An optimized Geant4 physics list for gamma radiotherapy

24th Geant4 Collaboration meeting, September 2019

P. Arce[1], J. I. Lagares[1], J. D. Azcona[2], P. B. Aguilar-Redondo[2]

[1]Unidad de Aplicaciones Médicas, CIEMAT

[2]Clínica Universitaria de Navarra

Outline

- Motivation
- Methodology
- CPU optimization
  - Optimize geometry
  - Study where time is spent
  - Optimize productions cuts & user limits
  - Bremsstrahlung splitting
  - Reuse particles that reach the phantom
  - Electromagnetic physics parameters
  - Multiple scattering models
- Beam parameters fitting
  - Centering and horizontality check of experimental data
  - Bremsstrahlung angular distributions
  - Resulting beam parameters
- Proposed physics list

Motivation

- Monte Carlo simulations in gamma radiotherapy are supposed to reach a **precision better than comercial Treatment Planning Systems, i.e. <u>1-2%</u>**

☹ But Geant4 simulations in radiotherapy are usually done without previously testing **which is the best (and CPU optimized)** physics list to reach this precision

☹ …And usually results are presented with **big error bars**

Methodology

Three setups, to widen the applicability of the resulting physics list

- VARIAN Clinac 2100 C/D 6 MV (E = 5.8 MeV)
- ELEKTA Versa HD 6 MV, without flattening filter (E=8 MeV)
- ELEKTA Versa HD 10 MV, without flattening filter (E=13.5 MeV)

1. Use standard way of **adjusting electron beam energy, width and dispersion angle in Monte Carlo to match experimental dose profiles in water**:
   - 8 jaws/MLCs apertures, from very small, 2x2 cm^2, to very big, 40x40 cm^2
   - 6 dose profiles: 1 Percentage Depth Dose, 5 transverse profiles, at dose peak and 5/10/20/30 cm depth
   - Small voxel size: 2x2x2 mm^3
   - Use precise devices for experimental data (0.5%)
   - **Calculate dose in Monte Carlo with 0.5% precision (~10^{13} events ~ O(100) CPU-years)**

CPU optimization

The dose precision of 0.5 % means **0.5-2 10^{13} events** $\Rightarrow$ Variance reduction is mandatory

- Geometry
- Production cuts and user limits
- Electromagnetic physics models
- Electromagnetic physics parameters
- Bremsstrahlung splitting
- Reusing of particles that reach the phantom

} Get the physics that best matches the exper. data and is optimal in CPU

But there are correlations among these parameters:

- Run with limited statistics to get a best guess of the parameters of each technique
- Adjust first those parameters that are more independent of other parameters
- If the final parameter value is far from its original value, and a strong correlation is foreseen with another parameter previously fixed, repeat the optimization of that fixed parameter

Optimize geometry

- ✓ Kill particles that exit the target not facing the phantom

  => CPU savings: ~5%/5%/1% (VARIAN 6, ELEKTA 6, ELEKTA 10)

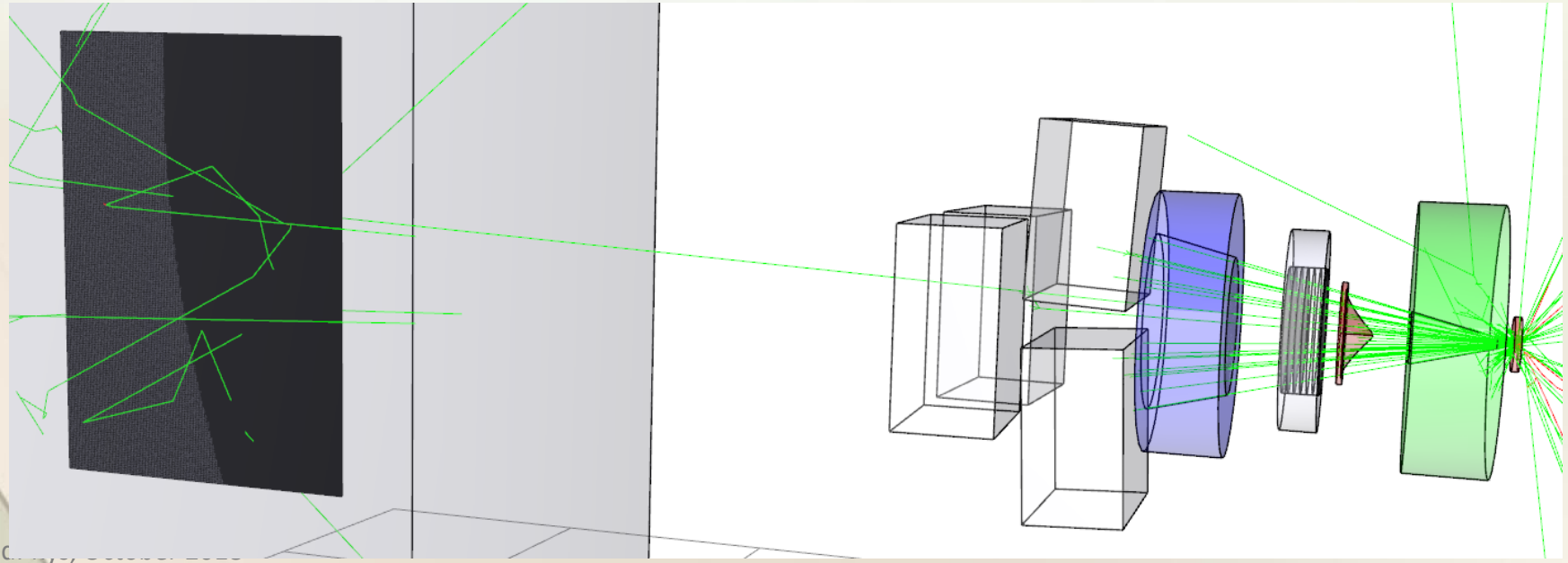
- ✓ Do not voxelize all phantom but only 2D planes where exp. data is taken

  => CPU savings: factor ~3/1.9/2.5

  - ➢ Create an special cross phantom in GAMOS

- ➢ Use Geant4 regular navigation with skipping contiguous voxels that have the same material

  => CPU savings: ~8%/8%/17%



Study where the time is spent

- It should be the first step, and it is quite easy in GAMOS:

/gamos/classifier partRegC GmCompoundClassifier GmClassifierByParticle GmClassifierByRegion

/gamos/userAction GmTimeStudyUA partRegC

	VARIAN 6MV	ELEKTA 6MV FFF	ELEKTA 10MV FFF
Particle / geometry region	Time / Total Time	Time / Total Time	Time / Total Time
e+ TOTAL	0.2 %	0.8 %	2.0 %
e- / target	79.0 %	58.0%	48.0 %
e- / other linac components	4.5 %	10.7 %	12.6 %
e- / outside linac components	0.6 %	0.8 %	1.5 %
e- TOTAL	84.1 %	69.2 %	62.6 %
gamma / target	7.8 %	15.4 %	15.1%
gamma / other linac components	4.7 %	10.0 %	7.4 %
gamma / outside linac components	3.4 %	4.6 %	12.9 %
gamma TOTAL	15.8 %	30.0 %	35.4 %

Optimize production cuts

It is usually a lengthy procedure:

- ☹ Have to try different sets of cuts per particle and region
- ☹ Need big statistics to observe small effects (~0.5%)
- ➔ **Too lengthy and difficult : nobody does it**

GAMOS proposes a method to automatically do it
with limited statistics
& in one job

Based on tagging all particles and calculating the effect they would have it they were not created because their falling below a cut

Similar for user limits, but in our case they do not have any sensible effect

Optimize production cuts in accelerator regions

Choose as cut values the highest ones that do not sensibly diminish the number of particles that reach the phantom (define XY plane at the entrance of the phantom)

```
/gamos/geometry/createRegion targetReg *TARGET*
/gamos/geometry/createRegion collimatorReg *COLLIMATOR*
….
/gamos/setParam RTCutsStudyFilter:PlaneZ -1000
/gamos/setParam RTCutsStudyFilter:PlaneXDim 150
/gamos/setParam RTCutsStudyFilter:PlaneYDim 150
/gamos/userAction GmProdCutsStudyUA RTCutsStudyFilter
```

=> CPU savings: ~25%/25%/25%

Geometry region	Prod. Cuts e-'s (mm) VARIAN 6/ELEKTA 6/10	Prod. Cuts gammas (mm) VARIAN 6/ELEKTA 6/10
Target	**1./1./0.1**	**10/10/3**
Primary collimator	**0.1/0.1/0.1**	**10./10./1.**
Flattening filter	**1.//**	**10.//**
Jaws/MLCs	**0.3/0.3/0.1**	**1./1./0.3**
Outside linac components	**10./10./1.**	**10./10./1.**

Optimize production cuts in phantom

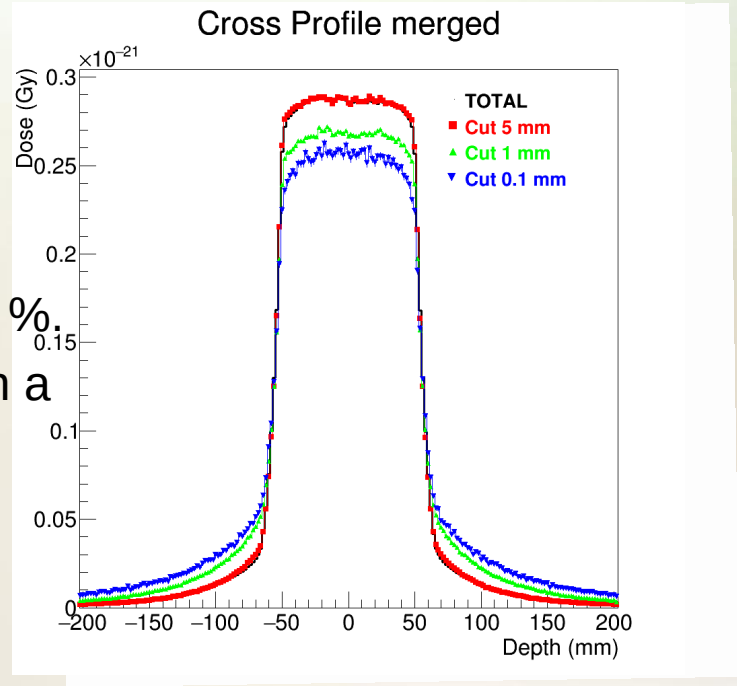
Choose highest cuts so not to kill but a small fraction of the particles that leave a fraction of the dose in a voxel different to the one where they are created

- /gamos/filter ProdCutFilterC5.5 GmProdCutOutsideVoxelFilter 5.*mm 5.*mm
- /gamos/scoring/addFilter2Scorer ProdCutFilterC5.5 doseScorerC5.5

➢ **Using dose histograms we can refine the method: if the dose lost by a cut has exactly the same shape than the total dose, the effect of this cut would be zero, because a simple renormalization would provide the same fit to the data:**

- Subtract the total dose minus the dose lost by a cut
- Normalize this dose to the total dose
- Subtract the total dose minus this normalized dose
- Check that this new dose distribution does not have any bin above 0.5 %.
- (This is quite a strict condition, as the statistical fluctuations might push a
- point above this value)

Cut value (mm) : 0.5/0.5/0.5



=> CPU savings: ~6%/7%/5%

EM physics parameters

There are many physics parameters that may affect the results and/or the CPU

- **Multiple scattering**
  - **Step limitation type**
  - ***RangeFactor***
  - ***GeomFactor***
  - ***Skin***
- **Energy loss**
  - ***RoverRange***
  - ***FinalRange***
- ***Data tables bins per decade***
- Auger
- Fluorescence
- lowKinE data tables
- Limit linear interpolation

EM physics parameters

How to optimize all these parameters (huge correlation among them)?

1. *Choose **very precise configuration***
   - *UseSafetyPlus* step limitation algorithm (recommended by M. Novak)
   - *RoverRange* = 0.01, *FinalRange* = 0.01 mm, *RangeFactor* = 0.01, *GeomFactor* = 2.5 and *Skin* = 3.
2. Set the rule of **not trying very different values of two parameters that would make that one of them is almost never appli**ed
3. Classify the possible combinations by the **average number of steps that electrons make** (what can be quickly calculated with good precision)
4. Make **dose calculation only for a few combinations of the parameters** that are separated in this classification table, comparing with precise parameters

EM physics parameters

5. Choose the **set of parameters that have the smallest CPU time** consumption while having a negligible difference in dose
6. **Test a few other parameter combinations** that have a similar number of electron steps that the one selected in the first phase, **to refine the optimization of CPU time performance**
7. Finally, **repeat optimization for the other three step limit algorith**ms: U*seSafety*, *UseDistanceToBoundary* and *Minimal*
   - Reduce the CPU time only a few % but have a non-negligible influence in the dose.

EM physics parameters

Parameter name	Precise value	Default Value
mscStepLimit	UseSafetyPlus	UseDistanceToBoundary
RoverRange	0.2	0.2
FinalRange	0.1	1.
RangeFactor	0.05	0.04
GeomFactor	2.5	2.5
Skin	3	1
lowKinE	1 keV	1 keV
binsDecade	20	7
linLossLimit	0.01	0.01

CPU gain (loss) by a factor of is -10%/-34%/-20%

Official Geant4 EM physics lists

	RESULT vs optimized	Ratio CPU time / optimized
G4EmLivermore	very similar	**1.6/1.6**
G4EmLowEP	LowEP Compton	1.5/1.4
G4EmPenelope	thinner angular distribution	1.2/1.1
G4EmStandard_opt0	dipBust angular brems	1.1/1.2
G4EmStandard_opt1	dipBust angular brems	0.35/0.4
G4EmStandard_opt2	too big steps	1.0/1.0
G4EmStandard_opt3	very similar	**1.3/1.2**
G4EmStandard_opt4	very similar	**2.4/2.4**

Bremsstrahlung splitting

Three different bremsstrahlung techniques developed in GAMOS

- Uniform bremsstrahlung splitting
- Directional bremsstrahlung splitting: only create brems. gammas if they are directed towards phantom (towards an XY rectangle)
- Equal-weight bremsstrahlung splitting: reweight each particle (from any process) so that gammas that arrive in the phantom has same weight
- + option to only bias primary particles
- + option to only bias once each particle

First fit XY rectangle for DBS & EWBS :

+20 cm more than field size

X / Y (mm)	DBS Relative Efficiency	EWBS Relative Efficiency
1000 / 1000	1	1.18
600 / 1000	1.21	1.18
400 / 800	1.64	1.20
200 / 600	2.70	1.06
150 / 550	-	1.01
100 / 500	-	1
50 / 450	-	1.50
2 / 400	2.61	11.1
2 / 50	3.46	16.6

Fit splitting number

	N splitting	Relative efficiency
No splitting	-	1
UBS	10	3.0
	100	3.8
	1000	0.2
DBS	10	3.3
	100	3.6
	1000	1.6
EWBS	10	3.5/4.0/2.1
	50	4.6/4.2/**2.4**
	100	**5.0**/**4.6**/2.3
	300	3.9/3.9/1.5
	1000	3.6/5e-4/4e-4

Reuse particles that reach the phantom

Usually this is done by writing a phase space with the particles that traverse the accelerator and reading them back

- GAMOS can do it in one go, without writing the phase space
  - ✓ No need of disk space
  - ✓ One job instead of two
  - ✓ Avoid the CPU penalty of writing and reading very big files

=> CPU savings: ~12%

N Reusing	Relative efficiency
-	1
10	6
25	22
50	33
100	48
250	55
500	49
1000	28
10000	5
100000	0.6

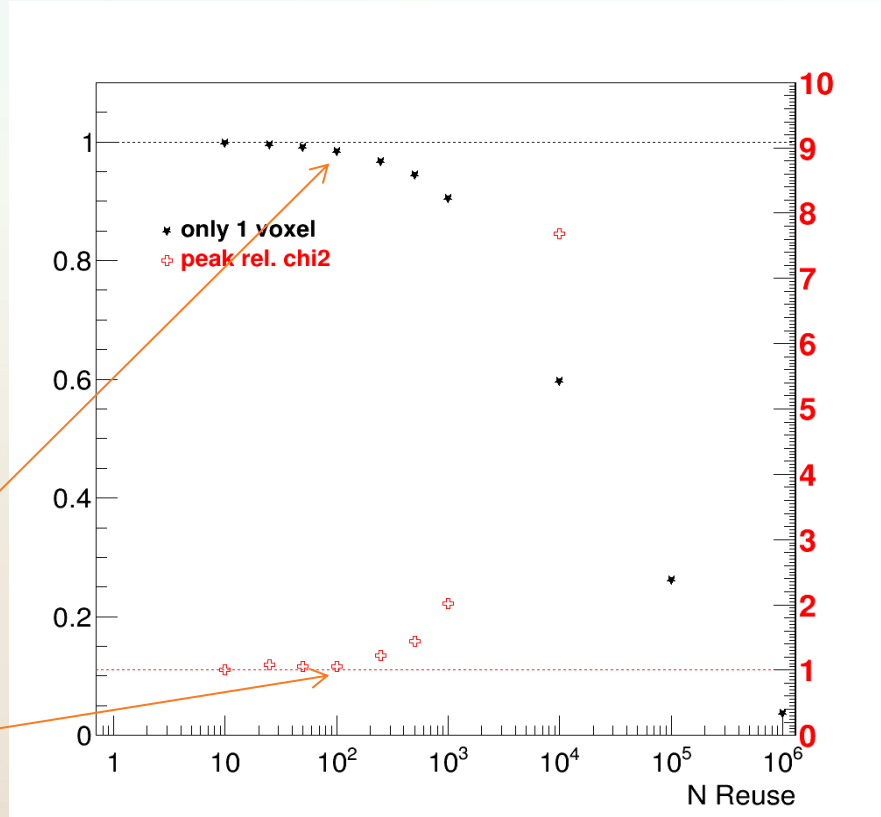
CPU gain:

X 48 / X 14 / X 12

Reuse particles that reach the phantom

But:
- The particle reuse should not be increased beyond the point at which the fraction of the total computer time spent in traversing the accelerator becomes relatively small compared to the time spent in dose calculations
- or to the point that the geometrical dose distributions created by different reused photons greatly overlap
- or, take into account the influence of the limited statistics: the use of a high number could produce an artificial peak in some region that would only be wiped out if the statistics are high enough. Therefore, to better understand the effect of reusing we have made two studies.

- Fraction of voxels that have dose depositions by only one of the reused photons
- Look for the presence of artificial peaks in the dose distributions in the cross profiles, making a pseudo chi-square



N=100 is safer than N=250

Beam parameters fitting

- We have to change the Monte Carlo beam parameters (energy, width and dispersion angle) to best fit the experimental data
- Big correlation with some EM parameters and models (mainly bremsstrahlung angular distributions)

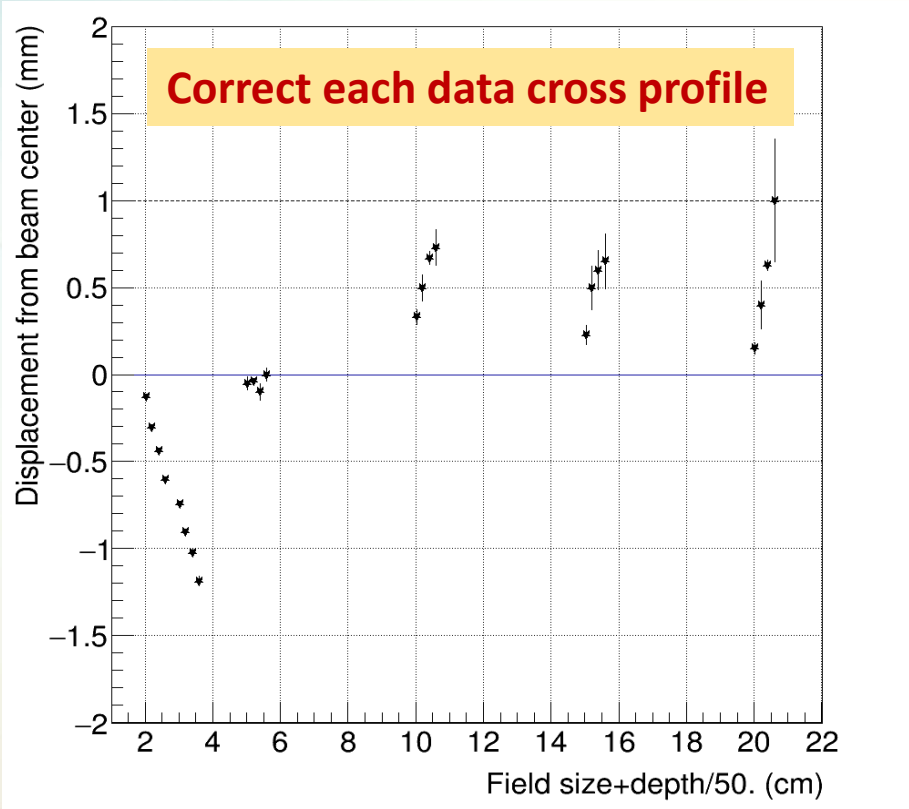
The method is:

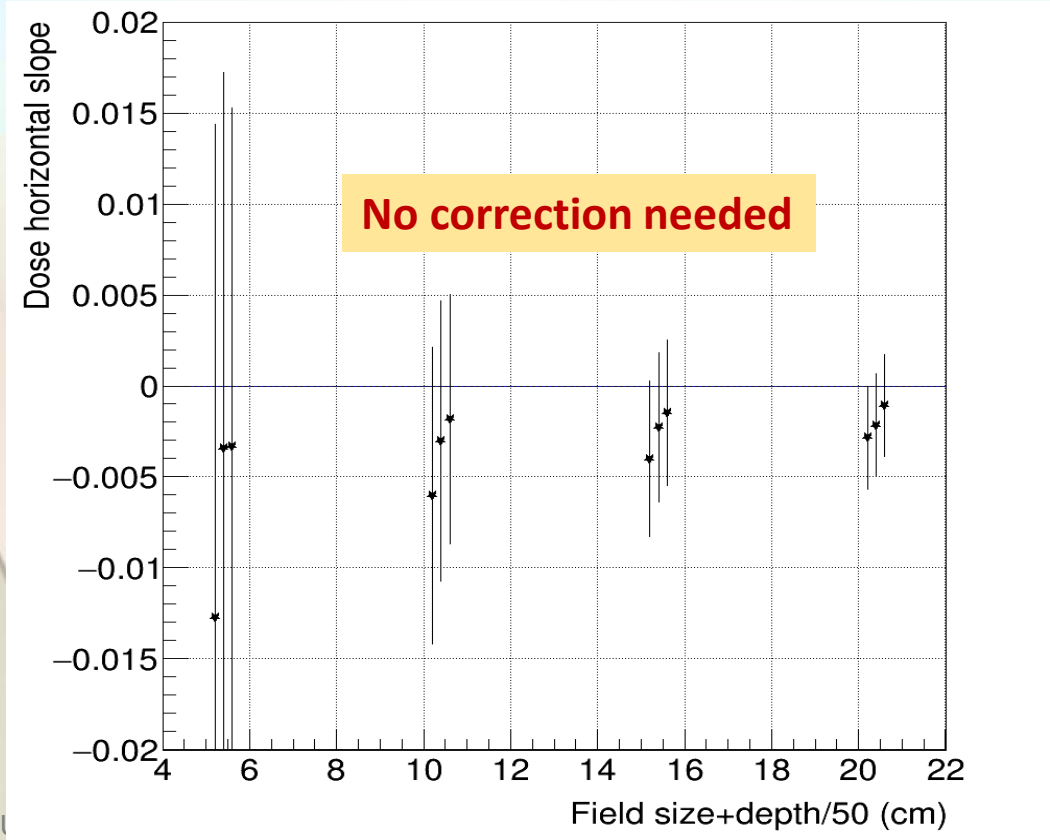
1. We select what we think is a good set of EM parameters and models and fit the experimental data
2. We change one EM parameter/model and if the fit is better, we refit the beam parameters

We do this with a 10x10 cm^2 and study PDD and the 5 cross-profiles

Correct entering and horizontality of exper. data

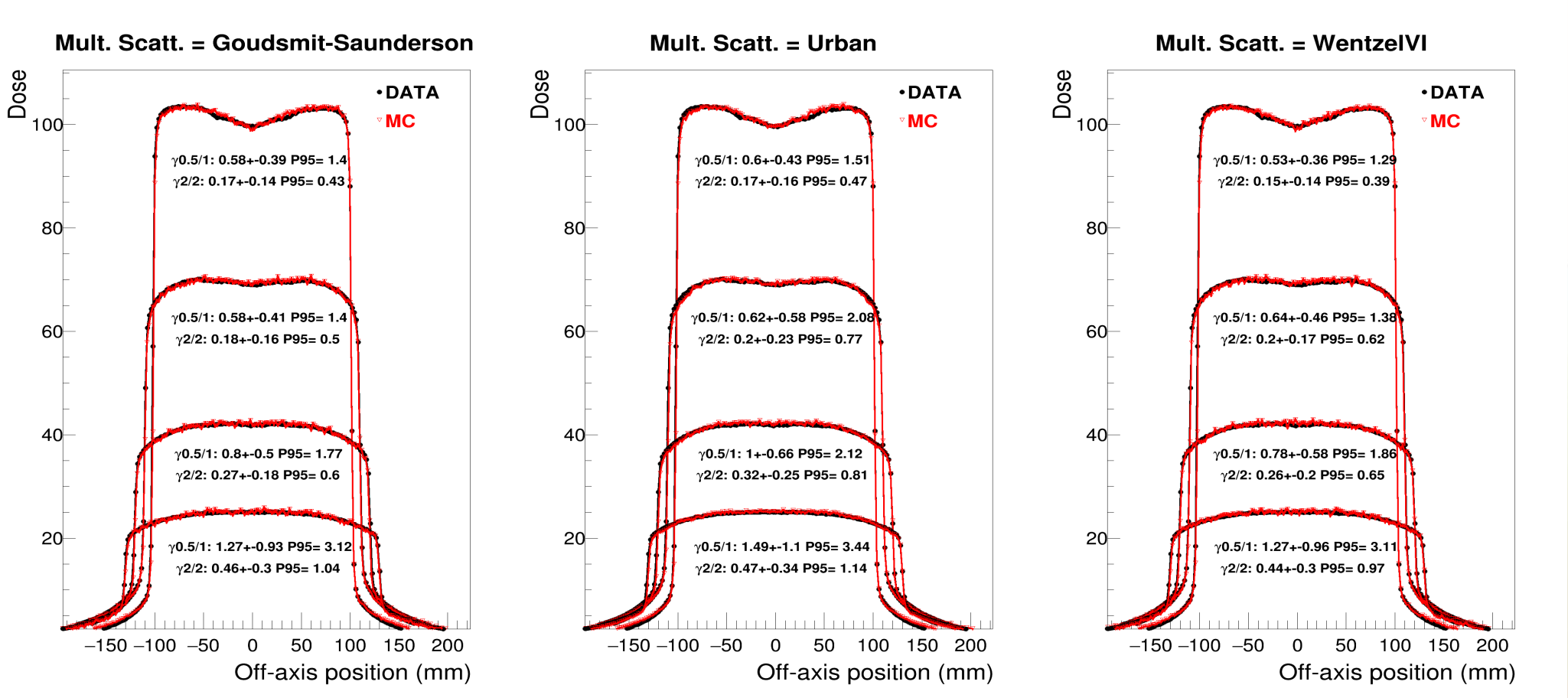
- **Centering:** compare half left and half right of data dose cross profiles





- **Horizontality:** Subtract the half left minus the mirrored half right of data dose cross profiles and fit to a straight line gives no slope (after centering)

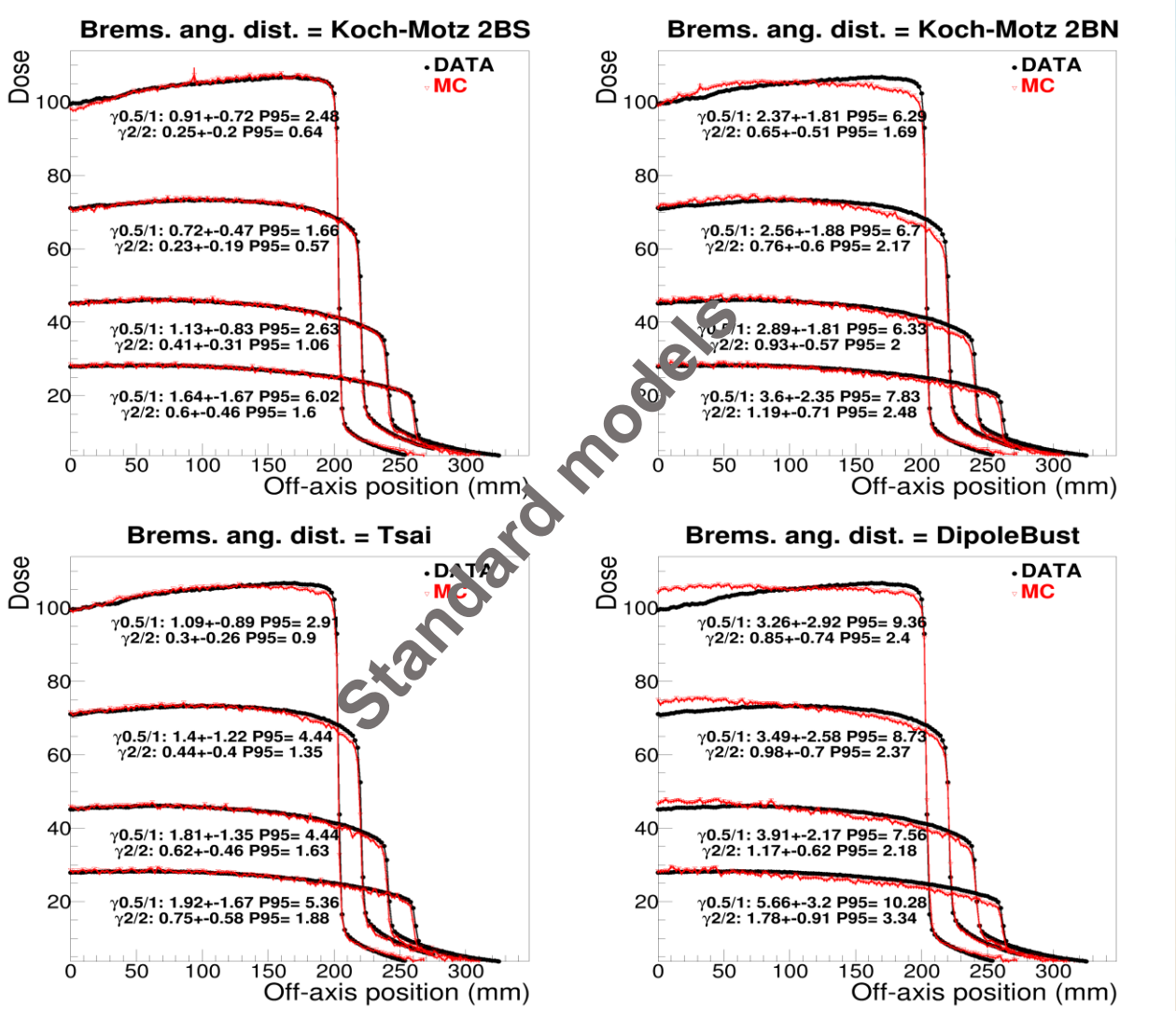
Multiple scattering models



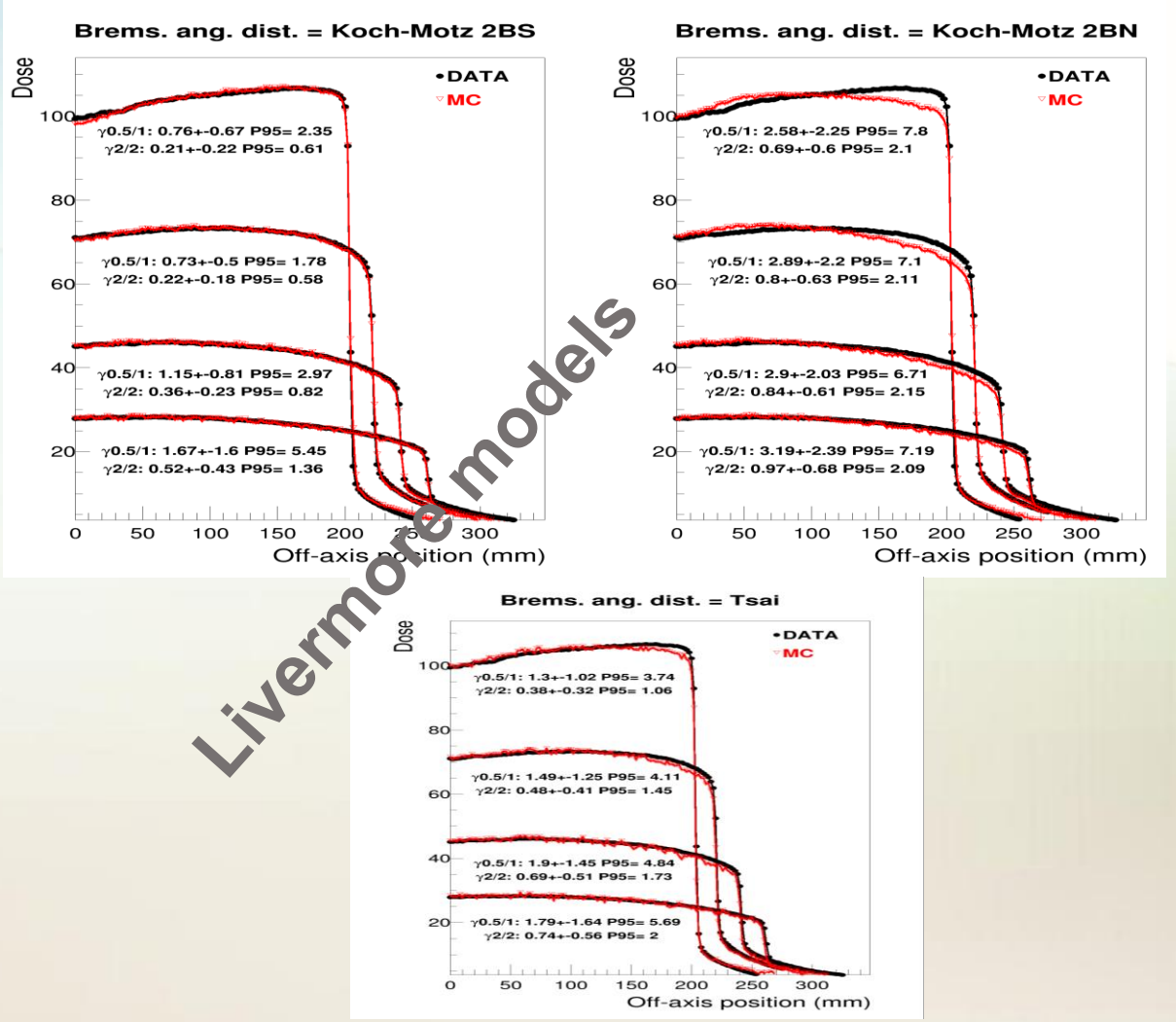
CPU ratio: **1.11** **1.** **1.19**

No difference, Urban is fastest

Bremsstrahlung angular distributions



Standard models

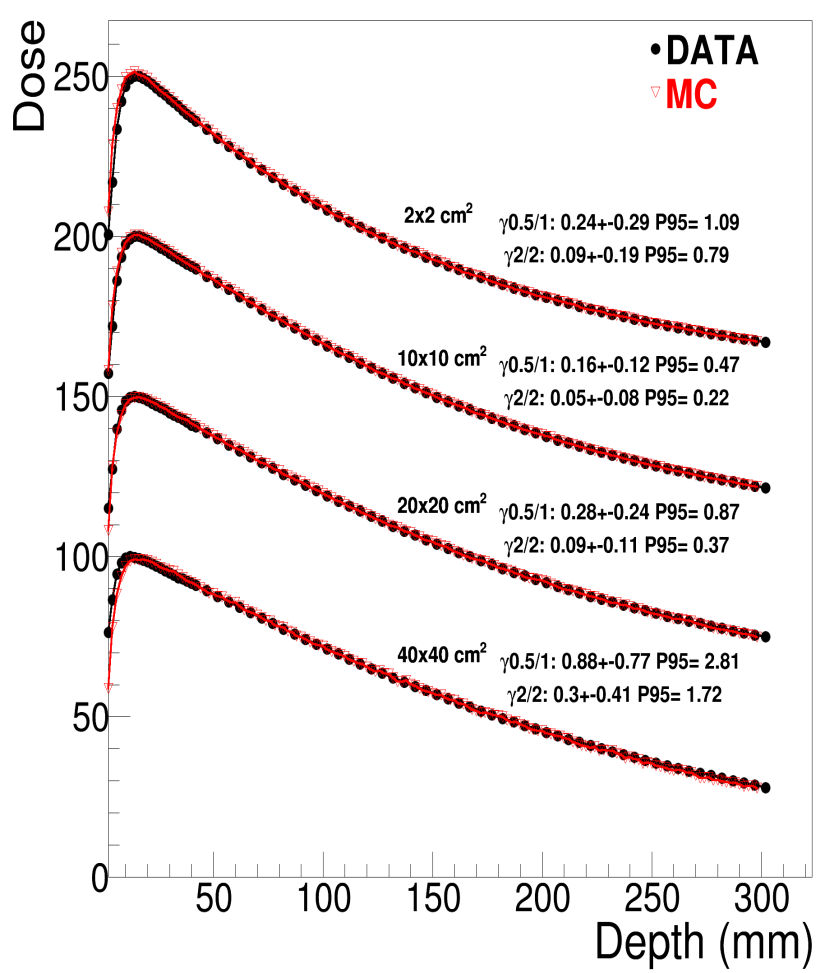


Livermore models

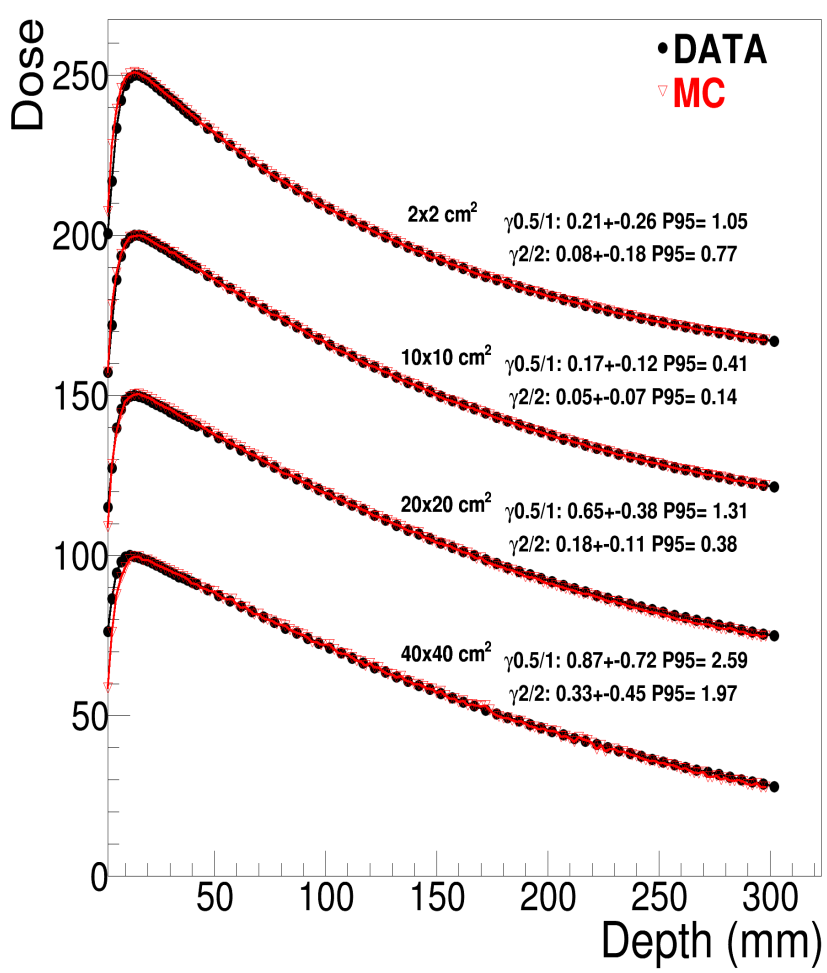
Tsai best for low angles, 2BS best for high angles

Beam parameters fitting: PDD

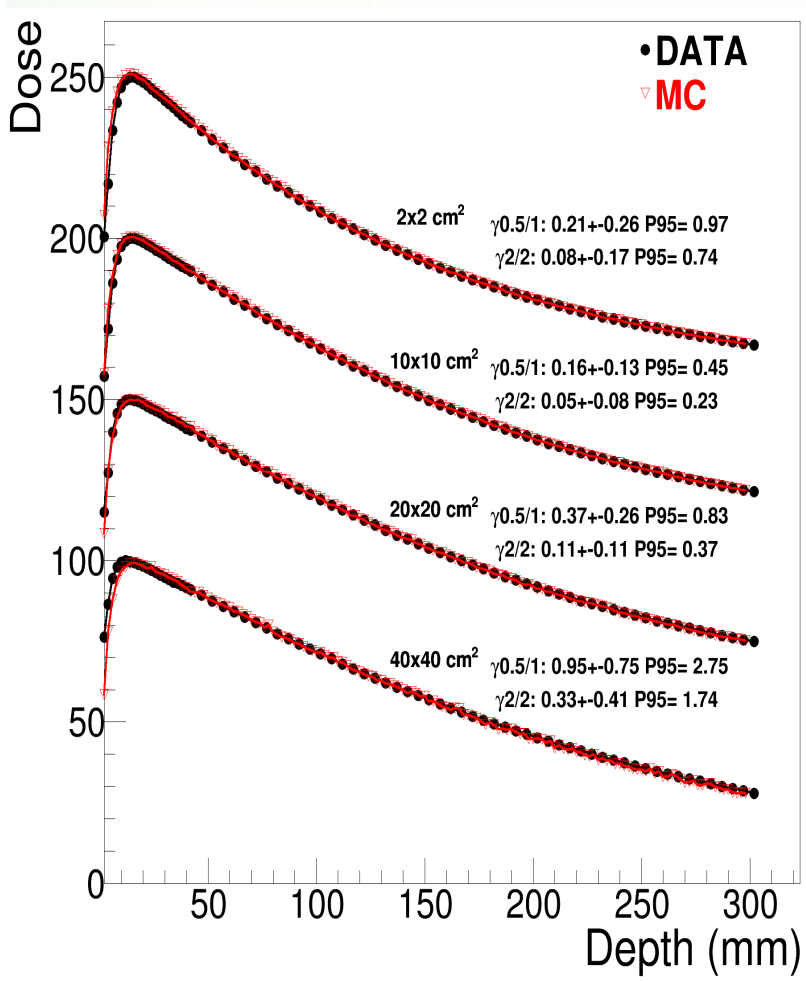
Standard



Livermore

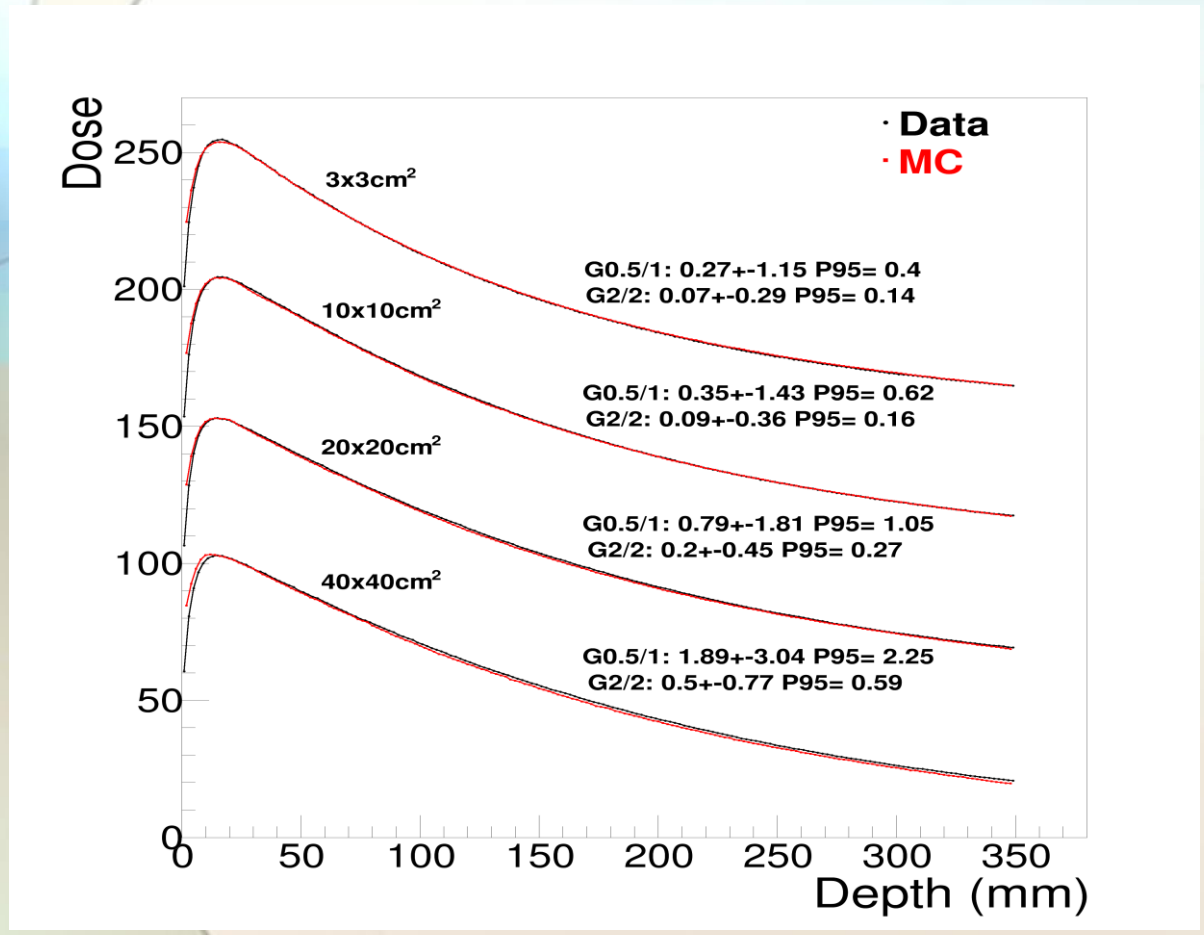


Penelope

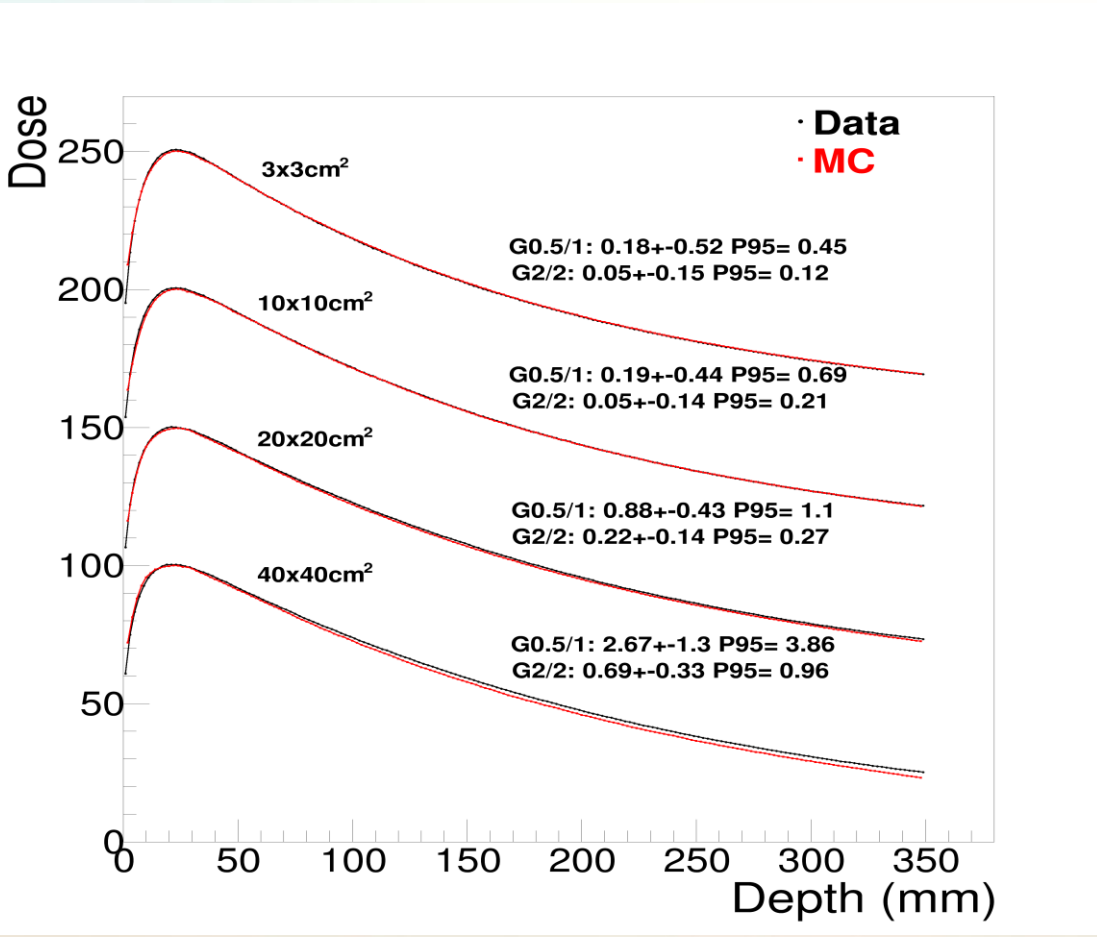


Beam parameters fitting: PDD

Standard 6 MV FFF



Standard 10 MV FFF

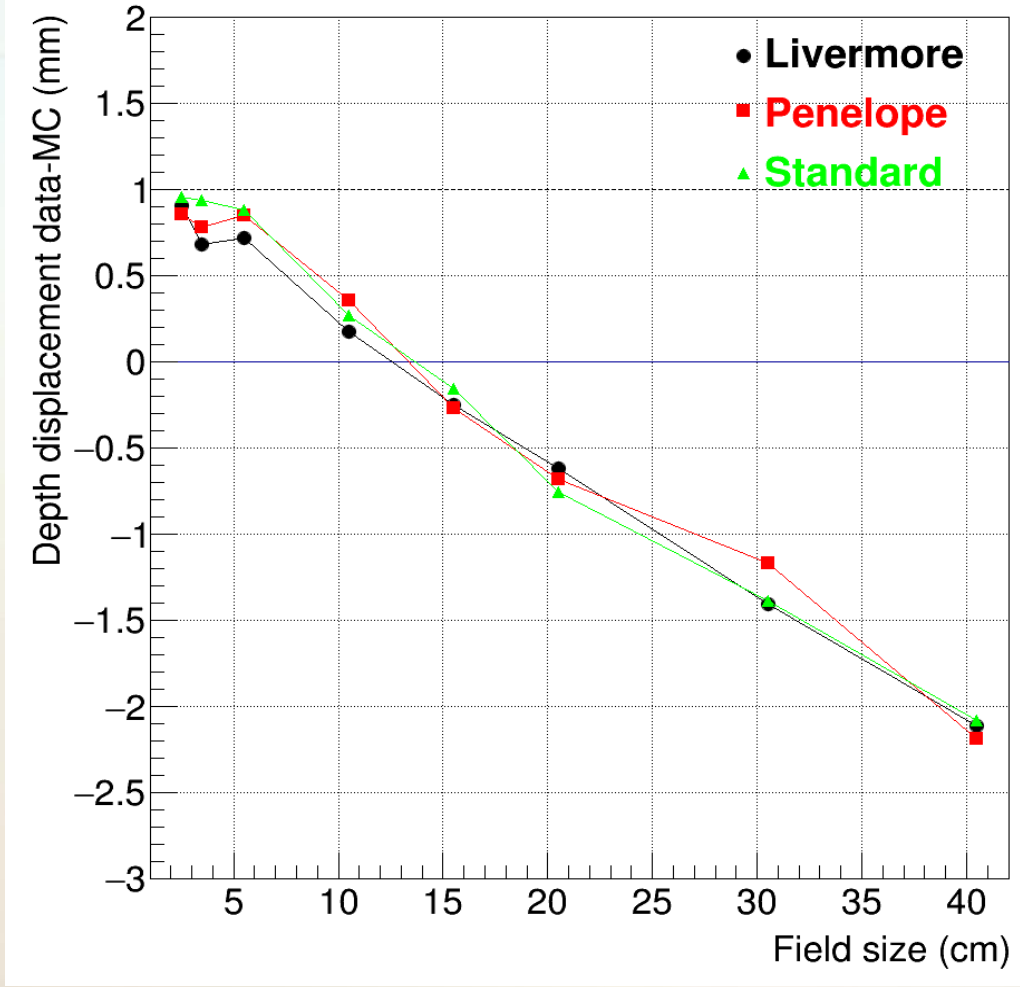


Beam parameters fitting: PDD

☺ Quite a good match Monte Carlo – experimental data

☹ But dose peak is slightly displaced

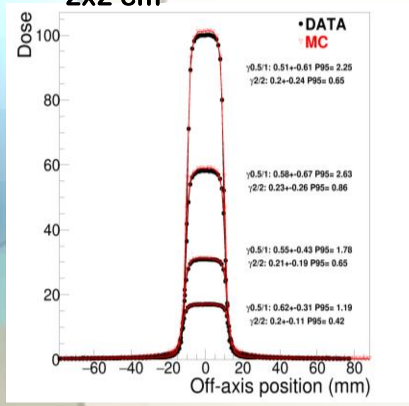
- ☹ Displacement vs. field size fits to a line value follows a line with slope ~0.008 mm^{-1}
- ☹ Similar for the three EM physics models



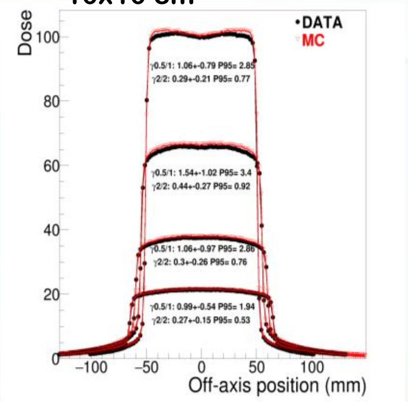
Beam parameters fitting: cross profiles

Standard

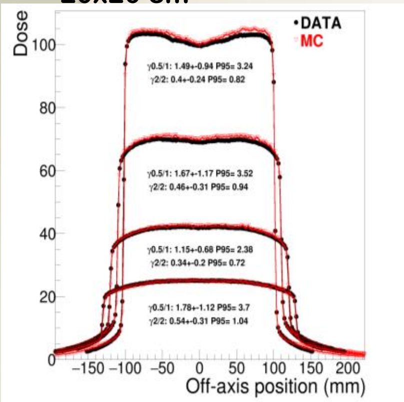
2x2 cm^2



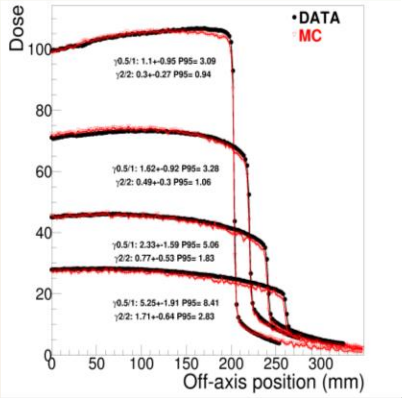
10x10 cm^2



20x20 cm^2

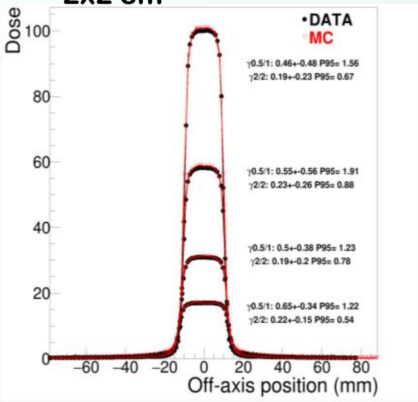


40x40 cm^2

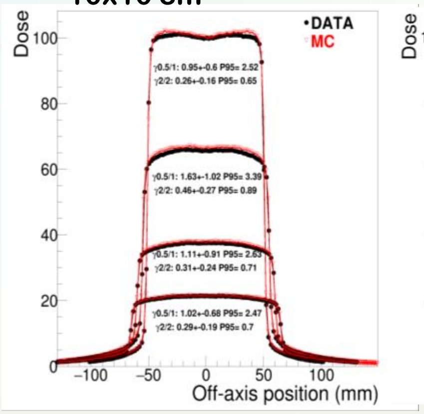


Livermore

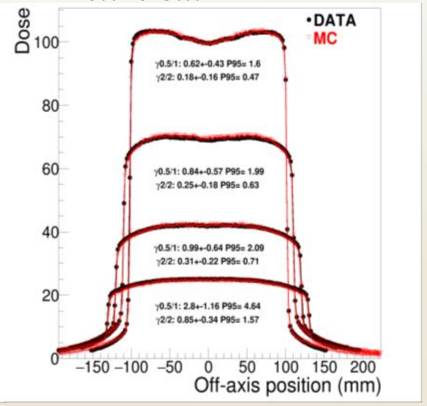
2x2 cm^2



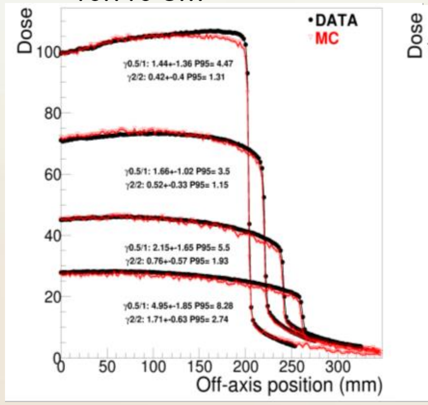
10x10 cm^2



20x20 cm^2

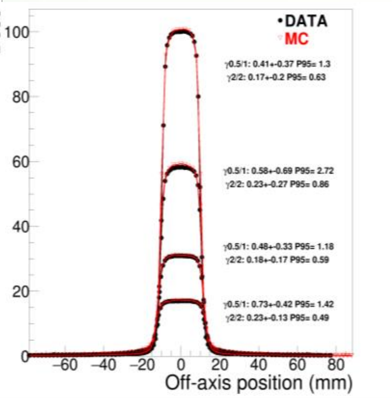


40x40 cm^2

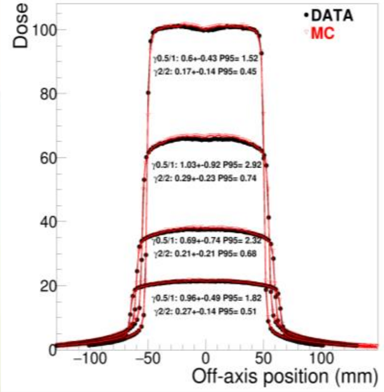


Penelope

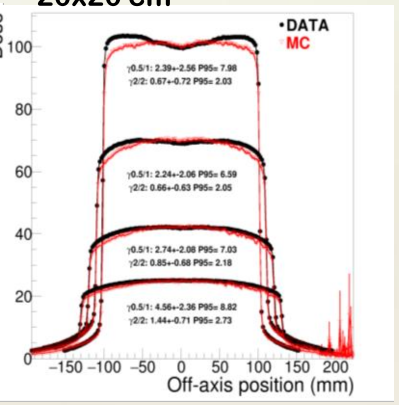
2x2 cm^2



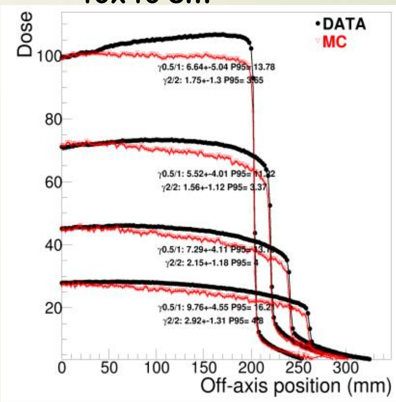
10x10 cm^2



20x20 cm^2

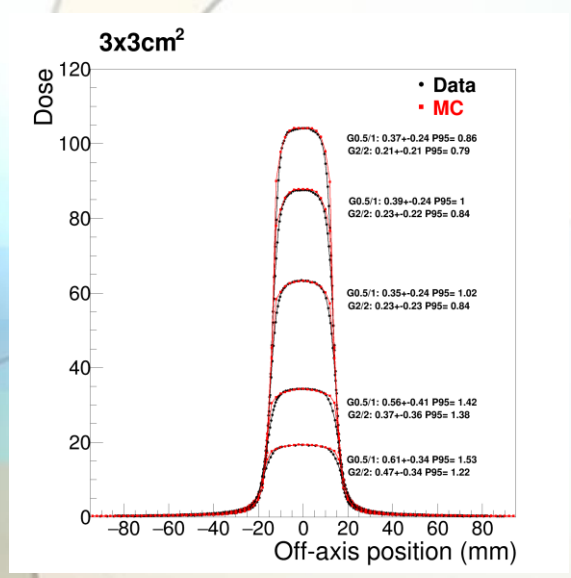


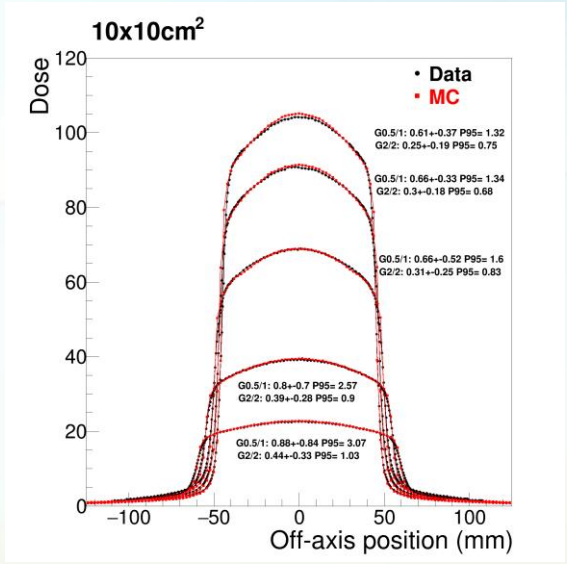
40x40 cm^2

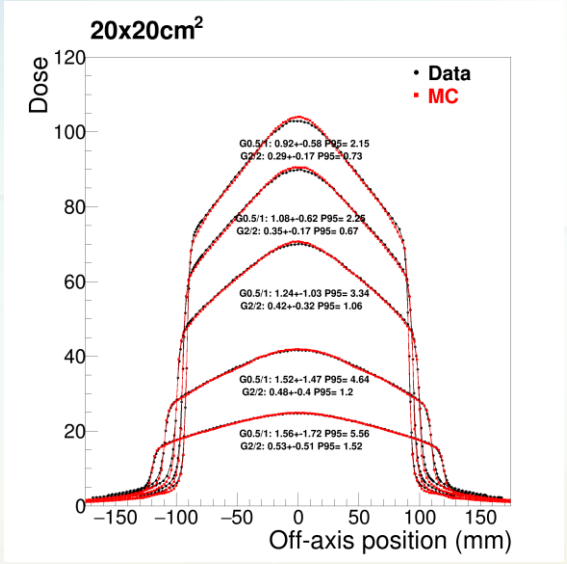


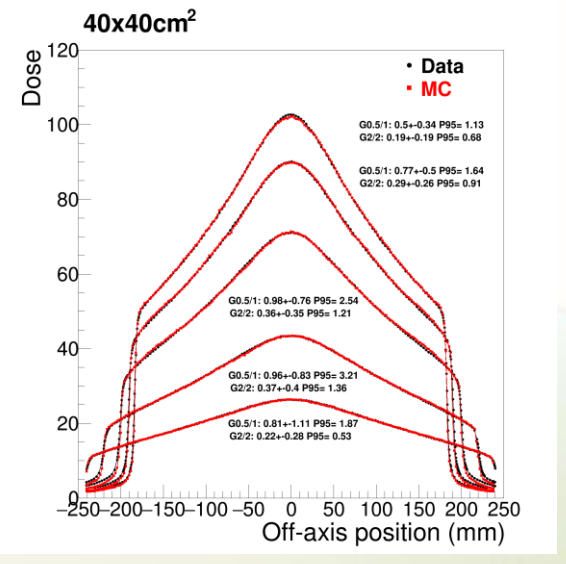
Beam parameters fitting: cross profiles

Standard ELEKTA 6 MV FFF

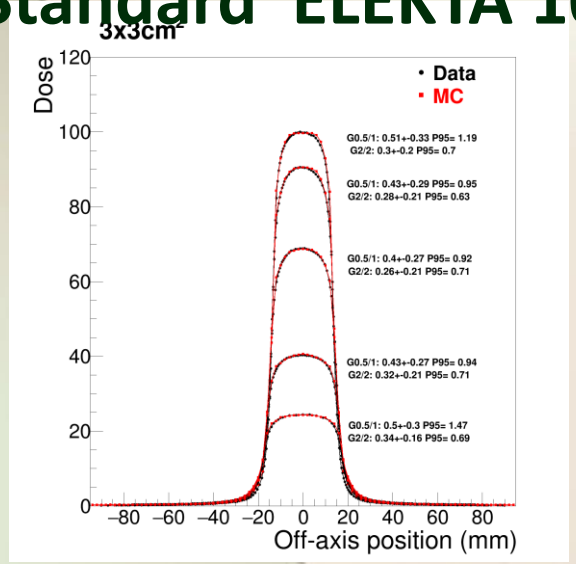


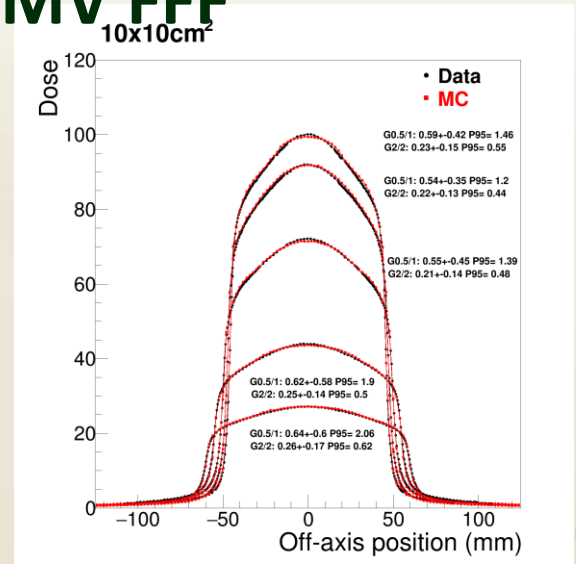


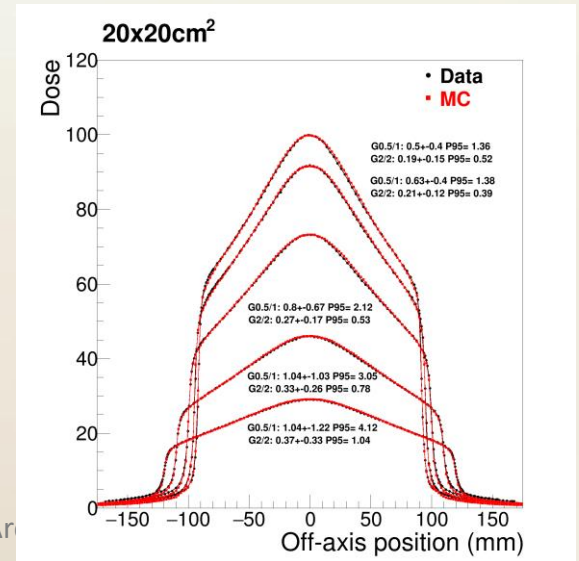


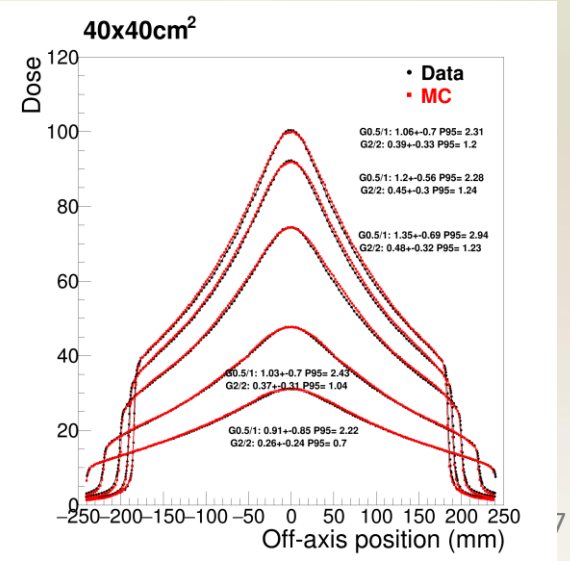


Standard ELEKTA 10 MV FFF









Conclusions

✓We have validated the GAMOS/Geant4 simulation for three gamma radiotherapy accelerators/energies

Several aspects should be highlighted in this work:

- The high precision of the data and Monte Carlo dose distributions: 0.5 % for 2x2x2 mm^3 voxels
- The big number of dose distributions used: one PDD and five cross profiles for 8 different fields, from 2x2 cm^2 to 40x40 cm^2
- The thorough optimization of the CPU time, which served to **improve the efficiency by a factor of around 525 / 350 / 225**

Proposed physics list for gamma radiotherapy

- **Electromagnetic physics models**: **standard**
- **Multiple scattering model**: **Urban** (DEFAULT)
- **Bremsstrahlung angular distribution**: **Tsai for small fields** (<20x20 cm), **Koch-Motz 2BS for big fields** (>20x20 cm)

- **Multiple scattering step limitation type = UseSafetyPlus**
- ***Msc RangeFactor***: **0.05**
- ***Msc GeomFactor***: **2.5**
- ***Msc Skin***: **3**
- **Eloss StepFunction**): **0.2/1** (DEFAULT)
- **Eloss & range tables bins per decade of energy for the energy loss and range tables**: **20** (DEFAULT)
- ***lowKinE***: **1 keV**
- ***linLossLimit***: **0.01**

➢ Quite close to G4EmStandard_option3 (factor 1.25 faster)