VII Seminario sul Software
per la Fisica Nucleare,
Subnucleare ed Applicata

University of Torino

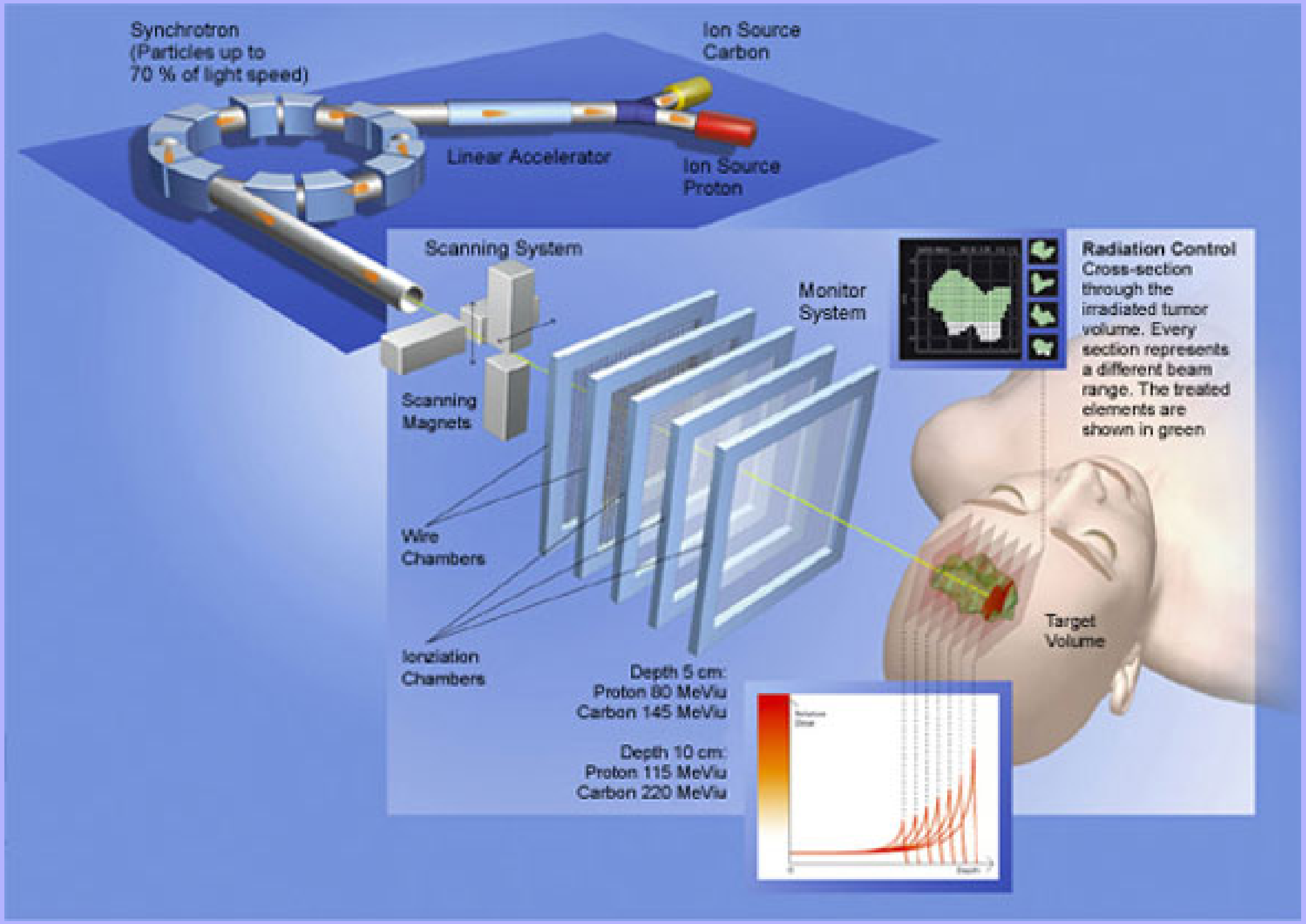
INFN-TPS project

The development of a carbon ion Treatment Planning System

Germano Russo

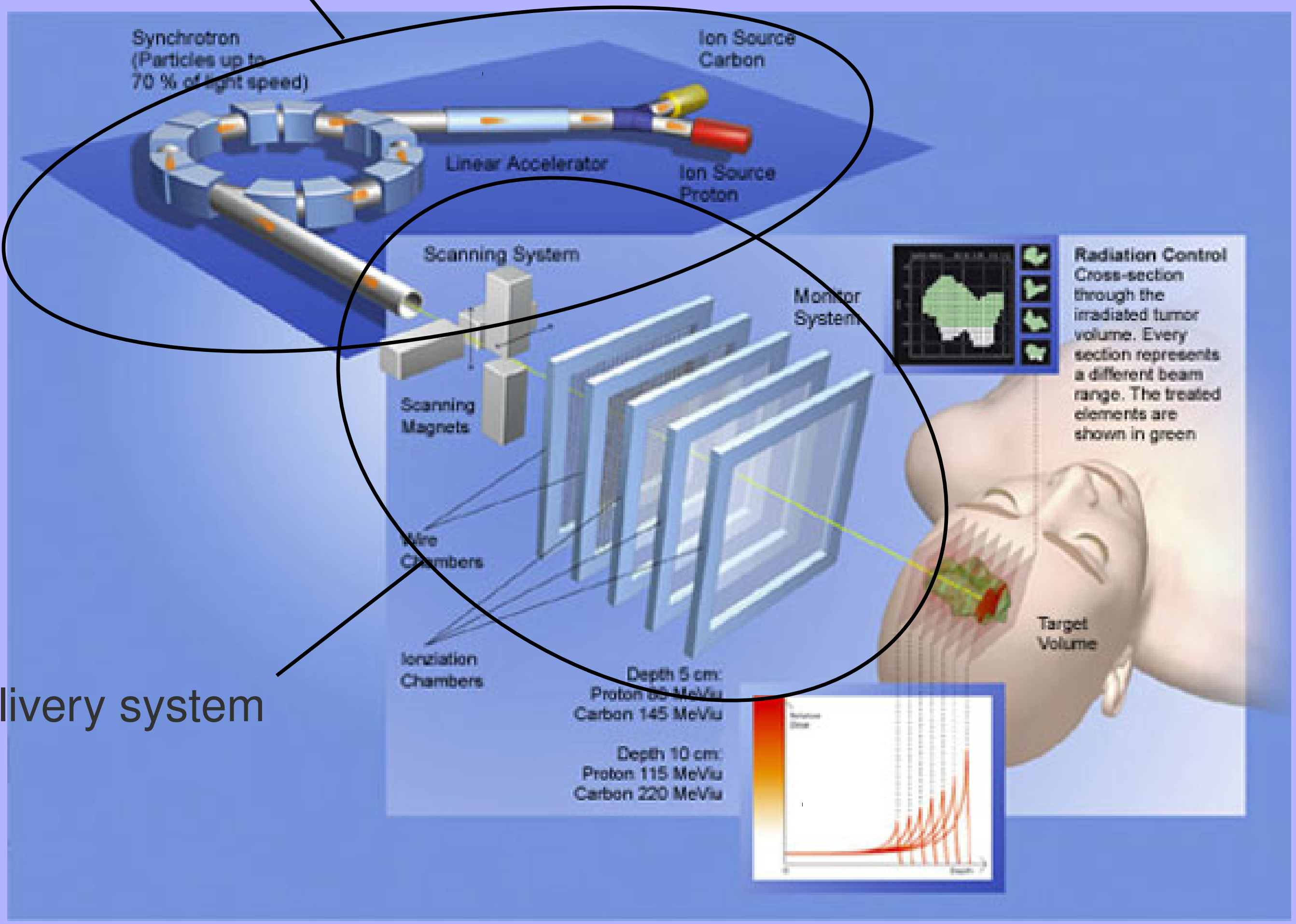
Alghero, 01/06/2010

The principles of ion therapy



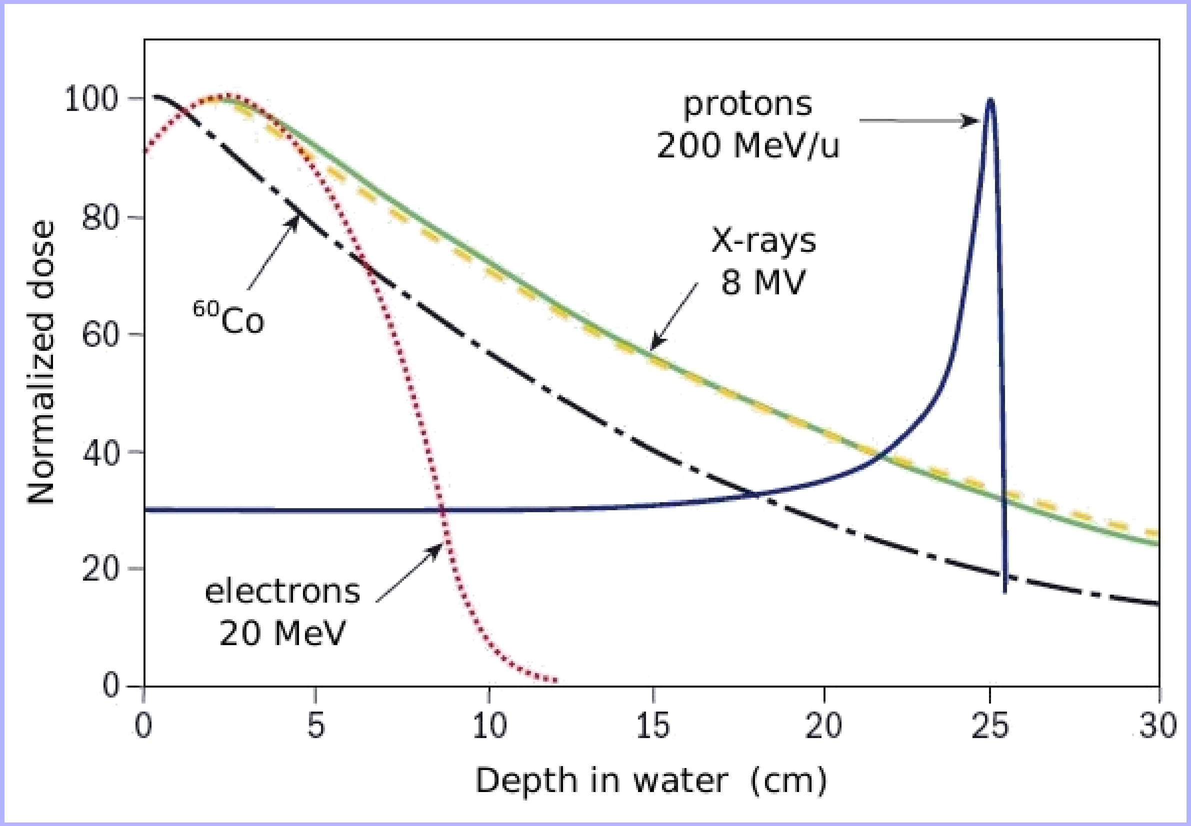
The principles of ion therapy

Accelerating system

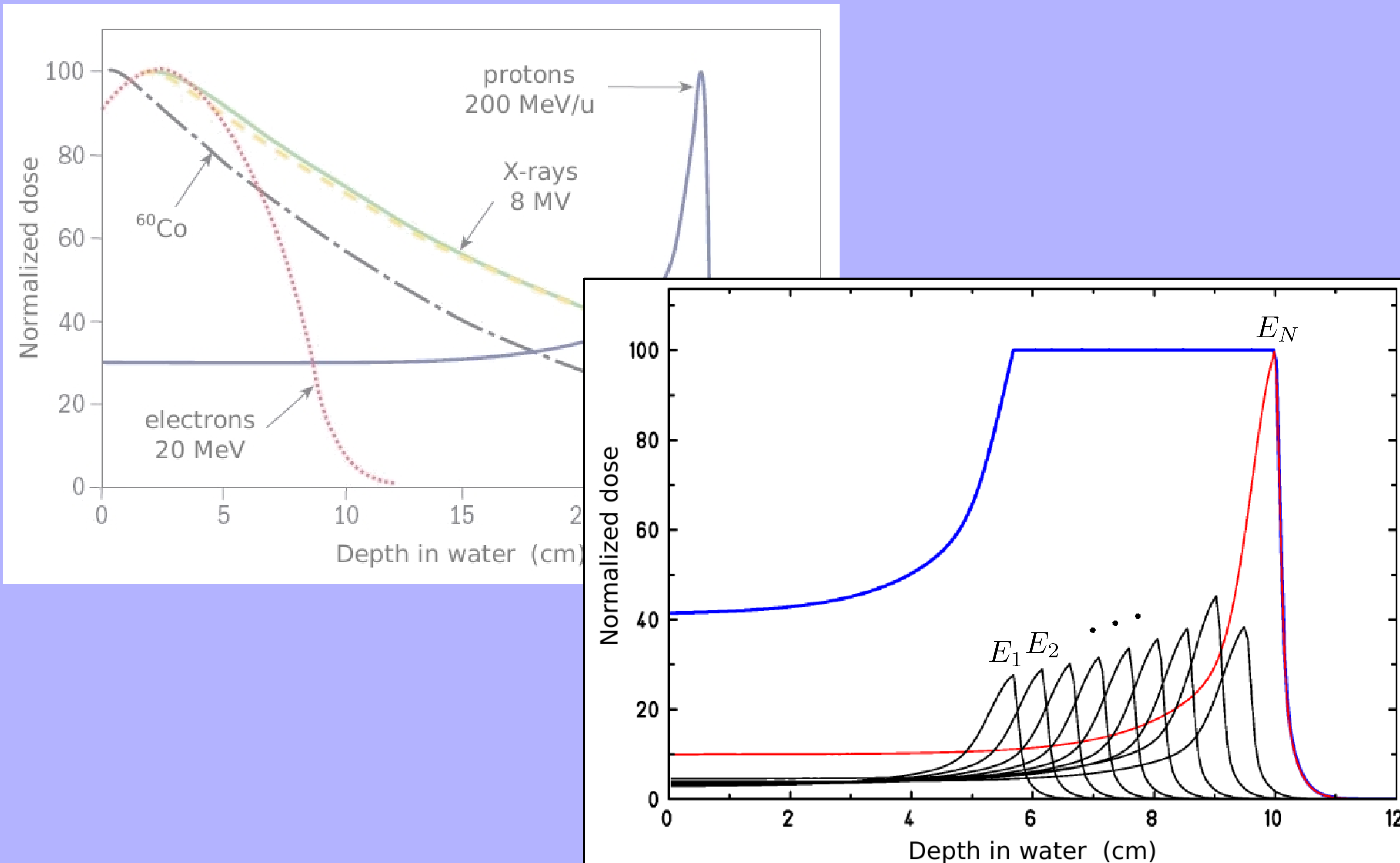


Delivery system

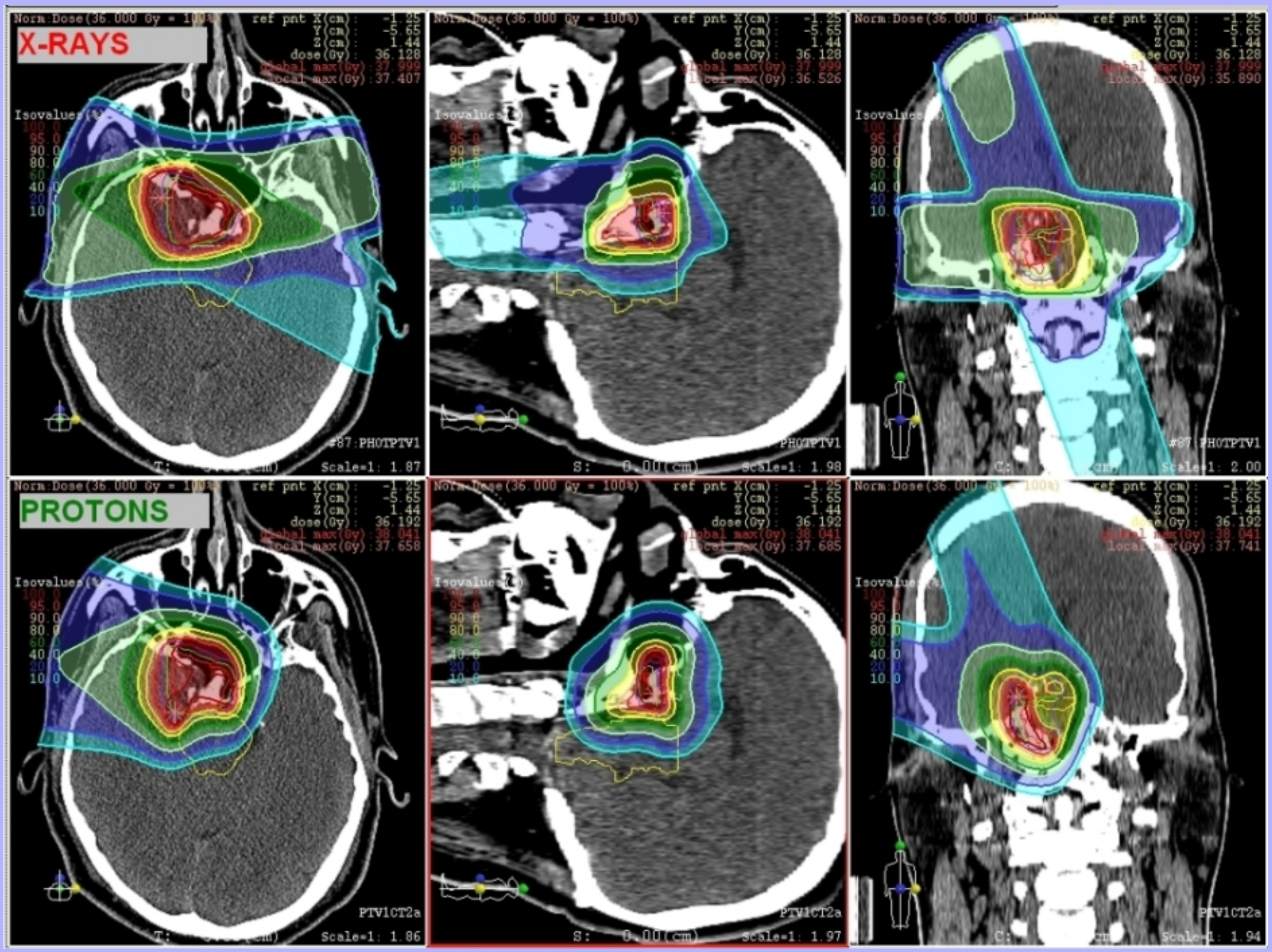
The principles of ion therapy | Bragg Peak



The principles of ion therapy | Spread Out Bragg Peak

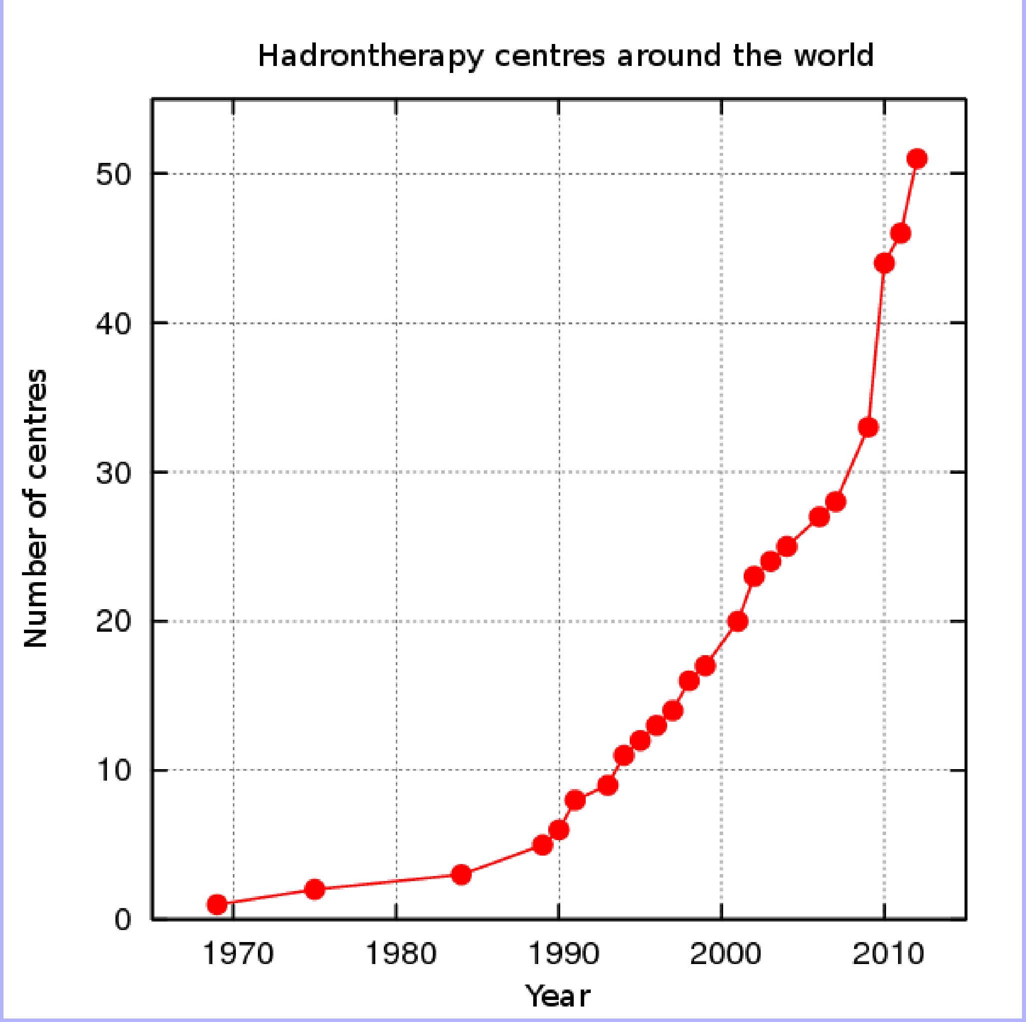


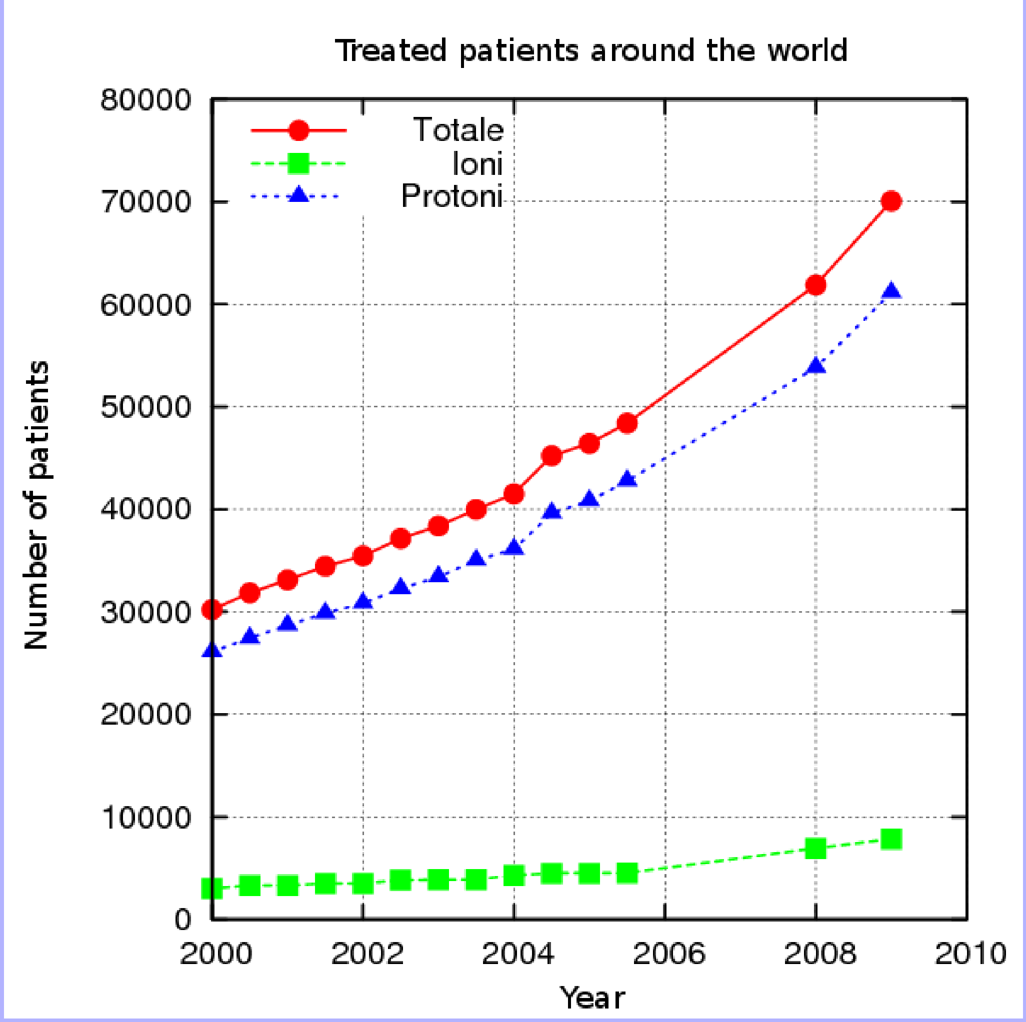
The principles of ion therapy | Dose conformity



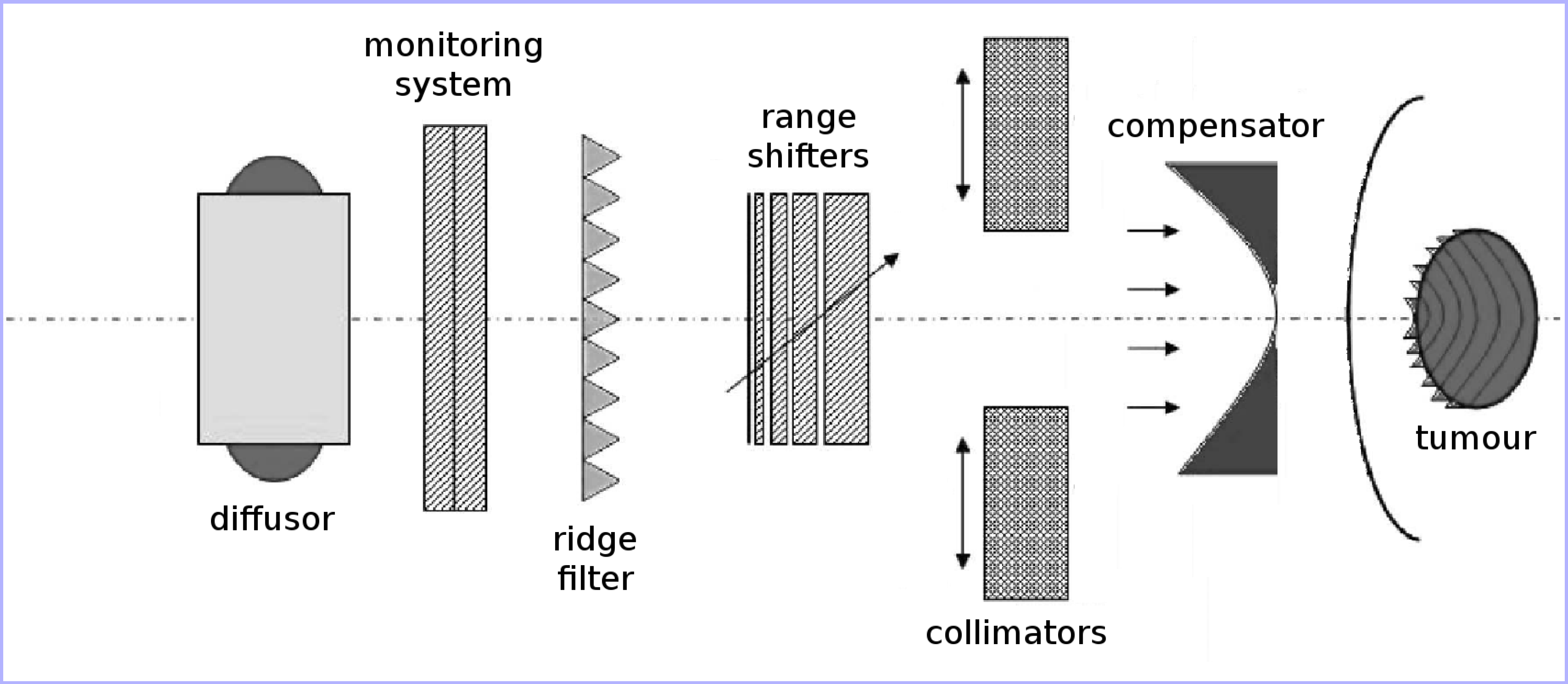
Courtesy of the Rinecker Proton Therapy Center, Germany

The principles of ion therapy | Spread





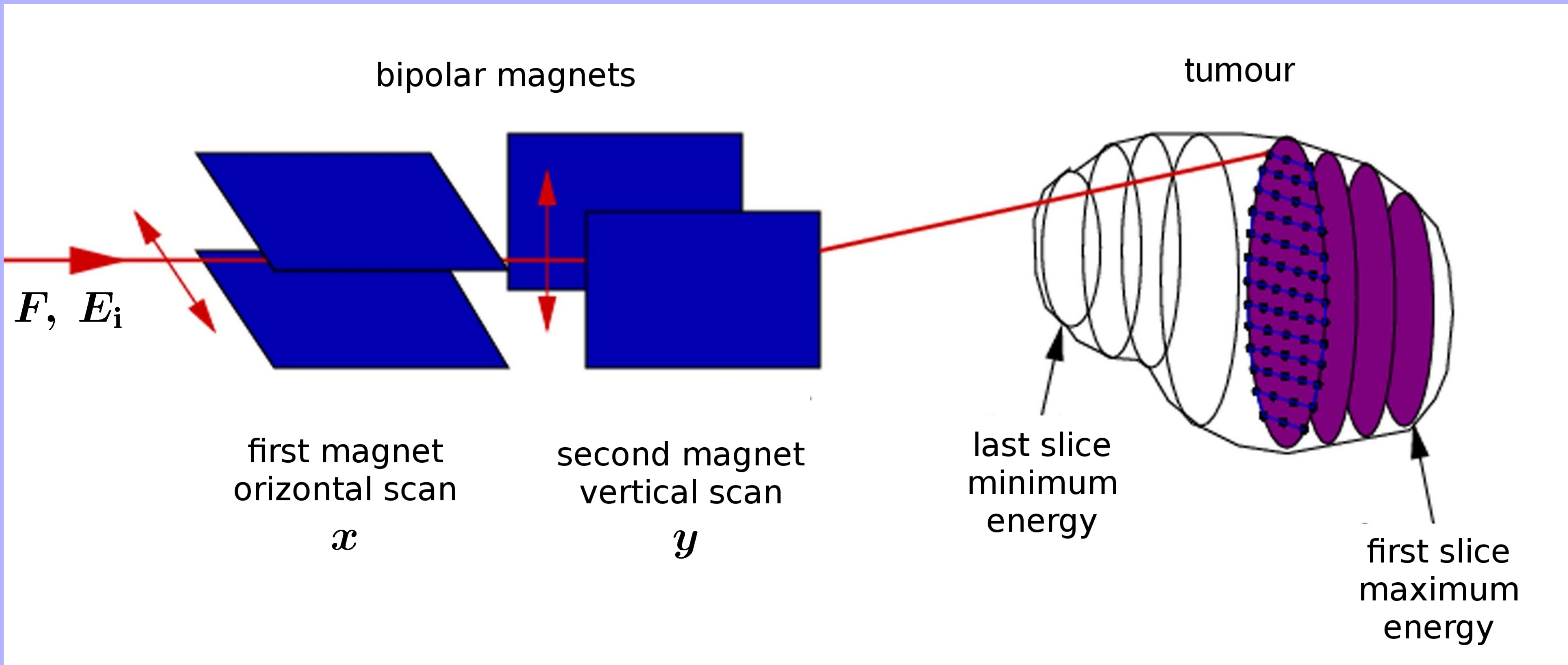
The principles of ion therapy | Passive delivery



HIBMC

HIMAC

The principles of ion therapy | Active scan



GSI

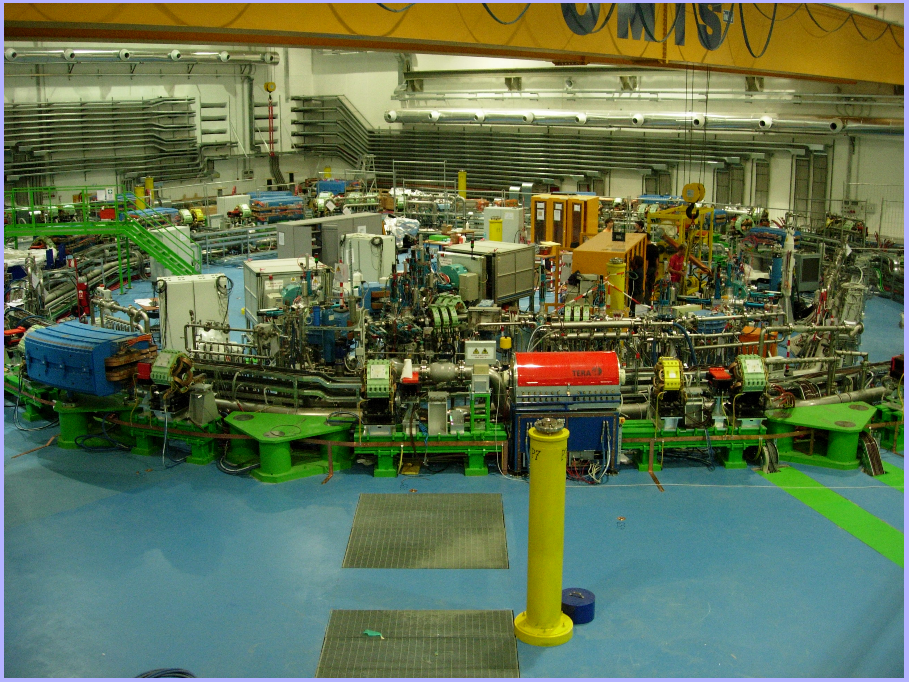
HIT
Heidelberger Ionenstrahl-Therapiezentrum

iba

CNAO

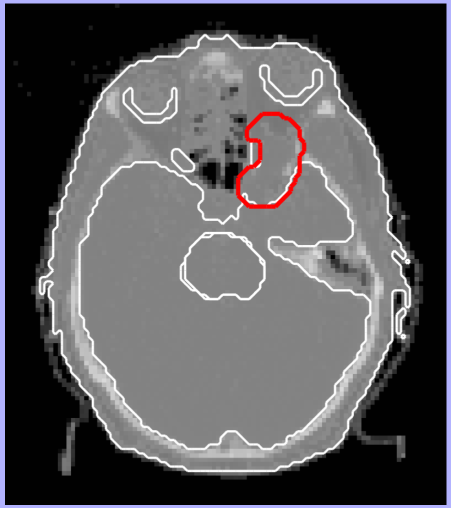
The principles of ion therapy | Active scan

Building of the Centro Nazionale di Adroterapia Oncologica (CNAO), Pavia, Italy – under construction



Accelerating ring of the CNAO

The principles of ion therapy | TPS

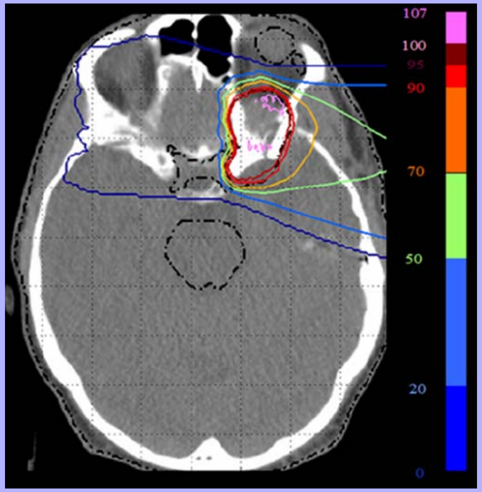


TPS

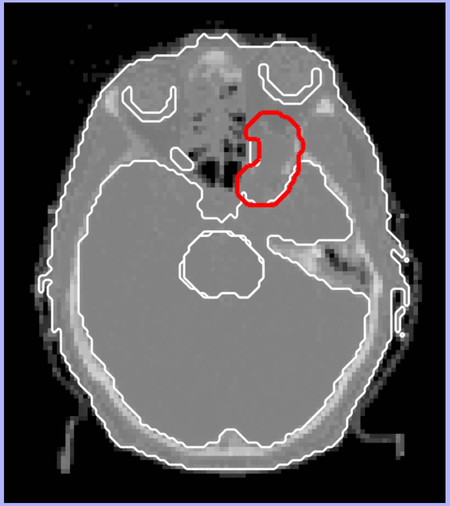
Physical Data (Geant4, Fluka)

Biological Data (LEM)

$$\{E_{\mathrm{i}}, F, x, y\}$$



The principles of ion therapy | TPS

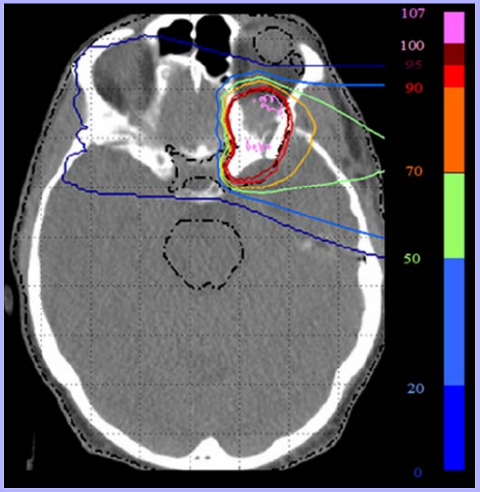


TPS

Physical Data
(Geant4, Fluka)

Biological Data
(LEM)

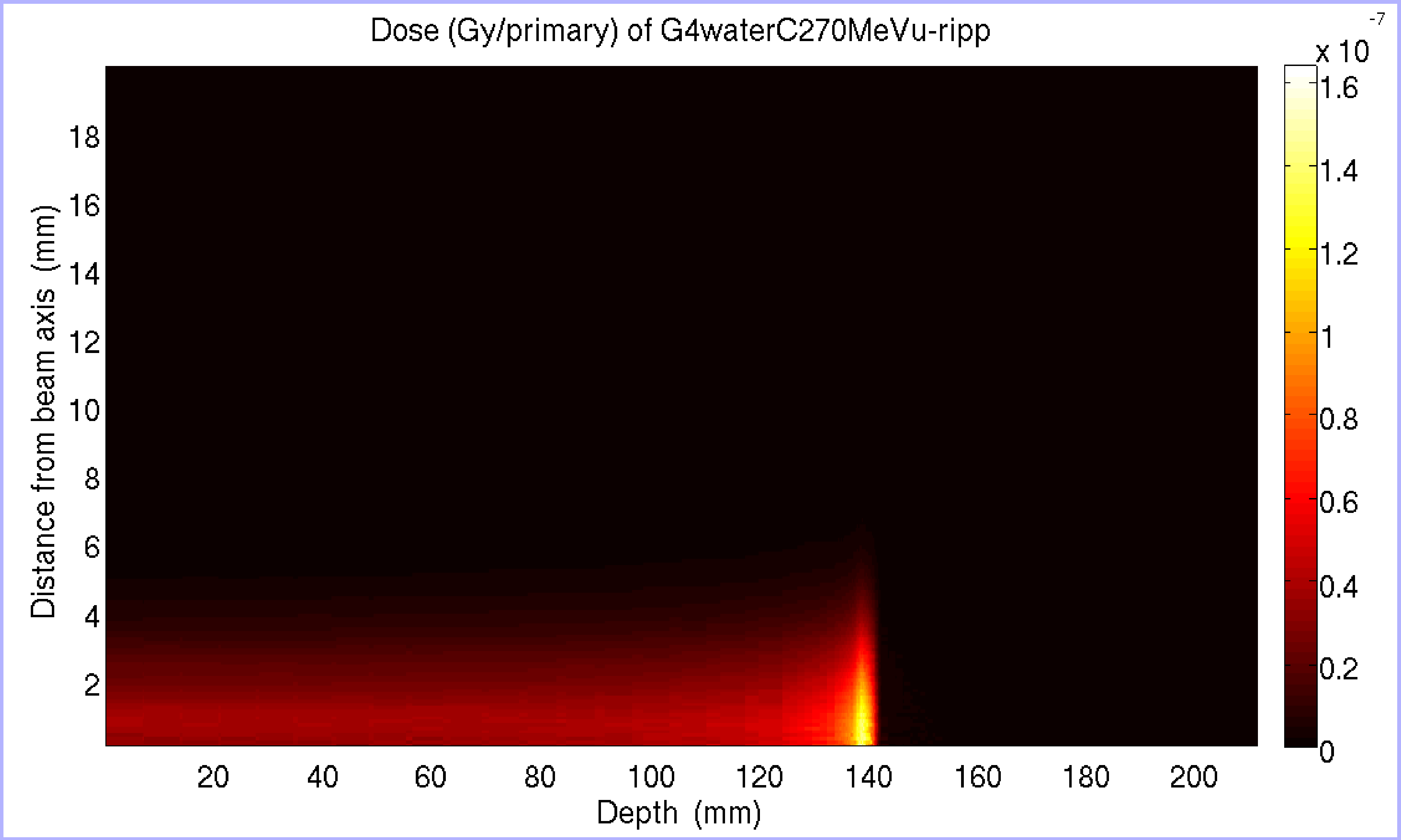
$$\{E_\mathrm{i}, F, x, y\}$$



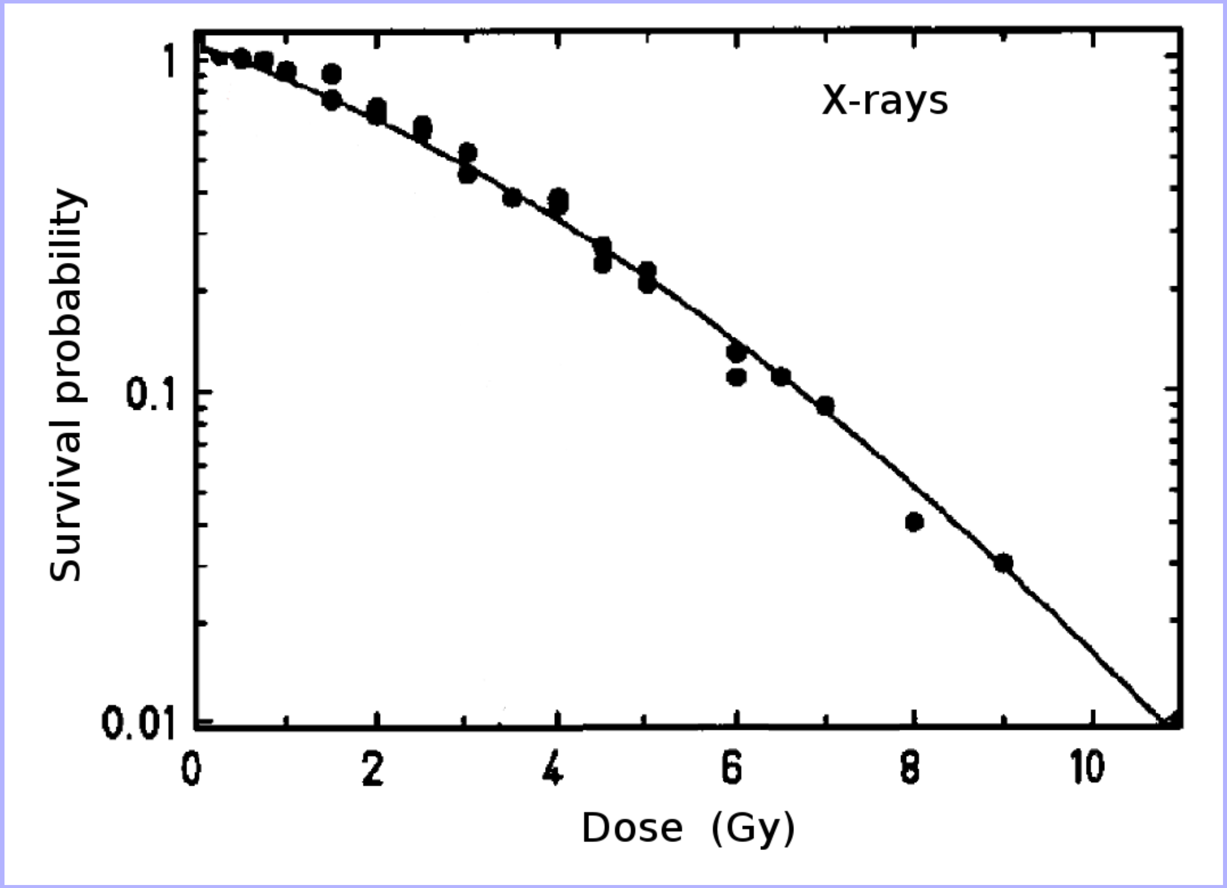
INFN TPS ⟷ iba

Cooperation Agreement
for the development of
a commercial carbon ion TPS
signed on June 2009

TPS Physical Data



TPS Biological Data | Survival curves



The biological effect depends on:

1) tissue type:

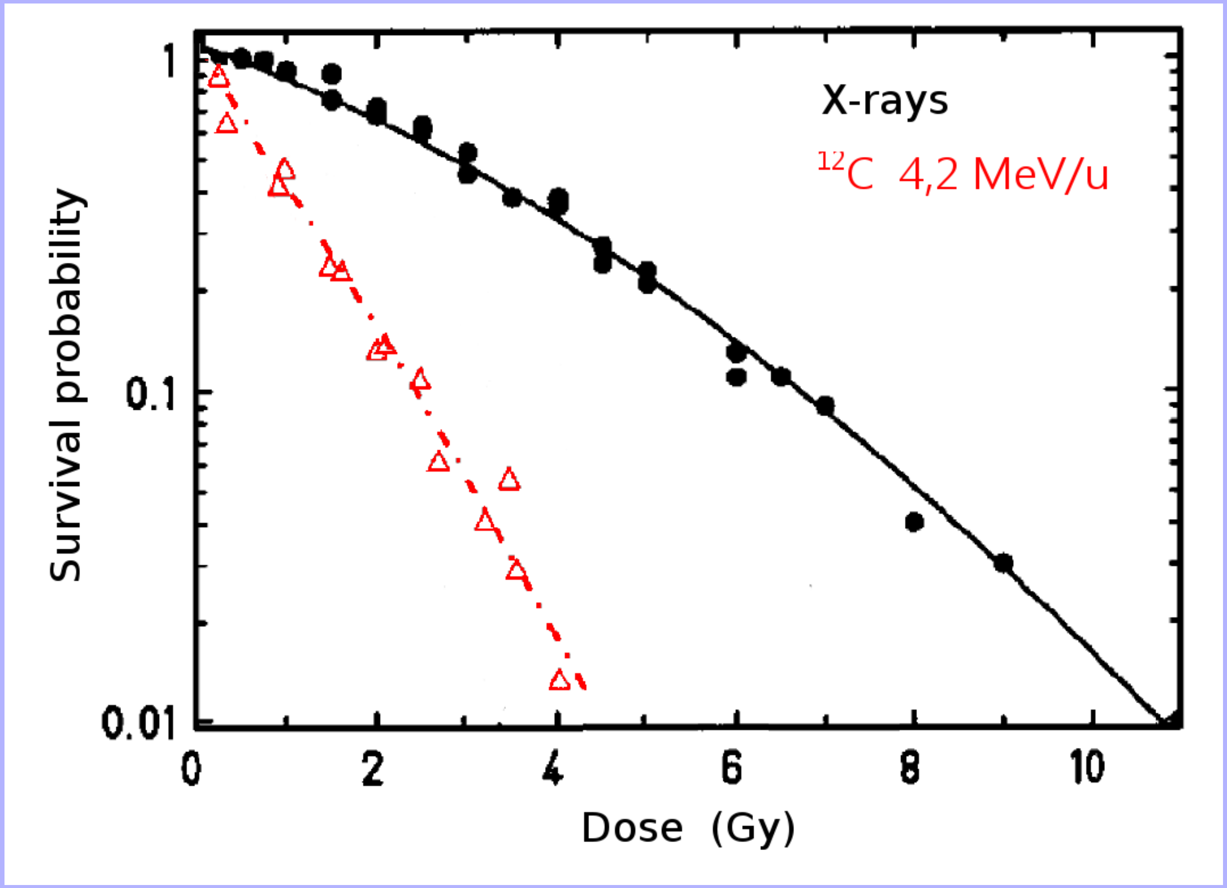
- cell line;
- oxygenation level;
- growing environment.

2) beam characteristics:

- dose released;
- radiation type.

$$S(D) = \mathrm{e}^{-\boldsymbol{\alpha}_{\mathbf{x}} D - \boldsymbol{\beta}_{\mathbf{x}} D^2}$$

TPS Biological Data | Survival curves



The biological effect depends on:

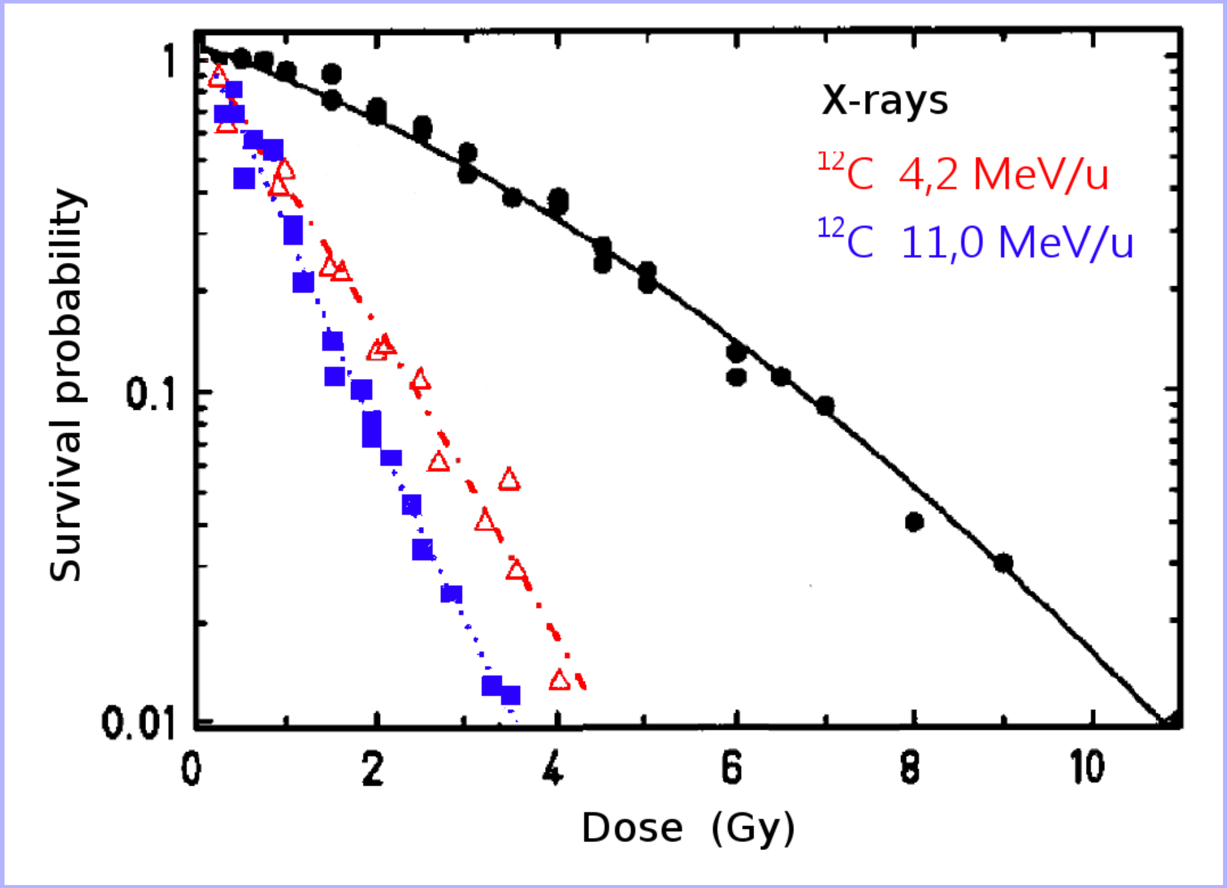
1) tissue type:
  - cell line;
  - oxygenation level;
  - growing environment.

2) beam characteristics:
  - dose released;
  - radiation type.

$$S(D) = \mathrm{e}^{-\boldsymbol{\alpha}_{\mathbf{x}} D - \boldsymbol{\beta}_{\mathbf{x}} D^2}$$

$$S(D) = \mathrm{e}^{-\boldsymbol{\alpha} D - \boldsymbol{\beta} D^2}$$

TPS Biological Data | Survival curves



The biological effect depends on:

1) tissue type:

- cell line;
- oxygenation level;
- growing environment.

2) beam characteristics:

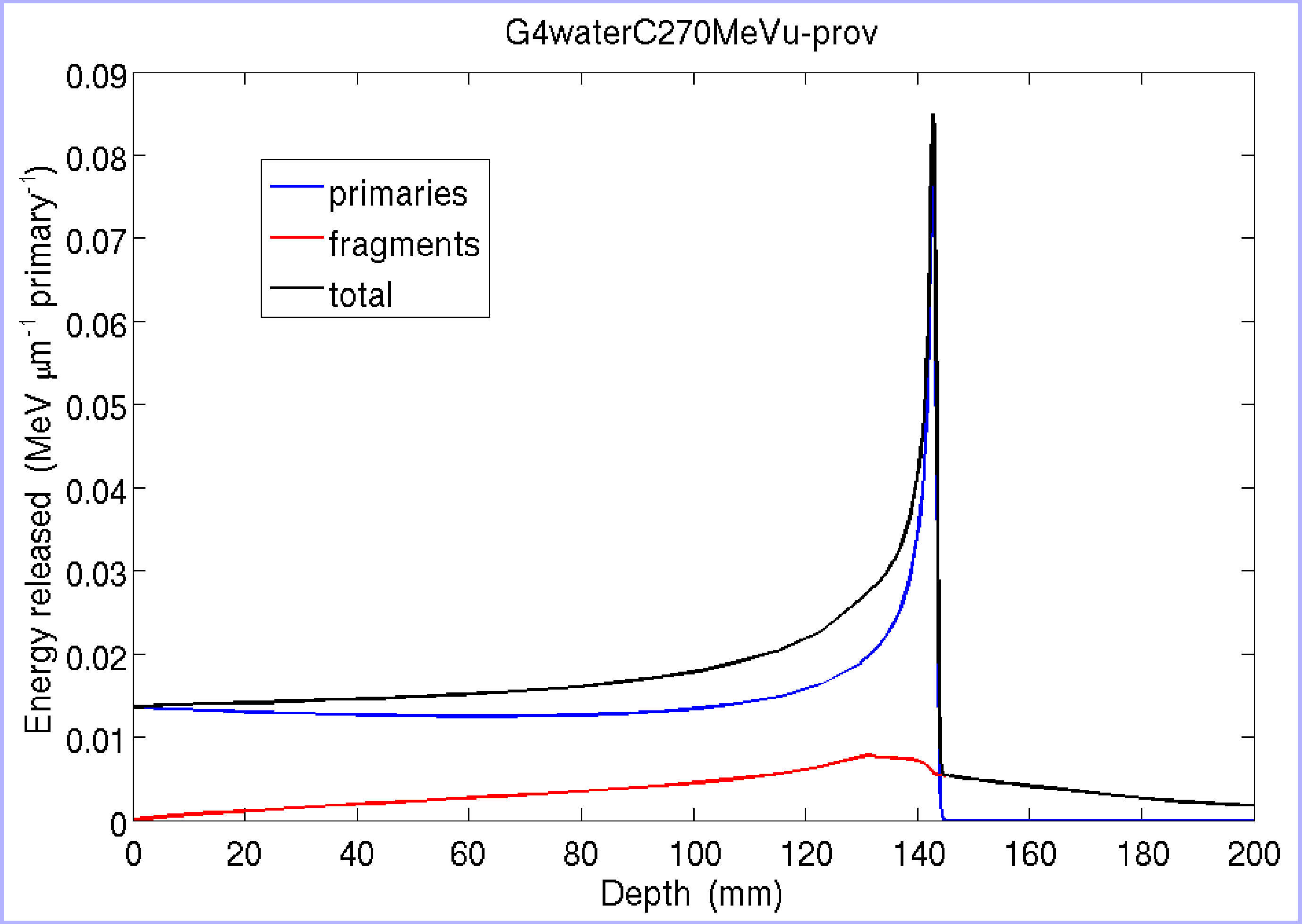
- dose released;
- radiation type;
- $p(e_c, LET)$.

$$S(D) = e^{-\boldsymbol{\alpha}_x D - \boldsymbol{\beta}_x D^2}$$

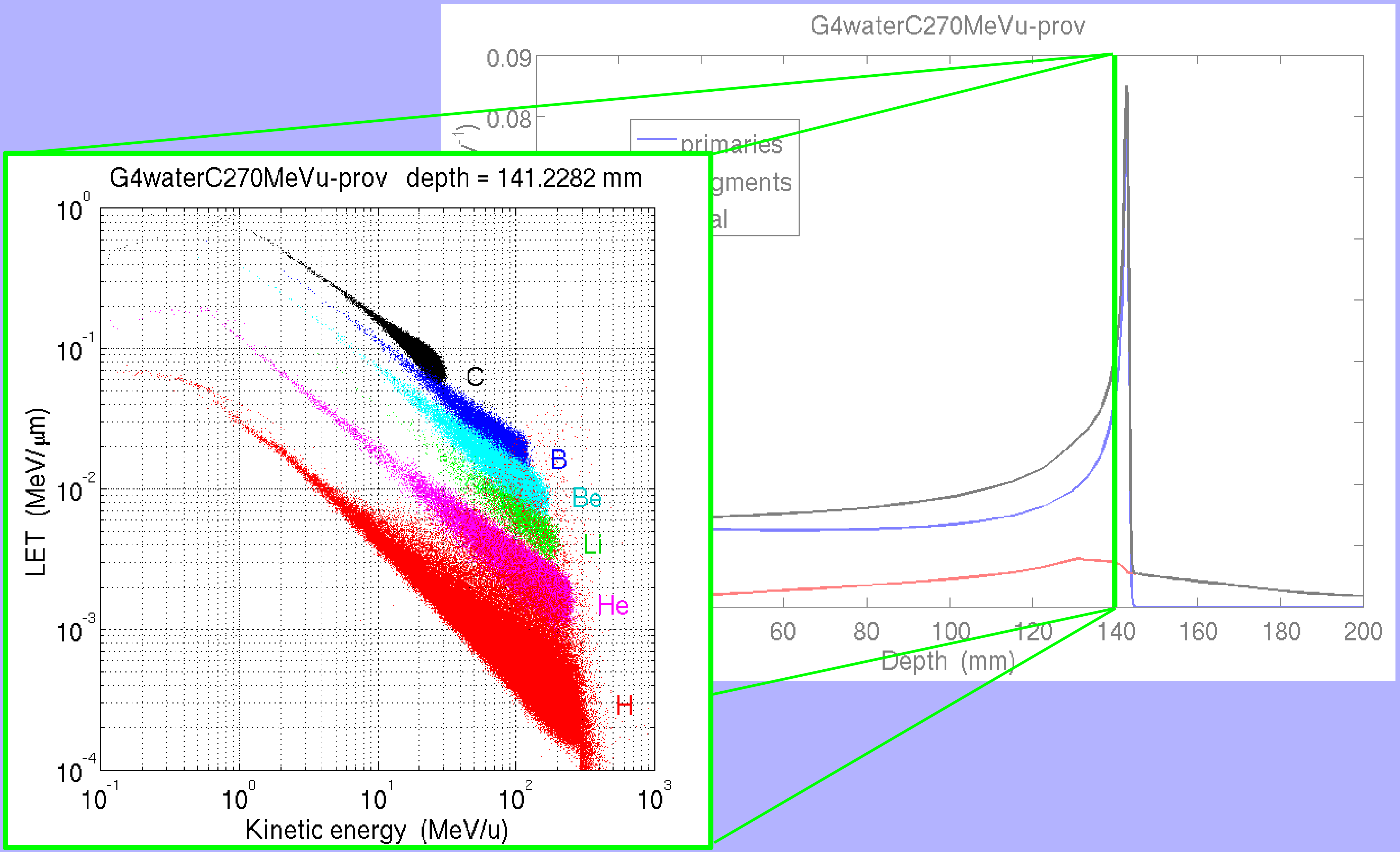
$$S(D) = e^{-\boldsymbol{\alpha} D - \boldsymbol{\beta} D^2}$$

$$S(D) = e^{-\boldsymbol{\alpha} D - \boldsymbol{\beta} D^2}$$

TPS Biological Data | Mixed fields



TPS Biological Data | Mixed fields

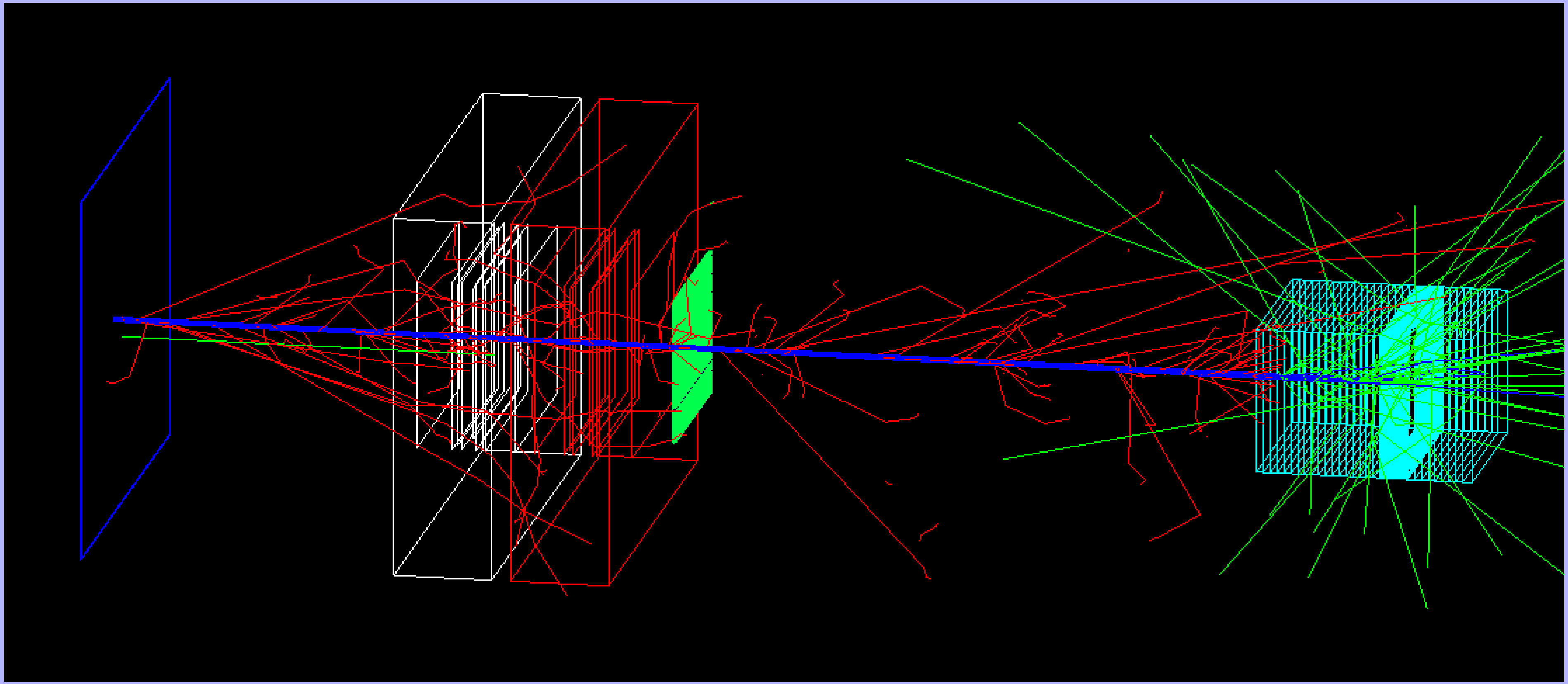


Outline of following topics

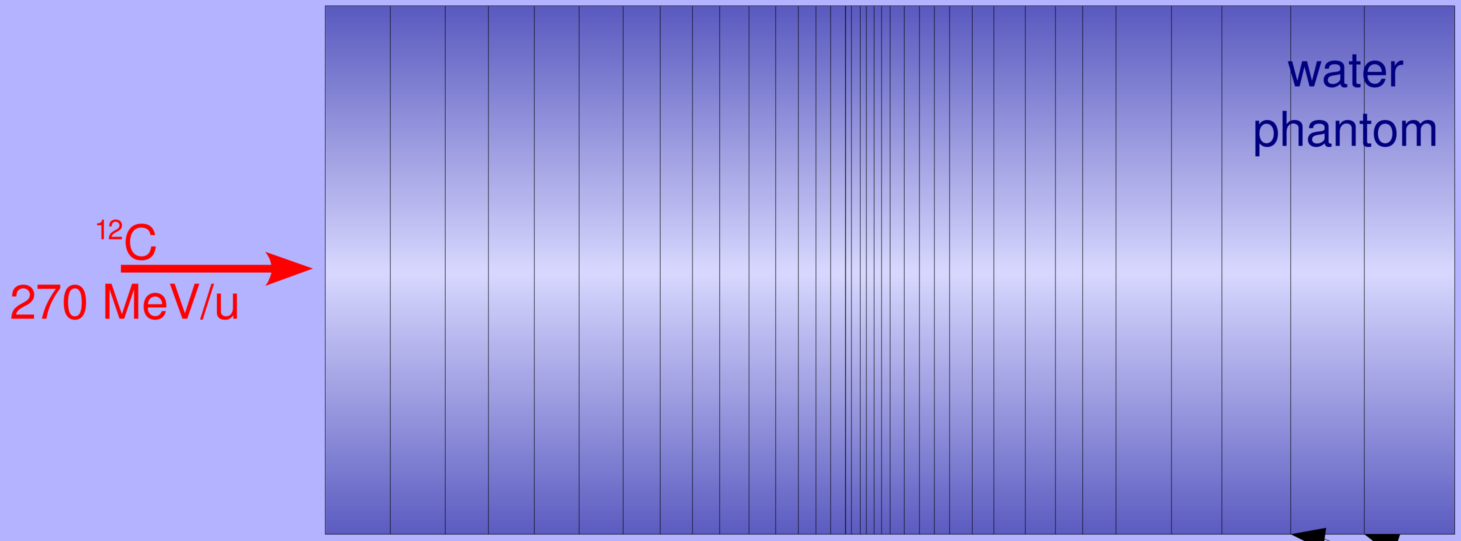
The remaining part of this presentation will show:

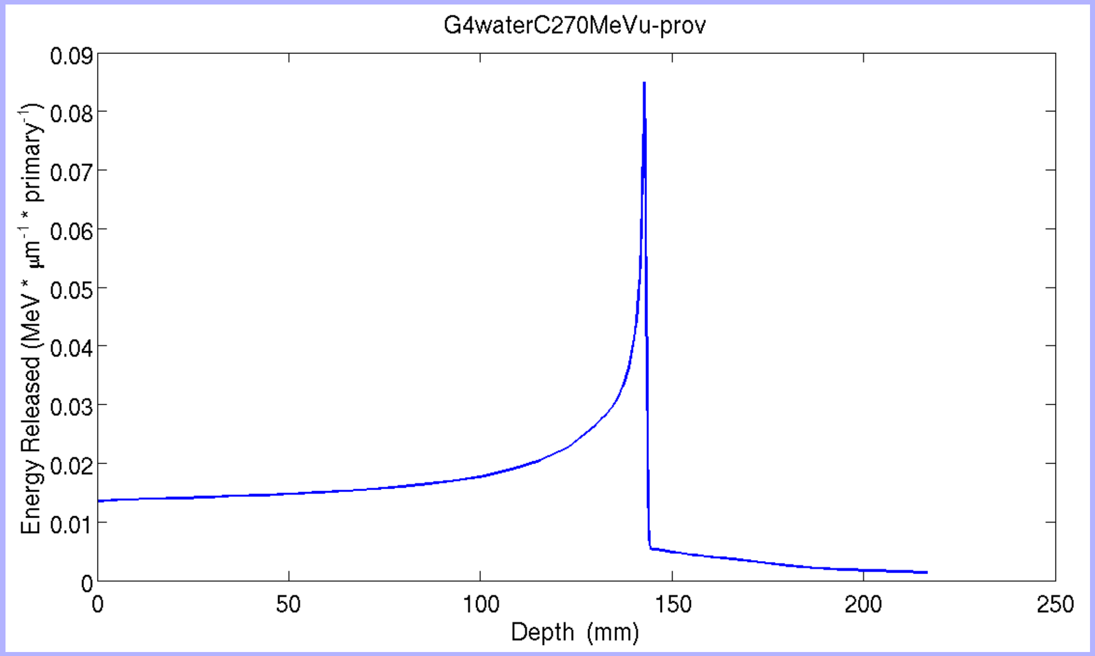
- how the Physical Data needed by TPS are created by means of **Geant4/Fluka Monte Carlo simulations**;
- how the Biological Data loaded by TPS are obtained using an **implementation of the Local Effect Model (LEM)**;
- the usage of **a relational database** for the storage and the analysis of physical and biological data;
- the way **the TPS Kernel** is structured;
- the **establishment of the Quality Assurance** management procedures needed for the certification of medical device softwares.

TPS Physical Data | Geant4 (Fluka) simulation



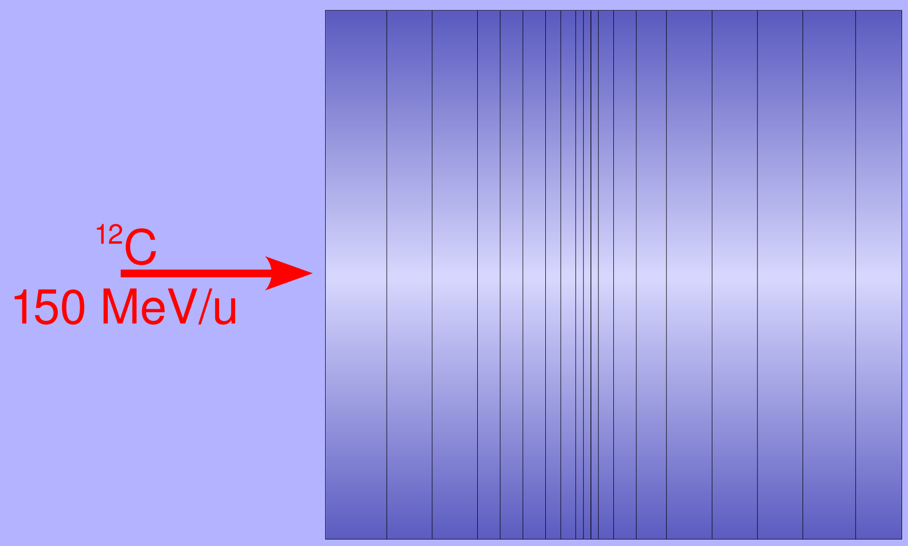
TPS Physical Data | Geant4 (Fluka) simulation



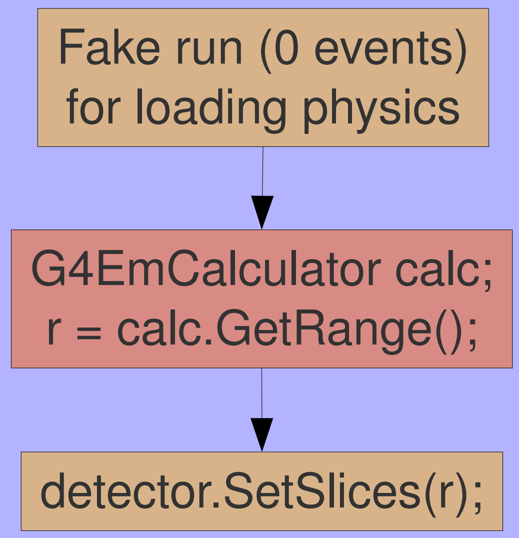


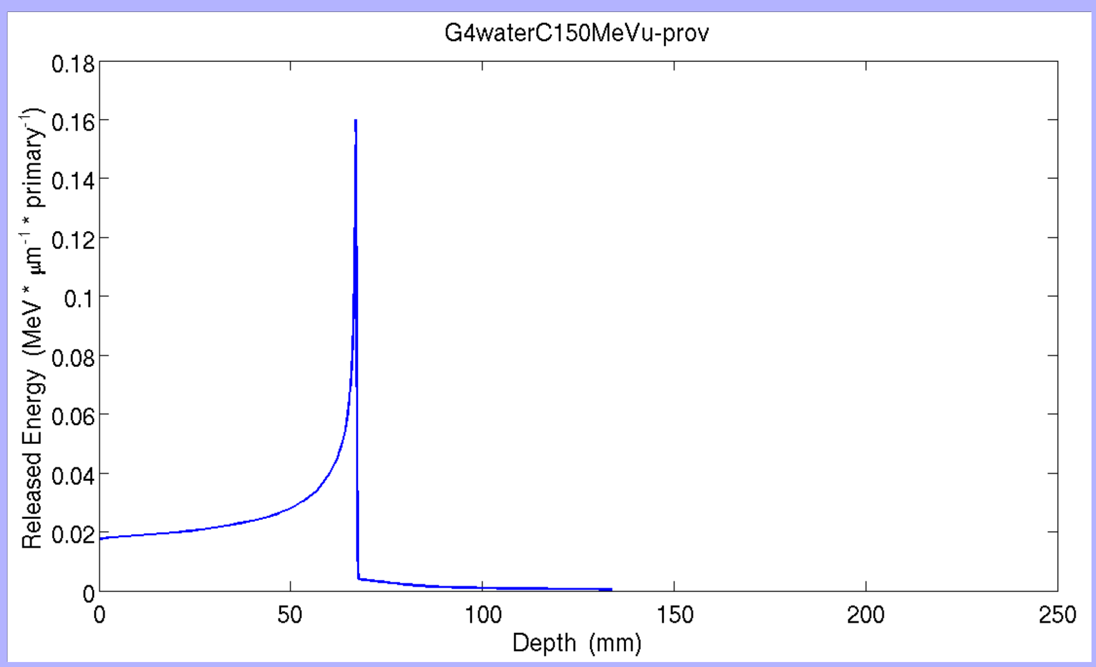
Slices as SensitiveDetectors (10 um wide to be comparable to cell dimensions)

TPS Physical Data | Geant4 (Fluka) simulation

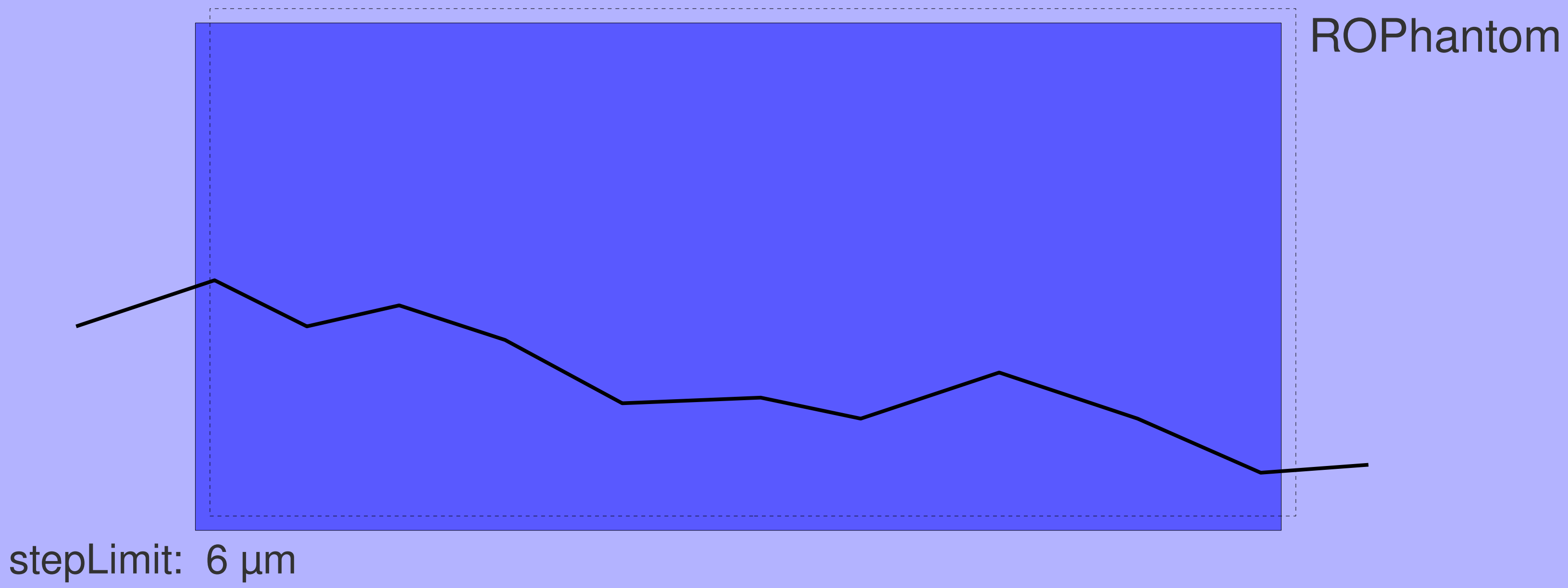


Adaptive geometry

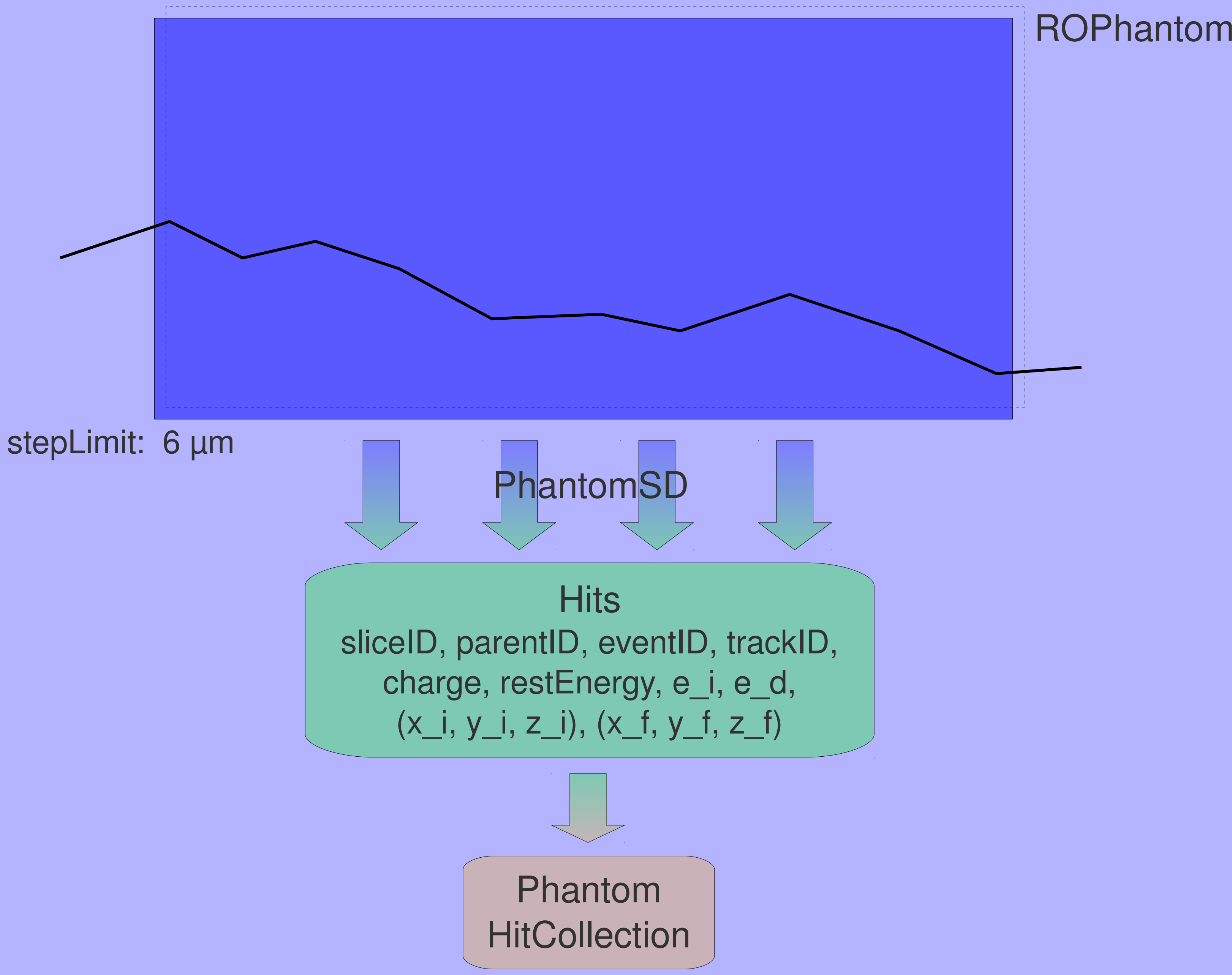




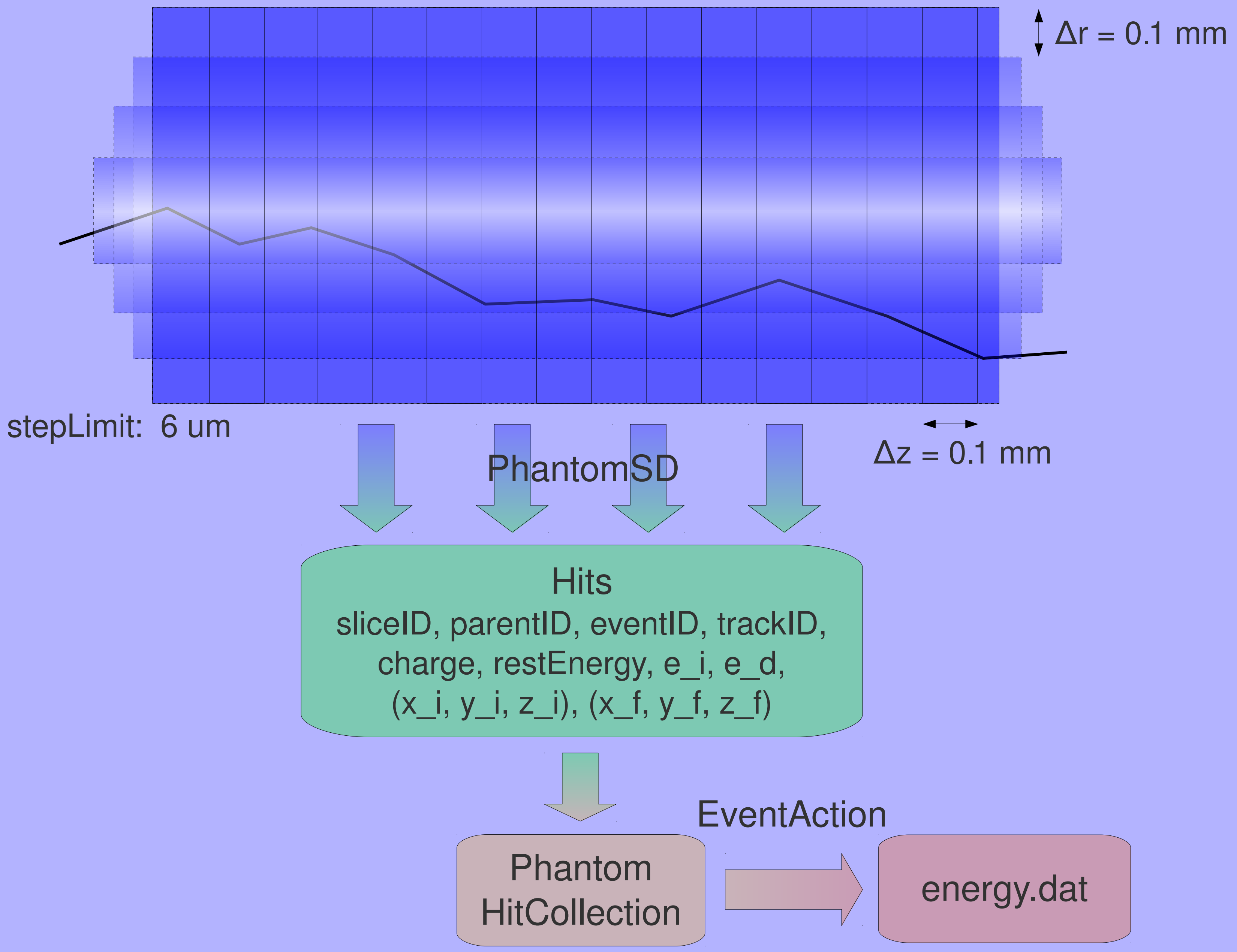
TPS Physical Data | Geant4 (Fluka) simulation



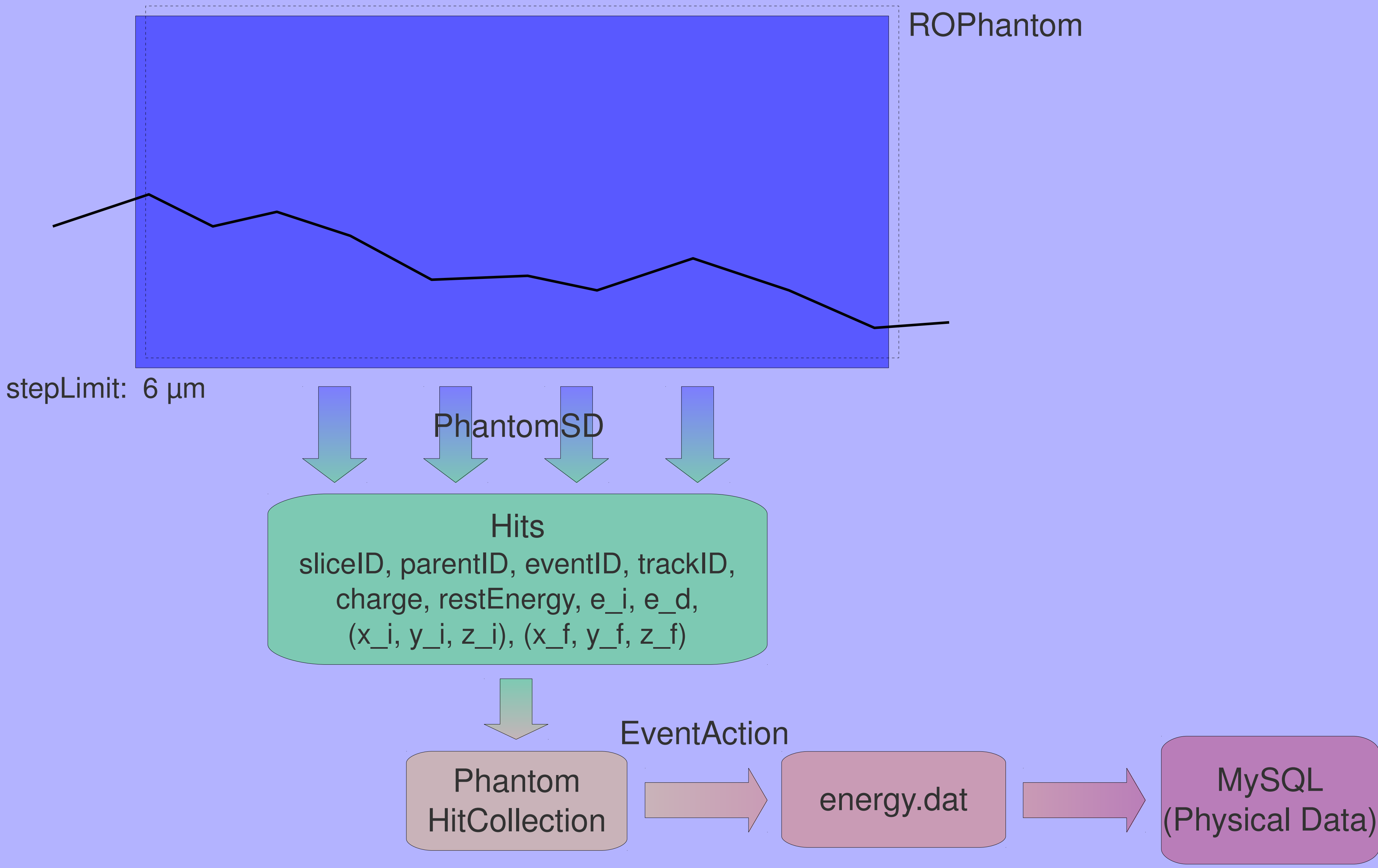
TPS Physical Data | Geant4 (Fluka) simulation



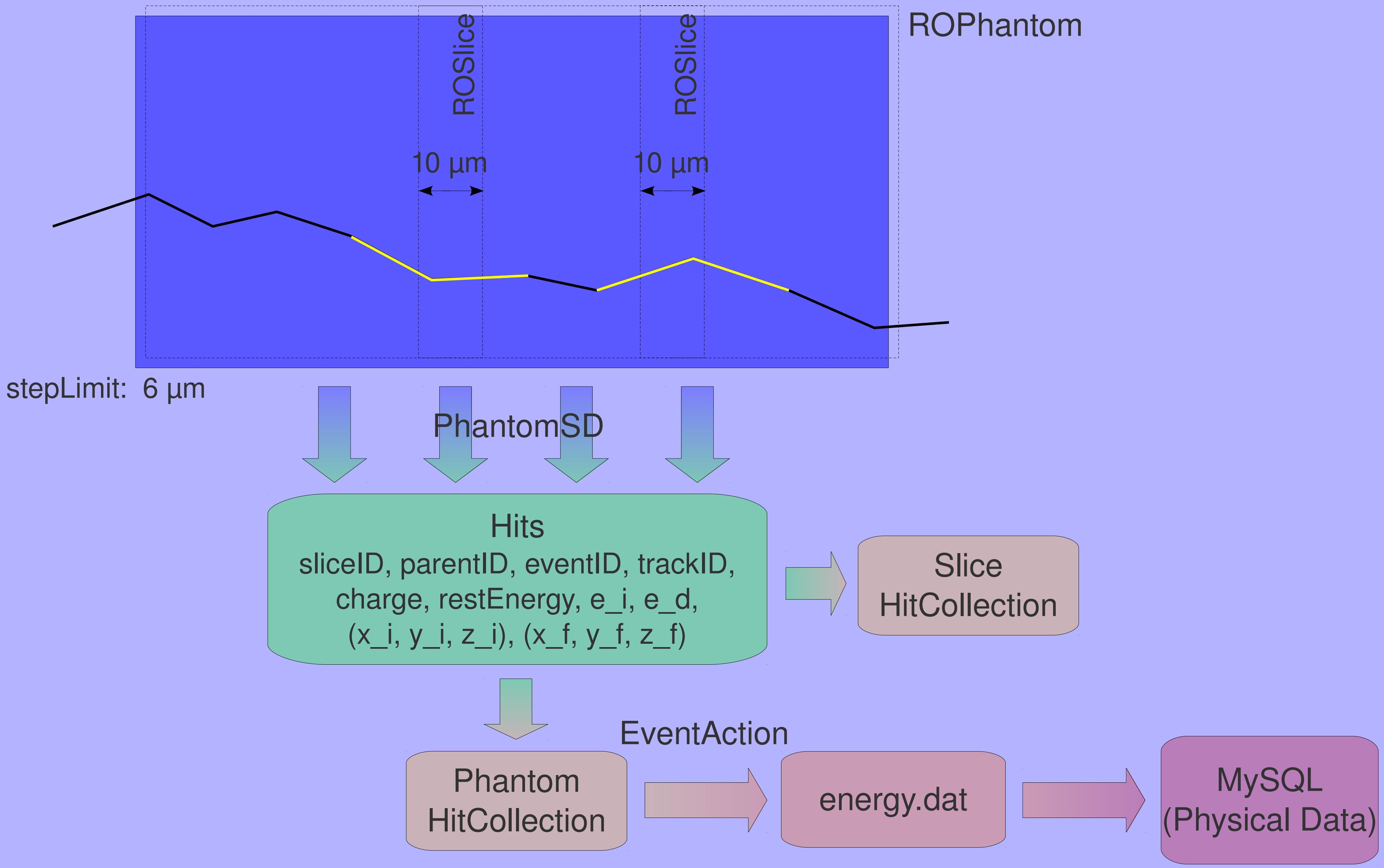
TPS Physical Data | Geant4 (Fluka) simulation



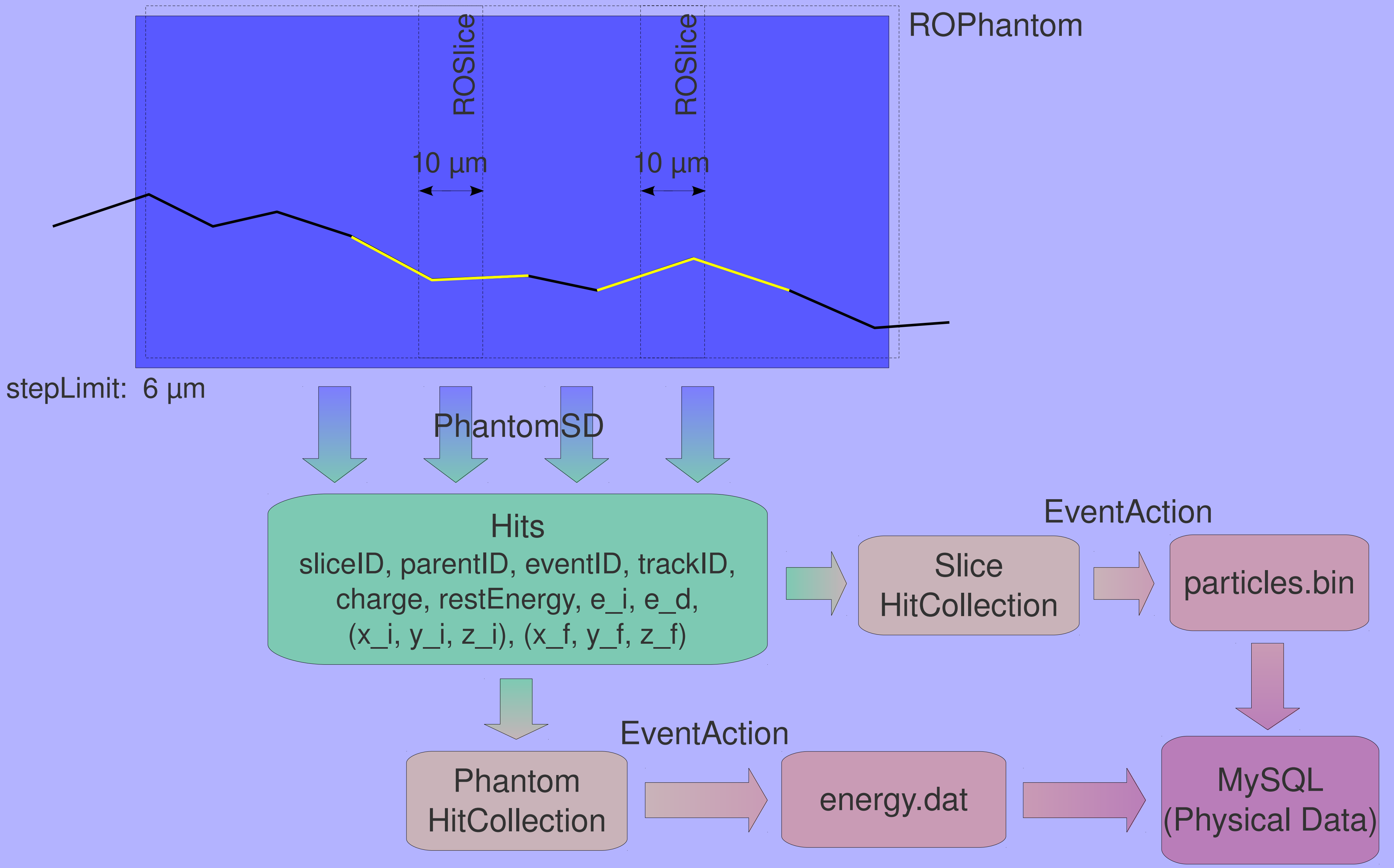
TPS Physical Data | Geant4 (Fluka) simulation



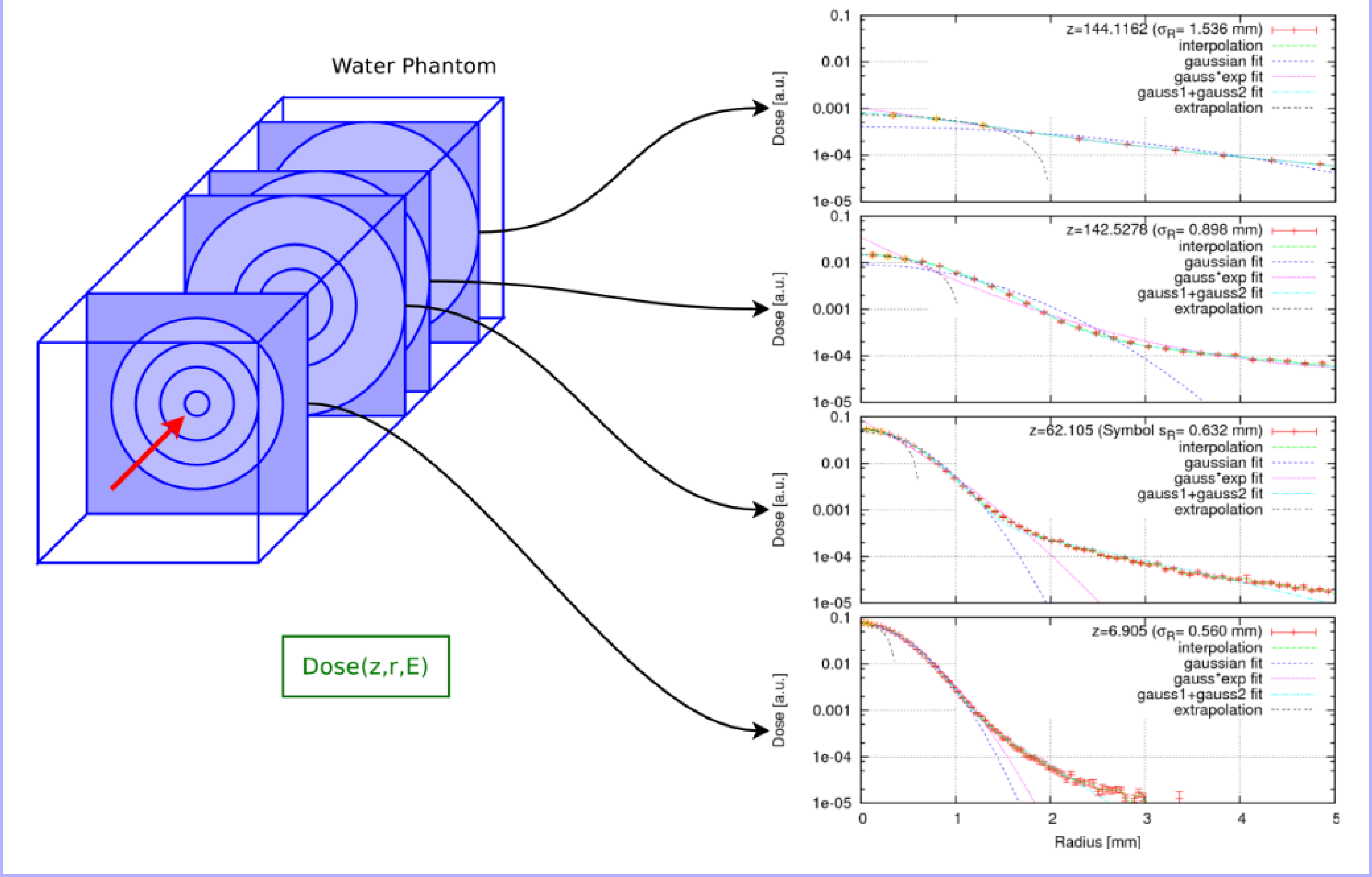
TPS Physical Data | Geant4 (Fluka) simulation



TPS Physical Data | Geant4 (Fluka) simulation

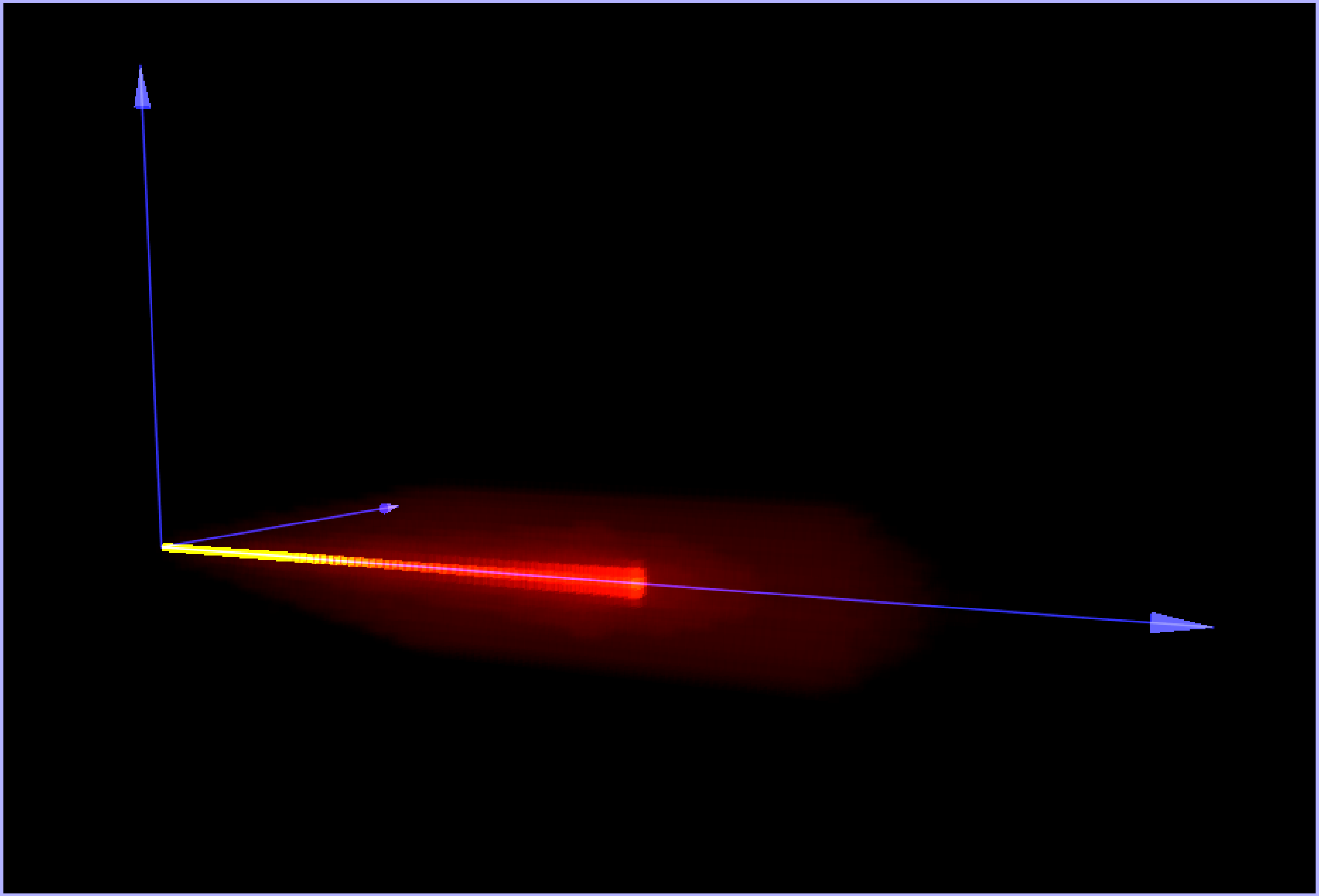


TPS Physical Data | Data analysis

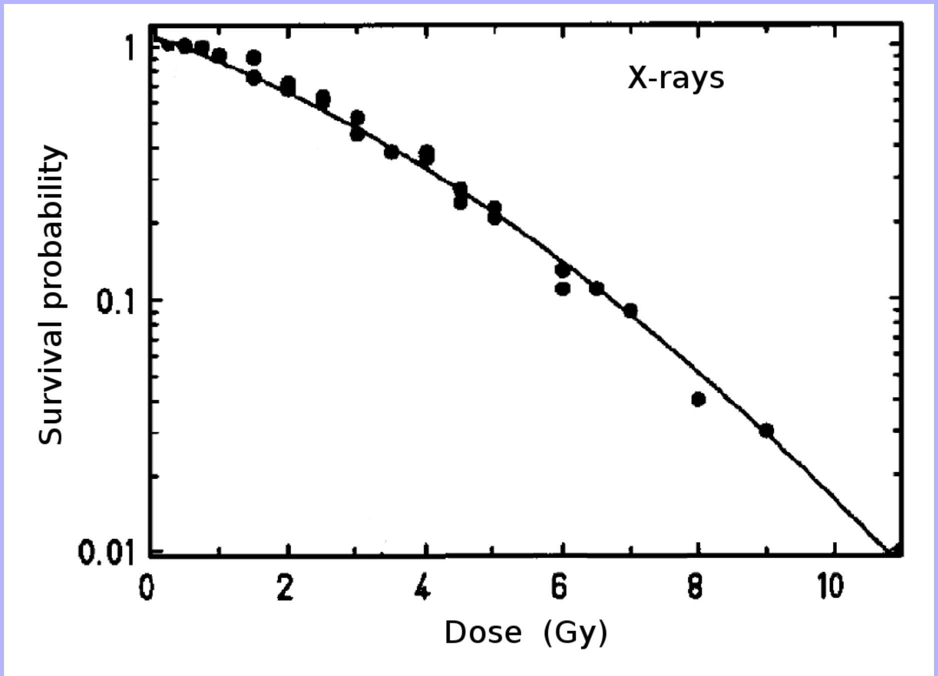


TPS Physical Data | Pencil beams

Spatial distribution of dose

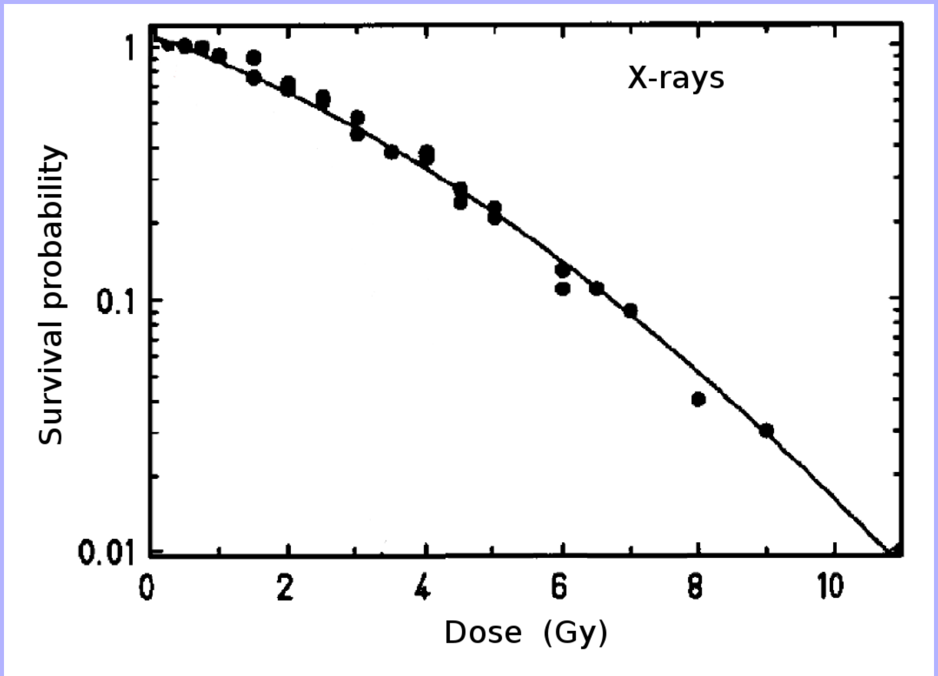


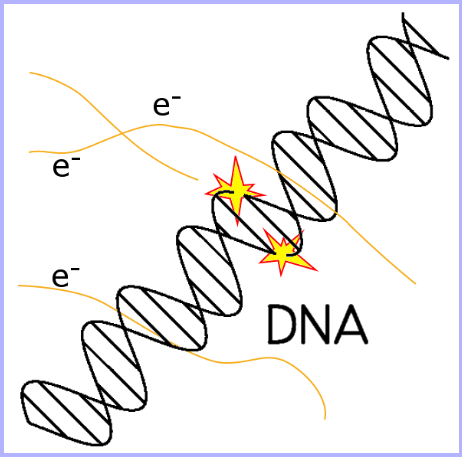
Implementation of the LEM | Lethal events



$$S_x(D) = e^{-\alpha_x D - \beta_x D^2}$$

Implementation of the LEM | Lethal events



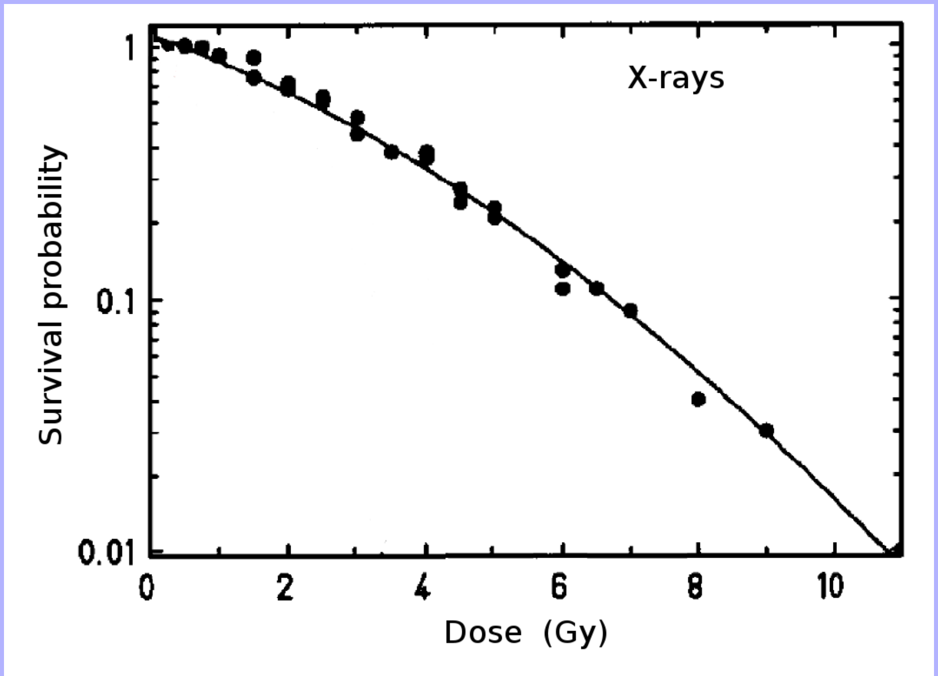


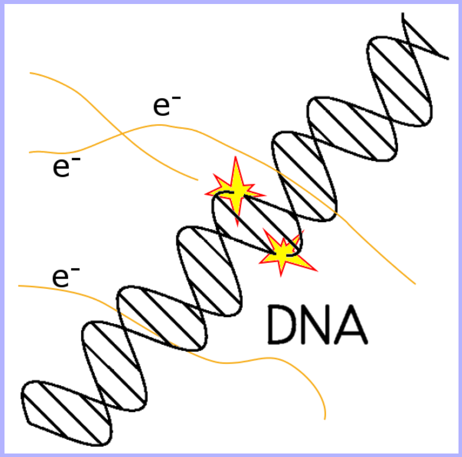
$$S_x(D) = e^{-\alpha_x D - \beta_x D^2}$$

$$N_{leth} = \mathrm{Pois}(\overline{N}_{leth})$$

$$S_x = P[N_{leth} = 0] = e^{-\overline{N}_{leth}}$$

Implementation of the LEM | Lethal events

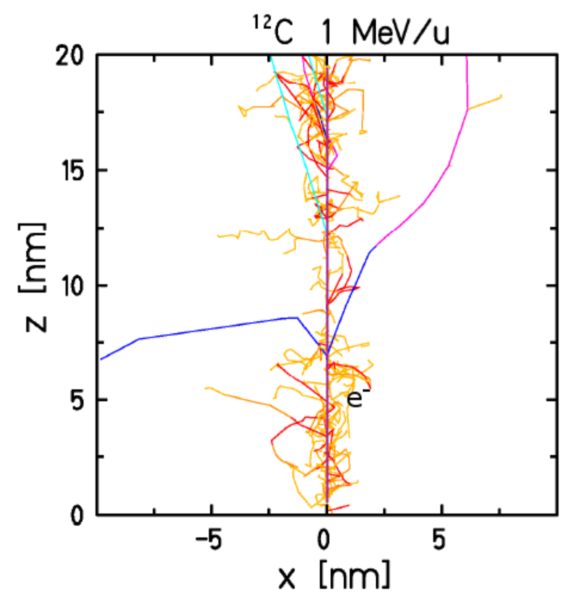




$$S_x(D) = e^{-\alpha_x D - \beta_x D^2}$$

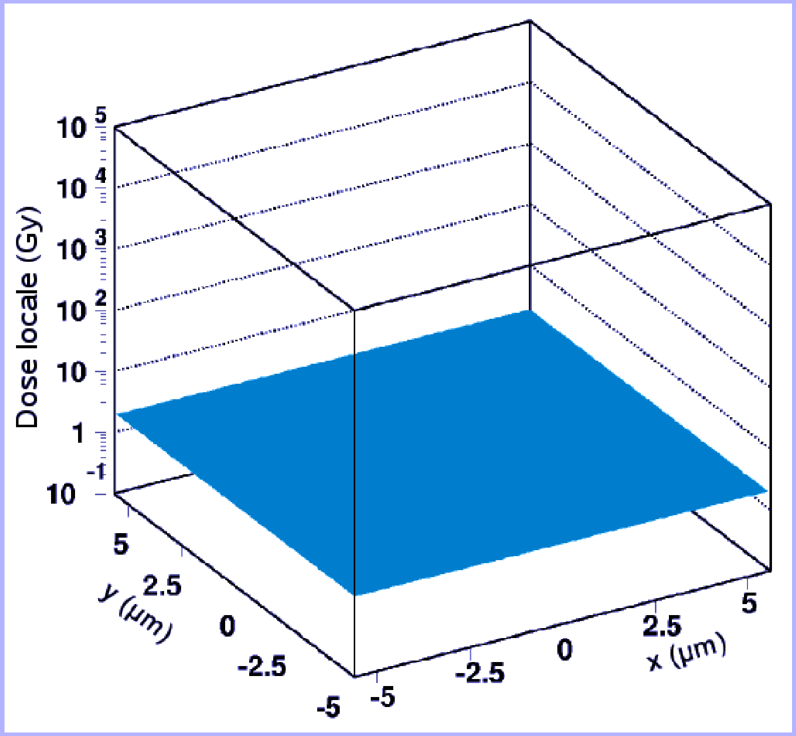
$$N_{\text{leth}} = \text{Pois}(\overline{N}_{\text{leth}})$$

$$S_x = P[N_{\text{leth}} = 0] = e^{-\overline{N}_{\text{leth}}}$$

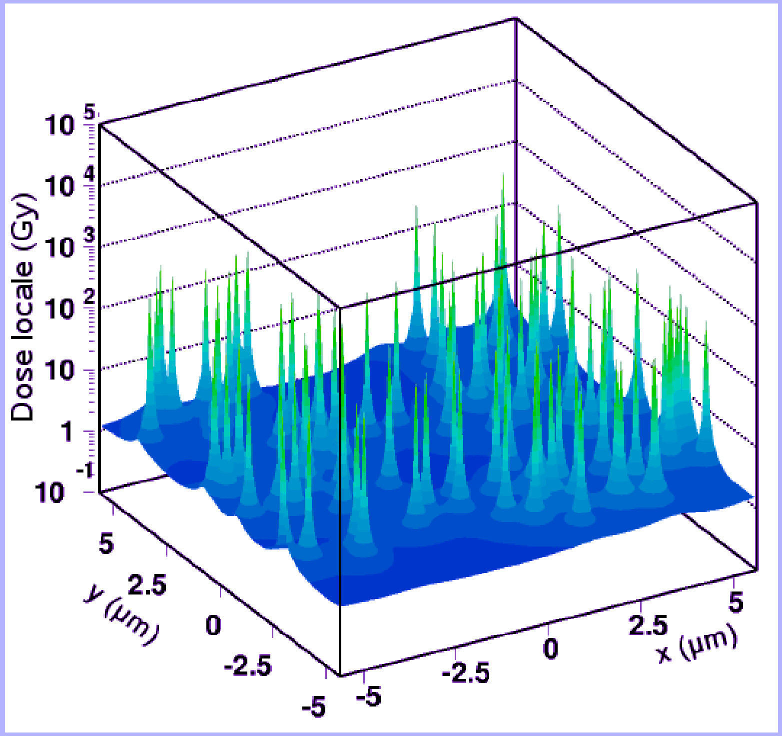


Implementation of the LEM | Photons vs ions

Total dose = 2 Gy



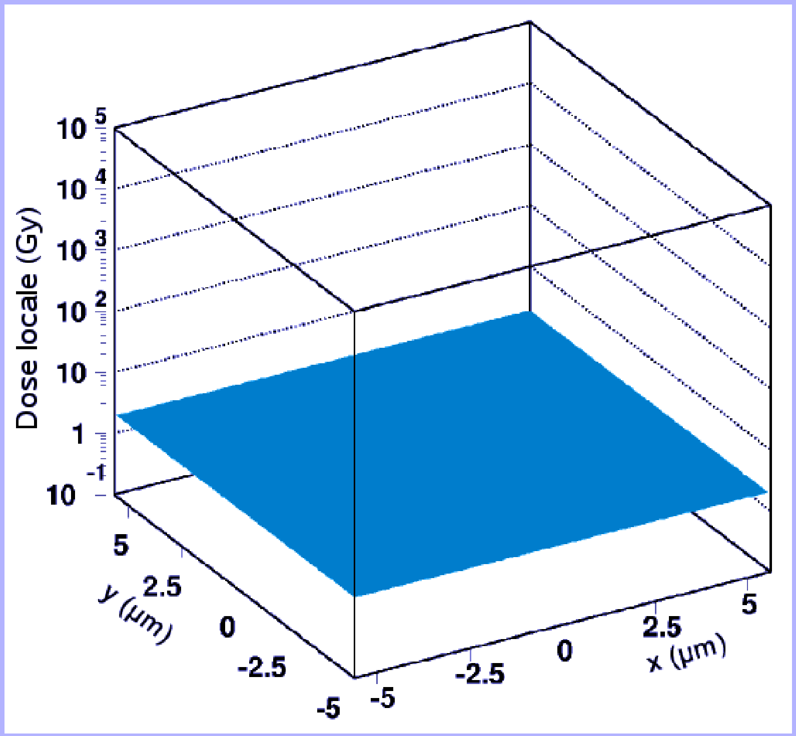
X-rays



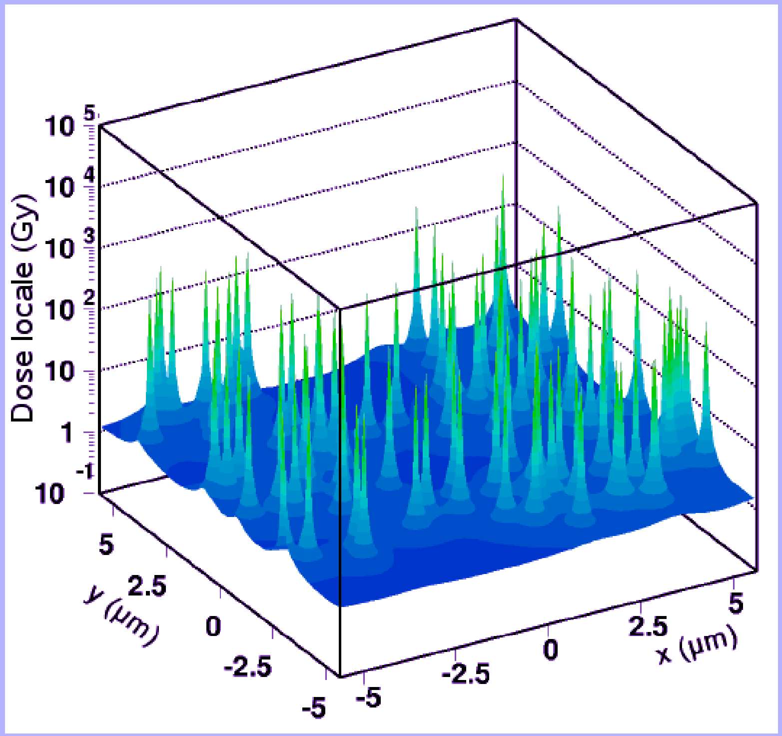
90 MeV/u carbon ions

Implementation of the LEM | Photons vs ions

Total dose = 2 Gy



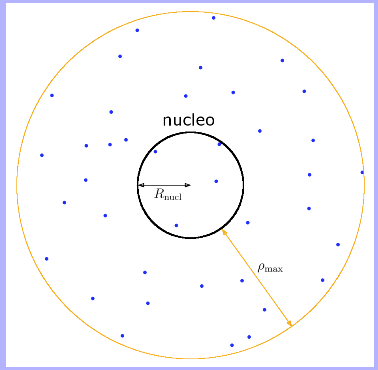
X-rays

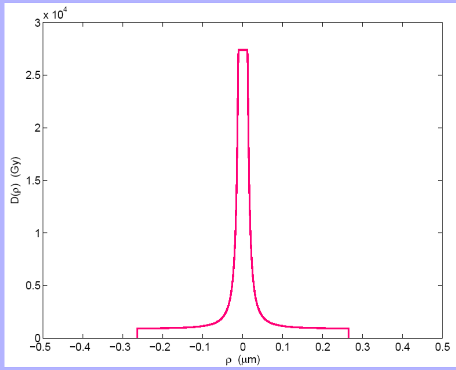


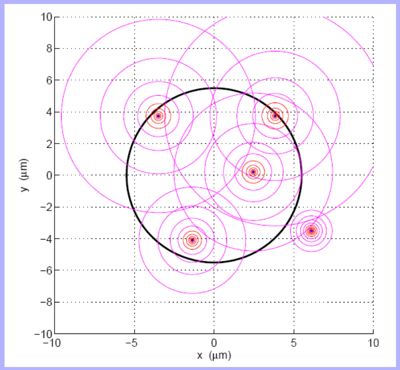
90 MeV/u carbon ions

$$\overline{N}_{\rm leth} = \alpha_{\rm x} D + \beta_{\rm x} D^2 \quad \longrightarrow \quad {\rm d}\overline{N}_{\rm leth} = (\alpha_{\rm x} d(\vec{r}) + \beta_{\rm x} d(\vec{r})^2) \frac{{\rm d}\vec{r}}{V_{\rm nucl}}$$

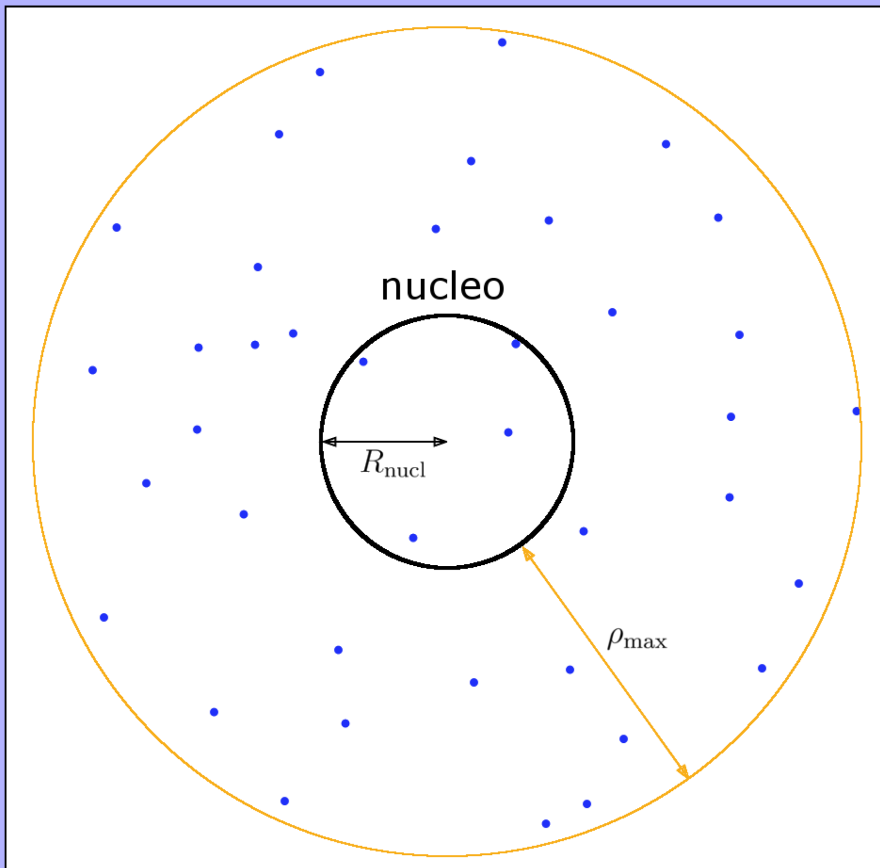
Implementation of the LEM | Event generation

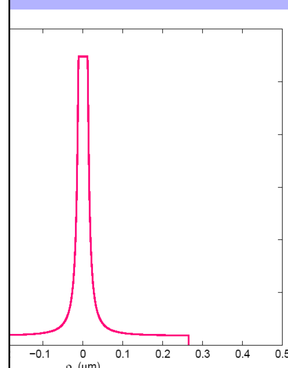


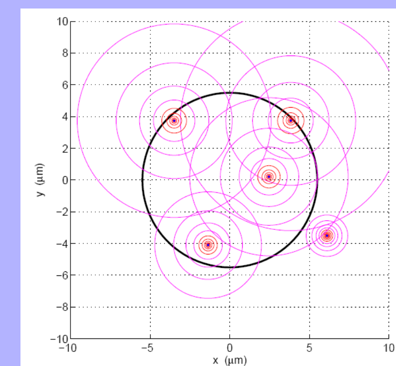




Implementation of the LEM | Event generation



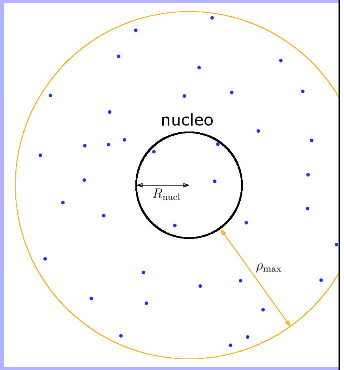


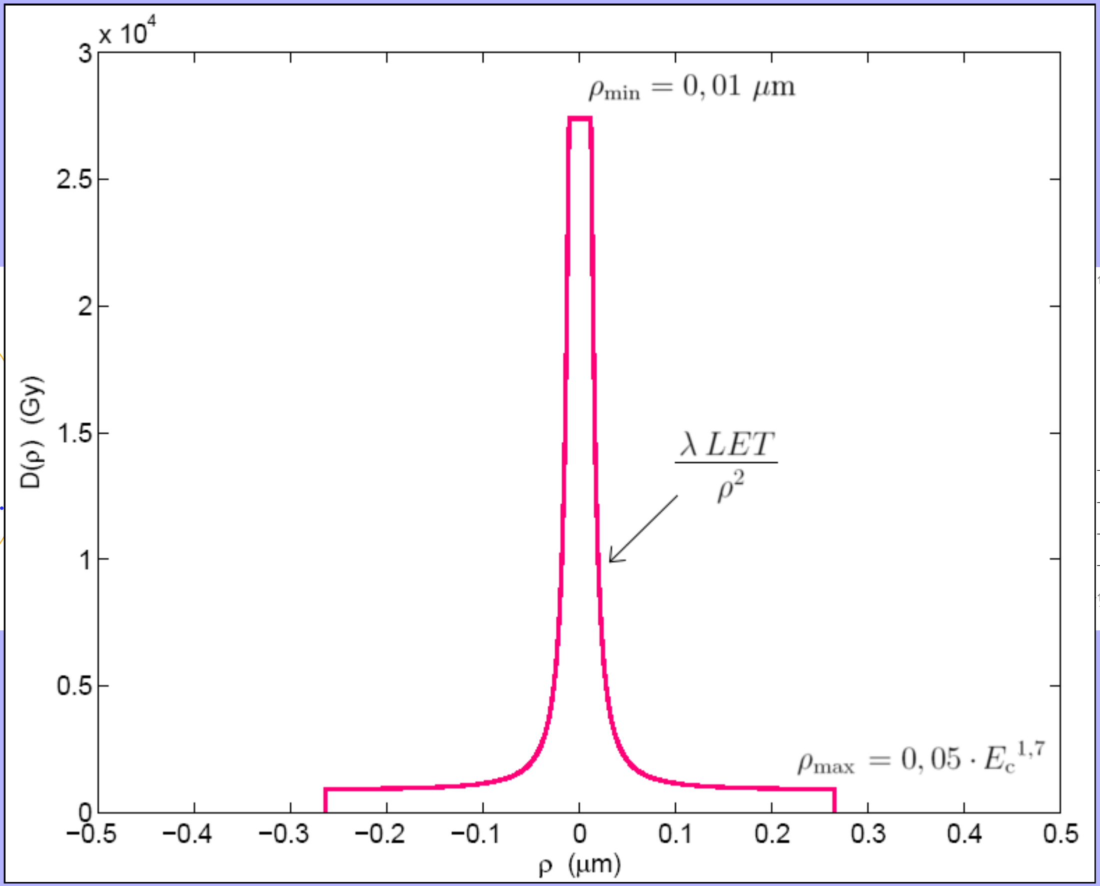


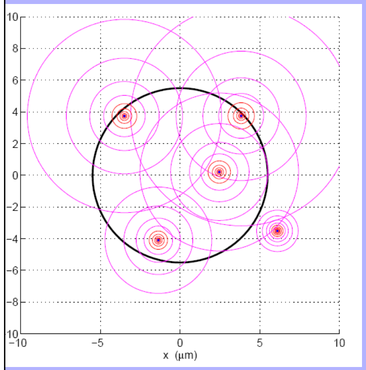
$$F = \frac{\varrho \langle D \rangle}{\overline{LET}}$$

$$\langle n_{\text{ion}} \rangle = F \cdot \pi (R_{\text{nucl}} + \rho_{\max})^2$$

Implementation of the LEM | Event generation

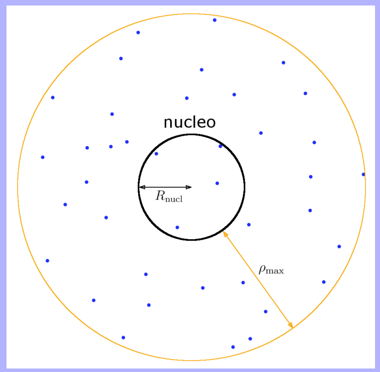


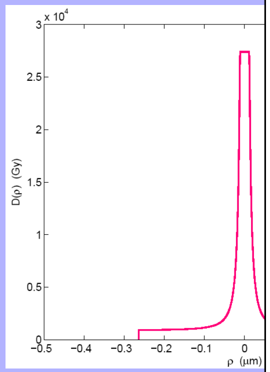


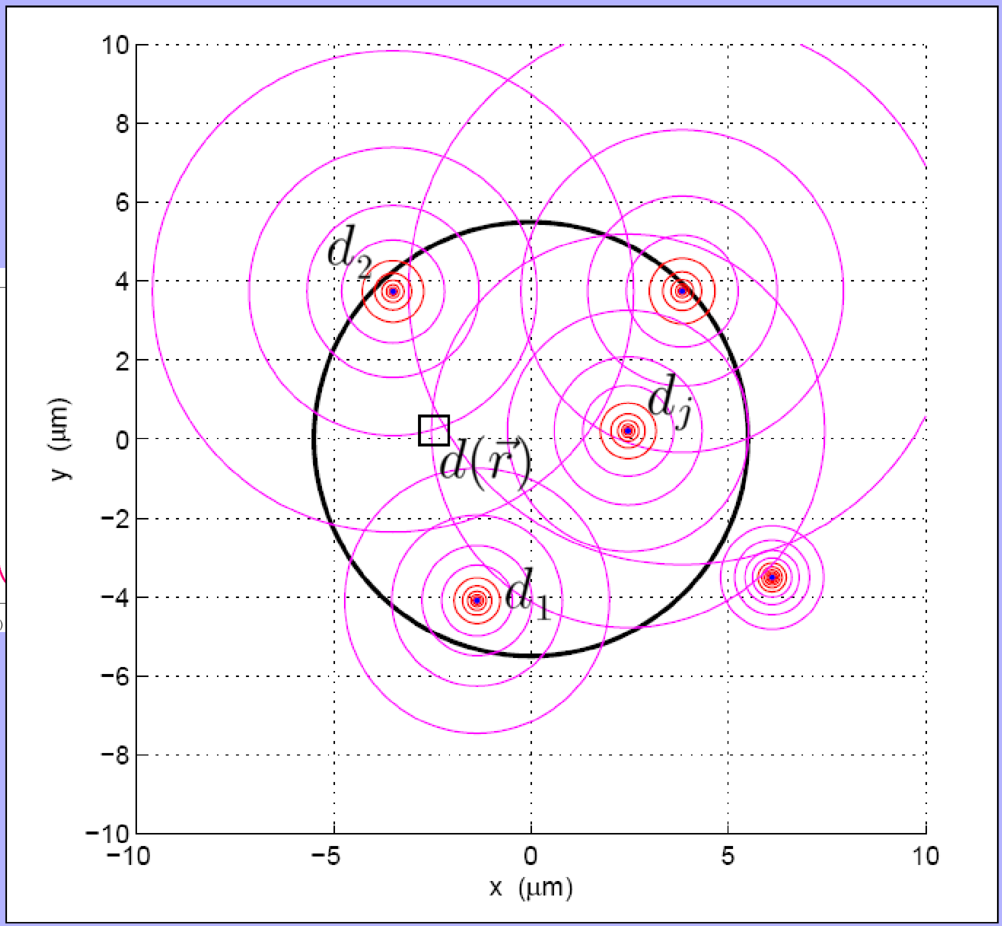


$$d(\rho) = \begin{cases} \lambda\, LET/\rho_{\rm min}^2 & \rho < \rho_{\rm min} \\ \lambda\, LET/\rho^2 & \rho_{\rm min} \leq \rho \leq \rho_{\rm max} \\ 0 & \rho > \rho_{\rm max} \end{cases}$$

Implementation of the LEM | Event generation







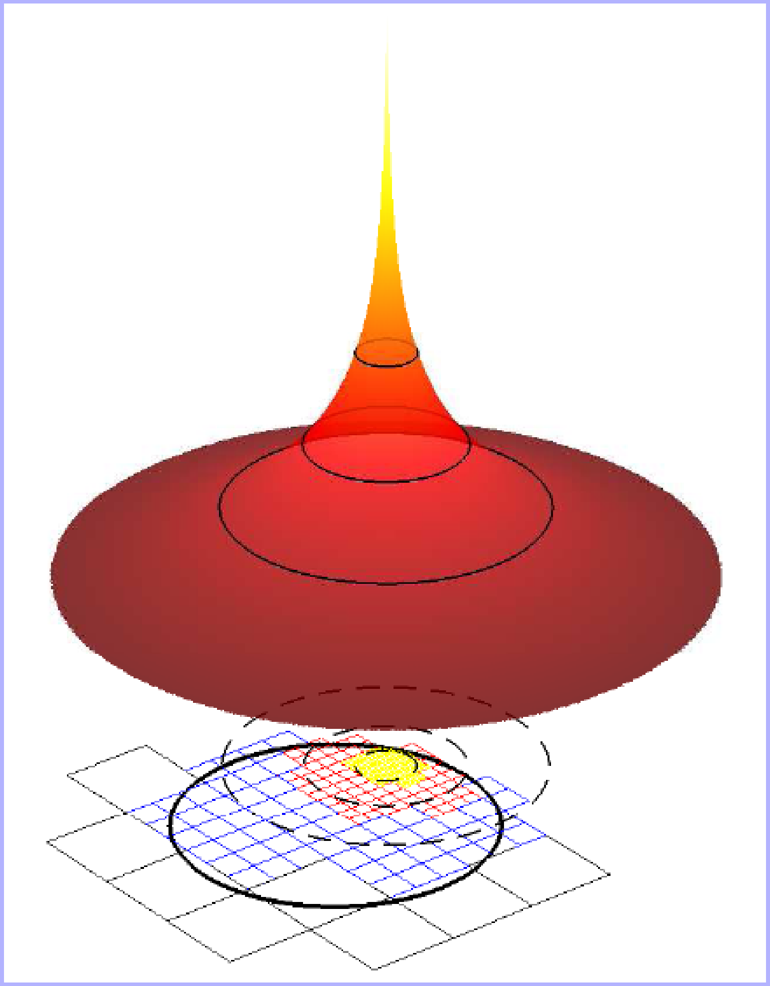
$$d(\vec{r}) = \sum_{\rm j=1}^{n_{\rm t}} d_{\rm j}(\vec{r})$$

Implementation of the LEM | Damage integration

$$d(\vec{r}) \longrightarrow \mathrm{d}\overline{N}_{\mathrm{leth}} = (\alpha_{\mathrm{x}} d(\vec{r}) + \beta_{\mathrm{x}} d(\vec{r})^2) \frac{\mathrm{d}\vec{r}}{V_{\mathrm{nucl}}}$$

$$\overline{N}_{\mathrm{leth}} = \frac{1}{V_{\mathrm{nucl}}} \int_{V_{\mathrm{nucl}}} (\alpha_{\mathrm{x}} d(\vec{r}) + \beta_{\mathrm{x}} d(\vec{r})^2)\, \mathrm{d}\vec{r}$$

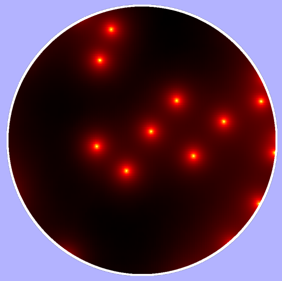
$$S = \mathrm{e}^{-\overline{N}_{\mathrm{leth}}}$$



Implementation of the LEM | Statistical fluctuations

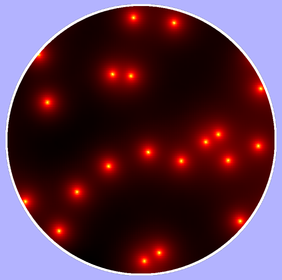
$$\langle D \rangle = 3\,\mathrm{Gy}$$

Implementation of the LEM | Statistical fluctuations



$D_1 = 2.02\ \mathrm{Gy}$

$S_1 = 0.228$

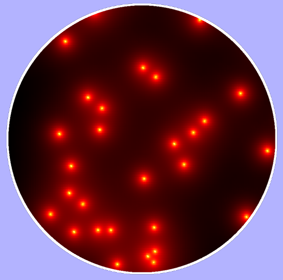


$D_2 = 2.99\ \mathrm{Gy}$

$S_2 = 0.13$

$\langle D \rangle = 3\ \mathrm{Gy}$

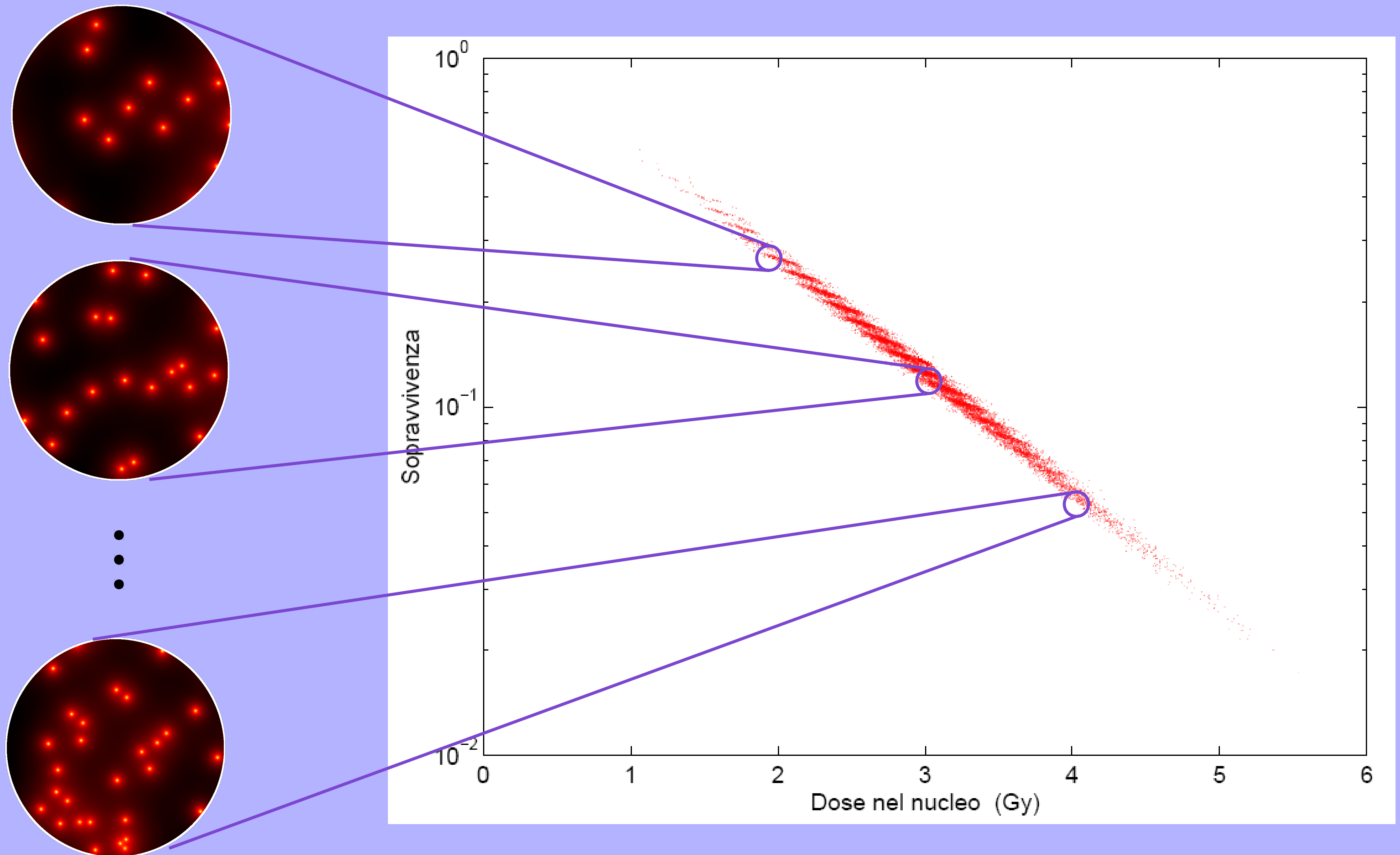
⋮



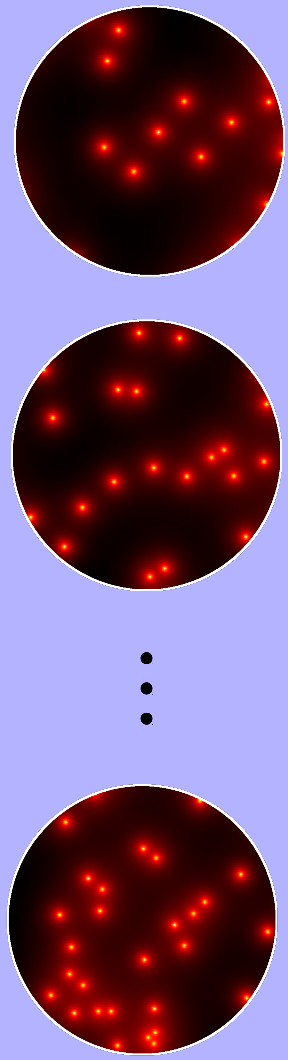
$D_N = 4.21\ \mathrm{Gy}$

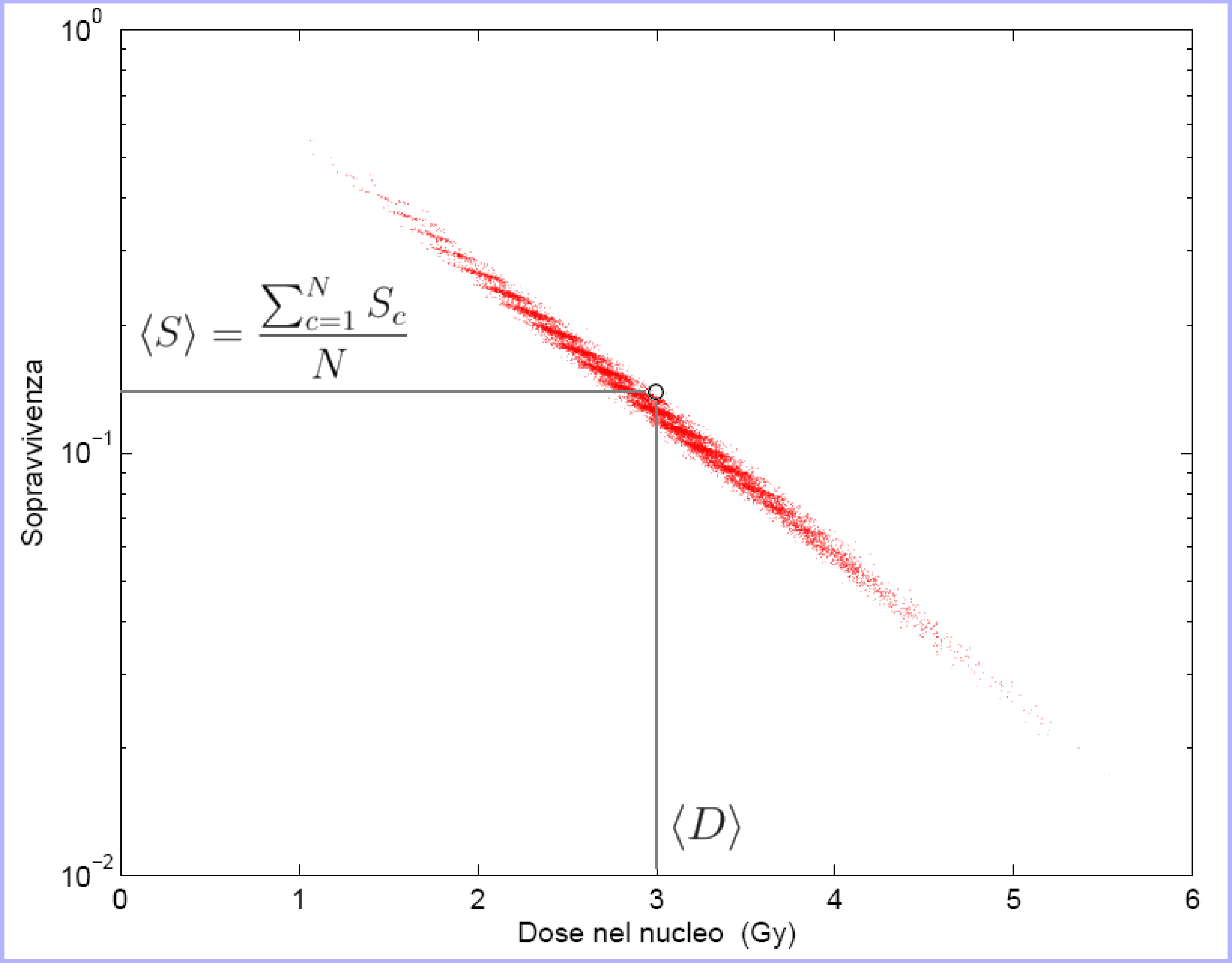
$S_N = 0.05$

Implementation of the LEM | Statistical fluctuations

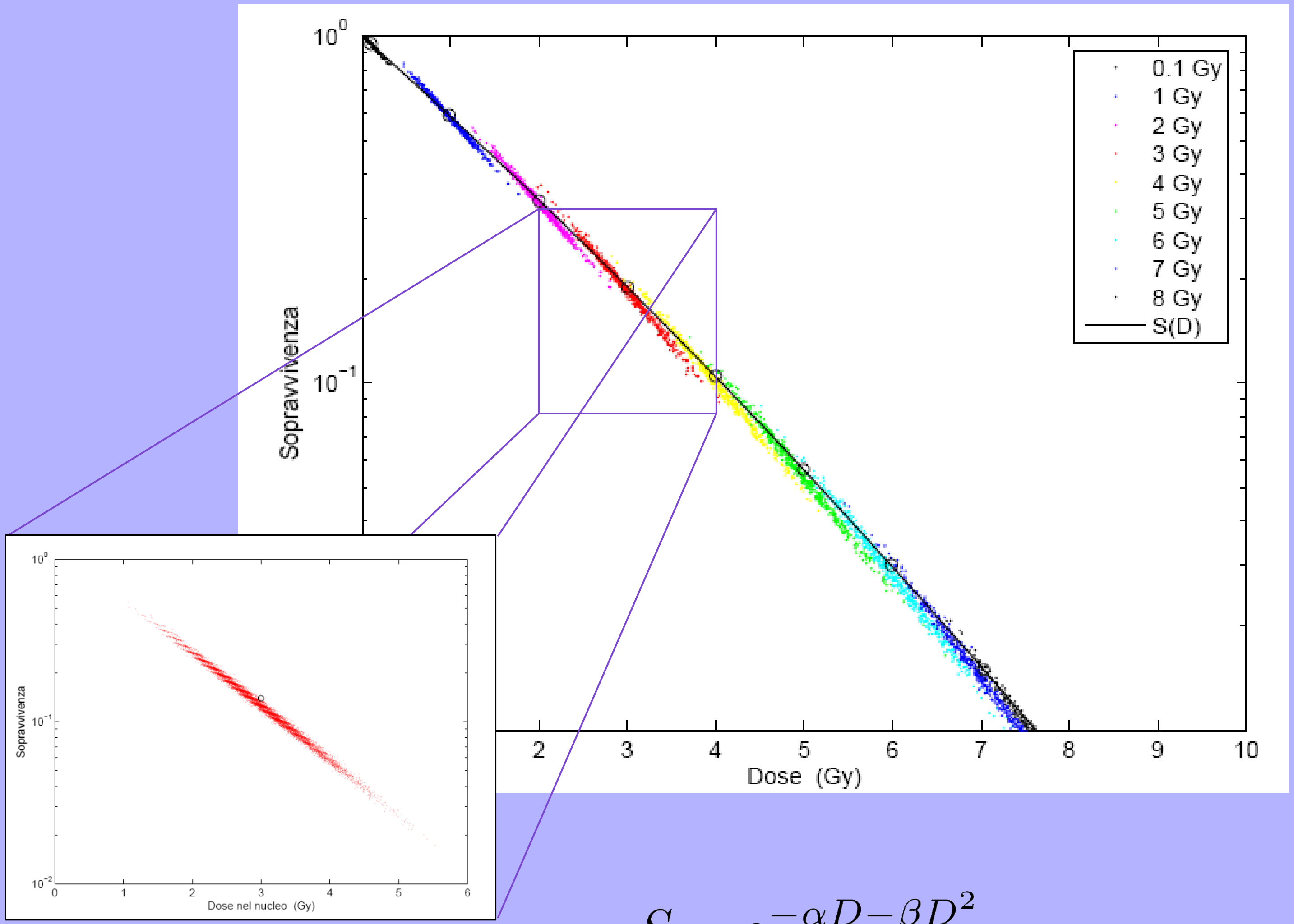


Implementation of the LEM | Statistical fluctuations



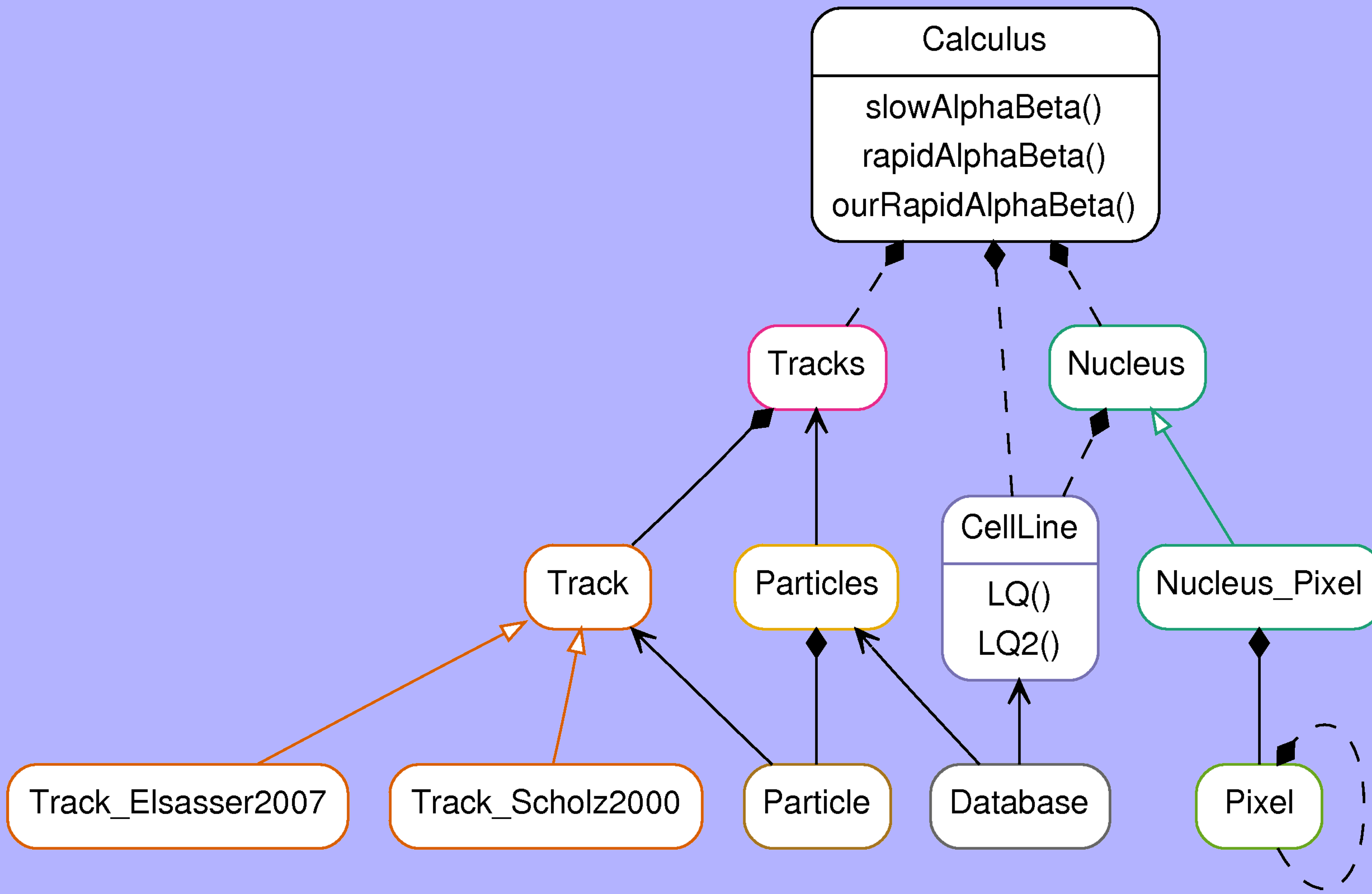


Implementation of the LEM | Survival curve



$$S = \mathrm{e}^{-\alpha D - \beta D^2}$$

Implementation of the LEM | C++ classes



MySQL database structure

Meta data

Simulations_MC
ID
name
tool
toolVersion
phantom
primaryType
e_i
description
author
particles_table
dose_table

Runs_MC
ID
simulation_MC
eventNumber
date
description

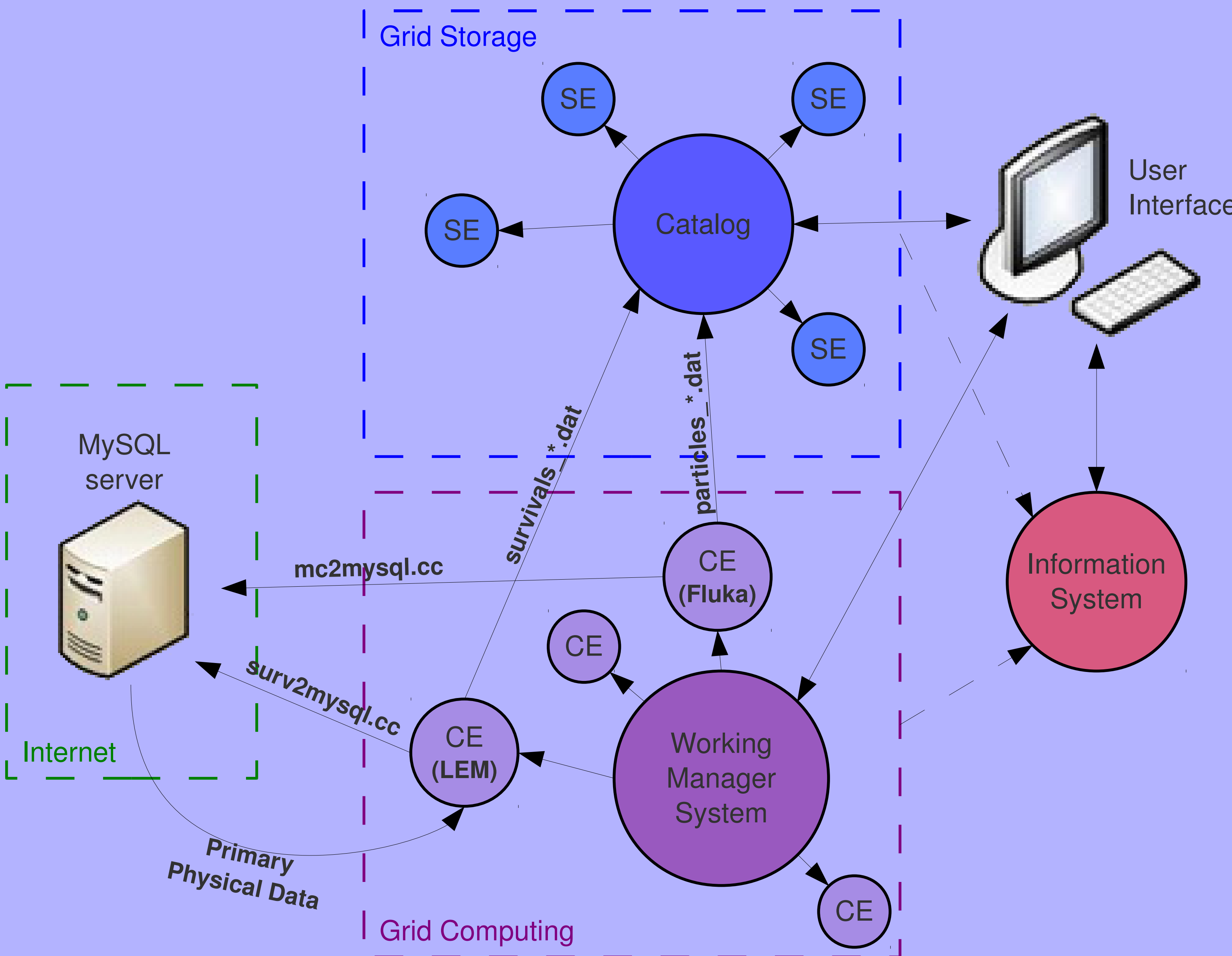
Primary Physical Data

Particles_*
ID
run_MC
simulation_MC
eventID
trackID
parentID
depth
type
charge
A
restEnergy
e_c
e_d
x_in
y_in
z_in
x_out
y_out
z_out

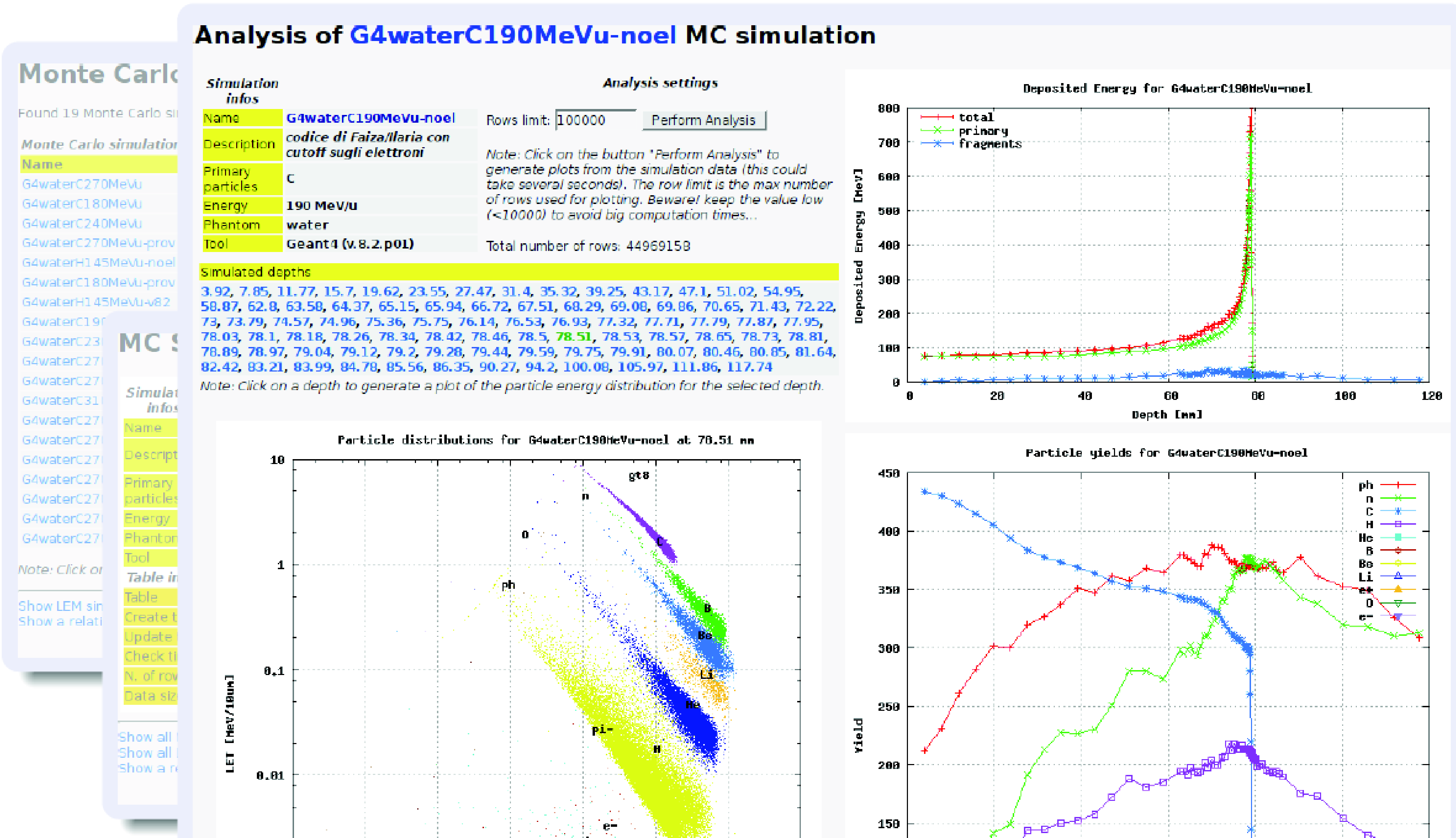
Pencil Beam Physical Data

Dose_*
ID
simulation_MC
e_i
depth
geometry
r_min
r_max
x_min
x_max
y_min
y_max
dose
deltaDose

Grid computations

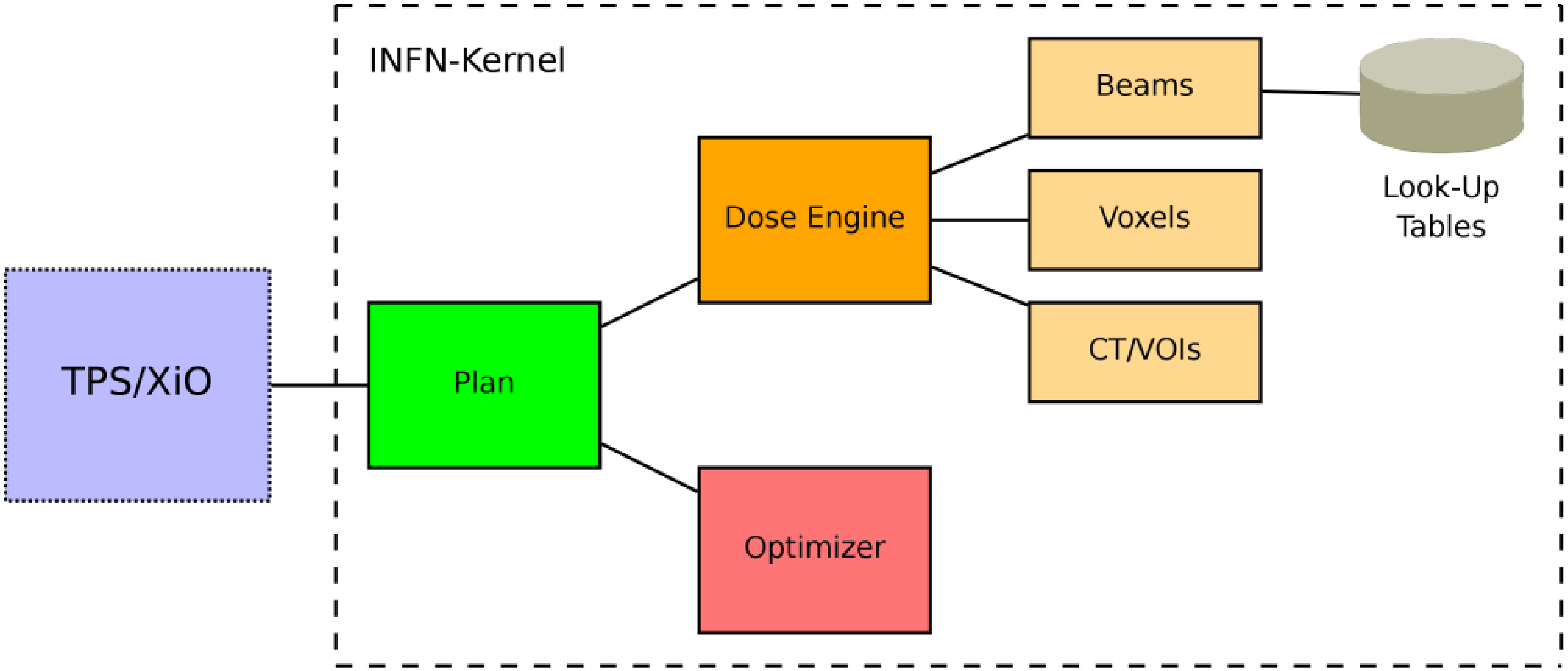


Database web interface



http://totlxl.to.infn.it/services/simulations_database_MC.php

TPS Kernel | High level structure



Look-up Tables: physical and biological description of elementary beams effectiveness in water;

Dose Engine: module responsible of the evaluation of the effect on the patient (in terms of dose and survival levels) for a fully specified irradiation pattern;

Optimizer: module in charge of finding the optimal irradiation pattern, in order to satisfy medical doctor's prescription;

Plan: module that manages the interaction with the external world and sets up Dose Engine and Optimizer;

XiO: module coded by the Elekta company which connects with the Oncological Information System and provides the GUI.

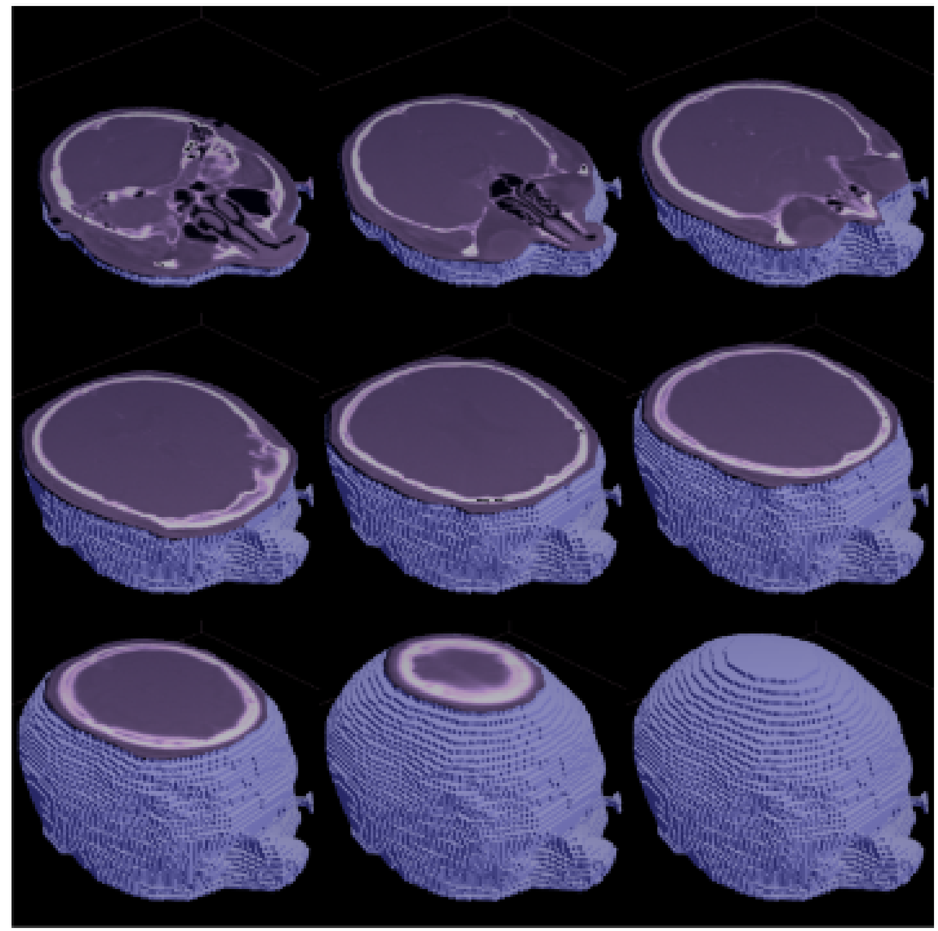
TPS Kernel | High level structure

CT meningioma;

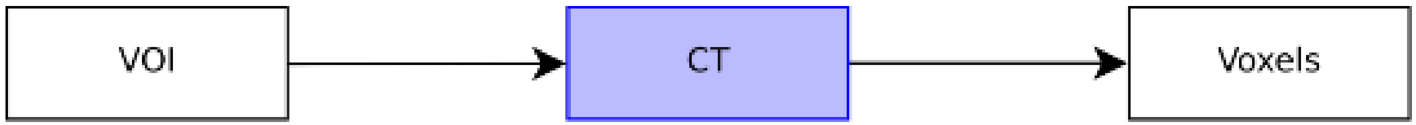
meningioma.LoadCT(*filename*);

meningioma.N_i' $\{N_j', N_k'\}$;
meningioma.dx' $\{dy', dz'\}$;
meningioma.x_0' $\{y_0', z_0'\}$;

meningioma.Hounsfield(i', j', k'); $\rightarrow H_{i'j'k'}$
meningioma.Density(i', j', k'); $\rightarrow \rho_{i'j'k'}$



TPS Kernel | High level structure

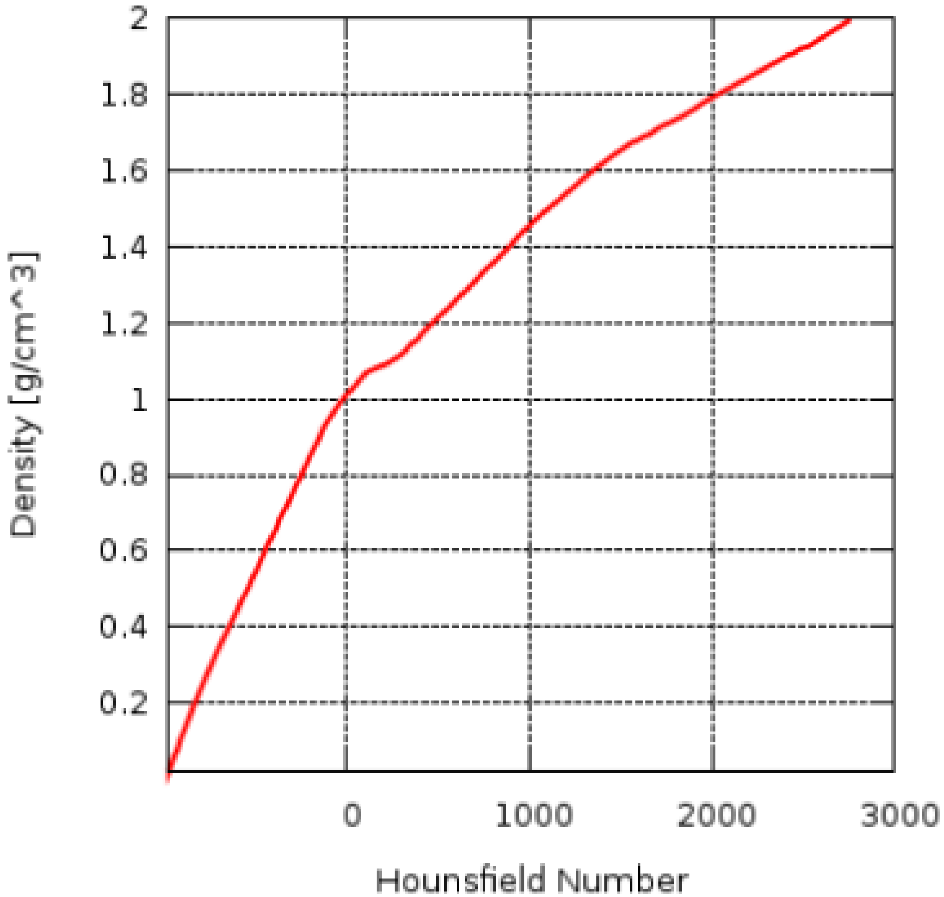


CT meningioma;

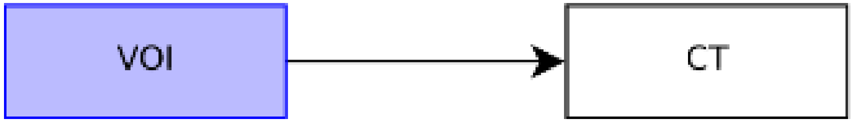
meningioma.LoadCT(*filename*);

meningioma.N'_i $\{N'_j, N'_k\}$;
meningioma.dx' $\{dy', dz'\}$;
meningioma.x'_0 $\{y'_0, z'_0\}$;

meningioma.Hounsfield(i', j', k'); $\rightarrow H_{i'j'k'}$
meningioma.Density(i', j', k'); $\rightarrow \rho_{i'j'k'}$



TPS Kernel | High level structure

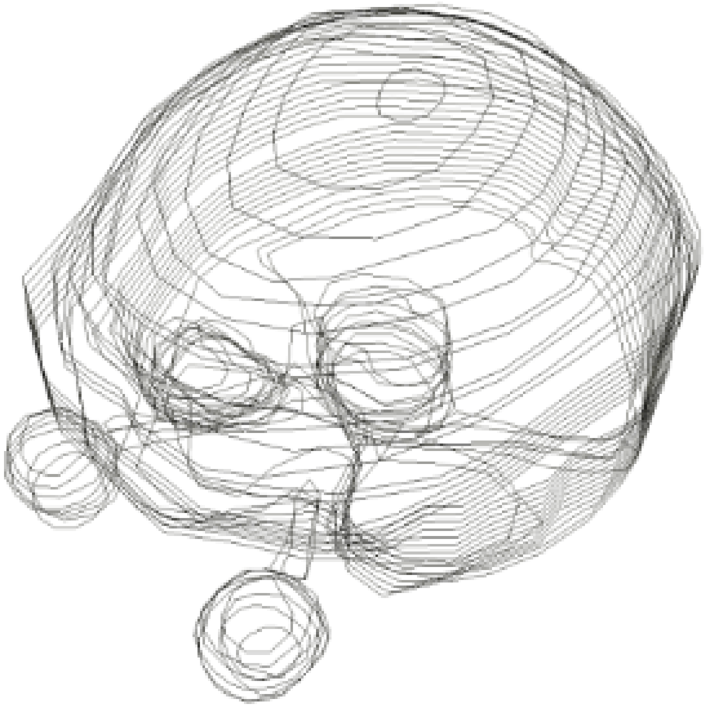


meningioma.LoadVOIs(*file1*, *file2*);
memingioma.NVOI;
memingioma.vois[*n*];

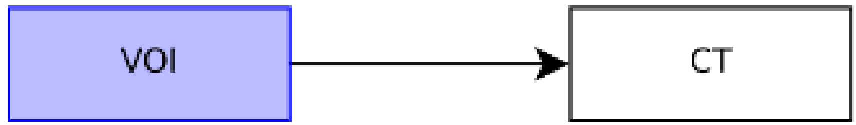
meningioma.vois[n] -> Name;
meningioma.vois[*n*] -> NPoints;
meningioma.vois[*n*] -> *x*[*i*] {*y*[*i*], *z*[*i*]};
meningioma.vois[*n*] -> Transform(****R*, **T*);

meningioma.vois[*n*] -> indexSlice;
meningioma.vois[*n*] -> zSlice;

meningioma.vois[*n*] -> IsIn(x', y', z'); → bool



TPS Kernel | High level structure

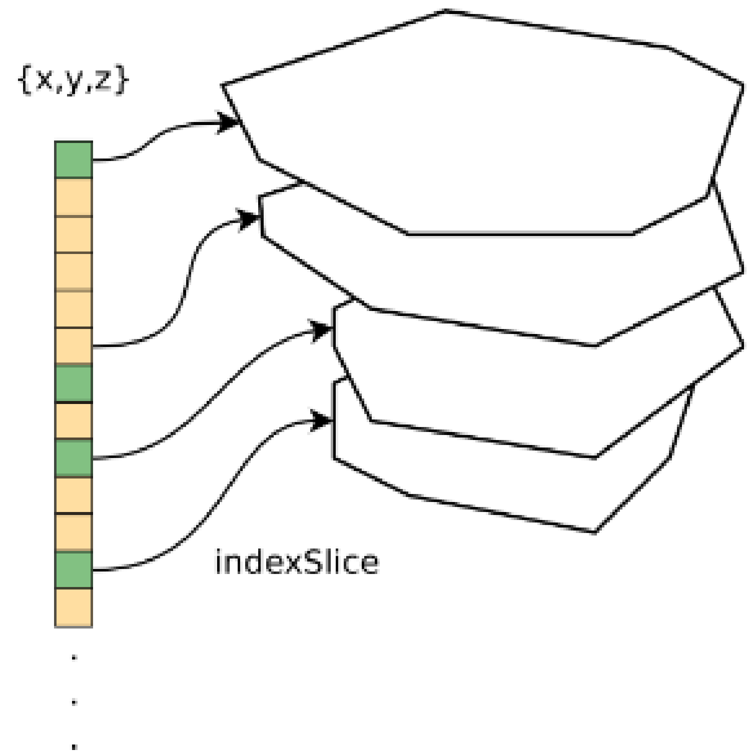


meningioma.LoadVOIs(*file1*, *file2*);
memingioma.NVOI;
memingioma.vois[*n*];

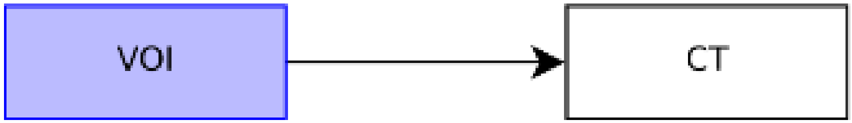
meningioma.vois[n] -> Name;
meningioma.vois[*n*] -> NPoints;
meningioma.vois[*n*] -> *x*[*i*] {*y*[*i*], *z*[*i*]};
meningioma.vois[*n*] -> Transform(****R*, **T*);

meningioma.vois[*n*] -> indexSlice;
meningioma.vois[*n*] -> zSlice;

meningioma.vois[*n*] -> IsIn(x', y', z'); → bool



TPS Kernel | High level structure

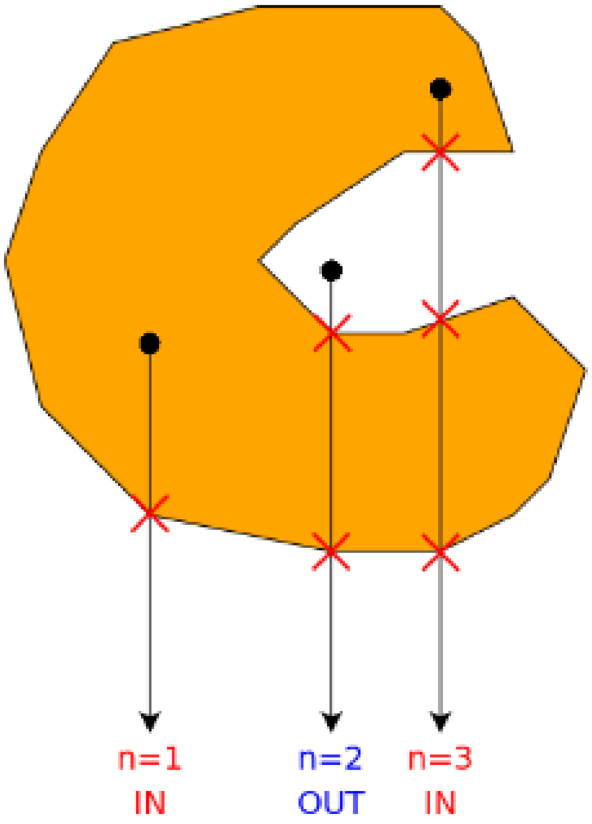


meningioma.LoadVOIs(*file1*, *file2*);
memingioma.NVOI;
memingioma.vois[*n*];

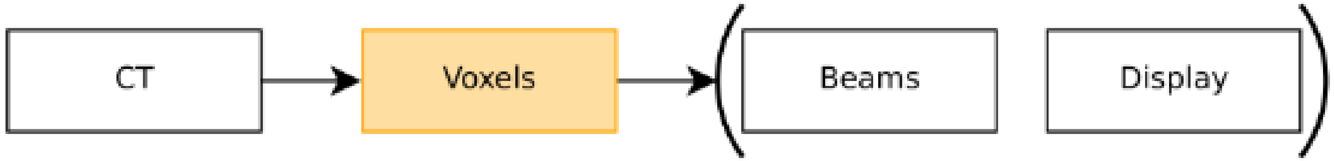
meningioma.vois[n] -> Name;
meningioma.vois[*n*] -> NPoints;
meningioma.vois[*n*] -> *x*[*i*] {*y*[*i*], *z*[*i*]};
meningioma.vois[*n*] -> Transform(****R*, **T*);

meningioma.vois[*n*] -> indexSlice;
meningioma.vois[*n*] -> zSlice;

meningioma.vois[*n*] -> IsIn(x', y', z'); $\rightarrow$ bool



TPS Kernel | High level structure



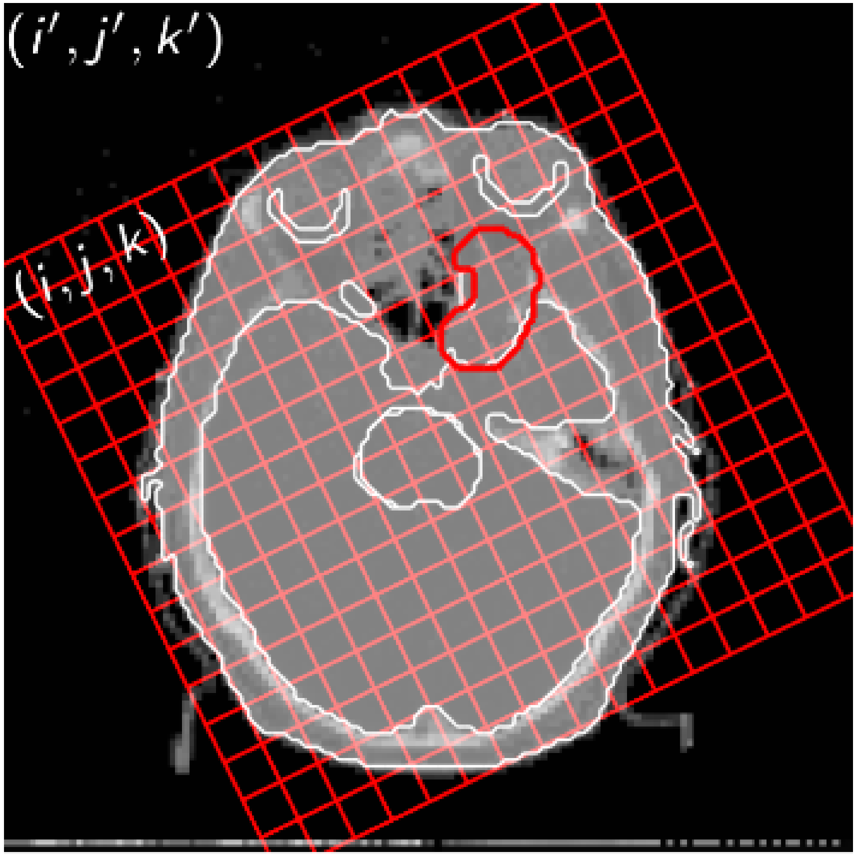
Voxels myVoxels;

myVoxels.CreateArrays(N_i, N_j, N_k);

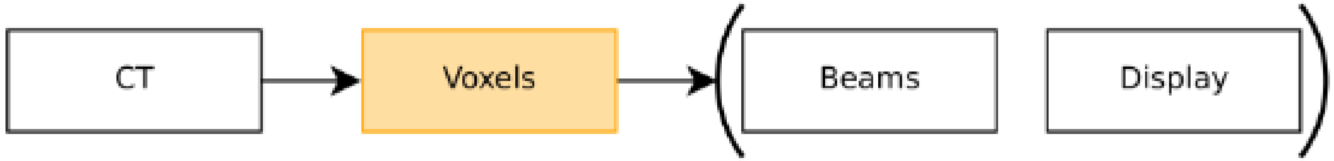
myVoxels.Density[i][j][k]; $\longrightarrow \rho_{ijk}$
myVoxels.Dose[i][j][k]; $\longrightarrow D_{ijk}$
myVoxels.Alpha[i][j][k]; $\longrightarrow \alpha_{ijk}$
myVoxels.Beta[i][j][k]; $\longrightarrow \beta_{ijk}$
myVoxels.VOIs[i][j][k]; $\longrightarrow$ *binary*

myVoxels.SetCT(meningioma);
myVoxels.SetReferenceFrame($***R'$,$*T'$);
myVoxels.SetDimensions(dx, dy, dz);
myVoxels.EvaluateCT($samples$);

myVoxels.EvaluateVOIs;



TPS Kernel | High level structure



Voxels myVoxels;

myVoxels.CreateArrays(N_i, N_j, N_k);

myVoxels.Density[i][j][k]; $\rightarrow \rho_{ijk}$
myVoxels.Dose[i][j][k]; $\rightarrow D_{ijk}$
myVoxels.Alpha[i][j][k]; $\rightarrow \alpha_{ijk}$
myVoxels.Beta[i][j][k]; $\rightarrow \beta_{ijk}$
myVoxels.VOIs[i][j][k]; $\rightarrow$ *binary*

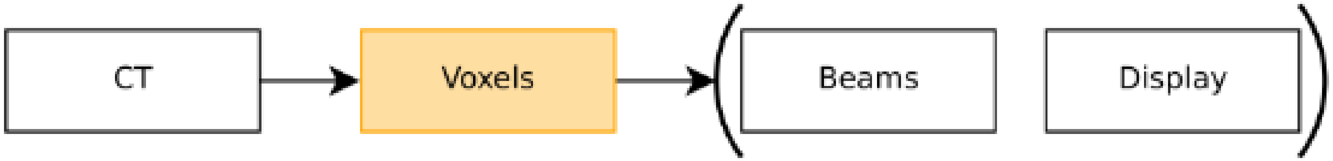
myVoxels.SetCT(meningioma);
myVoxels.SetReferenceFrame(***R',*T');
myVoxels.SetDimensions(dx, dy, dz);
myVoxels.EvaluateCT($samples$);

myVoxels.EvaluateVOIs;

Direct evaluation:

- Voxels::SetDirectCT(&CT) {
  myCT = &CT;
  SetDimensions(myCT->dx, myCT->dy, myCT->dy);
  SetReferenceFrame($\mathbb{I}$, $\mathbf{0}$);
  CreateArrays(myCT->N_i, myCT->N_j, myCT->N_k);
  EvaluateCT(1);
  return;
  }

TPS Kernel | High level structure



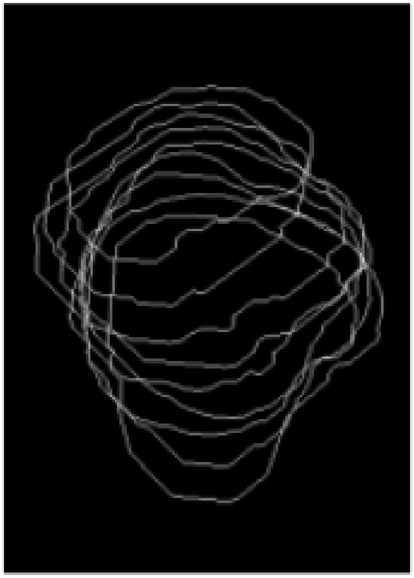
Voxels myVoxels;

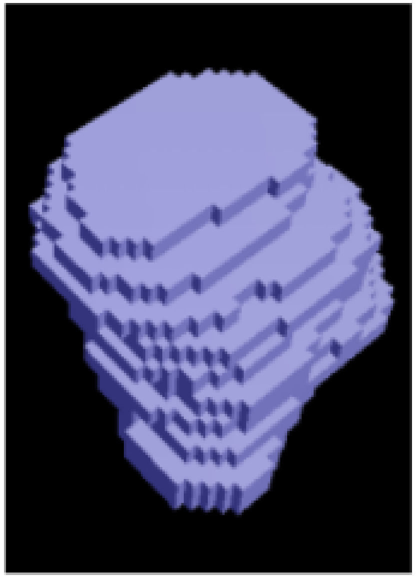
myVoxels.CreateArrays(N_i, N_j, N_k);

myVoxels.Density[i][j][k]; $\longrightarrow \rho_{ijk}$
myVoxels.Dose[i][j][k]; $\longrightarrow D_{ijk}$
myVoxels.Alpha[i][j][k]; $\longrightarrow \alpha_{ijk}$
myVoxels.Beta[i][j][k]; $\longrightarrow \beta_{ijk}$
myVoxels.VOIs[i][j][k]; $\longrightarrow$ *binary*

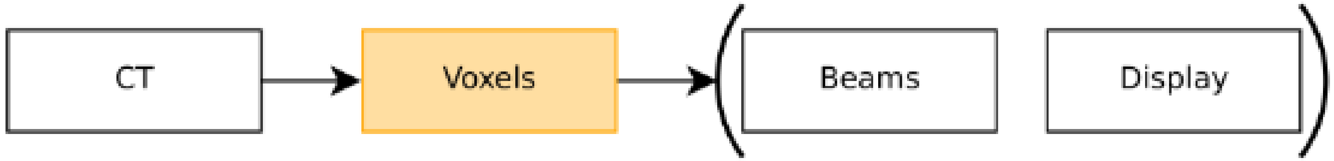
myVoxels.SetCT(meningioma);
myVoxels.SetReferenceFrame($***R', *T'$);
myVoxels.SetDimensions(dx, dy, dz);
myVoxels.EvaluateCT($samples$);

myVoxels.EvaluateVOIs;





TPS Kernel | High level structure



Voxels myVoxels;

myVoxels.CreateArrays(N_i, N_j, N_k);

myVoxels.Density[i][j][k]; $\rightarrow \rho_{ijk}$
myVoxels.Dose[i][j][k]; $\rightarrow D_{ijk}$
myVoxels.Alpha[i][j][k]; $\rightarrow \alpha_{ijk}$
myVoxels.Beta[i][j][k]; $\rightarrow \beta_{ijk}$
myVoxels.VOIs[i][j][k]; $\rightarrow$ *binary*

myVoxels.SetCT(meningioma);
myVoxels.SetReferenceFrame($***R', *T'$);
myVoxels.SetDimensions(dx, dy, dz);
myVoxels.EvaluateCT($samples$);

myVoxels.EvaluateVOIs;

VOIs

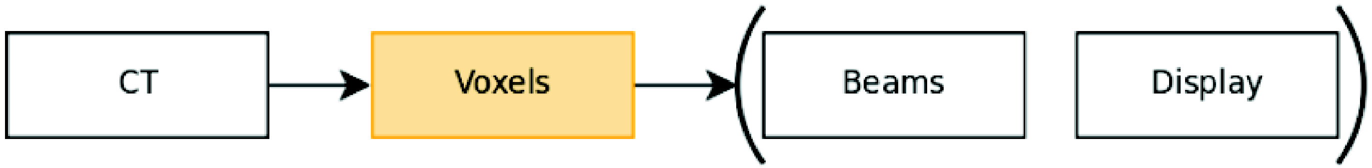
Values:

- voi[0] = 00000001
- voi[1] = 00000010
- voi[2] = 00000100
- [...]

Bitwise functions examples:

- Voxels::PutVOI(n, i, j, k) {
  VOIs[i][j][k] |= voi[n];
  }
- Voxels::IsVOI(n, i, j, k) {
  return(VOIs[i][j][k] & voi[n]);
  }

TPS Kernel | High level structure



Raytracing:

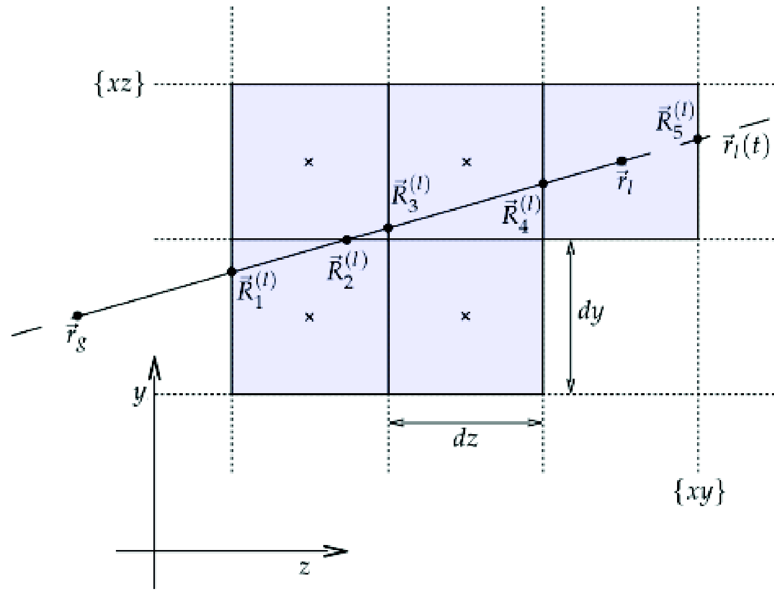
Member Functions

Logical Output:

- $\text{Voxel}[i][j][k]\text{.Tissue} \rightarrow T_{ijk}$
- $\text{Voxel}[i][j][k]\text{.Organs} \rightarrow \text{binary}$

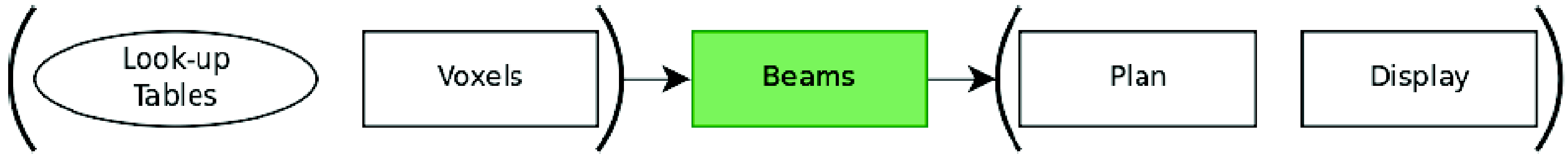
Ray-Tracing:

- $\text{Voxels.SetRay}(\vec{r_g}, \vec{\theta}) \rightarrow n$
- $\text{Voxels.GetIntersections}(n) \rightarrow \{\vec{R}_p^{(n)}\}$
- $\text{Voxels.WEPL}(n, r) \rightarrow WEPL^{(n)}(r)$



$$WEPL^{(n)}(r) = \sum_{p<p_r} |\vec{R}_{p+1}^{(n)} - \vec{R}_p^{(n)}| \rho_{i_p j_p k_p}$$

TPS Kernel | High level structure



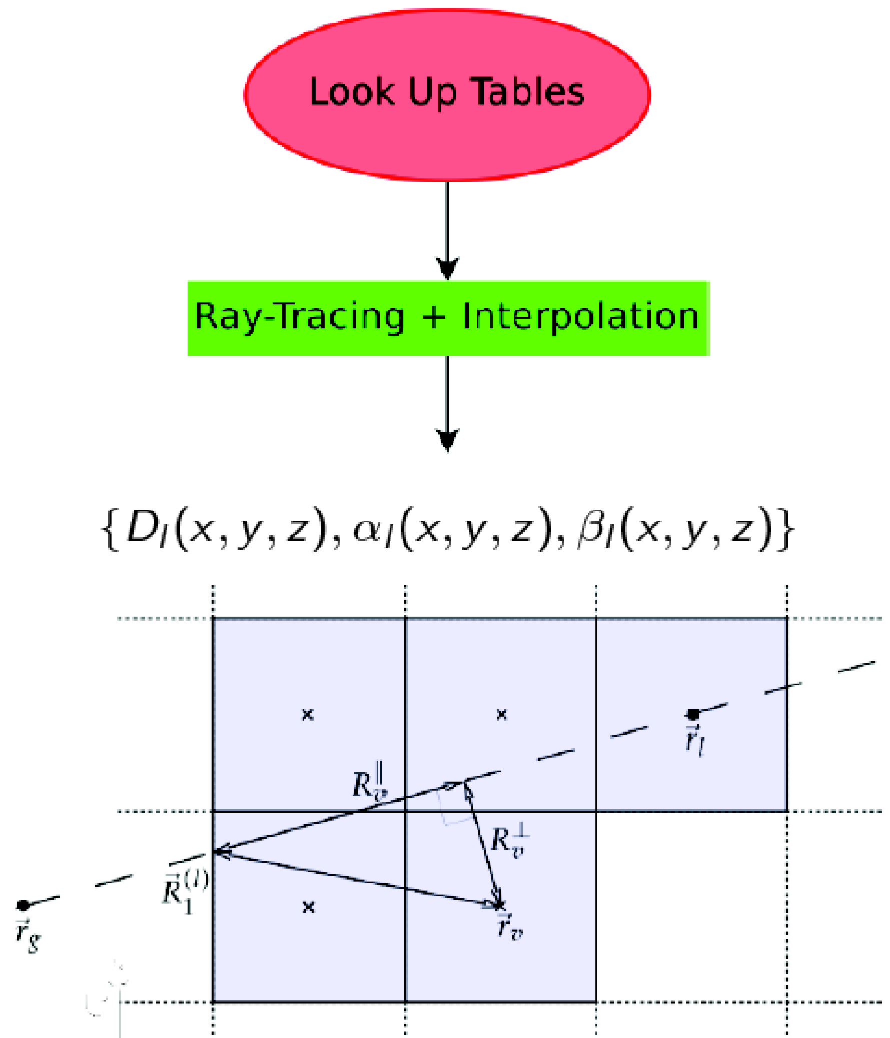
Member Functions

Setting:

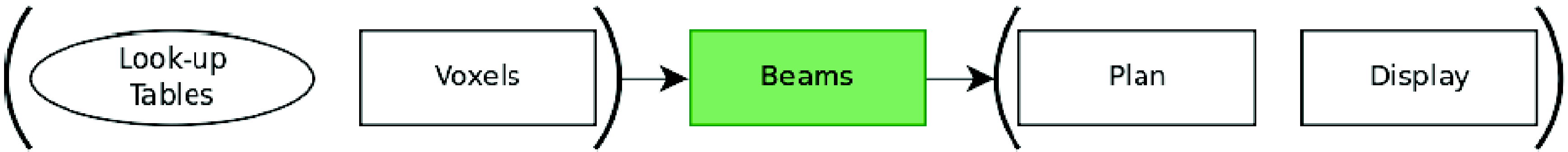
- Beams.SetVoxels(&voxels)
- Beams.CreateBeam() $\rightarrow I$
- Beam[I].SetEnergy(E)
- Beam[I].SetDirection($\vec{g}, \vec{r}$)
- Beam[I].SetFluence(ϕ)

Physical and Biological model (using look-up tables):

- Beam[I].Dose(x, y, z) $\rightarrow D_I(x, y, z)$
- Beam[I].Alpha(x, y, z) $\rightarrow \alpha_I(x, y, z)$
- Beam[I].Beta(x, y, z) $\rightarrow \beta_I(x, y, z)$
- Beam[I].SetEnergyForSpot($\vec{r_g}, \vec{r_s}$) $\rightarrow E$



TPS Kernel | High level structure



Member Functions

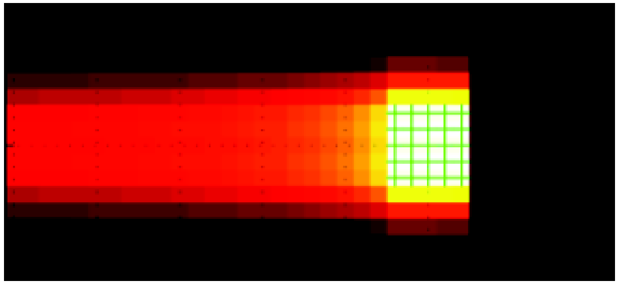
Evaluation on Voxels:

- Beams.SetDoseCutoff(D_c)
- Beams.EvaluateNetDose()
- Beams.EvaluateNetAlpha()
- Beams.EvaluateNetBeta()

↓

Voxels Data Structure

- Voxel[i][j][k].Dose
- Voxel[i][j][k].Alpha
- Voxel[i][j][k].Beta



$$D_{ijk} = \sum_l D_{ijkl}$$

$$\alpha_{ijk} = \frac{\sum_l \alpha_{ijkl} D_{ijkl}}{D_{ijk}}$$

$$\beta_{ijk} = \frac{\left(\sum_l \sqrt{\beta_{ijkl}} D_{ijkl}\right)^2}{D_{ijk}^2}$$

TPS Kernel | High level structure

Treatment Specifications

VOI	Requirements	
PTV	Prescription dose	80 Gy
	Maximum dose	82 Gy
	Minimum dose	78 Gy
	95% of volume	$\geqslant$ 79 Gy
Rectum	Maximum dose	76 Gy
	70% of volume	$\leqslant$ 32 Gy
Bladder	Maximum dose	78 Gy
	70% of volume	$\leqslant$ 32 Gy

$\Downarrow$

Constraints
Matrix

TPS Kernel | High level structure

Treatment Specifications

⇓

Constraint Matrix

λ	VOI name	D_λ^{max} [Gy]	$v_\lambda^{(u)}$ [%]	$w_\lambda^{(u)}$	D_λ^{min} [Gy]	$v_\lambda^{(l)}$ [%]	$w_\lambda^{(l)}$
1	PTV	80	100	0.9	80	100	1.0
2	PTV	82	100	0.9	78	100	1.0
3	PTV	-	-	-	79	95	1.0
4	rectum	76	100	1.0	-	-	-
5	rectum	32	70	1.0	-	-	-
6	bladder	78	100	1.0	-	-	-
7	bladder	32	70	1.0	-	-	-

TPS Kernel | High level structure

Constraints
Matrix

$$\Downarrow$$

$$\chi^2(\phi) = \sum_{\lambda} \left(w_{\lambda}^{(u)} \sum_{ijk \in \mathrm{VOI}_{\lambda}(v_{\lambda}^{(u)})} \max(0, D_{ijk} - D_{\lambda}^{\max})^2 + w_{\lambda}^{(l)} \sum_{ijk \in \mathrm{VOI}_{\lambda}(v_{\lambda}^{(l)})} \max(0, D_{\lambda}^{\min} - D_{ijk})^2 \right)$$

"Biological" Cost Function:

$$D_{ijk} \rightarrow D_{ijk}^{\mathrm{bio}} = D_{ijk} \times \mathrm{RBE}_{ijk}, \quad D_{ijk}^{\mathrm{bio}} = \frac{-\alpha_X + \sqrt{\alpha_X^2 + 4\beta_X N_{ijk}^{\mathrm{leth}}}}{2\beta_X}, \quad N_{ijk}^{\mathrm{leth}} = \alpha_{ijk} D_{ijk} + \beta_{ijk} D_{ijk}^2$$

TPS Kernel | High level structure

Member Functions

Setting:

- Optimization.SetContraints(&Constraints)
- Optimization.SetCostFunction(&CostFunction)

Evaluation:

- Optimization.ConjugateGradient() (GSL)
- Optimization.Simplex() (GSL)
- Optimization.SimulatedAnnealing()
- Optimization.ParallelTempering()
- Optimization.GeneticAlgorithm()
- [...]

Minimization:

- Global minimum
- Non-linear
- Multi-dimensional

$$N\{\phi_I\} \sim 10^4$$

(ϕ_I:= beams fluences)

TPS Kernel | High level structure

Member Functions

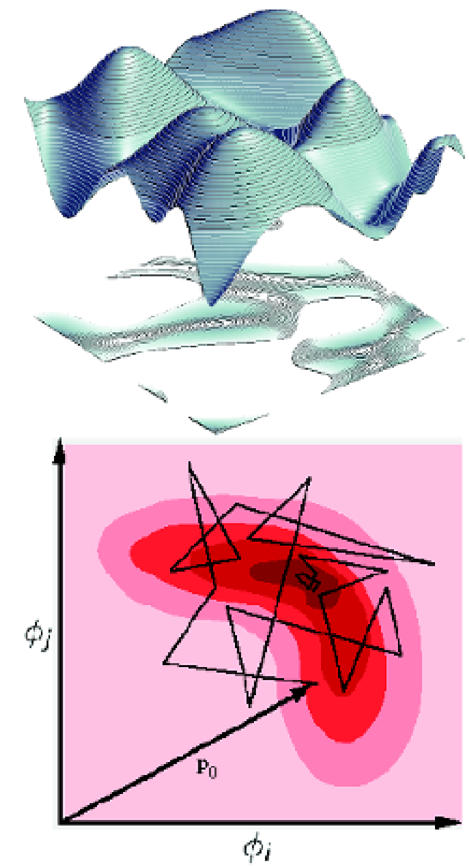
Setting:

- Optimization.SetContraints(&Constraints)
- Optimization.SetCostFunction(&CostFunction)

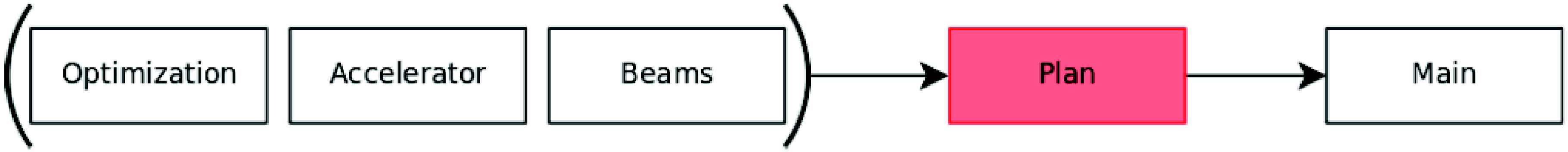
Evaluation:

- Optimization.ConjugateGradient() (GSL)
- Optimization.Simplex() (GSL)
- Optimization.SimulatedAnnealing()
- Optimization.ParallelTempering()
- Optimization.GeneticAlgorithm()
- [...]

$$\frac{d\chi^2}{d\{\phi_i\}} = 0$$



TPS Kernel | High level structure



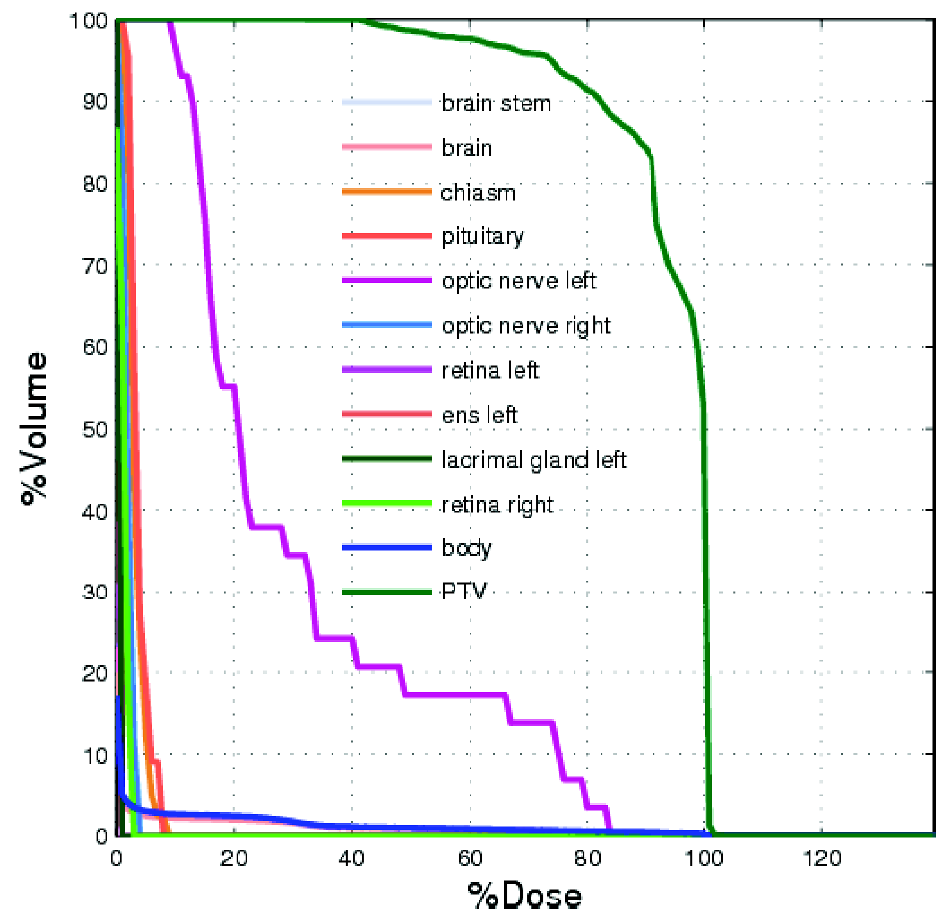
Member Functions

Plan Evaluation:

- Plan.Evaluate()

Output:

- Plan.DVH(%D) $\rightarrow$ %V
- Plan.Dose(i, j, k) $\rightarrow$ %D_{ijk}
  - Voxel[i][j][k].Dose
- Plan.Survival(i, j, k) $\rightarrow$ %S_{ijk}
  - Voxel[i][j][k].Survival
- Plan.*OptimizationInfos*()
- [...]



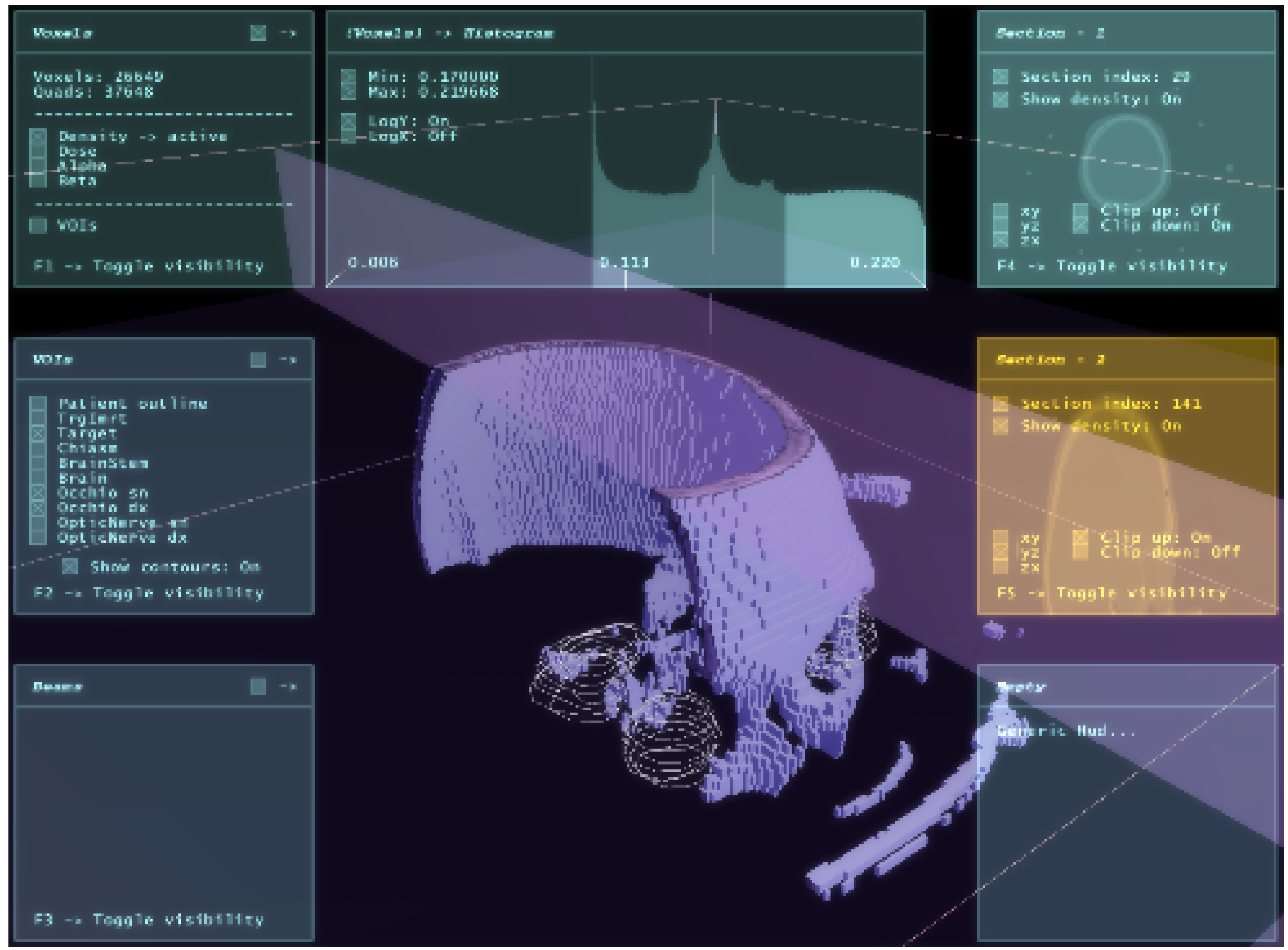
TPS Kernel | High level structure

Display myDisplay;

myDisplay $\ll$ meningioma;
myDisplay $\ll$ myVoxels;
myDisplay $\ll$ myBeams;
myDisplay $\ll$ myPlan;

myDisplay.SetGeometry(s);
myDisplay.ScreenWidth;
myDisplay.ScreenHeight;

myDisplay.BeginIntSession();



Quality Assurance

Q-Spin Audit (28 September 2009):

Quality Management audit for assessing compliance of INFN with the ISO13485:2003 for the development of the calculation module of the Carbon Therapy Treatment Planning calculation module to become an EC/92/42 approved product for CE marking.

$\Downarrow$

Corrective Actions

Q-Spin Training workshop (2-4 November 2009):

- Project Life Cycle
- Project Planning
- Risk Management
- Project Monitoring and Control
- Verification and Validation
- Requirement Management
- Configuration Management
- Issue Management
- Unified Modeling Language (UML)
- Unit Testing
- Software Architecture

Conclusions

Ion therapy represents an alternative to X-ray conventional cancer treatments, particularly suited for treating radio-resistant, deep-seated tumours.

Proton therapy is a well-established technique for treating cancer; its advantages have been demonstrated with clinical trials.

Carbon ion therapy is a promising technique which is expected to provide additional benefits; these benefits have still to be proven, hence the use of carbon ions in the clinical practice is limited.

Italy is doing, together with Germany and Japan, pioneer work with carbon ions, both in the development of a dedicated medical centre and in the realization of a Treatment Planning System.

The work of the INFN-TPS collaboration consists in the development of a commercial TPS, in cooperation with the IBA company.

The TPS has to provide the medical doctor with the evaluation of the effectiveness of different irradiation patterns, allowing him to choose the one to be delivered to the patient.

Conclusions

In addition, the capability of guiding the medical doctor towards a solution that optimizes the adherence with the stated prescription is required.

In the TPS-INFN project, this complex task is achieved pre-computing the effectiveness of the elementary beams in water and then loading and rescaling them depending on the actual patient morphology.

The TPS is being written in C++ and up to now involves the use of Geant4, Fluka, MySQL, GRID, OpenGL, GNU Scientific Libraries, CppUnit.

The requests in terms of maximum time for the evaluation (in the order of minutes) and of big treatment volumes (more than 1 litre) impose a trade-off between efficiency, memory consumption and precision.

Claiming to become a commercial medical product, the TPS must obtain the EC marking. Therefore it has to fulfill the Q&A rules specified in the appropriate ISO standards.

Conclusions

In addition, the capability of guiding the medical doctor towards a solution that optimizes the adherence with the stated prescription is required.

In the TPS-INFN project, this complex task is achieved pre-computing the effectiveness of the elementary beams in water and then loading and rescaling them depending on the actual patient morphology.

The TPS is being written in C++ and up to now involves the use of Geant4, Fluka, MySQL, GRID, OpenGL, GNU Scientific Libraries, CppUnit.

The requests in terms of maximum time for the evaluation (in the order of minutes) and of big treatment volumes (more than 1 litre) impose a trade-off between efficiency, memory consumption and precision.

Claiming to become a commercial medical product, the TPS must obtain the EC marking. Therefore it has to fulfill the Q&A rules specified in the appropriate ISO standards.

THANK YOU!