C-Mod Analysis Documentation - v1

Marco A. Miller

July 2024

1 data_access.py

This file contains many functions that have been stripped from Mark Chilenski's profiletools and eqtools packages primarily. They will be re-built in the future using the xarray framework.

Calls to data_access.py are made in the following files:

- `cmod_tools.py`
- `power_balance.py`
- `single_shot.py`

Instantiation of data_access objects in each file

`cmod_tools.py`

- `e = data_access.CModEFITTree(int(shot), tree='EFIT20', length_unit='m')`
- `e = data_access.CModEFITTree(int(shot), tree='analysis', length_unit='m')`
- `p_Te = data_access.Te(int(shot), include=['ETS'], abscissa='sqrtpsinorm', t_min = tmin, t_max=tmax, efit_tree=e)`
- `p_ne = data_access.Te(int(shot), include=['ETS'], abscissa='sqrtpsinorm', t_min = tmin, t_max=tmax, efit_tree=e)`
- `p_Te = data_access.Te(int(shot), include=['CTS'], abscissa='sqrtpsinorm', t_min = tmin, t_max=tmax, efit_tree=e)`
- `p_ne = data_access.Te(int(shot), include=['CTS'], abscissa='sqrtpsinorm', t_min = tmin, t_max=tmax, efit_tree=e)`
- `data_access.BivariatePlasmaProfile(X_dim=1, X_units=[''], y_units='kPa', X_labels= ['$sqrtpsinorm$'], y_label=r'$p_e$')`

`power_balance.py`

- `e = data_access.CModEFITTree(int(shot), tree='EFIT20', length_unit='m')`
- `e = data_access.CModEFITTree(int(shot), tree='analysis', length_unit='m')`

`single_shot.py`

- `e = data_access.CModEFITTree(int(shot), tree='EFIT20', length_unit='m')`
- `e = data_access.CModEFITTree(int(shot), tree='analysis', length_unit='m')`

Classes/functions needed: `CModEFITTree()`, `ne()`, `Te()`, `BivariatePlasmaProfile()`

Calls to sub-functions needed for instantiation

`CModEFITTree()`:

Contains:

- `__init__()`
- `getFluxVol()`

Calls:

- `EFITTree()` [*class*]
- `EFITTree().__init__()` [*function*]
  - Calls outlined below
- `self.getFluxVol()` [*function*]
  - `self._getLengthConversionFactor()` [*function*] (belongs to `Equilibrium()`)

`EFITTree()`:

Contains:

- `__init__()`
- `getTimeBase()`
- `getFluxGrid()`
- `getFluxAxis()`
- `getVolLCFS()`
- `getQProfile()`
- `getRmidPsi()`

Calls:

- `Equilibrium()` [*class*]
- `self.getTimeBase()` [*function*]
- `self.getFluxGrid()` [*function*]
- `self.getFluxAxis()` [*function*]
- `self.getVolLCFS()` [*function*]
- `self.getQProfile()` [*function*]
- `self.getRmidPsi()` [*function*]

`Equilibrium()`:

Contains:

- `__init__()`

`ne()`:

Calls:

- `CModEFITTree()` [*class*]
- `neETS()` [*function*]
  - `BivariatePlasmaProfile()` [*class*]
  - `CModEFITTree()` [*class*]
  - `self.add_data()` [*function*] (belongs to `Profile()`)
  - `self.remove_points()` [*function*] (belongs to `Profile()`)
  - `self.convert_abscissa()` [*function*] (belongs to `BivariatePlasmaProfile()`)
- `neCTS()` [*function*]
  - `BivariatePlasmaProfile()` [*class*]
  - `CModEFITTree()` [*class*]
  - `self.add_data()` [*function*] (belongs to `Profile()`)
  - `self.remove_points()` [*function*] (belongs to `Profile()`)
  - `self.convert_abscissa()` [*function*] (belongs to `BivariatePlasmaProfile()`)
- `BivariatePlasmaProfile.add_profile()` [*function*]

`Te()`:

Calls:

- `CModEFITTree()` [*class*]
- `TeETS()` [*function*]
  - `BivariatePlasmaProfile()` [*class*]
  - `CModEFITTree()` [*class*]
  - `self.add_data()` [*function*] (belongs to `Profile()`)
  - `self.remove_points()` [*function*] (belongs to `Profile()`)
  - `self.convert_abscissa()` [*function*] (belongs to `BivariatePlasmaProfile()`)
- `TeCTS()` [*function*]
  - `BivariatePlasmaProfile()` [*class*]
  - `CModEFITTree()` [*class*]
  - `self.add_data()` [*function*] (belongs to `Profile()`)
  - `self.remove_points()` [*function*] (belongs to `Profile()`)
  - `self.convert_abscissa()` [*function*] (belongs to `BivariatePlasmaProfile()`)
- `BivariatePlasmaProfile.add_profile()` [*function*]

`BivariatePlasmaProfile()`:

Contains:

- `remake_efit_tree()`

- `convert_abscissa()`
- `add_profile()`
- `drop_axis()`

Calls:

- `Profile()` [*class*]
- `Profile().__init__()` [*class*]

`Profile()`:

Contains:

- `__init__()`
- `add_data()`
- `add_profile()`
- `remove_points()`
- `drop_axis()`
- `plot_data()`

Calls:

- `Channel` [*class*]

`BivariatePlasmaProfile.add_profile()`:

Calls:

- `other.convert_absissa()` [*function*] (belongs to `BivariatePlasmaProfile()`)
- `Profile.add_profile()` [*function*]

`BivariatePlasmaProfile.convert_abscissa()`:

Calls:

- `self.efit_tree.rz2rho()` [*function*] (belongs to `Equilibrium()`)
    - `self.rz2psinorm()` [*function*] (belongs to `Equilibrium()`)
- `self.efit_tree.rho2rho()` [*function*] (belongs to `Equilibrium()`)
    - `self.rmid2rho()` [*function*] (belongs to `Equilibrium()`)
    - `self.psinorm2rho()` [*function*] (belongs to `Equilibrium()`)

`Equilibrium.rz2psinorm()`:

Calls:

- `self.rz2psi()` [*function*] (belongs to `Equilibrium()`)
- `getFluxLCFS()` [*function*] (belongs to `EFITTree()`)
- `getFluxAxis()` [*function*] (belongs to `EFITTree()`)

`Equilibrium.rmid2rho()`:

Calls:

- `self.rmid2psinorm()` [*function*] (belongs to `Equilibrium()`)

`Equilibrium.psinorm2rho()`:

Calls:

- `self.psinorm2rmid()` [*function*] (belongs to `Equilibrium()`)

`Equilibrium.rz2psi()`:

Calls:

- `self._processRZt()` [*function*]
  - `self._getLengthConversionFactor()` [*function*]
  - `self._checkRZ()` [*function*]
    - `getRGrid()` [*function*] (belongs to `EFITTree()`)
    - `getZGrid()` [*function*] (belongs to `EFITTree()`)
  - `getTimeBase()` [*function*] (belongs to `EFITTree()`)
  - `_getNearestIdx()` [*function*]
- `self._getFluxBiSpline()` [*function*]
  - `getRGrid()` [*function*]
  - `getZGrid()` [*function*]
  - `getFluxGrid()` [*function*]
- `self.getCurrentSign()` [*function*] (belongs to `EFITTree()`)
  - `self.getIpMeas()` [*function*] (belongs to `EFITTree()`)

`Equilibrium.rmid2psinorm()`:

Calls:

- `self._psinorm2Quan()` [*function*]
  - `self._processRZt()` [*function*]
- `self._getRmidToPsiNormSpline()` [*function*]
  - `getRGrid()` [*function*]
  - `rz2psinorm()` [*function*]
  - `getMagZ()` [*function*] (belongs to `EFITTree()`)
  - `getTimeBase()` [*function*]
  - `UnivariateInterpolator()` [*class*]
  - `BivariateInterpolator()` [*class*]

`Equilibrium.psinorm2rmid()`:

Calls:

- `self._getLengthConversionFactor()` [*function*]
- `self._psinorm2Quan()` [*function*]
- `self._getRmidSpline` [*function*]
  - `getMagR()` [*function*] (belongs to `EFITTree()`)
    - `_getLengthConversionFactor()` [*function*]
  - `getRGrid()` [*function*]
  - `rz2psinorm()` [*function*]
  - `getMagZ()` [*function*]
  - `getTimeBase()` [*function*]
  - `UnivariateInterpolator()` [*class*]
  - `BivariateInterpolator()` [*class*]

`UnivariateInterpolator()`:

Contains:

- `__init__()`

`BivariateInterpolator()`:

Contains:

- `__init__()`

`Channel()`:

Contains:

- `__init__()`

Other calls to object functions in each file

`cmod_tools.py`

- `e.rho2rho()`
- `p_[ne/Te].add_profile()`
- `p_[ne/Te].drop_axis()`
- `p_[ne/Te].remove_points()`
- `p_[ne/Te].remake_efit_tree()`
- `p_[ne/Te].add_data()`
- `p_[ne/Te].plot_data()`

`BivariatePlasmaProfile.drop_axis()`:

Calls:

- `Profile.drop_axis()` [*function*]

BivariatePlasmaProfile.remake_efit_tree():

Calls:

- CModEFITTree() [*class*]

Profile.plot_data(): no calls

power_balance.py

- e.rho2rho()
- e.rz2BT()
- e.rz2BZ()

rz2BT():

Calls:

- self._getLengthConversionFactor() [*function*]
- self.rz2F() [*function*] (belongs to Equilibrium())
  - self._RZ2Quan() [*function*] (belongs to Equilibrium())
  - self._getFSpline() [*function*] (belongs to Equilibrium())
- self.getBtVac() [*function*] (belongs to EFITTree())
- self.getMagR() [*function*]

rz2BZ():

Calls:

- self._processRZt() [*function*]
- self._getFluxBiSpline() [*function*]
- self.getCurrentSign() [*function*]

_RZ2Quan():

Calls:

- self.rz2psinorm() [*function*]
- self._psinorm2Quan() [*function*]

_getFSpline():

Calls:

- self.getF() [*function*]
- UnivariateInterpolator() [*class*]

single_shot.py

- e.rho2rho()