Status of Runge Kutta tracking in HYDRA

by Ilse Koenig,
GSI, Feb 2nd, 2006

Table of Contents

1. Overview

The aim of this report is to give a status of the Runge Kutta tracking code and to provide a set of pictures to show the capabilities and limitations of the current implementation. These pictures may serve as a reference for future developments.

All pictures in this report were done with **HYDRA code checked out from CVS at Feb 2nd, 2006.** Additional private classes actually not in CVS were used to store variables in the tree, which are otherwise not stored in the output.

Input were AUG04 simulations with 50000 PLUTO events, containing only single leptons.

The parameters were taken from Oracle for simulation reference run "aug04sim_mediumfield_fulltarg_align_gen1" (runid 7003) at history date Feb 2nd, 2006.

2. Runge Kutta tracking

The first versions of the Runge Kutta tracking were written by Alexander Ivashkin and integrated in HYDRA by Alexander Sadovsky in October 2004.

A formal description of the Runge Kutta formalism can be found in a the talk given by Alexander Ivashkin at the HADES collaboration meeting 2004:

http://www.gsi.de/documents/DOC-2004-Oct-56-1.pdf

and the method itself in **Numerical recipes in C**

http://www.library.cornell.edu/nr/bookcpdf.html

In June 2005 the code was refactored and the first version committed in CVS at July 6th. ,2005.

A detailed report by Alexander Sadovsky of the integration in HYDRA and the older versions can be found in the HADES-Wiki:

http://hades-wiki.gsi.de/cgi-bin/view/SimAna/RungeKuttaTracking

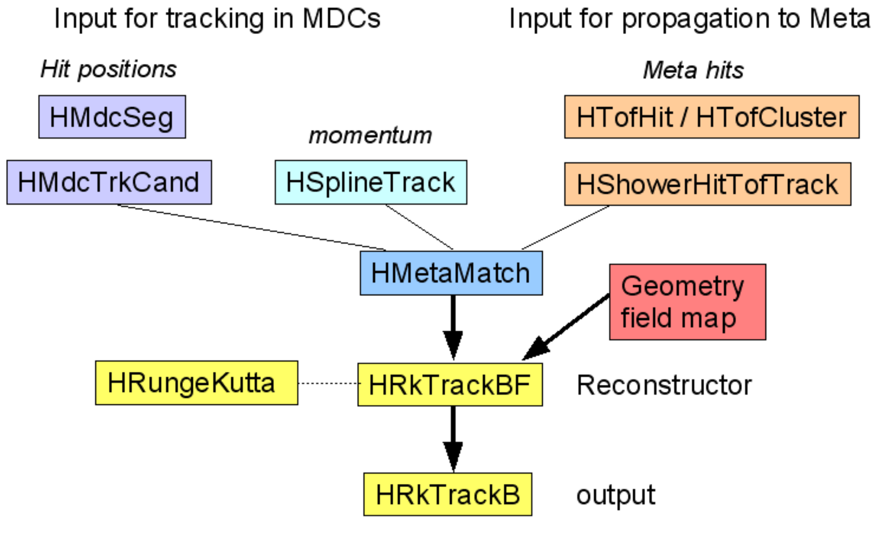
The main reason for the refactoring was the loss in efficiency due to the fact, that the older versions required fitted hits in MDC 1, MDC 2 and MDC3. In case of missing Motherboards the number of contributing layers drops below the required minimum value and the hit index is not stored in the MDC segment, although the segment itself may be rather well defined. In this case Runge Kutta tracking was skipped in the old code.

The second aim for the refactoring was to increase the performance by code restructuring and by implementation of Runge Kutta with adaptive stepsize.

2.1. Classes and program flow

The Runge Kutta tracking code consists of three classes

1. the public reconstructor **HRkTrackBF**
2. the private class **HRungeKutta**, which contains the code to propagate the particle through the magnetic field
3. the output category **HRkTrackB**, derived from **HBaseTrack**



Initialization (init() function):

The reconstructor **HRkTrackBF**

- gets pointers to all input categories
  - MetaMatch (creates an iterator on the category)
  - MdcTrkCand, MdcSeg, MdcHit (input for tracking in MDC)
  - SplineTrack or eventually KickTrack123B (input for initial momentum and polarity)
  - ShowerHitTof, respectively ShowerHitTofTrack for simulation, TofHit and TofCluster (input for track propagation to META)
- gets pointers to the parameter containers

  - MdcTrackGFieldPar and MagnetPar for the magnetic field
  - all geometry parameter containers: SpecGeomPar, MdcGeomPar, TofGeomPar, TofinoGeomPar, ShowerGeometry
- creates the output category HrkTrackB.

Actually the parameters used in the Runge Kutta tracking are hardcoded.

Reinitialization (reinit function):

- creates the HRungeKutta class object and sets the pointer to the field map, the field scaling factor and the MDC geometry transformations (in the sector coordinate system)
- calculates and stores the norm vectors on the TOF, Tofino and Shower modules.

Event loop (execute function):

For each event, the reconstructor loops over all MetaMatch objects.

For each entry, three tasks may be performed:

1. track fitting in the MDC system
2. propagation to the target
3. propagation to Meta

The results are stored in the output category HRkTrackB.

Track fitting in the MDC system:

RK tracking needs an inner and outer MDC segment and a momentum as start value.

It is skipped, if there is no outer segment. From the segments, the hit positions in the MDCs are calculated. The hit indexes of the first three chambers are not used and might be -1. If there is no hit index for MDC4 in the outer segment, only the calculated hit positions of the first three chambers are used in the tracking.

Typically the momentum and polarity from Spline are used as start values. Although possible, it is not recommended to used the kickplane values, because it takes more iterations to converge and is less robust due to the larger momentum resolution.

Tests showed, that Runge Kutta often calculates a too low momentum in the first iteration and increases the momentum in the next iterations again to the final value. This undershoot causes problems at very low momenta (<= 50MeV/c), because the tracks may curl and Runge Kutta does not converge. To increase the efficiency, the starting value for the momentum is set to 50 MeV/c, if Spline delivers a lower momentum. But this needs further investigation.

Starting with the hit position and direction at MDC1 and the momentum, the track is propagated through the magnetic field, the intersection points with the MDC planes are determined and χ^2 is calculated.

$$\chi^2 = \Sigma((m_i - f_i)/\sigma_i)$$

$m_i :=$ *measured hit position at MDC i*

$f_i :=$ *fitted hit position*

$\sigma_i :=$ *error of measurement*

To minimize χ^2, it needs six track calculations in each iteration: five times with changed start values for the x- and y-positions at MDC1, the x- and y-direction at MDC1 and the momentum (calculation of derivatives) and at last the track with the new values after solving the system of linear equations. χ^2 is then calculated and compared with the value for the last iteration. The minimization stops. If the new χ^2 agrees within 0.5% to the previous value or at least after 11 iterations.

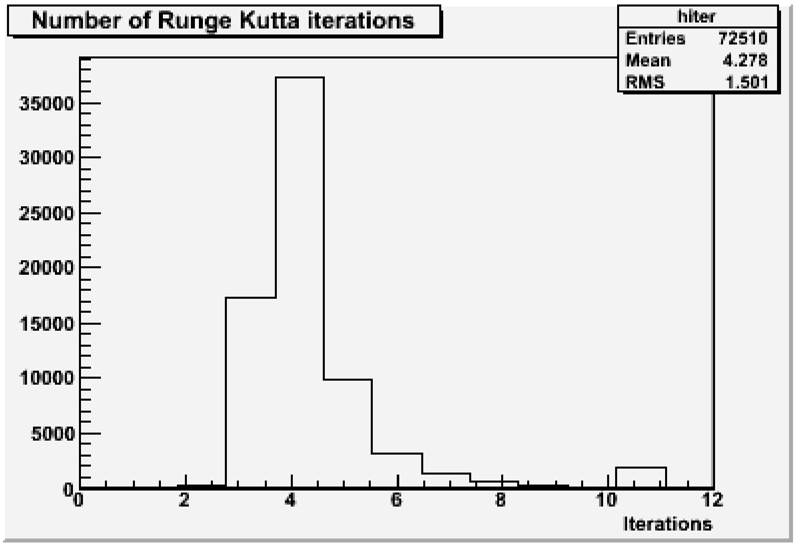


Figure 1: Number of Runge Kutta iterations per track

Figure 1 shows the number of iterations for a AUG04 simulation with PLUTO events (single leptons).

If the fit does not converges, χ^2 is set to 1000000 and the momentum to the initial Spline momentum.

Actually, the errors of the hit measurement are fixed values and correspond to the intrinsic resolution of the MDC chambers:

x-resolution: 280 micron, y-resolution: 140 micron

The errors from MdcSeg/MdcHit covariance matrix are **not** taken into account, because the behavior is not fully understood and needs further investigation. Therefore χ^2 cannot be normalized properly.

Propagation to the target:

If the fit converges the track is propagated from MDC1 back to the target. First, the track is propagated with Runge Kutta to a plane parallel to MDC 1 at a distance of 300 mm. With the position and direction at this point the new fitted RK inner segment is calculated. At this position the field should be small enough to calculated from there the intersection point with the middle plane of the target as a straight line.

The decision not to calculate the vertex (distance of closest approach to the z_axis) was taken, because a small error in the track direction results in a large error and therefore a wide spread of the vertex point in z-direction.

Propagation to Meta:

If a Meta hit exists, the track is propagated from the last MDC to the Meta with Runge Kutta. The difference in the fitted hit positions to the original hit positions is stored in the output, as well as the overall pathlength (distance from mid-target to TOF/Tofino).

The position and direction at the Meta is used to calculate the new fitted RK outer segment.

If there is no Meta hit, the pathlength is -1. But to calculate anyhow the outer segment in a field-free region, the track is propagated to a (infinitely large) shower plane.

To limit the number of unnecessary steps, propagation to Meta ends, when the track leaves the sector. In this case the track length is 0.

The difference between the original Meta hit position and the intersection point of the Runge Kutta track on the Meta is also stored in the output and could be used to tighten the matching windows.

2.2. Adaptive stepsize

The first implementation of Runge Kutta tracking used a fixed stepsize of 10 mm. Near the planes the stepsize was 5 mm, respectively 2.5 mm to get a more precise intersection point. This resulted in a large number of unnecessary small steps in the low field region or for higher momenta with small track curvature. Near the coils or for low momenta, the stepsize of 10 mm is already to large.

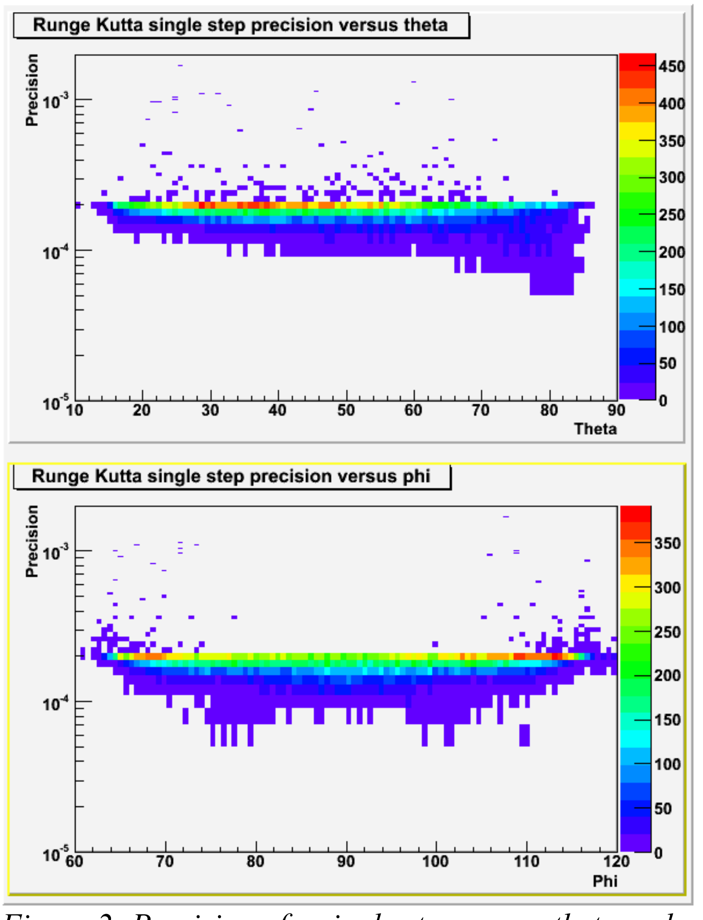


Figure 2: Precision of a single step versus theta and phi

To gain performance without loosing precision, respectively to increase precision, the actual RK tracking code adapt the stepsize depending on the momentum and field strength. For each step the precision is calculated and must be inside a predefined window. If the precision is to low, the step is repeated with a by 25% smaller stepsize. If it is to high, the next step will be 25% larger.

Actually the minimum required precision is 2.e-4 and the maximum 2.e-5. To avoid grid effects of the field map, the stepsize is not decreased further if it is already below 10 mm (typically 7.5 mm).

Figure 2 shows the angular distribution of the maximum precision of single steps. Only near the coils, the precision exceeds the required lower limit.

Near the planes, the stepsize may be larger then the distance to the plane. If the distance is smaller then 2.5 mm, the intersection point is calculated with a straight track. If it is larger, an additional step with the distance as stepsize is done.

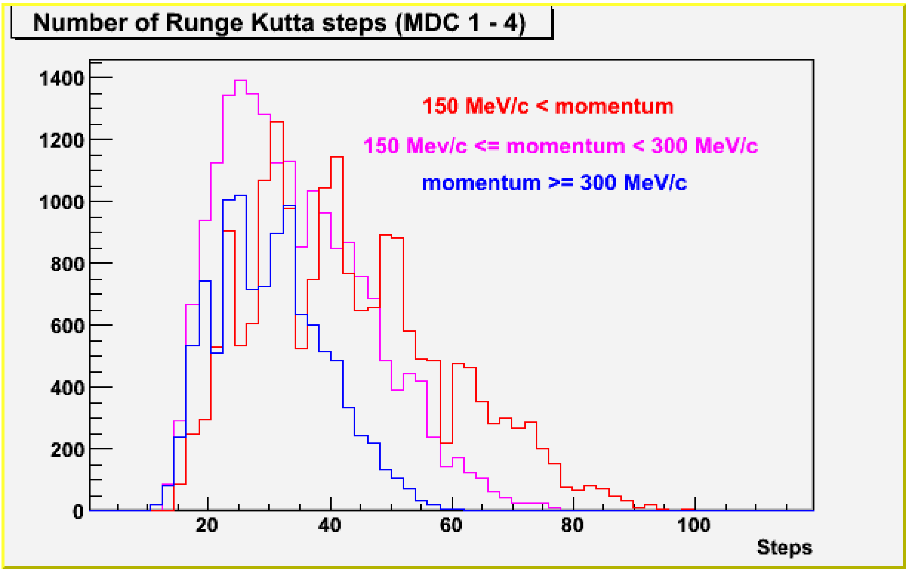


Figure 3: Number of Runge Kutta steps for one iteration

Fig. 3 shows the number of steps (including the repeated steps) needed to calculate the track between MDC1 and MDC4 for three different momentum bins. It clearly shows the decrease of steps with momentum.

The implementation of the adaptive stepsize was the main reason for the gain in performance by at least 40% as compared to the old code (under the same input conditions).

3. Comparison with GEANT

3.1. Ideal tracking of leptons for a simulation without any physics process

Runge Kutta does not take into account energy loss and multiple scattering. For low-energetic leptons the resolution is dominated by multiple scattering. To compare the results directly to GEANT, all physics processes (multiple scattering, secondaries, bremsstrahlung, ...) were switch off in the simulation and the data were analyzed with ideal tracking to avoid the smearing of the hit positions by the of MDC digitizer and the fitting.

Since this scenario is not realistic, it was not investigated in detail.

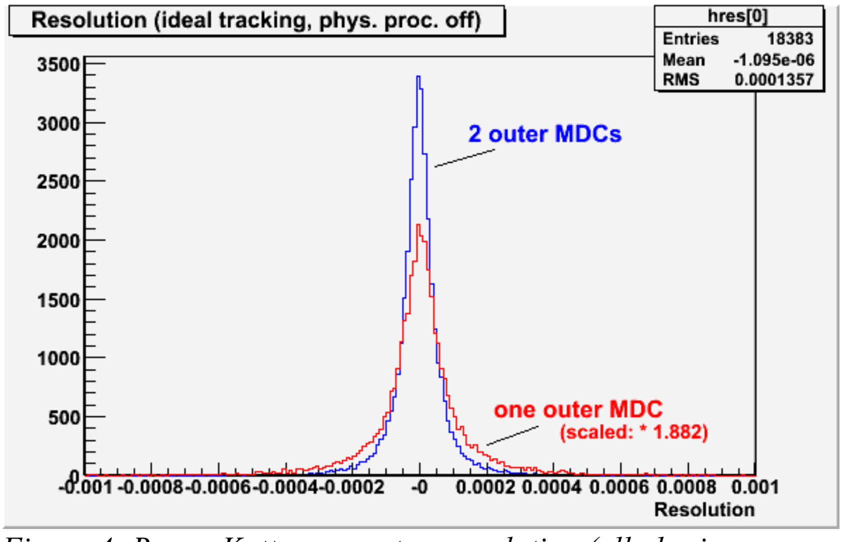


Figure 4: Runge Kutta momentum resolution (all physics processes switched off in the simulation, analyzed with ideal tracking)

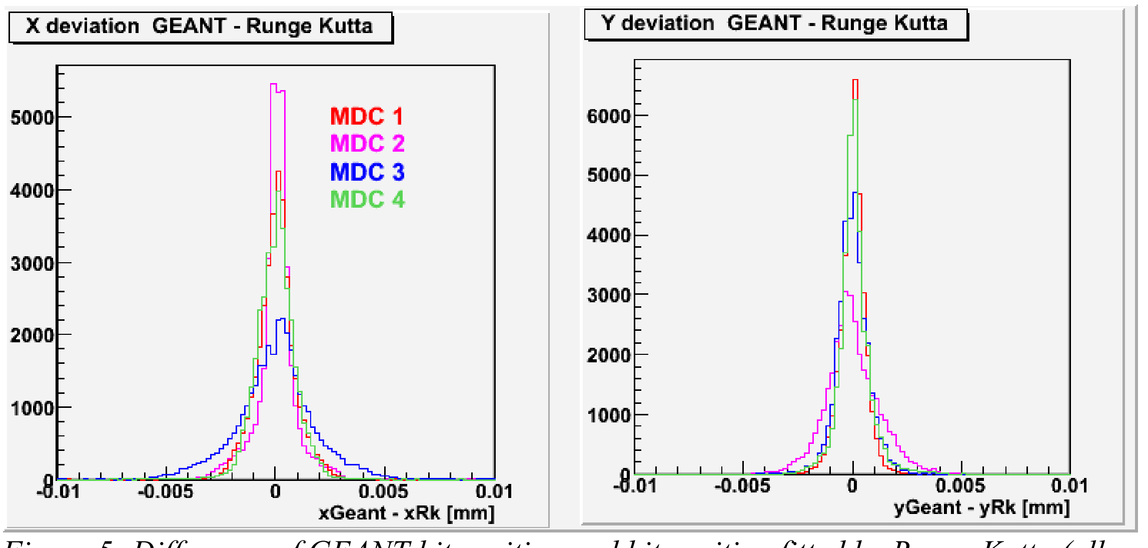


Figure 5: Difference of GEANT hit position and hit position fitted by Runge Kutta (all physics processes switched off in the simulation, analyzed with ideal tracking)

Although the hit positions and the momentum fitted with Runge Kutta agree almost perfect with the GEANT values, the theta and phi distribution of the inner segment calculated during the propagation to the target show a long tail especially for electrons at low momenta (Fig.6).

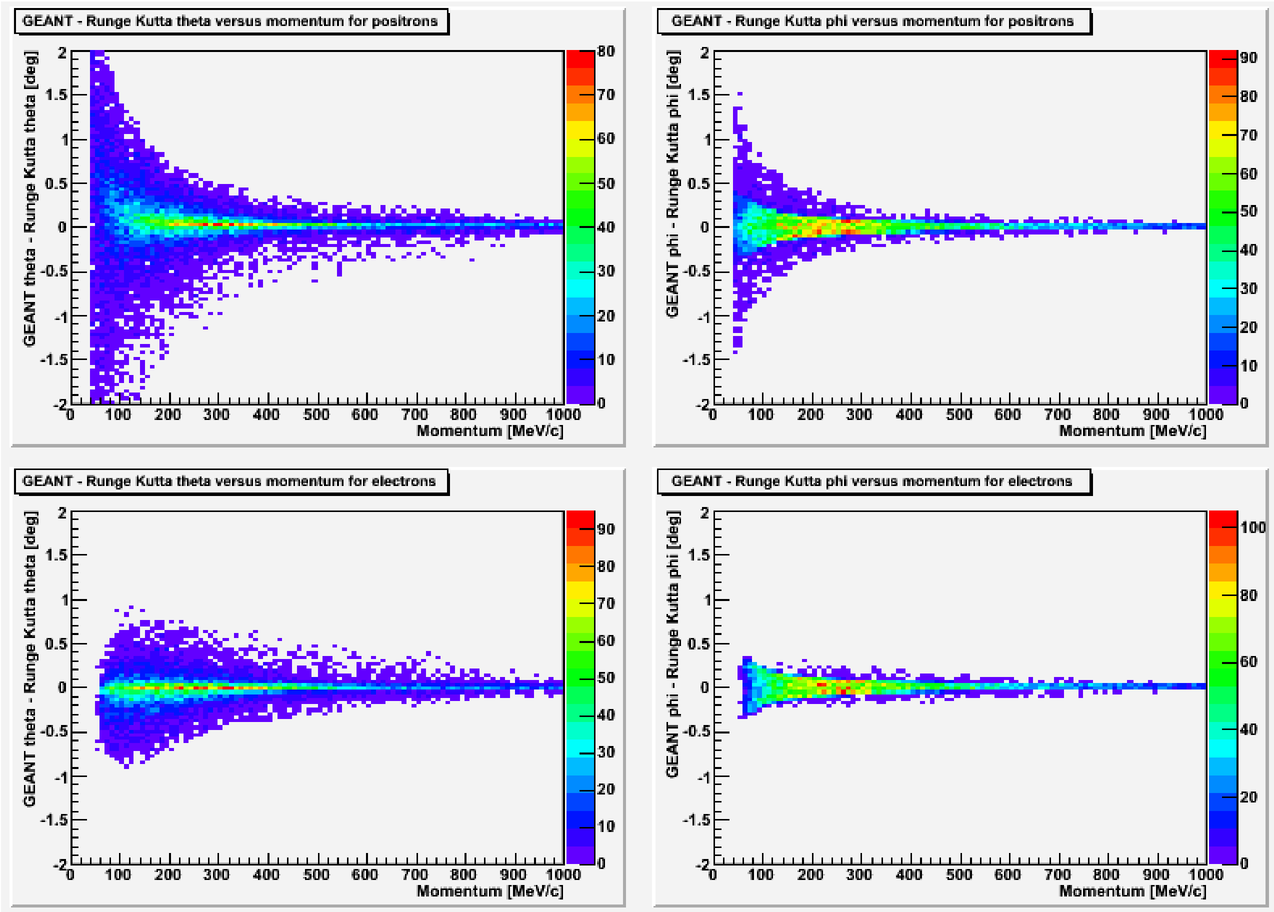


Figure 6: Difference in theta (left side) and phi (right side) for GEANT (Kine) and Runge Kutta inner segment versus momentum for positrons (upper pictures) and electrons (lower pictures) (all physics processes switched off in the simulation, analyzed with ideal tracking)

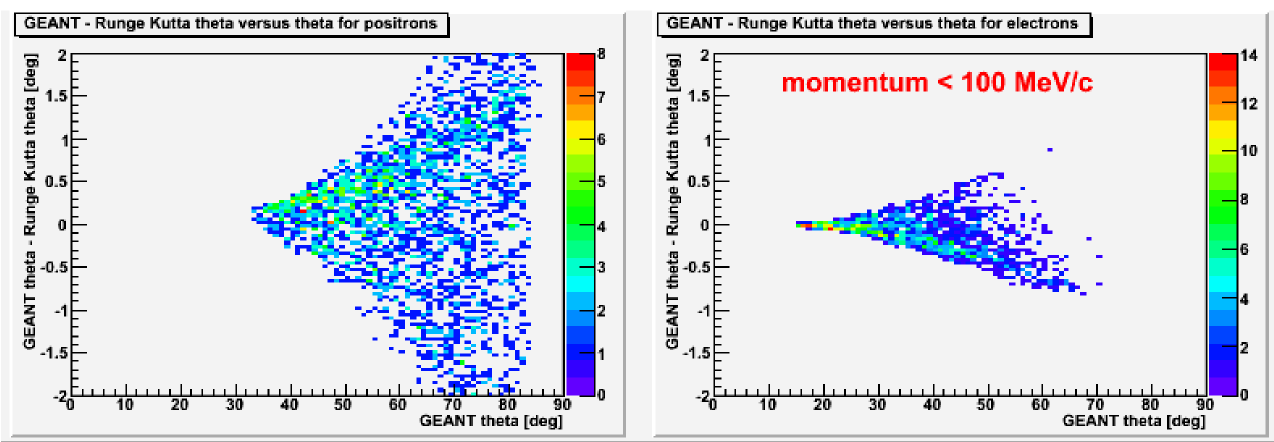


Figure 7: Difference in theta for GEANT (Kine) and Runge Kutta inner segment versus GEANT theta for positrons (left) and electrons (right) with momenta below 100 MeV/c (all physics processes switched off in the simulation, analyzed with ideal tracking)

The inner segment is calculated at a plane parallel to MDC1 at a distance of 300 mm. Wolfgang Koenig suggested, that eventually the result is still affected by the fringe field especially at large theta. Fig. 7 would support this speculation, but a recent test with a plane in a distance of 600 mm did not show any change. Non of the other variables (momentum and position resolution, single step precision) differ very much from the results at higher momenta and there is actually no explanation for the behavior. Looks like the Pluto input (with afterburner for conversion) puts multiple scattering in (double counting of multiple scattering, if used as GEANT input).

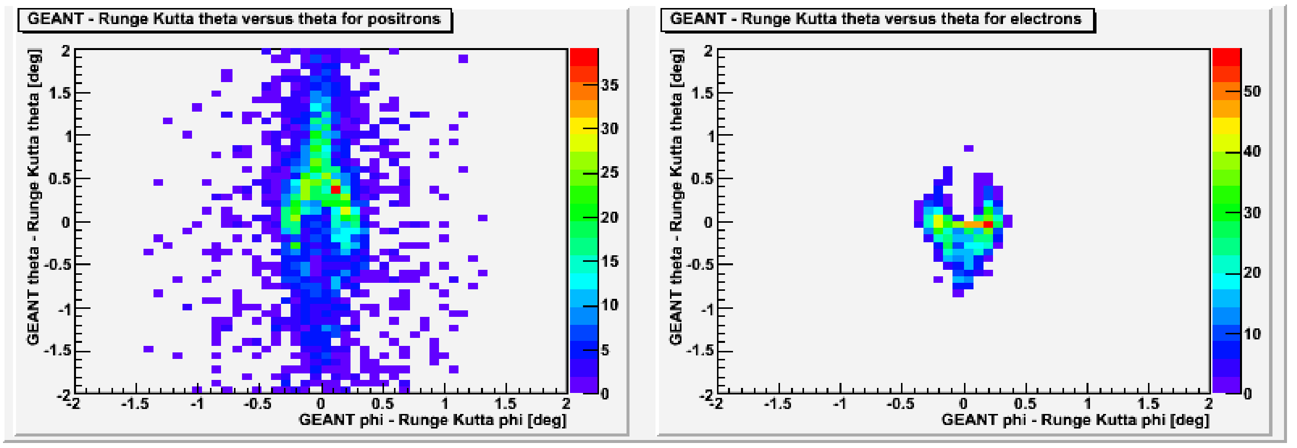


Figure 8: Difference in theta for GEANT (Kine) and Runge Kutta inner segment versus difference in phi for positrons (left) and electrons (right) with momenta below 100 MeV/c (all physics processes switched off in the simulation, analyzed with ideal tracking)

3.2. *Ideal tracking of leptons without multiple scattering*

The next pictures show the results for a simulation where all physics processes besides multiple scattering where switch on.

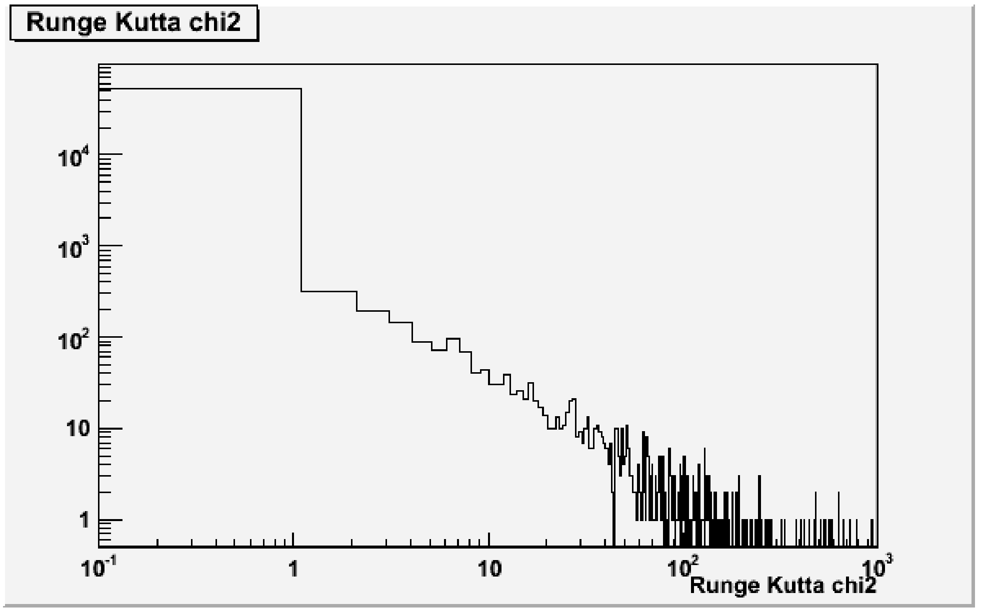


Figure 9: χ^2 distribution for ideal tracking without multiple scattering. Binning is delta χ^2 =1.

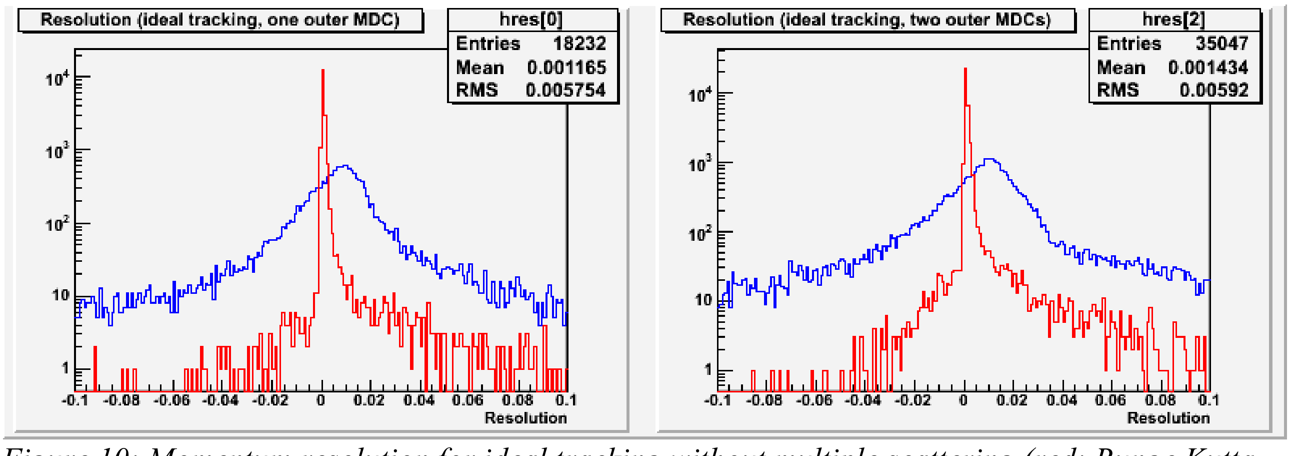


Figure 10: Momentum resolution for ideal tracking without multiple scattering (red: Runge Kutta, blue: Spline)

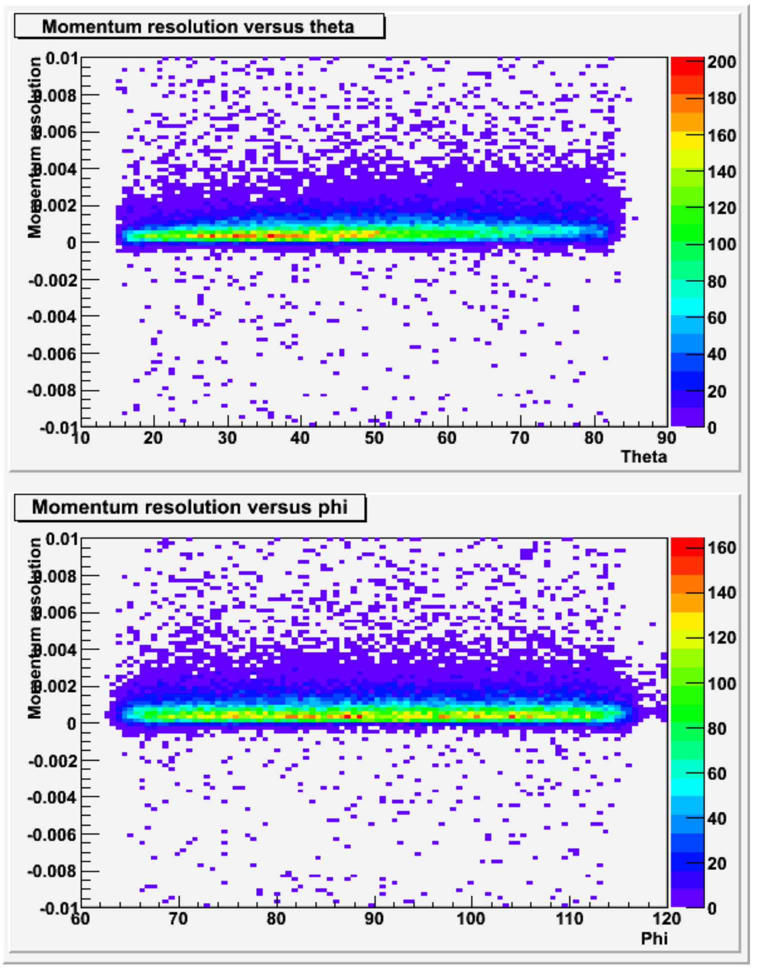


Figure 11: Angular distribution of the momentum resolution for ideal tracking without multiple scattering

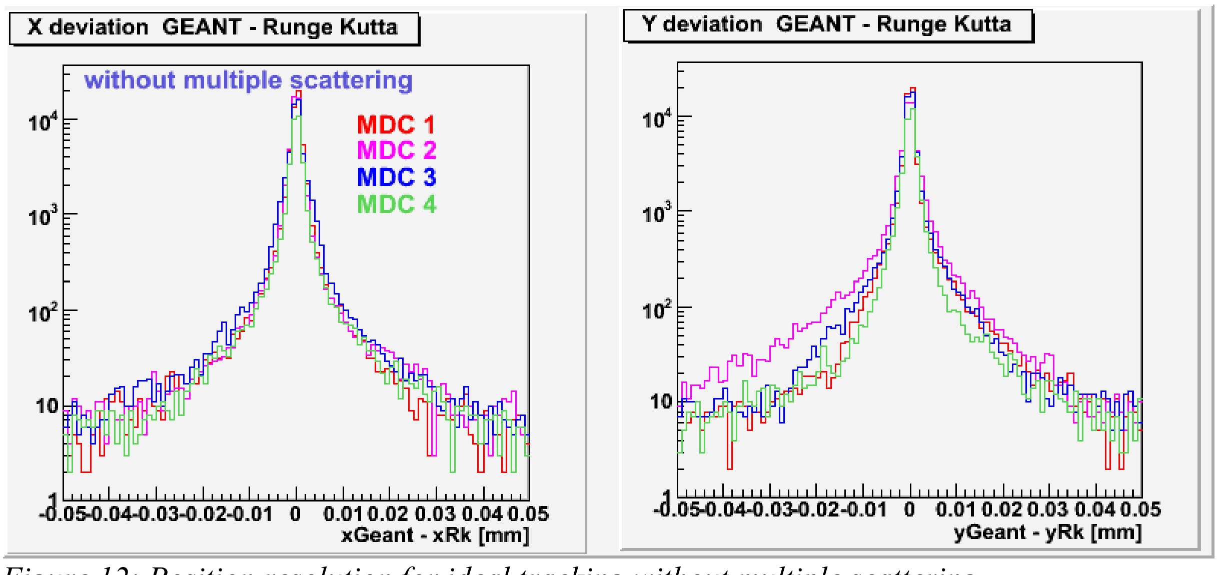


Figure 12: Position resolution for ideal tracking without multiple scattering

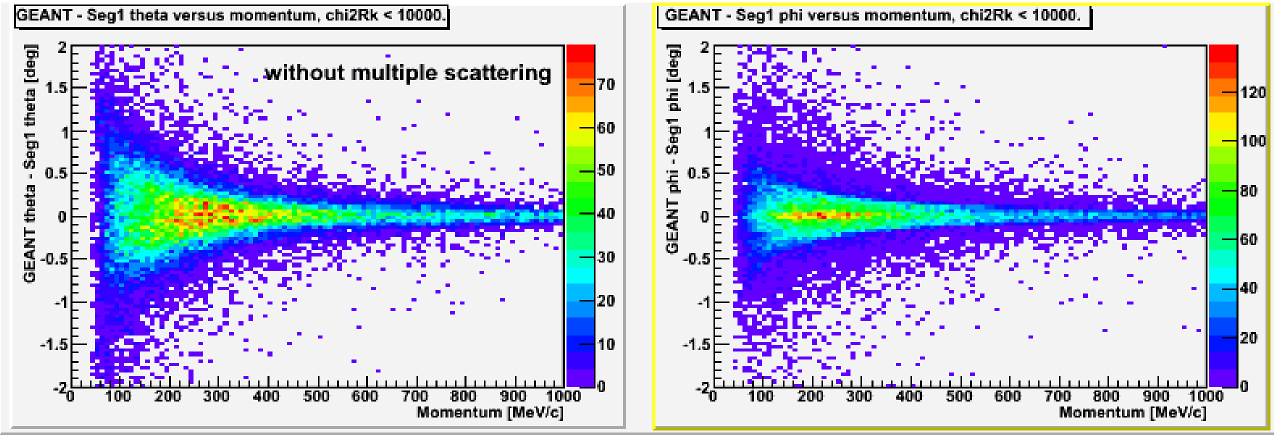


Figure 13: Difference in theta and phi of GEANT (Kine) and inner segment from MdcSeg for ideal tracking without multiple scattering (with Runge Kutta χ^2 cut < 10000. as in Fig. 9, Pluto events)

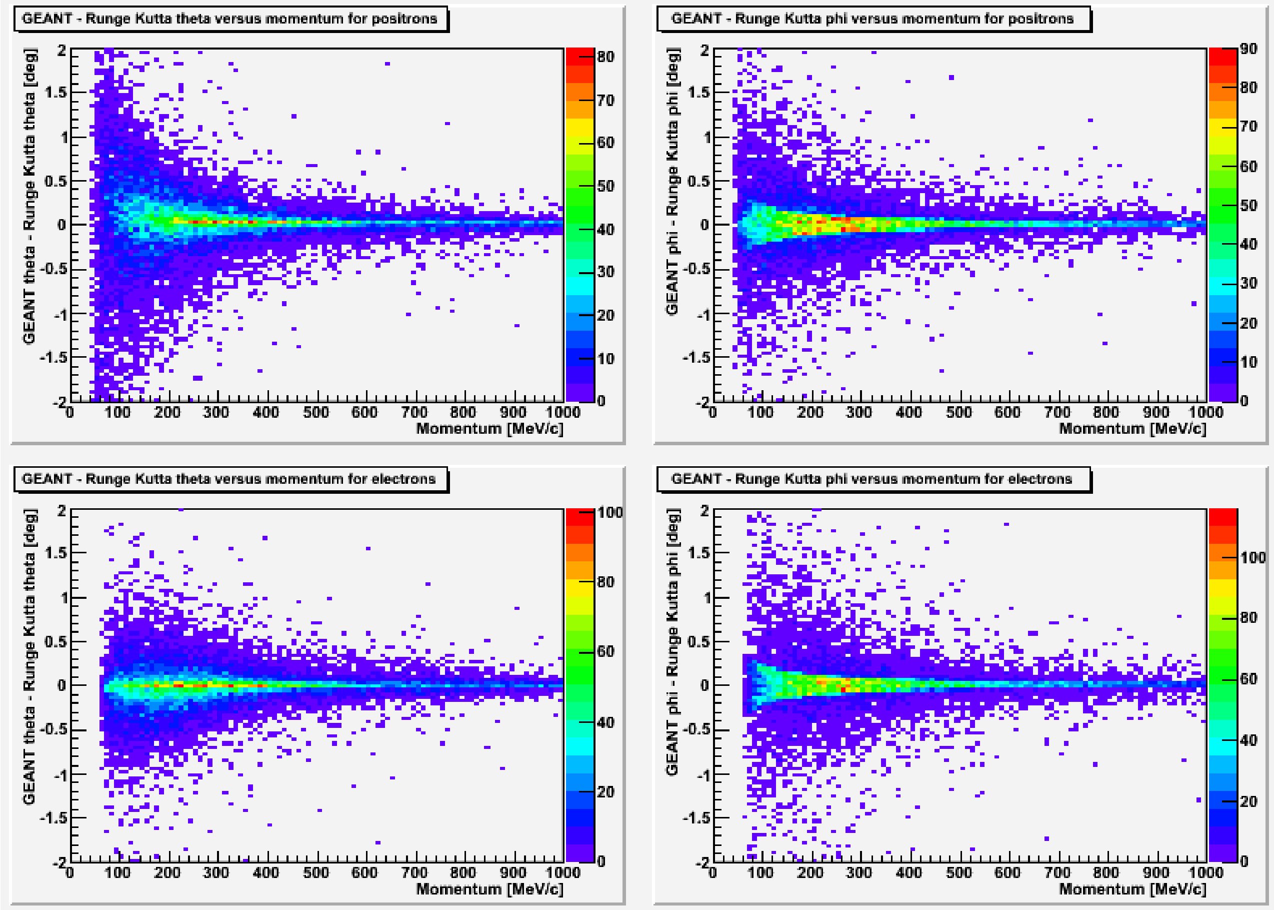


Figure 14: Same as Fig.9, but separately for positrons(upper) and electrons(lower)

3.3. *Ideal tracking of leptons with multiple scattering*

All pictures in this chapter use as input a simulation, where all physics processes were switched on including multiple scattering. MDC tracking was done with ideal tracking.

Only primary tracks where selected, where the track number of the inner segment is the same as the track number of the outer segment. Tracks without an outer segment were skipped.

Momentum resolution

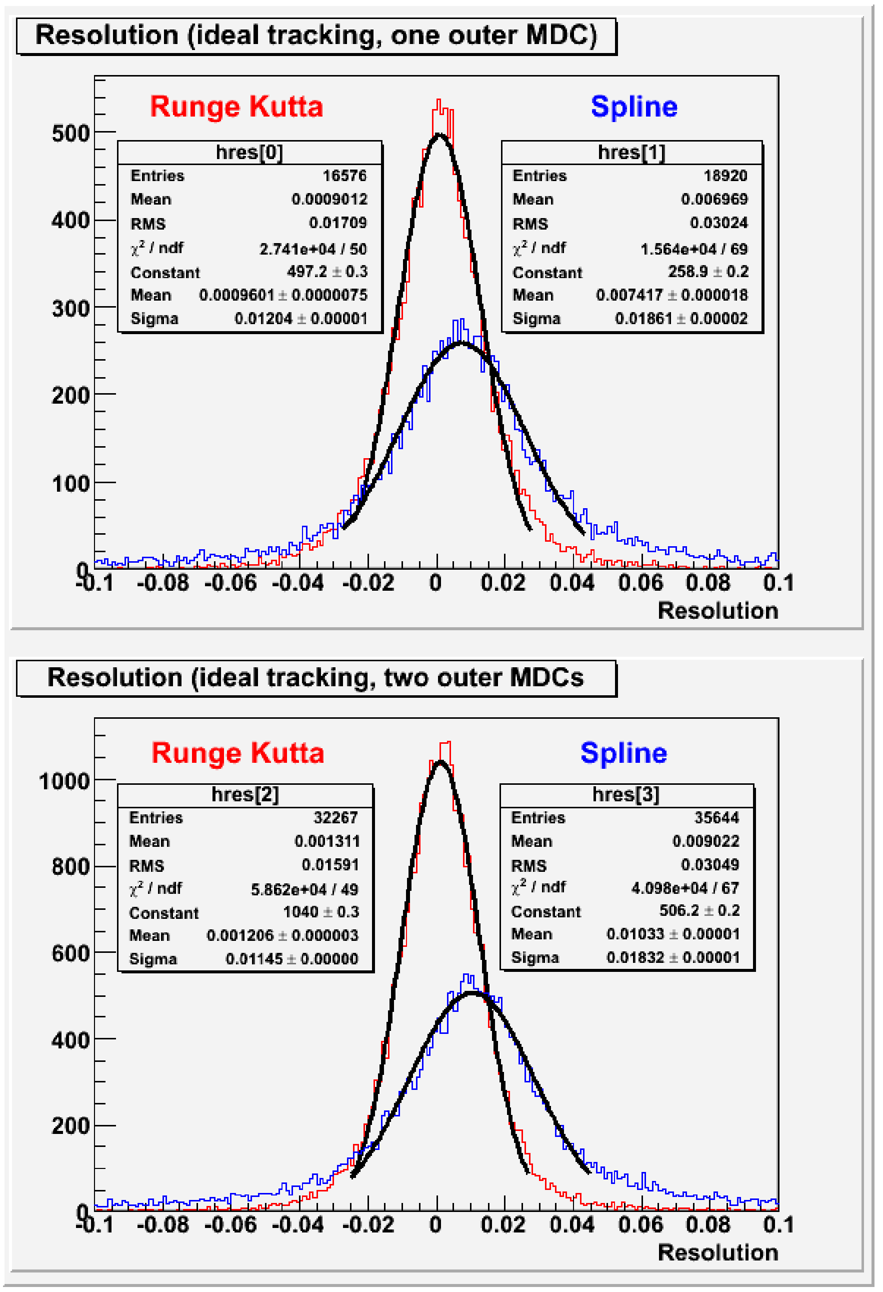


Figure 15: Spline and Runge Kutta momentum resolution for ideal tracking (Runge Kutta $\chi^2 < 100000$)

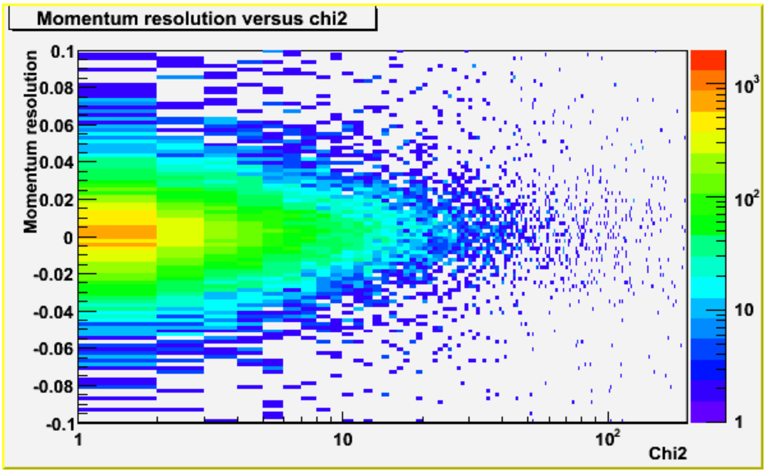


Figure 16: Runge Kutta momentum resolution versus χ^2 for ideal tracking

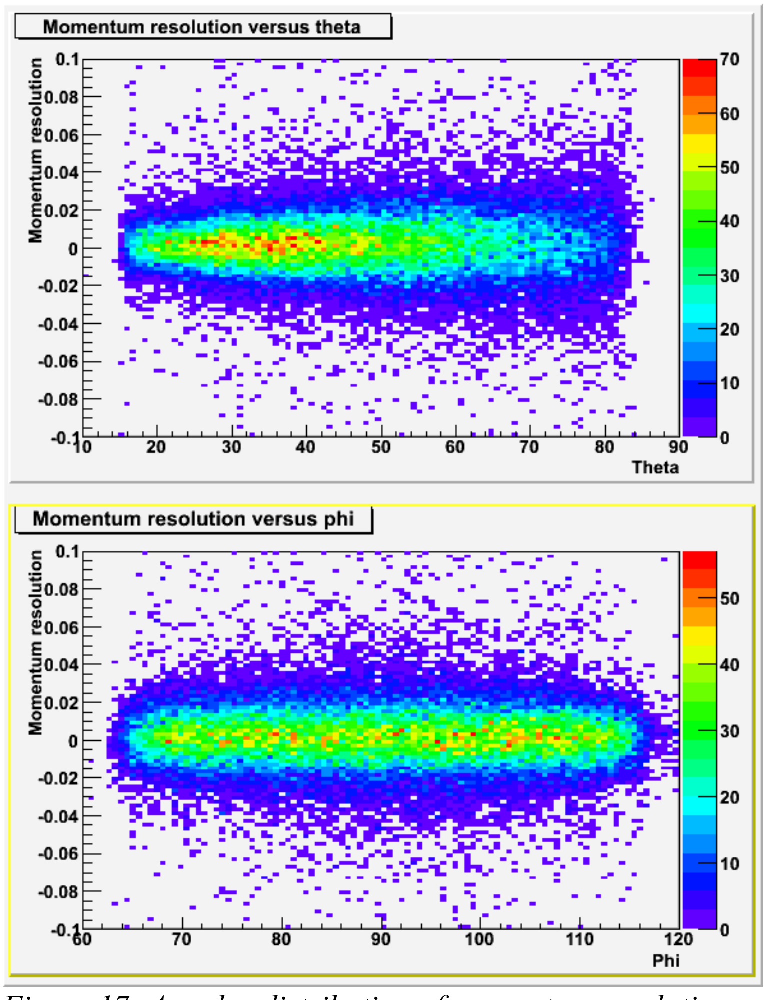


Figure 17: Angular distribution of momentum resolution for ideal tracking (Runge Kutta $\chi^2 < 100000$)

Position resolution

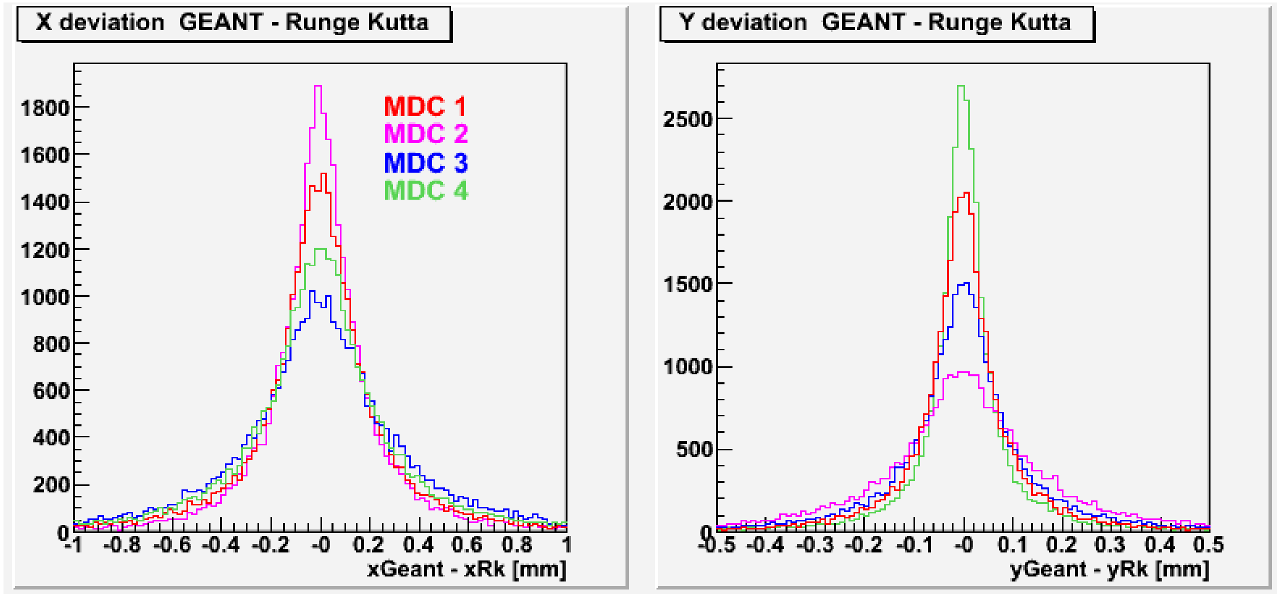


Figure 18: Difference of GEANT hit position and hit position fitted by Runge Kutta (χ^2<100000., two outer MDCs)

χ^2 distribution

The initial χ^2 , show on the y-axis in Fig. 19, is the χ^2 calculated with the original Spline momentum and the original hits calculated from the MDC segments.

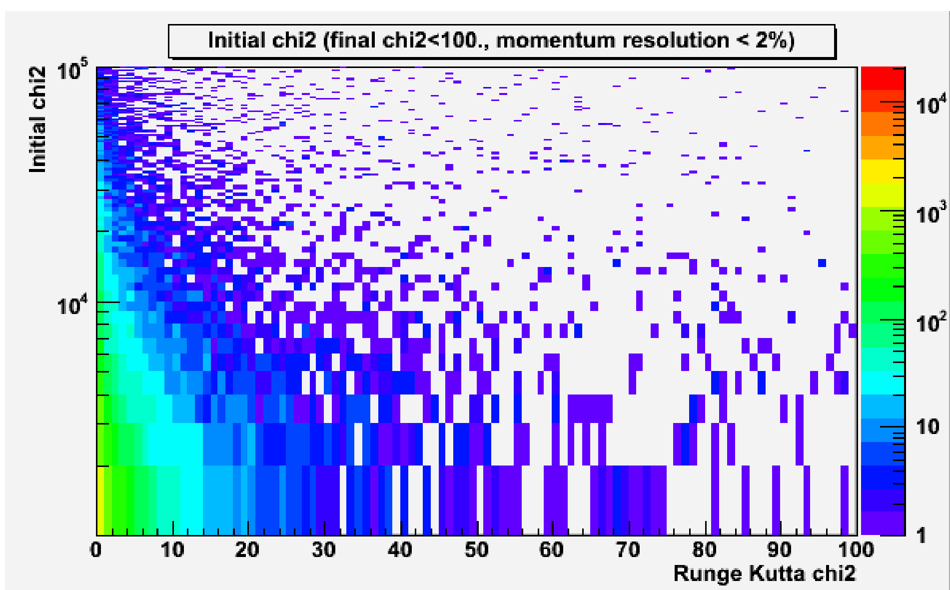


Figure 19: Runge Kutta χ^2 calculated before fitting (y-axis) versus χ^2 after fitting for a momentum resolution < 2%
(no cut on momentum and Spline quality and without requiring a hit in Meta)

The large initial χ^2 is caused by large Spline momentum deviations from the GEANT momentum, especially in cases for only one outer MDC.

Efficiency

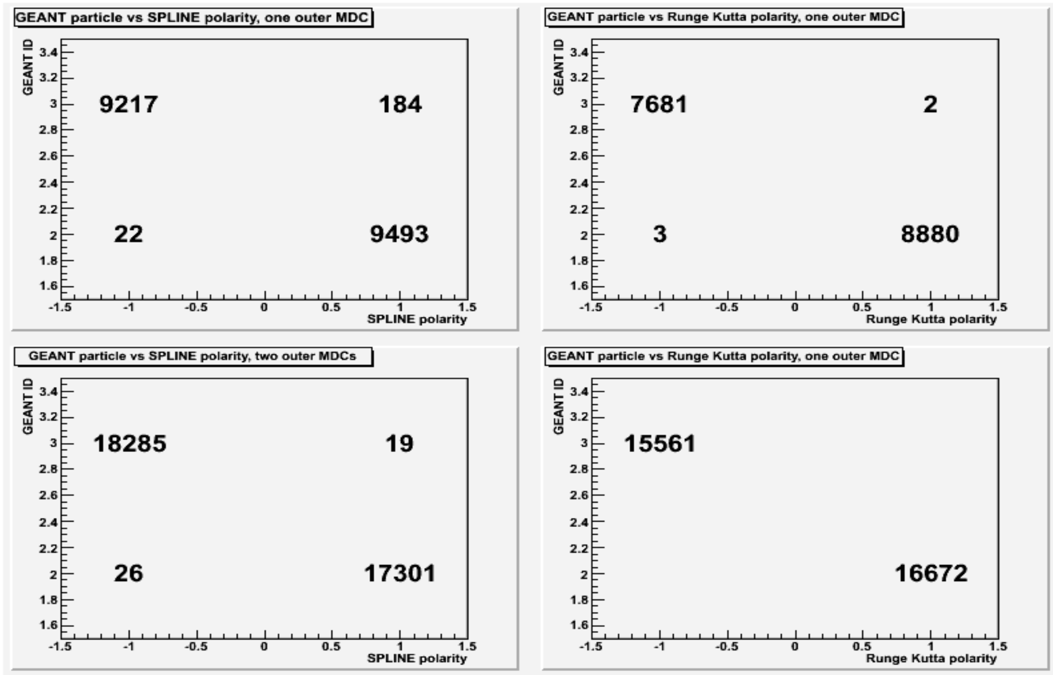


Figure 20: GEANT particle id versus polarity for ideal tracking (id 2: positrons, id 3: electrons), on th left side for Spline and on the right side for Runge Kutta for one(upper pictures) and two (lower pictures) outer MDCs (cuts: qSpline > -1, RK $\chi^2 < 10000$.)

Without a proper cut on the Spline quality, the spline efficiency is almost 100% for good tracks (primary tracks, track number of inner segment = track number of outer segment), although the momentum might differ significantly from the GEANT momentum.

There table below shows the Runge Kutta efficiency normalized to Spline for $\chi^2 > 10000$ for all momenta without requiring a hit in Meta..

	One outer MDC	***Two outer MDCs***
electrons	83.3%	93.5%
positrons	85.1%	96.4%

Fig. 21 shows the difference between the GEANT and the Spline momentum versus the GEANT momentum for the cases, when Runge Kutta tracking fails and Fig. 22 the cases, where it succeeds.

At low momenta, Runge Kutta is very sensitive to a wrong initial momentum. After the first iteration, the momentum might be too small and the track start curling. In this case, Runge Kutta does not converge.

In the actual implementation of the code, the initial momentum is set to 50 MeV/c, in the cases, where the Spline momentum is lower. As Fig 21 might indicate, that this is eventually too high for positrons. But a recent test showed almost no improvement after lowering the value to 30 MeV/c.

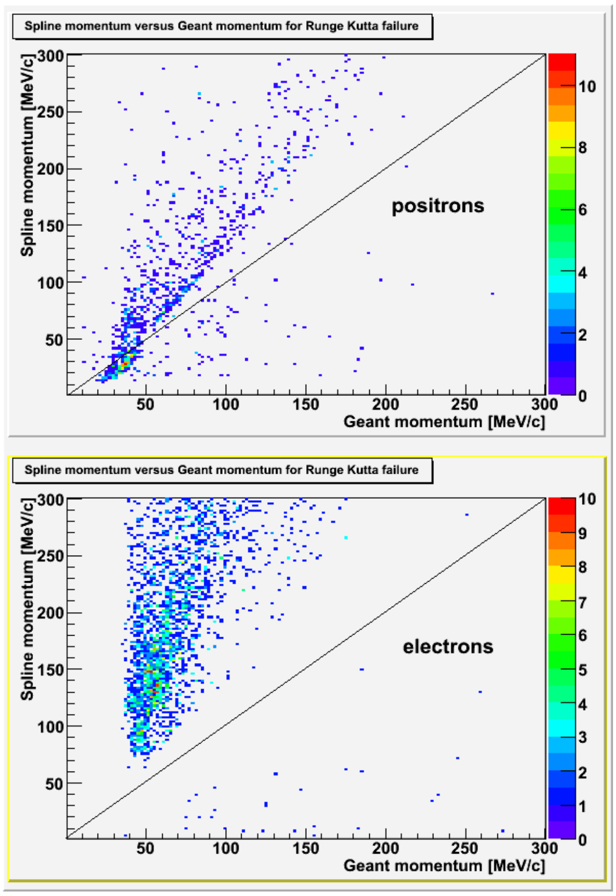


Figure 21: Spline momentum versus GEANT momentum for Runge Kutta $\chi^2 >= 10000$.

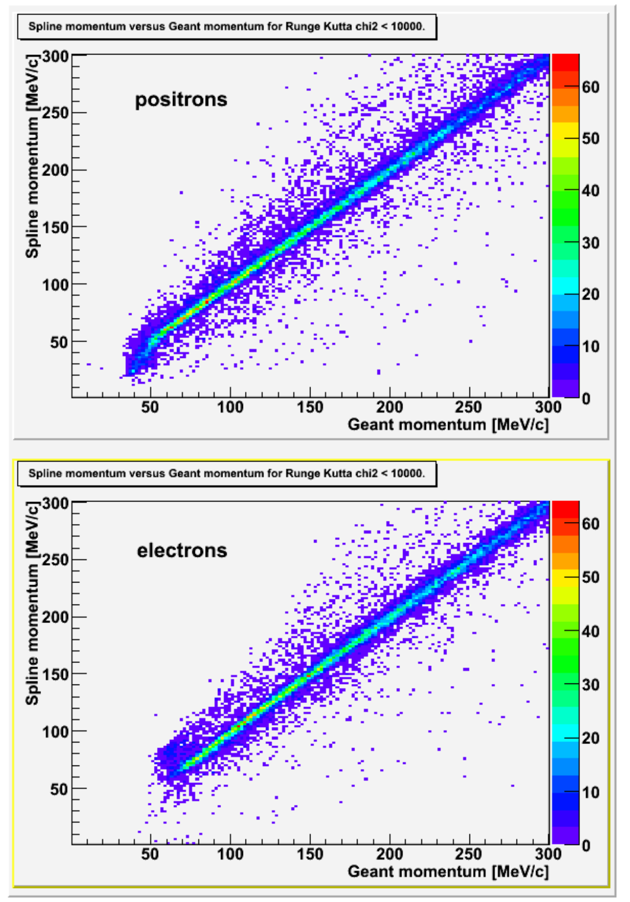


Figure 22: Spline momentum versus GEANT momentum for Runge Kutta $\chi^2 < 10000$.

Pathlength

The Spline pathlength is systematically too short, while the Runge Kutta pathlength agrees quite well with the GEANT tracklength, besides a tail at low momenta for hits in TOF (Fig. 23).

The beta value calculated with the Runge Kutta pathlength and the time-of-flight from the TOF hit is shifted by 1.3% and even more for Tofino(Fig. 24). If one calculates beta with the GEANT time-of-flight, the shift disappears. For a more detailed investigation see entry "**Systematic shift of Runge Kutta beta value by 1.3% for leptons**" in HADES Forum, HADES-Tracking.

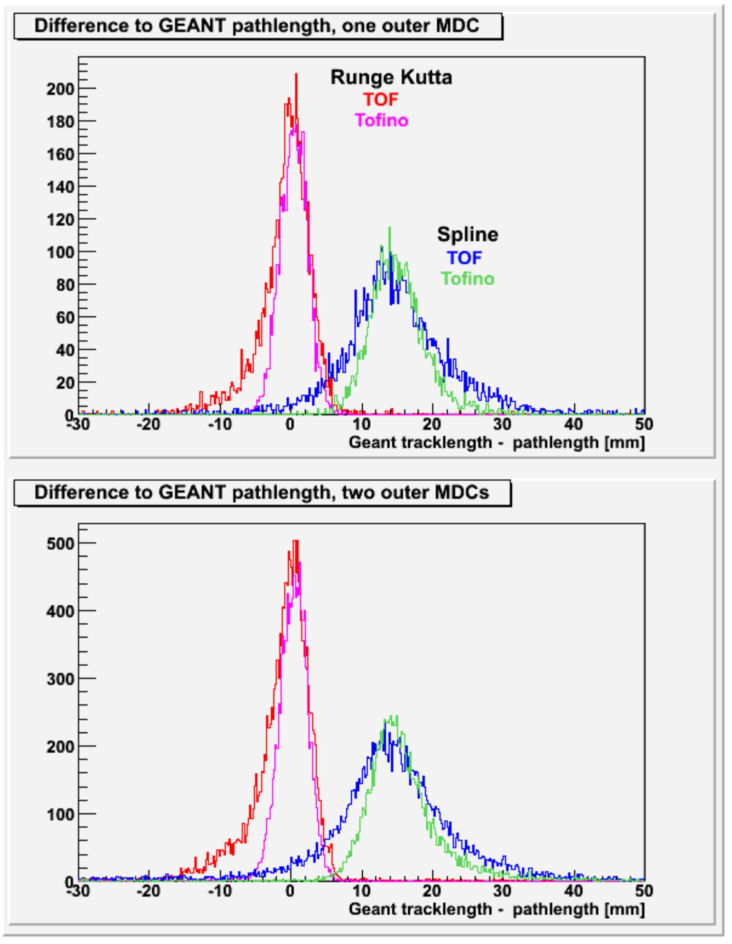


Figure 23: Comparison with GEANT pathlength for three (upper) and four (lower) MDCs (Runge Kutta $\chi^2 < 10000$)

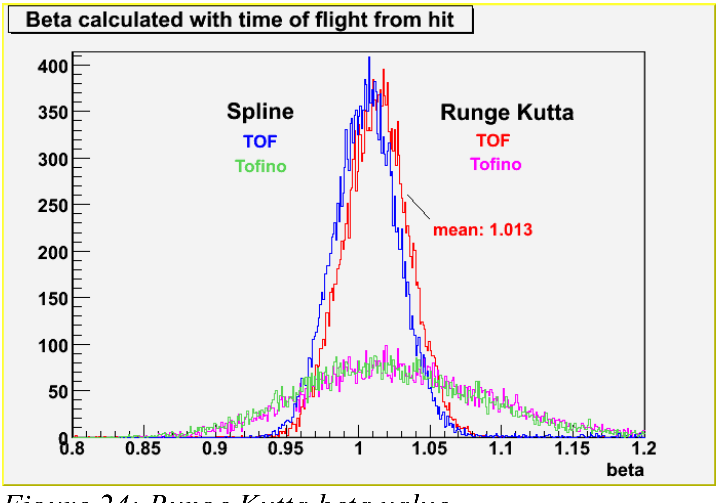


Figure 24: Runge Kutta beta value

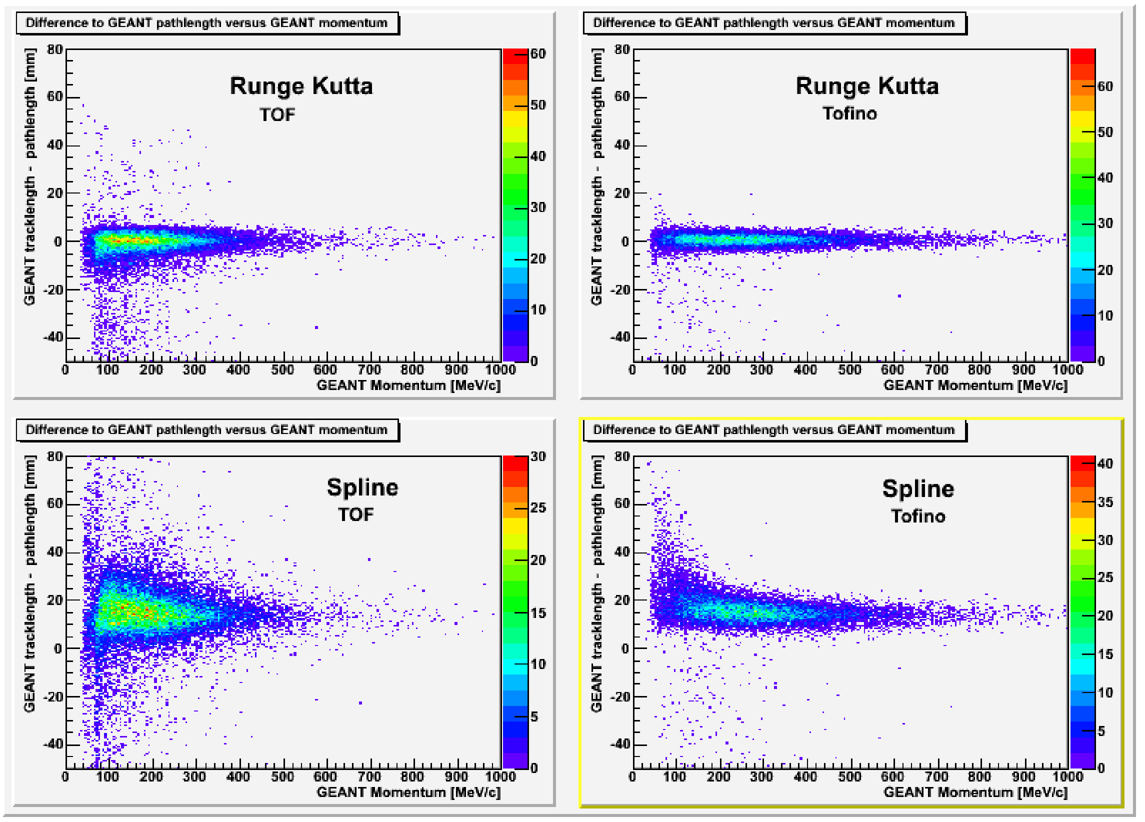


Figure 25: Comparison with GEANT pathlength as a function of the GEANT momentum at MDC1(Runge Kutta $\chi^2 < 10000$)

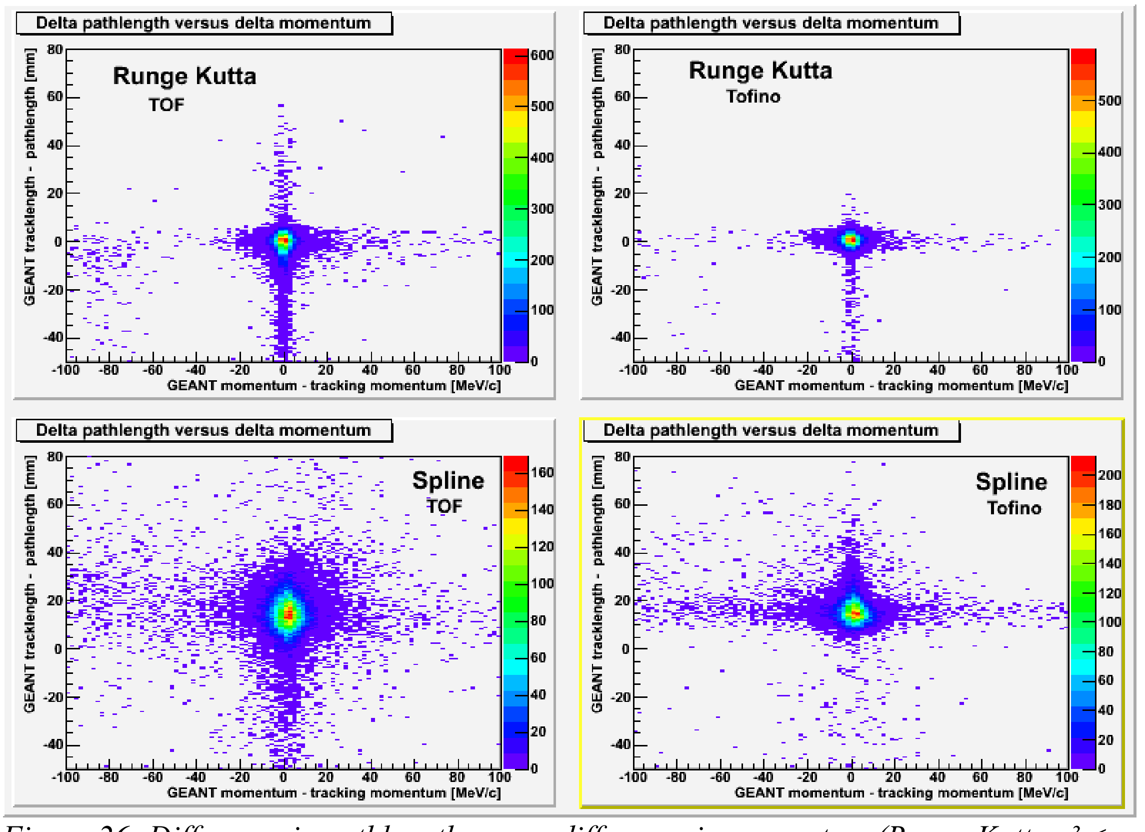


Figure 26: Difference in pathlength versus difference in momentum (Runge Kutta $\chi^2 < 10000$)

Inner segment fitted by Runge Kutta

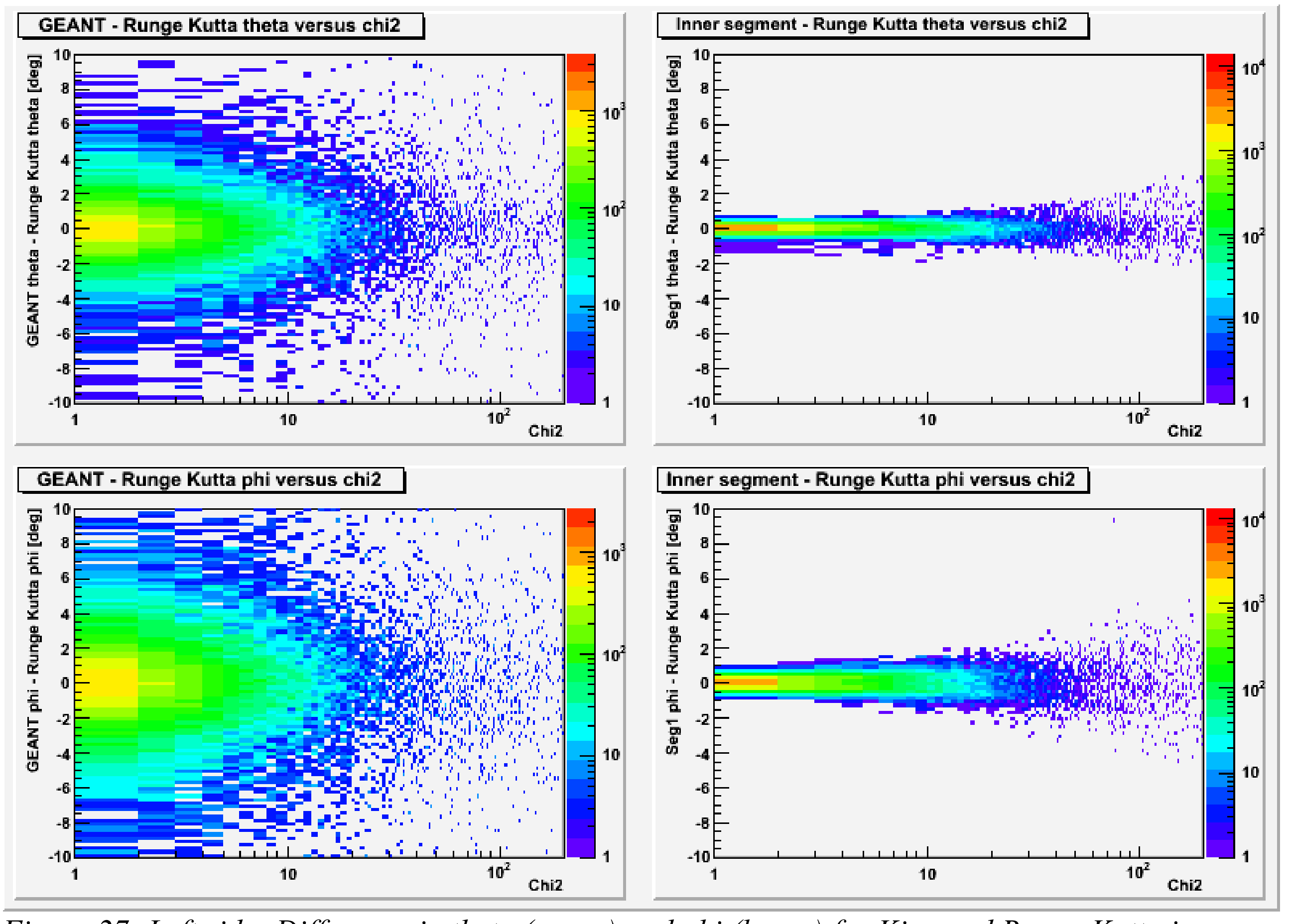


Figure 27: Left side: Difference in theta (upper) and phi (lower) for Kine and Runge Kutta inner segment versus χ^2; Right side: Difference between the MDC inner segment and Runge Kutta

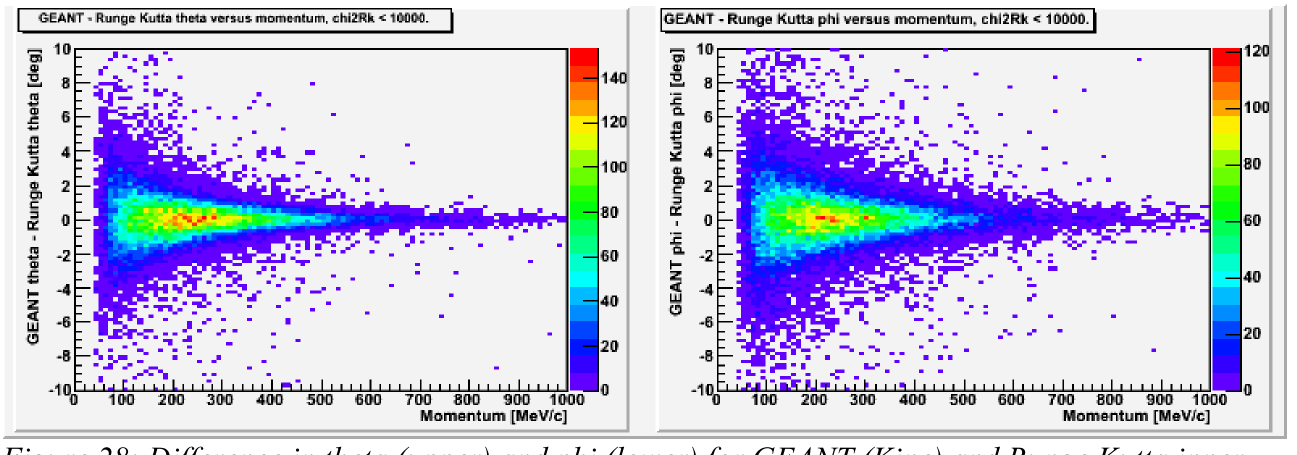


Figure 28: Difference in theta (upper) and phi (lower) for GEANT (Kine) and Runge Kutta inner segment versus momentum

The widening of the angular distribution is caused by the multiple scattering in the target and the RICH, not taken into account by Runge Kutta.

3.4. Real tracking

The same GEANT simulation used as input for the pictures in chapter 3.3 were also analyzed with real tracking.

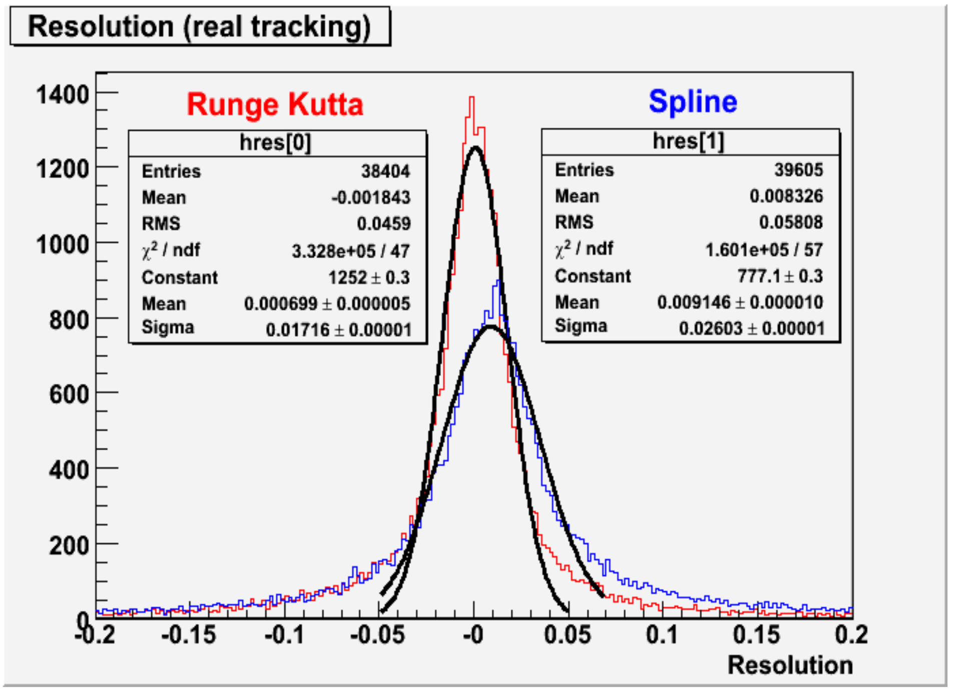


Figure 29: Spline and Runge Kutta momentum resolution (cuts: Runge Kutta $\chi^2 < 10000$), qSpline>-1, hit in Meta)

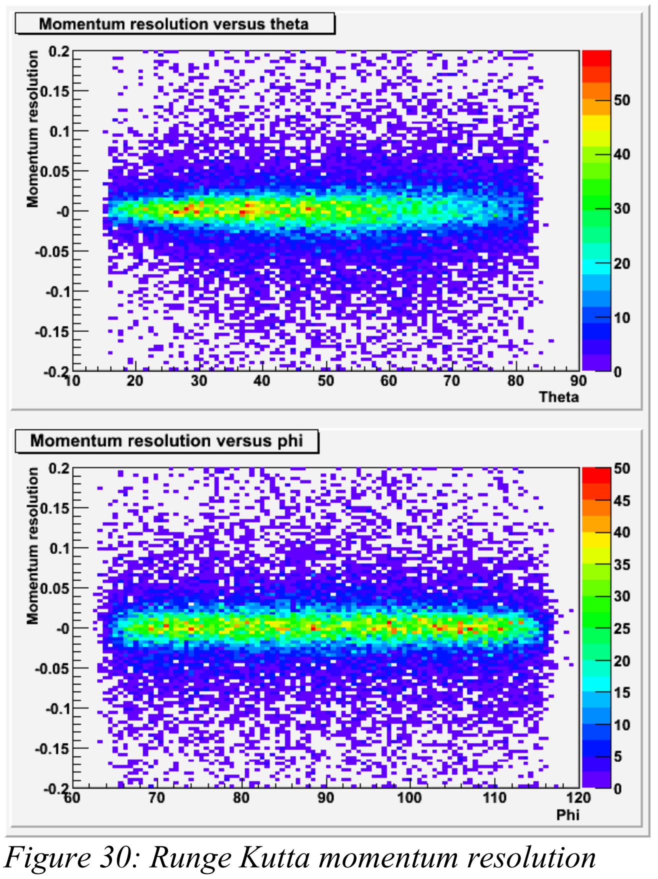


Figure 30: Runge Kutta momentum resolution versus theta and phi of MDC inner segment (same cuts as in Fig. 29)

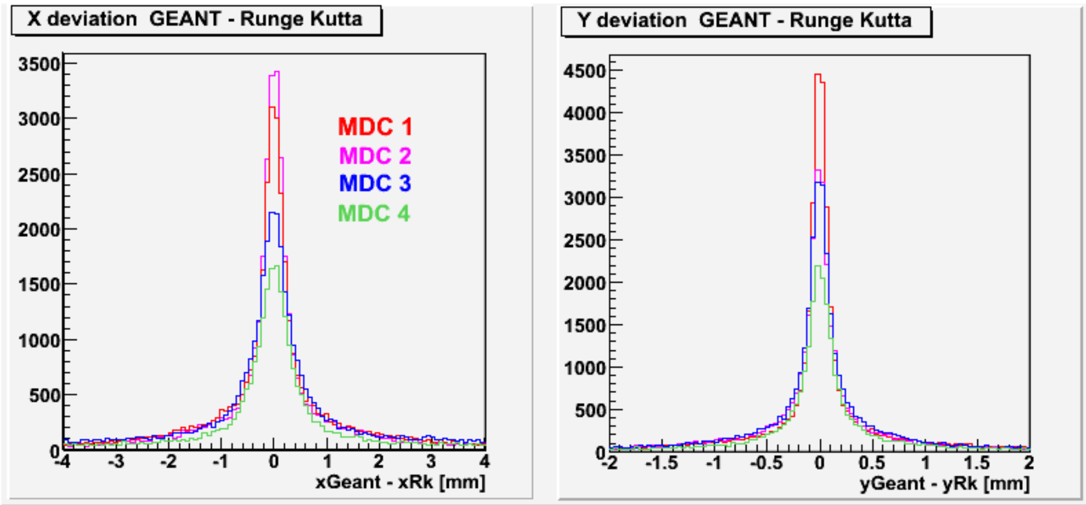


Figure 31: Difference of GEANT hit position and hit position fitted by Runge Kutta (cuts: Runge Kutta $\chi^2 < 10000$), qSpline>-1, hit in Meta)

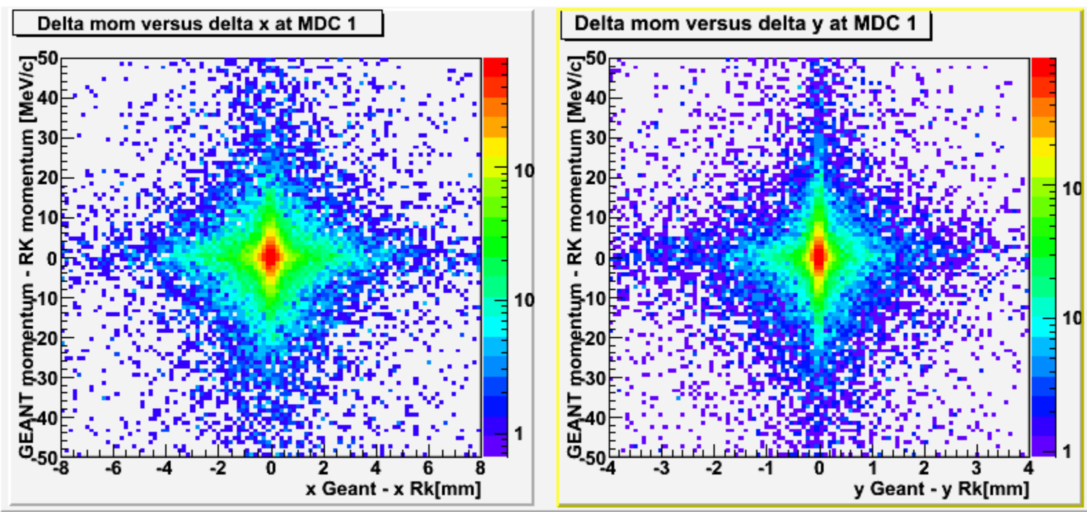


Figure 32: Difference of GEANT and Runge Kutta momentum versus difference of hit positions in MDC1 (same cuts as in Fig. 31)

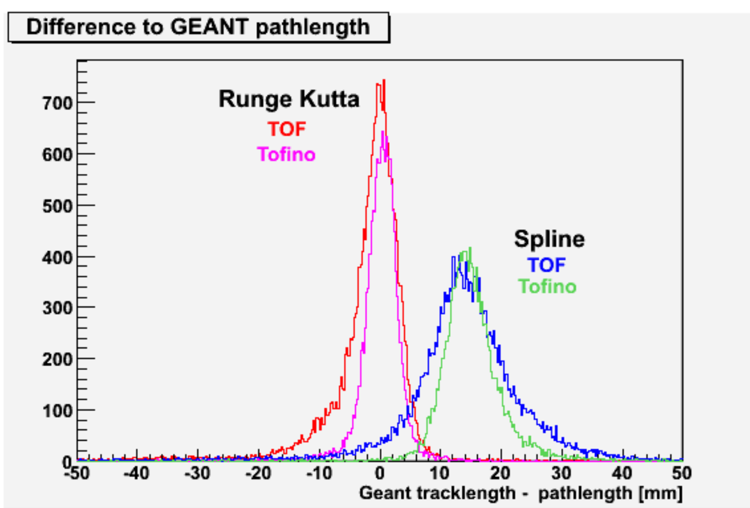


Figure 33: Difference between GEANT and Runge Kutta track length for real tracking (same cuts as in Fig. 31)

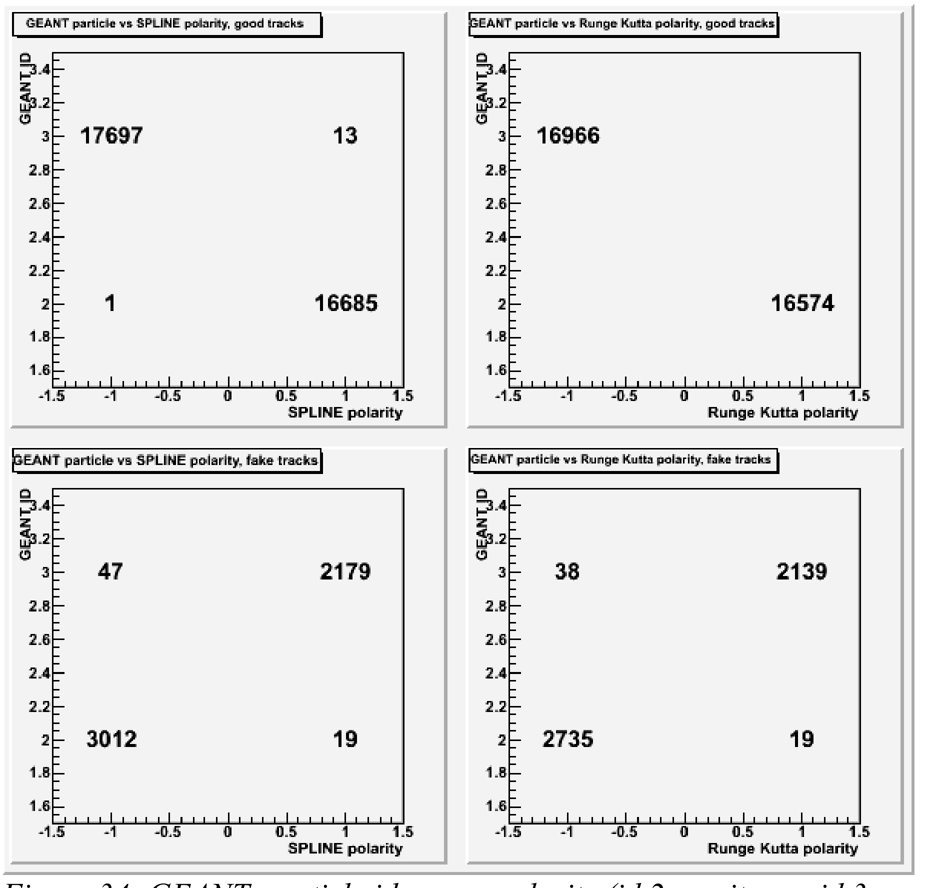


Figure 34: GEANT particle id versus polarity (id 2: positrons, id 3: electrons), on th left side for Spline and on the right side for Runge Kutta for good tracks (upper pictures) and fake tracks (lower pictures) (cuts: qSpline > -1, RK χ^2 < 10000., hit in Meta)

"Good" tracks are defined as tracks with same track number in the inner and outer segment, "fake" tracks are tracks, where the track numbers are different (combining an electron in the inner segment with a positron in the outer segment). For small opening angles, Runge Kutta may fit these fakes, although the χ^2 is larger.

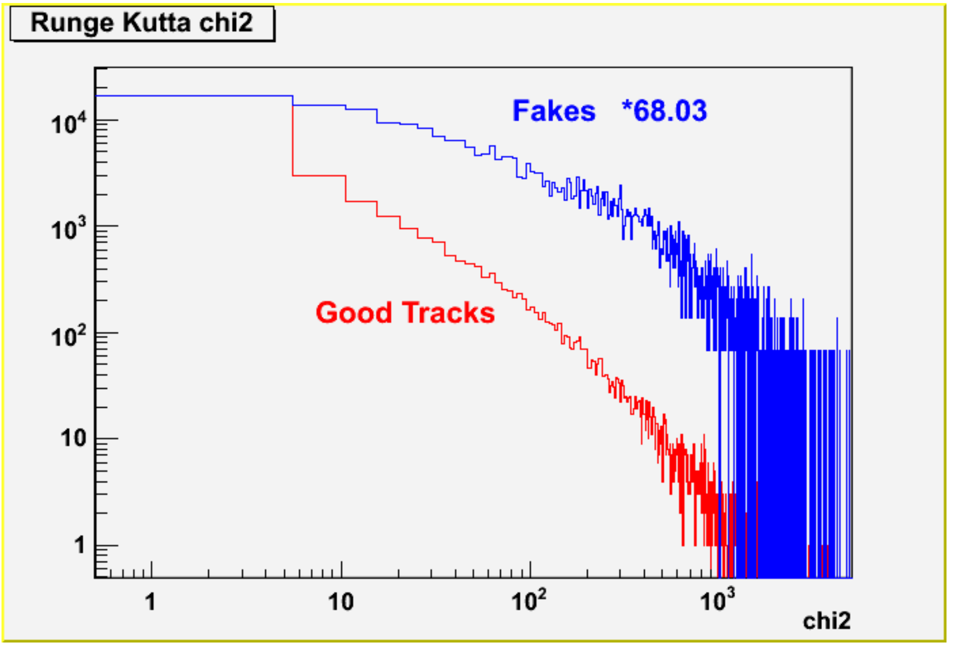


Figure 35: Runge Kutta χ^2 distribution for good and fake tracks (cuts: qSpline > -1, hit in Meta)
The fakes are scaled to the first bin (<50) of the good tracks.

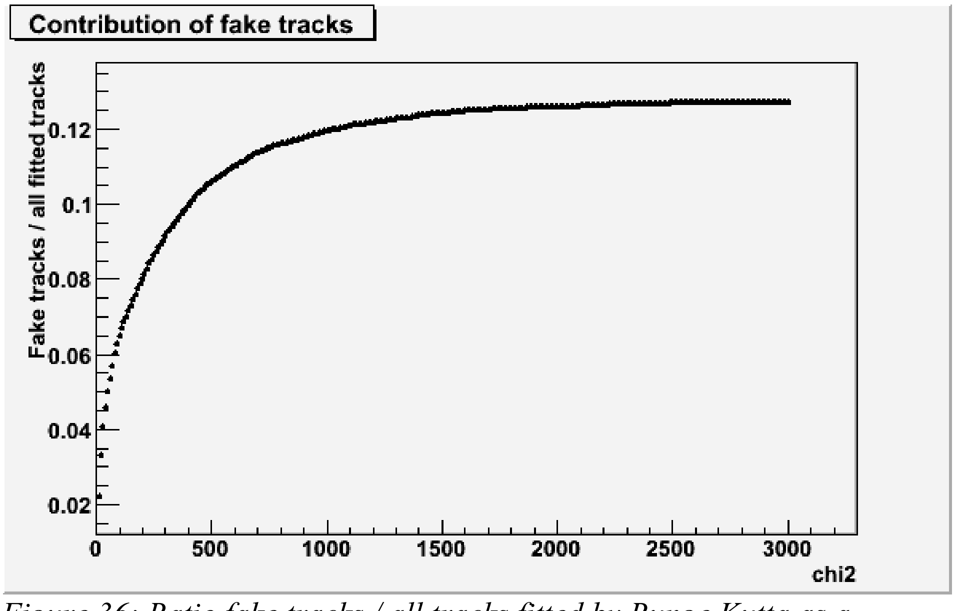


Figure 36: Ratio fake tracks / all tracks fitted by Runge Kutta as a function off the χ^2 cut (same cuts as in Fig. 35)

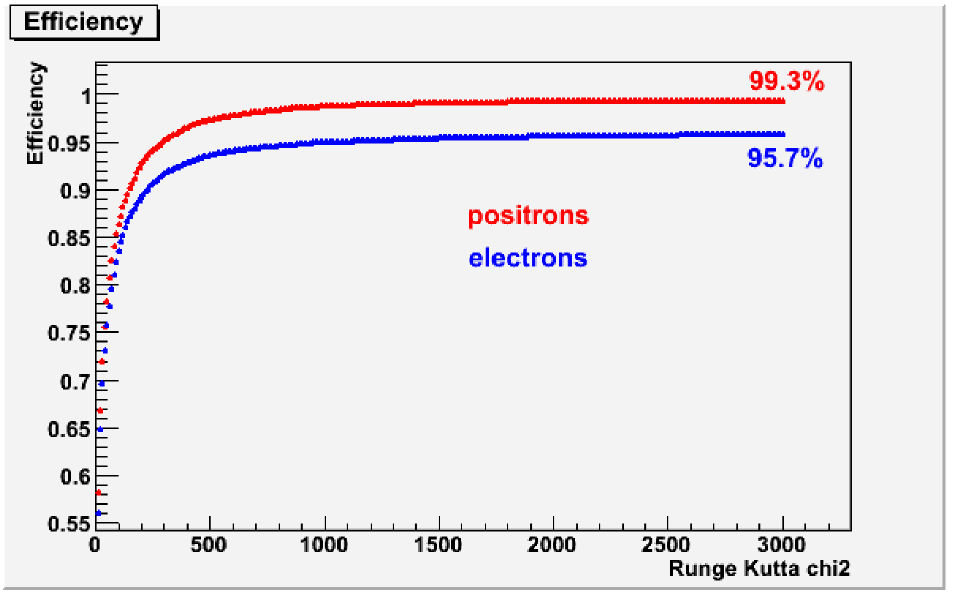


Figure 37: Runge Kutta efficiency as a function off the χ^2 cut (cuts: qSpline > -1, RK χ^2 < 10000., hit in Meta)

The efficiency in Fig. 36 is defined as

$$effiency = \frac{number\ of\ \ good\ tracks\ fitted\ with\ Runge\ Kutta}{number\ of\ all\ good\ tracks}$$