

The *am* atmospheric model

SMA technical memo #152 · version 14.0 · September 2024

Scott Paine
Smithsonian Astrophysical Observatory

DOI (this version): <https://doi.org/10.5281/zenodo.13748391>

DOI (all versions): <https://doi.org/10.5281/zenodo.640645>

Contents

Contents.....	iii
Preface.....	vii
Chapter 1: Introduction.....	1
1.1 Layers and Columns.....	1
1.1.1 Hydrostatic models.....	5
1.1.2 Defining temperatures at layer boundaries.....	6
1.1.3 Hydrostatic layer thickness.....	7
1.1.4 Choosing a layer parametrization.....	8
1.2 Optical depth and radiative transfer computations.....	8
1.2.1 Layer opacities.....	9
1.2.2 Radiative transfer across a single layer.....	9
1.2.3 Radiative transfer through the layer stack.....	10
1.2.4 Spectra derived from radiance and opacity.....	11
References.....	12
Chapter 2: Using <i>am</i>.....	15
2.1 Getting started.....	15
2.1.1 Testing the installation.....	15
2.1.2 Using the program.....	16
2.1.3 Windows-specific issues.....	17
2.2 Configuration file examples.....	18
2.2.1 A laboratory propagation path.....	18
2.2.2 Adding a command-line parameter.....	20
2.2.3 Adding another column to the layer.....	22
2.2.4 A simple atmospheric model.....	23
2.2.5 A realistic model for Mauna Kea.....	26

2.3 Accelerating computations.....	34
2.3.1 Self-broadening and the disk cache.....	38
2.4 The <i>am</i> cookbook.....	40
References.....	40
Chapter 3: Line-by-line Absorption.....	43
3.1 The line-by-line absorption coefficient.....	43
3.2 Line shapes.....	45
3.2.1 Approximate time scales for line broadening processes.....	45
3.2.2 Collisional line width and line shift.....	47
3.3 Line shapes in <i>am</i>	47
3.3.1 Collisional line shapes.....	47
3.3.2 Line shape for coupled lines.....	50
3.3.3 Doppler and Voigt line shapes.....	51
3.3.4 Default line shapes.....	51
3.4 Molecular species in <i>am</i>	52
3.4.1 Water Vapor (column type <code>h2o_lines</code>).....	52
3.4.2 Ozone (column type <code>o3</code>).....	55
3.4.3 Oxygen (column type <code>o2</code>).....	55
3.4.4 Methane (column type <code>ch4</code>).....	60
3.4.5 Carbon Monoxide (column type <code>co</code>).....	60
3.4.6 Carbon Dioxide (column type <code>co2</code>).....	60
3.4.7 Nitrous Oxide (column type <code>n2o</code>).....	62
3.4.8 Carbonyl Sulfide (column type <code>ocs</code>).....	62
3.5 A note on computation.....	64
References.....	65
Chapter 4: Continuum Absorption.....	69
4.1 Collision-induced absorption.....	69
4.1.1 Binary absorption coefficients.....	70
4.1.2 N ₂ -N ₂ CIA.....	70
4.1.3 N ₂ -O ₂ , O ₂ -N ₂ , and O ₂ -O ₂ CIA.....	74
4.1.4 Using CIA column types.....	76
4.2 H ₂ O continuum absorption.....	77
4.2.1 Water vapor in <i>am</i> models.....	79
4.3 Liquid water and ice absorption in clouds.....	81
4.3.1 Absorption and scattering in the Rayleigh limit.....	81
4.3.2 Liquid water and ice in <i>am</i> models.....	83
4.4 Gray opacity.....	85
References.....	86
Chapter 5: Modeling Radiometers.....	89
5.1 Instrumental line shapes.....	89
5.1.1 Instrumental line shapes in <i>am</i>	90
5.1.2 Using instrumental line shapes.....	94

5.2 Difference spectra.....	97
5.3 IF spectra.....	98
5.3.1 Producing IF spectra from <i>am</i> models.....	99
5.3.2 System temperature and spectral Y-factor.....	101
References.....	102
Chapter 6: Delay Spectra.....	105
6.1 Delay computation.....	105
6.1.1 Kramers–Kronig relations.....	105
6.1.2 Opacity and delay for spectral lines.....	107
6.1.3 The Smith–Weintraub equation.....	108
6.1.4 Complete delay computation.....	109
6.2 Examples.....	110
References.....	114
Chapter 7: Fits.....	117
7.1 A worked example.....	118
7.1.1 The spectral data.....	118
7.1.2 Setting up the model.....	120
7.1.3 Running the fit.....	127
7.1.4 Bootstrap analysis of statistical uncertainties.....	131
7.1.5 Systematic calibration accuracy.....	135
7.2 More about fits.....	136
References.....	136
Chapter 8: Jacobians.....	139
8.1 Computing Jacobians with <i>am</i>	140
8.2 Examples.....	141
8.2.1 An example from water vapor radiometry.....	141
8.2.2 An example from ozone line radiometry.....	145
References.....	148
Chapter 9: Geometry and Refraction.....	151
9.1 Spherical geometry and refraction in <i>am</i> models.....	152
9.1.1 Path geometries and interpolated levels.....	152
9.1.2 Levels outside the atmosphere.....	155
9.1.3 Geodetic parameters.....	157
9.1.4 Refraction models.....	159
9.2 Examples.....	159
9.2.1 A point-to-point propagation utility.....	160
9.2.2 Distance to the radio horizon.....	169
9.2.3 Refraction and airmass vs. zenith angle at the SMA.....	172
9.2.4 Background loading on a balloon-borne detector.....	176
References.....	180

Appendix A: Installation and Performance.....	181
A.1 Installation.....	181
A.1.1 Installing on Windows.....	181
A.1.2 Installing on GNU/Linux, Unix, and macOS.....	182
A.2 Compile-time options.....	184
A.3 Environment Variables.....	185
A.3.1 Program-specific environment variables.....	185
A.3.2 OpenMP environment variables.....	186
A.4 Performance.....	188
A.4.1 Line-by-line computations.....	189
A.4.2 Delay spectra and ILS convolution.....	193
References.....	197
Appendix B: Reference Manual.....	199
B.1 Keywords and statement syntax.....	199
B.2 Fit and differentiation variables.....	199
B.3 Keyword reference.....	200
B.4 Column types.....	222
B.4.1 Line-by-line column types.....	222
B.4.2 Collision-induced absorption and H ₂ O continuum.....	227
B.4.3 Liquid water and ice absorption.....	228
B.4.4 Gray attenuation.....	229
B.4.5 Optical refractivity.....	230
B.4.6 Composite column types.....	231
B.5 Units.....	232
References.....	236

Preface

am is a tool for optical depth, radiative transfer, and refraction computations involving propagation through the atmosphere and other media at microwave through submillimeter wavelengths. The atmosphere can be modeled as hydrostatic plane-parallel or spherical layers. More generally, the program can be used for any problem that can be modeled as a narrow beam propagating through a sequence of path segments having user-specified composition, pressure, and temperature under conditions of local thermodynamic equilibrium.

Spectra that can be computed with *am* include absorption, thermal emission, and excess delay. The program can modify these spectra to model the response of spectrometers and receiving systems, and can compute Jacobians of output spectra differentiated with respect to selected model parameters. Measured spectra can be fitted using a simple scheme for flagging model parameters as fit variables. The program can also be used as a convenient tool for path geometry and refraction computations without invoking any spectral computations.

The original motivating applications for *am* were in radio astronomy and ground-based atmospheric radiometry, involving terrestrial propagation paths between ground and sky. However, the program has since been applied to a variety of other radiative transfer problems and path geometries, and has been used for applications including laboratory receiver testing, millimeter-wave radio, and radio spectrum management.

am operates on configuration files that contain statements defining the atmospheric model and propagation path, together with other statements that control the spectroscopic and radiative transfer computations, instrument models, and program outputs. A configuration file can be an actual file, or it can be a stream generated by another program. Likewise, spectral data for fits can come from a file or from a stream.

am is written in portable C and runs in a variety of environments, including GNU/Linux, Unix, macOS, and Windows. When built with a compiler such as *gcc* that supports OpenMP, *am* will make efficient use of multicore and shared-memory multiprocessor machines.

This document is organized as follows:

Chapter 1 introduces the layer model used in *am*, and its extension to the particular case of a hydrostatic atmosphere in vertical pressure coordinates. It also describes the radiative transfer model implemented in *am*.

Chapter 2, *Using am*, is a tutorial introduction to the program and to the model configuration files which control its operation. It explains how to run the program, and introduces the structure of *am* configuration files through a series of examples. Chapter 2 also discusses methods for accelerating spectral computations.

Chapter 3, Line-by-line Absorption, and Chapter 4, Continuum absorption, provide more detailed information about the physics and computation of spectral absorption coefficients for the various species implemented in *am*. These chapters serve as a guide and reference to the underlying literature and catalog data, and as top-level documentation for the *am* source code. They are also intended to help the user assess the applicability of *am* to their particular problem.

Chapter 5, Modeling Radiometers, discusses how the instrumental properties of spectrometers or receiver systems can be incorporated into models to simulate measured spectra.

Chapter 6, Delay Spectra, describes basic concepts of radio refractivity and the Hilbert transform method used by *am* to compute delay spectra.

Chapter 7, Fits, explains how to use *am* to fit measured spectra by flagging model parameters as fit variables.

Chapter 8, Jacobians, explains how to compute Jacobians of any output spectrum differentiated with respect to a set of one or more model parameters at a given model reference state.

Chapter 9, Geometry and Refraction, explains how spherical geometry, refraction, and layer interpolation can be applied to *am* models in hydrostatic pressure coordinates, and presents examples of applications in a spherical atmosphere.

Appendix A, Installation and Performance, tells how to obtain and install the program, and how to optimize its performance in a particular environment.

Appendix B, Reference Manual, provides detailed information about configuration file keywords and syntax, absorbing species (called *column types*) implemented in *am*, and units recognized by the program.

It is a pleasure to acknowledge those who have contributed to this program through many helpful discussions and comments, starting from its initial conception as an easy-to-use, scriptable tool, and continuing through its ongoing development, validation, and documentation: especially James Battat, Ray Blundell, Hugh Gibson, Paul Grimes, Richard Hills, Todd Hunter, Jon Kawamura, Nathaniel Livesey, Dan Marrone, Eli Mlawer, and Bob Wilson. Thanks also to those who have taken the time to report bugs in the program or errors in the documentation; please send these reports to spaine@cfa.harvard.edu.

Chapter 1: Introduction

am is a program for optical depth, radiative transfer, and refraction computations at microwave through submillimeter wavelengths, for radiation paths that can be modeled as a sequence of atmospheric layers or user-defined path segments. This chapter provides the essential background for using the program. The layer model used by *am* is introduced first, and the various ways that the layers can be parametrized are described. This is followed by a discussion of how *am* computes radiative transfer through the layers.

In this chapter we'll consider only simple planar layers and plane-parallel atmospheric layers in hydrostatic vertical pressure coordinates, and refraction will be neglected. In many practical situations encountered in radio astronomy and ground-based remote sensing, this is an excellent approximation, and the accuracy of our computations will be limited by other factors. Spherical geometry and refractive bending of the propagation path, which must be considered for limb paths and paths near the astronomical horizon, are discussed in Chapter 9.

The mechanics of actually running the program will be taken up in Chapter 2. Some readers may prefer to browse ahead to the examples there to get a feel for how *am* works, then return here for a more formal introduction.

1.1 Layers and Columns

The simplest *am* propagation path model is illustrated in Figure 1.1. Radiation, initially having a Planck spectrum at a background temperature T_0 , propagates through a series of *layers*. Most often, these layers will represent horizontal strata in the atmosphere. More generally, however, a layer simply represents a segment of a propagation path. For example, a single layer could represent a horizontal path at constant pressure and temperature between two points in the atmosphere, a path within a building, or a path through the contents of a gas cell in the laboratory. Figure 1.1 also shows the first of several possible parametrizations of the layer pressures and temperatures that can be used in *am*—here, P and T are defined at the layer midpoints, and are assumed constant throughout each layer. Other ways to parametrize layer pressures and

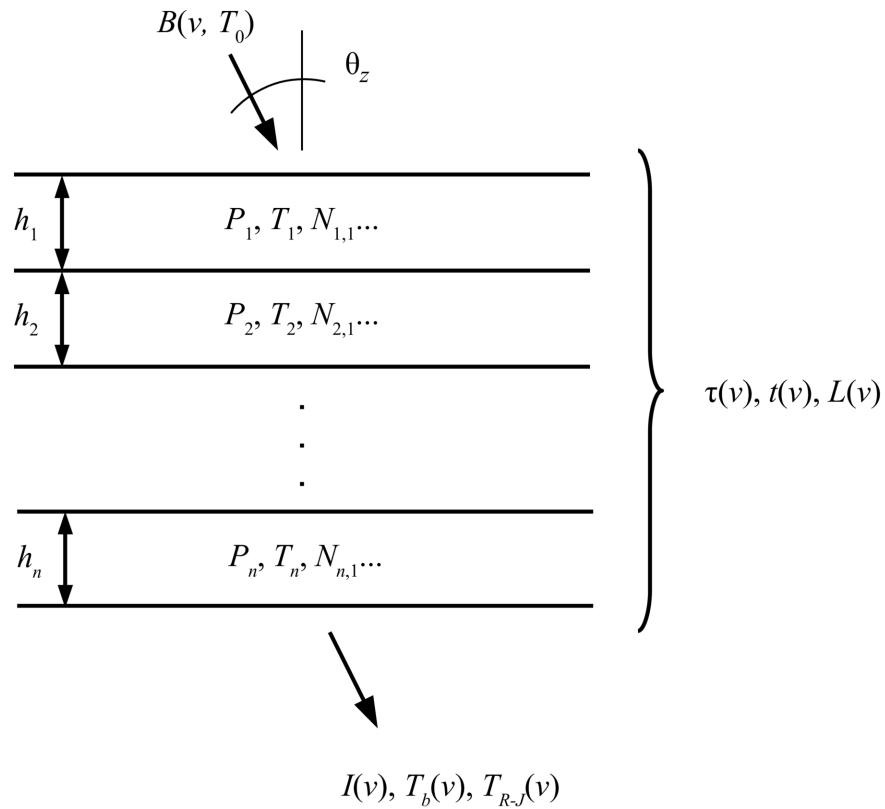


Figure 1.1 The *am* layer model. In this example, layer pressures and temperatures have been defined at the layer midpoints

temperatures are introduced later in this section.

Each layer can contain a mixture of absorbing species between the layer boundaries. The type and column density of the various species contained in a layer can be individually defined, and each such definition within a layer is called a *column*. As will be seen later, the notion of a column in *am* is generalized beyond the specification of individual molecular species. A column definition can refer to a mixture of species, such as dry air, or to a particular absorption process associated with a given species or mixture, such as line-by-line or collision-induced absorption. Special column types are also available for modeling absorption phenomenologically.

Figure 1.1 also shows the basic spectra that can be computed by *am*. Some of these are properties of the layer stack, including the opacity (optical depth) $\tau(\nu)$, transmittance $t(\nu)$, and excess propagation delay $L(\nu)$. Others are spectral properties of the radiation exiting the n^{th} layer, including radiance $I(\nu)$, Planck brightness temperature $T_b(\nu)$, and Rayleigh-Jeans brightness temperature $T_{R-J}(\nu)$. More details about the computation of these spectra are given in Section 1.2, below. From these basic spectra, other spectra modeling the response of radiometers or receiving systems can be derived as discussed in Chapter 5. In addition, bare spectral absorption coefficients can be requested as a program output—details on how to do this are given in the entry for the `output` keyword in Appendix B.

By default, *am* does not impose a strict relationship between the total pressure P_i of layer i and the column density N_{ij} of the j^{th} column defined on that layer. This is because definition of the layer thickness h_i is optional. If h_i is undefined, the layer pressure P_i is simply treated as a background air pressure used for computing pressure-broadened line widths and collision-induced absorption coefficients, assuming default dry-air mixing ratios. (Here, as always in *am*, the term *mixing ratio* refers to a volume mixing ratio, i.e. the ratio of the number density of the species in question to the total number density of all species.)

Alternatively, if h_i has been defined, it is possible to derive the total column density N_i for all species on for a layer from the ideal gas law:

$$N_i = h_i \cdot n_0 \cdot \left(\frac{P_i}{P_0} \right) \cdot \left(\frac{T_0}{T_i} \right). \quad (1.1)$$

Here, n_0 is the Loschmidt constant,

$$n_0 = 2.6867811(15) \cdot 10^{19} \text{ cm}^{-3} \quad (1.2)$$

defined at standard pressure

$$P_0 = 1013.25 \text{ mbar}, \quad (1.3)$$

and temperature

$$T_0 = 273.15 \text{ K}. \quad (1.4)$$

For a given column on a layer, the volume mixing ratio is

$$x_{ij} = \frac{N_{ij}}{N_i}. \quad (1.5)$$

So, on a layer for which h_i has been defined, the column density N_{ij} can either be defined directly, or indirectly via a mixing ratio as

$$N_{ij} = x_{ij} \cdot N_i. \quad (1.6)$$

For components of dry air having non-zero default mixing ratios, *am* also allows column densities to be computed automatically. Any explicitly-defined mixing ratios on a layer, typically for variable trace species such as water vapor or ozone, are first subtracted from unity. The result is applied as a scaling factor to the default mixing ratios of any remaining components on the layer. Default mixing ratios for the different column types in *am* are listed in the tables at the end of

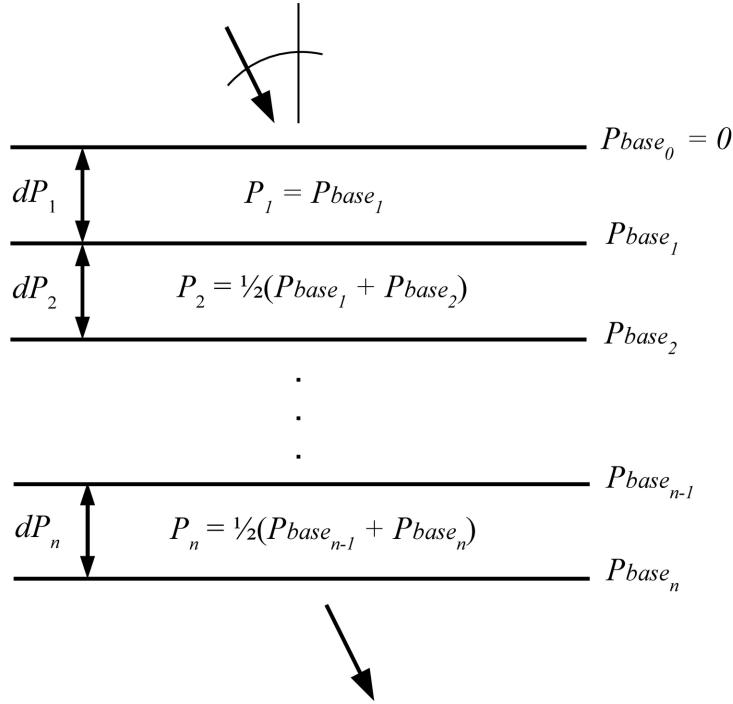


Figure 1.2 Hydrostatic layers.

Appendix B.

As long as there are no such automatically computed mixing ratios on a layer, the sum of the mixing ratios for all of the columns contained in that layer is allowed to be less than unity—this is because the total pressure P_i may include background gases that contribute to pressure broadening but are not themselves radiatively active in the frequency band of interest. Conversely, if the sum of the mixing ratios on a layer exceeds unity, *am* will issue a warning but carry on with the computation assuming the total layer pressure is nonetheless still P_i , in blithe disregard of Dalton's law of partial pressures. This makes it possible for different radiation mechanisms (e.g. line-by-line versus collision-induced absorption), involving the same species (e.g. O₂), to be specified using different column types. Examples of *am*'s mixing ratio rules in action are given in Chapter 2.

On layers with h_i defined, any mixing ratios not already given explicitly can be derived from the column density. It is then possible to compute pressure-broadened line widths and collision-induced absorption coefficients using actual mixing ratios rather than standard dry-air mixing ratios for every species on the layer. However, as a performance-related approximation, the default behavior of *am* is to ignore small differences between actual and default mixing ratios when computing the self-broadening contribution to collisional line widths. This is discussed further in Section 2.3.1.

1.1.1 Hydrostatic models

The typical application for *am* is the terrestrial atmosphere in hydrostatic equilibrium. The model layers then represent horizontal strata, and it is natural to use vertical pressure coordinates for the layer boundaries, as illustrated in Figure 1.2. A key advantage of using pressure as the vertical coordinate is that, in hydrostatic equilibrium, mass is conserved between pressure levels. If N is the total gas column density for all species on a layer, and \bar{M} is the mean molecular mass, then the gravitational force per unit area $N \bar{M} g$ is balanced by the pressure drop dP across the layer, so that

$$N = \frac{dP}{\bar{M} \cdot g}. \quad (1.7)$$

In computing \bar{M} , radiatively inactive background gas and radiatively active gas specified by column definitions on the layer must both be accounted for. In *am* models, these can be present in any combination, and the column densities of the radiatively active gases can either be defined explicitly, or defined implicitly by mixing ratio. If the background gas has mean molecular mass M_0 , while species defined by column densities N_j have molecular masses M_j , and species defined by mixing ratios x_k have respective molecular masses M_k , then

$$\bar{M} = M_0 \cdot \frac{1 - \sum_k x_k \left(1 - \frac{M_k}{M_0}\right)}{1 + \sum_j \frac{N_j}{N_0} \left(1 - \frac{M_j}{M_0}\right)}, \quad (1.8)$$

where

$$N_0 = \frac{dP}{M_0 \cdot g}. \quad (1.9)$$

By default, *am* assigns to M_0 the standard dry air value

$$M_0 = 28.964 \text{ amu}, \quad (1.10)$$

and assigns to g the standard value

$$g = 9.80665 \text{ m} \cdot \text{s}^{-2}, \quad (1.11)$$

although M_0 or g can be assigned different values by the user if needed. Normally, \bar{M} in Eq. 1.8 will be very nearly equal to M_0 . When evaluating Eq. 1.8, the default behavior of *am* is to consider water vapor and ozone only, but this too can be changed by the user.

In non-hydrostatic as well as hydrostatic models, absorption coefficients are assumed to be constant across a layer, and are computed at the midpoint pressure and at an interpolated temperature as discussed below. The midpoint pressure for hydrostatic models is defined as

$$P_i = \frac{1}{2} \cdot (P_{base_i} + P_{base_{i-1}}). \quad (1.12)$$

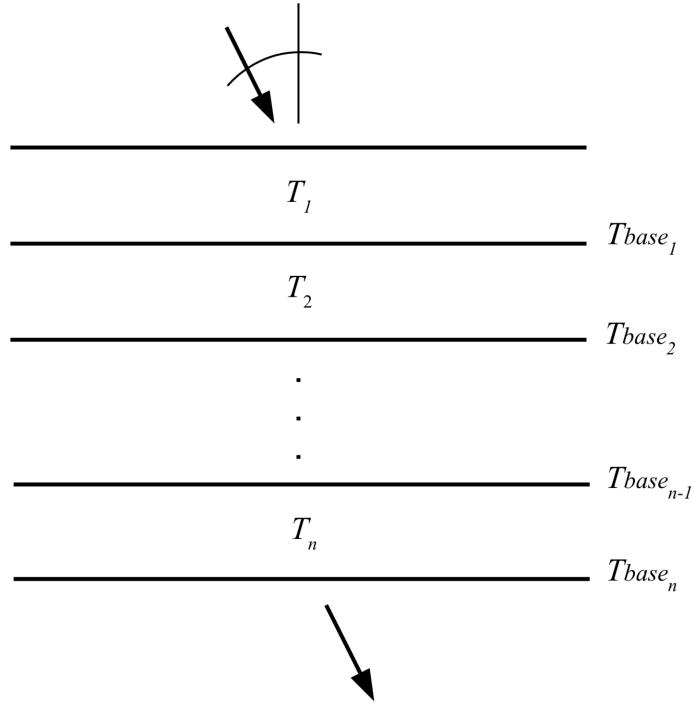


Figure 1.3 Layer temperatures defined using base temperatures. Absorption coefficients are computed at the layer midpoint temperatures and pressures, and are assumed constant across a layer. Planck functions are assumed to vary linearly with optical depth across a layer.

The top layer, however, is an exception. For this layer, the hydrostatic column density is computed assuming the upper layer boundary has $P=0$, but the effective layer pressure is not interpolated to the midpoint, so

$$dP_1 = P_{base_1} , \quad P_1 = P_{base_1} . \quad (1.13)$$

Where this is not desirable, an empty layer, containing no columns, can be placed at the top of the layer stack to set the upper boundary pressure of the first non-empty layer. This can also be desirable in order to obtain a meaningful airmass factor for the top layer in spherical models, because a layer with upper bound at infinity has an airmass factor asymptotically equal to unity independent of the zenith angle at the layer base level.

1.1.2 Defining temperatures at layer boundaries

In models with pressures defined on layer boundary levels, temperatures are normally defined on the boundary levels also, as illustrated in Figure 1.3. When layer temperatures are defined this way, absorption coefficients within a layer are still assumed constant, and are computed at an interpolated temperature

$$T_i = u_i \cdot T_{base_i} + (1 - u_i) \cdot T_{base_{i-1}}, \quad (1.14)$$

where, for hydrostatic models,

$$u_i = \frac{\log(P_i) - \log(P_{base_{i-1}})}{\log(P_{base_i}) - \log(P_{base_{i-1}})}, \quad (1.15)$$

so that T_i is interpolated linearly in $\log(P)$, locally approximating an adiabatic temperature profile. For non-hydrostatic models, T_i is simply set to the mean of the boundary temperatures,

$$T_i = \frac{1}{2}(T_{base_i} + T_{base_{i-1}}). \quad (1.16)$$

In either case, the top layer is an exception, with

$$T_1 = T_{base_1}. \quad (1.17)$$

As with the pressure, this behavior can be circumvented by making the top layer an empty layer.

Defining layer temperatures at boundaries versus at midpoints has an important effect on *am*'s treatment of the Planck radiance spectrum within a layer. Namely, when temperatures are defined at layer midpoints, the Planck radiance spectrum computed at the midpoint temperature is assumed to be constant throughout each layer. In contrast, when temperatures are defined on layer boundaries, a better approximation is used in which the Planck radiance is assumed to vary linearly with optical depth across each layer. This is discussed below in Section 1.2.2.

1.1.3 Hydrostatic layer thickness

In applications involving plane-parallel hydrostatic models, we can usually work in pressure coordinates only. However, for spherical models, discussed in Chapter 9, we need to know the geometric thickness of the layers. Noting that the total molecular column density N of a thin layer is $n dz$, where dz is the layer thickness and n is the volume density, Eq. 1.7 for hydrostatic mass balance can be written as

$$n dz = -\frac{dP}{\bar{M} \cdot g}. \quad (1.18)$$

Using the ideal gas law $P = nkT$ to eliminate n , we obtain

$$\begin{aligned} dz &= -\frac{kT}{\bar{M}g} \frac{dP}{P} \\ &= -\frac{kT}{\bar{M}g} d(\log P) \end{aligned} \quad (1.19)$$

To find the thickness of the i^{th} model layer, bounded by pressure levels P_{base_i} and $P_{base_{i-1}}$, we approximate \bar{M} and g as constant across the layer and integrate Eq. 1.19 to obtain

$$z_{base_{i-1}} - z_{base_i} = -\frac{k}{\bar{M}g} \int_{P_{base_i}}^{P_{base_{i-1}}} T d(\log P) \quad (1.20)$$

Eq. 1.20 may be rewritten (Andrews 2000) as

$$z_{base_{i-1}} - z_{base_i} = \frac{k \bar{T}}{M g} \log \left(\frac{P_{base_i}}{P_{base_{i-1}}} \right), \quad (1.21)$$

where

$$\bar{T} = \frac{\int_{P_{base_{i-1}}}^{P_{base_i}} T d(\log P)}{\int_{P_{base_{i-1}}}^{P_{base_i}} d(\log P)} \quad (1.22)$$

is the mean value of T integrated over $\log P$. As noted in the previous section, for a typical adiabatic temperature profile T is locally linear in $\log P$ so that

$$\bar{T} = \frac{1}{2} (T_{base_i} + T_{base_{i-1}}). \quad (1.23)$$

This is the approximation used in *am* when temperatures are defined on layer boundaries.

As a special case, *am* also allows a thickness h to be directly specified for a layer within a hydrostatic model, if and only if that layer has identical upper and lower boundary pressures. This makes it possible to insert short path segments at constant pressure into a model, for instance to represent a path segment within an instrument or through an environmental feature such as a gas plume. Such directly specified thicknesses are ignored when summing layer thicknesses to determine total geometric height z .

1.1.4 Choosing a layer parametrization

The choice of how to parametrize model layers depends on the application. For paths through the terrestrial atmosphere that are not too close to the astronomical horizon, hydrostatic plane-parallel layers with the layer temperatures defined on the layer boundaries will be a suitable choice. In spherical geometry modes, the layers are parametrized the same way in vertical hydrostatic pressure coordinates, and the geometry is worked out by *am* at run time as discussed in Chapter 9.

Simple point-to-point paths, paths through a laboratory setup, or paths made up of segments in the form of *am* layer data generated by another program, might instead be represented using layers defined by midpoint P and T , with the thickness h of each layer defined explicitly.

The model configuration file examples in Chapter 2 will demonstrate both of these common parametrizations.

1.2 Optical depth and radiative transfer computations

The radiative transfer model in *am* makes several simplifying approximations. First, local thermodynamic equilibrium is assumed to hold. This means that molecular collisions suffice to maintain equality of the molecular excitation temperature and the gas kinetic temperature. Coupling to radiation that may be at a very different effective temperature—such as that from

other atmospheric regions or from the Sun, Earth, or cosmic background—is assumed to have negligible effect on molecular excitation temperature. In addition, multiple scattering is neglected, and radiation is assumed to be unpolarized. This section describes the radiative transfer computation under these assumptions, starting with the computation of the optical depth of a single layer. This is followed by a description of radiative transfer through a single layer and then iteratively through the entire layer stack.

1.2.1 Layer opacities

The total opacity or optical depth $\tau_i(\nu)$ of a layer is the sum of the opacities of all columns (here indexed by j) defined on the layer,

$$\tau_i(\nu) = \sum_j \tau_{i,j}(\nu). \quad (1.24)$$

The form of the opacity of a given column depends on the column type. In general, the opacity is proportional to the product of the column density and, for a plane-parallel layer, the secant of the angle of propagation through the layer. For collision-broadened line spectra,

$$\tau_{i,j}(\nu) = N_{i,j} \cdot \sec(\theta_z) \cdot k_{i,j}(\nu, P, T), \quad (1.25)$$

where $k_{i,j}(\nu, P, T)$ [cm^2] is a molecular absorption coefficient. In the absence of collisions, as for pure Doppler broadening, the absorption coefficient is independent of pressure,

$$\tau_{i,j}(\nu) = N_{i,j} \cdot \sec(\theta_z) \cdot k_{i,j}(\nu, T). \quad (1.26)$$

More details on computation of line spectra are given in Chapter 3.

For collision-induced absorption and other binary processes, the opacity depends on the product of the column density and the volume density n_p of a background perturber,

$$\tau_{i,j}(\nu) = N_{i,j} \cdot n_{p,i,j} \cdot \sec(\theta_z) \cdot k_{i,j}^{(b)}(\nu, T), \quad (1.27)$$

where $k_{i,j}^{(b)}(\nu, T)$ [cm^5] is a binary molecular absorption coefficient. More details are given in Chapter 4.

1.2.2 Radiative transfer across a single layer

For models having layer temperatures defined by the layer midpoint temperature T_i , the Planck radiance function

$$\begin{aligned} B(\nu, T) &= \frac{2kT\nu^2}{c^2} \left[\frac{\frac{h\nu}{kT}}{e^{h\nu/kT} - 1} \right] \\ &= \frac{2h\nu^3}{c^2} \left[\frac{1}{e^{h\nu/kT} - 1} \right] \end{aligned} \quad (1.28)$$

is computed at $T = T_i$ for each layer, and assumed to be constant across the layer. Under this assumption, downward radiative transfer across a layer takes the form

$$I_i(\nu) = B(\nu, T_i) \cdot (1 - e^{-\tau_i(\nu)}) + I_{i-1}(\nu) \cdot e^{-\tau_i(\nu)}, \quad (1.29)$$

where $I_i(\nu)$ is the radiance at the lower boundary of the i^{th} layer, and $\tau_i(\nu)$ is the total opacity of the layer.

For models having temperatures defined at the layer boundaries, the Planck function is assumed to vary linearly with optical depth across the layer. This is the so-called *linear-in- τ* approximation, under which the radiative transfer across a layer can be integrated exactly (Clough et al. 1992), yielding

$$I_i(\nu) = B(\nu, T_{\text{base}_i}) \cdot p(\tau_i(\nu)) + B(\nu, T_{\text{base}_{i-1}}) \cdot q(\tau_i(\nu)) + I_{i-1}(\nu) \cdot e^{-\tau_i(\nu)} \quad (1.30)$$

where

$$p(\tau) = 1 - \frac{1 - e^{-\tau}}{\tau} \quad (1.31)$$

and

$$q(\tau) = \frac{1 - e^{-\tau}}{\tau} - e^{-\tau}. \quad (1.32)$$

The optically thick limit of these expressions is simply

$$\begin{aligned} p(\tau) &\rightarrow 1 & \text{as } \tau \rightarrow \infty. \\ q(\tau) &\rightarrow 0 & \end{aligned} \quad (1.33)$$

In this limit the spectral radiance emerging from the layer is just the Planck radiance at the lower boundary temperature. In contrast, the optically thin limits are the Taylor series

$$\begin{aligned} p(\tau) &\rightarrow \frac{\tau}{2} - \frac{\tau^2}{6} + \frac{\tau^3}{24} - \dots \\ q(\tau) &\rightarrow \frac{\tau}{2} - \frac{\tau^2}{3} + \frac{\tau^3}{8} - \dots \end{aligned} \quad \text{as } \tau \rightarrow 0, \quad (1.34)$$

from which it can be seen that the effective Planck function of the layer approaches the average of the Planck functions at the layer boundaries as $\tau \rightarrow 0$.

1.2.3 Radiative transfer through the layer stack

The radiative transfer computation through the complete layer stack is carried out iteratively. For each layer, the spectral opacity $\tau_i(\nu)$ is first computed. Starting at the top of the stack (for downward or forward propagation), the initial spectral radiance is taken as

$$I_0(\nu) = B(\nu, T_0). \quad (1.35)$$

Eq. 1.29 or Eq. 1.30 is then applied sequentially through the layers until the spectral radiance emerging from the layer stack

$$I(\nu) = I_n(\nu) \quad (1.36)$$

has been computed. In addition, the accumulated total optical depth or opacity through the stack,

$$\tau(\nu) = \sum_i \tau_i(\nu) \quad (1.37)$$

is computed as well.

The default direction of propagation in *am* is downward from the top of layer 1 to the base

of layer n , which corresponds to the view towards the sky in plane-parallel hydrostatic models. A reverse propagation flag can be set so that $B(\nu, T_0)$ becomes the radiance incident on the base of layer n , and propagation is towards the top of layer 1. This gives the view towards the ground in hydrostatic models, with T_0 now being either the ground temperature or the temperature of some intermediate layer of the atmosphere which is assumed to be isothermal and optically thick across the band of interest. For reverse propagation through a layer, the functions $p(\tau)$ and $q(\tau)$ in Eq. 1.30 are exchanged. More complex path geometries are introduced in Chapter 9, including spherical limb paths, paths below the horizon, and paths to or from an interpolated observer or source level within the layer stack. For such paths, some or all of the model layers may be traversed twice, not at all, or partially if the layer is divided by an interpolated level.

1.2.4 Spectra derived from radiance and opacity

The spectral radiance $I(\nu)$ and opacity $\tau(\nu)$ are the two basic products of the radiative transfer and optical depth computations. From these, *am* can derive a number of other spectra and can apply various modifications to model the response of spectrometers and receivers.

From $I(\nu)$, an equivalent spectral brightness temperature $T_b(\nu)$ can be obtained by inverting Eq. 1.28 and substituting $I(\nu)$ for the Planck blackbody radiance $B(\nu, T)$:

$$T_b(\nu) = \frac{h\nu}{k \ln\left(1 + \frac{2h\nu^3}{c^2 I(\nu)}\right)}. \quad (1.38)$$

At a given frequency ν , $T_b(\nu)$ is the temperature at which a perfect blackbody radiating to free space would have spectral radiance $I(\nu)$. In *am* configuration files and output, $T_b(\nu)$ is referred to as the Planck (thermodynamic) brightness temperature, with symbol `tb`.

Alternatively, taking the limit of Eq. 1.28 when $h\nu \ll kT$, we obtain the Planck radiance in the classical Rayleigh-Jeans limit

$$B_{R-J}(\nu, T) = \frac{2kT\nu^2}{c^2}, \quad (1.39)$$

from which we can analogously define a spectral brightness temperature

$$\begin{aligned} T_{R-J}(\nu) &= \frac{c^2}{2k\nu^2} \cdot I(\nu) \\ &= T_b(\nu) \left[\frac{\frac{h\nu}{kT_b(\nu)}}{e^{h\nu/kT_b(\nu)} - 1} \right]. \end{aligned} \quad (1.40)$$

In *am* configuration files and output, $T_{R-J}(\nu)$ is referred to as the Rayleigh-Jeans brightness temperature, with symbol `trj`. $T_{R-J}(\nu)$ approaches $T_b(\nu)$ in the Rayleigh-Jeans limit. More generally, $kT_{R-J}(\nu)$ is the power per unit bandwidth at frequency ν incident on a detector coupled to a single-mode transmission line terminated by a matched load at physical temperature $T_b(\nu)$. The same is true if the transmission line is terminated by a matched antenna viewing a

scene in free space at thermodynamic temperature $T_b(\nu)$ that fills the antenna pattern. Since

$$kT_{R-J}(\nu) = \frac{c^2}{2\nu^2} I(\nu) = \frac{1}{2} \lambda^2 I(\nu), \quad (1.41)$$

it follows from thermodynamic arguments (Dicke 1946, Oliver 1965) that the antenna's integrated angular cross section, or area-solid angle product, is equal to λ^2 . The factor of $\frac{1}{2}$ accounts for the fact that $I(\nu)$ is the unpolarized radiance, whereas the antenna couples to a single polarization. In radio astronomy and microwave radiometry $T_{R-J}(\nu)$ as defined in Eq. 1.40 is often referred to as the *antenna temperature*. An excellent comprehensive treatment of the Rayleigh-Jeans approximation in the context of microwave radiometry and radiative transfer is Section 1.2 of Janssen (1993). A clearly-explained example of calibration of a practical instrument when $T_{R-J}(\nu)$ and $T_b(\nu)$ differ significantly is given in Bennett, et al. (1992), relating to the Differential Microwave Radiometer (DMR) on the *COBE* spacecraft.

The second of our two basic computed spectra is the opacity or optical depth $\tau(\nu)$. From $\tau(\nu)$, we can derive the transmittance $t(\nu)$ and excess delay $L(\nu)$. The transmittance is simply

$$t(\nu) = e^{-\tau(\nu)}. \quad (1.42)$$

The excess delay $L(\nu)$ is computed by *am* as the Hilbert transform of the opacity $\tau(\nu)$, with adjustments for above-band optical refractivity. Doing this accurately for a given spectrum requires that some attention be paid to the span and resolution of the frequency grid over which the model is computed. Details are given in Chapter 6.

To model the response of spectrometers, the transmittance and all spectra derived from the radiance can be convolved with various instrumental line shapes. For heterodyne receivers, *am* can compute IF spectra and other spectra including system temperature and spectral Y-factor. Details are given in Chapter 5.

References

- D. G. Andrews 2000, *An Introduction to Atmospheric Physics*. Cambridge University Press.
- C. L. Bennett, G. F. Smoot, M. Janssen, S. Gulkis, A. Kogut, G. Hinshaw, C. Backus, M. G. Hauser, J. C. Mather, L. Rokke, L. Tenorio, R. Weiss, D. T. Wilkinson, E. L. Wright, G. De Amici, N. W. Boggess, E. S. Cheng, P. D. Jackson, P. Keegstra, T. Kelsall, R. Kummerer, C. Lineweaver, S. H. Moseley, T. L. Murdock, J. Santana, R. A. Shafer, and R. F. Silverberg 1992, "COBE Differential Microwave Radiometers: Calibration Techniques." Ap. J. 391:466.
- S. A. Clough, M. J. Iacono, and J. Moncet 1992, "Line-by-Line Calculations of Atmospheric Fluxes and Cooling Rates: Application to Water Vapor." J. Geophys. Res. 97:15761.
- R. H. Dicke 1946, "The Measurement of Thermal Radiation at Microwave Frequencies." Rev. Sci. Inst. 17:268.

M. A. Janssen, ed. 1993, *Atmospheric Remote Sensing by Microwave Radiometry*. Wiley.

B. M. Oliver 1965, "Thermal and Quantum Noise." Proc. IEEE 5:436.

Chapter 2: Using *am*

This chapter covers the basic usage of *am*. In operation, *am* always begins by processing a *configuration file*, which defines the model atmosphere or other propagation medium, the path geometry, the desired outputs, and other variables that control the computation. We begin with a brief introduction to the *am* command line interface. After that, the structure of *am* configuration files is illustrated through a series of examples, starting out simply and building in complexity.

2.1 Getting started

2.1.1 Testing the installation

Installation of *am* is described in Appendix A. Once the program has been installed, typing **am** at a shell prompt gives information about the program version and usage (in this example and all those following, user input is shown in **bold**, and everything else is output that appears on the console):

```
$ am
am version 14.0
build date Sep 26 2024 13:20:00
64-bit multi-threaded (OpenMP version 201511)

Usage: am [option] FILE [param1 param2...]

FILE = model configuration file. Use "stdin" or "-" for the file name
      to take input from the console or from a pipe.
[param1 param2...] = optional list of parameters. All instances of
      %1 in FILE will be replaced by param1, %2 with param2, etc.

Options:
-a --atmosphere Atmospheric model only, no radiative transfer.
-b --benchmark Time execution of FILE, without writing output.
-e --environment Display environment information.
-h --help Display this help message.
-l --legal Display terms of use.
-r --references Display a list of references.
```

```
-v --version      Display version and build information.

For updates, documentation, and citation information, see

  https://doi.org/10.5281/zenodo.640645

$
```

Typing `am -e` will retrieve information about the current environment. This is especially useful after installing *am*, to verify that the disk cache is set up correctly and to examine the OpenMP environment for the multi-threaded version of the program.

```
$ am -e
OpenMP
  Available processors = 10
  OMP_NUM_THREADS = 8
  OMP_NESTED = false
  OMP_MAX_ACTIVE_LEVELS = 1
Disk cache
  AM_CACHE_PATH = /Users/spaine/.am/
  AM_CACHE_HASH_MODULUS = 1021 (default setting)
kcache
  AM_KCACHE_MEM_LIMIT = 0 (use maximum available memory)
Fit I/O
  AM_FIT_INPUT_PATH = current directory (default setting)
  AM_FIT_OUTPUT_PATH = current directory (default setting)

$
```

The use of `OMP_NESTED` and `OMP_MAX_ACTIVE_LEVELS` varies according to OpenMP version. Refer to Appendix A for details.

2.1.2 Using the program

Every computation with *am* is defined by a configuration file. An *am* configuration file, described in the next section, contains a description of the model atmosphere or other propagation medium, the path geometry, and other parameters that control the model computation or fit. The configuration file can be passed to *am* by name at the command line, as in

```
$ am model.amc
```

or taken from the standard input (`stdin`):

```
$ am - < model.amc
```

The latter is useful if the configuration file is generated by some other process. For example, if `make_model.sh` is a script which generates an *am* configuration file and writes it to `stdout`, a command to run the script and process the resulting configuration data with *am* would be:

```
$ make_model.sh | am -
```

A configuration file can contain replaceable parameters, labeled `%1`, `%2`, etc., which are replaced at run time with corresponding strings from the *am* command line. Examples of configuration files with replaceable parameters will be given later in this chapter.

When computing model spectra, *am* writes to two output streams. The output spectra are written to the standard output stream (`stdout`). The model configuration data are written back to the standard error stream (`stderr`), in the form of a standard *am* configuration file. This output configuration file serves to document the run, and can be used to reproduce it later if needed. In the output configuration file, comments from the original configuration file are discarded, and new diagnostic comments are added. These diagnostic comments provide run time performance data, mark any lines that triggered warnings, and show the value of certain parameters derived from the configuration data. Examining the output configuration data is a good way to be sure that the configuration file was interpreted as intended. The configuration data are always written (and flushed) before the output spectra are written, so that they can be prepended as a header to the output spectra by redirecting `stderr`. The command

```
$ am model.amc > model.out 2>&1
```

would accomplish this in the common Unix or GNU/Linux shells and in the Windows `cmd.exe` shell.

As *am* runs, it keeps a log of errors and warnings. The error log is written to `stderr` after all other output, which helps make it more likely to be noticed on the console. However, if `stderr` has been redirected as in the preceding example, then the error log will be likewise redirected to the output stream. In this case it is still possible to check for errors by testing the value returned by *am* to the shell, which is 1 if there were errors or warnings, or 0 otherwise. For example,

```
$ am model.amc >model.out 2>&1 && echo OK || echo ERROR
```

is one way to do this, which works in all of the shells just mentioned.

2.1.3 Windows-specific issues

Compared with Unix-like systems, choosing a command shell or environment for running text-based tools such as *am* on Windows involves extra considerations. In a standard Windows setup, users may find that using `cmd.exe` is simpler than using Windows PowerShell. This is because PowerShell, by default, writes text output using UTF-16 encoding with a prepended byte-order mark (BOM). The resulting output files are twice the size of those produced on systems that use UTF-8 or single-byte ASCII-compatible encodings, and additionally the byte-order mark can confuse other tools such as plotting utilities. In PowerShell, the character encoding can be modified using a construct like

```
PS> am model.amc | out-file -Encoding utf8 model.out
```

which will produce an output file with UTF-8 encoding, but this workaround makes it impossible to combine the `stdout` and `stderr` streams as discussed above. It is also possible to set a PowerShell preference variable to change the default encoding, but this carries the risk of breaking other scripts that expect UTF-16. An alternative is to install PowerShell Core, an open-

source cross-platform version of PowerShell which defaults to UTF-8 with no BOM.

Another option, with the added advantage of affording ready access to all the standard Unix tools, is to install a Windows-hosted Unix-like environment. Microsoft's native solution since Windows 10 is the Windows Subsystem for Linux (WSL), which makes it easy to run popular Linux distributions within Windows. Other popular options available for any version of Windows are MinGW-w64 and Cygwin.

2.2 Configuration file examples

Configuration files are the key to doing anything with *am*. The structure of *am* configuration files is best illustrated by example, so the rest of this chapter presents successive examples of increasing complexity. The discussion here focuses on the essentials—more detailed information can be found in Appendix B.

2.2.1 A laboratory propagation path

Imagine a laboratory setup in which a spectrometer views a 77 K liquid nitrogen cold load, through 1 meter of air, at 1 atm ambient pressure and 20 C ambient temperature. Suppose we want to compute the transmittance of the path, and the effective brightness temperature seen by the spectrometer, over the frequency range from 0 GHz–2500 GHz. Water vapor will be the only significant source of absorption—for this first example, we will assume a fixed water vapor volume mixing ratio of 0.012, which corresponds to 50% relative humidity at this temperature and pressure.

A simple configuration file to model this situation defines the frequency range and resolution, the background temperature, and a single layer containing a single absorbing species, H₂O. The complete text of the file is:

```
# Example 2.1 - a laboratory propagation path
f 0 GHz 2500 GHz 50 MHz      # frequency grid for computation
T0 77 K                         # background temperature

layer                           # introduces a layer
P 1013.25 mbar                 # layer pressure
T 20 C                          # layer temperature
h 1 m                           # layer thickness
column h2o vmr 0.012           # adds a column of h2o to this layer
```

Units must accompany all dimensioned quantities. (A complete table of units recognized by *am* appears in Appendix B.) Comments have been added using the '#' character, which tells *am* to ignore all subsequent text to the end of the line. Assuming the above text has been put into a file called `example2.1.amc`, a session to run the model proceeds as follows:

```
$ am example2.1.amc >example2.1.out
# am version 14.0 (build date Sep 26 2024 13:20:00)

# run time 0.009 s
# dcache hit: 0 miss: 1 discard: 0
```

```

f 0 GHz 2500 GHz 50 MHz
output text f GHz tx none Tb K

tol 0
selfbroad_vmr_tol 0.003

geometry plane-parallel
refract none
za 0 deg
PTmode midpoint

T0 77 K

layer
P 1013.25 mbar
T 20 C
h 1 m
lineshape VVH_750 h2o_lines
column h2o vmr 0.012 (3.00417e+19 cm^-2)

#
# column densities [cm^-2], including scale factors
#
# zenith           line-of-sight
# total (1 layer):
#          h2o 3.004172e+19 3.004172e+19
#                  (8.98726 um_pwv) (8.98726 um_pwv)
#
$

#

```

The command shown above redirects the output spectra from `stdout` to a file called `example2.1.out`. The program output to `stderr`, which appears on the console, gives additional data about default model configuration parameters not explicitly supplied in the configuration file. If `stderr` had been redirected to a file, that file would have constituted a valid *am* configuration file, equivalent to a more verbose version of `example2.1.amc`. For example, in the output configuration data, the H₂O column density computed from the volume mixing ratio was printed in parentheses. Like '#', '(' is a comment character, so these numerical annotations do not interfere with later re-reading of the configuration data by *am*. A complete list of comment characters is given in Appendix B.

The original file, `example2.1.amc`, did not contain a line defining the desired output spectra. The output configuration data shows that it defaulted to

```
output text f GHz tx none Tb K
```

which meant that the output file would contain three space-delimited columns of text containing frequency in GHz, transmittance, and Planck brightness temperature in Kelvin. If we had wanted different output spectra, different output units, or a different output format, an explicit `output` statement could have been added to the original configuration file. The contents of the output file, `example2.1.out`, are:

```
0.000000e+00 1.000000e+00 7.700000e+01
5.000000e-02 1.000000e+00 7.700000e+01
1.000000e-01 1.000000e+00 7.700000e+01
```

```

.
.
.
2.4999000e+03 7.073390e-01 1.438042e+02
2.4999500e+03 7.076346e-01 1.437401e+02
2.5000000e+03 7.079290e-01 1.436762e+02

```

These output spectra are plotted in Figure 2.1. Alternatives to this default space-delimited `text` format are `csv` for comma-delimited text and `npy` for binary output containing NumPy array data. See Appendix B for details.

In this example the frequency grid ran from 0 GHz to 2500 GHz by steps of 50 MHz, as might have been expected from the line

```
f 0 GHz 2500 GHz 50 MHz
```

at the beginning of the configuration file. However, there is a subtlety regarding the specification of frequency grids in *am*: the frequency grid is always aligned to the origin. That is, the grid frequencies are always integer multiples of the grid interval, even if the grid does not start at zero. Thus, if a frequency grid is specified by the statement

```
f 100 GHz 200 GHz 3 GHz
```

then the first grid frequency will be 102 GHz, and the last will be 198 GHz. This behavior is by design—it optimizes reuse of cached absorption coefficients, and facilitates computation of the Hilbert transform used to compute delay spectra.

2.2.2 Adding a command-line parameter

In the preceding example, most likely we knew the relative humidity beforehand, and had to hand-calculate the H₂O volume mixing ratio after looking up the saturation vapor pressure at the laboratory temperature. Then, we had to edit the configuration file to change the mixing ratio. We can make two changes that make the file much more efficient to use. First, we can specify the relative humidity directly, using the `RH` keyword. Second, we can make the humidity value a command-line parameter. The modified file appears as follows:

```

# Example 2.2 - laboratory propagation path, user supplies RH
?
? Usage: am example2.2.amc [RH%]
?
f 0 GHz 2500 GHz 50 MHz
T0 77 K

layer
P 1013.25 mbar
T 20 C
h 1 m
column h2o RH %1

```

The token `%1` is a reference to the first command-line parameter supplied by the user. Command line parameters are numbered in order, starting with `%1`, after the configuration file name. To run this file for a relative humidity of 60%, we would use the command:

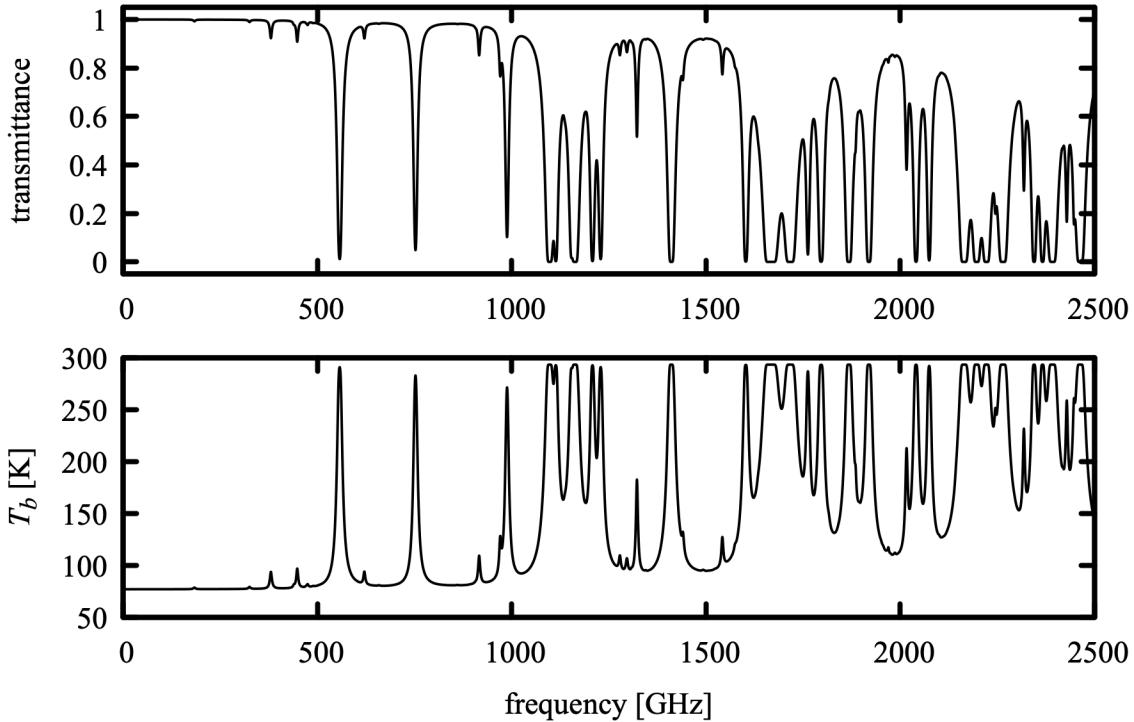


Figure 2.1 Plot of output spectra for Example 2.1, transmittance and Planck brightness temperature for a laboratory propagation path 1 m long at 50% RH.

```
$ am example2.2.amc 60% >example2.2.out
```

Note that humidity values in *am* are given as a percentage. The percent sign, with no preceding space, is required. Be aware that some shells may try to interpret characters such as * or % that appear in command-line arguments. The workaround is to escape the offending character or enclose the argument in quotation marks.

If the user fails to supply a required command line parameter, an error message is written to `stderr`, indicating the line on which the error occurred. In addition, using the special comment character '?' as shown in Example 2.2, a usage message can be embedded in the configuration file. The usage message appears after the parser errors triggered by the missing parameter, and before the error log summary:

```
$ am example2.2.amc >example2.2.out
example2.2.amc(12) :
  The replaceable parameter %1 has no matching command-line
  parameter.
example2.2.amc(12) :
  The file was not checked beyond this line.

Usage: am example2.2.amc [RH%]
```

```
! Error: parse error. Count: 2
$
```

Command line parameters are text strings, so they may be used as a substitute for any text string in a configuration file. There is no limit on the number of command line parameters, and a given parameter reference may be used as often as needed (or not at all) in the configuration file.

2.2.3 Adding another column to the layer

In the preceding two examples, H₂O was the only absorbing species included in the model laboratory path. This was realistic—over such a short distance the absorption from other atmospheric constituents is relatively small. Nevertheless, we can go ahead and add a second column, `dry_air`, to the model. The configuration file then becomes:

```
# Example 2.3 - path including dry air, at default mixing ratio
?
? Usage: am example2.3.amc [RH%]
?
f 0 GHz 2500 GHz 50 MHz
T0 77 K

layer
P 1013.25 mbar
T 20 C
h 1 m
column dry_air vmr
column h2o RH %1
```

As its name suggests, `dry_air` is a composite column type that includes the major constituents of dry air, in proportion to their terrestrial abundances. (See Appendix B for descriptions of the `dry_air` and alternative `dry_air_std` column types.) In this example, we omitted a value for the mixing ratio after the `vmr` keyword for the `dry_air` column. Doing so signals `am` to check for a default value (in this case 1.0), and then to scale it to account for all explicitly-defined mixing ratios on the layer. The annotated configuration data in the `stderr` output written back to the console shows how this works:

```
$ am example2.3.amc 60% >example2.3.out
# am version 14.0 (build date Sep 26 2024 13:20:00)

# run time 0.164 s
# dcache hit: 1 miss: 8 discard: 0

f 0 GHz 2500 GHz 50 MHz
output text f GHz tx none Tb K

tol 0
selfbroad_vmr_tol 0.003

geometry plane-parallel
refract none
za 0 deg
PTmode midpoint
```

```

T0 77 K

layer
P 1013.25 mbar
T 20 C
h 1 m
lineshape Gross ch4 co co2 n2o o2_uncoupled
lineshape VVW_coupled o2_coupled
lineshape VVH_750 h2o_lines
column dry_air vmr (0.986147) (2.4688e+21 cm^-2)
column h2o RH 60% (vmr 0.01385) (3.46803e+19 cm^-2)

#
# column densities [cm^-2], including scale factors
#
#          zenith           line-of-sight
# total (1 layer):
#          dry_air 2.468797e+21      2.468797e+21
#          h2o     3.468027e+19      3.468027e+19
#                  (10.3749 um_pwv) (10.3749 um_pwv)
#
$
```

Here, the H₂O mixing ratio computed from relative humidity, temperature, and pressure is shown in parenthesis. The total dry air mixing ratio was automatically adjusted to the value shown to normalize the sum of all mixing ratios on the layer. In the following section, we'll take advantage of this automatic adjustment of the default dry air mixing ratio in an atmospheric model which implements an adjustable water vapor profile across multiple layers.

2.2.4 A simple atmospheric model

The next example is a simple model of the terrestrial lower atmosphere, divided into five hydrostatic layers at 200 mbar intervals, with the surface pressure assumed to be 1000 mbar. The temperature at each layer boundary will be taken from the 1976 U.S. Standard Atmosphere (NOAA 1976). The uppermost layer will have a midpoint pressure of 100 mbar; consequently, we're only modeling the troposphere. Ozone will be omitted, and we'll model the water vapor profile by assuming constant relative humidity, with the humidity value supplied at the command line. In addition, we'll provide for a supercooled non-precipitating cloud layer between 800 mbar and 600 mbar, modeled using the lwp_abs_Rayleigh column type, with the total cloud liquid water path (LWP) supplied as a second command-line parameter. The output of the model will be the zenith millimeter-wave opacity spectrum.

The complete model configuration file is listed below. The model output is controlled by the line

```
output f GHz tau neper
```

meaning that the output will be two columns: frequency (in GHz), and opacity (in nepers, using the radiometric convention of nepers as the natural logarithm of a power ratio; see Table B.10 in Appendix B.). Since the opacity is the only requested output, the line

```
T0 2.7 K
```

is not really necessary. Even so, it is a wise precaution in case the `output` statement is later modified to include emission spectra, since if `TO` is left unspecified it will default to 0 K without any warning to the user. Notice that the units for the user-specified cloud LWP are entered as a third command-line parameter. This choice is a matter of style—we could have hard-coded the units into the configuration file instead. Also note that we've used `RHi` for relative humidity with respect to the saturated vapor pressure over ice in the upper two layers of the model, where the layer midpoint temperatures are below the spontaneous ice nucleation point:

```
# Example 2.4 - Simple atmospheric model with constant RH and cloud
#
#           T(P) from US standard atmosphere
?
? Usage: am example2.4.amc  RH%  cloud_LWP
? Example: am example2.4.amc  75%  200 g*m^-2
?

f 0 GHz 350 GHz 25 MHz
output f GHz tau neper

TO 2.7 K

layer # empty layer for top of atmosphere
Pbase 0 mbar
Tbase 217 K

layer
Pbase 200 mbar # z = 11.8 km
Tbase 217 K
column dry_air hydrostatic
column h2o RHi %1

layer
Pbase 400 mbar # z = 7.2 km
Tbase 241 K
column dry_air hydrostatic
column h2o RHi %1

layer
Pbase 600 mbar # z = 4.2 km
Tbase 261 K
column dry_air hydrostatic
column h2o RH %1

layer
Pbase 800 mbar # z = 1.9 km
Tbase 275 K
column lwp_abs_Rayleigh %2 %3 # cloud layer
column dry_air hydrostatic
column h2o RH %1

layer
Pbase 1000 mbar # z = 0.1 km
Tbase 288 K
column dry_air hydrostatic
column h2o RH %1
```

In Figure 2.2, the output of Example 2.4 is plotted for three cases. First, to compute the contribution of the dry atmosphere to the opacity, we set the humidity and cloud LWP to zero by running the model with the command

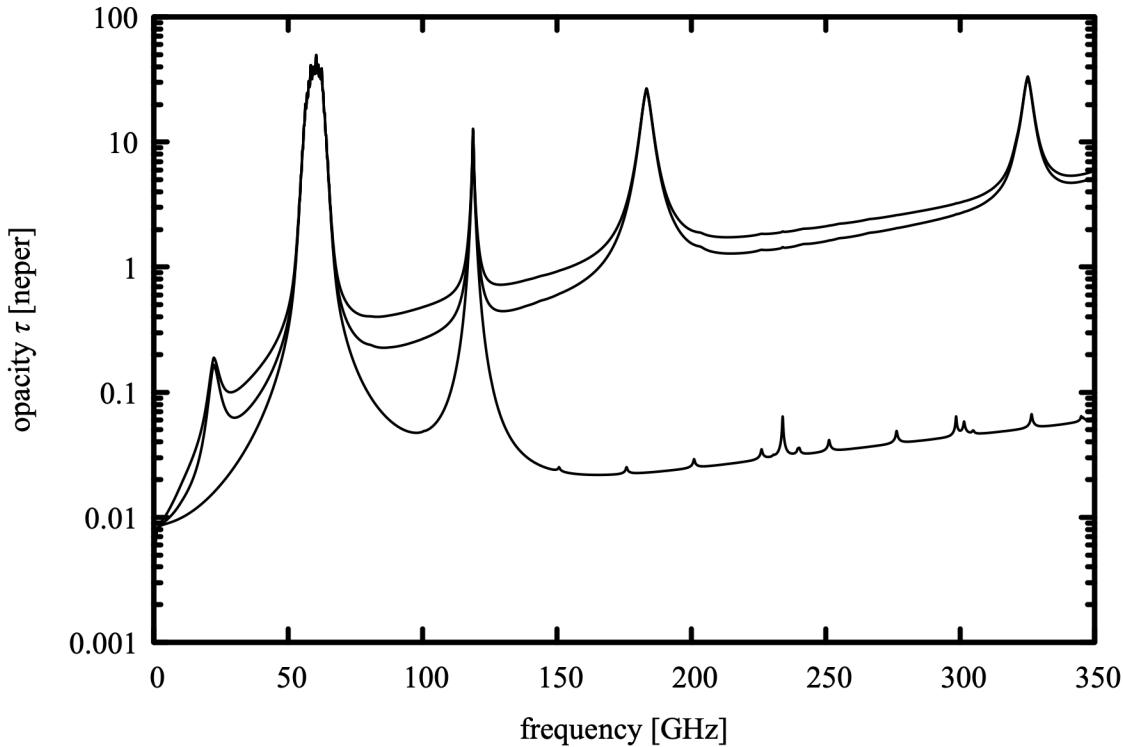


Figure 2.2 Output from example 2.4 for three cases. Bottom trace: dry air only; middle: 75% constant RH; top: 75% constant RH plus 200 g m⁻² cloud liquid water path between 800 mbar and 600 mbar (top).

```
$ am example2.4.amc 0% 0 g*m^-2 >example2.4_0_0.out
```

The output is plotted as the lowest trace in Figure 2.2. The main features are the O₂ spin-rotation band consisting of an overlapping cluster of lines near 60 GHz and an isolated line at 119 GHz. The weaker features are rotational lines of N₂O (spaced at 25 GHz intervals), O¹⁸O, and O¹⁷O. Next, we'll run the model with water vapor added at 75% constant relative humidity:

```
$ am example2.4.amc 75% 0 g*m^-2 >example2.4_75_0.out
```

The resulting output is the middle trace in Figure 2.2—the main new features are the H₂O rotational lines at 22 GHz, 183 GHz, and 325 GHz, with H₂O line wing and continuum absorption in between. If we were to examine the `stderr` output from running this command, we'd find that the water vapor adds up to about 20 precipitable millimeters, a typical mid-latitude total column density above sea level.

Finally, we can model the effect of adding non-precipitating cloud between 800 mbar and 600 mbar, having 200 g m⁻² total liquid water path, by running the command:

```
$ am example2.4.amc 75% 200 g*m^-2 >example2.4_75_200.out
```

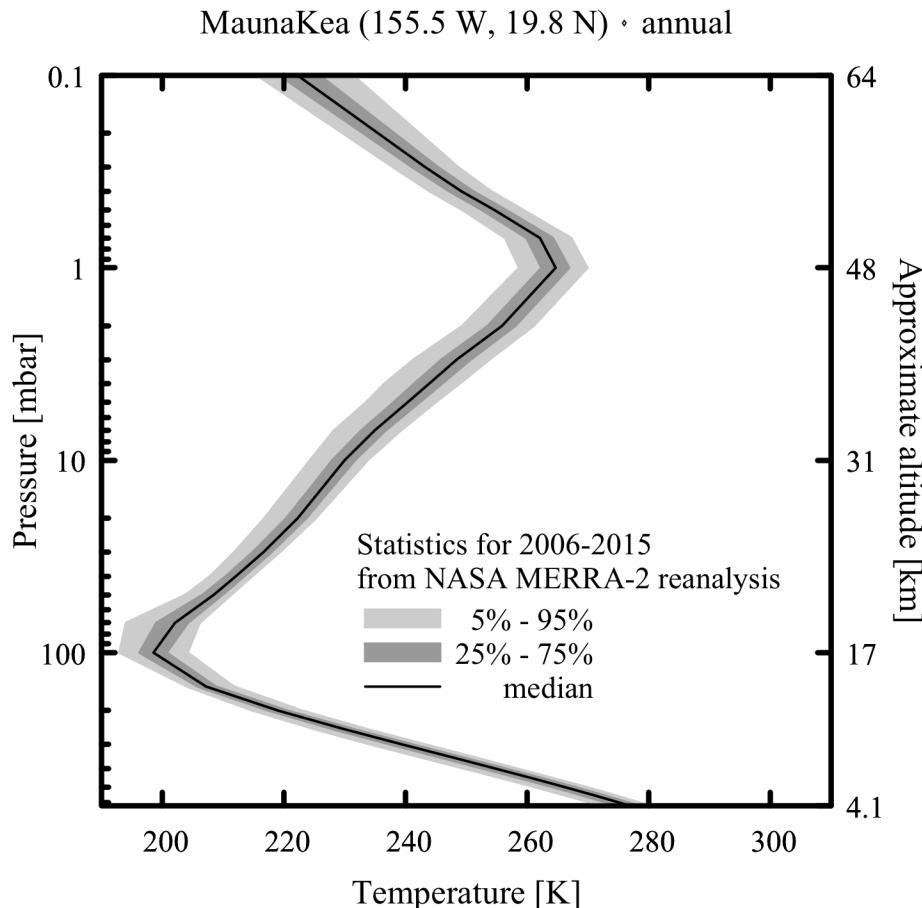


Figure 2.3 Year-round temperature quantile profiles over Mauna Kea, from NASA MERRA-2 reanalysis data.

The result is the top trace in Figure 2.2. This adds additional continuous absorption, increasing with frequency, which makes a significant contribution in the windows between the lines.

Though very simple, this model already serves to illustrate the spectral features which can be exploited to remotely sense water vapor and clouds by ground-based millimeter wave radiometry. Good introductions are the reviews by Cadeddu et al. (2013) and Westwater et al. (2005).

2.2.5 A realistic model for Mauna Kea

The final example in this chapter is a model configuration file for computing submillimeter opacity and Planck brightness temperature spectra above the SMA observatory on Mauna Kea, Hawaii. It incorporates realistic vertical profiles for temperature, water vapor, and ozone from the observatory pressure altitude of 625 mbar into the lower mesosphere at 0.1 mbar.

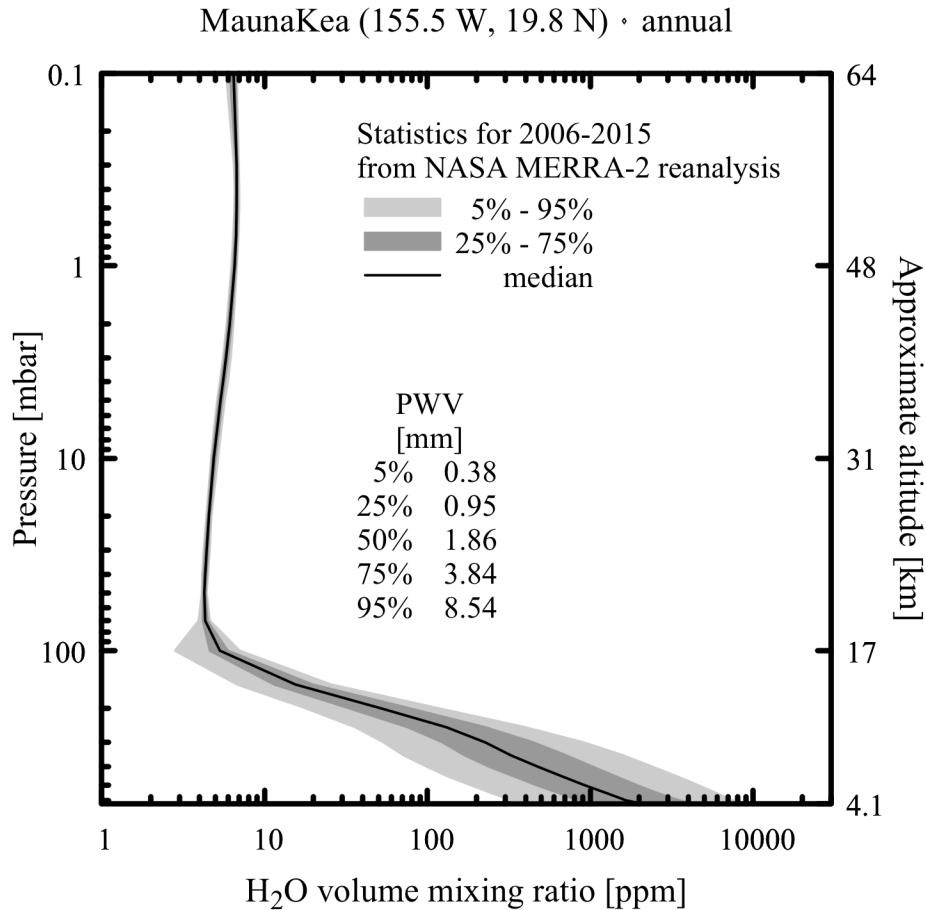


Figure 2.4 Year-round H_2O mixing ratio quantile profiles over Mauna Kea, from NASA MERRA-2 reanalysis data. In the stratosphere, the mixing ratio is limited by the “cold trap” at the tropopause. The increase with altitude above the tropopause is due to photochemically-driven oxidation of CH_4 (Le Texier, Solomon, and Garcia 1988).

Consequently, the layer description here is more detailed than in the simple examples presented so far.

Our model will be based on NASA MERRA-2 reanalysis data (Rienecker et al. 2011, Molod et al. 2015). Here, *reanalysis* means a retrospective run of a general circulation model assimilating archival satellite, surface, radiosonde, and other observations to produce a best-guess history of the past atmospheric state. MERRA-2 data are available re-gridded onto a $\frac{1}{2}^\circ$ latitude $\times \frac{5}{8}^\circ$ longitude $\times 42$ pressure level grid with a temporal resolution of 3 hours. To produce the vertical profiles for our model, MERRA-2 data for the ten year period 2006–2015 were first interpolated to the observatory latitude and longitude. Quantile statistics of the temperature, H_2O

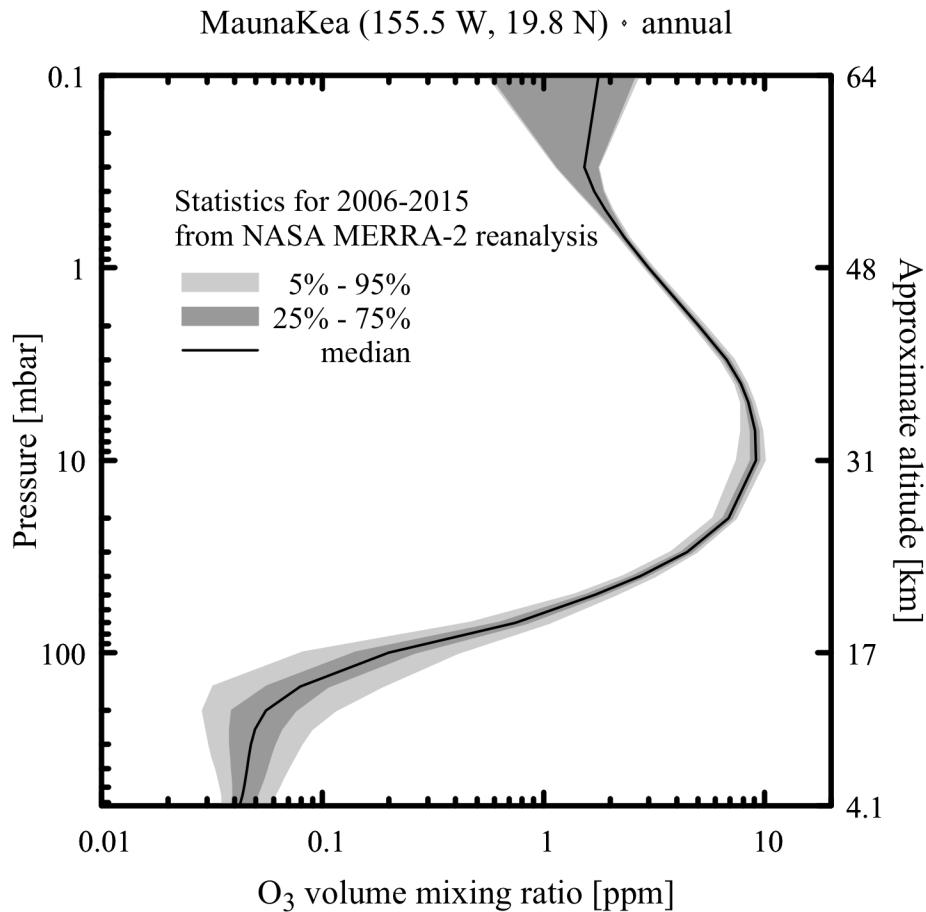


Figure 2.5 Year-round O₃ mixing ratio quantile profiles over Mauna Kea, from NASA MERRA-2 reanalysis data.

mixing ratio, and O₃ mixing ratio were then computed over the entire period, including all times of day, for each MERRA pressure level. The results are plotted in Figures 2.3 through 2.5.

For our model, we will use the median profiles, with layer boundaries coinciding with the MERRA pressure levels. When constructing an *am* model configuration file from this kind of profile data, layer base temperatures are taken directly from the temperature profile at the layer boundary pressure, but the mixing ratios of the various species should approximate their averages weighted by column density across the layer. Thus, in the model file below the layer base temperatures are the MERRA median temperatures at the corresponding pressure levels, whereas the H₂O and O₃ mixing ratios are the average of their values at the upper and lower boundary pressures of the layer.

Our model uses default pressure-broadened line shapes from ground level through the middle stratosphere. We intend to use this model up to 1000 GHz, and at this frequency Doppler

broadening becomes significant relative to pressure broadening near the stratopause. Consequently, in the upper stratosphere and mesosphere, the model uses a Voigt line shape for all species. The tradeoff being made involves both computational efficiency and appropriate approximation. The Voigt line shape is slow to compute, but accounts for Doppler broadening, whereas the default line shapes neglect Doppler broadening but do account for effects which are important at higher pressure, such as line mixing and the water vapor continuum. Chapter 3 has a detailed discussion of line shapes used in *am*.

The configuration file for our model takes several command-line parameters to set the range and resolution of the frequency grid, the zenith angle of observation, the surface temperature, and a scale factor applied to the tropospheric water vapor profile. This scale factor is used to model variations in the total precipitable water vapor over the observatory site in a manner which approximates the natural variation of the H₂O mixing ratio profile—water vapor and its fluctuations are largely confined below the “cold trap” at the tropopause as shown in Figure 2.4. The scale factor is implemented using an `Nscale` statement. A statement like

```
Nscale troposphere h2o 0.5
```

applies a scale factor of 0.5 to the `h2o` column density on any layer carrying the optional tag string `troposphere`; see Appendix B for details. To keep things simple, we haven't included a similar scale factor or factors for the median ozone profile. This is sufficient for applications such as planning astronomical observations—in contrast with the total water vapor column, which can range over more than an order of magnitude, the total ozone column density over Mauna Kea typically stays within about 20% of its median value throughout the year.

Having the ground temperature as a user parameter makes it possible to ensure that the spectral baseline saturates to the correct brightness temperature in the core of the strongest water lines. More information about the remaining command-line parameters is given in the usage text embedded in the configuration file, listed below in its entirety:

```
# Example 2.5 - A model for Mauna Kea based on median profiles computed from
# NASA MERRA-2 reanalysis data for the period 2006-2015.
#
# MERRA references:
#   M. M. Rienecker et al. (2011), MERRA: NASA's Modern-Era Retrospective
#   Analysis for Research and Applications. J. Climate 24:3624.
#
#   A. Molod et al (2015), Development of the GEOS-5 atmospheric general
#   circulation model: evolution from MERRA to MERRA2. Geoscience Model
#   Development 8:1339.
#
# MERRA is produced by the NASA/Goddard Global Modeling and Assimilation
# Office (GMAO). MERRA-2 data are archived and distributed by the Goddard
# Earth Sciences Data and Information Services Center (GES DISC).
#
# Data reference:
#   Global Modeling and Assimilation Office (GMAO) (2015),
#   MERRA-2 inst3_3d_asm_Np: 3d,3-Hourly,Instantaneous,Pressure-Level,
#   Assimilation,Assimilated Meteorological Fields,version 5.12.4,
#   Greenbelt, MD, USA:Godard Space Flight Center Distributed Active
#   Archive Center (GSFC DAAC),
#   Accessed 2016 Feb 10 at doi:10.5067/QBZ6MG944HW0
```

```

#
?
? Usage: am example2.5.amc fmin fmax df zenith_angle Tground trop_h2o_scale
?
? Example: am example2.5.amc 170 GHz 390 GHz 2 MHz 0 deg 277 K 1.0
?
? Tground is the surface temperature. At the summit of Mauna Kea, this is
? usually close to the median value of 277 K for the free atmosphere at the
? same altitude.
?
? trop_h2o_scale is a scaling factor on the median tropospheric water vapor
? profile. Spectra for other percentile profiles can be approximated by
? scaling the median profile to obtain the same total precipitable water
? vapor (pwv) column as follows:
?
?   H2O profile percentile |    5    |   25    |   50    |   75    |   95
?   trop_h2o_scale | 0.206 | 0.513 | 1.00 | 2.07 | 4.60
?   pwv [mm] | 0.384 | 0.952 | 1.86 | 3.84 | 8.54
? 225 GHz zenith opacity | 0.029 | 0.051 | 0.087 | 0.170 | 0.367
?
? Setting df larger than about 100 kHz may result in warnings of unresolved
? lines on the lowest-pressure levels in this model. These warnings may
? be ignored if detailed resolution of line cores is not needed.
?

f %1 %2 %3 %4 %5 %6
output f GHz tau Tb K
za %7 %8
tol 1e-4

Nscale troposphere h2o %11

T0 2.7 K

layer mesosphere
Pbase 0.1 mbar
Tbase 222.3 K
lineshape Voigt-Kielkopf
column dry_air vmr
column h2o vmr 6.46e-06
column o3 vmr 1.77e-06

layer mesosphere
Pbase 0.3 mbar
Tbase 243.3 K
lineshape Voigt-Kielkopf
column dry_air vmr
column h2o vmr 6.60e-06
column o3 vmr 1.65e-06

layer mesosphere
Pbase 0.4 mbar
Tbase 249.2 K
lineshape Voigt-Kielkopf
column dry_air vmr
column h2o vmr 6.75e-06
column o3 vmr 1.61e-06

layer mesosphere
Pbase 0.5 mbar
Tbase 254.6 K
lineshape Voigt-Kielkopf
column dry_air vmr
column h2o vmr 6.75e-06

```

```
column o3 vmr 1.80e-06

layer mesosphere
Pbase 0.7 mbar
Tbase 262.1 K
lineshape Voigt-Kielkopf
column dry_air vmr
column h2o vmr 6.72e-06
column o3 vmr 2.12e-06

layer stratosphere
Pbase 1 mbar
Tbase 264.7 K
lineshape Voigt-Kielkopf
column dry_air vmr
column h2o vmr 6.63e-06
column o3 vmr 2.66e-06

layer stratosphere
Pbase 2 mbar
Tbase 255.9 K
column dry_air vmr
column h2o vmr 6.34e-06
column o3 vmr 4.01e-06

layer stratosphere
Pbase 3 mbar
Tbase 248.4 K
column dry_air vmr
column h2o vmr 5.97e-06
column o3 vmr 5.88e-06

layer stratosphere
Pbase 4 mbar
Tbase 243.9 K
column dry_air vmr
column h2o vmr 5.69e-06
column o3 vmr 7.28e-06

layer stratosphere
Pbase 5 mbar
Tbase 240.3 K
column dry_air vmr
column h2o vmr 5.47e-06
column o3 vmr 8.13e-06

layer stratosphere
Pbase 7 mbar
Tbase 234.9 K
column dry_air vmr
column h2o vmr 5.25e-06
column o3 vmr 8.75e-06

layer stratosphere
Pbase 10 mbar
Tbase 230.0 K
column dry_air vmr
column h2o vmr 5.01e-06
column o3 vmr 9.10e-06

layer stratosphere
Pbase 20 mbar
Tbase 222.3 K
column dry_air vmr
```

```
column h2o vmr 4.72e-06
column o3 vmr 8.01e-06

layer stratosphere
Pbase 30 mbar
Tbase 216.6 K
column dry_air vmr
column h2o vmr 4.49e-06
column o3 vmr 5.67e-06

layer stratosphere
Pbase 40 mbar
Tbase 212.1 K
column dry_air vmr
column h2o vmr 4.37e-06
column o3 vmr 3.60e-06

layer stratosphere
Pbase 50 mbar
Tbase 208.5 K
column dry_air vmr
column h2o vmr 4.30e-06
column o3 vmr 2.23e-06

layer stratosphere
Pbase 70 mbar
Tbase 202.1 K
column dry_air vmr
column h2o vmr 4.30e-06
column o3 vmr 1.23e-06

layer troposphere
Pbase 100 mbar
Tbase 198.5 K
column dry_air vmr
column h2o vmr 4.83e-06
column o3 vmr 4.72e-07

layer troposphere
Pbase 150 mbar
Tbase 207.2 K
column dry_air vmr
column h2o vmr 1.04e-05
column o3 vmr 1.39e-07

layer troposphere
Pbase 200 mbar
Tbase 219.0 K
column dry_air vmr
column h2o vmr 3.40e-05
column o3 vmr 6.74e-08

layer troposphere
Pbase 250 mbar
Tbase 229.9 K
column dry_air vmr
column h2o vmr 9.15e-05
column o3 vmr 5.25e-08

layer troposphere
Pbase 300 mbar
Tbase 239.5 K
column dry_air vmr
column h2o vmr 1.79e-04
```

```

column o3 vmr 4.85e-08

layer troposphere
Pbase 350 mbar
Tbase 247.7 K
column dry_air vmr
column h2o vmr 2.76e-04
column o3 vmr 4.69e-08

layer troposphere
Pbase 400 mbar
Tbase 254.7 K
column dry_air vmr
column h2o vmr 3.99e-04
column o3 vmr 4.60e-08

layer troposphere
Pbase 450 mbar
Tbase 260.9 K
column dry_air vmr
column h2o vmr 5.71e-04
column o3 vmr 4.52e-08

layer troposphere
Pbase 500 mbar
Tbase 266.2 K
column dry_air vmr
column h2o vmr 7.94e-04
column o3 vmr 4.45e-08

layer troposphere
Pbase 550 mbar
Tbase 270.9 K
column dry_air vmr
column h2o vmr 1.09e-03
column o3 vmr 4.38e-08

layer troposphere
Pbase 600 mbar
Tbase 275.0 K
column dry_air vmr
column h2o vmr 1.46e-03
column o3 vmr 4.30e-08

layer troposphere
Pbase 625 mbar
Tbase %9 %10
column dry_air vmr
column h2o vmr 1.87e-03
column o3 vmr 4.23e-08

```

As an example of putting this file to use, we can compute the zenith opacity and Planck brightness temperature spectra for 1 mm pwv and a ground temperature of 277 K, for a set of frequency intervals of interest for millimeter and submillimeter astronomy. To get 1 mm pwv, the required scale factor on the median tropospheric column density is 0.5385. This can be found by interpolating between the first quartile and median pwv values in the table included in the file's usage text. To keep the compute time reasonably short for these broad band spectra, the grid resolution will be set to 10 MHz. This will cause *am* to generate a warning about unresolved lines in several of the lowest-pressure layers, but this can be safely ignored if we not interested in

the details of the narrow line cores. The commands are:

```
$ am example2.5.amc 0 GHz 200 GHz 10 MHz 0 deg 277 K 0.5385 \
>example2.5.0-200.out 2>example2.5.0-200.err
$ am example2.5.amc 170 GHz 390 GHz 10 MHz 0 deg 277 K 0.5385 \
>example2.5.170-390.out 2>example2.5.170-390.err
$ am example2.5.amc 370 GHz 570 GHz 10 MHz 0 deg 277 K 0.5385 \
>example2.5.370-570.out 2>example2.5.370-570.err
$ am example2.5.amc 550 GHz 760 GHz 10 MHz 0 deg 277 K 0.5385 \
>example2.5.550-760.out 2>example2.5.550-760.err
$ am example2.5.amc 740 GHz 1000 GHz 10 MHz 0 deg 277 K 0.5385 \
>example2.5.740-1000.out 2>example2.5.740-1000.err
```

The resulting output spectra are plotted in Fig. 2.6 through Fig. 2.10.

In these spectra, near the centers of the strongest water lines, T_b saturates as expected at the ground temperature, 277 K. Note that the relationship between opacity and brightness temperature depends on the various species contributing to the opacity at a given frequency and their respective vertical profiles. For example, the O₂ line at 487.25 GHz is more opaque at line center than the H₂O line at 448.0 GHz, but the line center brightness temperature of the O₂ line is a bit lower than that of the H₂O line, which saturates near ground temperature. This is because the H₂O profile is weighted towards a lower altitude range than that of O₂, and thus samples a warmer range of air temperature.

2.3 Accelerating computations

Models with high spectral resolution or many layers can take a long time to run, so *am* offers several means of accelerating computations. Most important is to use the OpenMP version of the code on multi-core machines. Line-by-line absorption coefficient computations are easily parallelized, and the *am* line-by-line code typically exhibits near-linear speedup in tests using up to 32 processors.

Example 2.5 introduced a second means of accelerating the model computation, the `tol` statement. This can be used to speed up most line-by-line computations by setting a numerical tolerance on the computed spectral absorption coefficients. The line

```
tol 1e-4
```

told *am* to compute line-by-line spectral absorption coefficients to a worst-case relative accuracy $\Delta k/k < 10^{-4}$. Further details about how this works are given in Section 3.5. The value used here is a reasonable choice, given the accuracy of line shape models and line catalog data. There is a trade-off between incurring the computational overhead involved in monitoring the tolerance and simply including all catalog lines. The benefit is greatest for molecules such as O₃ having many catalog lines. Examples of the effect of `tol` on compute speed are given in Section A.4.1 of Appendix A.

A third means of accelerating computations is to enable *am*'s disk cache by creating a directory referenced by the `AM_CACHE_PATH` environment variable as described in Appendix A. The disk cache is a directory containing recently-used spectral absorption coefficient

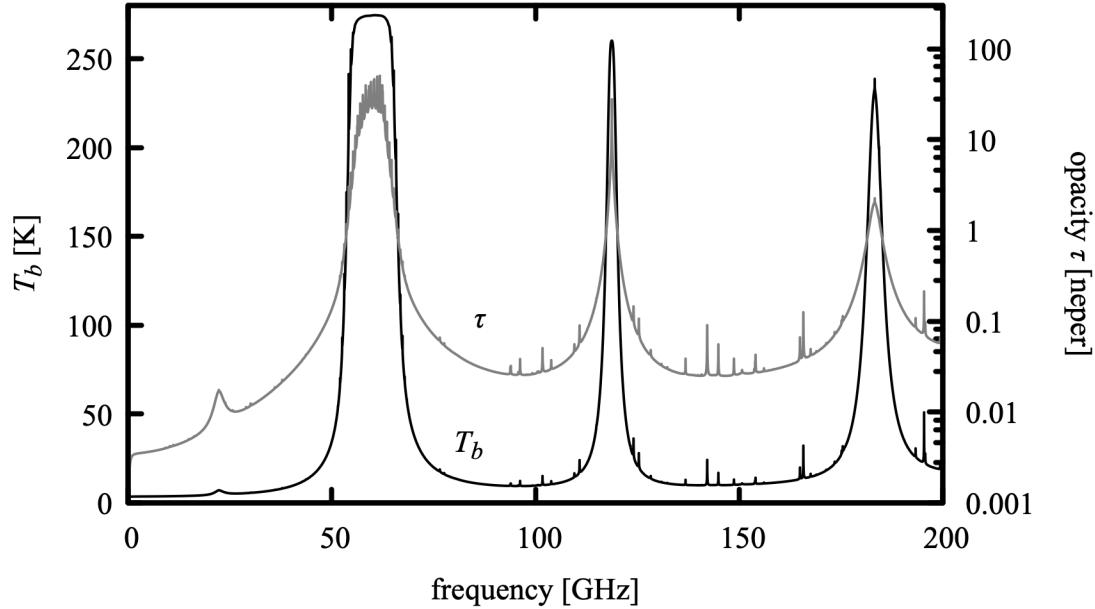


Figure 2.6 Output spectra from Example 2.5, 0–200 GHz

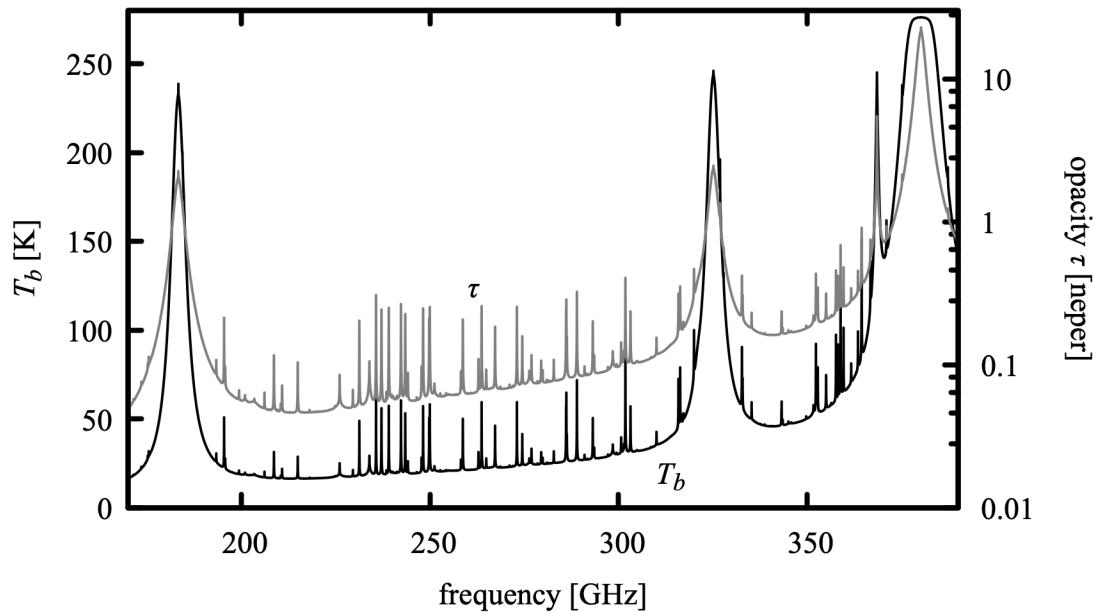


Figure 2.7 Output spectra from Example 2.5, 170 GHz–390 GHz.

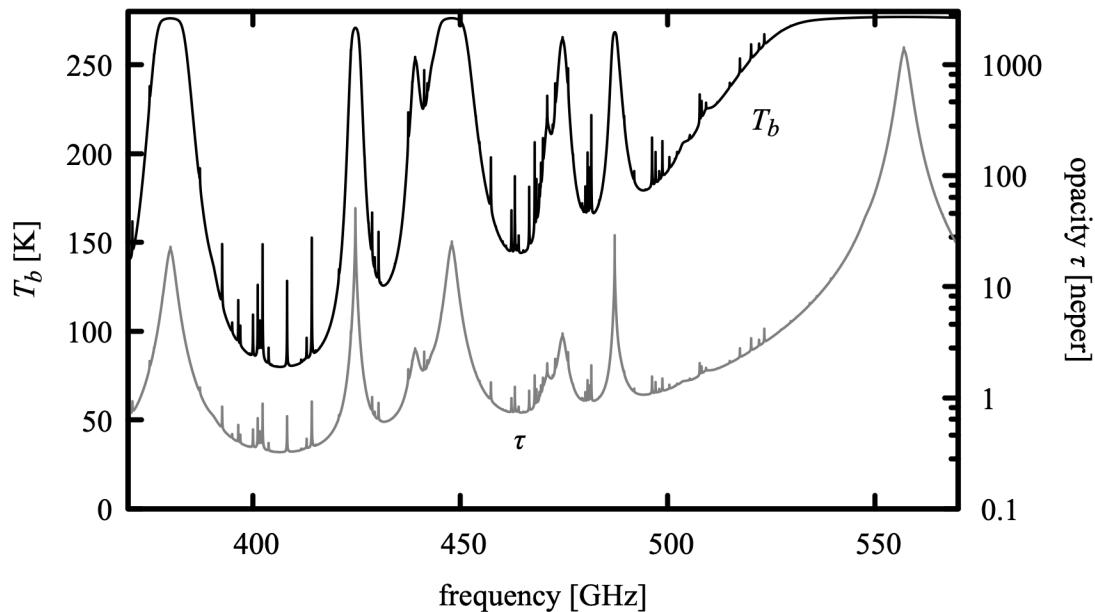


Figure 2.8 Output spectra from Example 2.5, 370 GHz–570 GHz.

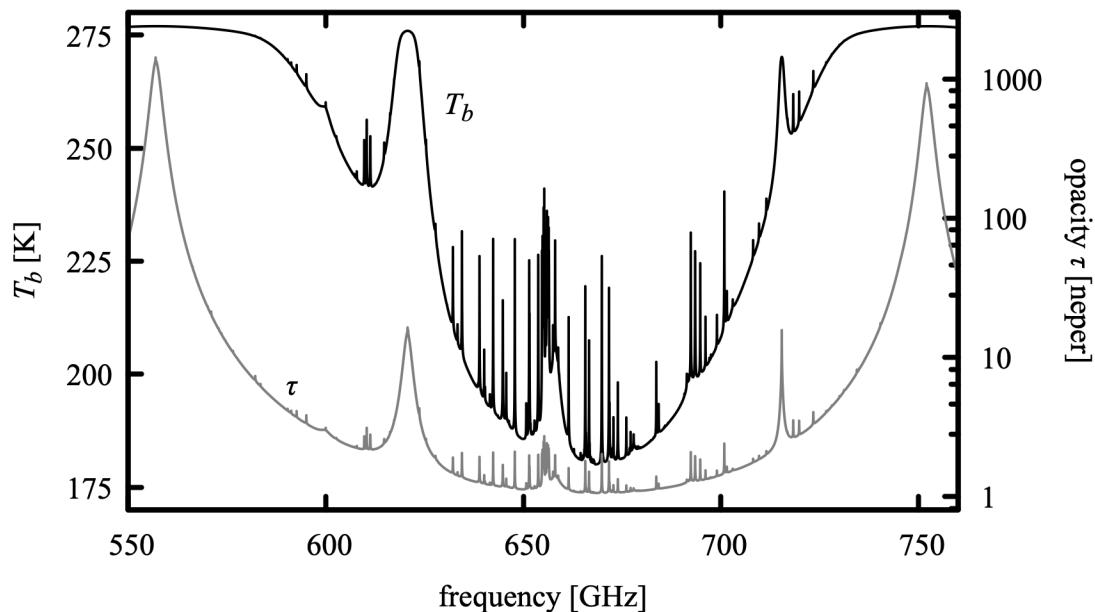


Figure 2.9 Output spectra from Example 2.5, 550 GHz–760 GHz.

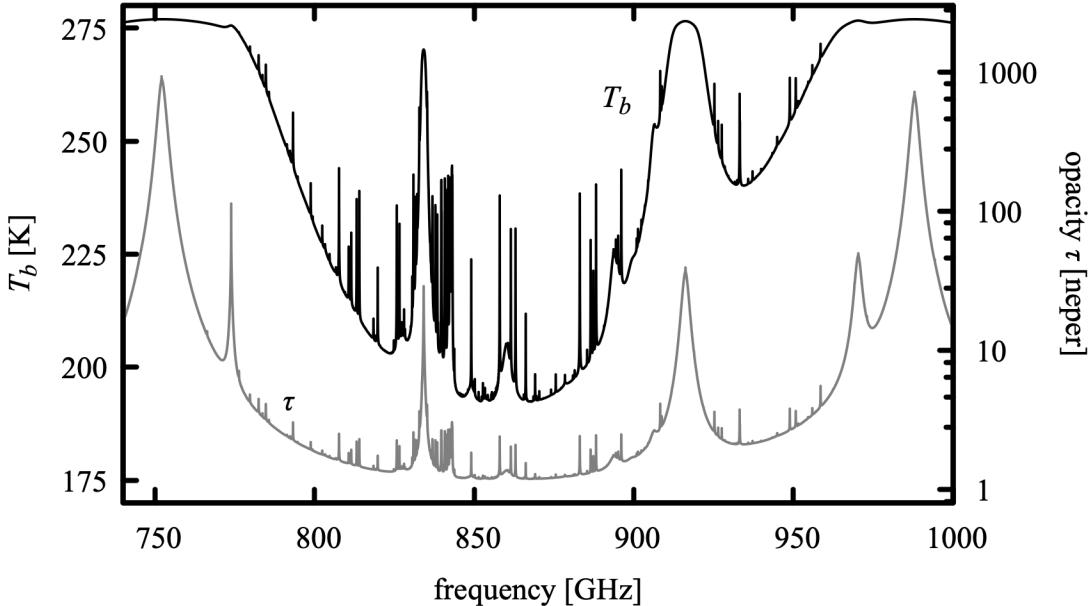


Figure 2.10 Output spectra from Example 2.5, 740 GHz–1000 GHz.

computations, organized for quick lookup via a hash function on the column type and all variables upon which the absorption coefficient depends. It is especially effective when rerunning a model with changes affecting only a subset of the model variables. Often, these changes will only involve differences in column density, and all the needed absorption coefficients will be found in the cache. Another function of the disk cache is checkpointing—a lengthy computation which is interrupted can be restarted later with minimal loss of time.

To illustrate the disk cache in action, we start with an empty cache directory, and run the command

```
$ am example2.5.amc 170 GHz 390 GHz 10 MHz 0 deg 277 K 0.5385 \
>example2.5.170-390.out
```

The output to `stderr` at the end of the run includes the comments:

```
# run time 4.146 s
# dcache hit: 0 miss: 290 discard: 0
```

that is, of the 290 spectral absorption coefficient arrays which were needed in this model, all had to be computed from scratch. Running the same command again, we get:

```
# run time 0.037 s
# dcache hit: 290 miss: 0 discard: 0
```

This time, all the needed absorption coefficients were found in the disk cache, and the model ran more than a hundred times faster. In fact, the needed files were probably still in the operating system's own file cache in main memory. Similar performance would have been obtained if all

we had wanted to change was the H₂O column density—the only new absorption coefficients needed would be on layers where the resulting new mixing ratios differed from their dry-air defaults by less than the self-broadening tolerance parameter `vmr_selfbroad_tol`, as discussed below.

Next, suppose we run the same model again, but with the surface temperature changed from 277 K to 280 K. The command is

```
$ am example2.5.amc 170 GHz 390 GHz 10 MHz 0 deg 280 K 0.5385 \
>example2.5.170-390.280.out
```

and the timing and cache usage information printed to `stderr` is

```
# run time 0.116 s
# dcache hit: 280 miss: 10 discard: 0
```

In this case, only the absorption coefficients on the base layer needed to be recomputed, still requiring just a small fraction of the time needed for the initial run.

A spectral absorption coefficient array covering a frequency interval contained within the frequency range of a previously cached computation will also be retrieved as a cache hit, provided that the frequency grid spacing is the same and the previously-cached result did not contain any spectral lines which were unresolved at the grid spacing. (The latter restriction is imposed because *am* stores only the total unresolved line count for cached arrays, and thus cannot recover the unresolved line count for a subarray.) An example of an application for which this is useful is a software tool that is repeatedly called upon to compute model atmospheric opacity in different frequency bands near selected astronomical observing frequencies. In this case, it is most effective to pre-fill the cache with an initial wide band computation covering the entire range of possible observing frequencies.

As discussed in Appendix A, there are limits on the total number of files in a cache directory and on the number of cache files per hash value. When a file needs to be discarded to make space for a newer one, the least recently used file with the same hash index as the new file will be discarded. The total number of such discards during a model run is reported along with the hit and miss counts. If *am* is being used for several unrelated jobs on the same machine, it can be useful to avoid discarding useful cache data by using a separate cache directory for each job. For example, a bash script containing an *am* computation might begin with

```
if ! [ -d am_cache ]; then
    mkdir am_cache
fi
export AM_CACHE_PATH=am_cache
```

to set up a subdirectory for *am* cache files in the current working directory.

2.3.1 Self-broadening and the disk cache

As will be discussed in the next chapter, the width of pressure-broadened lines of a particular molecule is approximated as the sum of two contributions: a self-broadened width due

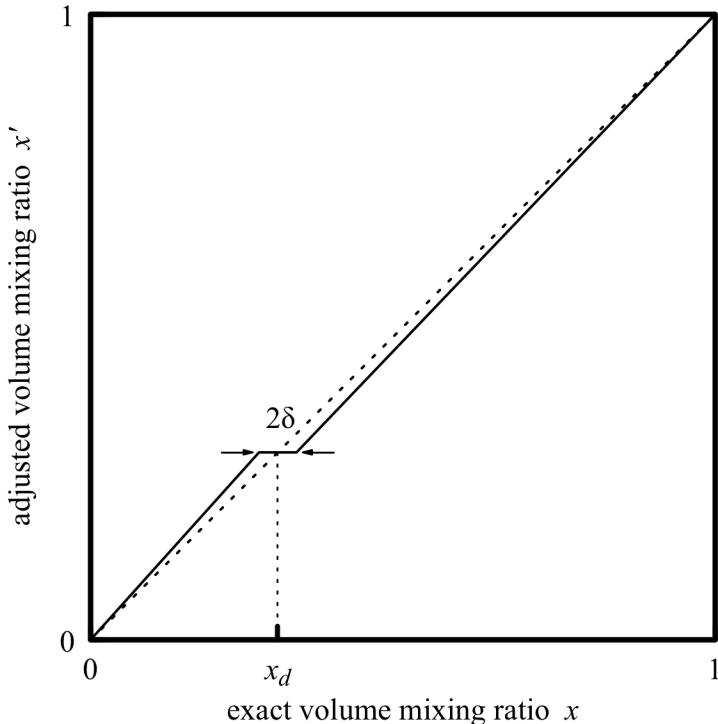


Figure 2.11 Adjustment of the volume mixing ratio applied when computing the self-broadening contribution to the pressure-broadened line width. x_d is the default mixing ratio, and δ is the value of the parameter `selfbroad_vmr_tol`. Here, δ is shown larger than the default value $\delta=0.003$.

to collisions with molecules of the same species, and an air-broadened width due to collisions with all other background gases. In general the cross sections for these two processes are different, so the pressure-broadened line width is a function of both the total pressure P and the volume mixing ratio x of the species in question. Thus we face the situation that both P and x must match for a prior computation to be a cache hit. Moreover, changes in these variables will trigger re-computation of dependent spectral absorption coefficient arrays already in memory when computing Jacobians or running a fit.

To optimize re-use of cached or previously-computed spectral absorption coefficients, we want to account for self-broadening when it is significant, and to ignore it when it is negligible. In the terrestrial atmosphere, self-broadening is negligible for trace species with the exception of H₂O. For the more abundant species O₂, the difference between air- and self-broadening cross-sections is small enough that minor deviations from its default dry air mixing ratio are also negligible. In contrast, self-broadening cannot be neglected in a laboratory gas cell containing a pure gas or a gas mixture including radiatively active species at appreciable concentrations.

am offers two means of controlling self-broadening computations. The first is an automatic

approximation controlled by the user-adjustable tolerance parameter `selfbroad_vmr_tol`. When `selfbroad_vmr_tol` is set to a non-zero value δ , *am* will use an adjusted mixing ratio x' rather than the exact mixing ratio x when computing pressure-broadened line widths. For a species with a (possibly zero) default mixing ratio x_d , $x' = x_d$ when $|x - x_d| \leq \delta$. For $|x - x_d| > \delta$, x' varies linearly with x , catching up with the exact value at the ends of the interval [0,1] as illustrated in Figure 2.11. The default value of `selfbroad_vmr_tol` is 0.003. This setting will be a good practical choice in most cases, but in applications such as detailed spectroscopic studies where accuracy takes precedence over speed, the statement

```
selfbroad_vmr_tol 0
```

can be added to the model configuration file to turn off this approximation globally.

To control self-broadening on a per-species, per-layer basis, `lineshape` statements can be used along with the `strict_self_broadening` flag. For example, the statement

```
lineshape  VVH_750 strict_self_broadening h2o_lines
```

would tell *am* to compute pressure-broadening using the exact H₂O mixing ratio on the layer where this statement appears, irrespective of the setting of `selfbroad_vmr_tol`. More details are given in Appendix B.

2.4 The *am* cookbook

A collection of examples of *am* configuration files, including examples from this manual, is available online in the *am* cookbook at <http://www.cfa.harvard.edu/~spaine/am/cookbook/>. The cookbook files are also packaged in the source code tar file and with the Windows installer.

References

- M. P. Cadeddu, J. C. Liljegren, and D. D. Turner 2013, "The Atmospheric radiation measurement (ARM) program network of microwave radiometers: instrumentation, data, and retrievals." *Atmos. Meas. Tech.* 6:2359.
- H. Le Texier, S. Solomon, and R. R. Garcia 1988, "The role of molecular hydrogen and methane oxidation in the water vapour budget of the stratosphere." *Q. J. R. Meteorol. Soc.* 114:281.
- A. Molod, L. Takacs, M. Suarez, and J. Bacmeister 2015, "Development of the GEOS-5 atmospheric general circulation model: evolution from MERRA to MERRA2." *Geosci. Model Dev.* 8:1339.
- M. M. Rienecker, M. J. Suarez, R. Gelaro, R. Todling, J. Bacmeister, E. Liu, M. G. Bosilovich, S. D. Schubert, L. Takacs, G.-K. Kim, S. Bloom, J. Chen, D. Collins, A. Conaty, A. da Silva,

W. Gu, J. Joiner, R. D. Koster, R. Lucchesi, A. Molod, T. Owens, S. Pawson, P. Pegion, C. R. Redder, R. Reichle, F. R. Robertson, A. G. Ruddick, M. Sienkiewicz, and J. Woollen 2011, “MERRA: NASA’s Modern-Era Retrospective Analysis for Research and Applications.” *J. Climate* 24:3624.

E. R. Westwater, S. Crewell, C. Mätzler, and D. Cimini 2005, “Principles of Surface-based Microwave and Millimeter wave Radiometric Remote Sensing of the Troposphere.” *Quaderni Della Societa Italiana di Elettromagnetismo* 1:50.

U.S. Standard Atmosphere 1976, Publication NOAA-S/T76-1562. Washington, D.C., U. S. Government Printing Office.

Chapter 3: Line-by-line Absorption

By line-by-line absorption, we mean absorption which can be modeled as a linear superposition of spectral lines, each associated with a particular radiative transition between internal states of individual gas molecules. Considering each gas molecule as an isolated system is an approximation, as molecules in the atmosphere undergo constant collisions. In the line-by-line picture, collisions are accounted for by their time- and ensemble-averaged effects on the spectral line shape, including *pressure broadening* of the spectral line width, *pressure shift* of the line center frequency, and *line mixing* associated with interference between spectrally overlapping lines. Line mixing is most significant for the 60 GHz O₂ band, as discussed later in this chapter. Pressure broadening and pressure shift are computed using linear pressure coefficients having a power law temperature dependence, derived principally from the HITRAN database (Gordon, et al. 2022). Collisions, besides affecting the shape and shift of spectral lines, maintain local thermodynamic equilibrium (LTE) between kinetic and internal molecular degrees of freedom. LTE conditions are always assumed to hold in *am*.

This chapter starts with the definition of the line-by-line absorption coefficient, and the definitions of the spectral line shapes available in the program. This is followed by an overview of the line-by-line molecular species included in *am*. The last section of this chapter describes the mechanics of the line-by-line computation, and ways that the computation can be accelerated.

3.1 The line-by-line absorption coefficient

The molecular absorption coefficient was defined in Eqs.(1.25) and (1.26) of Chapter 1, as part of the definition of layer opacities. To recap, if $N \cdot \sec(\theta_z)$ [cm⁻²] is the line-of-sight molecular column density in the plane-parallel approximation, the optical depth or opacity is

$$\tau(\nu) = N \cdot \sec(\theta_z) \cdot k(\nu), \quad (3.1)$$

where $k(\nu)$ [cm²] is the molecular spectral absorption coefficient. The line-by-line absorption coefficient is computed by summing over absorption coefficients associated with each of the allowed transitions of the molecule:

$$k(\nu) = \sum_{\eta\eta'} k_{\eta\eta'}(\nu) \quad (3.2)$$

Here, η' and η index the upper and lower states of a transition, respectively. These states may be degenerate over internal states that are not involved in the radiative transition, in which case $k_{\eta\eta'}(\nu)$ is the sum over the corresponding degenerate transitions. In LTE, the absorption coefficient may be written as the product of an integrated line strength $S_{\eta\eta'}(T)$, which depends on temperature only, and a normalized line shape function modeling the spectral dependence

$$k_{\eta\eta'}(\nu) = S_{\eta\eta'}(T) \cdot F(\nu, \nu_{\eta\eta'}, P, T). \quad (3.3)$$

Here, the line shape function has been written as a function of pressure and temperature, as appropriate for pressure-broadened lines. (In contrast, Doppler broadening alone depends on T only.) In the units used internally by *am*, $S_{\eta\eta'}$ is a line intensity in cm^2GHz , and F is a spectral density in GHz^{-1} .

In the HITRAN database, $S_{\eta\eta'}$ is tabulated as $S_{\eta\eta'}(T_{ref})$, where $T_{ref}=296\text{ K}$. In general, to compute the absorption coefficient, what is needed is the set of line strengths $S_{\eta\eta'}(T)$ at some other temperature. Computation of $S_{\eta\eta'}(T)$ from $S_{\eta\eta'}(T_{ref})$ under LTE conditions is discussed in the Appendix to the 1996 Edition of the HITRAN documentation (Rothman, et al. 1998). Summarizing the discussion there, we start with the definition of $S_{\eta\eta'}$ in terms of the Einstein B-coefficient $B_{\eta\eta'}$ for the transition:

$$S_{\eta\eta'} = \frac{h\nu_{\eta\eta'}}{c} w_\eta \left(1 - \frac{g_\eta w_{\eta'}}{g_{\eta'} w_\eta} \right) B_{\eta\eta'}, \quad (3.4)$$

where w_η is the probability for a molecule to be in the state η , and g_η is the state degeneracy. The only temperature-dependent terms on the right hand side of Eq. (3.4) are the probabilities w_η and $w_{\eta'}$. In LTE, the level populations are in equilibrium at a temperature T equal to the kinetic temperature of the gas, and the probabilities follow Boltzmann statistics

$$w_\eta = \frac{1}{Q(T)} g_\eta \exp\left(-\frac{E_\eta}{kT}\right), \quad (3.5)$$

where E_η is the energy of state η , and $Q(T)$ is the partition function

$$Q(T) = \sum_\eta g_\eta \exp\left(-\frac{E_\eta}{kT}\right). \quad (3.6)$$

Substituting Eq. (3.5) for w_η and $w_{\eta'}$ into Eq. (3.4),

$$S_{\eta\eta'}(T) = \frac{h\nu_{\eta\eta'}}{c} \frac{1}{Q(T)} g_\eta \exp\left(-\frac{E_\eta}{kT}\right) \left[1 - \exp\left(-\frac{h\nu_{\eta\eta'}}{kT}\right) \right] B_{\eta\eta'}, \quad (3.7)$$

where we have used $E_{\eta'} - E_\eta = h\nu_{\eta\eta'}$. Similarly, for $T=T_{ref}$,

$$S_{\eta\eta'}(T_{ref}) = \frac{h\nu_{\eta\eta'}}{c} \frac{1}{Q(T_{ref})} g_\eta \exp\left(-\frac{E_\eta}{kT_{ref}}\right) \left[1 - \exp\left(-\frac{h\nu_{\eta\eta'}}{kT_{ref}}\right) \right] B_{\eta\eta'}. \quad (3.8)$$

Taking the ratio of Eq. (3.7) and Eq. (3.8) and writing $T_\eta = E_\eta/k$,

$$\frac{S_{\eta\eta'}(T)}{S_{\eta\eta'}(T_{ref})} = \frac{\mathcal{Q}(T_{ref})}{\mathcal{Q}(T)} \frac{\exp\left(-\frac{T_\eta}{T}\right) \left[1 - \exp\left(-\frac{h\nu_{\eta\eta'}}{kT}\right)\right]}{\exp\left(-\frac{T_\eta}{T_{ref}}\right) \left[1 - \exp\left(-\frac{h\nu_{\eta\eta'}}{kT_{ref}}\right)\right]}. \quad (3.9)$$

For every catalogued molecular transition, the HITRAN database includes quantities needed to compute $S_{\eta\eta'}(T)$, including $S_{\eta\eta'}(T_{ref})$, E_η , and $\nu_{\eta\eta'}$. For each molecular isotopologue, interpolation tables for $\mathcal{Q}(T)$ are in the associated TIPS database of total internal partition sums (Gamache, et al. 2021). For the molecules included in *am*, these parameters have been converted into the program's internal units and compiled in as constant data. The line catalog data for each included molecule is in an eponymously-named *molecule.c* file in the *am* source code. Where HITRAN data have been supplemented by data from other sources, this is noted in comments in these files.

3.2 Line shapes

The line shape function $F(\nu, \nu_{\eta\eta'}, P, T)$ in Eq. (3.3) models the spectral dependence of each term of the spectral absorption coefficient sum, Eq. (3.2). The width and shape of $F(\nu, \nu_{\eta\eta'}, P, T)$ are governed by collisions and Doppler broadening. Models for collisional and motional broadening can be quite complex if they attempt to model processes taking place during a collision, or involve detailed consideration of the internal degrees of freedom of the colliding molecules. (Hartmann, et al. (2021) presents a useful overview.) In contrast, most practical line-by-line computations, including those in *am*, employ standard line shapes based on the simplest possible models for the processes responsible for line broadening. Before proceeding further, it is worth reviewing the time scales of the relevant processes to get a sense of the approximations involved and their domain of applicability.

3.2.1 Approximate time scales for line broadening processes

Near the center of a spectral line, the collisional width is governed by the mean time τ_c between collisions affecting the phase of the radiating molecule. At standard temperature and pressure, $\tau_c \sim 10^{-10}$ s, so collisional line widths $\gamma_c = (2\pi\tau_c)^{-1}$ are of order 1 GHz at STP. If n is the total gas density, v is the mean relative velocity between radiating molecule and colliding perturber, and σ_c is a cross-section, then $\tau_c = (\sigma_c n v)^{-1}$. Since the density $n = P/kT$, and the velocity $v \propto T^{1/2}$, it follows that $\tau_c \propto T^{1/2}/P$ and $\gamma_c \propto P \cdot T^{-1/2}$. The term *pressure broadening* arises from this pressure dependence. For the temperature dependence, the HITRAN database includes an empirical value for the exponent of T which improves agreement with experimental data, as discussed below.

A second key collisional time scale is the *duration of collision*, τ_c^* , which is independent of

gas density, and depends on the relative molecular velocity and an interaction length. The time scale for τ_c^* at atmospheric temperatures is in the range $\tau_c^* \sim 10^{-13}$ s to $\tau_c^* \sim 10^{-12}$ s. The *impact approximation*, which underlies the standard collision-broadened line shapes, consists of ignoring the details of processes taking place at time scales of order τ_c^* . The standard line shapes are thus characterized solely by the width γ_c , and cannot be expected to correctly represent the absorption at frequencies greater than $\gamma_c^* = (2\pi \tau_c^*)^{-1}$ from the line center, where $\gamma_c^* \sim 1$ THz. For this reason, empirical correction factors or cutoffs are often applied to the far wings of standard collisional line shapes.

Doppler broadening becomes important as the mean velocity of a molecule along the line of sight, $\bar{v} = (2kT/m)^{1/2}$, becomes high enough that molecular motion causes a significant random phase shift within one period of oscillation of the radiation field. The characteristic time scale is the mean time for the molecule to travel one wavelength $\lambda_{\eta\eta'}$ which is $\tau_D = \lambda_{\eta\eta'} / \bar{v}$. The corresponding Doppler line width is $\alpha_D = \tau_D^{-1}$, or equivalently $\alpha_D = v_{\eta\eta'}(\bar{v}/c)$.

For thermal radiation in the terrestrial atmosphere, Doppler broadening is negligible relative to pressure broadening in the troposphere, becoming important only in the stratosphere or higher, at a pressure which is approximately proportional to frequency. As a reference point, for a representative atmospheric temperature $T = 270$ K, $\bar{v} = 500 \text{ m}\cdot\text{s}^{-1}$ for the relatively light H₂O molecule and $\bar{v} = 300 \text{ m}\cdot\text{s}^{-1}$ for ozone. The factor \bar{v}/c is thus of order 10^{-6} . At 1 THz, $\alpha_D \approx 1$ MHz and Doppler broadening becomes comparable to pressure broadening at $P \approx 1$ mbar. At lower frequencies, this occurs at proportionally lower pressure (i.e. at higher altitude).

The last broadening time scales to consider are the *natural lifetime* $A_{\eta\eta'}^{-1}$ for spontaneous emission, and the reciprocal of the thermally-stimulated emission rate $B_{\eta\eta'} u(\nu, T)$, where $A_{\eta\eta'}$ and $B_{\eta\eta'}$ are Einstein A and B coefficients, and $u(\nu, T)$ is the spectral energy density of blackbody radiation at temperature T . For the (mostly rotational) transitions involved in the spectral domain covered by *am*, $A_{\eta\eta'}^{-1} \sim 1$ s, and only a small number of allowed decay transitions are possible from a given upper state η' . Consequently, spontaneous emission is typically negligible relative to collisional and Doppler broadening. The thermally-stimulated emission rate is

$$B_{\eta\eta'} u(\nu, T) = \frac{A_{\eta\eta'}}{e^{h\nu/kT} - 1} \approx \frac{kT}{h\nu} A_{\eta\eta'} \quad (3.10)$$

in the low-frequency limit $h\nu \ll kT$. For $T = 270$ K, $kT/h\nu = 1$ for $\nu = 5.6$ THz. At microwave frequencies of order 1 GHz, and at similar temperature, the thermally-stimulated rate will exceed the spontaneous rate by a factor of several thousand, but the associated broadening will nevertheless remain negligible relative to collisional and Doppler broadening in the terrestrial atmosphere.

3.2.2 Collisional line width and line shift

In the discussion above, the collisional line width was approximated as being proportional to a total background perturber pressure P . In reality, the collision broadening cross section is different for different radiator-perturber pairs. One clear distinction to make is between identical and non-identical radiator and perturber—in the former case there is a greater possibility for resonant interaction leading to a larger broadening cross section. The HITRAN database provides at least two broadening coefficients for each transition, γ_{air} and γ_{self} , for air- and self-broadening, respectively. The total pressure-broadened line width is computed as

$$\gamma(P, p_{self}, T) = \left(\frac{T_{ref}}{T}\right)^n [\gamma_{air}(P - p_{self}) + \gamma_{self} p_{self}] \quad (3.11)$$

(Rothman, et al. 1998), where P is the total pressure, and p_{self} is the partial pressure of the radiating species. The temperature exponent n , also cataloged separately for each transition, is the empirically-improved counterpart to the idealized value $n=1/2$ mentioned above. It should be noted that HITRAN is gradually being expanded to include broadening coefficients specific to foreign perturbers other than dry air, but these are not yet implemented in *am*.

In addition to pressure broadening, collisions can also induce a shift of the line frequency. In HITRAN, this is treated with a single coefficient $\delta_{\eta\eta'}$ for each transition in the catalog, which multiplies the total pressure. The shifted frequency is

$$\nu_{\eta\eta'}^* = \nu_{\eta\eta'} + \delta_{\eta\eta'} \cdot P \quad (3.12)$$

(Rothman, et al. 1998).

3.3 Line shapes in *am*

am supports several standard collisional line shapes, a pure Doppler line shape, and an approximation (Kielkopf 1973) of a Voigt line shape formed by the convolution of Lorentz and Doppler line shapes. The line shape functions and their names in *am* configuration files are summarized in Table 3.1.

3.3.1 Collisional line shapes

All of the collisional line shapes listed in Table 3.1 are developed within the impact approximation discussed above in Section 3.2.1, and depend on a single line width parameter $\gamma(P, p_{self}, T)$. The differences between them lie in the underlying physical model of the collision process. More precisely, since under the impact approximation collisions are taken to be effectively instantaneous, the differences between impact models amount to different assumptions about the distribution of post-collision states of the molecule.

If we consider the radiating molecule to be a classical harmonic oscillator, possessing a dipole moment coupled to an oscillating electromagnetic field, the underlying model for the Lorentz (1916) line shape is that collisions occur as a Poisson process with mean time $1/\gamma$ between collisions, and that the effect of a collision is to uniformly randomize the phase of the

TABLE 3.1. LINE SHAPE FUNCTIONS

UNCOUPLED LINES	
NAME	LINE SHAPE FUNCTION
Doppler	$F_D(\nu - \nu_0, \alpha) = \frac{1}{\alpha \sqrt{\pi}} \exp\left[-\left(\frac{\nu - \nu_0}{\alpha}\right)^2\right]$
Lorentz	$F_L(\nu - \nu_0, \gamma) = \frac{1}{\pi} \frac{\gamma}{(\nu - \nu_0)^2 + \gamma^2}$
Gross	$F_G(\nu, \nu_0, \gamma) = \frac{1}{\pi} \frac{4\nu^2 \gamma}{(\nu^2 - \nu_0^2)^2 + 4\nu^2 \gamma^2}$
Full_Lorentz	$F_{FL}(\nu, \nu_0, \gamma) = \left(\frac{\nu}{\nu_0}\right) [F_L(\nu - \nu_0, \gamma) - F_L(\nu + \nu_0, \gamma)]$
VVW	$F_{VVW}(\nu, \nu_0, \gamma) = \left(\frac{\nu}{\nu_0}\right)^2 [F_L(\nu - \nu_0, \gamma) + F_L(\nu + \nu_0, \gamma)]$
VVH	$F_{VH}(\nu, \nu_0, \gamma) = \left(\frac{\nu}{\nu_0}\right) \frac{\tanh\left(\frac{h\nu}{2kT}\right)}{\tanh\left(\frac{h\nu_0}{2kT}\right)} [F_L(\nu - \nu_0, \gamma) + F_L(\nu + \nu_0, \gamma)]$
VVH_750	$F_{VH_750}(\nu, \nu_0, \gamma) = \left(\frac{\nu}{\nu_0}\right) \frac{\tanh\left(\frac{h\nu}{2kT}\right)}{\tanh\left(\frac{h\nu_0}{2kT}\right)} [F_{L_750}(\nu - \nu_0, \gamma) + F_{L_750}(\nu + \nu_0, \gamma)]$ where $F_{L_750}(\nu - \nu_0, \gamma) = \begin{cases} F_L(\nu - \nu_0, \gamma) - F_L(750 \text{ GHz}, \gamma) & \nu - \nu_0 < 750 \text{ GHz} \\ 0 & \text{otherwise} \end{cases}$
Voigt-Kielkopf	$F_{V-K}(\nu - \nu_0, \alpha, \gamma) = \int_{-\infty}^{\infty} F_L(\nu - \nu', \gamma) F_D(\nu' - \nu_0, \alpha) d\nu'$ evaluated using Kielkopf (1973) approximation.

oscillator.

The Van Vleck–Weisskopf (VVW) line shape (Van Vleck and Weisskopf 1945) is developed under the modified assumption that, post collision, the probability distribution of the phase of the oscillating dipole is proportional to a Boltzmann factor in the interaction energy between the dipole and the instantaneous field. Since the notion of an instantaneous value of the

TABLE 3.1. LINE SHAPE FUNCTIONS, CONTINUED

COUPLED LINES	
NAME	LINE SHAPE FUNCTION
vvw_coupled	$F_{VVW_coupled} = \frac{1}{\pi} \left(\frac{\nu}{\nu_0} \right)^2 \left[\frac{y \cdot (1+g \cdot P^2) + y \cdot P \cdot (\nu - \nu_0 - \delta \nu \cdot P^2)}{y^2 + (\nu - \nu_0 - \delta \nu \cdot P^2)^2} \right. \\ \left. + \frac{y \cdot (1+g \cdot P^2) - y \cdot P \cdot (\nu + \nu_0 + \delta \nu \cdot P^2)}{y^2 + (\nu + \nu_0 + \delta \nu \cdot P^2)^2} \right]$ <p>where</p> $y = \left[y_0 + y_1 \cdot \left(\frac{T_{ref}}{T} - 1 \right) \right] \cdot \left(\frac{T_{ref}}{T} \right)^x$ $g = \left[g_0 + g_1 \cdot \left(\frac{T_{ref}}{T} - 1 \right) \right] \cdot \left(\frac{T_{ref}}{T} \right)^{2x}$ $\delta \nu = \left[\delta \nu_0 + \delta \nu_1 \cdot \left(\frac{T_{ref}}{T} - 1 \right) \right] \cdot \left(\frac{T_{ref}}{T} \right)^{2x}$ $x = 0.8 \text{ (Makarov et al. 2011)}$

field is only meaningful when the frequency is low compared with $1/\tau_c^*$, the VVW line shape is only applicable at millimeter-wave frequencies and lower. (In fact, the high-frequency asymptotic limit of the VVW line shape is constant, and its integral therefore diverges.) The VVW line shape reduces to the Debye theory for non-resonant absorption, important for O₂, in the zero-frequency limit. It has also been shown (Hill 1986) to more accurately model the 22 GHz H₂O line than the other simple collisional line shapes.

Continuing with the classical harmonic oscillator picture, the Gross (1955) line shape is derived under the assumption that collisions cannot instantaneously alter the position of the oscillator; only the momentum is allowed to change discontinuously. The resulting line shape is identical to that of the classical damped harmonic oscillator. The form of the Gross line shape given in Table 3.1 is normalized to unity.

The *principle of detailed balance* (Reif 1965) states that, in thermal equilibrium, absorption and emission rates must be equal within any spectral interval. This becomes important for broad lines, and in far line wings, when the Boltzmann factor $e^{h\nu/kT}$ governing the relative populations of the upper and lower states involved in the transition cannot be approximated as constant across the width of the line. The Van Vleck–Huber (VVH) line shape (Huber & Van Vleck 1966) is a modification to the VVW line shape which, unlike the collisional line shapes mentioned so far,

strictly satisfies the detailed balance condition. Like the VVW line shape, the unmodified VVH line shape is applicable at millimeter-wave and lower frequencies, and the integral taken to infinite frequency diverges, albeit logarithmically. A solution is to introduce a line wing cutoff at a chosen detuning from the line center. This cutoff is applied separately to the positive- and negative-frequency components of the line shape function, as shown in the entry for the `vvh_750` line shape function in Table 3.1. The choice of 750 GHz for the cutoff frequency is customary in much of the literature; it corresponds roughly to the reciprocal of the duration-of-collision time scale mentioned above in Section 3.2.1. With the cutoff imposed, the integral over all frequencies no longer diverges. Moreover, at high frequencies the line shape takes on a Lorentz form, which is an appropriate approximation there.

3.3.2 Line shape for coupled lines

The simple collisional line shapes in the preceding section apply to a single molecular transition in isolation, considered to be independent of other transitions which may occur within the same molecule. This approximation can break down when two or more transitions overlap spectrally, and when these overlapping transitions involve states which can be coupled by collisions. In this case, the overlapping transitions can interfere, leading to a significant modification of the line shape. This phenomenon is most often referred to as *line mixing* or *line coupling*. As pressure increases, increasing the collisional transfer rate, the effect of line coupling is to merge overlapping coupled lines into an unresolved band, and to narrow the overall width of this band. Line mixing is of particular importance for the 60 GHz fine structure band of O₂.

In practical treatments of line mixing (Gordon 1967, Rosenkranz 1975, Lam 1977, Smith 1981), the radiating molecule is described by a density matrix in *line space*, a doubled state space in which the base states correspond to allowed transitions rather than molecular energy levels. A complex relaxation matrix, whose elements depend on pressure and temperature, describes both the transition frequencies and the collisional transfer rates between states in line space. Computation of the net band shape for a set of coupled lines involves inverting this matrix at each frequency of interest. In practice (as for O₂ at atmospheric pressure), the significant coupling is limited to particular subsets of lines, and the off-diagonal coupling coefficients of the relaxation matrix are sufficiently small that it can be approximately inverted analytically. By doing so, the band shape can be computed efficiently in the form of a line-by-line sum, using a coupled line shape which depends on coupling coefficients associated with each transition. This approach is much faster computationally than inverting the full relaxation matrix, and also facilitates derivation of the coupling coefficients from experimental measurements. It is important to recognize, however, that the coupled line shape for a single transition is only meaningful in the context of the sum over the complete band of coupled lines. For this reason, techniques such as line strength or line wing cutoffs, which are useful for accelerating the computation of uncoupled line absorption, cannot be applied in the same way to a coupled band.

In *am*, coupled lines are described using the `vvw_coupled` line shape in Table 3.1, which is based on an approximate inversion of the relaxation matrix to second order in pressure (Smith 1981). The temperature dependence of the coupling coefficients follows the semi-

empirical scheme used in Makarov, et al. (2011), from which the O₂ line mixing data used in *am* are derived.

3.3.3 Doppler and Voigt line shapes

Two line shapes are available in *am* for treating Doppler broadening. The `Doppler` line shape is applicable to a gas which can be regarded as essentially collisionless. In this case the broadened line shape is the integral of the Doppler-shifted line frequency over the line-of-sight component of the Maxwell–Boltzmann velocity distribution, which results in a Gaussian profile.

When Doppler and pressure broadening are both significant, a standard approximation is to treat them as independent processes and model the spectral line shape as the convolution of pure Doppler and pressure broadened line shapes. At the low atmospheric pressures for which Doppler broadening becomes significant relative to pressure broadening, the pressure broadening contribution is modeled accurately with a Lorentz profile. The convolution of Doppler and Lorentz profiles is known as a Voigt profile. Direct evaluation of the convolution integral is computationally expensive, so a number of approximate algorithms have been developed involving different tradeoffs between accuracy and speed. (See Schreier (1992) for a modern review.) In *am*, a fast approximation due to Kielkopf (1973) is used, available as the `voigt-Kielkopf` line shape.

Use of the `voigt-Kielkopf` line shape was illustrated in Example 2.5 of Chapter 2. There, since the maximum frequency at which the model was intended to be used was 1 THz, the `voigt-Kielkopf` line shape was used from the 3 mbar level upwards. Below this level, and at lower frequencies, Doppler broadening is negligible while collisional effects such as line mixing which are not modeled by a simple Lorentzian become important. Moreover, computation of the `voigt-Kielkopf` line shape is about an order of magnitude slower than the pure pressure broadened line shapes, so it is desirable to avoid its use where it is not necessary.

It is important to note that although the Voigt function approaches a Lorentzian in the limit of small Doppler width, the opposite is not the case. That is, the Voigt function does not approach a Gaussian in the limit of small collisional width. This is because the wings of the Lorentz line shape, which fall off quadratically with detuning, eventually dominate the Doppler line wings, which fall off exponentially.

3.3.4 Default line shapes

am allows any of the line shapes in Table 3.1 to be applied to any line-by-line species, on a per-layer basis, using the `lineshape` keyword in the configuration file. Details are given in Appendix B. The default line shape for most line-by-line species is the `Gross` line shape. The exceptions are `h2o_lines`, `o2_coupled`, and `o` (atomic oxygen). For `h2o_lines`, the default line shape is `VVH_750`, for compatibility with the MT_CKD water vapor continuum model as discussed in Chapter 4: Continuum Absorption. The column type `o2` is a composite of the column types `o2_coupled`, which includes coupled ¹⁶O₂ nonresonant and fine structure band lines, and `o2_uncoupled`, which includes the remaining catalog lines of ¹⁶O₂ and all lines for

other O₂ isotopologues. The default line shape for o2_coupled is the vvw_coupled line shape, while for o2_uncoupled it is the gross line shape. The default line shape for o is voigt-kielkopf.

3.4 Molecular species in *am*

As mentioned at the beginning of this chapter, line-by-line data for *am* are derived primarily from the HITRAN database (Gordon, et al. 2022). In some cases, the HITRAN data have been modified or augmented as noted below. The basic line-by-line column types in *am* combine all significant isotopologues of a given molecule. For these, the column density or mixing ratio in a configuration file refers to the sum over all isotopologues, and the line strengths follow the HITRAN scheme of weighting the line strengths by terrestrial abundance. Additional column type names are available to specify particular isotopologues of a given molecule—these are listed in Table B.4 of Appendix B. For single-isotopologue column types, the column density or mixing ratio refers to the actual value for that particular isotopologue, and the HITRAN catalog line strengths are adjusted to remove the abundance weighting. So, for example,

```
column ch4 vmr 1.9e-6
```

is equivalent to

```
column 12ch4 vmr 1.87772e-06
column 13ch4 vmr 2.10959e-08
column 12ch3d vmr 1.16993e-09
```

The remainder of this section gives a graphical résumé of selected line-by-line species having particular importance in the Earth’s atmosphere or as laboratory test molecules. For each species, a plot of the absorption coefficient including all isotopologues is given for representative conditions of pressure and temperature. These plots are intended to serve as a guide to the spectral intervals where a given species is radiatively important, and to illustrate where the line catalogs have been cut off in the band wings. Many more species are available in the program but are not plotted here; these can be found in Table B.4 of Appendix B.

3.4.1 Water Vapor (column type h2o_lines)

The column type h2o_lines represents the line-by-line component of the absorption coefficient of water vapor, excluding the water vapor continuum absorption. Most configuration files will not use h2o_lines, but instead will use the composite h2o column type, which combines h2o_lines, h2o_continuum, and h2o_optical_refractivity. For specifying particular isotopologues of H₂O, the column types listed below h2o_lines in Section B.4 may be used. The line catalog data for H₂O are derived from a modified version of the HITRAN database (aer_v_3.8.1) maintained by the Radiative Transfer Working Group at AER, Inc. (<http://rtweb.aer.com>).

Figure 3.1 is a plot of the spectral absorption coefficient for h2o_lines, computed at a background air pressure $P=500\text{ mbar}$ and temperature $T=270\text{ K}$, using the default vvH_750 line

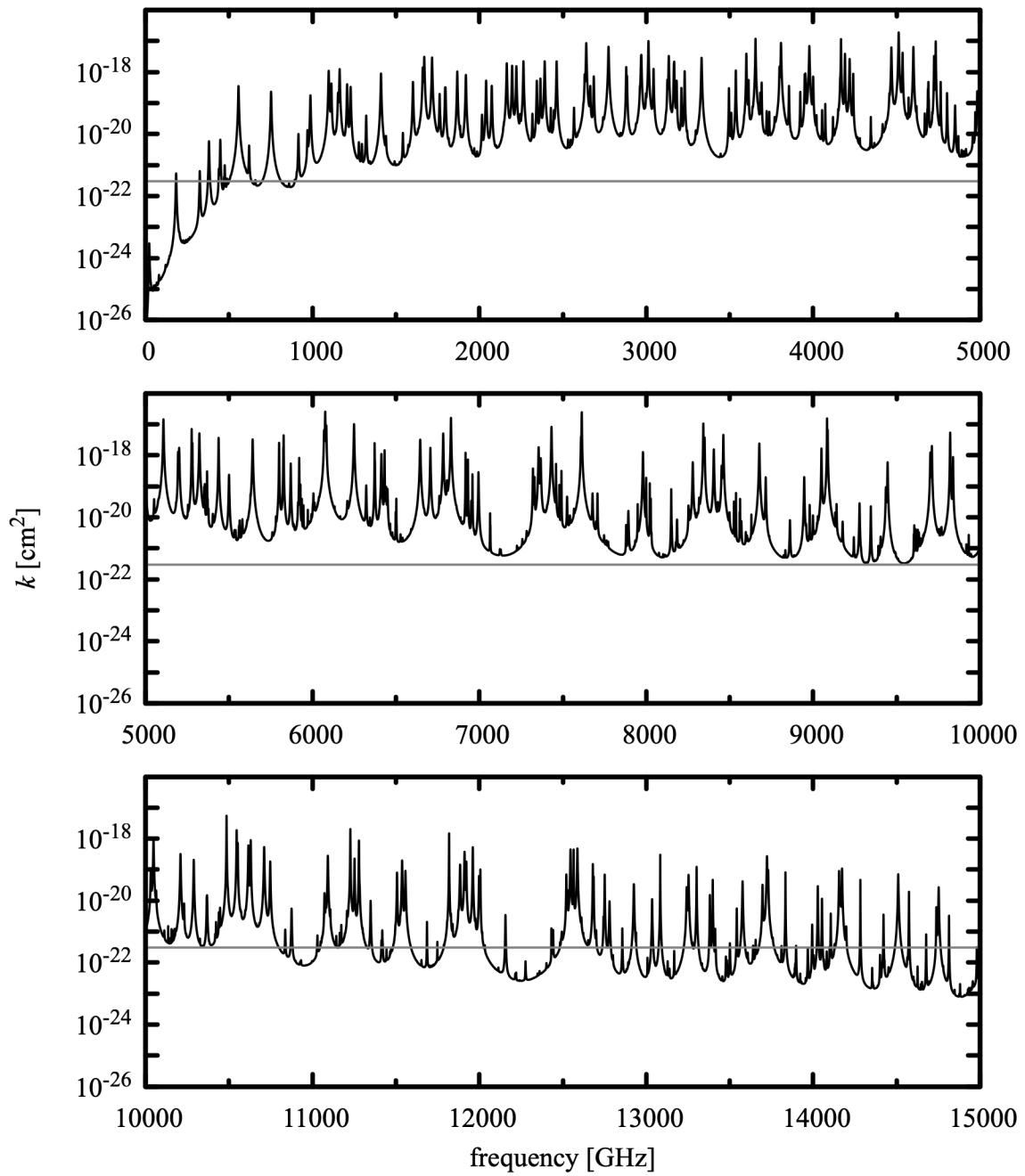


Figure 3.1 Absorption coefficient for water vapor (lines only, no continuum). $P=500 \text{ mbar}$, $T=270 \text{ K}$, vvh_750 line shape. The gray line corresponds to $\tau=1$ for a column density of 1 mm precipitable water ($n=3.3 \cdot 10^{21} \text{ cm}^{-2}$).

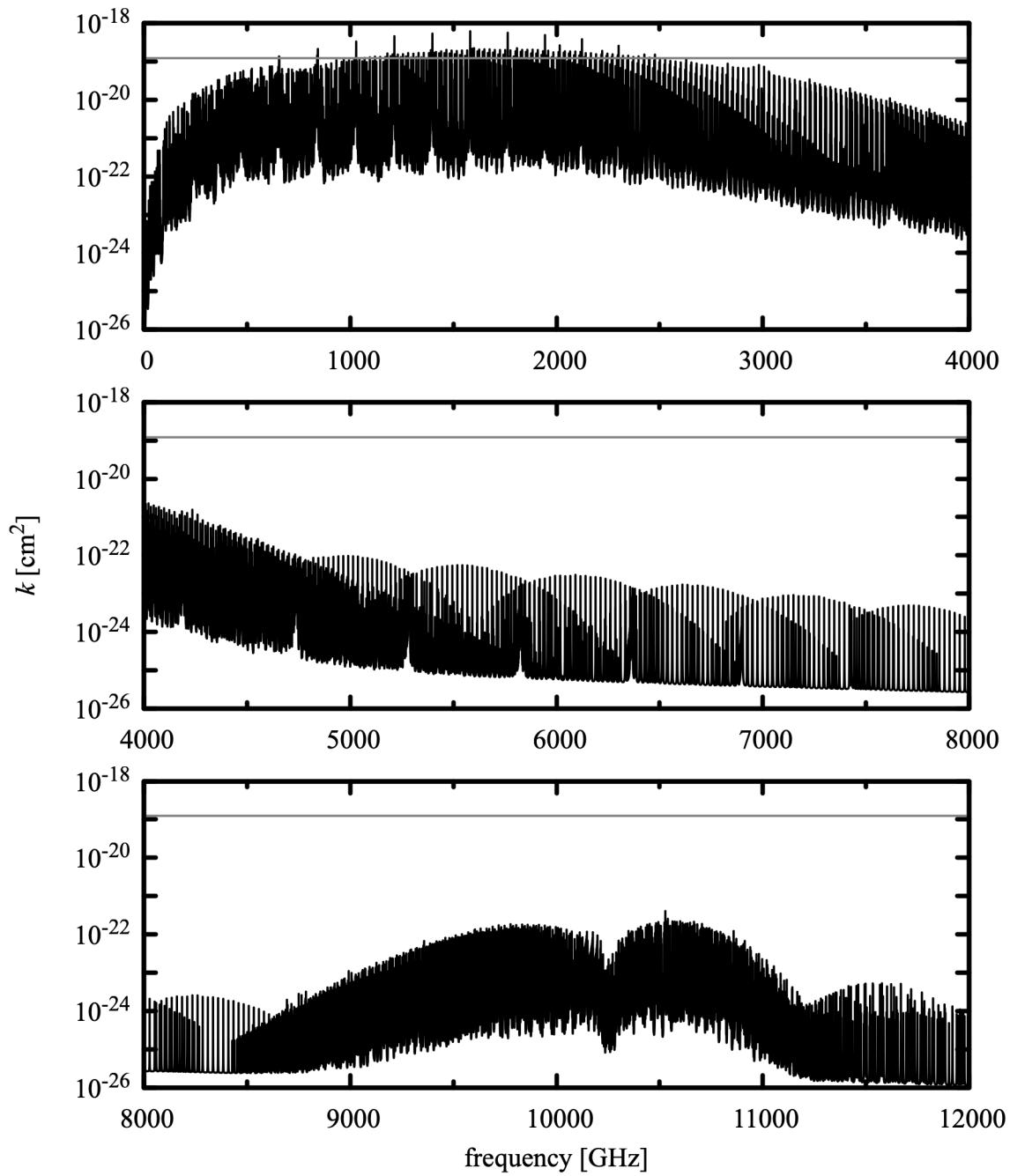


Figure 3.2 Absorption coefficient for ozone $P = 30$ mbar, $T = 220$ K, Gross line shape. The gray line corresponds to $\tau = 1$ for a typical total atmosphere zenith column density of 300 Dobson units ($n = 8.1 \cdot 10^{18}$ cm⁻²).

shape. Self-broadening has been neglected, and the water vapor continuum is not included. The light gray line at $k=3\cdot10^{-22}\text{cm}^2$ corresponds to optical depth $\tau=1$ at a column density $n=3.3\cdot10^{21}\text{cm}^{-2}$. This is equivalent to 1 mm precipitable water vapor.

For H₂O transitions, the pressure broadening coefficient for self-broadening can be up to an order of magnitude higher than that for foreign broadening. For this reason, self-broadening is especially important in the warm lower troposphere where the H₂O volume mixing ratio can be as high as 0.05. The default setting of `vmr_selfbroad_tol` will typically provide linewidth accuracy limited by the uncertainty of line catalog data as discussed in Section 2.3.1. For spectroscopic applications, setting `vmr_selfbroad_tol` to zero is recommended. Alternatively, self-broadening can be controlled selectively by use of `lineshape` statements with the `strict_self_broadening` flag as in earlier versions of *am*.

3.4.2 Ozone (column type o3)

The spectral absorption coefficient for ozone is plotted in Figure 3.2, for background air pressure and temperature $T=220\text{ K}$, approximating conditions in the stratosphere. k has been computed using the default `gross` line shape for the column type `o3`, and self-broadening has been neglected. The light gray line at $k=1.23\cdot10^{-19}\text{cm}^2$ corresponds to an optical depth $\tau=1$ for a column density $n=8.1\cdot10^{18}\text{cm}^{-2}$. This is equivalent to 300 Dobson units, a typical total O₃ zenith column density from the ground.

3.4.3 Oxygen (column type o2)

The O₂ molecule in its ground electronic state has electronic spin $S=1$, with an associated permanent magnetic dipole moment. Addition of S and the molecular rotational angular momentum N results a triplet of states for each rotational state N , having total angular momenta $J=N-1, N, N+1$. For ¹⁶O₂, and ¹⁸O₂, N is restricted by exchange symmetry to $N=1, 3, 5, \dots$. Coupling between S and N results in a fine structure splitting of each triplet, such that each state $J=N$ lies above the two states $J=N\pm1$ by an interval corresponding to approximately 60 GHz (the one exception is the interval $J=1, N=1 \rightarrow J=0, N=1$ at 118.75 GHz). Magnetic dipole transitions are allowed for $N, J=N \rightarrow N, J=N\pm1$ producing a dense band near 60 GHz. In addition, the states with $J=N\pm1$ have a constant non-zero projection of S along J which gives rise to nonresonant absorption associated with collisions which reorient J . (Van Vleck 1947) Collisions are also efficient at transferring population between adjacent triplets $N, N\pm2$, resulting in line mixing which strongly couples the lines within the fine structure band and, to a lesser extent, couples the fine structure band to the nonresonant absorption.

Besides the 60 GHz fine structure band and nonresonant absorption, rotational magnetic dipole transitions with $\Delta N=\pm2, \Delta J=0, \pm1$ are allowed. These appear in the submillimeter band as groups of three lines, with a separation between lines in each group of about 60 GHz corresponding to the fine structure splitting of the rotational levels, and an interval between groups of about 345 GHz corresponding to the $\Delta N=2$ rotational intervals. These rotational lines are not

significantly affected by line coupling.

In *am*, the absorption arising from coupled lines and uncoupled lines of O₂ is represented by two column types, `o2_coupled` and `o2_uncoupled`, respectively. These are combined into the more convenient composite column type `o2`. The reason for the division of `o2` into two subtypes is to support the assignment of different line shapes to the coupled and uncoupled lines; by default these are `vww_coupled` and `gross`, respectively. It should be noted that the column type `o2` does not include collision-induced absorption, discussed in the next chapter.

The line coupling data for `o2_coupled` are from Makarov, et al. (2011), and the frequencies and line broadening coefficients for these lines are taken from Tretyakov, et al. (2005). Nonresonant O₂ absorption is represented by a single ¹⁶O₂ line at zero frequency. The broadening parameter for this line was taken from Liebe (1985), and the strength was adjusted to fit the sea level mid-latitude absorption and emission measurements from various sources compiled in Table 2 of Danese and Partridge (1989). With the nonresonant line strength thus fixed, the bias b_i applied to the first-order mixing parameters y_i by Makarov, et al. (2011) was adjusted by +0.0009 bar⁻¹, or about 6% of the authors' original value. This adjustment improves the agreement of the high-frequency wing of the coupled band plus uncoupled lines with the laboratory measurements of Meshkov and DeLucia (2007), and also serves to make the coupled band absorption coefficient positive over the full atmospheric temperature range. For `o2_uncoupled`, the HITRAN catalog data are used unmodified apart from three submillimeter lines from the AER database mentioned above in Section 3.4.1.

Column types for individual isotopologues of O₂ are available, as listed in Table B.4. Because line coupling data are only available for ¹⁶O₂, the line-by-line absorption coefficients for all other isotopologues are computed entirely with uncoupled lines, even in the fine structure band.

Figure 3.3 is a plot of the spectral absorption coefficient for `o2`, computed at a total air pressure $P=500$ mbar and temperature $T=260$ K. Pressure broadening has been computed using the default O₂ volume mixing ratio $x=0.209$. The light gray line at $k=2.25 \cdot 10^{-25}$ cm² corresponds to an optical depth $\tau=1$ for a typical sea level zenith column density $n=4.4 \cdot 10^{24}$ cm⁻².

Figures 3.4 through 3.7 present comparisons of the output of *am* with experimental measurements at microwave through millimeter wavelengths dominated by O₂ absorption. Figure 3.4 shows sea-level zenith attenuation and atmospheric brightness temperature measurements, made in connection with various cosmic microwave background measurements and collected in Table 2 of Danese and Partridge (1989), with the output of *am* for a mid-latitude atmosphere derived from the CIRA-86 climatology (COSPAR 1986). Figure 3.5 and Figure 3.6 compare model computations for a dry atmosphere, again based on the CIRA-86 climatology, with zenith attenuation measurements from Reber (1972) in the 60 GHz band. These were made using the sun as a background source, correcting for foreground atmospheric emission and the relatively small water vapor attenuation. Recent millimeter-wave cavity ringdown measurements of the volume absorption coefficient of pure O₂ (Meshkov and De Lucia 2007) are compared with the corresponding *am* computation in Figure 3.7. For this comparison, the volume

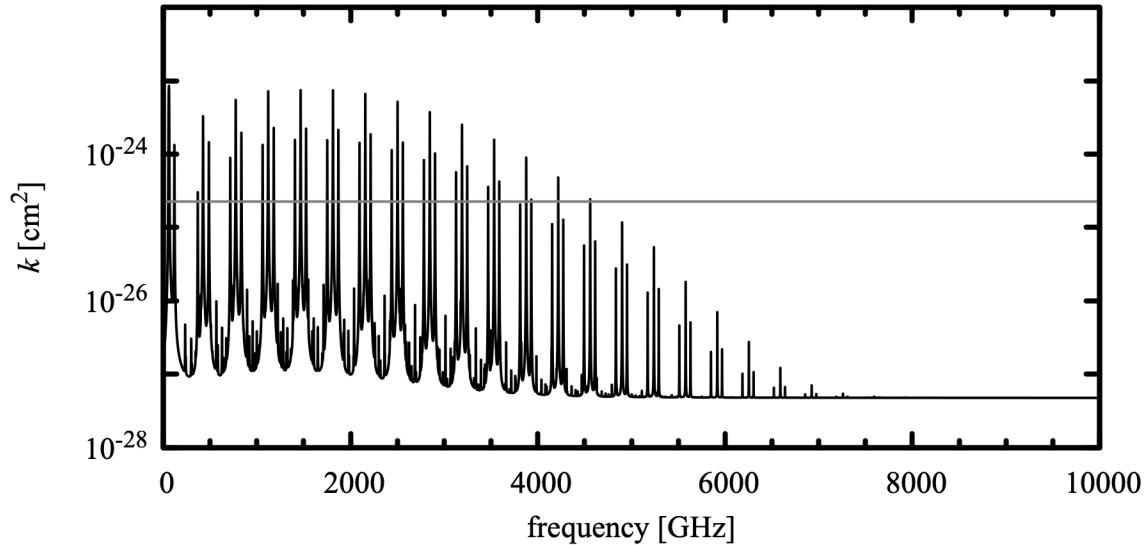


Figure 3.3 Absorption coefficient for oxygen. $P=500 \text{ mbar}$, $T=260 \text{ K}$. The gray line corresponds to $\tau=1$ for typical zenith sea level column density $n=4.4 \cdot 10^{24} \text{ cm}^{-2}$.

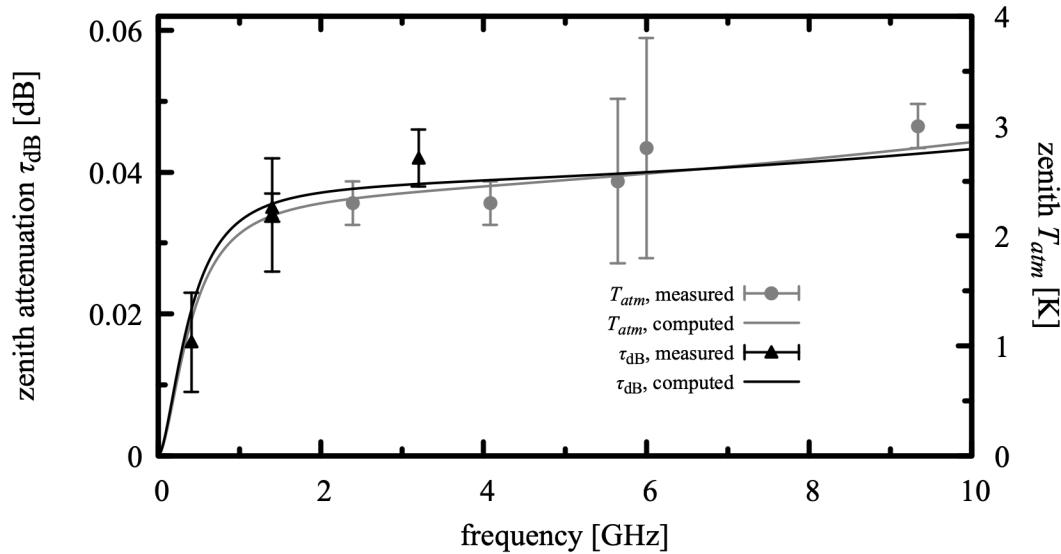


Figure 3.4 Output of *am*, for a sea level mid-latitude (40° N) dry atmosphere based on the CIRA-86 annual average climatology, compared with various zenith opacity and atmospheric brightness temperature measurements collected in Table 2 of Danese and Partridge (1989).

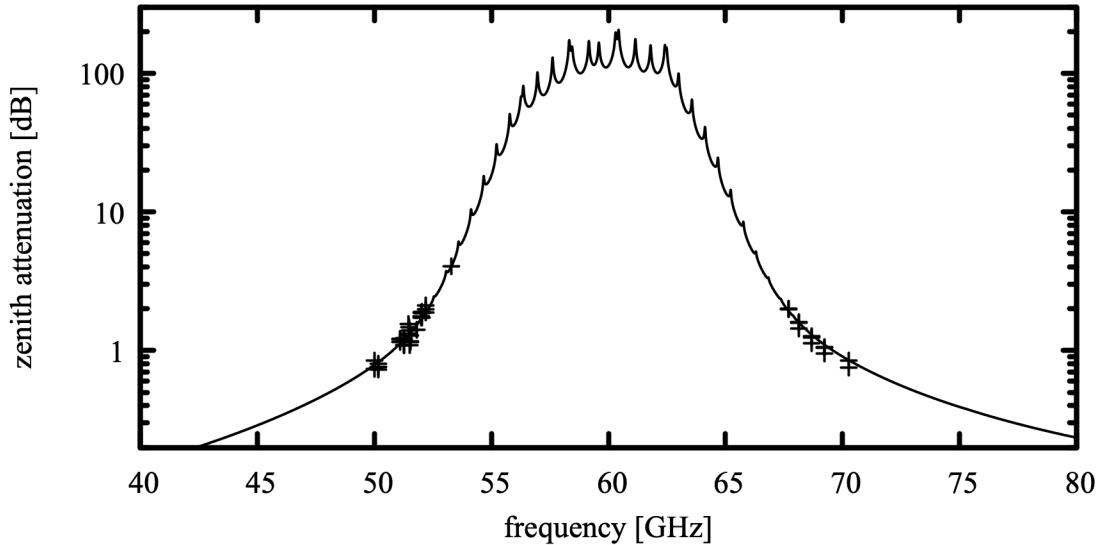


Figure 3.5 Output of *am*, for a dry atmosphere based on the CIRA-86 climatology for 35° N latitude in July, compared with zenith dry air attenuation measured at 2.7 km altitude from Reber (1972).

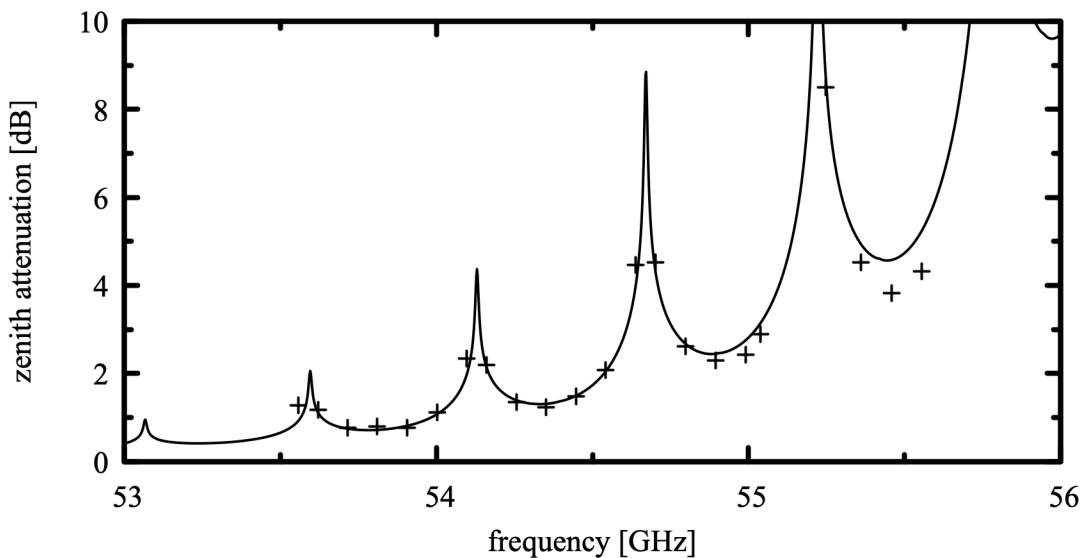


Figure 3.6 Output of *am*, for a dry atmosphere based on the CIRA-86 climatology for 35° N latitude in July, compared with zenith dry air attenuation measured at 12.2 km altitude from Reber (1972).

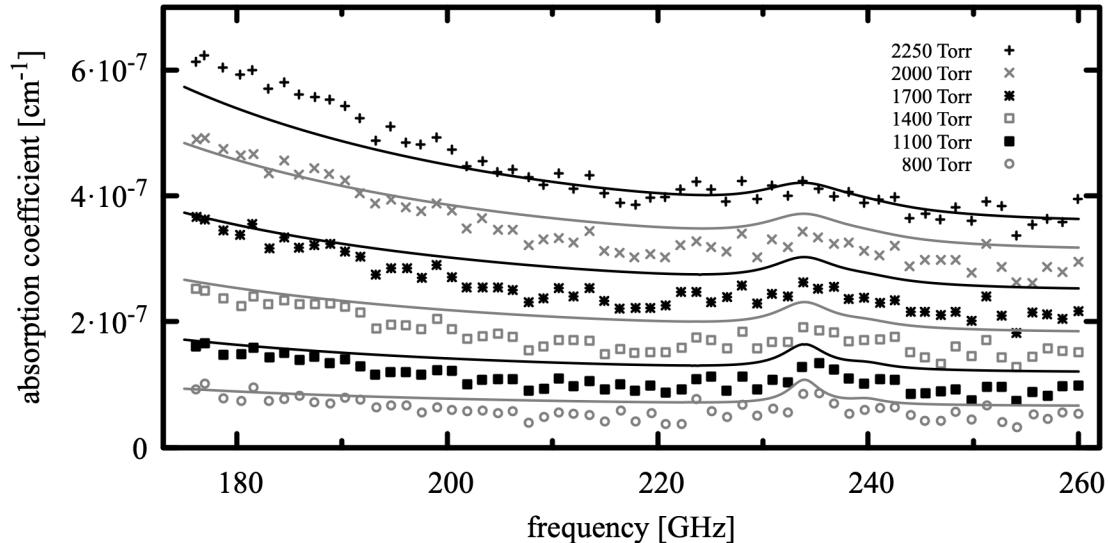


Figure 3.7 Volume absorption coefficient computed by *am* for pure O₂ at 297 K and various pressures, compared with cavity ringdown measurements of Meshkov and De Lucia (2007).

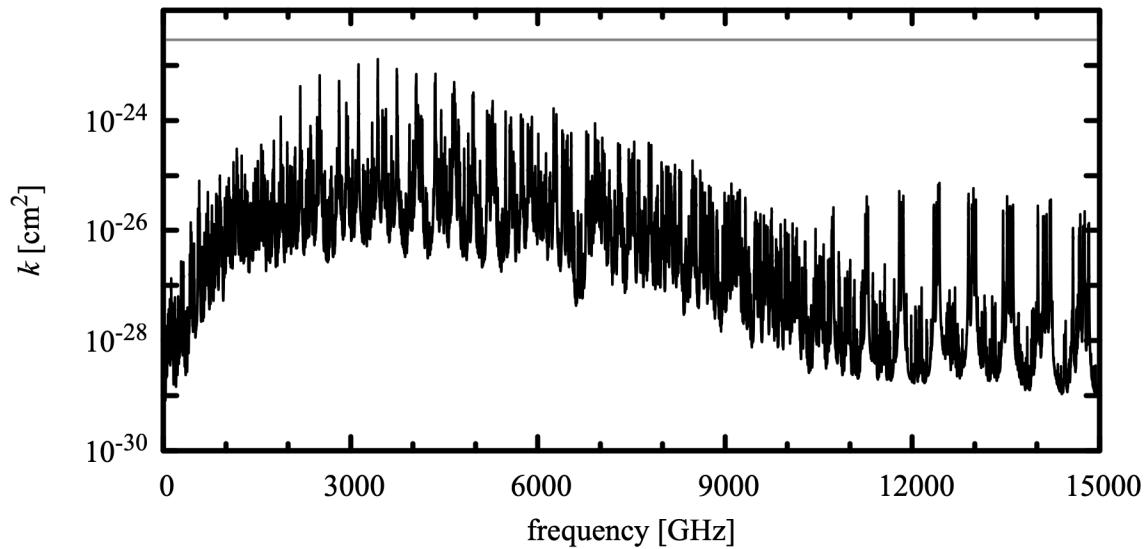


Figure 3.8 Absorption coefficient for methane. $P = 500 \text{ mbar}$, $T = 260 \text{ K}$, gross line shape. The gray line corresponds to $\tau = 0.001$ for nominal sea level zenith column density $n = 3.4 \cdot 10^{19} \text{ cm}^{-2}$.

absorption coefficient in cm^{-1} is found by computing the opacity of combined O_2 line-by-line and collision-induced absorption for a path length of 1 cm, as follows:

```
f 175 GHz 260 GHz 0.1 GHz
output f tau

selfbroad_vmr_tol 0.0

layer
P %1 Torr
T 297 K
h 1 cm
column o2 vmr 1.0
column o2o2 vmr 1.0
```

Because *am* does not have a “pure O_2 ” column type, here we needed to combine the o2 and o2o2 column types for line-by-line and collision-induced absorption, respectively. This construct will cause *am* to issue a warning about the sum of mixing ratios exceeding unity, but the computation will proceed correctly based upon the specified total layer pressure, as explained in Section 1.1. Figure 3.7 shows moderately good agreement between model and measurement at these super-atmospheric pressures, sufficient for terrestrial applications as the very weak O_2 absorption in this spectral interval is already very small relative to water vapor absorption in the troposphere.

3.4.4 Methane (column type ch4)

The absorption coefficient for methane, column type `ch4`, is plotted in Figure 3.8, for a background air pressure $P=500\text{ mbar}$ and temperature $T=260\text{ K}$. k has been computed with the `Gross` line shape. Ideally, CH_4 would have no dipole moment on account of its tetrahedral symmetry, but centrifugal distortion of the rotating molecule produces a small electric dipole moment. The optical depth of even the strongest submillimeter lines of CH_4 is quite small over non-limb atmospheric paths—the gray line at $k=2.9 \cdot 10^{-23}\text{ cm}^2$ corresponds to optical depth $\tau=0.001$ for a typical sea level zenith column density $n=3.4 \cdot 10^{19}\text{ cm}^{-2}$.

3.4.5 Carbon Monoxide (column type co)

The absorption coefficient for carbon monoxide, column type `co`, is plotted in Figure 3.9, for a background air pressure $P=500\text{ mbar}$ and temperature $T=260\text{ K}$, computed with the `Gross` line shape. The gray line at $k=6.8 \cdot 10^{-21}\text{ cm}^2$ corresponds to optical depth $\tau=0.01$ for a typical sea level zenith column density $n=1.5 \cdot 10^{18}\text{ cm}^{-2}$.

3.4.6 Carbon Dioxide (column type co2)

The principal isotopologue of CO_2 , $^{12}\text{C}^{16}\text{O}_2$, is a symmetrical linear molecule with no electric dipole moment and hence no rotational spectrum. The well-known vibrational bands near $15\text{ }\mu\text{m}$ wavelength (20 THz frequency) lie mostly outside the 15 THz upper cutoff frequency of the current version of *am*. The rotational bands of the most abundant asymmetric isotopologues, $^{16}\text{O}^{12}\text{C}^{18}\text{O}$ and $^{16}\text{O}^{12}\text{C}^{17}\text{O}$ are very weak over atmospheric paths, as demonstrated in Figure 3.10,

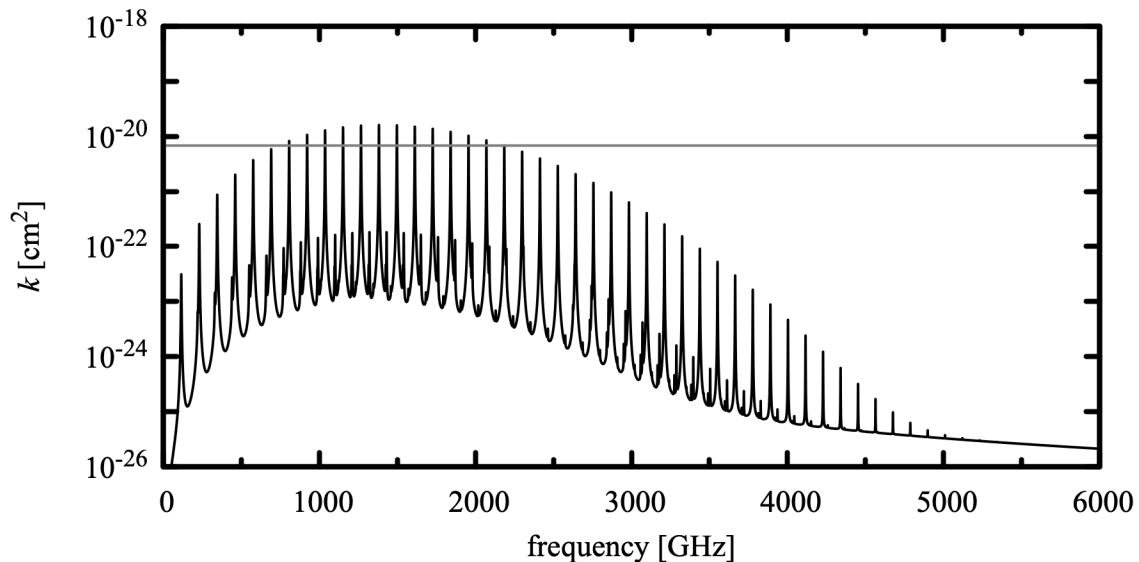


Figure 3.9 Absorption coefficient for carbon monoxide. $P=500$ mbar, $T=260$ K, Gross line shape. The gray line corresponds to $\tau=0.01$ for a nominal sea level zenith column density $n=1.5 \cdot 10^{18} \text{ cm}^{-2}$.

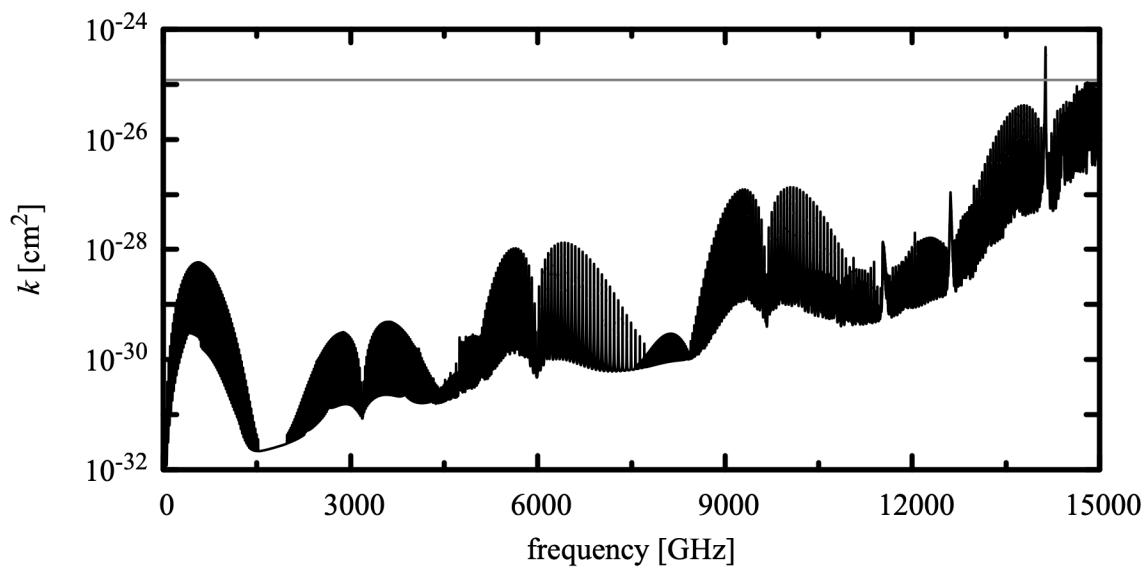


Figure 3.10 Absorption coefficient for carbon dioxide. $P=500$ mbar, $T=260$ K, Gross line shape. The gray line corresponds to $\tau=0.001$ for a nominal sea level zenith column density $n=8.3 \cdot 10^{21} \text{ cm}^{-2}$.

which is a plot of the absorption coefficient for CO_2 for a background air pressure $P=500\text{ mbar}$ and temperature $T=260\text{ K}$, computed with the `gross` line shape. The gray line at $k=1.21 \cdot 10^{-25}\text{ cm}^2$ corresponds to optical depth $\tau=0.001$ for a typical sea level zenith column density $n=8.3 \cdot 10^{21}\text{ cm}^{-2}$.

3.4.7 Nitrous Oxide (column type `n2o`)

N_2O is a linear molecule possessing a significant dipole moment by virtue of its asymmetric structure, N-N-O. Since HITRAN2016 (Gordon et al. 2017), the pure rotation band cutoff has been 2170 GHz, which is a reasonable level for atmospheric paths. However, like OCS discussed below, N_2O is also useful as a laboratory gas cell test molecule for submillimeter receivers operating at higher frequencies, so the HITRAN data have been merged with those for all N_2O isotopologues in the JPL catalog (Pickett, et al. 1998). In this merge process, the JPL line frequencies have been preferred for lines which appear in both catalogs; otherwise the HITRAN data are used. Pressure broadening parameters for lines which only appear in the JPL catalog have been generated using the polynomial fits from Rothman et al. (2005).

The absorption coefficient for nitrous oxide is plotted in Figure 3.11, for a background air pressure $P=500\text{ mbar}$ and temperature $T=260\text{ K}$, computed with the `gross` line shape. The gray line at $k=1.35 \cdot 10^{-21}\text{ cm}^2$ corresponds to optical depth $\tau=0.01$ for a typical sea level zenith column density $n=7.4 \cdot 10^{18}\text{ cm}^{-2}$.

3.4.8 Carbonyl Sulfide (column type `ocs`)

The many closely-spaced strong lines of the OCS molecule make it valuable for use in a gas cell as a thermal line source for testing receivers in the submillimeter band. For the main $^{16}\text{O}^{12}\text{C}^{32}\text{S}$ isotopologue the rotational constant $B_0=6.081\text{ GHz}$, resulting in a rotational spectrum with lines at 12 GHz intervals. In both the HITRAN and JPL catalogs, the upper cutoff frequency for the pure rotation band is 1200 GHz. OCS remains useful as a gas cell test molecule at frequencies somewhat above this, so for $^{16}\text{O}^{12}\text{C}^{32}\text{S}$ the HITRAN catalog data have been extended to $j=125$ (1522 GHz) using the rotational constants from Golubiatnikov et al. (2005), and the Padé approximants for the pressure broadening parameters given by Koshelev and Tretyakov (2009).

The absorption coefficient for OCS (for all significant isotopologues) plotted in Figure 3.12 has been computed with a self-broadened `gross` line shape at a pressure $P=5\text{ mbar}$, and temperature $T=290\text{ K}$. Only the pure rotation band has been plotted; a handful of vibration-rotation lines between 14.7 THz and *am's* frequency cutoff at 15 THz are not shown. The gray line at $k=8 \cdot 10^{-20}\text{ cm}^2$ corresponds to optical depth $\tau=1$ for a column density of $1.25 \cdot 10^{19}\text{ cm}^2$, which would be obtained in a gas cell 1 m long at this pressure and temperature. The gas cell can be modeled as follows:

```
f 0 THz 15 THz 0.01 GHz
output f k
```

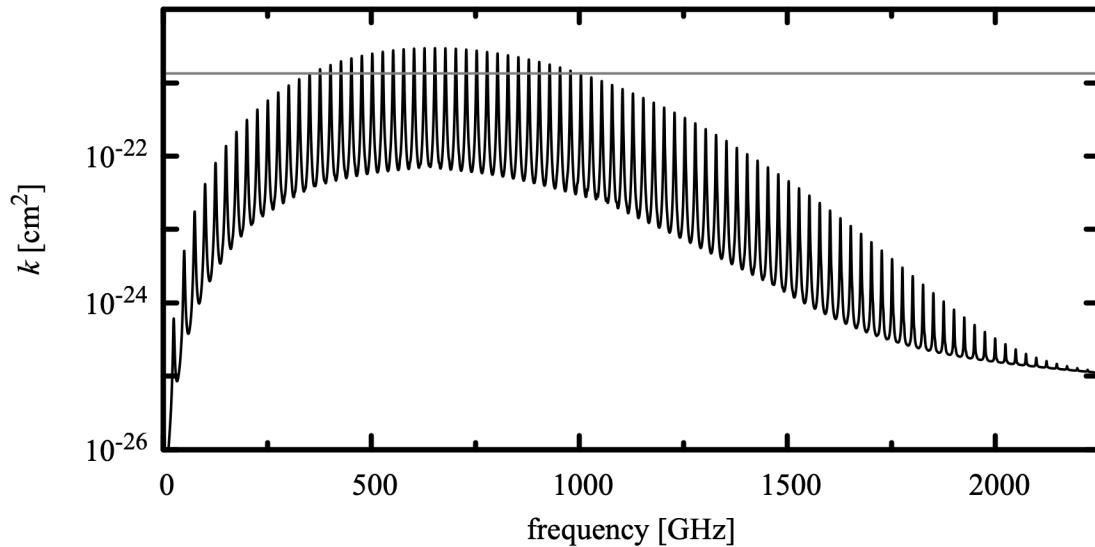


Figure 3.11 Absorption coefficient for nitrous oxide. $P=500$ mbar, $T=260$ K, Gross line shape. The gray line corresponds to $\tau=0.01$ for a nominal sea level zenith column density $n=7.4 \cdot 10^{18} \text{ cm}^{-2}$.

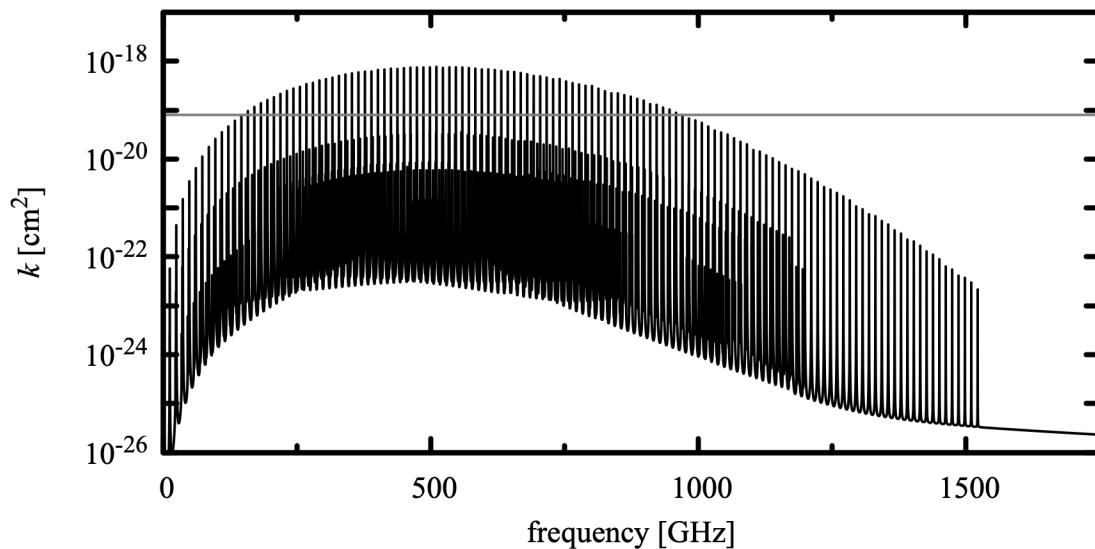


Figure 3.12 Absorption coefficient for carbonyl sulfide. $P=5$ mbar, $T=290$ K, Gross line shape. The gray line corresponds to $\tau=1$ for a column density $n=1.25 \cdot 10^{19} \text{ cm}^{-2}$, as would be obtained in a laboratory gas cell 1 m long at this pressure and temperature.

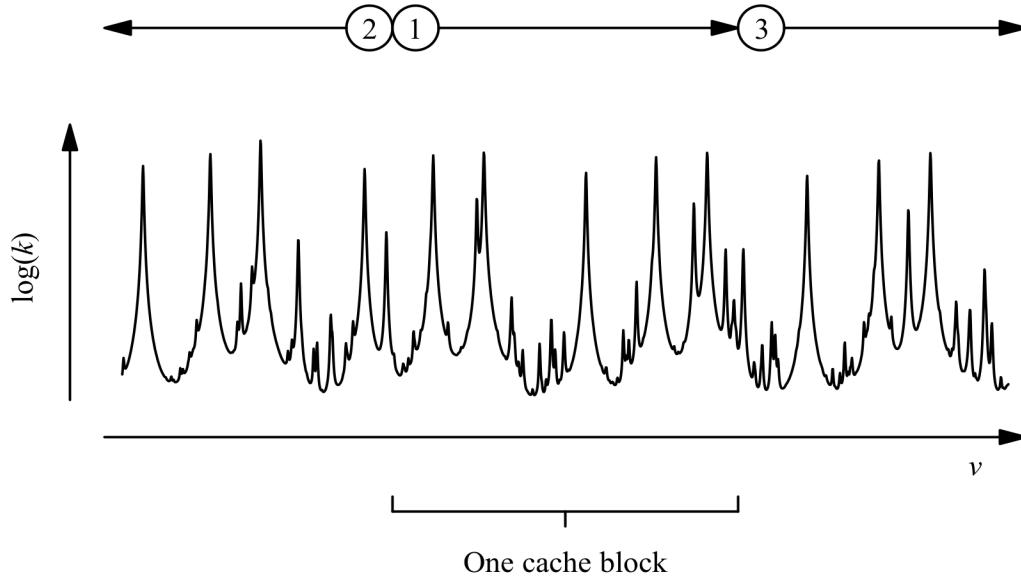


Figure 3.13 The order in which catalog lines are added when computing one block of a line-by-line absorption coefficient array.

```

layer
P 5 hPa
T 290 K
h 1 m
lineshape Gross strict_self_broadening ocs
column ocs vmr 1.0

```

For *am* version 9.2 and later, the `strict_self_broadening` keyword is no longer really necessary in this example—with the default setting of `selfbroad_vmr_tol`, pressure broadening will be computed as the self-broadened width for pure OCS.

3.5 A note on computation

When computing a spectral absorption coefficient array k , *am* divides the frequency grid into blocks such that all arrays involved in the computation of each block will fit into a processor's level 1 data cache. Multiple processors can work independently without conflict because they write to different blocks of the k array, and because the shared access to line catalog data is read-only. Even for a single processor, cache blocking results in faster computation by avoiding cache thrashing on k as each spectral line is added to the sum.

For each cache block, the line-by-line sum is done in three stages, as illustrated in

Figure 3.13. First, all in-band lines for the current cache block are added. Second, the high frequency wings of lines at frequencies below the frequency range of the current cache block are added in reverse frequency order. Third, the low-frequency wings of lines above the current cache block are added, in ascending frequency order. This breaks the usual rule of summing a series from smaller elements to larger to minimize roundoff errors, but the summation is done in this order to support the `tol` feature as discussed below. Moreover, roundoff error is not significant here, given the limited size of the line catalogs and the use of double-precision arithmetic.

The `tol` setting for line-by-line computations was introduced in the last chapter, in Section 2.3. If `tol` is non-zero, then during the second and third stages of the line-by-line sum, the relative contribution $\delta k/k$ made to each in-band element of k by an out-of-band line is monitored as each line is added. If, for a given line, the maximum $\delta k/k$ is less than tol/N , where N is the total number of lines in the catalog, the line strength $S(T)$ of the current line becomes a strength threshold below which subsequent lines at larger detuning will be ignored. This algorithm reliably ensures that the fractional error $|\Delta k/k|$ due to skipped catalog lines is less than `tol`. In the context of this discussion, it is important to remember that any line catalog already has a built-in line strength cutoff which is based on the anticipated temperature range over which the catalog will be used.

As noted in Section 3.3.2, for coupled lines the coupled line shape is only meaningful in the context of the complete coupled band. For this reason, the `tol` setting affects the behavior of the line-by-line summation for uncoupled lines only.

Besides the potential savings in compute time, setting a non-zero value for `tol` makes sense on physical grounds, given the limitations of collisional line shape models in the far line wings discussed earlier in this chapter. However, because monitoring $\delta k/k$ introduces extra computational overhead in the line-by-line sum, there may not always be a significant performance benefit. The performance gain is generally largest when the model frequency grid is short compared with the frequency span of the entire line catalog, or when computationally expensive Doppler-broadened line shapes are involved. Timing examples are presented in Appendix A.

References

COSPAR (Committee on Space Research) 1986, “The COSPAR International Atmosphere (CIRA-86) [Internet].” NCAS British Atmospheric Data Centre, 2006-2011. Available from <http://catalogue.ceda.ac.uk/uuid/4996e5b2f53ce0b1f2072adadaeda262>.

L. Danese and R. B. Partridge 1989, “Atmospheric emission models: Confrontation between observational data and predictions in the 2.5–300 GHz frequency range.” Ap. J. 342:604.

R. R. Gamache, B. Vispoel, M. Rey, A. Nikitin, V. Tyuterev, O. Egorov, I. E. Gordon, V. Boudon,

and V. B. Bastien 2021. “Total Internal Partition Sums for the HITRAN2020 database.” *J. Quant. Spectrosc. Radiat. Transfer* 271:107713.

G. Yu. Golubiatnikov, A.V. Lapinov, A. Guarneri, and R. Knöchel 2005, “Precise Lamb-dip measurements of millimeter and submillimeter wave rotational transitions of $^{16}\text{O}^{12}\text{C}^{32}\text{S}$.” *J. Mol. Spec.* 234:190.

I.E. Gordon, L.S. Rothman, R.J. Hargreaves, R. Hashemi, E.V. Karlovets, F.M. Skinner, E.K. Conway, C. Hill, R.V. Kochanov, Y. Tan, P. Wcisło, A.A. Finenko, K. Nelson, P.F. Bernath, M. Birk, V. Boudon, A. Campargue, K.V. Chance, A. Coustenis, B.J. Drouin, J.-M. Flaud, R.R. Gamache, J.T. Hodges, D. Jacquemart, E.J. Mlawer, A.V. Nikitin, V.I. Perevalov, M. Rotger, J. Tennyson, G.C. Toon, H. Tran, V.G. Tyuterev, E.M. Adkins, A. Baker, A. Barbe, E. Canè, A.G. Császár, A. Dudaryonok, O. Egorov, A.J. Fleisher, H. Fleurbaey, A. Foltynowicz, T. Furtenbacher, J.J. Harrison, J.-M. Hartmann, V.-M. Horneman, X. Huang, T. Karman, J. Karns, S. Kassi, I. Kleiner, V. Kofman, F. Kwabia-Tchana, N.N. Lavrentievac, T.J. LeeG, D.A. Longp, A.A. Lukashevskaya, O.M. Lyulin, V.Yu. Makhnev, W. Matt, S.T. Massie, M. Melosso, S.N. Mikhailenko, D. Mondelain, H.S.P. Müller, O.V. Naumenko, A. Perrin, O.L. Polyansky, E. Raddaoui, P.L. Raston, Z.D. Reed, M. Rey, C. Richard, R. Tóbiás, I. Sadiek, D.W. Schwenke, E. Starikova, K. Sung, F. Tamassia, S.A. Tashkun, J. Vander Auwera, I.A. Vasilenko, A.A. Vigasin, G.L. Villanueva, B. Vispoel, G. Wagner, A. Yachmenev, and S.N. Yurchenko 2022, “The HITRAN2020 molecular spectroscopic database.” *J. Quant. Spectrosc. Radiat. Transfer* 277:107949.

I.E. Gordon, L.S. Rothman, C. Hill, R.V. Kochanov, Y. Tan, P.F. Bernath, M. Birk, V. Boudon, A. Campargue, K.V. Chance, B.J. Drouin, J.-M. Flaud, R.R. Gamache, J.T. Hodges, D. Jacquemart, V.I. Perevalov, A. Perrin, K.P. Shine, M.-A.H. Smith, J. Tennyson, G.C. Toon, H. Tran, V.G. Tyuterev, A. Barbe, A.G. Császár, V.M. Devi, T. Furtenbacher, J.J. Harrison, J.-M. Hartmann, A. Jolly, T.J. Johnson, T. Karman, I. Kleiner, A.A. Kyuberis, J. Loos, O.M. Lyulin, S.T. Massie, S.N. Mikhailenko, N. Moazzen-Ahmadi, H.S.P. Müller, O.V. Naumenko, A.V. Nikitin, O.L. Polyansky, M. Rey, M. Rotger, S.W. Sharpe, K. Sung, E. Starikova, S.A. Tashkun, J. Vander Auwera, G. Wagner, J. Wilzewski, P. Wcisło, S. Yu, and E.J. Zak 2017, “The HITRAN2016 molecular spectroscopic database.” *J. Quant. Spectrosc. Radiat. Transfer* 203:3.

R. G. Gordon 1967, “On the Pressure Broadening of Molecular Multiplet Spectra.” *J. Chem. Phys.* 46:448.

E. P. Gross 1955, “Shape of Collision-Broadened Spectral Lines,” *Phys. Rev.* 97:395.

J.-M. Hartmann, C. Boulet, and Daniel Robert 2021, *Collisional Effects on Molecular Spectra*, Second Edition. Elsevier.

R. J. Hill 1986, "Water vapor absorption line shape comparison using the 22-GHz line: The Van Vleck-Weisskopf shape affirmed," *Radio Science* 21:447.

D.L. Huber and J.H. Van Vleck 1966, "The Role of Boltzmann Factors in Line Shape," *Rev. Modern Physics* 38:187.

J. F. Kielkopf 1973 "New approximation to the Voigt function with applications to spectral-line profile analysis," *J. Opt. Soc. Am.* 63:987.

M.A. Koshelev and M. Yu. Tretyakov 2009, "Collisional broadening and shifting of OCS rotational spectrum lines." *J. Quant. Spectrosc. Radiat. Transfer* 110:118.

K. S. Lam 1977, "Application of pressure broadening theory to the calculation of atmospheric oxygen and water vapor microwave absorption." *J. Quant. Spectrosc. Radiat. Transfer* 17:351.

H. J. Liebe 1985, "An updated model for millimeter wave propagation in moist air." *Radio Science* 20:1069.

H.A. Lorentz 1916, *The Theory of Electrons and its Applications to the Phenomena of Light and Radiant Heat*, Second Edition. B. G. Teubner; G.E. Stechert & Co. (Leipzig; New York); reprinted in 1952 by Dover Publications, New York. The oscillation amplitude of a bound electron driven by a sinusoidal electric field and subject to random interrupting collisions is discussed in note 57 to section 120.

D. S. Makarov, M. Yu. Tretyakov, and P. W. Rosenkranz 2011, "60-GHz oxygen band: Precise experimental profiles and extended absorption modeling in a wide temperature range." *J. Quant. Spectrosc. Radiat. Transfer* 112:1420.

A. I. Meshkov and F. C. De Lucia 2007, "Laboratory measurements of dry air atmospheric absorption with a millimeter wave cavity ringdown spectrometer." *J. Quant. Spectrosc. Radiat. Transfer* 108:256.

H.M. Pickett, R.L. Poynter, E.A. Cohen, J.L. Delitsky, J.C. Pearson, and H. S. P. Muller 1998, "Submillimeter, Millimeter, and Microwave Spectral Line Catalog." *J. Quant. Spectrosc. Radiat. Transfer* 60:883.

E. E. Reber 1972, "Absorption of the 4- to 6-Millimeter Wavelength Band in the Atmosphere." *J. Geophysical Research* 77:3831.

F. Reif 1965 "Fundamentals of Statistical and Thermal Physics." McGraw Hill.

P. W. Rosenkranz 1975, "Shape of the 5 mm Oxygen Band in the Atmosphere." IEEE Trans. Ant. Prop. AP-23:498.

P.W. Rosenkranz 1988, "Interference Coefficients for Overlapping Oxygen Lines in Air." J. Quant. Spectrosc. Radiat. Transfer 39:287.

L.S. Rothman, C.P. Rinsland, A. Goldman, S.T. Massie, D.P. Edwards, J.-M. Flaud, A. Perrin, C. Camy-Peyret, V. Dana, J.-Y. Mandin, J. Schroeder, A. McCann, R.R. Gamache, R.B. Wattson, K. Yoshino, K.V. Chance, K.W. Jucks, L.R. Brown, V. Nemtchinov, and P. Varanasi 1998, "The HITRAN molecular spectroscopic database and HAWKS (HITRAN Atmospheric Workstation): 1996 Edition." J. Quant. Spectrosc. Radiat. Transfer 60:665.

L. S. Rothman, D. Jacquemart, A. Barbe, D. Chris Benner, M. Birk, L.R. Brown, M.R. Carleer, C. Chackerian Jr., K. Chance, L.H. Coudert, V. Dana, V.M. Devi, J.-M. Flaud, R.R. Gamache, A. Goldman, J.-M. Hartmann, K.W. Jucks, A.G. Maki, J.-Y. Mandin, S.T. Massie, J. Orphal, A. Perrin, C.P. Rinsland, M.A.H. Smith, J. Tennyson, R.N. Tolchenov, R.A. Toth, J. Vander Auwera, P. Varanasi, and G. Wagner 2005, "The HITRAN 2004 molecular spectroscopic database." J. Quant. Spectrosc. Radiat. Transfer 96:139.

F. Schreier 1992 "The Voigt and complex error function: a comparison of computational methods." J. Quant. Spectrosc. Radiat. Transfer 48:743.

E. W. Smith 1981, "Absorption and dispersion in the O₂ microwave spectrum at atmospheric pressures." J. Chem. Phys. 74:6658.

M. Yu. Tretyakov, et al. 2005, "60-GHz oxygen band: precise broadening and central frequencies of fine-structure lines, absolute absorption profile at atmospheric pressure, and revision of mixing coefficients." J. Molecular Spectroscopy 231:1

J. H. Van Vleck and V. F. Weisskopf 1945, "On the Shape of Collision-Broadened Lines," Rev. Modern Physics 17:227.

J. H. Van Vleck 1947, "The Absorption of Microwaves by Oxygen." Phys. Rev. 71:413.

Chapter 4: Continuum Absorption

The title of this chapter is a catch-all for processes that yield smooth spectra lacking sharp features. One such process is collision-induced absorption (CIA), in which transitions that are forbidden in the isolated molecule acquire an interaction-induced transition moment during collisions. The short duration of these collisions gives rise to correspondingly broad transitions, which merge into an unresolved continuum. *am* includes models for CIA involving N₂ and O₂.

For molecules with an allowed spectrum, as in the preceding chapter, close collision processes which affect the far wings of collision-broadened spectral lines are not modeled by the classic collisional line shapes. This is a particularly significant problem for water vapor. The standard treatment of the problem is to apply a line wing cutoff in the line-by-line absorption coefficient computation, and then to add back a slowly-varying, empirically-modeled continuum. In *am*, this is done using an implementation of the MT_CKD water vapor continuum model.

Cloud liquid water and ice are another source of absorption that varies smoothly with frequency. *am* includes models for absorption by liquid water droplets and ice particles in the Rayleigh limit, when the particle size is small compared with the wavelength and the absorption depends on column mass density independent of particle size. These models are most suitable for non-precipitating clouds at microwave through millimeter wavelengths.

Finally, for any absorption processes which can be approximated as spectrally flat over some band of interest, *am* includes simple gray attenuation models which depend only on airmass factor or zenith angle. These are described in the last section of this chapter.

4.1 Collision-induced absorption

In the line-by-line absorption discussed in the previous chapter, the coupling between molecular transitions and the radiation field was via the electric or magnetic dipole moment of an isolated molecule. In contrast, in collision-induced absorption, the coupling arises from a transient dipole moment induced during a collision. Among the possible induction mechanisms are polarization at close range by the permanent multipole field surrounding a molecular collision

partner, overlap of electronic wave functions, dispersion and exchange interactions, and molecular frame distortion. The induced dipole moment only exists for a time on the order of the duration of collision time scale τ_c^* mentioned in Section 3.2.1, so the corresponding translational line width is broad, of order 1 THz. Consequently, a collision-induced rotational band will typically appear as an unresolved envelope, except for very light molecules such as H₂ having widely spaced rotational lines.

A very good general reference on collision-induced absorption is the text by Frommhold (1993). Poll and Hunt (1976) has a useful physical discussion, and also introduces the standard spherical harmonic expansion parameters used in the CIA literature.

4.1.1 Binary absorption coefficients

At sufficiently low gas density, we need only consider CIA arising from binary collisions, ignoring higher-order collisions involving three or more molecules. This approximation holds, for example, throughout the Earth's atmosphere. The time-averaged density of colliding pairs, and hence the volume absorption coefficient, is proportional to the product of the densities of the two colliding species. That is,

$$\alpha \propto n_a n_b k^{(b)} , \quad (4.1)$$

where α [cm⁻¹] is an extinction coefficient, n_a and n_b are the volume densities [cm⁻³] of the two species (which are the same for homogeneous collisions), and $k^{(b)}$ [cm⁵] is a binary molecular absorption coefficient. If we neglect double transitions, in which both molecules undergo internal transitions during a collision, then we can regard one species as the “active” species, and the other as a background perturber. Writing n and n_p for the respective densities of the active species and the perturber, the opacity over a path of length z is

$$\tau = \alpha z = z n n_p k^{(b)} . \quad (4.2)$$

am models are organized around zenith column densities normal to a layer, and the angle of propagation θ_z through the layer. Since $z = h \sec(\theta_z)$, where h is the layer thickness, and the zenith column density $N = hn$, we can rewrite Eq. 4.2 in a form which fits a standard *am* column specification,

$$\tau = N \cdot n_p \cdot \sec(\theta_z) \cdot k^{(b)}(\nu, T) . \quad (4.3)$$

The rules by which *am* computes the background perturber density n_p are similar to those used to compute pressure broadening, and are described in Section 4.1.4 below.

4.1.2 N₂-N₂ CIA

At submillimeter wavelengths, the most important CIA band in the terrestrial atmosphere is the rototranslational band associated with N₂-N₂ collisions. For this band, the dominant dipole induction mechanism is polarization of one molecule by the quadrupole and hexadecapole (16-pole) components of the electric field surrounding its collision partner. A thorough theoretical analysis of this band was carried out by Borysow and Frommhold (1986). A key goal of that

paper was to compute the far high-frequency wing of the band as accurately as possible at low temperatures, to support analysis of measurements of the atmosphere of Titan made by the IRIS spectrometer on the *Voyager* spacecraft. To that end, the analysis included minor contributions from overlap induction, from double transitions associated with quadrupolar induction, and from transitions between bound states of N₂-N₂ dimers. The implementation of the theory in *am*, described below, neglects these minor contributions, since their effect is insignificant for terrestrial propagation paths. Moreover, currently available experimental data do not significantly constrain the theory for these minor terms.

In Borysow and Frommhold (1986), Eq. 3, the complete rototranslational binary volume absorption coefficient is written as an incoherent sum over the various dipole induction mechanisms, and over the initial and final rotational angular momenta of the colliding molecules. The induction mechanisms are characterized by the four parameters $L \Lambda \lambda_1 \lambda_2$, which are the indices in a spherical harmonic expansion of the dipole induction interaction for a pair of colliding diatomic molecules (Poll and Hunt 1976). (Note that there are three pairs θ, ϕ of angular coordinates involved—the orientations of each of the two molecules, and the orientation of the intermolecular axis. If one pair of angular coordinates is taken to define an arbitrary spatial direction, then there are four remaining angular degrees of freedom governing the interaction.) For *am*, only the dominant quadrupolar, $(L \Lambda \lambda_1 \lambda_2) = (3\ 2\ 2\ 0), (3\ 2\ 0\ 2)$, and hexadecapolar $(L \Lambda \lambda_1 \lambda_2) = (5\ 4\ 4\ 0), (5\ 4\ 0\ 4)$ induction mechanisms are considered, and double transitions are neglected. That is,

$$k^{(b)}(\nu, T) = k_{3220}^{(b)}(\nu, T) + k_{5440}^{(b)}(\nu, T), \quad (4.4)$$

where

$$k_{3220}^{(b)}(\nu, T) = \nu(1 - e^{-h\nu/kT}) S_{3220}(T) \sum_{jj'} (2j+1) P_j(T) C(j\ 2\ j'; 00)^2 \Gamma_{BC}(\nu - \nu_{jj'}, T) \quad (4.5)$$

and

$$k_{5440}^{(b)}(\nu, T) = \nu(1 - e^{-h\nu/kT}) S_{5440}(T) \sum_{jj'} (2j+1) P_j(T) C(j\ 4\ j'; 00)^2 \Gamma_{EBC}(\nu - \nu_{jj'}, T) \quad (4.6)$$

are the quadrupolar and hexadecapolar terms, respectively. Here, $S(T)$ as defined by Borysow and Frommhold has been multiplied by $4\pi^3/3\hbar c$, giving $S(T)$ the dimensions [cm⁵] of a molecular binary absorption coefficient. Since we consider only single transitions, the sum over the states of the second molecule, which is unity, has been suppressed, and the symmetrical and equivalent sums for $(L \Lambda \lambda_1 \lambda_2) = (3\ 2\ 0\ 2)$ and $(L \Lambda \lambda_1 \lambda_2) = (5\ 4\ 0\ 4)$ have been absorbed into the band strengths $S_{3220}(T)$ and $S_{5440}(T)$, respectively.

Eqs. 4.5 and 4.6 have the structure of a line-by-line sum, where the line shape is the translational spectral function $\Gamma_{BC}(\nu, T)$ or $\Gamma_{EBC}(\nu, T)$, described below, shifted by the rotational transition frequencies $\nu_{jj'}$, where

$$\nu_{jj'} = (E_j - E_{j'})/h \quad (4.7)$$

and

$$E_j/h = B j(j+1) + D j^2(j+1)^2. \quad (4.8)$$

The rotational constants B and D for N_2 are $B=59.6459(4)\text{GHz}$ and $D=1.727(8)\cdot 10^{-4}\text{GHz}$ (Bendtsen 1974). The state probabilities P_j are

$$P_j(T) = g_j \cdot e^{-E_j/kT} / Q(T), \quad (4.9)$$

where the partition function $Q(T)$ is

$$Q(T) = \sum_j (2j+1) g_j e^{-E_j/kT}, \quad (4.10)$$

and the statistical weights g_j for $^{14}\text{N}_2$ are

$$g_j = \begin{cases} 6 & j \text{ even} \\ 3 & j \text{ odd} \end{cases}. \quad (4.11)$$

In the *am* code, $Q(T)$ is approximated by a three-term expansion in kT/hB ; details can be found in the source file `cia.c`. For the sums in Eq. 4.5 or Eq 4.6, an upper cutoff at $j=30$, corresponding to an excitation temperature $E_j/k=2650\text{K}$, gives sufficient accuracy for terrestrial temperatures.

$C(j\lambda j'; 00)$ is a Clebsch-Gordan coefficient. For quadrupolar induction, $\lambda=2$, this imposes a selection rule restricting the sum to rotational transitions $j'=j\pm 2$, and transitions with $j'=j$ corresponding to scattering with no change in rotational state other than a possible change in orientation. For hexadecapolar induction, $\lambda=4$, the selection rule is $j'=j, j\pm 2, j\pm 4$.

The spectral functions $\Gamma_{BC}(\nu, T)$ and $\Gamma_{EBC}(\nu, T)$ are analytic profiles which accurately model translational line shapes computed by Borysow and Frommhold quantum mechanically, starting from a theoretical intermolecular interaction potential. These model profiles provide a practical means of reproducing the quantum computations with minimal computational effort, using only a small number of temperature-dependent parameters. $\Gamma_{BC}(\nu, T)$ is the Birnbaum and Cohen (1976) profile,

$$\Gamma_{BC}(\nu, T) = \frac{\tau_1}{\pi} [e^{\tau_2/\tau_1 + 2\pi\tau_0\nu}] \frac{z K_1(z)}{1 + (2\pi\nu)^2 \tau_1^2}, \quad (4.12)$$

where $\tau_0=\hbar/2kT$, τ_1 and τ_2 are characteristic time constants having empirically-determined temperature dependence, $K_1(z)$ is a modified Bessel function of order 1, and

$$z = \frac{(\tau_0^2 + \tau_2^2)^{1/2}}{\tau_1} [1 + (2\pi\nu)^2 \tau_1^2]^{1/2}. \quad (4.13)$$

Borysow and Frommhold carried out the quantum translational line shape computation over a range of temperatures, 50 K–300 K, and fitted the temperature dependences of τ_1 , τ_2 , and $S_{3220}(T)$ to an empirical function of the form

$$y = A e^{Bx+Cx^2}, \quad (4.14)$$

where y is one of the parameters τ_1 , τ_2 , or $S(T)$, and $x=\log(T)$. The constants A , B , and C for each parameter are tabulated in Borysow and Frommhold (1986) Table 5. In the *am* code, the

A , B , and C constants for the strength $S_{3220}(T)$ of the quadrupolar induction term are modified from their original tabulated values to improve the fit to the published experimental data referenced below. In part, this makes up for neglecting the N₂-N₂ dimer contribution included in the Borysow–Frommhold fits.

The translational line shape for hexadecapolar induction is the extended Birnbaum-Cohen profile,

$$\Gamma_{EBC}(\nu, T) = \frac{1}{1+\epsilon} [\Gamma_{BC}(\nu, T) + \epsilon \Gamma_{K0}(\nu, T)], \quad (4.15)$$

where

$$\Gamma_{K0}(\nu, T) = \frac{\tau'_1}{\pi} [e^{(\tau'_1/\tau'_2 + 2\pi\tau_0\nu)}] K_0(z'), \quad (4.16)$$

and

$$z' = \frac{(\tau_0^2 + \tau'_2)^{1/2}}{\tau'_2} [1 + (2\pi\nu)^2 \tau'_2]^{1/2}. \quad (4.17)$$

The parameter ϵ controlling the admixture of the BC and $K0$ profiles was fixed at $\epsilon=0.3$, and the A , B , C , parameters for the temperature dependences of the parameters $S_{5440}(T)$, τ_1 , τ_2 , τ'_1 , and τ'_2 were fitted by Borysow and Frommhold to their quantum profiles over two temperature ranges, 50 K–140 K, and 140 K–300 K. For hexadecapolar induction, the fitted parameters from Borysow and Frommhold (1986) are used unmodified in the *am* code.

In Figs. 4.1 and 4.2, the N₂-N₂ binary absorption coefficient spectra computed with *am* are compared with the experimental measurements of Dore and Filabozzi (1987), Dagg et al. (1985), and Stone et al. (1984). Absorption coefficients are obtained as a program output from *am* by using the **k** output type. For example, the data for these figures were generated by the shell script:

```
for T in 93 126 149 179 212 228.3 253.5 277.5 297.5 322.6 343.0; do
    am - <<- end > am_n2n2_${T}K.out
    f 0 GHz 15000 GHz 50 GHz
    output f GHz k cm^5
    layer
    P 1 atm
    T ${T} K
    col n2n2 1e25 cm^-2
end
done
```

Here, since the only output is **k**, the specification of the layer pressure and the column density are superfluous, but they are needed to satisfy the *am* configuration file syntax. Although the temperature dependence of the A , B , C parameters for the translational line shapes was fitted by Borysow and Frommhold up to a maximum temperature $T=300$ K, the experimental data are nevertheless reproduced accurately up to the highest experimentally-measured temperature, which is 343 K.

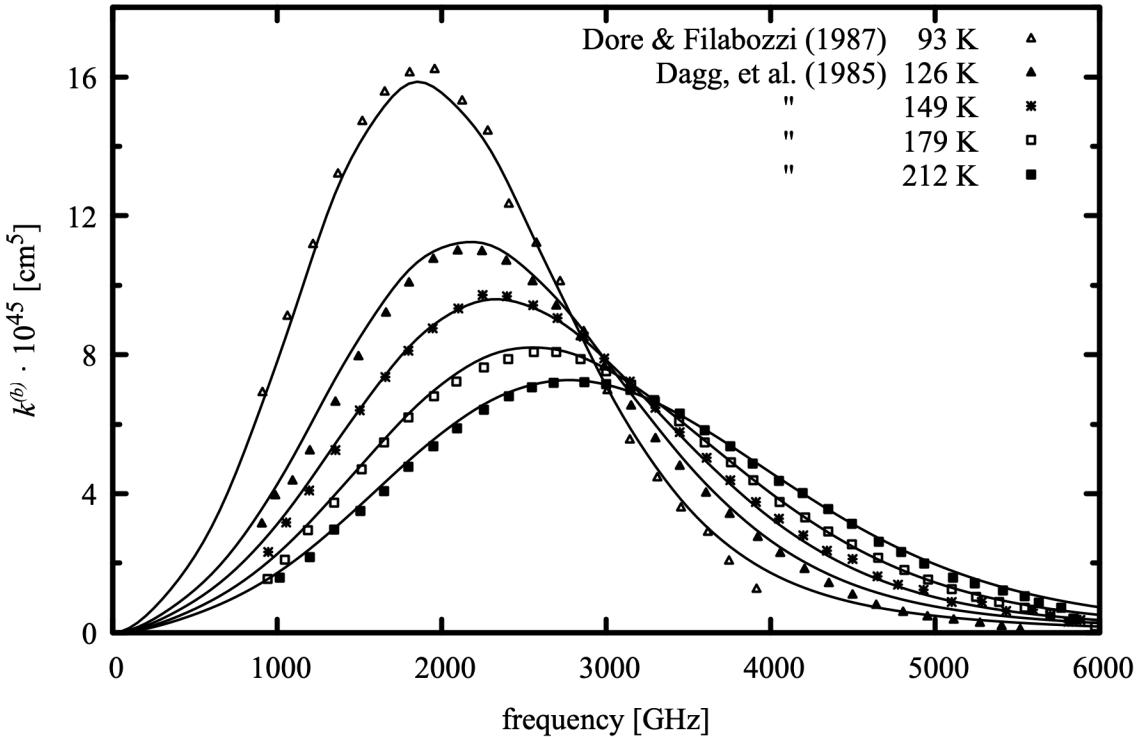


Figure 4.1 Comparison of the $\text{N}_2\text{-N}_2$ binary absorption coefficient computed by *am* (solid lines) with laboratory measurements by Dore and Filabozzi (1987) and Dagg et al. (1985), 93 K–212 K.

4.1.3 $\text{N}_2\text{-O}_2$, $\text{O}_2\text{-N}_2$, and $\text{O}_2\text{-O}_2$ CIA

After $\text{N}_2\text{-N}_2$, the most significant sources of submillimeter CIA in the Earth's atmosphere are, in descending importance, $\text{N}_2\text{-O}_2$, $\text{O}_2\text{-N}_2$, and $\text{O}_2\text{-O}_2$. Again considering only collisions in which at most one molecule undergoes an internal transition, the distinction between $\text{N}_2\text{-O}_2$ and $\text{O}_2\text{-N}_2$ is that when writing $\text{N}_2\text{-O}_2$, N_2 is considered to be the active molecule with O_2 being the polarizable background perturber, and vice-versa for $\text{O}_2\text{-N}_2$. Following Boissoles et al. (2003), $\text{N}_2\text{-O}_2$, $\text{O}_2\text{-N}_2$, and $\text{O}_2\text{-O}_2$ CIA are computed in *am* analogously to $\text{N}_2\text{-N}_2$, with adjustments to account for the different multipole moments and polarizability of O_2 versus N_2 , and for the different underlying rotational spectrum of O_2 .

The binary absorption coefficients for the mixed collisions, $\text{N}_2\text{-O}_2$ and $\text{O}_2\text{-N}_2$, are simply scaled from $\text{N}_2\text{-N}_2$ and $\text{O}_2\text{-O}_2$, respectively, using the ratios for the band-integrated intensities computed by Boissoles et al.,

$$k_{\text{N}_2\text{-O}_2}^{(b)} = 1.143 k_{\text{N}_2\text{-N}_2}^{(b)} \quad (4.18)$$

and

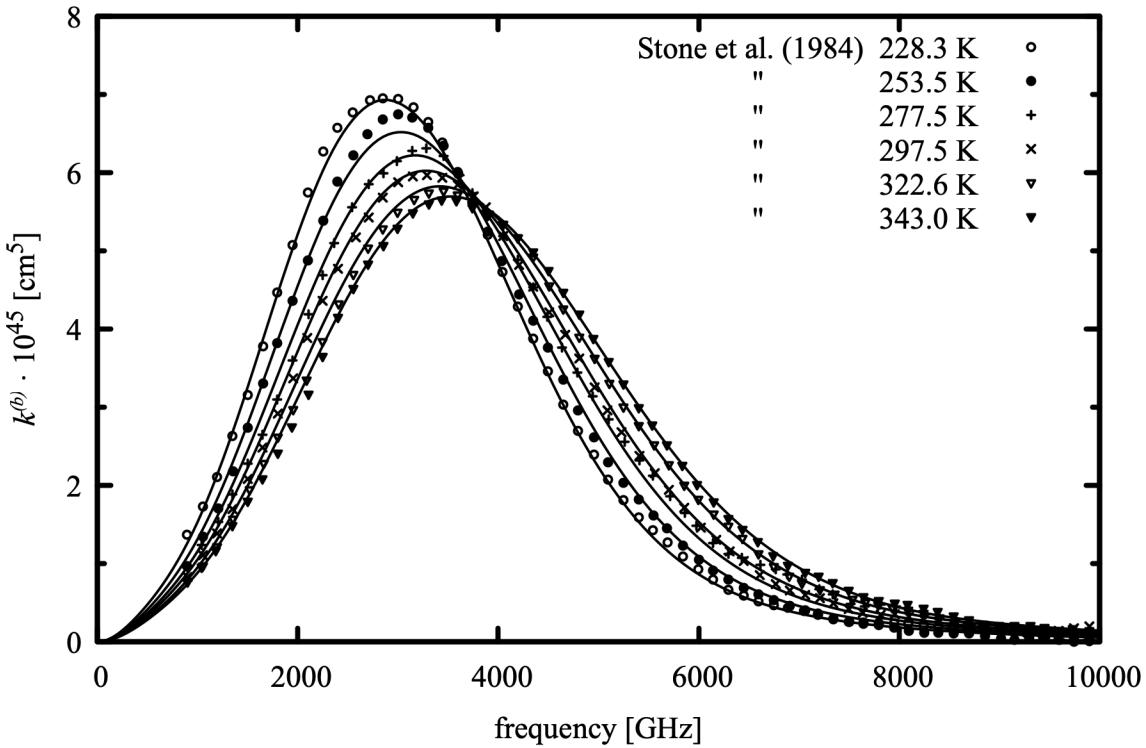


Figure 4.2 Comparison of the N₂-N₂ binary absorption coefficient computed by *am* (solid lines) with laboratory measurements by Stone, et al. (1984), 228 K–343 K

$$k_{O_2-N_2}^{(b)} = 0.822 k_{O_2-O_2}^{(b)}. \quad (4.19)$$

The O₂-O₂ binary absorption coefficient is computed as for N₂-N₂, above, accounting for the different underlying rotational spectrum, and modifying the strength, but not the shape or the temperature dependence, of the translational spectral functions. For ¹⁶O₂, in which only odd-*j* rotational states occur, the statistical weights *g_j* are

$$g_j = \begin{cases} 0 & j \text{ even} \\ 1 & j \text{ odd} \end{cases}. \quad (4.20)$$

The rotational constants for ¹⁶O₂ are *B*=43.1004438(9)GHz and *D*=1.45115(40)·10⁻⁴GHz (Golubiatnikov and Krupnov 2003). The sums, Eq. 4.5 and Eq 4.6, are evaluated up to *j*=35, corresponding to an excitation temperature *T*=2600 K. The spectral functions are scaled following Boissoles et al. (2003), with slight adjustments to minimize the squared residuals relative to the 300 K data discussed below. The strength *S*₃₂₂₀(*T*) for quadrupole induction is scaled by a factor of 0.0837 relative to N₂-N₂, *S*₅₄₄₀(*T*) for hexadecapole induction is scaled by 2.28, and the admixture coefficient *ε* in the EBC line profile is set to 0.52. The band strengths reflect the fact that the quadrupole moment of O₂ is about four times smaller than that of N₂,

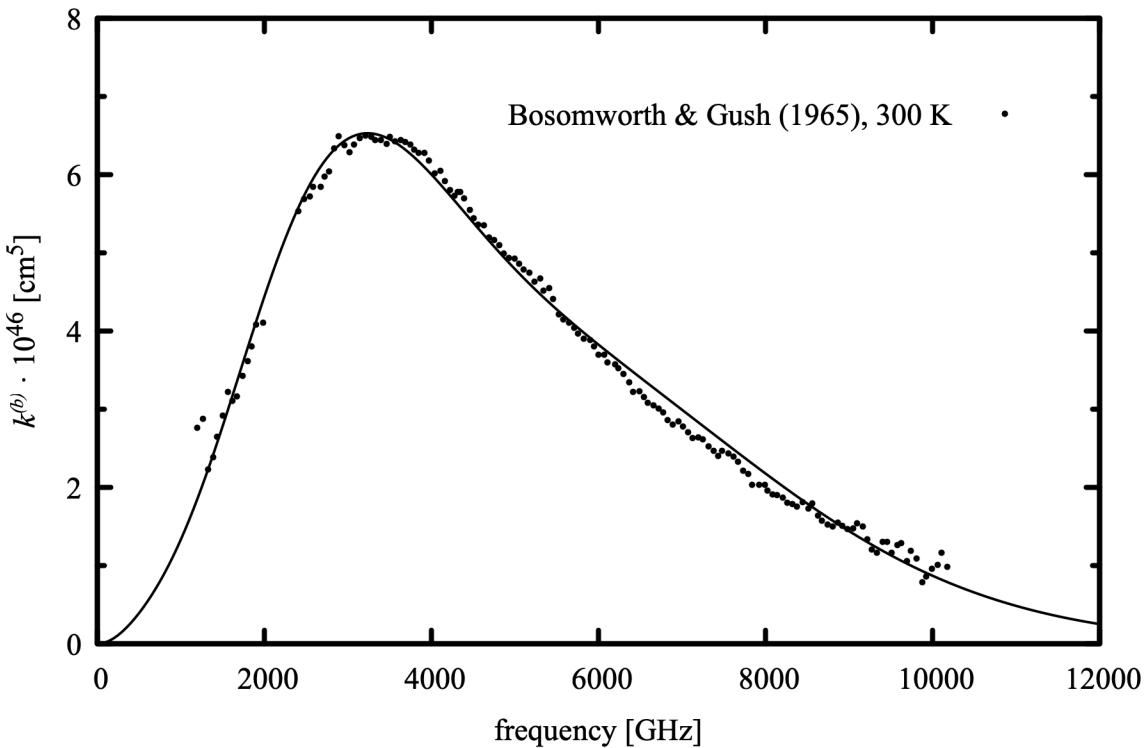


Figure 4.3 Comparison of the O₂-O₂ absorption coefficient computed by *am* at 300 K (solid line) with laboratory measurements by Bosomworth and Gush (1965) (symbols).

whereas the hexadecapole moments and polarizabilities for these molecules are comparable. In addition, a small contribution attributed to hexadecapolar induction via the anisotropic component of the O₂ polarizability (cf. Boissoles et al. Fig. 4) has been folded into the adjustment of $S_{5440}(T)$ in *am*.

The only available laboratory measurement for the O₂-O₂ rototranslational band is that of Bosomworth and Gush (1965). Since this measurement was made at a single temperature (300 K), it is not possible to validate the temperature dependence of the computed absorption coefficient, which simply mimics that of N₂-N₂. Figure 4.3 compares these laboratory data with the absorption coefficient computed by *am* at 300 K.

4.1.4 Using CIA column types

Four column types, in addition to the `dry_air` composite column type, are available for including N₂ and O₂ rototranslational CIA in *am* models. The `n2n2` and `o2o2` column types represent homogeneous N₂-N₂ or O₂-O₂ CIA. The column types `n2air` and `o2air` represent combined N₂-N₂ + N₂-O₂ (N₂ active) and O₂-O₂ + O₂-N₂ CIA (O₂ active), respectively. Each column definition included in a layer in an *am* model configuration file has a single field for

column density or mixing ratio, which controls the column density N of the active molecule in Eq. 4.3. For `n2n2` and `o2o2`, the perturber density n_p is computed from the layer pressure and a mixing ratio, exactly as it would be for the case of pressure self-broadening with the `strict_self_broadening` flag set. That is, if the mixing ratio is defined by the user, either explicitly or by specifying N directly on a layer with defined thickness h then that mixing ratio will also be used for computing n_p . Otherwise, the dry air default mixing ratio will be used.

For `n2air` and `o2air`, there are two background perturber densities which must be computed, for the homogeneous and the inhomogeneous perturber. In this case, the mixing ratio for the homogeneous perturber follows that of the active molecule as for `n2n2` and `o2o2`, but the mixing ratio for the inhomogeneous perturber is fixed at the default value for dry air.

The CIA column types are mainly useful for isolating the dry CIA contribution to the total atmospheric absorption, or for comparison with laboratory data as in the examples above. Most often, in applications intended to model the complete atmospheric absorption, the composite `dry_air` or `dry_air_std` column types, which incorporate both `n2air` and `o2air` with default mixing ratios, will be used. Details are in Appendix B.

4.2 H₂O continuum absorption

The absorption of water vapor cannot be accurately modeled as a line-by-line sum of dipole-allowed transitions using simple impact-broadened collisional line shapes. The difference between such a line-by-line computation and experimental measurements is a slowly varying residual spectrum called the H₂O continuum. This definition implies that the continuum spectrum is a function of the line shape used for the line-by-line computation. Nevertheless, in general the continuum is found to be well-modeled as a binary process depending on the product of water vapor density and the density of a background perturber. As with pressure broadening, there is a natural distinction between the *self-induced continuum* arising from H₂O-H₂O collisions, and the *air-induced* or *foreign-induced* continuum, arising from collisions between H₂O and the constituents of dry air, mainly N₂. With this distinction, the total volume absorption coefficient for water vapor may be written as the sum of line-by-line and continuum contributions as

$$\alpha_{H_2O}(\nu) = n_{H_2O} k_{lines}(\nu) + n_{H_2O}^2 k_{self}^{(b)}(\nu) + n_{H_2O} n_{air} k_{air}^{(b)}(\nu) \quad (4.21)$$

where n_{H_2O} and n_{air} are the densities [cm⁻³] of water vapor and air, $k_{lines}(\nu)$ [cm²] is the line-by-line absorption coefficient, and $k_{self}^{(b)}(\nu)$ and $k_{air}^{(b)}(\nu)$ [cm⁵] are binary absorption coefficients for the self- and air-induced continua, respectively. Experimentally, the self- and air-induced continua are observed to have significantly different characteristics (Burch and Gryvnak 1980). The self-induced continuum is stronger, and exhibits a strong negative temperature dependence, compared with a weak temperature dependence for the air-induced continuum.

The degree to which various physical processes contribute to the continuum is an unresolved and historically controversial subject. There is a large literature—a good starting point with a comprehensive reference list is Appendix V.A in Hartmann et al. (2021); the collected workshop proceedings in Deepak, et al. (1980) contain much of the founding work on

modern approaches to the subject. An important recent advance in measurement and modeling of the submillimeter and far-infrared water vapor continuum is Odintsova, et al. (2022), which includes a good discussion and references assessing the relative contributions of different processes contributing to the self- and foreign-induced continuum, and prospects for further theoretical development.

Because of the theoretical difficulties, a practical approach is needed. The most widely adopted has been the semi-empirical CKD model of Clough, Kneizys, and Davies (1989), and its successive revisions through the current MT_CKD model (Mlawer et al. 1998; Mlawer et al. 1999; Clough et al. 2005; Mlawer et al. 2012; Mlawer et al. 2019; Mlawer et al. 2023). MT_CKD is defined relative to a line-by-line absorption coefficient computed using a Van-Vleck Huber line shape, with the line wings truncated at 25 cm^{-1} (750 GHz), and a constant offset subtracted to bring the line shape function to zero at the truncation point. (This is the vvH_750 line shape defined in Table 3.1.) In the original CKD continuum formulation, the continuum was computed by multiplying the residual line wings and constant offset under the line core by a correction factor (χ -factor), having a functional form motivated on physical grounds, and applied uniformly to every line. The binary density dependence of the continuum absorption coefficient emerged naturally from the density dependence of the pressure-broadened line wings. The χ -factor was defined with a small number of adjustable parameters which were separately fitted to available experimental data for the self-continuum at various temperatures, and to the relatively temperature-independent foreign continuum. The form given to the χ -factor in the CKD model was devised so as to apply a super-Lorentzian increase to the line profile, of unspecified physical origin, in the neighborhood of the truncation point, and a sub-Lorentzian Gaussian roll-off, associated with a finite collision duration, in the far line wings.

In the MT_CKD model, the finite duration of collision is still modeled with a sub-Lorentzian roll-off in the allowed line wings, but the χ -factor never exceeds unity. Instead, the super-Lorentzian component of the continuum is represented by a more physically-motivated model, suggestive of a CIA spectrum associated with a transient modification of the dipole moment of the free molecule. This spectrum is computed as a line-by-line sum of a very broad line shape centered on the allowed H_2O transition frequencies, and scaled to the allowed line strengths. Compared with the original CKD formulation, the MT_CKD model has more free parameters, which were needed to accurately fit the growing base of experimental data.

The MT_CKD continuum is periodically adjusted to account for new laboratory and field measurements and is archived and distributed by the Radiative Transfer Working Group at AER, Inc. at https://github.com/AER-RC/MT_CKD. The water vapor continuum in the current version (14.0) of *am* is MT_CKD version 4.2 (April 2024). The data and accompanying copyright and usage information in the *am* source file `mt_ckd.c` have been extracted directly from this distribution. Note that the self-induced continuum has an explicit spectral temperature dependence, needed to fit the observed data as noted above, whereas the air-induced continuum has minimal temperature dependence apart from the radiation term $\nu \tanh[\hbar \nu/(2kT)]$. See Mlawer et al. (2023) for details.

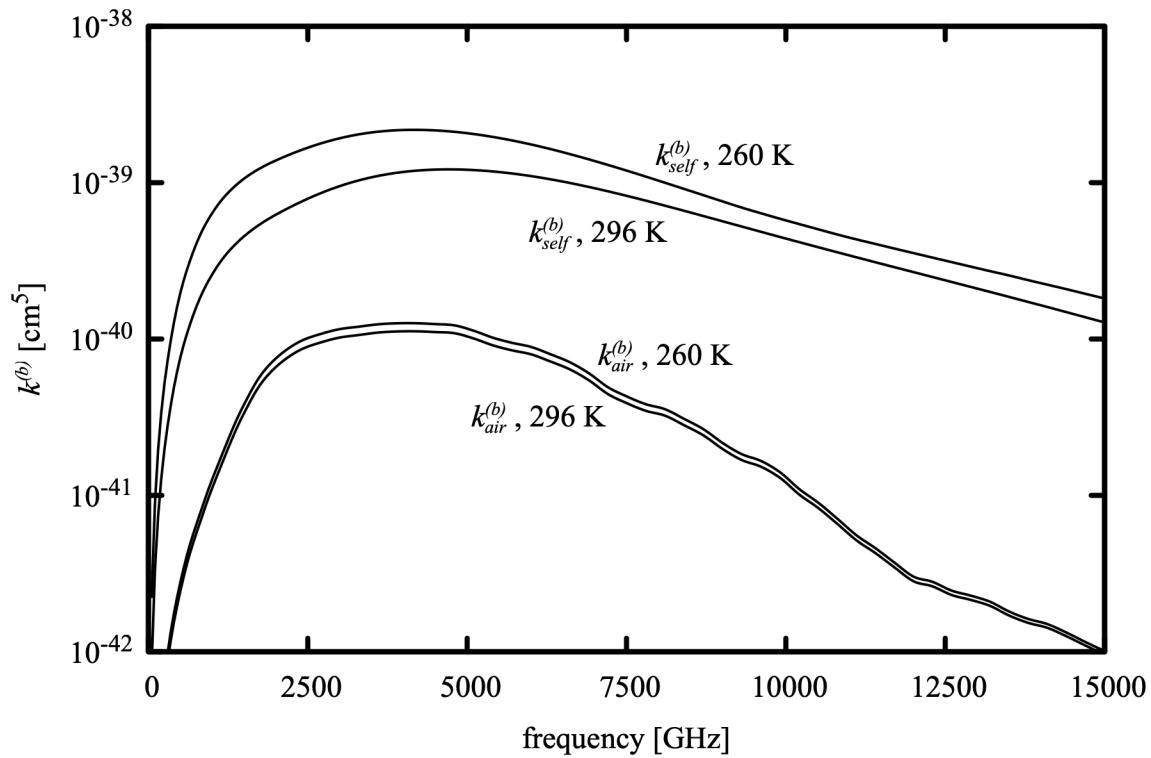


Figure 4.4 Binary absorption coefficients for the MT_CKD water vapor continuum at 260 K and 296 K. The weaker temperature dependence of the air-induced continuum arises only from the radiation term $\nu \tanh[h\nu/(2kT)]$

Fig. 4.4 compares the binary absorption coefficient spectra for the air- and self-induced continua at $T=260 \text{ K}$ and $T=296 \text{ K}$. Although in general $k_{\text{air}}^{(b)} \ll k_{\text{self}}^{(b)}$, it should be remembered that because the H₂O mixing ratio in the Earth's atmosphere ranges from around 0.05 downwards, it is possible for either the air-induced or self-induced term in Eq. 4.21 to be the larger of the two over the range of conditions encountered in the atmosphere.

4.2.1 Water vapor in *am* models

As noted previously, *am* model configuration files define the radiatively active species on a model layer by their zenith column densities—either directly, or implicitly via a mixing ratio. Background perturber densities for pressure broadening or for binary absorption processes are computed from the product of the perturber mixing ratio and the total gas density at the layer midpoint pressure and temperature. Multiplying both sides of Eq. 4.21 by the path length $h \sec(\theta_z)$ across a layer yields the layer H₂O opacity in a form fitting this description, in terms of the zenith column density $N_{\text{H}_2\text{O}}$, and background perturber densities $n_{\text{H}_2\text{O}}$ and n_{air} for the self-

and air-induced continua, respectively:

$$\tau_{H_2O}(\nu) = N_{H_2O} \sec(\theta_z) [k_{lines}(\nu) + n_{H_2O} k_{self}^{(b)}(\nu) + n_{air} k_{air}^{(b)}(\nu)]. \quad (4.22)$$

The `h2o` column type includes all of the terms in Eq. 4.22, with n_{H_2O} and n_{air} computed as they would be for pressure broadening. However, recall that in non-hydrostatic models, specification of the layer thickness is optional. If the layer thickness is omitted, as in

```
.
layer
P 1 atm
T 290 K
column h2o 1 mm_pwv
.
```

then *am* will not be able to derive an H₂O mixing ratio for the layer. The program will default to an effective H₂O mixing ratio of zero, and issue a warning to the user that the self-continuum is being neglected. In this case, the air-induced continuum will be computed assuming a total dry air mixing ratio of unity.

Three additional column types, `h2o_lines`, `h2o_self_continuum`, and `h2o_air_continuum` are available. These can be used to isolate the corresponding terms in Eq. 4.22, for example to assess their relative contributions to a computed spectrum. A simple way to do this in a model containing many layers is to make the water vapor column type a command-line parameter. For instance, if a file `model.amc` contains layer specifications of the form

```
.
.
layer
Pbase 600 mbar
Tbase 261 K
column dry_air hydrostatic
column %1 hydrostatic 2.5e-3

layer
Pbase 800 mbar
Tbase 275 K
column dry_air hydrostatic
column %1 hydrostatic 8.0e-3
.
```

then the command

```
am model.amc h2o > model.out
```

would compute the model using the complete H₂O opacity on each layer, whereas the command

```
am model.amc h2o_lines > model_h2o_lines_only.out
```

would have the effect of “turning off” the continuum.

The line-by-line column type `h2o_lines` and the component $k_{lines}(\nu)$ of the `h2o` column type both follow the default pressure broadening rule in *am*, whereby self-broadening is controlled by the parameter `selfbroad_vmr_tol` unless the `strict_self_broadening` flag has been set on the layer, using a statement such as

```
lineshape vvH_750 strict_self_broadening h2o
```

See Appendix B for details. Although the `lineshape` statement can also be used to override the default `vvH_750` line shape, it is not recommended to do so since the continuum definition is tied to this line shape.

4.3 Liquid water and ice absorption in clouds

Absorption and scattering by clouds and precipitation is, in general, a complicated subject. Widely ranging distributions of particle size, shape, and orientation are involved, which must be treated to varying degrees of approximation in scattering theory. However, a great simplification is possible in the Rayleigh limit, when the liquid droplets or ice particles are small compared with the wavelength, and have compact shapes or random orientation so that they can be approximated as spherical. Moreover, this limit encompasses many practical situations in microwave and millimeter wave remote sensing and radio astronomy. This is especially true for liquid water in non-precipitating clouds—suspended cloud droplets are spherical with typical dimensions of order 10 μm . Suspended ice occurs in a wider range of size and shape (or *habit*), and drag forces can cause coherent particle orientation leading to polarization effects, but the Rayleigh approximation is nonetheless useful over a more restricted set of conditions. A good comprehensive reference for related issues of cloud microphysics is Pruppacher and Klett (2010).

As discussed below, when the Rayleigh approximation is valid, cloud opacity is proportional to column mass density, and does not depend on particle size. The column mass density is normally referred to as *liquid water path* (LWP) for water clouds, and *ice water path* (IWP) for ice clouds. LWP for non-precipitating clouds typically falls in the range 0 – 0.5 kg m^{-2} . Note that 1 kg m^{-2} corresponds to the same molecular column density as 1 mm precipitable water vapor—clouds usually account for a small fraction of the total moisture in an atmospheric column.

4.3.1 Absorption and scattering in the Rayleigh limit

For spherical particles of radius r , we can define the size parameter

$$x = \frac{2\pi r}{\lambda} = \frac{2\pi r v}{c}, \quad (4.23)$$

Where x is the optical phase shift in radians corresponding to the particle radius. The Rayleigh approximation holds when this phase shift is negligible across the particle both externally and internally, so that the fields in the neighborhood of the particle can be treated statically. That is, when

$$\begin{aligned} x &\ll 1 \\ |n| x &\ll 1 \end{aligned} \quad (4.24)$$

Here, $n=\sqrt{\epsilon'+i\epsilon''}$ is the complex refractive index of the particle material, and ϵ' and ϵ'' are the corresponding real and imaginary parts of the complex permittivity ϵ . In the Rayleigh limit, the absorption and scattering efficiencies are (Mishchenko, Travis, and Lacis 2005):

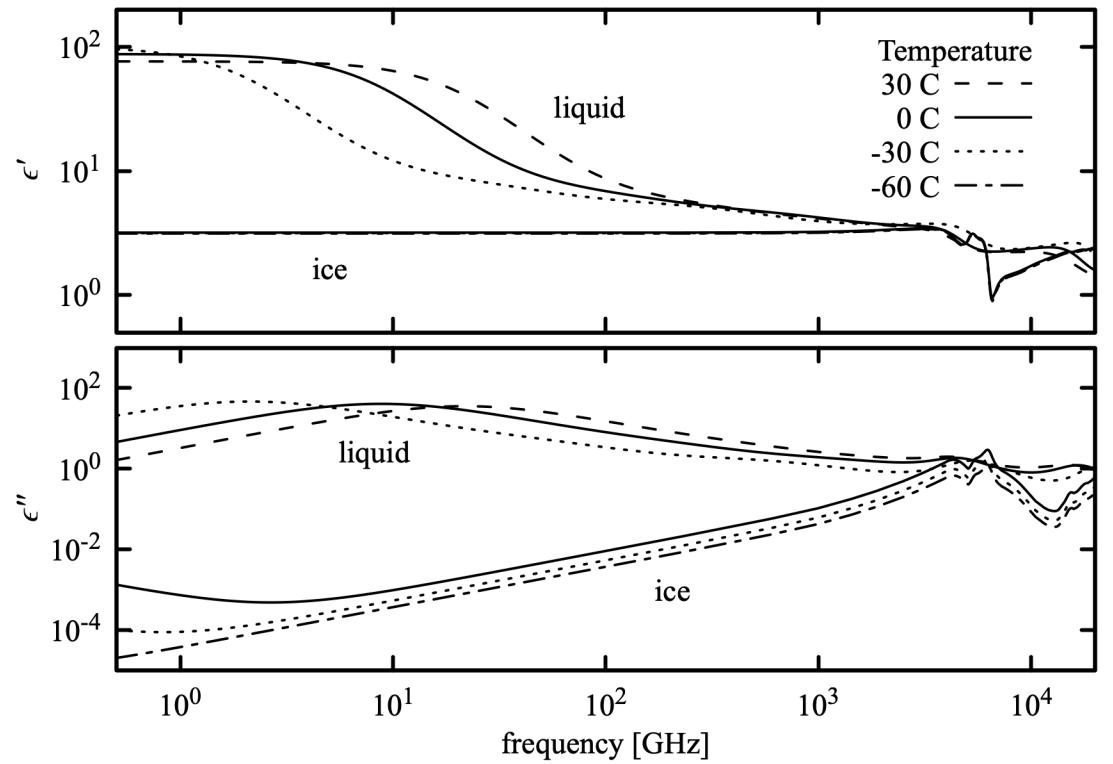


Figure 4.5 Real and imaginary parts of the complex permittivity of liquid water and ice for representative atmospheric temperatures.

$$\begin{aligned} Q_{\text{abs}} &= 4x \Im\left(\frac{\epsilon-1}{\epsilon+2}\right) \\ &= 12x \frac{\epsilon''}{|\epsilon+2|^2} \end{aligned} \quad (4.25)$$

and

$$Q_{\text{sca}} = \frac{8}{3}x^4 \left| \frac{\epsilon-1}{\epsilon+2} \right|^2. \quad (4.26)$$

The absorption or scattering efficiency Q is defined as the ratio of the corresponding cross section $C=Q\pi r^2$ to the projected geometric area πr^2 of the particle. The ratio of scattering to absorption efficiency is

$$\frac{Q_{\text{sca}}}{Q_{\text{abs}}} = \frac{2}{9}x^3 \frac{(\epsilon'-1)^2 + \epsilon''^2}{\epsilon''} \quad (4.27)$$

In *am*, the complex permittivity ϵ is computed for liquid water from the models of Turner et al. (2016) and Ellison (2007), and for ice from Mätzler (2006) and Warren and Brandt (2008). For reference, the real and imaginary parts of ϵ are plotted for liquid water and ice in Figure 4.5

for microwave through submillimeter frequencies and over a range of atmospheric temperatures. Near 300 GHz, $|n_{liquid}| \approx 2.5$ and $|n_{ice}| \approx 1.8$, so $|n| x \approx 0.1$ for non-precipitating cloud particles having typical radii $r \approx 10 \mu\text{m}$. For such particles, the Rayleigh approximation will be valid at frequencies of order 300 GHz and below. The accuracy of the Rayleigh approximation in absorption and scattering computations is analyzed in detail in Ku and Felske (1984); for the example just given, the numerical error of Eq. 4.25 relative to exact Mie scattering theory for ideal spherical particles is approximately 1%. The factor x^3 in Eq. 4.27 ensures that $Q_{sca} \ll Q_{abs}$ over this frequency range, so that scattering can be neglected.

In the Rayleigh limit, the absorption cross section of a particle is proportional to the particle volume:

$$C_{abs} = \pi r^2 Q_{abs} = 18 \pi v_p \frac{\nu}{c} \frac{\epsilon''}{|\epsilon+2|^2} \quad (4.28)$$

where we have used the definition of x in Eq 4.23, and the particle volume

$$v_p = \frac{4}{3} \pi r^3. \quad (4.29)$$

If N_p is the particle column density (particles per unit area), the corresponding opacity τ is

$$\tau = N_p C_{abs} = 18 \pi N_p v_p \frac{\nu}{c} \frac{\epsilon''}{|\epsilon+2|^2} \quad (4.30)$$

$N_p v_p$ is just the total particle volume per unit area, and the opacity is independent of the particle size. If ρ is the water or ice mass density, then noting that $N_p v_p \rho$ is the column mass density M/A , we can write the mass absorption coefficient k_m as

$$k_m = \frac{18 \pi \nu}{\rho c} \frac{\epsilon''}{|\epsilon+2|^2}. \quad (4.31)$$

With this definition,

$$\tau = k_m \frac{M}{A}. \quad (4.32)$$

Formally, even though we are dealing with particles that are aggregates of large numbers of molecules, we can define an equivalent molecular absorption coefficient

$$k = \frac{18 \pi m_{H_2O} \nu}{\rho c} \frac{\epsilon''}{|\epsilon+2|^2}. \quad (4.33)$$

Internally, *am* uses Eq. 4.33, but in configuration files LWP and IWP will normally be given in practical column mass density units such as kg m^{-2} .

4.3.2 Liquid water and ice in *am* models

Two column types are available for including cloud liquid water and ice absorption in *am* models—`lwp_abs_Rayleigh` for liquid water absorption and `iwp_abs_Rayleigh` for ice. We've already seen an example of `lwp_abs_Rayleigh` used in a simple model of absorption by

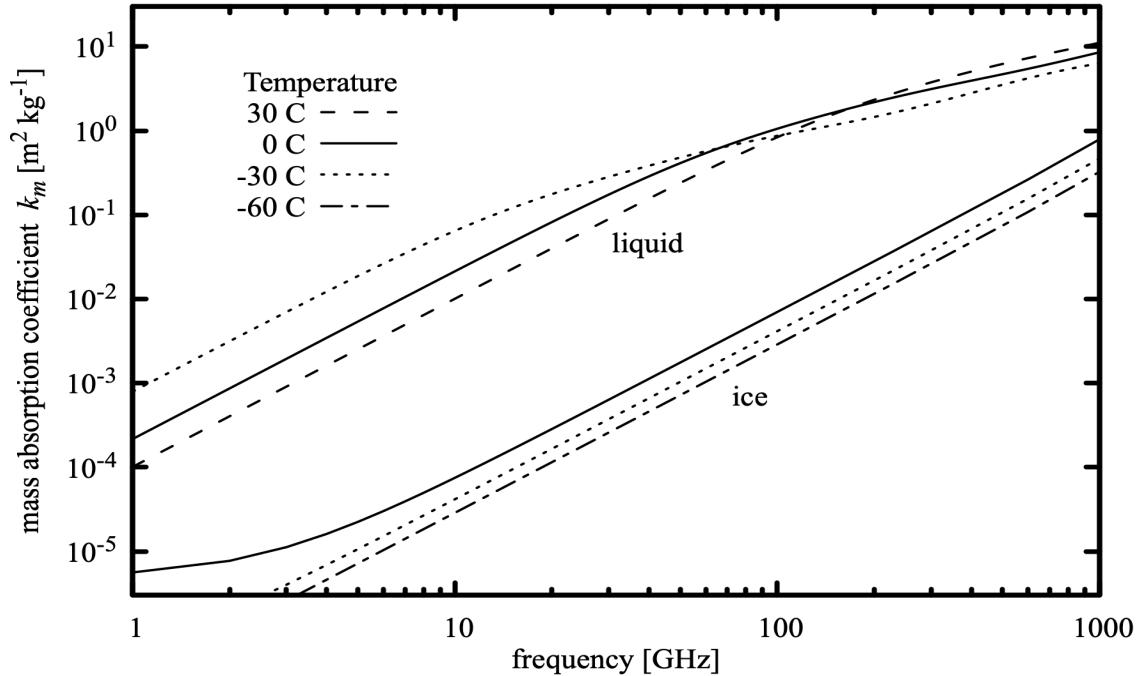


Figure 4.6 Mass absorption coefficient for liquid water and ice clouds in the Rayleigh limit.

supercooled cloud in Chapter 2 in Example 2.4. The following example shows how to use *am* in a bash shell script to compute the mass absorption coefficient for cloud water droplets and ice in the Rayleigh limit, over representative ranges of atmospheric temperature. This is accomplished by computing the opacity for unit LWP and IWP in kg m⁻², which yields the spectral absorption coefficient in m² kg⁻¹:

```
# Example 4.1 - Shell script to compute mass absorption coefficient
#                  spectra in m^2 kg-1 for liquid water and ice in the
#                  Rayleigh limit.

for T in -60 -30 0; do
  am - <<- end > am_iwp_${T}C.out
    f 0 GHz 1000 GHz 1 GHz
    output f GHz tau
    layer
    P 1 atm
    T ${T} C
    column iwp_abs_Rayleigh 1 kg*m^-2
  end
done

for T in -30 0 30; do
  am - <<- end > am_lwp_${T}C.out
    f 0 GHz 1000 GHz 1 GHz
    output f GHz tau
    layer
    P 1 atm
```

```

T ${T} C
column lwp_abs_Rayleigh 1 kg*m^-2
end
done

```

The results are plotted in Figure 4.6. Note that ice absorption is much weaker than water absorption at millimeter wavelengths. As noted above, for typical non-precipitating cloud particles, the Rayleigh approximation can be expected to be accurate up to about 300 GHz. Here we've plotted the absorption coefficient to higher frequency to show some context, but extension of the Rayleigh approximation to frequencies as high as 1000 GHz in normal clouds should be treated with caution.

In the terrestrial atmosphere, supercooled clouds are observed down to approximately -35 C (Hu et al. 2010). Placing liquid water on a model layer below this temperature will trigger an advisory warning. Spontaneous nucleation of water to ice occurs at approximately -45 C (Speedy and Angell 1976); *am* will treat any liquid water found on a model layer below this temperature as an error.

4.4 Gray opacity

For modeling continua having negligible frequency dependence across a band of interest, *am* provides several pure attenuation column types. These differ only in their dependence on airmass factor or zenith angle. The syntax for attenuation column types is similar to that for molecular absorbers, except that the column density field is replaced by an opacity in `neper` or `dB`. (Note that *am* uses the radiometric convention of `neper` as a the natural logarithm of a power ratio; see Table B.10 in Appendix B.)

Attenuation having no zenith angle dependence, such geometric spillover to ambient temperature in the optical system of a radiometer or radio telescope, can be modeled using the `atten` column type. For instance, a spillover loss of 10% to an ambient temperature of 290 K is modeled by adding the following layer as the last layer of a model:

```

layer
P 1 atm
T 290 K
column atten 0.1054 neper

```

Note that the opacity is the negative log of the layer transmittance, $-\log(0.9)=0.1054$. Expressed equivalently in dB, this layer would be

```

layer
P 1 atm
T 290 K
column atten 0.4576 dB

```

since $-10\log_{10}(0.9)=0.4576$. In these examples, a layer pressure had to be specified to satisfy the layer syntax, but otherwise had no effect.

Gray volume absorption can be modeled with the `atten_aimass` column type. This column type is similar to the `atten` type, with the difference that the opacity parameter represents

the zenith opacity, and is scaled by airmass (equal to $\sec(\theta_z)$ in plane-parallel models) for zenith angles θ_z different from zero. The effect is to model a layer of material having spectrally-flat absorption, for which refraction can be neglected. An example would be an aerosol layer, considered over a sufficiently narrow frequency range that the spectral dependence of the absorption coefficient can be neglected.

Two additional attenuation column types are available which operate similarly to the `atten_airmass` column type, but which scale the zenith opacity parameter by harmonic functions of the zenith angle. The `atten_sin^2(za)` and `atten_sin^2(2za)` column types scale their zenith opacity parameters by $\sin^2(\theta_z)$ and $\sin^2(2\theta_z)$ respectively. Their intended use is for modeling elevation-dependent spillover in a radio antenna.

References

- J. Bendtsen 1974, “The Rotational and Rotation-Vibrational Raman Spectra of $^{14}\text{N}_2$, $^{14}\text{N}^{15}\text{N}$ and $^{15}\text{N}_2$.” *J. Raman Spectroscopy* 2:133.
- G. Birnbaum and E. R. Cohen 1976, “Theory of line shape in pressure-induced absorption.” *Can. J. Phys.* 54:593.
- J. Boissoles, C. Boulet, R. H. Tipping, A. Brown, and Q. Ma 2003, “Theoretical calculation of the translation-rotation collision-induced absorption in $\text{N}_2\text{-N}_2$, $\text{O}_2\text{-O}_2$, and $\text{N}_2\text{-O}_2$ pairs.” *J. Quant. Spectrosc. Radiat. Transfer* 82:505.
- A. Borysow and L. Frommhold 1986, “Collision-Induced Rototranslational Absorption Spectra of $\text{N}_2\text{-N}_2$ Pairs for Temperatures from 50 to 300 K.” *Ap. J.* 311:1043. Erratum in *Ap. J.* 320:437
- D. R. Bosomworth and H. P. Gush 1965, “Collision-Induced Absorption of Compressed Gases in the Far Infrared, Part II.” *Can. J. Phys.* 43:751.
- D. E. Burch and D. A. Gryvnak 1980, “Continuum Absorption by H_2O Vapor in the Infrared and Millimeter Regions,” in *Atmospheric Water Vapor*, A. Deepak, T. D. Wilkerson, and L. H. Ruhnke, eds. Academic Press.
- S. A. Clough, F. X. Kneizys, and R. Davies 1989, “Line Shape and the Water Vapor Continuum.” *Atmospheric Research* 23:229.
- S. A. Clough, M. W Shephard, E. J. Mlawer, J. S. Delamere, M. J. Iacono, K. Cady-Pereira, S. Boukabara, and P. D. Brown 2005, “Atmospheric radiative transfer modeling: a summary of the AER codes.” *J. Quant. Spectrosc. Radiat. Transfer* 91:233. See also <http://rtweb.aer.com> for the most recent documentation and source code.

- I. R. Dagg, A. Anderson, S. Yan, W. Smith, and L. A. A. Read 1985, “Collision-induced absorption in nitrogen at low temperatures.” *Can. J. Phys.* 63:625.
- A. Deepak, T. D. Wilkerson, and L. H. Ruhnke 1980, *Atmospheric Water Vapor*. Academic Press.
- P. Dore and A. Filabozzi 1987, “On the nitrogen-induced far-infrared absorption spectra.” *Can. J. Phys.* 65:90.
- W. J. Ellison 2007, “Permittivity of Pure Water, at Standard Atmospheric Pressure, over the Frequency Range 0-25 THz and the Temperature Range 0-100 °C.” *J. Phys. Chem. Ref. Data* 36:1.
- L. Frommhold 1993, *Collision-induced Absorption in Gases*. Cambridge University Press.
- G. Yu. Golubiatnikov and A. F. Krupnov 2003, “Microwave Study of the Rotational Spectrum of Oxygen Molecule in the Range up to 1.12 THz.” *J. Mol. Spec.* 217:282.
- J.-M. Hartmann, C. Boulet, and Daniel Robert 2021, *Collisional Effects on Molecular Spectra*, Second Edition. Elsevier.
- Y. Hu, S. Rodier, K.-M. Xu, W. Sun, J. Huang, B. Lin, P. Zhai, and D. Josset 2010, “Occurrence, liquid water content, and fraction of supercooled water clouds from combined CALIOP/IIR/MODIS measurements.” *J. Geophys. Res. (Atmospheres)* 115:D00H34.
- J. C. Ku and J. D. Felske 1984, “The range of validity of the Rayleigh limit for computing Mie scattering and extinction efficiencies.” *J. Quant Spectrosc. Radiat. Transfer* 31:569.
- C. Mätzler in C. Mätzler, ed. 2006, *Thermal Microwave Radiation: Applications for Remote Sensing*, Section 5.3. Institution of Engineering and Technology, London.
- M. I. Mishchenko, L. D. Travis, and A. A. Lacis 2005, *Scattering, Absorption, and Emission of Light by Small Particles*. NASA / Cambridge University Press. The current electronic edition is available at <http://www.giss.nasa.gov/~crmim/books.html>.
- E. J. Mlawer, S. A. Clough, P. D. Brown, and D. C. Tobin 1998, “Collision-Induced Effects and the Water Vapor Continuum.” Eighth ARM Science Team Meeting Proceedings.
- E. J. Mlawer, S. A. Clough, P. D. Brown, and D. C. Tobin 1999, “Recent Developments in the

Water Vapor Continuum.” Ninth ARM Science Team Meeting Proceedings.

E. J. Mlawer, V. H. Payne, J.-L. Moncet, J. S. Delamere, M. J. Alvarado, and D. C. Tobin 2012, “Development and recent evaluation of the MT_CKD model of continuum absorption.” Phil. Trans. R. Soc. A 370:2520.

E. J. Mlawer, D. D. Turner, S. N. Paine, L. Palchetti, G. Bianchini, V. H. Payne, K. E. Cady-Pereira, R. L. Pernak, M. Alvarado, D. Gombos, J. S. Delamere, M. G. Mlynczak, and J. C. Mast 2019, “Analysis of Water Vapor Absorption in the Far-Infrared and Submillimeter Regions Using Surface Radiometric Measurements From Extremely Dry Locations.” J. Geophys. Res. Atmos. 124:8134.

E. J. Mlawer, K. E. Cady-Pereira, J. Mascio, and I. E. Gordon 2023, “The inclusion of the MT_CKD water vapor continuum model in the HITRAN molecular spectroscopic database.” J. Quant Spectrosc. Radiat. Transfer 306:108645.

T. A. Odintsova, A. O. Koroleva, A. A. Simonova, A. Campargue, and M. Yu. Tretyakov 2022, “The atmospheric continuum in the “terahertz gap” region (15–700 cm⁻¹): Review of experiments at SOLIEL synchrotron and modeling.” J. Mol. Spec. 386:111603.

J. D. Poll and J. L. Hunt 1976, “On the moments of the pressure-induced spectra of gases.” Can. J. Phys. 54:461.

H. R. Pruppacher and J. D. Klett 2010, *Microphysics of Clouds and Precipitation*. Second Revised Edition, Springer.

R. J. Speedy and C. A. Angell 1976, “Isothermal compressibility of supercooled water and evidence for a thermodynamic singularity at –45 °C.” J. Chem. Phys. 65:851

N. W. B. Stone, L. A. A. Read, A. Anderson, I. R. Dagg, and W. Smith 1984, “Temperature dependent collision-induced absorption in nitrogen.” Can. J. Phys. 62:338.

D. D. Turner, S. Kneifel, and M. P. Cadeddu 2016, “An Improved Liquid Water Absorption Model at Microwave Frequencies for Supercooled Liquid Water Clouds.” J. Atmos. Oceanic Technol. 33:33.

S. G. Warren and R. E. Brandt 2008, “Optical constants of ice from the ultraviolet to the microwave: A revised compilation.” J. Geophys. Res. 113:D14220.

Chapter 5: Modeling Radiometers

The calibrated output of a radiometer is a modified version of the spectrum received at its input. In general, information is lost, and it is not possible to recover the fully-resolved input spectrum from the radiometer output. Instead, the forward model which is used to interpret the radiometer output, or which serves as part of a retrieval method, must incorporate a model of the radiometer response in addition to the model for the atmosphere itself.

Several facilities for modeling the response of radiometers are available in *am*. The spectral response of a radiometer can be modeled by convolving spectra with various instrumental line shape (ILS) functions. Frequency conversions performed by heterodyne receivers can be modeled to produce single- or double-sideband intermediate frequency (IF) spectra. A simple spectrally-flat radiometer noise model can be used to produce system temperature spectra. Finally, the instrument model can include a reference black body load at specified temperature, to produce differential radiance or Y-factor spectra.

5.1 Instrumental line shapes

An instrumental line shape (ILS) models the spectral response of a radiometer to a monochromatic unit impulse spectrum at its input. The response to an arbitrary input spectrum is thus the convolution of the ILS and the input spectrum. Strictly, this should be the spectrum which corresponds directly to the signal processed by the instrument—for example $I(\nu)$ for a Fourier transform spectrometer, and $T_{R-J}(\nu)$ for a heterodyne receiver (see section 1.2.4). In practice, when the ratio of radiance to brightness temperature can be regarded as constant over the spectral width of the ILS, it is a good approximation to convolve the ILS with whatever radiance or brightness temperature spectrum will ultimately be produced as the calibrated product of the measurement. There may even be numerical advantages to doing so—a radiance spectrum converted to brightness temperature may have a flatter baseline, mitigating end effects in the convolution. Convolution of transmittance, $t(\nu)$, with the ILS is meaningful in the context of a measurement made against a background radiation source having a spectrum which is resolved by

the ILS. On the other hand, ILS convolution with opacity $\tau(\nu)$, absorption coefficients $k(\nu)$, and delay $L(\nu)$, is generally not useful.

5.1.1 Instrumental line shapes in *am*

In Fourier transform spectrometers, and in receiving systems equipped with autocorrelating or FFT spectrometers, the instrumental line shape (ILS) in the frequency domain is determined by a bounded weighting function, often called a *window function*, that is applied to the sampled signal in the time domain. In the simplest case, the time-domain window is a rectangle function determined, for example, by the length of the interferometer scan in an FTS, by the maximum lag in an autocorrelator, or by the length of the voltage sample buffer in an FFT spectrometer. Other window functions with nonuniform weighting are often used to obtain a desired ILS, a practice referred to as *apodization* in Fourier spectroscopy, or simply *windowing* in other signal processing contexts. Harris (1978) gives a good general review. Amato, et al. (1998) discuss apodization of FTS spectra in the context of remote sensing retrievals.

Table 5.1 lists ILS functions available in *am* that correspond to commonly-used window functions. The related pairs of ILS and window functions are expressed in the normalized units defined in the table footnote in terms of the maximum time delay T_m processed by the interferometer or spectrometer. In practice, the main purpose of windowing is to reduce ringing features (or *sidelobes*) in the ILS, and consequent cross-coupling between spectral channels. Sidelobe reduction is obtained at the cost of spectral resolution as illustrated in Figure 5.1.

Taking the functions in Table 5.1 in turn, the `sinc` ILS corresponds to equal weighting of all lags or time samples up to $t=T_m$. The `sinc_squared` or `Bartlett` ILS corresponds to the triangular weighting function defined in the table. Note that this triangle function is the self-convolution of a rectangle function having half the total width of the triangle. By the convolution theorem, the ILS is then the square of the sinc function ILS associated with the rectangle function, and thus non-negative. For this reason, a short triangular weighting function is often used for computing the low-resolution phase spectrum when phase-correcting FTS spectra, since it avoids phase reversals in the sidelobes of strong spectral features. In the time domain, the `Hann`, `Hamming`, and `Blackman` functions correspond to linear combinations of rectangle and cosine windows. The resulting ILS functions are linear combinations of unshifted and shifted sinc functions, and the coefficients of these functions are chosen to achieve partial cancellation of sidelobes. The `Hann` function is widely used in radio astronomy—it corresponds to the standard window applied by the ALMA correlator, for example. The slightly modified coefficients of the `Hamming` function improve cancellation of the first sidelobe compared with `Hann`, but at the expense of higher levels of subsequent sidelobes. The `Blackman` ILS achieves very low sidelobe levels with an additional cosine term, though at the cost of nearly a factor of two in resolution compared with a rectangular window. The remaining functions in Table 5.1, the `Norton-Beer` functions (Norton and Beer 1976), are empirically-optimized functions used in Fourier transform spectroscopy, offering three different tradeoffs between spectral resolution and sidelobe level.

TABLE 5.1. ILS FUNCTIONS WITH BOUNDED TIME-DOMAIN WINDOWS

NAME	FWHM	ILS FUNCTION $Q(a)$, WINDOW FUNCTION $W(x)$
sinc	3.791	$Q(a)=Q_0(a)$ $W(x)=\begin{cases} 1 & x \leq 1 \\ 0 & x > 1 \end{cases}$
sinc_squared, Bartlett	5.566	$Q(a)=\frac{1}{2}[Q_0(a/2)]^2$ $W(x)=\begin{cases} 1- x & x \leq 1 \\ 0 & x > 1 \end{cases}$
Hann	6.283	$Q(a)=\frac{1}{2}Q_0(a)+\frac{1}{4}[Q_0(a+\pi)+Q_0(a-\pi)]$ $W(x)=\begin{cases} \frac{1}{2}(1+\cos(\pi x)) & x \leq 1 \\ 0 & x > 1 \end{cases}$
Hamming	5.703	$Q(a)=0.54 \cdot Q_0(a)+0.23 \cdot [Q_0(a+\pi)+Q_0(a-\pi)]$ $W(x)=\begin{cases} 0.54+0.46 \cdot \cos(\pi x) & x \leq 1 \\ 0 & x > 1 \end{cases}$
Blackman	7.222	$Q(a)=0.42 \cdot Q_0(a)+0.25 \cdot [Q_0(a+\pi)+Q_0(a-\pi)] + 0.04 \cdot [Q_0(a+2\pi)+Q_0(a-2\pi)]$ $W(x)=\begin{cases} 0.42+0.50 \cos(\pi x)+0.08 \cos(2\pi x) & x \leq 1 \\ 0 & x > 1 \end{cases}$
Norton-Beer_I1	4.549	$Q(a)=0.384093 \cdot Q_0(a)-0.087577 \cdot Q_1(a)+0.703484 \cdot Q_2(a)$ $W(x)=\begin{cases} 0.384093-0.087577 \cdot (1-x^2)+0.703484 \cdot (1-x^2)^2 & x \leq 1 \\ 0 & x > 1 \end{cases}$

TABLE 5.1. ILS FUNCTIONS WITH BOUNDED TIME-DOMAIN WINDOWS,
CONTINUED

NAME	FWHM	ILS FUNCTION $\mathcal{Q}(a)$, WINDOW FUNCTION $W(x)$
Norton-Beer_I2	5.307	$\mathcal{Q}(a)=0.152442 \cdot \mathcal{Q}_0(a) - 0.136176 \cdot \mathcal{Q}_1(a) + 0.983734 \cdot \mathcal{Q}_2(a)$ $W(x)=\begin{cases} 0.152442 - 0.136176 \cdot (1-x^2) + 0.983734 \cdot (1-x^2)^2 & x \leq 1 \\ 0 & x > 1 \end{cases}$
Norton-Beer_I3	6.066	$\mathcal{Q}(a)=0.045335 \cdot \mathcal{Q}_0(a) + 0.554883 \cdot \mathcal{Q}_2(a) + 0.399782 \cdot \mathcal{Q}_4(a)$ $W(x)=\begin{cases} 0.045335 + 0.554883 \cdot (1-x^2)^2 + 0.399782 \cdot (1-x^2)^4 & x \leq 1 \\ 0 & x > 1 \end{cases}$

In this table, the time lag x is in normalized units $x=t/T_m$, where T_m is the maximum sampled lag. The frequency a and the FWHM of $\mathcal{Q}(a)$ are in normalized units $a=2\pi\nu T_m$. The functions $\mathcal{Q}_i(a)$ are defined as follows:

$$\begin{aligned} \mathcal{Q}_0(a) &= \frac{\sin(a)}{a} & \mathcal{Q}_2(a) &= -\frac{8}{a^2} \left[\left(1 - \frac{3}{a^2}\right) \frac{\sin(a)}{a} + \left(\frac{3}{a^2}\right) \cos(a) \right] \\ \mathcal{Q}_1(a) &= \frac{2}{a^2} \left[\frac{\sin(a)}{a} - \cos(a) \right] & \mathcal{Q}_4(a) &= \frac{384}{a^4} \left[\left(1 - \frac{45}{a^2} + \frac{105}{a^4}\right) \frac{\sin(a)}{a} + \left(\frac{5}{a^2}\right) \left(2 - \frac{21}{a^2}\right) \cos(a) \right] \end{aligned}$$

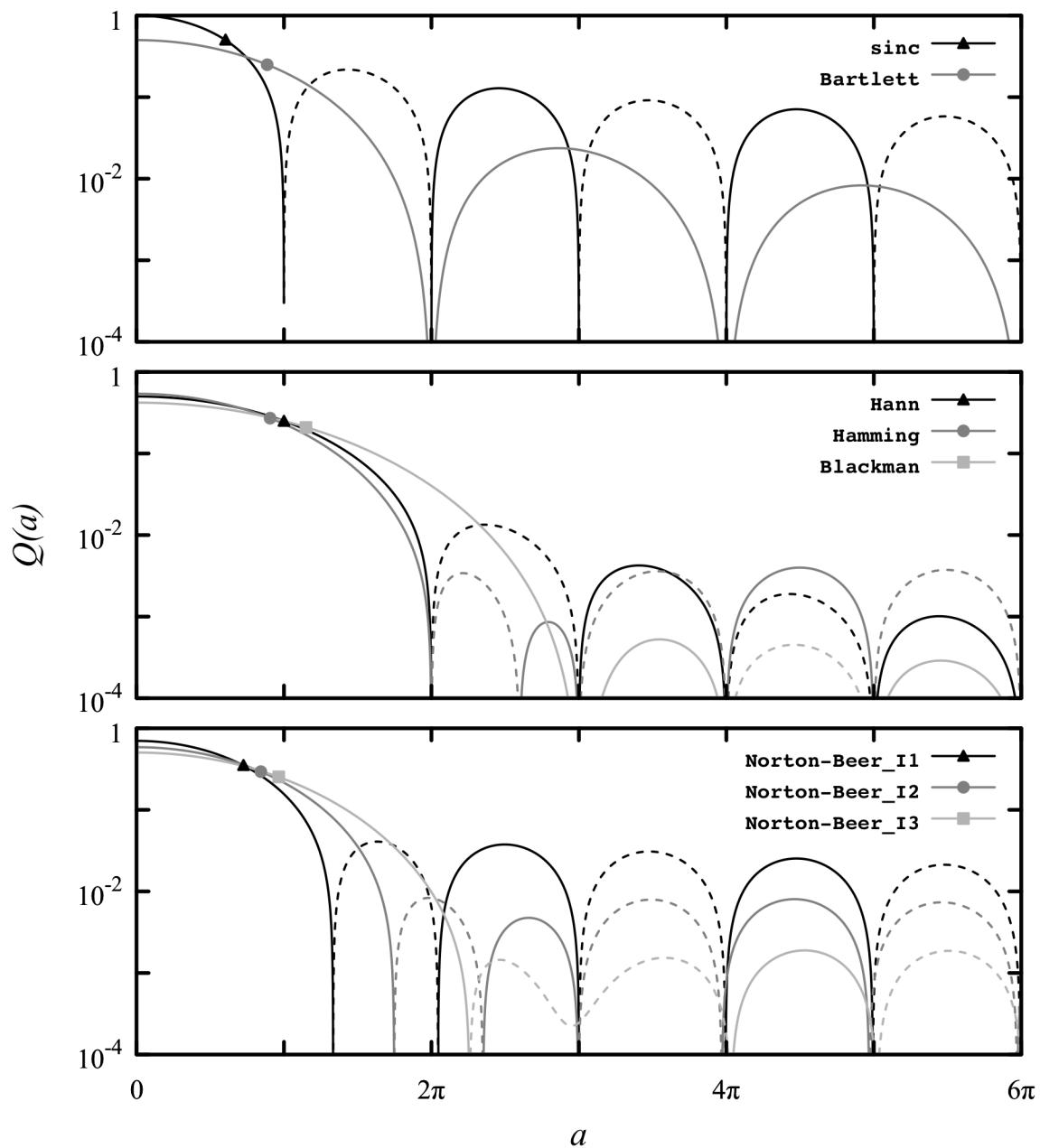


Figure 5.1 ILS functions from Table 5.1, plotted against frequency in normalized units where $a = 2\pi \nu T_m$. A dashed line indicates the function has negative sign. On each curve, the symbol marks the half-maximum point.

TABLE 5.2. ADDITIONAL ILS FUNCTIONS

NAME	ILS FUNCTION
Gaussian	$Q(a) = \sqrt{\pi} \exp(-a^2)$ $a = 2 \ln 2 \left(\frac{\nu}{\Delta \nu_{FWHM}} \right)$
rectangle	$Q(a, da) = \begin{cases} \text{impulse}(a, da) & da > \frac{1}{2} \\ \text{trapezoid}(a, da) & da \leq \frac{1}{2} \end{cases}$ $\text{impulse}(a, da) = \begin{cases} \frac{\pi}{da} \left(1 - \frac{ a }{da} \right) & a < da \\ 0 & a \geq da \end{cases}$ $\text{trapezoid}(a, da) = \begin{cases} \pi & a \leq \frac{1}{2} - da \\ \pi \left(\frac{1}{2} + \frac{1/2 - a }{2 da} \right) & \frac{1}{2} - da < a < \frac{1}{2} + da \\ 0 & a \geq \frac{1}{2} + da \end{cases}$ $a = \frac{\nu}{\Delta \nu_{FWHM}}, \quad da = \frac{d \nu}{\Delta \nu_{FWHM}}$

In addition to the functions in Table 5.1, two additional ILS functions, defined in Table 5.2, are available—a generic Gaussian function, and a frequency-domain rectangle function suitable for modeling slits and sharp bandpass filters. The rectangle function is actually a trapezoid, explicitly constructed to be correctly normalized on a discrete frequency grid. It collapses to a discrete impulse function if the user-specified full width at half maximum $\Delta \nu_{FWHM}$ is less than twice the frequency grid interval $d \nu$. The ILS functions in Table 5.1 and Table 5.2 are all normalized to have area π when integrated over the frequency a in normalized units; Internally in the *am* source code, the ILS is normalized to unity by multiplying by the factor $\left(\frac{d \nu}{\pi} \right) \left(\frac{a_{FWHM}}{\Delta \nu_{FWHM}} \right)$.

5.1.2 Using instrumental line shapes

Instrumental line shapes are incorporated into *am* model configuration files by adding a single line specifying the ILS name, the full width at half-maximum in real frequency units, and optionally the output arrays which are to be convolved with the ILS. If the output arrays are not

specified, then the ILS will be applied to all output arrays which are allowed to be convolved with an ILS; see the entry under `ils` in Appendix B for details.

For example, suppose we want to modify the model of Example 2.5 to compute the Planck brightness temperature spectrum convolved with the frequency response of a Fourier transform spectrometer which records an interferogram up to a maximum optical path difference $L=100$ mm from the zero lag position. The maximum time lag sampled is

$$T_m = \frac{L}{c} = 0.3336 \text{ ns} . \quad (5.1)$$

If we process the entire interferogram with a rectangular window, then the ILS is a sinc function with full width at half maximum

$$\Delta \nu_{FWHM} = \frac{3.791}{2\pi T_m} = 1.809 \text{ GHz} , \quad (5.2)$$

where we have converted the FWHM in normalized units from Table 5.1 to real units. To use this ILS, we would add the line

```
ils sinc 1.809 GHz
```

to the model configuration file. It doesn't matter where we put this line in the file, but conventionally it would go near the top, above the layer descriptions. Alternatively, if we chose to apodize the interferogram with a Blackman window, the ILS would be the Blackman ILS function with full-width at half-maximum

$$\Delta \nu_{FWHM} = \frac{7.222}{2\pi T_m} = 3.445 \text{ GHz} , \quad (5.3)$$

and we would instead include the line

```
ils Blackman 3.445 GHz
```

in the model file. Instead of hard-coding the ILS name and width into our modified version of Example 2.5, it can be more convenient to make them both command-line parameters, so that the first few lines of the modified file are:

```
.
.
f %1 %2 %3 %4 %5 %6
output f GHz tau Tb K
za %7 %8
ils %12 %13 %14
tol 1e-4
.
```

The original Example 2.5 is what's known as a *monochromatic* model—each output point corresponds to the fully-resolved spectral density at that frequency. When the monochromatic spectrum is convolved with an ILS, an implicit assumption is being made that the monochromatic spectral density at each frequency point approximates the local spectral density averaged over the frequency grid spacing. The accuracy of this approximation can vary widely from sharp line

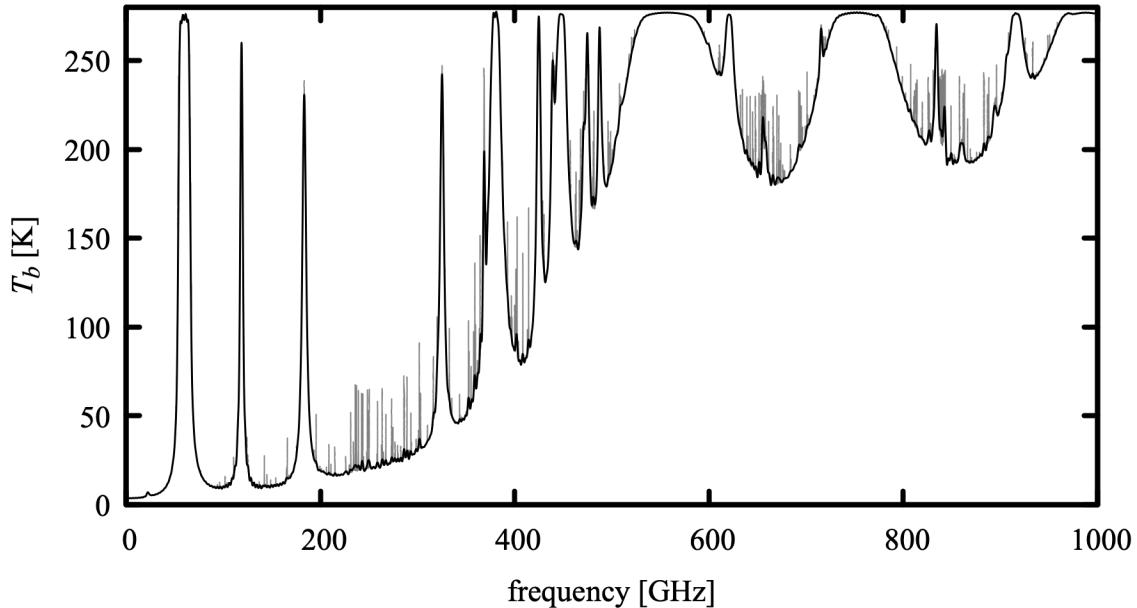


Figure 5.2 Fully-resolved spectrum computed from the model of Example 2.5 (light gray) compared with the same spectrum convolved with the sinc function ILS of an FTS which scans to a maximum optical path difference $L=100$ mm (black).

centers to smooth line wings. For a given model, some experimentation with frequency grid spacings may be needed to find a suitable balance between compute time and accuracy. Figures 5.2 and 5.3 each compare the output from the original model with the output from the modified one with our two example ILS functions applied to the spectrum. These were run at 2 MHz model grid resolution with the commands

```
$ am example2.5.amc 0 GHz 1000 GHz 2 MHz 0 deg 277 K 0.5385 \
>example2.5.0-1000.out
$ am example2.5.ils.amc 0 GHz 1000 GHz 2 MHz 0 deg 277 K 0.5385 \
Blackman 3.445 GHz >Blackman_3.445.out
$ am example2.5.ils.amc 0 GHz 1000 GHz 2 MHz 0 deg 277 K 0.5385 \
sinc 1.809 GHz >sinc_1.809.out
```

If we were to recompute these spectra at 20 MHz grid resolution, we would find the monochromatic spectrum to be under-sampled near the line centers, but the convolved T_b spectra would be only slightly changed, by no more than 50 mK. That is, the area under the sharp peaks doesn't contribute much to the convolution with our comparatively low-resolution instrumental function. This suggests that a useful approximation strategy for computing low resolution spectra quickly is simply to reduce the resolution of the pressure grid in the highest model layers, collapsing the top layers into a single layer at higher average pressure. Some trial-and-error experimentation might be needed to balance speed and accuracy as desired.

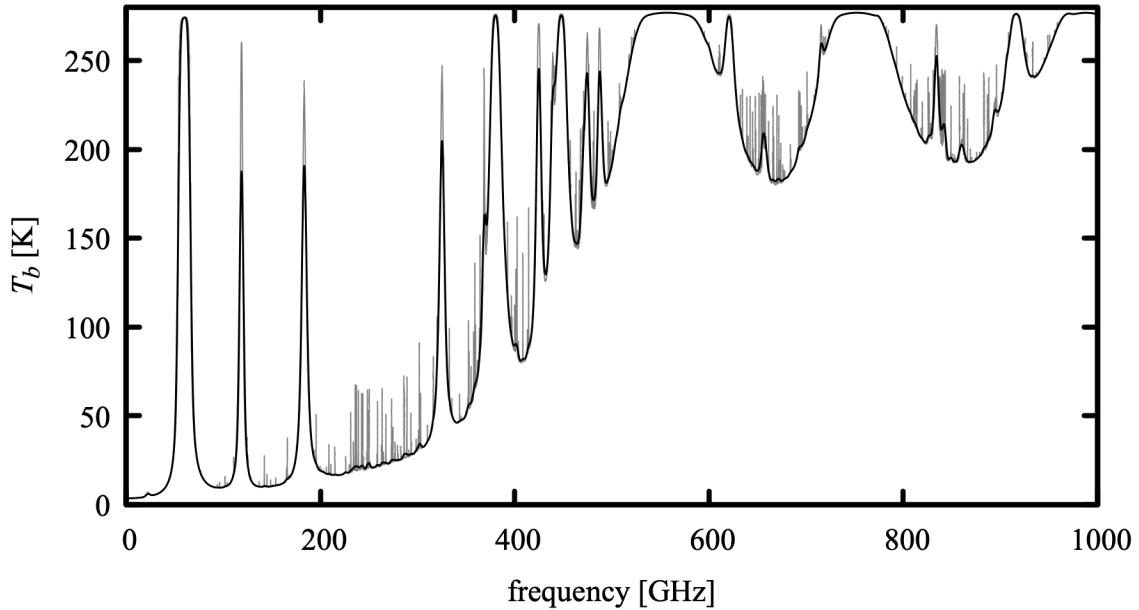


Figure 5.3 Comparison as in Figure 5.2, but with Blackman apodization applied to the FTS interferogram.

Comparing Figures 5.2 and 5.3 it is clear that for the same interferometer scan length, the Blackman apodization gives up considerable resolution to eliminate the modest amount of ringing in the unapodized spectrum. Nevertheless, in practical situations apodization can be of value in preventing noise or interference confined to one part of the spectrum from contaminating other spectral regions. Examples from Fourier transform spectroscopy are detector $1/f$ noise contaminating the low frequency end of the spectrum where the Planck radiance is also weakest, and spurious spectral features associated with instrumental vibrations or electrical interference contaminating nearby frequency channels.

5.2 Difference spectra

Many radiometers produce a signal which is not proportional to radiance directly, but rather to the difference in radiance between the input scene and a reference blackbody load. In the simplest case, the instrument may chop or switch between scene and reference load. Another example would be a Fourier transform spectrometer based on a polarizing interferometer, with one input polarization viewing the scene and the orthogonal polarization terminated on a reference load. In this case the signal power in the two output polarizations, as a function of the interferometer scan, forms two complementary interferograms having Fourier transforms proportional to the positive and negative spectral radiance differences between scene and reference load.

For such an instrument, the signal to noise ratio is proportional to the radiance difference signal

$$I_{\text{diff}}(\nu) = I(\nu) - B(\nu, T_{\text{ref}}). \quad (5.4)$$

Because of this, it may be desirable to cast the forward model (and corresponding Jacobians; see Chapter 8) in the same terms. Radiance difference spectra can be obtained as a program output from *am* by using an `output` statement specifying the `I_diff` output type, for example

```
output f GHz I_diff watt*cm-2*GHz-1*sr-1
```

The reference temperature T_{ref} is set using the `Tref` statement, as in

```
Tref 285 K
```

The `I_diff` output type is also useful when fitting difference spectra using *am*'s built-in fit facility, since for such spectra it is again the only way to correctly weight the spectral data points by signal to noise ratio.

5.3 IF spectra

In a heterodyne receiver, incident radiation (traditionally called the RF, for radio frequency, signal) is combined with a monochromatic local oscillator (LO) signal at frequency ν_{LO} in a mixer. The mixer is a device, such as a Schottky diode, SIS junction, or square-law detector, which responds nonlinearly to the sum of the RF and LO signal amplitudes. For a given RF frequency ν_{RF} , mixer output signals occur at intermediate frequencies ν_{IF} which are sums and differences of ν_{LO} , ν_{RF} , and their harmonics, so that

$$\nu_{\text{IF}} = m \nu_{\text{RF}} \pm n \nu_{\text{LO}}, \quad (5.5)$$

where ν_{IF} can be positive or negative.

For an ideal square-law detector, $m=n=1$, and the IF output power is linear with the input power. For a more general nonlinear device, other values of m and n can occur. Usually, one is still interested in isolating difference frequencies with $m=n=1$, which can be achieved through a combination of filtering, tuning of the device or its embedding circuit, or use of multiple devices in a balanced mixer configuration. (An exception is a subharmonically-pumped or harmonic mixer, which is optimized to operate with $n>1$.) Normally, the LO power is arranged to be much higher than the total RF power, so that the IF responds nearly linearly to the RF (although spectral variation in noise and conversion gain normally require calibration) and harmonics $m>1$ are relatively suppressed. The mixer output is typically filtered by a lowpass or bandpass filter restricting ν_{IF} to a desired frequency range, and usually $|\nu_{\text{IF}}| \ll \nu_{\text{LO}}$. A simple mixer operates in *double-sideband* (DSB) mode. That is, signals from the upper sideband (USB) with $\nu_{\text{RF}} > \nu_{\text{LO}}$, and corresponding signals from the lower sideband (LSB) with $\nu_{\text{RF}} < \nu_{\text{LO}}$ that convert to the same IF frequency, simply add together. Ideally, a DSB receiver responds equally to both sidebands with a *sideband ratio* of unity, but this is not always achieved in practice. By driving

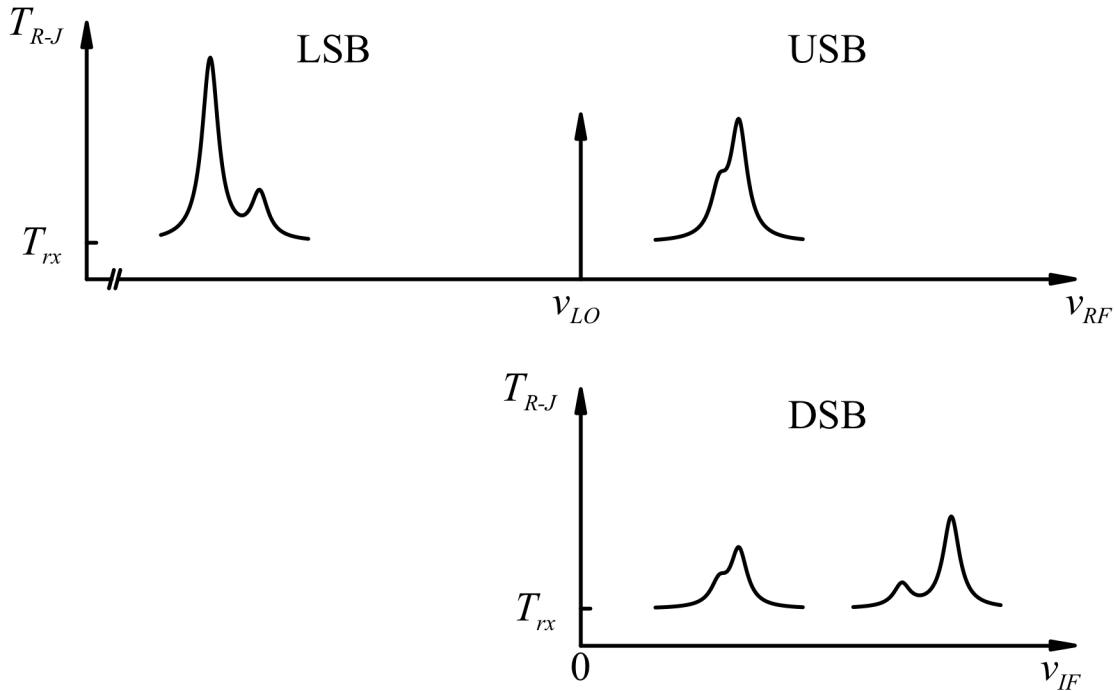


Figure 5.4 Frequency conversion in a heterodyne receiver. Note that the spectrum in the lower sideband, $\nu_{RF} < \nu_{LO}$, is inverted in the IF spectrum.

two matched mixers with either the RF or LO in phase quadrature, and subsequently combining their IF outputs in phase quadrature, it is possible to make a *sideband-separating* (2SB) receiver, which has separate outputs for the USB and LSB spectra. The relationship between the input and output spectra for heterodyne receivers is illustrated in Figure 5.4. As this figure suggests, the signal to noise ratio of a feature occurring in one sideband of a DSB receiver is half that for a comparable 2SB receiver, because the two sidebands, including the noise, are averaged together in the DSB case.

5.3.1 Producing IF spectra from *am* models

IF spectra are produced from *am* models by adding a configuration file statement specifying ν_{LO} , and whether a USB, LSB, or DSB spectrum is desired. Additional statements can be used to restrict the output frequency range, and to change the sideband ratio from its default value of unity. For computing system temperature and Y-factor spectra, a configuration file statement is available for setting a spectrally-flat receiver noise temperature.

To illustrate, assume that once again we have a model configuration file similar to Example 2.5 for Mauna Kea, starting out with the statements

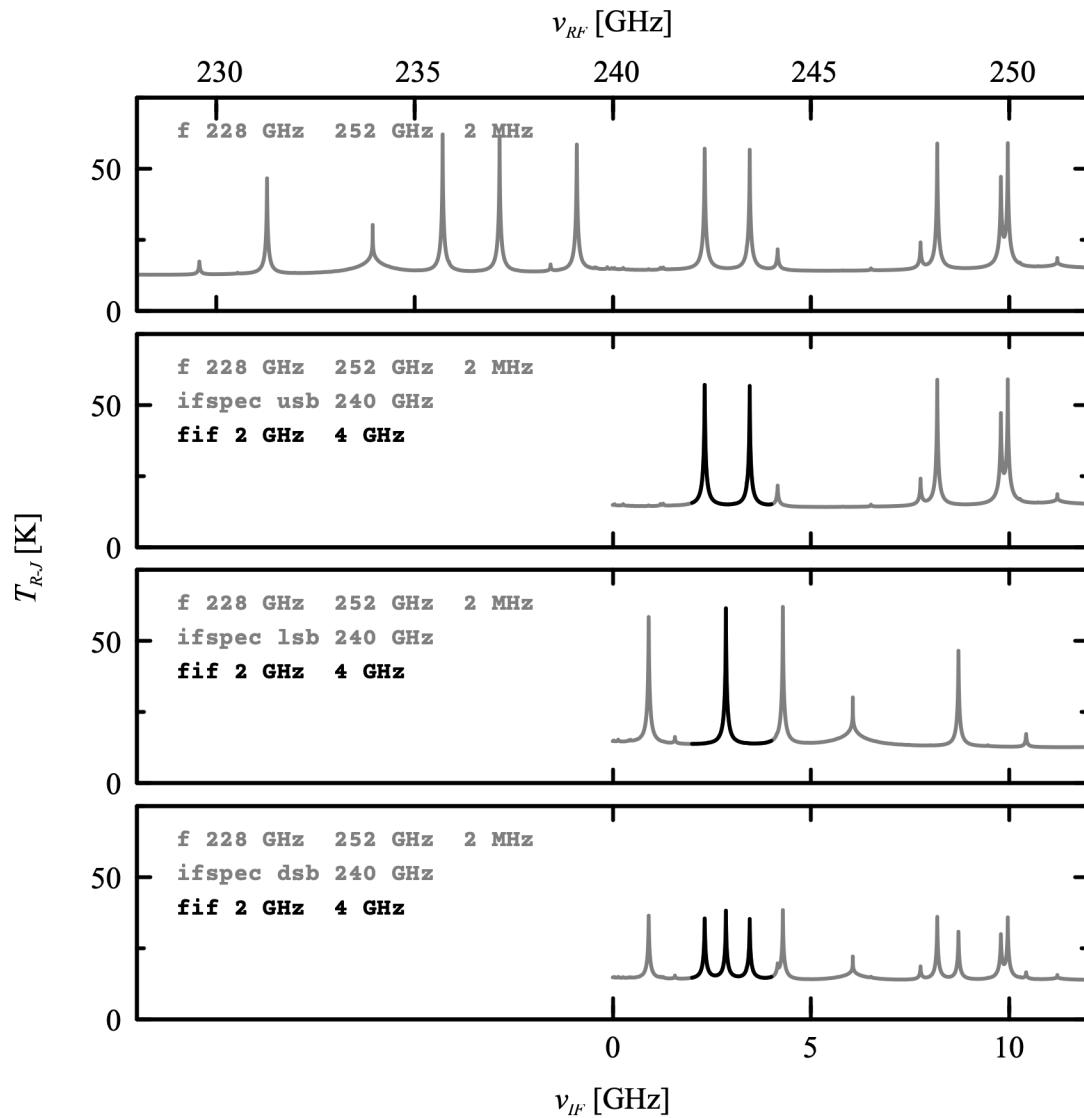


Figure 5.5 Model brightness temperature spectrum versus ν_{RF} (top panel), and corresponding IF spectra produced using the IF spectrum configuration statements shown.

```

f 228 GHz 252 GHz 2 MHz
output f GHz Trj K
.
.
```

Running this model produces the output plotted in the top panel of Figure 5.5. Next, to simulate the upper-sideband output of a sideband-separating heterodyne receiver with $\nu_{LO}=240\text{ GHz}$, we add the statement

```
ifspec usb 240 GHz
```

The output of the program changes to that plotted in the second panel of Figure 5.5. Similarly, changing the first argument of the `ifspec` command to `lsb` or `dsb` changes the output to LSB or DSB, respectively, as shown in the third and fourth panels. Note that the IF frequency range, 0–12 GHz, is the maximum range encompassed by the model frequency grid, 228 GHz–252 GHz. Also, while the model computation grid is always aligned to $\nu_{RF}=0$ as discussed in Section 2.2.1, the output IF spectrum will be on a frequency grid with the same grid interval, but interpolated to be aligned with $\nu_{IF}=0$.

The IF frequency range in the output can be restricted to a smaller range using the `fif` statement. For example, to restrict the output to IF frequencies from 2 GHz–4 GHz, we would add the statement

```
fif 2 GHz 4 GHz
```

which produces the output indicated by the bold segments of the plots in Figure 5.5. It may appear that setting the model frequency grid range broader than that implied by the `fif` statement, as here, results in wasted computation; absorption coefficients which are not found in the disk cache (see Section 2.3) will be computed over the entire model grid. However, this is desirable if one anticipates running the model over a range of LO tunings; once the absorption coefficients are in cache, subsequent runs will take much less time. It may also be desirable to use a computation grid which exceeds the IF range to provide spectral padding if an ILS is being used to model the channels of the back-end spectrometer of a heterodyne receiver. As an optimization, unless an ILS is being applied, `am` will carry out the radiative transfer computation only for those grid frequencies needed to compute the IF spectrum.

A sideband ratio different from unity can be incorporated into DSB models using the `dsb_utol_ratio` statement to specify the upper-to-lower sideband ratio r , defined such that

$$T_{R-J,DSB}(\nu) = \frac{r}{1+r} T_{R-J,USB}(\nu) + \frac{1}{1+r} T_{R-J,LSB}(\nu) \quad (5.6)$$

For example, an upper-to-lower sideband ratio $r=2$ would be set via the statement

```
dsb_utol_ratio 2
```

5.3.2 System temperature and spectral Y-factor

The spectra discussed so far correspond to fully-calibrated spectra, as would be obtained from a channel-by-channel, two-temperature calibration of a receiver. Two additional output spectra, `Tsys` and `y`, selectable using the `output` statement, are available for modelling partially-calibrated spectra, assuming a spectrally-flat receiver noise temperature T_{rx} given as a model parameter. Spectrally-flat T_{rx} is a desirable system characteristic, since it means that the same sensitivity is achieved across the receiver band. However, it should be noted that it is not always realized in practice, especially for very wide band systems.

The *system temperature* T_{sys} as computed by *am* follows the definition used in radiometry, which is simply the sum of the brightness temperature of the atmospheric scene and the receiver noise temperature:

$$T_{sys}(\nu) = T_{R-J}(\nu) + T_{rx}. \quad (5.7)$$

The receiver noise temperature T_{rx} is set with the `Trx` statement, for example

```
Trx 55 K
```

to set the receiver noise temperature to 55 K. In radio astronomy, an alternative definition of the system temperature is used which includes the effects of atmospheric absorption. This is the system temperature referred to the top of the atmosphere, defined as

$$\begin{aligned} T_{sys}^*(\nu) &= T_{sys}(\nu) e^{\tau(\nu)} \\ &= \frac{T_{sys}(\nu)}{t(\nu)}. \end{aligned} \quad (5.8)$$

$T_{sys}^*(\nu)$ can be computed with *am* by specifying both `tsys` and `tx` as outputs, and taking their ratio as above.

The *spectral Y-factor* is defined as the channel-by-channel ratio of the receiver output power when viewing an arbitrary scene, to the output power when viewing a reference load at a fixed physical temperature T_{ref} :

$$Y(\nu) = \frac{T_{R-J}(\nu) + T_{rx}}{T_{R-J, ref}(\nu) + T_{rx}}, \quad (5.9)$$

where

$$T_{R-J, ref}(\nu) = T_{ref} \left(\frac{h\nu/kT_{ref}}{\exp(h\nu/kT_{ref}) - 1} \right) \quad (5.10)$$

is the Rayleigh-Jeans brightness temperature of the reference load. Note the different conventions for T_{rx} and T_{ref} . T_{rx} is an equivalent Rayleigh-Jeans temperature associated with the receiver noise power spectral density. T_{ref} , on the other hand, is a physical load temperature, which must be converted to an equivalent Rayleigh-Jeans brightness temperature $T_{R-J, ref}$ for use in the Y-factor ratio computation.

Computation of the spectral Y-factor requires that both T_{rx} and T_{ref} be set in the model configuration file. The latter is set with the `Trref` statement; for example

```
Trref 293 K
```

sets a reference load temperature of 293 K.

References

- U. Amato, D. De Canditiis, and C. Serio 1998, "Effect of apodization on the retrieval of geophysical parameters from Fourier-transform spectrometers." *Applied Optics* 37:6537

F. J. Harris 1978, "On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform." Proc. IEEE 66:51.

R. H. Norton and R. Beer 1976, "New apodizing functions for Fourier spectroscopy." J. Opt. Soc. Am. 66:259. Erratum in J. Opt. Soc. Am. 66:419.

Chapter 6: Delay Spectra

Absorption and refraction are often, for convenience, treated as independent phenomena. In fact, as a consequence of the general physical principle of causality, they are inextricably linked. The real and imaginary parts of the refractive index are Hilbert transforms of one another—from absorption one can compute refraction, and vice versa. The catch is that the Hilbert transform is defined over an infinite domain, so in principle we need to know the absorption everywhere in order to compute the refractivity anywhere. Fortunately, in practice absorption can often be treated as approximately band-limited, and adjustments can be made for parts of the spectrum which are left out of the computation.

For a complete propagation path, the analog of the relationship between absorption and refraction is that between opacity τ and *excess delay* L/c . L/c is the increase in propagation time beyond that required to traverse the same path in vacuum; L alone is called the *excess path*. In *am*, $L(\nu)$ is computed from $\tau(\nu)$, with adjustments for refractivity originating from out-of-band optical transitions.

6.1 Delay computation

6.1.1 Kramers–Kronig relations

The real and imaginary parts of the complex refractive index $n = n_r + i \cdot n_i$ are related to one another through the Hilbert, or Kramers–Kronig (1926), transform pair

$$n_r(\nu) - 1 = \frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{n_i(\nu')}{\nu' - \nu} d\nu' \quad (6.1)$$

and

$$n_i(\nu) = -\frac{1}{\pi} P \int_{-\infty}^{\infty} \frac{n_r(\nu') - 1}{\nu' - \nu} d\nu', \quad (6.2)$$

where P denotes the Cauchy principal value of the integral, on account of the singularity in the

integrand at $\nu' = \nu$. Eqs. 6.1 and 6.2 are a general consequence of the properties that $(n-1) \rightarrow 0$ as $\nu \rightarrow \infty$, and that $n(\nu)$ arises from physical processes that satisfy causality. The latter means that $n(\nu)$ can be written as a Fourier transform of a real response function $\tilde{n}(t)$ in the time domain, for which $\tilde{n}(t)=0$ for all $t<0$, that is,

$$n(\nu)-1=\int_0^\infty \tilde{n}(t)e^{2\pi i \nu t} dt \quad (6.3)$$

Since $\tilde{n}(t)$ is real, Eq. 6.3 implies also that $n(-\nu)=n^*(\nu)$, so that relations such as Eqs. 6.1 and 6.2 can also be (and often are) written in the form

$$n_r(\nu)-1=\frac{2}{\pi}P\int_0^\infty \frac{\nu'n_i(\nu')}{\nu'^2-\nu^2}d\nu' \quad (6.4)$$

and

$$n_i(\nu)=-\frac{2}{\pi}P\int_0^\infty \frac{\nu'(n_r(\nu')-1)}{\nu'^2-\nu^2}d\nu'. \quad (6.5)$$

Relations analogous to the above exist for any complex quantity satisfying the same conditions, such as the susceptibility $\chi=n^2-1$. Good textbook derivations of the Kramers–Kronig relations are in Landau and Lifshitz (1960) and in Arfken (1985); see also Toll (1956).

To compute the excess path L from the opacity τ , we start by noting that if z represents distance along a propagation path, along which n can vary, then the opacity $d\tau$ contributed by a segment of path dz is

$$d\tau(\nu,z)=\frac{4\pi\nu}{c}n_i(\nu,\textcolor{blue}{z})dz, \quad (6.6)$$

and the excess path contributed by this segment is

$$dL(\nu,z)=(n_r(\nu,z)-1)dz. \quad (6.7)$$

Dividing Eqs. 6.6 and 6.7 each by dz , and substituting into Eq. 6.4, yields

$$\frac{dL(\nu,z)}{dz}=\frac{c}{2\pi^2}P\int_0^\infty \frac{d\tau(\nu',z)/dz}{\nu'^2-\nu^2}d\nu'. \quad (6.8)$$

Since the ν and z dependence of L and τ are separable, Eq. 6.8 may be integrated over z to obtain

$$L(\nu)=\frac{c}{2\pi^2}P\int_0^\infty \frac{\tau(\nu')}{\nu'^2-\nu^2}d\nu'. \quad (6.9)$$

Comparing with Eq. 6.4, it can be seen that $L(\nu)$ is the Hilbert transform of $\frac{c}{4\pi\nu}\tau(\nu)$. This relationship is independent of the details of the propagation path. That is, once we have computed the spectral opacity for a complete path, a process which may involve iterating through many model layers, we have the information we need to compute the spectral excess path or delay in a single Hilbert transform computation.

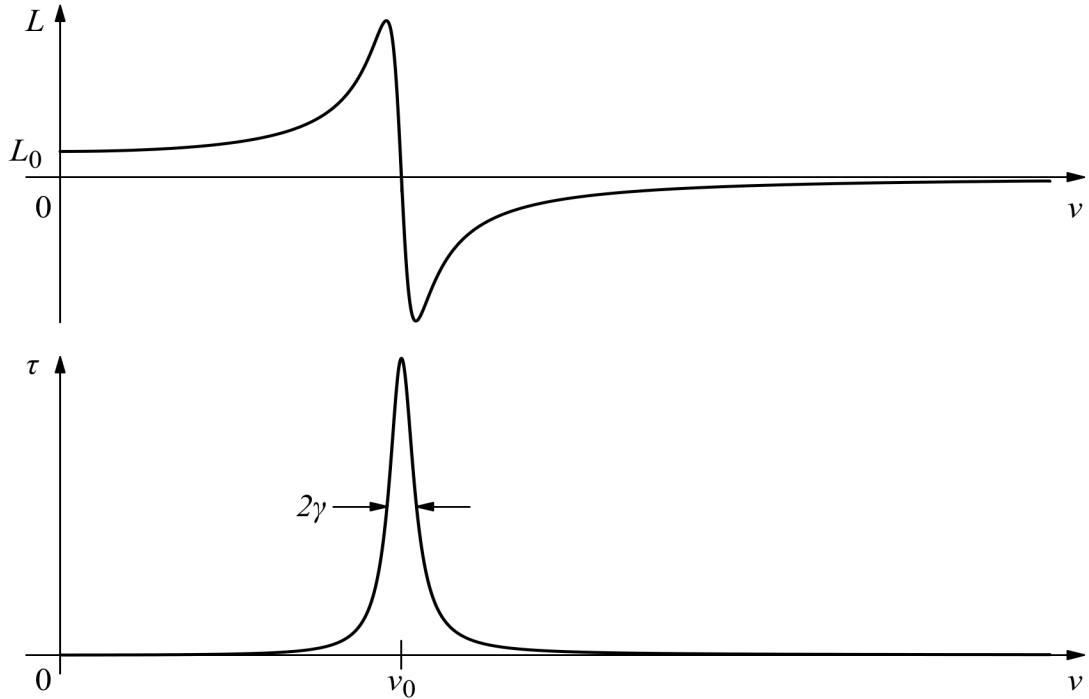


Figure 6.1 Opacity and delay for a single spectral line.

6.1.2 Opacity and delay for spectral lines

Figure 6.1 illustrates the opacity τ and excess path L for a single spectral line centered at ν_0 . If we can assume that the integrand in Eq. 6.9 falls to zero at frequencies far from ν_0 , then $L(\nu)$ has the following limiting behaviors. For $\nu \gg \nu_0$, the integrand is small and negative near $\nu' = \nu_0$, and as $\nu \rightarrow \infty$, $L \rightarrow 0$ from below. For $\nu \ll \nu_0$, ν can be neglected relative to ν' near $\nu' = \nu_0$, and $L(\nu)$ approaches a constant value L_0 , where

$$L_0 = \frac{c}{2\pi^2} P \int_0^\infty \frac{\tau(\nu')}{\nu'^2} d\nu'. \quad (6.10)$$

Recall from section 3.1 that the opacity $\tau(\nu)$ may be written as the product of a line-of-sight column density N , a temperature-dependent line strength $S(T)$, and a normalized line shape function $F(\nu)$,

$$\tau(\nu) = N S(T) F(\nu), \quad (6.11)$$

so that Eq. 6.10 becomes

$$L_0 = \frac{N c S(T)}{2\pi^2} P \int_0^\infty \frac{F(\nu')}{\nu'^2} d\nu'. \quad (6.12)$$

If the linewidth $2\gamma \ll \nu_0$, then in the denominator of the integrand in Eq. 6.12, ν' can be approximated as constant and equal to ν_0 across the width of the line, so

$$L_0 = \frac{N c S(T)}{2\pi^2 \nu_0^2}, \quad (6.13)$$

since $F(\nu)$ is normalized. Eq. 6.9 is linear in τ , so the excess path for a group of lines is simply the sum of that for each line taken individually.

The negative excess delay at frequencies above the line center in Figure 6.1 might at first appear surprising, as it corresponds to a monochromatic phase velocity exceeding the speed of light c in vacuum. However, satisfaction of the Kramers-Kronig relations guarantees that in the time domain, no disturbance in the electromagnetic field propagates faster than c .

6.1.3 The Smith-Weintraub equation

In the low-frequency limit, below all spectral lines for a given species, the excess path is the sum of terms like Eq. 6.13 for each line:

$$L(\nu \rightarrow 0) = \frac{N c}{2\pi^2} \sum_i \frac{S_i(T)}{\nu_i^2}. \quad (6.14)$$

For the molecular species and temperatures encountered in the atmosphere, Eq. 6.14 can typically be partitioned into a sum over vibrational and electronic transitions with $h\nu_i \gg kT$, and (for those species with a permanent molecular dipole moment), an additional sum over pure rotational transitions with frequencies $h\nu_i < kT$. In this case, it can be shown (Van Vleck, 1927) that the latter sum varies as T^{-1} , while the former is independent of temperature. Moreover, this partitioning corresponds to the two terms of the classical Langevin-Debye formula for the susceptibility,

$$\chi = n_v \left(\alpha + \frac{\mu^2}{3kT} \right), \quad (6.15)$$

where n_v is the molecular volume density, α is the molecular polarizability, and μ is the permanent molecular dipole moment. Recall that the susceptibility χ is related to the refractive index n through

$$n^2 = 1 + \chi. \quad (6.16)$$

For gases at atmospheric pressure $\chi \ll 1$, so the refractivity $n-1$ is

$$n-1 \approx \frac{1}{2} \chi. \quad (6.17)$$

In radio propagation, it is customary to work with the refractivity as a function of partial pressure p rather than the susceptibility as a function of gas density n_v . Using the ideal gas law $p=n_v kT$ in Eq. 6.15, we may write for the refractivity

$$n-1 = \frac{p}{2kT} \left(\alpha + \frac{\mu^2}{3kT} \right). \quad (6.18)$$

In the low-frequency limit, the refractivity is purely real, and Eq. 6.18 may be written as

$$n-1 = p \left(\frac{A}{T} + \frac{B}{T^2} \right) \quad (6.19)$$

with A and B regarded as empirical real coefficients derived directly from measurements.

For a gas mixture, the total refractivity is the sum of terms like the right-hand side of Eq. 6.19 for each component. A well-established two-component formulation, widely used to compute the radio refractivity of moist air for applications in meteorology, geodesy, and radio astronomy, is the Smith–Weintraub equation (Smith and Weintraub 1953):

$$10^6(n-1) = k_1 \frac{p_d}{T} + k_2 \frac{e}{T} + k_3 \frac{e}{T^2}. \quad (6.20)$$

Here, p_d and e are the respective partial pressures of dry air and water vapor; T is the absolute temperature; and k_1 , k_2 , and k_3 are empirical coefficients. The k_1 term is associated with the average polarizability of dry air species such as N₂, O₂, and Ar. Trace dry air species with permanent electric dipole moments and the weakly interactive magnetic dipole moment of O₂ are neglected, so there is no permanent dipole term for dry air. The k_2 and k_3 terms are associated with the induced and permanent dipole moments, respectively, of H₂O. From an analysis of available measurements, Bevis et al. (1994) derived a set of recommended values for the coefficients in Eq. 6.20:

$$\begin{aligned} k_1 &= 77.60(5) \text{ K mbar}^{-1} \\ k_2 &= 70.4(2.2) \text{ K mbar}^{-1} \\ k_3 &= 3.739(12) \cdot 10^5 \text{ K}^2 \text{ mbar}^{-1} \end{aligned} \quad (6.21)$$

Several of the measurements from which these values were derived disagreed with one another by significantly more than their stated uncertainties, so the uncertainties assigned by Bevis et al. involved a degree of subjective judgement.

6.1.4 Complete delay computation

A complete excess path spectrum in *am* is computed as the sum of the discrete Hilbert transform of $\frac{c}{4\pi\nu} \tau(\nu)$ plus a constant “optical” offset corresponding to the delay contributed by lines above the frequency grid. The Hilbert transform, Eq. 6.9, is computed by double Fourier transformation—for background see Ohta and Ishida (1988), also Johnson (1975), Peterson and Knight (1973) and Marple (1999). The offsets for optical refractivity are derived following Thayer (1974), modified so as to maintain consistency with the Bevis et al. (1994) recommended values for the k_1 and k_2 coefficients in the Smith–Weintraub equation. Details are provided in the *am* source file `phys_const.h`. In the current version of *am*, the dividing point between these two contributions is fixed at the maximum internal line catalog frequency, which is 15 THz; there is

no adjustment for lines that lie outside the model frequency grid. Consequently, for full accuracy when computing delay spectra, the frequency grid must be set to 0–15 THz with sufficient resolution to capture all strong lines. Often, the spectrum over this entire frequency range will be neither needed nor desired. In this case, the output frequency range can be limited to a subset of the full computation grid using the `fout` statement, as shown by example in the following section.

6.2 Examples

The excess delay can be computed for atmospheric models of arbitrary complexity. Here, however, we'll consider an example of contrived simplicity, the “wet delay” associated with a unit increment of water vapor column density at the SMA site near the summit of Mauna Kea. The wet delay contributes to the path delay differences between antennas located at different altitudes. (Depending on array configuration, SMA antennas can differ in altitude by up to 36 m.) For concreteness, we will assume the total pressure $P=625$ mbar, and the ambient temperature $T=4$ C. We'll assume a column density of 1 precipitable micron of water vapor across a path length of 1 meter, which happens to correspond closely to the median water vapor mixing ratio on Mauna Kea. To keep this example simple and focus on the wet component alone, we won't include dry air. We should note, however, that a realistic computation would likely involve a variable amount of water vapor mixed with dry air, at slowly-varying pressure and temperature. The dry excess delay, itself nearly flat spectrally, would significantly exceed the wet delay even near the dispersive peaks associated with strong water lines. Moreover, the displacement (at constant pressure) of dry air by water vapor would be a non-negligible effect. With that admonition, the configuration file for our simplified example is

```
# Example 6.1 - Wet excess delay on Mauna Kea

f 0 THz 15 THz 25 MHz
output f GHz L um
selfbroad_vmr_tol 0

layer
P 625 mbar
T 4 C
h 1 m
column h2o 1 um_pvw
```

Here, the output spectrum is specified to be delay expressed as excess path in microns; `am` can output delay spectra as excess path or time delay in any of the units listed in Table B.10 in Appendix B. Note that we've set `selfbroad_vmr_tol` to zero to force strict computation of self-broadening as discussed in Section 2.3.1. Running the file produces the following:

```
$ am example6.1.amc >example6.1.out
# am version 14.0 (build date Sep 26 2024 13:20:00)

# run time 0.322 s
# dcache hit: 0 miss: 1 discard: 0

f 0 THz 15 THz 25 MHz
output text f GHz L um
```

```

tol 0
selfbroad_vmr_tol 0

geometry plane-parallel
refract none
za 0 deg
PTmode midpoint

T0 0 K

layer
P 625 mbar
T 4 C
h 1 m
lineshape VVH_750 h2o_lines
column h2o 1 um_pwv (vmr 0.002047)

#
# column densities [cm^-2], including scale factors
#
# total (1 layer):           zenith           line-of-sight
#                           h2o   3.342700e+18   3.342700e+18
#                               (1 um_pwv)     (1 um_pwv)
#

```

Note that the H₂O volume mixing ratio $x_{H_2O} = 0.002047$, derived from the explicitly-defined column density of 1 precipitable micron, is reported as a comment next to the h2o column definition. This value is close to that at the base of the median H₂O profile over Mauna Kea, plotted in Figure 2.4. The computed excess path spectrum is plotted in Figure 6.2. Although for accuracy we needed to compute the spectrum from 0–15 THz, on Mauna Kea we are really only interested in frequencies up to 1000 GHz at most. To avoid writing the unneeded data, an fout statement can be used to specify that a subrange of the computed spectrum be written as output, as in

```

# Example 6.2 - Wet excess delay on Mauna Kea, restricted output range

f 0 THz 15 THz 25 MHz
fout 0 GHz 1000 GHz
output f GHz L um
selfbroad_vmr_tol 0

layer
P 625 mbar
T 4 C
h 1 m
column h2o 1 um_pwv

```

The output from running this file, plotted in Figure 6.3, is just the same as that from Example 6.1, but over the reduced frequency range.

For a path distance h , the low frequency limit of the wet delay based on the Smith–Weintraub equation is

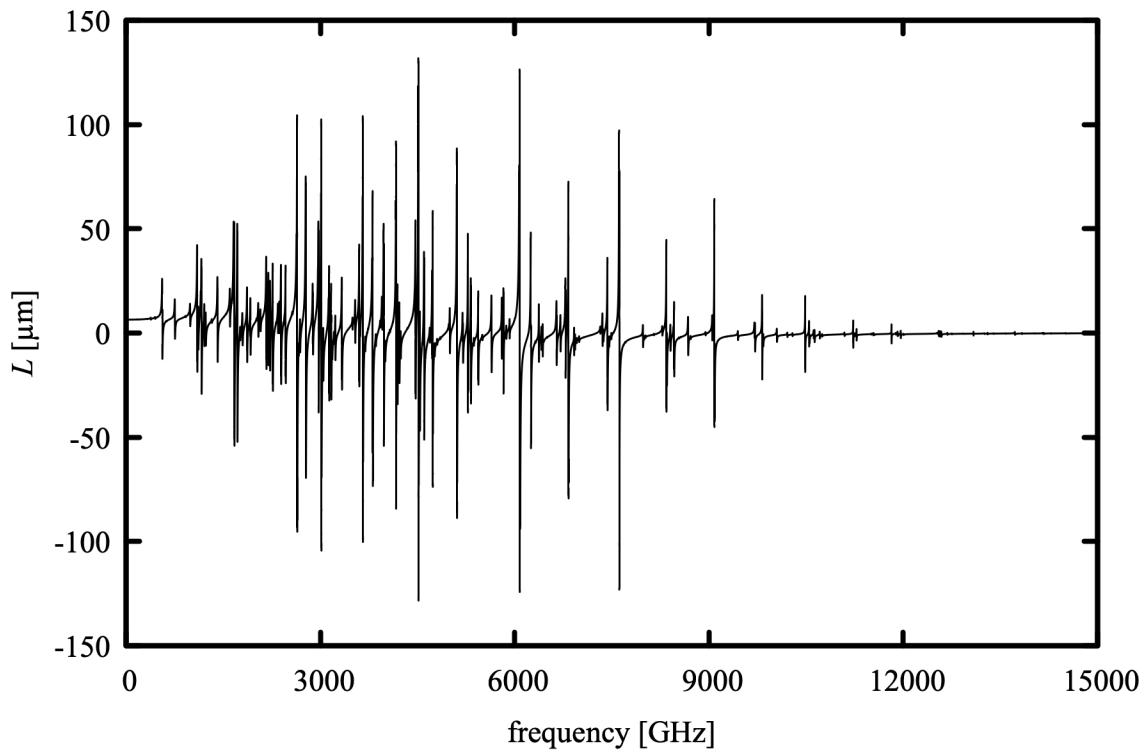


Figure 6.2 Output from Example 6.1—wet excess path for $1 \mu\text{m}$ precipitable water under median surface conditions on Mauna Kea, 0–15 THz showing the entire H_2O pure rotation band.

$$L_{S-W} = h \cdot (n-1) = h \cdot 10^{-6} \left(k_2 \frac{e}{T} + k_3 \frac{e}{T^2} \right). \quad (6.22)$$

For the conditions in our example, namely $h=1 \text{ m}$, $T=277.15 \text{ K}$, and $e=x_{\text{H}_2\text{O}} \cdot 625 \text{ mbar}$, the Smith-Weintraub wet delay is

$$L_{S-W} = 6.553(22) \mu\text{m}, \quad (6.23)$$

plotted as the light grey horizontal line in Fig. 6.3. The estimated error in Eq 6.23 is based on the estimated errors for k_2 and k_3 from Bevis et al. (1994), and is dominated by the error in the k_3 term, which is about 20 times larger than the k_2 term. In comparison, the *am* low-frequency wet delay from Example 6.2 is $L(\nu \rightarrow 0) = 6.413 \mu\text{m}$, which lies well outside this estimated error. Since the optical delay in *am* is pinned to k_2 , the discrepancy reflects the difference between low frequency direct measurements of k_3 and the low frequency limit of the Hilbert transform of the absorption spectrum computed by *am*.

In making this comparison, a factor to consider is that the pressure and temperature dependencies in the Smith-Weintraub equation are derived from the susceptibility of an ideal gas

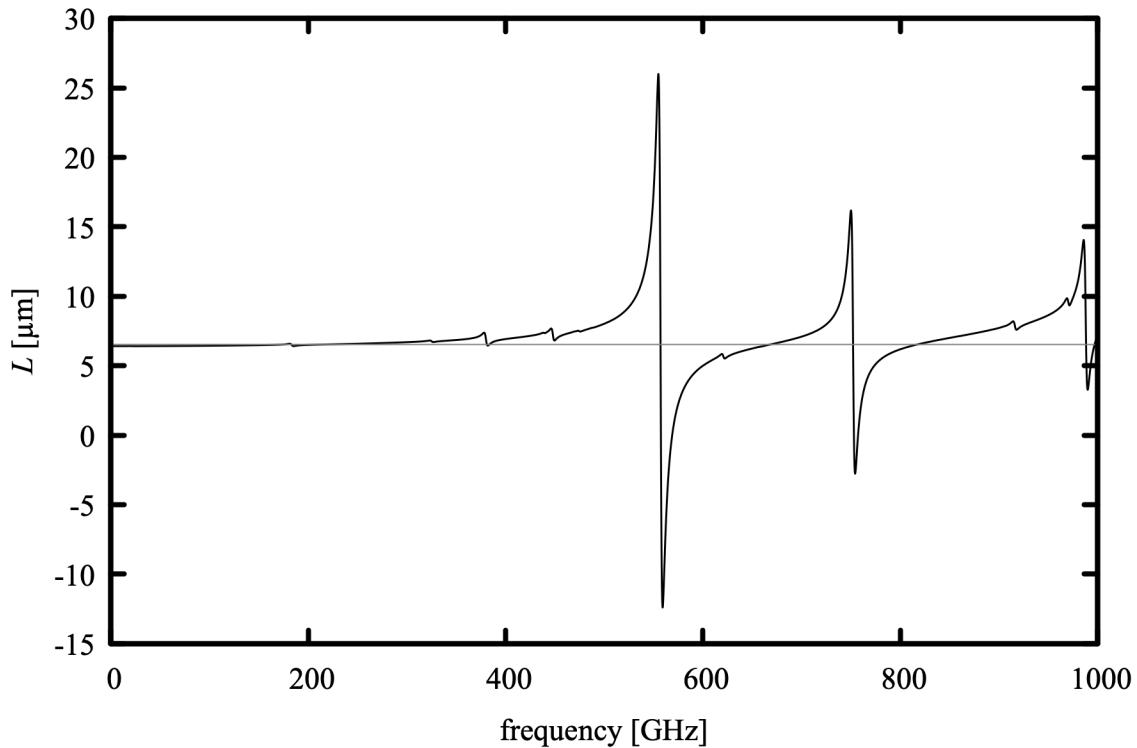


Figure 6.3 Output from Example 6.2—wet excess path for $1\text{ }\mu\text{m}$ precipitable water under median surface conditions on Mauna Kea, 0–1000 GHz. The gray line shows the wet excess path in the low-frequency limit from the Smith–Weintraub equation.

of isolated molecules, whereas the delay computed by *am* also contains bimolecular contributions from the foreign- and self-induced water vapor continua which scale quadratically in pressure as $p_d e$ and e^2 , respectively. To investigate the significance of this, we can separately compute the various contributions to the wet delay in Example 6.2 by replacing the composite column type `h2o` with its various components. This is easily done by embedding a call to *am*, together with its configuration data, within a short shell script that iterates over the relevant column type names as in the following example:

```
# Example 6.3 - Shell script to compute components of the wet excess
#                  delay on Mauna Kea, in the low frequency limit
#
for COLTYPE in \
    h2o_lines \
    h2o_optical_refractivity \
    h2o_air_continuum \
    h2o_self_continuum \
    h2o \
; do
L=$((
am - <<- end 2>/dev/null
f 0 THz 15 THz 25 MHz
```

```

fout 0 THz 0 THz
selfbroad_vmr_tol 0
output L um
layer
P 625 mbar
T 4 C
h 1 m
column $COLTYPE 1 um_pvv
end
)
echo $L $COLTYPE
done

```

Note the use here of the `output` and `fout` statements tell *am* to write the excess path only, at a single frequency point $\nu=0$, as its sole output. This output is assigned to the shell variable `L`, then written out next to the column type. Running this script produces the following:

```

$ ./example6.3.sh
6.033308e+00 h2o_lines
3.249026e-01 h2o_optical_refractivity
5.214141e-02 h2o_air_continuum
2.242827e-03 h2o_self_continuum
6.412595e+00 h2o
$

```

From this we see that the continuum accounts for about 0.9% of the wet radio refractivity for typical conditions near the summit of Mauna Kea. A similar computation for the typical conditions of a laboratory measurement of wet refractivity such as Essen (1953) (e.g. 15 mbar of pure water vapor at 20 C) gives a relative self-continuum contribution of less than 0.5%. Thus, the continuum is at most a minor contributor to the 2.1% difference between the computation in the foregoing examples and the Smith–Weintraub equation.

References

- G. Arfken 1985, *Mathematical Methods for Physicists*, 3rd ed. Academic Press.
- M. Bevis, S. Businger, S. Chiswell, T. A. Herring, R. A. Anthes, C. Rocken, and R. H. Ware 1994, “GPS Meteorology: Mapping Zenith Wet Delays onto Precipitable Water.” *J. Appl. Meteo.* 33:379.
- L. Essen 1953 “The Refractive Indices of Water Vapour, Air, Oxygen, Nitrogen, Hydrogen, Deuterium and Helium.” *Proc. Phys. Soc.* B66:189.
- R. J. Hill, R. S. Lawrence, and J. T. Priestley 1982, “Theoretical and calculational aspects of the radio refractive index of water vapor.” *Radio Science* 17:1251.
- D. W. Johnson 1975, “A Fourier series method for numerical Kramers–Kronig analysis.” *J. Phys A: Math. Gen.* 8:490.

- R. de L. Kronig 1926, "On the theory of dispersion of X-Rays." *J. Opt. Soc. Am.* 12:547.
- L. D. Landau and E. M. Lifshitz 1960, *Electrodynamics of Continuous Media*. Translated by J. B. Sykes and J. S. Bell., Pergamon
- S. L. Marple, Jr. 1999, "Computing the Discrete-Time "Analytic" Signal via FFT." *IEEE Trans. Signal Processing* 47:2600.
- K. Ohta and H. Ishida 1988, "Comparison Among Several Numerical Integration Methods for Kramers-Kronig Transformation." *Applied Spectroscopy* 42:952.
- C. W. Peterson and B. W. Knight 1973, "Causality calculations in the time domain: An efficient alternative to the Kramers-Kronig method." *J. Opt. Soc. Am.* 63:1238.
- E. K. Smith and S. Weintraub 1953, "The Constants in the Equation for Atmospheric Refractive Index at Radio Frequencies." *Proc. IRE* 41:1035.
- G. D. Thayer 1974, "An improved equation for the radio refractive index of air." *Radio Science* 9:803.
- J. S. Toll 1956, "Causality and the Dispersion Relation: Logical Foundations." *Phys. Rev.* 104:1760.
- J. H. Van Vleck 1927, "On Dielectric Constants and Magnetic Susceptibilities in the New Quantum Mechanics Part I. A General Proof of the Langevin-Debye Formula." *Phys. Rev.* 29:727.

Chapter 7: Fits

am includes a built-in facility for spectral fits involving a small number of adjustable model parameters. Given an *am* model configuration file, almost any numerical model parameter can be designated as a fit variable. For example, the column density of a given species on a single layer might be varied to fit a spectral measurement, or the total column density of a species having a column density profile defined across multiple model layers can be varied by making the corresponding `nscale` parameter a fit variable. Other examples of potential fit variables are layer temperatures, frequency offset to fit a line position (using the `ifspec` facility), instrumental line width, and receiver noise temperature.

Fits in *am* are performed by varying the designated fit variables to minimize a function, called the *fit estimator*, of the weighted residual differences between a measured spectrum and a modeled spectrum. The minimization is carried out using the downhill simplex method of Nelder and Mead (1965). This is a widely-used example of a *direct search* method (Kolda et al. 2003), a general class of optimization algorithms which do not require derivatives of the function being minimized or maximized. This supports the use of fit estimator functions such as the sum of absolute residuals, which may be desirably robust against outliers such as glitches or spurs in a spectrum, but which have derivatives that are discontinuous in the fit variables. Press, et al. (2007) contains good discussions of the downhill simplex method and of the use of robust goodness-of-fit estimators.

am's built-in fitting is intended as a convenient tool in simple situations for which such fits are stable. Users interested in using *am* as the forward model in a regularized statistical retrieval will find what they need in Chapter 8: Jacobians. There, we will see that the syntax for designating differentiation variables is similar to that described in this chapter for designating fit variables. Here, the use of *am*'s fit facility will be illustrated by way of a worked example, involving validation of a receiver calibration method from a test at the Submillimeter Array on Mauna Kea. A bootstrap analysis will be used to estimate the statistical uncertainties of the fit parameters, and it will be shown how to do this efficiently using *am*'s streaming fit capability.

7.1 A worked example

The worked example for this chapter is a fit to a double-sideband spectrum acquired with one of the facility astronomical receivers of the Submillimeter Array. The spectrum contains the $J K_a K_c = 16\ 1\ 15 \rightarrow 16\ 0\ 16$ pure rotation line of ozone, at 231.2815 GHz, observed in emission. The starting point for the fit will be an atmospheric model constructed from a combination of surface meteorological data, radiosonde profiles, and reanalysis data. The model profiles for water vapor and ozone will be scaled to fit the spectrum, and the column densities thus obtained will be compared with independent satellite and radiometer data. The goals are twofold—to validate the receiver calibration method, and to estimate the statistical uncertainty of total column ozone and water vapor retrieved from the spectrum.

7.1.1 The spectral data

The data to be fit are a double-sideband Rayleigh-Jeans brightness temperature spectrum $T_{sky}(\nu_{IF})$, plotted in Figure 7.1. This is the calibrated output from a 16 K channel digital spectrometer fed by an SIS receiver tuned to a LO frequency of 226.016563 GHz. The receiver was in turn fed by a radio telescope pointed 5 degrees off zenith. The on-sky integration time was $t_{sky}=1\text{ s}$. The calibration made use of a separate sky dip to find the channel-by-channel DSB receiver noise temperature $T_{rx}(\nu_{IF})$, and a reference scan taken on an ambient temperature load. These auxiliary measurements involved integration times much longer than the on-sky integration for this scan, so the standard deviation of $T_{sky}(\nu_{IF})$ is simply

$$\sigma(\nu_{IF}) = \frac{1}{\sqrt{B t_{sky}}} T_{sys}(\nu_{IF}), \quad (7.1)$$

where

$$T_{sys}(\nu_{IF}) = T_{sky}(\nu_{IF}) + T_{rx}(\nu_{IF}) \quad (7.2)$$

and $B=61\text{ kHz}$ is the channel bandwidth of the spectrometer. The standard deviation spectrum is plotted against the right-hand axis in Figure 7.1.

Fit data to be read by *am* can include up to four fields per spectral data point. These are the frequency ν_i , spectral data s_i , channel bandwidth b_i , and a weighting factor $w_i=\sigma_i^{-1}$, arranged by columns in an input file. The fields ν_i and s_i are required. The bandwidth b_i is required to be in the same units as ν_i , and the units of w_i are the reciprocal of the units of s_i . If the bandwidth and weighting factor fields are omitted, they default to $b_i=0$ and $w_i=1$, respectively. As implied above, the weighting factor w_i is normally the reciprocal of the estimated standard deviation of the corresponding spectral datum s_i . The purpose of the bandwidth field is to provide an averaging bandwidth for the forward model, as might be required when the radiometer provides broad channels, of possibly varying width, across which s_i cannot be assumed constant. Apart from this, b_i is not used in any noise computation—the per-channel noise estimate, including the effect of channel bandwidth, is explicitly provided via w_i .

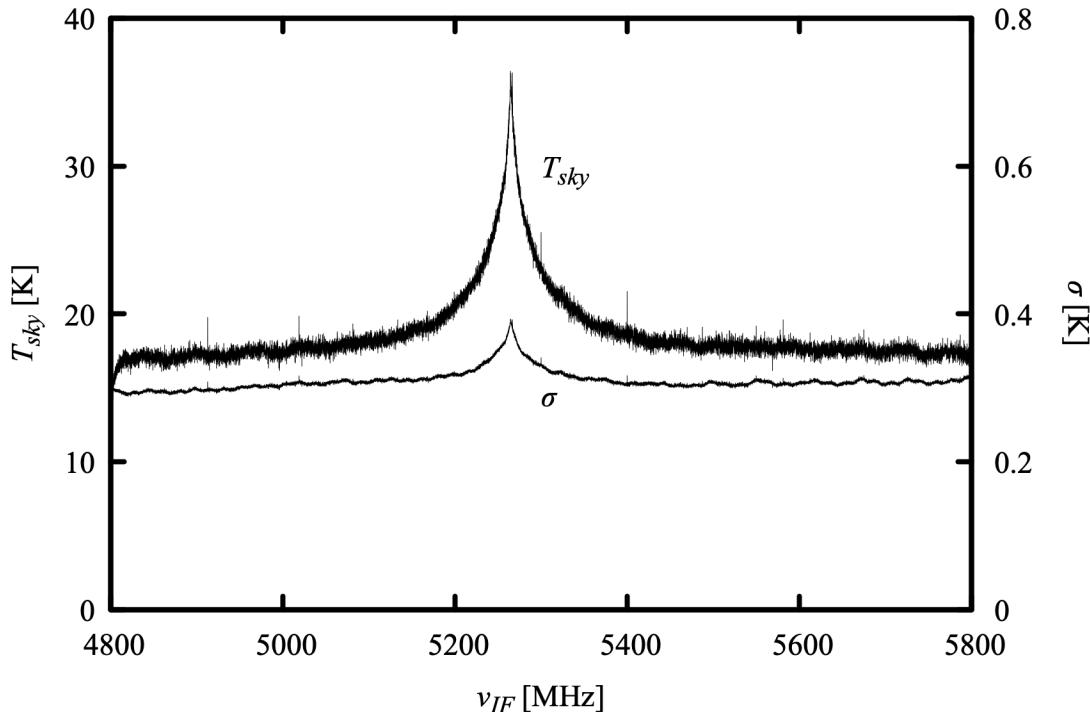


Figure 7.1 Input data for the fit in Example 7.1—calibrated double-sideband Rayleigh-Jeans brightness temperature spectrum, observed at a zenith angle of 5 degrees, and its associated standard deviation spectrum. Data were acquired at the SMA on 2010 September 8 at 19:49 UT.

For the present example, the atmospheric spectrum is completely resolved by the narrow spectral channels of the spectrometer, so the b_i field is unnecessary—only the ν_i , s_i , and w_i fields are needed. A sample of the data file, `example7.1.dat`, looks like this:

```
.
.
4921.7652 17.1206 3.38090
4921.8263 17.2327 3.37567
4921.8873 16.9435 3.38946
4921.9483 17.0716 3.38393
4922.0094 17.6600 3.35662
4922.0704 17.0672 3.38432
4922.1314 17.2868 3.37419
4922.1925 16.9548 3.38985
4922.2535 16.9734 3.38897
.
.
```

where the columns are ν_{IF} in MHz, $T_{sky}(\nu_{IF})$ in K, and $\sigma^{-1}(\nu_{IF})$ in K^{-1} . Here, the data are arranged in ascending frequency order, but this is not necessary—the fit data can be unordered, and redundant spectral points are allowed.

7.1.2 Setting up the model

With the data file in hand, the next step is to set up the model configuration file. In this section, we describe the configuration statements which precede the layer data and the data sources for the layer data themselves, followed by a presentation of the complete configuration file.

The first three statements in the configuration file will define the model frequency grid, IF spectrum parameters, and output frequency range for the IF spectrum:

```
f 220 GHz 232 GHz 0.1 MHz
ifspec dsb 226.01656 GHz
fif 4825 MHz 5775 MHz
```

The `ifspec` and `fif` statements were introduced in Section 5.3.1; further information is also given in Appendix B. We are restricting the IF spectrum to 4825 MHz to 5775 MHz even though the data run from 4800 MHz to 5800 MHz, to allow for a 25 MHz guard band at each end of the spectrum. This avoids frequencies distorted by the filters which define the spectrometer bandpass. Data points outside the model output frequency range are ignored.

Next, an `output` statement is used to set two program outputs, opacity and Rayleigh-Jeans brightness temperature:

```
output f MHz tau Trj K
```

Although `Trj` happens in this case to be the model spectrum to which the data will be fitted, the program outputs and the fitted spectrum are independent. It is possible to fit one spectrum type, but to compute a different set of spectra, using the converged fit parameters, as the final program output. At each iteration during the fit, only the spectrum being fitted is computed—to save time, computation of other spectra is turned off until the final computation using the converged fit parameters.

The next two model configuration statements are a `za` statement to set the zenith angle to $\theta_z = 5^\circ$ and a statement to set the tolerance on line-by-line computations to 10^{-4} :

```
za 5 deg
tol 1e-4
```

This tolerance is appropriate given the signal-to-noise ratio of this spectrum, and will speed up the initial model computation. Since this fit involves varying column densities only, however, the tolerance setting will have no effect on the speed of the fit once it is under way. The fitting process is controlled by the next group of statements:

```
fit Trj %1
fit_data_columns 1 2 0 3
fit_data_units MHz K
fit_verbose 1
```

The first of these, the `fit` statement, is the key statement which turns an *am* model configuration file into a fit file. It specifies the output spectrum to which the spectral data are fitted, in this case `Trj`, and specifies the file which serves as the data source. Here, the data file name is represented

by the command-line parameter `%1`, rather than being hard-coded into the file, since we intend to apply this fit to a number of data files. `am` will search for the data file in the directory named by the environment variable `AM_FIT_INPUT_PATH` if this variable has been defined, or in the current directory otherwise. The special file names `stdin` or `-` can be used to signify that the data is to be read from standard input rather than from a file. An analogous `fits` statement takes a file containing a list of file names, each file containing spectral data to be fit. More details are in Appendix B.

The `fit_data_columns` statement defines the data file columns; its arguments are, respectively, the column numbers corresponding to ν_i , s_i , b_i , and w_i . Here, we are not using the bandwidth field, so b_i is assigned a column number of 0. For the simple space-delimited data in this example, the `fit_data_columns` statement is all we need to read the file. Other formats, such as fixed-width or comma-delimited data, can be read by defining the format using the `fit_data_format` and `fit_data_delimiters` statements described in Appendix B.

Besides specifying how the spectral data are to be read, it is also necessary to specify the data units. The first argument of the `fit_data_units` statement gives the units of ν_i and b_i , while the second argument gives the dimensions of s_i and w_i^{-1} . Table B.10 gives the complete list of units recognized by `am`.

By default, `am` works silently during fits. The `fit_verbose` statement here turns on status reporting of the simplex vertices and fit statistics after each iteration of the fit. The status report, described below in the next section, is written to `stderr`.

The next two statements in this example define the two fit variables. These are a scale factor on the column densities for the entire ozone profile, and a scale factor on the water vapor profile in the troposphere only. The scale factors are implemented using the `nscale` statement, which was introduced in Section 2.2.5. Recall that a statement like

```
NSCALE troposphere h2o 1.2
```

applies a global scale factor, in this case 1.2, to the column density of all columns named `h2o` on all layers with the tag `troposphere` in the model. If no layer tag is given, the scale factor applies to all layers. To make the scale factor into a fit variable, one simply adds a second numerical argument, which gives the range of this variable over the fit simplex at the start of the downhill simplex minimization. In this case, we want to start with initial guess values of unity for the scale factors applied to the ozone and tropospheric water vapor profiles, and start with an initial simplex size spanning a 5% range on these scale factors. The statements are

```
NSCALE o3 1.0 0.05
NSCALE troposphere h2o 1.0 0.05
```

Adding an extra argument in this manner is the general method for designating a model parameter as a fit variable in an `am` fit configuration file. For dimensioned parameters, the extra argument must also carry dimensions, as in the statement

```
Tbase 285 K 2 K
```

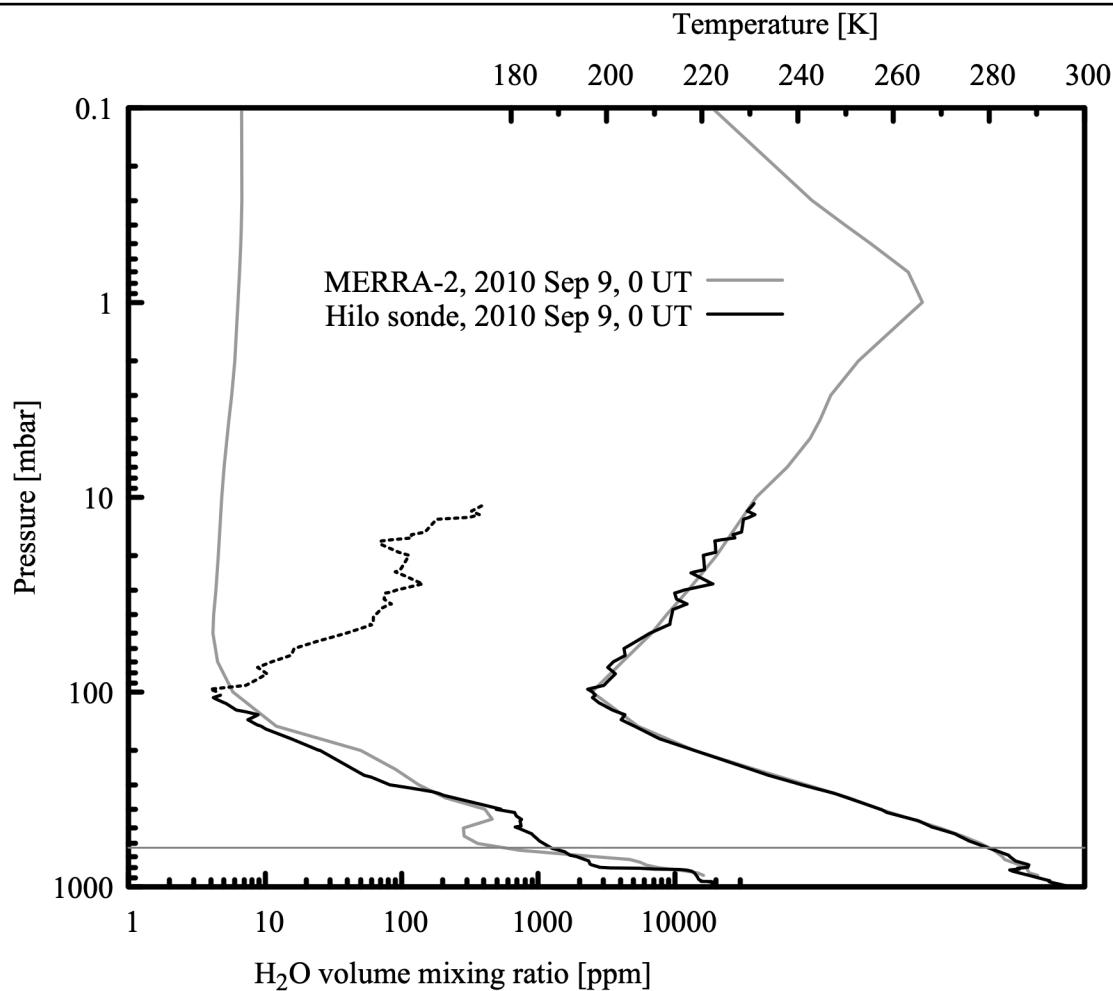


Figure 7.2 Temperature and water vapor mixing ratio profiles over Mauna Kea Hawaii on 2010 Sep 9. The profiles plotted in black are from the Hilo radiosonde launched at 0 UT, and those plotted in gray are from the MERRA-2 reanalysis. Exiting the cold tropopause, the sonde's humidity sensor is slow to respond towards zero RH and eventually retains a positive offset, resulting in a spurious water vapor profile (dotted line) that tracks the rising temperature profile. The more realistic MERRA-2 stratospheric water vapor profile is constrained by satellite climatologies.

which would designate a layer base temperature as a fit variable.

The remainder of the model configuration file consists of a `to` statement to set the cosmic background temperature, followed by the layer descriptions derived from a combination of data from the Hilo radiosonde launched at 0 UT, about 4 hours after the spectrum was acquired, and contemporaneous data from the NASA MERRA-2 reanalysis (Rienecker et al. 2011, Molod et al. 2015). The temperature and water vapor profiles from these two sources are plotted together in

Figure 7.2. They are not entirely independent—the Hilo sonde data will have been included among the many data sources statistically assimilated into the reanalysis. Our model uses the radiosonde profiles below the tropopause, and the MERRA-2 profiles above. In the troposphere, we favor the in-situ radiosonde measurements over the reanalysis model. However, the radiosonde's humidity sensor does not provide a useful measurement in the dry stratosphere above the tropopause, where MERRA-2 offers a more realistic profile constrained by satellite climatologies. The routine soundings from Hilo do not measure ozone, so we use the MERRA-2 ozone profile over the entire depth of the atmosphere. Finally, at the base of the model, we use the surface pressure and temperature measured at the SMA. The complete model file is listed below in its entirety, including comments referencing the layer data sources:

```
# Example 7.1 - fitting ozone and water vapor column density to a spectrum
# observed on Mauna Kea.

f 220 GHz 232 GHz 0.1 MHz
ifspec dsb 226.01656 GHz
fif 4825 MHz 5775 MHz
output f MHz tau Trj K
za 5 deg
tol 1e-4

fit Trj %1
fit_data_columns 1 2 0 3
fit_data_units MHz K
fit_verbose 1

Nscale o3 1.0 0.05
Nscale troposphere h2o 1.0 0.05

T0 2.7 K

# Temperature and water vapor profiles above the 100 mbar level, and the
# ozone profile across all layers, are from the NASA MERRA-2 reanalysis
# for 0 UT on 2010 Sep 9, interpolated to the position of Mauna Kea.
#
# MERRA references:
#   M. M. Rienecker et al. (2011), MERRA: NASA's Modern-Era Retrospective
#   Analysis for Research and Applications. J. Climate 24:3624.
#
#   A. Molod et al (2015), Development of the GEOS-5 atmospheric general
#   circulation model: evolution from MERRA to MERRA2. Geoscience Model
#   Development 8:1339.
#
# MERRA is produced by the NASA/Goddard Global Modeling and Assimilation
# Office (GMAO). MERRA-2 data are archived and distributed by the Goddard
# Earth Sciences Data and Information Services Center (GES DISC). The
# data access page is at
#
#   http://disc.sci.gsfc.nasa.gov/uui/search/%22MERRA-2%22
#

layer mesosphere
Pbase 0.1 mbar
Tbase 222.0 K
lineshape Voigt-Kielkopf
column dry_air vmr
column h2o vmr 6.7315e-06
column o3 vmr 5.9130e-07
```

```
layer mesosphere
Pbase 0.3 mbar
Tbase 243.0 K
lineshape Voigt-Kielkopf
column dry_air vmr
column h2o vmr 6.7456e-06
column o3 vmr 8.7350e-07

layer mesosphere
Pbase 0.4 mbar
Tbase 249.8 K
lineshape Voigt-Kielkopf
column dry_air vmr
column h2o vmr 6.7310e-06
column o3 vmr 1.3082e-06

layer mesosphere
Pbase 0.5 mbar
Tbase 255.3 K
lineshape Voigt-Kielkopf
column dry_air vmr
column h2o vmr 6.6693e-06
column o3 vmr 1.6147e-06

layer mesosphere
Pbase 0.7 mbar
Tbase 263.1 K
lineshape Voigt-Kielkopf
column dry_air vmr
column h2o vmr 6.5669e-06
column o3 vmr 2.0567e-06

layer stratosphere
Pbase 1.0 mbar
Tbase 266.1 K
lineshape Voigt-Kielkopf
column dry_air vmr
column h2o vmr 6.4133e-06
column o3 vmr 2.6951e-06

layer stratosphere
Pbase 2.0 mbar
Tbase 252.7 K
column dry_air vmr
column h2o vmr 6.1639e-06
column o3 vmr 4.1269e-06

layer stratosphere
Pbase 3.0 mbar
Tbase 246.9 K
column dry_air vmr
column h2o vmr 5.8462e-06
column o3 vmr 6.0874e-06

layer stratosphere
Pbase 4.0 mbar
Tbase 244.7 K
column dry_air vmr
column h2o vmr 5.5649e-06
column o3 vmr 7.4094e-06

layer stratosphere
Pbase 5.0 mbar
```

```
Tbase 242.6 K
column dry_air vmr
column h2o vmr 5.3478e-06
column o3 vmr 8.1292e-06

layer stratosphere
Pbase 7.0 mbar
Tbase 237.8 K
column dry_air vmr
column h2o vmr 5.1354e-06
column o3 vmr 8.6916e-06

layer stratosphere
Pbase 10.0 mbar
Tbase 231.3 K
column dry_air vmr
column h2o vmr 4.9131e-06
column o3 vmr 8.9839e-06

layer stratosphere
Pbase 20.0 mbar
Tbase 223.0 K
column dry_air vmr
column h2o vmr 4.6785e-06
column o3 vmr 8.0375e-06

layer stratosphere
Pbase 30.0 mbar
Tbase 217.1 K
column dry_air vmr
column h2o vmr 4.4530e-06
column o3 vmr 5.7961e-06

layer stratosphere
Pbase 40.0 mbar
Tbase 212.6 K
column dry_air vmr
column h2o vmr 4.2834e-06
column o3 vmr 3.5909e-06

layer stratosphere
Pbase 50.0 mbar
Tbase 209.4 K
column dry_air vmr
column h2o vmr 4.1749e-06
column o3 vmr 2.1261e-06

layer stratosphere
Pbase 70.0 mbar
Tbase 203.3 K
column dry_air vmr
column h2o vmr 4.3158e-06
column o3 vmr 1.1879e-06

layer troposphere
Pbase 100.0 mbar
Tbase 196.9 K
column dry_air vmr
column h2o vmr 5.1417e-06
column o3 vmr 4.7630e-07

# Layer data below the 100 mbar level use temperature and water vapor
# sounding data obtained from the University of Wyoming Department
# of Atmospheric Sciences archive:
```

```
#  
#   http://weather.uwyo.edu/upperair/sounding.html  
#  
# Sounding ID:  
#  
#   91285 PHTO Hilo Observations at 00Z 09 Sep 2010  
#  
# Layer boundaries here have been interpolated between the nearest sounding  
# levels. H2O mixing ratios are a hydrostatically-weighted average across  
# all intervening sounding layers.  
  
layer troposphere  
Pbase 150.0 mbar  
Tbase 205.8 K  
column dry_air vmr  
column h2o vmr 6.6012e-06  
column o3 vmr 1.1315e-07  
  
layer troposphere  
Pbase 200.0 mbar  
Tbase 218.4 K  
column dry_air vmr  
column h2o vmr 1.6319e-05  
column o3 vmr 5.9750e-08  
  
layer troposphere  
Pbase 250.0 mbar  
Tbase 230.2 K  
column dry_air vmr  
column h2o vmr 3.4939e-05  
column o3 vmr 4.2200e-08  
  
layer troposphere  
Pbase 300.0 mbar  
Tbase 240.8 K  
column dry_air vmr  
column h2o vmr 6.1228e-05  
column o3 vmr 3.4450e-08  
  
layer troposphere  
Pbase 350.0 mbar  
Tbase 250.3 K  
column dry_air vmr  
column h2o vmr 1.6145e-04  
column o3 vmr 3.1950e-08  
  
layer troposphere  
Pbase 400.0 mbar  
Tbase 257.2 K  
column dry_air vmr  
column h2o vmr 3.8779e-04  
column o3 vmr 3.2100e-08  
  
layer troposphere  
Pbase 450.0 mbar  
Tbase 263.9 K  
column dry_air vmr  
column h2o vmr 6.6104e-04  
column o3 vmr 3.4550e-08  
  
layer troposphere  
Pbase 500.0 mbar  
Tbase 268.8 K  
column dry_air vmr
```

```

column h2o vmr 7.3386e-04
column o3 vmr 3.6950e-08

layer troposphere
Pbase 550.0 mbar
Tbase 273.8 K
column dry_air vmr
column h2o vmr 8.3523e-04
column o3 vmr 3.8550e-08

layer troposphere
Pbase 600.0 mbar
Tbase 277.3 K
column dry_air vmr
column h2o vmr 1.0048e-03
column o3 vmr 4.0050e-08

layer troposphere
Pbase 626.7 mbar # SMA surface P, T at time of observation
Tbase 283.45 K
column dry_air vmr
column h2o vmr 1.1734e-03
column o3 vmr 4.1392e-08

```

7.1.3 Running the fit

With the data file and configuration file as described above, we can now run the fit. Recall that we made the data file name a command line parameter, so its name appears after the configuration file name on the command line. With the `fit_verbose` flag set, the run appears as follows:

```

$ am example7.1.amc example7.1.dat

file example7.1.dat : block 0
iter 0 : ds 1.41421 (0.0001) : time 18.827 s
vtx      E          x0          x1
-----
H 0    4.3808e+04   1.0000e+00   1.0000e+00
     1    3.9974e+04   1.0500e+00   1.0000e+00
L 2    1.7803e+04   1.0000e+00   1.0500e+00
-----
max - min   5.0000e-02   5.0000e-02

file example7.1.dat : block 0
iter 1 : ds 2.14762 (0.0001) : time 18.838 s
vtx      E          x0          x1
-----
L 0    1.7229e+04   1.0759e+00   1.0759e+00
H 1    3.9974e+04   1.0500e+00   1.0000e+00
     2    1.7803e+04   1.0000e+00   1.0500e+00
-----
max - min   7.5930e-02   7.5930e-02
.
.
.
file example7.1.dat : block 0
iter 79 : ds 0.000119054 (0.0001) : time 19.593 s
vtx      E          x0          x1
-----
L 0    1.4184e+04   1.0824e+00   1.0570e+00
H 1    1.4184e+04   1.0824e+00   1.0570e+00

```

```

2   1.4184e+04   1.0824e+00   1.0570e+00
----- -----
max - min   5.7058e-06   1.6966e-06

file example7.1.dat : block 0
iter 80 : ds 6.8314e-05 (0.0001) : time 19.603 s
vtx      E          x0          x1
-----
L 0   1.4184e+04   1.0824e+00   1.0570e+00
    1   1.4184e+04   1.0824e+00   1.0570e+00
H 2   1.4184e+04   1.0824e+00   1.0570e+00
----- -----
max - min   3.2462e-06   1.0627e-06

converged after restart 1 : dc 1.71659e-15 (0.0001)

! Warning: Double-sideband opacity spectra are only meaningful in
!           the optically-thin limit, or if tau(usb) and tau(lsb)
!           are approximately equal.

$
```

The warning at the end was generated because we asked for $\tau(\nu)$ as one of the final output spectra. We can safely ignore it since we are indeed in the optically-thin limit. In verbose mode, a summary table is written to `stderr` after each fit iteration. This gives the coordinates of the simplex vertices in fit variable space, and the value of the fit estimator at each simplex vertex. Simplex coordinates are displayed in internal program units, and listed in the order they appear in the configuration file. Here, `x0` is the `nscale` parameter for `o3`, and `x1` is `nscale` for `h2o`. The column labeled `E` is the value of the fit estimator function at each vertex, with the high and low values indicated by `H` and `L`. When not otherwise specified in the configuration file, the fit estimator function defaults to the weighted sum of the absolute residuals. For details, see the entry under `fit_estimator` in Appendix B.

Also shown in the summary data is the current scaled diameter d_s of the fit simplex, defined as

$$d_s = \sqrt{\sum_i \left(\frac{x_{i,max} - x_{i,min}}{x_{i,scale}} \right)^2}, \quad (7.3)$$

where the index i runs over all the fit parameters, $x_{i,max}$ and $x_{i,min}$ are the maximum and minimum values of fit variable x_i over the fit simplex, and $x_{i,scale}$ is the characteristic scale supplied by the user as the range of x_i over the initial simplex at the start of the fit. The fit is deemed to have converged when $d_s < \epsilon$, where ϵ is a convergence tolerance. The numerical value of ϵ is shown in parenthesis next to d_s in the summary table written after each iteration. The default is $\epsilon=10^{-4}$ which can be changed using the `fit_tol` statement described in Appendix B. Upon convergence, the minimization is restarted from a new simplex with the same initial size as before but starting at the converged values of the fitted parameters from the prior iteration. If the restarted fit converges to a parameter set at a scaled distance $d_c < \epsilon$ from the previous converged parameter set (where d_c is defined analogously to d_s) then the fit is deemed to have fully converged. The maximum number of restarts (default 5) can also be set using the `fit_tol` statement.

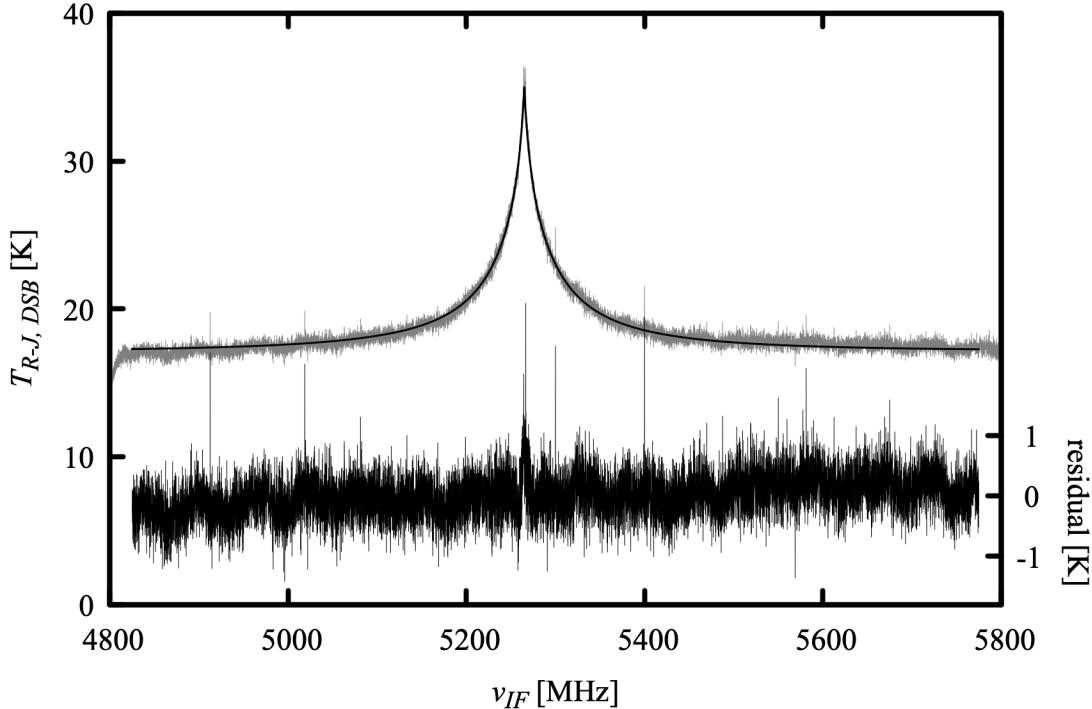


Figure 7.3 Original data (gray), model fit (black, left axis), and fit residuals (black, right axis) from Example 7.1. The reduced chi-squared statistic for this fit, based on the expected radiometer noise for the 1 s integration time, is $\chi_r^2=1.323$.

At the end of the fit, *am* writes three output files in the directory named by the environment variable `AM_FIT_OUTPUT_PATH`, if it exists, or in the current directory otherwise. The names of these files are formed by appending `.ams`, `.amc`, and `.amr` to the data file name. The file `datafile.ams` contains the usual spectral data, controlled by the `output` statement, just as *am* writes to `stdout` when not running a fit. The file `datafile.amc` contains the configuration data that *am* normally writes back to `stderr`, including additional data about the fit. These additional data include a line reporting the reduced chi-squared statistic,

$$\chi_r^2 = \frac{\chi^2}{N-M} = \frac{1}{N-M} \sum_i (w_i r_i)^2 \quad (7.4)$$

when weights w_i have been provided, or simply the standard deviation of the residuals

$$\sigma = \sqrt{\frac{1}{N-M} \sum_i r_i^2} \quad (7.5)$$

otherwise. Here, the $r_i = s_i - s_i^{mod}$ are the residual differences between the measured and model spectrum, N is the number of spectral channels fitted, and M is the number of fit variables. The

third output file, `datafile.amr`, contains five columns; in order these are ν_i , r_i , s_i^{mod} , b_i , and w_i .

Figure 7.3 is a plot of s_i^{mod} and r_i from the output file `example7.1.dat.amr`, along with the original data from `example7.1.dat`. The converged values of the fitted parameters are contained in the file `example7.1.dat.amc`. Leaving out the layer data, the contents of this file are:

```
# am version 14.0 (build date Sep 26 2024 13:20:00)

# fit run time 19.608 s (80 iterations)
# kcache hit: 0 miss: 0 discard: 0
# dcache hit: 0 miss: 290 discard: 0
# 15565 data points, 2 fit variables
#
# var           init      converged   units  description
#  0             1          1.08244    none   Nscale o3
#  1             1          1.05705    none   Nscale troposphere h2o
#
# reduced chi-squared 1.323

# fit Trj example7.1.dat
fit_data_columns 1 2 0 3
fit_data_delimiters " " "# " \t"
fit_data_units MHz K
fit_estimator absolute_residuals (1.418e+04)
fit_iter 1000
fit_reinit 0
fit_output config 1
fit_output spectrum 1
fit_output residual 1
fit_tol 0.0001 5 (6.831e-05)
fit_track_residuals off
fit_verbose 1
kcache 190 K 320 K 1 K
simplex_log 1

f 220 GHz 232 GHz 0.1 MHz
ifspec dsb 226.01656 GHz
fif 4825 MHz 5775 MHz
output text f MHz tau neper Trj K

dsb_utol_ratio 1
tol 0.0001
selfbroad_vmr_tol 0.003

geometry plane-parallel
refract none
za 5 deg
PTmode Pbase Tbase

g 980.665 cm*s^-2
dg_dz 0 s^-2

Nscale o3 1.08244 0.05 (3.25e-06)
Nscale troposphere h2o 1.05705 0.05 (1.06e-06)

.
.
.

#
```

```

# column densities [cm^-2], including scale factors
#
#               zenith           line-of-sight
# mesosphere (5 layers):
#       dry_air 1.484113e+22   1.489782e+22
#       h2o    9.913275e+16   9.951142e+16
#             (0.0296565 um_pwv) (0.0297698 um_pwv)
#       o3    2.151442e+16   2.159660e+16
#             (0.800745 DU)   (0.803804 DU)
#
# stratosphere (12 layers):
#       dry_air 1.469266e+24   1.474878e+24
#       h2o    6.616440e+18   6.641713e+18
#             (1.97937 um_pwv) (1.98693 um_pwv)
#       o3    6.659063e+18   6.684500e+18
#             (247.844 DU)   (248.79 DU)
#
# troposphere (12 layers):
#       dry_air 1.179985e+25   1.184493e+25
#       h2o    5.081044e+21   5.100453e+21
#             (1520.04 um_pwv) (1525.85 um_pwv)
#       o3    8.854577e+17   8.888400e+17
#             (32.9558 DU)   (33.0817 DU)
#
# total (29 layers):
#       dry_air 1.328396e+25   1.333470e+25
#       h2o    5.087760e+21   5.107194e+21
#             (1522.05 um_pwv) (1527.87 um_pwv)
#       o3    7.566035e+18   7.594936e+18
#             (281.6 DU)     (282.676 DU)
#

```

Note that the fitted water vapor and ozone profile scale factors are close to unity, and the reduced chi-squared statistic is close to the radiometer-noise limited value $\chi^2_r=1$, suggesting that the original layer data were a reasonable guess for the atmospheric state and that the effects of other systematic factors such as baseline ripple are modest. Recall that our goal is to compare the retrieved column densities with other data, and to estimate the statistical uncertainties of the retrieved parameters, which is the subject of the next section.

7.1.4 Bootstrap analysis of statistical uncertainties

The classic method of estimating the statistical uncertainties of the fitted parameters involves forming a quadratic Taylor approximation, around the local minimum χ^2_{min} at the converged fit parameter values, of the χ^2 surface in fit parameter space. Then, under the assumption that the residuals are normally distributed, the standard errors of the fitted parameters are the projections onto the fit variable axes of the generalized ellipsoid defined by $\chi^2 - \chi^2_{min} = 1$. In computing the required Hessian matrix of second derivatives of χ^2 with respect to the fit variables, the usual care must be taken to manage rounding and other issues associated with numerical differentiation.

As an alternative, the *bootstrap method* is a simple, robust means of estimating fitted parameter uncertainties which avoids assumptions about the statistical distribution of the residuals, and also avoids the need to compute numerical derivatives. Essentially, the method

consists of forming randomly-resampled data sets from the original data, rerunning the fit on each such set, and building up the resulting distribution of the converged values of the fit variables by brute force. Arguably, the method is more rigorous than the classical χ^2 method, since the statistics of the residuals are determined from analysis of the data, rather than being assumed, *a priori*, to be Gaussian normal. The cost is considerably more computation. For a good discussion of the bootstrap method, see Press, et al. (2007).

Implementation of the bootstrap method in *am* fits is very simple, using the streaming fit capability built into the program together with the Unix tools `awk` and `sort`. To use *am* in streaming mode, the input data file is set to `stdin` and a data source supplying a character stream in the same format as a normal data file is piped into the program. A single line in the input stream, containing the word `end`, marks the end of a data block; upon encountering the end of a block, *am* runs a fit, writes the output, then returns to the input stream to read another block of fit data. The usual three output files are replaced by three files named `stdin.ams`, `stdin.amc`, and `stdin.amr`, with the output data from successive fits appended to one another. In addition to the output files, a single summary line containing χ^2_r and the converged values of the fit variables is written to `stdout` at the end of each fit. If the output files are not needed, file writing can be turned off, as in the following example, which is `example7.1.amc` adapted to streaming mode:

```
# Example 7.2 - fitting ozone and water vapor column density to a spectrum
# observed on Mauna Kea. Streaming fit with outputs turned off.

f 220 GHz 232 GHz 0.1 MHz
ifspec dsb 226.01656 GHz
fif 4825 MHz 5775 MHz
output f MHz tau Trj K
za 5 deg
tol 1e-4

fit Trj stdin # set input file to stdin
fit_data_columns 1 2 0 3
fit_data_units MHz K
fit_verbose 0 # turns off verbose output
fit_output config 0 # turns off writing of stdin.amc
fit_output spectrum 0 # turns off writing of stdin.ams
fit_output residual 0 # turns off writing of stdin.amr

Nscale o3 1.0 0.05
Nscale troposphere h2o 1.0 0.05

T0 2.7 K
.
```

The remainder of the file is identical to `example7.1.amc`.

Next, we need a random data set generator. The following `awk` program reads a data file, then writes out 1000 data sets of the same length, chosen randomly with replacement from the original data file. The end of each randomly-selected data set is marked with a line containing the word `end`:

```
# rnd_select.awk - generate nsets randomly-selected data sets from a file
```

```

BEGIN { nsets = 1000; seed = 0 }

{ line[NR] = $0

END {
    srand(seed)
    while(nsets--) {
        for (i = 0; i < NR; ++i)
            print line[1 + int(NR * rand())]
        print "end"
    }
}

```

Putting it all together, the command to run the bootstrap analysis is

```
$ awk -f rnd_select.awk example7.1.dat | am example7.2.amc > example7.2.log
```

The output file, `example7.2.log`, contains the fit summary lines for each fit, written to `stdout` by `am`. The contents of this file are:

0	78	1.310894e+00	1.08206	3.90e-06	1.05761	2.10e-06	
1	70	1.305889e+00	1.08397	4.68e-06	1.05695	1.33e-06	
2	72	1.327160e+00	1.08044	3.50e-06	1.05706	1.32e-06	
3	70	1.306844e+00	1.08348	3.10e-06	1.05681	1.69e-06	
4	78	1.330558e+00	1.08029	2.57e-06	1.05719	5.49e-07	
.							
.							
.							
995	80	1.326858e+00	1.08394	3.57e-06	1.05688	1.46e-06	
996	81	1.307278e+00	1.07948	3.31e-06	1.05780	6.62e-07	
997	79	1.368397e+00	1.08112	2.96e-06	1.05725	2.01e-07	
998	80	1.321175e+00	1.08267	3.77e-06	1.05686	1.60e-06	
999	70	1.313849e+00	1.07946	2.85e-06	1.05702	3.61e-06	

From left to right, the columns are fit number, number of iterations to convergence, and χ^2_r , followed by two columns for each fit variable in the order the fitted parameters occur in the configuration file. These two columns are the converged parameter value and its range over the converged fit simplex. In this case, column 4 is `nscale` for `o3`, and column 6 is `nscale troposphere` for `h2o`. These can be converted to zenith column density by multiplying by the respective column densities of total O₃ and tropospheric H₂O in the initial guess profiles, which can be derived from the summary data at the end of the Example 7.1 fit. They are 281.6 / 1.08244 = 260.153 Dobson units (DU) for the O₃ total column density, and 1520.04 / 1.05705 = 1438.00 precipitable μm for tropospheric H₂O. To get the total H₂O column density, we also need to add in the unscaled column density (2.01 precipitable μm) from the stratosphere and mesosphere. The fitted O₃ and H₂O scale factors and corresponding column densities for the 1000 bootstrap trials are plotted in Figure 7.4.

A plot like Figure 7.4 is useful as a qualitative presentation of the joint distribution of the two fitted parameters. For a quantitative assessment, we can compute the cumulative distribution functions of the fitted column densities, using the unix tools `sort` and `awk`:

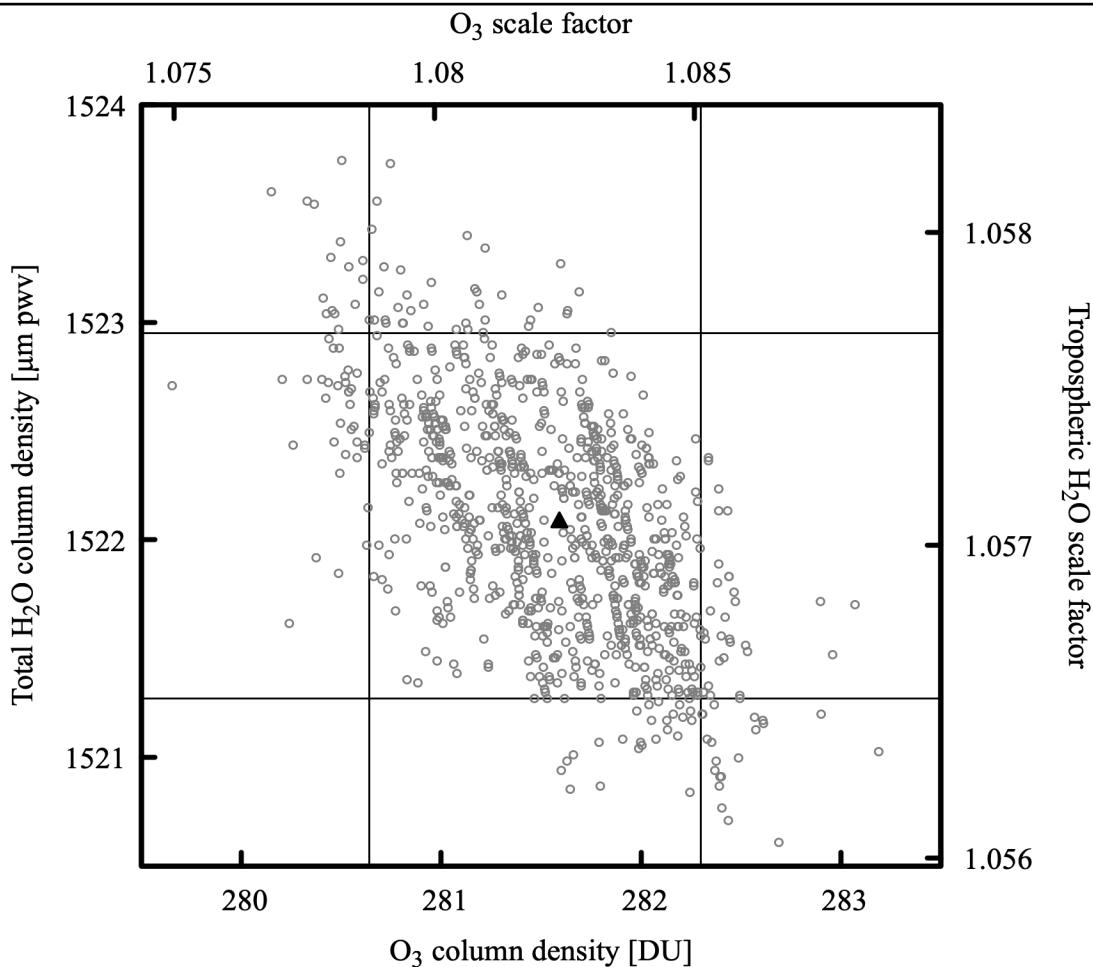


Figure 7.4 Fitted ozone and water vapor column densities for 1000 bootstrap trials from Example 7.2. The light gray lines indicate the 5th and 95th percentile values for the fitted parameters, and the filled triangle marks the intersection of the median values

```
sort -g -k 4 example7.2.log | awk '{print $4 * 260.153, NR / 1000.}' \
> o3.cdf
sort -g -k 6 example7.2.log | awk '{print $6 * 1438.00 + 2.01, NR / 1000.}' \
> h2o.cdf
```

These commands simply put the fitted column densities into sorted order, and count off the cumulative distribution. For example, the contents of the file o3.cdf are

```
279.654 0.001
280.148 0.002
280.203 0.003
280.239 0.004
280.258 0.005
.
.
```

```
.
282.898 0.996
282.901 0.997
282.958 0.998
283.072 0.999
283.19 1
```

A 90% statistical confidence range can then be found by reading off the 5th and 95th percentile values of the fitted column densities from these files. These are plotted as the gray lines in Figure 7.4, and the intersection of the median values is indicated by the filled triangle. In summary, for this 1 s integration, the retrieved ozone and water vapor columns and their 90% statistical confidence intervals are 281.6 ± 0.9 DU and 1522.1 ± 0.9 μm pwv, respectively. This characterizes the sensitivity of the observation; systematic accuracy is assessed next.

7.1.5 Systematic calibration accuracy

The foregoing analysis satisfied our first goal of characterizing the statistical errors of the ozone and water vapor column density retrieval. The systematic calibration accuracy can be investigated by comparing the retrieved ozone column density with an independent satellite measurement. This will be ozone column density data provided by the Tropospheric Emission Monitoring Internet Service (TEMIS, <http://www.temis.nl>), in particular, data from the SCIAMACHY sensor (Bovensmann, et al. 1999) on the ENVISAT-1 satellite, assimilated into a global ozone model providing daily point estimates of total O₃ column (Eskes, et al. 2003). Interpolated spatially to Mauna Kea, and temporally to the acquisition time for the spectrum in this example, the TEMIS O₃ column density is 285 ± 8 DU, which agrees well with the retrieved value above.

A useful independent check on the much more variable water vapor column density is more difficult to come by. The best available nearby contemporaneous data source was the NRAO 225 GHz tipping radiometer (Liu 1987) operated at the time by the Caltech Submillimeter Observatory. During the 1 hour period bracketing this measurement, this instrument made five tip scans of the opacity τ at 225 GHz; the mean and standard deviation of these measurements was $\tau_{225} = 0.060 \pm 0.010$. To make a comparison, we need first to compute a statistical confidence range for the 225 GHz zenith opacity from the fit. We begin by making the substitutions

```
.
.
f 225 GHz 225 GHz 0.1 MHz
output f MHz tau
za 0 deg
.

.

Nscale o3 %1
Nscale h2o %2
.
```

for the corresponding lines in `example7.1.amc`, removing the `ifspec`, `fif`, and `fit` statements, and calling the new file `tau225.amc`. Running this file, setting `Nscale` for `h2o` and `o3` to their

values at the lower right and upper left intersections of the 5th and 95th percentile lines in Figure 7.4, gives the corresponding range of τ_{225} :

```
$ am tau225.amc 1.08513 1.05651 2>/dev/null
6.934353e-02
$ am tau225.amc 1.07875 1.05768 2>/dev/null
6.940512e-02
```

This is in agreement with the range of 225 GHz radiometer measurements around the time the spectrum was acquired, although given the large fractional variation in the tipping radiometer measurements, they do not offer a very tight constraint.

7.2 More about fits

More information about fits in *am*, including features not covered in the example presented in this chapter, can be found in Appendix B. Of particular interest are the `kcache` facility, which is used to maintain an interpolation table for fast computation of absorption coefficient arrays on layers where temperature is a fit variable, and the `set` facility, which allows certain model configuration statements to be inserted between blocks of data during streaming fits.

References

- H. Bovensmann, J. P. Burrows, M. Buchwitz, J. Frerick, S. Noël, V. V. Rozanov, K. V. Chance, and A. P. H. Goede 1999, “SCIAMACHY: Mission Objectives and Measurement Modes.” *J. Atmos. Sci.* 56:127.
- H. Eskes, P. van Velthoven, P. Valks and H. Kelder 2003, “Assimilation of GOME total ozone satellite observations in a three-dimensional tracer transport model.” *Q. J. R. Meteorol. Soc.* 129: 1663.
- T. G. Kolda, R. M. Lewis, and V. Torczon 2003, “Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods.” *SIAM Review* 45:385.
- Liu, Zhong-Yi 1987, “225-GHz Atmospheric Receiver–User’s Manual.” Electronics Division Internal Report 271, NRAO.
- A. Molod, L. Takacs, M. Suarez, and J. Bacmeister 2015, “Development of the GEOS-5 atmospheric general circulation model: evolution from MERRA to MERRA2.” *Geosci. Model Dev.* 8:1339.
- J. A. Nelder and R. Mead 1965, “A simplex method for function minimization.” *Computer Journal* 7:308.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery 2007, *Numerical Recipes*, Third Edition. Cambridge University Press.

M. M. Rienecker, M. J. Suarez, R. Gelaro, R. Todling, J. Bacmeister, E. Liu, M. G. Bosilovich, S. D. Schubert, L. Takacs, G.-K. Kim, S. Bloom, J. Chen, D. Collins, A. Conaty, A. da Silva, W. Gu, J. Joiner, R. D. Koster, R. Lucchesi, A. Molod, T. Owens, S. Pawson, P. Pegion, C. R. Redder, R. Reichle, F. R. Robertson, A. G. Ruddick, M. Sienkiewicz, and J. Woollen 2011, “MERRA: NASA’s Modern-Era Retrospective Analysis for Research and Applications.” *J. Climate* 24:3624.

Chapter 8: Jacobians

Often we want to know the sensitivity of a model spectrum to changes in one or more model parameters. Or, we may want to linearize the model's dependence on a set of atmospheric state variables in the neighborhood of some reference state for use in a remote sensing retrieval. For these applications, we need the channel-by-channel partial derivatives of the model spectrum with respect to the model variables of interest. Suppose $\mathbf{F}^T = [F_1(\mathbf{x}), \dots, F_m(\mathbf{x})]$ is our m -channel spectral model at frequencies $\nu_i, i=1, \dots, m$, which depends on a vector $\mathbf{x}^T = [x_1, \dots, x_n]$ of n variable model parameters. Then the $m \times n$ matrix $K_{ij} = \partial F_i / \partial x_j$ of partial derivatives taken at a reference state $\mathbf{x} = \mathbf{x}_0$ is called the *Jacobian* of $\mathbf{F}(\mathbf{x})$ at \mathbf{x}_0 , and the corresponding linearized model is $\mathbf{F}(\mathbf{x}) = \mathbf{F}(\mathbf{x}_0) + \mathbf{K}(\mathbf{x} - \mathbf{x}_0)$. Jacobians can be computed by *am* for any output spectrum computed by the program, including spectra involving instrument models. Given a configuration file describing the model at its reference state, almost any collection of model parameters can be designated as differentiation variables.

A crucial part of numerical differentiation is error estimation. Any floating point computation is subject to roundoff error, and *am* uses computational methods that have the potential to introduce further discontinuities in addition to those caused by floating point roundoff alone. Besides roundoff error, there is also truncation error—an n -point numerical differentiation formula is essentially approximating a function with a polynomial truncated to n terms. To manage the accuracy of Jacobian computations, *am* can optionally output corresponding error estimate spectra, and always includes summary error statistics in the run configuration data written back to `stderr`.

In this chapter, computation of Jacobians with *am* is illustrated using two examples. The first is a computation of the spectral sensitivity of zenith brightness temperature to total water vapor column density in the neighborhood of an H₂O spectral line. The second example is a computation of the Jacobian matrix relating changes in ozone column density on each model layer to changes in zenith brightness temperature at different frequencies in the wing of an O₃ line observed from the ground.

8.1 Computing Jacobians with *am*

Setting up an *am* configuration file for computing Jacobians of one or more spectra involves two steps. Starting with a configuration file describing the model at the reference state, the first step is to add a `jacobian` statement to indicate which model output spectrum or spectra should be differentiated. The syntax is simple: the keyword `jacobian` is followed by a list of model output spectra, as in

```
jacobian I
```

to differentiate a radiance spectrum. The second step is to identify the differentiation variables. To make a model parameter into a differentiation variable, one adds a second (possibly dimensioned) numerical value after the reference state value in the configuration file. So, for example, if the water vapor column density on a particular layer at the reference state is specified as

```
column h2o 2 mm_pwv
```

then changing this to

```
column h2o 2 mm_pwv 2 mm_pwv
```

would make the water vapor column density on that layer a differentiation variable. The first value (`2 mm_pwv`) is the value of the variable at the reference state at which we will compute the Jacobian. The second value (`2 mm_pwv`), besides identifying the variable as a differentiation variable, serves a rough guess of the variable's curvature scale (Press, et al. 2007, section 5.7), which *am* will use to estimate the optimum differentiation step size. The exact value of the curvature scale estimate is not important—a good starting guess is simply the parameter value itself (if non-zero) or its expected physical range of variation. The differentiation step size used internally by *am* will be orders of magnitude smaller than the curvature scale, so it is important not to guess too small. In fact, the optimum curvature scale will be much larger than the reference value for parameters that control correspondingly small contributions to the spectrum being differentiated. The error estimates produced by *am* will tell you if you've guessed appropriately and serve as a guide when making adjustments.

When computing Jacobians, the columns in the output file are ordered as follows. Output spectra (for the model at the reference state) appear in the order given in the `output` statement. For each spectrum being differentiated, the partial derivative with respect to each model parameter defined as a differentiation variable appears in adjacent columns, in the order that the variables appear in the configuration file. For instance, if a configuration file contains the statements

```
output f Tb tau
jacobian Tb
```

and there are two differentiation variables x_1 and x_2 defined in the configuration file, then the output columns will be

$$\nu \quad T_b \quad \frac{\partial T_b}{\partial x_1} \quad \frac{\partial T_b}{\partial x_2} \quad \tau$$

in that order. In this example, the Jacobian matrix **K** consists of the third and fourth columns of the output file.

We can request detailed error estimate spectra by adding the `include_estimated_errors` keyword to the `jacobian` statement, as in

```
jacobian Tb include_estimated_errors
```

If we do so, then for each column of the Jacobian, the estimated rounding and truncation errors will be written in the next two adjacent columns. The order of the output columns in the present example would then become

$$\nu \quad T_b \quad \frac{\partial T_b}{\partial x_1} \quad \epsilon_{round,1} \quad \epsilon_{trunc,1} \quad \frac{\partial T_b}{\partial x_2} \quad \epsilon_{round,2} \quad \epsilon_{trunc,2} \quad \tau,$$

where ϵ_{round} and ϵ_{trunc} are in the same units as the corresponding derivatives. For output files with many columns, it can be helpful to use the `headers` statement described in Appendix B to keep track of output column assignments.

8.2 Examples

In this section, we present two examples illustrating Jacobian computations with *am*. Both examples will be motivated by applications in ground-based remote sensing, and in both cases we'll take Example 2.5, our standard model for Mauna Kea, as a starting point from which to make the necessary modifications. The first example is a simple 1-column Jacobian involving differentiation of a brightness temperature spectrum with respect to a single scale factor on a water vapor profile. The second example will be a Jacobian matrix of brightness temperature differentiated with respect to ozone concentration across a vertical profile.

8.2.1 An example from water vapor radiometry

Ground-based radiometry of the H₂O rotational lines at 22.2 GHz and 183.3 GHz is an established technique for passive remote sensing of the H₂O vertical profile or integrated column density (Gaut 1967; Askne & Westwater 1986). Historically, water vapor radiometry for both meteorological and radio propagation applications developed contemporaneously—humidity is a fundamental meteorological variable, and its nonuniformity and time dependence are the dominant source of variable radio propagation delay in the troposphere. Of the two lines mentioned, the stronger 183.3 GHz line is most suitable for probing the low column densities found at very arid or high altitude sites. For this reason, the 183.3 GHz line was used in early

demonstrations of the use of water vapor radiometry to correct for atmospheric delay affecting submillimeter interferometers (Wiedner, et al. 2001), and 183.3 GHz water vapor radiometers are now part of the facility instrumentation of the ALMA interferometer in Chile (Nikolic et al. 2013).

Taking water vapor radiometry as our motivating application, the example here will be a computation of the zenith brightness temperature spectrum and its Jacobian with respect to total tropospheric water vapor column density in the neighborhood of the 183.3 GHz H₂O line, from the SMA site on Mauna Kea. Such a computation provides the expected channel brightness temperatures and channel sensitivities to total water vapor column density for a hypothetical radiometer. We'll start with a monochromatic (fully resolved) model, and then look at the effect of including a 2 GHz rectangular channel bandwidth.

As a starting point, we'll use the configuration file from Example 2.5, modifying a few lines to change the output and compute the Jacobian. The first few lines of the original Example 2.5 were:

```
.
f %1 %2 %3 %4 %5 %6
output f GHz tau Tb K
za %7 %8
tol 1e-4

Nscale troposphere h2o %11
.
```

To compute Rayleigh-Jeans brightness temperature and its Jacobian with respect to the scale factor on the tropospheric water column we make the following changes:

```
.
f %1 %2 %3 %4 %5 %6
output f GHz Trj K
jacobian Trj
za %7 %8
tol 1e-4

Nscale troposphere h2o %11 1.0
.
```

Here, we've changed the `output` statement to obtain the desired spectrum (`Trj`), and added a `jacobian` statement to request its partial derivatives with respect to any model parameters designated as differentiation variables. We've identified a single model parameter (`Nscale troposphere h2o`) as a differentiation variable by adding a second argument giving its estimated characteristic scale (1.0). Note that we've left the reference state value as a command-line parameter (%11) so we can conveniently compute the Jacobian at different values of the total water vapor column density.

Recall that in Example 2.5, the water vapor profile specified in the layer configuration data is the annual median profile over Mauna Kea. The `Nscale` statement applies a scale factor `x` to

the tropospheric part of this profile, and our differentiation variable is this scale factor, not the total water vapor column density itself. That is, we're computing $\partial T_{R-J}/\partial x$ and not $\partial T_{R-J}/\partial N_{H_2O}$. The relationship between the two is

$$\frac{\partial T_{R-J}}{\partial N_{H_2O}} = \left(\frac{\partial N_{H_2O}}{\partial x} \right)^{-1} \frac{\partial T_{R-J}}{\partial x} = \frac{1}{N_0} \frac{\partial T_{R-J}}{\partial x} \quad (8.1)$$

where N_0 is the unscaled median tropospheric column density at $x=1$. As documented in the listing of Example 2.5, $N_0=1.86$ precipitable millimeters. (We could also have found this by examining the column density summaries in the `stderr` output after a run with $x=1$.)

The modified Example 2.5 takes the same command line arguments as before. Like N_0 , the scale factors needed to compute spectra corresponding to the 5th, 25th, 50th, 75th, and 95th percentile pwv values can be found at the top of the file listing in Section 2.2.5. A short `bash` script to loop through these scale factors and compute the corresponding spectra in the neighborhood of the 183.3 GHz water line is:

```
for TROP_NSCALE in 0.206 0.513 1.00 2.07 4.60; do
    am example2.5.trop_h2o_Jacobian.amc \
        165 GHz 200 GHz 1 MHz 0 deg 277 K $TROP_NSCALE \
        > trop_h2o_Jacobian_${TROP_NSCALE}.out \
        2> trop_h2o_Jacobian_${TROP_NSCALE}.err
done
```

The resulting output spectra for T_{R-J} and $\partial T_{R-J}/\partial N_{H_2O}$ are plotted in black in Figure 8.1, where the Jacobians have been scaled from the program output by $1/N_0$ following Eq 8.1. As expected, with increasing tropospheric pwv the line saturates in the center and the Jacobian drops towards zero. The narrow feature at line center is the narrow emission from stratospheric H₂O, which we're (realistically) holding constant. All the other narrow lines are due to stratospheric O₃.

In contrast with the monochromatic case, a practical radiometer will need spectral channels having sufficient bandwidth to provide the required sensitivity in a given integration time. For example, a Schottky mixer based radiometer might have a system temperature $T_{sys}=1500\text{ K}$, and the integration time might be limited to $t=1\text{ s}$ to remain below the wind crossing time for an antenna beam. In this case, a channel bandwidth $B=2\text{ GHz}$ would provide a brightness temperature measurement sensitivity $T_{sys}/\sqrt{(Bt)}$ of about 30 mK, or about 50 mK for the change between two successive measurements. Taking $\partial T_{R-J}/\partial N_{H_2O} \approx 30\text{ K/mm pwv}$ as a typical value of the derivative in the near line wing under median conditions, the corresponding sensitivity to a change in pwv is about $1.6\text{ }\mu\text{m}$, which in turn corresponds to a change in wet delay of about $10\text{ }\mu\text{m}$ (cf. Figure 6.3). This would be a reasonable sensitivity goal for a water vapor radiometer intended for delay correction at submillimeter wavelengths.

To compute the T_{R-J} spectrum and its Jacobian at a rectangular channel bandwidth of 2 GHz, we simply add the following line to the configuration file

```
ils rectangle 2 GHz
```

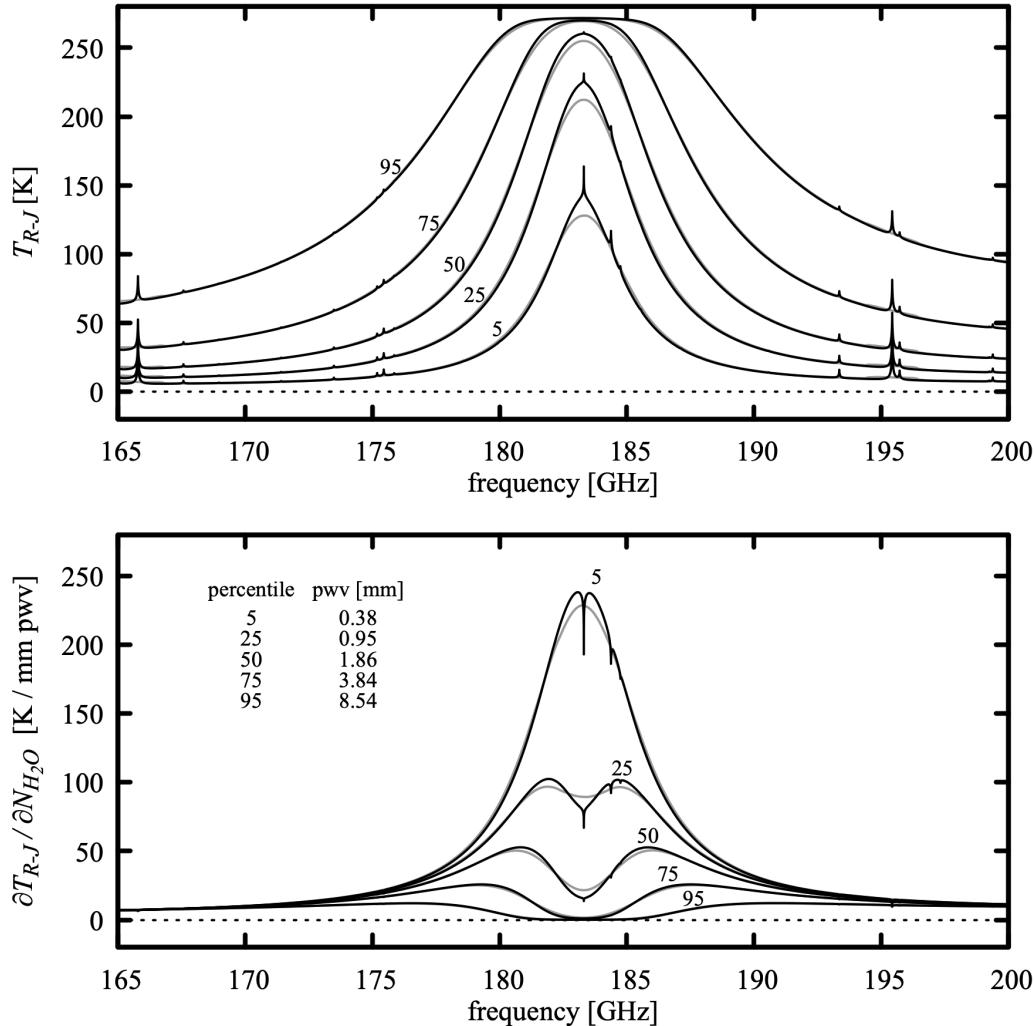


Figure 8.1 Rayleigh-Jeans brightness temperature (top) and its Jacobian with respect to variations in total tropospheric precipitable water vapor (bottom), plotted for various annual percentile values of pwv over Mauna Kea. The black curves are for a monochromatic model, and the gray curves are for a 2 GHz rectangular channel bandwidth.

and run the same cases as before. The resulting output spectra and Jacobians are plotted in gray in Figure 8.1; in this case the frequency axis corresponds to the center frequency for an arbitrarily placed 2 GHz wide channel. Note that if we wanted the Jacobian \mathbf{K} for a particular set of channel center frequencies, we would still need to run the model at higher resolution to obtain an accurate convolution of the monochromatic spectrum with the channel bandpass, then extract the desired rows of \mathbf{K} from the output file.

When *am* computes Jacobians, summary error estimates are written as comments to the

`stderr` output stream. For example, the file `trop_h2o_Jacobian_1.00.err` contains the following comment:

```
.
.
#
# Summary data for Jacobians
#
# Differentiation variable
#
# var           value          dvalue    units  description
# 0             1              1        none   Nscale troposphere h2o
#
#
# Relative error estimates for numerical derivatives (percentiles)
#
# array  var      0       5       25      50      75      95      100
#
#   Trj    0   4e-11   7e-11   1e-10   3e-10   5e-10   1e-09   4e-07
#
.
.
```

These error estimates are the sum of the estimated truncation and roundoff errors, expressed as a fractional error. Truncation error is estimated by comparing 3-point and 5-point derivatives, and roundoff errors are estimated based on machine accuracy; interested users can find the details in the source file `jacobian.c`. Errors are presented as percentiles to make it easy to recognize situations where the relative errors are large or meaningless in a small part of the spectrum. For example, in the neighborhood of a saturated spectral line the fractional error in the derivative may approach unity or zero as the derivative approaches zero. If the relative error percentiles hint at trouble, the detailed absolute error estimate spectra can be obtained as a program output using the `include_estimated_errors` keyword as described earlier.

8.2.2 An example from ozone line radiometry

Below the pressure altitude at which Doppler broadening becomes comparable to pressure broadening, the width of a spectral line is proportional to pressure. This makes it possible to retrieve information about the vertical concentration profile of a molecular species from the shape of the composite line observed from the ground. Ozone, which plays an important role in both atmospheric radiation and atmospheric chemistry, has been one such target for study. In particular, ground-based millimeter-wave radiometry was applied to interpretation of the Antarctic spring ozone hole phenomenon (Solomon 1988; Connor, et al. 1987), and at present a number of ozone profiling radiometers are in operation worldwide, including participants in the Network for the Detection of Atmospheric Composition Change (www.ndacc.org).

In the following example, we'll consider the $JK_aK_c=16\ 1\ 15 \rightarrow 16\ 0\ 16$ pure rotation line of O_3 at 231.28157 GHz, which has an optical depth at line center of order $\tau \approx 0.1$ and ground-state excitation temperature of 162 K. As pointed out by Conner, et al. (1987), if a line is optically thin with a ground-state excitation temperature close to typical atmospheric temperatures, then the brightness temperature profile of the line is only weakly dependent on

temperature. This means that the temperature profile need not be retrieved along with the concentration profile; a climatological temperature profile can be assumed. In this example we'll likewise assume a fixed temperature profile, and compute the Jacobian of the brightness temperature at a set of frequencies in the line wing, differentiated with respect to the O₃ mixing ratios on the model layers. In a practical retrieval, the linear model based on the Jacobian would be used to relate the measured spectrum and its estimated variance, together with a prior estimate of the mean and covariance of the mixing ratio profile, to a Bayesian estimate of the most likely profile and its associated covariance. Rodgers (2000) is the standard reference. Parrish et al. (1992) is an example of O₃ profile retrieval from ground-based O₃ line radiometry.

Once again the starting point is our standard example for Mauna Kea, Example 2.5. We modify the first few lines to read

```
.
.
f %1 %2 %3 %4 %5 %6
output f MHz Trj K
ifspec usb 231.28157 GHz
jacobian Trj
za %7 %8
tol 1e-4

Nscale troposphere h2o %11
.
.
```

The modified `output` statement changes the output spectrum to Rayleigh-Jeans brightness temperature, and changes the output frequency units to MHz. It will be convenient to consider channel frequencies as detunings relative to the line center, so we've added an `ifspec` statement to shift the zero of the frequency axis to the center frequency of the line; the `usb` keyword means we'll only get the positive frequency detunings (i.e. the upper sideband) in the output. As in the previous example the `jacobian` statement specifies that the Jacobian of the brightness temperature spectrum is to be computed. The next three statements are unchanged from the original Example 2.5. In particular, note that we've kept the `nscaler` statement for tropospheric H₂O. In practice, absorption from H₂O in the foreground of the O₃ line would need to be accounted for, for example through a calibration step such as a sky dip in the far wing of the O₃ line, or by making the tropospheric H₂O column density a retrieved parameter. Here we just treat it as a fixed model parameter.

The next step is to specify the differentiation variables, which are the O₃ mixing ratios on all 29 model layers. Each statement like

```
column o3 vmr 1.77e-06
```

is changed to

```
column o3 vmr 1.77e-06 1.77e-06
```

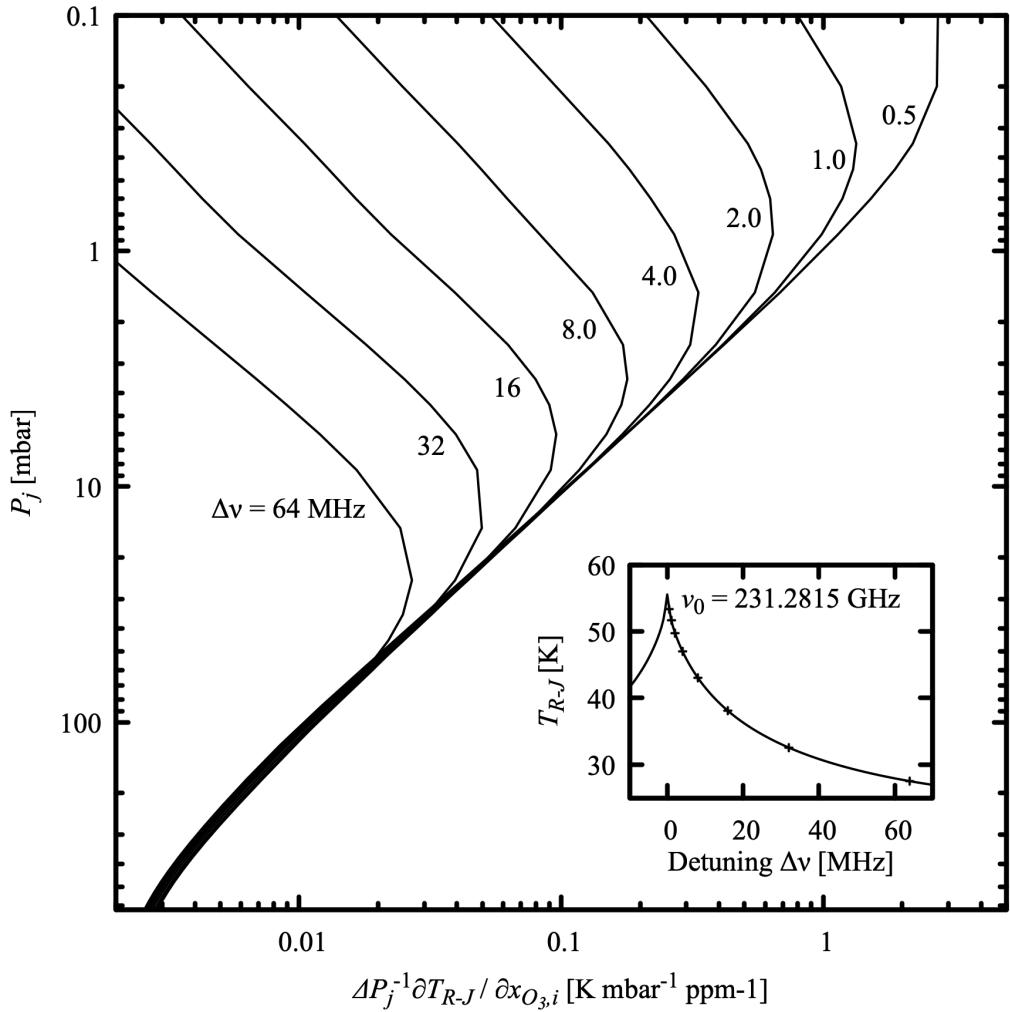


Figure 8.2 Selected rows of the Jacobian matrix of T_{R-J} differentiated with respect to ozone mixing ratios x_{O_3} , plotted as profiles against layer midpoint pressures P_j . Because the layer pressure drops ΔP_j are nonuniform, the elements of the Jacobian have been normalized to ΔP_j , making them proportional to derivatives with respect to O_3 column density.

to identify each layer's O_3 volume mixing ratio as a differentiation variable, keeping the reference state as the original annual median profile. As discussed above, here we simply set the curvature scale equal to the mixing ratio at the reference state.

With these modifications in place, we can run the following command to compute the Jacobian for a set of monochromatic channels at 500 KHz intervals in the high frequency wing of the line,

up to a maximum offset of 64 MHz:

```
$ am example2.5.o3_vmr_Jacobian.amc \
231.2815 GHz 231.346 GHz 500 kHz 0 deg 277 K 1.0 \
> K_o3_vmr.out 2> K_o3_vmr.err
```

The arguments are the same as for the original Example 2.5; here the last three arguments set the zenith angle to zero, the ground temperature to 277 K, and the scale factor on the median H₂O profile to unity. Note that although we've used the `ifspec` statement to shift the output frequency grid, we still have to set the computation frequency grid to cover the actual frequency range on the sky which will encompass our desired output frequency range, keeping in mind that the output frequency grid will be interpolated to detunings relative to the line frequency with the same 500 kHz spacing as the computation grid.

The contents of the output file, `K_o3_vmr.out`, are:

```
0.000000e+00 5.526953e+01 7.871439e+05 9.099245e+05 2.810412e+05 ...
5.000000e-01 5.332896e+01 2.737152e+05 5.426277e+05 2.195488e+05 ...
1.000000e+00 5.170448e+01 7.981190e+04 2.339590e+05 1.336338e+05 ...
.
.
.
6.300000e+01 2.762590e+01 2.439545e+01 8.769525e+01 7.198743e+01 ...
6.350000e+01 2.757509e+01 2.401801e+01 8.634436e+01 7.087964e+01 ...
6.400000e+01 2.752488e+01 2.365179e+01 8.502539e+01 6.979645e+01 ...
```

In all there are 31 columns in each row; the first two columns are the detuning $\Delta \nu_i$ and the Rayleigh-Jeans brightness temperature $T_{R-J}(\Delta \nu_i)$ at the reference state. The remaining 29 columns are the Jacobian matrix \mathbf{K} , starting in the third column of the output file with the derivatives $\partial T_{R-J}/\partial x_1$ with respect to the ozone mixing ratio x_1 on the top layer of the model, and working downwards through the layers as the columns proceed to the right.

Conceptually, the columns of \mathbf{K} are spectra of $\partial T_{R-J}(\Delta \nu_i)/\partial x_j$ versus detuning $\Delta \nu_1,..,\Delta \nu_m$ at constant layer pressure P_j . and the rows of \mathbf{K} are vertical profiles of $\partial T_{R-J}(\Delta \nu_i)/\partial x_j$ versus layer pressure $P_1,..,P_n$ at constant detuning $\Delta \nu_m$. Useful insight into the nature of the dependence of $T_{R-J}(\Delta \nu)$ on the O₃ profile can be gained from a plot of selected rows of \mathbf{K} as vertical profiles normalized to the layer pressure drops ΔP_j , shown in Fig. 8.2. Following a given profile at detuning $\Delta \nu_i$ from high altitude downwards, the derivative with respect to column density initially increases, reaching a sensitivity peak when the pressure-broadened line width $\gamma \approx \Delta \nu_i$. Continuing downwards, as γ increases linearly with pressure, sensitivity correspondingly decreases approximately linearly for frequencies which fall well within the line width. The slight bend towards higher sensitivity in the lowest levels is due to the increasing temperature with decreasing altitude.

References

J. I. H. Askne and E. R. Westwater 1986, "A review of Ground-Based Sensing of Temperature and Moisture by Passive Microwave Radiometers." IEEE Trans. Geosci. Remote Sensing 3:340.

- B. J. Connor, J. W. Barrett, A. Parrish, P. M. Solomon, R. L. de Zafra, and M. Jaramillo 1987, "Ozone Over McMurdo Station, Antarctica, Austral Spring 1986: Altitude Profiles for the Middle and Upper Stratosphere." *J. Geophys. Res.* 92:13221.
- N. E. Gaut 1967, "Studies of atmospheric water vapor by means of passive microwave techniques." Ph.D. Thesis, Massachusetts Institute of Technology Dept. of Meteorology. URI: <http://hdl.handle.net/1721.1/52946>
- B. Nikolic, R. C. Bolton, S. F. Graves, R. E. Hills, and J. S. Richer 2013, "Phase correction for ALMA with 183 GHz water vapour radiometers." *Astron. Astrophys.* 552:A104.
- A. Parrish, B. J. Connor, J. J. Tsou, I. S. McDermid, and W. P. Chu 1992, "Ground-Based Microwave Monitoring of Stratospheric Ozone." *J. Geophys. Res.* 97:2541.
- W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery 2007, *Numerical Recipes*, Third Edition. Cambridge University Press.
- C. D. Rodgers 2000, *Inverse Methods for Atmospheric Sounding*. World Scientific.
- S. Solomon 1988, "The Mystery of the Antarctic Ozone 'Hole'." *Rev. Geophys.* 26:131.
- M. C. Wiedner, R. E. Hills, J. E. Carlstrom, and O. P. Lay 2001, "Interferometric Phase Correction Using 183 GHz Water Vapor Monitors." *Ap. J.* 553:1036.

Chapter 9: Geometry and Refraction

Pressure is a natural vertical coordinate to use when measuring or modeling the state of the atmosphere. Weather stations and radiosondes typically measure local pressure along with other atmospheric state variables such as temperature and humidity. Remote sensing retrievals of vertical profiles use the fact that height is encoded in the pressure dependence of spectral lines. Hydrostatic mass column density is conserved between pressure levels, which simplifies dynamical models of the atmosphere when pressure is the vertical coordinate. The products of these measurements and models are routinely disseminated in pressure coordinates.

As we have seen, hydrostatic models in *am* are also defined in pressure coordinates. This makes it easy to construct configuration files from the data sources mentioned above, and offers an intuitive view of relative column densities and pressure-broadened line widths across the model. Often, as in all the examples preceding this chapter, assuming straight-line propagation through plane-parallel layers is an excellent approximation, and when this is the case we can work entirely in pressure coordinates.

However, for many applications we do need to bring geometry and refraction into the picture. For ground-based observations near the astronomical horizon, or limb observations from a satellite, Earth curvature and refraction are fundamental to the path geometry. For radio telescopes and other antennas with narrow beams, refraction has a significant effect on pointing even at relatively high elevation angles. Simply for convenience, we may want to interpolate an atmospheric model defined on a standard vertical pressure grid to the fixed altitude of an observatory or to the float altitude of a balloon.

In *am*, geometric vertical coordinates are derived automatically from the hydrostatic relation, starting from the user-defined base altitude of the lowest model layer. Refraction can be turned on using non-dispersive refractivity models appropriate for radio or optical wavelengths. Spherical geometry modes are available, and default geodetic parameters including planetary curvature radius, surface gravity, and vertical gravity gradient can be changed as needed to account for terrestrial latitude or to construct models for other planets. Interpolated levels for source, observer, and tangent height can be defined as needed in geometric or pressure

coordinates, independently of the model layer boundaries.

For some applications, such as correcting radio telescope pointing to account for refraction or finding the distance to the radio horizon, all we need are the atmospheric model and path geometry computations. For these tasks, *am* can be invoked with a command-line option to bypass the usual radiative transfer computations.

9.1 Spherical geometry and refraction in *am* models

An *am* configuration file defining a plane-parallel atmospheric model can be thought of as describing a radial profile in a spherical atmosphere. In many cases, modifying a plane-parallel model to account for spherical geometry and refraction is as simple as adding two configuration statements, such as

```
geometry spherical
refract radio
```

With no further changes, the base level of the lowest model layer will be assumed to correspond to the planetary surface, at altitude $z_0=0$, and default terrestrial values for the planetary surface curvature and gravity will be used. However, modifications to path geometry, geodetic variables, and refraction models are all possible. This section describes these and their associated configuration statements.

9.1.1 Path geometries and interpolated levels

Figure 9.1 depicts a propagation path through a spherically-symmetric atmosphere, greatly exaggerated in thickness relative to the size of its host planet. The layers are defined as in the plane-parallel case (Fig 1.2), but we have now added the planetary curvature radius R_0 and the altitude z_0 above the planetary surface at the base level of the lowest model layer. Above this level, the remaining pressure levels in the model are mapped onto altitude z using the hydrostatic relation Eq. 1.21. Because the real atmosphere is very thin relative to R_0 , treating it as locally spherically symmetric about the center of curvature C is an excellent approximation in the plane of the propagation path, even for an ellipsoidal planet like the Earth.

In addition to the layers delineated by solid lines in Fig 9.1, which define the atmospheric model, additional levels defining the pressure or geometric height of a source, observer, or tangent point on the propagation path are shown as dotted lines. When these levels are introduced, they are interpolated within the model definition layers without affecting the layer composition. This includes mixing ratios computed from relative humidity, which remain anchored to the model definition layers.

Any refracted path through the atmosphere lies along all or part of a ray trajectory such as $A-T-A'$ in Fig 9.1, symmetric about a tangent point T . Along this path, the refractive invariant

$$p = n(z) r \sin(\theta) \quad (9.1)$$

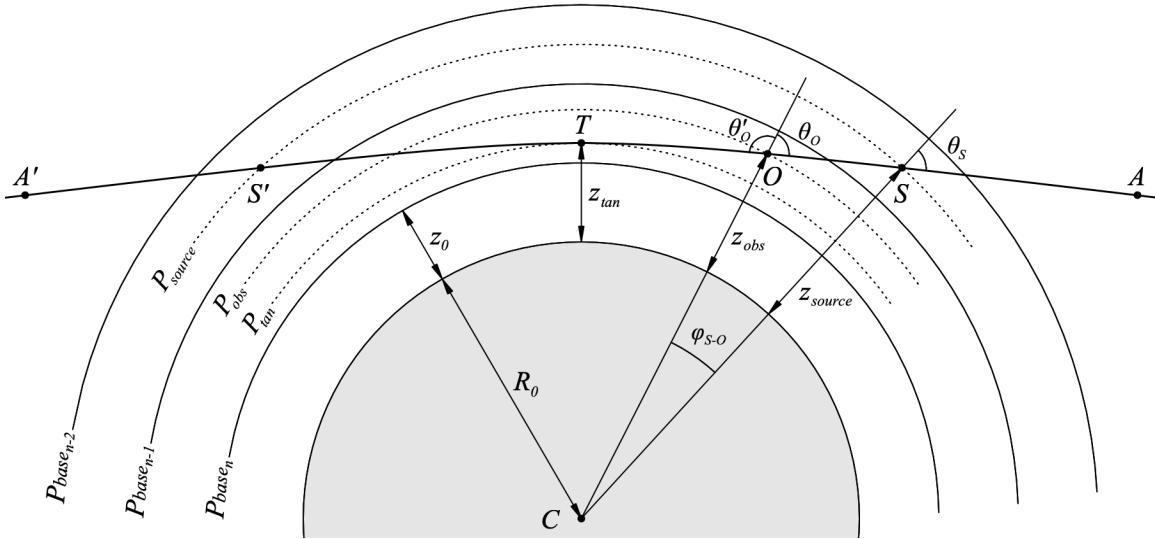


Figure 9.1 Propagation through a spherically-symmetric atmosphere.

is constant. Here, $n(z)$ is the refractive index at height z , $r=(R_0+z)$ is the radius from the planetary center, and θ is the local zenith angle. Paths that intersect the planetary surface such that the tangent point T would lie beneath it will nevertheless follow a trajectory of constant p in the atmosphere.

For typical vertical profiles of temperature and humidity, $\frac{dn}{dz} < 0$, so wavelength decreases towards lower altitude and the path bends towards the planetary center C . We define the total refractive bending ε between two points on the path to be positive in this case. For example, from S to O , the refractive bending is

$$\varepsilon_{S-O} = \varphi_{S-O} + (\theta_s - \theta_O). \quad (9.2)$$

Normally $\left|\frac{dn}{dz}\right|$ is small enough to ensure that the path curvature radius is larger than the planetary curvature radius everywhere, so that the path exits the atmosphere. However, exceptional variations in the temperature or humidity profile over a limited altitude range, such as can occur over the sea surface, can lead to the ray being trapped within the atmosphere in a *duct*. Users interested in further reading on atmospheric refraction and related phenomena are referred to the excellent review by Young (2006). The numerical ray tracing method implemented in *am* follows Kivalov (2007).

Returning to Figure 9.1, we may define three types of propagation path according to whether one, both, or none of the path endpoints lie within the atmosphere. For each type, the propagation path is fully defined by specifying the appropriate combination of geometry mode, observer zenith angle, and interpolated levels as described below:

One endpoint at infinity

The default propagation path in *am* corresponds to a typical astronomy or ground-based atmospheric radiometry scenario, in which radiation propagates from a point *A* outside the atmosphere, distant enough to be considered geometrically at infinity, to an observer at *O*, with the zenith angle θ_O at the observer specified by a `za` statement. Spherical geometry mode is selected by including the statement

```
geometry spherical
```

in the model configuration file.

By default, *O* is at the base of the lowest defined model layer. However, *O* can be placed at any height, either by specifying the observer pressure level with a `Pobs` statement such as

```
Pobs 630 mbar
```

or by specifying the observer's altitude with a `zobs` statement such as

```
zobs 4100 m
```

Either of these will cause *am* to insert an interpolated observing level, shown as a dotted arc in Figure 9.1. The layer containing this level will be split, and any layers not traversed by the propagation path will be ignored. For zenith angles below the horizon such as θ_O' towards *A'*, the path from *A'* to *O* passes through a tangent point *T* before reaching *O*. In this case *am* will solve for the tangent point height automatically and insert an additional interpolated level. The configuration data echoed to `stdout` by *am* at the end of a run is annotated with comments showing interpolated levels and split layers, layers traversed twice, and ignored layers. The roles of points *A* and *O* can be swapped with a `reverse` statement, placing the observer outside the atmosphere, but with the zenith angle still defined at the point *O*.

Two endpoints at finite distance

A path between two points at finite distance such as from a source at *S* to an observer at *O*, is defined by additionally specifying an interpolated source level with a `Psource` or `zsource` statement. The points *S* or *O* might be located outside the atmosphere, for example on a satellite orbit. If the source level lies above the observer, as in Figure 9.1, the path to the source is unambiguously determined by the observer zenith angle θ_O . In contrast, if the source level lies below the observer, then $\theta_O > 90^\circ$ and the line of sight intersects the source level twice, on opposite sides of the tangent point *T*. In this case, the default path in *am* is the path from the far side through the tangent point. The near side path can be chosen by adding the `near` keyword to the `Psource` or `zsource` statement, as in

```
zsource 15 km near
```

The zenith angle is always defined at the nominal observer level, but by using a `reverse`

statement to make the observer level function as the source level and vice-versa, a path can be specified by the zenith angle at the effective source level.

Both endpoints at infinity (limb path)

A limb path between two points A and A' geometrically at infinity, parametrized by tangent height, can be defined by setting the geometry mode with the statement

```
geometry limb
```

and specifying the tangent level pressure or altitude with a `Ptan` or `ztan` statement, respectively. This is the only geometry mode for which a user-specified tangent level is allowed.

A limb path with one or both endpoints at finite distance can also be specified in `spherical` mode as discussed below, parametrized by observer height and zenith angle.

9.1.2 Levels outside the atmosphere

Source, observing, and tangent levels can be defined outside the layers defining the atmospheric model, but `am` places limits on extrapolation. In addition, levels defined by a geometric height lying well outside the atmosphere, such as at the orbital height of a satellite, are treated specially as described below.

Formally, the top layer of a hydrostatic atmospheric model in `am` is always an isothermal slab bounded below by its defined base pressure and above by $P=0$, so that the height of any pressure level between these values can be located within this layer. Practically, isothermal extrapolation of the level height from the base of the top layer is unlikely to be physically realistic for pressures that are a small fraction of the layer base pressure. Consequently, `am` will log a warning if a source, observing, or tangent level defined by pressure is more than two pressure scale heights above the base pressure of the top model definition layer, but will nonetheless place the level at the corresponding isothermally extrapolated height. In contrast, if such a level is defined by height, and that height lies more than two pressure scale heights above the highest non-zero base level pressure, then the level will be regarded as outside the atmosphere and placed at $P=0$ at the specified geometric height. If source and observing levels are both placed outside the atmosphere in this way, such as to define a limb path between two satellites, the levels will be sorted into height order so that the propagation path is traversed in the correct order and the path distance is computed correctly.

For source, observing, or tangent levels placed below the lowest model definition layer, extrapolation is carried out assuming constant atmospheric composition and a user-defined temperature lapse rate $\Gamma = -dT/dz$ beneath this layer. The lapse rate is controlled by a `PTmode` statement in the model configuration file. The default behavior corresponds to

```
PTmode extend isothermal
```

which means that the position and temperature of the extrapolated level will be computed assuming that $\Gamma=0$. For models with temperatures T_{base} defined at layer boundaries, additional

options are

```
PTmode extend dry_adiabatic
PTmode extend moist_adiabatic
PTmode extend environmental
```

Taken in order, these correspond to assuming the lapse rate Γ is equal to the dry adiabatic lapse rate

$$\Gamma_{dry} = \frac{g}{c_p}, \quad (9.3)$$

the moist saturated pseudo-adiabatic lapse rate

$$\Gamma_{moist} = \frac{g}{c_p} \frac{1 + \frac{L\mu_s}{RT}}{1 + \frac{L^2\mu_s}{c_p R_v T^2}}, \quad (9.4)$$

or the local environmental lapse rate

$$\Gamma_{env} = -\frac{\Delta T_{base}}{\Delta z_{base}} \quad (9.5)$$

derived from the temperature and height differences between lowest two model definition levels for which the height difference $\Delta z_{base} > 0$. Eqs. 9.3 and 9.4 describe the change in temperature with height of an isolated parcel of dry air or saturated moist air, respectively (see, e.g., Andrews 2000). Here, g is the local gravitational acceleration and c_p is the dry air specific heat at constant pressure. For the latter, *am* assumes an Earth-appropriate atmosphere dominated by diatomic molecules in their vibrational ground state, for which

$$c_p = \frac{7}{2}R, \quad (9.6)$$

where R is the dry air specific gas constant defined by

$$R = \frac{k}{M_0} \quad (9.7)$$

with k the Boltzmann constant and M_0 the mean dry air molecular mass defined earlier in Eq. 1.10. Analogously, in the denominator of Eq. 9.4, R_v is the water vapor specific gas constant

$$R_v = \frac{k}{M_{H_{2o}}}. \quad (9.8)$$

Finally, L is the latent heat of vaporization of liquid water, which *am* approximates by its value at 0 C, and μ_s is the saturated water vapor mass mixing ratio at the temperature $T = T_{base}$ and pressure $P = P_{base}$ at the lowest model definition level.

Extrapolation of a new level below the lowest model definition level is limited to one pressure scale height whether the new level is defined by pressure or by geometric height. Attempting to extrapolate to higher pressure than this will generate an error. Similarly, if this

limit is reached when am is iteratively solving for a tangent level, an error will be logged and the program will exit. This generous high-pressure limit is merely protective; realistic applications should not normally extrapolate by more than a small fraction of the pressure at the lowest model definition level.

9.1.3 Geodetic parameters

The default values for the planetary surface curvature radius R_0 , surface gravity g , and vertical gravity gradient $\frac{dg}{dz}$ correspond to the configuration file statements

```
R0 6370 km
g 9.80665 m*s^-2
dg_dz 0 s^-2
```

Depending on latitude, use of the default values can contribute up to about 1% error to column density and path length computations in terrestrial models. For higher accuracy, it is necessary to account for the fact that the Earth is a rotating oblate ellipsoid, flattest at the poles. The curvature radius R_0 is a function of both latitude and azimuth angle, and the surface gravity g (including the centrifugal acceleration associated with Earth rotation) varies with latitude. The magnitude of the vertical gravity gradient at the surface, which is ignored by default, is such that the gravitational acceleration is reduced by approximately 0.3% at an altitude of 10 km. When geodetic parameters more accurate than the default values are needed, they can be derived as discussed below.

A standard approximation for the mean sea level equipotential surface of the Earth, or *geoid*, is the WGS 84 reference ellipsoid (U.S. National Imagery and Mapping Agency 2000). Many of the parameters defined in WGS 84, and reproduced below, are defined to a precision higher than their meaningful physical accuracy to facilitate consistency checking between software implementations. The WGS 84 reference ellipsoid is defined by the equatorial semi-major axis length

$$a=6378137.0 \text{ m} \quad (9.9)$$

and the flattening factor

$$f=1/298.257223563, \quad (9.10)$$

from which the polar semi-minor axis length is

$$b=a(1-f)=6356752.3 \text{ m}. \quad (9.11)$$

At a given latitude ϕ , the surface curvature radius in the meridional (North–South) plane is

$$R_{NS}=\frac{a^2 b^2}{(a^2 \cos^2 \phi + b^2 \sin^2 \phi)^{3/2}}, \quad (9.12)$$

and in a perpendicular East–West plane containing the local vertical, the curvature radius is

$$R_{EW} = \frac{a^2}{(a^2 \cos^2 \phi + b^2 \sin^2 \phi)^{1/2}}. \quad (9.13)$$

In a plane containing the local vertical and oriented at an arbitrary azimuth angle α measured with respect to North, the curvature radius is

$$R_\alpha = \left(\frac{\cos^2 \alpha}{R_{NS}} + \frac{\sin^2 \alpha}{R_{EW}} \right)^{-1}. \quad (9.14)$$

For reference, representative radii of curvature based on the WGS 84 reference ellipsoid are $R_p = 6399.594$ km at the poles, and $R_{NS} = 6335.439$ km, $R_{EW} = 6378.137$ km on the equator.

WGS 84 also defines a standard surface gravitational acceleration, including the centrifugal component. This varies with latitude from

$$g_a = 9.7803253359 \text{ m s}^{-2} \quad (9.15)$$

at the equator to

$$g_b = 9.8321849378 \text{ m s}^{-2} \quad (9.16)$$

at the poles. At arbitrary latitude ϕ , the variation follows the Somigliana formula (see, for example, Torge 1991) for the normal component of gravity on an equipotential ellipsoid:

$$g = \frac{a g_a \cos^2 \phi + b g_b \sin^2 \phi}{(a^2 \cos^2 \phi + b^2 \sin^2 \phi)^{1/2}}. \quad (9.17)$$

The vertical gradient of g is (Torge 1991)

$$\frac{dg}{dz} = -2 \frac{g}{a} (1 + f + m - 2f \sin^2 \phi). \quad (9.18)$$

The leading term is just the derivative of the gravitational acceleration with respect to distance from the center of mass at a distance a ; f is the flattening parameter introduced above, and

$$m = \frac{\omega^2 a^2 b}{GM}, \quad (9.19)$$

where WGS 84 defines the angular velocity ω of the Earth as

$$\omega = 7292115.0 \cdot 10^{-11} \text{ radian s}^{-1} \quad (9.20)$$

and the product of the gravitational constant G and the Earth's mass M as

$$GM = 3986004.418 \cdot 10^8 \text{ m}^3 \text{s}^{-2}. \quad (9.21)$$

The parameter m is approximately the ratio of the centrifugal to gravitational acceleration at the equator. Numerically, $m \approx 0.0034$, comparable in size to f . Because the latitude dependence of $\frac{dg}{dz}$ is small, it is normally sufficient to set

$$\frac{dg}{dz} = -3.085 \cdot 10^{-6} \text{ s}^{-2} \quad (9.22)$$

independently of latitude when it needs to be accounted for.

Putting this all together, we can derive some examples of configuration statements for accurate site-specific geodetic parameters. For 20 degrees North or South latitude and at an intermediate azimuth angle of 45 degrees from the meridian, approximating the SMA and ALMA observatory sites, suitable parameters are

```
R0 6362 km
g 9.786 m*s^-2
dg_dz -3.085e-6 s^-2
```

For the South Pole, where there is no azimuth dependence,

```
R0 6400 km
g 9.832 m*s^-2
dg_dz -3.085e-6 s^-2
```

would be an appropriate choice.

9.1.4 Refraction models

By default refraction is turned off, and must be enabled with a `refract` statement. For the purpose of path geometry computations, the refractive index is approximated as independent of frequency. There are two choices for this approximation:

```
refract radio
```

selects the low-frequency limit, corresponding to the Smith-Weintraub equation, Eq. 6.20, with coefficients Eq. 6.21. Alternatively, the statement

```
refract optical
```

selects a high-frequency (optical) limit. This corresponds to changing two coefficients in Eq. 6.20: setting $k_3=0$, which turns off the contribution to the wet refractivity from the H₂O pure rotation band; and reducing k_1 by 0.08%, which turns off a portion of the dry refractivity contributed mainly by the O₂ spin-rotation band. The optical limit is appropriate for computing refractive path bending at infrared through optical wavelengths, including for the auxiliary optical telescopes often used to help derive pointing models for radio telescopes.

9.2 Examples

The following examples illustrate the use of spherical geometry, refraction, and interpolated levels in `am`. In the first, we show how a simple atmospheric model based on the US standard atmosphere can be configured as a point-to-point radio propagation utility, and apply it to potential interference scenarios. In the second example, we use this same configuration file to compute path geometry only, showing how radio horizon distance versus transmitter height changes with relative humidity. In a third example, we compute refractive bending and airmass versus zenith angle from the SMA site on Mauna Kea, demonstrating the effects of temperature and humidity on radio pointing, and illustrating how a plane-parallel approximation of a spherical

atmosphere breaks down towards the horizon. The fourth and last example looks at background loading on an idealized balloon-borne cosmic microwave background (CMB) detector as a function of altitude. Further details on the use and syntax of the configuration file statements introduced in these examples can be found in Appendix B.

9.2.1 A point-to-point propagation utility

For a free-space radio link between two single-mode antennas, operating in the same polarization at wavelength λ and spaced a distance d apart, the ratio of received to transmitted power is given by the *Friis transmission equation* (Friis, 1946), originally stated in terms of the effective areas or cross sections of the antennas as

$$\frac{P_r}{P_t} = \frac{A_r A_t}{d^2 \lambda^2}. \quad (9.23)$$

Here, P_t and P_r are the transmitted and received powers or power spectral densities, and A_t and A_r are the effective areas of the transmitting and receiving antennas. Eq. 9.23 is valid when d is large enough that each antenna is in the far field of the other, which is the case when $d \gg \frac{A}{\lambda}$.

When this condition is met, $\frac{P_r}{P_t} \ll 1$. This is the normal situation in practical radio links, for which the received power is typically orders of magnitude below the transmitted power.

As noted previously in Section 1.2.4, the area-solid angle product of a single-mode antenna is $A\Omega = \lambda^2$, and we may further define the *directivity gain* G of an antenna relative to a hypothetical isotropic antenna that radiates into 4π sr as $G = \frac{4\pi}{\Omega}$. Putting these into Eq. 9.23, we obtain

$$\frac{P_r}{P_t} = G_r G_t \left(\frac{\lambda}{4\pi d} \right)^2. \quad (9.24)$$

The factor $\left(\frac{\lambda}{4\pi d} \right)^2$ is called the *free-space loss* and accounts for the geometric spreading of the transmitted power following the inverse square law.

An accounting of the gains and losses in a radio link is called a *link budget*. Because radio link losses and power levels range over orders of magnitude, it is usual in engineering computations to express the various powers, gains, and losses in the link budget logarithmically in dB. Powers in dB are defined as

$$P_{dB} = 10 \log_{10} \left(\frac{P}{P_0} \right), \quad (9.25)$$

most often with either $P_0 = 1 \text{ mW}$ (dBm) or $P_0 = 1 \text{ W}$ (dBW). For the antenna gains,

$$G_{dB} = 10 \log_{10}(G). \quad (9.26)$$

We may sometimes write G_{dBi} to indicate that the directivity gain is referenced to that of an

isotropic radiator.

In a link budget, it is usual to treat losses as reciprocal factors greater than unity, so that their values in dB are positive. By this convention, the free space loss in dB is

$$\begin{aligned} L_{fs, dB} &= -10 \log_{10} \left[\left(\frac{\lambda}{4\pi d} \right)^2 \right] \\ &= 20 \log_{10} \left(\frac{4\pi d}{\lambda} \right) \end{aligned} \quad (9.27)$$

For propagation paths in the atmosphere, atmospheric attenuation contributes another loss term $L_{atm, dB}$ to the link budget. This is related to the spectral transmittance $t(\nu)$ or the optical depth $\tau(\nu)$ by

$$\begin{aligned} L_{atm, dB} &= -10 \log_{10}(t(\nu)) \\ &= 10 \log_{10}(e) \tau(\nu) \text{ neper} \end{aligned} \quad (9.28)$$

Here we have shown explicitly how *am* converts from `neper` to `dB`, noting that *am* follows the radiometric convention of `neper` as the natural log of a power ratio, rather than the amplitude convention used in communications engineering. Putting it all together, a complete radio link budget including antenna gains, free space loss, and atmospheric attenuation is

$$P_{r, dB} = P_{t, dB} + G_{r, dB} + G_{t, dB} - L_{fs, dB} - L_{atm, dB}, \quad (9.29)$$

In practical applications, the antenna gains are frequency-dependent technical characteristics of the antennas. These may be main beam gains in the case of a radio link, or sidelobe gains for one or both antennas in an interference scenario. The loss terms $L_{fs, dB}$ and $L_{atm, dB}$ depend on frequency, propagation path geometry, and the state of the atmosphere.

The example that follows is an *am* configuration file that serves as a utility for computing $L_{atm, dB}(\nu)$, $L_{fs, dB}(\nu)$, and path geometry for a general point-to-point path in a spherical atmosphere. The atmosphere will be a simple model based on the US standard atmosphere, with constant humidity in the troposphere supplied as a command-line parameter, and with ozone omitted. The path is defined by command-line parameters defining source (transmitter) and observer (receiver) heights, zenith angle at the receiver, and the keyword `far` or `near` defining whether the source is on the far side or near side of the path tangent point, if any, relative to the observer. In addition we'll make the choice of refraction model, `radio` or `optical`, a command-line parameter so that we can use this same configuration file for optical refraction computations.

The output of the model will be frequency ν , spectral atmospheric attenuation loss $L_{atm, dB}(\nu)$, and spectral free-space loss $L_{fs, dB}(\nu)$ over a fixed frequency grid from 0 GHz to 150 GHz, covering a range of millimeter-wave applications. The `output` statement

```
output f GHz tau dB tau_fsl dB
```

gives us three columns of output: frequency in GHz, line-of-sight optical depth in dB along the refracted path, and free space loss in dB computed using the refracted path distance. Note that for consistency with `tau`, the default units for `tau_fsl` are `neper`, following the radiometric convention as noted above. In most applications involving free-space loss the desired output

units will be dB, which much be set explicitly as in this example. The complete configuration file, called `point2point.amc`, is listed here:

```

# File point2point.amc
#
# A simple (no ozone, no stratospheric H2O) config file for use
# as a point-to-point propagation and refraction utility. The
# atmosphere follows the US standard atmosphere temperature
# profile, with constant relative humidity in the troposphere as
# specified by the user. The propagation path is defined by
# source and observer height, observer zenith angle, the keyword
# 'far' or 'near' controlling source position relative to any
# path tangent point, and the keyword 'radio' or 'optical'
# controlling the refractivity model.
?
? Usage:
?   am point2point.amc zsource zobs  zenith_angle RH%  far|near radio|
optical
? Example:
?   am point2point.amc 10 km  2 km  80 deg  50%  far  radio
?

f 0 GHz  150 GHz  2.5 MHz
output f GHz  tau dB  tau_fsl dB

geometry spherical
refract %9
za %5 %6

zsource %1 %2 %8 # source altitude and side of tangent point
zobs    %3 %4      # observer altitude

layer
Pbase   1.109 mbar
Tbase  270.65 K
column dry_air hydrostatic

layer
Pbase   8.680 mbar
Tbase 228.65 K
column dry_air hydrostatic

layer
Pbase  54.75 mbar
Tbase 216.65 K
column dry_air hydrostatic

layer
Pbase 226.32 mbar
Tbase 216.65 K
column dry_air hydrostatic
column h2o RHi %7

layer
Pbase 358.58 mbar
Tbase 236.48 K
column dry_air hydrostatic
column h2o RHi %7

layer
Pbase 488.20 mbar
Tbase 250.78 K
column dry_air hydrostatic

```

```

column h2o RH %7

layer
Pbase 620.42 mbar
Tbase 262.48 K
column dry_air hydrostatic
column h2o RH %7

layer
Pbase 751.52 mbar
Tbase 272.23 K
column dry_air hydrostatic
column h2o RH %7

layer
Pbase 882.51 mbar
Tbase 280.68 K
column dry_air hydrostatic
column h2o RH %7

layer
Pbase 1013.25 mbar
Tbase 288.15 K
column dry_air hydrostatic
column h2o RH %7

```

As an example of using this configuration file, we'll consider potential interference to a satellite sensor in a hypothetical but typical 700 km orbit looking downward at a nadir angle of 45 degrees. We'd like to assess the degree of interference protection offered by atmospheric attenuation and free-space loss as a function of frequency for two scenarios involving transmitters situated in the sensor beam, in one case on the ground and in another on a commercial aircraft.

For the ground-based transmitter, we'll assume the transmitter is at sea level and the tropospheric humidity is 50%. The command to run this computation is

```
$ am point2point.amc 0 km 700 km 135 deg 50% near radio \
> p2p_0km_700km_135deg_rh50.out \
2> p2p_0km_700km_135deg_rh50.err
```

Note that *am* parametrizes the path in terms of observer zenith angle, and that a nadir angle of 45 degrees corresponds to a zenith angle of 135 degrees. We used the *near* keyword to choose the correct path between transmitter and satellite altitudes, avoiding the tangent point which in this case would be below the Earth's surface. The refraction model is set to *radio*.

For the airborne transmitter, we'll assume a cruise altitude of 13 km. At this altitude, the wet attenuation would be relatively negligible so we'll focus on the dry attenuation, setting the relative humidity to 0%. The corresponding command is

```
$ am point2point.amc 13 km 700 km 135 deg 0% near radio \
> p2p_13km_700km_135deg_dry.out \
2> p2p_13km_700km_135deg_dry.err
```

The *stderr* output from this second command to *p2p_13km_700km_135deg_dry.err* is lengthy, but worth examining to see how *am* handles layer splitting and levels outside the atmosphere:

```
# am version 14.0 (build date Sep 26 2024 13:20:00)
```

```

# run time 0.156 s
# dcache hit: 26 miss: 6 discard: 0

f 0 GHz 150 GHz 2.5 MHz
output text f GHz tau dB tau_fsl dB

tol 0
selfbroad_vmr_tol 0.003

geometry spherical
refract radio
za 135 deg
PTmode Pbase Tbase

g 980.665 cm*s^-2
dg_dz 0 s^-2

z0 0 m
R0 6370 km

T0 0 K

zobs 700 km
zsource 13 km near

| layer
| # za_base 45 deg
| Pbase 0 mbar # z = 700000.000 m (observing level)

#
# Layers above this line were not in the propagation path.
#

| layer
| # airmass 1.49483
| # za_base 51.1743 deg
| # P 1.109 mbar
| # T 270.65 K
| # dP 1.109 mbar
| Pbase 1.109 mbar # z = 47043.699 m
| Tbase 270.65 K
| Mair 28.964 h2o o3 (28.964)
| lineshape Gross ch4 co co2 n2o o2_uncoupled
| lineshape VVW_coupled o2_coupled
| column dry_air hydrostatic (1) (2.35127e+22 cm^-2)
! Warning: Column included 2332 unresolved lines.

layer | interpolated
Pbase 1.109 mbar
Tbase 270.65 K
Mair 28.964 h2o o3
lineshape Gross ch4 co co2 n2o o2_uncoupled
lineshape VVW_coupled o2_coupled
column dry_air hydrostatic

layer
# airmass 1.59792
# za_base 51.3417 deg
# P 4.8945 mbar
# T 240.345 K
# dP 7.571 mbar
Pbase 8.68 mbar # z = 32007.450 m
Tbase 228.65 K
Mair 28.964 h2o o3 (28.964)

```

```

lineshape Gross ch4 co co2 n2o o2_uncoupled
lineshape VWW_coupled o2_coupled
column dry_air hydrostatic (1) (1.60518e+23 cm^-2)

layer
# airmass 1.60317
# za_base 51.4752 deg
# P 31.715 mbar
# T 220.207 K
# dP 46.07 mbar
Pbase 54.75 mbar # z = 20003.904 m
Tbase 216.65 K
Mair 28.964 h2o o3 (28.964)
lineshape Gross ch4 co co2 n2o o2_uncoupled
lineshape VWW_coupled o2_coupled
column dry_air hydrostatic (1) (9.76765e+23 cm^-2)

| layer
| # airmass 1.60686
| # za_base 51.5514 deg
| # P 109.977 mbar
| # T 216.65 K
| # dP 110.453 mbar
| Pbase 165.203 mbar # z = 13000.000 m (source level)
| Tbase 216.65 K
| Mair 28.964 h2o o3 (28.964)
| lineshape Gross ch4 co co2 n2o o2_uncoupled
| lineshape VWW_coupled o2_coupled
| lineshape VVH_750 h2o_lines
| column dry_air hydrostatic (1) (2.34181e+24 cm^-2)
| column h2o vmr 0 (0 cm^-2)

#
# Layers below this line were not in the propagation path.
#

| layer
| # dP 61.1165 mbar
| Pbase 226.32 mbar # z = 11003.775 m
| Tbase 216.65 K
| Mair 28.964 h2o o3 (28.964)
| lineshape Gross ch4 co co2 n2o o2_uncoupled
| lineshape VWW_coupled o2_coupled
| lineshape VVH_750 h2o_lines
| column dry_air hydrostatic (1)
| column h2o vmr 0

layer | interpolated
Pbase 226.32 mbar
Tbase 216.65 K
Mair 28.964 h2o o3
lineshape Gross ch4 co co2 n2o o2_uncoupled
lineshape VWW_coupled o2_coupled
lineshape VVH_750 h2o_lines
column dry_air hydrostatic
column h2o RHi 0% (vmr 0)

layer
# dP 132.26 mbar
Pbase 358.58 mbar # z = 7951.694 m
Tbase 236.48 K
Mair 28.964 h2o o3 (28.964)
lineshape Gross ch4 co co2 n2o o2_uncoupled
lineshape VWW_coupled o2_coupled

```

```

lineshape VVH_750 h2o_lines
column dry_air hydrostatic (1)
column h2o RH 0% (vmr 0)

layer
# dP 129.62 mbar
Pbase 488.2 mbar # z = 5751.079 m
Tbase 250.78 K
Mair 28.964 h2o o3 (28.964)
lineshape Gross ch4 co co2 n2o o2_uncoupled
lineshape VVW_coupled o2_coupled
lineshape VVH_750 h2o_lines
column dry_air hydrostatic (1)
column h2o RH 0% (vmr 0)

layer
# dP 132.22 mbar
Pbase 620.42 mbar # z = 3950.639 m
Tbase 262.48 K
Mair 28.964 h2o o3 (28.964)
lineshape Gross ch4 co co2 n2o o2_uncoupled
lineshape VVW_coupled o2_coupled
lineshape VVH_750 h2o_lines
column dry_air hydrostatic (1)
column h2o RH 0% (vmr 0)

layer
# dP 131.1 mbar
Pbase 751.52 mbar # z = 2450.375 m
Tbase 272.23 K
Mair 28.964 h2o o3 (28.964)
lineshape Gross ch4 co co2 n2o o2_uncoupled
lineshape VVW_coupled o2_coupled
lineshape VVH_750 h2o_lines
column dry_air hydrostatic (1)
column h2o RH 0% (vmr 0)

layer
# dP 130.99 mbar
Pbase 882.51 mbar # z = 1150.144 m
Tbase 280.68 K
Mair 28.964 h2o o3 (28.964)
lineshape Gross ch4 co co2 n2o o2_uncoupled
lineshape VVW_coupled o2_coupled
lineshape VVH_750 h2o_lines
column dry_air hydrostatic (1)
column h2o RH 0% (vmr 0)

layer
# dP 130.74 mbar
Pbase 1013.25 mbar # z = 0.000 m
Tbase 288.15 K
Mair 28.964 h2o o3 (28.964)
lineshape Gross ch4 co co2 n2o o2_uncoupled
lineshape VVW_coupled o2_coupled
lineshape VVH_750 h2o_lines
column dry_air hydrostatic (1)
column h2o RH 0% (vmr 0)

#
# column densities [cm^-2], including scale factors
#
# airmass is line-of-sight / (zenith between observing and source level)
#

```

```

#                                         zenith      line-of-sight      airmass
# total (12 layers)
# (4 zenith, 0 down, 4 up):
#           dry_air 3.502603e+24      5.620507e+24      1.60467
#           h2o    0.000000e+00      0.000000e+00
#           (0 um_pwv)          (0 um_pwv)
#
#           all   3.502603e+24      5.620507e+24      1.60467
#
# source to observer distance
# refracted : 1030.59 km
# geometric : 1030.59 km
# projected : 728.839 km
#
# total refraction : 15.2461 arcsec
#
! Warning: Encountered in-band lines narrower than the frequency
!           grid spacing. The output configuration data includes
!           the unresolved line count after each column definition
!           for which this occurred. Count: 4

```

When the model is run, *am* first integrates the hydrostatic equation from the bottom layer upwards to find the height z at the base of each model definition layer. These altitudes are shown as comments added to each layer's `Pbase` statement. The user-specified source and observing levels, specified by height in this example, can then be located relative to the original model levels, and the layers that they fall within are split and interpolated as required to introduce these new model levels. Each original model definition layer that is split in this way at run time is labeled with a comment

```
layer | interpolated
```

and the layer data above each interpolated layer show how the layer was subdivided into new layers internally. In the output to `stderr`, the original model definition data are left intact and the new layer data are introduced by the '`|`' comment character, so that re-running the `stderr` output as a configuration file will exactly reproduce the original model run. Here, the source level is inserted by splitting the original parent model definition layer bounded below by a level at $P_{base}=226.32$ mbar into two new layers to accommodate a new level at $z=13$ km and $P_{base}=165.203$ mbar. The temperature of this new level is interpolated linearly in $\log(P)$ when layer temperatures are defined, as here, by base temperature T_{base} . (In the case of isothermal model definition layers, the new level simply picks up the temperature of the parent layer.) Column densities are split hydrostatically maintaining the volume mixing ratios of the parent layer, and water vapor mixing ratios that were defined by relative humidity on the parent layer are not recomputed at the new layer midpoint temperatures. In this way the mixing ratio profile defined by the original model layer definitions remains unaltered.

For the observing level at the satellite orbital altitude, a new layer has been inserted at the top of the layer stack following the rules described above in Section 9.1.2. Because the observing level is more than two scale heights above the base level of the highest model definition layer in the atmosphere, this new layer has been placed outside the atmosphere at $P=0$. This layer is bounded below at the observing level and above at infinity, so the propagation path stops at the

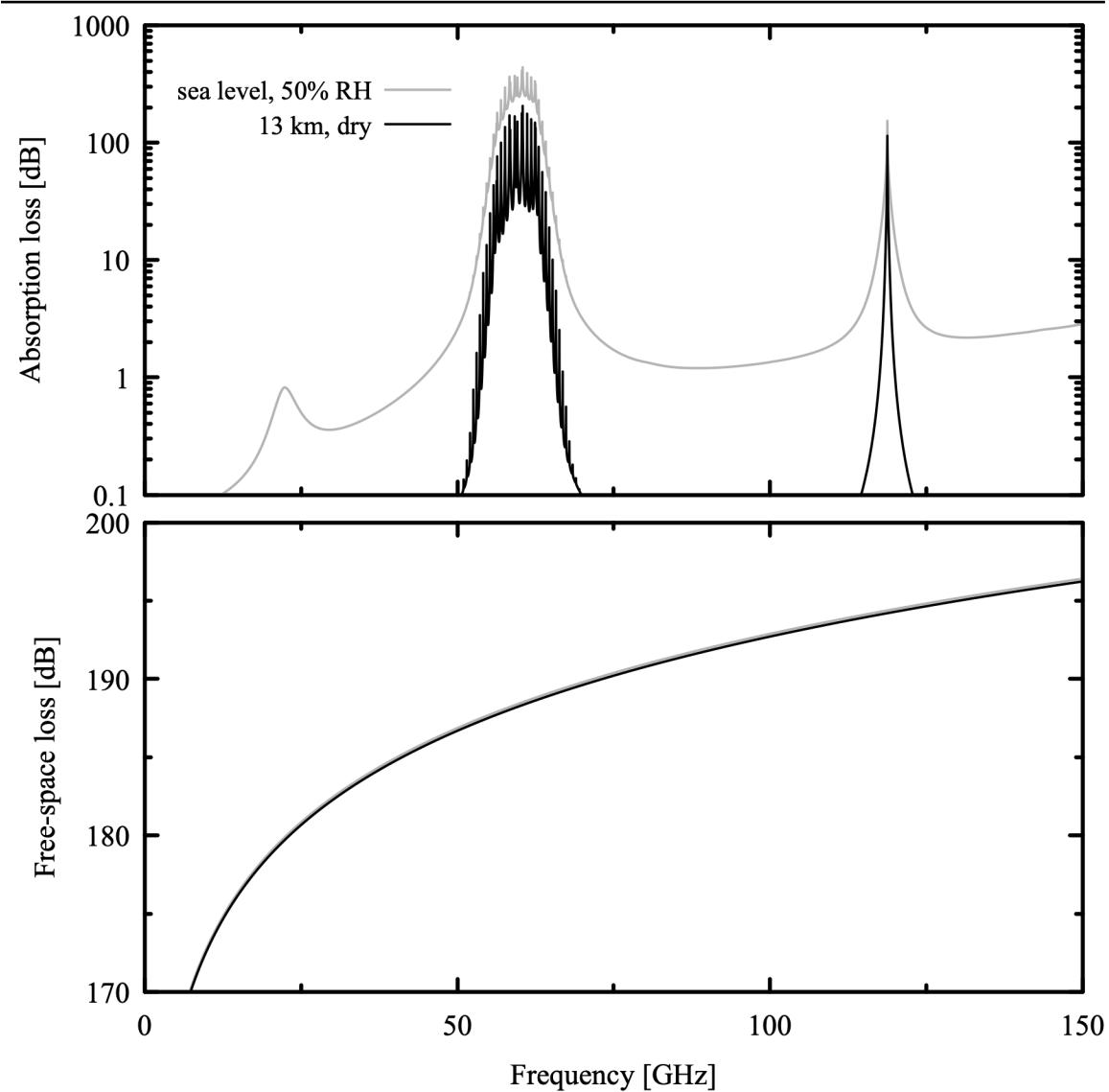


Figure 9.2 Atmospheric absorption loss (top) and free-space loss (bottom) for paths from a transmitter at sea level (gray) and an airborne transmitter at 13 km altitude (black) to a satellite sensor at 700 km altitude observing towards a nadir angle of 45 degrees.

base of this layer and does not traverse it, as indicated in the output annotation. Note that the model definition layer at the top of the atmosphere still has $P=P_{base}$, following the usual rule for the highest layer in the original model definition as discussed in Section 1.1.1.

Turning our attention to the end of the `stderr` output, we see that for spherical models the usual column density summary table is augmented with an added column giving the airmass factor, defined as the ratio of line-of-sight column density to zenith column density. The airmass

factor here reflects the fact that the zenith angles across the atmospheric layers, indicated by added comments on each layer, are somewhat larger than the 45 degree zenith angle at the satellite altitude owing to Earth curvature. The column density table is followed by comments giving several computed parameters related to the path geometry. These are, in order, the distance from source to observer measured along the refracted path; the geometric straight-line distance from source to observer; the projected track or map distance along the planetary surface, and finally the total refractive bending ε along the path as defined in Eq. 9.2. At the very end of the `stderr` output we see a warning triggered by the fact that the frequency grid as defined does not fully resolve a large number of spectral lines in the top layer of the model atmosphere. Nevertheless, it is useful to have this layer in the model in case we later edit the configuration file to zoom in on part of the spectrum at finer resolution.

The computed atmospheric attenuation and free-space loss spectra for both the sea level and airborne transmitters are shown in Figure 9.2. In both cases, atmospheric attenuation affords little shielding from interference to a satellite-borne radiometer except near the center of the O₂ spin-rotation band near 60 GHz (and the associated line at 118.75 GHz), a band used for multi-channel temperature sounding. Even towards the center of the 60 GHz band, the atmospheric shielding in the airborne transmitter case is not especially high near the attenuation minima between O₂ lines, and an interference analysis would need to consider this in detail. For the sea-level transmitter, at frequencies outside the O₂ band, the moist troposphere does not provide significant shielding even at the moderate humidity level considered here. The absorption would be even less in dry weather, in which an operational sensor supporting weather forecasting, for instance, would still be depended upon to provide uncorrupted data.

The free-space loss shown in the bottom panel of Figure 9.2 is approximately the same for both cases, simply because the relative difference in path distance is small. Although this loss looks impressively high across the millimeter-wave band, it has to be viewed in the context of typical antenna gains for millimeter-wave remote-sensing radiometers ranging from 30 dBi to 62 dBi (International Telecommunication Union 2021), and acceptable interference levels ranging from -189 dBW to -169 dBW received power (International Telecommunication Union 2012). Putting these ranges into a link budget like Eq. 9.9, one finds that even a single transmitter in the sensor footprint having microwatt to milliwatt-level equivalent isotropic radiated power can pose an interference threat, underscoring the importance of the passive remote sensing bands that are protected in international law by primary allocations in the ITU Radio Regulations. Some of these protected bands are associated with particular molecular emission lines or bands that enable measurement of atmospheric temperature and composition. Others are suitably located with respect to atmospheric windows and spectral signatures associated with surface properties such as soil moisture, ocean salinity, or ocean surface roughness, from which wind can be deduced.

9.2.2 Distance to the radio horizon

Invoking *am* with the `-a` command-line option will cause the program to process the atmospheric model and compute the path geometry, but bypass any spectral computations. Instead of writing spectra to `stdout`, *am* writes a single line containing path geometry data. In

in this example, we'll use the same configuration file we used in the previous example to compute the distance to the radio horizon as a function of transmitter height and humidity, with the temperature profile once again following the US Standard Atmosphere. We'll compare this to the distance computed without accounting for refraction, and use this to justify a frequently-used approximation for the horizon distance.

The radio horizon will be defined as the distance from the transmitter at which the refracted line-of-sight from the observer just reaches the transmitter over the curvature of the Earth, assuming a smooth planetary surface and neglecting diffraction. At the horizon, the observer zenith angle is 90 degrees. So, for example, for a transmitter height of 100 m, we can find the horizon distance at a relative humidity of 50% (and at the surface layer temperature of the US standard atmosphere) using the configuration file `point2point.amc` from our previous example with the command:

```
$ am -a point2point.amc 100 m 0 m 90 deg 50% near radio
```

With the `-a` option, the output to `stdout` is a single line with six data columns:

```
90      409.353   317.75   40.9353  40.9353  40.9350
```

These are, in order, the zenith angle at the observer; the airmass factor; the refractive bending ε from source to observer in arc seconds; the source to observer path distance; the straight-line geometric distance; and the projected track distance along the planetary surface. The three distances are only meaningful in the context of point-to-point paths and do not appear otherwise. The units for these distances are the same as the units for the planetary radius R_0 , which default to kilometers unless different units have been used in an `ro` statement in the configuration file. For point-to-point paths such as this one, the airmass factor is defined as the ratio of the path integrated column density to the vertical column density between source and observer levels. Consequently, here it is simply the ratio of the path distance to the transmitter height. Note also that for this gently curved and nearly horizontal path, the path distance, geometrical distance, and projected track distances are all practically the same.

We can compare the refracted path distance to the distance computed neglecting refraction. (This is not the same as the straight-line geometric distance reported by `am`, which is that between the source and observer positions as determined from the refracted path.) Referring to Figure 9.1, in the absence of refraction the path segment OS is a straight line segment and OCS is a proper triangle. From the law of cosines, the length of OS is

$$|OS| = \sqrt{R_{src}^2 - R_{obs}^2 \sin^2 \theta_O} - R_{obs} \cos \theta_O, \quad (9.30)$$

where $R_{src} = R_0 + z_{src}$ and $R_{obs} = R_0 + z_{obs}$. At the horizon, $z_{obs} = 0$, the zenith angle $\theta_O = 90 \text{ deg}$, and Eq. 9.30 reduces to the Pythagorean equation:

$$|OS| = \sqrt{(R_0 + z_{src})^2 - R_0^2}. \quad (9.31)$$

For $z_{src} \ll R_0$, this simplifies to

$$|OS| \approx \sqrt{2 R_0 z_{src}}. \quad (9.32)$$

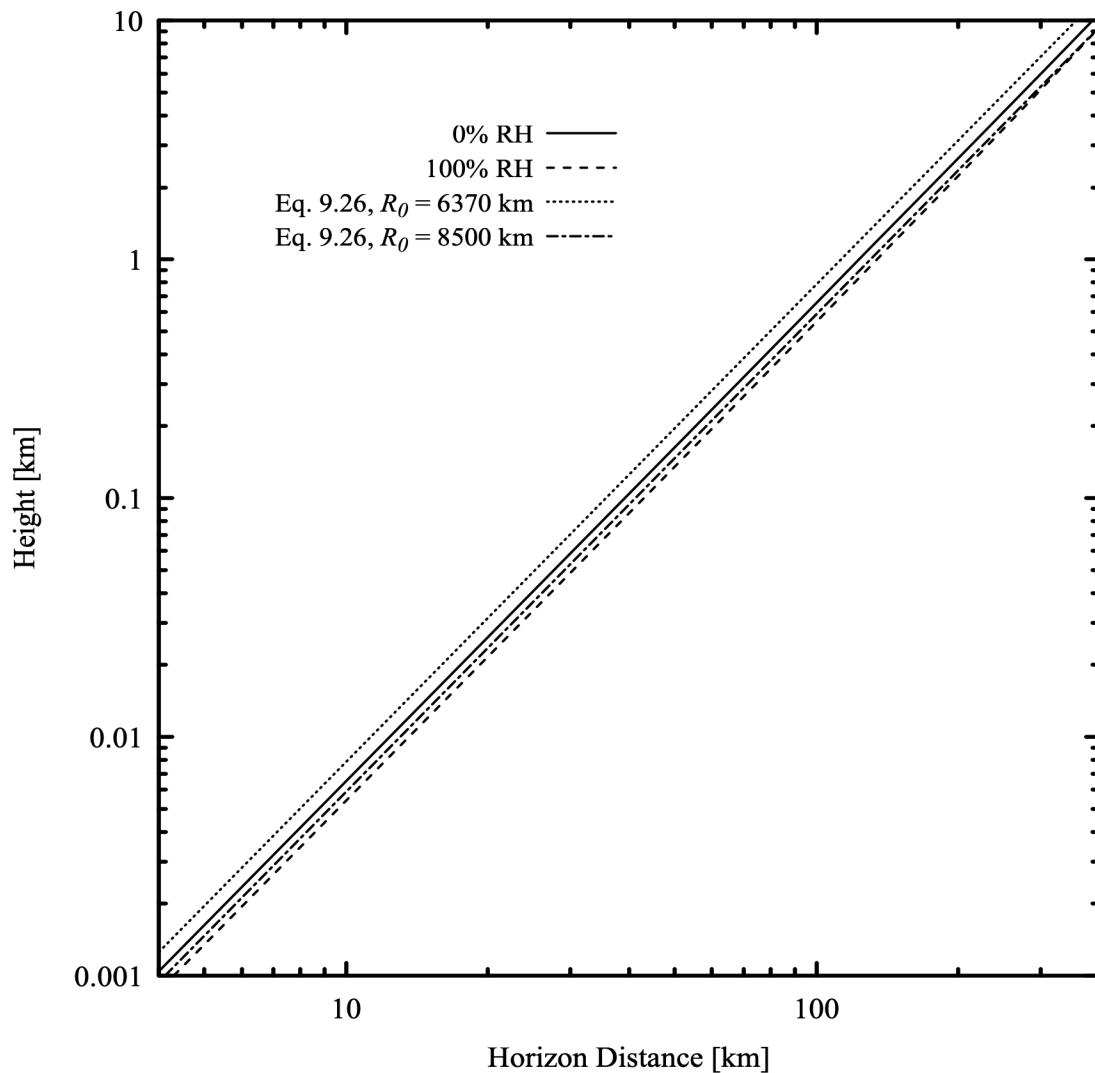


Figure 9.3 Horizon distance versus transmitter height

For our comparison, we can compute the refracted horizon distance for a range of logarithmically-spaced transmitter heights from 1 m to 10 km, at 0% and 100% relative humidity, with a short shell script:

```
for HEIGHT in $(awk '
BEGIN {
    f = exp(log(10.) / 100.)
    for (h = 0.001; h <= 10; h *= f)
        printf("%e\n", h)
}
'); do
printf "%s " $HEIGHT >> radio_horiz_dry.out
'
```

```

am -a point2point.amc $HEIGHT km 0 km 90 deg    0% near radio \
>> radio_horiz_dry.out

printf "%s " $HEIGHT >> radio_horiz_wet.out
am -a point2point.amc $HEIGHT km 0 km 90 deg 100% near radio \
>> radio_horiz_wet.out

done

```

The results are plotted in Figure 9.3. Adding moisture to the atmosphere at constant humidity makes the refractivity more sensitive to both pressure and temperature. The consequent increase in the vertical refractivity gradient results in greater refractive bending and a correspondingly increased horizon distance. Also shown in Figure 9.3 is Eq. 9.32 evaluated for two values of R_0 . Note that by using an effective radius R_0 larger than the actual Earth radius, it is possible to compensate for the true path curvature and improve the approximation significantly. This effective radius approximation is commonly used for computing path distances in terrestrial microwave links (Ippolito 1986).

9.2.3 Refraction and airmass vs. zenith angle at the SMA

The third example is a modification of Example 2.5 to compute refraction and airmass versus zenith angle from the SMA site on Mauna Kea. Refraction causes an astronomical source to appear higher in elevation than its true geometric position, and must be accounted for in telescope pointing. We'll examine the effects of temperature and humidity, and also compare spherical and plane-parallel computations to assess when the latter is a sufficient approximation.

We begin by adding the following lines to the original Example 2.5:

```

geometry spherical
refract radio
z0 4080 m

```

Other than these added lines, the contents of the file remain the same. To compute airmass and refraction only, we run *am* with the *-a* command-line switch as in the previous example. For example, assuming the modified file is named `example2.5.spherical.amc`, the command to run the computation for a zenith angle of 70 degrees, surface temperature of 4.0 C, and unit scaling of the median water vapor profile is

```
am -a example2.5.spherical.amc 0 GHz 0 GHz 0 GHz 70 deg 4.0 C 1.0
```

Here, we needed to include dummy arguments for the frequency grid range because the original file had placeholders for them. However, when a configuration file is run using the *-a* switch, the file does not need have a statement defining a frequency grid.

When we run this command, the output to `stderr` includes the augmented column density summary described previously, with the airmass factor tabulated individually for each layer group and column type as well as for the entire path:

```

:
:
```

```

#
# column densities [cm^-2], including scale factors
#
# airmass is line-of-sight / (zenith above observing level)
#
#               zenith      line-of-sight      airmass
# mesosphere (5 layers):
#       dry_air 1.484112e+22    3.729608e+22    2.51302
#       h2o     9.880028e+16    2.490325e+17    2.52056
#           (0.029557 um_pwv) (0.0745004 um_pwv)
#       o3      2.696866e+16    6.797331e+16    2.52046
#           (1.00375 DU)      (2.5299 DU)
#
# stratosphere (12 layers):
#       dry_air 1.469267e+24    4.200076e+24    2.85862
#       h2o     6.695272e+18    1.912806e+19    2.85695
#           (2.00295 um_pwv) (5.72234 um_pwv)
#       o3      6.161173e+18    1.755622e+19    2.84949
#           (229.313 DU)      (653.425 DU)
#
# troposphere (12 layers):
#       dry_air 1.176312e+25    3.419954e+25    2.90735
#       h2o     6.196403e+21    1.808005e+22    2.91783
#           (1853.71 um_pwv) (5408.82 um_pwv)
#       o3      9.342006e+17    2.706675e+18    2.89732
#           (34.77 DU)        (100.74 DU)
#
# total (29 layers):
#       dry_air 1.324722e+25    3.843692e+25    2.90151
#       h2o     6.203197e+21    1.809943e+22    2.91776
#           (1855.74 um_pwv) (5414.61 um_pwv)
#       o3      7.122342e+18    2.033086e+19    2.85452
#           (265.086 DU)      (756.694 DU)
#
#           all   1.325343e+25    3.845504e+25    2.90152
#
# total refraction : 101.364 arcsec
#

```

In addition to the airmass factors, the `stderr` output also includes a comment giving the overall refractive bending ε from the start to the end of the path.

The output to `stdout` consists of a single line as in the previous example, but here since the source is at infinity there are only three columns, giving the zenith angle at the observer in degrees, the airmass factor for the entire path, and the refractive bending in arc seconds:

```
70      2.90152    101.36
```

By iterating over zenith angle for various values of surface temperature and tropospheric water vapor scale factor, we can explore the effect of these parameters on radio pointing corrections as a function of zenith angle. The results are shown in Figure 9.4. The upper panel shows the refractive bending ε versus zenith angle for median conditions, both for the spherical model and for a plane-parallel model with refraction turned on. For comparison, the SMA antennas have beam widths of order several tens of arc seconds so the required elevation correction ε is significant over most of the sky. The lower two panels show the differences $\Delta \varepsilon$ relative to the median spherical case for variations in humidity and surface temperature. The changes are significant, especially at low elevation, but small enough that an approximate accounting is

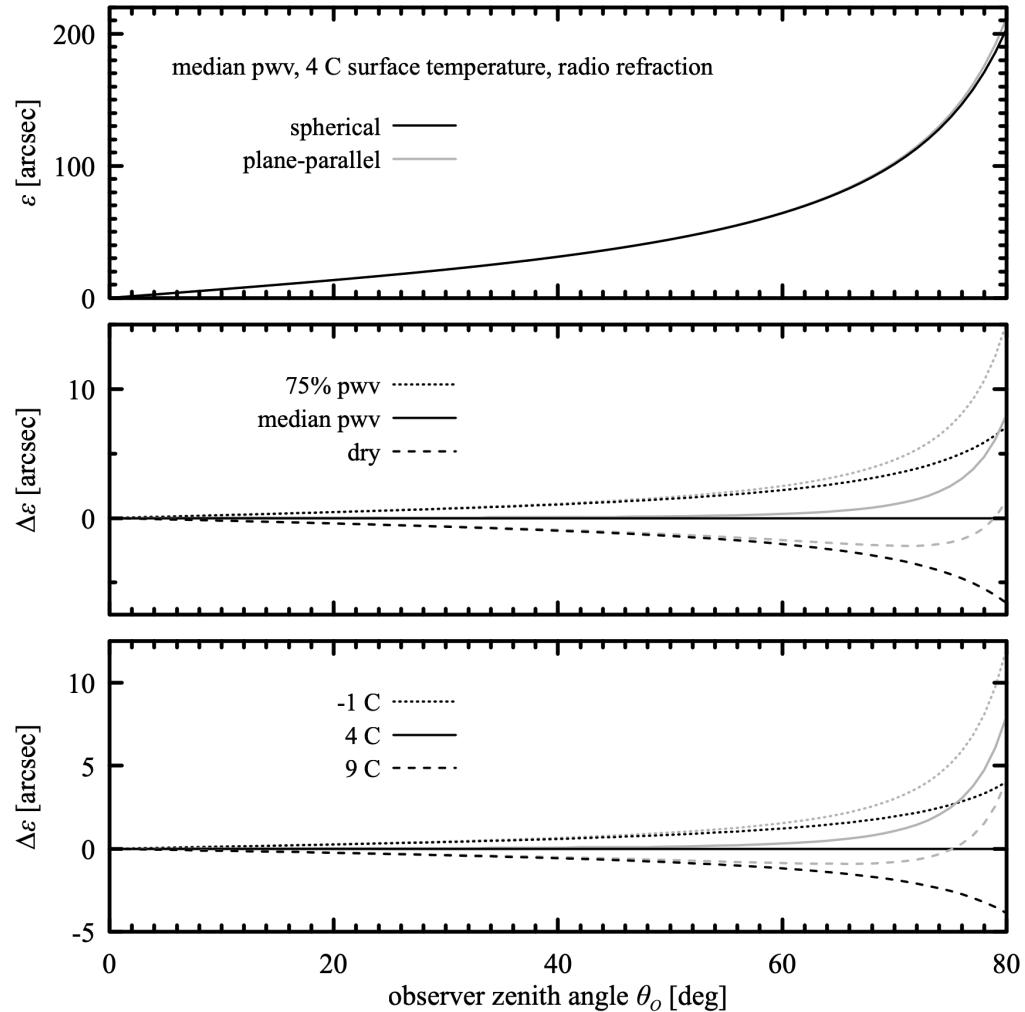


Figure 9.4 Total refractive bending ϵ versus zenith angle from Mauna Kea under median conditions (top), and difference $\Delta\epsilon$ from the median for changes in humidity (center) and temperature (bottom).

sufficient at the arc second level of accuracy. Note that at low elevation, refraction is significantly overestimated by a plane-parallel model.

At low elevations, the curvature of the atmosphere also causes the zenith angle to decrease significantly with altitude along the line of sight. As a result, the variation with θ_o of the airmass of a given atmospheric constituent depends on its vertical profile, especially as the line of sight approaches the horizon. Water vapor, concentrated in the troposphere, has a significantly larger airmass factor near the horizon than ozone, concentrated in the stratosphere. We can examine this quantitatively by parsing the column density tables written to `stdout` to extract airmass versus

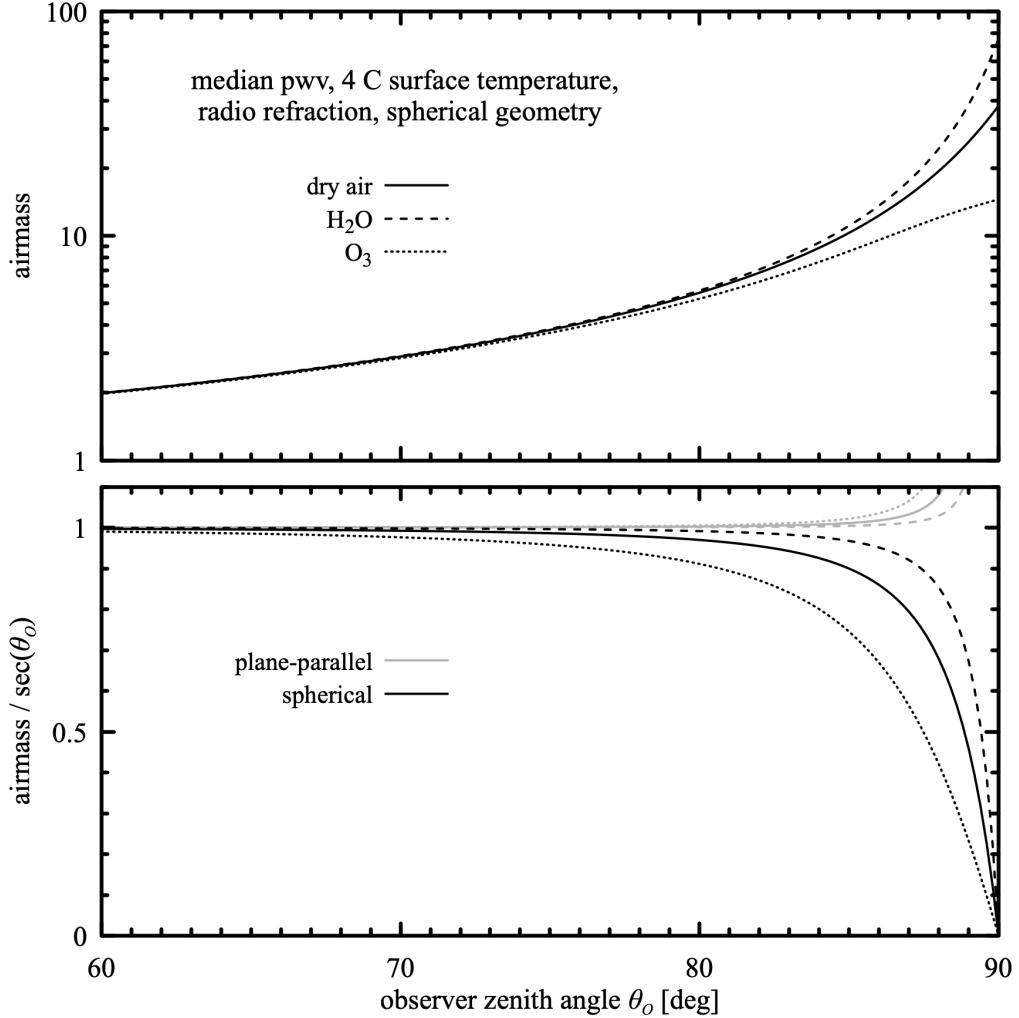
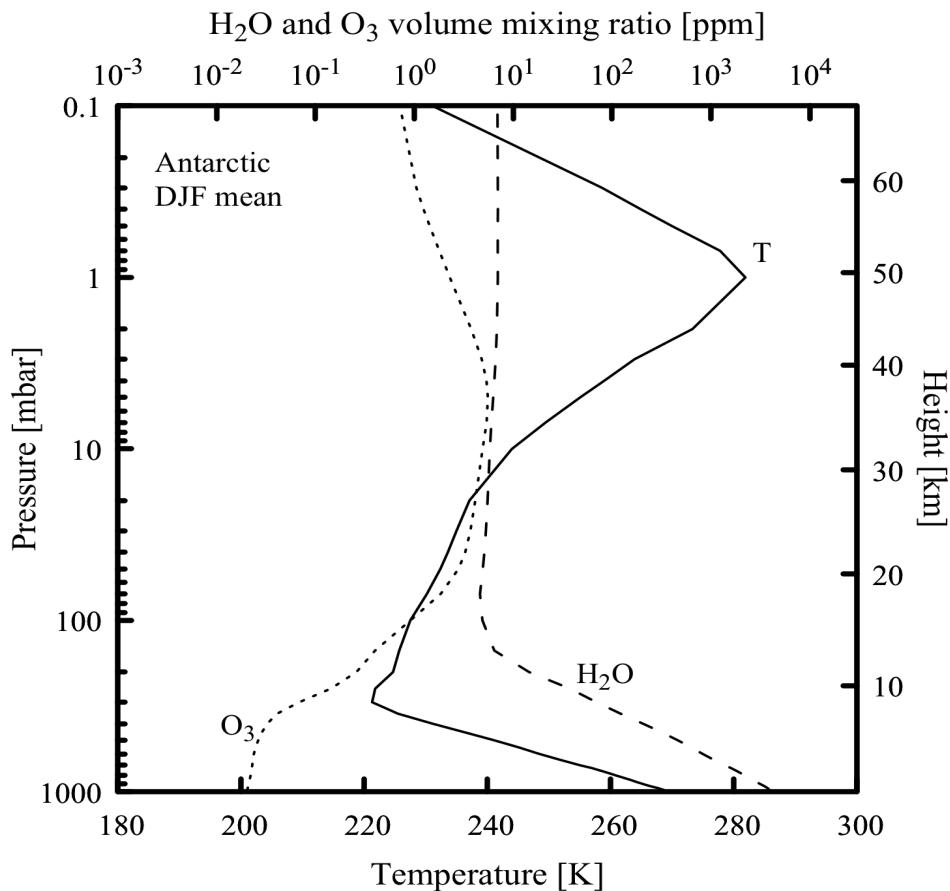


Figure 9.5 Top: airmass relative to zenith for dry air, water vapor, and ozone from the SMA site on Mauna Kea, computed for a spherical atmosphere. Bottom: ratio of the computed airmass to the secant approximation for a spherical atmosphere and for a plane-parallel atmosphere with refraction turned on.

observer zenith angle by species. The results for median conditions on Mauna Kea are shown for dry air, water vapor, and ozone in the top panel of Figure 9.5. In the lower panel the ratio of the airmass computed for spherical and plane-parallel models with refraction is compared with the secant approximation, which corresponds to a plane-parallel atmosphere with no refraction. Note that for ozone, the secant approximation is in error by nearly 10 percent at a zenith angle of 80 degrees, and the ratio drops to zero as $\sec(\theta_o)$ diverges at $\theta_o=90^\circ$. Turning on refraction in the plane-parallel approximation makes the approximation worse since the divergence is encountered



Compiled from NASA MERRA-2 reanalysis data. MERRA is produced by the NASA/Goddard Global Modeling and Assimilation Office (GMAO). MERRA data are archived and distributed by the Goddard Earth Sciences Data and Information Services Center (GES DISC).

Figure 9.6 Vertical profiles of temperature, water vapor, and ozone for summer (DJF) Antarctic atmosphere.

earlier, when $\theta_z=90^\circ$ at the top of the atmosphere.

9.2.4 Background loading on a balloon-borne detector

The last example in this chapter considers the background loading on a single-mode detector carried by a high-altitude balloon. We'll suppose that this detector is one pixel of a cryogenic focal plane array of antenna-coupled bolometers fed by a larger wide-field telescope in an idealized cosmic microwave background (CMB) experiment, and that we want to know the radiative loading on each pixel as a function of the balloon float altitude and telescope zenith angle. We'll assume the focal plane array views the sky through a cold filter having a perfectly rectangular bandpass 40 GHz wide and centered at 150 GHz, and compute the combined radiative

loading on the detector from the atmosphere and from the CMB itself. Sources of radiation in the cryostat and telescope optics will be ignored.

We'll assume the balloon flight takes place in the Antarctic summer, and use the *am* cookbook file `Antarctic_DJF.amc` as a starting point. The vertical profiles contained within this file were compiled from NASA MERRA-2 data, zonally averaged over latitudes from 66 degrees to 90 degrees South, and time-averaged over the months of December, January, and February for the decade 2006-2016; the profiles are plotted in Fig. 9.6.

We'll make several modifications to the cookbook file. The output spectrum will be set to be the Rayleigh-Jeans brightness temperature `Trj`. Recall from Eq. 1.41 that $kT_{R-J}(\nu) = \frac{\lambda^2}{2} I(\nu)$ is the power per unit bandwidth accepted by an antenna coupled to a single-mode transmission line. For a rectangular bandpass B , if we compute the band-averaged $\langle T_{R-J} \rangle$ using a rectangle instrumental line shape, then the total power coupled to the detector is $k \langle T_{R-J} \rangle B$. For the spectral computation, we'll set up a frequency grid wide enough to cover the 40 GHz rectangular bandpass, with grid resolution (10 MHz) sufficient to integrate accurately across stratospheric ozone lines. However, the output will be limited to a single frequency point at the center frequency of the filter using an `fout` statement. A `za` statement for the observer zenith angle, and a `zobs` statement for the observer altitude will take command-line arguments to set these parameters. Finally, we'll use spherical geometry so that the computation will be correct down to the horizon. The contents of the modified file (omitting the details of all the layers) are:

```
# File CMB_det_load_Antarctic_DJF.amc
#
# A modified version of the file Antarctic_DJF.amc from the am
# cookbook, for computing single-mode detector loading versus
# observing altitude. Altitude and zenith angle are taken from the
# command line. Detector frequency and rectangular averaging bandwidth
# are adjusted by editing this file.
#
# Output is band-averaged Rayleigh-Jeans brightness temperature in K.
# Multiply this by Boltzmann's constant and the bandwidth to get the
# detector loading power.
#
# The profiles in this file were derived from NASA MERRA2 reanalysis
# data, temporally and spatially averaged using the Giovanni online
# data system developed and maintained by the NASA Goddard Earth
# Sciences Data and Information Services Center (GES DISC).
#
# Files:
#   g4.curtainTime.M2IMNPASM_5_12_4_H.20061201-20161130.180W_90S_180E_66S.nc
#   g4.curtainTime.M2IMNPASM_5_12_4_O3.20061201-
20161130.180W_90S_180E_66S.nc
#   g4.curtainTime.M2IMNPASM_5_12_4_QV.20061201-
20161130.180W_90S_180E_66S.nc
#   g4.curtainTime.M2IMNPASM_5_12_4_T.20061201-20161130.180W_90S_180E_66S.nc
#
# MERRA references:
#   M. M. Rienecker et al. (2011), MERRA: NASA's Modern-Era Retrospective
#   Analysis for Research and Applications. J. Climate 24:3624.
#
#   A. Molod et al (2015), Development of the GEOS-5 atmospheric general
#   circulation model: evolution from MERRA to MERRA2. Geoscience Model
#   Development 8:1339.
```

```

#
# MERRA is produced by the NASA/Goddard Global Modeling and Assimilation
# Office (GMAO). MERRA data are archived and distributed by the
# GES DISC.
?
? Usage:
?   am CMB_det_load_Antarctic_DJF.amc  zobs zenith_angle
?
? Example:
?   am CMB_det_load_Antarctic_DJF.amc  35 km  60 deg
?

# The frequency grid should extend at least half the bandwidth to
# either side of the band center frequency. 10 MHz resolution will
# provide adequate accuracy for integrals across stratospheric O3
# lines. Warnings about unresolved high-altitude lines can be
# ignored.
f 130 GHz 170 GHz 10 MHz

# This line sets up a single-point output at the band center frequency
fout 150 GHz 150 GHz

# This is the detector bandwidth
ils rectangle 40 GHz

output Trj K

geometry spherical
refract radio

# Observing altitude from the command line
zobs %1 %2

# Zenith angle from the command line
za %3 %4

tol 1e-4

T0 2.7 K

layer mesosphere
Pbase 0.1 mbar # z = 66887 m
Tbase 231.1 K
lineshape Voigt-Kielkopf
column dry_air vmr
column h2o vmr 6.94e-06
column o3 vmr 7.15e-07
.
.
# other layers from Antarctic_DJF.amc
.

layer troposphere
Pbase 975 mbar # z = 104 m
Tbase 268.7 K
column dry_air vmr
column h2o vmr 3.83e-03
column o3 vmr 2.05e-08

```

We've included all the layers down to the surface, but when we run the file the layer containing the balloon float altitude will be split, and the layers below the balloon will be ignored.

Next, we set up a bash script to iterate over altitude and zenith angle. Here, we don't need

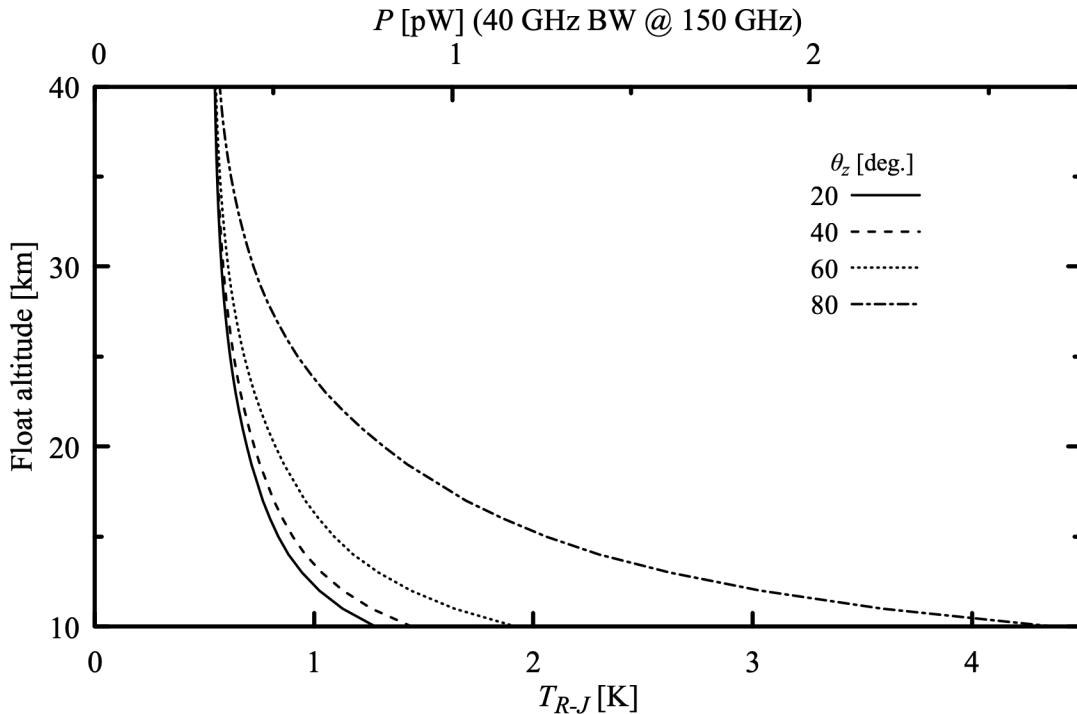


Figure 9.7 Background loading from atmospheric emission and the cosmic microwave background on a balloon-borne single-mode detector in the Antarctic summer stratosphere. Float altitude is on the vertical axis, and the curves correspond to different zenith pointing angles of the telescope.

the `stderr` output, so it is redirected to `/dev/null`. During development, we might instead capture `stderr` to a log file. The script is:

```
for Z in $(seq 10.0 1.0 40.5); do
    printf "%4.1f" $Z
    for ZA in $(seq 20.0 20.0 81.0); do
        TRJ=$(am CMB_det_load_Antarctic_DJF.amc $Z km $ZA deg 2>/dev/null)
        printf " %e" $TRJ
    done
    printf "\n"
done
```

Running this script will produce a table with lines for float altitudes in the stratosphere ranging from 10 km to 40 km in 1 km steps, and columns containing the float altitude followed by $\langle T_{R-J} \rangle$ at 20, 40, 60, and 80 degrees zenith angle. The results are plotted in Fig 9.7; the upper scale gives the power loading $P=k\langle T_{R-J} \rangle B$ in picowatts. Note that the background loading on the detector is asymptotically limited by the CMB itself as the float altitude approaches the upper stratosphere.

References

- D. G. Andrews 2000, *An Introduction to Atmospheric Physics*. Cambridge University Press.
- H. T. Friis, 1946, "A Note on a Simple Transmission Formula." Proceedings of the IRE 34:254-256.
- International Telecommunication Union 2021, "Typical technical and operational characteristics of Earth exploration-satellite service (passive) systems using allocations between 1.4 and 275 GHz." Recommendation ITU-R RS.1861-1.
- International Telecommunication Union 2012, "Performance and interference criteria for satellite passive remote sensing." Recommendation ITU-R RS.2017-0.
- L. J. Ippolito, Jr. 1986, *Radiowave Propagation in Satellite Communications*. Van Nostrand Reinhold Company.
- S. N. Kivalov 2007, "Improved ray tracing air mass numbers model." Applied Optics 46:7091-7098.
- W. Torge 1991, *Geodesy*, Second Edition. Walter de Gruyter & Co.
- A. T. Young 2006, "Understanding Astronomical Refraction." The Observatory 126:82-115.
- U.S. National Imagery and Mapping Agency 2000, "Department of Defense World Geodetic System 1984." NIMA TR8350.2. (Available from US National Geospatial-Intelligence Agency at <https://earth-info.nga.mil/GandG/publications/tr8350.2/wgs84fin.pdf>)

Appendix A: Installation and Performance

A.1 Installation

A.1.1 Installing on Windows

A precompiled version of *am* is available as a Windows installer file. The executable has been compiled with OpenMP support using Microsoft Visual Studio Community 2022 (version 17.11.4), and packaged with Caphyon Advanced Installer (version 22.0). The installer will install the required OpenMP and C runtime redistributable libraries from Microsoft if they are not already present on the system.

The installer will create a directory for the *am* cache files, and create a new user environment variable `AM_CACHE_PATH` pointing to the default cache directory, which will be

```
c:\Users\<username>\AppData\Local\am
```

The executable file, together with the manual, source code, and example files from the *am* cookbook, will be installed by default in the (64-bit) Program Files directory at

```
c:\Program Files\Smithsonian Institution\am-14.0
```

The installation directory path will be appended to the installing user's `PATH` environment variable.

Uninstalling will undo the above, with two exceptions. First, if the cache directory is not empty, it will not be removed. Second, if multiple users install the program on the same system, and it is subsequently uninstalled, the cache directory (if empty) will be removed and environment variables will be modified only for the user doing the uninstall. Other users would then need to remove these manually if desired.

To install:

1. Obtain the installer file, `am-14.0-x64.msi`. The latest version of the installer file, and links to earlier versions, can be found in the Zenodo repository at

<https://doi.org/10.5281/zenodo.1193770>

2. Unzip the archive file, then run the installer file by double-clicking it, or by running the command

```
msiexec /i <path-to-installer-file>
```

To remove:

1. Go to Control Panel->Uninstall a program, or Settings->Apps->Apps & Features.
2. Select "am-14.0" from the list of installed programs and choose "Uninstall".

A.1.2 Installing on GNU/Linux, Unix, and macOS

On these systems, *am* is installed by compiling from source and copying the resulting executable to the desired directory. The cache directory and user environment are set up manually. The procedure below assumes that the compiler is `gcc`, which is available on most GNU/Linux systems and many other Unix-like systems. The Nvidia (PGI) `nvc` and Intel `icc` compilers are also supported. More information on using these compilers can be found in the Makefile help message, which can be read using the command `make` or `make help` in the *am* source directory.

Compiling on macOS requires a few extra steps. The minimum is to download and install the Apple Xcode Developer Tools. To provide basic `gcc` compatibility, Xcode installs a `gcc` command in `/usr/bin`, but this is actually just a front-end to the `clang` compiler used by Xcode. The current version (Xcode 16.0) does not provide OpenMP support, so it can only compile the serial version of *am*. This works, but is not recommended.

Instead, the recommended way to build *am* on macOS is to install `gcc` using a package manager such as Brew, Fink, or MacPorts. These are an excellent way to install and maintain up-to-date `gcc` compilers and many other tools for technical computing. In particular, building *am* with MacPorts has been thoroughly tested. MacPorts requires Xcode as a prerequisite, and the MacPorts web site (www.macports.org) has helpful instructions for installing Xcode.

The following installation steps are for a typical GNU/Linux system and `gcc` compiler; the procedure on other Unix-like systems and macOS will be similar:

1. Check the version of `gcc` installed on the system with the command:

```
gcc --version
```

If the version is 4.2 or higher, then OpenMP is supported.

2. Obtain a copy of the archive file `am-14.0.tgz`. The latest version of the archive, and links to earlier versions, can be found in the Zenodo repository at

```
https://doi.org/10.5281/zenodo.1193770
```

3. In a suitable directory, unpack the archive with the command:

```
tar -xzf am-14.0.tgz
```

This creates a new directory `am-14.0/` containing the source files in the subdirectory `src/`, and example files from the *am* cookbook in the subdirectory `cookbook/`.

4. Change to the `am-14.0/src/` directory, and run

```
make am
```

to build the OpenMP version (recommended), or

```
make serial
```

to build a single-threaded version.

5. Copy the resulting executable file `am` to a suitable directory in the user's path. This could be, for example, a private bin directory `~/bin` for a single user, or `/usr/local/bin` for all users. Giving the command

```
make install
```

as root, or

```
sudo make install
```

will copy `am` to `/usr/local/bin`, and set the appropriate file attributes. (On macOS, `/usr/local/bin` is listed in `/etc/paths`, which is read by the `path_helper` utility that constructs the shell `PATH` environment variable. However, on a fresh macOS installation, this directory may not yet exist, so the `install` target will create it if needed.)

6. Create a directory for the *am* cache files, and set an environment variable so that *am* can find it. The cache directory can be private, or it can be shared by multiple users—concurrent instances of *am* can safely share the cache. For good performance, it is important that the cache directory reside on a locally-connected disk, rather than on a disk mounted across a network. A typical user-private setup would be to create a directory `.am` in the user's home directory, and add the line

```
export AM_CACHE_PATH=~/am
```

to the user's `.bashrc` (typical on GNU/Linux) or `.bash_profile` or `.zprofile` (typical on macOS, where every terminal session runs a login shell). C shell or tcsh users would add the

line

```
setenv AM_CACHE_PATH ~/am
```

to their `.cshrc` file.

A.2 Compile-time options

The most important factors for obtaining good performance are to set up the disk cache as described above, and to compile with OpenMP support on multicore or multi-CPU machines. There are a few macros, listed below, which can be set at compile time to override the default values of certain parameters. With `gcc` and many other compilers, this is done by passing an argument of the form `-Dmacro=value` to the compiler. The `am` Makefile help message shows how to accomplish this using the `EXTRA_CFLAGS` variable on the `make` command line. In most cases, overriding the default parameters won't be necessary.

L1_CACHE_BYTES

Size of the L1 data cache per core in bytes. The default value is 0x8000 (32 kB). This controls cache blocking of absorption coefficient computations, and sets the point at which FFT and FHT computations switch over from recursive to iterative. Setting this parameter larger than the actual L1 cache size will hurt performance, whereas setting it somewhat smaller won't matter much.

L1_CACHE_WAYS

Associativity of the L1 data cache. The default value is 8. Together with `L1_CACHE_BYTES`, this controls cache blocking of absorption coefficient computations.

The default L1 cache settings are appropriate for all modern Intel and AMD processors. These settings will also give within 5 percent of peak performance on Apple M1 despite the larger L1 cache on these processors, which is 128 kB for performance cores and 64 kB for efficiency cores. In contrast, on older AMD Opteron processors with the "Bulldozer" microarchitecture that have 16 kB 4-way L1 data caches, a significant performance improvement is obtained by compiling `am` with appropriate non-default L1 cache parameters. This is easily done by defining the `EXTRA_CFLAGS` macro on the `make` command line as follows:

```
$ make gcc-omp 'EXTRA_CFLAGS = -DL1_CACHE_BYTES=0x4000 -DL1_CACHE_WAYS=4'
```

Typing `make help`, or simply `make` with no arguments will give a few more examples. Note that spaces are only allowed around the first '=', as shown here.

L2_CACHE_BYTES

Size of the L2 data cache per core in bytes. The default is 0x100000 (1 MB). This sets the size of certain benchmarks run with `am -b`, and has no effect on performance.

LINESUM_MIN THD_BLOCKSIZE

This is the smallest number of frequency grid points which will be given to a thread doing a block of a line-by-line computation. The default value is 8, which is the number of 8-byte double-precision numbers which will fit into the 64-byte cache lines found on many machines. To avoid false sharing, it is recommended not to make this value any smaller. False sharing occurs when two or more threads, running on different CPUs, access distinct, unshared data that reside in the same cache line. Each time one thread writes data to this cache line, it invalidates all other copies of the same cache line, including the unshared data, held by other CPUs.

FFT_UNIT_STRIDE
FHT_UNIT_STRIDE

These select whether iterative FFT's and FHT's are done with unit-stride memory access (at the expense of extra trig computations), or with non-unit-stride memory access. These computations are done entirely in L1 cache, minimizing the cost of non-unit-stride access, so the default setting for both of these parameters is 0. Long ago, some machines (e.g. Sun Ultra 3) ran faster if these parameters were set to 1.

A.3 Environment Variables

The environment variables described below affect the run-time behavior of *am*. These include *am*'s own program-specific environment variables, and environment variables that control OpenMP program execution.

A.3.1 Program-specific environment variables

AM_CACHE_PATH

This is the path to the *am* cache directory. If this variable is not defined, or is set to an empty string, the disk cache is disabled. The disk cache is described in Section 2.3, Accelerating computations.

AM_CACHE_HASH_MODULUS

This can be used to modify the number of hash buckets in the disk cache. By default, there are 1021 hash buckets containing up to 4 files per bucket, meaning that the maximum number of files in the cache directory is limited to 4084. For best cache efficiency prime numbers are recommended, and care should be taken not to exceed the file system's practical limit on files per directory. If the hash modulus is changed, any existing cache files in the active cache directory will become unusable and will eventually be evicted from the cache.

AM_FIT_INPUT_PATH
AM_FIT_OUTPUT_PATH

By default, *am* reads fit data from the current directory, and writes fit output files to the

current directory. These variables are used to set alternative input and output directories.

AM_KCACHE_MEM_LIMIT

The kcache is an in-memory cache of absorption coefficient arrays, computed as needed on a fixed temperature grid and interpolated to intermediate temperatures. It is used to accelerate fits involving variable layer temperatures. By default, *am* will allocate memory for the kcache as needed, up to the maximum memory available to user processes on the system. Alternatively, **AM_KCACHE_MEM_LIMIT** may be used to set an upper limit (in bytes) on memory that will be allocated for the kcache. In either case, once kcache memory has reached the maximum allocation limit, older cache entries will be discarded to free memory for newer ones.

A.3.2 OpenMP environment variables

A complete list of the environment variables that control the run-time behavior of a particular OpenMP version is given in that version's specification document, which may be found at www.openmp.org. The subset of OpenMP environment variables listed below are those that an *am* user is most likely to need to set or change.

OMP_NUM_THREADS

This sets the number of threads which will be used for parallel regions of the program. If **OMP_NUM_THREADS** is not defined, an implementation-dependent default value is used. This is typically the number of logical processors seen by the operating system. As discussed below and demonstrated in the following section, this may not be the optimal setting. The command "`am -e`" will give information on the number of logical processors seen by OpenMP and the current setting of **OMP_NUM_THREADS**.

On systems with processors that feature simultaneous multithreading (SMT, called *hyper-threading* on Intel processors) there is an important distinction between logical processors and physical processor cores. In a multithreaded processor core, a relatively small set of thread-specific resources such as registers, register renaming tables, and program counters are duplicated. These duplicate resources enable the core to appear to the operating system as multiple logical processors, with these logical processors sharing the functional units of the core. The object is to achieve higher utilization of the core's functional units by having multiple threads ready to have their instructions dispatched to functional units as they become free. However, if one thread on its own is able to saturate a shared resource, hardware multithreading can gain no speed advantage and may even slow down execution by introducing extra contention for this resource. For this reason, when the processor supports the option SMT is often disabled in high performance computing environments, as in the examples below. On a personal machine with an SMT processor it may be optimal to set **OMP_NUM_THREADS** to a number no greater than the number of actual processor cores.

A recent trend is heterogeneous processors with a mix of cores optimized for performance or efficiency. An example is the Apple M1 Pro discussed below, with eight performance cores and two efficiency cores. On machines with this processor, the `gcc` OpenMP runtime defaults to

`OMP_NUM_THREADS=10`, but using this default setting can result in erratic performance caused by fast cores waiting on slow cores to finish their work. In contrast, setting `OMP_NUM_THREADS=8` on these machines results in consistent good performance as discussed in the next section.

Finally, it is important to note that in shared computing environments, `OMP_NUM_THREADS` is likely to default to the total number of CPUs on the node where *am* is running, regardless of the number of CPUs requested when submitting a job. When this is the case, it is essential to set `OMP_NUM_THREADS` equal to the requested number of CPUs to avoid overloading the node and interfering with other jobs.

OMP_NESTED *deprecated since OpenMP 5.0 (November 2018)*

OMP_MAX_ACTIVE_LEVELS *introduced in OpenMP 3.0 (May 2008)*

These environment variables control OpenMP nested parallelism. In early versions of OpenMP, nested parallelism was simply enabled or disabled according to whether `OMP_NESTED` was set to `true` or `false` respectively. There was no control on nesting depth—a dangerous design.

OpenMP 3.0 (July 2013) made amends for this by introducing `OMP_MAX_ACTIVE_LEVELS` which, when set to a non-negative integer, places a corresponding limit on the nesting depth of parallel regions of the program. However, this change rendered `OMP_NESTED` redundant, since setting `OMP_MAX_ACTIVE_LEVELS=1` is sufficient on its own to disable nested parallelism. Moreover, in some implementations, setting `OMP_MAX_ACTIVE_LEVELS>1` would override setting `OMP_NESTED=false`.

For this reason, the redundant internal control variable *nest-var* was eliminated in OpenMP 5.0 (November 2018). The associated API routines `omp_set_nested()` and `omp_get_nested()`, as well as the `OMP_NESTED` environment variable, were deprecated. Compiler implementations often lag well behind the release of OpenMP specification versions, but some implementations that report themselves as compliant with an older OpenMP specification nevertheless partially implement changes, such as this one, introduced in a newer one. The commands `am -v` and `am -e` can be used to determine the supported OpenMP version and the response of OpenMP internal control variables to environment variable settings.

Nesting affects the performance of the Fourier and Hartley transforms at the heart of *am*'s delay spectrum and instrumental line shape (ILS) convolution computations. The transforms use recursive divide-and-conquer into half-size sub-transforms down to the L1 cache size, at which point recursion stops and the sub-transforms are done iteratively. If nesting is enabled, threads are spawned at each recursion level unless either the number of transforms at the next recursion level would exceed `OMP_NUM_THREADS`, or `OMP_MAX_ACTIVE_LEVELS` would be exceeded. If nesting is disabled, no threads are spawned after the first recursive division into two sub-transforms and the transform is carried out by no more than two active threads.

These transforms involve highly non-local memory access patterns, and consequently the speedup achieved by adding processors is limited by the memory and inter-processor communication bandwidth of the system. For this reason, it is often the most efficient use of system resources to limit nested parallelism using `OMP_MAX_ACTIVE_LEVELS`, guided by timing

data such as those in the next section.

A.4 Performance

In this section, simple performance benchmarks are presented that quantify the effect of various factors on the speed of *am* computations. For line-by-line computations, these factors include the choice of line shape function, the value of the line-by-line tolerance parameter `tol`, and the setting of `OMP_NUM_THREADS`. For excess delay spectra and ILS convolutions, the settings of `OMP_NUM_THREADS`, `OMP_NESTED` (deprecated since OpenMP 5.0 as noted above), and `OMP_MAX_ACTIVE_LEVELS` all affect computational speed and efficiency. The benchmarks were run on three different systems:

1. Apple MacBook Pro 2021 notebook

Apple M1 Pro CPU

10-core processor with 8 performance cores and 2 efficiency cores

128 K 8-way L1 data cache per performance core

64 K 8-way L1 data cache per efficiency core

macOS 14.6.1 “Sonoma”, MacPorts gcc 14.2.0, OpenMP version 4.5 (201511)

2. Dell R640 two-socket server

2×Intel Xeon Gold 6148 “Skylake” CPUs with SMT disabled

20 cores per CPU

32 K 8-way L1 data cache per core

Rocky Linux 8.9 (Linux 4.18.0 kernel), gcc 14.2.0, OpenMP version 4.5 (201511)

3. Dell R6515 single-socket server

AMD EPYC 7713P “Milan” CPU with SMT disabled

64-core multi-chip module, with eight 8-core CPU chips and one common I/O chip

32 K 8-way L1 data cache per core

Rocky Linux 8.9 (Linux 4.18.0 kernel), gcc 14.2.0, OpenMP version 4.5 (201511)

The compute server runs were carried on the Smithsonian Institution High Performance Cluster (<https://doi.org/10.25572/SIHPC>). Of these, the Intel runs were executed in the interactive queue on a single node, while the AMD runs were executed on several identically-configured nodes of the same type in queue scheduled mode.

For benchmarks such as these, it is useful to observe the timing statistics over multiple runs for each test condition. The best-case timing is a measure of algorithm performance under conditions of optimal thread placement across processors, and the difference between median and best-case times is a measure of how consistently the operating system, the OpenMP implementation, and the queueing system in the case of the shared server work together to

achieve this. For the queue scheduled server runs, the various test conditions were randomly shuffled and sequentially queued to spread variations in server load and core placement across the various test conditions. In the plots below, best-case timings are plotted with black symbols, and median timings over nine runs are plotted in light gray; with a few exceptions these are very similar. Timings for single-threaded *am* are plotted with circled symbols for comparison. As we'll see, when interpreting timing results it is important to remember that there is often an interaction between processor clock speed and system load (or a proxy such as cpu temperature) that is not exposed to user control.

A.4.1 Line-by-line computations

The following script is used to evaluate line-by-line computation speed:

```
# Example A.1 - A line-by-line speed benchmark

am -b - <<- end
  f 0 GHz 1000 GHz 10 MHz
  tol ${TOL}
  layer
  P 30 mbar
  T 220 K
  lineshape ${LINESHAPE} o3
  col o3 1e19 cm^-2
end
```

This script uses the `bash` shell's “here document” feature to feed a minimal configuration file directly into *am* via standard input. The `-b` option tells *am* to run the computation, but to write no output other than its internally-measured elapsed wall clock time (for OpenMP) or CPU time (for single-threaded *am*). The `tol` setting and `lineshape` type are taken from the shell environment, making it easy to iterate over lists of these settings and observe their effects. Here, we look at the performance of the above three machines as a function of `OMP_NUM_THREADS` when running Example A.1 with `TOL` ranging from 0 to `1e-4`, and with `LINESHAPE` set to either `Gross` or `voigt-Kielkopf`. These data are useful as a guide to choosing the `tol` parameter, and also illustrate the relative cost and parallel performance of the simple `Gross` collisional line shape versus the more computationally-intensive `voigt-Kielkopf` line shape, which has an `exp()` call in its inner loop.

Figure A.1 shows execution times for the Apple M1 Pro. This is a heterogeneous processor with eight fast performance cores (organized as two clusters of four sharing L2 cache) and two slower efficiency cores (also sharing L2 cache). Normally, macOS will schedule user foreground processes on the fast cores and low-priority system background tasks on the slower cores. However, the OpenMP implementation tested here defaults to `OMP_NUM_THREADS=10`, and with this default setting the system will obligingly run *am* on both fast and slow cores, in some cases actually slowing performance by bringing a third L2 cache into play or causing fast cores to wait at synchronization points for slow cores to finish their work. Instead, on this CPU consistent high performance is achieved by setting `OMP_NUM_THREADS=8`. In addition, explicit control over core binding is possible using the macOS `taskpolicy` command. For example,

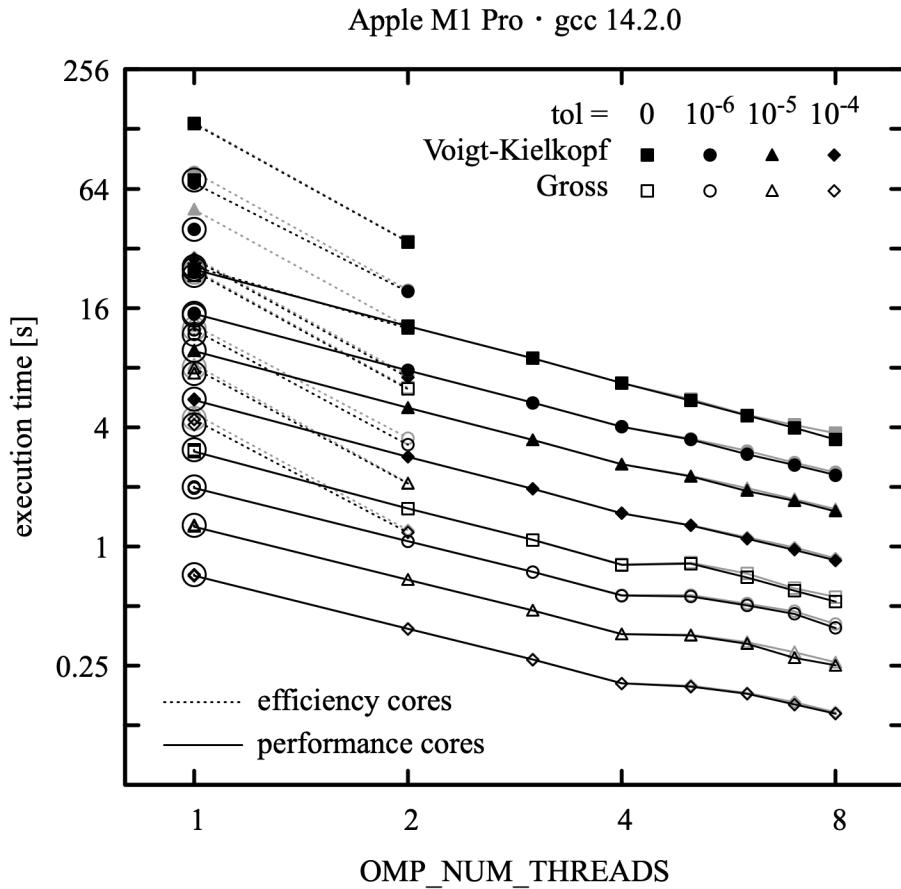


Figure A.1 Execution time for Example A.1 as a function of the number of OpenMP threads for a heterogeneous 10-core processor (Apple M1 Pro) with eight performance cores and two efficiency cores. Runs were bound exclusively to performance cores or efficiency cores as indicated. Best-case timings are in black and median timings over 9 runs are in gray. Circled symbols are for single-threaded *am*.

```
$ taskpolicy -a am config_file.amc
```

will force *am* to run on the performance cores, whereas

```
$ taskpolicy -b am config_file.amc
```

will force *am* to run on the efficiency cores. Both cases are plotted in Figure A.1. The organization of the eight performance cores into two clusters of four is reflected in the execution time versus thread count. For the cpu-intensive work of computing the Voigt line shape performance scales approximately linearly with `OMP_NUM_THREADS`, whereas for the less compute-

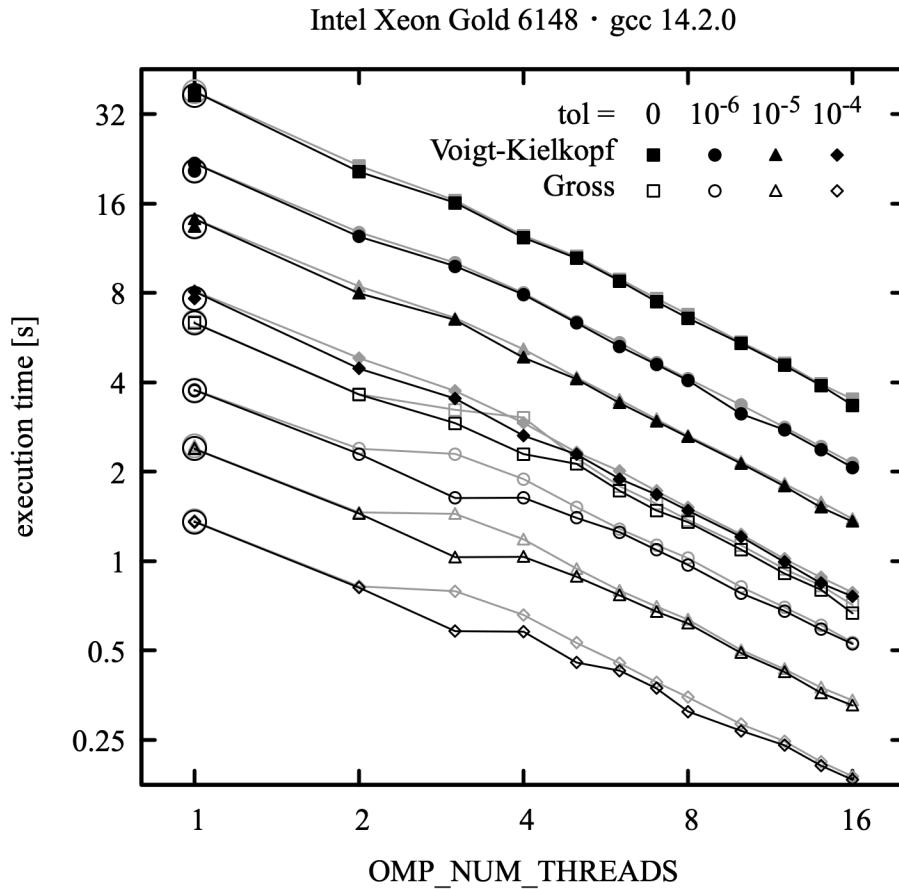


Figure A.2 Execution time for Example A.1 as a function of the number of OpenMP threads for a two-socket server with 20 cores per CPU (Intel Xeon Gold 6148) for a total of 40 cores. Best-case timings are in black and median timings over 9 runs are in gray. Circled symbols are for single-threaded *am*.

intensive Gross line shape the overhead of maintaining cache coherence for data shared between clusters becomes relatively more significant, resulting in a distinct break going from four to five cpus. Surprisingly at first, for the two efficiency cores performance nearly quadruples going from one thread to two for the OpenMP cases. Starting up the macOS `powermetrics` utility and logging its output during these runs reveals the cause of this apparent super-linear speedup—the “E-cluster” clock frequency increases from about 1 GHz to 2 GHz when the active load on the efficiency cores increases from one to two threads. This comes at a cost: going from one active thread to two, the `powermetrics` log shows that CPU power increases from approximately 100 mW (1 GHz, one active thread) to 600 mW (2 GHz, two active threads), so 6× the power is needed to do 4× the work per unit time. The benefit of this clock management strategy is

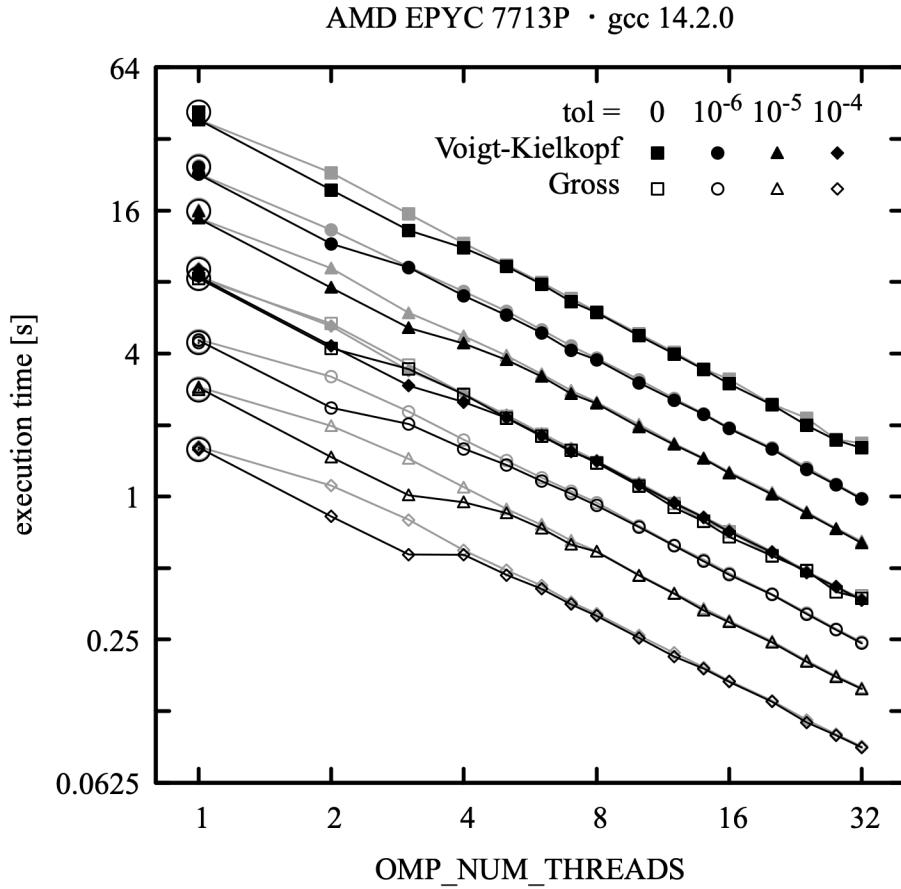


Figure A.3 Execution time for Example A.1 as a function of the number of OpenMP threads for a 64-core single-socket server CPU (AMD EPYC 7713P) in a shared environment. Best-case timings are in black, median timings over 9 runs are in gray. Circled symbols are for single-threaded *am*.

significant energy conservation on an idle system lightly loaded with mostly inactive background tasks.

Figure A.2 shows timings for a two-socket system with two 20-core Intel Xeon Gold 6148 processors for a total of 40 cores. The timing data show the expected speedup with number of OpenMP threads, with an additional factor attributable to boosted clock speed at light load. That is, best-case execution times (black) for three threads or fewer follow a similar logarithmic slope as for higher thread number, but with an offset towards shorter execution time attributable to higher clock speed. For two to three threads, the clock speed is in a variable transition region, as indicated by the spread between best-case (black) and median (gray) timings, before settling into a consistent fully-loaded clock speed beyond about four threads.

Figure A.3 shows timings for a single-socket system with a 64-core AMD EPYC 7713P processor. As in the previous example, the timing data show evidence of boosted clock speed at lighter load, with some variability between best and median case timing that may be associated with other activity on the node. With more than four active OpenMP threads, the CPU settles into its nominal fully-loaded clock frequency. This processor is a multi-chip module organized as 8 CPU chips, each with 8 cores sharing L3 cache, and a single common I/O chip supporting 8 memory channels. The 8 memory channels can be configured at boot time to partition the processor symmetrically into 1, 2, or 4 nonuniform memory access (NUMA) domains. In this instance, the system was configured as a single NUMA domain, which is a good fit for a cache-friendly but NUMA-unaware application like *am*. With no special attention to core placement, consistent speedup is observed as a function of `OMP_NUM_THREADS` even when running on a set of cores spanning multiple CPU chips.

In general, *am* will perform best when running on a set of CPU cores in the same NUMA domain, especially on multi-socketed systems. Lameter (2013) offers an excellent summary of NUMA issues arising in hardware, applications, and operating systems, and tools for analyzing and controlling NUMA placement of a task. However, since a likely reason for running *am* on a large shared system is to compute many independent cases as part of a statistical study or data analysis, the most efficient strategy may be to run the program with one thread and parallelize across cases. When predicting the total run time for such a job, remember that a fully loaded CPU will likely run at a lower clock speed than it will when running a single instance as a timing test. Also bear in mind that *am* compiled as a single-threaded program may (or may not) measurably outperform multi-threaded *am* running with `OMP_NUM_THREADS=1`.

A.4.2 Delay spectra and ILS convolution

To measure the performance of excess delay computations, the excess delay is computed for a null spectrum using the following script:

```
# Example A.2 - Excess delay computed for a null spectrum.

am -b - <<- end
    f 0 GHz 15 THz 10 MHz
    output L
end
```

Similarly, to measure the performance of ILS convolution, we apply a rectangular ILS to a null spectrum:

```
# Example A.3 - ILS convolution with a null spectrum.

am -b - <<- end
    f 0 GHz 15 THz 10 MHz
    ils rectangle 1 GHz
end
```

Figures A.4 through A.6 plot the execution times for these two scripts as a function of `OMP_NUM_THREADS` for the same three machines covered in the previous section. For these runs,

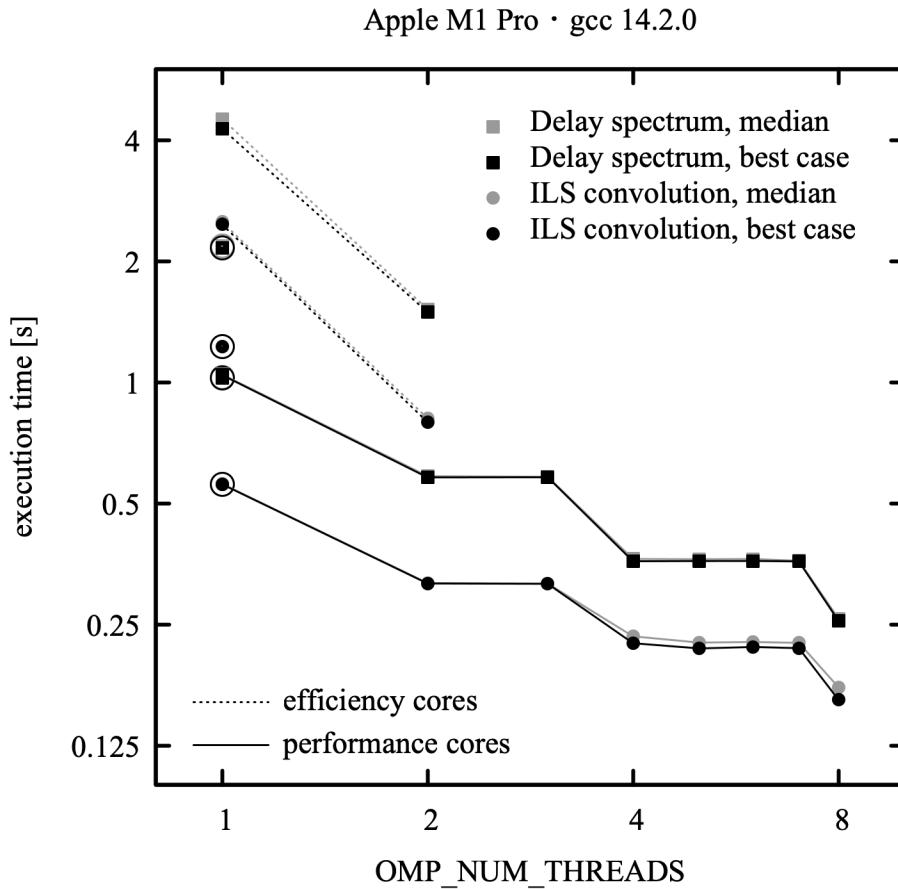


Figure A.4 Execution time for Examples A.2 and A.3 as a function of the number of OpenMP threads for a heterogeneous 10-core processor (Apple M1 Pro) with eight performance cores and two efficiency cores. Runs were bound exclusively to performance cores or efficiency cores as indicated. Best-case timings are in black and median timings over 9 runs are in gray. Circled symbols are for single-threaded *am*.

`OMP_NESTED` was set to `true`, and `OMP_MAX_ACTIVE_LEVELS` was not set; these environment variable settings are appropriate for the OpenMP version (4.5) used here (version 4.5), but take note of the deprecation of `OMP_NESTED` in later versions of OpenMP, discussed above in section A.3.2.

For all three machines, the initial speedup going from one to two threads is fairly consistent, averaging about 1.8 \times . As in the line-by-line case discussed above, the efficiency cores on the M1 Pro are an exception, exhibiting super-linear speedup in the OpenMP cases—at the expense of power consumption—related to the macOS CPU clock management strategy. This clock

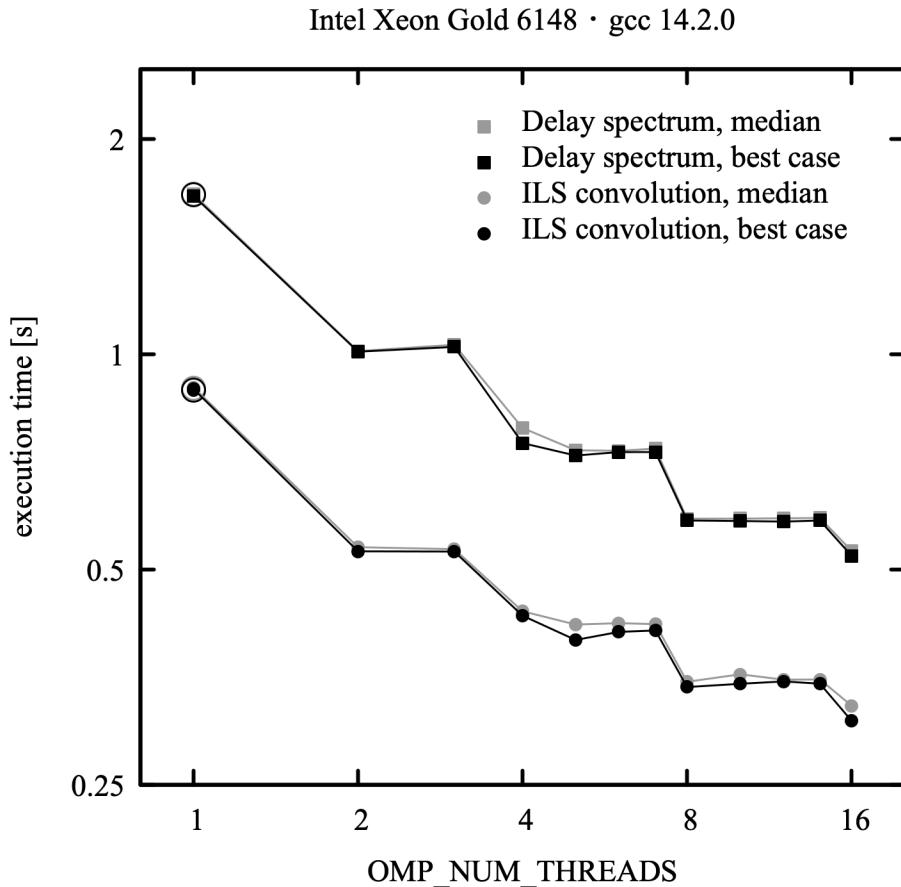


Figure A.5 Execution time for Examples A.2 and A.3 as a function of the number of OpenMP threads for a two-socket server with 20 cores per CPU (Intel Xeon Gold 6148) for a total of 40 cores. Best-case timings are in black and median timings over 9 runs are in gray. Circled symbols are for single-threaded *am*.

management strategy sometimes causes inconsistent results in the single-threaded runs, since if any one of many system background processes happen to wake up during the run, the clock frequency will be bumped up by a factor of two. Here, this is observed in the single-threaded runs.

These computations use a recursive divide-and-conquer algorithm for which speedups occur at factor-of-two increases in thread number. They involve significant data sharing between threads, and consequently the speedup obtained by adding threads diminishes as the available data bandwidth between cores is saturated. Nevertheless, on the notebook processor setting `OMP_NUM_THREADS=4` and `OMP_NESTED=true` (or `OMP_MAX_ACTIVE_LEVELS=2` on later OpenMP versions) would offer a useful performance benefit. On the compute servers, although an

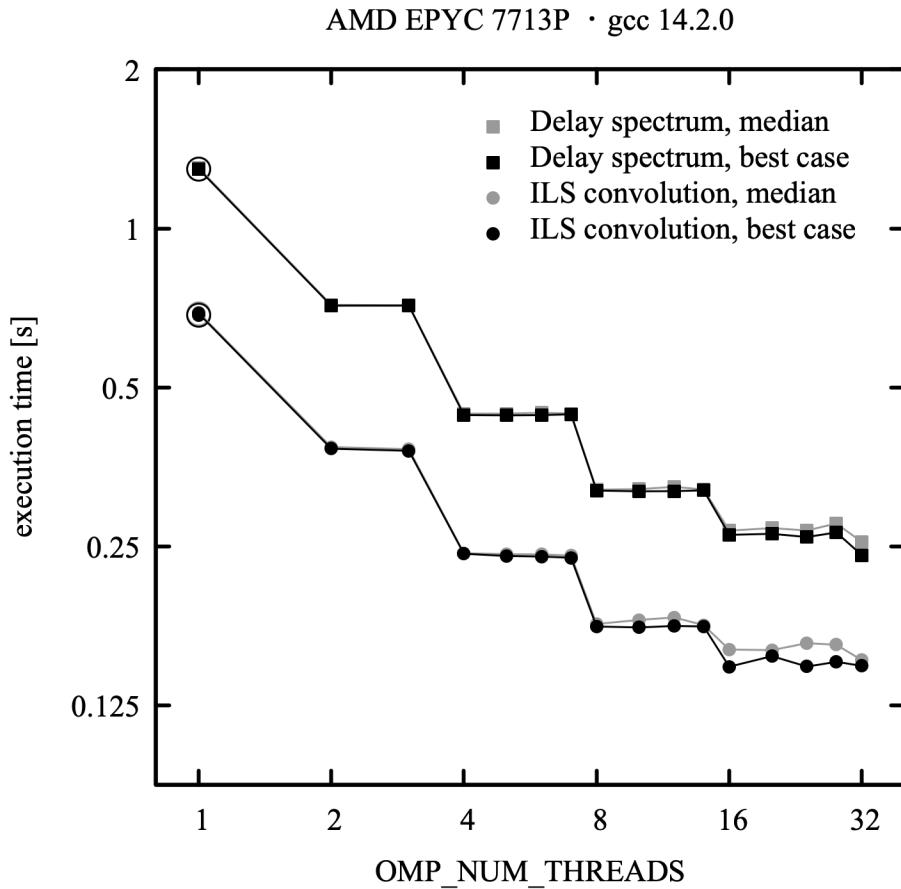


Figure A.6 Execution time for Examples A.2 and A.3 as a function of the number of OpenMP threads for a 64-core single-socket server CPU (AMD EPYC 7713P) in a shared environment. Best-case timings are in black, median timings over 9 runs are in gray. Circled symbols are for single-threaded *am*.

incremental boost in performance can be obtained by going to higher levels of nesting, it is undesirable to obtain relatively small gains by saturating data bandwidth shared with other users. Instead, limiting nesting to two levels would be a reasonable tradeoff between speedup and use of bandwidth. As noted above, for problems involving runs over many independent cases the most efficient strategy may be to run *am* single-threaded and parallelize across cases.

References

- C. Lameter 2013, “NUMA (Non-Uniform Memory Access): An Overview.” ACM Queue 11:40. (<https://doi.org/10.1145/2508834.2513149>)

Appendix B: Reference Manual

B.1 Keywords and statement syntax

All *am* configuration file statements start with a keyword, for which there is an associated syntax. Configuration file comments, and blank lines, are allowed as well. Any of the characters

```
~!@#$^&*()|{}[]:;?></
```

occurring as the first character of a token on any line, outside of a quoted string, will cause the rest of that line to be ignored. The character '?' is a special comment character. If it is the first character on a line, it flags that line as help text. Help text will be echoed to `stderr` if a user fails to supply all the command line parameters needed by a configuration file.

Configuration file keywords recognized by *am*, and their associated syntax and usage, are listed alphabetically in Section B.3, Keyword reference, below. In each heading, the first word is the keyword, followed by the argument syntax. Keywords appear in `normal` typeface, and arguments appear in *italics*. Many of the arguments are dimensioned constants consisting of a number followed by a unit. A complete list of units recognized by *am* is given in Section B.5, Units, which appears at the end of this appendix.

In the syntax description, certain sequences have the meanings given in Table B.1. Parenthesis are used to group terms for clarity, but do not themselves form part of the syntax.

B.2 Fit and differentiation variables

Many model parameters can be flagged as fit variables or as differentiation variables for Jacobians. In the keyword syntax descriptions below, if *parameter* is allowed to be a fit or differentiation variable, then there is an optional argument *dparameter*, possibly with an associated *unit*, included in the syntax. For fits, *dparameter* sets the range of *parameter* over the initial fit simplex, and is also used as a characteristic scale for determining convergence. For Jacobians, *dparameter* is an estimate of the curvature scale of the parameter, and is used for

TABLE B.1. KEYWORD SYNTAX NOTATION

SEQUENCE	MEANING
[arg]	arg may appear zero or one time
{arg}* {arg}+	arg may appear zero or more times
arg1 arg2	arg1 or arg2 may appear

computing the optimal step size for numerical differentiation. If a configuration file does not contain a `fit`, `fits`, or `jacobian` statement, then `dparameter` is ignored.

B.3 Keyword reference

`col`

`col` is a synonym for `column`, below.

```
column column_type col_density unit [dcol_density unit] |
column column_type (hydrostatic | vmr) [vmr [dvmr]] |
column h2o_column_type RH RH% [dRH%]
```

A `column` statement (synonym `col`) adds a column of an absorbing species to the current layer. The species is indicated by `column_type`. For a complete guide to column types, see Section B.4.

Column density can be specified directly, or implicitly via a volume mixing ratio or relative humidity. To specify the column density directly, the first form is used. Some examples are:

```
column co 1.7e14 cm^-2
column h2o 2.4 mm_pwv
column o3 310 DU
column lwp_abs_Rayleigh 0.5 kg*m^-2
```

If the column density is a fit or differentiation variable, this is indicated by the presence of a second dimensioned value giving its initial range or characteristic scale, as in

```
column co 1.7e14 cm^-2 1.0e15 cm^-2
```

To specify the column density by volume mixing ratio, the second form is used. The mixing ratio is introduced by the keyword `vmr` or `hydrostatic` in hydrostatic models, and by the keyword `vmr` in non-hydrostatic models. If no volume mixing ratio is given, a default value is used if one is defined for the given column type. Some examples are:

```
column h2o hydrostatic 0.012
column o3 vmr 4.3e-6
column o2 vmr # use default value
```

The column density for a column specified this way is the product of the volume mixing ratio and the total gas column density on the layer. The total column density is computed using either Eq. 1.1 for non-hydrostatic models, or Eq. 1.7 for hydrostatic models. For non-hydrostatic models, this implies that the layer thickness h must be specified.

If the mixing ratio is a fit or differentiation variable, this is indicated by the presence of a second number dvmr indicating its initial range or characteristic scale, as in

```
column h2o vmr 0.05 0.01
```

Note that the syntax does not allow a default value of the mixing ratio to be implied in this case; it must be given explicitly.

When a default mixing ratio is used, it is scaled to account for any user-defined mixing ratios on the same layer. For example, on a layer defined as

```
layer
Pbase 1013.25 hPa
Tbase 23 C
column dry_air hydrostatic
column h2o hydrostatic 0.02
```

the default mixing ratio of 1.0 for the `dry_air` column type will be rescaled to 0.98 to maintain normalization of the mixing ratios on the layer. Automatic scaling is never applied to user-defined mixing ratios. If the sum of user-defined mixing ratios for all the columns on a layer exceeds unity, *am* will issue a warning, but the computation will proceed anyway. This allows constructs such as

```
.column o2 0.21    # line-by-line absorption
.column o2air 0.21 # collision-induced absorption
```

to be used to manually overlay different absorption modes of the same molecular species on the same layer (as opposed to using a composite column type such as `dry_air`). On such a layer, all mixing ratios should be user-defined, otherwise default mixing ratios will not be correctly normalized.

For water vapor column types, the mixing ratio may also be specified by relative humidity, using the third form of the `column` statement. Relative humidity values must always be followed by a percent sign, with no intervening space. For example

```
column h2o RH 50%
```

Following meteorological convention, relative humidity is based on the saturation vapor pressure over a flat liquid surface. This convention applies even at sub-freezing temperatures, and for vapor in equilibrium with small droplets. In *am*, the saturation vapor pressure is computed at the layer midpoint temperature, using Eq. 10 of Murphy and Koop (2005). This formula is numerically valid at temperatures from 332 K down to 123 K, well below the spontaneous ice nucleation point for water, which is about -45 C. If desired, a more physically-motivated relative

humidity based on saturation vapor pressure over ice may be used by substituting the keyword `RHi` for `RH`. This uses Eq. 7 of Murphy and Koop (2005) for hexagonal ice Ih, valid for temperatures from the triple point at 273.16 K down to 111 K.

When computing mixing ratios from relative humidity, liquid of standard isotopic composition is assumed, and changes in isotopic fraction between phases are ignored. This is an approximation, as significant isotopic fractionation can occur in nature—see, for example, Kakiuchi and Matsuo (1979) and Bolot, Legras, and Noyer (2013).

The keywords `RH_offset` and `RH_scale`, described below, may be used to apply global calibration adjustments to the RH values in a configuration file.

`dg_dz gradient unit`

Sets the vertical gradient dg/dz of the gravitational acceleration g . The default is a constant-gravity approximation:

```
dg_dz 0 s^-2
```

For the Earth, the gradient varies by less than 1% with latitude; the value at 45 degrees latitude can be set with

```
dg_dz -3.086e-6 s^-2
```

The corresponding reduction in g at 10 km altitude is about 0.3%.

`dP pressure unit [dpressure unit]`

For hydrostatic models, a `dP` statement can be used to set the pressure difference dP across the current layer.

All layer pressures in a model must be specified using the same pressure variable. Normally, it is most convenient to specify hydrostatic layer pressures using the base pressure P_{base} , thereby defining the model in fixed pressure coordinates. However, to guarantee that the layer order remains unchanged during fits, P_{base} is not allowed to be a fit variable. (Otherwise layer base pressures could cross.) Specifying layer pressures in terms of dP allows pressure to be introduced as a fit or differentiation variable in hydrostatic models, although it should be noted that interpolated source, observing, and tangent levels, which are allowed as fit or differentiation variables, will cover most needed cases.

`dsb_utol_ratio ratio [dratio]`

Sets the USB/LSB sideband response ratio for the `ifspec` mode and the `dsb` instrumental line shape mode. By default, `ratio` is 1.0.

f

`f` is a synonym for `frequency`, below.

`fif fmin unit fmax unit`

The `fif` statement restricts the range of IF spectra. By default, IF spectra cover the maximum IF frequency range possible, given the LO frequency and sideband type specified in the `ifspec` statement, and the model computation grid specified in the `frequency` statement. This means, among other things, that double-sideband spectra always start at $\nu_{IF}=0$ by default. A savings in computation and I/O can be realized by restricting the IF output frequency range.

For example, a double-sideband spectrum over an IF frequency range of 4 GHz–8 GHz, with an LO frequency of 225 GHz, could be set up using the following statements in the model configuration file:

```
f 215 GHz 235 GHz 0.5 MHz
ifspec dsb 225 GHz
fif 4 GHz 8 GHz
```

If the `fif` statement were omitted, the output frequency range would be 0–9.9995 GHz.

`fit data_type filename`

Indicates that a fit to a single data file or stream is to be performed. The argument `data_type` indicates the type of spectral data being fit, and can be any of the output array types tabulated under `array_name` in `output`, below. Note that the output array type selected for fitting does not need to be specified as a program output.

`filename` names the fit data source. If `filename` contains spaces or `am` comment characters, it must be enclosed in quotes. A `filename` of `stdin` or `-` indicates that the data are to be read from the standard input. By default, the data are assumed to consist of two columns, containing frequency data in the first column, and spectral data in the second, in the default units tabulated under `unit` in `output`, below. The column order and format can be changed using the `fit_data_columns`, `fit_data_delimiters`, and `fit_data_format` keywords. The `fit_data_columns` keyword can also be used to specify optional fields for channel bandwidth and weighting factors. The units for the fit data can be modified using the `fit_data_units` keyword.

Two special keywords can be used in data files or streams. The `end` keyword signals the end of a data block, enabling multiple data sets to be included in a file or stream and fitted sequentially. The `set` keyword can be used to embed certain configuration statements in the fit data, enabling certain model parameters to be modified on-the-fly between fits. See `set`, below.

`fits data_type filename`

Same as `fit`, but the file named by `filename` contains a list of data files to be fit sequentially. The special file name `restart` will cause `am` to start over from the head of the list.

`fit_data_cols`

`fit_data_cols` is a synonym for `fit_data_columns`, below.

fit_data_columns c1 c2 [c3 [c4]]

By default, *am* assumes that fit data are in the first two columns of a data file, with frequency in the first column, and spectral data in the second column. The **fit_data_columns** statement is used to change the column order, or to designate columns containing the optional channel bandwidth and weighting factor fields. The arguments are:

- c1* is the column number for frequency data ν_i
- c2* is the column number for spectral data s_i
- c3* is the column number for the optional channel bandwidth b_i
- c4* is the column number for the optional weighting factor w_i

Normally, the weighting factor w_i is the reciprocal of the standard deviation σ_i of spectral point s_i . If *c3* or *c4* are not defined, or are set to zero, then default values $b_i=0$ or $w_i=1$ are assumed for the bandwidth or weighting factor, respectively. The units for b_i and ν_i must be the same, and the units for w_i are the reciprocal of the units of s_i . Frequency and spectral data units can be set with the **fit_data_units** keyword, described below.

The default setting of **fit_data_columns** corresponds to

```
fit_data_columns 1 2
```

A data source with frequency and channel bandwidth in the first two columns, and spectral data in the third column, would be read using

```
fit_data_columns 1 3 2
```

If the data include weighting factors for each point, but no channel bandwidth, the bandwidth column number must be included, but set to zero. For example, if the frequency, spectral data, and weights are in columns 1, 4, and 5 respectively, the data would be read using

```
fit_data_columns 1 4 0 5
```

To change the column delimiters, use the **fit_data_delimiters** keyword. For data sources with more complex formats, see **fit_data_format**.

fit_data_delimiters string

By default, the column delimiters for fit data are spaces and tabs. The **fit_data_delimiters** statement can be used to define an alternative set of delimiter characters. For example, to read comma-separated data, use

```
fit_data_delimiters ","
```

fit_data_format string

The **fit_data_format** statement is for reading data which are not in simply-delimited columns. *string* is a C standard library `scanf()`-style format string which can be used to read up to four columns. If needed, these can be put into the right order using the

TABLE B.2. FIT ESTIMATOR TYPES

<i>estimator_type</i>	GOODNESS-OF-FIT ESTIMATOR
<code>absolute_residuals</code> (default)	$E = \sum_i w_i \cdot r_i $
<code>squared_residuals</code>	$E = \sum_i (w_i \cdot r_i)^2$
<code>logarithmic</code>	$E = \sum_i \ln\left(1 + \frac{1}{2}(w_i \cdot r_i)^2\right)$

`fit_data_columns` command. For example, suppose the data were in fixed-width fields of 8 characters, containing frequency, channel bandwidth, a third field to be ignored, and the spectral data, in that order. Then the statements

```
fit_data_format "%8lf%8lf%*8lf%8lf"
fit_data_columns 1 3 2
```

could be used to read three numbers for (ν_i, b_i, s_i) from the first, second, and fourth fields on each line, and interpret them in the correct order (ν_i, s_i, b_i).

`fit_data_units freq_unit spectrum_unit`

By default, the frequency and spectral units for fit data are assumed to be the same as the model units. The `fit_data_units` statement enables the user to specify alternate units for the frequency and spectral data in a fit data file. Frequency and channel bandwidth data must be in the same units, specified by `freq_unit`. For dimensionless quantities such as transmittance or opacity, `spectrum_unit` must be specified as “none”. Units recognized by `am`, and default units, are tabulated in section B.5, Units.

`fit_estimator estimator_type`

`fit_estimator` specifies the function of the spectral residuals which serves as the goodness-of-fit estimator to be minimized during fits. Table B.2 lists the goodness-of-fit estimator functions for the possible choices of `estimator_type`. For each measured frequency channel centered at frequency ν_i , if the measured spectral quantity is s_i , and the corresponding modeled spectral quantity is $s_i^{(mod)}$, then the spectral residual is defined as $r_i = s_i - s_i^{(mod)}$. The translation from the model frequency grid to the data frequency ν_i depends on the channel bandwidth b_i . When $b_i = 0$, $s_i^{(mod)}$ is the model spectrum interpolated to frequency ν_i . If $b_i > 0$, $s_i^{(mod)}$ is a trapezoidal integral of the model spectrum over a frequency interval b_i centered at ν_i . The measured data points may also be associated with weighting factors w_i , which are normally the reciprocal of the estimated error σ_i for the measured spectral data point s_i . If weighting

factors are not supplied with the data set, then they are assumed to be equal to unity for all data points, formally carrying units which are the reciprocal of the units of the spectral data.

fit_iter *n_iter*

`fit_iter` sets *n_iter*, the maximum number of iterations allowed in the downhill simplex minimization. The default corresponds to

```
fit_iter 1000
```

fit_output (*config* | *spectrum* | *residual*) 0|1

By default, *am* writes three files during fits, containing model spectra, fit residuals, and post-fit model configuration data. The `fit_output` statement can be used to enable or disable output of each of these files. For example,

```
fit_output residual 0
```

will suppress output of a file containing fit residuals. The `fit_output` statement can be used on the fly from within a fit data file or stream, using the `set` facility. This is useful in multi-data-block files for dumping complete fit output for a particular data block or blocks, with only summary fit output being written to `stdout` for the other data blocks.

fit_reinit 0|1

For sequential fits, either to multiple data blocks within a single data file or stream, or to multiple data files using `fits`, the default behavior of *am* is to use the converged fit parameters from the prior fit as the starting point for the current fit. The statement

```
fit_reinit 1
```

will instead cause the initial values to be reset to those given in the configuration file. In either case, the initial size of the fit simplex at the start of each fit will be set to the ranges given in the configuration file. The statement

```
fit_reinit 0
```

will set or restore the default behavior.

fit_tol *tol* [*max_restarts*]

`fit_tol` sets the convergence tolerance for fits. When the scaled simplex diameter defined in Eq. 7.3 is less than *tol*, the fit is deemed to have converged. The fit is always restarted from the converged parameters, using the original simplex dimensions, and run again. If two successive minimizations converge to the same set of fit parameters, to within a scaled distance *tol*, then the fit terminates successfully. The optional argument *max_restarts* specifies the maximum number of restarts before the fit will terminate unsuccessfully. The default behavior of *am* corresponds to

```
fit_tol 1e-4 5
```

fit_track_residuals gain | reset | off

fit_track_residuals controls a recursive filter which computes an exponentially-weighted average of the residual spectra of sequential fits, and subtracts it from the reported residuals and their associated statistics. This is an experimental feature.

fit_verbose 0|1

Setting

```
fit_verbose 1
```

will cause the vertex coordinates of the fit simplex, the value of the fit estimator at each vertex, and other diagnostic information to be written to **stderr** after each iteration as a fit proceeds. The default is quiet operation, corresponding to

```
fit_verbose 0
```

fout fmin unit fmax unit

The **fout** statement restricts output spectra to a subset of the frequency grid over which the model is calculated. This saves file space or I/O overhead when all the points are not needed in the final output. For example, if the configuration file starts out with:

```
f 130 GHz 170 GHz 10 MHz
fout 150 GHz 150 GHz
```

the model will be calculated over a grid ranging from 130 GHz to 170 GHz, but the output will be restricted to a single frequency point at 150 GHz. In the design of a CMB experiment utilizing the 150 GHz atmospheric window, this could be used to convolve a spectrum with an instrument function up to 40 GHz wide to obtain the total power in a channel centered at 150 GHz.

Another application where **fout** is useful is in the computation of delay spectra as discussed in Chapter 6.

freq

freq is a synonym for **frequency**, below.

frequency fmin unit fmax unit df unit

Sets the frequency grid for computing model spectra. The frequency grid is always aligned to the origin of the frequency axis. That is, it consists of all points $i \cdot d\nu$, where i is an integer, on the closed interval $[\nu_{min} \dots \nu_{max}]$. This alignment is required for computation of delay spectra by discrete Hilbert transformation. It also facilitates caching of absorption coefficients to disk, since all grids with the same $d\nu$ are aligned to one another.

g acceleration unit

In hydrostatic models, the **g** statement is used to modify g , the combined gravitational and centrifugal acceleration at the planetary surface. The default value is the BIPM standard value $g=9.80665\text{ m s}^{-2}$, a triumph of legal precision over geophysical variation, which ranges from $g=9.78\text{ m s}^{-2}$ at the equator to $g=9.83\text{ m s}^{-2}$ at the poles.

By default, the vertical gradient $\frac{dg}{dz}=0$, making g constant through the atmosphere. The gradient can be modified with the **dg_dz** statement.

geometry plane-parallel | spherical | limb

Sets the geometry mode. The default is **plane-parallel**.

h length unit [dlength unit]

For non-hydrostatic models, sets the thickness h of the current layer. For hydrostatic models, a layer thickness h can be defined only if there is no pressure drop across the layer. This allows a constant-pressure propagation path of fixed length, such as the internal optical path of an instrument or a cross-section of a gas plume, to be included in these models.

headers 0|1

By default, *am* does not print headers above output data columns. The statement

```
headers 1
```

in the configuration file will cause space- or comma-delimited column headers to be printed in a single line at the top of **text** or **csv** output files, respectively. For **npy** output files, when headers are off the output is stored as a single NumPy matrix with columns corresponding to output spectra and rows corresponding to frequency. When headers are turned on, **npy** files are written as a NumPy structured array consisting of a single column of structures, with fields labeled by the *am* header strings. See **output**, below.

ifspec (dsb | usb | lsb) flo unit [dflo unit]

The **ifspec** statement is used to produce spectra which model the output of a heterodyne receiver. It can also be used as a means of implementing a general frequency shift.

To produce IF spectra, spectra computed on the model frequency grid are first shifted from the model computation frequency grid to the IF frequency grid, according to $\nu_{if}=\nu-\nu_{lo}$. If ν_{lo} does not lie on the original model frequency grid, the model spectra are interpolated such that the IF frequency grid points are aligned to $\nu_{if}=0$. For *upper-sideband* (**usb**) spectra, points at negative IF frequencies are discarded. For *lower-sideband* (**lsb**) spectra, points at positive IF frequencies are discarded, and the negative IF frequency points are reflected onto the positive IF frequency axis. *Double-sideband* (**dsb**) spectra are a weighted average of the **usb** and **lsb** spectra. By default, the upper and lower sidebands are equally weighted. The default upper- to

lower- sideband response ratio can be modified with the `dsb_utol_ratio` statement described above.

By default, the frequency range of the IF spectrum will be the maximum possible, given the model computation frequency grid range, the LO frequency, and the sideband mode. The frequency range can be made narrower using the `fif` statement. This is particularly useful for avoiding unnecessary computations in the gap between sidebands contributing to double-sideband spectra.

The LO frequency can be made a fit variable by giving its range `dflo` over the initial fit simplex. Note that if this is being done to fit a small frequency shift near $\nu_{lo}=0$ which might be negative, logarithmic mapping of the simplex coordinates, which is enabled by default, must be disabled using

```
simplex_log 0

ils type_name fwhm unit [dfwhm unit] {array_name}*  
    
```

Specifies an instrumental line shape (ILS) of type `type_name`, with full width at half maximum equal to `fwhm`. An optional list of output spectra, `{array_name}*, may be given to specify the output spectra to which the ILS will be applied. See output, below, for a table of array_names and corresponding spectra. ILS type_names are listed in Table 5.1 and Table 5.2. The arrays which may be convolved with an ILS are I, I_diff, tx, Tb, Trj, Tsys, and y. If the array_name list is omitted, all output arrays for which an ILS is allowed will be convolved with the ILS.`

For any ILS type, if `fwhm` is less than twice the frequency grid interval `df`, the ILS will be replaced by a discrete impulse function, and `am` will issue a warning to the user.

```
ilsmode normal | ((dsb | usb | lsb) fif unit [dfif unit])  
    
```

The `ilsmode` statement can be used with an instrumental line shape function to produce spectra which model the total or single-channel output of a heterodyne receiver *as a function of the LO frequency*. (To compute IF spectra, see `ifspec`, above.) For example, to simulate the total power output of a double-sideband heterodyne receiver with an IF frequency of 5 GHz, and a 100 MHz IF filter, the statements

```
ilsmode dsb 5 GHz
ils rectangle 100 MHz  
    
```

would be added to the configuration file. For double-sideband spectra, the sideband response ratio is unity by default, and can be changed with the `dsb_utol_ratio` statement.

```
jacobian {array_name}+ [include_estimated_errors]  
    
```

The `jacobian` statement specifies that Jacobians shall be computed for a list of one or more output arrays `{array_name}+`. These arrays must also be listed as program outputs with an `output` statement elsewhere in the configuration file. Partial differentiation is carried out with respect to the model variables identified using the syntax described in Section B.2 above. The

output units for partial derivatives will be the array output units divided by the units used to specify the value of the differentiation variable at the reference model state defined in the configuration file. In the program output, the columns of the Jacobian occur in left to right order corresponding to the order in which the differentiation variables occur in the configuration file.

For assessing the accuracy of the numerical differentiation, the `stderr` output stream includes a summary table containing estimated fractional error percentiles for each column of the Jacobian spectrum. Detailed channel-by-channel error estimate spectra can be included in the output by adding the `include_estimated_errors` flag to the `jacobian` statement. For details, see Chapter 8.

kcache off | *Tmin* *unit* *Tmax* *unit* *dT* *unit*

The `kcache` statement controls in-memory caching of absorption coefficients during fits. Absorption coefficients cached in memory are computed on a discrete temperature grid, and interpolated to the required temperature.

Line-by-line absorption coefficients are normally a function of temperature, pressure, and mixing ratio. In the case when layer pressure and composition are fixed, and only the layer temperature is a fit variable, a fit can be greatly accelerated by computing the spectral absorption coefficient at discrete temperatures as needed and interpolating to obtain the absorption coefficient at intermediate temperatures. This in-memory cache of spectral absorption coefficients is also written through to the disk cache. `Tmin`, `Tmax`, and `dT` define the range and resolution of the temperature grid. The default behavior corresponds to

```
kcache 190 K 320 K 1 K
```

The statement

```
kcache off
```

will disable caching. This can be used to validate a computation performed with interpolated absorption coefficients.

Absorption coefficient arrays are added to the `kcache` as needed, whenever the absorption coefficient at a particular grid temperature is needed to bracket a new temperature value encountered during a fit. Consequently, the memory requirements for the cache depend mainly on the resolution `dT`, and less on the range set by `Tmin` and `Tmax`, provided they cover the range of layer temperatures encountered during the course of a fit. To control memory usage, the environment variable `AM_KCACHE_MEM_LIMIT` can be used to set a user-defined limit (in bytes) on memory allocated to the `kcache`. For example, the `bash` command

```
$ export AM_KCACHE_MEM_LIMIT=0x40000000
```

would set a 1 GB limit on `kcache` memory usage. When *am* reaches either a user- or system-defined memory limit, memory for new absorption coefficient arrays is freed by discarding older cached arrays.

layer [tag]

Indicates the start of a new layer in a configuration file. A complete layer specification takes the following form:

```
layer [tag]
(P | Pbase | dP) statement
(T | Tbase) statement
[h statement]
[Mair statement]
{lineshape statement}*
{column statement}*
```

The statements following the `layer` statement may appear in any order.

The optional argument `tag` is a *layer tag*. Layer tags can be used to identify groups of layers by name, such as `troposphere` or `boundary_layer`. A layer tag can be any single word which is not a column type name. Groups of layers which share a common layer tag can have the column density of one or more species scaled independently of the other layers in a model. (See `Nscale` for details.) In the total column density summary which appears in the output to `stderr` after a model run, the total column densities for all layers sharing the same layer tag are reported in addition to the totals for the entire model.

**lineshape type_name [strict_self_broadening] {k_type}*
lineshape Doppler**

The `lineshape` statement sets line shape types to be used for line-by-line computations on the current layer. A single line shape can be applied to all line-by-line computations on a layer, or different line shapes can be applied to different species. The optional `strict_self_broadening` flag controls computation of pressure-broadened line widths.

The allowed values of `type_name` are given in Table 3.1. The default line shapes for all line-by-line column types are given in Table B.4.

If no `k_type` is given after the line shape, as in

```
lineshape Doppler
```

then the specified line shape will be applied to all line-by-line computations on the current layer. Multiple line shape statements are allowed on a layer, as in

```
lineshape Gross ch4 co co2 n2o o2_uncoupled o3
lineshape VVV_coupled o2_coupled
lineshape VVH_750 h2o_lines
```

which corresponds to the default settings on a layer containing `dry_air`, `o3`, and `h2o`.

For collisional line shapes, the line width γ is the sum of self- and air-broadened widths. By default, these are computed from the layer pressure and the volume mixing ratio of the species in question using the approximation controlled by the `selfbroad_vmr_tol` parameter as discussed in Section 2.3.1. When this approximation is not sufficient, and when it is not desired to simply turn it off globally by setting `selfbroad_vmr_tol` to zero, fine-grained control over self-broadening on a per-layer, per-species basis is possible using the `strict_self_broadening` keyword in a `lineshape` statement or statements. For example, the following statement would

turn on strict self broadening for water vapor on the current layer:

```
lineshape vvH_750 strict_self_broadening h2o_lines
```

This method could be used, for example, to apply strict self broadening to water vapor in the warm, humid lower layers of a model atmosphere, while leaving self broadening controlled by `selfbroad_vmr_tol` elsewhere.

**Mair [M0] {column_type}*
[Mair]**

In hydrostatic models, the total hydrostatic column density of a layer is computed from the average molecular mass as discussed in Section 1.1.1. The `Mair` statement controls computation of the average relative molecular mass on a layer. The average relative molecular mass is computed assuming a starting “dry air” mass `M0`, which is then adjusted to account for any column types on the layer in the `{column_type}*
[Mair]` list having non-default column densities. The default behavior is equivalent to

```
Mair 28.964 h2o o3
```

meaning that the average molecular mass on the layer will be the standard terrestrial dry air value of 28.964 grams per mole, adjusted for any water vapor or ozone included on the layer via the column types `h2o` and `o3`. Water vapor included on a layer via another column type such as `h2_16o` will not be included in the average mass computation unless the default behavior is overridden by an explicit `Mair` statement such as

```
Mair h2_16o o3
```

**Nscale [tag] column_type scale_factor [dscale_factor]
[Nscale]**

The `Nscale` statement applies a common scale factor to the column density of all columns of type `column_type` which occur in the model. If the optional argument `tag` is present, then the scale factor will only be applied to the specified column type on layers having the layer tag `tag`. (See `layer` above.) The main purpose of the `Nscale` keyword is to provide a means of varying the total column density of a variable species, such as H2O or O3, by scaling a profile of fixed form. For example, the statement

```
Nscale h2o 0.9
```

would scale the entire H2O profile defined in the model by a factor 0.9, whereas the statement

```
Nscale troposphere h2o 0.9
```

would scale the H2O column density only on those layers tagged with the layer tag `troposphere`. `Nscale` factors combine multiplicatively—if both of the above two statements appeared in the same configuration file, H2O would be scaled by 0.81 on layers tagged `troposphere`, and by 0.9 on all other layers.

For hydrostatic models, or for non-hydrostatic models with the layer thickness `h` defined on

the layer, the `nscale` factor for a column type multiplies both the column density and the volume mixing ratio for that column type. Furthermore, this is true even if the column density is defined implicitly via a default mixing ratio. Since this behavior is slightly subtle, `am` will log a warning to alert the user when a default mixing ratio is scaled.

In the configuration file echoed by `am` to `stdout`, all `nscale` factors which have been defined are included near the top of the file, and the column densities and mixing ratios reported in the individual layer definitions are the unscaled values. However, any `nscale` factors are included in the total column densities reported in the comments at the bottom of the file.

It is important to note that `nscale` operates on the column type name that appears in the configuration file, and does not “see inside” composite column types such as `h2o` or `dry_air`. This means, for example, that the statement

```
nscale h2o_lines 0.5
```

would have no effect on the composite column type `h2o`, even though `h2o` includes `h2o_lines` as a component.

```
output {array_name [unit]}+ |  
output format {array_name [unit]}*
```

The `output` statement specifies which spectra are to be written as output, and can also be used to specify an output file format. Table B.3 lists the possible values for `array_name` and `unit`; `format` is either `text` (the default), `csv`, or `npy` as discussed further below.

If no `output` statement appears in the configuration file the default output configuration corresponds to

```
output text f GHz tx none Tb K
```

If no units are specified after an `array_name`, the default unit indicated in bold in Table B.3 will be assumed. For example,

```
output f tau I
```

is equivalent to

```
output f GHz tau neper I watt*cm-2*GHz-1*sr-1
```

All output spectra must be specified in a single `output` statement. However, the output format and spectra may be specified separately. That is,

```
output npy f GHz L cm
```

is equivalent to

```
output npy  
output f GHz L cm
```

TABLE B.3. OUTPUT TYPES

OUTPUT SPECTRUM	<i>array_name</i>	<i>unit</i>
frequency	f	Hz, kHz, MHz, GHz , THz, cm-1
opacity	tau	neper , dB
transmittance	tx	none
radiance	I	watt*cm-2*GHz-1*sr-1, mW*m-2*(cm-1)-1*sr-1, RU, watt*m-2*Hz-1*sr-1, Jy*arcsec-2
radiance difference	I_diff	watt*cm-2*GHz-1*sr-1, mW*m-2*(cm-1)-1*sr-1, RU, watt*m-2*Hz-1*sr-1, Jy*arcsec-2
Planck brightness temperature	Tb	K
Rayleigh-Jeans brightness temperature	Trj	K
Rayleigh-Jeans system temperature	Tsys	K
Y-factor	Y	none
excess path or delay	L	km, m, cm , mm, um, s, ps, fs
free-space loss	tau_fsl	neper , dB
absorption coefficient	k	auto

Compared with other output spectra, the `k` (absorption coefficient) output type is a special case. Normally, the `k` output will be used in a minimal configuration file invoking a single spectral absorption coefficient computation. Otherwise, the `k` output spectrum will be the first spectral absorption coefficient encountered in the model, and a warning to that effect will be issued to inform the user. The default unit type for `k` in the `output` statement is `auto`, which will automatically set the output units to cm^2 for molecular absorption coefficients, and cm^5 for binary absorption coefficients. If units are explicitly specified in the `output` statement, and the dimensions do not match the actual output, `am` will issue a warning and correct the units. This behavior is new as of `am` version 12.3; in prior versions, the `k` output type produced a multi-column dump of every spectral absorption coefficient computed for a given model. With the new behavior, `k` works like any other output spectrum, and can be the target of a Jacobian or fit computation.

As noted above, there are three available output file formats: `text`, `csv`, and `npy`. The `text` format is the default if none is specified, and corresponds to the space-delimited text format presented in the examples throughout this manual. The `csv` format is comma-delimited text with

CRLF line breaks. The `npy` format is the compact binary format used by the NumPy `numpy.save()` and `numpy.load()` routines for persistent, machine-independent storage of NumPy arrays. This format is useful for integrating *am* into Python workflows, and provides full double-precision output in about half the storage space of the less-precise default `text` format.

By default, *am* does not write a header line to identify output columns. Headers can be turned on with the `headers` statement. Header text is written as a single space- or comma-delimited line at the top of `text` or `csv` output files, respectively. For `npy` output, when headers are off the output file contains a single NumPy matrix with columns corresponding to output spectra and rows corresponding to frequency. When headers are turned on, the file contains a NumPy structured array consisting of a single column of structures, with fields labeled by the *am* header strings. See `headers`, above.

P pressure unit [dpressure unit]

For non-hydrostatic models, sets the midpoint pressure P of the current layer.

Pbase pressure unit

For hydrostatic models, sets the layer base pressure P_{base} of the current layer. Note that P_{base} is not allowed as a fit or differentiation variable, because this could affect the ordering of model definition layers. However, interpolated source, observing, and tangent levels are allowed as fit or differentiation variables, which will cover most needed cases. Hydrostatic models can also be defined using the individual layer pressure thicknesses `dP`, and `dP` is allowed to be a fit or differentiation variable.

Pobs pressure unit [dpressure unit]

By default, the observing level pressure P_{obs} in hydrostatic models corresponds to the base level of the lowest model layer. The `Pobs` statement allows P_{obs} to be placed at an interpolated pressure level relative to the model definition layers. Extrapolation outside the defined layers is also possible, but will generate an advisory warning. When the `reverse` propagation flag is set, a level defined by `Pobs` becomes the source level.

Psource pressure unit [dpressure unit] [near | far]

By default, the source level in hydrostatic models is at $P=0$, at the top of the top model layer. (Formally, this level is geometrically at $z \rightarrow \infty$.) The `Psource` statement allows the source level to be placed at an interpolated pressure level relative to the model definition layers. When the `reverse` propagation flag is set, the level defined by `Psource` becomes the observing level.

In spherical geometry modes, when the source level is below the observer level, the projected line of sight is below the astronomical horizon and intersects the source level twice, on the near and far sides of the tangent point relative to the observer. The optional `near` or `far` keywords can be used to select the desired path. If not specified, the default is `far`, corresponding to propagation downwards from the source level, through the tangent point, then upwards to the observer level. If the line of sight defined by the zenith angle at the observer does

not intersect the source level, an error message is generated.

Ptan pressure unit [dpressure unit]

For limb paths, the default pressure P_{tan} of the tangent point is the base pressure of the lowest model layer. The **Ptan** statement allows P_{tan} to be placed at an interpolated pressure level relative to the model definition layers. Extrapolation outside the defined layers is also possible, but will generate an advisory warning.

```
PTmode midpoint | ((P | Pbase | dP) [T | Tbase]) |
PTmode extend (isothermal | dry_adiabatic | moist_adiabatic | environmental)
```

The first form of the **PTmode** statement sets the pressure and temperature variables which will be used to parametrize the model layers, as discussed in Section 1.1. If no such statement is present, *am* will deduce the **PTmode** from the first layer specification encountered. This makes the first form of the **PTmode** statement somewhat redundant in current versions of *am*, but the program will nevertheless write an informational **PTmode** statement to the output configuration data sent to **stderr**. The statement

```
PTmode midpoint
```

is a synonym for

```
PTmode P T
```

It is recognized as a valid configuration statement to retain compatibility with files written for older versions of the program.

For hydrostatic models with layers defined by base pressures and temperatures, corresponding to

```
PTmode Pbase Tbase
```

the second form of the **PTmode** statement controls the lapse rate $\Gamma = -dT/dz$ used to compute the extrapolated position of observing, source, or tangent levels placed below the lowest atmospheric model definition level, as discussed in Section 9.1.2. The default behavior corresponds to

```
PTmode extend isothermal
```

R0 radius unit

In spherical geometry modes, sets the curvature radius R_0 of the planetary surface. The default value corresponds to

```
R0 6370 km
```

Note that R_0 is the local surface radius of curvature, which typically differs from the distance between a point on the surface and the planetary center. The default value is sufficient for most

terrestrial applications. Expressions for computing a more accurate radius of curvature for a particular latitude and azimuth can be found in Section 9.1.3.

refract none | radio | optical

Sets the refraction mode used for computing path geometry. The default is `none`, meaning refraction is ignored. Refractivity models for the radio and optical limits can be applied with the respective keywords. The refractivity models account for the spectrally-averaged refractive effects of both dry air and water vapor in these limits, but do not include spectral dispersion. Dispersion can be significant in delay computations, as discussed in Chapter 6, but is typically negligible when computing the path geometry.

reverse 0|1

Reverses the order of propagation through the layers. The default is

`reverse 0`

which corresponds to the view towards the sky (downward propagation of radiation) in astronomical or ground-based remote sensing scenarios. The view from the sky towards the ground can be obtained by specifying

`reverse 1`

and setting `to` to the ground temperature. When interpolated source or observing levels have been defined, `reverse 0` corresponds to propagation from source to observer, and `reverse 1` corresponds to propagation from observer to source. In either case, `to` sets the initial radiation temperature at the starting level.

**RH_offset offset [doffset]
RH_scale scale [dscale]**

`RH_offset` and `RH_scale` are used to apply a global adjustment to all relative humidity values in a configuration file. This adjustment is defined as

$$RH_{adj} = scale * (RH - offset).$$

Volume mixing ratios are then computed using RH_{adj} . Both `offset` and `scale` can be fit parameters. Internally, `offset` is stored as $\exp(offset)$, which is always positive, so it is not necessary to disable log mapping of simplex coordinates when `offset` is a fit parameter. The default parameter settings correspond to

`RH_offset 0
RH_scale 1.0`

`RH_offset` and `RH_scale` can be used to apply a calibration correction to a profile obtained from a humidity sounding.

runtime 0|1

Turns on detailed logging of compute time for layers and columns. For fits, the logged times are for the last iteration of the fit, which can be very short if absorption coefficients did not need to be updated, or were interpolated from kcache data. The default is

```
runtime 0
```

rx_gain_factor [dfactor]

Sets a receiver gain correction factor used when computing T_{sys} or Y-factor spectra as outputs. `rx_gain_factor` is mainly intended to be used for modeling fluctuations in receiver gain when fitting time series of IF spectra. The default is

```
rx_gain_factor 1
```

sec_za sec(za) [dsec(za)]

The `sec_za` statement is an alternative to the `za` statement for setting $\sec(\theta_z)$, equal to the airmass factor in plane-parallel models, rather than setting θ_z directly.

selfbroad_vmr_tol delta

Controls the tolerance parameter for the self-broadening approximation described in Section 2.3.1. The default setting corresponds to

```
selfbroad_vmr_tol 0.003
```

The purpose of this setting is to optimize disk cache performance by ignoring negligible changes in self-broadening, which makes it more likely that cached spectral absorption coefficients can be re-used. In spectroscopic studies and other applications for which strict numerical accuracy relative to line catalog data is needed, approximate self-broadening can be disabled by setting

```
selfbroad_vmr_tol 0
```

set configuration_statement**set layer layer_number configuration_statement**

The `set` keyword is used to embed a configuration statement in a stream of fit data, allowing certain model parameters to be changed on-the-fly. Using the first form of the syntax, many top-level model parameters can be changed. For example, here the zenith angle is changed between successive blocks of radiometer data taken during a sky dip, by using `set` to insert a `za` statement at the head of each block:

```
set za 59.043000 deg
178.066 109.061 2.626 13.6652
180.087 170.375 1.327 9.71414
181.363 228.718 0.836 7.71031
182.422 261.589 0.210 3.86437
184.194 260.761 0.221 3.96428
```

```

185.262 228.320 0.822 7.64548
186.482 173.202 1.366 9.85585
188.527 114.509 2.688 13.8256
end
set za 60.00 deg
178.066 112.787 2.626 13.6652
180.087 175.186 1.327 9.71414
181.363 232.264 0.836 7.71031
182.422 262.530 0.210 3.86437
184.194 262.352 0.221 3.96428
185.262 231.848 0.822 7.64548
186.482 178.219 1.366 9.85585
188.527 118.513 2.688 13.8256
end

```

The second form of the syntax is for changing layer pressures and temperatures. For example

```
set layer 8 Tbase 270 K
```

would set T_{base} on layer 8 to 270 K. Layers are numbered starting from 0 at the top of the model. Besides pressures, temperatures, h and M_{air} , no other layer parameters can be changed using `set`.

Certain keywords, such as those that involve global properties of the model configuration, or those which would change the layout of layers and columns, cannot be used with `set`. The complete list of disallowed keywords is:

```

col column
f freq frequency
fit
fits
geometry
jacobian
kcache
lineshape
output
PTmode
refract
selfbroad_vmr_tol
simplex_log
tol tolerance

```

simplex_log 0|1

Controls logarithmic mapping of simplex coordinates. The default is

```
simplex log 1
```

which means that the coordinates in the simplex space are logarithms of the fit variables. Most often, fit or differentiation variables in *am* are positive quantities. (For an exception, see `ifspec` above.) The default setting ensures that fit variables remain positive, and results in simplex moves that involve proportional rather than arithmetic changes of the fit variables. Logarithmic mapping can be turned off with

```
simplex log 0
```

This statement must occur before the definition of any fit or differentiation variables in the configuration file.

T temperature unit [dtemperature unit]

Sets the midpoint temperature T of the current layer.

TO temperature unit [dtemperature unit]

Sets the background temperature T_0 .

Tbase temperature unit [dtemperature unit]

Sets the base temperature T_{base} of the current layer.

Tref temperature unit [dtemperature unit]

Sets the reference temperature T_{ref} used for computing radiance difference spectra and Y-factor spectra. See sections 5.2 and 5.3.2.

Trx temperature unit [dtemperature unit]

Sets the receiver noise temperature T_{rx} used for computing T_{sys} or Y-factor spectra. See Section 5.3.2.

tol delta

`tol` is a synonym for `tolerance`, below.

tolerance delta

Sets the numerical tolerance on line-by-line absorption coefficient computations. The default is

`tol 0`

which means that every catalog line is included in the computation of the absorption coefficient at every frequency grid point. If `delta` is set to a non-zero value, line-by-line computations will be accelerated by excluding lines from the line-by-line sum while attempting to maintain a fractional accuracy `delta` at every frequency grid point. A reasonable choice for many applications is

`tol 0.0001`

For more information, see Section 3.5.

zo altitude unit

Sets the altitude z_0 at the base level of the lowest model layer, which serves as the reference level from which `am` uses the hydrostatic relation to find the layer base altitudes above. The default setting corresponds to

```
z0 0 m
```

meaning that the layers are defined all the way down to the planetary surface.

Layer base altitudes z_{base} are reported to `stderr` in the units used to specify z_0 . By default, z_{base} is not reported for plane-parallel hydrostatic models unless a `z0` statement is present in the configuration file.

za angle deg|rad [dangle deg|rad]

Sets the zenith angle for propagation through the model layers. The default is

```
za 0
```

which corresponds to propagation perpendicular to the layer boundaries. In spherical geometry modes, or when refraction is turned on in plane-parallel models, `za` sets the zenith angle at the observing level, which becomes the source level if the `reverse` flag is set.

zobs altitude unit [daltitude unit]

By default, the observing level in hydrostatic models is the base of the lowest model layer. The `zobs` statement allows the observing level to be placed at an interpolated altitude level relative to the model definition layers. This is done at run time after `am` has determined the altitudes of the layer boundaries using the hydrostatic relation. Extrapolation outside of the defined model layers is also possible, but will generate an advisory warning. When the `reverse` propagation flag is set, a level defined by `zobs` becomes the source level.

zsource altitude unit [daltitude unit] [near | far]

By default, the source level in hydrostatic models is at $P=0$, at the top of the top model layer. (Formally, this level is geometrically at $z \rightarrow \infty$.) The `zsource` statement allows the source level to be placed at an interpolated altitude level relative to the model definition layers. This is done at run time after `am` has determined the altitudes of the layer boundaries using the hydrostatic relation. When the `reverse` propagation flag is set, the level defined by `psource` becomes the observing level.

In spherical geometry modes, when the source level is below the observer level, the projected line of sight is below the astronomical horizon and intersects the source level twice, on the near and far sides of the tangent point relative to the observer. The optional `near` or `far` keywords can be used to select the desired path. If not specified, the default is `far`, corresponding to propagation downwards from the source level, through the tangent point, then upwards to the observer level. If the line of sight defined by the zenith angle at the observer does not intersect the source level, an error message is generated.

ztan altitude unit [daltitude unit]

By default, the tangent level in hydrostatic models is the base of the lowest model layer. The `ztan` statement allows the tangent level to be placed at an interpolated altitude level relative to the model definition layers. This is done at run time after `am` has determined the altitudes of

the layer boundaries using the hydrostatic relation. Extrapolation outside of the defined model layers is also possible, but will generate an advisory warning.

B.4 Column types

Column types available in *am* are divided here into five categories: line-by-line, collision-induced absorption and continua, gray attenuation, optical refractivity, and composite column types which combine more than one of the preceding types. Conventionally, column names are written in lower case as shown, but upper case (e.g. CH₄) works also. More information can also be found under `column` in the keyword reference above.

B.4.1 Line-by-line column types

The line-by-line column types available in *am* are listed in Table B.4. These column types are for molecular species having line spectra associated with a molecular spectral absorption coefficient $k(\nu)$ [cm²]. The absorption coefficient is computed using data from the HITRAN 2020 database (Gordon et al. 2022), supplemented by other sources as described in Chapter 3: Line-by-line Absorption.

For each column type, a default line shape is given. Default line shapes may be altered by layer and column type with a `lineshape` statement. For collisional line shapes, the line width is the sum of air- and self-induced contributions. The default behavior of *am* when computing pressure-broadened line widths is to ignore small deviations from default dry air mixing ratios as discussed in Section 2.3.1. This behavior can be controlled globally via the `selfbroad_vmr_tol` statement, or controlled for particular species on particular layers with the `strict_self_broadening` flag on a `lineshape` statement, as discussed under `lineshape` in the keyword reference above.

TABLE B.4. LINE-BY-LINE COLUMN TYPES

COLUMN TYPE	DEFAULT MIXING RATIO	DEFAULT LINE SHAPE	MOLECULE	RELATIVE ABUNDANCE
ch4	1.9e-06	Gross	CH ₄	1.0
12ch4	1.87772e-06	Gross	¹² CH ₄	9.88274e-01
13ch4	2.10959e-08	Gross	¹³ CH ₄	1.11031e-02
12ch3d	1.16993e-09	Gross	¹² CH ₃ D	6.15751e-04
ch3cn	0.0	Gross	CH ₃ CN	1.0
12ch3_12c14n	0.0	Gross	¹² CH ₃ ¹² C ¹⁴ N	9.73866e-01
ch3oh	0.0	Gross	CH ₃ OH	1.0
12ch3_16oh	0.0	Gross	¹² CH ₃ ¹⁶ OH	9.85930e-01
co	7.0e-08	Gross	CO	1.0
12c_16o	6.90581e-08	Gross	¹² C ¹⁶ O	9.86544e-01
13c_16o	7.75852e-10	Gross	¹³ C ¹⁶ O	1.10836e-02
12c_18o	1.38475e-10	Gross	¹² C ¹⁸ O	1.97822e-03
12c_17o	2.57507e-11	Gross	¹² C ¹⁷ O	3.67867e-04
13c_18o	1.55575e-12	Gross	¹³ C ¹⁸ O	2.22250e-05
13c_17o	2.89304e-13	Gross	¹³ C ¹⁷ O	4.13292e-06
co2	4.1e-04	Gross	CO ₂	1.0
12c_16o2	4.03524e-04	Gross	¹² C ¹⁶ O ₂	9.84204e-01
13c_16o2	4.53353e-06	Gross	¹³ C ¹⁶ O ₂	1.10574e-02
16o_12c_18o	1.61830e-06	Gross	¹⁶ O ¹² C ¹⁸ O	3.94707e-03
16o_12c_17o	3.00935e-07	Gross	¹⁶ O ¹² C ¹⁷ O	7.33989e-04
16o_13c_18o	1.81813e-08	Gross	¹⁶ O ¹³ C ¹⁸ O	4.43446e-05
16o_13c_17o	3.38095e-09	Gross	¹⁶ O ¹³ C ¹⁷ O	8.24623e-06
12c_18o2	1.62251e-09	Gross	¹² C ¹⁸ O ₂	3.95734e-06
clo	0.0	Gross	ClO	1.0
35cl_16o	0.0	Gross	³⁵ Cl ¹⁶ O	7.55908e-01
37cl_16o	0.0	Gross	³⁷ Cl ¹⁶ O	2.41720e-01
hbr	0.0	Gross	HBr	1.0
h_79br	0.0	Gross	H ⁷⁹ Br	5.06781e-01
h_81br	0.0	Gross	H ⁸¹ Br	4.93063e-01

TABLE B.4. LINE-BY-LINE COLUMN TYPES, CONTINUED

COLUMN TYPE	DEFAULT MIXING RATIO	DEFAULT LINE SHAPE	MOLECULE	RELATIVE ABUNDANCE
hcn	0.0	Gross	HCN	1.0
h_12c_14n	0.0	Gross	H ¹² C ¹⁴ N	9.85114e-01
h_13c_14n	0.0	Gross	H ¹³ C ¹⁴ N	1.10676e-02
h_12c_15n	0.0	Gross	N ¹² C ¹⁵ N	3.62174e-03
h2co	0.0	Gross	H ₂ CO	1.0
h2_12c_16o	0.0	Gross	H ₂ ¹² C ¹⁶ O	9.86237e-01
h2_13c_16o	0.0	Gross	H ₂ ¹³ C ¹⁶ O	1.10802e-02
h2_12c_18o	0.0	Gross	H ₂ ¹² C ¹⁸ O	1.97761e-03
hcl	0.0	Gross	HCl	1.0
h_35cl	0.0	Gross	H ³⁵ Cl	7.57587e-01
h_37cl	0.0	Gross	H ³⁷ Cl	2.42257e-01
hf	0.0	Gross	HF	1.0
h_19f	0.0	Gross	H ¹⁹ F	9.99844e-01
hno3	0.0	Gross	HNO ₃	1.0
h_14n_16o3	0.0	Gross	H ¹⁴ N ¹⁶ O ₃	9.89110e-01
h2o_lines	0.0	Vvh_750	H ₂ O	1.0
h2_16o	0.0	Vvh_750	H ₂ ¹⁶ O	9.97317e-01
h2_18o	0.0	Vvh_750	H ₂ ¹⁸ O	1.99983e-03
h2_17o	0.0	Vvh_750	H ₂ ¹⁷ O	3.71884e-04
hd_16o	0.0	Vvh_750	HD ¹⁶ O	3.10693e-04
hd_18o	0.0	Vvh_750	HD ¹⁸ O	6.23003e-07
hd_17o	0.0	Vvh_750	HD ¹⁷ O	1.15853e-07
h2o2	0.0	Gross	H ₂ O ₂	1.0
h2_16o2	0.0	Gross	H ₂ ¹⁶ O ₂	9.94952e-01
ho2	0.0	Gross	HO ₂	1.0
h_16o2	0.0	Gross	H ¹⁶ O ₂	9.95107e-01
hocl	0.0	Gross	HOCl	1.0
h_16o_35cl	0.0	Gross	H ¹⁶ O ³⁵ Cl	7.55790e-01
h_16o_37cl	0.0	Gross	H ¹⁶ O ³⁷ Cl	2.41683e-01

TABLE B.4. LINE-BY-LINE COLUMN TYPES, CONTINUED

COLUMN TYPE	DEFAULT MIXING RATIO	DEFAULT LINE SHAPE	MOLECULE	RELATIVE ABUNDANCE
h2s	0.0	Gross	H ₂ S	1.0
h2_32s	0.0	Gross	H ₂ ³² S	9.49884e-01
h2_34s	0.0	Gross	H ₂ ³⁴ S	4.21369e-02
h2_33s	0.0	Gross	H ₂ ³³ S	7.49766e-03
nh3	0.0	Gross	NH ₃	1.0
14nh3	0.0	Gross	¹⁴ NH ₃	9.95872e-01
15nh3	0.0	Gross	¹⁵ NH ₃	3.66129e-03
n2o	3.3e-07	Gross	N ₂ O	1.0
14n2_16o	3.26810e-07	Gross	¹⁴ N ₂ ¹⁶ O	9.90333e-01
14n_15n_16o	1.20151e-09	Gross	¹⁴ N ¹⁵ N ¹⁶ O	3.64093e-03
15n_14n_16o	1.20151e-09	Gross	¹⁵ N ¹⁴ N ¹⁶ O	3.64093e-03
14n2_18o	6.55321e-10	Gross	¹⁴ N ₂ ¹⁸ O	1.98582e-03
14n2_17o	1.21862e-10	Gross	¹⁴ N ₂ ¹⁷ O	3.69280e-04
no	0.0	Gross	NO	1.0
14n_16o	0.0	Gross	¹⁴ N ¹⁶ O	9.93974e-01
no2	0.0	Gross	NO ₂	1.0
14n_16o2	0.0	Gross	¹⁴ N ¹⁶ O ₂	9.91616e-01
o	0.0	Voigt-Kielkopf	O (atomic)	1.0
16o	0.0	Voigt-Kielkopf	¹⁶ O (atomic)	9.97628e-01
o2_coupled	0.20946	VVW_coupled	O ₂ , coupled lines	1.0
16o2_coupled	0.208468	VVW_coupled	¹⁶ O ₂ , coupled lines	9.95262e-01
o2_uncoupled	0.20946	Gross	O ₂ , uncoupled lines	1.0
16o2_uncoupled	0.208468	Gross	¹⁶ O ₂ , uncoupled lines	9.95262e-01
16o_18o_uncoupled	8.36041e-04	Gross	¹⁶ O ¹⁸ O, all lines	3.99141e-03
16o_17o_uncoupled	1.55469e-04	Gross	¹⁶ O ¹⁷ O, all lines	7.42235e-04

TABLE B.4. LINE-BY-LINE COLUMN TYPES, CONTINUED

COLUMN TYPE	DEFAULT MIXING RATIO	DEFAULT LINE SHAPE	MOLECULE	RELATIVE ABUNDANCE
o3	0.0	Gross	O ₃	1.0
16o3	0.0	Gross	¹⁶ O ₃	9.92901e-01
16o2_18o	0.0	Gross	¹⁶ O ₂ ¹⁸ O	3.98194e-03
16o_18o_16o	0.0	Gross	¹⁶ O ¹⁸ O ¹⁶ O	1.99097e-03
16o2_17o	0.0	Gross	¹⁶ O ₂ ¹⁷ O	7.40475e-04
16o_17o_16o	0.0	Gross	¹⁶ O ¹⁷ O ¹⁶ O	3.70237e-04
ocs	0.0	Gross	OCS	1.0
16o_12c_32s	0.0	Gross	¹⁶ O ¹² C ³² S	9.37395e-01
16o_12c_34s	0.0	Gross	¹⁶ O ¹² C ³⁴ S	4.15828e-02
16o_13c_32s	0.0	Gross	¹⁶ O ¹³ C ³² S	1.05315e-02
16o_12c_33s	0.0	Gross	¹⁶ O ¹² C ³³ S	7.39908e-03
18o_12c_32s	0.0	Gross	¹⁸ O ¹² C ³² S	1.87967e-03
oh	0.0	Gross	OH	1.0
16oh	0.0	Gross	¹⁶ OH	9.97473e-01
18oh	0.0	Gross	¹⁸ OH	2.00014e-03
16od	0.0	Gross	¹⁶ OD	1.55371e-04
so2	0.0	Gross	SO ₂	1.0
32s_16o2	0.0	Gross	³² S ¹⁶ O ₂	9.45678e-01
34s_16o2	0.0	Gross	³⁴ S ¹⁶ O ₂	4.19503e-02

B.4.2 Collision-induced absorption and H₂O continuum

Collision-induced absorption and H₂O continuum column types, listed in Table B.5, are for processes which give rise to unresolved (continuum) spectra, associated with a binary spectral absorption coefficient $k_b(\nu)$ [cm⁵].

TABLE B.5. CIA AND H₂O CONTINUUM COLUMN TYPES

COLUMN TYPE	DEFAULT MIXING RATIO	DESCRIPTION
h2o_continuum	0.0	An implementation of the H ₂ O continuum from the MT-CKD model (Mlawer et al. 2012) v. 3.5. Included in the composite column type h2o.
h2o_air_continuum	0.0	air-induced component only of h2o_continuum
h2o_self_continuum	0.0	self-induced component only of h2o_continuum
n2n2	0.78084	N ₂ -N ₂ CIA
n2air	0.78084 (N ₂) 0.20946 (O ₂)	N ₂ -N ₂ + N ₂ -O ₂ CIA (N ₂ active) The O ₂ mixing ratio is fixed.
o2o2	0.20946	O ₂ -O ₂ CIA
o2air	0.78084 (N ₂) 0.20946 (O ₂)	O ₂ -O ₂ + O ₂ -N ₂ CIA (O ₂ active) The N ₂ mixing ratio is fixed.

B.4.3 Liquid water and ice absorption

The column types listed in Table B.6 are for modeling absorption by liquid water droplets and ice particles in the Rayleigh limit, meaning that the optical path length across the particle diameter, both internally and externally, is much smaller than a wavelength. The optical constants of water and ice are such that when the Rayleigh limit applies, absorption dominates over scattering, so scattering can be neglected. These models are most suitable for modeling non-precipitating clouds at microwave to millimeter wavelengths.

TABLE B.6. LIQUID WATER AND ICE ABSORPTION

COLUMN TYPE	DEFAULT MIXING RATIO	DESCRIPTION
lwp_abs_Rayleigh	0.0	Absorption by liquid water droplets in the Rayleigh limit, expressed as liquid water path (LWP)
iwp_abs_Rayleigh	0.0	Absorption by water ice particles in the Rayleigh limit, expressed as ice water path (IWP)

B.4.4 Gray attenuation

The `atten` column types, listed in Table B.7, all model frequency-independent (gray) absorption. They differ in their dependence on zenith angle. In a column specification, they take an optical depth or attenuation in `neper` or `dB` as an argument in place of the usual column density. As noted below in Section B.5, Units, *am* uses the radiometric convention of neper as the natural logarithm of a power ratio, consistent with optical depth.

For example, the configuration file line

```
column atten_airmass 0.5 neper
```

sets up a column which will have opacity $\tau=0.5$ towards zenith, increasing with zenith angle in proportion to the airmass factor, which is $\sec(z\alpha)$ in plane-parallel models. The attenuation can be designated as a differentiation or fit variable by adding a second dimensioned argument giving the characteristic scale, as in

```
column atten_airmass 0.1 dB 0.02 dB
```

TABLE B.7. GRAY ATTENUATION

COLUMN TYPE	DESCRIPTION
<code>atten</code>	A fixed loss, independent of zenith angle. Can be used for modeling loss in a receiving system.
<code>atten_airmass</code>	Loss proportional to the layer airmass factor. Models loss through a layer of absorbing material with a frequency-independent absorption coefficient.
<code>atten_sec(za)</code>	Loss proportional to $\sec(z\alpha)$, where $z\alpha$ is the zenith angle θ_z . (Deprecated in favor of <code>atten_airmass</code> , which is equivalent for plane-parallel models.)
<code>atten_sin^2(za)</code>	Loss proportional to $\sin^2(z\alpha)$. For modeling elevation-dependent spillover in an antenna system.
<code>atten_sin^2(2za)</code>	Loss proportional to $\sin^2(2 \cdot z\alpha)$. For modeling elevation-dependent spillover in an antenna system.

B.4.5 Optical refractivity

These column types, listed in Table B.8, are used to incorporate a fixed optical delay into excess delay computations. Normally they are not needed alone, as they are incorporated into composite column types.

TABLE B.8. OPTICAL REFRACTIVITY COLUMN TYPES

COLUMN TYPE	DEFAULT MIXING RATIO	DESCRIPTION
<code>dry_air_optical_refractivity</code>	1.0	Adds optical refractivity for dry air. Included in the composite types <code>dry_air</code> and <code>dry_air_std</code> .
<code>h2o_optical_refractivity</code>	0.0	Adds optical refractivity for H ₂ O. Included in the composite type <code>h2o</code> .

B.4.6 Composite column types

Composite column types, listed in Table B.9, combine multiple column types under a single type name. Any line-by-line column types incorporated therein are still affected by the `lineshape` setting for that column type on a given layer. However, the internal column names are hidden from any global `nscale` factor associated with that column name, as described in Section B.3 above.

TABLE B.9. COMPOSITE COLUMN TYPES

COLUMN TYPE	DEFAULT MIXING RATIO	DESCRIPTION
<code>dry_air</code>	1.0	Combines <code>ch4</code> , <code>co</code> , <code>co2</code> , <code>n2o</code> , <code>o2_coupled</code> , <code>o2_uncoupled</code> , <code>n2air</code> , and <code>o2air</code> , with partial mixing ratios corresponding to the default mixing ratio for each component. Includes <code>dry_air_optical_refractivity</code> for delay computations.
<code>dry_air_std</code>	1.0	Combines <code>co2</code> , <code>o2_coupled</code> , <code>o2_uncoupled</code> , <code>n2air</code> , and <code>o2air</code> , with partial mixing ratios corresponding to the default mixing ratio for each component. Includes <code>dry_air_optical_refractivity</code> for delay computations.
<code>h2o</code>	0.0	Combines <code>h2o_lines</code> , <code>h2o_continuum</code> , and <code>h2o_optical_refractivity</code> .
<code>h2_16o_lines_plus_continuum</code>	0.0	Combines <code>h2_16o</code> , <code>h2o_continuum</code> , and <code>h2o_optical_refractivity</code> .
<code>o2</code>	0.20946	Combines <code>o2_coupled</code> and <code>o2_uncoupled</code>
<code>16o2</code>	0.208468	Combines <code>16o2_coupled</code> and <code>16o2_uncoupled</code>
<code>16o_18o</code>	8.36041e-04	Equivalent to <code>16o_18o_uncoupled</code>
<code>16o_17o</code>	1.55469e-04	Equivalent to <code>16o_17o_uncoupled</code>

B.5 Units

Units recognized by *am* are listed in Table B.10, in the notation used in configuration files. Some symbols, such as `cm-1` and `cm^-1`, are synonyms. In each category of unit, the unit used internally in the program is indicated.

TABLE B.10. UNITS

TEMPERATURE	
K	<i>am</i> internal unit
C	
COLUMN DENSITY	
cm^{-2} , cm^{-2}	<i>am</i> internal unit
m^{-2} , m^{-2}	
cm_pwv , $\text{g} \cdot \text{cm}^{-2}$, $\text{g} \cdot \text{cm}^{-2}$	Precipitable water vapor, IWP, and LWP. 1 cm pwv = $3.3427 \cdot 10^{22} \text{ cm}^{-2}$
mm_pwv , $\text{kg} \cdot \text{m}^{-2}$, $\text{kg} \cdot \text{m}^{-2}$	
um_pwv , $\text{g} \cdot \text{m}^{-2}$, $\text{g} \cdot \text{m}^{-2}$	
dobson, DU	Dobson units (DU), customarily used for O ₃ and SO ₂ . 1 DU = $2.6868 \cdot 10^{16} \text{ cm}^{-2}$, which is equivalent to a 10 μm thick column at STP.
PRESSURE	
mbar, hPa	<i>am</i> internal unit
bar	
Pa	
atm	
Torr	
FREQUENCY	
GHz	<i>am</i> internal unit
Hz	
kHz	
MHz	
THz	
cm^{-1} , cm^{-1}	Wavenumbers. 1 cm^{-1} = 29.9792458 GHz
ANGLE	
rad, radian	<i>am</i> internal unit
deg, degree	
arcmin	
arcsec	

TABLE B.10. UNITS, CONTINUED

SPECTRAL RADIANCE	
watt*cm^-2*GHz^-1*sr^-1, watt*cm^-2*GHz^-1*sr^-1	<i>am</i> internal unit
mW*m^-2*(cm^-1)^-1*sr^-1, mW*m^-2*(cm^-1)^-1*sr^-1	
RU Radiance Units. 1 RU = 1 mW·m ⁻² ·(cm ⁻¹) ⁻¹ ·sr ⁻¹	
watt*m^-2*Hz^-1*sr^-1, watt*m^-2*Hz^-1*sr^-1	
Jy*arcsec^-2, Jy*arcsec^-2	
Jy*sr^-1, Jy*sr^-1	
DISTANCE, EXCESS PATH, AND DELAY	
cm	<i>am</i> internal unit
um	
mm	
m	
km	
ps	time delay = excess path / c
fs	
s	
GRAVITATIONAL ACCELERATION	
cm*s^-2, cm*s^-2	<i>am</i> internal unit
Gal	1 Gal = 1 cm·s ⁻²
m*s^-2, m*s^-2	
GRAVITY GRADIENT	
s^-2, s^-2	<i>am</i> internal unit
Gal*cm^-1, Gal*cm^-1	
mGal*m^-1, mGal*m^-1	
OPTICAL DEPTH (OPACITY) & FREE-SPACE LOSS	
neper*	<i>am</i> internal unit
dB	

* *am* uses the radiometric convention of neper as the natural logarithm of a power ratio, consistent with optical depth. This differs from the amplitude ratio convention used in communications engineering. See the discussion in Annex 1 section 2 of (International Telecommunication Union 2015).

TABLE B.10. UNITS, CONTINUED

MOLECULAR ABSORPTION COEFFICIENT	
cm^2	<i>am</i> internal unit for molecular absorption coefficients
BINARY ABSORPTION COEFFICIENT	
cm^5	<i>am</i> internal unit for binary absorption coefficients
DIMENSIONLESS QUANTITIES	
none	<i>am</i> unit name for dimensionless quantities

References

- M. Bolot, B. Legras, and E.J. Moyer 2013, “Modelling and interpreting the isotopic composition of water vapor in convective updrafts.” *Atmos. Chem. Phys.* 13:7903.
- I.E. Gordon, L.S. Rothman, R.J. Hargreaves, R. Haschemi, E.V. Karlovets, F.M. Skinner, E.K. Conway, C. Hill, R.V. Kochanov, Y. Tan, P. Wcisło, A.A. Finenko, K. Nelson, P.F. Bernath, M. Birk, V. Boudon, A. Campargue, K.V. Chance, A. Coustenis, B.J. Drouin, J.-M. Flaud, R.R. Gamache, J.T. Hodges, D. Jacquemart, E.J. Mlawer, A.V. Nikitin, V.I. Perevalov, M. Rotger, J. Tennyson, G.C. Toon, H. Tran, V.G. Tyuterev, E.M. Adkins, A. Baker, A. Barbe, E. Canè, A.G. Császár, A. Dudaryonok, O. Egorov, A.J. Fleisher, H. Fleurbaey, A. Foltynowicz, T. Furtenbacher, J.J. Harrison, J.-M. Hartmann, V.-M. Horneman, X. Huang, T. Karman, J. Karns, S. Kassi, I. Kleiner, V. Kofman, F. Kwabia-Tchana, N.N. Lavrentievac, T.J. LeeG, D.A. Longp, A.A. Lukashevskaya, O.M. Lyulin, V.Yu. Makhnev, W. Matt, S.T. Massie, M. Melosso, S.N. Mikhailenko, D. Mondelain, H.S.P. Müller, O.V. Naumenko, A. Perrin O.L. Polyansky, E. Raddaoui, P.L. Raston, Z.D. Reed, M. Rey, C. Richard, R. Tóbiás, I. Sadiek, D.W. Schwenke, E. Starikova, K. Sung, F. Tamassia, S.A. Tashkun, J. Vander Auwera, I.A. Vasilenko, A.A. Vigasin, G.L. Villanueva, B. Vispoel, G. Wagner, A. Yachmenev, and S.N. Yurchenko 2022, “The HITRAN2020 molecular spectroscopic database.” *J. Quant. Spectrosc. Radiat. Transfer* 277:107949.
- International Telecommunication Union 2015, “Use of the decibel and the neper in telecommunications.” Recommendation ITU-R V.574-5.
- M. Kakiuchi and S. Matsuo 1979, “Direct measurements of D/H and $^{18}\text{O}/^{16}\text{O}$ fractionation factors between vapor and liquid water in the temperature range from 10 to 40 C.” *Geochem. Journal* 13:307.
- E. J. Mlawer, V. H. Payne, J.-L. Moncet, J. S. Delamere, M. J. Alvarado, and D. C. Tobin 2012, “Development and recent evaluation of the MT_CKD model of continuum absorption.” *Phil. Trans. R. Soc. A* 370:2520.
- D. M. Murphy and T. Koop 2005, “Review of the vapour pressures of ice and supercooled water for atmospheric applications.” *Quarterly J. Royal Meteorol. Soc.* 131:1539.