Introduction

While working on coupling the fusion codes - XGC and GENE, a sufficient understanding of the domain representation and data transfer process/ interaction was desired. Thus birthing the need for this document. Below are some of the questions asked and the corresponding answers as evident within the code.

Questions asked

1. What are the fields being coupled and how can they be described ?

   The field being coupled is the Consistent Electric Potential field (efield) which is a 3D field derived from solving the Poisson's equation. This field is sent from XGC to GENE and is used in the electron push procedure. Also, it is sent as a combination of a 2D array dpot(: , :) and a 1D array pot0(:) as defined within the cce_send_field function called within the es_main program.

   Additionally, GENE sends its ion charge density as a "field" to XGC. This "ion charge density" quantity is sent in the form of a 1D array as expressed within the function cce_send_density (psn%idensity) called within the chargei module. This is a discrete field that is deposited by the ions it simulates.

   Additionally, other relevant fields within the code were :

   (a) A magnetic field (Bfield) associated with the grid, written as "grid%bfield" and can be represented in both the magnetic Straight Feild Line (SFL) and GENE's (x, y, z) coordinate systems.

   (b) Two (2) auxilliary fields (aux_field) that are derived from Maxwell's electromagnetic PDE's expressed within the "calc_aux_fields function" - Electrostatic and Electromagnetic fields – emfields

2. Describe the GENE coordinate system ?

   GENE simulates a 5D domain with 3 spatial coordinates and 2 velocity coordinates. The coupling works with the 3D spatial domain it simulates. GENE uses a variation of the magnetic straight field line coordinate system (x, y, z) = (radial, binormal and parallel) directions and it can be represented as a 3D grid with poloidal cuts made into it (define_cuts function) to create 2D planes that would match XGC's poloidal planes.

   GENE's coordinate system is designed to be aligned to the background magnetic field to save on computation with its information on poloidal planes reconstructed using binormal and parallel boundary conditions i.e. periodicity in both y and z. Also, GENE has a background grid that is defined as 1D grids in both the radial and parallel direction – x_GENE_wb and z_GENE_wb defined within the set_GENE_aux_grids function in the poloidal_planes.F90.

   PS: Creating the GENE poloidal planes can be achieved via two ways;

   a) Importing XGC grid into GENE via reading the XGC nodes, and doing a 2D 3rd order Lagrange interpolation on the R and Z bounding box as can be seen in the import_XGC_grid.F90.

b) Generating the non-uniform grid internally using the xval_a and zval (initialized in coordinates). Chease hdf5 geometry arrays and attributes are used in creating this auxiliary GENE grids and accruing poloidal grids with RZ values as seen in the generate_grid_nonuni function in create_nonuni_grid.F90.

3. How does GENE interpolate data from and to its background grid ?

   GENE's auxuliary background grid x_GENE_wb is derived from xval_a onto which data transfers (interpolations) are performed as seen within the read_numerical_profiles function in the profile module. – poloidal_planes.F90

   For the coupling, z_out_nonuni is the underlying background grid. This is a flux surface data array that is poloidal and present within both XGC and GENE.

   So, the implementation of the data transfer from "z_gene" to "z_xgc" has the following steps:

   (a) Compute the SFL angle for each XGC unstructured grid node on each poloidal plane. The get_miller mapping function is used to generate this z_xgc grid
   (b) Given that GENE's parallel grid's [z_gene] span (¡2*Pi), Fourier invert the GENE data for higher accuracy
   (c) Then interpolate the data on each flux surface and each y location from z_gene to z_xgc

   The function for projecting data from the GENE's xyz coordinate system to the poloidal planes is found within poloidal_planes.F90:

   initialize_planes

   prepare_matrices_nonuniform

   diag_field

   diag_planes

   xyz_to_planes_nonuni

   invert_extend_nonuni - interpolates from the z_GENE_wb to z_out_nonuni (data on flux surface) on finer poloidal grid before doing a matrix multiply in the real space.

   Also, planes_to_xyz_nonuni function in poloidal_planes_nonuni.F90 is used to interpolate data from the poloidal planes back to GENE's xyz coordinate system.

   The routine for mapping data from poloidal planes back to GENE's xyz coordinate system lies within aux_fields.F90:

   calc_aux_fields

   compute_emfields

   planes_to_xyz_nonuni – This interpolates the data across its flux surfaces and interpolates on zval. Then dumps the transformed data and does an inverse FFT in y from the poloidal plane to nonunim grid.

import_planes_nonuni - receives & processes XGC's field, reshuffles nodes & data structure

import_GENE_density – reads the charge density data already written into the file

perform data interpolation- inverse matrix multiply the data from the flux surfaces, then Lagrange interpolate the data from z_out_nonuni unto zval. Afterwards perform a reverse fourier transform in y

PS: XGC uses bsplines for all interpolations

4. Are there functions to retrieve the value of these fields at any point within the domain ?

   Yes, there are functions to do so.

   In XGC, b_interpol(r,z,phi) function gives the magnetic field value for a given r, z and phi.

   Also, in XGC's push.F90, we have a field object that has both magnetic and electric potential attributes and whose value can be calculated at a given point and time – field(fld, t, rz_outside) function.

   In GENE, the above discussed calc_aux_field routine is used in various computations to retrieve value of the auxilliary fields.

   PS: cce_initialize is called before the transformation matrix is computed within the run-gene call in the main program.

5. In XGC? How do we get values at non-poloidal plane points/locations ?

   Finite differnce is used within the search and col_f_module (f-grid collision) routines but not directly related to the fields being coupled. So, it still remains unclear how to get the explicit value of the fields at non-poloidal locations.

   I am still checking these routines for deeper understanding.