Towards Gate 10 for ion beam therapy applications

M. Favaretto[1], Y. Jia[1,2], L. Grevillot[1], M. Stock[1,3], A. F. Resch[1]

[1]MedAustron Ion Therapy Centre, Wiener Neustadt, Austria

[2]Medical University of Vienna, Vienna, Austria

[3]Karl Landsteiner University of Health Sciences, Wiener Neustadt, Austria

MedAustron Ion Therapy Center

IDEAL v2

IDEAL v1

- Simulations: **GateRTion v1** (C++/mac)

- **3** different python **programs** to:
  - Preprocess the DICOM input
  - Write mac file to start the simulations in Gate
  - Postprocess dose output

- **Single threaded**
  - RAM limitation
  - need for external parallelization tools (HT Condor)

IDEAL v2

IDEAL v1

- Simulations: **GateRTion v1** (C++/mac)

- **3** different python **programs** to:
  - Preprocess the DICOM input
  - Write mac file to start the simulations in Gate
  - Postprocess dose output

- **Single threaded**
  - RAM limitation
  - need for external parallelization tools (HT Condor)

IDEAL v2

- Simulations: **Gate10** (python interface, C++ core)

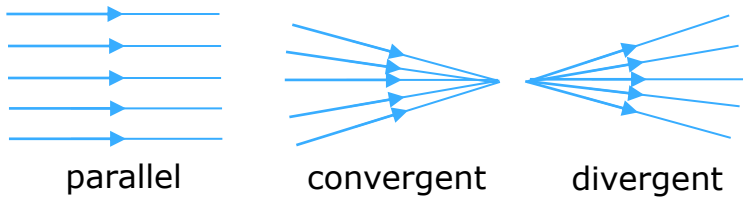
- **Single** python **program**:
  - Can directly read DICOM files
  - Simulations are started directly from python
  - Dose output available in python

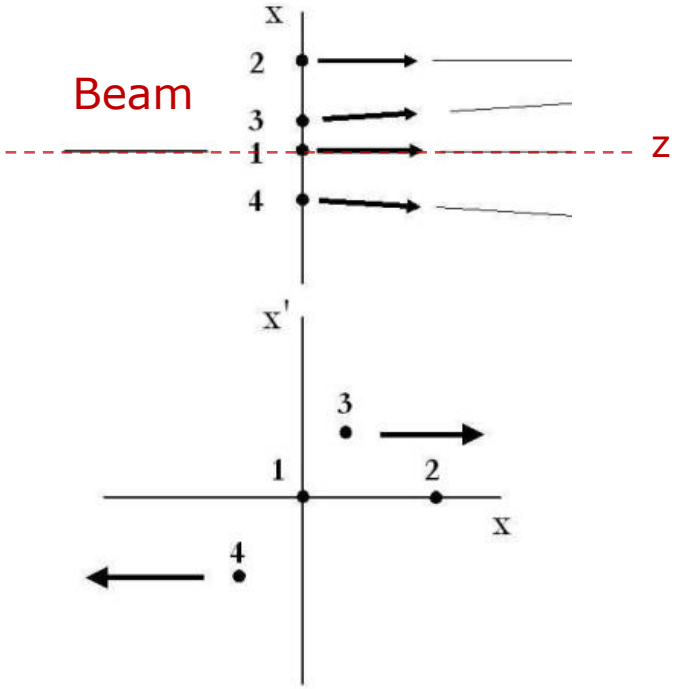
- **Multi-threading** possible
  - Potentially no need for external parallelization tools

Pencil beam source

Toward IDEAL v2: pencil beam source

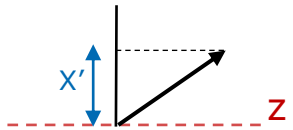
- Treatment plan is made of multiple pencil beams
- Beam has a convergence





- Pencil beam modelled according to Fermi-Eyges theory*
- Each particle is characterized by:
  - position x, y
  - direction x', y' (projected angles on x and y)
  - x and x' are correlated

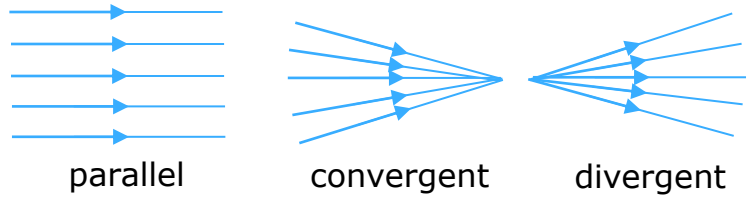
  -> the bigger x' the bigger will be the drift in x direction

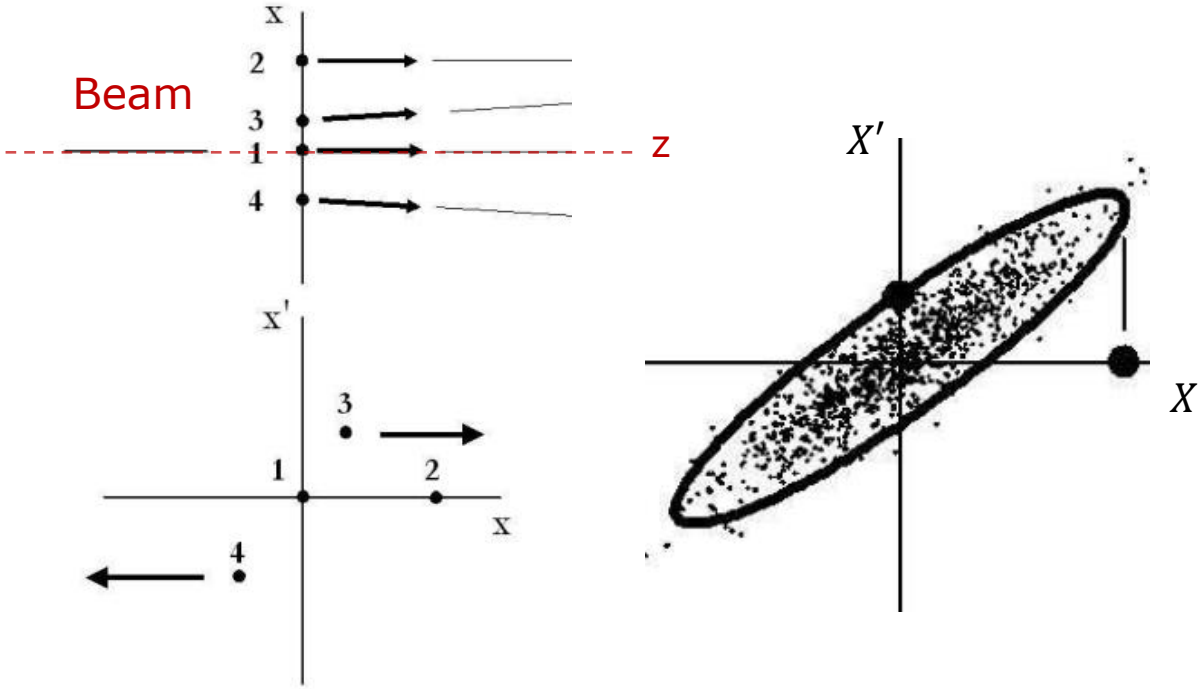


*Techniques of Proton Radiotherapy: Transport Theory B. Gottschalk May 1, 2012

Toward IDEAL v2: pencil beam source

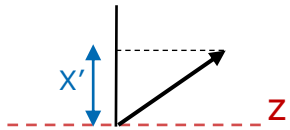
- Treatment plan is made of multiple pencil beams
- Beam has a convergence





- Pencil beam modelled according to Fermi-Eyges theory*
- Each particle is characterized by:
  - position x, y
  - direction x', y' (projected angles on x and y)
  - x and x' are correlated

  -> the bigger x' the bigger will be the drift in x direction



*Techniques of Proton Radiotherapy: Transport Theory B. Gottschalk May 1, 2012

Toward IDEAL v2: pencil beam source

- 8 parameters to describe the correlated momentum spread (each for the x and y)
  - spot size σ
  - divergence θ
  - emittance ε
  - convergence flag $[1,0]$

-> sample particle's position and direction

- 3 parameters to describe the energy type, mean energy and energy spread

-> spread of Bragg peak

Toward IDEAL v2: pencil beam source

- 8 parameters to describe the correlated momentum spread (each for the x and y)
  - spot size σ
  - divergence θ
  - emittance ε
  - convergence flag $[1,0]$

-> sample particle's position and direction

- 3 parameters to describe the energy type, mean energy and energy spread

-> spread of Bragg peak

ADD TO SIMULTION

```
source = sim.add_source("PencilBeamSource", "mysource")
source.energy.type = 'gauss'
source.energy.mono = 1440 * MeV
source.energy.sigma_gauss = 0.1
source.particle = "ion 6 12"  # carbon
source.position.type = "disc"
source.position.rotation = Rotation.from_euler("y", -90, degrees=True).as_matrix()
source.position.translation = [-100 * mm, 20 * mm, 30 * mm]
source.n = 20000
source.direction.partPhSp_x = [
    2.3335754 * mm,
    2.3335754 * mrad,
    0.00078728 * mm * mrad,
    0,
]
source.direction.partPhSp_y = [
    1.96433431 * mm,
    0.00079118 * mrad,
    0.00249161 * mm * mrad,
    0,
]
```

Toward IDEAL v2: pencil beam source

- 8 parameters to describe the correlated momentum spread (each for the x and y)
  - spot size σ
  - divergence θ
  - emittance ε
  - convergence flag $[1,0]$

-> sample particle's position and direction

- 3 parameters to describe the energy type, mean energy and energy spread

-> spread of Bragg peak

ADD TO SIMULTION

```
source = sim.add_source("PencilBeamSource", "mysource")
source.energy.type = 'gauss'
source.energy.mono = 1440 * MeV
source.energy.sigma_gauss = 0.1
source.particle = "ion 6 12"  # carbon
source.position.type = "disc"
source.position.rotation = Rotation.from_euler("y", -90, degrees=True).as_matrix()
source.position.translation = [-100 * mm, 20 * mm, 30 * mm]
source.n = 20000
source.direction.partPhSp_x = [
    2.3335754 * mm,
    2.3335754 * mrad,
    0.00078728 * mm * mrad,
    0,
]
source.direction.partPhSp_y = [
    1.96433431 * mm,
    0.00079118 * mrad,
    0.00249161 * mm * mrad,
    0,
]
```

Toward IDEAL v2: pencil beam source

- 8 parameters to describe the correlated momentum spread (each for the x and y)
  - spot size σ
  - divergence θ
  - emittance ε
  - convergence flag $[1,0]$

-> sample particle's position and direction

- 3 parameters to describe the energy type, mean energy and energy spread

-> spread of Bragg peak

ADD TO SIMULTION

```
source = sim.add_source("PencilBeamSource", "mysource")
source.energy.type = 'gauss'
source.energy.mono = 1440 * MeV
source.energy.sigma_gauss = 0.1
source.particle = "ion 6 12"  # carbon
source.position.type = "disc"
source.position.rotation = Rotation.from_euler("y", -90, degrees=True).as_matrix()
source.position.translation = [-100 * mm, 20 * mm, 30 * mm]
source.n = 20000
source.direction.partPhSp_x = [
    2.3335754 * mm,
    2.3335754 * mrad,
    0.00078728 * mm * mrad,
    0,
]
source.direction.partPhSp_y = [
    1.96433431 * mm,
    0.00079118 * mrad,
    0.00249161 * mm * mrad,
    0,
]
```

Toward IDEAL v2: pencil beam source

- 8 parameters to describe the correlated momentum spread (each for the x and y)
  - spot size σ
  - divergence θ
  - emittance ε
  - convergence flag $[1,0]$

-> sample particle's position and direction

- 3 parameters to describe the energy type, mean energy and energy spread

-> spread of Bragg peak

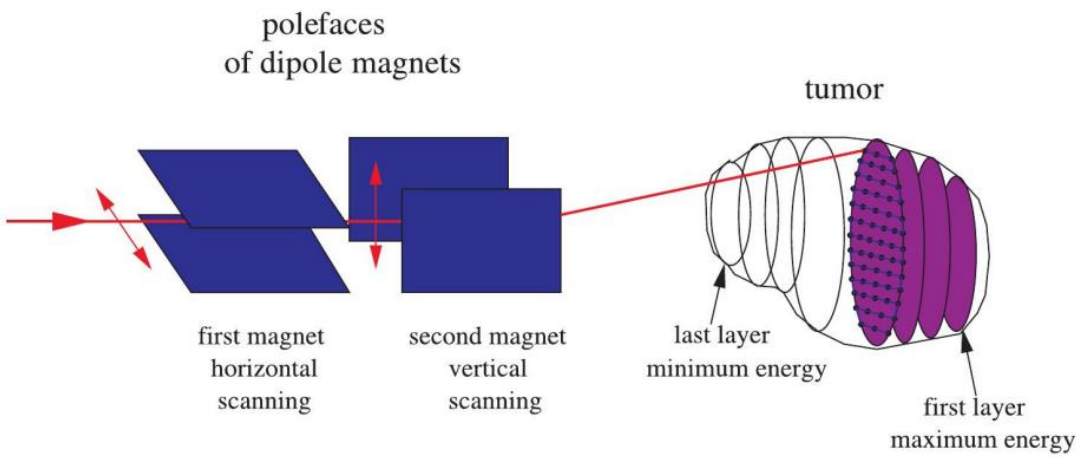
ADD TO SIMULTION

```
source = sim.add_source("PencilBeamSource", "mysource")
source.energy.type = 'gauss'
source.energy.mono = 1440 * MeV
source.energy.sigma_gauss = 0.1
source.particle = "ion 6 12"  # carbon
source.position.type = "disc"
source.position.rotation = Rotation.from_euler("y", -90, degrees=True).as_matrix()
source.position.translation = [-100 * mm, 20 * mm, 30 * mm]
source.n = 20000
source.direction.partPhSp_x = [
    2.3335754 * mm,
    2.3335754 * mrad,
    0.00078728 * mm * mrad,
    0,
]
source.direction.partPhSp_y = [
    1.96433431 * mm,
    0.00079118 * mrad,
    0.00249161 * mm * mrad,
    0,
]
```

➡ Available in latest release !
(test044_pbs*)

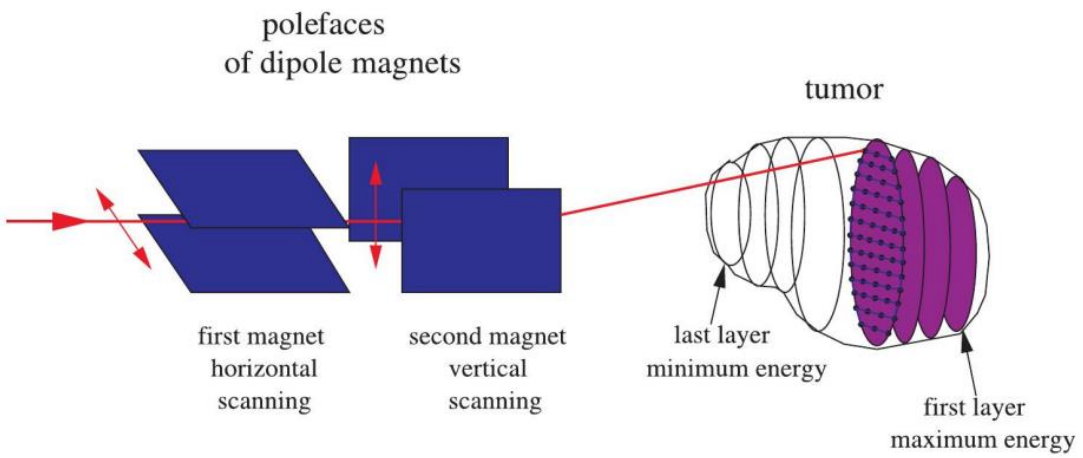
Treatment plan source

Toward IDEAL v2: treatment plan source



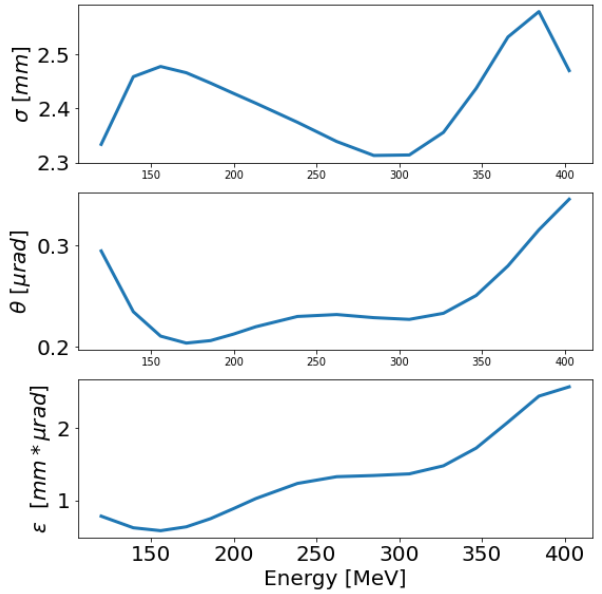
- Treatment plan is made of multiple pencil beams, each irradiating a single spot
- Each spot is characterized by:
  - position (x, y)
  - energy -> depth in the target
  - weight -> number of particles to deliver

Toward IDEAL v2: treatment plan source

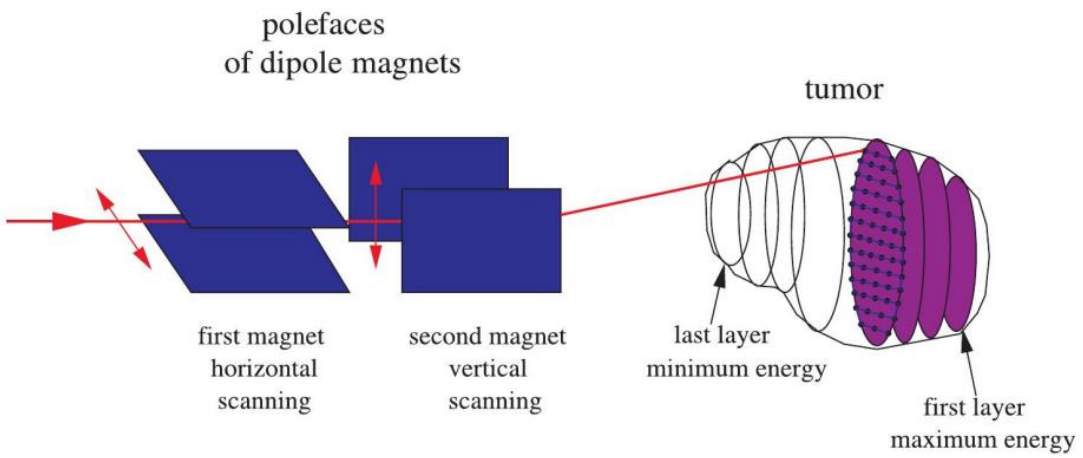


- Treatment plan is made of multiple pencil beams, each irradiating a single spot
- Each spot is characterized by:
  - position (x, y)
  - energy -> depth in the target
  - weight -> number of particles to deliver

- Need to describe beam at nozzle exit:
  - description of energy dependent parameters
  - geometry: position of steering magnets

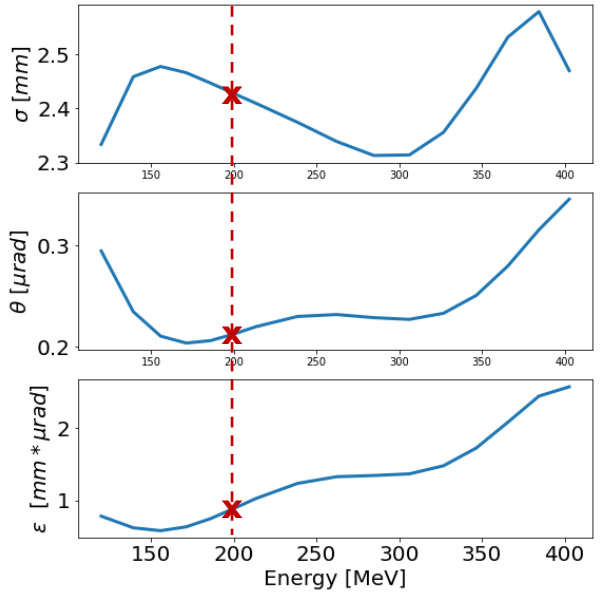


Toward IDEAL v2: treatment plan source



- Treatment plan is made of multiple pencil beams, each irradiating a single spot
- Each spot is characterized by:
  - position (x, y)
  - energy -> depth in the target
  - weight -> number of particles to deliver

- Need to describe beam at nozzle exit:
  - description of energy dependent parameters
  - geometry: position of steering magnets



Toward IDEAL v2: treatment plan source

- Gate10 Treatment Plan source: array of Pencil Beam sources, one for each spot
  - Only on python side, no Cpp implementation needed

- Initialization:
  - Spots to scan
    - from DICOM RT plan file path
    - from .txt → backward compatibility Gate 9
    - each spot manually → testing and debugging
  - Beamline model
    - set Pencil Beam energy-dependent parameters
  - Total number of particles to simulate

ADD TO SIMULTION

From RT dicom path

```
tps = gate.TreatmentPlanSource("RT_plan", sim)
tps.set_beamline_model(IR2HBL)
tps.set_particles_to_simulate(nSim)
tps.set_spots_from_rtplan(rt_plan)
tps.initialize_tpsource()
```

From .txt

```
spots, ntot, energies, G = gate.spots_info_from_txt(
    ref_path / "TreatmentPlan4Gate.txt", "ion 6 12"
)
tps.set_spots(spots)
```

Toward IDEAL v2: treatment plan source

- Gate10 Treatment Plan source: array of Pencil Beam sources, one for each spot
  - Only on python side, no Cpp implementation needed

- Initialization:
  - Spots to scan
    - from DICOM RT plan file path
    - from .txt → backward compatibility Gate 9
    - each spot manually → testing and debugging
  - Beamline model
    - set Pencil Beam energy-dependent parameters
  - Total number of particles to simulate

ADD TO SIMULTION

From RT dicom path

```
tps = gate.TreatmentPlanSource("RT_plan", sim)
tps.set_beamline_model(IR2HBL)
tps.set_particles_to_simulate(nSim)
tps.set_spots_from_rtplan(rt_plan)
tps.initialize_tpsource()
```

From .txt

```
spots, ntot, energies, G = gate.spots_info_from_txt(
    ref_path / "TreatmentPlan4Gate.txt", "ion 6 12"
)
tps.set_spots(spots)
```

→ Not yet available in latest release !
(PR #123)

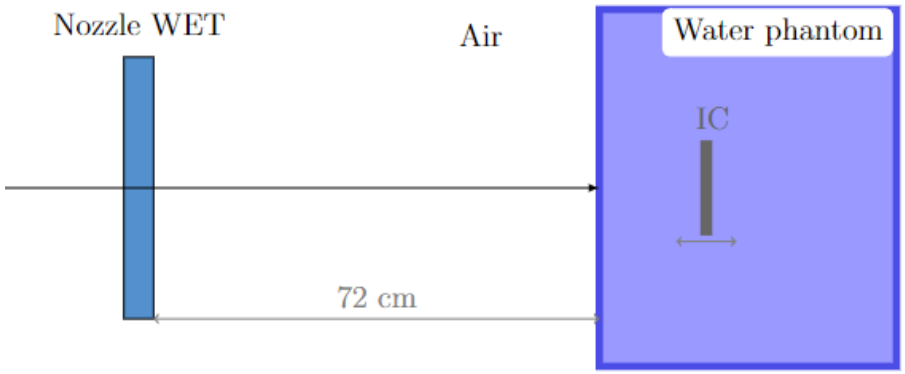
Benchmark against GateRTion v1

Range in water, absolute dose, optics, gantry rotation

Benchmark test cases

Test Gate 10 vs GateRTion

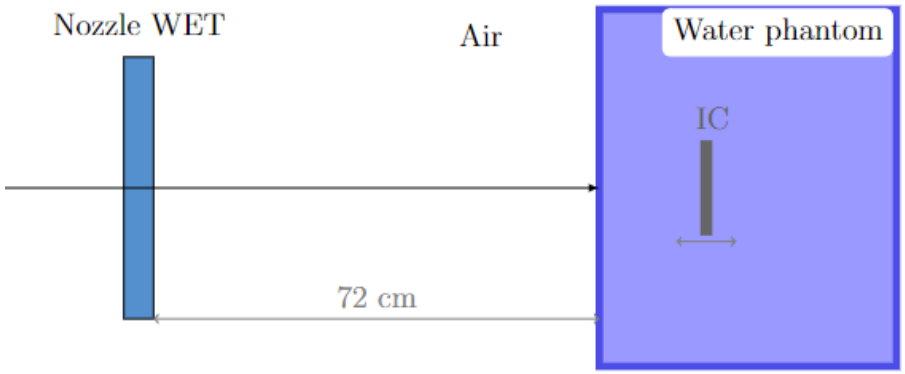
- Range in water
  - test051_TPsource_range_ref.py



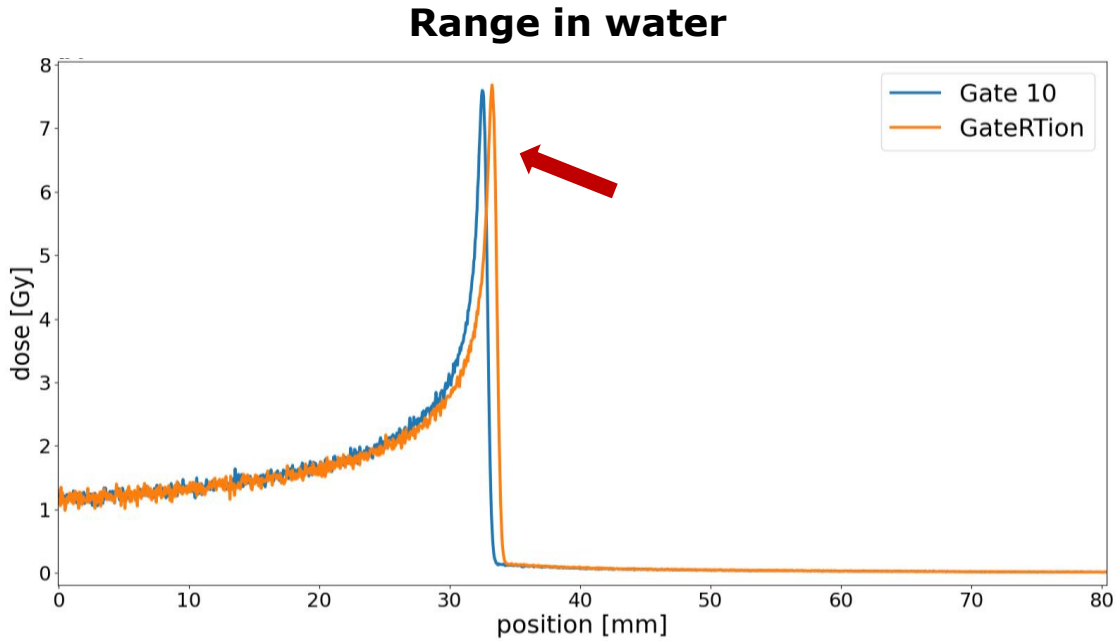
Benchmark test cases

Test Gate 10 vs GateRTion

- Range in water
  - test051_TPsource_range_ref.py



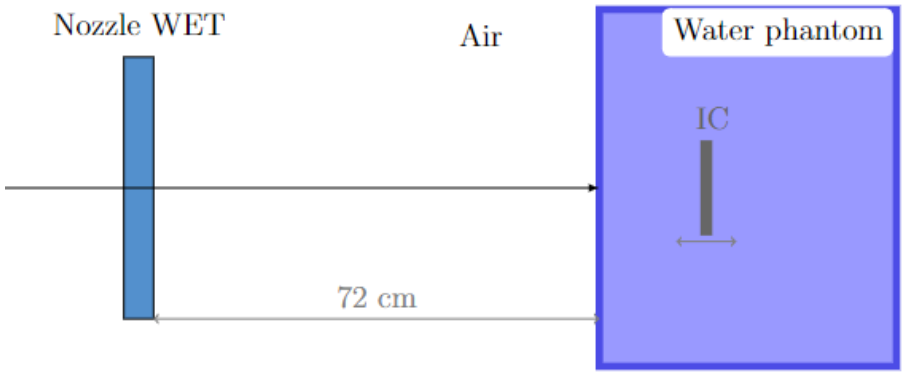
beam



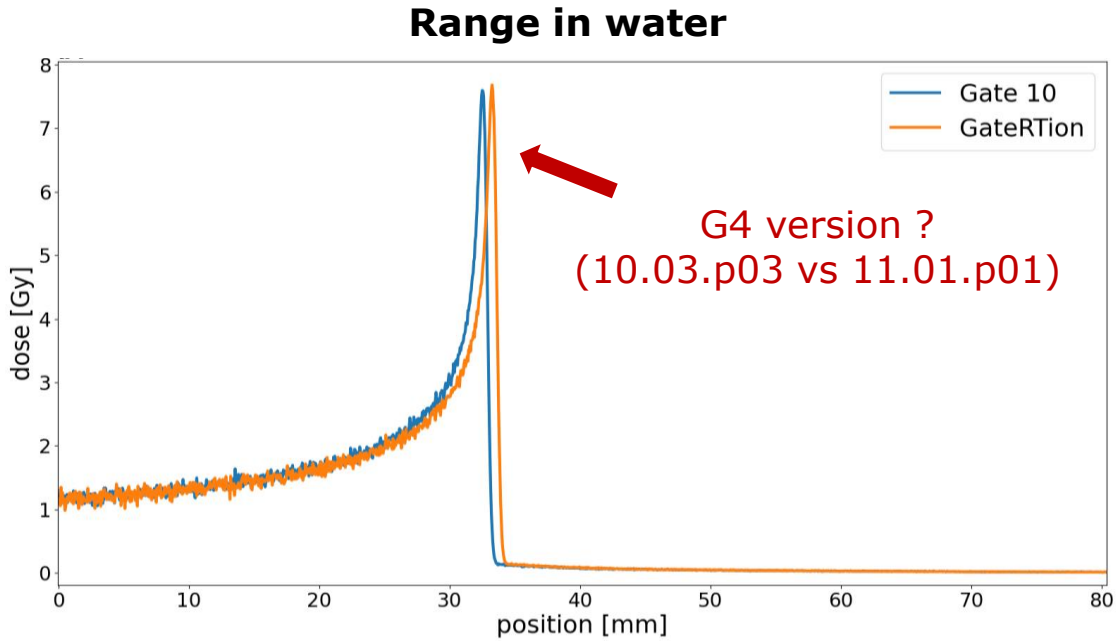
Benchmark test cases

Test Gate 10 vs GateRTion

- Range in water
  - test051_TPsource_range_ref.py



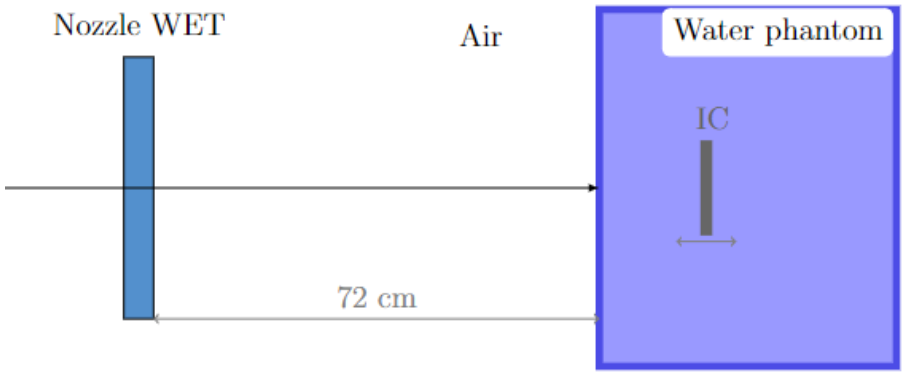
beam



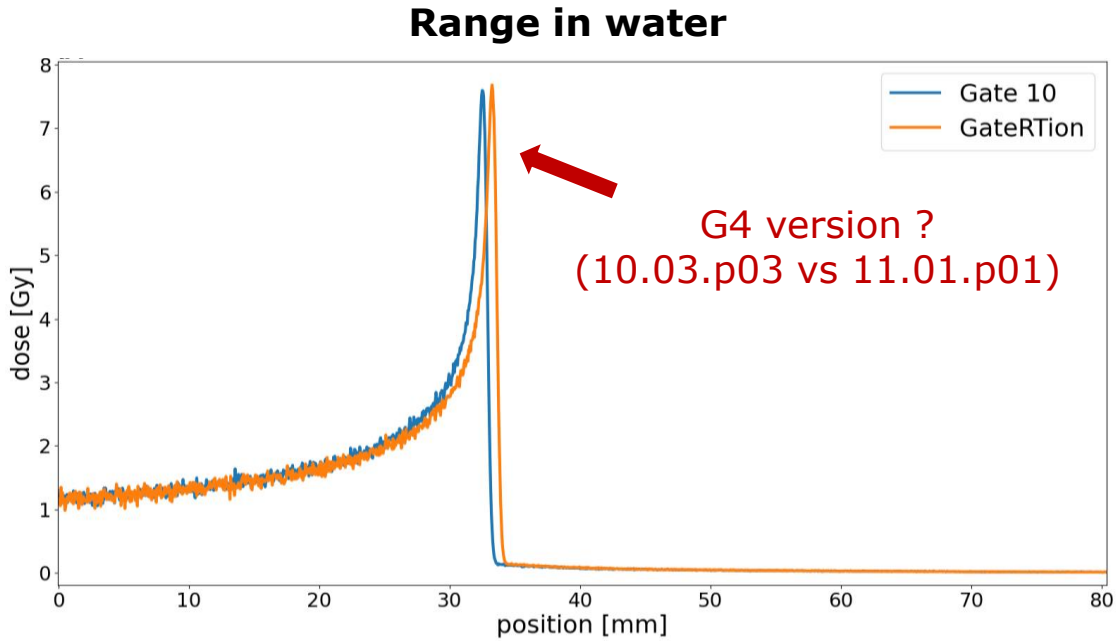
Benchmark test cases

Test Gate 10 vs GateRTion

- Range in water
  - test051_TPsource_range_ref.py ✓



beam

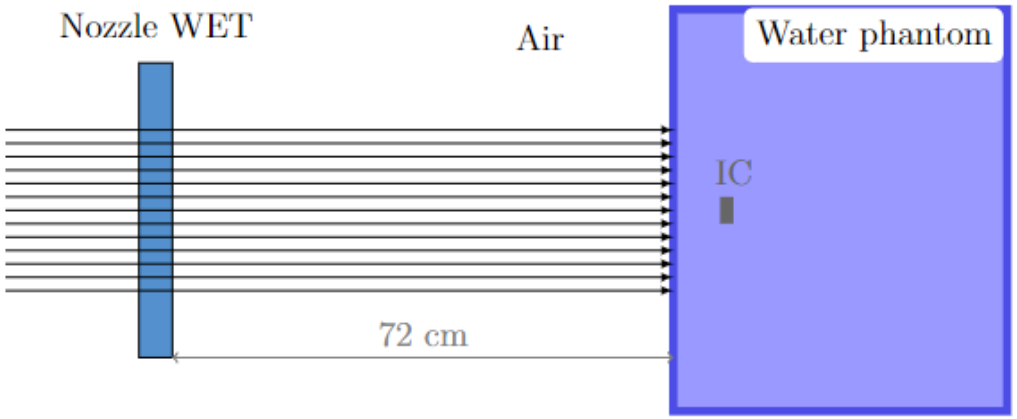


Range difference < 1 mm

Benchmark test cases

Test Gate 10 vs GateRTion

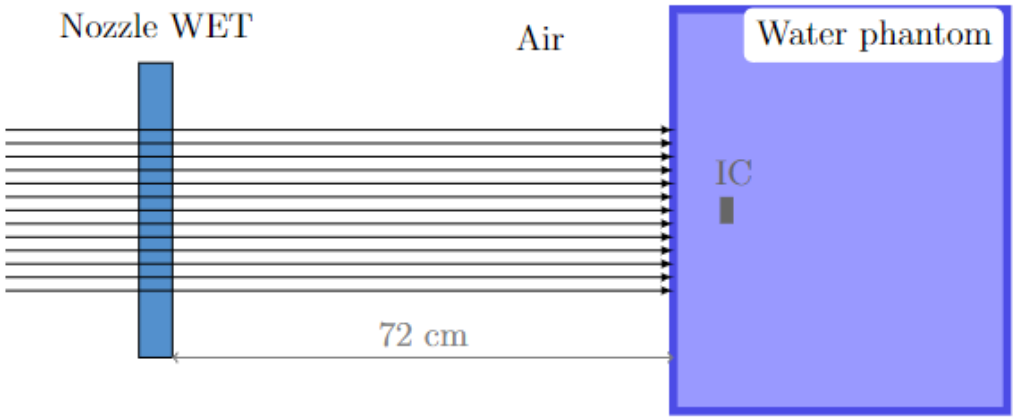
- Range in water
  - test051_TPsource_range_ref.py ✓
- Abs dose
  - test051_Tpsource_abs_dose.py



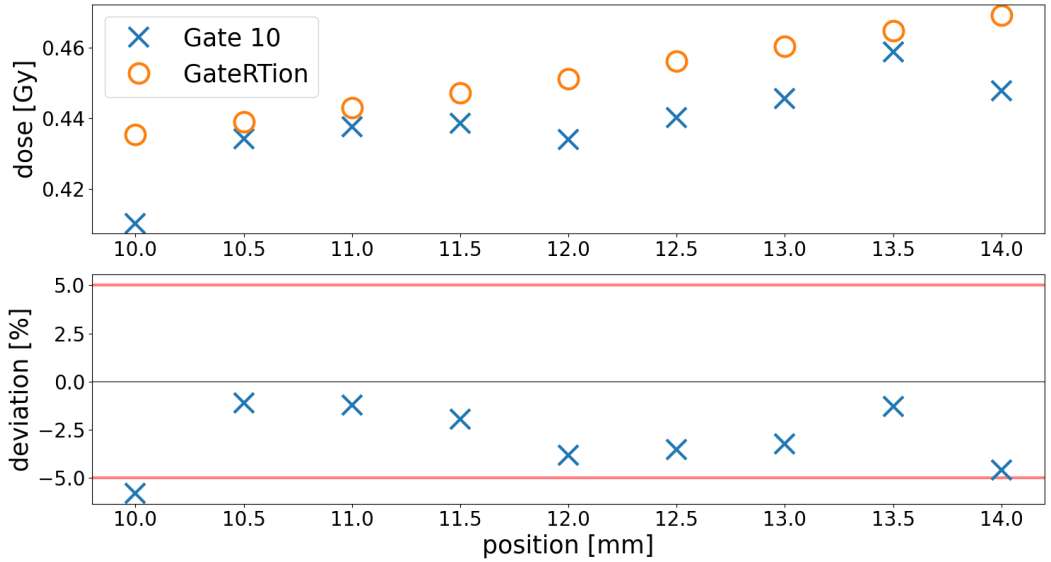
Benchmark test cases

Test Gate 10 vs GateRTion

- Range in water
  - test051_TPsource_range_ref.py ✓
- Abs dose
  - test051_Tpsource_abs_dose.py



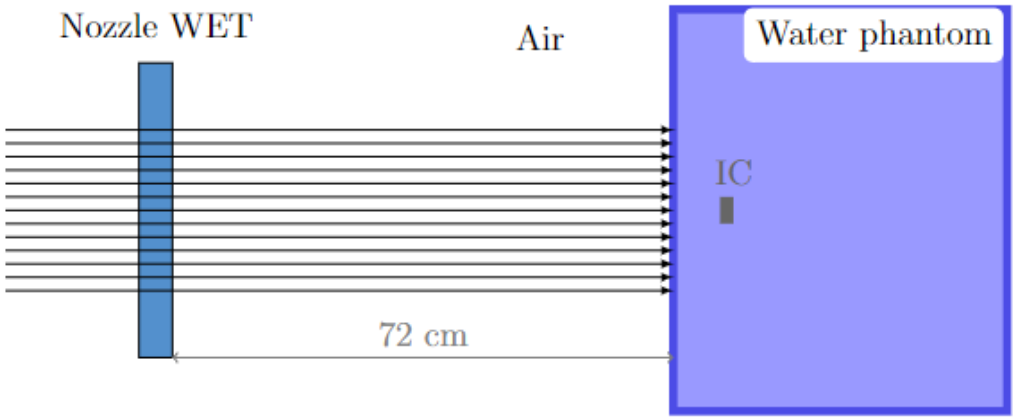
Absolute dose



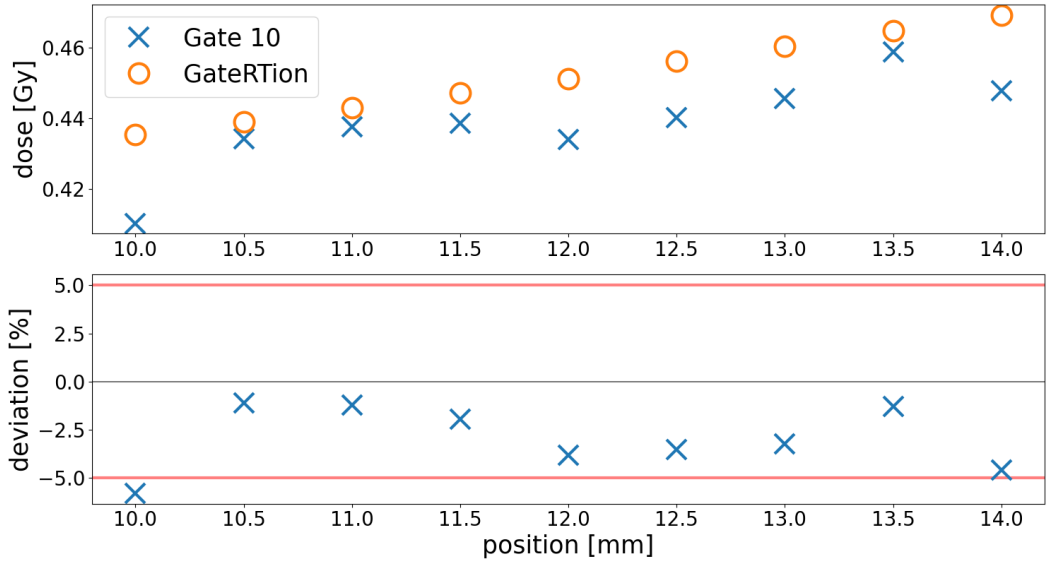
Benchmark test cases

Test Gate 10 vs GateRTion

- Range in water
  - test051_TPsource_range_ref.py ✓
- Abs dose
  - test051_Tpsource_abs_dose.py ✓



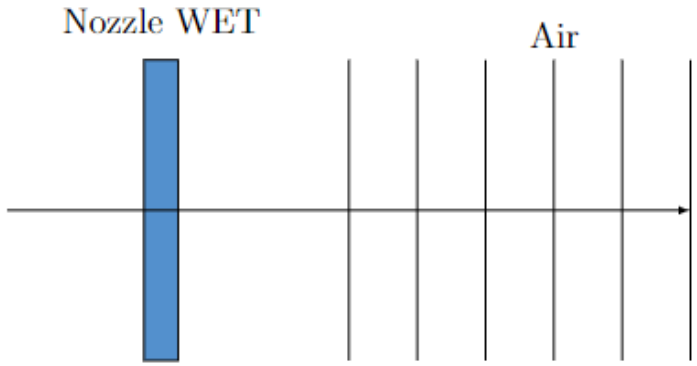
Absolute dose



Benchmark test cases

Test Gate 10 vs GateRTion

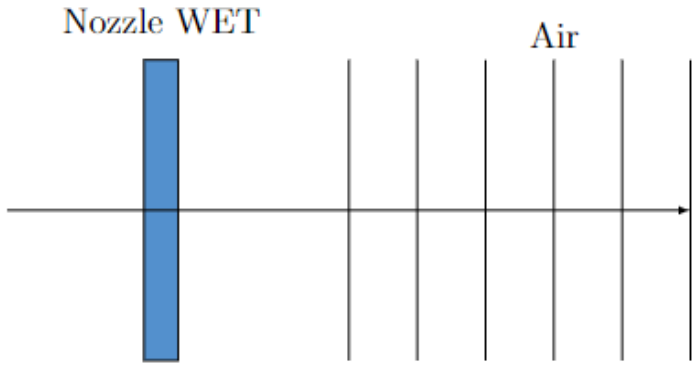
- Range in water
  - test051_TPsource_range_ref.py ✓
- Abs dose
  - test051_Tpsource_abs_dose.py ✓
- Optics: spots sizes and positions
  - test051_TPsource_optics.py
  - test051_TPsource_optics_vbl.py



Benchmark test cases

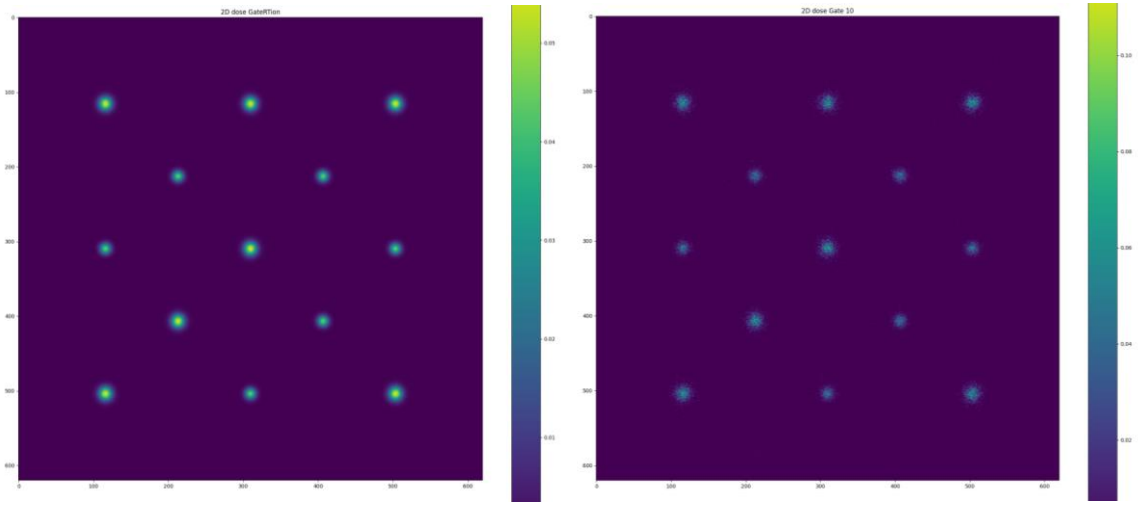
Test Gate 10 vs GateRTion

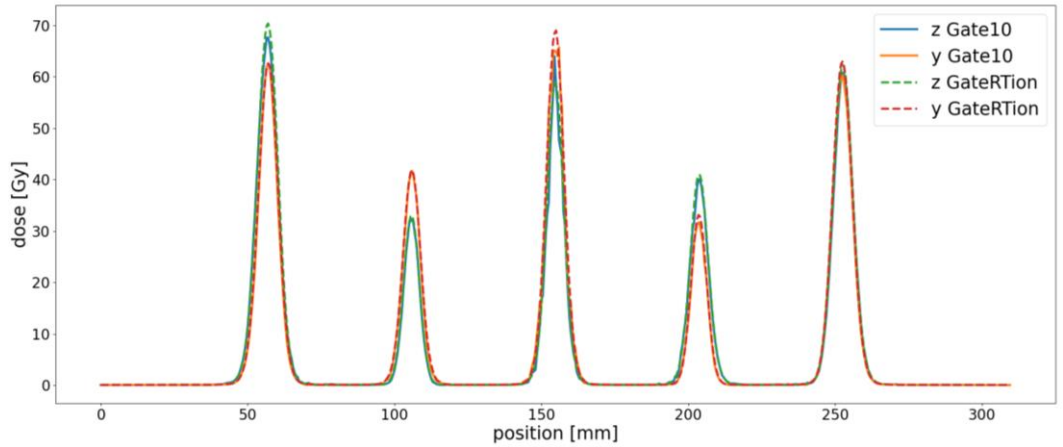
- Range in water
  - test051_TPsource_range_ref.py ✓
- Abs dose
  - test051_Tpsource_abs_dose.py ✓
- Optics: spots sizes and positions
  - test051_TPsource_optics.py
  - test051_TPsource_optics_vbl.py



Optics

GateRTion **Gate 10 (lower statistics)**

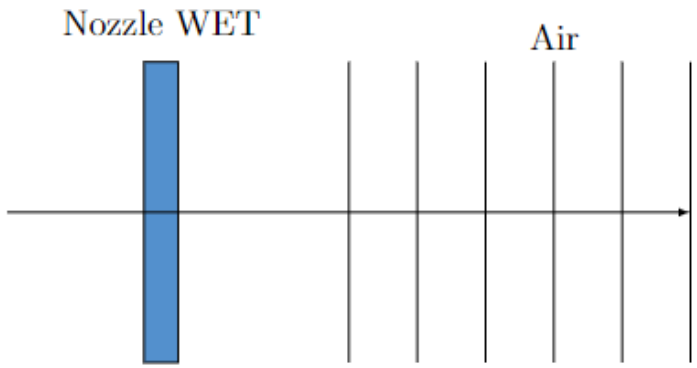




Benchmark test cases

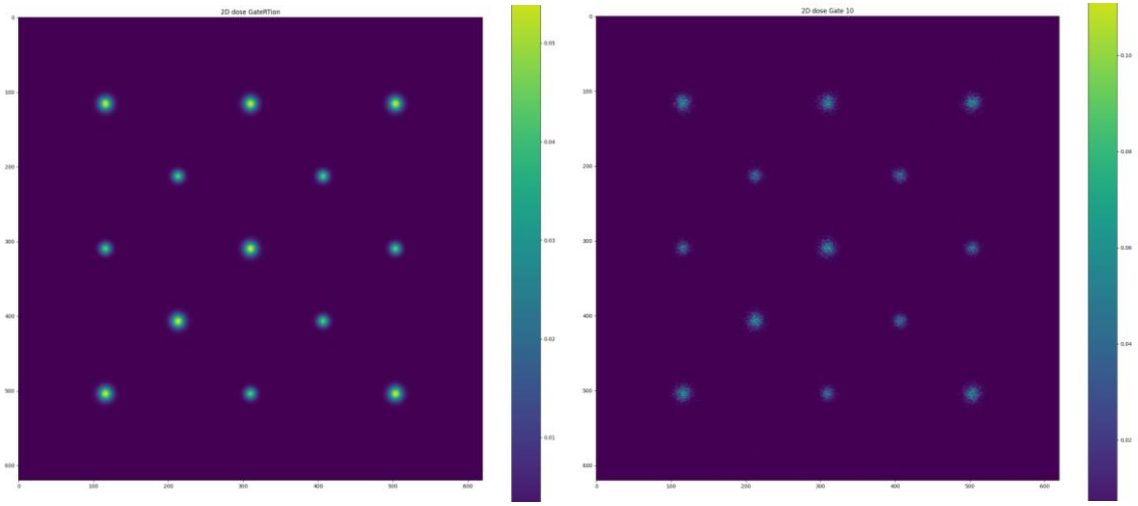
Test Gate 10 vs GateRTion

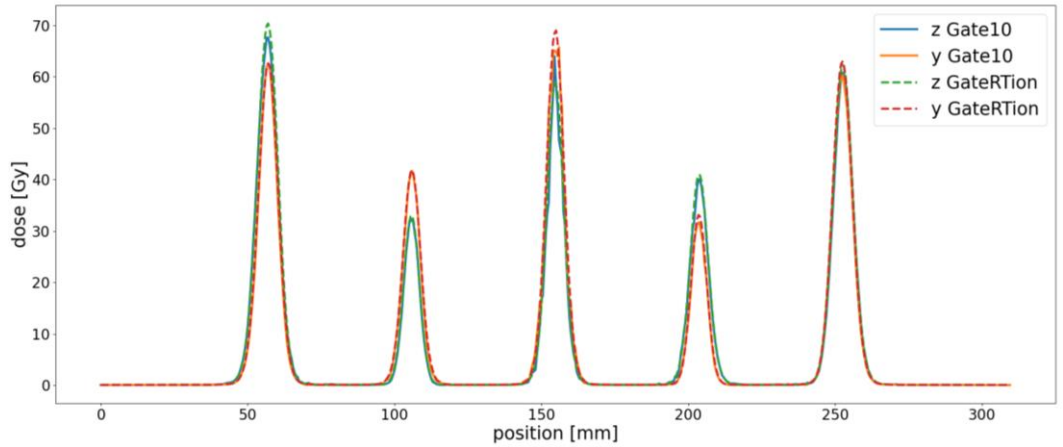
- Range in water
  - test051_TPsource_range_ref.py ✓
- Abs dose
  - test051_Tpsource_abs_dose.py ✓
- Optics: spots sizes and positions
  - test051_TPsource_optics.py
  - test051_TPsource_optics_vbl.py ✓



Optics

GateRTion **Gate 10 (lower statistics)**





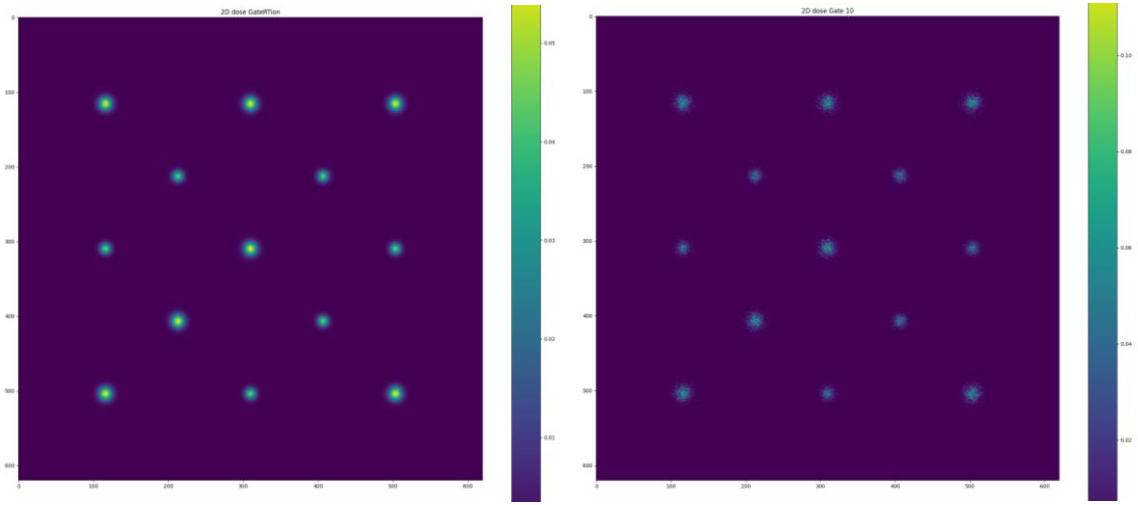
Benchmark test cases

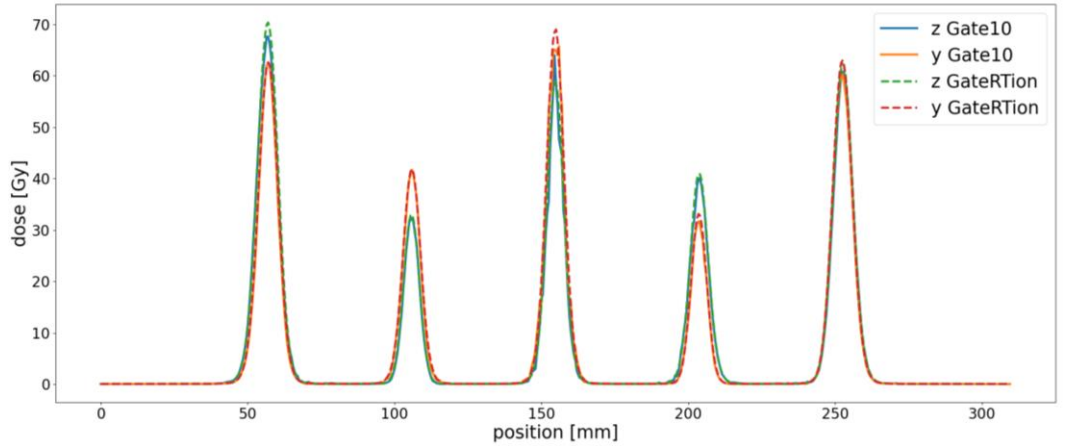
Test Gate 10 vs GateRTion

- Range in water
  - test051_TPsource_range_ref.py ✓
- Abs dose
  - test051_Tpsource_abs_dose.py ✓
- Optics: spots sizes and positions
  - test051_TPsource_optics.py
  - test051_TPsource_optics_vbl.py ✓
- Gantry rotation: range in water HBL vs VBL
  - test051_TPsource_gantry_rot.py

Optics

GateRTion **Gate 10 (lower statistics)**





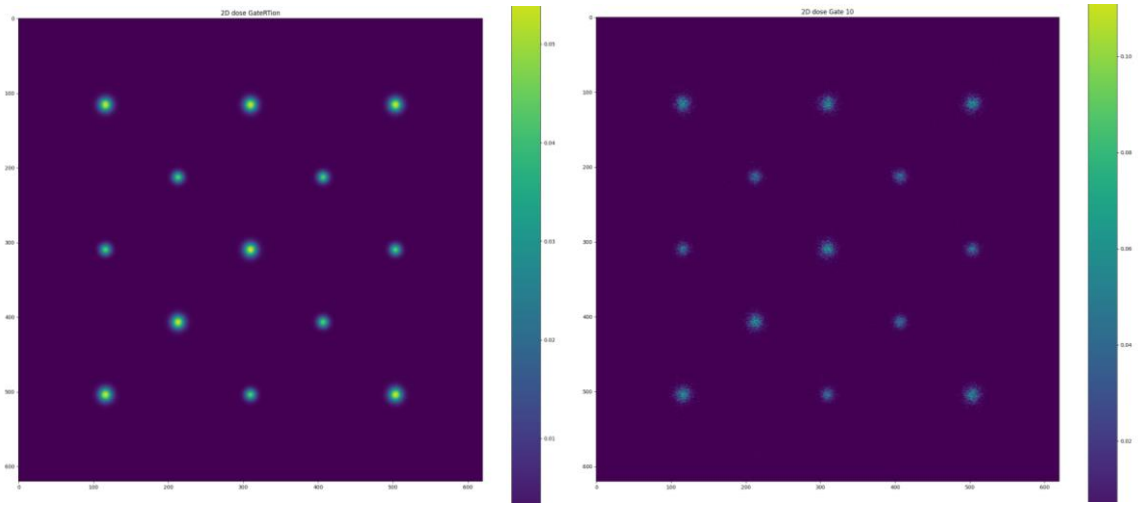
Benchmark test cases

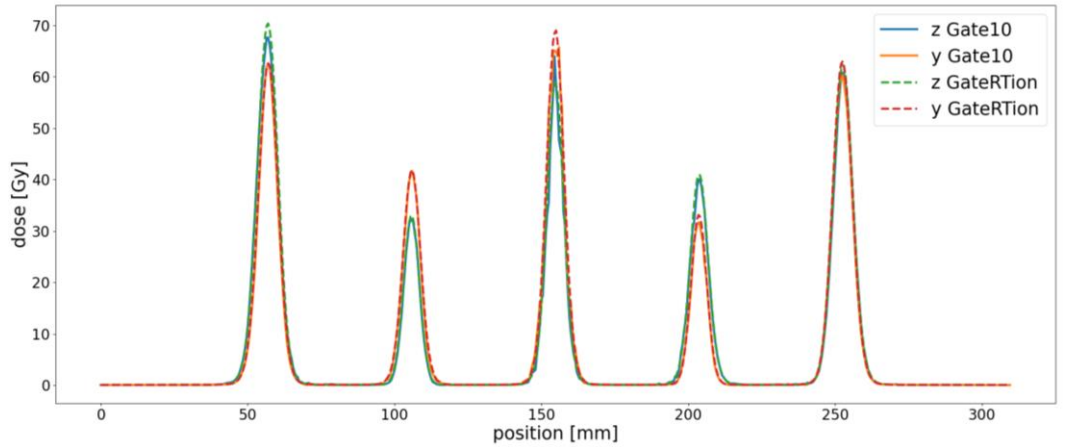
Test Gate 10 vs GateRTion

- Range in water
  - test051_TPsource_range_ref.py ✓
- Abs dose
  - test051_Tpsource_abs_dose.py ✓
- Optics: spots sizes and positions
  - test051_TPsource_optics.py
  - test051_TPsource_optics_vbl.py ✓
- Gantry rotation: range in water HBL vs VBL
  - test051_TPsource_gantry_rot.py ✓

Optics

GateRTion **Gate 10 (lower statistics)**





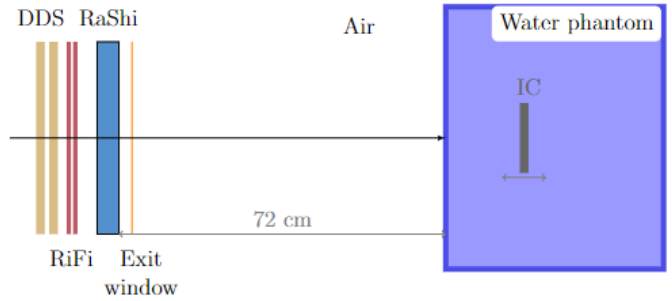
Benchmark against measurements

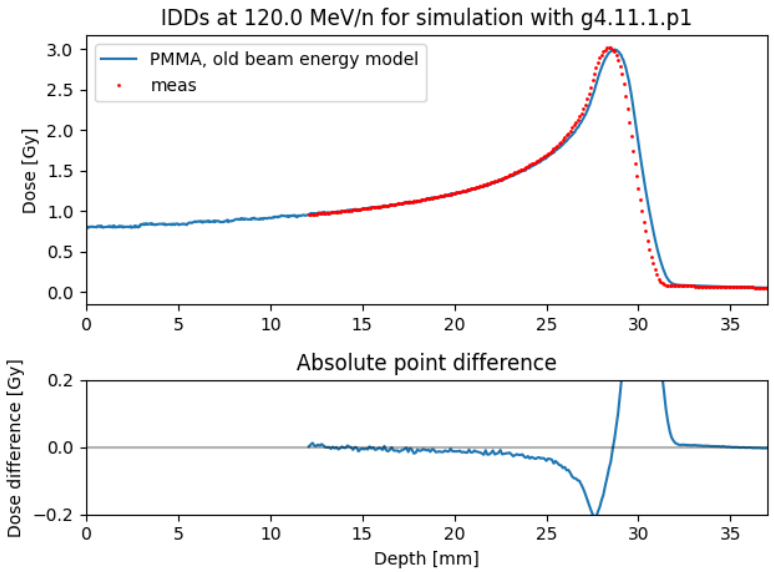
Range in water (MedAustron's data)

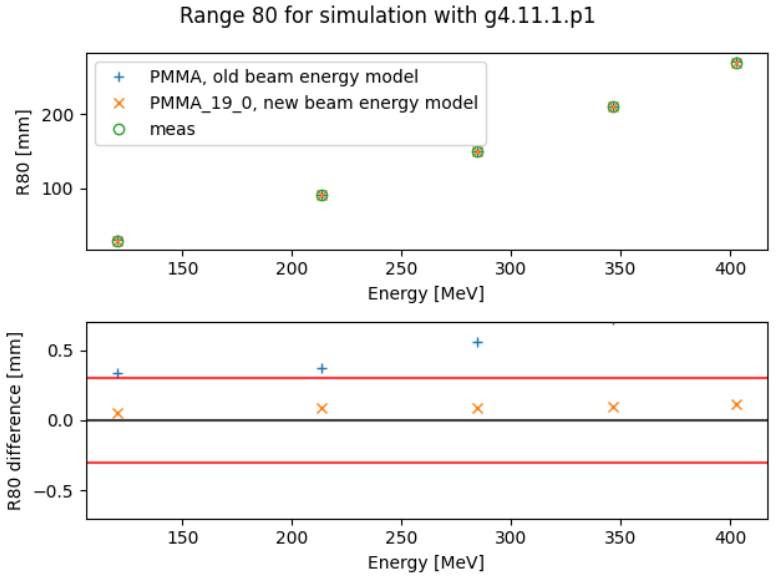
Benchmark against measurements

Tests against MedAustron measurement data

- MedAustron nozzle simulated in Gate 10
- Results before and after re-tuning beam energy model
- MA measurement vs Gate 10 (G4 11.1.p1)



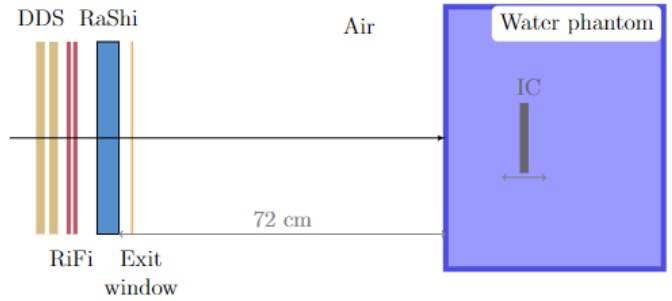


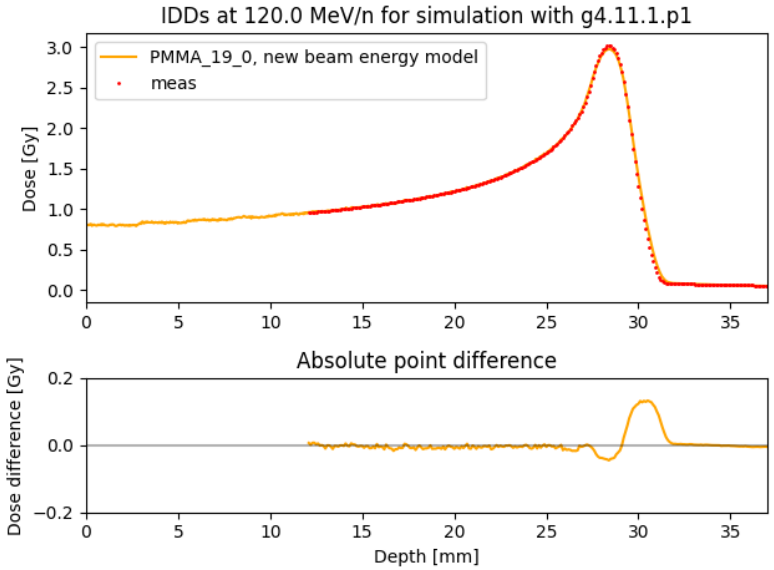


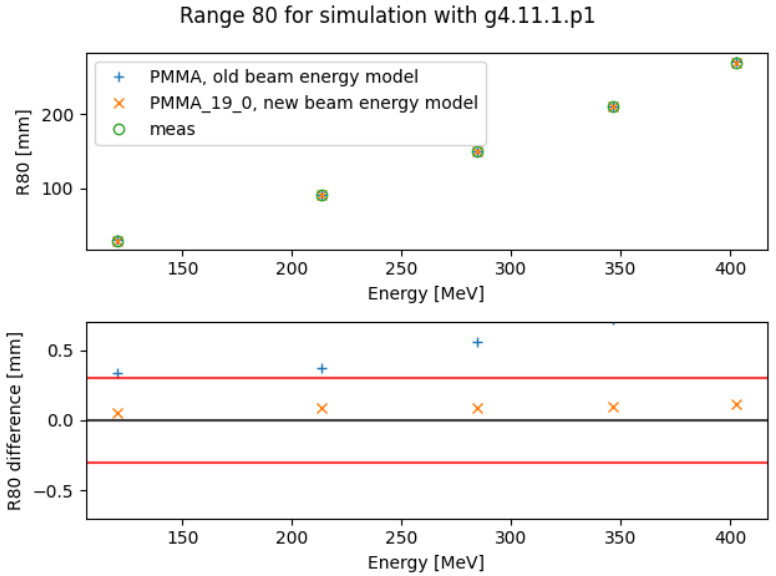
Benchmark against measurements

Tests against MedAustron measurement data

- MedAustron nozzle simulated in Gate 10
- Results before and after re-tuning beam energy model
- MA measurement vs Gate 10 (G4 11.1.p1)

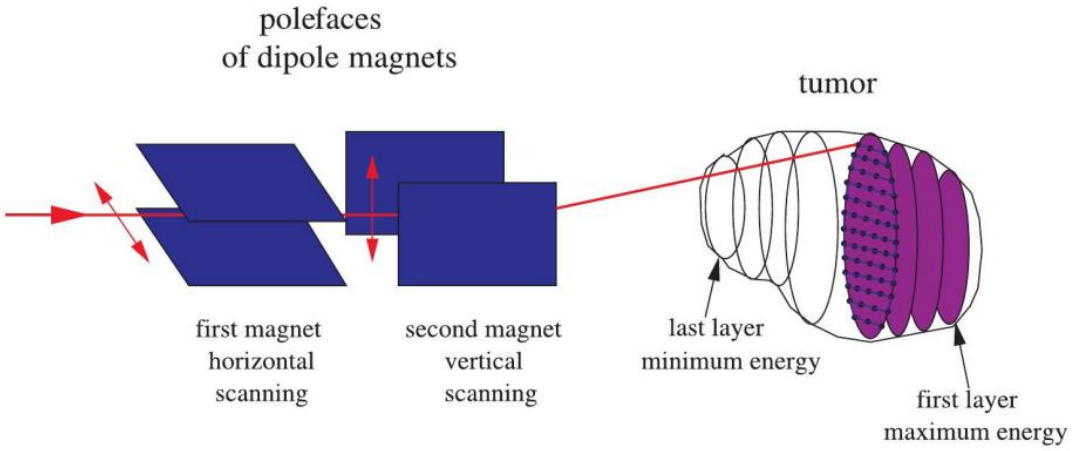






Conclusions

- IDEAL v2: simplify software architecture using Gate 10
- Features developed in Gate 10 for IDEAL v2:
  - **Pencil beam source** -> included in latest release
  - **Treatment plan source** -> pull request #123
- Treatment plan source validation
- First validation against MA measurements promising → **MORE COMING**



Range in water ✓
test051_TPsource_range_ref.py

Abs dose ✓
test051_Tpsource_abs_dose.py

Optics: spots sizes and positions ✓
test051_TPsource_optics.py
test051_TPsource_optics_vbl.py

Gantry rotation: range in water HBL vs VBL ✓
test051_TPsource_gantry_rot.py

Thank you for your attention!

Thanks to the collaboration of: David Sarrut, Thomas Baudier and all the others

Contacts:
martina.favaretto@medaustron.at

Kofinanziert von der Europäischen Union

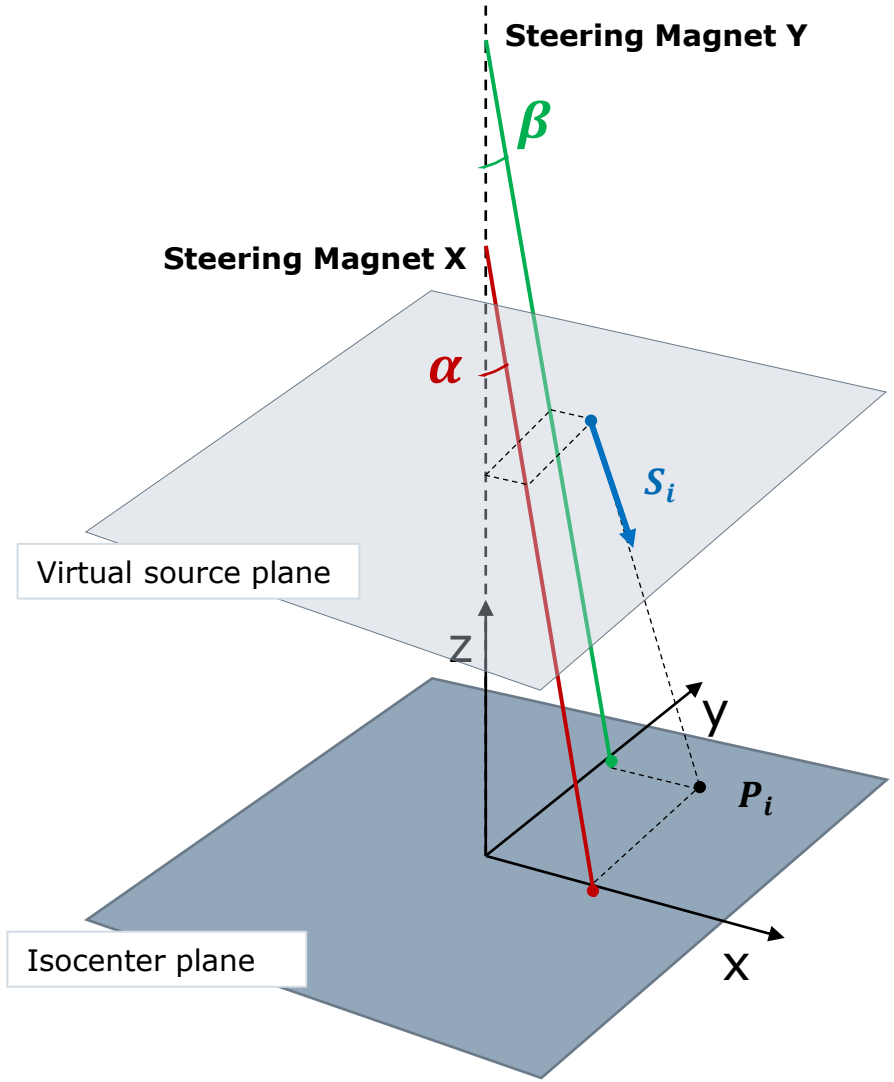
MEDICAL UNIVERSITY OF VIENNA

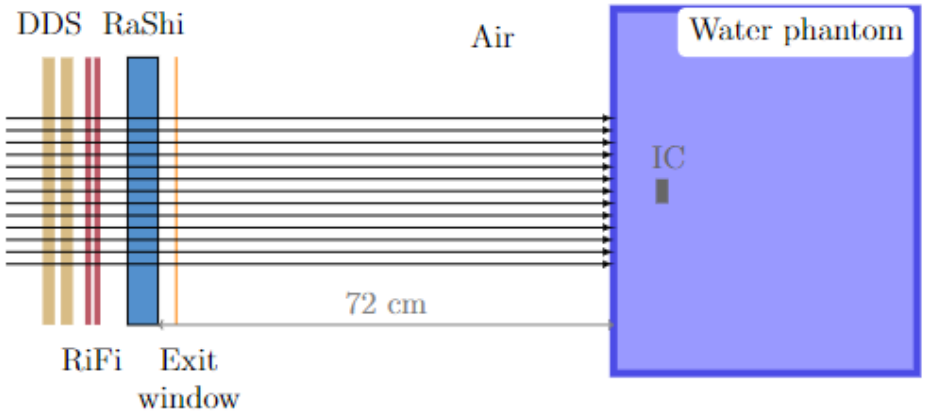
MedAustron Ion Therapy Center

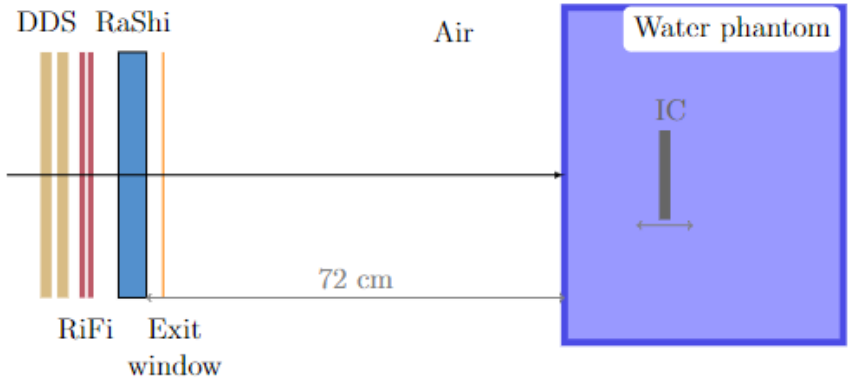
The financial support from the "Niederösterreichischen Wirtschafts- und Tourismusfonds" and European union in form of the EFRE fonds.

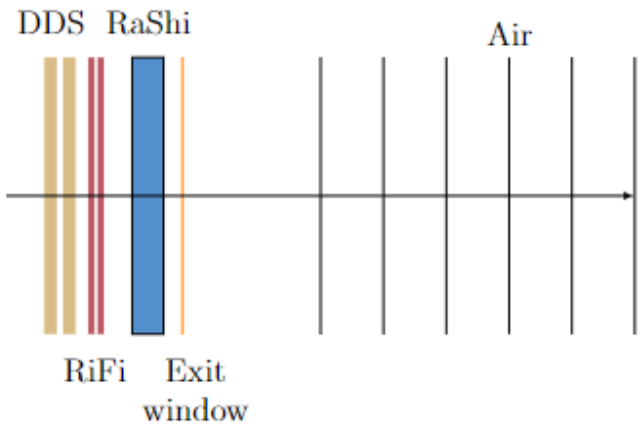
Implementation and features: (conventions are directly transferred from Gate 9)

- Each PBS is initialized at the nozzle exit and position and direction are calculated according to the beamline model provided, so that the beam reaches the planned spot.
- By default, the TPS is initialized at gantry angle 0 (vertical beamline).
- TPS rotation can be read from the treatment plan (if provided) or set manually.
- TPS translation is also possible.
- Each spot gets a fraction of the particles to simulate, proportional to the planned particles for that spot.




DDS
RaShi
Air
Water phantom
IC
72 cm
RiFi
Exit
window


DDS
RaShi
Air
Water phantom
IC
72 cm
RiFi
Exit
window


DDS
RaShi
Air
RiFi
Exit
window