Particle physics tutorial

Luis Manzanillas – SOLEIL synchrotron

EPIC2- 12/10/2022

Outlook

- Introduction to Julia programming language
- Basics of detectors for particle physics
- Basics of HPGe detectors
- Hands on training – How to reconstruct the energy of photons
  - Reading h5 files containing raw data from a HPGe detector
  - Plotting and understanding the data
  - Finding events in raw data
  - Filters and calibration
  - Energy reconstruction and selection of events
  - Histograms and interpretation of results

- Why Julia?
- Science needs code - but how to write it?
  - Choice of programming language(s) matter!

  - Need to balance:
    - Learning time
    - Productivity
    - Performance
    - Usually involves compromises

- Programming Language Options
  - C++:
    - Pro: Very fast (in expert hands)
    - Pro: Really cool new concepts (even literally) in C++11/14/17/...
    - Con: Complex, takes long time to learn and much longer to master
    - Con: Straightforward tasks often result in lengthy code
    - Con: No memory management (General protection faults)
    - Con: No universal package management
    - Con: Composability isn't great

- Programming Language Options
  - Python:
    - Pro: Broad user base, popular first programming language
    - Pro: Easy to learn, good standard library
    - Con: Can't write time-critical loops in Python,
      - workarounds like Numba/Cython have many limitations, don't compose well
    - Con: Language itself fairly primitive, not very expressive
    - Con: Duck-Typing necessitates lots of test code
    - Con: No effective multi-threading
    - Con: Composability isn't great

- The 97 and the 3 Percent
  - We should forget about small efficiencies, say about 97% of the time: premature optimization is the root of all evil. Yet we should not pass up our opportunities in that critical 3%.
    - Donald E. Knuth

- Some programming languages (e.g. Python) great for the 97% - but can't make the 3% fast.
- Some other languages (e.g. C/C++, Fortran) can handle the 3% - but makes the 97% complicated.

- The Two-language Problem
- Common approach nowadays:
  - Write time critical code in C/C++, rest in Python
  - Pro: End-user can code comfortably in Python, with good performance
  - Con: Complexity of C/C++ plus complexity of Python
  - Con: Need proficiency in two languages, barrier that prevents non-expert users from contributing to important parts of code
  - Con: Limits generic implementation of algorithms
  - Con: Severely limits metaprogramming, automatic differentiation, etc.

Julia

- Designed for scientific/technical computing
- Originated at MIT, first public version 2012
- Covers the whole wish-list
- Clear focus on user productivity and software quality
- Rapid growth of user base and software packages
- Current version: Julia v1.7
- Julia Language Properties
  - Fast: compilation to native CPU and GPU code
  - Multiple-dispatch (more powerful than object-oriented):
  - Solves the expression problem
  - Dynamically typed
  - Very powerful type system, types are first-class values
  - Functional programming and metaprogramming
  - First-class math support (like Fortran or Matlab)
  - Local and distributed code execution
  - State-of-the-art multi-threading: parallel code
  - Can call parallel code that can call parallel code, ...,
  - Without oversubscribing threads
  - Software package management:
  - Trivial to create and install packages
  - Excellent REPL (console)
  - Easy to call Fortran, C/C++ and Python code

The physics of particle detectors

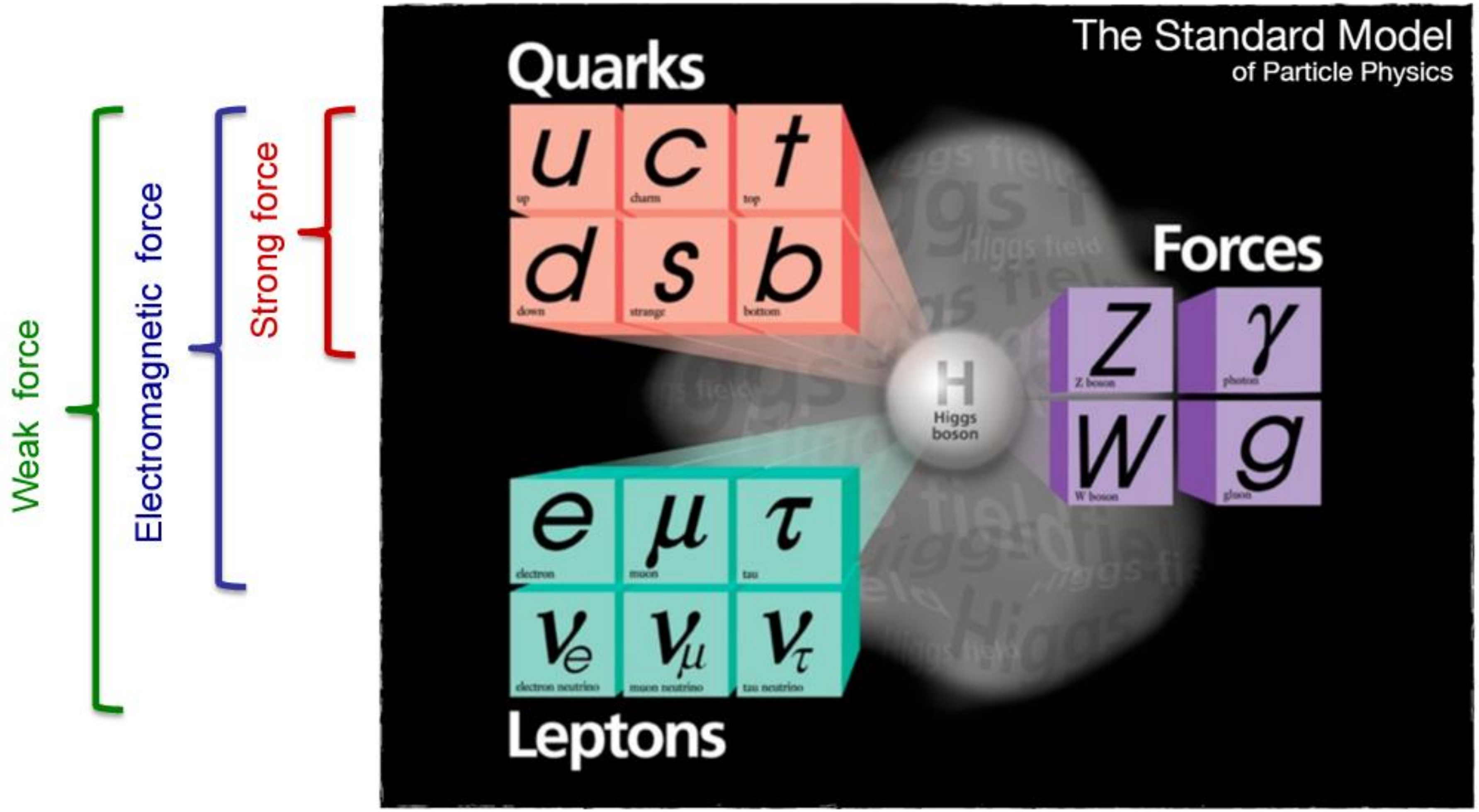
What is a **Particle**?

What is a **Detector**?

How to detect a particle?

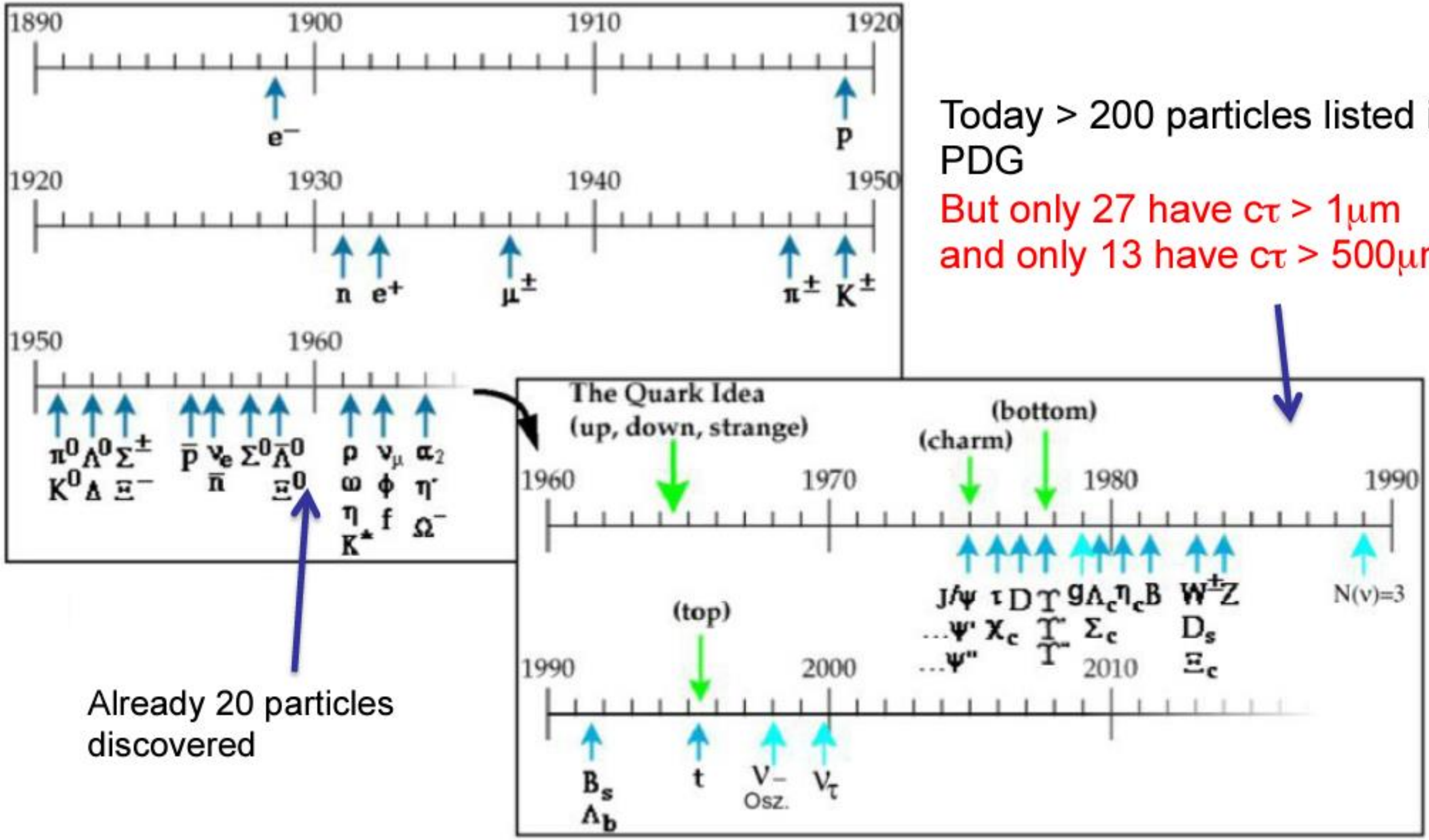
This tutorial

What is a particle



***More details in particle physics related talks of this school**

A history of discoveries



Today > 200 particles listed in PDG

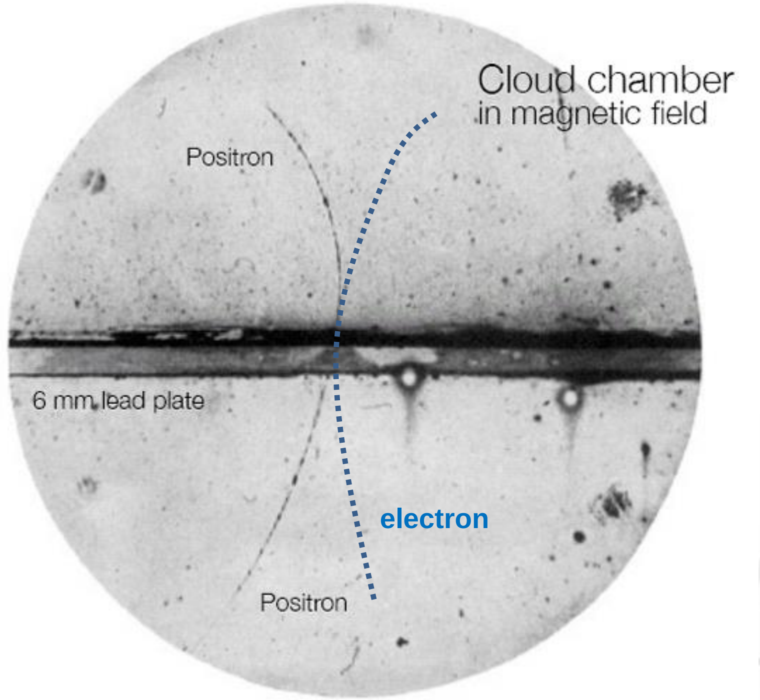
But only 27 have cτ > 1μm and only 13 have cτ > 500μm

Already 20 particles discovered

Example of discovery

- Positron discovery
  - Carl Andersen 1933 [**Nobel prize 1936**]

- Detector setup
  - Magnetic field 15000 Gauss,
  - chamber diameter 15cm.
  - A 63 MeV positron passes through a 6mm lead leaving the plate with energy 23MeV.
  - The ionization of the particle, and its behavior in passing through the foil is the same as those of an electron.



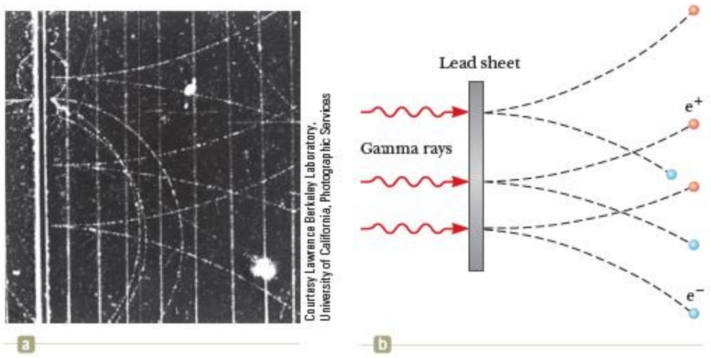
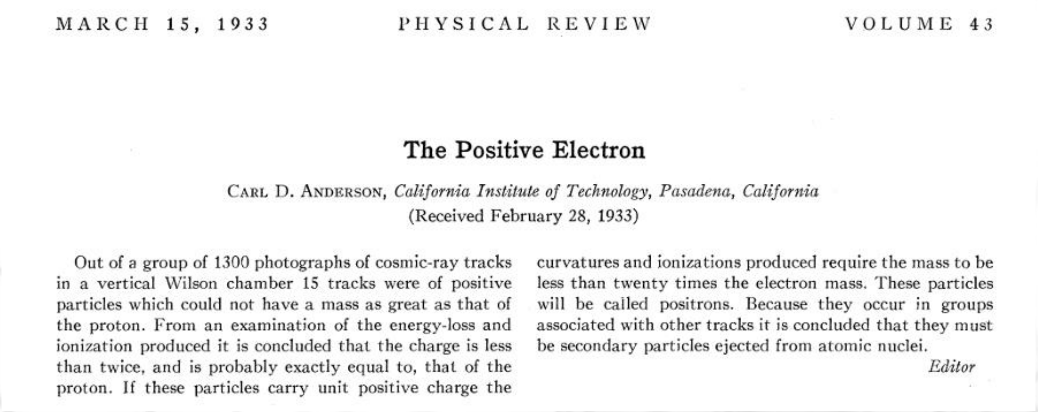


Figure 44.2 (a) Bubble-chamber tracks of electron–positron pairs produced by 300-MeV gamma rays striking a lead sheet from the left. (b) The pertinent pair-production events. The positrons deflect upward and the electrons downward in an applied magnetic field.



MARCH 15, 1933 PHYSICAL REVIEW VOLUME 43

The Positive Electron

CARL D. ANDERSON, *California Institute of Technology, Pasadena, California*
(Received February 28, 1933)

Out of a group of 1300 photographs of cosmic-ray tracks in a vertical Wilson chamber 15 tracks were of positive particles which could not have a mass as great as that of the proton. From an examination of the energy-loss and ionization produced it is concluded that the charge is less than twice, and is probably exactly equal to, that of the proton. If these particles carry unit positive charge the curvatures and ionizations produced require the mass to be less than twenty times the electron mass. These particles will be called positrons. Because they occur in groups associated with other tracks it is concluded that they must be secondary particles ejected from atomic nuclei.

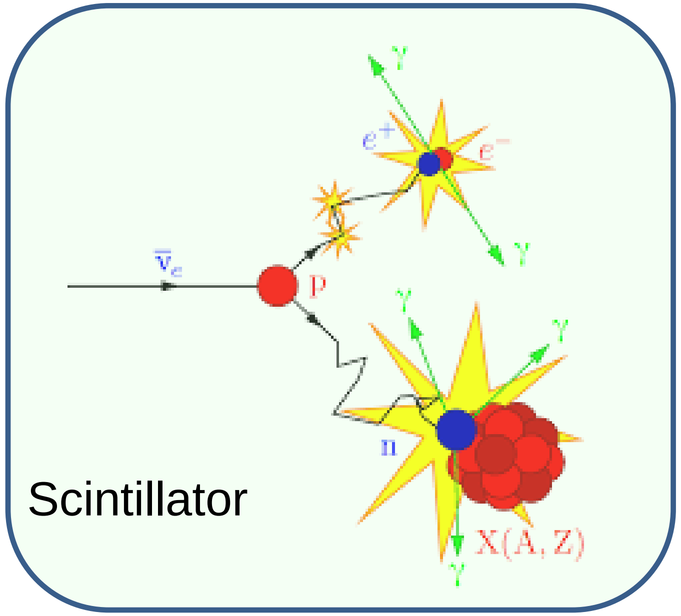
Editor

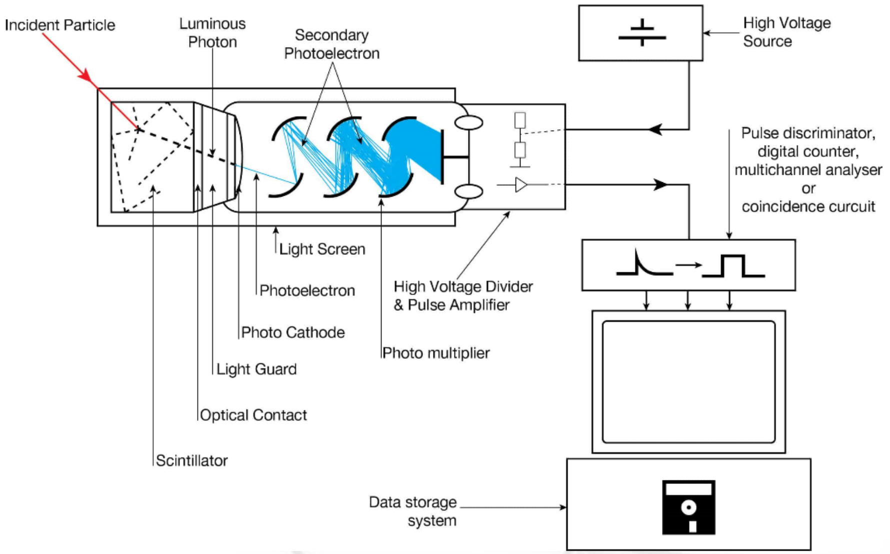
What is a detector

- In experimental physics, a **particle detector** or radiation detector is an **instrument used to detect, to track and to identify elementary particles** by measuring one or more properties of them
- **Particle detectors** are devices producing an **observable signal** when they are crossed by a particle. Usually they are made by an active element (such that there is some interaction with the particle) and by a readout system ("forming" the signal and sending it to the data acquisition chain)
  - The basic operation principle of ALL the detectors is to convert the energy lost in the active part in a "concrete signal" that can be "measured" (current, voltage, light, heat,...).
- "Universal" detectors, sensitive to all the particles over all the range of their "properties" **DO NOT exist**

Particle detection

- The detector sees only "stable" particles (cτ > 500μm)
- The 8 most frequently produced are:
  - e±,μ±,γ,π±,K±,K0,p±,n
- In order to detect a particle, it has to **interact - and deposit energy**
- Ultimately, the **signals are obtained from the interactions of charged particles**
- Neutral particles (photons, neutrons) have to transfer their energy to charged particles to be measured
  - Example: Inverse beta decay detection in scintillators

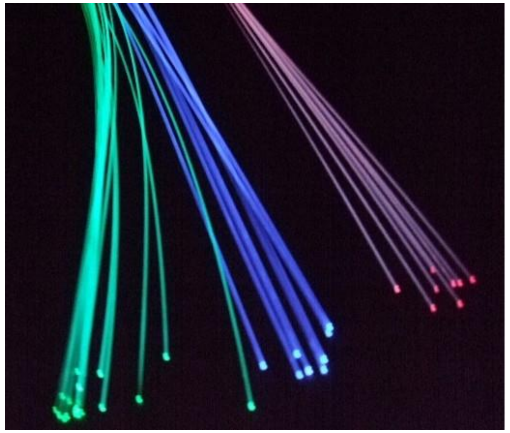


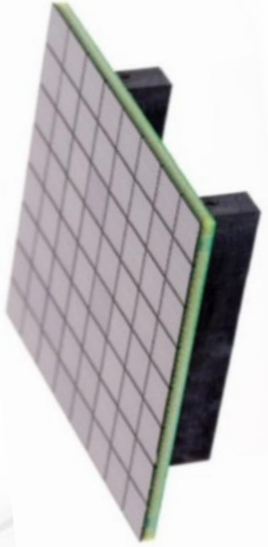


Few examples of detectors

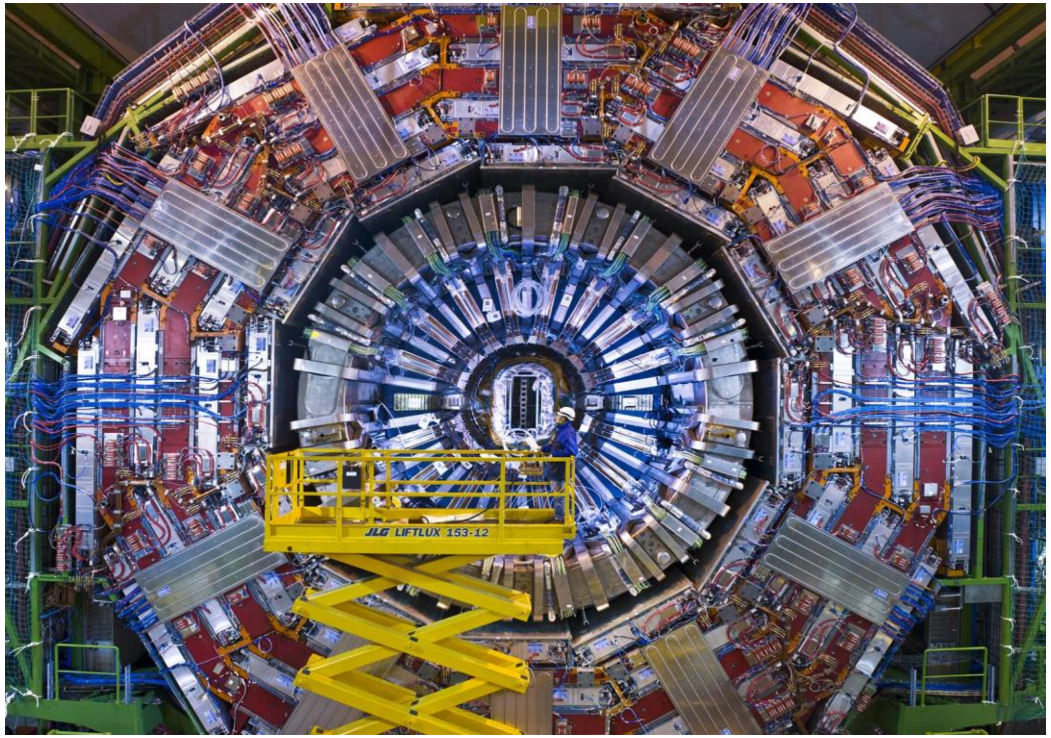


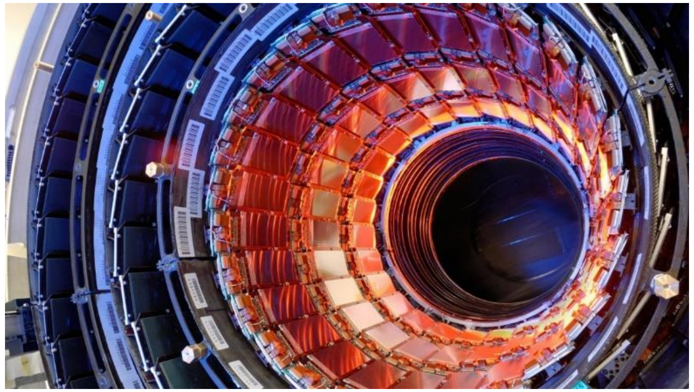


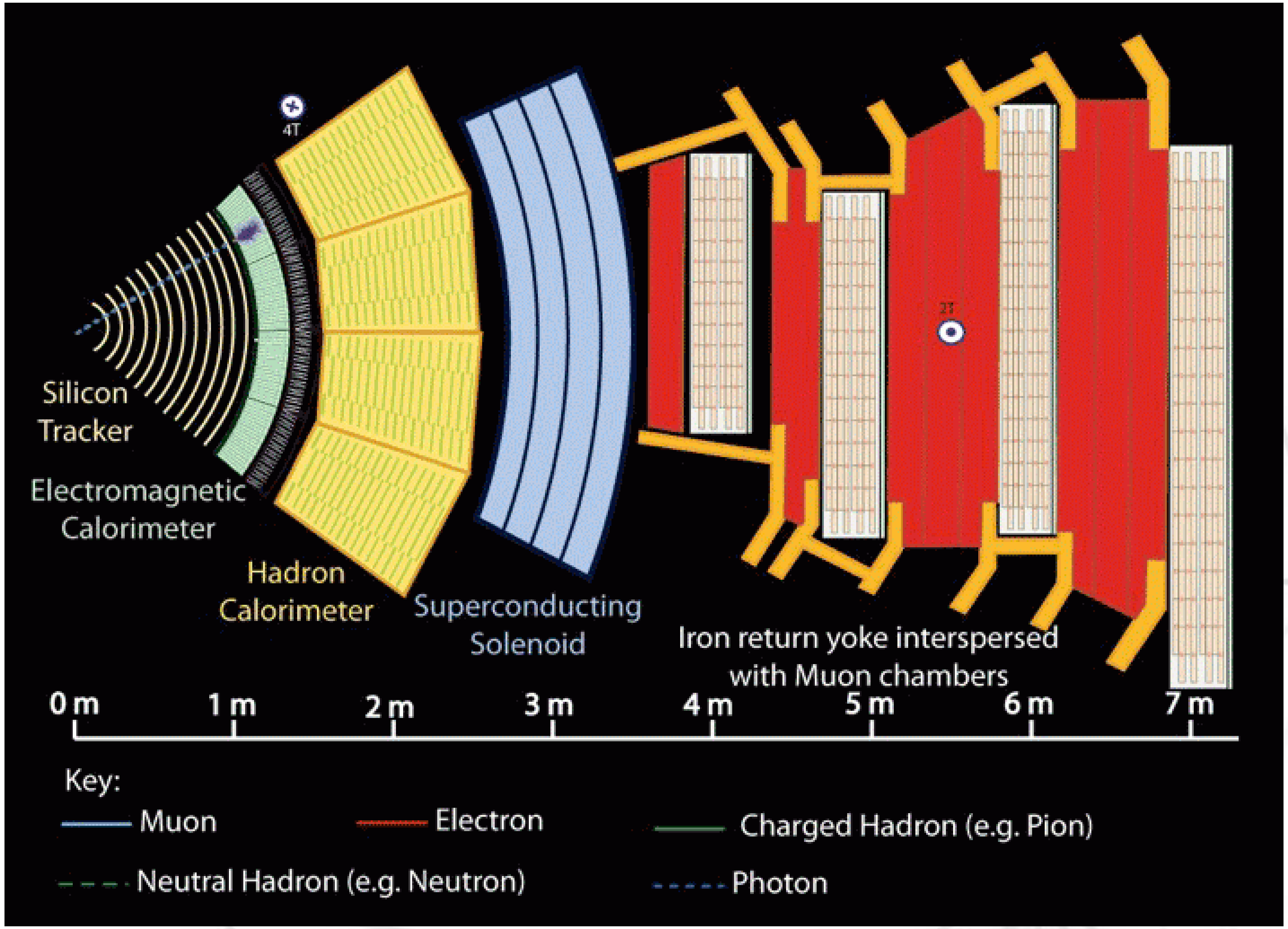




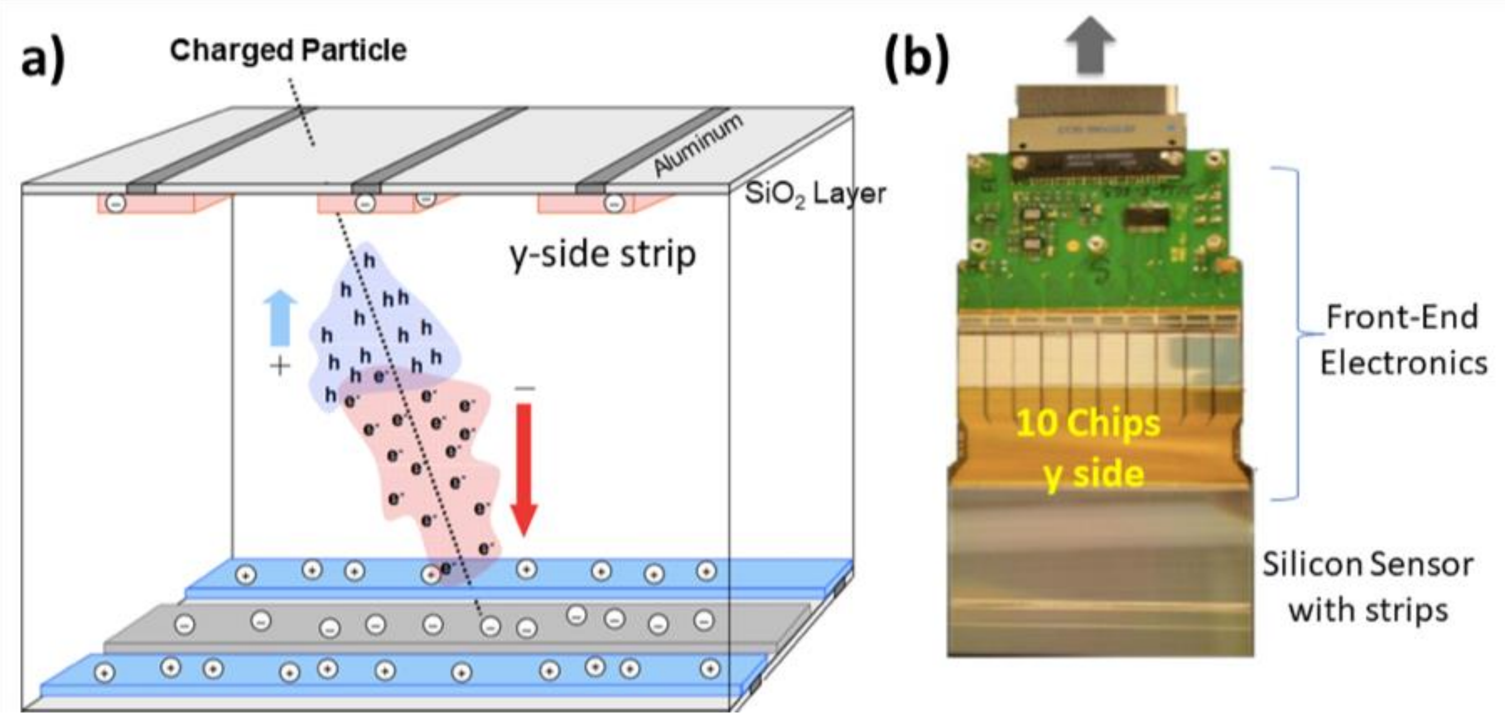
Complex detectors (CMS example)

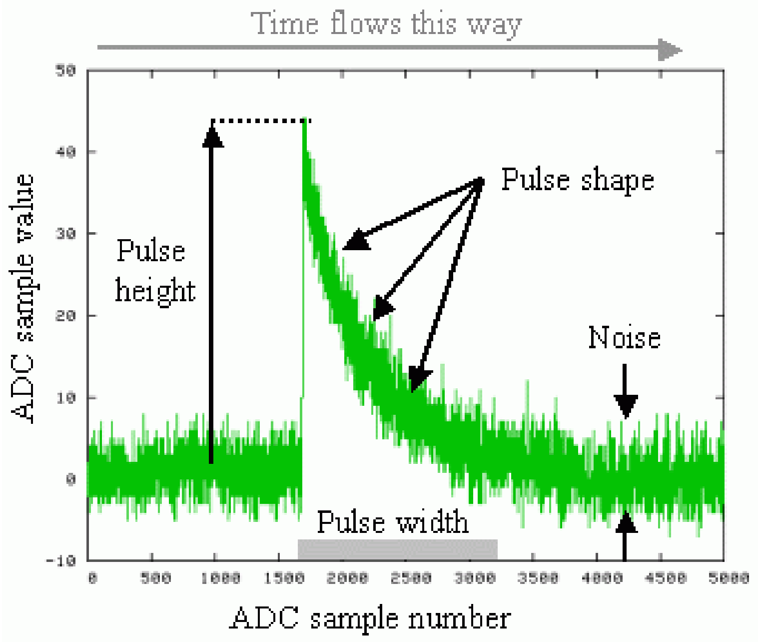






Signals in semiconductors

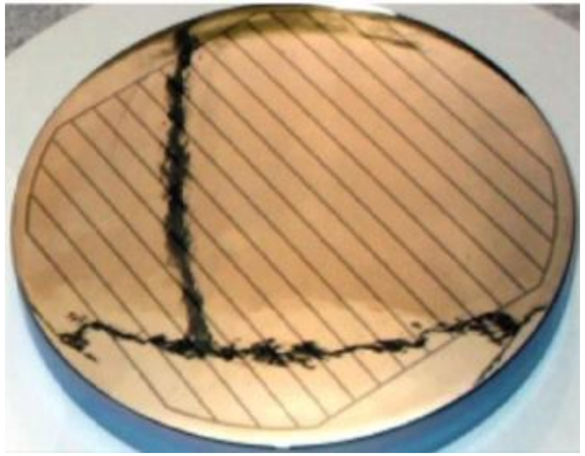




- Voltage applied to one side of pixel produces electric field
  - Electric charges will drift to the electrodes
- Charged particles produce electron-hole pairs that are drifted to the electrodes originating an electric signal

Types of HPGe detectors

Coaxial

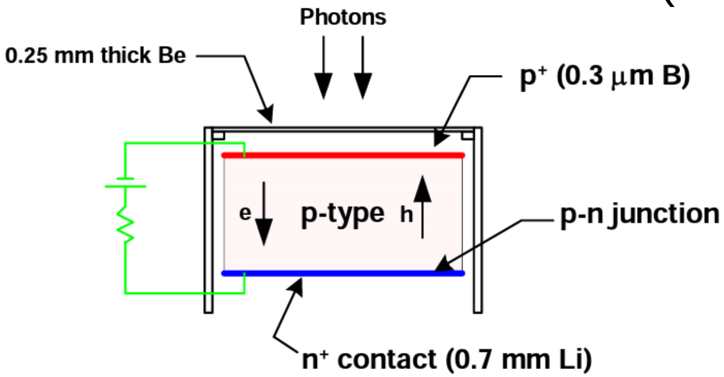


Planar

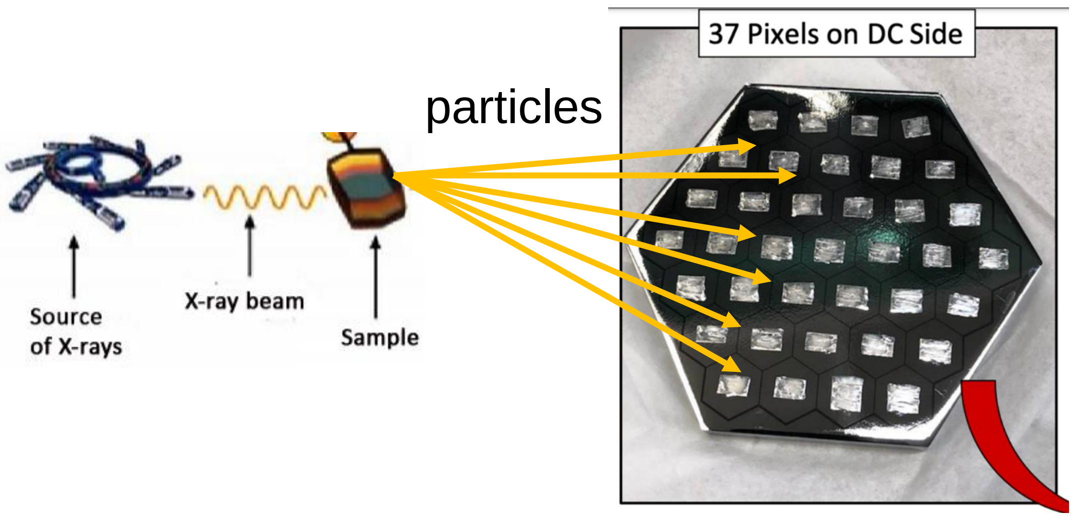
Inverted coaxial point contact (ICPC)

Segmented (pixelized)

Ge detector: Like a parallel plate capacitor made from a single crystalline slab of Ge

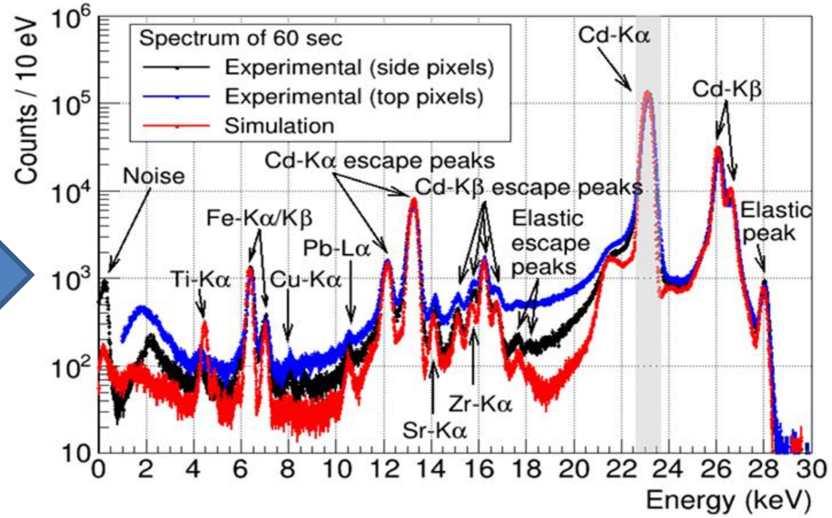


HPGe detection chain



Pixelized Ge detector

From interaction of particles with HPGe to an energy spectrum



- We need to understand the whole process from interaction of particles with matter to the detector response to obtain information about position, energy, momentum, etc
- HPGe detectors → outstanding energy resolution
  - Goal: **Obtain the best possible energy resolution with the lowest background**
    - R&D for **optimization** of detectors

The HPGe signals

Detector geometry
Impurity density
Bias voltage

→

Electric potential
Electric field
Charge drift model

→

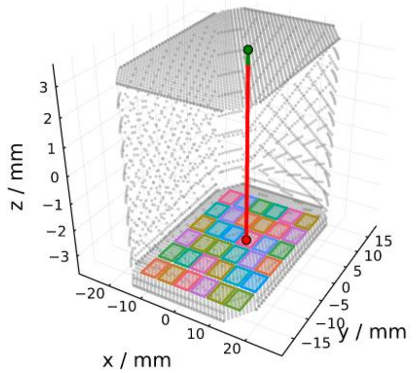
Drift paths
Weighting potentials

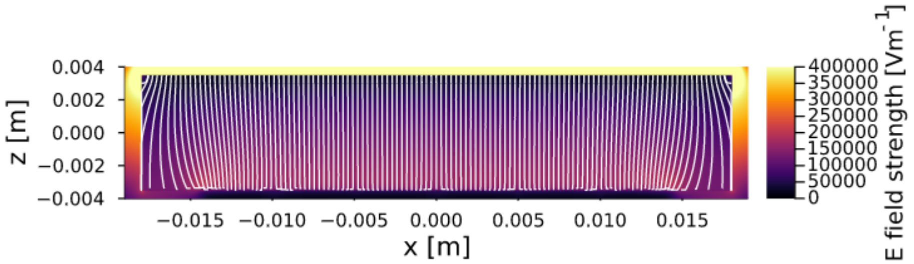
→

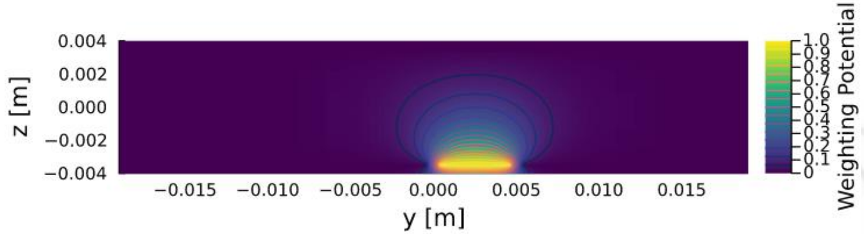
particle interaction
Perfect pulses

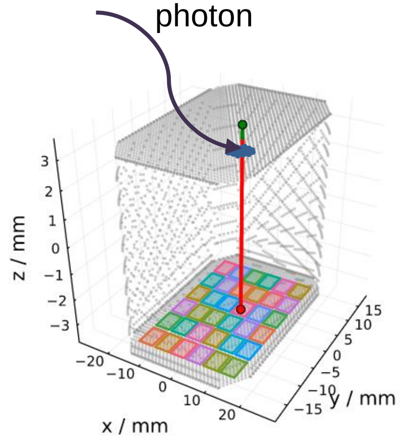
→

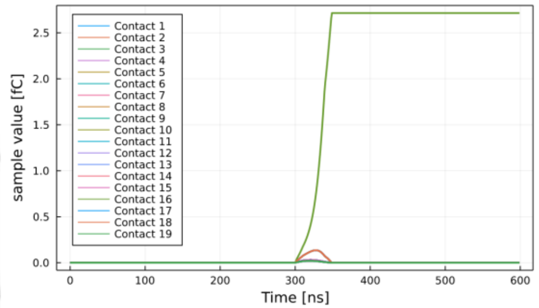
Signal amplification
Electronics noise
Real-like waveforms

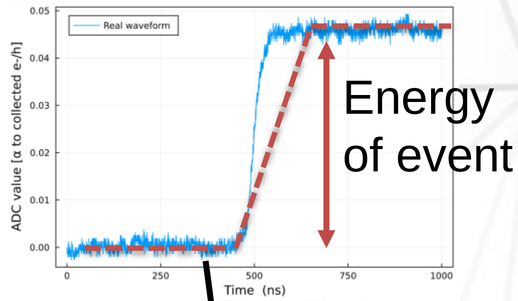


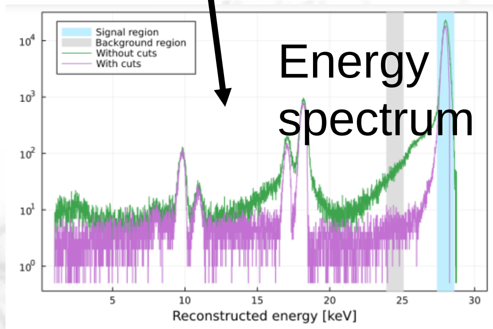




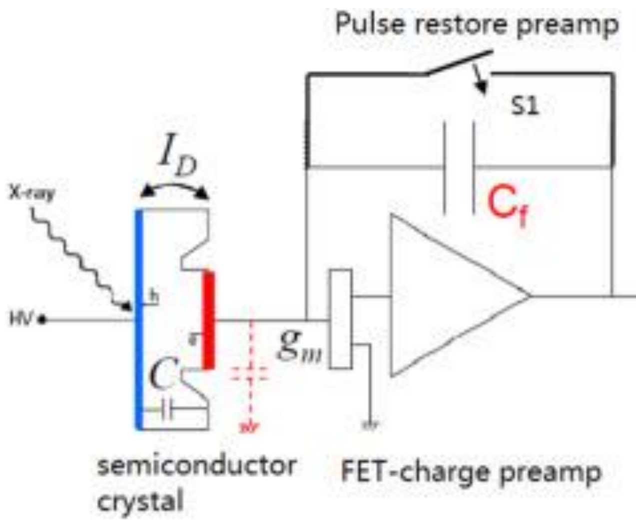


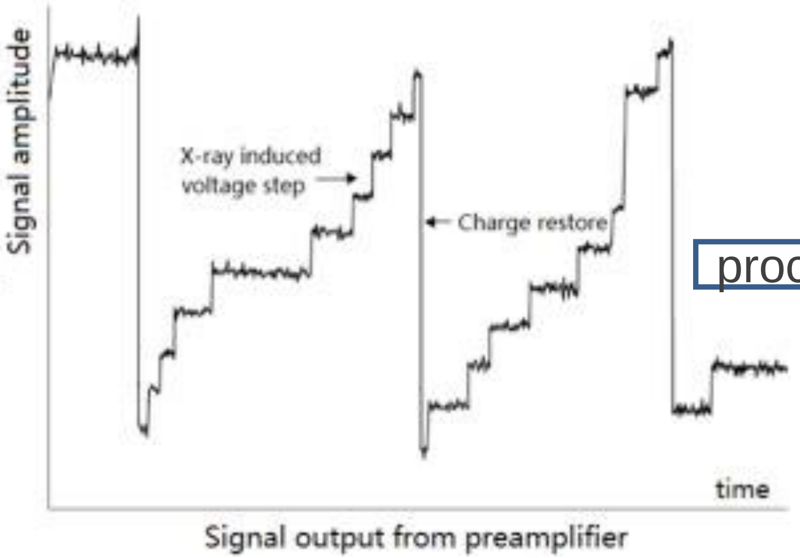




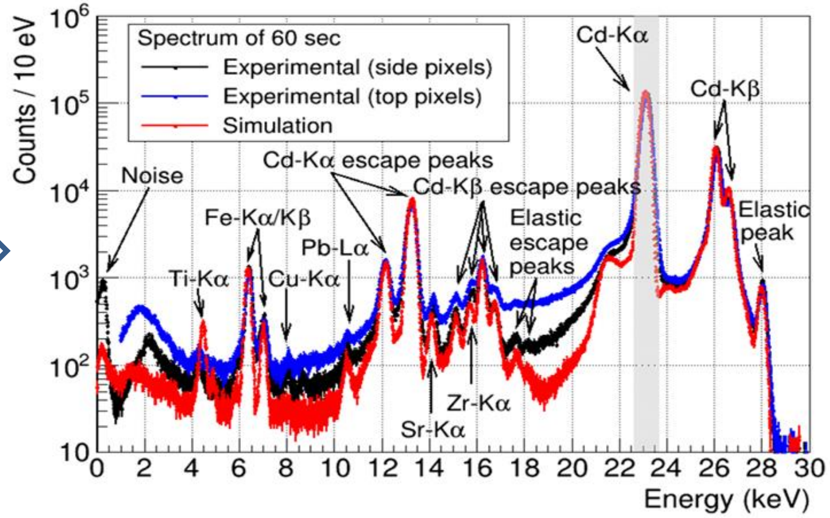


- The raw data consists of waveforms containing steps with amplitude proportional to the energy of the incident particle





processing



Hands on training

- Questions before hands on training?

- Work on jupyter notebook

Thanks for your attention!

Questions?