User manual for the Geant4 based Magnetic shielding tool

February 24, 2011

Author Desorgher Laurent

Issued by SpaceIT GmbH

1 Introduction

We have developed different C+++ classes that allow to define magnetic field models in Geant4 as well as to control the tracking of charged particles in the magnetic field. This document represents the user guide of this code.

2 Installation of the code

All the files needed to install and test the code are contained in the archive file *mag_field_manager.tar.gz*.
The different files contained in the distribution of the code are organised like this:

- The directories *src* and *include* that contain all the new C++ classes (subdirectories include src).
- The *doc* directory that contains this user guide.
- The *example1* directory that contains some files needed to test and illustrate the use of he code.

The code is installed by doing the following:

- Type *tar -xvzf mag_field_manager.tar.gz*.
- Copy the content of the *src* and *include* subdirectories in the source of the Geant4 application where you want to use the code.
- Modify the main code of your Geant4 application by creating somewhere an instance of the static class MagneticFieldModelManager (MagneticFieldModelManager::GetInstance()).
- Recompile your G4 application.

3 Magnetic field models

The new G4 macro commands contained in the directoy /MagneticFieldModels and its subdirectories allow the user to define interactively some type of magnetic fields. With these commands the user control the magnetic field model manager that is reponsible in the code

for all what concerns the magnetic field. The magnetic field model manager contained different built-in magnetic field models that are described in section 3.2. These models depend on different parameters that the user can define interactively by using the commands contained in the subdirectory "/MagneticFieldModels/*ModelName*" associated to each model where *ModelName* represents the name of a given model. After the definition of the paratmeters, a field of a given model can be created by using the command "/MagneticFieldModels/CreateAParametrizedField *ModelName FieldName*" where *FieldName* defines the name of the created field. The user can define and create several fields during the same run. The field that will be used in the simulation is selected from the list of the already created fields by using the command "/MagneticFieldModels/SetGlobalField *FieldName*". The magnetic field is switched off by using the command "/MagneticFieldModels/RemoveGlobalField". It is also possible to create a compound field resulting from the addition of already created fields (see section 3.2.2)

The Magnetic field model manager source and all its associated classes can be easily imported in any other Geant4 application. To use it in his application the user has to add the line

"G4MagneticFieldModelMagner* theFieldModelManager =
G4MagneticFieldModelManager::GetInstance()"

somewhere in the source of the code.

3.1 General commands

/MagneticFieldModels/CreateAParametrizedField *model_name field_name*

Parameters

string *model_name* Type of the field to be created
string *field_name* Name of the field to be created

Description

Create a magnetic field of type *model_name*. The parameters defining the field have to be defined before by using the commands contained in the directory /MagneticFieldModels/*model_name*

/MagneticFieldModels/SelectGlobalField *field_name*

Parameters

string *field_name* Name of the field that will be used for the simulation.

Description

Select the field that will be used in the simulation from the list of already created fields.

/MagneticFieldModels/RemoveGlobalField

Parameters

None

Description

If this command is used no magnetic field will be considered in the simulation.

/MagneticFieldModels/ComputeJCurrentOnACartesianGrid
field_name X_0 dX nX Y_0 dY nY Z_0 dZ nZ h ***unit_length file_name***

Parameters

string *field_name*	Name of the field
double X_0, Y_0, Z_0	Cartesian coordinates of the first node of the grid
double *dX, dY, dZ*	The X,Y and Z dimension of a grid cell
int *nX, nY, nZ*	The number of cells in the X,Y and Z direction.
double *h*	Lenght of the step used in the numerical derivation
string *unit_length*	Unit of length
string *file_name*	Name of the ASCII file where the results will be written.

Description

Compute the current density over a 3D cartesian grid and store the results in an ASCII file. The current density $\vec{J}$ is computed from the magnetic field defined by *field_name* by considering the Maxwell equation $\vec{J} = \mu_0 \vec{\nabla} \times \vec{B}$. The parameter h defines the length step used for the numerical derivation of $\vec{\nabla} \times \vec{B}$.

/MagneticFieldModels/ComputeBfieldOnACartesianGrid
$field\_name$ X_0 dX nX Y_0 dY nY Z_0 dZ nZ $length\_unit$ $file\_name$

Parameters

string *field_name*	Name of the field
double X_0, Y_0, Z_0	Cartesian coordinates of the first node of the grid
double *dX, dY, dZ*	The X,Y and Z dimension of a grid cell
int *nX, nY, nZ*	The number of cells in the X,Y and Z direction.
string *lenght_unit*	
string *file_name*	ASCII file where the results of the computation are registered

Description

Compute the magnetic field over a 3D cartesian grid and store the results in an ASCII file.

/MagneticFieldModels/ComputeBfieldOnAnCylindricalGrid
$field\_name$ ϕ_0 $d\phi$ $n\phi$ R_0 dR nR Z_0 dZ nZ $file\_name$

Parameters

string *field_name*	Name of the field
double ϕ_0	start of the meshing for the ϕ coordinate
double $d\phi$	size of the meshing step for the ϕ coordinate
double $n\phi$	number of meshing step for the ϕ coordinate
string *angle_unit*	
double $R0$	start of the meshing for the cylindrical coordinate R
double dR	distance between two consecutive nodes along R
int nR	Number of nodes along R
double *Z0*	Z Coordinate of the first node of the grid
double dZ	Z distance between succesive nodes
int nZ	Number of nodes along the Z axis
string *length_unit*	
string *file_name*	ASCII file where the results of the computation are registered

Description

Compute the magnetic field on 3D cylindrical grid.

3.2 Available magnetic field models

The different magnetic field models that are available to the user are: a uniform field, a replicated field (see section3.2.3), the field produced by a linear current section, the field produced by a curved current section, the field produced by a rectangular coil, and a compound field. These models depend on different parameters that the user can define interactively by using the commands contained in the subdirectory "/MagneticFieldModels/*ModelName*" associated to each model where *ModelName* represents the name of a given model (see below). After the definition of its parameters a field of a specific model is created by using the command "/MagneticFieldModels/CreateParametrisedField *ModelName FieldName*"

3.2.1 Creation of a uniform field

The command /MagneticFieldModels/UniformField/SetBvec allows to define the component of a uniform field. After this definition the field is created by using the command "/MagneticFieldModels/CreateParametrisedField UniformField *FieldName*"

/MagneticFieldModels/UniformField/SetBvec B_x B_y B_z $bfield\_unit$

Parameters

double B_x, B_y, B_z	Cartesian components of the magnetic field
string $bfield\_unit$	Candidates (T,nT,G)

Description

Define a uniform magnetic field.

3.2.2 Creation of a compound field

A compound field is the sum of different magnetic fields already created by the user. Each of these fields can be associated to a specific coordinate transformation that defines the orientation and position of the coordinate system in which the field is defined into the world coordinate system. The way to define the different coordinate transformations is described in section 3.3 A compound field should be defined by using the commands contained in the directory /MagneticFieldModels/CompoundField before it is created created by using the command "/MagneticFieldModels/CreateParametrisedField CompoundField *FieldName*"

/MagneticFieldModels/CompoundField/AddField *field_name trans_name*

Parameters

string *field_name*	Name of the field to be added
string *trans_name*	Name of the transformation associated to the field to be added. It should be the name of a transformation already created. If no name is given no coordinate transformation will be considered for the new added field.

Description

Add a field to the list of fields already contained in the compound field and define its corresponding coordinate transformation.

/MagneticFieldModels/CompoundField/Reset

Parameters
None
Description
Reset the list of field contained in the next compound field to be created.

3.2.3 Creation of a replicated field

A replicated field is formed by different copies of the same field arranged such that the center of the i^{th} copy is defined in cylindrical coordinate by $(r = R,\ \phi = \phi_0 + (i-1)\ d\phi,\ Z = 0)$ while its internal coordinate system is rotated around the Z axis by an angle $(i-1)\ d\phi$ compared to the world coordinate system. The parameters R, ϕ_0, $d\phi$, the number of copy N, and the name of the field to replicate are defined by the user using the command contained in the directory "/MagneticFieldModels/ReplicateField/". After the definition of these parameters the field can be created by using the command "/MagneticFieldModels/CreateParametrisedField ReplicateField *NewFieldName*"

/MagneticFieldModels/ReplicateField/SelectField *field_name*

Parameters
string *field_name* Name of the field to be replicated
Description
Define the name of the field that wil be replicated.

/MagneticFieldModels/ReplicateField/SetR *R length_unit*

Parameters
double *R*
string *length_unit* Length unit in which *R* is defined
Description
Define the parameter *R*. See description above.

/MagneticFieldModels/ReplicateField/SetPhi0 ϕ_0 *angle_unit*

Parameters
double ϕ_0
string *angle_unit* Candidates (degree, rad)
Description
Defines the parameter ϕ_0. See description above.

/MagneticFieldModels/ReplicateField/SetdPhi $d\phi$ *angle_unit*

Parameters
double $d\phi$
string *angle_unit* Candidates (degree, rad)
Description
Define the parameters $d\phi$. See description above.

/MagneticFieldModels/ReplicateField/SetNCopies *N*

Parameters
int *N*
Description
Define the number of times the field will be replicated.

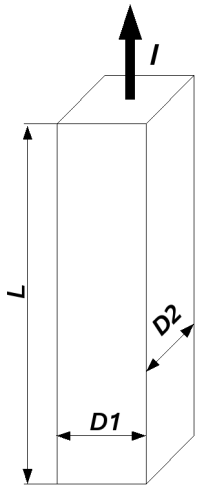

Figure 1: Description of a linear current segment. The current of magnitude I is flowing in the Z direction upward and is contained in a parallelepiped of lentgh L_z and cross sections $L_x \times L_y$.

3.2.4 Creation of the field produced by a linear current section

The commands contained in the directory /MagneticFieldModels/LinearSegment can be used to define a magnetic field produced by a linear current segment. The geometry of such a segment is described in Figure 1. The current of magnitude I is flowing in the Z direction upward and is contained in a box of length L_z and cross sections $L_x \times L_y$. The center of the box is placed at the position (0,0,0). After the definition of the parameters of the linear current segment the field is created by using the command "/MagneticFieldModels/CreateParametrisedField LinearSegment *NewFieldName*"

/MagneticFieldModels/LinearSegment/SetLx L_x *lenght_unit*

Parameters

double L_x

string *lenght_unit* Unit in which L_x is defined

Description

Define the length L_x of a linear current segment. See Figure1

/MagneticFieldModels/LinearSegment/SetLy L_y *lenght_unit*

Parameters
double L_y
string *lenght_unit* Unit in which L_y is defined
Description
Define the length L_y of a linear current segment. See Figure1

/MagneticFieldModels/LinearSegment/SetLz L_z *lenght_unit*

Parameters
double L
string *lenght_unit* Unit in which L_z is defined
Description
Defines the lenght L_z of the linear current segment. See Figure1.

/MagneticFieldModels/LinearSegment/SetI

Parameters
double I
string *current_unit* Unit in which I is defined
Description
Define the currrent I carried by the linear segment. See Figure1.

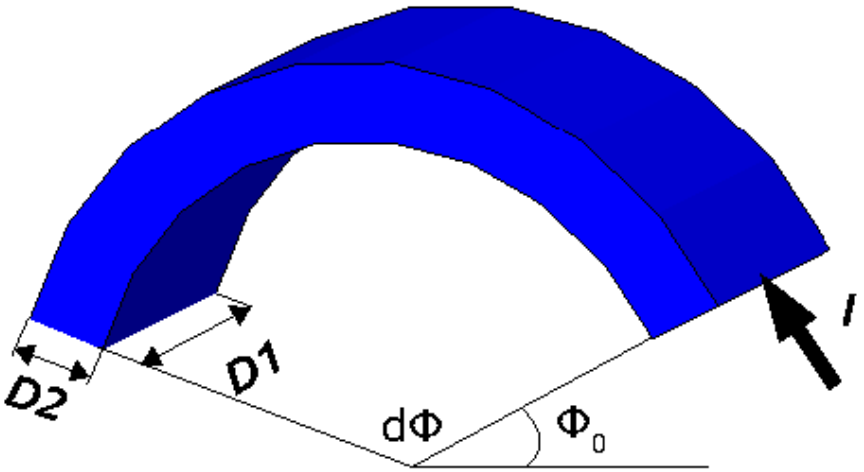

Figure 2: Description of a curved current segment. It is is formed by a section of a tube with a rotation axis parallel to the X direction. The parameters $D1$, R_{in}, R_{out}, ϕ_0, and $d\phi$ represent the length, inner radius, outer radius, start ϕ and opening angle of tube section. The current of magnitude I is flowing anticlockwise parallel to the YZ plane.

3.2.5 Creation of the field produced by a curved current section

The commands contained in the directory /MagneticFieldModels/CurvedSegment can be used to define a magnetic field produced by a curved current segment. The geometry of such a segment is described in Figure 2

/MagneticFieldModels/CurvedSegment/SetD1 $D1$ *length_unit*

Parameters
double *D1*
string *length_unit* Unit in which $D1$ is defined
Description
Define the length $D1$ of the curved segment. See Figure 2.

/MagneticFieldModels/CurvedSegment/SetRout R_{out} *length_unit*

Parameters
double R_{out}
string *lenght_unit* Unit in which R_{out} is defined
Description
Define the outer radius of the curved segment. See Figure 2

/MagneticFieldModels/CurvedSegment/SetRin R_{in} *length_unit*

Parameters
double R_{in}
string *lenght_unit* Unit in which R_{in} is defined
Description
Define the inner radius of the curved segment. See Figure 2

/MagneticFieldModels/CurvedSegment/SetPhi0 ϕ_0 *angle_unit*

Parameters
double ϕ_0
string *angle_unit* Unit in which ϕ_0 is defined
Description
Define the starting angle of the curved segment. See Figure 2

/MagneticFieldModels/CurvedSegment/SetDphi $d\phi$ *length_unit*

Parameters
double $d\phi$
double *angle_unit* Unit in which $d\phi$ is defined
Description
Define the angle covered by the curved segment. See Figure 2

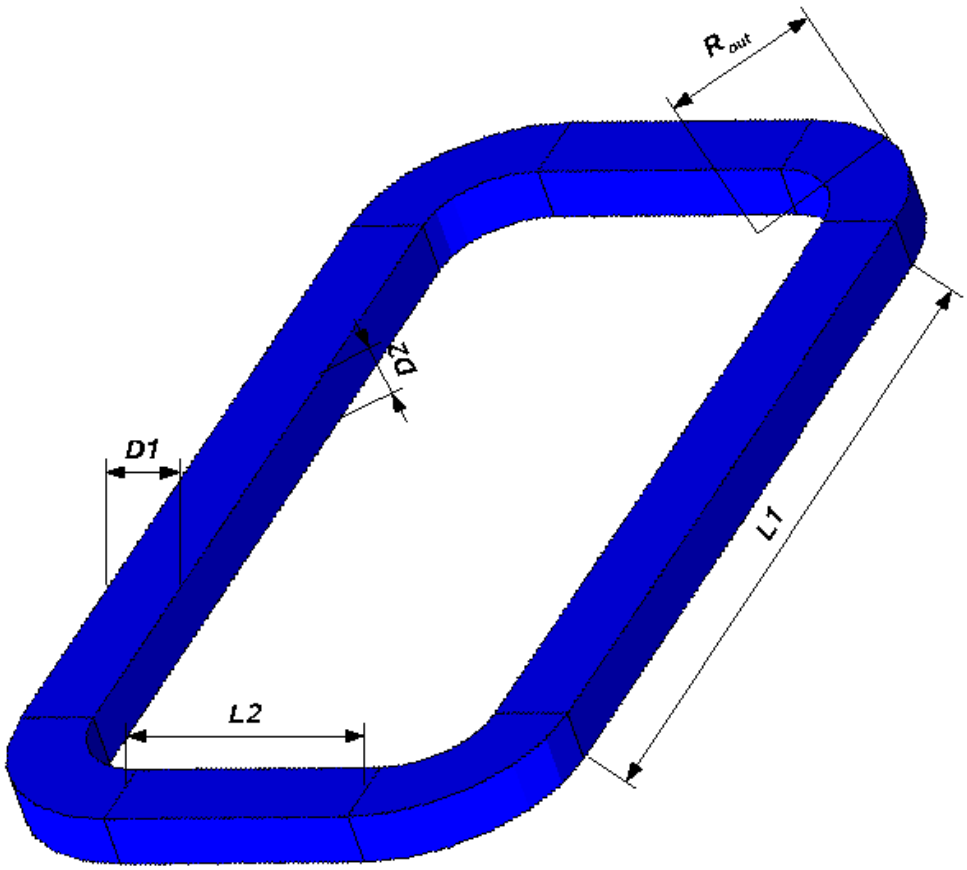

Figure 3: Definition of the geometry of a rectangular coil. It is made of two vertical linear segments of length $L1$, two horizontal linear segments of length $L2$, joined by four circular segments covering an angle of 90 degrees and with an external radius R_{out}. The cross section of the coil is rectangular and of dimension $D1 \times D2$. The coil is oriented paralle to the YZ plane.

3.2.6 Create the field produced by a rectangular coil

Using the macro commands given in the directory /MagneticFieldModels/Coil it is possible to define a current coil with the geometry described in Figure 3. Such coil is made of two vertical segments of length $L1$, two horizontal segments of length $L2$, joint together by four circular segments with an opening angle of 90 degree and a external radius R_{out}. The cross section of each current segment is rectangular and of dimension $D1 \times D2$. The coil is oriented parallel to the the YZ plane and its center is placed at the center of the coordinate system. The current of magnitude I is turning anti-clockwise in the coil.

/MagneticFieldModels/Coil/SetRout R_{out} *length_unit*

Parameters

double R_{out}

string *length_unit* Unit in which R_{out} is defined

Description

Define the external radius of the different curved sections in the coil. See Figure 3.

/MagneticFieldModels/Coil/SetD1 *D*1 *length_unit*

Parameters
double *D1*
string *length_unit* Unit in which *D*1 is defined
Description
Define the length *D*1 of the cross section of the coil. See Figure 3.

/MagneticFieldModels/Coil/SetD2 *D*2 *length_unit*

Parameters
double *D2*
string*length_unit* Unit in which *D*2 is defined
Description
Define the length *D*2 of the cross section of the coil. See Figure 3.

/MagneticFieldModels/Coil/SetL1*L*1 *length_unit*

Parameters
double *L1*
string *length_unit* Unit in which *L*1 is defined
Description
Defines the length *L*1 of the vertical linear current sections. See Figure 3.

/MagneticFieldModels/Coil/SetL2

Parameters
double *L2*
string *length_unit* Unit in which *L*2 is defined
Description
Defines the length *L*2 of the horizontal linear current sections. See Figure 3.

/MagneticFieldModels/Coil/SetI *I current_unit*

Parameters
double *I*
string *current_unit* Unit in which *I* is defined
Description
Define the currrent *I* carried by the coil.

3.2.7 Creation of an interpolated field

It is possible to create a field that is obtained by interpolation of the field precomputed on a cartesian grid or a cylindrical grid. The component of the field on the grid is read in a ASCII file by using the command /MagneticFieldModels/InterpolatedField/SetGridFileName. This file can be produced by using the commands /MagneticFieldModels/ComputeBfieldOnACartesianGrid or /MagneticFieldModels/ComputeBfieldOnACylindricalGrid. The field is created by using the command "/MagneticFieldModels/CreateParametrisedField InterpolatedField *NewFieldName*"

For the case of the cartesian grid, that is the X,Y, and Z components of the field that are interpolated, and the field vanishes for positions situated outside the grid limits. For the case

of the cylindrical grid, that is the ρ, ϕ and Z components of the field that are interpolated and the field vanish for positions that have their ρ and Z cordinates outside the grid limits. If the grid is covering only an azimuthal sector, the field is considered as having an azimuthal symetry, such that two positions that differs by a rotation around the Z axis with an angle representing n times the angle covered by the grid have their field with the same cylindrical components. For this particular case, it is important that the grid covered an entire fraction of 2π.

/MagneticFieldModels/InterpolatedField/SetGridFileName *file_name*

Parameters
string *file_name*
Description
Set the name of the file from which the field components pre-computed on a grid will be readed.

3.3 Definition of coordinate transformations

Coordinate transformation that can be associated to a field in a compound field can be created by using the commands contained in the directory /CoordinateTransformation/. A new transformation is obtained by succesive rotation and/or translation starting from the unit transformation or from a transformation selected in the list of previously created transformations by using the command "/CoordinateTransformation/Select". Once a transformation is finished it is registered for later use by using the commands /CoordinateTransformation/Register.

/CoordinateTransformation/Select *transfo_name*

Parameters
string *transfo_name* Name of the transformation to select
Description
Select a transformation in the list of already created transformation, as a start for the creation of a new transformation.

/CoordinateTransformation/ResetToUnit

Parameters
None
Description
Reset the transformation being created to the unit transformation in order to start the creation of a new transformation.

/CoordinateTransformation/AddRotationX ϕ *angle_unit*

Parameters
double ϕ Angle of rotation
string *angle_unit* Candidates (degree, radian)
Description
Add a rotation around the X axis, to the transformation being created.

/CoordinateTransformation/AddRotationY ϕ *angle_unit*

Parameters

double ϕ	Angle of rotation
string *angle_unit*	Candidates (degree, radian)

Description

Add a rotation around the Y axis, to the transformation being created.

/CoordinateTransformation/AddRotationZ ϕ *angle_unit*

Parameters

double ϕ	Angle of rotation
string *angle_unit*	Candidates (degree, radian)

Description

Add a rotation around the Z axis, to the transformation being created.

/CoordinateTransformation/AddRotationAroundAnAxis A_x A_y A_z ϕ *angle_unit*

Parameters

double A_x, A_y, A_z	Define the axis of rotation
double ϕ	Angle of rotation
string *angle_unit*	Candidates (degree, radian)

Description

Add a rotation around the axis defined by the direction (A_x, A_y, A_z), to the transformation being created.

/CoordinateTransformation/AddTranslation T_x T_y T_z *length_unit*

Parameters

double T_x, T_y, T_z	Define the translation
string *length_unit*	Candidates (m, cm,km,mm,...)

Description

Add the translation (T_x, T_y, T_z) to the transformation being created.

/CoordinateTransformation/MultiplyTransformations $trans_1$ $trans_2$

Parameters

string $trans_1$, $trans_2$	Names of the transformation to multiply

Description

The transformation that is being created is set to the product of the transformation $trans_1$ and $trans_2$.

/CoordinateTransformation/MultiplyBy $trans_1$

Parameters

string $trans_1$	Name of a transformation

Description

The transformation that is being created is multiplied by the transformations $trans_1$

/CoordinateTransformation/Register *trans_name*

Parameters
string *trans_name*
Description
The transformation that is being created is registered in the list of craeted transformation with the name *trans_name*

3.4 Numerical integration of the BiotSavart equation

The magnetic field induced by linear and curved current segments is obtained in the code by the numerical integration of the BiotSavart equation

$$\mathbf{B}(\mathbf{r}) = \frac{\mu_0}{4\pi}\int \frac{\mathbf{J}(\mathbf{r}')\times(\mathbf{r}-\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|^3}\mathbf{dr}' \tag{1}$$

where B, J, $\mathbf{r}$ represents the magnetic field, the current density and the position respectively. In the following we explain the different numerical methods of integration that are used in the code for this purpose. The user can define the numerical integration parameters by using the commands contained in the subdirecory /MagneticFieldModels/BiotSavartIntegrator/

3.4.1 Linear current segment

The magnetic field produced by a linear current segment as the one described in Figure 1 is given by

$$\vec{B} = \frac{\mu_0 I}{4\pi\ L_x L_y}\int_{-L_x/2}^{L_x/2} dx' \int_{-L_y/2}^{L_y/2} dy' \int_{-L_z/2}^{L_z/2} dz'\ \frac{(y-y')\vec{e}_{x'} - (x-x')\ \vec{e}_{y'}}{((x-x')^2+(y-y')^2+(z-z')^2)^{3/2}} \tag{2}$$

by integrating over x and z we obtain

$$B_x = \frac{\mu_0 I}{4\pi\ L_x L_y}\int_{-L_y/2}^{L_y/2} dy' \left[\left[\ atan(\frac{(z'-z)(x'-x)}{(y'-y)\ r})\right]_{x'=-L_x/2}^{x'=L_x/2}\right]_{z'=-L_z/2}^{z'=L_z/2} \tag{3}$$

$$B_y = \frac{\mu_0 I}{4\pi\ L_x L_y}\int_{-L_y/2}^{L_y/2} dy' \left[\left[\frac{log(r+z'-z)-log(r-z'+z)}{2}\right]_{x'=-L_x/2}^{x'=L_x/2}\right]_{z'=-L_z/2}^{z'=L_z/2} \tag{4}$$

where $r = \sqrt{(x'-x)^2+(y'-y)^2+(z'-z)^2}$

From the last equations we can see that the magnetic field produced by a current surface of dimension $L_x \times L_z$, parallel the XZ plane, with its center placed at the position $(0, y_0, 0)$ and with the current flowing in the Z direction is given by

$$B_x = \frac{\mu_0 I}{4\pi\ L_x}\left[\left[\ atan(\frac{(z'-z)(x'-x)}{(y_0-y)\ r})\right]_{x'=-L_x/2}^{x'=L_x/2}\right]_{z'=-L_z}^{z'=L_z} \tag{5}$$

$$B_y = \frac{\mu_0 I}{4\pi\ L_x}\left[\left[\frac{log(r+z'-z)-log(r-z'+z)}{2}\right]_{x'=-L_x/2}^{x'=L_x/2}\right]_{z'=-L_z}^{z'=L_z} \tag{6}$$

where $r = \sqrt{(x'-x)^2+(y_0-y)^2+(z'-z)^2}$

The method of integration that is used in the code to compute the field obtained by a linear current segment consist in the addition of the field produced by N_y different XZ surface currents, placed along the Y axis at the positions

$$y_0 = -L_y/2 + (1/2 + i)\frac{L_y}{N_y} \quad i = 0, N_y - 1$$

the current I/N_y of the total. The user can select the number of intregration step N_y by using the command "/MagneticFieldModels/BiotSavartIntegrator/SetNy" described below.

3.4.2 Curved current segment

In the following we explain how the field of a curved current segment (see Figure 2) is computed numerically from the Biot-Savart law. For this purpose we are working in the cylindrical coordinate system (x', ρ', ϕ') where

$$y' = \rho' cos(\phi') \tag{7}$$
$$z' = \rho' sin(\phi') \tag{8}$$

The current is going along the vector $\vec{e}_{\phi'}$ from the angle ϕ_1 to ϕ_2 and covers a rectangular section. The magnetic field produced by this current on the point (x, ρ, ϕ) is

$$\vec{B} = \frac{\mu_0 I}{4\pi\ 2l_x(\rho_o - \rho_i)} \int_{-l_x}^{l_x} dx' \int_{\rho_i}^{\rho_o} \rho' d\rho' \int_{\phi_1}^{\phi_2} d\phi'\ \frac{(\rho' - \rho cos(\phi' - \phi))\ \vec{e}_{x'} + (x - x')\ \vec{e}_{\rho'}}{((\rho' - \rho cos(\phi' - \phi))^2 + \rho sin(\phi' - \phi)^2 + (x - x')^2)^{3/2}} \tag{9}$$

After integration over x' we ρ' we obtain

$$B_x = C \int_{\phi_1}^{\phi_2} d\phi' \left[\left[(x' - x) log(\rho' - a + r) - a\ log(x' - x + r) - b\ atan(\frac{(\rho' - a)(x' - x)}{b\ r}) \right]_{x'=-l_x}^{x'=l_x} \right]_{\rho'=\rho_i}^{\rho'=\rho_o} \tag{10}$$

$$\vec{B}_\perp = C \int_{\phi_1}^{\phi_2} d\phi' \left[[\vec{e}_{\rho'}(\ r + a\ log(\rho' - a + r))]_{x'=-l_x}^{x'=l_x} \right]_{\rho'=\rho_i}^{\rho'=\rho_o} \tag{11}$$

where $a = \rho cos(\phi' - \phi)$, $b = \rho sin(\phi' - \phi)$, $r = \sqrt{(\rho' - \rho cos(\phi' - \phi))^2 + \rho sin(\phi' - \phi)^2 + (x - x')^2}$, and $C = \frac{\mu_0 I}{4\pi\ 2l_x(\rho_o - \rho_i)}$

The development of the equation (6) gives

$$B_y = C \int_{\phi_1}^{\phi_2} d\phi' cos(\phi') \left[[r + a\ log(\rho' - a + r)]_{x'=-l_x}^{x'=l_x} \right]_{\rho'=\rho_i}^{\rho'=\rho_o} \tag{12}$$

$$B_z = C \int_{\phi_1}^{\phi_2} d\phi' sin(\phi') \left[[r + a\ log(\rho' - a + r)]_{x'=-l_x}^{x'=l_x} \right]_{\rho'=\rho_i}^{\rho'=\rho_o} \tag{13}$$

The component B_x, B_y, B_z are obtained by a numerical integration over ϕ of the equations 12, 14 and 15. The user can select the number of integration step N_ϕ by using the command "/MagneticFieldModels/BiotSavartIntegrator/SetNphi" described below.

/MagneticFieldModels/BiotSavartIntegrator/SetNy N_y

Parameters
int N_y Number of integration step for an integration over Y
Description
Define the number of steps used for the numerical integration of the Biot Savart law when calculating the field produced by a current line segment (see above). .

/MagneticFieldModels/BiotSavartIntegrator/SetNphi

Parameters
int Ny Number of integration step for an integration over the azimuth angle
Description
Define the number of steps used for the numerical integration of the Biot Savart law when calculating the field produced by a curved line segment (see above).

4 Control of the numerical integration of the Lorentz Equation

The transport of charged particles trough an electromagnetic field is obtained in Geant4 by integrating numerically the Lorentz equation of motion. This numerical integration depends on several parameters. For a precise description of these parameters and of the propagation of charged particles in Geant4 we refer to the section 4.3 of the Geant4 User's Guide for Application Developers. Some of the integration parameters can be fixed by using the new macro commands in the directory /LorentzEquationIntegrator

/LorentzEquationIntegrator/SetPrecision ϵ

Parameters
double ϵ integration precision
Description
Set the Geant4 minimum and maximum epsilon (relative precision) step parameters for integration.

/LorentzEquationIntegrator/SetMaxMissDistance d_{max} *unit*

Parameters
double d_{max} maximum miss distance
string *unit* unit of length
Description
Set the maximum miss distance representing an upper bound of the distance between the real curved trajectory and the approximate linear trajectory of an intregation step. This parameter is important for controlling the precision of the detection of volume boundary.

/LorentzEquationIntegrator/SetDeltaIntersection δ *unit*

Parameters

double δ	precision for boundary crossing
string *unit*	unit of length

Description

Set the precision for the crossing of boundary.)

/LorentzEquationIntegrator/SetLargestAcceptableStep h *unit*

Parameters

double δ	largest acceptable length for the integration step
string *unit*	unit of length

Description

Set the largest acceptable length of thwe integration step. The largest acceptable step should be set to a distance larger than the biggest reasonable step. By default it is set to $1km$. For geometry bigger than typical high energy physics geometry this parameter should be scaled to the size of the geometry.

/LorentzEquationIntegrator/SetStepperModel *model_name*

Parameters

string *model_name*	name of the integration method

Description

Select one of the stepper/integration numerical methods available in Geant4 for the tracking of charged particle in magnetic field.

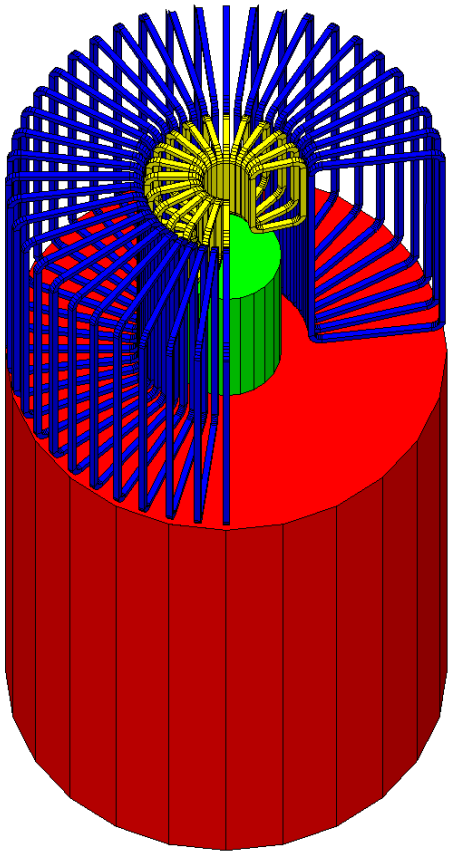

Figure 4: Configuration magnet modeled in the example 1. See text for details.

5 example1

Tha G4 macro files to run this example are contained under the directory example1. The geometry of the magnet considered in this example is plotted in Figure 4. The magnetic field is produced by the blue main toroid (5 m height, 5 m external radius) made of 48 rectangular coils and the yellow upper cap toroid made of 24 rectangular coils. In this figure only three quarters of the coils in the toroids are shown. The coils in both toroids have a $0.11 \times 0.11\ m^2$ section. The coils have a current density of $157 A/m^2$, representing a total current of $1.9\ 10^6 A$ carried by each coil. Different G4 macrofiles are provided to test the modeling of the toroids by the magnetic field model manager.

A full test can be run by typing *python launch_short_test_results.py G4executable_name* The test consists into

- The computation from the Biot-Savart law of the magnetic field produced by the main and upper toroids on cylindrical grids by running the macro files *MainToroidSmallGridComputation.g4mac* and *UpperToroidSmallGridComputation.g4mac*. The results

of these runs are registered in the files *main_toroid_small_grid.txt* and *upper_toroid_small_grid.txt*. In the rest of the test, the magnetic field is computed rapidly by interpolation of these pre-computed grids.

- Computation on a cylindrical grid of the total field by interpolation of the previously computed magnetic field grids. The macro file used for this run is the file *SmallGridTest.g4mac*. The results of this run are registered in the file *total_small_grid.txt*.

- Computation of the current density for one coil of the main toroid as a check of the correct magnetic field computation by the Biot-Savart law. The macro file used for this run is the file *CoilJTest.g4mac*. The results are registered in the file *coil_jtest.txt*.

If the *numpy* and *pylab* PYTHON modules are properly installed on the user machine, results of the test can be plotted by typing *python plot_short_test_results.py* in the *example1* directory. This will results in the production of three plots *B_small_grid.**, *JCoil.** in three different formats: pdf, ps and png. Note that the quality of the plots in the different formats depend on the *pylab* installation.

Figure 5 represents the spatial variation of the magnitude B of the magnetic field registered in the file *total_small_grid.txt*. It corresponds to the plot *B_small_grid.pdf* produced by typing *python plot_short_test_results.py*. As expected the field is mainly confined inside the toroids. The blue dark region in the upper top panel represents a region with $B < 50$ *Gauss* while in the white region $B < 1$ *Gauss*. In the upper left and lower panels the field inside the main toroid is equivalent to the field that should be expected from the Ampere's law represented in green in the plots. Figure 6 represents the magnitude J of the computed current density of one coil of the main toroid registered during the test in the file *coil_jtest.txt*. It corresponds to the plot *JCoil.pdf* produced by typing *python plot_short_test_results.py* . The computed current is confined in the coil and the current density is nearly uniform with its value close to the expected value of $157 A/mm^2$ (green horizontal line). The slight differences observed between the computed and expected current densities are due to numerical approximations. In conclusion from Figure 5 and 6 it can be concluded that the computation of the magnetic field is correct.

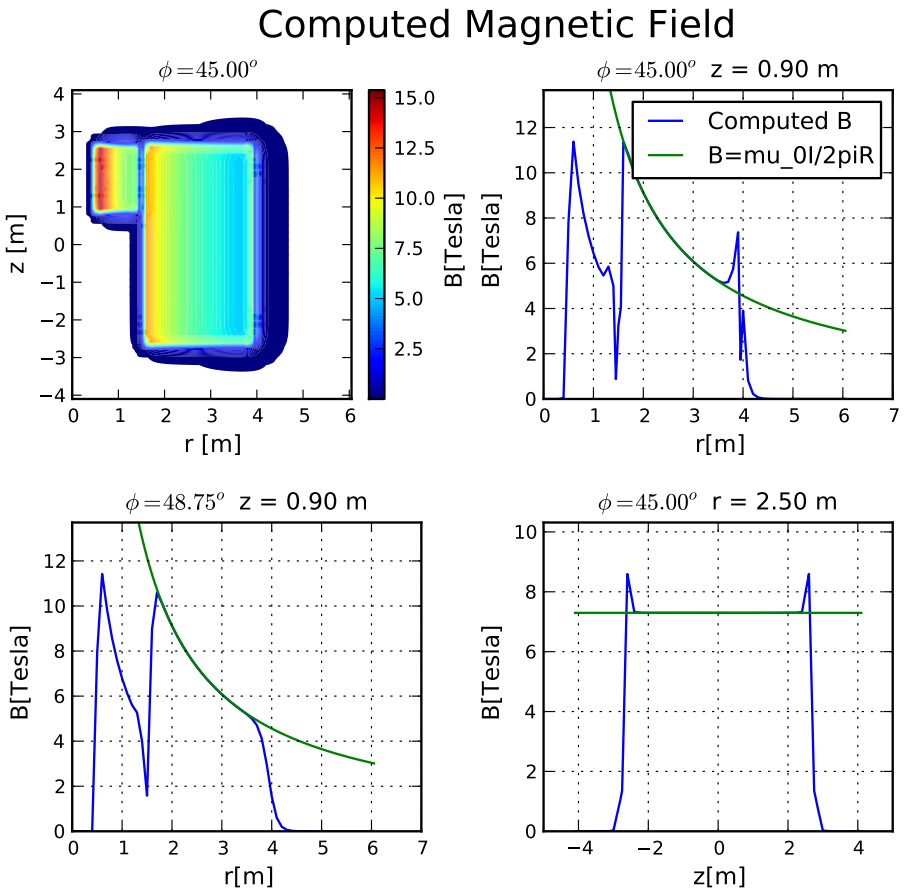

Figure 5: Computed magnetic field produced by the toroids of the active system modeled in the example 1. The results presented here where obtained during the run of the short test.

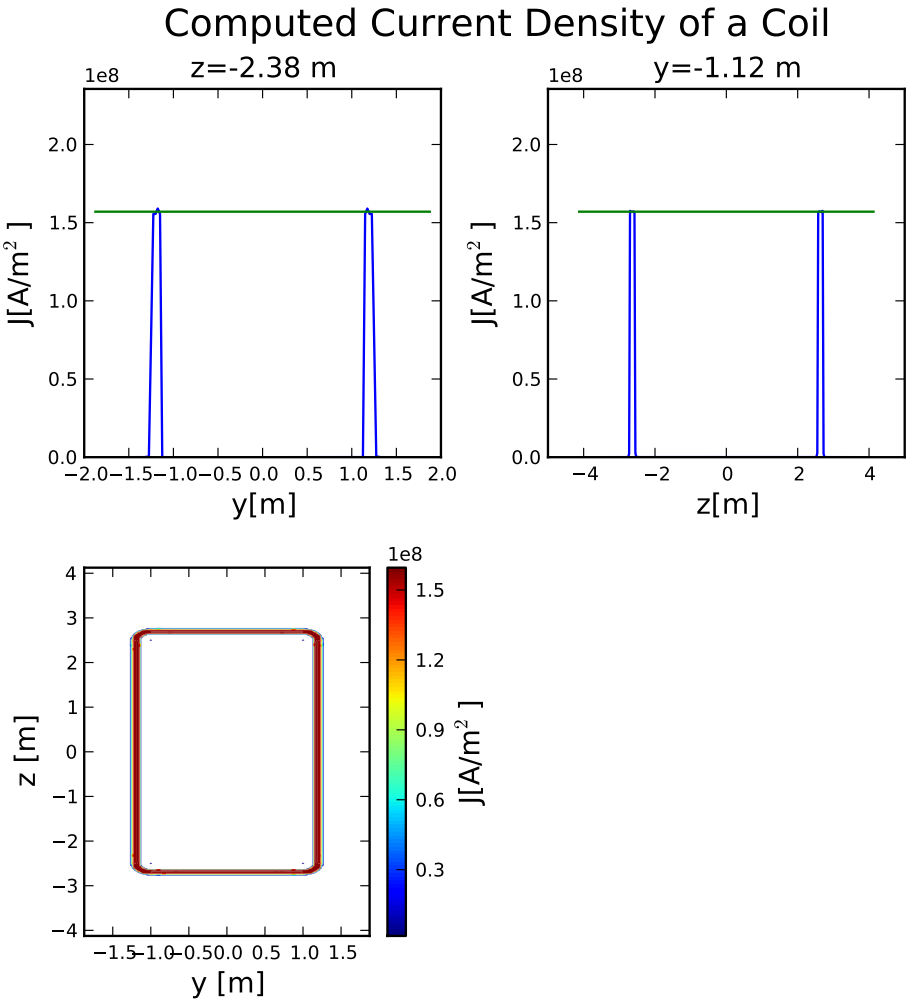

Figure 6: Computed magnitude of the current density of a coil of the main toroid of the active system modeled in the example 1. The results presented where obtained during the run of the short test.

References

Agostinelli et al., Geant4 a Simulation Toolkit, NIM A, 506, 250–303 (2003).

Geant4 Physics Reference Manual, http://geant4.web.cern.ch/geant4/UserDocumentation/-UsersGuides/PhysicsReferenceManual/html/index.html

Geant4 User's Guide For Application Developers, http://geant4.web.cern.ch/geant4/User-Documentation/UsersGuides/ForApplicationDeveloper/html/index.html

Geant4 Hadronic Physics List http://geant4.cern.ch/support/proc_mod_catalog/physics_lists-/physicsLists.shtml