Chapter 1
MODTRAN® Modeling Capability

1.1 Introduction

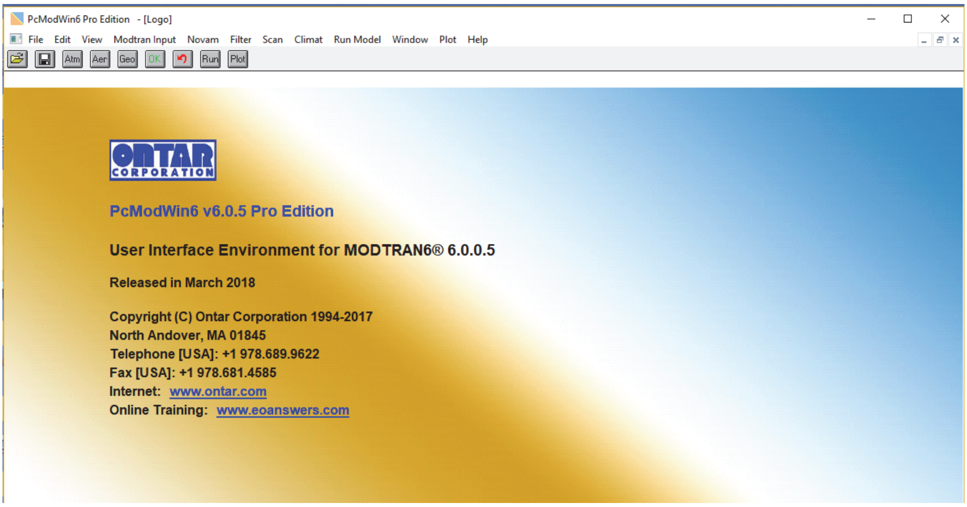
Atmospheric optical properties play an important role in many problems, such as military-systems analysis, climatology, pollution monitoring, global energy balance, etc. Unfortunately, atmospheric effects from molecular absorption/scattering, aerosol absorption/scattering, clouds (ice and water), solar scattering, path refraction and surface reflections, are very complex, sometimes overwhelming.[1,2] In the early 1970s, a strong need was recognized for a comprehensive and validated model to account for all the effects possible from the Earth's atmosphere. The USAF Cambridge Laboratory responded with long-term model development to meet this need, which led to MODTRAN®.[5–10]

MODTRAN, shorthand for MODerate-resolution TRANsmission, calculates the transmission and emission from either one of a set of pre-stored, or a custom, atmosphere for the analysis of electro-optical/infrared system performance and natural phenomena. In other words, it uses a generalized geometry package to describe atmospheric absorption, scattering, and emission paths by leveraging either included databases that describe (1) atmospheric profiles, (2) atmospheric aerosols, and (3) water and ice cloud representations, or user-defined atmospheres and/or characteristics. Its spectral coverage spans the optical wavelength range from the near ultraviolet (UV) through the far infrared (IR).

MODTRAN is a compiled FORTRAN and C++ program that can be run from the command line. It is compiled from over 50,000 lines of source code and gathers inputs from a primary text file, reads many other text and binary files with key parameter information, and then calculates the desired optical properties and stores them in other files. Unfortunately, its inputs and outputs are stored as ASCII files that are not the most desirable user format in today's world for fast computing. This generated a need for a better interface for the user and software.

PcModWin© is a Windows-based graphical user interface (GUI) for MODTRAN, which is shown in Fig. 1.1. It organizes MODTRAN inputs into more user-friendly screens, generates the resulting input files, and launches MODTRAN, as well as reads and interprets output files, and provides plotting and numerous other tools for systems and research analyses.

This chapter provides a high-level overview of the MODTRAN model definitions, functions, and geometries from government-approved resources.[5–10]



PcModWin6 v6.0.5 Pro Edition

User Interface Environment for MODTRAN6® 6.0.0.5

Released in March 2018

Copyright (C) Ontar Corporation 1994-2017
North Andover, MA 01845
Telephone [USA]: +1 978.689.9622
Fax [USA]: +1 978.681.4585
Internet: www.ontar.com
Online Training: www.eoanswers.com

Figure 1.1 PcModWin opening page.

1.2 A Short History

The initial modeling capability was developed as the “LOWTRAN” code at the USAF Cambridge Laboratory in the early 1970s. This tool addressed the need to rapidly estimate atmospheric transmission in the IR bands. It started as a series of tables, graphs, and a LOWTRAN slide rule but soon became an ever-increasing stack of cards and more complex code over time as users demanded a broader set of radiometric characterizations. This led to the more capable set of codes LOWTRAN2 through LOWTRAN7. This capability had a relatively low resolution (20 cm^{-1}), hence the origin of code’s name.

Based on the original community it served, MODTRAN worked in wavenumbers, which are expressed in inverse centimeters (cm^{-1}). A wavenumber describes the number of waves in a specific length interval, i.e., cycles per unit distance. It is used by spectroscopists, who were the first big users of the Cambridge model, and by some particle and quantum physicists. However, for everyone else, the preferred metric for research and analysis is the wavelength itself, say, in units of microns. The simple conversion equation between wavelengths and wavenumbers is given by

$$\lambda(\mu m) = 10000/\nu\left(cm^{-1}\right), \tag{1.1}$$

where λ is the wavelength of the optical radiation in microns, and ν again is its wavenumber in inverse centimeters. For example, 5-μm light has a wavenumber that equals 2000 cm^{-1}.

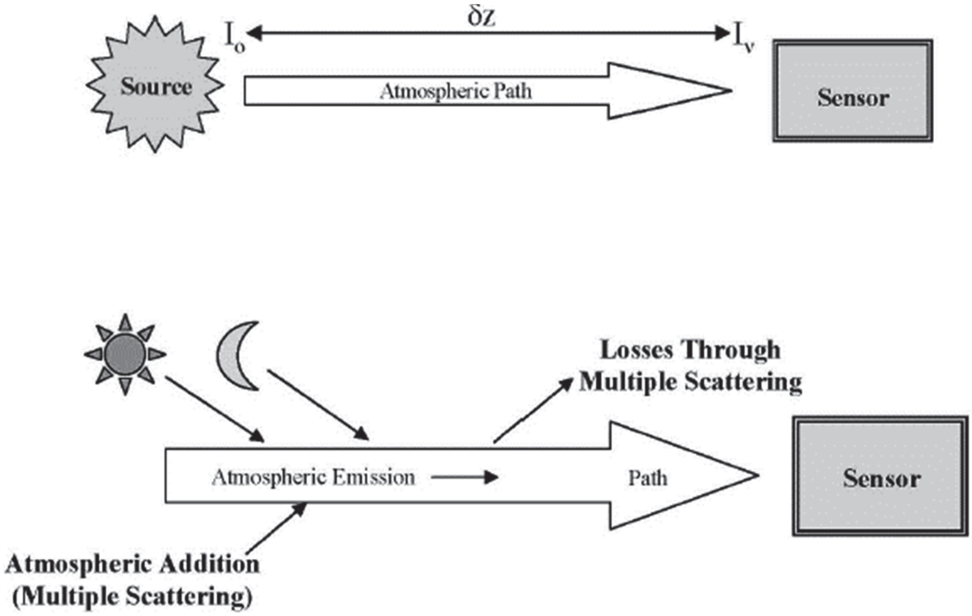


Figure 1.2 The MODTRAN view of transmittance, scattering, and emission.

Figure 1.2 illustrates what MODTRAN views as transmittance and radiance redistribution when an optical ray propagates from a source to a sensor. (Chapter 2 will discuss the basic loss mechanisms for the atmosphere, and molecular and particulate absorption and scattering.) This is the situation seen in the top picture of Fig. 1.2. Absorption takes any optical beam of energy that reduces what can be detected by the sensor. On the other hand, single and multiple scattering redistributes the original radiance distribution, which may or may not deprive the sensor of detected energy. This situation is represented by the lower picture in Fig. 1.2. Any energy loss will be determined by whether the redistribution stays within the field of view of the sensor. However, the lower picture also shows that external sources such as the Sun, Moon, stars, etc. can provide additional optical energy to the sensor (noise) by the same single and multiple scattering the source rays experience. Finally, atmospheric constituent emission can contribute optical radiation to the sensor in some cases. All of this can be modeled using MODTRAN.

The underlying physics and algorithms used in MODTRAN are well established.[11] Recent MODTRAN updates have focused on converting to a modern programming style and language, facilitating interfacing/integration with other applications, and utilizing the new Application Programming Interface (API) capability to provide code-parallelization speedup options. This capability is summarized in Fig. 1.3.

The key advancements included in the most recent MODTRAN release include

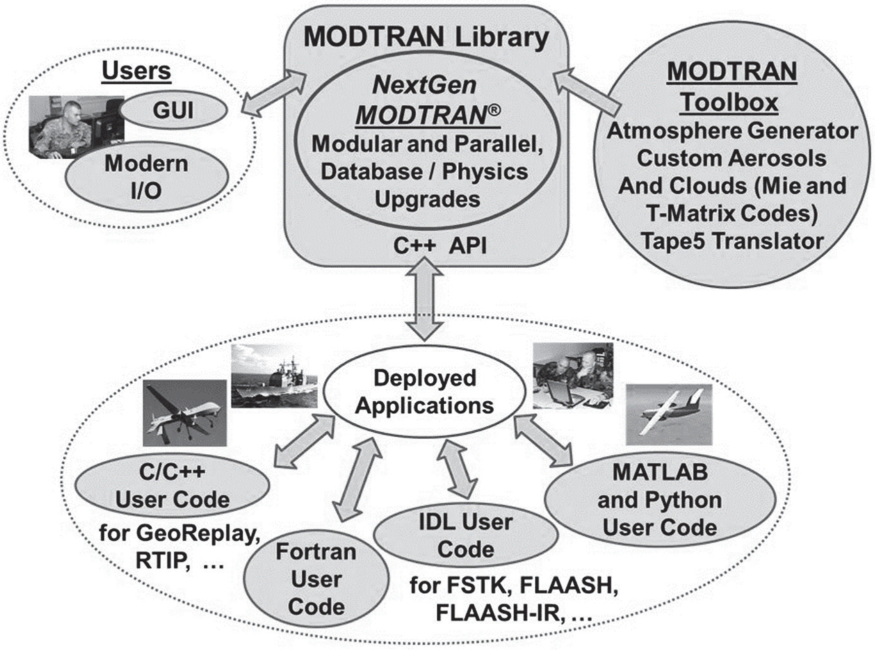


Figure 1.3 Overview of MODTRAN architecture, capabilities, and applications.

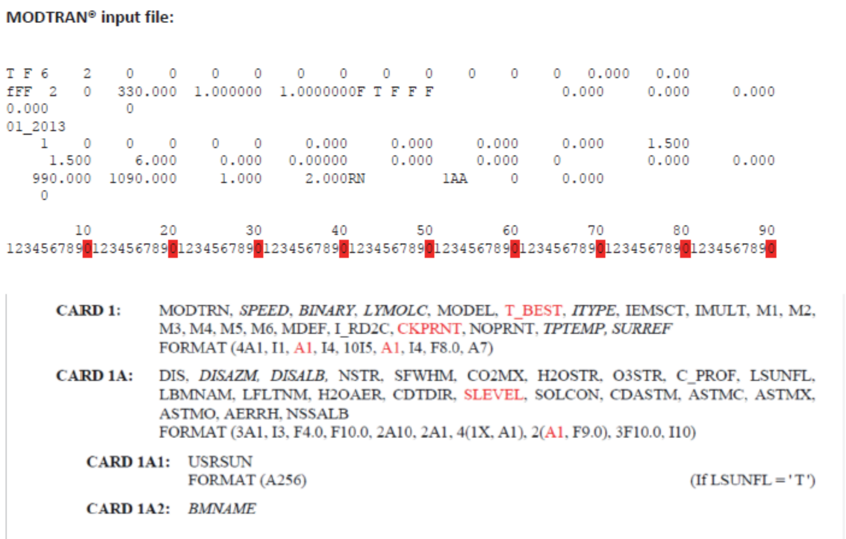
- **Integration**: MODTRAN can be incorporated with operational systems and third-party software.
- **Physics upgrade**: A line-by-line algorithm to enable higher-spectral-resolution applications and validation of the band model algorithms.
- **Algorithm speedup**: MODTRAN's updated algorithms and parallelization techniques can leverage advanced computing architectures to reduce wall clock time.
- **Toolkits**: Tools are available to convert legacy input files to a JavaScript Object Notation (JSON) format for defining radiosonde and climatology-based vertical profiles, and for generating custom aerosol optical and profile data.
- **Documentation**: A revised user's manual,[6] an Algorithm Theoretical Basis Document (ATBD) describing the fundamental radiative transfer physics,[11] an Interface Control Document (ICD) to facilitate API implementations, and Robodoc software documentation come with the installation.

Throughout its history, MODTRAN has been extensively validated, e.g., Berk et al.,[9] and it still serves as the community-standard atmospheric band model.

1.3 New User Interface

Multiple options exist for running the MODTRAN software. Experienced users may continue to run MODTRAN from an ASCII ‹name›.tp5 text file, but a JSON-formatted input file is recommended. Unlike a ‹name›.tp5 text input, JSON is not a fixed format. Variable and value associations made available by JSON make it easier for the user to identify the values of every MODTRAN input and provide the inputs in a flexible order. MODTRAN also allows placement of comments into a JSON file, as shown in the following figures. In addition, a few new MODTRAN features, such as the generation of ‹name›.csv (comma separate value) and ‹name›.sli (ENVI® spectral library) output, are only accessible from the JSON format. The tp5tojson utility creates a ‹name›.json file from an existing ‹name›.tp5 file. Default JSON inputs are delineated in a keywords.json file, located in the MODTRAN DATA directory. Figures 1.4 and 1.5 show original ACSII FORTRAN and FORTRAN-JSON input file formats, respectively.

Code developers wishing to integrate MODTRAN6 into their own software can now run MODTRAN using a library (modtran.dll) file with an application. For example, a class wrapper for the MODTRAN atmospheric-radiative-transfer code is available in the MATLAB® library. This wrapper encapsulates a run of a MODTRAN case. It provides methods for reading and writing MODTRAN case files, creating and modifying case files, calling MODTRAN, and reading the MODTRAN results. There also are many methods for plotting MODTRAN outputs. Note that the wrapper does not provide a GUI, and MODTRAN itself is not included; the wrapper requires a MODTRAN5 distribution, which can be purchased from Ontar Corporation. In addition, a version of MATLAB that supports the new MATLAB class files is necessary (post-2010a MATLAB release).



MODTRAN® input file:

```
T F 6    2    0    0    0    0    0    0    0    0    0    0    0   0.000   0.00
fFF  2   0   330.000  1.000000  1.0000000F T F F F               0.000     0.000     0.000
0.000         0
01_2013
    1    0    0    0    0    0     0.000     0.000     0.000     0.000     1.500
     1.500     6.000     0.000   0.00000     0.000     0.000    0          0.000     0.000
   990.000  1090.000     1.000     2.000RN         1AA     0     0.000
    0

        10        20        30        40        50        60        70        80        90
123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
```

CARD 1: MODTRN, *SPEED*, *BINARY*, *LYMOLC*, MODEL, T_BEST, *ITYPE*, IEMSCT, IMULT, M1, M2, M3, M4, M5, M6, MDEF, I_RD2C, CKPRNT, NOPRNT, *TPTEMP*, *SURREF*
FORMAT (4A1, I1, A1, I4, 10I5, A1, I4, F8.0, A7)

CARD 1A: DIS, *DISAZM*, *DISALB*, NSTR, SFWHM, CO2MX, H2OSTR, O3STR, C_PROF, LSUNFL, LBMNAM, LFLTNM, H2OAER, CDTDIR, SLEVEL, SOLCON, CDASTM, ASTMC, ASTMX, ASTMO, AERRH, NSSALB
FORMAT (3A1, I3, F4.0, F10.0, 2A10, 2A1, 4(1X, A1), 2(A1, F9.0), 3F10.0, I10)

CARD 1A1: USRSUN
FORMAT (A256) (If LSUNFL = 'T')

CARD 1A2: *BMNAME*

Figure 1.4 Example of original FORTRAN ASCII MODTRAN input format.