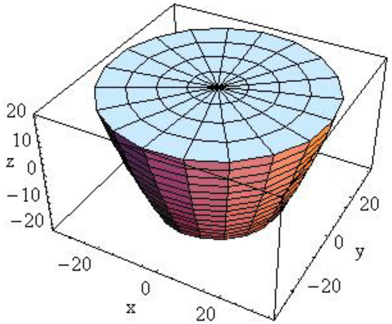
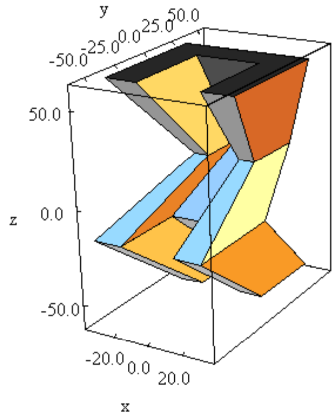
Geant4 workshop, Hebden Bridge 13-19 september 2007



G4Paraboloid

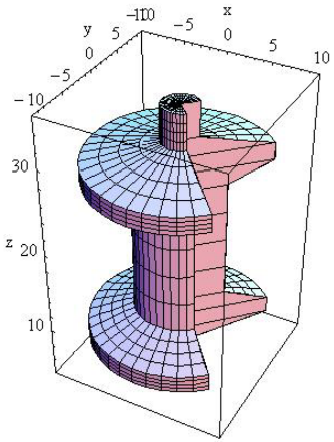


G4ExtrudedSolid

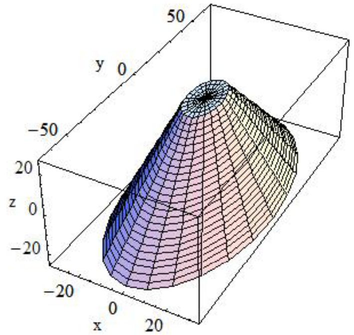
Geometry and Field:
New features and Developments

T. Nikitina

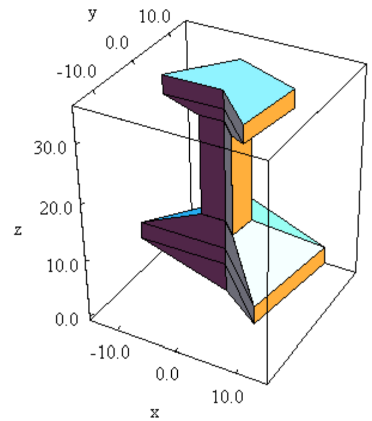
For Geometry Working Group



G4Polycone



G4EllipticalCone



G4Polyhedra

Summary

Geometry

- **Overview of** New features in Geometry
- Future planes

Geometry/ Magnetic Field

- Improvements in LocateIntersectionPoint()
- New Steppers : RKG3_Stepper and HelixMixedStepper
- Steppers in non-homogeneous Field (Gradient Field , ATLAS Toroid Field)
- Conclusions and future plans

Parallel Geometries & Navigation

since 4.8.2, default for biasing in 4.9

- Possibility to define geometry trees which are "parallel" and overlapping to the tracking geometry
  - Each assigned to a dedicated navigator object
  - Navigation transparently happens in sync with the normal tracking (mass) geometry
    - Applies transparently to transport in magnetic field
- Use cases: fast shower parameterisation, geometrical biasing, particle scoring, readout geometries, etc ...
- Parallel transportation activated only after registration of the parallel geometry in the setup

Tunable geometrical tolerance

since 4.9

- Cartesian tolerance (`kCarTolerance`) value for accuracy of tracking on the surfaces can be defined *relative* to the extent of the world volume
  - `G4GeometryManager::GetInstance()`
    `->SetWorldMaximumExtent(WorldExtent);`
  - Call must be done **before** defining any geometrical component and can be done only **once**
- An absolute small value (`10E-9 mm`) of accuracy may be redundant and inefficient for use on simulation of detectors of big size or macroscopic dimensions

- `G4GeometryTolerance` class holds the values for: `kCarTolerance`, `KRadTolerance` and `kAngTolerance`

Checking overlaps at construction

- "Overlaps" in the geometry can be detected at volume positioning stage
  - Boolean flag to be activated in the physical-volume constructor
    - Valid for placements and parameterised volumes
  - With explicit call to *CheckOverlaps()* through the pointer of the physical-volume
- Resolution of the check can be tuned
  - 1000 points generated on each solid surface by default
  - Positioning of each point is checked against volumes already positioned
  - Points detected *inside* existing positioned volumes at same level are flagged as overlap
  - Points detected *outside* current mother volume are flagged as overlap
  - Points of other volumes at the same level detected *inside* the volume being positioned are detected as overlap
- By default, precision of overlaps determined by the surface tolerance
  - Surface tolerances can be computed relative to the geometry topology
  - Optional: a custom tolerance for the check can be specified

Feature extended since the first version released in Geant4 8.0

Error propagation module **Geant4e**

since 4.9

Integration of Geant4e , error propagation module
for Track Reconstruction by Pedro Arce (CIEMAT)

- Details in Pedro's talks at Geant4 Workshop (Saturday) and at CHEP06 :

  http://indico.cern.ch/contributionDisplay.py?contribId=85&sessionId3&confId=048

- New Classes added in Geometry/Tracking and Navigation

  Magnetic field : G4ErrorMag_UsualEdRhs

  Management : G4ErrorTarget, G4ErrorSurfaceTarget,
  G4ErrorTanPlaneTarget, G4ErrorPlaneSurfaceTarget
  G4ErrorCylSurfaceTarget

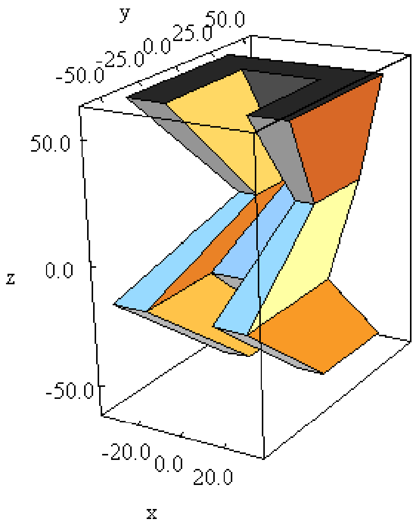
  Navigation : G4ErrorPropagationNavigator

  Global : G4ErrorPropagatorData

- Examples/extended : errorpropagation

New solids : G4ExtrudedSolid and G4Paraboloid

Since 8.3

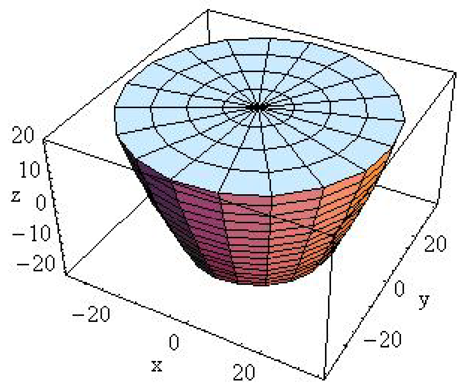


G4ExtrudedSolid by
Ivana Hrivnacova (IPN,Orsay,France)

G4ExtrudedSolid represents the extrusion Z direction of an arbitrary polygon

- For each Z section position with offset and scale factor is defined.
- The solid is implemented as a specification of G4TessellatedSolid

Scheduled for 4.9.1



G4Paraboloid by
Lukas Lindroos (Summer Student)

G4Paraboloid is a full Paraboloid with possible cut in Z

- Equation : $z = a*r^2+b$
- On the picture:
  G4Paraboloid(Name,dz,R1,R2)
  dZ = 20 and
  R1 = 20 (at z=-20)
  R2 = 35 (at z=20)

Fixes and improvements in Geometry

- **Solids**

  Revision of G4EllipticalCone : visualization, DistanceToIn(),
  DistanceToOut(),tests

  Fixes in G4Polycone, G4Polyhedra, G4Tubs and G4Cons, G4TwistedTrap

- **Overlaps Checking**

  CheckOverlaps() was improved for G4Polycone and G4Polyhedra

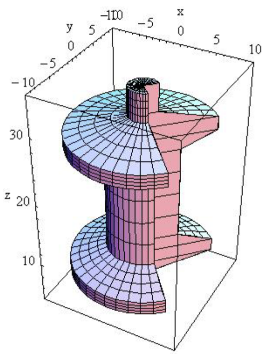
  Other Overlapping Facility : recursive_test was improved :

  Problem #784 closed: recursive_test causes a segmentation fault

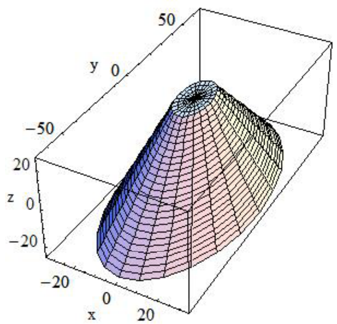
  Recursive_test was improved for solids with opening in Phi

- **Fixes in Geometry/Magnetic Field**

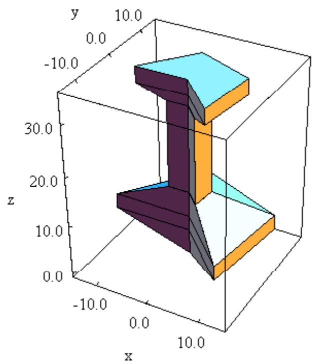
  Fix for NaN detected in G4MagIntegratorDriver (since 4.8.3)



G4Polycone



G4EllipticalCone



G4Polyhedra

In Progress

- **G4Torus** (NaN in CalculateExtend() reported by FNAL team)

To do

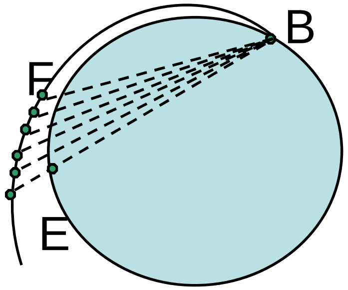
- **Continue Test Suite for solids**
  BREPS
  BREPS with Boolean operations

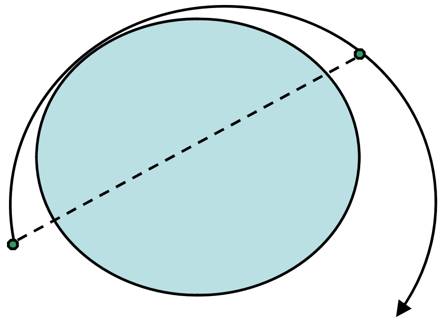
- **Code review of solids**
  CPU performance

G4PropagatorInField : Improved LocateIntersectionPoint()

<u>Motivation:</u> In ATLAS simulation rarely Intersection wasn't found after **10,000 trials** and events were aborted

<u>Examples</u> of some 'difficult' cases for finding intersection: particle moving around Cylinder with almost the same Radius





Since 4.8.2 version :

Add to existing algorithm for Location of Intersection

- check on 'slow' progress in finding Intersection after 10 trials
- in case of to 'slow' progress add faster algorithm.

  This algorithm consists in **recursive** division of the Full Chord in two

Good improvement for **CPU performance** for 'difficult ' intersection:
needs **5*times less** trials to find intersection
for particle moving around Cylinder with almost the same Radius

In progress :

Others algorithms for intersection were tested:
- Second order Brent's method, method using SurfaceNormal()
- They are ready to be included

Steppers in Geant4

- **Fix G4RKG3Stepper or Nystrom Stepper** re-enabling use **<u>since 4.8.3</u>**
  Uses 3 calls to field (rather than 4 of G4ClassicalRK) per integration step

- **Revision of Helical Steppers**
  - Improved calculation of DistChord() for Stepping Angle > π
  - CPU performance : *G4HelicalExplicitEuler made **25-30 %** faster*

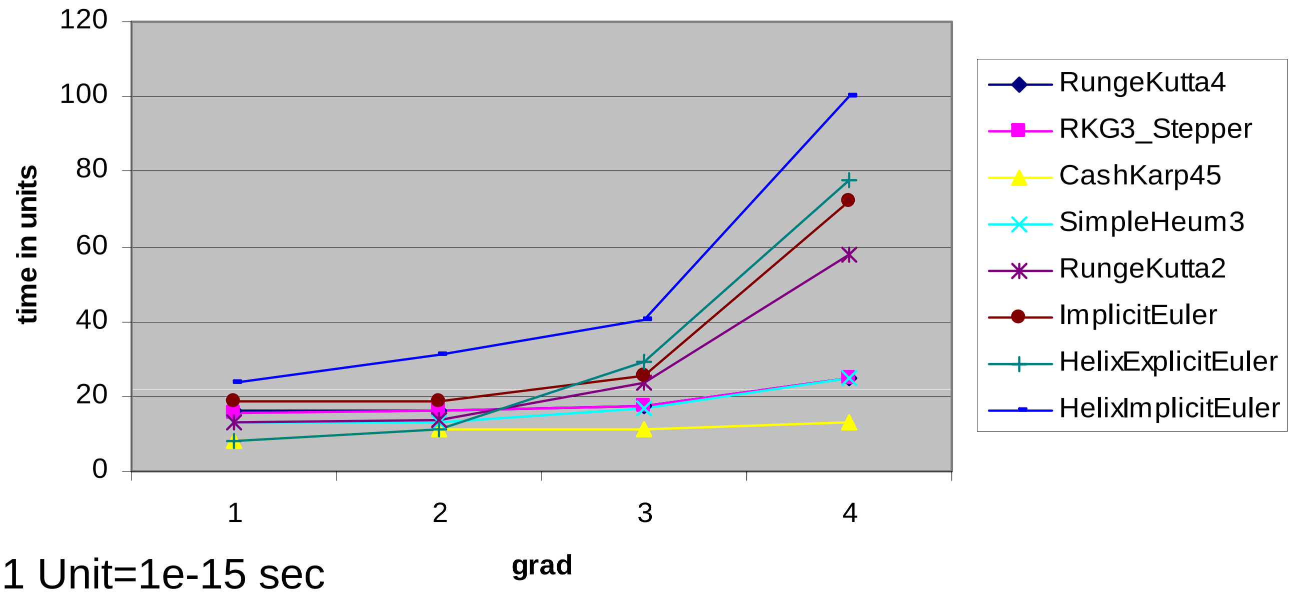
- **New class G4HelixMixedStepper is added since <u>4.9.0 version</u>**
  For small steps uses G4ClassicalRK4(default Stepper)
  For long steps uses G4HelicalExplicitEuler Stepper
  Change of Stepper when Stepping Angle > 0.33 π (fixed value)
- **Timing and accuracy in non-uniform fields**

Table. Timing Steppers

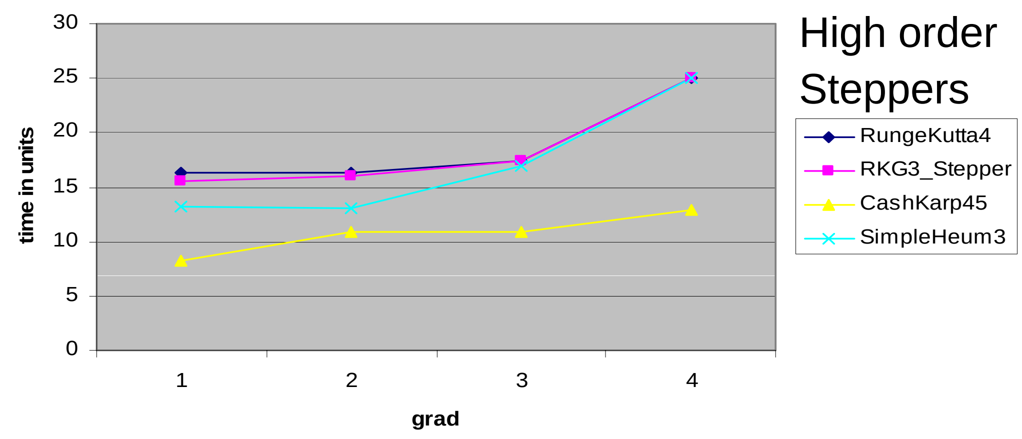
Stepper	Order	Time for One Step Unit=1e-15 sec	Time for Accurate Advance 120 mm in Quadripole Field : Bx=Grad*x; By=-Grad*y Gradient=1 T/m , Epsilon=1e-5
RungeKutta4	4^{th}	4.58 units	**17.4 units**
RKG3_Stepper	4^{th}	4.33	**17.4**
CashKarp45	5^{th}	**2.58**	**12.9**
SimpleHeum	3^{th}	3.16	**17.0**
RungeKutta2	2^{th}	**2.08**	23.83
ExactHelixStepper (revised)		**1.58**	Only Uniform Field
ExactHelixStepper (old)		1.7	Only Uniform Field
HelixExplicitEuler (revised)	1^{th}	**2.7**	28.92
HelixExplcitEuler(old)	1^{th}	3.5	Old version
HelixImplicitEuler	2^{th}	6.9	40.66
ExplicitEuler	1^{th}	**1.08**	673
ImplicitEuler	2^{th}	**2.16**	25.7

Low order steppers and CashKarp45 are faster **per step**
But high order steppers are more precise and need less steps **per trajectory**

Timing for AccurateAdvance in Uniform Field and in Quadripole Field



1 Unit=1e-15 sec



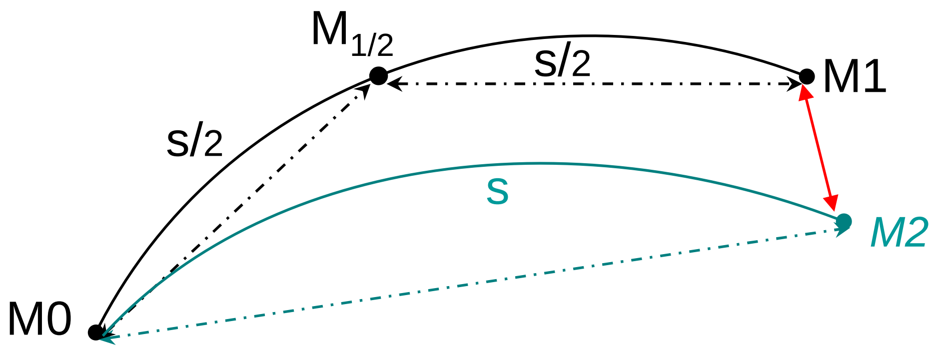
1 Uniform Field (2T)
Uniform Field (2T)+
Quadripole Field : Bx=Grad*x;
By=-Grad*y
2 Grad=0.1 T/m
3 Grad=1 T/m
4 Grad=10 T/m

- CashKarp45 is the fastest
- In UniformField or in Fields with small gradient HelixExplicitEuler is fast

Accuracy for Steppers in Uniform Field and in Quadripole Field

‘Good’ Stepper = large S , small error ($\Delta R, \Delta P$)
Estimated Error/Actual Error < 1

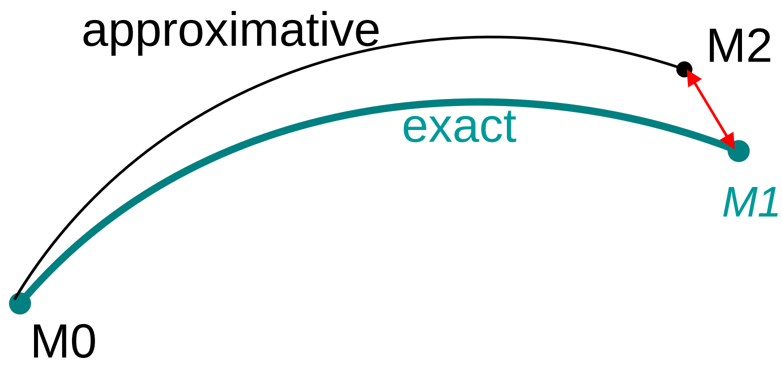
Calculated Error ($\Delta R, \Delta P$) for One Step (R=Position, P=Momentum)



M1 = Two half steps for Stepper

M2 = One full step for Stepper

Difference in Position $\delta R = |R(M2)-R(M1)|$ and in Momentum $\delta P = |P(M2)-P(M1)|$ for One Step

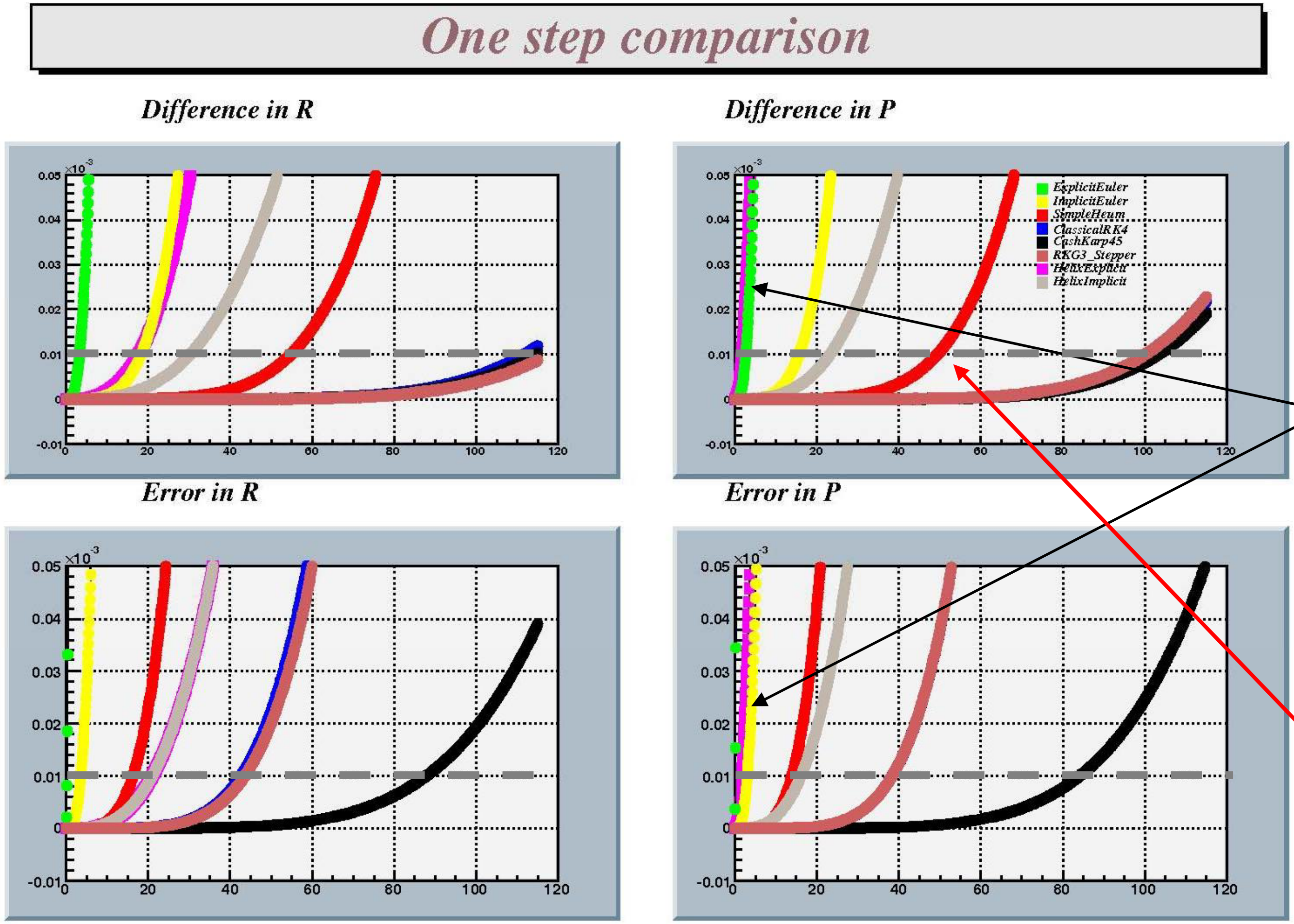


M2 = One Step for Stepper

M1 = ‘Exact’ Solution :
- ‘ExactHelix’ for Uniform Field,
- ‘AccurateAdvance’ with very small step for Quadripole Field

How long can be One Step ?

Example 1 : UniformField (2T) + QuadripoleField : Bx=grad*x,By=-grad*y
Grad = 0.1 T/m , particle starting with Radius=0.4511 m



In Field with
Small Gradient :

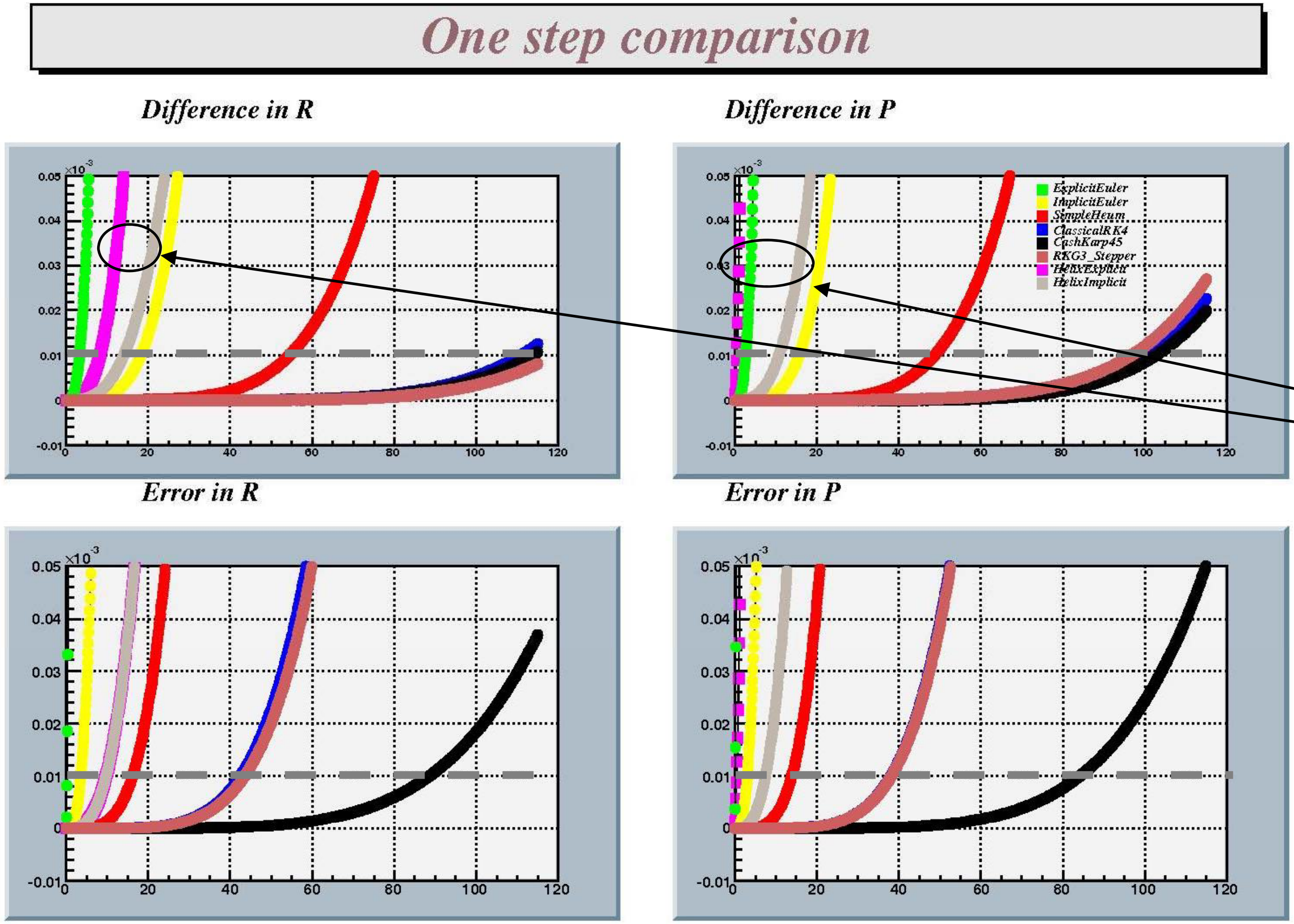
HelixExplicit has
Big error, specially in P

SimpleHeum has
Small Error in P

The error, calculated by Steppers, is an Over Estimation (it has to be)

How long can be One Step ?

Example 2 : UniformField (2T) + QuadripoleField : Bx=grad*x,By=-grad*y
grad = 1 T/m , particle starting with Radius=0.4511 m

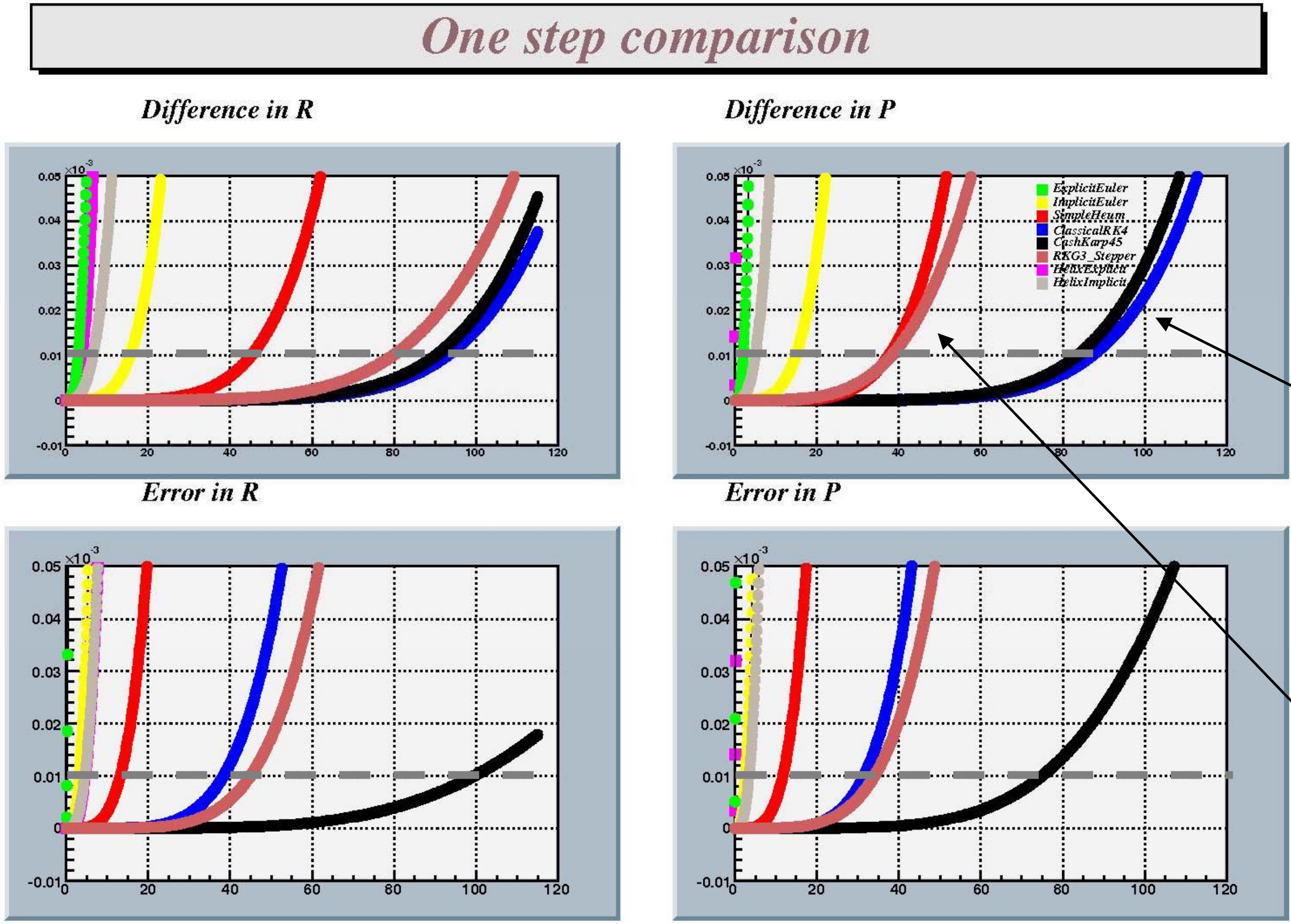


In Field with
Medium Gradient :

Helical Steppers have
Big Error in R and P

How long can be One Step ?

Example 3 : UniformField (2T) + QuadripoleField : Bx=grad*x,By=-grad*y
grad = 10 T/m , particle starting with Radius=0.4511 m



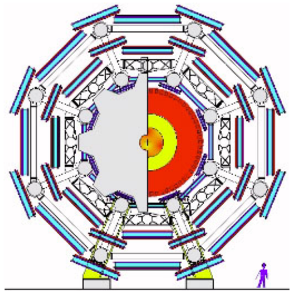
In Field with
Strong Gradient :

ClassicalRK4 has Smallest difference, But not Smallest Error

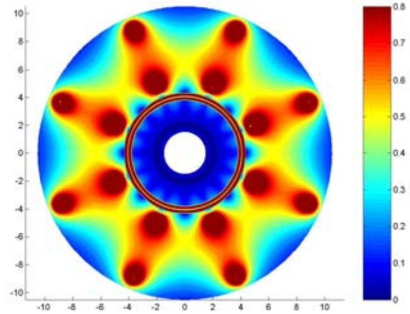
RKG3_Stepper is Comparable to SimpleHeum (3th order)

Pure tracking in ATLAS detector Magnetic Field

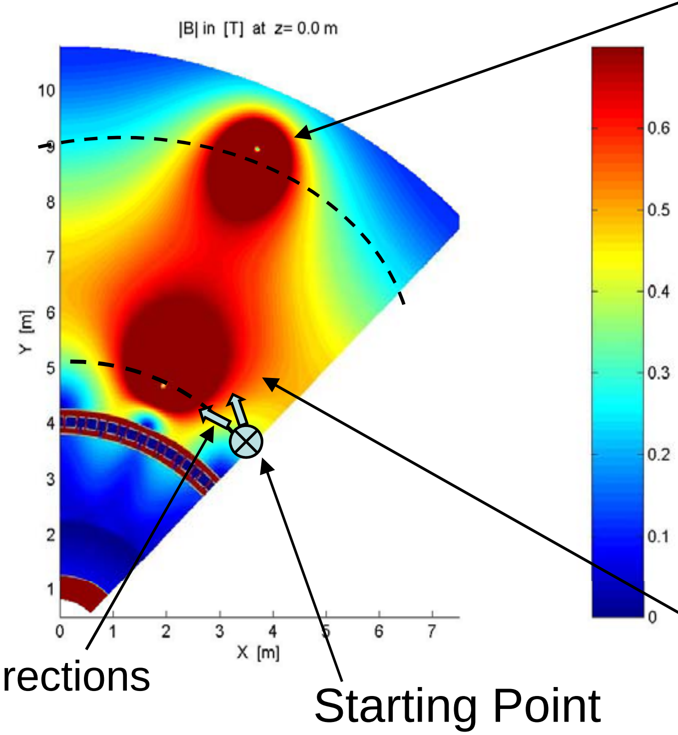
ATLAS detector



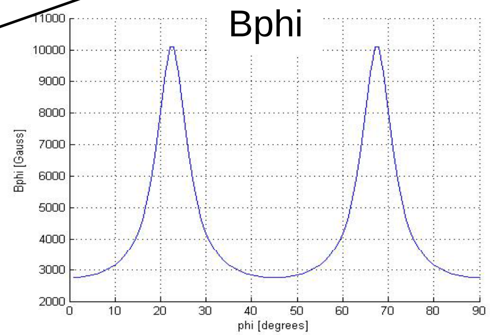
ATLAS Field using ATLM

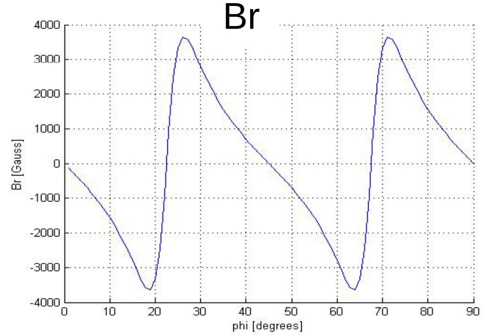


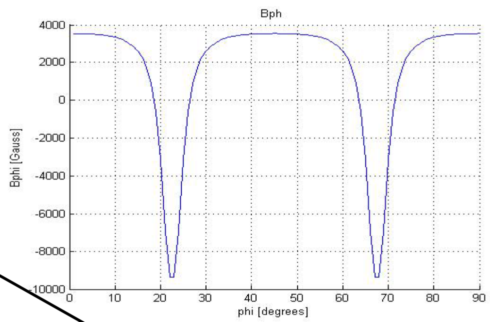
Toroid Field Profile

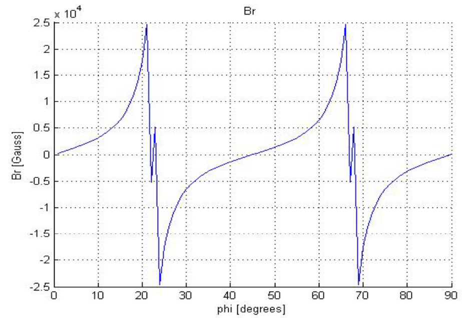


R=9 m, Z=0 (center of detector)









R=5 m, Z=0 (center of detector)

<u>Example :</u> Accuracy for muons in high non-homogeneous Field

(Exact Calculation of Field for each call ;only tracking, no physics, epsilon=1.e-5)

Test consists to shoot the muons of different energy in direction of the Toroid coil.
Store values of Maximum Field and Minimum Radius of projected Helix 'seen' by muons

Stepper	**Energy 0.1 GeV** 70 m of track Bmax = 1.6 T Rmin = 0.2 m	**Energy 1 GeV** 13 m of track Bmax = 2.2 T Rmin= 0.24 m	**Energy 10 GeV** 11 m of track Bmax = 2.2 T Rmin = 0.8 m
CassicalRK4	0.003 mm	0.0002 mm	0.000018 mm
RKG3_Stepper	0.004 mm	0.0008 mm	0.00002 mm
CashKarpRK45	0.006 mm	0.0025 mm	0.00013 mm
SimpleHeum	0.009 mm	0.002 mm	0.00025 mm
SimpleRunge	0.02 mm	0.004 mm	0.0006 mm
HelixExplicitEuler	200 mm	9 mm	0.0043 mm
HelixImplicitEuler	7 mm	0.011 mm	0.00096 mm

- Helical Steppers are not precise in ATLAS field
- ClassicalRK4 has very good accuracy
- CashKarpRK45 and RKG3_Stepper are less precise that ClassicalRK4
- SimpleHeum can be a good alternative (fast and good error)

In progress : Tests with Geometry and not-Uniform Field

1)LarCalorimeter example with " Exact " **ATLAS** Solenoid Field or Toroid Field (not a Field Map)

- For intersection studies and accuracy
- First Test : Accuracy of propagation

  Shooting muons (Different Energies, Different Angles)

  Parameters of Propagation as in ATLAS

  (DeltaIntersection=0.00001 mm, DeltaOneStep=0.0001 mm, MaxEpsilon=0.001, MinEpsilon=0.00001)

  Difference between full and empty geometry in order of **1.e-5 mm for 3 m** of track

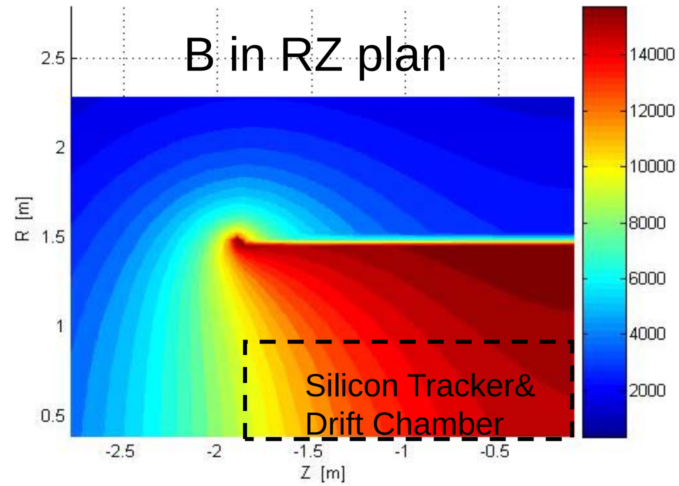
2)NTST test with Uniform, Quadripole Field and "Tabulated" Solenoid Field

Semi Analytic formula for BaBar Solenoid : 1.5 T, Rext=1.5 m, Z half= 1.9 m

-BaBar Silicon Tracker and 40 layer Drift Chamber

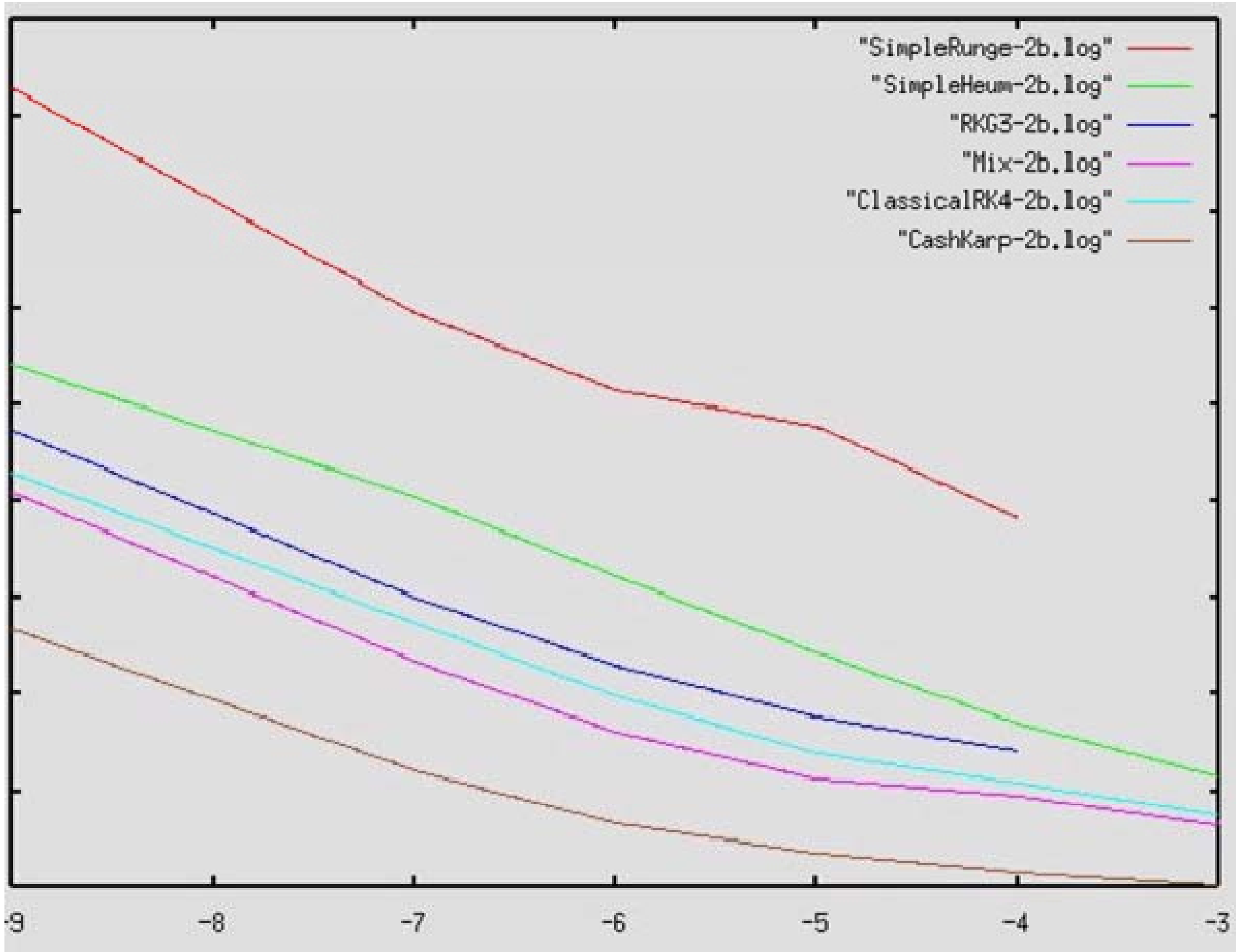
-B-Bbar events

- -For CPU benchmarks and accuracy
- -For Testing Steppers



Some first results

Time for different steppers in Uniform Field
run-2b.mac= 1000 events, cutEmin=1 MeV



Asked Accuracy (Relative Error, power)

Conclusions

-The error, calculated by stepper, is an Over Estimation(good)
-SimpleHeum has small error in momentum
-In Fields with small gradients Helical Steppers are faster but can be not precise
-In Fields with strong gradients Better to use ClassicalRK4
CashKarpRK45 and RKG3_Stepper are less precise

In Progress

Tests with NTST test and LarCalorimeter example

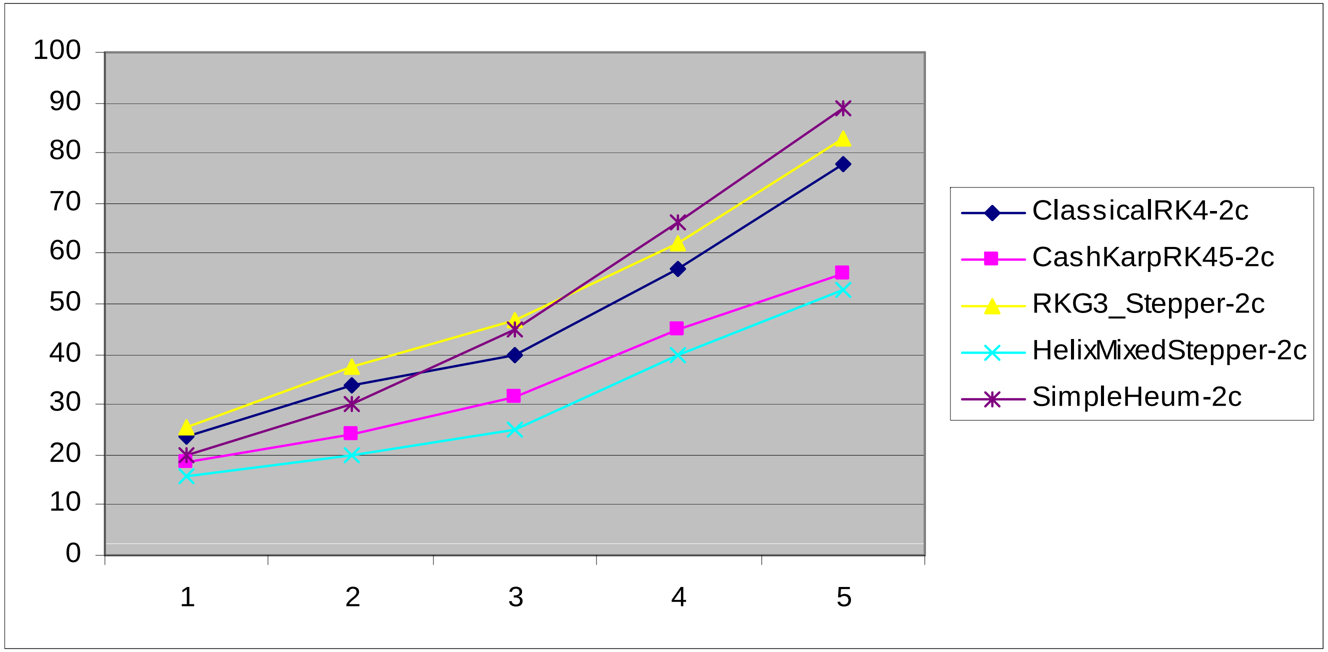
To do

- New test case with full detector geometry, CMS GDML, and non-uniform Field
- New stepper depending on field variations by region
- Review of tracking parameters
(user gives desired accuracy, algorithm decides Stepper and his parameters)

- Field map study and construction of efficient Field map
(not just linear interpolation)
- Would be very instructive to make some tests with ATLAS realistic geometry
and realistic field (ATLM is included in ATHENA already)

More results for NTST test with Gradient Field

Time per event for different steppers in Uniform +Gradient Field
run-2c.mac= 1000 events, no Energy Cut by NTST as for run-2b.mac



Gradients: 0.005,0.01,0.05 and 0.1 T/m