

COMPUTING 3D MAGNETIC FIELDS FROM INSERTION DEVICES

P. Elleaume, O. Chubar, J. Chavanne, ESRF, B.P. 220, F-38043 Grenoble CEDEX, France

Abstract

A 3D magnetostatics computer code optimized for Undulators and Wigglers is described. The code uses a boundary integral method and makes extensive use of analytical expressions for the field and field integrals along a straight line. The code outperforms currently available finite element packages in the area of simple data input, CPU time of the solver and accuracy reached for the estimation of field integrals. It is written in C++ and takes advantage of object-oriented programming. The code is interfaced to Mathematica [1]. Pre- and post-processing of the field data is done in the Mathematica Language. It has been extensively benchmarked with respect to a commercial finite element code. All ESRF Insertion Devices built during the last 4 years have been designed using this code or an older version.

1 INTRODUCTION

The most common approach used to compute the magnetic field produced by an ensemble of permanent magnets, iron pieces and current coils is based on the Finite Element Method (FEM). The popular code POISSON which computes magnetic fields in 2D geometry belongs to this category. Several 3D packages are available commercially. Their development typically requires several man-years of programming. The input of geometry, materials and proper meshing requires significant skills and time from users.

In this paper we present an alternate approach based on the Boundary Integral Method (BIM) [2]. The Radia code has been continuously developed by the authors over the past 8 years representing a total effort of 1.5 man-year. During the development, the code has been regularly and successively checked against a commercial 3D FEM package called FLUX3D [3] and against field and field integral measurements made on real magnetic devices built in the ESRF Insertion Device laboratory. The code consists in a set of functions implemented in C++ and available in the Mathematica programming environment. The Mathematica language is used for pre- and post-processing. Versions of the code are currently available for Power Macintosh and Windows 95/NT [4].

2 PRINCIPLES OF THE CODE

A user of Radia creates several types of objects and combines them to solve the problem. The basic type of objects are the source objects capable of creating magnetic fields. This includes magnetized volumes, current coils of various shapes and container objects. The magnetized volumes can be subdivided into smaller

objects of the same kind with independent magnetizations. Another type of object describes material properties through the magnetization vs field law which can be a linear anisotropic (as for NdFeB or SmCo) or nonlinear isotropic (as for iron). The next type of objects are space transformations such as translation, rotation or plane symmetry. After creating the necessary objects, the user builds the model by linking the objects together.

The essential step consists in solving the problem in terms of magnetizations in all the volumes with particular material properties. This is done by creating in memory a large matrix also called an Interaction Matrix and applying a relaxation scheme to it. After the relaxation, magnetic field and field integrals can be computed anywhere in space by summing the field created by all the sources. The field from each source is computed with efficient analytical formulas. The accuracy of the computed field only depends on the level of segmentation of the iron and magnet pieces.

2.1 Analytical Expressions for Field and Field Integrals

Magnetic field \mathbf{H} produced by an object of arbitrary shape with its volume uniformly magnetized according to the vector \mathbf{M} can be expressed by a matrix relation

$$\mathbf{H} = \mathbf{QM}. \quad (1)$$

If the object is a rectangular parallelepiped block with faces parallel to XY, XZ and YZ planes of the Cartesian frame, the components of the matrix \mathbf{Q} are represented by the well-known formulas

$$\begin{aligned} Q_{xx} &= \frac{1}{4\pi} \sum_{i,j,k=1}^2 (-1)^{i+j+k+1} \tan^{-1} [x_i^{-1} y_j z_k (x_i^2 + y_j^2 + z_k^2)^{-1/2}], \\ Q_{xy} &= \frac{1}{4\pi} \ln \left[\prod_{i,j,k=1}^2 [z_k + (x_i^2 + y_j^2 + z_k^2)^{1/2}]^{(-1)^{i+j+k}} \right], \end{aligned} \quad (2)$$

where

$$x_{1,2} = x_c - x_0 \mp w_x/2, \quad y_{1,2} = y_c - y_0 \mp w_y/2, \quad z_{1,2} = z_c - z_0 \mp w_z/2, \quad (3)$$

(x_c, y_c, z_c) are Cartesian coordinates of the block's center, (w_x, w_y, w_z) the block's dimensions, (x_0, y_0, z_0) coordinates of an observation point. $Q_{yy}(Q_{zz})$ is deduced from Q_{xx} by exchanging $x \leftrightarrow y$ ($x \leftrightarrow z$) in (2); $Q_{xz}(Q_{yz})$ is deduced from Q_{xy} by exchanging $y \leftrightarrow z$ ($x \leftrightarrow z$). In addition,

$$Q_{ll'} = Q_{l'l}, \quad l, l' = x, y, z. \quad (4)$$

In accelerator physics, the magnetic field integrated along a straight line from $-\infty$ to $+\infty$ is of major importance. Numerical integration of the field is time consuming and fundamentally inaccurate. In Radia, the

field integrals are computed from an analytical integration except for a few cases when such expressions cannot be derived easily.

Let \mathbf{I} be the field integral along a line parallel to a vector $\mathbf{v}=(v_x, v_y, v_z)$, $|\mathbf{v}|=1$, and passing through a point \mathbf{r}_0 . For a rectangular parallelepiped block with faces parallel to XY, XZ and YZ planes and uniform magnetization \mathbf{M} , \mathbf{I} can be expressed as

$$\mathbf{I} \equiv \int_{-\infty}^{+\infty} \mathbf{H}(\mathbf{r}_0 + \mathbf{v}s) ds = \mathbf{G}\mathbf{M}, \quad (5)$$

$$\begin{aligned} G_{xx} = & \frac{1}{2\pi} \sum_{i,j,k=1}^2 (-1)^{i+j+k} \left[(v_z z_k u_y^{-1} - v_y y_j u_z^{-1}) \tan^{-1} \left[\frac{u_x x_i - (v_y y_j + v_z z_k) v_x}{v_z y_j - v_y z_k} \right] + \right. \\ & + v_x x_i \left[u_y^{-1} \tan^{-1} \left[\frac{u_y y_j - (v_x x_i + v_z z_k) v_y}{v_x z_k - v_z x_i} \right] + u_z^{-1} \tan^{-1} \left[\frac{u_z z_k - (v_x x_i + v_y y_j) v_z}{v_x y_j - v_y x_i} \right] \right] + \\ & + [(v_x y_j - v_y x_i) v_z u_z^{-1} + (v_x z_k - v_z x_i) v_y u_y^{-1}] L_{ijk} \Big], \\ G_{xy} = & \frac{1}{2\pi} \sum_{i,j,k=1}^2 (-1)^{i+j+k} \left[(v_y x_i - v_x y_j) u_z^{-1} \tan^{-1} \left[\frac{u_z z_k - (v_x x_i + v_y y_j) v_z}{v_x y_j - v_y x_i} \right] + \right. \\ & + [(v_x x_i + v_y y_j) v_z u_z^{-1} - z_k] L_{ijk} \Big], \\ L_{ijk} = & \ln[(v_x y_j - v_y x_i)^2 + (v_x z_k - v_z x_i)^2 + (v_z y_j - v_y z_k)^2] / 2, \\ u_l = & 1 - v_l^2, \quad l = x, y, z, \end{aligned} \quad (6)$$

where $x_{1,2}, y_{1,2}, z_{1,2}$ are given by (3) with (x_0, y_0, z_0) now being coordinates of the point \mathbf{r}_0 on the line along which the integration is done. The other components of the matrix \mathbf{G} can be derived using the same rules as for the corresponding components of the matrix \mathbf{Q} (the exchanging procedure concerns now also the components of the vector \mathbf{v}). Eqs. (6) only apply if the line does not cross the body of the block. If $v_x = v_z = 0$ (integration along the y edge), Eqs. (6) reduce to

$$\begin{aligned} G_{xx} = & \frac{w_y}{2\pi} \sum_{i,k=1}^2 (-1)^{i+k} \tan^{-1}(x_i z_k^{-1}), \quad G_{zz} = \frac{w_y}{2\pi} \sum_{i,k=1}^2 (-1)^{i+k} \tan^{-1}(x_i^{-1} z_k), \\ G_{xz} = & \frac{w_y}{4\pi} \ln \left[\frac{(x_1^2 + z_2^2)(x_2^2 + z_1^2)}{(x_1^2 + z_1^2)(x_2^2 + z_2^2)} \right], \quad G_{yy} = G_{xy} = G_{yz} = 0 \end{aligned} \quad (7)$$

The linearity of (1) and (5) is the general property applicable to any shape. The specificity of the shape only appears in (2) and (6). The corresponding Matrices \mathbf{Q} and \mathbf{G} derived for the general case of a volume enclosed by a set of planar polygons will be detailed in a future publication. At present, Radia handles the rectangular parallelepipeds and any volumes defined by a thick polygon (straight prism with a polygonal base).

The formulas analogous to (1)-(3), for the case of a current carrying rectangular parallelepiped block with constant current density can be found in [5]. We have also implemented a current carrying arc element with constant rectangular cross-section (details to be published later). Finally, the vector potential can be computed on request for all the elements using the corresponding analytical formulas.

2.2 Space Transformations

Space transformations are used for orientation of field source elements in space as well as for simulation of boundary conditions by mirroring. There are four basic space transformations in use: translation, plane symmetry, rotation around an arbitrary axis, and field inversion. Several space transformations can be combined (through matrix multiplication) into a new space transformation. Let \mathbf{T} be a transformation that applies to a point \mathbf{r} or to a source object S , $\mathbf{H}(\mathbf{r}, S)$ be the field produced by the object S at the point \mathbf{r} . The field produced by the transformed object $\mathbf{T}S$ can be computed by means of the following identity:

$$\mathbf{H}(\mathbf{r}, \mathbf{T}S) = \mathbf{T}\mathbf{H}(\mathbf{T}^{-1}\mathbf{r}, S). \quad (8)$$

It is important to note that \mathbf{T} applies not only to the point \mathbf{r} , but also to the field components. The laws of the transformation may differ for various types of vectors. Radial and axial vectors transform differently.

A transformation \mathbf{T} may be applied to a source S with some multiplicity m . $m = 2$ is typically used to simulate a boundary condition, while $m > 2$ may be used to simulate a multipole field symmetry. In such a case, the field produced by S results in the sum of the fields created by the objects $\mathbf{T}^i S$ with i varying from 0 to $m-1$. Applying transformations with a multiplicity can be understood as an efficient use of the symmetries in the model being solved. This results in a minimum number of degrees of freedom and therefore dramatically reduces the memory requirements and CPU time needed for the solution.

2.3 Interaction Matrix and Relaxation

Let us consider N source objects having magnetization vectors \mathbf{M}_i , $i=1,2,\dots,N$. Let \mathbf{H}_i be the field strength in the center of object i . Due to the linearity of the Eq. (1), \mathbf{H}_i can be expressed as

$$\mathbf{H}_i = \sum_{k=1}^N \mathbf{Q}_{ik} \mathbf{M}_k + \mathbf{H}_{\text{ex}i}, \quad i = 1, 2, \dots, N, \quad (9)$$

where $\mathbf{H}_{\text{ex}i}$ is an external field in the center of the object i . \mathbf{Q}_{ik} with $i,k=1,2,\dots,N$ gives the Interaction Matrix. For each particular i and k , \mathbf{Q}_{ik} itself is a 3 x 3 matrix. In the simplest case of a rectangular parallelepiped with no symmetries its components are defined by the Eqs. (2) - (4). If a space transformation with some multiplicity m was applied to the source object k , then \mathbf{Q}_{ik} is the sum of m contributions deduced from Eqs. (2) - (4) after series of transformations of the observation point and field.

For each of the N objects, the magnetization \mathbf{M}_i is related to the field strength \mathbf{H}_i by a material relation:

$$\mathbf{M}_i = \mathbf{f}_i(\mathbf{H}_i), \quad i = 1, 2, \dots, N. \quad (10)$$

Eqs. (9) - (10) describe the problem. They can be solved by an iterative procedure that must be properly

selected to ensure rapid convergence. We use the following procedure where the magnetization $\mathbf{M}_{i,p}$ and field strength $\mathbf{H}_{i,p}$ in each object $i = 1, 2, \dots, N$ at iteration p are deduced from the collection of $\mathbf{M}_{i,p-1}$ and $\mathbf{H}_{i,p-1}$ derived for all the objects at the iteration $p-1$ by:

$$\begin{aligned}\tilde{\mathbf{H}}_{\text{ex},p} &= \mathbf{H}_{\text{ex}} + \sum_{k=1}^{i-1} \mathbf{Q}_{ik} \mathbf{M}_{k,p} + \sum_{k'=i+1}^N \mathbf{Q}_{ik'} \mathbf{M}_{k',p-1}, \\ \mathbf{H}_{i,p} &= [\mathbf{E} - \mathbf{Q}_{ii} \chi_i(\mathbf{H}_{i,p-1})]^{-1} (\tilde{\mathbf{H}}_{\text{ex},p} + \mathbf{Q}_{ii} \mathbf{M}_{r,i}), \\ \mathbf{M}_{i,p} &= \mathbf{f}_i(\mathbf{H}_{i,p}),\end{aligned}\quad (11)$$

where \mathbf{E} is the 3×3 unit matrix, the function $\chi_i(\mathbf{H})$ gives the local susceptibility tensor, $\mathbf{M}_{r,i}$ is the remanent magnetization in the object i . For a linear material, χ_i does not depend of \mathbf{H} . For a non-linear isotropic one with the material relation being $\mathbf{M} = f_i(|\mathbf{H}|) \mathbf{H}/|\mathbf{H}|$,

$$\chi_i(\mathbf{H}) = \begin{cases} [f_i(|\mathbf{H}|)/|\mathbf{H}|] \mathbf{E}, & |\mathbf{H}| \neq 0, \\ f_i'(0) \mathbf{E}, & |\mathbf{H}| = 0, \end{cases} \quad (12)$$

$\mathbf{M}_{r,i} = \mathbf{0}$.

The iterations are stopped when the magnetizations $\mathbf{M}_{i,p}$ ($i = 1, 2, \dots, N$) are stable to a specified precision. For almost all cases of linear and non-linear materials that we have studied and all sorts of subdivisions, the iterative solution (11) - (12) was found to be fast and stable such that no additional relaxation parameter was needed.

3 COMPARISON WITH MEASUREMENTS

The Radia code has been systematically and quite successfully used in the past few years for the design of Insertion Devices at ESRF. Figure 1 presents a comparison of a peak field and field integral measured on a real device and computed with Radia. For almost any ID that we have designed, the computed peak field agrees with the measurement within 2%. The discrepancies come essentially from partially unknown material properties.

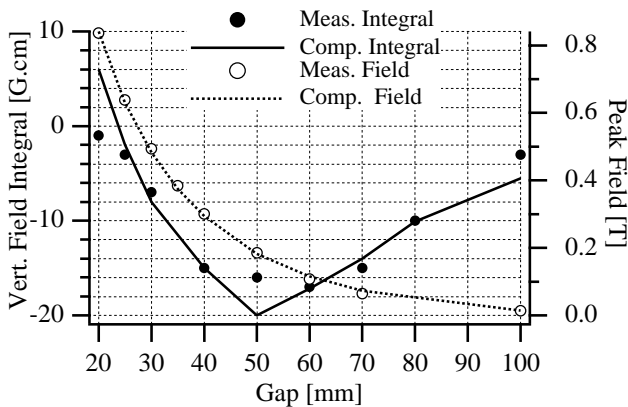


Figure 1: Comparison of the peak field and field integral predicted by Radia and measured on a 70 mm period hybrid wiggler. The computed and measured small variations of the field integral with the gap are in good qualitative agreement.

4 COMPARISON WITH FEM CODES

The proposed approach presents a number of advantages over FEM codes.

- + Geometries opened to infinity are more easily simulated.
- + The geometry creation requires only a few lines of the Mathematica code.
- + For the same precision, the number of elements and CPU time required to compute the central field of an undulator is roughly 20 times smaller in our approach than with second order tetrahedron elements in FEM codes. For example, the peak field of a hybrid undulator can be obtained with a precision better than 1% in less than 10 seconds on a PowerMac 4400, while it would take a few minutes with an FEM code.
- + The precision of the field integral computation only depends on the level of segmentation of the iron, but not on the field sampling and boundary conditions applied at infinity as is the case with FEM codes. Taking the example of a hybrid 2T wiggler, we obtain reliable estimation for the field integrals within several minutes of CPU time (PowerMac 4400) which is extremely difficult (may be impossible) with an FEM code. The design of termination of a hybrid wiggler can therefore be optimized over one night by a systematic exploration of a few geometrical parameters.

There are nevertheless some weak points as well.

- In the present version of the code, the memory required scales proportionally to the square of the number of elements. For example, one roughly needs 100M (25M) bytes of memory to solve a geometry made of 1000 (500) elements. We plan to improve this in a future version.
- Another drawback is a discontinuity in the value of the magnetic field inside subdivided iron objects, occurring at the boundary between any adjacent sub-objects. It is harmless if one is interested in the field sufficiently far outside the iron. This will also be improved in future versions.

REFERENCES

- [1] Mathematica is a registered trademark of Wolfram Research, Inc.
- [2] For basics of the FEM and BIM see, for example, T. Tortschanoff: 'Survey of Numerical Methods in Field Calculations', IEEE Trans. on Magnetics, vol. MAG-20, No.5, 1984, p.1912.
- [3] FLUX3D, Copyright Cedrat Recherche, CEDRAT SA, 10, Chemin du Pre Carre ZIRST 4301 38943 MEYLAN Cedex FRANCE.
- [4] For more information see "<http://www.esrf.fr/machine/support/ids/Public/index.html>".
- [5] L.K. Urankar: 'Vector Potential and Magnetic Field of Current-Carrying Finite Arc Segment in Analytical Form, Part III: Exact Computation for Rectangular Cross Section', IEEE Trans. on Magnetics, vol. MAG-18, No.6, p.1860.