Wilhelm-Ostwald-Schule, Gymnasium der Stadt Leipzig

Dokumentation zur Besonderen Lernleistung

im Fachbereich: Informatik

Thema:	Entwicklung einer Software zur Auswertung und Visualisierung von Versuchen zur Materialerwärmung durch elektromagnetische Wellen
vorgelegt von:	Valentin Roland
Schuljahr:	2014/2015
Externe(r) Betreuer:	Prof. Dr.-Ing. Lutz Nietner Hochschule für Technik, Wirtschaft und Kultur (HTWK) Leipzig Fakultät Bauwesen, LG Bausanierung / Baustoffe
Interner Betreuer:	Herr Knospe

Leipzig, 19.12.14

Abstract

The aim of this study was to develop software which assists users with the analysis and evaluation of experiments related to material heating. To achieve that, it allows to correct errors and merge all measured temperature data into one data format. This data can then be used to calculate a three-dimensional temperature distribution throughout a given three-dimensional model. Using the finite element method, the software determines the object's energy content. Furthermore, the temperature distribution can be visualized in two or three dimensions. Finally, the data can be exported in several ways for further use. For better usability there is a manual and source code documentation, as the programmes are open source. At the Helmholtz-Zentrum für Umweltforschung - UFZ in Leipzig, Germany, the software was successfully used in the evaluation of an experimental series investigating dielectric heating.

Kurzfassung

Ziel der BeLL an der HTWK Leipzig war die Entwicklung von Software, die den Nutzer bei der Auswertung von Versuchen zur Materialerwärmung und der Aufbereitung sowie der Interpretation der Ergebnisse unterstützt. Bei derartigen Experimenten wird ein Material mit verschiedenen Methoden, in der aktuellen Anwendung speziell durch elektromagnetische Wellen, erwärmt. Dabei wird die Temperatur an mehreren Punkten innerhalb des erwärmten Objekts über einen bestimmten Zeitraum unter Verwendung verschiedenartiger Messgeräte gemessen. Aufgabe der Software ist es nun, die Daten der Messgeräte von Messfehlern zu bereinigen und in eine einheitliche Datenstruktur zusammenzuführen. Anschließend erfolgt die Berechnung der Temperaturverteilung über ein dreidimensionales Modell des erwärmten Gegenstandes. Diese kann bezüglich des Energiegehalts mithilfe der Finite-Elemente-Methode ausgewertet und zwei- sowie dreidimensional visualisiert werden. Die Ergebnisse können durch mehrere Exportmöglichkeiten für die weitere Verwendung zur Verfügung gestellt werden. Um die Nutzbarkeit der Software für jeden Anwender zu gewährleisten, sind dabei Benutzerfreundlichkeit und Dokumentation wichtig, weshalb auch ein Handbuch zur Software erstellt wurde. Für eine spätere Weiterentwicklung der entstandenen Open-Source-Software steht eine Dokumentation des Quelltextes zur Verfügung. Die Software wurde bereits im Rahmen einer Versuchsserie eingesetzt, bei der am Helmholtz-Zentrum für Umweltforschung - UFZ im Rahmen eines Forschungsprojektes verschiedene Heizverfahren eingesetzt wurden, um das Anwendungspotenzial der dielektrischen Erwärmung mittels Radiowellen im Vergleich zur Mikrowellen- und Infraroterwärmung zu ermitteln.

Inhaltsverzeichnis

1 Einleitung

Bei umfangreicheren wissenschaftlichen Versuchen werden oft große Mengen an Messdaten ermittelt, die nicht mehr ohne Zuhilfenahme von Computern zu verarbeiten sind. Die dabei verwendete Software muss jedoch auf eine bestimmte Art von Experimenten spezialisiert sein, um so effektiv wie möglich eingesetzt werden zu können. Solche Spezialsoftware kann die Aufbereitung der Messwerte von den Rohdaten der Messgeräte bis zur Analyse und Visualisierung übernehmen. Auch die Bereinigung und Vereinigung von Daten aus verschiedenen Quellen kann in den Verarbeitungsprozess eingegliedert werden. Insgesamt ist mit den Einsatz spezialisierter Software eine wesentlich schnellere und effektivere Auswertung eines Versuchstyps möglich.
Eine solche Software soll auch bei der Auswertung von Versuchen zur Erwärmung von Materialien mithilfe elektromagnetischer Wellen am Helmholtz-Zentrum für Umweltforschung - UFZ in Leipzig sowie an der HTWK Leipzig zur Anwendung kommen. Die dort entwickelte Technologie ermöglicht die Erwärmung von Materialien mittels Radiowellen. Dies ist vergleichbar mit der Funktionsweise einer Mikrowelle, jedoch werden kleinere Frequenzen (MHz- statt GHz-Bereich) verwendet. Dadurch dringt die Strahlung tiefer in das Material ein und heizt dieses gleichmäßiger auf. Ein weiterer Vorteil ist die Flexibilität der Technik, da das Material nicht durch Einbringen von Heizelementen beschädigt wird. So können auch viele Baustoffe schnell aus dem Inneren aufgeheizt werden. Dadurch ergeben sich mögliche Anwendungen bei der Trocknung von Gemäuern. Aber auch chemikalenfreier Holzschutz durch das Abtöten von Schädlingen oder das Austreiben von Schadstoffen aus Materialen ist möglich. (Hoyer u. a., 2014) (Roland u. a., 2011)
Dabei ist es in der Regel wichtig, dass sich das Objekt kontrolliert und gleichmäßig aufheizt.
Um dies zu überprüfen, soll die Software eine dreidimensionale Temperaturverteilung im untersuchten Objekt ermitteln können. Aus dieser Temperaturverteilung soll der Gehalt an Wärmeenergie bestimmt werden. Damit die Ergebnisse besser vom Nutzer erfasst und interpretiert werden können, ist eine visuelle Aufbereitung dieser von Vorteil. Dies soll in Form einer dreidimensionalen Darstellung oder einer Schnittebenenauswertung geschehen. Analyseergebnisse und zur Visualisierung generierte Grafiken müssen aus der Anwendung exportierbar sein, um eine Eingliederung in bestehende Abläufe zu ermöglichen.
Zur effektiven Vereinfachung des Auswertungsprozesses muss die Software jedoch nicht nur die nötigen Funktionen bieten, sondern auch schnell erlernbar und einfach bedienbar sein. Dies

umfasst nicht nur eine entsprechend gestaltete Programmoberfläche bzw. Benutzerschnittstelle, sondern auch die Erstellung eines Handbuches und einer Dokumentation des Quelltextes zur Weiterentwicklung des Programms.

2 Vorüberlegungen

2.1 Grundlagen der dielektrischen Erwärmung

Bei der dielektrischen Erwärmung werden nicht leitende Stoffe durch elektromagnetische Wellen erwärmt. Dies geschieht dadurch, dass die die geladenen Teilchen in den Molekülen bzw. Strukturen dem wechselnden äußeren elektrischen Feld folgen. So kommt es zur schnellen Neuausrichtung der Teilchen und zu Wechselwirkungen zwischen diesen Teilchen mit ihrer mikroskopischen Umgebung im Material. Dadurch erhöht sich die innere Energie des Materials und somit dessen Temperatur. (Metaxas u. Meredith, 1983).

2.2 Versuchsaufbau

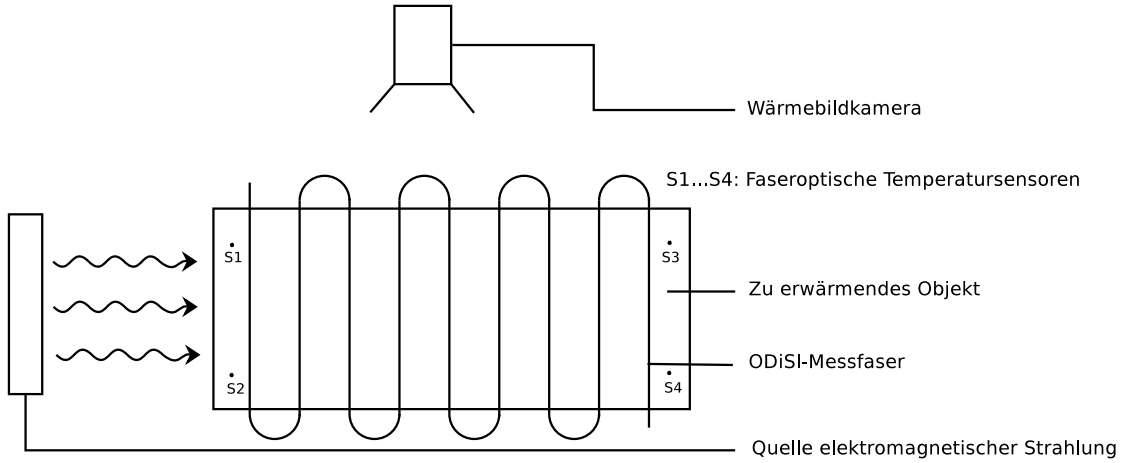
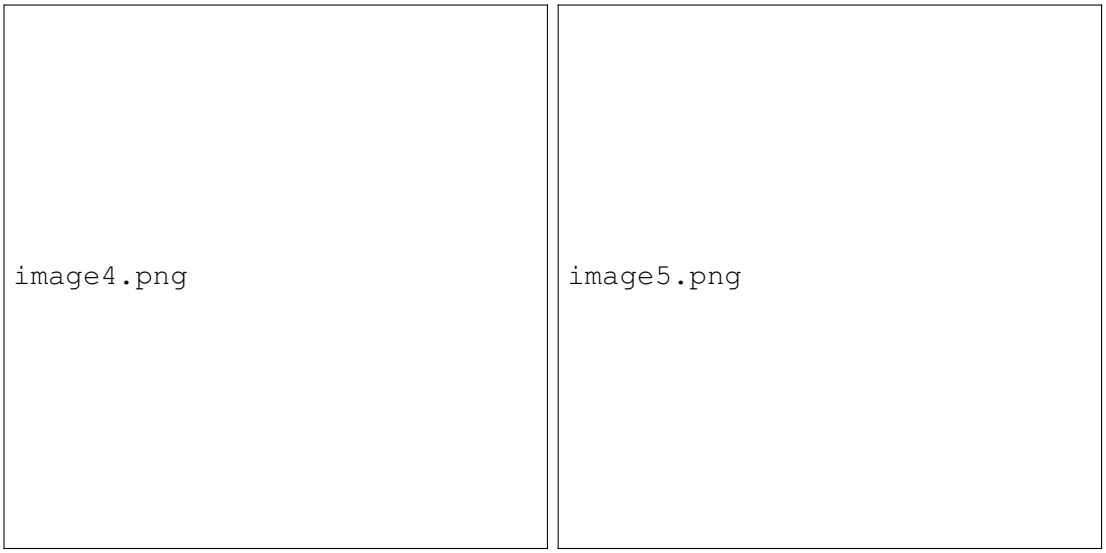


Abbildung 1: Aufbau des Standardversuchs

Beim Standardversuch zur dielektrischen Erwärmung wird ein Versuchsobjekt hochfrequenter elektromagnetischer Strahlung (Mikrowellen oder Radiowellen) ausgesetzt (s. Abb. 1). Die Temperaturveränderungen werden dabei mit verschiedenen Sensoren, hier einer Wärmebildkamera (s. 2.3.1), faseroptischen Sensoren (s. 2.3.2) und Messfasern des ODiSI-Messsystems von Luna Inc. (s. 2.3.3) erfasst. Die Lage der Sensoren ist in der Abbildung verdeutlicht. Während die



(a) Verlauf der ODiSI-Faser (b) Position der faseroptischen Sensoren

Abbildung 2: Beispiele für die Positionierung der Sensoren (UFZ, 2013)

Punktsensoren im Volumen in Bohrlöchern verteilt sind, verläuft die Messfaser in Schleifen durch das Objekt (s. Abb. 2). Von außen wird mit der Wärmebildkamera gefilmt, so dass später aus den Aufnahmen Temperaturwerte für einige Punkte an der Oberfläche ermittelt werden können.

2.3 Verwendete Temperaturmessverfahren

2.3.1 Wärmebildkamera

Mithilfe einer Wärmebildkamera kann die Oberflächentemperatur eines Festkörpers bestimmt werden. Das Messverfahren beruht darauf, dass jeder Körper mit einer Temperatur oberhalb von 0 K ein charakteristisches Spektrum abstrahlt, aus dem die Oberflächentemperatur bestimmt werden kann. Dieses Messverfahren wird beispielsweise auch eingesetzt, um die Wärmeabstrahlung von Gebäuden zu messen und thermische Schwachstellen zu ermitteln. Bei den durchgeführten Versuchen erlaubt die Messung mittels Wärmebildkamera die Messung der Temperaturverteilung auf der Oberfläche des Versuchsobjekts, sofern diese sichtbar ist.

2.3.2 Faseroptische Sensoren

Die Temperaturmessung unter dem Einfluss von elektromagnetischen Feldern erfordert eine spezielle Messtechnik, da metallhaltige Messinstrumente wie Widerstandsthermometer oder Ther-

moelemente gestört werden oder durch sie die Feldverteilung und damit das Versuchsergebnis beeinflusst werden. Deshalb werden optische Sensoren eingesetzt. Bei den verwendeten faseroptischen Punktsensoren befindet sich am Ende einer Lichtleitfaser ein Halbleiterkristall, dessen optische Eigenschaften mit einem Spektrometer erfasst werden. Der energetische Abstand zwischen Valenz- und Leitungsband dieses Halbleiters (Bandlücke) ändert sich mit der Temperatur. Um diese zu bestimmen, muss die Größe der Bandlücke gemessen werden: Dazu wird Licht verschieden hoher Energie auf den Halbleiter gestrahlt. Bei hoher Lichtenergie können Elektronen aus dem Valenzband in das Leitungsband wechseln und das Licht wird absorbiert. Ist die Energie des Lichts zu gering, geschieht dies nicht mehr. Aus der Charakteristik der Lichtabsorption in Abhängigkeit von der Wellenlänge lässt sich die die Größe der Bandlücke und daraus die Temperatur ermitteln.

2.3.3 ODiSI-Messsystem

Das ODiSI (Optical Distributed Sensor Interrogator)-Messystem basiert auf der Messung der Lichtstreuung von monochromatischem Licht entlang einer Lichtleitfaser. Die verwendete Faser besitzt spezifische ortsabhängige Materialunregelmäßigkeiten, wodurch eine ortsabhängige Rückstreuung des Lichts hervorgerufen wird. Somit kann für verschiedene Wellenlängen ein spezifisches Rückstreuprofil der Faser ermittelt werden. Über die verschiedenen Wellenlängen kann das zurückgestreute Licht einer Position entlang der Faser durch eine mathematische Prozedur zugeordnet werden (Fourier-Transformation). Ändert sich die Temperatur an einer Stelle, dehnt sich die Faser aus und ändert ihr Streumuster an dieser Position. Kennt man das Ausdehnungsverhalten der Faser, lässt sich aus dem veränderten Rückstreuprofil die Temperatur für eine große Anzahl an Messstellen entlang der Faser ermitteln.

3 Methoden

3.1 Finite-Elemente-Methode

Die Auswertung der dreidimensionalen Temperaturverteilung erfolgt unter Verwendung der Finite-Elemente-Methode. Dabei wird ein dreidimensionaler Körper, der ein möglichst exaktes Modell des in der Realität untersuchten Objekts ist, in eine begrenzte (finite) Anzahl kleinerer Elemente aufgeteilt (Rieg u. a., 2012). Dabei erfolgt eine Gliederung des Gesamtobjekts in verschiedene Materialien durch eine Zuordnung der Elemente zu einem bestimmten Material. Die Elemente können theoretisch beliebige Struktur besitzen. Für diese Umsetzung wurde jedoch ein unregelmäßiger Tetraeder aufgrund seiner einfachen Struktur und dem Vorhandensein entsprechender Bibliotheken (s. 4.4.2) gewählt. Anschließend wird für jedes Element der untersuchte Wert bestimmt. In diesem Fall wird also für jeden Eckpunkt aller Tetraeder ein Temperaturwert ermittelt.
Die Wärmeenergie des Objekts ließe sich für $\Delta T = const.$ nach dem Grundgesetz der Wärmelehre berechnen:

$$Q = m \cdot c \cdot \Delta T \tag{1}$$

Da ΔT jedoch nicht überall im Objekt gleich ist, wird stattdessen die Temperaturänderung in den jeweiligen Unterobjekten als konstant angenommen. Der Durchschnitt der den Tetraeder definierenden Messwerte wird als Temperatur des Elements eingesetzt. ΔT ergibt sich aus der Differenz der Elementtemperatur zu 0 K. Die Wärmeenergie für ein Element ergibt sich somit als:

$$Q_{Element} = m_{Element} \cdot c \cdot \Delta \bar{T}_{Element} \tag{2}$$

Die Gesamtenergie des Objekts lässt sich nun als Summe der Energien der Elemente beschreiben:
$n \ldots$ Anzahl der Elemente

$$Q = \sum_{i=1}^{n} m_{Element_i} \cdot c \cdot \Delta \bar{T}_{Element_i} \tag{3}$$

Dabei wird ausgenutzt, dass sich die Masse eines Elements aus dessen Volumen und der Dichte des Materials berechnen lässt, welche einfacher zu messen sind:

$$Q = \sum_{i=1}^{n} V_{Element_i} \cdot \rho_{Material} \cdot c \cdot \Delta\bar{T}_{Element_i} \tag{4}$$

Je größer die Anzahl der verwendeten Elemente dabei ist, desto genauer kann sich das Ergebnis der Analyse im Rahmen der Ausgangsdaten an die Realität annähern.

3.2 Interpolation

3.2.1 Lineare Interpolation

Um eine Temperaturverteilung innerhalb eines Objekts ermitteln zu können, ist ein Verfahren nötig, mit dem es möglich ist, die Temperatur an einem beliebigen Punkt, ausgehend von den gemessenen Temperaturen zu berechnen. Solche Verfahren bezeichnet man als Interpolationsverfahren. In der Mathematik spricht man von einer Interpolationsfunktion f_1, die mit der Ausgangsfunktion f_0, von der nur bestimmte Punkte $(x_0|f_0(x_0))...(x_n|f_0(x_n))$ bekannt sind, diese Punkte gemein hat. Mithilfe dieser Funktion f_1 kann nun der Wert für ein beliebiges $x \in \mathbb{R}$ ermittelt werden. Wie realistisch dieser Wert für den Sachverhalt ist, der beschrieben wird, hängt vom Typ der Interpolationsfunktion und dem gewählten Definitionsbereich ab. Für den Fall $x < x_0 \vee x > x_n$ wird der ermittelte Wert als extrapolierter Wert bezeichnet.
Eine lineare Interpolationsfunktion beschreibt den Bereich zwischen zwei bekannten Stellen x_0 und x_1 $(x_0 < x_1 < ... < x_n)$ jeweils durch eine lineare Funktion. Der Funktionswert für ein beliebiges $x \geq x_0 \wedge x \leq x_1$ ergibt sich also wie folgt (Zeidler u. a., 2013):

$$f_1(x) = (x - x_0)\frac{f_0(x_1) - f_0(x_0)}{x_1 - x_0} + f_0(x_0) \tag{5}$$

3.2.2 Trilineare Interpolation

Um Werte für beliebige Raumpunkte eines geradlinigen orthogonalen Koordinatensystems statt für Stellen einer Funktion zu berechnen, wird die trilineare Interpolation verwendet. Grundlage dafür ist ein Quader, für dessen Eckpunkte ($C_{000}...C_{111}$, s. Abb. 3) die Werte bekannt sind und dessen Kanten parallel zu den Achsen des Koordinatensystems sind:

Gesucht ist der Wert für den Punkt $C(x|y|z)$. Im ersten Schritt wird nun linear auf den Kanten, die in Richtung der X-Achse verlaufen, interpoliert. Dabei entsprechen die X-Koordinaten der Punkte der Kante den Stellen x_0 und x_1, die gesuchte Stelle x entspricht der entsprechenden

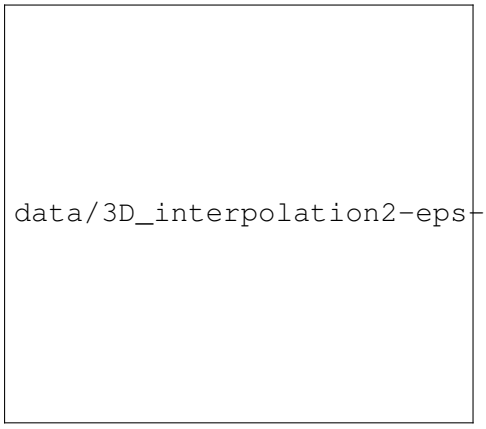


Abbildung 3: Trilineare Interpolation (Marmelad (CC BY-SA 3.0), 2008)

Koordinate des Punktes C. Dadurch ergeben sich vier neue Punkte $C_{00}...C_{11}$, deren Wert durch lineare Interpolation bestimmt wurde und die mit C in einer Ebene liegen und ein Rechteck aufspannen. Die Y-Koordinaten dieser Punkte werden nun im zweiten Schritt zur Interpolation genutzt. Die Stellen x_0 und x_1 der linearen Interpolationsfunktion entsprechen nun den Y-Koordinaten der Punkte der jeweils in Richtung der Y-Achse verlaufenden Kanten. Die gesuchte Stelle der Funktion entspricht dann der Koordinate y von C. Die sich dadurch ergebenden Punkte C_0 und C_1 liegen mit C auf einer geraden in Richtung der Z-Achse. Dementsprechend werden die Z-Koordinaten der Punkte im dritten Schritt zur Interpolation verwendet, wodurch sich der Wert für C bestimmen lässt.

3.2.3 Tetraeder als Ausgangselement

Das Verfahren der trilinearen Interpolation setzt einen Quader als Ausgangselement voraus. Die zu entwickelnde Software wird jedoch mit Tetraedern als Grundelement arbeiten (s. 3.1), weshalb das Interpolationsverfahren auf jene übertragen werden muss. Dafür wird von einem unregelmäßigen Tetraeder mit den Eckpunkten $T_1...T_4$ ausgegangen, gesucht ist der Wert für den Punkt $C(x|y|z)$, der sich innerhalb des Tetraeders befindet (s. Abb. 4). Im ersten Schritt wird wie in 3.2.2 auf den Kanten des Ausgangsobjekts interpoliert. Bei der Interpolation am Tetraeder werden dabei drei Punkte auf drei Kanten, in der Abbildung beispielsweise D_1 auf $\overline{T_1T_2}$, D_2 auf $\overline{T_1T_3}$ und D_3 auf $\overline{T_1T_4}$, ausgewählt, so dass sie mit C in einer Ebene liegen, die zu einer der Tetraederflächen, hier der Grundfläche $\triangle T_2T_3T_4$, parallel ist (wie in Abb. 4 dargestellt).

Um die Berechnung der Punkte zu vereinfachen, wird der Tetraeder in ein entsprechend angepasstes Koordinatensystem mit dem Ursprung T_2 überführt. Die Ortsvektoren der Punkte

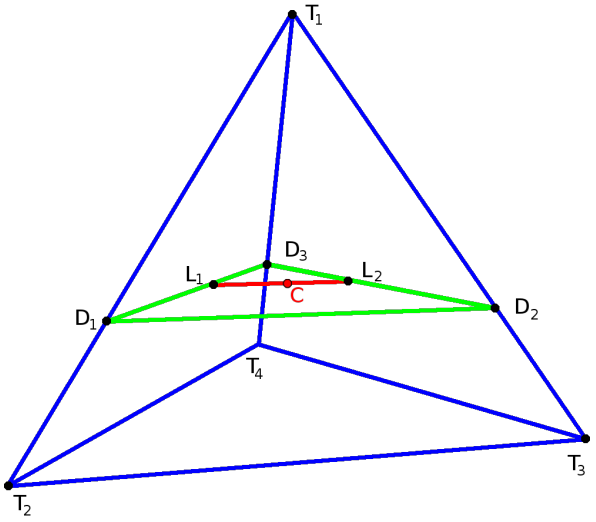


Abbildung 4: Interpolation am Tetraeder

im neuen Koordinatensystem $T_1'...T_4'$ ergeben sich somit wie folgt (die Grundflächennormale $\overrightarrow{N} = \overrightarrow{T_2T_3} \times \overrightarrow{T_2T_4}$ muss in Richtung der Tetraederspitze zeigen):

Winkel zwischen Grundflächennormalen und $\overrightarrow{T_1T_2}$:

$$\cos(\alpha) = \frac{\overrightarrow{T_1T_2} \cdot \overrightarrow{N}}{|\overrightarrow{T_1T_2}| \cdot |\overrightarrow{N}|}$$

Projektion von T_1 entlang der Grundflächennormalen auf die Grundfläche:

$$\overrightarrow{OT}_{proj} = \overrightarrow{T_2T_1} - \frac{\overrightarrow{N}}{|\overrightarrow{N}|} \cdot \cos(\alpha) \cdot |\overrightarrow{T_2T_1}|$$

Lot von T_4 auf $\overrightarrow{T_2T_3}$:

$$\overrightarrow{l} = \overrightarrow{T_2T_3} \times \overrightarrow{N}$$
$$\overrightarrow{l} = \overrightarrow{l} \cdot \text{sign}(\overrightarrow{l} \cdot \overrightarrow{OT}_{proj})$$

Winkel zwischen $\overrightarrow{T_1T_2}$ und $\overrightarrow{T_2T_{proj}}$. Um Mehrdeutigkeiten bei der Position der Spitze über der Grundfläche des Tetraeders auszuschließen, ist die Belegung des Winkels mit einem Vorzeichen nötig:

$$\beta = \arccos\left(\frac{\overrightarrow{T_2T_1} \cdot \overrightarrow{OT}_{proj}}{|\overrightarrow{T_2T_1}| \cdot |\overrightarrow{OT}_{proj}|}\right) \cdot \text{sign}(\overrightarrow{l} \cdot \overrightarrow{OT}_{proj})$$

$$\Longrightarrow \qquad \overrightarrow{OT_1'} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \tag{6a}$$

$$\overrightarrow{OT_2'} = \begin{pmatrix} |\overrightarrow{T_2T_3}| \\ 0 \\ 0 \end{pmatrix} \tag{6b}$$

$$\overrightarrow{OT_3'} = \begin{pmatrix} \cos(\angle T_3T_2T_4) \cdot |\overrightarrow{T_2T_4}| \\ \sin(\angle T_3T_2T_4) \cdot |\overrightarrow{T_2T_4}| \\ 0 \end{pmatrix} \tag{6c}$$

$$\overrightarrow{OT_4'} = \begin{pmatrix} \cos(\beta) \cdot |\overrightarrow{OT}_{proj}| \\ \sin(\beta) \cdot |\overrightarrow{OT}_{proj}| \\ \cos(\alpha) \cdot |\overrightarrow{T_2T_1}| \end{pmatrix} \tag{6d}$$

Der Ortsvektor zum gesuchten Punkt lässt sich im Koordinatensystem des Tetraeders wie folgt beschreiben:

Winkel zwischen Grundflächennormale und $\overrightarrow{T_2C}$:

$$\cos(\gamma) = \frac{\overrightarrow{T_2C} \cdot (\overrightarrow{N})}{|\overrightarrow{T_2C}| \cdot |\overrightarrow{N}|} \tag{7}$$

Höhe des gesuchten Punktes über der Grundfläche:

$$h = \cos(\gamma) \cdot |\overrightarrow{T_2C}| \tag{8}$$

Projektion von C auf die Grundfläche:

$$\overrightarrow{OC_{proj}} = \overrightarrow{T_2C} - \frac{\overrightarrow{N}}{|\overrightarrow{N}|}h \tag{9}$$

Winkel zwischen projiziertem Punkt $\overrightarrow{OC_{proj}}$ und der Kante $\overrightarrow{T_2T_3}$ mit Vorzeichen zur Unterscheidung der Position des Punktes in innerhalb und außerhalb der Grundfläche. Die Verwendung des Vorzeichens ermöglicht die Anwendung des Verfahrens auch auf Punkte C außerhalb des Tetraeders:

$$\delta = \arccos\left(\frac{\overrightarrow{T_2T_3} \cdot \overrightarrow{OC_{proj}}}{|\overrightarrow{T_2T_3}| \cdot |\overrightarrow{OC_{proj}}|}\right) \cdot \text{sign}(\overrightarrow{l} \cdot \overrightarrow{OC_{proj}}) \tag{10}$$

$$\Longrightarrow \qquad \overrightarrow{OC'} = \begin{pmatrix} \cos(\delta) \cdot |\overrightarrow{OC_{proj}}| \\ \sin(\delta) \cdot |\overrightarrow{OC_{proj}}| \\ h \end{pmatrix} \tag{11}$$

Vom transformierten Tetraeder ausgehend werden nun die Ortsvektoren der zur Interpolation verwendeten Punkte $D_1 ... D_3$ ermittelt, ϵ sei dabei jeweils der Winkel zwischen der Kante $\overrightarrow{T'_{1+n}T'_1}$ ($n \in \{1, 2, 3\}$) und der Normale der Grundfläche, die im Koordinatensystem des Tetraeders $T'_1 ... T'_4$ gleich $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ ist.

$$\overrightarrow{OD_n} = \overrightarrow{OT'_{1+n}} + \frac{\overrightarrow{T'_{1+n}T'_1}}{|\overrightarrow{T'_{1+n}T'_1}|} \cdot \frac{h}{\cos(\epsilon)} \tag{12}$$

Der interpolierte Wert für diesen Punkt beträgt:

$$f_1(D_n) = f_0(T_1 + n) + (f_0(T_1) - f_0(T_1 + n)) \cdot \frac{h}{\cos(\epsilon)|\overrightarrow{T'_{1+n}T'_1}|} \tag{13}$$

Zur Vereinfachung der Betrachtung des $\triangle D_1 D_2 D_3$ wurde dieses in ein zweidimensionales, orthogonales, geradliniges Koordinatensystems überführt, dessen Ursprung D_1 ist. In diesem lassen sich die Ortsvektoren der Punkte $D'_1 ... D'_3$ wie folgt berechnen:

$$\overrightarrow{OD'_1} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{14a}$$

$$\overrightarrow{OD'_2} = \begin{pmatrix} |\overrightarrow{D_1D_2}| \\ 0 \end{pmatrix} \tag{14b}$$

$$\overrightarrow{OD'_3} = \begin{pmatrix} \cos(\angle D_3D_1D_2) \cdot |\overrightarrow{D_1D_3}| \\ \sin(\angle D_3D_1D_2) \cdot |\overrightarrow{D_1D_3}| \end{pmatrix} \tag{14c}$$

Der Ortsvektor des gesuchten Punktes ist dementsprechend:

$\angle CD_1D_2$ (Um den Definitionsbereich für C auf Punkte außerhalb des Dreiecks erweitern zu können, ist wieder die vorzeichenrichtige Betrachtung des Winkels nötig):

$$\alpha' = \arccos\left(\frac{\overrightarrow{D_1D_2}\cdot\overrightarrow{D_1C}}{|\overrightarrow{D_1D_2}|\cdot|\overrightarrow{D_1C}|}\right)\cdot\operatorname{sign}\left(\overrightarrow{D_1C}\cdot\begin{pmatrix}1\\0\\0\end{pmatrix}\right)$$

$$\Longrightarrow \qquad \overrightarrow{OC''} = \begin{pmatrix}\cos(\alpha')\cdot|\overrightarrow{D_1C}|\\ \sin(\alpha')\cdot|\overrightarrow{D_1C}|\end{pmatrix} \tag{15}$$

Auf zwei Seiten des transformierten Dreiecks $\triangle D_1'D_2'D_3'$ werden nun Werte für die Punkte L_1 und L_2 interpoliert, deren Ortsvektoren wie nachfolgend beschrieben ermittelt werden (tiefgestellte Buchstaben markieren das Element des Vektors für die X- bzw. Y-Achse):

$$\overrightarrow{OL_1} = \begin{pmatrix}\frac{(\overrightarrow{OC''})_y}{\tan(\angle D_3'D_1'D_2')}\\ (\overrightarrow{OC''})_y\end{pmatrix} \tag{16a}$$

$$\overrightarrow{OL_2} = \begin{pmatrix}(\overrightarrow{D_1'D_2'})_x - \frac{(\overrightarrow{OC''})_y}{\tan(\angle D_1'D_2'D_3')}\\ (\overrightarrow{OC''})_y\end{pmatrix} \tag{16b}$$

Die durch lineare Interpolation ermittelten Werte für die Punkte ergeben sich somit als:

$$f_1(L_1) = f_0(D_1') + (f_0(D_3') - f_0(D_1'))\cdot\operatorname{sign}((\overrightarrow{OC''})_y)\cdot\frac{|\overrightarrow{OL_1'}|}{|\overrightarrow{D_1'D_3'}|} \tag{17a}$$

$$f_1(L_2) = f_0(D_2') + (f_0(D_3') - f_0(D_2'))\cdot\operatorname{sign}((\overrightarrow{OC''})_y)\cdot\frac{|\overrightarrow{OL_2'} - \overrightarrow{D_1'D_2'}|}{|\overrightarrow{D_2D_3}|} \tag{17b}$$

Durch die anschließende Interpolation zwischen L_1 und L_2 kann schließlich der Wert für C bestimmt werden:

$$f_1(C) \hat{=} f_1(\overrightarrow{OC''}) = f_1(L_1) + (f_1(L_2) - f_1(L_1))\frac{(\overrightarrow{OC''})_x - L_1x}{L_2x - L_1x} \tag{18}$$

4 Konzept

4.1 Primär unterstütztes Betriebssystem

Grundsätzlich ist es zwar erstrebenswert, dass die Endversion der Software auf möglichst vielen Systemen lauffähig ist, dies wäre jedoch zumeist mit erheblich größerem Entwicklungs- und Testaufwand verbunden. Als primär unterstütztes Betriebssystem wurde daher Debian GNU/Linux und im Besonderen sein Derivat Ubuntu gewählt, da es frei verfügbar ist und so die Nutzung des Programms nicht an eine kostspielige Lizenz für ein Betriebssystem gebunden ist. Dazu vereinfacht das Paketsystem von Debian die Installation der Software mit sämtlichen eventuell benötigten Zusatzprogrammen oder Bibliotheken.

4.2 Lizenz

Um die freie Verfügbarkeit für jene, die an seiner Nutzung im Sinne des Wissensgewinns oder der Weiterentwicklung des Programms interessiert sind, sicherzustellen, soll die fertige Software unter einer im Verständnis der Free Software Foundation freien Lizenz veröffentlicht werden. Dies wären beispielsweise die GNU General Public License (GPL) oder ein Derivat. Die GNU GPL stellt sicher, dass Endnutzern das Recht der Nutzung (auch kommerziell), Weitergabe und Modifikation erhalten bleibt. Derivate des Programms müssen also ebenfalls unter der GNU GPL veröffentlicht werden. Ob diese Lizenz anwendbar ist, hängt von der Kompatibilität zu Lizenzen eventuell verwendeter Bibliotheken ab.

4.3 Trennung von Datenerhebung und Analyse

Aufgabe der Software ist es auch, Messfehler in den Daten der Messgeräte zu korrigieren und diese zusammenzuführen. Als Messfehler werden Werte angesehen, die zu stark vom Vorherigen abweichen und deshalb unplausibel sind. Durch diese Heuristik gefundene Werte werden durch interpolierte Werte ersetzt (s. 5.2.2).
Da die Ausgangsdaten in verschiedenen Formaten vorliegen, müssen sie zuerst in ein einheitliches Format umgewandelt werden, in welchem sie anschließend in einen Datensatz zusammengeführt

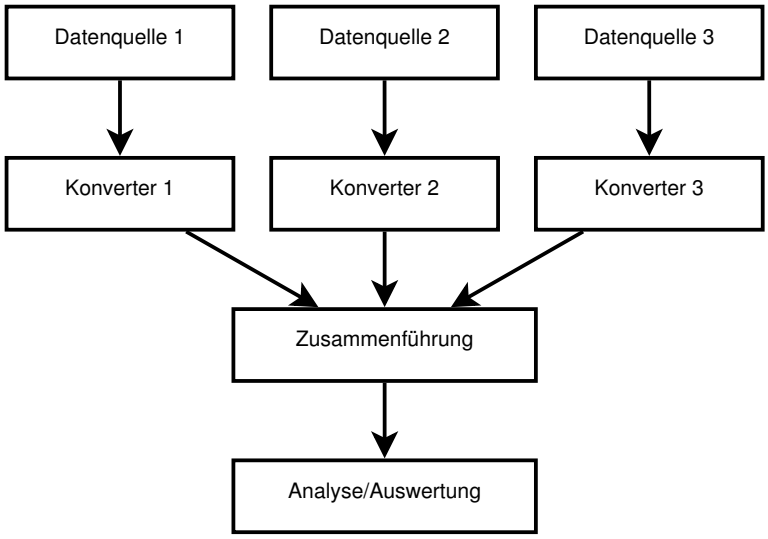


Abbildung 5: Ablauf der Datenverarbeitung

werden können. Würde dieser Vorgang geschlossen von einem einzelnen Programm durchgeführt, müssten stets alle Ausgangsdaten bereits vorliegen, ein späteres Einfügen weiterer Datensätze wäre nicht möglich. Für eine Erweiterung um neue Datenquellen müsste das gesamte Programm verändert werden. Um dies zu vermeiden, sollten Programme zum Umwandeln der Daten (Konverter) aus den jeweiligen Quellen sowie ein Werkzeug zum Zusammenfügen der so gewonnenen (Teil-)Datensätze als eigenständige Programme entwickelt werden (s. Abb. 5).
Dadurch vermindert sich auch die Komplexität der Einzelprogramme, eine grafische Benutzeroberfläche ist daher nicht zwingend zur einfachen Bedienbarkeit erforderlich. Eine Bedienung über Kommandozeilenparameter ermöglicht zudem eine einfache Automatisierung über Shell-Skripts. Die so erstellten Datensätze können anschließend von einem Analyseprogramm mit GUI (graphical user interface - grafische Benutzeroberfläche) eingelesen und verarbeitet werden.
Für den in 2.2 beschriebenen Versuch werden Daten durch faseroptische Sensoren, ODiSI-Messfasern und eine Wärmebildkamera erhoben, wobei faseroptische Sensoren und die Wärmebildkamera das CSV-Format nutzen. Die ODiSI-Daten liegen jedoch in einem eigenen Format vor. Es werden also ein CSV- und ein ODiSI-Koverter benötigt, um alle erhobenen Messdaten auswerten zu können.

4.4 Verwendete Software und Bibliotheken

4.4.1 C++ als Programmiersprache

Die Entwicklung eines stabilen und benutzerfreundlichen Programms ist in vielen Programmiersprachen möglich. Interpretierte oder teilweise interpretierte Sprachen wie PHP oder Python bieten zwar eine hohe Stabilität, die Lauffähigkeit des Programms ist jedoch vom Vorhandensein der erforderlichen Version des Interpreters abhängig und die Ausführung zumeist weniger schnell, was bei rechenintensiven Aufgaben wie der Finite-Elemente-Berechnung von Nachteil ist. Daher wird C++ als am weitesten verbreitete, umfangreiche und auf fast allen Plattformen unterstützte Sprache bei der Entwicklung der Software verwendet. Dabei kommt der bisher neueste Standard C++11 zum Einsatz.

4.4.2 TetGen als Finite-Elemente-Generator

TetGen ist ein Programm zur Generierung von Tetraeder-Netzen von durch Polygone definierten Volumen und kann auch als Bibliothek genutzt werden. Hier wird es verwendet, um Eingabeobjekte in Netze mit einstellbarer Genauigkeit in Tetraeder zu zerlegen, die anschließend als Elemente für die weitere Finite-Elemente-Analyse (s. 3.1) genutzt werden. Die zur Zeit aktuellste Version ist Version 1.5.0 vom 4. November 2013, die u.a. unter der GNU Affero General Public License 3 oder höher veröffentlicht wurde (Si, 2013).

4.4.3 wxWidgets als GUI-Bibliothek

Da das Analyseprogramm eine grafische Benutzeroberfläche besitzen soll, ist es sinnvoll, für deren Entwicklung eine entsprechende Bibliothek zu verwenden. Diese Bibliothek stellt Datentypen, Klassen und Makros zur Verfügung, mit denen das Programm ein interaktives Fenster zur Steuerung der Programmfunktionen anzeigen kann, ohne dass bei der Programmierung direkt auf die API[1] des Betriebssystems zugegriffen werden muss. wxWidgets zeichnet sich dadurch aus, dass ein Programm weitestgehend plattformunabhängig geschrieben werden kann, die Oberfläche jedoch auf die entsprechenden Standardkomponenten des Systems zurückgreift. So erkennt der Benutzer die Bedienelemente seines Betriebssystems wieder und weiß bereits, wie diese funktionieren. Die Bibliothek wird in der Version 2.8 eingesetzt, wie sie in den Paketquellen für Ubuntu 12.04 und später enthalten ist.

[1] application programming interface (Programmierschnittstelle)

4.4.4 pBuilder als Paketerstellungswerkzeug

In Debian-basierten Betriebssystemen wird Software in Paketen verwaltet, die neben dieser auch Informationen über Version, Lizenz und Abhängigkeiten zu anderen Programmen in komprimierter Form enthalten. Liegt die Software in Quelltextform vor, handelt es sich um ein Quellpaket. Aus diesem lässt sich ein architekturspezifisches Binärpaket erstellen, das ohne große Vorkenntnisse mithilfe eines Paketmanagers auf dem System installiert werden kann und so die enthaltenen Programme bereitstellt. Auch durch das automatische Installieren der angegebenen Abhängigkeiten wird die korrekte Installation erheblich vereinfacht. Durch die klare Organisation als Paket sind einfaches Aktualisieren oder Entfernen der Software möglich. Zur Erzeugung dieser Pakete kann ein Paketerstellungswerkzeug, beispielsweise pBuilder, verwendet werden. Dieses automatisiert große Teile des Erstellungsprozesses und schließt Fehler in den Abhängigkeiten des Pakets durch die Verwendung eines virtuellen Systems weitgehend aus.

5 Implementierung

Wie in 4.3 dargestellt, wird die Funktionalität der Software in Unterprogramme gegliedert. Diese werden für die Weitergabe als Softwarepaket zusammengefasst (s. 4.4.4), sind aber getrennt lauffähig. Die einzelnen Programme haben folgende Funktionen:

Tabelle 1: Die Einzelprogramme und ihre Funktionen

Programm	Funktion
csvtosd	Konverter zum Umwandeln von Messdaten im CSV-Format in das TSD-Format.
odisitosd	Konverter zum Umwandeln von ODiSI-Messdaten in das TSD-Format. Zusätzlich beinhaltet es eine Heuristik zur Messfehlerkorrektur.
mergetsd	Werkzeug zum Zusammenfügen von Messwertsätzen im TSD-Format.
simpleanalyzer-gui	Programm mit grafischer Oberfläche zur Analyse, Visualisierung und Aufbereitung der Messdaten.

Im Folgenden werden das Dateiformat für Sensordaten und die Funktionsweise der entwickelten Programme erläutert. Der Aufbau programmspezifischer Konfigurations- oder Eingabedateien sowie die Verwendung der Programme ist im beiliegenden Handbuch (Anlage 3) beschrieben. Implementationsdetails sind dagegen der HTML-Dokumentation (Anlage 4) oder dem Quellcode (Anlage 2) zu entnehmen.

5.1 SD- und TSD-Formate

Ein einheitliches Dateiformat für Sensordaten muss im einfachsten Fall Koordinaten und Daten der Sensoren enthalten. Dieses einfachste Dateiformat wird hier als SD- bzw. .sd-Format (für "sensor data") bezeichnet. Eine Datei dieses Formats darf ausschließlich ASCII-Zeichen enthalten, muss mit UNIX-Zeilenenden (\n) versehen sein und ist wie folgt aufgebaut:

```
#Kommentar
s X Y Z V
```

```
#zum Beispiel:
s -1.0 1.0 0.0 100.0
```

Ist eine Zeile leer oder mit einer Raute als erstem Zeichen versehen (Kommentar), wird sie nicht interpretiert. Anderenfalls enthält eine Zeile Informationen über je einen Messpunkt. Eine solche Zeile beginnt stets mit dem Buchstaben "s", gefolgt von den kartesischen Koordinaten (X,Y,Z) und dem Messwert des entsprechenden Sensors (V), jeweils durch Leerzeichen getrennt. Als Dezimaltrennzeichen wird ein Punkt verwendet. Da so jedoch nur ein Datensatz gespeichert werden kann, ist dieses Format ausschließlich zum Speichern eines bestimmten einzelnen Datensatzes oder zu Testzwecken sinnvoll einsetzbar. Daher wurde das Format zum TSD- bzw. .tsd-Format (für "‘timed sensor data"’) mit folgendem Aufbau erweitert:

```
#Kommentar
t TIMESTAMP
n NAME
s X Y Z V
s X Y Z V

#zum Beispiel:
t 1
n Datensatz_1
s -1.0 1.0 0.0 100.0
s 1.0 -1.0 0.0 0.0
```

Einzelne Datensätze werden nun mit vorangestelltem Zeitstempel (TIMESTAMP) und Namen (NAME) versehen. Zeitstempel und Name werden dabei durch "t" bzw. "n" als erstes Zeichen der Zeile gefolgt von einem Leerzeichen und dem Wert definiert. Der Zeitstempel wird bei der späteren Verarbeitung als Ganzzahl betrachtet. Anschließend folgen die Messpunktdefinitionen wie oben beschrieben, bis der nächste Zeitstempel einen neuen Datensatz einleitet. So können sämtliche Daten des Versuchs mit ihrer zeitlichen Einordnung in einer Gesamtdatei gespeichert werden.

5.2 Implementation der Konverterprogramme

Um die Daten aus sämtlichen Datenquellen vereinigen zu können, müssen diese jeweils in das TSD-Format überführt werden (s. 4.3). Diese Datenquellen sind für den in 2.2 beschriebenen Versuchsaufbau faseroptische Sensoren, Wärmebildkamera und ODiSI-Messystem. Die Einheiten der Messwerte werden dabei nicht betrachtet, es wird lediglich der vorliegende Zahlenwert

übertragen. Dies ist nicht anders möglich, da in den Ausgangsdaten keine oder nur uneinheitliche Informationen über die verwendeten Einheiten vorhanden sind. Wenn der Benutzer sich also gegen die ausschließliche Verwendung von SI-Einheiten entscheidet, muss er selbst auf die Einheiten der Ergebnisse achten. Anderenfalls sind alle Werte in SI-Einheiten angegeben, es sei denn, eine andere Einheit wird explizit genannt.
Informationen zum Aufbau von Konfigurations-, Eingabe- und Sensordefinitionsdateien sind dem Handbuch zu entnehmen.

5.2.1 Funktionsweise des CSV-Konverters (csvtosd)

Wie bereits beschrieben, liegen die Ausgangsdaten für Wärmebildkamera und faseroptische Sensoren als CSV-Datei ("character-separated values") vor, wobei die Koordinaten der Messpunkte in einer externen Sensordefinitionsdatei gespeichert sind. Im Folgenden wird die Funktion der vom Programm verwendeten Dateien erläutert, deren Struktur ist dem Handbuch zu entnehmen.

Tabelle 2: Funktionen der benötigten Dateien

Datei	Funktion
Konfigurationsdatei	Um das Konverterprogramm möglichst flexibel an Messdaten beliebiger Geräte anpassen zu können, sind allgemeine Parameter über die Struktur und die Verarbeitung der Eingabedaten in einer Konfigurationsdatei festgelegt.
Eingabedatei	Die CSV-Eingabedatei enthält die Messdaten der Sensoren über die gesamte Zeitspanne des Versuchs. Sie enthält jedoch nicht die Koordinaten der Messpunkte, diese sind in der Sensordefinitionsdatei festgelegt.
Sensordefinitionsdatei	Die Festlegung der Koordinaten der Messpunkte im kartesischen Koordinatensystem erfolgt in der Sensordefinitionsdatei. Sie wird nicht von den verwendeten Messgeräten erzeugt und muss daher manuell erstellt werden. In der Regel muss dies für jeden Versuchsaufbau neu geschehen, Versuche mit derselben experimentellen Anordnung erlauben oft die Wiederverwendung der Sensordefinitionsdatei.

Der Durchlauf des Konverters (s. Abb. 6) beginnt mit dem Einlesen der Konfigurationsdatei und dem Speichern der dort festgelegten Einstellungen in einer klassenweit verfügbaren Struktur. Anschließend werden die beim Programmaufruf übergebenen Argumente ausgelesen und

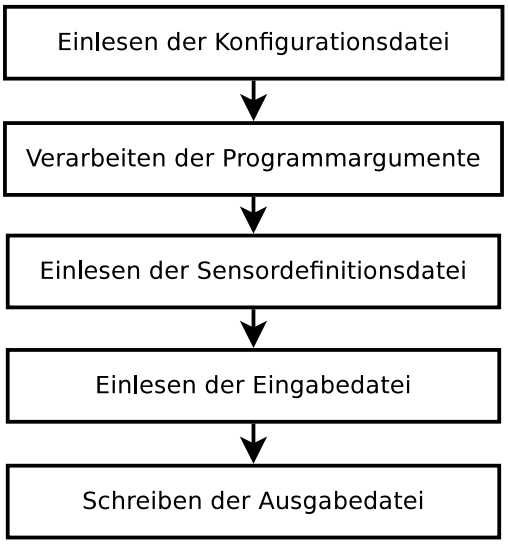


Abbildung 6: Programmablauf csvtosd

gespeichert. Soll der Hilfetext ausgegeben werden (Argument *-h*), folgt nach dieser Ausgabe der Abbruch des Programms. Anderenfalls wird mit dem Lesen der Sensordefinitionen fortgefahren. Anschließend werden alle Datensätze mit Zeitstempel, Name und Wert für die vorher definierten Sensoren aus der Eingabedatei gelesen. Diese Daten werden abschließend im TSD-Format in die Ausgabedatei geschrieben.

5.2.2 Funktionsweise des ODiSI-Konverters (odisitosd)

Die Struktur der Messwertdateien des ODiSI-Messsystems ist keine CSV-Datei. Daher wird für diese ein eigener Konverter benötigt. Die Funktion der hier verwendeten Dateien ist ähnlich zu der des CSV-Konverters, doch ihre Struktur unterscheidet sich (s. Handbuch). Bei den Messwerten handelt es sich hier nicht um Werte einzelner Sensoren, sondern um Temperaturen an verschiedenen Positionen entlang der Länge der Faser. Die Sensordefinitionsdatei beschreibt somit nicht mehr die Position der einzelnen Messpunkte, sondern den Verlauf der Faser im untersuchten Material durch Festlegung von Ein- und Austrittspunkten sowie Richtung des Faserverlaufs. Die erhöhte Anfälligkeit des Messsystems für zufällige Messfehler macht hier eine Heuristik zur Messwertkorrektur erforderlich, deren Ergebnisse in einer optionalen Logdatei protokolliert werden.

Auch bei diesem Konverter beginnt der Programmablauf mit dem Einlesen der Ausgangsdateien und dem Verarbeiten der Programmargumente (s. Abb. 7). Beim Schreiben der Ausgabedatei können die Werte jedoch nicht unverändert in das TSD-Format umgeschrieben werden. Stattdessen werden bei der Verarbeitung eines Datensatzes zuerst die Messstellen der Faser aussortiert,

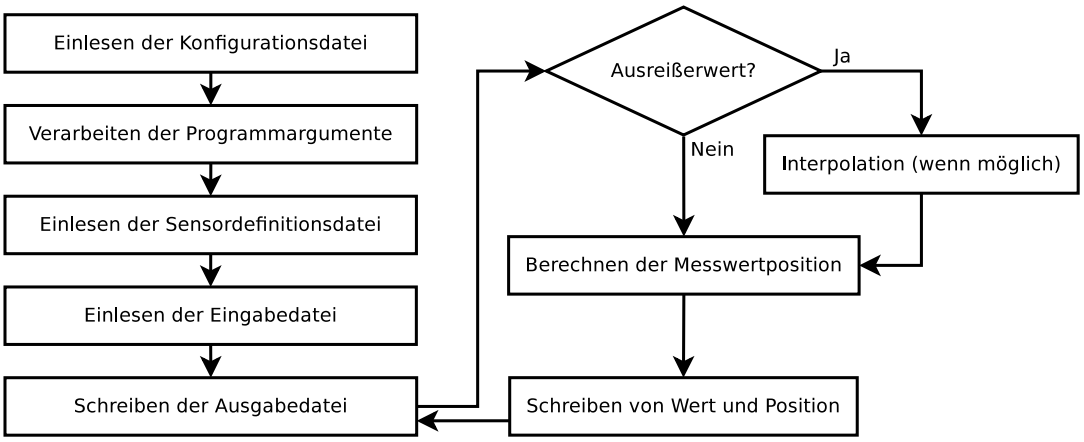


Abbildung 7: Programmablauf odisitosd

die außerhalb des untersuchten Objekts, also vor einem Fasereingang oder nach einem Faserausgang, liegen. Die übrigen Werte werden heuristisch auf Messfehler in Form von "Ausreißerwerten" geprüft. Das sind Werte, die eine unplausibel große Differenz zu den umgebenden Werten aufweisen und deshalb herausstechen. Dazu wird jeweils mit dem Vorgängerwert verglichen. Überschreitet die Wertedifferenz eine bestimmte in der Konfigurationsdatei festgelegte Schwelle, wird der Wert der entsprechenden Messstelle im nächsten Datensatz gesucht und auf Plausibilität geprüft. Der maximal zulässige zeitliche Abstand ist dabei proportional zum zeitlichen Abstand des Datensatzes vom ursprünglich untersuchten Datensatz. Wird ein plausibler Wert gefunden, werden die Ausreißerwerte dazwischen durch zwischen dem Vorgängerwert des untersuchten Datensatzes und dem gefundenen plausiblen Wert linear interpolierte Werte ersetzt. Die maximal zulässige zeitliche Entfernung eines zum Interpolieren verwendeten Datensatzes vom untersuchten Datensatz kann ebenfalls in der Konfigurationsdatei festgelegt werden. Dabei müssen die Korrekturparameter so gewählt werden, dass sie auf den jeweiligen Anwendungsfall passen: Ist die Toleranz zu groß, zeigt die Fehlerkorrektur kaum Wirkung, ist sie zu klein, werden korrekte Messergebnisse verfälscht. Wird ein Wert korrigiert oder kein gültiger Datensatz in der festgelegten zeitlichen Entfernung gefunden, schreibt das Programm eine Bemerkung in die Logdatei, sofern ein Pfad für diese angegeben wurde.
Nach der Messwertkorrektur folgt die Berechnung der Position der Messpunkte auf der Faser im kartesischen Koordinatensystem. Die Faser liegt dabei stets in einer zur X-Z-Ebene parallelen Ebene und alle Ein- und Ausgänge der Faser liegen auf zwei zur X-Achse parallelen Geraden in dieser Ebene. Die Faserhöhe (y) ist für alle Messwerte konstant und kann als Programmargument übergeben werden. Für eine Messstelle an der Faserlänge l_{pos} seien x_{in} und x_{out} die X-Koordinaten des umgebenden Fasereingangs und -ausgangs, l_{in} und l_{out}, deren Position auf

der Messfaser als Faserlänge. Da die Faser durch Bohrlöcher verläuft, kann ein gerader Verlauf im Objekt angenommen werden. Die Koordinaten des Messpunkts ergeben sich somit als:

Winkel der Faser zur Z-Achse:

$$\sin(\alpha) = \frac{x_{out} - x_{in}}{l_{out} - l_{in}} \tag{19}$$

$$x = x_{in} + \sin(\alpha)(l_{pos} - l_{in}) \tag{20}$$

$$y = const. \tag{21}$$

Wenn die Faser mit der Z-Achse verläuft:

$$z = \cos(\alpha)(l_{pos} - l_{in}) \tag{22}$$

Wenn die Faser entgegen der Z-Achse verläuft:

$$z = \cos(\alpha)(l_{out} - l_{in}) - \cos(\alpha)(l_{pos} - l_{in}) \tag{23}$$

Nun können die Messwerte mit Position und Wert in die Ausgabedatei geschrieben werden. Zeitstempel und Namen für die Datensätze werden auch hier aus den für diesen Zweck angegebenen Spalten der Eingabedatei entnommen.

5.3 Implementation des Zusammenführungswerkzeugs (mergetsd)

Die durch die Konverter gewonnenen Messwertsätze müssen, um diese in ihrer Gesamtheit bei der Auswertung des Versuchs nutzen zu können, zu einem gesamten Datensatz zusammengeführt werden. Dabei ist die zeitliche Synchronisation der jeweils zusammengeführten Datensätze wichtig. Mergetsd benötigt keine Konfigurationsdatei, daher wird sofort mit dem Einlesen der Programmargumente begonnen (s. Abb. 8).

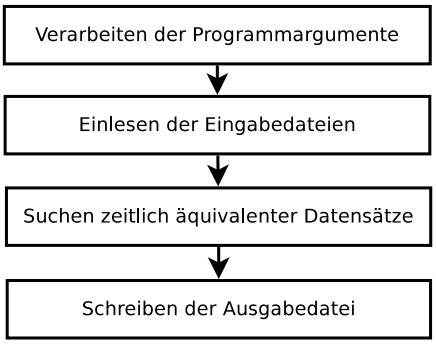


Abbildung 8: Programmablauf mergetsd

Es können bei jedem Durchlauf zwei Messwertsätze zusammengeführt werden. Wenn mehr TSD-Dateien vorliegen, müssen diese durch Kombination jeweils zweier Dateien zu einer Gesamtdatei vereinigt werden. Nach dem Einlesen der Eingabedateien folgt die Suche zeitlich äquivalenter Datensätze. In der Standardeinstellung wird angenommen, dass die jeweils ersten Datensätze der Ausgangsdateien gleichzeitig aufgenommen worden sind. Wird die Option *-auto-offset* nicht explizit deaktiviert, wird die Zeitdifferenz von der ersten zur zweiten Datei (Δt_{auto}) auf die Zeitstempel der zweiten Datei addiert, wodurch die jeweils ersten beiden Datensätze eine effektive Zeitdifferenz von 0 haben. Wurde ein zusätzlicher Versatz (Δt_{custom}) durch *-offset* übergeben, wird dieser zum Zeitstempel der zweiten Datei addiert. Zwei Datensätze gelten dann als äquivalent, wenn ihre Abweichung voneinander kleiner oder gleich der vom Nutzer als maximal zulässig angegebenen Zeitdifferenz Δt_{max} (*-max-dt*) ist. Es muss also gelten:

$$\Delta t_{max} \geq |t_1 - (t_2 + \Delta t_{auto} + \Delta t_{custom})| \tag{24}$$

Ist dies der Fall, werden die Daten beider Datensätze kombiniert, indem die Daten der Messpunkte aus beiden Datensätzen hintereinander in einen neuen Datensatz geschrieben werden. Dabei werden Zeitstempel und Name des Ausgangsdatensatzes aus der ersten Datei übernommen.

5.4 Implementation des Analyse- und Visualisierungsprogramms (simpleanalyzer-gui)

5.4.1 Laden von Eingabedateien und Datenhaltung

Für die Berechnung einer dreidimensionalen Temperaturverteilung sind neben den Sensordaten Informationen über die Geometrie des untersuchten Objekts nötig. Da dreidimensionale Objekte mit mehreren Materialien untersucht werden, wird das Versuchsobjekt als Gesamtheit von für jedes Material einzelnen, geschlossenen Polygonobjekten dargestellt. Aufgrund der einfachen Struktur, des hohen Verbreitungsgrades und der Editierbarkeit mit einem Texteditor wurde das von Wavefront Technologies entwickelte OBJ-Format (Wavefront .obj) als Speicherformat für die Objektgeometrie gewählt. Des Weiteren ermöglicht es die Einbindung einer Materialbibliothek (.mtl), in der Materialeigenschaften wie Dichte oder spezifische Wärmekapazität gespeichert werden können.
Die Funktionalität zum Laden von Sensordaten und Objektgeometrie ist in der Klasse *Importer* implementiert. Sowohl Sensordaten als auch Materialinformationen und Geometrie werden bereits beim Importieren mit dem an die Importmethode übergebenen Objekt verknüpft. Für die Datenhaltung kommen verschiedene Strukturen zum Einsatz (s. Abb. 9): Objektinformationen wie Name oder Zerlegungsparameter werden in der Klasse *ObjectData* gespeichert. Da

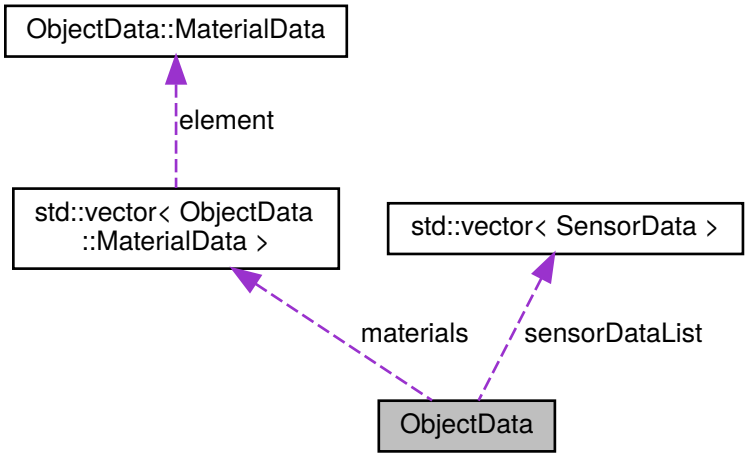


Abbildung 9: Zu *ObjectData* verknüpfte Klassen

ein Objekt aus mehreren Materialien bestehen kann und die jeweiligen Einzelobjekte nur einem Material zugehörig sind, sind diese nur mit den entsprechenden Materialeigenschaften verknüpft. Gespeichert werden die Geometrien der Einzelobjekte im von der Tetgen-Bibliothek (s. 4.4.2) verwendeten Format (*tetgenio*). Auch nach dem Laden der Objektgeometrie können weitere Sensordatensätze (SD- oder TSD-Dateien) mit dem Objekt verknüpft werden, wobei der Index des aktuell verwendeten Sensordatensatzes in *ObjectData* gespeichert ist. Die Objektgeometrie ist jedoch unveränderlich.
Da die geladenen Objekte und ihre zugehörigen Daten programmweit verfügbar sein müssen, werden Verweise auf alle Instanzen von *ObjectData* in einer globalen Liste gespeichert.

5.4.2 Berechnung der dreidimensionalen Temperaturverteilung

Die Grundlage für die weitere Auswertung und Aufbereitung der Messdaten ist die Berechnung einer dreidimensionalen Temperaturverteilung über das Versuchsobjekt. Es muss also für hinreichend viele Punkte der Geometrien aller Materialien eine Temperatur ermittelt werden. Dazu werden zuerst alle Teilobjekte der Materialien mittels der Tetgen-Bibliothek so in Tetraeder zerlegt, dass ihr gesamtes Volumen durch Tetraeder eines bestimmten maximalen Volumens ausgefüllt ist, wie im Struktogramm in Abb. 10 beschrieben. Das so verarbeitete Teilobjekt wird zusätzlich zur Originalgeometrie als *tetgenio*-Objekt (Tetgen-Datenstruktur) gespeichert und mit den Materialeigenschaften verknüpft. Anschließend wird in den Materialeigenschaften eine Liste (ggf. neu) angelegt, in der später gespeichert wird, ob ein Punkt des zerlegten Teilobjekts extrapoliert oder interpoliert ist. Die ermittelten Temperaturen für die Punkte können dagegen in der Tetgen-Datenstruktur gespeichert werden. Nun wird für alle Punkte des zum

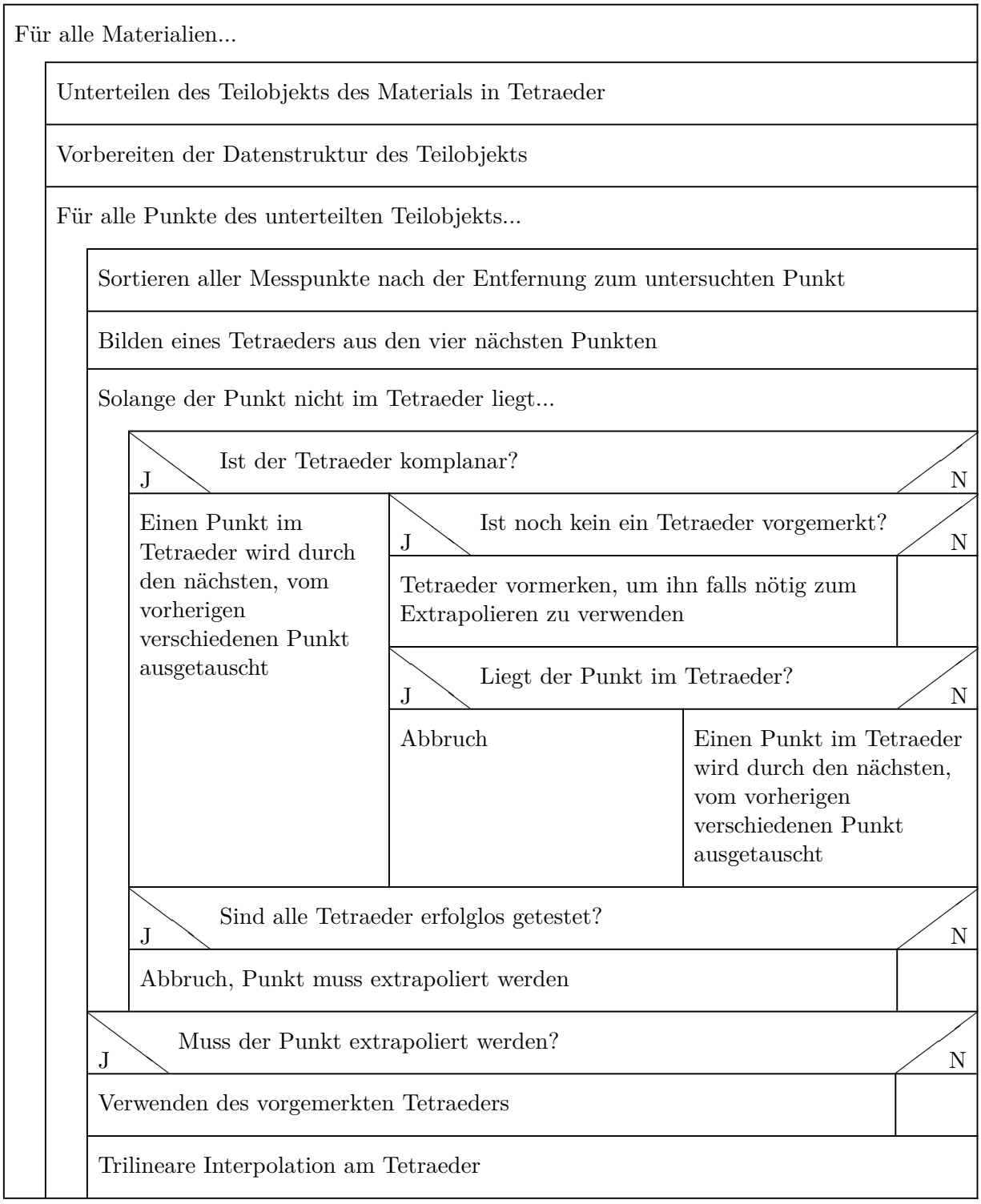


Abbildung 10: Ablauf der Berechnung der Temperaturverteilung

Material gehörigen Objekts ein Temperaturwert ermittelt, indem versucht wird, aus der Kombination von vier Messpunkten einen Tetraeder zu finden, der den gesuchten Punkt enthält. Dabei werden zuerst die Messpunkte nach der Entfernung zum gesuchten Punkt sortiert. Den ersten untersuchten Tetraeder bilden die vier zum gesuchten Punkt nächsten Messpunkte. Ist dieser Tetraeder komplanar, kann dieser nicht zur Werteberechnung verwendet werden, da die Messpunktkombination keine Informationen über Punkte außerhalb der gemeinsamen Ebene enthält. Daher wird eine neue Messpunktkombination ermittelt, indem ein Punkt des Tetraeders durch den nächsten Punkt in der sortierten Liste ausgetauscht wird, und der Vorgang beginnt erneut (s. Abb. 10). Ist dies nicht der Fall, wird die aktuelle Punktkombination, sofern dies noch nicht geschehen ist, als Basis für eine eventuell nötige Extrapolation vorgemerkt. Dadurch wird bei der Extrapolation stets die Punktkombination mit dem geringsten Abstand zum gesuchten Punkt verwendet, was in den meisten Fällen realitätsnahe Werte liefert.
Ob sich ein Punkt in einem Tetraeder befindet, kann anhand der Normalen der Tetraederflächen bestimmt werden. Nachdem sichergestellt ist, dass alle Normalen in den Tetraeder hinein zeigen, kann für jede Fläche durch das Skalarprodukts von Normale und Vektor vom Eckpunkt einer Fläche zum gesuchten Punkt die Lage des Punktes zur Fläche (innen oder außen) bestimmt werden. Liegt der Punkt für alle Flächen innen, befindet sich der Punkt im Tetraeder und es kann mit der trilinearen Interpolation an diesem Tetraeder (s. 3.2.3) fortgefahren werden. Anderenfalls muss auch hier eine neue Messpunktkombination ermittelt werden. Wird keine Kombination von Messpunkten gefunden, deren aufgespannter Tetraeder den gesuchten Punkt enthält, muss das Verfahren der Interpolation mit dem zur Extrapolation vorgemerkten Tetraeder aus Messpunkten durchgeführt werden und der gesuchte Punkt des Teilobjekts wird in den Materialeigenschaften als extrapolierter Punkt markiert.
Um diesen Vorgang zu beschleunigen, kann eine Messpunktkombination, beispielsweise die für den letzten Punkt ermittelte, übergeben werden, die zuerst auf Beinhalten des gesuchten Punktes getestet wird. So kann z.B., wenn sich in der Nähe des zu verarbeitenden Punktes viele komplanare Punkte befinden, das Testen dieser übersprungen werden, falls der durch die übergebene Messwertkombination aufgespannte Tetraeder den gesuchten Punkt beinhaltet.

5.4.3 Berechnung der zweidimensionalen Temperaturverteilung

Ist für den Nutzer die Temperaturverteilung in einer bestimmten Ebene von Interesse, ist die Berechnung einer zweidimensionalen, höher aufgelösten Temperaturverteilung und deren Darstellung als Grafik für diese Ebene sinnvoll. Die Grundlage dafür bildet das in 5.4.2 beschriebene Verfahren. Jedoch wird die Temperatur nun für Punkte einer durch den Nutzer vorgegebenen Ebene statt für Punkte eines dreidimensionalen Objekts berechnet. Die Ebene wird durch ein

durch drei Punkte aufgespanntes Dreieck ($\triangle P_1P_2P_3$) definiert. P_1 wird nun als Mittelpunkt des zu berechnenden Ausschnitts angenommen. Die zu berechnenden Punkte sind in einem Raster um diesen Mittelpunkt angeordnet und entsprechen jeweils einem Pixel der resultierenden Grafik. Ein Punkt wird durch Vielfache der Vektoren $\overrightarrow{x} = \overrightarrow{P_1P_2}$ und $\overrightarrow{y} = \overrightarrow{P_1P_2} \times \overrightarrow{N}$ mit der Normalen der Dreiecksfläche als $\overrightarrow{N}$ definiert. Für die Größe des zu bearbeitenden Ausschnitts der Ebene sind die angestrebte Auflösung der Grafik (Breite w, Höhe h) und der gewählte Maßstab (d) ausschlaggebend. Dieser wird aufgrund der besseren Darstellbarkeit in Tausendstel der für die Geometrie verwendeten Einheit pro Pixel eingegeben (also meist $\frac{mm}{Pixel}$) und bestimmt den Abstand zweier Pixel als Anteil an den normierten Spannvektoren x und y. Die Position des Punktes $P_{m,n}$ für den Pixel der Grafik mit den Koordinaten $(m|n)$ im Raster wird also wie folgt berechnet:

$$\overrightarrow{OP_{m,n}} = \overrightarrow{OP_1} + \frac{\overrightarrow{x}}{|\overrightarrow{x}|}\frac{d \cdot (m - \frac{1}{2}w)}{1000} + \frac{\overrightarrow{y}}{|\overrightarrow{y}|}\frac{d \cdot (n - \frac{1}{2}h)}{1000} \tag{25}$$

Es sollen nur Werte für die Punkte dargestellt werden, die im Versuchsobjekt liegen. Dies ist mathematisch die Schnittmenge aus untersuchtem Objekt und darzustellender Ebene. Dazu wird mithilfe der Tetgen-Bibliothek eine möglichst wenige Tetraeder enthaltende Zerlegung der Teilobjekte berechnet. Wenn einer der so entstandenen Tetraeder den gesuchten Punkt enthält, befindet sich dieser innerhalb des Versuchsobjekts und wird wie in 5.4.2 beschrieben berechnet. Anderenfalls wird das entsprechende Pixel in der entstehenden Grafik vollständig transparent gefärbt. Für die Einfärbung der übrigen Punkte in der Grafik wird die in 5.4.4 beschriebene Methode verwendet.
Um die Berechnung der Temperaturverteilung zu beschleunigen, kann die zu berechnende Grafik vertikal in Bildbereiche aufgeteilt werden, deren Verarbeitung auf die vorhandenen Prozessorkerne verteilt und parallel durchgeführt wird. Die Rechenzeit für die einzelnen Bildbereiche kann jedoch stark variieren, da für Punkte außerhalb des Objekts die Interpolation übersprungen werden kann und anderenfalls verschiedene Suchzeiten für einen gültigen Tetraeder aus Messwerten für die Interpolation entstehen.

5.4.4 Visualisierung der Temperatur durch Farbe

Soll die Temperatur eines Objekts intuitiv erfasst werden können, bietet sich die farbliche Visualisierung an. Die Zuordnung eines Farbwertes zu einem Temperaturwert erfolgt über das HSV-Farbmodell, wobei Sättigung (S) und Dunkelstufe (V) mit 100% konstant bleiben und sich nur der Farbwert (H) abhängig von der Temperatur ändert. Dabei ist Rot ($H = 0°$) die Farbe für die heißesten, Blau ($H = 240°$) die Farbe für die kältesten Stellen (s. Abb. 11). Für den Farbwert ist die Angabe als Winkel auf einem Farbkreis im Bereich von 0° bis 360° üblich.

Abbildung 11: HSV-Farbverlauf von Blau zu Rot

Welche Temperaturen diesen Farbwerten entsprechen, kann durch den Nutzer festgelegt werden und die entsprechende Einstellung wird in den global verfügbaren Visualisierungsoptionen gespeichert. Temperaturen, die zwischen beiden Temperaturwerten liegen, werden als Zwischentöne zwischen Rot und Blau dargestellt, liegen sie jenseits dieser Randwerte, werden sie ebenfalls als Rot bzw. Blau dargestellt.

5.4.5 Finite-Elemente-Analyse

Zur Auswertung eines Versuchs ist oft von Interesse, wie viel Energie in das Versuchsobjekt bzw. dessen einzelne Materialien eingetragen wurde. Um die Wärmeenergie der Teilobjekte der Materialien (und damit des Gesamtobjektes) zu ermitteln, wird, wie in 3.1 beschrieben, die Finite-Elemente-Analyse verwendet. Die Analyse eines gesamten Objekts und die Datenstrukturen für die Speicherung der Ergebnisse sind in der Klasse *Analyzer* implementiert (s. Abb. 12).

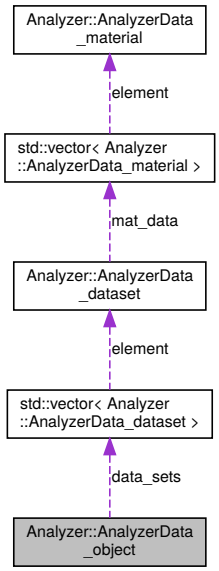


Abbildung 12: Datenstruktur zur Speicherung der Analyseergebnisse

Das Volumen des Gesamtobjekts bleibt dabei konstant, muss also nur einmal gespeichert werden (*AnalyzerData_object*). Mit dem Gesamtobjekt sind beliebig viele Sensordatensätze verknüpft,

die verschiedene Temperaturverteilungen beschreiben und somit jeweils verschiedene Ergebnisse liefern. Die Gesamtwärmeenergie und der Name des Datensatzes sind deshalb in *AnalyzerData_dataset* gespeichert. Einzelne Zeitpunkte von TSD-Datensätzen müssen als "zu analysieren" markiert werden. Diese gelten dann als einzelner Sensordatensatz. Um Aussagen über die einzelnen Materialien zu treffen, wird für jeden Sensordatensatz Volumen, Wärmeenergie sowie Name der Materialien in *AnalyzerData_material* gespeichert und mit dem Sensordatensatz verknüpft. Die so gewonnenen Daten sind anschließend auf der Programmoberfläche als Tabelle darstellbar (s. 5.4.7).

5.4.6 Oberfläche des Hauptfensters

Die Oberfläche des Hauptfensters sollte schnellen Zugriff auf alle Funktionen des Programms bieten, muss jedoch so strukturiert sein, dass die Funktionen logisch gegliedert und schnell aufzufinden sind. Sie ist deshalb in vier Bereiche gegliedert (s. Abb. 13):

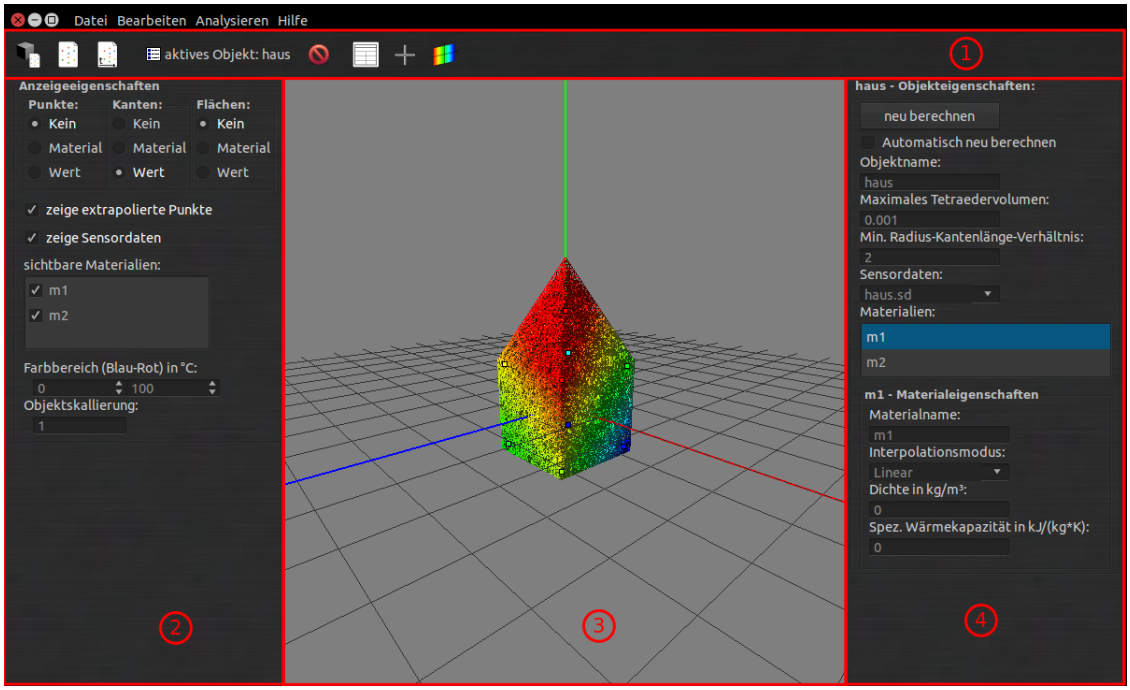


Abbildung 13: Die Bereiche des Hauptfensters

Die Toolbar (1 in Abb. 13) bietet zusätzlich zum Programmmenü schnellen Zugriff auf Funktionen zum Laden von Objekten und Sensordatensätzen sowie die Analysewerkzeuge. Des Weiteren ist in ihr das aktive Objekt auswählbar, auf dass sich alle Operationen in den anderen Bereichen

beziehen. Daher ist sie den anderen Fensterbereichen übergeordnet.
In der Mitte des Fensters wird in der 3D-Ansicht (3) das aktuelle Objekt dargestellt und ermöglicht interaktives Betrachten der farblich visualisierten Temperaturverteilung. Für die dreidimensionale Darstellung wird die OpenGL-Bibliothek verwendet, die durch das Speichern der Geometrie auf dem Grafikspeicher (in s.g. *display lists*) plattformübergreifend die Darstellung großer Anzahlen an Punkten, Linien oder Flächen ermöglicht.
Um die Visualisierung an die Bedürfnisse des Nutzers anzupassen, wird in den Anzeigeeigenschaften (2) die Art und Weise der Visualisierung festgelegt. Dies sind beispielsweise das Ein- und Ausblenden verschiedener Elemente und die Anpassung der Farbgebung mit sofortiger Darstellung der Änderungen im 3D-Fenster. Zum Vergleich der errechneten Temperaturverteilung mit den Messwerten können auch die Sensordaten als Punkte angezeigt werden.
Den Anzeigeeigenschaften sind die Objekteigenschaften (4) gegenübergestellt. Sie beziehen sich auf das Versuchsobjekt und ermöglichen die Anpassung der zur Auswertung relevanten Parameter[1]. Da eine Neuberechnung des Objekts nach einer Veränderung seiner Eigenschaften meist aufwändig ist, wird diese Prozedur in der Standardeinstellung nicht automatisch durchgeführt. Stattdessen erfolgt eine Warnung des Nutzers vor noch nicht angewendeten Änderungen.
Die Synchronisation der Daten auf der Programmoberfläche von Anzeigeeigenschaften und Objekteigenschaften mit dem Fachkonzept erfolgt jeweils getrennt, da sie sich aufgrund der funktionalen Trennung nicht beeinflussen. Sobald eine Visualisierungseigenschaft geändert wird, werden die aktuellen Werte der Anzeigeeigenschaften in das Fachkonzept übertragen, die Objekteigenschaften werden dadurch jedoch nicht verändert und daher nicht synchronisiert (und umgekehrt).

5.4.7 Analysedatenübersicht

Um die durch die Finite-Elemente-Analyse gewonnenen Daten nutzen zu können, ist eine übersichtliche Darstellung erforderlich. Dafür wurde eine Tabellenform gewählt, die ähnlich wie die in 5.4.5 beschriebene Datenstruktur aufgebaut ist und die gesamte und im jeweiligen Material vorhandene Wärmeenergie sowie die jeweiligen Volumina für alle geladenen Objekte beinhaltet (s. Abb. 14). So kann unmittelbar zwischen den geladenen Versuchsobjekten und deren verschiedenen Sensordatensätzen bzw. deren verschiedenen Zeitpunkten verglichen werden.

Wird eine Objekteigenschaft geändert und angewandt, muss das Objekt neu berechnet werden. Dabei wird automatisch auch die Analyse der in der Tabelle dargestellten Daten erneut durch-

[1] Eine Erklärung aller Objekteigenschaften und eine detailliertere Beschreibung der Oberfläche sind im der Software zugehörigen Handbuch beschrieben. Das Handbuch kann auch aus dem Programm heraus über den Menüpunkt *Hilfe* aufgerufen werden.

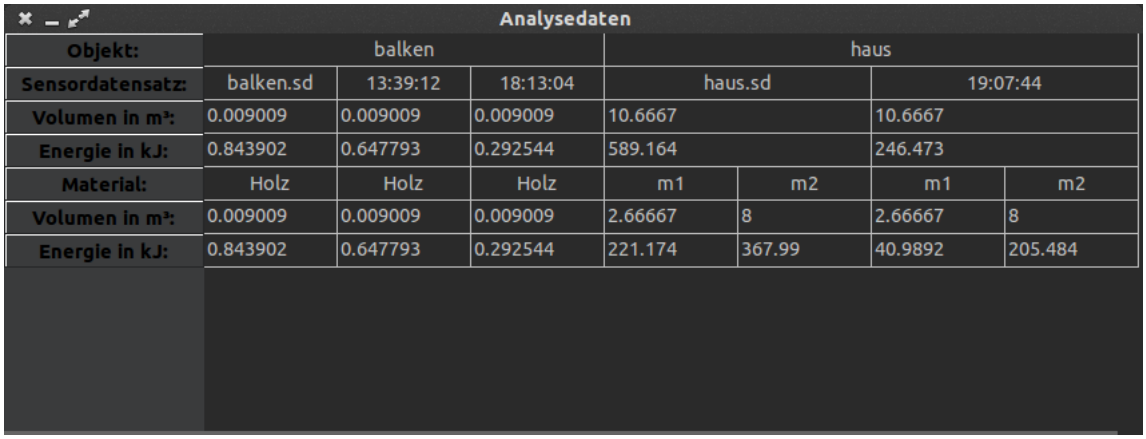


Abbildung 14: Oberfläche der Analysedatenübersicht

geführt, um die Aktualität der Analyseergebnisse sicher zu stellen. Dadurch kann der Nutzer sofort die Auswirkungen seiner Änderungen an den Versuchsparametern begutachten.

5.4.8 Oberfläche zur Berechnung einer 2D-Temperaturverteilung

Das Fenster zur Berechnung einer zweidimensionalen Temperaturverteilung gliedert sich in Einstellungen (oben) und Anzeigefenster (unten) (s. Abb. 15). In den Einstellungen können zum Einen die Parameter des in 5.4.3 beschriebenen Verfahrens angepasst werden, zum Anderen stehen Optionen zum Erstellen einer den Visualisierungseinstellungen entsprechenden Farbskala und Exportmöglichkeiten zur Verfügung.
Zur intuitiven Bedienbarkeit können der im Anzeigefenster angezeigte Bildausschnitt und die Größe sowie die Position der Farbskala interaktiv verändert werden. Eine Statusleiste ermöglicht das einfache Ablesen von Temperaturwerten an einer Position.
Da der Fortschritt bei der Berechnung der Temperaturverteilung unregelmäßig ist, wäre eine prozentuale Fortschrittsanzeige wenig aussagekräftig. Stattdessen wird das Anzeigefenster aktualisiert, so dass die bisher berechneten Pixel der Grafik bereits angezeigt werden und der Nutzer den Fortschritt des Prozesses bewerten kann.
Diese Aktualisierung während der Berechnung wirkt sich nur in sehr geringem Maße auf die Berechnungsdauer aus. Bei der Verwendung mehrerer Prozessorkerne fällt die Mehrbelastung auf nur einen der Kerne.
Die berechnete Temperaturverteilung kann als CSV-Datei oder PNG-Bild mit transparentem Hintergrund exportiert werden. Dies ermöglicht die Verwendung der Grafik zu Visualisierungszwecken, beispielsweise in Präsentationen, oder die Weiterverarbeitung der als CSV-Datei exportierten Daten in eine andere Software.

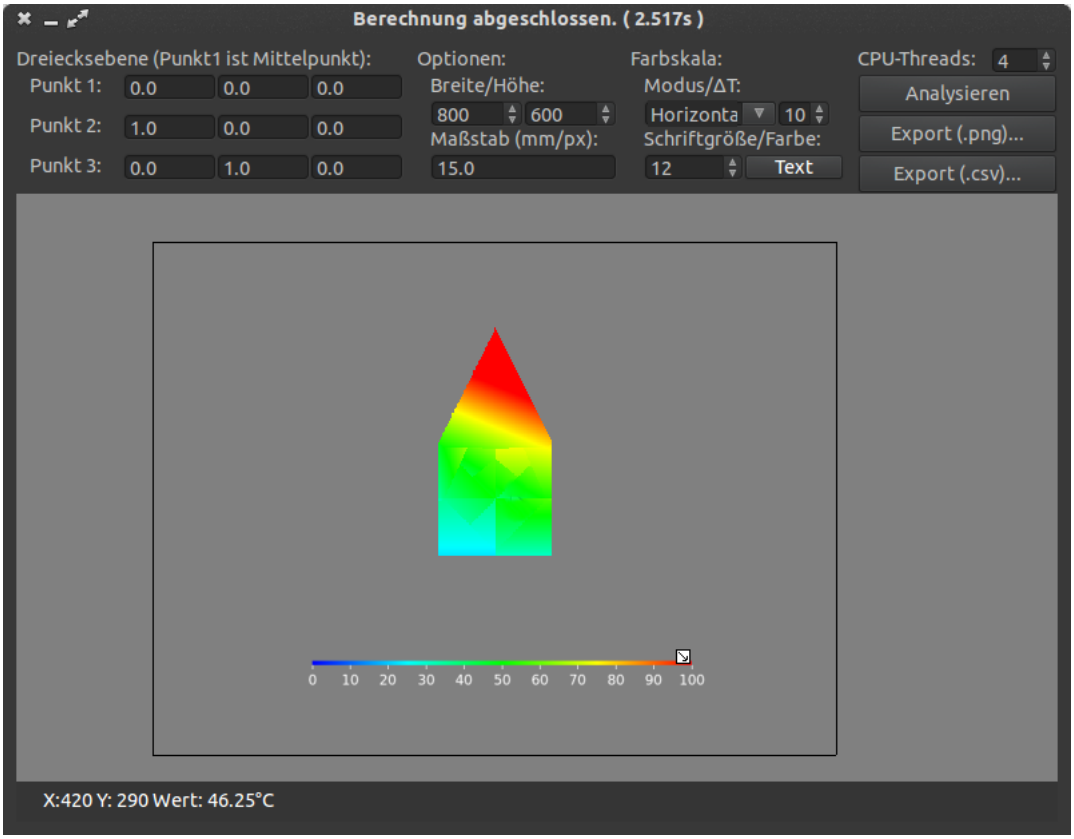


Abbildung 15: Oberfläche des Fensters zur Berechnung einer 2D-Temperaturverteilung

5.4.9 Exportmöglichkeiten

Neben den bereits in 5.4.8 beschriebenen Exportmöglichkeiten für die zweidimensionale Temperaturverteilung können auch Screenshots der 3D-Ansicht und die dreidimensionale Temperaturverteilung im VTK-Format des Visualisierungssystems *Visualization Toolkit*[2] für die weitere Verwendung gespeichert werden. Der Hintergrund der Screenshots wird dabei mittels des Tiefenpuffers des angezeigten Bildes vom gewünschten Bildinhalt unterschieden und durch Transparenz ersetzt, was für die Einbindung in Dokumente oder Präsentationen meist nötig ist.
Beim Export der VTK-Datei wird die ältere (*Legacy*), aber auch öfter unterstützte und einfacher strukturierte Version 3 im ASCII-Format verwendet. In diesem wird das dem Versuch zu Grunde liegende 3D-Modell in seiner aktuellen Zerlegung in Tetraeder gespeichert und zu jedem Punkt des Modells der Index des jeweiligen Materials und die Temperatur an diesem Punkt gespeichert. Es enthält also alle für eine Visualisierung wichtigen Daten über die Temperaturverteilung.

[2] s. `http://www.vtk.org/`

5.5 Erstellung des Handbuchs

Das Handbuch soll den Zweck, die Installation und die Verwendung der Software erläutern und dem Nutzer, auch bei eventuell auftretenden Fehlern, als Nachschlagewerk dienen. Dazu sind eine klare Beschreibung der Funktionalität und eine logische Gliederung notwendig, die ein schnelles Finden der benötigten Informationen ermöglicht. Das Handbuch ist daher in eine Einführung zur Klärung des Anwendungsgebietes und der gebotenen Funktionen, Erläuterungen zur Installation und Verwendung, eine schrittweise Anleitung zur Auswertung eines Versuchs, Hilfestellungen zur Fehlerbehebung und Informationen über die Lizenz der Programme mit ihren jeweiligen Unterpunkten gegliedert. Bei der Formulierung der Anweisungen wurde die aktive Form und eine direkte Adressierung des Lesers verwendet, um Missverständnisse zu reduzieren und das Nachvollziehen der Anweisungen zu vereinfachen.
Für die Erstellung des Handbuchs wurde das Textsatzsystem LaTeXverwendet, das durch die Formatierung mithilfe von Makros eine einheitliche Formatierung gleichartiger Textelemente sicherstellt und somit der Leser nicht durch inkonsequente Formatierungen irritiert wird. So können wiederkehrende Elemente wie Hinweise, Warnungen, URLs, Dateiinhalte oder Befehle und Knöpfe einer Programmoberfläche sofort voneinander unterschieden werden (s. Abb. 16).

Hinweis:
Ein Hinweis.

Warnung:
Eine Warnung.

`www.eine-web-adresse.de`

`befehl -arg1 -arg2`

[Button]

Abbildung 16: Wiederkehrende Formatierungen des Handbuchs

Des Weiteren ist der verwendete LaTeX-Compiler in der Lage, interaktive Referenzen innerhalb der resultierenden PDF-Dateien zu erstellen. Dadurch kann der Nutzer, wenn er das Handbuch über das Hilfe-Menü des Analysewerkzeugs aufruft, direkt vom Inhaltsverzeichnis zum gesuchten Abschnitt springen oder direkt einem Weblink folgen, was die Benutzung des Handbuchs weiter vereinfacht. Zur besseren Übersicht über eine Seite und aufgrund der einfacheren Handhabung in gedruckter Form wurde das A5-Format für die Seiten des Handbuchs gewählt.

5.6 Erstellung der Dokumentation

Für eine spätere Weiterentwicklung ist ein gut dokumentierter Quellcode erforderlich. Zusätzlich ist bei umfangreicherem Quelltext ein übersichtliches Nachschlagewerk erforderlich, da Informationen in Form von Quelltextkommentaren zu sehr über diesen verteilt sind. Zu diesem Zweck kam bei der Erstellung der HTML-Dokumentation das Programm *Doxygen*[3] zum Einsatz. Es ermöglicht die Generierung einer durchsuchbaren Dokumentation mit Klassen- und Methodenbeschreibungen, Abhängigkeiten dieser voneinander und zusätzlichen Informationen aus den im Quelltext zu diesem Zweck verfassten Kommentaren. Dadurch sind die Informationen sowohl in Quelltext als auch in Dokumentation enthalten und die Aktualität jener ist sichergestellt.

[3] s. `http://www.stack.nl/~dimitri/doxygen/`

6 Ergebnisse

6.1 Auswertung eines Beispielversuchs

6.1.1 Ziel des Versuchs

Zur beispielhaften Auswertung wurde ein Versuch innerhalb einer Versuchsreihe gewählt, in dem ein Holzbalken mit Hilfe von Radiowellen mit einer Frequenz von 13,56 MHz erwärmt wurde. Die Untersuchungen dienten dem Ziel, die verschiedenen Heizmethoden (Mikrowellenerwärmung, Radiowellenerwärmung, Infrarotbestrahlung) hinsichtlich der entstehenden Temperaturprofile und der Gleichmäßigkeit der Erwärmung sowie des energetischen Wirkungsgrades zu vergleichen. Das Programm ermöglichte es, die erhaltenen Messdaten zusammenzuführen, mithilfe der Finite-Elemente-Methode über das Versuchsobjekt zu interpolieren und außerhalb des Messbereiches zu extrapolieren sowie den Wirkungsgrad für einen bestimmten Zeitabschnitt zu ermitteln.

6.1.2 Vorüberlegungen

Der Versuchsaufbau im auszuwertenden Experiment entsprach in Grundzügen dem in 2.2 beschriebenen. Wie in Abb. 17 dargestellt hat der untersuchte Balken die Maße $296\,\mathrm{mm} \times 495\,\mathrm{mm} \times 120\,\mathrm{mm}$. Er wird im in Abb. 17 dargestellten kartesischen Koordinatensystem betrachtet.

Zur kontinuierlichen Temperaturmessung kamen 24 faseroptische Sensoren, 3 ODiSI-Messfasern und eine Wärmebildkamera zum Einsatz. Die Leistung des HF-Generators[1] während der Aufheizphase betrug 0,7 kW. Die Koordinaten der Messpunkte von Wärmebildkamera und faseroptischen Sensoren sowie die Lage der Messfasern sind in den Dateien des Beispielversuchs unter `Ausgangsdaten/Koordinaten` zu finden, die Daten dazu befinden sich unter `Ausgangsdaten/Daten`.

Der verwendete Holzbalken besteht aus Kiefernholz und besaß zu Beginn des Versuchs eine Feuchte von 12 % bezüglich seiner Trockenmasse. Er besitzt daher eine Dichte von ca. $520\,\mathrm{kg\,m^{-3}}$ und eine gewichtete spezifische Wärmekapazität von ca. $1{,}7\,\mathrm{kJ\,kg^{-1}\,K^{-1}}$. Die eingebrachte Energie verlässt im Verlaufe des Versuches immer mehr auch durch Wärmeübergang in die Umgebung oder Wasserverdampfung den bei der Temperaturmessung bilanzierten Raum. Deshalb wird zur

[1] Hochfrequenz-Generator

Abbildung 17: Maße des Versuchsobjekts und Lage im Koordinatensystem (UFZ, 2013)

Berechnung des Wirkungsgrades ein Bereich gesucht, in dem die Feuchtigkeit des Holzes und damit dessen spezifische Wärmekapazität und Dichte annähernd konstant sind. Dazu sollten Wärmeverluste sowie Wasserverdampfung noch eine geringe Rolle spielen (adiabatische Verhältnisse). Dies trifft am besten auf die ersten Minuten nach dem Start des Versuchs zu.
Der Wirkungsgrad η, der aus den oben genannten Gründen auch als adiabatischer Wirkungsgrad bezeichnet wird, ist der Quotient aus im Objekt enthaltener Wärmeenergie ΔQ_{obj} und in Form elektromagnetischer Strahlung im Radiowellenbereich zugeführter Energie E_{Radio}. Die zugeführte Energie ergibt sich aus der eingetragenen Strahlungsleistung P_{Radio} und der Zeit der Bestrahlung Δt. Die Wärmeenergie des Objekts wurde durch das Programm wie in 3.1 beschrieben über die spezifische Wärmekapazität, Masse und Temperaturdifferenz ermittelt. Der Wirkungsgrad für diesen Versuch kann also wie folgt berechnet werden:

$$\eta = \frac{\Delta Q_{obj}}{P_{Radio} \cdot \Delta t} \tag{26}$$

6.1.3 Durchführung der Auswertung

Damit die Daten ausgewertet werden können, müssen sie zunächst in das TSD-Format umgewandelt und zusammengeführt werden. Aus den in den Ausgangsdateien angegebenen Koordinaten lassen sich die Sensordefinitionsdateien für faseroptische Sensoren (`Sensordefinitionsdateien/faseropt_sensoren.sdef`), ODiSI-Fasern (`Sensordefinitionsdateien/odisi1-3.sdef`) und Wärmebildkamera (`Sensordefinitionsdateien/waermebild.sdef`) erstel-

len. Zur Vereinfachung wurden alle Längenangaben in Meter umgerechnet.

Mittels des Befehls *dos2unix <Datei>* werden sämtliche verwendeten Dateien mit UNIX-Zeilenenden versehen und ASCII-codiert, um Fehler bei der Umwandlung zu vermeiden. Die Daten der IR-Kamera stehen nicht an der durch die Indizes in der Konfigurationsdatei angegebenen Position in der Eingabedatei und der Zeitstempel ist in Minuten angegeben. Dies wurde mithilfe eines Tabellenkalkulationsprogrammes korrigiert und die neue Datei unter `Ausgangsdaten/Daten/IR-Kamera2.csv` gespeichert. Durch die Ausführung der Befehle

```
csvtosd -i ../Ausgangsdaten/Daten/Faseroptische\ Sensoren.csv -o
   faseropt_sensoren.tsd -s ../Sensordefinitionsdateien/
   faseropt_sensoren.sdef
csvtosd -i ../Ausgangsdaten/Daten/IR-Kamera2.csv -o waermebildkamera.
   tsd -s ../Sensordefinitionsdateien/waermebild.sdef
```

im Verzeichnis `TSD_Daten` werden nun die Messdaten von faseroptischen Sensoren und Wärmebildkamera in das TSD-Format umgewandelt. Dabei muss die Konfigurationsdatei dahingehend angepasst werden, dass die Faser nicht an der Z-Achse gespiegelt werden soll (s. Handbuch). Für die drei ODiSI-Fasern werden folgende Befehle verwendet (Die Höhe der Fasern ist der Koordinatendatei entnommen):

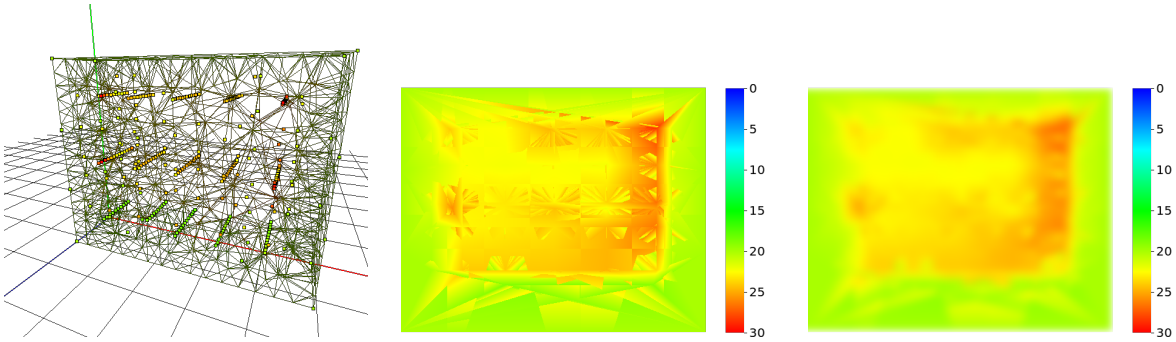
```
odisitosd -i ../Ausgangsdaten/Daten/ODISI-Faser\ 1_231\ \(oben\).csv
   -o odisi1.tsd -s ../Sensordefinitionsdateien/odisi1.sdef -log
   odisi1.log -height .328
odisitosd -i ../Ausgangsdaten/Daten/ODISI-Faser\ 2_223\ \(mitte\).csv
    -o odisi2.tsd -s ../Sensordefinitionsdateien/odisi2.sdef -log
   odisi2.log -height .2
odisitosd -i ../Ausgangsdaten/Daten/ODISI-Faser\ 3_429\ \(unten\).csv
    -o odisi3.tsd -s ../Sensordefinitionsdateien/odisi3.sdef -log
   odisi3.log -height .076
```

Nun müssen die gewonnenen Sensordatensätze, teilweise mit einer auf 5 s erhöhten Toleranz, mittels des Werkzeugs *mergetsd* zusammengeführt werden:

```
mergetsd -i1 odisi1.tsd -i2 odisi2.tsd -o odisi1_2.tsd
mergetsd -i1 odisi1_2.tsd -i2 odisi3.tsd -o odisi.tsd
mergetsd -i1 faseropt_sensoren.tsd -i2 waermebildkamera.tsd -o
   faseropt_kamera.tsd -max-dt 5
```

```
mergetsd -i1 faseropt_kamera.tsd -i2 odisi.tsd -o gesamt.tsd -max-dt
    5
```

Die Datei `TSD_Daten/gesamt.tsd` enthält nun die zusammengefassten Sensordatensätze für alle Zeitpunkte, zu denen von allen Messgeräten Daten erfasst wurden.
Das 3D-Modell des Versuchs wurde mithilfe des Programms *Blender*[2] erstellt und als OBJ-Datei exportiert (`beispielversuch.obj`). Da durch das Auswertungsprogramm mit einem Modell stets Sensordaten geladen werden, wurde eine Datei mit vier Sensorpunkten mit der Temperatur 0 °C zum ersten Laden erstellt (`beispielversuch.sd`). Diese Datei fungiert als Platzhalter, damit die zusammengeführten Sensordaten nicht aus dem Sensordatenverzeichnis verschoben und umbenannt werden müssen.
Mithilfe des Auswertungsprogramms können zwei- und dreidimensionale Temperaturverteilungen visualisiert und exportiert werden, wie in Abbildung 18 für den Zeitpunkt 14:40:13 Uhr dargestellt.



(a) dreidimensionale Temperaturverteilung

(b) Schnittebene (*simpleanalyzer-gui*)

(c) Schnittebene (extern geglättet)

Abbildung 18: Visualisierung durch das Programm

Durch das Programm wurden nun die ersten Minuten nach dem Start des Versuchs um 14:37:37 Uhr untersucht. Dabei ergaben sich folgende im Objekt enthaltenen thermische Energien:

Tabelle 3: Ermittelte Energien im Versuchsobjekt

Zeit:	14:37:37	14:38:13	14:39:13	14:40:13	14:41:13	14:45:13
Q_{obj} in kJ:	402,9	414,2	439,3	461,5	478,4	533,2

[2] s. `http://www.blender.org/`

6.1.4 Ermittlung des Wirkungsgrades

Der durchschnittliche Wirkungsgrad für eine bestimmte Zeitspanne ergibt sich nach oben genannter Formel. Beispielsweise beträgt dieser für eine Zeitspanne von 14:38:13 Uhr bis 14:39:13 Uhr somit:

$$\eta = \frac{439{,}25\,\mathrm{kJ} - 414{,}2\,\mathrm{kJ}}{0{,}7\,\mathrm{kW} \cdot 60\,\mathrm{s}} \tag{27}$$

$$\eta \approx 0,598 = \underline{\underline{59{,}8\,\%}} \tag{28}$$

Für die übrigen in der Tabelle enthaltenen Zeitspannen erfolgt die Berechnung analog:

Tabelle 4: Wirkungsgrade in den untersuchten Zeitabschnitten

Zeit	37:37-38:13	38:13-39:13	39:13-40:13	40:13-41:13	41:13-45:13
η	45,0 %	59,8 %	52,9 %	40,2 %	26,1 %

$\Longrightarrow$ Der maximale Wirkungsgrad beim Erwärmen des Holzbalkens mittels Radiowellen betrug unter den gegebenen Bedingungen ca. 60 %. Aufgrund der zunehmenden Wärmeverluste in die Umgebung bei fehlender thermischer Isolation sowie einsetzender Wasserverdampfung, die nicht bilanziert wurde, nimmt der Wirkungsgrad im Verlauf des Experimentes signifikant ab.

6.1.5 Fehlerbetrachtung

Sowohl die Messung als auch die Verarbeitung und Auswertung der Messwerte sind fehlerbehaftet und können das Ergebnis des Versuchs beeinflussen. Bei der Messung ist die Genauigkeit der verwendeten Messgeräte von Bedeutung. Doch auch die Annahme der Konstanz von Dichte und spezifischer Wärmekapazität sowie die erfassten Werte für die ins Objekt eingetragene Strahlungsleistung für einen bestimmten Bereich sind vereinfacht. Die Erfassung der für die Verarbeitung notwendigen Messpunktkoordinaten bzw. der den Faserverlauf beschreibenden Angaben ist ebenfalls fehlerbehaftet. Des Weiteren ist es kaum möglich, dass alle Datenquellen ihre Messwerte zur exakt gleichen Zeit aufnehmen, weshalb die verwendete Toleranz beim Zusammenfügen der Datensätze (3 s) eine weitere Fehlerquelle darstellt. Anschließend werden der Ermittlung der Temperaturverteilung durch das Programm nur eine begrenzte Auflösung von finiten Elementen und das einfache lineare Interpolationsverfahren verwendet. Bei der Berechnung des Wirkungsgrades wird die Änderung der Wärmeenergie über einen bestimmten, nicht unendlich kleinen Zeitabschnitt betrachtet, der resultierende Wirkungsgrad ist somit ein Durchschnittswert für den festgelegten Zeitraum.

6.2 Zusammenfassung

Im Rahmen dieser Besonderen Lernleistung wurde ein Paket von Programmen erstellt, mit dessen Hilfe Messdaten von Versuchen zur Materialerwärmung durch elektromagnetische Wellen aufbereitet und ausgewertet werden können. Diese können aus verschiedenen Quellen stammen. Die Software ermöglicht dabei eine teilweise Fehlerbereinigung, die Zusammenführung und Auswertung der Daten über eine grafische Benutzeroberfläche. Dadurch wird die Auswertung dieser Art von Versuchen beschleunigt und vereinfacht. Das Programmpaket kann überall dort eingesetzt werden, wo Temperaturverteilungen verschiedensten Ursprungs auszuwerten sind. Darüber hinaus ist die Übertragung auf örtliche und zeitliche Verteilungen beliebiger physikalischer Messgrößen aufgrund des offen gelegten Quelltextes möglich. Die Auswertung umfasst die Berechnung von zwei- und dreidimensionalen Temperaturverteilungen für Versuchsobjekte mit beliebig vielen Materialien. Einfache oder zeitbezogene Messwertsätze können bezüglich der im Objekt enthaltenen thermischen Energie ausgewertet oder die Temperatur für einen beliebigen Punkt im Objekt bestimmt werden. Zur besseren Erfassung der dreidimensionalen Temperaturverteilung wird diese auf vom Benutzer anpassbare Art und Weise als interaktives 3D-Modell visualisiert. Wenn nötig, können die dreidimensionale Temperaturverteilung als Visualization-Toolkit(VTK)-Datei oder die Temperaturverteilung über eine Ebene als CSV-Datei zur Weiternutzung exportiert werden. Auch die Visualisierung lässt sich in Form von Screenshots exportieren.
Um die Verwendung der Software für den hier behandelten Versuchstyp zu erläutern und die Anwendung außerhalb des Rahmens dieser BeLL möglich zu machen, wurde ein Handbuch zum entwickelten Programmpaket verfasst. Die Verwendung der GNU Affero GPL-Lizenz ermöglicht die spätere Weiterentwicklung der Programme, z.B. die Erstellung eines neuen Konverters für ein neues Messgerät. Dafür ist die erstellte Dokumentation des Quelltextes von Nutzen.
Die entstandenen Programme werden bereits erfolgreich im Rahmen eines Forschungsprojektes am Helmholtz-Zentrum für Umweltforschung - UFZ in Leipzig eingesetzt. Auch in einem Praktikum zweier Studenten von der Universität Leipzig im Studiengang Physik wurde die Software bereits verwendet (s. Anlage 6). Auf Basis dieser Praxiserfahrung konnten das Handbuch und die Bedienbarkeit der Konverterprogramme verbessert werden.

7 Ausblick

7.1 Portierung auf andere Plattformen

Bisher werden Debian Linux und seine Derivate als einzige Plattform unterstützt. Um das potentielle Einsatzgebiet der Software zu vergrößern, könnte eine Portierung der Programme auf andere Betriebssysteme wie beispielsweise Microsoft Windows erfolgen. Durch die Verwendung plattformunabhängiger Bibliotheken und die Vermeidung betriebssystemspezifischen Quelltextes ist diese ohne den Austausch größerer Programmteile möglich und schon teilweise geschehen. Die aktuelle Version des Quellcodes kann mit einer aktuellen Version des GNU-C++-Compilers (mindestens 4.8) und den entsprechenden Bibliotheken auf Windows compiliert werden und ist grundlegend funktional. Jedoch ist das Analyseprogramm (*simpleanalyzer-gui*) durch verschiedene Compilerversionen und Unterschiede in der Windows-Version der GUI-Bibliothek teilweise instabil und die Oberfläche weist zum Teil größere Fehler auf, welche die Bedienbarkeit einschränken. Mit entsprechender Entwicklungszeit könnte der Quelltext so erweitert werden, dass beim Compilieren des Quelltextes auf die Eigenheiten des verwendeten Betriebssystems eingegangen wird. Die dafür nötige Zeit steht jedoch bisher in keinem günstigen Verhältnis zum zu erwartenden Nutzen, da die Linux-Version bereits erfolgreich eingesetzt wird.

7.2 Parallelisierung der Berechnung der dreidimensionalen Temperaturverteilung

Vor allem die Berechnung der Temperaturverteilung für höher aufgelöste Objekte könnte durch die parallele Ausführung der Interpolation für mehrere Punkte unter Ausnutzung der Rechenleistung mehrerer Prozessorkerne beschleunigt werden, wie dies in einfacher Form bereits bei der Berechnung der zweidimensionalen Temperaturverteilung geschieht. Dazu ist jedoch ein erheblicher Aufwand nötig, da während der Suche nach geeigneten Punkten für die Interpolation die Liste der Messpunkte umgeordnet wird und sich mehrere derartige Prozesse somit bei der Parallelausführung stören würden. Der Prozess müsste also so umstrukturiert werden, dass die Messdaten entweder nicht verändert oder redundant gespeichert würden.
Die Parallelisierung des Prozesses könnte auch über eine entsprechende Bibliothek wie bei-

spielsweise OpenCL umgesetzt werden, was zusätzlich die Nutzung der Rechenkapazität einer kompatiblen Grafikkarte ermöglicht. Das Programm würde so die Kapazitäten leistungsfähiger Grafikkarten ausnutzen können, dabei jedoch auch allein auf der CPU lauffähig bleiben. Aus Zeitgründen und der bisher nicht vordergründig vorhandenen Notwendigkeit der Steigerung der Verarbeitungsgeschwindigkeit wurde dies im Rahmen der Arbeit nicht umgesetzt.

7.3 Kombination mit anderer Software

Damit die Funktionalität der Software besser für weitere Forschungsprojekte genutzt werden kann, könnten Schnittstellen mit anderen Spezialprogrammen, beispielsweise zur Modellierung elektromagnetischer Felder oder zur Modellierung von Stofftransportprozessen, geschaffen werden. Dies wäre z.B. durch gemeinsame, eventuell neue Dateiformate, aber auch durch direkte Verknüpfung der Funktionalität über eine Erweiterung des Quelltextes möglich. Durch den VTK-Export wurde dies schon ansatzweise umgesetzt, doch aufgrund der begrenzten Zeit nicht erweitert. Für neue Forschungsprojekte oder Arbeitsabläufe kann die Software, aufgrund der Open-Source-Lizenz auch außerhalb des UFZ, zukünftig angepasst werden.

Literaturverzeichnis

[Hoyer u. a. 2014] Hoyer, Christian ; Pfütze, Christian ; Plarre, Rudy ; Trommler, Ulf ; Steinbach, Steffen ; Klutzny, Kerstin ; Holzer, Frank ; Rabe, Carsten ; Höhlig, Björn ; Kopinke, Frank-Dieter ; Schmidt, Detlef ; Roland, Ulf: Chemikalienfreie Bekämpfung von Holzschädlingen durch dielektrische Erwärmung mit Radiowellen und Mikrowellen. In: *Chemie Ingenieur Technik* 86 (2014), Nr. 8, 1187–1197. http://dx.doi.org/10.1002/cite.201300091. – DOI 10.1002/cite.201300091. – ISSN 1522–2640

[Marmelad (CC BY-SA 3.0) 2008] Marmelad (CC BY-SA 3.0): *interpolation2.svg*. http://upload.wikimedia.org/wikipedia/commons/9/97/3D_interpolation2.svg, 2008. – (25.4.14,18:30)

[Metaxas u. Meredith 1983] Metaxas, A.C. ; Meredith, R.J.: *Industrial Microwave Heating*. P. Peregrinus, 1983 (IEE power engineering series). – ISBN 9780906048894

[Rieg u. a. 2012] Rieg, F. ; Hackenschmidt, R. ; Alber-Laukant, B.: *Finite Elemente Analyse für Ingenieure*. Carl Hanser Verlag GmbH & Company KG, 2012. – ISBN 9783446434691

[Roland u. a. 2011] Roland, Ulf ; Holzer, Frank ; Trommler, Ulf ; Pfütze, Christian ; Kopinke, Frank-Dieter ; Pfütze, Christian ; Freytag, Olaf: Radiowellenunterstützte thermische Behandlung als neue Technologie zur Trocknung und Dekontamination von Bauteilen. In: *Chemie Ingenieur Technik* 83 (2011), Nr. 3, 254–261. http://dx.doi.org/10.1002/cite.201000141. – DOI 10.1002/cite.201000141. – ISSN 1522–2640

[Si 2013] Si, Hang: *TetGen*. http://wias-berlin.de/software/tetgen/, 2013. – (28.4.14,17:48)

[UFZ 2013] UFZ, Helmholtz-Zentrum für Umweltforschung: *Bild*. 2013/2014

[Zeidler u. a. 2013] Zeidler, E. ; Bronštejn, I.N. ; Semendjaev, K.A. ; Grosche, G. ; Ziegler, V. ; Ziegler, D.: *Springer-Handbuch der Mathematik I: Begründet von I.N. Bronstein und K.A. Semendjaew Weitergeführt von G. Grosche, V. Ziegler und D. Ziegler Herausgegeben von E. Zeidler*. Springer Fachmedien Wiesbaden, 2013 (Springer-Handbuch der Mathematik). – ISBN 9783658002855

Abbildungsverzeichnis

Tabellenverzeichnis

Anlagen

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst und keine anderen Hilfsmittel als angegeben verwendet habe. Insbesondere versichere ich, dass ich alle wörtlichen und sinngemäßen Übernahmen aus anderen Werken als solche kenntlich gemacht habe. Weiterhin erkläre ich, dass diese Arbeit schulintern Verwendung finden kann.

Leipzig, den 19.12.14