

SPHERLS Python Scripts

1.0

Generated by Doxygen 1.7.6.1

Tue Apr 9 2013 00:58:20

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	3
2.1	File List	3
3	Class Documentation	5
3.1	plot_file.Axis Class Reference	5
3.1.1	Detailed Description	5
3.1.2	Constructor & Destructor Documentation	5
3.1.2.1	__init__	5
3.1.3	Member Function Documentation	6
3.1.3.1	load	6
3.2	plot_profile.Axis Class Reference	6
3.2.1	Detailed Description	6
3.2.2	Constructor & Destructor Documentation	7
3.2.2.1	__init__	7
3.2.3	Member Function Documentation	7
3.2.3.1	load	7
3.3	calculate_residuals_of_light_curve_fit.Bin Class Reference	7
3.3.1	Constructor & Destructor Documentation	8
3.3.1.1	__init__	8
3.3.2	Member Function Documentation	8
3.3.2.1	addPoint	8
3.3.2.2	getMean	8
3.3.2.3	getSTDD	8

3.4	calculate_residuals_of_light_curve_fit.BinnedData Class Reference	9
3.4.1	Member Function Documentation	9
3.4.1.1	addEvenBins	9
3.4.1.2	binData	9
3.4.1.3	getBinCenters	9
3.4.1.4	getMean	10
3.4.1.5	getSTDD	10
3.5	plot_file.Curve Class Reference	10
3.5.1	Detailed Description	11
3.5.2	Constructor & Destructor Documentation	11
3.5.2.1	__init__	11
3.5.3	Member Function Documentation	11
3.5.3.1	load	11
3.6	plot_profile.Curve Class Reference	12
3.6.1	Detailed Description	12
3.6.2	Constructor & Destructor Documentation	12
3.6.2.1	__init__	12
3.6.3	Member Function Documentation	13
3.6.3.1	load	13
3.7	datafile.DataFile Class Reference	13
3.7.1	Detailed Description	13
3.7.2	Member Function Documentation	14
3.7.2.1	readFile	14
3.7.2.2	readFileFixed	14
3.7.2.3	readFileUnFixed	14
3.8	calculate_residuals_of_light_curve_fit.DataFunction Class Reference	14
3.9	plot_file.DataSet Class Reference	15
3.9.1	Detailed Description	15
3.9.2	Constructor & Destructor Documentation	15
3.9.2.1	__init__	15
3.9.3	Member Function Documentation	15
3.9.3.1	getCurve	15
3.9.3.2	load	16
3.10	plot_profile.DataSet Class Reference	16

3.10.1 Detailed Description	16
3.10.2 Constructor & Destructor Documentation	16
3.10.2.1 __init__	16
3.10.3 Member Function Documentation	17
3.10.3.1 getCurve	17
3.10.3.2 load	17
3.11 dump.dump Class Reference	17
3.11.1 Constructor & Destructor Documentation	18
3.11.1.1 __init__	18
3.11.2 Member Function Documentation	18
3.11.2.1 printHeader	18
3.11.2.2 printVar	19
3.11.2.3 read	19
3.11.2.4 readAsciiVar	19
3.11.2.5 readBinaryVar	19
3.11.2.6 readHeader	19
3.11.2.7 readHeaderAscii	20
3.11.2.8 readHeaderBinary	20
3.11.2.9 setVarIDs	20
3.12 eos_interp.eosTable Class Reference	20
3.12.1 Detailed Description	21
3.12.2 Constructor & Destructor Documentation	21
3.12.2.1 __init__	21
3.12.3 Member Function Documentation	21
3.12.3.1 interpolate	21
3.12.3.2 load	22
3.12.3.3 plotLogE	22
3.12.3.4 plotLogP	22
3.12.3.5 write	22
3.13 eos_interp.eosTableManager Class Reference	23
3.13.1 Detailed Description	23
3.13.2 Constructor & Destructor Documentation	23
3.13.2.1 __init__	23
3.13.3 Member Function Documentation	24

3.13.3.1	getTableFromComp	24
3.13.3.2	interpComp	24
3.13.3.3	load	24
3.13.3.4	plotGrid	24
3.14	plot_2DSlices.File2DSlice Class Reference	24
3.14.1	Member Function Documentation	25
3.14.1.1	load	25
3.15	make_hdf.fileSet Class Reference	25
3.15.1	Constructor & Destructor Documentation	26
3.15.1.1	__init__	26
3.15.2	Member Function Documentation	26
3.15.2.1	convertDumpToHDF	26
3.15.2.2	makeHDFFiles	26
3.16	make_hdf2.fileSet Class Reference	27
3.16.1	Constructor & Destructor Documentation	27
3.16.1.1	__init__	27
3.16.2	Member Function Documentation	28
3.16.2.1	convertDumpToHDF	28
3.16.2.2	makeHDFFiles	28
3.17	make_hdf.hdfFile Class Reference	28
3.17.1	Member Function Documentation	29
3.17.1.1	write	29
3.18	eos_interp.interpTable Class Reference	29
3.18.1	Detailed Description	29
3.18.2	Constructor & Destructor Documentation	30
3.18.2.1	__init__	30
3.18.3	Member Function Documentation	30
3.18.3.1	interpolate	30
3.18.3.2	plotLogE	30
3.18.3.3	plotLogK	31
3.18.3.4	plotLogP	31
3.18.3.5	read	31
3.19	eos_interp.interpTableManager Class Reference	32
3.19.1	Constructor & Destructor Documentation	32

3.19.1.1	<code>__init__</code>	32
3.19.2	Member Function Documentation	32
3.19.2.1	<code>createTables</code>	32
3.20	<code>make_hdf.interpVar</code> Class Reference	32
3.21	<code>light_curve.LightCurve</code> Class Reference	33
3.21.1	Member Function Documentation	34
3.21.1.1	<code>calculateCurve</code>	34
3.21.1.2	<code>readBoloCorr</code>	34
3.21.1.3	<code>readProfiles</code>	34
3.21.1.4	<code>write</code>	34
3.22	<code>eos_interp.opacityTable</code> Class Reference	34
3.22.1	Detailed Description	35
3.22.2	Constructor & Destructor Documentation	35
3.22.2.1	<code>__init__</code>	35
3.22.3	Member Function Documentation	35
3.22.3.1	<code>fillInDepNans</code>	35
3.22.3.2	<code>interpolate</code>	36
3.22.3.3	<code>load</code>	36
3.22.3.4	<code>plotLogK</code>	36
3.23	<code>eos_interp.opacityTableManager</code> Class Reference	37
3.23.1	Detailed Description	37
3.23.2	Constructor & Destructor Documentation	37
3.23.2.1	<code>__init__</code>	37
3.23.3	Member Function Documentation	38
3.23.3.1	<code>getTableFromComp</code>	38
3.23.3.2	<code>interpComp</code>	38
3.23.3.3	<code>load</code>	38
3.23.3.4	<code>plotGrids</code>	38
3.24	<code>work_plot.PdVPlotSettings</code> Class Reference	38
3.25	<code>plot_file.Plot</code> Class Reference	39
3.25.1	Detailed Description	39
3.25.2	Constructor & Destructor Documentation	40
3.25.2.1	<code>__init__</code>	40
3.25.3	Member Function Documentation	40

3.25.3.1	load	40
3.26	plot_profile.Plot Class Reference	40
3.26.1	Detailed Description	40
3.26.2	Constructor & Destructor Documentation	41
3.26.2.1	__init__	41
3.26.3	Member Function Documentation	41
3.26.3.1	load	41
3.26.3.2	setLimits	41
3.27	work_plot.Settings Class Reference	41
3.27.1	Constructor & Destructor Documentation	42
3.27.1.1	__init__	42
3.27.2	Member Function Documentation	42
3.27.2.1	parseXML	42
3.28	plot_file.Text Class Reference	43
3.28.1	Detailed Description	43
3.28.2	Constructor & Destructor Documentation	43
3.28.2.1	__init__	43
3.29	make_hdf.variable Class Reference	43
3.30	work_plot.WorkPlotSettings Class Reference	44
4	File Documentation	45
4.1	average_PKE.py File Reference	45
4.1.1	Detailed Description	45
4.2	combine_bins.py File Reference	45
4.2.1	Detailed Description	45
4.3	combine_bins_persistent.py File Reference	45
4.3.1	Detailed Description	45
4.4	compare_sedov_blasts.py File Reference	46
4.4.1	Detailed Description	46
4.5	cp_files.py File Reference	46
4.5.1	Detailed Description	46

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

plot_file.Axis	5
plot_profile.Axis	6
calculate_residuals_of_light_curve_fit.Bin	7
calculate_residuals_of_light_curve_fit.BinnedData	9
plot_file.Curve	10
plot_profile.Curve	12
datafile.DataFile	13
calculate_residuals_of_light_curve_fit.DataFunction	14
plot_file.DataSet	15
plot_profile.DataSet	16
dump.dump	17
eos_interp.eosTable	20
eos_interp.eosTableManager	23
plot_2DSlices.File2DSlice	24
make_hdf.fileSet	25
make_hdf2.fileSet	27
make_hdf.hdfFile	28
eos_interp.interpTable	29
eos_interp.interpTableManager	32
make_hdf.interpVar	32
light_curve.LightCurve	33
eos_interp.opacityTable	34
eos_interp.opacityTableManager	37
work_plot.PdVPlotSettings	38
plot_file.Plot	39
plot_profile.Plot	40
work_plot.Settings	41
plot_file.Text	43
make_hdf.variable	43

[work_plot.WorkPlotSettings](#) 44

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

average_PKE.py	45
combine_bins.py	45
combine_bins_persistent.py	45
compare_sedov_blasts.py	46
cp_files.py	46

Chapter 3

Class Documentation

3.1 `plot_file.Axis` Class Reference

Public Member Functions

- `def __init__`
- `def load`

Public Attributes

- `plots`
- `xlabel`
- `limits`
- `bMinorTics`
- `ticks`
- `grid`
- `plotHeightWeights`

3.1.1 Detailed Description

This class holds all the information needed for a particular x-axis.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 `def plot_file.Axis.__init__(self, element, options)`

This function initializes the axis object.

References `plot_file.Plot.bMinorTics`, `plot_file.Axis.bMinorTics`, `plot_profile.Curve.bTime`, `plot_profile.Axis.bTime`, `plot_file.Plot.grid`, `plot_file.Axis.grid`, `plot_file.Plot.limits`,

plot_file.Axis.limits, plot_profile.Axis.period, plot_file.Axis.plotHeightWeights, plot_file.-Axis.plots, plot_file.Plot.ticks, plot_file.Axis.ticks, and plot_file.Axis.xlabel.

3.1.3 Member Function Documentation

3.1.3.1 `def plot_file.Axis.load(self, files, options)`

This function loads the values needed for the x-axis data from the `fileData` argument

References `plot_file.Axis.plots`.

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

- `plot_file.py`

3.2 `plot_profile.Axis` Class Reference

Public Member Functions

- `def __init__`
- `def load`

Public Attributes

- `plots`
- `bTime`
- `period`
- `nColumn`
- `xlabel`
- `x`
- `formulaOrig`
- `formula`
- `phase`
- `code`
- `limits`
- `bMinorTics`
- `grid`

3.2.1 Detailed Description

This class holds all the information needed for a particular x-axis. An axis can either be either of time, or of some column in the data files.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 `def plot_profile.Axis.__init__(self, element, options)`

This function initializes the axis object.

References `plot_file.Plot.bMinorTics`, `plot_profile.Plot.bMinorTics`, `plot_file.Axis.bMinorTics`, `plot_profile.Axis.bMinorTics`, `plot_profile.Curve.bTime`, `plot_profile.Axis.bTime`, `plot_profile.Curve.code`, `plot_profile.Axis.code`, `plot_profile.Axis.formula`, `make_hdf.variable.formula`, `make_hdf.interpVar.formula`, `plot_profile.Curve.formulaOrig`, `plot_profile.Axis.formulaOrig`, `plot_file.Plot.grid`, `plot_profile.Plot.grid`, `plot_file.Axis.grid`, `plot_profile.Axis.grid`, `plot_file.Plot.limits`, `plot_profile.Plot.limits`, `plot_file.Axis.limits`, `plot_profile.Axis.limits`, `plot_profile.Curve.nColumn`, `plot_profile.Axis.nColumn`, `plot_profile.Axis.period`, `plot_profile.Axis.phase`, `plot_file.Axis.plots`, `plot_profile.Axis.plots`, `plot_file.Text.x`, `plot_file.Curve.x`, `calculate_residuals_of_light_curve_fit.DataFunction.x`, `plot_profile.Axis.x`, `plot_file.Axis.xlabel`, and `plot_profile.Axis.xlabel`.

3.2.3 Member Function Documentation

3.2.3.1 `def plot_profile.Axis.load(self, fileData, options, dataSet, nFileCount)`

This function loads the values needed for the x-axis data from the `fileData` argument

References `plot_profile.Curve.bTime`, `plot_profile.Axis.bTime`, `plot_profile.Curve.code`, `plot_profile.Axis.code`, `plot_profile.Curve.formulaOrig`, `plot_profile.Axis.formulaOrig`, `plot_file.Plot.limits`, `plot_profile.Plot.limits`, `plot_file.Axis.limits`, `plot_profile.Axis.limits`, `plot_profile.Curve.nColumn`, `plot_profile.Axis.nColumn`, `plot_profile.Axis.period`, `plot_file.Axis.plots`, `plot_profile.Axis.plots`, `plot_file.Text.x`, `plot_file.Curve.x`, `calculate_residuals_of_light_curve_fit.DataFunction.x`, `plot_profile.Axis.x`, `plot_file.Axis.xlabel`, and `plot_profile.Axis.xlabel`.

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

- `plot_profile.py`

3.3 calculate_residuals_of_light_curve_fit.Bin Class Reference

Public Member Functions

- `def __init__`
- `def addPoint`
- `def getMean`
- `def getSTDD`

Public Attributes

- `mean`

- `sigma`
- `lowerBound`
- `upperBound`
- `center`
- `dataAddedSinceMeanCal`
- `dataAddedSinceSigmaCal`
- `points`

3.3.1 Constructor & Destructor Documentation

3.3.1.1 `def calculate_residuals_of_light_curve_fit.Bin.__init__(self, lowerBound, upperBound)`

Set bounds of the bin and the center of the bin

References `calculate_residuals_of_light_curve_fit.Bin.center`, `calculate_residuals_of_light_curve_fit.Bin.dataAddedSinceMeanCal`, `calculate_residuals_of_light_curve_fit.Bin.dataAddedSinceSigmaCal`, `calculate_residuals_of_light_curve_fit.Bin.lowerBound`, `calculate_residuals_of_light_curve_fit.Bin.mean`, `calculate_residuals_of_light_curve_fit.Bin.points`, `calculate_residuals_of_light_curve_fit.Bin.sigma`, and `calculate_residuals_of_light_curve_fit.Bin.upperBound`.

3.3.2 Member Function Documentation

3.3.2.1 `def calculate_residuals_of_light_curve_fit.Bin.addPoint(self, x, y)`

Add a point to the bin

References `calculate_residuals_of_light_curve_fit.Bin.dataAddedSinceMeanCal`, and `calculate_residuals_of_light_curve_fit.Bin.dataAddedSinceSigmaCal`.

3.3.2.2 `def calculate_residuals_of_light_curve_fit.Bin.getMean(self)`

Returns the mean of the bin, calculating if needed

References `calculate_residuals_of_light_curve_fit.Bin.dataAddedSinceMeanCal`, `calculate_residuals_of_light_curve_fit.Bin.lowerBound`, `calculate_residuals_of_light_curve_fit.Bin.mean`, `calculate_residuals_of_light_curve_fit.Bin.points`, and `calculate_residuals_of_light_curve_fit.Bin.upperBound`.

Referenced by `calculate_residuals_of_light_curve_fit.Bin.getSTDD()`.

3.3.2.3 `def calculate_residuals_of_light_curve_fit.Bin.getSTDD(self)`

Returns the standard deviation calculating if needed

References `calculate_residuals_of_light_curve_fit.Bin.dataAddedSinceSigmaCal`, `calculate_residuals_of_light_curve_fit.Bin.getMean()`, `calculate_residuals_of_light_curve_fit.Bin.points`, and `calculate_residuals_of_light_curve_fit.Bin.sigma`.

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

- `calculate_residuals_of_light_curve_fit.py`

3.4 calculate_residuals_of_light_curve_fit.BinnedData Class - Reference

Public Member Functions

- `def __init__`
- `def addEvenBins`
- `def binData`
- `def getMean`
- `def getBinCenters`
- `def getSTDD`

Public Attributes

- `bins`

3.4.1 Member Function Documentation

3.4.1.1 `def calculate_residuals_of_light_curve_fit.BinnedData.addEvenBins(self, domainMin, domainMax, numBins)`

Sets the number and edges of the bins

3.4.1.2 `def calculate_residuals_of_light_curve_fit.BinnedData.binData(self, data)`

Puts points into bins

References `calculate_residuals_of_light_curve_fit.BinnedData.bins`.

3.4.1.3 `def calculate_residuals_of_light_curve_fit.BinnedData.getBinCenters(self)`

Returns a list of bin centers

References `calculate_residuals_of_light_curve_fit.BinnedData.bins`.

3.4.1.4 `def calculate_residuals_of_light_curve_fit.BinnedData.getMean(self)`

Returns a list of the mean values in each bin

References `calculate_residuals_of_light_curve_fit.BinnedData.bins`.

3.4.1.5 `def calculate_residuals_of_light_curve_fit.BinnedData.getSTDD(self)`

Returns a list of standard deviations of each bin

References `calculate_residuals_of_light_curve_fit.BinnedData.bins`.

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

- `calculate_residuals_of_light_curve_fit.py`

3.5 `plot_file.Curve` Class Reference

Public Member Functions

- `def __init__`
- `def load`

Public Attributes

- `nColumnX`
- `nColumnY`
- `nColumnErr`
- `y`
- `x`
- `error`
- `index`
- `formulaOrigY`
- `formulaOrigX`
- `formulaOrigErr`
- `formulaX`
- `formulaY`
- `formulaErr`
- `codeY`
- `codeX`
- `codeErr`
- `style`
- `color`
- `markerfacecolor`
- `markeredgeicolor`

- **markersize**
- **linewidth**
- **label**
- **fileReference**
- **nRowShiftErr**
- **nRowShiftX**
- **nRowShiftY**
- **marker**
- **ecolor**
- **elinewidth**
- **capsize**

3.5.1 Detailed Description

This class holds all the information for a curve on a plot.

3.5.2 Constructor & Destructor Documentation

3.5.2.1 `def plot_file.Curve.__init__(self, element)`

This method initializes a curve object, the type parameter allows checking curve syntax against axis syntax to see if they match.

References `plot_file.Curve.capsize`, `plot_file.Curve.codeErr`, `plot_file.Curve.codeX`, `plot_file.Curve.codeY`, `plot_file.Curve.color`, `plot_file.Curve.ecolor`, `plot_file.Curve.elinewidth`, `plot_file.Curve.error`, `plot_file.Curve.fileReference`, `plot_file.Curve.formulaErr`, `plot_file.Curve.formulaOrigErr`, `plot_file.Curve.formulaOrigX`, `plot_file.Curve.formulaOrigY`, `plot_file.Curve.formulaX`, `plot_file.Curve.formulaY`, `plot_2DSlices.File2DSlice.index`, `plot_file.Curve.index`, `plot_file.Curve.label`, `plot_file.Curve.linewidth`, `plot_file.Curve.marker`, `plot_file.Curve.markeredgcolor`, `plot_file.Curve.markerfacecolor`, `plot_file.Curve.markersize`, `plot_file.Curve.nColumnErr`, `plot_file.Curve.nColumnX`, `plot_file.Curve.nColumnY`, `plot_file.Curve.nRowShiftErr`, `plot_file.Curve.nRowShiftX`, `plot_file.Curve.nRowShiftY`, `plot_file.Curve.style`, `plot_file.Text.x`, `plot_file.Curve.x`, `calculate_residuals_of_light_curve_fit.DataFunction.x`, `plot_file.Text.y`, `plot_file.Curve.y`, and `calculate_residuals_of_light_curve_fit.DataFunction.y`.

3.5.3 Member Function Documentation

3.5.3.1 `def plot_file.Curve.load(self, files, options)`

This method adds a y value and index to the curve for the current fileData.

References `plot_file.Curve.codeErr`, `plot_file.Curve.codeX`, `plot_file.Curve.codeY`, `plot_file.Curve.fileReference`, `plot_file.Curve.nColumnErr`, `plot_file.Curve.nColumnX`, `plot_file.Curve.nColumnY`, `plot_file.Curve.nRowShiftErr`, `plot_file.Curve.nRowShiftX`, and `plot_file.Curve.nRowShiftY`.

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

- `plot_file.py`

3.6 `plot_profile.Curve` Class Reference

Public Member Functions

- `def __init__`
- `def load`

Public Attributes

- `nColumn`
- `zone`
- `nCurveIDForZoneRef`
- `y`
- `index`
- `bTime`
- `formulaOrig`
- `code`
- `style`
- `color`
- `markersize`
- `linewidth`
- `testZoneAdjust`
- `label`
- `ID`
- `indexOfLastFileLoad`

3.6.1 Detailed Description

This class holds all the information for a curve on a plot.

3.6.2 Constructor & Destructor Documentation

3.6.2.1 `def plot_profile.Curve.__init__(self, element, type, curveIndex)`

This method initializes a curve object, the `type` parameter allows checking curve syntax against axis syntax to see if they match.

References `plot_profile.Curve.bTime`, `plot_profile.Curve.code`, `plot_profile.Curve.color`, `plot_file.Curve.color`, `plot_profile.Axis.formula`, `make_hdf.variable.formula`, `make_hdf.interpVar.formula`, `plot_profile.Curve.formulaOrig`, `plot_profile.Curve.ID`, `plot_2DSlices.File2DSlice.index`, `plot_profile.Curve.index`, `plot_file.Curve.index`, `plot_profile.Curve.indexOfLastFileLoad`, `plot_profile.Curve.label`, `plot_file.Curve.label`, `plot_profile.Curve-`

linewidth, plot_file.Curve.linewidth, plot_profile.Curve.markersize, plot_file.Curve.-markersize, plot_profile.Curve.nColumn, plot_profile.Curve.nCurveIDForZoneRef, plot_profile.Curve.style, plot_file.Curve.style, plot_profile.Curve.testZoneAdjust, plot_file.-Text.y, plot_profile.Curve.y, plot_file.Curve.y, calculate_residuals_of_light_curve_fit.-DataFunction.y, and plot_profile.Curve.zone.

3.6.3 Member Function Documentation

3.6.3.1 def plot_profile.Curve.load(self, fileData, options, dataSet, nFileCount)

This method adds a y value and index to the curve for the current fileData.

References plot_profile.Curve.bTime, plot_profile.Curve.code, plot_profile.Curve.-formulaOrig, plot_profile.Curve.ID, plot_profile.Curve.indexOfLastFileLoad, plot_profile.Curve.nColumn, plot_profile.Curve.testZoneAdjust, and plot_profile.Curve.zone.

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

- plot_profile.py

3.7 datafile.DataFile Class Reference

Public Member Functions

- def **setFileSize**
- def [readFile](#)
- def [readFileFixed](#)
- def [readFileUnFixed](#)

Public Attributes

- **sFileName**

Static Public Attributes

- **sColumnNames** = None
- **fColumnValues** = None
- **sHeader** = None

3.7.1 Detailed Description

A generic class for holding a file consisting of a header and columns of floats

3.7.2 Member Function Documentation

3.7.2.1 `def datafile.DataFile.readFile(self, sFileName)`

a wrapper to determine which `readFile` function should be used

References `datafile.DataFile.fColumnValues`, `datafile.DataFile.readFileFixed()`, `datafile.DataFile.readFileUnFixed()`, and `datafile.DataFile.sFileName`.

3.7.2.2 `def datafile.DataFile.readFileFixed(self, sFileName)`

Reads in a file when the size has already been set using `\ref setFileSize`, or by a previous file read using `\ref readFileUnFixed`.

References `datafile.DataFile.fColumnValues`, `datafile.DataFile.sColumnNames`, and `datafile.DataFile.sHeader`.

Referenced by `datafile.DataFile.readFile()`.

3.7.2.3 `def datafile.DataFile.readFileUnFixed(self, sFileName)`

Reads in a file when the size is not fixed and needs to be determined from the input file being read in

References `datafile.DataFile.fColumnValues`, `datafile.DataFile.sColumnNames`, and `datafile.DataFile.sHeader`.

Referenced by `datafile.DataFile.readFile()`.

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

- `datafile.py`

3.8 `calculate_residuals_of_light_curve_fit.DataFunction` Class - Reference

Public Member Functions

- `def __init__`
- `def getPointByLinearInt`

Public Attributes

- `x`
- `y`
- `maxRange`

- `minRange`

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

- `calculate_residuals_of_light_curve_fit.py`

3.9 `plot_file.DataSet` Class Reference

Public Member Functions

- `def __init__`
- `def load`
- `def getCurve`

Public Attributes

- `axes`
- `files`

3.9.1 Detailed Description

This class holds all the information for a single `dataSet`, which includes the `baseFileName` of the dataset, the range of the `dataSet` (start-end), the times and phases of the files within the range of the `dataSet`, and the plots made from the `dataSet`.

3.9.2 Constructor & Destructor Documentation

3.9.2.1 `def plot_file.DataSet.__init__(self, element, options)`

Initilizes the `dataSet` by setting `baseFileName`, `start`, `end`, and intilizing plots from an `xml` element

References `plot_file.DataSet.axes`, and `plot_file.DataSet.files`.

3.9.3 Member Function Documentation

3.9.3.1 `def plot_file.DataSet.getCurve(self, ID)`

Returns a curve object that has `ID`, `ID`

References `plot_file.DataSet.axes`.

3.9.3.2 `def plot_file.DataSet.load(self, options)`

Loads the dataSet, this means that it sets, time, phases, and plots data

References `plot_file.DataSet.axes`, and `plot_file.DataSet.files`.

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

- `plot_file.py`

3.10 `plot_profile.DataSet` Class Reference

Public Member Functions

- `def __init__`
- `def load`
- `def getCurve`

Public Attributes

- `baseFileName`
- `start`
- `end`
- `axes`
- `nNumFiles`
- `fileIndices`
- `hasNonTimeAxis`
- `eosFile`

3.10.1 Detailed Description

This class holds all the information for a single dataSet, which includes the `baseFileName` of the dataset, the range of the dataSet (start-end), the times and phases of the files within the range of the dataSet, and the plots made from the dataSet.

3.10.2 Constructor & Destructor Documentation

3.10.2.1 `def plot_profile.DataSet.__init__(self, element, options)`

Intilizes the dataSet by setting `baseFileName`, `start`, `end`, and intilizing plots from an xml element

References `plot_file.DataSet.axes`, `plot_profile.DataSet.axes`, `plot_profile.DataSet.baseFileName`, `plot_profile.DataSet.end`, `plot_2DSlices.File2DSlice.eosFile`, `light_curve.LightCurve.eosFile`, `plot_profile.DataSet.eosFile`, `plot_profile.DataSet.fileIndices`, `plot_profile.DataSet.hasNonTimeAxis`, `light_curve.LightCurve.nNumFiles`, `plot_profile.DataSet.nNumFiles`, and `plot_profile.DataSet.start`.

3.10.3 Member Function Documentation

3.10.3.1 `def plot_profile.DataSet.getCurve(self, ID)`

Returns a curve object that has ID, ID

References `plot_file.DataSet.axes`, and `plot_profile.DataSet.axes`.

3.10.3.2 `def plot_profile.DataSet.load(self, options)`

Loads the `dataSet`, this means that it sets, time, phases, and plots data

References `plot_file.DataSet.axes`, `plot_profile.DataSet.axes`, `plot_profile.DataSet.baseFileName`, `plot_profile.DataSet.end`, `plot_2DSlices.File2DSlice.eosFile`, `light_curve.LightCurve.eosFile`, `plot_profile.DataSet.eosFile`, `light_curve.LightCurve.numFiles`, `plot_profile.DataSet.numFiles`, and `plot_profile.DataSet.start`.

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

- `plot_profile.py`

3.11 dump.dump Class Reference

Public Member Functions

- `def __init__`
- `def read`
- `def setVarIDs`
- `def readHeader`
- `def readHeaderBinary`
- `def readHeaderAscii`
- `def readBinaryVar`
- `def readAsciiVar`
- `def printHeader`
- `def printVar`
- `def printDump`

Public Attributes

- `fileName`
- `f`
- `type`
- `varIDs`
- `varNames`
- `numDims`

- **version**
- **time**
- **timeStepIndex**
- **delta_t_nm1half**
- **delta_t_np1half**
- **alpha**
- **eosStringLen**
- **eosString**
- **gamma**
- **av**
- **avthreshold**
- **globalDims**
- **boundaryConditions**
- **num1DZones**
- **numGhostCells**
- **numVars**
- **varInfo**
- **varSize**
- **vars**

3.11.1 Constructor & Destructor Documentation

3.11.1.1 `def dump.dump.__init__(self, fileName)`

Initilizes the dump by reading in a binary file.

References `dump.dump.read()`.

3.11.2 Member Function Documentation

3.11.2.1 `def dump.dump.printHeader(self)`

Prints the header of a binary dump file to the standard output.

References `dump.dump.alpha`, `dump.dump.av`, `dump.dump.avthreshold`, `dump.dump.boundaryConditions`, `dump.dump.delta_t_nm1half`, `dump.dump.delta_t_np1half`, `dump.dump.eosString`, `dump.dump.eosStringLen`, `dump.dump.gamma`, `dump.dump.globalDims`, `dump.dump.num1DZones`, `dump.dump.numDims`, `dump.dump.numGhostCells`, `dump.dump.numVars`, `dump.dump.time`, `dump.dump.timeStepIndex`, `dump.dump.type`, `dump.dump.varInfo`, `dump.dump.varSize`, and `dump.dump.version`.

Referenced by `dump.dump.printVar()`.

3.11.2.2 def dump.dump.printVar(self, var)

Prints a variable to the standard output.

References dump.dump.boundaryConditions, dump.dump.num1DZones, dump.dump.numGhostCells, dump.dump.numVars, dump.dump.printHeader(), dump.dump.printVar(), dump.dump.varInfo, dump.dump.vars, and dump.dump.varSize.

Referenced by dump.dump.printVar().

3.11.2.3 def dump.dump.read(self, fileName)

Reads in a binary dump file.

References dump.dump.f, dump.dump.fileName, dump.dump.numVars, dump.dump.readAsciiVar(), dump.dump.readBinaryVar(), dump.dump.readHeader(), dump.dump.setVarIDs(), and dump.dump.type.

Referenced by dump.dump.__init__().

3.11.2.4 def dump.dump.readAsciiVar(self, var)

Read in a variable from an ascii dump file. Must be called with var increasing from 0 to self.numVars.

References dump.dump.boundaryConditions, dump.dump.num1DZones, dump.dump.numGhostCells, dump.dump.varInfo, and dump.dump.varSize.

Referenced by dump.dump.read().

3.11.2.5 def dump.dump.readBinaryVar(self, var)

Read in a variable from a binary dump file. Must be called with var increasing from 0 to self.numVars.

References dump.dump.boundaryConditions, dump.dump.num1DZones, dump.dump.numGhostCells, dump.dump.varInfo, and dump.dump.varSize.

Referenced by dump.dump.read().

3.11.2.6 def dump.dump.readHeader(self)

Reads header information from binary dump file.

References dump.dump.readHeaderAscii(), dump.dump.readHeaderBinary(), and dump.dump.type.

Referenced by dump.dump.read().

3.11.2.7 `def dump.dump.readHeaderAscii(self)`

Reads a header from a ascii file, after the type has been read in.

References `dump.dump.alpha`, `dump.dump.av`, `dump.dump.avthreshold`, `dump.dump.boundaryConditions`, `dump.dump.delta_t_nm1half`, `dump.dump.delta_t_np1half`, `dump.dump.eosString`, `dump.dump.eosStringLen`, `dump.dump.gamma`, `dump.dump.globalDims`, `dump.dump.num1DZones`, `dump.dump.numDims`, `dump.dump.numGhostCells`, `dump.dump.numVars`, `dump.dump.time`, `dump.dump.timeStepIndex`, `dump.dump.varInfo`, `dump.dump.vars`, `dump.dump.varSize`, and `dump.dump.version`.

Referenced by `dump.dump.readHeader()`.

3.11.2.8 `def dump.dump.readHeaderBinary(self)`

Reads a header from a binary file, after the type has been read in.

References `dump.dump.alpha`, `dump.dump.av`, `dump.dump.avthreshold`, `dump.dump.boundaryConditions`, `dump.dump.delta_t_nm1half`, `dump.dump.delta_t_np1half`, `dump.dump.eosString`, `dump.dump.eosStringLen`, `dump.dump.gamma`, `dump.dump.globalDims`, `dump.dump.num1DZones`, `dump.dump.numDims`, `dump.dump.numGhostCells`, `dump.dump.numVars`, `dump.dump.time`, `dump.dump.timeStepIndex`, `dump.dump.varInfo`, `dump.dump.vars`, `dump.dump.varSize`, and `dump.dump.version`.

Referenced by `dump.dump.readHeader()`.

3.11.2.9 `def dump.dump.setVarIDs(self)`

Sets names for the interger values of the grid variables

References `dump.dump.gamma`, `dump.dump.numDims`, `dump.dump.varIDs`, and `dump.dump.varNames`.

Referenced by `dump.dump.read()`.

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

- `dump.py`

3.12 `eos_interp.eosTable` Class Reference

Public Member Functions

- `def load`
- `def write`
- `def plotLogE`
- `def plotLogP`
- `def interpolate`
- `def __init__`

Public Attributes

- **status**
- **X**
- **Z**
- **logT**
- **logD**
- **logP**
- **logE**
- **sFileName**

3.12.1 Detailed Description

Holds equation of state data.

3.12.2 Constructor & Destructor Documentation

3.12.2.1 `def eos_interp.eosTable.__init__(self, sFileName=None)`

Returns a new instance of eosTable.

If sFileName is set it will use that to set the filename to load the data from.

References eos_interp.eosTable.logD, eos_interp.eosTable.logE, eos_interp.eosTable.logP, eos_interp.eosTable.logT, datafile.DataFile.sFileName, eos_interp.eosTable.sFileName, eos_interp.eosTable.status, light_curve.LightCurve.temperature, eos_interp.eosTable.X, and eos_interp.eosTable.Z.

3.12.3 Member Function Documentation

3.12.3.1 `def eos_interp.eosTable.interpolate(self, gridConfig, setExtrapolatedToNan=True)`

Interpolate from self's table to the gridding specified by:

```
logDMin: first (smallest) logD value of grid
logDDel: spacing in logD
numLogD: number of logD grid points
logTMin: first (smallest) logT value of grid
logTDel: spacing in logT
numLogT: number of logT grid points
```

References eos_interp.eosTable.__fillDepNans(), eos_interp.eosTable.logD, eos_interp.eosTable.logE, eos_interp.eosTable.logT, eos_interp.eosTable.X, and eos_interp.eosTable.Z.

3.12.3.2 def eos_interp.eosTable.load(self)

Reads in an OPAL equation of state file.

It puts the resulting file info into:
 self.X: the hydrogen mass fraction
 self.Z: the metal mass fraction
 self.logD: numpy array of log density grid points [g/cm³]
 self.logT: numpy array of log tempeature grid points [K]
 self.logE: numpy array of log energy [ergs/g]
 self.logP: numpy array of log pressure [dynes/cm²]

self.logD, self.logT, self.logE, and self.logP are all the same size numpy arrays, empty emelents have logE and logP as nans.

References eos_interp.eosTable.__fillInDepNans(), eos_interp.eosTable.__gmass(), eos_interp.eosTable.logD, eos_interp.eosTable.logE, eos_interp.eosTable.logP, eos_interp.eosTable.logT, datafile.DataFile.sFileName, eos_interp.eosTable.sFileName, eos_interp.eosTable.status, eos_interp.eosTable.X, and eos_interp.eosTable.Z.

3.12.3.3 def eos_interp.eosTable.plotLogE(self, otherTables = None, logDIndexList = None, wireFrame = True)

Plots LogE

Keywords:

otherTables: a list of other eosTables to include in the plot
 logDIndexList: a list of integers corresponding to which densities to plot the tables at
 wireFrame: if set to true (the default) and logDIndexList is set to None it will plot a 3D wireframe of logE.

References eos_interp.eosTable.logD, eos_interp.eosTable.logE, and eos_interp.eosTable.logT.

3.12.3.4 def eos_interp.eosTable.plotLogP(self, otherTables = None, logDIndexList = None, wireFrame = True)

Plots LogP

Keywords:

otherTables: a list of other eosTables to include in the plot
 logDIndexList: a list of integers corresponding to which densities to plot the tables at
 wireFrame: if set to true (the default) and logDIndexList is set to None it will plot a 3D wireframe of logP.

References eos_interp.eosTable.logD, eos_interp.eosTable.logP, and eos_interp.eosTable.logT.

3.12.3.5 def eos_interp.eosTable.write(self, args)

Generic write function that calls either writeToScreen, or writeToFiel depending on if a file name is specified or not.

References eos_interp.eosTable.__writeToFile(), and eos_interp.eosTable.__writeToScreen().

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

- eos_interp.py

3.13 eos_interp.eosTableManager Class Reference

Public Member Functions

- def [load](#)
- def [interpComp](#)
- def [plotGrid](#)
- def [getTableFromComp](#)
- def [__init__](#)

Public Attributes

- **Z**
- **X**
- **eosFileName**
- **eosTables**

3.13.1 Detailed Description

Manages equation of state files, including how they are interpolated between.

3.13.2 Constructor & Destructor Documentation

3.13.2.1 def eos_interp.eosTableManager.__init__(self, eosFileName = None)

Returns a new instance of eosTableManager.

if eosFileName is set it will call __initFromFile to load settings from a file to initialize the new eosTableManager.

References eos_interp.eosTableManager.__initFromFile(), eos_interp.eosTableManager.__quad(), eos_interp.eosTableManager.__quadInterpInZ(), eos_interp.eosTableManager.eosFileName, eos_interp.eosTableManager.eosTables, eos_interp.eosTable.X, eos_interp.opacityTable.X, eos_interp.opacityTableManager.X, eos_interp.eosTableManager.X, eos_interp.eosTable.Z, eos_interp.opacityTable.Z, eos_interp.opacityTableManager.Z, and eos_interp.eosTableManager.Z.

3.13.3 Member Function Documentation

3.13.3.1 `def eos_interp.eosTableManager.getTableFromComp(self, X, Z)`

Returns a shallow copy of the eos table with matching composition. If none found it returns None.

References `eos_interp.eosTableManager.eosTables`.

3.13.3.2 `def eos_interp.eosTableManager.interpComp(self, X, Z)`

Interpolates a set of eos files and opacities to the desired X and Z, and returns an eosManager with this new set of files which can then be interpolated to the desired rho and T's.

References `eos_interp.opacityTableManager.__cubicSplineInX()`, `eos_interp.eosTableManager.__cubicSplineInX()`, `eos_interp.eosTableManager.__quadInterpInZ()`, `eos_interp.eosTable.X`, `eos_interp.opacityTable.X`, `eos_interp.opacityTableManager.X`, `eos_interp.eosTableManager.X`, `eos_interp.eosTable.Z`, `eos_interp.opacityTable.Z`, `eos_interp.opacityTableManager.Z`, and `eos_interp.eosTableManager.Z`.

3.13.3.3 `def eos_interp.eosTableManager.load(self)`

Loads eos files.

Sets the following:

`self.Z`: a list of Z (metal mass fraction) values of the equation of state files
`self.X`: a list of X (hydrogen mass fraction) values of the equation of state files

References `eos_interp.eosTableManager.eosTables`, `eos_interp.eosTable.X`, `eos_interp.opacityTable.X`, `eos_interp.opacityTableManager.X`, `eos_interp.eosTableManager.X`, `eos_interp.eosTable.Z`, `eos_interp.opacityTable.Z`, `eos_interp.opacityTableManager.Z`, and `eos_interp.eosTableManager.Z`.

3.13.3.4 `def eos_interp.eosTableManager.plotGrid(self, eosIndex)`

Plot rho and T points that form the grid

References `eos_interp.eosTableManager.eosTables`.

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

- `eos_interp.py`

3.14 `plot_2DSlices.File2DSlice` Class Reference

Public Member Functions

- `def load`

Public Attributes

- **fileName**
- **index**
- **planeType**
- **time**
- **eosFile**
- **gamma**
- **coordinateNames**
- **coordinates**
- **dataNames**
- **data**

3.14.1 Member Function Documentation

3.14.1.1 def plot_2DSlices.File2DSlice.load(self, fileName)

sets:
 fileName, file name of the 2D slice
 planeType, type of the 2D slice ("rt","rp", "tp")
 eosFile, file name of the equation of state file, if using a gamma-law gas it is None
 gamma, value of gamma for a gamma-law gas, if using an equation of state table it is None
 coordinateNames, Names of the coordinates
 coordinates, values of the coordinates
 dataNames, names of the data columns
 data, the data columns

References plot_2DSlices.File2DSlice.coordinateNames, plot_2DSlices.File2DSlice.coordinates, plot_2DSlices.File2DSlice.data, make_hdf.hdfFile.data, make_hdf2.fileSet.data, plot_2DSlices.File2DSlice.dataNames, make_hdf2.fileSet.dataNames, plot_2DSlices.File2DSlice.eosFile, light_curve.LightCurve.eosFile, dump.dump.fileName, plot_2DSlices.File2DSlice.fileName, plot_2DSlices.File2DSlice.gamma, dump.dump.gamma, plot_2DSlices.File2DSlice.index, plot_2DSlices.File2DSlice.planeType, plot_2DSlices.File2DSlice.time, light_curve.LightCurve.time, and dump.dump.time.

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

- plot_2DSlices.py

3.15 make_hdf.fileSet Class Reference

Public Member Functions

- def [__init__](#)
- def [makeHDFFiles](#)
- def [convertDumpToHDF](#)

Public Attributes

- **fileRange**
- **timeFile**
- **variables**
- **interpVars**
- **supportedNodeAttributes**

3.15.1 Constructor & Destructor Documentation

3.15.1.1 `def make_hdf.fileSet.__init__(self, element)`

Initialize an fileSet from an xml node

References `make_hdf.fileSet.__checkSuppotedNodeAttributes()`, `make_hdf.fileSet.__setSupportedNodeAttributes()`, `plot_profile.DataSet.baseFileName`, `plot_profile.DataSet.end`, `make_hdf.fileSet.fileRange`, `make_hdf.hdfFile.interpVars`, `make_hdf.fileSet.interpVars`, `plot_profile.DataSet.start`, `make_hdf.fileSet.supportedNodeAttributes`, `make_hdf.fileSet.timeFile`, `make_hdf.hdfFile.variables`, and `make_hdf.fileSet.variables`.

3.15.2 Member Function Documentation

3.15.2.1 `def make_hdf.fileSet.convertDumpToHDF(self, dump)`

Converts a dump ifle to an hdf file formatted in the way sepcified in the xml configuration file

References `make_hdf.hdfFile.interpVars`, `make_hdf.fileSet.interpVars`, `make_hdf.hdfFile.variables`, and `make_hdf.fileSet.variables`.

Referenced by `make_hdf2.fileSet.makeHDFFiles()`, and `make_hdf.fileSet.makeHDFFiles()`.

3.15.2.2 `def make_hdf.fileSet.makeHDFFiles(self, options)`

Makes HDF files specified by settings

References `plot_profile.DataSet.baseFileName`, `make_hdf.fileSet.convertDumpToHDF()`, `plot_profile.DataSet.end`, and `plot_profile.DataSet.start`.

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

- `make_hdf.py`

3.16 make_hdf2.fileSet Class Reference

Public Member Functions

- def [__init__](#)
- def [makeHDFFiles](#)
- def [convertDumpToHDF](#)
- def [setAdditionalVariables](#)
- def [getDataFromDump](#)

Public Attributes

- [fileRange](#)
- [timeFile](#)
- [frequency](#)
- [outputPath](#)
- [radialCutZone](#)
- [includeBoundaries](#)
- [numRInterp](#)
- [supportedNodeAttributes](#)
- [data](#)
- [dataMax](#)
- [dataMin](#)
- [dataShape](#)
- [dataNames](#)
- [dataIDs](#)

3.16.1 Constructor & Destructor Documentation

3.16.1.1 def make_hdf2.fileSet.__init__(self, element)

Initialize an fileSet from an xml node

References [make_hdf2.fileSet.__checkSuppotedNodeAttributes\(\)](#), [make_hdf.fileSet.___checkSuppotedNodeAttributes\(\)](#), [make_hdf2.fileSet.__setSupportedNodeAttributes\(\)](#), [make_hdf.fileSet.__setSupportedNodeAttributes\(\)](#), [plot_profile.DataSet.baseFileName](#), [plot_profile.DataSet.end](#), [make_hdf2.fileSet.fileRange](#), [make_hdf.fileSet.fileRange](#), [make_hdf2.fileSet.frequency](#), [light_curve.LightCurve.frequency](#), [make_hdf2.fileSet.includeBoundaries](#), [make_hdf2.fileSet.numRInterp](#), [make_hdf2.fileSet.outputPath](#), [make_hdf2.fileSet.radialCutZone](#), [plot_profile.DataSet.start](#), [make_hdf2.fileSet.supportedNodeAttributes](#), [make_hdf.fileSet.supportedNodeAttributes](#), [make_hdf2.fileSet.timeFile](#), and [make_hdf.fileSet.timeFile](#).

3.16.2 Member Function Documentation

3.16.2.1 `def make_hdf2.fileSet.convertDumpToHDF(self, dump)`

Converts a dump ifle to an hdf file formatted in the way sepcified in the xml configuration file

References `make_hdf.hdfFile.__interpolateLinearIn1DI()`, `make_hdf2.fileSet.__interpolateLinearIn1DI()`, `make_hdf.hdfFile.data`, `make_hdf2.fileSet.data`, `make_hdf2.fileSet.dataIDs`, `make_hdf.hdfFile.dataMax`, `make_hdf2.fileSet.dataMax`, `make_hdf.hdfFile.dataMin`, `make_hdf2.fileSet.dataMin`, `make_hdf2.fileSet.dataNames`, `make_hdf2.fileSet.dataShape`, `make_hdf2.fileSet.getDataFromDump()`, `make_hdf2.fileSet.includeBoundaries`, `make_hdf2.fileSet.numRInterp`, `make_hdf2.fileSet.outputPath`, `make_hdf2.fileSet.radialCutZone`, and `make_hdf2.fileSet.setAdditionalVariables()`.

Referenced by `make_hdf2.fileSet.makeHDFFiles()`.

3.16.2.2 `def make_hdf2.fileSet.makeHDFFiles(self, options)`

Makes HDF files specified by settings

References `plot_profile.DataSet.baseFileName`, `make_hdf2.fileSet.convertDumpToHDF()`, `make_hdf.fileSet.convertDumpToHDF()`, `plot_profile.DataSet.end`, `make_hdf2.fileSet.frequency`, `light_curve.LightCurve.frequency`, `plot_profile.DataSet.start`, `make_hdf2.fileSet.timeFile`, and `make_hdf.fileSet.timeFile`.

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

- `make_hdf2.py`

3.17 `make_hdf.hdfFile` Class Reference

Public Member Functions

- `def __init__`
- `def printVarToScreen`
- `def write`

Public Attributes

- `variables`
- `interpVars`
- `varNames`
- `varIDs`
- `data`
- `dataMax`
- `dataMin`

3.17.1 Member Function Documentation

3.17.1.1 def make_hdf.hdfFile.write(self)

this function writes the data specified in the configuration file to a new hdf file. It does this by interpolating where nessacary to get data at the right location

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

- make_hdf.py

3.18 eos_interp.interpTable Class Reference

Public Member Functions

- def [interpolate](#)
- def [read](#)
- def [plotLogE](#)
- def [plotLogP](#)
- def [plotLogK](#)
- def [__init__](#)

Public Attributes

- eosAtNewComp
- opacityAtNewComp
- eosTable
- opacityTable
- sFileName
- numLogR
- X
- Z
- gridConfig
- logD
- logT
- logP
- logE
- logK
- outputFile
- plot
- setNans

3.18.1 Detailed Description

This class reads in and holds data for an equations of state and opacities from a file formated in the same was as read to and written by the class defined in eos.h, and implemented in eos.cpp.

3.18.2 Constructor & Destructor Documentation

3.18.2.1 `def eos_interp.interpTable.__init__(self, tableElement = None)`

Reads in an interpolation table info from from the xml element tableElement.

References eos_interp.interpTable.eosTable, eos_interp.interpTable.gridConfig, eos_interp.interpTable.opacityTable, eos_interp.interpTable.outputFile, eos_interp.interpTable.plot, eos_interp.interpTable.setNans, eos_interp.eosTable.X, eos_interp.opacityTable.X, eos_interp.opacityTableManager.X, eos_interp.eosTableManager.X, eos_interp.interpTable.X, eos_interp.eosTable.Z, eos_interp.opacityTable.Z, eos_interp.opacityTableManager.Z, eos_interp.eosTableManager.Z, and eos_interp.interpTable.Z.

3.18.3 Member Function Documentation

3.18.3.1 `def eos_interp.interpTable.interpolate(self, eosSet, opacitySet, withoutNans = None)`

creates the interpolated table and writes it out

References eos_interp.interpTable.__writeCompleteEOS(), eos_interp.interpTable.eosAtNewComp, eos_interp.interpTable.eosTable, eos_interp.interpTable.gridConfig, eos_interp.interpTable.opacityAtNewComp, eos_interp.interpTable.opacityTable, eos_interp.interpTable.outputFile, eos_interp.interpTable.plot, eos_interp.interpTable.setNans, eos_interp.eosTable.X, eos_interp.opacityTable.X, eos_interp.opacityTableManager.X, eos_interp.eosTableManager.X, eos_interp.interpTable.X, eos_interp.eosTable.Z, eos_interp.opacityTable.Z, eos_interp.opacityTableManager.Z, eos_interp.eosTableManager.Z, and eos_interp.interpTable.Z.

3.18.3.2 `def eos_interp.interpTable.plotLogE(self, otherTables = None, logDIndexList = None, logDRangeList = None, wireFrame = True, rstride = 1, cstride = 1, outputfile = None)`

Plots LogE

Keywords:

otherTables: a list of other eosTables to include in the plot

logDIndexList: a list of integers corresponding to which densities to plot the tables at

wireFrame: if set to true (the default) and logDIndexList is set to None it will plot a 3D wireframe of logE.

References eos_interp.eosTable.logD, eos_interp.interpTable.logD, eos_interp.eosTable.logE, eos_interp.interpTable.logE, eos_interp.eosTable.logT, eos_interp.opacityTable.logT, eos_interp.interpTable.logT, datafile.DataFile.sFileName, eos_interp.eosTable.sFileName, eos_interp.opacityTable.sFileName, and eos_interp.interpTable.sFileName.

3.18.3.3 `def eos_interp.interpTable.plotLogK(self, otherTables = None,
logDIndexList = None, logDRangeList = None, wireFrame = True, outfile =
None)`

Plots opacity

Keywords:

otherTables: a list of opacity tables to also be plotted

logDIndex: a list of integers used to indicate a specific logR index to plot 2D line plots at.

References eos_interp.eosTable.logD, eos_interp.interpTable.logD, eos_interp.opacityTable.logK, eos_interp.interpTable.logK, eos_interp.eosTable.logT, eos_interp.opacityTable.logT, eos_interp.interpTable.logT, datafile.DataFile.sFileName, eos_interp.eosTable.sFileName, eos_interp.opacityTable.sFileName, and eos_interp.interpTable.sFileName.

3.18.3.4 `def eos_interp.interpTable.plotLogP(self, otherTables = None,
logDIndexList = None, logDRangeList = None, wireFrame = True, outfile =
None)`

Plots LogP

Keywords:

otherTables: a list of other eosTables to include in the plot

logDIndexList: a list of integers corresponding to which densities to plot the tables at

wireFrame: if set to true (the default) and logDIndexList is set to None it will plot a 3D wireframe of logP.

References eos_interp.eosTable.logD, eos_interp.interpTable.logD, eos_interp.eosTable.logP, eos_interp.interpTable.logP, eos_interp.eosTable.logT, eos_interp.opacityTable.logT, eos_interp.interpTable.logT, datafile.DataFile.sFileName, eos_interp.eosTable.sFileName, eos_interp.opacityTable.sFileName, and eos_interp.interpTable.sFileName.

3.18.3.5 `def eos_interp.interpTable.read(self, sFilename)`

Reads in an interpolated table

References eos_interp.interpTable.gridConfig, eos_interp.eosTable.logD, eos_interp.interpTable.logD, eos_interp.eosTable.logE, eos_interp.interpTable.logE, eos_interp.opacityTable.logK, eos_interp.interpTable.logK, eos_interp.eosTable.logP, eos_interp.interpTable.logP, eos_interp.eosTable.logT, eos_interp.opacityTable.logT, eos_interp.interpTable.logT, eos_interp.interpTable.numLogR, datafile.DataFile.sFileName, eos_interp.eosTable.sFileName, eos_interp.opacityTable.sFileName, eos_interp.interpTable.sFileName, eos_interp.eosTable.X, eos_interp.opacityTable.X, eos_interp.opacityTableManager.X, eos_interp.eosTableManager.X, eos_interp.interpTable.X, eos_interp.eosTable.Z, eos_interp.opacityTable.Z, eos_interp.opacityTableManager.Z, eos_interp.eosTableManager.Z, and eos_interp.interpTable.Z.

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

- eos_interp.py

3.19 eos_interp.interpTableManager Class Reference

Public Member Functions

- def [createTables](#)
- def [__init__](#)

Public Attributes

- **configFile**
- **eosSet**
- **opacitySet**
- **tables**

3.19.1 Constructor & Destructor Documentation

3.19.1.1 `def eos_interp.interpTableManager.__init__(self, configFile=None)`

Initializes interpTableManager from the given configuration file.

References eos_interp.interpTableManager.__readInterpTableConfigs(), eos_interp.interpTableManager.configFile, eos_interp.interpTableManager.eosSet, eos_interp.interpTableManager.opacitySet, and eos_interp.interpTableManager.tables.

3.19.2 Member Function Documentation

3.19.2.1 `def eos_interp.interpTableManager.createTables(self, withoutNans=None)`

Creates interpolated tables and write them out.

References eos_interp.interpTableManager.eosSet, eos_interp.interpTableManager.opacitySet, and eos_interp.interpTableManager.tables.

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

- eos_interp.py

3.20 make_hdf.interpVar Class Reference

Public Member Functions

- def [__init__](#)

Public Attributes

- **numPoints**
- **name**
- **formula**

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

- `make_hdf.py`

3.21 light_curve.LightCurve Class Reference

Public Member Functions

- `def __init__`
- `def create`
- `def readProfiles`
- `def readBoloCorr`
- `def calculateCurve`
- `def write`

Public Attributes

- **boloCorrFile**
- **columnBC**
- **withAcceleration**
- **inputFileRange**
- **frequency**
- **eosFile**
- **zonesFromSurf**
- **outputFile**
- **nNumFiles**
- **luminosity**
- **temperature**
- **interiorMass**
- **time**
- **gridVelocity**
- **radius**
- **TMin**
- **loggMin**
- **TDel**
- **loggDel**
- **numLogg**
- **numT**
- **BC**

3.21.1 Member Function Documentation

3.21.1.1 `def light_curve.LightCurve.calculateCurve(self)`

Creates the light curve by converting luminosity to bolometric magnitude and then applying a bolometric correction and returns a 2D list of times and light curve magnitudes.

References `light_curve.LightCurve.BC`, `light_curve.LightCurve.gridVelocity`, `light_curve.LightCurve.interiorMass`, `light_curve.LightCurve.loggDel`, `light_curve.LightCurve.loggMin`, `light_curve.LightCurve.luminosity`, `light_curve.LightCurve.numLogg`, `light_curve.LightCurve.numT`, `light_curve.LightCurve.radius`, `light_curve.LightCurve.TDel`, `light_curve.LightCurve.temperature`, `light_curve.LightCurve.time`, `dump.dump.time`, `light_curve.LightCurve.TMin`, and `light_curve.LightCurve.withAcceleration`.

3.21.1.2 `def light_curve.LightCurve.readBoloCorr(self)`

Reads in the bolometric correction table

References `light_curve.LightCurve.BC`, `light_curve.LightCurve.boloCorrFile`, `light_curve.LightCurve.columnBC`, `light_curve.LightCurve.loggDel`, `light_curve.LightCurve.loggMin`, `light_curve.LightCurve.numLogg`, `light_curve.LightCurve.numT`, `light_curve.LightCurve.TDel`, and `light_curve.LightCurve.TMin`.

3.21.1.3 `def light_curve.LightCurve.readProfiles(self, options)`

Reads the needed data to create the light curve from the radial profile files

References `plot_profile.DataSet.baseFileName`, `plot_profile.DataSet.end`, `light_curve.LightCurve.eosFile`, `light_curve.LightCurve.frequency`, `light_curve.LightCurve.gridVelocity`, `light_curve.LightCurve.interiorMass`, `light_curve.LightCurve.luminosity`, `light_curve.LightCurve.nNumFiles`, `light_curve.LightCurve.radius`, `plot_profile.DataSet.start`, `light_curve.LightCurve.temperature`, `light_curve.LightCurve.time`, `dump.dump.time`, and `light_curve.LightCurve.zonesFromSurf`.

3.21.1.4 `def light_curve.LightCurve.write(self, curve)`

Writes out the light curve to the specified output file.

References `light_curve.LightCurve.outputFile`, and `eos_interp.interpTable.outputFile`.

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

- `light_curve.py`

3.22 `eos_interp.opacityTable` Class Reference

Public Member Functions

- def [load](#)
- def [plotLogK](#)
- def [interpolate](#)
- def [__init__](#)
- def [fillInDepNans](#)

Public Attributes

- **multitableFile**
- **X**
- **Z**
- **sFileName**
- **logT**
- **logR**
- **logK**

3.22.1 Detailed Description

Holds opacity table data.

Initialize with a composition (X,Z), file name and whether the file name contains multiple.

3.22.2 Constructor & Destructor Documentation

3.22.2.1 `def eos_interp.opacityTable.__init__(self, X=None, Z=None, sFileName=None, multitableFile=None)`

Initializes the opacity object.

```
sets:
self.X: the hydrogen mass fraction
self.Z: the metal mass fraction
self.sFileName: the file name to load the table from
self.multitableFile: whether or not the file has more than one table in it
```

References `eos_interp.opacityTable.logK`, `eos_interp.opacityTable.logR`, `eos_interp.eosTable.logT`, `eos_interp.opacityTable.logT`, `eos_interp.opacityTable.multitableFile`, `datafile.DataFile.sFileName`, `eos_interp.eosTable.sFileName`, `eos_interp.opacityTable.sFileName`, `eos_interp.eosTable.X`, `eos_interp.opacityTable.X`, `eos_interp.eosTable.Z`, and `eos_interp.opacityTable.Z`.

3.22.3 Member Function Documentation

3.22.3.1 `def eos_interp.opacityTable.fillInDepNans(self)`

Fills in logR and logT values to make a rectangular grid

References `eos_interp.opacityTable.logR`, `eos_interp.eosTable.logT`, and `eos_interp.opacityTable.logT`.

3.22.3.2 `def eos_interp.opacityTable.interpolate(self, gridConfig, setExtrapolatedToNan = True)`

Interpolate from self's table to the gridding specified by:

parameters:

`logDMin`: first (smallest) `logD` value of grid

`logDDel`: spacing in `logD`

`numLogD`: number of `logD` grid points

`logTMin`: first (smallest) `logT` value of grid

`logTDel`: spacing in `logT`

`numLogT`: number of `logT` grid points

keyword:

`setExtrapolatedToNan`: controls whether extrapolated points are set to nans (default is True)

returns:

an opacity table interpolated to the specified grid. In addition to the regular members of an opacity table `logD` is also included.

References `eos_interp.eosTable.__fillDepNans()`, `eos_interp.opacityTable.__fillDepNans()`, `eos_interp.opacityTable.logK`, `eos_interp.opacityTable.logR`, `eos_interp.eosTable.logT`, `eos_interp.opacityTable.logT`, `eos_interp.eosTable.X`, `eos_interp.opacityTable.X`, `eos_interp.eosTable.Z`, and `eos_interp.opacityTable.Z`.

3.22.3.3 `def eos_interp.opacityTable.load(self)`

Load from a file an opacity table for composition of the current opacity object. It does this by advancing a file until the composition is matched and then calls `__loadTableFromFile` to load the `logR`, `logT`, and `logK` values.

References `eos_interp.opacityTable.__loadTableFromFile()`, `eos_interp.opacityTable.multitableFile`, `datafile.DataFile.sFileName`, `eos_interp.eosTable.sFileName`, `eos_interp.opacityTable.sFileName`, `eos_interp.eosTable.X`, `eos_interp.opacityTable.X`, `eos_interp.eosTable.Z`, and `eos_interp.opacityTable.Z`.

3.22.3.4 `def eos_interp.opacityTable.plotLogK(self, otherTables = None, logRIndex = None, wireFrame = True)`

Plots opacity

Keywords:

`otherTables`: a list of opacity tables to also be plotted

`logRIndex`: a list of integers used to indicate a specific `logR` index to plot 2D line plots at

References `eos_interp.opacityTable.logK`, `eos_interp.opacityTable.logR`, `eos_interp.eosTable.logT`, and `eos_interp.opacityTable.logT`.

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

- eos_interp.py

3.23 eos_interp.opacityTableManager Class Reference

Public Member Functions

- def [load](#)
- def [interpComp](#)
- def [plotGrids](#)
- def [getTableFromComp](#)
- def [__init__](#)

Public Attributes

- **opacityConfigFileName**
- **opacityFileNames**
- **opacityTables**
- **Z**
- **X**

3.23.1 Detailed Description

Manages opacity files, including how they are interpolated between in composition.

3.23.2 Constructor & Destructor Documentation

3.23.2.1 `def eos_interp.opacityTableManager.__init__(self, opacityConfigFile = None)`

Creates a new instance of opacityTableManager.

If opacityConfigFile is set it will try to parse it for xml settings to get all the file names of the opacity files to include in the opacityTableManager.

References eos_interp.opacityTableManager.__getCompositions(), eos_interp.opacityTableManager.__merge2files(), eos_interp.opacityTableManager.opacityConfigFileName, eos_interp.opacityTableManager.opacityFileNames, eos_interp.opacityTableManager.opacityTables, eos_interp.eosTable.X, eos_interp.opacityTable.X, eos_interp.opacityTableManager.X, eos_interp.eosTable.Z, eos_interp.opacityTable.Z, and eos_interp.opacityTableManager.Z.

3.23.3 Member Function Documentation

3.23.3.1 `def eos_interp.opacityTableManager.getTableFromComp(self, X, Z)`

Returns a shallow copy of the opacity table with matching composition.

References `eos_interp.opacityTableManager.opacityTables`.

3.23.3.2 `def eos_interp.opacityTableManager.interpComp(self, X, Z)`

Interpolates a set of opacity files to the desired X and Z, and returns an the interpolated opacityTable.

Parameters:

X: hydrogen mass fraction

Z: metal mass fraction

References `eos_interp.opacityTableManager.__bicubicSplineInXZ()`, `eos_interp.eosTable.X`, `eos_interp.opacityTable.X`, `eos_interp.opacityTableManager.X`, `eos_interp.eosTable.Z`, `eos_interp.opacityTable.Z`, and `eos_interp.opacityTableManager.Z`.

3.23.3.3 `def eos_interp.opacityTableManager.load(self)`

Loads opacity files and merge files at duplicate compositions (i.e. merges low and high temperature opacity tables).

Sets the following:

`self.X`: list of hydrogen mass fractions covered by opacity tables

`self.Z`: list of metal mass fractions covered by opacity tables

References `eos_interp.opacityTableManager.__merge()`, `eos_interp.opacityTableManager.__setCompLists()`, and `eos_interp.opacityTableManager.opacityTables`.

3.23.3.4 `def eos_interp.opacityTableManager.plotGrids(self, opacityIndex)`

Plot LogR and LogT points that form the opacity grid.

Parameters:

`opacityIndex`: a list of integers used to select which opacity tables will be plotted

References `eos_interp.opacityTableManager.opacityTables`.

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

- `eos_interp.py`

3.24 `work_plot.PdVPlotSettings` Class Reference

Public Member Functions

- def `__init__`
- def `parseXML`

Public Attributes

- `startZone`
- `points`
- `grid`
- `format`
- `outputFile`
- `lines`
- `show`

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

- `work_plot.py`

3.25 plot_file.Plot Class Reference

Public Member Functions

- def `__init__`
- def `load`

Public Attributes

- `ylabel`
- `curves`
- `texts`
- `limits`
- `grid`
- `bMinorTicks`
- `legendloc`
- `numpoints`
- `weightHeight`
- `ticks`

3.25.1 Detailed Description

This class holds all the information for a single plot, namely the list of curves for that plot.

3.25.2 Constructor & Destructor Documentation

3.25.2.1 `def plot_file.Plot.__init__(self, element)`

This method initlizes the plot object

References `plot_file.Plot.bMinorTics`, `plot_file.Plot.curves`, `plot_file.Plot.grid`, `plot_file.Plot.legendloc`, `plot_file.Plot.limits`, `plot_file.Plot.numpoints`, `plot_file.Plot.texts`, `plot_file.Plot.ticks`, `plot_file.Plot.weightHeight`, and `plot_file.Plot.ylabel`.

3.25.3 Member Function Documentation

3.25.3.1 `def plot_file.Plot.load(self, files, options)`

loads the data for a plot, y-data is stored in the curves, and sets the ylabel from the first file read in

References `plot_file.Plot.curves`.

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

- `plot_file.py`

3.26 `plot_profile.Plot` Class Reference

Public Member Functions

- `def __init__`
- `def load`
- `def setLimits`

Public Attributes

- `ylabel`
- `curves`
- `limits`
- `grid`
- `bMinorTics`
- `legendloc`

3.26.1 Detailed Description

This class holds all the information for a single plot, namely the list of curves for that pl

3.26.2 Constructor & Destructor Documentation

3.26.2.1 `def plot_profile.Plot.__init__(self, element, type)`

This method initlizes the plot object

References `plot_file.Plot.bMinorTics`, `plot_profile.Plot.bMinorTics`, `plot_file.Axis.bMinorTics`, `plot_file.Plot.curves`, `plot_profile.Plot.curves`, `plot_file.Plot.grid`, `plot_profile.Plot.grid`, `plot_file.Axis.grid`, `plot_file.Plot.legendloc`, `plot_profile.Plot.legendloc`, `plot_file.Plot.limits`, `plot_profile.Plot.limits`, `plot_file.Axis.limits`, `plot_file.Plot.ylabel`, and `plot_profile.Plot.ylabel`.

3.26.3 Member Function Documentation

3.26.3.1 `def plot_profile.Plot.load(self, fileData, options, dataSet, nFileCount)`

loads the data for a plot, y-data is stored in the curves, and sets the ylabel from the first file read in

References `plot_file.Plot.curves`, `plot_profile.Plot.curves`, `plot_file.Plot.ylabel`, and `plot_profile.Plot.ylabel`.

3.26.3.2 `def plot_profile.Plot.setLimits(self, x, xlimits, bTimeAxis)`

Sets the y limits from the maximum, and minimum y values in the curves on the plot in the xrange specified, over all files.

References `plot_file.Plot.curves`, `plot_profile.Plot.curves`, `plot_file.Plot.limits`, `plot_profile.Plot.limits`, and `plot_file.Axis.limits`.

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

- `plot_profile.py`

3.27 work_plot.Settings Class Reference

Public Member Functions

- `def __init__`
- `def parseXML`

Public Attributes

- `pColumn`
- `pColumnHeader`
- `tColumn`

- `tColumnHeader`
- `rhoColumn`
- `rhoColumnHeader`
- `QColumn`
- `QColumnHeader`
- `deltaMColumn`
- `deltaMColumnHeader`
- `AV`
- `outputFile`
- `plotPdVCurves`
- `files`
- `workPlotSettings`
- `PdVPlotSettings`

3.27.1 Constructor & Destructor Documentation

3.27.1.1 `def work_plot.Settings.__init__(self, oldColumns = False)`

Initialize settings

References `work_plot.Settings.AV`, `work_plot.Settings.deltaMColumn`, `work_plot.Settings.deltaMColumnHeader`, `work_plot.WorkPlotSettings.outputFile`, `light_curve.LightCurve.outputFile`, `work_plot.PdVPlotSettings.outputFile`, `work_plot.Settings.outputFile`, `eos_interp.interpTable.outputFile`, `work_plot.Settings.pColumn`, `work_plot.Settings.pColumnHeader`, `work_plot.WorkPlotSettings.plotPdVCurves`, `work_plot.Settings.plotPdVCurves`, `work_plot.Settings.QColumn`, `work_plot.Settings.QColumnHeader`, `work_plot.Settings.rhoColumn`, `work_plot.Settings.rhoColumnHeader`, `work_plot.Settings.tColumn`, and `work_plot.Settings.tColumnHeader`.

3.27.2 Member Function Documentation

3.27.2.1 `def work_plot.Settings.parseXML(self, fileName)`

Get user settings from XML file

References `work_plot.Settings.AV`, `work_plot.Settings.files`, `plot_file.DataSet.files`, `work_plot.Settings.PdVPlotSettings`, `work_plot.WorkPlotSettings.plotPdVCurves`, `work_plot.Settings.plotPdVCurves`, and `work_plot.Settings.workPlotSettings`.

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

- `work_plot.py`

3.28 `plot_file.Text` Class Reference

Public Member Functions

- `def __init__`

Public Attributes

- `x`
- `y`
- `text`

3.28.1 Detailed Description

This class holds informatin for a text object on a plot.

3.28.2 Constructor & Destructor Documentation

3.28.2.1 `def plot_file.Text.__init__(self, element)`

This method initializest a text object from an xml element

References `plot_file.Text.text`, `plot_file.Text.x`, `calculate_residuals_of_light_curve_fit.DataFunction.x`, `plot_file.Text.y`, and `calculate_residuals_of_light_curve_fit.DataFunction.y`.

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

- `plot_file.py`

3.29 `make_hdf.variable` Class Reference

Public Member Functions

- `def __init__`

Public Attributes

- `indep`
- `fillValue`
- `formula`

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

- `make_hdf.py`

3.30 work_plot.WorkPlotSettings Class Reference

Public Member Functions

- def `__init__`
- def `parseXML`

Public Attributes

- `minTemp`
- `ylim`
- `grid`
- `points`
- `lines`
- `plotPdVCurves`
- `startZone`
- `temperatureProfileFile`
- `format`
- `outputFile`

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

- `work_plot.py`

Chapter 4

File Documentation

4.1 average_PKE.py File Reference

Functions

- `def average_PKE.parseOptions`
- `def average_PKE.main`
- `def average_PKE.averagePKE`

4.1.1 Detailed Description

4.2 combine_bins.py File Reference

Functions

- `def combine_bins.main`
- `def combine_bins.combine_bin_files`

4.2.1 Detailed Description

4.3 combine_bins_persistent.py File Reference

Functions

- `def combine_bins_persistent.main`

4.3.1 Detailed Description

4.4 `compare_sedov_blasts.py` File Reference

Functions

- `def compare_sedov_blasts.main`

4.4.1 Detailed Description

4.5 `cp_files.py` File Reference

Functions

- `def cp_files.main`
- `def cp_files.cp_files`

4.5.1 Detailed Description

Index

`__init__`
 `calculate_residuals_of_light_curve_fit::Bin`, 8
 `dump::dump`, 18
 `eos_interp::eosTable`, 21
 `eos_interp::eosTableManager`, 23
 `eos_interp::interpTable`, 30
 `eos_interp::interpTableManager`, 32
 `eos_interp::opacityTable`, 35
 `eos_interp::opacityTableManager`, 37
 `make_hdf2::fileSet`, 27
 `make_hdf::fileSet`, 26
 `plot_file::Axis`, 5
 `plot_file::Curve`, 11
 `plot_file::DataSet`, 15
 `plot_file::Plot`, 40
 `plot_file::Text`, 43
 `plot_profile::Axis`, 7
 `plot_profile::Curve`, 12
 `plot_profile::DataSet`, 16
 `plot_profile::Plot`, 41
 `work_plot::Settings`, 42

`addEvenBins`
 `calculate_residuals_of_light_curve_fit::BinnedData`, 9

`addPoint`
 `calculate_residuals_of_light_curve_fit::Bin`, 8

`average_PKE.py`, 45

`binData`
 `calculate_residuals_of_light_curve_fit::BinnedData`, 9

`calculate_residuals_of_light_curve_fit::Bin`, 7

`calculate_residuals_of_light_curve_fit::BinnedData`, 9

`calculate_residuals_of_light_curve_fit::DataFunction`, 14

`calculate_residuals_of_light_curve_fit::Bin`
 `__init__`, 8
 `addPoint`, 8
 `getMean`, 8
 `getSTDD`, 8

`calculate_residuals_of_light_curve_fit::BinnedData`
 `addEvenBins`, 9
 `binData`, 9
 `getBinCenters`, 9
 `getMean`, 9
 `getSTDD`, 10

`calculateCurve`
 `light_curve::LightCurve`, 34

`combine_bins.py`, 45

`combine_bins_persistent.py`, 45

`compare_sedov_blasts.py`, 46

`convertDumpToHDF`
 `make_hdf2::fileSet`, 28
 `make_hdf::fileSet`, 26

`cp_files.py`, 46

`createTables`
 `eos_interp::interpTableManager`, 32

`datafile.DataFile`, 13

`datafile::DataFile`
 `readFile`, 14
 `readFileFixed`, 14
 `readFileUnFixed`, 14

`dump.dump`, 17

`dump::dump`
 `__init__`, 18
 `printHeader`, 18
 `printVar`, 18
 `read`, 19
 `readAsciiVar`, 19
 `readBinaryVar`, 19
 `readHeader`, 19
 `readHeaderAscii`, 19
 `readHeaderBinary`, 20

- setVarIDs, 20
- eos_interp.eosTable, 20
- eos_interp.eosTableManager, 23
- eos_interp.interpTable, 29
- eos_interp.interpTableManager, 32
- eos_interp.opacityTable, 34
- eos_interp.opacityTableManager, 37
- eos_interp::eosTable
 - __init__, 21
 - interpolate, 21
 - load, 21
 - plotLogE, 22
 - plotLogP, 22
 - write, 22
- eos_interp::eosTableManager
 - __init__, 23
 - getTableFromComp, 24
 - interpComp, 24
 - load, 24
 - plotGrid, 24
- eos_interp::interpTable
 - __init__, 30
 - interpolate, 30
 - plotLogE, 30
 - plotLogK, 30
 - plotLogP, 31
 - read, 31
- eos_interp::interpTableManager
 - __init__, 32
 - createTables, 32
- eos_interp::opacityTable
 - __init__, 35
 - fillInDepNans, 35
 - interpolate, 36
 - load, 36
 - plotLogK, 36
- eos_interp::opacityTableManager
 - __init__, 37
 - getTableFromComp, 38
 - interpComp, 38
 - load, 38
 - plotGrids, 38
- fillInDepNans
 - eos_interp::opacityTable, 35
- getBinCenters
 - calculate_residuals_of_light_curve_ - fit::BinnedData, 9
- getCurve
 - plot_file::DataSet, 15
 - plot_profile::DataSet, 17
- getMean
 - calculate_residuals_of_light_curve_ - fit::Bin, 8
 - calculate_residuals_of_light_curve_ - fit::BinnedData, 9
- getSTDD
 - calculate_residuals_of_light_curve_ - fit::Bin, 8
 - calculate_residuals_of_light_curve_ - fit::BinnedData, 10
- getTableFromComp
 - eos_interp::eosTableManager, 24
 - eos_interp::opacityTableManager, 38
- interpComp
 - eos_interp::eosTableManager, 24
 - eos_interp::opacityTableManager, 38
- interpolate
 - eos_interp::eosTable, 21
 - eos_interp::interpTable, 30
 - eos_interp::opacityTable, 36
- light_curve.LightCurve, 33
- light_curve::LightCurve
 - calculateCurve, 34
 - readBoloCorr, 34
 - readProfiles, 34
 - write, 34
- load
 - eos_interp::eosTable, 21
 - eos_interp::eosTableManager, 24
 - eos_interp::opacityTable, 36
 - eos_interp::opacityTableManager, 38
 - plot_2DSlices::File2DSlice, 25
 - plot_file::Axis, 6
 - plot_file::Curve, 11
 - plot_file::DataSet, 15
 - plot_file::Plot, 40
 - plot_profile::Axis, 7
 - plot_profile::Curve, 13
 - plot_profile::DataSet, 17
 - plot_profile::Plot, 41
- make_hdf.fileSet, 25
- make_hdf.hdfFile, 28
- make_hdf.interpVar, 32
- make_hdf.variable, 43

- make_hdf2.fileSet, [27](#)
- make_hdf2::fileSet
 - __init__, [27](#)
 - convertDumpToHDF, [28](#)
 - makeHDFFiles, [28](#)
- make_hdf::fileSet
 - __init__, [26](#)
 - convertDumpToHDF, [26](#)
 - makeHDFFiles, [26](#)
- make_hdf::hdfFile
 - write, [29](#)
- makeHDFFiles
 - make_hdf2::fileSet, [28](#)
 - make_hdf::fileSet, [26](#)
- parseXML
 - work_plot::Settings, [42](#)
- plot_2DSlices.File2DSlice, [24](#)
- plot_2DSlices::File2DSlice
 - load, [25](#)
- plot_file.Axis, [5](#)
- plot_file.Curve, [10](#)
- plot_file.DataSet, [15](#)
- plot_file.Plot, [39](#)
- plot_file.Text, [43](#)
- plot_file::Axis
 - __init__, [5](#)
 - load, [6](#)
- plot_file::Curve
 - __init__, [11](#)
 - load, [11](#)
- plot_file::DataSet
 - __init__, [15](#)
 - getCurve, [15](#)
 - load, [15](#)
- plot_file::Plot
 - __init__, [40](#)
 - load, [40](#)
- plot_file::Text
 - __init__, [43](#)
- plot_profile.Axis, [6](#)
- plot_profile.Curve, [12](#)
- plot_profile.DataSet, [16](#)
- plot_profile.Plot, [40](#)
- plot_profile::Axis
 - __init__, [7](#)
 - load, [7](#)
- plot_profile::Curve
 - __init__, [12](#)
 - load, [13](#)
- plot_profile::DataSet
 - __init__, [16](#)
 - getCurve, [17](#)
 - load, [17](#)
- plot_profile::Plot
 - __init__, [41](#)
 - load, [41](#)
 - setLimits, [41](#)
- plotGrid
 - eos_interp::eosTableManager, [24](#)
- plotGrids
 - eos_interp::opacityTableManager, [38](#)
- plotLogE
 - eos_interp::eosTable, [22](#)
 - eos_interp::interpTable, [30](#)
- plotLogK
 - eos_interp::interpTable, [30](#)
 - eos_interp::opacityTable, [36](#)
- plotLogP
 - eos_interp::eosTable, [22](#)
 - eos_interp::interpTable, [31](#)
- printHeader
 - dump::dump, [18](#)
- printVar
 - dump::dump, [18](#)
- read
 - dump::dump, [19](#)
 - eos_interp::interpTable, [31](#)
- readAsciiVar
 - dump::dump, [19](#)
- readBinaryVar
 - dump::dump, [19](#)
- readBoloCorr
 - light_curve::LightCurve, [34](#)
- readFile
 - datafile::DataFile, [14](#)
- readFileFixed
 - datafile::DataFile, [14](#)
- readFileUnFixed
 - datafile::DataFile, [14](#)
- readHeader
 - dump::dump, [19](#)
- readHeaderAscii
 - dump::dump, [19](#)
- readHeaderBinary
 - dump::dump, [20](#)
- readProfiles
 - light_curve::LightCurve, [34](#)

setLimits
 plot_profile::Plot, [41](#)
setVarIDs
 dump::dump, [20](#)

work_plot.PdVPlotSettings, [38](#)
work_plot.Settings, [41](#)
work_plot.WorkPlotSettings, [44](#)
work_plot::Settings
 __init__, [42](#)
 parseXML, [42](#)
write
 eos_interp::eosTable, [22](#)
 light_curve::LightCurve, [34](#)
 make_hdf::hdfFile, [29](#)