

Inhaltsverzeichnis

1	Einleitung	2
1.1	Zielsetzung	2
1.2	Hysteresemodelle	2
1.3	Parameteridentifikation	3
1.4	Automatisches Differenzieren	3
1.5	Vorgehen	3
2	Jiles-Atherton-Modell	6
2.1	Konstituierende Gleichungen	6
2.2	Herleitung der Differentiale	7
3	Numerische Integration	8
3.1	Explizites Euler-Verfahren	9
3.2	Implizites Euler-Verfahren	9
3.3	Kurzer Vergleich	10
3.4	Runge-Kutta-Methoden	11
3.5	Integration des Jiles-Atherton-Modells	12
3.6	Numerische Details	13
4	Anwendung des Jiles-Atherton-Modells	14
4.1	Stromanregung	15
4.2	Spannungsanregung	16
4.3	Allgemeiner Fall	17
5	Parameteridentifikation	17
5.1	Least-Squares-Methode	17
5.2	Probleme bei der Parameteridentifikation für das Jiles-Atherton-Modell	18
6	Automatisches Differenzieren	19
6.1	Forward Mode	20
6.2	Reverse Mode	21
6.3	Weiterführendes	23
7	Implementation für das Jiles-Atherton-Modell	24
7.1	Resultate	24
8	Ausblick	25

1 Einleitung

1.1 Zielsetzung

Ziel des Projektes ist es, die Kernverluste in induktiven Bauelementen besser vorhersagen zu können als mit den bisherigen Methoden. Das typische Vorgehen bei der Abschätzung dieser Verluste basiert derzeit meist auf der Steinmetz-Formel bzw. auf ihren Erweiterungen. Diese beschreibt rein phänomenologisch die Verluste abhängig von Flussdichte, Frequenz und Temperatur. Dabei wird jedoch weder die Strom- noch die Kernform einbezogen.

1.2 Hysteresemodelle

Um hier eine Verbesserung erzielen zu können, ist es notwendig, die Hysterese transient zu beschreiben und mit diesem Modell die Verluste zu berechnen. Die Wissenschaft hat hierfür einige Modelle hervorgebracht:

Komplexe Permeabilität

Jiles-Atherton

Preisach

Coleman-Hodgdon

Krasnosel'skii–Pokrovskii

Maxwell-Slip

Bouc–Wen

Duhem

VINCH

Die verschiedenen Modelle unterscheiden sich insbesondere in der Anzahl der Modellparameter. Während das Jiles-Atherton mit fünf Parametern auskommt, können etwa im Preisach- oder VINCH-Modell beliebig viele Parameter eingeführt werden – dies führt dazu, dass gemessene Hysteresekurven genauer angenähert werden können, jedoch vermutlich die Allgemeingültigkeit ein Stück weit verloren geht.

Ein entscheidendes Problem bei fast allen erwähnten Modellen ist die Parameteridentifikation – die Modellparameter müssen an Messkurven angeglichen werden. Bei dem Modell der komplexen Permeabilität fällt dies noch recht leicht, da hier einer einfachen Leistungsmessung ausreicht (noch dazu ist das Modell

linear). Für die anderen Modelle, die alle nichtlinear sind, gestaltet sich dies jedoch deutlich schwieriger.

1.3 Parameteridentifikation

Das übliche Vorgehen ist, eine Kostenfunktion aufzustellen, die ihr Minimum dann annimmt, wenn Messdaten und Simulationsdaten am besten übereinstimmen. Diese Kostenfunktion muss anschließend mit einem geeigneten Algorithmus minimiert werden. Für nichtlineare Modelle wie die vorliegenden gestaltet sich dies jedoch oft sehr schwierig – die meisten Algorithmen benötigen für eine schnelle Konvergenz die Jacobi- oder sogar auch die Hesse-Matrix der zu minimierenden Kostenfunktion. Diese sind analytisch jedoch nicht zugänglich. Ein alternativer Ansatz ist, gradientenfreie Algorithmen zu verwenden (genetische Algorithmen, linear Programming etc.). Dies wurde in der Literatur auch wiederholt vorgenommen – die Konvergenz kann hierbei jedoch nicht garantiert werden, noch dazu sind diese Algorithmen recht langsam.

1.4 Automatisches Differenzieren

Ein Ausweg aus diesen Problemen versprechen die neuen Möglichkeiten aus dem Gebiet des automatischen Differenzierens. Durch die stetige Erhöhung der Rechenleistung von Computern und die neuen Anwendungsmöglichkeiten im Bereich der künstlichen Intelligenz konnten hier in den letzten Jahren viele Verbesserungen erreicht werden.

Mit dem automatischen Differenzieren ist es möglich, Computerprogramme abzuleiten, also Jacobi-, Hessematrix etc. für Programme zu bestimmen. Die Idee hinter diesem Projekt ist es nun, die Fortschritte in diesem Bereich zu nutzen und auf das dargestellte Problem der Parameteridentifikation von Hysteresemodellen anzuwenden.

1.5 Vorgehen

In Abbildung 1 wird das Vorgehen schematisch dargestellt, zumindest der erste Teil des Projektes, die Lösung der Aufgabe im nicht-orts aufgelösten Fall. Der Projektplan für den zweiten Teil ist im Anhang, Abbildung 11 bildlich dargestellt. Als Beispielmmodell wird das Jiles-Atherton-Modell verwendet, da es relativ populär ist und somit viele Erfahrungswerte in der wissenschaftlichen Literatur

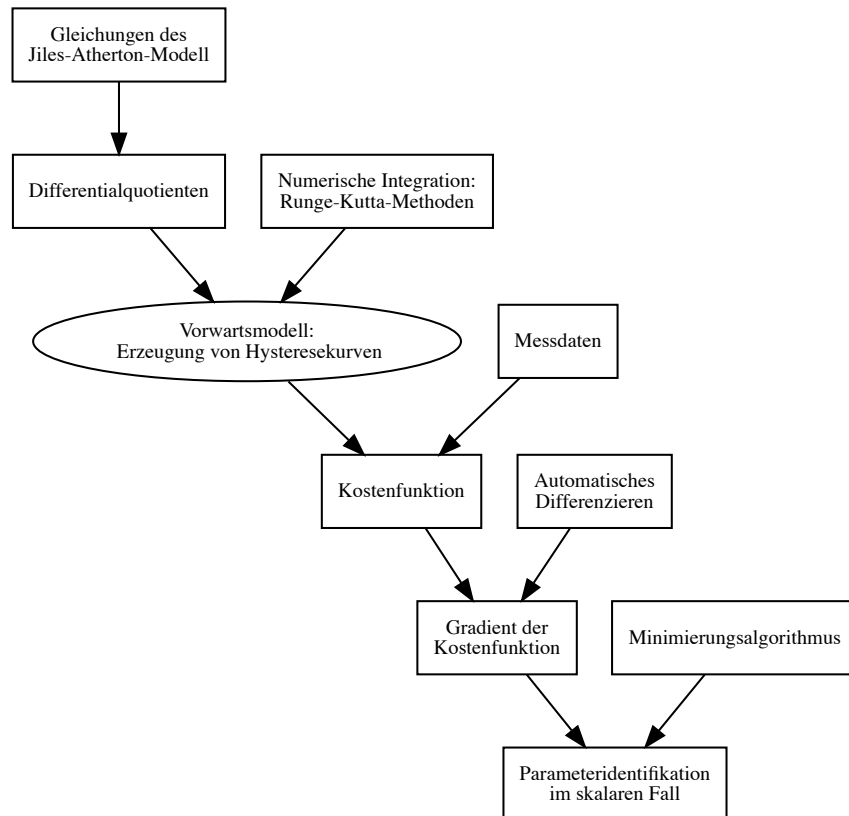


Abbildung 1: Vorgehen zur Parameteridentifikation im Fall ohne Ortsauflösung.

Ausgangspunkt sind die konstituierenden Gleichungen des Jiles-Atherton-Modells, aus denen man die Differentialquotienten gewinnen kann, die man für die numerische Integration benötigt. Mit dieser können Hysteresekurven zu gegebenen Parametern erzeugt werden. Die Differenz von erzeugten Hysteresekurven und den Messdaten führen zur Kostenfunktion, die mit der Methode des automatischen Differenzierens nach den Parametern abgeleitet werden kann. Mit einem Minimierungsalgorithmus kann man das Optimierungsproblem lösen und so die Parameter identifizieren.

vorhanden sind. Außerdem hat es eine recht kleine Parameteranzahl, was einige Vereinfachungen mit sich bringt. Dieses Modell wird in Abschnitt 2 genauer dargestellt.

Skalares Vorwärtsmodell Der nächste Schritt ist, das Vorwärtsmodell für den skalaren Fall (keine Ortsauflösung von Feld und Hysterese) zu entwerfen. Dieses Vorwärtsmodell ist effektiv eine Abbildung vom Parameterraum in die Menge der Hysteresekurven – für jede Parameterkombination erhält man eine (zeitlich diskretisierte) B-H-Kurve.

Integration Dies wird durch die Integration der differentiellen Magnetisierung, die das Jiles-Atherton-Modell beschreibt, erreicht. Da die verfügbaren AD-Tools nicht mit den integrierten Algorithmen der üblichen Mathematik-Pakete umgehen können, muss diese Integration manuell implementiert werden. Als Algorithmus wird ein Runge-Kutta-Verfahren gewählt, welches in einem Fachartikel als geeignet für das Jiles-Atherton-Modell beschrieben wurde. Eine kurze Übersicht zum Thema der numerischen Integration und den Runge-Kutta-Verfahren ist in Abschnitt 3 gegeben.

Inverses Problem Der nächste Schritt ist der Entwurf des inversen Problems: Gesucht ist nun das Urbild einer Hysteresekurve (Messung) im Parameterraum. Dazu wird zuerst eine Kostenfunktion aufgestellt, von der anschließend mit den Methoden des automatischen Differenzierens Jacobi- und Hessematrix ermittelt werden. Daraufhin wird die Kostenfunktion mit einem geeigneten Algorithmus minimiert. Dadurch erhält man die optimierten Parameter zur gegebenen Messkurve.

Ortsaufgelöstes Vorwärtsmodell Die bisherigen Berechnungen galten stets für ein skalares Modell, der Kern wurde also betrachtet als würde in jedem Punkt das gleiche Feld herrschen. Dies ist jedoch natürlich nur eine Näherung. Daher soll im nächsten Schritt das Jiles-Atherton-Modell auf einen ausgedehnten Kern übertragen werden. Die Zeitdiskretisierung wird genauso wie vorher mit finiten Differenzen (Runge-Kutta-Methode) vorgenommen, die Ortsdiskretisierung basiert auf der Finite-Elemente-Methode. Mit dem ortsaufgelösten Vorwärtsmodell kann nun die Hysterese im gesamten Kern ortsaufgelöst und unter Berücksichtigung der Maxwell-Gleichungen modelliert werden.

Inverses orts aufgelöstes Problem Wie beim skalaren Modell gilt es nun, das inverse Problem zu implementieren. Hierfür wird abermals auf das Automatische Differenzieren zurückgegriffen – in diesem Fall ist bei einem einzelnen Zeitschritt jedoch nicht eine einzelne Gleichung zu lösen, sondern ein ganzes FEM-Problem. Mit der Software *dolfin-adjoint* kann jedoch die Differenziation von FEM-Modellen abstrahiert und so stark vereinfacht werden. Die Kostenfunktion wird nun zu einem Kostenfunktional, also einer Funktion der Lösung der einzelnen FEM-Probleme aus den einzelnen Zeitschritten. Mit einem geeigneten Optimierungsalgorithmus kann nun abermals aus Messwerten auf die Parameter geschlossen werden.

2 Jiles-Atherton-Modell

Das Jiles-Atherton-Modell ist ein physikalisch motiviertes Modell zur Beschreibung der Hysterese von magnetischen Materialien.

2.1 Konstituierende Gleichungen

$$H_e = H + \alpha M \quad \text{Effektives Feld} \quad (1)$$

$$M_{\text{an}} = M_{\text{sat}} L\left(\frac{H_e}{a}\right) \quad \text{Anhysterische Magnetisierung} \quad (2)$$

$$\frac{dM_{\text{irr}}}{dH_e} = \frac{M_{\text{an}} - M_{\text{irr}}}{k \operatorname{sign}\left(\frac{dH}{dt}\right)} \quad \text{Pinning} \quad (3)$$

$$M = M_{\text{rev}} + M_{\text{irr}} \quad \text{Gesamte Magnetisierung} \quad (4)$$

$$M_{\text{rev}} = c(M_{\text{an}} - M_{\text{irr}}) \quad \text{Irreversible Magnetisierung} \quad (5)$$

In diesen fünf Modellgleichungen kommen fünf Modellparameter vor:

α Interdomänenkopplung

a Domänenwanddicke

M_{sat} Sättigungsmagnetisierung

k Pinning-Energie

c Magnetisierungsreversibilität

Die Funktion $L(x)$ ist die sogenannte Langevin-Funktion, die die anhysterische Magnetisierungskurve beschreibt. Sie ist definiert durch

$$L(x) = \coth(x) - \frac{1}{x} \quad (6)$$

2.2 Herleitung der Differentiale

Ziel ist es nun, einen differentiellen Zusammenhang zwischen B und H zu finden. Dazu:

$$M \stackrel{(4)}{=} M_{\text{rev}} + M_{\text{irr}} \stackrel{(5)}{=} c(M_{\text{an}} - M_{\text{irr}}) + M_{\text{irr}} = cM_{\text{an}} + (1 - c)M_{\text{irr}}. \quad (7)$$

Es gilt also

$$dM = (1 - c) dM_{\text{irr}} + c dM_{\text{an}} \quad (8)$$

und

$$M_{\text{irr}} = \frac{M - cM_{\text{an}}}{1 - c}. \quad (9)$$

Weiterhin stimmt jeweils

$$dM_{\text{irr}} = \frac{dM_{\text{irr}}}{dH_e} dH_e \quad (10)$$

$$dM_{\text{an}} = \frac{dM_{\text{an}}}{dH_e} dH_e \quad (11)$$

$$dH_e \stackrel{(1)}{=} dH + \alpha dM. \quad (12)$$

Man erhält nun für dM zusammengefasst:

$$\begin{aligned} dM &\stackrel{(8)}{=} (1 - c) dM_{\text{irr}} + c dM_{\text{an}} \\ &= (1 - c) \frac{dM_{\text{irr}}}{dH_e} dH_e + c \frac{dM_{\text{an}}}{dH_e} dH_e, \end{aligned} \quad (13)$$

in dieser Darstellung sind nun die aus den konstituierenden Gleichungen 2 und 3 leicht zugänglichen Größen dM_{irr}/dH_e und dM_{an}/dH_e enthalten.

$$\Rightarrow \frac{dM}{dH_e} = (1 - c) \frac{dM_{\text{irr}}}{dH_e} + c \frac{dM_{\text{an}}}{dH_e} \quad (14)$$

Dabei gilt mit Gleichung 2:

$$\frac{dM_{\text{an}}}{dH_e} = \frac{M_{\text{sat}}}{a} \frac{dL(H_e/a)}{d(H_e/a)} \quad (15)$$

Mit Gleichung 12 erhält man:

$$\begin{aligned}
 dM &= \frac{dM}{dH_e} dH_e \\
 &= \frac{dM}{dH_e} (dH + \alpha dM) \\
 \Rightarrow \frac{dM}{dH} &= \frac{dM/dH_e}{1 - \alpha dM/dH_e}
 \end{aligned} \tag{16}$$

Aus $B = \mu_0(H + M)$ folgt nun

$$dB = \mu_0(dH + dM) \tag{17}$$

und somit

$$\begin{aligned}
 \frac{dB}{dH} &= \mu_0 \left(1 + \frac{dM}{dH} \right) \\
 &= \mu_0 \frac{1 + (1 - \alpha) dM/dH_e}{1 - \alpha dM/dH_e}.
 \end{aligned} \tag{18}$$

Entsprechend gilt für die vom B -Feld abhängige Formulierung

$$\frac{dH}{dB} = \frac{1}{\mu_0} \frac{1 - \alpha dM/dH_e}{1 + (1 - \alpha) dM/dH_e} \tag{19}$$

Abhängigkeiten Die Differentiale sind, wenn man alle Terme ausschreibt, abhängig von B , H , dH/dt und dem Parametervektor $\vec{p} = (\alpha, a, M_{\text{sat}}, k, c)^T$:

$$\frac{dB}{dH} = f \left(B, H, \frac{dH}{dt}, \vec{p} \right) \tag{20}$$

3 Numerische Integration

Das Jiles-Atherton-Modell beschreibt – wie oben gezeigt – nur einen differentiellen Zusammenhang. Um Aussagen über den zeitlichen Verlauf der Magnetisierung und somit über die Kernverluste zu gewinnen, muss die jeweilige gewöhnliche Differentialgleichung integriert werden, zum Beispiel

$$B(H) = \int \frac{dB}{dH} dH + \text{const.} \tag{21}$$

Diese Integration ist analytisch nicht zugänglich und muss numerisch ausgeführt werden. Man verwendet hierfür typischerweise integrierte Lösungen wie den Solver *ode45* in Matlab oder das Python-Äquivalent *scipy.integrate.ode*. Diese Methoden implementieren die üblichen Verfahren, explizit das Runge-Kutta-Fehlberg-Verfahren. Dieses ist eine Variante des klassischen Runge-Kutta-Verfahrens mit adaptiver Schrittweite.

Da im Nachfolgenden jedoch das Ergebnis einer solchen Integration nach dem Parametervektor \vec{p} differenziert werden soll, benötigt man einen Integrationsalgorithmus, der den Werkzeugen der Automatischen Differenziation zugänglich ist. Aus diesem Grund muss die Integration manuell implementiert werden.

3.1 Explizites Euler-Verfahren

Das einfachste Integrationsverfahren ist das explizite Euler-Verfahren (Forward-Euler, Polygonzugverfahren), es entspricht dem einfachsten expliziten Finite-Differenzen-Ansatz:

$$\begin{aligned} \frac{dy}{dt} &= f(y, t) \\ \Rightarrow \frac{y_n - y_{n-1}}{t_n - t_{n-1}} &\approx f(y_{n-1}, t_{n-1}) \end{aligned} \quad (22)$$

So erhält man für den jeweils nächsten Zeitschritt

$$y_n \approx y_{n-1} + f(y_{n-1}, t_{n-1})(t_n - t_{n-1}) \quad (23)$$

3.2 Implizites Euler-Verfahren

Im vorherigen Abschnitt wurde ein explizites Verfahren gewählt, d.h. die Funktion $f(y, t)$ wurde durch bereit bekannte Werte von y und t angenähert, $f(y, t) \approx f(y_{n-1}, t_{n-1})$. Alternativ kann man auch einen impliziten Ansatz wählen, $f(y, t) \approx f(y_n, t_n)$, dann erhält man eine implizite Gleichung für den nächsten Zeitschritt

$$y_n = y_{n-1} + f(y_n, t_n)(t_n - t_{n-1}). \quad (24)$$

Diese Gleichung ist jedoch – da sie implizit und im Allgemeinen nichtlinear ist – ungleich schwerer zu lösen als die explizite, man benötigt hier üblicherweise numerische Methoden wie die Fixpunktiteration oder etwa, falls der Gradient vorhanden ist, das Newton-Verfahren.

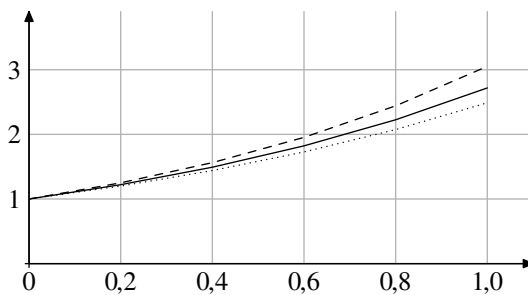


Abbildung 2: Vergleich von einfachen numerischen Integrationsverfahren (vgl. ODE 25). Die durchgezogene Linie ist die exakte Lösung, gepunktet sieht man das Resultat des expliziten Euler-Verfahrens, das implizite Verfahren ist gestrichelt dargestellt. Die vertikalen Gitterlinien beschreiben die diskretisierten Zeiten.

Der erhöhte Rechenaufwand der impliziten Verfahren wird durch eine erhöhte Stabilität ausgeglichen, insbesondere bei der Problemklasse der steifen Probleme. Die Integration des Jiles-Atherton-Modells stellt jedoch keine hohen Anforderungen an die Stabilität, wodurch explizite Verfahren verwendet werden können.

3.3 Kurzer Vergleich

Der Nachteil von diesen einfachen Verfahren ist ihre geringe Genauigkeit, bzw. die sehr kleine Schrittweite und somit der erhöhte Rechenaufwand, welche man wählen muss, um eine ausreichende Genauigkeit zu erhalten. Als Beispiel soll nun die wohl einfachste Differentialgleichung

$$\frac{dy}{dt} = y(t), \quad y(0) = 1 \quad (25)$$

mit der Lösung $y(t) = e^t$ betrachtet werden.

Wie man in Abbildung 2 erkennt, akkumulieren sich in beiden Verfahren (bei großer Schrittweite) schnell große Fehler. Das explizite Verfahren nutzt die Steigung im aktuellen Punkt für den nächsten Schritt, unterschätzt bei der e-Funktion also stets die Steigung, während das implizite Verfahren jeweils die Steigung im nächsten Punkt als Näherung verwendet. Die führt in diesem Fall jeweils zu einer Überschätzung der Steigung.

Um die Schrittweite nicht zu klein wählen zu müssen und somit den Rechenaufwand zu reduzieren, wurden viele verschiedene Integrationsverfahren entworfen. Die meisten sind Einschnitt-Verfahren, wie etwa die Runge-Kutta-Methoden – für den nächsten Zeitschritt werden dabei jedoch Zwischenschritte berechnet, die die Näherung verbessern. Man erhöht so zwar den Aufwand für einen einzelnen Schritt, kann aber dafür die Schrittweite deutlich größer wählen.

Eine weitere Möglichkeit, den Rechenaufwand zu reduzieren ist eine adaptive Schrittweitensteuerung: Man schätzt mit einem weiteren Zwischenschritt den

Fehler und kann so die Schrittweite jeweils so groß wie möglich wählen (vgl. Runge-Kutta-Fehlberg-Methode).

Die Wahl des richtigen Algorithmus ist stets vom genauen Problem abhängig, für die vorliegende Integration des Jiles-Atherton-Modells eignet sich laut [1] jedoch der viel verwendete RK4-Algorithmus mit fester Schrittweite.

3.4 Runge-Kutta-Methoden

Die Euler-Verfahren sind Sonderfälle der Runge-Kutta-Methoden, welche eine wichtige Klasse von Einschrittverfahren darstellen. Der m -stufige Algorithmus benötigt Gewichte $\alpha_i, \gamma_i, 1 \leq i \leq m$ und $\beta_{i,l}, 1 \leq i, l \leq m$. Weiterhin müssen die Schrittweiten h_i vorgegeben werden, zur Vereinfachung soll jedoch eine feste Schrittweite h angenommen werden.

Für den nächsten Datenpunkt berechnet man:

$$\begin{aligned}
 \text{Nächster Zeitpunkt} \quad t_n &= t_{n-1} + h \\
 \text{i-ter Hilfwert} \quad k_i &= f \left(y_{n-1} + h \sum_{l=1}^m \beta_{i,l} k_l, t_{n-1} + \alpha_i h \right) \\
 \text{Nächster Wert} \quad y_n &= y_{n-1} + h \sum_{l=1}^m \gamma_l k_l
 \end{aligned} \tag{26}$$

Die Gewichte werden meist in einem sogenannten Butcher-Tableau dargestellt

$$\begin{array}{c|ccc}
 \alpha_1 & \beta_{1,1} & \cdots & \beta_{1,m} \\
 \alpha_2 & \beta_{2,1} & \cdots & \beta_{2,m} \\
 \vdots & \vdots & \ddots & \vdots \\
 \alpha_m & \beta_{m,1} & \cdots & \beta_{m,m} \\
 \hline
 & \gamma_1 & \cdots & \gamma_m
 \end{array}$$

Die entstehenden Gleichungen für die Hilfwerte k_i sind im Allgemeinen implizit, sie müssten also iterativ gelöst werden. Sind jedoch die Werte k_i nur von schon vorher berechneten Werten $k_j, j < i$ abhängig, entsteht ein explizites Verfahren. Die Koeffizienten $\beta_{i,l}$ lassen sich im Butcher-Tableau nun als untere Dreiecksmatrix anordnen – das klassische Runge-Kutta-Verfahren (RK4) hat

beispielsweise das Tableau

$$\begin{array}{c|cccc}
 0 & & & & \\
 \frac{1}{2} & \frac{1}{2} & & & \\
 \frac{1}{2} & 0 & \frac{1}{2} & & \\
 1 & 0 & 0 & 1 & \\
 \hline
 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6}
 \end{array} \quad (27)$$

Schreibt man für dieses die einzelnen Schritte aus, so ist die Berechnung des nächsten Wertes im RK4-Verfahren:

$$\begin{aligned}
 t_n &= t_{n-1} + h \\
 k_1 &= f(y_{n-1}, t_{n-1}) \\
 k_2 &= f(y_{n-1} + hk_1/2, t_{n-1} + h/2) \\
 k_3 &= f(y_{n-1} + hk_2/2, t_{n-1} + h/2) \\
 k_4 &= f(y_{n-1} + hk_3, t_{n-1} + h) \\
 y_n &= y_{n-1} + \frac{h}{6} (k_1 + 2k_2 + 2k_3 + k_4)
 \end{aligned} \quad (28)$$

3.5 Integration des Jiles-Atherton-Modells

Genau dieses klassische Runge-Kutta-Verfahren soll nun auf das Jiles-Atherton-Modell angewendet werden. Das Differential dB/dH aus Gleichung 18 wird nun als die zu integrierende Funktion f betrachtet. Es gilt also

$$dB = f\left(B, H, \frac{dH}{dt}, \vec{p}\right) dH. \quad (29)$$

Als Eingabe benötigt der Algorithmus je ein Array für das magnetische Feld H und für die Zeitpunkte t , weiterhin eines für die Parameter. Da das RK4-Verfahren für jeden Schritt auch Funktionsauswertungen bei $t_n + h/2$ benötigt, müssen auch für H Werte bei $t_n + h/2$ vorliegen. Dies wird erreicht, indem die Schritte der Integration jeweils zwei Zeitschritte der Eingabedaten umfassen:

H_{2n} und B_n beschreiben stets den gleichen Zeitpunkt t_{2n} . Für B existieren somit nur zu jedem zweiten Zeitpunkt Werte. Das Ergebnis B hat entsprechend nur die halbe Samplingrate der Eingangsdaten.

Dabei ist wichtig, dass der Zeitpunkt t_{2n+1} stets genau in der Mitte zwischen t_{2n} und t_{2n+2} für alle $n \in \mathbb{N}$ liegt. Da Messdaten typischerweise ohnehin äquidistant

vorliegen ist das keine starke Einschränkung.
Entsprechend der Definition in 28 gilt nun

$$\begin{aligned}
 h &= H_{2n} - H_{2n-2} \\
 k_1 &= f\left(B_{n-1}, \quad H_{2n-2}, \quad \left(\frac{dH}{dt}\right)_{2n-2}, \quad \vec{p}\right) \\
 k_2 &= f\left(B_{n-1} + h\frac{k_1}{2}, \quad H_{2n-1}, \quad \left(\frac{dH}{dt}\right)_{2n-1}, \quad \vec{p}\right) \\
 k_3 &= f\left(B_{n-1} + h\frac{k_2}{2}, \quad H_{2n-1}, \quad \left(\frac{dH}{dt}\right)_{2n-1}, \quad \vec{p}\right) \\
 k_4 &= f\left(B_{n-1} + hk_3, \quad H_{2n}, \quad \left(\frac{dH}{dt}\right)_{2n}, \quad \vec{p}\right) \\
 B_n &= B_{n-1} + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)
 \end{aligned} \tag{30}$$

Dies beschreibt jeweils einen Schritt des Integrationsverfahrens. Führt man diesen für alle Zeitpunkte durch, erhält man die Magnetisierung M für die gewählte Feldvorgabe H und die Modellparameter \vec{p} .

Der beschriebene Algorithmus wurde, um eine hohe numerische Effizienz zu erzielen, in der Programmiersprache C++ implementiert. Dabei wurde jedoch nur die Variante mit dem H-Feld verwendet, da die andere numerisch sehr instabil erschien. Um diese funktionsfähig zu machen muss vermutlich auf implizite Verfahren zurückgegriffen werden. Eine genauere Untersuchung dieses Problems steht jedoch aus.

3.6 Numerische Details

Um die Stabilität des Algorithmus zu optimieren, gibt es einige Details, die beachtet werden müssen.

Die Langevin-Funktion Die Definition der Langevin-Funktion lautet

$$L(x) = \coth(x) - \frac{1}{x}, \tag{31}$$

die beiden Terme haben jeweils in der Umgebung der 0 eine Polstelle, sind dabei jedoch etwa gleich groß. Durch die sehr großen Werte, die sich jedoch

fast vollständig kompensieren tritt bei numerischen Berechnungen unweigerlich ein Signifikanzverlust auf. Entgegenwirken kann man diesem durch eine Approximation in der Nähe der 0. Es gilt laut [2]

$$\coth(x) \approx \frac{1}{x} + \frac{x}{3} - \frac{x^3}{45} + \dots, \quad (32)$$

für die Langevinfunktion folgt also in der Nähe der 0

$$L(x) \approx \frac{x}{3}. \quad (33)$$

Entsprechend folgt für die Ableitung:

$$\frac{dL(x)}{dx} = \frac{1}{x^2} - \frac{1}{\sinh(x)^2} \approx \frac{1}{3} - \frac{1}{15}x^2 + \dots \quad (34)$$

4 Anwendung des Jiles-Atherton-Modells

Im folgenden Abschnitt sollen nun einige simulierte Kurven präsentiert und anhand dieser die Eigenschaften des Modells erläutert werden.

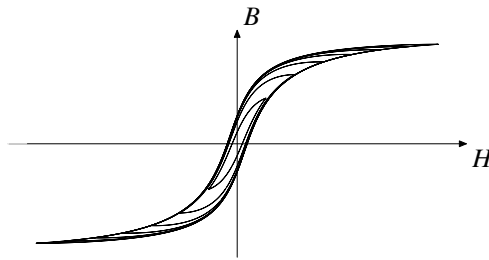


Abbildung 3: Schar von Hysteresekurven für verschiedene Aussteuerungen. Der nicht-hysterische Teil wird von der Langevinfunktion festgelegt.

Hysteresekurven Als erstes Beispiel dient eine Kurvenschar, die Hysteresekurven für verschiedene Aussteuerungen zeigt (Abbildung 3). Man erkennt, dass das Modell die Charakteristik typischer ferromagnetischer Hysteresekurven nachbildet:

Bei hohen Feldstärken wächst die Induktion stark an, dies ist die typische Sättigungserscheinung bei Kernmaterialien. Nachdem der Umkehrpunkt erreicht wurde, läuft die Kurve auf einem anderen Weg zurück (Hyterese), sodass die Flussdichte bei verschwindender Feldstärke nicht mehr den Wert 0 annimmt, sondern eine Restmagnetisierung (Remanenz) übrig bleibt. Im weiteren Verlauf

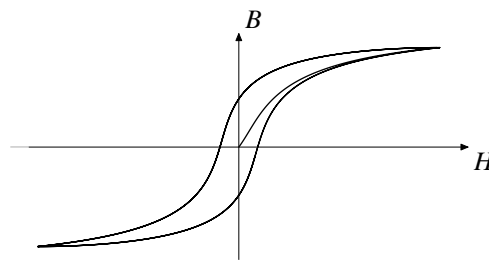


Abbildung 4: Einzelne Hysteresekurve mit Neukurve.

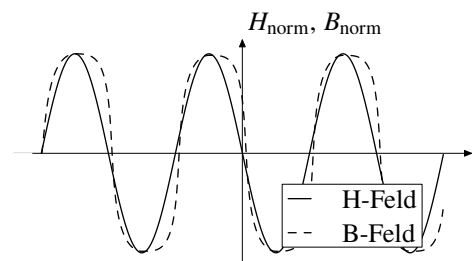
kreuzt die Flussdichte nun die H -Achse, den entsprechenden Achsenabschnitt nennt man Koerzitivfeldstärke.

In Abbildung 4 ist eine einzelne Hysteresekurve gezeigt. In diesem Fall ist auch die Neukurve dargestellt, die vom Modell ebenfalls repliziert wird. Im Gegensatz zu den meisten Messkurven beginnt die Neukurve im Jiles-Atherton-Modell jedoch näherungsweise linear.

4.1 Stromanregung

Wie oben gezeigt, kann man entweder dB/dH oder dH/dB integrieren. Gibt man einen Strom vor, so ist dH festgelegt, man verwendet also dB/dH . Wird der Kern mit einer Spannung angeregt, nutzt man dH/dB . Im Allgemeinen wird man den Kern ohnehin eingebettet in einem Netzwerk betrachten und auch die Integration entsprechend anpassen.

Abbildung 5: Verlauf von H - und B -Feld bei sinusförmiger Stromanregung. Durch die nicht-linearen Kerneigenschaften wird die Form des Sinus stark verbreitert, entsprechend treten hohe Steigungen bei den Nulldurchgängen der magnetischen Flussdichte auf.



In Abbildung ist der Verlauf von B und H -Feld bei Anregung mit einem sinusförmigen Strom gezeigt – die Sättigungscharakteristik rundet die Spitzen des Sinus deutlich ab, je mehr die Aussteuerung in die Sättigung geht, desto mehr nähert sich die zeitliche Form des B -Feldes einer Rechteckschwingung an. Da das Induktionsgesetz die induzierte Spannung mit der zeitlichen Ableitung der magnetischen Flussdichte verknüpft, wird deutlich, dass hier Probleme mit

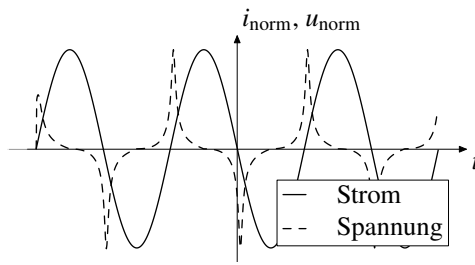


Abbildung 6: Verlauf von Strom und Spannung bei sinusförmiger Stromanregung. Die steilen Flanken der Hysteresekurve führen zu großen Spannungsspitzen durch $\nabla \times E = -\partial B / \partial t$.

Überspannungen auftreten können – die steilen Flanken im Verlauf des B -Feldes führen zu Spannungsspitzen, die die Isolation des induktiven Bauelements beschädigen können.

Die Anregung mit Strömen tritt in der Praxis nur selten auf, weshalb dieses Problem nur in einigen Spezialfällen Probleme verursachen dürfte, ein Beispiel wären Stromwandler, die sehr große Wechselströme heruntertransformieren, um sie Messgeräten zuzuführen.

4.2 Spannungsanregung

Der weitaus üblichere Fall ist, dass der Kern näherungsweise von einer Spannungsquelle angeregt wird. Dies ist etwa der typische Fall in einem Transformator, bei dem der Magnetisierungsstrom in erster Näherung von der Primärspannung abhängt.

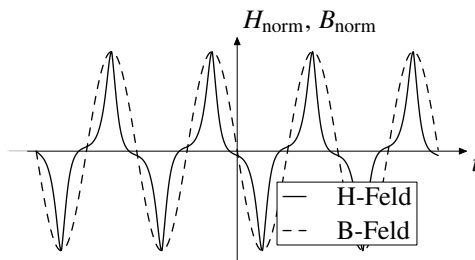


Abbildung 7: Verlauf von H - und B -Feld bei sinusförmiger Spannungsanregung. Der typische Verlauf des Magnetisierungsverlauf wird deutlich – kommt die Induktion in die Nähe der Sättigung, steigt das H -Feld und somit der Strom stark an.

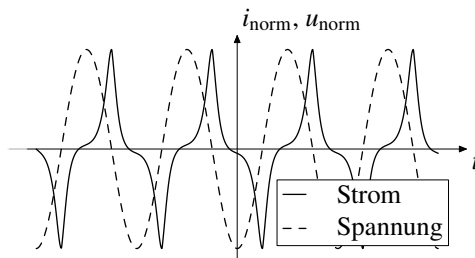


Abbildung 8: Verlauf von Strom und Spannung bei sinusförmiger Spannungsanregung.

Wie man in den Abbildungen oder auch den Hysteresekurven sieht, wächst der Strom für hohe Induktionen extrem stark an, die Sättigung tritt ein und verursacht enorm hohe Verluste in den Leitern. Daher muss dieser Zustand bei der Auslegung von induktiven Bauelementen im Allgemeinen unbedingt vermieden werden.

4.3 Allgemeiner Fall

Im allgemeinen Fall kann man das induktive Bauelement natürlich nicht isoliert betrachten. Vielmehr muss man das Modell des Kerns in ein allgemeines Schaltungsmodell einbinden und dieses transient lösen. So kann man dann auch diverse Themen wie das Einschaltverhalten, das Schaltungsverhalten an der Grenze zur Sättigung oder die Kernverluste im Betrieb untersuchen.

5 Parameteridentifikation

Wie am Anfang beschrieben wurde, ist das Ziel dieses Projektes einerseits, das Jiles-Atherton zu untersuchen, andererseits aber vor allem, eine neue Möglichkeit zu finden, um die Modellparameter aus Messdaten zu extrahieren.

Der neue Ansatz ist hierbei, die Methoden des algorithmischen Differenzierens zu nutzen, um Jacobi- und Hesse-Matrix für Optimierungsalgorithmen bereitzustellen.

5.1 Least-Squares-Methode

Die Standardmethode um Modelle an Messdaten anzupassen, ist die Methode der kleinsten Fehlerquadrate:

Man nimmt an, dass die beste Übereinstimmung von Modell und Messung gegeben ist, wenn die Summe der quadratischen Abweichungen minimal ist. Ist $f(\vec{x}, \vec{p})$ die Modellfunktion, die von Eingangswerten \vec{x} und Parametern \vec{p} abhängt, so muss man das Minimum der Kostenfunktion

$$\text{cost}(\vec{p}) = \sum_i (f(x_i, \vec{p}) - y_i)^2 \quad (35)$$

finden, um die beste Übereinstimmung des Modells mit den gemessenen Daten \vec{y} zu erhalten.

Die zu lösende mathematische Aufgabe ist also die Minimierung einer im Allgemeinen nichtlinearen und nicht konvexen Funktion. Für die Lösung dieser

Problemstellung wurden diverse Algorithmen entworfen, die sich grob in zwei Klassen unterteilen lassen können: Ableitungsfreie Methoden und Methoden, die Ableitungen – zumindest die Erste – der zu minimierenden Funktion benötigen. Im Allgemeinen sind Methoden, die Ableitungen nutzen, erfolgreicher, da sie für den nächsten Optimierungsschritt jeweils über Informationen verfügen, in welcher Richtung der Umgebung des aktuellen Schätzwertes die Kostenfunktion am steilsten abfällt. So kann insbesondere die Zahl der Funktionsauswertungen reduziert werden, was insbesondere dann relevant ist, wenn die Funktion rechnerisch teuer ist. Daher kommen gradientenfreie Verfahren üblicherweise erst dann zum Einsatz, wenn Gradientenverfahren nicht mehr einsetzbar sind – dies ist dann der Fall, wenn Informationen über die Ableitungen nur schwer zu erhalten sind.

5.2 Probleme bei der Parameteridentifikation für das Jiles-Atherton-Modell

In der Literatur zum Jiles-Atherton-Modell finden sich viele Ansätze, um mittels gradientenfreier Verfahren auf die Parameter zu schließen. Bei fast all diesen Ansätzen gibt es jedoch ein entscheidendes Problem: Die Geometrie der Probe wird schlicht vernachlässigt – während man dies bei toroidförmigen Kernen herausrechnen kann, führt es bei Kernen mit keiner derartig großen Symmetrie zu Fehlern. Dies wirkt sich insbesondere auf die Messung von Schichtblech- und Schnittbandkernen aus, da diese nicht toroidförmig erhältlich sind.

Die Hysteresekurve wird in solchen Geometrien an verschiedenen Punkten im Material mit verschiedenen Phasenlagen und Amplituden durchlaufen, die messbaren Größen, Spannung und Strom, lassen somit nur einen Schluss auf die integrale und somit geometrieabhängige Hysteresekurve zu.

Ziel dieser Arbeit ist es nun, die Geometrie in die Rechnung einzubeziehen. Dazu muss das Jiles-Atherton-Modell in eine orts- und zeitaufgelöste Simulation integriert werden (vgl. [3], [4]), anschließend müssen Strom und Spannung aus dem Modell gewonnen und mit den Messdaten in die Kostenfunktion überführt werden.

Offensichtlich ist nun die ursprünglich numerisch wenig aufwändige Funktionsauswertung des Vorwärtsmodells um Größenordnungen teurer geworden. Ein gradientenfreies Verfahren, welches eine kleinere Konvergenzrate hat, dafür aber keine Gradienten berechnen muss, kommt nun nicht mehr infrage – die Zahl der Funktionsauswertungen ist um Größenordnungen größer als bei Gradientenverfahren.

Um das Ziel der Parameteridentifikation unter Beachtung der Geometrie zu ermöglichen, muss also eine Möglichkeit gefunden werden, die Ableitung(en) der Kostenfunktion zu bestimmen. Dafür gibt es drei Möglichkeiten:

- Symbolische Berechnung
- Numerische Berechnung (Differenzenquotienten)
- Automatisches Differenzieren

Um die Ableitungen symbolisch zu bestimmen, müssten die relativ komplizierten Gleichungen einzeln differenziert und die jeweiligen Funktionen ebenfalls in den Programmcode geschrieben werden. Dies stellt gerade bei großer Parameterzahl einen enormen Entwicklungsaufwand dar, ist fehleranfällig und meist relativ ineffizient.

Die Effizienz ist bei der Berechnung mittels Differenzenquotienten kein Problem – sie ist hier besonders hoch. Im Kontrast dazu ist jedoch die Genauigkeit sehr klein: Ist die Schrittweite sehr klein, liegen die zwei Funktionswerte eines Quotienten sehr nah beieinander – dies kann z.B. dazu führen, dass sich zwei Zahlen, die jeweils 32 Bit breit sind, nur in den letzten 5 Stellen unterscheiden. Ein enormer Genauigkeitsverlust ist die Folge.

Einen Mittelweg zwischen Geschwindigkeit und Genauigkeit bietet die dritte Methode, das automatische Differenzieren.

6 Automatisches Differenzieren

Die Grundidee des automatischen Differenzierens ist relativ simpel: Die Ableitung einer gegebenen Funktion wird mithilfe der Kettenregel so lange umgeformt, bis nur noch Basisfunktionen vorkommen, für die die Ableitung hinterlegt ist.

Man unterscheidet zwischen zwei grundlegenden Vorgehensweisen, *Forward*- und dem *Reverse-Mode*. Um den Unterschied zwischen diesen zu veranschaulichen betrachten wir die Funktion

$$f(x_1, x_2) = \frac{\sin(x_1/x_2)}{\exp(x_1/x_2)}. \quad (36)$$

Es ist klar einsehbar, dass man diese Funktion als Graph darstellen kann, der die Abfolge der einzelnen Rechenschritte veranschaulicht: Die Kanten stellen Eingangs- und Zwischenwerte dar, die Knoten die Rechenoperatoren. Der Graph

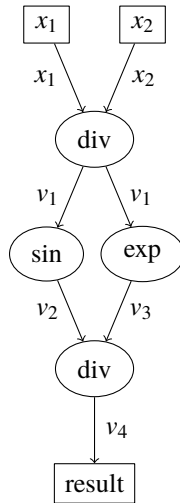


Abbildung 9: Darstellung der Beispielfunktion aus Gleichung 36 als Rechengraph (*computational graph*). Es gilt $v_1 = x_1/x_2$, $v_2 = \sin(v_1)$, $v_3 = \exp(v_1)$ und $v_4 = v_2/v_3$.

ist in Abbildung 9 dargestellt. Für die Zwischenwerte gilt:

$$v_1 = \frac{x_1}{x_2} \quad v_2 = \sin(v_1)$$

$$v_3 = \exp(v_1) \quad f = v_4 = \frac{v_2}{v_3}$$

Aufgabe ist es nun, aus diesem Graphen die Ableitung der Funktion (also dem Wert v_4) nach den Eingangsparametern x_1 und x_2 zu bestimmen. Für die Anwendung der beiden im Folgenden dargestellten Algorithmen müssen neben der Struktur der Funktion lediglich die Ableitungen der Operatoren (runde Knoten) bekannt sein. Es gilt:

$$\frac{\partial v_1}{\partial x_1} = \frac{1}{x_2} \quad \frac{\partial v_1}{\partial x_2} = -\frac{x_1}{x_2^2} \quad \frac{\partial v_2}{\partial v_1} = \cos(v_1)$$

$$\frac{\partial v_3}{\partial v_1} = \exp(v_1) \quad \frac{\partial v_4}{\partial v_2} = \frac{1}{v_3} \quad \frac{\partial v_4}{\partial v_3} = -\frac{v_2}{v_3^2}.$$

Diese Ableitungen müssen hinterlegt werden. Da die Basisfunktionen jedoch in den meisten Fällen aus einer relativ kleinen Gruppe stammen, sind diese in den Paketen für automatische Differentiation meistens schon definiert. Für Funktionen, die darüber hinaus gehen, müssen der Operator und seine Ableitung bzw. Ableitungen registriert werden.

6.1 Forward Mode

Im *Forward-Mode* wertet man die Funktion nun genauso aus wie sonst – bezüglich der Klammern von innen nach außen. Dabei notiert man jedoch auch neben der

Funktion stets die Ableitungen der einzelnen Ausdrücke. Soll beispielsweise die Ableitung nach x_2 im Punkt (1, 2) ermittelt werden:

$$\begin{aligned}
 x_1 &= 1 \\
 \dot{x}_1 &= 0 \\
 x_2 &= 2 \\
 \dot{x}_2 &= 1 \\
 v_1 = x_1/x_2 &= 0,5 \\
 \dot{v}_1 = \frac{\partial v_1}{\partial x_1} \dot{x}_1 + \frac{\partial v_1}{\partial x_2} \dot{x}_2 = \frac{1}{x_2} \dot{x}_1 - \frac{x_1}{x_2^2} \dot{x}_2 = \frac{1}{2} 0 - \frac{1}{2^2} 1 &= -0,25 \\
 v_2 = \sin(v_1) = \sin(1/2) &= 0,479\,425\,539 \\
 \dot{v}_2 = \frac{\partial v_2}{\partial v_1} \dot{v}_1 = \cos(v_1) \dot{v}_1 = \cos(0,5)(-0,25) &= 0,219\,395\,640 \\
 v_3 = \exp(v_1) &= 1,648\,721\,271 \\
 \dot{v}_3 = \frac{\partial v_3}{\partial v_1} \dot{v}_1 = \exp(v_1) \dot{v}_1 &= -0,412\,180\,317 \\
 v_4 = \frac{v_2}{v_3} &= 0,290\,786\,288 \\
 \dot{v}_4 = \frac{\partial v_4}{\partial v_2} \dot{v}_2 + \frac{\partial v_4}{\partial v_3} \dot{v}_3 = \frac{1}{v_3} \dot{v}_2 - \frac{v_2}{v_3^2} \dot{v}_3 &= -0,060\,373\,610
 \end{aligned}$$

Wie man sieht, tritt durch das Differenzieren kein Signifikanzverlust auf, wie bei der Verwendung von Differenzenquotienten. Weiterhin ist der Algorithmus relativ einfach und somit nicht viel langsamer als die Berechnung der Ableitungen mit Differenzenquotienten. Das algorithmische Differenzieren schafft also den oben genannten Mittelweg zwischen symbolischer Differenziation und numerischer Differenziation, den Mittelweg zwischen Genauigkeit und Geschwindigkeit.

Der dargestellte Algorithmus weist jedoch im Fall großer Parameterzahlen einen Nachteil auf: Bei jedem Zwischenschritt muss sowohl die Funktionsauswertung selbst, als auch die Ableitung nach jedem Parameter gespeichert werden. Bei großen Parameterzahlen wird der Algorithmus viel Speicher und Rechenleistung brauchen, die Rechenzeit skaliert mit der Parameterzahl.

6.2 Reverse Mode

Einen Ausweg aus diesem Problem bietet der zweite Basisalgorithmus des automatischen Differenzierens, der *Reverse Mode*. Bei diesem steigt der Aufwand

für Ableitungen nach zusätzlichen Parametern nicht an, wie das folgende Beispiel deutlich macht. Als Funktion wird wieder die aus Gleichung 36 betrachtet. Wie der Name *Reverse Mode* sagt, wird der Informationsfluss für die Gradientenbildung rückwärts durchschritten. Damit dies möglich ist, muss also die Funktion vorher einmal in Vorwärtsrichtung ausgeführt und alle Zwischenwerte aufgenommen worden sein. Es gilt also:

$$\begin{aligned}x_1 &= 1 \\x_2 &= 2 \\v_1 &= 0,5 \\v_2 &= 0,479\,425\,539 \\v_3 &= 1,648\,721\,271 \\f = v_4 &= 0,290\,786\,288\end{aligned}$$

$$\begin{aligned}df &= \frac{\partial f}{\partial v_2} dv_2 + \frac{\partial f}{\partial v_3} dv_3 \\dv_2 &= \frac{\partial v_2}{\partial v_1} dv_1 = \cos(v_1) dv_1 \\dv_3 &= \frac{\partial v_3}{\partial v_1} dv_1 = \exp(v_1) dv_1 \\dv_1 &= \frac{\partial v_1}{\partial x_1} dx_1 + \frac{\partial v_1}{\partial x_2} dx_2 = \frac{1}{x_2} dx_1 - \frac{x_1}{x_2^2} dx_2\end{aligned}$$

Zusammengefasst folgt

$$df = \frac{1}{v_3} \cos(v_1) \left(\frac{1}{x_2} dx_1 - \frac{x_1}{x_2^2} dx_2 \right) - \frac{v_2}{v_3^2} \exp(v_1) \left(\frac{1}{x_2} dx_1 - \frac{x_1}{x_2^2} dx_2 \right). \quad (37)$$

Wie man sieht, erhält man sofort die Ableitungen nach den beiden Parametern:

$$\frac{df}{dx_1} = \frac{1}{v_3} \cos(v_1) \frac{1}{x_2} - \frac{v_2}{v_3^2} \exp(v_1) \frac{1}{x_2} \quad (38)$$

und

$$\frac{df}{dx_2} = \frac{1}{v_3} \cos(v_1) \left(-\frac{x_1}{x_2^2} \right) - \frac{v_2}{v_3^2} \exp(v_1) \left(\frac{x_1}{x_2^2} \right) \quad (39)$$

Für große Parameteranzahlen ist somit immer der *Reverse Mode* zu bevorzugen, der *Forward Mode* hat seine Vorteile dort, wo eine Vielzahl von Funktionen nach einem Parametersatz abgeleitet werden soll. Dies kommt in der Praxis üblicherweise seltener vor.

Ein zentrales Beispiel für die Ableitung einer Funktion nach sehr vielen Parametern ist die Backpropagation in Neuronalen Netzen aus dem Forschungsfeld

der künstlichen Intelligenz. Hier werden Netze aus nichtlinearen Funktionen programmiert, die näherungsweise jede nichtlineare Funktion nachbilden können. Die genaue Übertragungsfunktion eines solchen neuronalen Netzes wird dabei von den sogenannten *weights* und *biases* bestimmt, dies sind mehr oder weniger skalare Eingangsparameter der Funktion.

Nach genau diesen Gewichten wird abgeleitet, wenn ein neuronales Netz trainiert wird: Mittels Optimierungsalgorithmen werden die Gewichte so angepasst, dass die Trainingsdaten möglichst genau repliziert werden. Das Anlernen eines neuronalen Netzes ist also genau die gleiche Problemstellung wie die Anpassung von Modellparametern an Messdaten – lediglich die innere Struktur des Modells und des neuronalen Netzes unterscheiden sich.

6.3 Weiterführendes

Die beiden dargestellten Grundalgorithmen lassen sich auf beliebige numerische Programme anwenden – einzige Bedingung ist, dass für alle Grundoperationen die einfache Ableitung hinterlegt ist.

Auch Unterprogramme, Verzweigungen oder Schleifen sind kein Problem – der Rechengraph kann stets durch „Mitschreiben“ der ausgeführten Blöcke generiert werden. Eine typische Technik hierfür ist das Überladen von Datentypen und Operatoren – immer wenn ein Operator auf eine Variable angewandt wird, wird im Rechenbaum eine Kante für die Variable und ein Knoten für den Operator hinterlegt. Beliebig komplexe numerische Modelle können so in einen Rechengraphen überführt und anschließend mit den Methoden des automatischen Differenzierens abgeleitet werden.

Der andere Ansatz, um das automatische Differenzieren zu implementieren, ist die Quelltexttransformation. Statt den Rechengraphen zur Laufzeit aufzustellen, wird er aus dem vorliegenden Vorwärts-Algorithmus im Vorhinein generiert. Dies ist technisch schwieriger umzusetzen, bietet jedoch Vorteile in der Geschwindigkeit, da bei der anschließenden Kompilierung Optimierungen vorgenommen werden. Abgesehen von den beiden Basisalgorithmen wurden weitere Methoden entwickelt, um die Komplexität und damit die Rechenzeit der Algorithmen weiter zu reduzieren. Diese basieren oft auf der Idee der Taylor-Entwicklung oder der Operator-Algebra nach Heaviside.

7 Implementation für das Jiles-Atherton-Modell

Es gibt Aufsätze, in denen versucht wird, mit einem neuronalen Netz die Optimierung der Parameter des Jiles-Atherton-Modells vorzunehmen. Mathematisch betrachtet ist dies nur das Einbringen von zusätzlichen Freiheitsgraden/Variablen, zusätzlich zu den fünf Parametern α , a , M_{sat} , k und c . Sinnvoller erscheint es, die Hysterese selbst mit einem neuronalen Netz zu modellieren, wie es andere Aufsätze machen.

Die ersten Versuche, das Jiles-Atherton-Modell abzuleiten, fanden aufgrund der hohen Entwicklungsgeschwindigkeit in der Programmiersprache *Python* statt. Der hohen Entwicklungsgeschwindigkeit stand jedoch eine sehr kleine Rechengeschwindigkeit des resultierenden Algorithmus entgegen.

Aus diesem Grund wurden das Modell und die Differenziation in die Programmiersprache *C++* übertragen, welche numerisch deutlich effizienter ist. Um dennoch schnelle und einfache Auswertungen zu ermöglichen, wurde ein Wrapper implementiert, dadurch kann man den schnellen Algorithmus aus *Python* aufrufen.

Für die algorithmische Differenziation wurde auf das Paket *CppAD*¹ zurückgegriffen, welches sehr effiziente AD-Routinen bietet. *CppAD* wurde von Bradley M. Bell entwickelt und ist Teil des Mathematik-Software-Projekt *COIN-OR*²

7.1 Resultate

Um die Funktion der Algorithmen zu testen, wurden Messdaten simuliert, es wurden also mithilfe des Jiles-Atherton-Modells Kurven erzeugt, welche mit Rauschen versehen wurden. Anschließend wurden davon abweichende Startwerte für die Optimierung festgelegt und die Optimierung (unter Verwendung verschiedener Algorithmen) durchgeführt. Es konnte so gezeigt werden, dass die (simulierten) Messdaten in den meisten Fällen vom Programm repliziert werden konnten.

Als robuster Optimierungsalgorithmus hat sich TNC³ erwiesen, weitere Untersuchungen hierfür stehen jedoch noch aus. Neben der grundsätzlichen Funktionsfähigkeit konnte auch gesehen werden, dass die Verfahren, die Gradienten verwenden, tatsächlich deutlich weniger Funktionsaufrufe benötigen als gradientenfreie Verfahren wie zum Beispiel das Verfahren von Nelder und Mead. Dies

¹<https://www.coin-or.org/CppAD/>

²Computational Infrastructure for Operations Research, <https://www.coin-or.org>

³Truncated Newton Method

bestätigt den Weg, automatische Differenzierung für die Parameteridentifikation im Jiles-Atherton-Modell zu nutzen.

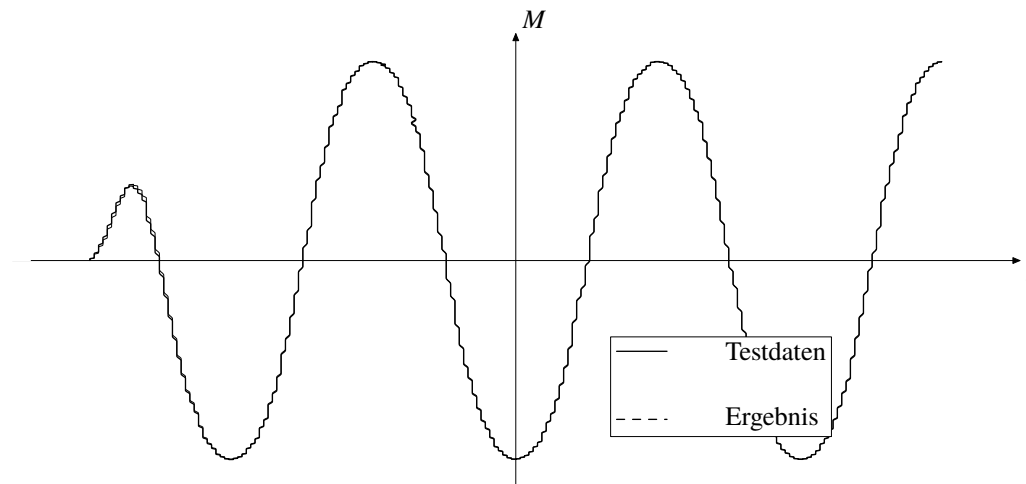


Abbildung 10: Verlauf von Strom und Spannung bei sinusförmiger Stromanregung. Die steilen Flanken der Hysteresekurve führen zu großen Spannungsspitzen durch $\nabla \times E = -\partial B / \partial t$.

8 Ausblick

Wie zu Beginn beschrieben, ist das Ziel, das Jiles-Atherton-Modell ortsaufgelöst anzuwenden und insbesondere ein ortsaufgelöstes inverses Modell zu entwickeln, mit dem man die Geometrieabhängigkeit und auch den Einfluss von Wirbelströmen auf die Hysteresemessungen eliminieren kann. Die Grundlage hierfür wurden mit den ausgeführten Betrachtungen gelegt. Der nächste Schritt ist der Entwurf des ortsaufgelösten Vorwärts-Modells mithilfe des Finite-Elemente-Pakets *Fenics*. Im nächsten Schritt gilt es, einen Jiles-Atherton-Funktionsblock für die Software *dolfin-adjoint* zu schreiben. *dolfin-adjoint* ist ein Softwarepaket, welches das Konzept des automatischen Differenzierens für Finite-Elemente-Berechnungen abstrahiert und somit *Reverse-Mode*-Berechnungen auf einem ganzen Finite-Elemente-Modell zulässt.

Neben diesen theoretischen Arbeiten müssen Messdaten gewonnen werden. Dazu ist geplant, einen automatisierten Messaufbau zu entwerfen, der Kernverluste und Hysteresekurven Frequenz-, Temperatur- und Amplituden-abhängig misst.

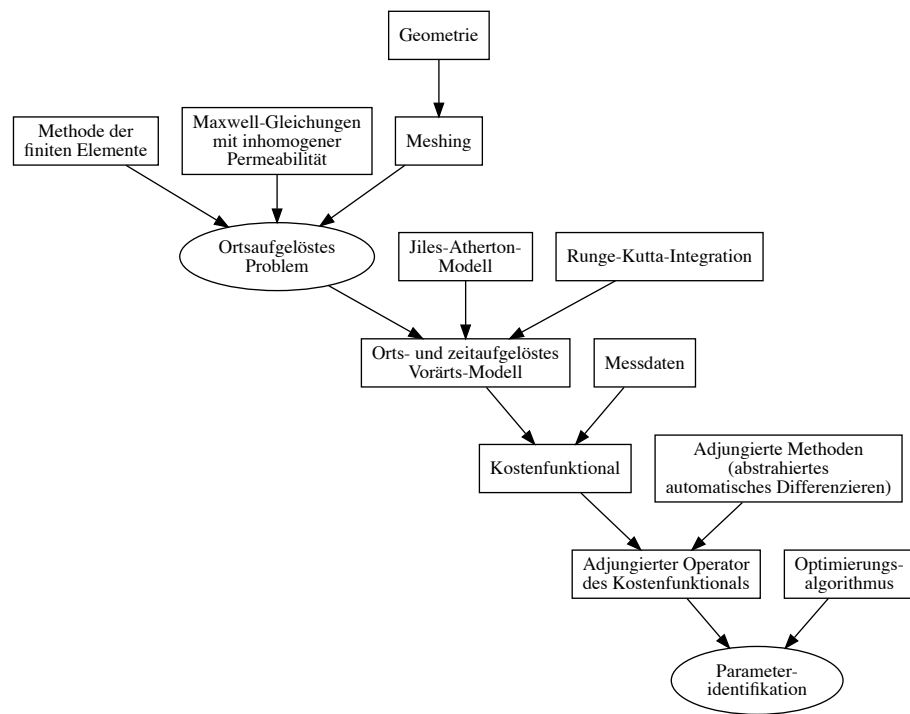


Abbildung 11: Vorgehen zur Parameteridentifikation im ortsaufgelösten Fall.

Literatur

- [1] R. Szewczyk, „Computational Problems Connected with Jiles-Atherton Model of Magnetic Hysteresis“, in *Recent Advances in Automation, Robotics and Measuring Techniques*, R. Szewczyk, C. Zieliński und M. Kaliczyńska, Hrsg., Cham: Springer International Publishing, 2014, S. 275–283, ISBN: 978-3-319-05353-0.
- [2] M. Abramowitz und I. A. Stegun, *Handbook of Mathematical Functions*, Ser. Applied Mathematics Series. National Bureau of Standards, Juni 1964, Bd. 55, ISBN: 9781614276173.
- [3] N. Sadowski, N. J. Batistela, J. P. A. Bastos und M. Lajoie-Mazenc, „An inverse Jiles-Atherton model to take into account hysteresis in time-stepping finite-element calculations“, *IEEE Transactions on Magnetics*, Jg. 38, Nr. 2, S. 797–800, März 2002, ISSN: 0018-9464. DOI: 10.1109/20.996206.
- [4] A. J. Bergqvist, „A simple vector generalization of the Jiles-Atherton model of hysteresis“, *IEEE Transactions on Magnetics*, Jg. 32, Nr. 5, S. 4213–4215, Sep. 1996, ISSN: 0018-9464. DOI: 10.1109/20.539337.