SMEFT19 Documentation

Release 3.0

Jorge Alda

Dec 15, 2021

CONTENTS:

This is the documentation for the code that I have written to calculate the results in Chapters 6 and 7 of my PhD Thesis, as well as in some published papers: 2012.14799, 2105.05095, 2110.12240, 2109.07405.

To see the code in action, go to this page.

CHAPTER

ONE

SMEFTGLOB

Common functions used to calculate likelihood values and pulls of the fits.

`SMEFT19.SMEFTglob.`**`fastmeas`**`(`*obs*`)`

Checks if the observable is part of a <<fast-measurement>> in smelli.

`SMEFT19.SMEFTglob.`**`likelihood_fits`**`(`*x*, *wfun*`)`

Calculates the log-likelihood of a NP hypothesis for several classes of observables.

Arguments

- x: Point in parameter space to be evaluated.
- wfun: Function that takes a point in parameter space and returns a dictionary of Wilson coefficents.

Returns

- A dictionary of log-likelihoods, for each of the classes of observables defined by *smelli*.

`SMEFT19.SMEFTglob.`**`likelihood_global`**`(`*x*, *wfun*`)`

Calculates the global log-likelihood of a NP hypothesis.

Arguments

- x: Point in parameter space to be evaluated.
- wfun: Function that takes a point in parameter space and returns a dictionary of Wilson coefficents.

Returns

- The global log-likelihood.

`SMEFT19.SMEFTglob.`**`loadobslist`**`(`*new=False*`)`

Loads from a *.yaml* file a list of all observables available, ordered by their pull in the SM. If the file does not exist, this functions creates it.

Returns

- A list with all observables available.

`SMEFT19.SMEFTglob.`**`newlist`**`()`

Creates a *.yaml* file with a list of all observables available, ordered by their pull in the SM.

`SMEFT19.SMEFTglob.`**`prediction`**`(`*x*, *obs*, *wfun*`)`

Interfaces *flavio* to compute the NP prediction of a given observable.

Arguments

- x: Point in parameter space to be evaluated.

- obs: observable, as defined by flavio, whose prediction will be computed. If the observable does not depend on any parameter, obs is a string. If the observable depends on numerical parameters (such as q2), obs is a list containing a string and one or more floats.
- wfun: Function that takes a point in parameter space and returns a dictionary of Wilson coefficents.

Returns

- The prediction of the observable.

`SMEFT19.SMEFTglob.`**`pull_obs`**`(`*x*, *obs*, *wfun*`)`

Calculates the pull, in sigmas, of the prediction of a given observable in NP with respect to its experimental value.

Arguments

- x: Point in parameter space to be evaluated.
- obs: observable, as defined by *flavio*, whose prediction will be computed. If the observable does not depend on any parameter, obs is a string. If the observable depends on numerical parameters (such as q2), obs is a list containing a string and one or more floats.
- wfun: Function that takes a point in parameter space and returns a dictionary of Wilson coefficents.

Returns

- The pull of the observable.

`SMEFT19.SMEFTglob.`**`restart_smelli`**`(`*include_likelihoods=None*, *add_measurements=None*, *remove_measurements=None*, *custom_likelihoods=None*`)`

Re-starts smelli's Global Likelihood with new parameters.

Arguments

- include_likelihoods: If not None, only the specified likelihoods will be included.
- add_measurements: Adds more experimental measurements not included by smelli.
- remove_measurements: Removes more experimental measurements not included by smelli.
- custom_likelihoods: Adds new likelihoods.

CHAPTER

TWO

SCENARIOS

This module contains all the NP hypothesis that we have considered and some auxiliar functions to implement them.

`SMEFT19.scenarios.`**`idemp`**(*a*, *b*)

Creates an idempotent hermitic 3x3 matrix using to parameters.

Arguments

- a, b: Parameter of the matrix.

Returns

- A *np.matrix*.

`SMEFT19.scenarios.`**`massrotation`**(*x*)

NP affects only the third generation in the interaction basis and then is rotated to the mass basis. Couplings to the first generation not negligible. C1 and C3 can be different.

Arguments

- x: Coordinates in the parameter space of the fit. x = [C1, C3, alpha_l, beta_l, alpha_q, beta_q].

Returns

- A dictionary containing the SMEFT Wilson Coefficients of the fit.

`SMEFT19.scenarios.`**`matrixwc`**(*num*, *C*, *ll*, *lq*)

Returns the Wilson coefficients for Clq1 or Clq3 given the parameters of the idemp matrix.

`SMEFT19.scenarios.`**`rot2lqU1`**(*x*, *M=1.5*)

Coupling of the U(1) leptoquarks obtained from the Wilson Coefficients.

Arguments

- x: Coordinates in the parameter space of the fit. It assumes scenario BI if len(x)==3 or scenario BII if len(x)==5.
- [M: Mass of the leptoquark, in TeV. Default=1.5.]

Returns

- A *np.matrix* containing the couplings.

`SMEFT19.scenarios.`**`rotBI`**(*x*)

Scenario BI: NP affects only the third generation in the interaction basis and then is rotated to the mass basis. Couplings to the first generation negligible.

Arguments

- x: Coordinates in the parameter space of the fit. x = [C, beta_l, beta_q].

Returns

- A dictionary containing the SMEFT Wilson Coefficients of the fit.

`SMEFT19.scenarios.`**`rotBII`**`(x)`

Scenario BII: NP affects only the third generation in the interaction basis and then is rotated to the mass basis. Couplings to the first generation not negligible.

Arguments

- x: Coordinates in the parameter space of the fit. x = [C, alpha_l, beta_l, alpha_q, beta_q].

Returns

- A dictionary containing the SMEFT Wilson Coefficients of the fit.

`SMEFT19.scenarios.`**`scI`**`(x)`

Scenario I: NP only affects to electrons.

Arguments

- x: Coordinates in the parameter space of the fit.

Returns

- A dictionary containing the SMEFT Wilson Coefficients of the fit.

`SMEFT19.scenarios.`**`scII`**`(x)`

Scenario II: NP only affects to muons.

Arguments

- x: Coordinates in the parameter space of the fit.

Returns

- A dictionary containing the SMEFT Wilson Coefficients of the fit.

`SMEFT19.scenarios.`**`scIII`**`(x)`

Scenario III: NP only affects to taus.

Arguments

- x: Coordinates in the parameter space of the fit.

Returns

- A dictionary containing the SMEFT Wilson Coefficients of the fit.

`SMEFT19.scenarios.`**`scIV`**`(x)`

Scenario IV: NP only affects to electrons and muons.

Arguments

- x: Coordinates in the parameter space of the fit.

Returns

- A dictionary containing the SMEFT Wilson Coefficients of the fit.

`SMEFT19.scenarios.`**`scIX`**`(x)`

Scenario IX: NP affects to electrons and taus equally, and to muons by an opposite ammount.

Arguments

- x: Coordinates in the parameter space of the fit.

Returns

- A dictionary containing the SMEFT Wilson Coefficients of the fit.

`SMEFT19.scenarios.`**`scV`**`(x)`

Scenario V: NP only affects to electrons and taus.

Arguments

- x: Coordinates in the parameter space of the fit.

Returns

- A dictionary containing the SMEFT Wilson Coefficients of the fit.

`SMEFT19.scenarios.`**`scVI`**`(x)`

Scenario VI: NP only affects to muons and taus.

Arguments

- x: Coordinates in the parameter space of the fit.

Returns

- A dictionary containing the SMEFT Wilson Coefficients of the fit.

`SMEFT19.scenarios.`**`scVII`**`(x)`

Scenario VII: NP affects to electrons, muons and taus.

Arguments

- x: Coordinates in the parameter space of the fit.

Returns

- A dictionary containing the SMEFT Wilson Coefficients of the fit.

`SMEFT19.scenarios.`**`scVIII`**`(x)`

Scenario VIII: NP affects to electrons, muons and taus equally.

Arguments

- x: Coordinates in the parameter space of the fit.

Returns

- A dictionary containing the SMEFT Wilson Coefficients of the fit.

`SMEFT19.scenarios.`**`scX`**`(x)`

Scenario X: NP affects to electrons and muons by an opposite ammount.

Arguments

- x: Coordinates in the parameter space of the fit.

Returns

- A dictionary containing the SMEFT Wilson Coefficients of the fit.

`SMEFT19.scenarios.`**`scXI`**`(x)`

Scenario XI: NP affects to electrons and to muons by an opposite ammount, and to taus independently.

Arguments

- x: Coordinates in the parameter space of the fit.

Returns

- A dictionary containing the SMEFT Wilson Coefficients of the fit.

CHAPTER

THREE

ELLIPSE

Assuming that the likelihood of the fit follows a gaussian distribution (Central Limit Theorem), and therefore the log-likelihood is characterized by a quadratic form around the minimum, this script finds this quadratic form, and parametrizes (ellipsoidal) sections of constant likelihood.

`SMEFT19.ellipse.`**`load`**(*filename*)

Loads a ellipse saved in a *.yaml* file to a python dictionary.

Arguments

- filename: Path to the *.yaml* file where the shape of the ellipse has been saved by the *save* method.

Returns

A *python* dictionary containing:

- bf: *np.array* with the point in parameter space with the best fit.
- v: *np.matrix* containing the orientation of the axes of the ellipsoid.
- d: *np.array* containing the principal axes of the ellipsoid.
- L: Log-likelihood at the best fit point.
- [name: Name of the fit.]
- [fit: Scenario used in the fit.]

`SMEFT19.ellipse.`**`minimum`**(*fit*, *x0*)

Finds the minimum of the fit function and approximates its neighbourhood by an ellipsoid.

Arguments

- fit: function that takes one point in parameter space and returns its negative log-likelihhod. Example: *-SMEFTglob.likelihood_global(x, scenarios.scVI)*.
- x0: list or *np.array* containing an initial guess.

Returns

- bf: np.array with the point in parameter space with the best fit.
- v: Unitary matrix containing the axes of the ellipse.
- d: diagonal matrix containing the inverse of the squares of the semiaxes.
- Lmin: Log-likelihood at the best fit point.

`SMEFT19.ellipse.`**`notablepoints`**(*fin*, *fout*, *fit*)

Finds the extrema of the ellipse, the intersection with the coordinate axis and the closest and furthest point from the origin.

Arguments

- fin: Path to *.yaml* file containing the information about the ellipse.
- fout: Path to *.tex* file to save a table with the coordinates of the notable points.
- fit: Function used in the minimization.

`SMEFT19.ellipse.`**`parametrize`**(*x*, *bf*, *v*, *d*, *nsigmas=1*)

Maps points on the unit hypersphere to points on the ellipsoid of constant likelihood.

Arguments

- x: *np.array* containing a point in the surface of the unit *n*-hypersphere.
- bf: *np.array* with the point in parameter space with the best fit.
- v: *np.matrix* containing the orientation of the axes of the ellipsoid.
- d: *np.array* containing the principal axes of the ellipsoid.
- [nsigmas: significance of the isoprobability hypersurface with respect to the best fit.]

Returns

- xe: Projection of the point x in the ellipsoid of equal probability

`SMEFT19.ellipse.`**`save`**(*bf*, *v*, *d*, *L*, *filename*, *name=None*, *fit=None*)

Saves the results of the minimization in a *.yaml* file.

Arguments

- bf: *np.array* with the point in parameter space with the best fit.
- v: *np.matrix* containing the orientation of the axes of the ellipsoid.
- d: *np.array* containing the principal axes of the ellipsoid
- filename: Path to the *.yaml* file where the shape of the ellipse will be saved.
- L: Log-likelihood at the best fit point.
- [name: Descriptive name of the fit.]
- [fit: scenario used to fit the data.]

CHAPTER

FOUR

OBSUNCERT

Module used to compute the uncertainty of some observables using a MonteCarlo analysis.

`SMEFT19.obsuncert.calculate`(*wfun*, *minx*, *maxx*, *fout*, *fin*, *name*, *num=50*, *cores=1*, *mode='exact'*)

Computes the central value and uncertainty of a selection of observables, using a MonteCarlo analysis. The observables are $R_{K^{(*)}}$ and $R_{D^{(*)}}$, and can be modified by editing the variable *obsuncert.obslist*.

Arguments

- wfun: Function that takes a point in parameter space and returns a dictionary of Wilson coefficents.
- minx: Minimum of the search region. If the fit is multidimensional, *minx* is a list containing the minimum of the search region in each direction.
- minx: Maximum of the search region. If the fit is multidimensional, *maxx* is a list containing the maximum of the search region in each direction.
- fout: Path to the *.yaml* file where the statistical values will be saved.
- bf: Coordinates of the best fit point.
- name: Name of the fit.
- [num: Number of MonteCarlo points used to compute the uncertainty. Default=50.]
- [cores: number of cores used to parallel-compute the uncertainty. Default=1 (no parallelization).]

CHAPTER

FIVE

COMPAREPULLS

This module contains several functions used to compare between different NP scenarios and the Standard Model.

SMEFT19.comparepulls.**compare**(*wfun*, *fin*, *fout*)

Lists the comparison between the pull of each observable in the NP hypothesis and the SM.

Arguments

- wfun: Function that takes a point in parameter space and returns a dictionary of Wilson coefficents.
- fin: Path to the file *.yaml* where the ellipsoid is saved.
- fout: Path to the *.tex* file where the comaparison table will be written. The observables are ordered by their SM pull, and are shaded in green if the NP improves this pull and in red otherwise.

SMEFT19.comparepulls.**notablepulls**(*wfun*, *fin*)

Determines the observables whose pull changes the most between the best fit and the notable points of the ellipsoid.

Arguments

- wfun: Function that takes a point in parameter space and returns a dictionary of Wilson coefficents.
- fin: Path to the file *.yaml* where the ellipsoid is saved.

SMEFT19.comparepulls.**pointpull**(*x*, *wfun*, *bf*, *printlevel=1*, *numres=5*)

Determines the observable whose pull changes the most between two NP hypothesis.

Arguments

- x: Point in space parameter of the tested NP hypothesis.
- wfun: Function that takes a point in parameter space and returns a dictionary of Wilson coefficents.
- bf: Point in space parameter of the reference NP hypothesis (e.g. the best fit).
- [printlevel: 0 for silent mode, 1 for verbose mode.]
- [numres: Number of observables displayed. Default=5.]

Returns

- A multi-line string. Each line contains the id number of the observable, its name and the squared difference of the pulls.

SMEFT19.comparepulls.**pullevolution**(*obscode*, *wfun*, *fin*, *direction*)

Calculates the variation of the pull along a line connecting two opposite notable points of the ellipsoid.

Arguments

- obscode: ID-Number of the observable, as returned by comparepulls.pointpull
- wfun: Function that takes a point in parameter space and returns a dictionary of Wilson coefficents.
- fin: Path to the file .yaml where the ellipsoid is saved.
- direction: string with the following format:
  - 'wc' + str(i): for the i-th Wilson coefficient.
  - 'ax' + str(i): for the i-th principal axis of the ellipsoid.
  - 'sm': for the direction joining the bf and sm points.

CHAPTER

SIX

PLOTS

This module contains functions to plot the results of the fits.

`SMEFT19.plots.`**`binerrorbox`**(*binmin*, *binmax*, *central*, *error*, *centralline=False*, *rect_args=None*, *line_args=None*)

Plots an error box.

Arguments

- binmin: Minimum *x* value of the error box.
- binmax: Maximum *x* value of the error box.
- central: Central *y* value.
- error: Error in the *y* direction. *error* must be a float if the error is symmetric, or a tuple (*error_inf*, *error_sup*) if the error is not symmetric.
- centralline: True to plot a horizontal line for the central value. Default: False.
- rect_args: Optional arguments passed to plot the box.
- line_args: Optional arguments passed to plot the central line.

`SMEFT19.plots.`**`compare_plot`**(*wfun*, *fin*, *fout*, *sigmas=1*)

Plots the pull of each observable in the SM and in the NP hypothesis.

Arguments

- wfun: Function that takes a point in parameter space and returns a dictionary of Wilson coefficents.
- fin: Path to the *.yaml* file where the ellipsoid is saved.
- fout: Path to the files where the plots will be saved. Two files are created, one *.pdf* and one *.pgf* (to use in TeX). Extensions are added automatically.

`SMEFT19.plots.`**`darken_color`**(*color*, *amount=0.5*)

Darkens the given color by multiplying luminosity by the given amount. Input can be matplotlib color string, hex string, or RGB tuple. Examples: >> darken_color('g', 0.3) >> darken_color('#F034A3', 0.6) >> darken_color((.3,.55,.1), 0.5)

`SMEFT19.plots.`**`error_plot`**(*fout*, *plottype*, *flist*, *flist2=None*, *legend=0*)

Plots the uncertainty intervals for several observables in NP scenarios, SM and experimental values.

Arguments

- fout: Path to the files where the plots will be saved. Two files are created, one *.pdf* and one *.pgf* (to use in TeX). Extensions are added automatically.
- plottype: Selects the observables to be plotted:

  - 'RK': Plots RK in the [1.1,6.0] bin and RK* in the [0.045,1.1] and [1.1,6] bins.
  - 'RD': Plots RD, and RD* using only muons or muons+electrons.
- flist: List of paths to files created by *obsuncert.calculate*.
- flist2: Additional list of paths to files created by *obsuncert.calculate*.
- legend: 0 for no legend, 1 for legend next to the plot and 2 for legend inside the plot.

`SMEFT19.plots.`**`evolution_plot`**(*obscodes*, *wfun*, *fin*, *direction*, *fout*, *obsnames=None*)

Plots the vairation of the pull of several observables along a line connecting two opposite notable points of the ellipsoid.

Arguments

- obscodes: List of ID-Numbers of the observables, as returned by *comparepulls.pointpull*
- wfun: Function that takes a point in parameter space and returns a dictionary of Wilson coefficents.
- fin: Path to the *.yaml* file where the ellipsoid is saved.
- direction: string with the following format:
  - 'ax' + str(i): for the i-th principal axis of the ellipsoid.
  - 'sm': for the direction joining the bf and sm points.
- fout: Path to the files where the plots will be saved. Two files are created, one *.pdf* and one *.pgf* (to use in TeX). Extensions are added automatically.

`SMEFT19.plots.`**`hatch_contour`**(*x*, *y*, *z*, *levels*, *interpolation_factor=1*, *interpolation_order=2*, *col=0*, *label=None*, *hatched=True*, *contour_args=None*, *contourf_args=None*)

Plots coloured and hatched confidence contours (or bands) given numerical input arrays.Based on the *flavio* function

Arguments

- x, y: 2D arrays containg x and y values as returned by numpy.meshgrid
- z: value of the function to plot. 2D array in the same shape as *x* and *y*. The lowest value of the function should be 0 (i.e. the best fit point).
- levels: list of function values where to draw the contours. They should be positive and in ascending order.
- [interpolation factor:: in between the points on the grid, the functioncan be interpolated to get smoother contours. This parameter sets the number of subdivisions (default: 1, i.e. no interpolation). It should be larger than 1.]
- [col: number between 0 and 9 to choose the color of the plot from a predefined palette.]
- [label: label that will be added to a legend created with *maplotlib.pyplot.legend()*.]
- [contour_args: dictionary of additional options that will be passed to *matplotlib.pyplot.contour()* (that draws the contour lines).]
- [contourf_args: dictionary of additional options that will be passed to *matplotlib.pyplot.contourf()* (that paints the contour filling).]

`SMEFT19.plots.`**`likelihood_plot`**(*grid*, *xmin*, *xmax*, *ymin*, *ymax*, *axlabels*, *fout=None*, *locleg=0*, *n_sigma=(1, 2)*, *colors=None*, *styles=None*, *widths=None*, *ticks=0.5*, *bf=None*)

Plots a contour plot of the log-likelihood of the fit.

Arguments

- grid: List containing the x coordinates, y corrdinates and a dictionary for the likelihood values in the grid.
- xmin: Minimum value of the *x* coordinate.
- xmax: Maximum value of the *x* coordinate.
- ymin: Minimum value of the *y* coordinate.
- ymax: Maximum value of the *y* coordinate.
- axlabels: List containing two strings to label the *x* and *y* axes.
- [fout: Path to the files where the plots will be saved. Two files are created, one *.pdf* and one *.pgf* (to use in TeX). Extensions are added automatically.]
- [locleg: Position of the legend of the plot, using *matplotlib*'s syntaxis. Default=0 (best position).]
- [n_sigma: List containing the significance (in sigmas) of each contour. Default = (1,2).]
- [colors: List with the colors of each contour. Default: flavio palette.]
- [styles: List with the linestyles of each contour. Default: All solid.]
- [widths: List with the linewidths of each contour. Default: All 1pt.]
- [ticks: Interval between ticks in both axes. Default:0.5]
- [bf: Coordinates of the best fit point(s). It can be *None* (no point marked) or a list containing two floats (one point marked). Default: *None*.]

CHAPTER

SEVEN

ML

This module contains the functions needed to train a Machine Learning-based Montecarlo scan, and to assess its performance.

`SMEFT19.ml.`**`SHAP_bf`**(*fmodel*, *bf*)
Computes the SHAP values of the best fit point

Arguments

- fmodel: Path to the file where the model was saved.
- bf: Best fit point.

`SMEFT19.ml.`**`SHAP_param`**(*fmodel*, *points*, *param*, *header=None*)
Creates an scatter plot displaying how the SHAP values change as functions of each parameter of the fit.

Arguments

- fmodel: Path to the file where the model was saved.
- points: Pandas Dataframe containing the dataset.
- param: Fit parameter. 0 = C, 1 = al, 2 = bl, 3 = aq, 4 = bq.
- header: If the data file contains headers in the first row, 0.

`SMEFT19.ml.`**`SHAP_summary`**(*fmodel*, *points*, *fout*, *header=None*)
Creates a summary plot of the average SHAP values on a dataset.

Arguments

- fmodel: Path to the file where the model was saved.
- points: Pandas Dataframe containing the dataset.
- fout: Path to save the plot (pdf only).
- header: If the data file contains headers in the first row, 0.

`SMEFT19.ml.`**`hist`**(*ML*, *vpoints*, *fout*)
Plots an histogram for the predicted and actual likelihoods, and compares them to the chi-square distribution

Arguments

- ML:The Machine Learning scan module.
- vpoints: Path to the file containing the points in the validation dataset.
- fout: Path to save the histogram.

`SMEFT19.ml.`**`lh`**(*x*)
Pickle-able function for the likelihood in scenario BII.

`SMEFT19.ml.`**`load_model`**(*fmodel*, *vpoints*, *bf*)

Loads a XGBoost model previously saved

Arguments

- fmodel: Path to the file where the model was saved.
- bf: Best fit point.

Returns

- Machine Learning scan.

`SMEFT19.ml.`**`regr`**(*ML*, *vpoints*, *fout*)

Plots the predicted likelihod vs the actual likelihood and computes their regression coefficient

Arguments

- ML:The Machine Learning scan module.
- vpoints: Path to the file containing the points in the validation dataset.
- fout: Path to the output regression plot (pdf only).

Returns

- A tuple containing the Perason r coefficient and the p-value of the regression

`SMEFT19.ml.`**`train`**(*dataset*, *fval*, *fmodel*, *bf*, *headers=None*)

Trains the Machine Learning algorithm with the previously computed Metropolis points

Arguments

- dataset: Path to the file or list of files containing the Montecarlo pre-computed points.
- fval: Path to the file where the validation points will be saved.
- fmodel: Path to the file where the XGBoost model will be saved.
- bf: Best fit point.
- headers: Header lines in the dataset files. None if there is no header, 0 if the first line contains the header. Admits list if using several dataset files.

Returns

- The Machine Learning scan module, already trained and ready to be used

CHAPTER EIGHT

UTILS

This module contains some auxiliary functions

`SMEFT19.utils.`**`distrsphere`**`(`*dim*`)`

Returns a random vector with norm 1.

Arguments

- dim: Dimension of the vector

`SMEFT19.utils.`**`listpoint`**`(`*x*`)`

If passed a single 2-tuple (representing one point), it returns a list containing the tuple. If passed more than 1 2-tuple (representing several points), returns the argument. TODO: Rewrite in python 3.10 using match&case syntax.

`SMEFT19.utils.`**`roundsig`**`(`*x*, *num=4*`)`

Rounds a number to a fixed number of significative digits.

`SMEFT19.utils.`**`sign`**`(`*x*, *y*`)`

Returns -1 if the first argument is strictly less than the second one, and 1 otherwise.

`SMEFT19.utils.`**`tex`**`(`*obs*`)`

Returns the TeX representation for a given flavio observable. If the observable includes arguments (e. g. q2), they are represented as superindex.

`SMEFT19.utils.`**`texnumber`**`(`*x*, *prec=3*`)`

Returns the TeX representation of a number in scientific notation.

PYTHON MODULE INDEX

INDEX