



universität
wien

MASTER THESIS

Titel der Master Thesis / Title of the Master's Thesis

„Numerische Lösung der Maxwell Gleichungen beim
Eindringen einer ebenen elektromagnetischen Welle in
mehrschichtiges, verlustbehaftetes Material“

verfasst von / submitted by

Dipl.-Ing. Dr.techn. Hans Georg Stöhr

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Arts (MA)

Wien, 2021 / Vienna 2021

Studienkennzahl lt. Studienblatt /
Postgraduate programme code as it appears on
the student record sheet:

Universitätslehrgang lt. Studienblatt /
Postgraduate programme as it appears on
the student record sheet:

Betreut von / Supervisor:

UA 992 499

Studium Generale – Das nachberufliche
Studium an der Universität Wien

ao. Univ.-Prof. Dr. Gerhard Krexner

Meiner Familie gewidmet

Zusammenfassung

Ziel der Arbeit ist die Lösung der Maxwell-Gleichungen unter speziellen Bedingungen (ebene elektromagnetische Welle konstanter Frequenz, 2-dimensional bzw. quasi-3-dimensional, nicht-magnetisches Medium).

Ebenso wird die Bedeutung und historische Entwicklung der Maxwell-Gleichungen im Allgemeinen dargestellt.

Der betrachtete Raumbereich (3-dimensional) wird in seinen Eigenschaften bezüglich des Verhaltens im Feld einer Elektromagnetischen Welle beschrieben durch die (magnetische) Permeabilität μ bzw. die (elektrische) Dielektrizitätskonstante ϵ .

Zur Berücksichtigung von Verlusten wird die komplexe Dielektrizitätskonstante (auch „Permittivität“ genannt) $\underline{\epsilon} = \epsilon_0 \cdot \epsilon_r + i \cdot \sigma / \omega$ verwendet.

Hierbei bezeichnet $\epsilon_0 = 8.85419 \cdot 10^{-12} \frac{As}{Vm}$ die elektrische Feldkonstante (eine Naturkonstante), und ϵ_r die (stoffabhängige) relative Permittivität.

σ ist die elektrische Leitfähigkeit [S / m] und ω ist die Kreisfrequenz [1 / s].

Es erfolgt eine Beschränkung auf ebene, polarisierte Elektromagnetische Wellen konstanter Frequenz und Amplitude, mit welcher der betrachtete Raumbereich durchdrungen wird, sowie die Betrachtung nicht-magnetischer Medien.

Die Feldverteilung lässt sich mit den Maxwell-Gleichungen prinzipiell berechnen. Allerdings sind diese Differentialgleichungen für nicht spezielle Geometrien kaum analytisch lösbar, sodass stattdessen die numerische Lösung eines Differenzengleichungssystems sinnvoll ist. Dazu wird der betrachtete Bereich im 3-Dimensionalen in kleine Abschnitte geteilt und jeder Abschnitt mit individuellen Materialeigenschaften beschrieben.

Das Differenzengleichungssystem wird durch eine Matrixgleichung lösbar.

Zur weiteren Vereinfachung werden die Lösungen im 2-Dimensionalen durchgeführt, d.h. der Raum wird „in Scheiben geschnitten“ und die Teillösungen danach wieder zusammengesetzt.

Ein weiterer Abschnitt widmet sich der numerischen Berechnung der Wärmeleitung im festen, dielektrischen Medium. Damit soll eine Abschätzung bzw. Quantifizierung der Erwärmung des Mediums ermöglicht werden.

Als „Test-Materialien“ werden biologische Strukturen (erhalten durch CT-Techniken) herangezogen.

In diesem Zusammenhang wird ebenfalls auf die mögliche biologische Wirkung von Elektromagnetischen Wellen auf den biologischen Organismus eingegangen.

Die Softwarerealisierung erfolgt mit Standardsprachen bzw. Tools wie z.B. gnuplot oder MATLAB.

Abstract

Goal of this scientific work is to solve Maxwell's equations under special conditions (plane electromagnetic wave of constant frequency, 2-dimensional or quasi-3-dimensional, non-magnetic medium).

The meaning and historical development of Maxwell's equations in general is also discussed.

Properties of the considered spatial volume (3-dimensional) are described with regard to its response to an electromagnetic wave by the (magnetic) permeability μ and the (electrical) permittivity ε .

The complex dielectric constant $\underline{\varepsilon} = \varepsilon_0 \cdot \varepsilon_r + i \cdot \sigma / \omega$ is used to take losses into account.

$\varepsilon_0 = 8.85419 \cdot 10^{-12} \frac{As}{Vm}$ denotes the vacuum permittivity (a universal constant), and ε_r the (material-dependent) relative permittivity.

σ is the electrical conductivity [S / m] and ω is the angular frequency [1 / s].

There is a restriction on a flat, polarized electromagnetic wave (of constant frequency and amplitude) by which the considered spatial volume is penetrated.

The field distribution can, in principle, be calculated using the Maxwell equations. However, an analytical solution of the differential equations with non-special geometries is hardly feasible, so that a numerical solution of a system of difference equations is indicated.

For this purpose, the volume under consideration is divided into small sections in three dimensions and each section is described with individual material properties. The system of difference equations can be solved by a matrix equation.

For further simplification, the calculations are carried out in two dimensions, i.e. the 3D space is "cut into slices" and the partial 2D solutions are then reassembled.

Another section is devoted to the numerical calculation of heat conduction in a solid, dielectric medium. The goal is to obtain an estimate or quantification of the heating of the medium.

Biological structures (obtained by CT techniques) are used as "test materials". In this context, reference is also made to the possible biological effects of electromagnetic waves on biological organisms.

The software is implemented using standard languages or tools such as gnuplot or MATLAB.

Inhaltsverzeichnis

Einleitung	1
James Clerk Maxwell.....	1
Historische Entwicklung der Maxwell-Gleichungen	6
Die vier Maxwell-Gleichungen in der heutigen Form	9
Berücksichtigung von Materialeigenschaften	11
Graphische Darstellung der Maxwell-Gleichungen	12
Wellengleichung – elektromagnetische Welle	13
Elektromagnetische Wellen im täglichen Leben	14
Elektromagnetische Wellen als Informationsquelle für die Kosmologie	15
Problematik der Lösung der Wellengleichung	17
Verwendete Hardware.....	17
Allgemeiner Lösungsansatz für harmonisch (d.h. sinus-förmig) variierende Felder in komplexer Schreibweise.....	17
Beispiel1: Eindimensionale Lösung - Eindringen einer ebenen EM-Welle in ein mehrschichtiges Dielektrikum mit Reflexion.....	19
Beispiel 2: Zweidimensionale Lösung - Eindringen einer ebenen EM-Welle in einen dielektrischen Zylinder	22
Numerische Lösung der Maxwellgleichungen.....	26
Numerische Lösung nach dem Cholesky-Verfahren	33
Numerische Lösung mit einer Band-Matrix in PYTHON.....	39
Numerische Lösung mit MATLAB	40
Erweiterung auf drei Dimensionen.....	43
Biologische MRT-Modelle als Testkörper.....	44
Zur Problematik „Rasterung – Frequenz/Wellenlänge – Rechenzeit“	52
Reflexion und Transmission	53
Thermisches Verhalten	54
Biologische Wirkungen	61
Fakten betreffend eventuelle Gefährdung durch Mobilfunk-Telefonie	63
Spezifische Absorptionsrate SAR.....	64
Abschätzung des elektrischen Feldstärkeverlaufs einer Sendestation.....	64
Schlussbemerkungen.....	65
Literaturverzeichnis	66
Anhang	70

Einleitung

Vor gut 42 Jahren wurde vom Verfasser dieser Arbeit im Rahmen einer wissenschaftlichen Untersuchung /1/ unter anderem in einem Kapitel die „numerische Lösung von Maxwell Gleichungen“ behandelt.

Die praktische Durchführung stieß zu der damaligen Zeit auf gewisse Schwierigkeiten; die computerisierten Rechenkapazitäten an der Technischen Universitäten Wien waren naturgemäß im Vergleich zu heutigen Möglichkeiten eher bescheiden und in der zeitlichen Nutzbarkeit ziemlich beschränkt.

Glücklicherweise wurde am damaligen Institut für Elektrische Messtechnik, TU-Wien (Univ.Professor Dr. Rupert Patzelt), ein Prozessrechner neu in Betrieb genommen. Die Type des Prozessrechners und die genauen technischen Daten sind heute nicht mehr eindeutig eruierbar, es handelte sich aber auf jeden Fall um ein Gerät der Serie DEC 20, ein 36-bit PDP-10 Mainframe Computer der Firma Digital Equipment Corporation /2/.

Diese Rechenanlage konnte während der Nachtstunden ausgiebig benutzt werden und stand in der Phase der Inbetriebnahme praktisch alleinig zur Verfügung. Die oftmaligen Rechendurchläufe wurden also am Abend gestartet, und – falls nicht kurzzeitig ein Fehler einen Abbruch auslöste – konnten am nächsten Morgen kontrolliert, ausgewertet, verbessert, erweitert, etc. werden. Ein typischer Rechendurchgang dauerte ca. 8 Stunden. Die Speicherplatzressourcen betrugen ca. 512 kilowords of solid state RAM, als Auslagerungsspeicher stand ein externer Plattenspeicher zur Verfügung. Als Programmiersprache wurde FORTRAN-77 eingesetzt.

Es entstand nun die Idee, den Teilbereich der Feldberechnungen der oben angeführten Arbeit in erweiterter Form nochmals aufzugreifen und mit den Rechenmöglichkeiten eines heutigen Desktop-Notebooks durchzuführen.

James Clerk Maxwell

Mit zwei physikalischen Grundkräften ist jedermann bewusst oder unbewusst ständig konfrontiert: Gravitation und Elektromagnetismus. Die zwei anderen Grundkräfte (nach heutigem anerkanntem physikalischem Wissensstand), die schwache Kernkraft sowie die starke Kernkraft sind für die Existenz des Universums natürlich genauso unerlässlich, doch „nur“ im subatomaren Bereich wirksam.

Die physikalischen Grundlagen des Elektromagnetismus wurden im Jahr 1865 von James Clerk Maxwell (geb. 13. Juni 1831 in Edinburgh; gest. 5. November 1879 in Cambridge) erstmals in ihrer Gesamtheit mathematisch beschrieben /3/.

Maxwell, ein schottischer Physiker, war einer der größten Wissenschaftler, die je gelebt haben.

Stellvertretend lassen einige Statements die Bedeutung von Maxwell erahnen:

Albert Einstein sagte: "Die spezielle Relativitätstheorie verdankt ihren Ursprung den Maxwell'schen Gleichungen des elektromagnetischen Feldes." (Autob. S. 23) /4/

Einstein sagte auch: "Seit Maxwell's Zeit wird die physikalische Realität als durch kontinuierliche Felder dargestellt angesehen und kann nicht mechanisch interpretiert werden. Diese Änderung in der Konzeption der Realität ist die tiefgreifendste und fruchtbarste, die die Physik seit der Zeit von Newton erlebt hat". /5/

Ivan Tolstoy schrieb in seiner Maxwell-Biographie: "Maxwells Bedeutung in der Geschichte des wissenschaftlichen Denkens ist vergleichbar mit der von Einstein (den er inspirierte) und der von Newton (dessen Einfluss er einschränkte)." /6/

Und schließlich: für Ludwig Boltzmann waren die Maxwell'schen Gleichungen von einer solchen Schönheit, dass sich der Physiker – in Anlehnung an ein Goethezitat aus dem Faust – fragte: War es ein Gott, der diese Zeilen schrieb? /7/

Aber auch in populärwissenschaftlichen Büchern werden die Maxwell-Gleichungen gewürdigt, z.B. wenn sie zu den 17 wichtigsten Gleichungen, die die Welt veränderten, gehören /8/.

Um die Maxwell-Gleichungen im zeitlichen Umfeld der Entwicklungen und Entdeckungen in der Physik richtig einordnen zu können, ist es sinnvoll den „Stand der Technik“ Mitte des 19.Jhds zu betrachten, auch um festzustellen, auf welche grundlegenden Erkenntnisse Maxwell aufbauen konnte /9/:

Bis zum Ende des 18. Jahrhunderts waren viele der grundlegenden experimentellen Merkmale der Elektrostatik und Magnetostatik etabliert. In den 1770er und 1780er Jahren führte Charles-Augustin Coulomb sehr empfindliche Experimente durch, bei denen die inversen Quadratgesetze der Elektrostatik und Magnetostatik direkt festgelegt wurden, die zu dieser Zeit als recht getrennte physikalische Phänomene erschienen.

Coulomb-Gesetz für Elektrostatik:
$$\vec{F} = \frac{Q \cdot Q'}{4\pi \cdot \epsilon_0 \cdot r^2} \cdot \vec{r}_0$$

$\vec{F}$ ist die Kraft, die zwischen zwei Ladungen Q bzw. Q' wirkt, r ist der Abstand und $\vec{r}_0$ ist der Einheitsvektor in der Verbindungslinie. $\epsilon_0 = 8.85419 \cdot 10^{-12} \frac{As}{Vm}$ ist die sogenannte Dielektrizitätskonstante (Permittivität).

1791 zeigte der italienische Anatom Luigi Galvani, dass elektrische Effekte die Kontraktion der Froschschenkel stimulieren können („Froschschenkel-Experiment“). Beim Versuch verwendete Galvani zwei unterschiedliche Metalle zur Herstellung einer Verbindung zwischen Nerv und Muskel in den Beinen der Frösche und löste damit eine der natürlichen Muskelkontraktion ähnliche Bewegung aus. Dies wurde als Entdeckung der tierischen Elektrizität angekündigt. Alessandro Volta vermutete, dass der elektrische Strom mit dem Vorhandensein verschiedener Metalle in Kontakt mit einem feuchten Körper

zusammenhängt. Im Jahr 1800 demonstrierte er dies durch den Bau einer Voltaik-Zelle, die aus Schichten von Kupfer und Zink bestand, die durch eine, mit leitender Flüssigkeit getränkte, Pappschicht getrennt waren. Der wichtigste Aspekt von Volta's Experimenten bestand darin, dass er eine Stromquelle entdeckt hatte, die Urform der heutigen Batterie.

Ein wichtiger experimenteller Fortschritt wurde 1820 erzielt, als Hans Christian Ørsted zeigte, dass mit einem elektrischen Strom immer ein Magnetfeld verbunden ist - dies war der Beginn der Wissenschaft des Elektromagnetismus.

Von Jean-Baptiste Biot und Félix Savart wurde erstmals die Stärke des Magnetfeldes im Abstand r von einem Stromelement der Länge dl , in dem ein Strom I fließt, beschrieben. Das sogenannte Biot-Savart-Gesetz lautet /10/:

$$dB = \frac{\mu_0 \cdot I \cdot dl}{4\pi r^2} \cdot \sin \alpha \quad \text{bzw. in vektorieller Form} \quad d\vec{B} = \frac{\mu_0 \cdot I \cdot d\vec{l} \times \vec{r}}{4\pi r^3}$$

Dabei bedeutet dl die Länge des Stromelements in Richtung des Stroms I , und $\vec{r}$ wird vom Stromelement $I \cdot d\vec{l}$ bis zum betrachteten Punkt im Vektorabstand $\vec{r}$ gemessen. α ist der Winkel, den die Richtung von $d\vec{l}$ mit der von $\vec{r}$ einschließt.

Als nächstes erweiterte André-Marie Ampère das Biot-Savart-Gesetz zum Ampère-Gesetz.

Das Gesetz setzt das Kurvenintegral des magnetischen Feldes entlang einer geschlossenen Kurve in Beziehung zum Strom, der durch die von dieser Kurve eingeschlossene Fläche fließt:

$$\oint \vec{B} \cdot d\vec{s} = \mu_0 \cdot I \quad \text{wobei der Weg } S \text{ beliebig, aber geschlossen ist /10/}.$$

Von Ampère stammt auch die Gleichung für die Kraft zwischen zwei Stromelementen $d\vec{l}_1$ und $d\vec{l}_2$, die die Ströme I_1 und I_2 führen:

$$F = \mu_0 \cdot \frac{I_1 \cdot I_2 \cdot l}{2\pi \cdot r} \quad \text{gilt, wenn die beiden Leiter parallel laufen; bzw. allgemein:}$$

$d\vec{F}_2 = \frac{\mu_0 \cdot I_1 \cdot I_2}{4\pi r^3} d\vec{l}_1 \times (d\vec{l}_2 \times \vec{r})$ $d\vec{F}_2$ ist die Kraft, die vom Stromelement $I_1 \cdot d\vec{l}_1$ auf das Stromelement $I_2 \cdot d\vec{l}_2$ wirkt, wobei $\vec{r}$ von $d\vec{l}_1$ aus gemessen wird.

1827 formulierte Georg Simon Ohm die Beziehung zwischen der Potentialdifferenz U und dem Strom I , dem heutigen Ohm'schen Gesetz $U = RI$, wobei R der elektrische Widerstand des Materials ist, durch das der Strom fließt.

Zu beachten ist, dass die obigen Beziehungen für statische Verhältnisse gelten. Zeitabhängigkeiten wurden bislang noch nicht berücksichtigt.

In den 1820er Jahren wurden viele Artikel in wissenschaftlichen Fachzeitschriften einge-reicht, die elektromagnetische Effekte beschreiben und in denen versucht wurde, diese zu erklären /9/. Michael Faraday begann mit systematischen Untersuchungen elektro-magnetischer Phänomene. Diese Experimente veranlassten Faraday, das Schlüsselkonzept der magnetischen Kraftlinien einzuführen, um die Richtung darzustellen, in der die auf einen

Magnetpol wirkende Kraft wirkt, wenn dieser in ein Magnetfeld gebracht wird. Je größer die Anzahl der Kraftlinien pro Flächeneinheit in der Ebene senkrecht zu den Feldlinien ist, desto größer ist die auf den Magnetpol wirkende Kraft.

Faraday glaubte fest an die Symmetrie der Natur und vermutete, dass es möglich sein müsse, einen elektrischen Strom aus einem Magnetfeld zu erzeugen, da umgekehrt ein elektrischer Strom ein Magnetfeld erzeugen konnte. 1831 erfuhr er von Joseph Henry's Experimenten, bei denen sehr starke Elektromagnete verwendet wurden. Faraday hatte sofort die Idee, die durch die Kraftlinien verursachte Belastung des Materials eines starken Elektromagneten zu beobachten. Er baute einen starken Elektromagneten, indem er einen Isolierdraht, durch den ein Strom geleitet werden konnte, auf einen dicken Eisenring wickelte und so ein Magnetfeld innerhalb des Rings erzeugte. Die Auswirkungen der Dehnung sollten durch eine andere Wicklung am Ring erfasst werden, die an einem Galvanometer angebracht war, um die Größe des erzeugten elektrischen Stroms zu messen.

Der Effekt war überhaupt nicht das, was Faraday erwartet hätte. Als der Primärkreis geschlossen wurde, zeigte sich ein Ausschlag der Galvanometernadel in der Sekundärwicklung - im Sekundärdraht war durch das Medium des Eisenrings ein elektrischer Strom induziert worden. Galvanometer-Ausschläge wurden nur beobachtet, wenn der Strom im Elektromagneten ein- und ausgeschaltet wurde. Dies war die Entdeckung der elektromagnetischen Induktion. Bereits 1831 hatte Faraday die qualitative Form seines Induktionsgesetzes in Bezug auf das Konzept der Kraftlinien festgelegt - die in einer Stromschleife induzierte elektromotorische Kraft steht in direktem Zusammenhang mit der Geschwindigkeit, mit der Magnetfeldlinien geschnitten werden.

1834 formulierte Lenz das Gesetz, das die Frage nach der Richtung der induzierten elektromotorischen Kraft im Stromkreis klärte - die elektromotorische Kraft wirkt in einer Richtung, die der Änderung des magnetischen Flusses entgegenwirkt.

Faraday konnte seine theoretischen Ideen nicht mathematisch formulieren, war jedoch überzeugt, dass das Konzept der Kraftlinien den Schlüssel zum Verständnis elektromagnetischer Phänomene darstellte. 1846 spekulierte er bereits, dass Licht eine Form von Störung sein könnte, die sich entlang der Feldlinien ausbreitet, aber die Idee wurde mit erheblicher Skepsis aufgenommen. Faraday glaubte instinktiv an die Einheit der Naturkräfte, insbesondere von Licht, Elektrizität und Magnetismus.

1846 leitete Faraday in einem starken Magnetfeld Licht durch Blei-Glas. Er demonstrierte ein Phänomen, das heute als Faraday-Rotation bekannt ist und bei dem die Polarisationssebene von linear polarisiertem Licht gedreht wird, wenn sich die Lichtstrahlen durch ein transparentes Dielektrikum entlang der Magnetfeldrichtung bewegen. Damit erbrachte er erstmals den Nachweis, dass Licht und Magnetismus zwei miteinander verbundene physikalische Phänomene sind.

William Thomson, später Lord Kelvin, interpretierte dieses Phänomen als Beweis dafür, dass Magnetismus im Wesentlichen rotatorischer Natur war, was Maxwell's Denken über die physikalische Natur von Magnetfeldern stark beeinflussen sollte.

Ein weiterer Puzzlestein, auf den Maxwell in weiterer Folge aufbauen konnte, wurde bereits durch Jean-Baptiste le Rond d'Alembert gelegt, der 1747 die (eindimensionale) Wellengleichung der schwingenden Saite (d'Alembert-Gleichung) aufstellte und löste /11/.

Mit dem d'Alembert-Operator $\square$ lässt sich z.B. die homogene Wellengleichung für eine Welle mit einer Zeit- und Orts-abhängigen Amplitude u einfach anschreiben:

$$\square = \frac{1}{c^2} \cdot \frac{\partial}{\partial t^2} - \frac{\partial}{\partial x^2} \quad \square u = 0$$

Die Konstante c hat die Dimension einer Geschwindigkeit, und es zeigt sich, dass es sich dabei um die Ausbreitungsgeschwindigkeit der Welle handelt.

Im damaligen Verständnis wurde angenommen, dass die Ausbreitung einer Welle ein Medium benötigt und dass die Ausbreitungsgeschwindigkeit von den Eigenschaften des Mediums abhängt.

Eine spezielle Problematik stellte in der Vor-Maxwell-Zeit das Wesen des Lichts dar. Galileo Galilei versuchte als einer der ersten, die Ausbreitungsgeschwindigkeit des Lichts ernsthaft zu messen (ca. 1630), jedoch ohne Erfolg. Dies gelang erst Ole Rømer anhand von Beobachtungsdaten der Jupitermonde 1675. Zwar betrug die Abweichung seines Messwerts (ca. $2,1 \cdot 10^8$ m/s) vom tatsächlichen Wert (ca. $3 \cdot 10^8$ m/s) rund 30 %. Die eigentliche Leistung Rømers bestand jedoch darin, nachzuweisen, dass sich das Licht mit endlicher Geschwindigkeit ausbreitet. Die Natur des Lichts blieb jedoch weiter ungeklärt. Im späten 17. Jahrhundert versuchte Isaac Newton mit seiner Korpuskeltheorie, die Ausbreitung des Lichts durch die Bewegung von kleinen Teilchen zu erklären. Damit konnte man zwar die Reflexion verstehen, nicht jedoch manche andere optische Phänomene, wie die Beugung, bei der es sich eindeutig um ein Wellenphänomen handelt. Zur gleichen Zeit begründeten Christiaan Huygens und andere die Wellentheorie des Lichts /12/. Die Suche nach dem Äther als Träger der Lichtwellen blieb naturgemäß erfolglos, bis die Äthertheorie erst durch die Spezielle Relativitätstheorie endgültig aufgegeben wurde.

Carl Friedrich Gauß (1777-1855) und Wilhelm Weber (1804-1891) waren die Begründer des Zentimeter-Gramm-Sekunde-Maßsystems (cgs), womit die Einheiten der Elektrizitäts- und Wärmelehre sowie der Optik auf die Grundeinheiten der Mechanik zurückgeführt werden sollten. Erst 1960 wurde beschlossen, in allen Bereichen des täglichen Lebens sowie der Technik und Physik das Internationale Einheitensystem (SI - *Système international d'unités*) zu verwenden /13/.

Historische Entwicklung der Maxwell-Gleichungen

Maxwell's Abhandlung „A Dynamic Theory of the Electromagnetic Field“ wurde 1864 verfasst /3/.

Wie Siegel /14/, /9/ hervorgehoben hat, hatte Maxwell von Beginn an bei seinen Forschungen zum Elektromagnetismus drei Ziele:

1. eine Theorie zu entwickeln, die alle bekannten Fakten von Elektrizität und Magnetismus umfasst;
2. zur Erreichung dieses Zieles Michael Faradays Ansatz für Elektrizität und Magnetismus zu verwenden, d.h. die Kraftlinien oder den Feldansatz;
3. und damit eine neue Theorie zu entwickeln, die darüber hinausgeht, was aus dem Experiment schon bekannt war. . .

Maxwell's Publikation ist ziemlich umfangreich (54 Seiten) und verwendet naturgemäß Abkürzungen und Formelzeichen, die heute nicht mehr gebräuchlich sind, sodass die Arbeit eher schwer verständlich ist. Einige Autoren /9/,/14/,/15/,/16/ haben glücklicherweise eine zeitgemäße Aufbereitung des Inhalts durchgeführt. Im Folgenden soll eine kurze Zusammenfassung gegeben werden. Wenn heute von den vier Maxwell-Gleichungen gesprochen wird, so sind es in der Originalarbeit tatsächlich zwanzig, teilweise wegen der Aufteilung in die 3 Koordinatenrichtungen. Tab.1 wurde /15/ entnommen.

Dabei bedeuten:

p', q', r'	Gesamtstrom (inkl. Verschiebungsstrom)
p, q, r	Leitungsstrom
f, g, h	dielektrische Verschiebung
α, β, γ	magnetische Feldstärke
F, G, H	Vektorpotential (original: electromagnetic momentum)
P, Q, R	elektrische Feldstärke
f, g, h	elektrische Flussdichte (original: electric displacement)
ψ	elektrisches Potential
k	$= \varepsilon^{-1}$
ρ	spezifischer Widerstand, $= \sigma^{-1}$
e	freie Ladung
v	Geschwindigkeit
ϕ	elektrisches Potential (neue Schreibweise)
A	Vektorpotential (neue Schreibweise)

Gl.	Maxwells Original-Gleichungen in [1]	Transkription	Vektor-Schreibweise
(A)	$p' = p + \frac{df}{dt}$ $q' = q + \frac{dg}{dt}$ $r' = r + \frac{dh}{dt}$	$(f,g,h) = \mathbf{D}$ $(p,q,r) = \mathbf{j}_0$ $(p',q',r') = \mathbf{j}$	$\mathbf{j} = \mathbf{j}_0 + \frac{\partial \mathbf{D}}{\partial t}$
(B)	$\mu\alpha = \frac{dH}{dy} - \frac{dG}{dz}$ $\mu\beta = \frac{dF}{dz} - \frac{dH}{dx}$ $\mu\gamma = \frac{dG}{dx} - \frac{dF}{dy}$	$(F,G,H) = \mathbf{A}$ $(\alpha,\beta,\gamma) = \mathbf{H}$	$\mu\mathbf{H} = \text{nabla} \times \mathbf{A} = \text{rot } \mathbf{A}$
(C)	$\frac{d\gamma}{dy} - \frac{d\beta}{dz} = ?4\pi? p'$ $\frac{d\alpha}{dz} - \frac{d\gamma}{dx} = ?4\pi? q'$ $\frac{d\beta}{dx} - \frac{d\alpha}{dy} = ?4\pi? r'$	$(F,G,H) = \mathbf{A}$ $(p',q',r') = \mathbf{j}$	$\text{nabla} \times \mathbf{H} = \text{rot } \mathbf{H} = \mathbf{j}$
(D)	$P = \mu(\gamma \frac{dy}{dt} - \beta \frac{dz}{dt}) - \frac{dF}{dt} - \frac{d\Psi}{dx}$ $Q = \mu(\alpha \frac{dz}{dt} - \gamma \frac{dx}{dt}) - \frac{dG}{dt} - \frac{d\Psi}{dy}$ $R = \mu(\beta \frac{dx}{dt} - \alpha \frac{dy}{dt}) - \frac{dH}{dt} - \frac{d\Psi}{dz}$	$(P,Q,R) = \mathbf{E}$ $(\alpha,\beta,\gamma) = \mathbf{H}$ $(F,G,H) = \mathbf{A}$ $\Psi = \varphi$	$\mathbf{E} = \mu \mathbf{v} \times \mathbf{H} - \frac{\partial \mathbf{A}}{\partial t} - \text{grad } \varphi$ $(\text{nabla } \varphi = \text{grad } \varphi)$
(E)	$P = kf$ $Q = kg$ $R = kh$	$k = \varepsilon^{-1}$ $(P,Q,R) = \mathbf{E}$ $(f,g,h) = \mathbf{D}$	$\mathbf{E} = \varepsilon^{-1} \mathbf{D}$
(F)	$P = -\rho p$ $Q = -\rho q$ $R = -\rho r$	$-\rho = \sigma^{-1}$ $(P,Q,R) = \mathbf{E}$ $(p,q,r) = \mathbf{j}_0$	$\mathbf{E} = \sigma^{-1} \mathbf{j}_0$
(G)	$?-? e + \frac{df}{dx} + \frac{dg}{dy} + \frac{dh}{dz} = 0$	$e = \rho$	$-\rho + \text{nabla} \cdot \mathbf{D} = -\rho + \text{div } \mathbf{D} = 0$
(H)	$\frac{de}{dt} + \frac{dp}{dx} + \frac{dq}{dy} + \frac{dr}{dz} = 0$	$(p,q,r) = \mathbf{j}_0$	$\frac{\partial \rho}{\partial t} + \text{div } \mathbf{j}_0 = 0$

Tabelle1: Die Gleichungen aus 1865 (entnommen aus /15/)

Diese Gleichungen enthalten zwei triviale Fehler: Für die Größe e in den Gleichungen (G) und (H) besteht eine Diskrepanz im Vorzeichen von e, ein offensichtlicher Druckfehler. Weiter ist in (C) der durch ?4π? markierte Faktor 4π wegzulassen, da offenbar auf Einheitenwechsel beruhend /15/.

Die heutigen 4 Maxwell-Gleichungen stellen offensichtlich eine Untermenge der Originalgleichungen dar.

Zuerst werden noch einige Umformungen gemacht:

aus (A, Tab.1) und (C, Tab.1) folgt $\text{rot } \vec{H} = \nabla \times \vec{H} = \vec{j} = \vec{j}_0 + \frac{\partial \vec{D}}{\partial t}$ [I]

mit (B, Tab.1) $\mu \vec{H} = (\nabla \times \vec{A})$ und beidseitiger Bildung der Divergenz ergibt sich

$$\mu \nabla \cdot \vec{H} = 0 \quad \text{bzw.} \quad \nabla \cdot \vec{B} = 0 \quad [\text{II}]$$

mit (D, Tab.1) $\vec{E} = \mu \cdot (\vec{v} \times \vec{H}) - \frac{\partial \vec{A}}{\partial t} - \nabla \varphi$ und beidseitiger Rotation ergibt sich

$$\nabla \times \vec{E} = \nabla \times (\vec{v} \times \mu \cdot \vec{H}) - \nabla \times \frac{\partial \vec{A}}{\partial t} - \nabla \times \nabla \varphi$$

$$\nabla \times \vec{E} = \nabla \times (\vec{v} \times (\nabla \times \vec{A})) - \nabla \times \frac{\partial \vec{A}}{\partial t} - \nabla \times \nabla \varphi$$

Für $\vec{v} = 0$ folgt (der Spezialfall $\vec{v} \neq 0$ wird erst von A. Einstein behandelt):

$$\nabla \times \vec{E} = -\nabla \times \frac{\partial \vec{A}}{\partial t} = -\frac{\partial}{\partial t} (\nabla \times \vec{A}) = -\frac{\partial}{\partial t} \mu \vec{H} = -\frac{\partial \vec{B}}{\partial t} \quad [\text{III}]$$

Damit wird das in der praktischen Anwendung eher unübliche Vektorpotential $\vec{A}$ eliminiert, und es ergibt sich bereits:

Coulomb-Gauß Gesetz [aus (G, Tab.1)] $\text{div } \vec{D} = \nabla \cdot \vec{D} = \rho$ 1.Maxwell-Gl.

Nichtexistenz magn. Ladungen [aus II] $\text{div } \vec{B} = \nabla \cdot \vec{B} = 0$ 2.Maxwell-Gl.

Induktionsgesetz [aus III] $\text{rot } \vec{E} = \nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$ 3.Maxwell-Gl.

Ampère-Maxwell Gesetz [aus I] $\text{rot } \vec{H} = \nabla \times \vec{H} = \vec{j} + \frac{\partial \vec{D}}{\partial t}$ 4.Maxwell-Gl.

Diese Gleichungen bilden die Maxwell-Gleichungen in differentieller Form.

In SI-Einheiten gilt für die verwendeten Größen:

E: elektrische Feldstärke	[V / m]
H: magnetische Feldstärke	[A / m]
D: elektrische Verschiebung	[As / m ²]
B: magnetische Induktion	[Vs / m ²]
j: elektrische Stromdichte	[A / m ²]
ρ : elektrische Ladungsdichte	[As / m ³]
ϵ : elektrische Dielektrizitätskonstante	[As / Vm]
μ : magnetische Permeabilität	[Vs / Am]

Was macht nun das Besondere der Maxwell-Gleichungen aus?

Es ist wohl die Erweiterung der Stromdichte, bei der zu einem Anteil des Leitungsstroms auch ein Anteil eines Verschiebungsstroms dazukommt. Die Ergänzung des Ampère'schen Gesetzes durch die Maxwell'sche Erweiterung eröffnet die Theorie der Elektromagnetischen Wellen.

(A)	$p' = p + \frac{df}{dt}$ $q' = q + \frac{dg}{dt}$ $r' = r + \frac{dh}{dt}$	$(f,g,h) = \vec{D}$ $(p,q,r) = \vec{j}_0$ $(p',q',r') = \vec{j}$	$\vec{j} = \vec{j}_0 + \frac{\partial \vec{D}}{\partial t}$
-----	--	--	---

Die Essenz von Maxwells Gleichungen besteht darin, dass sie sich auch mit zeitlich variierenden Phänomenen befassen.

Zu Maxwell's Zeit waren Elektromagnetische Wellen noch unbekannt. Allerdings vermutete Maxwell bereits, dass Licht eine derartige Welle sein könnte, speziell da auf Grund seiner Formeln sich zeigen ließ, dass sich eine EM-Welle mit Lichtgeschwindigkeit ausbreitet (siehe unten).

Es ist dabei als ein Charakteristikum der modernen theoretischen Physik anzusehen, dass etliche der am Modell gewonnenen Hypothesen erst Jahre später (im Fall „Elektromagnetischer Wellen“ erst nach Maxwell's Tod durch Heinrich Hertz im Jahr 1887) experimentell verifiziert werden konnten /7/.

Die vier Maxwell-Gleichungen in der heutigen Form

Mit Hilfe des Gauß'schen Integralsatzes (Divergenzsatz) /17/

$$\int_V (\nabla \cdot \vec{F}) dv = \oint_A \vec{F} \cdot d\vec{a}$$

V: Volumen, eingeschlossen von der geschlossenen Oberfläche A

bzw. des Stokes'schen Integralsatzes /17/

$$\int_A (\nabla \times \vec{F}) \cdot d\vec{a} = \oint_L \vec{F} \cdot d\vec{l}$$

A: Fläche, eingeschlossen von der geschlossenen Kurve L

lassen sich die 4 Maxwell-Gleichungen von der differentiellen Form in die Integralform überführen:

1. Maxwell-Gleichung: $\nabla \cdot \vec{D} = \rho$ beidseitige Integration über beliebiges Volumen

$$\int_V (\nabla \cdot \vec{D}) dv = \int_V \rho dv$$

Aus der rechten Seite $\int_V \rho dv$ ergibt sich die Gesamtladung Q (Ladungsdichte ρ über das betrachtete Volumen V integriert)

Die Anwendung des Gauß'schen Integralsatzes auf die linke Seite führt zu:

$$\Psi = \oint_A \vec{D} \cdot d\vec{a} = Q$$

Der elektrische Fluss Ψ (Verschiebungsfluss) durch eine geschlossene Fläche A hindurch ist also gleich der elektrischen Ladung Q, die von der Fläche eingeschlossen wird.

Einen Spezialfall stellt das Coulomb-Gesetz dar:

$$E = \frac{1}{4\pi \cdot \epsilon_0} \cdot \frac{Q}{r^2}$$

2. Maxwell-Gleichung: $\nabla \cdot \vec{B} = 0$ beidseitige Integration über beliebiges Volumen

$$\int_V (\nabla \cdot \vec{B}) dv = 0$$

Anwendung des Gauß'schen Integralsatzes:

$$\Psi_m = \oint_A \vec{B} \cdot d\vec{a} = 0$$

Der magnetische Fluss Ψ_m durch eine geschlossene Fläche A hindurch ist Null.

3. Maxwell-Gleichung: $\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$ beidseitige Integration über beliebige Fläche

$$\int_A (\nabla \times \vec{E}) \cdot d\vec{a} = -\frac{\partial}{\partial t} \int_A \vec{B} \cdot d\vec{a}$$

Anwendung des Stokes'schen Integralsatzes:

$$\oint_L \vec{E} \cdot d\vec{l} = -\frac{\partial}{\partial t} \int_A \vec{B} \cdot d\vec{a}$$

entspricht der elektr. Spannung
entlang der Kurve L (begrenzt A)

entspricht der zeitlichen Änderung des
magn. Fluss Ψ_m durch die Fläche A

Das Minus-Zeichen resultiert aus der Lenz-Regel (eine Änderung des magnetischen Flusses durch eine Leiterschleife induziert eine Spannung, so dass der dadurch fließende Strom ein Magnetfeld erzeugt, welches der Änderung des magnetischen Flusses entgegenwirkt).

4. Maxwell-Gleichung: $\nabla \times \vec{H} = \vec{j} + \frac{\partial \vec{D}}{\partial t}$ beidseitige Integration über beliebige Fläche

mit $\int_A \vec{j} \cdot d\vec{a} = I$

$$\int_A (\nabla \times \vec{H}) \cdot d\vec{a} = I + \frac{\partial}{\partial t} \int_A \vec{D} \cdot d\vec{a}$$

Anwendung des Stokes'schen Integralsatzes:

$$\oint_L \vec{H} \cdot d\vec{l} = I + \frac{\partial}{\partial t} \int_A \vec{D} \cdot d\vec{a}$$

Die durch diese Gleichung definierte magnetische Spannung U_m wird sowohl durch einen Leitungsstrom wie auch einen zeitlich sich ändernden elektrischen Fluss verursacht.

Die differentielle Form der Maxwell-Gleichungen ist günstig für die Berechnung des elektrischen und magnetischen Feldes an einem Ortspunkt.

Die integrale Form ist besser geeignet um die Felder in einem gesamten Raumbereich zu berechnen. Sie eignet sich gut für die Berechnung von symmetrischen Problemen, wie die Berechnung des Feldes einer geladenen Kugel, eines geladenen Zylinders oder einer geladenen Ebene.

Die differentielle Form eignet sich besser zur Berechnung komplizierter numerischer Probleme mit Hilfe von Computern oder beispielsweise zur Herleitung der Elektromagnetischen Wellen. Außerdem sieht die differentielle Form viel kompakter aus als die integrale Form.

Berücksichtigung von Materialeigenschaften

Dies gelingt durch die Verwendung von drei materialbeschreibenden Kenngrößen:

Elektrische Leitfähigkeit σ (sigma) $[(\Omega \cdot m)^{-1}] = [S \cdot m^{-1}] = [V \cdot A^{-1} \cdot m^{-1}]$

Permeabilität (magnetische Leitfähigkeit) μ $[V \cdot s \cdot A^{-1} \cdot m^{-1}]$

Permittivität (dielektrische Leitfähigkeit) ϵ $[A \cdot s \cdot V^{-1} \cdot m^{-1}]$

Sowohl die Permeabilität wie auch die Permittivität werden als Produkt der Kenngröße des Vakuums und derjenigen des betrachteten Materials dargestellt.

$\mu = \mu_0 \cdot \mu_r$ μ_r : relative Permeabilität $\mu_0 \approx 4 \cdot \pi \cdot 10^{-7} \text{ VsA}^{-1}\text{m}^{-1}$

$\epsilon = \epsilon_0 \cdot \epsilon_r$ ϵ_r : relative Permittivität $\epsilon_0 \approx 1/(36\pi \cdot 10^9) \text{ AsV}^{-1}\text{m}^{-1}$

Im Allgemeinen muss davon ausgegangen werden, dass σ , μ_r bzw. ϵ_r nicht konstant, nicht isotrop und unter anderem temperaturabhängig sind.

In Materie können die Größen μ , ϵ und σ Tensoren 2. Stufe darstellen, es handelt sich dann um nichtisotrope Medien, wo z.B. Doppelbrechung auftritt.

Wenn sich das Material in Ruhe befindet, sich linear verhält und isotrop ist, gelten folgende Materialgleichungen:

$$\vec{D} = \epsilon \cdot \vec{E} \quad \vec{B} = \mu \cdot \vec{H} \quad \vec{I} = \sigma \cdot \vec{E}$$

Für Vakuum können die 4 Maxwell-Gleichungen umgeschrieben werden:

$$\text{div } \vec{E} = \nabla \cdot \vec{E} = \frac{\rho}{\epsilon_0}$$

$$\text{div } \vec{B} = \nabla \cdot \vec{B} = 0$$

$$\text{rot } \vec{E} = \nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$$

$$\text{rot } \vec{B} = \nabla \times \vec{B} = \mu_0 \vec{j} + \mu_0 \varepsilon_0 \frac{\partial \vec{E}}{\partial t}$$

Graphische Darstellung der Maxwell-Gleichungen

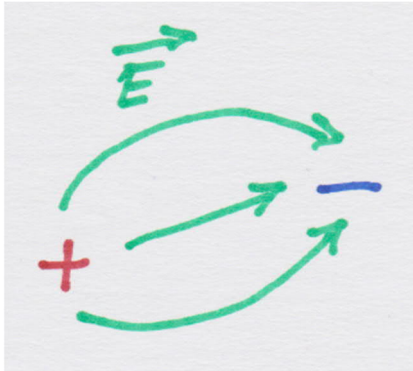


Abb. 1: zur 1. Maxwell-Gleichung

Positive und negative Ladungen sind Quellen und Senken des elektrischen Feldes $\vec{E}$.

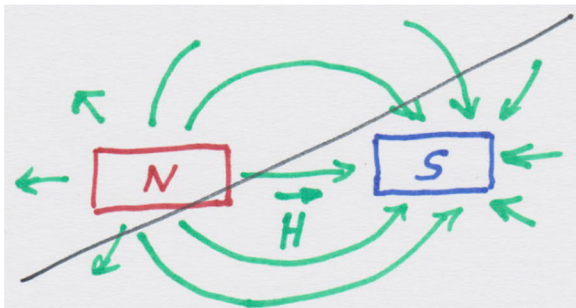


Abb. 2: zur 2. Maxwell-Gleichung

Es gibt keine magnetischen Monopole !

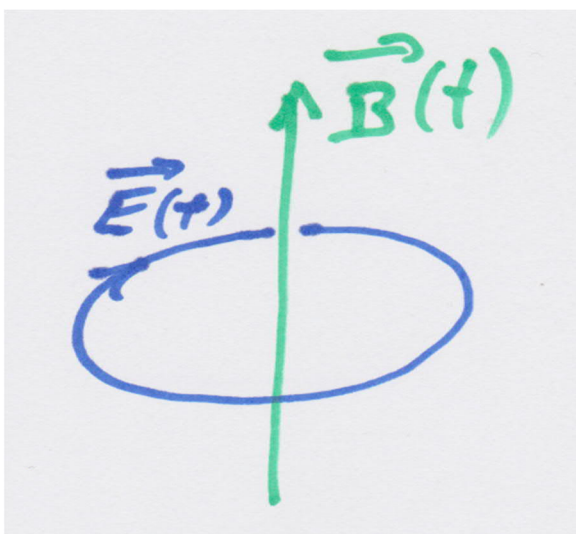


Abb. 3: zur 3. Maxwell-Gleichung

Ein sich zeitlich änderndes Magnetfeld $\vec{B}(t)$ verursacht ein elektrisches Wirbelfeld $\vec{E}(t)$.

Wird entlang der elektrischen Feldlinien eine Leiterschleife gelegt, so wird durch eine zeitliche Änderung des Magnetfelds eine Spannung induziert, bzw. es wird bei geschlossenem Stromkreis ein Induktionsstrom hervorgerufen.

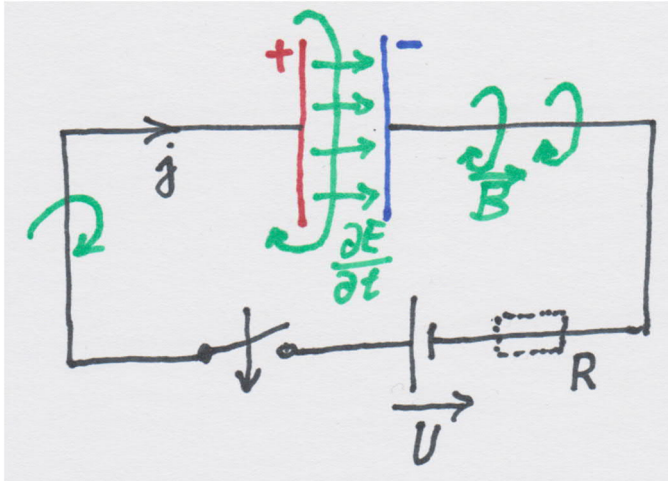


Abb. 4: zur 4. Maxwell-Gleichung

Ein Magnetfeld entsteht immer dort, wo Ströme fließen.

Unter der Annahme, dass der Kondensator anfänglich ungeladen ist, wird nach Schließen des Stromkreises ein Leitungsstrom j fließen, der ein Wirbel-Magnetfeld $\vec{B}$ erzeugt. Im Kondensator wird ein elektrisches Feld $\vec{E}$ aufgebaut. Der Feldaufbau ist zeitlich variabel ($\frac{\partial \vec{E}}{\partial t}$). Maxwell erkannte, dass nicht nur ein Leitungsstrom, sondern auch ein Verschiebungsstrom ein Magnetfeld verursacht.

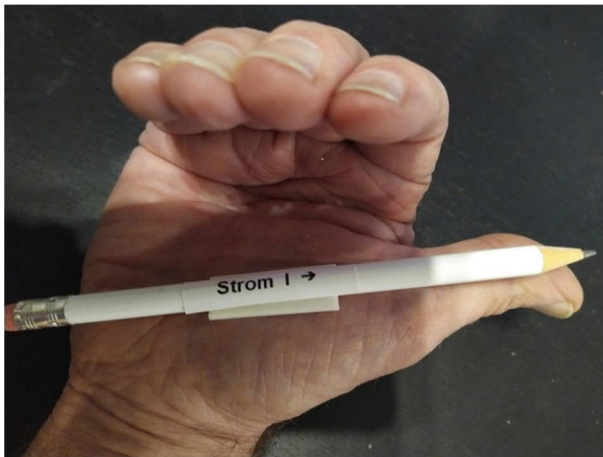


Abb. 5: Rechte-Hand-Regel

Umfasst man den Leiter mit der rechten Hand derart, dass der Daumen in die technische Stromrichtung zeigt, so zeigen die Finger die Richtung des zugehörigen magnetischen Feldes an.

Wellengleichung – elektromagnetische Welle

Im Materie-freien Raum gibt es keine Ladungen und auch keinen Leitungsstrom.

Die Gleichungen reduzieren sich zu:

$$\begin{aligned} \nabla \cdot \vec{E} &= 0 & \nabla \cdot \vec{B} &= 0 \\ \nabla \times \vec{E} &= -\frac{\partial \vec{B}}{\partial t} & \nabla \times \vec{B} &= \mu_0 \epsilon_0 \frac{\partial \vec{E}}{\partial t} \end{aligned}$$

Anwendung der Rotation auf beiden Seiten der 3. Maxwell-Gleichung:

$$\begin{aligned} \nabla \times (\nabla \times \vec{E}) &= -\nabla \times \frac{\partial \vec{B}}{\partial t} \\ \nabla \times (\nabla \times \vec{E}) &= -\frac{\partial}{\partial t} (\nabla \times \vec{B}) \end{aligned}$$

Mit der 4. Maxwell-Gleichung:

$$\nabla \times (\nabla \times \vec{E}) = -\frac{\partial}{\partial t} (\mu_0 \varepsilon_0 \frac{\partial \vec{E}}{\partial t}) = -\mu_0 \varepsilon_0 \frac{\partial^2 \vec{E}}{\partial t^2}$$

Aus der Vektoranalysis ergibt sich:

$$\nabla \times (\nabla \times \vec{E}) = \nabla \cdot (\nabla \cdot \vec{E}) - \Delta \vec{E}$$

Mit der 1. Maxwell-Gleichung ergibt sich somit:

$$\Delta \vec{E} = \mu_0 \varepsilon_0 \frac{\partial^2 \vec{E}}{\partial t^2}$$

Diese Beziehung entspricht der oben erwähnten Wellengleichung (vgl. d'Alembert-Gleichung).

Die elektrische Feldstärke $\vec{E}$ verhält sich also so wie eine Welle; die Ausbreitungsgeschwindigkeit c ergibt sich zu

$$c = \frac{1}{\sqrt{\mu_0 \varepsilon_0}} \approx \sqrt{\frac{36\pi \cdot 10^9}{4\pi \cdot 10^{-7}}} \frac{m}{s} = 3 \cdot 10^8 \frac{m}{s}$$

Es handelt sich um eine Transversalwelle, wobei die Ausbreitungsrichtung $\vec{k}$ sowie die Feldstärkenkomponenten $\vec{E}$ und $\vec{B}$ jeweils zueinander orthogonal stehen.

Diese sogenannte „Elektromagnetische Welle“ wurde also von Maxwell erstmals beschrieben.

In den Maxwell folgenden Jahren wurde die Bedeutung der elektromagnetischen Wellen immer mehr erkannt.

Elektromagnetische Wellen im täglichen Leben

Elektromagnetische Strahlung, die sich in Form elektromagnetischer Wellen ausbreitet, begegnet uns im Alltag in vielen verschiedenen Formen wie beispielsweise als sichtbares Licht, Wärmestrahlung im Infrarot-Bereich, ultraviolette Strahlungsanteile des Sonnenlichts, Ausstrahlung von Rundfunk- und Fernseh-Sendestationen, Mobilfunk („Handys“), Mikrowellen, Radar, Röntgen- und Gammastrahlung, Abstrahlung von Hochspannungsleitungen, etc.

Dabei sind nur Licht und Wärmestrahlung unseren Sinnesorganen unmittelbar zugänglich (vgl. Abb. 6).

Das Spektrum der EM-Wellen zieht sich also durch den gesamten Frequenzbereich.

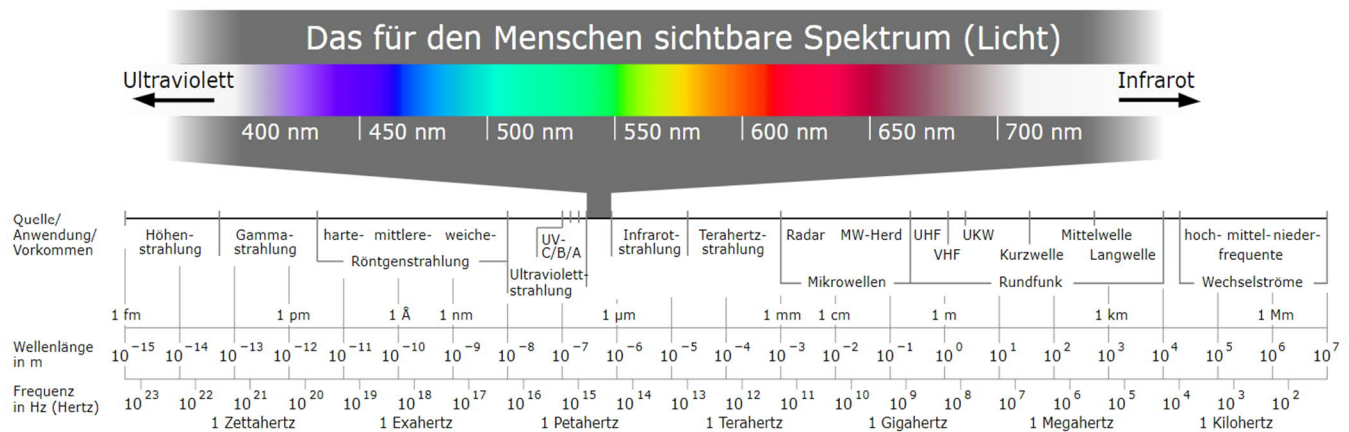


Abb. 6: Spektrum der EM-Wellen /19/

Die für uns maßgeblichste Quelle elektromagnetischer Strahlung ist die Sonne. Die Erde wird von ihr mit Energie versorgt, hauptsächlich in Form von Infrarotstrahlung (Wärme, ca. 47% der Sonnenenergie), sichtbarem Licht (ca. 46%) und Ultraviolettstrahlung (ca. 7%) /20/.

Täglich beträgt die auf der Erde ankommende Sonnenenergie etwa $4.3 \cdot 10^{15}$ kWh. Dies ist etwa 100mal mehr, als der jährliche Energieverbrauch der gesamten Erdbevölkerung

Die Energiedichte bzw. der Energiefluss ergibt sich durch folgende Zusammenhänge /21/:

Elektrische Energiedichte $W_e = \frac{1}{2} \cdot \epsilon \cdot E^2$ [W.s.m⁻³]

Magnetische Energiedichte $W_m = \frac{1}{2} \cdot \mu \cdot H^2$ [W.s.m⁻³]

Verlustleistungsdichte $P_V = j \cdot E = \sigma \cdot E^2$ [W.m⁻³]

Poynting Vektor (Fluss elektromagnetischer Energie) $\vec{S} = \vec{E} \times \vec{H}$ [W.m⁻²]

Elektromagnetische Wellen als Informationsquelle für die Kosmologie

Ein entscheidender Unterschied zwischen Laborphysik und Forschungen in der Astrophysik besteht darin, dass Astronomen die zu untersuchenden Objekte nicht gezielten Experimenten unterwerfen können. Sie müssen vielmehr passiv verfolgen, was ihnen bei der Beobachtung von Himmelsereignissen gerade angeboten wird.

Elektromagnetische Wellen von anderen Sonnen (Sternen), reflektierte bzw. durch Ionisation angeregte Strahlung von Planeten, planetarischen Nebeln und interstellarem Staub, etc., gelangt in verschiedensten Wellenbereichen des elektromagnetischen Spektrums zur Erde und ist die hauptsächliche Informationsquelle über die Geschichte und den Aufbau des gesamten Universums. Erst in den letzten Jahren werden Erkenntnisse, die mit Hilfe von EM-Wellen gewonnen werden können, ergänzt durch Messung und Auswertung von

Gravitationswellen. Eine Einführung in Phänomene der Astrophysik, auch in Zusammenhang mit EM-Wellen, ist z.B. in /22/ gut zusammengestellt.

Die Durchlässigkeit der Erdatmosphäre für elektromagnetische Strahlung ist sehr stark wellenlängenabhängig (Abb. 7). Die Tatsache, dass bis auf zwei „Fenster“ die Atmosphäre stark absorbierend wirkt, sowie die Beeinträchtigungen durch diverse atmosphärische Störungen generell („Seeing“), schränkt erdgebundene Beobachtungen stark ein. Satellitenobservatorien sind daher unverzichtbar.

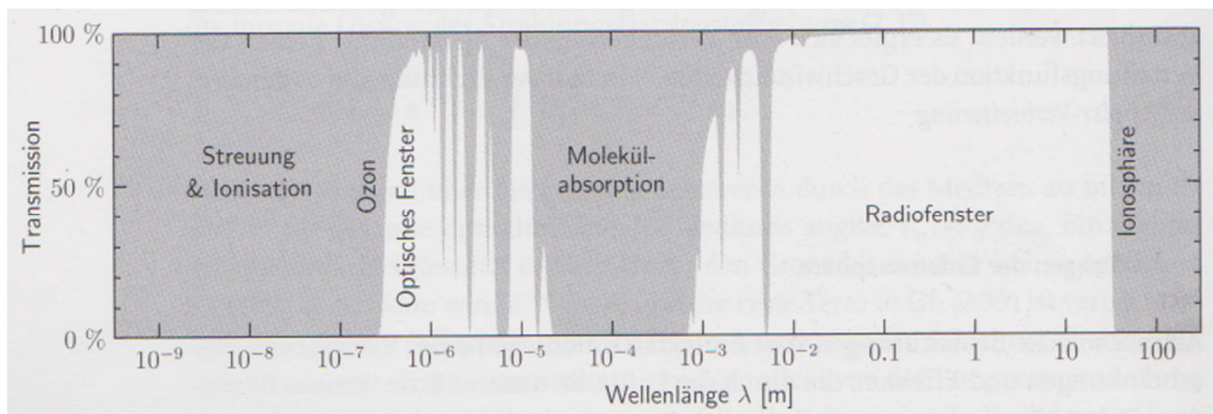


Abb. 7: Durchlässigkeit der Erdatmosphäre als Funktion der Wellenlänge, entnommen aus /22/

Welche Informationen können nun von Elektromagnetischen Wellen über das Universum gewonnen werden?

Abgesehen von optischen Aufnahmen in unterschiedlichen Frequenzbereichen (sichtbarer Bereich, Infrarotaufnahmen, Bildgebung im Röntgenbereich, Radioastronomie) gelangen in den letzten 100 Jahren revolutionäre Erkenntnisse in der Astronomie.

Einige Highlights dazu sind:

- Auswertung von Spektrallinien zur Elementbestimmung,
- Geschwindigkeitsmessung von Himmelskörpern durch Anwendung des Dopplereffekts,
- Entdeckung der Expansion des Universums durch Edwin Hubble (1924/25),
- Entfernungsbestimmung mit Hilfe der gemessenen Rotverschiebung und der Hubble-Konstante,
- Begründung der Urknalltheorie durch Georges Lemaître (1927),
- Bestimmung des Alters des Universums,
- Entdeckung der kosmischen (Mikrowellen-)Hintergrundstrahlung durch Arno Penzias und Robert Woodrow Wilson (1964).
- Hochauflösende Aufnahme des aktiven Kerns („Schwarzes Loch“) der Galaxie M87 mit Hilfe des „Event Horizon Telescope (EHT, Ereignishorizontteleskop)“, einem Verbund von Radioteleskopen (2019).

Problematik der Lösung der Wellengleichung

Die Wellen-Differentialgleichung stellt sich äußerst kompakt und plausibel dar. Allerdings, auf Grund der Vielfältigkeit in der Anwendung, der unterschiedlichen Geometrien und der unterschiedlichen Anfangs- und Randbedingungen ist es nicht möglich, eine allgemein gültige Lösungsmethodik anzugeben.

Nur für stark vereinfachte Annahmen (z.B. eindimensional, Zylindersymmetrie, ebene Welle, etc.) ist es möglich, analytisch zu lösen, also eine Lösungsfunktion (das Integral) explizit anzugeben.

Grundsätzlich können sehr viele mathematische Probleme nur durch approximative Methoden gelöst werden. Eine Approximation geschieht meist, indem Raum und Zeit durch ein Rechengitter in endlich viele Teile zerlegt werden (Diskretisierung). Die Ableitungen werden dann nicht mehr durch einen Grenzwert dargestellt, sondern durch Differenzen approximiert.

Die diskretisierte Differentialgleichung enthält keine Ableitungen mehr, sondern nur noch rein algebraische Ausdrücke. Damit ergibt sich entweder eine direkte Lösungsvorschrift oder ein lineares oder nichtlineares Gleichungssystem, welches dann mittels numerischer Verfahren gelöst werden kann /33/.

Verwendete Hardware

Die folgenden Berechnungen wurden auf einem relativ preisgünstigen (< EUR 800.-) Mittelklassen-Notebook durchgeführt.

ASUS VivoBook 15 90NBONL1-M10080

Intel® Core™ i5-8300H Prozessor (bis zu 4 GHz), Quad-Core

16 GB RAM, 512 GB SSD

NVIDIA GeForce GTX1650 Grafik (4096 MB)

Betriebssystem: Windows 10

Allgemeiner Lösungsansatz für harmonisch (d.h. sinusförmig) variierende Felder in komplexer Schreibweise

Beachte: im Weiteren wird j nicht als Symbol für die Stromdichte verwendet, sondern für die komplexe Einheit $j = \sqrt{-1}$.

Komplexe Größen werden durch einen „Unterstrich“ gekennzeichnet (z.B.: $\underline{\vec{E}}$).

Die Felder sollen eine Kreisfrequenz $\omega = 2\pi \cdot f$ aufweisen.

Allgemein kann dann das Feld $\vec{E}$ bzw. $\vec{H}$ in drei Dimensionen dargestellt werden durch:

$$\vec{E}(x, y, z, t) = \frac{1}{2} \left[\underline{\vec{E}}(x, y, z) \cdot \exp(j\omega t) + \underline{\vec{E}}^*(x, y, z) \cdot \exp(-j\omega t) \right]$$

$$\vec{H}(x, y, z, t) = \frac{1}{2} \left[\underline{\vec{H}}(x, y, z) \cdot \exp(j\omega t) + \underline{\vec{H}}^*(x, y, z) \cdot \exp(-j\omega t) \right]$$

$\underline{\vec{E}}(x, y, z)$ und $\underline{\vec{H}}(x, y, z)$ stellen komplexe Zeiger dar.

$$\underline{\vec{E}}(x, y, z) = \vec{E}(x, y, z) \exp(j\varphi_E)$$

$$\underline{\vec{H}}(x, y, z) = \vec{H}(x, y, z) \exp(j\varphi_H)$$

mit den reellen Amplituden $\vec{E}(x, y, z)$ und $\vec{H}(x, y, z)$

und den Phasenwinkeln φ_E und φ_H .

Damit lässt sich die 4. Maxwell-Gleichung umschreiben ($\frac{\partial}{\partial t} \rightarrow j\omega$):

$$\text{rot } \underline{\vec{H}} = j\omega \left(\underline{\varepsilon} - \frac{j \cdot \sigma}{\omega} \right)$$

Mit der Abkürzung: $\underline{\varepsilon} = \varepsilon - \frac{j\sigma}{\omega}$ (komplexe Dielektrizitätskonstante/Permittivität)

lautet die 3. Und 4. Maxwell-Gleichung:

$$\text{rot } \underline{\vec{H}} = j\omega \underline{\varepsilon} \underline{\vec{E}} \quad \text{rot } \underline{\vec{E}} = -j\omega \underline{\mu} \underline{\vec{H}}$$

Die Wellengleichung in komplexer Schreibweise lautet:

$$\Delta \underline{\vec{E}} + \underline{\varepsilon} \cdot \underline{\mu} \cdot \omega^2 \cdot \underline{\vec{E}} = 0$$

Die Komponentenschreibweise von $\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$ bzw. $\text{rot } \vec{E} = -\mu \frac{\partial \vec{H}}{\partial t}$ lautet:

$$\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} = -\mu \frac{\partial H_x}{\partial t}$$

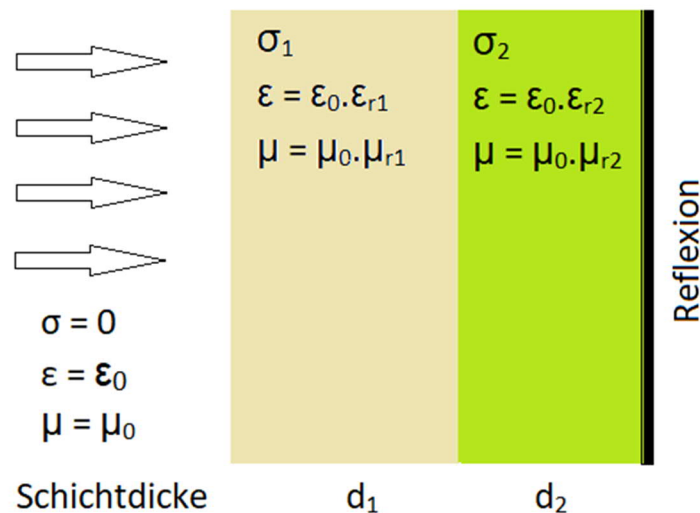
$$\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} = -\mu \frac{\partial H_y}{\partial t}$$

$$\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} = -\mu \frac{\partial H_z}{\partial t}$$

Als Beispiele für analytische Lösungen der Wellengleichungen werden im Folgenden zwei Berechnungen für einfache Geometrien behandelt.

Beispiel1: Eindimensionale Lösung - Eindringen einer ebenen EM-Welle in ein mehrschichtiges Dielektrikum mit Reflexion

Eine ebene Welle tritt in die Schichten ein und wird (abgeschwächt) an einer leitfähigen Wand reflektiert.



Der Nullpunkt der x-Koordinate wird am Eintrittspunkt in die Schicht 1 gewählt

Abb.8: Schichtenmodell

Wenn als Ausbreitungsrichtung x angenommen wird, dann sind für eine Ebene Welle die partiellen Ortsableitungen der Felder nach y und z gleich Null.

$$\frac{\partial E_z}{\partial x} = -\mu \frac{\partial H_y}{\partial t}$$

$$\frac{\partial E_y}{\partial x} = -\mu \frac{\partial H_z}{\partial t}$$

E_x und H_x sind Null und kommen in den obigen zwei Gleichungen auch nicht vor.

Für die elektrische Feldstärke können in den drei Schichten folgende Ansätze gemacht werden (Ausbreitungsrichtung: x , Polarisationsrichtung von E : z)

$$\underline{E_{z0}}(x) = \exp(-j \cdot \underline{\gamma_0} \cdot x) + \underline{A} \cdot \exp(j \cdot \underline{\gamma_0} \cdot x)$$

$$\underline{E_{z1}}(x) = \underline{B} \cdot \exp(-j \cdot \underline{\gamma_1} \cdot x) + \underline{C} \cdot \exp(j \cdot \underline{\gamma_1} \cdot x)$$

$$\underline{E_{z2}}(x) = \underline{D} \cdot \exp(-j \cdot \underline{\gamma_2} \cdot x) + \underline{F} \cdot \exp(j \cdot \underline{\gamma_2} \cdot x)$$

Für die drei Ausbreitungskonstanten gilt (hier wird angenommen, dass $\mu_r = 1$):

$$\underline{\gamma}_0 = \omega \cdot \sqrt{\varepsilon_0 \mu_0}$$

$$\underline{\gamma}_1 = \sqrt{\varepsilon_{r1} \varepsilon_0 \mu_0 \omega^2 - j \omega \mu_0 \sigma_1}$$

$$\underline{\gamma}_2 = \sqrt{\varepsilon_{r2} \varepsilon_0 \mu_0 \omega^2 - j \omega \mu_0 \sigma_2}$$

Die Konstanten $\underline{A}, \underline{B}, \underline{C}, \underline{D}, \underline{F}$ können durch Randbedingungen bzw. durch die Notwendigkeit, dass H und E stetig beim Schichtwechsel übergehen müssen, festgelegt werden.

Entsprechend $\text{rot } \vec{E} = -j\omega \mu \vec{H}$

lässt sich H_y ermitteln: $H_y = -\frac{1}{j\omega\mu_0} \cdot \frac{\partial E_z}{\partial x}$

Bei $x = 0$ muss gelten: $1 + \underline{A} = \underline{B} + \underline{C}$

$$\underline{\gamma}_0 \cdot (-1 + \underline{A}) = \underline{\gamma}_1 \cdot (-\underline{B} + \underline{C})$$

Bei $x = d_1$ muss gelten:

$$\underline{B} \cdot \exp(-j \cdot \underline{\gamma}_1 \cdot d_1) + \underline{C} \cdot \exp(j \cdot \underline{\gamma}_1 \cdot d_1) = \underline{D} \cdot \exp(-j \cdot \underline{\gamma}_2 \cdot d_1) + \underline{F} \cdot \exp(j \cdot \underline{\gamma}_2 \cdot d_1)$$

$$\underline{\gamma}_1 \cdot (-\underline{B} \cdot \exp(-j \cdot \underline{\gamma}_1 \cdot d_1) + \underline{C} \cdot \exp(j \cdot \underline{\gamma}_1 \cdot d_1))$$

$$= \underline{\gamma}_2 \cdot (-\underline{D} \cdot \exp(-j \cdot \underline{\gamma}_2 \cdot d_1) + \underline{F} \cdot \exp(j \cdot \underline{\gamma}_2 \cdot d_1))$$

Bei $x = d_1 + d_2$ muss gelten:

die elektrische Feldstärke $E_z(d_1 + d_2)$ muss an der Reflexionsebene verschwinden.

$$\underline{D} \cdot \exp(-j \cdot \underline{\gamma}_2 \cdot (d_1 + d_2)) + \underline{F} \cdot \exp(j \cdot \underline{\gamma}_2 \cdot (d_1 + d_2)) = 0$$

Die konkrete Berechnung bzw. grafische Darstellung lässt sich komfortabel in MATLAB /23/, /24/ durchführen (Lizenzierte Version R2021a, MATLAB and Simulink Student Suite)

Um die Lösung des linearen Gleichungssystems zur Ermittlung der Faktoren $\underline{A}, \underline{B}, \underline{C}, \underline{D}, \underline{F}$ übersichtlicher zu gestalten, werden einige Hilfsgröße eingeführt:

$$h1 = \exp(-j \cdot \underline{\gamma}_1 \cdot d1) \quad h2 = \exp(j \cdot \underline{\gamma}_1 \cdot d1) \quad h3 = \exp(-j \cdot \underline{\gamma}_2 \cdot d1)$$

$$h4 = \exp(j \cdot \underline{\gamma}_2 \cdot d1) \quad h5 = \exp(-j \cdot \underline{\gamma}_2 \cdot (d1 + d2)) \quad h6 = \exp(j \cdot \underline{\gamma}_2 \cdot (d1 + d2))$$

$$h7 = \underline{\gamma}_1 \cdot h1 \quad h8 = \underline{\gamma}_1 \cdot h2 \quad h9 = \underline{\gamma}_2 \cdot h3 \quad h10 = \underline{\gamma}_2 \cdot h4$$

Für die Behandlung des linearen Gleichungssystems ergibt sich so die Koeffizientenmatrix a und die Spaltenmatrix b der inhomogenen Terme:

$$a = \begin{pmatrix} -1 & 1 & 1 & 0 & 0 \\ \gamma_0 & \gamma_0 & -\gamma_1 & 0 & 0 \\ 0 & h1 & h2 & -h3 & -h4 \\ 0 & -\gamma_1 h1 & \gamma_1 h2 & \gamma_2 h3 & -\gamma_2 h4 \\ 0 & 0 & 0 & h5 & h6 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ \gamma_0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad x = \begin{pmatrix} \underline{A} \\ \underline{B} \\ \underline{C} \\ \underline{D} \\ \underline{F} \end{pmatrix} \quad a \cdot x = b$$

Zur Demonstration des Ergebnisse wird eine Berechnung (MATLAB-Programm „SchichtenReflexion.m“, siehe Anhang) mit folgenden Parametern durchgeführt:

Schichtdicken	$d1 = 0.2m$	$d2 = 0.3m$
Frequenz	$f = 2 \text{ GHz}$	
Dielektrikum Schicht1	$\varepsilon_{r1} = 5$	$\sigma_1 = 0.03 \text{ Sm}^{-1}$
Dielektrikum Schicht2	$\varepsilon_{r2} = 50$	$\sigma_2 = 0.1 \text{ Sm}^{-1}$

Wie erwartet wird die Feldstärke im Dielektrikum gedämpft, die Wellenlänge nimmt entsprechend $\sqrt{\varepsilon_r}$ ab und die Feldstärke wird bei $x = d1 + d2 = 0.5m$ Null.

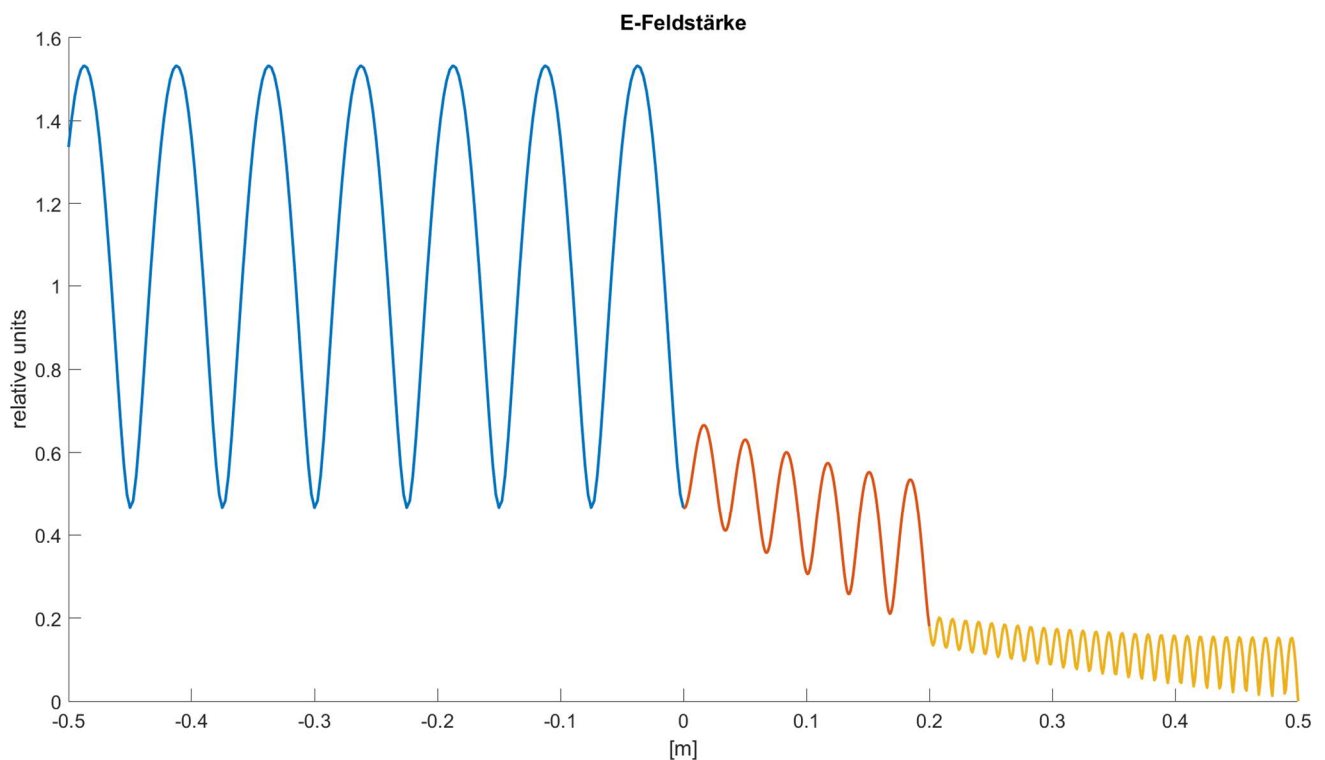


Abb. 9: Verlauf einer elektromagnetischen Welle beim Eindringen in dielektrische Schichten

Beispiel 2: Zweidimensionale Lösung - Eindringen einer ebenen EM-Welle in einen dielektrischen Zylinder

Entsprechend Abb. 10 liegt die Zylinderachse in der z-Achse des Koordinatensystems. Die Fortpflanzungsrichtung der Welle verläuft parallel zur x-Achse.

Es können zwei Spezialfälle betrachtet werden: die einfallende Welle ist z-polarisiert, oder sie ist y-polarisiert (Richtung des E-Feldes).

Bei zylindersymmetrischen Problemen ist es meist günstiger ein Zylinder-Koordinatensystem zu benutzen. Statt x, y, z wird φ, ρ, z verwendet.

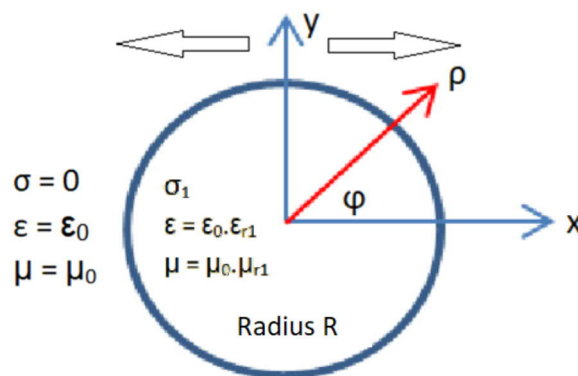


Abb. 10: Ebene EM-Welle trifft auf dielektrischen Zylinder

Hier wird angenommen, dass das elektrische Wechselfeld seinen Ausgang bei $x = 0$ mit der relativen Amplitude "1" hat. Die Welle trifft auf den Zylinder, ändert dadurch im Zylinder und im Umfeld des Zylinders ihre Intensität und läuft (bei der hier angenommenen Geometrie) bis ins Unendliche. Eine (eventuelle) Reflexion ist hier nicht vorgesehen.

Das Gesamtfeld der EM-Welle setzt sich zusammen aus einer einfallenden $\vec{E}^i$ Komponente und einer gestreuten Komponente $\vec{E}^s$.

Der allgemeine Ansatz für eine in x-Richtung einfallende Welle mit z-Polarisation (kartesische Koordinaten) ist

$$E_z^i = E_0 \cdot \exp(-j \cdot \gamma \cdot x) = E_0 \cdot \exp(-j \cdot \gamma \cdot \rho \cdot \cos \varphi)$$

Laut [25] ist der entsprechende Ansatz für Zylinderkoordinaten:

$$E_z^i = E_0 \cdot \sum_{n=-\infty}^{\infty} j^{-n} \cdot J_n(\gamma \rho) \cdot \exp(j \cdot n \cdot \varphi)$$

Und für das Streufeld:

$$E_z^s = E_0 \cdot \sum_{n=-\infty}^{\infty} j^{-n} \cdot a_n \cdot H_n^{(2)}(\gamma \rho) \cdot \exp(j \cdot n \cdot \varphi)$$

$J_n(x)$ bzw. $H_n^{(2)}(x)$ sind die „Bessel-Funktionen erster Gattung“ bzw. „Bessel-Funktionen dritter Gattung (auch bekannt als Hankel-Funktionen)“.

$H_n^{(1)}(x)$ bzw. $H_n^{(2)}(x)$ sind Linearkombinationen der Bessel-Funktionen erster und zweiter Gattung $Y_n(x)$ (auch Neumann-Funktionen genannt) [26], [27], [28].

$$H_n^{(1)}(x) = J_n(x) + j \cdot Y_n(x)$$

$$H_n^{(2)}(x) = J_n(x) - j \cdot Y_n(x)$$

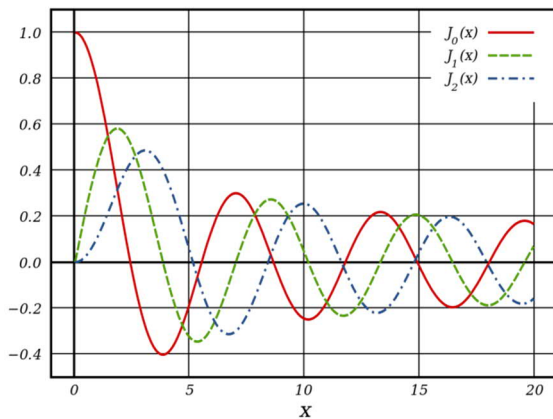


Abb. 11a: Bessel-Funktionen erster Gattung

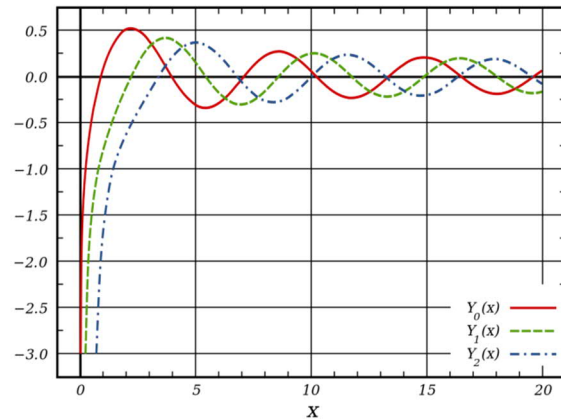


Abb. 11b: Bessel-Funktionen zweiter Gattung

Der Ansatz für das Gesamt-E-Feld ergibt sich nun zu

$$E_z = E_0 \cdot \sum_{n=-\infty}^{\infty} j^{-n} \cdot \left[J_n(\gamma_o \rho) + a_n \cdot H_n^{(2)}(\gamma_o \rho) \right] \cdot \exp(j \cdot n \cdot \varphi)$$

Im Zylinder existiert das Feld E_z^z

$$E_z^z = E_0 \cdot \sum_{n=-\infty}^{\infty} j^{-n} \cdot b_n \cdot J_n(\gamma_1 \rho) \cdot \exp(j \cdot n \cdot \varphi)$$

(im Zylinder darf keine Neumann-Funktion vorkommen, da für $\rho \rightarrow 0$ $Y(\gamma \rho) \rightarrow \infty$, was physikalisch auszuschließen ist.)

An der Zylinderberandung müssen die Felder stetig ineinander übergehen. Die Randbedingung ermöglicht die Berechnung der a_n und b_n .

Für $\rho = R$: $E_z = E_z^z$ bzw. $H_\varphi = H_\varphi^z$

Es gilt:
$$H_\varphi = \frac{1}{j\omega\mu} \cdot \gamma \cdot \frac{\partial E_z}{\partial \gamma \rho} \quad \text{mit} \quad \gamma = \sqrt{\varepsilon \cdot \mu \cdot \omega^2 - j \cdot \omega \cdot \mu \cdot \sigma}$$

Damit ergibt sich:

$$\left[J_n(\gamma_o R) + a_n \cdot H_n^{(2)}(\gamma_o R) \right] = b_n \cdot J_n(\gamma_z R)$$

$$b_n = \frac{J_n(\gamma_o R) + a_n \cdot H_n^{(2)}(\gamma_o R)}{J_n(\gamma_1 R)}$$

Eine längere Ableitung liefert

$$a_n = \frac{\gamma_1 \cdot J_n(\gamma_o R) \cdot J'_n(\gamma_1 R) - \gamma_o \cdot J'_n(\gamma_o R) \cdot J_n(\gamma_1 R)}{\gamma_o \cdot H_n^{(2)'}(\gamma_o R) \cdot J_n(\gamma_1 R) - \gamma_1 \cdot H_n^{(2)}(\gamma_o R) \cdot J'_n(\gamma_1 R)}$$

Dabei ist:
$$J'_n(\gamma_o R) = \frac{\partial J_n(\gamma_o \rho)}{\partial \gamma \rho} \text{ bei } \rho = R$$

$$H_n^{(2)'}(\gamma_o R) = \frac{\partial H_n^{(2)}(\gamma_o \rho)}{\partial \gamma \rho} \text{ bei } \rho = R$$

Für die Ableitung der Zylinderfunktionen (für alle 4 Gattungen, d.h. Besselfunktion, Neumannfunktion, Hankelfunktion 1. und 2. Art) gilt allgemein:

$$\frac{\partial Z_n(x)}{\partial x} = \frac{1}{2} \cdot (Z_{n-1}(x) - Z_{n+1}(x))$$

Für den Fall einer y-polarisierten Welle ist die Lösung dual zu der vorigen:

$$H_z = H_z \cdot \sum_{n=-\infty}^{\infty} j^{-n} \cdot \left[J_n(\gamma_o \rho) + c_n \cdot H_n^{(2)}(\gamma_o \rho) \right] \cdot \exp(j \cdot n \cdot \varphi)$$

$$H_z^z = E_z \cdot \sum_{n=-\infty}^{\infty} j^{-n} \cdot d_n \cdot J_n(\gamma_1 \rho) \cdot \exp(j \cdot n \cdot \varphi)$$

Mit:
$$-\frac{\partial H_z}{\partial \rho} = \varepsilon \cdot \frac{\partial E_\varphi}{\partial t}$$

ergibt sich:

$$c_n = \frac{\gamma_1 \cdot \varepsilon_o \cdot J_n(\gamma_o R) \cdot J'_n(\gamma_1 R) - \gamma_o \cdot \varepsilon_1 \cdot J'_n(\gamma_o R) \cdot J_n(\gamma_1 R)}{\gamma_o \cdot \varepsilon_1 \cdot J'_n(\gamma_1 R) \cdot H_n^{(2)'}(\gamma_o R) - \gamma_1 \cdot \varepsilon_o \cdot J'_n(\gamma_1 R) \cdot H_n^{(2)'}(\gamma_o R)}$$

$$d_n = \frac{J_n(\gamma_o R) + c_n \cdot H_n^{(2)}(\gamma_o R)}{J_n(\gamma_1 R)}$$

Dabei ist:

$$\varepsilon_1 = \varepsilon_0 \cdot \varepsilon_{r1} - \frac{j\sigma_1}{\omega}$$

Abb. 13 zeigt eine Berechnung des Verlaufs eines z-polarisierten E-Feldes beim Auftreffen auf einen dielektrischen Zylinder in z-Richtung. (MATLAB-Programm „Zylinder.m“, siehe Anhang) mit folgenden Parametern:

Betrachtetes Areal	1 Meter x 1 Meter (aufgelöst in 300 x 300 Rasterpunkten)		
Zylinderradius	$R = 0.2m$		
Frequenz	$f = 2 GHz$		
Dielektrikum des Zylinders	$\varepsilon_{r1} = 5$	$\sigma_1 = 0.03 Sm^{-1}$	

Die Berechnung der Feldstärke im Punkt (ρ, φ) benötigt eine Summenbildung von $n = -\infty$ bis $n = +\infty$. Die Summationsglieder nehmen mit steigenden n ab, sodass sie für höheres n vernachlässigt werden können. Empirisch zeigt sich, dass bei ca. $n = 20$ die Summation unbedenklich abgebrochen werden kann. In Abb. 12 ist zu Testzwecken $\gamma_1 = \gamma_0$ gesetzt (d.h. der Zylinder ist nicht existent) und es zeigt sich der ungestörte sinusförmige Verlauf der Feldstärke.

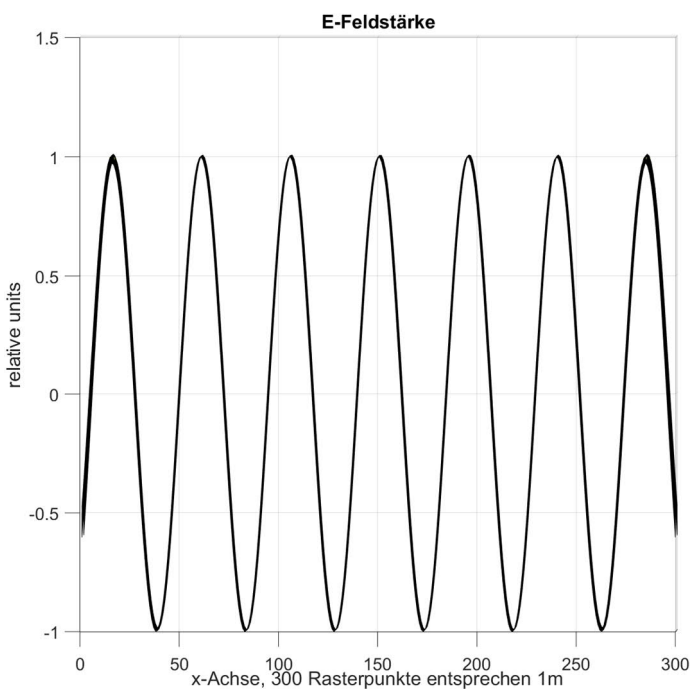


Abb. 12: Testplot für $\gamma_0 = \gamma_1$

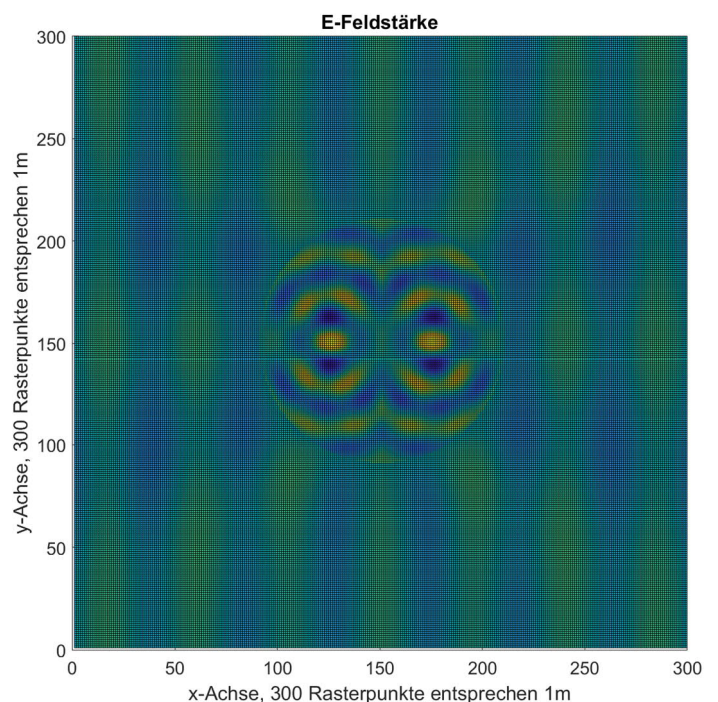


Abb. 13: Ebene Welle auf dielektr. Zylinder

Numerische Lösung der Maxwellgleichungen

Wie bereits erwähnt, ist die Berechnung von Feldverteilungen mit analytischen Methoden bestenfalls mit stark vereinfachenden Annahmen möglich. Für kompliziertere Geometrien muss daher auf numerische Verfahren übergegangen werden.

Im Folgenden soll an einigen Beispielen dargestellt werden, wie derartige Lösungsansätze prinzipiell aufgebaut sind.

Einfache numerische Algorithmen beruhen auf der Finite-Differenzen-Methode. Die grundlegende Idee des Verfahrens ist es, die Ortsableitungen in der Differenzialgleichung an endlich vielen (= „finiten“), äquidistanten Gitterpunkten durch Differenzenquotienten zu approximieren. Die approximierten Lösungen der Differenzialgleichung an den Gitterpunkten lassen sich dann durch das entsprechende Gleichungssystem berechnen /29/, /30/, /31/.

Die folgenden Beispiele orientieren sich an der aus Abb. 14 ersichtlichen Geometrie. Ein gewisser 2D-Bereich wird in Rasterpunkten zerlegt. Die Rasterkoordinaten laufen von $i = 1$ bis $i = I$, bzw. von $j = 1$ bis $j = J$. Ein „Testkörper“, der den Feldverlauf „stört“ soll relativ klein gegenüber den Gesamtabmessungen sein. Damit kann an den Seiten für $j = 1$ bzw. $j = J$ ein ungestörter Feldverlauf (näherungsweise) angenommen werden. Ohne Beschränkung der Allgemeinheit wird auch angenommen, dass am Rand bei $i = I$ eine (Total-)Reflexion stattfindet (z.B. durch eine gut leitfähige Wand), sodass dort die Feldstärke erzwungenermaßen auf Null abnimmt. Mit diesen Annahmen lassen sich in weiterer Folge Randbedingungen für die Berechnung definieren. Die einfallende Welle bewegt sich entlang der Rasterung mit steigendem i .

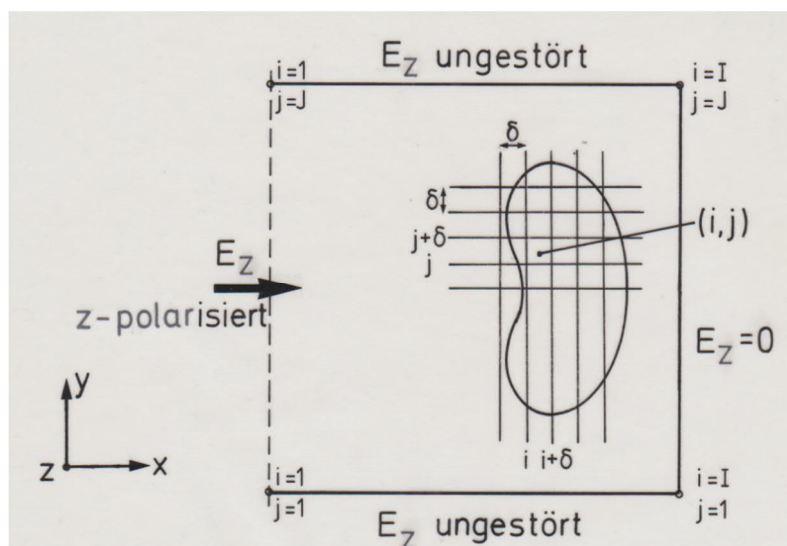


Abb. 14: Rasterung der 2-dimensionalen Ebene /1/

In kartesischen Koordinaten gilt für die Rotation:

$$\text{rot } \vec{E} = \vec{\nabla} \times \vec{E} = \begin{pmatrix} \frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} \\ \frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} \\ \frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} \end{pmatrix} = \det \begin{vmatrix} \vec{e}_x & \vec{e}_y & \vec{e}_z \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ E_x & E_y & E_z \end{vmatrix}$$

Für den von z unabhängigen Fall mit

$$\frac{\partial}{\partial z} = 0$$

reduziert sich dies zu:

$$\begin{aligned} \frac{\partial E_z}{\partial y} &= -j \cdot \omega \cdot \mu \cdot H_x & \frac{\partial H_z}{\partial y} &= j \cdot \omega \cdot \underline{\varepsilon} \cdot E_x \\ -\frac{\partial E_z}{\partial x} &= -j \cdot \omega \cdot \mu \cdot H_y & -\frac{\partial H_z}{\partial x} &= j \cdot \omega \cdot \underline{\varepsilon} \cdot E_y \\ \frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} &= -j \cdot \omega \cdot \mu \cdot H_z & \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} &= j \cdot \omega \cdot \underline{\varepsilon} \cdot E_z \end{aligned}$$

$$\underline{\varepsilon} = \varepsilon - j \cdot \frac{\sigma}{\omega} \quad \text{ist ebenfalls ortsabhängig}$$

$$\underline{\varepsilon} = \varepsilon(i, j) - j \cdot \frac{\sigma(i, j)}{\omega}$$

Die einfallende Welle ist nur z-polarisiert → damit erfolgt eine Reduktion zu:

$$\begin{aligned} \frac{\partial E_z}{\partial y} &= -j \cdot \omega \cdot \mu \cdot H_x \\ \frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} &= j \cdot \omega \cdot \underline{\varepsilon} \cdot E_z \\ -\frac{\partial E_z}{\partial x} &= -j \cdot \omega \cdot \mu \cdot H_y \end{aligned}$$

Aus Differentialgleichungen werden Differenzengleichungen:

x, y, z → i, j, k

$$\frac{\partial F(i, j)}{\partial x} \longrightarrow \frac{F(i+\delta, j) - F(i, j)}{\delta} \quad \text{bzw.:}$$

$$\frac{\partial F(i, j)}{\partial x} \longrightarrow \frac{F(i+1, j) - F(i, j)}{\delta}$$

$$\frac{\partial F(i,j)}{\partial y} \longrightarrow \frac{F(i,j+1) - F(i,j)}{\delta}$$

mit $\delta = d(i, i+1) = d(j, j+1)$

Pro Rasterpunkt (i,j) gibt es 3 Differenzengleichungen mit 3 Unbekannten: E_z , H_x , H_y

$$\frac{\underline{E}_z(i,j+1) - \underline{E}_z(i,j)}{\delta} = -j \cdot \omega \cdot \mu \cdot \underline{H}_x(i,j)$$

$$\frac{\underline{E}_z(i+1,j) - \underline{E}_z(i,j)}{\delta} = j \cdot \omega \cdot \mu \cdot \underline{H}_y(i,j)$$

$$\frac{\underline{H}_y(i+1,j) - \underline{H}_y(i,j) - (\underline{H}_x(i,j+1) - \underline{H}_x(i,j))}{\delta} = j \cdot \omega \cdot \underline{\epsilon} \cdot \underline{E}_z(i,j)$$

Aus:

$$\underline{H}_x = -\frac{1}{j \cdot \omega \cdot \mu} \frac{\partial \underline{E}_z}{\partial y} \quad \underline{H}_y = \frac{1}{j \cdot \omega \cdot \mu} \frac{\partial \underline{E}_z}{\partial x}$$

$$\frac{\partial \underline{H}_y}{\partial x} - \frac{\partial \underline{H}_x}{\partial y} = \frac{1}{j \cdot \omega \cdot \mu} \left(\frac{\partial^2 \underline{E}_z}{\partial x^2} + \frac{\partial^2 \underline{E}_z}{\partial y^2} \right) = j \cdot \omega \cdot \underline{\epsilon} \cdot \underline{E}_z$$

Mit:

$$\frac{\partial^2 F(i,j)}{\partial x^2} \longrightarrow \frac{F(i+1,j) - 2 \cdot F(i,j) + F(i-1,j)}{\delta^2}$$

$$\frac{\partial^2 F(i,j)}{\partial y^2} \longrightarrow \frac{F(i,j+1) - 2 \cdot F(i,j) + F(i,j-1)}{\delta^2}$$

$$\underline{E}_z(i+1,j) - 2 \cdot \underline{E}_z(i,j) + \underline{E}_z(i-1,j) + \underline{E}_z(i,j+1) - 2 \cdot \underline{E}_z(i,j) + \underline{E}_z(i,j-1) = \delta^2 \cdot (j \cdot \omega \cdot \mu) \cdot (j \cdot \omega \cdot \underline{\epsilon}) \cdot \underline{E}_z(i,j)$$

$$\underline{E}_z(i+1,j) + \underline{E}_z(i-1,j) + \underline{E}_z(i,j+1) + \underline{E}_z(i,j-1) - [4 - \delta^2 \cdot \omega^2 \cdot \mu \cdot \underline{\epsilon}] \cdot \underline{E}_z(i,j) = 0$$

Definition von Randbedingungen

Die einfallende Welle ist z-Polarisiert.

Verlauf der Welle im ungestörten Gebiet (Vakuum), zum Zeitpunkt $t=0$:

$$E_{z0}(x) = 1 \cdot \cos\left(2\pi \cdot \frac{x}{\lambda}\right)$$

$$H_{x0}(x) = 0$$

$$H_{y0}(x) = \frac{1}{\eta} \cdot \cos\left(2\pi \cdot \frac{x}{\lambda} + \frac{\pi}{2}\right)$$

Im Koordinatensystem mit $i = 1, 2, \dots, i_{\max}$ in x-Richtung bzw. $j = 1, 2, \dots, j_{\max}$ in y-Richtung gilt:

Anfangswellenfront:

$$\begin{aligned} E_z(1, j) &= E_{z0} \\ H_x(1, j) &= 0 \\ H_y(1, j) &= H_{y0} \end{aligned}$$

Mit (metallischer) Reflexionswand: an der metallischen Fläche verschwindet die elektrische Feldstärke ($i = i_{\max}$):

$$\begin{aligned} E_z(i_{\max}, j) &= 0 \\ E_x(i_{\max}, j) &= 0 \\ H_y(i_{\max}, j) &= \frac{1}{\eta} \cdot E_{z0} \\ \eta &= \sqrt{\frac{\mu}{\epsilon}} \end{aligned}$$

Die seitliche Randbedingung ($j=1$ sowie $j=j_{\max}$) ergeben sich zu:

$$E_z(i, 1) = E_z(i, j_{\max}) = 1 \cdot \cos\left(2\pi \cdot \frac{(i-1) \cdot \delta}{\lambda}\right)$$

$$E_x(i, 1) = E_x(i, j_{\max}) = 0$$

$$H_y(i, 1) = H_y(i, j_{\max}) = \frac{1}{\eta} \cdot \cos\left(2\pi \cdot \frac{(i-1) \cdot \delta}{\lambda} + \frac{\pi}{2}\right)$$

Anwendung der Differenzengleichung:

(die Indizes laufen von 1 weg, und nicht wie sonst üblich von 0, damit dies mit der Matrix-Indizierung von MATLAB zusammenpasst)

Für $i = 1$ $E_z(1, 1) = 1$

$$E_z(1, 2) = 1 \cdot \cos\left(2\pi \cdot \frac{1 \cdot \delta}{\lambda}\right)$$

$$E_z(1, 3) = 1 \cdot \cos\left(2\pi \cdot \frac{2 \cdot \delta}{\lambda}\right)$$

usw.

Für $i = 2 \quad j = 2, 2, \dots, j_{\max} - 1$

$$\underline{E}_z(3, j) + \underline{E}_z(1, j) + \underline{E}_z(2, j + 1) + \underline{E}_z(2, j - 1) - [4 - \delta^2 \cdot \omega^2 \cdot \mu \cdot \underline{\varepsilon}] \cdot \underline{E}_z(2, j) = 0$$

Für $i = 3 \quad j = 2, 2, \dots, j_{\max} - 1$

$$\underline{E}_z(4, j) + \underline{E}_z(2, j) + \underline{E}_z(3, j + 1) + \underline{E}_z(3, j - 1) - [4 - \delta^2 \cdot \omega^2 \cdot \mu \cdot \underline{\varepsilon}] \cdot \underline{E}_z(3, j) = 0$$

USW.

Für $i = i_{\max} \quad E_z(i_{\max}, 1) = 1$

$$E_z(i_{\max}, 2) = 1 \cdot \cos\left(2\pi \cdot \frac{1 \cdot \delta}{\lambda}\right)$$

$$E_z(i_{\max}, 3) = 1 \cdot \cos\left(2\pi \cdot \frac{2 \cdot \delta}{\lambda}\right)$$

USW.

Als Matrix, mit $[\%] = [-4 + \delta^2 \cdot \omega^2 \cdot \mu \cdot \underline{\varepsilon}]$

$i = 1 - i_{\max}, \quad j = 1 - j_{\max}$ wobei $I = i_{\max}$ bzw. $J = j_{\max}$ hier als Bsp.: $i_{\max} = j_{\max} = 6$

$$\begin{array}{lcl} i = 2: & & \\ E[3,2] + E[2,3] + [\%] \cdot E[2,2] & & = -E[1,2] - \\ E[2,1] & & \\ E[3,3] + E[2,4] + E[2,2] + [\%] \cdot E[2,3] & & = -E[1,3] \\ E[3,4] + E[2,5] + E[2,3] + [\%] \cdot E[2,4] & & = -E[1,4] \\ E[3,5] + E[2,4] + [\%] \cdot E[2,5] & & = -E[1,5] - E[2,6] \\ i = 3: & & \\ E[4,2] + E[2,2] + E[3,3] + [\%] \cdot E[3,2] & & = - \\ E[3,1] & & \\ E[4,3] + E[2,3] + E[3,4] + E[3,2] + [\%] \cdot E[3,3] & & = 0 \\ E[4,4] + E[2,4] + E[3,5] + E[3,3] + [\%] \cdot E[3,4] & & = 0 \\ E[4,5] + E[2,5] + E[3,4] + [\%] \cdot E[3,5] & & = -E[3,6] \\ i = 4: & & \\ E[5,2] + E[3,2] + E[4,3] + [\%] \cdot E[4,2] & & = -E[4,1] \\ E[5,3] + E[3,3] + E[4,4] + E[4,2] + [\%] \cdot E[4,3] & & = 0 \\ E[5,4] + E[3,4] + E[4,5] + E[4,3] + [\%] \cdot E[4,4] & & = 0 \\ E[5,5] + E[3,5] + E[4,4] + [\%] \cdot E[4,5] & & = -E[4,6] \\ i = 5: & & \\ E[3,2] + E[5,3] + [\%] \cdot E[5,2] & & = -E[6,2] - E[5,1] \\ E[3,3] + E[5,4] + E[5,2] + [\%] \cdot E[5,3] & & = -E[6,3] \\ E[3,4] + E[5,5] + E[5,3] + [\%] \cdot E[5,4] & & = -E[6,4] \\ E[3,4] & & + E(5,4) + [\%] \cdot E[5,5] = -E[6,5] - E[5,6] \end{array}$$

Beschränkt auf 6 x 6 Feldpunkte (d.h. 4 x 4 Unbekannte) ergibt sich eine Bandmatrix folgender Gestalt:

$$\text{mit } A_{i,j} = [-4 + \delta^2 \cdot \omega^2 \cdot \mu \cdot \underline{\varepsilon}(i,j)]$$

	(i _{max} -2)Spalten j = 2				(i _{max} -2)Spalten j = 3				(i _{max} -2)Spalten j = 4				(i _{max} -2)Spalten j = 5					
(jmax-2)	A _{i,j}	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	=>	-E(1,2)
Zeilen	1	A _{i,j}	1	0	0	1	0	0	0	0	0	0	0	0	0	0	=>	-E(1,3)
i = 2	0	1	A _{i,j}	1	0	0	1	0	0	0	0	0	0	0	0	0	=>	-E(1,4)
	0	0	1	A _{i,j}	0	0	0	1	0	0	0	0	0	0	0	0	=>	-E(1, j _{max} -1=5)
(jmax-2)	1	0	0	0	A _{i,j}	1	0	0	1	0	0	0	0	0	0	0	=>	-E(2, j _{max} =6)
Zeilen	0	1	0	0	1	A _{i,j}	1	0	0	1	0	0	0	0	0	0	=>	-E(3,1)
i = 3	0	0	1	0	0	1	A _{i,j}	1	0	0	1	0	0	0	0	0	=>	0
	0	0	0	1	0	0	1	A _{i,j}	0	0	0	1	0	0	0	0	=>	0
(jmax-2)	0	0	0	0	1	0	0	0	A _{i,j}	1	0	0	1	0	0	0	=>	-E(3, j _{max} =6)
Zeilen	0	0	0	0	0	1	0	0	1	A _{i,j}	1	0	0	1	0	0	=>	-E(4,1)
i = 4	0	0	0	0	0	0	1	0	0	1	A _{i,j}	1	0	0	1	0	=>	0
	0	0	0	0	0	0	0	1	0	0	1	A _{i,j}	0	0	0	1	=>	0
(jmax-2)	0	0	0	0	0	0	0	0	1	0	0	0	A _{i,j}	1	0	0	=>	-E(4, j _{max} =6)
1=5,1)	0	0	0	0	0	0	0	0	1	0	0	0	0	A _{i,j}	1	0	=>	-E(i _{max} =6,2)
Zeilen	0	0	0	0	0	0	0	0	1	0	0	1	A _{i,j}	1	0	0	=>	-E(i _{max} =6,3)
i = 5	0	0	0	0	0	0	0	0	0	1	0	0	1	A _{i,j}	1	0	=>	-E(i _{max} =6,4)
1=5, j _{max} =6)	0	0	0	0	0	0	0	0	0	0	1	0	0	1	A _{i,j}	0	=>	-E(i _{max} =6, j _{max} -1=5) - E(i _{max} -1=5, j _{max})

In der leitenden Ebene: E(i_{max},j) = 0

In der Startebene: E(1,j) = E₀

In den Randebenen: E(i,1) = E(i, j_{max}) = f(x)

$$A = \begin{pmatrix} A_{1,1} & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & A_{1,2} & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & A_{1,3} & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & A_{1,4} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & A_{2,1} & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & A_{2,2} & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & A_{2,3} & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & A_{2,4} & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & A_{3,1} & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & A_{3,2} & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & A_{3,3} & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & A_{3,4} & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & A_{4,1} & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & A_{4,2} & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & A_{4,3} & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & A_{4,4} \end{pmatrix}$$

werden muss um eine genügende Auflösung gewährleisten zu können. Typische Minimalwerte sind $I, J \geq 200$.

Für die Lösung von großen Matrix-Systemen bietet die Literatur verschiedenste Lösungsansätze. Eine Einführung in die Problematik ist z.B. in /33/ gegeben.

Die Methoden zur Lösung von linearen Gleichungssystemen werden in iterative und direkte Verfahren unterteilt /32/. Beispiele für direkte Verfahren sind das Einsetzungsverfahren, das Gleichsetzungsverfahren und das Additionsverfahren für einfache Gleichungssysteme sowie das auf dem Additionsverfahren basierende gauß'sche Eliminationsverfahren, das ein Gleichungssystem auf Stufenform bringt. Eine Variante des Gauß-Verfahrens ist die Cholesky-Zerlegung, die nur für symmetrische, positiv definite Matrizen funktioniert. Doppelt so viel Aufwand wie das Gauß-Verfahren braucht die QR-Zerlegung, die dafür stabiler ist. Die Cramer'sche Regel verwendet Determinanten, um Formeln für die Lösung eines quadratischen linearen Gleichungssystems zu erzeugen, wenn dieses eindeutig lösbar ist. Für die numerische Berechnung ist sie auf Grund des hohen Rechenaufwands jedoch nicht geeignet.

Iterative Verfahren sind beispielsweise die zur Klasse der Splitting-Verfahren gehörenden Gauß-Seidel- und Jacobi-Verfahren. Diese konvergieren nicht für jede Matrix und sind für viele praktische Probleme sehr langsam. Modernere Verfahren sind etwa vorkonditionierte Krylow-Unterraum-Verfahren, die insbesondere für große dünnbesetzte Matrizen sehr schnell sind, sowie Mehrgitterverfahren zur Lösung von Systemen, die aus der Diskretisierung bestimmter partieller Differentialgleichungen stammen.

Numerische Lösung nach dem Cholesky-Verfahren

In der folgenden Beispielanwendung wird das Cholesky-Verfahren (Symmetrische Zerlegung positiver definitiver Bandmatrizen) angewandt.

In /33/ ist eine praktische Lösungsmöglichkeit beschrieben und auch ein kompaktes, übersichtliches Programm zur Lösung in der „historischen“ Programmiersprache ALGOL publiziert (Abb. 15, Abb. 16).

4. ALGOL programs

```

procedure chobanddet (n, m) data: (a) result: (l, d1, d2) failure exit: (fail);
value m, n; integer d2, m, n; real d1; array a, l; label fail;
comment The lower half of a positive definite symmetric band matrix, A, with
m lines on either side of the diagonal is stored as an  $n \times (m+1)$  array,
a[i, k],  $i=1(1)n$ ,  $k=0(1)m$ , a[i, m] being the diagonal elements. The
Cholesky decomposition  $A=LU$ , where U is the transpose of L, is
performed and L is stored in l[i, k] in the same form as A. The reci-
procals of the diagonal elements are stored instead of the elements
themselves. A is retained so that the solution obtained can subse-
quently be improved. However, L and A can be identified in the call
of the procedure. The determinant,  $d1 \times 2^{\uparrow d2}$ , of A is also computed.
The procedure will fail if A, modified by the rounding errors, is not
positive definite;
begin integer i, j, k, p, q, r, s;
real y;
d1 := 1;
d2 := 0;
for i := 1 step 1 until n do
begin p := (if i > m then 0 else m - i + 1);
r := i - m + p;
for j := p step 1 until m do
begin s := j - 1;
q := m - j + p;
y := a[i, j];
for k := p step 1 until s do
begin y := y - l[i, k] × l[r, q];
q := q + 1
end k;
if j = m then
begin d1 := d1 × y;
if y = 0 then
begin d2 := 0;
go to fail
end;
L1: if abs(d1) ≥ 1 then
begin d1 := d1 × 0.0625;
d2 := d2 + 4;
go to L1
end;
L2: if abs(d1) < 0.0625 then
begin d1 := d1 × 16;
d2 := d2 - 4;
go to L2
end;
if y < 0 then go to fail;
l[i, j] := 1/sqrt(y)
end
else
l[i, j] := y × l[r, m];
r := r + 1
end j
end i
end chobanddet;

```

Abb 15: Programmbeispiel zum „Cholesky-Verfahren“, entnommen aus /32/

```

procedure chobandsol (n, m) data:(r, l, b) result:(x);
value m, n, r; integer m, n, r; array l, b, x;
comment Solves  $Ax=b$ , where  $A$  is a positive definite symmetric band matrix
with  $m$  lines on either side of the diagonal and  $b$  is an  $n \times r$  matrix
of  $r$  right-hand sides. The procedure chobandsol must be preceded by
chobanddet in which  $L$  is produced in  $l[i, k]$ , from  $A$ .  $Ax=b$  is solved
in two steps,  $Ly=b$  and  $Ux=y$ . The matrix  $b$  is retained in order
to facilitate the refinement of  $x$ , but  $x$  is overwritten on  $y$ . However,
 $x$  and  $b$  can be identified in the call of the procedure;

begin integer i, j, k, p, q, s;
  real y;
  s := m - 1;
  for j := 1 step 1 until r do
    begin comment solution of  $Ly=b$ ;
      for i := 1 step 1 until n do
        begin p := (if i > m then 0 else m - i + 1);
          q := i;
          y := b[i, j];
          for k := s step -1 until p do
            begin q := q - 1;
              y := y - l[i, k] × x[q, j]
            end k;
          x[i, j] := y × l[i, m]
        end i;
      comment solution of  $Ux=y$ ;
      for i := n step -1 until 1 do
        begin p := (if n - i > m then 0 else m - n + i);
          y := x[i, j];
          q := i;
          for k := s step -1 until p do
            begin q := q + 1;
              y := y - l[q, k] × x[q, j]
            end k;
          x[i, j] := y × l[i, m]
        end i;
      end j
    end chobandsol;

```

Abb 16: Programmbeispiel zum „Cholesky-Verfahren“, entnommen aus /32/

Für die folgenden zweidimensionalen Demo-Berechnungen wird analog zur Anordnung in Abb. 17 folgender 3-schichtiger „Testkörper“ *Body* definiert:

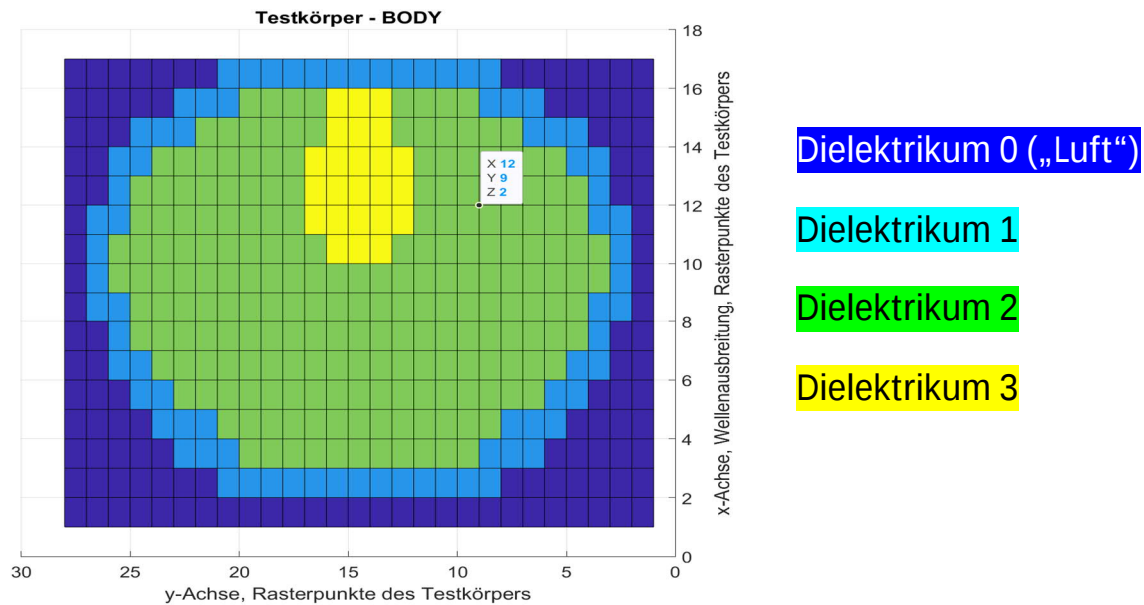


Abb. 17 „Testkörper“ *Body*

Die Anwendung des Codes von Abb. 15 bzw. Abb. 16 und transkribiert für MATLAB („BandMatrix.m“, siehe Anhang) liefert gut brauchbare Lösungen (Abb. 18, Abb. 19, Abb. 20).

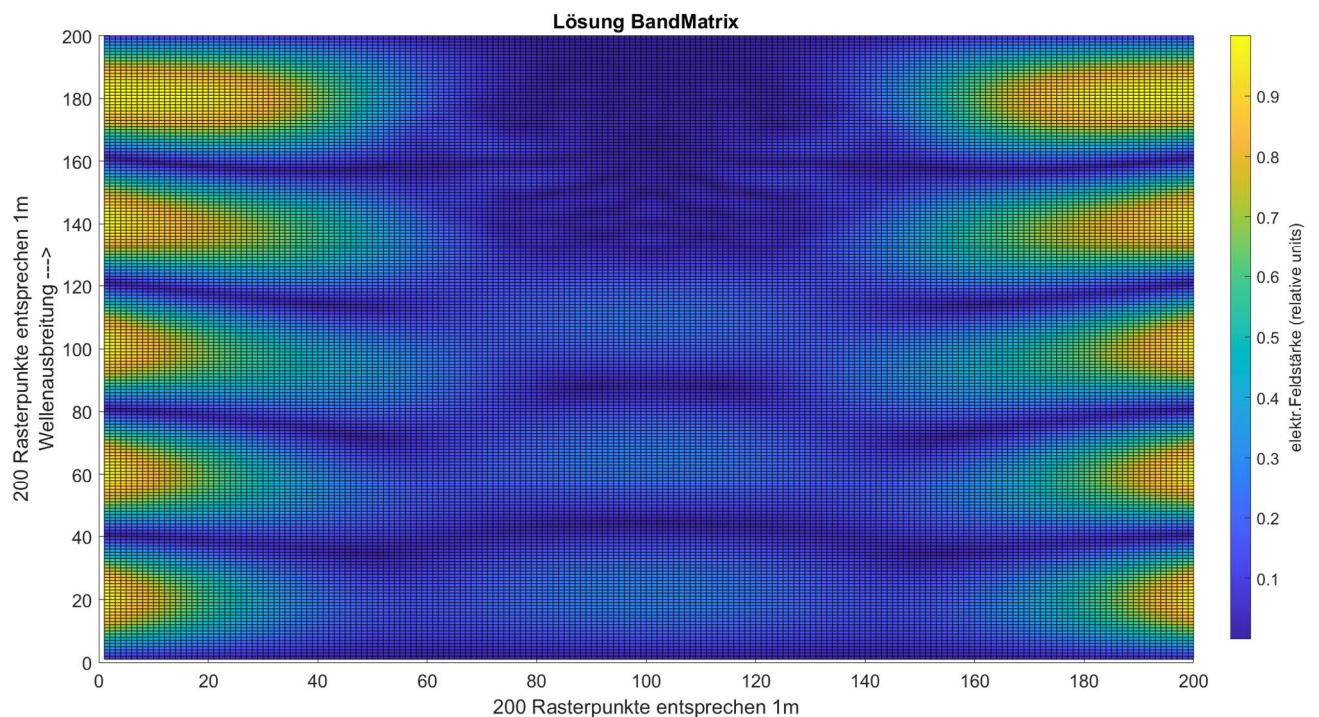


Abb. 18: Verlauf des elektrischen Feldes in- und außerhalb eines Testkörpers
(Anordnung des Testkörpers vgl. Abb. 19)

Der dargestellte Bereich beträgt 1 Meter x 1 Meter. Das Programm liefert einige Parameter:

Eingabe Rasteranzahl als Vielfaches von 50: 4

Frequenz: 1 GHz

EPS0: -3.993893 0.000000i

EPS1: -3.969464 -0.003298i

EPS2: -3.816782 -0.032979i

EPS3: -3.511418 -0.109931i

CHOBANDDET

Startzeit: 19-Mar-2021 10:10:03

M = 200

L[40000][201] X[40000] angelegt

CHOBANDSOL

Zwischenzeit: 19-Mar-2021 10:13:16

Endzeit: 19-Mar-2021 10:14:55

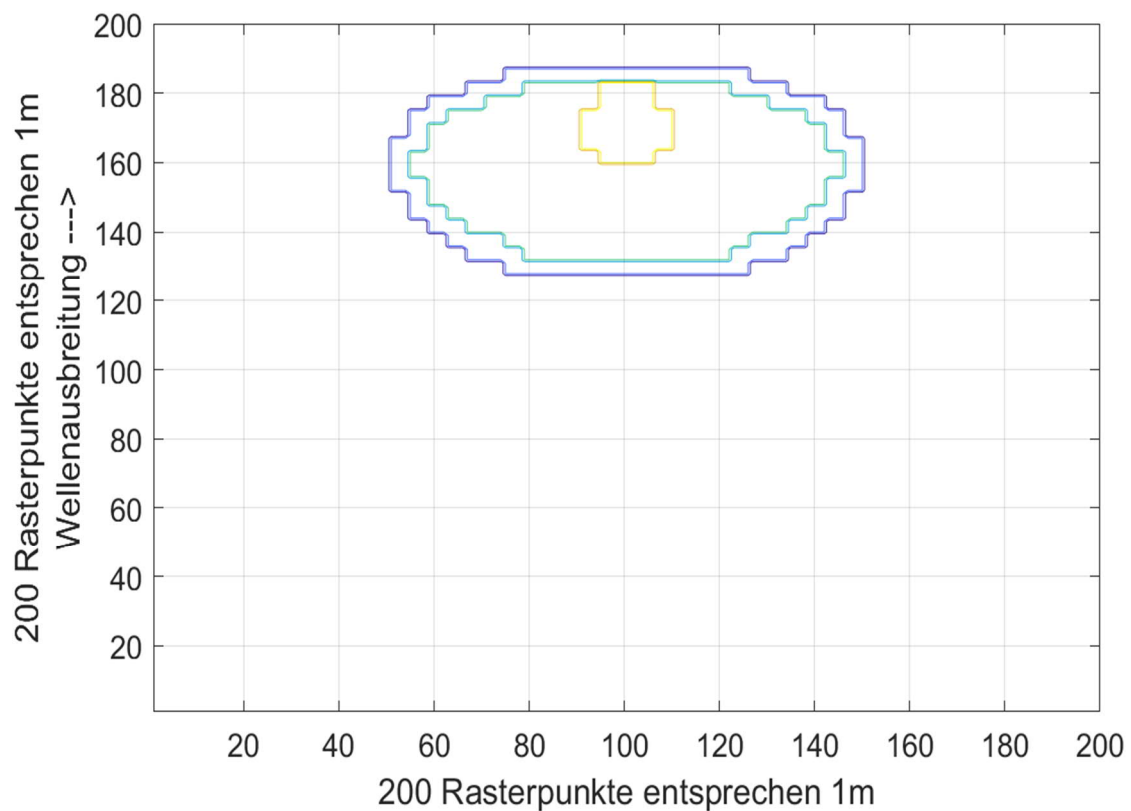


Abb. 19: Anordnung des Testkörpers, am oberen Ende befindet sich eine Reflexionsebene

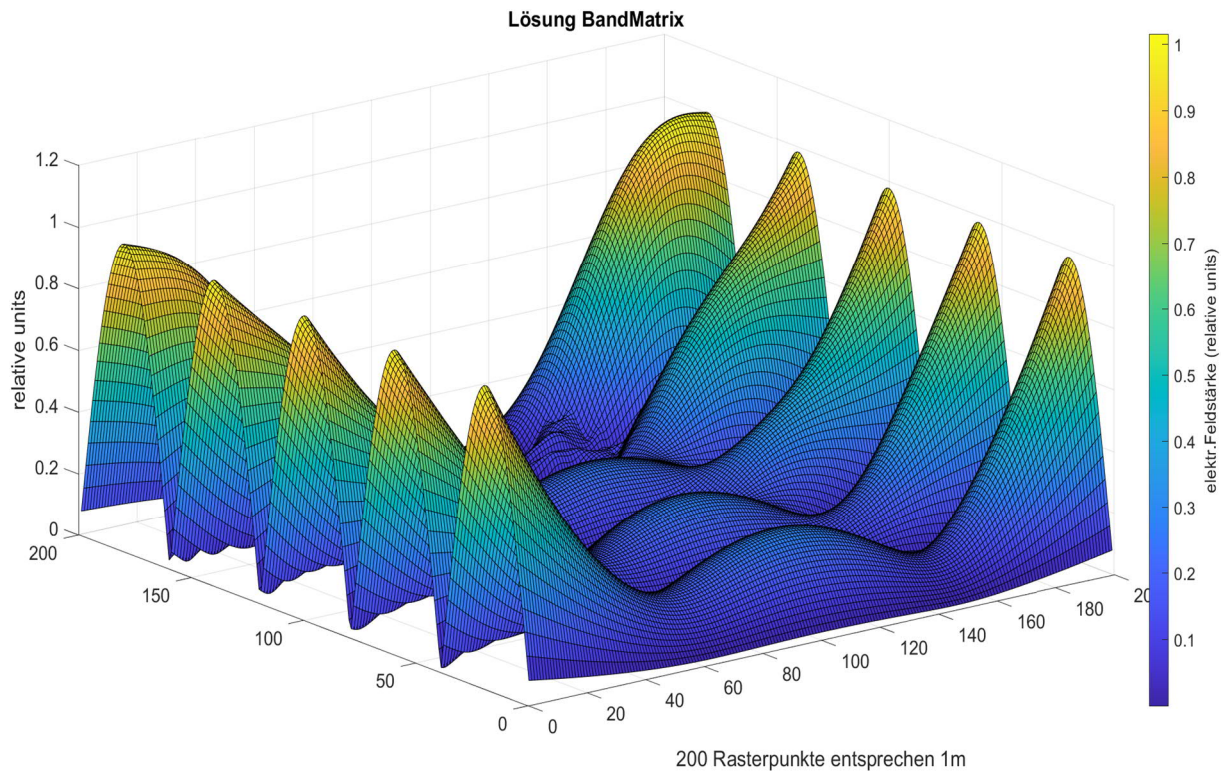


Abb. 20: Verlauf der elektrischen Feldstärke in „3D“-Darstellung

Wird zu Testzwecken $\text{EPS1} = \text{EPS2} = \text{EPS3} = \text{EPS0}$ gesetzt (d.h. der Testkörper „wird zu Luft“) zeigt sich der ungestörte Wellenverlauf.

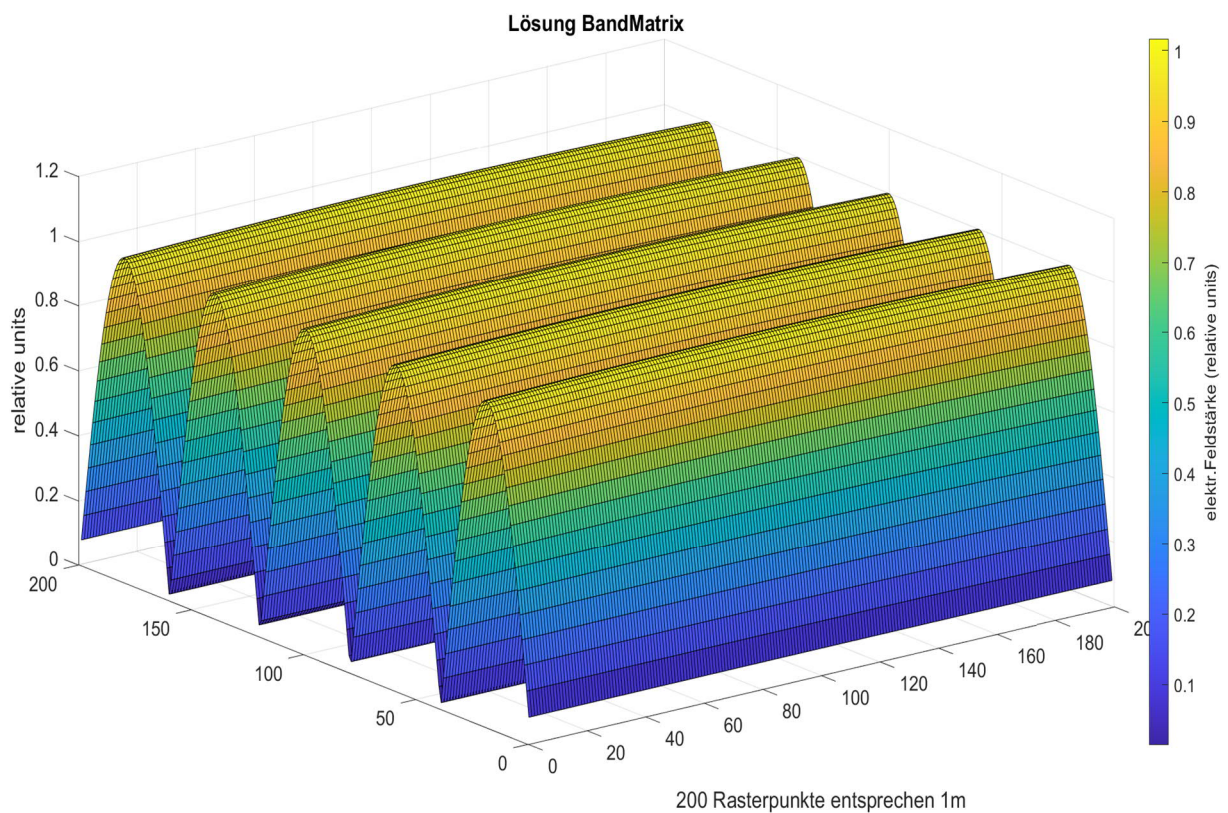


Abb. 21: Test, für $\text{EPS1} = \text{EPS2} = \text{EPS3} = \text{EPS0}$

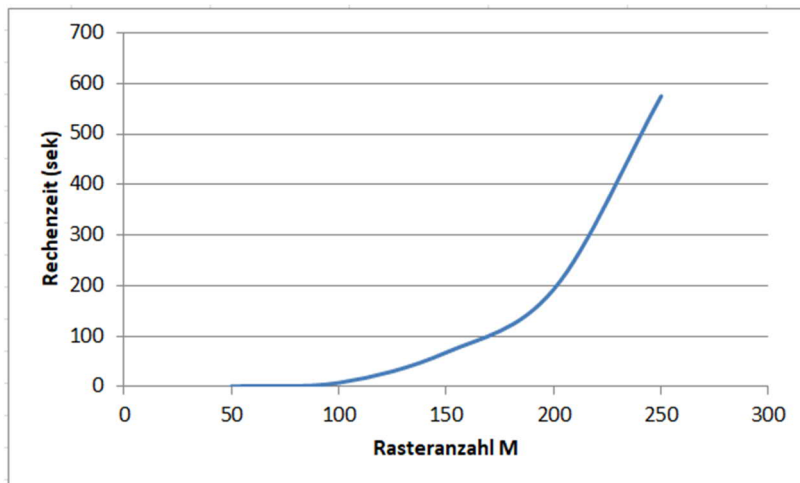
Die oben beschriebene Lösungsmethode hat jedoch einen entscheidenden Nachteil:

Mit steigender Auflösung (d.h. mit wachsendem „M“ im Programmbeispiel) steigt der Speicherplatzbedarf und die Rechenzeit extrem an.

Die angelegten Arrays (im Programm mit L und X bezeichnet) haben eine Größe

$L[M \times M][M+1]$ bzw. $X[M \times M]$, d.h. für $M = 200$:

$L[40000][201]$ $X[40000]$. Jede Variable (complex, hohe Genauigkeit) benötigt 16 Byte.



M	Sek
50	1
100	8
150	68
200	193
250	575

Abb. 21: Anstieg der Rechenzeit in Abhängigkeit der Rasterzahl M

Numerische Lösung mit einer Band-Matrix in PYTHON

Eine Alternative zu der Programmierung in MATLAB, das eine kostenpflichtige Lizenzierung voraussetzt, ist die Verwendung von PYTHON. Die Python Software Foundation (PSF) ist die Organisation hinter dem Open-Source-Projekt. Sie ist der Herausgeber und Rechtsinhaber der Python-Software-Foundation-Lizenz und besitzt die Markenrechte an Python. Die Python-Software-Foundation-Lizenz ist ähnlich der BSD-Lizenz („Berkeley Software Distribution“) und kompatibel mit der GNU General Public License /34/.

Für Python gibt es eine Reihe von zusätzlich installierbaren Programmbibliotheken. Z.B. „NumPy“, das eine einfache Handhabung von Vektoren, Matrizen oder generell großen mehrdimensionalen Arrays ermöglicht, und „Matplotlib“, das es erlaubt, mathematische Darstellungen aller Art anzufertigen /35/.

Zur Installation ist „pip“, ein Paketverwaltungsprogramm für Python-Pakete aus dem Python Package Index (PyPI), dienlich.

Eine analoge Software-Realisierung der obigen Feldberechnung „BandMatrix.m“ ist für Python mit „BandMatrix.py“ (siehe Anhang) realisiert (Abb. 22).

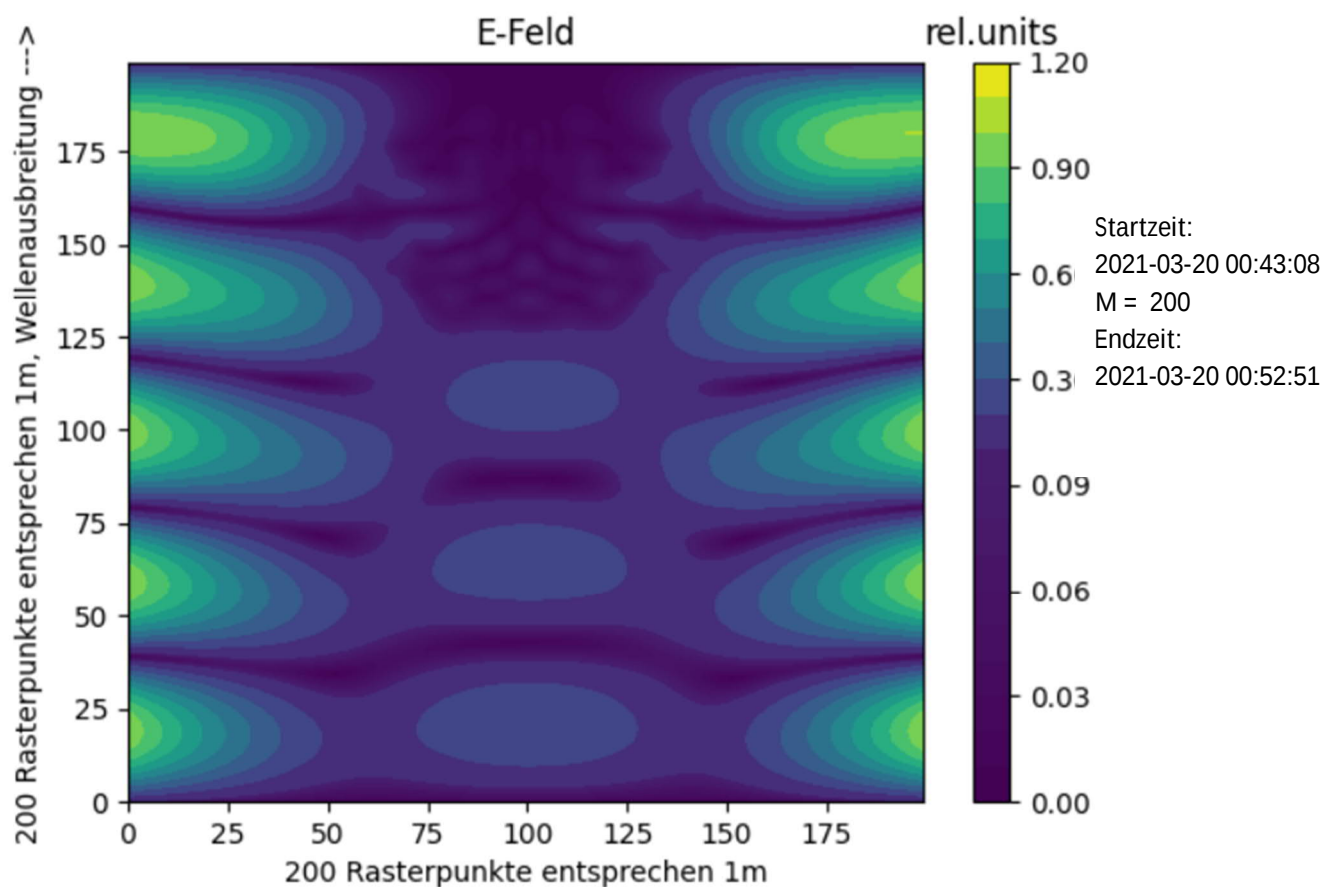


Abb. 23: Feld-Berechnung mit Python, „BandMatrix.py“

Das Ergebnis ist vergleichbar mit dem Ergebnis entsprechend Abb. 18

Numerische Lösung mit MATLAB

Am Aufbau der Koeffizientenmatrix A erkennt man, dass eine überwiegende Anzahl von Matrix-Elementen Null sind.

In der numerischen Mathematik bezeichnet man als dünnbesetzte oder schwachbesetzte Matrix (englisch sparse matrix) eine Matrix, bei der so viele Einträge aus Nullen bestehen, dass man nach Möglichkeiten sucht, dies insbesondere hinsichtlich Algorithmen sowie Speicherung auszunutzen /36/.

MATLAB bietet zur Lösung von linearen Gleichungssystemen den „Backslash-Operator“ an /37/.

$$A x = B$$

$$x = A \setminus B$$

Für den Matlab-Anwender ergeben sich damit folgende Vorteile:

- Er braucht sich nicht um die Auswahl eines geeigneten Lösungsverfahrens zu kümmern. Der mit dem Backslash-Operator aufgerufene Algorithmus analysiert die übergebenen Matrizen A und B und wählt selbst ein geeignetes Verfahren. Die Algorithmen sind offensichtlich sehr gut implementiert und ausgesprochen schnell.
- Die Matrizen können auch im Speicherformat "Sparse" gespeichert sein, was speziell für sehr große Gleichungssysteme häufig die einzige Möglichkeit ist, um mit erträglichen Rechenzeiten und realisierbarem Speicherbedarf auszukommen.

Eine Übersicht über die in MATLAB verwendeten Verfahren ist z.B. in /38/ gegeben.

Zur Unterstützung dieser Techniken ist es notwendig, die Elemente-Anzahl einer Matrix so weit wie möglich zu reduzieren. Dies gilt speziell für Band-Matrizen.

Bandmatrizen werden normalerweise durch Speichern der Diagonalen im Band gespeichert; der Rest ist implizit Null /39/.

Beispielsweise hat die unten skizzierte 6-mal-6-Matrix die Bandbreite 1 (bedeutet je eine besetzte Nebendiagonale neben der Hauptdiagonale).

$$\begin{bmatrix} B_{11} & B_{12} & 0 & \dots & \dots & 0 \\ B_{21} & B_{22} & B_{23} & \ddots & \ddots & \vdots \\ 0 & B_{32} & B_{33} & B_{34} & \ddots & \vdots \\ \vdots & \ddots & B_{43} & B_{44} & B_{45} & 0 \\ \vdots & \ddots & \ddots & B_{54} & B_{55} & B_{56} \\ 0 & \dots & \dots & 0 & B_{65} & B_{66} \end{bmatrix}$$

Das Band wird gespeichert als 6-mal-3 Matrix

$$\begin{bmatrix} 0 & B_{11} & B_{12} \\ B_{21} & B_{22} & B_{23} \\ B_{32} & B_{33} & B_{34} \\ B_{43} & B_{44} & B_{45} \\ B_{54} & B_{55} & B_{56} \\ B_{65} & B_{66} & 0 \end{bmatrix}$$

In „SparseMatrix.m“ (siehe Anhang) wird die Feldberechnung mithilfe des Backslash-Operator demonstriert (Abb. 24, Abb. 25).

Das Ergebnis ist ident mit vorhergegangenen Lösungsansätzen, jedoch ist die Rechenzeit um einiges kürzer:

Für M = 200:

BandMatrix.m	193 Sekunden
SparseMatrix.m	11 Sekunden

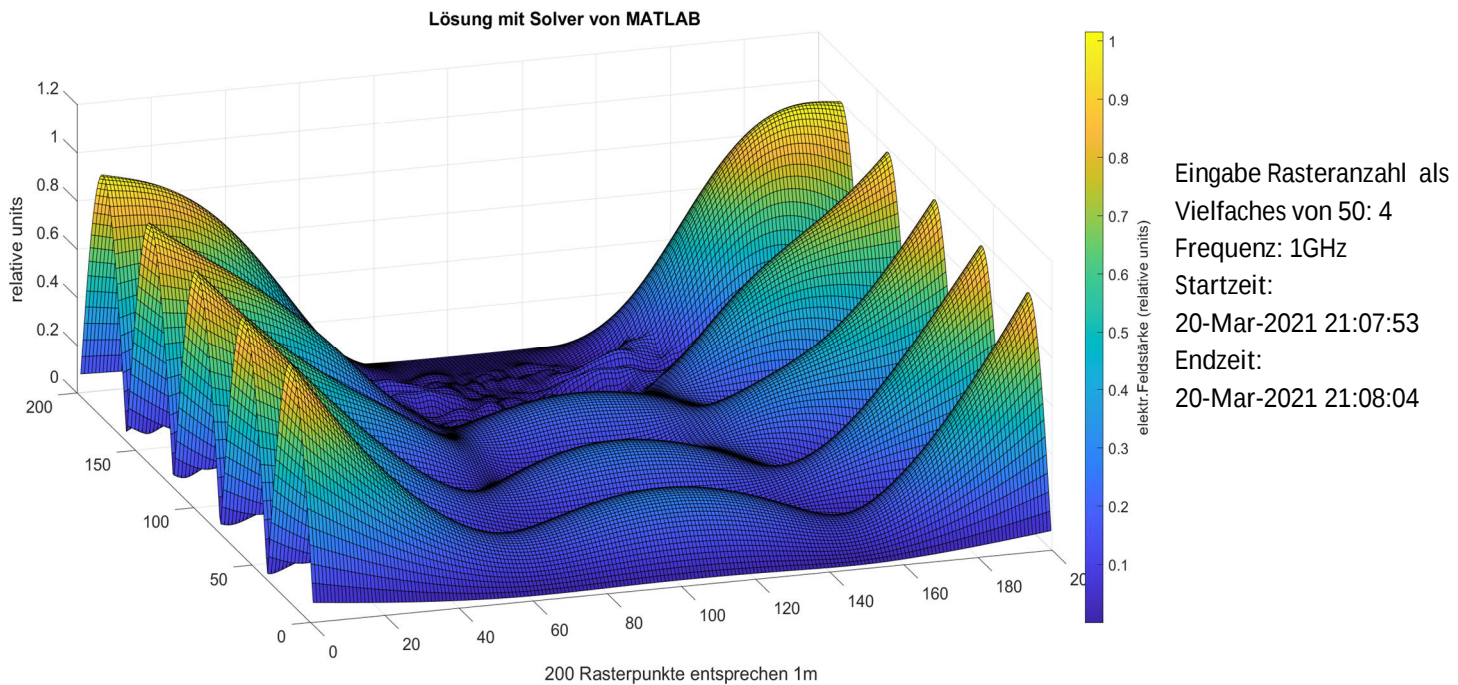


Abb. 24: E-Feld Berechnung mit „SparseMatrix.m“

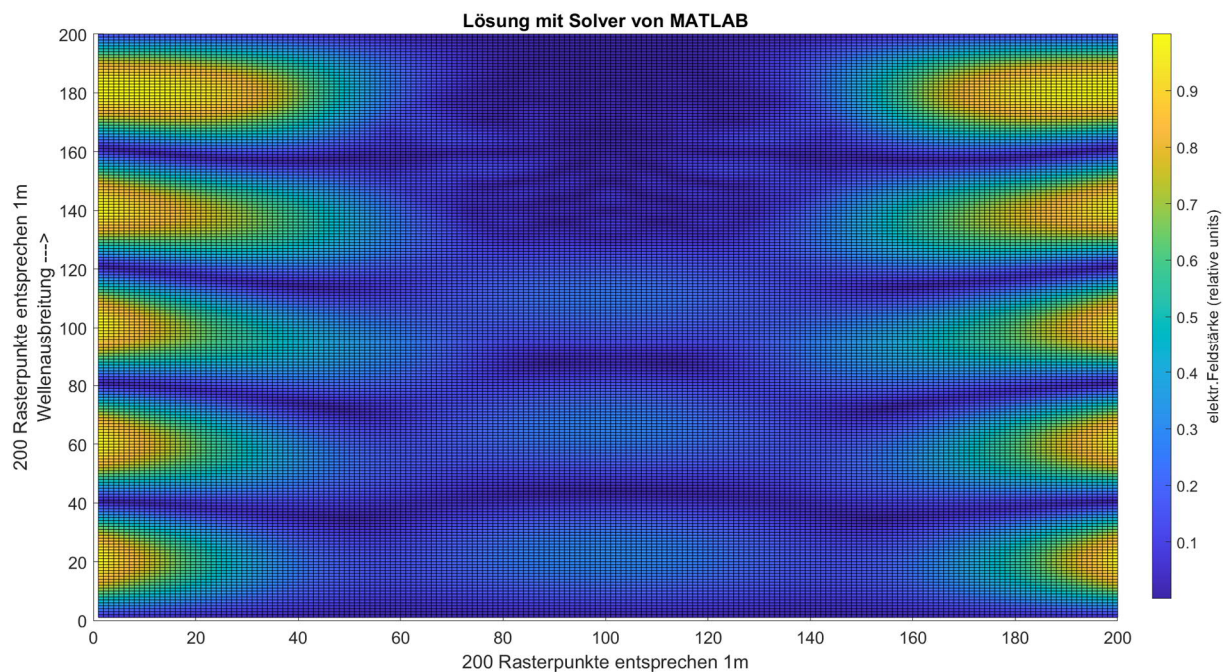


Abb. 25: E-Feld Berechnung mit „SparseMatrix.m“
(vergleiche Abb. 18 bzw. Abb. 23)

Erweiterung auf drei Dimensionen

In den vorangegangenen Kapiteln wurden Beispiele für die prinzipiellen Lösungsmöglichkeiten von Feldberechnungen im 2-dimensionalen Bereich gebracht.

Um im weiteren ein anschauliches Beispiel für das 3-dimensionale Verhalten einer polarisierten Welle in dielektrischem Material zu bekommen, wird folgender Lösungsweg gewählt: der zu untersuchende Bereich wird in Schichten geschnitten, jede Schicht wird 2-dimensional behandelt und die Ergebnisse der Feldberechnungen werden wieder zu drei Dimensionen zusammengesetzt. Man erhält so eine „quasi-3-dimensionale-Feldberechnung“ (vgl. Abb. 26). Die Wahl der Ausrichtung der (parallelen) Schnittebenen kann beliebig bezüglich des Koordinatensystems sein.

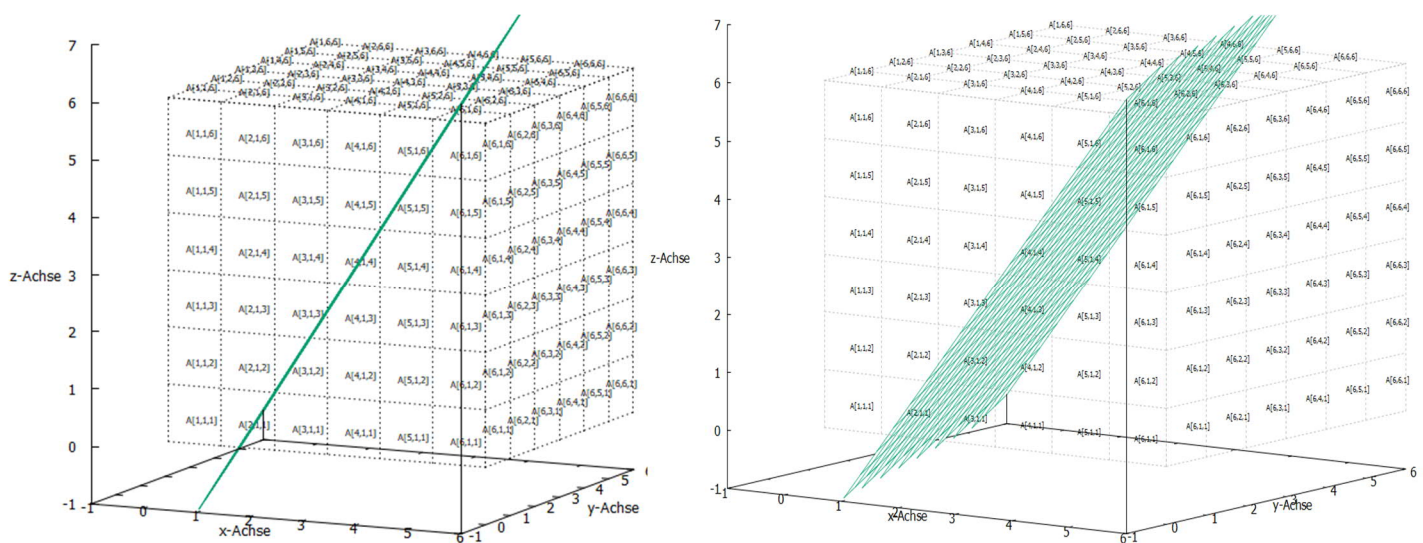


Abb. 26: zur Methode der „quasi-3-dimensionalen“ Feldberechnung (Anhang: „3Dgrid.plt“): je nach Sichtwinkel stellt sich die Schnittebene unterschiedlich dar.

Der Vollständigkeit halber soll ein weiteres Softwarepaket erwähnt sein, das ebenfalls gut geeignet für numerische Lösungsverfahren und ihre grafische Darstellung ist.

GNU Octave ist eine freie Software zur numerischen Lösung mathematischer Probleme, wie zum Beispiel Matrizenrechnung, Lösen von (Differential-)Gleichungssystemen, Integration etc. Berechnungen können in Octave mit einer Skriptsprache durchgeführt werden, die weitgehend zu dem proprietären MATLAB und zum ebenfalls freien FreeMat kompatibel ist [40].

Die Funktionskompatibilität von GNU Octave zur Basisversion von MATLAB (ohne Toolboxes) ist überwiegend gegeben. Fast vollständige Kompatibilität zu MATLAB lässt sich durch das Zusatzpaket octave-forge und andere freie Ersatzfunktionen erreichen.

Die hier vorgestellten Lösungsansätze werden allerdings direkt mit MATLAB durchgeführt.

Biologische MRT-Modelle als Testkörper

Es stellt sich nun die Frage, an welchem „Testkörper“ die Berechnungsmodelle im Weiteren angesetzt werden sollen. Als praktisch gut realisierbar, anschaulich und auch interessant wurden nun Modelle verwendet, die eigentlich für andere Fragestellungen entwickelt wurden. Es handelt sich um (humane) Testkörper bzw. Organteile, deren Gewebeaufbau für medizinische Zwecke durch Untersuchungen mit MRT (Magnetresonanztomografie) gut bekannt ist.

Die Daten werden bereitgestellt von der IT'IS Foundation /41/.

Die „Foundation for Research on Information Technologies in Society (IT'IS)“ ist eine unabhängige, gemeinnützige Forschungstiftung, die sich der Verbesserung und Verbesserung der Lebensqualität von Menschen durch Verbesserung der Sicherheit und Qualität neu entstehender elektromagnetischer Technologien widmet. Die Stiftung wurde 1999 auf Initiative und mit Unterstützung der Eidgenössischen Technischen Hochschule (ETH) Zürich, der globalen Mobilfunkbranche und mehrerer Regierungsbehörden gegründet und hat ihren Hauptsitz in Zürich, Schweiz.

Von /42/ lassen sich sowohl Körpermodelle wie auch die dielektrischen Eigenschaften /43/ verschiedenster Körpergewebe herunterladen.

Die „Virtual Population (ViP)“-Modelle sind eine Reihe detaillierter hochauflösender anatomischer Modelle, die aus Magnetresonanzbildern von Freiwilligen erstellt wurden. Seit ihrer Einführung sind die ViP-Modelle zum Goldstandard für biophysikalische Modellierungsanwendungen „in silico“ (bezeichnet Vorgänge, die im Computer ablaufen) geworden.

Für ausgewählte Körperregionen sind die Gewebe-Strukturdaten als Voxel-Werte in „*.mat, *.raw, *.nii“-Format download-bar und können so direkt von MATLAB-Routinen eingelesen werden /45/.

Voxel (zusammengesetzt aus dem englischen volume vox und el von elements) bezeichnet in der Computergrafik einen Gitterpunkt („Bild“punkt, Datenelement) in einem dreidimensionalen Gitter. Dies entspricht einem Pixel in einem 2D-Bild, einer Rastergrafik /44/.

Eine umfangreiche Datenbank für dielektrische Eigenschaften für unterschiedlichste Körpergewebe findet sich /46/.

Weiteres exzellentes Datenmaterial für humane Voxelgrafiken bietet die „Zubal Phantom site“ /47/, von der mehrere Sätze segmentierter Bilder von zwei lebenden menschlichen Männern heruntergeladen werden können. Das Torsophantom besteht aus einem CT-basierten Torso und einem CT-basierten Kopf /48/.

Für die folgenden Berechnungen wurde das „MRI Head Phantom“ verwendet. Es ist dies ein detailliertes Kopf / Gehirn-Phantom. Die Datei „det_head_u2.med“ ist ungefähr 8 MB groß

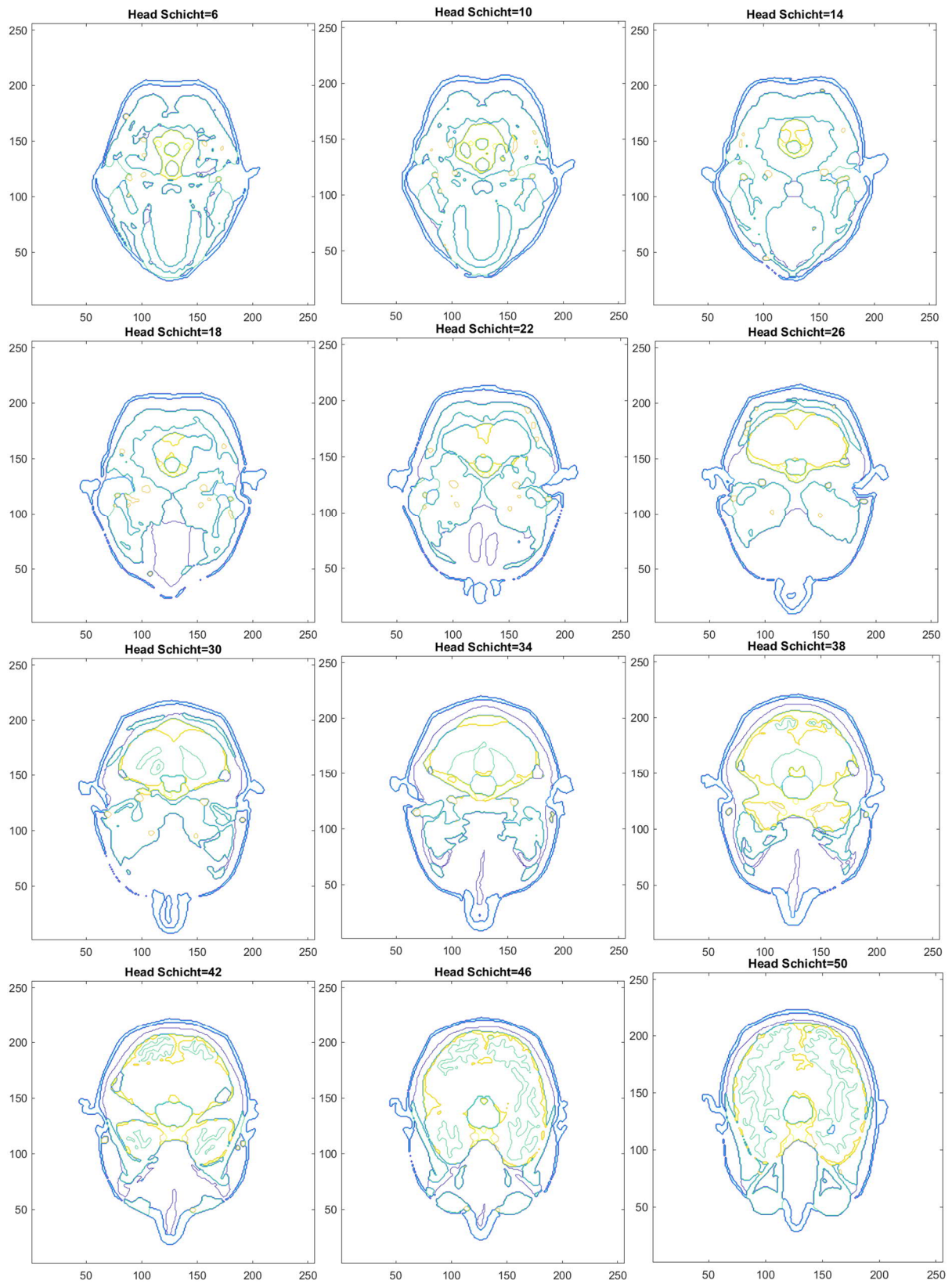
und beinhaltet 128 Schnittdateien mit je 256×256 Bytes von den MRT-Bildern. Die Pixelgröße in der x, y-Ebene beträgt 1,1 mm und die Auflösung in der z-Achse ergibt eine Schichtdicke von 1,4 mm.

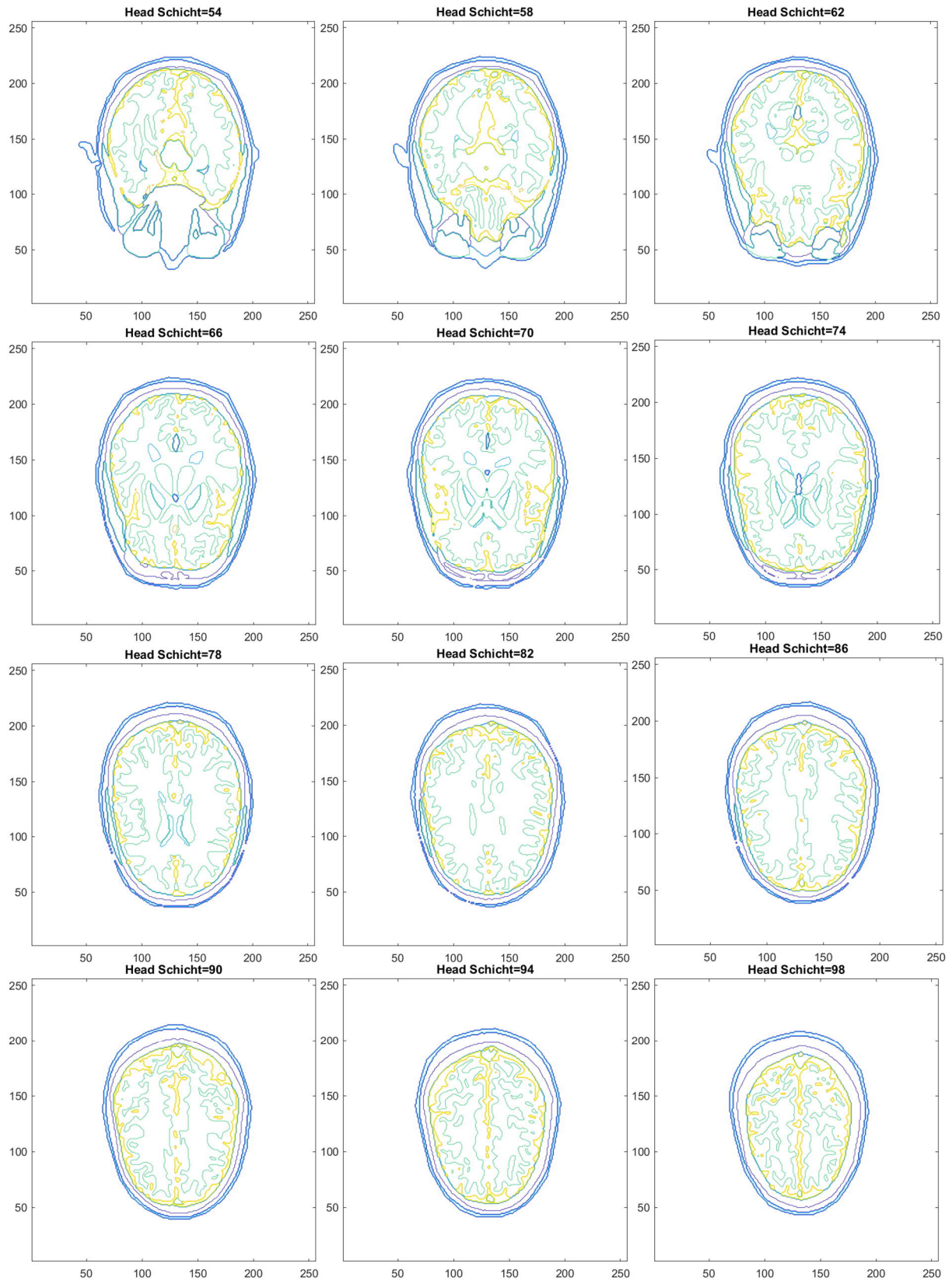
Zur Identifikation der Gewebeart eines Voxels ist für jedes Voxel zusätzlich zur Koordinate (x, y, z) eine Gewebe-ID angegeben.

Die Daten für die dielektrischen Gewebeigenschaften werden aus /46/ entnommen (vgl. Tabelle 2).

Electrical Properties Assignment (Permittivity, Conductivity)		Conductivity S/m (2.5 GHz)	Permittivity (2.5 GHz)
Tissue Type	ID		
Freespace (Lossless)	0,15,104	0	1
Skin Dry	1	1.4876	37.952
Skin Wet	72	1.6223	42.783
Fat	22, 98	0.10672	5.2749
Skull (Bone Cortical)	4,5,70,71,76,81,82,99,102,125	0.40411	11.352
Spongy bone (Bone Cancellous)	100	0.82286	18.491
White Matter	83,107,111	1.2403	36.107
Gray Matter	89,95,96,101,103,105,106,108,109,112,114,117,118,124	1.843	48.83
Blood	23	2.5878	58.181
CSF (Cerebrospinal fluid)	2,75,92,115,122	3.5041	66.168
Dura	84,97,113	1.6983	41.971
Bonemarrow	26	0.097266	5.2914
Cerebellum	77	2.1342	44.716
Spinal cord	3,85,91,120,123	1.1093	30.095
Eye tissue (Sclera)	74,110,116,119	2.0702	52.558
Eye (Lense)	121	1.5347	44.57
Cartilage	30	1.7949	38.663
Muscle (Parallel)	9	1.773	52.668
Muscle (Tongue)	78	1.8396	52.558
Muscle (Trachea)	29	1.4774	39.676
Stomach (esophagus)	16	2.2546	62.078

Tabelle2: Zuordnung Tissue-ID → Conductivity/Permittivity @ 2.5 GHz





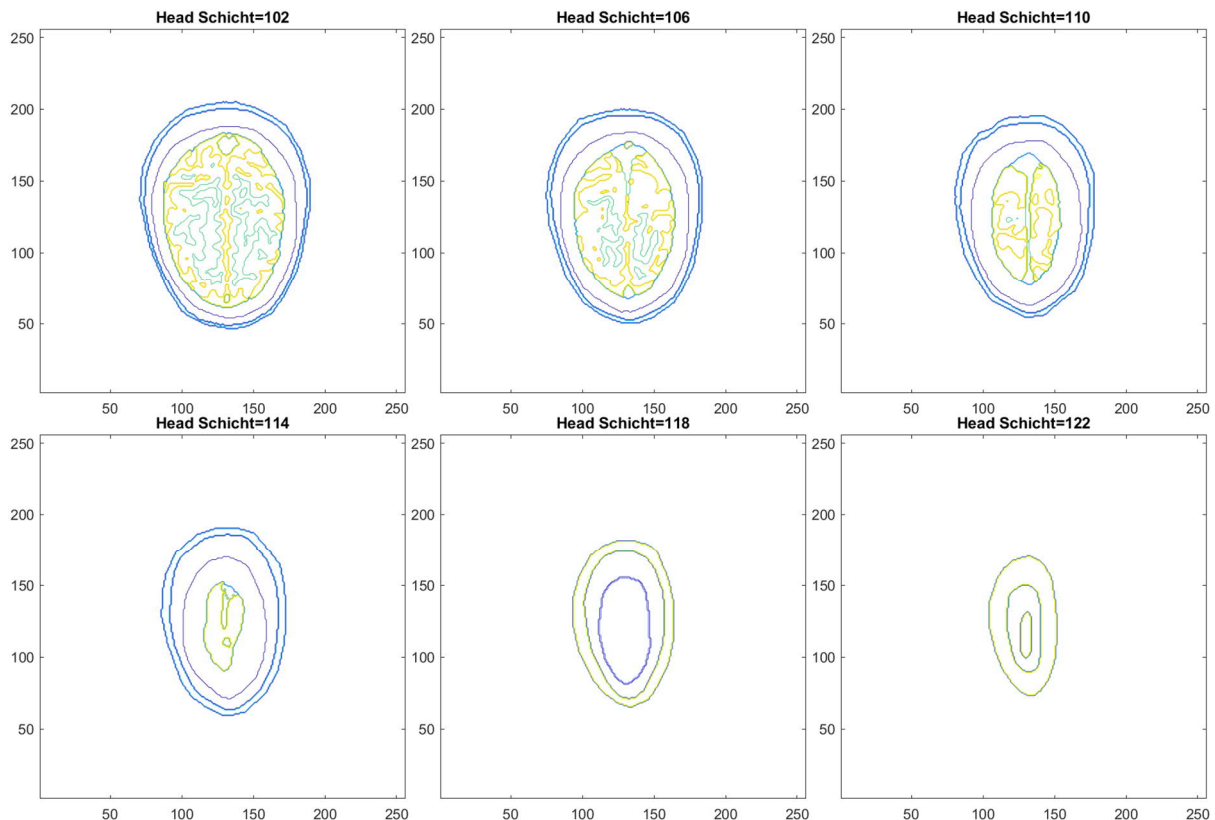


Abb. 27: Schnittbilder, Gewebegrenzungen, CT-basierter Kopf aus /48/
(in dieser Übersicht wird nur jede 4. Schicht dargestellt)

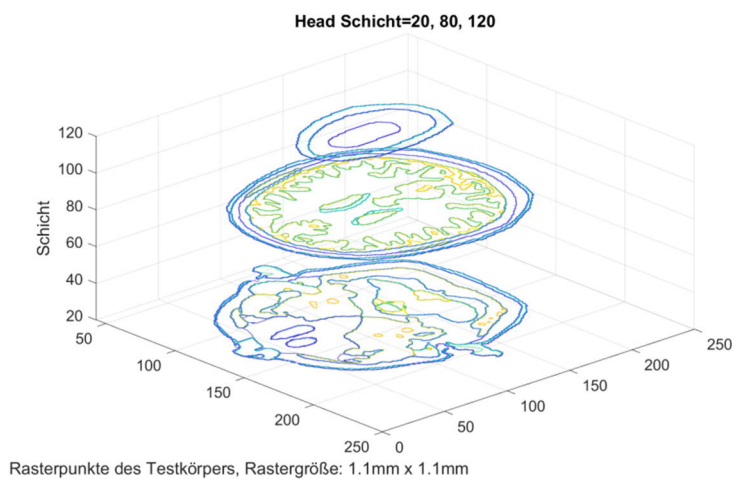


Abb. 28a: Kopf3D_Zubal.m

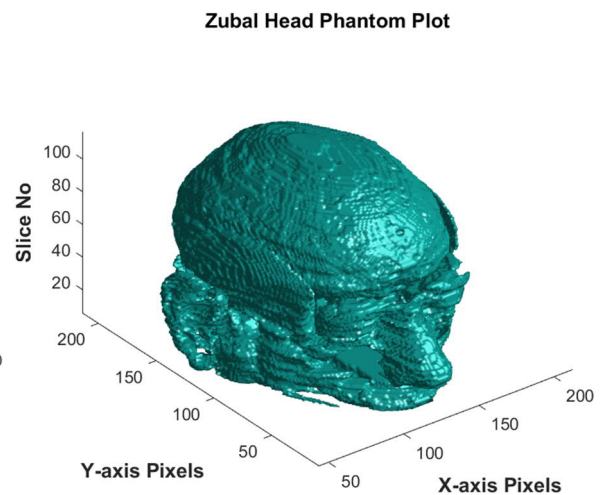


Abb. 28b: Zubal.m

Im Programm „Zubal.m“ (siehe Anhang) wird demonstriert, wie die CT-Gewebesschicht – Daten mit den zugehörigen Dielektrizitätsgrößen (Dielektrizitätskonstante/Permittivität ϵ_r bzw. Leitfähigkeit σ) hinterlegt werden. Die zu untersuchende Schicht (z.B. Schicht = 50) wird in das von der EM-Welle (z-polarisiert, Wellenausbreitung in y-Richtung, frontal zur Gesichtsebene) bestrahlte MxM-Feld gelegt (Abb. 28a) und es wird die Verlustleistungsdichte berechnet.

Die Verlustleistungsdichte P_V ergibt sich aus

$$P_V = \sigma \cdot E^2 = \omega \cdot \text{Imag}(\underline{\varepsilon}) \cdot E^2 \quad [\text{W} \cdot \text{m}^{-3}]$$

In Abb. 29b bzw. Abb. 29c zeigt sich, dass es in einem Bereich (ca. im Bereich Nase / Augen) zu einer signifikanten Überhöhung der lokalen Verlustleistungsdichte kommt. Es ist dies ein Zeichen eines „Hot-Spots“, verursacht durch eine stehende Welle, wenn die Geometrie der Gewebsschichten „zufälligerweise“ dies ermöglicht.

Zusätzlich ist bei der hier untersuchten geometrischen Anordnung am „hinteren“ Ende des MxM-Feldes eine Reflexionsebene angeordnet (vgl. „Definition von Randbedingungen“, Seite 28 bzw. Abb. 14). Das Auftreten von Hot-Spots ist jedoch hauptsächlich durch Reflexionen und damit verbundene stehende Wellen an den Schichtübergängen innerhalb des Testkörpers verursacht.

Die Grafiken sind nicht gedacht um Absolutwerte darzustellen, sondern sie sollen einen Überblick über das prinzipielle Wirken der EM-Welle aufzeigen.

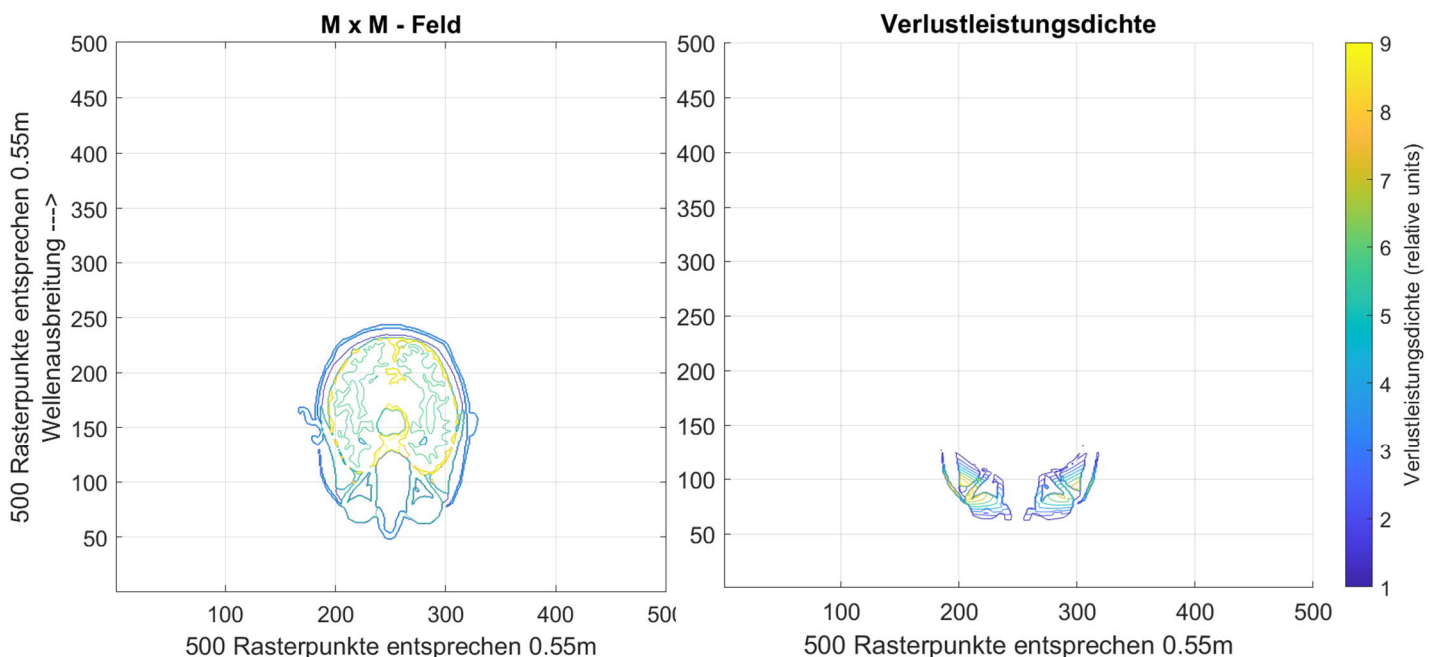


Abb. 29a: zu untersuchende Schicht im MxM-Feld

Abb. 29b: Region erhöhter P_V , „Hot Spots“

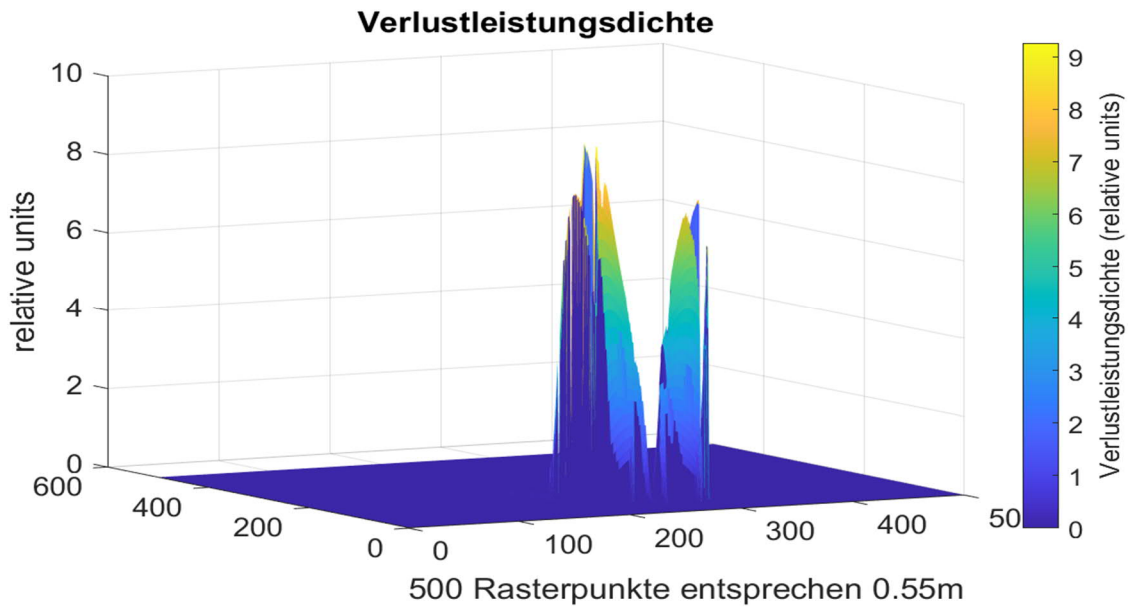


Abb. 29c: Region erhöhter Verlustleistungsdichte P_V , „Hot Spots“

Das Programm liefert einige Parameter:

Eingabe von M: 500

M = 500 $f = 2.5 \text{ GHz}$

Startzeit: 23-Mar-2021 17:10:16

Voxel = 0.0011 m Abstand Start - Reflex-Ebene: 0.55 m

Zwischenzeit: 23-Mar-2021 17:10:49 (MxM gebildet)

Zwischenzeit: 23-Mar-2021 17:20:17 (sparse Matrix A ist angelegt)

Zwischenzeit: 23-Mar-2021 17:20:17 (Matrix B ist angelegt)

Endzeit: 23-Mar-2021 17:20:19

Eine Erweiterung auf drei Dimensionen wird nun im Programm „SchichtenZubal.m“ (siehe Anhang) gezeigt.

Es werden mehrere Schichten gleichzeitig berechnet und es kann auch die Einfallsrichtung der EM-Welle festgelegt werden. Die Schnitt-Richtung zur Schichtenbildung kann mit Hilfe eines „Normalvektors“ definiert werden. Im Programm-Beispiel ist aus Gründen der Anschaulichkeit der Normalvektor parallel zur z-Achse gelegt, und die Schichtenanzahl ist auf 6 begrenzt.

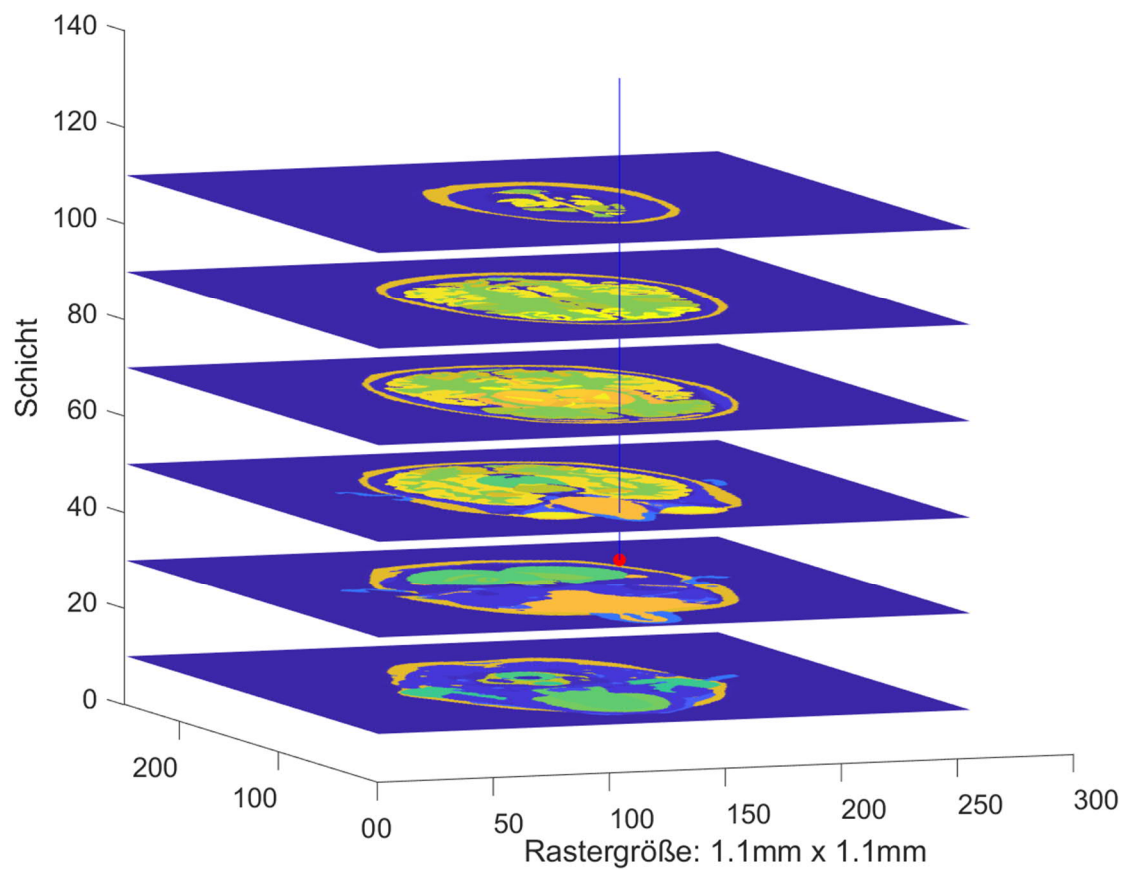


Abb. 30a: Festlegung der Schnitttrichtung mittels Normalvektor (SchichtenZubal.m)

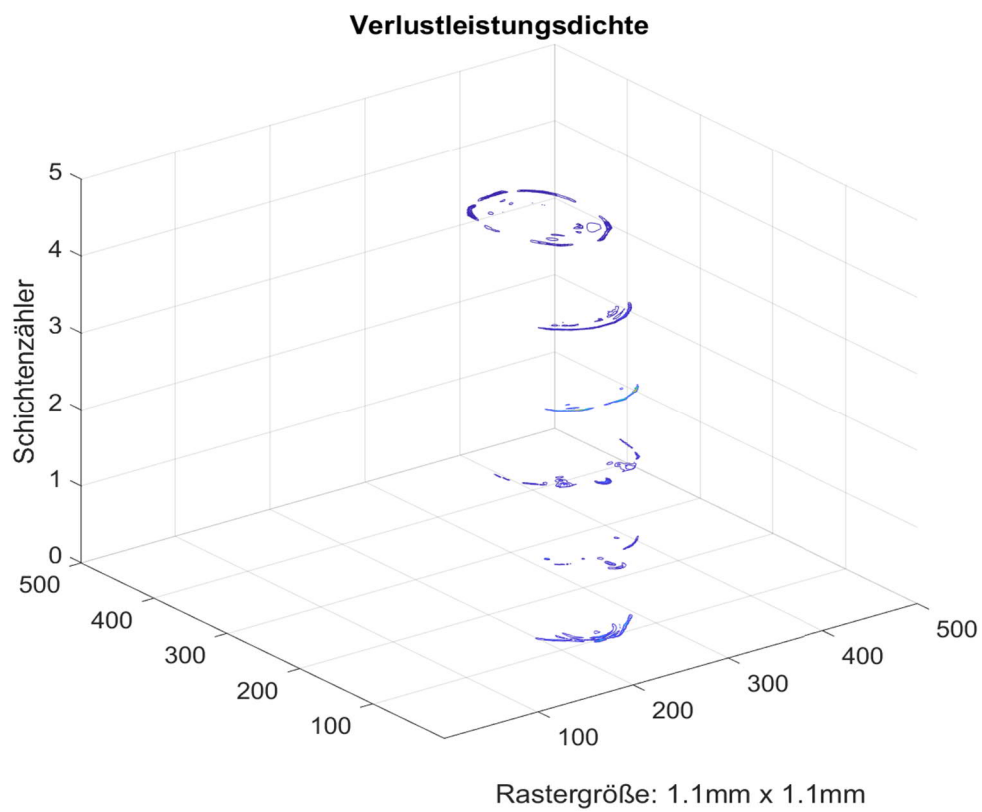


Abb. 30b: Verlustleistungsdichte mit „Hot Spot“ Intensität (SchichtenZubal.m)

Die Rechenzeit für das obige Beispiel (6 Schichten) beträgt ca. 70 Minuten:

M = 500 f = 2.5 GHz

Startzeit: 24-Mar-2021 19:22:26

Voxel = 0.0011 m Abstand Start - Reflex-Ebene: 0.55 m

Endzeit: 24-Mar-2021 20:32:57

Zur Problematik „Rasterung – Frequenz/Wellenlänge – Rechenzeit“

Bei der numerischen Lösung der Feldverteilung wird eine Rasterung vorgenommen (Vgl. Abb. 14). Bei der Wahl der Rasterbemessung (in Abb. 14 mit δ bezeichnet, bzw. Größe der Voxel) sowie der Anzahl der Rasterpunkte M ist zu beachten, dass δ (möglichst) klein gegenüber der Wellenlänge der einfallenden EM-Welle sein muss. Dabei muss die Wellenlänge im Medium herangezogen werden, wobei die Wellenlänge λ um den Faktor $\sqrt{\epsilon_r}$ gegenüber der Wellenlänge im leeren Raum verringert ist.

Für die im obigen Beispiel verwendeten Verhältnisse (f = 2.5 GHz) sind die Werte von ϵ_r aus Tabelle 2 zu entnehmen.

Folgende grobe Abschätzung gibt eine Vorstellung über den Zusammenhang:

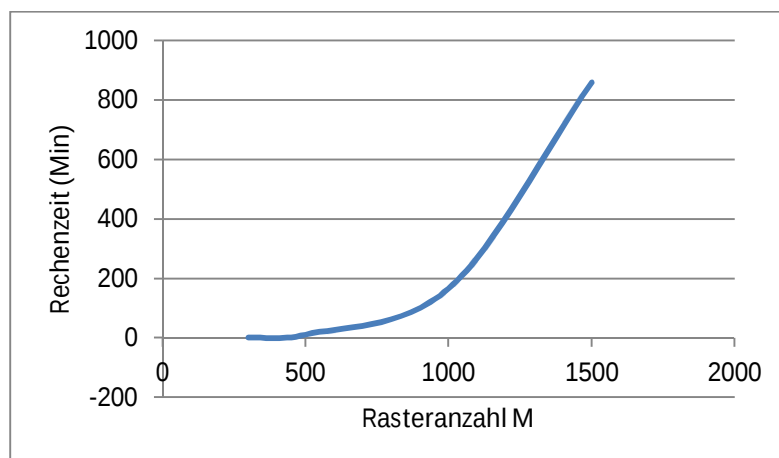
Für den „worst-case“ „reines Wasser“ ist $\epsilon_r \approx 81$. Daher ist der Verkürzungsfaktor der Wellenlänge $\sqrt{\epsilon_r} \approx 9$.

Für f = 2.5GHz beträgt die Wellenlänge im leeren Raum $\lambda_{2.5GHz} = \frac{c}{f} = 0.12m$. Im Wasser ist daher $\lambda_{H_2O} = 0.0133m = 13.3mm$. Dies ist (gerade noch) groß gegenüber der Voxel-Größe von 1.1mm !

Es ist also günstig, M möglichst groß zu wählen. Allerdings steigt die Rechenzeit (auch mit den optimierten Methoden von MATLAB) mit steigendem M rasant an.

Für M=1500 beträgt die Rechenzeit ca. 14 Stunden für eine Schicht-Ebene!

Eine eventuelle Verkürzung ist mit einer besseren Rechnerkonfiguration in gewissem Maß möglich.



M	Min
300	1.5
500	12
1000	167
1500	860

Berechnung einer Schicht
mit „SchichtenZubal.m“

Abb. 31: Anstieg der Rechenzeit in Abhängigkeit der Rasterzahl M mit MATLAB-Routinen

Reflexion und Transmission

Die obigen Beispiele zum Eindringen und Verhalten (Dämpfung, Reflexion) setzen Rand- und Anfangsbedingungen voraus. Mit diesen Bedingungen wird an definierten Orten eine gewisse Amplitude der Welle (Feldstärke) erzwungen.

Dies betrifft insbesondere die Ebene, bei der die (ebene) Welle in den zu berechnenden (Raum-)Abschnitt eindringt, Berandungsflächen und eventuelle Reflexionsflächen „hinter“ dem betrachteten Berechnungsbereich.

Üblicherweise wird die Elektromagnetische Welle durch einen „Sender“ erzeugt, wobei die Welle als freie Welle eine gewisse Distanz bis zum Auftreffen auf Materie durchläuft.

Die in der Welle enthaltene Energie dringt jedoch nicht vollständig in die Materie ein, sondern es tritt an der Eintrittsebene Reflexion und nur teilweise Transmission auf.

Eine quantitative Abschätzung soll im Folgenden gegeben werden.

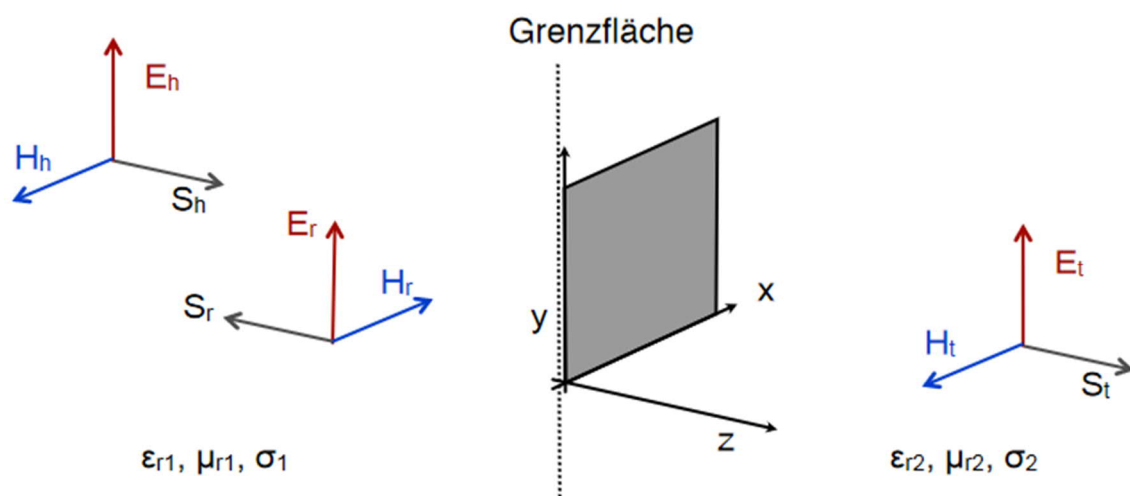


Abb. 32: Reflexion, Transmission beim Durchgang einer EM-Welle durch eine Grenzfläche (entnommen aus /49/)

Hinlaufende Welle	$\vec{E}_h \cdot \exp(-j \cdot \underline{k}_1 \cdot z)$
Rücklaufende (reflektierte) Welle	$\vec{E}_r \cdot \exp(+j \cdot \underline{k}_1 \cdot z)$
Eindringende (transmittierte) Welle	$\vec{E}_t \cdot \exp(-j \cdot \underline{k}_2 \cdot z)$
Wellenzahl $\underline{k}$	$\underline{k} = \omega \cdot \sqrt{\underline{\mu} \cdot \underline{\epsilon}}$
Wellenimpedanz $\underline{Z}$	$\underline{Z} = \sqrt{\frac{\underline{\mu}}{\underline{\epsilon}}}$
Komplexe Permittivität $\underline{\epsilon}$	$\underline{\epsilon} = \epsilon_0 \cdot \epsilon_r - \frac{j\sigma}{\omega}$
Reflexionsfaktor $\underline{\Gamma}$	$\vec{E}_r = \underline{\Gamma} \cdot \vec{E}_h \quad \underline{\Gamma} = \frac{\underline{Z}_2 - \underline{Z}_1}{\underline{Z}_2 + \underline{Z}_1}$ (beachte: $\underline{\Gamma}$ ist negativ)

Transmissionsfaktor $\underline{T}$

$$\vec{E}_t = \underline{T} \cdot \vec{E}_h \quad \underline{T} = 1 + \underline{\Gamma} = \frac{2 \cdot \underline{Z}_2}{\underline{Z}_2 + \underline{Z}_1}$$

Eindringtiefe

die Strecke, nach der die Amplitude der EM-Welle um den Faktor $1/e$ abgeklungen ist.

Zu beachten ist, dass beim Übergang vom leeren Raum (Luft) entsprechend dem Reflexionsfaktor Γ nur ein gewisser Anteil in die Materie eindringt.

Eine beispielhafte Abschätzung typischer Verhältnisse beim Eindringen in biologisches Gewebe zeigt Tabelle 4.

Die Haut bildet auch für EM-Wellen eine gewisse Schutzschicht. Die typische Dicke der Haut kann mit ca. 1.5mm angenommen werden.

Für den betrachteten Frequenzbereich gelangen nur etwa 30% der einfallenden Welle in den Körper. Die Eindringtiefe sinkt stark mit steigender Frequenz und somit wird ein Großteil der in der Welle enthaltenen Energie bereits in den äußersten Schichten absorbiert. Die Haut ist bestens mit Blutgefäßen versorgt, sodass eine eventuelle Erwärmung gut abgeleitet werden kann.

	Frequency [Hz]	Wavelength [m]	Relative permittivity	Conductivity [S/m]	Complex permittivity [As/Vm]	Penetration depth [m]	Wellenzahl k [1/m]	Wellenimpedanz Z [Ohm]	Reflexionsfaktor Γ	Transmissionsfaktor $\underline{T}$
Luft			1	0	8.85419E-12			376.730		
Haut 500MHz	500000000	0.083371	48.621	0.70444	4.305E-10-2.242E-10i	0.054198	75.364-18.451i	49.422+12.099i	-0.767+0.050i	0.233+0.050i
Haut 1GHz	1000000000	0.043708	45.711	0.88181	4.047E-10-1.403E-10i	0.041293	143.754-24.217i	53.409+8.997i	-0.751+0.037i	0.249+0.037i
Haut 2.5GHz	2500000000	0.018169	42.783	1.6223	3.788E-10-1.033E-10i	0.021598	345.830-46.299i	56.073+7.507i	-0.740+0.030i	0.260+0.030i
Haut 5GHz	5000000000	0.0094068	39.611	3.5744	3.507E-10-1.138E-10i	0.0094666	667.939-105.632i	57.663+9.119i	-0.734+0.036i	0.266+0.036i
Haut 10GHz	10000000000	0.0050416	33.528	8.951	2.969E-10-1.425E-10i	0.0035267	1246.250-283.548i	60.237+13.705i	-0.723+0.054i	0.277+0.054i
Haut 20GHz	20000000000	0.0029008	23.765	19.708	2.104E-10-1.568E-10i	0.0013919	2166.028-718.403i	65.680+21.784i	-0.699+0.084i	0.301+0.084i
Haut 30GHz	30000000000	0.0021815	17.743	27.521	1.571E-10-1.460E-10i	0.00088361	2880.125-1131.707i	71.243+27.994i	-0.675+0.105i	0.325+0.105i

Tabelle 4: typische Werte betreffend Reflexion/Transmission für die „Haut“ /46/

Thermisches Verhalten

Wenn eine elektromagnetische Welle auf Materie trifft, so treten Wechselwirkungen auf.

Die von der EM-Welle transportierte Energie E ist abhängig von der Frequenz. Für ein Photon gilt

$$E = h \cdot f$$

$$h = 6.626 \cdot 10^{-34} \text{ Js} = 4.136 \cdot 10^{-15} \text{ eVs} \quad (\text{Planck'sches Wirkungsquantum})$$

Typische Bindungsenergien zwischen Atomen liegen bei Molekülen zwischen 2 und 7 eV pro Bindung. Dementsprechend wird als (definierte) Grenze zwischen ionisierender und nicht-ionisierender Strahlung eine elektromagnetische Welle mit ca.

$$f = 750 \text{ TeraHertz (entspricht } \lambda = 400\text{nm)} \quad \text{festgelegt.}$$

Dies entspricht einer Energie von ca. $E = 3 \text{ eV}$.

Ionisierende Strahlung ist eine Bezeichnung für elektromagnetische Strahlung (oder auch Teilchenstrahlung), die in der Lage ist, Elektronen aus Atomen oder Molekülen zu entfernen (bei Teilchen durch Stoßprozesse), sodass positiv geladene Ionen oder Molekülreste zurückbleiben (Ionisation).

Die Energie von Wellen, deren Frequenz unterhalb der Ionisations-Frequenz liegt, führt bei Absorption zwar nicht zu Ionisation, jedoch zu Temperaturerhöhung.

Im Folgenden werden die Grundmechanismen der „dielektrischen Erwärmung“ /50/ durch elektromagnetische Wellen zusammenfassend besprochen. Ionisierende Wirkungen werden hier nicht behandelt.

Dielektrische Erwärmung ist der Prozess, bei dem ein elektrisches Hochfrequenzfeld oder elektromagnetische Radiowellen- bzw. Mikrowellenstrahlung ein dielektrisches Material erwärmt. Bei höheren Frequenzen wird diese Erwärmung durch molekulare Dipolrotation innerhalb des Dielektrikums verursacht. Molekulare Rotation tritt in Materialien auf, die polare Moleküle mit einem elektrischen Dipolmoment enthalten, mit der Folge, dass sie sich in einem elektromagnetischen Feld ausrichten und im Takte der Hochfrequenz eine oszillierende Drehbewegung ausführen. Dies wird als Dipolrotation oder dipolare Polarisation bezeichnet. Die Temperatur ist durch die durchschnittliche kinetische Energie (Bewegungsenergie) der Atome oder Moleküle in einem Material bestimmt. Wenn also die Moleküle zu Drehbewegungen angeregt werden, erhöht sich die Temperatur des Materials.

Die im privaten Bereich häufigste Anwendung der dielektrischen Erwärmung ist der „Mikrowellenofen“, der am effektivsten mit flüssigem Wasser und weniger effektiv mit Fetten und Zuckern arbeitet. Dies liegt daran, dass Fette und Zuckermoleküle weitaus weniger polar sind als Wassermoleküle und daher weniger von den Kräften beeinflusst werden, die durch die elektromagnetischen Wechselfelder erzeugt werden.

Im industriellen Bereich kann der Effekt im Allgemeinen zum Erhitzen von Feststoffen, Flüssigkeiten oder Gasen verwendet werden, vorausgesetzt, sie enthalten einige elektrische Dipole.

Für die dielektrische Erwärmung ist die erzeugte Leistungsdichte pro Volumen gegeben durch:

$$Q = \sigma \cdot E^2 = \omega \cdot \text{Imag}(\underline{\varepsilon}) \cdot E^2 \quad [\text{W} \cdot \text{m}^{-3}]$$

Der Imaginärteil der (frequenzabhängigen) Permittivität ist also ein Maß für die Fähigkeit eines dielektrischen Materials, elektromagnetische Feldenergie in Wärme umzuwandeln.

Wenn die Leitfähigkeit σ des Materials klein oder die Frequenz hoch ist, sodass $\sigma \ll \omega \cdot \varepsilon$, dann ist die dielektrische Erwärmung der dominierende Mechanismus des Energieverlusts des elektromagnetischen Felds an das Medium.

Die in Erwärmung resultierende Leistungsdichte Q ist natürlich ident mit der im vorigen Kapitel verwendeten Verlustleistungsdichte P_V . $Q \equiv P_V$

Mikrowellen durchdringen Materialien, deren Leitfähigkeit typisch für viele Lebensmittel ist. Das Eindringen hört im Wesentlichen dort auf, wo die gesamte eindringende Mikrowellenenergie in Wärme im Gewebe umgewandelt wurde.

Mikrowellenöfen zum Erhitzen von Lebensmitteln sind nicht auf die Frequenz für eine optimale Absorption durch Wasser eingestellt. Wenn dies der Fall wäre, würde das betreffende Stück Lebensmittel oder Flüssigkeit die gesamte Mikrowellenstrahlung in seiner äußeren Schicht absorbieren, was zu einem kühlen, nicht erhitzten Zentrum und einer überhitzten Oberfläche führen würde. Stattdessen lässt die gewählte Frequenz die Energie tiefer in das erhitzte Lebensmittel eindringen. Die Frequenz eines Haushaltsmikrowellenofens beträgt 2,45 GHz, während die Frequenz für eine optimale Absorption durch Wasser etwa 10 GHz beträgt.

Wie aus der „Wärmelehre“ bekannt /51/, /52/, gibt es drei Arten von Wärmetransportvorgängen:

- Wärmeleitung durch mechanischen Kontakt,
- Konvektion, das Mitführen thermischer Energie in einem strömenden Medium,
- Wärmestrahlung, also elektromagnetische Wellen.

Konvektion tritt nur in strömenden Gasen oder Flüssigkeiten auf und nicht in festen Dielektrika, die hier behandelt werden.

Auch die Wärme(ab)strahlung soll hier nicht besprochen werden, da sie im gegenwärtigen Zusammenhang eher unwesentlich ist. Daher bezieht sich das Folgende auf die „Wärmeleitung“.

Die allgemeine Wärmeleitungsgleichung /52/ lautet:

$$\frac{dT}{dt} = \frac{\lambda}{c \cdot \rho} \cdot \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right) = \frac{\lambda}{c \cdot \rho} \cdot \Delta T = a \cdot \Delta T \quad \Delta \dots \text{Laplace-Operator}$$

$c \dots$ spezifische Wärmekapazität [$\text{J} \cdot \text{kg}^{-1} \cdot \text{K}^{-1}$]

$\lambda \dots$ Wärmeleitvermögen oder Wärmeleitfähigkeit [$\text{W} \cdot \text{m}^{-1} \cdot \text{K}^{-1}$]

$\rho \dots$ Dichte [$\text{kg} \cdot \text{m}^{-3}$]

$a = \frac{\lambda}{c \cdot \rho} \dots$ Temperaturleitfähigkeit [m^2/s]

Bei zusätzlichem Vorliegen von Wärmequellen, bzw. Senken, gilt die inhomogene Wärmeleitungsgleichung:

$$\frac{dT}{dt} = \frac{\lambda}{c \cdot \rho} \cdot \nabla^2 T + f(x, t) \quad \frac{dT}{dt} = a \cdot \nabla^2 T + f(x, t)$$

$f(x,t)$: Quotient aus volumenbezogener Wärmequellendichte (der pro Volumen und Zeit produzierten Wärmemenge) und der volumenbezogenen Wärmekapazität (dem Produkt aus Dichte und massebezogener Wärmekapazität) $[f(x,t)] = K \cdot s^{-1}$

Bzw.:

$$c \cdot \rho \cdot \frac{dT}{dt} = \lambda \cdot \nabla^2 T + Q - Q_P \quad \begin{array}{l} Q, Q_P \dots \text{zu-/abgeführte} \\ \text{volumsbezogene Leistung} \end{array} \quad [W \cdot m^{-3}]$$

Z.B. könnte sein /53/:

$$Q_P \dots \text{Wärmeverlust durch Blutperfusion } [W \cdot m^{-3}] \quad Q_P = \rho_{bl} \cdot c_{bl} \cdot (T - T_{bl})$$

$c_{bl} \dots$ spezifische Wärmekapazität von Blut

$\rho_{bl} \dots$ Dichte von Blut

$T_{bl} \dots$ Bluttemperatur

$Q \dots$ Wärmeenergie durch externe Wärme, RF (radiofrequency), MW (microwave), Laser oder FUS-Energie (focused ultrasound) erzeugt $[W \cdot m^{-3}]$

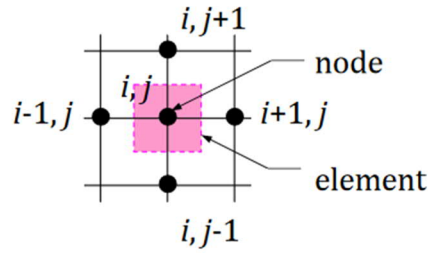
Die obige Differenzialgleichung kann nun entsprechend der Finite-Differenzen-Methode für eine numerische Berechnung aufbereitet werden /54/:

$$\frac{T_{n+1}(i,j) - T_n(i,j)}{\Delta \tau} = \frac{\lambda}{\rho \cdot c_P} \cdot \left(\frac{T_n(i,j+1) - 2 \cdot T_n(i,j) + T_n(i,j-1)}{\delta^2} + \frac{T_n(i+1,j) - 2 \cdot T_n(i,j) + T_n(i-1,j)}{\delta^2} \right) + \frac{Q_n(i,j)}{\rho \cdot c_P}$$

$$\text{mit} \quad S = \frac{\lambda}{\rho \cdot c_P \cdot \delta^2} = \frac{a}{\delta^2}$$

$$\begin{aligned} \frac{T_{n+1}(i,j) - T_n(i,j)}{\Delta \tau} &= s \cdot (T_n(i,j+1) - 2 \cdot T_n(i,j) + T_n(i,j-1)) + s \cdot (T_n(i+1,j) - 2 \cdot T_n(i,j) + T_n(i-1,j)) \\ &+ \frac{Q_n(i,j)}{\rho \cdot c_P} \end{aligned}$$

$T_n(i,j)$ i,j = location (node numbers) n = time (time step number)



$$\frac{T_{n+1}(i, j) - T_n(i, j)}{\Delta \tau} = s \cdot (T_n(i, j+1) - 4 \cdot T_n(i, j) + T_n(i, j-1) + T_n(i+1, j) + T_n(i-1, j)) + \frac{Q_n(i, j)}{\rho \cdot c_p}$$

Im stationären Zustand sind Zeitableitungen Null:

$$0 = a \cdot \frac{T_n(i, j+1) - 4 \cdot T_n(i, j) + T_n(i, j-1) + T_n(i+1, j) + T_n(i-1, j)}{\delta^2} + \frac{Q_n(i, j)}{\rho \cdot c_p}$$

$$0 = \lambda \cdot \frac{T_n(i, j+1) - 4 \cdot T_n(i, j) + T_n(i, j-1) + T_n(i+1, j) + T_n(i-1, j)}{\delta^2} + Q_n(i, j)$$

$$T_n(i, j) = \frac{1}{4} \cdot (T_n(i, j+1) + T_n(i, j-1) + T_n(i+1, j) + T_n(i-1, j) + \frac{Q_n(i, j) \cdot \delta^2}{\lambda})$$

Als Randbedingung wird die abhängige Variable auf dem Rand direkt vorgegeben ("Dirichlet Randbedingung").

Bekannt sind: $T(1, j)$, $T(i_{\max}, j)$ mit $1 \leq j \leq j_{\max}$ sowie $T(i, 1)$, $T(i, j_{\max})$ mit $1 \leq i \leq i_{\max}$.

Das DG-System lässt somit die inneren Werte $T(i, j)$ finden (mit $i = 2, 3, \dots, i_{\max}-1$ in x-Richtung bzw. $j = 2, 3, \dots, j_{\max}-1$ in y-Richtung).

5						
	T[1,5]	T[2,5]	T[3,5]	T[4,5]	T[5,5]	
4						
	T[1,4]	T[2,4]	T[3,4]	T[4,4]	T[5,4]	
3						
	T[1,3]	T[2,3]	T[3,3]	T[4,3]	T[5,3]	
2						
	T[1,2]	T[2,2]	T[3,2]	T[4,2]	T[5,2]	
1						
	T[1,1]	T[2,1]	T[3,1]	T[4,1]	T[5,1]	
0						
	0	1	2	3	4	5
	x-Achse (i=1 ... i _{max})					

$$T(i+1, j) + T(i-1, j) - 4 \cdot T(i, j) + T(i, j+1) + T(i, j-1) = \frac{Q(i, j) \cdot \delta^2}{\lambda} = \delta^2 \cdot f(i, j)$$

Durch die EM-Welle wird die Verlustleistungsdichte P_V wirksam $[\text{W/m}^3] \rightarrow Q(i, j)$

λ ... Wärmeleitvermögen oder Wärmeleitfähigkeit $[\text{W.m}^{-1}.\text{K}^{-1}]$

Damit kann eine Matrix-Beziehung aufgestellt werden (hier für $i_{\max} = j_{\max} = 5$):

$$\begin{array}{cccccccccccc}
 -4T_{22} + T_{23} & & & + T_{32} & & & & & & & & & T_{21} & T_{12} \\
 T_{22} - 4T_{23} + T_{24} & & & & + T_{33} & & & & & & & & T_{13} \\
 & T_{23} - 4T_{24} & & & & + T_{34} & & & & & & & T_{25} & T_{14} \\
 T_{22} & & & - 4T_{32} + T_{33} & & + T_{32} & & & & & & & T_{31} \\
 & + T_{23} & & + T_{32} - 4T_{33} + T_{34} & & & + T_{43} & & & & & & \\
 & & + T_{24} & & + T_{33} - 4T_{34} & & & + T_{44} & & & & & \\
 & & & + T_{32} & & & - 4T_{42} + T_{43} & & & & & & T_{35} \\
 & & & & + T_{33} & & + T_{42} - 4T_{43} + T_{44} & & & & & & T_{41} & T_{52} \\
 & & & & & + T_{34} & & + T_{43} - 4T_{44} & & & & & T_{53} \\
 & & & & & & & & & & & & T_{45} & T_{54}
 \end{array}$$

$$\begin{array}{cccccccccccc}
 i & j= & 2 & 3 & 4 & 2 & 3 & 4 & 2 & 3 & 4 \\
 2 & \parallel & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 2 & \parallel & 1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
 2 & \parallel & 0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 \\
 3 & \parallel & 1 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 \\
 3 & \parallel & 0 & 1 & 0 & 1 & 4 & 1 & 0 & 1 & 0 \\
 3 & \parallel & 0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 \\
 4 & \parallel & 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\
 4 & \parallel & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\
 4 & \parallel & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4
 \end{array}
 \parallel \begin{array}{c} T_{2,2} \\ T_{2,3} \\ T_{2,4} \\ T_{3,2} \\ T_{3,3} \\ T_{3,4} \\ T_{4,2} \\ T_{4,3} \\ T_{4,4} \end{array}
 + \parallel \begin{array}{c} T_{2,1}+T_{1,2} \\ T_{1,3} \\ T_{2,5}+T_{1,4} \\ T_{3,1} \\ 0 \\ T_{3,5} \\ T_{4,1}+T_{5,2} \\ T_{5,3} \\ T_{4,5}+T_{5,4} \end{array}
 = \parallel \begin{array}{c} \delta^2 \cdot f_{2,2} \\ \delta^2 \cdot f_{2,3} \\ \delta^2 \cdot f_{2,4} \\ \delta^2 \cdot f_{3,2} \\ \delta^2 \cdot f_{3,3} \\ \delta^2 \cdot f_{3,4} \\ \delta^2 \cdot f_{4,2} \\ \delta^2 \cdot f_{4,3} \\ \delta^2 \cdot f_{4,4} \end{array}$$

$$\parallel \begin{array}{cccccccccccc}
 -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 1 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 1 & 4 & 1 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 & 0
 \end{array}
 \parallel \begin{array}{c} T_{2,2} \\ T_{2,3} \\ T_{2,4} \\ T_{3,2} \\ T_{3,3} \\ T_{3,4} \\ T_{4,2} \\ T_{4,3} \\ T_{4,4} \end{array}
 = \parallel \begin{array}{c} \delta^2 \cdot f_{2,2} - T_{2,1} - T_{1,2} \\ \delta^2 \cdot f_{2,3} - T_{1,3} \\ \delta^2 \cdot f_{2,4} - T_{2,5} - T_{1,4} \\ \delta^2 \cdot f_{3,2} - T_{3,1} \\ \delta^2 \cdot f_{3,3} - 0 \\ \delta^2 \cdot f_{3,4} - T_{3,5} \\ \delta^2 \cdot f_{4,2} - T_{4,1} - T_{5,2} \\ \delta^2 \cdot f_{4,3} - T_{5,3} \\ \delta^2 \cdot f_{4,4} - T_{4,5} - T_{5,4} \end{array}$$

$$AT = B$$

A ... Koeffizientenmatrix

B ... Spaltenmatrix der inhomogenen Glieder (beinhaltet Randbedingungen)

T ... Ergebnis-Spaltenmatrix

i = 2	-4	1	0	1	0	0	0	0	0
j = 2	T _{i,j}	T _{i,j+1}		T _{i+1,j}					
i = 2	1	-4	1	0	1	0	0	0	0
j = 3	T _{i,j-1}	T _{i,j}	T _{i,j+1}		T _{i+1,j}				
i = 2	0	1	-4	0	0	1	0	0	0
j = 4		T _{i,j-1}	T _{i,j}			T _{i+1,j}			
i = 3	1	0	0	-4	1	0	1	0	0

$$-4T_{22} + T_{23} + T_{32}$$

$$T_{22} - 4T_{23} + T_{24} + T_{33}$$

$$T_{23} - 4T_{24} + T_{34}$$

$$T_{22} - 4T_{32} + T_{33} + T_{42}$$

T ₂₁ + T ₁₂
T _{i,j-1} T _{i-1,j}
T ₁₃
T _{i-1,j}
T ₂₅ + T ₁₄
T _{i,j+1} T _{i-1,j}
T ₃₁

j = 2	Ti-1,j			Ti,j	Ti,j+1		Ti+1,j		
i = 3	0	1	0	1	-4	1	0	1	0
j = 3		Ti-1,j		Ti,j-1	Ti,j	Ti,j+1		Ti+1,j	
i = 3	0	0	1	0	1	-4	0	0	1
j = 4			Ti-1,j		Ti,j-1	Ti,j			Ti+1,j
i = 4	0	0	0	1	0	0	-4	1	0
j = 2				Ti-1,j			Ti,j	Ti,j+1	
i = 4	0	0	0	0	1	0	1	-4	1
j = 3					Ti-1,j		Ti,j-1	Ti,j	Ti,j+1
i = 4	0	0	0	0	0	1	0	1	-4
j = 4						Ti-1,j		Ti,j-1	Ti,j

$$T_{23} + T_{32} - 4T_{33} + T_{34} + T_{43}$$

$$T_{24} + T_{33} - 4T_{34} + T_{44}$$

$$T_{32} - 4T_{42} + T_{43}$$

$$T_{33} + T_{42} - 4T_{43} + T_{44}$$

$$T_{34} + T_{43} - 4T_{44}$$

Ti,j-1	
T35	
Ti,j+1	
T41 + T52	
Ti,j-1	Ti+1,j
T53	
Ti+1,j	
T45 + T54	
Ti,j+1	Ti+1,j

Die Softwarelösung „HeatKreis.m“ (siehe Anhang) gibt ein Beispiel der Realisierung.

Folgende Parameter werden angenommen:

Das MxM-Feld habe eine Wärmeleitfähigkeit $\lambda_{Air} = 0.026 \text{ Wm}^{-1}\text{K}^{-1}$ und einen Wärmeverlust $Q_{Air} = 0.05 \text{ Wm}^{-3}$.

Ein Kreis-förmiger Bereich mit einer Wärmeleitfähigkeit $\lambda_{Kreis} = 0.5 \text{ Wm}^{-1}\text{K}^{-1}$ und einer Wärmezufuhr $Q_{Kreis} = -100 \text{ Wm}^{-3}$ wird in das MxM-Feld eingebettet.

Als Randbedingung wird die Temperatur an den Grenzen des MxM-Feldes auf $T_{Rand} = 20^\circ\text{C}$ gehalten.

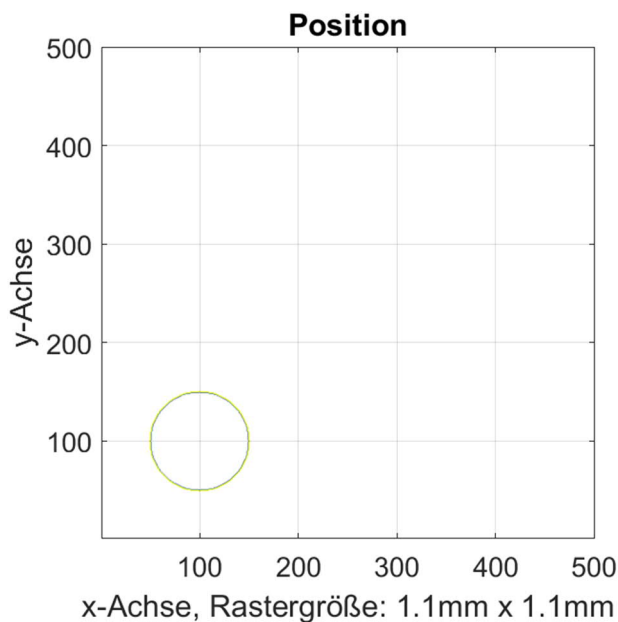


Abb. 33a: Position des kreisförmigen Bereichs

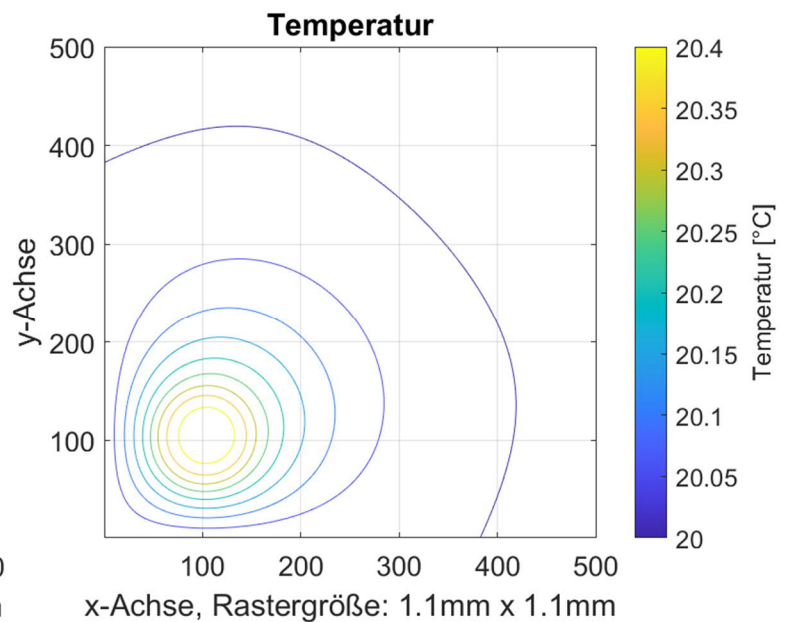


Abb. 33b: Temperaturverteilung

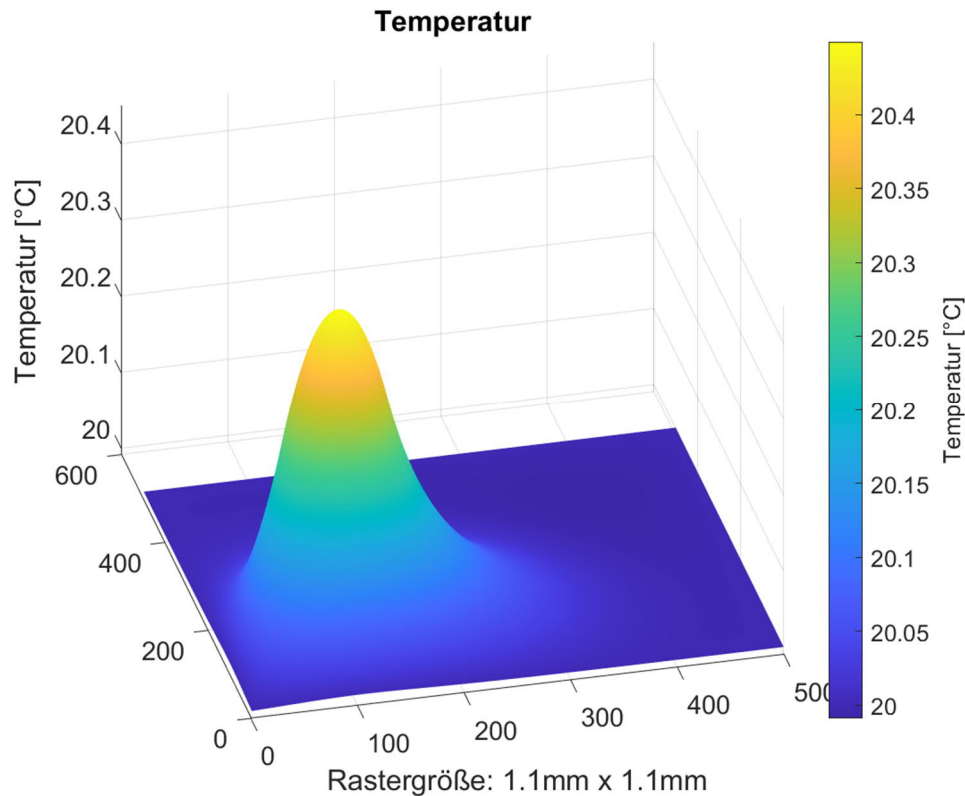


Abb. 33c: Temperaturverteilung

Die Rechenzeit ist vernachlässigbar kurz:

Eingabe von M: 500

M = 500

Startzeit: 25-Mar-2021 20:37:47

Voxel = 0.0011 m

Zwischenzeit: 25-Mar-2021 20:37:47 (sparse Matrix A ist angelegt)

Endzeit: 25-Mar-2021 20:37:48

Biologische Wirkungen

Dass elektromagnetische Wellen Auswirkungen auf biologisches Material haben, ist unbestritten. Die Wirkungen können durchaus dramatisch und auch tödlich sein.

Es gibt zahlreiche Untersuchungen, die sich mit dieser Problematik beschäftigen.

Allerdings sind viele Publikationen „pseudo-wissenschaftlich“ und müssen daher kritisch beurteilt werden. Speziell seit der allgemeinen Nutzung von Funk-Telefonen (Smartphones, „Handys“) sind viele „Meinungen“, subjektive Wahrnehmungen, Befürchtungen, etc. in den Medien zu finden, die eher den Charakter von Verschwörungstheorien aufweisen, ohne objektiven wissenschaftlichen Background.

Andererseits gibt es auch seriöse Untersuchungen von naturwissenschaftlich orientierten Instituten (Universitäten, staatliche Prüfanstalten, Behörden) die über die Gefährlichkeit oder auch Unbedenklichkeit von elektromagnetischen Wellen gute Richtlinien geben.

Exemplarisch dafür können z.B. die Informationen des Bundesamts für Strahlenschutz (BfS) der Bundesrepublik Deutschland /53/ genannt werden.

Basierend auf objektiven Beurteilungen werden rechtlich verbindliche Grenzwerte in Normen festgelegt. Für Österreich ist der „OVE Österreichischer Verband für Elektrotechnik“ die maßgebliche Stelle, die entsprechende Publikation ist die „OVE Richtlinie R 23-1“ /56/.

Zusammenfassend ist in Abb. 34 der als unbedenklich eingestufte elektrische Feldstärkenverlauf (in Abhängigkeit von der Frequenz) ersichtlich.

Abb. 35 zeigt den Grenzwert der Leistungsdichte, der mit 10 W m^{-2} angegeben ist.

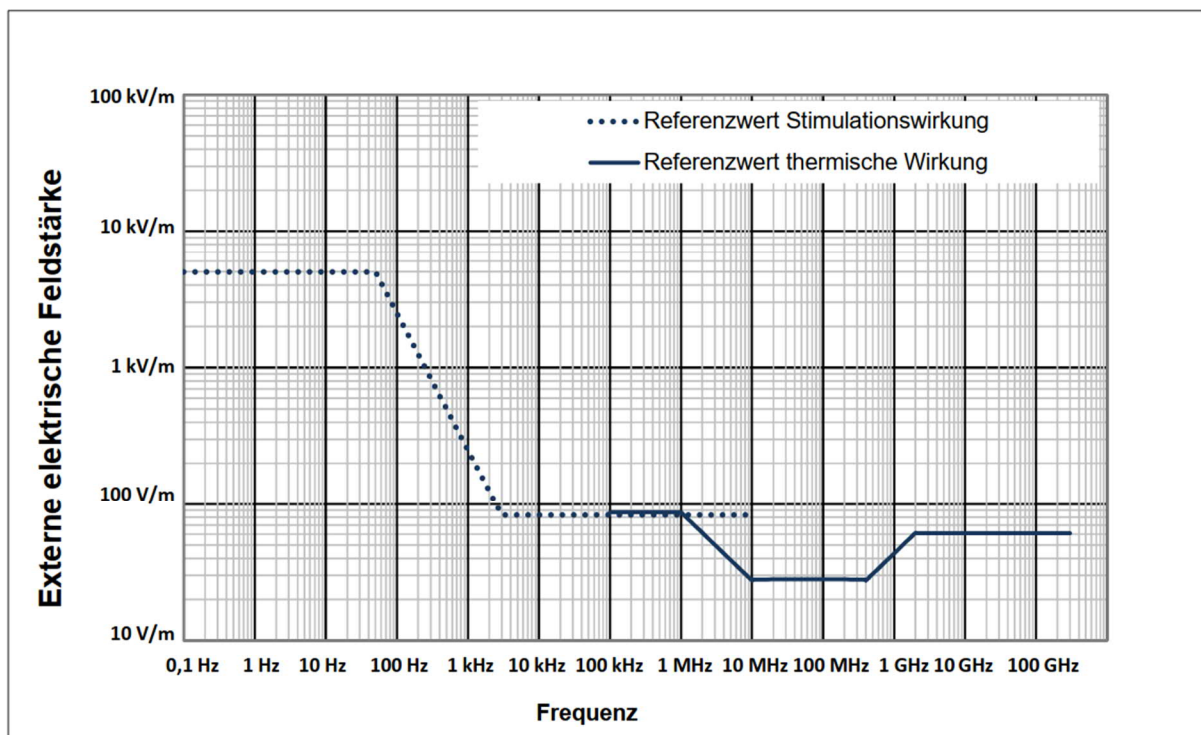


Abb. 34: Referenzwerte für Expositionen gegenüber zeitlich veränderlichen sinusförmigen elektrischen Feldstärken (entnommen aus /55/)

Die OVE Richtlinie nimmt auch Bezug auf die Tatsache, dass bei eher niedrigen Frequenzen Stimulationen von Nerven- bzw. Muskeln ausgelöst werden können, während bei höheren Frequenzen die thermische Wirkung entscheidend ist (siehe „Referenzwertverlauf Stimulationswirkung / thermische Wirkung“ in Abb.34).

Frequenzbereich	Leistungsdichte
	Wm^{-2}
$10 \text{ GHz} \leq f \leq 300 \text{ GHz}$	10
(a) Leistungsdichten werden über eine beliebige Fläche von 20 cm^2 gemittelt, wobei der Mittelwert über 1 cm^2 das 20-fache der angegebenen Grenzwerte nicht überschreiten darf. (b) Die Leistungsdichte wird über ein zeitliches Intervall von $(68/f^{1,05})$ Minuten gemittelt, wobei f in GHz eingesetzt wird. In keiner Position des Zeitfensters darf eine Überschreitung auftreten. (c) Im Fall der gleichzeitigen Exposition gegenüber Feldern verschiedener Frequenz ist nach Abschnitt 8.2 vorzugehen.	

Abb. 35: Basisgrenzwerte für Leistungsdichten bei Frequenzen von 10 GHz bis 300 GHz (entnommen aus /55/)

Generell kann festgestellt werden, dass im täglichen Normalfall diese Grenzwerte beim „Normalbürger“ durch keine technische Anwendung auch nur annähernd erreicht werden!

Fakten betreffend eventuelle Gefährdung durch Mobilfunk-Telefonie

Da es oft Diskussionen betreffend eventueller Gesundheitsgefährdung in Zusammenhang mit Elektromagnetischen Wellen, ausgesendet von Mobilfunkbasisstationen bzw. vom „Handy“ (Smartphone) selbst gibt, sollen hier einige Fakten festgestellt werden /57/, /59/:

- Sendeleistung: Mobilfunkbasisstationen (Sende- und Empfangsanlagen) haben Sendeleistungen zwischen 10 und 50 Watt.
Die Sendeleistung von SmartPhones beträgt maximal 2 Watt.
Im Vergleich dazu haben Hörfunk- und TV-Sender eine Sendeleistung bis zu mehreren 100.000 Watt.
- Frequenzbereiche: Für das 4G- sowie auch für 5G-Netz werden diverse Frequenzbänder im Bereich 700MHz bis (derzeit) 3.8GHz verwendet. Erst bei zukünftigen Frequenzvergaben erfolgt eine Erweiterung durch das 26GHz-Band.
Hierzu gibt es von der RTR (Rundfunk und Telekom Regulierungs-GmbH) noch keinen Terminplan.
- Zur Begrenzung von (eventuellen) Auswirkungen von EM-Wellen auf den Organismus wird der SAR-Wert (spezifische Absorptionsrate) als gesetzlich vorgegebener Grenzwert verwendet (siehe unten).

Spezifische Absorptionsrate SAR

Die spezifische Absorptionsrate ist ein Maß für die Absorption von elektromagnetischen Feldern in einem Material und wird als Leistung pro Masse in der Einheit W/kg ausgedrückt /58/.

$$SAR = \frac{1}{2} \cdot \frac{\sigma \cdot |\vec{E}|^2}{\rho}$$

ρ Dichte des Gewebes [kg/m³]

σ Elektrische Leitfähigkeit [S/m]

Für Mobilfunktelefone gibt es gesetzlich festgelegte Grenzwerte.

Der SAR-Grenzwert für Felder von Mobilfunkbasisstationen beträgt 0,08 Watt/kg für die allgemeine Bevölkerung. Dieser Wert ist über den gesamten Körper gemittelt. Er stellt sicher, dass die mögliche Temperaturerhöhung des ganzen Körpers in der Nähe von Mobilfunkbasisstationen unter 0,02 °C liegt.

Für die elektromagnetischen Felder, die beim Gebrauch des Handys in der Höhe des Kopfes entstehen, gilt der Teilkörpergrenzwert. Er beträgt 2 Watt/kg und ist über 10 g Körpergewebe gemittelt. Er garantiert, dass die örtliche Temperaturerhöhung, die beim Gebrauch eines Handys in Teilen des Körpers entsteht, geringer ist als 0,1 °C /57/.

Die in den Spezifikationen angegebenen SAR-Werte von Handys beziehen sich immer auf die maximal mögliche Sendeleistung. Diese wird aber nur unter den ungünstigsten Sendebedingungen erreicht. Sie reduziert sich beispielsweise, je geringer die Entfernung zwischen Handy und Mobilfunkantenne ist.

Die Bestimmung des SAR-Wertes erfolgt messtechnisch mit Hilfe eines Phantommodelles, welches den dielektrischen Eigenschaften des Körpers nachgebildet ist.

Abschätzung des elektrischen Feldstärkeverlaufs einer Sendestation

Die Aussendung und der Empfang von Elektromagnetischen Wellen erfolgt meist durch Dipolantennen in verschiedensten Ausführungsformen. Die theoretische Beschreibung der Feldverläufe ist aufwendig und kompliziert, aber weitgehend für idealisierte Bedingungen (Hertz'scher Dipol) gelöst. Eine kompakte Zusammenfassung ist z.B. in /60/ nachlesbar.

Für eine praktische, rasche Abschätzung finden sich in diversen Publikationen zu Antennen einfache Zahlenwertgleichungen. Z.B. in /61/:

$$E = \frac{\sqrt{30 \cdot P_S \cdot G}}{r} \quad \text{Feldstärke im „Fernfeld“, d.h. für } r \gg \lambda.$$

E ... elektrische Feldstärke [V/m]

P_S ... Sendeleistung [W]

G ... Antennengewinn [dB] r ... Entfernung vom Sender λ ... Wellenlänge ($\lambda = \frac{c}{f}$)

Der Antennengewinn (G) ist eine relative Größe, die sich auf eine Referenzantenne (meist Hertz'scher Dipol) bezieht.

Die Umrechnung zum absoluten Zahlenwert ergibt sich zu $g = 10 \cdot \log(G) [dB]$

Schlussbemerkungen

Die besprochenen Wirkungen von Elektromagnetischen Wellen zeigen auf, dass die dazugehörigen mathematischen Behandlungen durchgehend schlüssig, konsistent und nachvollziehbar sind.

Das Grundwerkzeug dazu sind die, von J.C. Maxwell aufgestellten, Beziehungen zwischen Elektrischem und Magnetischem Feld.

Numerische Rechenmethoden zur computerisierten Berechnung sind durch allgemein verfügbare Rechentools mit vernünftig vertretbarem Aufwand gut anwendbar, auch wenn die notwendige Rechenzeit und der Speicherplatzbedarf durchaus beträchtlich sein kann. Die vorliegende Arbeit kann vielleicht dem Leser als Grundstock für weiterführende mathematische Untersuchungen dienen!

Literaturverzeichnis

- /1/ H.G. Stöhr: Untersuchungen zur Optimierung einer drahtlosen Energieübertragung mittels elektromagnetischer Felder für medizinische Anwendungen. Dissertation, TU-Wien, eingereicht April 1980
- /2/ DECSYSTEM-20, <https://en.wikipedia.org/wiki/DECSYSTEM-20>, abgerufen 11.3.2021
- /3/ James Clerk Maxwell: A Dynamical Theory of the Electromagnetic Field. In: Royal Society Transactions, 155, 1865, S. 459–512, eingereicht 1864. <http://www.bem.fi/library/1865-001.pdf>, abgerufen 11.3.2021
- /4/ Albert Einstein: Autobiographisches in Paul Arthur Schilpp (Hrsg.): Albert Einstein als Philosoph und Naturforscher. Vieweg, Braunschweig 1983, ISBN 978-3-528-08538-4
<https://link.springer.com/content/pdf/10.1007%2F978-3-322-88795-5.pdf>, abgerufen 11.3.2021
- /5/ Einstein A.: Maxwell's influence on the development of conception of physical reality. In James Clerk Maxwell—a commemorative volume 1831–1931 (eds JJ Thomson et al.), pp. 66–73. Cambridge, UK: Cambridge University Press.
- /6/ Ivan Tolstoy: James Clerk Maxwell: A Biography by Ivan Tolstoy (1982-08-01); Published by Canongate, Edinburgh (1981). ISBN 978-0862410100
- /7/ Adel Moussa: Die Maxwellgleichungen: „War es ein Gott, der diese Zeilen schrieb?“-Notizen zur gedanklichen Genese der Maxwellgleichungen. Didaktik der Physik Frühjahrstagung Hannover 2010
<http://phydid.physik.fu-berlin.de/index.php/phydid-b/article/download/146/155>, abgerufen 11.3.2021
- /8/ Ian Stewart: Seventeen Equations that Changed the World, Profile Books; Main Edition (13. Juni 2013), ISBN 978-1846685323
- /9/ M. Longair: '...a paper ...I hold to be great guns': a commentary on Maxwell (1865) 'A dynamical theory of the electromagnetic field'. Phil. Trans. R. Soc. A 373: 20140473.
- /10/ L. Bergmann, C. Schaefer: Lehrbuch der Experimentalphysik (Band 2 - Elektromagnetismus) 9th ed. - de Gruyter, 2006
- /11/ Wellengleichung, <https://de.wikipedia.org/wiki/Wellengleichung> abgerufen 12.3.2021
- /12/ Licht, <https://de.wikipedia.org/wiki/Licht>, abgerufen 12.3.2021
- /13/ Internationales Einheitensystem, https://de.wikipedia.org/wiki/Internationales_Einheitensystem abgerufen 12.3.2021
- /14/ DM Siegel: Maxwell's contributions to electricity and magnetism. In James Clerk Maxwell: perspectives on his life and work (eds R Flood, M McCartney, A Whitaker), pp. 187–203. Oxford, 2014, UK: Oxford University Press.
- /15/ Gerhard W. Bruhn: Die Maxwell-Gleichungen – vom Original zur modernen Schreibweise. Technische Universität Darmstadt.
<http://www2.mathematik.tu-darmstadt.de/~bruhn/Original-MAXWELL.htm>, abgerufen 13.3.2021
- /16/ André Waser: Über Schreibweisen der Feldgleichungen zur Elektrodynamik. © (2000) by André Waser; www.andre-waser.ch, abgerufen 13.3.2021

/17/ Peter Puschnig: Vektoranalysis, Vorlesungsskriptum SS 2015, Institut für Physik, Fachbereich Theoretische Physik, <https://docplayer.org/18895887-Vektoranalysis-phy-e10.html> abgerufen 13.3.2021

/18/ Stephan Mueller: Maxwellgleichungen Einführung und Erklärungen, https://www.youtube.com/watch?v=IXBCy5grxdU&ab_channel=StephanMueller, abgerufen 13.3.2021

/19/ Horst Frank / Phrood / Anony - Horst Frank, Jailbird and Phrood, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=3726606>, abgerufen 14.3.2021

/20/ TU-Dresden, Institut für Photogrammetrie und Fernerkundung, <https://tu-dresden.de/bu/umwelt/geo/ipf/fern/studium/tutorial/atmosphaerische-korrekturen/folder-2008-04-07-5554546418/2008-10-28.002003101>, abgerufen 26.6.2021

/21/ Gerhard Schiffner: Optische Nachrichtentechnik, B. G. Teubner Verlag, 2005, ISBN: 978-3-519-00446-2 <https://link.springer.com/content/pdf/10.1007%2F978-3-322-80061-9.pdf>, abgerufen 14.3.2021

/22/ Alfred Weigert, Heinrich J. Wendker, Lutz Wisotzki: Astronomie und Astrophysik, WILEY-VCH Verlag, 2010, ISBN: 978-3-527-40793-4

/23/ <https://de.mathworks.com/>, abrufen 15.3.2021

/24/ Felix Lucka, et al.: Skript zum Kompaktkurs „Einführung in die Programmierung mit MATLAB“, <https://www.uni-muenster.de/AMM/num/Vorlesungen/wuebbeling/matlab.pdf>, abgerufen 15.3.2021

/25/ Roger F. Harrington: Cylindrical Wave Functions, Wiley-IEEE Press ISBN:9780471208068
Download von IEEE Xplore Digital Library, <https://ieeexplore.ieee.org/document/5264722>,
abgerufen 15.3.2021

/26/ Hans Joachim Oberle: Skript zur Vorlesung Differentialgleichungen für Studierende der Ingenieurwissenschaften, Universität Hamburg, Fachbereich Mathematik, <https://www.math.uni-hamburg.de/home/oberle/skripte/diffgl/dgl2-08.pdf>, abgerufen 15.3.2021

/27/ Bessel'sche Differentialgleichung, https://de.wikipedia.org/wiki/Besselsche_Differentialgleichung,
abgerufen 15.3.2021

/28/ Dokumentation MATLAB: besselj, besselh <https://de.mathworks.com/help/matlab/ref/besselj.html>,
<https://de.mathworks.com/help/matlab/ref/besselh.html>, abgerufen 16.3.2021

/29/ Finite-Differenzen-Methode, <https://de.wikipedia.org/wiki/Finite-Differenzen-Methode>,
abgerufen 17.3.2021

/30/ Klaus Betzler: Numerische Lösung Partieller Differentialgleichungen, <https://www.betzler.physik.uni-osnabrueck.de/Manuskripte/short/pdenum.pdf>, abgerufen 17.3.2021

/31/ Lineares Gleichungssystem
https://de.wikipedia.org/wiki/Lineares_Gleichungssystem, abgerufen 18.3.2021

- /32/ R.S. MARTIN and J.H. WILKINSON: Symmetric Decomposition of Positive Definite Band Matrices, Numerische Mathematik 7, 355-361 (1965), Handbook Series Linear Algebra, <https://fddocuments.in/document/symmetric-decomposition-of-positive-definite-band-matrices.html>, abgerufen 18.3.2021
- /33/ Lars Grüne, Oliver Junge: Gewöhnliche Differentialgleichungen; Eine Einführung aus der Perspektive der dynamischen Systeme. Springer Spektrum. Springer Fachmedien Wiesbaden 2009, 2016, ISBN 978-3-658-10240-1
- /34/ Python Software Foundation, https://de.wikipedia.org/wiki/Python_Software_Foundation, abgerufen 20.3.2021
- /35/ Bernd Klein: Numerisches Python, Arbeiten mit NumPy, Matplotlib und Pandas, ISBN: 978-3-446-45076-9
- /36/ Dünnbesetzte Matrix, https://de.wikipedia.org/wiki/D%C3%BCnnbesetzte_Matrix, abgerufen 20.3.2021
- /37/ Jürgen Dankert, Helga Dankert: Technische Mechanik, Springer, 2013, ISBN 978-3-8348-2235-2, <https://www.springer.com/gp/book/9783834818096>, http://www.tm-mathe.de/Themen/html/matlab_zauberstab_backslash-.html, abgerufen 20.3.2021
- /38/ Andreas Meister: Numerik linearer Gleichungssysteme, Springer Vieweg, ISBN 978-3-8348-9446-5, <https://www.springer.com/de/book/9783834894465>, <https://link.springer.com/content/pdf/bbm%3A978-3-8348-9446-5%2F1.pdf>, abgerufen 21.3.2021
- /39/ Band matrix, https://en.wikipedia.org/wiki/Band_matrix, abgerufen 21.3.2021
- /40/ GNU Octave, https://de.wikipedia.org/wiki/GNU_Octave, abgerufen 21.3.2021
- /41/ IT'IS Foundation, <https://itis.swiss/news-events/news/latest-news/>, abgerufen 22.3.2021
- /42/ Human Models, Information Technologies in Society (IT'IS), <https://itis.swiss/virtual-population/virtual-population/overview/>, abgerufen 22.3.2021
- /43/ Tissue Properties, Information Technologies in Society (IT'IS), <https://itis.swiss/virtual-population/tissue-properties/database/dielectric-properties/>, abgerufen 22.3.2021
- /44/ Voxel: <https://de.wikipedia.org/wiki/Voxel>, abgerufen 22.3.2021
- /45/ PHM Repository, <https://itis.swiss/virtual-population/regional-human-models/phm-repository/v1-0/>, abgerufen 22.3.2021
- /46/ Dielectric Properties of Body Tissues, Institute for Applied Physics, ITALIAN NATIONAL RESEARCH COUNCIL, <http://niremf.ifac.cnr.it/tissprop/>, <http://niremf.ifac.cnr.it/tissprop/htmlclie/htmlclie.php>, abgerufen 22.3.2021
- /47/ Zubal Phantom site, <http://noodle.med.yale.edu/zubal>, abgerufen 22.3.2021

/48/ Zubal, I.G., Harrell, C.R, Smith, E.O, Rattner, Z., Gindi, G. and Hoffer, P.B., Computerized three-dimensional segmented human anatomy, Medical Physics, 21(2):299-302, 1994.

/49/ Wellen und Leitungen, Vorlesung und Übungen, Duale Hochschule Baden-Württemberg, [http://bachelor.srupp.de/Wellen und Leitungen T2ELN2001_1_SR.pdf](http://bachelor.srupp.de/Wellen_und_Leitungen_T2ELN2001_1_SR.pdf), abgerufen 17.6.2021

/50/ Dielectric heating, [https://en.wikipedia.org/wiki/Dielectric heating](https://en.wikipedia.org/wiki/Dielectric_heating), abgerufen 24.3.2021

/51/ L. Bergmann, C. Schaefer: Lehrbuch der Experimentalphysik (Band 1 – Mechanik, Akustik, Wärme) 9th ed. - de Gruyter, 2006

/52/ Wolfgang Demtröder: Experimentalphysik 1, Mechanik und Wärme, Springer, ISBN 978-3-540-79294-9

/53/ Harry H. Pennes: Analysis of Tissue and Arterial Blood Temperatures in the Resting Human Forearm, Journal of Applied Physiology, Vol. 1, Number 2, Aug 1948, <https://journals.physiology.org/doi/pdf/10.1152/jappl.1948.1.2.93>, abgerufen 25.3.2021

/54/ Two-Dimensional Conduction: Finite-Difference Equations and Solutions, <https://www.visualslope.com/Library/FDM-for-heat-transferring.pdf>, abgerufen 25.3.2021

/55/ Bundesamt für Strahlenschutz (BfS), Bundesrepublik Deutschland, https://www.bfs.de/DE/themen/emf/hff/wirkung/hff-nachgewiesen/hff-nachgewiesen_node.html, abgerufen 25.3.2021

/56/ OVE Österreichischer Verband für Elektrotechnik: Elektrische, magnetische und elektromagnetische Felder im Frequenzbereich von 0 Hz bis 300 GHz, OVE Richtlinie R 23-1, Ausgabe: 2017-04-01

/57/ informationszentrum-mobilfunk.de: Was bezeichnet der SAR-Wert bei Mobiltelefonen und Basisstationen?, <https://www.informationszentrum-mobilfunk.de/gesundheits/elektromagnetische-felder/spezifische-absorptionsrate-sar>, abgerufen 17.6.2021

/58/ Spezifische Absorptionsrate, [https://de.wikipedia.org/wiki/Spezifische Absorptionsrate](https://de.wikipedia.org/wiki/Spezifische_Absorptionsrate), abgerufen 17.6.2021

/59/ Rundfunk und Telekom Regulierungs-GmbH, Österreich, https://www.rtr.at/TKP/was_wir_tun/telekommunikation/spectrum/bands/FRQ_spectrum.de.html abgerufen 17.6.2021

/60/ Reconfigurable Antenna Laboratory, <https://www.waves.utoronto.ca/prof/svhum/ece422/notes/> abgerufen 20.11.2021

/61/ A.H. Systems, inc., <https://www.ahsystems.com/EMC-formulas-equations/field-intensity-calculation.php> abgerufen 20.11.2021

Anhang

SchichtenReflexion.m

```
clear all;          % clear variables window
clc;               % clear command window display
close all;         % close all figures
frequ = 2.0e9;
omega = 2.0 * pi * frequ;
E0 = 1;
d1 = 0.2;
d2 = 0.3;
epsr1 = 5.0;
sigma1 = 0.03;
epsr2 = 50.0;
sigma2 = 0.1;
eps0 = 1.0 / (36.0e9 * pi);
my0 = 4.0 * 1e-7 * pi;
wlang = 1.0/(sqrt(eps0*my0)*frequ);
gamma0 = omega * sqrt(eps0 * my0);
gamma1 = sqrt(epsr1 * eps0 * my0 * omega * omega - j * omega * my0 * sigma1);
gamma2 = sqrt(epsr2 * eps0 * my0 * omega * omega - j * omega * my0 * sigma2);
h1 = exp(-1j*gamma1*d1);
h2 = exp(1j*gamma1*d1);
h3 = exp(-1j*gamma2*d1);
h4 = exp(1j*gamma2*d1);
h5 = exp(-1j*gamma2*(d1+d2));
h6 = exp(1j*gamma2*(d1+d2));
h7 = gamma1 * h1;
h8 = gamma1 * h2;
h9 = gamma2 * h3;
h10 = gamma2 * h4;
a = [-1 1 1 0 0; gamma0 gamma1 -gamma1 0 0; 0 h1 h2 -h3 -h4; 0 -h7 h8 h9 -h10; 0 0 0 h5 h6];
b = [1; gamma0; 0; 0; 0];
x = a \ b;
A = x(1);
B = x(2);
C = x(3);
D = x(4);
F = x(5);
fprintf('\nd1 = %.3f m      d2 = %.3f m', d1, d2);
fprintf('\nf = %iHz\n', frequ);
figure;
x = -0.5:0.5/200:0;
y = abs(E0*exp(-j*gamma0*x)+A*exp(j*gamma0*x));
hold on;
plot(x,y,'LineWidth',2);
x = 0:d1/200:d1;
y = abs(B*exp(-j*gamma1*x)+C*exp(j*gamma1*x));
plot(x,y,'LineWidth',2);
x = d1:(d1+d2)/500:(d1+d2);
y = abs(D*exp(-j*gamma2*x)+F*exp(j*gamma2*x));
plot(x,y,'LineWidth',2);
title('E-Feldstärke');
ylabel('relative units');
xlabel('[m]');
ax = gca;
ax.FontSize = 14;
```

Zylinder.m

```
clear all;          % clear variables window
clc;               % clear command window display
close all;         % close all figures
frequ = 2.0e9;
omega = 2.0 * pi * frequ;
E0 = 1;
R = 0.2;          % Zylinderradius in Meter
M = 299;          % entspricht +/-M Rasterpunkte im einen 1Meter x 1Meter Feld
N = 30;           % Summenzaehler: von -N bis +N
epsr1 = 5;
sigma1 = 0.03;
eps0 = 1.0 / (36.0e9 * pi);
my0 = 4.0 * 1e-7 * pi;
wlang = 1.0/(sqrt(eps0*my0)*frequ);
```

```

gamma0 = omega * sqrt(eps0 * my0);
gamma1 = sqrt(epsr1 * eps0 * my0 * omega * omega - j * omega * my0 * sigma1);
% gamma1 = gamma0; % zum Test
fprintf("\nStartzeit: %s", datetime('now'));
An = zeros(2*N+1,1); % Koeffizienten berechnen (fuer n = -N bis +N)
Bn = zeros(2*N+1,1);
for n=-N:N
    An(n+N+1) = an(n, gamma0, gamma1, R);
    Bn(n+N+1) = bn(n, gamma0, gamma1, R);
end
MxM = zeros(M+1);
for I=1:M+1
    for J=1:M+1
        rho = sqrt((I-M/2-1)*(I-M/2-1)+(J-M/2-1)*(J-M/2-1)) / M;
        phi = atan((I-M/2-1)/(J-M/2-1));
        if (rho > R)
            MxM(I,J) = 0;
            for n=-N:N
                MxM(I,J)=MxM(I,J)+((1j.^(-
n))*(besselj(n,gamma0*rho)+An(n+N+1)*besselh(n,2,gamma0*rho,1))*exp(1j*n*phi));
            end
            MxM(I,J) = real(MxM(I,J));
        else
            MxM(I,J) = 0;
            for n=-N:N
                MxM(I,J)=MxM(I,J)+((1j.^(-n))*Bn(n+N+1)*besselj(n,gamma1*rho)*exp(1j*n*phi));
            end
            MxM(I,J) = real(MxM(I,J));
        end
    end
end
end
figure;
surf(MxM);
title('E-Feldstärke');
grid on;
axis('square');
xlabel('x-Achse, 300 Rasterpunkte entsprechen 1m');
ylabel('y-Achse, 300 Rasterpunkte entsprechen 1m');
zlabel('relative units')
ax = gca;
ax.FontSize = 14;
fprintf("\nEndzeit: %s\n", datetime('now'));
memory
function result = an(n, gamma0, gammaz, R)
    result = gammaz*besselj(n,gamma0*R)*dbesselj(n,gammaz*R) -
gamma0*dbesselj(n,gamma0*R)*besselj(n,gammaz*R);
    result = result / (gamma0*dbesselh2(n,gamma0*R)*besselj(n,gammaz*R) -
gammaz*besselh(n,2,gamma0*R,1)*dbesselj(n,gammaz*R));
end
function result = bn(n, gamma0, gammaz, R)
    result = (besselj(n,gamma0*R) + an(n, gamma0, gammaz,
R)*besselh(n,2,gamma0*R))/besselj(n,gammaz*R);
end
function result = dbesselj(n, x)
    result = 0.5 * (besselj(n-1,x) - besselj(n+1,x));
end
function result = dbesselh2(n, x)
    result = 0.5 * (besselh(n-1,2,x,1) - besselh(n+1,2,x,1));
end

```

BandMatrix.m

```

% Lösung "symmetric-decomposition-of-positive-definite-band-matrice"
clear all; % clear variables window
clc; % clear command window display
close all; % close all figures
FAKTOR=input('\n\nEingabe Rasteranzahl als Vielfaches von 50: ');
M = FAKTOR * 50;
N = M * M;
F = 1.0e9;
fprintf("Frequenz: %.0f Hz\n", F);
EPSR1 = 5.0;
sigma1 = 0.03;
EPSR2 = 30.0;
sigma2 = 0.3;
EPSR3 = 80.0;
sigma3 = 1.00;

```

```

EPS0 = 1.0 / (36.0e9 * pi);
EPS1 = EPS0*EPSR1 + sigma1/(2j*pi*F);
EPS2 = EPS0*EPSR2 + sigma2/(2j*pi*F);
EPS3 = EPS0*EPSR3 + sigma3/(2j*pi*F);
%EPS1 = EPS0;
%EPS2 = EPS0;
%EPS3 = EPS0;
MY0 = 4.0 * 1e-7 * pi;
WLANG = 1.0/(sqrt(EPS0*MY0)*F);
ABST = 0.75;
DELTA = ABST / (M+1);
H0 = (2.0 * pi * F * DELTA)*(2.0 * pi * F * DELTA) * MY0 * EPS0 - 4.0;
H1 = (2.0 * pi * F * DELTA)*(2.0 * pi * F * DELTA) * MY0 * EPS1 - 4.0;
H2 = (2.0 * pi * F * DELTA)*(2.0 * pi * F * DELTA) * MY0 * EPS2 - 4.0;
H3 = (2.0 * pi * F * DELTA)*(2.0 * pi * F * DELTA) * MY0 * EPS3 - 4.0;
fprintf("EPS0: %f %fi\n",real(H0),imag(H0));
fprintf("EPS1: %f %fi\n",real(H1),imag(H1));
fprintf("EPS2: %f %fi\n",real(H2),imag(H2));
fprintf("EPS3: %f %fi\n",real(H3),imag(H3));
BODY_X = 27;
BODY_Y = 17;
BODY = zeros(BODY_X,BODY_Y);
BODY(2,:) = [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0];
BODY(3,:) = [0, 0, 0, 0, 0, 1, 1, 1, 2, 2, 1, 1, 1, 0, 0, 0, 0, 0];
BODY(4,:) = [0, 0, 0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 0, 0, 0, 0];
BODY(5,:) = [0, 0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0, 0, 0];
BODY(6,:) = [0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 0, 0, 0];
BODY(7,:) = [0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0, 0];
BODY(8,:) = [0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 0, 0];
BODY(9,:) = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0];
BODY(10,:) = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0];
BODY(11,:) = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0];
BODY(12,:) = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 2, 2, 1, 0, 0];
BODY(13,:) = [0, 1, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 1, 0, 0];
BODY(14,:) = [0, 1, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 1, 0, 0];
BODY(15,:) = [0, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 1, 0, 0];
BODY(16,:) = [0, 1, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 2, 2, 1, 0, 0, 0];
BODY(17,:) = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0];
BODY(18,:) = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0];
BODY(19,:) = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0];
BODY(20,:) = [0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 0, 0];
BODY(21,:) = [0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0, 0];
BODY(22,:) = [0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 0, 0, 0];
BODY(23,:) = [0, 0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0, 0, 0];
BODY(24,:) = [0, 0, 0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 0, 0, 0, 0];
BODY(25,:) = [0, 0, 0, 0, 0, 1, 1, 1, 2, 2, 1, 1, 1, 0, 0, 0, 0, 0];
BODY(26,:) = [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0];
ABSTAND_X = floor((M-BODY_X*FAKTOR)/2) - 3; % muss eventuell angepasst werden
ABSTAND_Y = 30 * FAKTOR;
Dielektrikum = zeros(M,M);
for I=1:M
    for J=1:M
        if ((J > ABSTAND_Y) & (J <= ABSTAND_Y+BODY_Y*FAKTOR) & (I > ABSTAND_X) & (I <=
(ABSTAND_X+BODY_X*FAKTOR)))
            help1 = floor((I-ABSTAND_X)/FAKTOR);
            if (help1 < 1)
                help1 = 1;
            end
            help2 = floor((J-ABSTAND_Y)/FAKTOR);
            if (help2 < 1)
                help2 = 1;
            end
            Dielektrikum(J,I) = BODY(help1,help2);
        else
            Dielektrikum(J,I) = 0;
        end
    end
end
end
figure;
contour(Dielektrikum)
grid on;
ylabel({'200 Rasterpunkte entsprechen 1m', 'Wellenausbreitung --->'});
xlabel('200 Rasterpunkte entsprechen 1m');
ax = gca;
ax.FontSize = 14;
Dielektrikum = changem(Dielektrikum,[H0],[0]);

```

```

Dielektrikum = changem(Dielektrikum,[H1],[1]);
Dielektrikum = changem(Dielektrikum,[H2],[2]);
Dielektrikum = changem(Dielektrikum,[H3],[3]);
% CALL CHOBANDET
fprintf("\nCHOBANDET");
fprintf("\nStartzeit: %s", datetime('now'));
L = zeros(N,M+1);
X = zeros(N,1);
fprintf("\nM = %i", M);
fprintf("\nL[%i][%i] X[%i] angelegt", N,M+1,N);
for I=1:N
    if (mod(I,5000) == 0)
        fprintf("\nI = %i %s", I, datetime('now'));
    end
    if (I > M)
        P = 0;
    else
        P = M-I+1;
    end
    R = I-M+P;
    for J=P:M
        S = J - 1;
        Q = M - J + P;
        if (J == 0)
            Y = 1.0;
        else
            Y = 0.0;
            if (J == M-1)
                K1 = mod((I-1),M);
                if (K1 ~= 0)
                    Y = 1.0;
                end
            end
        end
        if (J == M)
            IX = floor((I-1)/M) + 1;
            IY = mod(I-1,M) + 1;
            Y = Dielektrikum(IX,IY);
        end
        if (P <= S) % überprüft, ob S=-1 (wegen der Korrektur von J möglich)
                    % eigentlich nicht notwendig, da die for-Loop K=P:S
                    % ohnedies nicht durchgeführt werden würde
            for K=P:S
                Y = Y - L(I,K+1) * L(R,Q+1);
                Q = Q + 1;
            end
        else
            % fprintf("(P <= S) nicht erfüllt P=%i S=%i\n",P,S);
        end
        if (J == M)
            L(I,J+1) = 1.0/sqrt(Y);
        else
            L(I,J+1) = Y * L(R,M+1);
        end
        R = R+1;
    end
end
% CALL CHOBANDSOL
fprintf("\nCHOBANDSOL");
fprintf("\nZwischenzeit: %s", datetime('now'));
S = M - 1;
for I=1:N
    if (I>M)
        P=0;
    else
        P = M-I+1;
    end
    Q = I;
    Y = 0.0;
    U = mod((I-1),M);
    if (U == 0)
        Y = sin(2.0*pi*DELTA/WLANG*(M-floor((I-1)/M)));
    end
    U = mod(I,M);
    if (U == 0)
        Y = sin(2.0*pi*DELTA/WLANG*(M-floor(I/M)+1));
    end
end

```

```

end
if (I <= M)
    Y = Y + sin(ABST*2.0*pi/WLANG);
end
for K=S:-1:P
    Q = Q - 1;
    Y = Y - L(I,K+1) * X(Q);
end
X(I) = Y * L(I,M+1);
end
for I=N:-1:1
    if (N-I > M)
        P = 0;
    else
        P = M-N+I;
    end
    Y = X(I);
    Q = I;
    for K=S:-1:P
        Q = Q + 1;
        Y = Y - L(Q,K+1) * X(Q);
    end
    X(I) = Y * L(I,M+1);
end
Dielektrikum = zeros(M,M);
for I=1:M
    for J=1:M
        Dielektrikum(I, J) = abs(X((I-1)*M+J));
    end
end
figure;
surf(Dielektrikum);
grid on;
title('Lösung BandMatrix');
ylabel({'200 Rasterpunkte entsprechen 1m', 'Wellenausbreitung --->'});
xlabel('200 Rasterpunkte entsprechen 1m');
zlabel('relative units');
ax = gca;
ax.FontSize = 14;
c = colorbar;
c.Label.String = 'elektr.Feldstärke (relative units)';
fprintf("\nEndzeit: %s\n", datetime('now'));

```

BandMatrix.py

```

# Lösung als Band-Matrix
import math, cmath
import numpy as np
import matplotlib.pyplot;
import time;
pi = math.pi;
FAKTOR = 4;
M = 50 * FAKTOR;          # M x M Unbekannte, (M+2) x (M+2) Feldpunkte
N = M * M;
f = 10000000000.0;
epsR1 = 5.0;
sigma1 = 0.03;
epsR2 = 30.0;
sigma2 = 0.3;
epsR3 = 80.0;
sigma3 = 1.00;
eps0 = 1.0 / (36.0e9 * pi);
eps1 = eps0*epsR1 + sigma1/(2j*pi*f);
eps2 = eps0*epsR2 + sigma2/(2j*pi*f);
eps3 = eps0*epsR3 + sigma3/(2j*pi*f);
#eps1 = eps0;
#eps2 = eps0;
#eps3 = eps0;
my0 = 4.0 * 1e-7 * pi;
wlang = 1.0/(math.sqrt(eps0*my0)*f);
abst = 0.75;
delta = abst / (M+1);
eps0 = (2.0 * pi * f * delta)*(2.0 * pi * f * delta) * my0 * eps0 - 4.0;
eps1 = (2.0 * pi * f * delta)*(2.0 * pi * f * delta) * my0 * eps1 - 4.0;
eps2 = (2.0 * pi * f * delta)*(2.0 * pi * f * delta) * my0 * eps2 - 4.0;
eps3 = (2.0 * pi * f * delta)*(2.0 * pi * f * delta) * my0 * eps3 - 4.0;

```

```

print("\neps0: ", eps0);
print("eps1: ", eps1);
print("eps2: ", eps2);
print("eps3: ", eps3);
body_X = 27;
body_Y = 17;
body = np.zeros((body_X, body_Y));
body[1][0:body_Y] = [0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0];
body[2][0:body_Y] = [0, 0, 0, 0, 0, 1, 1, 1, 2, 2, 1, 1, 1, 0, 0, 0, 0];
body[3][0:body_Y] = [0, 0, 0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 0, 0, 0];
body[4][0:body_Y] = [0, 0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0, 0];
body[5][0:body_Y] = [0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 0, 0];
body[6][0:body_Y] = [0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0];
body[7][0:body_Y] = [0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 0];
body[8][0:body_Y] = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0];
body[9][0:body_Y] = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0];
body[10][0:body_Y] = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0];
body[11][0:body_Y] = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 2, 2, 1, 0];
body[12][0:body_Y] = [0, 1, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 1, 0];
body[13][0:body_Y] = [0, 1, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 1, 0];
body[14][0:body_Y] = [0, 1, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 1, 0];
body[15][0:body_Y] = [0, 1, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 2, 2, 1, 0];
body[16][0:body_Y] = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0];
body[17][0:body_Y] = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0];
body[18][0:body_Y] = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0];
body[19][0:body_Y] = [0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 0];
body[20][0:body_Y] = [0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0];
body[21][0:body_Y] = [0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 0, 0];
body[22][0:body_Y] = [0, 0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0, 0];
body[23][0:body_Y] = [0, 0, 0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 1, 0, 0, 0];
body[24][0:body_Y] = [0, 0, 0, 0, 0, 1, 1, 1, 2, 2, 1, 1, 1, 0, 0, 0];
body[25][0:body_Y] = [0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0];
abstand_X = (M-body_X*FAKTOR) // 2 - 3; # muss eventuell angepasst werden
abstand_Y = 30 * FAKTOR;
Dielektrikum = np.zeros((M,M), dtype=complex);
I = 0;
while I < M:
    J = 0;
    while J < M:
        if ((J > Abstand_Y) & (J <= Abstand_Y+body_Y*FAKTOR) & (I > Abstand_X) & (I <=
(abstand_X+body_X*FAKTOR))):
            help1 = (I-abstand_X) // FAKTOR;
            if (help1 < 1):
                help1 = 1;
            help2 = (J-abstand_Y) // FAKTOR;
            if (help2 < 1):
                help2 = 1;
            Dielektrikum[J][I] = body[help1-1][help2-1];
        else:
            Dielektrikum[J][I] = 0;
        J += 1;
    I += 1;
matplotlib.pyplot.contour(Dielektrikum);
matplotlib.pyplot.title('Body');
matplotlib.pyplot.ylabel('Wellenausbreitung -->');
matplotlib.pyplot.show();
I = 0;
while I < M:
    J = 0;
    while J < M:
        if (Dielektrikum[I][J] == 0):
            Dielektrikum[I][J] = eps0;
        if (Dielektrikum[I][J] == 1):
            Dielektrikum[I][J] = eps1;
        if (Dielektrikum[I][J] == 2):
            Dielektrikum[I][J] = eps2;
        if (Dielektrikum[I][J] == 3):
            Dielektrikum[I][J] = eps3;
        J += 1;
    I += 1;
# CALL CHOBANDET
print('CHOBANDET');
print("Startzeit: ", time.strftime('%Y-%m-%d %H:%M:%S', time.localtime()));
L = np.zeros((N,M+1), dtype=complex);
X = np.zeros((N,1), dtype=complex);
Y = np.zeros((1), dtype=complex);

```

```

print('M = ', M);
print('L[', N, '][', M+1, ']' X['', N, ']'   angelegt');
# in Python läuft der Index von 0 weg --> alle Indizes, die auf ein Array weisen werden um 1
verringert (im Vergleich zu BandMatrix.m)
I = 1;
while I <= N:
    if ((I % 5000) == 0):
        print('I = ', I, ' ', time.strftime('%Y-%m-%d %H:%M:%S', time.localtime()));
    if (I > M):
        P = 0;
    else:
        P = M-I+1;
    R = I-M+P;
    J = P;
    while J <= M:
        S = J - 1;
        Q = M - J + P;
        if J == 0:
            Y = 1.0 + 0.0j;
        else:
            Y = 0.0 + 0.0j;
            if J == M-1:
                K1 = (I-1) % M;
                if (K1 != 0):
                    Y = 1.0 + 0.0j;
            if (J == M):
                IX = ((I-1) // M) + 1;
                IY = ((I-1) % M) + 1;
                Y = Dielektrikum[IX-1][IY-1];
            if (P <= S):      # überprüft, ob S=-1 (wegen der Korrektur von J möglich)
                            # eigentlich nicht notwendig, da die for-Loop K=P:S
                            # ohnedies nicht durchgeführt werden würde
                K = P;
                while K <= S:
                    Y = Y - L[I-1][K+1-1] * L[R-1][Q+1-1];
                    Q = Q + 1;
                    K += 1;
            if (J == M):
                L[I-1][J+1-1] = 1.0/cmath.sqrt(Y);
            else:
                L[I-1][J+1-1] = Y * L[R-1][M+1-1];
            R = R+1;
            J += 1;
        I += 1;
# CALL CHOBANDSOL
print('CHOBANDSOL');
print("Zwischenzeit: ", time.strftime('%Y-%m-%d %H:%M:%S', time.localtime()));
S = M - 1;
I = 1;
while I <= N:
    if (I>M):
        P=0;
    else:
        P = M-I+1;
        Q = I;
        Y = 0.0;
        U = (I-1) % M;
        if (U == 0):
            Y = math.sin(2.0*pi*delta/wlang*(M - (I-1)//M));
        U = I % M;
        if (U == 0):
            Y = math.sin(2.0*pi*delta/wlang*(M - (I//M) + 1));
        if (I <= M):
            Y = Y + math.sin(abst*2.0*pi/wlang);
        K = S;
        while K >= P:
            Q = Q - 1;
            Y = Y - L[I-1][K+1-1] * X[Q-1];
            K -= 1;
        X[I-1] = Y * L[I-1][M+1-1];
        I += 1;
I = N;
while I >= 1:
    if (N-I > M):
        P = 0;
    else:

```

```

        P = M-N+I;
        Y = X[I-1];
        Q = I;
        K = S;
        while K >= P:
            Q = Q + 1;
            Y = Y - L[Q-1][K+1-1] * X[Q-1];
            K -= 1;
            X[I-1] = Y * L[I-1][M+1-1];
            I -= 1;
E = np.zeros((M,M));
I = 0;
while I < M:
    J = 0;
    while J < M:
        E[I][J] = np.absolute(X[I*M+J]);
        J += 1;
    I += 1;
print("Endzeit: ", time.strftime('%Y-%m-%d %H:%M:%S', time.localtime()));
levels = [0.0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.2, 0.3, 0.4, 0.5,
0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2];
matplotlib.pyplot.contourf(E, levels);
matplotlib.pyplot.title('E-Feld');
matplotlib.pyplot.ylabel('Wellenausbreitung -->');
matplotlib.pyplot.show();# Version August 2020
# Lösung als Band-Matrix

import math, cmath
import numpy as np
import matplotlib.pyplot;
import time;

pi = math.pi;
FAKTOR = 4;
M = 50 * FAKTOR;          # M x M Unbekannte, (M+2) x (M+2) Feldpunkte
N = M * M;
f = 1000000000.0;
epsR1 = 5.0;
sigma1 = 0.03;
epsR2 = 30.0;
sigma2 = 0.3;
epsR3 = 80.0;
sigma3 = 1.00;
eps0 = 1.0 / (36.0e9 * pi);
eps1 = eps0*epsR1 + sigma1/(2j*pi*f);
eps2 = eps0*epsR2 + sigma2/(2j*pi*f);
eps3 = eps0*epsR3 + sigma3/(2j*pi*f);
#eps1 = eps0;
#eps2 = eps0;
#eps3 = eps0;
my0 = 4.0 * 1e-7 * pi;
wlang = 1.0/(math.sqrt(eps0*my0)*f);
abst = 0.75;
delta = abst / (M+1);
eps0 = (2.0 * pi * f * delta)*(2.0 * pi * f * delta) * my0 * eps0 - 4.0;
eps1 = (2.0 * pi * f * delta)*(2.0 * pi * f * delta) * my0 * eps1 - 4.0;
eps2 = (2.0 * pi * f * delta)*(2.0 * pi * f * delta) * my0 * eps2 - 4.0;
eps3 = (2.0 * pi * f * delta)*(2.0 * pi * f * delta) * my0 * eps3 - 4.0;
print("\neps0: ", eps0);
print("eps1: ", eps1);
print("eps2: ", eps2);
print("eps3: ", eps3);
body_X = 27;
body_Y = 17;
body = np.zeros((body_X, body_Y));
body[1][0:body_Y] = [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0];
body[2][0:body_Y] = [0, 0, 0, 0, 0, 1, 1, 1, 2, 2, 1, 1, 1, 0, 0, 0, 0];
body[3][0:body_Y] = [0, 0, 0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 0, 0, 0];
body[4][0:body_Y] = [0, 0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0, 0];
body[5][0:body_Y] = [0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 0, 0];
body[6][0:body_Y] = [0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0];
body[7][0:body_Y] = [0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 0];
body[8][0:body_Y] = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0];
body[9][0:body_Y] = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0];
body[10][0:body_Y] = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0];
body[11][0:body_Y] = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 2, 2, 1, 0];

```

```

body[12][0:body_Y] = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 1, 0];
body[13][0:body_Y] = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 1, 0];
body[14][0:body_Y] = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 1, 0];
body[15][0:body_Y] = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 2, 2, 1, 0];
body[16][0:body_Y] = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0];
body[17][0:body_Y] = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0];
body[18][0:body_Y] = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0];
body[19][0:body_Y] = [0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 0];
body[20][0:body_Y] = [0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0];
body[21][0:body_Y] = [0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 0, 0];
body[22][0:body_Y] = [0, 0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0, 0];
body[23][0:body_Y] = [0, 0, 0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 0, 0, 0];
body[24][0:body_Y] = [0, 0, 0, 0, 0, 1, 1, 1, 2, 2, 1, 1, 1, 0, 0, 0, 0];
body[25][0:body_Y] = [0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0];
abstand_X = (M-body_X*FAKTOR) // 2;
abstand_Y = 30 * FAKTOR;
Dielektrikum = np.zeros((M,M), dtype=complex);
I = 0;
while I < M:
    J = 0;
    while J < M:
        if ((J > Abstand_Y) & (J <= Abstand_Y+body_Y*FAKTOR) & (I > Abstand_X) & (I <=
(abstand_X+body_X*FAKTOR))):
            help1 = (I-abstand_X) // FAKTOR;
            if (help1 < 1):
                help1 = 1;
            help2 = (J-abstand_Y) // FAKTOR;
            if (help2 < 1):
                help2 = 1;
            Dielektrikum[J][I] = body[help1-1][help2-1];
        else:
            Dielektrikum[J][I] = 0;
        J += 1;
    I += 1;
matplotlib.pyplot.contour(Dielektrikum);
matplotlib.pyplot.title('Body');
matplotlib.pyplot.ylabel('200 Rasterpunkte entsprechen 1m, Wellenausbreitung --->');
matplotlib.pyplot.xlabel('200 Rasterpunkte entsprechen 1m');
matplotlib.pyplot.show();
I = 0;
while I < M:
    J = 0;
    while J < M:
        if (Dielektrikum[I][J] == 0):
            Dielektrikum[I][J] = eps0;
        if (Dielektrikum[I][J] == 1):
            Dielektrikum[I][J] = eps1;
        if (Dielektrikum[I][J] == 2):
            Dielektrikum[I][J] = eps2;
        if (Dielektrikum[I][J] == 3):
            Dielektrikum[I][J] = eps3;
        J += 1;
    I += 1;
print('CHOBANDET');
print("Startzeit: ", time.strftime('%Y-%m-%d %H:%M:%S', time.localtime()));
L = np.zeros((N,M+1), dtype=complex);
X = np.zeros((N,1), dtype=complex);
Y = np.zeros((1), dtype=complex);
print('M = ', M);
print('L[' , N, '][' , M+1, ']' , X[' , N, ']' , 'angelegt');
# in Python läuft der Index von 0 weg --> alle Indizes, die auf ein Array weisen werden um 1
verringert (im Vergleich zu BandMatrix.m)
I = 1;
while I <= N:
    if ((I % 5000) == 0):
        print('I = ', I, ' ', time.strftime('%Y-%m-%d %H:%M:%S', time.localtime()));
    if (I > M):
        P = 0;
    else:
        P = M-I+1;
    R = I-M+P;
    J = P;
    while J <= M:
        S = J - 1;
        Q = M - J + P;
        if J == 0:

```

```

        Y = 1.0 + 0.0j;
    else:
        Y = 0.0 + 0.0j;
        if J == M-1:
            K1 = (I-1) % M;
            if (K1 != 0):
                Y = 1.0 + 0.0j;
        if (J == M):
            IX = ((I-1) // M) + 1;
            IY = ((I-1) % M) + 1;
            Y = Dielektrikum[IX-1][IY-1];
        if (P <= S):      # überprüft, ob S=-1 (wegen der Korrektur von J möglich)
                        # eigentlich nicht notwendig, da die for-Loop K=P:S
                        # ohnedies nicht durchgeführt werden würde
            K = P;
            while K <= S:
                Y = Y - L[I-1][K+1-1] * L[R-1][Q+1-1];
                Q = Q + 1;
                K += 1;
        if (J == M):
            L[I-1][J+1-1] = 1.0/cmath.sqrt(Y);
        else:
            L[I-1][J+1-1] = Y * L[R-1][M+1-1];
        R = R+1;
        J += 1;
    I += 1;
print('CHOBANDSOL');
print("Zwischenzeit: ", time.strftime('%Y-%m-%d %H:%M:%S', time.localtime()));
S = M - 1;
I = 1;
while I <= N:
    if (I>M):
        P=0;
    else:
        P = M-I+1;
    Q = I;
    Y = 0.0;
    U = (I-1) % M;
    if (U == 0):
        Y = math.sin(2.0*pi*delta/wlang*(M - (I-1)//M));
    U = I % M;
    if (U == 0):
        Y = math.sin(2.0*pi*delta/wlang*(M - (I//M) + 1));
    if (I <= M):
        Y = Y + math.sin(abst*2.0*pi/wlang);
    K = S;
    while K >= P:
        Q = Q - 1;
        Y = Y - L[I-1][K+1-1] * X[Q-1];
        K -= 1;
    X[I-1] = Y * L[I-1][M+1-1];
    I += 1;
I = N;
while I >= 1:
    if (N-I > M):
        P = 0;
    else:
        P = M-N+I;
    Y = X[I-1];
    Q = I;
    K = S;
    while K >= P:
        Q = Q + 1;
        Y = Y - L[Q-1][K+1-1] * X[Q-1];
        K -= 1;
    X[I-1] = Y * L[I-1][M+1-1];
    I -= 1;
E = np.zeros((M,M));
I = 0;
while I < M:
    J = 0;
    while J < M:
        E[I][J] = np.absolute(X[I*M+J]);
        J += 1;
    I += 1;
print("Endzeit: ", time.strftime('%Y-%m-%d %H:%M:%S', time.localtime()));

```

```

levels = [0.0, 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.2, 0.3, 0.4, 0.5,
0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2];
matplotlib.pyplot.contourf(E, levels);
matplotlib.pyplot.title('E-Feld');
matplotlib.pyplot.ylabel('200 Rasterpunkte entsprechen 1m, Wellenausbreitung --->');
matplotlib.pyplot.xlabel('200 Rasterpunkte entsprechen 1m');
clb=matplotlib.pyplot.colorbar();
clb.ax.set_title('rel.units');
matplotlib.pyplot.show();

```

SparseMatrix.m

```

% Lösung "Sparse-Matrix mit MATLAB"
clear all;           % clear variables window
clc;                 % clear command window display
close all;           % close all figures
FAKTOR=input('\n\nEingabe Rasteranzahl als Vielfaches von 50: ');
M = FAKTOR * 50;     % M x M Unbekannte, (M+2) x (M+2) Feldpunkte
N = M * M;
N1 = N-M;
F = 1.0e9;
fprintf("Frequenz: %.0f Hz\n", F);
EPSR1 = 5.0;
sigma1 = 0.03;
EPSR2 = 30.0;
sigma2 = 0.3;
EPSR3 = 80.0;
sigma3 = 1.00;
EPS0 = 1.0 / (36.0e9 * pi);
EPS1 = EPS0*EPSR1 + sigma1/(2j*pi*F);
EPS2 = EPS0*EPSR2 + sigma2/(2j*pi*F);
EPS3 = EPS0*EPSR3 + sigma3/(2j*pi*F);
% EPS1 = EPS0;
% EPS2 = EPS0;
% EPS3 = EPS0;
MY0 = 4.0 * 1e-7 * pi;
WLANG = 1.0/(sqrt(EPS0*MY0)*F);
ABST = 0.75;
DELTA = ABST / (M+1);
H0 = (2.0 * pi * F * DELTA)*(2.0 * pi * F * DELTA) * MY0 * EPS0 - 4.0;
H1 = (2.0 * pi * F * DELTA)*(2.0 * pi * F * DELTA) * MY0 * EPS1 - 4.0;
H2 = (2.0 * pi * F * DELTA)*(2.0 * pi * F * DELTA) * MY0 * EPS2 - 4.0;
H3 = (2.0 * pi * F * DELTA)*(2.0 * pi * F * DELTA) * MY0 * EPS3 - 4.0;
fprintf("EPS0: %f %fi\n",real(H0),imag(H0));
fprintf("EPS1: %f %fi\n",real(H1),imag(H1));
fprintf("EPS2: %f %fi\n",real(H2),imag(H2));
fprintf("EPS3: %f %fi\n",real(H3),imag(H3));
BODY_X = 27;
BODY_Y = 17;
ABSTAND_X = floor((M-BODY_X*FAKTOR)/2);
ABSTAND_Y = 30 * FAKTOR;
BODY = zeros(BODY_X,BODY_Y);
BODY(2,:) = [0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0];
BODY(3,:) = [0, 0, 0, 0, 0, 1, 1, 1, 2, 2, 1, 1, 0, 0, 0, 0];
BODY(4,:) = [0, 0, 0, 0, 1, 1, 2, 2, 2, 2, 2, 1, 1, 0, 0, 0];
BODY(5,:) = [0, 0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0, 0];
BODY(6,:) = [0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 0, 0];
BODY(7,:) = [0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0];
BODY(8,:) = [0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 0];
BODY(9,:) = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0];
BODY(10,:) = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0];
BODY(11,:) = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0];
BODY(12,:) = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 2, 1, 0];
BODY(13,:) = [0, 1, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 1, 0];
BODY(14,:) = [0, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 1, 0];
BODY(15,:) = [0, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 1, 0];
BODY(16,:) = [0, 1, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 2, 1, 0];
BODY(17,:) = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0];
BODY(18,:) = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0];
BODY(19,:) = [0, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0];
BODY(20,:) = [0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0];
BODY(21,:) = [0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0];
BODY(22,:) = [0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 0, 0];
BODY(23,:) = [0, 0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 1, 0, 0, 0];
BODY(24,:) = [0, 0, 0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 0, 0];
BODY(25,:) = [0, 0, 0, 0, 0, 1, 1, 1, 2, 2, 1, 1, 1, 0, 0, 0];
BODY(26,:) = [0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0];

```

```

ABSTAND_X = floor((M-BODY_X*FAKTOR)/2) - 3; % muss eventuell angepasst werden
ABSTAND_Y = 30 * FAKTOR;
Dielektrikum = zeros(M,M);
for I=1:M
    for J=1:M
        if ((J > ABSTAND_Y) & (J <= ABSTAND_Y+BODY_Y*FAKTOR) & (I > ABSTAND_X) & (I <=
(ABSTAND_X+BODY_X*FAKTOR)))
            help1 = floor((I-ABSTAND_X)/FAKTOR);
            if (help1 < 1)
                help1 = 1;
            end
            help2 = floor((J-ABSTAND_Y)/FAKTOR);
            if (help2 < 1)
                help2 = 1;
            end
            Dielektrikum(J,I) = BODY(help1,help2);
        else
            Dielektrikum(J,I) = 0;
        end
    end
end
figure;
contour(Dielektrikum)
grid on;
ylabel('Wellenausbreitung -->');
Dielektrikum = changem(Dielektrikum,[H0],[0]);
Dielektrikum = changem(Dielektrikum,[H1],[1]);
Dielektrikum = changem(Dielektrikum,[H2],[2]);
Dielektrikum = changem(Dielektrikum,[H3],[3]);
fprintf("\nStartzeit: %s", datetime('now'));
% Lösungsversuch mit schwachbesetzte Matrix
S = zeros(N, 5);
d = [-M;-1;0;1;M];
for I=1:N
    if (I<=N1)
        S(I,1) = 1;
    end
    if (mod(I,M))
        S(I,2) = 1;
    end
    if (mod(I-1,M))
        S(I,4) = 1;
    end
    if (I>M)
        S(I,5) = 1;
    end
    IX = floor((I-1)/M) + 1;
    IY = mod(I-1,M) + 1;
    Y = Dielektrikum(IX,IY);
    S(I, 3) = Y;
end
A = spdiags(S,d,N,N);
fprintf("\nsparse Matrix A ist angelegt");
fprintf("\nZwischenzeit: %s", datetime('now'));
B = zeros(N,1);
Z1 = 1;
Z2 = 1;
for I=1:N
    U = floor((I-1)/M) + 2;
    V = mod(I, M);
    if (U == 2)
        B(I,1) = -sin(ABST*2.0*pi/WLANG); % Startebene
        Z2 = Z2 + 1;
    end
    if (U == M+1)
        B(I,1) = 0; % leitende Ebene
        Z2 = Z2+1;
    end
    if ((V == 1) | (V == 0)) % j=1 oder j = jmax
        B(I,1) = B(I,1) - sin(2.0*pi*DELTA/WLANG*(M-floor((I-1)/M)));
    end
end
fprintf("\nMatrix B ist angelegt");
fprintf("\nZwischenzeit: %s", datetime('now'));
X = A \ B;

```

```

for I=1:M
    for J=1:M
        E(I, J) = abs(X((I-1)*M+J));
    end
end
figure;
surf(E)
title('Lösung mit Solver von MATLAB');
grid on;
ylabel({'200 Rasterpunkte entsprechen 1m', 'Wellenausbreitung --->'});
xlabel('200 Rasterpunkte entsprechen 1m');
zlabel('relative units');
ax = gca;
ax.FontSize = 14;
c = colorbar;
c.Label.String = 'elektr.Feldstärke (relative units)';

```

3Dgrid.plt

```

#!/usr/bin/gnuplot
# Plot cube
reset
unset key
set label font 'Verdana,6'
set grid lt -1
unset grid
set xr [-1 : 7]
set yr [-1 : 7]
set zr [-1 : 7]
set xtics 1
set ytics 1
set ztics 1
set for [x = 0:6:1] arrow from x, 0, 0 to x, 0, 6 nohead lt 0
set for [z = 0:6:1] arrow from 0, 0, z to 6, 0, z nohead lt 0
set for [y = 0:6:1] arrow from 6, y, 6 to 0, y, 6 nohead lt 0
set for [x = 0:6:1] arrow from x, 0, 6 to x, 6, 6 nohead lt 0
set for [y = 0:6:1] arrow from 6, y, 0 to 6, y, 6 nohead lt 0
set for [z = 0:6:1] arrow from 6, 0, z to 6, 6, z nohead lt 0
k = 5
do for [i = 0:5:1] for [j = 0:5:1] { set label sprintf("A[%d,%d,%d]",i+1,j+1,k+1) font ",6" at
(i)+0.1,(j)+0.5,(k)+1.1 }
j = 0
do for [i = 0:5:1] for [k = 0:5:1] { set label sprintf("A[%d,%d,%d]",i+1,j+1,k+1) font ",6" at
(i)+0.1,(j)+0.5,(k)+0.3 }
i = 5
do for [j = 0:5:1] for [k = 0:5:1] { set label sprintf("A[%d,%d,%d]",i+1,j+1,k+1) font ",6" at
(i)+0.1,(j)+1.8,(k)+0.3 }
set size ratio 1
#set size 1,1
set view equal xyz
set hidden3d
set xyplane 0
set view 80,25
set iso 25
set xlabel 'x-Achse'
set ylabel 'y-Achse'
set zlabel 'z-Achse'
splot [x=-1:6] [y=-1:6] 1.5*x+0.5*y-2

```

Kopf3D_Zubal.m

```

clear all; % clear variables window
clc; % clear command window display
close all; % close all figures
frequ = 2.5e9;
epsrAir = 1;
sigmaAir = 0;
epsrSkinDry = 37.952;
sigmaSkinDry = 1.4876;
epsrSkinWet = 42.783;
sigmaSkinWet = 1.6223;
epsrFat = 5.2749;
sigmaFat = 0.10672;
epsrSkull = 11.352;
sigmaSkull = 0.40411;
epsrSpongyBone = 18.491;
sigmaSpongyBone = 0.82286;

```

```

epsrWhiteMatter = 36.107;
sigmaWhiteMatter = 1.2403;
epsrGrayMatter = 48.83;
sigmaGrayMatter = 1.843;
epsrBlood = 58.181;
sigmaBlood = 2.5878;
epsrCSF = 66.168;
sigmaCSF = 3.5041;
epsrDura = 41.971;
sigmaDura = 1.6983;
epsrBoneMarrow = 5.2914;
sigmaBoneMarrow = 0.097266;
epsrCerebellum = 44.716;
sigmaCerebellum = 2.1342;
epsrSpinalCord = 30.095;
sigmaSpinalCord = 1.1093;
epsrEyeTissue = 52.558;
sigmaEyeTissue = 2.0702;
epsrEyeLense = 44.57;
sigmaEyeLense = 1.5347;
epsrCartilage = 38.663;
sigmaCartilage = 1.7949;
epsrMuscle = 52.668;
sigmaMuscle = 1.773;
epsrMuscleTongue = 52.558;
sigmaMuscleTongue = 1.8396;
epsrMuscleTrachea = 39.676;
sigmaMuscleTrachea = 1.4774;
epsrStomach = 62.078;
sigmaStomach = 2.2546;
eps0 = 1.0 / (36.0e9 * pi);
my0 = 4.0 * 1e-7 * pi;
epsAir = eps0*epsrAir + sigmaAir/(2j*pi*frequ);
epsSkinDry = eps0*epsrSkinDry + sigmaSkinDry/(2j*pi*frequ);
epsSkinWet = eps0*epsrSkinWet + sigmaSkinWet/(2j*pi*frequ);
epsFat = eps0*epsrFat + sigmaFat/(2j*pi*frequ);
epsSkull = eps0*epsrSkull + sigmaSkull/(2j*pi*frequ);
epsSpongyBone = eps0*epsrSpongyBone + sigmaSpongyBone/(2j*pi*frequ);
epsWhiteMatter = eps0*epsrWhiteMatter + sigmaWhiteMatter/(2j*pi*frequ);
epsGrayMatter = eps0*epsrGrayMatter + sigmaGrayMatter/(2j*pi*frequ);
epsBlood = eps0*epsrBlood + sigmaBlood/(2j*pi*frequ);
epsCSF = eps0*epsrCSF + sigmaCSF/(2j*pi*frequ);
epsDura = eps0*epsrDura + sigmaDura/(2j*pi*frequ);
epsBoneMarrow = eps0*epsrBoneMarrow + sigmaBoneMarrow/(2j*pi*frequ);
epsCerebellum = eps0*epsrCerebellum + sigmaCerebellum/(2j*pi*frequ);
epsSpinalCord = eps0*epsrSpinalCord + sigmaSpinalCord/(2j*pi*frequ);
epsEyeTissue = eps0*epsrEyeTissue + sigmaEyeTissue/(2j*pi*frequ);
epsEyeLense = eps0*epsrEyeLense + sigmaEyeLense/(2j*pi*frequ);
epsCartilage = eps0*epsrCartilage + sigmaCartilage/(2j*pi*frequ);
epsMuscle = eps0*epsrMuscle + sigmaMuscle/(2j*pi*frequ);
epsMuscleTongue = eps0*epsrMuscleTongue + sigmaMuscleTongue/(2j*pi*frequ);
epsMuscleTrachea = eps0*epsrMuscleTrachea + sigmaMuscleTrachea/(2j*pi*frequ);
epsStomach = eps0*epsrStomach + sigmaStomach/(2j*pi*frequ);
fid=fopen('det_head_u2med.dat','r');
a_read=fread(fid);
fclose(fid);
a=reshape(a_read,[256 256 128]); % Reshape column vector into Matrix and Pages format
[m,n,p] = size(a); % determine size of 3D volume array
a = rot90(a,3); % um 3*90° gegen Uhrzeiger drehen
% tissueID ersetzen
Head = changem(a,[epsAir epsAir epsAir],[0 15 104]);
Head = changem(Head,[epsSkinDry],[1]);
Head = changem(Head,[epsSkinWet],[72]);
Head = changem(Head,[epsFat epsFat],[22 98]);
Head = changem(Head,[epsSkull epsSkull epsSkull epsSkull epsSkull epsSkull epsSkull epsSkull],
[4 5 70 71 76 81 82 99 102 125]);
Head = changem(Head,[epsSpongyBone],[100]);
Head = changem(Head,[epsWhiteMatter epsWhiteMatter epsWhiteMatter],[83 107 111]);
Head = changem(Head,[epsGrayMatter epsGrayMatter epsGrayMatter epsGrayMatter epsGrayMatter
epsGrayMatter epsGrayMatter epsGrayMatter epsGrayMatter epsGrayMatter epsGrayMatter
epsGrayMatter epsGrayMatter epsGrayMatter],[89 95 96 101 103 105 106 108 109 112 114 117 118
124]);
Head = changem(Head,[epsBlood],[23]);
Head = changem(Head,[epsCSF epsCSF epsCSF epsCSF epsCSF],[2 75 92 115 122]);
Head = changem(Head,[epsDura epsDura epsDura],[84 97 113]);
Head = changem(Head,[epsBoneMarrow],[26]);

```

```

Head = changem(Head,[epsCerebellum],[77]);
Head = changem(Head,[epsSpinalCord epsSpinalCord epsSpinalCord epsSpinalCord epsSpinalCord],[3
85 91 120 123]);
Head = changem(Head,[epsEyeTissue epsEyeTissue epsEyeTissue epsEyeTissue],[74 110 116 119]);
Head = changem(Head,[epsEyeLense],[121]);
Head = changem(Head,[epsCartilage],[30]);
Head = changem(Head,[epsMuscle],[9]);
Head = changem(Head,[epsMuscleTongue],[78]);
Head = changem(Head,[epsMuscleTrachea],[29]);
Head = changem(Head,[epsStomach],[16]);
Schicht1 = 20;
Schicht2 = 80;
Schicht3 = 120;
figure;
contourslice(real(Head),[],[],[Schicht1,Schicht2,Schicht3], 20);
grid on;
Ueberschrift = sprintf("Head Schicht=%i, %i, %i", Schicht1,Schicht2,Schicht3);
title(Ueberschrift);
view ([45 ,40]);

```

Zubal.m

```

clear all;          % clear variables window
clc;                % clear command window display
close all;          % close all figures
M=input('\nEingabe von M: ');
N = M * M;          % M x M Unbekannte, (M+2) x (M+2) Feldpunkte
N1 = N-M;
frequ = 2.5e9;
fprintf('M = %i      f = %i Hz', M, frequ);
fprintf("\nStartzeit: %s", datetime('now'));
epsrAir = 1;
sigmaAir = 0;
epsrSkinDry = 37.952;
sigmaSkinDry = 1.4876;
epsrSkinWet = 42.783;
sigmaSkinWet = 1.6223;
epsrFat = 5.2749;
sigmaFat = 0.10672;
epsrSkull = 11.352;
sigmaSkull = 0.40411;
epsrSpongyBone = 18.491;
sigmaSpongyBone = 0.82286;
epsrWhiteMatter = 36.107;
sigmaWhiteMatter = 1.2403;
epsrGrayMatter = 48.83;
sigmaGrayMatter = 1.843;
epsrBlood = 58.181;
sigmaBlood = 2.5878;
epsrCSF = 66.168;
sigmaCSF = 3.5041;
epsrDura = 41.971;
sigmaDura = 1.6983;
epsrBoneMarrow = 5.2914;
sigmaBoneMarrow = 0.097266;
epsrCerebellum = 44.716;
sigmaCerebellum = 2.1342;
epsrSpinalCord = 30.095;
sigmaSpinalCord = 1.1093;
epsrEyeTissue = 52.558;
sigmaEyeTissue = 2.0702;
epsrEyeLense = 44.57;
sigmaEyeLense = 1.5347;
epsrCartilage = 38.663;
sigmaCartilage = 1.7949;
epsrMuscle = 52.668;
sigmaMuscle = 1.773;
epsrMuscleTongue = 52.558;
sigmaMuscleTongue = 1.8396;
epsrMuscleTrachea = 39.676;
sigmaMuscleTrachea = 1.4774;
epsrStomach = 62.078;
sigmaStomach = 2.2546;
eps0 = 1.0 / (36.0e9 * pi);
my0 = 4.0 * 1e-7 * pi;
wlang = 1.0/(sqrt(eps0*my0)*frequ);
delta = 1.1E-3;      % Pixelgröße in x, y-Ebene

```

```

ABST = delta * M;          % Abstand Start - Reflex-Ebene
fprintf('\nVoxel = %.4f m      Abstand Start - Reflex-Ebene: %.2f m', delta, ABST);
epsAir = eps0*epsrAir + sigmaAir/(2j*pi*frequ);
epsSkinDry = eps0*epsrSkinDry + sigmaSkinDry/(2j*pi*frequ);
epsSkinWet = eps0*epsrSkinWet + sigmaSkinWet/(2j*pi*frequ);
epsFat = eps0*epsrFat + sigmaFat/(2j*pi*frequ);
epsSkull = eps0*epsrSkull + sigmaSkull/(2j*pi*frequ);
epsSpongyBone = eps0*epsrSpongyBone + sigmaSpongyBone/(2j*pi*frequ);
epsWhiteMatter = eps0*epsrWhiteMatter + sigmaWhiteMatter/(2j*pi*frequ);
epsGrayMatter = eps0*epsrGrayMatter + sigmaGrayMatter/(2j*pi*frequ);
epsBlood = eps0*epsrBlood + sigmaBlood/(2j*pi*frequ);
epsCSF = eps0*epsrCSF + sigmaCSF/(2j*pi*frequ);
epsDura = eps0*epsrDura + sigmaDura/(2j*pi*frequ);
epsBoneMarrow = eps0*epsrBoneMarrow + sigmaBoneMarrow/(2j*pi*frequ);
epsCerebellum = eps0*epsrCerebellum + sigmaCerebellum/(2j*pi*frequ);
epsSpinalCord = eps0*epsrSpinalCord + sigmaSpinalCord/(2j*pi*frequ);
epsEyeTissue = eps0*epsrEyeTissue + sigmaEyeTissue/(2j*pi*frequ);
epsEyeLense = eps0*epsrEyeLense + sigmaEyeLense/(2j*pi*frequ);
epsCartilage = eps0*epsrCartilage + sigmaCartilage/(2j*pi*frequ);
epsMuscle = eps0*epsrMuscle + sigmaMuscle/(2j*pi*frequ);
epsMuscleTongue = eps0*epsrMuscleTongue + sigmaMuscleTongue/(2j*pi*frequ);
epsMuscleTrachea = eps0*epsrMuscleTrachea + sigmaMuscleTrachea/(2j*pi*frequ);
epsStomach = eps0*epsrStomach + sigmaStomach/(2j*pi*frequ);
fid=fopen('det_head_u2med.dat','r');
a_read=fread(fid);
% for J=1:128
%     found = 0;
%     for I=1:256*256*128
%         if (a_read(I) == J)
%             found = 1;
%             continue;
%         end
%     end
%     if (found == 0)
%         fprintf("%i, ", J);
%     end
% end
fclose(fid);
a=reshape(a_read,[256 256 128]); % Reshape column vector into Matrix and Pages format
[m,n,p] = size(a);           % determine size of 3D volume array
a = rot90(a,3);              % um 3*90° gegen Uhrzeiger drehen
% Reverse Slices to have Top Down view
a_reverse= ones(m,n,p); % to define matrix dimensions
q = 1;
for k=128:-1:1
a_reverse(1:m,1:n, q)= a(1:m,1:n, k); % reverse matrix a only
q = q+1;
end
% a = zeros(n,m,p);          % zum Test: Körper = Air
% tissueID ersetzen
Head = changem(a_reverse,[epsAir epsAir epsAir],[0 15 104]);
Head = changem(Head,[epsSkinDry],[1]);
Head = changem(Head,[epsSkinWet],[72]);
Head = changem(Head,[epsFat epsFat],[22 98]);
Head = changem(Head,[epsSkull epsSkull epsSkull epsSkull epsSkull epsSkull epsSkull epsSkull],
[4 5 70 71 76 81 82 99 102 125]);
Head = changem(Head,[epsSpongyBone],[100]);
Head = changem(Head,[epsWhiteMatter epsWhiteMatter epsWhiteMatter],[83 107 111]);
Head = changem(Head,[epsGrayMatter epsGrayMatter epsGrayMatter epsGrayMatter epsGrayMatter
epsGrayMatter epsGrayMatter epsGrayMatter epsGrayMatter epsGrayMatter epsGrayMatter
epsGrayMatter epsGrayMatter epsGrayMatter],[89 95 96 101 103 105 106 108 109 112 114 117 118
124]);
Head = changem(Head,[epsBlood],[23]);
Head = changem(Head,[epsCSF epsCSF epsCSF epsCSF epsCSF],[2 75 92 115 122]);
Head = changem(Head,[epsDura epsDura epsDura],[84 97 113]);
Head = changem(Head,[epsBoneMarrow],[26]);
Head = changem(Head,[epsCerebellum],[77]);
Head = changem(Head,[epsSpinalCord epsSpinalCord epsSpinalCord epsSpinalCord],[3
85 91 120 123]);
Head = changem(Head,[epsEyeTissue epsEyeTissue epsEyeTissue epsEyeTissue],[74 110 116 119]);
Head = changem(Head,[epsEyeLense],[121]);
Head = changem(Head,[epsCartilage],[30]);
Head = changem(Head,[epsMuscle],[9]);
Head = changem(Head,[epsMuscleTongue],[78]);
Head = changem(Head,[epsMuscleTrachea],[29]);
Head = changem(Head,[epsStomach],[16]);

```

```

E = zeros(m,n);
for Schicht=6:4:124
    HeadSlice = Head([1:m],[1:n],Schicht);
    for I=1:m
        for J=1:n
            E(I, J) = abs(HeadSlice(I,J));
        end
    end
    %surf(E)
    contour(E)
    axis equal;
    Ueberschrift = sprintf("Head Schicht=%i", Schicht);
    title(Ueberschrift);
    xlabel('Rasterpunkte des Testkörpers, Rastergröße: 1.1mm x 1.1mm');
    ylabel('Rasterpunkte des Testkörpers, Rastergröße: 1.1mm x 1.1mm');
    ax = gca;
    ax.FontSize = 14;
    zlabel('relative units');
    w = waitforbuttonpress;
end
Schicht = 50;
HeadSlice = Head([1:m],[1:n],Schicht);
for I=1:m
    for J=1:n
        E(I, J) = abs(HeadSlice(I,J));
    end
end
contour(E)
xlabel('Pixel x-Richtung');
ylabel('Pixel y-Richtung');
axis equal;
Ueberschrift = sprintf("Head Schicht=%i", Schicht);
title(Ueberschrift);
xlabel('Rasterpunkte des Testkörpers, Rastergröße: 1.1mm x 1.1mm');
ylabel('Rasterpunkte des Testkörpers, Rastergröße: 1.1mm x 1.1mm');
ax = gca;
ax.FontSize = 14;
zlabel('relative units');
% Reverse Slices to have Top Down view
Kopf = zeros(m,n,p);
q = 1;
% for k=p:-1:1 % Reverse Slices to have Top Down view
for k=1:1:p % just copy
    Kopf(1:m,1:n, q)= abs(Head(1:m,1:n, k));
    q = q+1;
end
%Kopf = rot90(Kopf,3); % um 3*90° gegen Uhrzeiger drehen
figure;
isosurface(Kopf); % fast but rough 3-D view
% figure;
% Using smooth filter
% isosurface(smooth3(Kopf)); % slow but fine 3-D view
title('Zubal Head Phantom Plot','FontSize',12,'FontWeight','bold');
xlabel('X-axis Pixels','FontSize',12,'FontWeight','bold');
ylabel('Y-axis Pixels','FontSize',12,'FontWeight','bold');
zlabel('Slice No','FontSize',12,'FontWeight','bold');
axis equal;
% view(2); %2-D view
BODY_X = m;
BODY_Y = n;
ABSTAND_X = floor((M-BODY_X)/2);
ABSTAND_Y = 20;
MxM = zeros(M,M);
for I=1:M
    for J=1:M
        if ((J > ABSTAND_X) && (J <= ABSTAND_X+BODY_X) && (I > ABSTAND_Y) && (I <=
(ABSTAND_Y+BODY_Y)))
            MxM(I,J) = HeadSlice(I-ABSTAND_Y,J-ABSTAND_X);
        else
            MxM(I,J) = epsAir;
        end
    end
end
end
figure;
contour(abs(MxM))
title('M x M - Feld');

```

```

grid on;
axis equal;
ylabel({'500 Rasterpunkte entsprechen 0.55m', 'Wellenausbreitung --->'});
xlabel('500 Rasterpunkte entsprechen 0.55m');
zlabel('relative units');
ax = gca;
ax.FontSize = 14;
%c = colorbar;
%c.Label.String = 'Verlustleistungsdichte (relative units)';
fprintf("\nZwischenzeit: %s (MxM gebildet)", datetime('now'));
% Lösungsversuch mit schwachbesetzte Matrix
S = zeros(N, 5);
S = complex(S, 0);
d = [-M;-1;0;1;M];
ABSTAND_X = floor((M-BODY_X)/2);
ABSTAND_Y = 70;
for I=1:N
    if (I<=N1)
        S(I,1) = 1;
    end
    if (mod(I,M))
        S(I,2) = 1;
    end
    if (mod(I-1,M))
        S(I,4) = 1;
    end
    if (I>M)
        S(I,5) = 1;
    end
    II = floor((I-1)/M) + 1;
    JJ = mod((I-1),M) + 1;
    if ((JJ > ABSTAND_X) && (JJ <= ABSTAND_X+BODY_X) && (II > ABSTAND_Y) && (II <=
(ABSTAND_Y+BODY_Y)))
        S(I,3) = (2.0*pi*frequ*delta)^2 * my0*HeadSlice(II-ABSTAND_Y, JJ-ABSTAND_X) - 4.0;
    else
        S(I,3) = (2.0*pi*frequ*delta)^2 * my0*epsAir - 4.0;
    end
end
A = spdiags(S,d,N,N);
fprintf("\nZwischenzeit: %s (sparse Matrix A ist angelegt)", datetime('now'));
B = zeros(N,1);
B = complex(B, 0);
Test = zeros(M,1);
Z1 = 1;
Z2 = 1;
for I=1:N
    U = floor((I-1)/M) + 2;
    V = mod(I, M);
    if (U == 2)
        B(I,1) = -sin(ABST*2.0*pi/wlang); % Startebene
        % fprintf("Z2=%i I=%i B(I,1)=%f\n", Z2, I, B(I,1))
        Z2 = Z2 + 1;
    end
    if (U == M+1)
        B(I,1) = 0; % leitende Ebene
        % fprintf("Z2=%i I=%i B(I,1)=%f\n", Z2, I, B(I,1))
        Z2 = Z2+1;
    end
    if ((V == 1) || (V == 0)) % j=1 oder j = jmax
        B(I,1) = B(I,1) - sin(2.0*pi*delta/wlang*(M-floor((I-1)/M)));
        if (V == 1)
            Test(Z1,1) = B(I,1);
            Z1 = Z1+1;
        end
    end
end
end
fprintf("\nZwischenzeit: %s (Matrix B ist angelegt)", datetime('now'));
figure;
plot(Test)
title('anregende Welle');
grid on;
xlabel('in y-Richtung');
% Lösungsansatz basierend auf Matrix A und B, Lösung durch MATLAB
clear X;
X = A \ B;
for I=1:M

```

```

        for J=1:M
            E(I, J) = abs(X((I-1)*M+J));
        end
    end
figure;
contour(E)
title('Elektr.Feldstärke');
grid on;
ylabel('Wellenausbreitung -->');
axis equal;
for I=1:M
    for J=1:M
        E(I, J) = abs(E(I,J) * E(I,J) * imag(MxM(I,J)) * 2*pi*frequ);
    end
end
figure;
h = surf(E)
set(h, 'LineStyle', 'none');
title('Verlustleistungsdichte');
grid on;
%ylabel({'500 Rasterpunkte entsprechen 0.55m', 'Wellenausbreitung --->'});
xlabel('500 Rasterpunkte entsprechen 0.55m');
zlabel('relative units');
ax = gca;
ax.FontSize = 14;
c = colorbar;
c.Label.String = 'Verlustleistungsdichte (relative units)';
figure;
hold on;
contour(E);
title('Verlustleistungsdichte');
grid on;
%ylabel({'500 Rasterpunkte entsprechen 0.55m', 'Wellenausbreitung --->'});
xlabel('500 Rasterpunkte entsprechen 0.55m');
zlabel('relative units');
ax = gca;
ax.FontSize = 14;
c = colorbar;
c.Label.String = 'Verlustleistungsdichte (relative units)';
fprintf("\nEndzeit: %s\n", datetime('now'));
memory
return;

```

SchichtenZubal.m

```

clear all;          % clear variables window
clc;                % clear command window display
close all;          % close all figures
Faktor = 1;
M = 500 * Faktor;    % M x M Unbekannte, (M+2) x (M+2) Feldpunkte
N = M * M;
N1 = N-M;
frequ = 2.50E+09;
epsrAir = 1.0000;
sigmaAir = 0.0000;
epsrSkinDry = 37.9520;
sigmaSkinDry = 1.4876;
epsrSkinWet = 42.7830;
sigmaSkinWet = 1.6223;
epsrFat = 5.2749;
sigmaFat = 0.1067;
epsrSkull = 11.3520;
sigmaSkull = 0.4041;
epsrSpongyBone = 18.4910;
sigmaSpongyBone = 0.8229;
epsrWhiteMatter = 36.1070;
sigmaWhiteMatter = 1.2403;
epsrGrayMatter = 48.8300;
sigmaGrayMatter = 1.8430;
epsrBlood = 58.1810;
sigmaBlood = 2.5878;
epsrCSF = 66.1680;
sigmaCSF = 3.5041;
epsrDura = 41.9710;
sigmaDura = 1.6983;
epsrBoneMarrow = 5.2914;
sigmaBoneMarrow = 0.0973;

```

```

epsrCerebellum = 44.7160;
sigmaCerebellum = 2.1342;
epsrSpinalCord = 30.0950;
sigmaSpinalCord = 1.1093;
epsrEyeTissue = 52.5580;
sigmaEyeTissue = 2.0702;
epsrEyeLense = 44.5700;
sigmaEyeLense = 1.5347;
epsrCartilage = 38.6630;
sigmaCartilage = 1.7949;
epsrMuscle = 52.6680;
sigmaMuscle = 1.7730;
epsrMuscleTongue = 52.5580;
sigmaMuscleTongue = 1.8396;
epsrMuscleTrachea = 39.6760;
sigmaMuscleTrachea = 1.4774;
epsrStomach = 62.0780;
sigmaStomach = 2.2546;
eps0 = 1.0 / (36.0e9 * pi);
my0 = 4.0 * 1e-7 * pi;
wlang = 1.0/(sqrt(eps0*my0)*frequ);
delta = 1.1E-3 / Faktor; % Pixelgröße in x, y-Ebene
ABST = delta * M; % Abstand Start - Reflex-Ebene
fprintf('M = %i f = %i Hz', M, frequ);
fprintf('\nStartzeit: %s', datetime('now'));
fprintf('\nVoxel = %.3e m Abstand Start - Reflex-Ebene: %.2f m wlang = %.4f m', delta,
ABST, wlang);
epsAir = eps0*epsrAir + sigmaAir/(2j*pi*frequ);
epsSkinDry = eps0*epsrSkinDry + sigmaSkinDry/(2j*pi*frequ);
epsSkinWet = eps0*epsrSkinWet + sigmaSkinWet/(2j*pi*frequ);
epsFat = eps0*epsrFat + sigmaFat/(2j*pi*frequ);
epsSkull = eps0*epsrSkull + sigmaSkull/(2j*pi*frequ);
epsSpongyBone = eps0*epsrSpongyBone + sigmaSpongyBone/(2j*pi*frequ);
epsWhiteMatter = eps0*epsrWhiteMatter + sigmaWhiteMatter/(2j*pi*frequ);
epsGrayMatter = eps0*epsrGrayMatter + sigmaGrayMatter/(2j*pi*frequ);
epsBlood = eps0*epsrBlood + sigmaBlood/(2j*pi*frequ);
epsCSF = eps0*epsrCSF + sigmaCSF/(2j*pi*frequ);
epsDura = eps0*epsrDura + sigmaDura/(2j*pi*frequ);
epsBoneMarrow = eps0*epsrBoneMarrow + sigmaBoneMarrow/(2j*pi*frequ);
epsCerebellum = eps0*epsrCerebellum + sigmaCerebellum/(2j*pi*frequ);
epsSpinalCord = eps0*epsrSpinalCord + sigmaSpinalCord/(2j*pi*frequ);
epsEyeTissue = eps0*epsrEyeTissue + sigmaEyeTissue/(2j*pi*frequ);
epsEyeLense = eps0*epsrEyeLense + sigmaEyeLense/(2j*pi*frequ);
epsCartilage = eps0*epsrCartilage + sigmaCartilage/(2j*pi*frequ);
epsMuscle = eps0*epsrMuscle + sigmaMuscle/(2j*pi*frequ);
epsMuscleTongue = eps0*epsrMuscleTongue + sigmaMuscleTongue/(2j*pi*frequ);
epsMuscleTrachea = eps0*epsrMuscleTrachea + sigmaMuscleTrachea/(2j*pi*frequ);
epsStomach = eps0*epsrStomach + sigmaStomach/(2j*pi*frequ);
fid=fopen('det_head_u2med.dat', 'r');
a_read=fread(fid);
fclose(fid);
a=reshape(a_read,[256 256 128]); % Reshape column vector into Matrix and Pages format
[m,n,p] = size(a); % determine size of 3D volume array
a = rot90(a,3); % um 3*90° gegen Uhrzeiger drehen
% Reverse Slices to have Top Down view
a_reverse= ones(m,n,p); % to define matrix dimensions
q = 1;
for k=128:-1:1
    a_reverse (1:m,1:n, q)= a(1:m,1:n, k); % reverse matrix a only
    q = q+1;
end
Schichten = zeros(256,256,6);
figure;
hold on;
view(-20,7);
point = [130 60 40];
normal = [0 0 100];
plot3(point(1),point(2),point(3),'or','MarkerFaceColor','r');
plot3(point(1)+[0 normal(1)], point(2)+[0 normal(2)], point(3)+[0 normal(3)],'-
b','MarkerFaceColor','b');
SchichtenAnzahl = 1;
for slice = 10:20:110
    point1 = [20 40 slice];
    [Schichten(:,:,SchichtenAnzahl), x, y, z] = obliqueslice(a_reverse, point1, normal);
    surf(x, y, z, Schichten(:,:,SchichtenAnzahl),'EdgeColor','None','HandleVisibility','off');
    Slice(SchichtenAnzahl) = slice;

```

```

        SchichtenAnzahl = SchichtenAnzahl + 1;
    end
    xlabel('Rastergröße: 1.1mm x 1.1mm');
    ax = gca;
    ax.FontSize = 14;
    zlabel('Schicht');
    SchichtenAnzahl = SchichtenAnzahl -1;
    VerlustLeistungsDichte = zeros(M,M,SchichtenAnzahl);
    for SchichtenZaehler = 1:SchichtenAnzahl
        fprintf("\nSchichtenzaehler: %i", SchichtenZaehler);
        % tissueID ersetzen
        HeadSlice = changem(Schichten(:, :, SchichtenZaehler), [epsAir epsAir epsAir], [0 15 104]);
        HeadSlice = changem(HeadSlice, [epsSkinDry], [1]);
        HeadSlice = changem(HeadSlice, [epsSkinWet], [72]);
        HeadSlice = changem(HeadSlice, [epsFat epsFat], [22 98]);
        HeadSlice = changem(HeadSlice, [epsSkull epsSkull epsSkull epsSkull epsSkull epsSkull
epsSkull epsSkull epsSkull epsSkull], [4 5 70 71 76 81 82 99 102 125]);
        HeadSlice = changem(HeadSlice, [epsSpongyBone], [100]);
        HeadSlice = changem(HeadSlice, [epsWhiteMatter epsWhiteMatter epsWhiteMatter], [83 107
111]);
        HeadSlice = changem(HeadSlice, [epsGrayMatter epsGrayMatter epsGrayMatter epsGrayMatter
epsGrayMatter epsGrayMatter epsGrayMatter epsGrayMatter epsGrayMatter epsGrayMatter
epsGrayMatter epsGrayMatter epsGrayMatter epsGrayMatter], [89 95 96 101 103 105 106 108 109 112
114 117 118 124]);
        HeadSlice = changem(HeadSlice, [epsBlood], [23]);
        HeadSlice = changem(HeadSlice, [epsCSF epsCSF epsCSF epsCSF epsCSF], [2 75 92 115 122]);
        HeadSlice = changem(HeadSlice, [epsDura epsDura epsDura], [84 97 113]);
        HeadSlice = changem(HeadSlice, [epsBoneMarrow], [26]);
        HeadSlice = changem(HeadSlice, [epsCerebellum], [77]);
        HeadSlice = changem(HeadSlice, [epsSpinalCord epsSpinalCord epsSpinalCord epsSpinalCord
epsSpinalCord], [3 85 91 120 123]);
        HeadSlice = changem(HeadSlice, [epsEyeTissue epsEyeTissue epsEyeTissue epsEyeTissue], [74
110 116 119]);
        HeadSlice = changem(HeadSlice, [epsEyeLense], [121]);
        HeadSlice = changem(HeadSlice, [epsCartilage], [30]);
        HeadSlice = changem(HeadSlice, [epsMuscle], [9]);
        HeadSlice = changem(HeadSlice, [epsMuscleTongue], [78]);
        HeadSlice = changem(HeadSlice, [epsMuscleTrachea], [29]);
        HeadSlice = changem(HeadSlice, [epsStomach], [16]);
        %figure;
        %contour(real(HeadSlice));
        %title('Slice in Image Plane');
        %xlabel('Pixel x-Richtung');
        %ylabel('Pixel y-Richtung');
        %axis equal;
        BODY_X = m;
        BODY_Y = n;
        ABSTAND_X = floor((M-BODY_X*Faktor)/2);
        ABSTAND_Y = 70*Faktor;
        MxM = zeros(M,M);
        for I=1:M
            for J=1:M
                if ((J > ABSTAND_X) && (J <= ABSTAND_X+BODY_X*Faktor) && (I > ABSTAND_Y) && (I <=
(ABSTAND_Y+BODY_Y*Faktor)))
                    I1 = round((I-ABSTAND_Y)/Faktor);
                    if (I1 < 1)
                        I1 = 1;
                    end
                    J1 = round((J-ABSTAND_X)/Faktor);
                    if (J1 < 1)
                        J1 = 1;
                    end
                    MxM(I,J) = HeadSlice(I1,J1);
                else
                    MxM(I,J) = epsAir;
                end
            end
        end
        end
        figure;
        contour(abs(MxM))
        Ueberschrift = sprintf('M x M - Feld SliceNr: %i', Slice(SchichtenZaehler));
        title(Ueberschrift);
        grid on;
        axis equal;
        %ylabel('Wellenausbreitung -->');
        xlabel('Rastergröße: 1.1mm x 1.1mm');
    end
end

```

```

ax = gca;
ax.FontSize = 14;
xlabel('Schicht');
fprintf("\nZwischenzeit: %s (MxM gebildet)", datetime('now'));
% Lösungsversuch mit schwachbesetzte Matrix
S = zeros(N, 5);
S = complex(S, 0);
d = [-M;-1;0;1;M];
ABSTAND_X = floor((M-BODY_X*Faktor)/2);
ABSTAND_Y = 70*Faktor;
for I=1:N
    if (I<=N1)
        S(I,1) = 1;
    end
    if (mod(I,M))
        S(I,2) = 1;
    end
    if (mod(I-1,M))
        S(I,4) = 1;
    end
    if (I>M)
        S(I,5) = 1;
    end
    II = floor((I-1)/M) + 1;
    JJ = mod((I-1),M) + 1;
    if ((JJ > ABSTAND_X) && (JJ <= ABSTAND_X+BODY_X*Faktor) && (II > ABSTAND_Y) && (II <=
(ABSTAND_Y+BODY_Y*Faktor)))
        II1 = round((II-ABSTAND_Y)/Faktor);
        if (II1 < 1)
            II1 = 1;
        end
        JJ1 = round((JJ-ABSTAND_X)/Faktor);
        if (JJ1 < 1)
            JJ1 = 1;
        end
        S(I,3) = (2.0*pi*frequ*delta)^2 * my0*HeadSlice(II1,JJ1) - 4.0;
    else
        S(I,3) = (2.0*pi*frequ*delta)^2 * my0*epsAir - 4.0;
    end
end
fprintf("\nZwischenzeit: %s (Matrix S ist angelegt)", datetime('now'));
A = spdiags(S,d,N,N);
fprintf("\nZwischenzeit: %s (sparse Matrix A ist angelegt)", datetime('now'));
B = zeros(N,1);
B = complex(B, 0);
Test = zeros(M,1);
Z1 = 1;
Z2 = 1;
for I=1:N
    U = floor((I-1)/M) + 2;
    V = mod(I, M);
    if (U == 2)
        B(I,1) = -sin(ABST*2.0*pi/wlang); % Startebene
        % fprintf("Z2=%i I=%i B(I,1)=%f\n", Z2, I, B(I,1))
        Z2 = Z2 + 1;
    end
    if (U == M+1)
        B(I,1) = 0; % leitende Ebene
        % fprintf("Z2=%i I=%i B(I,1)=%f\n", Z2, I, B(I,1))
        Z2 = Z2+1;
    end
    if ((V == 1) || (V == 0)) % j=1 oder j = jmax
        B(I,1) = B(I,1) - sin(2.0*pi*delta/wlang*(M-floor((I-1)/M)));
        if (V == 1)
            Test(Z1,1) = B(I,1);
            Z1 = Z1+1;
        end
    end
end
end
fprintf("\nZwischenzeit: %s (Matrix B ist angelegt)\n", datetime('now'));
%figure;
%plot(Test)
%title('anregende Welle');
%grid on;
%xlabel('in y-Richtung');
% Lösungsansatz basierend auf Matrix A und B, Lösung durch MATLAB

```

```

clear X;
X = A \ B;
for I=1:M
    for J=1:M
        E(I, J) = abs(X((I-1)*M+J));
    end
end
%figure;
%contour(E)
%Ueberschrift = sprintf('Elektr.Feldstärke SliceNr: %i', Slice(SchichtenZaehler));
%title(Ueberschrift);
%grid on;
%ylabel('Wellenausbreitung -->');
%axis equal;
%figure;
%surf(E)
%Ueberschrift = sprintf('Elektr.Feldstärke SliceNr: %i', Slice(SchichtenZaehler));
%title(Ueberschrift);
%grid on;
%ylabel('Wellenausbreitung -->');
for I=1:M
    for J=1:M
        VerlustLeistungsDichte(I, J, SchichtenZaehler) = abs(E(I,J) * E(I,J) *
imag(MxM(I,J)) * 2*pi*frequ);
    end
end
figure;
surf(VerlustLeistungsDichte(:,:,SchichtenZaehler));
Ueberschrift = sprintf('Verlustleistungsdichte SliceNr: %i', Slice(SchichtenZaehler));
title(Ueberschrift);
grid on;
%ylabel('Wellenausbreitung -->');
xlabel('Rastergröße: 1.1mm x 1.1mm');
ax = gca;
ax.FontSize = 14;
xlabel('relative units');
figure;
hold on;
contour(VerlustLeistungsDichte(:,:,SchichtenZaehler));
Ueberschrift = sprintf('Verlustleistungsdichte SliceNr: %i', Slice(SchichtenZaehler));
title(Ueberschrift);
grid on;
%ylabel('Wellenausbreitung -->');
xlabel('Rastergröße: 1.1mm x 1.1mm');
ax = gca;
ax.FontSize = 14;
xlabel('relative units');
memory
end
save('saveVerlustLeistungsDichte', 'VerlustLeistungsDichte');
example = matfile('saveVerlustLeistungsDichte');
Pv = example.VerlustLeistungsDichte;
%Filename = sprintf("saveA%i.mat", M);
%save(Filename, 'A');
% example = matfile('saveA.mat');
% A = example.A;
figure;
hold on;
title('Verlustleistungsdichte');
grid on;
%ylabel('Wellenausbreitung -->');
xlabel('Rastergröße: 1.1mm x 1.1mm');
ax = gca;
ax.FontSize = 14;
xlabel('Schichtenzähler');
for SchichtenZaehler = 1:SchichtenAnzahl
    [~,h] = contour(Pv(:,:,SchichtenZaehler));
    h.ContourZLevel = SchichtenZaehler - 1;
end
view(3);
fprintf("\nEndzeit: %s\n", datetime('now'));
memory

```

HeatKreis.m

```

clear all;      % clear variables window
clc;           % clear command window display

```

```

close all;          % close all figures
TV = 20;           % Temperatur j=1
TH = 20;           % Temperatur j=M+2
TL = 20;           % Temperatur i=1
TR = 20;           % Temperatur i=M+2
LambdaAir = 0.026;
LambdaKreis = 0.5;
QAir = +0.05;
QKreis = -100;
delta = 1.1E-3;
fAir = QAir * delta * delta / LambdaAir;
fKreis = QKreis * delta * delta / LambdaKreis;
M=input('\nEingabe von M: ');
N = M * M;
N1 = N-M;
fprintf('M = %i', M);
fprintf("\nStartzeit: %s", datetime('now'));
fprintf('\nVoxel = %.4f m', delta);
[I,J] = ndgrid(1:M,1:M);
r = 50;
ic = 100;
jc = 100;
MxM = double((I-ic).^2+(J-jc).^2<=r^2);
MxM = changem(MxM,[fAir fKreis],[0 1]);
figure;
contour(MxM);
grid on;
axis equal
xlabel('x-Achse');
ylabel('y-Achse');
title('Position');
% Lösungsversuch mit schwachbesetzte Matrix
B = zeros(N,1);
S = zeros(N, 5);
d = [-M;-1;0;1;M];
for I=1:N
    if (I<=N1)
        S(I,1) = 1;
    end
    if (mod(I,M))
        S(I,2) = 1;
    end
    if (mod(I-1,M))
        S(I,4) = 1;
    end
    if (I>M)
        S(I,5) = 1;
    end
    II = floor((I-1)/M) + 2;
    JJ = mod((I-1),M) + 2;
    S(I,3) = -4;
    Rand = 0;
    if (JJ+1>M+1) Rand = Rand + TH;
    end
    if (JJ-1<2) Rand = Rand + TV;
    end
    if (II+1>M+1) Rand = Rand + TR;
    end
    if (II-1<2) Rand = Rand + TL;
    end
    B(I,1) = -Rand + MxM(floor((I-1)/M)+1,mod(I, M)+1);
end
A = spdiags(S,d,N,N);
fprintf("\nZwischenzeit: %s (sparse Matrix A ist angelegt)", datetime('now'));
X = A \ B;
for I=1:M
    for J=1:M
        E(I, J) = abs(X((I-1)*M+J));
    end
end
figure;
h = surf(E);
set(h,'LineStyle','none');
title('Temperatur');
colorbar;
grid on;

```

```
xlabel('x-Achse');
ylabel('y-Achse');
zlabel('z-Achse');
figure;
contour(E)
title('Temperatur');
grid on;
colorbar;
xlabel('x-Achse');
ylabel('y-Achse');
zlabel('z-Achse');
axis equal;
fprintf("\nEndzeit: %s\n", datetime('now'));
```