Sample rst2pdf doc

version

Your Name

■■■ 20, 2020

Contents

Welcome to phlab's documentation! 1

About phlab 1

Installation 1

Quick start 1

Input description 2

Basic (1D oscillator) 2

2D oscillator (2 modes problem) 2

Displaced and Distorted Oscillator 2

Modules 2

main 2

models 4

experiment 7

visual 7

Index 9

Python Module Index 11

Welcome to phlab's documentation!

About phlab

This package aims to create a convenient set of tools for fitting experimental data and subtracting electron-phonon coupling values from phonon contribution in resonant inelastic X-ray scattering cross-section.

It currently includes:

- Model for 1D harmonic oscillator interacting with a single electronic level.
- Model for 2D harmonic oscillator (two modes active).
- Model for 1D harmonic oscillator, distorted and displaced in the excited state (excited-state potential energy surface (PES), differs from the ground state PES).

Installation

It is a python based package and to install it you can simply run in terminal:

`$pip install phlab`

Download examples folder from this page and use it as a template for your projects. You may want to use Jupyter Notebooks, which comes with the examples. If you don't have Jupyter install it via `pip` as well:

`$pip install jupyterlab`

Quick start

First things first

```
import phlab
```

Now let's create our work space which is a wrapper for all the experiments and models:

```
workspace = phlab.rixs()
```

One of the main objects is a model. You can create any number of models and fit them to the exeperiment. Here were are starting with single harmonic oscillator model. Check `./model_name/` for input and output files.

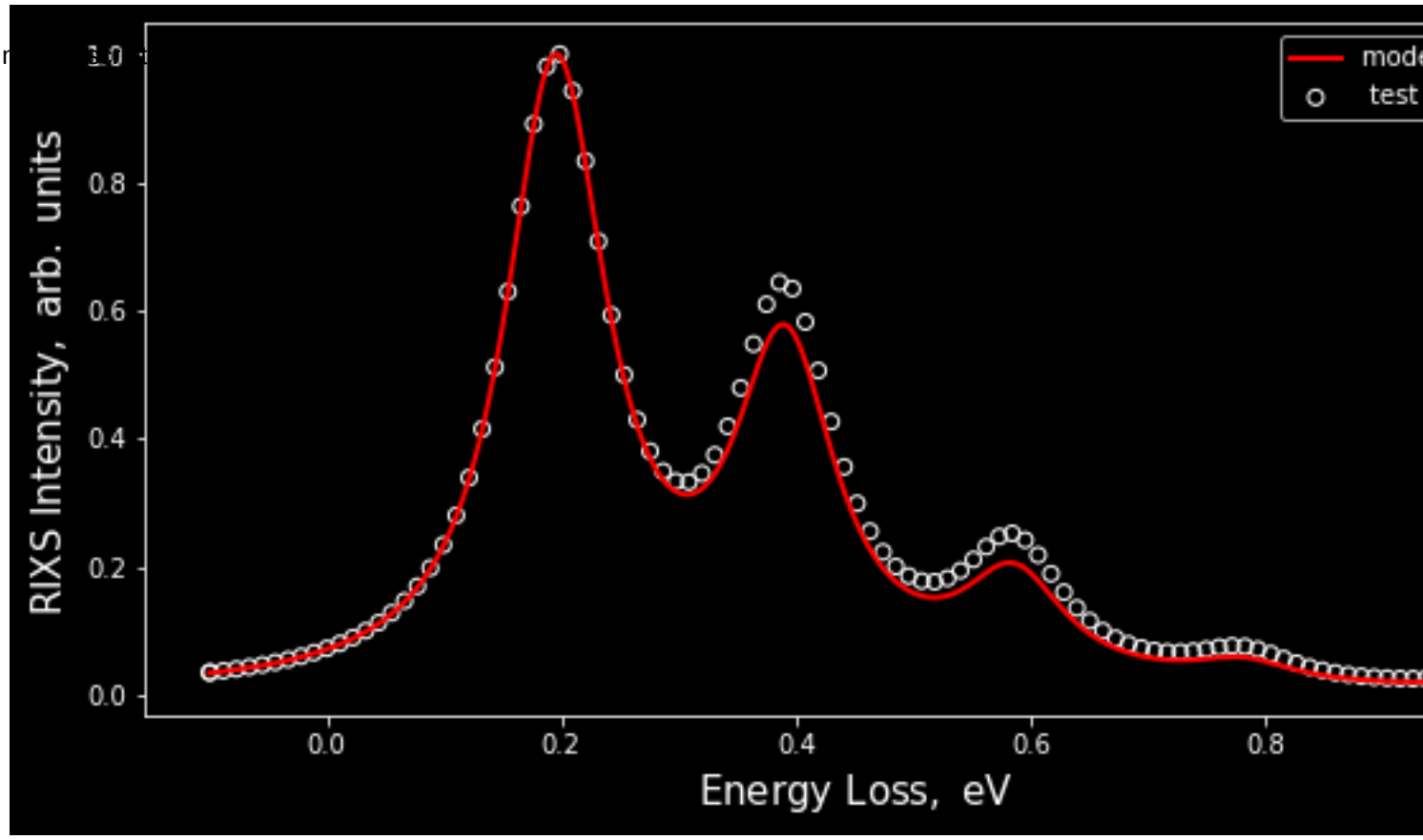
```
model = workspace.model_single_osc(name = '1d')
```

```
creating model : /Users/lusigeondzian/github/phlab/examples/01_example/1d
/Users/lusigeondzian/github/phlab/examples/01_example/1d/_input/
no input found
creating new input
warning : please check new input
number of models : 1
```

Input is normally reading from `./model_name/_inputs/input_model_{nm}.json` and is an atribute of the model

```
model.input
```

```
{'problem_type': 'rixs',
 'model': '1d',
 'method': 'fc',
 'vib_space': 1,
 'coupling': 0.1,
 'omega_ph': 0.195,
 'nf': 10.0,
 'nm': 100.0,
 'energy_ex': 10.0,
 'omega_in': 10.0,
```



Input description

Basic (1D oscillator)

2D oscillator (2 modes problem)

Displaced and Distorted Oscillator

Modules

main

class phlab.**rixs** (project_name='', out_dir='/_output/', inp_dir='/_input/')

Class rixs exists as a wrapper around both models and experiment objects.

Args:

problem_name: str

name of the project.

out_dir: str

name of the ouptu directory.

inp_dir: str

name of the ouptu directory.

Attributes:

nmodel: int

number of models created within this project.

nexp: int

number of exp created within this project.

abs_path: str

absolute path to the working directory

experiment (file='', col=[0, 1]**,** name=''**)**

Experiment.

Args:

file: str

path to the file with the exp data

col: list

[column x ; column y] defines which columns to read from the file

name: str

name of the experiment

Returns:

experiment.experiment(): object

calls experiment sub-package

model_dist_disp_osc (name=''**)**

Model describing distorted and displaced in the excited-state harmonic oscillator which interacts with a single electronic level.

Args:

name: str

name of the model

Note:

input and output files are located inside ‘./name/’ directory

Returns:

model.dist_disp_osc(): object

calls model sub-package

model_double_osc (name=''**)**

Model describing 2D harmonic oscillator which interacts with a single electronic level.

Args:

name: str

name of the model

Note:

input and output files are located inside ‘./name/’ directory

Returns:

model.double_osc(): object

calls model sub-package

model_single_osc (name=''**)**

Model describing a harmonic oscillator interacting with a single electronic level.

Args:

name: str

name of the model

Note:

input and output files are located inside ‘./name/’ directory

Returns:

model.single_osc(): object

calls model sub-package

visual (model_list=[]**,** exp=[]**)**

Creates visual object within the current project (works space).

Args:

model_list: list

list of models to plot

exp: object

experiment to plot

Returns:

visual.plot(): object

calls visual sub-package

models

class phlab.model.**dist_disp_osc** (inp_dir='./_input/', out_dir='./_output/', nmodel=0, name='')

Creates object for distorted and displaced harmonic oscillator model.

Args:

inp_dir: str

name of the input directory.

out_dir: str

name of the output directory.

nmodel: int

serial number of the model.

name: str

name of the model.

Attributes:

input_default: dict

dictionary with default input parameters.

input: dict

dictionary with current input parameters.

npoints: int

number of points in the spectrum.

spec_max: float

max limt of enrgy loss.

spec_min: float

min limt of enrgy loss.

param2fit: object

parameters to fit.

nruns: int

number of runs.

color: str

color of the line.

input_class: object

returns input_handler for this model.

x: float

energy loss in eV for the phonon contribution.

y: float

rixs intensities (arb. units) for the phonon contribution.

y_norm: float

normalized rixs intensities (arb. units) for the phonon contribution.

class phlab.model.**double_osc** (inp_dir='./_input/', out_dir='./_output/', nmodel=0, name='')

Creates object for 2D harmonic oscillator model.

Args:

inp_dir: str
name of the input directory.

out_dir: str
name of the output directory.

nmodel: int
id number of the model.

name: str
name of the model.

Attributes:

input_default: dict
dictionary with default input parameters.

input: dict
dictionary with current input parameters.

npoints: int
number of points in the spectrum.

spec_max: float
max limt of enrgy loss.

spec_min: float
min limt of enrgy loss.

param2fit: object
parameters to fit.

nruns: int
number of runs.

color: str
color of the line.

input_class: object
returns input_handler for this model.

x: float
energy loss in eV for the phonon contribution.

y: float
rixs intensities (arb. units) for the phonon contribution.

y_norm: float
normalized rixs intensities (arb. units) for the phonon contribution.

class phlab.model.**input_handler** (input_default={}, inp_dir='./_input/', nmodel=1, inp_name='input_model_{nm}.json', model_name='1d')

Contains methods to read and update input.

Args:

input_default: dict
dictionary with input parameters

inp_dir: str
name of the input directory

nmodel: int

id number of the model

model_name: str

name of the model

Attributes:

input: dict

dictionary with input parameters

class phlab.model.**parameters2fit**

Defines paramters to fit.

Attributes:

dict: dict

dictionary with parameters to fit

class phlab.model.**single_osc** (inp_dir='./_input/', out_dir='./_output/', nmodel=0, name='')

Creates object for 1D harmonic oscillator model.

Args:

inp_dir: str

name of the input directory.

out_dir: str

name of the output directory.

nmodel: int

id number of the model.

name: str

name of the model.

Attributes:

input_default: dict

dictionary with default input parameters.

input: dict

dictionary with current input parameters.

npoints: int

number of points in the spectrum.

spec_max: float

max limt of enrgy loss.

spec_min: float

min limt of enrgy loss.

param2fit: object

parameters to fit.

nruns: int

number of runs.

color: str

color of the line

input_class: object

returns input_handler for this model.

x: float

energy loss in eV for the phonon contribution.

y: float

rixs intensity (arb. units) for the phonon contribution.

y_norm: float

normalized rixs intensity (arb. units) for the phonon contribution.

experiment

class phlab.experiment.**experiment** (expfile='', columns=[0, 1], nexp=1, name='')

Experiment.

Args:

file: str

path to the file with the exp data.

col: list

[column x ; column y] defines which columns to read from the file.

name: str

name of the experiment.

nexp: int

id number of the given experiment in the given project.

Attributes:

x: float

energy loss readings from exp file.

y: float

rixs intensity readings from exp file.

max: float

max value of y.

y_norm: float

normalized y.

name: str

name of the experiment.

xmin: float

min value of x.

xmax: float

max value of x.

visual

class phlab.visual.**plot** (model_list=[], exp=[])

Visualization.

Args:

model_list: list

list of models.

exp: object

experiment.

Attributes:

if_exp: boolen

returns True if experiment (object) is specified.

Index

D

dist_disp_osc (class in phlab.model)

double_osc (class in phlab.model)

E

experiment (class in phlab.experiment)

experiment() (phlab.rixs method)

I

input_handler (class in phlab.model)

M

model_dist_disp_osc() (phlab.rixs method)

model_double_osc() (phlab.rixs method)

model_single_osc() (phlab.rixs method)

module

- phlab
- phlab.experiment
- phlab.model
- phlab.visual

P

parameters2fit (class in phlab.model)

phlab

- module

phlab.experiment

- module

phlab.model

- module

phlab.visual

- module

plot (class in phlab.visual)

R

rixs (class in phlab)

S

single_osc (class in phlab.model)

V

visual() (phlab.rixs method)

Python Module Index

p

phlab

phlab.experiment

phlab.model

phlab.visual