2021/09/13

User Manual of the TASK/FP Code

Contents

1 Structure

1.1 Source File

- Environment routines
  - `fpcomm.f90`: Definition of variable module, allocation of arrays
    - Module `fpcomm_parm`: Definition of constants and input parameters (USE bpsd_kinds, bpsd_constants, commpi, plcomm, obcomm_parm)
    - Module `fpcomm`: Definition of common variables
    - Subroutine `fp_allocate`: Allocation of common arrays
    - Subroutine `fp_deallocate`: Deallocation of common arrays
    - Subroutine `fp_allocate_ntg1`: Allocation of time history `ntg1` array
    - Subroutine `fp_deallocate_ntg1`: Deallocation of time history `ntg1` array
    - Subroutine `fp_adjust_ntg1`: Adjust (save, deallocate, allocate larger, recover) time history `ntg1` array
    - Subroutine `fp_allocate_ntg2`: Allocation of profile time history `ntg2` array
    - Subroutine `fp_deallocate_ntg2`: Deallocation of profile time history `ntg2` array
    - Subroutine `fp_adjust_ntg2`: Adjust (save, deallocate, allocate larger, recover) profile time history `ntg2` array
  - `fpmain.f90`: Main routine
    - Initialization of MPI and graphics
    - Initialization of input parameters of module, pl, eq, and fp
    - Read input namelist file `fpparm`
    - Start menu input loop `fpmenu`
    - Termination of graphics and MPI
  - `fpinit.f90`: Initialization of input parameters
    - Module `fp_init`: Set default values of input parameters
  - `fpmenu.f90`: Process control
    - Module `fp_menu`: menu command input and execute
      - `P` or with `=`: namelist read of input parameters
      - `V`: view all input parameters
      - `R`: start new calculation (set initial condition and go)
      - `C`: continue previous calculation (just go)
      - `G`: graphic output of calculated results
      - `F`: file output of calculated results
      - `S`: save present status of calculation into a file
      - `L`: load a file to continue previous calculation
      - `W`: print out interesting quantities
      - `Y`: debug output [temporal]

            - `Z`: debug output [transport coefficients]
            - `Q`: quit menu input and stop
    - `fpparm.f90`: Read and check input parameters
        * Subroutine `fp_parm`: analyze parameter input
            - read one line from standard input and analyze command or namelist
            - read namelist input from a file
            - read one phrase and analyze as a namelist
    - `fpbroadcast.f90`: Broadcast input parameters
    - `fpview.f90`: Show input parameters

- Execution routines

    `fpprep.f90`: Preparation of run (mesh, initial profile, initialization)

    `fpbounc.f90`: Preparation of bounce parameters

    `fploop.f90`: Time loop for execution

    `fpexec.f90`: Execution of one time step

- Calculation of coefficients of equations

    `fpcoef.f90`: Calculation of various coefficients

    `fpcalc.f90`: Calculation of collisional term (linear operator)

    `fpcalcn.f90`: Calculation of collisional term (nonlinear operator)

    `fpcalcnr.f90`: Calculation of collisional term (relativistic nonlinear operator)

    `fpcale.f90`: Calculation of static electric field term

    `fpcalr.f90`: Calculation of radial diffusion term

    `fpcalw.f90`: Calculation of quasi-linear term (given wave field)

    `fpcalwm.f90`: Calculation of quasi-linear term (using wm results)

    `fpcalwr.f90`: Calculation of quasi-linear term (using wr results)

    `fpcdbm.f90`: Calculation of CDBM radial diffusion coefficients

    `fpnfrr.f90`: Calculation of fusion reaction term (isotropic distribution)

    `fpnflg.f90`: Calculation of fusion reaction term (anisotropic distribution)

    `fpdisrupt.f90`: Calculation of disruption-related tems

- Calculation of transport coefficients (by Ota)

    `fpcaldeff.f90`: Effective particle diffusion coefficients

    `fpcalchieff.f90`: Effectiv thermal diffusion coefficients

    `fpcaltp.f90`: Particle confinement time, tauP

    `fpcalte.f90`: Energy confinement time, tauE

    `fpchecknc.f90`: Radial diffusion coefficients (neoclassical diffusion)

- File IO routines

    `fpfile.f90`: Save and load restart data

    `fpfout.f90`: File output of graphic data

    `fpoutdata.f90`: File output of intermediate data (by Ota)

`fpread.f90`: Read FIT3D data from file

`fpreadeg.f90`: Read Experimental data from file (by Nuga)

`fpsave.f90`: File output of various data (by Nuga)

`fpwmin.f90`: File input of full-wave analysis (wm) results

`fpwrin.f90`: File input of ray/beam tracing analysis (wr) results

`fpwrite.f90`: File output of trcoef data (by Ota)

- Graphic routines

  `fpgout.f90`: Graphic output for gsaf

  `fpcont.f90`: Graphic subroutines

- Library routines

  `fpmpi.f90`: MPI interface for fp

  `fpsub.f90`: Subroutine library (FPMXWL, FPNEWTON)

- Orbit-averaging routines (by Ota)

  `fowcomm.f90`: Definition of fow variables and allocation

  `fowprep.f90`: Preparation of fow, initialization of variables

  `foworbit.f90`: Interface to ob

  `fowclassify.f90`: Orbit classification and data output to file

  `fowdistribution.f90`: Distribution conversion and integrated quantities

  `fowloop.f90`: Time loop for execution

  `fowexec.f90`: Execution of one time step

  `fowcoef.f90`: Calculation of various coefficients

  `fowsource.f90`: Calculation of source terms

  `fowlib.f90`: Library for fow