Bluejay expansion from 1-D to 2-D

Note that this is a simplified version with 4 species and 7 altitude bins.

## Nomenclature

* I added a new variable, *n\_horiz*, which is the number of vertical columns or slices. It is currently set to 3, but can be set to 1, which should produce the same output as the original 1-D version, or to any other number.
* In many places in the code, I set things to iterate through each vertical column. In these cases, I always named the index of the vertical column *ihoriz*, e.g., for ihoriz in 1:n\_horiz…

## What I changed:

* Where the atmospheric state is saved, values for each vertical column can be saved. For example:
  + Dictionary atm\_dict now has structure {species1:[[densities with altitude for column 1][densities with altitude for col. 2][densities with alt. for col. 3]…], species2:[[…}
* For functions chemJmat and ratefn, a loop over n\_horiz was added, so that sparse matrices (dimensions n\_horiz\*length(active\_longlived)\*num\_layers by ihoriz\*length(active\_longlived)\*num\_layers) are produced. When this was implemented, the outputs for each vertical column were identical, as expected. Note that this was only tested with four species in the model.
* Horizontal transport:
  + A new function update\_horiz\_transport\_coefficients, calculates horizontal transport coefficients at the edges and between columns.
  + A new function boundaryconditions\_horiz in Core.jl allows edge boundary conditions (i.e., into and out of the first and last vertical columns) to be set and saved in bc\_dict\_horiz, which now has structure [[0.0 0.0; 0.0 0.0] for ialt in 1:GV.num\_layers] for each species. Currently, these are set to zero flux. If density or deposition velocity boundary conditions are required instead, this will need some further work.
  + Dictionary fluxcoefs\_horiz\_all has structure Dict{Symbol, Array{Float64}}(s=>zeros(9,2) for s in GV.all\_species), and has not yet been expanded to include different values for each column (because at the moment all values are zero for each column). This needs to be expanded. Eventually fluxcoefs\_horiz\_all and fluxcoefs\_all should have the same dimensions as one another.
* Vertical transport for each column:
  + The dictionary that stores the lower and upper boundary conditions, bc\_dict, now has structure [[X X; X X] for ihoriz in 1:n\_horiz] for each species (defined in Core.jl).
  + The vertical transport variables fluxcoefs\_all and fluxcoef\_dict have been expanded to include values for each vertical column. They now have structures Dict{Symbol, Vector{Array{ftype\_ncur}}} ([s=>[fill(0., GV.n\_all\_layers, 2) for ihoriz in 1:n\_horiz] for s in species\_list]). Note that the values have not yet been changed to be different for each column, and vertical transport is currently the same for each column.
  + The diffusion coefficient dictionary, Dcoef\_dict*,* has been expanded to include multiple columns. It now has structure: Dict{Symbol, Vector{Vector{ftype\_ncur}}}([s=>[[Dcoefs with altitude] for ihoriz in 1:n\_horiz] for s in species\_list])
  + The eddy diffusion structure, *K*, has been expanded to include different eddy diffusion profiles for each vertical column. It now has structure: [[K with altitude] for ihoriz in 1:n\_horiz].
  + The dictionary with scale height values (H0\_dict) has been expanded to take values for all vertical columns. It now has the structure: Dict{String, Vector{Vector{ftype\_ncur}}}("neutral"=>[[values with altitude] for ihoriz in 1:n\_horiz], "ion"=>[values with altitude] for ihoriz in 1:n\_horiz]).
  + Different lower and upper boundary conditions for each vertical column can be set in MODEL\_SETUP.jl. Dictionary speciesbclist now has a vector of n\_horiz vectors for each species and BC type.

## My next planned steps were:

* I was about to expand MODEL\_SETUP.jl to incorporate multiple temperature profiles for each vertical column.
* Currently, horizontal transport is set to zero. The model is currently not set up for non-zero horizontal fluxes. Need to add the terms for the horizontal transport to the ∂ni/∂t equation (see Mike’s summary sheet with equations). I had just added horiz\_wind\_v in MODEL\_SETUP.jl to include some horizontal velocities (set to 0 currently).
* All of the places that might need to be looked at/checked/adapted in the future are indicated with the comment “#MULTICOL WARNING”

## My medium-term planned steps were (not necessarily in a linear order):

* Test that the model is doing the right thing.
  + Make sure that if 1) there are zero flux edge boundary conditions and horizontal transport is set to zero, 2) the lower and upper boundary conditions are the same for each vertical column, and 3) that the other inputs (e.g., temperature, vertical mixing) are the same for each vertical column, the output densities in each vertical column are identical.
  + Make sure that **if n\_horiz is 1**, the model produces exactly the same output as the original 1-D version.
  + Set up a test model with two altitude bins by 2 vertical slices and without chemistry. Compare to analytical solution.
* Connect the front and back edge such that the model is cyclic rather than linear.
* I reduced the species list to four species (O, O2, O3 and Opl); put all the species back in.
* I reduced the altitude grid to 7 bins (by reducing zmax in MODEL\_SETUP.jl); put zmax back to 250e5 (line commented out).

## Small things to note:

* Some further edits will need to be made to function diffusion\_timescale in AnalyzeChemAndTransport if this is ever called (it isn’t set to be called in the current set-up).
* In some places where function n\_tot is called, I have hardcoded the argument ihoriz as 1 for the time being, which will need to be changed for flexibility going forward. There is a ‘#MULTICOL WARNING’ comment in each instance of this.