

Simple Web Interface for Function Fitting and Report Writing

Pedro José Sebastião

*Departamento de Física, Instituto Superior Técnico,
TU Lisbon, Av. Rovisco Pais 1, 1049-001 Lisboa, Portugal;
Centro de Física da Matéria Condensada
Av. Prof. Gama Pinto 2, 1649-003 Lisboa, Portugal;
email: pedro.jose.sebastiao@ist.utl.pt*

November 16, 2009

Abstract

This document presents a simple tutorial on how to use the simple WebFfit interface. From registration on the Fitting Web Server to file sharing, from one dimensional fitting to multidimensional fitting procedures, from the fitting of simple mathematical expressions to more elaborated user defined functions, different examples are present that can be used as starting points for other user defined fitting problems.

1 Introduction

“What is the need for a web based fitting program?”

Over the years I realised as a Physics teacher and experimental physics researcher that the possibility to perform data analysis and function fitting, including report writing, over the Internet using a web browser would help me a lot on my everyday activities.

As a teacher I feel that it would be great to have a system that I could use to explain the procedures of non-linear numerical function fitting using the least-square minimization method and that the students could use without licence restrictions and operating system limitations.

As an experimental physics researcher I have collaborations with colleagues in different countries and I lack the possibility of using a common software that we can all use without the bonds imposed by software licences and operating systems flavors.

In 1993 after my thesis I was using a software package written in C and PERL using the minimization routine MINUIT from the CERN library [1]. This package was running first on Unix and later on Linux.

After several ports of this package to different flavors of Linux, live Linux CDs, to CygWin, and to Linux virtual machine guests running on Windows hosts I came to the conclusion that the best would be concentrate the development of this package on a single operating system and provide a web interface to operate with it.

2 Basic characteristics

Currently the WebFfit server runs on a PC with Debian Linux at IST (its URL is <http://lince.ist.utl.pt>). The full system includes a webserver, MySQL, C, Perl, Grace, L^AT_EX, ImageMagick, and, of course, gcc, to mention the main software packages.

2.1 User privilege levels

WebFfit has two type of users: *Normal Users* (NUsers) and *Privileged Users* (PUsers).

- *Normal Users* are restricted only by the fact that they **can not** write free C code. The fitting function is restricted to a single line of C code (including calls to math functions and to other math functions provided by WebFfit).
- *Privileged Users* can write their own functions with almost complete freedom.

The change of privileges is performed by the system manager when required by the users.

2.2 WebFfit running modes

WebFfit has three running modes: *Basic*, *Advanced*, and *Expert*.

- The *Basic* mode is directed to those who want to perform a 1D fit of a function $y = f(p_1, p_2, \dots, p_n, x)$ where y is the dependent variable, x is the independent variable and p_1, \dots, p_n are the fitting parameters. A XY plot is produced according to the user settings.
- The *Advanced* mode makes possible to fit 2D functions in the form of $z = f(p_1, p_2, \dots, p_n, x, y)$ where z is the dependent variable, x is the 1st independent variable, y is the 2nd independent variable, and p_1, \dots, p_n are the fitting parameters. For each value of y a three columns' table $xzErr_Z$ has to be introduced and a XZ plot is produced according to the user settings.
- The *Expert* mode gives to possibility to fit function of the type

$$z = f(p_1, p_2, \dots, p_n, x, y)$$

where z is the dependent variable, x is the 1st independent variable, y is the 2nd independent variable, and p_1, \dots, p_n are the fitting parameters. Three columns' tables $xzErr_Z$ for y values and/or $yzErr_Z$ for x values can be included and both YZ and XZ plots are produced according to the user settings. Please note that PUsers can take full advantage of this fitting mode.

2.3 Report writing

In addition to the fitting module the WebFfit package has also a report writing module that is used by the fitting module to produce fitting reports in a quick way.

The *Report Writing* module is nevertheless independent of the fitting module and can be used to write more elaborated documents with the possibility to include (upload) figures other than those produced by the fitting module.

Report Writing uses L^AT_EX (actually PDFL^AT_EX) to produce the PDF documents. The document class is by default *RevTeX*.

2.4 Registration

The registration requires a valid email account since the first password is sent by email to the users account as part of the validation process.

The password sent by email can then be used by the user to change the password to something of his choosing. The passwords are encrypted in the system and can not be recovered. The system uses a secured webserver for this process, therefore your passwords are not sent open on the web. The resetting of the passwords is done by the same process.

The registration is completed with very Little information.



Figure 1: <http://lince.cii.fc.ul.pt/>

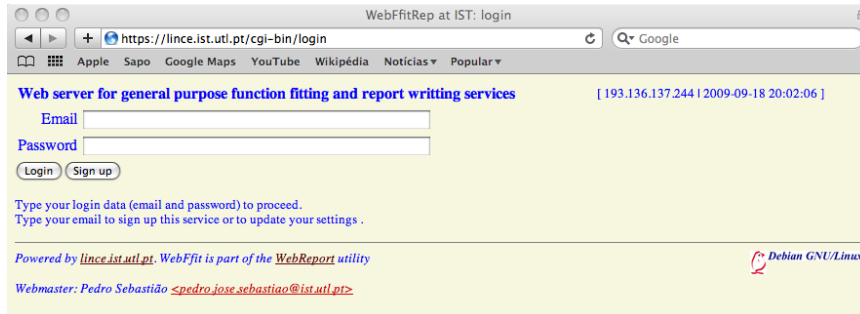


Figure 2: WebFfit login window

3 Tutorial

3.1 Login

After registration the user can login the system by accessing the lince web site with URL <http://lince.cii.fc.ul.pt/>, as illustrated in fig. 1. The user signs in using the email as his username and the password.

The first webpage after signing in allows the user to browse his personal working area, as illustrated in fig. 5. A list of folders are presented. The first time there is just one project available entitled *Tutorial1*.

Besides the links to the working areas the user may find a log out link and some useful information concerning used/free available memory space. For each project area there are two additional links one that allows the user to *browse* list the folder's contents and access each file. The other is *manage* and gives the possibility not only to list the folders' content but also to remove one or more files from the folder. Actually, if the file "." is remove the entire folder is removed. *manage* utility allows also the user to share the folder with other users. For this the owner of the folder needs to know the other user's email.

If another user is sharing a project folder with you, it will be easy to identify it, as illustrated in fig. 6. In this case the *manage* link is replaced by *remove* since a shared folder can not be directly changed by a guest user. The guest user can only remove this link form his own MyFits area.

3.2 Tutorial 1

The project *Tutorial 1* can be very useful as a starting point to make different types of function fits. The main page is illustrated in fig. 7.

The page has three main regions: *header*, *body*, and *footer* (I will not spend time with the footer).

- The *header* presents on the right side the links that allow the user to change



Figure 3: WebFfit Registration window 1

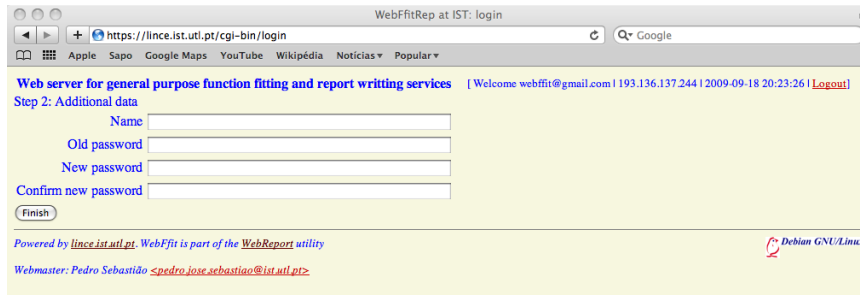


Figure 4: WebFfit Registration window 2

the fitting modes (*Basic*, the default, *Advanced*, and *Expert*), to *Upload* the contents of a previous fit that was saved locally in you computer, to return to the *MyFits* page, or to *Logout*. In addition the user can list the different fits available in the project area, *Recover* the chosen one, or *Download* it and save it in you personal computer. On the left side the user has the possibility to save the current fit and choose the project area name and the name of the current fit. The date and time will be added to the name of the fit so that you will be able to go back in history.

- The *body* has three main regions that actually correspond to the **three** steps necessary to perform a function fit to a set of experimental data: 1° *Data*, 2° - *Plot Parameters*, and 3° - *Function and Parameters*.
 1. *Data* has two text boxes. The first has by default a few comment lines that start with “#” and a few lines with numbers arranged in three columns and correspond the values of the variables x , y , ϵ_y , where ϵ_y is the error estimate of y . The second box has a few lines that start by “#” by default and correspond to examples of operations which you can perform with the data in the *Data Box*. I think it is self explanatory but you should try it and contact me if you have problems with it. *Calculate* does what it means applies the calculation rules to the data.
 2. *Plot Parameters* allows for the set the plot details and I think it is self explanatory. One suggest you to check the grace doc file on how to write subscripts, superscripts, etc..
 3. The *Function and Parameters* part is the most interesting one. *NU* users will see it like in fig. 7. *PU* users will see it like in fig. 8. The difference is the option to call the *AuxCode Editor* and the *Additional C code* box. Depending on the *Fitting mode* the contents of this area might change. You might try to change to *Advanced* mode and *Recover* the fit **Advanced-1**.
Once you defined the variables name, declare them, write your C code to calculate the value of the dependent variable you have to *Compile* the



Figure 5: Myfits page



Figure 6: Myfits page with a shared folder

function. If you get back the string **Compiled without errors** everything is OK and you should see a table below where you will identify the fitting parameters that you defined in your function. There is a compilation log file where you can check what is wrong with your function. This table has buttons to *fix* or *free* each parameter, and three columns with boxes where: *Value* can receive the starting value in the numerical fit and will contain the output result of your fit, *Min*, and *Max* allow you to set the lower and upper limits in the parameters space where the fitting program will have to try to find the best fit. A last column *Error* will present you the error estimate of each fitted parameter.

3.3 Arithmetic

Arithmetic operations are allowed in the fitting parameter cells and in the place of *data* column values. These operations are performed prior to any other action.

3.4 Basic examples

3.4.1 Straight line curve fit

If you successfully compiled the function $y = a + bx$ that is set by default when you entered the *Tutorial 1* area and you fit the default data you will get a plot like this Please check the various link available now after the fit was made. Please check also the *WebReport* link.

3.4.2 Other basic examples

I invite you to go to the *Basic* fitting mode and try to recover each *Basic-1,2,3,4,5* examples and study them. I think they are self explanatory.

I would like to point to you that if your function can be written as the sum of different functions like in

$$y = y = f_1(p_1, p_2, \dots, p_n, x) + f_2(p_1, p_2, \dots, p_n, x) + \dots \quad (1)$$

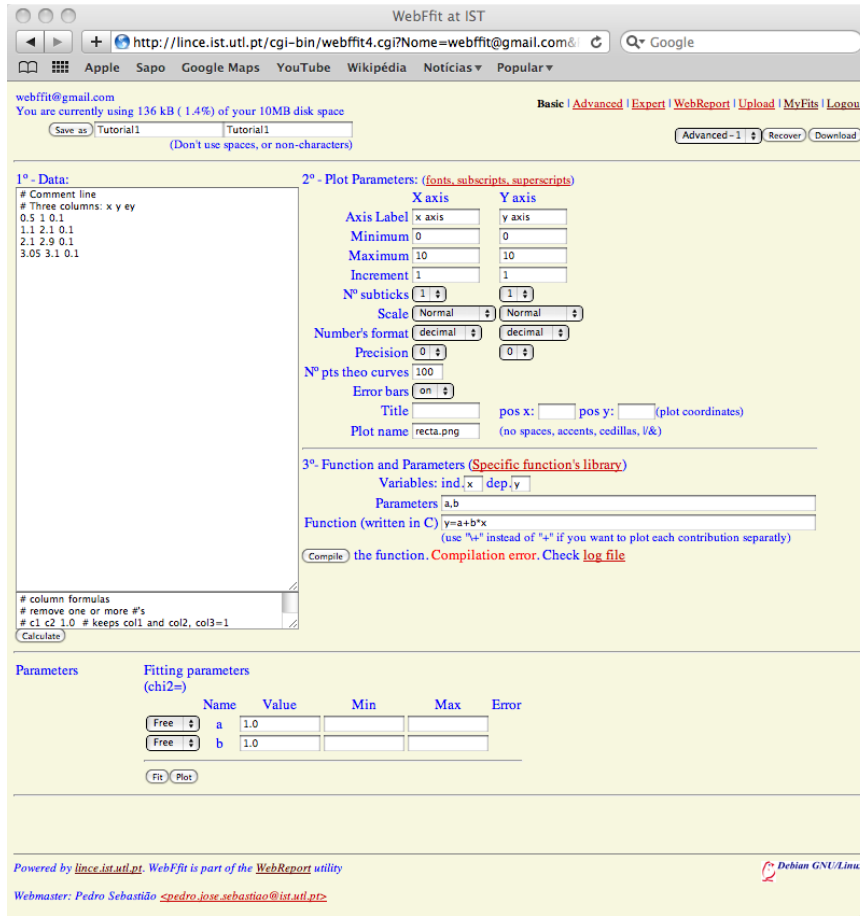


Figure 7: Fitting page

you have the possibility to produce plots where each function f_1, f_2, \dots is represented by lines of your choosing. In order to obtain this you write the following

$$y = a \setminus + bx \quad (2)$$

as your function and compile it. Fig. 10 was produced after selecting *dashed* and *dot dashed* curves in red and green colors for the two components in the plot menu.

3.5 Advanced example

After going to the *Advanced* fitting mode *Recover* the “Advanced-1” example. This example corresponds to the fitting of

$$y = \frac{A_0}{m\sqrt{((2\pi f)^2 - (K/m)^2)^2 + (2\pi fb/m)^2}} \quad (3)$$

to a several sets of experimental results obtained in a spring-mass oscillating systems for different values of mass m . b is the friction parameter, K is the spring’s elastic constant, f is the frequency of the external oscillating force, and A_0 is a constant.

There are several data sets (f, y, ϵ_y) one for each value of m . Please note how the plots are defined.

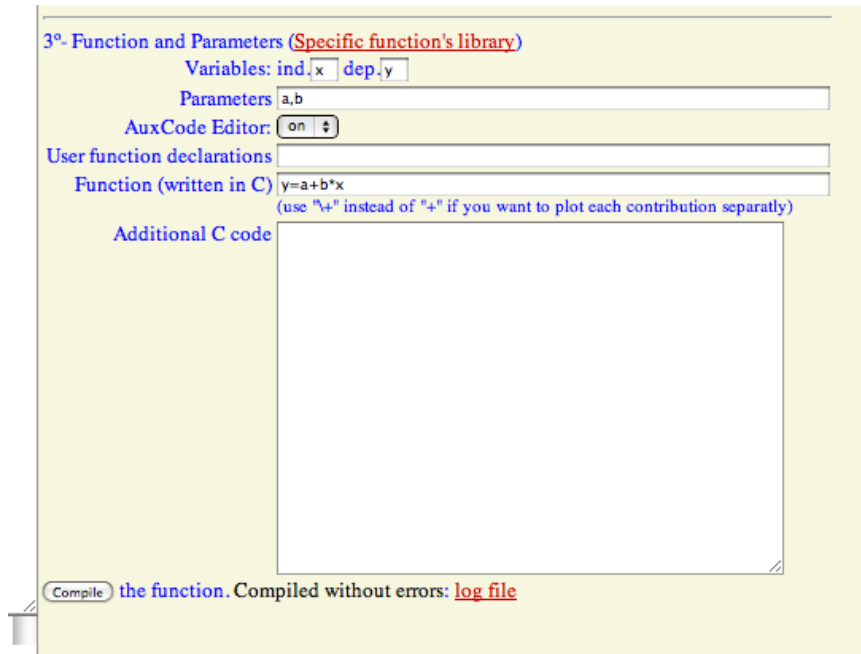


Figure 8: Fitting page

3.6 Expert example

Please go to the *Expert* fitting mode *Recover* the “Expert-1” example. This example refers to the fit of eq. 3 but with the following assumption: if the value of the first column is less than 0.1 then it is assumed that it refers to a value of frequency, f , and the data set was obtained for a given value of m ; if the value of the first column is larger than 0.1 then it refers to a value of mass m and the data set was obtained for a fixed frequency f . The plots must therefore be different. Please check the different menus and discover the differences with respect to the previous example.

PUsers can take full advantage of this fitting mode since they can program elaborated functions where the data sets can be coded in such a way that a n -dimensions fit can be performed.

4 Acknowledgments

I would like to thank the open source community for the wonderful software available on the web that made possible to provide you with this webserver.

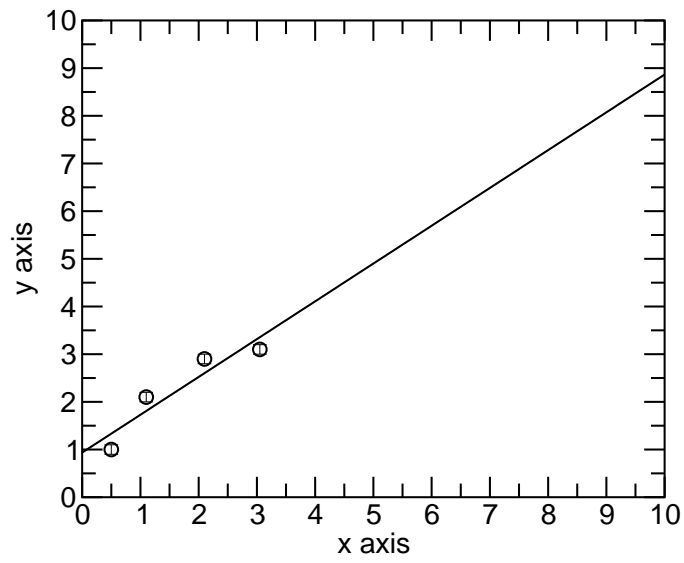


Figure 9: Straight line curve fit

References

- [1] <http://lcgapp.cern.ch/project/cls/work-packages/mathlibs/minuit/home.html>

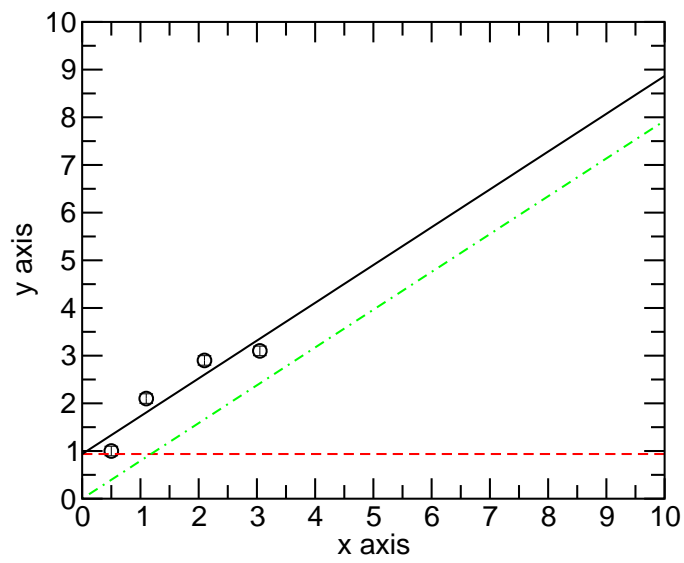


Figure 10: Straight line curve fit with each component shown.