Circular weighing code

# Versions

Last updated 20/07/2017, maorbenshahar@gmail.com

"wx": Version 2.8.10

"visa": 1.8

wx is needed in order to run the GUI. Without it, it is difficult but both the circular weghing least squares and the final mass calculations have the ability to read a csv file, and operate on that.

visa is needed to communicate with the instruments, for the automatic gathering of data. PyVisa 1.8 is not compatible with 1.5 or earlier, but hopefully it maintains backwards compatibility with 1.8 for a while. To run visa, the national instruments visa needs to be downloaded (NIVISA541full\_doaloader.exe) as well as the python wrapper "pyvisa", can be downloaded using "pip install pyvisa" in the command prompt.

# The code

There are three windows used, main, circ(ular) and collect. Main is the main frame, circ is the window for the circular weighing algorithm and analysis, and collect is for data collection.

A “gui” prefix such as “gui\_main.py” is for files generated by wxBuilder, they should not be edited.

A “main” prefix, such as “main\_circ.py” is the editable file, where the actual content of the functions is written. Each “main\_” file is a subclass of its corresponding “gui\_” file, overriding the blank event driven buttons actions.

“calc\_” prefixes are for calculations, they are kept separate form the rest of the program. These are imported by “main\_” files and made use of for analysis. The calcualtions make use of numpy for accurate calculations and for speed.

|  |  |
| --- | --- |
| Weighing.py | Weighing sequence to collect data from balance, things like ‘raw\_input’ will not be compatible with python 3. |
| Visa2.py | Simulated visa for testing |
| Stuff.py | Extra functions, not edited. |
| Pywxgrideditmixin.py | Mixing for the grid, allows user to copy and paste info. |
| Main\_collect.py | The control window for communicating with balances and collecting info |
| Main\_circ.py | Window for circular weighing tables, allows to analyse a set of circular weighing results. Can open a main\_collect window to collect results. |
| MAIN.py | The main window and tables, can be used to load a main\_circ window to update a weighing difference/collect data. |
| Calc\_mass.py | The larger and more complex final mass calculation. Does a least squares on the differences to find the actual masses given sufficiently many constraints. Works for python 3 and 2. |
| Calc\_circ.py | Least squares for a single circular weighing. Works in python 3 and 2. |

# Further development

The code needs to be able to communicate with all the balances, perhaps what is currently written in weighing.py is not easily generalised. I think there is a new balance that will be introduced soon?

The calc\_ files for calculations will run with both python 2 and 3, so the hardest part of modernising to python 3 is the wx part. The gui is not necessary for operation though, if you don’t mind typing into a command prompt and writing the tables yourself (:

The results could return a simple ‘result’ object which has attributes ‘mass’, ‘differences’ etc. At the moment it is an array, [mass1,mass2,difference,uncertainty], which is pretty messy for printing to tables and saving. It is messy to continuously use the .replace(x,y) functionas well as .split(s).

Weights and their readings could also be some object, similar (or using?) GTC. It is sufficient that it has a label and a value (if known), then results objects could carry mass objects in them.

Unknown sets: when a single set is sent it is not recognised as single.

Capture empty data errors