README

`CalorimeterShower(unsigned int in_enlargement, unsigned int in_n_modules_x, unsigned int in_n_modules_y, unsigned int in_dimensions, DetectorType in_det_type)` : initiallizes the class CalorimeterShower using a magnification of `in_enlargement`, length of the high resolution image in x direction `in_n_modules_x` and in y direction `in_n_modules_y`, the dimension of the image `in_dimension` and the detector type. The detector type can be one of the following:

- `OLGA`
- `MAINZ`
- `GAMS_ECAL1`
- `SHASHLIK`
- `GAMSRH`
- `GAMS_ECAL2`

`set_x_range(double,double)` : sets the range in x direction in mm

`set_y_range(double,double)` : sets the range in y direction in mm

`set_n_modules_x(unsigned int)` : defines the calorimeter cells in x direction that are used with the shower center in the middle

`set_n_modules_y(unsigned int)` :defines the calorimeter cells in y direction that are used with the shower center in the middle

`set_n_events(unsigned int)` : sets the number of events for a toy MC generation

`generate_shower(unsigned int n_events, unsigned int dim)` :generates a toy MC shower with a set number of events `n_events` and a dimension `dim`, which can be 1 or 2 (in theory any natural number)

`generate_shower(unsigned int)` : generates a 1D toy MC shower with a given number of events

`generate_2D_shower(unsigned int)` : generates a 2D toy MC shower with a given number of events

`write_file()` : writes all plots

`histo_to_txt(TH2D*,string)` : writes the histogram as a txt file for Manim

`set_fit_attempts(unsigned int)` : sets the fit attempts for the debiasing the Lasso solution

`lednev_fit(string plot_name=‘‘best fit;1’’)` : performs the phenomenological fit on the debiased AMP solution named `plot_name` in the file created by `multiresolution(...)`

`read_plot(string toolboxFileName, string toolboxDirectory, string toolboxHisto)` : first string defines Toolbox' output file (has to be a .root file) used here for the high resolution approximation named `toolboxHisto` in the root directory `toolboxDirectory`. In case no directory is present, you can use `read_plot(string toolboxFileName, string toolboxHisto)` instead

`multiresolution(uint n_cells_x, uint n_cells_y)` : performs the AMP algorithm on a cut out, which is defined from `-n_cells_x` to `+n_cells_x` in x-direction and from `-n_cells_y` to `+n_cells_y` in y-direction with (0,0) being the maximum, of the high resolution approximation
`multiresolution(uint)` : same as other method, but just in x-dimension
`calibrate_penalty()` : performs a minimization for λ, one part of the penalty factor
`set_direct_fit()` : the Lednev fit will now not fit to the high resolution image, but wil be transformed and fitted to the high resolution approximation
`reduce_fit_range_x(uint n)` : removes `n` cells in the high resolution image or high resolution approximation from both ends of the x axis for the Lednev fit
`reduce_fit_range_y(uint n)` :removes `n` cells in the high resolution image or high resolution approximation from both ends of the y axis for the Lednev fit
`shift_approximation_in_x(uint n)` : by default the cut out is centered around the maximum shower value, but in case this maximum is not the geometrical center, you can move center by `n` cells in x direction
`shift_approximation_in_y(uint n)` : by default the cut out is centered around the maximum shower value, but in case this maximum is not the geometrical center, you can move center by `n` cells in y direction
`set_number_of_parameter_pairs(uint)` : defines the number of parameter pairs (a_i, b_i) for the Lednev fit
`add_initial_limits_for_a(double, double)` : sets the range to draw the initial a from. First call of this method defines a_1, second a_2 and so on
`add_initial_limits_for_b(double, double)` : sets the range to draw the initial b from. First call of this method defines b_1, second b_2 and so on
`write_stats_into_file(string out_file_name)` : prints informations (enlargement, iterations, lowest χ^2, MSE between showers and NCDFs, duration) about the AMP into a txt file
`set_min_lednev_fits(uint)` : defines a minimum number of successful fits that must be reached before termination
`filter_indices_radially(uint in=3)` : includes only cells within a Chebyshev distance of `in`, then excludes the cells in the next larger Chebyshev distance `in+1`, includes the next cells with `in+2`, excludes `in+3` and `in+4`, includes `in+5` and so on
`set_amp_file(string in)` : sets file name of the root file containing the AMP results, read-only
`set_lednev_file(string in)` : sets output file with file name `in` for the phenomenological fit
`set_debias_tolerance(double)` : defines the tolerance for the least squares fit debiasing the AMP result
`set_debias_tolerance_increase(double)` : if the debiasing fails often enough with the given tolerance, the latter will be increased by a factor defined here
`set_lednev_tolerance(double)` : defines the tolerance used for the phenomenological fit
`set_lednev_tolerance_increase(double)` : if the phenomenological fit fails often enough with the given tolerance, the latter will be increased by a factor defined here
`set_max_ncdf_deviation(double)` : defines the maximum absolute difference

between maxima or minima respectively of the high resolution data's NCDF and the Lednev fit result

Minimal working example:

```
unsigned int mag=5;
unsigned int n_modules=5;
unsigned int dim=2;
CalorimeterShower shower(mag, n_modules, n_modules, dim, SHASHLIK);
shower.set_number_of_parameter_pairs(3);
shower.add_initial_limits_for_a(0,0.2);
shower.add_initial_limits_for_a(0.7,0.9);
shower.add_initial_limits_for_b(4,6);
shower.add_initial_limits_for_b(2,2.4);
shower.add_initial_limits_for_b(10,200);
shower.shift_approximation_in_x(0);
shower.shift_approximation_in_y(1);
shower.read_plot("my_shower_cell2368_40GeV_x5.root", "highResApprox");
shower.multiresolution(2,2);
shower.set_lednev_file("lednev_fit_x5_3par_25x25.root");
shower.lednev_fit("best fit;1");
```