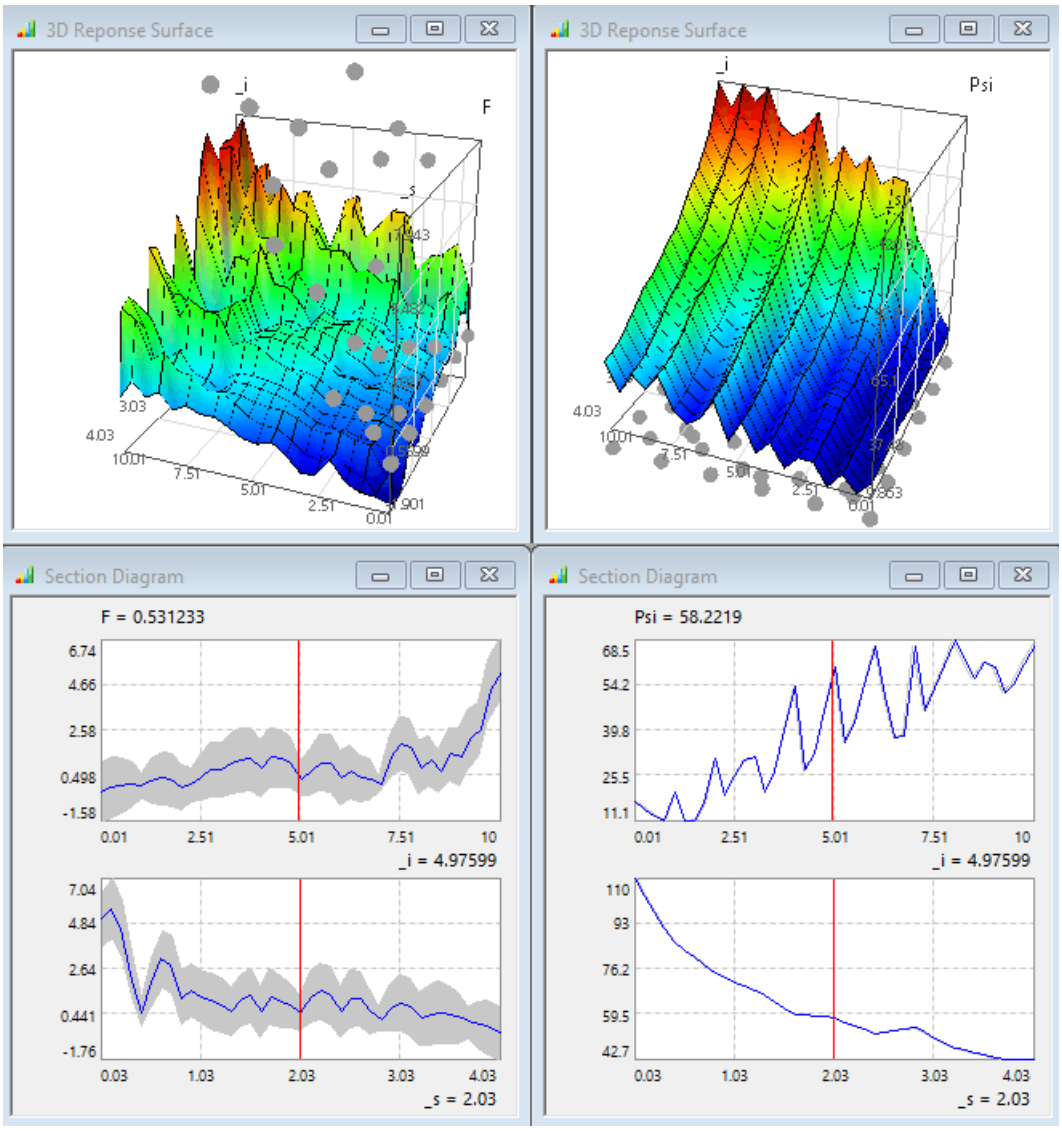


Schnittdiagramme:

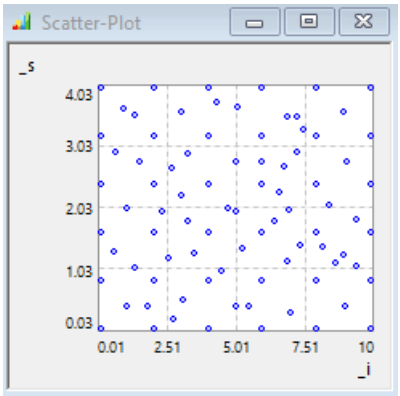


Adaptiver Gauß-Prozess:

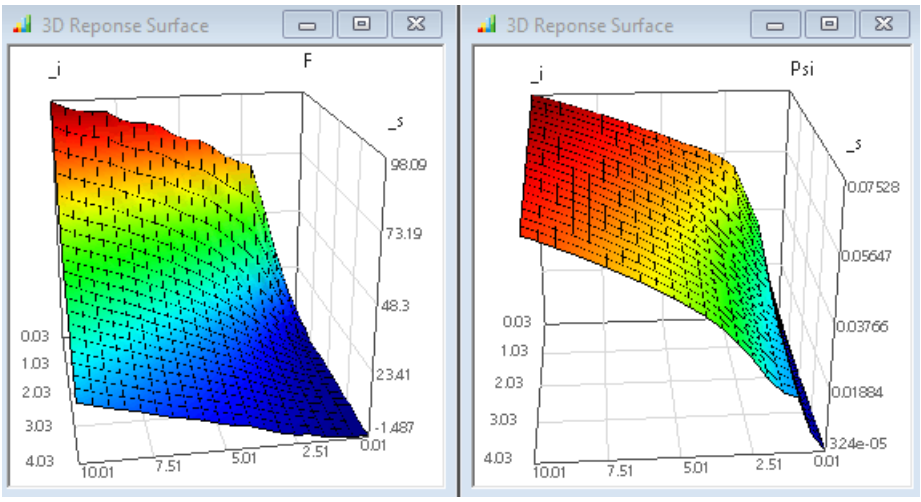
- Nach erfolgreicher Initialisierung und Visualisierung der Ausgangslösung aktivieren wir wieder "**Adaptives Sampling=True**"
- Neuberechnen der Antwortflächen startet dann den iterativen adaptiven Gauß-Prozess und ergänzt in jeder Iteration zwei weitere Punkte.
- Unter Einbeziehung dieser zwei neuen Punkte wird dann der "normale" Gauß-Prozess berechnet und visualisiert. So kann man Live diesen Approximationsprozess beobachten.
- Hinweis:
  - Leider funktioniert die Beobachtung des Approximationsfortschrittes in der aktuellen OptiY-Version nur sehr eingeschränkt.
  - Bei genauem Hinschauen erkennt man, dass jeweils zwei neue Stützstellen mit dem Modell berechnet werden.
  - Daraus ergeben sich jedoch jeweils sehr wellige Antwortflächen, welche kein sinnvolles Zwischenergebnis widerspiegeln:



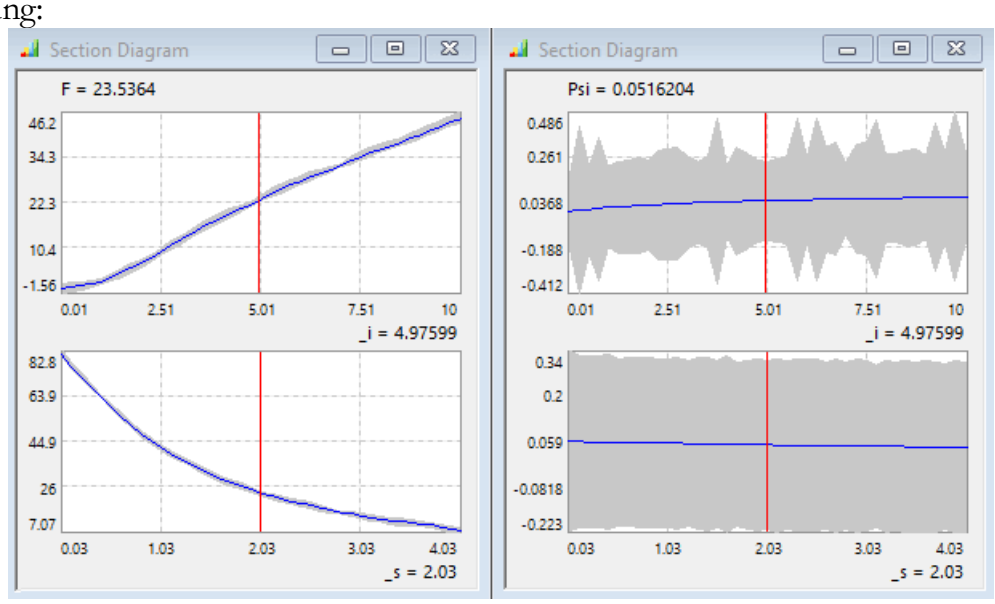
Die zusätzlichen Abtaststellen verteilen sich weitestgehend gleichmäßig über den gesamten Streubereich:



- Der Abbruch erfolgte im Beispiel nach Erreichen der max. zusätzlichen Punkteanzahl (also vor dem Erreichen der geforderten Genauigkeit). Die dann korrekt berechneten Antwortflächen zeigen jedoch noch physikalisch nicht begründbare Welligkeiten, negative Werte und "Knicke":



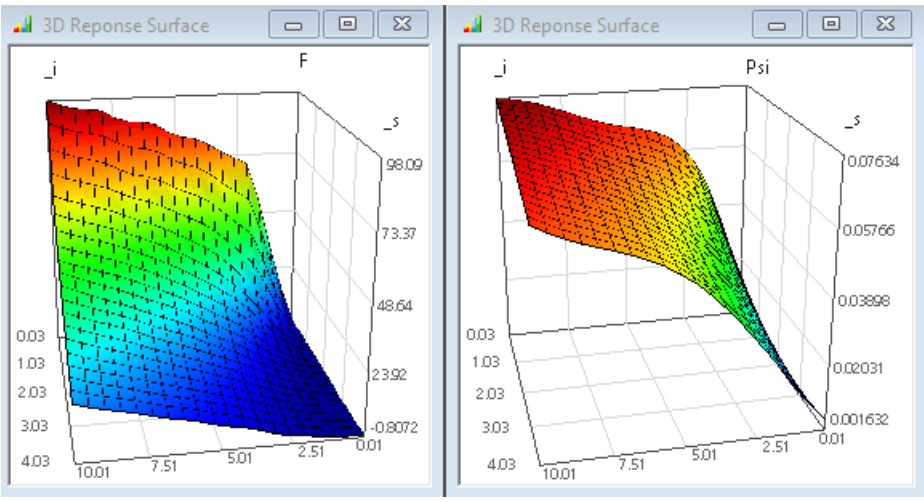
- Das 95%-Erwartungsintervall verringerte sich für beide Ersatzfunktionen im Vergleich zur Ausgangslösung:



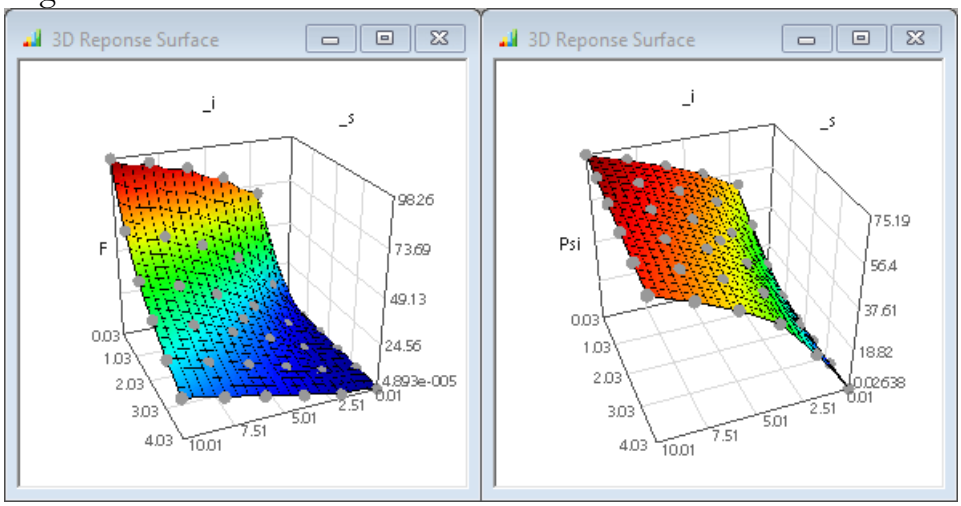
Fazit

Der adaptive Gauß-Prozess verbessert schrittweise die Qualität eines Ersatzmodells ausgehend von der möglichst kleinen globalen Abtastung des zu berücksichtigenden Werte-Bereiches (hier 6-stufiges *Full Factory Design*, um eine qualitativ bereits anschauliche Ersatzfunktion zu erhalten):

- Mit wenig Vorwissen zum Verlauf der realen Antwortfläche erhält man bei einer hinreichend großen maximal zulässigen Punkte-Anzahl eine glaubwürdige Approximation auch an kritischen Stellen einer nichtlinearen Übertragungsfunktion.
- Die Punktedichte passt sich automatisch an die Nichtlinearitäten der Übertragungsfunktion an.
- Mit dem erhaltenen Ergebnis kann man noch experimentieren, indem man das "Adaptive Sampling" deaktiviert und mit verschiedenen Konfigurationen (Covarianz und Polynomordnung) jeweils die Antwortflächen neu berechnet. Bei automatischer ermittlung der "Best Covarianz" ergaben sich z.B. physikalisch durchaus sinnvollere Ersatzfunktionen (geringere negative Kraft, monotoner Psi-Verlauf):



- Für "gutmütige" Übertragungsfunktionen (wie in unserem Magnet-Beispiel) bringt der adaptive Gauß-Prozess meist keine Berechnungs- und Genauigkeitsvorteile im Vergleich zur klassischen Abtastung mittels **Rastersuche**.
- Eine einmalige, vertrauenswürdige Approximation der Antwortflächen mittels relativ feiner Abtastung bildet die Grundlage, um z.B. mit einer gröberen Abtastung im weiteren Prozess trotzdem hinreichend genaue Ersatzmodelle zu erzeugen.
- Unser Beispiel zeigte, das der "normale" Gauß-Prozess bei einem 6-stufigem *Full Factory Design* ähnlich genaue Ergebnisse liefern kann:



- Nach einer Änderung des FEM-Modell kommt man mit diesem Wissen relativ schnell wieder zu einem hinreichend genauem Ersatzmodell für die Dynamik-Simulation des Magnet-Antriebs.

← →

Abgerufen von „http:///index.php?title=Software:_FEM_-_Tutorial_-_Magnetfeld_-_C-Kennfeld_Adaptiver_Gauss-Prozess&oldid=25240“