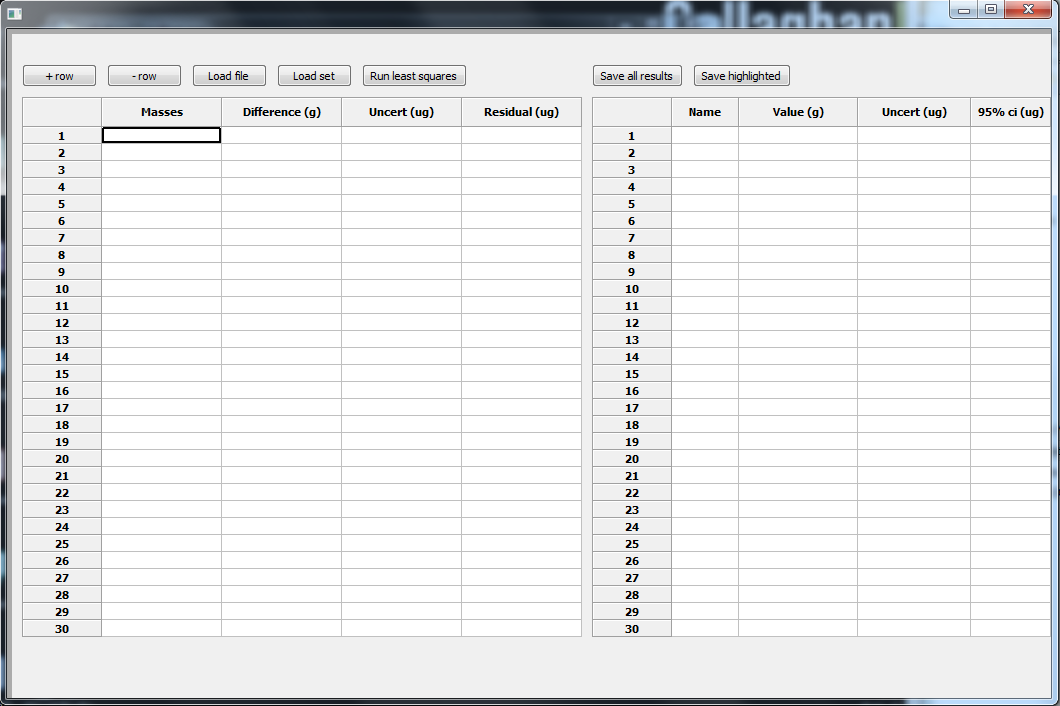
Circular weighing – User guide.

The software was written to replace the old software on the MSDOS machine. It has several components; data gathering (incomplete), circular weighing analysis to determine differences between up to five masses, and the final least squares analysis to calculate the values of individual masses. To run the software, double click “MAIN.py”.

# Getting started

The starting state is two empty frames, one for input data and one for output results.

A calibration routine can be entered manually, by typing into the left-hand table or it can be loaded from a file (Load file). The +rows and -rows will change the grid size, in case additional data needs to be entered. Loading from a file will automatically match the grid size to the file size. If some rows are highlighted (clicking their number on the left) the -rows will delete the highlighted rows only. Pressing “Load set” will require selecting a set file, and choosing a suffix for all the mass values there. This is appended to the bottom of the table.

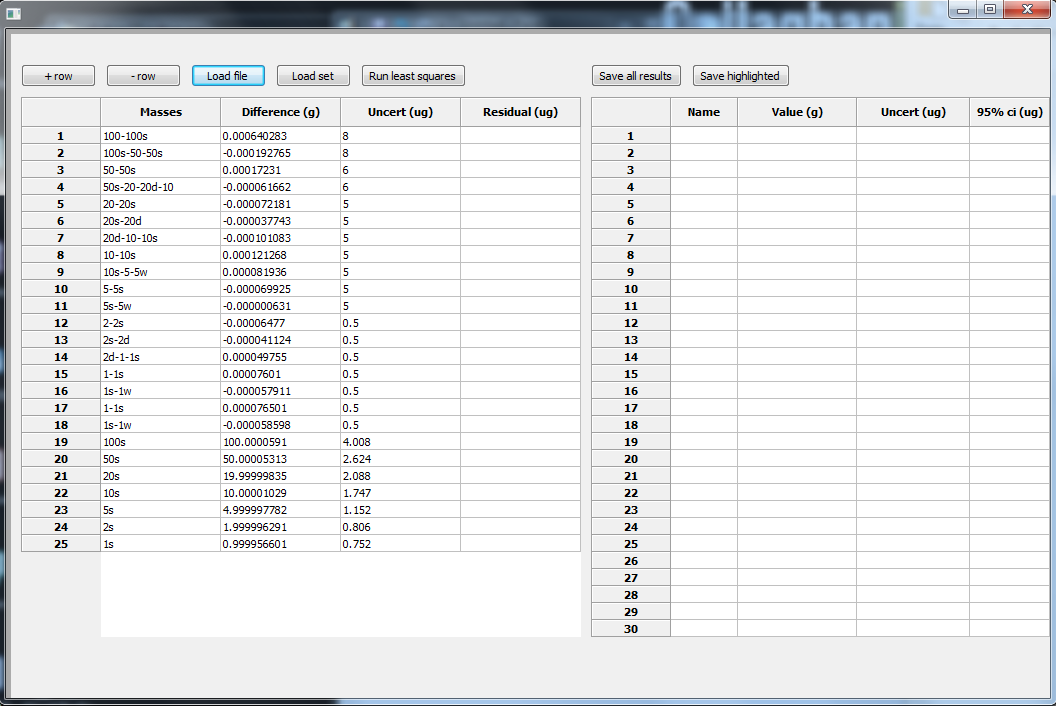
**Masses:** this has text inputs of the form “100-50s-50” where each number is a label for a mass (normally its nominal mass value).

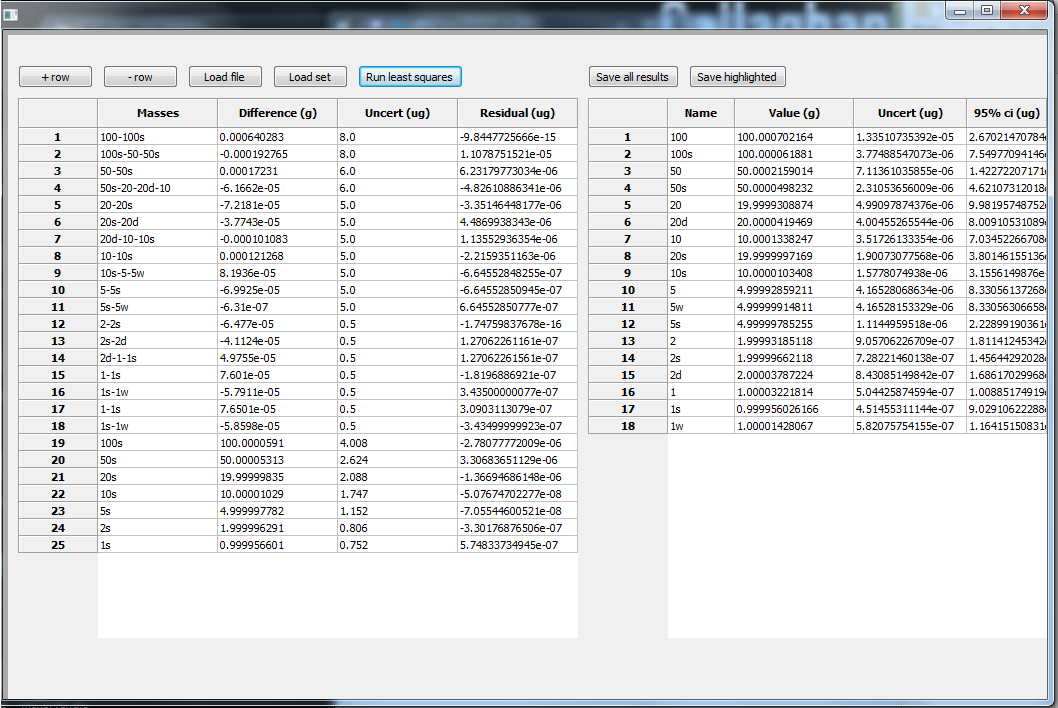
**Difference:** This is the calculated difference in grams. It can be entered, calculated (using the circular weighing algorithm) or loaded in the file. This entry should be a number, such as “0.00034”.

**Uncert:** This is the uncertainty in the balance used for that particular measurement.

**Residual:** Computed and entered by the program, when ‘run least squares’ is pressed.

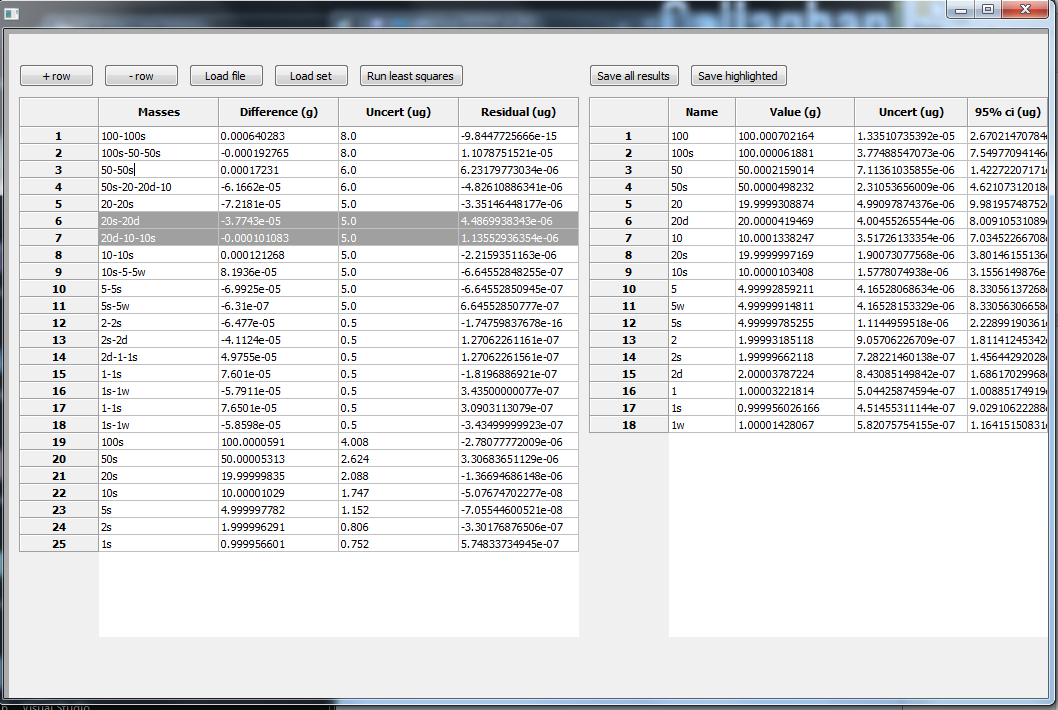
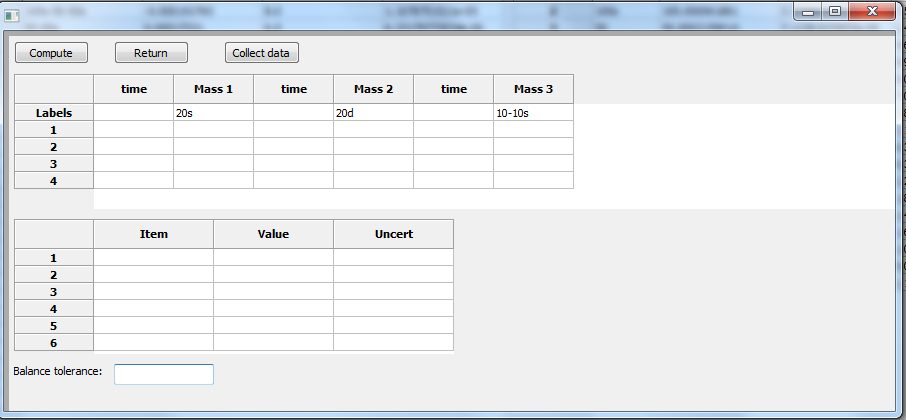
# Computing actual mass values

When data (and differences) are entered, the table should look like this:

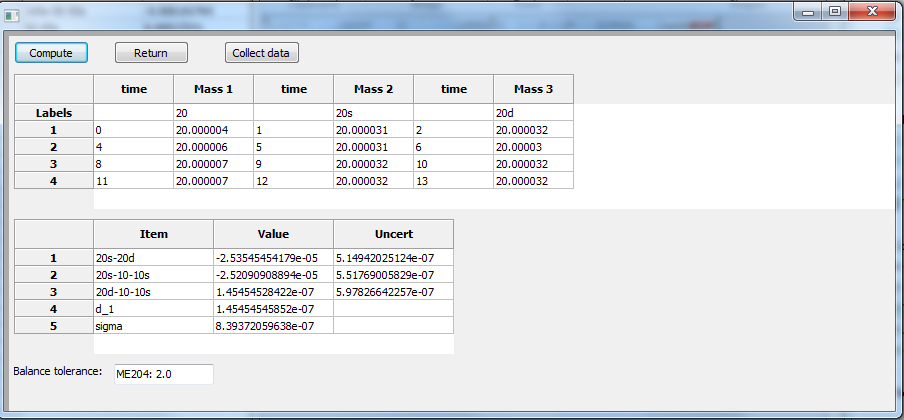
The constraints at the bottom are entered as a single mass, and so the ‘difference’ column contains nothing but its value. Running the least squares analysis will compute residuals and enter results into the results table, as well as resize it to fit all the data entries. Not all data entries have to exist, the program identifies empty rows (or rows with insufficient info, for example a missing difference value) and ignores them.

# The circular weighing

If differences are not entered by hand, they can be retrieved from data of a circular weighing. To add a circular weighing, highlight the rows that contain the names of masses to be examined by clicking their row label names. Highlighted rows appear in grey or blue. RIGHT CLICK the mouse anywhere in the table to open the circular weighing window, for the selected rows.

The smaller popup window contains two tables, the top table for the input data, and the bottom table for results. Data can be pasted, typed, or gathered automatically. The program sets the number of readings to 7-number of masses, or to 3 since this seems to be what was done in the past.

Once data is entered, the “compute” button will run the least squares analysis on it and put the values into the lower grid of results, and a balance uncertainty is printed to the textbox below. Later these uncertainties can be edited in the main table.

All possible differences are computed and printed, as well as their uncertainty. Not all of them are useful, since not that in this case the third is dependent on the previous two.

Below the differences, the coefficients of the time-fit are printed. In this case, only one is there since the drift was very linear. The program attempts to fit:

It does three fits, once with only a linear, then a parabola, then a cubic and selects the one with the lowest residual (sigma) printed at the bottom. Higher order approximations can be done too, but near around order 6, the computer fails to calculate reliably.

Pressing the return button will return the mass differences to the highlighted rows in the table. Several of these circular weighing windows can be open at once, each knows where its differences came from. The return button will also save a csv file of the data and results, with a date stamp and the names of the masses involved. This is in case the analysis needs to be edited or checked later. The file can be used to copy and paste the info back to the table.

# Manual data entry:

Data can be pasted form excel straight into the table. If no times are specified the program will assume time differences are constant and uniform.

If readings are done manually, once a mass value is entered into a cell clicking the right mouse button on that mass cell will enter a time stamp in the corresponding time column. The order for taking readings is left to right, going down row by row.

# Automatic data gathering:

The “collect data” button allows for data to be collected automatically by the computer. Pressing the button opens a third control window, from which the balances are controlled.

# Data collection window

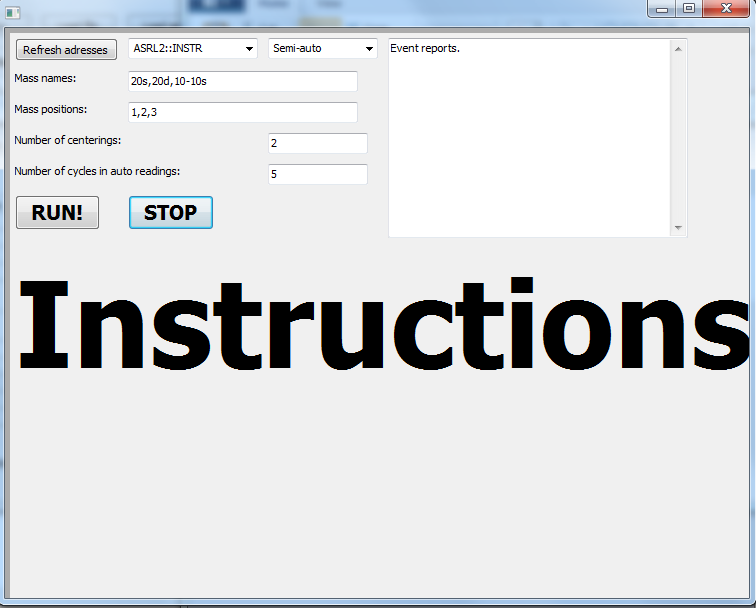
“Refresh addresses” reads the available addresses (GPIB and RS232) using PyVisa and puts them in a selection box to the right. To the right of that is another selection box, for automatic or half-automatic measuring sequences.

Mass names is not editable, but shows the names and order of the masses to be examined. Below it, mass positions should contain a list corresponding to the positions on the comparator of each mass. Order is important. Entries here should be like “1,2,3”, numbers separated by commas. The three masses here are the 20s, 20d and a combined mass, 10 and 10s together.

Number of centrings is the number of times the program will centre each mass.

RUN sets the thread running, and STOP stops it.

The big text box containing “Event reports” will contain a log of all the processes the program went through, such as reading and writing to instruments, how long it is waiting for and so on.

Instructions at the bottom will be presented if the user is going to be using the semi-automatic option. These will be telling the user which mass to put on.

Once measurements are taken, the window returns them to the parent window (tables) where data can be analysed. If it is within the accepted range it can be returned to the main table.

# Using the software without the GUI:

The software GUI relies on an external library. It might be useful to be able to run the software without the library, especially considering its version and that it is relying on c++ packages.

A calibration is split into 3 components, collecting data, calculating individual differences (using the cicular weighing algorithm) and calculating the final least squares to obtain actual mass values.

## Collecting data

The thread can still be used to collect data automatically, even without the controlling GUI. To run, open the file and edit the bottom under “if \_\_name\_\_ == “\_\_main\_\_”. This statement checks if the program is being run as a main program (in this case it will be true) or if it has been imported as a module. The parameters match those specified in the GUI:

|  |  |
| --- | --- |
| port | The visa port to connect to. |
| masses | Ordered names of the masses. Note that if masses are used together, like 20s+20, they shoult be one element of the array as a single string ‘20+20s’. |
| mass\_positions | Ordered positions of the masses, corresponds to the above. |
| reads\_per\_mass | Number of readings to do for each mass in each cycle. Normally, this is 7-number of masses (for 4 or less masses), or 3 (for 5 or more masses). |
| run\_options | ‘AUTO’ or ‘SEMI’ |
| centerings | Number of centerings to do (for automatic msmnt) |
| Num\_cycles | Number of measurement cycles, each cycle has read\_per\_mass readings on each mass. |
| parent | A parent object that will have all sorts of functions to receive info from the thread. If no parent is specified, the data gets saved into a csv with a format suitable for reading by calc\_circ.py. |

The default parameter “simulated” can be set to True so that the program will use visa2. It can be completely omitted as well and will default to false.

## Circular calculation

Once data is created as above, or if it is manually recorded it needs to be saved into a csv with the following format:

|  |  |  |  |
| --- | --- | --- | --- |
| Circular algorithm data and results | | | |
| Data: |  |  |  |
|  | 100A |  | 100 |
| 0 | 108.9297969 | 0.033 | 104.7948921 |
| 0.066000001 | 105.7345396 | 0.098999997 | 106.7520692 |
| 0.131999997 | 109.9293455 | 0.164999998 | 108.1440693 |
| 0.197999998 | 111.4587108 | 0.230999998 | 108.395304 |
| 0.263999999 | 109.3106177 | 0.296999999 | 111.4366809 |
|  |  |  |  |

There are columns for time, columns for the masses and 3 roows before the data begins (two for titles and any additional info, and one for the mass labels). In the bottom of the calc\_circ.py file, the name of the data file to be analysed needs to be specified. The file needs to be saved to the same directory as the calculation module, and then it can run. Results can be printed in whatever way is convenient and they will need to be manually entered into the larger table for use in the final least squares fit.

## Final least squares fit

After collecting individual differences, the final table needs to be constructed like so:

|  |  |  |
| --- | --- | --- |
| Verifying old data | | |
| Times are in grams | |  |
| 100-100s | 0.000640283 | 8 |
| 100s-50-50s | -0.000192765 | 8 |
| 50-50s | 0.00017231 | 6 |
| 50s-20-20d-10 | -0.000061662 | 6 |
| 20-20s | -0.000072181 | 5 |
| 20s-20d | -0.000037743 | 5 |
| 20d-10-10s | -0.000101083 | 5 |
| 10-10s | 0.000121268 | 5 |
| 10s-5-5w | 0.000081936 | 5 |
| 5-5s | -0.000069925 | 5 |
| 5s-5w | -0.000000631 | 5 |
| 2-2s | -0.00006477 | 0.5 |
| 2s-2d | -0.000041124 | 0.5 |
| 2d-1-1s | 0.000049755 | 0.5 |
| 1-1s | 0.00007601 | 0.5 |
| 1s-1w | -0.000057911 | 0.5 |
| 1-1s | 0.000076501 | 0.5 |
| 1s-1w | -0.000058598 | 0.5 |
| 100s | 100.0000591 | 4.008 |
| 50s | 50.00005313 | 2.624 |
| 20s | 19.99999835 | 2.088 |
| 10s | 10.00001029 | 1.747 |
| 5s | 4.999997782 | 1.152 |
| 2s | 1.999996291 | 0.806 |
| 1s | 0.999956601 | 0.752 |

Two rows for titles, and then information in columns. The columns are, text differences of the mass names (or single mass name if it is a constraint). The difference of the two masses (or mass value if it is a constraint) and the final column is the balance uncertainty (or uncertainty in the mass value). This file can be saved to the same directory as ‘calc\_mass.py’, and again at the bottom of the code the file name needs to be specified. Running this will print all the mass values, and save them to a csv too.