ITS upgrade meeting

"Vertexig and tracking"

Entirely based on works and results by:
S. Rossegger, R. Shahoyan, A. Mastroserio, C. Terrevoli

Outline:

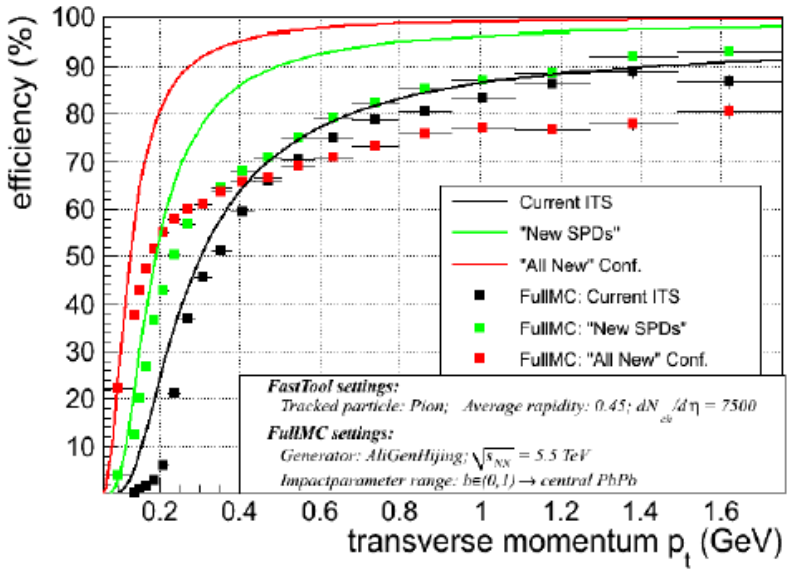
- Comparison Fast simulation vs. MC Transport code
- Fast "quasi MC" tool: FMCT
- New upgraded configuration

Comparison Fast simulation vs. MC Transport code

from last ITS upgrade plenary meeting

Tracking efficiency studies

12th of September, 2011



The scenarios:

- **New SPDs (exchange only Pixels)**
- **All New (build a completely new detector)**

The tools:

- **Fast (semi-analytical) Tool**
- **ITSupgrade code in AliRoot**

The problem:

- **The tools do not agree …**

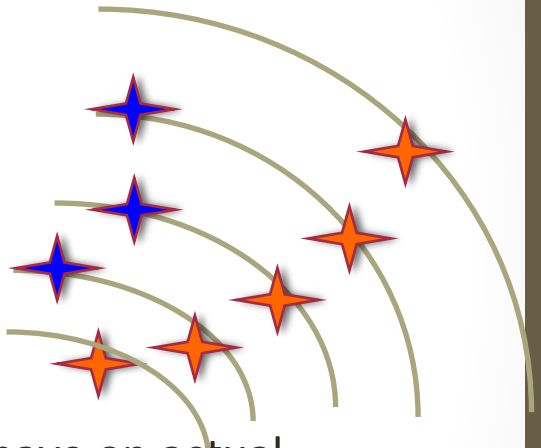
12/09/2011 Stefan Rossegger 1

a lot of works since then:

- moving to realistic multiplicities (dN_{ch}/dy from~ 5500 to ~2000 charged)
- a few "bugs"
- common definition of efficiency and fake ratio

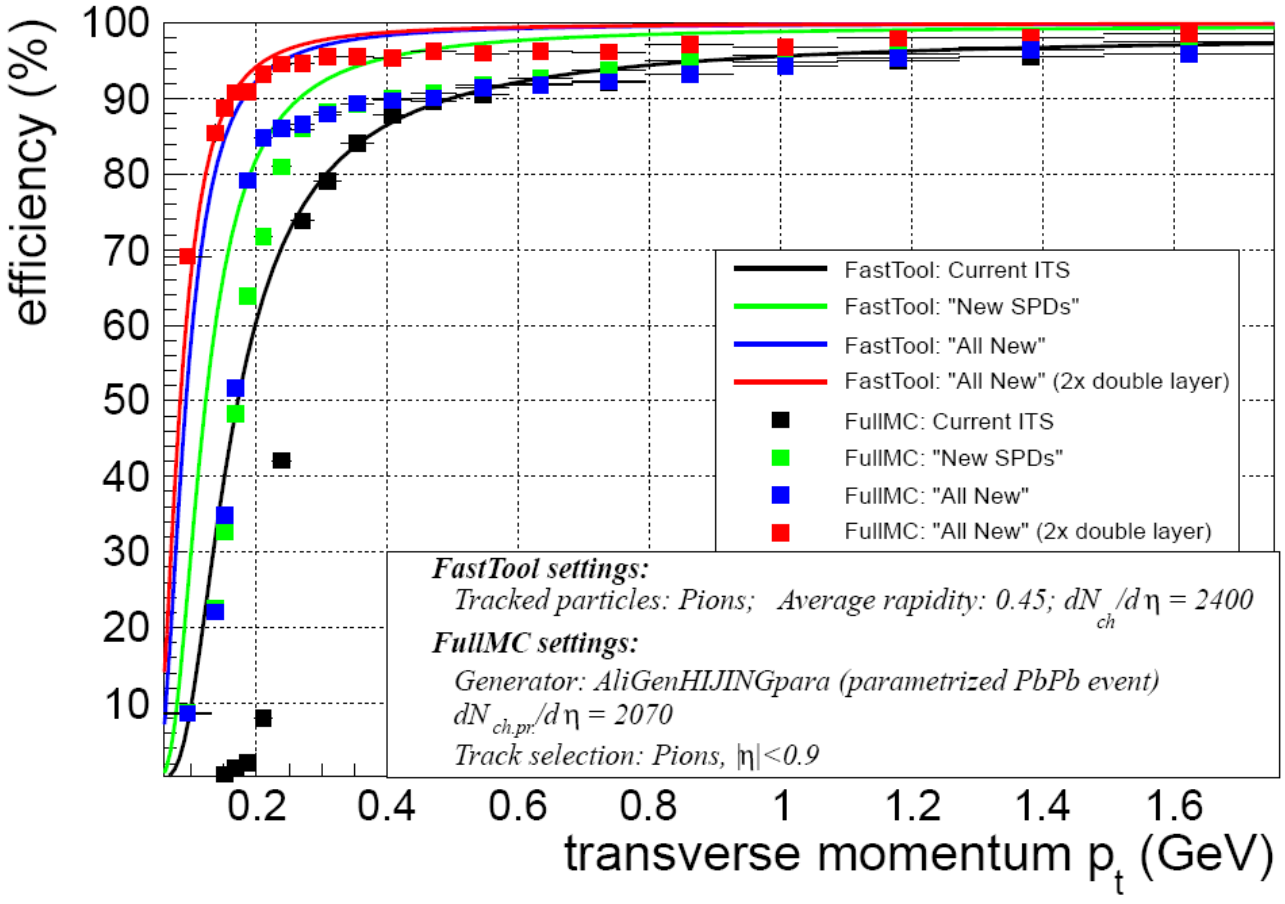
Discussion points for the efficiency estimates

- Multiplicity, event topology
  - PbPb (central Hijing) with b [0,3.5] fm and dN/dEta at mirapidity 2000

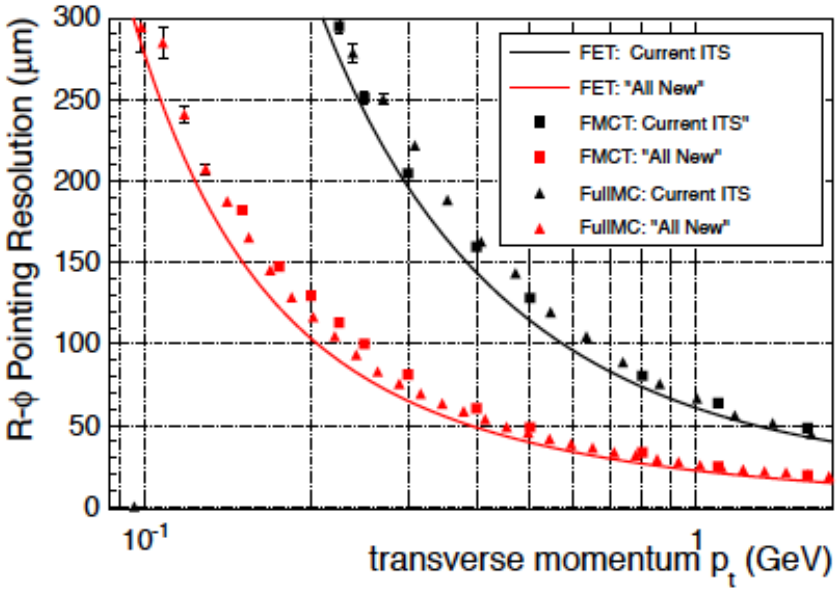


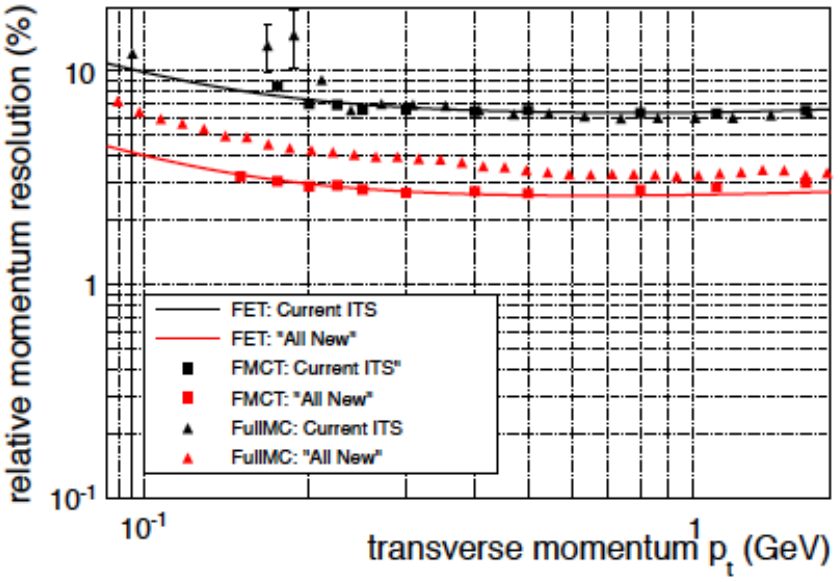
- Track/particle selection
  - MC trackable (3 TrackRefs) too loose
  - Tracks built in Inward direction
  - Tracks built with the vertex constraint
  - Further contraints should be considered in order to have an actual MC trackable
    - eta selection
    - Primary selection
- Revisited layout for an improved "seeding" for the inward tracking (close radii in outermost layers)

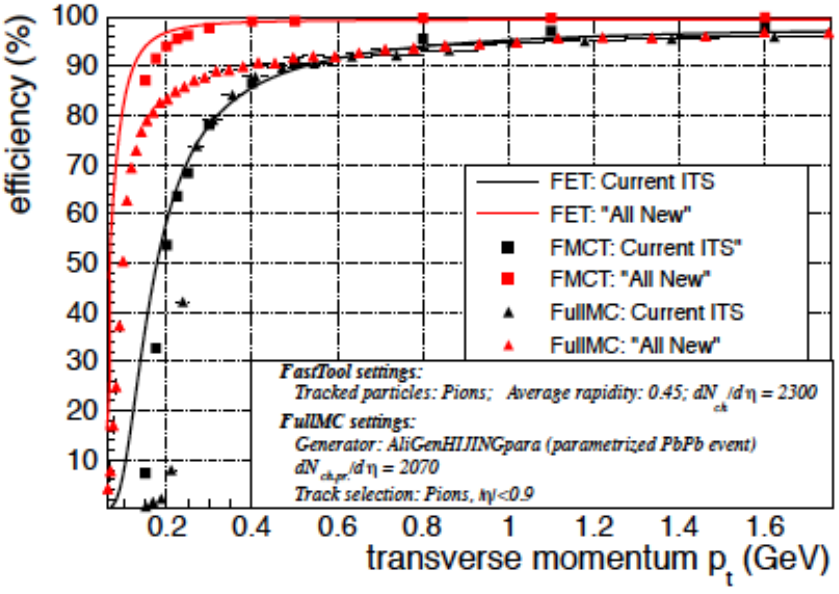
Latest results (Stefan)



"AllNew setting" : two couple of close layers at two outermost radii







From yesterday version of CDR Cap.3

Figure 3.14: Comparison of the "Fast tools" versus the "Slow simulation" for two configurations, namely the current ITS layout and the upgrade scenario "All-New" (see text for details); $r\phi$ resolution, p_t resolution, tracking efficiency.

Conclusion on Fast tool vs. MC

- after some works a better qualitative agreement between the estimate of tracking efficiency
  - quantitative agreement on pointing resolution since long
- Estimate of the tracking efficiency from the Fast tool corroborated by the new development of Ruben (FMCT, next item)
- the remaining differences are attributed to the fact that the tracking algorithm of the MC is not yet optimized

Fast “quasi MC” tool: FMCT

Addition of quasi-MC to analytical tool:

SolveSingleTrackViaKalman(mass,pt,theta) :

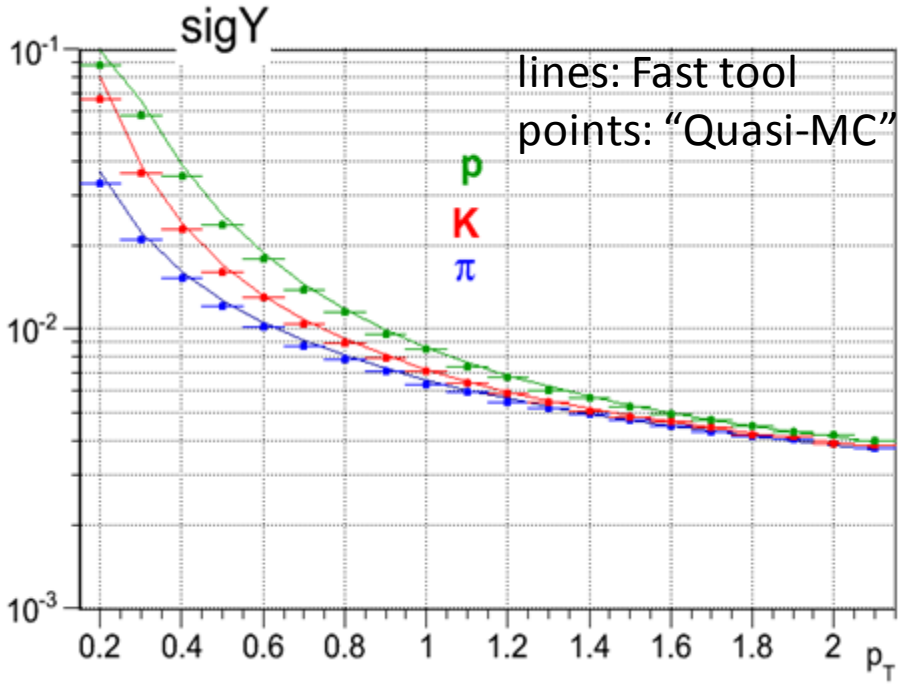
- ✓ prepares "ideal track" (no MS) in the whole detector.
- ✓ reconstructs using ideal clusters and produces analytical estimate of resolutions at each layer (**as before**)

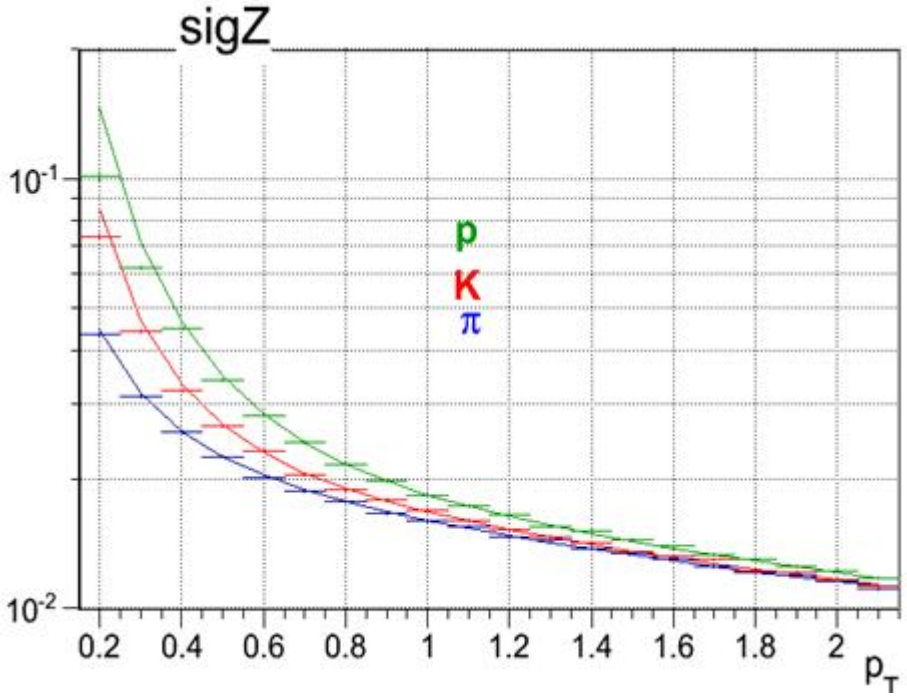
SolveSingleTrackViaKalmanMC(int offset) :

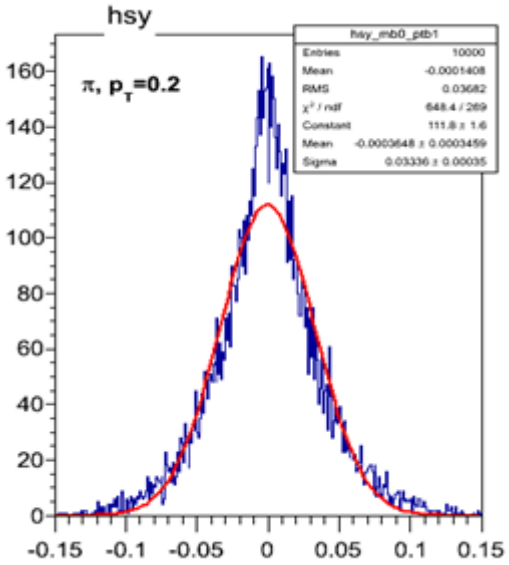
- ✓ Does almost real MC (propagation with random MS+deteministic energy loss) of the original track used in analytical estimate, up to the the last ITS layer (+offset if requested for TPC/ITS tracking):
- ✓ Add random "fake" clusters according to occupancy at each layer and within a window accessible for any fake track which may "steel" the correct cluster
- ✓ Seeding with "ideal track" propagated to last layer, do the full reconstruction allowing competition between "correct MC" and "fake" clusters.
- ✓ Range all tracks propagated to vertex according to χ^2 /NDF (penalized for missed layers), suppress those tracks which don't use any "correct" cluster, and select the winner.

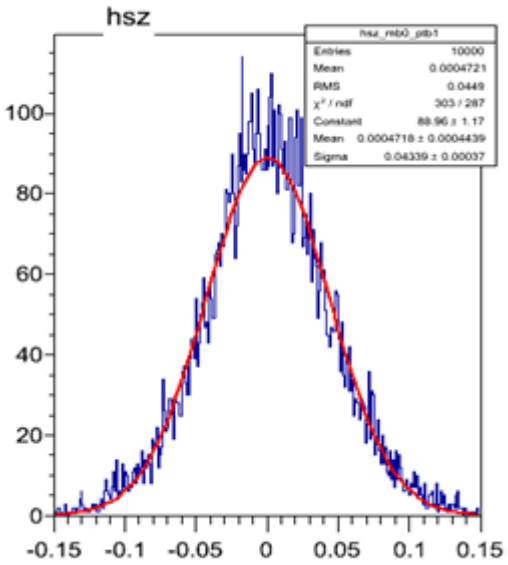
- ✓ The deviation of "real parameters" from "ideal" ones is fitted to produce resolutions estimates.

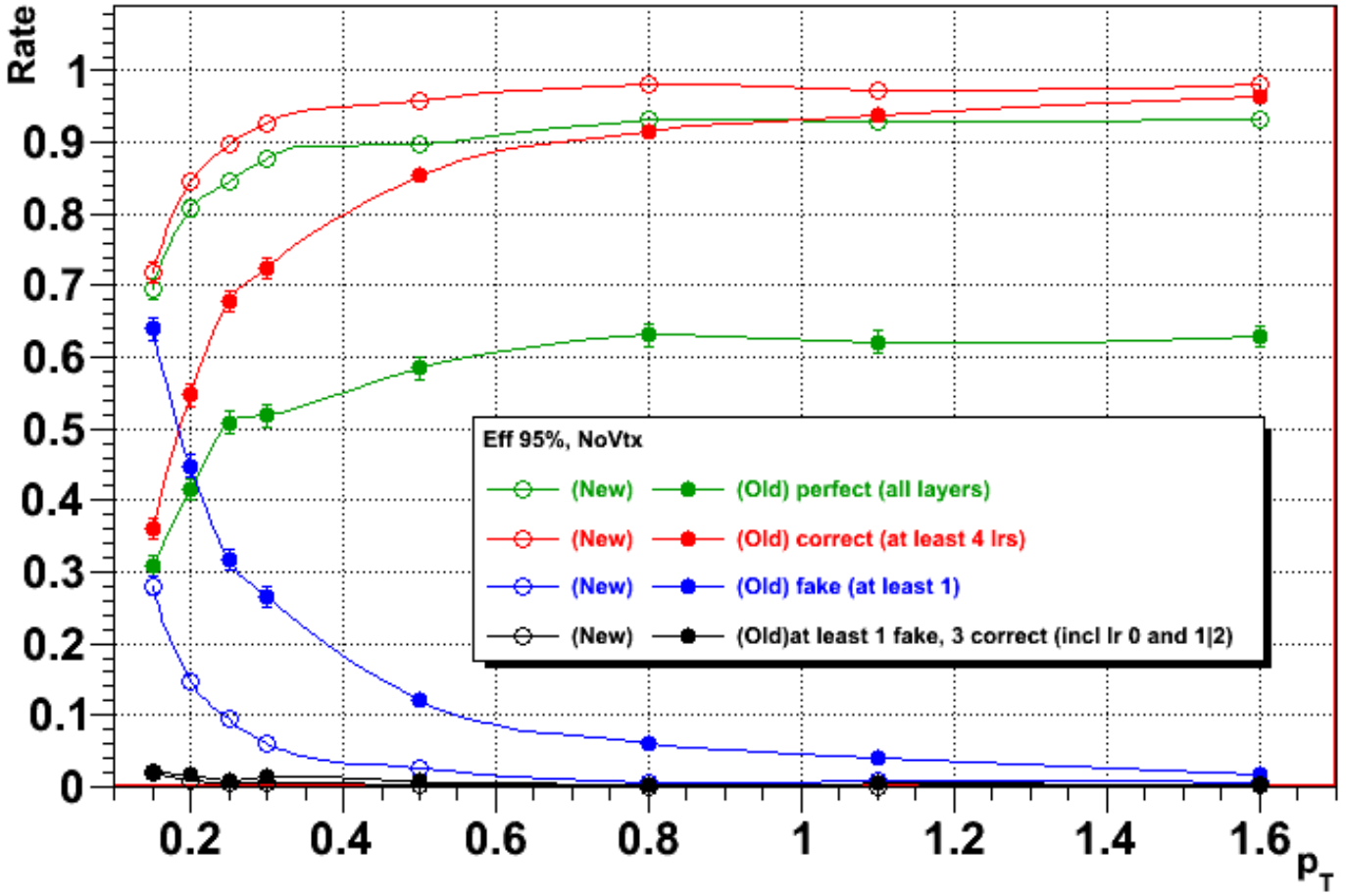
Log term in MS is ignored





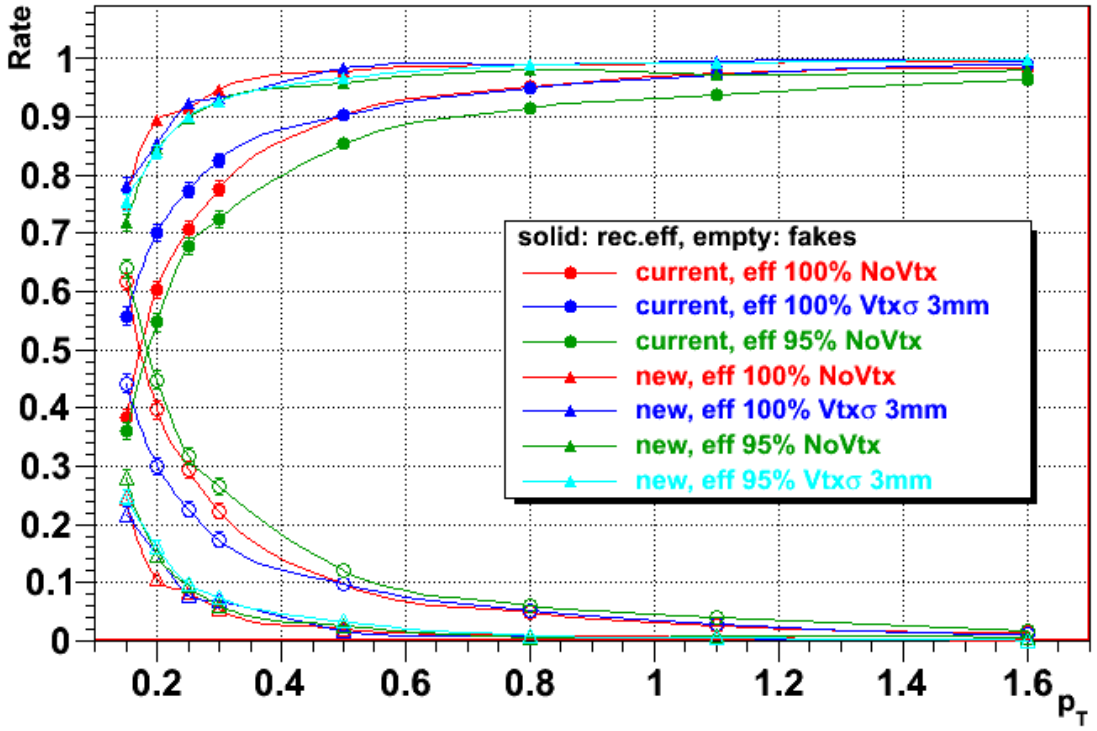






"new" is for standard "AllNew" in Stefans' definition, 7 layers with 4μm resolution.

Different definitions/selections of correct tracks are possible:
from “prefect” (correct hits on all layers)
up to “correct hits on some key layers”



"NoVtx" vs "Vtxsigma 3mm":
Current full reco SA algorithm uses the vertex point for track finding.
Not used explicitly in the fit, but this helps to reject the fakes candidates (even those with good chi2 w/o vertex constraint).
$\Rightarrow$ possible loss of correct off-vertex tracks, increasing with p_T.
$\Rightarrow$the eff. obtained with full SA reco, is (in principle) valid only for prompt tracks...

Only to make FastMC results comparable with full MC reco, optionally I add track-to-vertex χ^2 to total track χ^2. assigning some lose vertex error (3mm >> σ DCA, just for test, no relation to effective constraint of full MC reco with SA algorithm).

New upgraded configuration

New upgraded configuration

- so far we have focused in the CDR on two configurations:
  - "new SPD" "all new ITS"

Table 3.3: Characteristics of the Upgrade Scenario 1 - "New-SPDs"

Layer / Type	r [cm]	$\pm z$ [cm]	Nominal resolution $r\phi \times z$ [μm^2]	Material budget X/X_0 [%]
Beam pipe	2.0	-	-	0.22
1 / new pixel	2.2	10.5	4×4 (to) 6×6	0.30 (to) 0.50
2 / new pixel	4.7	13.5	4×4 (to) 6×6	0.30 (to) 0.50
3 / new pixel	9.0	18.5	4×4 (to) 6×6	0.30 (to) 0.50
Th. shield	11.5	-	-	0.65
4 / drift	15.0	22.2	35×25	1.13
5 / drift	23.9	29.7	35×25	1.26
Th. shield	31.0	-	-	0.65
6 / strip	38.0	43.1	20×830	0.83
7 / strip	43.0	48.9	20×830	0.83

Table 3.4: Characteristics of the Upgrade Scenario 2 - "All-New"

Layer / Type	r [cm]	$\pm z$ [cm]	Nominal resolution $r\phi \times z$ [μm^2]	Material budget X/X_0 [%]
Beam pipe	2.0	-	-	0.22
1 / new pixel	2.2	10.5	4×4 (to) 6×6	0.30 (to) 0.50
2 / new pixel	2.8	11.2	4×4 (to) 6×6	0.30 (to) 0.50
3 / new pixel	3.6	12.3	4×4 (to) 6×6	0.30 (to) 0.50
4 / new pixel / strip	20.0	30.7	4×4 (to) 50×50	0.30 (to) 0.50
5 / new pixel / strip	22.0	33.0	4×4 (to) 50×50	0.30 (to) 0.50
6 / new pixel / strip	41.0	54.5	4×4 (to) 50×50	0.30 (to) 0.50
7 / new pixel / strip	43.0	56.8	4×4 (to) 50×50	0.30 (to) 0.50

- choice driven by pointing resolution
- now the estimate of ITS stand-alone tracking efficiency under control → optimization also based on track. eff.

Comparisons to "our old" layouts

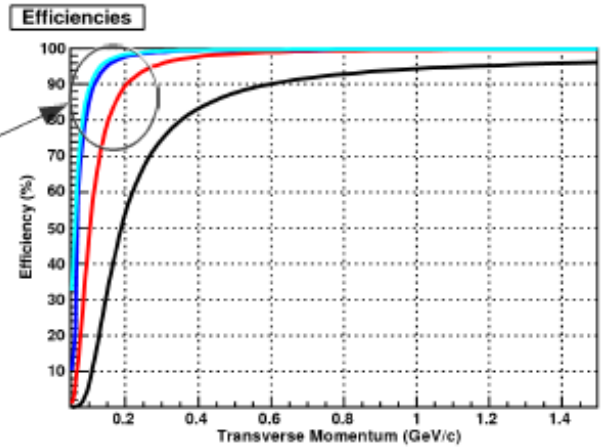
Black – Current ITS

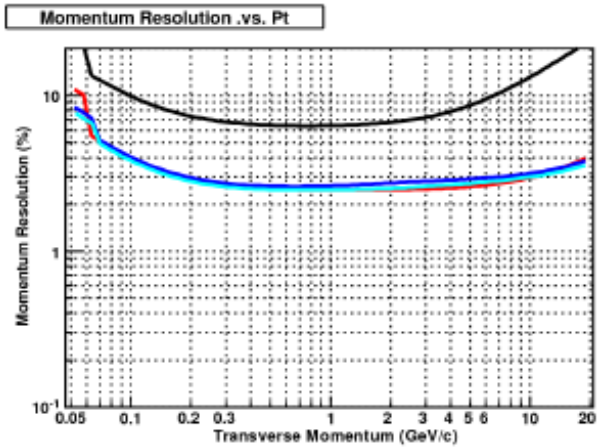
RED – *AllNew ITS (7 layers, from CDR) → (2.2, 3.6, 6.8, 12.4, 23.5, 39.6, 43.0) cm

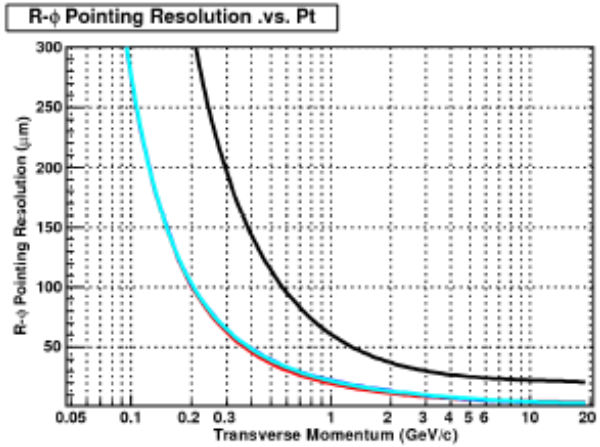
BLUE – *Newer AllNew (7 layers) → (2.2, 2.8, 3.6, 20.0, 22.0, 41.0, 43.0) cm

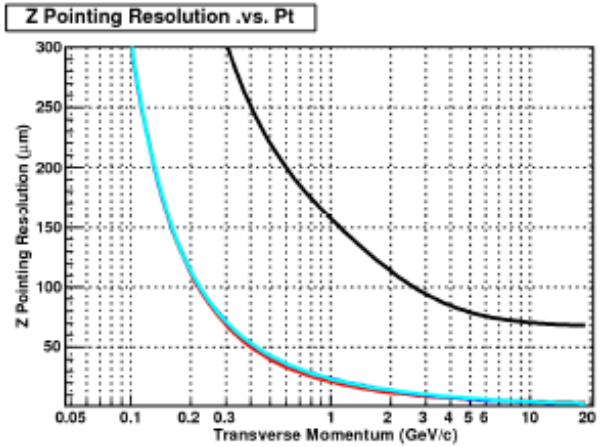
CYAN – *Newest Newer AllNew (9 layers) → (2.2, 2.8, 3.6, 4.2, 20.0, 22.0, 33.0, 43.0, 43.6) cm

ENHANCEMENT !







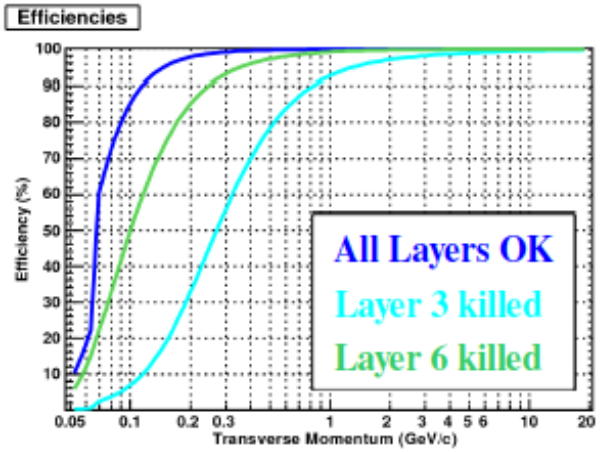


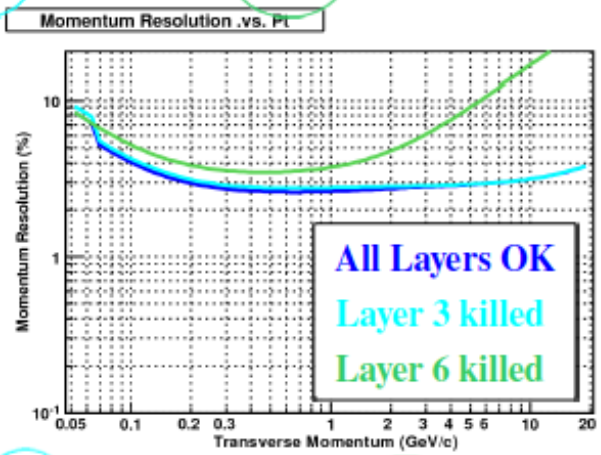
** we definitefily have to find better names*

Redundancy considerations?

- The fight with focus on redundancy, the 9 new layers version would clearly win.
- Is it also feasible in terms of COSTS?

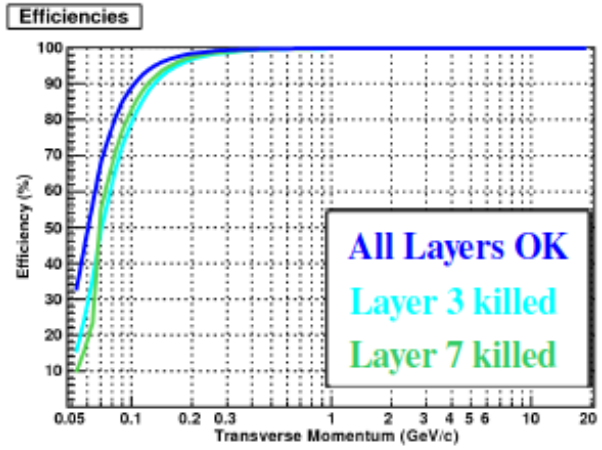
***Newer AllNew (7 layers) → (2.2, 2.8, 3.6, 20.0, 22.0, 41.0, 43.0) cm**

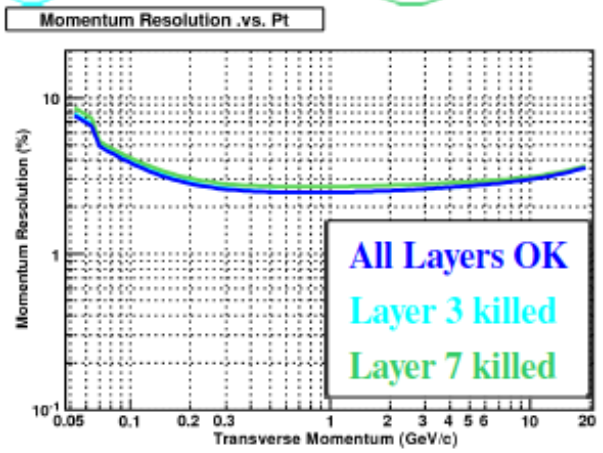




***Newest Newer AllNew (9 layers) → (2.2, 2.8, 3.6, 4.2, 20.0, 22.0, 33.0, 43.0, 43.6) cm**

MUCH BETTER





Conclusion

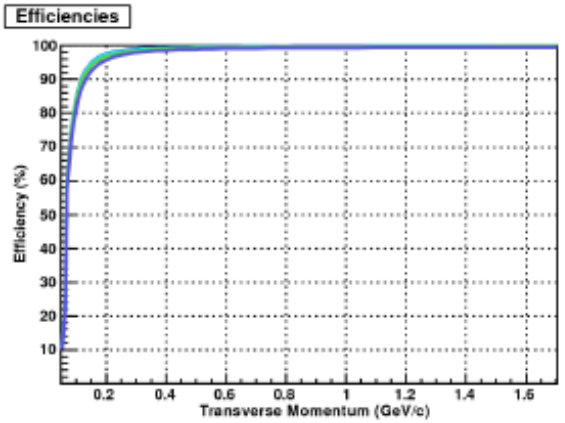
- We propose to add considerations and improved efficiency plots for either the "Newer-AllNew" or the "Newest-Newer-AllNew" layout …
- No time to consider "major" changes also in the other chapters?
- Pt and d0 resolution are the same (therefore no change in the "physics motivation"), but the efficiency would increase !

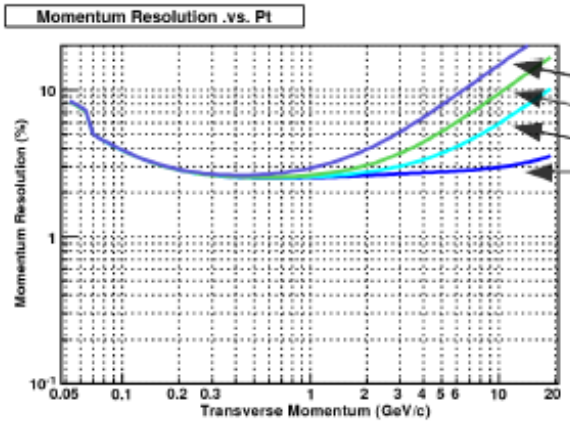
Extras

Eventually courser resolution?

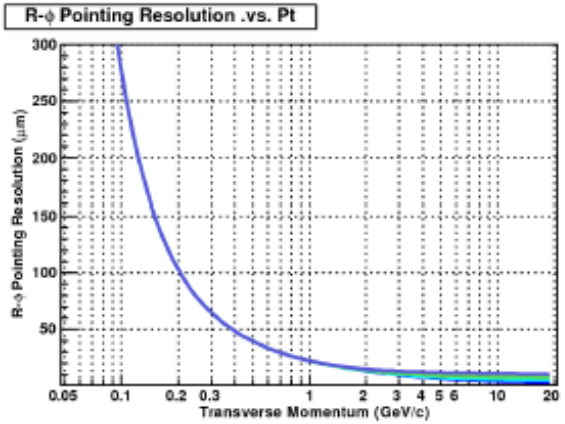
- Layers further out (r>20cm) could have courser resolution than 4 to 6 μm?
- Tracking efficiency is not much affected for up to 40 microns
- But, the momentum resolution gets worse ..

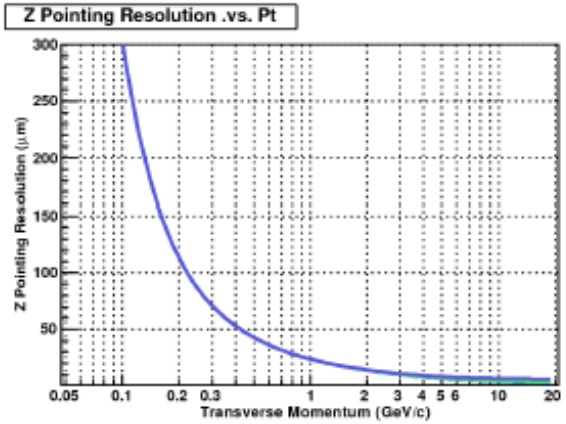
BLUE – *Newer AllNew (7 layers) → **(2.2, 2.8, 3.6, 20.0, 22.0, 41.0, 43.0) cm**





Res = {4, 20, 40, 100} μm



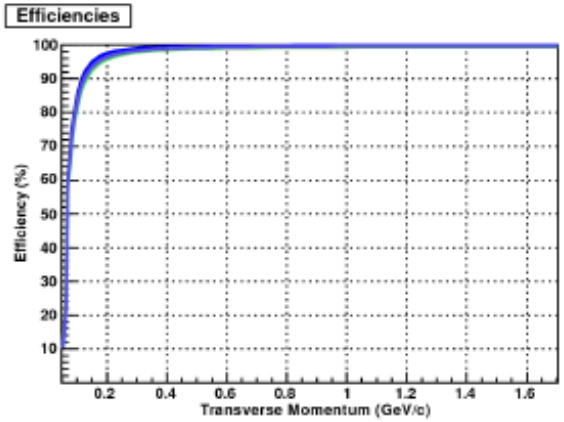


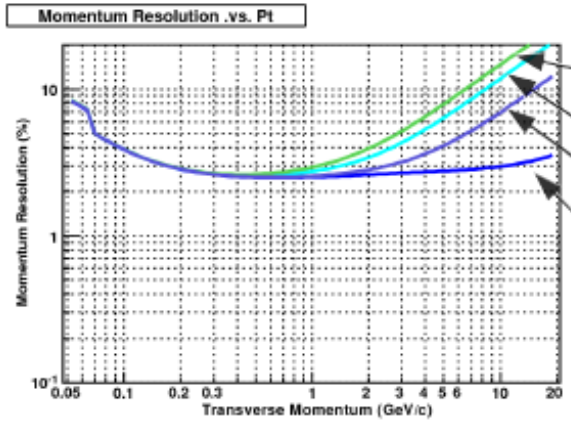
Question related to trigger capabilities:
Up to which Extend do we need the standalone pt resolution?

Eventually courser resolution?

- Pt resolution is effected by changes on all the layers

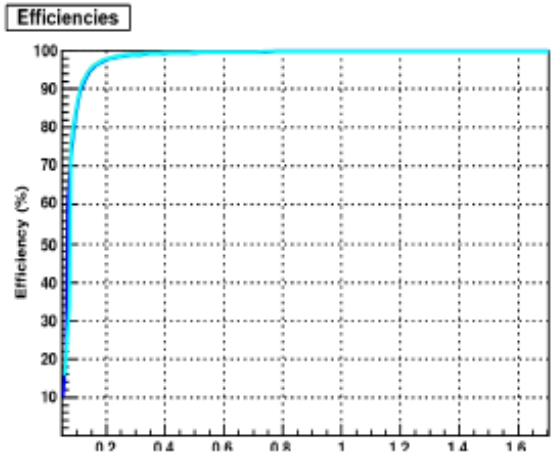
BLUE – *Newer AllNew (7 layers) → **(2.2, 2.8, 3.6, 20.0, 22.0, 41.0, 43.0) cm**

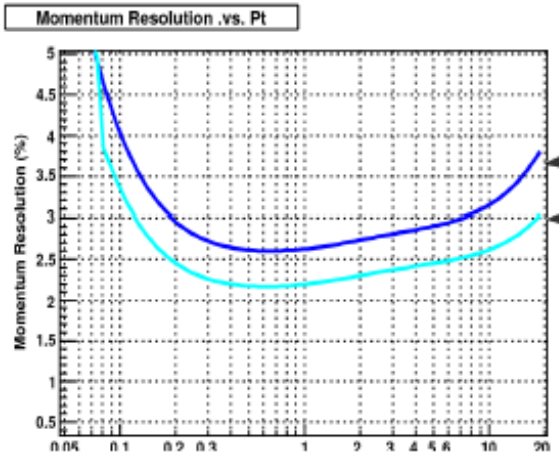




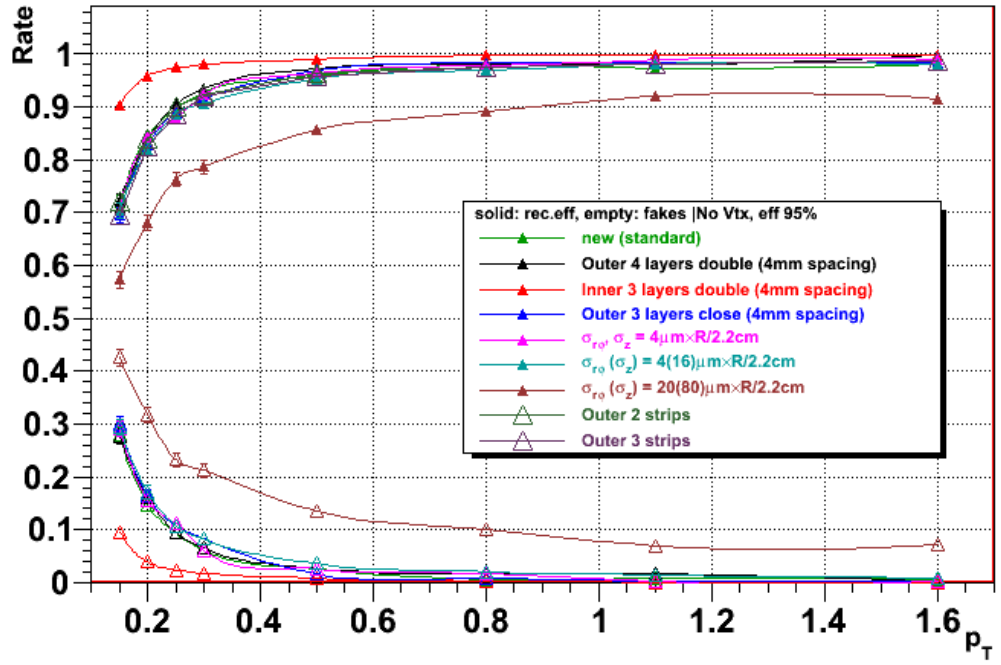
Case 4: {100,100,100, 20} μm
Case 3: {100,100, 20, 20} μm
Case 2: {100, 20, 20, 20} μm
Case 1: { 4, 4, 4, 4} μm

- Last layer(s) at ~ 50 cm would help a bit (but improv. is prop. to level arm)



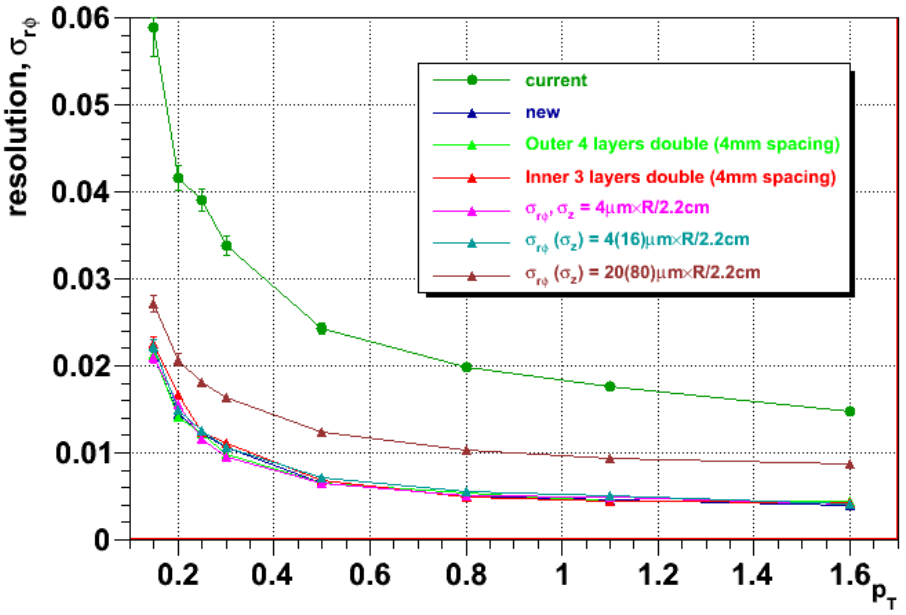


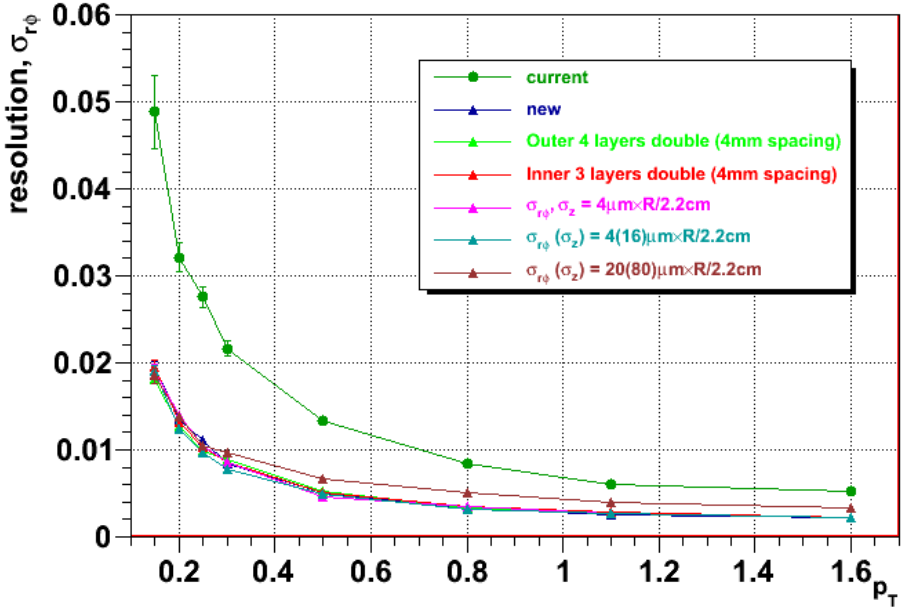
Last four layers further out :
R=(20, 22, 41, 43) cm
R=(25, 27, 48, 50) cm

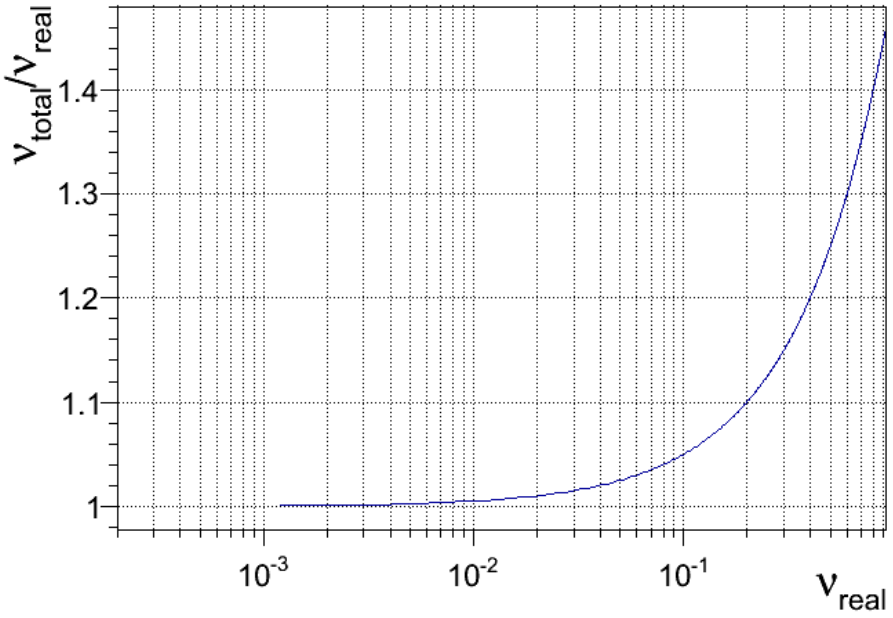


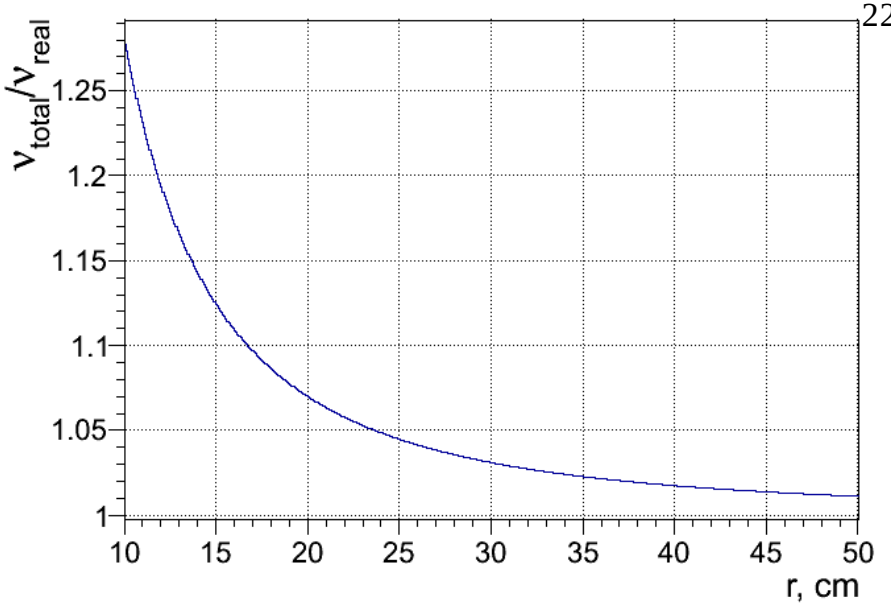
Test with different layouts

4 μm R/2.2: i-th layer resolutions are scaled as layer_i_r/layer_0.



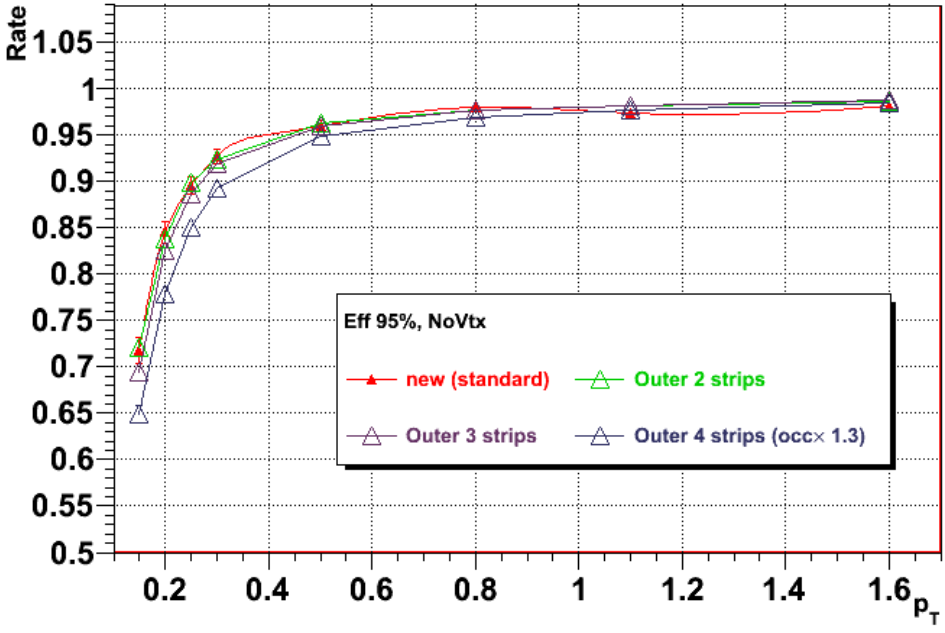






Test with strips (L:2cm × d:80μm, 35 mrad angle φ)

Fake strip crossings: n real hits in rectangle $L^2 tg(\varphi)$ will create n(n+1)/2 crossings.
ν: <number> of of such crossings in rectangle.



Number of Kalman updates per track
(brute force reconstruction with very rough optimization)