



Genfit: Esecuzione di fit online con ROOT

Francesco G. Gravili - Università del Salento &
INFN Sezione di Lecce



Informazioni utili

Idea:

Eseguire un fit a dati sperimentali senza installare nulla sul proprio computer e lavorando su un unico file di configurazione

Manuale d'uso (in lingua inglese):

<https://frgigr.gitlab.io/genfit/>

Sito internet su cui operare direttamente:

<https://mybinder.org/v2/gl/frgigr%2Fgenfitforstudents/HEAD?labpath=fit.ipynb>

Genfit: File di configurazione

- Il file di configurazione è scritto in formato JSON
- JSON è un formato tabulare, non contano le indentazioni o l'allineamento
- Semplice file di testo, modificabile con qualsiasi editor
- Gli *oggetti* sono racchiusi tra { }
- Le *stringhe di testo* sono racchiuse tra " "
- I *vettori* sono racchiusi tra []
- I numeri e le variabili booleane non hanno nulla attorno
- Attenzione ai separatori:
 - Punto . come separatore interi-decimali
 - Virgola , come separatore tra campi o elementi di un vettore
- **NB:** I campi da modificare sono visualizzati in viola!

```
measurements.json
1 {
2   "title": "My awesome TGraph with Errors",
3   "x-title": "X axis title",
4   "x-units": "X units",
5   "y-title": "Y axis title",
6   "y-units": "Y units",
7   "errFlag": "xyerr",
8   "fit": {
9     "formula": "[0]+[1]*x",
10    "parameters": [5.1, 2.2],
11    "parNames": ["q", "m"],
12    "rangeMin": 0.0,
13    "rangeMax": 10.0
14  },
15  "points": [
16    { "point": [0.485, 6.00, 0.003, 0.05], "use": true },
17    { "point": [0.927, 7.00, 0.002, 0.05], "use": true },
18    { "point": [1.335, 8.00, 0.003, 0.05], "use": true },
19    { "point": [1.773, 9.00, 0.002, 0.05], "use": true },
20    { "point": [2.226, 10.00, 0.002, 0.05], "use": true },
21    { "point": [2.670, 11.00, 0.002, 0.05], "use": true },
22    { "point": [3.121, 12.00, 0.003, 0.05], "use": true },
23    { "point": [3.573, 13.00, 0.002, 0.05], "use": true },
24    { "point": [4.043, 14.00, 0.002, 0.05], "use": true }
25  ]
26 }
```

File di configurazione: introduzione

- **Title:** titolo del vostro fit
- **x-title:** titolo della grandezza in ascissa
- **x-units:** unità di misura della grandezza in ascissa
- **x-title:** titolo della grandezza in ordinata
- **x-units:** unità di misura della grandezza in ordinata

NB: In caso di grandezze adimensionali, lasciare la stringa vuota, cioè ""

```
measurements.json
1 {
2   "title": "My awesome TGraph with Errors",
3   "x-title": "X axis title",
4   "x-units": "X units",
5   "y-title": "Y axis title",
6   "y-units": "Y units",
7   "errFlag": "xyerr",
8   "fit": {
9     "formula": "[0]+[1]*x",
10    "parameters": [5.1, 2.2],
11    "parNames": ["q", "m"],
12    "rangeMin": 0.0,
13    "rangeMax": 10.0
14  },
15  "points": [
16    { "point": [0.485, 6.00, 0.003, 0.05], "use": true },
17    { "point": [0.927, 7.00, 0.002, 0.05], "use": true },
18    { "point": [1.335, 8.00, 0.003, 0.05], "use": true },
19    { "point": [1.773, 9.00, 0.002, 0.05], "use": true },
20    { "point": [2.226, 10.00, 0.002, 0.05], "use": true },
21    { "point": [2.670, 11.00, 0.002, 0.05], "use": true },
22    { "point": [3.121, 12.00, 0.003, 0.05], "use": true },
23    { "point": [3.573, 13.00, 0.002, 0.05], "use": true },
24    { "point": [4.043, 14.00, 0.002, 0.05], "use": true }
25  ]
26 }
```

File di configurazione: fit

Impostazioni della funzione usata per eseguire il fit

- **formula:** formula da utilizzare con relativi parametri
- **parameters:** valori di inizializzazione dei parametri
- **parNames:** nomi associati ai parametri
- **rangeMin:** valore minimo da cui iniziare il fit
- **rangeMax:** valore massimo in cui terminare il fit

NB: Il range fa riferimento alle ascisse!

```
measurements.json
1 {
2   "title": "My awesome TGraph with Errors",
3   "x-title": "X axis title",
4   "x-units": "X units",
5   "y-title": "Y axis title",
6   "y-units": "Y units",
7   "errFlag": "xyerr",
8   "fit": {
9     "formula": "[0]+[1]*x",
10    "parameters": [5.1, 2.2],
11    "parNames": ["q", "m"],
12    "rangeMin": 0.0,
13    "rangeMax": 10.0
14  },
15  "points": [
16    { "point": [0.485, 6.00, 0.003, 0.05], "use": true },
17    { "point": [0.927, 7.00, 0.002, 0.05], "use": true },
18    { "point": [1.335, 8.00, 0.003, 0.05], "use": true },
19    { "point": [1.773, 9.00, 0.002, 0.05], "use": true },
20    { "point": [2.226, 10.00, 0.002, 0.05], "use": true },
21    { "point": [2.670, 11.00, 0.002, 0.05], "use": true },
22    { "point": [3.121, 12.00, 0.003, 0.05], "use": true },
23    { "point": [3.573, 13.00, 0.002, 0.05], "use": true },
24    { "point": [4.043, 14.00, 0.002, 0.05], "use": true }
25  ]
26 }
```

File di configurazione: dati sperimentali ed incertezze

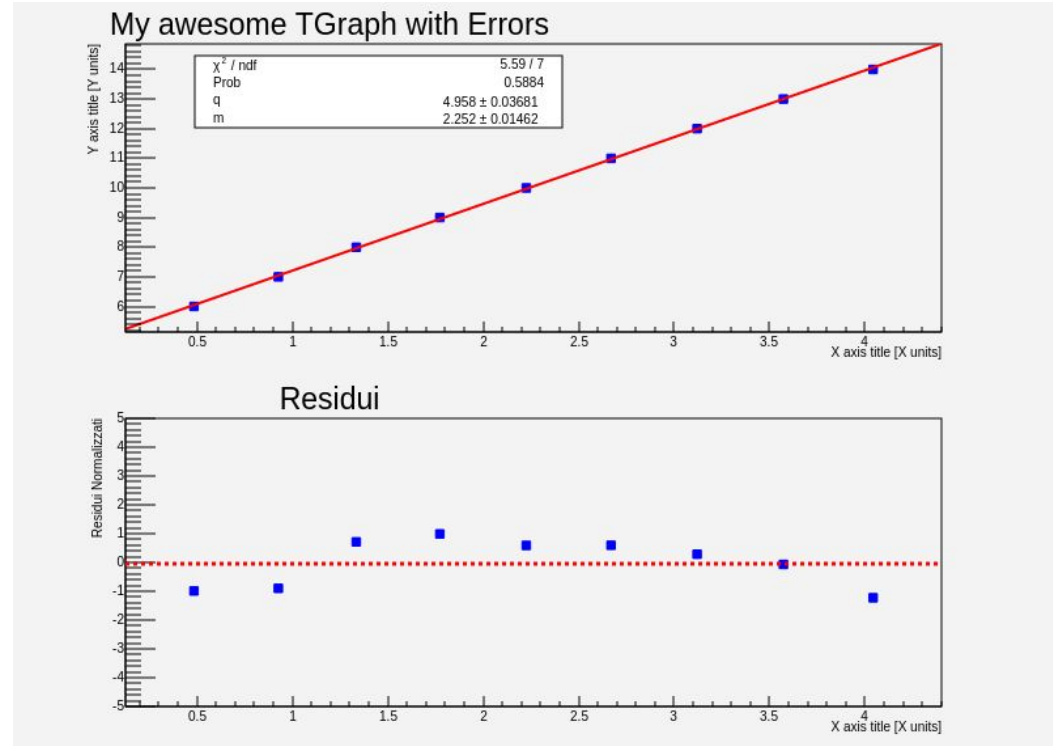
- Formato variabile a seconda delle incertezze:
 - Primo e secondo elemento: x , y
 - Terzo elemento: *incertezza sulla variabile x*
 - Quarto elemento: *incertezza sulla variabile y*
- x e y sono obbligatori, SEMPRE!
- **errFlag**, 3 opzioni:
 - **noerr** (incertezze assenti)
Il generico punto sarà: *"point": [x, y]*
 - **yerr** (incertezze solo sulla y)
Il generico punto sarà: *"point": [x, y, σ_y]*
 - **xyerr** (incertezze su entrambe le variabili)
- **use**: true/false per includere/escludere il punto dal fit

```
measurements.json
1 {
2   "title": "My awesome TGraph with Errors",
3   "x-title": "X axis title",
4   "x-units": "X units",
5   "y-title": "Y axis title",
6   "y-units": "Y units",
7   "errFlag": "xyerr",
8   "fit": {
9     "formula": "[0]+[1]*x",
10    "parameters": [5.1, 2.2],
11    "parNames": ["q", "m"],
12    "rangeMin": 0.0,
13    "rangeMax": 10.0
14  },
15  "points": [
16    { "point": [0.485, 6.00, 0.003, 0.05], "use": true },
17    { "point": [0.927, 7.00, 0.002, 0.05], "use": true },
18    { "point": [1.335, 8.00, 0.003, 0.05], "use": true },
19    { "point": [1.773, 9.00, 0.002, 0.05], "use": true },
20    { "point": [2.226, 10.00, 0.002, 0.05], "use": true },
21    { "point": [2.670, 11.00, 0.002, 0.05], "use": true },
22    { "point": [3.121, 12.00, 0.003, 0.05], "use": true },
23    { "point": [3.573, 13.00, 0.002, 0.05], "use": true },
24    { "point": [4.043, 14.00, 0.002, 0.05], "use": true }
25  ]
26 }
```

Genfit: grafico del fit

Informazioni visualizzate:

- χ^2 / gradi di libertà
- Probabilità di ottenere un chi-quadro maggiore
- Parametri liberi del fit
- Grafico del fit
- Grafico dei residui normalizzati (non fondamentale, al momento)



Genfit: main page

fit.ipynb

File Edit View Run Kernel Tabs Settings Help

Python 3 (ipykernel)

GenFit

Upload your own file in JSON format or edit `measurements.json` directly. Before uploading a new file, remember, check JSON format consistency otherwise validation will fail!

All fields are mandatory! Be careful, you may want to change input file name (in the `configFile` variable) with respect to the default one.

The code below makes use of `JavaScript ROOT`, so the output canvas is fully interactive. Feel free to play with it!

Author: E.Gorini (email: edoardo.gorini@le.infn.it)

Review and Adaptation: F.G.Gravili (email: francesco.giuseppe.gravili@le.infn.it)

NOTE: Ignore errors related to missing `libGL.so` and `libEve.so` libraries. They're optional ROOT libraries and results will not be affected at all

```
[ ]: from ROOT import TCanvas, TF1, TGraph, TGraphErrors, TLine, TMath, gStyle
from ROOT import kBlue, kRed
from array import array
import json
from jsonschema import validate
import os

# parse input file with general settings and data points, before validating it as JSON
configFile = os.path.join(os.getcwd(), 'measurements.json')
jschema = json.load(open(os.path.join(os.getcwd(), 'data_schema.json')))
data = []
try:
    with open(configFile, 'r') as jsData: data = json.load(jsData)
except Exception as exc:
    raise RuntimeError('FATAL: Unable to open measurement file with general settings and data points') from exc

The following command will validate your JSON file: in case of errors, do NOT ignore them. Please, fix inconsistencies in the configuration file before proceeding!

[ ]: validate(instance=data, schema=jschema)

[ ]: # set some graphics
gStyle.SetTitleStyle(0)
gStyle.SetTitleBorderSize(0)
gStyle.SetTitleX(0.3)
gStyle.SetTitleY(1.0)
gStyle.SetTitleH(1)
```

Simple Python 3 (ipykernel) | Idle Mem: 141.20 / 4096.00 MB Mode: Command Ln 1, Col 1 fit.ipynb 1

Genfit: main page

Struttura a blocchi eseguibili singolarmente:

- Pulsante per visualizzare il contenuto online
- Ignorare il resto e concentrarsi su due files:
 - *fit.ipynb* -> fit
 - *measurements.json* -> file dati

I blocchi sono evidenziati lateralmente da numeri e strisce blu. Tra le parentesi quadre, compaiono due tipi di informazioni:

- *Numero*, ad esempio [1]: il blocco è stato eseguito
- *Asterisco* [*]: il blocco è in fase di esecuzione

The screenshot shows the JupyterLab interface for the 'fit.ipynb' notebook. The file browser on the left lists files: 'utils', 'data_sche...', 'environme...', 'fit.ipynb', and 'measur...'. The code editor displays the following code:

```
GenFit

Upload your own file in JSON format or edit measurements.json directly. Before uploading a ne

All fields are mandatory! Be careful, you may want to change input file name (in the configFile variab

The code below makes use of JavaScript ROOT, so the output canvas is fully interactive. Feel free to

Author: E. Gorini (email: edoardo.gorini@le.infn.it)

Review and Adaptation: F.G. Gravili (email: francesco.giuseppe.gravili@le.infn.it)

NOTE: Ignore errors related to missing libGL.so and libEve.so libraries. They're optional ROC

[ ]: from ROOT import TCanvas, TF1, TGraph, TGraphErrors, TLine, TMath, gStyle
from ROOT import kBlue, kRed
from array import array
import json
from jsonschema import validate
import os

# parse input file with general settings and data points, before validating it
configFile = os.path.join(os.getcwd(), 'measurements.json')
jschema = json.load(open(os.path.join(os.getcwd(), 'data_schema.json')))
data = []
try:
    with open(configFile, 'r') as jsData: data = json.load(jsData)
except Exception as exc:
    raise RuntimeError('FATAL: Unable to open measurement file with general set

The following command will validate your JSON file: in case of errors, do NOT ignore them. Please,

[ ]: validate(instance=data, schema=jschema)

[ ]: # set some graphics
gStyle.SetTitleStyle(0)
gStyle.SetTitleBorderSize(0)
gStyle.SetTitleX(0.3)
gStyle.SetTitleY(1.0)
gStyle.SetTitleH(.1)
```

Genfit: main page

Struttura a blocchi eseguibili singolarmente:

- **Blocco 1: Introduzione**
Nessuna operazione da eseguire
- **Blocco 2: Apertura file dati**
NB: Ignorare qualsiasi errore che coinvolge le librerie *libGL.so* e *libEve.so*
- **Blocco 3: Validazione file dati**
Nel caso sia stato commesso qualche errore nella struttura. Se non è visualizzato alcun errore, procedere al blocco successivo, altrimenti sistemare il file!

The screenshot shows the Jupyter Notebook interface for 'fit.ipynb'. The left sidebar displays a file explorer with folders like 'utils' and files like 'data_sche...', 'environme...', 'fit.ipynb', and 'measurme...'. The main area contains the notebook content, which is annotated with colored boxes and arrows:

- Red box:** Contains the 'GenFit' title and introductory text: 'Upload your own file in JSON format or edit `measurements.json` directly. Before uploading a new file, make sure you have the correct permissions. All fields are mandatory! Be careful, you may want to change input file name (in the `configFile` variable). The code below makes use of JavaScript `ROOT`, so the output canvas is fully interactive. Feel free to try it out. Author: E.Gorini (email: edoardo.gorini@le.infn.it). Review and Adaptation: F.G.Gravili (email: francesco.giuseppe.gravili@le.infn.it). NOTE: Ignore errors related to missing `libGL.so` and `libEve.so` libraries. They're optional ROOT dependencies.' A red arrow points from this box to the text 'Nessuna operazione da eseguire'.
- Blue box:** Contains the Python code for loading and validating the JSON file. A blue arrow points from this box to the text 'NB: Ignorare qualsiasi errore che coinvolge le librerie *libGL.so* e *libEve.so*'.
- Orange box:** Contains the validation command: `validate(instance=data, schema=jsonschema)`. An orange arrow points from this box to the text 'altrimenti sistemare il file!'.

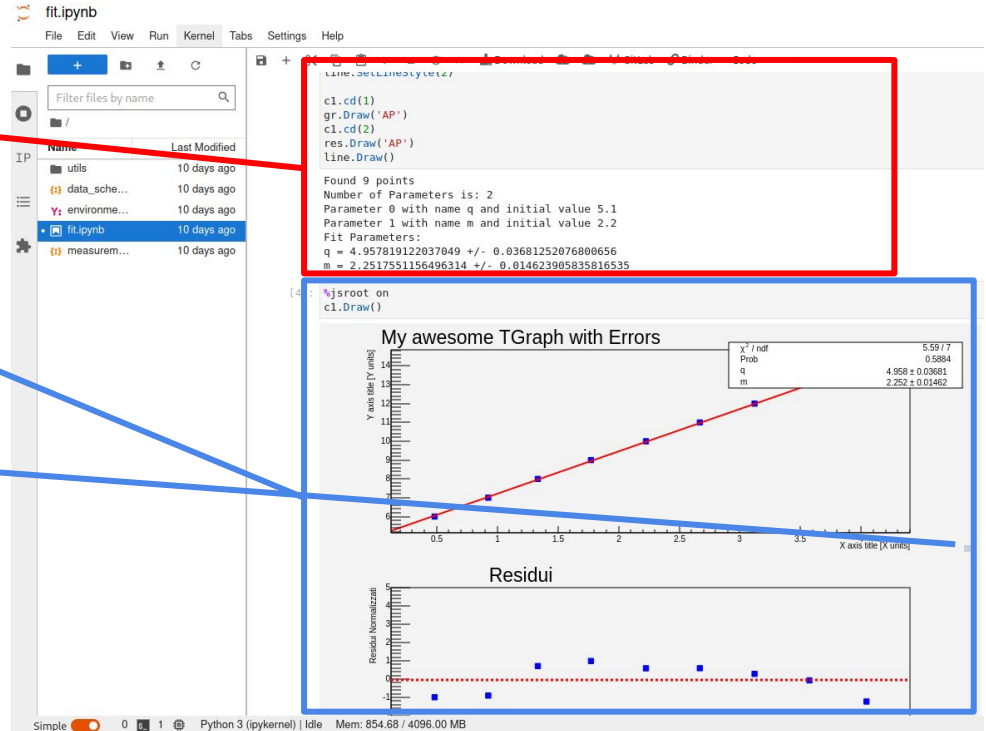
Genfit: main page

- **Blocco 4: Fit**

Valutare la presenza di eventuali avvisi o messaggi di errore

- **Blocco 5: Visualizzazione dei risultati**

La struttura della canvas è altamente interattiva per eventuali modifiche a valle. Tra le varie opzioni, vi è la possibilità di esportazione in formato *png*. Fare riferimento alle icone quadrate



Genfit: esecuzione

Due modalità di esecuzione dei blocchi:

1. Pulsante apposito
2. Da tastiera: *Shift+Enter*
3. Menu *Run* -> *Run Selected Cells*

Una volta giunti alla fine, nel caso si voglia ripetere il fit o cambiare qualche parametro, prima di ripartire dall'inizio, occorre ricaricare il kernel:

1. Pulsante apposito
2. Menu *Kernel* -> *Restart Kernel*

The screenshot shows the GenFit web interface. The top navigation bar includes 'File', 'Edit', 'View', 'Run', 'Kernel', 'Tabs', 'Settings', and 'Help'. The 'Run' and 'Kernel' menus are highlighted with red and blue boxes respectively. A red arrow points from the 'Run' menu to the first list item, and another red arrow points from the 'Run Selected Cells' option to the second list item. A blue arrow points from the 'Kernel' menu to the second list item. A blue arrow points from the 'Restart Kernel' option to the first list item. The main content area shows a file list with 'fit.ipynb' selected. The right sidebar contains the 'GenFit' title and instructions for uploading files and using JavaScript R.

Name	Last Modified
/	
utils	10 days ago
data_sche...	10 days ago
environme...	10 days ago
fit.ipynb	a minute ago
measuram...	10 days ago

GenFit

Upload your own file in JSON format or edit

All fields are mandatory! Be careful, you ma

The code below makes use of [JavaScript R](#)

Author: E.Gorini (email: edoardo.gorini@le.i)

Review and Adaptation: F.G.Gravili (email: f.g.gravili@le.i)

NOTE: Ignore errors related to missing `li`

Genfit: chiusura ed uscita

Al termine dei lavori, chiudere la sessione:

1. File -> Shut Down
2. Confermare lo spegnimento, nella finestra visualizzata

Shutdown confirmation

Please confirm you want to shut down JupyterLab.



**NB: Una volta confermato l'arresto, tutte le modifiche andranno perse!
Ricordarsi di scaricare e salvare il file di configurazione localmente**

