Alternative integration methods and utilization of dense output for field propagation

Dmitry Sorokin (GSoC 2016),

John Apostolakis (CERN)

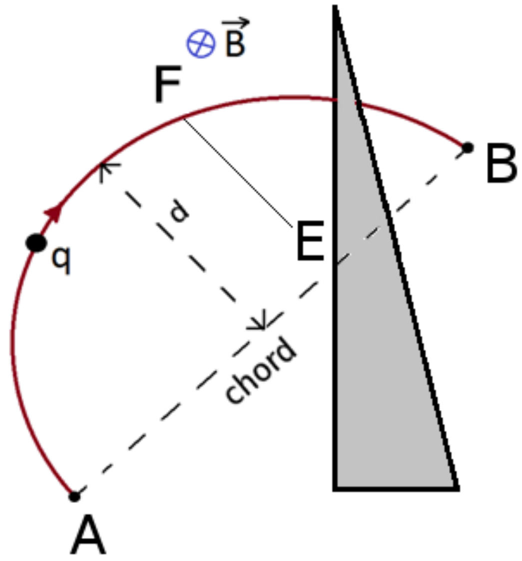
Somnath Banerjee (GSoC 2015) & Jason Suagee (GSoC 2015),

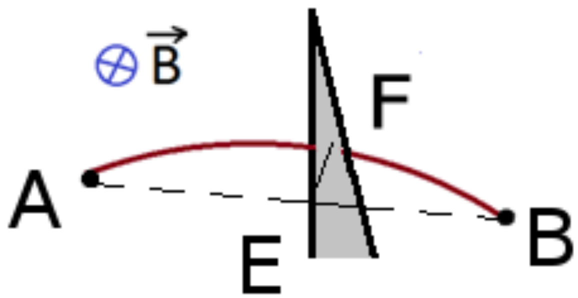
Outline

- Overview of field propagation in Geant4
- New methods in 10.3-beta
  - Integrated several 2015 developments
- New developments
  - The Bulirsch-Stoer method
  - Using interpolation to speed up intersection with volume boundary
  - Verification

Field propagation in Geant4

- Uses Runge-Kutta integration with adaptive step size control
  1. d < deltaChord
  2. ΔB < deltaOneStep
- Locates boundary intersection using iterative intersections of chords



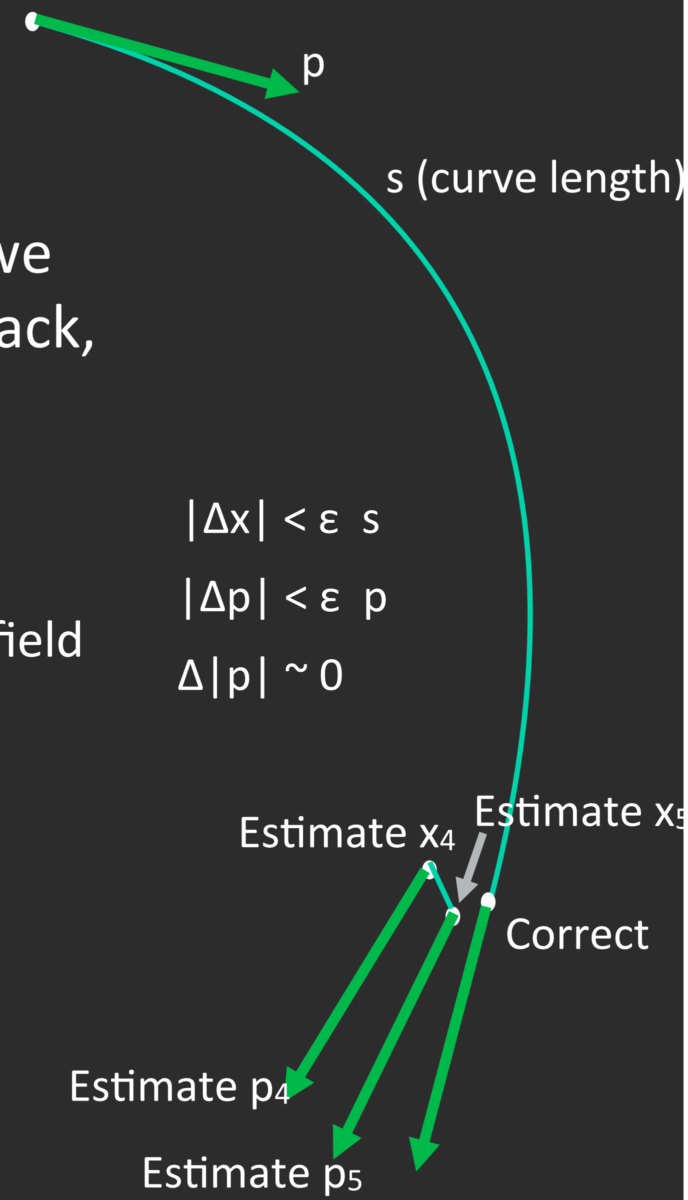


Until |F – E| < deltaIntersection

TEXT

Integrating efficiently

- Given a detector’s field B(x,y,z) [or B+E] we need to integrate the trajectory of each track, taking care
  - to stay within a relative accuracy ε
  - to be fast - using as few calls as possible to the field method
- Typically choose Runge-Kutta methods
  - No history. Ability to adjust step size



TEXT

Embedded Runge-Kutta methods

$f_1 = F(x_0, \; y_0)$

$f_2 = F(\; x_0 + a_2\, h \;, \;\; y_0 + h\, b_{21}\, f_j \;)$

$f_3 = F(\; x_0 + a_3\, h \;, \;\; y_0 + h\, b_{31}\, f_1 + h\, b_{32}\, f_2$

a_i $\quad b_{ij}$

0			
$\frac{1}{2}$	$\frac{1}{2}$		
$\frac{3}{4}$	0	$\frac{3}{4}$	
1	$\frac{2}{9}$	$\frac{1}{3}$	$\frac{4}{9}$

c_j

- "Integrate" $dy/dx = F(x, y)$ from x_0 to x_0+h
- Uses evaluations of $F(x, y)$:
  - $f_i = F(\; x_0 + a_i\, h \;, \;\; y_0 + h\, \Sigma_{j<i}\, b_{ij}\, f_j \;)$
  - $y_{estim}(x_0 + h) = \Sigma_i\, c_i\, f_i$

$y_{RBS3} = 2f_1/9 + f_2/3 + 4f_3/9$

- Each method has its 'tableau' of a_i, b_{ij}, c_i
- Key Parameters of an RK method:
  - Number of 'stages' = number of evaluations of f()
  - 'Order' = the expected scaling of the error ~ h^N
  - Embedded method = 2nd 'line' to estimate error

c'_j

$\frac{7}{24}$	$\frac{1}{4}$	$\frac{1}{3}$	$\frac{1}{8}$

$y'\,(x_0 + h) = \Sigma_i\, c'_i\, f_i$

$\Delta y = \Sigma_i\, (c'_i - c_i)\, f_i$

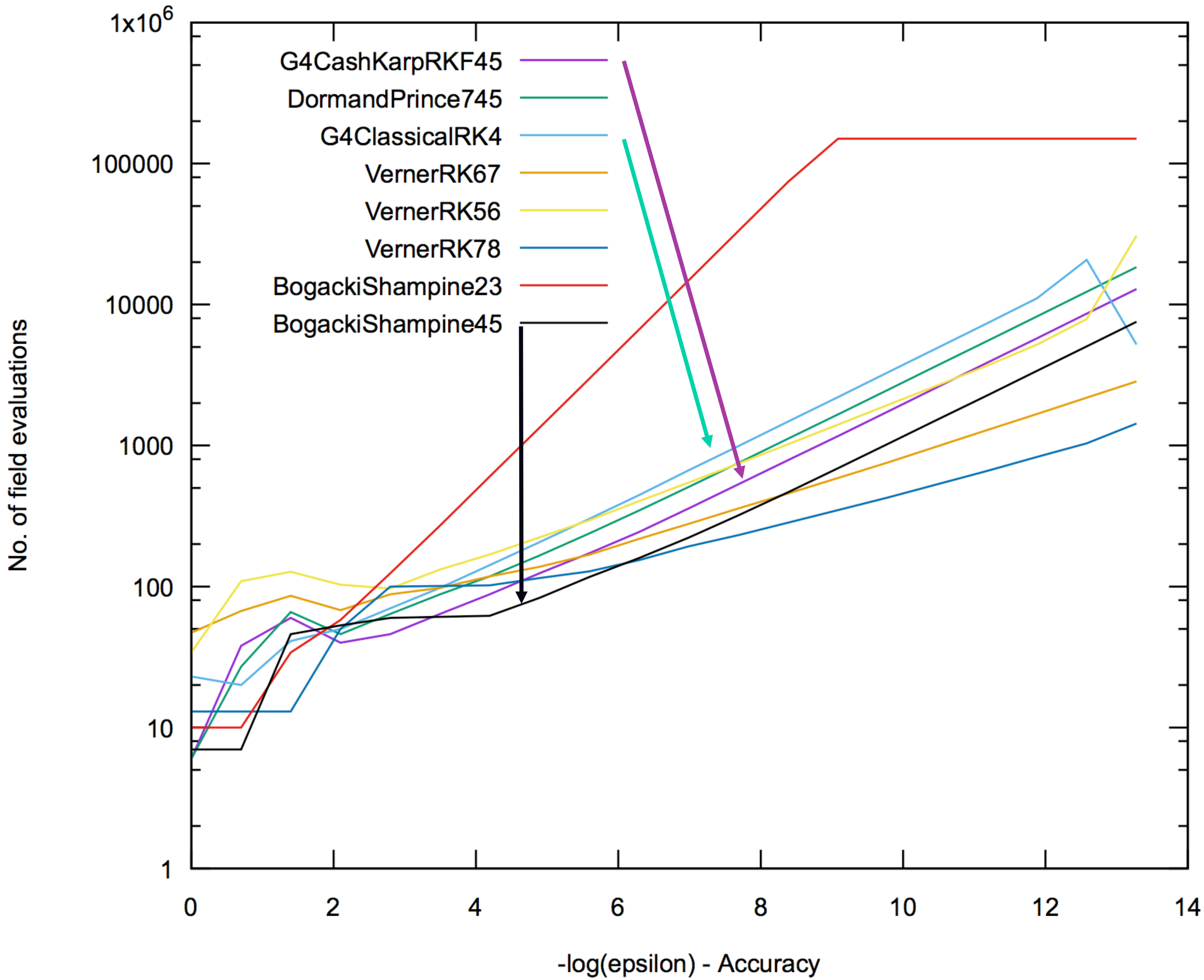
TEXT

Geant4 steppers

- ‘Classical RK4’ = original 4th order method, 4 stages
  - Error estimate from breaking step in two => 11 eval.
- Embedded methods provides built-in error estimate
  - Cash-Karp (1990) - uses difference of 5th & 4th order method, six stages = 6 evaluations of derivative
- AtlasRK4/NystromRK4: 3 field evaluations + evaluation of error using numerical estimate of 4th derivative - restricted to B-fields
- Helix - for constant field
- Lower order RK methods for short steps, and/or lower accuracy

SELECTED NEW METHODS

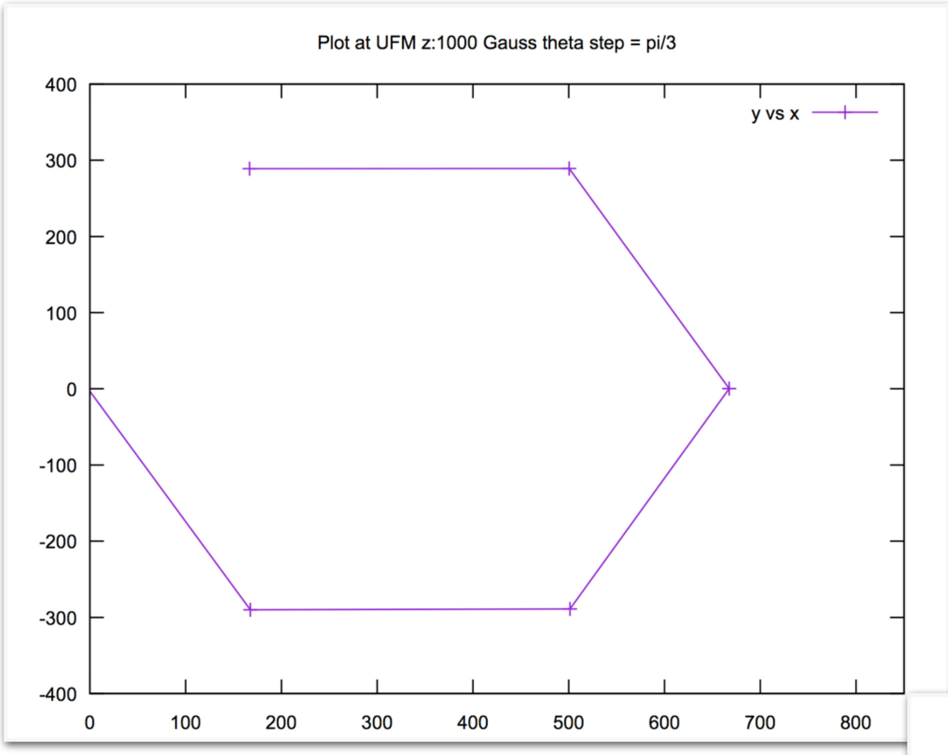
Name / Authors	Order	Stages	Error Estim.	#Evaluations Failed	#Evaluations Good	FSAL	Interpolation (Order)	Extra evaluations
Classical	4	4	N	11	11	No	No	-
CashKarp	5	6	Y	5	5	No	No	-
Dormant-Prince 5 "DoPri5"	5	7	Y	6	5	Yes	Yes - 2 ways (4/5)	0/2
Bogacki-Shampine45	5	8	Y	7	6	Yes	Yes	2
Dormand-Prince8	8	13	Y	12	11	No	No	
Verner78 'efficent'	8	13	Y	12	12	No	Yes - 2 (7 / 8)	4/8



G4CashKarpRKF45
DormandPrince745
G4ClassicalRK4
VernerRK67
VernerRK56
VernerRK78
BogackiShampine23
BogackiShampine45
1x10^6
100000
10000
1000
100
10
1
0
2
4
6
8
10
12
14
No. of field evaluations
-log(epsilon) - Accuracy

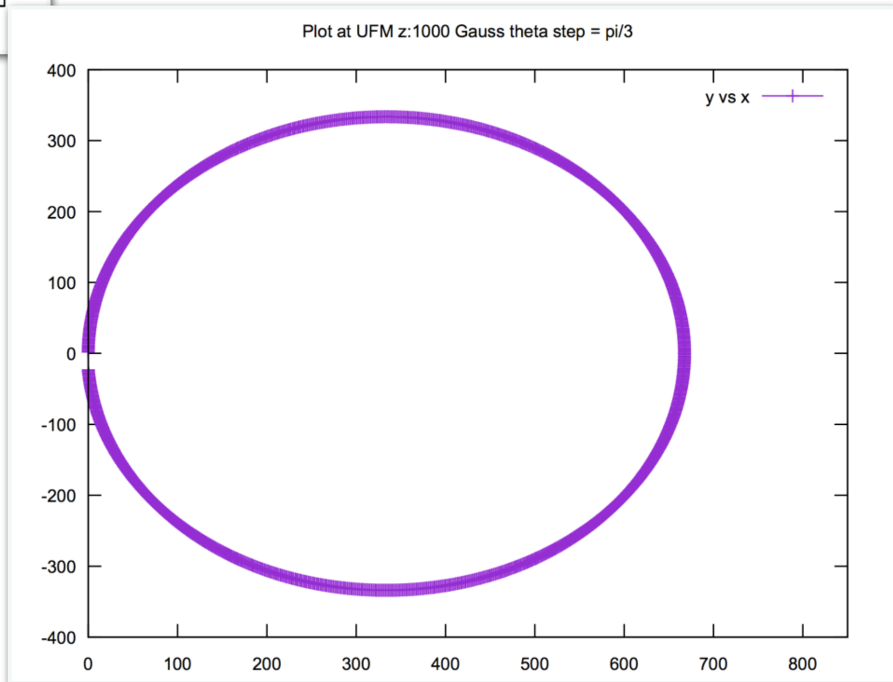
Integration of new steppers

- Now integrated in Geant4 10.3 beta:
  - BogackiShampine 2/3 and 4/5
  - DormandPrince 4/5 (7 stages = evaluations)
- A few RK tableaus not integrated
  - Expect 'very high' (>6) order Verner methods
- FSAL & Interpolation available – but not used
  - Full 'move' to using FSAL 'simple' (simpler)
- Issue with interpolation of DormandPrince 4/5
  - Fixed recently



6 computed points in a circular trajectory

Calling 100 times between each pair of points, to give "Dense" output



Interpolated result

Alternative integration methods and utilisation of dense output for field propagation

by Dmitry Sorokin (MIPT, Moscow Russia)

Mentored by: John Apostolakis

GSoC 2016 Project Outline

- Goals
- The Bulirsch-Stoer method
- Alternative integration strategy for dense output methods
- Verification
  - Propagation in uniform magnetic field
  - NTST test

Project goals

- Implement the Bulisrch-Stoer method
- ***Extrapolation methods*** *are more efficient than the Runge-Kutta methods for smooth functions and large steps*
- *One of thea* ***most efficient*** *is the Bulirsch-Stoer method*
- Use interpolation to improve integration & volume intersection
- *Using* ***dense output*** *(interpolation) the solution can be evaluated for any point within the integration interval.*
- *For* ***a fixed number*** *of extra field/function* ***evaluations*** *and provides an estimate of the 'solution' for any number of intermediate points*
  - *versus a new set of 'N' evaluations for each intermediate point – the existing alternative (e.g. N=4 for classical RK4, N=6 for)*

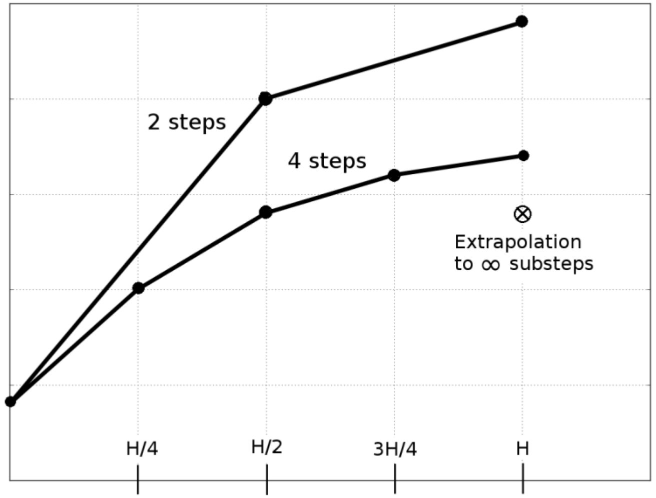
The Bulirsh-Stoer method

Idea:

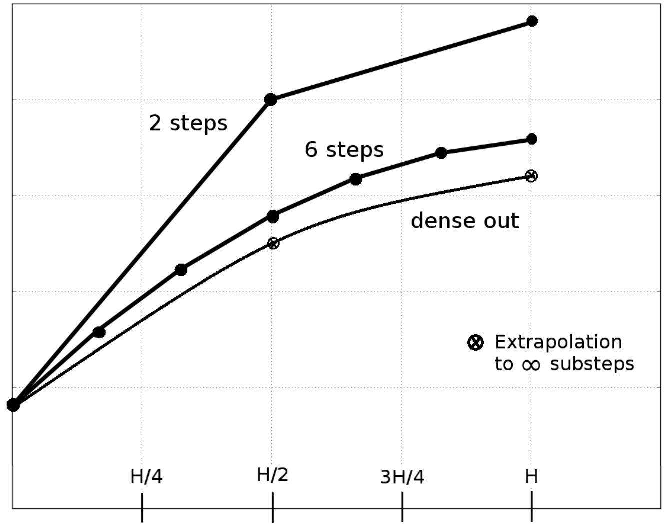
- Use midpoint method to estimate integral
- Vary number of intermediate points
- Approximate the integral using rational functions
- Extrapolate to n = ∞

Advantages:

- Step size and order control
- Very good for smooth problems and large steps
- Can provide interpolation / dense output



n = 2i+2, for i = 0,1,2….



n = 4i+2, for i = 0,1,2….

Alternative integration strategy for methods with dense output

Old strategy

- Make series of steps without error control to predict the step size (satisfying $d < \delta_{chord}$)
- Make a step with error control to improve the accuracy

  (Δ Endpoint $< \delta_{OneStep}$)
- If the chord intersects:

  make a series of substeps with error control to locate the intersection point

New strategy

- Make one step with error control
- Use dense output to divide the step to substeps (satisfying d < deltaChord)
- For each substep:

  if the chord intersects:

  Use dense output to locate the intersection point

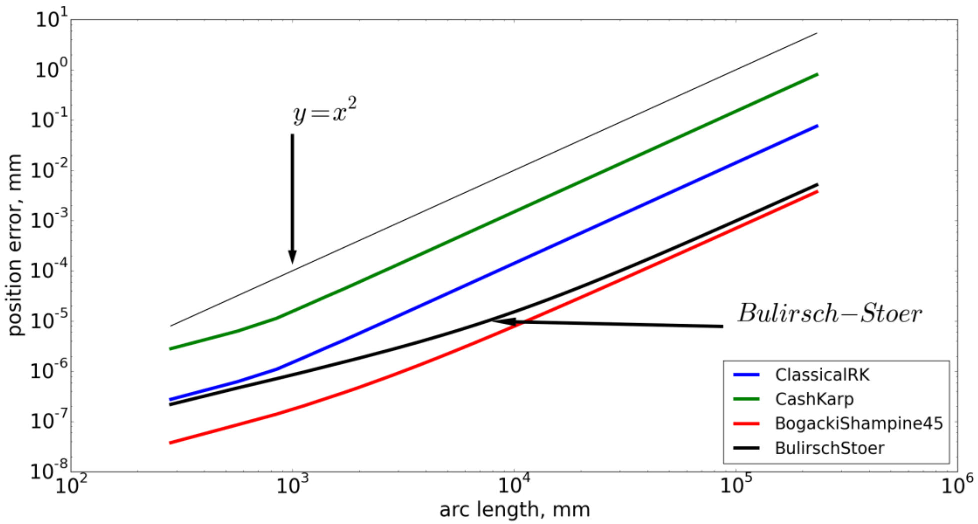
Pros: A lot fewer field evaluations required for large steps

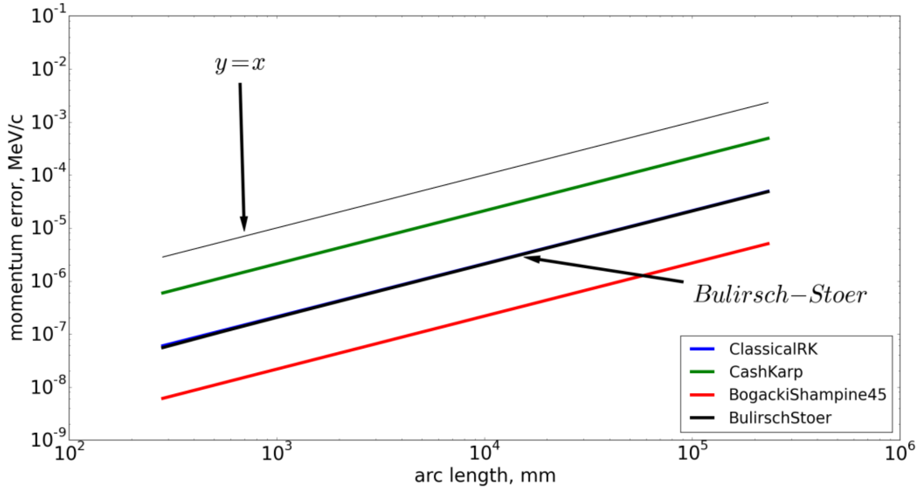
Cons: Dense output is less accurate than the solution

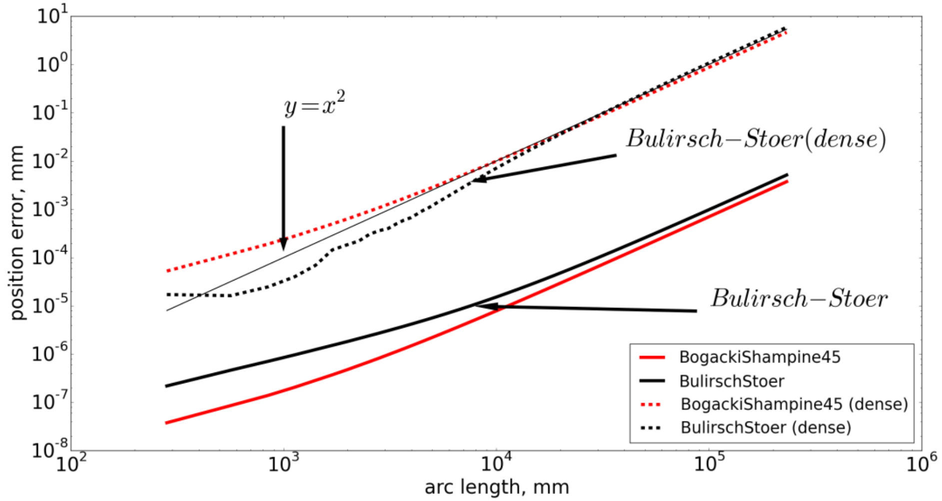
Propagation in uniform magnetic field

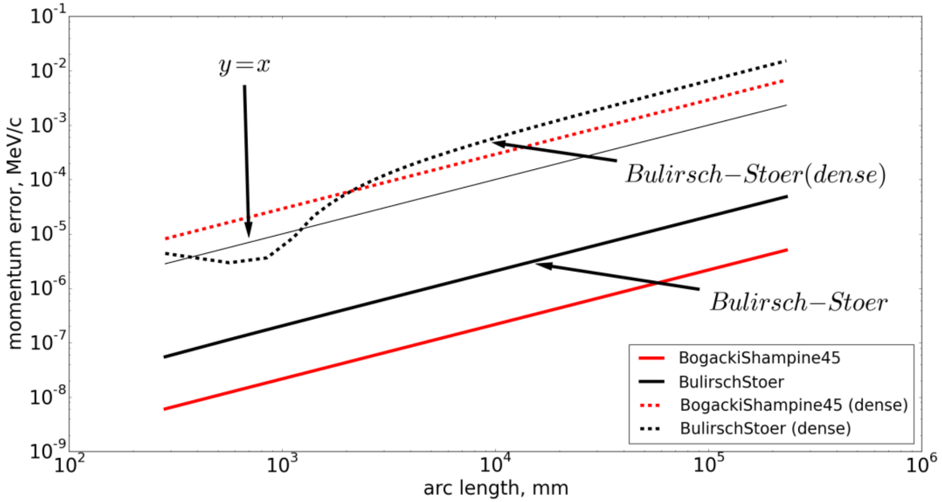
1 MeV proton in the uniform 1 tesla magnetic field. Radius of the circle in xz plane is 102.20 mm

Momentum is 43.33 MeV/c. deltaOneStep = 1e-4 mm, deltaIntersection = 1e-5 mm

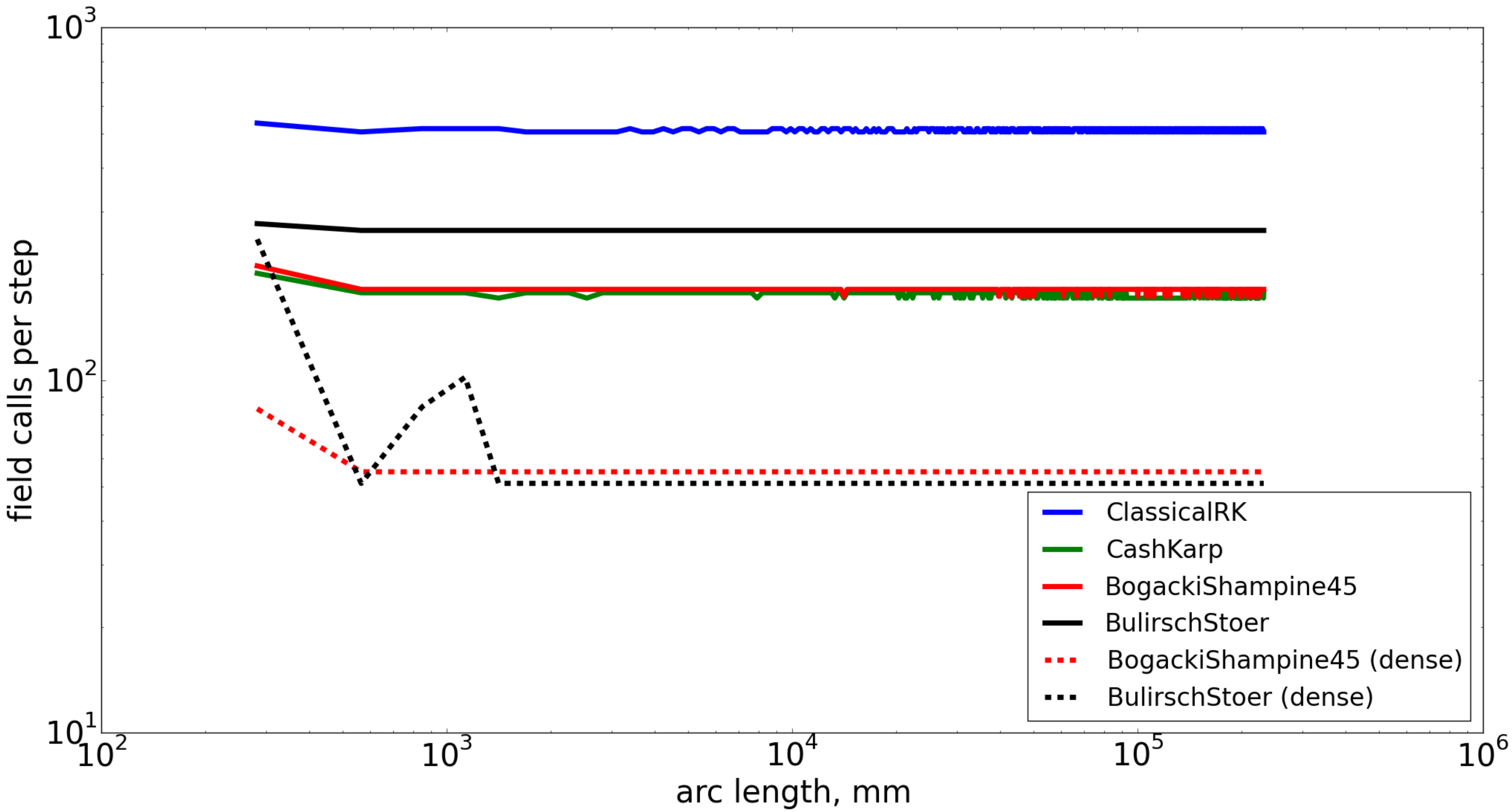




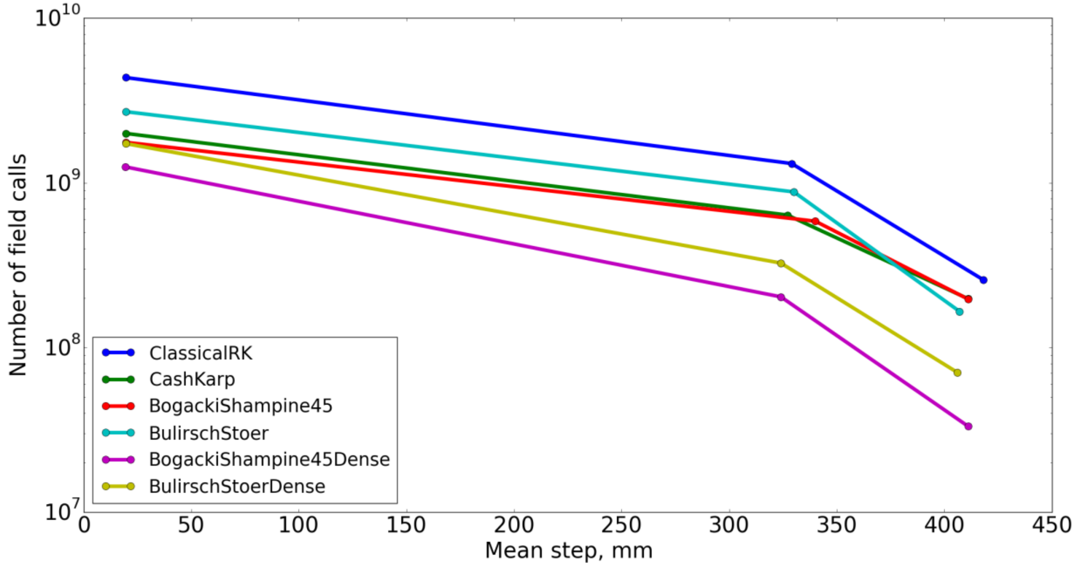


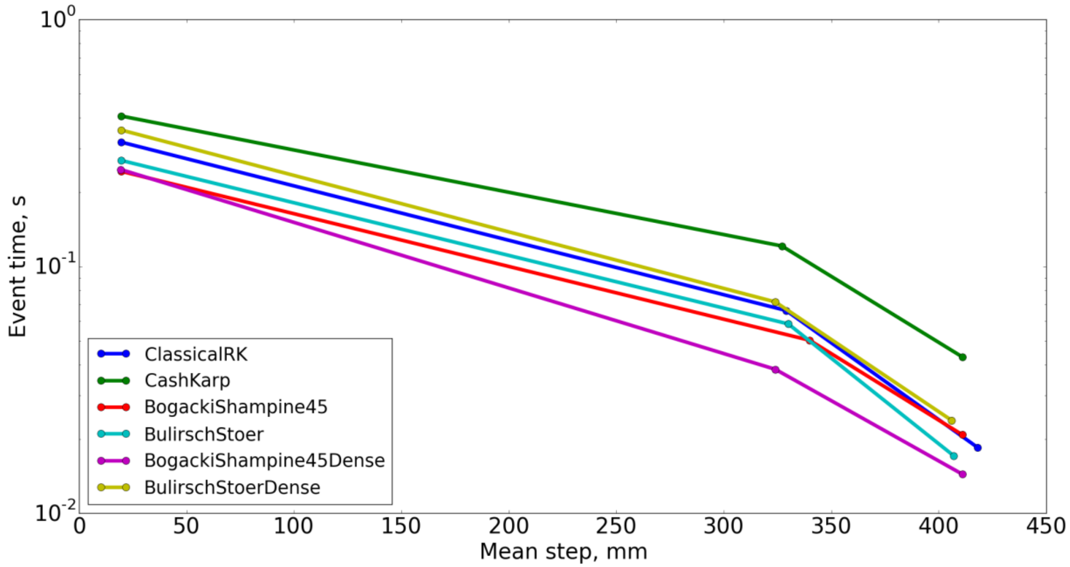


Propagation in uniform magnetic field



NTST test





Geant4 benchmark test. Simulates the BaBar silicon vertex tracker and 40-layer drift chamber.

File	range cut	looper cut	min Ecat
run2a.mac	1 mm	200 MeV	1 MeV
run2b.mac	1 mm	Not applied	1 MeV
run2c.mac	1 mm	Not applied	Not applied

Thank you

FROM 'RANDOM' NEXT POINT TO INTERSECTION

- Found endpoint of integration C: $\vec{x}_C, \vec{p}_C$
- How to find the intersection point E of the curve with the next volume boundary?
- Assume we already have or calculate intermediate points B_3 & B_6 also on curve
- First identify an approximate intersection - in this case D_3
- Then refine it to be 'close enough' to curve - as close as possible to true intersection E