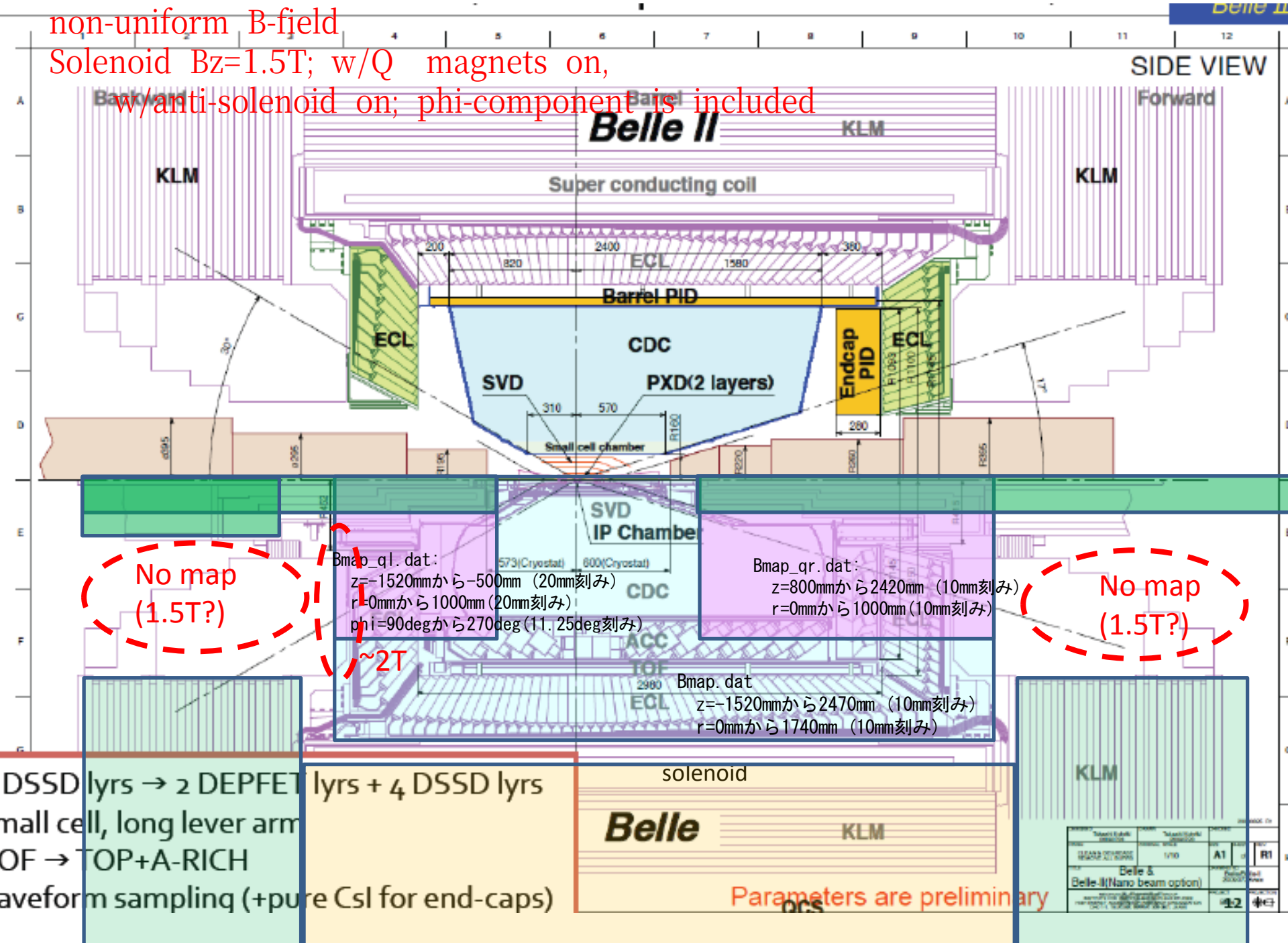
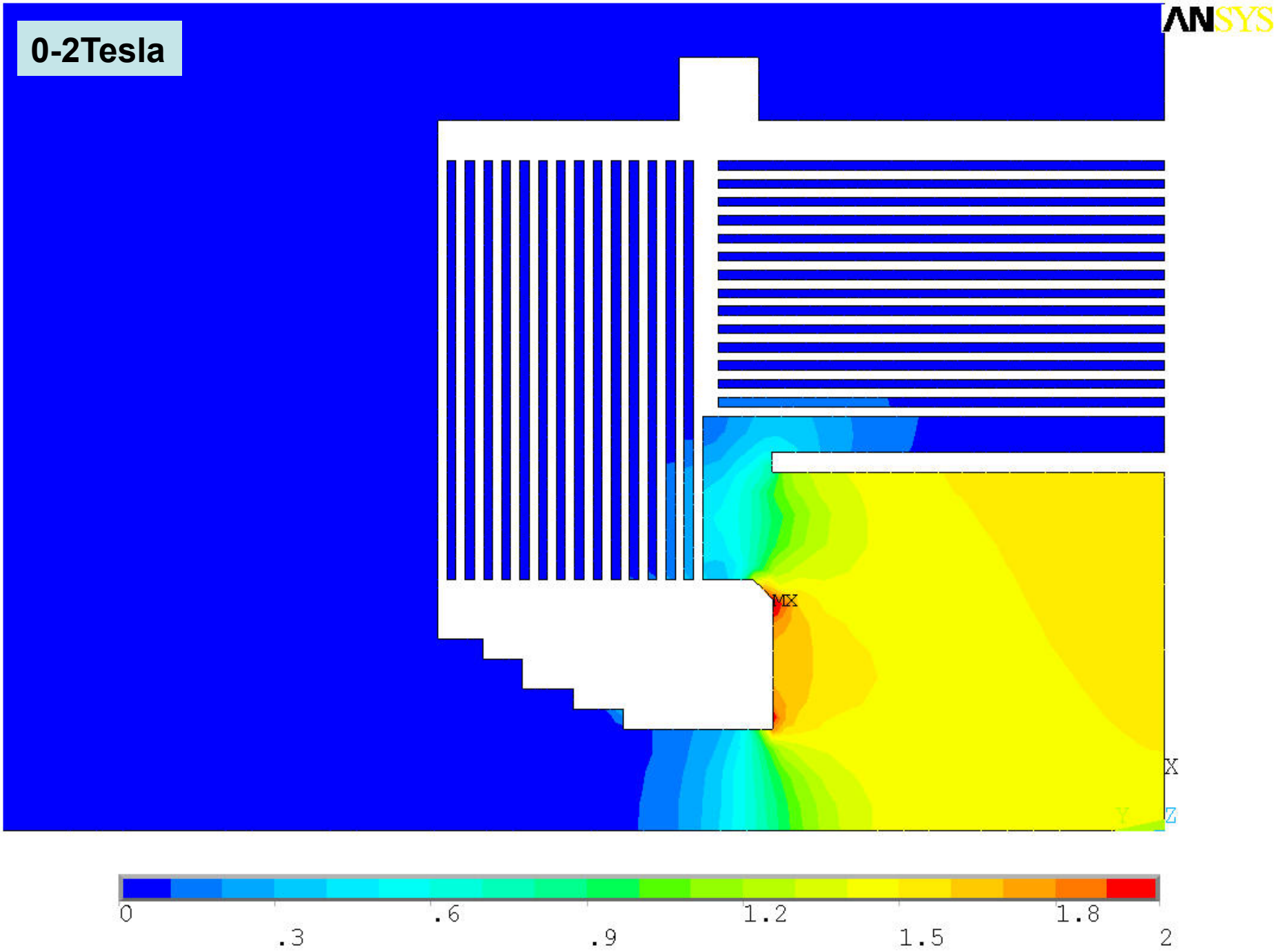
non-uniform B-field
Solenoid Bz=1.5T; w/Q magnets on,
w/anti-solenoid on; phi-component is included




0-2Tesla
ANSYS
MX
X
Y
Z
0
.3
.6
.9
1.2
1.5
1.8
2

likewise geometry, different

Belle

discrete B-field maps

B-field 1.5T is hardcoded in tracking tools (trasan, trak)

mesh size : 1cm in r and z, no ϕ-dependence

70000 lines : ~2MB

cylindrical coord.

Possible improvement

unified method to get the B-field (for sim/recon)

200MB need to be loaded
when the process begins ???
(3000000 lines)

need to optimize the mesh size

B-field in detector acceptance (1cm in r, 2cm in z, 5deg in ϕ)

B-field at around QCS (calcurate in each step ?)

need finer map ?

better to handle several B-field maps (?)

Single π+ p_t=2.0GeV/c, θ = 90deg, produced @ IP (0,0,0)

Belle B-field

gsim : map# 21
trak : map# 21

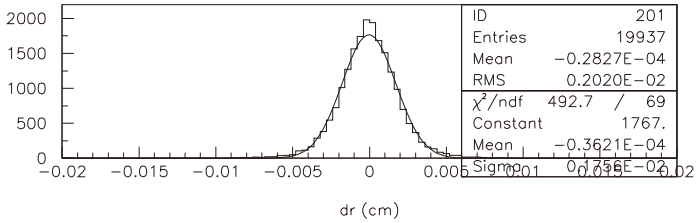
Belle II HC-opt.

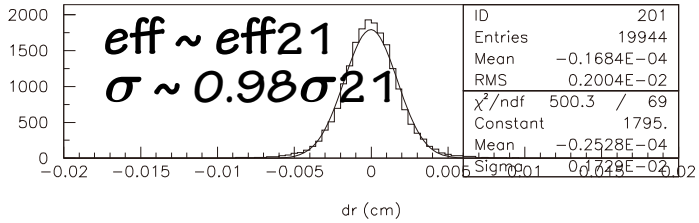
gsim : map# 31
trak : map# 31

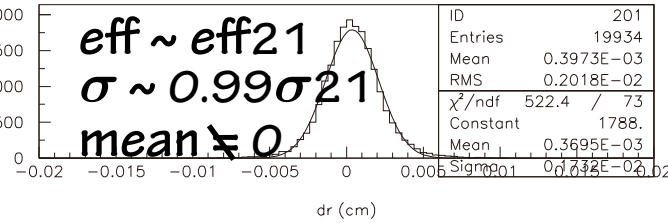
mismatched B

gsim : map# 31
trak : map# 21

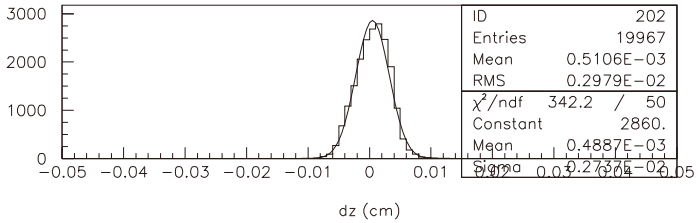
dr

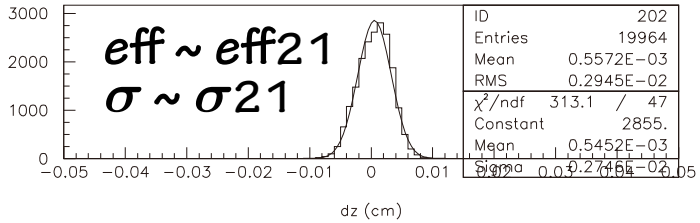


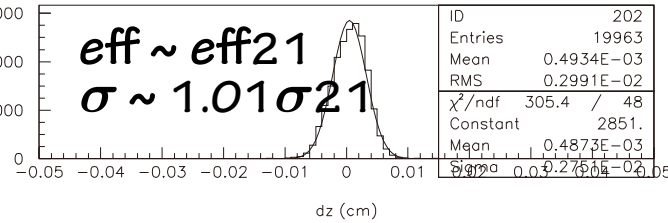




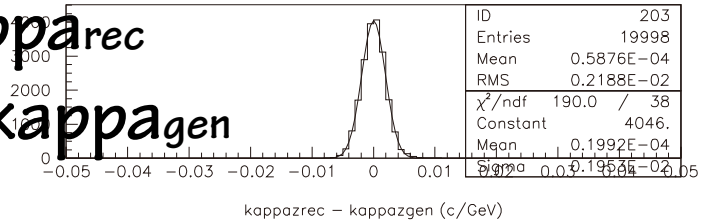
dz

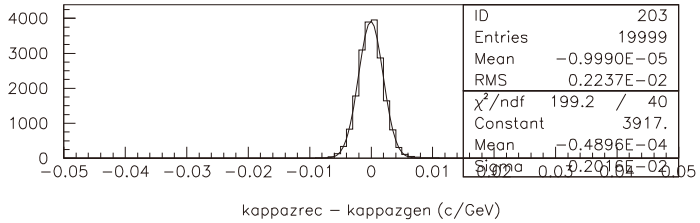


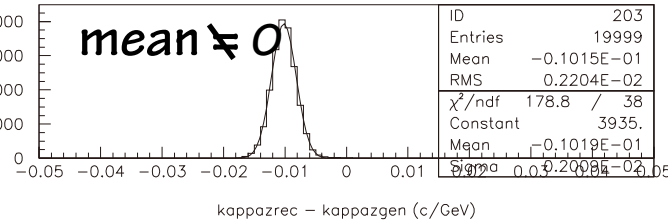




$kappa_{rec}$ - $kappa_{gen}$







Single π+ p_t=2.0GeV/c, θ = 20deg, produced @ IP (0,0,0)

Belle B-field

gsim : map# 21
trak : map# 21

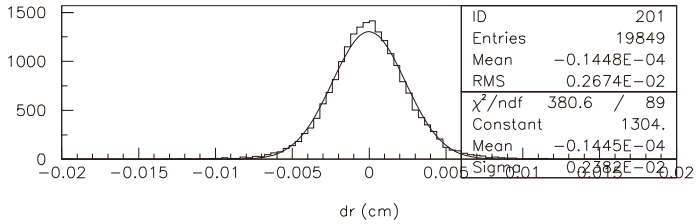
Belle II HC-opt.

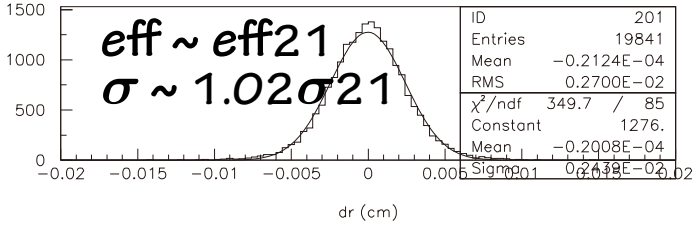
gsim : map# 31
trak : map# 31

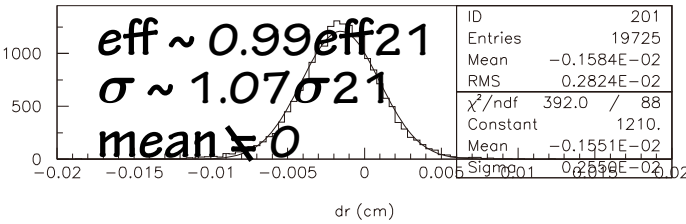
mismatched B

gsim : map# 31
trak : map# 21

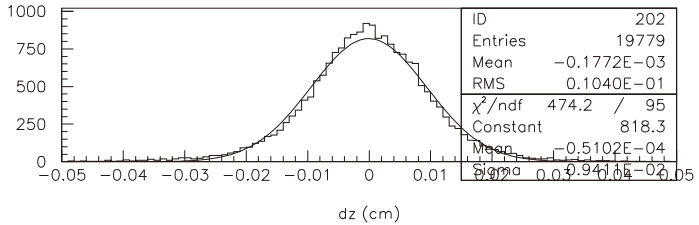
dr

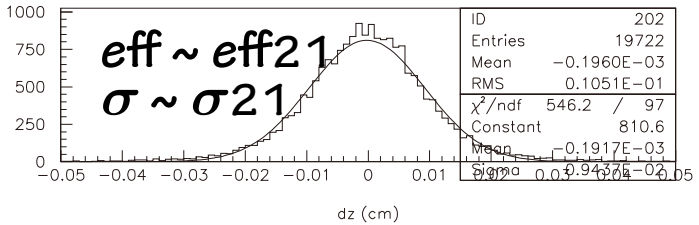


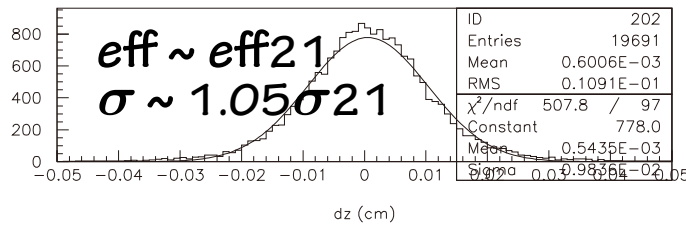




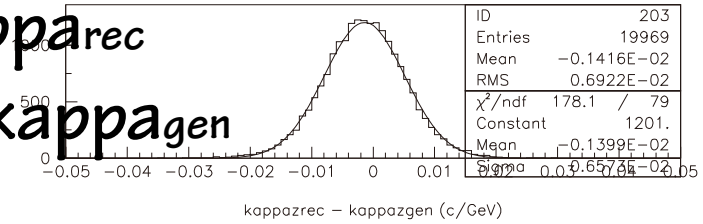
dz

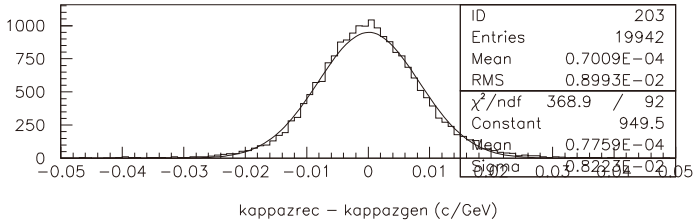


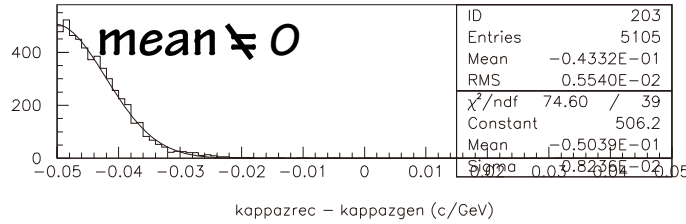




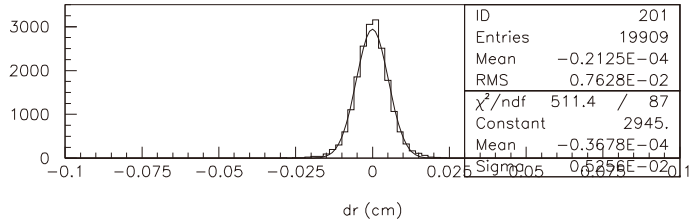
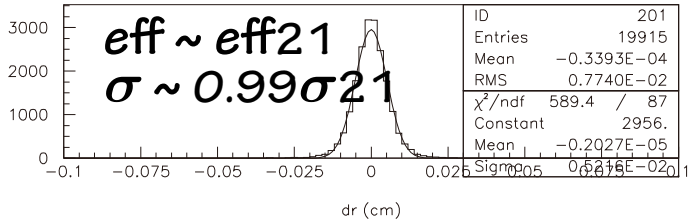
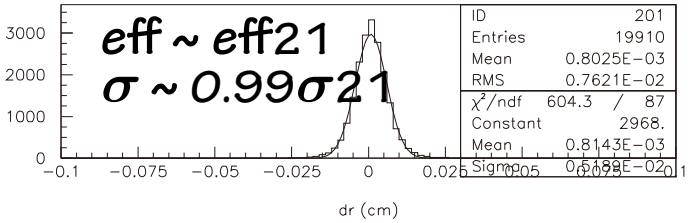
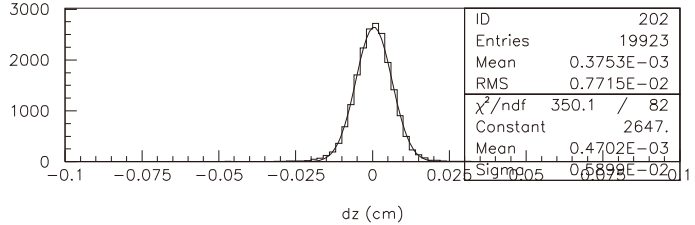
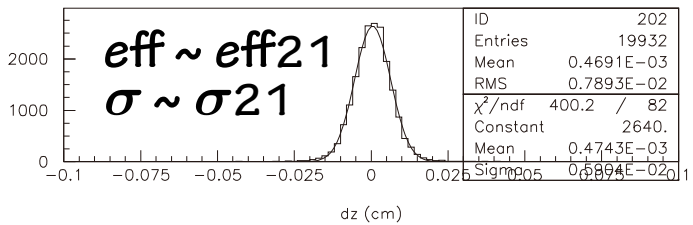
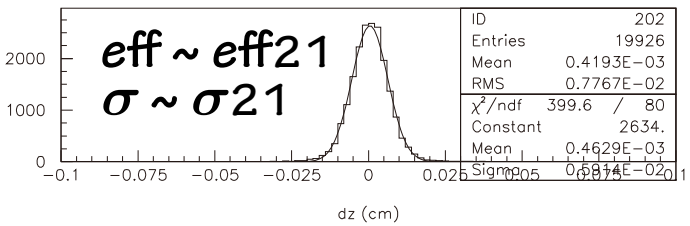
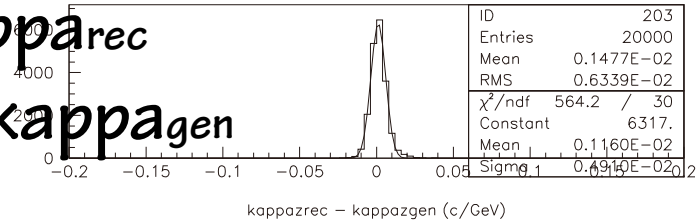
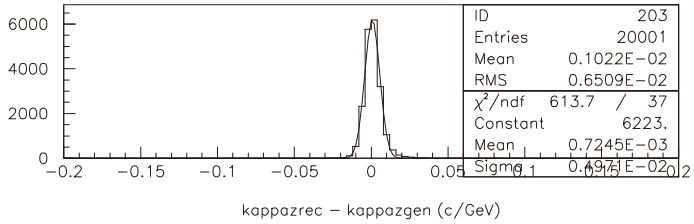
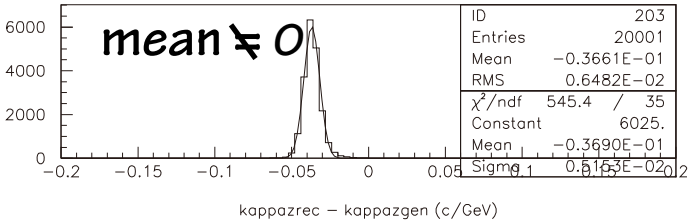
kappa_{rec} - kappa_{gen}







Single π+ p_t=0.5GeV/c, θ = 90deg, produced @ IP (0,0,0)

	Belle B-field	Belle II HC-opt.	mismatched B
	gsim : map# 21 trak : map# 21	gsim : map# 31 trak : map# 31	gsim : map# 31 trak : map# 21
dr			
dz			
$kappa_{rec}$ - $kappa_{gen}$			

Single π+ p_t=0.5GeV/c, θ = 20deg, produced @ IP (0,0,0)

Belle B-field

gsim : map# 21
trak : map# 21

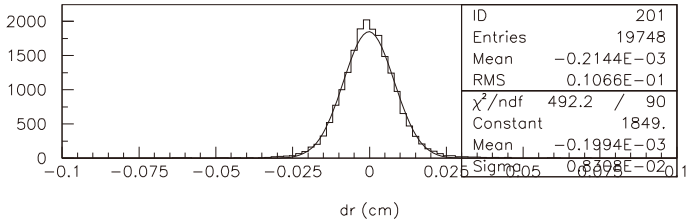
Belle II HC-opt.

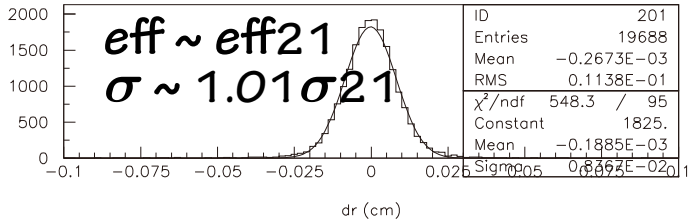
gsim : map# 31
trak : map# 31

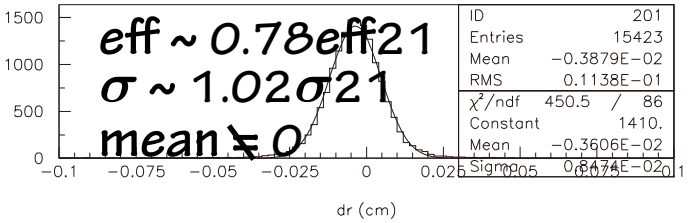
mismatched B

gsim : map# 31
trak : map# 21

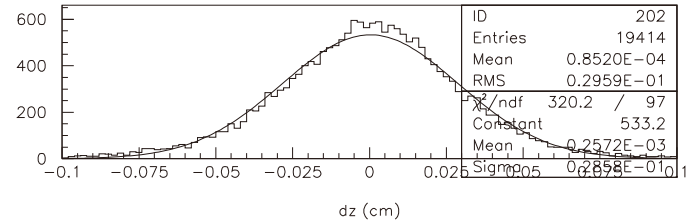
dr

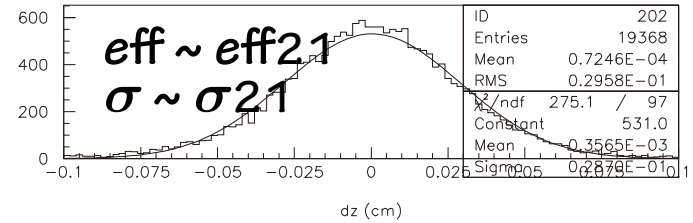


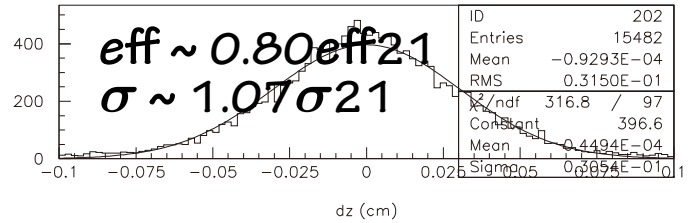




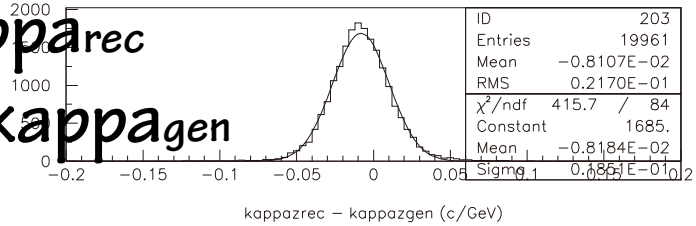
dz

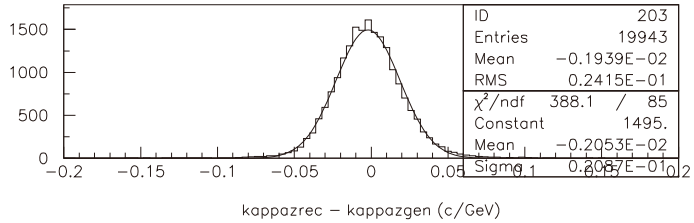


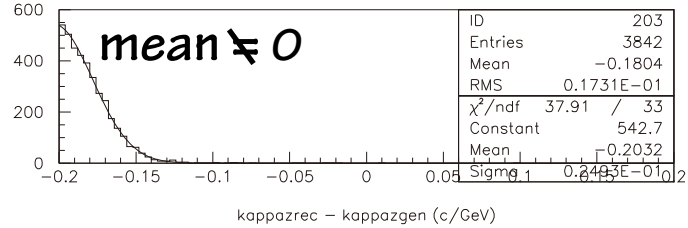




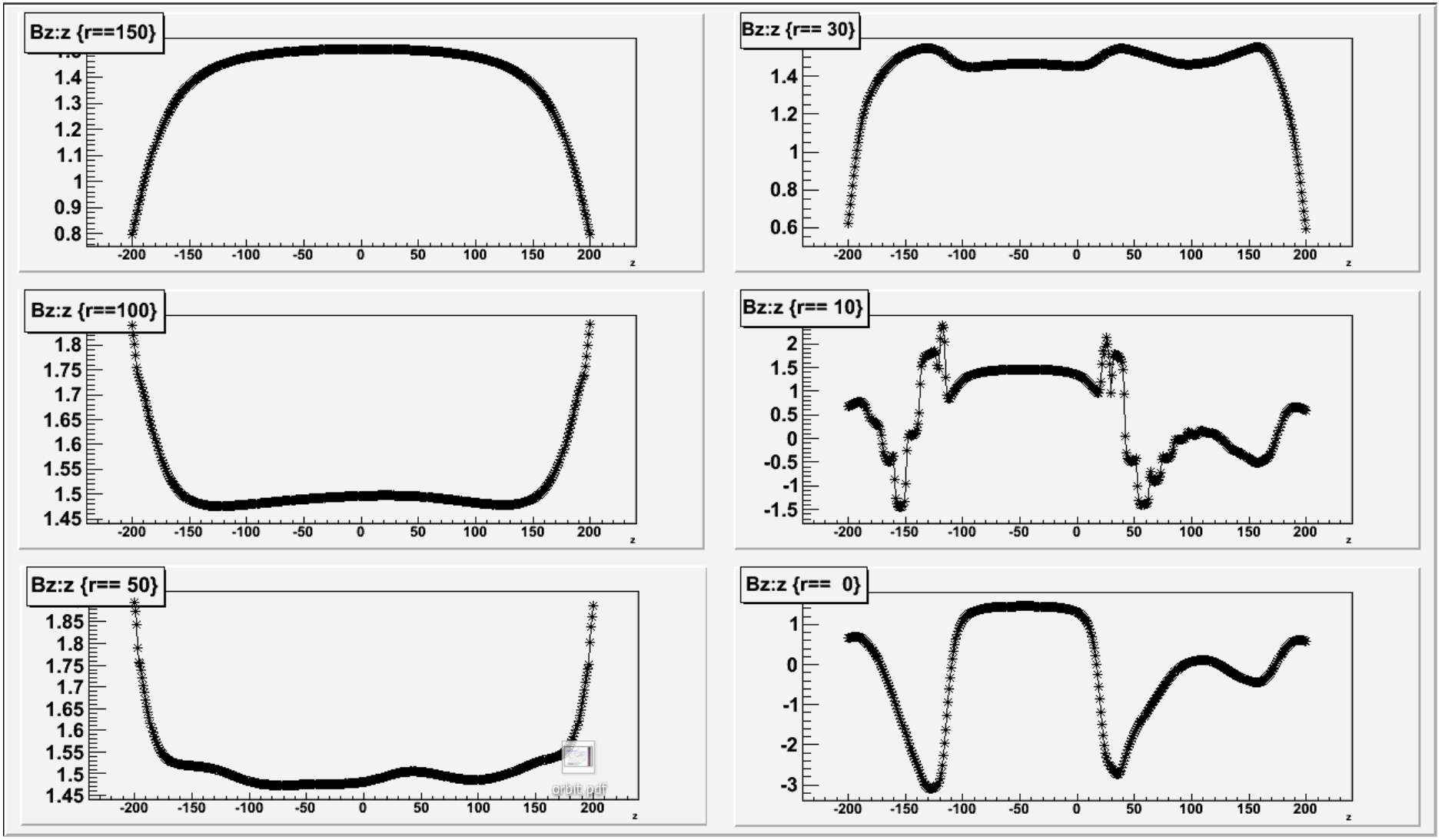
$kappa_{rec}$ - $kappa_{gen}$



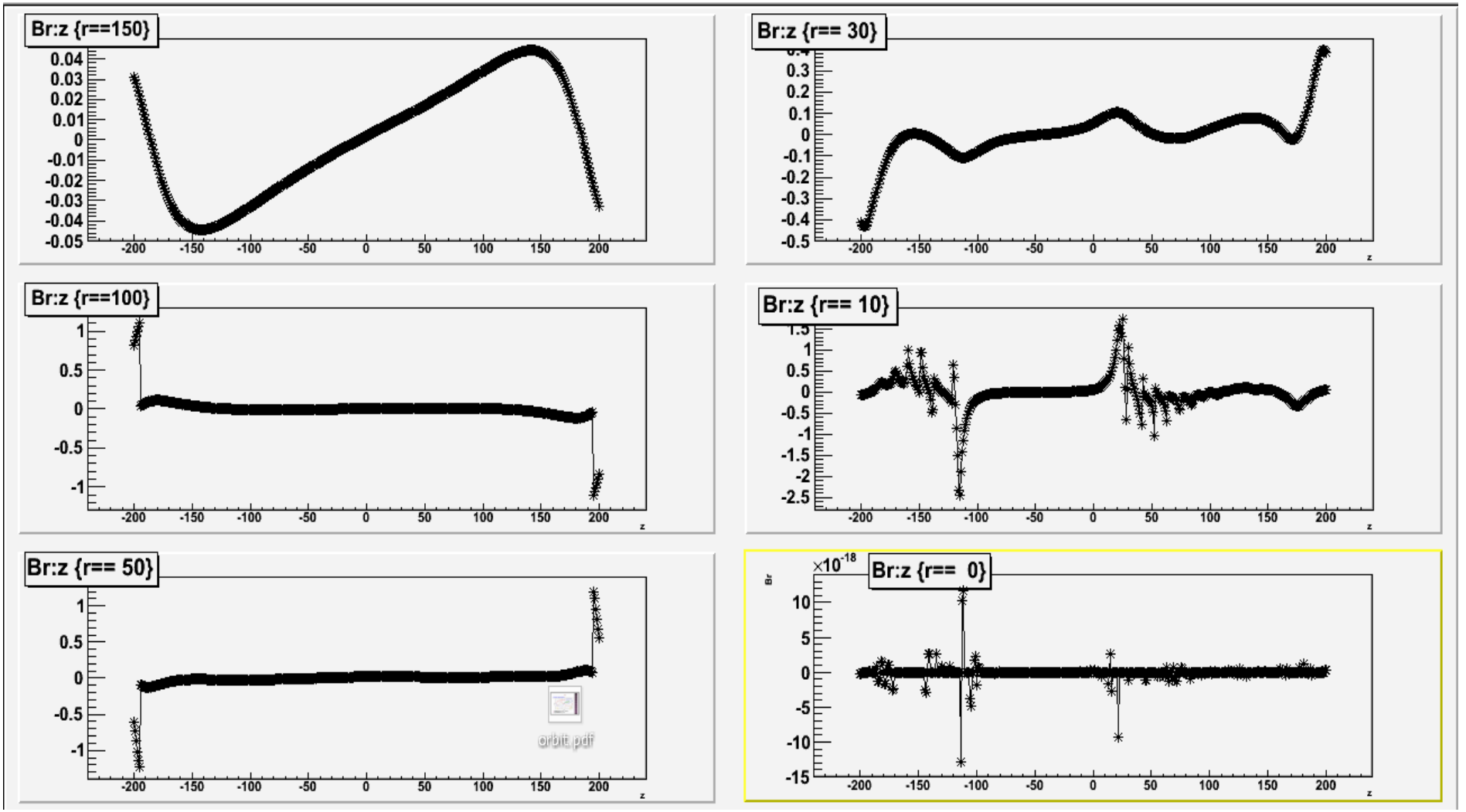




B_sol+ant_sol @ Belle-II



B_sol+ant_sol @ Belle-II



TGeoVolumeAssembly

> For the simplified geometry, I think I discussed this once with Christian Oswald: One could think of a switch in the code to determine the level of detail. For online tracking, speed is definitely an issue and having too many volumina would slow down things.

I think, the number of volumes is not the only problem. Navigation speed is according to ROOT documentation mainly influenced by the structure of the geometry: A tree structure is preferable to a flat structure. Composed shapes and TGeoVolumeAssemblies are to be avoided. My impression is that there are a lot of TGeoVolumeAssemblies....

1. Which is the current mother volume? => Belle II detector 2. Which daughter volumes are contained? => PXD(TGeoVolumeAssembly), SVD(TGeoVolumeAssembly), CDC(TGeoVolumeAssembly) HERE COMES THE PROBLEM! CDC is internally a box containing the PXD and SVD. The navigation code cannot decide in which of these TGeoVolumeAssemblies it is. In the worst case it has to check all daughter volumes of PXD, SVD, CDC. So much more volumes are to be checked than necessary! Solution: Make a tube which contains PXD, a tube which contains SVD, a tube which contains CDC. Then the navigation code would need to check only these three containers and would then continue with the correct subdetector e.g. PXD, skipping checking if the current point is contained in some volume of the other detectors.

Of course the whole thing becomes even worse because the SVD has 4 Layers which are 4 TGeoVolumeAssemblies...