

A three-dimensional magnetostatics computer code for insertion devices

Oleg Chubar,* Pascal Elleaume and Joel Chavanne

ESRF, BP 220, F-38043 Grenoble CEDEX, France.
E-mail: chubar@esrf.fr

(Received 4 August 1997; accepted 13 October 1997)

RADIA is a three-dimensional magnetostatics computer code optimized for the design of undulators and wigglers. It solves boundary magnetostatics problems with magnetized and current-carrying volumes using the boundary integral approach. The magnetized volumes can be arbitrary polyhedrons with non-linear (iron) or linear anisotropic (permanent magnet) characteristics. The current-carrying elements can be straight or curved blocks with rectangular cross sections. Boundary conditions are simulated by the technique of mirroring. Analytical formulae used for the computation of the field produced by a magnetized volume of a polyhedron shape are detailed. The *RADIA* code is written in object-oriented C++ and interfaced to *Mathematica* [*Mathematica* is a registered trademark of Wolfram Research, Inc.]. The code outperforms currently available finite-element packages with respect to the CPU time of the solver and accuracy of the field integral estimations. An application of the code to the case of a wedge-pole undulator is presented.

Keywords: insertion devices; undulators; wigglers; computer code.

1. Principle

Contrary to the majority of three-dimensional magnetostatics computer codes which are based on the finite-element method, *RADIA* uses a boundary integral method. A brief description of the basics of the finite-element and boundary integral methods can be found, for example, in the overview by Tortschanoff (1984). The general features and some details of the computation methods used in *RADIA* have been published recently (Elleaume *et al.*, 1997). In this paper, the basic principles of the code are briefly recalled and the method of computation of the field produced by a uniformly magnetized polyhedron is presented.

1.1. Problem description

To describe a problem, a user of *RADIA* creates different types of objects and links them together properly. The basic type of object is a source object capable of creating a magnetic field. This includes magnetized volumes, current-carrying coils and container objects. The volumes can be divided into a number of smaller volumes for which uniform magnetization is assumed. Another type of object describes the magnetic material properties through the magnetization *versus* field-strength law. Linear anisotropic (such as NdFeB or SmCo) and non-linear isotropic (such as iron) magnetic materials are supported. The next type of object includes space transformations such as translation, rotation or plane symmetry. Planar boundary conditions are simulated using the mirroring technique. The use of such mirrors

allows a dramatic reduction in the memory requirements and shortens the CPU time needed for the solution.

1.2. Solving the problem

An essential step in solving the problem consists of computing a so-called interaction matrix, which is, for N volumes with unknown magnetization, a $3N \times 3N$ matrix. Multiplication of the interaction matrix by a $3N$ vector representing magnetization in each volume gives a $3N$ vector representing the field produced by all the volumes in the centre of each individual volume. The final magnetization values in all the volumes are obtained by applying a proper relaxation scheme which involves the interaction matrix, material properties of the volumes and external field sources. Details of the relaxation scheme used in *RADIA* have been published recently (Elleaume *et al.*, 1997). Following the relaxation procedure, the magnetization in each volume is known and the field or field integral anywhere in space is obtained by adding together the contributions from all the magnetized volumes and external field sources. The final accuracy of the computed field (field integral) depends only on the level of subdivision of the magnetized volumes.

2. Field produced by a volume with uniform magnetization

2.1. Generalities

Let V' be a volume bound by the closed surface S' and magnetized according to $\mathbf{M}(\mathbf{r}')$. This volume produces a magnetic field strength $\mathbf{H}(\mathbf{r}_0)$ at the observation point \mathbf{r}_0 given by the well known formulae of magnetostatics (in SI units)

$$\mathbf{H}(\mathbf{r}_0) = \frac{1}{4\pi} \int_{V'} \frac{(\mathbf{r}' - \mathbf{r}_0) \nabla \mathbf{M}}{|\mathbf{r}' - \mathbf{r}_0|^3} dV' - \frac{1}{4\pi} \oint_{S'} \frac{(\mathbf{r}' - \mathbf{r}_0)(\mathbf{M} \mathbf{n}_{S'})}{|\mathbf{r}' - \mathbf{r}_0|^3} dS', \quad (1)$$

where $\mathbf{n}_{S'}$ is a unit vector normal to the surface S' and pointing outside the volume, and \mathbf{r}' is a point inside the volume or on the surface. If \mathbf{M} is uniform inside the volume, the first integral in (1) vanishes and we obtain the following matrix representation

$$\mathbf{H}(\mathbf{r}_0) = \mathbf{Q}(\mathbf{r}_0) \mathbf{M}, \quad (2)$$

$$\mathbf{Q}(\mathbf{r}_0) = \frac{1}{4\pi} \oint_{S'} \frac{(\mathbf{r}_0 - \mathbf{r}') \otimes \mathbf{n}_{S'}}{|\mathbf{r}_0 - \mathbf{r}'|^3} dS', \quad (3)$$

where \otimes is the symbol of the dyadic (or tensor) product of three-dimensional vectors.† The 3×3 matrix \mathbf{Q} is a geometrical entity which only depends on the observation point and the shape of the surface S' .

From (2) and (3) one derives the following expression for the field integrals along a straight line parallel to a unit vector \mathbf{v} and passing through the point \mathbf{r}_0 ,

$$\mathbf{I}(\mathbf{r}_0, \mathbf{v}) \equiv \int_{-\infty}^{\infty} \mathbf{H}(\mathbf{r}_0 + \mathbf{v}l) dl = \mathbf{G}(\mathbf{r}_0, \mathbf{v}) \mathbf{M}, \quad (4)$$

$$\mathbf{G}(\mathbf{r}_0, \mathbf{v}) = \frac{1}{2\pi} \oint_{S'} \frac{\{[(\mathbf{r}' - \mathbf{r}_0) \times \mathbf{v}] \times \mathbf{v}\} \otimes \mathbf{n}_{S'}}{|\mathbf{r}' - \mathbf{r}_0 + \mathbf{v}|^2} dS', \quad (5)$$

where the matrix \mathbf{G} is fully defined by the line and the shape of the surface S' .

† The dyadic product of three-dimensional vectors \mathbf{a} and \mathbf{b} is a 3×3 matrix such that for any three-dimensional vector \mathbf{c} , $(\mathbf{a} \otimes \mathbf{b}) \mathbf{c} \equiv \mathbf{a}(\mathbf{b} \cdot \mathbf{c})$.

For current-carrying volumes, a representation of the field and field integrals similar to (2)–(5), with the uniform magnetization replaced by the uniform current density, can be obtained. The corresponding matrices are derived from the Biot–Savart law.

For a wide variety of shapes, both in the case of the volumes with uniform current density and uniform magnetization, analytical expressions for the field and field integral can be found. Moreover, often the components of the matrices \mathbf{Q} and \mathbf{G} can be expressed through elementary transcendental functions for which efficient computation algorithms are well known. Several authors have proposed analytical formulae for the field induction and vector potential produced by the current-carrying volumes of various shapes (Urankar, 1982; Ciric, 1992). The formulae for the field produced by a uniformly magnetized rectangular parallelepiped are well known (see, for example, Marechal *et al.*, 1990). The latter particular case was treated in a previous publication on *RADIA* (Elleau *et al.*, 1997). In the next section, an expression for the matrix \mathbf{Q} is given for a more general case when the shape is a polyhedron (see Fig. 1).

2.2. Field produced by a uniformly magnetized polyhedron

Let us consider an arbitrary polyhedron bound by N faces. Each face, labelled by the index $\sigma = 1, 2, \dots, N$, is a planar polygon with vertex points $\mathbf{r}'_{\sigma,p}$, $p = 1, 2, \dots, N_{\sigma}$. The polyhedron is uniformly magnetized according to \mathbf{M} . The magnetic field produced by such a polyhedron at the observation point \mathbf{r}_0 is described by (2) in which the matrix \mathbf{Q} can be expressed as a sum of contributions from all the faces

$$\mathbf{Q} = \frac{1}{4\pi} \sum_{\sigma=1}^N \mathbf{T}_{\sigma} \mathbf{Q}_{\sigma} \otimes \mathbf{n}_{\sigma}, \quad (6)$$

$$\mathbf{Q}_{\sigma} = \sum_{\substack{p=1 \\ x_{\sigma,p+1} \neq x_{\sigma,p}}}^{N_{\sigma}} \mathbf{q}[x_{\sigma,p}, x_{\sigma,p+1}, (y_{\sigma,p+1} - y_{\sigma,p}) / (x_{\sigma,p+1} - x_{\sigma,p}), \\ (x_{\sigma,p+1} y_{\sigma,p} - x_{\sigma,p} y_{\sigma,p+1}) / (x_{\sigma,p+1} - x_{\sigma,p}), z_{\sigma}], \quad (7)$$

where $\mathbf{n}_{\sigma} = (n_{x\sigma}, n_{y\sigma}, n_{z\sigma})$ is a unit vector normal to the face σ and pointing outside the volume. \mathbf{T}_{σ} is a 3×3 rotation matrix that transforms the laboratory Cartesian frame into a local frame of the face σ , in which the unit vector parallel to the z axis $(0, 0, 1)$ is normal to the face σ and points outside the volume. It can be expressed as

$$\mathbf{T}_{\sigma} = \begin{bmatrix} n_{y\sigma}^2 (n_{z\sigma} + 1)^{-1} + n_{z\sigma} & -n_{x\sigma} n_{y\sigma} (n_{z\sigma} + 1)^{-1} & n_{x\sigma} \\ -n_{x\sigma} n_{y\sigma} (n_{z\sigma} + 1)^{-1} & n_{x\sigma}^2 (n_{z\sigma} + 1)^{-1} + n_{z\sigma} & n_{y\sigma} \\ -n_{x\sigma} & -n_{y\sigma} & n_{z\sigma} \end{bmatrix}. \quad (8)$$

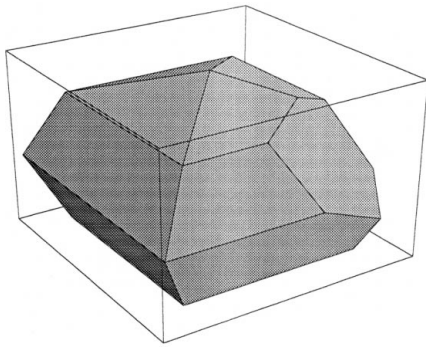


Figure 1
The polyhedron is defined as a volume bound by a set of planar polygons.

$\mathbf{Q}_{\sigma} \otimes \mathbf{n}_{\sigma}$ in (6) is the contribution to the matrix \mathbf{Q} from the face σ computed in the local frame of the face σ . $x_{\sigma,p}$, $y_{\sigma,p}$, z_{σ} are components of the vector $\mathbf{r}_{\sigma,p} = (x_{\sigma,p}, y_{\sigma,p}, z_{\sigma})$ defined by

$$\mathbf{r}_{\sigma,p} = \mathbf{T}_{\sigma}^{-1}(\mathbf{r}'_{\sigma,p} - \mathbf{r}_0). \quad (9)$$

The expressions for $\mathbf{r}_{\sigma, N_{\sigma}+1}$ required in (7) are given by the cyclic identity $\mathbf{r}_{\sigma, N_{\sigma}+1} \equiv \mathbf{r}_{\sigma, 1}$. The vector function of five scalar arguments $\mathbf{q}(x_1, x_2, a, b, z)$ is defined by the following set of auxiliary functions:

$$\mathbf{q}(x_1, x_2, a, b, z) = [q_x(x_1, x_2, a, b, z), q_y(x_1, x_2, a, b, z), \\ \times q_z(x_1, x_2, a, b, z)],$$

$$q_x(x_1, x_2, a, b, z) = -aq_y(x_1, x_2, a, b, z) \\ + \ln[f_1(x_1, a, b, z)/f_1(x_2, a, b, z)],$$

$$q_y(x_1, x_2, a, b, z) = (a^2 + 1)^{-1/2} \\ \times \ln[f_2(x_1, a, b, z)/f_2(x_2, a, b, z)],$$

$$q_z(x_1, x_2, a, b, z) = \tan^{-1}[f_3(x_1, a, b, z)/f_4(x_1, a, b, z)] \\ - \tan^{-1}[f_3(x_2, a, b, z)/f_4(x_2, a, b, z)] \\ + \begin{cases} C(x_1, x_2, a, b, z), & d(a, b, z) > 0 \\ 0, & d(a, b, z) \leq 0, \end{cases}$$

where

$$C(x_1, x_2, a, b, z) = \pi \operatorname{sgn}[(x_2 - x_1)z] \\ \times \sum_{k=1}^2 (-1)^k \beta[\xi_k(a, b, z), a, b, z] \\ \times \theta\{[x_1 - \xi_k(a, b, z)][\xi_k(a, b, z) - x_2]\} \\ \times \operatorname{sgn}\{[b - a\xi_k(a, b, z)]f_3[\xi_k(a, b, z), a, b, z]\}, \\ f_1(x, a, b, z) = ax + b + \rho(x, a, b, z), \\ f_2(x, a, b, z) = ab + (a^2 + 1)x \\ + (a^2 + 1)^{1/2} \rho(x, a, b, z), \\ f_3(x, a, b, z) = 2(x^2 + z^2)abz^2 \\ + (az^2 + bx)(a^2z^2 + b^2)f_1(x, a, b, z), \\ f_4(x, a, b, z) = [(a^2z^2 - b^2)(x^2 + z^2) \\ + (ax - b)(a^2z^2 + b^2)f_1(x, a, b, z)]z, \\ \beta(x, a, b, z) = \theta\{(b - ax)[(a^2z^2 - b^2)(x^2 + z^2) \\ + (a^2x^2 - b^2)(a^2z^2 + b^2)]\}, \\ \rho(x, a, b, z) = [x^2 + (ax + b)^2 + z^2]^{1/2}, \\ \xi_k(a, b, z) = [(a^2z^2 + b^2)ab + (-1)^{k+1}|a^2z^2 - b^2| \\ \times d(a, b, z)^{1/2}] / [4a^2b^4 - (a^2 + 1) \\ \times (a^2z^2 - b^2)^2], \\ d(a, b, z) = [(a^2 + 1)z^2 + b^2][4(a^2z^2 + b^2)a^2b^2 \\ - (a^2z^2 - b^2)^2], \quad (10)$$

where one makes use of the logarithm and arctangent of a real argument, step function $\theta(x) \dagger$ and the sign function $\operatorname{sgn}(x)$. Symbolic calculation features built into *Mathematica* were applied when deriving (6)–(10) from (3). In the *RADIA* code, (6)–(10) are used for the construction of the interaction matrix required for the relaxation, as well as for the field computation.

$\dagger \theta(x) \equiv 1$ for $x \geq 0$, or 0 for $x < 0$.

For the field integral along a straight line, formulae similar to (6)–(10) were obtained from (5) and implemented in the code.

In *RADIA*, two particular cases of polyhedron are treated separately: an extruded polygon (or a straight prism with polygonal base) and a rectangular parallelepiped. For these shapes one can further simplify (6)–(10) in order to reduce the CPU time.

3. Application to a wedge-pole undulator

This section presents an example in which the *RADIA* code is used to model a 35 mm-period hybrid wedge-pole undulator. In this example, magnetized volumes of polyhedron shape are used extensively. Fig. 2 presents a three-dimensional plot of the geometry under study, produced by using a combination of *Mathematica* and *RADIA* function calls. Since the geometry is symmetrical with respect to mirror planes $x = 0$, $y = 0$ and $z = 0$, only the 1/8 part of the total geometry (Fig. 2b) and boundary conditions at the mirror planes were really defined. The boundary conditions enforce the field strength at $x = 0$ ($y = 0$) to be parallel to the plane $x = 0$ ($y = 0$), the field at $z = 0$ to be perpendicular to the plane $z = 0$. As a result of these boundary conditions, the number of degrees of freedom and the CPU time needed to solve the problem have been reduced by a factor of eight and the memory required reduced by a factor of 64.

The poles and magnets are subdivided into a number of smaller volumes. The magnetization in iron poles being less uniform than in magnet blocks, a finer subdivision is used for the poles than for the magnets. The subdivision is indicated in Fig. 2 by lines drawn at the surface of the poles and magnets.

For an absolute accuracy of the undulator peak field better than 1%, one requires 45 s of CPU time and 2.5 MB of memory using a PC with Intel Pentium 133. As a result of the solution, the magnetization in each volume of the poles and magnets is known. The field and field integral can then be quickly computed anywhere in space by adding together the contributions from all the volumes. Testing the absolute accuracy can be performed by refining the segmentation of poles and magnets and re-solving. Fig. 3 presents a plot of the vertical field profile on the axis of the electron beam ($x = z = 0$) as a function of the longitudinal coordinate y .

4. *RADIA* versus finite-element codes

For a number of undulator and wiggler structures, we have compared the results produced by *RADIA* with those from the finite-element code *FLUX3D* (copyright Cedrat Recherche; <http://www.cedrat-grenoble.fr>). Once the necessary precautions, such as a sufficiently fine meshing in *FLUX3D* and sufficient subdivision in *RADIA*, have been taken, both codes always agree to better than 1% in the peak field. Having checked the consistency of the results from both codes, we have noticed a number of differences.

(i) Geometries opened to infinity are more easily simulated with *RADIA*.

(ii) The precision of the results produced by *RADIA* only depends on the refinement of subdivision of the poles and magnets. With the finite-element codes one needs to mesh the whole space (including air) up to infinity and the estimation of the dominant source of errors is difficult (pole, magnet, air, boundary condition at infinity *etc.*).

(iii) There is an area where *RADIA* is particularly successful: the prediction of the field integral from an undulator or wiggler. Comparison of the field integral predicted by *RADIA* with that measured on a real device results in concurrence (Elleaume *et al.*, 1997), allowing the use of *RADIA* for a proper optimization of the terminations of undulators and wigglers. Reliable prediction of the field integral from a finite-element code has never been reported. This is not surprising in view of the fact that the accuracy in the field integral is very sensitive to the truncation at infinity, the step of the numerical integration in addition to the precision required for the computation of the field at each

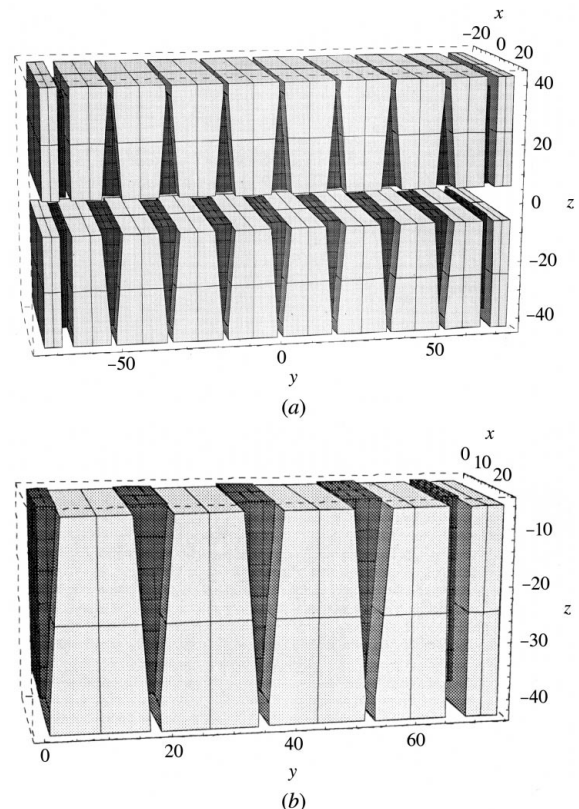


Figure 2 Three-dimensional geometry plot of a 35 mm-period hybrid wedge-pole undulator simulated by *RADIA*: (a) total geometry; (b) the part of the geometry really defined before applying boundary conditions.

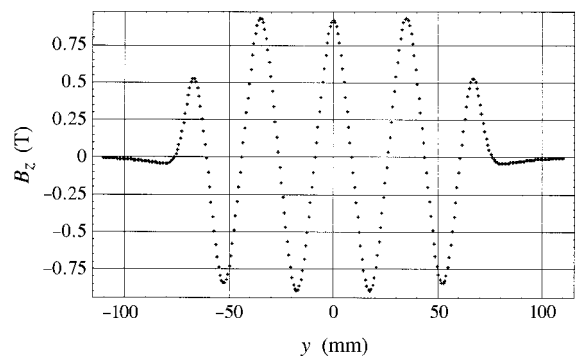


Figure 3 Vertical field profile on the axis of the electron beam ($x = z = 0$) as a function of the longitudinal coordinate y for the structure shown in Fig. 2.

point. On the other hand, *RADIA* is insensitive to the truncation at infinity and to the step size since it uses analytical formulae for the field integral.

(iv) An experienced user solves the peak field of an undulator in 20 times less CPU time with *RADIA* than with *FLUX3D* for the same accuracy. The time required to set up a parametrized model of the geometry is also much shorter with *RADIA*. A smaller number of comparisons have shown that the situation is similar (precision and CPU time) when comparing *RADIA* with *TOSCA* (copyright Vector Fields; <http://www.vector-fields.co.uk>).

(v) At this time, the most important drawback to *RADIA* is the discontinuity of the field inside the iron at the border between individual volumes introduced by the subdivision. It originates from the assumption of a uniform magnetization in each individual volume. We are investigating methods to improve this. Note

that this drawback is harmless for the precision of the field computation outside the magnetized volumes.

Versions of the *RADIA* code for PowerMac (Mac OS) and PC with Intel Pentium processor (Windows95/NT) are available at <http://www.esrf.fr/machine/support/ids/Public/index.html>.

References

- Ciric, I. R. (1992). *IEEE Trans. Magn.* **28**(2), 1064–1067.
Elleume, P., Chubar, O. & Chavanne, J. (1997). *Proc. IEEE*, PAC-97, 9P27. In the press.
Marechal, X., Chavanne, J. & Elleume, P. (1990). ESRF-Synchrotron Radiation/ID-90-43. ESRF, Grenoble, France.
Tortschanoff, T. (1984). *IEEE Trans. Magn.* **20**(5), 1912–1917.
Urankar, L. K. (1982). *IEEE Trans. Magn.* **18**(6), 1860–1867.