

Image Rectification and Atmospheric Modeling for EXES

I. Globus-Harris

*Department of Physics, University of California, Davis, 1 Shields Ave. Davis, CA 95616 and
Department of Physics, Reed College, 3203 SE Woodstock Blvd, Portland, OR 97202*

In collaboration with Dr. Matt Richter

We develop a manual image rectification and measurement system for the EXES Echelon-Cross-Echelle Spectrograph on SOFIA, the Stratospheric Observatory for Infrared Astronomy. We then compare reduced medium-mode sky images to spectra produced by a Line-by-Line Radiative Transfer Model (LBLRTM). Results and their use in improving observation time on SOFIA are discussed.

I. INTRODUCTION

When light from astronomical phenomena pass through the earth’s atmosphere, water and other abundant atmospheric molecules absorb and emit photons at certain wavelengths. This leads to a varying transmittance of light and emission of background photons as a function of wavelength. In the infrared, there is a large atmospheric window in ground-based observations between 8 and 14 μm , but light between 5 and 8 μm is almost entirely absorbed by water vapor.

A. SOFIA: Stratospheric Observatory for Infrared Astronomy

SOFIA is a 2.7 m infrared-range telescope inside a modified Boeing 747, built in a collaboration between NASA and the German Aerospace Center (DLR). Its flight capabilities enable it to operate at altitudes up to 13.7 kilometers, above 99.8 percent of the water vapor in the atmosphere. This reduced H_2O column depth allows for clear observations in the infrared range. [1] The difference in transmission of light at SOFIA’s operational altitude is visible in Fig. 4, which compares the transmission of light based on SOFIA’s model atmosphere to that of Maunakea.

B. EXES: Echelon-Cross-Echelle Spectrograph

The Echelon-Cross-Echelle Spectrograph (EXES) enables high spectral resolution observations in this mid infrared range, such as observations of water vapor. As molecules are important tracers of density, temperature, velocity, and chemical profiles of stars and molecular clouds, this creates opportunities to study new aspects of astrophysical phenomena.

As the mid-infrared region is difficult to observe from the ground, and as all existing and planned space telescopes do not observe in this range to EXES’ spectral resolution, EXES is unique in its operation.



FIG. 1: Incident and reflected light on an echelon with blaze angle θ_B .

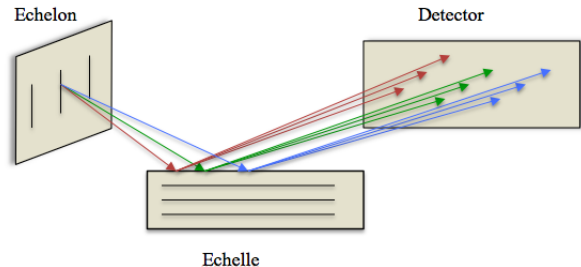


FIG. 2: Light diffracted from an echelon grating is incident on the echelle grating. This diffracts the different orders based on the wavelengths in each order.

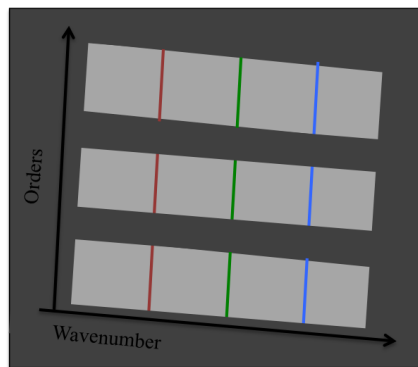


FIG. 3: Incident light on the detector

The EXES spectrograph uses an echelon, or coarse and steeply blazed (angled) diffraction grating, as shown in Fig. 1. The echelon grating’s unique geometry means that a small number of parallel plates can give the same

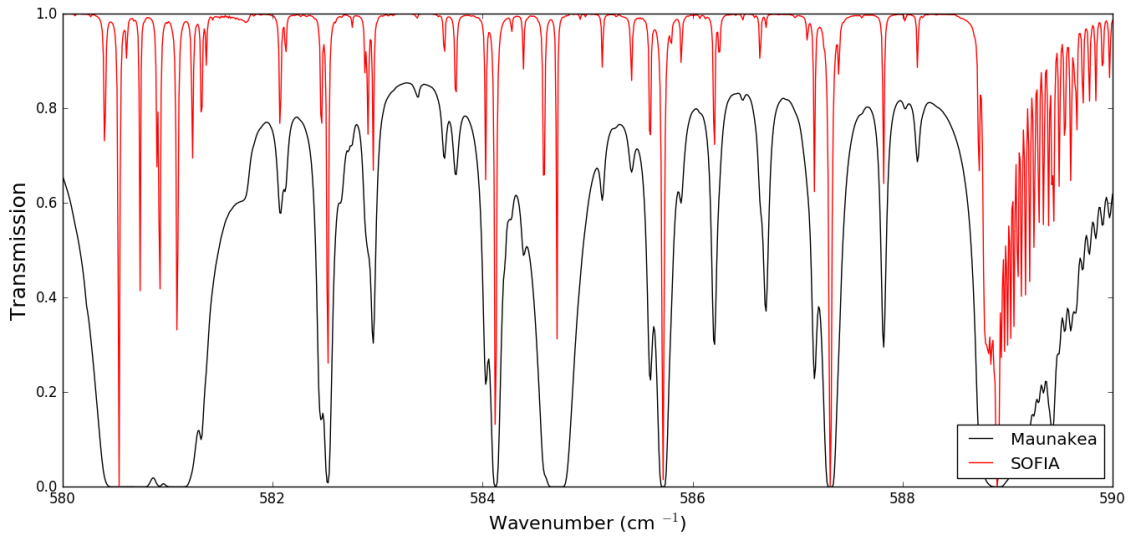


FIG. 4: Model transmission of spectra from Maunakea and SOFIA

high resolution as a much larger traditional diffraction grating. This means that incident beam size can be substantially smaller as well. In spatially constrained observations like those from SOFIA, this is appropriate.

An echelle grating is then used for cross dispersion, as shown in Fig. 2. The echelon’s operation separates the light into its different wavelength components, and then the echelle separates those based on order. The final image recorded by the detector is displayed in Fig. 3. Additionally, the entire system has to be kept in a cryogenic system to prevent the random collisions of the electrons within the system itself that will otherwise saturate the detector.

EXES’ echelle and echelon gratings are used in varying configurations to create four possible modes, as shown in Fig. 5. In high resolution mode, the echelle grating is used to cross disperse the light from the echelon, as described previously. In medium resolution mode, the echelle is used as the primary disperser, and the echelon isn’t used at all. In high-low resolution mode, the echelle grating is rotated so that light is incident on the back of the diffraction grooves. This results in a high number of spectral lines at lower resolution. Additionally, there is a low-resolution mode not pictured. [2]

C. Telluric feature correction

The influence of atmospheric emission and instrumental signature on the observed spectra can be easily removed by nodding the telescope, or viewing the observation object at slightly different angles, then subtracting off the difference.

Since the absorption spectra of atmospheric models can’t be corrected for in this way, the normal procedure

is to observe a star whose features are simple and well understood that is near (in terms of telescope pointing direction) to the desired observing object. As atmospheric conditions, and thus emissions, change over time, this is best done directly before and/or after the observation. Additionally, a blackbody must be observed prior to the other observations, in order to divide out any instrumental noise.

In EXES’ operational wavelengths, only Vega and Sirius are bright enough to use as correcting observations at high resolution. This complicates flight plans, and these extra observations take up to 15 percent of the limited flight time. Use of an atmospheric model to determine atmospheric emission, combined with a profile of instrumental gains, removes these steps from the observation process, and increases the amount of usable flight time. Given SOFIA’s high operation cost, this would be especially impactful.

II. PROJECT DESIGN

In order to analyze the effectiveness of EXES’ current image reduction pipeline, a manual reduction process was developed. EXES’ medium mode images of the sky with no stellar objects were reduced by this method, and wavenumber vs brightness plots were produced in a well-understood wavenumber range. Next, a Line-by-Line Radiative Transfer Model (LBLRTM) was used to produce a model of the sky in the same wavenumber range. Functionality of the model was verified using example input parameters. Then, a model of the sky at SOFIA altitude and its associated wavenumber vs brightness are produced, and compared with the manually reduced data. Results are discussed.

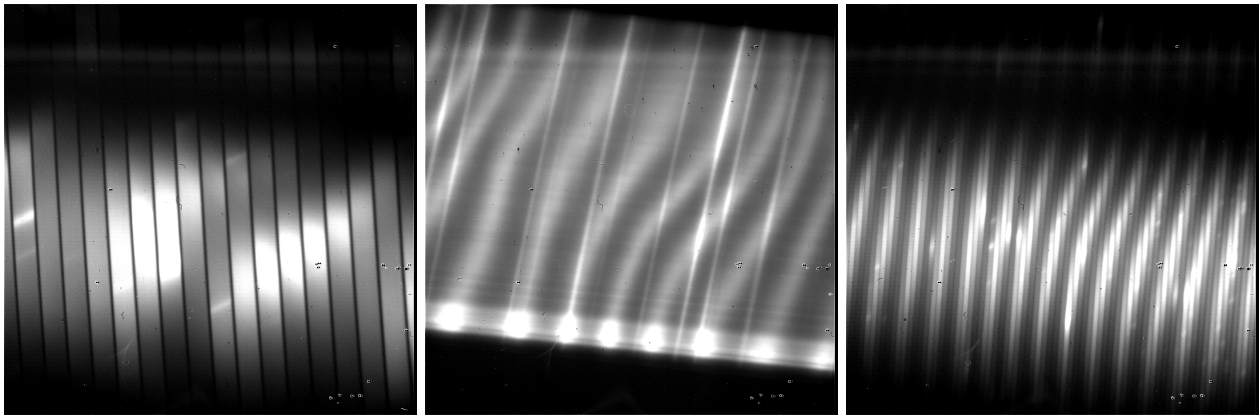


FIG. 5: High, medium, and low resolution images of the sky

III. LBLRTM

As described in detail in appendix B, the LBLRTM uses an extensive line parameter database to model the molecular lines in our atmosphere. This line parameter database is fed into a line file creation program, and is then inputted along with any desired parameters to the main modeling code. The line parameter database used is much more complete than EXES' current model, and additionally, input parameters are much more versatile. [3]

IV. DATA REDUCTION

Observations from EXES are stored as two dimensional arrays, with a pixel value associated with each item in the array. However, the true observations across the diffraction slit are slightly rotated within the array, as seen in Fig. 5. In order to account for this instrumental effect, the medium-mode image used was first rotated manually. Then, a single row was extracted. Appendix A describes the graphical user interface designed for this manual image reduction, as well as additional measurements, in detail.

The LBLRTM outputs transmission values per wavenumber, but the values recorded by EXES are in terms of emission, in terms of recorded pixel value. In order to compare the two, the pixel values of the rotated image were subtracted from and divided by the pixel values of their corresponding blackbody.

V. RESULTS

As shown in Fig. 6 and 7, there is a significant correlation between the LBLRTM and the currently used atmospheric models in the wavenumber range of 580 to 590 cm^{-1} . This is expected, as the wavelength range chosen is known to be fairly complete in the current models. In

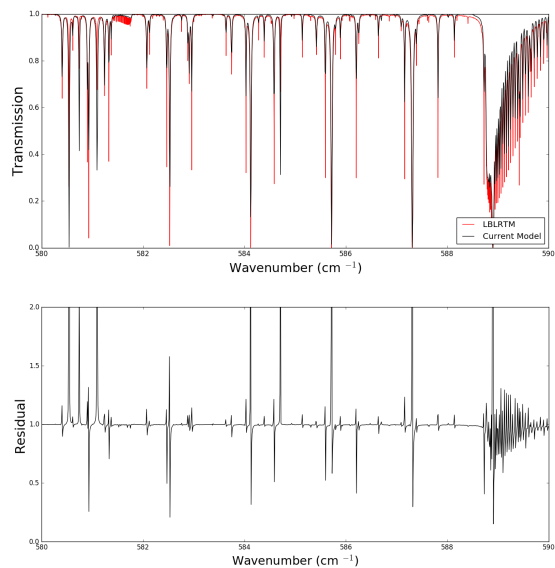


FIG. 6: Comparison of LBLRTM of transmission at SOFIA flight altitude and current model's transmission for wavenumbers in 580 to 590 cm^{-1} range.

the 589 cm^{-1} CO_2 band, the LBLRTM model displays a finer structure than the current model. Fig. 8 compares the LBLRTM output at SOFIA altitude to EXES observations. In general, there is a good correlation between the two.

VI. CONCLUSION

These results show that the LBLRTM is a viable atmospheric model, and matches the current models in well understood regions. Extensions of this work include use of the LBLRTM in wavenumber regions where the current model atmosphere is known to be insufficient. If the LBLRTM parameters could be altered to emission plots instead of transmission, this would ease the comparison

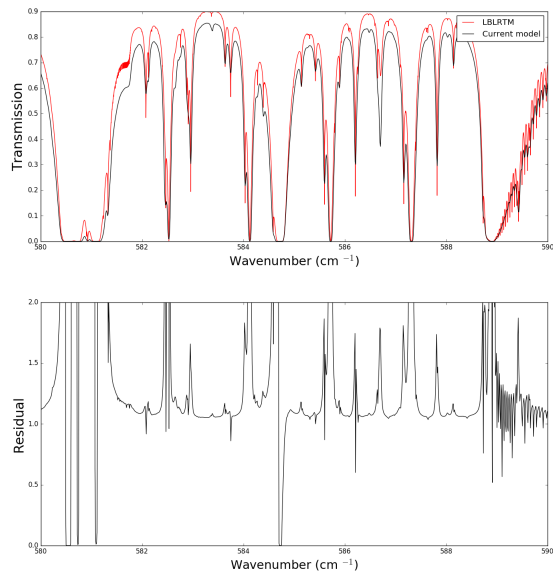


FIG. 7: Comparison of LBLRTM of transmission at Maunakea altitude and current model's transmission for wavenumbers in 580 to 590 cm^{-1} range.

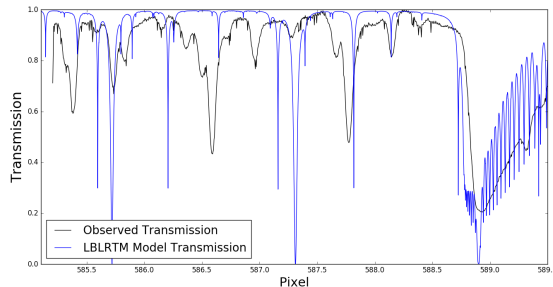


FIG. 8: Comparison of LBLRTM of transmission at SOFIA altitude and with observed transmission for wavenumbers in 580 to 590 cm^{-1} range.

process. Additionally, an interface could be created for the LBLRTM input tapes to ease the readability of user inputs.

Ultimately, the LBLRTM model and a model of instrumentation gains could be fully integrated into the current data reduction pipeline. This would fully automate the image correction process, which in turn would increase usable flight time by up to 15 percent and significantly save on SOFIA's operation cost and the ease of flight planning.

VII. ACKNOWLEDGMENTS

This research was funded by NSF Award 1263201. Much thanks to Dr. Matt Richter for advising, and to Prof. Zieve for organization of the UC Davis REU.

VIII. BIBLIOGRAPHY

- [1] Young et al., *Early Science with SOFIA, the Stratospheric Observatory for Infrared Astronomy* (2012).
- [2] Richter et al., *EXES: A Progress Report on the Development of a High-Resolution, Mid-Infrared Grating Spectrograph for SOFIA* (2000).
- [3] Clough SA, et al., *Atmospheric Radiative Transfer Modeling: a Summary of the AER Codes*, short communication, *J Quant. Spectroscopy and Radiative Transfer*, 91, 233-244 (2005).

Appendix A: Manual Image Rectification GUI User Guide

1. Overview

This Graphical User Interface (GUI) is designed for .fits files from the EXES Echelon-Cross-Echelle Spectrograph. Images from .fits files are displayed in a tkinter window, where the user can interactively rectify the images and measure line widths.

2. Installation Guide

First, download the following files from ____.

- fitsgui.py
- README.txt
- INSTALL.txt

In order to run the GUI, you will need to have python 2.7 or later installed on your computer. In addition to this, you will need the following python libraries:

- os
- tkinter
- numpy
- matplotlib
- astropy.io

Installing these is easiest to do with pip, Python’s package installation manager, which is available for download online. After you have installed pip, run the following command in your terminal window: `pip install <name of library needed>`, then proceed through any additional steps as dictated by the terminal window.

3. Running the GUI

1. In a terminal window, navigate to the directory that your .fits files are in.
2. Run the following command: `python fitsgui.py` (Or, if you have a different command name for python, use that instead of python, i.e python3)
3. A tkinter GUI window will pop up and display the first .fits image in the folder
4. To exit, press the “Quit” button at the bottom of the screen.

5. If you click the standard exit button instead, it is likely that your terminal will throw an error and you will need to manually quit. In the terminal window, type `control-z` to do so. Then make sure that the tkinter python application isn’t still running in the background by checking your task manager and force-quitting it there.

4. Instructions for Use

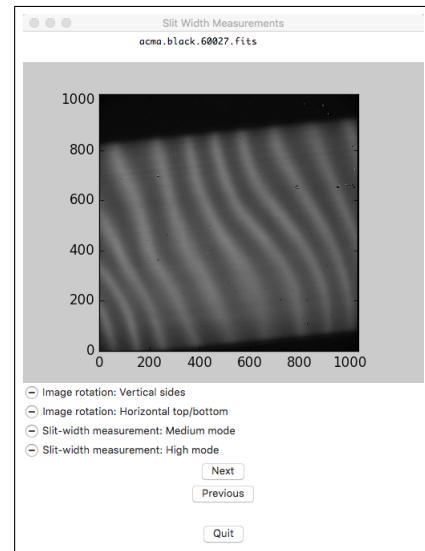


FIG. 9: The GUI window as it displays .fits file acma.black.60027.fits. Note the mode selection buttons at the base of the window, as well as navigation and quit buttons. In this particular image, the horizontal axis indicates wavenumber and the vertical axis is spatial.

a. Navigation

“Next” and “Previous” buttons are used to click through all fits files in the directory that the code is run in. The name of the fits file currently in use is displayed at the top of the GUI window.

b. Mode selection

There are three possible modes: image rotation, slit-width measurement for medium mode, and slit-width measurement for high mode. To select one, click it in the GUI window. A text box will appear at the bottom of the screen to confirm your selection.

a. *FWHM modes* The FWHM mode is designed to measure the Full-Width-Half-Maximum (FWHM) slit width of lines on a .fits image. When the .fits files are in EXES’ MEDIUM mode, these lines are vertical (thus the

width will be horizontal). When they are in HIGH_MED and HIGH_LOW, the lines are horizontal (so the width will be vertical).

The user draws lines to pinpoint where to run the calculation on the image, then the Full-Width-Half-Maximum (FWHM) of the given slit width is calculated and written to an output file, slitwidth.txt.

b. Image rotation mode Image rotation mode is designed for .fits images that need to be rotated by some angle due to instrumentation effects. It is essentially a linear transformation of a parallelogram with vertical sides and a slanted top and bottom within the rectangular image that transforms that parallelogram to a rectangle, and ignores the points outside of the parallelogram.

The user draws a line across the top edge of the parallelogram. The program then extrapolates the other vertices of the parallelogram, then writes the data to a new file, with the same name as the input file but with an `_rot` at the end.

c. Output files When in rotation mode, a rotated .fits file with the same name as the initial file, and `_rot` appended at the end of it, will be outputted to the folder. Additionally, a text file, rotation.txt, with information about the rotation will be outputted. When in slit-width mode, a text file slitwidth.txt will be outputted, with information about the slit width measured.

Appendix B: Line-by-Line Radiative Transfer Model (LBLRTM) User Guide

The Line-by-Line Radiative Transfer Model (LBLRTM) outputs files based on a line parameter database (HITRAN). First, this line parameter database is linked to a line file creation program (LNFL). The resulting output, TAPE3, is inputted into the LBLRTM. Input and output files to the LBLRTM are all named TAPE files. This is shown in Fig. 13.

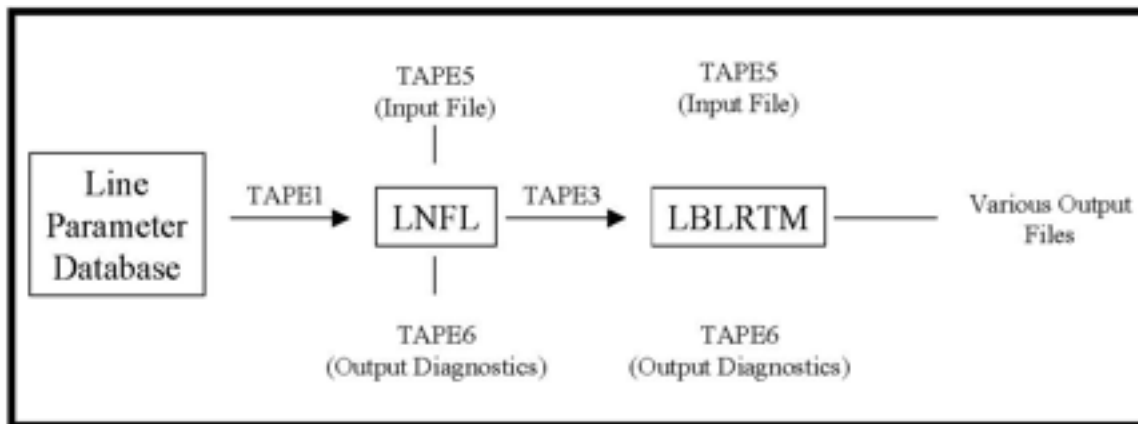


FIG. 10: Structure of the LBLRTM program, from FAQ_LBLRTM.pdf. The line parameter database is linked to TAPE1, which in turn is used along with an input file (TAPE5) to create a line file (TAPE3). This, along with input parameters specified in TAPE5, are inputs to the LBLRTM, which subsequently outputs a variety of files dependent on input parameters.

1. Installation Guide

1. Go to http://rtweb.aer.com/lblrtm_code.html, and fill out the information to download `aer_lblrtm_v12.4_lnf1_v3.0.tar.gz` and `aer_v_3.4`. The first of these `.tar.gz` files contains both the LBLRTM code (in the `lblrtm` folder) and the LNFL creation code (in the `aerlnf1_v3.0` folder). The second, `aer_v_3.4`, is the line parameter database.
2. After filling out your contact information, the developers will email you links to the proper files. The download links will expire after two hours.
3. Move the files to whatever folder that you want to work in (i.e., create an LBLRTM folder on your computer).
4. Unzip the files. If you have a mac, you should unzip them manually with `gunzip FILE.tar.gz`; `tar -xvf FILE.tar`. After doing so, you should have the following subdirectories in your LBLRTM folder: `aer_lblrtm_v12.4_lnf1_v3.0`, and `aer_v_3.4`.
5. Navigate to the `aer_lblrtm_v12.4_lnf1_v3.0` folder. There should be two `tar.gz` files in it: `aer_lblrtm_v12.4.tar.gz` and `aerlnf1_v3.0.tar.gz`. Unzip them with `FILE.tar.gz`; `tar -xvf FILE.tar`. After unzipping them, you should have the following subdirectories in the `aer_lblrtm_v12.4_lnf1_v3.0` folder: `aerlnf1_v3.0` and `lblrtm`.
6. Navigate to `aerlnf1_v3.0/build`.
7. There is a `README` with instructions in the `build` subdirectory. Note that the file extension is `README.build_instructions`, so your computer may try to read it as a file that has type `".build_instructions"`. If so, rename it with a `.txt` at the end, and ignore your computer's warnings while doing so, or view it in the terminal with `view` or `cat` commands.
8. As per the `README` instructions, navigate in the terminal to the `build` directory, then type `gmake -f make_lnf1 <TARGET>`. If you have a mac, use `make` not `gmake`; they are the same thing. The target I used was `osxGNUsg1`.
9. Navigate back to `aer_lblrtm_v12.4_lnf1_v3.0` and go to `lblrtm/build`. As per the build instructions there, navigate in the terminal to that `build` directory and type `gmake -f make_lblrtm <TARGET>`, or use `make` instead of `gmake` if you have a mac. I used the target `osxGNUsg1` again.

2. Directions for LNFL Creation

1. In the LNFL folder, navigate to the `run_examples` directory. The only subfolder there should be `run_example_infrared`. Navigate to that folder in your terminal.
2. The folder will have several files in it: `aer_v3.4`, `wv_co2_brd_param`, etc. These are all aliases that linked to specific files in the line file database. However, all of the links are broken, so the code won't run.
3. Link up the files correctly:
 - (a) In another terminal or folder navigation window, navigate to the line file database, `aer_v3.4`. There will be several subfolders (`extra_brd_params`, `line_file`, `line_files_By_Molecule`, `spd_dep`, `spectral_lines_for_MonoRTM_xs_files_v3.4`).
 - (b) Navigate to the `line_file` subfolder. There will be two files, `aer_v3.4.4`, and `lncpl_lines`.
 - (c) As shown in the flowchart on the first page, the line parameter database should be linked to TAPE1. Do this by navigating in terminal to the `run_example_infrared` directory of the line file folder `aerlnfl_v3.0`. Then type `ln <the path to aer_v3.4.4 in the line parameter database aer_v3.4> TAPE1`. You can get the path to `aer_v3.4.4` on macs by just dragging the file from your finder window to the terminal window, which is a handy trick.
 - (d) After you have entered this, a new file called TAPE1 should show up in the `run_example_infrared` folder.
 - (e) If the alias links to the other line parameter files in the `run_example_infrared` folder are broken like they were for me, you additionally need to fix all of those. Remove the old broken aliases, and look in the different folders in the line parameter database `aer_v3.4` to find the originals
 - i. In `extra_brd_params`, there are the files `co2_co2_brd_param`, `co2_h2o_brd_param`, `o2_h2o_brd_param`, and `wv_co2_brd_param`.
 - ii. In `spd_dep` is the file `spd_dep_param`
4. If you have a TAPE3 file in the `run_example_infrared` directory, remove it.
5. Run the lnfl creation program! There is a misleading executable named `run_lnfl` in the `run_example_infrared` folder. However, all this does is it removes existing TAPE files from if you've already run the line creation code, and tries to link in TAPE1 and link to the executable line file creation program. On my computer, this doesn't work, because it assumes that the executable you have is `lnfl_v3.0_linux_pgi_sgl`. Since I have a mac, that isn't the case. So don't try to run `run_lnfl`. It will spit errors about how you don't have any tape files to remove (you don't) and how it couldn't find the right things to link to (if your operating system is different).
6. The executable that you actually want to use is the executable a few file folders back, in `aerlnfl_v3.0`. On my computer, it's `lnfl_v3.0_OS_X_gnu_sgl`. The name will vary depending on how you did the build and your operating system.
7. However, the executable still has to actually be run in the `run_example_infrared` folder. So, navigate to that folder in terminal if you aren't there already.
8. To run the executable there, you just need to know the path to that file, type it in the terminal window and press enter. I find it easiest to just navigate to `aerlnfl_v3.0` in my finder window, then drag and drop the executable (`lnfl_v3.0_OS_X_gnu_sgl` in my case) from the finder window to the terminal window, and press enter.
9. If successful, output TAPE files will be created in the working directory.

3. Directions for Running the LBLRTM

To confirm the functionality of the LBLRTM, it is simplest to use one of the example TAPE5 input files that was provided, e.g. the atmospheric upwelling example. First, navigate to `lblrtm/run_examples/run_example_built_in_atm_upwelling`. Next, link TAPE3 to `run_examples/TAPE3_files/TAPE3_aer_v3.4_ex_little_endian`. Then, run the executable (e.g. `lblrtm_v12.4_OS_X_gnu_sgl`) in the example directory. Note that once again, there is a misleading script in the example directory, `script_run_example.pl`. However, at least in my directories, the links that it specifies do not work properly, so it is easiest to run manually instead. The example code contains sample outputs (`TAPE6_ex`, `TAPE27_ex`, and `TAPE28_ex`), so you can compare your output files to these to ensure functionality.

4. Summary of Documentation

- For general set-up, refer to the readme files in the different build subdirectories, as well as the FAQ_LBLRTM.pdf file in ../lblrtm/docs.
- For the TAPE5 input format in the LNFL creation program, refer to ../aerlnfl_v3.0/docs/lnfl_instructions.
- For the TAPE5 input format in the LBLRTM, refer to ../lblrtm/docs/html/lblrtm_instructions_frame.html

5. Input File TAPE5 and How to Read FORTRAN-Style Documentation

```

Record 1.1 -----> $ Simple TAPE5
Record 1.2 -----> HI=1 F4=1 CN=1 AE=0 EM=1 SC=0 FI=0 PL=0 TS=0 AM=1 MG=0 LA=0 OD=0 XS=0 00 00
Record 1.3 -----> 1000.0000 1200.0000 0.0 0.0 0.0 0.0000 0.000 0 0.0
Record 1.4 -----> 294.2000 1.0000
Record 3.1 -----> 6 2 0 0 0
Record 3.2 -----> 100.000 0.000 180.000

Record 1.1 -----> -1.
Record 1.2 -----> $ Transfer to ASCII plotting data
Record 12.1 -----> HI=0 F4=0 CN=0 AE=0 EM=0 SC=0 FI=0 PL=1 TS=0 AM=0 MG=0 LA=0 MS=0 XS=0 0 0
Record 12.2 -----> # Plot title not used
Record 12.2A -----> 1000.0000 1200.0000 10.2000 100.0000 5 0 12 0 1.000 0 0 0
Record 12.3A -----> 0.0000 1.2000 7.0200 0.2000 4 0 1 1 0 0 0 3 27
Record 12.3A -----> 1000.0000 1200.0000 10.2000 100.0000 5 0 12 0 1.000 0 0 0
Record 12.3A -----> 0.0000 1.2000 7.0200 0.2000 4 0 1 1 1 0 0 3 28

EOF -----> -1.
EOF -----> % US STD Atmosphere: Nadir viewing from 100 km down to the surface
  
```

FIG. 11: Sample TAPE5

The documentation for the input files requires a level of familiarization with fortran style. Since fortran is an old language, conciseness is prioritized in parameter entry. This makes efficient, but difficult to read, input. The documentation is thorough, but only if you know what any of it means.

The way the LBLRTM parses the input file is based on the values of inputs on each line and column of the input file. This means that every input variable corresponds to a specific line and column location of the input file. In the documentation, each new "record" value corresponds roughly to a new line. However, some records are only included if certain input parameters are set.

RECORD RECORD 1.3 (required if IHIRAC > 0; IAERSL > 0; IEMIT = 1; IATM = 1; or ILAS > 0; otherwise omit)

a	V1,	V2,	SAMPLE,	DVSET,	ALFAL0,	AVMASS,	DPTMIN,	DPTFAC,	ILNFLG,	DVOUT,	NMOL_SCAL
b	1-10,	11-20,	21-30,	31-40,	41-50,	51-60,	61-70,	71-80,	85,	90-100,	105
c	E10.3,	E10.3,	E10.3,	E10.3,	E10.3,	E10.3,	E10.3,	E10.3,	4X,I1,	5X,E10.3,	3x,I2

V1 beginning wavenumber value for the calculation
(VLAS = V1 for ILAS = 1,2)

V2 ending wavenumber value for the calculation
((V2-V1) must be less than 2020 cm - 1)

FIG. 12: Sample documentation from lblrtm_instructions.html

For example, if the LBLATM (the built in atmospheric model) is set to "1" in line 2 of the file, then all of record 2 (a user-specified atmosphere) isn't used, so those lines of code won't exist in that particular TAPE5 file. In this case, when the code runs, it will jump from parsing line record 1 to line record 3. Unfortunately, this means that you

have to work closely with the documentation and run through every record line, to make sure that you are getting all the records that you need. After the first few lines, arbitrary record "N.n" will no longer correspond to line "n". Fig. 11, a sample TAPE5 file with annotated lines, displays this phenomenon.

In general, the documentation for each record matches the format of Fig. 12. The first line lists the record discussed, and when it is applicable. If it is not applicable, omit it. The next three rows (**a** through **c**) describe the input variables associated with the line. **Row a** is the name of the variable, **row b** is its placement in terms of column number, and **row c** describes the data type that can be inputted.

For example, from Fig. 12, V2 is the ending wavenumber, and is located in columns 11 through 20 of Record 1.3. Its data type is E10.3. This means that it is in exponential form (E), has a width of 10, and has 3 values after the decimal point. In Fig. 11, this is the value highlighted in the second red box, 1200.0000. Admittedly, this number has 4 values after the decimal point, not 3, which just goes to show that the documentation isn't perfect. Since they are all 0's it doesn't matter though. V2 is 9 characters not 10 because there is an extra space in front of the first number which is read as a 0, i.e. "01200.0000."

Other values have slightly different format editor descriptors. Fig. 13, which describes record 1.2 (as shown in Fig. 11), is a good example of this. Again, **row a** (including the portion that has wrapped around to a second line) shows the name of the variable. As a general rule, they begin with "I" to indicate that they are integer valued. Here, the documentation doesn't indicate a range of position values in **row b** for each variable, but rather a single column. This is because the only column that is read for the integer is that specific column. The "4X.I1" format listed in **row c** means that each value is "4 columns of whatever" then the single integer. In Fig. 11's TAPE5, those extra 4 spaces are used to make the value of the variable more clear. For example, "HI=1" instead of "___1." The program won't read the extra characters, but they help make the input a little more readable.

```
RECORD RECORD 1.2

aIHIRAC, ILBLF4, ICNTNM, IAERSL, IEMIT, ISCAN, IFILTR, IPLIT, ITEST, IATM, IMRG, ILAS, IOD, IXSECT, MPTS,
  NPTS, ISOTPL, IBRD
b  5,   10,   15,   20,   25,   30,   35,   40,   45,   50, 54-55,   60,   65,   70, 72-75, 77-80,   85
  90
c  4X,I1, 4X,I1, 4X,I1, 4X,I1, 4X,I1, 4X,I1, 4X,I1, 4X,I1, 4X,I1, 4X,I1, 3X,A2, 4X,I1, 4X,I1, 4X,I1, 1X,I4, 1X,I4,
  4X,I1,4X,I1
```

FIG. 13: Sample documentation from lblrtm_instructions.html

6. Explanation of Output Files

The main LBLRTM program outputs many files, but most are in binary format so not human readable. TAPE27 and TAPE28, however, are conversions to ASCII. They, and a theoretically unlimited number of other files, may be created by having the LBLRTM run twice in TAPE5: first to produce all the desired results, then a second time with the plotting flag set to 1, i.e. PL=1 in the second half of Fig. 11. Then, depending on the following lines, a variety of output files that can be used to generate plots are created as per record 12 documentation.

As vaguely described in the record 1.1 documentation, the program knows to start back over at record 1.1 due to the "-1" on the previous line. If the following line's first character is a "\$", then it will proceed to run the program again. If it encounters a "%" instead, as in the final line of Fig. 11, this indicates to the program that it should stop running.