

Institut für Parallele und Verteilte Systeme

Abteilung Parallele Systeme

Universität Stuttgart
Universitätsstraße 38
D - 70569 Stuttgart

Studienarbeit Nr. 2322

**Implementierung von Algorithmen zur
MESH-Erzeugung und numerischen
Impedanzberechnung**

Alexander Braunstein

Studiengang:	Informatik
Prüfer:	Prof. Dr.-Ing. Sven Simon
Betreuer:	Dipl.-Ing. Jürgen Hillebrand
begonnen am:	02.11.2010
beendet am:	04.05.2011
CR-Klassifikation:	F.2.1, G.1.4

Abstract

In dieser Studienarbeit werden verschiedene Meshverfahren, sowie numerische Methoden zur elektromagnetischen Feldberechnung, basierend auf dem Ergebnis eines Meshings, vorgestellt. Als Vorbereitung zur Bestimmung der Leitungsimpedanz wird ein hexahedralisiertes Mesh erzeugt. Mit Hilfe dieses Meshes werden Potentiale auf diskreten Rechennetzpunkten berechnet, welche von der Finiten-Integral-Methode zur Berechnung der Leitungsimpedanz benutzt werden.

Various processes of meshing and numerical methods for electromagnetic field calculation, based on the result of a meshing process, are presented in this thesis. As preparation for the determination of the line impedance, a hexahedral mesh is generated. Potentials on the discret computational grid points are calculated with the help of this mesh. The line impedance is calculated by the Finite Integration Theory (FIT) using these potentials.

Inhaltsverzeichnis

Abstract.....	1
1. Einleitung – Motivation.....	5
1.1 Meshing.....	5
1.2 Numerische Methoden zur elektromagnetischen Feldberechnung.....	8
1.2.1 Finite-Elemente-Methode (FEM).....	9
1.2.2 Finite-Differenzen-Methode (FDM).....	9
1.2.3 Finite-Differenzen-Methode im Zeitbereich (FDTD).....	10
1.2.4 Finite-Integral-Methode (FIT).....	11
1.2.5 Finite-Volumen-Methode (FVM).....	11
2. Mathematische und physikalische Hintergründe.....	13
2.1 Laplace-Gleichung.....	13
2.1.1 Fünf-Punkte-Stern-Diskretisierung.....	14
2.2 Finite-Integral-Methode.....	16
3. 3D-ACIS-Modeler.....	19
3.1 Methode – API-Solid-Block.....	19
3.2 Methode – Api-Restore-Entity-List.....	20
3.3 Methode – Api-Clash-Bodies.....	21
4. ImpCalc.....	22
4.1 Klasse – ImportObject.....	23
4.2 Diskretisierungskorrektur.....	23
4.4 Klasse – Mesh.....	24
4.4.1 Methode – ClashTest().....	25
4.4.2 Methode – Impedance().....	26
5. Ergebnisse.....	27
6. Zusammenfassung.....	28
7. Ausblick.....	29
8. Installation und Inbetriebnahme der Software.....	30
9. Quellenverzeichnis.....	32

Abbildungsverzeichnis

Abbildung 1: Meshzellen eines hexahedralisierten und tetrahedralisierten Meshes.....	6
Abbildung 2: Hybrides Mesh.....	6
Abbildung 3: Conformal Mesh.....	7
Abbildung 4: Hexahedralisiertes Mesh.....	7
Abbildung 5: Tetrahedralisiertes Mesh.....	8
Abbildung 6: Yee-Zelle.....	10
Abbildung 7: Diskretisierung für die z-Komponente von Gleichung (2).....	11
Abbildung 8: Potential im Punkt P abhängig von seinen vier Nachbarknoten.....	13
Abbildung 9: Potentialberechnung in einem inhomogenen Medium.....	15
Abbildung 10: Finite-Integral-Methode: Geschlossener Integrationspfad.....	16
Abbildung 11: Ableitungshierarchie der Klasse „Construct Bodies“.....	20
Abbildung 12: Ableitungshierarchie der Klasse „Save and Restore“.....	21
Abbildung 13: Ableitungshierarchie der Klasse „Entity Clash Calculation“.....	21
Abbildung 14: Vorgehensweise in ImpCalc zur Berechnung der Leitungsimpedanz.....	22
Abbildung 15: Nummerierungsschema der einzelnen 3D Meshzellen im Volumenmesh...	24
Abbildung 16: Volumenmesh.....	25
Abbildung 17: 2D Mesh: Eine Scheibe des Volumenmeshes aus Abb.16.....	26
Abbildung 18: Streifenleitung.....	27
Abbildung 19: Differentielle Impedanz.....	29
Abbildung 20: Registrierung des License-Key.....	30
Abbildung 21: Anpassung von MyAcis.cpp.....	31

Tabellenverzeichnis

Tabelle 1: Auswirkung des Diskretisierungsfehlers.....	24
Tabelle 2: Ergebnisse.....	27

1. Einleitung – Motivation

Durch die stetige Weiterentwicklung der Datenübertragungstechnik mit Datenraten im Gbit/s-Bereich ist es erforderlich, die charakteristischen Eigenschaften von Datenleitungen zu bestimmen. In dieser Studienarbeit wird ein Verfahren zur Bestimmung der Leitungsimpedanz mit Hilfe von numerischen Methoden vorgestellt. Auf der Grundlage eines 3D-CAD-Modells wird ein Volumenmesh erstellt, welches das CAD-Modell in endlich viele kleine 3D Gitterzellen unterteilt. Das Volumenmesh wird mit der API des 3D-ACIS-Modeler der Firma Spatial erstellt und dient als Vorbereitung für die Berechnung der Leitungsimpedanz. Anschließend wird mit einem gegebenen, jedoch leicht modifizierten und für die Aufgabenstellung angepassten Impedanzberechnungsverfahren der Widerstand einer elektrischen Leitung bestimmt. Dieses Berechnungsverfahren gliedert sich grob in zwei Stufen. In der ersten Stufe werden, durch das Lösen einer partiellen Differentialgleichung, Potentiale berechnet. In der zweiten Stufe wird schließlich die Leitungsimpedanz mittels der Finiten-Integral-Methode bestimmt.

Während der Arbeit ist ein Programm (ImpedanceCalculator – ImpCalc) zur Berechnung der Leitungsimpedanz entstanden. Durch die Leitungsimpedanz ist die Bestimmung der Streuparameter für die *Finite Difference Time Domain* (FDTD) möglich [23]. Somit ist das hier implementierte Verfahren hilfreich für die Benutzung der FDTD.

1.1 Meshing

Mit Hilfe des Meshings wird eine Objektrepräsentation vorgenommen. Aus einem 3D Mesh, welches auch Volumenmesh genannt wird, erhält man ein 2D Mesh, indem man die durch einen Schnitt entstandene Schnittfläche des Volumens betrachtet. Für die Lösung der hier gestellten Aufgabe werden beide Meshformen herangezogen. Die Geometrie des 3D-CAD-Modells wird somit durch ein Netz von einfachen Primitiven¹ abgebildet. Die Schwierigkeit beim Meshing besteht darin, einen Kompromiss der drei Faktoren: Genauigkeit, Rechenzeit und Speicherbedarf zu finden. Häufig findet man in der Anwendung drei unterschiedliche Mesh-Typen, die sich in der Ausprägung der Gitterzellen unterscheiden. Man spricht von tetrahedralisierten, hexahedralisierten und hybriden Meshes (Abb. 1). Hybride Meshes sind eine Mischform von tetrahedralisierten und hexahedralisierten Meshes.

1 Geometrische Grundformen, wie sie in DIN EN ISO 19107 definiert sind.

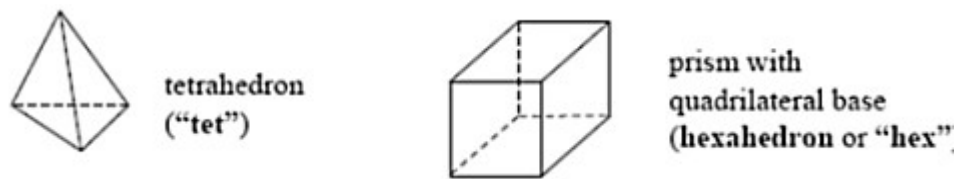


Abbildung 1: Meshzellen eines hexahedralisierten und tetrahedralisierten Meshes

[<http://www.cctech.co.in/images/CFD-FAQs/Type%20of%20CFD%20Cells.gif>]

Bei einem tetrahedralisierten Mesh besitzen die Meshzellen im 2D eine dreieckige Form. Beim Volumenmesh sind sie tetraederförmig. Somit besteht jede Meshzelle im 3D aus vier dreieckigen Flächen und vier Eckpunkten vom Grad 3². Ein solch aufgebautes Mesh wird meist als „unstrukturiert“ bezeichnet, wie in Abbildung 5 zusehen ist. Vorteilhaft ist die bessere Anpassung an komplexe Geometrien und die Möglichkeit zur automatischen Meshgenerierung.

Anders als beim tetrahedralisierten Mesh ist die Geometrie der Meshzellen bei einem hexahedralisierten Mesh im 2D viereckig. Im 3D besteht die quaderförmige Zelle aus 6 viereckigen Flächen sowie 6 Eckpunkten vom Grad 3. Aufgrund der strukturierten Form des Meshes werden 4-10 mal weniger Meshzellen benötigt um eine Geometrie zu

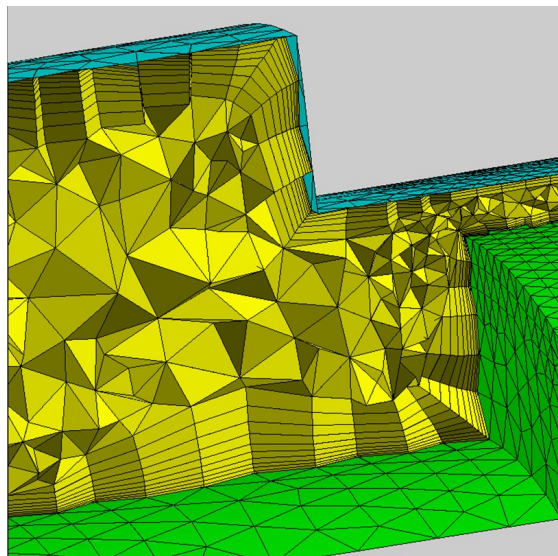


Abbildung 2:Hybrides Mesh

[http://www.centaursoft.com/cms/files/imagegallerymodule/@random45a4763109669/gallery8/Hybrid_Example1_trim.med.png]

zerlegen. Somit ist der Speicherbedarf deutlich geringer. Da sich ein hexahedralisiertes Mesh einfacher partitionieren lässt, eignet es sich besser zur parallelen Berechnung. Es kann jeder CPU ein klar strukturierter Bereich zur Berechnung zugeordnet werden. Die

2 Unter dem Grad eines Punktes wird die Anzahl der Kanten in diesem Punkt verstanden.

Zusammenführung der Teilergebnisse zum Gesamtergebnis ist durch die regelmäßige Struktur des hexahedralisierten Meshes unproblematisch.

Eine weitere erwähnenswerte Gruppe sind die Conformal Meshes, welche durch die Komprimierung von Meshzellen zum einen den Speicherbedarf und zum anderen die Berechnungszeit erheblich reduzieren. Bei einem Conformal Mesh darf an einer Schnittstelle von zwei Elementen keine Lücke vorhanden sein. Ein weiteres Merkmal für ein Conformal Mesh ist das exakte Meshing von gekrümmten Objekten. Da sich die Meshzellen an das gekrümmte Objekt anpassen, muss die Krümmung nicht mehr, wie bei den traditionellen Meshverfahren, durch viele kleine Meshzellen approximiert werden. Beispielsweise werden bei der Simulationssoftware EmPro von Agilent³, für die Finite-Differenzen-Methode im Zeitbereich (FDTD), Conformal Meshes benutzt. Berechnungsverfahren, die mit „non-conformal“ Meshes arbeiten, besitzen eine höhere Ungenauigkeit.

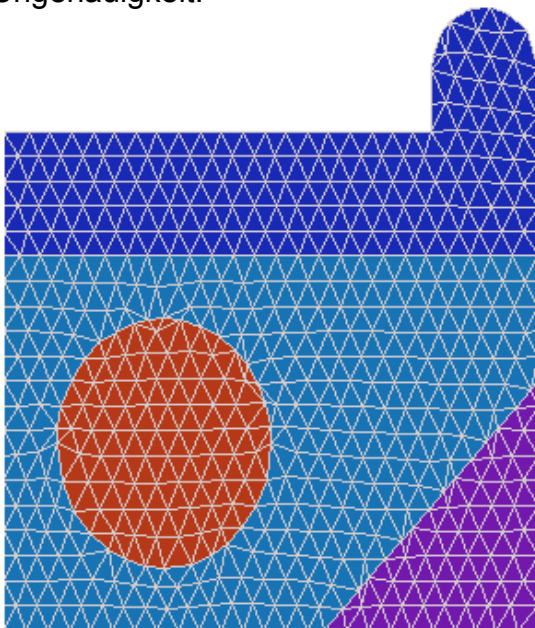


Abbildung 3: Conformal Mesh
[http://www.fieldp.com/images/mesh_illus.gif]

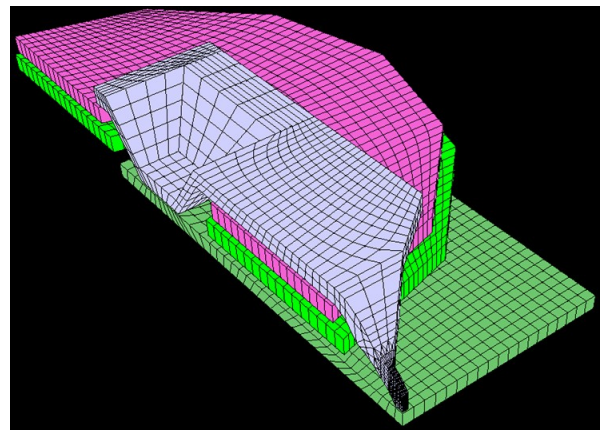


Abbildung 4: Hexahedralisiertes Mesh
[http://hbd.ist.hokudai.ac.jp/noguchi/mesh_head.png]

Auf das Resultat des Meshings, einer vereinfachten Darstellung des Objekts, können unter anderem numerische Verfahren zur elektromagnetischen Feldberechnung, wie zum Beispiel die Finite-Differenzen-Methode (siehe 1.2.2 – FDM) oder auch die Finite-Elemente-Methode (siehe 1.2.1 – FEM), welche beide eine numerische Lösung von partiellen Differentialgleichungen liefern, angewendet werden.

³ <http://www.home.agilent.com/agilent/product.jsp?cc=DE&lc=ger&ckey=1938471&nid=-34278.970421.00&id=1938471&pselect=SR.GENERAL>

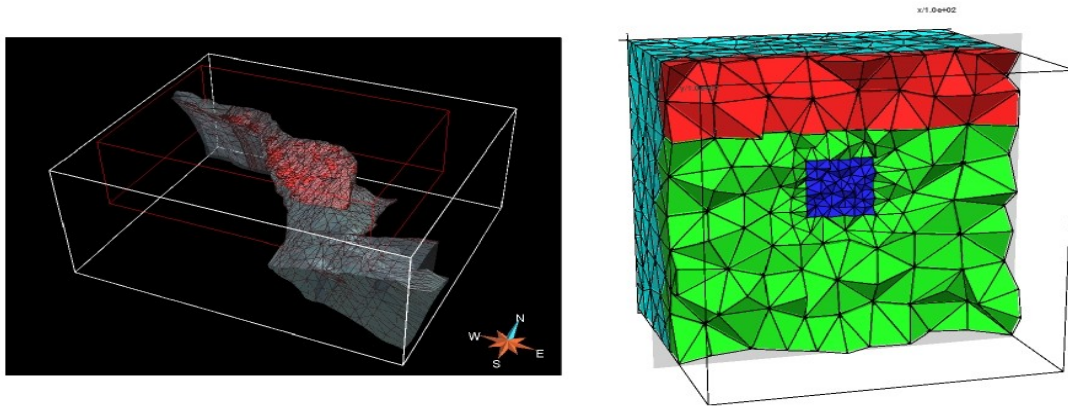


Abbildung 5: Tetrahedralisiertes Mesh

[<http://www.esd.mun.ca/~farq/images/meshes.jpg>]

1.2 Numerische Methoden zur elektromagnetischen Feldberechnung

Da es nur für sehr einfache Geometrien mit einer homogenen Materialverteilung möglich ist, das elektromagnetische Feld analytisch zu bestimmen, werden heutzutage für komplexe Geometrien mit unterschiedlichem Materialverhalten numerische Methoden zur Feldberechnung eingesetzt⁴.

Das heißt, das kontinuierliche Feldverhalten wird durch eine Diskretisierung approximiert. Die Maxwell-Gleichungen [14] in differentieller Form lassen sich durch einen verknüpfenden Integralsatz (Gauß, Stokes) in integraler Form beschreiben.

$$\operatorname{div} \vec{D} = \rho \quad \leftrightarrow \quad \oint_{\partial V} \vec{D} \cdot d\vec{A} = Q(V)$$

$$\operatorname{div} \vec{B} = 0 \quad \leftrightarrow \quad \oint_{\partial V} \vec{B} \cdot d\vec{A} = 0$$

$$\operatorname{rot} \vec{E} + \frac{\partial \vec{B}}{\partial t} = 0 \quad \leftrightarrow \quad \oint_{\partial A} \vec{E} \cdot d\vec{s} + \frac{d}{dt} \left(\iint_A \vec{B} \cdot d\vec{A} \right) = 0$$

$$\operatorname{rot} \vec{H} = \vec{j}_l + \frac{\partial \vec{D}}{\partial t} \quad \leftrightarrow \quad \oint_{\partial A} \vec{H} \cdot d\vec{s} = \iint_A \vec{j}_l \cdot d\vec{A} + \frac{d}{dt} \left(\iint_A \vec{D} \cdot d\vec{A} \right)$$

Maxwell-Gleichungen in differentieller und integraler Form.

Die Finite-Differenzen-Methode im Frequenzbereich (siehe 1.2.2 - FDM) und die Finite-Differenzen Methode im Zeitbereich (siehe 1.2.3 - FDTD) sind numerische Verfahren zur Lösung der Maxwell-Gleichungen in differentieller Form. Für die Lösung der Gleichungen in integraler Form eignen sich die Finite-Elemente-Methode und die Finite-Integral-Methode.

⁴ http://www.iem.rwth-aachen.de/index.pl/numerische_feldberechnung

1.2.1 Finite-Elemente-Methode (FEM)

Die Finite-Elemente-Methode ist das am weitesten verbreitete und am häufigsten verwendete numerische Verfahren zur Lösung von partiellen Differentialgleichungen. Die FEM findet Anwendung in den Bereichen Dynamik, Feldanalyse elektromagnetischer Felder, Wettervorhersage und vielen mehr [15]. Um eine partielle Differentialgleichung mit dem Computer lösen zu können, ist eine Diskretisierung notwendig, welche die partielle Differentialgleichung in ein algebraisches Gleichungssystem transformiert. Bei der Finiten-Elemente-Methode wird ein großer Bereich in viele kleine endliche (= finit) Elemente zerlegt. Die einzelnen Elemente sind an bestimmten Stellen, den sogenannten Knoten, miteinander verbunden. Der Vorteil ist, dass die einzelnen Bereiche parametrierbar und einfacher zu berechnen sind. Die Einzellösungen der gebietsweisen Berechnungen ergeben zusammengenommen die approximierte Gesamtlösung der partiellen Differentialgleichung [16].

1.2.2 Finite-Differenzen-Methode (FDM)

Bei der Finiten-Differenzen-Methode handelt es sich um die einfachste Methode eine gewöhnliche oder partielle Differentialgleichung zu approximieren. Wie auch die FEM ist die FDM auf zeitinvariante Differentialgleichungen anzuwenden. Somit wird ein stationärer Zustand berechnet. Für die Diskretisierung wird bei der FDM ein Gitter über das Gebiet gelegt. Die Berechnungen finden lediglich an den Gitterknoten statt. Dadurch erhält man bei der Finiten-Differenzen-Methode keine geschlossene Lösungsfunktion, sondern einzelne Lösungswerte, die nur einer Näherung entsprechen und somit fehlerbehaftet sind.

Um die Ableitungen der Differentialgleichung durch algebraische Ausdrücke zu ersetzen, bedient man sich der Taylorreihe. Die Genauigkeit dieser Approximation hängt von der Anzahl der verwendeten Taylor-Terme ab. Je mehr Taylor-Terme berücksichtigt werden, desto genauer ist die Näherung am analytischen Ergebnis. Angewendet auf die Gitterstruktur wird die Ableitung an den Gitterknoten durch Differenzen approximiert. Problematisch bei dieser Methode wird es, wenn ein zu feinmaschiges Gitternetz gewählt wird. In diesem Fall kann es zu Rundungsfehlern kommen, da die Zahlendarstellung im PC begrenzt ist. Diese Tatsache darf man nicht außer Acht lassen [17].

1.2.3 Finite-Differenzen-Methode im Zeitbereich (FDTD)

Im Gegensatz zur FDM, bei der eine Berechnung im Frequenzbereich stattfindet, ist mit der Finite-Differenzen-Methode eine Berechnung im Zeitbereich möglich. Durch eine Fast Fourier Transformation (FFT) kann das Ergebnis auch wieder in den Frequenzbereich transformiert werden. Bei der FDTD wird das Gebiet durch Elementarquader, sogenannte Yee-Zellen (Abb. 6), diskretisiert. Die FDTD ist hilfreich um dynamische Untersuchungen durchzuführen. E-Felder werden anhand diskretisierter Maxwell-Gleichungen repräsentiert. Somit werden die Maxwell'schen Gleichungen in Raum und Zeit diskretisiert. Schaut man sich unter Quellen-, und Verlustfreiheit die integrale Darstellung der ersten beiden Maxwell-Gleichungen an, erhält man folgende Beschreibung:

$$\oint \vec{H} d\vec{s} = \iint \dot{\vec{D}} d\vec{A} \quad (1)$$

$$\oint \vec{E} d\vec{s} = - \iint \dot{\vec{B}} d\vec{A} \quad (2)$$

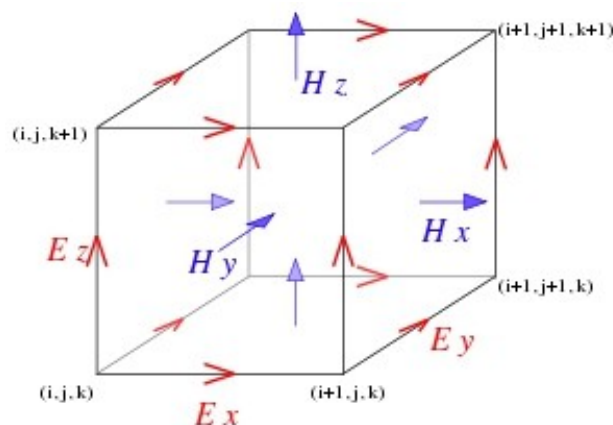


Abbildung 6: Yee-Zelle

[<http://image.absoluteastronomy.com/images/encyclopiainage/s/y/ye/yee-cube.svg.png>]

Die Materialgleichungen für ein homogenes und isotropes Medium lauten:

$$\vec{D} = \epsilon_0 \epsilon_r \vec{E}$$

$$\vec{B} = \mu_0 \mu_r \vec{H}$$

Diskretisiert man beispielsweise die Gleichung (2) in Raum und Zeit für die z-Komponente (Abb. 7), dann berechnet sich das Linienintegral durch eine Multiplikation der elektrischen Feldstärke mit der entsprechenden Kantenlänge. Das Flächenintegral berechnet sich analog mit dem Unterschied, dass die magnetische Feldstärke mit der dazugehörigen Fläche multipliziert wird [4].

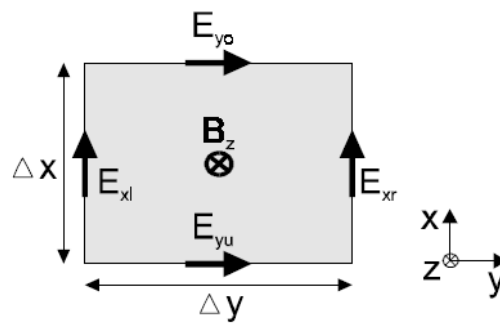


Abbildung 7: Diskretisierung für die z-Komponente von Gleichung (2).

[http://opus.kobv.de/tuberlin/volltexte/2000/118/pdf/huynh_ngoc.pdf]

Nach Durchführung der Diskretisierung von Gleichung (2) für die z-Komponente ergibt sich Gleichung (3).

$$E_{xl}^n \Delta x + E_{yo}^n \Delta y - E_{xr}^n \Delta x - E_{yu}^n \Delta y = - \frac{B_z^{(n+1/2)} - B_z^{(n-1/2)}}{\Delta t} \Delta x \Delta y \quad (3)$$

Der obere Index n steht für den Zeitpunkt, zu dem die Feldstärke berechnet wird.

1.2.4 Finite-Integral-Methode (FIT)

Anders als bei den bereits vorgestellten numerischen Verfahren, die ihren Ursprung eher in der Mechanik haben, hat die Finite-Integral-Methode ihren Ursprung in der Elektrotechnik. Somit ist sie auf die Lösung der Maxwell'schen Gleichungen regelrecht zugeschnitten. Bei der FIT wird das Gebiet in Gitterzellen eingeteilt und dementsprechend diskretisiert. Jede Gitterzelle besitzt eine Materialeigenschaft. Aufgrund des Ursprungs der FIT berechnet sie die Maxwell-Gleichungen in Integralform ohne Approximationsfehler. Zudem werden durch die Integralapproximation die Materialgleichungen diskretisiert berechnet. Anwendung findet die Finite-Integral-Methode sowohl im Frequenzbereich als auch im Zeitbereich. Im Zeitbereich wird das Gebiet durch duale, zueinander versetzte Gitter diskretisiert, damit die Wechselwirkung von elektrischem und magnetischem Feld berücksichtigt werden kann [18].

1.2.5 Finite-Volumen-Methode (FVM)

Die Finite-Volumen-Methode ist ein numerisches Verfahren zur Lösung von partiellen Differentialgleichungen basierend auf einem Erhaltungssatz, z.B. dem Energieerhaltungssatz [19]. Das kommt daher, weil der Ursprung der FVM in der

Strömungsmechanik liegt. Durch die von Shankar et. al. [1] beschriebene Methode, ist die FVM auch zur Berechnung von E-Feldern verwendbar. Die Vorgehensweise der Finiten-Volumen-Methode gliedert sich in drei Schritte [5]:

Gebietszerlegung:

Das Gebiet wird in finite 3D Gitterzellen zerlegt. Dabei fällt das lückenlose Gitternetz mit den Rändern der zu Grunde liegenden Volumengeometrie zusammen. Durch die Gebietszerlegung besteht das Volumen aus vielen kleinen Kontrollvolumina⁵.

Diskretisierung:

Ähnlich wie bei der Finiten-Elemente-Methode wird auch bei der Finten-Volumen-Methode eine stückweise lineare Funktion angewendet, nur mit dem Unterschied, dass die Funktion dieses Mal nicht auf die Knoten, sondern auf die Gitterzellen angewandt wird. Jedes Kontrollvolumen besitzt einen Berechnungspunkt, welcher für das komplette Kontrollvolumen gültig ist.

Lösung des diskretisierten Gleichungssystems:

Um das entstandene algebraische Gleichungssystem zu lösen, wird entweder ein direktes oder ein iteratives Lösungsverfahren angewandt. Dementsprechende Verfahren sind in [2] und [3] ausführlich erläutert.

Ein großer Vorteil ist, dass bei diesem Verfahren unstetige Lösungen kein Problem darstellen. Da man in der Strömungsmechanik häufig mit unstetigen Lösungen in Kontakt tritt, eignet sich die FVM besonders gut zur Berechnung. Zudem bieten die wenigen Anforderungen an die Gitterzellen eine große geometrische Flexibilität. Lösungen mit großen Feldgradienten numerisch robust berechnen zu können, ist ein weiterer Vorzug der FVM.

⁵ Ein Volumen wird in kleine finite Kontrollvolumina zerlegt, wobei diese Volumina das Gebiet ohne Lücken oder Überschneidungen vollständig überdecken sollen [5].

2. Mathematische und physikalische Hintergründe

Die Berechnung der Leitungsimpedanz⁶ Z_L setzt zunächst die Berechnung der Potentiale V auf den Gitternetzknotten voraus. Diese werden durch die Laplace-Gleichung

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial z^2} = 0 \quad \text{bestimmt. Des Weiteren wird mit der Finiten-Integral-Methode ein}$$

Kurvenintegral über die elektrische Feldstärke einer Gitterzelle berechnet, woraus mit Hilfe der Maxwell'schen Gleichungen - im stationären Fall - die Impedanz abgeleitet werden kann.

2.1 Laplace-Gleichung

Damit die partielle Differentialgleichung zweiter Ordnung mit einem Computer berechnet werden kann, ist eine Diskretisierung nach dem Ort notwendig. Die hier verwendete Diskretisierung wird auch Fünf-Punkte-Stern-Diskretisierung genannt [20]. Das heißt, das Potential eines Gitternetzknottes wird in Abhängigkeit von seinen vier Nachbarknoten berechnet (Abb. 8).

Nach der Ortsdiskretisierung wird die algebraische Gleichung iterativ gelöst. Das Abbruchkriterium wird so gewählt, dass sich bei der späteren Aufsummierung der Potentiale zum Umlaufintegral (Ladung) eine bestimmte Genauigkeit ergibt, die durch eine Variable bestimmt wird.

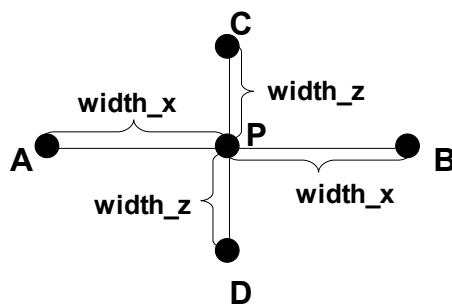


Abbildung 8: Potential im Punkt P abhängig von seinen vier Nachbarknoten

⁶ Die Leitungsimpedanz ist der Widerstand, den eine Leitung der Ausbreitung einer elektromagnetischen Welle entgegenbringt [12].

2.1.1 Fünf-Punkte-Stern-Diskretisierung

Für die Diskretisierung der Laplace-Gleichung wird auf das Paper von Green [6] verwiesen. Jedoch wurde im Folgenden die Diskretisierung für die Potentialberechnung auf nicht quadratische Gitterzellen modifiziert. Das heißt, die Gitterzellen sind zunächst in der Breite und Höhe variabel. Jedoch müssen alle Gitterzellen des Gitternetzes dieselbe Größe haben.

Für eine numerische Lösung der Laplace-Gleichung bedient man sich der Taylorreihe (1). Diese wird nach der zweiten Ableitung abgebrochen und liefert somit eine gute Approximation der partiellen Differentialgleichung.

$$f(x \pm \Delta x) = f(x) \pm \Delta x f'(x) + \frac{(\Delta x)^2}{2!} f''(x) \pm \frac{(\Delta x)^3}{3!} f'''(x) + \dots \quad (1)$$

Wie bereits erwähnt bezieht sich die folgende Diskretisierung auf äquidistante Gitterzellen mit zunächst frei wählbaren Gitterzellenweiten in x-, und z-Richtung (siehe Abb. 8).

Diskretisierung in x-Richtung:

$$V_B - V_P = \text{width}_x \frac{\partial V}{\partial x} + \frac{\text{width}_x^2}{2!} \frac{\partial^2 V}{\partial x^2} + \frac{\text{width}_x^3}{3!} \frac{\partial^3 V}{\partial x^3} + \frac{\text{width}_x^4}{4!} \frac{\partial^4 V}{\partial x^4} + \dots \quad (2)$$

$$V_A - V_P = -\text{width}_x \frac{\partial V}{\partial x} + \frac{\text{width}_x^2}{2!} \frac{\partial^2 V}{\partial x^2} - \frac{\text{width}_x^3}{3!} \frac{\partial^3 V}{\partial x^3} + \frac{\text{width}_x^4}{4!} \frac{\partial^4 V}{\partial x^4} - \dots \quad (3)$$

(2) + (3):

$$\begin{aligned} \frac{\partial^2 V}{\partial x^2} &= \frac{V_A + V_B - 2V_P}{\text{width}_x^2} - \frac{\text{width}_x^2}{12} \frac{\partial^4 V}{\partial x^4} - \dots \\ \frac{\partial^2 V}{\partial x^2} &\simeq \frac{V_A + V_B - 2V_P}{\text{width}_x^2} \end{aligned} \quad (4)$$

analoge Diskretisierung in z-Richtung:

$$\begin{aligned} \frac{\partial^2 V}{\partial z^2} &= \frac{V_C + V_D - 2V_P}{\text{width}_z^2} - \frac{\text{width}_z^2}{12} \frac{\partial^4 V}{\partial z^4} - \dots \\ \frac{\partial^2 V}{\partial z^2} &\simeq \frac{V_C + V_D - 2V_P}{\text{width}_z^2} \end{aligned} \quad (5)$$

(4) und (5) in die Laplace-Gleichung einsetzen:

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial z^2} = \frac{(V_A + V_B - 2V_P) \cdot \text{width}_z^2 + (V_C + V_D - 2V_P) \cdot \text{width}_x^2}{\text{width}_x^2 \cdot \text{width}_z^2} = 0$$

$$\rightarrow V_P = \frac{\text{width}_z^2 \cdot (V_A + V_B) + \text{width}_x^2 \cdot (V_C + V_D)}{2 \cdot (\text{width}_x^2 + \text{width}_z^2)} \quad (6)$$

Gleichung (6) beschreibt die Berechnungsvorschrift für das Potential V eines Punktes P in einem homogenen Medium unter Berücksichtigung seiner 4 Nachbarknoten.

Je feiner das Gitternetz gewählt wird, desto genauer entspricht die numerische Lösung der analytischen Lösung. Allerdings steigt der Rechenaufwand mit höheren Auflösungen immens an.

Für die Berechnung des Potentials V im Punkt P eines inhomogenen Mediums müssen die verschiedenen Dielektrika der Gitterzellen berücksichtigt werden. Deshalb sind die Epsilons an den Grenzflächen mit in die Gleichung (6) einzubeziehen (Abb. 9). Somit ergibt sich Gleichung (7):

$$\epsilon_{12} = \frac{\epsilon_1 + \epsilon_2}{2}; \quad \epsilon_{13} = \frac{\epsilon_1 + \epsilon_3}{2}; \quad \epsilon_{24} = \frac{\epsilon_2 + \epsilon_4}{2}; \quad \epsilon_{34} = \frac{\epsilon_3 + \epsilon_4}{2}$$

$$V_P = \frac{V_A \cdot \text{width}_z^2 \cdot \epsilon_{13}}{(\text{width}_x^2 + \text{width}_z^2) \cdot (\epsilon_{13} + \epsilon_{24})} + \frac{V_B \cdot \text{width}_z^2 \cdot \epsilon_{24}}{(\text{width}_x^2 + \text{width}_z^2) \cdot (\epsilon_{13} + \epsilon_{24})} + \frac{V_C \cdot \text{width}_x^2 \cdot \epsilon_{34}}{(\text{width}_x^2 + \text{width}_z^2) \cdot (\epsilon_{12} + \epsilon_{34})} + \frac{V_D \cdot \text{width}_x^2 \cdot \epsilon_{12}}{(\text{width}_x^2 + \text{width}_z^2) \cdot (\epsilon_{12} + \epsilon_{34})} \quad (7)$$

Zur Lösung des Randwertproblems, gibt es mehrere Verfahren. Bei dem hier verwendeten Verfahren, namens Radiation Boundary, werden nach jedem Iterationsschritt die

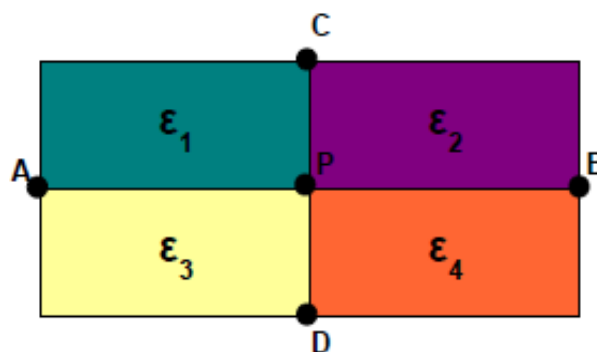


Abbildung 9: Potentialberechnung in einem inhomogenen Medium

Potentialwerte der nächstgelegenen, berechenbaren Gitternetzkonten nach „außen“ (über den Rand hinaus) kopiert. Es gilt, falls es sich bei $V(x \pm \Delta x, y \pm \Delta y)$ um einen Randwert handelt: $V(x \pm \Delta x, y \pm \Delta y) = V(x, y)$.

Bei der Spiegelladungsmethode werden den Gitternetzknotten außerhalb des berechenbaren Bereichs, die invertierten Potentialwerte der Gitternetzknotten innerhalb des berechenbaren Bereichs zugewiesen. Das heißt, falls $V(x \pm \Delta x, y \pm \Delta y)$ ein Randwert ist, gilt: $V(x \pm \Delta x, y \pm \Delta y) = -V(x, y)$ [7].

Eine weitere, jedoch unrealistische Annahme ist, dass die Wände, welche den berechenbaren Bereich umschließen, perfekte elektrische Leiter sind. Somit wird den Gitternetzknotten, welche einen Randwert darstellen, das Potential $V=0$ zugewiesen. Obwohl diese Annahme unrealistisch ist, liefert sie brauchbare Ergebnisse. Diese Methode zur Lösung des Randwertproblems heißt PEC-Boundary [8].

2.2 Finite-Integral-Methode

Die Finite-Integral-Methode ist ein numerisches Simulationsverfahren zur näherungsfreien Bestimmung der elektromagnetischen Grundgleichungen nach Maxwell⁷. Nach erfolgreicher und abgeschlossener Berechnung der Potentiale an den Gitternetzknotten, wird ein geschlossener Pfad um den elektrischen Leiter gesucht. Der geschlossene Pfad muss so dicht wie möglich um den elektrischen Leiter herum verlaufen. Hier verläuft er im kleinstmöglichen Abstand von zwei Gitterzellen um den elektrischen Leiter (Abb. 10).

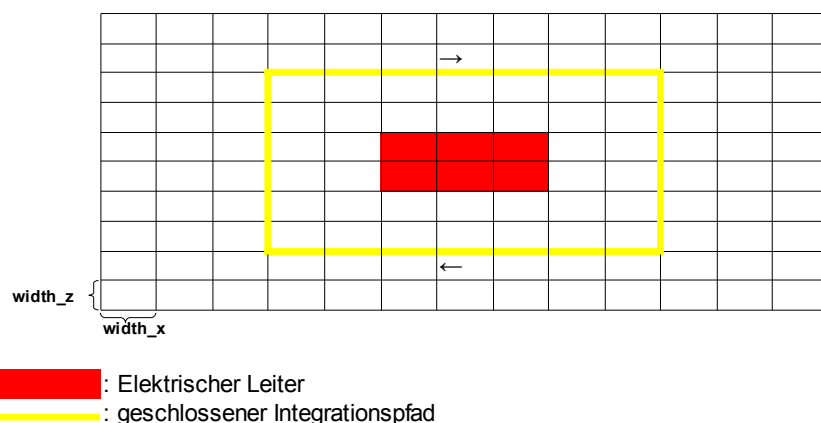


Abbildung 10: Finite-Integral-Methode: Geschlossener Integrationspfad

7 Enzyklo. Online-Enzyklopädie, Begriff: Finite-Integral-Methode

Für die folgenden Berechnungen und Herleitungen der Formeln wird auf das Paper von Green [6] verwiesen.

Für die Berechnung der Ladung um den elektrischen Leiter ist die Ableitung der Potentiale auf dem Integrationspfad in Normalenrichtung n notwendig.

$$\frac{\partial V}{\partial n} = \frac{V_B - V_A}{2 \cdot \text{width}_x}, \text{ für das linke und rechte Liniensegment}$$

$$\frac{\partial V}{\partial n} = \frac{V_D - V_C}{2 \cdot \text{width}_z}, \text{ für das obere und untere Liniensegment}$$

Daraus ergibt sich für vier Liniensegmente und r Gitternetzknoten die Ladung Q :

$$Q = \varepsilon_0 \left[2 \cdot \text{width}_x \cdot \sum_{p=1}^r \varepsilon_r \left(\frac{\partial V}{\partial n} \right)_n + 2 \cdot \text{width}_z \cdot \sum_{p=1}^r \varepsilon_r \left(\frac{\partial V}{\partial n} \right)_n \right]$$

Die modifizierte Summe $\sum_{p=1}^r$ bedeutet, dass das erste und letzte Glied der Summe

halbiert wird. Das verhindert die doppelte Addition der Eckknoten.

Aus der Ladung Q und dem Potentialunterschied V_t zwischen den Leitern berechnet sich die Kapazität C wie folgt:

$$C = \frac{Q}{V_t}$$

Die Impedanz Z_L berechnet sich schließlich aus der Lichtgeschwindigkeit ($v_0 = 2,997925 \times 10^8$) und den Kapazitäten C (mit Dielektrikum) und C_0 (ohne Dielektrikum) in einem inhomogenen Medium:

$$Z_L = \sqrt{\left(\frac{L'}{C'}\right)} = \sqrt{\left(\frac{L' C'}{C'^2}\right)} = \frac{\sqrt{(L' C')}}{C'} = \frac{1}{C' \frac{1}{\sqrt{(L' C')}}}, \text{ mit } C' = C$$

$$Z_L = \frac{1}{C \cdot v}, \text{ mit } v = \frac{1}{\sqrt{(L' C')}} = v_0 \sqrt{\left(\frac{C_0}{C}\right)}$$

$$Z_L = \frac{1}{C \cdot v_0 \sqrt{\left(\frac{C_0}{C}\right)}}$$

$$Z_L = \frac{1}{v_0 \cdot \sqrt{(C \cdot C_0)}}$$

L' : Induktivitätsbelag

C' : Kapazitätsbelag

v : Phasengeschwindigkeit

3. 3D-ACIS-Modeler

Der 3D-ACIS-Modeler ist eine kommerzielle Software der Firma Spatial, die in mehr als 350 Kundenanwendungen mit über 1,5 Millionen Firmensitzen weltweit zum Einsatz kommt⁸. Die 3D-Modeler-API ist eine objektorientierte, in C++ geschriebene Programmierschnittstelle und bietet neben den zahlreichen Funktionen zur Modellierung von Drahtgittern, Oberflächen und Festkörpern, basierend auf einer gemeinsamen und einheitlichen Datenstruktur, eine hervorragende Dokumentation. Die Hauptaufgaben des 3D-Modeler lassen sich in drei Kategorien einteilen: 3D-Modellierung, 3D-Modellverwaltung und 3D-Modellvisualisierung [21].

Als Export-, und Importschnittstelle bietet der 3D-Modeler die Dateiformate SAT und SAB an. Dateien mit der Endung .sat sind Standard ASCII Text Dateien, welche Zeilenumbrüche und andere Formatierungen enthalten, um es für den Menschen in einem gewöhnlichen Texteditor lesbar zu machen. Mit einer SAB-Datei besteht die Möglichkeit Modelle in eine binäre (Standard ASCII Binary) Datei zu exportieren. Die Schnittstelle zwischen dem 3D-CAD-Modell, welches untersucht werden soll, und ImpCalc ist eine SAT-Datei.

Die in der Software ImpCalc verwendeten Komponenten der Programmierschnittstelle werden in den nächsten Abschnitten näher erläutert.

3.1 Methode – API-Solid-Block

Mit der Funktion „api_solid_block“, die eine Funktion der Klasse „Construct Bodies“ ist, wird ein Festkörper der Form eines Quaders erstellt. Diese stellen in ImpCalc die einzelnen Meshzellen des Volumenmeshes dar. Die Übergabeparameter von „api_solid_block“ sind [22]:

- SPAposition *pt1*: Die Klasse „SPAposition“ repräsentiert einen Vektor mit x-, y- und z-Koordinate. Dieser Vektor beschreibt einen Eckpunkt des Quaders.
- SPAposition *pt2*: *pt2* beschreibt den diagonal gegenüberliegenden Eckpunkt von *pt1*.

8 Stand: April 2011

- *BODY block*: Die Klasse BODY repräsentiert in diesem Fall einen Festkörper. Somit werden die einzelnen Meshzellen in einem Objekt der Klasse BODY gespeichert.

Die Klasse BODY, welche die Topologie eines Objektes beschreibt, ist von der Klasse ENTITY abgeleitet. In ENTITY werden die gemeinsamen Daten und Funktionalitäten gespeichert. Allerdings repräsentiert eine ENTITY kein Objekt in der ACIS-Modellwelt. Eng verbunden mit der Klasse ENTITY ist die Klasse ENTITY_LIST. Diese Klasse stellt eine Liste mit variabler Länge zur Verfügung, in welche die einzelnen Entities angehängt werden. Es stehen unter anderem neben Funktionen zum Hinzufügen, Löschen und Auffinden einer Entity auch Listenverwaltungsfunktionen, wie zum Beispiel die Bestimmung der Anzahl von Entities in einer Liste, zur Verfügung.

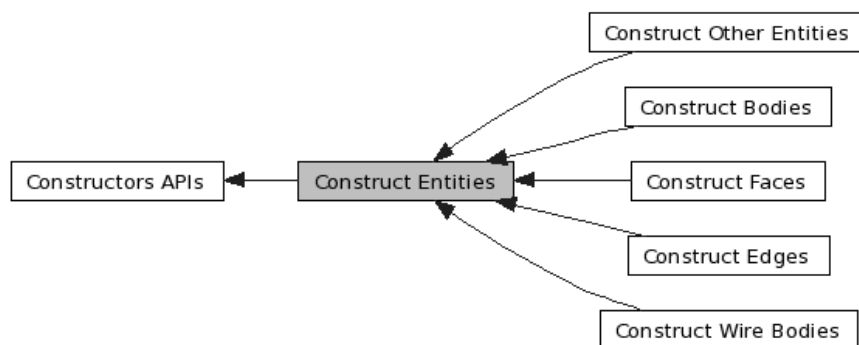


Abbildung 11: Ableitungshierarchie der Klasse „Construct Bodies“.

3.2 Methode – Api-Restore-Entity-List

Mit Hilfe dieser Methode ist es möglich aus einer externen SAT-Datei, die ein 3D-CAD-Modell beschreibt, die einzelnen Entities bzw. Objekte des Modells zu extrahieren. Diese werden in einer ENTITY_LIST gespeichert und können für weitere Berechnungen bzw. Objektmanipulationen verwendet werden. Diese Methode bietet die Möglichkeit ein externes 3D-CAD-Modell in die Software ImpCalc zu laden. Die Methode „api_restore_entity_list“ ist eine Funktion der Klasse „Save and Restore“ und besitzt folgende Übergabeparameter [22]:

- FILE *file_ptr*: File-Deskriptor, welcher die zu importierende SAT-Datei öffnet.
- BOOL *text_mode*: TRUE, falls die SAT-Datei eine Textdatei ist. FALSE, falls die

SAT-Datei im binären Format vorliegt.

- ENTITY_LIST *entities*: Angabe der Entity-List, in der die Objekte des SAT-Files gespeichert werden sollen.



Abbildung 12: Ableitungshierarchie der Klasse „Save and Restore“

3.3 Methode – Api-Clash-Bodies

Um einen Zusammenstoß von zwei Objekten vom Typ BODY zu erkennen, kann die Funktion „api_clash_bodies“ des 3D-ACIS-Modeler verwendet werden. Sie ist eine Methode der Klasse „Entity Clash Calculation“. In ImpCalc ist diese Funktion notwendig, um eine Zuordnung von den Objekten des Import-CAD-Modells zu den Gitternetzstellen des Volumenmeshes herzustellen. Die Gitternetzstellen übernehmen die Eigenschaften der jeweiligen Objekte. Die wichtigsten Übergabeparameter neben einigen Einstellungsparameter sind [22]:

- BODY body_a: Objekt a vom Typ BODY.
- BODY body_b: Objekt b vom Typ BODY.
- BODY result: Ergebnis, ob eine Berührung zwischen Objekt a und Objekt b besteht.

Bei dieser Methode gibt es verschiedene Varianten, welche die Art und Weise des Zusammenstoßes charakterisieren. In ImpCalc wird die einfachste Variante gewählt, bei welcher lediglich ausgegeben wird, ob eine Berührung von Objekt und Gitternetzstelle besteht oder nicht (siehe 4.4.1 Methode - ClashTest). Diese Variante ist zugleich die am wenigsten rechenintensivste.

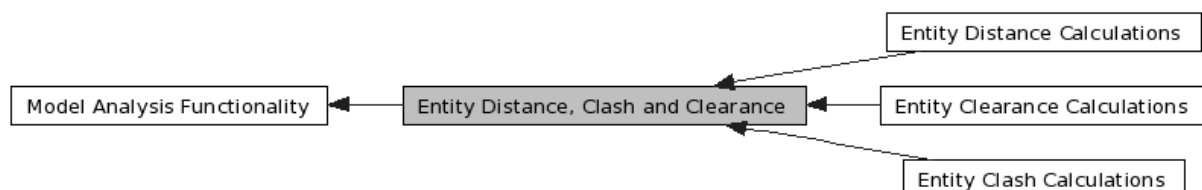


Abbildung 13: Ableitungshierarchie der Klasse „Entity Clash Calculation“

4. ImpCalc

Die Software ImpCalc wurde für die Problemstellung dieser Studienarbeit entworfen. Die Aufgabe der Software ist die Bestimmung der Impedanz einer elektrischen Leitung in möglichst kurzer Zeit und mit möglichst geringem Fehler.

Die grobe Vorgehensweise gliedert sich in fünf Schritte (Abb. 14). Im ersten Schritt wird ein Volumenmesh erzeugt, welches das 3D-CAD-Modell mit endlich vielen kleinen 3D Gitterzellen diskretisiert. Als zweites werden den einzelnen Volumenmeshzellen die jeweiligen Materialien von den Objekten des Import-CAD-Modells zugewiesen. In den letzten Schritten werden aus dem Volumenmesh mehrere 2D Meshes erzeugt. Über jedes 2D Mesh wird ein Rechennetzgitter gelegt. Anhand der Laplace-Gleichung werden mit einem iterativen Algorithmus die Potentiale an den Rechennetzpunkten, abhängig von den Materialien der Gitternetzstellen, berechnet. Mit Anwendung der Finiten-Integral-Methode lässt sich schließlich die Impedanz der elektrischen Leitung berechnen.

ImpCalc gliedert sich in 3 Klassen: ImportObject.h, Mesh.h sowie einer Startklasse MyAcis.h mit einer main-Routine.

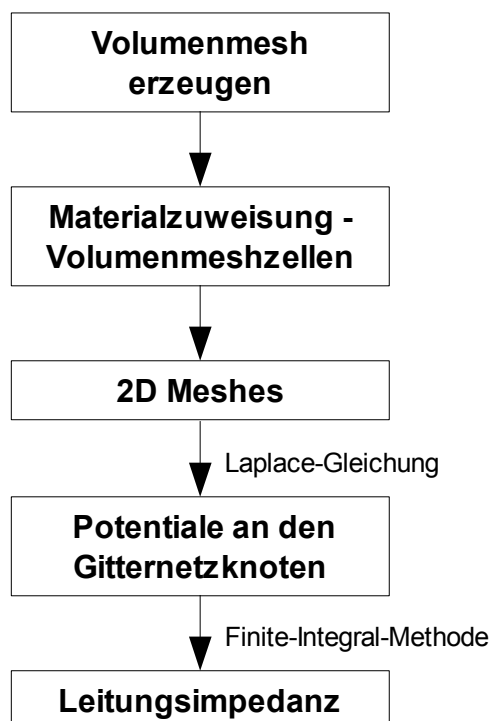


Abbildung 14: Vorgehensweise in ImpCalc zur Berechnung der Leitungsimpedanz.

4.1 Klasse – ImportObject

In dieser Klasse werden alle Informationen bezüglich des zu untersuchenden 3D-CAD-Modells gespeichert. In erster Linie geht es um die Materialzuweisung per CSV-Datei und um die Vorbereitung zur Volumenmeshgenerierung. Im Konstruktor der Klasse „ImportObject“ werden die Objekte des 3D-CAD-Modells aus einer SAT-Datei importiert, mit einer ID versehen und in einem Array gespeichert. Für die Materialzuweisung sind die folgenden zwei Methoden der Klasse zuständig:

- **void CreateMaterialFile():** Es wird die Datei AttrFile.csv erstellt. In dieser Datei sind die einzelnen Objekte mit ihrer ID aufgelistet. Diese Datei wird vom Anwender mit den Schlüsselwörtern „**MATERIAL=**“ und „**V=**“ erweitert. Unter V ist das „Start-Potential“ der Materialien zu verstehen.
Bei „Material=COPPER“ ist die Angabe des Attributs V zwingend. Als Werte werden V=1 und V=0 akzeptiert. Bei V=0 handelt es sich um das Bezugspotential.
Bei anderen Materialien (ungleich COPPER) ist die Angabe des Attributs V optional; wird nichts angegeben wird als Defaultwert V=0 gesetzt.
- **void SetMaterial():** Das Material sowie das „Start-Potential“ aus AttrFile.csv wird in einer Materialdatenbank zur weiteren Verarbeitung gespeichert.

Als Vorbereitung für die Klasse „Mesh“ wird die Position und Dimension des Volumenmeshes festgelegt. Durch die Methode **void BoundingBox()** werden die Start-, und Endpunkte des Volumenmeshes bestimmt.

4.2 Diskretisierungskorrektur

Um einen Diskretisierungsfehler zu vermeiden wird eine Korrektur durchgeführt. Diese ist jedoch nur bei einem Schnitt in der x-z-Ebene möglich. Das bedeutet, die zu untersuchende Leitung muss in Richtung der y-Achse verlaufen.

Bei dieser Funktion wird zunächst der Leiter (Potential = 1.0) gesucht und von einer BoundingBox umhüllt, um die Position des Leiters zu bestimmen.

Anschließend werden die Dimensionen der Gitterzellen in soweit angepasst, dass die Gitterzellen bei der Diskretisierung in x-, und z-Richtung mit der Form des Leiters exakt zusammenfallen.

Da sich der Diskretisierungsfehler ziemlich stark auf die Berechnung der Impedanz

auswirkt, ist solch eine Korrektur notwendig.

Beispiel für die Auswirkung des Diskretisierungsfehlers:

Leiter mit: Weite = 317,5 μm , Dicke = 35,56 μm , Substrathöhe = 645,16 μm , $\epsilon_r = 3,25$, Impedanz $Z_L = 50 \text{ Ohm}$ (berechnet mit LineCalc).

Diskretisierungsfehler in der Dicke (μm)	Diskretisierungsfehler in der Breite (μm)	Erwartetes Ergebnis (Ω)	Berechnetes Ergebnis (Ω)	Abweichung
$\sim 9,3$	$\sim 82,2$	50	42,1344	7,87 (15,7%)
~ 0	~ 0	50	48,3475	1,65 (3,3%)

Tabelle 1: Auswirkung des Diskretisierungsfehlers

Die Potentialberechnung wurde bei beiden Impedanzberechnungen (Tabelle 1) solange iteriert, bis die Differenz der aufsummierten Potentiale auf dem Integrationspfad (Ladung) von zwei aufeinanderfolgenden Iterationsschritten kleiner 10^{-18} war.

4.4 Klasse – Mesh

Diese Klasse übernimmt die Hauptaufgabe der Software. Zum einen wird das Volumenmesh erzeugt und zum anderen laufen die Algorithmen zur Berechnung der Leitungsimpedanz ab. Die Erzeugung des Volumenmeshes ist im Konstruktor **Mesh()** verankert.

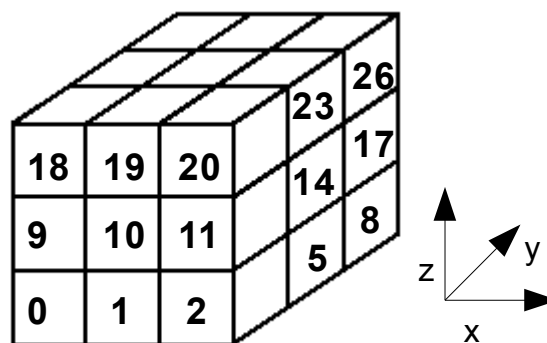


Abbildung 15: Nummerierungsschema der einzelnen 3D Meshzellen im Volumenmesh

Hier wird anhand der Wertvorgabe des Benutzers, für die Meshzellenweite (x_width , y_width , z_width) und der zuvor berechneten Start-, und Endpunkte, ein Volumenmesh erzeugt (Abb.16). Die einzelnen 3D Meshzellen im Volumenmesh werden beginnend mit 0 fortlaufend, nach dem Schema in Abbildung 15, durchnummeriert.

Die Klasse besitzt noch zwei weitere Methoden. Zum einen die Funktion **ClashTest()** und zum anderen die mächtigste Funktion der Software **Impedance()**. Die Funktionsweisen dieser Methoden werden in den nächsten beiden Abschnitten erläutert.

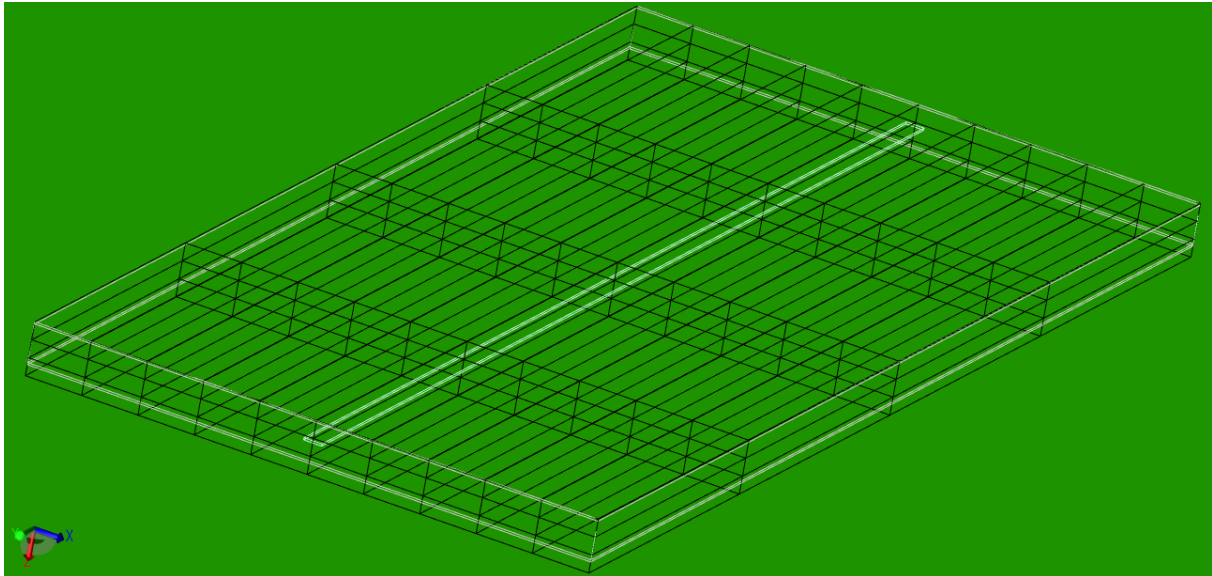


Abbildung 16: Volumenmesh.

Schwarzer Umriss: Einzelne Meshzellen mit der Meshzellenweite (x_width , y_width , z_width).

Weißer Umriss: oben: Kupferplatte mit Potential = 0, unten: Kupferplatte mit Potential = 0, Mitte: elektrische Kupferleitung mit Potential = 1.

4.4.1 Methode – ClashTest()

Die Aufgabe von ClashTest() ist es herauszufinden welche Meshzellen des Volumenmeshes sich mit welchen Objekten des 3D-CAD-Modells (Importobjekt) berühren. Dies ist durch zwei verschachtelte Schleifen realisiert. Während die äußere Schleife jede Meshzelle des Volumenmeshes durchläuft, geht die innere Schleife über jede Entity des Importobjekts. Somit wird jede Möglichkeit einer Berührung von Meshzelle und Importobjekt überprüft. Liegt eine Berührung vor, so wird der entsprechenden Meshzelle dasselbe Material zugewiesen, welches auch das Objekt des 3D-CAD-Modells besitzt. Meshzellen, die sich mit keinem Objekt des CAD-Modells berühren, besitzen als Defaultwert das Material VACUUM. Neben dem Material wird jeder Meshzelle auch ein „Start-Potential“ zugewiesen. Im Falle von Kupfer werden die „Start-Potentiale“ bereits im AttrFile.csv vom Benutzer festgelegt. Das ist wichtig, da sich bei der späteren, iterativen Potentialberechnung die Potentialwerte an jenen Gitternetzknotten nicht mehr verändern

dürfen.

4.4.2 Methode – Impedance()

Mit dem resultierenden Volumenmesh aus der Methode ClashTest() wird nun die Impedanz der elektrischen Leitung berechnet. Als erstes wird das Volumenmesh in „Scheiben geschnitten“. Auf Basis der daraus entstehenden 2D Meshes (Abb. 17) findet die Impedanzberechnung statt. Über jede einzelne Scheibe wird eine Matrix (Rechennetzgitter) gelegt, wobei jeder Eintrag der Matrix einem Eckpunkt der Meshzelle entspricht. Diese Matrix heißt in ImpCalc **Slice-Matrix**. Zu jeder Meshzelle gehören vier Rechennetzpunkte.

Die Rechennetzpunkte besitzen eine Variable für das Potential und eine als Lock-Flag, welches auf TRUE gesetzt wird, sofern es sich um eine Meshzelle mit Material COPPER handelt. Wie bereits schon erwähnt, ist das Lock-Flag wichtig, da bei der Berechnung jene Rechennetzpunkte ihr Potential nicht verändern dürfen.

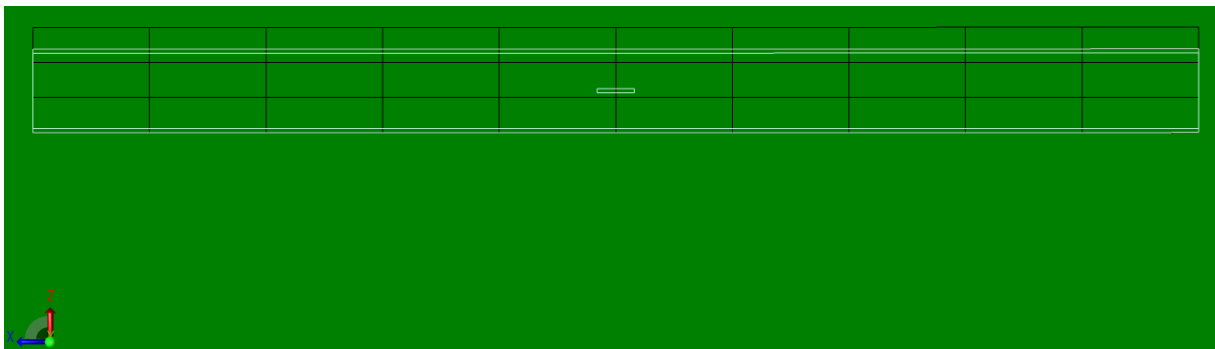


Abbildung 17: 2D Mesh: Eine Scheibe des Volumenmeshes aus Abb. 16

Mit der Variablen **slice_type** besteht die Möglichkeit die Schnittebene auszuwählen. Je nachdem wird mit den Werten 0, 1 oder 2 das Volumenmesh entlang der x-z-, x-y- oder y-z-Ebene in Scheiben zerlegt. Anschließend werden an den Gitternetzknotten jeder Scheibe die Potentiale berechnet (siehe 2.1 – Laplace-Gleichung). Um nach erfolgreicher Berechnung der Potentiale einen Wert für die Impedanz zu bekommen, ist das Auffinden des Leiters (Kupfer mit Potential $V = 1.0$) notwendig. In ImpCalc ist das Auffinden durch ein vierseitiges Eingrenzen von oben, unten, links und rechts realisiert. Somit werden die vier Eckpunkte des Leiters berechnet. Anhand dieser Punkte ist die Lage des Leiters eindeutig und es kann das nächstliegende Umlaufintegral mit Hilfe der Finiten-Integral-Methode berechnet werden (siehe 2.2 – Finite Integral-Methode).

5. Ergebnisse

Um die Software zu testen wurde die Impedanz einer Streifenleitung numerisch berechnet. Die Geometrie dieser Leitung ist in Abbildung 18 zusehen. Die Breite der kompletten Geometrie beträgt 10,0mm und die Länge 20,0mm.

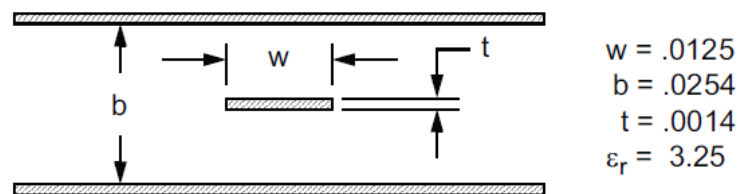


Abbildung 18: Streifenleitung: Die Maßangaben sind in Zoll zu verstehen [13].

Die Leitungsimpedanz dieser Geometrie wurde mit LineCalc berechnet und beträgt 50Ω [13]. Im folgenden werden die Ergebnisse von ImpCalc mit den Ergebnissen von LineCalc verglichen.

Die Leitungsimpedanz wurde mit ImpCalc und verschiedenen Gitterzellenweiten numerisch berechnet. Um eine Verfälschung des Ergebnisses durch einen Diskretisierungsfehler zu vermeiden, wurde die Diskretisierungskorrektur in ImpCalc aktiviert. Wie in Tabelle 2 zusehen ist, erreicht das numerische Ergebnis von ImpCalc, bereits mit einer Granularität von ca. 80.000 Meshzellen, sehr nah den berechneten Wert von LineCalc.

Diese Berechnungen wurden mit einem Intel Core i5 processor 520M (2,4 Ghz, 3MB L3 Cache) und 4GB Memory durchgeführt.

Gitterzellenweite			Anzahl Gitterzellen	Berechnungsdauer (s)	Impedanz (Ω)	Abweichung zu LineCalc
width_x (mm)	width_x (mm)	width_x (mm)				
0,05290	20,00000	0,00509	26790	238	48,3575	3,29%
0,03970	20,00000	0,00396	45612	448	48,7974	2,41%
0,02887	20,00000	0,00297	83974	1095	49,2915	1,42%
0,02880	20,00000	0,00199	125614	3045	49,3306	1,34%

Tabelle 2: Ergebnisse

Den immensen Anstieg an Rechenaufwand mit steigender Anzahl von Gitterzellen ist ebenfalls der Tabelle 2 zu entnehmen. Da jede Gitterzelle einen Speicherplatz im Hauptspeicher belegt, ist bei einer Anzahl von ca. 125000 Gitterzellen bereits mit einem Speicherbedarf von ca. 2GB zu rechnen.

6. Zusammenfassung

Nach einer Einführung in das Meshing und der Vorstellung von hexahedralisierten, tetrahedralisierten und hybriden Meshes wurden verschiedene numerische Methoden zur elektromagnetischen Feldberechnung, basierend auf diesen Mesh-Typen, beschrieben. Ein besonderes Augenmerk wurde auf das hexahedralisierte Mesh zusammen mit der Finiten-Integral-Methode geworfen, da diese in dem hier vorgestellten Leitungsimpedanzberechnungsverfahren zum Einsatz kommen. Die Meshgenerierung wurde in ImpCalc durch die Applikationsschnittstelle 3D-ACIS-Modeler der Firma Spatial realisiert. Die Ergebnisse einer Testberechnung anhand einer Referenzgeometrie spiegeln die Performanz von ImpCalc wieder. Es ist zu erkennen, dass der berechnete Impedanzwert von ImpCalc, bereits bei einem eher groben Meshing, ziemlich nahe an dem von LineCalc berechneten Sollwert liegt.

7. Ausblick

Um die Performanz der Software weiter zu steigern, könnte man die Berechnungen parallelisieren. Somit könnte jeder vorhandene Kern ausgelastet werden. Dies würde vor allem bei den zehntausenden Schleifendurchläufen (Potentialberechnung) eine große Zeitersparnis ergeben. Ein Ansatz für die Parallelisierung wäre die Programmierschnittstelle Open mulit-processing (OpenMP). Mit dieser, auch für Visual Studio C++ erhältlichen Schnittstelle, ist eine Schleifenparallelisierung einfach zu realisieren.

Eine andere Möglichkeit wäre ein irreguläres⁹ Gitternetz, das heißt die Gitterzellen werden an den „interessanten“ Stellen feiner und an den „uninteressanten“ Stellen gröber. Interessante Stellen wären in der hier vorgestellten Testgeometrie zum einen die Kupferleitung in der Mitte und zum anderen die Kupferplatten oben und unten. Das Dielektrikum könnte demzufolge gröber gemesht werden. Für ein irreguläres Gitternetz müssten jedoch die Berechnungen für das Potential und das Umlaufintegral angepasst werden.

Eine Erweiterung für die Software ImpCalc wäre die Möglichkeit zur Berechnung der differentiellen Impedanz (Abb. 18). Hierbei wird die Impedanz zwischen zwei Leitungen berechnet. Bei einem parallelen Verlauf der Streifenleitungen kann das Signal invers in die Leitungen eingespeist werden. Dieser Modus wird Odd-Mode (Gegentakt) genannt. Im Gegensatz dazu gibt es den Even-Mode, bei welchem das Signal im Gleichtakt in die Leitungen eingespeist wird. Die Auswahl des jeweiligen Modus hängt vom Schaltungsdesign ab [10] [11].

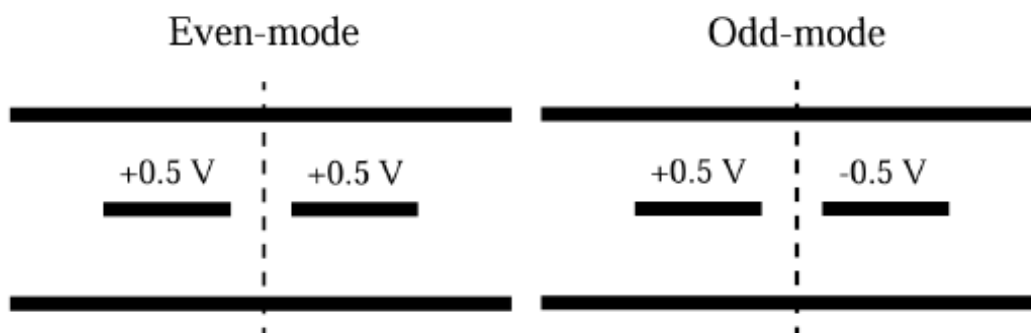


Abbildung 19: Differentielle Impedanz [9]

⁹ hier: individuelle Dimension jeder einzelnen Meshzelle innerhalb eines Gitters.

8. Installation und Inbetriebnahme der Software

Um eine eigene Anwendung zu entwickeln, welche Applikationen von Spatial benutzen, ist ein Lizenzschlüssel notwendig. Diesen Schlüssel kann man direkt bei Spatial¹⁰ erwerben. Die folgende Anleitung zeigt die Vorgehensweise von der Installation bis zur Inbetriebnahme und Benutzung der Software ImpCalc.

1. Folgenden Link öffnen und alle Schritte unter dem Abschnitt „Building as a Standalone Application“ durchführen.
http://doc.spatial.com/index.php/Building_Your_Application
2. Laden des Visual-Studio-Projekts ImpCalc.
3. Den erworbenen LicenseKey (cpp-File) in den Projektordner ImpCalc kopieren und in das Visual-Studio-Projekt einfügen.
4. License-Key registrieren:
 1. spa_lic_app.exe ausführen (liegt im Installationsverzeichnis unter /bin/.../)
 2. Reiter Application anklicken
 3. Daten von LicenseKey eintragen (Abb. 20).

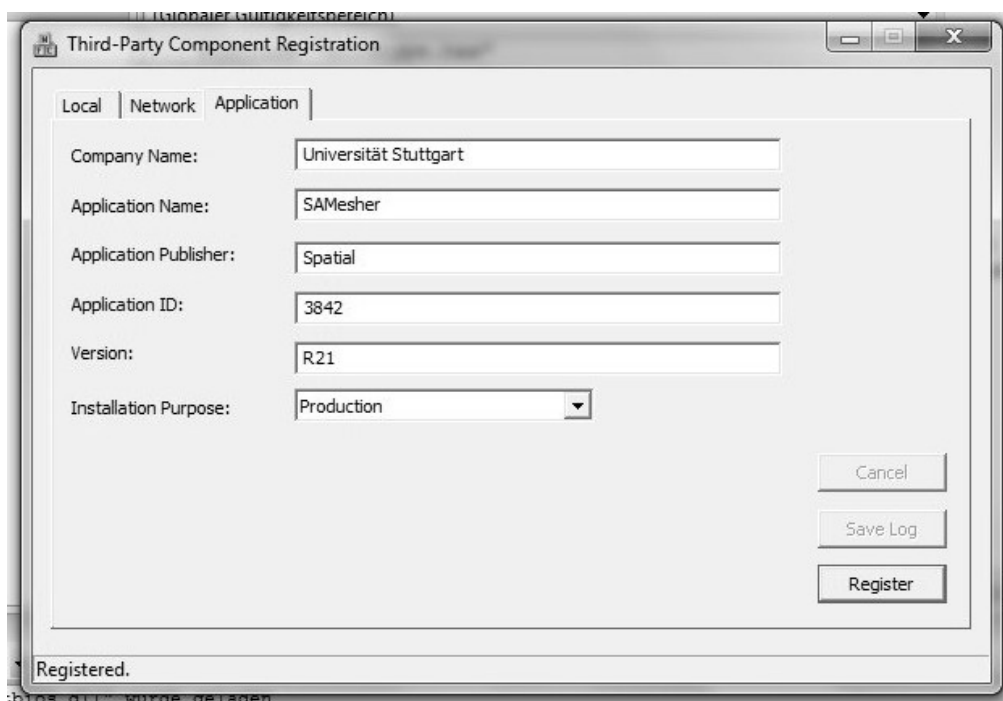


Abbildung 20: Registrierung des License-Key

¹⁰ <http://www.spatial.com>

4. Anpassen der Datei MyAcis.cpp im Projekt ImpCalc (Abb. 21)

1. Hinzufügen von fileinfo.hxx
2. Unlock-Funktion abhängig vom erworbenen License-Key definieren.
3. Unlock-Funktion aufrufen.

```
#include "fileinfo.hxx"
```

1.

```
void process(outcome result);  
void process_file(FILE* fp);  
ENTITY_LIST retrieve_sat_file (char *fileName);  
void create_sat_file (ENTITY *ent, char *fileName);  
void create_sat_file (ENTITY_LIST elist, char *fileName);  
void Initialize_ACIS ();  
void Terminate_ACIS ();  
void Check_Entity (ENTITY* input);  
void unlock_spatial_products_3842 ();
```

2.

```
void main()  
{  
    api_start_modeller(0);  
    Initialize_ACIS ();
```

```
    //add unlock license key here
```

```
    unlock_spatial_products_3842 ();
```

3.

Abbildung 21: Anpassung von MyAcis.cpp

5. Die Bedienung der Software ist als Kommentar im Quellcode beschrieben.

9. Quellenverzeichnis

- [1] V. Shankar, W. F. Hall, and A. H. Mohammadian, "A Time-Domain Differential Solver for Electromagnetic Scattering Problems", Proc. of IEEE, Vol. 77, no. 5, pp. 709-721, 1989.
- [2] J. H. Ferziger and M. Peric. *Computational Methods for Fluid Dynamics, Second Edition*. Springer, Berlin-Heidelberg, 1999.
- [3] H. K. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method. Second edition*, Pearson Education Limited, Harlow, 2007.
- [4] Dipl.-Ing. Ngoc-Hoa Huynh. Dissertation: Verbesserung der Effizienz der FDTD-Methode für die Analyse von koplanaren MMICs. Elektrotechnik der Technischen Universität Berlin, 2000.
- [5] Wassim Abu Abed. Masterarbeit: Finite Volumen Methode zur Lösung Advektions-Diffusionsgleichungen. Institut für Bauinformatik – Leibniz Universität Hannover, 2007.
- [6] Green, H. E.. The Numerical Solution of Some Important Transmission-Line Problems. IEEE Transactions on Microwave Theory and Techniques, vol. 13, issue 5, pp. 676-692, 1965.
- [7] Mütter, Herbert. Vorlesung Physik 2: Vorlesungsskript Kapitel 1 – Elektrostatik, Seite 39ff, Sommersemester 2003, Institut für Theoretische Physik, Eberhard-Karls-Universität Tübingen. <http://solid13.tphys.physik.uni-tuebingen.de/muether/physik2/skript/01-Elektrostatik.pdf>
- [8] Kraus, Christof. Diplomarbeit: Finite Element-Time Domain (FETD) Simulation. Paul Scherrer Institut (PSI) & Institut für Computational Science, ETHZ – Eidgenössische Technische Hochschule Zürich, 2006.
- [9] Swanson, Jr., D. G., "What's My Impedance?" IEEE Microwave Magazine, Vol. 2, No. 4, 2001, pp. 72–82.
- [10] Prof.Dr.Rainer Thüringer. Verbindungstechnik, Kapitel 2. AVT: HighTech-/HighSpeed-Baugruppen, FB Elektro- und Informationstechnik der FH Gießen-Friedberg, 2008. http://www.et.fh-jena.de/scholz/F%C3%BCr-Mechatroniker/Skripte-G--ET-Mechatroniker/FH--Gie%C3%9Fen-Friedberg/Verbindungstechnik/avt_highspeed_bg_kap2.pdf
- [11] Polar Instruments Ltd. Applikationsabschrift 157, Even-Mode Impedanz — eine Einführung, 2002. http://www.polarinstruments.com/de/support/cits/AP157_D.html

-
- [12] Elektronik Kompendium. 2011. <http://www.elektronik-kompendium.de/sites/grd/0301036.htm>
- [13] Daniel G., Jr. Swanson, Wolfgang J. R. Hofer. Buch: Microwave Circuit Modeling Using Electromagnetic Field Simulation. Verlag: Artech House. Auflage: illustrated edition, Juni 2003
- [14] Prof. Dr. W. Stanek. MAXWELL THEORY HOME, 2010.
<http://www.wolfram-stanek.de/maxwell/index.htm>
- [15] Prof. Dr.-Ing. Habil. Jürgen Dankert. Numerische Methoden – Kapitel 4. HAE Hamburg, 2007. <http://www.rzbt.haw-hamburg.de/dankert/numet.html>
- [16] Dr.-Ing. Günter Müller, Geschäftsführer CADFEM GmbH. Finite-Elemente-Methode, 2011. <http://www.cadfem.de/unternehmen/portrait/finite-element-methode.html>
- [17] Blockkursunterlagen. Finite Differenzen Methode. Technische Universität Berlin, 2003.
<http://www.math.tu-berlin.de/ppm/skripte/fdm2.0-2auf1.pdf>
- [18] Dipl.-Ing. Sebastian Kytzia. Dissertation: Analyse, Optimierung und Entwicklung von Mikrowellen-angeregten Plasmaquellen mittels numerischer Simulation. Kapitel 2.3.2 Numerische FI-Methode. Fachbereich Elektrotechnik, Informationstechnik und Medientechnik. Bergische Universität Wuppertal, 2009.
<http://elpub.bib.uni-wuppertal.de/servlets/DerivateServlet/Derivate-1096/de0902.pdf>
- [19] Dipl.-Phys. Thomas Lau. Dissertation: Numerische Methoden zur Simulation teilchengenerierter elektromagnetischer Felder in der Beschleunigerphysik. Elektrotechnik und Informationstechnik. Technische Universität Darmstadt, 2006.
<http://tuprints.ulb.tu-darmstadt.de/772/1/ThomasLau-Thesis.pdf>
- [20] Matthias Bollhöfer. Kompaktkurs: Geometrische Mehrgitterverfahren und Algebraische Mehrgitterverfahren. Technische Universität Berlin, 2001.
<http://www.math.tu-berlin.de/numerik/mt/Sources/Bollhoefer/lectures/hamburg2001/kompaktkurs.pdf.gz>
- [21] Spatial Corp. 2011. <http://www.spatial.com/products/3d-acis-modeling>
- [22] Spatial Docs – External Product Documentation. Begriffe: Api_Solid_Block, Api_Restore_Entity_List, Api_Clash_Bodies, 2011. http://doc.spatial.com/index.php/Main_Page
- [23] Tilmann Wittig. Buch: Zur Reduzierung der Modellordnung in elektromagnetischen Feldsimulationen. Verlag: Cuvillier, E; Auflage: 1., Aufl. (August 2004).

Erklärung

Hiermit versichere ich, diese Arbeit selbstständig verfasst und nur die angegebenen Quellen benutzt zu haben.

Unterschrift:

Stuttgart, 04.05.2011