Monte Carlo software packages for Medical Physics

A brief (and incomplete) overview

Alexandre Lindote

What is a MC software package?

- ☐ More specifically, a MC package for radiation interaction tracking — which typically can also be used for Medical Physics
- ☐ You now know how to sample a given distribution (either using the inverse of the cumulative function or the rejection method)
- ☐ You also know how to use that to simulate a simple physics process (e.g. Compton scattering)
- ☐ A MC package tries to make our life easier, by offering:
  - ☐ a set of algorithms to simulate multiple (and simultaneous) physics processes (may be limited to a given set of physics, e.g. electromagnetic, or more general)
  - ☐ a philosophy to set up the geometry of an experiment (materials and shapes)
  - ☐ an algorithm for tracking particles inside the set-up (may include support for fields)

Requirements

What we want from a MC software package

- ☐ Reliability of the results
- ☐ Versatility and completeness
- ☐ Computational speed
- ☐ Ease of use (not mandatory, but nice to have)
- ☐ Open source code (not mandatory, but nice to have)

Requirements

What we want from a MC software package

- ☐ **Reliability of the results**
  - extensive and continuous comparison with published data

Requirements

What we want from a MC software package

- ☐ **Reliability of the results**
  - extensive and continuous comparison with published data
- ☐ **Versatility and completeness**
  - ideally it should be able to simulate a wide range of physics processes (e.g. electromagnetic, hadronic, optical, etc.)

Requirements

What we want from a MC software package

- ☐ **Reliability of the results**
  - extensive and continuous comparison with published data
- ☐ **Versatility and completeness**
  - ideally it should be able to simulate a wide range of physics processes (e.g. electromagnetic, hadronic, optical, etc.)
- ☐ **Computational speed**
  - we need to be able to quickly plan a treatment, or to simulate a large amount of events for study and development of new techniques

Requirements

What we want from a MC software package

- ☐ **Reliability of the results**
  - extensive and continuous comparison with published data
- ☐ **Versatility and completeness**
  - ideally it should be able to simulate a wide range of physics processes (e.g. electromagnetic, hadronic, optical, etc.)
- ☐ **Computational speed**
  - we need to be able to quickly plan a treatment, or to simulate a large amount of events for study and development of new techniques
- ☐ **Ease of use**
  - offer tools that help set up a new simulation and analyse the results

Requirements

What we want from a MC software package

- ☐ **Reliability of the results**
  - extensive and continuous comparison with published data
- ☐ **Versatility and completeness**
  - ideally it should be able to simulate a wide range of physics processes (e.g. electromagnetic, hadronic, optical, etc.)
- ☐ **Computational speed**
  - we need to be able to quickly plan a treatment, or to simulate a large amount of events for study and development of new techniques
- ☐ **Ease of use**
  - offer tools that help set up a new simulation and analyse the results
- ☐ **Open source code**
  - an open source code ensures more people can inspect the code, leading to quicker debugging, fixing problems, optimisation and benchmarking

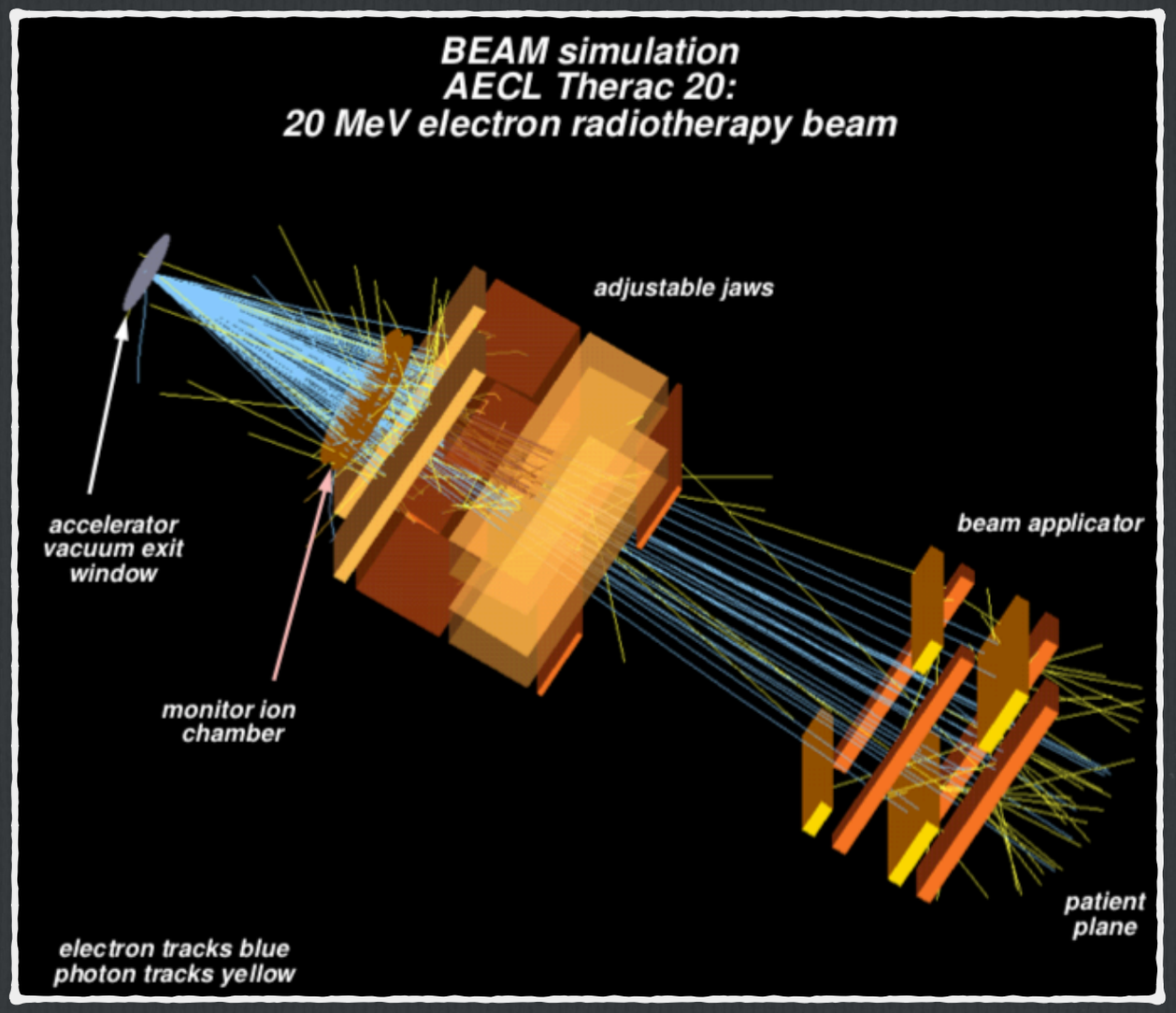
A non-comprehensive list…

- ☐ ETRAN (NIST 1978)
- ☐ SRIM (Ziegler and Biersack 1983)
  (www.srim.org — Windows only, recent-ish versions available)
- ☐ EGS4 (SLAC 1970s)
  (www.slac.stanford.edu/egs)
  - ☐ EGSnrc (NRCC 2000) — Canadian “branch”
    (www.irs.inms.nrc.ca/inms/irs/irs.html)
  - ☐ EGS5 (KEK-SLAC 2005) — Japanese “branch”
    (www.kek.jp/research/egs/egs5.html)
- ☐ Penelope (U. Barcelona 1999)
  (www.nea.fr/lists/penelope.html)

A non-comprehensive list…

- ☐ Fluka (CERN-INFN 2005) (www.fluka.org)
- ☐ Geant3 (CERN 1986) — superseded by GEANT4, but still used (www.cern.ch)
- ☐ Geant4 (CERN++ 1999) (geant4.web.cern.ch/geant4)
- ☐ MARS (FNAL) (www-ap.fnal.gov/MARS)
- ☐ MCNP (LANL 1970s) (mcnpx.lanl.gov)

EGS — Electron Gamma Shower

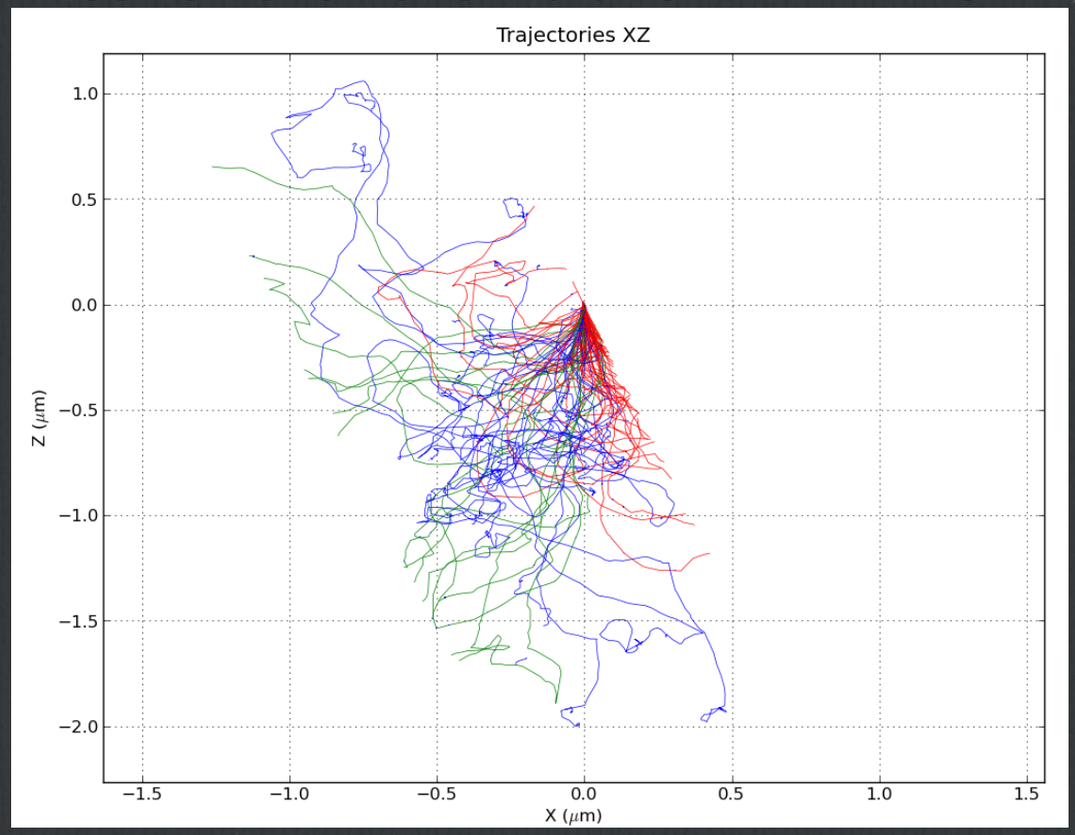


- For electrons, positrons and gammas (electromagnetic processes only)
- keV — GeV range
- Currently split in two releases:
  - EGS5 (maintained by KEK)
  - EGSnrc (maintained by the NRC of Canada)
- In particular, BEAMnrc is widely used for LINAC simulations (X-rays and high energy electrons for radiation therapy)
- Open source (NRC on GitHub), Fortran base but C++ interface to build new sims
- Online documentation (including tutorials)

Penelope

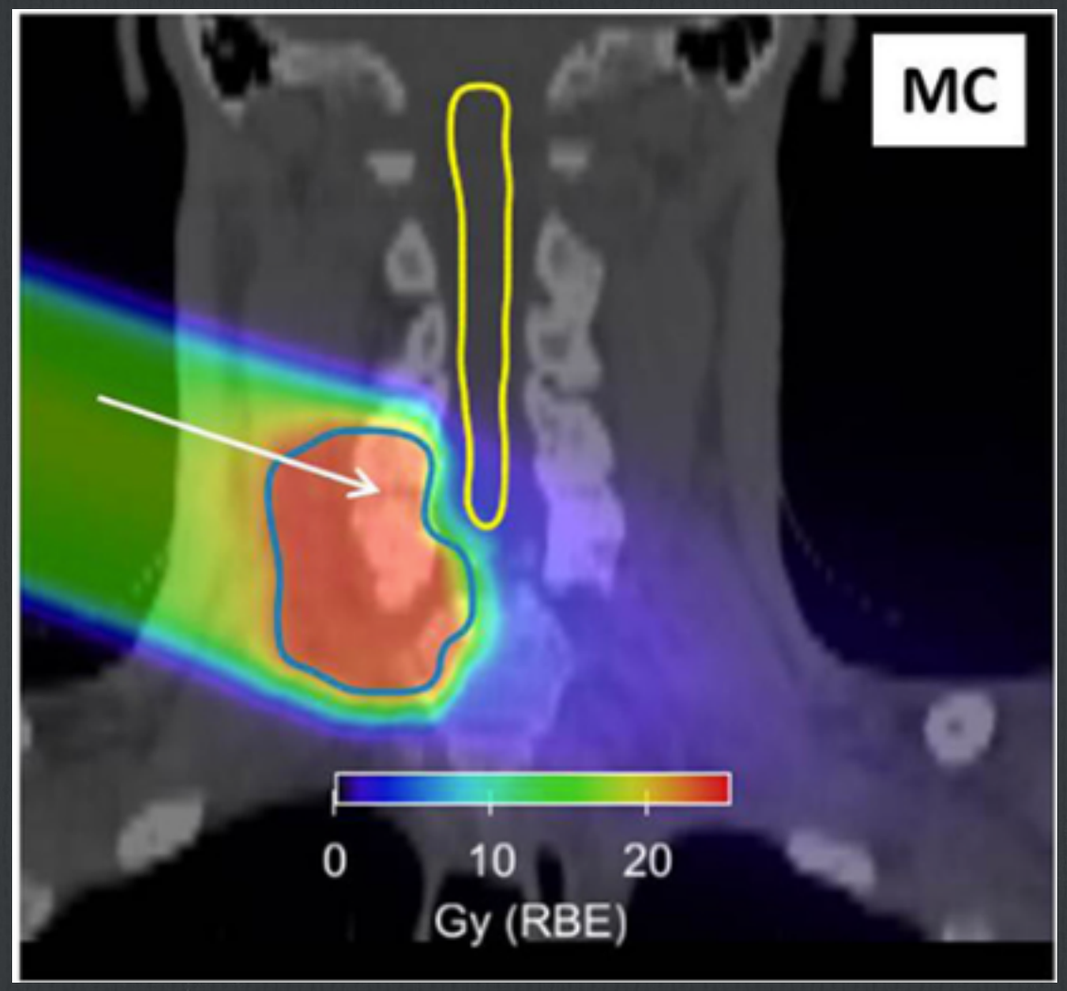
- Penetration and ENErgy LOss of Positrons and Electrons
- Developed in Fortran
- For tracking of electrons, positrons and photons (EM processes only)
- 50 eV — 1 GeV (but large uncertainties below 1 keV)
- Check out pyPENELOPE, a Python interface to the Fortran core code
- The Penelope code is only available upon request (must be authorised)

Electron shower simulation in PENELOPE



FLUKA

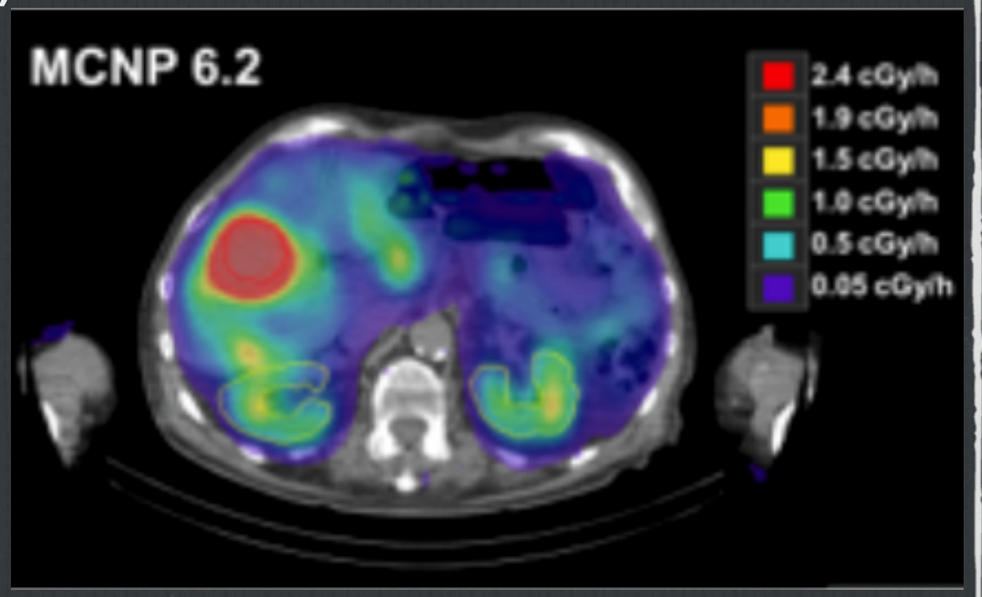
- General purpose code, includes EM and hadronic interactions
- Can also track optical photons
- Code base in Fortran
- User can create simple new applications without programming, using “card” files (sequences of commands following a specific format and alignment, have a look at subsection 2.2.13 here)
- New release ~1 month ago (http://www.fluka.org)



FLUKA simulation of the dose distribution for a Carbon beam therapy, superimposed with a CT planning image (see full article here)

MCNP

- Developed in the Los Alamos National Laboratory (where the Monte Carlo technique was invented in the 1940s)
- Another general purpose MC package, also in Fortran
- Applications can be developed through the use of input files with ‘cards’ (command sequences)
- It is known to be the ‘standard’ for nuclear reactions and neutron transport and interactions
- Used in safety codes in nuclear reactors (analyse the safety of a reactor and simulate possible accidents)
- Code maintained and controlled by LANL, requests must be approved and non-US citizens do not have access to the source code
  - distribution is governed by US laws and Department of Energy regulations



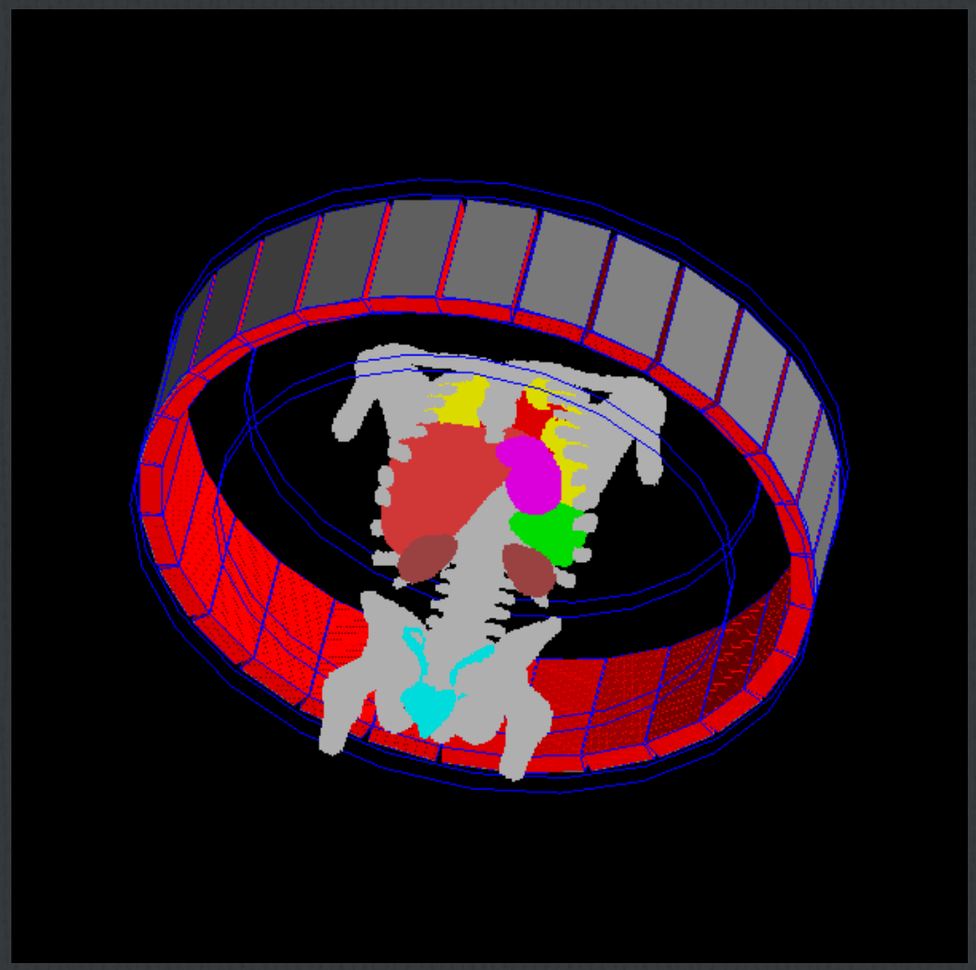
Example use of MCNP6.2 to estimate the dose distribution in the use of ^{177}Lu and ^{131}I sources, poster available here

GEANT4

- General purpose package, includes a large variety of physics processes (EM, hadronic, optical, nuclear)
- Developed in C++, using an Object Oriented Programming paradigm
- Open source code, very active development (new releases every 6 months)
- Maintained by a large, world-wide collaboration
- Often multiple models available for each process
  (e.g. straightforward to use a Penelope physics list for EM processes)
- The fact that it is open source and the very large user community ensure extensive benchmarking against experimental results and correction of errors
- Initially developed for HE particle physics (LHC experiments) but has found users in other areas and now has a large community of users in medical physics
- Includes multiple tools for visualisation and analysis
- Developing new simulations requires some programming knowledge

GATE

- Open source code targeted for Medical Physics
- Uses GEANT4 as backend, offering a user interface (macro files) for typical medical applications
- Currently supports PET and SPECT, CT, Optical Imaging (Bioluminescence and Fluorescence) and Radiotherapy
- Includes time varying geometries (e.g. breathing)
- We will explore it in one of the next classes!



TOPAS

- Another toolkit for medical physics built on top of GEANT4 — available here
- Particularly targeted for radiotherapy
- Allows modelling of the patient geometry from CT scans
- Also 4D, to include patient and source movement
- No programming knowledge required, set-up is done via macro files
- User registration required, free for educational and non-profit research purposes
  - New users must attend a mandatory introductory course