Field precision
and performance

John Apostolakis

representing the effort of
the Geometry & Transportation WG
of Geant4

Version 0.41

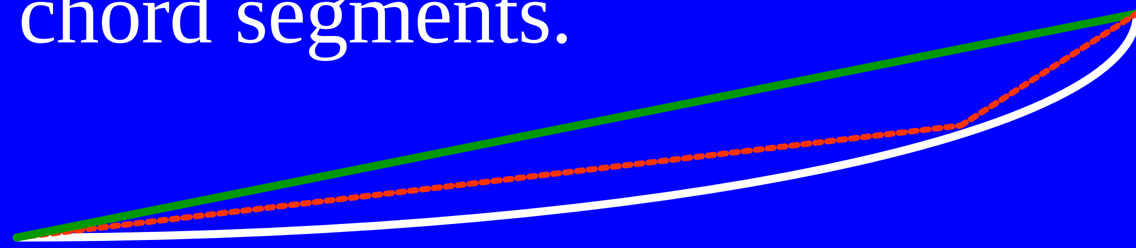
12th November 2002

Magnetic field: overview

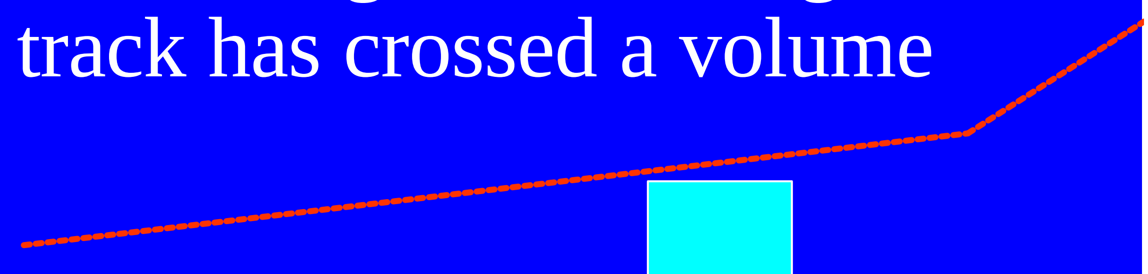
- In order to propagate a particle inside a field (e.g. magnetic, electric or both), we solve the equation of motion of the particle in the field.
- We use a Runge-Kutta method for the integration of the ordinary differential equations of motion.
  - Several Runge-Kutta 'steppers' are available.
- In specific cases other solvers can also be used:
  - In a uniform field, using the analytical solution.
  - In a nearly uniform field (BgsTransportation/future)
  - In a smooth but varying field, with new RK+helix.

Magnetic field: overview (cont)

- Using the method to calculate the track's motion in a field, Geant4 breaks up this curved path into linear chord segments.

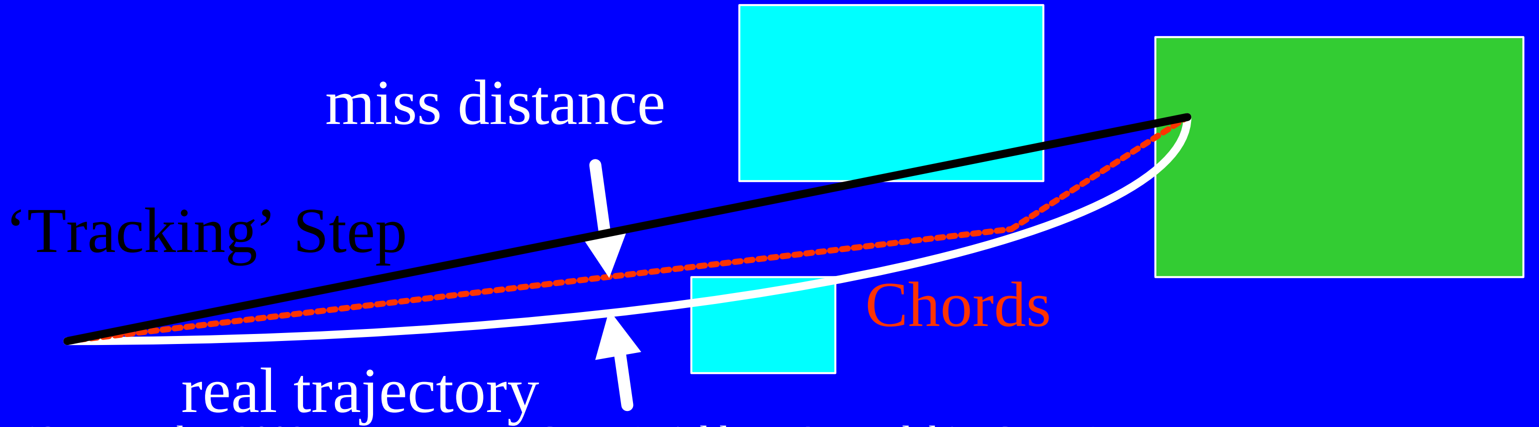


- We determine the chord segments so that they closely approximate the curved path.
- We use the chords to interrogate the Navigator, to see whether the track has crossed a volume boundary.



Stepping and accuracy

- You can set the accuracy of the volume intersection,
  - by setting a parameter called the “miss distance”
    - it is a measure of the error in whether the approximate track intersects a volume.
    - Default “miss distance” is 3 mm.
- One physics/tracking step can create several chords.
  - In some cases, one step consists of several helix turns.



Global and local fields

- One field manager is associated with the 'world'
  - Set in G4TransportationManager
- Other volumes can override this
  - By associating a field manager with any logical volume
    - By default this is propagated to all its daughter volumes

```
G4FieldManager* localFieldMgr=
   new G4FieldManager(magField);
logVolume->setFieldManager(localFieldMgr,
  true);
```

where '`true`' makes it push the field to all the volumes it contains.

Precision parameters

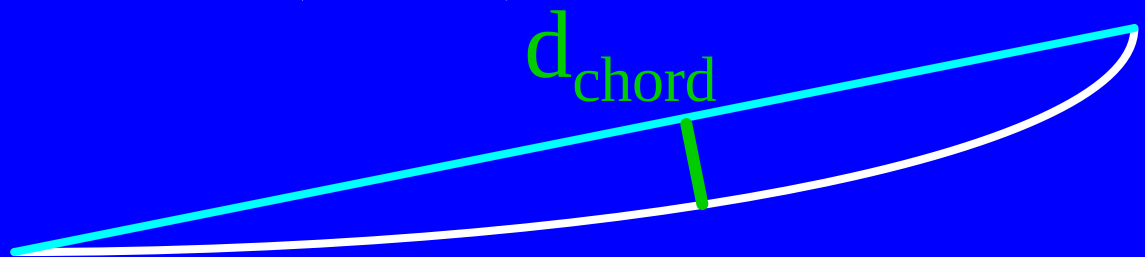
- Errors come from
  - Break-up of curved trajectory into linear chords
  - Numerical integration of equation of motion
    - or potential approximation of the path,
  - Intersection of path with volume boundary.
- Precision parameters enable the user to limit these errors and control performance.
  - The following slides attempt to explain these parameters and their effects.

Volume miss error

Due to the approximation of the curved path by linear sections (chords)

$d_{chord} < \delta_{chord}$

Parameter



| δ_{chord} value

- Parameter δ_{chord}
- Effect of this parameter as $\delta_{chord} \rightarrow 0$

$$s_{1step}^{propagator} \simeq (8\, \delta_{chord}\, R_{curv})^{1/2}$$

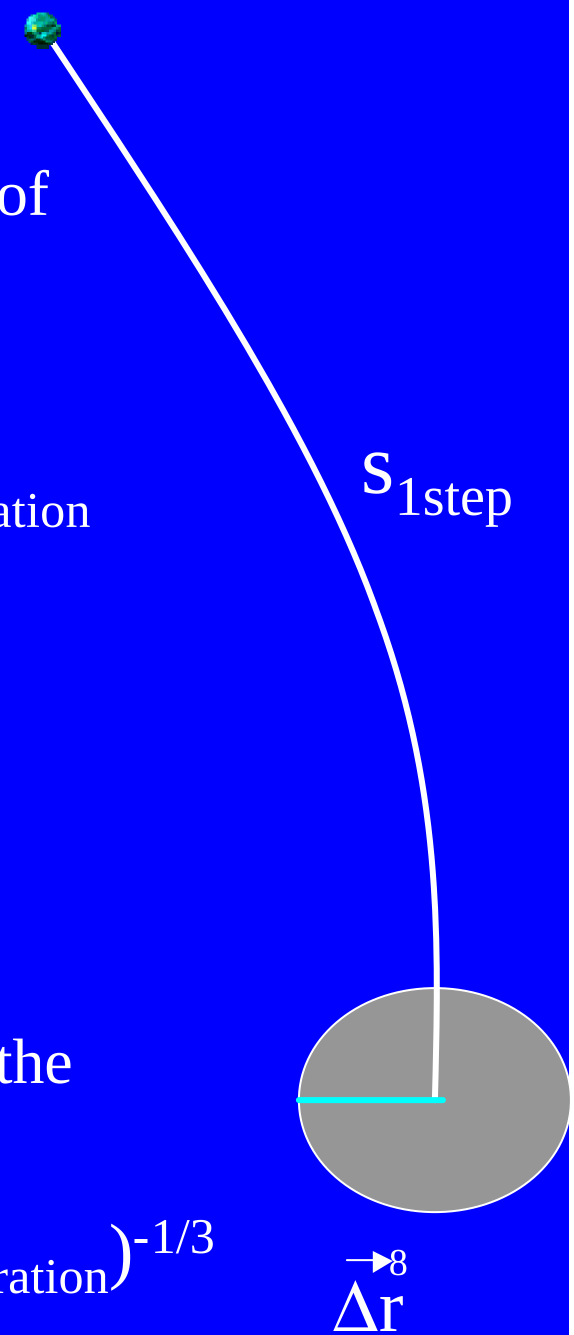
so long as $s^{propagator} < s^{phys}$ and $s^{propagator} > d_{min}^{integr}$

Integration error

Due to error in the numerical integration (of equations of motion)

Parameter(s): $\varepsilon_{integration}$

$$\max\left(\| \Delta\vec{r} \| / s_{step} \,,\; \|\Delta\vec{p}\| / \|\vec{p}\| \right) < \varepsilon_{integration}$$



- It limits the size of the integration step.
- For ClassicalRK4 Stepper

$$s_{1step}^{integration} \sim (\varepsilon_{integration})^{1/3}$$

for small enough $\varepsilon_{integration}$

- The integration error should be influenced by the precision of the knowledge of the field (measurement or modeling).

$$N_{steps} \sim (\varepsilon_{integration})^{-1/3}$$

Integration errors (cont.)

- In practice

  $\varepsilon_{integration}$ is currently represented by 3 parameters — Defaults

  - epsilonMin, a minimum value (used for big steps) — $0.5*10^{-7}$
  - epsilonMax, a maximum value (used for small steps) — 0.05
  - DeltaOneStep, a distance error (for intermediate steps) — 0.25 mm

$$\varepsilon_{integration} = \delta_{one\ step} / s_{physics}$$

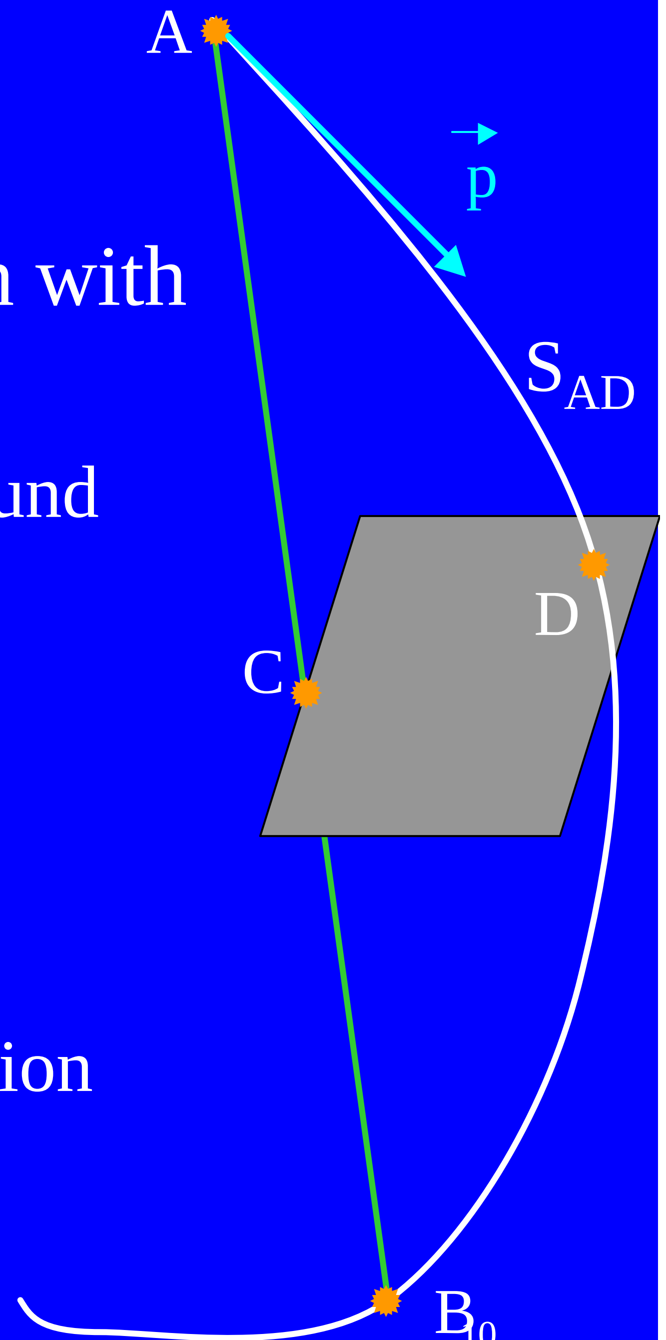
- Determining a reasonable value
  - I suggest it should be the minimum of the ratio (accuracy/distance) between sensitive components, ..
- Another parameter — Default
  - d_{min} is the minimum step of integration — 0.01 mm
    - (newly enforced in Geant4 4.0)

Intersection error

- In intersecting approximate path with volume boundary
  - In trial step AB, intersection is found with a volume at C
  - Step is broken up, choosing D, so

    $S_{AD} = S_{AB} * |AC| \ / \ |AB|$

  - If $|CD| < \delta_{intersection}$
    - Then C is accepted as intersection point.
  - So δ_{int} is a position error/bias

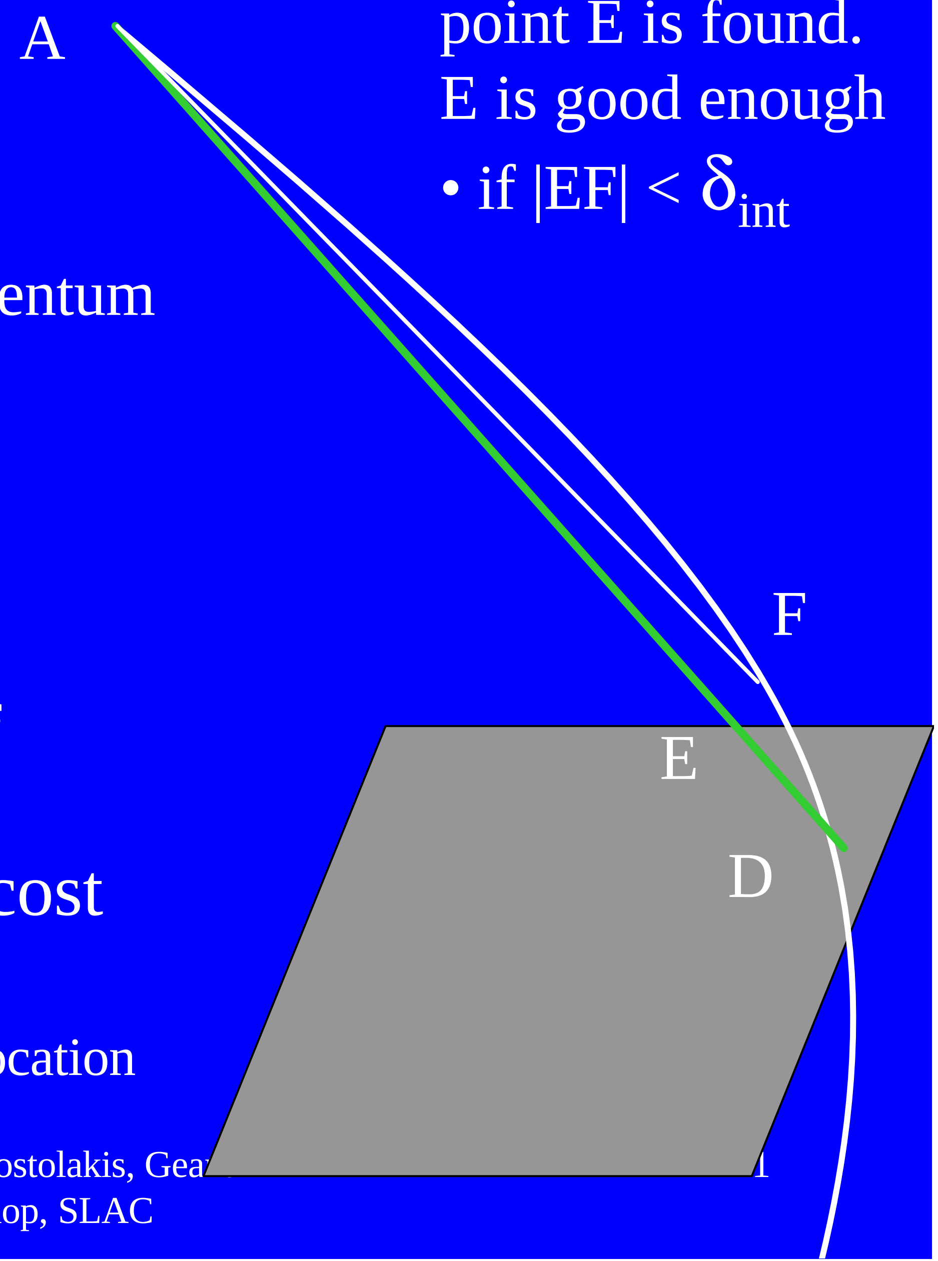


Intersection error (cont)

If C is rejected,
a new intersection
point E is found.
E is good enough

- if |EF| < δ_{int}

- So δ_{int} must be small
  - compared to tracker hit error
  - Its effect on reconstructed momentum estimates should be calculated
    - And limited to be acceptable
- Cost of small δ_{int} is less
  - than making δ_{chord} small
  - Is proportional to the number of boundary crossings – not steps.
- Quicker convergence / lower cost
  - Possible with optimization
    - adding std algorithm, as in BgsLocation



The ‘driving force’

- Distinguish cases according to the factor driving the tracking step length
  - ‘physics’, eg in dense materials
  - fine-grain geometry
- Distinguish the factor driving the propagator step length (if different)
  - Need for accuracy in ‘seeing’ volume
  - Integration inaccuracy
    - Strongly varying field

Potential Influence

G4 Safety improvement

Other Steppers, tuning d_{min}

What if time does not change much?

- If adjusting these parameters (together) by a significant factor (10 to 100) does not produce results,
  - Then it is likely that the field propagation is not the dominant (most CPU intensive) part of your program.
  - Look into alternative measures
    - modifying the physics ‘cuts’ – ie production thresholds
      - To create fewer secondaries, and so track fewer particles
    - determining the number of steps of neutral vs charged particles,
      - To find whether neutrons, gammas ‘dominate’
    - profiling your application
      - You can compile using G4PROFILE=yes, run your program and then use “gprof” to get an execution profile.

Where to find the parameters

Parameter	Name	Class	Default
δ_{miss}	DeltaChord	ChordFinder	3.0 mm
d_{min}	stepMinimum	ChordFinder	0.01 mm
$\delta_{intersection}$	DeltaIntersection	FieldManager	0.10 mm
$\delta_{one\ step}$	`DeltaOneStep`	FieldManager	0.25 mm
	epsilonMin	PropagatorInField	$5\ 10^{-7}$
	epsilonMax	PropagatorInField	0.05

Contributors to Field sub-category

- John Apostolakis
- Simone Giani
- Vladimir Grichine
- Wolfgang Wander

with thanks to David Williams

- for useful discussions
  - and expected contribution