

ENTWICKLUNG EINER INTERFACE-SOFTWARE
UND VISUALISIERUNG IN PYTHON FÜR EINE
ATMOSPÄREN-STRAHLUNGSTRANSFERMODELL-
SOFTWARE FÜR FERNERKUNDUNGSANALYSEN

BACHELORARBEIT IM FACH MECHATRONIK
AN DER FAKULTÄT FÜR INGENIEURWISSENSCHAFTEN

AN DER HOCHSCHULE ROSENHEIM,
UNIVERSITY OF APPLIED SCIENCES



ERSTPRÜFER
PROF. DR.-ING. PETER ZENTGRAF, M.SC.

ZWEITPRÜFER
PROF. DR. ELMAR JUNKER

VORGELEGT VON
MICHAEL JOCKISCH

ROSENHEIM, 3. JULI 2012

Michael Jockisch: *Entwicklung einer Interface-Software und Visualisierung in Python für eine Atmosphären-Strahlungstransfermodell-Software für Fernerkundungsanalysen* , © 3. Juli 2012

ERSTPRÜFER:
Prof. Zentgraf

ZWEITPRÜFER:
Prof. Junker

BETREUER IM BETRIEB:
Werner Hupfer

ORT:
83115 Neubeuern, Deutschland

ABGABETERMIN:
3. Juli 2012

ZUSAMMENFASSUNG

Künftige satellitenbasierte Fernerkundungsmissionen zur Erforschung der Erde stützen sich auf hoch entwickelte optische Instrumente. Die komplexe Natur der zu gewinnenden Daten erfordert bereits Jahre vor dem eigentlichen Start eine realistische Simulation der am Satelliten detektierten Strahlung. Die daraus resultierenden Erkenntnisse sind essentiell für das Design der optischen Instrumente. Für die Simulation des Strahlenganges von der Sonne, durch die Atmosphäre, der Reflektion am Boden hin zum Erdbeobachtungssatelliten wird auf Strahlungstransfermodelle zurückgegriffen.

Die vorliegende Arbeit sucht eine kostenfreie Alternative zu dem weit verbreiteten kommerziellen Strahlungstransfermodell *MODTRAN₄*. Dabei fiel die Wahl auf das unter der *GNU General Public License* stehende *libRadtran*. In einem Vergleich beider Modelle, wird die Substituierbarkeit des kostenpflichtigen *MODTRAN₄* mit dem frei zugänglichen *libRadtran* nachgewiesen. Darüber hinaus stellt die Arbeit den Anspruch, als ergänzende Dokumentation für *libRadtran* zu fungieren. Um den Zugang zur Materie zu erleichtern, erklärt sie die zum Verständnis eines Strahlungstransfermodells benötigten physikalischen Zusammenhänge.

Während einem Strahlungstransfer durch die Atmosphäre treten verschiedenartigste Begleiterscheinungen auf. Dabei hat vor allem der *Adjacency-Effekt* einen schwerwiegenden Einfluss auf die am Sensor detektierte Strahlung. Dieser besagt, dass bei der satellitenbasierte Fernerkundung nicht nur das zu beobachtende Ziel die detektierte Strahlung beeinflusst, sondern auch jene Strahlung, welche durch die Umgebung des Ziels in den Strahlengang zum Sensor reflektiert wird. Der *Adjacency-Effekt* wird im Rahmen der vorliegenden Arbeit mittels einem Python-Skript in *libRadtran* implementiert. In einem abschließenden Test wird dessen Funktionalität erfolgreich sichergestellt.

Zudem soll gezeigt werden, wie die Auflösung des Strahlungstransfer-Modells *libRadtran* durch die Implementierung einer *line-by-line*-Simulation signifikant erhöht werden kann. Dazu wird über das Python-Skript *Py4CATS* auf Spektralparameter der molekularen Absorptions-Datenbank *HITRAN* zugegriffen, wodurch eine Simulation sämtlicher Spektrallinien ermöglicht wird. Dieser zusätzliche Funktionsumfang sorgt für die endgültige Überlegenheit *libRadtrans* gegenüber *MODTRAN₄*.

Never worry about numbers.
Help one person at a time,
and always start with the
person closest you.

— Mother Teresa

DANKSAGUNG

Es gibt eine Reihe von Leuten, ohne denen diese Thesis sicherlich nicht in dieser Form zustande gekommen wäre.

Allem voran gilt mein größter Dank Werner Hupfer, der mir seit mehr als einem halben Jahrzehnt die Welt der Ingenieure eröffnet. Ganz gleich, um welches Problem es sich handelte, Werner war nie mehr als eine Schreibtischlänge und einen Mate-Tee weit entfernt, um mir mit seinem Rat beizustehen.

Ein großes Dankeschön geht auch an Dr. Xavier Gnata, der mir mit seinem großen Fachwissen und Anekdoten aus der IT stets auf die Sprünge half.

Mein Dank gilt auch Stefan Pauss, der mich immer sofort über die wichtigsten Neuigkeiten mit Python unterrichtete. In diesem Zusammenhang möchte ich auch Juan Irizar danken, der mir des Öfteren die entscheidenden Denkanstöße zur Softwareumsetzung gab.

Ebenso bin ich meinem persönlichen Chauffeur Erich Weichs zu Dank verpflichtet, ohne dem mich das Leben als Pendler in den finanziellen und moralischen Ruin getrieben hätte.

Ein großer Dank gilt auch meinen Prüfern Prof. Peter Zentgraf und Prof. Elmar Junker, welche mir die essentiellsten Ratschläge zum Erstellen wissenschaftlicher Schriften mit auf den Weg gaben.

INHALTSVERZEICHNIS

I	PROBLEMSTELLUNG UND HINFÜHRUNG ZUM THEMA	1
1	EINLEITUNG	2
1.1	Motivation und Begriffsklärung	2
1.2	Aufgabenstellung	3
2	GRUNDLAGEN EINES STRAHLUNGSTRANSFERMODELLS	5
2.1	Physikalische Grundlagen	5
2.2	Gesuchte physikalische Ausgabegrößen	9
2.2.1	Radiance	11
2.2.2	Irradiance	12
2.2.3	Transmission	12
2.3	Absorptionsbestimmung mit molekularen Datenbanken .	13
2.4	Zentrale Strahlungstransfergleichung	15
2.5	Berechnung der Ausgabegrößen über die optische Dicke .	18
3	STRAHLUNGSTRANSFERMODELLE	21
3.1	libRadtran (frei verfügbar)	21
3.1.1	Strahlungstransfersimulation mit <i>uvspec</i>	21
3.1.2	Installation und Ausführung von <i>uvspec</i>	22
3.1.3	Erstellung der <i>uvspec</i> Input-Datei	25
3.1.4	Struktur der <i>uvspec</i> Output-Datei	28
3.1.5	Interpretation der <i>uvspec</i> Output-Datei	31
3.1.6	Beispiel: bewölkter Himmel	34
3.1.7	Beispiel: Smog	37
3.1.8	Beispiel: Wellenlängenabhängigkeit	40
3.2	MODTRAN (kommerziell)	44
3.2.1	MODTRAN mit der Benutzeroberfläche <i>MODO</i> . .	44
3.2.2	Beispiel: Berechnung der Radiance TOA	45
4	VERWENDETE ENTWICKLUNGS-SOFTWARE	48
4.1	Programmiersprache <i>Python</i>	48
4.1.1	Mathematische Bibliothek <i>NumPy</i>	49
4.1.2	Wissenschaftliche Bibliothek <i>SciPy</i>	52
4.1.3	Bibliothek zur Visualisierung <i>matplotlib</i>	52
4.2	Texteditor <i>Kate</i>	53
4.3	Unix-Tool <i>awk</i>	54
4.4	Python-Skript zur Visualisierung	56
4.5	Molekulare Absorptionsdatenbank <i>HITRAN</i>	58
4.6	Bearbeitung von <i>HITRAN</i> mit <i>Py4CATS</i>	59
4.6.1	Extrahieren der Spektralparameter mit <i>extract</i> . . .	60
4.6.2	Berechnung der optischen Dicke mit <i>lbl2od</i>	62

4.6.3	Anpassung der <i>molecular_tau_file</i> für <i>uvspec</i> . . .	62
4.7	Nachbearbeitung mit Vektorgrafik-Editor <i>Inkscape</i>	63
II	UMSETZUNG DER AUFGABENSTELLUNG	65
5	VERGLEICH VON LIBRADTRAN UND MODTRAN	66
5.1	Schreiben der <i>uvspec</i> Input-Datei	66
5.2	Versuchsaufbau des Vergleichs	67
5.3	Auswertung, Interpretation und Ergebnis	69
6	IMPLEMENTIERUNG DES ADJACENCY EFFEKTS	71
6.1	Mathematische Grundlagen	72
6.2	Umsetzung in Python	77
6.2.1	Albedo-Karten	77
6.2.2	Schnelle Fourier-Transformation	78
6.2.3	Look-Up-Table	79
6.2.4	Struktur des Skripts	81
6.3	Funktionstest des Skripts	83
6.4	Verifizierung der verwendeten Gleichungen	85
7	ZUGRIFF AUF MOLEKULAR-DATENBANK HITRAN	88
7.1	Beschreibung der einzelnen Arbeitsschritte	88
7.1.1	Interpolation des Sonnenspektrums	89
7.1.2	Anpassung der <i>uvspec</i> Input-Datei	92
7.2	Vergleich <i>line-by-line</i> -Simulation mit <i>Bändermodell</i>	93
8	AUSBLICK	95
III	APPENDIX	96
9	ANHANG	97
9.1	Installationsproblem bei <i>libRadtran</i>	97
9.2	Dopplerverbreiterung	98
	LITERATURVERZEICHNIS	99

ABKÜRZUNGEN

API	A pplication P rogramming I nterface
CSV	C omma- S eperated V alues
DFT	D iscret F ourier T ransformation
DISORT	D iscret O rdinate R adiative T ransfer
DLR	D eutschen Z entrum für L uft- und R aumfahrt
ESA	E uropean S pace A gency
FOV	F ield O f V iew
FFT	F ast F ourier T ransformation
GUI	G raphical U ser I nterface
GSD	G round S ampling D istance
HITRAN	H igh R esolution T ransmission
IDE	I ntegrated D evelopment E nvironment
IR	I nfrarot
KDE	K Desktop E nvironment
LUT	L ook U p T able
MODTRAN	M oderate T ransition C ode
NASA	N ational A eronautics and S pace A dministration
NIR	N ahes I nfrarot
RTE	R adiative T ransfer E quation
TOA	T op O f A tmosphere
TOC	T op O f C anopy
USAF	U.S. A ir F orce

Teil I

PROBLEMSTELLUNG UND HINFÜHRUNG
ZUM THEMA

EINLEITUNG

1.1 MOTIVATION UND BEGRIFFSKLÄRUNG

Seit Anbeginn seiner Zeit lebt der Mensch in fortwährender Koexistenz mit seiner Umwelt. Das Wissen oder Unwissen um seine Welt entscheidet jeher über nicht weniger als seine Schicksal. Deswegen nutzt er heute mehr denn je seine technologischen Errungenschaften, um die naturgegebenen und anthropogenen Veränderungen seiner Umwelt zu beobachten und zu verstehen. Dabei stellt die Fernerkundung mittels Satelliten eines seiner wirksamsten Werkzeuge dar.

Eine *Fernerkundungsanalyse* bedeutet in diesem Zusammenhang das Auswerten der durch Satelliten gewonnenen Informationen. Die Ergebnisse der vorliegenden Arbeit sollen dabei helfen, die Erreichbarkeit des gewünschten Informationsgewinns durch künftige Satelliten zu analysieren.

Zivile Einsatzgebiete von Erdbeobachtungssatelliten erstrecken sich über die Meteorologie, Umweltbeobachtung, Geologie oder Kartierung. Abhängig von dem zu beobachtenden Szenario kommen verschiedene Instrumente zum Einsatz. Diese unterscheiden sich primär darin, welchen Bereich des elektromagnetischen Spektrums sie detektieren können. Jedoch ist der spektrale Transmissionsgrad der Atmosphäre in starkem Maße von der Wellenlänge abhängig. Aus diesem Grund bedarf es im Vorfeld ein *Atmosphären-Strahlungstransfermodell*, um die Absorption der einzelnen Wellenlängen zu berechnen. Die durch dieses Modell gewonnenen Erkenntnisse sind fundamental für die Wahl des Instruments und das optische Design des künftigen Satelliten.

Atmosphären-Strahlungstransfermodelle sind heute als *Software* verfügbar und können an jedem aktuellem Computer in annehmbarer Zeit berechnet werden. Sollen jedoch zusätzliche Effekte berücksichtigt werden, welche durch gängige Strahlungstransfermodelle noch nicht abgedeckt werden, so müssen diese nachträglich implementiert werden. Dazu bedarf es einer *Interface Software*, welche die Schnittstelle zwischen den Programmkomponenten, also zwischen dem bestehenden Modell und einer Erweiterung, darstellt. Die Entwicklung eines Interfaces ist ein wesentlicher Bestandteil dieser Arbeit. Dabei spielt das Schreiben der Software selbst allerdings eine untergeordnetere Rolle. Voraussetzung für die Softwareentwicklung ist das Verständnis des bereits vorliegen-

den Modells, wozu davor wiederum die atmosphärischen Einflüsse zur Gänze verstanden werden müssen.

Damit die Ergebnisse dem Nutzer intuitiver vermittelt werden können, bietet es sich an die Resultate zu *visualisieren*. Dabei werden sowohl die Interface Software, als auch die Visualisierungs-Skripte, in der Programmiersprache *Python* geschrieben.

1.2 AUFGABENSTELLUNG

Diese Arbeit verfolgt drei Ziele, welche in Teil ii ab Seite 65 umgesetzt werden. Zuvor werden jedoch die zum Verständnis der Umsetzung notwendigen Grundkenntnisse in Teil i beschrieben. Die drei Ziele dieser Arbeit lauten:

1. Vergleich von *libRadtran* mit *MODTRAN 4*
2. Implementierung des *Adjacency-Effekts* in *libRadtran*
3. Zugriff auf die molekulare Absorptions-Datenbank *HITRAN* mit *libRadtran*

Dem ersten Ziel, dem Vergleich zweier Strahlungstransfermodelle geht die Wahl eines frei zugänglichen Modells vor, welches das in Verwendung befindliche *MODTRAN 4* ersetzen soll. Nach Berücksichtigung der in Kapitel 3 beschriebenen Anforderungen, fällt die Wahl auf *libRadtran*. Zuvor erfolgt in Abschnitt 2.1 eine Betrachtung der physikalischen und mathematischen Grundlagen, welche den verwendeten Modellen zugrunde liegen. Mit diesem Rüstzeug wird näher auf das alternative Modell *libRadtran* und der zum Einsatz kommenden Software eingegangen, ehe in Kapitel 5 das primäre Ziel umgesetzt wird und ein finaler Vergleich zwischen den Modellen erfolgt. Dazu wird ein identisches Beobachtungsszenario separat mit beiden Modellen berechnet und deren Ergebnisse auf ihre Gemeinsamkeiten und Unterschiede hin untersucht. Die Intention lautet, eine Unabhängigkeit von *MODTRAN 4* zu erlangen, indem man auf ein kostenfreies und im Idealfall ebenso genaues Strahlungstransfermodell ausweicht.

Das zweite in Kapitel 6 umgesetzte Ziel ist die Implementierung des *Adjacency-Effekts* in das Strahlungstransfermodell *libRadtran*. Dieser Effekt hat einen signifikanten Einfluss auf die vom Erdbeobachtungssatelliten detektierte Strahlung, weswegen dessen Simulation unabdingbar ist. Nachdem lediglich die neueste und der Abteilung nicht vorliegende Version 5 von *MODTRAN* diesen Effekt zu simulieren vermag, stellt dessen Implementierung einen großen Mehrwert dar.

Das dritte Ziel lautet, die molekulare Absorptions-Datenbank *HITRAN* derart aufzubereiten, dass *libRadtran* auf die darin enthaltenen Spektraldaten zugreifen kann (siehe Kapitel 7). Während *libRadtran* als auch *MODTRAN 4* nativ lediglich über gemittelte Bändermodelle verfügen, erlaubt *HITRAN* die Betrachtung sämtlicher in einem Band auftretenden Spektrallinien. Diese zusätzlichen Informationen bringen eine signifikant höhere Auflösung der gesuchten Daten mit sich und steigern den Nutzwert von *libRadtran* damit enorm. Folglich bedeutet die Verwendbarkeit der *HITRAN*-Datenbank durch *libRadtran* nicht nur einen kommerziellen Vorteil, sondern auch einen wissenschaftlichen Fortschritt.

GRUNDLAGEN EINES STRAHLUNGSTRANSFERMODELLS

Die Intention eines Atmosphärentransfermodells ist es, den kompletten Strahlengang von der Sonne, durch die Atmosphäre und die Reflexion an der Erdoberfläche hin zum Satelliten zu simulieren. Dabei sollen möglichst viele der dabei auftretenden Effekte berücksichtigt werden. In Abschnitt 2.1 werden zunächst die physikalischen Grundlagen bezüglich der in der Atmosphäre auftretenden Effekte erklärt, ehe in Abschnitt 2.2 eine Beschreibung der gesuchten Ausgabegrößen erfolgt. Daraufhin wird in den Abschnitten 2.3 bis 2.5 die Strahlungstransfergleichung hergeleitet, welche die Grundlage der Strahlungstransfermodelle darstellt. Erst dann werden in Kapitel 3 die im Rahmen dieser Arbeit verwendeten Strahlungstransfermodelle und deren Gebrauch umfassend beschrieben.

2.1 PHYSIKALISCHE GRUNDLAGEN

Ein *Strahlungstransfermodell* ist ein mathematisches Modell, mit welchem Absorptions-, Transmissions- und Reflexionsvorgänge in der Atmosphäre simuliert werden können.

Auch wenn verschiedenartige Atmosphärentransfermodelle existieren, ist die zugrunde liegende physikalische Datenbank in den meisten Fällen identisch. Wesentlich für die Berechnung des Strahlengangs durch die Atmosphäre ist die molekulare Lichtabsorption, wobei verschiedene Frequenzbereiche des Lichts unterschiedlich stark absorbiert werden. Wie man Abbildung 1 entnehmen kann, ist diese Absorption stark wellenlängenabhängig [Alberz, 2001]. Aus diesem Grund kommen bei der Fernerkundung auch nur diejenigen Wellenlängenbereiche in Betracht, für welche die Atmosphäre weitgehend durchlässig ist. Die Ursachen für den wellenlängenabhängigen Transmissionsgrad sollen im Folgenden näher untersucht werden.

Die von der Sonne abgestrahlte extraterrestrische Sonnenstrahlung E_0 erreicht zunächst die oberste Schicht der Erdatmosphäre. Die Grenze zwischen Atmosphäre und Weltall wird in der Fernerkundung als *Top-of-Atmosphere (TOA)* bezeichnet. Auf dem Weg von der Sonne bis zum Erreichen der TOA sei die extraterrestrische Sonnenstrahlung eine Konstante, da Wechselwirkungen mit der verschwindend geringen Materie im Weltall vernachlässigt werden. Ein ebenfalls vernachlässigter Teil

von E_0 wird in den oberen Schichten der Atmosphäre zurück in das Weltall reflektiert. Der signifikante Anteil jedoch, unterliegt auf seinem Weg zur Erdoberfläche der Refraktion, Absorption und Streuung. Ebenso auf dem Weg von der Erdoberfläche hin zum Satelliten.

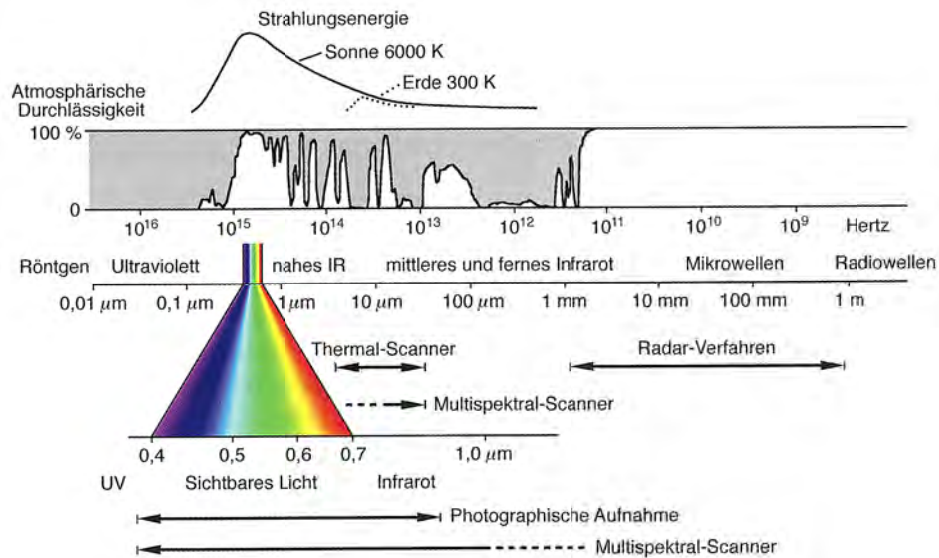


Abbildung 1: Strahlungsenergie der Sonne und atmosphärische Durchlässigkeit über den Wellenlängenbereich des elektromagnetischen Spektrums

Refraktion ist die Brechung von Lichtstrahlen beim Durchgang durch die Grenzflächen einzelner Atmosphärenschichten. In diesem speziellen Zusammenhang wird die Refraktion auch als *atmosphärische Strahlenbrechung* bezeichnet. Für die Brechung der Lichtstrahlen an der Grenzfläche gilt

$$\sin \alpha = \text{const} \cdot \sin \beta. \quad (2.1)$$

Dabei ist α der Winkel zwischen dem einfallendem Strahl und dem Einfallslot und β der Winkel zwischen dem gebrochenen Strahl und dem Einfallslot, wie auch der Abbildung 2 zu entnehmen ist [Niedrig, 1992]. Die Konstante setzt sich aus den optischen Materialeigenschaften beider Medien zusammen. Weißt man jedem Medium eine eigene Konstante, die *optische Brechzahl* n zu, hat das *Snellius'sche Brechungsgesetz* folgende Form [Hecht, 2005]:

$$\frac{\sin \alpha}{\sin \beta} = \frac{n_2}{n_1} = \text{const}. \quad (2.2)$$

Im Vakuum gilt eine Brechzahl $n_{vac} = 1$. Zwar muss eine höhenabhängige *atmosphärische Strahlenbrechung* und Reflexion in genaueren

photometrischen Auswertungen berücksichtigt werden, jedoch kann sie in der Fernerkundung vernachlässigt werden [Alberz, 2001]. Dies ist darauf zurückzuführen, dass der Brechungsindex für die Luft auf Meereshöhe

$$n_{Luft,0km} = 1,00029$$

beträgt. Zudem nimmt die Luftdichte nach oben hin exponentiell ab, sodass die Brechzahl in etwa 8 km Höhe nur noch

$$n_{Luft,8km} = 1,00011$$

beträgt [Wikipedia, 2012a]. Folglich darf wegen der Beziehung $n_{Luft} \approx n_{vac}$ in einem Strahlungstransfermodell der vereinfachende Zusammenhang $\alpha = \beta$ gelten.

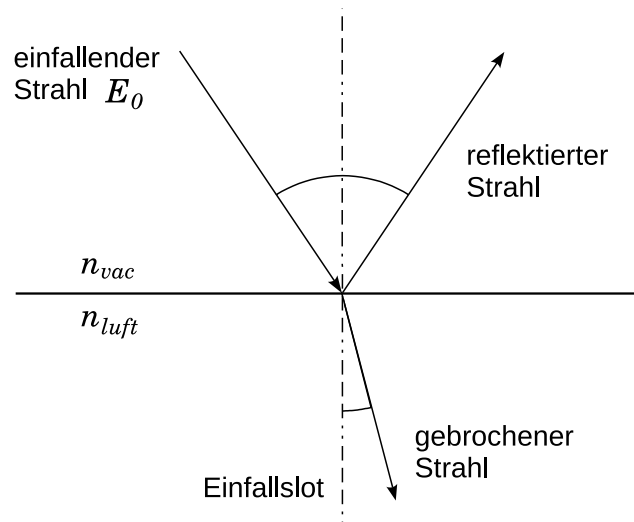


Abbildung 2: Reflexion und Brechung von Licht an einer Grenzfläche

Von weitaus größerer Bedeutung sind die Einflüsse durch Absorption und Streuung. Die *Absorption* ist eine Energieumwandlung, bei der Strahlung in Wärme oder andere Energieformen umgewandelt wird. Trifft ein Photon mit der Energie $E = h\nu$, wobei h das Planck'sche Wirkungsquantum¹ und ν die Frequenz der Lichtwelle sind, auf ein Atom, so kann dieses das Photon absorbieren und geht dabei in einen energetisch höheren Zustand über. Die Energie des einfallenden Photons entspricht dabei genau der Energiedifferenz zwischen Anfangszustand und angeregtem Zustand des Atoms, wie auch der Abbildung 4a zu

¹ $h = 6,626 \cdot 10^{-34} \text{ J s}$

entnehmen ist. In diesem Fall spricht man von einer *Resonanzabsorption* [Tipler and Mosca, 2006].

Werden die Gasatome in der Erdatmosphäre einer Strahlung ausgesetzt, so zeigt der transmittierte (durchgelassene) Strahlungsanteil mehr dunkle Linien als zuvor. Diese Absorptionsspektren sind von den in der Atmosphäre vorhandenen Atomen und Molekülen abhängig. Vor allem Wasserstoff, Kohlendioxid und Ozon absorbieren einige Spektralbereiche nahezu vollständig. Ultraviolette Strahlung unterhalb von $0,3 \mu\text{m}$ wird sogar vollständig durch Stickstoff und Sauerstoff absorbiert, da diese den größten Anteil in der Atmosphärenzusammensetzung ausmachen. Am anderem Ende des Spektrums, oberhalb von wenigen mm , ist die Atmosphäre für Mikrowellen wiederum vollständig durchlässig. Die Durchlässigkeit der Atmosphäre wird durch den *spektralen Transmissionsgrad* τ gekennzeichnet. Abbildung 3 verdeutlicht die Abhängigkeit des Transmissionsgrads von der Wellenlänge [Alberz, 2001]. Besonderer Augenmerk liegt auf den durchlässigen Bereichen des Spektrums, da lediglich diese für die Fernerkundung in Frage kommen. Deswegen werden diese auch anschaulich als *atmosphärische Fenster* bezeichnet.

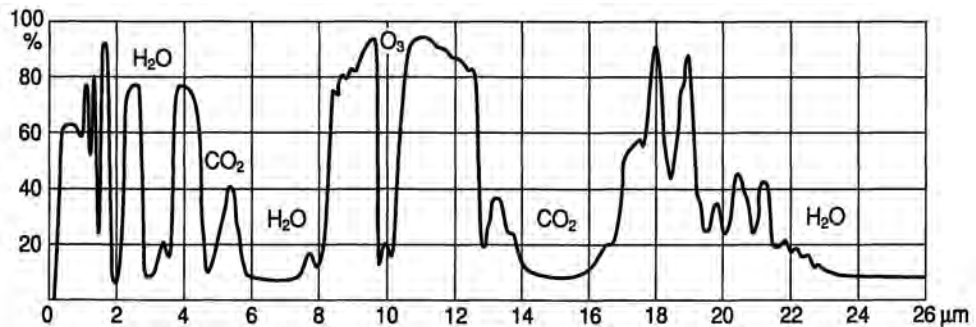


Abbildung 3: spektraler Transmissionsgrad τ der Atmosphäre

Wie in der Atmosphäre, erfolgt auch bei der Reflektion an der Erdoberfläche selbst eine gewisse *Absorption*, welche in der Literatur mit dem *Albedo*

$$\rho(\lambda) = \frac{\Phi_{\lambda r}}{\Phi_{\lambda i}} \quad (2.3)$$

zum Ausdruck gebracht wird. Dabei ist $\Phi_{\lambda i}$ die einfallende und $\Phi_{\lambda r}$ die vom Erdboden reflektierte Strahlungsleistung. Dieses Verhältnis kann einen Wert zwischen 0 und 1 annehmen. Folglich entspricht ein Albedo von 1 einer Totalreflektion während ein Rückstrahlvermögen von 0 eine vollständige Absorption der einfallenden Strahlung bedeutet. Es muss

beachtet werden, dass diese Größe stark von der Oberflächenbeschaffenheit, der Einfallrichtung der Strahlung und der Wellenlänge abhängt [Palmer and Grant, 2010].

Desweiteren wird die Strahlung in der Atmosphäre durch zusätzliche *Streuung* an kleinen Materieteilchen, wie zum Beispiel Aerosole, Wasser- oder Eiswolken geschwächt, durch welche die Strahlung in alle Richtungen hin abgelenkt wird. Aufgrund dieser Streuung wird die ganze Atmosphäre mit einer diffusen Strahlung erfüllt. Von der Erdoberfläche gesehen kann diese diffuse Strahlung als *Himmelslicht*² wahrgenommen werden (Abbildung 5). Mit zunehmendem Anteil der Materieteilchen, nimmt auch das diffuse Himmelslicht zu. Der direkte Anteil der extraterrestrischen Sonnenstrahlung E_0 , welcher den Boden erreicht, nimmt dabei jedoch zeitgleich ab. Betrachtet man die Streuung auf atomarer Ebene, wird ein Photon welches mit einem Atom zusammenstößt, von diesem abgelenkt. Im Gegensatz zur Absorption geht jedoch nicht die ganze Energie des Photons in das Atom über. Hat das Photon nach dem Zusammenstoß noch seine komplette Energie, so spricht man von *elastischer Streuung* (Abbildung 4b). Gibt es jedoch einen Teil seiner Energie an das Atom ab und versetzt es damit in einen angeregten Zustand, so spricht man von *unelastischer Streuung* oder *Stokes-Raman-Streuung* (Abbildung 4c). Es bleibt anzumerken, dass hier lediglich die bedeutendsten Wechselwirkungen zwischen Photonen und Atomen dargestellt wurden. Eine vollständige Übersicht aller Reaktionen kann in [Tipler and Mosca, 2006] gefunden werden.

Obwohl Absorption und Streuung zwar völlig unterschiedliche Ursachen besitzen, haben sie dennoch die selbe Wirkung. Beide schwächen die durch die Atmosphäre verlaufende Strahlung. Die Phänomene der Absorption und Streuung werden deswegen oftmals unter dem Begriff *Extinktion* zusammengefasst.

2.2 GESUCHTE PHYSIKALISCHE AUSGABEGRÖSSEN

Im Rahmen dieser Arbeit soll ein Strahlungstransfermodell letztlich dazu dienen, mehr Informationen über die am Sensor detektierbare Strahlung zu erhalten. Dabei sind vor allem die Größen der Bestrahlungsstärke, der spektralen Strahldichte und der Transmission von Interesse. Um

² Befinden sich im Himmel verhältnismäßig wenig Materieteilchen, so überwiegt der am Boden direkt ankommende ultraviolette bis blaue Spektralbereich und ist als blauer Himmel wahrnehmbar. Der Grund hierfür ist die in Tipler and Mosca [2006] beschriebene *Rayleigh-Streuung*.

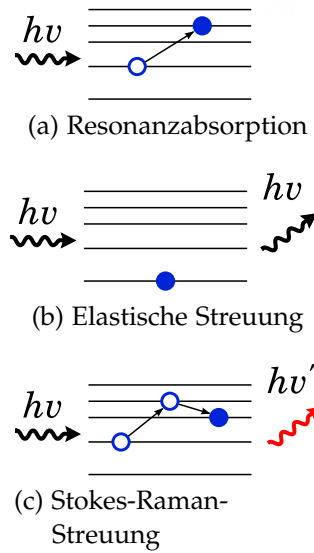


Abbildung 4: Wechselwirkungen zwischen Photon und Atom

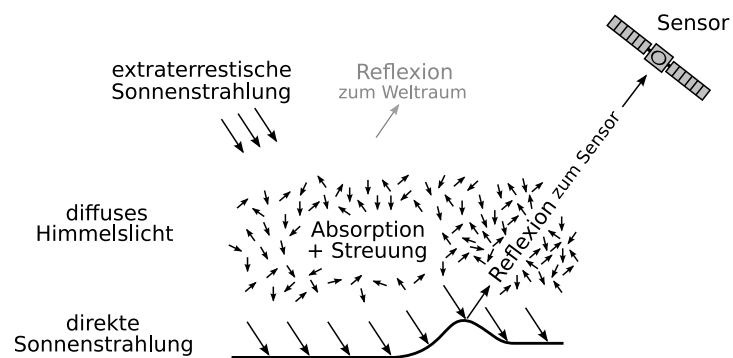


Abbildung 5: Strahlung in der Atmosphäre

sich der englischen Fachliteratur anzupassen und um Verwechslungen auszuschließen, wird im Weiteren die Bestrahlungsstärke als *Irradiance* und die spektrale Strahldichte als *Radiance* bezeichnet.

2.2.1 Radiance

Eine fundamentale Größe ist die Strahlungsenergie Q . Beobachtet man die Änderung der Strahlungsenergie dQ über einen Zeitraum dt , so erhält man die Strahlungsleistung Φ , die auch als strahlungsbedingter Fluss (*radiant flux*) bezeichnet wird. Es gilt:

$$\Phi = \frac{dQ}{dt} \quad (2.4)$$

Allerdings gibt die Strahlungsleistung keine Information über die räumlichen Verteilung. Um die Flächen- und Richtungsabhängigkeit zu berücksichtigen, wird die Strahldichte L als die Energie pro Fläche und Raumwinkel (Einheit: $W m^{-2} sr^{-1}$) definiert:

$$\begin{aligned} L &= \lim_{\Delta A_q, \Delta \omega \rightarrow 0} \left(\frac{\Delta \Phi}{\Delta A_q \cos \theta \Delta \omega} \right) \\ &= \frac{d^2 \Phi}{dA_q \cos \theta d\omega} \\ &= \frac{d^2 \Phi}{dA_q d\Omega} \\ &= \frac{d^2 \Phi}{dA_p d\omega}. \end{aligned} \quad (2.5)$$

Dabei ist θ der Polarwinkel zwischen der Normalen der Quelle und der Richtung des Empfängers. A_q ist die Fläche der Quelle und A_p ist die projizierte Fläche am Empfänger. Die Richtungsabhängigkeit der Strahldichte wird bestimmt durch den projizierte Raumwinkel $d\Omega$, bzw. dem durch die Fläche des Empfängers beschränkten Raumwinkels $d\omega$. Zwischen $d\Omega$ und $d\omega$ besteht der Zusammenhang

$$d\Omega = d\omega \cos \theta = \sin \theta \cos \theta d\theta d\phi, \quad (2.6)$$

welcher auch in Abbildung 6 wiedergegeben wird. ϕ ist dabei der Azimutwinkel [Palmer and Grant, 2010].

Im Falle der Auswertung von Strahlungstransfermodellen ist die *spektrale Strahldichte*, im Weiteren *Radiance* genannt, von besonderem Interesse. Im Vergleich mit der bisherigen Strahldichte, berücksichtigt sie die Wellenlängenabhängigkeit der Strahlung und hat damit die Einheit $W m^{-2} sr^{-1} nm^{-1}$. Die Radiance lässt sich durch Integration über die Wellenlänge wieder in die ursprüngliche Form zurückführen.

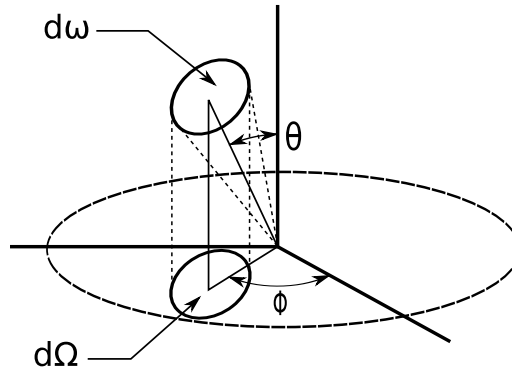


Abbildung 6: Parameter zur Beschreibung der Radiance

2.2.2 Irradiance

Die Irradiance E ist die Strahlungsleistung pro Flächeneinheit, welche auf eine Oberfläche trifft (Einheit: $W m^{-2}$) und ist definiert als:

$$E = \lim_{\Delta A_q \rightarrow 0} \left(\frac{\Delta \Phi}{\Delta A_q} \right) = \frac{d\Phi}{dA_q}. \quad (2.7)$$

Man erhält die Irradiance auch, indem man die Strahldichte L aus der Gleichung 2.5 über den projizierten Raumwinkel Ω der Hemisphäre integriert:

$$E = \int_{\pi} L d\Omega. \quad (2.8)$$

Integriert man die Irradiance E (Gleichung 2.7) über die Fläche, so erhält man die Strahlungsleistung Φ [Palmer and Grant, 2010]:

$$\Phi = \int_A E dA. \quad (2.9)$$

2.2.3 Transmission

Eine Transmission tritt auf, wenn Strahlung ein Medium auf einer anderen, meist gegenüberliegenden Seite verlässt, als es dort eingetreten ist. Der spektrale Transmissionsgrad $\tau(\lambda)$ ist dabei ein Maß für die von der Wellenlänge abhängigen Durchlässigkeit des Mediums. Er ist das Verhältnis zwischen der transmittierten spektralen Strahlungsleistung $\Phi_{\lambda t}$ und der einfallenden (*incident*) spektralen Strahlungsleistung $\Phi_{\lambda i}$:

$$\tau(\lambda) = \frac{\Phi_{\lambda t}}{\Phi_{\lambda i}}. \quad (2.10)$$

Der gesamte Transmissionsgrad τ gibt an, welcher Strahlungsanteil Φ_t von der Gesamtstrahlung Φ_i durch das Medium gelangt und wird beschrieben durch:

$$\tau = \frac{\Phi_{at}}{\Phi_i} = \frac{\int_0^\infty \tau(\lambda) \Phi_{\lambda i} d\lambda}{\int_0^\infty \Phi_{\lambda i} d\lambda} \neq \int_\lambda \tau(\lambda) d\lambda \quad (2.11)$$

Dabei wird ersichtlich, wie der gesamte Transmissionsgrad nicht durch die Integration über die Wellenlängen des spektralen Transmissionsgrads $\tau(\lambda)$ gewonnen werden kann, sondern durch die einfallenden spektrale Strahlungsleistung $\Phi_{\lambda i}$ gewichtet wird [Palmer and Grant, 2010].

Der Transmissionsgrad ist dimensionslos und kann nur Werte zwischen 0 und 1 annehmen. In Strahlungstransfermodellen ist er insofern von Bedeutung, als dass die Transmissivität der Atmosphäre großen Einfluss auf die am Sensor detektierbare Radiance hat.

2.3 ABSORPTIONSBESTIMMUNG MIT MOLEKULAREN DATENBANKEN

Für die Berechnung der Absorption greift man auf physikalische Datenbanken zurück, welche über das molekulare Absorptionsverhalten der jeweiligen Wellenlänge Auskunft erteilen. Dabei unterscheidet man zwei Verfahren:

Beim *line-by-line* Verfahren wird jede in der Datenbank vorhandene Spektrallinie zur Bestimmung der molekularen Absorption herangezogen. Eine solche Berechnung ist zwar am genauesten, aber auch sehr umfangreich und somit äußerst zeitintensiv.

Alternativ kann man auf ein *Bändermodell* zurückgreifen, in welchem die einzelnen Spektrallinien über geeignete Wellenlängenbereiche (Bänder) parametrisiert werden. Die genaueste dieser Näherungen ist die *correlated-k approximation* [Ricchiazzi et al., 1998]. Damit kann die Rechenzeit auf Kosten der Genauigkeit, drastisch reduziert werden. Abbildung 7 veranschaulicht, wie das *Bändermodell* aus einer *line-by-line*-Datenbank zustande kommt.

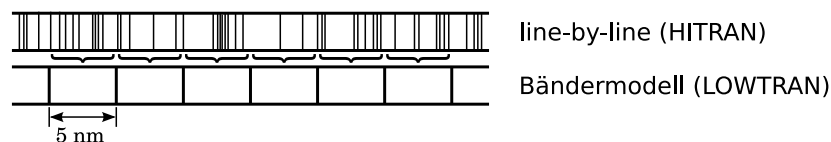


Abbildung 7: Erstellung des *Bändermodells* anhand einer *line-by-line*-Datenbank

Abbildung 8 verdeutlicht in einem extremen Beispiel, wie drastisch die Genauigkeit unter der Verwendung einer *correlated-k approximation*

nach [Kato et al., 1999] leiden kann, indem diese einer *line-by-line* Simulation mit *MODTRAN₃* nach [Kneizys et al., 1988] gegenübergestellt wird [Lohmann, 2006].

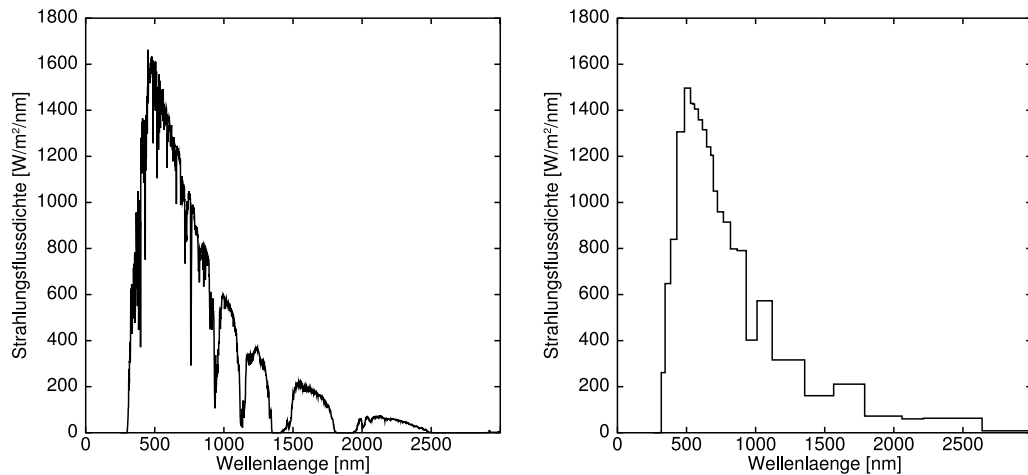


Abbildung 8: Vergleich der solaren Bestrahlungsstärke unter Verwendung eines *line-by-line*- (links) und Bändermodells (rechts)

Die Datenbank *HITRAN* (*High Resolution Transmission*) stellt den Standard für Simulationen und Berechnungen vom infraroten bis hin zum ultravioletten Bereich des Lichtspektrums dar. Die Datenbank erfasst alle bekannten und für die Absorption relevanten Spektrallinien, was die Grundlage einer *line-by-line* Berechnung darstellt. Ursprünglich wurde *HITRAN* ins Leben gerufen, um eine genauere Kenntnis über die infraroten Eigenschaften der Atmosphäre zu erlangen. Seit Ende der 60er-Jahre bis heute wird diese Datenbank von den AIR FORCE CAMBRIDGE RESEARCH LABORATORIES (AFCL) gepflegt und erweitert. Wegen seiner langen Lebensdauer mit zahlreichen Modifikationen kann *HITRAN* als sehr genau eingestuft werden [Harvard-Smithsonian, 2012] und wurde aus diesem Grund für die vorliegende Arbeit ausgewählt. Zudem ist *HITRAN* nach einer kostenlosen Registrierung frei zugänglich.

LOWTRAN hingegen ist eine von *HITRAN* abgeleitete *correlated-k* Parametrisierung. Dieses Bändermodell hat zwar nur eine niedrige Auflösung von 5 nm im sichtbaren Lichtspektrum, ist aber sehr schnell und daher gut geeignet für erste grobe Analysen über ein breites Spektrum [Ricchiazzi et al., 1998]. Im Kapitel 3.1.8 wird beispielhaft auf *LOWTRAN* zurückgegriffen, um die atmosphärischen Fenster zügig identifizieren zu können.

Falls ein Spektrum untersucht wird, in welchem die molekulare Absorption vernachlässigt werden kann oder soll, so ist die Verwendung einer molekularen Absorptionsdatenbank nicht obligatorisch. Beim Lösen der Strahlungstransfergleichung können molekulare Absorptionseinflüsse auch ignoriert, und die Rechendauer stark verkürzt werden.

2.4 ZENTRALE STRAHLUNGSTRANSFERGLEICHUNG

Um ein Strahlungstransfermodell aufzubauen und zu lösen, werden *Gleichungslöser-Programme*, auch *Solver* genannt, eingesetzt. Je nach geometrischen Szenario oder gewünschtem Output, muss ein geeigneter Solver gewählt werden. Um ein besseres Verständnis des Problems zu erlangen, wird im Folgendem die Strahlungstransfergleichung hergeleitet, welche von Solvern gelöst werden kann.

Allgemein kann die statistische Verteilung von Photonen in einem verdünnten Gas durch die Boltzmann-Gleichung beschrieben werden [Mayer et al., 2011]:

$$\frac{\partial f}{\partial t} + \nabla_{\mathbf{r}}(\mathbf{v} f) + \nabla_{\mathbf{p}}(\mathbf{F} f) = Q(\mathbf{r}, \hat{n}, \nu, t). \quad (2.12)$$

Dabei ändert sich die Funktion für die Photonenverteilung $f(\mathbf{r}, \hat{n}, \nu, t)$ mit dem Ort r , der Ausbreitungsrichtung \hat{n} , der Frequenz ν und der Zeit t . Es wird definiert, dass

$$f(\mathbf{r}, \hat{n}, \nu, t) c \hat{n} d\mathbf{S} d\Omega d\nu dt \quad (2.13)$$

die Anzahl der Photonen mit einer Frequenz zwischen ν und $\nu + d\nu$ ist, die eine Oberfläche dS in die Richtung \hat{n} , den Raumwinkel $d\Omega$ und in der Zeit dt durchquert [Stamnes, 1986]. Die Einheit von $f(\mathbf{r}, \hat{n}, \nu, t)$ ist $\text{cm}^{-3} \text{sr}^{-1} \text{Hz}^{-1}$ und c ist die Lichtgeschwindigkeit. Des Weiteren sind $\nabla_{\mathbf{r}}$ bzw. $\nabla_{\mathbf{p}}$ räumliche Differentialoperatoren im Konfigurations- bzw. Impulsraum. Sie werden durch das Nabla-Symbol ∇ beschrieben, das im dreidimensionalen Fall in kartesischen Koordinaten folgende Form hat:

$$\nabla = \begin{pmatrix} \frac{\partial}{\partial x_1} \\ \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_3} \end{pmatrix}. \quad (2.14)$$

Die Komponenten des Nabla-Operators sind als partielle Ableitungsoperatoren aufzufassen. $\frac{\partial}{\partial x_i}$ schreibt also die partielle Ableitung nach x_i vor [Bronstein et al., 2001]. Der Term $\mathbf{F}(\mathbf{r}, \hat{n}, \nu, t)$ beschreibt die äußere Kraft, welche auf die Photonen wirken kann. Die rechte Seite $Q(\mathbf{r}, \hat{n}, \nu, t)$ der Gleichung 2.12 stellt die Photonenquellen und -Senken dar, wie sie beispielsweise durch die extraterrestrische Strahlung, molekulare Absorption oder Reflexionen an der Erdoberfläche auftreten. Vernachlässigt man relativistische Effekte, so ist $\mathbf{F} = 0$ und die Photonen breiten sich mit der Geschwindigkeit $\mathbf{v} = c \hat{n}$ aus. Der Differentialoperator

des Konfigurationsraumes kann, gemäß der Produktregel der Differentialrechnung [Brauch et al., 2006], geschrieben werden als:

$$\nabla_{\mathbf{r}}(\mathbf{v}f) = f(\nabla_{\mathbf{r}}\mathbf{v}) + \mathbf{v}(\nabla f) = \mathbf{v}(\nabla f), \quad (2.15)$$

wobei \mathbf{r} und \mathbf{v} unabhängige Variablen sind. Gleichung 2.12 kann somit folgendermaßen umgeformt werden:

$$\frac{\partial f}{\partial t} + c(\hat{n}\nabla)f = Q(\mathbf{r}, \hat{n}, \nu, t). \quad (2.16)$$

Der Index \mathbf{r} des Nabla-Operators ∇ wurde dabei ausgelassen. Die differentielle Energie, die von der Photonenverteilung abhängt, ist

$$dE = c h \nu f \hat{n} d\mathbf{S} d\Omega d\nu dt. \quad (2.17)$$

Setzt man den Term

$$\hat{n} d\mathbf{S} = \cos \theta dS \quad (2.18)$$

in die Gleichung 2.17 ein, so lässt sich die spezifische Intensität der Photonen $I(\mathbf{r}, \hat{n}, \nu, t)$ über

$$dE = I(\mathbf{r}, \hat{n}, \nu, t) \cos \theta dS d\Omega d\nu dt \quad (2.19)$$

herleiten und ergibt

$$I(\mathbf{r}, \hat{n}, \nu, t) = c h \nu f(\mathbf{r}, \hat{n}, \nu, t). \quad (2.20)$$

Im Falle eines stationären Zustands, also $\frac{\partial f}{\partial t} = 0$, kann die Gleichung 2.16 geschrieben werden als

$$(\hat{n}\nabla)I(\mathbf{r}, \hat{n}, \nu) = h \nu Q(\mathbf{r}, \hat{n}, \nu), \quad (2.21)$$

was die *Strahlungstransfergleichung* in allgemeiner Geometrie beschreibt. Eine weitere gängige Bezeichnung für die Strahlungstransfergleichung ist RTE (*Radiative Transfer Equation*).

Der Term $Q(\mathbf{r}, \hat{n}, \nu)$ repräsentiert alle Quellen und Senken des Strahlungsganges mit der Ausbreitungsrichtung \hat{n} und der Frequenz ν , wie sie durch Streuung, Absorption und Wärmestrahlung auftreten können. Dieser Term muss angegeben werden, damit die Gleichung gelöst werden kann. Für dessen Herleitung sei auf [Chandrasekhar, 1960] verwiesen.

Der Strömungsterm $\hat{n}\nabla$ definiert die Geometrie. Die Atmosphäre eines Planeten kann mittels einer kartesischen oder sphärischen Geometrie definiert werden. Während bei kartesischen Geometrien für gewöhnlich planparallele Näherungen verwendet werden, finden bei sphärischen

Geometrien meist pseudospherische Näherungen Anwendung. Im Folgendem soll lediglich die für unsere Zwecke ausreichende kartesische Planetengeometrie beschrieben werden.

In einem kartesischem Koordinatensystem kann der Strömungsterm gemäß [Rottmann, 1991] geschrieben werden als

$$\begin{aligned}\hat{n}\nabla &= n_x \frac{\partial}{\partial x} + n_y \frac{\partial}{\partial y} + n_z \frac{\partial}{\partial z} \\ &= \cos \phi \sqrt{1 - \mu^2} \frac{\partial}{\partial x} + \sin \phi \sqrt{1 - \mu^2} \frac{\partial}{\partial y} + \mu \frac{\partial}{\partial z}.\end{aligned}\quad (2.22)$$

Dabei sind n_x , n_y und n_z die Komponenten des Einheitsvektors, ϕ ist der Azimuthwinkel und $\mu = \cos \theta$.

In Worte gefasst, verwendet die planparallele Geometrie eine flache Erde als Näherung. Dabei erstrecken sich die einzelnen Schichten der Atmosphäre unendlich in die x - und y -Richtung. Damit kann der Strömungsterm letztendlich vereinfacht werden zu

$$\hat{n}\nabla = n_z \frac{\partial}{\partial z}.\quad (2.23)$$

Mit festgelegten Randbedingungen kann die Strahlungstransfergleichung 2.21 gelöst werden und gibt dann das Strahlungsfeld in der Atmosphäre wider. Die Randbedingungen sind dabei die TOA und TOC³ einfallende Strahlung. Während am Boden der Atmosphäre ein Teil der einfallenden Strahlung reflektiert wird, ist die TOA einfallende Strahlung durch die extraterrestrische Sonnenstrahlung festgelegt. Die Strahlungsintensität I in der Atmosphäre kann also beschrieben werden durch:

$$I(z, \mu, \phi) = I^{dir}(z, \mu_0, \phi_0) + I^{sca}(z, \mu, \phi).\quad (2.24)$$

Dabei ist I^{dir} der direkte Anteil der extraterrestrischen Sonnenstrahlung und I^{sca} der gestreute (*scattered*) Anteil. z ist die Höhe über dem Meeresspiegel, μ bzw. ϕ ist der Sonnenzenit- bzw. Sonnenazimuthwinkel und μ_0 bzw. ϕ_0 deren Kosinus. Setzt man Gleichung 2.24 in Gleichung 2.21, so erhält man die Strahlungstransfergleichung in der Form

$$-\mu \frac{dI^{dir}(z, \mu_0, \phi_0)}{\beta^{ext} dz} = -\mu \frac{dI^{dir}(z, \mu_0, \phi_0)}{d\tau} = I^{dir}(z, \mu_0, \phi_0),\quad (2.25)$$

wobei die optische Dicke als $d\tau = \beta^{ext} dz$ definiert ist [Mayer et al., 2011]. Der Extinktionskoeffizient β^{ext} wird in [Stamnes, 1986] definiert. Die Lösung der Gleichung 2.25 für die direkte Strahlung, ergibt das *Lambert-Beersches Gesetz*:

$$I^{dir}(\tau, \mu_0) = I^0 \exp\left(-\frac{\tau}{\mu_0}\right).\quad (2.26)$$

³ Während *Top-of-Atmosphere* (TOA) das obere Ende der Atmosphäre darstellt, ist *Top-of-Canopy* (TOC) das untere Ende.

Die Gleichung 2.25 ist eine zentrale Formel, welche von zahlreichen Gleichungslösern in Strahlungstransfermodellen gelöst wird. Allem voran durch den RTE Solver *DISORT* (*DIScrete ORdinate Radiative Transfer*), welcher in dieser Arbeit aufgrund seiner Eignung für die vorliegenden Szenarien, verwendet wird. *DISORT* ist stark ausgereift, vielseitig einsetzbar und erfreut sich in der Atmosphärenwissenschaft einer großen Beliebtheit [Stamnes et al., 2000].

2.5 BERECHNUNG DER AUSGABEGRÖSSEN ÜBER DIE OPTISCHE DICKE

Die durch Absorption und Streuung bedingte Strahlungsschwächung in einem optisch aktivem Medium wie der Atmosphäre, wird anhand der *optischen Dicke* τ bemessen. Sie ist insbesondere bei Anwendung des *line-by-line* Verfahrens von Bedeutung, weil hier die aus der molekularen Absorptionsdatenbank gewonnenen Parameter, in die optische Dicke überführt werden. Gleichung 2.26 zeigt, wie anhand der optischen Dicke τ die Strahlungstransfergleichung gelöst werden kann. Aufgrund ihrer Signifikanz soll in diesem Kapitel gezeigt werden, wie sie sich durch die Parameter aus einer molekularen Absorptionsdatenbank bestimmen lässt.

Die optische Dicke einer Atmosphärenschicht $\Delta\tau_\nu$ in der Höhe z über dem Meeresspiegel ist nach [Evans, 2001a]

$$\Delta\tau_\nu(z) = k_\nu(z)\Delta u(z) = \left[\sum_i S_i(T)f(\nu, \alpha_i) \right] \Delta u(z), \quad (2.27)$$

wobei der Index ν auf die Wellenlängenabhängigkeit der Variablen hinweist. $\Delta u(z)$ ist die absorbierte Masse zwischen den Höhen z_1 und z_2 und hat die Einheit g/cm^2 . Gemäß [Evans, 2001b] kann $\Delta u(z)$ beschrieben werden durch:

$$\Delta u(z) = \int_{z_1}^{z_2} \rho_a(z) dz, \quad (2.28)$$

wobei ρ_a die Dichte der dämpfenden Materie, also den in der Atmosphäre vorhandenen Molekülen ist.

Der linke Term der Gleichung 2.27 ist der Absorptionskoeffizient k_ν und hat die Definition:

$$k_\nu = S f(\nu, \alpha), \quad (2.29)$$

wobei S die Stärke der spektralen Absorption und $f(\nu, \alpha)$ eine Linienformfunktion ist. Bislang wurde davon ausgegangen, dass die Energieniveaus bei der Extinktion scharf sind und nur bei der Wellenzahl ν_0 erfolgen. In Wahrheit haben sowohl die Niveaus als auch die Spektrallinien

eine Unschärfe, was mit der Linienformfunktion berücksichtigt wird [Eichler and Eichler, 2010]. Die Zusammenstöße der Moleküle in der Luft reduzieren die Lebensdauer der hohen Energiezustände und verbreitern somit die Absorptionslinien. Diese Stoßverbreiterung wird mit der Linienform $f(\nu, \alpha)$ beschrieben und wird als *Lorentz-Profil* bezeichnet [Evans, 2001a]:

$$f(\nu, \alpha) = \frac{\frac{\alpha}{\pi}}{(\nu - \nu_0)^2 + \alpha^2}. \quad (2.30)$$

In diesem Zusammenhang kann nun die Bedeutung von ν und ν_0 anhand der Abbildung 9 deutlich gemacht werden.

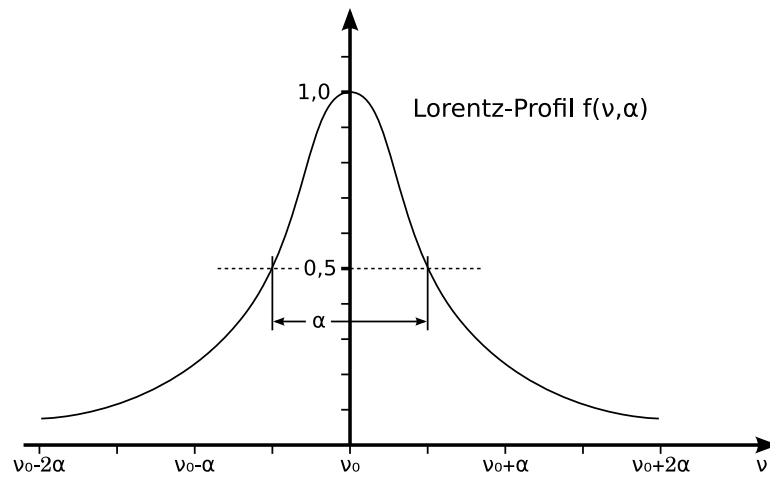


Abbildung 9: Linienformfunktion: *Lorentz-Profil*

Die *Halbwertbreite* α wird über die Skalierung des Drucks und der Temperatur bestimmt:

$$\alpha = \alpha^0 \left(\frac{p(z)}{p_0} \right) \left(\frac{T_0}{T(z)} \right)^n. \quad (2.31)$$

Die Verbreiterung α^0 setzt sich aus den Halbwertbreite der Luft $\alpha_{L,air}^0$ und des Moleküls $\alpha_{L,self}^0$ zusammen. Dabei wird auch das Mischungsverhältnis q beachtet, welches der Atmosphären-Datei entnommen werden kann:

$$\alpha^0 = (1 - q)\alpha_{L,air}^0 + q\alpha_{L,self}^0 \quad (2.32)$$

Die Gleichung 2.27 wird auf sämtliche in der Atmosphäre vorhandenen Moleküle angewendet. Die sich dabei ergebenden optischen Dicken werden dann beispielsweise folgendermaßen aufsummiert

$$\Delta\tau_\nu(z) = \Delta\tau_{\nu,H_2O}(z) + \Delta\tau_{\nu,O_3}(z) + \Delta\tau_{\nu,O_2}(z). \quad (2.33)$$

Abschließend wird $\Delta\tau_\nu(z)$ über die Höhe integriert, wodurch sich die gesamte optische Dicke τ_ν der Atmosphäre für die Wellenlänge mit der Wellenzahl ν ergibt [Evans, 2001a]:

$$\tau_\nu = \sum_j \Delta\tau_\nu(z_j). \quad (2.34)$$

In einer molekularen Absorptionsdatenbank wie *HITRAN* sind alle benötigten Parameter (vgl. Seite 61) gegeben, um die optische Dicke τ zu bestimmen. Setzt man Gleichung 2.34 in Gleichung 2.26 ein, so kann die Strahlungstransfergleichung gelöst werden.

STRAHLUNGSTRANSFERMODELLE

Aufgrund der Erfahrungswerte aus früheren Projekten, wurden eine Strahlungstransfermodell gesucht, welches folgende Kriterien erfüllt:

- Abdeckung des Spektrums von 300 nm bis 15 μm
- hohe Auflösung des Spektrums
- Möglichkeit zur *line-by-line* Berechnung
- das Strahlungstransfermodell muss auf die Datenbank *HITRAN* zurückgreifen können
- sämtliche verwendete Software muss kostenlos und frei zugänglich sein

Eine umfassende Übersicht zu allen gängigen Strahlungstransfermodellen ist unter [Wikipedia, 2012b] zu finden. Letztendlich fiel die Wahl auf *libRadtran*, welches alle Anforderungen erfüllt und das Spektrum weit über die Voraussetzungen hinaus abdeckt.

3.1 LIBRADTRAN (FREI VERFÜGBAR)

libRadtran ist eine Bibliothek mit Programmen und Routinen, mit welchen Strahlungstransfermodelle simuliert und gelöst werden können. Zudem steht die Bibliothek unter der GNU GENERAL PUBLIC LICENSE und ist somit frei zugänglich. Wesentlicher Bestandteil des *libRadtran* Pakets ist das Strahlungstransferprogramm *uvspec*¹, das ein Spektrum von 120 nm bis 100 μm umfasst [Mayer et al., 2011, Seite 1]. Es zeichnet sich durch seine Benutzerfreundlichkeit und Vielzahl von Einstellungsmöglichkeiten aus. So ist es beispielsweise möglich die Atmosphäre mit Aerosolen, Molekülen, Wasser- oder Eiswolken zu modifizieren. Des Weiteren lässt sich in *uvspec* auch das Beobachtungsszenario über Sonnenstand und Beobachtungsposition definieren.

3.1.1 Strahlungstransfersimulation mit *uvspec*

Die Strahlungstransfergleichung 2.21 kann mit *uvspec* und unter Beachtung der Randbedingungen nach dem Strahlungsfeld I aufgelöst wer-

¹ Der Name *uvspec* ist historisch bedingt und mag irreführend sein, da in frühen Versionen lediglich das UV-Spektrum abgedeckt wurde.

den. Die Angabe der Randbedingungen in *uvspec* erfolgt durch eine Input-Datei, die vom Nutzer anzufertigen ist. Eine Auflistung und Erklärung sämtlicher Input-Optionen ist in [Mayer et al., 2011, Kapitel 6] zu finden.

Der Abbildung 10 kann man entnehmen, in welche drei Teile *uvspec* gegliedert ist [Mayer et al., 2011, Kapitel 3]. Zuerst muss eine Beschreibung der Atmosphäre in Form der Input-Datei und Datenbanken erfolgen, woraus die benötigten optischen Eigenschaften der Atmosphäre errechnet werden. Die Form einer solchen Input-Datei wird im Kapitel 3.1.3 beschrieben.

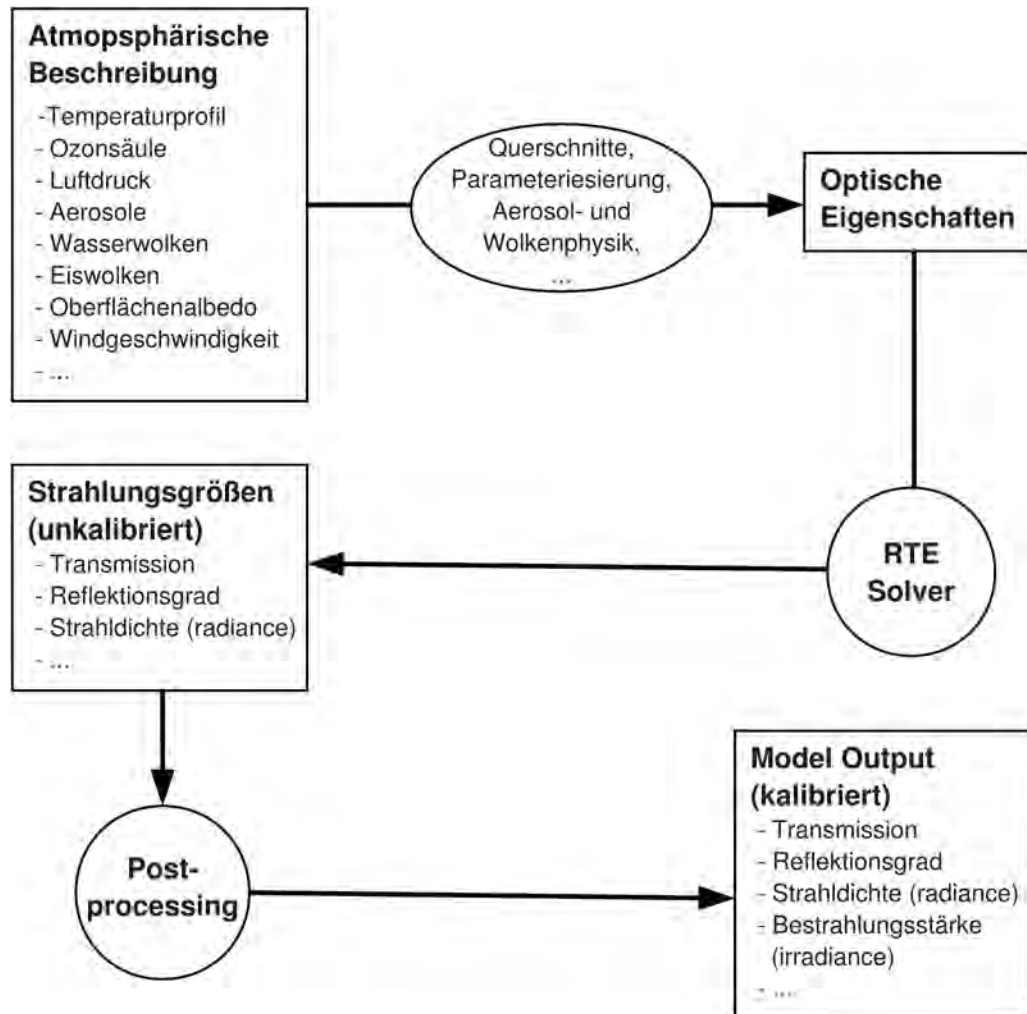
Im zweiten Schritt ermittelt der RTE Solver aus den gewonnenen optischen Eigenschaften die Strahlungsgrößen in der Atmosphäre, wie zum Beispiel deren Transmissions-, Reflexions- und Absorptionsgrad oder auch die höhenabhängige Irradiance. Schließlich durchlaufen die noch unkalibrierten Strahlungsgrößen im letzten Schritt ein Postprocessing. Dabei wird unter anderem das Ergebnis mit einem Korrekturfaktor der extraterrestrischen solaren Bestrahlungsstärke multipliziert, welcher aus dem Sonnenabstand resultiert. Des Weiteren kann das Ergebnis über die Wellenlänge integriert, oder auch durch Faltung mit einer Schlitzfunktion geglättet werden.

Einmalig ist das Feature von *uvspec*, dass es zehn verschiedene Solver für Strahlungstransfergleichungen zur Verfügung stellt. Ein Übersicht aller Solver und deren Beschreibung ist in [Mayer et al., 2011, Seite 26] zu finden. Der zu verwendende Solver kann durch einen Parameter in der *uvspec* Input-Datei festgelegt werden.

3.1.2 Installation und Ausführung von *uvspec*

libRadtran kann sowohl unter Windows als auch auf einem Unix-System ausgeführt werden². Im Rahmen dieser Arbeit wurde mit der Version *1.6-beta* gearbeitet. Die wichtigste Neuerung von diesem Release ist die Konvertierung des RTE Solvers *DISORT* von der Programmiersprache *Fortran* nach C. Der Grund hierfür liegt in der größeren Beliebtheit von C gegenüber dem in die Jahre gekommenen *Fortran*. So wird der Zugang zu *DISORT* vereinfacht und ermöglicht es einem größerem Kreis von Programmierern, dessen Weiterentwicklung voranzutreiben.

² Die freie Software *libRadtran* kann auf der Seite <http://www.libradtran.org/doku.php?id=download> heruntergeladen werden. Ältere Versionen und Installationsanleitungen sind dort ebenfalls verfügbar. (Stand: 20.05.2012)

Abbildung 10: Aufbau von *uvspec*

Außerdem konnte durch die Umstellung die Rechenzeit signifikant reduziert werden, da mit C eine dynamische Speicherzuweisung ermöglicht wird [Mayer, 2011]. Darunter versteht man die Möglichkeit, Speicher erst zur Laufzeit des Programms anzufordern, sodass die Größe von Arrays nicht bereits zum Zeitpunkt des Übersetzens des Programms festgelegt zu sein braucht.

Im Rahmen dieser Arbeit wurde *libRadtran* auf einer Workstation mit dem Betriebssystem *Kubuntu 12.04*³ ausgeführt. Bei Unix-Systemen müssen generell die Abhängigkeiten zwischen den Paketen (*package-dependencies*) beachtet werden. Weil die Abwärtskompatibilität einer neueren Version nicht immer gewährleistet ist, muss die in Tabelle 1 aufgeführte Software mit der angegebenen Version installiert sein, damit *libRadtran 1.6-beta* mit Sicherheit ausgeführt werden kann. Besonderes Augenmerk ist auf *gfortran* zu legen, das in der Version *4:4.6.1-2ubuntu5* installiert werden muss, obwohl bereits eine neuere Version erschienen ist. In diesem Zusammenhang kann beim Konfigurieren des Quellcodes auch ein Fehler auftreten, wenn zwei Fortran-Compiler installiert sind [libRadtran, 2012a]. Der Konflikt kann gelöst werden, indem der Konfigurations-Datei vorgeschrieben wird, welcher Compiler zu verwenden ist. Am einfachsten geschieht dies durch den Kommandozeilen-Befehl

```
export F77=gfortran
```

welcher vor dem Aufruf der Konfigurations-Datei erfolgen muss. Sollte sich *libRadtran* noch immer nicht ordnungsgemäß installieren lassen, so ist im Anhang in Abschnitt 9.1 die Behebung eines häufig auftretenden Programmfehlers beschrieben.

PROGRAMM	VERSION
GNU make	3.81-8.1ubuntu
GNU C compiler (gcc)	4.5.3-9ubuntu1
Fortran compiler (f2c)	20090411-1
gfortran	4:4.6.1-2ubuntu5
fort77	1.15-8
Flex	2.5.35.10ubuntu1
Perl modules	5.12.4-4
netcdf	1.9.1-dgb
libnetcdf-devel	1:4.1.1-5build2

Tabelle 1: von *libRadtran 1.6-beta* benötigte Software

³ *Kubuntu 12.04* ist ein freies Derivat der Distribution *Ubuntu* und bietet einen Support bis zum April 2017. Es ist erhältlich unter <http://www.kubuntu.org/getkubuntu>.

Soll *libRadtran* auf einem Windows-System ausgeführt werden, wird eine zusätzliche Kompatibilitätsschicht, auch Wrapper genannt, zwischen der Umgebung von Microsoft Windows und dem für Unix entwickelten Programm benötigt. Für die Übersetzung von *libRadtran* wird der kostenlose API-Wrapper *Cygwin* von RED HAT empfohlen [Baumann, 2008]. Dabei ist es auch unter *Cygwin* notwendig, die oben genannten Paketabhängigkeiten zu beachten.

Nach der Installation von *libRadtran* kann *uvspec* sowohl unter *Cygwin*, als auch nativ unter Unix, über die Kommandozeile ausgeführt werden. Dazu muss der Befehl

```
uvspec < input_file.INP > output_file.OUT
```

ausgeführt werden. Dabei muss das Kommando im selben Pfad aufgerufen werden, in dem auch die Dateien *input_file.INP* und *input_file.OUT* zu finden sind. Anderenfalls muss der Aufruf der In- und Output-Dateien den vollständigen Pfad, wie im folgenden Beispiel, beinhalten:

```
uvspec < /home/user/input.INP > /home/user/output.OUT
```

Warnungen, Fehlermeldungen und Zwischenwerte wie die berechneten optischen Eigenschaften können über den Befehl

```
uvspec < input.INP > output.OUT >& diagnostic.txt
```

in die Textdatei *diagnostic.txt* geschrieben werden. Dazu muss die Input-Datei jedoch den Befehl *verbose* beinhalten. Die Datei *diagnostic.txt* wird im selben Verzeichnis abgelegt, in dem auch die In- und Output-Dateien zu finden sind.

3.1.3 Erstellung der *uvspec* Input-Datei

Eine *uvspec* Input-Datei ist eine benutzerfreundliche Textdatei, die mit einem beliebigen Texteditor bearbeitet werden kann. In der Datei *input_file.INP* stellt jede Zeile einen Befehl dar.

Listing 1 stellt eine einfache Input-Datei dar, welche in drei Teile strukturiert ist. Der erste Abschnitt setzt allgemeine Rahmenbedingung wie den Wellenlängenbereich, der untersucht werden soll und wie stark dieser geglättet werden soll. Über den Parameter *atmosphere_file* können auch standardisierte Atmosphären geladen werden, die über den Temperatur-, Druck- oder Luftdichtenverlauf über die Höhe festlegen. Hervorzuheben ist, dass an dieser Stelle der Datei auch die Wahl des RTE Solver *DISORT* stattfindet. Im zweiten Abschnitt ist beispielhaft die Definition von Aerosolen dargestellt und der letzte Abschnitt beschreibt die Geometrie des Beobachtungsszenarios. Genauere Beschreibungen

der Parameter können dem Benutzerhandbuch von *libRadtran* entnommen werden [Mayer et al., 2011, Kapitel 6].

Listing 1: einfache Input-Datei für uvspec

```

1 # SET GENERAL CONDITION:
2 atmosphere_file ../data/atmmod/afglus.dat
3                                     # location of atmospheric profile file
4 solar_file ../data/solar_flux/atlas_plus_modtran
5                                     # location of extraterrestrial spectrum
6 ozone_column 300.                   # scale ozone column to 300.0 DU
7 day_of_year 170                      # correct for Earth-Sun distance
8 albedo 0.6                           # surface albedo
9 rte_solver disort                    # choose RTE solver
10 deltam on                            # delta-M scaling on
11 nstr 6                               # number of streams
12 wavelength 399.0 541.0              # wavelength range [nm]
13 spline 500 540 0.5                  # integrate wavelength
14                                     # from 500 nm to 400 nm in step 0.5 nm
15 slit_function_file ../examples/TRI_SLIT.DAT
16                                     # location of slit function
17 verbose                              # write diagnostic output
18
19 # SET AEROSOL PROPERTIES:
20 aerosol_vulcan 1                     # aerosol type above 2 km
21                                     # 1: no volcanic aerosols
22 aerosol_haze 5                       # aerosol type below 2 km
23                                     # 5: urban type aerosols
24 aerosol_season 1                     # 1: summer season
25 aerosol_visibility 15.0              # horizontal visibility in km
26 aerosol_angstrom 1.1 0.2            # scale aerosol optical depth
27                                     # using Angstrom alpha and beta
28                                     # coefficients
29 aerosol_scale_ssa 0.85               # scale the single scattering albedo
30                                     # for all wavelengths
31 aerosol_set_gg 0.70                  # set the asymmetry factor
32 aerosol_tau_file ../examples/AERO_TAU.DAT
33                                     # file with optical depth profile
34
35 # SET GEOMETRY:
36 zout TOA                             # sensor altitude above ground
37                                     # satellite is on Top Of Atmosphere
38 phi 90.0                             # sensor azimuth
39 umu 0.9659258                        # cosine of viewing zenith angle
40 phi0 210.0                           # solar azimuth
41 sza 15.0                              # solar zenith angle

```

PARAMETER	GEOMETRISCHE BEDEUTUNG
zout	Höhe des Sensors (Satellit) über dem Meeresspiegel
phi	Azimut des Sensors
umu	Kosinus vom Zenitwinkel des beobachtenden Sensors
phi0	Azimut der Sonne
sza	Zenitwinkel der Sonne

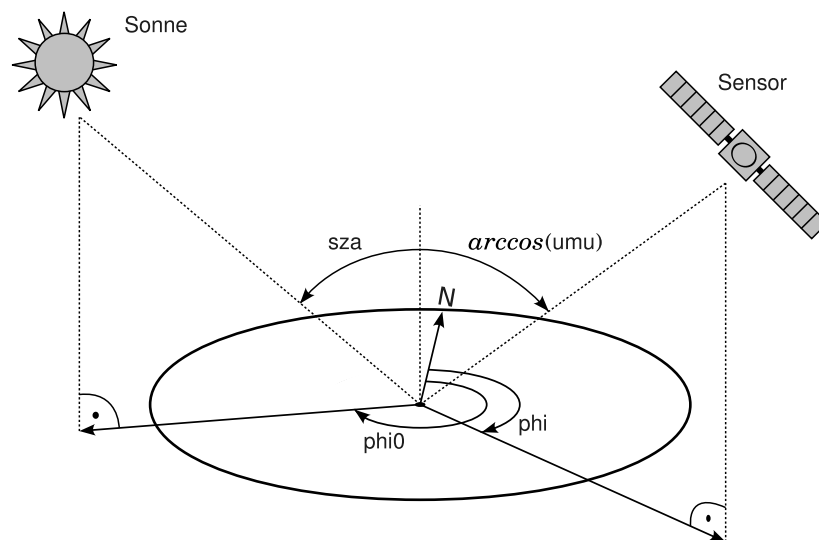
Tabelle 2: geometrische Parameter der *uvspec* Input-Datei

Abbildung 11: Geometrische Parameter zur Beschreibung des Beobachtungsszenarios

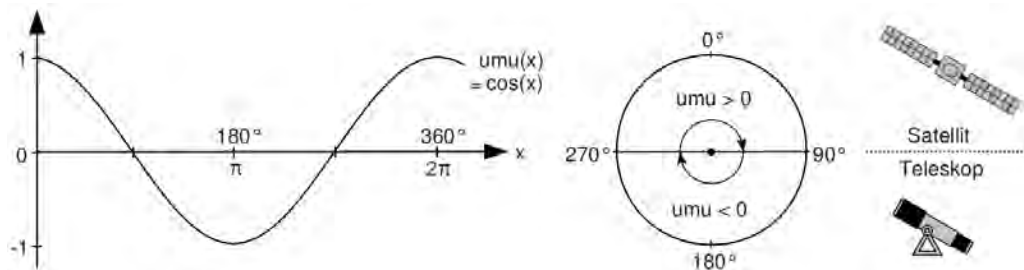
Abbildung 12: Visualisierung des geometrischen Parameters umu

Abbildung 11 visualisiert die Parameter, welche zur geometrischen Beschreibung des Beobachtungsszenarios notwendig sind. In Tabelle 2 werden diese Parameter nochmals samt einer Beschreibung aufgeführt. Besondere Beachtung muss man dem Gebrauch des Parameters `umu` gelten. Da in dieser Arbeit von dem Szenario eines Erdbeobachtungssatelliten ausgegangen wird, muss der Kosinus vom Zenitwinkel des Satelliten stets positiv sein. Hätte `umu` einen negativen Wert, so wäre der Sensor nicht der Erde zugewendet, sondern würde wie ein Teleskop in den Himmel blicken. Dieser Umstand wird in der Abbildung 12 anschaulich dargestellt.

Zur weiteren Vertiefung eignet sich das Handbuch von *libRadtran* [Mayer, 2012], welches eine kurze Einführung in die Erstellung der *uvspec* Input-Dateien darstellt. Zudem werden in [Mayer et al., 2011, Seite 29 ff.] weitere ausführlich erklärte Beispiele aufgeführt.

Um ein besseres Gefühl für die Größen der Ausgabe-Parameter zu entwickeln, sind in den Kapiteln 3.1.6, 3.1.7 und 3.1.8 drei gelöste Beispielszenarien zu finden.

3.1.4 Struktur der *uvspec* Output-Datei

Wie bereits die *uvspec* Input-Datei, liegt auch die Output-Datei in Form einer Textdatei vor. Das Format des Outputs ist jedoch stark von der Wahl des RTE Solvers abhängig. Im Rahmen dieser Arbeit wird nur der Output des Solver *DISORT* betrachtet. Alle anderen RTE-Solver sind in [Mayer et al., 2011, Seite 22 ff.] beschrieben.

Für jede berechnete Wellenlänge wird in der Output-Datei ein Block erstellt. Wurde der Input-Parameter `spline` gesetzt, gibt jeder Block den Wert über den integrierten Wellenlängenbereich an. Anderenfalls wird über den kürzesten Wellenlängenbereich integriert den die Auflösung des extraterrestrischen Spektrums zulässt. Dieses Spektrum ist in einer Datenbank hinterlegt und kann mit dem Input-Parameter `solar_file` gewählt werden (Listing 1, Zeile 2).

Wird *uvspec* mit allen in Tabelle 2 angegebenen geometrischen Input-Parametern ausgeführt, hat ein Block in der Output-Datei das in Tabelle 3 dargestellte Format. Die Bedeutungen der darin verwendeten Parameter wird in Tabelle 4 angegeben.

lambda	edir	edn	eup	uavgdir	uavgdn	uavgup
		phi				
umu	u0u	uu				

Tabelle 3: Format eines Blockes in der *uvspec* Output-Datei

PARAMETER	BEDEUTUNG	EINHEIT
lambda	Wellenlänge	nm
edir	Direkte Bestrahlungsstärke (<i>direct beam irradiance</i>)	$mW m^{-2} nm^{-1}$
edn	Diffuse Bestrahlungsstärke nach unten (<i>diffuse down irradiance</i>)	$mW m^{-2} nm^{-1}$
eup	Diffuse Bestrahlungsstärke nach oben (<i>diffuse up irradiance</i>)	$mW m^{-2} nm^{-1}$
uavgdir	Beitrag des direkten Strahlenganges zur mittleren Intensität (<i>direct radiation contribution to mean intensity</i>)	$mW m^{-2} nm^{-1}$
uavgdn	Beitrag des diffusen Strahlenganges nach unten zur mittleren Intensität (<i>diffuse downward radiation contribution to mean intensity</i>)	$mW m^{-2} nm^{-1}$
uavgup	Beitrag des diffusen Strahlenganges nach oben zur mittleren Intensität (<i>diffuse upward radiation contribution to mean intensity</i>)	$mW m^{-2} nm^{-1}$
umu	Kosinus vom Zenitwinkel des Sensors	
u0u	über das Azimut gemittelte Intensität	$mW m^{-2} nm^{-1} sr^{-1}$
uu	Strahldichte (<i>radiance</i>) am Sensor	$mW m^{-2} nm^{-1} sr^{-1}$

Tabelle 4: Parameter der *uvspec* Output-Datei

Beispielhaft findet man im folgendem Listing 2 den von *uvspec* generierten Output, welchen man durch die in Listing 1 angeführte Input-Datei erhält. Um eine bessere Übersicht zu gewährleisten wurden nur die drei Wellenlängen 550.0, 550.5 und 551.0 nm berechnet.

Listing 2: uvspec Output-Datei

```
1 550.000 1.753647e+03 -6.440003e-12 9.726609e+02 1.444736e+02
   -6.520612e-13 1.444556e+02
2
   90.000
3 0.9659 3.212738647e+02 3.218305054e+02
4
5 550.500 1.747524e+03 -1.273301e-11 9.690656e+02 1.439691e+02
   -1.352709e-12 1.439076e+02
6
   90.000
7 0.9659 3.200940552e+02 3.206469727e+02
8
9 551.000 1.739299e+03 -4.958180e-12 9.642057e+02 1.432915e+02
   -5.804360e-13 1.431641e+02
10
   90.000
11 0.9659 3.185056458e+02 3.190541077e+02
```

3.1.5 Interpretation der *uvspec* Output-Datei

Ehe jedoch zur Interpretation der Ausgangsparameter übergegangen werden kann, muss zunächst noch einmal näher auf den Input-Parameter *zout* eingegangen werden, der die Höhe des Sensors festlegt. Je nachdem, welchen Wert *zout* annimmt, können drei verschiedene geometrische Szenarien eintreten. Diese haben einen unmittelbaren Einfluss auf die Strahlengänge der resultierenden Ausgabeparameter. Die im Folgenden beschriebenen Erkenntnisse werden nochmals in der Abbildung 13 veranschaulicht, welche die Abhängigkeit der Strahlengänge von der Sensorhöhe *zout* visualisiert. Zugleich werden in dieser Abbildung die von der *uvspec* Output-Datei generierten Parameter dargestellt, welche ebenfalls in diesem Abschnitt beschrieben werden.

- zout=TOC** In diesem Fall liegt der Sensor am Boden (Abbildung 13a). Dieses Szenario gilt für Teleskope auf der Erde. Dabei ist darauf zu achten, dass der Kosinus des Zenitwinkels *umu* einen negativen Wert annimmt, damit die Blickrichtung des Teleskops nach oben gerichtet ist. Nachteil dieses niedrigen Standortes ist es, dass sämtliche in der Atmosphäre auftretende Reflexions- und Absorptionsvorgänge die Beobachtung beeinflussen.
- TOC>zout>TOA** In diesem Fall befindet sich der Sensor zwischen dem Boden und dem oberem Ende der Atmosphäre (Abbildung 13b). Ein solches Szenario stellt beispielsweise den Einsatz eines Teleskops an Bord eines Flugzeuges dar. Zum Beispiel entwickelten die NASA und das DLR gemeinsam das fliegende Observatorium SOFIA (Stratosphären-Observatorium für Infrarot Astronomie) mit einem 2,7 Meter-Teleskop an Bord einer BOING 747 SP [DLR, 2010].
- zout=TOA** In diesem Fall befindet sich der Sensor genau oberhalb der Atmosphäre (Abbildung 13c). Dieses Szenario spielt im Rahmen der vorliegenden Arbeit die größte Rolle. Im Hinblick auf die detektierbare Radiance ist die Höhe TOA sogar gleichbedeutend mit jeder anderen Höhe, die ein Erdbeobachtungssatellit annehmen könnte, da das Modell oberhalb der Atmosphäre keine Extinktions-Effekte berücksichtigt.

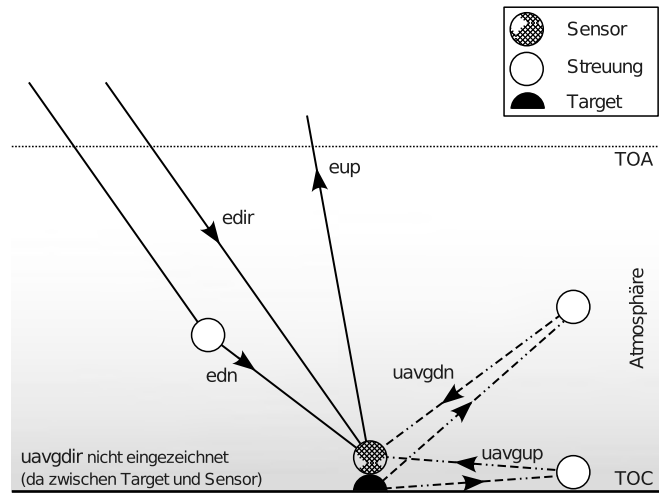
An dieser Stelle wird auch der Zusammenhang zwischen Radiance und Irradiance deutlich. Die Irradiance ist die gesamte Leistung der eingehenden elektromagnetischen Energie am Ort des Sensors. Die in

Tabelle 4 aufgeführten Ausgabeparameter beschreiben dabei die jeweiligen Anteile der einzelnen Strahlengänge. Hingegen ist die Radiance uu am Satelliten die pro Steradian und Wellenlängenintervall aufgenommene Strahlungsleistung. Sie ist unter anderem stark abhängig von der Blickrichtung, welche mit den Ausgabeparametern ϕ und umu definiert wurde. Im Folgendem sollen die in der *uvspec* Output-Datei enthaltenen Irradiance-Werte den Strahlengängen zugeordnet werden, welche letztlich zur der am Sensor detektierten Radiance uu beitragen:

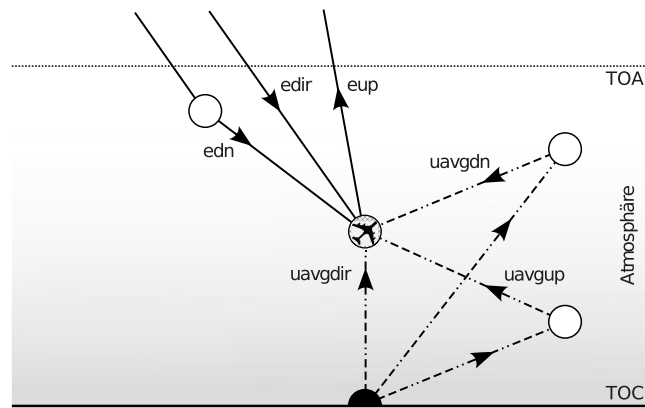
- edir** ist die Irradiance des Strahlengangs, welche direkt von der Sonne aus auf zum Sensor gelangt.
- edn** ist diejenige Irradiance der Strahlengänge, welche von in der Atmosphäre vorhanden Aerosolen oder Wolken zwar reflektiert werden, aber dennoch den Sensor erreichen.
- eup** ist die aufwärts gerichtete diffuse Strahlung in der Höhe $zout$, welche aus den Strahlengängen von $edir$, edn , $uavgdir$, $uavgdn$ und $uavgup$ resultiert. Diese Größe spielt für die am Sensor berechnete Radiance und Irradiance keine Rolle.
- uavgdir** ist der Anteil von der Summe aus $edir(zout=0)$ und $edn(zout=0)$, welcher vom Target direkt zum Sensor reflektiert wird. Diese Größe ist proportional zum Albedo des Targets.
- uavgup** bezieht sich auf den Strahlengang, der zwar vom Target reflektiert wird, aber den Sensor erst über weitere Reflexionen in aufwärtiger Richtung erreicht.
- uavgnd** ist dem Strahlengang von $uavgup$ sehr ähnlich. Jedoch wird $uavgnd$ nicht in aufwärtiger Richtung in den Strahlengang zum Sensor reflektiert, sondern gelangt erst durch weitere Reflexionen in der Atmosphäre in abwärtiger Richtung dorthin.

Um die Abhängigkeiten zwischen den Output-Parametern zu veranschaulichen, wird dieses Kapitel mit drei Beispielen abgerundet. Im ersten Fall soll die Auswirkung eines bewölkten Himmels auf die Erdbeobachtung deutlich gemacht werden. Im zweiten Fall soll gezeigt werden, welche Konsequenzen die Luftverschmutzung über Städten hat. Und abschließend soll ein praxisnaher Anwendungsfall die Wellenlängenabhängigkeit der am Satelliten gemessenen Radiance zeigen.

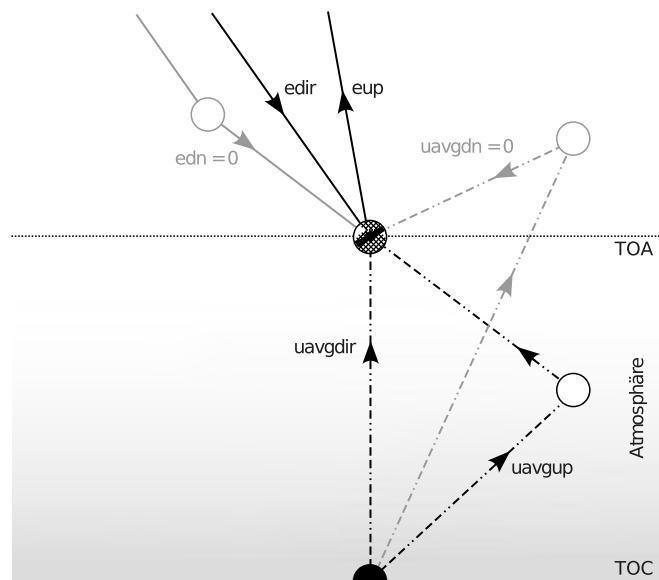
In den drei folgenden Beispielen muss jedoch stets bedacht werden, dass für die Simulation der Wolken und Aerosole ein stark vereinfachtes Modell gewählt wurde. Die präsentierten Ergebnisse stellen folglich nur eine Näherung dar. Detailliertere Modelle werden in [Shettle, 1989] beschrieben und können bei Bedarf in *libRadtran* implementiert werden.



(a) Sensor Top of Canopy ($z_{out} = TOC$)



(b) Sensor in der Atmosphäre ($TOC < z_{out} < TOA$)



(c) Sensor Top of Atmosphere ($z_{out} = TOA$)

Figure 13: zu den *uvspec* Output-Parametern gehörende Strahlengänge, mit Abhängigkeit von z_{out}

3.1.6 Beispiel: bewölkter Himmel

Dieses Beispiel hat zum Ziel, die Abhängigkeit der *uvspec* Ausgabeparameter von dem geometrischen Parameter *zout* aufzuzeigen. Es soll also der Einfluss der Höhe auf die Irradiance dargelegt werden.

Des Weiteren soll der Einfluss von Wasserwolken bei der Erdbeobachtung im sichtbaren Bereich des Spektrums erörtert werden.

Als Ausgangspunkt dient das einfache Szenario, wie es bereits durch die Input-Datei auf Seite 26 beschrieben wurde. Damit es die Anforderungen dieses Beispiels erfüllt, muss die Datei erweitert werden und hat schließlich folgende Form:

Listing 3: Input-Datei zur Beschreibung von Wasserwolken

```

1 # SET GENERAL CONDITION:
2 atmosphere_file ../data/atmmod/afglus.dat
3                                     # location of atmospheric profile file
4 solar_file ../data/solar_flux/atlas_plus_modtran
5                                     # location of extraterrestrial spectrum
6 ozone_column 300.                   # scale ozone column to 300.0 DU
7 day_of_year 170                     # correct for Earth-Sun distance
8 albedo 0.6                          # surface albedo
9 rte_solver disort                   # choose RTE solver
10 deltam on                          # delta-M scaling on
11 nstr 6                              # number of streams
12 wavelength 550                     # wavelength range [nm]
13
14 # SET GEOMETRY:
15 zout 0 1 1.5 2 2.5 3 3.5 4 4.5 5 5.5 6 7      ...      119 120 TOA
16                                     # loop: sensor altitude above ground
17 phi 90.0                            # sensor azimuth
18 umu 0.9659258                      # cosine of viewing zenith angle
19 phi0 210.0                          # solar azimuth
20 sza 15.0                            # solar zenith angle
21
22 # SET WATERCLOUDS:
23 wc_set_tau 15                       # set total watercloud optical depth
24 wc_file ../WC_SIMPLE.DAT # location of watercloud file

```

Damit das Beispiel anschaulich bleibt und unnötige Komplexität vermieden wird, soll stellvertretend für das sichtbare Spektrum, lediglich eine Wellenlänge⁴ von 550 nm untersucht werden, wie durch den Eingabeparameter in Listing 3, Zeile 12 zu entnehmen ist.

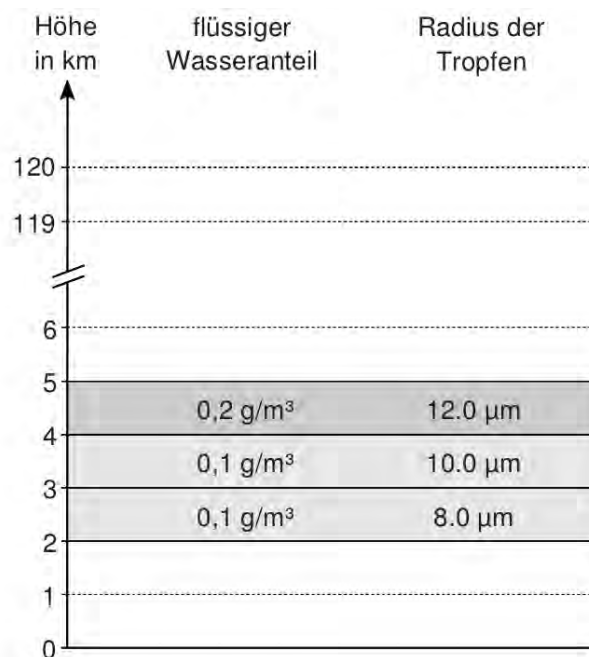
⁴ Simulationen über den ganzen Bereich des sichtbaren Spektrums (etwa 380 bis 780 nm) haben ergeben, dass diese Vereinfachung den zu zeigenden Sachverhalt näherungsweise wiedergeben kann.

Zudem soll das Target am Boden durch Wolken verdeckt sein. Damit *wvspec* die Wasserwolken berücksichtigen kann, werden sie am Ende der Datei definiert. Dabei gibt `wc_set_tau` (Zeile 23) die optische Dicke der Wasserwolke an. Sie ist ein Maß dafür, wie gut die Sonnenstrahlung eine Materieschicht (hier die Wolke) durchqueren kann [Hecht, 2005]. Als nächstes werden über die Datei `WC_SIMPLE.DAT` die Ausmaße und Bestandteile der Wolke definiert (Zeile 24). Der beschreibende Code für eine Wolke kann zum Beispiel folgendermaßen aussehen:

Listing 4: Datei zur Beschreibung einer Wasserwolke

```
# z      LWC   R_eff
# (km) (g/m3) (um)
#
5.000   0     0
4.000   0.2  12.0
3.000   0.1  10.00
2.000   0.1   8.0
```

Diese Datei hat drei Spalten. Die erste gibt die Höhe z in km an, die zweite den flüssigen Wasseranteil LWC (Liquid Water Content) in g/m^3 und die dritte den idealisierten Tropfenradius in μm . Wie in Abbildung 14 dargestellt, wird die Wolke schichtweise beschrieben. In diesem Beispiel erstreckt sie sich von 2 bis 5 km über dem Meeresspiegel. In der Höhe zwischen 2 und 4 km ist der flüssige Wasseranteil $0,1 g/m^3$, während er in der Höhe zwischen 4 und 5 km $0,2 g/m^3$ beträgt.

Abbildung 14: eine durch `WC_SIMPLE.DAT` beschriebene Wasserwolke

Um die Abhängigkeit von der Sensorhöhe z_{out} zu ermitteln, muss *uvspec* mit mehreren Werten für z_{out} ausgeführt werden. Dazu muss in der Input-Datei hinter dem Eingabeparameter z_{out} einfach eine Liste sämtlicher sinnvoller Höhen folgen wie in Listing 3, Zeile 15 dargestellt. Die bei dieser Simulation verwendete U.S. Standardatmosphäre erstreckt sich bis zu einer Höhe von 120 km , weswegen ein größerer Wert keinen Sinn macht. Als Auflösung von z_{out} wurde im Bereich der Wolken $0,5\text{ km}$ und sonst 1 km oder weniger gewählt. Es gilt zu beachten, dass *uvspec* für einen jeden zusätzlichen Wert von z_{out} , ein weiteres mal ausgeführt wird. Um die Rechenzeit in Grenzen zu halten, sollten folglich nicht mehr Werte als nötig angegeben werden.

Nachdem *uvspec* mit dieser Input-Datei ausgeführt wurde, erhält man eine Output-Datei mit aus Tabelle 3 bekannten Blöcken. Diesmal repräsentiert jedoch ein Block die Ergebnisse für einen bestimmten Wert von z_{out} . Wie man die benötigten Werte aus der Output-Datei extrahiert und graphisch darstellt, wird in Abschnitt 4.3 beschrieben. Die drei Graphen in Abbildung 15 zeigen jeweils den Verlauf der Ausgabeparameter ed_{ir} , ed_n und u_u über die Sensorhöhe z_{out} .

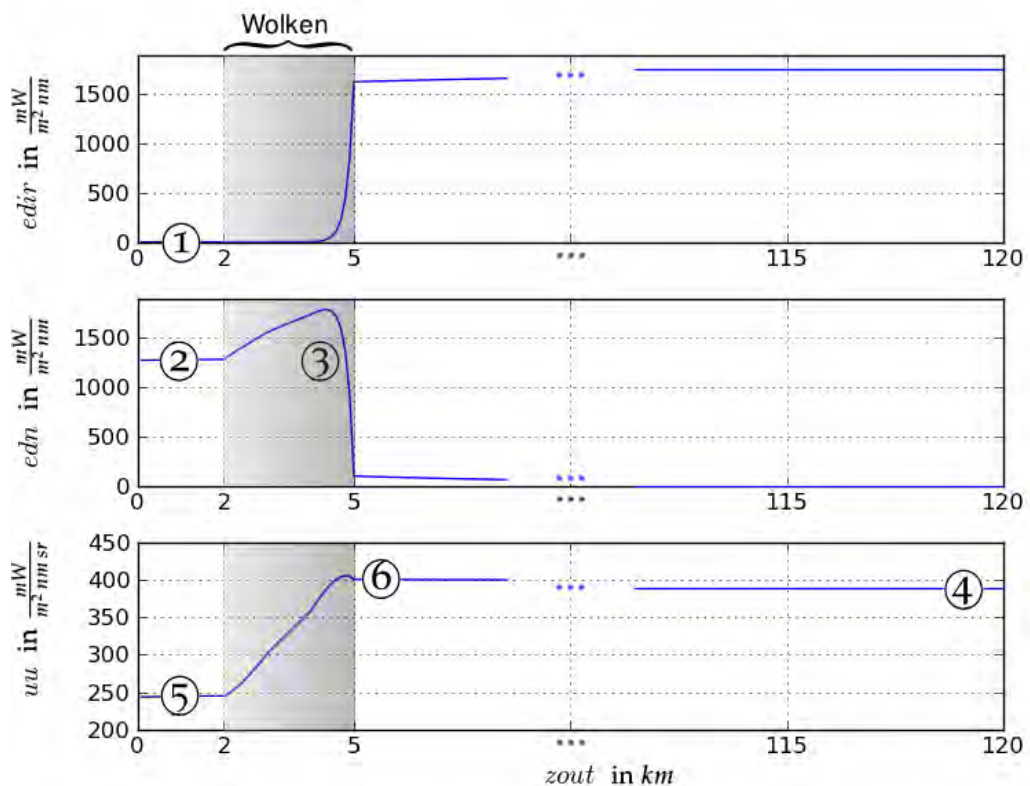


Abbildung 15: die Ausgabeparameter ed_{ir} , ed_n und u_u in Abhängigkeit der Sensorhöhe z_{out}

Betrachtet man die Ergebnisse, so stellt man fest, dass ed_{ir} unterhalb der Wolkenschicht einen sehr niedrigen Wert annimmt ①. Der Grund hierfür ist, dass die Wolken den direkten Strahlengang zwischen der Sonne und dem Boden unterbinden. Stattdessen führt der Wasseranteil in den Wolken zu Reflexionen, sodass letztlich nur diffuses Licht den Boden erreichen kann. Dies resultiert in einem hohen Wert für ed_n im bodennahem Bereich ②. Innerhalb der Wolken ist sogar eine noch intensivere abwärtige Streuung zu erkennen ③.

Dennoch ist die Radiance, welche den Satelliten TOA in 120 km Höhe erreicht, überraschend hoch ④. Das Target am Boden kann jedoch als Quelle ausgeschlossen werden, da dort nur eine sehr niedrige Radiance gemessen werden kann ⑤. Stattdessen sind es die Wolken selbst, welche mit ihrem hohen Albedo einen Großteil der Sonnenstrahlung zum Sensor hin reflektieren ⑥.

Abschließend lässt sich festhalten, dass die vom bewölktem Target reflektierte Strahlung nur einen geringen Einfluss auf die Radiance am Sensor hat. Wasserwolken stellen somit bei der Erdbeobachtung im sichtbaren Bereich des Spektrums einen problematischen Faktor dar⁵.

3.1.7 Beispiel: Smog

Wie auch im vorangegangenen Beispiel soll die Abhängigkeit zwischen Ausgabeparametern ed_{ir} , ed_n , uu und dem geometrischen Parameter $zout$ aufgezeigt werden. Zudem soll der Einfluss von Aerosolen in der Atmosphäre auf die Erdbeobachtung im sichtbaren Bereich des Lichtspektrums, festgestellt werden.

Damit die Größenordnungen der auftretenden Effekte besser mit dem vorangegangenen Beispiel mit bewölktem Himmel verglichen werden können, sei wieder dasselbe Szenario gegeben. Die allgemeinen Randbedingungen und die Geometrie in der Input-Datei bleiben unverändert erhalten, während die Parameter der Wolkeneigenschaften mit denen der Aerosole ersetzt werden. Denn das Target soll nun inmitten eines mit Smog belastetem urbanem Gebiet liegen. Um dieses Szenario umzusetzen, muss die Input-Datei folgende Form aufweisen:

⁵ lediglich aktive Radarsysteme im Mikrowellenlängenbereich können aufgrund ihrer Wellencharakteristik Wolken durchdringen und erlauben so eine wetterunabhängige Fernerkundung [Alberz, 2001]

Listing 5: Input-Datei zur Beschreibung von Aerosolen

```

1 # SET GENERAL CONDITION:
2 atmosphere_file ../data/atmmod/afglus.dat
3                                     # location of atmospheric profile file
4 solar_file ../data/solar_flux/atlas_plus_modtran
5                                     # location of extraterrestrial spectrum
6 ozone_column 300.                   # scale ozone column to 300.0 DU
7 day_of_year 170                       # correct for Earth-Sun distance
8 albedo 0.6                             # surface albedo
9 rte_solver disort                       # choose RTE solver
10 deltam on                             # delta-M scaling on
11 nstr 6                                 # number of streams
12 wavelength 550                        # wavelength range [nm]
13
14 # SET GEOMETRY:
15 zout 0 1 1.5 2 2.5 3 3.5 4 4.5 5 5.5 6 7 ... 119 120 TOA
16                                     # loop: sensor altitude above ground
17 phi 90.0                               # sensor azimuth
18 umu 0.9659258                         # cosine of viewing zenith angle
19 phi0 210.0                             # solar azimuth
20 sza 15.0                               # solar zenith angle
21
22 # SET AEROSOL PROPERTIES:
23 aerosol_vulcan 1                       # aerosol type above 2 km
24                                     # 1: no vulcanic aerosols
25 aerosol_haze 5                         # aerosol type below 2 km
26                                     # 5: urban type aerosols
27 aerosol_season 1                       # 1: summer season
28 aerosol_visibility 15.0                # horizontal visibility in km
29 aerosol_angstrom 1.1 0.2              # scale aerosol optical depth
30                                     # using Angstrom alpha and beta
31                                     # coefficients
32 aerosol_scale_ssa 0.85                 # scale the single scattering albedo
33                                     # for all wavelengths
34 aerosol_set_gg 0.70                    # set the asymmetry factor
35 aerosol_tau_file ../examples/AERO_TAU.DAT
36                                     # file with optical depth profile

```

Die Bedeutung der Eingabeparameter ist den angehängten Kommentaren oder [Mayer et al., 2011, Kapitel 6] zu entnehmen. Die Konzentration der Aerosolpartikel in der Atmosphäre wurde so hoch gesetzt, dass sie in Form von Smog wahrgenommen werden können. Wird *uvspec* mit der in Listing 5 beschriebenen Input-Datei ausgeführt und die daraus generierte Output-Datei visualisiert, erhält man die in Abbildung 16 dargestellten Graphen. Zusätzlich zum vorherigen Beispiel mit

den Wolken, werden nun auch die Parameter $uavgdir$ und $uavgdn$ in die Betrachtung mit einbezogen.

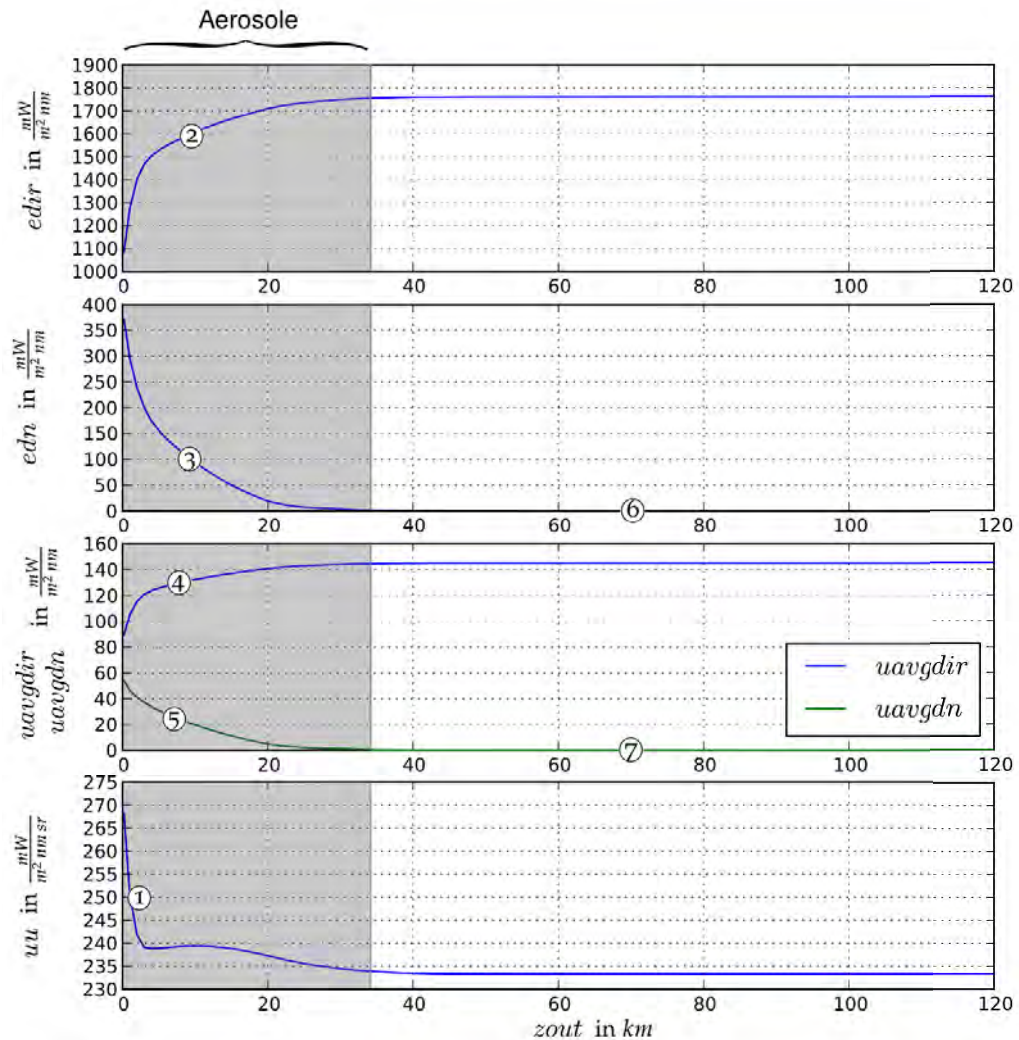


Abbildung 16: die Ausgabeparameter $edir$, edn und uu in Abhängigkeit der Sensorhöhe $zout$

Bei der Interpretation lässt sich schnell erkennen, dass sich der Einfluss der Aerosole über eine sehr viel größere Höhendifferenz hinweg erstreckt, als bei Wasserwolken. Durch das deutliche nicht-lineare Verhalten der Graphen lässt sich ausmachen, dass Aerosole vom Boden bis in etwa 35 km Höhe in unterschiedlicher Konzentration auftreten. Andere Ursachen für dieses Verhalten können ausgeschlossen werden, da mit der Input-Datei einzig die Aerosole definiert wurden. Aus der Radiance-Änderung $\frac{d uu}{d zout}$ lässt sich auch bestimmen wie stark der Einfluss der Aerosole ist. Vor allem im Bereich unterhalb von 2 km, wo

laut Listing 5, Zeile 25 die urbanen Aerosole auftreten, ist die Wirkung auf die Radiance u signifikant ①. Mit zunehmender Höhe wird die Konzentration der Aerosole schwächer, womit auch ihr Einfluss auf die Strahldichten nachlässt.

Stellt man man die Verläufe der direkten Sonnenstrahlung $edir$ ② und der diffusen abwärtigen Strahlung edn ③ gegenüber, so lässt sich erkennen, dass der diffuse Anteil edn im selben Verhältnis zunimmt, wie der direkte Anteil $edir$ abnimmt. Das selbe Verhalten ist auch zwischen den Strahlengängen $uavgdir$ ④ und $uavgdn$ ⑤ zu beobachten. Dieses Phänomen ist mit dem eingangs erwähnten Reflexionsverhalten des Aerosols zu erklären. Ein Aerosol ist genau genommen eine Mischung aus Schwebeteilchen und einem Gas [Gassmann, 1994]. Trifft die direkte Sonnenstrahlung auf ein solches Aerosolteilchen, so wird sie daran reflektiert oder auch teilweise absorbiert. Der Betrag

$$edir(TOA) - [edir(zout) + edn(zout)] \quad (3.1)$$

ist folglich jener Teil der Strahlung, welcher auf dem Weg zum Target von Gasen und Aerosolen absorbiert wurde, oder in eine andere Richtung als zum Target hin abgelenkt wurde.

Des Weiteren kann am Verlauf von edn und $uavgdn$ die in Abbildung 13c aufgestellte Behauptung bewiesen werden, dass diese beiden Strahlengänge bei TOA gegen Null gehen ⑥ ⑦. Begründet werden kann das damit, weil oberhalb der TOA keine Teilchen mehr vorhanden sind, welche das Licht zurück Richtung Erde reflektieren könnten. Die diffuse abwärtige Strahlung $uavgdn$ ist oberhalb der TOA folglich Null.

Zusammenfassend kann festgehalten werden, dass die diffuse Streuwirkung der Aerosole eine kontrastmindernde Wirkung hat. Zusätzlich lässt sich am Verlauf der Radiance u nachverfolgen, wie Aerosole die Strahlung vom Target zum Sensor dämpfen und somit abschwächen.

3.1.8 Beispiel: Wellenlängenabhängigkeit

Während die ersten beiden Beispiele dazu dienen, ein besseres Verständnis der Ausgabeparameter zu vermitteln, ist dieses Beispiel sehr praxisnah. Bisher wurde stets darauf hingewiesen, dass die detektierte Radiance nicht nur von der Sensorhöhe $zout$, sondern auch von der Wellenlänge abhängt. Eben diese Abhängigkeit soll nun mit *libRadtran* untersucht werden. Auf welche Weise die Ermittlung der Wellenlängenabhängigkeit bewerkstelligt wird, ist unter anderem deswegen von Bedeutung, da im Kapitel 5 ein Vergleich von *libRadtran* mit *MODTRAN* erfolgt.

Prinzipiell wird die Erde mit der extraterrestischen Sonnenstrahlung E_0 bestrahlt. Jedoch erreicht, aufgrund der in Abschnitt 2.1 dargelegten

Extinktions-Effekte, nur ein bestimmter Anteil dieser Strahlung den Boden. Konkret soll in diesem Beispiel der Transmissionsgrad der Atmosphäre und die Irradiance an der Erdoberfläche ermittelt werden. Abschließend soll die Irradiance am Erdboden mit der Irradiance oberhalb der Atmosphäre, also der extraterrestischen Sonnenstrahlung, verglichen werden.

Die extraterrestische Sonnenstrahlung muss nicht ermittelt werden, da sie einen Teil der Datenbank von *libRadtran* darstellt. Wie in den bisherigen Input-Dateien zu erkennen ist, muss stets der Eingabeparameter `solar_file <Datei>` vorhanden sein. Bei der daraufhin angegebenen Datei handelt es sich um die Beschreibung der extraterrestischen Sonnenstrahlung. Diese Textdatei hat zwei Spalten, wobei erstere die Wellenlänge in *nm* angibt und zweitere die dazugehörige extraterrestische Sonnenstrahlung in $\frac{mW}{m^2nm}$. Unter dem Installations-Verzeichnis von *libRadtran*, im Pfad `/data/solar_flux/` befinden sich mehrere Dateien zur Beschreibung der Sonnenstrahlung. In erster Linie unterscheiden sie sich in der Auflösung der beschriebenen Wellenlänge und haben so einen starken Einfluss auf die Rechendauer und die Auflösung der Ergebnisse. Für dieses Beispiel wurde eine Auflösung von `1 nm` gewählt, wie sie durch die Datei `kurudz_1.0nm.dat` zur Verfügung gestellt wird. Abbildung 17 zeigt den unter anderem den Plot der durch diese Datei definierten extraterrestische Sonnenstrahlung über die Wellenlänge.

Um die Irradiance am Boden zu ermitteln, wird wieder auf das in Listing 1 vorgestellte Szenario zurückgegriffen und erweitert. Damit die gewünschte Ausgabe berechnet werden kann muss die Input-Datei folgende Form annehmen:

Listing 6: Input-Datei zur Berechnung der Irradiance

```

1 # SET GENERAL CONDITION:
2 atmosphere_file /home/o222840/Michael/libRadtran-1.6-beta/data/
   atmod/afglms.dat # location of atmospheric profile file
3 solar_file /home/o222840/Michael/libRadtran-1.6-beta/data/solar_flux
   /kurudz_1.0nm.dat # location of extraterrestrial spectrum
4 albedo 0.4 # surface albedo
5 rte_solver disort # choose RTE solver
6 wavelength 250 3000 # wavelength range [nm]
7 verbose # write diagnostic output
8 aerosol_default # set aerosol properties
9
10 # SET GEOMETRY:
11 zout TOA # altitude of sensor
12 phi 0 # sensor azimuth
13 umu -1 # cosine of viewing zenith angle

```



```

14 | phi0 0           # solar azimuth
15 | sza 0           # solar zenith angle
16 |
17 | correlated_k lowtran # select LOWTRAN molecular absorption

```

Damit die molekulare Absorption zu berücksichtigt wird, bedarf es den Befehl “correlated_k lowtran” in Zeile 20. Wie in Abschnitt 2.3 beschrieben, kann die molekulare Absorption entweder über Bänder oder *Line-by-Line* für jede Spektrallinie berechnet werden. Da in diesem Beispiel ein breites Spektrum von 250 nm bis 3000 nm betrachtet wird, kommt nur ein Bändermodell in Frage. Andernfalls würde bei der Berücksichtigung sämtlicher Spektrallinien in diesem Spektrum, die Berechnungsdauer etwa eine Woche betragen. Infolgedessen wurde für die pseudospektrale Berechnung der molekularen Absorption, auf das Bändermodell *LOWTRAN/SBDART* von [Ricchiazzi et al., 1998] zurückgegriffen. Wie die Berücksichtigung sämtlicher Spektrallinien erfolgt, ist dem Kapitel 7 zu entnehmen. Dort wird beschrieben wie *libRadtran* auf die molekulare Absorptions-Datenbank *HITRAN* zugreifen kann und sämtliche Spektralparameter für die Simulation zugänglich macht.

Die Berechnung des atmosphärischen Transmissionsgrads τ erfolgt parallel zur Berechnung der Irradiance. Dabei kann dieselbe Input-Datei wie in Listing 6 verwendet werden, wobei jedoch der Eingabeparameter *transmittance* in einer zusätzlichen Zeile angehängt werden muss. Dadurch werden in der daraus resultierenden Output-Datei anstelle der Irradiance-Werte, die jeweiligen Transmissionsgrade angegeben. Abbildung 17 stellt die Ergebnisse der beiden Output-Dateien graphisch dar.

Anhand der Abbildung 17 kann der zu zeigende Zusammenhang

$$E_{TOC} = E_0 \cdot \tau$$

zwischen der extraterrestrischen Sonnenstrahlung E_0 und der Irradiance E_{TOC} an der Erdoberfläche nachgewiesen werden. Multipliziert man den atmosphärischen Transmissionsgrad τ mit der extraterrestrischen Irradiance E_0 , so erhält man die am Erdboden einfallenden Irradiance E_{TOC} . Da in diesem Beispiel ein klarer Himmel definiert ist, wird der Transmissionsgrad im Wesentlichen durch das molekulare Absorptionsverhalten bestimmt. Abbildung 18 zeigt, welche Moleküle für die jeweiligen Absorptionsbänder verantwortlich sind. Dabei wird auch gleichzeitig das Verhältnis zwischen extraterrestrischer Sonnenstrahlung und der Strahlung an der Erdoberfläche auf Meeresspiegel veranschaulicht.

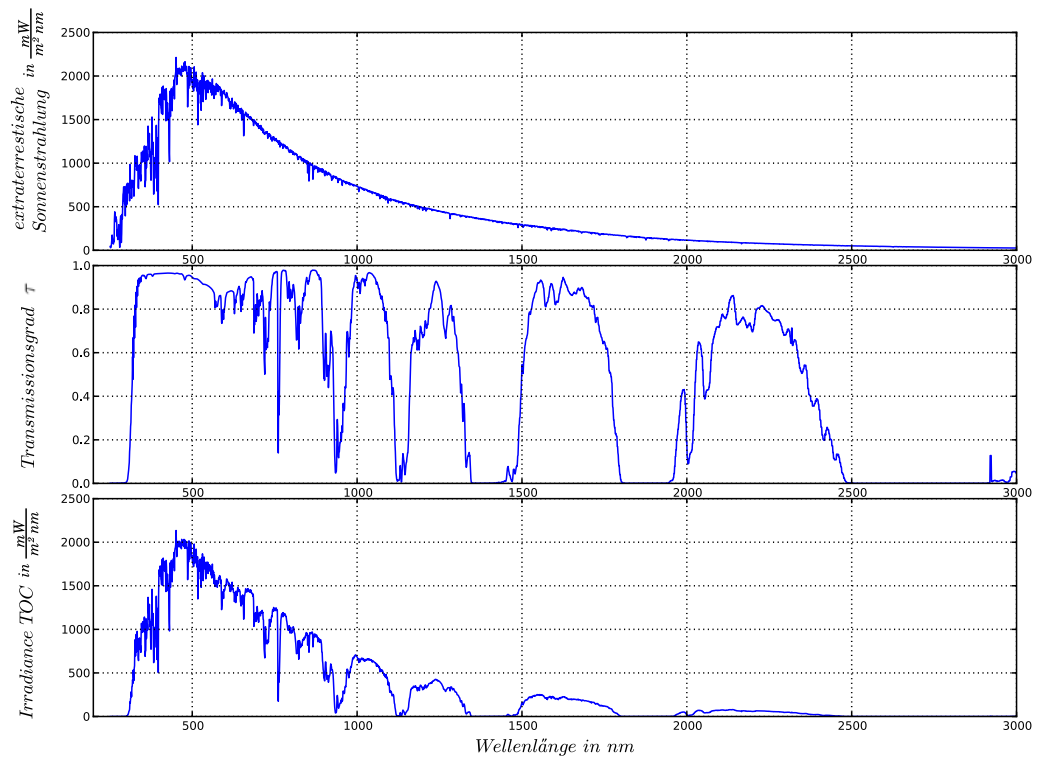


Abbildung 17: Gegenüberstellung der Irradiance TOA, dem Transmissionsgrad der Atmosphäre und der Irradiance TOC

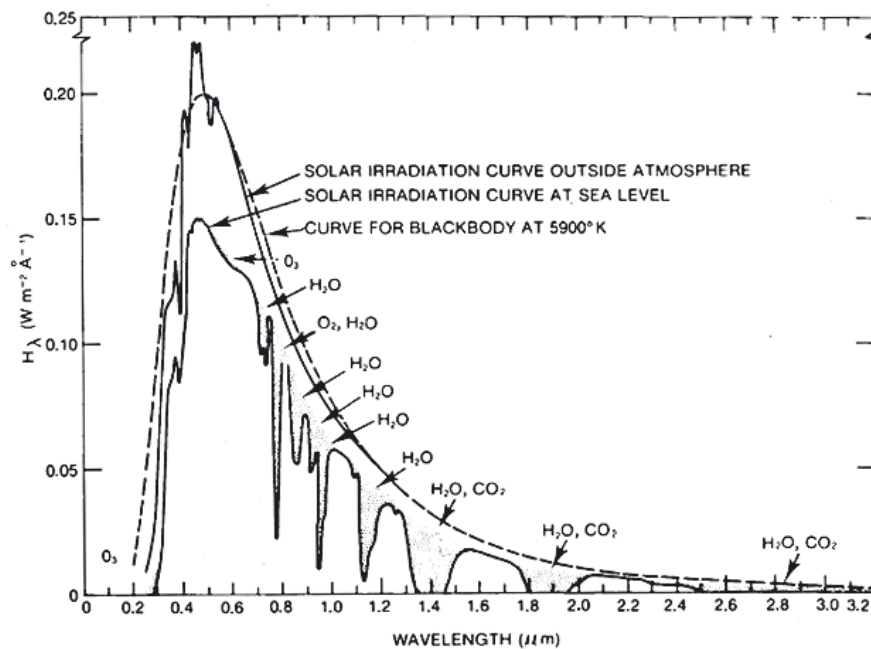


Abbildung 18: Irradiance TOA und TOC, mit Lage und Ursache der atmosphärischen Absorption [Short and Blair, 1986]

3.2 MODTRAN (KOMMERZIELL)

Im Gegensatz zu *libRadtran* handelt es sich bei *MODTRAN* (MODerate TRANsition Code) um keine frei zugängliche Software. *MODTRAN* wurde in den späten 80er-Jahren von der U.S. AIR FORCE (USAF) in Zusammenarbeit mit SPECTRAL SCIENCES, INC. entwickelt. Das Modell deckt einen Bereich von 200 nm bis 10 mm ab [Berk et al., 2003], was auch in Einklang mit unseren Anforderungen steht. Der größte Vorteil des kommerziellen *MODTRAN* liegt darin, dass es seit knapp drei Jahrzehnten das Standardmodell der USAF darstellt und daher eine stete Pflege und Weiterentwicklung erfahren hat. Allerdings trifft dieser stetige Fortschritt nicht auf das User-Interface zu. Vergleichbar mit *libRadtran* muss auch bei *MODTRAN* eine Input-Datei geschrieben werden. Jedoch verlangen diese Dateien, bei *MODTRAN* als *Tapes* bezeichnet, nach einer strikten Formatvorlage. Zudem entstehen je nach gesetztem Parameter Abhängigkeiten zwischen anderen Tapes. Dieses Relikt ist historisch bedingt und lässt sich auf Zeiten zurückführen, als die Eingabeparameter noch mittels einer Lochkarte dem Programm mitgeteilt werden mussten. Um dieses Manko zu beseitigen wurde von REMOTE SENSING APPLICATIONS SCHLÄPFER die grafische Benutzeroberfläche *MODO* in der Programmiersprache *IDL* entwickelt. Zwar unterliegt der ONTAR CORPORATION der exklusive Vertrieb von *MODTRAN* [Ontar Corporation, 2012], jedoch hat diese mit REMOTE SENSING APPLICATIONS SCHLÄPFER eine Lizenzvereinbarung getroffen. Infolge dieser Vereinbarung kann *MODTRAN* zusammen mit *MODO* über REMOTE SENSING APPLICATIONS SCHLÄPFER bezogen werden [Schläpfer and Odermatt, 2006].

Im Folgendem soll die rudimentäre Handhabung geschildert werden, welche für den Vergleich mit *libRadtran* erforderlich ist. Dabei ist zu betonen, dass die Bedienung von *MODTRAN* und dessen Benutzeroberfläche *MODO* kein zentrales Thema dieser Arbeit darstellen. Für einen intensiveren Zugang zu diesen Programmen sei auf deren Benutzerhandbücher [Berk et al., 2003] und [Schläpfer and Odermatt, 2006] verwiesen.

3.2.1 MODTRAN mit der Benutzeroberfläche MODO

Das *tape5* hat für *MODTRAN* die selbe Funktion wie die *uvspec* Input-Datei für *libRadtran*. In ihm wird sowohl die Atmosphäre, als auch die Beobachtungsgeometrie beschrieben. Das *tape5* selbst besteht aus sechs oder mehreren *CARDS*. Diese sind als separate Zeilen des *tape5* zu verstehen und weisen starke gegenseitige Abhängigkeiten auf. Zudem unterliegen sie einer strengen ASCII Formatierung, welche bei der Eingabe strikt eingehalten werden muss. Wie bereits erwähnt wird die Erstellung

eines *tape5* durch die grafische Benutzeroberfläche *MODO* wesentlich erleichtert. Wie der Abbildung 19 im nächsten Kapitel entnommen werden kann, erfolgt die Parametrisierung der *CARDs* über vorgefertigte Eingabefelder und der Wahl aus vorgegebenen Optionen. Zum Ausführen der *tape5*, muss *MODTRAN* nicht separat gestartet werden, sondern wird direkt von *MODO* aufgerufen. Die Ausgabe der Ergebnisse erfolgt in das *tape7*, welches ebenfalls als Text-Datei gelesen werden kann. Die Auswertung des *tape7* kann durch dessen Visualisierung erleichtert werden, wobei *MODO* auch hier die benötigten Daten selbstständig aus der Ausgabe-Datei extrahiert und in einer Bilddatei ausgibt [Schläpfer and Odermatt, 2006].

3.2.2 Beispiel: Berechnung der Radiance TOA

Dieses Beispiel soll zeigen wie mit *MODTRAN* die Radiance Top-of-Atmosphäre L_{TOA} berechnet wird. Es wird von einer klaren Atmosphäre ohne Aerosolen und Wolken ausgegangen, wobei sich sowohl die Sonne als auch der Beobachter im Zenit befinden. Die Definition dieses Szenarios erfolgt mit der grafischen Benutzeroberfläche *MODO*. Die dabei verwendeten Eingabeparameter werden in Abbildung 19 veranschaulicht. Mit diesem Screenshot wird ebenfalls deutlich, wie übersichtlich sich die Atmosphärenbeschreibung mit *MODO* gestaltet. Zum Vergleich wird das durch *MODO* generierte und kryptisch anmutende *tape5* in Listing 7 wiedergegeben. Diese Gegenüberstellung führt vor Augen, wie sehr *MODO* die Parametrisierung des *tape5* erleichtert.

Nachdem die Berechnung des Strahlungstransfers mit *MODO* gestartet wurde, generiert *MODTRAN* anhand der *tape5* die Ausgabe-Datei *tape7*. Sie enthält die gesuchten Radiance Daten, wie sie auszugsweise in Listing 8 wiedergegeben werden. Darin wird ersichtlich, dass die Daten genauso wie bei *libRadtran*, Zeile für Zeile ausgegeben werden, wobei jede Zeile einer bestimmten Wellenlänge entspricht. Als Besonderheit erfolgt bei *MODTRAN* die Ausgabe jedoch nicht in einer Wellenlänge, sondern als Wellenzahl. Durch die Umrechnung gemäß Gleichung 4.1 kann dem *tape7* dennoch für jede Wellenlänge die zugehörige Radiance entnommen werden. Wird dieser Extraktionsvorgang durch *MODO* automatisiert und die Daten anschließend visualisiert, so erhält man den in Abbildung 20 dargestellten Plot der Radiance L_{TOA} über die Wellenlänge.

Betrachtet man den Verlauf der Radiance Top-of-Atmosphäre aus Abbildung 20 und stellt ihm den Verlauf der Irradiance Top-of-Canopy aus Abbildung 17 gegenüber, so wird ersichtlich, dass beide nahezu dieselbe Form aufweisen. Zwar unterscheiden sich deren Zahlenwerte, da einmal

die Radiance und einmal die Irradiance betrachtet werden, jedoch sind auch in diesem Beispiel die atmosphärischen Fenster deutlich erkennbar.

Abschließend sei noch bemerkt, dass es sich bei dem vorliegenden Beispiel um das selbe Szenario handelt, welches für den zentralen Vergleich zwischen *MODTRAN 4* und *libRadtran* in Kapitel 5 herangezogen wird.

Listing 7: Input-Datei tape5 zur Berechnung der Radiance TOA (gekürzt)

```

1 MM 2 2 2 1 0 0 0 0 0 0 0 0 0 293.150 0.4
2 TF 8F 5 365.00000          F F F T F          0.000
3   0 0 1 3 0 0 0.00000 0.00000 0.00000 0.00000 0.00000
4   100.000 0.000 180.000 0.000 0.000 0.000 0.000
5   1 2 180 0
6   0.000 0.000 0.000 0.000 12.000 0.000 0.000 0.000
7   400.0 2500.0 1.0 1.0 N- next -N AA
8   0

```

Listing 8: Output-Datei tape7 zur Berechnung der Radiance TOA (gekürzt)

```

1 MM 2 2 2 1 2 2 2 2 2 2 1 0 0 293.1500.4
2   0 1 1 3 0 0 0.00000 0.00000 0.00000 0.00000 0.00000
3  -99.000 -99.000 -99.000 -99
4  -99.00000 -99.00000 -99.00000
5   36MIDLATITUDE SUMMER
6  100.00000 0.00000 180.00000 100.00000 0.00000 0.00000 0
7   1 2 180 0
8   0.00000 0.00000 23.22600 359.22501 12.00000 0.00000 0.00000
9   3995 25130 1 1
10  0 0 0.000 0 0.000 0 1.000
11  FREQ TOT TRANS ... TOTAL RAD REF SOL SOL@OBS DEPTH
12  3995.00 0.00000000 ... 4.1832E-11 0.00E+00 3.10E-06 23.176
13  3996.00 0.00668214 ... 1.5763E-10 2.46E-10 3.10E-06 5.008
14  3997.00 0.02878339 ... 6.2219E-10 4.06E-09 3.10E-06 3.548
15  3998.00 0.10798939 ... 3.9559E-09 3.22E-08 3.10E-06 2.226
16  3999.00 0.05706948 ... 2.2002E-09 1.74E-08 3.10E-06 2.863
17  4000.00 0.04645408 ... 1.6982E-09 1.32E-08 3.10E-06 3.069

```

Editing Tape5 for MODTRAN 4

Modtran: Medium Midlatitude Summer Vert. Path from/to Full Radiance Multiple Scat. at H1 > Set Profile

Default Gases: To Selected Model Tracers Profile Last Profs Standard Output T-Boundary: 293.150 Albedo: 0.4

Disort Alg. No Azi 8 iter Orig Sun Resol: 5 CO2(ppm): 365.000 H2O: 03: TOA:

Files: New Kurusz 1 cm-1 Standard Orig. Resolution Modify Aero

Default No Aerosol Attenuation Default Season as Model Normal Volcan Background Std Marit. No clouds No VSA

Visibil.(km): 0.000000 Windspeed(m/s): 0.000000 24h-Windspeed: 0.000000 Rain Rate(mm/h): 0.000000 Ground Alt./km: 0.000000

Sensor Altitude H1(km): 100.000 Final Altitude H2(km): 0.000000 Sensor Zenith (degree): 180.000 Angle H1-H2: 0.000000

Path Length (km): 0.000000 Earth Radius (Def.0): 0.000000 Path short(0)/long(1): 0 Target Zenith: 0.000000

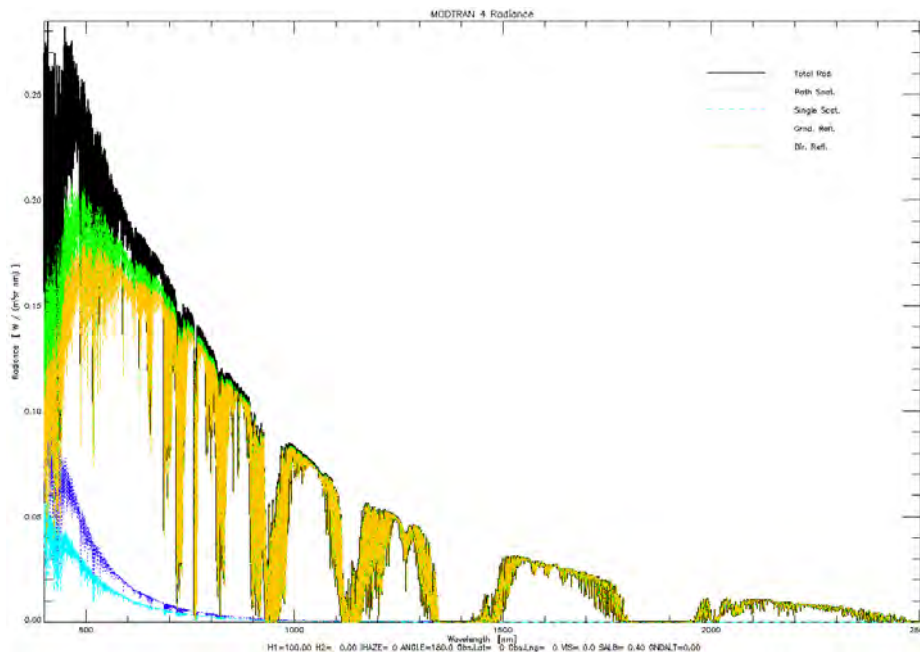
Only Observer Mie Phase Function Day-number of the year: 180 Source Sun

Observer Latitude: 0.000000 Observer Longitude: 0.000000 Source Latitude: 0.000000 Source Longitude: 0.000000

Dec.Greenwich Time: 12.0000 Path Azimuth: 0.000000 Sun/Moon Angle: 0.000000 Greenst. Asymmetry: 0.000000

Range: 400.00000 to 2500.0000 Resol: 1.00000 FWHM: 1.00000 NoPlt nm Trian nofix Flags: N AA End Modtrn

Help ? Select Save Show Save As Clone michael_test << 1 >> Kill Run Modtran DONE

Abbildung 19: Erstellung der Eingabe-Datei *tape5* mit *MODO*Abbildung 20: Visualisierung der Ausgabe-Datei *tape7* durch *MODO*

People think computers will prevent them from making mistakes.
They're wrong. With computers you make mistakes just faster.

— Adam Osborne

4

VERWENDETE ENTWICKLUNGS-SOFTWARE

Für das Ausführen des Strahlungstransfermodells *libRadtran*, der Entwicklung der Interface Software, oder dem Aufbereiten der Ergebnisse, wird zusätzliche Software benötigt. In diesem Kapitel soll sämtliche verwendete Software vorgestellt und deren Handhabung kurz beschrieben werden. Sowohl die benötigten Programme zum Vergleich der beiden Strahlungstransfermodelle, als auch die zur Implementierung des Adjacency-Effekts, sind nahezu deckungsgleich, weswegen diese nicht separat vorgestellt werden müssen. Welche Programme letztlich für die Umsetzung der jeweiligen Aufgabenstellung benötigt werden, ist den zugehörigen Kapiteln in Teil II zu entnehmen.

4.1 PROGRAMMIERSPRACHE *python*

Python ist eine stark ausgereifte Programmiersprache, welche beispielsweise auch von der NASA oder GOOGLE eingesetzt wird [Kaiser and Ernesti, 2007]. *Python*, dessen Entwurfsphilosophie die Programmlesbarkeit betont, ist eine multiparadigmatische Sprache, welche eine objektorientierte, imperative oder auch funktionale Programmierung erlaubt [Hajji, 2008]. Jedoch ist diese Einsteigerfreundlichkeit und die Vielseitigkeit auf Kosten der Geschwindigkeit erkaufte. Deswegen ist für zeitkritische Aufgaben eine betriebssystemnähere Programmiersprache, wie beispielsweise *C* oder *Fortran* empfehlenswert. Im Rahmen dieser Arbeit wird die Wahl dennoch auf *Python*, da die Entwicklungszeit enorm verkürzt werden kann und sie den Standard in der betreuenden Abteilung darstellt. Des Weiteren kann in *Python* eine Vielzahl optionaler Bibliotheken eingebunden werden, was den Funktionsumfang dieser Sprache noch mehr erweitert. Während der Durchführung dieser Arbeit wurde auf drei Bibliotheken zurückgegriffen, welche in den folgenden Kapiteln noch separat vorgestellt werden.

Als Entwicklungsumgebung kommt *Eclipse 3.7.0* zum Einsatz. Diese leistungsfähige und weit verbreitete IDE (*Integrated Development Environment*) ist modular aufgebaut und kann über Plug-Ins erweitert werden.

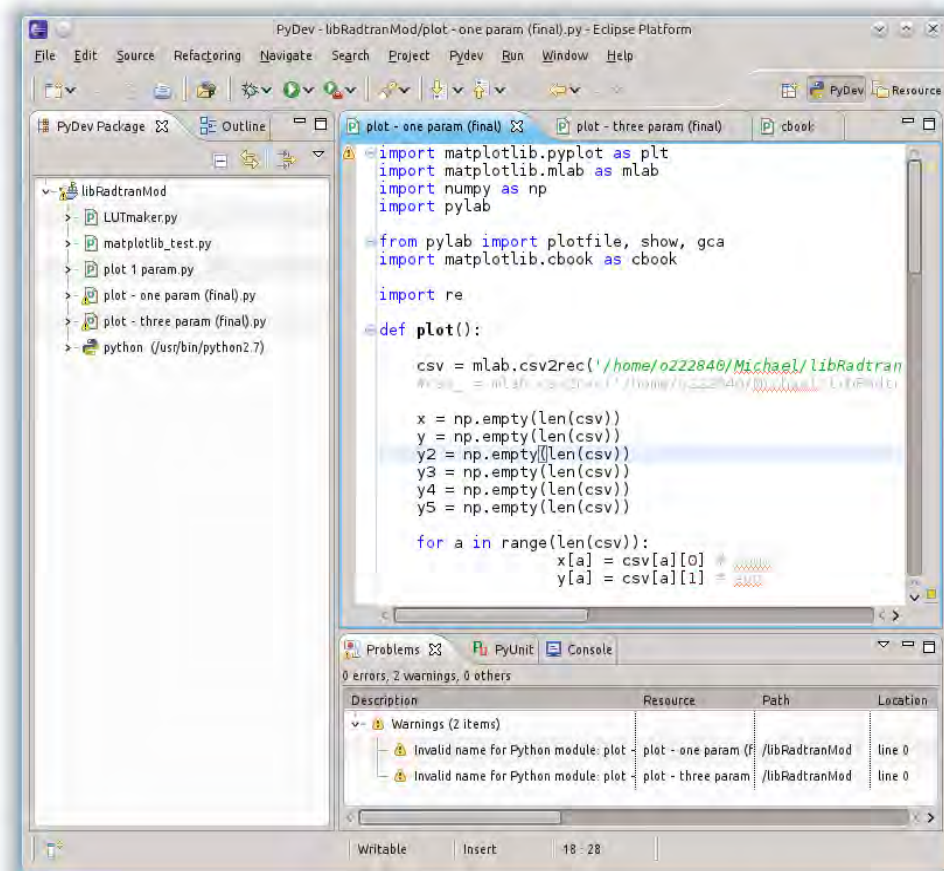
Da sie in *Java* geschrieben ist, kann sie plattformunabhängig auf allen gängigen Betriebssystemen ausgeführt werden [Holzner, 2004]. Ehe unter *Eclipse* Skripte in *Python* entwickelt werden können, wird *Eclipse* mit der Entwicklungsumgebung *PyDev 2.4.0* auf *Python* vorbereitet. *PyDev* ist ein Plug-In für *Eclipse*, welches zum Beispiel Syntax Highlighting, Textvervollständigung oder einen Debugger für *Python* zur Verfügung stellt [PyDev.org, 2012]. Man beachte, dass *Python* selbst und *Eclipse* mit *PyDev* unabhängig voneinander installiert werden. Erst durch die entsprechende Konfiguration von *Eclipse*, wie sie in [Dermitzel, 2012] beschrieben ist, können *Python*-Skripte entwickelt werden. Die Bibliotheken, welche im Verzeichnis von *Python* installiert wurden, sind dann auch in *Eclipse* zugänglich.

Abbildung 21 zeigt einen Screenshots der Entwicklungsumgebung *Eclipse* mit dem Plug-In für *Python*. Darin kann man erkennen, wie zum Beispiel im linken Fenster die Skripte des aktuellen Projektes aufgelistet werden, oder wie im Hauptfenster das Skript einer Ausgabe-Visualisierung entwickelt wird. Abschließend stellt Abbildung 22 die Funktionalitäten des Debug-Modus dar.

4.1.1 Mathematische Bibliothek NumPy

NumPy ist eine mächtige Bibliothek für technische und ingenieurwissenschaftliche Anwendungen, wobei der Name *NumPy* sich von *Numerical Python* ableiten lässt. Nachdem bestimmte mathematische Algorithmen vom Standard-*Python* nur verhältnismäßig langsam ausgeführt werden, bietet *NumPy* einen effizienteren Datentyp. Dieser erlaubt die Verwendung und Verarbeitung von mehrdimensionalen Arrays. Der Datentyp *array* ist sehr maschinennah und kann daher wesentlich schneller verarbeitet werden als andere Datentypen. Des Weiteren wird *Python* dank *NumPy* um Funktionen der linearen Algebra und der Fourier-Transformation erweitert [Hajji, 2008], auf welche im Folgendem ebenfalls zurückgegriffen wird.

In dieser Arbeit wird *NumPy* für die Visualisierung der *uvspec* Output-Dateien benötigt. Erst nachdem die Informationen der Output-Dateien in korrekter Form in einem Array abgelegt wurden, können die Prozeduren der Bibliothek *matplotlib* auf die Daten zugreifen, um sie letztendlich in einer Grafik auszugeben. Auch bei der Implementierung des Adjacency-Effekts, spielt der durch *NumPy* eingeführte Datentyp eine große Rolle. Details sind dem Kapitel 6 zu entnehmen.

Abbildung 21: *Python* in der Entwicklungsumgebung *Eclipse*

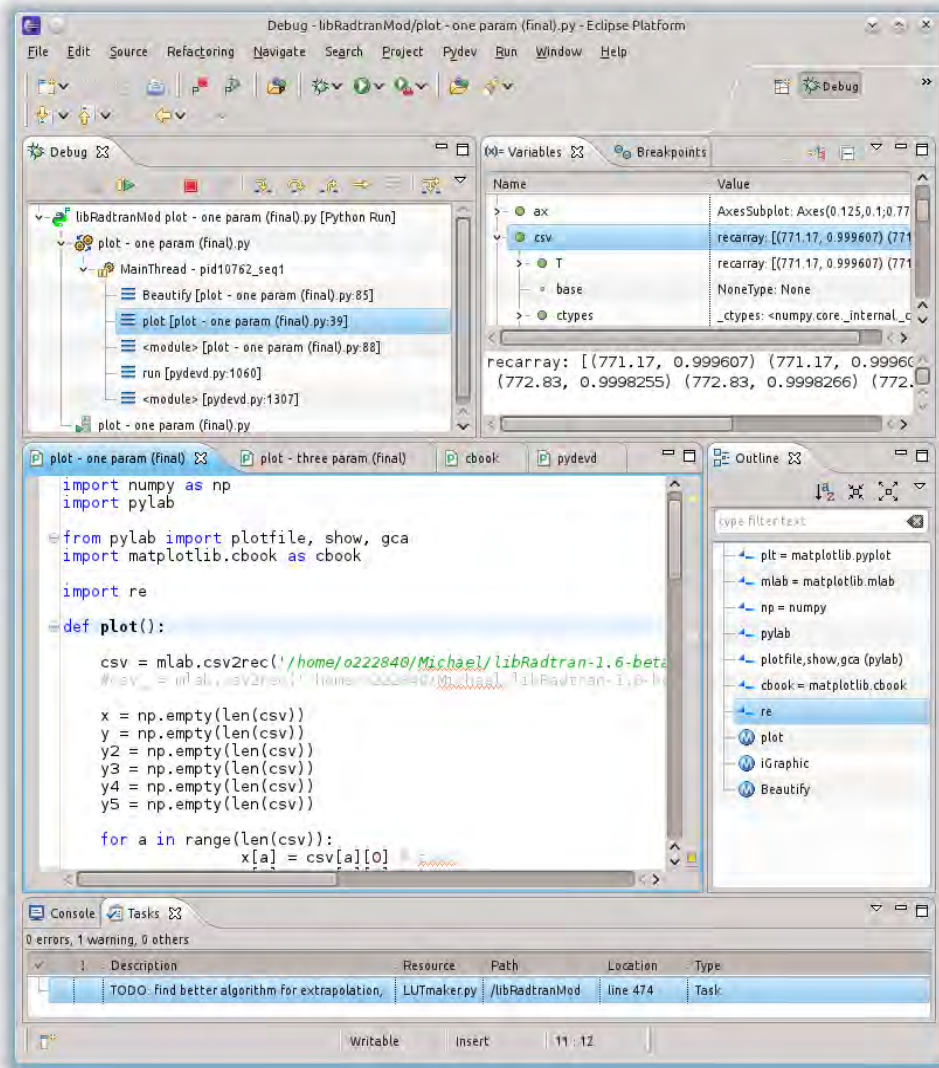


Abbildung 22: Eclipse im Debug-Modus

4.1.2 *Wissenschaftliche Bibliothek SciPy*

Die Bibliothek *SciPy* ist von *NumPy* abhängig, da sie auf deren Funktionen für die schnelle Bearbeitung mehrdimensionaler Arrays zurückgreift. *SciPy* ist ebenso wie *NumPy* eine Sammlung mathematischer Algorithmen. Auf Grund ihres Einsatzbereiches wurde der Name *Scientific Python*, kurz *SciPy* gewählt. Unter anderem stellt es Funktionen für folgende Bereiche zur Verfügung [SciPy.org, 2012]:

- Statistik
- numerische Integration
- lineare Algebra
- Fourier Transformationen
- Signalverarbeitung
- Bildverarbeitung

SciPy wird, je nachdem welche Visualisierung erfolgen soll, von *matplotlib* und *Py4CATS* benötigt.

4.1.3 *Bibliothek zur Visualisierung matplotlib*

matplotlib ist ein Python-Package, um zwei- oder mehrdimensionale Graphen in druckreifer Qualität zu erzeugen. Die vielseitigen Einsatzmöglichkeiten sind exemplarisch in [matplotlib.org, 2012] dargestellt. Um ein fundiertes Verständnis des Quellcode von *matplotlib* zu erlangen, welches eine große Individualisierung der Graphen erlaubt, sei einem [Tosi, 2009] nahegelegt.

Ein essentieller Part dieser Arbeit besteht darin, die von Strahlungstransfermodellen generierten Daten zu visualisieren. Im Idealfall liegen diese Daten als *Comma-Separated Values (CSV)* vor. CSV-Dateien sind ein beliebtes Format zum Austausch von Datenbanken. Die Funktionsweise von CSV-Dateien soll anhand eines trivialen Beispiels mit dem Programm *libreOffice Calc* (ähnlich *Microsoft Excel*) veranschaulicht werden. Wird die in Listing 9 beschriebene CSV-Datei von *libreOffice Calc* geöffnet, so wird diese wie in Abbildung 23 dargestellt.

Listing 9: CSV-Datei mit Header, 2 Spalten und 4 Zeilen

```
1 lambda, radiance
2 550.000, 3.218305054e+02
3 550.500, 3.206469727e+02
4 551.000, 3.190541077e+02
```

	A	B
1	lambda	radiance
2	550,000	3.218305054e+02
3	550,500	3.206469727e+02
4	551,000	3.190541077e+02

Abbildung 23: CSV-Datei in *libreOffice Calc*

In der vorliegenden Arbeit werden CSV-Dateien benötigt, um die von *libRadtran* oder *MODTRAN* generierten Output-Dateien zu visualisieren. Dazu wird die Output-Datei von dem Programm *awk* gelesen, welches aus den gewünschten Ausgabe-Parametern eine CSV-Datei generiert. Die Handhabung von *awk* wird im Abschnitt 4.3 beschrieben.

4.2 TEXTEDITOR *kate*

Da während der Entwicklungsphase mit vielen Text-Dateien in unterschiedlichsten Formaten hantiert wird, ist es sinnvoll einen mächtigen Texteditor zu verwenden. Dabei fiel die Wahl auf *Kate* (*KDE*¹ *advanced text editor*), welcher ein projektbasiertes Arbeiten mit mehreren Dateien unterstützt. Vor allem das implementierte Kommandozeilen-Fenster und Syntax-Highlighting erleichtern das Editieren von Dateien. Hilfreich ist auch, dass mehrere Dateien gleichzeitig bearbeitet und getrennt dargestellt werden können [Plötner and Wendzel, 2012]. Abbildung 24 veranschaulicht die genannten Funktionalitäten anhand eines Screenshots von *Kate*.

¹ *KDE* (*K Desktop Environment*) ist eine grafische Benutzeroberfläche für Unix-Betriebssysteme. So weist der Letter "K" in *Kubuntu* auch darauf hin, dass es sich um eine *Ubuntu*-Distribution mit der Oberfläche *KDE* handelt.

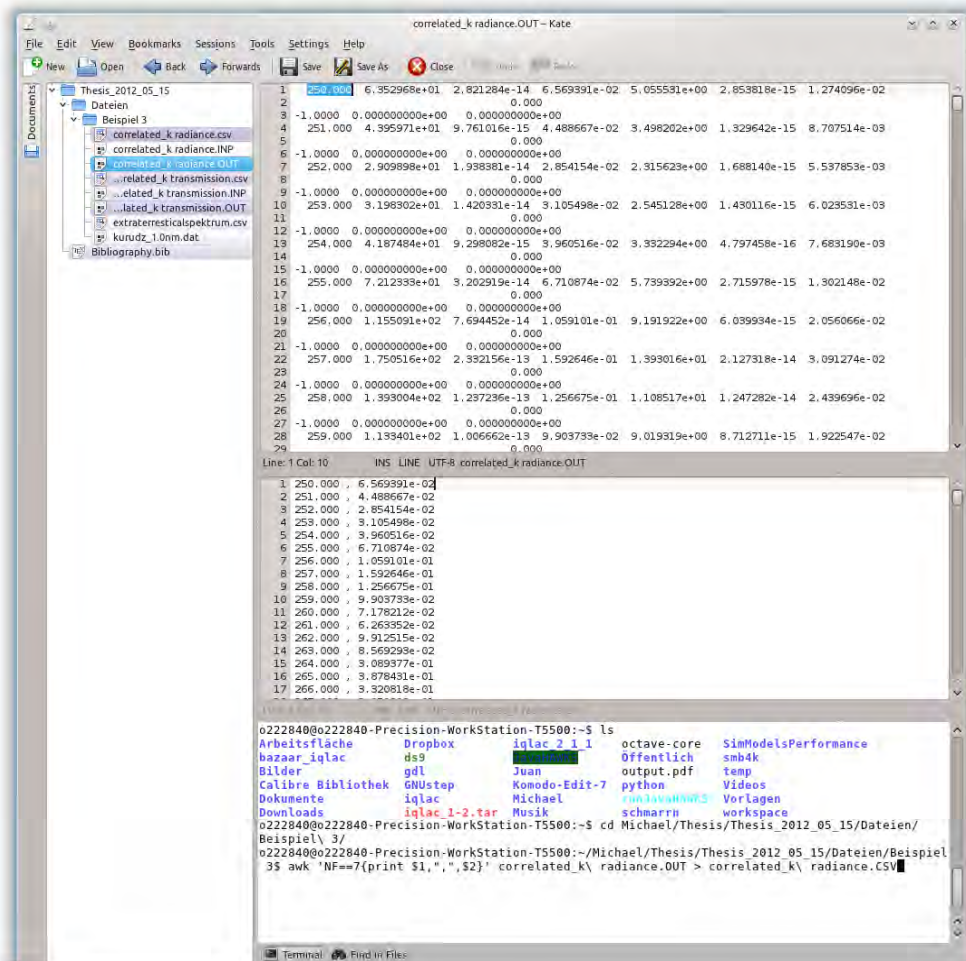


Abbildung 24: Texteditor *Kate* mit zwei geöffneten Dateien und Kommandozeilen-Fenster

4.3 UNIX-TOOL *awk*

awk ist ein kommandozeilenbasiertes Unix-Programm um strukturierte Text-Dateien auszuwerten und/oder zu schreiben. *Strukturierte Text-Dateien* sind zum Beispiel die Output-Dateien von *MODTRAN* oder *uvspec*, da diese einen festen Aufbau folgen, wie auf Seite 28 beschrieben. Auch CSV-Dateien, wie sie als Input für einige *matplotlib*-Skripte benötigt werden, können mit *awk* gelesen oder geschrieben werden.

Die zentrale Funktion von *awk* ist es, Textdateien nach Zeilen zu durchsuchen, die ein festgelegtes Muster beinhalten. Entspricht eine Zeile diesem Muster, so wird ein definierter Befehl ausgeführt. Auf diese Weise

arbeitet *awk* nacheinander alle Zeilen des Textdokuments ab, bis die Datei zu Ende ist [Robbins and Surmeli, 2002].

awk wird über die Kommandozeile aufgerufen. Dabei wird das Suchmuster, die auszuführende Aktion und die Ein- und Ausgabedatei beschrieben. Es ist zu beachten, dass *awk* im selben Verzeichnis ausgeführt wird, in welchem sich die Ein- und Ausgabedatei befinden. Anderenfalls muss zusätzlich der relative oder absolute Pfad mit angegeben werden. Des Weiteren existiert die Möglichkeit, mehrere Suchmuster und die daraus resultierenden Aktionen zu definieren. Dazu bedarf es folgender Syntax:

```
awk 'muster_1 {aktion_1}; muster_2 {aktion_2}'  
zu_lesende_datei > zu_schreibende_datei
```

Im Rahmen dieser Arbeit wird auf *awk* zurückgegriffen, um die von *uvspec* generierten Output-Dateien zu lesen, gezielt Parameter zu extrahieren, welche daraufhin in eine neue Datei geschrieben werden. Dieser Vorgang wird auch als *Parsing* bezeichnet. Dabei muss die neu generierte Datei ein Format darstellen, das von dem Visualisierungsskript *matplotlib* gelesen werden kann.

Sollen beispielsweise die Wellenlänge und die Radiance von der auf Seite 30 gezeigten *uvspec* Output-Datei (Listing 2) extrahiert und in eine CSV-Datei geschrieben werden, muss *awk* folgendermaßen aufgerufen werden:

```
awk 'NF==7{lambda=$1}; NF==3{print lambda,",",,$3}'  
radiance.OUT > radiance.CSV
```

Das Muster "NF==7" bedeutet, dass die Aktion "lambda=\$1" lediglich auf Zeilen mit sieben Feldern angewandt werden soll. "lambda=\$1" besagt wiederum, dass der String des ersten Feldes in der Variable *lambda* gespeichert werden soll. In der *uvspec* Output-Datei steht im ersten Feld die zum Block zugehörige Wellenlänge. Sobald das Muster "NF==3" entdeckt wurde, also eine Zeile aus drei Feldern besteht, wird die Variable *lambda* mit dem Befehl "print" in die Ausgabedatei *radiance.CSV* geschrieben. In dieselbe Zeile wird ebenso das Komma "," und der String "\$3", also die Radiance im dritten Feld, geschrieben. Wurde *awk* in obiger Form auf alle Zeilen der Eingabedatei *radiance.OUT* angewendet, sieht die neue Ausgabedatei *radiance.CSV* folgendermaßen aus:

Listing 10: *uvpsec* Output-Datei CSV

```
1 550.000 , 3.218305054e+02  
2 550.500 , 3.206469727e+02  
3 551.000 , 3.190541077e+02
```

4.4 PYTHON-SKRIPT ZUR VISUALISIERUNG

Sobald eine CSV-Datei vorliegt, kann diese mit der in Abschnitt 4.1.3 vorgestellten Python-Bibliothek *matplotlib* visualisiert werden. Nachdem im Rahmen dieser Arbeit immer wieder derselbe Datentyp aufbereitet wird, soll im Folgendem ein lauffähiges Python-Skript vorgestellt werden, mit welchem beispielsweise die Abbildung 41 (Ausschnitt 1) auf Seite 94 generiert wurde. Listing 11 zeigt besagtes Python-Skript, welches im weiteren Verlauf erörtert werden soll.

Listing 11: Python-Skript zur Visualisierung von CSV-Dateien

```
1 # import required libraries:
2 import matplotlib.pyplot as plt
3 import matplotlib.mlab as mlab
4 import numpy as np
5 import pylab
6
7 # define main function:
8 def plot():
9     # import data (wavelength and radiance) from CSV-files:
10    csv = mlab.csv2rec('/home/MIK/MODTRAN.csv', delimiter=',')
11    csv_2 = mlab.csv2rec('/home/MIK/UVSPEC.csv', delimiter=',')
12    csv_3 = mlab.csv2rec('/home/MIK/HITRAN.csv', delimiter=',')
13    # initialise arrays with given size:
14    x = np.empty(len(csv))
15    y = np.empty(len(csv))
16    y1 = np.empty(len(csv_3))
17    y2 = np.empty(len(csv_3))
18    y3 = np.empty(len(csv_3))
19    x3 = np.empty(len(csv_3))
20    # because the array 'csv3' from HITRAN is longer than 'csv1' and
21    # 'csv2', it is necessary to allocate the corresponding
22    # wavelengths (wl) to each other and duplicate radiance values:
23    for a in range(len(csv)):
24        x[a] = csv[a][0] # write wl from MODTRAN into 'x'
25        point = 0 # reset for each checked wl
26        for a in range(len(csv_3)):
27            x3[a] = csv_3[a][0] # write wl from HITRAN into 'x3'
28            y3[a] = csv_3[a][1] # write rad from HITRAN into 'y3'
29            for m in range(len(x)):
30                # if wl from HITRAN and MODTRAN/UVSPEC are equal,
31                # write corresponding rad into 'y1' and 'y2',
32                # which have same dimension as HITRAN's array:
33                if x3[a]==x[m]:
```

```

34         y1[point:a+m]= csv[m][1]*10
35         y2[point:a+m]= csv_2[m][1]
36         point=a
37     Fig = iGraphic('')          # set dynamic font- and labelsizes
38     ax = Fig.add_subplot(111) # create just one plot
39     ax.plot(x3,y1)              # plot data form MODTRAN
40     ax.plot(x3,y2)              # plot data form UVSPEC
41     ax.plot(x3,y3)              # plot data form HITRAN
42     ax.set_xlim((588,602))n     # set wl to be plotted
43     L = ('MODTRAN 4', 'libRadtran', 'libRadtran with HITRAN')
44     xL = '$wavelength$+' '+' '$in$+' '+' '$nm$'
45     yL = '$radiance$+' '+' '$(TOA)$+' '+' '$in$+' '+' '$mW m2 nm sr$'
46     Beautify(ax,xL,yL,L,T)      # modify graphic with funtion Beautify
47
48 # set font- and labelsizes with function iGraphic:
49 def iGraphic(FontSize=''):
50     params = {'axes.labelsize': 20,
51              'text.fontsize': 20,
52              'axes.titlesize': 25,
53              'legend.fontsize': 15,
54              'xtick.labelsize': 10,
55              'ytick.labelsize': 10 }
56     # store parameter in dictionary-like variable 'rcparams':
57     pylab.rcParams.update(params)
58     return fig
59
60 # modify graphic with function Beautify:
61 def Beautify(hAx,Capx,Capy,Caption,Title):
62     hAx.set_xlabel(Capx,style='italic', fontsize='large')
63     hAx.set_ylabel(Capy,style='italic', fontsize='large')
64     hAx.set_title(Title) # set title
65     hAx.grid(True)      # use grid
66
67 plot()                 # execute function plot()
68 plt.show()             # show plot

```

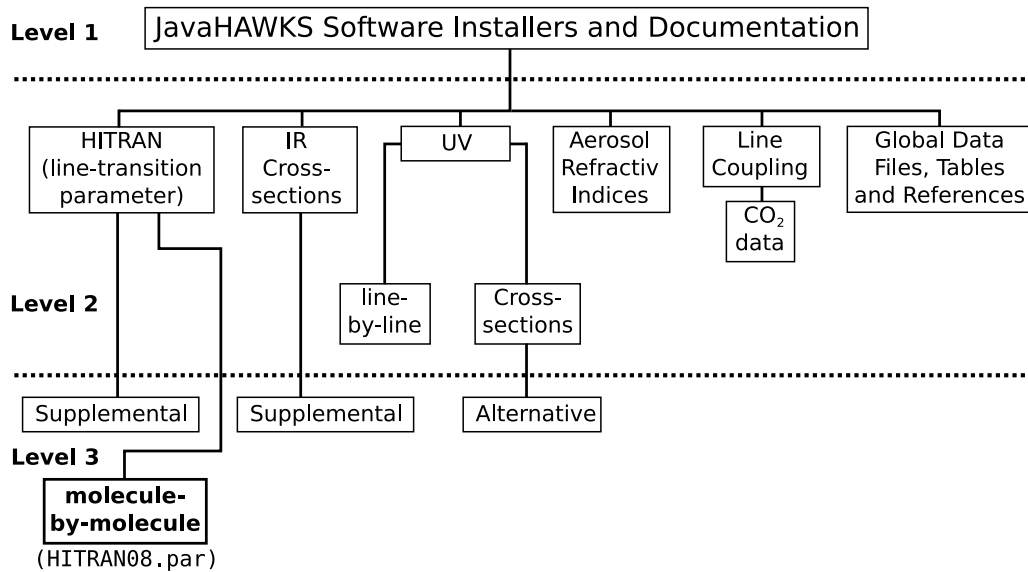
Das Skript beginnt mit dem Import der Bibliotheken, auf welche während der Programmausführung zurückgegriffen wird (Zeile 2-5). Prinzipiell ist dieses Skript in drei Funktionen unterteilt, die nacheinander definiert und abschließend aufgerufen werden. Diese Aufteilung sorgt für eine bessere Übersicht, dank der eine spätere Modifikation erleichtert werden soll. So können beispielsweise über die Funktion *iGraphic()* (Zeile 49-58) die einzelnen Fontgrößen schnell ausfindig gemacht und editiert werden. Des Weiteren können über die Funktion *Beautify()* (Zeile 61-65) der Fonttyp, sowie allgemeine Eigenschaften des Plots einfach geändert werden.

Die Funktion *plot()* stellte das zentrale Element des Skriptes dar (Zeile 8-46), worin die CSV-Dateien eingelesen und in Arrays gespeichert werden (Zeile 10-12). Die größte Herausforderung stellt das Umformen der Arrays mit dem von *MODTRAN* und *uvspev* generierten Output dar. Wie in Abbildung 7 auf Seite 13 dargestellt, werden beim *line-by-line*-Verfahren wesentlich mehr Wellenlängen berücksichtigt, als in einem Bändermodell. In diesem Fall bedeutet dies, dass das von *HITRAN* generierte Array wesentlich länger ist, als diejenigen von *MODTRAN* oder *uvspev*. Dennoch sollen alle drei Arrays über dieselbe Wellenlänge geplottet werden. Um die Arrays aneinander anzupassen, wird auf drei ineinander verschachtelte *for*-Schleifen zurückgegriffen (Zeile 23-36). Hier wird als Abszissenachse das Array *x3* gewählt, welches sämtliche auftretenden Wellenlängen von *HITRAN* beinhaltet. Dementsprechend werden die zugehörigen Funktionswerte, also die Radiance, in das Array *y3* geschrieben. In der innersten *for*-Schleife (Zeile 29-36) wird die niedriger aufgelöste Radiance von *MODTRAN* und *uvspev* "gedehnt", sodass jeder in *HITRAN* enthaltenen Wellenlänge eine Radiance zugeordnet werden kann. Werden die bearbeiteten Arrays visualisiert, stellen die "gedehnten" Radiance-Werte von *MODTRAN* und *uvspev* die horizontalen Geraden in der Abbildung 41 (Ausschnitt 1) dar. Letztendlich wird mit der horizontalen Darstellung in diesem Fall eine bessere Vergleichbarkeit mit dem höher aufgelösten *HITRAN* erzielt.

4.5 MOLEKULARE ABSORPTIONSDATENBANK *hitran*

Wie bereits in Abschnitt 2.3 beschrieben, ist *HITRAN* eine Datenbank für die Molekularabsorption in der Erdatmosphäre. Im Rahmen dieser Arbeit wird lediglich eine Datenbank von den vielen aus der *HITRAN Compilation* benötigt. Die komplette *Compilation* ist nach einer Registrierung frei zugänglich². Abbildung 25 zeigt die Struktur der *HITRAN Compilation*. Anhand dieser kann die Datenbank *molecule-by-molecule* zugeordnet werden, welche primär von Interesse ist. Sie beinhaltet sämtliche molekularen Absorptionsparameter, welche für das *line-by-line*-Verfahren notwendig sind [Rothman et al., 2009]. Die benötigte Datenbank ist an dem Namen "*HITRAN08.par*" zu erkennen und kann auch separat heruntergeladen werden.

² Nach der Registrierung unter <http://www.cfa.harvard.edu/hitran/> erhält man per E-Mail die Zugangsdaten für einen FTP-Server, auf welchem die *HITRAN Compilation* hinterlegt ist. (Stand: 17.05.2012)

Abbildung 25: Struktur der *HITRAN Compilation* [Rothman et al., 2004]

Zusammen mit der *HITRAN Compilation* wird auch das Software-Paket *JavaHAWKS* zum Verwalten und Sichten der Datenbanken geliefert. Das Manual [Rothman et al., 2004] bietet eine umfassende Beschreibung der *HITRAN Compilation* und erklärt auch die Handhabung von *JavaHAWKS*. Man könnte damit die gewünschten Daten zwar extrahieren, doch bietet es sich an, mit den Python Skripten von *Py4CATS* zu arbeiten. Diese Methode erlaubt es, die komplette Extraktion, sowie alle anderen notwendigen Bearbeitungsschritte in einem zukünftigen Meta-Skript zu vereinen.

4.6 BEARBEITUNG VON *hitran* MIT *py4cats*

Py4CATS steht für *Python scripts for Computational Atmospheric Spectroscopy* und ist eine Sammlung von *Python*-Skripten zur Bearbeitung der Datenbank *HITRAN*. In der vorliegenden Arbeit sollen damit ausgewählte Absorptionsparameter aus *HITRAN* extrahiert und so formatiert werden, dass sie von *libRadtran* gelesen werden können.

Damit *Py4CATS* ausgeführt werden kann, muss *Python* (Version 2.5 oder besser) mit *NumPy*, *SciPy* und *matplotlib* installiert sein. *Py4CATS* kann von der *libRadtran*-Homepage³ bezogen werden, wo auch eine Installations-Anleitung zu finden ist.

Für die Umsetzung des *line-by-line* Verfahrens werden die Skripte *extract* und *lbl2od* benötigt, welche über die Kommandozeile ausgeführt

³ Die genaue Adresse lautet: <http://www.libradtran.org/doku.php?id=lbl2od> (Stand: 17.05.2012)

werden. Im ersten Schritt werden mit *extract* die gewünschten Spektralparameter aus der *HITRAN*-Datenbank extrahiert. Dabei besteht die Möglichkeit sich auf einen Wellenlängenbereich oder ausgewählte Moleküle zu beschränken. Liegen im gewählten Extraktionsspektrum verschiedene Moleküle und wurde keine molekulare Auswahl vorgenommen, so generiert *extract* für jedes auftretende Molekül eine separate Datei mit den molekularen Spektralparametern (siehe Abbildung 26). Im nächsten Schritt wandelt *lbl2od* die spektralen Parameter *line-by-line* in die optische Dicke (*optical depth*) der Atmosphäre [Schreier, 2012]. Diese Datei kann dann von in der *uvspec* Input-Datei als Parameter *molecular_tau_file* referenziert werden und ermöglicht so eine *line-by-line* Berechnung mit *libRadtran*.

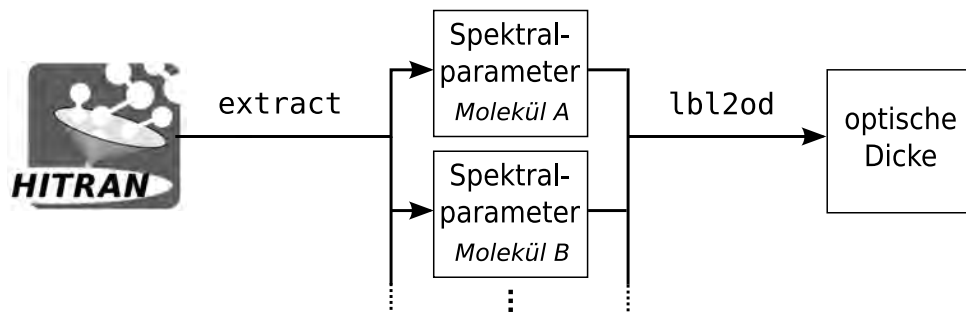


Abbildung 26: Software-Flussdiagramm von *Py4CATS*

Des Weiteren stellt *Py4CATS* Atmosphären-Dateien in unterschiedlichen Auflösungen zur Verfügung. Diese sind genauer als die von *libRadtran* mitgelieferten Atmosphärenbeschreibungen und stellen daher eine empfehlenswerte Alternative dar. Sie sind Bestandteil des *PY4CATS Package* und sind an der Endung “*.xy” zu erkennen.

4.6.1 Extrahieren der Spektralparameter mit *extract*

Wie bereits erwähnt, wird *extract* über die Kommandozeile ausgeführt, wobei der Aufruf folgende allgemeine Form hat:

```
extract [options] line_parameter_database
```

Die Optionen spezifizieren, welche molekularen Spektralparameter extrahiert werden sollen und “line_parameter_database” stellt den Pfad zu der auszulesenden *HITRAN*-Datenbank dar. Im Konkreten kann der Aufruf beispielhaft folgende Form annehmen:

```
extract -f vSEasni -x 12903,13245 -m 02  
-o /home/.../02.vSEasni /home/.../HITRAN08.par
```

Die Option “-f vSEasni” legt das Format der zu generierenden Datei fest. Soll die *HITRAN*-Datenbank letztlich in *libRadtran* eingelesen werden, so ist das Format “vSEasni” zwingend erforderlich. Dabei entspricht jeder Letter einer Spalte in der generierten Datei. Die Bedeutung der Letter ist in Tabelle 5 beschrieben. Man beachte, dass diese Parameter die Grundlage für die in Abschnitt 2.5 vorgestellte Berechnung der optischen Dicke und somit der Lösung der Strahlungstransfergleichung sind.

LETTER	BEDEUTUNG	FORMELZEICHEN
v	Wellenzahl	ν
S	Stärke der Absorption	S
E	niedriger Energiezustand	
a	Ausdehnung der Luft bei 296 K	$\alpha_{L,air}^0$
s	eigene Ausdehnung bei 296 K	$\alpha_{L,self}^0$
n	Temperaturexponent der Druckerhöhung	n
i	Isotopen-Nummer	

Tabelle 5: Parameter zur Definition des Ausgabeformats von *extract*

Die Option “-x a,b” legt die untere und obere Grenze des zu extrahierenden Spektrums fest. a und b werden als Wellenzahl in der Einheit cm^{-1} angegeben. Die Umrechnung zwischen der Wellenzahl k in cm^{-1} und der Wellenlänge λ in nm erfolgt mit:

$$k = \frac{10^7}{\lambda} \quad (4.1)$$

Mit der Option “-m” kann die Extraktion auf ausgewählte Moleküle beschränkt werden. Eine Auflistung aller 39 in *HITRAN* vertretenen Molekül ist in [Rothman et al., 2004] zu finden. Wird die Option “-m” nicht verwendet, so werden alle im ausgewählten Spektrum auftretenden Moleküle berücksichtigt.

Wurde lediglich ein Molekül ausgewählt, so muss mit der Option “-o path” angegeben werden, unter welchem Pfad und Dateinamen die extrahierte Datenbank gespeichert werden soll. Der Dateiname muss dabei die selbe Endung aufweisen, wie mit der Option “-f vSEasni” festgelegt wurde. Sollen mehrere oder alle Moleküle extrahiert werden, entfällt die Option “-o path” und es wird für jedes Molekül eine eigene Datei mit der richtigen Endung erzeugt.

4.6.2 Berechnung der optischen Dicke mit *lbl2od*

Das Python-Skript *lbl2od* berechnet, wie in Abschnitt 2.5 hergeleitet, aus den extrahierten Spektralparametern die optische Dicke der Atmosphäre. *lbl2od* wird über der die Kommandozeile ausgeführt und hat folgende allgemeine Form:

```
lbl2od [options] atm_file line_parameter_file(s)
```

“*atm_file*” ist die Atmosphärenbeschreibung im Format “*.xy”. Sollten die in *libRadtran* vorhandenen Atmosphärenbeschreibungen verwendet werden, kann das Python-Skript *atmos* diese in das xy-Format konvertieren. *atmos* ist Bestandteil des *PY4CATS Package*. Wie bereits erwähnt, ist es jedoch empfehlenswerter die von *PY4CATS* mitgelieferten Atmosphärenbeschreibungen zu verwenden.

Die “*line_parameter_file(s)*” sind die von *extract* generierten Datenbanken. Deren Pfad muss beim Aufruf von *lbl2od* ebenfalls mit angegeben werden und kann beispielhaft folgende Form annehmen:

```
lbl2od -o /home/.../optDepth.txt --nm -r
/home/.../US_standard50.xy
/home/.../O2.vSEasni /home/.../H2O.vSEasni
```

Mit der Option “--nm” erfolgt die Ausgabe der optischen Dicke über die Wellenlänge in *nm*, anstelle der Wellenzahl in cm^{-1} . Zusätzlich muss durch die Option “-r” die vertikale Anordnung der optischen Dicke invertiert werden. Diese Formatierungen sind notwendig, damit die generierte Datei von *libRadtran* gelesen werden kann.

Die Option “-o path” besagt, in welcher Ausgabedatei die optische Dicke gespeichert werden soll. Wie bereits erwähnt, kann diese Datei in der *uvspec* Input-Datei als *molecular_tau_file* referenziert werden und ermöglicht so die Berücksichtigung der Spektralparameter. Zuvor ist jedoch eine kleine Modifikation an dieser Datei von Nöten, welche im nächsten Abschnitt beschrieben werden soll.

4.6.3 Anpassung der *molecular_tau_file* für *uvspec*

Leider entspricht das Format der *molecular_tau_file* noch nicht ganz den Anforderungen von *libRadtran*. Ehe sie von *uvspec* gelesen werden kann, muss die von *lbl2od* erzeugte Datei, wie sie in Listing 12 dargestellt wird, editiert werden: Die in Zeile 1 von Listing 12 aufgeführten Höhen müssen hinter den Kommentaren nochmals eingefügt werden, wie in der Zeile 8 von Listing 13 gezeigt⁴. Die so angepasste Datei kann als *molecular_tau_file* in der *uvspec* Input-Datei referenziert werden.

⁴ Um eine bessere Übersicht zu gewährleisten, wurden die in Listing 12 und Listing 13 gezeigten Dateien gekürzt und deren Zahlenwerte teilweise gerundet.

Listing 12: durch *lbl2od* generierte Datei mit der optischen Dicke

```

1 # altitudes [km]:      70.0    60.0    50.0    40.0    30.0    25.0
2 # temperatures [K]:   219.60  247.00  270.70  250.40  226.50  221.60
3 # pressures [mb]:     0.0522  0.219   0.798   2.87    12     25.5
4 # gases:              02     H20
5 # vcd [molec/cm^2]:  4.6e+24
6 # wavelength optical depth
7 # nm
8 759.546  3.034e-09 4.546e-08 5.539e-07 8.259e-06 2.059e-05 9.540e-05
9 759.556  3.044e-09 4.695e-08 5.562e-07 8.278e-06 2.209e-05 9.535e-05
10 759.567  3.055e-09 4.705e-08 5.543e-07 8.299e-06 2.102e-05 9.583e-05
11 759.598  3.062e-09 4.721e-08 5.365e-07 8.396e-06 2.135e-05 9.587e-05
12 ...

```

Listing 13: angepasste Datei mit der optischen Dicke

```

1 # altitudes [km]:      70.0    60.0    50.0    40.0    30.0    25.0
2 # temperatures [K]:   219.60  247.00  270.70  250.40  226.50  221.60
3 # pressures [mb]:     0.0522  0.219   0.798   2.87    12     25.5
4 # gases:              02     H20
5 # vcd [molec/cm^2]:  4.6e+24
6 # wavelength optical depth
7 # nm
8 70.0     60.0     50.0     40.0     30.0     25.0
9 759.546  3.034e-09 4.546e-08 5.539e-07 8.259e-06 2.059e-05 9.540e-05
10 759.556  3.044e-09 4.695e-08 5.562e-07 8.278e-06 2.209e-05 9.535e-05
11 759.567  3.055e-09 4.705e-08 5.543e-07 8.299e-06 2.102e-05 9.583e-05
12 759.598  3.062e-09 4.721e-08 5.365e-07 8.396e-06 2.135e-05 9.587e-05
13 ...

```

4.7 NACHBEARBEITUNG MIT VEKTORGRAFIK-EDITOR *inkscape*

Inkscape ist ein Editor, mit welchem Vektorgrafiken erstellt und editiert werden können. Mit diesem Programm entstanden beispielsweise auch die meisten Abbildungen der vorliegenden Arbeit. Bedeutender ist jedoch, dass *matplotlib* ebenfalls Vektorgrafiken generiert und diese mit *Inkscape* nachbearbeitet werden können. Dadurch ist ein hoher Grad an Individualisierbarkeit erreichbar, mit welchem die durch Strahlungstransfer-Modelle generierten Daten gezielt grafisch ausgewertet werden können. Abbildung 27 zeigt beispielsweise die Anpassung der Strichstärke.

Vektorgrafiken haben den Vorteil beliebig skalierbar zu sein, da sie nur aus den Koordinatenangaben von Punkten bzw. Strecken bestehen. Eine Größenveränderung der Grafik kann daher ohne Verlust an Bild-details mittels einer einfachen Koordinatentransformation erfolgen [Fischer, 2010]. Dies spielt vor allem beim Vergleich zweier Strahlungstransfermodelle eine große Rolle, da deren visualisierter Output ohne einhergehenden Kompressionsverlust der Grafik gegenübergestellt werden kann.

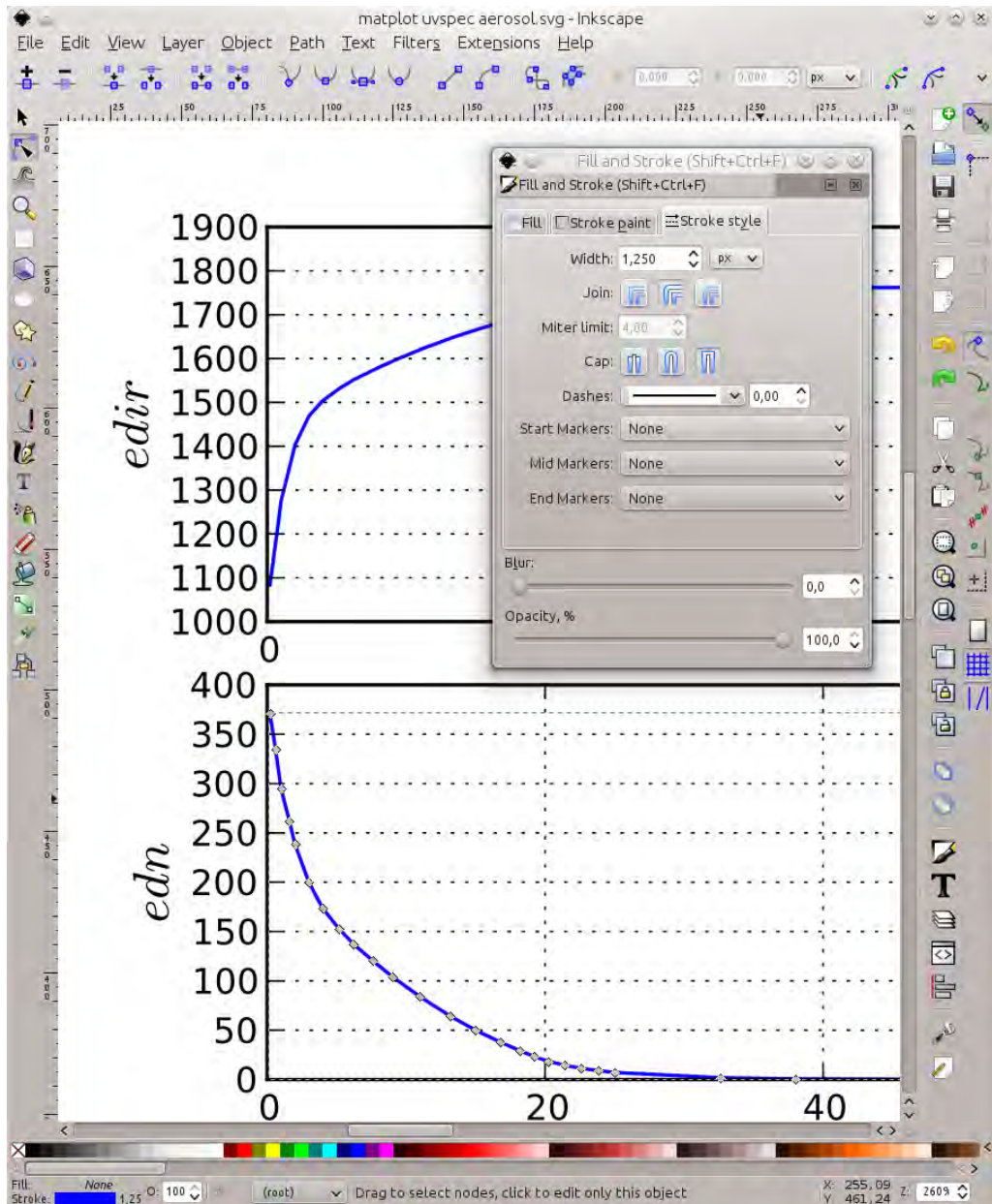


Abbildung 27: Vektorgrafik-Editor *Inkscape*

Teil II

UMSETZUNG DER AUFGABENSTELLUNG

VERGLEICH VON LIBRADTRAN UND MODTRAN

Die zentrale Frage dieser Arbeit lautet, ob das kommerzielle *MODTRAN* mit dem frei zugänglichen *libRadtran* ersetzt werden kann. Um diese Frage zu beantworten, müssen beide Strahlungstransfer-Modelle miteinander verglichen werden. Dazu wird von Herrn Hupfer ein *MODTRAN*₄-Szenario bereitgestellt, welches einem typischen Anwendungsfall entspricht. Dieses Szenario beinhaltet die Eingabe-Datei *tape5*, sowie die daraus generierte Ausgabe-Datei *tape7*. Die eigentliche Arbeit beginnt damit, eine *uvspec* Input-Datei zu schreiben, welche das selbe Szenario beschreibt wie das gegebene *tape5*.

Für den Vergleich beider Strahlungstransfer-Modelle werden letztlich die von *MODTRAN*₄ gegebene Ausgangs-Datei *tape7* und die von *libRadtran* generierte *uvspec* Output-Datei gegenübergestellt. Sollte die Differenz beider Ausgabe-Daten sehr niedrig oder gar identisch sein, können die Modelle als gleichwertig erachtet werden und sind folglich miteinander austauschbar.

Bei der Interpretation der generierten Radiance-Werte muss berücksichtigt werden, dass die von *MODTRAN*₄ berechneten Ergebnisse um den Faktor 10 kleiner sind, als die von *libRadtran* berechneten. Dies ist darauf zurückzuführen, dass beide Programme ihre Ergebnisse in unterschiedlichen Einheiten angeben. Für die Gegenüberstellung beider Radiance-Werte muss folgender Zusammenhang berücksichtigt werden:

$$L_{MODTRAN} = L_{libRadtran}$$

$$10 \frac{W}{cm^2 \mu m sr} = 1 \frac{W}{m^2 nm sr}$$

5.1 SCHREIBEN DER *uvspec* INPUT-DATEI

Nachdem die Eingabe-Datei *tape5* für *MODTRAN* gegeben ist, muss eine äquivalente *uvspec* Input-Datei erstellt werden, welche ein identisches Szenario beschreibt und dieselbe Ausgabegröße berechnet. Während das *tape5* bereits in Abschnitt 3.2.2 als Screenshot wiedergegeben wird, stellt Listing 14 die äquivalente *uvspec* Input-Datei dar. Die zum Schreiben der *uvspec* Input-Datei benötigten Zahlenwerte dürfen dabei vom *MODTRAN*-Screenshot übernommen werden.

Um sich dem *tape5* nach Abbildung 19 anzupassen, muss über Zeile 2 des Listing 14 die standardisierte Atmosphäre "*midlatitude summer*" aus-

gewählt werden. Des Weiteren wird in *MODTRAN* der Sonnenzenithwinkel *sza* nicht explizit, sondern über Längengrad, Breitengrad und die Zeit angegeben. Um eine Umrechnung zu vermeiden, wird der *sza* in *libRadtran* ebenfalls auf diese Weise umgesetzt. Dies geschieht in den Zeilen 16 bis 18 mit den Parametern *latitude*, *longitude* und *time*.

Listing 14: *uvspec* Input-Datei mit identischem Szenario wie die Eingabe-Datei *tape5* für *MODTRAN*

```

1 # SET GENERAL CONDITION:
2 atmosphere_file ../data/atmmod/afglms.dat
3                                     # atmosphere file for midlatitude summer
4 solar_file ../data/solar_flux/atlas_plus_modtran.dat
5                                     # extraterrestrial solar spectrum
6 albedo 0.4                           # surface albedo
7 wavelength 399 2501                   # wavelength range [nm]
8 rte_solver disort                       # choose RTE solver disort
9
10 # SET GEOMETRY:
11 zout TOA                               # sensor altitude is TOA
12 phi 0                                  # sensor azimuth
13 umu 1                                  # cosine of viewing zenith angle
14 phi0 0                                 # solar azimuth
15 # calculate solar zenith angle sza by latitude, longitude and time:
16 latitude 0
17 longitude 0
18 time 2012 06 28 12 00 00

```

Sobald äquivalente Eingabe-Dateien für *MODTRAN* als auch *libRadtran* bereitgestellt wurden, kann mit der eigentlichen Versuchsdurchführung begonnen werden.

5.2 VERSUCHSAUFBAU DES VERGLEICHS

Der Versuchsaufbau für den Vergleich beider Modelle setzt sich aus den in Kapitel 3 und 4 vorgestellten Programmen zusammen. Die Beziehungen zwischen den Programmen werden in Abbildung 28 als Software-Flussdiagramm veranschaulicht.

Im Wesentlichen besteht der Aufbau aus drei Blöcken: den beiden Modellen und einem Python-Skript zum Auswerten der generierten Daten. In einem ersten Schritt wird das bereits in Abschnitt 3.2.2 beschriebene Szenario von *MODTRAN 4* berechnet. Dabei erfolgen die Schritte (1),

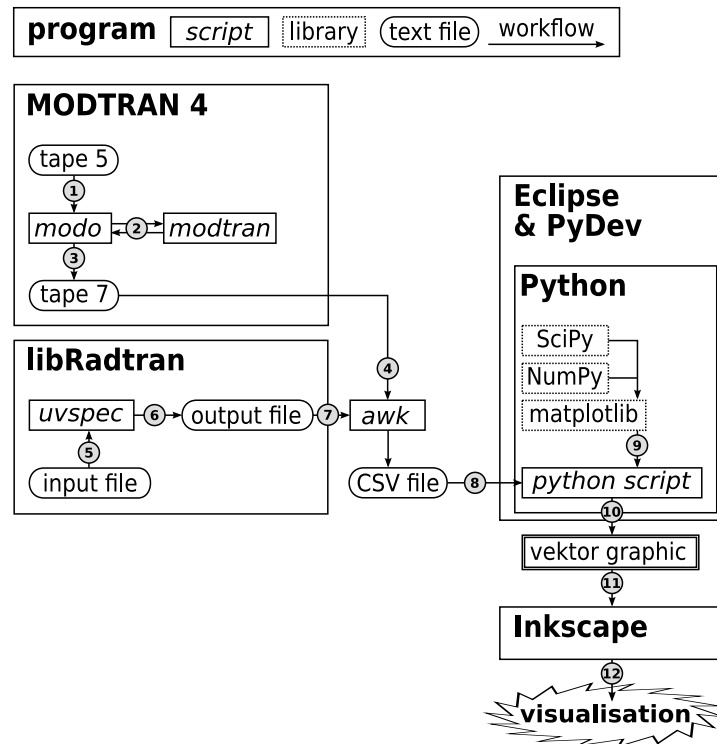


Abbildung 28: Software-Flussdiagramm zum Vergleich der Strahlungstransfermodelle

(2) und (3) über die graphische Benutzeroberfläche *MODO*. Diese Arbeit wurde bereits durch Herrn Hupfer ausgeführt. Die in der Ausgabe-Datei *tape7* enthaltenen Radiance-Werte werden daraufhin mit dem Unix-Tool *awk* extrahiert und in eine CSV-Datei geschrieben (4).

Parallel dazu muss in *libRadtran* eine *uvspec* Input-Datei generiert werden, welche dasselbe Szenario beschreibt, wie das *tape5* bei *MODTRAN* (siehe Abschnitt 5.1). Daraufhin kann *uvspec* anhand dieser Input-Datei auf der Kommandozeile ausgeführt werden (5) und die Output-Datei generieren (6). Wiederum werden die in der Output-Datei enthaltenen Radiance-Werte mit *awk* extrahiert (7) und in eine CSV-Datei geschrieben.

Mit den nun vorliegenden Ausgabe-Dateien beider Strahlungstransfer-Modelle, kann mit dem Vergleich begonnen werden. Dazu werden die beiden CSV-Dateien durch das in Abschnitt 4.4 vorgestellte Python-Skript eingelesen (8) und die Visualisierung in eine Vektor-Grafik ausgegeben (10). Für das Funktionieren des Skriptes wird auf die in Abschnitt 4.1 beschriebenen Bibliotheken zurückgegriffen (9). Um eine bessere Vergleichbarkeit der von *matplotlib* generierten Grafiken zu erzielen, werden diese in *Inkscape* geöffnet (11) und die durch *MODTRAN₄* und *libRadtran* generierten Daten gegenübergestellt (12). Die Auswertung der resultierenden Graphen erfolgt im nächsten Abschnitt.

5.3 AUSWERTUNG, INTERPRETATION UND ERGEBNIS

Abbildung 29 zeigt das mit *Inkscape* aufbereitete Ergebnis des Vergleichs beider Strahlungstransfer-Modelle. Auf den ersten Blick ist zu erkennen, dass es im betrachteten Wellenlängenbereich von 400 bis 2500 nm keine gravierende Unterschiede zwischen den berechneten Verläufen gibt. So können zum Beispiel sämtliche atmosphärischen Fenster in beiden Modellen identisch korrekt ermittelt werden. In der Tat liegt der gemessene Unterschied selten über 5%.

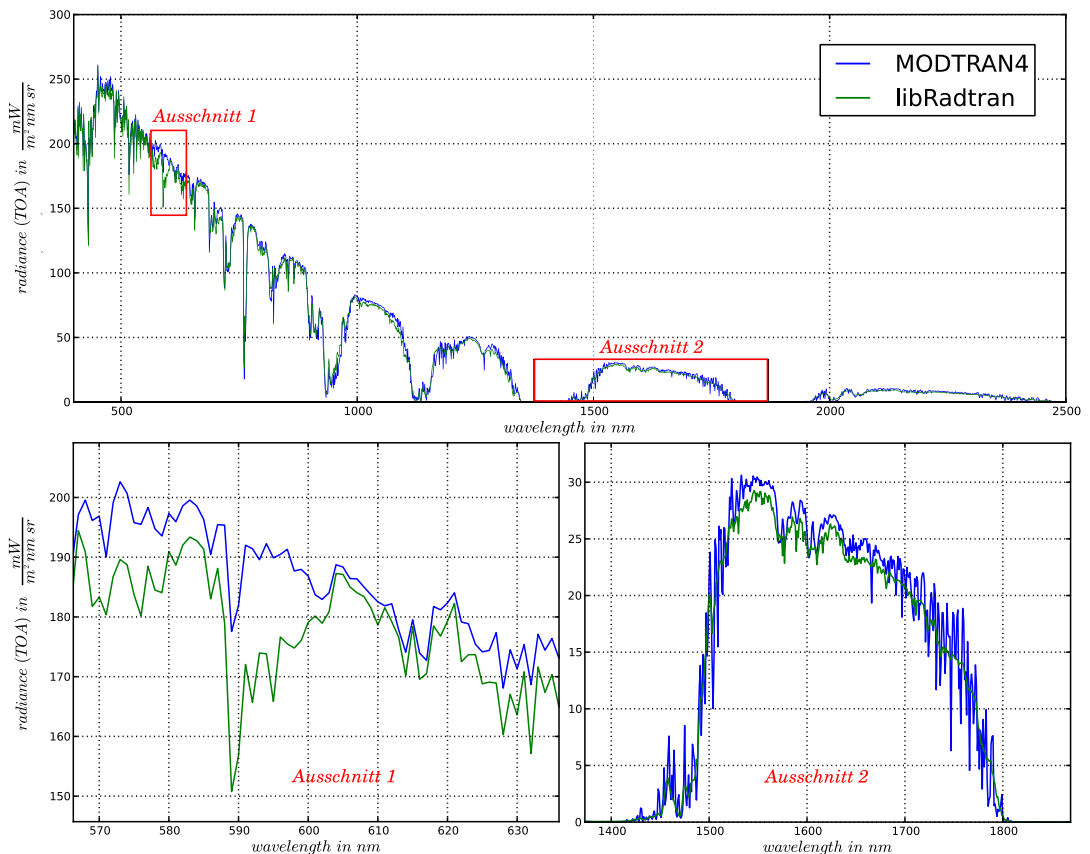


Abbildung 29: Vergleich der Strahlungstransfermodelle *MODTRAN₄* und *libRadtran* mit vergrößerten Ausschnitten.

Eine Ausnahme stellt unter anderem der Bereich zwischen 590 und 600 nm dar (siehe Abbildung 29, Ausschnitt 1). Dort beträgt der Unterschied teilweise signifikante 17%. Solche Differenzen können zwei Ursachen haben. Zum einen ändert sich die extraterrestrische Irradiance der Sonne über das Jahr und über einen Sonnenzyklus. Diese Schwankungen können bis zu $\pm 3,4\%$ betragen [Stine et al., 2012]. Dabei wurden etwaige Messfehler noch nicht berücksichtigt. Abhängig davon, wann das im Modell verwendete extraterrestrische Sonnenspektrum erstellt wurde, können damit die kleineren Differenzen begründet werden. Jedoch hat die starke Abweichung um 590 nm noch eine zweite Ursache.

Sowohl bei *MODTRAN₄* als auch *libRadtran* wurde in diesem Versuch auf ein Bändermodell zurückgegriffen. Wie bereits in Abschnitt 2.3 dargelegt, resultiert ein Bändermodell aus der Mittelung über ein Band. Dabei ist es sehr wahrscheinlich, dass bei den hier eingesetzten Strahlungstransfer-Modellen verschiedene Moleküle berücksichtigt werden. Dies führt zu einem unterschiedlichem Absorptionsverhalten der entsprechenden Bänder und somit zu einer signifikanten Abweichung, wie sie in diesem Fall vorliegt.

Anhand des beispielhaften Ausschnitt 2 der Abbildung 29 soll ein weiteres Unterscheidungsmerkmal der Modelle dargelegt werden. Im Wellenlängenbereich zwischen 1400 und 1800 nm weist der Radiance-Verlauf von *MODTRAN* (blau) wesentlich größere Sprünge auf als der von *libRadtran* (grün). Obwohl der von *libRadtran* ermittelte Verlauf geglättet erscheinen, stimmt er mit dem gemitteltem Verlauf von *MODTRAN* dennoch in weiten Teilen überein. Der sprunghafte Verlauf von *MODTRAN* lässt sich auf ein höher aufgelöstes extraterrestrisches Sonnenspektrum zurückführen. Denn ein Vergleich der Output-Dateien mit dem jeweils verwendeten Sonnenspektrum (*solar_file* bei *uvspec*) ergibt, dass die von *MODTRAN* verwendete Datenbank über eine höhere Auflösung verfügt. In diesem Fall kann die Auflösung der Strahlungstransfer-Modelle nur so hoch sein, wie die des zugrunde liegenden Sonnenspektrums. Dieses Manko von *libRadtran* kann durch die Verwendung einer besser aufgelösten Datenbank beseitigt werden, wie es auch in Kapitel 7 im Verlauf einer Line-by-Line-Simulation vorgeführt wird.

Abschließend lässt sich festhalten, dass die betrachteten Strahlungstransfer-Modelle zwar marginal unterschiedliche Ergebnisse liefern, deren Ursachen aber bekannt sind. Der in *libRadtran* verwendete RTE-Solver *disort* genügt somit den gestellten Anforderungen. Das Strahlungstransfer-Modell *MODTRAN₄* darf mit *libRadtran* ersetzt werden.

IMPLEMENTIERUNG DES ADJACENCY EFFEKTS

Der *Adjacency-Effekt* besagt, dass der Erdbeobachtungssatellit nicht nur die vom Target emittierte Strahlung detektiert, sondern auch die gestreute Strahlung von der Umgebung des Targets. Dieser Effekt resultiert aus der atmosphärischen Streuung, wie der Abbildung 30 zu entnehmen ist. Seine Intensität ist abhängig vom Kontrast zwischen dem Target und seiner Umgebung. Umso höher der Kontrast, desto stärker wirkt sich der Adjacency-Effekt auf die am Sensor detektierte Strahlung aus. Seine Auswirkungen kann man vor allem im Spektrum zwischen 400 und 1000 *nm* beobachten und lassen mit abnehmender Wellenlänge nach. Für ein pragmatisches Modell ist es ausreichend, nur die Umgebung von 1 *km* Radius um das Target zu berücksichtigen. Je weiter der Ursprung der gestreuten Strahlung vom Target entfernt ist, desto niedriger ist sein Anteil an der Strahlung, welche am Sensor detektiert wird [Richter et al., 2006].

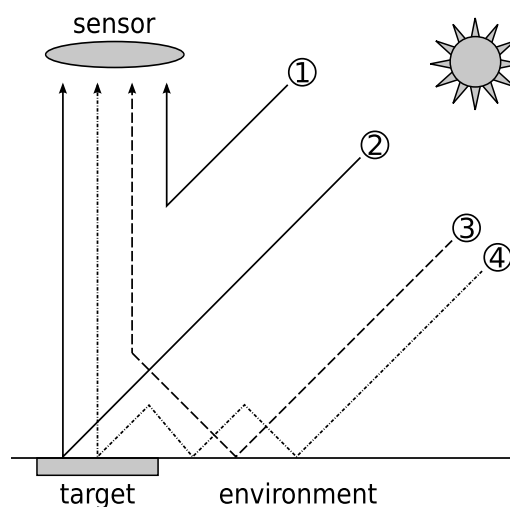


Abbildung 30: Adjacency-Effekt

Die Strahlungsgänge ① und ② der Abbildung 30 werden bereits durch *libRadtran* berücksichtigt, während die Komponenten ③ und ④ den Adjacency-Effekt darstellen und erst in das Modell implementiert werden müssen. Anhand der Abbildung wird auch ersichtlich, dass der Adjacency-Effekt aus zwei Anteilen besteht. Zum einen die von der Umgebung reflektierten Sonnenstrahlung ③, welche ein- oder mehrfach

in der Atmosphäre gestreut wurde und letztlich in das *Field of View* (FOV) des Sensors gelangt. Und zum anderen der mehrfach zwischen Boden und Atmosphäre reflektierten Strahlung ④, welche ebenfalls das FOV des Sensors erreicht. Der Letztere Anteil stellt nur einen vernachlässigbaren kleinen Anteil dar und wird daher in der hier verwendeten analytischen Funktion nicht beachtet. Soll dieser Strahlengang dennoch simuliert werden, empfiehlt es sich, nach der *Monte Carlo Methode* zu verfahren, wodurch die Berechnungsdauer jedoch massiv verlängert wird [Guorui et al., 2008].

Darüber hinaus muss beachtet werden, dass die Implementierung des Adjacency-Effekts unabhängig von dem in Kapitel 5 beschriebenem Vergleich erfolgt. Durch die Berücksichtigung dieses Effekts wird die Genauigkeit des von *libRadtran* verwendeten Modells lediglich verbessert.

6.1 MATHEMATISCHE GRUNDLAGEN

Die Implementierung des Adjacency-Effekts ist auch insofern von Bedeutung, da dieser von *MODTRAN 5* simuliert werden kann. Zwar ist die Berücksichtigung des Adjacency-Effekts mit dem in Verwendung befindlichen *MODTRAN 4* nativ nicht möglich, dennoch ist es wünschenswert diese Funktionalität in *libRadtran* zu integrieren. Bislang wurde der Adjacency-Effekt abteilungsintern mit einem Skript in der Programmiersprache IDL in *MODTRAN 4* eingebunden. Bei der Umsetzung der darin verwendeten mathematischen Grundlagen für *libRadtran* stellte sich heraus, dass das in *MODTRAN 4* verwendete Modell zur Berechnung des Adjacency-Effekts nicht korrekt sein kann. Die verwendeten Gleichungen aus [Verhoef and Bach, 2003] sind bei genauerer Betrachtung zu ungenau. So lautet dort die zur Bestimmung der Radiance Top-of-Atmosphäre zugrunde liegende Gleichung in *MODTRAN 4*:

$$L_{TOA} = L_{ATM} + L_{TAR} + L_{ENV}. \quad (6.1)$$

Dabei ist L_{TOA} die am Satelliten detektierte Radiance, welche sich aus der Radiance zusammensetzt, die von der Atmosphäre (L_{ATM}), dem Target (L_{TAR}) und der Umgebung (L_{ENV}) reflektiert wird. Das Problem liegt in der Ermittlung der Umgebungs-Radiance L_{ENV} . Nach [Verhoef and Bach, 2003] wird diese über die Gleichung

$$L_{ENV} = L_{@alb,Tar} - L_{@alb,Env} - L_{@alb,0} \quad (6.2)$$

bestimmt, wobei die Strahlungstransfergleichung für verschiedene Albedos der Erdoberfläche gelöst wird, um die Summanden zu erhalten. Dies stellt gleichzeitig einen immensen Mehraufwand für die Rechenleistung dar. $L_{@alb,Tar}$ ist die Lösung der Strahlungstransfergleichung nach

der Radiance Top-of-Atmosphäre mit dem Albedo des Targets. Parallel dazu sind $L_{@alb,Env}$ bzw. $L_{@alb,0}$ die Radiance Top-of-Atmosphäre mit dem mittlerem Albedo der Umgebung bzw. einem Albedo von Null. Gleichung 6.2 würde besagen, dass sämtliche von der Umgebung in die Atmosphäre gestreute Strahlung in denselben Strahlengang gelangt, wie die vom Target zum Sensor gestreute Strahlung. Dieser Anteil ist um ein vielfaches zu hoch angesetzt, weswegen die Gleichung 6.1 als zu ungenau bezeichnet werden kann.

Eine ausgereifere Modellierung ist [Richter and Schläpfer, 2011] zu entnehmen. Dabei werden alle Auswirkungen des Adjacency-Effekts auf ein Korrektur-Albedo zurückgeführt. Wird die Strahlungstransfergleichung daraufhin mit dem Korrektur-Albedo ρ_{cor} gelöst, so erhält man die Radiance Top-of-Atmosphäre mit Berücksichtigung des Adjacency-Effekts. Das Korrektur-Albedo wird durch folgende Gleichung ermittelt:

$$\rho_{cor} = \rho_{tar} - q (\rho_{tar} - \bar{\rho}) . \quad (6.3)$$

Dabei ist ρ_{tar} das ursprüngliche Albedo des Targets und $\bar{\rho}$ das mittlere Umgebungs-Albedo. Der Gewichtungsfaktor q ist das Verhältnis zwischen dem diffusen und direkten Transmissionsgrad T_{dif} und T_{dir} . Da zur Ermittlung der Transmissionsgrade weitere Berechnungen mit *libRadtran* notwendig wären, ist es eleganter q auf bereits verfügbare Größen zurückzuführen:

$$q = \frac{T_{dif}}{T_{dir}} = \frac{\frac{E_{dif}}{E_0}}{\frac{E_{dir}}{E_0}} = \frac{E_{dif}}{E_{dir}} \hat{=} \frac{uavgup + uavgdn}{uavgdir} . \quad (6.4)$$

Die Größen *uavgup*, *uavgdn* und *uavgdir* können der *vospev* Output-Datei entnommen werden und sind in Abschnitt 3.1.5 beschrieben. Bei Betrachtung der Radiance Top-of-Atmosphäre ist der Summand *uavgdn* gleich Null, da auf dieser Höhe keine abwärts gerichtete diffuse Strahlung vorhanden ist. Das Verhältnis q reduziert sich im vorliegenden Fall somit auf:

$$q = \frac{uavgup}{uavgdir} . \quad (6.5)$$

Das mittlere Umgebungs-albedo $\bar{\rho}$ kann in einem ersten Ansatz über die Beziehung

$$\bar{\rho}(x, y) = \frac{1}{N^2} \sum_{i,j=1}^N \rho_{tar i,j}(x, y) \quad (6.6)$$

ermittelt werden [Richter and Schläpfer, 2011]. Dazu werden über ein Fenster von $N \times N$ Pixeln Seitenlänge, sämtliche darin enthaltenen Albe-

dos gemittelt. Dieses Vorgehen stellt nur eine grobe Näherung dar. Sinnvoller und genauer ist es, das Umgebungsalbedo zu gewichten. Die empirisch ermittelte Funktion

$$F(r) = 0,9(1 - \exp(-0,84r)) \quad (6.7)$$

beschreibt am besten, wie der Einfluss der diffus gestreuten Strahlung mit dem Abstand r zum Mittelpunkt des Targetpixels exponentiell abnimmt [Tanre et al., 1981]. Um einen Filter zu erhalten, der im Folgenden *Kernel* K genannt wird, muss Gleichung 6.7 differenziert und durch die Fläche eines Ringes mit der Breite GSD dividiert werden:

$$K(r) = \frac{\frac{dF}{dr}}{A_r} = \frac{0,9 \cdot 0,84 \cdot \exp(-0,84r)}{2\pi r \cdot GSD}. \quad (6.8)$$

Dieser Kernel wird mit der Umgebung des Targets gefaltet, wodurch eine Gewichtung der Umgebungs-Pixel erzielt wird. Bei der Ringfläche A_r handelt es sich lediglich um eine Näherung, da der mittlere Kreisumfang $2\pi r$ mit der Breite GSD multipliziert wird. Da im vorliegenden Fall das Verhältnis Breite zu Radius sehr gering ist, kann diese Näherung jedoch als ausreichend erachtet werden. Dieser Zusammenhang wird nochmals mit Abbildung 31 veranschaulicht. GSD steht für *Ground Sampling Distance* und beschreibt die Bodenauflösung des Erdbeobachtungssatelliten in Kilometern. Ebenso muss der Radius r in Kilometern bemessen werden. Damit der Kernel in die Programmiersprache Python umgesetzt werden kann, muss dieser als zweidimensionale Fläche beschrieben werden. Dazu wird der Radius r über den Satz von Pythagoras berechnet als

$$r = \sqrt{x^2 + y^2}, \quad (6.9)$$

wobei x und y die Längen der am rechten Winkel anliegenden Katheten sind. Die Längen werden in Pixeln angegeben und können einen Wert zwischen $-\frac{N}{2}$ und $+\frac{N}{2}$ annehmen. Dabei muss der Kernel eine ungerade Seitenlänge in Pixeln aufweisen, da andernfalls keine Zentrierung auf das Target-Pixel erfolgen kann. Somit kann der Kernel folgendermaßen in Programmiersprache umgesetzt werden:

$$K(x, y) = \frac{0,9 \cdot 0,84 \cdot \exp(-0,84 \cdot \sqrt{x^2 + y^2})}{2\pi r \cdot GSD}. \quad (6.10)$$

Für den Fall $r = 0$ geht der Kernel gegen Unendlich. Nachdem es sich in diesem Fall zugleich um das Target-Pixel handelt, welches nicht zur Gewichtung der Umgebungs-Pixel beitragen soll, muss das mittige Kernel-Pixel Null gesetzt werden:

$$K(0, 0) = 0. \quad (6.11)$$

Abschließend muss die Summe aller Pixel im Kernel normiert werden:

$$K_n(x, y) = \frac{K(x, y)}{\sum K(x, y)}, \quad (6.12)$$

wobei die Summe aller normierten Pixel eines Kernel gleich Eins ist:

$$\sum K_n(x, y) = 1. \quad (6.13)$$

Der normierte Kernel $K_n(x, y)$ kann nun mit der Umgebung des Targets gefaltet werden, woraus ein gewichtetes Umgebungs-Albedo ρ_{env} resultiert, welches das mittlere Umgebungs-Albedo $\bar{\rho}$ aus Gleichung 6.6 ersetzt. Während Abbildung 31 auf der nächsten Seite die beschriebenen Größen nochmals veranschaulicht, wird in Abbildung 32 der normierte Kernel $K_n(x, y)$ mit dazugehörigen Zahlenwerten dreidimensional visualisiert.

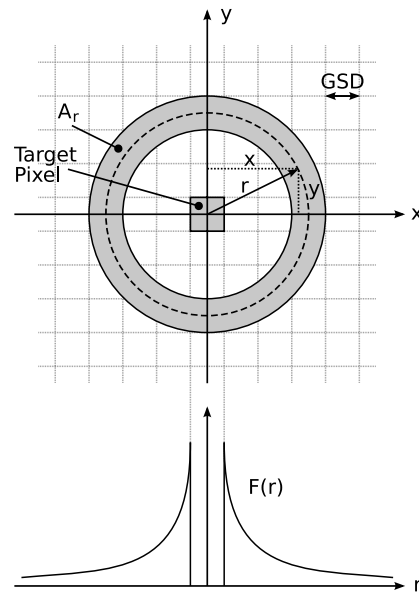


Abbildung 31: zur Beschreibung des Kerns notwendige Größen und dessen exponentieller Verlauf

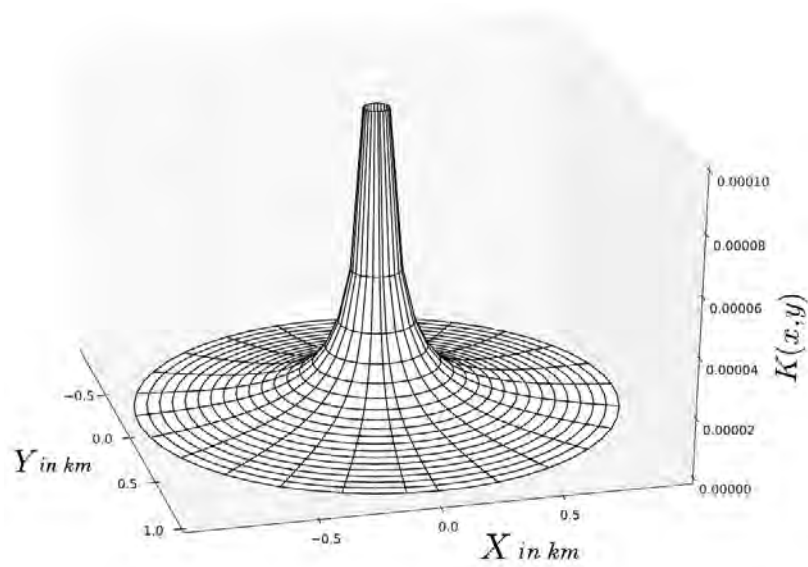


Abbildung 32: dreidimensionale Darstellung des normierten Kerns

6.2 UMSETZUNG IN PYTHON

6.2.1 Albedo-Karten

Die Zielsetzung lautet, eine vorhandene Target-Albedo-Karte $M_{tar}(x, y)$ der Erdoberfläche in eine Radiance-Karte $M_{RAD}(x, y)$ mit den Werten für Top-of-Atmosphäre zu überführen, wobei sämtliche auftretenden Effekte beachtet werden sollen. Jedes Pixel der vorhandenen Target-Albedo-Karte $M_{tar}(x, y)$ steht für ein Target und dessen Albedo $\rho_{tar}(x, y)$. Dies impliziert, dass die Pixel nur einen Wert zwischen Null und Eins annehmen können. Wie im vorherigen Kapitel beschrieben, kann die Radiance Top-of-Atmosphäre über das in Gleichung 6.3 beschriebene Korrektur-Albedo ermittelt werden. Soll die Radiance nicht nur anhand eines einzigen Targets, sondern mit Berücksichtigung des Adjacency-Effekts anhand einer zweidimensionalen Karte ermittelt werden, so ist es zweckmäßig, die Umgebungs-Albedo-Karte $M_{env}(x, y)$ zu generieren. Auch auf dieser Karte steht jedes Pixel für ein Target-Pixel wobei dessen Wert das Umgebungs-Albedo $\rho_{env}(x, y)$ beschreibt, welches das mittlere Umgebungs-Albedo $\rho(\bar{x}, y)$ in Gleichung 6.3 ersetzt.

Die Berechnung des Umgebungs-Albedos $\rho_{env}(x, y)$ anhand der Target-Albedo-Karte $M_{tar}(x, y)$, erfolgt mit dem in Gleichung 6.10 beschriebenen Kernel. Am naheliegensten wäre es, den Kernel auf das Target-Pixel zu zentrieren, die umliegenden Albedowerte mit dem Filter zu gewichten und aus deren Summe das Umgebungs-Albedo $\rho_{env}(x, y)$ zu bestimmen. Folgendes Rechenbeispiel zeigt jedoch, welchen Rechenaufwand ein solches Verfahren mit sich bringen würde. Dazu sei die zu beobachtende Erdoberfläche ein Rechteck mit der Seitenlänge von 3 km bei einer Auflösung (GSD) von 2 m. Die Anzahl der Pixel px_{tar} berechnet sich folglich zu:

$$px_{tar} = \left(\frac{\text{Seitenlänge}_{Bild}}{GSD} \right)^2 = \left(\frac{3000 \text{ m}}{2 \text{ m}} \right)^2 = 2,25 \cdot 10^6. \quad (6.14)$$

Der Kernel K weist eine quadratische Fläche auf und berücksichtigt somit die Anzahl px_{env} an Umgebungs-Pixeln, welche in dem Quadrat mit der Seitenlänge von 1 km liegen:

$$px_{env} = \left(\frac{\text{Seitenlänge}_{Kernel}}{GSD} \right)^2 = \left(\frac{1000 \text{ m}}{2 \text{ m}} \right)^2 = 2,5 \cdot 10^5. \quad (6.15)$$

An dieser Stelle kann bemängelt werden, dass diejenigen Pixel, deren Abstand zum Rand der Karte $M_{tar}(x, y)$ kleiner ist, als der Radius des Kernels, kein korrektes Umgebungs-Albedo zugeordnet wird. Da jenseits der Karte keine Informationen vorliegen, kann der Kernel auch nicht alle zum Adjacency-Effekt beitragenden Pixel berücksichtigen. Dieses Problem wird übergangen, indem die Karte $M_{tar}(x, y)$ aneinander

gekachelt wird, wie die Abbildung 33 veranschaulicht. Es wird davon ausgegangen, dass die in der Karte auftretenden Target-Albedos nicht stark voneinander abweichen und diese Näherung somit keinen relevanten Fehler aufweist. Folglich sind zur Mittelung sämtlicher Umgebungs-Albedos sind m Multiplikationen nötig:

$$m = px_{tar} \cdot px_{env} = 5,625 \cdot 10^{11}. \quad (6.16)$$

Derart viele Multiplikationen würden eine sehr lange Rechendauer erfordern. Deswegen ist es sinnvoller auf ein intelligenteres Verfahren auszuweichen, welches im nächsten Abschnitt vorgestellt werden soll.

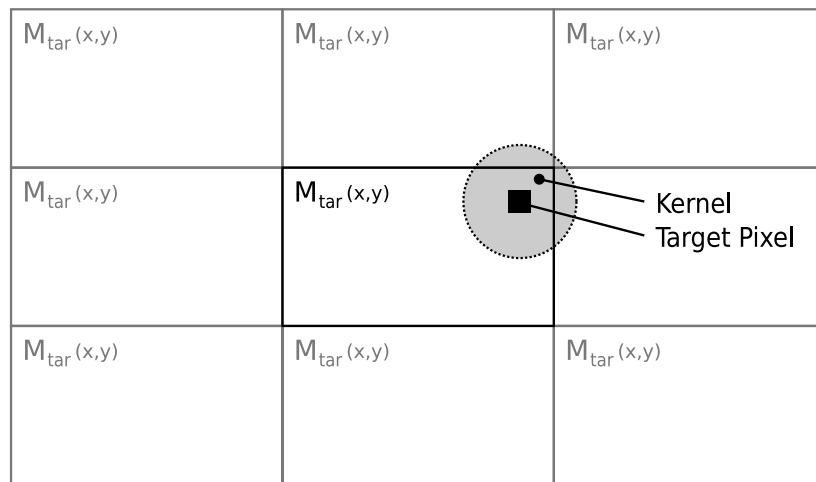


Abbildung 33: Kachelung der Albedo-Karte

6.2.2 Schnelle Fourier-Transformation

Nachdem eine Singulärwertzerlegung [Press et al., 1981] des Kerns aufgrund seiner mangelnden Separierbarkeit und der daraus resultierenden Abweichung bei der Faltung fehlschlug, wird auf die Diskrete Fourier-Transformation zurückgegriffen. Dank des *Konvolution-Theorems* [Press et al., 1981]

$$\mathcal{F}\{g \star h\} = \mathcal{F}\{g\} \cdot \mathcal{F}\{h\} \quad (6.17)$$

kann die Faltung des Kerns mit der Target-Albedo-Karte signifikant beschleunigt werden. Dabei stellen die Funktionen g bzw. h den Kernel $K_n(x, y)$ bzw. die Target-Albedo-Karte $M_{tar}(x, y)$ dar. Wesentliche Aussage des Theorems ist, dass das Produkt der Funktionen g und h , gleich dem Produkt von deren Fourier-Transformierten $\mathcal{F}(g)$ und $\mathcal{F}(h)$ ist. Die Faltungsoperation $g \star h$ im Ortsraum ist folglich gleichbedeutend mit einer Multiplikation im Fourierraum. Dieser Umstand trägt

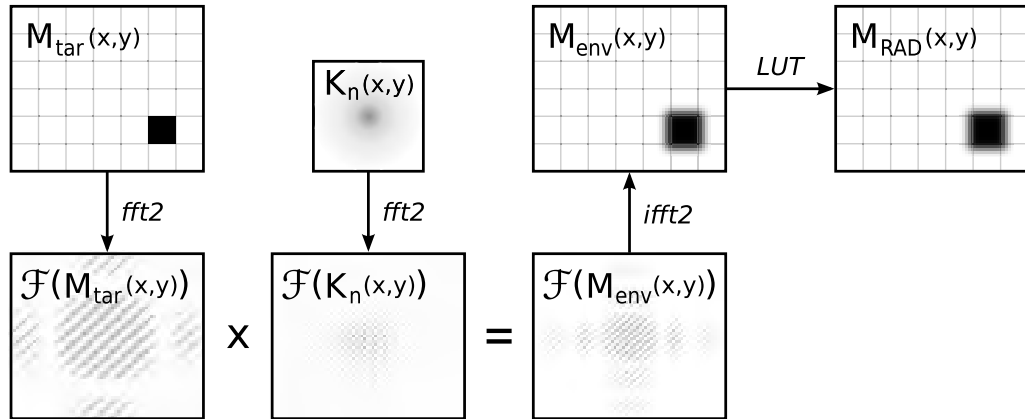


Abbildung 34: Flussdiagramm zur Erzeugung einer Radiance-Karte

entscheidend zu der erwähnten Performance-Verbesserung bei, da die Verarbeitung im Fourierraum schneller vonstatten geht, als die Durchführung der m Multiplikationen bei der Faltung im Ortsraum. Die Bibliothek *NumPy* stellt Algorithmen zur Verfügung, mit welchem die Werte einer *diskreten Fourier-Transformation (DFT)* berechnet werden können. So berechnet die Funktion *fft2* die DFT eines zweidimensionalen Arrays, wobei auf den effizienten Algorithmus der *schnellen Fourier-Transformation (FFT)* zurückgegriffen wird. Die FFT beschleunigt die Berechnung der DFT etwa um den Faktor 10^6 [Press et al., 1981]. Diese Methode wurde bereits im Jahre 1805 von Gauss hergeleitet und 1965 in eine maschinenlesbare Form gebracht [Cooley and John, 1965], wie sie auch hier verwendet wird. Entsprechend kann mit der Funktion *ifft2* die inverse DFT \mathcal{F}^{-1} aus einem zweidimensionalen Array gebildet werden, wobei sich diese Funktion ebenfalls der FFT bedient:

$$c = g \star h = \mathcal{F}^{-1} \{ \mathcal{F} \{g\} \cdot \mathcal{F} \{h\} \} \quad (6.18)$$

Dabei ist c die Umgebungs-Albedo-Karte $M_{env}(x,y)$, welche das Ergebnis der Faltung $g \star h$ ist. Abbildung 34 fasst nochmals zusammen, auf welchem Weg die Radiance-Karte $M_{RAD}(x,y)$ zustande kommt.

6.2.3 Look-Up-Table

Nachdem nun sowohl die Target-Albedo-Karte $M_{tar}(x,y)$ und die Umgebungs-Albedo-Karte $M_{env}(x,y)$ verfügbar ist, kann aus diesen die gesuchte Radiance-Karte $M_{RAD}(x,y)$ ermittelt werden. Dazu wird für jedes Pixel bei (x,y) das Target-Albedo $\rho_{tar}(x,y)$ aus $M_{tar}(x,y)$ und das gewichtete Umgebungs-albedo $\rho_{env}(x,y)$ aus $M_{env}(x,y)$ extrahiert. Zusammen mit dem Gewichtungsfaktor q werden diese Albedo-Werte in

die Gleichung 6.3 eingesetzt, welche nun in folgender Form beschrieben werden kann:

$$\rho_{cor}(x, y) = \rho_{tar}(x, y) - q(\rho_{tar}(x, y) - \rho_{env}(x, y)). \quad (6.19)$$

Um nun die gesuchte Radiance Top-of-Atmosphäre des Target-Pixels bei (x, y) zu erhalten, müsste die Strahlungstransfergleichung mit dem Korrektur-Albedo $\rho_{cor}(x, y)$ gelöst werden. Bedenkt man, dass die zu verarbeitenden Bilddateien in der Regel mehrere 10^6 Pixel aufweisen, müsste das Strahlungstransfermodell dementsprechend oft ausgeführt werden. Die hierzu benötigte Rechendauer würde beim aktuellen Stand der Technik mehrere Monate betragen. Aus diesem Grund wird auf eine *Look-Up-Table*, kurz *LUT* zurückgegriffen, welche die Lösungen für ausgewählte Kombinationen zwischen ρ_{tar} und ρ_{env} beinhaltet. Im vorliegenden Fall dient ein vierdimensionales Array als LUT. Die ersten beiden Dimensionen sind die Achsen für das Target-Albedo $\rho_{tar}(x, y)$ und dem Umgebungs-Albedo $\rho_{env}(x, y)$, wobei sich (x, y) auf das Target-Pixel der Albedo-Karten bezieht. Die dritte Dimension gibt die aus dieser Kombination resultierende Radiance Top-of-Atmosphäre $L_{TOA}(x, y)$ wieder. Eine vierte Dimension ist notwendig um die Radiance Top-of-Atmosphäre über mehrere Wellenlängen darzustellen, wie der Abbildung 35 zu entnehmen ist.

Der große Nutzen einer LUT liegt darin, dass das Strahlungstransfermodell nur noch für alle Kombinationen zwischen ρ_{tar} und ρ_{env} ausgeführt werden muss. Im vorliegenden Fall ist es ausreichend, das Albedo in 11 Werte mit einer Schrittweite von 0,1 zwischen 0 und 1 zu unterteilen. Folglich muss die Strahlungstransfergleichung lediglich

$$11 \cdot 11 = 121$$

mal gelöst werden, was nur wenige Minuten in Anspruch nimmt. Anhand dieser 121 Stützpunkte können alle anderen Albedo-Kombinationen interpoliert werden. Da in Tests festgestellt wurde, dass sich die Radiance nahezu linear zum Albedo verhält, kann eine lineare Interpolation zwischen den 4 naheliegensten Stützpunkten L_1 bis L_4 erfolgen [Jähne, 1989]. Der interpolierte Wert $L_{TOA}(x, y)$ ist die gesuchte Radiance Top-of-Atmosphäre für das Target-Pixel bei (x, y) . Dabei ist die Rechenzeit für dessen Interpolation wesentlich kürzer, als die Lösung des Strahlungstransfermodells mit dem entsprechendem Korrektur-Albedo ρ_{cor} . Bei der Interpolation entsteht zwar ein Fehler, welcher jedoch erfahrungsgemäß gegenüber der Ungenauigkeit des ganzen Strahlungstransfermodells vernachlässigt werden kann. Abbildung 35 veranschaulicht nochmals das vierdimensionale Array der LUT und die Interpolation zwischen den Stützpunkten.

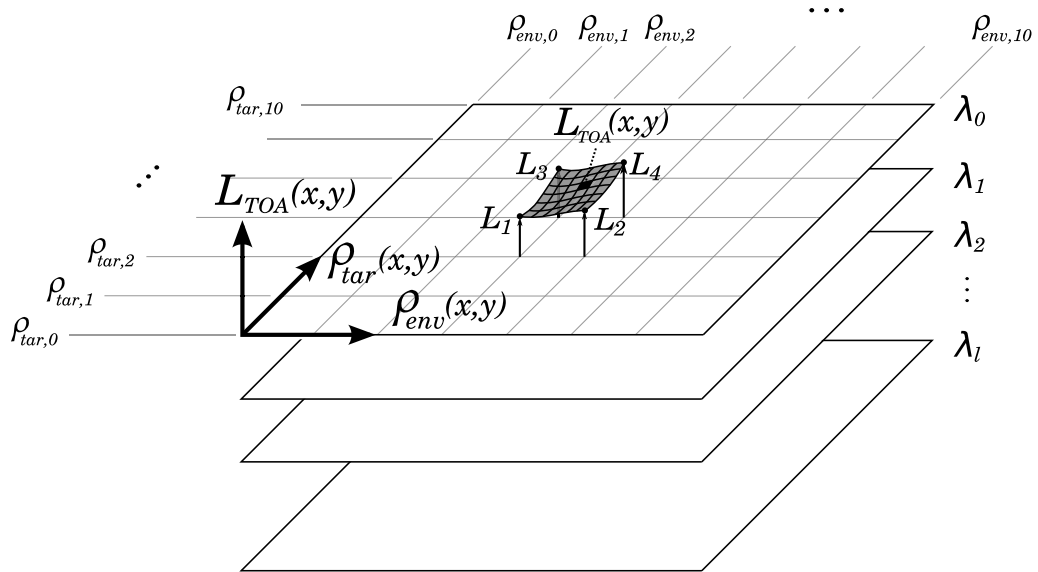


Abbildung 35: Struktur der Look-Up-Table und Interpolation der Radiance Top-of-Atmosphäre $L_{TOA}(x, y)$

6.2.4 Struktur des Skripts

Nachdem es einen unverhältnismäßigen Aufwand darstellen würde, den Quellcode von *libRadtran* oder *uvspec* zu editieren und den Adjacency-Effekt auf diese Weise zu implementieren, ist die naheliegenste Lösung, ein Interface zur vorhandenen Software zu schaffen. In diesem Fall wird die Schnittstelle durch ein Python-Skript realisiert, welches *uvspec* ausführt und dessen In- und Output-Dateien zweckmäßig ändert und ausliest. Die zur Implementierung des Adjacency-Effekts erstellten Funktionen und deren Beziehungen zueinander, werden in einem Software-Flussdiagramm (Abbildung 36) dargestellt und sollen im Folgendem erläutert werden. Die nummerierten Klammern im Text beziehen sich dabei auf die entsprechenden Schritte in der Abbildung.

Ehe mit den Berechnungen begonnen werden kann, muss eine *uvspec* Input-Datei generiert worden sein, welche das Beobachtungsszenario beschreibt und den Anforderungen genügt, wie sie in Abschnitt 3.1.3 beschrieben werden. Diese Datei wird anfangs vom Python-Skript eingelesen (1) und so modifiziert, dass sie den Output generiert, welcher zur Berechnung des Gewichtungsfaktors q nach Gleichung 6.4 benötigt wird. Die modifizierte Input-Datei wird als Text-Datei ausgegeben (2), welche von *uvspec* eingelesen (4) werden kann. Die Ausführung von *uvspec* erfolgt durch einen Aufruf durch das Python-Skript (3). Danach werden die aus der neu generierten *uvspec* Output-Datei (5) benötigten Parameter ausgelesen (6) und daraus der Gewichtungsfaktor q berechnet (7). Dieser wird an eine weitere Funktion weitergereicht (8), welche

das Korrektur-Albedo ρ_{cor} berechnen wird. Für dessen Berechnung nach Gleichung 6.19 ist neben q auch das Target-Albedo ρ_{tar} und Umgebungs-Albedo ρ_{env} nötig. Im vorangegangenen Abschnitt wurde dargelegt, dass für die LUT lediglich eine überschaubare Anzahl von Stützpunkten berechnet werden muss. Deswegen werden die Albedowerte über zwei ineinander verschachtelte Schleifen (9) und (10) an die Funktion (11) übergeben. Somit sind alle Werte vorhanden, um das Korrektur-Albedo zu bestimmen. Wird dieses Albedo ρ_{cor} nun in die originale *uvspec* Input-Datei geschrieben (12), so enthält die daraus generierte Output-Datei (13) die gesuchte Radiance Top-of-Atmosphere L_{TOA} , welche den Adjacency-Effekt berücksichtigt. Diese Radiance wird aus der Output-Datei gelesen und für sämtliche, sich aus den beiden Schleifen (9) und (10) ergebenden Stützpunkte, in die LUT geschrieben (14). Dieses Vorgehen liefert das in Abschnitt 6.2.3 beschriebene, vierdimensionale Array.

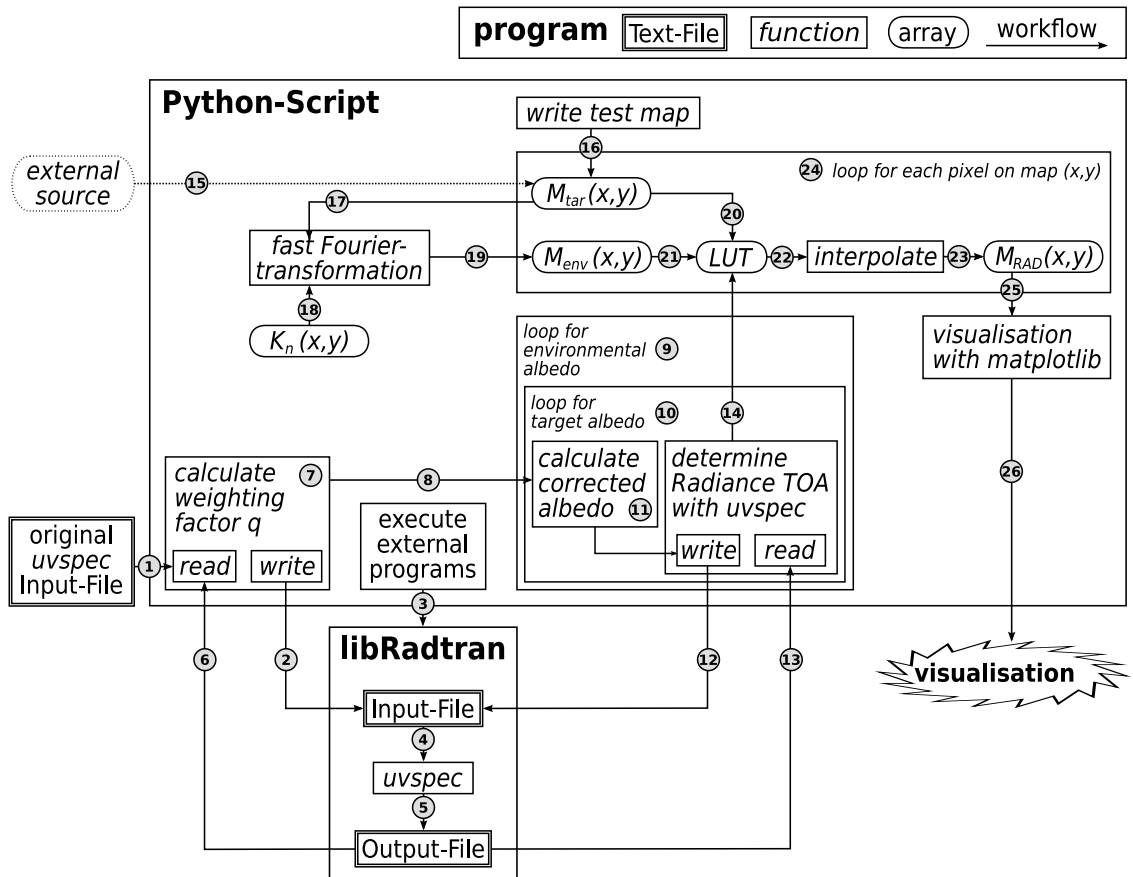


Abbildung 36: Software-Flussdiagramm zur Implementierung des Adjacency-Effekts und Berechnung der Radiance Top-of-Atmosphere $L_{TOA}(x,y)$

Nachdem die LUT vollständig generiert wurde, kann sie auf Albedo-Karten angewendet werden. Ausgangspunkt ist die Target-Albedo-Karte

$M_{tar}(x, y)$, welche entweder aus einer externen Quelle importiert (15) oder zu Testzwecken intern generiert (16) werden kann. Des Weiteren wird der Kernel $K_n(x, y)$ nach Gleichung 6.10 initialisiert. Wird die Target-Albedo-Karte $M_{tar}(x, y)$ (17) mit dem Kernel $K_n(x, y)$ (18) über eine Multiplikation im Fourier-Raum gefaltet, so erhält man über die inverse Fourier-Transformation des Produkts (19), die Umgebungs-Albedo-Karte $M_{env}(x, y)$. Diese Berechnungen erfolgen gemäß der Gleichung 6.18.

Aus den nun zur Verfügung stehendem Target-Albedo (20) und Umgebungs-Albedo (21), kann der LUT die zugehörige Radiance Top-of-Atmosphäre $L_{TOA}(x, y)$ entnommen werden. Dabei wird das Ergebnis durch die in Abschnitt 6.2.3 beschriebene Interpolation ermittelt (22) und in die Radiance-Karte $M_{RAD}(x, y)$ geschrieben (23). Mittels einer Schleife (24) werden die Schritte (20) bis (23) für jedes durch (x, y) beschriebene Pixel auf der Karte durchgeführt. Abschließend wird die daraus resultierende Radiance-Karte $M_{RAD}(x, y)$ mit den durch *matplotlib* bereitgestellten Funktionen visualisiert (25). Die Grafik kann editiert und in einem gängigen Format gespeichert werden (26).

6.3 FUNKTIONSTEST DES SKRIPTS

Anhand eines Tests soll ermittelt werden, ob der Adjacency-Effekt korrekt implementiert wurde. Mittels eines trivialen Boden-Szenarios und der daraus resultierenden Radiance-Map wird überprüft, ob das Python-Skript problemlos ausgeführt wird und ob die verwendeten Gleichungen richtig angewendet werden.

Das triviale Boden-Szenario besteht aus einer Hütte, dessen Dach mit einem Albedo von 0,2 verhältnismäßig "dunkel" ist, also stark absorbierend wirkt. Sie ist von reflektierendem Schnee umgeben, der ein höheres Albedo von 0,5 aufweist. Um die Auswirkung des Adjacency-Effekts besser zu verdeutlichen, habe die Hütte eine Seitenlänge von 50 m. Dieses Szenario, welches auch in Abbildung 37(oben) veranschaulicht wird, dient als Target-Albedo-Karte $M_{tar}(x, y)$ und wird intern vom Python-Skript generiert. Des Weiteren ist der Abbildung ein Querschnitt zu entnehmen, dessen entsprechender Albedowert unterhalb in einem zweiten Diagramm aufgetragen ist. Dabei zeigt der strichlierte Graph das Target-Albedo über den Querschnitt der Karte. Sein rechteckiger und linearer Verlauf verdeutlicht die Idealisierung des Szenarios.

Mit dieser Karte und der *uvspec* Input-Datei, wie sie durch Listing 1 auf Seite 26 beschrieben ist, wird das Python-Skript ausgeführt. Um das Test-Szenario einfach zu halten, beschränkt man sich auf die Simulation von nur einer Wellenlänge mit 500 nm, anstatt wie in der Praxis üblich, über ein Spektrum zu integrieren. Nachdem eine Visualisierung der re-

sultierenden Radiance-Karte $M_{RAD}(x, y)$ keine neuen Erkenntnisse bringen würde, da deren Grauverlauf nahezu identisch zu der anfänglichen Target-Albedo-Karte ist und sich nur in den Zahlenwerten unterscheidet, wird stattdessen nur deren Querschnitt betrachtet. Dieser wird als gepunkteter Graph im bestehendem Diagramm Abbildung 37(unten) dargestellt. Um eine bessere Vergleichbarkeit mit dem ursprünglichen Target-Albedo zu erzielen, wurde die Radiance so normiert, dass deren Maximum mit dem Maximum des Target-Albedos übereinstimmt. Aufgrund der direkten Proportionalität zwischen Korrektur-Albedo und Radiance, führt diese Normierung gleichzeitig zu einem Graphen, welcher identisch mit dem Korrektur-Albedo ist:

$$\rho_{cor}(x, y) = L_{TOA, Norm}(x, y) .$$

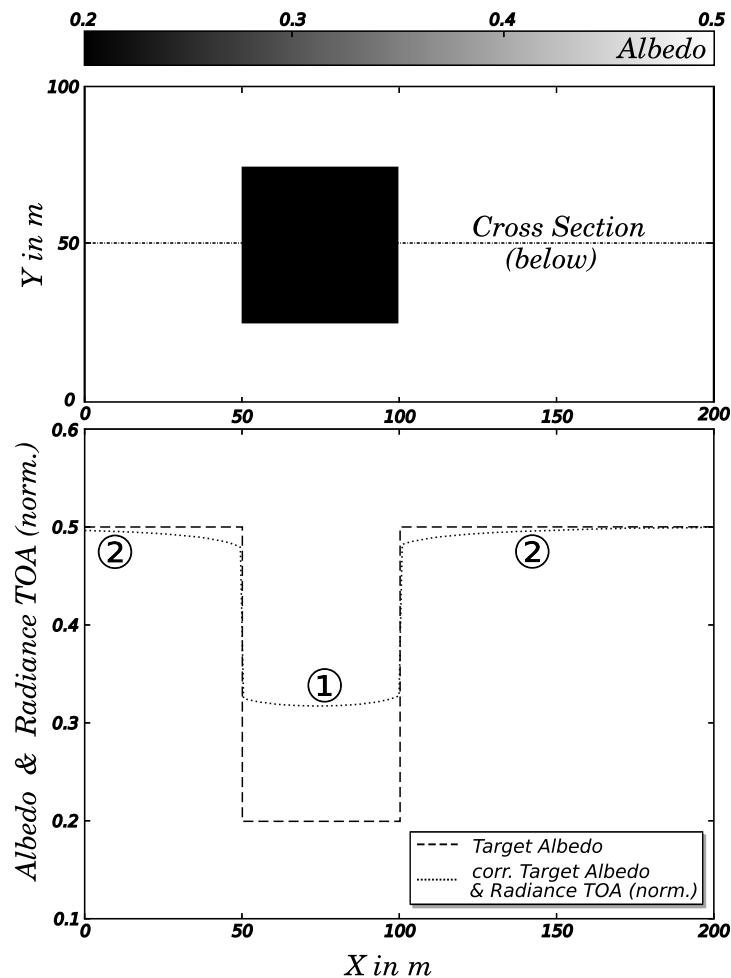


Abbildung 37: Wirkung des Adjacency-Effekts am Beispiel der "Hütte im Schnee"

Der Verlauf des Korrektur-Albedos bzw. der normierten Radiance Top-of-Atmosphäre ist für den Test von zentraler Bedeutung. Dieser soll die Korrektheit der Gleichung

$$\rho_{cor} = \rho_{tar} - q (\rho_{tar} - \bar{\rho})$$

bestätigen, welche bereits bei den Grundlagen in Abschnitt 6.1 eingeführt wurde (vgl. Gleichung 6.3). So kann an ① abgelesen werden, dass das korrigierte Albedo der "Hütte" wesentlich höher ist als deren ursprüngliches Target-Albedo. Unter der Berücksichtigung, dass der Kernel zur Gewichtung des Umgebungs-Albedos einen mehrfach größeren Durchmesser als die Hütte hat, wird der Betrag des gewichteten Umgebungs-Albedos wesentlich vom Albedo außerhalb der Hüttenfläche beeinflusst. Der rechte Summand $q (\rho_{tar} - \bar{\rho})$ wird im vorliegenden Szenario negativ, und summiert sich folglich, unter Beachtung seines negativen Vorzeichens, zu dem linken Summanden ρ_{tar} auf, was zu einem erhöhten Korrektur-Albedo ρ_{cor} führt. Gleichzeitig ist der gekrümmte Verlauf bei ① damit zu erklären, dass die Pixel, welche tiefer im dunklen Bereich der Hütte liegen, weniger von der Umgebung, also dem Schnee mit hohem Albedo, beeinflusst werden. Umso "tiefer" die Pixel im Albedo-schwachen Bereich liegen, desto kleiner wird das Umgebungs-Albedo $\bar{\rho}$, was zu einer größerem Betrag von $q (\rho_{tar} - \bar{\rho})$ führt und letztlich ein niedrigeres Korrektur-Albedo ρ_{cor} mit sich bringt. Umgekehrt verhält es sich bei ②. Je weiter ein Pixel von der Hütte entfernt ist, desto weniger wird das niedrige Albedo der Hütte durch den exponentiell abfallenden Kernel gewichtet. Die Differenz $(\rho_{tar} - \bar{\rho})$ zwischen dem Target- und Umgebungs-Albedo wird mit zunehmenden Abstand zur Hütte immer geringer, was in einem Korrektur-Albedo resultiert, welches zunehmend mit dem Target-Albedo übereinstimmt. Liegt die Hütte endgültig außerhalb des Kerns, so ist das Korrektur-Albedo ρ_{cor} gleich dem Target-Albedo ρ_{tar} .

Dieses Verhalten bestätigt die Funktionalität der zugrunde liegenden Gleichung 6.19 und des durch Gleichung 6.10 beschriebenen Kerns. Des Weiteren kann die fehlerfreie Ausführung des Skriptes und die Visualisierung der Ergebnisse samt deren Individualisierung gewährleistet werden.

6.4 VERIFIZIERUNG DER VERWENDETEN GLEICHUNGEN

Während der vorherige Test aus Abschnitt 6.3 lediglich bestätigen kann, dass der Adjacency-Effekt erfolgreich implementiert wurde, ist es wesentlich komplexer dessen Richtigkeit oder Genauigkeit zu überprüfen. Für die Verifizierung des Modells und vor allem des verwendeten Kerns bedarf es der Lösung eines *inversen Problems*, bei welchem die *Wirkung* bekannt, die *Ursache* jedoch unbekannt ist. Für die vorliegende

Problemstellung bedeutet dies, dass die gemessene Wirkung, also die vom Satelliten beobachtete Radiance L_{TOA} , auf die aus der Erdoberfläche resultierende Ursache zurückgeführt werden muss [Starks and V., 1998]. In der Praxis wird diese Problematik durch die sogenannte *Vicarious Calibration*¹ angegangen [Ryan, 2007]. Dabei wird die tatsächliche Beschaffenheit der Erdoberfläche vermessen und die daraus gewonnenen Informationen zur Kalibrierung der Satellitenaufnahmen genutzt. Dazu sind spezielle Strahlungsmessungen im Gelände erforderlich, die entweder direkt am Boden oder von entsprechend ausgestatteten Flugzeugen durchgeführt werden. Der *gerichtete Reflexionsgrad* und somit auch das Albedo lässt sich in Abhängigkeit vom Zenitwinkel der Sonne und dem Zenitwinkel der Beobachtung sowie der Azimutdifferenz zwischen diesen Richtungen beschreiben (siehe Abbildung 11 auf Seite 27). Die hierzu erforderlichen Größen können mit Vorrichtungen wie einem *Goniometer* gemessen werden. Das in Abbildung 38a dargestellte Feld-Goniometer kann im Abstand von 2 m aus beliebigen Richtungen auf die Messfläche gerichtet werden [Alberz, 2001].

Wie Abbildung 38b vermittelt, werden als Alternative zu natürlichem Terrain auch kalibrierte Abdeckplanen verwendet. Anhand der bekannten Eigenschaften dieser Ziele, können hochauflösende Satellitenaufnahmen anhand der Planen kalibriert werden. Voraussetzung ist jedoch, dass die Abdeckplanen groß genug sind, damit sie vom Sensor des Satelliten detektiert werden [NCAVEO, 2012].

Auf diese oder ähnliche Methoden werden vollständige Landstriche über den ganzen Globus verteilt kartographiert. Heute existieren auf jedem Kontinent *Vicarious Calibration Sites* mit unterschiedlichen Ausmaßen, Höhenlagen und klimatischen wie geologischen Bedingungen.



(a) transportables *Feld-Goniometer* im Einsatz [Alberz, 2001]



(b) Abdeckplanen zur Kalibrierung einer Satellitenaufnahme [NCAVEO, 2012]

Abbildung 38: Methoden der Strahlungsmessung im Gelände

¹ zu deutsch: *stellvertretende Kalibrierung*

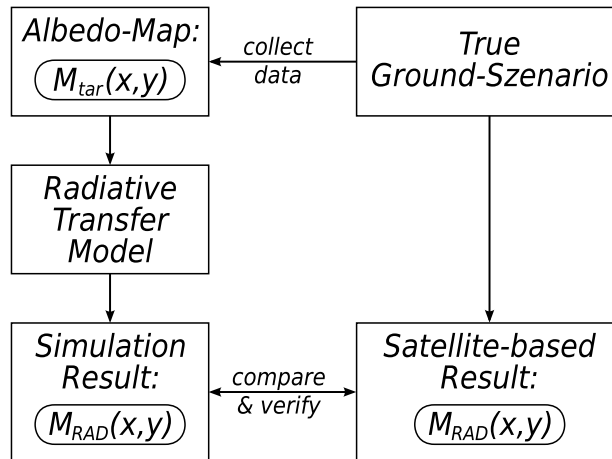


Abbildung 39: Schema der *Vicarious Calibration* zur Verifizierung des Strahlungstransfermodells samt Adjacency-Effekt

Soll das vorliegende Modell des Adjacency-Effekts verifiziert werden, so bedarf es zweierlei Dinge. Zum Einen wird die Albedo-Karte eines realen Bodenszenarios benötigt, wie sie Dank einer exakt vermessenen *Vicarious Calibration Site* zur Verfügung gestellt werden kann. Zweitens bedarf es einer unkalibrierten Satellitenaufnahme von exakt jenem realen Bodenszenario, welche Auskunft über die Top-of-Atmosphere detektierte Radiance L_{TOA} gibt. Stehen diese beiden Karten zur Verfügung, so kann die reale Albedo-Karte als Input an das Strahlungstransfermodell übergeben werden. Die weiteren Eingabeparameter müssen so gewählt werden, dass sie das bestehende Szenario bei der Satellitenaufnahme rekonstruieren. Das auf diese Weise generierte Ergebnis der Simulation wird schließlich mit der bestehenden Satellitenaufnahme verglichen werden. Die zur Verifizierung durchzuführende Prozedur wird in Abbildung 39 zusammengefasst. Die Differenz beider Karten erteilt Auskunft über Genauigkeit des Modells samt implementieren Adjacency-Effekt.

In diesem Zusammenhang muss erwähnt werden, dass der Betrieb auf die von Instituten zur Verfügung gestellten Modellen und Gleichungen vertraut. Es ist nicht die Intention des Betriebs diese Modelle und Gleichungen zu verifizieren. Denn selbst eine nicht perfekte Modellierung des Adjacency-Effekts ist besser als die Umgebungsreflexion überhaupt nicht zu berücksichtigen.

ZUGRIFF AUF MOLEKULAR-DATENBANK HITRAN

Die Vorteile einer *line-by-line*-Simulation, welche durch die molekulare Absorptionsdatenbank *HITRAN* ermöglicht wird, wurden bereits in den Abschnitten 2.3 und 4.5 dargelegt. In diesem Kapitel soll aufgezeigt werden, **wie** die dazu erforderlichen Spektralparameter so für *libRadtran* aufbereitet werden, dass eine Simulation mit Berücksichtigung sämtlicher Spektrallinien erfolgen kann.

7.1 BESCHREIBUNG DER EINZELNEN ARBEITSSCHRITTE

Die Beziehungen zwischen den Programmen und die einzelnen Arbeitsschritte, welche für die Verarbeitung der molekularen Datenbank benötigt werden, sind in einem Software-Flussdiagramm (Abbildung 40) veranschaulicht.

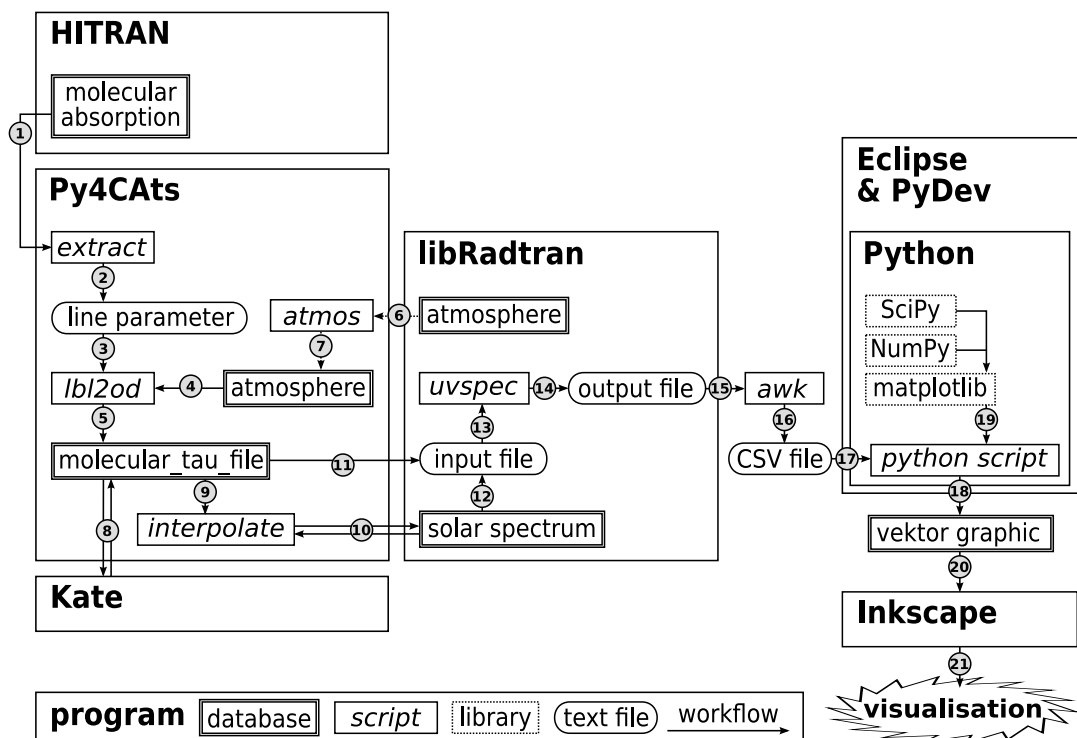


Abbildung 40: Software-Flussdiagramm zur Berechnung der Radiance Top-of-Atmosphäre unter Berücksichtigung der molekularen Absorption mit *HITRAN*

Als erstes müssen mit dem Python-Skript *extract* die benötigten Spektralparameter aus *HITRAN* extrahiert (1) und in eine temporäre Datenbank geschrieben werden (2). Der hierfür benötigte Kommandozeilenbefehl wird in Abschnitt 4.6.1 beschrieben. Die extrahierten Spektralparameter werden von *lbl2od* eingelesen (3) und zusammen mit der Atmosphärenbeschreibung (4) in eine *molecular_tau_file* gewandelt (5), welche die optische Dicke für das betrachtete Spektrum enthält. Soll nicht die in *Py4CATS* enthaltene Atmosphärenbeschreibung verwendet werden, sondern die von *libRadtran* bereitgestellte (6), kann diese optional mit dem Skript *atmos* in das *xy*-Format konvertiert werden (7). Die Handhabung der Skripte *lbl2od* und *atmos* werden in Abschnitt 4.6.2 beschrieben.

Ehe die so generierte *molecular_tau_file* in der *uvspec* Input-Datei referenziert werden kann, ist noch eine kleine Anpassung in deren Struktur notwendig. Die mit einem Texteditor durchzuführende Modifikation (8) wird in Abschnitt 4.6.3 beschrieben.

Daraufhin muss eine in Abschnitt 7.1.1 vorgestellte Interpolation des extraterrestrischen Sonnenspektrums erfolgen. Dazu werden die in der *molecular_tau_file* verwendeten Wellenlängen von einem Skript in Python ausgelesen (9) und die dazugehörige extraterrestrische Irradiance aus der von *libRadtran* verwendeten *solar_file* interpoliert (10). Das in der *solar_file* beschriebene extraterrestrische Spektrum wird dadurch so erweitert (10), dass *uvspec* für sämtliche auftretenden Spektrallinien aus der *molecular_tau_file* (11) die dazugehörige extraterrestrische Irradiance auslesen kann (12).

Nachdem alle benötigten Datenbanken bereitgestellt bzw. aufbereitet wurden, kann *uvspec* mit der Input-Datei ausgeführt werden (13) und die dazugehörige Output-Datei generieren (14). Die berechneten Radiance-Werte werden, wie in Abschnitt 4.3 beschrieben, mit *awk* aus der Output-Datei gelesen (15) und in eine CSV-Datei geschrieben (16). Diese kann wie gehabt von einem Python-Skript ausgelesen werden (17) um sie in einer Vektor-Grafik auszugeben (18). Für das Funktionieren des Skriptes wird wiederum auf die in Abschnitt 4.1 beschriebenen Bibliotheken zurückgegriffen (19). Um eine bessere Vergleichbarkeit der von *matplotlib* generierten Grafik zu erzielen, wird diese mit *Inkscape* geöffnet (20) und den aus Abschnitt 5.3 bekannten Ergebnissen gegenübergestellt (21). Die Auswertung des resultierenden Graphen erfolgt im Abschnitt 7.2.

7.1.1 Interpolation des Sonnenspektrums

Wie bereits erwähnt, muss das von *uvspec* verwendete extraterrestrische Sonnenspektrum für sämtliche von *HITRAN* berechneten Spektrallinien

interpoliert werden. Dieser Umstand resultiert daraus, dass *uvspec* nur den Strahlengang jener Wellenlängen zu simulieren vermag, deren extraterrestrische Irradiance es kennt. Dabei offenbart sich, dass die in *libRadtran* bereitgestellten Sonnenspektren mit einer Auflösung von $0,1 \text{ nm}$ über keine hinreichende Auflösung verfügen, um *HITRAN*'s Spektrallinien mit einer Auflösung von 10^{-6} nm zu simulieren. Auch ein alternativ verwendetes, mit 10^{-3} nm spektral hochauflösendes Sonnenspektrum [Chance and Kurucz, 2010] reicht noch nicht aus, um die von *HITRAN* gegebenen Absorptionslinien zu berechnen. Dabei stellt die Auflösung von 10^{-3} nm des Sonnenspektrums nicht das Problem dar. Diese Auflösung ist völlig ausreichend, da die auf der Erde ankommende Sonnenstrahlung auf Grund der *Dopplerverbreiterung* eine wesentlich größere Spektrallinienbreite aufweist, als die von *HITRAN* gegebenen Absorptionslinien der Erde. Weitere Erläuterungen zum Dopplereffekt und der daraus resultierenden Verbreiterung von Spektrallinien sind dem Abschnitt 9.2 im Anhang zu entnehmen.

Das eigentliche Problem besteht darin, dass *uvspec* die Irradiance jener extraterrestrischen Sonnenstrahlung benötigt, welche sich den präzisen Spektrallinien von *HITRAN* in der *molecular_tau_file* zuordnen lässt. Nachdem wegen der Dopplerverbreiterung eine ausreichend genaues Sonnenspektrum zur Verfügung gestellt werden kann, ist es legitim dieses Spektrum zu interpolieren. Das in Listing 15 gezeigt Skript liest die Wellenlängen der von *HITRAN* generierten Spektrallinien und generiert anhand einer in der *uvspec* Input-Datei referenzierten *solar_file*, eine neue *solar_file* mit den interpolierten Werten.

Listing 15: Python-Skript zur Interpolation der extraterrestrischen Irradiance des Sonnenspektrums für die Wellenlängen aus der *molecular_tau_file*

```
1 # import required libraries:
2 import matplotlib.mlab as mlab
3 import numpy as np
4 import re
5
6 # read preprocessed CSV into arrays:
7 solar_file = mlab.csv2rec('/home/MIK/solar_flux.dat', delimiter=' ')
8 mol_tau = mlab.csv2rec('/home/MIK/optDepth_WL.txt', delimiter=',')
9
10 # split solar_file into single arrays for better interpolation:
11 wl_and_irr = np.empty([len(solar_file),2])
12 for a in range(len(solar_file)):
13     solar_file_temp = re.split('\s',str(solar_file[a]))
14     solar_file_temp[0] = re.sub('([,]',',',solar_file_temp[0])
```

```

15     solar_file_temp[1] = re.sub('[,]', '', solar_file_temp[1])
16     wl_and_irr[a,0] = float(solar_file_temp[0])
17     wl_and_irr[a,1] = float(solar_file_temp[1])
18
19     # correct format of wavelengths from molecular_tau_file:
20     mol_tau_cor = np.empty(len(mol_tau))
21     for a in range(len(mol_tau)):
22         mol_tau_temp = str(mol_tau[a])
23         mol_tau_temp = re.sub('[,]', '', mol_tau_temp)
24         mol_tau_temp = re.sub('[()]', '', mol_tau_temp)
25         mol_tau_cor[a] = float(mol_tau_temp)
26
27     # interpolate Irradiance for wavelengths in molecular_tau_file
28     # from values in solar_file:
29     interp = np.interp(mol_tau_cor, wl_and_irr[:,0], wl_and_irr[:,1])
30
31     # create new array with wavelengths from molecular_tau_file
32     # and corresponding interpolated irradiance:
33     irr_intp = np.empty([len(mol_tau_cor), 2])
34     for a in range(len(mol_tau_cor)):
35         irr_intp[a,0] = mol_tau_cor[a]
36         irr_intp[a,1] = interp[a]
37
38     # write new solar_file which can be referenced
39     # from uvspec input file:
40     new_file = "/home/MIK/interpolated_solar_flux.dat"
41     new_solar_file = open(new_file, "a") # 'a': append to existing file.
42     for a in range(len(mol_tau_cor)):
43         line = str(irr_intp[a,0]) + ' ' + str(irr_intp[a,1]) + 'e+00\n'
44         new_solar_file.write(line)
45     new_solar_file.close()

```

Zuerst erfolgt der Import aller im Skript benötigten Bibliotheken (Zeile 2-4). Daraufhin werden die `solar_file` und `molecular_tau_file` eingelesen (Zeile 7-8), wobei diese zuvor in das Format einer CSV-Datei gebracht werden müssen. Das in [Chance and Kurucz, 2010] bereitgestellte extraterrestische Sonnenspektrum wird zum Beispiel in einer *Excel*-Tabelle zur Verfügung gestellt, weswegen man die benötigten Wellenlängen und die dazugehörige Irradiance einfach als CSV-Datei abspeichern kann. Die zu interpolierenden Wellenlängen sind der von *HITRAN* generierten `molecular_tau_file` zu entnehmen. Diese werden einfach mit dem Unix-Tool *awk* extrahiert und in eine CSV-Datei gespeichert. Der hierfür nötige Kommandozeilenaufruf lautet

```
awk 'NF==50{print $1}' optDepth.txt > optDepth_WL.txt
```



```

19 | phi0 0                               # solar azimuth
20 | # calculate solar zenith angle sza by latitude, longitude and time:
21 | latitude 0
22 | longitude 0
23 | time 2012 06 28 12 00 00

```

Dabei ist in Zeile 4 das interpolierte extraterrestrische Sonnenspektrum referenziert, welches mit dem in Abschnitt 7.1.1 dargelegten Python-Skript generiert wurde. Des Weiteren ist in Zeile 6 jene Datei angegeben, welche die mit *Py4CATS* aus *HITRAN* extrahierten und in die optische Dicke überführten Spektralparameter enthält (siehe Abschnitt 4.6).

Besondere Aufmerksamkeit ist der Wahl des zu betrachtenden Wellenlängenbereichs in Zeile 11 zu schenken. *uvspec* unterstützt lediglich eine Auflösung bis zu 10^{-3} nm , weswegen hier auch keine genauere Angabe erfolgen darf. Dabei muss der zu betrachtende Wellenlängenbereich innerhalb des in der *solar_file* beschriebenen extraterrestrischen Sonnenspektrums liegen. Sollten dabei die in Zeile 11 angegebenen Grenzen mit einer Auflösung von 10^{-3} nm nicht in der *solar_file* definiert sein, so müssen diese Wellenlängen und die dazugehörige Irradiance manuell mit einem Text-Editor in der *solar_file* nachgetragen werden.

7.2 VERGLEICH *line-by-line*-SIMULATION MIT *bändermodell*

Für den Vergleich einer *line-by-line*-Simulation mit einem Bändermodell, soll ein bereits bekannter Wellenlängenbereich betrachtet werden. Der Bereich zwischen 570 und 630 *nm*, welcher bereits in Abschnitt 5.3 eine Diskrepanz zwischen *MODTRAN₄* und *libRadtran* offenbarte (siehe Abbildung 29, Ausschnitt 1), soll nochmals unter Berücksichtigung der *HITRAN*-Datenbank untersucht werden. Das Python-Skript, welches diese drei Datensätzen aneinander anpasst und diese mit Abbildung 41 einander gegenüberstellt, wird in Abschnitt 4.4 erklärt.

Um die Zuordnung des zu betrachtenden Spektrums zu erleichtern, zeigt Abbildung 41 links oben eine Übersicht mit dem bereits aus Abschnitt 4.4 vertrautem Wellenlängenbereich. Zudem verdeutlicht diese Übersicht, welche beiden Ausschnitte im Detail betrachtet werden. Ausschnitt 1 zeigt nochmals den Bereich zwischen 588 und 601 *nm*, in welchem der Unterschied zwischen *MODTRAN₄* (blau) und *libRadtran* (grün) bis zu 17% beträgt. Vergleicht man diese Bändermodelle mit der *line-by-line*-Simulation (rot), so offenbart deren Verlauf die beachtliche, durch *HITRAN* ermöglichte spektrale Auflösung. Ihr sprunghafter Verlauf ergibt sich aus den in *HITRAN* hinterlegten Spektralparametern. Beson-

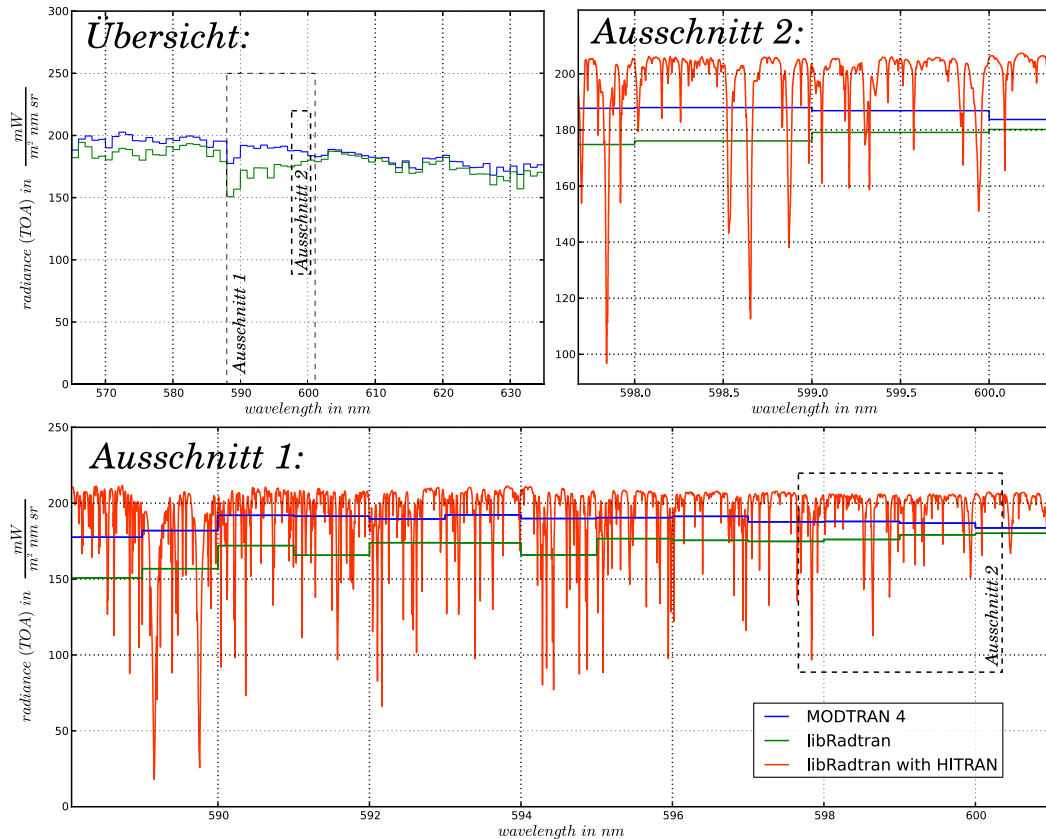


Abbildung 41

ders “tiefe” Ausreißer veranschaulichen jene Spektrallinien mit einer starken molekularen Absorption.

Ausschnitt 2 vergrößert den Bereich zwischen 597,5 und 600,5 nm und verdeutlicht noch offensichtlicher die extrem hohe Auflösung. Aus praktischen Erfahrungswerten der Abteilung OPTICAL DESIGN & PERFORMANCE innerhalb des betreuenden Betriebs geht hervor, dass diese Auflösung mehr als ausreichend ist. Darüber hinaus konnte ein derart hoher Informationsgehalt bislang nicht abteilungsintern gewonnen werden, weswegen die freie Nutzung der *HITRAN*-Datenbank einen Fortschritt für den Betrieb bedeutet.

Wie bereits in Abschnitt 6.4 dargelegt, handelt es sich hier um die Ergebnisse einer Simulation. Um deren Wahrheitsgehalt zu bestimmen, benötigt es entweder einen Abgleich mit bereits verifizierten Ergebnissen einer anderen Simulation oder einer Gegenüberstellung mit satellitenbasierten Radiance-Werten im Rahmen einer *Vicarious Calibration*.

AUSBLICK

Nachdem diese Arbeit die Tauglichkeit des Strahlungstransfermodells *libRadtran* bewiesen hat und die Verwendung der molekularen Absorptionsdatenbank *HITRAN* ermöglicht, macht es Sinn dessen Funktionsumfang den Anforderungen entsprechend zu erweitern.

Zum Beispiel bietet das bestehende Python-Skript zur Berechnung des Adjacency-Effekts Potential zur weiteren Optimierung. So wäre es möglich, die rechenintensive Interpolation der Look-Up-Table für einen Multi-Kern Prozessor anzupassen. Auf der zur Verfügung stehenden Workstation könnte die Berechnung somit um den Faktor 16 beschleunigt werden. Der einhergehenden Performance-Gewinn könnte eine erhöhte Auflösung der Look-Up-Table investiert werden. Dies erhöht zwar einerseits wieder die Rechenzeit, erhöht aber andererseits auch die Genauigkeit der zu berechnenden Radiance. Es würde sich anbieten, die Auflösung der Look-Up-Table dynamisch zu gestalten, sodass die Umgebung jener häufiger auftretende Albedo-Kombinationen eine höhere Auflösung erfahren. Zudem werden bislang identische mehrmalig auftretende Albedo-Kombination jedes mal von neuem interpoliert. Auch dies stellt einen unnötigen Mehraufwand dar und kann durch eine Erweiterung des Python-Skripts unterbunden werden.

Des Weiteren würde eine an *MODO* angelehnte graphische Benutzeroberfläche die Benutzerfreundlichkeit wesentlich verbessern. Zum Beispiel könnte die Berechnung einer Radiance-Karte anhand einer Albedo-Karte unter Berücksichtigung des Adjacency-Effekts und der Datenbank *HITRAN* vollständig über das GUI erfolgen. Neben den bereits aus *MODO* bekannten Funktionalitäten könnten komplette Karten visualisiert werden. Dies würde auch ein Verständnis der Programmiersprache Python und der *uvspec* In- und Output-Datei überflüssig machen. Dabei können die in der *uvspec* Input-Datei vorgenommen Eingaben auf ihre Plausibilität überprüft werden und den Nutzer bei falschen Eingaben warnen. Ein weiteres sinnvolles Feature wäre es, eine berechnete Radiance-Karte auch über die Wellenlänge darstellen zu können. Dazu wird auf der Karte, welche ja lediglich eine Wellenlänge visualisieren kann, ein beliebiges Pixel angewählt und daraufhin dessen Radiance über die Wellenlänge in einem separaten Fenster dargestellt.

Wünschenswert wäre abschließend die Verifizierung der verwendeten Modelle mittels einer *Vicarious Calibration* oder dem Vergleich mit bereits verifizierten Daten.

Teil III

APPENDIX

ANHANG

9.1 INSTALLATIONSPROBLEM BEI *libradtran*

Unter *Ubuntu* und seinen Derivaten wie *Kubuntu*, können bei der Kompilierung von *libRadtran* Fehler auftreten. Dies äußert sich durch folgende Fehlermeldung:

```
/usr/lib/gcc/i686-linux-gnu/4.6.1/.../lib/libgsl.so:  
undefined reference to 'cblas_dasum'
```

Der Grund für diesen Fehler liegt an der Reihenfolge, in welcher die internen Bibliotheken an den Linker übergeben werden. Dieses Problem tritt erst bei jüngeren Versionen von *Ubuntu* auf, da in den älteren Versionen die Reihenfolge noch nicht von Bedeutung war. Der Programmfehler lässt sich durch das Editieren der Datei `configure.in` beseitigen. Dort muss die Zeile

```
LIBGSL="-lgslcblas -lgsl -lm"
```

in folgende Form umgeschrieben werden:

```
LIBGSL="-lgsl -lgslcblas -lm"
```

Danach muss anhand der Datei `configure.in` eine neue Konfigurationsdatei erstellt werden, was durch den Kommandozeilenbefehl

```
autoconf configure.in
```

geschieht. Das Programm *autoconf* wird üblicherweise nicht zusammen mit dem Betriebssystem geliefert. Daher muss es erst über ein Software-Verwaltungsprogramm wie *Apper* oder *Muon* installiert werden. Nun kann die Konfiguration-Datei über den Befehl

```
./configure
```

ausgeführt werden [libRadtran, 2012b]. Der weitere Installationsverlauf ist identisch zu der Prozedur, wie sie der mitgelieferten Datei `INSTALL` zu entnehmen ist.

9.2 DOPPLERVERBREITERUNG

Unter der Dopplerverbreiterung versteht man die Verbreiterung von Spektrallinien wegen dem Dopplereffekt, welcher durch die Geschwindigkeit der Moleküle hervorgerufen wird. Die Maxwell-Boltzmann-Verteilung der Geschwindigkeiten resultiert in eine Dopplerverbreiterung $D(\nu)$ jeder emittierten oder absorbierten monochromatischer Frequenz, gemäß der Verteilung

$$D(\nu) = \frac{1}{\sqrt{\pi}\Delta\nu_D} \exp\left[-\left(\frac{\nu - \nu_D}{\Delta\nu_D}\right)^2\right]. \quad (9.1)$$

Dabei ist die Dopplerbreite $\Delta\nu_D$ einer Spektrallinie gegeben durch

$$\Delta\nu_D = \frac{\nu_0}{c} \sqrt{\frac{2kT}{m_A}}, \quad (9.2)$$

wobei m_A die Masse des betrachteten Atmos und ν_0 die Zentralfrequenz ist. Nachdem typische Dopplergeschwindigkeiten mehrere km/s aufweisen, beträgt die resultierende Halbwertbreite einige $10^{-3} nm$ [Baudis, 23. November 2006].

LITERATURVERZEICHNIS

- Jörg Alberz. *Einführung in die Fernerkundung*. Wissenschaftliche Buchgesellschaft, Berlin, Deutschland, 2. Auflage, 2001. (Zitiert auf Seiten 5, 7, 8, 37 und 86.)
- Heinz Niedrig. *Physik*. Springer-Verlag, Berlin Heidelberg, 1992. ISBN 3-540-54470-4. (Zitiert auf Seite 6.)
- Eugene Hecht. *Optik*. Oldenburg Wissenschaftsverlag, München, 2005. ISBN 978-3486273595. (Zitiert auf Seiten 6 und 35.)
- Wikipedia. *Brechungsindex*. Aufgerufen am 30. Mai 2012a. URL <http://de.wikipedia.org/wiki/Brechungsindex>. (Zitiert auf Seite 7.)
- Paul Tipler and Gene Mosca. *Physik für Wissenschaftler und Ingenieure*. 2. Auflage. Elsevier, 2006. ISBN 3-8274-1164-5. (Zitiert auf Seiten 8 und 9.)
- James Palmer and Barbara Grant. *The Art of Radiometry*. SPIE Press, 2010. ISBN 978-0-8194-7245-8. (Zitiert auf Seiten 9, 11, 12 und 13.)
- P. Ricchiazzi, S. Yang, C. Gautier, and D. Sowle. *SBDART A Research and Teaching Software Tool for plane-parallel Radiative Transfer in the Earths Atmosphere*, volume 79, Bulletin of the American Meteorological Society, Seiten 2101–2114. 1998. (Zitiert auf Seiten 13, 14 und 42.)
- S. Kato, T. Ackermann, J. Mather, and E. Clothiaux. *The k -distribution method and correlated- k approximation for a shortwave radiative transfer model*. *Journal of Quantitative Spectroscopy and Radiative Transfer*, volume 62, Seiten 109–121. Elsevier, 1999. (Zitiert auf Seite 14.)
- F.X. Kneizys, E.P. Shettle, L.W. Abreu, and J.H. Chetwynd. *Users Guide to LOWTRAN7*. Air Force Geophysics Laboratory, Bedford, MA, 1988. (Zitiert auf Seite 14.)
- Sina Lohmann. *Langzeitvariabilität der globalen und direkten Solarstrahlung für Solarenergieanwendungen – Dissertation an der Fakultät für Physik an der Ludwig–Maximilians–Universität München*. Deutsches Zentrum für Luft- und Raumfahrt e. V., 2006. (Zitiert auf Seite 14.)
- Center for Astrophysics Harvard-Smithsonian. *The HITRAN Database*. Aufgerufen am 17. Mai 2012. URL <http://www.cfa.harvard.edu/hitran/>. (Zitiert auf Seite 14.)

- Bernhard Mayer, Claudia Emde, and Arve Kylling. *libRadtran User's Guide*. Edition for libRadtran version 1.6-beta. 3. Mai 2011. URL <http://www.libradtran.org/doc/libradtran.pdf>. (Zitiert auf Seiten 15, 17, 21, 22, 26, 28 und 38.)
- Knut Stamnes. The Theory of Multiple Scattering of Radiation in Plane Parallel Atmospheres. *Reviews of Geophysics*, 24:299–310, 1986. (Zitiert auf Seiten 15 und 17.)
- Bronstein, Semendjajew, Musiol, and Mühlige. *Taschenbuch der Mathematik*. 5. Auflage. Verlag Harri Deutsch, 2001. ISBN 3-8171-2005-2. (Zitiert auf Seite 15.)
- Wolfgang Brauch, Hans-Joachim Dreyer, and Wolfhart Haacke. *Mathematik für Ingenieure*. 11. Auflage. Teubner Verlag, 2006. ISBN 3-8351-0073-4. (Zitiert auf Seite 16.)
- Subrahmanyan Chandrasekhar. *Radiative Transfer*. Dover, New York, 1960. (Zitiert auf Seite 16.)
- Karl Rottmann. *Mathematische Formelsammlung*. B.I.-Hochschultaschenbuch, 1991. ISBN 3-411-70134-X. (Zitiert auf Seite 17.)
- Knut Stamnes, Tsay Si-Chee, and Laszlo Istavan. *DISORT, a General-Purpose Fortran Program for Discrete-Ordinate-Method Radiative Transfer in Scattering and Emitting Layered Media: Documentation of Methodology*. März 2000. (Zitiert auf Seite 18.)
- Frank Evans. *Radiative Processes in Planetary Atmospheres – Lecture Note Week 4*. University of Colorado, 2001a. URL <http://nit.colorado.edu/atoc5560/>. (Zitiert auf Seiten 18, 19 und 20.)
- Frank Evans. *Radiative Processes in Planetary Atmospheres – Lecture Note Week 2*. University of Colorado, 2001b. URL <http://nit.colorado.edu/atoc5560/>. (Zitiert auf Seite 18.)
- Hans-Joachim Eichler and Jürgen Eichler. *Laser – Bauformen, Strahlführung, Anwendungen*. 7. Auflage. Springer Verlag, 2010. ISBN 978-3-642-10461-9. (Zitiert auf Seite 19.)
- Wikipedia. *Atmospheric Radiative Transfer Codes*. Aufgerufen am 17. Mai 2012b. URL http://en.wikipedia.org/wiki/Atmospheric_radiative_transfer_codes. (Zitiert auf Seite 21.)
- Bernhard Mayer. libradtran Newsletter 13. *libRadtran-1.6-beta*, 4. März 2011. URL <http://www.libradtran.org/doku.php?id=newsletters>. (Zitiert auf Seite 24.)

- libRadtran. *FAQ*. Aufgerufen am 29. Mai 2012a. URL <http://libradtran.org/doku.php?id=faq>. (Zitiert auf Seite 24.)
- Joachim Baumann. *Groovy*. dpunkt.Verlag GmbH, 2008. (Zitiert auf Seite 25.)
- Bernhard Mayer. *Basic Usage*. Aufgerufen am 20. Mai 2012. URL http://libradtran.org/doku.php?id=basic_usage. (Zitiert auf Seite 28.)
- DLR. *SOFIA Stratosphären-Observatorium für Infrarot Astronomie*. Datenblatt. Mai 2010. URL <http://www.ontar.com/Software/ProductDetails.aspx?item=modtran>. (Zitiert auf Seite 31.)
- Eric Shettle. *Models of Aerosols, Clouds and Precipitation for Atmospheric Propagation Studies*. Conference Proceedings: in AGARD, Atmospheric Propagation in the UV, Visible, IR, and MM-Wave Region and Related Systems Aspects. Kopenhagen, 1989. (Zitiert auf Seite 32.)
- Fritz Gassmann. *Was ist los mit dem Treibhaus Erde*. vdf Verlag, 1994. ISBN 3-7281-1935-0. (Zitiert auf Seite 40.)
- Nicholas Short and Robert Blair. *Geomorphology from Space: A Global Overview of Regional Landforms*. United States Government Printing, 1986. ISBN 978-9998281073. (Zitiert auf Seite 43.)
- A. Berk, G.P. Anderson, P.K. Acharya, M.L. Hoke, J.H. Chetwynd, L.S. Bernstein, E.P. Shettle, M.W. Matthew, and S.M. Adler-Goldenand. *MODTRAN₄ User's Manual*. Version 3, Revision 1. 11. Februar 2003. (Zitiert auf Seite 44.)
- Ontar Corporation. *MODTRAN₅ Overview*. Aufgerufen am 30. Mai 2012. URL <http://www.ontar.com/Software/ProductDetails.aspx?item=modtran>. (Zitiert auf Seite 44.)
- Daniel Schläpfer and Daniel Odermatt. *MODO User Manual, Version 3*. ReSE Applications Schläpfer, 2006. (Zitiert auf Seiten 44 und 45.)
- Peter Kaiser and Johannes Ernesti. *Python – Das umfassende Handbuch – Aktuell zu Python 2.5*. 1. Auflage. Galileo Computing, 2007. ISBN 978-3-8362-1110-9. (Zitiert auf Seite 48.)
- Farid Hajji. *Das Python Praxisbuch*. 1. Auflage. Addison–Wesley, 2008. ISBN 978-3-8273-2543-3. (Zitiert auf Seiten 48 und 49.)
- Steve Holzner. *Eclipse*. 1. Auflage. O'Reilly Media, 2004. ISBN 3-89721-385-0. (Zitiert auf Seite 49.)
- PyDev.org. *PyDev Homepage*. Aufgerufen am 16. Mai 2012. URL <http://pydev.org/>. (Zitiert auf Seite 49.)

- Philipp Dermitzel. *Anleitung zu Eclipse und PyDev*. 07. Mai 2011, aufgerufen am 17. Mai 2012. URL mtmc.googlecode.com/files/eclipse-und-pydev-Anleitung.pdf. (Zitiert auf Seite 49.)
- SciPy.org. *SciPy Homepage*. Aufgerufen am 16. Mai 2012. URL <http://www.scipy.org/>. (Zitiert auf Seite 52.)
- matplotlib.org. *matplotlib Homepage*. Aufgerufen am 17. Mai 2012. URL <http://matplotlib.sourceforge.net/gallery.html>. (Zitiert auf Seite 52.)
- Sandro Tosi. *Matplotlib for Python Developers*. 1. Auflage. PACKT Publishing, 2009. ISBN 978-1-847197-90-0. (Zitiert auf Seite 52.)
- Johannes Plötner and Steffen Wendzel. *Linux – Das umfassende Handbuch*. 5. Auflage. Galileo Computing, 2012. ISBN 978-3-8362-1822-1. (Zitiert auf Seite 53.)
- Arnold Robbins and Dimitrij Surmeli. *sed und awk. Kurz und gut*. 2. Auflage. O'Reilly Media, 2002. ISBN 978-3-89721-246-6. (Zitiert auf Seite 55.)
- L.S. Rothman, I.E. Gordon, and A. Barbe. The HITRAN 2008 molecular spectroscopic database. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 110:533–572, 2009. (Zitiert auf Seite 58.)
- Laurence Rothman, John Schroeder, and Kuilian Tang. *Java HAWKS Manual*. Atomic and Molecular Physics Division of Harvard-Smithsonian Center for Astrophysics, 2004. URL <http://www.cfa.harvard.edu/hitran/docs.html>. (Zitiert auf Seiten 59 und 61.)
- Franz Schreier. *Py 4CATS – Download and Installation*. Aufgerufen am 18. Mai 2012. URL <http://www.libradtran.org/doku.php?id=lbl2od>. (Zitiert auf Seite 60.)
- Marcus Fischer. *Ubuntu GNU/Linux*. 7. Auflage. Galileo Computing, 2010. ISBN 978-3-8362-1654-8. (Zitiert auf Seite 64.)
- William Stine, Michael Geyer, and Sharon Stine. *Power From The Sun*. Aufgerufen am 10. Juni 2012. URL <http://www.powerfromthesun.net/index.html>. (Zitiert auf Seite 69.)
- Rudolf Richter, Martin Bachmann, Wouter Dorigo, and Andreas Müller. *Influence of the Adjacency Effect on Ground Reflectance Measurements*. *Geoscience and Remote Sensing Letters*. IEEE. Issue 4. Seiten 565–569, 2006. (Zitiert auf Seite 71.)

- Jia Guorui, Zhao Huijie, and Li Na. *Simulation of Hyperspectral Scene with Full Adjacency Effect*. Geoscience and Remote Sensing Symposium. IGARSS. IEEE International. Issue 3. Seiten 724–727, 2008. (Zitiert auf Seite 72.)
- Wout Verhoef and Heike Bach. *Simulation of Hyperspectral and Directional Radiance Images using Coupled Biophysical and Atmospheric Radiative Transfer Models*. *Remote Sensing of Environment*. Issue 87. Seiten 23–41. Elsevier, 2003. (Zitiert auf Seite 72.)
- Rudolf Richter and Daniel Schläpfer. *Atmospheric and Topographic Correction for Satellite Imagery*. DLR – German Aerospace Center, 2011. (Zitiert auf Seite 73.)
- D. Tanre, M. Herman, and PY. Deschamps. *Influence of the Background Contribution upon Space Measurements of Ground Reflectance*. *Applied Optics*. Issue 20. Seiten 3676–3684. Elsevier, 1981. (Zitiert auf Seite 74.)
- William Press, Saul Teukolsky, William Vetterling, and Brian Flannery. *Numerical Recipes in C – The Art of Scientific Computing*. 2.te Auflage. Cambridge University Press, 1981. (Zitiert auf Seiten 78 und 79.)
- James Cooley and Tukey John. *An Algorithm for the Machine Calculation of Complex Fourier Series*. *Mathematics of Computation*. Issue 19. Seiten 297–301. American Mathematical Society, 1965. (Zitiert auf Seite 79.)
- Bernd Jähne. *Digitale Bildverarbeitung*. *Mathematics of Computation*. Issue 19. Seiten 297–301. American Mathematical Society, 1989. (Zitiert auf Seite 80.)
- S. A. Starks and Kreinovich V. *Multi-Spectral Inverse Problems in Satellite Image Processing*. NASA Pan-American Center for Earth and Environment Studies (PACES). University of Texas at El Paso, 1998. (Zitiert auf Seite 86.)
- Robert Ryan. *A Simplified Approach to Radiometric Vicarious Calibration*. Civil Commercial Imagery Evaluation Workshop. NASA at the John C. Stennis Space Center, Mississippi, 2007. (Zitiert auf Seite 86.)
- NCAVEO. *Vicarious Calibration*. Aufgerufen am 5. Juni 2012. URL <http://www.ncaveo.ac.uk/calibration/radiometry/in-flight/>. (Zitiert auf Seite 86.)
- K. Chance and R.L. Kurucz. An Improved High-Resolution Solar Reference Spectrum for Earths Atmosphere Measurements in the Ultraviolet, Visible, and Near Infrared. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 111:1289–1295, 2010. (Zitiert auf Seiten 90 und 91.)

libRadtran. *Known Bugs*. Aufgerufen am 19. Mai 2012b. URL http://libradtran.org/doku.php?id=known_bugs. (Zitiert auf Seite 97.)

laura Baudis. *Sterne II: Atmosphären, die äusseren Schichten der Sonne*. Physikalisches Institut Ib, RWTH Aachen, 23. November 2006. (Zitiert auf Seite 98.)

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both \LaTeX and \LyX :

<http://code.google.com/p/classicthesis/>

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured at:

<http://postcards.miede.de/>

Final Version as of 29. Juni 2012 (`classicthesis` Version 1.0).

ERKLÄRUNG ZUR SELBSTÄNDIGEN ANFERTIGUNG

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer, als der im Literaturverzeichnis angegebenen Hilfsmittel, angefertigt habe. Die aus fremden Quellen, einschließlich der elektronischen Quellen, direkt oder indirekt übernommenen Gedanken sind ausnahmslos als solche kenntlich gemacht.

Neubeuern, 3. Juli 2012

Michael Jockisch