FoMpy Documentation

Release 0.2.0

Gabriel Espiñeira

Apr 01, 2019

CONTENTS:

CHAPTER

ONE

GETTING STARTED

FoMPy is an effective tool that extracts the main figures of merit (FoM) of a semiconductor's IV curve and provides useful statistical parameters for variability studies. It includes several methods to extract the threshold voltage.

In the figure below the used can see the basic workflow behind the FoMpy library:

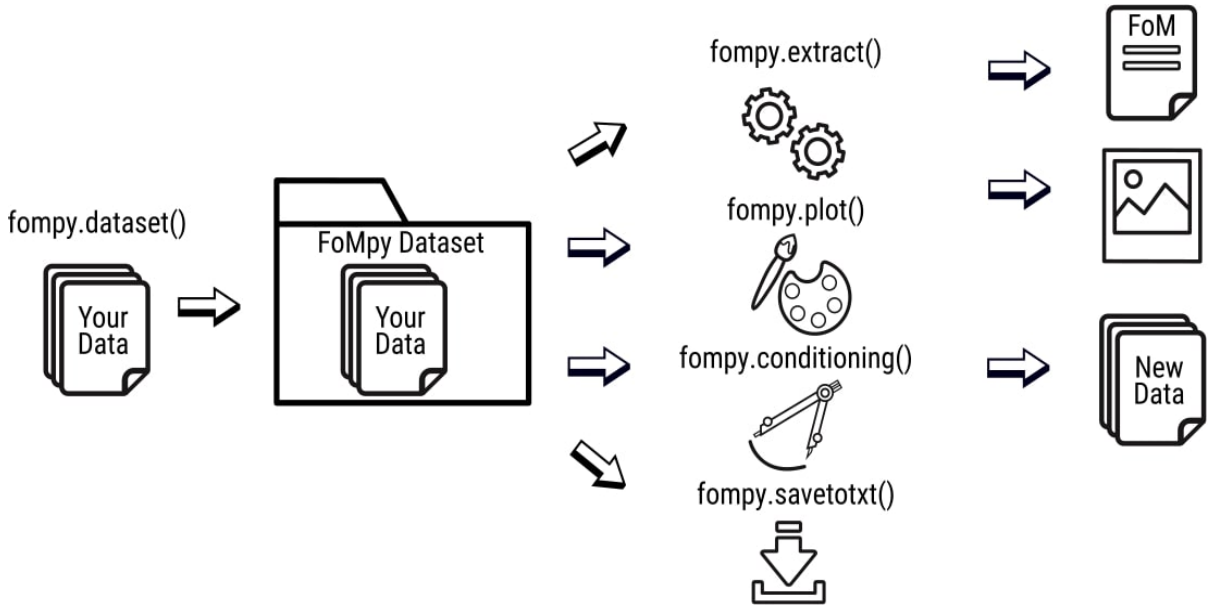

Fig. 1: Diagram showing the basic methodology of the FoMpy library. After loading the data into a FoMpy Dataset, using the various tools implemented in the library, the user is able to process and extract important parameters from a given dataset.

Next, a full list of all the components of the FoMpy library will be presented.

CHAPTER

TWO

MODULES

2.1 aux.py

This module includes several auxiliar routines used during the usage of FoMpy.

Example

In order find the closest index between a given value and a array of values, the function find_closest is used. For example during the extraction of IOFF:

```
voltage_index = find_closest(x_interp, vg_ext)
parameter.append(y_interp[voltage_index])
```

In order calculate the discrete derivate of a given curve the function get_diff is used as follows (second derivative method extraction):

```
d2 = get_diff(curve, order = 2)
lower_bound=find_closest(d2[:,0],10*d2[1,0])
higher_bound=find_closest(d2[2:,0],0.8*d2[-1,0])
warning_interval_limit_low = int(np.round(lower_bound*1.5))
warning_interval_limit_high = int(np.round(higher_bound*0.8))
vt_temp = np.argmax(d2[lower_bound:higher_bound,1])
```

where these the maximum value of the second derivative is found leaving out values outside a defined interval where the derivatives can be too noisy.

Finally, the function checkPath is used mostly in FoMpy to check if a directory exists and if it does not it creates it when saving either a plot or extracted results.

fompy.aux.**checkPath**(*cwd*)
Function that checks if a given path exists, and if it doesn't, it creates it.

Parameters **cwd** (*str*) – Path that the user wishes to check if it exists.

fompy.aux.**find_closest**(*A*, *target*)
Function that finds the closest item inside an array A to a target value. It returns the index of A with the closest value to the target ($A[idx] \approx target$).

Parameters

- **A** (*array_like*) – Array containing all the values to be compared. It usually is an array of currents of a FoMpy dataset.
- **target** (*float*) – Target value to be found inside the array A.

Returns **idx** – Index of the item contained in the array A with a value closest to the target.

Return type int

fompy.aux.**get_diff**(*arr*, *order*, *type='central'*)

Function for calculating the n-order derivative of a discrete array

Parameters

- **order** (*int*) – Order indicating the type of derivative to compute
- **type** (*str*) – The type of derivate can either be calculated in a forward way, central or backward, depending on the evaluation coordinates of the derivative function (see https://en.wikipedia.org/wiki/Finite_difference)

2.2 conditioning.py

This module includes the routines used to precondition the input data before any extraction is done. It includes routines for interpolating the data points, normalizing the data and filtering points that are too noisy.

Example

In order to normalize data the following steps have to be executed either in a script or python3 terminal:

```
import fompy
path_file_JCJB = './data/sim_FinFET_vd_high/'
fds = fompy.dataset(path_file_JCJB, parser=fompy.JCJB)
norm_value = 35.8/10**9
fompy.normalize(fds, norm_value)
```

In order to filter data the following steps have to be executed either in a script or python3 terminal:

```
import fompy
path_file_JCJB = './data/sim_FinFET_vd_high/'
fds = fompy.dataset(path_file_JCJB, parser=fompy.JCJB)
fompy.filter(fds, theta_crit = 1.52)
```

Additionally, by default FoMpy uses the cubic spline interpolation in order to extract accurately the FoMs. If the user wishes to change the interpolation methods, there are several implemented and tested methods that can be selected. The list of available methods includes: 'akima','pchip' or 'linear'. In order to change the interpolation method the following steps have to be executed either in a script or python3 terminal:

```
import fompy
path_file_JCJB = './data/sim_FinFET_vd_high/'
fds = fompy.dataset(path_file_JCJB, parser=fompy.JCJB)
fds.interpolation = 'linear'
```

fompy.conditioning.**angle_wrt_0**(*x*, *y*)

Function that calculates the polar angle of a point with respect to the origin of coordinates. It returns the angle in degrees and it is assummed that the angle is contained in the 1rst cuadrant.

Parameters

- **x** (*float*) – The x-axis coodinate of the point.
- **y** (*float*) – The y-axis coodinate of the point.

Returns **theta** – Angle between the given point and the origin of coordinates in degrees. It is assummed that the angle is contained in the 1rst cuadrant.

Return type float

class fompy.conditioning.**filter_tool**

Class for filtering noisy data from a semiconductor's IV curve.

...

polar_filter(*fds*, *theta_crit*, *show_theta = False*)

Class method that filters data from a semiconductor's IV curve checking the increase in the angle of the points with respect to the origin of an IV curve and removes all the increments that go above defined a threshold.

Parameters

- **fds** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve.
- **theta_crit** (*float*) – Threshold value for the biggest allowed increase in the angle between to consecutive points of an IV curve.
- **show_theta** (*bool*) – If True it prints all the angle increments between two consecutive points so the user can choose a suitable theta crit.

class fompy.conditioning.**interpolator**

Class containing several tested interpolation strategies that can be used in a semiconductor's IV curve. For further documentation go to https://docs.scipy.org/doc/scipy/reference/interpolate.html

...

spline_interpol(*x*, *y*, *n*, *d*, *s*)

Interpolate data with a piecewise cubic polynomial which is twice continuously differentiable [1]. The result is represented as a PPoly instance with breakpoints matching the given data. The natural boundary condition is selected by default.

Parameters

- **x** (*array_like, shape (n,)*) – 1-d array containing values of the independent variable.
- **y** (*array_like*) – Array containing values of the dependent variable. It can have arbitrary number of dimensions, but the length along axis must match the length of x. Values must be finite.
- **n** (*int*) – Length of the output interpolated array.
- **d** (*int*) – Degree of the smoothing spline. Must be <= 5. Default is k=3, a cubic spline.
- **s** (*float*) – Positive smoothing factor used to choose the number of knots.

akima_interpol(*x*, *y*, *n*)

Fit piecewise cubic polynomials, given vectors x and y. The interpolation method by Akima uses a continuously differentiable sub-spline built from piecewise cubic polynomials. The resultant curve passes through the given data points and will appear smooth and natural.

Parameters

- **x** (*array_like, shape (n,)*) – 1-D array of monotonically increasing real values.

- **y** (*array_like*) – N-D array of real values. The length of y along the first axis must be equal to the length of x.
- **n** (*int*) – Length of the output interpolated array.

pchip_interpol (*x*, *y*, *n*)
Convenience function for pchip interpolation. xi and yi are arrays of values used to approximate some function f, with yi = f(xi). The interpolant uses monotonic cubic splines to find the value of new points x and the derivatives there.

Parameters

- **x** (*array_like, shape (n,)*) – A 1-D array of monotonically increasing real values. x cannot include duplicate values (otherwise f is overspecified)
- **y** (*array_like*) – A 1-D array of real values. y's length along the interpolation axis must be equal to the length of x. If N-D array, use axis parameter to select correct axis.
- **n** (*int*) – Length of the output interpolated array.

lin_interpol (*x*, *y*, *n*)
Interpolate a 1-D function.x and y are arrays of values used to approximate some function f: y = f(x). This class returns a function whose call method uses interpolation to find the value of new points.

Parameters

- **x** (*array_like, shape (n,)*) – A 1-D array of real values.
- **y** (*array_like*) – A N-D array of real values. The length of y along the interpolation axis must be equal to the length of x.
- **n** (*int*) – Length of the output interpolated array.

class fompy.conditioning.**normalizer**
Normalizer class.

...

normalize (*fds*, *norm*)
Class method used for normalizing the currents a semiconductor's IV curve.

Parameters

- **fds** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve.
- **norm** (*float*) – float value used to normalize the currents contained in the FoMpy dataset IV curves.

2.3 fds.py

This module includes the routines used to import data into a FoMpy Dataset. Some useful examples showing how to import the data using these functions can be seen below:

Example

If the file parser is defined a simple voltage-current file is defined as input:

```
import fompy
path_file = './data/default/'
fds = fompy.dataset(path, parser=fompy.file)
```

or if an array is passed as input:

```
import fompy
#Here several arrays are defined
arr1 =np.array([[0.00e+00, 1.00e-09],[1.00e-01, 2.20e-08],[2.00e-01, 3.20e-07],[3.00e-
↪01, 2.74e-06],[4.00e-01, 9.90e-06],[5.00e-01, 2.20e-05],[6.00e-01, 3.22e-05],[7.00e-
↪01, 4.16e-05],[8.00e-01, 5.23e-05],[9.00e-01, 6.04e-05],[1.00e+00, 6.60e-05]])
arr2 =np.array([[0.00e+00, 1.00e-09],[1.00e-01, 2.15e-08],[2.00e-01, 3.18e-07],[3.00e-
↪01, 2.72e-06],[4.00e-01, 9.85e-06],[5.00e-01, 2.12e-05],[6.00e-01, 3.16e-05],[7.00e-
↪01, 4.10e-05],[8.00e-01, 5.46e-05],[9.00e-01, 6.15e-05],[1.00e+00, 6.57e-05]])
arrays = np.stack((arr1, arr2)) #Here the arrays are put together
fds = fompy.dataset(arr = arrays, parser=fompy.array)

#Also for a single IV curve
x = ([0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0])
y = ([1.00e-09, 2.20e-08, 3.20e-07, 2.74e-06, 9.90e-06, 2.20e-05, 3.22e-05, 4.16e-05,␣
↪5.23e-05, 6.04e-05, 6.60e-05])
fds = fompy.iv(arr = (x,y), parser=fompy.curve)
```

Moreover, when parsers for another file-types are be selected:

```
import fompy
path_file_JCJB = './data/sim_FinFET_vd_high/'
fds = fompy.dataset(path_file_JCJB, parser=fompy.JCJB)

path_file_mc = './data/mc_data/'
fds = fompy.dataset(path_file_mc, parser=fompy.MC)
```

If the user wishes to remove several IV curves included in the parent folder, two options called 'interval' and 'exclude' can be passed, so the indexes defined are removed from the FoMpy Dataset:

```
import fompy
path_file_JCJB = './data/sim_FinFET_vd_high/'
fds = fompy.dataset(path_file_JCJB, parser=fompy.JCJB, exclude=[5,6])
fds = fompy.dataset(path_file_JCJB, parser=fompy.JCJB, interval=[0,8])
```

class fompy.fds.**FompyDataset**(***kwargs*)

Class containing the simulated IV curves and their parameters.

...

dataset

List containing all IV curves

Type array: double[]

n_sims

Number of simulated IV curves

Type int

sanity_array
Array of ones by default. If a simulation has failed, its index is converted to zero.

Type array: double[]

norm
Normalization value applied to the IV curve

Type double

ext_method
FompyDataset default method used to extrad the figures of merit

Type str

drain_bias_label
Either high or low.

Type str

drain_bias_value
Drain voltage value used to simulate the IV curves.

Type double

interpolation
IV curve interpolation method. Can either be 'cubic_spline', 'akima','pchip' or 'linear'.

Type str

filter
IV curve filter method. So far only the polar filter has been implemented ('polar_filter').

Type str

print_parameters()
Prints all the non-array attributes.

fompy.fds.**JCJB**(*fds*, *path*, *path_subdirs*, *path_filenames*, *interval*, *exclude*)
Function that imports the simulated data from a JCJB file and stores it into a FoMpy Dataset.

Parameters

- **fds** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve.
- **path** (*str*) – Parent path where the simulations are stored
- **path_subdirs** (*str*) – List of alphabetically sorted subdirectories found inside the parent directory.
- **path_filenames** (*str*) – List of alphabetically sorted files found inside the parent directory.
- **interval** (*array_like*) – List of two int values: start(index of the first simulation to load into the Fompy Dataset) and end(index of the last simulation to load into the Fompy-Dataset)
- **exclude** (*array_like*) – Index values of simulations to exclude.

Returns **fds** – Structure of data containing the FoMpy Dataset.

Return type FoMpy Dataset

fompy.fds.**MC**(*fds*, *path*, *path_subdirs*, *path_filenames*, *interval*, *exclude*)
Function that imports the simulated data from a MC set of files and stores it into a FoMpy Dataset.

Parameters

- **fds** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve.
- **path** (*str*) – Parent path where the simulations are stored
- **path_subdirs** (*str*) – List of alphabetically sorted subdirectories found inside the parent directory.
- **path_filenames** (*str*) – List of alphabetically sorted files found inside the parent directory.
- **interval** (*array_like*) – List of two int values: start(index of the first simulation to load into the Fompy Dataset) and end(index of the last simulation to load into the Fompy-Dataset)
- **exclude** (*array_like*) – Index values of simulations to exclude.
- **skiprows** (*int*) – Number of rows to skip at the begining of a file. 0 rows are skipped by default.
- **comments** (*str*) – All the lines starting with this character are considered comments. '#' is used by default.

Returns **fds** – Structure of data containing the FoMpy Dataset.

Return type FoMpy Dataset

fompy.fds.**array**(*fds*, *arr=None*)

Function that imports the simulated data from a given an array and stores it into a FoMpy Dataset.

Parameters

- **fds** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve.
- **arr** (*array_like*) – Array of data containing one or more semiconductor's IV curves.

Returns **fds** – Structure of data containing the FoMpy Dataset.

Return type FoMpy Dataset

fompy.fds.**curve**(*fds*, *iv=None*)

Function that imports the simulated data from a single array and stores it into a FoMpy Dataset.

Parameters

- **fds** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve.
- **iv** (*array_like*) – Single IV curve.

Returns **fds** – Structure of data containing the FoMpy Dataset.

Return type FoMpy Dataset

class fompy.fds.**daoFile**(*parser=None*)

Data Acces Object used to import from a set of files the simulated IV curves to a FompyDataset.

...

parser

Type of parser, user-defined, that imports the simulated IV curves in a specific format.

Type str

load(*path=None*, *arr=None*, *iv=None*, *parser=None*, *interval=None*, *exclude=None*)

load(*path*, *filename_user = None*, *parser = None*, *interval = None*, *exclude = None*)
Class method that extracts V_{TH} of a semiconductor's IV curve.

Parameters

- **path** (*array_like, shape (n,)*) – 1-d array containing values of the independent variable.
- **arr** (*array_like*) – Array of data containing one or more semiconductor's IV curves.
- **iv** (*array_like*) – Single IV curve.
- **parser** (*function*) – Function that implements how the data is imported to a Fompy Dataset. The list of available functions includes: 'file','array','JCJB' and 'MC'.
- **interval** (*array_like*) – List of two int values: start(index of the first simulation to load into the Fompy Dataset) and end(index of the last simulation to load into the Fompy-Dataset)
- **exclude** (*array_like*) – Index values of simulations to exclude.
- **skiprows** (*int*) – Number of rows to skip at the begining of a file. 0 rows are skipped by default.
- **comments** (*str*) – All the lines starting with this character are considered comments. '#' is used by default.

class fompy.fds.**dataDAO**
Data Acces Object Interface used to import the simulated IV curves to a FompyDataset.

fompy.fds.**exclude_indexes**(*fds*, *interval*, *exclude*)
Function that generates an array of zeros and ones. If the simulation number 5 has failed, either because the folder is empty, or not enough voltages have been simulated, then fds.sanity_array[4] is set to zero.

Parameters

- **fds** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve.
- **interval** (*array_like*) – List of two int values: start(index of the first simulation to load into the Fompy Dataset) and end(index of the last simulation to load into the Fompy-Dataset)
- **exclude** (*array_like*) – Index values of simulations to exclude.

Returns **fds** – Structure of data containing the sanity array.

Return type FoMpy Dataset

fompy.fds.**file**(*fds*, *path*, *path_subdirs*, *path_filenames*, *interval*, *exclude*)
Function that imports the simulated data from a given file and stores it into a FoMpy Dataset. The format of this file is asummed to have comments starting with '#', without a header and two columns, one for the voltage and one for the currents, separated by ' ' delimiters.

Parameters

- **fds** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve.
- **path** (*str*) – Parent path where the simulations are stored

- **path_subdirs** (*str*) – List of alphabetically sorted subdirectories found inside the parent directory.
- **path_filenames** (*str*) – List of alphabetically sorted files found inside the parent directory.
- **interval** (*array_like*) – List of two int values: start(index of the first simulation to load into the Fompy Dataset) and end(index of the last simulation to load into the Fompy-Dataset)
- **exclude** (*array_like*) – Index values of simulations to exclude.

Returns **fds** – Structure of data containing the FoMpy Dataset.

Return type FoMpy Dataset

2.4 fom.py

This module includes the routines used to extract several commonly used FoMs in semiconductor simulations.

Example

After the data is imported into a FoMpy Dataset, the user may use several routines implemented in FoMpy to extract the 'vth', 'ioff', 'ion', 'ss', 'ratio', 'power' or 'dibl'. Code examples explaining how to use them can be seen below.

In order to extrac the vth of a given IV curve, the following commands have to be used either in a script or in a python3 command line:

```
import fompy
path_file_high = './data/sim_FinFET_vd_high/'
fds = fompy.dataset(path_file_JCJB, parser=fompy.JCJB)
fds.drain_bias_label = 'High'
vth_array = fompy.extract(fds, fom = 'vth')
```

If you are not using a JCJB file, the attribute of drain_bias_label has to be defined when using the data of a FoMpy dataset at high drain bias, otherwise low drain bias formulas will be used. Also if a different FoM is needed, the user has to change the keyword fom, from 'vth' to another one contained in the list shown before.

Additionally, if the user wants to obtain the DIBL, two different FoMpy Datasets have to be imported:

```
import fompy
path_file_JCJB = './data/sim_FinFET_vd_high/'
path_file_low = './data/sim_FinFET_vd_low/'

fds_hdb = fompy.dataset(path_file_JCJB, parser='JCJB')
fds_ldb = fompy.dataset(path_file_low, parser='JCJB')
fds_hdb.drain_bias_value = 0.7
fds_ldb.drain_bias_value = 0.05
dibl_array = fompy.extract(fds_hdb, fds_ldb, fom = 'dibl')
```

class fompy.fom.**dibl_ext**

extraction(*fds1*, *fds2*, *method=None*, *cc_criteria=None*)

extraction(*fds1*, *fds2*, *method = None*)

Class method that extracts the $DIBL$ of two given semiconductor's IV curves.

Parameters

- **fds1** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve. Needed for the extraction of any FoM.
- **fds2** (*FoMpy Dataset*) – additional structure of data containing the most important parameters of a semiconductor's IV curve. Needed for generating the plot of the calibration and the DIBL.
- **method** (*str*) – Keyword indicating the desired method of extraction of the FoMs. The list of available methods includes: 'SD', 'CC', 'TD' and 'LE'. If method is not defined the 'SD' is selected by default.

plot (*fds1*, *parameter = None*, *method = None*, *cc_crit = None*, *curves = None*, *save = None*, *A=None*, *B=None*)
Class method that plots the extracted $DIBL$ values.

Parameters

- **fds1** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve. Needed for generating the plot of any FoM.
- **curve_high** (*array_like*) – Array of data containing the IV curves at high drain bias.
- **curve_low** (*array_like*) – Array of data containing the IV curves at low drain bias.
- **parameter_vt_high** (*float*) – Voltage value of high drain bias.
- **parameter_vt_low** (*float*) – Voltage value of low drain bias.
- **corriente_low** (*float*) – Current at the vth value extracted for the curve at low drain bias.
- **backend** (*str*) – String containing the name of the backend chosen to either plot or save the plots. The backends available are: 'Agg', which only works whenever saving plots to files (non-GUI) and 'TkAgg' a GUI tools for visualizing the plots. 'TkAgg' requires the package python3-tk installed in order to run.
- **save_plot** (*bool*) – If True the generated plot is save to the defined path.

save_results_to_file (*path*, *parameter*)
Class method that saves the extracted $DIBL$ values.

Parameters

- **path** (*str*) – Defines the path where the extracted results are saved to a file.
- **parameter** (*array_like*) – Array of extracted FoM values to be saved into the file.

fompy.fom.**interpol** (*x=None*, *y=None*, *n=None*, *strategy=None*, *d=None*, *s=None*)
Wrapper function for interpolating imported data from a semiconductor's IV curve.

Parameters

- **x** (*array_like, shape (n,)*) – 1-d array containing values of the independent variable.
- **y** (*array_like*) – Array containing values of the dependent variable. It can have arbitrary number of dimensions, but the length along axis must match the length of x. Values must be finite.
- **strategy** (*str*) – Keyword for defining the selected interpolation method: The list of available methods includes: 'akima', 'pchip' and 'linear'.

- **d** (*int*) – Degree of the smoothing spline. Must be <= 5. Default is k=3, a cubic spline.
- **s** (*float*) – Positive smoothing factor used to choose the number of knots.

class fompy.fom.**ioff_ext**

Child class of _extractor that obtains the I_{OFF} figure of merit from a semiconductor's IV curve.

extraction(*fds1=None*, *vg_ext=None*)

extraction(*fds1*, *method=None*, *cc_criteria = None*)

Class method that extracts I_{OFF} of a semiconductor's IV curve.

Parameters

- **fds1** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve. Needed for the extraction of any FoM.
- **vg_ext** (*float*) – Gate voltage value used to calculate IOFF at.

plot(*fds1*, *parameter = None*, *method = None*, *cc_crit = None*, *curves = None*, *save = None*, *A=None*, *B=None*)

Class method that plots the extracted I_{OFF} values.

Parameters

- **fds1** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve. Needed for generating the plot of any FoM.
- **parameter** (*array_like*) – Array of extracted FoM values to be plotted.
- **vg_ext** (*float*) – Gate voltage value used to calculate IOFF at.
- **curves** (*array_like*) – Array of data containing the IV curves.
- **backend** (*str*) – String containing the name of the backend chosen to either plot or save the plots. The backends available are: 'Agg', which only works whenever saving plots to files (non-GUI) and 'TkAgg' a GUI tools for visualizing the plots. 'TkAgg' requires the package python3-tk installed in order to run.
- **save_plot** (*bool*) – If True the generated plot is save to the defined path.

save_results_to_file(*path*, *parameter*)

Class method that saves the extracted I_{OFF} values.

Parameters

- **path** (*str*) – Defines the path where the extracted results are saved to a file.
- **parameter** (*array_like*) – Array of extracted FoM values to be saved into the file.

class fompy.fom.**ion_ext**

Child class of _extractor that obtains the I_{ON} figure of merit from a semiconductor's IV curve.

extraction(*fds1*, *vg_ext=None*, *vth=None*)

extraction(*fds1*, *vg_ext = None*, *vth = None*)

Class method that extracts I_{ON} of a semiconductor's IV curve.

Parameters

- **fds1** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve. Needed for the extraction of any FoM.

- **vg_ext** (*float*) – Gate voltage value used to calculate IOFF at.
- **vth** (*array_like*) – Array of vth extracted values used for obtaining ION (using the vth-dependant formula).

plot(*fds1*, *parameter = None*, *method = None*, *cc_crit = None*, *curves = None*, *save = None*, *A=None*, *B=None*)
Class method that plots the extracted I_{ON} values.

Parameters

- **fds1** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve. Needed for generating the plot of any FoM.
- **parameter** (*array_like*) – Array of extracted FoM values to be plotted.
- **curves** (*array_like*) – Array of data containing the IV curves.
- **parameter_vth** (*array_like*) – Array of extracted vth values, as a method to obtain ION depends on them.
- **vg_ext** (*float*) – Gate voltage value used to calculate IOFF at.
- **backend** (*str*) – String containing the name of the backend chosen to either plot or save the plots. The backends available are: 'Agg', which only works whenever saving plots to files (non-GUI) and 'TkAgg' a GUI tools for visualizing the plots. 'TkAgg' requires the package python3-tk installed in order to run.
- **save_plot** (*bool*) – If True the generated plot is save to the defined path.

save_results_to_file(*path*, *parameter*)
Class method that saves the extracted I_{ON} values.

Parameters

- **path** (*str*) – Defines the path where the extracted results are saved to a file.
- **parameter** (*array_like*) – Array of extracted FoM values to be saved into the file.

class fompy.fom.**ss_ext**

extraction(*fds1*, *vth=None*, *vg_start=None*, *vg_end=None*)

extraction(*fds1*, *vg_start = None*, *vg_end = None*)
Class method that extracts SS of a semiconductor's IV curve.

Parameters

- **fds1** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve. Needed for the extraction of any FoM.
- **vth** (*array_like*) – Array of vth extracted values used for obtaining SS (using the vth-dependant formula).
- **vg_start** (*float*) – Gate voltage defining the start of the interval in which the Subthreshold Swing is extracted.
- **vg_end** (*float*) – Gate voltage defining the end of the interval in which the Subthreshold Swing is extracted.

plot (*fds1*, *parameter = None*, *method = None*, *cc_crit = None*, *curves = None*, *save = None*, *A=None*, *B=None*)
Class method that plots the extracted SS values.

Parameters

- **fds1** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve. Needed for generating the plot of any FoM.
- **parameter** (*array_like*) – Array of extracted FoM values to be plotted.
- **curves** (*array_like*) – Array of data containing the IV curves.
- **parameter_ss** (*array_like*) – Array of extracted ss values.
- **vg_start** (*float*) – Gate voltage defining the start of the interval in which the Subthreshold Swing is extracted.
- **vg_end** (*float*) – Gate voltage defining the end of the interval in which the Subthreshold Swing is extracted.
- **vg_sd_medio** (*float*) – Value in the middle of the interval between zero and vth extracted with the SD method. It is used only for defining a limit in the plot.
- **backend** (*str*) – String containing the name of the backend chosen to either plot or save the plots. The backends available are: 'Agg', which only works whenever saving plots to files (non-GUI) and 'TkAgg' a GUI tools for visualizing the plots. 'TkAgg' requires the package python3-tk installed in order to run.
- **save_plot** (*bool*) – If True the generated plot is save to the defined path.

save_results_to_file (*path*, *parameter*)
Class method that saves the extracted SS values.

Parameters

- **path** (*str*) – Defines the path where the extracted results are saved to a file.
- **parameter** (*array_like*) – Array of extracted FoM values to be saved into the file.

class fompy.fom.**vth_ext**
Child class of _extractor that obtains the V_{TH} figure of merit from a semiconductor's IV curve.

extraction (*fds1*, *method=None*, *cc_criteria=None*)

extraction (*fds1*, *method=None*, *cc_criteria = None*)
Class method that extracts V_{TH} of a semiconductor's IV curve.

Parameters

- **fds1** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve. Needed for the extraction of any FoM.
- **method** (*str*) – Keyword indicating the desired method of extraction of the FoMs.The list of available methods includes: 'SD', 'CC', 'TD' and 'LE'. If method is not defined the 'SD' is selected by default.
- **cc_criteria** (*float, optional*) – Float value used for the extraction of several FoMs using the constant current method.

plot (*fds1*, *parameter = None*, *method = None*, *cc_crit = None*, *curves = None*, *save = None*, *A=None*, *B=None*)
Class method that plots the extracted V_{TH} values.

Parameters

- **fds1** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve. Needed for generating the plot of any FoM.
- **parameter** (*array_like*) – Array of extracted FoM values to be plotted.
- **method** (*str*) – Keyword indicating the desired method of extraction of the FoMs. The list of available methods includes: 'SD', 'CC', 'TD' and 'LE'. If method is not defined the 'SD' is selected by default.
- **cc_criteria** (*float*) – Current criteria used to extract vth with the CC criteria for the fomplot.
- **curves** (*array_like*) – Array of data containing the IV curves.
- **backend** (*str*) – String containing the name of the backend chosen to either plot or save the plots. The backends available are: 'Agg', which only works whenever saving plots to files (non-GUI) and 'TkAgg' a GUI tools for visualizing the plots. 'TkAgg' requires the package python3-tk installed in order to run.
- **save_plot** (*bool*) – If True the generated plot is save to the defined path.
- **B** (*A,*) – Parameters obtained during the vth LE extraction method used for the plots.

save_results_to_file (*path*, *parameter*)

Class method that saves the extracted V_{TH} values.

Parameters

- **path** (*str*) – Defines the path where the extracted results are saved to a file.
- **parameter** (*array_like*) – Array of extracted FoM values to be saved into the file.

2.5 plots.py

This module includes the routines used to generate several common figures in semiconductor simulations.

Example

After the FoMs have been extracted, FoMpy includes several useful visualizing routines, commonly used in semiconductor simulations. Code examples explaining how to use them can be seen below.

In order to generate a plot of the FoMs (fomplot):

```
import fompy
path_file_high = './data/sim_FinFET_vd_high/'
fds = fompy.dataset(path_file_JCJB, parser=fompy.JCJB)
vth_array = fompy.extract(fds, fom = 'vth')
fompy.plot(fds, fom = 'vth', save_plot='./vth_plots/sd/')
```

and FoMpy will generate a single graph for each simulation showing the IV curve with its correspondent FoM extraction criteria and the extracted FoM.

Most of the FoMs are plotted the same way, except for the DIBL. This is because, if we want to calculate the DIBL two FoMpy Datasets are needed:

```
import fompy
path_file_JCJB = './data/sim_FinFET_vd_high/'
path_file_low = './data/sim_FinFET_vd_low/'

fds_hdb = fompy.dataset(path_file_JCJB, parser='JCJB')
fds_ldb = fompy.dataset(path_file_low, parser='JCJB')
fds_hdb.drain_bias_value = 0.7
fds_ldb.drain_bias_value = 0.05

fompy.plot(fds_hdb, fds_ldb, fom = 'dibl', save_plot='./dibl/')
```

Additionally, other types of plots can be generated. The list of available plots includes: 'iv', 'hist', 'qq', 'varplot', 'calib' and 'fomplot'. These are some examples:

```
path_file_var = './data/simulations/'
fds_var = fompy.dataset(path_file_var, parser=fompy.JCJB)
vth_array = fompy.extract(fds_var, fom = 'vth')

fompy.plot(fds, plot_type='hist', parameter=vth_array, bins=10,cont_parameter= 0.38,␣
↪save_plot='./variability/')
fompy.plot(fds, plot_type='qq', parameter=vth_array, save_plot='./variability/')

path_file_var = './data/simulations/'
fds_var = fompy.dataset(path_file_var, parser=fompy.JCJB)
fompy.plot(fds, plot_type='varplot', save_plot='./variability/')

path_file_JCJB = './data/sim_FinFET_vd_high/'
path_file_low = './data/sim_FinFET_vd_low/'

fds_hdb = fompy.dataset(path_file_JCJB, parser=fompy.JCJB)
fds_hdb.print_parameters()
fds_ldb = fompy.dataset(path_file_low, parser=fompy.JCJB)
fds_ldb.print_parameters()

norm_value = 35.8/10**9
fompy.normalize(fds_hdb, norm_value)
fompy.normalize(fds_ldb, norm_value)

fompy.plot(fds_hdb,fds_ldb, plot_type='calib', save_plot='./calibration/')
```

fompy.plots.**interpol**(*x=None*, *y=None*, *n=None*, *strategy=None*, *d=None*, *s=None*)

Wrapper function for interpolating imported data from a semiconductor's IV curve.

Parameters

- **x** (*array_like, shape (n,)*) – 1-d array containing values of the independent variable.
- **y** (*array_like*) – Array containing values of the dependent variable. It can have arbitrary number of dimensions, but the length along axis must match the length of x. Values must be finite.
- **strategy** (*str*) – Keyword for defining the selected interpolation method: The list of available methods includes: 'akima', 'pchip' and 'linear'.
- **d** (*int*) – Degree of the smoothing spline. Must be <= 5. Default is k=3, a cubic spline.
- **s** (*float*) – Positive smoothing factor used to choose the number of knots.

class fompy.plots.**plotStrategy**
Abstract class containing several commonly used plots for semiconductor simulations.

class fompy.plots.**plotter**
Class used for generating common plots in semiconductor simulations. For extensive documentation on how to modify this code go to https://matplotlib.org/tutorials/index.html

fomplot(*i*, *fds1*, *fom=None*, *currents=None*, *voltages=None*, *parameter=None*, *method=None*, *cc_criteria=None*, *parameter_vth=None*, *vg_ext=None*, *curve_high=None*, *curve_low=None*, *vth_high=None*, *vth_low=None*, *corriente_low=None*, *backend=None*, *save_plot=None*, *A=None*, *B=None*, *vg_start=None*, *vg_end=None*, *vt_sd_medio=None*)

fomplot(i, fom = None, voltages = None, currents = None, parameter = None,method =

vg_ext = None, curve_high = None, curve_low = None, vth_high = None, vth_low = None
Plot the most common figures of merit of a semiconductor's IV curve.

Parameters

- **fom** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve.
- **currents** (*path or None, optional*) – Path indicating the folder where the user wishes to save the generated plots.
- **voltages** (*path or None, optional*) – Path indicating the folder where the user wishes to save the generated plots.
- **backend** (*str*) – String containing the name of the backend chosen to either plot or save the plots. The backends available are: 'Agg', which only works whenever saving plots to files (non-GUI) and 'TkAgg' a GUI tools for visualizing the plots.'TkAgg' requires the package python3-tk installed in order to run.

hist(*bins=None*, *parameter=None*, *cont_parameter=None*, *backend=None*, *save_plot=None*)

hist(bins = None, parameter = None, save_to_file = None):
Plot a histogram.The return value is a tuple (n, bins, patches) or ([n0, n1, ...], bins, [patches0, patches1,...]) if the input contains multiple data.

Parameters

- **bins** (*int*) – If an integer is given, bins + 1 bin edges are calculated and returned, consistent with numpy.histogram
- **parameter** (*array_like, shape (n,)*) – Input values, this takes either a single array or a sequence of arrays which are not required to be of the same length.
- **save_plot** (*path or None, optional*) – Path indicating the folder where the user wishes to save the generated plots.

iv(*fds*, *backend=None*, *save_plot=None*)

iv(*fds*, *save_plot = None*)
Class method that filters data from a semiconductor's IV curve using Gaussian filtering.

Parameters

- **fds** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve.

- **save_plot** (*path or None, optional*) – Path indicating the folder where the user wishes to save the generated plots.

qq(*parameter*, *backend=None*, *save_plot=None*)

qq(parameter, save_to_file = None):
Plot a Quantile plot.Plotting positions are converted into quantiles or Z-scores based on a probability distribution

Parameters

- **parameter** (*array_like, shape (n,)*) – Input values, this takes either a single array or a sequence of arrays which are not required to be of the same length.
- **save_plot** (*path or None, optional*) – Path indicating the folder where the user wishes to save the generated plots.

varplot(*fds*, *backend=None*, *save_plot=None*)

varplot(self,fds, save_plot = None):
Plot all the IV curves for a common variability source.

Parameters

- **fds** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve.
- **save_plot** (*path or None, optional*) – Path indicating the folder where the user wishes to save the generated plots.

2.6 wrappers.py

This module includes all the wrapper functions used in the FoMpy library. Code examples on how to use them can be seen either in the complete guide, the repository quickstart or at the beginning of each file of the source code.

fompy.wrappers.**dataset**(*path=None*, *arr=None*, *parser=None*, *save_to_file=None*, *interval=None*, *exclude=None*)
Wrapper function that creates a FoMpy dataset.

Parameters

- **path** (*str*) – Path to the file containing the IV curves
- **arr** (*array_like*) – Array of data containing one or more semiconductor's IV curves.
- **parser** (*void*) – Format to read the file
- **save_to_file** (*str*) – Path of the file to store all the FoMpy dataset
- **interval** (*array_like*) – List of two int values: start(index of the first simulation to load into the FompyDataset) and end(index of the last simulation to load into the Fompy-Dataset)
- **exclude** (*array_like*) – Index values of simulations to exclude.

Returns Class containing the most important parameters of a semiconductor IV curve

Return type FompyDataset

fompy.wrappers.**extract**(*fds1*, *fds2=None*, *fom=None*, *method=None*, *cc_criteria=None*, *vg_ext=None*, *print_fom=None*, *vg_start=None*, *vg_end=None*)

Wrapper function that extracts the most common figures of merit from a FoMpy dataset.

Parameters

- **fds1** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve. Needed for the extraction of any FoM.
- **fds2** (*FoMpy Dataset*) – additional structure of data containing the most important parameters of a semiconductor's IV curve. Needed for the extraction of the DIBL.
- **fom** (*str*) – Keyword indicating the desired FoMs to extract. The list of available FoMs includes: 'vth', 'ioff', 'ion', 'ss', 'ratio', 'power' and 'dibl'.
- **method** (*str*) – Keyword indicating the desired method of extraction of the FoMs.The list of available methods includes: 'SD', 'CC', 'TD' and 'LE'. If method is not defined the 'SD' is selected by default.
- **cc_criteria** (*float, optional*) – Float value used for the extraction of several FoMs using the constant current method.
- **vg_ext** (*float*) – Gate voltage value used to calculate a FoM like IOFF or ION.
- **print_fom** (*bool*) – If True all the FoMs are extracted from a FoMpy Dataset (except for the DIBL)
- **vg_start** (*float*) – Gate voltage defining the start of the interval in which the Subthreshold Swing is extracted.
- **vg_end** (*float*) – Gate voltage defining the end of the interval in which the Subthreshold Swing is extracted.

Returns **parameter** – 1-d array containing the extracted FoMs.

Return type array_like

fompy.wrappers.**filter**(*fds1*, *theta_crit*, *show_theta=False*)

Wrapper class for filtering noisy data from a semiconductor's IV curve.

Parameters

- **fds** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve.
- **theta_crit** (*float*) – Threshold value for the biggest allowed increase in the angle between to consecutive points of an IV curve.
- **show_theta** (*bool*) – If True it prints all the angle increments between two consecutive points so the user can choose a suitable theta crit.

fompy.wrappers.**iv**(*path=None*, *arr=None*, *parser=None*, *save_to_file=None*)

Wrapper function that creates a FoMpy dataset from a single IV curve.

Parameters

- **path** (*str*) – Path to the file containing the IV curves
- **arr** (*array_like*) – Single IV curve.
- **parser** (*void*) – Format to read the file
- **save_to_file** (*str*) – Path of the file to store all the FoMpy dataset

Returns Class containing the most important parameters of a semiconductor IV curve

Return type FompyDataset

fompy.wrappers.**normalize**(*fds*, *norm*)
Wrapper function that normalizes the currents a semiconductor's IV curve.

Parameters

- **fds** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve.
- **norm** (*float*) – float value used to normalize the currents contained in the FoMpy dataset IV curves.

fompy.wrappers.**plot**(*fds1*, *fds2=None*, *plot_type=None*, *fom=None*, *parameter=None*, *method=None*, *bins=None*, *cc_criteria=None*, *vg_ext=None*, *vg_start=None*, *vg_end=None*, *cont_parameter=None*, *backend=None*, *save_plot=None*)
Wrapper function that plots the most common figures in semiconductor simulations.

Parameters

- **fds1** (*FoMpy Dataset*) – Structure of data containing the most important parameters of a semiconductor's IV curve. Needed for generating the plot of any FoM.
- **fds2** (*FoMpy Dataset*) – additional structure of data containing the most important parameters of a semiconductor's IV curve. Needed for generating the plot of the calibration and the DIBL.
- **plot_type** (*str*) – Keyword indicating the type of plot to generate. The list of available plots includes: 'iv', 'hist', 'qq', 'varplot', 'calib' and 'fomplot'.
- **fom** (*str*) – Keyword indicating the FoM to be plotted. The list of available methods includes: 'vth', 'ioff', 'ion', 'ss', 'ratio', 'power' and 'dibl'.
- **parameter** (*array_like*) – Array of extracted FoM values to be plotted.
- **method** (*str*) – Keyword indicating the desired method of extraction of the FoMs. The list of available methods includes: 'SD'(default), 'CC', 'TD' and 'LE'. If method is not defined the 'SD' is selected by default.
- **bins** (*int*) – It defines the number of equal-width bins in the given range.
- **cc_criteria** (*float*) – Current criteria used to extract vth with the CC criteria for the fomplot.
- **vg_ext** (*float*) – Gate voltage value used to calculate a FoM like IOFF or ION.
- **vg_start** (*float*) – Gate voltage defining the start of the interval in which the Subthreshold Swing is extracted.
- **vg_end** (*float*) – Gate voltage defining the end of the interval in which the Subthreshold Swing is extracted.
- **backend** (*str*) – String containing the name of the backend chosen to either plot or save the plots. The backends available are: 'Agg'(default), which only works whenever saving plots to files (non-GUI) and 'TkAgg' a GUI tool for visualizing the plots on a pop-up window. 'TkAgg' requires the package python3-tk installed in order to run.
- **save_plot** (*bool*) – If True the generated plot is save to the defined path.

fompy.wrappers.**savetotxt**(*path*, *fom*, *parameter*)
Wrapper function that saves to a text file the extracted FoMs.

Parameters

- **path** (*str*) – Defines the path where the extracted results are saved to a file.

- **fom** (*str*) – Keyword indicating the FoM to be plotted. The list of available methods includes: ‘vth’, ‘ioff’, ‘ion’, ‘ss’, ‘ratio’, ‘power’ and ‘dibl’.
- **parameter** (*array_like*) – Array of extracted FoM values to be plotted.

fompy.wrappers.**version**()

(TEST)Function that prints the current installed version of FoMpy

INDEX

Q

S

V