DNA Code Documentation

I. The 'physical_parameters' namelist

`nu [real]`: collision frequency (normalized to v_{ti}/R).

`omt [real]`: temperature gradient scale length R/L_{ti}.

`omn [real]`: density gradient scale length R/L_n.

`Ti0Te [real]`: ratio of background ion to electron temperature.

II. The 'numerical_parameters' namelist

`kxmin [real]`: minimum radial wavenumber.

`kymin [real]`: minimum binormal wavenumber.

`kzmin [real]`: minimum parallel wavenumber.

`nkx0 [integer]`: number of radial wavenumbers. Note that the code directly calculates the wavenumbers $k_x = 0$ through $k_{x,\max}$ =(`nkx0`$-1)\times$`kxmin`. The negative k_x modes are implicitly defined via the reality constraint.

`nky0 [integer]`: number of binormal wavenumbers, including positive and negative wavenumbers.

`nkz0 [integer]`: number of parallel wavenumbers, including positive and negative wavenumbers.

`nv0 [integer]`: number of Hermite polynomials.

`hyp_x [real]`: prefactor for radial hyperdiffusion.

`hyp_y [real]`: prefactor for binormal hyperdiffusion.

`hyp_z [real]`: prefactor for parallel hyperdiffusion.

`hyp_v [real]`: prefactor for hyper-collisions.

`hypx_order [integer]:` order of radial hyperdiffusion.

`hypy_order [integer]:` order of binormal hyperdiffusion.

`hypz_order [integer]:` order of parallel hyperdiffusion.

`hypv_order [integer]:` order of hyper-collisions.

`hyp_conv [real]:` prefactor for a Krook-like term acting on the $k_z = 0$, $n = 0$ part of the distribution function (where n is the Hermite number). The net prefactor is `hyp_conv`×`nu`, so that setting this equal to one extends the minimum collisionality to $k_z = 0$, $n = 0$. This part of the distribution function is otherwise often subject to slow growth over the course of a simulation.

`num_v_procs [integer]:` number of processors. Note that the code is only parallelized over the Hermite coordinate at this time.

`courant [real]:` courant factor for the nonlinear time step adaptation.

III. The 'diagnostics' namelist

`diagdir [character]:` output directory.

`istep_ffm [integer]:` parameter defining how frequently to activate diagnostics calculating ϕ^2 and the heat flux summed over all coordinates. This data is output to the file 'ffm.dat.' As with all of the following '`istep`' parameters, the diagnostic is activated every `istep`-th time step.

`istep_energy3d [integer]:` parameter defining how frequently to activate diagnostics calculating the three dimensional free energy, energy drive, collisional dissipation and any dissipation from artificial hyperdiffusion. This data is output to the file 'energy3d.dat.'

`istep_energy [integer]:` parameter defining how frequently to activate diagnostics calculating the total value of various energy-related quantities–1. the 'entropy' part of the free energy, 2.

the electrostatic part of the free energy, 3. total RHS of the free energy evolution equation, 4. free energy drive, 5. collisional dissipation, 6. hyper-collisional dissipation, 7. dissipation via other artificial hyperdiffusion, 8. nonlinear contribution to the energy equation (should be extremely small), 9. calculated time derivative of the free energy. This data is output to the file 'energy3d.dat.'

`istep_hermite [integer]:` parameter defining how frequently to activate diagnostics calculating the free energy as a function of Hermite polynomial, summed over all spatial coordinates. This data is output to the file 'energy_hermite.dat.'

`istep_gout [integer]:` parameter defining how frequently to output the entire distribution function. This data is output to the file 'gout.dat.'

`istep_nlt [integer]:` parameter defining how frequently to activate nonlinear energy transfer diagnostics.

`istep_eshells [integer]:` parameter defining how frequently to activate energy diagnostics formulated in terms of $k_\perp$ shells.

`min_shell_width [integer]:` parameter for determining the $k_\perp$ shells.

`istep_real [integer]:` parameter defining how frequently to enforce the reality constraint on $k_x = 0$ modes.

`istep_fmom3d [integer]:` parameter defining how frequently to activate diagnostics calculating the three dimensional electrostatic potential and pressure fluctuations. This data is output to the file 'fmom3d.dat.'

`istep_gamma [integer]:` parameter defining how frequently to calculate the growth rate for linear initial-value simulations.

`istep_schpt [integer]:` parameter defining how frequently to write a security checkpoint in case the simulation stops unexpectedly.

`output_nlt_n [integer]:` flag for calculating nonlinear transfer functions for various Hermite polynomials.

`istep_gk [integer]:` parameter defining how frequently to output the distribution function for certain wavenumbers. This is useful if memory constraints limit the frequency at which the entire distribution function can be output.

`gk_ky_index [integer]:` index defining the k_y mode for a k_z scan associated with `istep_gk`.

`gk_kz_index [integer]:` index defining the k_z mode for a k_y scan associated with `istep_gk`.

IV. The '`flags`' namelist

`nonlinear [logical]:` 'T' denotes a nonlinear simulation.

`calc_dt [logical]:` 'T' activates calculation of the initial time step. Otherwise `dt_max` must be set.

`comp_type [character]:` 'IV' activates initial value computation, 'EV' activates eigenvalue computation.

`checkpoint_read [logical]:` 'T' activates a checkpoint read as an initial condition in case of restarts. Additional data output will be appended to existing files.

`etg_factor [real]:` factor determining the adiabatic response for $k_y = 0$ modes. `etg_factor`=0.0 defines to an 'ETG' simulation, and `etg_factor`=1.0 defines to an 'ITG' simulation.

V. The '`eigensolve`' namelist

`n_ev [logical]:` number of eigenvalues to solve for (maximum is `nv0`).

`left_ev [logical]:` 'T' activates calculation of the left eigenvectors.

`kxmax0 [real]:` maximum k_x wavenumber. Whatever is input is overwritten and the correct value is output in the resulting parameters.dat file. This is then used in post-processing diagnostics.

`kymax0 [real]:` maximum k_y wavenumber (not including the Nyquist wavenumber). Whatever is input is overwritten and the correct value is output in the resulting parameters.dat file. This is then used in post-processing diagnostics.

`kzmax0 [real]:` maximum k_z wavenumber (not including the Nyquist wavenumber). Whatever is input is overwritten and the correct value is output in the resulting parameters.dat file. This is then used in post-processing diagnostics.

VI. The '`initial_value`' namelist

`max_itime [integer]:` maximum number of time steps before termination of the simulation.

`max_walltime [integer]:` maximum walltime before termination of the simulation.

`max_time [real]:` maximum time in R/v_{ti} before termination of the simulation.

`dt_max [real]:` initial time step. The time step will be calculated automatically if `calc_dt`=T, and will be automatically adapted in the nonlinear regime.