data_analysis.**aspnes_equation**(*Energy*, *amplitude*, *phase*, *Eg*, *gamma*)
$$\Re\left[Ae^{i\gamma}(E - E_g + i\Gamma)^{-3}\right]$$

data_analysis.**aspnes_guess**(*x_data*, *y_data*, *data_range*)
Calculate a guess for starting parameters for the fit

> **Parameters**
>
> - **x_data** (`list of list of float`) – list of x data
>
> - **y_data** (`list of list of float`) – list of y data
>
> - **data_range** (`list of float`) – x range over which the data is fit, [min, max]
>
> **Returns** **guess_array** – array of guess arrays, each row in array is a list of values for the fit function variables
>
> **Return type** list of list of float

data_analysis.**broadening_fit**(*T*, *g0*, *g_la*, *g_lo*, *o_lo*, *g_imp*)
$$\Gamma_0 + \Gamma_{LA}T + \frac{\Gamma_{LO}}{\left[\exp(\frac{\hbar\omega_{LO}}{k_B T})\right]} + \Gamma_{Imp}\exp\left(-\frac{E_B}{k_B T_H}\right)$$

data_analysis.**fit_all**(*x_data*, *y_data*, *guess_data*, *fit_func*)
Takes a list of x values and y values and fits them all to a function

> **Parameters**
>
> - **x_data** (`list of list of float`) – set of x data where rows of array correspond to sets of x data
>
> - **y_data** (`list of list of float`) – set of y data where rows of array correspond to sets of y data
>
> - **guess_data** (`list of list of float`) – set of guess values that has the same number of rows as the x and y data and the same number of columns as the number of fitting coefficients as the function
>
> - **fit_func** (`func`) – function to which the data is fit, must be in the form that curve_fit accepts
>
> **Returns**
>
> - **fit_array** (*list of float*) – array of fit values which has the same shape as guess_data
>
> - **rsq_array** (*list of float*) – array of r_squared values

data_analysis.**fit_plot**(*x_data*, *y_data*, *fit_func*, *fit_values*, *fit_variables*, *x_label='X Axis'*, *y_label='Y Axis'*, *title='Title'*)
Plot a given set of variables and the fit

> **Parameters**
>
> - **x_data** (`list of float`) – set of x_data where rows of array correspond to sets of x_data
>
> - **y_data** (`list of float`) – set of y_data where rows of array correspond to sets of y_data
>
> - **fit_func** (`func`) – function to which the data is fit, must be in the form that curve_fit accepts
>
> - **fit_values** (`list of float`) – values to be passed to the fit_func
>
> - **fit_variables** (`list of str`) – string array which is used for the legend of the plot
>
> - **x_label** (`str`) –

- **y_label** (*str*) –

- **title** (*str*) –

`data_analysis.`**`fit_plot_all`** (*x_data*, *y_data*, *fit_func*, *fit_values*, *fit_variables*, *slider_range*)

Create a slider to check fits of files in a directory

Parameters

- **x_data** (*list of list of float*) – set of x_data where rows of array correspond to sets of x_data

- **y_data** (*list of list of float*) – set of y_data where rows of array correspond to sets of y_data

- **fit_func** (*func*) – function to which the data is fit, must be in the form that curve_fit accepts

- **fit_values** (*list of list of float*) – values to be passed to the fit_func

- **fit_variables** (*list of str*) – string array which is used for the legend of the plot

- **slider_range** (*list of float*) – range over which the slider should go, should be an array of the same length as x and y data

`data_analysis.`**`fwhm_all`** (*x_data*, *y_data*, *data_range*)

Takes a list of x values and y values and returns the fwhm of data. Assumes there is a gaussian peak in the range

Parameters

- **x_data** (*list of list of float*) – set of x data where rows of array correspond to sets of x data

- **y_data** (*list of list of float*) – set of y data where rows of array correspond to sets of y data

- **data_range** (*list of float*) – range in which the data is found, should be a array of two values where the first value corresponds to the minimum value in x and the second value is the largest value in x

Returns fwhm_array – an array where each row contains the fwhm of the given x and y data

Return type list of float

`data_analysis.`**`gaussian_guess`** (*x_data*, *y_data*, *data_range*)

Calculate a guess for starting parameters for the fit

Parameters

- **x_data** (*list of list of float*) – list of x data

- **y_data** (*list of list of float*) – list of y data

- **data_range** (*list of float*) – x range of data, [min, max]

Returns guess_array – array of guess arrays, each row in array is a list of values for the fit function variables

Return type list of list of float

`data_analysis.`**`gaussian_guess_with_d`** (*x_data*, *y_data*, *data_range*)

Calculate a guess for starting parameters for the fit

Parameters

- **x_data** (*list of list of float*) – list of x data

- **y_data** (`list of list of float`) – list of y data

- **data_range** (`list of float`) – x range of data, [min, max]

Returns guess_array – array of guess arrays, each row in array is a list of values for the fit function variables

Return type list of list of float

data_analysis.**import_file**(*directory*, *pattern*, *identifier=''*, *return_file=False*)
Takes directory and file identifier and returns the numeric values in the file either as arrays or a single values

Parameters

- **directory** (`str`) – directory in which data files are found or file path

- **pattern** (`regex`) – regex expression which signifies what part of the filename should be saved

- **identifier** (`str`) – the starting variables of the filenames, if none is given all .txt files are opened

- **return_file** (`bool`) – if 'True' pattern is ignored and filename is returned as is

Returns

- **filename_array** (*list of list of str*) – array of either filenames or values extracted from filename, if only file is passed as an argument only a str is returned

- **data_array** (*list of list of list of float*) – multidimensional array of values extracted from file. [file][line in file][value in row], if only file is passed as an argument list of list of str is returned

data_analysis.**integrated_i_plot**(*start_directory*, *wavelength_range*, *jv_identifier='NaN'*, *pl_identifier='.txt'*, *td=True*, *plot_type='sub'*, *time=False*, *temps='all'*)
Create plots of integrated intensity and peak intensity vs temp or bias for files in a directory

Parameters

- **start_directory** (`str`) – directory in which files are found

- **wavelength_range** (`list of float`) – wavelength range over which should be integrated and peak should be found, [min, max]

- **jv_identifier** (`str`) – starting string of jv files

- **pl_identifier** (`str`) – starting string of pl files

- **td** (`bool`) – whether or not the subplots should be organized by temperature or bias

- **plot_type** (`str`) – 'sub' or 'single', determines how data should be plotted

- **time** (`bool`) – if true data is plotted against file creation time

- **temps** (`list of float`) – temperatures to be included in the plot

data_analysis.**one_gaussian**(*x*, *a*, *b*, *c*)
$\frac{A}{c\sqrt{\frac{\pi}{4\ln(2)}}} \exp\left(\frac{-4\ln(2)(x-b)^2}{c^2}\right)$

data_analysis.**one_gaussian_with_d**(*x*, *a*, *b*, *c*, *d*)
$d + \frac{A}{c\sqrt{\frac{\pi}{4\ln(2)}}} \exp\left(\frac{-4\ln(2)(x-b)^2}{c^2}\right)$

data_analysis.**plot_all**(*x_data*, *y_data*, *slider_range*, *x_label='X Axis'*, *y_label='Y Axis'*, *title='Title'*)
Create a slider to check fits of files in a directory

Parameters

- **x_data** (`list of list of float`) – set of x_data where rows of array correspond to sets of x_data

- **y_data** (`list of list of float`) – set of y_data where rows of array correspond to sets of y_data

- **slider_range** (`list of float`) – range over which the slider should go, should be an array of the same length as x and y data

- **x_label** (`str`) –

- **y_label** (`str`) –

- **title** (`str`) –

data_analysis.**plot_directory**(*directory*, *pattern*, *x_range*, *x_col=0*, *y_col=1*, *fit_func='none'*, *guess_func='none'*, *plot_type='waterfall'*, *scatter_size=1*, *linewidth=1*, *color='range'*, *offset='auto'*, *rmv_baseline=False*, *subplot_values='all'*, *return_plot=False*, *fit_variable='none'*, *identifier=''*)

Plot and/or fit data in files in a given directory

Parameters

- **directory** (`str`) – path of directory containing files to be plotted

- **pattern** (`regex`) – pattern to be extracted from the filename

- **x_range** (`list of float`) – x range over which the data is fit, [min, max]

- **x_col** (`int`) – column in txt file which contains x data

- **y_col** (`int`) – column in txt file which contains y data

- **fit_func** (`func`) – function to which the data is fit, must be in the form that curve_fit accepts

- **guess_func** (`func`) – function which generates the guess data must have form f(x_data, y_data, range)

- **plot_type** (`str`) – type of plot: waterfall, subplot, slider

- **scatter_size** (`float`) – size of points on plot

- **linewidth** (`float`) – linewidth of plot

- **color** (`str`) – color of plot

- **offset** (`float`) – offset between plot for waterfall plot

- **rmv_baseline** (`bool`) – whether or not to remove baseline (usually necessary for photoreflectance)

- **subplot_values** (`list of float`) – what values (from filename) should be included in the subplot

- **return_plot** (`bool`) – whether or not to return plot. It is necessary to return the plot in order to edit the plot such as titles or labels

- **fit_variable** (`int`) – which fit variable should be plotted. If this argument is passed only a plot of the fit variable will be returned

- **identifier** (`str`) – starting string of filename, used to filter out files in directory

Returns

- **x_data** (*list of list of float*) – x data extracted from files

- **y_data** (*list of list of float*) – y data extracted from files

- **fit** (*list of list of float*) – fit values (this is only returned if a fit function is passed as an argument)

data_analysis.**plot_scale**(*array*, *percent*)
    Create ranges of min and max for a given array and percentile

        Parameters

- **array** (`list`) – array to be plotted

- **percent** (`float`) – percent of array to be included as padding on either side of the array

        Returns **new_range** – new min and max scale values

        Return type list

data_analysis.**waterfall**(*x_data*, *y_data*, *s=1e-05*, *size=0.25*, *log=True*, *offset=1000*, *title='Waterfall Plot of Data'*, *xlabel='Wavelength (nm)'*, *ylabel='Signal (arb. units)'*, *fit_func=0*, *guess_data=0*, *scatter=True*)
    Creates waterfall plot for a given set of x and y data and optionally includes the fit

        Parameters

- **x_data** (`list of list of float`) – set of x data where rows of array correspond to sets of x data

- **y_data** (`list of list of float`) – set of y data where rows of array correspond to sets of y data

- **s** (`float`) – percent increase between each plot

- **size** (`float`) – size of points or line on graph

- **log** (`bool`) – variable determining whether graph should be scaled logarithmically

- **offset** (`float`) – amount each plot should be offset from the x axis

- **title** (`str`) –

- **xlabel** (`str`) –

- **ylabel** (`str`) –

- **fit_func** (`func`) – function to which the data should be fit, must be in format that curve_fit accepts

- **guess_data** (`list of list of float`) – set of guess values that has the same number of rows as the x and y data and the same number of columns as the number of fitting coefficients as the function

- **scatter** (`bool`) – variable determining whether data should be plotted as a scatter plot or line plot