

NEXT GENERATION MODTRAN® FOR IMPROVED ATMOSPHERIC CORRECTION OF SPECTRAL IMAGERY

Alexander Berk, et al.

**Spectral Sciences, Inc.
4 Fourth Avenue
Burlington, MA 01803-3304**

29 January 2016

Final Report

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.



**AIR FORCE RESEARCH LABORATORY
Space Vehicles Directorate
3550 Aberdeen Ave SE
AIR FORCE MATERIEL COMMAND
KIRTLAND AIR FORCE BASE, NM 87117-5776**

DTIC COPY

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the PRS OPSEC Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-RV-PS-TR-2016-0105 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

//SIGNED//

Dr. Jeannette van den Bosch
Program Manager, AFRL/RVBYE

//SIGNED//

Dr. Thomas R. Caudill, Acting Chief
AFRL Battlespace Environment Division

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE				<i>Form Approved</i> OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 29-01-2016		2. REPORT TYPE Final Report		3. DATES COVERED (From - To) 21 Sep 2012 to 06 Jul 2015	
4. TITLE AND SUBTITLE Next Generation MODTRAN® for Improved Atmospheric Correction of Spectral Imagery				5a. CONTRACT NUMBER FA9453-12-C-0262	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER 64775F	
6. AUTHOR(S) Alexander Berk, Patrick Conforti, Fred Hawes, Tim Perkins, Chona Guiang, Prabhat Acharya, Rosemary Kennett, Brian Gregor, and Jeannette van den Bosch				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Spectral Sciences, Inc. 4 Fourth Avenue Burlington, MA 01803-3304				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory Space Vehicles Directorate 3550 Aberdeen Avenue SE Kirtland AFB, NM 87117-5776				10. SPONSOR/MONITOR'S ACRONYM(S) AFRL/RVBYI	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S) AFRL-RV-PS-TR-2016-0105	
12. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited. (OPS-16-10895 dtd 29 April 2016)					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT The MODTRAN® (MODerate resolution atmospheric TRANsmission) computer code is widely used throughout the Department of Defense (DoD), in many other Federal Government departments and agencies, and worldwide by research scientists in many fields for the prediction and analysis of optical measurements through the atmosphere. The code is embedded in many DoD operational and research sensor and data processing systems, particularly those involving the removal of atmospheric effects, commonly referred to as atmospheric correction, in remotely sensed multi- and hyperspectral imagery. These MODTRAN calculations are the most time-consuming part of the atmospheric correction process. Given the ever increasing capabilities of spectral sensors to quickly generate enormous quantities of data, combined with the need for image analysts to provide actionable information to the Warfighter in a timely manner, significantly speed up of MODTRAN processing was desired. Furthermore, MODTRAN was written in an outdated programming style and language (Fortran77), making it time-consuming and costly to upgrade and maintain, and difficult to integrate with other software. To address these limitations, SSI has developed the NextGen MODTRAN code (MODTRAN6), with a focus on current Air Force needs related to remote sensing applications used for Wide Area Surveillance.					
15. SUBJECT TERMS MODTRAN, Radiative Transfer, Line-By-Line, Atmospheric Correction, Transmittance, Radiance, Infrared/Visible					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Unlimited	18. NUMBER OF PAGES 72	19a. NAME OF RESPONSIBLE PERSON Dr. Jeannette van den Bosch
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (include area code)

This page is intentionally left blank.

Table of Contents

1. NEXT GENERATION MODTRAN EXECUTIVE SUMMARY	1
2. BACKGROUND	2
3. METHODS, ASSUMPTIONS, AND PROCEDURES	3
4. RESULTS AND DISCUSSION	4
4.1 Fortran Radiative Transfer Code	4
4.2 The MODTRAN6 Approach to Line-By-Line Radiative Transfer	10
4.2.1 The Traditional Line-By-Line Method	10
4.2.2 The MODTRAN Band Model RT Approach	11
4.2.3 The Initial MODTRAN6 Line-By-Line RT Approach	12
4.2.4 The Revised MODTRAN6 Line-By-Line RT Approach	14
4.2.5 Validations of the MODTRAN6 Line-By-Line and Band Model RT Approach	23
4.2.6 Summary and Future Work	30
4.3 MODTRAN6 Automated Testing Process and Tools	30
4.3.1 The comparebinaries Change Detection Software	31
4.3.2 The buildbot Code Monitoring Software	33
4.4 C/C++ Application Programming Interface	36
4.4.1 MATLAB® Interface	39
4.4.2 IDL® Interface	40
4.4.3 Python™ Interface.....	43
4.4.4 The MODTRAN6 GUI	43
4.4.5 Control of Output File Creation	47
4.5 Auxiliary Codes	48
4.5.1 The <i>tp5tojson</i> tool	48
4.5.2 Aerosol Toolkit	50
4.5.3 Atmosphere Generator Toolkit	55
4.6 Parallel Execution	59
5. CONCLUSIONS	61
REFERENCES	62

Note: The MODTRAN Trademark is being used with Express Permission of the Owner, The United States of America, as Represented by the United States Air Force.

Approved for public release; distribution is unlimited.

List of Figures

1. Overview of MODTRAN6 Architecture, Deliverables, Applications, and Milestones	2
2. Modular Partitioning of the MODTRAN6 Code.....	3
3. Connection Graph for the Basic Modules	7
4. More Detailed Connection Graph for the Various ‘Loop Modules’	7
5. (left) CH ₄ Optical Depth (left) and Line Tail Absorption (right) Curves.....	11
6. MODTRAN6 Ck Validation (black) against Initial MODTRAN6 LBL Calculations (red)	13
7. Continuum Absorption Cross-Section Contributions from a Single Line.....	16
8. Optical Depth from CH ₄ Lines Contributing to the 3000.0 and 3000.1 cm ⁻¹ Spectral Bin	16
9. (left) Candidate Interpolation/Extrapolation Curves for Fitting Line Tail Absorption	22
10. Validation of the MODTRAN6 LBL Algorithm (red) Against LBLRTM (black)	23
11. A Narrow Sub-region of the 9.6 μm O ₃ Band Exhibiting the ‘Residual Hash’ of Figure 10.....	24
12. A Narrow Sub-Region Containing the Largest Magnitude Residual from Figure 10.	25
13. Figure 10 Residual Arising from Differences in Air-Broadened Half-Widths	25
14. Validation of the MODTRAN6 Band Model against the MODTRAN6 LBL Algorithm.....	26
15. Same Conditions as Figure 14, but Highlighting Residuals at 1032.45 cm ⁻¹	27
16. Same Conditions as Figure 14, but Highlighting Residuals at 891.3 cm ⁻¹	27
17. Same Conditions as Figure 14, but Highlighting Residuals at 1010.8 cm ⁻¹	28
18. Comparison of MODTRAN6 Ck and LBL Radiances with Isaacs 2-Stream Scattering	29
19. Comparison of MODTRAN6 Ck and LBL Radiances with DISORT 8-Stream Scattering.....	30
20. A Set of <code>buildbot</code> Comparisons for the CirrusProfile Test Case.....	33
21. CirrusProfile Test Case Absolute (top) and Relative (bottom) Difference Histograms	34
22. A <code>buildbot</code> Status Webpage for MODTRAN6 (green passed, red failed)	35
23. Final Results of a <code>buildbot</code> Comparison of CirrusProfile Output Files	36
24. Validation of C-API Scanned Radiance Output Data	37
25. Visualizing a MODTRAN Spectral Library Output File Using ENVI	38
26. Example of a tabular Text Output Generated from the MODTRAN API.....	38
27. The Basic Structure of the MATLAB Interface	39
28. A MODTRAN Run Case Defined in a MATLAB Script	40
29. The Basic Structure of the IDL MODTRAN Interface	41
30. Example IDL Script to Define, Run, and Plot a MODTRAN Run	42
31. ACES-HY Data in GeoReplay [®] ’s GeoPaint [®] Display.....	43
32. MODTRAN6 Settings for ACES-HY Data.....	44
33. GUI Main Window.....	44

List of Figures (Continued)

34. GUI Case Editing/Options Window Showing Tabs for Six Types of Options	45
35. Custom Aerosol Input Window	46
36. Plotted MODTRAN Results and Drop-Down Menu of Output Options	47
37. The Derived Type node as Used in the tp5tojson Tool	48
38. Tree Structure of MODTRAN Linked List Using Generic Node Data Type of Figure 37	49
39. Sample New Input JSON Format.....	49
40. Diagram of the Architecture of the Aerosol Toolkit	50
41. Lognormal Size Distributions for the Longtin Desert Aerosol Model	51
42. Number Densities Profiles for the Desert Aerosol Components.....	52
43. Complex Refractive Index Data for the Aerosol Components.....	52
44. Desert Aerosol Extinction and Scattering Coefficients at Altitudes from 100 to 190 m.....	53
45. Combined Scattering Phase Functions for the Full Aerosol, at 8 and 15 μm Wavelengths	54
46. MODTRAN6 Simulations with Built-In and Aerosol Toolkit Desert Models.....	54
47. Diagram of the Atmosphere Generator Toolkit	55
48. NCEP/NCAR Long Term Near Surface Global Mean Temperatures for February	56
49. Pressure Profiles from Raw Radiosonde Data and the AGT 61 Level Fit.....	57
50. Temperature and H ₂ O Profiles from Raw Radiosonde Data and the AGT 61 Level Fit.....	57
51. Comparison of NCEP/NCAR 45° August to MODTRAN Mid-Latitude Summer Profiles.....	58
52. MODTRAN Limb Radiances Computed for the Profiles of Figure 51.....	59
53. Abbreviated Configuration Example of a Sweep of One Input Parameter, H2OSTR.....	60
54. Acceleration Benchmark of the Example Parallel MODTRAN Application, using Between 1 and 21 Processors to Execute 41 Separate MODTRAN Calculations	60

List of Tables

1. Descriptions of the MODTRAN6 Modules	4
2. Contents of the <i>outDta</i> Derived Data Type.....	8
3. Contents of the Five Different Derived Data Types Describing the Output	9
4. Path Segment/Level Dependent Derived Data Types, Listed in Order of Compilation.....	9
5. Typical Output From <i>comparebinaries</i> Utility.....	32
6. 10 m/s Wind Speed Desert Aerosol Particle Component Properties.....	51

1. NEXT GENERATION MODTRAN EXECUTIVE SUMMARY

The MODTRAN® (MODerate resolution atmospheric TRANsmission) computer code [1,2] is widely used throughout the Department of Defense (DoD), in many other Federal Government departments and agencies, and worldwide by research scientists in many fields for the prediction and analysis of optical measurements through the atmosphere. MODTRAN was developed and is still maintained through a longstanding collaboration between Spectral Sciences, Inc. (SSI) and the Air Force Research Laboratory (AFRL). The code is embedded in many DoD operational and research sensor and data processing systems, particularly those involving the removal of atmospheric effects, commonly referred to as atmospheric correction, in remotely sensed multi- and hyperspectral imagery (MSI and HSI). This process results in the retrieval of the surface spectral reflectance for each image pixel. These retrieved spectral “fingerprints” are directly compared to libraries of previously measured material reflectance spectra to identify materials and targets in an image.

The atmospheric correction process involves running MODTRAN many times for a range of possible atmospheric conditions, such as humidity level, atmospheric temperature, visibility, etc. These MODTRAN calculations have traditionally been the most time-consuming part of the atmospheric correction process. Given the ever increasing capabilities of spectral sensors to quickly generate enormous quantities of data, combined with the need for image analysts to provide actionable information to the Warfighter in a timely manner, significant speed up of MODTRAN processing was desired. Furthermore, MODTRAN was written in an outdated programming style and language (Fortran77), making it time-consuming and costly to upgrade and maintain, and difficult to integrate with other software. To address these limitations, SSI has developed the NextGen MODTRAN code (henceforth referred to as MODTRAN6), with a focus on current Air Force needs related to remote sensing applications used for Wide Area Surveillance (WAS). Benefits of this program include: (1) development of new real-time data processing capabilities for faster transmission of actionable information to the warfighter, (2) reduced development and maintenance costs for optical data processing systems, and (3) assurance of continuing and improving optical modeling accuracy in the future. The overview of this effort is displayed in Figure 1.

The underlying physics and algorithms used in MODTRAN are well established. The focus of this effort was on the conversion to a modern programming style and language, facilitating interfacing/integration with other applications, and utilizing the new API (Application Programming Interface) capability to provide code parallelization speedup options. The key objectives of this effort, as laid out in the statement of work, included:

1. Interface and I/O Upgrades: Develop a user-friendly GUI (Graphical User Interface) and a generalized API to allow for easy integration with other codes and operational systems.
2. Software Architecture: Design a modern, object-oriented architecture that allows for multiple code developers, facilitates code maintenance, enables physics and I/O upgrades to be more easily implemented, and simplifies integration protocols with other codes.
3. Physics Upgrade: Introduce a high spectral resolution line-by-line algorithm for higher spectral resolution applications and validation of the band model algorithms.
4. Code Speedup: Provide a parallelization method for NextGen MODTRAN using a Message Passing Interface (MPI) protocol.”
5. Integration: Demonstrate methods for integrating MODTRAN6 into third party software.

Each of these tasks were successfully completed and validated as is demonstrated in the body of this final report.

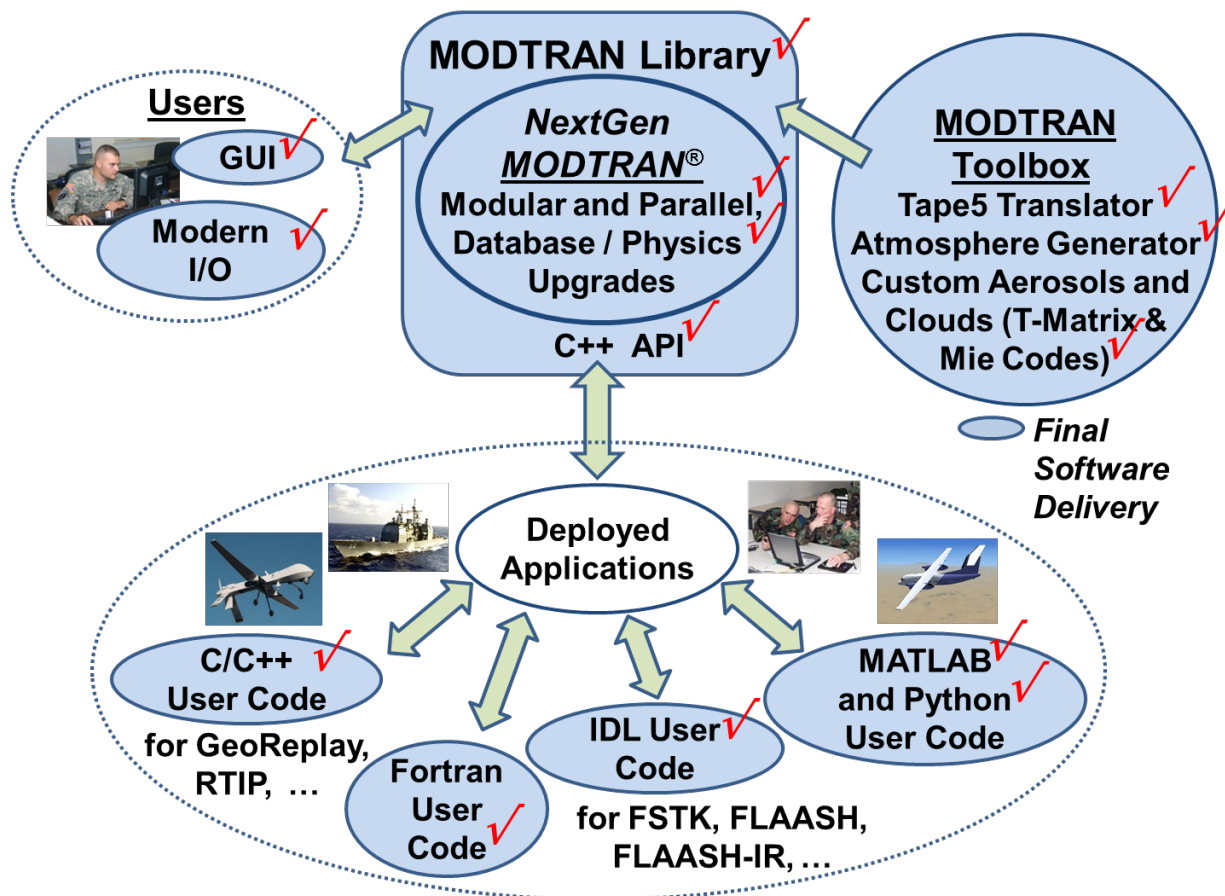


Figure 1. Overview of MODTRAN6 Architecture, Deliverables, Applications, and Milestones

2. BACKGROUND

The MODTRAN5 code, the precursor of MODTRAN6, computes line-of-sight (LOS) atmospheric spectral transmittances and radiances over the ultraviolet through long wavelength infrared spectral regime ($0 - 50,000 \text{ cm}^{-1}$; $> 0.2 \text{ }\mu\text{m}$). The radiation transport (RT) physics within MODTRAN5 provides accurate and fast methods for modeling stratified, horizontally homogeneous atmospheres. The core of the MODTRAN5 RT is an atmospheric "narrow band model" algorithm. The atmosphere is modeled via constituent profiles, both molecular and particulate, defined either using built-in models or by user-specified vertical profiles. The band model provides resolution as fine as 0.2 cm^{-1} from its 0.1 cm^{-1} band model. MODTRAN5 solves the radiative transfer equation including the effects of molecular and particulate absorption/emission and scattering, surface reflections and emission, solar/lunar illumination, and spherical refraction.

The main goal of this effort was to modernize the MODTRAN software, retaining its full functionality while modularizing the components, improving I/O interfacing and providing parallel computing options. In many respects, MODTRAN5 was an ideal candidate for this modernization. The MODTRAN5 software is comprised of more than 150,000 lines of Fortran77 code that generally follow the basic structure laid out in the 1980's, with common blocks, static arrays, block data and fixed format inputs. Fortunately, the components of the MODTRAN5 code partitions naturally into distinct modules. An illustration of how the MODTRAN5 building blocks were organized into the modular structure to form an overall framework for guiding the development of MODTRAN6 is presented in Figure 2.

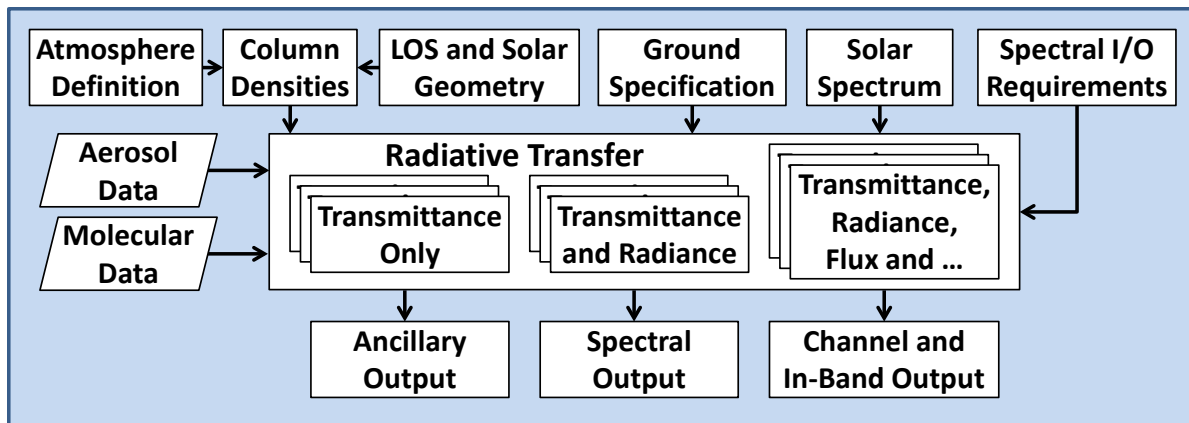


Figure 2. Modular Partitioning of the MODTRAN6 Code

3. METHODS, ASSUMPTIONS, AND PROCEDURES

Early efforts concentrated on redesign of MODTRAN to facilitate incorporation of modern programming practices, such as object-oriented programming and internal memory management. Outdated paradigms, such as memory overlays, deprecated statements, block data and other elements that remained from its development over several decades were eliminated. These obsolescent programming practices were handicapping the performance of MODTRAN and increasing the cost of incorporating new features. MODTRAN6 introduces modern means of being run, i.e., as an API, with easy-to-understand text and GUI inputs, and in library form for a myriad of platforms and languages. It can be run as a parallel application. Several key attributes of our approach and the resulting MODTRAN6 code include:

- **Technical Readiness Level (TRL):** MODTRAN5 is extensively validated, widely used by researchers and analysts, and embedded in many operational systems; thus, it has a TRL of 9. With the completion of the NextGen MODTRAN effort, MODTRAN6 now also has a TRL equal to 9, as it has been extensively tested and initiation of integration into several on-going operational systems has begun.
- **Transition Plan:** MODTRAN6 along with its newly developed toolbox, including all source code and data bases, is freely available (i.e., at no user cost) to all U.S. Government (USG) users and contractors working on USG programs.
- **Metrics:** MODTRAN is the industry standard for atmospheric effects modeling. MODTRAN6 retains all the current capabilities of MODTRAN5 and adds new ones. During the development process, each new prototype version of MODTRAN6 was run against the previously extensive and now enlarged further suite of test cases developed for MODTRAN. This insured that a working and validated version of MODTRAN6 was always available. The new line-by-line (LBL) capability was validated against test cases already established, and additional verification and validation test cases were introduced that compare directly to predictions from the Atmospheric and Environmental Research (AER) model LBLRTM [3,4].

MODTRAN6 also benefits from current and future Air Force (AF) operational programs in several key ways, including:

- **Enhanced AF capabilities:** The capabilities of new and future multi- and hyperspectral imaging sensors to quickly generate enormous quantities of data are rapidly outpacing the capabilities of data processing software to analyze the data and provide timely, actionable information to the warfighter. The parallel execution of MODTRAN6 provides an option for considerable speedup over MODTRAN5, and thereby, enables MODTRAN6 processing to keep up with current and future data requirements. Science upgrades, such as LBL radiation transport, the state-of-the-art

MT_CKD water vapor continuum, and more extensive libraries of custom atmospheres, improve the fidelity of the atmospheric correction process and results in more accurate retrieval of target and background spectra.

- Reduces Development and Sustainment costs: The interfaces and I/O handling in MODTRAN6 represents a major advance over MODTRAN5 in the ease of embedding an atmospheric effects model into both operational and off-line data processing systems. The modern architecture for MODTRAN6 insures its accessibility to both software engineers and scientists, making it easier to add new I/O and science features that may be needed for future systems. Finally, MODTRAN6 is maintained under an on-going licensing agreement between Spectral Sciences, Inc. and AFRL, thus, assuring continued code availability and access to improved optical models and performance in the future. Future capability enhancements are also being pursued, including a polarimetric generalization of the model.

4. RESULTS AND DISCUSSION

4.1 Fortran Radiative Transfer Code

The MODTRAN6 code base is fully recoded into Fortran-90 modules. All the common blocks have been transitioned to module attributes or into structures. All block data statements have been removed; these are now data or parameter statements incorporated inside modules. All modules have ROBODoc-style headers (<http://rfsber.home.xs4all.nl/Robo/robodoc.html>) at the top of the definition file. ROBODoc is a tool which extracts documentation comments from source code into a single document. It was used to extract the descriptions of the modules presented in Table 1.

Table 1. Descriptions of the MODTRAN6 Modules

MODULE	Contents/Description
aernsm_mod	Contains data and routines for describing the altitude, pressure, temperature, molecular, aerosol, cloud and rain profiles.
bnwDta_mod	Contains data & routines for writing binary tape8 output.
chaDta_mod	Contains derived data type for input atmospheric profiles.
ckdOut_mod	Contains routines to write the spectral and range-dependent correlated-k or high resolution LBL data to files.
ckdSav_mod	Contains routines to save the spectral and range-dependent k-distribution or high resolution LBL data in the output data structure, and to provide pointers to the components. The interfaces for the data-saving routines roughly mirror those in module kout_mod.
cldrr_mod	Contains cloud and rain data.
cool_mod	Contains Data and routines for handling cooling rates
crkDta_mod	Contains data types and routines to save correlated-k or high resolution LBL data in the output data structure, for access through the C API.
d_loop_mod	Contains routine D_LOOP, which loops over DISORT multiple scatter path segments at each spectral frequency.
dgrd_mod	Contains data and routines for spectral slit function processing.
disort_mod	Contains DISORT and all routines called by DISORT the exception of routines also used by DISUSR.
disusr_mod	Contains DISUSR and all routines called by DISUSR.
doslit_mod	Contains data and routines for handling slit.
dptDta_mod	Contains 1D double precision array for segments in LOS.
driver_mod	Contains data and routines for driving Modtran(R).
fill_mod	Contains data & routines related to filling column amounts.

Table 1. Descriptions of the MODTRAN6 Modules (continued)

MODULE	Contents/Description
filmgr_mod	Contains data & routines for managing files
filopn_mod	Contains routines for opening files which do not use the built-in error-handling system.
filter_mod	Contains filter-related data and functions.
filtrd_mod	Contains data structures and routines to read filter function files into linked lists of channels, for processing by subroutine filter in filter_mod.
flxDta_mod	Contains data and routines for handling and outputting fluxes.
frq5dt_mod	Contains data and routines for handling the layer-dependent 5 cm ⁻¹ spectral data.
gamfog_mod	Contains data & routines for modeling fog.
geodrv_mod	Contains data & routines for geometry.
i_loop_mod	Contains routines I_LOOP*, which loop over Isaacs multiple scatter path segments at each spectral frequency.
ierDta_mod	Contains an error handling module.
inpDta_mod	Contains input data structure.
iptDta_mod	Contains definition and auxiliary routines for 1D integer storage over LOS segments.
kout_mod	Contains data and routines for writing out the k-distribution data when KPRINT is TRUE.
L_loop_mod	Contains routines L_LOOPE and L_LOOPR to loop over radiance path segments (no MS) at each frequency, performing line-by-line RT.
linshp_mod	Contains routines for generating line-by-line absorption cross-section data.
loop_mod	Contains routine LOOP, which loops over radiance path segments (no MS) at each frequency.
lpshar_mod	Contains data & routines shared by the various versions of loop.
lyaDta_mod	Contains the data for an atmospheric level.
mach_mod	Contains hardware-specific constants.
math_mod	Contains mathematical data & routines.
mbmDta_mod	Contains Modtran band model data and handling routines.
mc_mod	Contains variables and routines related to generating data used in 3D Monte Carlo simulations.
merge_mod	Contains utilities for the task of merging sorted arrays, e.g. of atmospheric data, while eliminating/combining redundant elements.
mltDta_mod	Initializes the 6 built-in atmospheric profiles and other atmospheric profile constants.
modlib_mod	Defines interface types used in the C-API.
msrDta_mod	Contains data and routines for handling the DISORT interface.
outDta_mod	Contains output data structure.
outPlm_mod	Contains derived data type for data from running IEMSCT=1 with plume.
outPsc_mod	Contains derived data type for data from running IEMSCT=2 with plume.
outSol_mod	Contains derived data type for data from running IEMSCT=3.
outStr_mod	Contains derived data type for data from running IEMSCT=1 with no plume.
outTml_mod	Contains derived data type for data from running IEMSCT=1.
outTrn_mod	Contains derived data type for data from running IEMSCT=0.
outTsc_mod	Contains derived data type for data from running IEMSCT=2.
p_loop_mod	Contains routine P_LOOP, which loops over DISORT multiple scatter path segments at each spectral frequency for plume.
patDta_mod	Contains path segment pressure and temperature data.
path_mod	Contains data and routines for multiple scattering (MS) in each line-of-sight path segment.
phys_mod	Contains physical data and routines, including the molecular weights of constituent molecules and air.
plot_mod	Contains variables and routine for the PLT output.

Table 1. Descriptions of the MODTRAN6 Modules (continued)

MODULE	Contents/Description
prfDta_mod	Contains data & routines related to aerosol profiles.
rain_mod	Contains data & routines for rain rate.
rflDta_mod	Contains data & routines for handling surface reflectances.
rpaDta_mod	Contains derived data type of lines-of-sight.
rptDta_mod	Contains definition and auxiliary routines for 1D real storage over LOS segments.
runall_mod	Contains data and routines for controlling the runs of Modtran®.
s5pDta_mod	Contains altitude-dependent 5 cm ⁻¹ spectral grid data for local plume.
sapDta_mod	Contains Spectral Aerosol Profile (SAP) absorption and extinction data.
segDta_mod	Contains altitude-dependent 5 cm ⁻¹ spectral grid data.
sg2Dta_mod	Contains derived data type of 2D array usually over path segments and a 2 nd quantity.
shade_mod	Contains data and routines for solar scattering.
source_mod	Contains data & routines for calculating the source (Sun or Moon) irradiance.
sppDta_mod	Contains line-of-sight and segment dependent 5 cm ⁻¹ spectral grid data for local plume.
sspDta_mod	Contains derived data type for SAP data for a single path (multiple segments).
tinDta_mod	Contains derived data type for Band model temperature and pressure interpolation data for single LOS.
tisDta_mod	Contains derived data type for band model temperature and pressure interpolation data for a solar path.
tp8out_mod	Contains data and routines for writing out tape8.
tpnDta_mod	Contains derived data type for band model temperature and pressure interpolation data for a line-of-sight segment.
tpsDta_mod	Contains derived data type for band model temperature and pressure interpolation data for a single segment of a solar illumination path.
trnDta_mod	Contains TRANS-related data and functions.
tvoigt_mod	Contains routines to calculate VOIGT function.
ucdDta_mod	Contains data and routines for handling cloud data from an external file.
wdsDta_mod	Contains derived data type for Curtis-Godson Lorentz width squared sum.
wfxDta_mod	Contains derived data type for altitude dependent flux data.
wmdDta_mod	Contains derived data type for Curtis-Godson sums.
wpaDta_mod	Contains derived data type for column columns in single segment of a line-of-sight.
wppDta_mod	Contains derived data type for column columns in single segment of a local chemical plume.
wrtDta_mod	Contains data and routines for writing out Modtran results.

A graph showing the connections among the modules is presented in Figure 3 below. Leaf modules are highlighted by red ellipses. For clarity, many of the modules with the *Dta* suffix are excluded from the diagram. A generic ‘loop module’, labeled *x_loop*, is shown in Figure 3. A more detailed connection graph focusing on the individual loop modules is presented in Figure 4. For clarity, pairs of modules in Figure 4 that are called by the same set of routines have been grouped together: the *mbm* and *seg* modules, the *phys* and *shade* modules, and the *bnw* and *kout* modules.

A major change to the MODTRAN code has been the addition of derived data types for handling input and output. The output depends on the value that the user selects for the radiation transport mode, IEMSCT (Integer Emission and SCaTtering flag), and whether or not a local gas cloud is present. Thermal path radiance calculated when IEMSCT is 1; thermal plus solar path radiance calculated when IEMSCT is 2; and a solar irradiance is calculated when IEMSCT=3. The content of the output derived data type *outDta* is listed in Table 2. In Table 3, the listing of the different output is given for the different IEMSCT runs, and whether the run calls for a localized cloud. All quantities in Table 3 are output at

Table 2. Contents of the *outDta* Derived Data Type

Name	Type	Description
NSPCDT	integer	Number of spectral frequencies in output
NLOSDT	integer	Number of lines-of-sight
VBNDMN	real	Computational bandpass minimum frequency [cm^{-1}]
VBNDMX	real	Computational bandpass maximum frequency [cm^{-1}]
FACMUL	double precision	Multiplicative conversion factor for radiance
FAC1	double precision	Amount of initial frequency radiance to subtract from SUM to compute bandpass radiance
FACN	double precision	Amount of final frequency radiance to subtract from SUM to compute bandpass radiance
ALTKM	double precision(:)	Altitudes [km]
VLCHAN	double precision(:, :, :)	Channel-integrated quantities (when filter provided)
ierErr	ierDta pointer	Pointer to error messages if any, otherwise NULL
ierWrn	ierDta pointer	Pointer to warning messages if any, otherwise NULL
OUTW	double precision(:)	Spectral frequencies
TRNLOS	real(:)	Line-of-sight spectral transmittances.
AMBD	outSum	Summed quantities & extremes for ambient atmosphere
PLMD	outSum pointer	Summed quantities & extremes for local chemical plume
OUTP	outPlm(:) pointer	Output specific to IEMSCT = 1 with plume present
OUPR	outPsc(:) pointer	Output specific to IEMSCT = 2 with plume present
OUTS	outSol	Output specific to IEMSCT = 3
OUTE	outStr(:) pointer	Output specific to IEMSCT = 1
OUTT	outTrn(:) pointer	Output specific to IEMSCT = 0 (together with TRNLOS)
OUT2	outTsc(:) pointer	Output specific to IEMCT = 2
OFLX	outFlx	Spectral flux profiles
CKDA	kdata(:)	Correlated- <i>k</i> or line-by-line spectral/range data for ambient line-of-sight
CKDP	kdata(:)	Correlated- <i>k</i> or line-by-line spectral/range data for local plume line-of-sight
ODGRD	outDta pointer	Output spectrally convolved (degraded) over slit function

Table 3. Contents of the Five Different Derived Data Types Describing the Output

<u>Output Variable:</u> description [type]	<u>outStr</u> Thermal No Plume	<u>outTsc</u> Thm + Sol No Plume	<u>outPlm</u> Thermal with Plume	<u>outPsc</u> Thm + Sol with Plume	<u>outSol</u> Sol Irrad No Plume
DEMIS: ground directional emissivity [real]	X	X	X	X	
RFLSS: direct sun-ground-sensor radiance [real]		X		X	
S0: top-of-atmosphere (TOA) solar irradiance [real]		X		X	X
SUMMSS: multiple scatter path radiance [real]		X		X	
SUMSSS: single scatter solar path radiance [real]		X		X	
TRNLOS: line-of sight transmittance [real]	X	X	X	X	X
TS0: transmitted (TOA) solar irradiance [real]					X
TSNOBS: sun-sensor solar irradiance [real]		X		X	
TSNREF: sun-final_altitude-sensor solar irradiance [real]		X		X	
OUTL: line-of sight radiance components [outTml]	X	X	X	X	

Table 4. Path Segment/Level Dependent Derived Data Types, Listed in Order of Compilation

Name	Definition	Implementations
tpnDta	Band model P/T interpolation data for 1 LOS segment containing FT GT HT KT LT P2 PT T5	
tinDta	1D array of tpnDta over LOS segments	pT_MC pTPMS pTINA(NLOS) pTNLA(NLOS)
tpsDta	Band model P/T interpolation data for 1 solar path segment containing FTS JTS P2S PTS T5S PSATM TSSEG	
tisDta	1D array of tpsDta over solar path segments	pTMSS, pTPSA(NLOS), pTSLA(NLOS)
patDta	P and T arrays over segments	PAMS, pPALS(0:NLOS), pPAPL(0:NLOS), pPATH(NLOS)
pthDta	1D D.P. array over segments	PTHALT(NLOS) PTHRNG(NLOS) Y_RANG(NLOS)
sg2Dta	2D S.P. array over segments	SPCHI/LO(NLOS,NPATH), SPCML/LO(NPATH), WPTHMS, WSPTHM, pWPTH(0:NLOS), pWSPTH(NLOS), RNGDEN(NLOS), SLEGEN(NLOS)
rptDta	1D S.P. array RSEG over segments	1D arrays over NLOS: FACALT, PF_HG, PF_ICE, PF_WAT, PHSANG, PHSCOS, PHSFAC, PTHRH, PTHY_T, RNG_T, TAUPTH, U_MU, U_PHI 1D array over NPATH: SSAPHM, SSAPLM

Table 4. Path Segment/Level Dependent Derived Data Types, Listed in Order of Compilation (cont.)

Name	Definition	Implementations
		2D arrays over (NLOS,NPATH): SSAPH, SSAPL
iptDta	1D integer array ISGP over segments	1D arrays over NLOS: IUPPHS LAY_D LOWALT MAPPTH
sapDta	SAP data for 1 segment, containing spectral arrays XSAP and ASAP	
sspDta	1D array of sapDta over segments	APSAP, BSAP, SAPMS, SAPSM,, pSAPA(0:NLOS) pSPSA(1:NLOS)
wppDta	Local chemical plume molecular column densities for a single segment containing WP_1 WP_2 WP_5 WP_6 WP_8 WP_9 WP10 WP59 WP60 WP63	
wpaDta	1D array of wppDta over path segments	WPTHPA(0:NLOS), WSPPA(NLOS)
s5pDta	5 cm ⁻¹ spectral data for plume containing S_3P S_5P S_8P S_9P S17P S18P S33P S34P S35P S36P S38P	
sppDta	1D array of s5pDta over segments	2D arrays over (NLOS,NPATH): p5PHI p5PLO
rpsDta	Assemblage of rptDta objects	DMSSOL(KNTRVL), DMSTHM(KNTRVL)

4.2 The MODTRAN6 Approach to Line-By-Line Radiative Transfer

A Line-By-Line (LBL) option has been developed for MODTRAN6. The motivation for this development was two-fold. Firstly, when MODTRAN is validated against an independent LBL model, it is often difficult to isolate the source of discrepancies. One must verify consistency between pressure, temperature and density profiles, between column density calculations, between continuum and particulate data, between spectral convolution methods, and more. Introducing a LBL option directly within MODTRAN will insure common elements for all calculations other than those used to compute molecular transmittances. Thus, the new option will facilitate testing and subsequent refinement of the MODTRAN band model.

The second motivation for the LBL upgrade is that it will enable users to compute high spectral resolution transmittances and radiances not only for the full range of current MODTRAN applications, but also for radiative transfer problems outside of current purview of MODTRAN. In particular, MODTRAN current applications include methods for solving the multiple scattering problems. Introducing the LBL feature into MODTRAN will enable first-principle calculations of scattered radiances, an option that is often not readily available with LBL models.

MODTRAN6 computes LBL transmittances within 0.1 cm⁻¹ spectral bins marching through the full requested band pass. The LBL algorithm uses the highly accurate, pressure- and temperature-dependent MODTRAN Padé approximant fits to the contribution from line tails to define the absorption from all molecular transitions centered more than 0.05 cm⁻¹ from each 0.1 cm⁻¹ spectral bin. The beauty of this approach is that the on-the-fly computations for each 0.1 cm⁻¹ bin will only require explicit LBL summing of transitions centered within a 0.2 cm⁻¹ spectral region. That is, the contribution from the more distant lines is pre-computed via the Padé approximants.

4.2.1 The Traditional Line-By-Line Method

LBL methods compute monochromatic spectral transmittances and radiances on an arbitrarily fine grid of spectral points by explicitly solving the radiative transfer (RT) equation. Most LBL codes model the temperature- and pressure-dependent molecular absorption by summing line-shape contributions from every molecular transition centered within 25 cm⁻¹ of each spectral frequency grid point. Distant lines, i.e., those centered more than 25 cm⁻¹ from the calculational frequency, are modeled via temperature-dependent continua databases. Given a typical spectral resolution of 0.001 cm⁻¹, an 8.0 to 12.5 μm band would contain 450,000 spectral points (800.0005, 800.0015, ..., 1249.995 cm⁻¹). At each one of these points, the absorption sum must be computed for each segment along a line-of-sight (LOS). The 2012

HITRAN molecular transition database [5] contains 27,651 lines centered within 25 cm^{-1} of 1000.05 cm^{-1} ($\sim 10 \text{ }\mu\text{m}$) for the default MODTRAN band model species, the first 12 molecules in the HITRAN compilation. Clearly, the computation of high-resolution molecular transmittances involves significant computational effort.

The LBLRTM radiative transfer software [3,4] distributed by the AER generates spectral transmittances and thermal emitted radiances at a 0.001 cm^{-1} increment over the 8.0 to $12.0 \text{ }\mu\text{m}$ band in about 20 seconds for a 60° off-nadir view through the 1976 U.S. Standard (USS) model atmosphere. By comparison, the MODTRAN6 band model run with its statistical correlated- k algorithm at 0.1 cm^{-1} spectral resolution performs the same calculation on the same machine, a 32 core Advanced Micro Devices, Inc. (AMD) Opteron with 512 Gbytes of RAM, more than 35 times faster.

4.2.2 The MODTRAN Band Model RT Approach

Band model RT algorithms use statistical data on the distribution of line strengths in contiguous, regularly spaced spectral intervals to compute bin-averaged transmittances and radiances, bypassing explicit sampling of individual spectral lines. The MODTRAN band model (BM) transmittance formulation partitions molecular absorption into 3 components. Absorption from distant lines is modeled via temperature-dependent continua databases, identical to the method used in LBL models. However, MODTRAN treats the molecular absorption within each BM spectral bin, whether it be of width 0.1 , 1.0 , 5.0 or 15.0 cm^{-1} , as arising from two additional sources: the contribution from lines centered within the spectral bin, and the remaining line-tail absorption from molecular transitions centered outside of the bin but within 25 cm^{-1} . Figure 5 (left) illustrates the absorption optical depth from the set of HITRAN 2012 CH_4 lines centered within the single 0.1 cm^{-1} bin between 3000.0 and 3000.1 cm^{-1} (a homogeneous segment containing 30 atm-cm CH_4 at $T=296 \text{ K}$ and $P=0.1 \text{ atm}$). As the figure illustrates, one obtains significant spectral structure from the line center contributions, but the line-tail absorption is a smooth function that falls off precipitously with distance from the spectral bin at 0.1 atm pressure (note that the ordinate axis is logarithmic). Upon summing the line-tail contributions (blue curves) of all CH_4 transitions centered outside of the spectral bin (but within 25 cm^{-1}), one obtains a smooth curve (in black) with at most one minimum, Figure 5 (right).

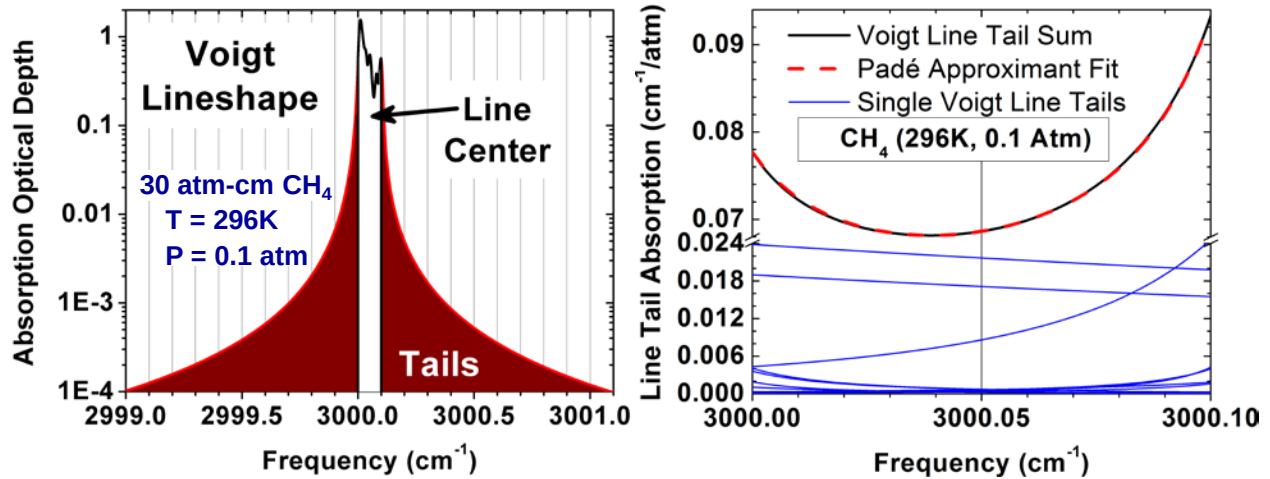


Figure 5. (left) CH_4 Optical Depth (left) and Line Tail Absorption (right) Curves

For MODTRAN, the line-tail absorption cross-section spectral curves (divided by pressure, P), $a_P(\delta_\nu)$, within each band model bin of width $\Delta\nu$, are fit to a $[2, 2]$ Padé approximant, i.e., the ratio of quadratic polynomials in δ_ν ; δ_ν is defined as the spectral frequency, ν , displacement from bin center, ν_{cen} , normalized to range from -1 to $+1$:

$$a_p(\delta_\nu) = \frac{1}{P} \left(\frac{\sigma_0 + \sigma_1 \delta_\nu + \sigma_2 \delta_\nu^2}{1 + d_1 \delta_\nu + d_2 \delta_\nu^2} \right) ; \quad \delta_\nu \equiv \frac{\nu - \nu_{cen}}{\Delta\nu/2} ; \quad \left| \begin{array}{l} \sigma_0 + \sigma_1 \delta_\nu + \sigma_2 \delta_\nu^2 \geq 0 \\ 1 + d_1 \delta_\nu + d_2 \delta_\nu^2 > 0 \end{array} \right. . \quad (1)$$

For the band model, the curves are fit for two values of pressure (1.0 and 0.1 atm); at a series of temperatures (180 to 330 K in 5K increments); and for *all* the HITRAN line compilation molecules. The five parameters [σ_0 , σ_1 and σ_2 with cross-section units (cm^2/atm), and dimensionless d_1 and d_2] that define each Padé approximant are determined from the summed absorption cross-sections at the bin center and bin edges; from the integral of the absorption cross-section curve over the interval; and from the spectral derivative of the absorption cross-section at bin center. As Figure 5 (right) illustrates for a single sample case, the fits are extremely accurate (dashed red curve); generally, relative residuals are of order 10^{-5} . To help insure excellent fits, MODTRAN has traditionally translated inward all line center positions that are too close to a bin edge. This helps avoid the zero slope at line center and the relative sharp Gaussian (Doppler) drop off that can occur at the lower 0.1 atm pressure. The stipulation is that the residuals introduced by shifting line center positions a fraction of a spectral bin (no more than $0.25 \Delta\nu$) will be small compared to the inherent errors of the statistical band model. In sub-section 4.2.3.2, this hypothesis is tested by comparing the MODTRAN6 BM and LBL 0.1 cm^{-1} spectral bin transmittances.

4.2.3 The Initial MODTRAN6 Line-By-Line RT Approach

The initial MODTRAN6 LBL approach simply provided an option to replace the statistical band model *line center* absorption calculation with a first-principles LBL integration of the absorption line-shape of all molecular transitions centered in each 0.1 cm^{-1} interval. A new MODTRAN6 input, N_RES, defines the number of spectral grid points in each 0.1 cm^{-1} bin. For example, if N_RES is set to 100, the default value, then 0.001 cm^{-1} spectral resolution calculations are performed; the 100 spectral frequencies for the 3000.0 to 3000.1 cm^{-1} bin are 3000.0005 , 3000.0015 , ..., $3000.9995 \text{ cm}^{-1}$. At each spectral grid point, ν_i , the explicit LBL calculation of line center absorption is added to the sum of the Padé fit of the line-tails and the spectrally interpolated continuum at ν_i .

4.2.3.1 Features of the Initial MODTRAN6 Line-By-Line RT Approach

Three beneficial features of the initial MODTRAN6 LBL approach are noted in this sub-section. First, the number of calculations is reduced dramatically from the traditional LBL approach. As enumerated earlier, a total of 27,651 HITRAN lines for the MODTRAN default BM molecules fall within the 50 cm^{-1} domain centered at 1000.05 cm^{-1} ; the MODTRAN6 LBL algorithm performs on-the-fly line-shape absorption cross-section calculations for molecular transitions centered within the narrow 0.1 cm^{-1} bin containing 1000.05 cm^{-1} , only 55 lines.

A second feature of the initial MODTRAN6 LBL approach is that it provides an option to directly measure of the accuracy of the statistical band model line-center transmittance calculation, since the computation of extinction from all other sources is identical for the two methods.

The third feature of the initial MODTRAN6 LBL approach is that it enables specification of segment and spectral dependent optical depth (OD) cut-offs for use in speeding up processing. Within an initial loop over all candidate lines for a given spectral bin, a pre-computed 296 K line center absorption cross-section is multiplied by the molecule's largest column density from the current MODTRAN geometry calculations, a value determined prior to entering the frequency loop. If the resulting OD exceeds the spectrally-independent cut-off value, ODCut, currently set at 10^{-8} , then the molecular transition is added to a list of active lines. Prior to beginning the explicit LBL calculations for each path segment, the particulate extinction, the Rayleigh scattering and the molecular continuum and line-tail optical depths are summed at each grid point within the current spectral bin. A path segment and spectral bin *dependent* cut-off value, ODmin, is set to the minimum spectral grid point extinction OD multiplied by 10^{-4} , but not be less than ODCut. In the loop over active lines, spectral Voigt ODs are computed outward from line

center. Once a line's absorption OD drops below OD_{min} , the computationally expensive calculation is terminated. For baseline 8.0 to 12.0 μm test cases, downward viewing from 20 km, the OD tolerance technique reduces processing time by ~50%.

4.2.3.2 Validation of MODTRAN's Band Model Line-Center Transmittance Calculations

Spectral transmittances, generated using the initial MODTRAN6 LBL approach, were convolved with the MODTRAN 0.1 cm^{-1} non-overlapping, contiguous rectangle slit functions to provide a direct comparison to the finest resolution band model calculations. These comparisons invoked MODTRAN6's highest fidelity spectral bin transmittance options, the MODTRAN 0.1 cm^{-1} BM and its statistical Correlated- k (Ck) algorithm. The results of an 8.0 to 12.0 μm comparison are illustrated on the left in Figure 6 for a nadir path from 20 km altitude through the model mid-latitude summer (MLS) atmosphere. The Ck (black) and LBL (red) spectral transmittances are plotted in the lower half of the figure; above, LBL minus Ck residuals, offset by 1, are plotted on an expanded scale.

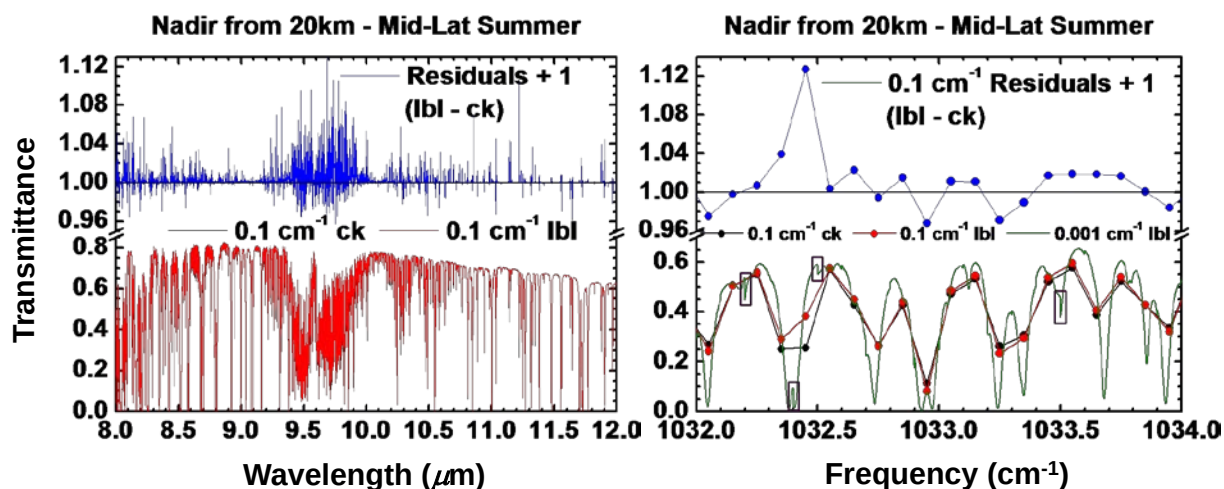


Figure 6. MODTRAN6 Ck Validation (black) against Initial MODTRAN6 LBL Calculations (red)

The 9.6 μm O_3 band has always been the most difficult molecular band for MODTRAN to model accurately; the relative flat O_3 vertical density profile below 35 km (number densities between 1 and $6 \times 10^{12} / cm^3$) and overlap with low altitude H_2O absorption create problems for both the BM and Ck methods. This is clearly evident in Figure 6, where 9.6 μm band transmittance residuals range between -0.05 to $+0.13$. Outside of this region, most residuals have a magnitude less than 0.02, and there appears to be little bias towards too much or too little absorption. This latter fact suggests that degrading the data to coarser resolution will significantly lower these residuals; it is generally recommended that MODTRAN spectral output be degraded with a slit function whose full-width at half-maximum (FWHM) is at least twice the BM resolution ($0.2 cm^{-1}$ in this case).

The LBL to Ck comparison was performed at the MODTRAN band model finest spectral resolution ($0.1 cm^{-1}$) to facilitate studying the source of the residuals. To this end, a $2.0 cm^{-1}$ region containing the largest residual, $t_{lbl} - t_{ck} = 0.13$, is shown on the right in Figure 6. This figure also contains the fine, $0.001 cm^{-1}$ spectral resolution LBL transmittance curve, shown in green. Focusing on the $0.1 cm^{-1}$ spectral bin between 1032.4 and $1032.5 cm^{-1}$, the LBL spectrum seems to illustrate that there is a strong absorption doublet near the left edge (see the purple box just above the tick mark at $1032.4 cm^{-1}$). In actuality, there are 3 strong O_3 lines near this bin's left edge, with 1 atm pressure transition frequencies of 1032.3799 , 1032.4007 and $1032.4057 cm^{-1}$; the 296 K strength of these lines are 0.9514 , 0.2754 and $0.6018 cm^2/atm$, respectively. The MODTRAN band model assumes absorption lines are randomly located within each spectral bin. Having two strong lines centered so close to an edge and to each other is a statistical anomaly. MODTRAN also translates the line centers $0.025 cm^{-1}$ away from the edge for calculation of the

line tail absorption coefficient curves. This shifting is performed to avoid possible problems fitting the line-tail contributions to a Padé approximant in the neighboring spectral bin, as was described above. For the 1032.4 to 1032.5 cm⁻¹ bin, these small shifts lead to a large over-prediction of the absorption by the band model. A smaller over-prediction occurs for the neighboring spectral bin to the left, because it has an even stronger line, located 0.0201 cm⁻¹ from the edge, shifted inward. The LBL 0.1 cm⁻¹ spectral bin transmittance at 1032.45 cm⁻¹ is 0.39, while the CK transmittance is 0.26. One observes from this analysis the current line center shifting is introducing errors in absorption that are unacceptably large; the band model methodology needs to be refined to reduce these residuals.

4.2.3.3 Spectral Discontinuities

A second problem is evident on the right in Figure 6. Purple boxes centered at 1032.2, 1032.4, 1032.5 and 1033.5 highlight large spectral bin edge discontinuities in the LBL spectral transmittances. This is unacceptable for a LBL model, which should provide the highest possible accuracy. The need to eliminate these discontinuities led to the reformulation of the MODTRAN LBL algorithm, as described in Section 4.2.4.

4.2.4 The Revised MODTRAN6 Line-By-Line RT Approach

The challenge of the MODTRAN6 LBL algorithm is to eliminate discontinuities at spectral bin edges. The problem arises because MODTRAN6 solves the LBL problem within disjoint 0.1 cm⁻¹ bins, and then concatenates these distinct solutions. A line centered in one bin is modeled via an explicit LBL calculation in that bin, but its contributions to the neighboring bins are determined from temperature- and pressure-interpolated pre-computed line-tail data. With the initial MODTRAN6 LBL model, the discontinuity error is largely due to the fact that the line-tail cross-sections are generated for lines whose centers are shifted if centered too close to an edge; these same lines are not shifted when their LBL line-center calculations are performed. Revising the approach to eliminate the need for line shifting reduces the magnitude of the discontinuities considerably, but their presence persisted. As is shown below, additional upgrades to the formalism were required to produce smooth transitions across all spectral bin edges.

A number of other problems had to be dealt with to validate the MODTRAN6 LBL algorithm against the chosen validation source, LBLRTM predictions. Previously, MODTRAN has never concerned itself with modeling Lorentz self-broadening or modeling air pressure (or density) induced line shifts. These details cannot be ignored at LBL spectral resolutions. Another problem is that MODTRAN uses the Voigt line-shape function, which does not exhibit correct behavior in the microwave spectral region.

All of the above problems are addressed in the revised MODTRAN6 LBL algorithm. The following subsections describe the algorithm's updates to the line shape function, its use of a 2-bin line-center approach, the treatment of air-pressure (or density) induced line shifts and the modeling of Lorentz self-broadening.

4.2.4.1 Updating the Line-shape Function

MODTRAN6 models the absorption cross-section, $k_\nu(T; \gamma_c, \gamma_d)$, at spectral frequency ν for a molecular transition of strength $S_{if}(T)$ with a Voigt line-shape function, f_ν :

$$k_\nu(T; \gamma_c, \gamma_d) = S_{if}(T) [f_\Delta(\gamma_c, \gamma_d) + f_\Sigma(\gamma_c, \gamma_d)]; \quad \Delta = \nu - \nu_{if}; \quad \Sigma = \nu + \nu_{if}$$

$$S_{if}(T) \propto \frac{\nu_{if}}{Q(T)} \left(1 - e^{-c_2 \nu_{if}/T}\right) e^{-c_2 E_i/T}; \quad f_\Delta(\gamma_c, \gamma_d) = \int_{-\infty}^{\infty} \frac{\gamma_c \exp(-t^2)}{\gamma_c^2 + (\Delta - \gamma_d t)^2} \frac{dt}{\pi^{3/2}}. \quad (2a)$$

Here, γ_c is the Lorentz (collision) half-width at half maximum (HWHM), γ_d is the Doppler half-width at 1/e of maximum, ν_{if} is the transition frequency from the initial (*i*) lower state to the final (*f*) upper state,

$Q(T)$ is the total partition function at temperature T , c_2 is the second radiation constant, and E_i is the lower state energy. It is well known that the Voigt line-shape function has an incorrect form at low spectral frequencies; in particular, the absorption cross-section should approach zero as the frequency approaches zero, but it does not. To remedy this problem, LBLRTM uses the Van Vleck and Huber (VVH) [6] line-shape function, $\tilde{f}_v$, a form that transitions to the Voigt line-shape whenever the ratio of the computation frequency to the transition frequency, ν/ν_{if} , is near unity:

$$\begin{aligned} \tilde{k}_v(T; \gamma_c, \gamma_d) &= \tilde{S}_{if}(T) \left[\tilde{f}_\Delta(\gamma_c, \gamma_d) + \tilde{f}_\Sigma(\gamma_c, \gamma_d) \right] \\ \tilde{S}_{if}(T) &\equiv \frac{S_{if}(T)}{\nu_{if} \tanh(\frac{1}{2} c_2 \nu_{if} / T)}; \quad \tilde{f}_\Delta(\gamma_c, \gamma_d) \equiv \nu \tanh(\frac{1}{2} c_2 \nu / T) f_\Delta(\gamma_c, \gamma_d) \quad . \end{aligned} \quad (2b)$$

Throughout most of the infrared, ν/ν_{if} is indeed near unity for frequencies within 25 cm^{-1} of ν_{if} ; so there is essentially no difference between the Voigt and VVH treatments in these regions.

In the revised MODTRAN6 Line-By-Line RT implementation, the MODTRAN LBL calculations have been upgraded to use the VVH line-shape function. In principle, the MODTRAN line-tail database should also be generated using the VVH function. This upgrade has yet to be implemented for the following reason. One of the quantities used to determine the Padé approximant parameters is the integral of the line-shape curve over the spectral bin (this was noted in Section 4.2.2). Forcing the integral conditions insures preservation of the total integrated strength. In the program that generates these parameters, the Voigt line-shape function is approximated using the original expansion due to Humlíček [7,8]. The advantage of the Humlíček expansion over more modern (and accurate) algorithms is that its terms can be analytically integrated spectrally. The computation of these integrals is more difficult when the Voigt line-shape function is multiplied by the $(\nu/\nu_{if}) \tanh(\frac{1}{2} c_2 \nu / T) / \tanh(\frac{1}{2} c_2 \nu_{if} / T)$ factor. To solve this problem, the $\tanh(\frac{1}{2} c_2 \nu / T)$ term can be expanded in a Taylor series about the central frequency for each spectral bin, ν_{cen} , yielding an analytically integrable expression; the variation of the hyperbolic tangent over the 0.1 cm^{-1} spectral bin should be small and the convergence of the Taylor series quick. This change to the line tail calculations will be instituted the next time the line tail database is updated..

There is one additional upgrade to be made to MODTRAN6 to provide a consistent implementation of the VVH line-shape function. As noted earlier, line-tail absorption contributions more than 25 cm^{-1} from line center are modeled via continua databases. This suggests that the continuum absorption cross-section contribution from a single Voigt molecular transition has the *No Center* form shown on the left in Figure 7. Such contributions would produce spectral continua with multiple discontinuities. To create smoothly varying continua, the continuum data for a single Voigt line is defined to include a flat plateau within $\pm 25 \text{ cm}^{-1}$ of line center, Figure 7 (center). When molecular absorption within this 50 cm^{-1} central region is modeled, the absorption cross-section value at $\pm 25 \text{ cm}^{-1}$ must be subtracted. The VVH line-shape function is not symmetric about the transition frequency. The question arises as to what value must be subtracted, or, equivalently, how is the continuum contribution defined within the central region? The answer is that the flat plateau is multiplied by the same factor as that used to convert the Voigt to a VVH line-shape, namely, $(\nu/\nu_{if}) \tanh(\frac{1}{2} c_2 \nu / T) / \tanh(\frac{1}{2} c_2 \nu_{if} / T)$. On the right in Figure 7, the effect of this factor is illustrated at standard temperature (273.15 K) for a molecular transition centered at 200 cm^{-1} . Although the plateau appears to simply be tilted, the true slope is not constant due to the very slight non-linear variations in the hyperbolic tangent term.

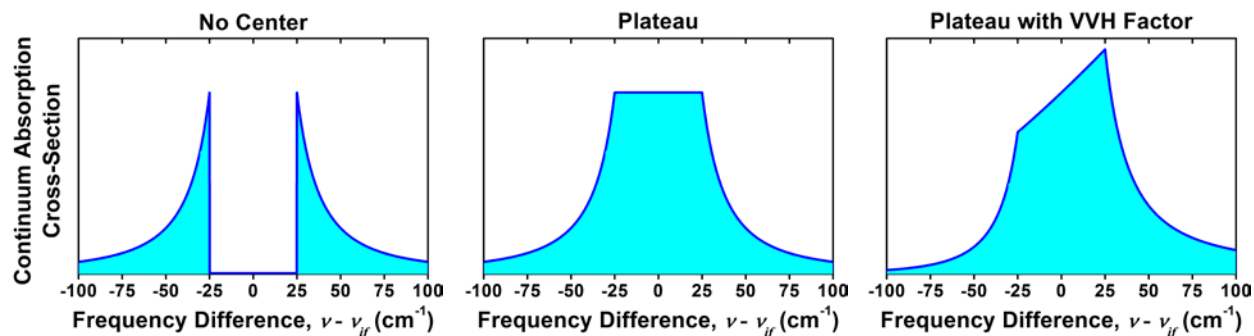


Figure 7. Continuum Absorption Cross-Section Contributions from a Single Line.

In the MODTRAN band model, molecular lines that are located too close to a band model spectral bin edge are translated inward towards the center of the bin. This is done because Padé approximant spectral fits of line-tails may be inaccurate for molecular transitions located just beyond a bin edge. The spectral slope for such lines approach zero and Doppler (Gaussian) contributions can be strong at low pressures. The stipulation is that the form of the Padé fits does not provide the flexibility to properly capture this spectral shape.

To avoid the need to shift line center positions within the MODTRAN6 LBL implementation, the Voigt line-shape of each molecular transition is explicitly modeled in two 0.1 cm^{-1} bins *instead of one*. Line-tails are still fit to a Padé approximant, but the extra bin provides a spectral buffer between the line center and its line-tail contributions. As Figure 8 illustrates, the spectral bin between 3000.0 and 3000.1 cm^{-1} has a set of molecular transitions that it shares with its left neighboring bin and a set that it shares with its right neighboring bin. In principle, each line is assigned the bin containing the line and its nearest neighbor. With such a selection, each line would be centered half a bin width or more from its line-tail region. This is illustrated in Figure 8 where the fine spectral structure for both the red and blue curves is separated from the modeled line-tails by $\sim 1/2$ of a spectral bin, i.e., 0.05 cm^{-1} .

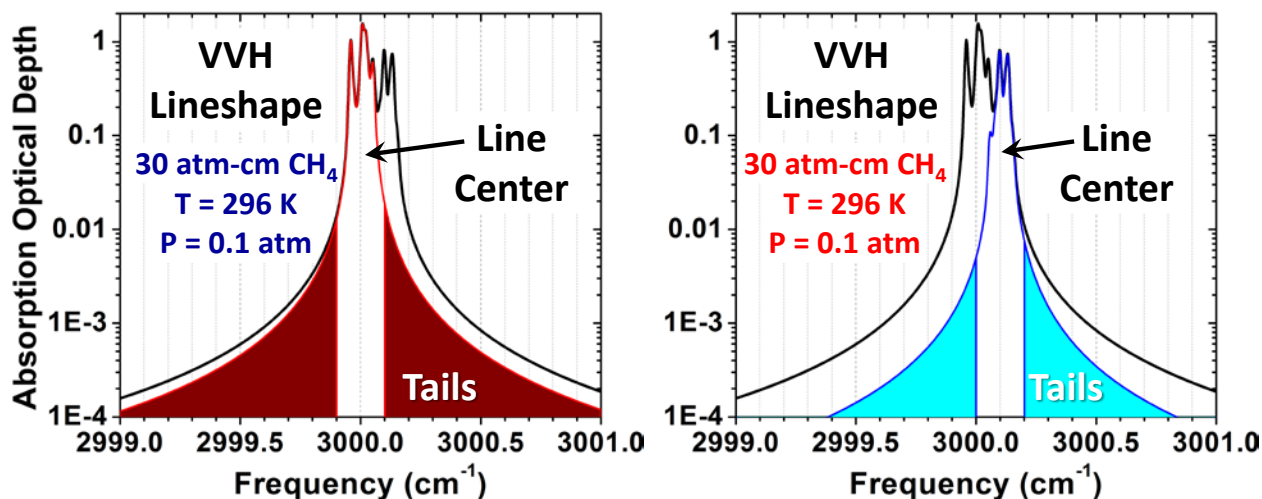


Figure 8. Optical Depth from CH_4 Lines Contributing to the 3000.0 and 3000.1 cm^{-1} Spectral Bin

The spectral bin assignment procedure is more complicated than just described due to air pressure- or density-induced line shifts. According to HITRAN documentation [5], transition frequencies vary with pressure according to the equation

$$\text{HITRAN : } \quad \nu_{if}^H(P) \equiv \nu_{if}^{\text{vacuum}} + \nu_{if}^{\text{shift}} \left(\frac{P}{1 \text{ atm}} \right) . \quad (3a)$$

Within LBLRTM, temperature dependence is added, by defining an air density induced line shift:

$$\text{LBLRTM : } \quad \nu_{if}^L(P, T) = \nu_{if}^L(P/T) \equiv \nu_{if}^{\text{vacuum}} + \nu_{if}^{\text{shift}} \left(\frac{P}{1 \text{ atm}} \right) \frac{296 \text{ K}}{T} . \quad (3b)$$

The MODTRAN6 LBL option is being validated against LBLRTM, so adopting the air density-induced line shift is the natural choice. However, there is a concern that Eq. (3b) may be inaccurate or untested for very low temperatures. The minimum MODTRAN band model temperature for terrestrial calculations is set to 180 K. For the LBL algorithm, this same temperature is used to provide a cut-off on the temperature dependence of the induced line shift:

$$\text{MODTRAN 6 : } \quad \nu_{if}^M(P, T) = \nu_{if}^{\text{vacuum}} + \nu_{if}^{\text{shift}} \left(\frac{P}{1 \text{ atm}} \right) \frac{296 \text{ K}}{\max(T, 180 \text{ K})} . \quad (3c)$$

Returning to the two-bin line center approach, the air density induced line shift can alter which pairs of spectral bins is closest to the transition frequency as pressures and temperatures vary along a LOS. If the molecular transitions contributing to line-tail data for a given spectral bin were allowed to change depending on the pair of (P, T) values, then interpolating line-tail data over pressure and temperature would itself introduce spectral discontinuities. To avoid such problems, each HITRAN line is pre-assigned its two line center bins based on the range of line center positions between the vacuum frequency $\nu^{\text{vac}} \equiv \nu_{if}^{\text{vacuum}}$ and the frequency at 1 atm pressure and 180 K, $\nu_{180} \equiv \nu_{if}^M(1 \text{ atm}, 180 \text{ K})$. The intermediate transition frequency value, ν_{296} , defined as $\nu_{if}^M(1 \text{ atm}, 296 \text{ K})$, is also used in the assignment process. By definition, the spectral bin containing ν_{296} is bin N , where $N \times 0.1 \text{ cm}^{-1} < \nu_{296} \leq (N + 1) \times 0.1 \text{ cm}^{-1}$. The following hierarchical procedure is used to assign the two line center bins:

1. If ν^{vac} and ν_{180} are both in spectral bin N , then
 - a. If $\min(\nu^{\text{vac}}, \nu_{180})$ is closer to bin $N - 1$ than $\max(\nu^{\text{vac}}, \nu_{180})$ is to bin $N + 1$, **select bins $N - 1$ and N**
 - b. Else, **select bins N and $N + 1$**
2. Else if $\min(\nu^{\text{vac}}, \nu_{180})$ is in bin $N - 1$ and $\max(\nu^{\text{vac}}, \nu_{180})$ is in bin N , **select bins $N - 1$ and N**
3. Else if $\min(\nu^{\text{vac}}, \nu_{180})$ is in bin N and $\max(\nu^{\text{vac}}, \nu_{180})$ is in bin $N + 1$, **select bins N and $N + 1$**
4. Else if ν^{vac} (and ν_{296}) are in spectral bin N , then
 - a. If ν_{180} is below bin N , **select bins $N - 1$ and N**
 - b. Else, **select bins N and $N + 1$**
5. Else if ν^{vac} is below bin N , **select bins $N - 1$ and N**
6. Else, **select bins N and $N + 1$.**

For all HITRAN 2012 lines, either criterion 1a, 1b, 2 or 3 is satisfied. If the other criteria were needed, it would imply that the range of line centers spans more than 2 spectral bins. If that had been the case for one of the HITRAN molecular transitions, it would have required that more than 2 bins be assigned as LBL bins for that line.¹

After each HITRAN line is assigned its set of two line-center bins, the HITRAN database is reordered so that all lines assigned to bins 0 and 1 are listed first, followed by all lines assigned to bins 1 and 2, and so on. The first pair, bins 0 and 1, has the most lines, a total of 2,883. The N^{th} record of a binary, direct-access file named HITRAN2013.map contains a single integer equal to the number of the last line that contributed to bins $N - 1$ and N (if no lines contributed to bins $N - 1$ and N , the N^{th} record contains the number of the last line that did contribute to an earlier bin pair). Record #1 of HITRAN2012.map has the integer value 2,883 and Record #473,647, the last record, has the integer value 4,425,596 equal to the total number of HITRAN2012 lines.

The MODTRAN6 LBL algorithm reads in structured data for 4,096 HITRAN lines at a time. Two sets of the structured arrays are stored. Whenever new lines are needed for a given spectral bin pair, they are read into the older of the two structured arrays. The older array is guaranteed to no longer be required since no bin pair contains more than 2,883 ($< 4,096$) lines.

4.2.4.3 Modeling Lorentz Self-broadening

The Lorentz collisional HWHM, γ_c , is a function of air temperature T and pressure P as well as the partial pressure of the absorbing molecule. These dependencies are generally modeled via the equations

$$\gamma_c \equiv \gamma_c(P_{\text{self}}, P, T) = \gamma_c^{\text{air}}(T)(P - P_{\text{self}}) + \gamma_c^{\text{self}}(T)P_{\text{self}} \quad (4a)$$

$$\gamma_c^{\text{air}}(T) \equiv \gamma_c^{\text{air}} \left(\frac{296\text{K}}{T} \right)^n \quad \text{and} \quad \gamma_c^{\text{self}}(T) \equiv \gamma_c^{\text{self}} \left(\frac{296\text{K}}{T} \right)^n .$$

Here, γ_c^{air} and γ_c^{self} are the air- and pressure-broadened half-widths [$\text{cm}^{-1}/\text{atm}$] at 296 K and 1 atm pressure respectively; $\gamma_c^{\text{air}}(T)$ and $\gamma_c^{\text{self}}(T)$ are the corresponding temperature-dependent half-widths at 1 atm pressure; n is temperature dependence exponent. MODTRAN has, to date, modeled the self-broadened half-width as being equal to the air-broadened value, but such an assumption is inappropriate for high spectral resolution LBL modeling. For H_2O , the ratio of the self-broadened half-width to its air-broadened complement, $\gamma_c^{\text{self}}/\gamma_c^{\text{air}}$, is typically of order 5. Since the partial pressure of H_2O at the ground for a moist atmosphere can exceed 2%, the self-broadened correction can lead to a 10% increase in the Lorentz half-width.

The Voigt line-shape is defined as the spectral convolution of the Doppler and Lorentz forms. The challenge associated with implementation of Eq. (4a) is that pre-calculated data, such as the line-tail absorption cross-sections, are defined before atmospheric constituent partial pressures are known and the Voigt line-shape depends on these quantities. The tables of Padé approximant line-tail parameters are defined as functions of air pressure and temperature; these tables would be unwieldy if they were to also depend on partial pressures. One avoids this complication by noting that the Lorentz line-shape varies with pressure only through its dependence on γ_c . Since the Doppler line-shape is independent of pressure, the Voigt line-shape must also only depend on pressure through γ_c .² For a given line, one can define a Lorentzian pressure, P_L , by rearranging Eq. (4a):

¹ Computational speed could be increased by assigning a single line-center bin to the subset of lines which have transition frequencies that remain in the central portion of a single bin; that option complicates programming, so it has not been implemented.

² This is not completely accurate. The line-shape also depends on the pressure-induced line shift; however, the difference between the nominal line-shift and that due to the Lorentzian pressure of Eq. (4b) is inconsequential.

$$\gamma_c = \gamma_c^{air}(T) \left[P + \left(\frac{\gamma_c^{self}}{\gamma_c^{air}} - 1 \right) P_{self} \right] \equiv \gamma_c^{air}(T) P_L \quad . \quad (4b)$$

It follows that the Voigt line-shape for this molecular transition satisfies the relationship

$$f_v(\gamma_c, \gamma_d) = f_v[\gamma_c^{air}(T) P_L, \gamma_d(T)] \quad . \quad (5)$$

This equation implies that one can pre-compute a line-tail database based solely on the Doppler half-width and Lorentz air-broadened half-width, and interpolate the pressure-dependence based on the Lorentzian pressure, P_L , to model self-broadening.

The MODTRAN6 band model line-tail Padé approximant fits of spectral absorption cross-sections over pressure, Eq. (1), are stored at two pressures, 1.0 and 0.1 atm. At high pressures, these line-tails can be approximated by their Lorentzian form:

$$a_{P_L}(\delta_\nu) \approx \frac{1}{P_L} \sum_j \frac{(\gamma_c)_j / \pi}{\Delta_j^2 + (\gamma_c)_j^2} = \sum_j \frac{\left[\gamma_c^{air}(T) \right]_j / \pi}{\Delta_j^2 + \left[\gamma_c^{air}(T) \right]_j^2 P_L^2}; \quad \Delta_j \equiv \nu - (\nu_{if})_j \quad . \quad (6)$$

The summation is over all molecular lines j contributing line-tail absorption to a given band model spectral bin (see Figures 5 and 8). This approximation illustrates why the absorption cross-section curves are divided by pressure; since $\gamma_c = \gamma_c^{air}(T) P_L$, factoring out the pressure dependence in the numerator removes the dominant pressure dependence. The remaining term varies much more slowly with pressure, which facilitates accurate interpolation and enables coarse pressure gridding. For the MODTRAN band model, the line-tail values are linearly interpolated in pressure-squared using the atmospheric pressure, and the resulting value is multiplied by atmospheric pressure to obtain band model line-tail spectral cross-sections, $\sigma_{bm}(\delta_\nu)$:

$$\sigma_{bm}(\delta_\nu) = \left(a_{1.0}(\delta_\nu) + [a_{0.1}(\delta_\nu) - a_{1.0}(\delta_\nu)] \frac{1.00 - P^2}{1.00 - 0.01} \right) P \quad (7)$$

To incorporate the effects of self-broadening, one needs to replace the use of atmospheric pressure, P , for interpolation with the Lorentzian pressure, P_L . There is a problem, however, because the partial pressure coefficient in Eq. (4b), $\gamma_c^{self}/\gamma_c^{air} - 1$, can differ for every molecular line contributing to a spectral bin's line tail absorption. In actuality, the value of the half-width ratio is generally relatively constant or modeled as such if measurements or calculations are lacking. For the MODTRAN6 LBL algorithm, we introduce a new temperature-dependent line-tail band model parameter equal to an absorption cross-section weighted-average value of $\gamma_c^{self}/\gamma_c^{air} - 1$:

$$\left\langle \frac{\gamma_c^{self}}{\gamma_c^{air}} - 1 \right\rangle \equiv \frac{\sum_j \left(\frac{(\gamma_c^{self})_j}{(\gamma_c^{air})_j} - 1 \right) k_{v_cen}[T; (\gamma_c)_j, (\gamma_d)_j]}{\sum_j k_{v_cen}[T; (\gamma_c)_j, (\gamma_d)_j]} \quad . \quad (8)$$

The sums are computed over all lines j contributing to the line-tail parameters for a given spectral bin, and ν_{cen} is the bin's central frequency. The parameter $\langle \gamma_c^{self}/\gamma_c^{air} - 1 \rangle$ is computed for each band model

pressure and temperature grid point, but there is, understandably, no appreciable pressure dependence, so only the values at 1 atm pressure are stored.

For each spectral bin, the temperature-dependent Lorentzian pressure, P_L , is defined by the equation

$$P_L = P + \left\langle \frac{\gamma_c^{self}}{\gamma_c^{air}} - 1 \right\rangle P_{self} . \quad (9)$$

Using P_L for interpolations immediately introduces a new problem. The Lorentzian pressure at the ground often exceeds 1 atm since γ_c^{self} usually exceeds γ_c^{air} . As noted earlier, the original line-tail data only had two pressure grid points, 1.0 and 0.1 atm. Extrapolating below 0.1 atm is not as serious of a problem as extrapolating to pressures above 1.0 atm since line-tail contributions are weaker at lower pressures. Since interpolations are performed on the pressure-squared and since uniform gridding is advantageous, two pressure-square grid points were added. The first was chosen half-way between 1.000 atm² and 0.010 atm², at 0.505 atm²; the second value was chosen to exceed 1.000 atm² at 1.495 atm². Correspondingly, the new set of pressure values is {0.1000, 0.7196, 1.0000 and 1.2227 atm}. No Lorentzian pressure above this upper limit is expected for terrestrial calculations.

The Lorentzian pressure was inserted into Eq. (7) to determine the line-tail cross-sections for the MODTRAN6 LBL algorithm:

$$\sigma_{lbl}(\delta_\nu) = \left(a_i(\delta_\nu) + [a_{i+1}(\delta_\nu) - a_i(\delta_\nu)] \frac{P_i^2 - P_L^2}{P_i^2 - P_{i+1}^2} \right) P_L ; \quad \left\{ \begin{array}{l} P_i \geq P_L \geq P_{i+1} \\ a_i(\delta_\nu) \equiv a_{p_i}(\delta_\nu) \end{array} \right. \quad \left(\begin{array}{l} 1^{st} \text{ interpolation} \\ \text{approach :} \\ \text{Unsuccessful} \end{array} \right) . \quad (10a)$$

The MODTRAN6 LBL algorithm was run and compared to LBLRTM at high spectral resolution (0.001 cm⁻¹). Unfortunately, the residual curves continued to exhibit small discontinuities at bin edges when the interpolation scheme of Eq. (10a) was implemented (this is illustrated below in Figure 13). Exponential interpolation in pressure-squared,

$$\sigma_{lbl}(\delta_\nu) = a_i(\delta_\nu) \left(\frac{a_{i+1}(\delta_\nu)}{a_i(\delta_\nu)} \right)^{\frac{P_i^2 - P_L^2}{P_i^2 - P_{i+1}^2}} P_L ; \quad P_i \geq P_L \geq P_{i+1} \quad \left(\begin{array}{l} 2^{nd} \text{ interpolation} \\ \text{approach :} \\ \text{Unsuccessful} \end{array} \right) , \quad (10b)$$

fared no better. An examination of the test case exhibiting the largest discontinuity showed that it was occurring at a spectral bin edge dominated by the contribution from a single strong line. On one side of the edge, the line was modeled explicitly using a direct LBL calculation; on the other side, the line-tail data was interpolated in pressure-squared. Eq. (6) was revisited to consider the case of a single strong dominant line; in that case,

$$a_p(\delta_\nu) \approx \sum_j \frac{(\gamma_c^{air})_j / \pi}{\Delta_j^2 + (\gamma_c^{air})_j^2 P^2} \approx \frac{(\gamma_c^{air})_J / \pi}{\Delta_J^2 + (\gamma_c^{air})_J^2 P^2} ; \quad (\text{Single strong line}) , \quad (6')$$

where $j = J$ is the dominant line. In this case, the reciprocal of $a_p(\delta_\nu)$ is a linear function of pressure-squared. This dependence suggests that the *reciprocal* of the line-tail absorption cross-section (divided by P) should be linearly interpolated in pressure-squared:

$$\sigma_{lbl}(\delta_v) = \frac{P_L}{\frac{1}{a_i(\delta_v)} + \left[\frac{1}{a_{i+1}(\delta_v)} - \frac{1}{a_i(\delta_v)} \right] \frac{P_i^2 - P_L^2}{P_i^2 - P_{i+1}^2}} ; \quad \left\{ \begin{array}{l} P_i \geq P_L \geq P_{i+1} \\ \frac{1}{a_i(\delta_v)} = P_i \left(\frac{1 + d_1 \delta_v + d_2 \delta_v^2}{\sigma_0 + \sigma_1 \delta_v + \sigma_2 \delta_v^2} \right) \end{array} \right. \quad (10c)$$

(3rd interpolation approach : *Successful*)

When this interpolation method was implemented, all significant observed transmittance discontinuities *in near surface calculations* were eliminated. One discontinuity is observed in residual plots between MODTRAN LBL and LBLRTM, but its transmittance magnitude is only 0.001 (see description of Figure 13 below).

4.2.4.4 Modeling Doppler Line-Tail Contributions

At lower pressures, Doppler contributions become important. Even though the line tails are less important at lower pressures, their contributions remain relevant. Not surprisingly, the expression for $P_L / \sigma_{lbl}(\delta_v)$, based on linear interpolation in pressure-squared between pressure grid points, does not remain accurate for higher altitude (lower pressure) path segments. Since the line tails are computed at 4 pressure points, uniformly spaced in P^2 , a set of four 4-pt interpolation/extrapolation formulas were examined to see which approach minimized the spectral discontinuities. Each of the four proposed methods included a constant and pressure-squared term so that the dominant dependence of Eq. (10c) could be retained:

a) 4-Point Spline Interpolation in P-squared

$$g(P_L) \equiv \frac{P_L}{\sigma_{lbl}(\delta_v)} = A + B P_L^2 + C P_L^4 + D P_L^6 \quad (11a)$$

b) 4-Point Asymptotic Series in P-squared

$$g(P_L) \equiv \frac{P_L}{\sigma_{lbl}(\delta_v)} = \frac{D}{P_L^4} + \frac{C}{P_L^2} + B + A P_L^2 \quad (11b)$$

c) 4-Point Padé Approximate in P-squared

$$g(P_L) \equiv \frac{P_L}{\sigma_{lbl}(\delta_v)} = A + B P_L^2 + \frac{C}{D + P_L^2} \quad (11c)$$

d) 4-Point Spline Interpolation in P [with $P_i^2/\text{atm}^2 = 0.01 + (4 - i) 0.495$; $g_i \equiv g(P_i)$; $i = 1, 2, 3, 4$]

$$\begin{aligned} g(P_L) &\equiv \frac{P_L}{\sigma_{lbl}(\delta_v)} = A + B P_L + C P_L^2 + D P_L^3 \\ &= \frac{(P_L - P_2)(P_L - P_3)(P_L - P_4)}{(P_1 - P_2)(P_1 - P_3)(P_1 - P_4)} g_1 + \frac{(P_L - P_1)(P_L - P_3)(P_L - P_4)}{(P_2 - P_1)(P_2 - P_3)(P_2 - P_4)} g_2 \\ &\quad + \frac{(P_L - P_1)(P_L - P_2)(P_L - P_4)}{(P_3 - P_1)(P_3 - P_2)(P_3 - P_4)} g_3 + \frac{(P_L - P_1)(P_L - P_2)(P_L - P_3)}{(P_2 - P_4)(P_2 - P_3)(P_2 - P_1)} g_4 \end{aligned} \quad (11d)$$

The 4-point spline fit in pressure-squared, method (a), is a series expansion generalization of the quadratic fit. Since the $g(P_L)$ term is the denominator of the line tail absorption coefficient, $\sigma_{lbl}(\delta_v)$, the asymptotic expansion, method (b), introduces a P_L^4 factor in the numerator when P_L is small; this expansion insures

that the line tail rapidly drops towards zero as the pressure decreases. Padé approximates often exhibit better convergence than polynomial fits, so method (c) was introduced; the drawback of the Padé approach is that one must protect against the denominator approaching zero.

These first 3 methods all model $g(P_L) \equiv P_L / \sigma_{bl}(\delta_\nu)$ as a function of Lorentzian pressure-squared. Indeed, the form of Eq. (2a) verifies that P^2 dependence for positive pressures since the Lorentz half-width is proportional to pressure and the Doppler half-width is independent of pressure. However, the Doppler limit is independent of pressure. This suggests that as pressure approaches zero, many terms (actually, an infinite number) are required to make $P_L / g(P_L) \equiv P_L / \hat{g}(P_L^2)$ converge to a pressure-independent value. The 4-point spline interpolation in P_L was introduced to help improve convergence of $g(P_L)$ to a value proportional to pressure.

All four methods were tested by focusing on an observed discontinuity in optical depth at 6704.6 cm^{-1} arising from a path containing 44.43 atm-cm of H_2O at 240.1 K and a Lorentzian pressure of 0.3095 atm. Short of 6704.6 cm^{-1} , the LBL calculations account for more than 90% of the optical depth; long of 6704.6 cm^{-1} that contribution is modeled via the pre-computed line tails interpolated over pressure and temperature. As illustrated in the inset on the left in Figure 9, all the pressure-interpolation techniques produce curves that pass through the gridded (magenta diamonds) values. For pressures above ~ 0.7 atm ($P_L^2 = 0.49 \text{ atm}^2$), they all yield similar interpolated values. However, as Doppler contributions become increasingly important, the methods diverge. The asymptotic interpolation, method (b), completely fails, producing negative line tails between 0.01 and 0.83 atm^2 ; based on these results, this form was eliminated from further consideration.

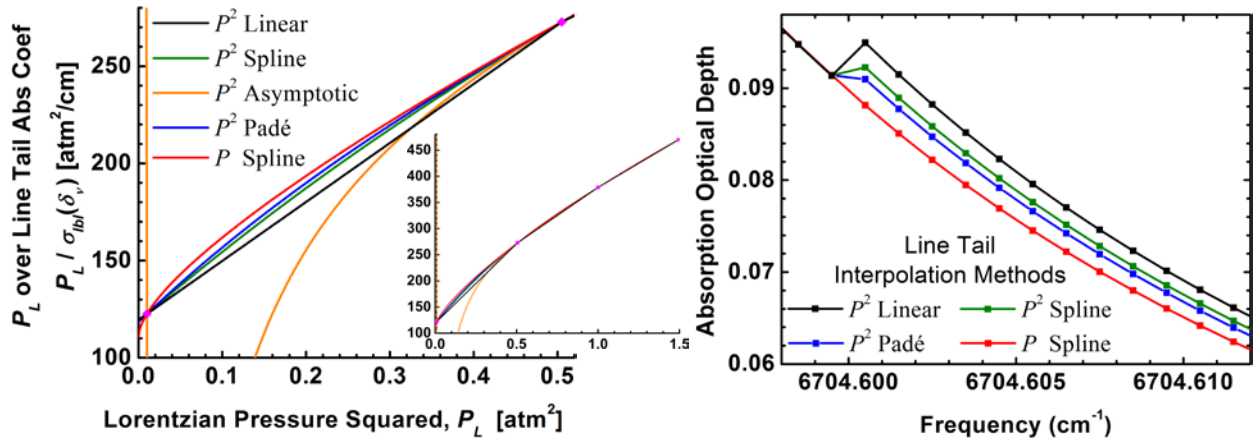


Figure 9. (left) Candidate Interpolation/Extrapolation Curves for Fitting Line Tail Absorption

The other 3 methods produce physically reasonable curves (Figure 9 left), with $g(P_L)$ values exceeding the linear fit for P_L^2 values between 0.010 and 0.505 atm^2 ; this translates to smaller optical depths in this region. These three 4-point methods and the linear interpolation all extrapolate to a positive constant value as P_L approaches zero atm with $g^{\text{linear}}(0) = 119.486$, $g^{\text{P2spline}}(0) = 118.825$, $g^{\text{P2pade}}(0) = 118.399$, and $g^{\text{Pspline}}(0) = 110.434$. This would suggest that the line tail optical depth drops to zero in the Doppler limit. While the Doppler line tail is not exactly zero, it does become inappreciably small.

To select between the 4 interpolation techniques (P_L^2 linear, P_L^2 spline, P_L^2 Padé and P_L spline), spectral absorption optical depth curves near 6704.6 cm^{-1} are plotted in Figure 9 (right) for the selected test case scenario (44.43 atm-cm of H_2O at 240.1 K and a Lorentzian pressure of 0.3095 atm). As the figure illustrates, the P_L^2 linear interpolated optical depth exceeds the exact value by more than 0.007 (8% too high) at 6704.6005 cm^{-1} . The P_L^2 spline and P_L^2 Padé methods do better, but they still predict too much absorption optical depth by 0.004 and 0.003, respectively. For this one case, at least, the P_L spline

interpolation clearly yields the best result, a smooth transition from LBL to line-tail pressure interpolation. Further testing confirms that the P_L spline interpolation method is best.

4.2.5 Validations of the MODTRAN6 Line-By-Line and Band Model RT Approach

Initial 8.0 to 12.0 μm band calculations have been run to validate the new MODTRAN6 LBL algorithm against LBLRTM and, in turn, to validate the MODTRAN6's band model against its new LBL algorithm. The band model validations are based on the scenario of Figure 6, a nadir view from 20 km altitude through the MLS model atmosphere. The scenario for validation against LBLRTM is similar, but the USS model atmosphere was chosen and the view angle was changed to 60° off-nadir to increase opacity and the influence of upper altitudes. The results are described in the following two sub-sections.

4.2.5.1 Validation of MODTRAN6 LBL against LBLRTM

Figure 10 contains the 8.0 to 12.0 μm transmittance comparison of MODTRAN6 LBL to LBLRTM at 0.001 cm^{-1} spectral resolution. For frequencies below $\sim 1180 \text{ cm}^{-1}$ ($> 8.47 \mu\text{m}$), the vast majority of the transmittance residuals have a magnitude less 0.01; the residuals ramp up to ~ 0.03 near 1240 cm^{-1} ($8.06 \mu\text{m}$). Individual spectral points exhibit large differences between the two LBL models. Questions that need to be answered include:

- Why do low spectral resolution base-lines differ?
- What is the source of the *residual hash* in the 9.6 μm O_3 band spectral region?
- Why are there large residuals at sporadic and isolated spectral points?

The narrow spectral regions highlighted with green boxes are examined in more detail to shed light on these issues.

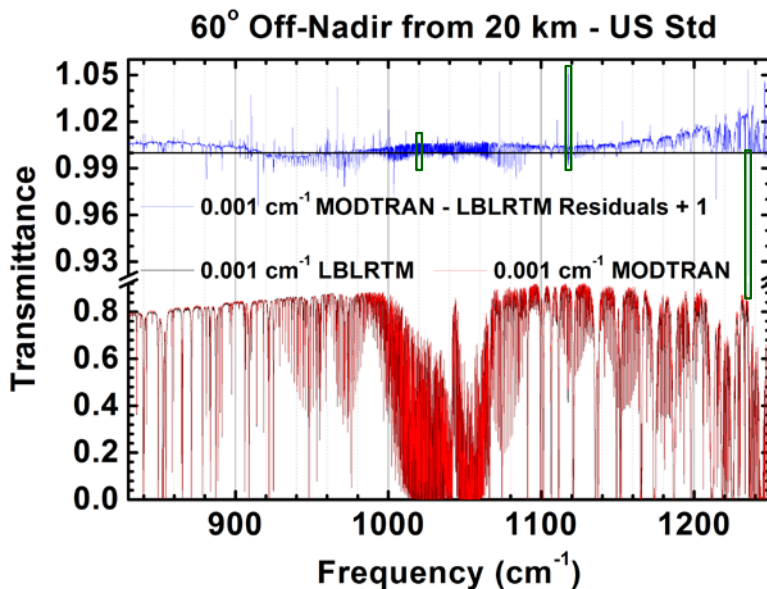


Figure 10. Validation of the MODTRAN6 LBL Algorithm (red) Against LBLRTM (black)

Figure 11 contains the plot of the *residual hash* between 1020.0 and 1020.9 cm^{-1} . One immediately observes that the MODTRAN (red) and LBLRTM (black) curves are very close; this good agreement is difficult to decipher from overcrowded spectra of Figure 10. The residual blue curve has an upper baseline value near 1.006; the upper baseline value occurs for transmittances between ~ 0.25 and ~ 0.65 . MODTRAN transmittances are too high, absorptivity too low. This translates to an error in absorption optical depth of ~ 0.013 [$\approx \ln(0.456 / 0.450)$]. At low transmittances (< 0.1), the MODTRAN and LBLRTM values are in good agreement. For a transmittance of 0.1, an optical depth error of 0.013

translates to a small transmittance error of 0.0013 $\{ \approx 0.1 - \exp [\ln(0.1) - 0.013] \}$. This seems to suggest that the *residual hash* arises from a relatively constant absorption optical depth difference. The only significant low-resolution extinction source that was modeled differently by MODTRAN6 and the current release of LBLRTM in these calculation was the H₂O continuum. At this point in its development, MODTRAN relied on the Clough-Kneizys H₂O continuum data extracted from LBLRTM in circa 2000. LBLRTM uses the state-of-the-art MT_CKD H₂O continuum [9,10,11]. The H₂O continuum in MODTRAN6 has since been updated to MT_CKD. This upgrade eliminated the low-resolution optical depth differences between the two models, the source of the *residual hash* in the 9.6 μm O₃ band spectral region.

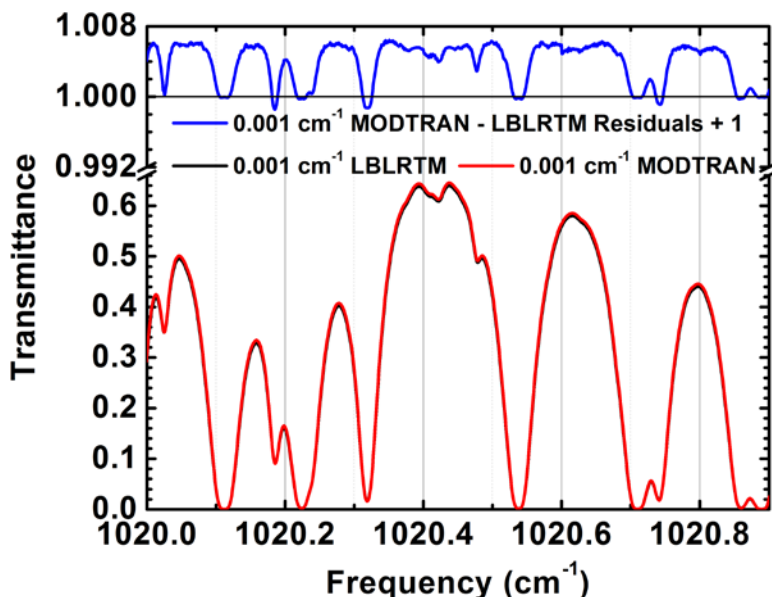


Figure 11. A Narrow Sub-region of the 9.6 μm O₃ Band Exhibiting the ‘Residual Hash’ of Figure 10

It should be noted that a small bin edge discontinuity is present in Figure 11. There is no evidence of this discontinuity in the MODTRAN transmittance plot itself. However, the residual plot, on its expanded vertical scale, does exhibit a jump of magnitude of ~ 0.001 at 1020.6 cm^{-1} .

The residual of greatest magnitude in Figure 10 occurs near 1235.2 cm^{-1} ($8.096 \mu\text{m}$). The MODTRAN6 LBL to LBLRTM comparison is re-plotted for a 0.2 cm^{-1} spectral region centered on the location of this residual in Figure 12. The calculations exhibit a 0.025 baseline transmittance difference, suggesting that the MODTRAN6 H₂O continuum optical depth for this calculation is ~ 0.03 $[\approx \ln(0.850/0.825)]$ less than the MT_CKD value (this baseline was eliminated with the introduction of MT_CKD into MODTRAN6). The absorption at 1235.2 cm^{-1} results from a doublet of H₂O lines. The two LBL calculations differ because they use different line compilation data. The MODTRAN6 LBL algorithm relies directly on HITRAN2012 data. AER, the distributors of LBLRTM, tailor the data for individual HITRAN lines based on their validations against field measurements. For the lines in question, the total HITRAN line strength, $3.2 \times 10^{-25} \text{ cm/molec}$, is about 10% higher than the LBLRTM value, $2.9 \times 10^{-25} \text{ cm/molec}$. The half-widths and lower state energies are the same in the two line files, but AER assigns slightly higher frequencies to the line centers. These changes produce the spectral transmittance difference illustrated in Figure 12.

A final residual from Figure 10 is highlighted in Figure 13. This residual, located at 1117.7 cm^{-1} ($8.947 \mu\text{m}$), corresponds to the largest MODTRAN minus LBLRTM transmittance difference. In this case, AER has introduced very small changes to the H₂O line’s transition frequency and line strength, but the air-broadened half-width has been decreased by 20%. This produces a significant increase in the line center absorption, as illustrated.

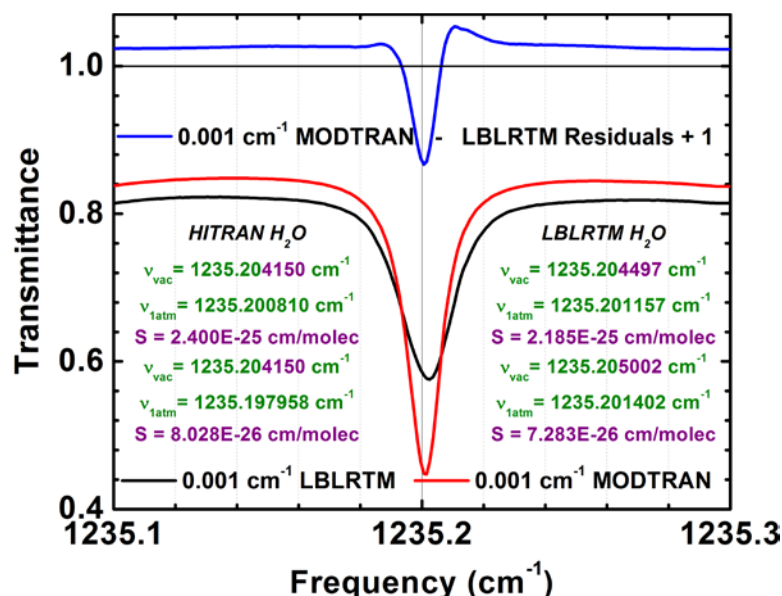


Figure 12. A Narrow Sub-Region Containing the Largest Magnitude Residual from Figure 10.

Figure 13 also contains small snippets of the residual curve (in green) that was obtained when the Padé line-tail data was interpolated linearly in pressure-squared, Eq. (10a). A significant (0.003) discontinuity is present at the 1117.7 cm^{-1} bin edge. Upgrading to a four-point spline fit in pressure for the pressure divided by the Padé line-tail absorption coefficient, Eq. (11d), completely eliminates both this continuity and the smaller one located two spectral bins away, at 1117.5 cm^{-1} .

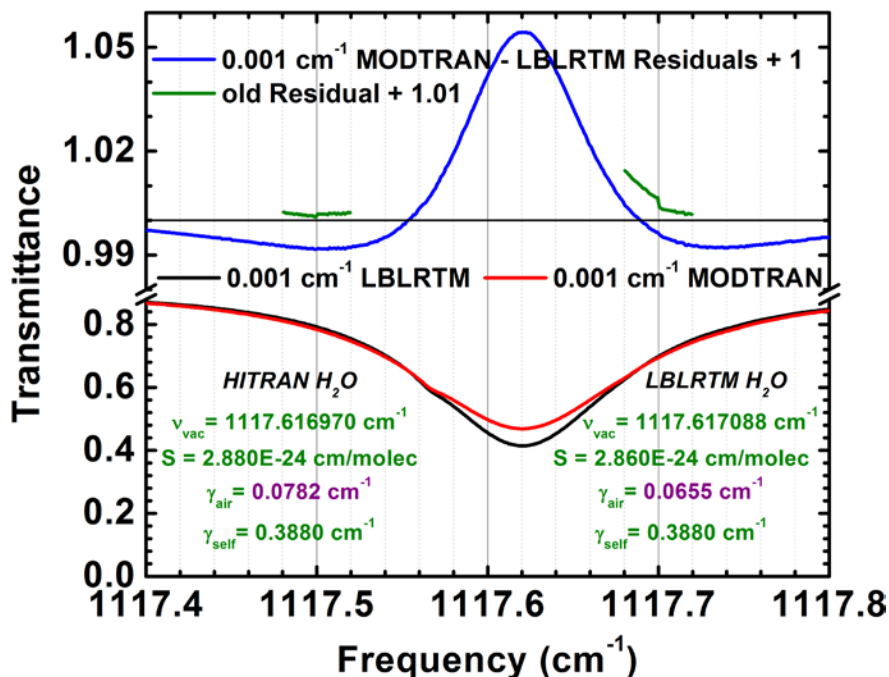


Figure 13. Figure 10 Residual Arising from Differences in Air-Broadened Half-Widths

4.2.5.2 Validation of the MODTRAN6 Band Model against MODTRAN6 LBL

In Sub-section 4.2.3.2 and Figure 6, the MODTRAN band model line-center transmittances were validated against the initial MODTRAN6 LBL results. In some respects, these were unfair comparisons. The LBL line center calculations were performed for the true molecular transition frequencies, but the band model line center data included shifted line centers. The calculations used the same set of line-tail data. For the correct validation, one must compare the 0.1 cm^{-1} MODTRAN6 band model molecular transmittances against the MODTRAN6 LBL molecular transmittances convolved over the same 0.1 cm^{-1} bins, rather than just comparing the line center transmittance contributions. Those comparisons are presented in this section.

Nadir calculations from 20 km through the MLS model atmosphere are plotted in Figure 14. In contrast to Figure 6, the data are plotted as a function of frequency not wavelength. This was done to facilitate selection of sub-regions for further investigation. The goal here is to understand the source of the MODTRAN band model errors so that refinements to the statistical approach can be proposed, developed and validated. Three of the largest residuals are examined. They are highlighted by orange boxes in Figure 14.

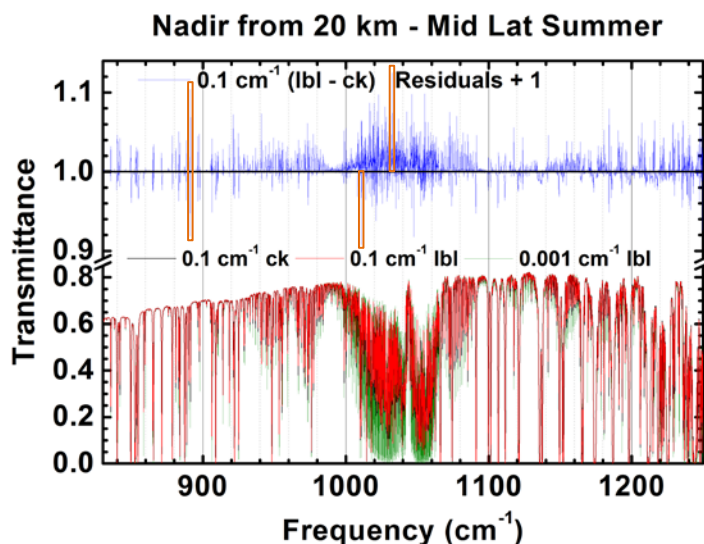


Figure 14. Validation of the MODTRAN6 Band Model against the MODTRAN6 LBL Algorithm

Figure 15 contains the updated version of Figure 6 (right) using the final MODTRAN6 LBL formalism. All of the spectral bin edge discontinuities are gone (it may appear as if some discontinuities are still present, but examination of the high resolution data verifies that all changes in slope actually occur near but not at the bin edges). The large transmittance residual at 1032.45 cm^{-1} has the same magnitude (0.13) as before. The shifting of the three H_2O lines located near the 1032.5 cm^{-1} by the band model approach still produces an error (see Sub-section 4.2.3.2 for the detailed description). Correcting the LBL approach did not have any appreciable effect of the magnitude of the residual.

A pair of positive and negative transmittance residuals of substantial magnitude (~ 0.1) are observed for the two spectral bins sharing the 891.3 cm^{-1} edge, Figure 16. The source of this absorption is an isolated H_2O line with vacuum and 1 atm pressure transition frequencies of 891.30071 and 891.29522 cm^{-1} , respectively, at 296 K. The air pressure-induced line-shift pushes the line center into the 891.25 cm^{-1} centered spectral bin. This is a classic case of a line straddling a bin edge. The band model shifts the molecular spectral inward to 891.275 cm^{-1} . This produces too much absorption for the 891.25 cm^{-1} centered bin and too little for the 891.35 cm^{-1} centered bin. This is once again an example where the band model line shifting is generating unacceptably large errors.

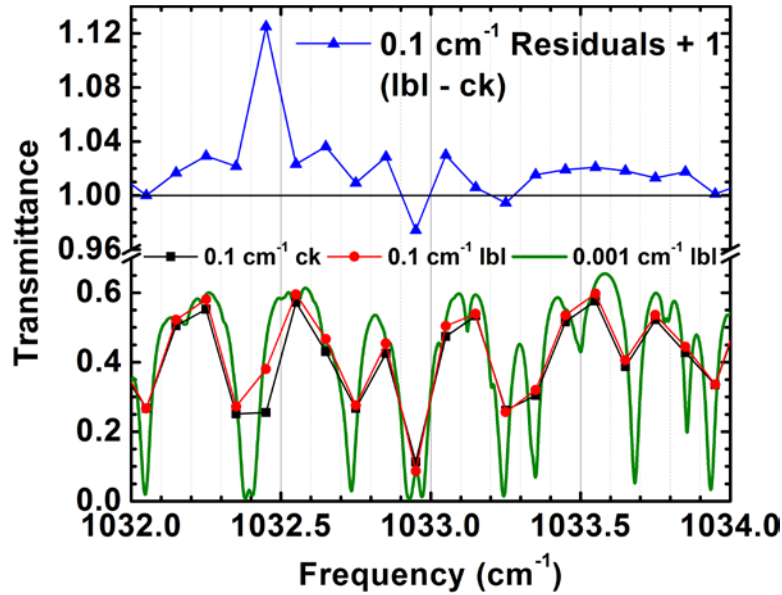


Figure 15. Same Conditions as Figure 14, but Highlighting Residuals at 1032.45 cm^{-1}

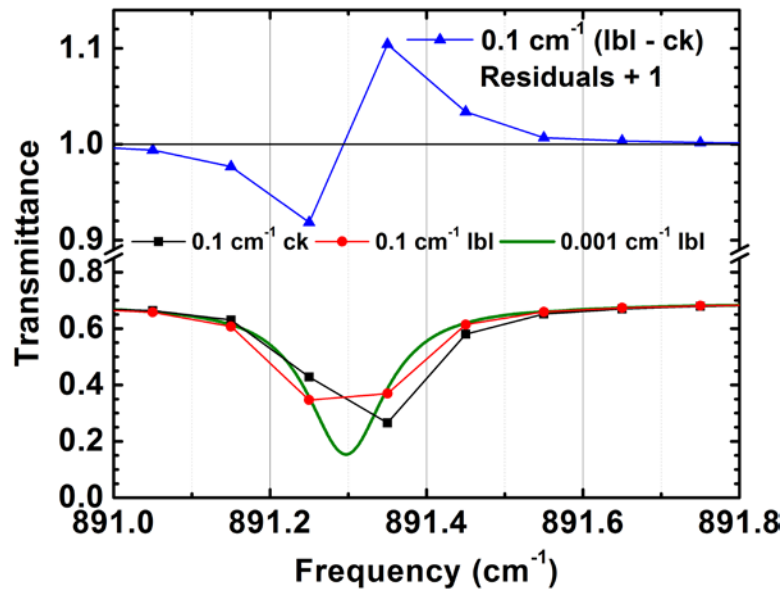


Figure 16. Same Conditions as Figure 14, but Highlighting Residuals at 891.3 cm^{-1}

Possibly, the most interesting 0.1 cm^{-1} spectral bin transmittance residual case is the one illustrated in Figure 17. As before, there is a H_2O line straddling a bin edge with a 1 atm, 296 K air pressure-induced line shifted transition frequency of 1010.79863 cm^{-1} . The band model shifting of this line center towards the center of the 1010.75 cm^{-1} bin produces too much BM absorption in that bin, and too little in the higher frequency neighboring bin.

There are also multiple strong O_3 absorption lines in the region, the strongest being a line centered near 1010.835 cm^{-1} . The 1 atm, 296 K air-broadened Lorentz half-width for this line, γ_c , is 0.0723 cm^{-1} , substantially larger than the 0.0425 cm^{-1} air-broadened Lorentz half-width of the H_2O line. Even so, the absorption from O_3 lines is much *less* broad than that from the H_2O line. The reason is fairly obvious. The geometry of the observation is nadir from 20 km. A significant fraction of the ozone absorption occurs at the higher altitudes due to the relatively constant O_3 profile over the altitude range of the simulation. The

H₂O, on the other hand, primarily resides in the boundary and lower tropospheric layers. Since the Lorentzian width varies proportional to pressure, the O₃ absorption features are much narrower than the H₂O absorption feature.

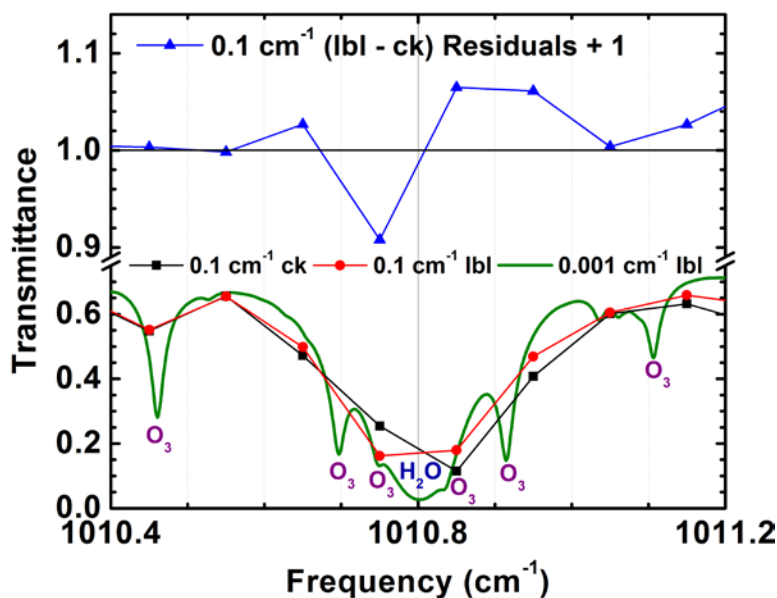


Figure 17. Same Conditions as Figure 14, but Highlighting Residuals at 1010.8 cm⁻¹

The overlap between H₂O and O₃ absorption within the two spectral bins sharing the 1010.8 cm⁻¹ edge introduces another concern, and source of error when compared to the LBL calculations. The MODTRAN band model calculations assume that the absorption from all molecular species is randomly correlated. In practice, that means the product of their path transmittances equals the combined species transmittance. That is a reasonable approximation for the spectrum of Figure 17. However, the band model calculation was run using MODTRAN's statistical Correlated-*k* algorithm. With CK, the working assumption regarding line overlap is different. CK models absorption as being spectrally correlated among path segments. In other words, the spectral location within a band model bin of the strongest absorption source for path segments near the ground needs to be the same as its location for upper altitude path segments. Clearly, the H₂O and O₃ absorption features are not spectrally correlated for the two spectral bins under consideration. One must conclude that the lack of spectral correlation is a second source of error in the comparisons against the LBL data.

Figure 18 contains a comparison of MODTRAN6 Ck and LBL spectral radiances, run with the Isaacs two-stream multiple scattering algorithm. A nadir view through the 1976 U.S. Standard atmosphere was simulated from 20 km altitude with a 60° solar zenith. The very hazy (visibility 5 km) rural aerosol model was selected and the ground was modeled as a Lambertian reflector with constant 10% surface albedo. Although calculations were performed from 475 to 745 nm, Figure 18 only displays the results between 487.0 and 488.4 nm for visual clarity. The graph contains the 0.1 cm⁻¹ correlated-*k* radiance (thick black curve), the 0.1 cm⁻¹ LBL radiance (dashed red curve), the high resolution LBL radiances (thin blue curve) and the absolute difference between the 0.1 cm⁻¹ curves (green curve). Outside of spectral absorption bands, the baseline radiances are ~8 μW cm⁻² sr⁻¹ / nm; Ck minus LBL residuals approach ±0.3 μW cm⁻² sr⁻¹ / nm. These residuals arise primarily from the line position offsets of the line tail data used in MODTRAN's band model and Ck algorithms.

Figure 19 shows a comparison of the Ck (black curve with squares), bin-averaged LBL (red curve with squares), and high resolution (0.001 cm⁻¹) LBL (olive curve) spectral radiances, all computed with the DISORT multiple scattering algorithm. The Ck and degraded LBL results were computed with a 0.1 cm⁻¹

non-overlapping square slit. A 3 km altitude observer peers downward with a 45° off-nadir angle at the 10% reflective ground; the 30° sun is in the principal plane in front of the observer (105° scattering angle). As with Figure 18, the atmosphere is defined by the 1976 U.S. Standard model and MODTRAN's 5 km visibility rural boundary layer aerosol. The graph shows a close-up of near-IR radiances in the spectral range from 13648.3 to 13650.7 cm⁻¹ (732.563 to 732.692 nm), which includes 2 strong H₂O lines. The HITRAN [5] vacuum and 1 atm pressure transition frequencies for the lower frequency line are $\nu^{vac} = 13,648.707488$ and $\nu_{296} = 13,648.685268$ cm⁻¹; the values for the higher frequency line are $\nu^{vac} = 13,650.026253$ and $\nu_{296} = 13,650.008303$ cm⁻¹. The band model data used in MODTRAN's statistical Ck algorithm is defined based on vacuum transition frequencies. For the lower frequency transition, this assignment centers the line in the wrong spectral bin, the 13,748.7 to 13,748.8 cm⁻¹ bin even though the shifted line center is below 13,748.7 cm⁻¹ (see the true line center as portrayed in the high resolution LBL curve in Figure 19). Furthermore, MODTRAN avoids placing line centers too close to a bin edge, so that the modeled line position is actually 13,748.725 cm⁻¹. Given these mis-assignments, the relative large residuals shown in blue are not unexpected. The higher frequency H₂O transition is assigned to the correct spectral bin, but its true location, being very close to the bin edge, is translated and still produces relatively large Ck radiance residuals. The MODTRAN6 LBL capability makes these types of analysis much easier to perform.

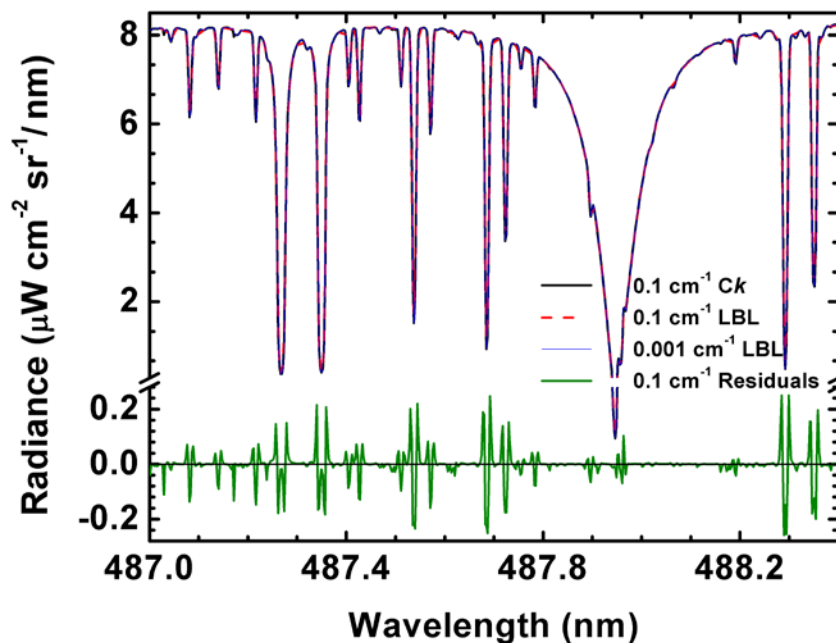


Figure 18. Comparison of MODTRAN6 Ck and LBL Radiances with Isaacs 2-Stream Scattering

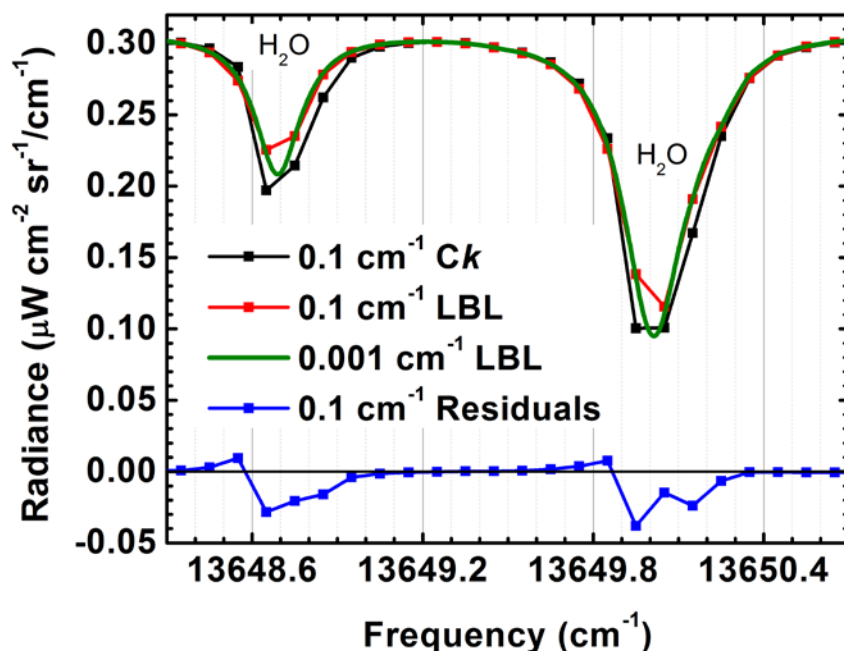


Figure 19. Comparison of MODTRAN6 Ck and LBL Radiances with DISORT 8-Stream Scattering

4.2.6 Summary and Future Work

A novel approach for performing LBL spectral transmittance calculations using MODTRAN6 has been developed and validated. The algorithm is unique in that high spectral resolution calculations are performed independently within 0.1 cm^{-1} spectral bins, and these pockets of spectral data are subsequently stitched together. The processing is efficient because on-the-fly spectral line-shape calculations at each spectral grid point are only performed for lines centered within a 0.2 cm^{-1} domain containing the spectral point. Line-tail absorption cross-sections are pre-computed on a fine temperature and coarse pressure grid for each molecule. A Lorentzian pressure band model parameter is introduced for accurately modeling self-broadening. Spectral discontinuities at bin edges are a potential problem, but essentially eliminated by a 4-term spline fit in pressure of the ratio of pressure to line-tail cross-sections.

Validations of the MODTRAN6 LBL algorithm against LBLRTM found two sources of discrepancies. Large residuals for individual lines arose because of differences in the line data files. Future comparisons will be made by utilizing common lines files for two LBL models. Low resolution residuals arose due to different molecular continua. MODTRAN6 has since been upgraded with the current version of the MT_CKD H_2O continuum. Remaining continua utilized by LBLRTM will also be integrated into MODTRAN6.

The MODTRAN6 LBL algorithm development and the MODTRAN6 band model validation efforts both led to recommendations for the band model. The development effort illustrated how the band model can be upgraded to include the effects of Lorentz self-broadening, to model air density-induced line shifts and to improve the interpolation of line-tail data. The validation test cases highlighted problems associated with shifting line centers to avoid bin edges, and with assuming spectral correlation; both of these problems will require additional study to find effective solutions.

4.3 MODTRAN6 Automated Testing Process and Tools

As part of the ongoing MODTRAN verification process, SSI monitors code changes not only with SVN configuration control, but also with a MODTRAN6 comparebinaries program and with *Buildbot* (<http://buildbot.net/>) software. These latter two tools are described in this subsection.

4.3.1 The comparebinaries Change Detection Software

A MODTRAN6 `comparebinaries` program has been developed to compare binary output files generated by a pair of distinct MODTRAN executables and run with the same input stream. This program tracks changes to output when any modifications are made either in the source files or to the build configuration. `comparebinaries` performs file comparisons for the output files produced for the full suite of MODTRAN test case inputs, collects the results of these comparisons, and prints a table of differences organized by file type and rootname, Table 5. It also provides a facility to convert MODTRAN6 binary output files to ASCII when they differ, so that the MODTRAN software developer can examine the residuals.

The test program is invoked from the command line as

```
comparebinaries dirpathStd dirpathNew {-btoa{=N}}
```

The first two arguments, *dirpathStd* and *dirpathNew*, are required and specify the directory paths to the *standard* and *new* output file sets, respectively. The last optional argument, `-btoa{=N}` (binary to ASCII), specifies how extensively binary files are to be converted to ASCII. If present, *N* should be a digit in the range from 0 (no conversions) through 4 (convert all binary files) with

- N* = 0 ⇒ No conversions to ASCII
- N* = 1 ⇒ Convert to ASCII only those output files with *physically significant* differences
- N* = 2 ⇒ Convert to ASCII only those files with differences *above the numerical noise threshold*
- N* = 3 ⇒ Convert to ASCII only those binary output files *which differ*
- N* = 4 ⇒ Convert *all binary output files* to ASCII

If `-btoa` is used without a number, the default behavior is the same as for `-btoa=2`.

The `comparebinaries` program also requires a `mod5root.tst` text file in the current working directory. The `mod5root.tst` file lists the MODTRAN runs for which output is to be compared. The format of the `mod5root.tst` file is the same as that used by the MODTRAN `mod5root.in` input file. It lists the root names of output file sets to be compared, one per line. The `comparebinaries` program reads and processes successive root names from the `mod5root.tst` file until it reaches the end of the file or reads in a blank line. For each rootname, `comparebinaries` determines which binary output file types are present in both the standard and new directories, and then performs comparisons between the standard and new files, run-by-run or section-by-section. Results of the comparisons are stored in a linked list of data structures for later testing and printing. The comparison metrics saved for each run or section of a file include:

- The total number of *independent* and dependent data items (e.g., when considering spectral data, the wavelengths and frequencies are classified as an *independent* data item, while the spectral transmittances and radiances are classified as a *dependent* data item);
- The number of *independent* and *dependent* data items not identical between files;
- The total number of data differences for which the relative error is greater than the numerical-noise threshold ϵ_{rel} (currently set to 5×10^{-6}); and
- The total number of data differences large enough to be physically significant. This test is only done for records in the tape7 binary files. A physically significant difference is defined as (1) any difference in transmittances over one part in 10^4 , or (2) any difference in radiance components over one part in 10^4 , if the brightness temperature is above 50 K.

After all root names have been processed, `comparebinaries` generates the sorted table of data differences. Table 5 contains an example of `comparebinaries` output.

Table 5. Typical Output From comparebinaries Utility

comparebinaries.f90								
=====								
Compare sets of Modtran binary output files.								
Standard directory: /home/MODTRAN/COMPARE/mod53g95COMPARE/								
New directory: TEST_b/r457_g95z/								
Using relative error threshold eps_r = 5.0E-06. No conversion of binary files to ascii.								
for BINwrite compare:								
ascii and binary tape7 files								
binary tape8 file								
binary plot file								
NOTE subroutine cmp_btp8files is stubbed.								
No tests performed here, returning OK status.								
for lut01BIN compare:								
ascii and binary tape7 files								
binary plot file								
binary flux file								
binary ACD file								
for UserPathBinary compare:								
ascii and binary tape7 files								
binary corr-k radiance file								
binary corr-k transmittance file								
=====								
For TAPE7 files:								
Independent data values are spectral bin centers.								
Dependent data depend on the value of IEMSCT, but include transmittances,								
radiances, fluxes, and brightness temperatures.								
For TAPE8 files:								
Independent data values are (still to add).								
Dependent data values are (still to add).								
For PLOT files:								
Independent data values are spectral points (units can vary from run to run).								
Dependent data are either transmittances or radiances.								
For FLUX files:								
Independent data values are spectral points.								
Dependent data are spectral fluxes at atmosphere levels (3 per level).								
For ACD files:								
Independent data values are spectral points.								
Dependent data are correlated-k subinterval weights, transmittances,								
and the spherical albedo at bottom of atmosphere.								
For CORRELATED-K files:								
Independent data values are spectral bin centers, and path altitudes								
and ranges at LOS segment boundaries.								
Dependent data are correlated-k subinterval weights, and either radiances								
or transmittances on subintervals (depending on file type).								
=====								
Rootname	File	Run	N Indep.	N Dep.	N Indep.	N Dep.	Errors.....	
	Type		Values	Values	Errors	Any	Over eps_r	Physical
=====								
TAPE7 FILES:								
BINwrite	tp7	1	1276	19140	0	5473	1295	319
lut01BIN	tp7	1	304164	4562460	0	1570027	265235	25729
UserPathBinary	tp7	1	41	615	0	0	0	0
UserPathBinary	tp7	2	41	615	0	0	0	0
PLOT FILES:								
BINwrite	plt	1	1276	1276	0	390	2	
lut01BIN	plt	1	4828	304164	0	297271	48705	
FLUX FILES:								
lut01BIN	flx	1	1074	70950	0	54600	23926	
ACD FILES:								
lut01BIN	acd	1	304164	1824984	0	954044	120342	
CORRELATED-K RADIANCE FILES:								
UserPathBinary	ckr	1	615	4558	0	0	0	
UserPathBinary	ckr	2	615	4558	0	0	0	
CORRELATED-K TRANSMITTANCE FILES:								
UserPathBinary	ckt	1	533	3901	0	0	0	
UserPathBinary	ckt	2	533	3901	0	0	0	

If the `-btoa` option has been used, then `comparebinaries` will also generate `cvtb_to_a` Bourne Shell scripts in both the standard and new data directories, to convert binary files to ASCII. The analyst may compare them with WinMerge (<http://winmerge.org>), an open source differencing and merging tool for Windows, or other similar utilities in order to find which records and items differ.

4.3.2 The `buildbot` Code Monitoring Software

A functioning continuous integration environment exists for testing and comparing output files produced by MODTRAN6. The environment primarily consists of `modtrantest.py`, a custom generated Python™ (<https://www.python.org/about/>) test script which interfaces with an installation of the continuous integration software entitled *Buildbot*, (<http://buildbot.net/>). The test script and the *Buildbot* software are currently installed and running on an SSI Linux server.

The test script reads in and stores data from any of ASCII output files generated by MODTRAN6, including *.tp7, *.tp8, *.r_k, *.t_k, and *.chn files. The script accounts for the different output formats within each output file, such as the existence of multiple lines of sight. The script then compares and plots the data. Comparisons are made between data sets that are fixed and not fixed. The specific comparisons performed and the plots generated depend on the output file. For example, for every run within a *.tp7 file, the comparison script reads and stores the frequency as well as the dependent values for the properties corresponding to the specified frequency. Afterwards, the script reads and stores the respective values for the corresponding fixed data set. The script compares the dependent values, separating them by property, frequency, and line-of-sight. A distinct set of comparisons and threshold values are used depending on the specific property being compared. Specifically, properties are separated depending on whether they are considered to be transmittances or radiances. For example, all transmittances are compared based on absolute differences; physically significant differences in transmittances are recorded if the values are above an epsilon value of 0.0001. If the differences for the dependent values within a given *.tp7 file are below a specified threshold, the data passes the test. If not, both data sets are plotted for visual comparison with the aid of the graphing capabilities of `modtrantest.py`. Figure 20 displays a set of comparison plots generated by `modtrantest.py`. In addition, the *.tp6 files generated by MODTRAN6 are compared for the given test case.

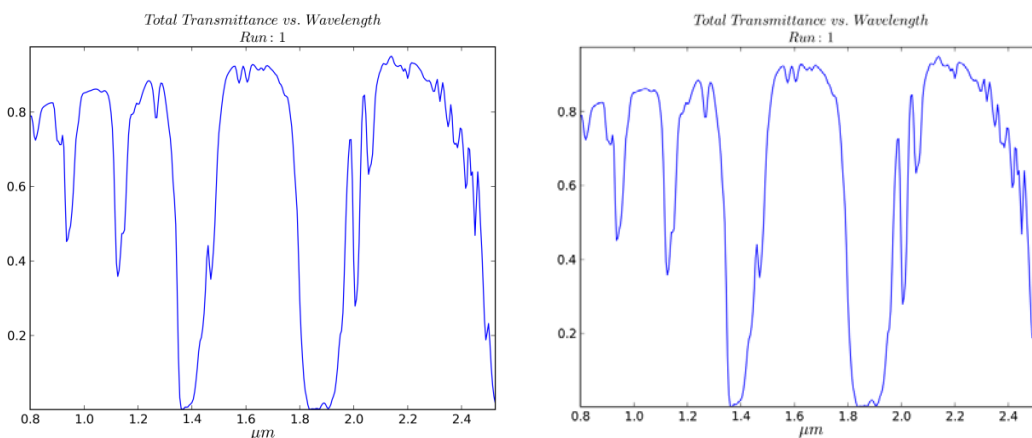


Figure 20. A Set of `buildbot` Comparisons for the CirrusProfile Test Case

In either case, a series of histograms is generated for both data sets with the aid of the test script. These histograms are separated based upon the property, line-of-sight, and run. In addition, two summary histograms are generated for each output file, the first illustrating the overall differences between the transmittances, if any, and the second illustrating the overall differences between the radiances, if any. In order to facilitate the detection of non-zero differences all histograms are truncated to the maximum value for the count of a non-zero difference. Figure 21 displays a `modtrantest.py` generated set of overall difference histograms.

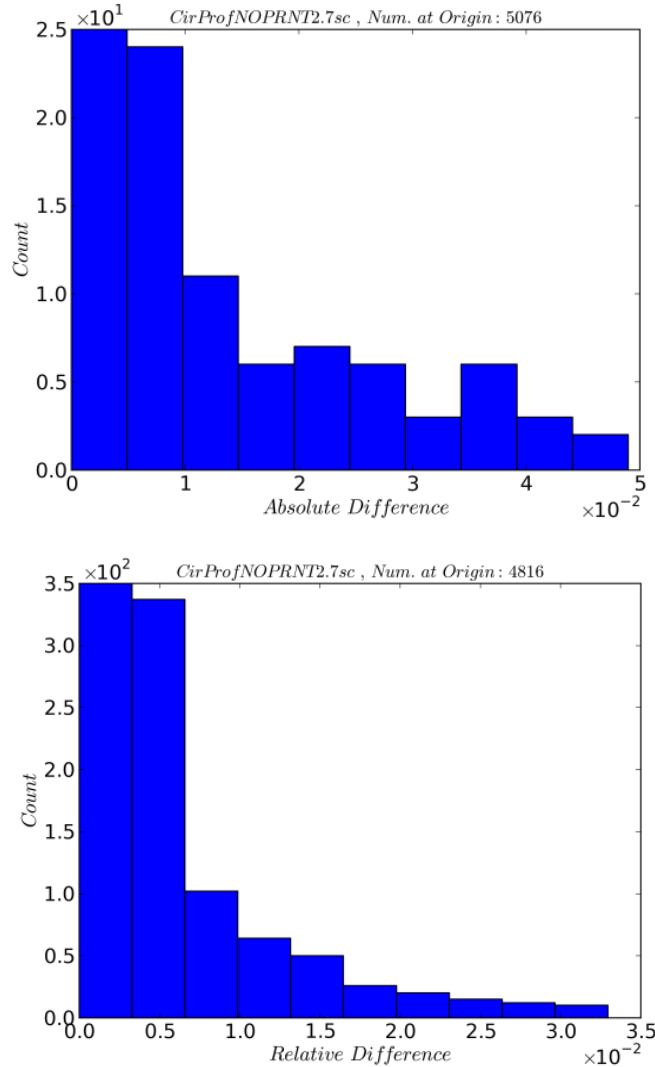


Figure 21. CirrusProfile Test Case Absolute (top) and Relative (bottom) Difference Histograms

The continuous integration `buildbot` software is configured to interface with the `modtrantest.py` test script, and once started generates an interactive website on a local server. A series of separate “builders” are setup depending on the specific build to perform for MODTRAN6. Included within these builders are those supporting the ifort, gfortran, g95, and pgf90 compilers. The `buildbot` software is connected to the SVN (Apache Subversion, <http://subversion.apache.org>) repository where it checks out the code and performs a series of builds on a nightly basis. Once a build is started, the MODTRAN6 source code is checked out from the repository, an executable is built with the appropriate compiler, and a series of over ninety test cases are run. Figure 22 displays a webpage created by `buildbot` which indicates the ongoing status of a build. After each test case is complete, the MODTRAN6 generated output files are compared with the appropriate set of fixed files by `modtrantest.py`. Figure 23 displays an example of an html file created by `modtrantest.py`. An additional html file is generated that allows for inspection of the *.tp5, *.tp6, or *.wrn files associated with a given test case.

The `buildbot` software can be started such that all tests are run in parallel and the final results of every test are displayed in tabular format on the website. At the end of the series of tests, an email is sent to the appropriate tester indicating the final results.

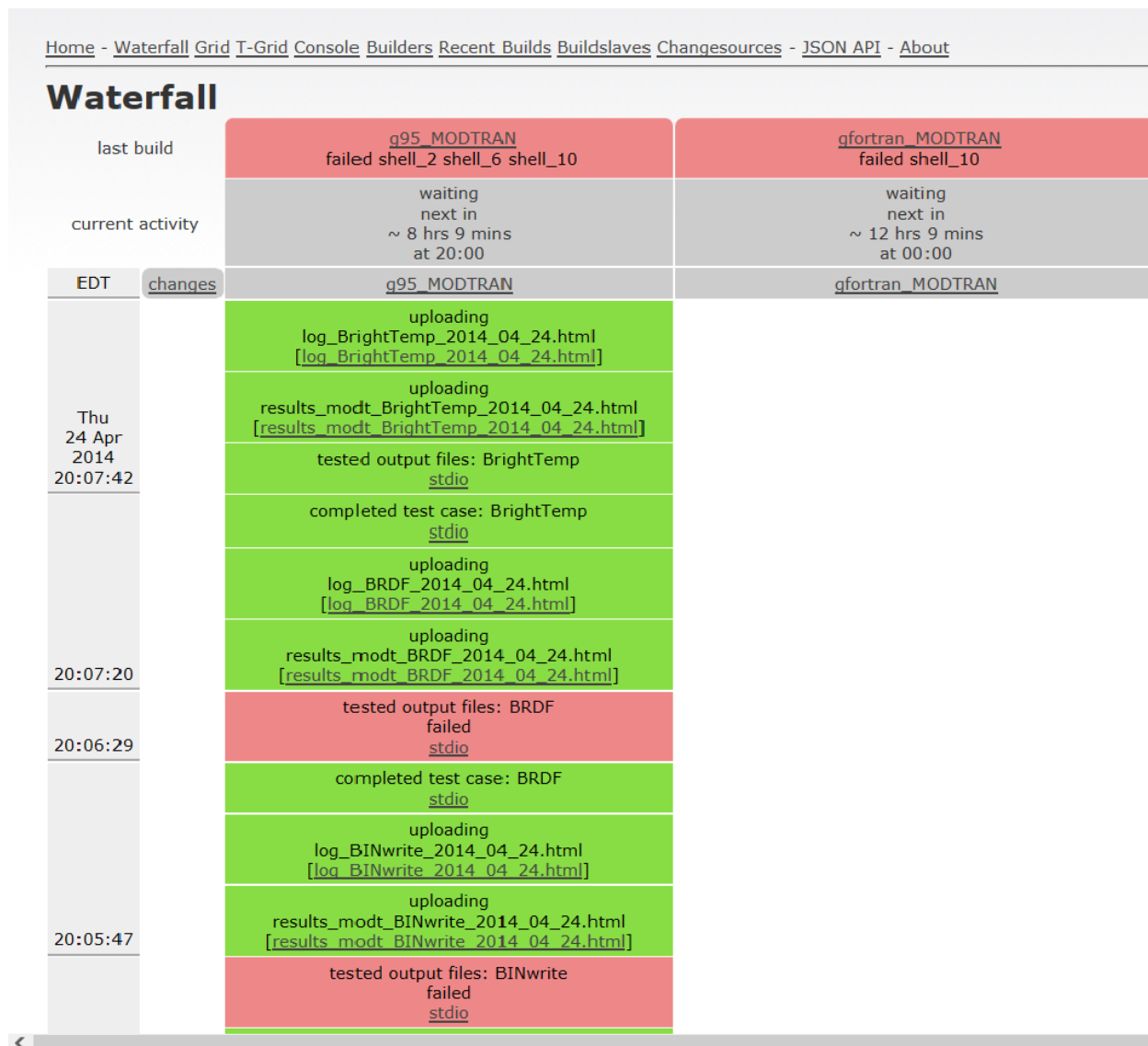


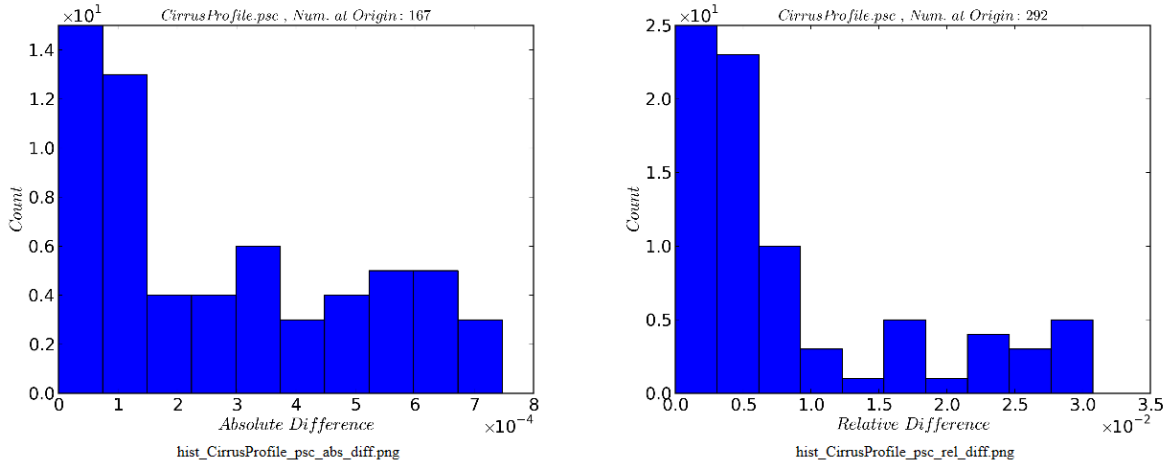
Figure 22. A buildbot Status Webpage for MODTRAN6 (green passed, red failed)

Results:

For: CirrusProfile CirrusProfile.tp7 Skipped.
No ASCII data in: CirrusProfile.tp7 to perform comparisons.

For: CirrusProfile CirrusProfile.psc Pass.

CirrusProfile.psc histograms: ([Detailed histograms for: CirrusProfile.psc](#))



For: CirrusProfile CirrusProfile.7sc Fail.

For run: 1 Thrml Em, exceeds specified tolerance, 0.0001	Max. Rel. Error: 0.0161645848641	Ind. Errors: 0 Dep. Errors: 335
For run: 1 Thrml Sct, exceeds specified tolerance, 0.0001	Max. Rel. Error: 0.00808585273046	Ind. Errors: 0 Dep. Errors: 153
For run: 1 Surf Emis, exceeds specified tolerance, 0.0001	Max. Rel. Error: 0.0125813997598	Ind. Errors: 0 Dep. Errors: 180

Figure 23. Final Results of a bui ldbot Comparison of CirrusProfile Output Files

4.4 C/C++ Application Programming Interface

The application programming interface (API) defines the way external applications and toolboxes communicate with MODTRAN6 and provides a bridge between the Fortran code and other languages. The API operates by allocating “CARD-like” structures for the various input and output parameters, and provides methods to execute the routines within MODTRAN. In the API design process, existing parameters in the tape5 input are separated and grouped into specific data types based on the information each represents. The grouping does not necessarily follow the layout in the CARD system, because many of the CARD system parameters were created by convenience as the software capabilities evolved over time rather than by design. Moving and grouping similar parameters together presents an logical interface to the end-user, and simplifies the process of adding new parameters to the system in future updates.

The basic interfaces of the API include data structures defining:

- RT Settings: MODTRAN runtime options, RT solver selection.
- Geometry: the LOS and time/location.
- Atmosphere: model data, constituent scaling, altitude/pressure profiles, aerosols.
- Ground: Surface reflectivity, temperature, elevation.
- Output: Spectral data arrays, options to select output.
- Logs: Reports generated during runtime.

The API provides three distinct types of interfaces:

1. An external interface, which defines methods and data types for communicating with external applications.

2. An internal interface, which interacts with the core library of MODTRAN routines.
3. A general plug-in interface, which allows the integration of any number of auxiliary codes into the framework.

Based on this API design, a common set of structured data types can be defined to handle communication between the internal, external, and plug-in interface functions. These data types include structures corresponding to the input found in the existing tape5 file, the output traditionally written to various MODTRAN output files, and the new input/output data used by the toolbox codes.

A language bridge between the C-API input structures and the core MODTRAN code permits the API to pass input settings from the user into variables used internally by the legacy Fortran routines. Coding for this work relies on the ISO_C_BINDING language extension, introduced into the Fortran2003 standard to provide portable interoperability between the C and Fortran data types.

Figure 24 illustrates an example of data validation of the scanned output data reported from the API. In this test case the *Total Radiance* output has been resampled from a 1 cm^{-1} band model calculation using a triangular scanning function with a FWHM of 2 cm^{-1} . The outputs reported from the legacy 7SC file and API structures for this case are numerically identical, within the limit of single-precision accuracy, indicating that the interface functions of the API are faithfully returning the generated output.

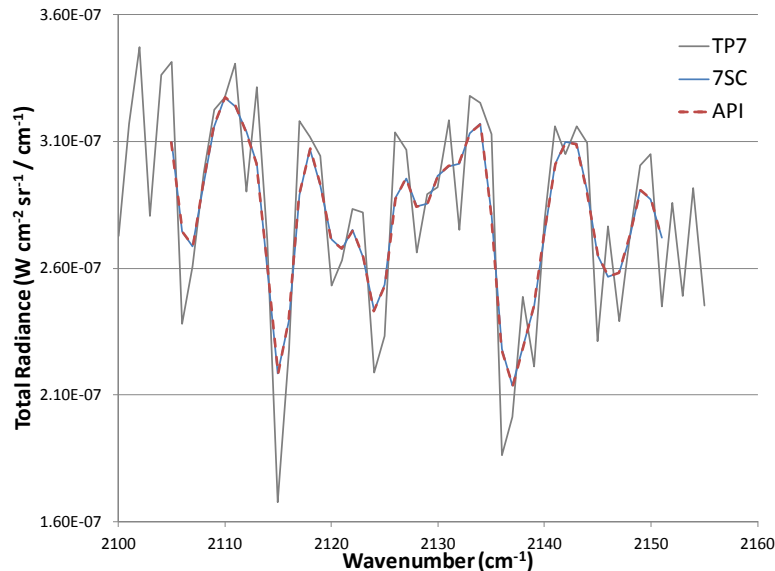


Figure 24. Validation of C-API Scanned Radiance Output Data

Besides the direct access to output data through interface functions, the API includes options to write the output into a number of flexible file formats. One available option is the ENVI[®] spectral library format (<http://www.exelisvis.com/ProductsServices/ENVIProducts/ENVI.aspx>). This format offers a convenient and efficient method to store the various combinations of spectra generated by MODTRAN in a compact binary file, along with a human-readable text header that describes the spectra contained in the library. These spectra can include the radiance, irradiance, and transmission output values, or similar types of array-based data. Each spectrum is represented by a numerical vector, stored sequentially in increasing wavelength or frequency. The contents are defined on a common wavelength or frequency grid, sharing a common numerical precision, such as byte, integer, longword integer or float. While the shared frequency grid somewhat limits the format's utility by accommodating only a single MODTRAN calculation per file, the file format follows a simple and compact structure for data storage, making it an efficient file-based option for interfacing with ENVI or other software systems. Figure 25 illustrates use of the ENVI Spectral Library Viewer to open and visualize a spectral library output file generated by MODTRAN.



Figure 25. Visualizing a MODTRAN Spectral Library Output File Using ENVI

Another output file option is a basic tabular text output, similar to the legacy *TP7* files but formatted to provide improved interface compatibility with software systems. These files record the collection of spectral values as a series of columns, with each value separated by either a tab or comma. Figure 26 contains an excerpt of an output file using the tab-delimited format. A brief header identifies the contents and dimensions of the data table, followed by the columnar list of spectral values. While somewhat less efficient than the other file output options (due to the interleaving of spectral values), this type of comma-separated-value (CSV) or tab-delimited file format is readily readable by most existing spreadsheet or graphing software packages, such as Microsoft Excel or Origin.

case index	0								
LOS index	0								
num freq	223								
num col	16								
Freq	Tot Trans	Thrml Emis	Thrml Scat	Surf Emis	Mult Scat	Sing Scat	Grnd Rflt	Drect Rflt	
13995	0.3493146	1.93E-32	4.48E-31	3.10E-31	2.27E-07	2.97E-08	1.03E-07	2.02E-08	
14085	0.3548864	9.58E-33	2.99E-31	2.06E-31	2.42E-07	3.04E-08	1.08E-07	2.05E-08	
14175	0.3484529	6.58E-33	1.94E-31	1.32E-31	2.37E-07	3.03E-08	1.04E-07	1.95E-08	
14265	0.3410817	4.63E-33	1.26E-31	8.38E-32	2.26E-07	2.97E-08	9.70E-08	1.81E-08	
14355	0.336505	3.01E-33	8.17E-32	5.38E-32	2.26E-07	3.01E-08	9.57E-08	1.77E-08	
14445	0.3209838	2.24E-33	5.14E-32	3.35E-32	2.00E-07	2.89E-08	8.46E-08	1.62E-08	
14535	0.2998491	1.68E-33	3.10E-32	2.02E-32	1.73E-07	2.71E-08	7.30E-08	1.45E-08	
14625	0.3369263	6.53E-34	2.28E-32	1.47E-32	2.46E-07	3.10E-08	9.97E-08	1.75E-08	
14715	0.3382998	3.84E-34	1.51E-32	9.64E-33	2.53E-07	3.16E-08	1.01E-07	1.74E-08	
14805	0.3356037	2.47E-34	9.85E-33	6.22E-33	2.51E-07	3.15E-08	9.88E-08	1.68E-08	

Figure 26. Example of a tabular Text Output Generated from the MODTRAN API

The API has been used to develop a C++ MODTRAN6 executable, which wraps the Fortran code, and provides more flexible input and output options. The C++ executable reads JavaScript Object Notation (JSON) input files, which directly describe settings for the data structures in the API using easily readable pairs of keywords and values (which may in turn be data structures). Running MODTRAN6 through the C++ executable with JSON input files allows the user to control or suppress the writing of the legacy Fortran output files, and to select the ENVI and/or CSV output formats described above. Details of the JSON file structure are described in Section 4.5.1 on the tape5 to JSON converter.

4.4.1 MATLAB® Interface

The MATLAB (<http://www.mathworks.com>) interface is designed using MATLAB classes which interact with a MATLAB mex compiled library. The MATLAB mex library allows for callable functions from the MATLAB command prompt using a dynamically loadable library (for MS Windows) or a shared object library (for Linux/Mac). The basic structure of the interface is shown in Figure 27. Within the MATLAB interface, the user defines the appropriate variables for the classes including RT options, the atmosphere description, the aerosol description, the geometry definition, the spectral range definitions, the surface properties, and the ‘file’ properties. Each of these classes corresponds to a structure in the MODTRAN6 API. There is one structure embedded in the MATLAB MODTRAN definition which maps the more descriptive MATLAB variables to the MODTRAN variables. The MATLAB mex library is compiled C++ code with the appropriate definitions from a MATLAB header file to properly map the input and output to MATLAB variables. The mex library interacts directly with the MODTRAN6 API, which in turn accesses the Fortran RT for the calculation. An example MATLAB script setting up and running MODTRAN is shown in Figure 28. This script shows the basics of instantiating, setting up the MODTRAN run properties/variables, and calling the MODTRAN run method. Some of the variables are defined via enumerated types for clarity and ease. The output structure (out in the figure) contains the standard tape7 variables for the particular chosen run (i.e. transmittance, solar, solar and thermal).

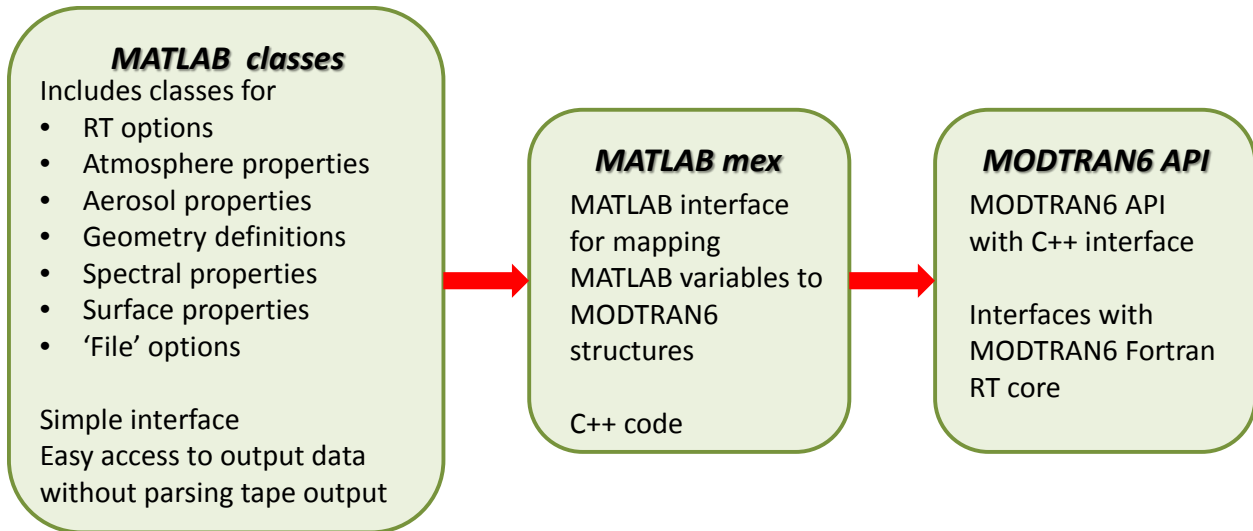


Figure 27. The Basic Structure of the MATLAB Interface

```

1 simplerun = modtran();
2
3 simplerun.rt_mode = rt_mode.RT_SOLAR_AND_THERMAL;
4 simplerun.multiple_scattering = multiple_scattering.RT_ISAACS_2STREAM_AT_OBS;
5
6 simplerun.atm_model = atm_model.ATM_US_STANDARD_1976;
7 simplerun.atm_mh2o = atm_model.ATM_US_STANDARD_1976;
8 simplerun.atm_mo3 = atm_model.ATM_US_STANDARD_1976;
9 simplerun.atm_mch4 = atm_model.ATM_US_STANDARD_1976;
10 simplerun.atm_mn2o = atm_model.ATM_US_STANDARD_1976;
11 simplerun.atm_mco = atm_model.ATM_US_STANDARD_1976;
12
13 simplerun.atm_co2_mixing_ratio = 380.;
14
15 simplerun.aer_aerosol_model = aer_ihaze.AER_RURAL_DENSE;
16
17 simplerun.geo_path = geometric_path.PATH_BETWEEN_TWO_ALTITUDES;
18 geo_los = line_of_sight(20., 0., 180., 0, 0, 0, false, 0, 0, 0);
19 simplerun.geo_los = geo_los;
20
21 simplerun.surf_surface_type = surface_type.REFL_CONSTANT;
22 simplerun.surf_surface_reflectance = 0.1;
23
24 simplerun.spec_initial_freq = 13000;
25 simplerun.spec_final_freq = 14000;
26 simplerun.spec_delta_freq = 1.;
27 simplerun.spec_fwhm = 2.;
28 simplerun.spec_units = spectral_units.WAVENUMBERS;
29 simplerun.spec_bandmodel_resolution = bandmodel.WAVENUMBER_0P1;
30
31 simplerun.file_run_name = 'mod6';
32 simplerun.file_data_directory = 'C:\\no_backup\\MODTRAN6.0.0.0-beta2\\MODTRAN6.0.0.0-beta2_win\\DATA\\';
33
34 out = simplerun.run();

```

Figure 28. A MODTRAN Run Case Defined in a MATLAB Script

4.4.2 IDL® Interface

An IDL (<http://www.exelisvis.com/ProductsService/IDL.aspx>) interface, similar in functionality to the MATLAB interface, was developed. It defines a MODTRAN run within the IDL environment and provides access to the output data without use of text-based I/O files. IDL objects define run variables (within RT option objects, the atmosphere description, the aerosol description, the geometry definition, the spectral range definitions, the surface properties, and the ‘file’ properties). A dynamically loadable library or shared object is invoked along with a dynamically loadable module (dln) with IDL. The output is then available via a property of the MODTRAN run object. In Figure 29, a diagram analogous to Figure 27 is shown. The only differences are the IDL objects are presented rather than the MATLAB class objects and the interaction from the MODTRAN API wrapper. This also shows how convenient the MODTRAN6 API is in allowing access to the MODTRAN input/output for high-level languages. A sample IDL script for running MODTRAN is shown in Figure 30. Many defaults are set in the MODTRAN IDL object definitions. The only variables which are changed from the defaults are re-defined in the script. Line 67 of Figure 30 shows the execution of the MODTRAN RT calculation and line 68 the legacy tape7 quantities are plotted.

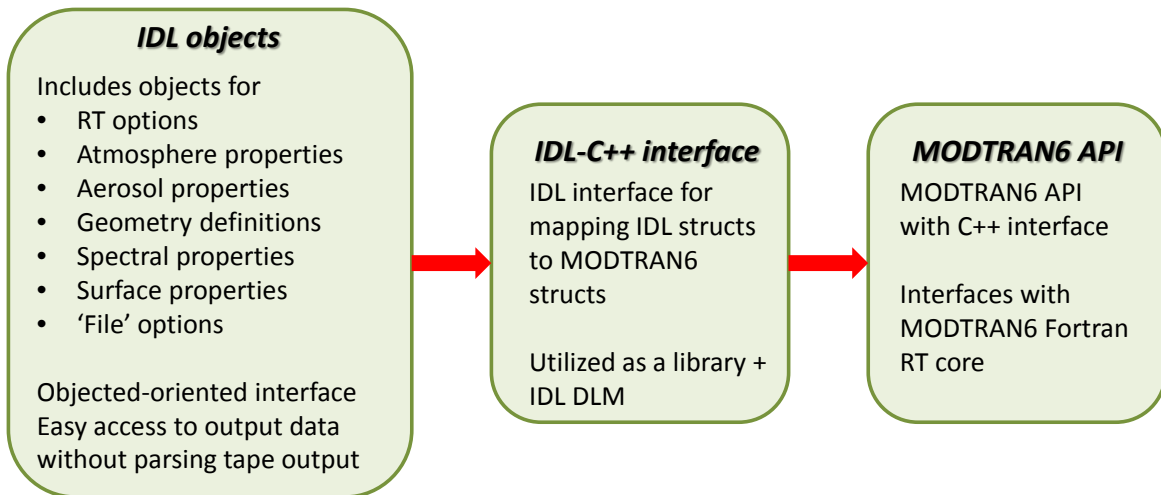


Figure 29. The Basic Structure of the IDL MODTRAN Interface

```

1  pro testModtranRun
2
3  ; rt options
4  rt = obj_new('rtoptions')
5  rt->setProperty, rt_run_mode='RT_SOLAR_AND_THERMAL'
6
7  ; atmosphere
8  atm = obj_new('atmosphere')
9  atm->setProperty, h2o_column=1.0
10 atm->setProperty, h2o_units='SCALED'
11 ; set a custom profile (altitude)
12 prof1 = obj_new('customprofile')
13 prof1->setProfile, findgen(100)
14 prof1->setProperty, profile_type='PROF_ALTITUDE'
15 prof1->setProperty, profile_units='UNT_KILOMETERS'
16 atm->addcustomprof, prof1
17 ; set a custom profile (delta temperature)
18 prof2 = obj_new('customprofile')
19 arr=fltarr(100)
20 prof2->setProfile, arr
21 prof2->setProperty, profile_type='PROF_TEMPERATURE'
22 prof2->setProperty, profile_units='UNT_TDELTA_KELVIN'
23 atm->addcustomprof, prof2
24
25 ; aerosols
26 aer = obj_new('aerosols')
27
28 ; geometry
29 geo = obj_new('geometry')
30 geo->setProperty, geometric_path='PATH_BETWEEN_TWO_ALTITUDES'
31 los = obj_new('lineofsight')
32 los->setProperty, initial_altitude=10.
33 los->setProperty, final_altitude=0.
34 los->setProperty, zenith_angle=180.
35 geo->add_los, los=los
36 geo->setProperty, solar_lunar=2
37 geo->setProperty, day_of_year=1
38 geo->setProperty, parm2=60.
39
40 ; surface
41 surf = obj_new('msurface')
42 surf->setProperty, surface_type='REFL_CONSTANT'
43 surf->setProperty, reflectance=0.05
44
45 ; spectral
46 spec = obj_new('spectral')
47 spec->setProperty, initial_freq=700.
48 spec->setProperty, final_freq=1400.
49 spec->setProperty, delta_freq=1.0
50 spec->setProperty, fwhm=2.0
51 spec->setProperty, bandmodel_resolution='p1'
52 spec->setProperty, bandmodel_year='2009'
53
54 ; set file options
55 fopts = obj_new('fileoptions')
56 fopts->setProperty, data_directory='C:\no_backup\MODTRAN6.0.0.0-beta1\MODTRAN6.0.0.0-beta1_win64\DATA\'
57
58 ; set the modtran run properties & execute
59 modRun = obj_new('modtranrun')
60 modRun->setProperty, rtoptions=rt
61 modRun->setProperty, atmosphere=atm
62 modRun->setProperty, aerosols=aer
63 modRun->setProperty, geometry=geo
64 modRun->setProperty, surface=surf
65 modRun->setProperty, spectral=spec
66 modRun->setProperty, fileoptions=fopts
67 modRun->exe, rvalue, tp7
68 ; plot the transmittance
69 plot, tp7[*,0], tp7[*,1]
70
71 ; cleanup
72 obj_destroy, modRun
73 obj_destroy, rt
74 obj_destroy, prof1
75 obj_destroy, prof2
76 obj_destroy, atm
77 obj_destroy, aer
78 obj_destroy, los
79 obj_destroy, geo
80 obj_destroy, surf
81 obj_destroy, spec
82 obj_destroy, fopts
83
84 end

```

Figure 30. Example IDL Script to Define, Run, and Plot a MODTRAN Run

4.4.3 Python™ Interface

Harris Corporation, formerly Space Computer Corp., created a Python interface for MODTRAN6 input data. This interface was used to integrate MODTRAN6 into their GeoReplay® target filtering software product, illustrated in Figure 31, a “Hyperspectral visualization tool with faster-than-realtime geo-registration of HSI data stream with automatic anomaly and signature-match target detection.” [\[http://www.mdcap.org/UserFiles/File/ESdocs/Presentations/ARCHER_Tools_Briefing_20071018.pdf\]](http://www.mdcap.org/UserFiles/File/ESdocs/Presentations/ARCHER_Tools_Briefing_20071018.pdf) GeoReplay® is used extensively for HSI data collection, real-time processing, target filtering and material identification, and post-processing data exploitation for several sensor platforms, including ACES-HY and MaRS+. Post-integration testing consisted of running the same cases in the pre-integration MODTRAN6 module and MODTRAN6 in GeoReplay®, Figure 32, and comparing the results. All computed values were identical within the numerical precision for the two modules (GeoReplay® uses slightly less precision than does the stand-alone module).

Timing improvements were slight relative to overall processing time for the new API; however, the incremental improvement will be beneficial when post-processing large numbers of ACES-HY scans.

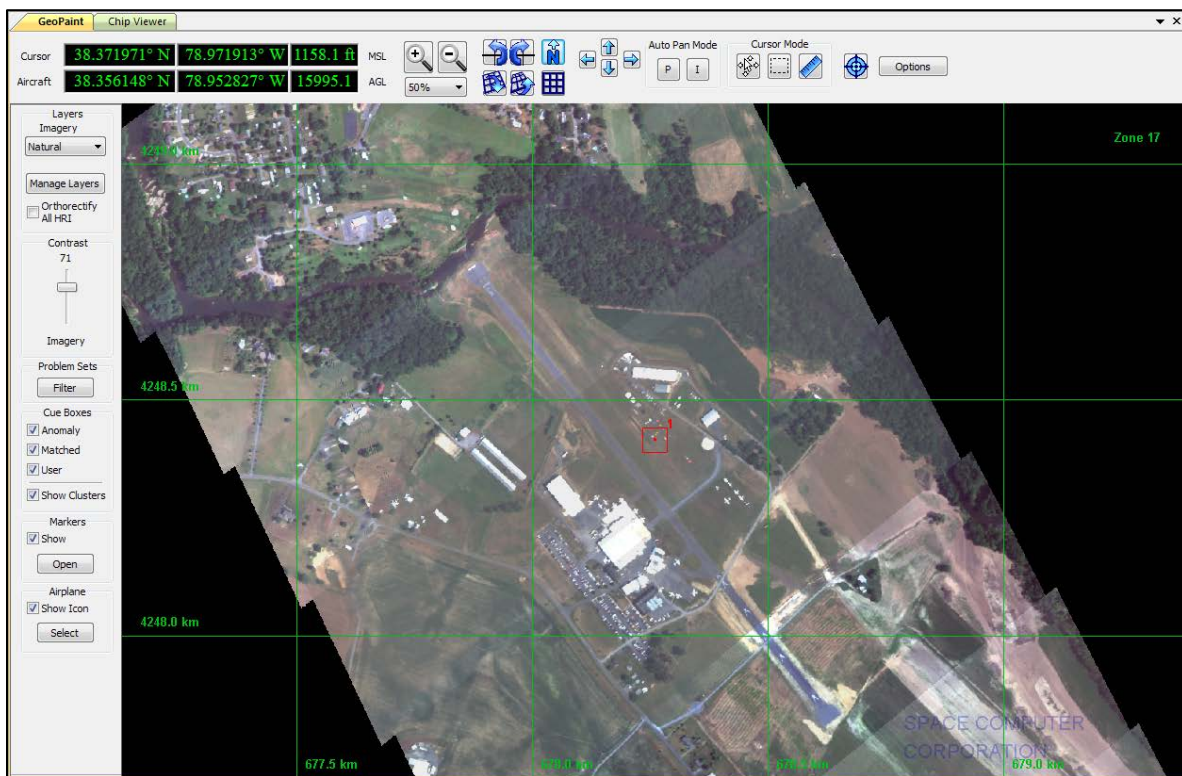


Figure 31. ACES-HY Data in GeoReplay®'s GeoPaint® Display

4.4.4 The MODTRAN6 GUI

A Java graphical user interface (GUI) has also been developed for MODTRAN6. Installation and user instructions are defined in the README_JavaGUI file, found in directory src/tools/JavaGUI/cpp. The GUI invokes the MODTRAN C API functions via the Java Native Interface (JNI). The JNI is a well-defined and Java supported interface for Java to invoke functions in a C/C++ library [either a Windows dynamically loadable library (.dll) or Linux/Mac shared object library (.so)]. The GUI produces a JSON string with all necessary input parameters defining a MODTRAN case, and this string is passed via the JNI to the MODTRAN API. The API then runs the MODTRAN Fortran RT engine. The resulting output is passed from the Fortran code through the API to the GUI, again via the JNI, for display.

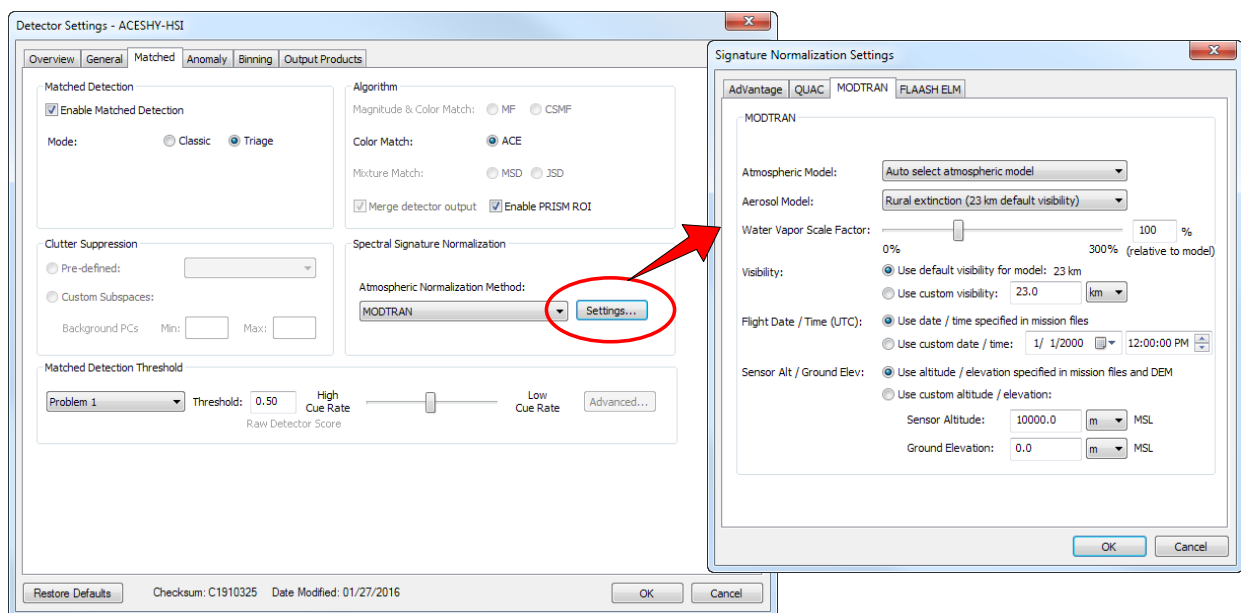


Figure 32. MODTRAN6 Settings for ACES-HY Data

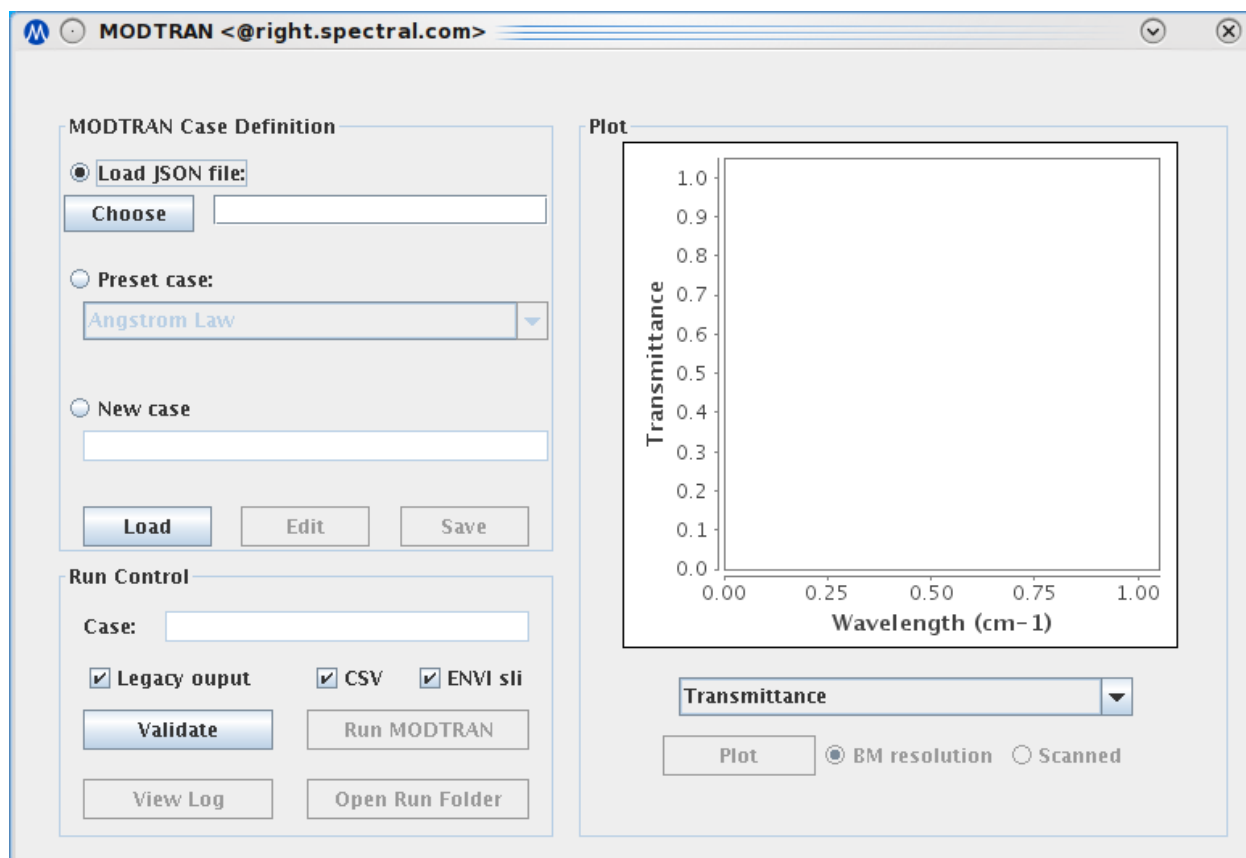


Figure 33. GUI Main Window

A screenshot of the GUI main window is shown in Figure 33. In the MODTRAN Case Definition box, the user can load one of several preset test cases, load a case defined by a JSON file, or input a new test case from scratch for which as a JSON file will be generated. Test cases can be edited after loading, and the modified JSON files saved.

As shown in the MODTRAN options (test case editing) window, Figure 34, there are six main tabs that are used to categorize all the MODTRAN input options. These define the RT Options (thermal only or thermal plus solar; with/without multiple scattering; band model, Correlated- k or line-by-line; etc.), line-of-sight and solar/lunar Geometry, Atmosphere profiles for the molecular gases, profile and optical data for Cloud & Aerosols, characterization of ground Surfaces, and Spectral Options. While many of the main options available in the Tape5 CARDS are presented in the corresponding tabs, advanced input windows are opened when necessary. For instance, the Custom Aerosols button in Figure 34 opens the input window shown in Figure 35, for Tabular Optical Properties, Spectral Aerosol Profile (SAP) file input, or Angstrom Law scaling.

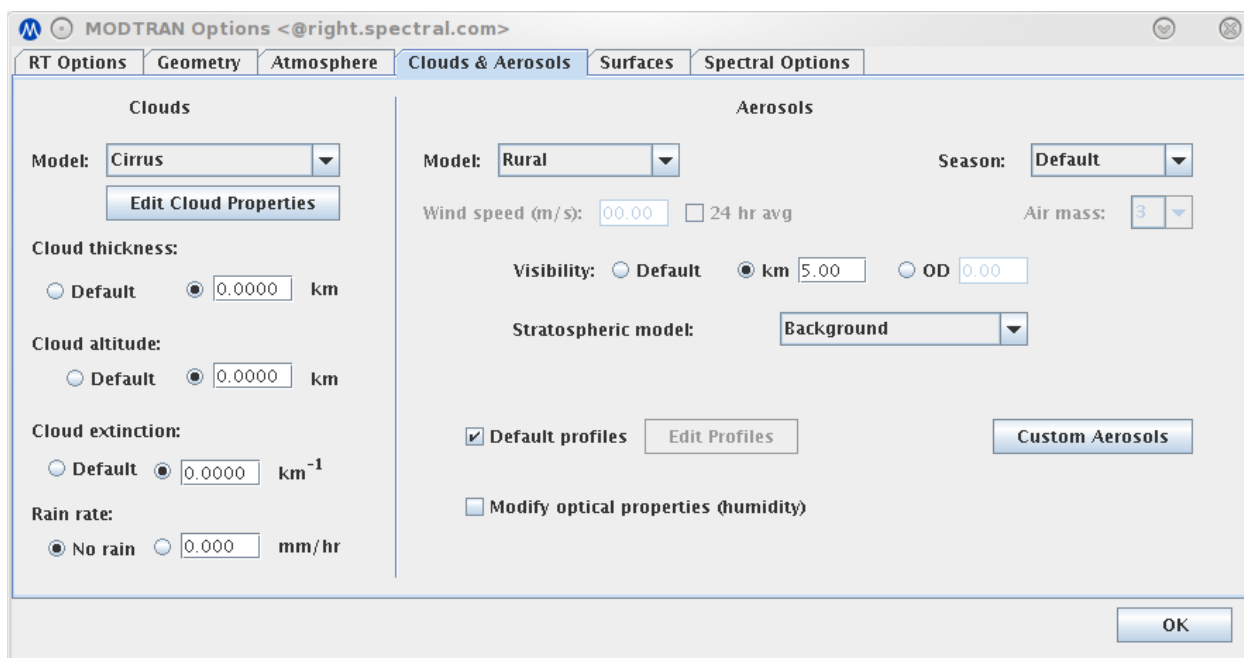


Figure 34. GUI Case Editing/Options Window Showing Tabs for Six Types of Options

Custom Aerosols <@right.spectral.com>

☐ **Tabular Optical Properties**

☐ Region 1 ☐ Region 2 ☐ Region 3 ☐ Region 4

Region 1	Region 2	Region 3	Region 4
Wavelength	Extinction	Absorption	Asymmetry
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

☒ **Spectral Aerosol Profile (SAP)**

☐ **Angstrom Law**

☐ **Include Tropospheric** **Exponent:** **Coefficient:** **Offset:**

Figure 35. Custom Aerosol Input Window

Upon completion of an RT calculation, the results are displayed in the Plot panel of the main window as shown in Figure 36; the generated I/O files are saved in a Run Folder in the user's directories. The plotting package includes capabilities such as a graphically controllable zoom function to display detailed spectral features within the full-spectral-range plot.

The GUI is built as a Java distribution with a Java Archive (JAR) file and a subdirectory containing ancillary packages; it is platform-independent and can be run on any system with Java installed. It can be invoked from the command line in Linux, or by clicking the GUI icon in Windows.

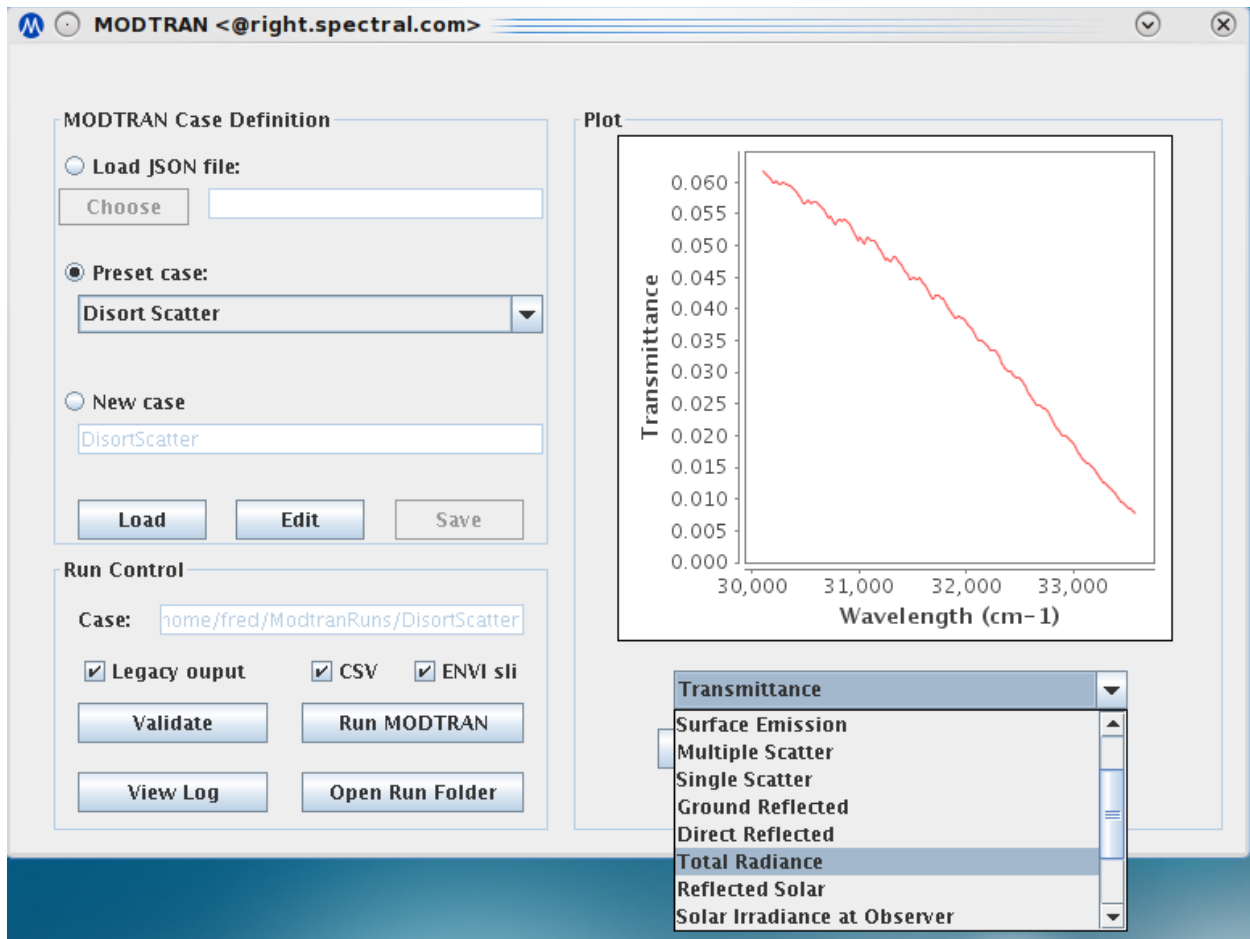


Figure 36. Plotted MODTRAN Results and Drop-Down Menu of Output Options

4.4.5 Control of Output File Creation

MODTRAN has historically provided all its output through files written to disk. API users do not need these files, because there are functions to return the results of a calculation in appropriately designed data structures. The tasks of managing files, formatting results and writing them are also time-consuming. To remedy this, functionality was added to control the creation and writing of the output files.

An integer variable, NOFILE, was added to the file manager module, along with tests on its values, to veto the creation and writing of different output files. If NOFILE is set to be less than or equal zero, all output files for a run are created and written as before. If NOFILE is set to one, only the tape6 and warning files are written. If NFILE is set equal or greater than two, only the warning file is written.

The input data structure has a NOFILE input variable, used to set the value for the file manager; the default value is zero to preserve the behavior of MODTRAN run as a stand-alone Fortran program. NOFILE is also mirrored in the C/C++ input structure, and the API function to process inputs sets the value in the file manager, so that the file creation behavior of MODTRAN can be controlled from the API (which is the only use case where the ability to turn off file creation is needed).

4.5 Auxiliary Codes

4.5.1 The *tp5tojson* tool

The *tp5tojson* tool converts the *tape5* format to JSON (Java Script Object Notation) text. This off-line tool, located in the MODTRAN6 bin/ directory, reuses Fortran MODTRAN code to perform the reading of the *tape5*. In particular, the READCD subroutine within the *inpDta* module is utilized. In using the READCD routine, the tool has access to the main input data type, *inpDta*, where all of the input variables are stored. A generic derived type is defined, shown in Figure 37, to output MODTRAN variables in a structured tree for JSON text. The derived type *node* has attributes *level* (the location in the tree), *attribute* (a character string of descriptive text), and arrays of either characters, integers, single precision, or double precision values. The *node* type is a linked list with pointers to the parent node, the children nodes (if they exist), and sibling nodes (if they exist). An overloaded method to add nodes to the tree exists wherein the new node name is defined as well as the parent node and the data to be added. The program then traverses the tree to find the parent node and adds the new data as one of its children. This layout allows an easily extensible method to build a tree structure. Once the tree is built from the MODTRAN input data, the structure has the form shown in Figure 38. All variables in the structure are of *node* type and include the compulsory information. This tree is easily (recursively) written to a text file with JSON compatibility.

type node
integer :: level
character :: attribute
character, dimension(:), pointer :: cval
character, dimension(:), pointer :: ival
character, dimension(:), pointer :: rval
character, dimension(:), pointer :: dval
type(node), pointer :: parent
type(node), pointer :: child
type(node), pointer :: sibling

Figure 37. The Derived Type *node* as Used in the *tp5tojson* Tool

The output of the *tp5tojson* tool is JSON formatted text that mirrors the Figure 38 tree structure. The JSON input parsing code has been modified to digest this input. The master file that defines keywords and their default values (DATA/keywords.json) has a nearly identical format. The input file can readily contain multiple MODTRAN runs using the features of the JSON format. The most notable change in the new input definition is that the legacy CARD labels has been discarded and replaced with groupings of input parameters into logical associations. For example, all spectral-related parameters are in the “Spectral” section of the JSON file.

The format of the JSON file is now as follows. The top level is a JSON array labeled “MODTRAN”. The elements of this array are JSON objects (key/value pairs) that each define a MODTRAN run. This array must always be defined, even in the case of a single run. Each object contains several sub-objects, each one corresponding to a labeled boxes in Figure 38 (except for “MODTRAN Run”). In turn, those objects contain key/value pairs for any non-default inputs the user specifies. For the most part, these are scalar integer, floating point, Boolean, or string values. There are a few one dimensional arrays and some are JSON objects that further define sub-arrays. The MODTRAN6 user manual has been updated to reflect these new value definitions. Figure 39 shows an abbreviated example of a JSON input file.

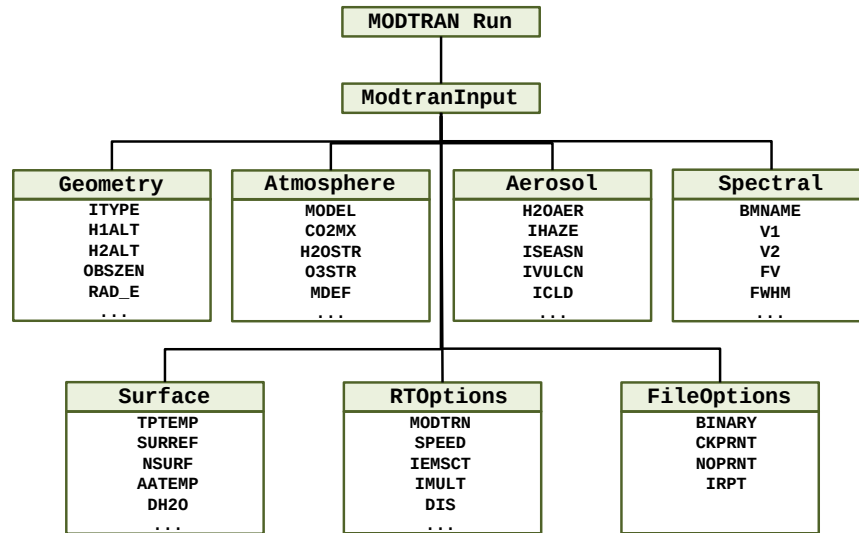


Figure 38. Tree Structure of MODTRAN Linked List Using Generic Node Data Type of Figure 37

```

{
  "MODTRAN" : [
    {
      "ModtranInput" : {
        "name" : "A+USS.tp5",
        "case" : 1,
        "RTOptions" : {
          "MODTRN" : "M",
          "IEMSCT" : 0,
          "IMULT" : 0,
          "DIS" : "F"
        },
        "Atmosphere" : {
          "MODEL" : 4,
          ... etc ...
        },
        "Aerosol" : {
          "APLUS" : "A+",
          "IHAZE" : 1,
          "ISEASN" : 1,
          "ARUSS" : "USS",
          ... etc ...
        },
        "Geometry" : {
          "ITYPE" : 3,
          "H1ALT" : 0.49000E+01,
          "H2ALT" : 0.00000E+00,
          "OBSZEN" : 0.18000E+03
          ... etc ...
        },
        "Surface" : {
          "TPTEMP" : 0.00000E+00,
          "SURREF" : ".0500",
          "GNDALT" : 0.00000E+00
          ... etc ..
        },
        "Spectral" : {
          "V1" : 0.25000E+04,
          "V2" : 0.26000E+04,
          "DV" : 0.25000E+02,
          ... etc ...
        },
        "FileOptions" : {
          "NOPRNT" : -1,
          "IRPT" : 0
        }
      }
    }
  ]
}

```

Figure 39. Sample New Input JSON Format

Approved for public release; distribution is unlimited.

4.5.2 Aerosol Toolkit

An Aerosol Toolkit (ATK) was developed as a pre-processing adjunct to the MODTRAN code. This suite of software tools generates aerosol optical property and profile data, in MODTRAN's SAP (Spectral Aerosol Profile) format, from user input aerosol specifications. The ATK allows users to create data files for "custom" aerosols with multiple components (specifying particle types, spectral refractive index data, particle size distributions, and number density profiles). Single particle optical properties for each aerosol component can be calculated directly using Mie code for spherical particles, or T-Matrix codes for more general axisymmetric particles. A diagram of the Toolkit architecture is shown in Figure 40.

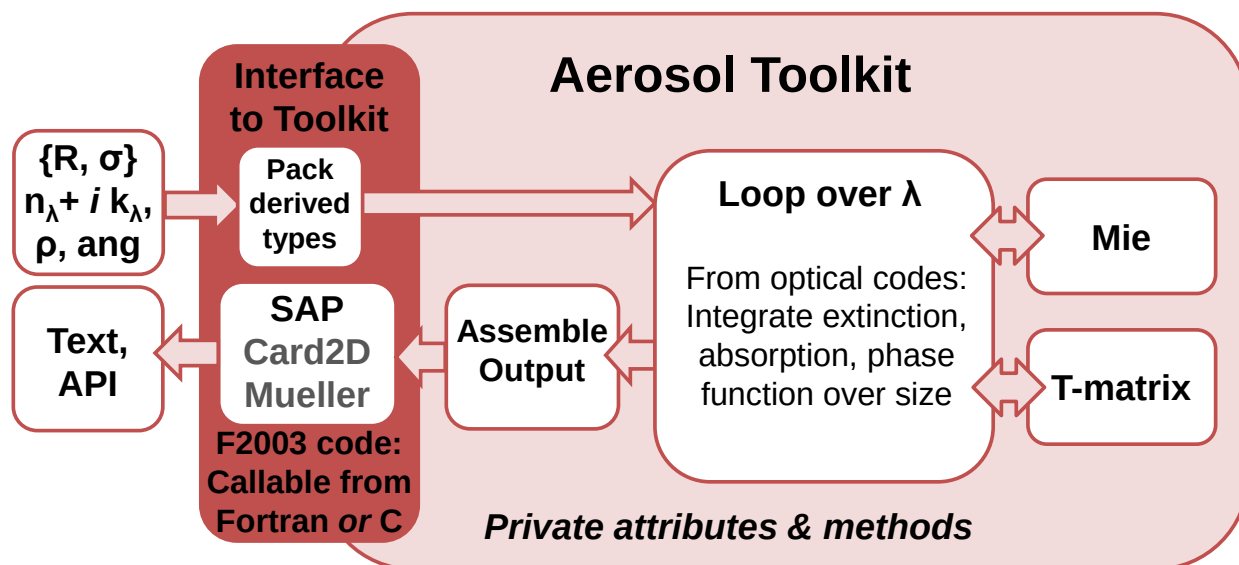


Figure 40. Diagram of the Architecture of the Aerosol Toolkit

The Toolkit uses Mie and T-Matrix software developed by Michael Mishchenko at the NASA Goddard Institute for Space Studies (<http://www.giss.nasa.gov/~crmim/>). The first-principles calculations of aerosol properties are described in text by Mishchenko *et al.* [12]. The Mie and T-matrix subroutines return both arrays of scattering matrix elements as a function of user-specified scattering angles, as well as angular function expansions of the matrix elements. The ATK also has the potential to output polarized scattering data.

Several internal data structures have been defined to describe individual aerosol components, to contain monochromatic particle properties from a Mie or T-matrix calculation for one component, and to contain a full SAP data set, possibly also including polarized scattering data. Routines perform Mie and T-Matrix calculations for user-defined aerosols over a user-input spectral range, and full Fortran and C/C++ APIs (Application Programming Interfaces) have been written. The APIs allow the user to define an aerosol as a set of components with specified size distributions, shapes, spectral refractive index data and number density profiles, perform either scalar or polarimetric calculations, and provide access to scalar or polarimetric optical properties. Although the Toolkit exists primarily as a library against which user programs can be linked, it also has a stand-alone program that provides a simple text-file user interface.

4.5.2.1 Example Aerosol Calculation

To test the aerosol toolkit, a sample calculation was performed for a multi-component aerosol approximating the 10 m/s wind speed desert model of Longtin *et al.*, [13]. The test case demonstrates the combining of optical properties from multiple aerosol components into a single SAP data structure, and its subsequent run within MODTRAN6. The Longtin aerosol model contain four aerosol components, each defined with a lognormal size distribution characterized by a mode radius r_g and a standard deviation, σ :

- Carbonaceous particles ($r_g = 0.0118 \mu\text{m}$, $\log \sigma = 0.3010$, 0.0005 to 100 μm radii);
- Water-soluble particles ($r_g = 0.0285 \mu\text{m}$, $\log \sigma = 0.3500$, 0.0005 to 100 μm radii); and
- O-ray sand (10/90% hematite/silicate mass loadings, $r_g = 7.760 \mu\text{m}$, $\log \sigma = 0.3310$, 0.05 to 300 μm radii)
- E-ray sand (distribution identical to o-ray sand).

Silica is optically birefringent. Light scattered from ordinary (o-ray) silica particles is polarized perpendicular to the incoming direction of propagation; light scattered from extraordinary (e-ray) silica particles is polarized along the incoming direction.

The 10 m/s size distributions for the components are shown in Figure 41.

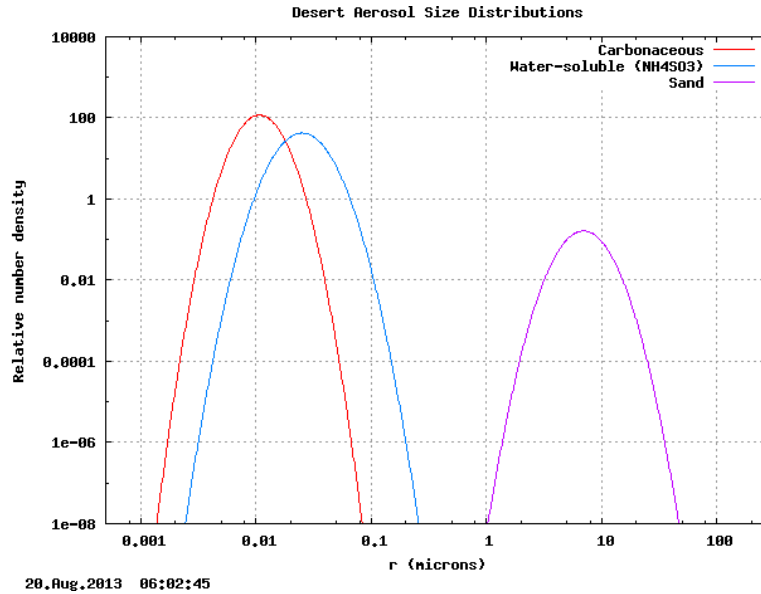


Figure 41. Lognormal Size Distributions for the Longtin Desert Aerosol Model

The mode radius, and the volume and mass loadings for the sand components increase with wind speed. Table 6 lists the volume concentrations v_i for 10 m/s wind speed, along with the average single-particle volume $\langle V_i \rangle$ for each component and the corresponding number density ($n_i = v_i / \langle V_i \rangle$) in particles/cm³.

Table 6. 10 m/s Wind Speed Desert Aerosol Particle Component Properties

Aerosol Component i	Concentration, v_i (μm^3 aerosol / cm^3)	Avg. Volume, $\langle V_i \rangle$ (μm^3)	Number Density, N_i (particles / cm^3)
Carbonaceous (soot)	0.022	1.035×10^{-5}	2.1263×10^3
Water-soluble (NH_4SO_3)	6.613	1.683×10^{-4}	3.9298×10^4
Sand (2/3 o-ray, 1/3 e-ray)	395.5	3.207×10^3	0.12331

A simulated 10 m vertical resolution number density profile was introduced for testing, Figure 42. The constant values of Longtin were used up to 100 m above ground. Above 100 meters, the number densities for carbonaceous and water-soluble particles remained constant up to 190 m. The sand number density decreased linearly with altitude from 100 to 200 m. At 200 m, all the densities were set to zero. Since the sand particles are birefringent, their o-ray and e-ray number densities are split statistically, 2/3 to 1/3, respectively.

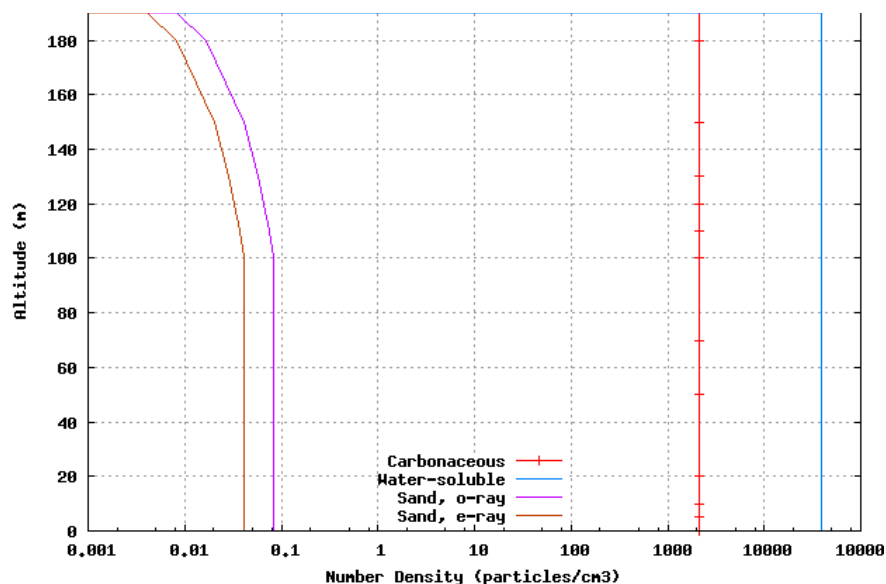


Figure 42. Number Densities Profiles for the Desert Aerosol Components

Complex refractive index data for the Longtin desert aerosol components are shown in Figure 43. The spectrally flat carbonaceous data is based on carbon black and soot measurements; dry NH_4SO_3 data was used for the water-soluble component; and sand components were modeled as quartz with 10% hematite by mass. The o-ray and e-ray refractive index data are similar below $\sim 12 \mu\text{m}$, but differ at longer wavelengths.

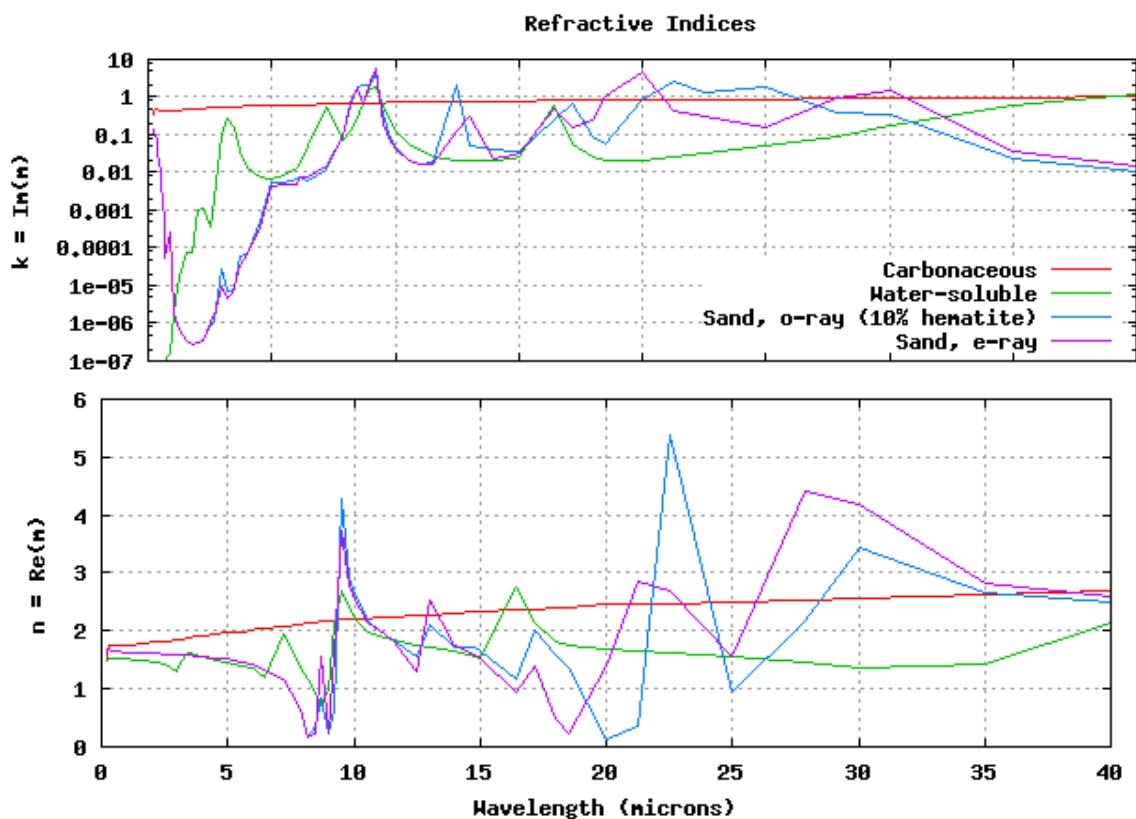


Figure 43. Complex Refractive Index Data for the Aerosol Components

For the MODTRAN simulation, an LWIR spectral range (8 to 15 μm) was partitioned into 129 points equally spaced in frequency. The Aerosol Toolkit Mie calculations for the 4 aerosol components required 2.31 CPU hours on one core of an Intel workstation (Core i7 at 3.4 GHz maximum).

The total aerosol extinction and scattering coefficients (km^{-1}) below 11.5 μm are plotted for each of the altitude grid points from 100 to 190 m in Figure 44 (remember, the aerosol properties are constant below 100 m). Figure 45 contains the 100 m altitude combined scattering phase functions for the composite aerosol at the spectral band end points, 8 and 15 μm . The scattering phase function variation with altitude is minimal, because the sand aerosol component completely dominates the scattering.

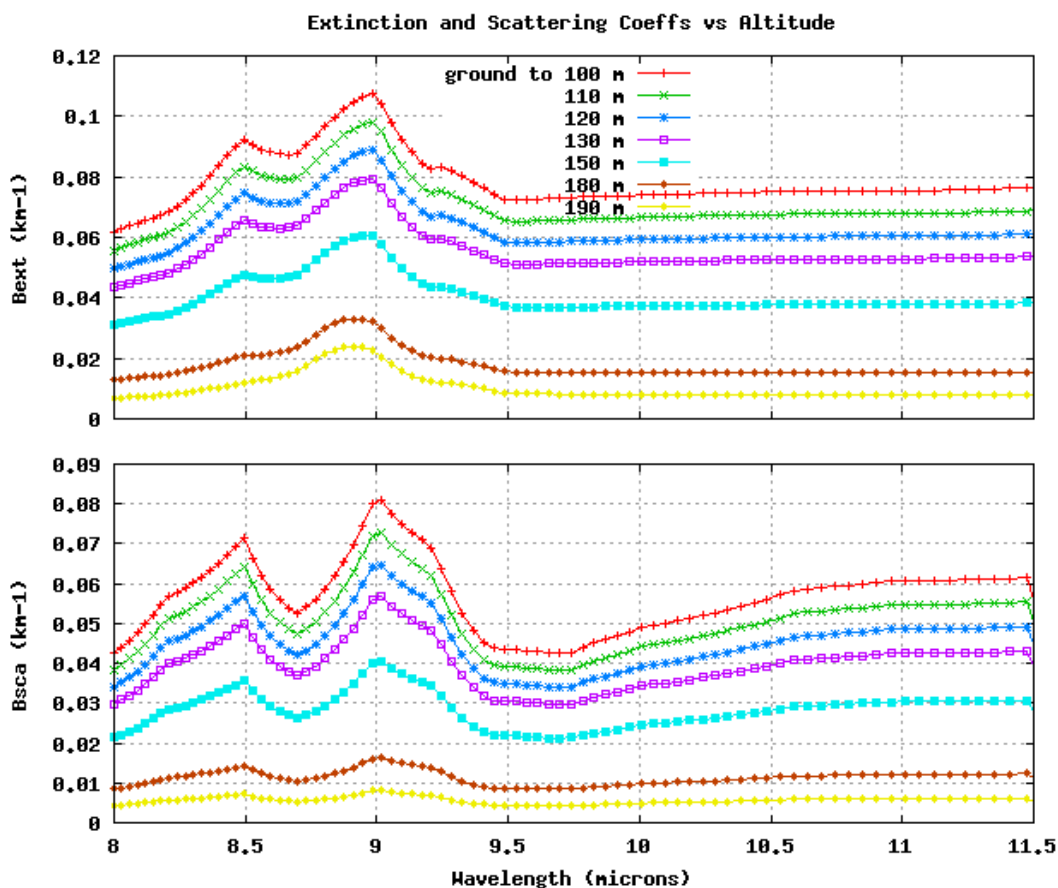


Figure 44. Desert Aerosol Extinction and Scattering Coefficients at Altitudes from 100 to 190 m

The availability of the MODTRAN aerosol toolkit facilitates analysis of the observed radiance response to changes in aerosol composition and properties. In Figure 46, MODTRAN6 simulations are compared. In the first calculation (black curves), MODTRAN6 is run with its built-in desert aerosol model and a wind speed set to 10 m/s. The sensor is 190 m above ground level (AGL) for this long (19 km), nearly horizontal path to ground. The elevation angle at the ground, back to the sensor, is 0.5 degree. This path length was chosen to produce path emission, path scatter and ground emission radiance components all of similar magnitude in the center of the 8 to 13 μm atmospheric window. The ground temperature is matched to the surface air temperature of 294.2 K, and Lambertian surface emissivity is 0.95. Path spectral transmittance is plotted on the left and observed radiance, on the right, both with a 20 nm FWHM resolution. In the second MODTRAN6 run (red curves), the Aerosol Toolkit generated SAP file, described above, defines the desert aerosol. The two desert models differ in that the sand component concentration in the SAP file drops off above 100 m as shown in Figure 42. The surface visibility in the

first run was set to 22 km so that the line-of-sight (LOS) aerosol optical depths from the 2 calculations matched at 10.4 μm . The mid-latitude summer (MLS) atmosphere model defined the molecular profiles; however, the CO_2 mixing ratio was increased to 398 ppmV to reflect current (2016) measurements, and the moist MLS vertical water column was reduced by 90% to be more indicative of desert conditions. The two calculations exhibit similar spectral structure, as expected, since their molecular profiles are identical. Between 10.0 and 11.8 μm , the spectral transmittances are in good agreement (remember, the visibility was chosen to provide a pivot point at 10.4 μm). Presumably, the fall off of sand particles above 100 m decreases aerosol extinction below 10 μm and increases it near 12.2 μm . The effect of these extinction differences is evident in the ground emission residuals (magenta curve, built-in aerosol model minus SAP input values) on the right. The fact that there are significant path emission residuals (blue curve) can only be explained by differences in the aerosol absorption profiles since the altitude grid and temperature profiles for the two calculations were identical. Differences in the aerosol scattering profiles produce path scatter residuals anti-correlated with the path emission values.

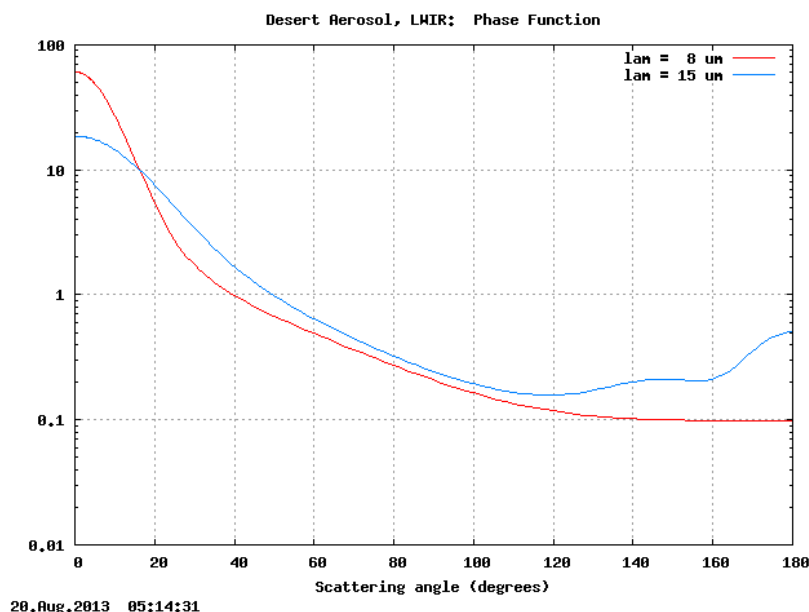


Figure 45. Combined Scattering Phase Functions for the Full Aerosol, at 8 and 15 μm Wavelengths

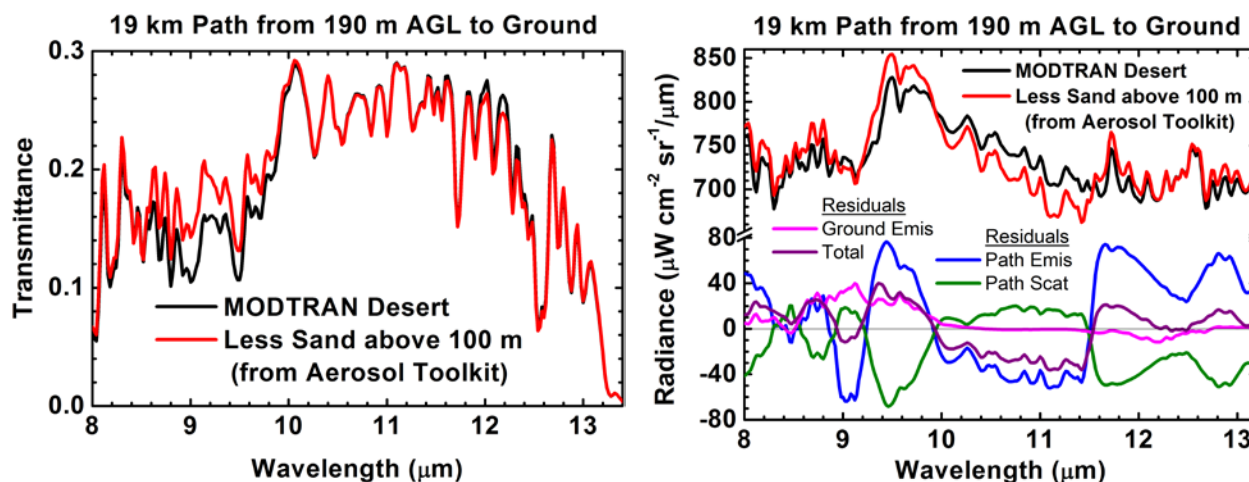


Figure 46. MODTRAN6 Simulations with Built-In and Aerosol Toolkit Desert Models

4.5.3 Atmosphere Generator Toolkit

For many applications, a user may need to incorporate atmospheric profile measurements in MODTRAN as a substitute for one of the six built-in model atmospheres. MODTRAN now includes a toolkit utility to simplify the process of incorporating measurements of pressure, temperature and H₂O concentration vertical profiles. The Atmosphere Generator Toolkit (AGT) provides MODTRAN users the ability to input custom atmospheres into MODTRAN in a straightforward manner. This general process is shown in Figure 47. The toolkit must be provided with either a radiosonde or location (latitude, longitude, date, and time) data. The toolkit outputs custom atmospheres using the NOAA/NCEP Reanalysis database (Reanalysis) interpolation algorithm if location information is provided or an internal radiosonde converter if radiosonde data is provided. Output is provided either in the form of a tape5 atmosphere structure or in a JSON formatted text file.

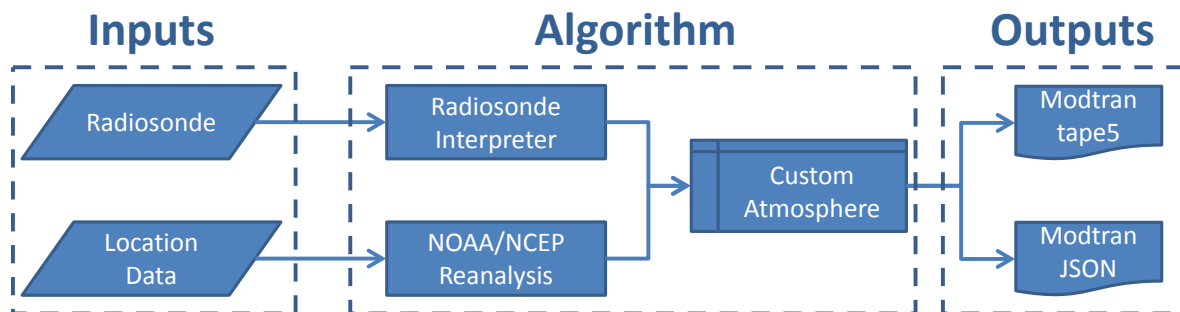


Figure 47. Diagram of the Atmosphere Generator Toolkit

Radiosondes and Reanalysis data alone are insufficient to describe a complete MODTRAN atmosphere. The AGT supplements the provided atmospheric information with data from standard MODTRAN atmospheres. Data from the six built-in models in MODTRAN are used above the highest available altitude provided by a radiosonde or Reanalysis database. The appropriate MODTRAN model is chosen by least-squares fits of temperature and H₂O profiles in the data to each of the MODTRAN models and choosing the models that have the least discrepancy in H₂O and temperature values.

4.5.3.1 Allowed Radiosonde Formats

The AGT can incorporate three different types of radiosondes into MODTRAN. These three types are:

- **ASCII radiosonde data:** This data consist of several thousand lines, e.g., 2500 or more, of altitude data of pressure, temperature and relative humidity, ranging from the ground level to about 30 km. For raw radiosonde data with thousands of altitudes, the data is smoothed and the number of altitudes reduced to less than 100 for input to MODTRAN. The column density of H₂O is preserved, and as are “bends” in the temperature profile, for example at the tropopause.
- **University of Wyoming formatted radiosonde:** This post-processed data is taken at a number of weather stations with a central repository (<http://weather.uwyo.edu/upperair/>) maintained by the University of Wyoming. This data typically consists of processed radiosonde measurements at 100s (not thousands) of altitudes. Typically this data extends up to 25–30 km.
- **NOAA/ERSL formatted radiosonde:** The Global Systems Division [formerly the Forecast Systems Laboratory (FSL)] of the Earth System Research Laboratory (ESRL) at the National Oceanic and Atmospheric Administration (NOAA) introduced the FSL radiosonde format. The NOAA/ESRL archived online Radiosonde Database (<http://www.esrl.noaa.gov/raobs/>) contains WMO station data. Please refer to http://esrl.noaa.gov/raobs/intl/fsl_format-new.cgi for a complete description of the FSL format.

These databases contain similar but not identical units. Each line contains a single set of altitude, pressure, temperature and humidity values. Occasionally there will be altitude lines with missing data. AGT fills in missing values using neighboring measurements.

4.5.3.2 The Reanalysis Database

NOAA maintains the NCEP/NCAR Reanalysis Monthly Means and Other Derived Variables global time-averaged database (<http://www.esrl.noaa.gov/psd/data/gridded/reanalysis/>). This database provides long term monthly means of air temperature, geopotential height, and relative humidity on a 2.5 degree latitude by 2.5 degree longitude global grid for 17 pressure levels using data from 1981 – 2010. An example of the global output is shown in Figure 48. The temperature output in Kelvin is shown for February for the first pressure level globally.

The AGT uses this database to provide reasonable atmospheric profiles given only location data. A logarithmic interpolation of the database pressure levels is performed to determine pressure as a function of height; typically, the profile maximum height is ~ 40 km. Above 40 km, the atmosphere reverts to the standard MODTRAN profiles. A similar method is used for temperature though linear interpolation is used. For humidity, the top data level at all latitudes is ~ 9 km. A logarithmic extrapolation of the mixing ratio, based on MODTRAN's model atmospheres, is performed above the tropopause, minimum of the temperature profile.

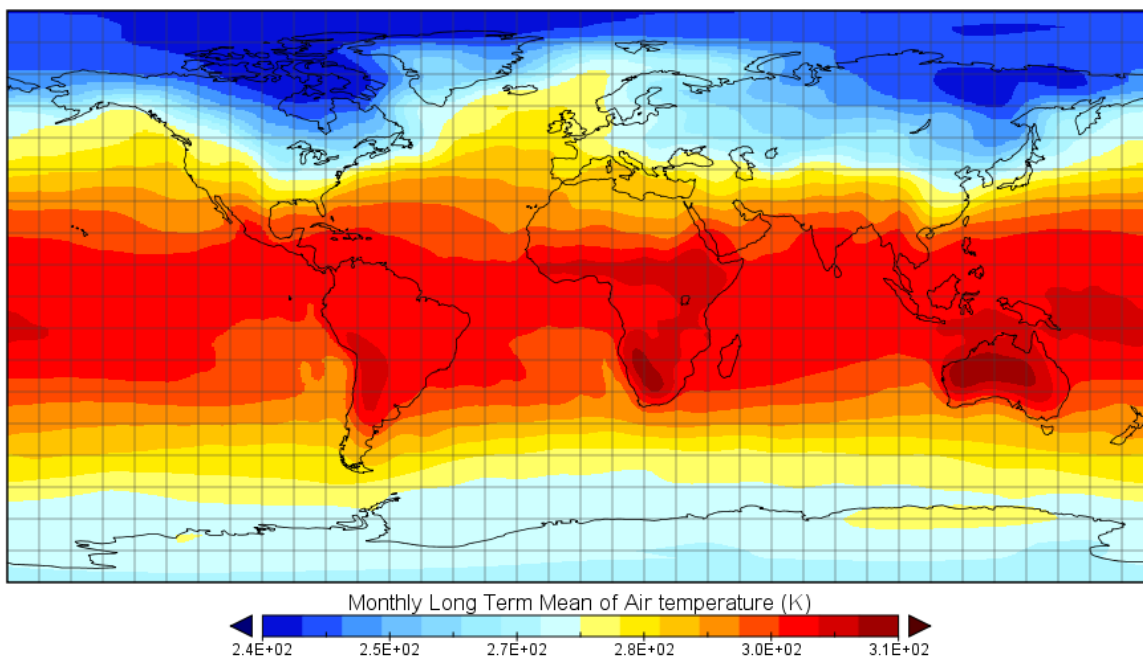


Figure 48. NCEP/NCAR Long Term Near Surface Global Mean Temperatures for February

4.5.3.3 Examples

Several MODTRAN input profiles were created from the profile measurements described above. These are described in the following sub-sections

4.5.3.3.1 ASCII Radiosonde Example

Figure 49 shows the pressure profile from raw radiosonde measurements at several thousands of altitude levels, and the AGT fit. In this example, the best fit MODTRAN model for extrapolation above 25 km is the US Standard model. The limited number of radiosonde levels between 25 and 27 km were determined to be too noisy and were discarded. The MODTRAN pressure profile is defined at 61 altitude levels from 0-70 km.

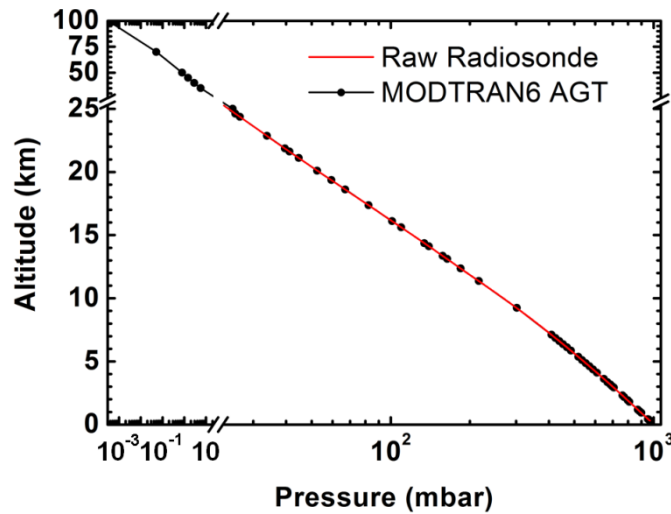


Figure 49. Pressure Profiles from Raw Radiosonde Data and the AGT 61 Level Fit

As shown in Figure 50, the AGT radiosonde utility is designed to preserve local minima and maxima for temperature and H₂O profiles along with the H₂O total column amounts. The figures contain the raw radiosonde data and the 61 level AGT fits. The discarded noisy data at the top of the measurement profiles is evident, as is the extrapolation to higher altitudes based on a best fit selection of one of MODTRAN's built-in model profiles.

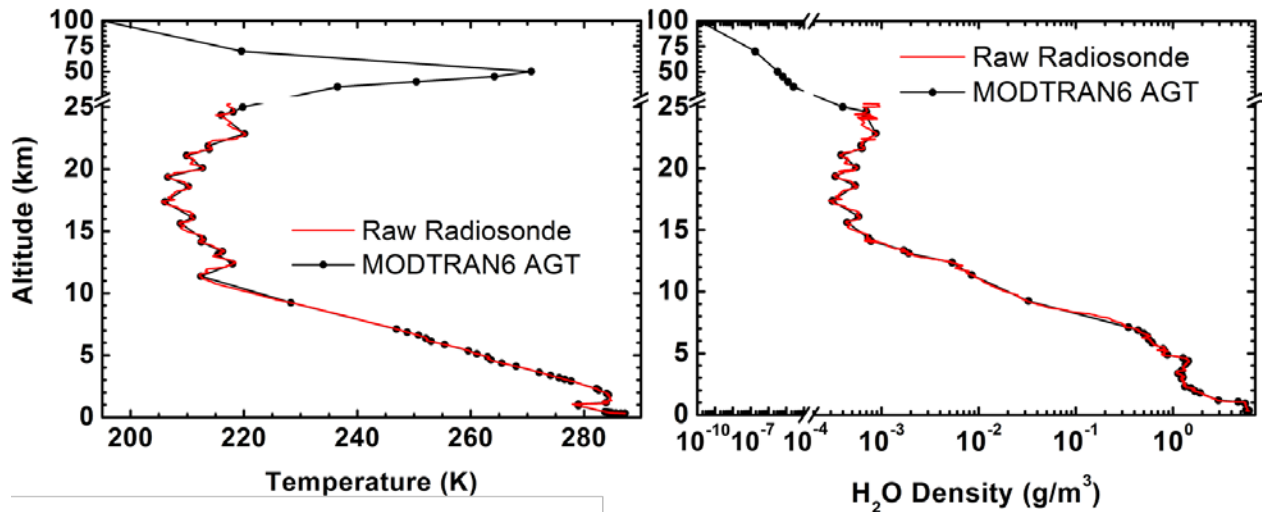


Figure 50. Temperature and H₂O Profiles from Raw Radiosonde Data and the AGT 61 Level Fit

4.5.3.3.2 NCEP/NCAR Reanalysis Database Example

Figure 51 compares 45°N central US (Gettysburg, SD) summer (August) mean monthly averaged profiles from the NCEP/NCAR Reanalysis database to the standard MODTRAN Mid-Latitude Summer (MLS) atmospheric profiles. The comparisons agree strikingly well, which serves as a true testament to the validity of the 50 year old models [NASA (1966), U.S. Standard Atmosphere Supplements, 1966, U.S. Government Printing Office, Washington, DC]. That being said, the difference are important. The monthly averages temperature near the ground at the specific latitude and longitude of Gettysburg, SD is more than 10 K warmer than the MLS values. Not surprising, the warmer boundary layer and tropospheric profiles contain more atmospheric H₂O.

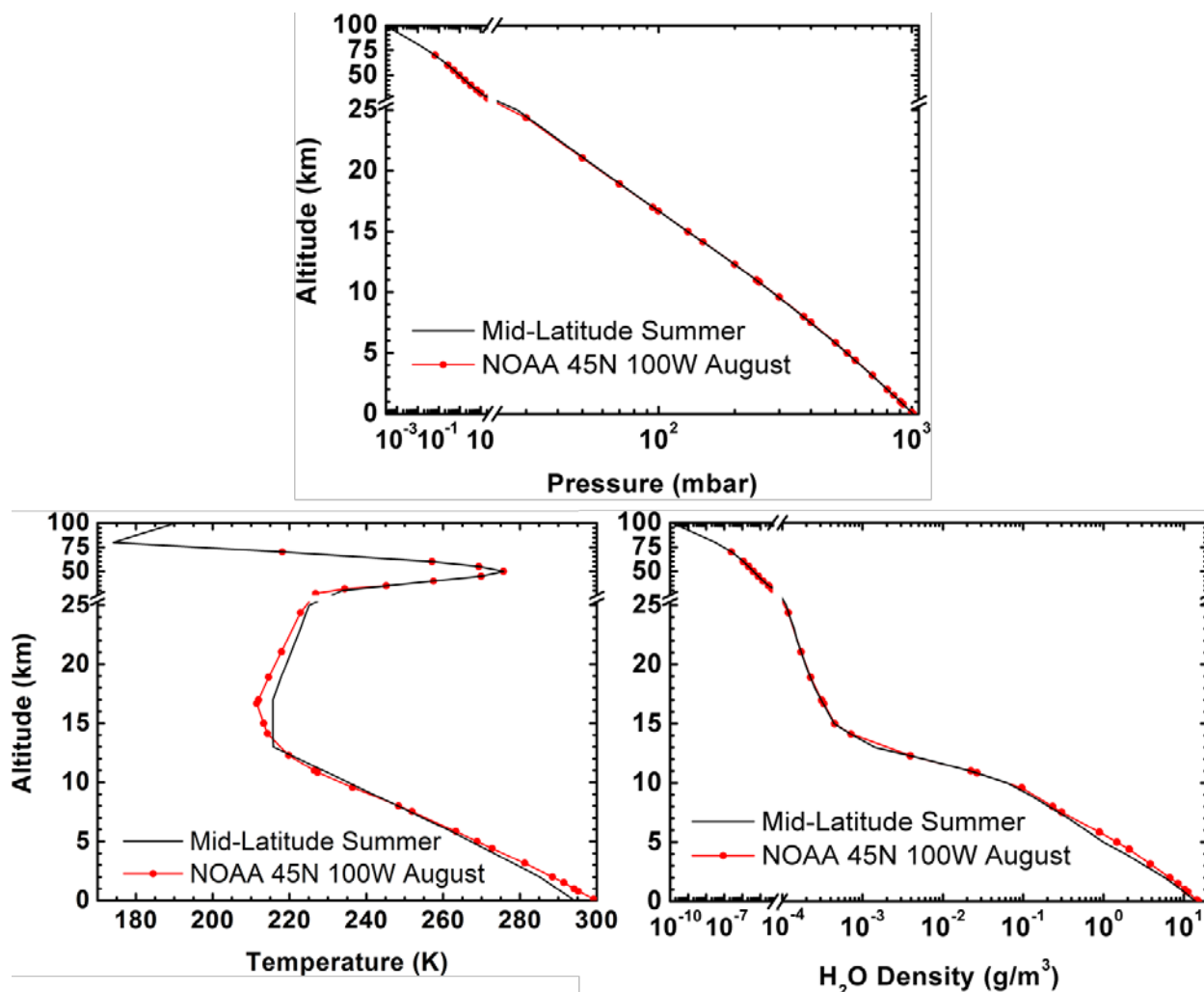


Figure 51. Comparison of NCEP/NCAR 45° August to MODTRAN Mid-Latitude Summer Profiles

4.5.3.3.3 Impact on Measured Radiance

Space – to – space MODTRAN limb radiance calculations with a tangent (minimum) height of 100 m, Figure 52, were run using both sets of profiles from Figure 51. The solar zenith was set to 45° and the relative azimuth at the tangent point was 90°, perpendicular to the line-of-sight vertical plane. The mid-wave infrared (MWIR) calculations spanned the 2.2 to 3.0 μm region, so solar scatter is the dominant source of radiation. If calculations were performed long of 4.0 μm , where thermal dominates, one would expect that the NOAA (NCEP/NCAR) profiles would produce the larger spectral radiance. In the solar regime, the atmospheric variable of greatest weight in these calculations is the H₂O density. The higher water amounts for the NOAA profiles increase attenuation of the solar illumination along the line-of-sight. As a result, the spectral radiances computed with the MLS profiles exceed the NOAA atmosphere values in Figure 52. For the current example, the Gettysburg, SD radiances at ~4% less than the nominal MLS values. This simulation illustrates the type of improvements one might expect from using the improved climatological data.

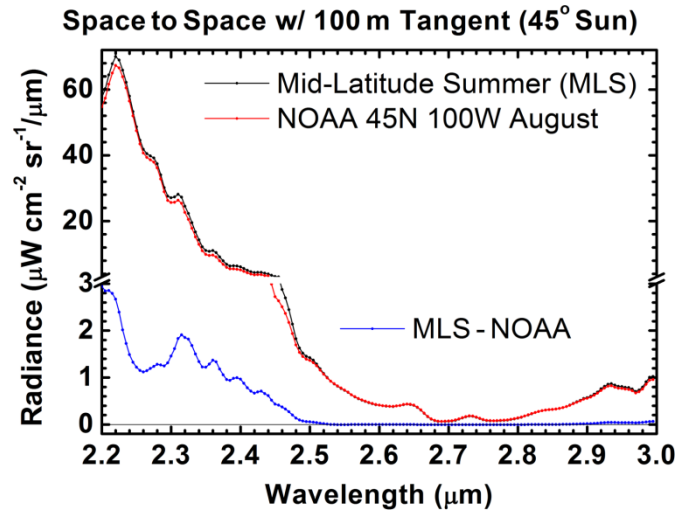


Figure 52. MODTRAN Limb Radiances Computed for the Profiles of Figure 51

4.6 Parallel Execution

The API developed for MODTRAN provides a user-friendly interface for easy access and use by software developers. We have also created an example of a basic console application that combines the MODTRAN6 API with a message passing interface (MPI) to accelerate a series of calculations on a parallel processing platform. This MPI-MODTRAN application could either operate standalone, using file-based input and output, or the concept illustrated by the source code could easily be ported into an existing application to reproduce our parallel execution approach.

The example application is modeled on a work queue of N processes, with a single master process controlling the scheduling among the $N-1$ processing elements to provide optimal load balancing. Input configurations are read as a sequence of case configurations, defined in a JSON or tape5 input file, and each case is distributed to an independent "worker" processor for parallel execution. Any input file compatible with the standard MODTRAN application will operate with this MPI program. However, to avoid overwriting of output the case configurations should provide unique file names in the FILEOPTIONS section of a JSON input.

One possible use-case of the parallel code is a parameter-sweep, or look-up table generation of MODTRAN calculations. These types of calculations are 'embarrassingly parallel,' requiring no inter-job communication. This is also the best type of parallelization in terms of load balancing, as each case will require approximately an equal amount of time to complete. An abbreviated example of this type of input is shown in Figure 53, which defines a sequence of calculations with varying water vapor concentrations using H2OSTR. The actual JSON input example, supplied with the application code, defines 41 cases for execution.

Figure 54 depicts the results of a performance benchmark generated by processing the described sample input of 41 cases on a multiprocessor system using between 1 and 21 workers. The acceleration score, shown in red, indicates the reduction in runtime as compared to a single process (i.e. 4 is 4x faster). The blue curve identifies the maximum number of cases executed by a single process for the testing conditions. Given this problem size, the test case provides excellent scaling with 7 or fewer worker processes. Beyond that point, the parallel acceleration becomes restricted as the amount of available tasks is divided among too many processors, leaving some workers idle while others are busy. This effect illustrates that if efficiency is important, the overall size of a problem should be tailored to make the best use of the available processing resources. If runtime performance is the top priority, utilizing too many processors will not hinder the performance to any large degree, and in some cases, may result in better load balancing.

```

{
  "MODTRAN": [
    {
      "MODTRANINPUT": {
        "NAME": "water_table",
        "CASE": 0,
        "RTOPTIONS": {
          "MODTRN": "RT_MODTRAN",
          "IEMSCT": "RT_SOLAR_AND_THERMAL",
          "IMULT": "RT_ISAACS_SCALED",
          "NSTR": 8
        },
      },
      ...
      "MODTRANINPUT": {
        "CASE": 1,
        "CASE_TEMPLATE": 0,
        "ATMOSPHERE": {
          "H2OSTR": 0.20
        }
      }
    }
  ]
}

```

Figure 53. Abbreviated Configuration Example of a Sweep of One Input Parameter, H2OSTR

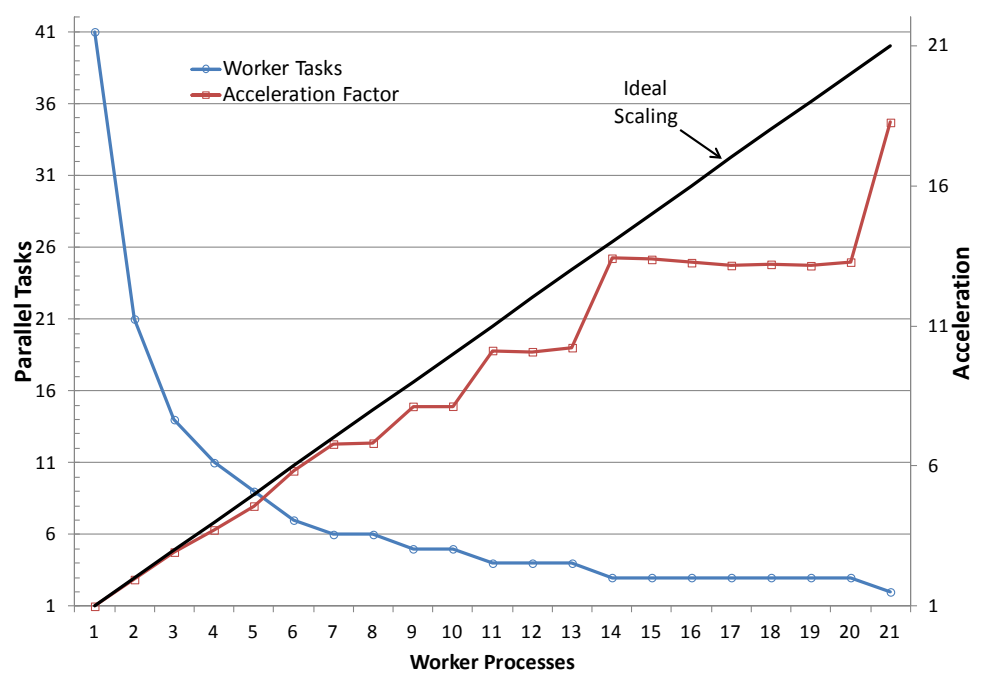


Figure 54. Acceleration Benchmark of the Example Parallel MODTRAN Application, using Between 1 and 21 Processors to Execute 41 Separate MODTRAN Calculations

5. CONCLUSIONS

MODTRAN serves as the US Air Force standard MODerate spectral resolution TRANsmittance code to solve the radiative transport equation, developed by Spectral Sciences, Inc. (SSI) and maintained collaboratively by SSI and the Air Force Research Laboratory (AFRL) for 30 years. MODTRAN computes line-of-sight (LOS) atmospheric spectral transmittances and radiances over the ultraviolet (UV) through long wavelength infrared (IR) spectral regime. The radiation transport (RT) physics within MODTRAN provides accurate and fast methods for modeling stratified, horizontally homogeneous atmospheres. MODTRAN5 introduced many new features to the code which included a 0.1 cm^{-1} band model resolution, the addition of auxiliary species, advanced aerosol profile options, and a local chemical cloud option. While the physics within MODTRAN was well-established, certain legacy features of the code made it difficult for users and developers to use and integrate in an efficient manner.

The NextGen MODTRAN effort, which has led to the development of MODTRAN6, was built upon the core Fortran77 radiative transport (RT) legacy code. MODTRAN6 retains all of the features of MODTRAN5 while making it easier for both the average user to utilize and software developers to incorporate into their applications. This was accomplished by (1) developing a new text input style and (2) developing a C/C++ application programming interface (API). The core RT remains in the Fortran language. However, all of the Fortran code has been converted to Fortran95 or Fortran2003 in order to remove antiquated features of Fortran77 and introduce managed, dynamically allocated memory and modular programming. The new text based input and the API facilitates better user interaction with MODTRAN and alleviates many of the drawbacks of and the steep learning curve associated with using earlier versions of MODTRAN.

In addition to the enhanced programming features, MODTRAN6 now includes a line-by-line (LBL) capability. As mentioned above, MODTRAN was originally designed exclusively as a band model based code, MODTRAN4 introduced a Correlated- k algorithm and MODTRAN5 provides a computational spectral resolution of 0.1 cm^{-1} . The band model spectral binning limited the range of application to systems operating at narrow to coarse spectral resolution, generally greater than 0.2 cm^{-1} (i.e., includes some filter function averaging). For systems involving active illumination using lasers, much higher spectral resolution is required. While software such as FASCODE2 FASE3 (FASCODE for the Environment), and LBLRTM4 (line-by-line radiative transfer model) exist to provide LBL calculations, MODTRAN has added an LBL option in order to provide a single code to compute high resolution RT calculations and serve as a means of validating the more approximate, but much faster, band model RT algorithms. MODTRAN also facilitates the calculation of LBL scattering, a non-standard option in most LBL codes.

Finally, a new MODTRAN toolkit provides an extended number of utility programs that makes it much easier for a non-expert user to generate and incorporate common but supplementary information, such as optical properties data and weather data, into a form suitable for MODTRAN.

REFERENCES

- [1] Berk, A., P. Conforti, R. Kennett, T. Perkins, F. Hawes, and J. van den Bosch, MODTRAN6: a major upgrade of the MODTRAN radiative transfer code, SPIE 9088, Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XX, 90880H (13 June 2014), doi: 10.1117/12.2050433.
- [2] Berk, A., P.K. Acharya, L.S. Bernstein, G.P. Anderson, P. Lewis, J.H. Chetwynd, and M.L. Hoke, Band Model Method for Modeling Atmospheric Propagation at Arbitrarily Fine Spectral Resolution, U.S. Patent #7433806, issued October 7, 2008.
- [3] Clough, S.A., M.J. Iacono, and J.-L. Moncet, Line-by-line calculation of atmospheric fluxes and cooling rates: Application to water vapor, J. Geophys. Res., 97, pp. 15761-15785, 1992.
- [4] Clough, S.A., M.W. Shephard, E.J. Mlawer, J.S. Delamere, M.J. Iacono, K. Cady-Pereira, S. Boukabara, and P.D. Brown, Atmospheric radiative transfer modeling: a summary of the AER codes, Short Communication, J. Quant. Spectrosc. Radiat. Transfer, 91, pp. 233-244, 2005.
- [5] Rothman, L.S., *et al.*, The HITRAN2012 molecular spectroscopic database, J. Quant. Spectrosc. Radiat. Transfer, 130, pp. 4-50, 2013.
- [6] Van Vleck, J.H. and D.L. Huber, Absorption, emission, and linebreadths: a semihistorical perspective, Rev. Mod. Phys., 49, p. 939, 1977.
- [7] Humlíček, J., The Voigt function and its derivatives, J. Quant. Spectrosc. Radiat. Transfer, 21, pp. 309-313, doi:10.1016/0022-4073(79)90062-1, 1979
- [8] Humlíček, J., Optimized computation of the Voigt and complex probability function, J. Quant. Spectrosc. Radiat. Transfer, 27, pp. 437-444, doi:10.1016/0022-4073(82)90078-4, 1982.
- [9] Delamere, J.S., S.A. Clough, V.H. Payne, E.J. Mlawer, D.D. Turner, and R.R. Gamache, A far-infrared radiative closure study in the Arctic: Application to water vapor, J. Geophys. Res., 115, D17106, doi:10.1029/2009JD012968, 2010.
- [10] Payne, V.H., E.J. Mlawer, K.E. Cady-Pereira, and J.-L. Moncet, Water vapor continuum absorption in the microwave, IEEE Trans. Geosci. Remote Sens, 49(6), pp. 2194-2208, doi 10.1109/TGRS.2010.209141, 2011.
- [11] Mlawer, E.J., V.H. Payne, J.-L. Moncet, J.S. Delamere, M.J. Alvarado, and D.D. Tobin, Development and recent evaluation of the MT_CKD model of continuum absorption, Phil. Trans. Roy. Soc. A, 370, pp. 1-37, doi:10.1098/rsta.2011.0295, 2012.
- [12] Mishchenko, M.I., L.D. Travis, and A.A. Lacis, Multiple Scattering of Light by Particles: Radiative Transfer and Coherent Backscattering, Cambridge University Press, Cambridge, MA, 2006.
- [13] Longtin, D.R., E. Shettle, J. Hummel, and J. Pryce, A wind dependent desert aerosol model: radiative properties, AFGL-TR-88-0112, Optimetrix, Inc., Burlington,, MA, 1988.

DISTRIBUTION LIST

DTIC/OCF

8725 John J. Kingman Rd, Suite 0944

Ft Belvoir, VA 22060-6218 1 cy

AFRL/RVIL

Kirtland AFB, NM 87117-5776 2 cys

Official Record Copy

AFRL/RVBYI/Dr. Jeannette van den Bosch 1 cy

This page is intentionally left blank.