The Bragg curve

P. Ronchese
Dipartimento di Fisica e Astronomia “G.Galilei”

Università di Padova

“Object oriented programming and C++” course

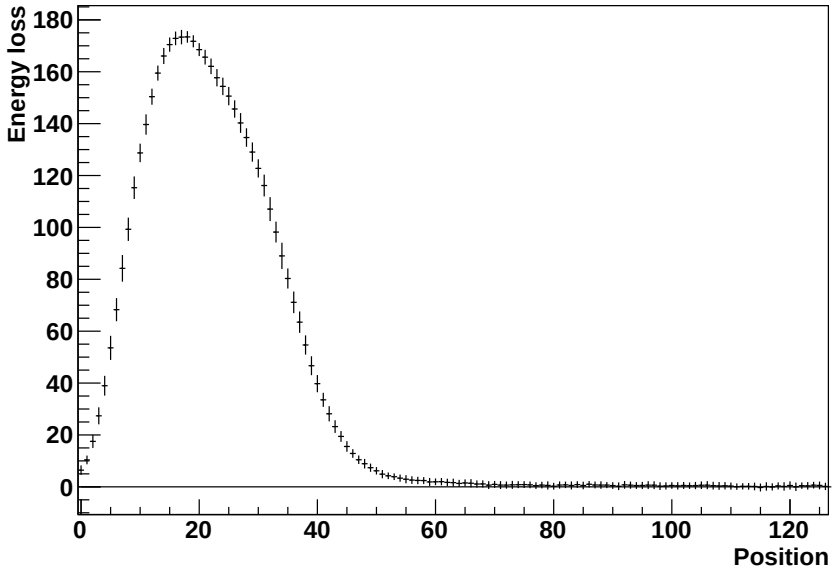
Energy measurement in α decay

Data to read and analyze have been collected in the nuclear physics lab. course.

Radioactive elements emit α particles, absorbed by a detector giving the energy lost by the particles while it goes through the material.

- The energy loss rises up to a maximum.
- The energy loss, after having reached the maximum, decreases to zero.
- This behaviour can be drawn on a plot, the “Bragg curve”
- The total energy is given by the sum of the energy losses in all the steps

Bragg curve



Both axes use arbitrary units, there's no special physical meaning in this plot.

- Energy loss at each step is computed from the measurements by subtracting a constant, called "background".
- The background can be estimated by averaging the measurements in the last part of the tail
- The plot above has been produced after background subtraction.

Events

The file `bragg_events.txt` contains the data, written in text form

Each event is written on a line of text containing:

- 1 event identifier (i.e. an `int`)
- 1 number, comprised between 120 and 128, of energy measurements,
- the list of energy measurements, each given as an `int` .

Energies data dump - version 1

Read the text file and produce a dump onto the screen (use the “Particle data dump - version 1” example as reference)

- Create an array of `int`s to contain energies.
- Create functions to:
  - read an event from file,
  - dump an event onto the screen,
- Create a main function to loop over the file and dump the events.

Energies data dump - version 2

Read the text file and produce a dump onto the screen (use the “Particle data dump - version 2” example as reference)

- Create a `struct Event` to contain event data, with members corresponding to the data listed above.
- Create functions to:
  - read an event from file,
  - dump an event onto the screen,
  - free the memory used by the event.
- Create a main function to loop over the file and dump the events.

Energies data mean - version 1

Read the text file and compute mean and r.m.s. energies for each point
(use the “Particle data mean - version 1” example as reference)

- Create functions to:
  - compute energy and energy squares sums for each point,
  - select events with total energy between 6000 and 7500,
  - compute energy mean and r.m.s. for each point.
- Modify the main function to hold sums and call statistical functions.

Energies data mean - version 2

Read the text file and compute mean and r.m.s. energies for 3 Bragg curves and background (use the “Particle data mean - version 2” example as reference)

- Modify the version 1 to use `class`es:
  - create a `class` to contain event data,
  - create a `class` to compute statistics.
  - select events with total energy in the following ranges:
    - between 3000 and 5000,
    - between 6000 and 6499,
    - between 6500 and 6799,
    - between 6800 and 7200.
- Modify the main function to use the new `class`es.

Energies data mean - version 3

Read the text file and compute mean and r.m.s. energies
for 3 Bragg curves and background
(use the “Particle data mean - version 3” example as reference)

- Modify the version 2 to use STL:
  - use a `std::string` to handle input file name
  - use a `std::vector` to store energies
- Modify the main function to use the modified `class`es.

Energies data mean - version 4

Read the text file and compute mean and r.m.s. energies for 3 Bragg curves and background (use the “Particle data mean - version 4” example as reference)

- Modify the version 3 to use only `class`es in place of global functions:
  - create a `class` to read events,
  - create a `class` to dump events,
  - create a `class` to compute mean and rms energies: inside a `class` performing the general analysis steering create several (i.e. 3+1) instances of a `class` computing mean and rms energies, selecting different total energy ranges
- Create the `class`es as derivations of interfaces to get events and process them.
- Modify the main function to use the new `class`es.

Energies data plot - version 1

Plot mean and r.m.s. energies
for 3 Bragg curves and background

- Modify the mean-version 4 to include graphic plots and allow multiple plots handling:
  - inside the general analysis steering `class` pair each object computing mean and rms energies with a `TH1F` object
  - create, set and save plots in the same `class` , too.
- All other `class`es and the main function stay unchanged.

Energies data plot - version 2

Plot mean and r.m.s. energies
for 3 Bragg curves and background
(use the “Particle mass reconstruction - version 2” as reference)

- Modify the version 1 to use a “factory” to create a data source.
- Take `AnalysisInfo` and `SourceFactory` from the particle analysis example.
- Modify the main function to use the new `class`es .

Energies data plot - version 3

Plot mean and r.m.s. energies
for 3 Bragg curves and background
(use the “Particle mass reconstruction - version 3” as reference)

- Modify the version 2 to use a “factory” to create analyzer objects:
  - take the new versions of `AnalysisSteering` and `AnalysisFactory` from the particle analysis example,
  - modify `EventDump` and `EnergyDist` to be handled by `AnalysisFactory` .
- Modify the main function to use the new `class` .

Energies data plot - version 4

Plot mean and r.m.s. energies
for 3 Bragg curves and background
(use the “Particle mass reconstruction - version 4” as reference)

- Modify the version 3 to use a “dispatcher” to loop over events:
  - take the new versions of `AnalysisSteering` and `EventSource` from the particle analysis example,
  - modify `EventDump` and `EnergyDist` to be `ActiveObserver`s,
- create a new analyzer `BGCalc` to compute background and printout it at execution end,
- create a `class TotalEnergy` to compute total energy, and declare it being a `LazyObserver` .
- Modify the main function to use the new `EventSource` .

Energies data plot - version 5

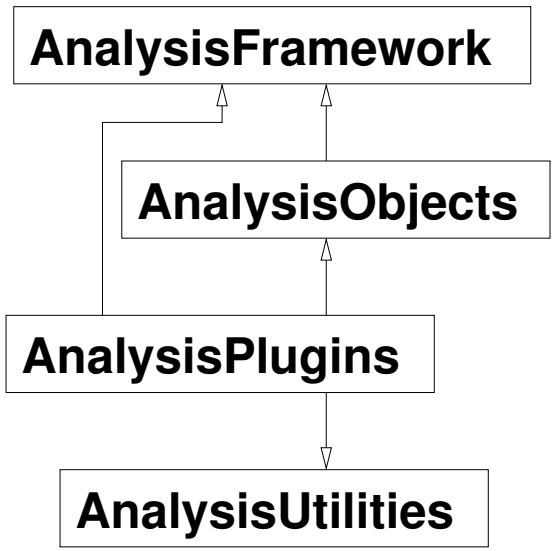
Plot mean and r.m.s. energies
for 3 Bragg curves and background
(use the “Particle mass reconstruction - version 5” as reference)

- Modify the version 4 to use a “dispatcher” to handle begin/end of analysis: take the new versions of `main` , `AnalysisInfo` and `AnalysisSteering` from the particle analysis example.
- Add a `class Constants` to contain the background mean and rms.
- Add a new function to `TotalEnergy` to compute energy after background subtraction, and use it to select and draw Bragg curves.
- In `EnergyDist` get the energy ranges from a text file with name specified in the command line.

Energies data plot - version 6

Plot mean and r.m.s. energies
for 3 Bragg curves and background
(use the “Particle mass reconstruction - version 6” as reference)

- Modify the version 5 to organize the code in packages:
  - create 4 packages: `AnalysisFramework`, `AnalysisPlugins`, `AnalysisObjects`, `AnalysisUtilities`,
  - move all source files, including `main.cc`, into those packages, avoiding circular dependencies,
  - compile each package into a library but `AnalysisPlugins`, where each analyzer is to be compiled to a distinct library.



- Produce the executable by using a dummy source code.