Title

COMPUTER SIMULATION OF MAGNETIC FIELDS

Author

Colonias, J.S.

Publication Date

1984-02-01

LBL-17348
Preprint c.2

Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

Computing Division

To be published in Industrial Research
and Development

COMPUTER SIMULATION OF MAGNETIC FIELDS

J.S. Colonias

February 1984

Prepared for the U.S. Department of Energy under Contract DE-AC03-76SF00098

To be published in Industrial Research & Development

COMPUTER SIMULATION OF MAGNETIC FIELDS*

John S. Colonias

Lawrence Berkeley Laboratory
University of California
Berkeley, California 94720

February 1984

* This work was supported by the U.S. Department of Energy under contract no. DE-AC03-76SF00098.

I. INTRODUCTION

Industrial applications of magnet technology have mushroomed in the last decade. Such applications have ranged widely and include exotic superconducting magnets such as those used in particle accelerators to magnetic actuators used in the design of computer disk units.

Such dramatic growth would not have been possible had it not been for the rapid advances in simulation models that approximate and solve the nonlinear Maxwell's equations that describe the magnetic fields generated from such magnets.

In this article we endeavor to describe this methodology and its application to various industrial processes. We will briefly describe the mathematical models and proceed to describe the importance of such modeling in terms of providing the engineer or scientist with an extremely valuable tool to aid in the optimization of any device that includes magnetic fields as a component. Examples of such applications will reinforce our believe that mathematical modeling is an important and indispensible part of an overall system design.

II. MATHEMATICAL MODELING

Even though it is not the intention of this article to outline the mathematical models used to simulate electromagnetic fields, nevertheless I think it is important for the magnetic field designer to be aware of their existence and not take a detached view and disentangle physical concepts from algebraic ones.

In general, the simulation of magnetic fields involves the solution of extremely complex equations (partial differential equations or integral equations) imposed on a grid on which the contemplated magnet geometry is outlined.

During the last decade, various formulations have been proposed and programmed. Generally, these formulations solve either a set of partial differential equations using a vector potential such as Eq. 1, or using a scalar potential such as shown in Eq. 2.

$$\nabla \cdot \frac{1}{\mu} A \cdot \vec{A} = -4 \Pi \vec{J} \qquad \text{Eq. 1}$$

where μ = permeability of material

$\vec{A}$ = vector potential

$\vec{J}$ = current density

$$\nabla^2 V^* = -\nabla \cdot \vec{M} \qquad \text{Eq. 2}$$

where $\nabla X \vec{M} = \vec{J}$

V^* = scalar potential

Formulations such as the above solve two-dimenstional problems, since, for example, the vector potential $\vec{A}$ appearing in Eq. 1 has only one non-zero component ($A_Z \neq 0$), thus necessitates less computer storage and produces results in a reasonable computer time.

Numerical methods developed to solve these equations involve their discretization and by means of a "finite difference" scheme the continuous field distribution problem is replaced by a discrete one.

Recently, however, "finite element" methods have become popular. These methods use the integral formulation approach and solve equations like Eq. 3, for the magnetization vector $\vec{M}$

$$\vec{M} = (\mu_r - 1)\left[M_0 H_0 - \frac{1}{4\Pi}\nabla \cdot \int_{\text{iron}} \frac{\vec{M} \cdot \vec{r}}{r^3}\, dV\right] \qquad \text{Eq. 3}$$

or similar expressions for the vector $\vec{H}$.

Algebraically, both methods may lead to similar computational schemes involving the solution of thousands of simultaneous equations.

Most recently, hybrid methods have been evolved which use both the differential and integral formulation in the same problem by solving the interior problem by differential equations while the boundary of the problem is solved by the integral formulation. Progress in arriving at better formulations is a continuous process that involves many researchers at various laboratories.

Depending on the degree of sophistication of the mathematical model, the resulting algorithm can solve two-dimensional or three-dimensional problems, problems containing iron (non-linear) or problems containing coil geometries only (linear problems). Table I lists some of the most important computer programs for magnetic field simulation.

The methods outlined above solve the "magneto-static" problem; that is, the problem during which time is still. Time-dependent magnetic field problems can also be solved by both finite difference and finite element methods. Such problems result also in complex equations that are solved iteratively.

The references at the end of this article enumerate chronologically some of the important papers in this field.

Table I

Name	Dimension	Method of Solution	No. of Points	Computer	Linear or Nonlinear
TRIM(1)	2*	Finite-diff. vector pot.	4000-10000	CDC,IBM,VAX	Both
EFFI(2)	3	Elliptic integrals	No limit	CDC,CRAY	Linear
POISSON(3)	2*	Finite-diff. vector pot.	4000-10000	CDC,IBM,VAX	Both
GFUN(4)	3	Integral magnetization		CDC,CRAY,IBM	Both
TOSCA(5)	3	Integral mod. scalar		CDC,CRAY,IBM	Both

* Axially symmetric geometries may also be considered.

III. USING A NUMERICAL MODEL

Why use a numerical model and not the real thing? Actually, a numerical model represents a real-life system and depending on how good the model is, it will represent accordingly the real-world system it attempts to simulate. In most cases, the model is cheaper to construct than the real thing, it is possible to experiment with, and parameters can be optimized to produce a better "real thing". The model we use to represent Maxwell's equation involves too many variables for the unaided human mind to juggle at once.

To provide some insight into the mechanics of using a mathematical model, we will assume the formulation shown by Eq. 1 and investigate in some detail the behavior of such a model.

The computer program written to solve this equation uses the finite difference formulation, which in essence replaces the partial differential equation, Eq. 1, by a set of difference equations applied over a set of points in the region of interest. This technique is perhaps the most popular numerical method for solving such equations and recently is being challenged by the finite-element method. The approximate solution to the continuous problem is obtained by solving a very large system of simultaneous linear equations and the resulting solution is usually the potential defined at a finite number of points, making up the grid, rather than continuously over the region of interest. This potential ($\vec{A}$) is subsequently differentiated to produce the magnetic field components B_x, B_y or B_r and B_z, depending on whether the problem has Cartesian or Axial Symmetry.

Let us consider as a test problem a pipe of .5" thickness and 4.0" ID (see Fig. 1) which is exposed to a constant magnetic field of varying intensity (for example, from 1000 gauss to 10,000 gauss). To model this problem, we must describe the geometry of the pipe into a grid system such as that shown in Fig. 2. Note that this grid is triangular and consists of 60^{o} intersecting lines. We find such grids to be more appropriate for this type of problem since they simulate curved boundaries more smoothly than rectangular grids. Actually, it can be shown that for interior regions, right-angle elements with two equal sides produce equations that are identical to the finite-element equations.

The mesh size depends on the type of accuracy one desires. Obviously, the finer the grid the finer the degree of approximation, and the resulting discretization of the partial differential equations is more accurate. However, increasing the mesh increases the computer memory requirements as well as the computer time.

For the example under consideration I chose 31x31 mesh size. On this mesh we describe the geometry we would use to simulate, in this case, the steel pipe.

One might argue that the pipe outlined on the grid in Fig. 2 is far from being circular. That is true, however, this grid should be considering as a "stretched membrane", which assumes the given boundary by stretching the corresponding boundary points to lie on the circular surface.

The input data to describe this geometry would require a substantial effort if one had to enter every x and y coordinate describing the two circles that make up the pipe. However, we have written an automatic conic section generator which requires a minimum number of points to describe this geometry. Actually, the data needed to describe the two circles are:

2	2	S						$ REGION 2
-2	16	16	3	3	2	7	C	$ REGIONAL POINTS
3	1	S						$ REGION 3
-2	15	15	3	3	1.7	3	C	$REGIONAL POINTS

In the above sequence the first entry specifies the region number the type of the material (2 = iron) while the next entry gives the following information:

-2	Code to signify that this is a circle
16	L - mesh line of center of circle (see Fig. 2)
16	K - mesh line of center of circle (see Fig. 2)
3.0	Y - coordinate of center of circle in units of length
3.0	X - coordinate of center of circle in units of length
2.0	R - radius of circle
7	NPTS - number of mesh lines to describe radius

A similar interpretation is given for the other two lines of input. The automatic mesh generator is extremely general to include parabolas, hyperbolas, straight lines, etc., connected in any desired way to produce the required geometry. Thus very little effort is required in preparing the input data for a complicated geometry.

Once the input is executed, the program produces the mesh which will be used by the second phase of the program to calculate the resulting magnetic field distribution. This mesh is shown in Fig. 3. Note the distortion that the mesh has undergone to produce the required exact geometry shown in Fig. 1. Note also that the circles are discretized so that they actually consist of straight line segments. Obviously, the more points one allocates to simulate such geometry, the shorter the straight line segments would be, and the resulting cylinder would be more accurately simulated.

Once the mesh has been successfully generated, the program proceeds to calculate the vector potential $\vec{A}$, the derivatives of which produce the B_x and B_y components of the magnetic field.

During this calculation the program uses magnetization curves that describe the B & H characteristics for the iron used. These curves are compiled from manufacturer's data and are stored internally in the computer, or supplied externally by the user. The program runs iteratively; that is, the mesh is scanned many times, usually 200-300 times. At each scan or iteration, the potentials are subjected to various tests for convergence and upon successful completion, various printouts and plots are obtained. Fig. 4 shows a series of flux plots for various input conditions. For example, in Fig. 4A the applied external magnetic fields is 2000 gauss while that of Fig. 4B is 10,000 gauss. Notice that at low field the iron pipe barely manages to contain the magnetic flux, while at 1 Tesla the pipe saturates and most of the flux is leaking outside the pipe.

There are many modes that such programs operate. In this example the program was used as a boundary value problem, that is, the boundary was specified and fixed and the magnetic field distribution within that boundary was requested; however, many other options may be exercised. Some of the options are outlined below:

a. Constant μ

In this option the magnetic material is considered to have constant permeability μ ($\mu = B/H$).

b. For a given geometry

Given current)
Find flux distribution)

In this case a certain current excitation in ampere-turns is given and the program calculates the magnetic field that results from such current distribution.

c. For a given geometry, given the magnetic field at some point, find necessary current to produce it

In this case, we define the total magnetic field at some point and the program calculates the current excitation required to produce it.

Besides the computer plots showing the flux lines resulting from the computations performed, the program also prints tables of the magnetic field components at strategic points or areas defined by the user, energy stored in the system as well as many other pertinent quantities.

The above brief description gives an idea of the type of calculations and results that one should expect by using such simulation codes.

There are a lot of fine points that have not been discussed, particularly that of boundary conditions that numerical models assume. Such conditions not only save computer space and execution time, but improve the accuracy and use of the program. For example, a look at Fig. 4 will convince the reader that there is a four-fold symmetry in this problem. That is, if one draws a horizontal and vertical line through the center of the pipe, as suggested in Fig. 5, then any one of the four quadrants is an image of the other properly rotated. This means that if we use the appropriate boundary conditions, we can simulate only one quarter of the proposed geometry.

In our example, we could have run in the computer, say, the first quadrant with boundary conditions known on Neumann in the horizontal plane, and Dirichlet on the vertical, and the results obtained would have been identical to those obtained by running the whole pipe. Obviously, we save a substantial number of mesh lines as well as computer memory and computer time.

IV. FUTURE OF MODELING

The discussion elaborated above assumes no interaction between user and machine; however, with the advent of microprocessors and CAD/CAM it is extremely important to create such a dialog. As far back as 1972, the writer has advocated computer interaction[6,7] to solve problems of this type. Actually, only a portion of such programs need be executed in a large machine. A good practice is to solve part of the problem on a personal computer, and connect to a large mainframe for that part of the problem that requires "number crunching". For example, in the case discussed earlier, the input stage as well as the mesh generation and display could be performed on a personal computer, and once the input resembles the model to be simulated, then that "file" is transmitted to the mainframe for solving. This way the user has local control of his program and files, performs editing and plotting functions and does not have to tax a large machine for such simple operations. Similarly, once the large mainframe has completed the solution process, the resulting "output" file can be transferred from mainframe to micro-computer for performing whatever output functions are needed.

Such "pre" and "post" processing is becoming an extremely desirable mode of operation, since it puts the scientist or engineer in the computational loop. Such interaction, extremely expensive a few years ago, is becoming easy to accomplish and reasonable to implement. The writer is presently engaged in preparing such a program to run on a small personal computer.

In the discussion above, intentionally, I make no mention of the calculation of magnetic fields in three-dimension or about time varying fields. Even though there exist programs that calculate such fields, their description is quite a bit more complicated to describe, and the reader is referred to the references at the end of this article.

The last topic I would like to touch upon is that of the calculation of magnetic fields for permanent magnets. Such problems may also be solved using various simulation models. Actually, the program described above can be used to simulate some permanent magnetic material (such as summarium cobalt) by replacing the permanent magnetic material with two very thin current sheets and assigns the permeability of free space in the remainder of the magnet. Such methods are appropriate only if the demagnetization curve (B & H) is a straight line. Recently, Halbach[8] has published a paper in which a computer program is described that produces reasonable computational results for a variety of permanent magnetic materials.

LITERATURE CITED

1. J. S. Colonias, TRIM: A Magnetostatic Computer Program for the CDC and VAX Computer Systems; LBL-16276, Lawrence Berkeley Laboratory, Berkeley, CA 94720 (Feb. 1984).

2. S. J. Sackett, Calculation of Electromagnetic Fields and Forces in Coil Systems of Arbitrary Geometry; UCRL-77244, Lawrence Berkeley Laboratory, Berkeley, CA 94720 (Nov. 1975).

3. K. Halbach, R. Holsinger, POISSON; Private communication.

4. Armstrong et. al, New Developments in the Magnet Design Program GFUN; RL-75-066, Rutherford Laboratory (1975).

5. J. Simkin, C. W. Trowbridge, Three Dimensional Computer Program (TOSCA) for Non-Linear Electromagnetic Fields; RL-79-097, Rutherford Laboratory (Dec. 1979).

6. J. S. Colonias, Particle Accelerator Design: Computer Programs; Academic Press, 1974, page 38.

7. J. S. Colonias, Computer Design of Optimization of Iron Cored Particle Focussing Devices; Nuclear Instruments and Methods 99 (1972).

8. K. Halbach, Design of Permanent Multipole Magnets with Oriented Rare Earth Cobalt Material; Nuclear Instruments and Methods 169 (1980).

Fig. 1. Cross Sectional View of a Metal Pipe

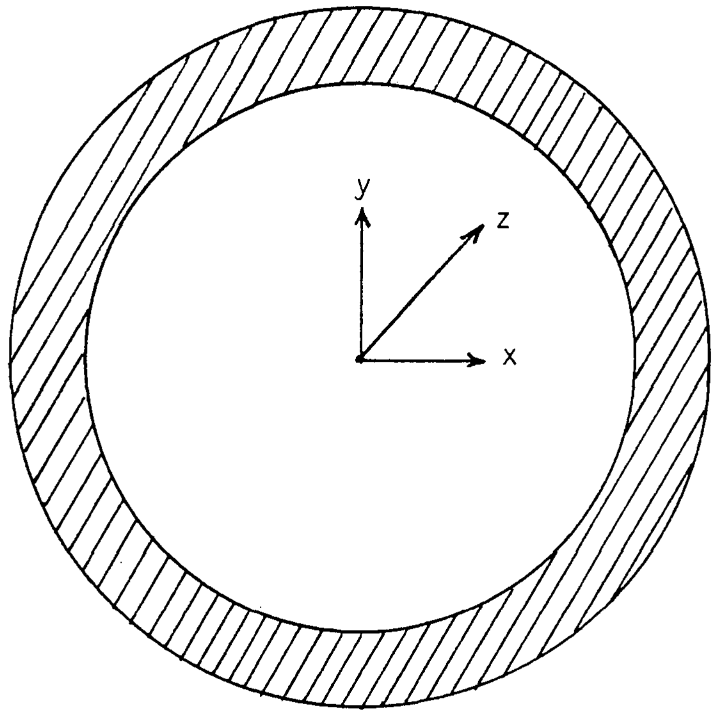


Fig. 2. Logical Diagram of Metal Pipe to be Simulated

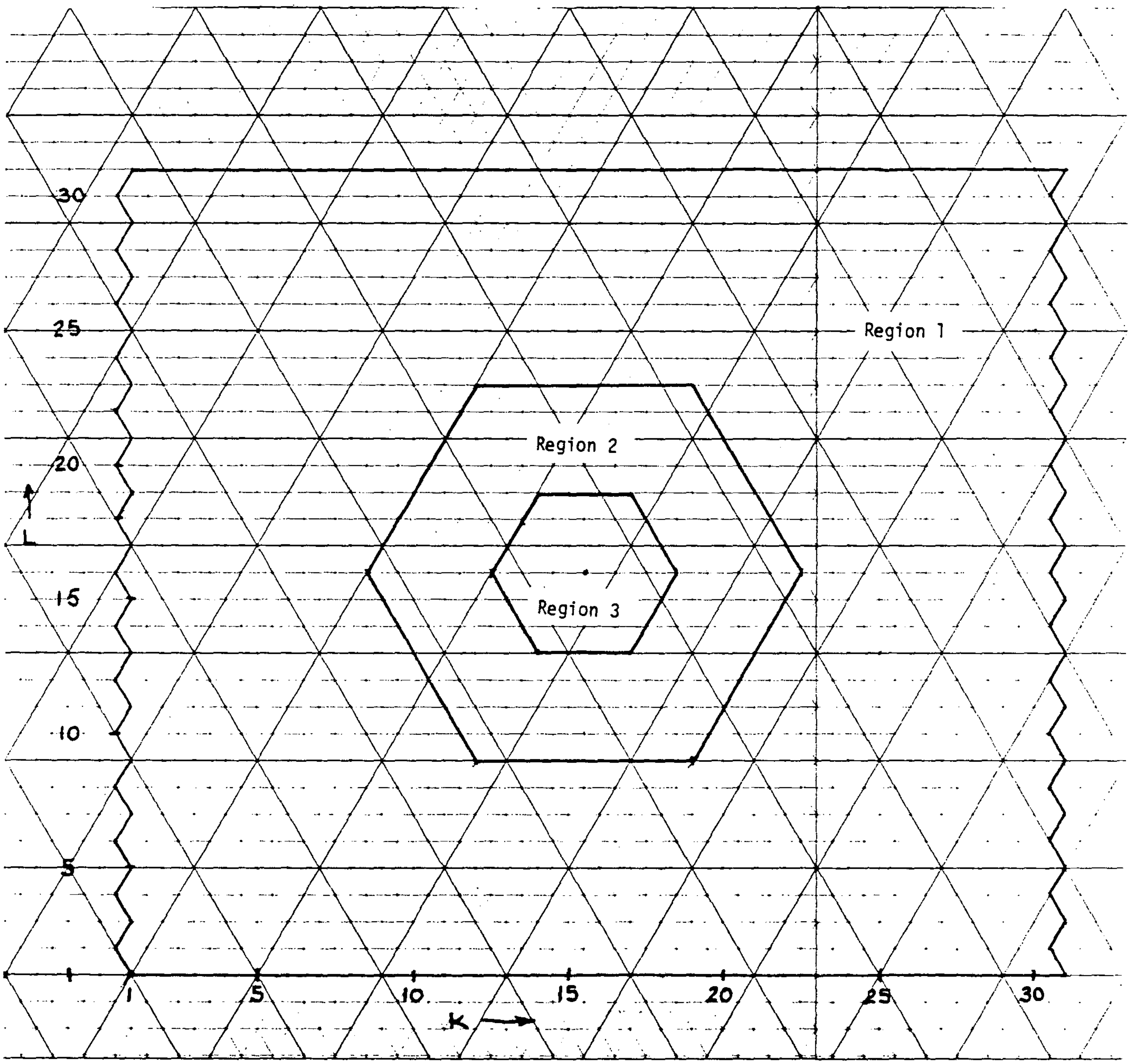
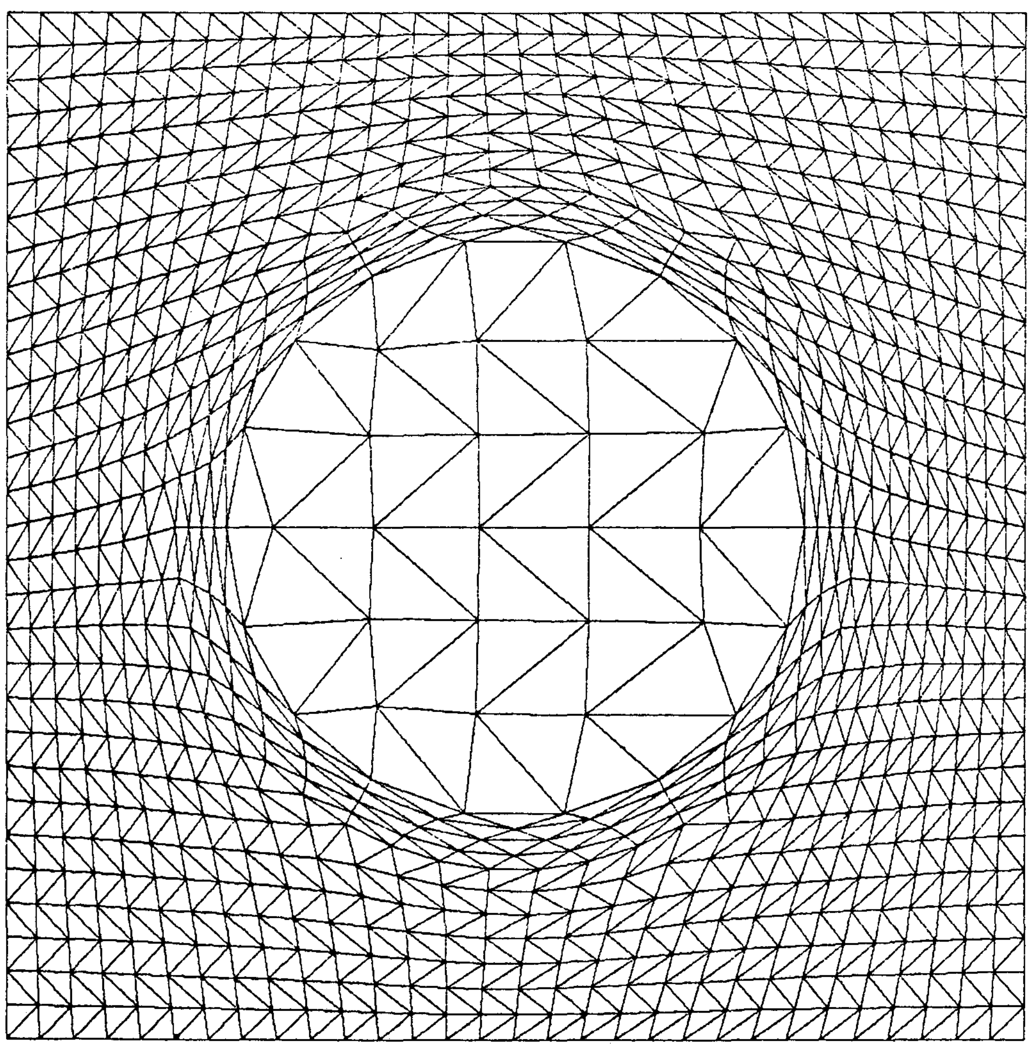


Fig. 3. Computer Generated Mesh from Logical Diagram

TEST CASE . METAL CYLINDER IN A UNIFORM MAGNETIC FIELD



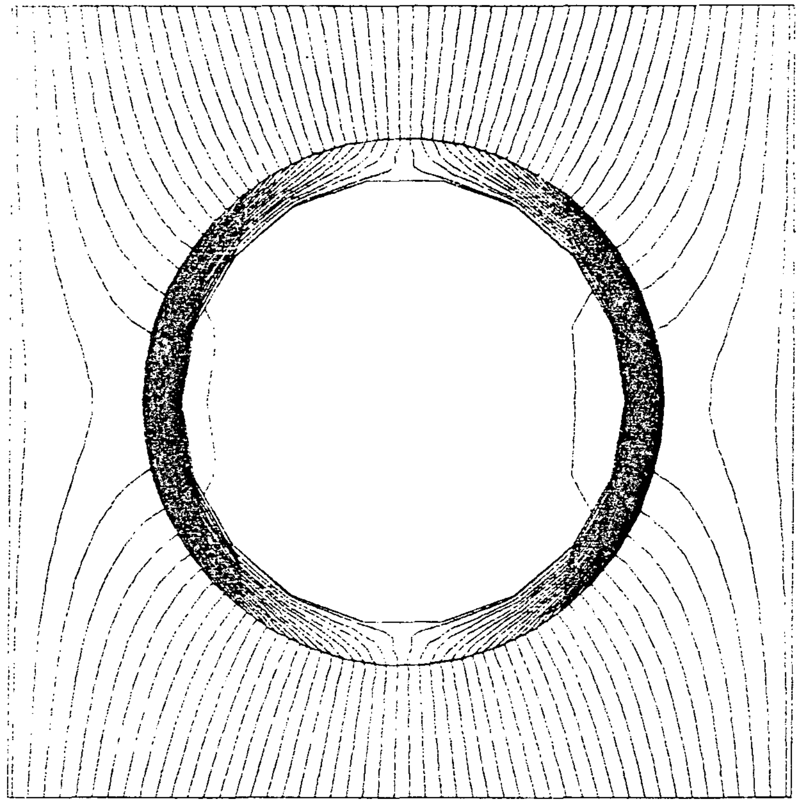


Fig. 4A. Flux Distribution at .2 Teslas. Note that most flux is retained within the pipe.

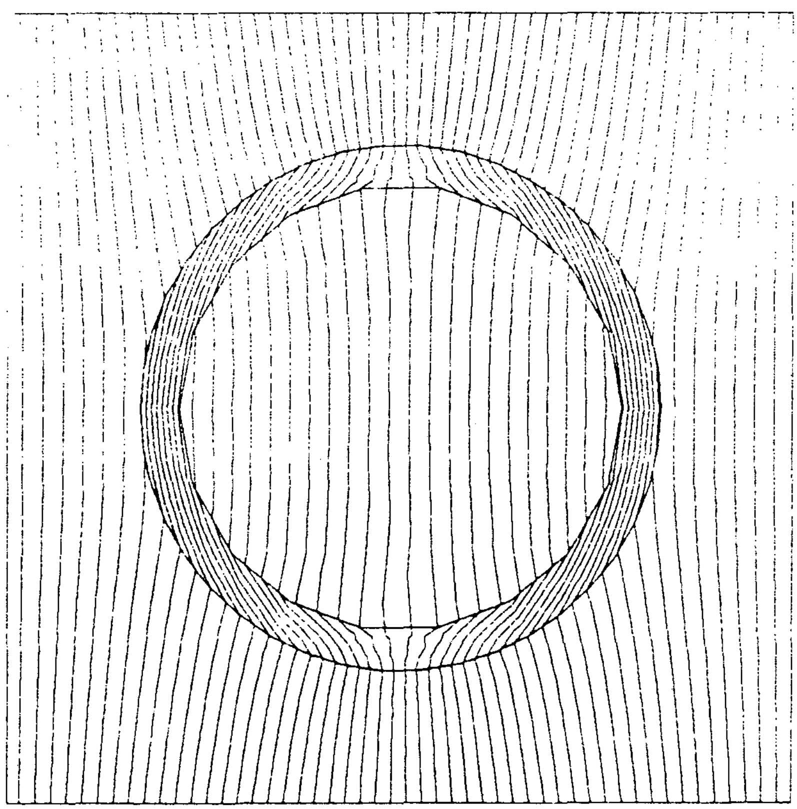


Fig. 4B. Flux distribution at 1 Tesla. Note that pipe has saturated and most flux passes through as if pipe was not there.

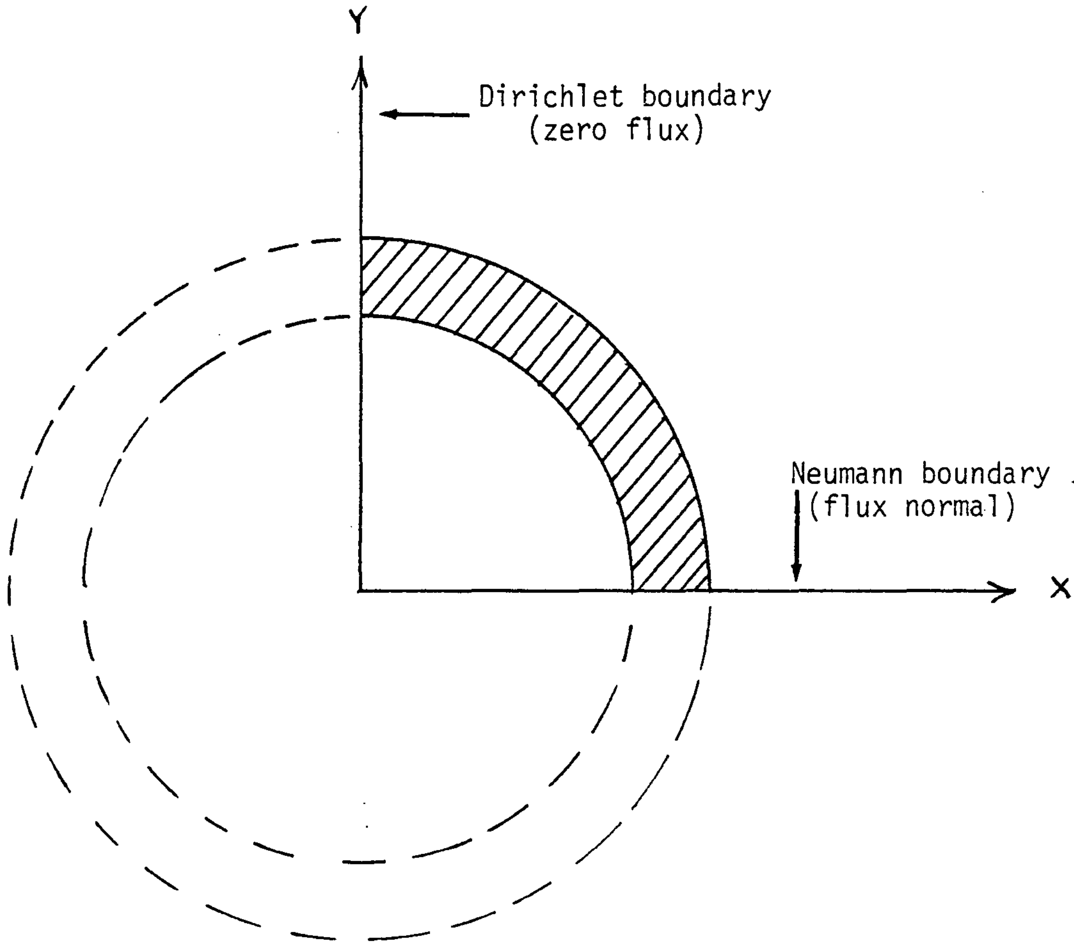


Fig. 5. Application of Boundary Conditions
Reduces Size of Problem.