

Guide to Software for Physics of the Tropical Atmosphere and Tropical Cyclones

K. Emanuel, January 2025

Here is a guide to the scripts and programs that support the textbook *Physics of the Tropical Atmosphere and Tropical Cyclones*. All of the scripts were written for MATLAB but all are compatible with recent releases of GNU Octave, freely available software that is quite similar (but not identical) to MATLAB. A passing familiarity with MATLAB/Octave will be helpful, but you should not need a thorough grounding. (Some scripts, such as *time_dependent_model*, are considerably slower in Octave than in MATLAB.) Three packages: the single-column model, the Hadley model, and the hurricane model require one-time compiles of FORTRAN code, but you do not need to know FORTRAN to use any of these models.

Preparation

Unzip this package of software into a clean directory on your machine. You should see a collection of .m files: these are scripts that can be run in MATLAB or Octave, and several folders. In most cases, you can just run these scripts in MATLAB or Octave. If you do not have MATLAB and cannot get access to it, you should install the GNU Octave package, freely available online, on your machine. (Many Linux installations at universities routinely have Octave installed.) Both MATLAB and Octave have versions that run on PCs, MACs, and Linux machines.

There are three tasks you need to do, once and for all, on each machine you plan to run these scripts on. All of these require that you have a FORTRAN compiler installed. (Most university-based installations already have one or more FORTRAN compilers installed.) If you do not have one already, there are several freely available compilers that you can install on your machine; for example, GNU FORTRAN. Follow the compiler instructions to install the compiler appropriate for your machine.

Now navigate to the SCM folder, which contains a FORTRAN code named *scm.f*. Compile that program into an executable called *scm.exe*. For example, with GNU FORTRAN, you would execute a system command

```
gfortran -o scm.exe scm.f
```

Check that the file *scm.exe* is now in the directory. You can also issue this command from within MATLAB or Octave, e.g. `status=system('gfortran -o scm.exe scm.f');`

Next, navigate to the Hadley folder and do the same for the FORTRAN program *Hadley.f*:

```
gfortran -o Hadley.exe Hadley.f
```

Finally, navigate to the Hurricane folder and issue a command like

```
gfortran -o hurricane.exe hurricane.f
```

Note that the hurricane program may take tens of minutes or hours to run. In this case, it can be quite helpful to use an advanced FORTRAN compiler, like NVIDIA FORTRAN, with optimizations. For example,

```
pgf95 -fast -o hurricane.exe hurricane.f
```

In each case, it is important that the executable files be named *scm.exe*, *Hadley.exe*, and *hurricane.exe*, respectively. (The case of the letters is important here.)

That's it! You should now be all set to run all of the programs in this package. The Hadley, single-column model (SCM), and hurricane programs need to be run from within their respective folders; all the others can be run from within the main directory.

Scripts

The following describes the various scripts in the order in which they are found in the textbook.

1. **Single-column model:** This model and how to use it are described in some detail in the document *Single_Column_Model_Guide.pdf* in the SCM folder. To use this model, you must have compiled the FORTRAN program *scm.f* following the instructions stated in the previous section *Preparation*. Very briefly: First navigate to the SCM folder. Use any editor to open the ascii file *scm.in*. Edit the first column of the file as you see fit, but preserve the formatting of the file (e.g. do not delete blank lines, etc.) Then, inside MATLAB or Octave, type *run_scm*. You will be asked to give a name to the output file. If you specified the simulations to begin with fields from a previous integration, you will be asked to provide the name of that folder. Then you should see in the command window a count of the number of days completed. At the end of the simulations, a menu will appear with choices for graphs of the output. You may use the script *scm_plot.m* to plot any previous output you have saved.
2. **soundplot.m:** This script fetches a user-requested radiosonde sounding from a University of Wyoming website, plots it on a skew T-log p chart, and allows the user to lift parcels from various levels by various processes, which will also be plotted on the chart. Figure 3.10 in the textbook is an example of the graphical output. The script and its use are described extensively in its header; open it in the MATLAB or Octave editor.
3. **buoycon.m:** Like *soundplot.m*, this retrieves a user-requested sounding from a University of Wyoming website, but here plots the difference between the density temperature of a lifted parcel and that of its environment, as a function of the level from which the parcel is lifted and the level to which it is lifted. It also plots the integrated positive buoyancy (CAPE) and the convective inhibition (CIN) as a function of the level from which the parcel is lifted. An example of this output is shown in Figure 3.11 in the textbook. The script and its use are described extensively in its header; open it in the MATLAB or Octave editor.

4. **thetabplot.m**: Like *soundplot.m* and *buoycon.m*, but plots the buoyancy potential temperature as a function of altitude and user-chosen parcel origin levels. This was used to produce Figure 3.12 in the textbook. The script and its use are described extensively in its header; open it in the MATLAB or Octave editor.
5. **aggplot.m**. This script calculates and plots solutions to the self-aggregation equations expressed by equations (3.94) and (3.95) in the text. An example is shown in Figure 3.40 in the text. Parameter values are specified near the top of the script and can be changed using the MATLAB or Octave editor.
6. **Hadley**: This FORTRAN program runs a two-dimensional (latitude-altitude) model with the same convective and radiative physics as used in the single-column model. It is described extensively in the documents *Hadley_Model_Guide.pdf* in the Hadley folder. To use this model, you must have compiled the FORTRAN program *Hadley.f* following the instructions stated in the previous section *Preparation*. Very briefly: First navigate to the Hadley folder. Use any editor to open the ascii file *scm.in*. Edit the first column of the file as you see fit, but preserve the formatting of the file (e.g. do not delete blank lines, etc.) Then, inside MATLAB or Octave, type *run_Hadley*. You will be asked to give a name to the output file. If you specified the simulations to begin with fields from a previous integration, you will be asked to provide the name of that folder. Then you should see in the command window a count of the number of days completed. At the end of the simulations, a menu will appear with choices for graphs of the output. You may use the script *Hadley_plot.m* to plot any previous output you have saved.
7. **Walker.m**: This script solves the simple model of the Walker circulation presented in Chapter 5. Examples of the solutions are shown in Figures 5.54 and 5.55 in the textbook. Parameters may be set near the top of the script; open in MATLAB or Octave editor.
8. **Matsuno.m**: This script plots the nondimensional dispersion curves for Matsuno waves as well as the eigenfunctions of modes as described in Chapter 6 of the textbook. This script was used to create Figures 6.8 and 6.9 in the textbook. Parameter values may be set near the beginning of the script; open in MATLAB or Octave editor.
9. **Gill.m**: This script solves the steady state equations for the equatorial atmospheric response to prescribed sea surface saturation entropy anomalies, in the case of vanishing momentum drag, as described in Chapter 6 of the textbook. Three maps are produced. This was used to generate Figure 6.10 in the textbook. Parameter values may be set near the beginning of the script; open in MATLAB or Octave editor.
10. **GillD.m**: This script solves time-dependent equations for the equatorial atmospheric response to prescribed sea surface saturation entropy anomalies, as a way of estimating the steady state response with surface drag, as described in Chapter 6 of the textbook. If the combinations of parameters is linearly unstable, the program states this and stops. This was used to generate Figure 6.11 in the textbook. Parameter values may be set near the beginning of the script; open in MATLAB or Octave editor. Note that this script can be quite slow in Octave.

11. **equamodes.m**: Program for finding growth rates and frequencies of equatorial modes driven by cloud-radiation and WISHE, as solutions of (7.16) in Chapter 7 of the textbook. Also finds modes with geostrophic and WTG approximations. This script calculates modes but only plots eigenvalues, not eigenfunctions. Use *equaplot.m* to plot both. These scripts were used to generate Figures 7.26 – 7.30 in the textbook. Parameter values may be set near the beginning of the script; open in MATLAB or Octave editor.
12. **equaplot.m**: This script plots eigenvalues and eigenfunctions from the most recent run of the script *equamodes.m*. Note that this script only plots eigenfunctions of the complete equation set or the geostrophic modes, not modes under the WTG approximation. Only the largest growth rates for each mode number and zonal wavenumber are plotted.
13. **steady_state_model.m**: Numerically solves the set of equations developed by Emanuel and Rotunno (2011) for the steady-state structure and amplitude of tropical cyclones whose outflow is self-stratifying, as discussed in Chapter 9 of the textbook. It uses a mindless iteration to find that value of the radius of the outflow point, r_t , that causes the outflow temperature to attain its ambient tropopause value at the radius of maximum winds. This script was used to create Figures 9.9 and 9.10 in the textbook. Parameter values may be set near the beginning of the script; open in MATLAB or Octave editor.
14. **PBLSlabModel.m**: Integrates the nondimensional PBL equations in time until a steady state is achieved, given a gradient wind profile, as described in Chapter 9 of the textbook. Plots all three velocity components at discrete intervals in time. Please see the document *Axisymmetric Slab Boundary layer Model.pdf* for a complete description, and note that the script includes horizontal mixing which is absent from the model described in the textbook. Parameter values may be set near the beginning of the script; open in MATLAB or Octave editor. Note that this script takes considerably longer to run in Octave than in MATLAB.
15. **time-dependent-model.m**: Integrates the simple time-dependent TC model described in Chapter 10 of the textbook. Menus presenting plot choices appear after the script runs. Used to create Figures 10.2-10.6 in the textbook. Parameter values may be set near the beginning of the script; open in MATLAB or Octave editor. Note that this script takes much longer to run in Octave than in MATLAB.
16. **Hurricane**: This set of scripts runs and displays the output of the axisymmetric, non-hydrostatic hurricane model of Rotunno and Emanuel (1987). Navigate to the Hurricane folder and read the document *Hurricane_Model_Guide.pdf* for detailed instructions on how to use this model. Note that this is the most complex and sophisticated code in this package, and profits from being compiled using an advanced FORTRAN compiler with optimizations. It may take an hour to run on a standard desktop and for this reason it should mostly be run in background.