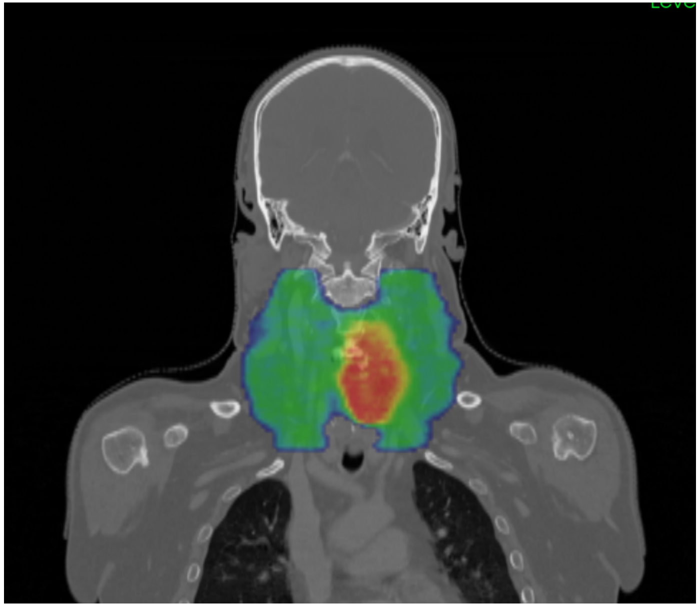
Development of LINAC-Specific Variance Reduction Methods for GATE10

Maxime Jacquet
CREATIS Lyon

GATE activities @Creatis (Lyon)

- Linac and radiotherapy simulation (Maxime, collab IGR)
  - Complete dynamic multileaf motion + variance reduction techniques
- Dose and dose-rate for radionuclide therapy (Laure, Eduardo)
  - ^{177}Lu from SPECT imaging + collaboration with SPECTRUM Veriton CT
- Partial volume correction (Théo)
  - Deep learning trained on simulated database of SPECT images
- SPECT Veriton CT model (Ane)
- PET imaging validation ^{89}Zr (collab. Groeningen, Philip)
- SPECT imaging validation (Nantes, Strasbourg, Caen, Lyon)
- GATE10 (Nils, David)

Out-Of-Field (OOF) dose for the Volume Modulated Arc Therapy (VMAT)



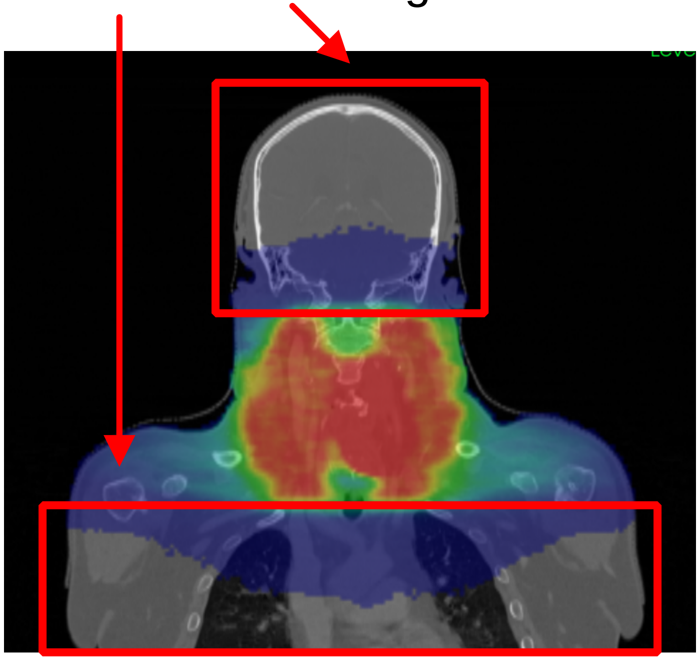
TPS calculation of a VMAT modality

TPS calculation

- Accurate (in-field)
- Fast

Out-Of-Field (OOF) dose for the Volume Modulated Arc Therapy (VMAT)

Potential immune effects to investigate



TPS calculation of a VMAT modality

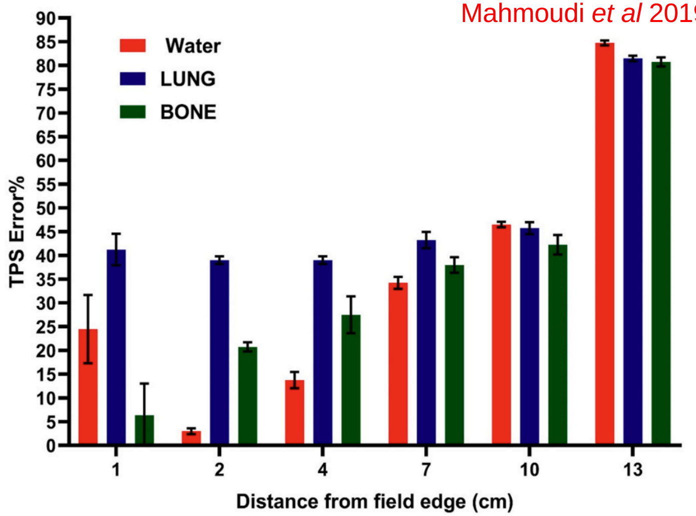
TPS calculation

- Accurate (in-field)
- Fast
- Less precise (OOF)
  - ~ 100 % of error

MC simulations

- Accurate
- Time consuming

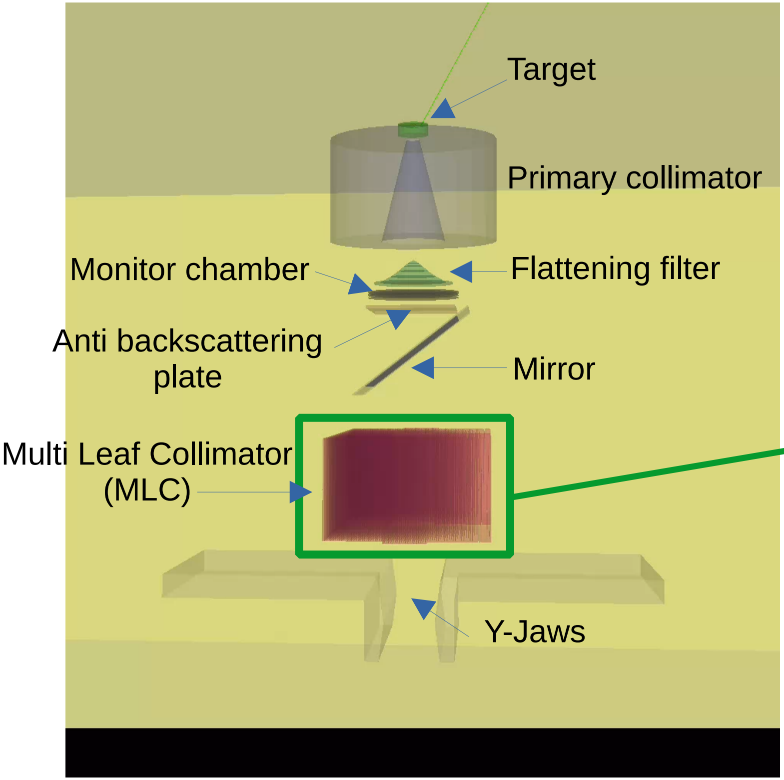
Mahmoudi *et al* 2019



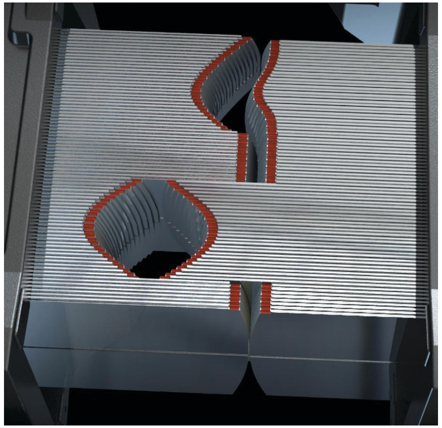
Difference between dose measurements and Monaco TPS predictions

Elekta Versa HD

Elekta Versa HD



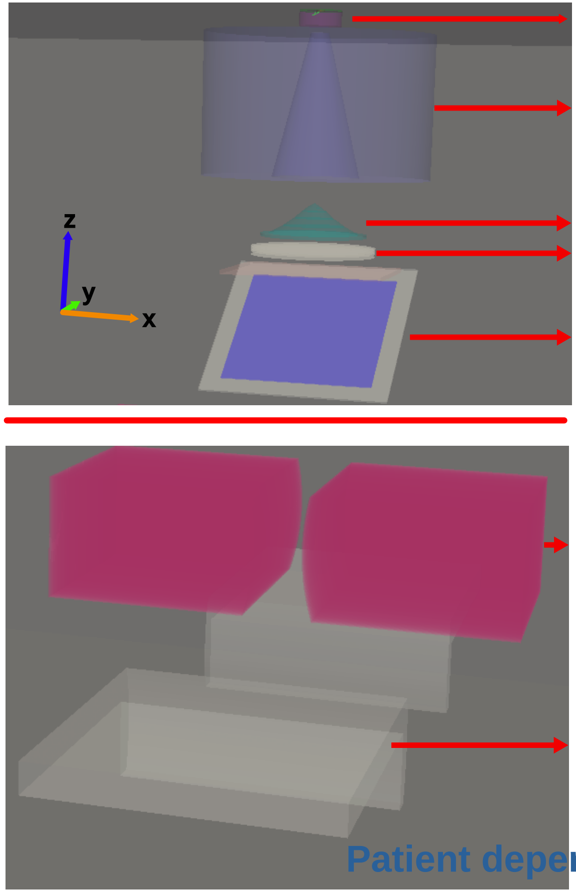
Elekta LINAC VERSA HD 6 MV simulation



Agility MLC

LINAC model building

Patient independent

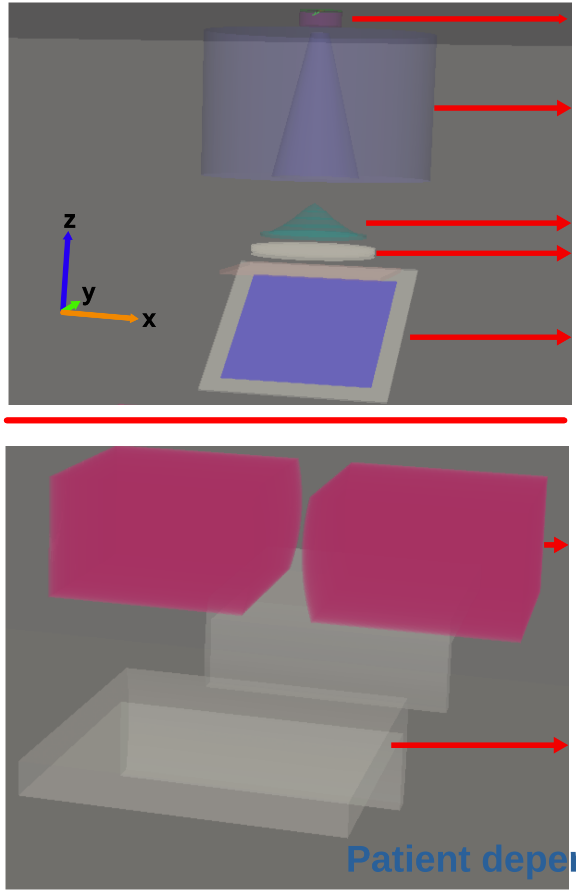


Patient dependent

- Model available :
  - Contrib/linacs/elektaversa
- Two ways to do the simulation :
  - Electron source
    - add_electron_source
  - Phase space source
    - add_phase_space_source

LINAC model building

Patient independent



Patient dependent

Static simulation

Rectangular field

- Leaves and Y-jaws position :
- Field at a given source axis distance

```
def set_rectangular_field(sim, mlc, jaws, x_field, y_field, sad=1000):
```

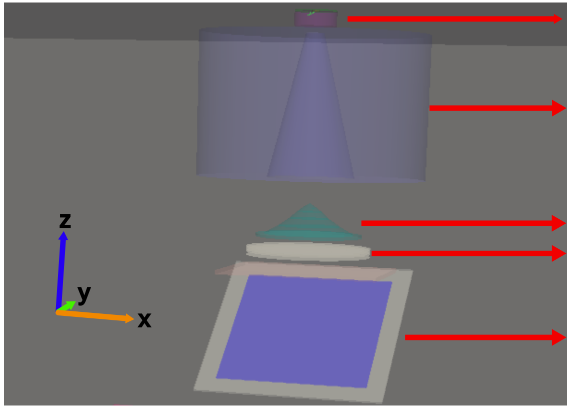
Dynamic simulation

Simulation according to a DICOM RT plan

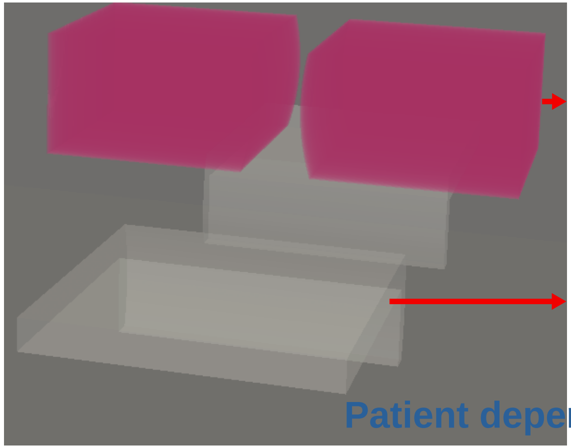
- Parameter of interest : python dictionnary
- One run : One set of parameter
    - Leaves and Y-jaws position
    - Linac head rotation
    - Number of monitor units to deliver

LINAC model building

Patient independent



Phase space source



Patient dependent

Static simulation

Rectangular field

- Leaves and Y-jaws position :
- Field at a given source axis distance

```
def set_rectangular_field(sim, mlc, jaws, x_field, y_field, sad=1000):
```

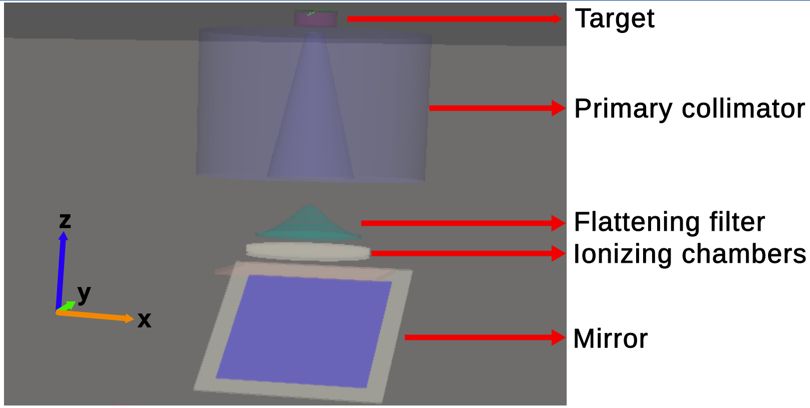
Dynamic simulation

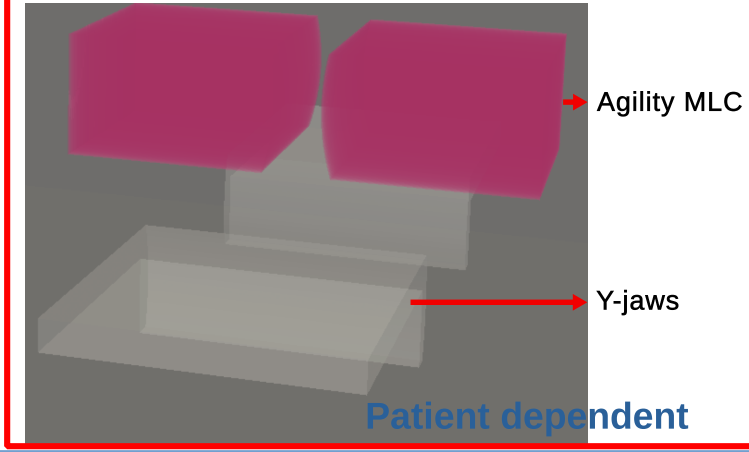
```
def jaw_translation(sim, linac_name, jaw, jaw_positions, side, cp_id="all_cp", sad=1000):...
def mlc_leaves_translation(sim, linac_name, mlc, leaves_position, cp_id="all_cp", sad=1000):...
def set_linac_head_motion(sim, linac_name, jaws, mlc, rt_plan_parameters, cp_id="all_cp", sad=1000):...
def set_time_intervals_from_rtplan(sim, rt_plan_parameters, cp_id="all_cp"):...
```

- Number of monitor units to deliver

LINAC model building

Patient independent

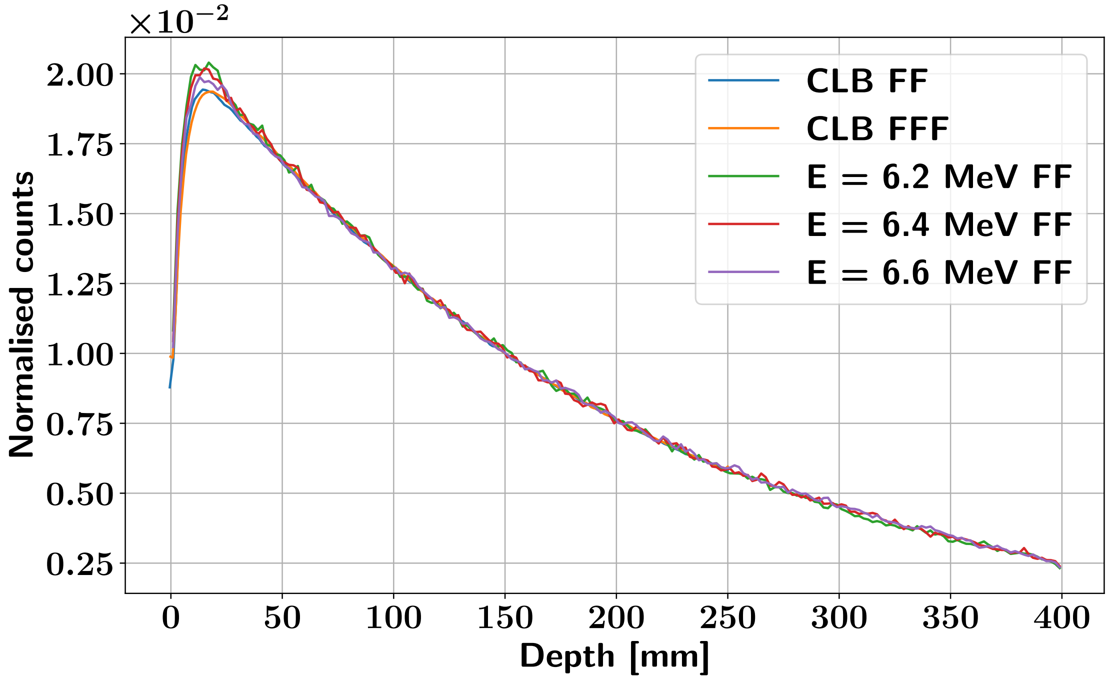




Patient dependent

```
sim = gate.Simulation()
sim.g4_verbose = False
sim.check_volumes_overlap = False
sim.number_of_threads = 1
sim.random_seed = 123456789
sim.check_volumes_overlap = True
mm = gate.g4_units.mm

linac = versa.add_linac(sim, "versa")
source = versa.add_electron_source(sim, linac.name, linac.rotation)
source.n = 8e4 / sim.number_of_threads

jaws = versa.add_jaws(sim, linac.name)
mlc = versa.add_mlc(sim, linac.name)
versa.set_rectangular_field(sim, mlc, jaws, 100*mm,100*mm)

sim.physics_manager.physics_list_name = "G4EmStandardPhysics_option3"
sim.physics_manager.set_production_cut("world", "all", 1 * mm)
versa.enable_brem_splitting(sim, linac.name, splitting_factor=10)
```

In-field validation: PDD

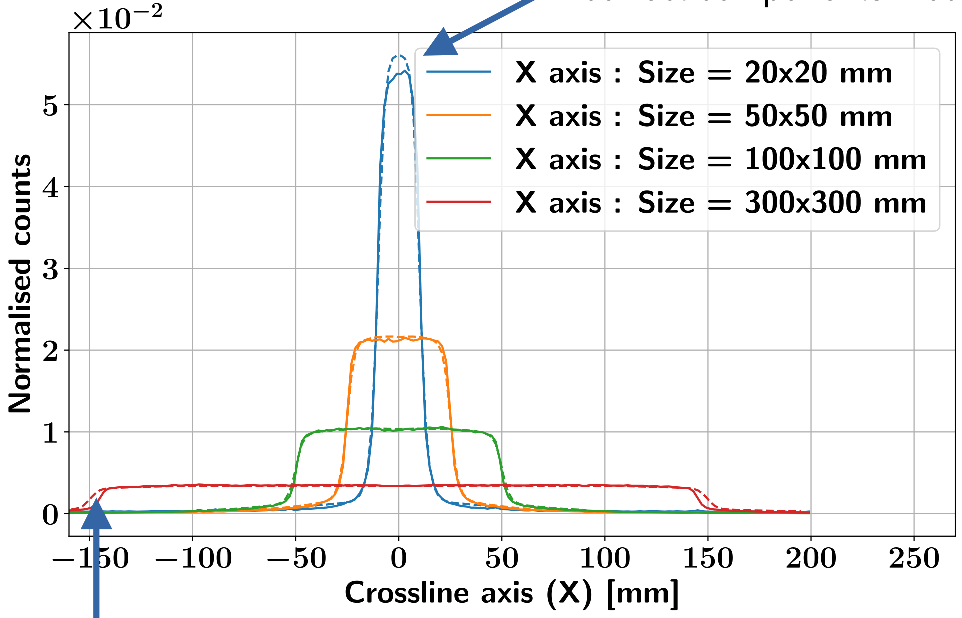


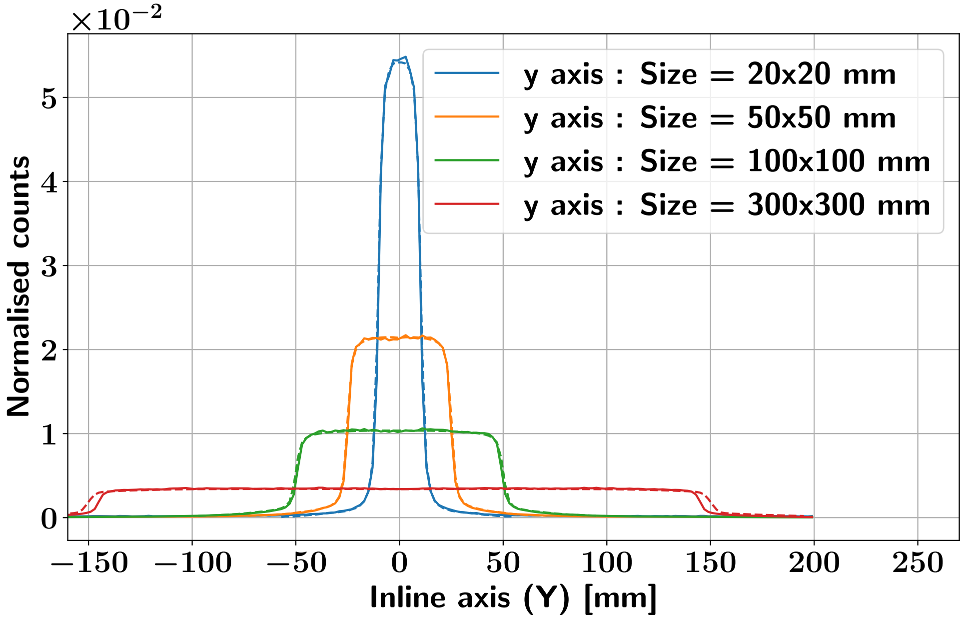
Percent Depth Dose (PDD) profile for a 10 × 10 cm² field

- Fine tuning of electron source energy
  - Correct photon attenuation
  - Discrepancies at the beginning
    - Electron contamination
    - Low energy photon
  - Manufacturer data ...

In-field validation: X and Y profile

Incorrect components model

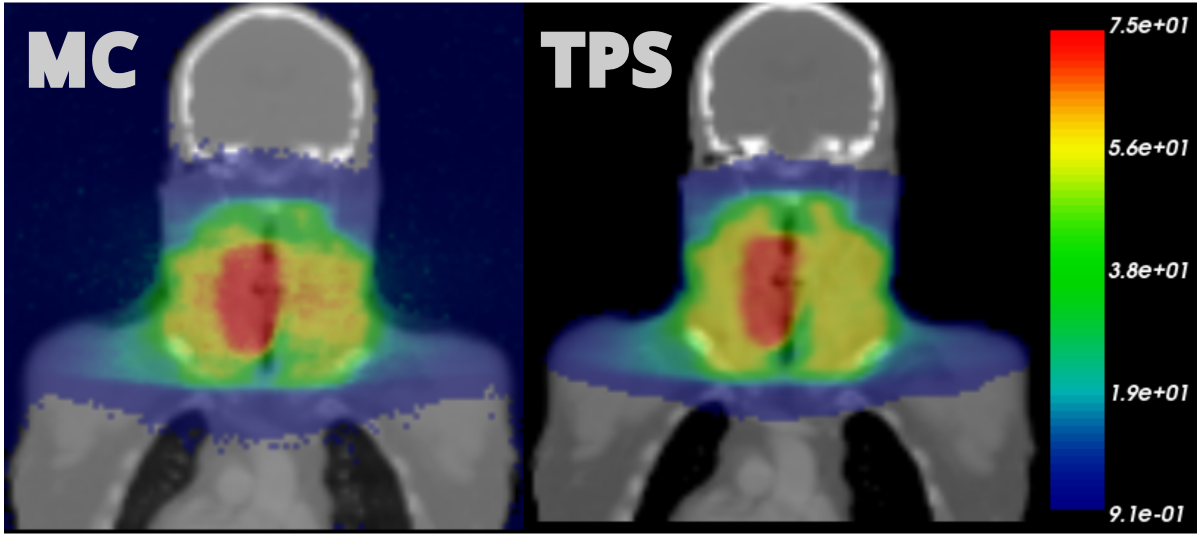




Electron source divergence to tune

X and Y-axis profiles for different irradiation field-of-view

VMAT simulation



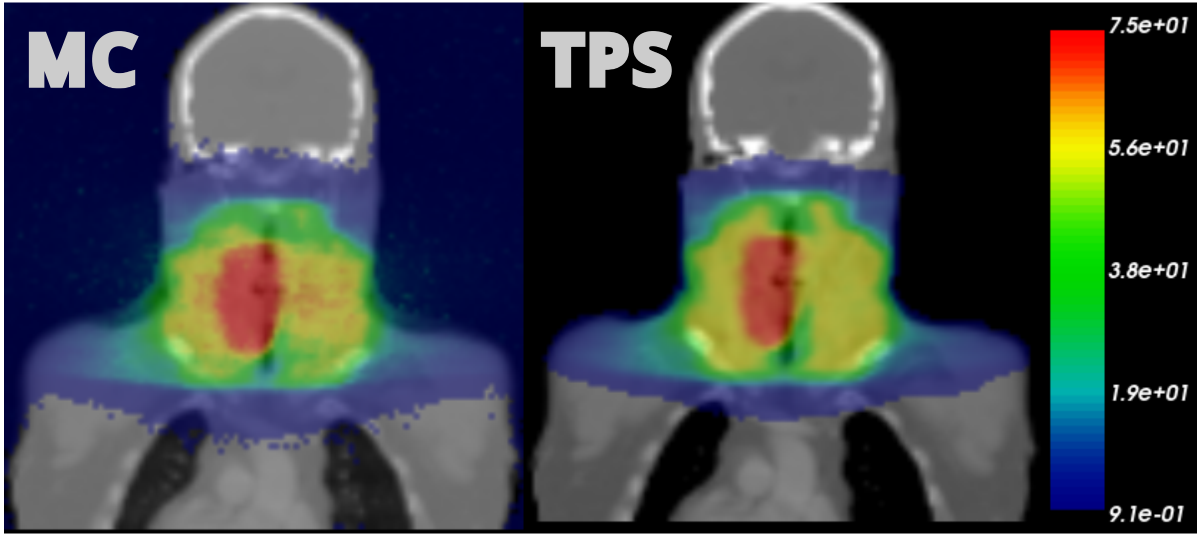
Comparison between VMAT MC simulation and VMAT Monaco calculation

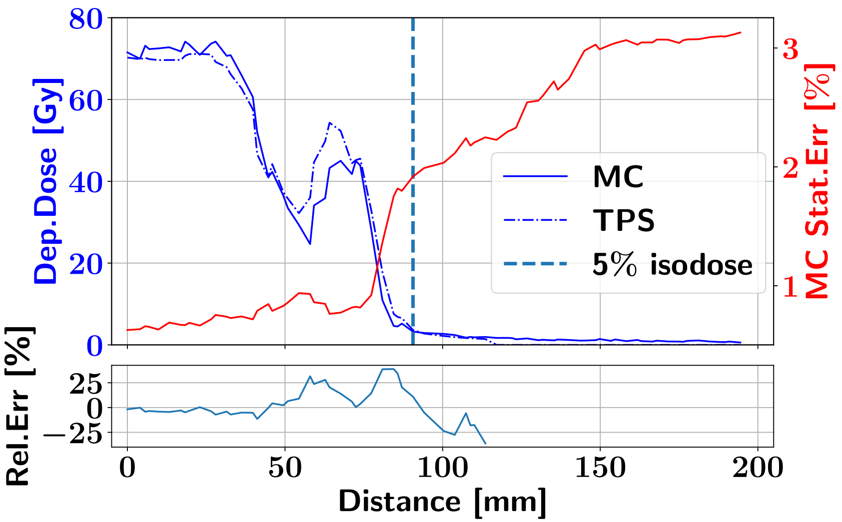
DICOM RT plan

230 runs with defined parameters

- Linac head rotation
- 160 leaves + 2 Y-jaws position
- Number of particle to send

VMAT simulation

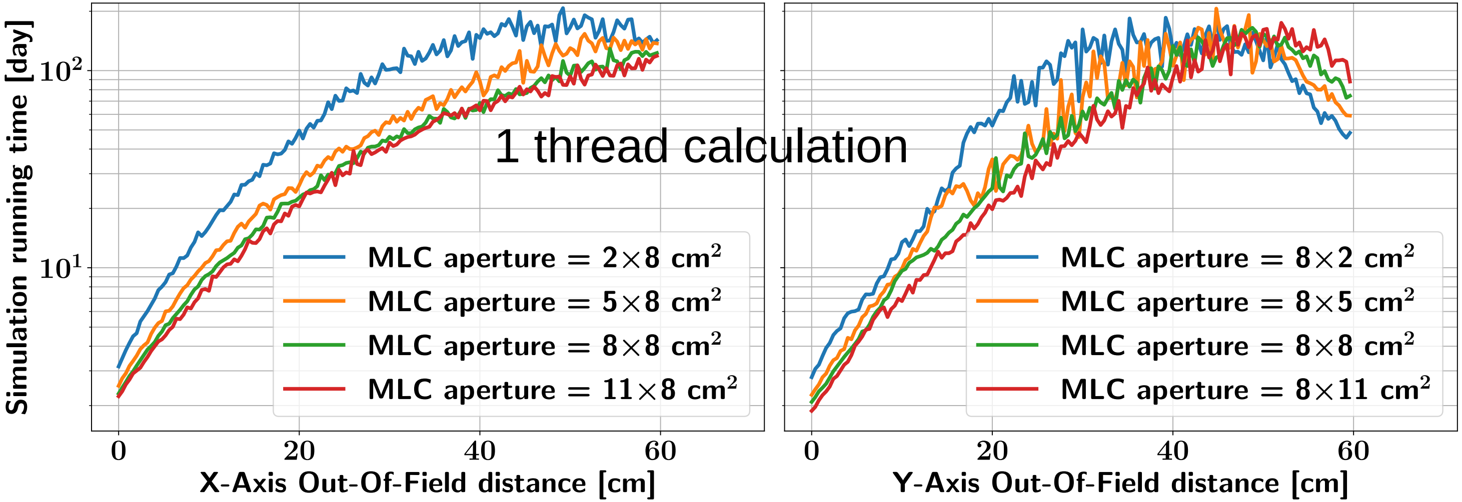




Comparison between VMAT MC simulation and VMAT Monaco calculation

Discrepancies but LINAC model to refine

Simulation running time



Simulation running time to achieve a dD/D of 5 % in the OOF deposited dose

- Far OOF (~40 cm):
  - 5 % of precision
  - ~ 10^{11} photons
  - 50 - 200 days
- If targeted precision = 1 %
  - Running time × 25

~ A day of simulation with hundreds of thread ⇒ Variance reduction methods

Variance Reduction Methods applied (VRM)

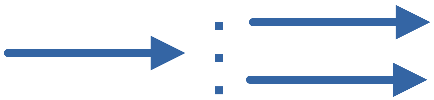
VRM :

Procedure used :

- Increase the accuracy
- For a given computational effort
- Keep the same expectancy

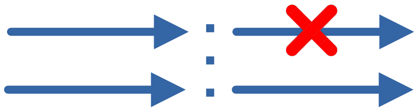
$$\epsilon = \frac{1}{\mathbb{V}[\hat{D}] \times t}$$

Splitting



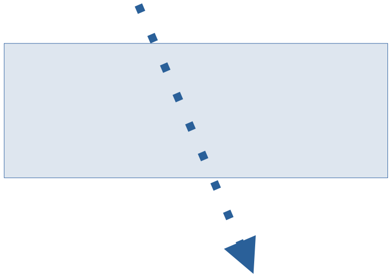
- Creation of N photons
- Weight = weight × 1/N

Russian roulette



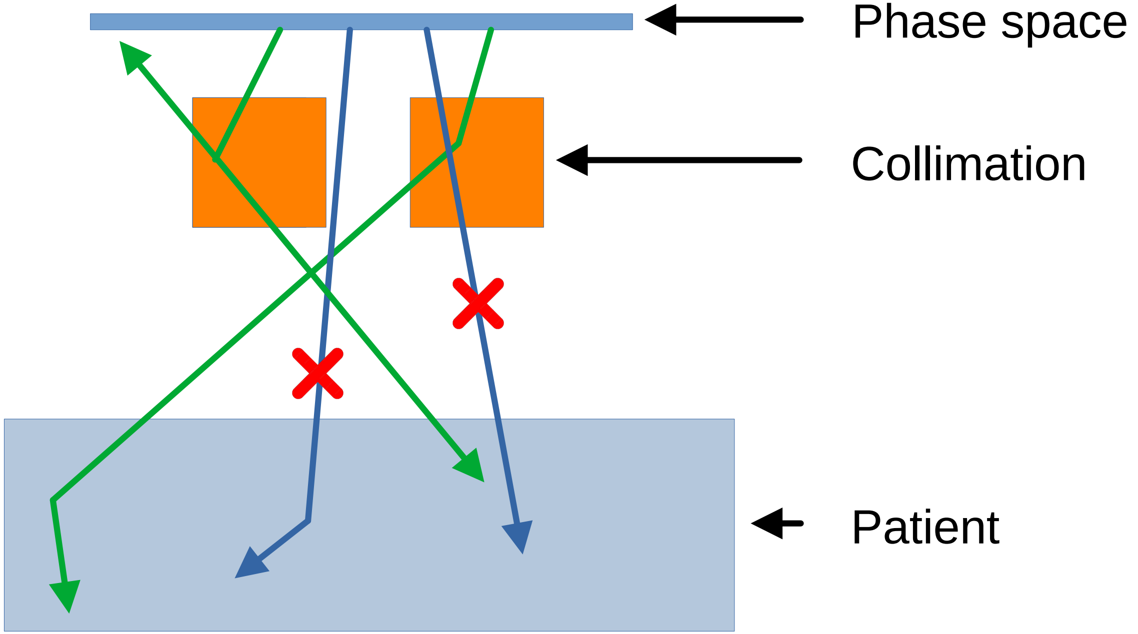
- Photon killing
  - 1/N probability
- Weight = weight × N

Free flight



- No interaction
- Weight update :
  - After each step
  - Beer-Lambert

Variance reduction for secondary photons :



$D_{sec.linac}$:

- Secondaries from scattered photons
- Linac head VRM
- Patient VRM

VRM applied to the collimation system

Particles entering in the collimation system

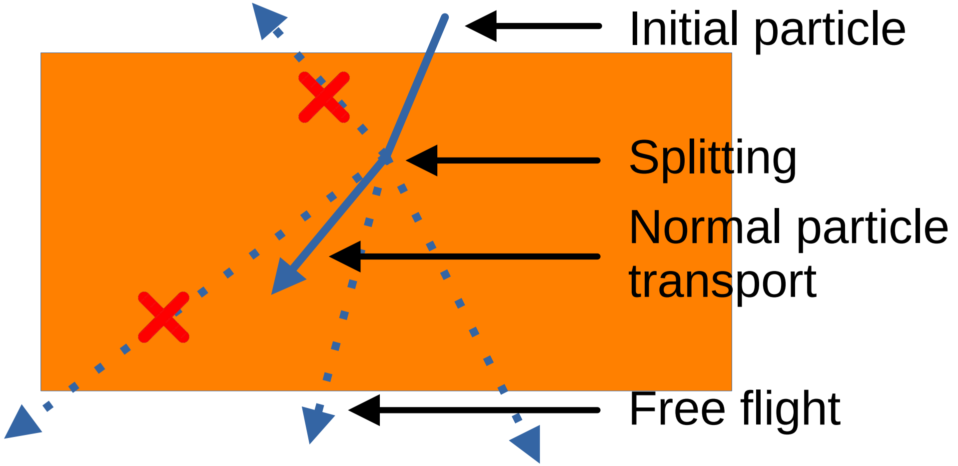
Biased processes

Compton	e-/+ pair
Rayleigh	eBrem

Splitting

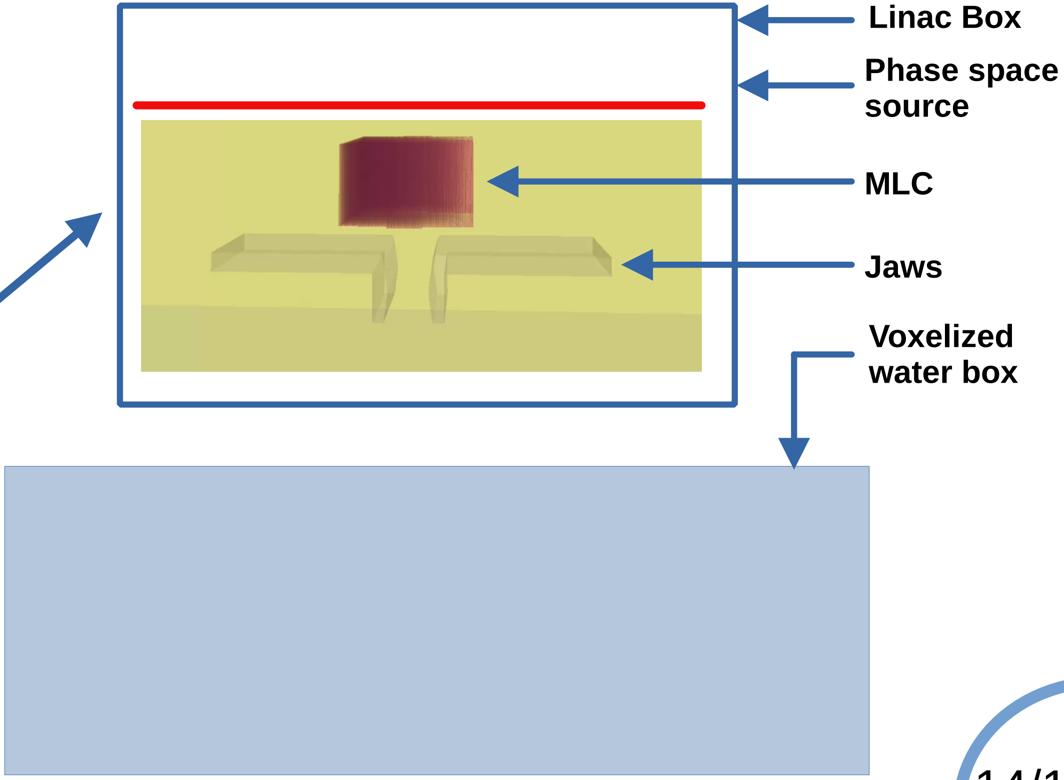
Initial particle kill

Free flight/Russian roulette

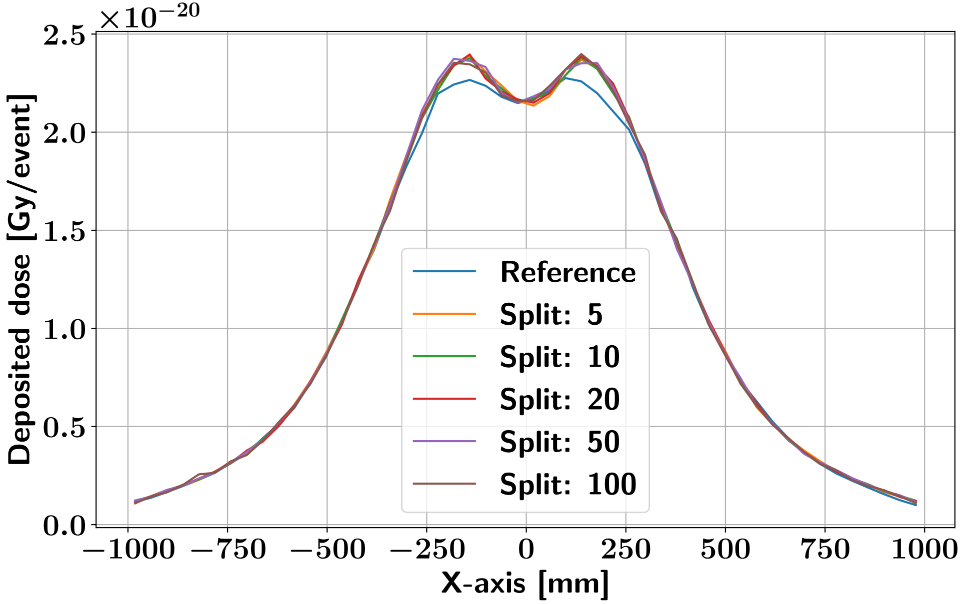


VRM implementation

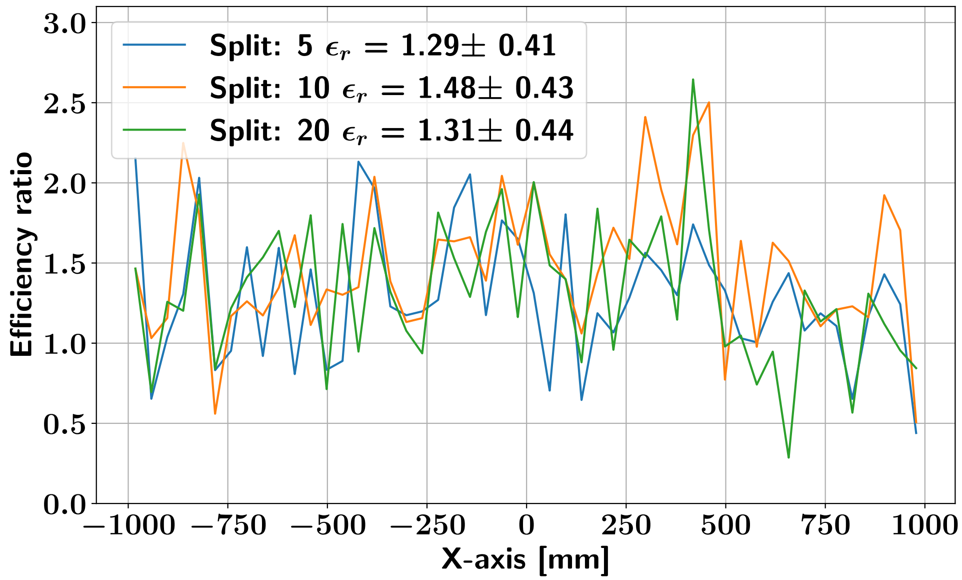
```python
pseudo_transportation_actor = sim.add_actor("ComptPseudoTransportationActor",
                                            "pseudo_transportation_actor")
pseudo_transportation_actor.mother = small_linac.name
pseudo_transportation_actor.splitting_factor = nb_split_linac
pseudo_transportation_actor.relative_min_weight_of_particle = 10**(-4)
pseudo_transportation_actor.russian_roulette_for_weights = True

pseudo_transportation_actor.russian_roulette_for_angle = True
pseudo_transportation_actor.max_theta = 90 * deg
pseudo_transportation_actor.vector_director = [0, 0, -1]
list_processes_to_bias_gamma = pseudo_transportation_actor.gamma_processes
list_processes_to_bias_electron = pseudo_transportation_actor.electron_processes
list_processes_to_bias_positron = pseudo_transportation_actor.positron_processes
```



Results: efficiency for simulation of secondary particles



Average deposited dose along the X-axis



Efficiency ratio between analog and biased methods along the X-axis

- 5 % of bias
- Gain in efficiency of ~ 50 %
  - Upper bound : efficiency ratio of 13

Conclusion

- A complete Linac model
  - Available in contrib/linacs/elektaversa.py
  - Static and dynamic simulations
- VRM is a work in progress :
  - A more efficient linac head VRM
  - VRM for the patient