Ben Ghertner November 20, 2025

This document is intended to be a step-by-step guide to how I setup and use CM1 (https://www2.mmm.ucar.edu/people/bryan/cm1/) and give a brief description of the background and initialization routines I have added.

Download and Run CM1

- **Download CM1**: https://www2.mmm.ucar.edu/people/bryan/cm1/getcode.html.
- **Install a FORTRAN Compiler**: This will be needed to compile CM1.
- **Install netcdf-fortran package**: This is used for writing output in .nc format.
- **Install MPI or OpenMP (optional)**: These packages are used for multiprocessing.
- **Edit Makefile**: This is the file that is used to compile CM1 found in the src folder. You will need to link your netcdf-fortran installation and uncomment the section corresponding to your multiple processors and FORTRAN compiler option. You may need to include the paths to the "include" and "lib" folders for those packages on your machine as well.
- **Run a test (optional)**: This is a good point to do a test run.

  1. Compile CM1 by navigating, in the terminal, to the src folder and running the command:

     $ make

     If CM1 has compiled correctly then there should be a executable file "cm1.exe" in the run folder.
  2. In the run folder there is a file named "namelist.input". This file is where you set the parameters for your CM1 run. They are organized into sections, param1, param2, param3, etc. For the test run, change "timax" in the param1 section to 100. This will set the total simulation time to 100s. In the param9 section, change "output_format" to 2. This sets CM1 to output a netcdf file.
  3. You are now ready to run CM1. Navigate in the terminal to the run folder and run CM1 with the command:

     $.\cm1.exe &> cm1.print.out

     If CM1 has run successfully, you should have a document "cm1.print.out" with the standard output from the run and a file "cm1out.nc" with the output of the CM1 simulation. You can check that the run was successful by looking at the end of "cm1.print.out" for the line "Program terminated normally".

Background Atmosphere

- **Add base_blg.F file**: My background atmosphere routines are all in the file "base_blg.F". Put this file in the src folder. Included routines:

  1. **inversionfog_base**: This builds a fog layer embedded within an inversion layer (setup from my thesis see appendix B).
  2. **mixedcldlayer_base**: This creates a background atmosphere with a cloud layer aloft with a 0 saturated N^2 in the cloud and a constant unsaturated N^2 above and below.
  3. **mixedunsatlayer_base**: This creates a background atmosphere with three layers. In the middle is a layer with zero unsaturated N^2 (well-mixed). Above and below the atmosphere has the same constant unsaturated N^2. Relative humidity is every constant.
  4. **mixeddrylayer_base**: This is a simpler version of *mixedunsatlayer_base* with no water ($r_T = 0$).
  5. **marginalcld_base**: This creates an atmosphere with constant unsaturated N^2 that is everywhere at the saturation point ($r_T = r^*$).
  6. **cldoverclear_base**: This is my version of the background initialization for Walsh and Muraki (2024). Creates a marginal background as in *marginalcld_base* then perturbs the water to add a cloud in the upper part of the domain.
  7. **linearshear_wind**: Creates the background winds to build a linear shear layer. Constant wind from 0 to $H_c - H_d$, linear shear from $H_c - H_d$ to $H_c + H_d$, then constant wind from $H_c + H_d$ to the domain top.

- **Update Makefile**: You need to add my initialization routines to the Makefile so they are compiled and linked to the rest of the code. Below the comment that say "You shouldn't need to change anything below here" ;) is SRC = constants.F ... then a list of all the .F files. At the end of the list add *base_blg.F*.

  Below the comment "DEPENDENCIES" you will need to link the new .F file to the other routines it uses and which use it. Only *base.F* calls routines from *base_blg.F*, so at the line starting with "base.o:" add "base_blg.o" to the end of the list of .o files.

  After that head to the end of the Makefile and add the line

  base_blg.o: constants.o input.o cm1libs.o

  to give *base_blg.F* access to the other files it uses routines from.

- **Add routine calls to base.F**: Now you need to edit *base.F* in the src folder so that you can call the initialization routines in *base_blg.F*. First, you need to add the line

  use base_blg_module

so that *base.F* can call routines from *base_blg.F*. After the declaration of "subroutine base" you will see a list of "use ..." lines (around line 35). Add "use base_blg_module" to the end of this list.

Now you need to add the calls to the routines in *base_blg.F*. Around line 130 you will see the start of an if-elseif-else block. The first one should say "IF(isnd.eq.1)THEN" and then some background initialization code. The parameter "isnd" is set in the namelist.input file (section param2) and is how you pick the background initialization routine. Add the new options with isnd numbers. For instance my first couple are

```
IF(isnd.eq.1300)THEN
    call inversionfog_base(zh,pi0,prs0,th0,thl0,qv0,qc0,rst0,t0)
ELSEIF(isnd.eq.1301)THEN
    call mixedcldlayer_base(zh,pi0,prs0,th0,thl0,qv0,qc0,rst0,t0)
```

and so on and so forth.

Finally, you will need to add the background wind initialization routine. This goes in the iwnd section (the parameter to set the background wind initialization). The iwnd section should start around line 2300. Like before, add the lines

```
IF(iwnd.eq.1300)then
    call linearshear_wind(zh,u0,v0)
```

- **Add water initialization to init3d.F**: The disturbance initialization needs to be told that there is water in the background (even if you aren't initializing with a disturbance mode). Go into the file *init3d.F* and around line 295 you'll see a comment "This is here to ensure that certain idealized cases work" below this is a list of isnd options

```
IF( (isnd.eq.4 .or. isnd.eq.9 .or. isnd.eq.10 .or. isnd.eq.11 .or. isnd.eq.15) )THEN
```

  add your background isnd option numbers to this list.

- **Add access to $\bar{\theta}_\ell$ and $\bar{r}^*$**: I wanted to have access to $\bar{\theta}_\ell$ and $\bar{r}^*$ generated in the background initialization when I created my disturbance initial conditions. To do this we need to change the code so that $\bar{\theta}_\ell$ and $\bar{r}^*$ are outputs of the background initialization routines and inputs to the disturbance initialization routines. We will need to make changes to three of the .F files: CM1.F, base.F, and init3d.F.

  - **CM1.F**: Here you need to declare the new variables, allocate the memory for them, then add them to the routines calls in *base.F* and *init3d.F*. To declare them add "thl0" and "rst0" to the lists of variables being declared on lines 97 or 98 (you should see pi0, rho0, ... and qc0, qi0, ... on these lines).

    Allocations start around line 1380. You need to add the lines

```
allocate( thl0(ib:ie,jb:je,kb:ke) )
thl0 = 0.0
allocate( rst0(ib:ie,jb:je,kb:ke) )
rst0 = 0.0
```

to the list of allocation commands.

Around line 1466 is the call to the routine base. You need to add thl0 and rst0 to the list of inputs. The variable thl0 should go between th0 and rth0 while rst0 should go between dum2 and reqs_u.

Around line 2394 is the call to the init3d routine and you need to do the same again. Here thl0 goes between th0 and rth0 while rst0 goes between rrf0 and rain.

- **base.F**: Now you need to add thl0 and rst0 to the *base.F* file. First in the declaration of subroutine base (line 25) you need to add thl0 and rst0 is inputs in the same location as you just put them in *CM1.F*. The variable thl0 should go between th0 and rth0 while rst0 should go between rh0 and reqs_u.

  Now you just need to declare these variables, at line 56 and 57 the other background thermodynamic variables are declared. Add thl0 and rst0 to these lists (they should start with pi0, prs0, ... and qc0, qi0, ...)

- **init3d.F**: Finally, you just need to repeat the previous step but for the init3d.F file. Make sure you have added the new variables to the subroutine declaration in the same order as in the *CM1.F* file and make sure that you have declared them as arrays of the correct dimensions (same as pi0, prs0, etc.).

Initial Disturbances

- **Add init3d_blg.F file**: My disturbance initialization routines are all in the file "init3d_blg.F". Put this file in the src folder. Included routines:

  1. **inversionwave**: Computes the initial perturbation for the linear inversion trapped wave from my thesis.
  2. **linearshear_mode**: Generates the initial perturbation for a linear wave mode in the well mixed ($N^2 = 0$) layer backgrounds (cloud, unsaturated, or dry are all the same). You will need to compute the wave speed externally for this method. I generally use this to do unstable runs but it can initialize the stable modes as well.
  3. **cldoverclear_wave**: My version of the cloud edge wave experiment of Walsh and Muraki (2024). This code hasn't been tested with a Boussinesq convergence test so may not work as well as their runs but it can make a traveling wave that looks good at least.

- **Modify Makefile and add routine calls in init3d.F**: Refer to the corresponding sections for the background routines. Following the same steps, but for this file, modify the Makefile (add init3d.F to the SRC list, add init3d_blg.o as a dependency of init3d.o,

and include the dependencies of init3d_blg.o). Then modify *init3d.F* by adding the use init3d_blg_module and calls to the new initialization routines (these go with parameter option of iinit).

My CM1 Workflow

- **Make a directory for this experiment**: To stay organized I usually start by making a separate directory for the CM1 experiment I'm running. In this directory I copy the namelist.input file and symbolically link cm1.exe,

  $ ln −s <relative path to run>\run\cm1.exe cm1.exe

- **Compute Background Atmosphere**: My program *CM1_quick_viewer.py* requires a separate background file so I usually make this file first. In namelist.input configure the section param0 and param1 with the grid information for your run. In param2 set iconly = 1, this will tell CM1 to only generate initial conditions and not integrate forward. Set up the rest of param2 with the settings for your run including the background initialization (isnd) and background winds (iwnd) but make sure that iinit = 0 so there is no disturbance added to your background. Additionally, you will need to set any of the parameters for your initialization routine in param8. Then run CM1, which should be quick because you are just generating the background conditions. Rename the output file so that it is not overwritten during the next step.

- **Preform the CM1 run**: Now you're ready to run your CM1 experiment. Return to the namelist.input file and change iconly = 0 and set iinit to your disturbance initialization routine. Make sure you have set the run time the way you want it in param1 and run CM1.

- **Use the quick viewer program**: You can now get a quick view of your run using the program *CM1_quick_viewer.py*. Just change the paths on line 30 and 31 to your run and run the *CM1_quick_viewer.py* program. You can use the interactive window to view different fields and click through the frames from your output.