MAGNUS: A COMPUTER PROGRAM FOR 3-D MAGNET DESIGN

S. Pissanetzky*, M. Fan* G.R. Kalbfleish†, and E. Whalen†
*Texas Accelerator Center
2319 Timberloch Place, Bay H
The Woodlands, TX 77380
† University of Oklahoma
440 W. Brooks
Norman, OK 73019

Abstract

MAGNUS is a Finite Element program for the calculation of magnetic fields in 3-D. MAGNUS is very versatile and has been successfully used for the design of a variety of magnets, including accelerator magnets, ends of SSC dipoles, quadrupoles and sextupoles, shielding problems, recording heads, etc. The mesh generator KUBIK uses a modular concept: modules are separately defined and then assembled into the final 3-D structure. MAGNUS has a library of conductor elements, out of which any 3-D conductor shape can be constructed. Conductor specification is independent of the mesh. It also has a library of magnetic properties, that includes several US and Japanese steels. The solver is very efficient and runs on VAX or CRAY. The postprocessor EPILOG runs interactively on VAX and allows the calculation of many derived quantities: Z-averaged harmonics, energy, field integrals, tables, plots, etc. Preformatted data exist for common magnets to simplify data input. MAGNUS is interfaced with the international Graphics Kernel System GKS for graphics and is available on the NMFECC network.

Introduction

A problem of nonlinear magnetostatics is defined in a three-dimensional domain that contains current-carrying conductors, nonlinear magnetic materials and permanent magnets, with suitable boundary conditions on the boundaries. The program MAGNUS can solve such a problem using the two-scalar-potentials formulation and the finite element method. The method [1], the theory [2] and the application to magnetically anisotropic problems [3] have been described elsewhere. So also was preprocessing and mesh generation [4-5], and results and measurements for some accelerator magnets and other magnetic devices designed with MAGNUS have been published [6-9]. Some of the sophisticated computational techniques used in MAGNUS have been theoretically described in a book [10]. Postprocessing has been considerably improved, and is one of the subjects of this paper.

MAGNUS is the finest available program for 3-D nonlinear magnetostatics. It is a professional, fully tested, state-of-the-art, well documented program. Sophisticated mathematics and careful programming have been used to improve accuracy, efficiency and simplicity of use. There are of course, limitations, and, like any good program, development of MAGNUS will never end.

Description and Practical Use of MAGNUS

The mathematical specification of a problem of nonlinear magnetostatics requires a geometric description of the problem domain and the magnetic bodies inside it, the geometric description and currents of the conductors in space, and a definition of the magnetic properties and/or permanent magnetization of the magnetic materials involved in the problem. Boundary conditions must also be specified on the boundaries. Since MAGNUS uses the finite element method, it is also necessary to specify a mesh of finite elements.

Mesh generation and geometric description are associated in MAGNUS, and are both performed by the preprocessor KUBIK. KUBIK employs a modular "machine shop" approach. Simple objects called modules are described in space, each with its own mesh inside. The modules are then joined together until the final structure is obtained. This method placed no limitations on the complexity of the problem geometry, except as dictated by the computer resources, and gives the user complete control over the characteristics of the mesh. Bodies and modules are not the same thing, and a body, or region, in the problem domain can be described independently of the modules. KUBIK is used by means of commands, and options are available for the user to define the modules and to assemble them in space as required. KUBIK is interfaced with the international Graphics Kernel System GKS, and views of the structure can be obtained at any stage on any of the graphic devices supported by GKS. KUBIK also has its own basic graphics facility.

Conductors

There is a library of conductor elements available in MAGNUS. The library includes filament straight conductors of finite, semi-infinite or infinite length; filament arcs and loops; solid bars of finite, infinite or semi-infinite length with various cross-sections; solid arcs and solenoids, etc. A given conductor in space is represented by one or a few elements of the library, and even a complex conductor shape in space can be approximated by several elements. Conductors are independent of the mesh and of each other. Commands are available for the user to take an element from the library, place it in the correct position in space and specify its current in both magnitude and direction. Additional commands take advantage of symmetries, and allow the user to specify conductors by the reflection, rotation or translation of previously specified conductors, with or without current reversal. Plots or views of the conductors can be obtained at any stage.

Magnetic Properties and Permanent Magnets

A magnetic body is a region in space described in MAGNUS by set of finite elements. It can have any shape and is independent of the modules used to construct the mesh, although of course the mesh geometry must be prepared in such a way that magnetic bodies can be defined. Magnetic bodies are then defined by means of commands which determine the region by specifying its boundaries or other geometric or topological properties. Many such regions can be defined and each of them can be assigned different nonlinear magnetic properties, or become a permanent magnet by assigning a given magnetization to it.

There is a library of magnetization tables available in MAGNUS. The library contains several american steels at room temperature and at liquid helium temperature, some Japanese steels, and ideal materials such as pure iron or pure nickel. The user can also provide additional magnetization tables. Any of the internal tables or the additional tables can be assigned to any of the geometric regions, causing the region to become a nonlinear magnetic material. In the course of the iterative solution of the problem, the magnetization at each point of a magnetic material is considered to be a function of the total field H at that point. The function is defined by the table and by sophisticated and accurate interpolation techniques [11] between the points of the table. There are commands that allow the user to print a table as is, or to print interpolated values and verify the quality or details of the interpolation. Interpolation accuracy may be critical for problems where saturation effects are important.

Permanent magnets are defined in much the same way, except that a permanent magnetization rather than a table is assigned to the corresponding region of space.

The Solver

Once the geometry of the problem, the conductors and the magnetic properties have been specified, the user specifies the boundary conditions. This completes the mathematical specification of the problem. The user is further required to provide a few additional parameters, like the accuracy expected from the solution and the maximum number of iterations allowed. Then the solver starts execution.

The solver is a preconditioned conjugate gradient one that uses sparse matrices and only stores and operates upon the nonzero elements of the matrix. All data structures and algorithms have been developed and optimized for positive definite symmetric matrices, like the ones found in MAGNUS, which makes this solver to be among the best available in the world for sequential in-core processing. At the beginning of each iteration, the magnetization in the magnetic materials is assumed to be a known function of x, y and z. The system of finite element equations is assembled and solved, new fields are obtained, and a new magnetization is calculated, again as a function x, y and z. This process is repeated until convergence is achieved. Convergence is monitored by two parameters: the root mean square difference between two successive solutions, and the average permeability of the magnetic materials. This last parameter is very sensitive because it is related to the total magnetic energy, which must consistently decrease as the correct solution is approached. The solver stores all intermediate results, and can be restarted for more iterations if greater accuracy is required.

Postprocessing

The result from the solver is the complete solution to the problem, which consists of two continuous functions, the two scalar potentials, defined at each and every point of the corresponding domain. The purpose of the postprocessor EPILOG is to use the solution to calculate secondary quantities, or to generate output in the form of tables or plots. EPILOG is operated by means of commands, which are all independent and interchangeable. This makes it possible for the user to examine fields and potentials or to calculate magnetic parameters at the time they are needed and without having to run MAGNUS again, rather than having to specify the output at the time the input is prepared. Solutions to different problems or different cases can be stored and examined when convenient, thus providing a very valuable data base of magnetic solutions.

EPILOG can produce tables of potential, field H or induction B at selected points of space, tables of magnetic permeability, and many more. It can also plot field lines, equipotential surfaces, etc. EPILOG can also calculate harmonic coefficients, magnetic energy, forces, line integrals, magnetic flux, etc. Some of these facilities are described below.

Z-Averaged Harmonic Coefficients

In the design of accelerator magnets, it is frequently necessary to deal with harmonic expansions of a two-dimensional magnetic potential $\Phi'(x,y)$ or two-dimensional field $H'(x,y)$, obtained as an average of the corresponding 3-D potential or field over a z interval defined by two values, Z_1 and Z_2 of z.

$$\Phi'(x,y) = \frac{1}{Z_2 - Z_1}\int_{Z_1}^{Z_2} \Psi(x,y,z)dz$$

$$H_x'(x,y) = \frac{1}{Z_2 - Z_1}\int_{Z_1}^{Z_2} H_x(x,y,z)dz$$
$$H_y'(x,y) = \frac{1}{Z_2 - Z_1}\int_{Z_1}^{Z_2} H_y(x,y,z)dz$$

If R_N is the normalization radius, the expansions are, in terms of cylindrical coordinates ρ, θ:

$$\Phi' = \sum_{n=1}^{\infty}\left(\frac{\rho}{R_N}\right)^n (A_n \sin n\theta + B_n \cos n\theta)$$
$$H_x' = H_{x0}' + \sum_{n=1}^{\infty}\left(\frac{\rho}{R_N}\right)^n (b_n \sin n\theta + a_n \cos n\theta)$$
$$H_y' = H_{y0}' + \sum_{n=1}^{\infty}\left(\frac{\rho}{R_N}\right)^n (b_n \cos n\theta - a_n \sin n\theta)$$

where the coefficients are related by

$$b_n = -\frac{n+1}{R_N}A_{n+1}$$
$$a_n = -\frac{n+1}{R_N}B_{n+1}$$

Command ZAV MULTIPOLES will automatically calculate both sets of coefficients by direct numerical integration of $\Psi(x,y,z)$, giving the user many options such as: specifying the averaging interval Z_1, Z_2; specifying the integration and normalization radii; and using various symmetries and boundary conditions .

Energy, Forces and Magnetic Flux

Command ENERGY will calculate the total magnetic energy in each region and material by direct integration of the fields, in joules. Command FORCE calculates the total magnetic force.

$$F_i = \frac{1}{\mu_0}\oint_S \left(B_i \mathbf{B}.\mathbf{n} - \frac{1}{2}B^2 n_i\right) dS$$

on a closed surface S. Shapes currently available for S include a rectangular box and a cylinder, and other shapes are being inplemented. A particularly important one is the force on a given conductor. Command B FLUX calculates the magnetic flux

$$\Phi = \int_S B_n dS$$

for a given open surface S in space. Available shapes include a flat rectangle, a flat circle or circular ring and a cylinder. Because of the relation $\Phi = Li$, flux is the primary quantity to calculate the inductance of a circuit.

Other Commands

Command LINE INT H calculates

$$\int_{line} (H_x dx + H_y dy + H_z dz)$$

for a variety of lines in space. There is a command that will calculate the field of the conductors alone (assuming no magnetic materials) at points given by the user. Another command can plot the conductors as seen from a given point of view. Command SHOW PARAM will print all relevant parameters of the current problem. There are commands to plot equipotential surfaces or field lines in space. PRINT commands exist which print values of the potential and fields at points inside a box, or along a line or an arc in space specified by the user, or values of the permeability μ in a specified region. EPILOG is in a continuous state of development, and new commands are added as required by users.

Performance

For a typical medium-size problem with 10,000 elements, KUBIK runs usually in 10 to 15 CPU minutes on VAX-780. MAGNUS, the solver, is where all heavy computation takes place: around 6 to 8 CPU hours on VAX-780, or 10 minutes on CRAY-2, or 20 minutes on CYBER-205, or 2 hours on VAX-8600 for 40 or 50 iterations, as usually necessary to achieve convergence. It is even conceivable to run MAGNUS on an IBM-PC AT with a floating point accelerator, and certainly on a Micro-VAX or a SUN computer. The programs are written in standard Fortran and the source code is provided, so that the user can implement them on any computer. EPILOG runs very fast, depending on the command. Most commands take a minute or two on VAX-780, except ENERGY, which requires extended integration and may take 1 or 2 hours. All programs are interfaced with GKS, and have, in addition, their own simple graphics package which supports devices like the terminal VT-240, the laser printer LN-03 and the dot-matrix printer LA-100. A library of preformatted data exists for common magnets to simplify data input.

References

[1] J. Simkin and C.W. Trowbridge. "Three-dimensional Nonlinear Electromagnetic Field Computations Using Scalar Potentials". IEEE Proc., vol. 127, pp. 368-374, 1980.

[2] S. Pissanetzky. "The New Version of the Finite Element 3-D Magnetostatics Program MAGNUS". Computational Electromagnetics. Z.J. Cendes (editor), Elsevier Science Publishers B.V. (North Holland), pp. 121-132, 1986.

[3] S. Pissanetzky. "Solution of Three-dimensional, Anisotropic, Nonlinear Problems of Magnetostatics Using Two Scalar Potentials, Finite and Infinite Multipolar Elements and Automatic Mesh Generation".IEEE Trans. on Magnetics, vol. MAG-18, pp. 346-350, March 1982.

[4] S. Pissanetzky. "KUBIK: An Automatic Three-dimensional Finite Element Mesh Generator". Int. J. Num. meth. Engng.,vol. 17, pp. 255-269, 1981.

[5] S. Pissanetzky. "Automatic Three-dimensional Finite Element Mesh Generation Using the Program KUBIK". Comp. Phys. Comm., vol. 32, pp. 245-265, 1984.

[6] S. Pissanetzky. "The Design of Superferric Magnets for the SSC and the New Program MAGNUS for Three-dimensional Magnetostatics". IEEE Trans. on Magn., vol. 21, pp. 2457-2460, 1985.

[7] H. Bingham, R. Carcagno, M. Davidson, H. Hinteberger, R. Huson, M. Kobayashi, W.N. MacKay, S. Pissanetzky, R. Rocha, W. Schmidt, and R. Stegman. "Design of a Superferric Dipole for the SSC". Proc. VI Latin American Congress on Comp. Meth. for Engineering, Panama', Argentina, Oct. 15-18, 1985. Comp. Mech. Int., Southampton, England, to appear.

[8] S. Pissanetzky. "Magnetic Design of a Flat-coil Cosine-theta Primary Correction Package". Texas Accelerator CenterThe Woodlands, Texas 77380, June 1986.

[9] R. Carcagno and S. Pissanetzky. "Numerical Modeling of a Bending Dipole Magnet With Magnetic Welds". Texas Accelerator Center report TAC 0912/85, 1985.

[10] S. Pissanetzky. "Sparse Matrix Technology". London: Academic Press, 1984. Translated into Russian.

[11] S. Pissanetzky. "The Interpolation of Magnetization Tables". COMPEL, Int. J. Comp. Math. in Electrical and Electronic Engng., vol. 5, pp. 41-56, 1986.