PX915 Summer Project - Ben Gosling

Generated by Doxygen 1.8.17

Chapter 1

Data Type Index

1.1 Data Types List

Here are the data types with brief descriptions:

Chapter 2

Data Type Documentation

2.1 py_scripts.distribution_calc.dist_f Class Reference

dist_f Class.

Public Member Functions

- def __init__ (self, dir)

  init
- def read_dist_data (self)

  read_dist_data
- def plot_p_dist_func (self, smooth=False, scaled_x=False, plot_hot_e=False)

  plot_p_dist_func

Public Attributes

- **directory**
- **files**
- **nfiles**
- **p_range**
- **p_max**
- **p_min**
- **res**
- **epoch_data**
- **v_th**
- **p_norm**
- **times**
- **dist_funcs**
- **smooth_dist_funcs**
- **dist_funcs_av**
- **smooth_dist_funcs_av**
- **momenta_bins**
- **energy_bins**
- **energy_bins_keV**
- **area_av**
- **max_boltz_eq**
- **plot_dist_funcs**
- **plot_dist_funcs_av**
- **y_max**
- **y_min**

2.1.1 Detailed Description

dist_f Class.

Class that reads and ouptputs the electron momentum distrubution function from output dist_ .. .sdf files.

2.1.2 Constructor & Destructor Documentation

2.1.2.1 __init__()

```
def py_scripts.distribution_calc.dist_f.__init__ (
              self,
              dir )
```

init

The constructor

Parameters

self	The object pointer
dir	Directory where data is stored (str)

2.1.3 Member Function Documentation

2.1.3.1 plot_p_dist_func()

```
def py_scripts.distribution_calc.dist_f.plot_p_dist_func (
              self,
              smooth = False,
              scaled_x = False,
              plot_hot_e = False )
```

plot_p_dist_func

Plots all distribution functions

Parameters

self	The object pointer
smooth	(Logical) Smooths out function of random zeros
scaled_x	(Logical) Scales momentum using p_norm
plot_hot↩ _e	(Logical) Plots hot electron tail

2.1.3.2 read_dist_data()

```
def py_scripts.distribution_calc.dist_f.read_dist_data (
              self )
```

read_dist_data

Read and store distrinution functions at output times

Parameters

self	The object pointer

The documentation for this class was generated from the following file:

- /home/u1706745/summer_project/epoch_surra/py_scripts/distribution_calc.py

2.2 py_scripts.fields_calc.EM_fields Class Reference

EM_fields Class.

Public Member Functions

- def __init__ (self, dir)

  init
- def get_2D_Electric_Field_x (self)

  get_2D_Electric_Field_x
- def get_2D_Electric_Field_y (self)

  get_2D_Electric_Field_y
- def get_2D_Magnetic_Field_z (self)

  get_2D_Magnetic_Field_z
- def get_2D_FFT (self, field, square_mod=True)

  get_2D_FFT
- def get_time_FFT (self, field, square_mod=True)

  get_time_FFT
- def get_space_FFT (self, field, square_mod=True)

  get_space_FFT
- def get_filtered_signals (self, laser=False, plot_E=False, plot_B=False)

  get_filtered_signals
- def get_flux (self, laser=False, time_series=False)

  get_flux
- def get_flux_grid_av (self, ncells=50, laser=False)

  get_flux_grid_av

Public Attributes

- **directory**
- **epoch_data**
- **timesteps**
- **nx**

2.2.1 Detailed Description

EM_fields Class.

Class that reads and calculates field quantities from fields_ output files.

2.2.2 Constructor & Destructor Documentation

2.2.2.1 __init__()

```
def py_scripts.fields_calc.EM_fields.__init__ (
              self,
              dir )
```

init

The constructor

Parameters

self	The object pointer
dir	Directory where data is stored (str)

2.2.3 Member Function Documentation

2.2.3.1 get_2D_Electric_Field_x()

```
def py_scripts.fields_calc.EM_fields.get_2D_Electric_Field_x (
              self )
```

get_2D_Electric_Field_x

Get time and space data of Ex field i.e Ex(x,t)

Parameters

self	The object pointer

2.2.3.2 get_2D_Electric_Field_y()

```
def py_scripts.fields_calc.EM_fields.get_2D_Electric_Field_y (
              self )
```

get_2D_Electric_Field_y

Get time and space data of Ey field i.e Ey(x,t)

Parameters

self	The object pointer

2.2.3.3 get_2D_FFT()

```
def py_scripts.fields_calc.EM_fields.get_2D_FFT (
              self,
              field,
              square_mod = True )
```

get_2D_FFT

Get 2D FFT (i.e space and time) for specific field

Parameters

self	The object pointer
field	EM Field to FFT (inputs are either 'Ex', 'Ey', 'Bz')
square_mod	(Logical) outputs the sqaured modulus of the FFT

2.2.3.4 get_2D_Magnetic_Field_z()

```
def py_scripts.fields_calc.EM_fields.get_2D_Magnetic_Field_z (
              self )
```

get_2D_Magnetic_Field_z

Get time and space data of Bz field i.e Bz(x,t)

Parameters

self	The object pointer

2.2.3.5 get_filtered_signals()

```
def py_scripts.fields_calc.EM_fields.get_filtered_signals (
              self,
              laser = False,
              plot_E = False,
              plot_B = False )
```

get_filtered_signals

Finds filtered signals of Ey and Bz fields (either laser signal or SRS signal)

Parameters

self	The object pointer
laser	(Logical) Whether to output laser sginal (true) or SRS signal (false)
plot↩ _E	(Logical) Whether to plot the filter result at set grid point to test if it works (Ey field)
plot↩ _B	(Logical) Whether to plot the filter result at set grid point to test if it works (Bz field)

2.2.3.6 get_flux()

```
def py_scripts.fields_calc.EM_fields.get_flux (
              self,
              laser = False,
              time_series = False )
```

get_flux

Finds Poynting flux in x direction Sx = (EyBz-ByEz)/mu0 (SRS produces scattered light with same polarisation as the laser (i.e Ez and By are negliable) thus Sx = EyBz/mu0)

Parameters

self	The object pointer
laser	(Logical) Whether to use laser sginal (true) or SRS signal (false)
plot↩ _E	(Logical) Whether to output the Sx time series (true) or the time average (false)

2.2.3.7 get_flux_grid_av()

```
def py_scripts.fields_calc.EM_fields.get_flux_grid_av (
              self,
              ncells = 50,
              laser = False )
```

get_flux_grid_av

Averages Poynting flux over ncells (near LH boundary for backscatter SRS and RH boundary for laser)

Parameters

self	The object pointer
ncells	Number of cells to average over (default 50)
laser	(Logical) Whether to use laser sginal (true) or SRS signal (false)

2.2.3.8 get_space_FFT()

```
def py_scripts.fields_calc.EM_fields.get_space_FFT (
              self,
              field,
              square_mod = True )
```

get_space_FFT

Produces 1D space FFT for specific field

Parameters

self	The object pointer
field	EM Field to FFT (inputs are either 'Ex', 'Ey', 'Bz')
square_mod	(Logical) outputs the sqaured modulus of the FFT

2.2.3.9 get_time_FFT()

```
def py_scripts.fields_calc.EM_fields.get_time_FFT (
              self,
              field,
              square_mod = True )
```

get_time_FFT

Produces 1D time FFT for specific field

Parameters

self	The object pointer
field	EM Field to FFT (inputs are either 'Ex', 'Ey', 'Bz')
square_mod	(Logical) outputs the sqaured modulus of the FFT

The documentation for this class was generated from the following file:

- /home/u1706745/summer_project/epoch_surra/py_scripts/fields_calc.py

2.3 py_scripts.hot_elec_calc.hot_electron Class Reference

Public Member Functions

- def __init__ (self, dir)

  init
- def get_flux_dist (self, plot=False, log=False)

  get_flux_dist
- def split_dist (self, n=3, plot=False, log=False)

  split_dist
- def fit_maxwellians (self, n=3, plot=False, log=False)

  fit_maxwellians
- def get_hot_e_temp (self, n=3, av=True, plot=False)

  get_hot_e_temp
- def get_energy_frac (self)

  get_energy_frac

Public Attributes

- **directory**
- **epoch_data**
- **nbins**
- **E_data**
- **E_bins**
- **idx**
- **flux_parts**
- **E_parts**
- **T_vals**
- **scaled_fits**
- **MB_fits**
- **amplitudes**
- **scaled_fits_full**
- **final_fit**
- **T_hot**
- **T_hot_av**
- **E_hot_frac**

2.3.1 Constructor & Destructor Documentation

2.3.1.1 __init__()

```
def py_scripts.hot_elec_calc.hot_electron.__init__ (
              self,
              dir )
```

init

The constructor

Parameters

self	The object pointer
dir	Directory where data is stored (str)

2.3.2 Member Function Documentation

2.3.2.1 fit_maxwellians()

```
def py_scripts.hot_elec_calc.hot_electron.fit_maxwellians (
              self,
              n = 3,
              plot = False,
              log = False )
```

fit_maxwellians

Fit Maxwellian like distributions to split distribution

Parameters

self	The object pointer
n	Number of segments (default is 3)
plot	(Logical) Plot distribution
log	(Logical) Set y-axis scaling to log

2.3.2.2 get_energy_frac()

```
def py_scripts.hot_elec_calc.hot_electron.get_energy_frac (
              self )
```

get_energy_frac

Estimates the fraction of outgoing energy being due to hot electrons. Taken to be the ratio of the area under the curve of the hot electron region to the area of the whole distribution. The hot electron region is assumed to be zero if distribution never signifcantly deviates from the initial Maxwellian.

Parameters

self	The object pointer
n	Number of segments (default is 3)
plot	(Logical) Calculate average value across various number of fits
plot	(Logical) Plot T_hot vs number of fits

2.3.2.3 get_flux_dist()

```
def py_scripts.hot_elec_calc.hot_electron.get_flux_dist (
              self,
              plot = False,
              log = False )
```

get_flux_dist

Produce energy distribution/histogram of outgoing electron flux

Parameters

self	The object pointer
plot	(Logical) Plot distribution
log	(Logical) Set y-axis scaling to log

2.3.2.4 get_hot_e_temp()

```
def py_scripts.hot_elec_calc.hot_electron.get_hot_e_temp (
              self,
              n = 3,
              av = True,
              plot = False )
```

get_hot_e_temp

Estimate singular hot electron temperature from weighted average of T from the Maxwellian fits, where the weights correspond to the found amplitude.

Parameters

self	The object pointer
n	Number of segments (default is 3)
plot	(Logical) Calculate average value across various number of fits
plot	(Logical) Plot T_hot vs number of fits

2.3.2.5 split_dist()

```
def py_scripts.hot_elec_calc.hot_electron.split_dist (
              self,
              n = 3,
              plot = False,
              log = False )
```

split_dist

Split found distribution into similar sized segments

Parameters

self	The object pointer
n	Number of segments (default is 3)
plot	(Logical) Plot distribution
log	(Logical) Set y-axis scaling to log

The documentation for this class was generated from the following file:

- /home/u1706745/summer_project/epoch_surra/py_scripts/hot_elec_calc.py

2.4 py_scripts.plasma_calc.Laser_Plasma_Params Class Reference

Laser_Plasma_Params Class.

Public Member Functions

- def __init__ (self, dir)

  init
- def read_data (self)

  read_data
- def get_spatio_temporal (self, mic=False)

  get_spatio_temporal
- def get_plasma_param (self)

  get_plasma_param
- def get_matching_conds (self, ne)

  get_matching_conds
- def get_srs_phase_vel (self, ne)

  get_srs_phase_vel

Public Attributes

- **directory**
- **intensity**
- **wavelength**
- **timesteps**
- **omega0**
- **critical_density**
- **k0_vac**
- **ppc**
- **Ln**
- **grid_data**
- **field_data_0**
- **field_data_1**
- **data_final**
- **grid**
- **nodes**
- **dx**
- **Lx**
- **nx**
- **dt**
- **t_end**
- **time**
- **k_space**
- **omega_space**
- **ne_data**
- **ne**
- **ne_min**
- **ne_max**
- **omega_pe_data**
- **omega_pe**
- **omega_pe_min**
- **omega_pe_max**
- **k0**
- **Te_data**
- **Te**
- **Te_kelvin**
- **v_th**
- **deb_len**
- **k_epw_bs**
- **omega_epw_bs**
- **k_bs**
- **omega_bs**
- **omega_bs_norm**
- **k_bs_norm**
- **k_epw_bs_norm**
- **k_epw_fs**
- **omega_epw_fs**
- **k_fs**
- **omega_fs**
- **omega_fs_norm**
- **k_fs_norm**
- **k_epw_fs_norm**
- **v_phase**

2.4.1 Detailed Description

Laser_Plasma_Params Class.

Class that reads and calculates plasama and grid quantities from grid_ output files.

2.4.2 Constructor & Destructor Documentation

2.4.2.1 __init__()

```
def py_scripts.plasma_calc.Laser_Plasma_Params.__init__ (
              self,
              dir )
```

init

The constructor

Parameters

self	The object pointer
dir	Directory where data is stored (str)

2.4.3 Member Function Documentation

2.4.3.1 get_matching_conds()

```
def py_scripts.plasma_calc.Laser_Plasma_Params.get_matching_conds (
              self,
              ne )
```

get_matching_conds

Calculates SRS scattered wavenumber and frequency

Parameters

self	The object pointer
ne	Electron number density

2.4.3.2 get_plasma_param()

```
def py_scripts.plasma_calc.Laser_Plasma_Params.get_plasma_param (
              self )
```

get_plasma_param

Calculates plasma parameters/variables

Parameters

self	The object pointer

2.4.3.3 get_spatio_temporal()

```
def py_scripts.plasma_calc.Laser_Plasma_Params.get_spatio_temporal (
              self,
              mic = False )
```

get_spatio_temporal

Reads the initial grid data and other files required to find sim data

Parameters

self	The object pointer
mic	(Logical) Output grid in microns

2.4.3.4 get_srs_phase_vel()

```
def py_scripts.plasma_calc.Laser_Plasma_Params.get_srs_phase_vel (
              self,
              ne )
```

get_srs_phase_vel

Calculates SRS (backscatter) phase velocity at n = ne

Parameters

self	The object pointer
ne	Electron number density

2.4.3.5 read_data()

```
def py_scripts.plasma_calc.Laser_Plasma_Params.read_data (
              self )
```

read_data

Reads the initial grid data and other files required to find sim data

Parameters

self	The object pointer

The documentation for this class was generated from the following file:

- /home/u1706745/summer_project/epoch_surra/py_scripts/plasma_calc.py

Index