Ambipolar Diffusion and Dynamo Requirements

Art provided a list of variables needed for ionospheric (and electrodynamic?) calculations to be added to the WACCM-X h5. model. Here is a list of the variables and ideas of where to access these variables in the current model:

- Magnetic latitude and longitude: Apex coordinates used in WACCM. Variables called 'alatm' and 'alonm' in mo_apex.F90 module called in module iondrag.F90, Need to add these to the 'pbuf' physics buffer with pbuf_add subroutine.
- Magnetic field components: Calculated in mo_apex.F90 module called in module iondrag.F90. B_y is variable 'bnorth', B_x is variable 'beast', B_z is variable 'bdown', and summation of 3 components is variable 'bmag'. These are 2D variables but need to be expanded to 3D variables to include the vertical dimension
- ExB/B^2 components(v): Calculated in exbdrift.F90 module which is called in iondrag.F90 module. v_x is variable 'ue', v_y is variable 've' and v_z is variable 'we'. These are 2D variables and need to be expanded in the vertical to 3D. v_z needs to be converted to interface levels('wei'). 2D 'ue', 've', and 'we' are stored in the 'pbuf' physics buffer.
- Temperature and winds: T, U, V are in 'state' physics structure at midpoint vertical levels. W is in 'state' physics structure at midpoints named 'omega'. Need to convert to interface levels and add to 'state' structure ('omegai' or 'wi'?) and update each time step.
- n(O),n(O_2),n(N_2),n(H),n(He): All but He in 'state' structure 'q' array. He not critical now so revisit later.
- Chapman function radiation attenuation near terminator?
- Scale height: Needs to be calculated and added to 'state' structure and update each time step
- Gravity: Would prefer altitude dependent but revisit later
- Geopotential altitude: In 'state' structure as 'zm' at midpoint and 'zi' at interface levels
- Zenith angle: Cosine of zenith angle calculated in zenith.F90 module
- Masses of molecules: 'const_mw' array in constituents.F90 module accessed through 'cnst_get_ind' subroutine
- Masses of ions: Found ions of O,O_2,N,N_2,NO but not H and He in 'q' array of 'state' structure. Need to add H (and He later).
- Ion production rates: Need to locate where production rates calculated, maybe near end of chemistry? To solve for transport due to ambipolar diffusion, we will use implicit solver. The production/loss terms form the RHS of the solver, which should help assuring numerical stability.
- Solar spectrum and auroral spectra: Solar spectrum need to look at photochemistry to see where used. Probably in mo_aurora.F90 module accessed in iondrag.F90 module. (We will get a total production/loss rate, and don't need to get individually ion/electron production from solar photoionization or auroral production.)
- Heating rates: Look in mo_waccm_hrates.F90 module?
- Conduction coefficients at interface levels for n(O),n(N_2),n(O_2),n(H),n(He),T,[T_e,n_i, n_e]: Molecular conductivity in 'state' structure calculated based on O_2, N_2, and O. (The heating rates and thermal conduction are needed for calculating the electron/ion temperature, which will be done before the transport.
- Values at previous time of n(O^+),n(H^+),n(He^+),T_i,T_e,[n(O_2^+),n(NO^+),...]: 'state' structure contains previous time step values until updated. Need to find where each updated but H^+ and He^+ are not present so H^+ needs to be added (He later).
- Boltzmann constant: Use shr_const_mod variable is shr_const_kboltz
- Electron number density: In 'state' structure variable 'q' state%q(i,k,id_elec). Units are cm^-3^?
- Dip angle: calculated in iondrag_calc and needs to be put in physics buffer 'pbuf'
- Neutral temperature: In 'state' structure variable 't' midpoints. Need to convert to interface like in diffusion_solver.F90 module
- Electron temperature: set to neutral temperature at first step and calculate and update in energy routine
- O,O_2,N_2 (and ions of these) number density: Convert from 'state' structure 'q' variable mmrs using 'invariants'?, mmr2vmr?, similar to what is done in mo_waccm_photo.F90 module
- Pressure: In 'state' structure 'pmid' variable in Pascals
- Constituent dependent gas constant: In 'state' structure 'rairv' variable

Notes:

Where interface variables are needed and not currently available need to calculate and add 'i' to midpoint name

Dip angle is calculated from magnetic components (already done in iondrag_calc)

iondrag_calc subroutine from iondrag.F90 module called near end of 'tphysac' routine (tphysac.F90 module) which is called from 'physrun2' routine (physpkg.F90 module) for each chunk. 'physrun2' is called from 'cam_run2' routine (cam_comp.F90 module)