More on Geometry: Magnetic Field

I. Hrivnacova, IPN Orsay

Credits: T. Nikitina, J.Apostolakis, G.Cosmo, A. Lechner (CERN), M. Asai (SLAC) and others

Geant4 ED PHENIICS Tutorial,
13 - 17 May 2019, Orsay

Outline

- Defining magnetic field
- Tunable parameters of propagation in magnetic field

Describe Your Detector

- To describe your detector you have to derive your own concrete class from G4VUserDetectorConstruction abstract base class.
- Implement the virtual method Construct(), where you
  - Instantiate all necessary materials
  - Instantiate volumes of your detector geometry
- Optionally, implement the virtual method ConstructSDandField(), where you
  - Instantiate your sensitive detector classes and set them to the corresponding logical volumes
  - Instantiate magnetic (or other) field
- Optionally you can define
  - Regions for any part of your detector
  - Visualization attributes (color, visibility, etc.) of your detector elements

Field Manager

- The magnetic field is applied to geometry with means of G4FieldManager
- One field manager is associated with the 'world' and it is set in G4TransportationManager, it handles the **global field**
  - The global field manager need not to be created by the user

```
G4FieldManager* fieldManager
      = G4TransportationManager::GetTransportationManager()
        ->GetFieldManager();
```

- An alternative field manager can be associated with any logical volume, it handles then the **local field**
  - By default this is propagated to all its daughter volumes
  - The field must accept position in global coordinates and return field in global coordinates

```
G4FieldManager* fieldManager = new G4FieldManager(magField);
logVolume->SetFieldManager(fieldManager, true);
```

    – Where 'true' means to propagate field to all the volumes it contains

Magnetic field

- Magnetic field class defines the strength of magnetic field within the world (global field) or within a given volume (local field)
- Magnetic field class:
  - Users can define their own concrete class derived from G4MagneticField and implement GetFieldValue method:

```
void MyField::GetFieldValue(
                 const double point[4], double* field) const;
```

  - where point[0..2] represents the position in global coordinate system and point[3] time
  - field[0..2] return the field value in the given position
- To define a **uniform magnetic field**, users need not to define their own class, but can use G4UniformMagField:

```
G4MagneticField* magField
   = new G4UniformMagField(G4ThreeVector(0, 0, 1.*Tesla));
```

Global Magnetic Field

```
void MyDetectorConstruction::CreateSDandField()
{
   // Magnetic field
   MyMagneticField* myField = new MyMagneticField();

   // Field manager
   G4FieldManager* fieldManager
      = G4TransportationManager::GetTransportationManager()
        ->GetFieldManager();
   fieldManager->SetDetectorField(myField);
   fieldManager->CreateChordFinder(myField);

}
```

Local Magnetic Field

```
void MyDetectorConstruction::CreateSDandField()
{
   // Magnetic field
   MyMagneticField* myField = new MyMagneticField();

   // Field manager
   G4Fieldmanager* fieldManager = new G4FieldManager();
   fieldManager->SetDetectorField(myField);
   fieldManager->CreateChordFinder(myField);

   // Set field to a logical volume
   G4bool forceToAllDaughters = true;
   magneticLogical
     ->SetFieldManager(fieldManager, forceToAllDaughters);
}
```

See also basic example B5

Global Field Messenger

- A helper class, G4GlobalMagFieldMessenger, is available since Geant4 10.00
  - It creates **the global uniform magnetic field**
  - **The field** is **activated** (set to the G4TransportationManager object) only when its fieldValue is non zero vector.
  - It can be also used to change the field value (and activate or inactivate the field again

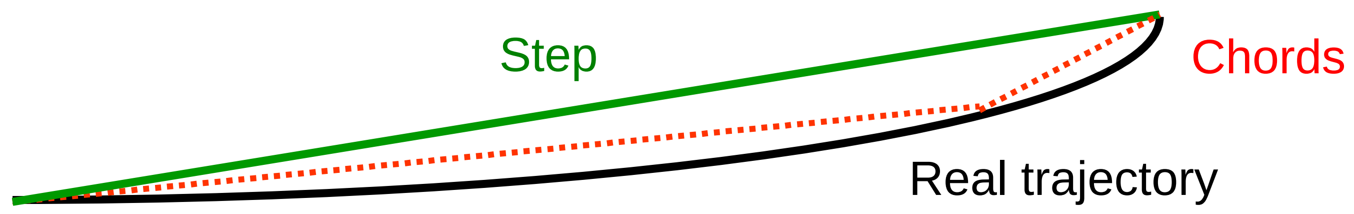
```
void MyDetectorConstruction::CreateSDandField
{
   // Global magnetic field & its messenger
   G4ThreeVector fieldValue = G4ThreeVector();
   G4GlobalMagFieldMessenger* magFieldMessenger
      = new G4GlobalMagFieldMessenger(fieldValue);

   // Register the messenger for deleting
   G4AutoDelete::Register(myFieldMessenger);
}
```

See basic examples B2 and B4

Propagation in Field
Tunable Parameters

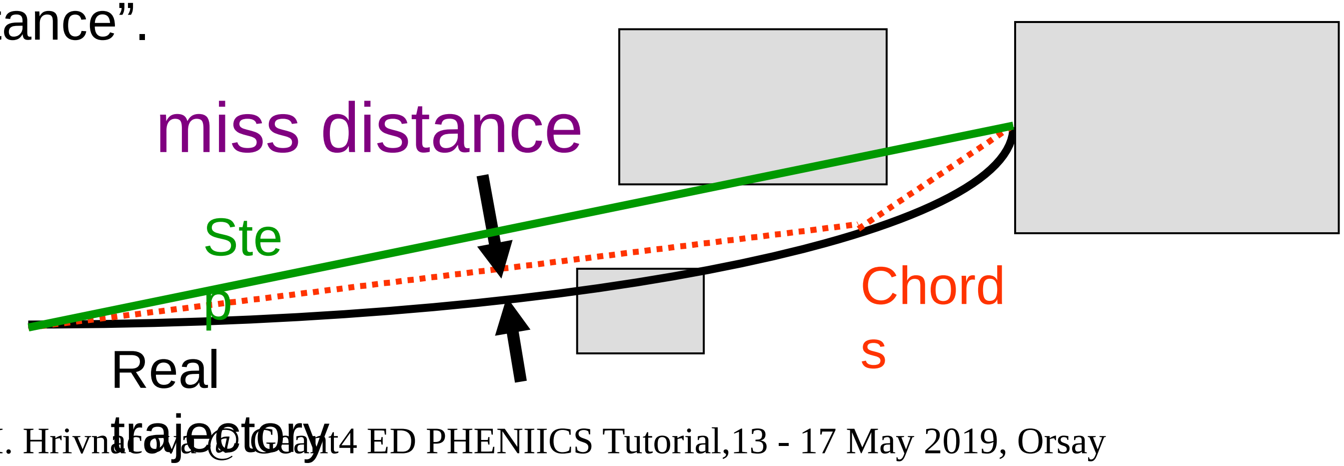
Propagation in Field (1)

- 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 analytic solution.
  - In a smooth but varying field, with RK+helix.
- 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.



Tunable Parameters

- We use the chords to interrogate the G4Navigator, to see whether the track has crossed a volume boundary.
- One physics/tracking step can create several chords.
  - In some cases, one step consists of several helix turns.
- User can set the accuracy of the volume intersection,
  - By setting a parameter called the “miss distance”
    - The curved trajectory will be approximated by chords, so that the maximum estimated distance between curve and chord is less than the the miss distance.
    - It is quite expensive in CPU performance to set too small “miss distance”.



11

Tunable Parameters (2)

- The "delta intersection" parameter is the accuracy to which an intersection with a volume boundary is calculated.
  - If a candidate boundary intersection is estimated to have a precision better than this, it is accepted.
  - This parameter is especially important because it is used to limit a bias that our algorithm (for boundary crossing in a field) exhibits.
- The "delta one step" parameter is the accuracy for the endpoint of 'ordinary' integration steps, those which do not intersect a volume boundary.
  - This parameter is a limit on the estimation error of the endpoint of each physics step.