

qtFit Documentation

Hassan Saadaoui

saadaoui@triumf.ca

April 23, 2015

Abstract

`qtFit` is a program used to fit ASCII data. The graphical user interface (GUI) of `qtFit` is designed using QT software. The regression is done using CERN's MINUIT routine and plotting is performed with the `QCustomPlot` library. This document is a manual of the `qtFit` program, and includes instructions about installation, features description, and how-to-use examples.

Contents

1	Introduction	2
2	Requirements and structure	3
3	Installation	3
3.1	QT	3
3.2	MINUIT	3
3.3	qtFit	4
4	Description of the GUI	5
4.1	Menu bar	5
4.2	Regression	5
4.3	Results	8
4.4	Graphics	9
4.5	Database	10
5	Fitting functions	11
6	How-To-Use examples	12
6.1	Example 1	12
6.2	Example 2	12

1 Introduction

qtFit is used to search, view, and analyze ASCII data. It can be extended by the user to read any type of data. This program is developed by Hassan Saadaoui and is maintained as needed. It is open source and released under the General Public License (GPL). No warranty or guarantee of the results is implied. Please acknowledge the author if you are using this program. For any questions, please email at saadaoui@triumf.ca.

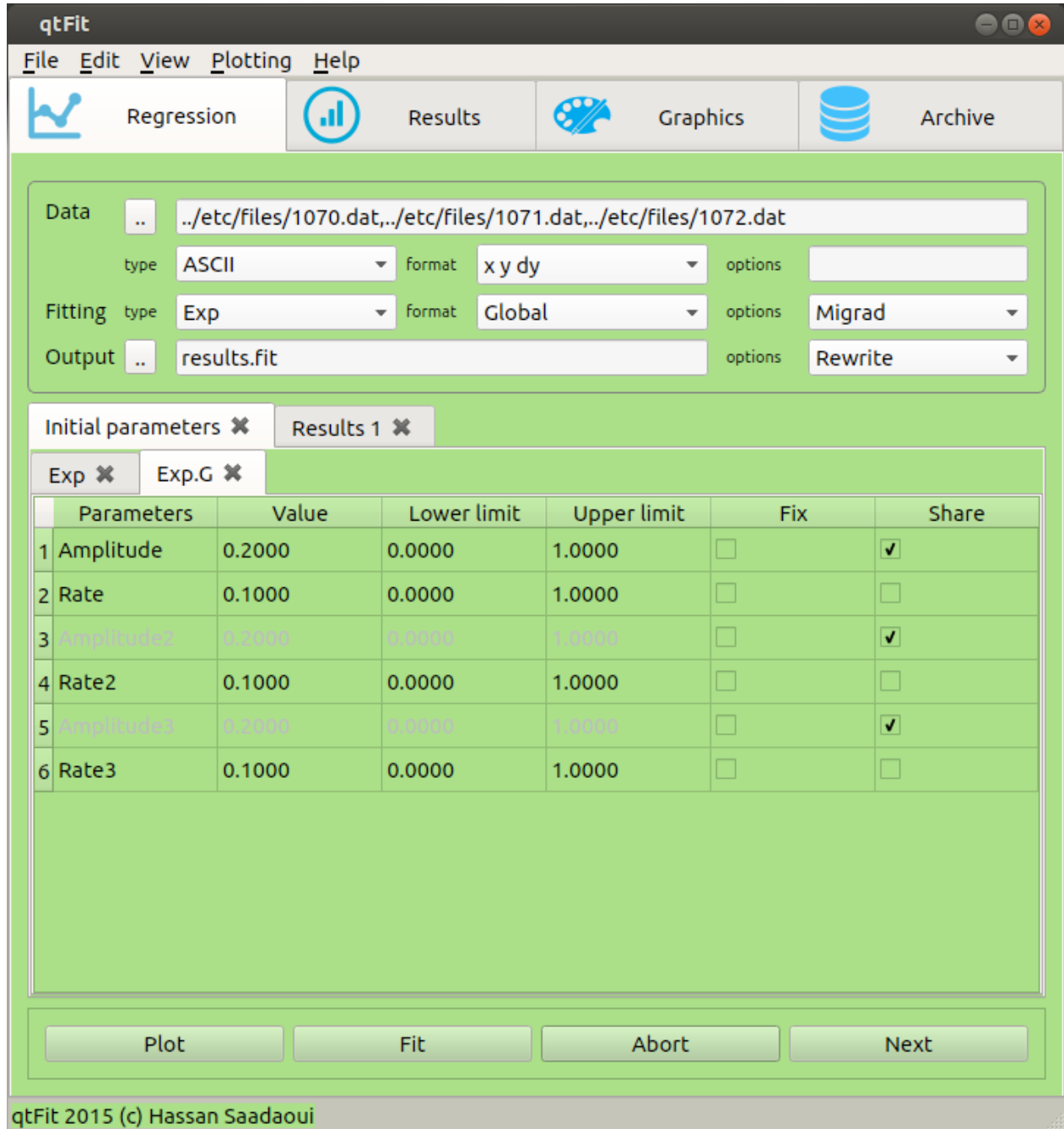


Figure 1: Regression page of the qtFit graphical user interface.

2 Requirements and structure

The program is written in C++ and QT. The latter is used for general programming as well as the graphical user interface. Version 4.8.x or 5.x is needed. The regression is done using MINUIT minimization routine developed at CERN. This package is originally in Fortran and later converted to C++. It is very powerful and well tested. QCustomPlot library (included within the package) is used for data visualization. In addition to the above requirements, and depending on your system, you may also need other packages such as gcc compiler, and automake. The qtFit package contains 3 sub-folders and 5 files.

- `src/`: contains the source code, and includes the shared `main.cpp`, and `mainwindow.(cpp,ui,h)` and a sub-folder for each page.
- `fct/` contains the fitting functions and the script "compile" to execute the codes and create the libraries.
- `etc/` for documentation, images, data, scripts and templates, and the `resources.qrc` file needed by QT.
- `qtfit.pro` used to generate the makefile.
- `AUTHOR` for authorship attributions.
- `COPYING` supplies the GPL agreement.
- `README.md` for installation instructions.

3 Installation

3.1 QT

You need to install QT 4.X.X or QT 5.x binary and source packages. Download the QT on-line installer from <http://www.qt.io/download-open-source/>.

3.2 MINUIT

To compile, do the following steps.

1. Download latest Minuit2 located at
<http://seal.web.cern.ch/seal/snapshot/work-packages/mathlibs/minuit/release/download.html>
2. Unpack and cd \$ `tar -xvf minuit.tar.gz -C ~/`
\$ `cd ~/minuit`
3. To install follow the instructions at
<http://seal.web.cern.ch/seal/snapshot/work-packages/mathlibs/minuit/gettingStarted/autoconf.html>
4. Make SURE that the tests in the tutorial are running as described in the link
<http://seal.web.cern.ch/seal/snapshot/work-packages/mathlibs/minuit/gettingStarted/testOutput.html>
5. Copy (as superuser) the miniut libraries from
`minuit/src/.lib/liblbg_Minuit.*` to `/usr/lib/`
\$ `sudo cp minuit/src/.lib/liblbg_Minuit.* /usr/lib/`
6. Update ldconfig
\$ `sudo ldconfig`

Extra notes: It is somewhat a challenge to compile Minuit2. These extra notes maybe useful.

- Depending on your system, you may need to modify few codes namely `src/MnUserTransformation.cpp` to add `#include <cstdio>` or `#include <cstdio.h>` just below `#include <algorithm>` and re-compile.
- Locate where libraries and header files are, hopefully in `/usr/local/include/Minuit2`, and `/usr/local/lib/`
- Add the path `/usr/local/lib/` to `/etc/ld.so.conf` as described here <http://stackoverflow.com/questions/1099981/why-cant-python-find-shared-objects-that-are-in-directories-in-sys-path>

```
$ export LD_LIBRARY_PATH=/usr/local/lib
or
$ export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH
```

- Run `ldconfig`
\$ `sudo ldconfig`

3.3 qtFit

1. Download the latest qtFit from local computers, sourceforge, or github.
\$ `wget https://sourceforge.net/projects/qtfit/files/qtfit.tar.gz/download`
2. Unpack it
\$ `tar -xvf qtfit.tar.gz -C ~/`
3. cd to the downloaded package
\$ `cd ~/qtfit`
4. If QT binaries are not in your path, set the env (locate where qmake is)
\$ `PATH=/usr/Qt/5.4/gcc_64/bin:$PATH` (change `"/usr/Qt/5.4/gcc_64/bin"` as per your system)
\$ `export PATH`
5. Run qmake
\$ `qmake`
6. Run make and make install (as root)
\$ `make`
\$ `sudo make install`
7. cd to directory `fct/` and compile all the libraries using the script `compile`
\$ `cd fct`
\$ `sudo ./compile`
8. To test the gui, invoke
\$ `qtfit`
or
\$ `./qtfit` if not installed as root.

4 Description of the GUI

The GUI has a menu bar at the very top and a tab widget below it. This tab widget contains 4 tabs (pages): **Regression**, **Results**, **Plotting**, and **Archive**. Each of these contains widgets for user input and push buttons for issuing signals. The menu bar and the pages functionalities will be described next.

4.1 Menu bar



Figure 2: Menubar of the GUI.

File has 2 options; (i) invoke a new window, and (ii) quit/close the window.

Edit is empty for now, not finished...zzz.

View to change the widget type of the GUI (default is **fuse+**), and its color (default is "Green-white").

Plotting not finished...zzzz.

Help This contains the "About" dialog for authorship and version of the current GUI, "Tips" dialog which does nothing but remind the user that by hovering the mouse index onto labels one can get the tool-tips for each widget. "Tutorial" invokes an HTML page with these instructions.

4.2 Regression

Data input

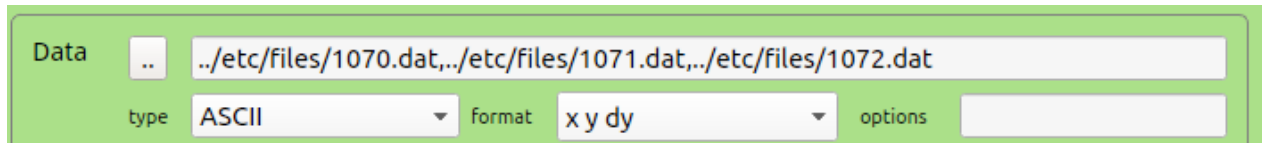


Figure 3: Data input of the regression page.

The user can choose to fit either ASCII text data or something else, for now it is only ASCII. The user can locate the data using the tool button next to Data for ASCII files. The data file must reside in the working directory, otherwise its full name with path should be given. Filenames of the data to fit should be either given in the lineEdit (direct method); or using a file of **.list** or **.inf** suffix, and the user specifies the name of this file (eg: example.list or example.inf) in the lineEdit.

- Direct input, the user can write in the lineEdit "Data" something like "file1.txt,file2.txt,file3.txt"
- Indirect input, the user can write in the lineEdit "Data" something like **example.list**, and this file has nothing but a single ascii line like "file1.txt,file2.txt,file3.txt"
- Indirect input, the user can write in the lineEdit "Data" something like **example.inf**, and this file contains the columns
filenames year variable
file1.txt 2015 100
file2.txt 2014 200
file3.txt 2015 300

The inner format of the ASCII file must be set in the field **format**. The file must be in column format separated by space and no other characters than numbers. The limits of xmin and xmax values can be set in the options (settings) field. These must be numbers separated by commas.

Fitting selection



Figure 4: Fitting functions input.

The user can select the function to use, the mode of fitting (single or global) and type of errors. The functions are defined in the folder **fct/** and the user can add new ones by invoking the selection "Create New" in the functions comboBox. The user must follow the instructions in the pop-up window and then select "Update" from the comboBox. This will add the newly defined function to the list.

For the global method the user can choose to show all parameters for each run or not. These settings can be changed by double clicking on the initial parameters tab-widget.

The errors are defined by MINUIT routine, and are symmetric (Migrad) or asymmetric (Minos) errors. The latter are heavy to compute and the program may become unresponsive for sometime while the computation is ongoing. For further details read <http://seal.web.cern.ch/seal/documents/minuit/mnerror.pdf>.

Results output

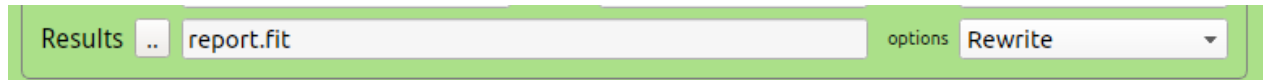


Figure 5: Fitting results output.

The fitting results are written to this file. The user must specify a name, or browse for an old file. The results can be either appended (using **Append**) to the old file keeping its content (useful for doing run by run fitting), or the old file is overwritten using **Rewrite**.

Parameters input

The initial parameters are read from the function library. The table contains 5 columns for the single method, and 6 columns for the global method. These columns are; (1) parameter name, (2) initial value of the parameter, (3) lower limit, (4) upper limit, (5) fix the parameter checkBox, and (6) share the parameter checkBox.

These parameters can be changed, and saved in a template for future use by right-clicking on the specific table and then choose "save as a template". This creates a text file template with a prefix ".tab". The user can change this text file as required, and the template can be loaded later for a similar function.

Parameters output

This prints out the output of the fit. The number of significant figures can be set by double-clicking on the results tab. One can also change the number of errors to show, and the way the filename is displayed.

Initial parameters ✕

Exp ✕

	Parameters	Value	Lower limit	Upper limit	Fix
1	Amplitude	0.2000	0.0000	1.0000	<input type="checkbox"/>
2	Rate	0.1000	0.0000	1.0000	<input type="checkbox"/>

Initial parameters ✕
Results 1 ✕

Exp ✕
Exp.G ✕

	Parameters	Value	Lower limit	Upper limit	Fix	Share
1	Amplitude	0.0848	0.0000	1.0000	<input type="checkbox"/>	<input checked="" type="checkbox"/>
2	Rate	0.0651	0.0000	1.0000	<input type="checkbox"/>	<input type="checkbox"/>
3	Amplitude2	0.0848	0.0000	1.0000	<input type="checkbox"/>	<input checked="" type="checkbox"/>
4	Rate2	0.0651	0.0000	1.0000	<input type="checkbox"/>	<input type="checkbox"/>
5	Amplitude3	0.0848	0.0000	1.0000	<input type="checkbox"/>	<input checked="" type="checkbox"/>
6	Rate3	0.0651	0.0000	1.0000	<input type="checkbox"/>	<input type="checkbox"/>
7	Amplitude4	0.0848	0.0000	1.0000	<input type="checkbox"/>	<input checked="" type="checkbox"/>
8	Rate4	0.0651	0.0000	1.0000	<input type="checkbox"/>	<input type="checkbox"/>

Figure 6: Input parameters for (a) single and (b) global fits. (a) The fit starts from this table for each file, or from the results of the last file in the sequence enabled by double clicking on the "initial parameters". (b) If a parameter is shared between files, only the parameter of the first run is active and the same parameter for other files becomes inactive.

Initial parameters ✕ Results 1 ✕ Results 2 ✕				
	Files	Amplitude	Rate	Chisq
1	45122	0.1058±0.0185	0.1083±0.0392	1.0700
2	45123	0.1058±0.0185	0.1037±0.0368	1.0700
3	45124	0.1058±0.0185	0.1083±0.0364	1.0700
4	45125	0.1058±0.0185	0.0942±0.0373	1.0700

Figure 7: Results of a global fit where the shared parameter is Amplitude.

Input and output options

The window shown in figure 8 can be invoked by double clicking on the header of the parameters table. In the pop-up window, the user can choose to show the parameters for all files in the case of a global fit, or use a common template for all files. The user can also choose to start the fit with the results of best fitting parameters of the previous file in the sequence (only valid for single fit). For the output table, the user can choose the format of the filename label, parameters precision, and number of errors to display.

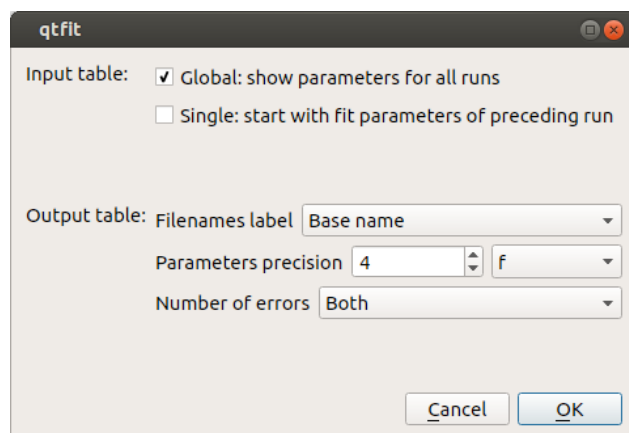


Figure 8: Input and output options.

4.3 Results

This page reads the files of fitting parameters created by the analysis page. It displayed a table with two columns, the left column represents the x-axis and the right column the y-axis. Each column contains all fields found in the specified file (as created during the fitting).

The user can check any of the fields, and a matrix of plots of y versus x will be displayed. The user can clear all choices using "Clear", and kill/delete the active table using "Purge".

The plots will be displayed on the Graphics page. Horizontal or vertical error bars are displayed if specified in the chosen parameter.

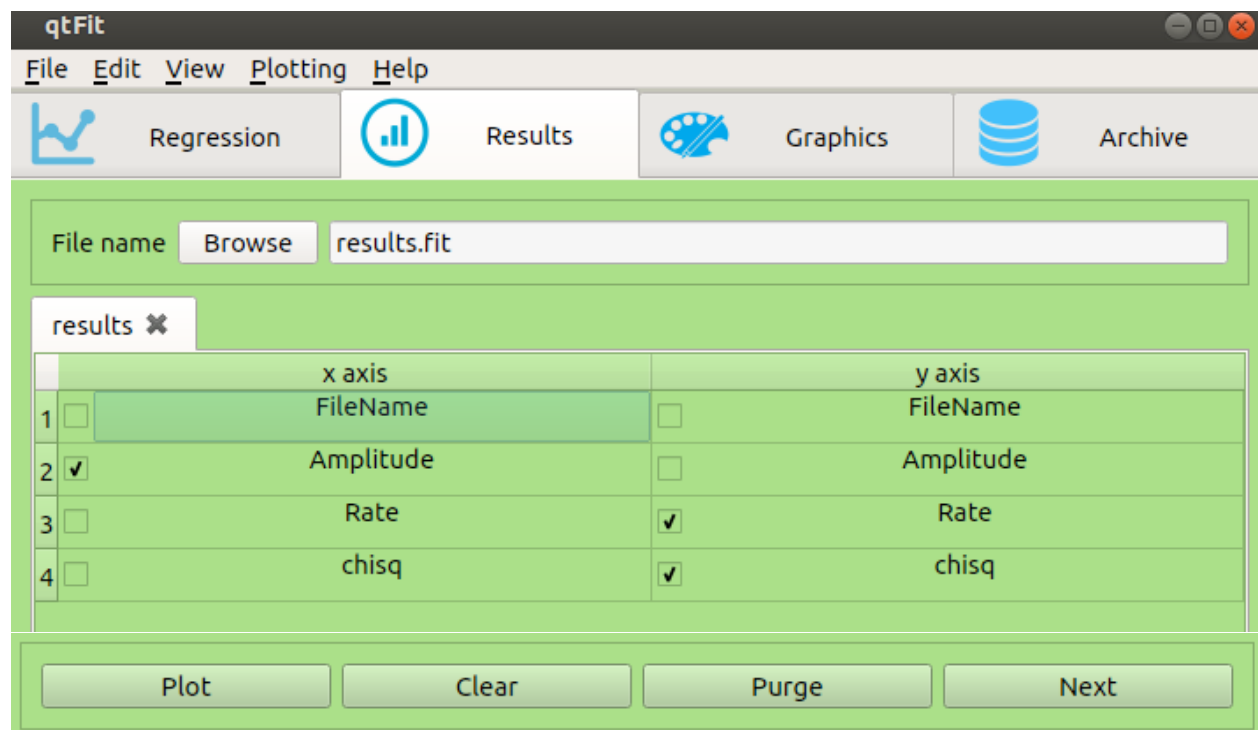


Figure 9: Results page snapshots.

4.4 Graphics

The plots of the regression page and results pages are displayed here. The user can **export** the graphs into png or pdf, or **Purge/Clear** (delete) the active window(s), or go the archive database with **Next**.

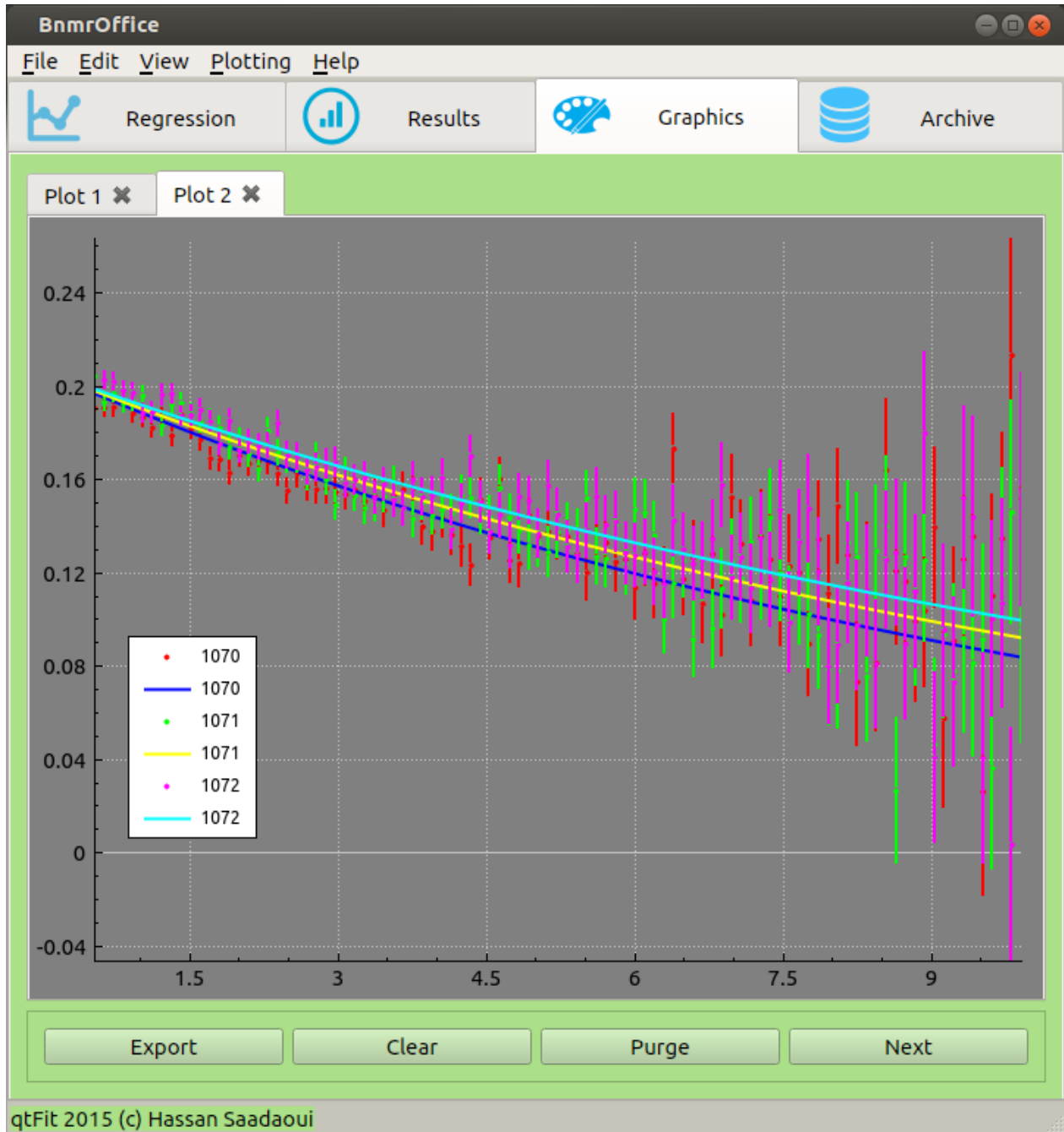


Figure 10: Graphics page.

4.5 Database

This page offers a user-friendly interface for databases, and uses SQLite language http://www.tutorialspoint.com/sqlite/sqlite_overview.htm. At the start, the user must select a database by clicking on the toolButton next to "Database", or create a new one from the "Query" lineEdit using SQL commands and hitting "Execute". It is advised to use an SQL manager (like the friendly browser extension SQLite manager) to create databases and tables. Then, one can use this interface to add/delete rows and edit cells, interact with the content of the database. But an experienced user can do everything from this page as well by executing the "Query" commands. To get familiar with the interface, "physics.sqlite" is supplied. Each contains several tables. The user can load any of these tables from the comboBox, and a model of the table will be displayed.

The user can execute any query to study the loaded table. Example;

`SELECT * FROM table_of_constants where Unit="kg"` will select all fields in the `table_of_constants` where the unit is in kg. The user must be familiar with SQL to execute from the Query field. Any table can be changed by adding or deleting rows. Also each cell can be edited, or displayed by clicking on "Open". This can be used to display a cell with a lot of text or view the cell as image if the full path of the image was given in that cell.

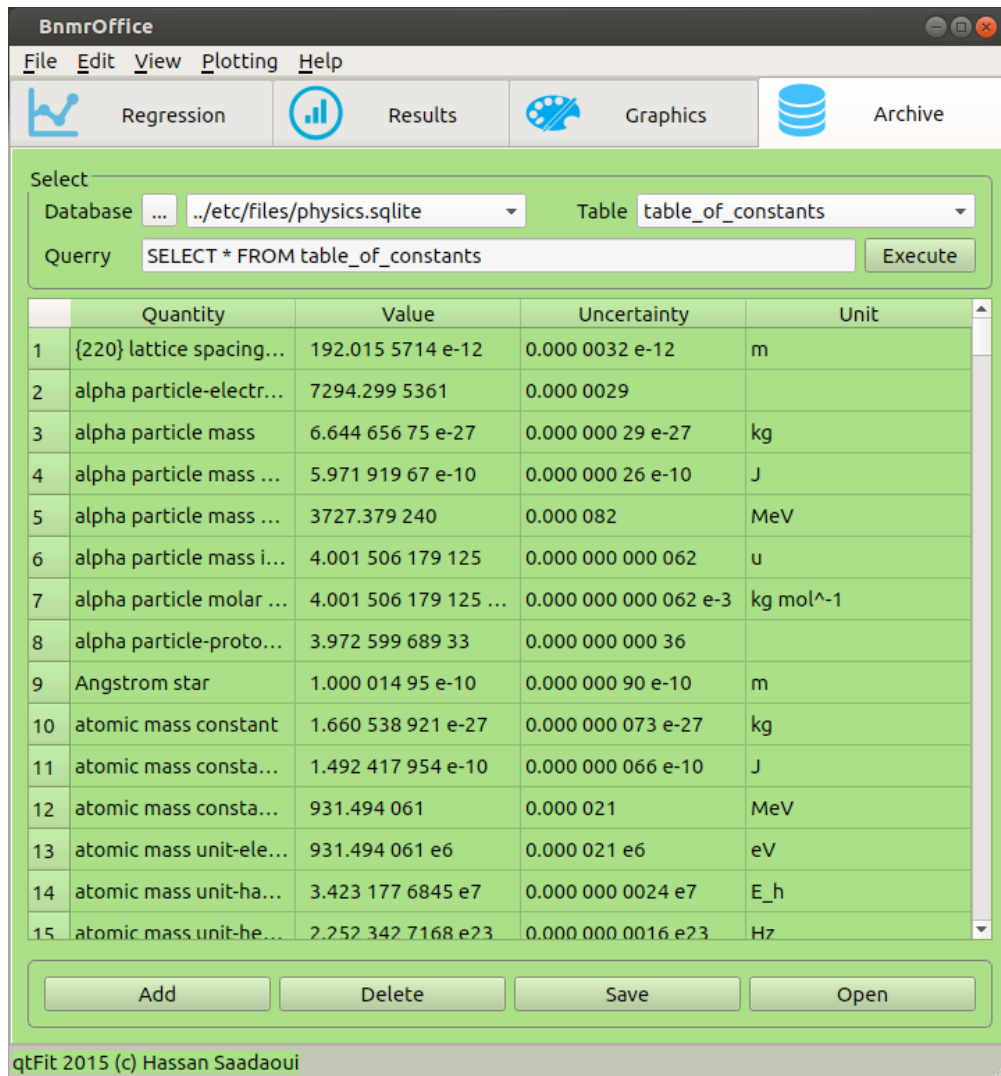


Figure 11: Database interface loaded with a table of physical constants.

5 Fitting functions

The user can write his own fitting functions in the directory `fct/`. A new function must be written in C++ but requires minimal programming knowledge of this language. At template of a typical function is as follows:

```
#include <iostream>
#include <fstream>
#include <math.h>
#include <stdio.h>
#include <vector>
#include <sstream>
#include <string.h>
#include <iostream>
using namespace std;
#include "parameters.h"

//Wrap in "C" for the compiler.
extern "C"
{
    //The default initial parameters loaded to the table.
    void defaultParameters(Parameters &defaults)
    {
        defaults.addParameter( "Amplitude" , 0.2, 0.001, 0.15, 1.0 );//par[0]
        defaults.addParameter( "Rate"      , 0.1, 0.001, 0.0, 1.0 );//par[1]
    }
    //This must have the same number of parameters as in the default parameters above.
    double function(double x, const std::vector<double> par)
    {
        return par[0]*exp( - par[1]*x); //par[0] is amplitude and par[1] is rate.
    }
}
```

The user must follow these instructions:

- Make a copy of the file `newFunction.cpp` found in `qtfit/etc/files`.
- Rename the file (eg: `newfct.cpp`) and save it in the folder `qtfit/fct/`.
- cd to the directory `qtfit/fct/`, and run the script `"compile"` as root
\$ `sudo ./compile name` (eg: `sudo ./compile newfct.cpp`)

This compiles the library and puts a copy in the functions folder (`/usr/local/qtfit/fct/`).

6 How-To-Use examples

6.1 Example 1

1. After the installation is complete, open the `qtFit` program from a terminal using the command `qtfit`.
2. Go to the page **Regression**.
3. To load a data file, click on the `toolButton` next to the label **"Data"**. Browse the directory `qtfit/etc/files` and load the files `1070.dat` to `1072.dat`, (Hold-on `Ctrl` key to select more than one file).
4. Select **ASCII** under "type" `comboBox`, and **xydy** under format. Leave the **Options** field empty for now.
5. Under Fitting; select the function **Exp** (for exponential fit), and format **Single** (for single fits), and **Migrad** (symmetric errors) for the type errors.
6. Write an appropriate name for the output file, any name is acceptable, or leave it as default `results.fit`.
7. Click on **Plot** `pushButton` at the bottom of the page to plot and view the selected files.
8. Click on **Fit** `pushButton` at the bottom of the page to perform the regression. If the fit converges, the results of the fit will be displayed in a new table. Go to **Graphics** page to see the plots of the raw data and fitting functions.
9. Click on **Next** to send the file `results.fit` to the results page. The columns of this file will be displayed, and the user can choose to plot a column against the other. On the left, check the box of **Amplitude** as x variable, and on the right-side of the table check the box **Rate** as y variable. Click on **Plot**, this will plot Rate versus Amplitude. Go to the page **Graphics** to see the new plot. If you click on **Clear** all checkboxes will be unselected. If you click on **Purge** all shown tables will be deleted. If you click on **Next** the **Graphics** window will be shown.

6.2 Example 2

1. To load a ".inf" file, click on the `toolButton` next to the label **"Data"**. Browse the directory `qtfit/etc/files` and load the file `example.inf`. The header of this file looks like this:

```
#FileName           Time(min) Temperature(K)
../etc/files/1070.dat 25          10,0.1
../etc/files/1071.dat 22          20,0.3
```

In this example, the ".inf" file contains three columns defined in the header (preceded by the sign #): the filename, a variable called Time in units of min, and a variable called Temperature in units of K. The arguments of each columns are given in the next rows. For Temperature, the given value is accompanied by the error. For example, "10,0.1" means Temperature = 10 ± 0.1 K for the file 1070.dat. In all columns (except the filename), the user can specify one or two errors (asymmetric), and all numbers must be separated by a comma. For example, something like "88,0.1,0.2" means Temperature = $[88 - 0.1, 88 + 0.2]$ K.

2. Repeat steps 4 to 9 in example 1.