DR-0401-4

SPARK Version 1.1 User Manual

MASTER

By

D.W. Weissenburger

JANUARY 1988

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PLASMA
PHYSICS
LABORATORY

PRINCETON UNIVERSITY
PRINCETON, NEW JERSEY

PREPARED FOR THE U.S. DEPARTMENT OF ENERGY,
UNDER CONTRACT DE-AC82-76-CO-3073.

REPRODUCTION OF THIS DOCUMENT IS UNLIMITED

SPARK Version 1.1 User Manual

PPPL--2494

DE88 006255

D. W. Weissenburger

Plasma Physics Laboratory
Princeton University
Princeton, New Jersey 08544

ABSTRACT

This manual describes the input required to use Version 1.1 of the SPARK computer code. SPARK 1.1 is a library of FORTRAN main programs and subprograms designed to calculate eddy currents on conducting surfaces where current flow is assumed zero in the direction normal to the surface. Surfaces are modeled with triangular and/or quadrilateral elements. Lorentz forces produced by the interaction of eddy currents with background magnetic fields can be output at element nodes in a form compatible with most structural analysis codes. In addition, magnetic fields due to eddy currents can be determined at points off the surface.

Version 1.1 features eddy current streamline plotting with optional hidden-surface-removal graphics and topological enhancements that allow essentially any orientable surface to be modeled. SPARK also has extensive symmetry specification options. In order to make the manual as self-contained as possible, six appendices are included that present summaries of the symmetry options, topological options, coil options and code algorithms, with input and output examples.

An edition of SPARK 1.1 is available on the Cray computers at the National Magnetic Fusion Energy Computer Center at Livermore, California. Another more generic edition is operational on the VAX computers at the Princeton Plasma Physics Laboratory and is available on magnetic tape by request. The generic edition requires either the GKS or PLOT10 graphics package and the IMSL or NAG mathematical package. Requests from outside the United States will be subject to applicable federal regulations regarding dissemination of computer programs.

I. INTRODUCTION

This manual describes the input required to use Version 1.1 of the SPARK computer code. SPARK 1.1 is a library of FORTRAN main programs and subprograms designed to calculate eddy currents flowing on conducting surfaces where current flow is assumed zero in the direction normal to the surface. Version 1.0 of SPARK was published in Ref.[1] along with detailed descriptions of the various algorithms employed, but the original User Manual and User Manual Supplement existed previously only as unpublished reports. Version 1.1 has the same basic organization as Version 1.0, but features some improved algorithms and a substantial number of additional options, particularly in the graphics output. Input created using the Version 1.0 conventions, however, is completely compatible with Version 1.1.

This manual is intended to be self-contained and a user should find the various code options presented in sufficient detail to use SPARK effectively without referring to earlier reports, but an extensive list of references is provided for those readers desiring additional information. Novice users are encouraged to run the example programs that accompany SPARK in order to learn the basic code capabilities. Experience has shown that users often experience the most difficulty with symmetries and topological properties. For each of these topics a more detailed exposition is included as an appendix to this manual.

The next section provides a summary of the code organization. The following five sections give full descriptions of the input parameters for each of the five SPARK main programs. In addition to Appendix A on Symmetries and Appendix B on Topological Properties mentioned above, four other Appendices are included: Appendix C summarizes the code modifications made from Version 1.0 to Version 1.1; Appendix D presents three sets of example input data; Appendix E presents the graphic output produced from the examples in Appendix D; and Appendix F describes the coil specification parameters employed in Steps 2 and 5.

1. Code Availability

Version 1.1 of SPARK is presently operational on the Cray computers at the National Magnetic Fusion Energy Computer Center (NMFECC) at Livermore, California, and on our local VAX computers here at the Princeton Plasma

Physics Laboratory (PPPL). NMFECC members can access an on-line documentation file that gives the necessary information to copy that edition of the code into their local areas. The VAX edition of the code is written in a nearly generic FORTRAN/77 and will be supplied on magnetic tape by request from PPPL. Requests from outside the United States will be subject to applicable federal regulations regarding dissemination of computer programs. Both editions of SPARK are furnished with several sets of example input files and their corresponding text output. In addition to serving as learning tools for users, the examples can check a particular installation of the code.

2. Additional Software Required

The present NMFECC/VAX Editions of SPARK use three subroutines from the proprietary International Mathematical and Statistical Library (IMSL) [2] for three operations :

- (1) Solving a system of simultaneous equations that can be expressed with known matrix $[A]$, known vector $\{ b \}$, and unknown vector $\{ x \}$ as $[A] \{ x \} = \{ b \}$;
- (2) Inverting a matrix;
- (3) Numerically integrating a set of simultaneous ordinary differential equations using a variable step Adams method.

An optional edition is also supplied that employs the corresponding Numerical Algorithm Group (NAG) [3] proprietary routines instead. As a third option, a user can substitute any applicable local routines he has available. In fact Version 1.0 was issued with a non-proprietary Runge-Kutta integration routine instead of the Adams method.

The present NMFECC Edition employs the GRAFLIB [4] package of graphical subroutines. It is expected that an optional edition employing the Graphical Kernel System (GKS) [5] will be available shortly. The VAX Edition is supplied with a GKS option or a PLOT10 [6] option. The graphics operations called externally by SPARK are all very primitive and can generally be straightforwardly substituted from other graphics packages if necessary.

The only other non-standard external calls made by SPARK are for the time and date and, with the PLOT10 option, the ASCII value of a character. The time and date can be commented out for an initial code installation and are not essential, but are very helpful in keeping track of runs. If no substitute routine is available for the ASCII value, one can be readily written by a user.

II. CODE ORGANIZATION

1. Modular Construction

SPARK 1.1 is divided into five principal modules corresponding to five main programs called: Step 1 (Pre-Processor); Step 2 (Driver); Step 3 (Inversion); Step 4 (Main Processor); Step 5 (Post Processor).

All the modules require subprograms from the small Common SPARK Library and Steps 2 and 5 use subprograms from the Basic Magnetic Field Design Library. Both these libraries are supplied with SPARK. In addition the necessary graphics package for a given computer system and the IMSL or NAG library is needed as described in the Introduction.

The data input files that are described in detail in Sections III-VII following are named on the Cray: I1, I2, I4, I5 for Steps 1, 2, 4, 5 respectively. No input data is required for Step 3. On the Vax the names are: I1.DAT, I2.DAT, I4.DAT, I5.DAT respectively. The text output files from each step are named on the Cray: O1, O2, O3, O4, O5 respectively for Steps 1-5. On the vax the output files are named: O1.DAT, O2.DAT, O3.DAT, O4.DAT, O5.DAT respectively.

SPARK saves all essential information on disk files that are passed from step to step. On the Cray, these files are called FILE20, FILE23, FILE26, etc. On the Vax, these files are called FILE20.DAT, FILE23.DAT, FILE26.DAT, etc. In addition there are occasional scratch files that are created and destroyed as necessary. Near the beginning of the main program source listing for each step, comment lines can be found that describe the permanent files that will be read and/or written by that step.

Once Step 1 has been executed for a particular geometry, Step 2 can be run an arbitrary number of times without rerunning Step 1. This also applies to Steps 3-5. Step 1 may also be run again, without altering the initial file data, to produce additional graphics or rename the output headings. Steps 3 and 4 apply only to transient analysis and can be skipped if a static analysis option is selected in Step 2. The two static analyses available in Version 1.1 are the pure resistive solution and the pure inductive solution. If only a pure resistive solution is desired, it is possible to specify calculation of only resistances and no inductances in Step 1 to save computation time.

2. STEP 1: Pre-Processor

Step 1 is called the Pre-Processor and has three main options: (1) read data for a new model and generate the basic bookkeeping files; (2) plot the input geometry; (3) generate the mesh resistance and inductance matrices.

The resistance and inductance calculations are the most time-consuming and are usually deferred until the model geometry has been at least preliminarily checked and plotted. SPARK does make a number of checks for internal consistency of the data, but errors are still possible. Appendices D and E give examples of many of the plot options available in Step 1.

In making inductance and resistance calculations, SPARK associates a rectangular plate with each side of the mesh as shown in Fig. 1. The side of the mesh is equivalent to a branch line and thus each plate is associated with a branch line. The plate has the same length as its branch line and has a

Fig. 1. General quadrilateral mesh and associated plates.

width equal to the perpendicular distance from the branch line to the mesh centroid. The plate thickness and resistivity are as specified for the mesh. Electric currents are assumed to be uniformly distributed across the width of the plate. The plate resistance is then readily calculated by a formula. The magnetic vector potential due to a plate carrying unit current is determined by more complicated formulas, with simpler expressions being employed when the distance from the plate is relatively large. The magnetic vector potential is numerically integrated along each branch line to determine a plate-to-plate inductance value. These plate resistances and inductances are then transformed into branch resistances and inductances and then finally transformed into mesh resistances and inductances. A fuller discussion of these topics can be found in Refs. [1] and [7].

3. STEP 2: DRIVER

Step 2 is called the Driver and calculates the external electromotive forces (emf's) that drive the eddy currents through the model. A mesh emf is equivalent to the negative derivative of the mesh flux with respect to time. The magnetic flux per unit driver current through each mesh is calculated in essentially the same way as Step 1, i.e., the magnetic vector potential due to each driver is numerically integrated around each mesh. The resulting flux values are multiplied by the negative derivative of the driver current with respect to time to obtain a mesh emf at a given time.

There are five basic driver types available in Version 1.1: (1) a set of coaxial circular coils with various cross sections; (2) a set of generally oriented circular coils with various cross sections; (3) a spatially uniform field in a specified direction; (4) a $1/R$ field dependence about the z -axis; (5) a set of straight line segments. Appendix F illustrates the parameters used to specify types (1) and (2).

First a table of time points is input to Step 2, then the geometry of all the desired drivers is defined, then a table of currents for each time point for each driver is given. In the case of options (3) and (4), pseudo-currents are defined that effectively produce fields in the same way as the true current options. Magnetic field values are interpolated linearly between time points. This yields a constant emf value for each driver between time points.

It is possible to approximate a coil that changes geometry as well as current with time by defining a series of "snapshot" coils that are turned on and off in succession. This effectively puts a "hat" function current on each coil that starts at zero from the previous time point, reaches its maximum value at the time point itself, and descends to zero again at the next time point. This is illustrated with the torus example in Appendices D and E.

Besides transient eddy current calculation, Step 2 also has two static calculation options. The first option is the pure inductance solution. The mesh currents obtained from this solution correspond to the negative values of the currents that would suddenly arise to maintain an initial driver flux configuration after a step change in the driving field. With this option, the driver currents input at time zero are assumed to be at time zero plus $\{ I_d(0^+) \}$ and SPARK will read an additional set of driver currents for time zero minus $\{ I_d(0^-) \}$.

Let: $\{ I_m \}$ = mesh current vector;
 $[L]$ = mesh inductance matrix;
 $[F_d]$ = mesh flux per unit current due to the external drivers matrix.

Then the inductive solution is obtained in SPARK by solving the equation

$$[L] \{ I_m \} = [F_d] \{ \{ I_d(0^+) \} - \{ I_d(0^-) \} \} \quad (11.1)$$

for the mesh currents. An initial inductive distribution is shown for the 1/8 square plate example in Appendices D and E.

The second option is called the pure resistance solution. The mesh currents obtained from this solution correspond to the steady-state solution that would be obtained after a very long time given a constant emf driver. If this option is selected, SPARK will read a set of driver current derivatives with respect to time $\{ dI_d/dt \}$. Let $[R]$ = mesh resistance matrix. Then the pure resistance solution is obtained by solving the equation

$$[R] \{ I_m \} = - [F_d] \{ dI_d/dt \} \quad (11.2)$$

for the mesh currents.

If no static solution is specified, the initial mesh currents are set to zero. This is shown in the torus example in Appendices D and E.

4. STEP 3: INVERSION

Step 3 is called Inversion. In this step the mesh inductance matrix is inverted from $[L]$ to $[L]^{-1}$ for later use in Step 4.

5. STEP 4: MAIN PROCESSOR

Step 4 is called the main processor. In this step the system of differential equations established by the model in Step 1 is integrated subject to the driving emf's and initial currents defined in Step 2. The mesh currents are output at time steps selected by the user to a text file and to a disk file, FILE40. The system of equations is solved using the inverted mesh inductance matrix obtained in Step 3. We are thus integrating the system

$$\{ dI_m/dt \} = [L]^{-1} \{ -[F_d] \{ dI_d/dt \} - [R] \{ I_m \} \}. \quad (11.4)$$

6. STEP 5: POST PROCESSOR

Step 5 is called the Post Processor. Three principal options are available for any time step on File 40: (1) streamline plotting; (2) remote field calculation; and (3) nodal force calculation. There also is an auxiliary option for branch current determination that must be enabled for remote fields and/or force calculation and is sometimes useful by itself.

Several streamline plotting examples are included in Appendices D and E. Version 1.1 has some significantly improved plotting options that are described in detail in Section VII and in Appendix C. A hidden surface removal option is now available. When a model is specified using the symmetry options described in Appendix A, it is not necessary to plot all the symmetries on a graph. The user may select the reflections and rotations to be exhibited. Version 1.1 has an enhanced flexibility in the selection of rotations to be exhibited. It is also optional to specify that only certain parts of the total model are to be exhibited.

Magnetic fields from the driver currents and the eddy currents may be calculated at points not on the surface of the model with Version 1.1. Additional static background fields can also be defined in Step 5. Background fields are specified with the same five source options as driver fields except

with constant current values. Static fields do not produce eddy currents, but do produce Lorentz forces when they interact with the eddy currents. If desired, remote fields can be calculated from eddy currents only in specified parts and symmetries. This procedure can be useful in checking calculations sometimes.

Nodal loads are calculated from the branch currents and the magnetic fields at the nodes using the principle of work equivalent loads described in Ref. [8]. The magnetic fields are assumed to vary linearly along the length of the branch. Consider a branch of length L from node i to node j loaded by a transverse force per unit length varying linearly from f_i to f_j respectively. We can write the nodal forces, F_i and F_j , and nodal moments, M_i and M_j , as

$$F_i = (3f_j + 7f_i) L/20 \quad (II.5)$$

$$F_j = (7f_j + 3f_i) L/20 \quad (II.6)$$

$$M_i = (2f_j + 3f_i) L^2/60 \quad (II.7)$$

$$M_j = (-3f_j - 2f_i) L^2/60 \quad (II.8)$$

Since Lorentz forces are given by $d\vec{F} = I d\vec{l} \times \vec{B}$, there is only a transverse load on each branch and no net axial load. In general there are three components on a branch, however, that must be calculated.

SPARK also sums up the net forces, net moments, and net moment of the forces about the global origin for the parts and symmetries that are specified. Unlike streamline plotting and remote field calculations, force calculations can only be exhibited for reflection symmetries and not rotation symmetries.

III. JOB STEP 1 (PRE PROCESSOR) INPUT

1. HEADING LINE.....1 line

HEAD(72)

FORMAT(72A1)

HEAD = run identification label.

If the input file consists of only a heading line, and no other data, SPARK assumes that the basic bookkeeping file, FILE20, already exists and will change the previous heading to the new heading. An error exit will occur if FILE20 cannot be found.

2.0 MASTER CONTROL DATA

2.1 FIRST LINE.....1 line

NSTEPA, NSTEPB, NSTEPC

FORMAT (3I5)

NSTEPA = 1 → create a new FILE20,

0 → use old FILE20 (error exit if not found);

NSTEPB = 1 → produce graphic output,

0 → do not produce graphic output;

NSTEPC = 1 → produce inductance and resistance files,

-1 → produce only resistance file (useful when only pure resistance solution desired),

0 → do not produce inductance or resistance files.

2.2 SECOND LINE.....1 line

NUMNP

FORMAT(I5)

NUMNP = total nodal points (must be numbered from 1 to NUMNP).

3.0 NODAL POINT DATA.....NUMNP lines
(in any order, one line for each point)

N, X(N), Y(N), Z(N)

FORMAT (I5,3F10.6)

N = point number;

X(N),Y(N),Z(N) = x,y,z coordinates of point in meters.

4.0 PLATE ELEMENT AND CLOSURE MESH DATA

4.1 PLATE ELEMENT AND CLOSURE MESH CONTROL LINE.....1 line

NUMEL, NUMMAT, NUM3, NUM4, NUMCM

FORMAT (I10,4I5)

NUMEL = total mesh elements (must be numbered from 1 to NUMEL).

NUMMAT = total materials (must be numbered from 1 to NUMMAT).

NUM3 = total triangular elements.

NUM4 = total quadrilateral elements (NUM3 + NUM4 = NUMEL).

NUMCM = total closure meshes.

4.2.0 MATERIAL PROPERTY DATA.....NUMMAT sets of lines
(in any order, two lines for each material)

4.2.1 MATERIAL PROPERTY FIRST LINE.....1 line

NMAT

FORMAT(I10)

NMAT = material identification number.

4.2.2 MATERIAL PROPERTY SECOND LINE.....1 line

RHO

FORMAT(E10.3)

RHO = resistivity in ohm-meters.

4.3 TRIANGULAR ELEMENT DATA.....NUM3 lines
(in any order, one line for each triangular mesh)

N, I(N), J(N), K(N), MAT(N), TK(N)

FORMAT(4I5,I15,F15.6)

N = mesh number;

I(N),J(N),K(N) = node numbers of mesh in rotational order;

MAT(N) = material identification number of mesh;

TK(N) = thickness of mesh in meters.

4.4 QUADRILATERAL ELEMENT DATA.....NUM4 lines
(in any order, one line for each quadrilateral mesh)

N, I(N), J(N), X(N), L(N), MAT(N), TK(N)

FORMAT(5I5,I10,F15.6)

N = mesh number;

I(N),J(N),K(N),L(N) = node numbers of mesh in rotational order;

MAT(N) = material identification number of mesh;

TK(N) = thickness of mesh in meters.

5.0 HOLE AND CUT LOOP

(each loop is one mesh that can consist of from 1 to 100 vectors)

5.1 LOOP CONTROL CARD.....1 line
NUMHOL

FORMAT(I5)

NUMHOL = total loops (must be numbered from 1 to NUMHOL).

(if NUMHOL=0, skip to section 6)

5.2.0 LOOP PARAMETERS.....NUMHOL sets of lines
(in any order, KH(N)+1 lines for each set)

5.2.1 VECTOR CONTROL LINE.....1 line
N, KH(N)

FORMAT (2I5)

N = loop number;

KH(N) = total vectors in loop (<100).

(If loop is a cut, input KH(N) as a negative number)

5.2.2 VECTOR DATA.....KH(N) lines
 (in any order, one line for each Vector)
 IH(N,NB), JH(N,NB)
 FORMAT(2I5)

IH(N,NB) = starting node number of vector;

JH(N,NB) = ending node number of vector.

The direction of a vector must agree with the direction of rotation of the loop. SPARK will arrange the vectors in order from NB=1* to NB=KH(N) according to initial node numbers, so that in general the input order is changed.

6.0 ZERO-STREAM-VALUE BOUNDARY DATA

6.1 ZERO-STREAM-VALUE CONTROL LINE.....1 line
 NUMS
 FORMAT(I5)

NUMS = total zero-stream-value boundary vectors (must be numbered from 1 to NUMS).

(if NUMS=0, skip to section 7)

6.2 VECTOR PARAMETERS.....NUMS lines
 (in any order, one line for each Vector)
 N, IS(N), JS(N)
 FORMAT(3I5)

N = vector number;

IS(N) = starting node number of vector;

JS(N) = ending node number of vector.

(direction of vectors for stream-zero boundaries is not important, transposition of the node numbers corresponding to IS and JS has no effect on subsequent calculations)

7.0 EQUIPOTENTIAL BOUNDARY DATA

7.1 EQUIPOTENTIAL CONTROL LINE.....1 line
 NUME
 FORMAT(I5)

NUME = total equipotential boundary vectors (must be numbered from 1 to NUME).
 (if NUME=0, skip to section 8)

7.2 VECTOR PARAMETERS.....NUME lines

(in any order, one line for each vector)

N, IE(N), JE(N)

FORMAT (3I5)

N = vector number;

IE(N) = starting node number of vector;

JE(N) = ending node number of vector.

(direction of vectors for equipotential boundaries is not important, transportation of the node numbers corresponding to IE and JE has no effect on subsequent calculations)

8.0 SYMMETRIES DATA

8.1 SYMMETRIES CONTROL LINE.....1 line

NUMSYM

FORMAT(I5)

NUMSYM = total symmetries (must be numbered from 1 to NUMSYM).

(if NUMSYM = 0, skip to section 9)

8.2 SYMMETRY PARAMETERS.....NUMSYM lines

(in any order, one line for each symmetry)

N, IXY(N), IYZ(N), IZX(N), JZ(N)

FORMAT(5I5)

N = symmetry number;

IXY(N), IYZ(N), IZX(N)=

(+1 + mirror reflection)

(0 + no reflection) about xy, yz, or zx

(-1 + reverse mirror reflection) plane respectively;

JZ = number of symmetric repetitions about z axis (0 is same as 1).

A discussion of the symmetry specifications is included as an appendix to this manual.

8.3 SYMMETRY GROUPS CONTROL LINE.....1 line

NGREL

FORMAT(I5)

NGREL = total groups of elements for which each element within the group will be assigned the same symmetries.

8.4 SYMMETRY GROUP PARAMETERS.....NGREL lines

(in any order, one line for each group)

NELS, NELF, IST, (ISY(I),I=1,IST)

FORMAT(13I5)

NELS = starting element number of group

NELF = ending element number of group

IST = total symmetries of group (<10)

ISY(I), I=1,IST = symmetry numbers of group in order from 1 to IST

9.0 GRAPHIC OUTPUT DATA

9.1. GRAPH CONTROL LINE.....1 line

NGRAPH,NUMPAR

FORMAT(2I5)

NGRAPH = total graphs requested;

(if NGRAPH = 0, no more data will be read for job step 1)

NUMPAR = total parts to be distinguished as separate graphs, if any, from entire set of nodes, elements, and boundaries (must be numbered from 1 to NUMPAR and <10);

(If NUMPAR = 0, skip to 9.3)

9.2 PARTS DATA.....NUMPAR LINES

(in any order)

NPAR, NPNOS(NPAR), NPNOE(NPAR),

NPELS(NPAR), NPELE(NPAR), NPHOS(NPAR), NPHOE(NPAR),

NPSTS(NPAR), NPSTE(NPAR), NPEQS(NPAR), NPEQE(NPAR)

FORMAT(11I5)

NPAR = part number,

NPNOS = starting node number,

NPNOE = ending node number,

NPELS = starting mesh number,

NPELE = ending mesh number,

NPHOS = starting hole number,

NPHOE = ending hole number,

NPSTS = starting zero-stream-value vector number,

NPSTE = ending zero-stream-value vector number,

NPEQS = starting equipotential vector number,

NPEQE = ending equipotential vector number.

Note: If NPxxS(NPAR)=0 → no xx for part no. NPAR.

Note: Only the parameters, which are actually graphed, need specification. For example, if only meshes are graphed, then only NPELS and NPELE are needed for each part and the remaining parameters can be set to zero.

9.3.0 GRAPH PARAMETERS.....3*NGRAPH lines
(three lines for each graph)

9.3.1 GRAPH PARAMETERS FIRST LINE.....1 line
IGRAPH, JGRAPH, KGRAPH, LPART, MPART(IP=1, LPART)
FORMAT (14I5)

IGRAPH = 1 → graph nodes

2 → graph branches and boundaries

3 → graph meshes with solid direction indicators

4 → graph boundaries

5 → graph meshes with open direction indicators

6 → graph branches dashed and loops double solid

7 → graph meshes dashed with solid direction indicators and loops
double solid

8 → graph meshes dashed with open direction indicators and loops
double solid

12-18 → subtract 10 from IGRAPH and use same attributes as above, but
also add nodes to plot.

negative → turn viewport clipping on

positive → turn viewport clipping off

JGRAPH = 0 → omit component numbers

<1 → plot component numbers with character size = JGRAPH*0.06 inch
(components will be nodes, branches, boundaries, and/or meshes
depending on IGRAPH)

KGRAPH = 0 → omit axes

1 → plot axes

LPART = total parts to be included on graph, default = 0 → graph entire set

MPART(IP), IP=1, LPART = part number(s) to be included on graph, if LPART not 0, in any order.

9.3.2 SECOND LINE.....1 line

CXPX, CXPY, CXPZ, CYPX, CYPY, CYPZ

FORMAT(6F10.4)

CXPX,CXPY,CXPZ = directions cosines of x,y,z axes with abscissa of graph. If all zero, defaults to (1,0,0).

CYPX,CYPY,CYPZ = directions cosines of x,y,z axes with ordinate of graph. If all zero, defaults to (0,1,0).

9.3.3 THIRD LINE1 line

SX, XMIN, XMAX, YMIN, YMAX

FORMAT(5F10.4)

If SX = -1.0 → scale graph with user supplied limits:

XMIN = minimum graph value of abscissa;

XMAX = maximum graph value of abscissa;

YMIN = minimum graph value of ordinate;

YMAX = maximum graph value of ordinate.

If SX = 0.0 → scale graph from limits calculated by code.

IV. JOB STEP 2 (DRIVER) INPUT

1. TIME HISTORY CONTROL LINE.....1 line

NUMTIM

FORMAT(I5)

NUMTIM = total time points (must be numbered from 1 to NUMTIM).

2. TIME HISTORY DATA.....NUMTIM lines

(in any order, one line for each point)

NTIM, TIME(NTIM)

FORMAT(I5,F15.6)

NTIM = number of point,

TIME(NTIM) = time of point in seconds.

Note: TIME(1) must equal 0.0!

3. EXTERNAL COIL SETS CONTROL LINE.....1 line

NUMEXT

FORMAT(I5)

NUMEXT = total external coil sets (must be numbered from 1 to NUMEXT).

(if NUMEXT=0, skip to section 5)

4.0 EXTERNAL COIL SET DATA.....(NUMEXT sets of lines)

4.1 EXTERNAL COIL SET CONTROL LINE.....1 line

NEXT, NUMCL, ICOAX, NAMEXT

FORMAT(3I5,55A1)

NEXT = external coil set number;

NUMCL = total coils in set;

ICOAX = coaxial index:

0 + set is coaxial*,

1 + set is generally-oriented*,

2 + uniform field set (NUMCL=1),

3 + ideal 1/R field set (NUMCL=1),

4 + series of straight line segments;

NAMEXT = set name.

*Note: Circular coil set specifications are described in Appendix F.

4.2.0.0 IF ICOAX = 0:.....(1+NUMCL) lines

4.2.0.1 COAXIAL SET CONTROL LINE.....1 line

TQX, TOY, TQZ, TDX, TDY, TDZ

FORMAT(6F10.4)

TQX,TQY,TQZ = x,y,z coordinates of assigned origin on axis of set;

TDX,TDY,TDZ = x,y,z direction cosines of axis of set

4.2.0.2.0 COAXIAL COIL DATA.....NUMCL lines

TA1, TA2, TZ1, TZ2, TEXT

FORMAT(5F10.4)

TA1, TA2, TZ1, TZ2 = defined in following subsections according to coil cross sections;

TEXT = total turns of coil.

4.2.0.2.1 rectangular cross-section coil:

TA1 = inner radius of coil,

TA2 = outer radius of coil,

TZ1 = z distance from origin to bottom of coil,

TZ2 = z distance from origin to top of coil.

4.2.0.2.2 filamentary cross-section coil:

TA1 = TA2,

TZ1 = TZ2.

4.2.0.2.3 ideal pancake cross-section coil:

TZ1 = TZ2.

4.2.0.2.4 ideal solenoidal cross-section coil:

TA1 = TA2.

4.2.0.2.5 semi-infinite solenoidal cross-section coil

if TZ1 < -10000, the solenoid is assumed to extend from TZ2 to negative infinity;

if TZ2 > +10000, the solenoid is assumed to extend from TZ1 to positive infinity.

4.B.0.2.6 circular cross-section coil:

TA1 = major radius of coil,

TA2 = minor radius of coil,

TZ1 = distance from origin to centroid of coil,

TZ2 is not used.

Special note: A coil can have a variable current density, which depends on the major coil radius, by supplying a function named CURDEN and setting TEXT > 1.0E20. This applies to all the various cross section types. See Appendix F for further details.

4.2.1.0 IF ICOAX = 1:.....NUMCL pairs of lines

4.2.1.1 FIRST LINE OF GENERALLY-ORIENTED COIL DATA.....1 line

TQX, TOY, TQX, TDY, TDZ

FORMAT(6F10.4)

TQX,TOY,TQZ = x,y,z coordinates of centroid of coil,

TDX,TDY,TDZ = x,y,z direction cosines of axis of coil.

4.2.1.2 SECOND LINE OF GENERALLY-ORIENTED COIL DATA.....1 line

TA1, TA2, TZ1, TZ2, TEXT

FORMAT (5F10.4)

The same definitions used above in 4.2.0.2 apply here to TA1,TA2,TZ1,TZ2,TEXT.

4.2.2.0 IF ICOAX = 2:.....1 line

4.2.2.1 UNIFORM FIELD PARAMETERS.....1 line

UBXPA, UBYPA, UBZPA

FORMAT(3F10.4)

UBXPA, UBYPA, UBZPA = components of uniform magnetic field in (x,y,z) directions per unit "current per turn" values input in section 5.

4.2.3 IF ICOAX = 3:

No additional lines will be read for this set. SPARK will multiply "current per turn" values input in section 5 by $1/R$ to determine fields.

4.2.4.0 IF ICOAX = 4:

4.2.4.1 STRAIGHT LINE SEGMENT PARAMETERS.....NUMCL lines
 XBEGIN, YBEGIN, ZBEGIN, XEND, YEND, ZEND
 FORMAT(6F10.4)

XBEGIN, YBEGIN, ZBEGIN = x,y,z coordinates of beginning of line segment;
 XEND, YEND, ZEND = x,y,z coordinate of end of line segment.

All line segments in a set will be assigned the "current per turn" value input in section 5.

5. CURRENT HISTORY OF EXTERNAL SETS.....NUMTIM Groups of lines

5.1 FIRST LINE OF GROUP.....1 line
 NTIM
 FORMAT(I5)

NTIM = number of time point.

5.2 SECOND LINE(S) OF GROUP1 to NUMEXT lines
 NEXT,AREAD
 FORMAT(I5,F15.6)

NEXT = external coil set number;

AREAD = current per turn of set.

Note: If NEXT = 0, SPARK will stop reading values for time step NTIM and give all unspecified sets the default value of zero current for this time point.

Note: Each time point must be defined even if all sets have the default value.

6. PURE SOLUTION CONTROL LINE.....1 line
 INSTEP, NUMXST
 FORMAT(2I5)

INSTEP = 0, do not solve for either pure inductance or pure resistance. Set initial mesh currents to zero.

= 1, solve for pure inductance. Set initial mesh currents to solution.

= 2, solve for pure resistance. Set initial mesh currents to solution.

NUMXST = total external coil sets which are to be included in solution.

(If INSTEP=0, no more data will be read for Job Step 2).

7. EXTERNAL SETS DATANUMXST lines
 (one line for each external coil set, in any order)

7.1 IF INSTEP = 1, then read step function data:.....1 line
 NEXT, AXOM
 FORMAT(I5,F15.4)

NEXT = external set number,

AXOM = current per turn at time zero minus.

7.2 IF INSTEP = 2, then read constant emf data:.....1 line
 NEXT, DIDT
 FORMAT(I5,F15.4)

NEXT = external set number,

DIDT = dI/dt (amperes/second) per turn.

V. JOB STEP 3 (INVERSION) INPUT

NO INPUT DATA IS REQUIRED FOR JOB STEP 3 (INVERSION).

VI. JOB STEP 4 (MAIN PROCESSOR) INPUT

TIME STEP CONTROL LINE(S).....* lines

NIN, NOUT, DT, TOL

FORMAT(2I5, 1PE10.3I, OPF10.4)

NIN = total steps in inner loop,

NOUT = total steps in outer loop,

DT = time increment of inner loop in seconds,

TOL = error tolerance for subroutine DGEAR (see IMSL documentation for further information). If zero or blank, default to 0.0001, which is almost universally applicable.

*Note: At time step no. 1 and at the end of each execution of the outer loop, the SPARK Main Processor will print and write onto disk the currently evaluated time and mesh current values. SPARK will continue time stepping from the last time step of each time step control line according to the specifications in the next time step control line for as many lines as are included in the data set. The number of data lines is subject to the constraint that the total number of cumulative outer loop steps (NOUT's) must be less than or equal to 500.

VII. JOB STEP 5 (POST PROCESSOR) INPUT

1.0 PARTS DATA

1.1 PARTS AND PRINT CONTROL LINE.....1 line

NUMPAR, NINTER, NTIME

FORMAT(3I5)

NUMPAR = total distinct parts (must be numbered from 1 to NUMPAR and ≤ 10)

= 0 + process all elements (skip to section 2);

NINTER = 0 + do not print intermediate calculation values,

= 1 + print intermediate calculation values;

NTIME = 0 + do not print intermediate costs,

= 1 + print intermediate costs.

1.2 PART DESCRIPTIONSNUMPAR lines

NPAR, NPNOS(NPAR), NPNOE(NPAR), NPELS(NPAR), NPELE(NPAR)

FORMAT(11I5)

NPAR = part number,

NPNOS = starting node number,

NPNOE = ending node number,

NPELS = starting mesh el. number,

NPELE = ending mesh el. number.

Note: If NPxxS(NPAR)=0+ no xx for part no. NPAR.

1.3 PART SETS CONTROL LINE.....1 line

NUMLP

FORMAT(I5)

NUMLP = total part sets (must be numbered from 1 to NUMLP and ≤ 10)

1.4 PART SET COMPOSITIONS.....NUMLP lines

LP, LPARTQ(LP), (MPARTQ(IP,LP), IP=1,LPARTQ(LP))

FORMAT(12I5)

LP = set number,

LPARTQ(LP) = total parts in set (≤ 10),

MPARTQ(IP,LP) = part numbers which compose set from 1 to LPARTQ(LP).

Note: The set of all elements is referred to as part number zero or NPART=0. If LPARTO(LP)=0, then set number LP is assumed identical to NPART=0.

2.0 SYMMETRIES DATA

2.1 SYMMETRIES CONTROL LINE.....line

NUMSYM

FORMAT(I5)

NUMSYM = total symmetries (must be numbered from 1 to NUMSYM).

(If NUMSYM = 0, skip to section 3.)

2.2 SYMMETRY DESCRIPTIONS.....NUMSYM lines

N, IXY(N), IYZ(N), IZX(N), JZ(N)

FORMAT (I5)

N = symmetry number;

IXY(N), IYZ(N), IZX(N) =

(+1 → mirror reflection)

(0 → no reflection) about xy, yz, or zx

(-1 → reverse mirror reflection) plane respectively;

JZ = number of symmetric repetitions about z axis (0 is same as 1).

A discussion of the symmetry specifications is included as an appendix to this manual.

Note: The set (IXY,IYZ,IZX) is the REFLECTION portion of the symmetry and the number JZ is the ROTATION portion.

2.3 SYMMETRY SET COMPOSITION FOR NPART=0.....1 line

ISYMT, (ISYM (II), II=1, ISYMT)

FORMAT (11I5)

ISYMT = total symmetries in set,

ISYM = symmetry numbers which compose set from 1 to ISYMT. (If ISYM < 0, use absolute value for actual symmetry number and reverse sign of surface normals for this symmetry in hidden surface determinations).

2.4 SYMMETRY SET COMPOSITION OF PARTS.....NUMPAR lines
 NPART, NSYMT(NPART), (NSYM(II,NPART), II=1, NSYMT(NPART))
 FORMAT (11I5)

NPART = part number,

NSYMT = total symmetries in set,

NSYM = symmetry numbers which compose set from 1 to NSYMT.

(If NSYM < 0, use absolute value for actual symmetry number and reverse sign of surface normals for this symmetry for this part in hidden surface determinations).

2.5 CONTROL CARD FOR SYMMETRIES TO BE EXHIBITED ON OUTPUT.....1 line
 NSYMEX
 FORMAT(15)

NSYMEX = 0 → do not exhibit any symmetries,

1 → exhibit some of all symmetries.

(If NSYMEX=0, skip to section 3)

2.6.0 DESCRIPTIONS OF SYMMETRIES TO BE EXHIBITED

2.6.1 REFLECTION PORTION EXHIBITION.....1 line
 (ISTEX(N), N=1,NUMSYM)
 FORMAT(10I5)

ISTEX = 0 → do not exhibit reflection,

= 1 → exhibit reflection (for symmetry numbers 1 to NUMSYM respectively).

2.6.2 ROTATION PORTION EXHIBITION.....1 line
 (JZTEX(N), N=1,NUMSYM)
 FORMAT(10I5)

JZTEX(N) = 0 → do not exhibit any rotations for symmetry N;

= JZHU * 100 + JZUN :

JZHU = 0 → exhibit rotations from 1 to JZUN
 for symmetry N;

JZHU > 0 → exhibit rotations from JZUN to JZUN + JZHU-1
 for symmetry N;

3.0 GRAPH AND CONTOUR DATA

3.1 GRAPH CONTROL CARD1 line

NGRAPH

FORMAT(I5)

NGRAPH = total graphs to be plotted at each time step (≤ 10).

(If NGRAPH = 0, skip to section 4.)

3.2 GRAPH PARAMETERS4*NGRAPH lines

(four lines for each Graph)

3.2.1 GRAPH PARAMETERS FIRST LINE.....1 line

IGRAPH, JGRAPH, KGRAPH, LP

FORMAT (4I5)

IGRAPH = IGRATE *10 + IGRAUN:

IGRAUN = 1 → graph only streamlines,

2 → graph meshes with streamlines,

3 → graph only streamlines,

4 → graph only streamlines;

IGRATE = ≥ 2 → posterize graph as IGRATE x IGRATE separate plots.

JGRAPH = 0 → omit mesh numbers,

≥ 1 → plot mesh numbers with character size = JGRAPH*0.6 inch, but only
if IGRAPH = 2.

KGRAPH = 0 → omit axes,

1 → plot axes.

LP = part set number to be graphed. If zero, defaults to entire system.

3.2.2 GRAPH PARAMETERS SECOND LINE.....1 line

CXPX, CXPY, CXP2, CYPX, CYPY, CYPZ

FORMAT (6F10.4)

CXPX,CXPY,CXPZ = direction cosines of x,y,z axes with abscissa of graph. If
all zero, defaults to (1,0,0).

CYPX,CYPY,CYPZ = direction cosines of x,y,z axes with ordinate of graph. If
all zero, defaults to (0,1,0).

3.3.3 GRAPH PARAMETERS THIRD LINE.....1 line

SX, XMIN, XMAX, YMIN, YMAX

Format (5F10.4)

If $SX < 0.0 \rightarrow$ scale graph with user supplied limits:

XMIN = minimum graph value of abscissa;

XMAX = maximum graph value of abscissa;

YMIN = minimum graph value of ordinate;

YMAX = maximum graph value of ordinate.

Scale streamline arrowhead size according to absolute value of SX.

(Standard size obtained with $SX=-1.0$).

If $SX = 0.0 \rightarrow$ scale graph from limits calculated by code.

Scale streamline arrowhead size according to value of XMIN.

(Standard size obtained with $XMIN=0.0$ or $XMIN=1.0$)

3.3.4 GRAPH PARAMETERS FOURTH LINE.....1 line

NZTROL, NCONTR, ZMIN, ZMAX

FORMAT(2I5,1P2E10.3)

NZTROL = 0 $\rightarrow$ normalize each part set to stream value extrema of part set itself at time selected for graphic display.

1 $\rightarrow$ normalize each part set to stream value extrema of all elements over all time points stored on File 40.

2 $\rightarrow$ normalize each part set to user supplied ZMIN and ZMAX.

3 $\rightarrow$ normalize each part set to stream value extrema of all elements at time point selected for graphic display

NCONTR = total stream contours to be plotted.

If $NCONTR=0$, only geometry will be plotted.

If $NCONTR$ is a positive number, then the stream contours will be determined by the relations:

$$DEL CV = (ZMAX - ZMIN) / (NCONTR + 1),$$

$$CV(N) = ZMIN + N * DEL CV, \text{ for } N=1 \text{ to } NCONTR.$$

If $NCONTR$ is a negative number, then the stream contours will be determined by the relations:

$$DEL CV = (ZMAX - ZMIN) / NCONTR,$$

$$CV(N) = ZMIN + (N - 0.5) * DEL CV, \text{ for } N=1 \text{ to } NCONTR.$$

ZMIN/ZMAX = minimum/maximum stream values.

ZMIN/ZMAX need only be supplied if $NZTROL=2$. Otherwise SPARK will determine their values.

4. PRIMARY FIELD CONTROL LINE.....1 line

NOFLD, NOPART, NUMREM, NRPART

FORMAT(4I5)

NOFLD = 0 + do not calculate any primary fields,
 = 1 + calculate primary fields with background coils only,
 = 2 + calculate primary fields with external coils only,
 = 3 + calculate primary fields with background and external coils.
 NOPART = part set for which nodal fields may be calculated (must be numbered
 from 1 to NUMREM).
 = 0 + calculate fields at all nodes,
 = -1 + do not calculate any nodal fields.
 NUMREM = total remote points at which fields may be calculated
 (must be numbered from 1 to NUMREM).
 NRPART = part set to be used as source of secondary fields,
 = 0 + calculate secondary fields from all elements,
 = -1 + do not calculate any secondary fields.

5. REMOTE FIELD POINT DATA.....NUMREM lines

(in any order)

NR, XREM, YREM, ZREM

FORMAT(I5,3F10.6)

NR = remote point number.
 (XREM,YREM,ZREM) = (x,y,z) coordinates of point in meters.

6. BACKGROUND COIL SETS CONTROL LINE.....1 line

NUMBCK

FORMAT(I5)

NUMBCK = total background coil sets (must be numbered from 1 to NUMBCK).
 If NUMBCK=0 + skip to section 9, time selection data.

7. BACKGROUND COIL SETS DATA

"background" coil sets are input using the same parameters as we
 described in Section IV.4, for "external" coil sets.

8. CURRENT PER TURN OF BACKGROUND SETS.....NUMBCK lines

NBCK, ABCK

FORMAT(I5,F15.2)

NBCK = background coil set number

ABCK = current per turn of set

9. TIME SELECTION DATA ** lines

(2 lines for each time selection)

****Note:** The SPARK post processor will continue reading as many pairs of time selection data lines as are included on text input file I5. When the data is exhausted, execution of Job Step 5 will stop.

9.1 TIME SELECTION FIRST LINE.....1 line

NSTBEG, NSTEND, NSTDEL, NHIDE, LAB, LCLIP, IBANDW, LVU, VUMIN, VUMAX

FORMAT (8I5, 2F10.5)

NSTBEG = first time step selected on FILE 40.

NSTEND = last time step selected on FILE 40.

NSTDEL = incremental time step increase between time step selections on FILE 40.

NHIDE = NHIDHU*100 + NHIDTE*10 + NHIDEUN :

NHIDHU = 0(1) + do not (do) compare surface normal vectors with view direction vector;

NHIDTE = 0(1) + do not (do) test for overlapping mesh centroids in plots coordinates;

NHIDEUN = 0(1) + do not (do) test for any intersection in plot coordinates of mesh sides.

LAB = -1 + do not plot any label;

0 + plot a short label on same frame;

1 + plot a long label on a separate frame.

LCLIP = 0(1) + turn clipping off (on).

IBANDW = not used in this edition, use zero as placeholder.

LVU = 0(1) + do not (do) limit plot to meshes that lie between VUMIN and VUMAX projected onto the view direction.

VUMIN/VUMAX = minimum/maximum projected view direction values.

9.2 TIME SELECTION SECOND LINE.....1 line

NGTROL, NBTROL, NRTROL, NFTROL

FORMAT(4I5)

NGTROL= -1 + do not exhibit graphs,

= 0 + exhibit graphs;

NBTROL= -1 + do not calculate branch currents,

= 0 + calculate branch currents;

NBTROL must be set to zero if nodal forces or remote fields are to be calculated.

NRTROL= -1 + do not calculate fields at remote points,

= 0 + calculate fields at remote points;

NFTROL= -1 + do not calculate nodal forces,

= 0 + calculate nodal forces and write values to text file 05 only,

= 1 + calculate nodal forces and write values to text files 05 and P5,

= 2 + calculate nodal forces and write values to text files P5, but
write only totals to 05,

= 3 + calculate nodal forces and write only totals to 05,

= 4 + do not calculate nodal forces, but write nodal fields to 05.

VIII. CONCLUSION

Version 1.1 of SPARK offers a significant increase in the types of surfaces that can be modeled with SPARK. Essentially any orientable surface that is within the size limitations of the code can be modeled. The code has been developed to provide increased user friendliness and convenience. The hidden surface removal option and the automatic graph scaling option exemplify these traits.

Other features are now under development and may be published if enough interest is found. These include anisotropic resistivities, power calculations with corresponding plots, and a major reduction in the size of the large scratch files in Step 1. An alternative version of the code that would allow more than two meshes per branch, but drop the streamline plotting option, is under consideration.

Users are encouraged to provide us with feedback about their experience with SPARK.

IX. ACKNOWLEDGMENTS

I would like to especially thank J. Bialek and U. Christensen of the Princeton Plasma Physics Laboratory for their advice and support in the development of this version of SPARK.

This work was supported by U. S. Department of Energy Contract No. DE-AC02-76-CH0-3073.

X. REFERENCES

- [1] D. W. Weissenburger, "SPARK Version One : Reference Manual," PPPL-2040, Plasma Physics Laboratory, Princeton University (1983) 230 pp.
- [2] IMSL manuals available from : IMSL, Inc., Sixth Floor - GNR Building, 7500 Bellaire Boulevard, Houston, Texas 77036.
- [3] NAG manuals available from : Numerical Algorithms Group, Ltd., Oxford, England; U. S. Address : 1131 Warren Avenue, Downers Grove, Illinois 60515.
- [4] K. O'Hair, P. Keller, J. Rowe, G. Cronshagen, "GRAFLIB User Manual," LCSD-425, Lawrence Livermore National Laboratory (1982) 186 pp.
- [5] F. R. A. Hopgood, D. A. Duce, J. R. Gallop, D. C. Sutcliffe, Introduction to the Graphical Kernel System (GKS), Academic Press, London (1983).
- [6] PLOT10 manuals available from : Tektronix, Inc., P. O. Box 500, Beaverton, Oregon 97077.
- [7] U. P. Christensen, "Time Varying Eddy Currents on a Conducting Surface in 3-D Using a Network Mesh Method," PPPL-1516, Plasma Physics Laboratory, Princeton University (1979) 36 pp.
- [8] D. W. Weissenburger and J. M. Bialek, "The Coupling of Mechanical Dynamics and Induced Currents in Plates and Structures," PPPL-2399, Plasma Physics Laboratory, Princeton University (1986) 32 pp.
- [9] D. W. Weissenburger and U. R. Christensen, "Transient Eddy Currents on Finite Plane and Toroidal Conducting Surfaces," PPPL-1517, Plasma Physics Laboratory, Princeton University (1979) 60 pp.
- [10] H. J. White and S. Tauber, Systems Analysis, pp.255-82, W. B. Saunders, Philadelphia (1969).
- [11] N. Balabanian and T. A. Bickart, Electrical Network Theory, Wiley, New York (1969).
- [12] L. Weinberg, Network Analysis and Synthesis, McGraw-Hill, New York (1962).
- [13] D. W. Weissenburger and U. R. Christensen, "A Network Mesh Method to Calculate Eddy Currents on Conducting Surfaces," IEEE Trans. Magn., MAG-18, pp.422-5 (1982)
- [14] D. W. Weissenburger and J. M. Bialek, "An Interface Between Eddy Current Calculation and Structural Analysis," IEEE Trans. Magn., MAG-19, pp.2619-22 (1983).

- [15] J. M. Bialek and D. W. Weissenburger, "Analyses of Eddy Current Loadings in Fusion Engineering Structures," Trans. 7th Int. Conf. Structural Mechanics in Reactor Technology, Chicago, 1983, Vol. N, pp. 163-9, North-Holland, Amsterdam (1983).
- [16] D. W. Weissenburger and H. M. Fan, "Dynamic Analysis of Eddy Currents and Forces in the TFTR Vacuum Vessel Due to Plasma Disruptions," IEEE Trans. Magn., MAG-21, pp.2347-50 (1985).
- [17] J. C. Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., articles 282a-b and 347, Clarendon Press, Oxford, England (1892), reprinted Dover, New York (1954).
- [18] G. Kron, Tensor Analysis of Networks, Wiley, New York (1939).
- [19] A. Bossavit and J. C. Verite, "A Mixed FEM-BIEM Method to Solve 3-D Eddy Current Problems," IEEE Trans. Magn., MAG-18, pp.431-5 (1982).
- [20] C. S. Harrold and J. Simkin, "Cutting Multiply Connected Regions," IEEE Trans. Magn., MAG 21, pp.2495-8 (1985).
- [21] P. A. Firby and C. F. Gardiner, Surface Topology, p. 25, Halsted Press, Wiley, New York (1982).
- [22] E. A. Lord and C. R. Wilson, The Mathematical Description of Shape and Form, p. 44, Halsted Press, Wiley, New York (1984).

APPENDIX A: SYMMETRY SPECIFICATIONS

It is impractical to model most physical systems with SPARK without taking advantage of symmetry. Even coarse models of many nonsymmetric systems produce enormous matrix sizes and consume large amounts of computation time to yield results with limited accuracy. In contrast many systems of interest are naturally symmetric or near enough to model symmetrically for most purposes. In this case a reduced model of the system will produce identical solutions to the full model and the matrix size necessary to solve a given problem can often become dramatically smaller. Correspondingly, a much finer reduced model of a system can also be employed to achieve higher accuracy results. The symmetry conventions used for SPARK 1.1 are the same as those that were originally presented in Ref. [1] for SPARK 1.0. This Appendix summarizes the types of symmetries that can be specified in SPARK and presents some basic examples for each type.

The basic symmetry principles used by SPARK were given by Christensen [7] and additional examples can be found in Ref. [9]. Three basic requirements must be met in order to reduce a system by symmetry:

- (1) The geometry of the conductors carrying the eddy currents must be symmetric.
- (2) The geometry of the magnetic fields driving the eddy currents must be symmetric and of the same degree or higher than (1).
- (3) The geometries in (1) and (2) must coincide.

In practice only the reduced geometry is input to SPARK and the code generates the remaining geometry from the symmetry specifications. Mathematically this produces no change in the mesh resistance matrix, but alters the inductance matrix to account for the mutual inductances to the additional geometry. The internal size of the matrices in SPARK remain unchanged with these procedures and thus the necessary computer memory allocation is determined by the size of the final reduced system as initially input to the code.

Three fundamental symmetry specifications are recognized by SPARK :

- (1) Positive reflection about the x-y, y-z, and/or z-x planes; i.e.,
 $IXY = +1$, $IYZ = +1$, and/or $IZX = +1$, respectively.
- (2) Negative reflection about the x-y, y-z, and/or z-x planes. i.e.,
 $IXY = -1$, $IYZ = -1$, and/or $IZX = -1$, respectively.

- (3) JZ-fold rotation about the z-axis, where the initial geometry is repeated at JZ locations equally spaced around the z-axis, including the initial geometry, e.g., JZ = 20.

SPARK refers to a set of symmetry specifications (IXY, IYZ, IZX, JZ) in abbreviated fashion as a "symmetry". If no symmetry specifications are given for a particular mesh, the symmetry is assumed to be: (0, 0, 0, 0). If a mesh has several symmetries, this default symmetry must be included as a distinct symmetry or it will be omitted from the calculations.

Figure A.1 illustrates positive reflection for an example quadrilateral labelled (1,2,3,4). The direction of positive mesh current is from 1 to 2 to 3 to 4 to 1. The reflection is about the y-z plane and is specified in SPARK as symmetry (0,1,0,0), i.e., IYZ = +1. For each node with coordinates (x,y,z) a reflected node is generated with coordinates (-x,y,z). The reflected geometry is labelled (1',2',3',4') and the original direction of positive mesh current is preserved from 1' to 2' to 3' to 4' to 1' as in a true mirror.

Figure A.2 illustrates negative reflection for the same example quadrilateral shown in Fig. A.1. Again the direction of positive mesh current is from 1 to 2 to 3 to 4 to 1. The reflection is again about the y-z plane, but this time the symmetry is specified in SPARK as (0,-1,0,0), i.e., IYZ = -1. As before for each node with coordinates (x,y,z) a reflected node is generated with coordinates (-x,y,z). The reflected mesh is again labelled (1',2',3',4'), but in this case the direction of positive mesh current is reversed from the original. For the reflected mesh, positive mesh current is from 1' to 4' to 3' to 2' to 1'.

Figure A.3 illustrates a compound reflection again starting with the same example quadrilateral as in Figs. A.1-2. In this case the symmetry is specified as (0,-1,1,0). This indicates a negative reflection about the y-z plane and a positive reflection about the z-x plane. The order of reflection does not matter as shown in the figure. One could first reflect about the y-z plane and get intermediate geometry (A) indicated as (1',2',3',4') and then reflect this intermediate geometry about the z-x plane to get the desired reflected geometry indicated by (1'',2'',3'',4''). Alternatively one could reflect about the z-x plane first to get intermediate geometry (B) indicated by (1''',2''',4''',5''') and then reflect about the y-z plane to get the same desired reflected geometry (1'',2'',3'',4'').

Figure A.4 illustrates rotation using the same original quadrilateral as in Figs. A.1-3. In this case the symmetry is specified as $(0,0,0,4)$, i.e., $JZ = 4$. This indicates a pure rotation about the z-axis to produce a 4-fold repetition of the original geometry. In the previous three figures we started with one mesh and produced one reflected mesh. Here we start with one mesh and produce four rotated meshes.

Figure A.5 illustrates two symmetries combining reflection and rotation operations on one original mesh to model a complete square loop. In order to reduce this problem by symmetry, it is necessary that the driving field have a similar or higher degree of symmetry about the z-axis as the mesh symmetry. For example, a spatially uniform field in the z-direction would be acceptable and is available as an option in SPARK. The original mesh A is given by $(1,2,3,4)$ in positive mesh nodal order. The first symmetry is shown on the figure as $(0,0,0,4)$. This is a pure rotation producing three additional meshes indicated by (A) on the figure. The second symmetry is shown as $(0,0,-1,4)$. The negative reflection about the z-x plane produces the mesh B indicated by $(1',4',3',2')$ in positive mesh nodal order. The rotational part of the second symmetry produces the three additional meshes indicated by (B) on the figure. The same final result could have been obtained by reflecting each of the rotated A meshes about the z-x plane, but this alternative procedure is not presently used in SPARK calculations.

We can observe in Fig. A.5, by using the right-hand-rule convention, that the positive normal to all the meshes for both symmetries is in the positive z-direction. This is exactly the configuration that we would expect from a full model of the system consisting of eight meshes. Since the mesh currents in A and B will always be the same magnitude because of symmetry, then the currents flowing along the $(1,2)$ sides will always be cancelled by the currents flowing along the $(1',2')$ sides. Similarly the currents flowing along the $(3,4)$ sides will be cancelled by the $(3',4')$ sides. Thus the $(1,2)$ and $(3,4)$ sides are equipotential boundaries, i.e., any currents must flow across and not along these boundaries. In contrast the $(2,3)$ and $(1,4)$ sides have current flow always along and never across themselves and thus are stream boundaries. Usually the $(2,3)$ side would be designated a zero-stream-value boundary and the $(1,4)$ side a hole boundary. Boundary topics are more fully presented in Appendix B : Topological Considerations.

If the original mesh in Fig. A.5 were subdivided into a finer network of quadrilaterals and triangles, the same symmetry specifications for the original mesh would apply to each mesh of the subdivision. Similarly the symmetry specifications for one mesh of a subdivision that has uniform symmetry properties can be applied to all the remaining meshes in the subdivision. This characteristic can be useful in determining the correct symmetry specifications for a given problem such as shown in Fig. A.6.

Figure A.6(a) illustrates a singly-curved plate subjected to a driving magnetic field B_{ext} that is symmetric about the z-axis, but not necessarily uniform. Figure A.6(b) illustrates a typical eddy current distribution that would be obtained with a full model of this system. The plate is assumed to be subdivided into some arbitrary network. Figure A.6(c) shows a representative mesh element indicated by (1,2,3,4) in positive mesh nodal order. The geometrically corresponding nodes obtained by reflections about the x-y and z-x planes are indicated by (1',2',3',4'), (1'',2'',3'',4''), and (1''',2''',3''',4'''), where we have not yet determined the positive order of circulation.

Examination of Fig. A.6(b) shows that the circulation normal according to the right-hand rule is in the positive x-direction above the x-y plane and in the negative x-direction below the x-y plane. Thus we would expect (1,2,3,4) to have the same circulation normal as its reflection about the z-x plane, i.e. (1',4',3',2'). Similarly we would expect (1,2,3,4) to have the opposite direction of circulation as its reflection below the x-y plane, i.e., (1''',2''',3''',4'''). This yields a negative reflection about the z-x plane: $IZX = -1$; and a positive reflection about the x-y plane: $IXY = 1$. It is seen from the figure that the remaining reflection is a compound of the z-x and x-y reflections and positive mesh nodal order is given by (1'',2'',3'',4''). Only the reflection part of the symmetries has been determined in this example. It would still be possible to rotate each of the symmetries around the z-axis. For example, if there were 20 plates equally-spaced around the z-axis, then JZ would be set to 20 for each of the four symmetries.

In some cases the mesh circulation may not be as obvious as in this problem, but the direction of a branch current and its reflected counterpart may be apparent. For example, if we knew only that branch current (1,2) was in the opposite direction to (1',2') then we could immediately determine that

we had a negative reflection about the z - x plane as before. We could derive the specifications for the remaining symmetries in a similar fashion.

Inspection of Fig. A.6(b) also shows that the x - y plane is a stream boundary and the z - x plane is an equipotential boundary. Thus no currents flow across the x - y plane or along the z - x plane. Since the outer perimeter of the plate would normally be designated a zero-stream-value boundary, the reduced model would also include the branches along the x - y plane as part of the zero-stream-value boundary. Boundary topics are more fully presented in Appendix B : Topological Considerations.

#87E0036

Fig. A.1. Positive reflection example for a quadrilateral mesh.

#87E0034

Fig. A.2. Negative reflection example for a quadrilateral mesh.

#87E0035

COMPOUND REFLECTION

Fig. A.3. Compound reflection example for a quadrilateral mesh.

87E0032

ROTATION

Fig. A.4. Rotation example for a quadrilateral mesh.

87E0033

Fig. A.5. Combined reflection and rotation example for a quadrilateral mesh.

#87E0037

Fig. A.6. (a) Cross section of an example plate subjected to an axisymmetric driving field. (b) typical eddy current pattern that would flow on plate. (c) Representative mesh element in one-quarter-plate model and corresponding symmetries that would apply to remainder of plate.

APPENDIX B: TOPOLOGICAL CONSIDERATIONS

1. Network Topology and Surface Topology

Eddy current calculations in SPARK are based on the well-known network mesh method of modeling electrical systems. The principles of this method are thoroughly presented in many other works, for example see Refs. [10-12], and will only be summarized in this Appendix. In SPARK a surface is modeled as a lumped-parameter network of branches, where each branch has a resistance, a self-inductance, and a set of mutual inductances to all the other branches. The algorithms used to obtain these quantities in SPARK 1.1 are the same as in Version 1.0 and are described along with the various calculation options in Refs. [1, 13-16].

From the network branch model, a system of simultaneous differential equations can be written that relate the constant branch parameters to the instantaneous values of the branch emf's and currents. James Clerk Maxwell [17] pointed out that the number of equations to be solved is considerably reduced by expressing the network in terms of mesh properties instead of branch properties. Once a mesh solution is obtained, the branch currents can be extracted from the mesh currents with relatively simple mathematical manipulations. Gabriel Kron formalized this entire procedure using matrix/tensor representations before the availability of the digital computer [18]. Kron introduced the mesh-branch incidence matrix to relate the mesh and branch topologies and corresponding properties. This matrix formally consists of zeroes and ones (positive or negative) and is generally sparse. In SPARK this incidence matrix is stored in a compact form and algorithms have been developed to replace the usual matrix operations performed with the incidence matrix that avoid increasing computer storage requirements.

In general any well-defined branch-node network can be solved using the network mesh method, but some constraints must be applied to networks that can be handled by SPARK. In principle Version 1.1 can be used to model any orientable surface. This eliminates tee-intersections and mobius-like topologies although such systems are acceptable in general network topology. The restrictions come rather from surface topology considerations.

A very desirable feature of Version 1.1 is the streamline plot option in Step 5. This is possible because the mesh current values calculated by SPARK

for surface meshes can be equivalenced to stream values at the geometric centroid of each mesh. Thus the net current flowing across any line between two points is equal to the difference in their stream values. In traditional network analysis, the mesh currents are regarded as only mathematical conveniences to derive the "real" branch currents. However, the mesh currents themselves can have physical significance, if a surface is represented by a suitably chosen network. In addition, only certain surfaces can have well-defined stream values and thus the constraints on Version 1.1.

Other related modeling methods that employ scalar potentials require similar enhancements like those in SPARK to accommodate general topologies [19]. Some efforts have been made to automatically incorporate any necessary additional specifications so that a user can avoid inputting any special topological data altogether [20].

2. Plane Topology

The simplest surfaces that can be modeled with SPARK are topologically equivalent to a plane network. By topologically equivalent we mean that a given surface can be continuously deformed without tearing or cutting from its initial configuration into some second configuration. In formal terminology the two configurations are said to be homeomorphic.

Figure B.1(a) illustrates an example branch-node network with nodes numbered from 1 to 24 and branches numbered from 1 to 37. For the present we will not impose any SPARK restrictions, but consider this example from the standpoint of general network topology. We use the term mesh and loop equivalently to mean a closed series of non-intersecting branches. More thorough presentations can be found in the references. Some authors are more restrictive in the use of the term mesh. The well-known equation that all network mesh topologies must follow is

$$l + n = b + 1, \quad (\text{B.1})$$

where l = total loops, n = total nodes, b = total branches.

This equation holds for any electrically independent network, whether planar or nonplanar. Some authors use n to represent the total independent node pairs, which is just our n minus one, and modify Eq.(B.1) accordingly. While Eq.(B.1) is a necessary condition that a network topology must satisfy, it is not sufficient to insure that a given system of loop equations is

independent and, thus, a mathematically acceptable representation of the network. This is equivalent to requiring that the rank of the mesh-branch incidence matrix is equal to the total loops, l .

Figure B.1(b) shows a possible tree structure for the example branch-node network in (a). For a given branch-node network, a tree is a set of branches that connects all the nodes, but does not form any loops. The remaining branches of the network that are not part of the tree are collectively called the cotree and separately called links. If each link is considered individually with the tree, when a link is added to the tree network, a unique loop will be formed that consists of the link itself and part of the tree as shown in Fig. B.1(c). Thus for each loop there is a link through which the branch current is identical to the loop current. Such a loop is referred to as a fundamental loop. Because all the loops in (c) are fundamental loops, the system of loop equations characterized by the mesh-branch incidence matrix for (c) is guaranteed to be independent and thus a mathematically acceptable representation of the network. This general method of creating a system of fundamental loops from a tree can be applied to any general network whether planar or nonplanar.

Figure B.1(d) shows an alternative acceptable loop topology. The loops in this topology are not all fundamental loops and no tree can be constructed to extract this system of loops. Each of the loops in (d) has no internal branches. We refer to such loops as canonical meshes. It is well-known that a planar network consisting only of canonical meshes produces an independent set of loop equations.

In contrast, Fig. B.1(e) shows an unacceptable loop topology. Although the total loops in (e) is 14 as in (c) and (d), the system of loop equations is not mathematically independent. This can be verified by examination of loop number 7. Loop current 7 is the only current flowing through branch 26 and branch 37. Thus this system of equations would only apply if the currents in branch 26 and 37 were always identical, but this is not true in general for this network. Determining the acceptability of a loop topology is not always as easy as in this case.

Figure B.1(f) shows another acceptable loop topology that should be compared with (d). Topologically all the loops are still canonical meshes as in (d). This leads us to modeling surfaces using the SPARK constraints.

3. Zero-Stream-Value Boundaries and Hole Loops

In SPARK, surfaces must be represented by networks of triangles or quadrilaterals. Holes in the surface must be included as hole loops which can consist of an arbitrary number of branches where each branch is input to SPARK in the direction of positive loop current circulation. All the loops in this type of model are canonical meshes. The surface must be orientable, that is all the surface meshes and hole loops must have consistent circulations. This can be accomplished by imposing a requirement for an outward-pointing normal for example.

If the networks in Fig. B.1(d) and (f) represented a surface with a hole, they would both be acceptable SPARK models. The difference between the two is that in (d), the hole loop is on the inner perimeter, while in (f), the hole loop is on the outer perimeter. In (d) we have implicitly assigned the stream value zero to the outer boundary and we must specify this information to SPARK by designating the branches on the outer perimeter as belonging to a zero-stream-value boundary. The direction in which each branch is assigned to a zero-stream-value boundary does not matter. Correspondingly each node on the boundary is given a stream value of zero, if any streamline plots are made. Similarly the branches on the inside perimeter are assigned to a hole loop and a mesh current equivalent to a stream value is calculated by SPARK for this loop. Each node of the hole loop is assigned this stream value, if any streamline plots are made.

For comparison the system of Fig. B.1(f) designates the outer perimeter as the hole loop and the inner perimeter as the zero-stream-value boundary. All the other meshes remain the same. Thus we expect that the mesh current solutions obtained for the two cases should differ only by some constant stream value. The streamline plots and branch currents would be the same in both cases. In fact we can determine what the stream values in (f) would be if we knew the stream values for (d). The current in loop 14 in case (f) would be the negative of the value in (d), because the circulation of loop 14 in (f) is opposite to (d). Similarly, all the other mesh currents in (f) would be equal to their (d) values reduced by the loop 14 value in (d).

4. Sphere-like Topology and Closure Meshes

It is readily seen that a planar network with no holes is topologically equivalent to a box with one face open as shown in Fig. B.2(a). Modeling the box with 8 nodes and 12 branches, a loop topology acceptable to SPARK is shown in (b) on the figure. There is one loop for each face of the box for a total of 5. The branches on the open face must be designated as belonging to a zero-stream-value boundary. If the open face of the box were closed as shown in (c), the number of branches and nodes would remain the same as for the open box. Thus the number of loops must also remain the same or Eq. (B.1) would not be satisfied. It follows that the loop topology shown in (d) on the figure is not an acceptable SPARK model.

The closed box is topologically equivalent to a sphere or ball, which is topologically distinct from a plane. It is impossible to transform a sphere into a plane without making a cut. The six loops shown for the closed box are not all independent. In order to create an acceptable loop network model of the closed box we must remove one of the six loops. This is done in SPARK by specifying the NUMCM highest numbered meshes as closure meshes. In this case, $\text{NUMCM} = 1$, and loop 6 would be automatically ignored in mathematical manipulations, but its parameters would be saved for streamline plotting purposes. This is equivalent to choosing mesh 6 to have a stream value of zero. Since mesh 6 is a surface mesh, this means that the centroid of mesh 6 is assigned the value zero and not the perimeter nodes as for a zero-stream-value boundary. Any closure meshes designated in Step 1 of SPARK require no further user interaction and are automatically incorporated into any streamline plotting performed in Step 5.

There is nothing significant about the mesh numbering shown in Fig. B.2(d). The meshes can be numbered arbitrarily and any loop can serve as mesh 6. The difference between the open box model and the closed box model is in the inductance and resistance matrices. The details of associating rectangular plates with the edges of surface meshes in order to calculate resistances and inductances can be found in Section II of this manual. In the open box model, the branches around the open face consist of only one plate, while all the other branches have two plates. In the closed box model, all the branches have two plates.

If one had a sphere-like model with a single hole, it would be modeled using a zero-stream-value boundary around the hole, but the resulting resistance matrices would become very close to a closed model as the hole became smaller and smaller. Any additional holes in the surface would be included in the SPARK model as hole loops. This is generally true for any surface: if any holes are present, no closure mesh is required. If a closure mesh is used, it must be equivalent to a regular surface mesh, that is, a triangle or rectangle. It is possible, but not generally recommended, to add a hole loop to a closed sphere topology containing a closure mesh instead of dropping the closure mesh designation and defining the hole loop as a zero-stream-value boundary.

5. Handles and Cut Loops

The final elementary topology that we require to model a general surface is called a handle. Figure B.3.(a) illustrates a representative handle topology being added to some initial network topology. We assume that the initial loop topology is well-defined and satisfies Eq.(B.1).

The figure shows only quadrilateral elements, but these can be readily generalized to more general surfaces including triangles and holes. Consider first adding a diagonal branch to a quadrilateral, thus producing two triangles. No new nodes are added, but one additional loop is added to the left side of Eq.(B.1) and one additional branch is added to the right side. Thus the equation remains in balance. Secondly, consider a loop, that is not part of the original topology of a well-defined network and that has an arbitrary number of branches surrounding a single interior node in such a way that the loop is subdivided into a set of triangles and/or quadrilaterals. If the interior node and all the interior branches are removed from the original network and replaced by the outer loop; for each interior loop removed on the left of Eq.(B.1), one interior branch is removed on the right; and the one loop added is balanced by the single node removed. Thirdly, consider the same configuration as just described except that the node to be removed lies on the boundary and not on the interior. In this case one less loop is removed on the left but one branch must also be added on the right to replace the two branches removed on the boundary, so balance is still maintained.

These three addition/removal procedures are each reversible and through a combination of them any general equivalent topology can be created starting from a basic representation such as shown in Fig. B.3.

Because the figure only contains quadrilaterals, the number of nodes around the perimeter at each of the handles is the same. In the general case, they could be different. Let us consider that the initial topology has been modified to obtain two hole loops, as shown in the figure, with total nodes NA and NB respectively that match the nodes on each end of the handle to be added. Let $L2$, $N2$, and $B2$ be the loops, nodes, and branches in the handle that is added to the modified initial topology. The handle topology is equivalent to a cylinder which in turn is equivalent to a plane with a hole. The handle would have a well-defined topology if one end were a hole loop and the other a zero-stream-value boundary. But the number of loops actually added to the original network is one less than a well-defined network and we can write

$$L2 + N2 = B2 . \quad (B.2)$$

When the two topologies are combined, the nodes and branches at each end of the handle are removed to be replaced by their equivalent original network counterparts. Since for each node removed there is a branch, $N2$ and $B2$ are both reduced by $(NA+NB)$ and the balance of Eq.(B.2) is maintained. However, the two hole loops in the modified original topology are not replaced and the combined network is two loops short of a well-defined topology.

We would like to keep the mesh topology that exists so far in order to be able to produce streamline graphics, but add two additional loops to satisfy Eq.(B.1). As with any general topology, the final set of loops must yield an independent system of equations to be acceptable. This can be accomplished as shown in Fig. B.3(b) where the two loops are labelled A and B and are called cut loops because they cut through the surface. Cut loops are analogous to cuts in surface topology, that is they are the cuts one would have to make in a surface in order to unfold it into a plane. In this case, if we assume the initial surface had a spherical topology for example, the resulting combined surface would be topologically equivalent to a sphere with one handle or torus as shown in Fig. B.4(a). It is well-known that a torus requires one cut around its minor circumference and one cut around its major circumference to be unfolded.

In SPARK cut loops can have an arbitrary number of branches and can cross each other, but they cannot share any nodes with hole loops or zero-stream-value boundaries. Cut loops correspond to discontinuities in stream values and required additional algorithm development for the streamline graphics in Step 5. Fortunately this is invisible to the user and taken care of automatically. Cut loops and hole loops are designated in Step 1 and require no further user interaction. The interested reader is referred to the code listing itself for further information on the plotting techniques. If one examines the example plots with cut loops carefully, sometimes a slight discontinuity is seen at a cut. This is because the stream value discontinuity is not always an exact integer multiple of the contour interval. This is not an error, but one of the limits on contouring surfaces.

Formally most topologists define a handle as removing not two discs but only one from the original surface. This is convenient for some topological proofs, but in our application the original surface elements would be replaced with identical elements and the same resultant surface is attained using either definition. Our definition was chosen to have more practical utility in building models.

Any orientable surface is either topologically equivalent to a plane or a sphere with n -handles [21,22]. Therefore we can theoretically model any general surface using the tools available in SPARK. We put no restrictions on the original surface to which we attached a handle, so the original surface could have been a plane or a torus or any well-defined topology. If it had been a plane, we would have produced a torus with a hole; if it had been a torus, we would have produced a torus with one handle otherwise known as a sphere with two handles. The presence of holes in the surface does not alter the basic topology.

One additional special topology deserves mention, this is the degenerate torus illustrated in Fig. B.4(b). The original network in (a) is pinched to a single node by removing nodes and branches, but no loops. We have removed X branches, but only $X-1$ nodes from the original well-defined topology. Therefore, we need to remove one loop from the original network to balance Eq.(B.1). The loop that needs to be removed is cut loop B and we require only cut loop A for this topology.

6. Equipotential Boundaries and Topologies Reduced By Symmetry

Appendix A: Symmetry Specifications presented the various symmetry options available in SPARK that can often be used to reduce a given network to an equivalent smaller model. In many cases the reduced system has a much simpler topology than the initial model. For example, a torus with top-bottom symmetry can be replaced by the top half only, which is topologically equivalent to a plate with a hole. Whether the topology changes or not, the reduced system must still meet the same mathematical requirements as the full models. Specifically, Eq.(B.1) must be satisfied and the loop topology must produce an independent set of equations.

Consider first a simple plane model as shown in Fig. B.5(a). We will assume that this system has a left-right symmetry such that on the dividing line between the left and right halves, current only flows across the line and not along the line. The dividing line is thus an equipotential, that is, the electrical potential or voltage is constant along the line. This corresponds to an electrical short. This means that locations immediately opposite each other across an equipotential line must have equal stream values and, therefore, the current through the branch on the equipotential line between them has zero current.

The topology of the symmetrically reduced model is shown in Fig. B.5(b). Branches that are specified to lie on an equipotential boundary are ignored for resistance and inductance calculations by SPARK. Equipotential boundaries and zero-stream-value boundaries are designated only in Step 1 and require no further user interaction. If any streamline plotting is performed in Step 5, nodal values are automatically assigned by SPARK that account for any boundary specifications. Nodes on a zero-stream-value boundary are, of course, assigned a zero value. Nodes on an equipotential boundary are assigned values that are interpolated between adjacent mesh centroid values that are projected perpendicularly onto the boundary. If a node lies on both an equipotential and a zero-stream-value boundary, the zero value is used. Similarly if a node lies on both an equipotential and a hole loop, the hole loop value is used.

The elimination of the branches on equipotential boundaries for calculational purposes in SPARK corresponds topologically to replacing all the nodes on the boundary by a single equivalent node as illustrated in Fig. B.5(c). Here the resulting topology is clearly seen to be a planar system of canonical meshes and thus a completely acceptable SPARK topology. This fact is perhaps not as readily apparent upon first examination of (b).

Now let us consider a plane with a hole as shown in Fig. B.6(a). We will assume the same left-right symmetry as before and the left-right dividing line is again an equipotential line. The reduced model would be input to SPARK as shown in (b). Although from (b) it might appear that there are two distinct equipotential boundaries, there actually is only one. In SPARK 1.0 dummy equipotential branches were required across the hole itself, but this is no longer necessary. In both the first model without any hole and the second with a hole, we have the same symmetry. In both cases the equipotential dividing line could be shorted by a perfectly conducting wire along its entire length and there would be no effect on the current distribution.

The equivalent topology for Fig. B.6(b) is shown in (c). As in Fig. B.5(c) we have a single equivalent node replacing all the nodes on the equipotential boundary and again we have an acceptable planar network of canonical meshes. The same resulting topology in Fig. B.6(c) could have been obtained directly from Fig. B.5(c) by substituting the desired hole in place of the applicable surface meshes.

Next let us consider the closed torus illustrated in Fig. B.7(a) with cut loops A and B. Let us assume that as in the first two examples we can divide the torus into two symmetrical halves about the plane containing cut loop B. We again assume that currents flow only across and not along the dividing plane, so that the plane represents an equipotential boundary. The reduced model that would be input to SPARK is shown in (b). Let the original total surface meshes be $2*L2$, the original total cut loops be C , the original total nodes not on the boundary be $2*N2$, and the original total branches not on the boundary be $2*B2$. Similarly, let the total nodes on perimeter one be PA and the total nodes on perimeter two be PB , where $PA + PB = P2$. The total branches on perimeter one and perimeter two will also be equal to $P2$. Since the torus is a closed surface, one of the surface meshes must be designated a closure mesh to obtain an acceptable topology. We can then write Eq.(B.1) for the original topology as

$$2*L2 - 1 + C + 2*N2 + P2 = 2*B2 + P2 + 1. \quad (B.3)$$

Since $C = 2$ for the original topology, we can rewrite Eq.(B.3) in the form

$$L2 + N2 = B2. \quad (B.4)$$

In line with the first two examples, we expect that shorting all the nodes on the equipotential plane will have no effect on the current distribution and that all the nodes on the equipotential plane can be replaced by a single equivalent node as shown in Fig. B.7(c). We can then use Eq.(B.1) to write for the equivalent topology

$$L2 - 1 + C + N2 + 1 = B2 + 1 . \quad (B.5)$$

Substituting Eq.(B.4) we get

$$C = 1 . \quad (B.6)$$

As shown in (c), only cut loop A is required in the reduced model. This topology is identical to the degenerate torus topology illustrated in Fig. B.4(b). In general whenever a network with a handle is reduced by symmetry such that the handle is divided across its minor circumference by an equipotential boundary, a degenerate handle topology arises in the reduced network.

Figure B.8(a) shows a network model that consists of a rectangular plate to which three handles have been added. In this example we will assume that the system has the same kind of symmetry about the z-x plane as in the previous three examples and has an additional symmetry about the x-y plane. The z-x symmetry is divided by an equipotential boundary and the x-y symmetry is divided by a stream boundary. Eddy currents flow only across and not along the equipotential plane and eddy currents flow only along but not across the stream plane. The symmetry specifications for this system are identical to those shown in Fig. A.6. The figure shows five cut loops labeled A-E. The perimeter of the rectangular plate is designated a zero-stream-value boundary and thus no closure mesh is required with this topology.

Figure B.8(b) shows the original system reduced by the symmetry about the z-x plane. The original handle determined by cut loops E and F in (a) has been reduced to a degenerate handle topology requiring only one cut loop that is relabeled as A in (b). The other two original handles in (a) are replaced by a single handle with cut loops that are relabeled as B and C in (b). An equivalent topology for (b) would be to replace all the nodes on the equipotential boundaries by one equivalent node.

Figure B.8(c) shows the original system reduced by the symmetry about the z - x plane and the symmetry about the x - y plane. We can derive this configuration by reducing the topology in (b) about the x - y plane. The x - y plane is a stream boundary in this case and cannot be replaced by a single equivalent node as an equipotential boundary. Where the x - y plane divides the rectangular plate becomes part of the zero-stream-value boundary. Where the x - y plane divides the handle that was determined by cut loops B and C in (b) a hole loop is required that is labeled as B in (c). If we replaced the system in (c) by an equivalent topology, the nodes on the equipotential boundaries could be replaced by a single equivalent node. The resulting topology is equivalent to a degenerate torus with two holes. Therefore, we require one cut loop that is labeled as A on (c). One of the holes is the zero-stream-value boundary and the other is hole loop B.

Figure B.8(d) shows a variation on the topology in (c). The full topology including all the symmetries would produce two parallel rectangular plates connected by four L-shaped legs. The topology in (d) is obtained by unfolding hole loop B in (c) and adding the additional hole loop C. This produces a topology equivalent to a degenerate torus with three holes in (d). One of the holes is still the zero-stream-value boundary, and the other two holes are hole loops.

Fig. R.1 (a) Example branch-node network. (b) Possible tree structure.

(c) Fundamental Loop Topology From Tree

(d) Acceptable Loop Topology

(e) Unacceptable Loop Topology

(f) Acceptable Loop Topology

Fig. B.1 (c) Fundamental loop topology derived from tree in (b). (d) An alternative acceptable loop topology. (e) An unacceptable loop topology. (f) Another acceptable loop topology.

87E0130

(a) Box With
One Side Open

(b) Branch-Node Network
For Open Box With
Acceptable Loop
Topology

(c) Closed Box

(d) Branch-Node Network
For Closed Box With
Unacceptable Loop
Topology

Fig. R.2 (a) Example box with one side open. (b) Branch-node network for open box with acceptable loop topology. (c) Example closed box. (d) Branch-node network for closed box with unacceptable loop topology.

87E0133

Fig. B.3 (a) Representative handle topology being added to some arbitrary initial network topology.

87E0128

Fig. B.3 (b) Combined handle and initial network topology.

87EO127

(a) Full Torus Topology

Fig. B.4 (a) Representative full torus topology.

87E0126

(b) Degenerate Torus Topology

Fig. 8.4 (b) Representative degenerate torus topology.

87E0125

(a) Original
Network :
Zero-Stream
Boundary on
Perimeter

(b) Symmetrically
Reduced
Network

(c) Equivalent
Topology
to (b)

Fig. A.5 (a) Example full model for a rectangular plate.
(b) Symmetrically-reduced model equivalent to (a).
(c) Network topology equivalent to (b).

87E0131

Fig. 8.6 (a) Example full model for a rectangular plate with a hole.
 (b) Symmetrically-reduced model equivalent to (a).
 (c) Network topology equivalent to (b).

87E0129

(a) Original Network With Two Cut Loops : A & B

Fig. B.7 (a) Example full model of a closed torus.

87E0124

(b) Symmetrically Reduced Network With One Cut Loop Only : A

Fig. 8.7 (b) Symmetrically-reduced model equivalent to (a).

87E0123

(c) Equivalent Topology to (b)

Fig. 8.7 (c) Network Topology equivalent to (h).

87E0132

Fig. R.8 (a) Example full model of rectangular plate with three handles.

Fig. B.8 (b) Equivalent model to (a) obtained by reducing model (a) about z-x plane of symmetry.

87E0120

Fig. R.8 (c) Equivalent model to (a) obtained by reducing model (b) about x-y plane of symmetry.

APPENDIX C: FEATURES OF VERSION 1.1

1. Array Dimensions

For each of the five Job Steps of SPARK 1.1 the maximum model size dimensions are set using PARAMETER statements at the beginning of each main program. In some cases parameters need to be correspondingly set in some subprograms as well. When this is necessary, there are clear comments at the main program PARAMETER statements directing the programmer to the required subprograms.

The standard editions of SPARK are issued with the following initial PARAMETER specifications :

```

NDSIZ = 400 , maximum surface meshes;
NOSIZ = 1000, maximum nodes;
NHSIZ = 10,  maximum hole and/or cut loops;
NVSIZ = 100, maximum branches per hole or cut loop;
NESIZ = 200, maximum branches on equipotential boundaries;
NSSIZ = 200, maximum branches on zero-stream-value boundaries;
NYSIZ = 10,  maximum different materials and maximum definable
               symmetries;
NCSIZ = 500, maximum driver and/or background coils;
NKSIZ = 10,  maximum background coil sets;
NRSIZ = 100, maximum remote field points;
NTSIZ = 100, maximum driver time points;
NXSIZ = 100, maximum driver coil sets;
NWSIZ = 500, maximum integration time steps at which sets of mesh
               currents can be written onto FILE40.

```

For convenience two additional parameters are defined from the primary list:

```

NMSIZ = NDSIZ + NHSIZ (maximum loops);
NBSIZ = 4*NDSIZ (maximum branches).

```

2. Input Formats

SPARK 1.1 of SPARK was written to be compatible with input that may have been created using the 1.0 conventions. The earlier conventions were developed to be compatible with the structural analysis code SAP.

Unfortunately, this produced an unfriendly rigidity in the input format requirements. The new format conventions still can read the earlier form, but allow a more straightforward input style, mainly by eliminating required blanks.

The new format conventions also allow a nearly format-free input style on the MFE Cray's and the local PPPL Vax's. Blank-delimited or comma-delimited lists can be substituted for all formats in 1.1 provided integers are input for integers and reals are input for reals. This alternative is not obtainable with all other available compilers and rigid adherence to the 1.0/1.1 formats may be required. If a user on such a machine desires to eliminate this rigidity, all the input read statements can be replaced with true format-free read specifications. But, if the code is so modified, it will no longer be fully compatible with Version 1.0.

3. Step 1: Three Substeps, Plot-Only Option, Rename Option

The six master control parameters in Version 1.0 have been reduced to three : NSTEPA, NSTEPB, NSTEPC. The number of nodal points, NUMNP, is input on a separate line if required. If NSTEPA = 0, no new input geometry will be read and the old geometry will be retrieved from disk. If NSTEPB = 1, however, new graphics data will be read. NSTEPC controls the inductance and resistance calculations. If only a heading line and no other data is present, SPARK will rename the existing old set of data with the new heading, but make no other changes.

4. Step 1 : Closure Meshes and Cut Loops

Version 1.1 can accept significantly more complicated topologies than was possible with Version 1.0 through the use of closure meshes and cut loops. These options allow essentially any orientable surface to be modeled. Both options are described in detail in Appendix B: Topological Properties. Once designated in Step 1, closure meshes, cut loops, hole loops, zero-stream-value boundaries, and equipotential boundaries require no further user interaction.

5. Step 1: Increased Plot Options

Several new display options are included in Step 1 that are specified with the IGRAPH parameter. Examples of many of these options can be found in Appendices D and E. A major new option has also been added for the graph scaling parameters that are now specified with the plot minimum and maximum values (XMIN, XMAX, YMIN, YMAX). If $SX = -1.0$, the user supplied values will be used; if $SX = 0.0$, SPARK will calculate the scaling by itself. SPARK will calculate a scaling such that the graph is centered on the plot, including axes if specified, and such that the plot size is maximized, but isometric. Similarly, if the direction cosines (CXPX, CXPY, CXPZ, and CYPX, CYPY, CYPZ) are input as zero, the default view direction will be looking down along the user-z-axis, with user-x-axis on the plot-x-axis and user-y-axis on the plot-y-axis.

6. Step 2 : Increased Driver Options, Axisymmetry Correction

There are now five basic driver options available in Step 2 : a set of coaxial circular coils with the same current per turn, a set of generally-oriented circular coils with the same current per turn, spatially uniform field in any specified direction, a $1/R$ type field about the z-axis, and a set of straight line segments with each segment carrying the same current. An arbitrary number of each type can be defined as necessary. Driver types are designated with the parameter ICOAX.

In Version 1.0 the magnetic vector potential produced from the uniform field option was not axisymmetric and this made it necessary to close any hole loops with dummy equipotential branches. The magnetic vector potentials in Version 1.1 are all axisymmetric and it is no longer necessary to include dummy equipotential branches.

7. Step 2: Variable Coil Option Deleted

The vestiges of the variable coil option that remained in Version 1.0 have been eliminated in 1.1. It is no longer necessary to designate a line specifying $NUMVAR = 0$. If any older data sets are used that include this specification, the line will be automatically ignored.

8. Step 4 : Modified Adams Integration

Version 1.1 runs on the MFE Cray's and our local Vax's with a variable step Adams integration routine that is obtained from either the IMSL or NAG libraries. The two routines are not exactly identical and a user can substitute any integration routine he desires. The automatic step size adjustment is very convenient for the transient problems we usually encounter. In almost all cases the IMSL routine will adjust the desired output step size to some smaller internal value when starting integration and transmit a warning message to the user that a change was made. By itself these warnings can be ignored, but users are always urged to review their results with care to insure that no genuine errors have occurred.

9. Step 5: Modified Nodal Stream Value Interpolation Algorithm

The algorithms for interpolating the stream values from the mesh centroids to the nodes has been modified to eliminate ambiguities produced with sharp edges and corners in Version 1.0. The new algorithm is actually a less complicated procedure and this means faster calculations and a more robust behavior in general.

The basic technique of the new algorithm is to associate a quadrilateral sub-area with each mesh adjoining a node. Mesh centroids for triangles and quadrilaterals are now both determined as the average of the nodal coordinates. A sub-area is defined by the node itself, the mesh centroid, and the mid-points of the the two sides of the mesh adjacent to the node. For each mesh adjoining a node, the mesh stream value is weighted by the inverse of its respective sub-area. The weighted average of all the stream values determines the nodal value. The complete implementation of the new algorithms can be found in the source code for Step 5.

10. Step 5 : Hidden Surface Removal Options

A major new option available in Version 1.1 is hidden-surface- removal. The algorithms that have been developed for this purpose are relatively simple and hence relatively fast in their computation. As such, however, they do not always produce a perfect removal, but do produce a fairly good approximation. Unfortunately hidden-surface-removal becomes exponentially more costly at each new level of accuracy. For purposes of analysis it is expected that the options available in SPARK will be more than satisfactory.

There are three levels of calculation available in SPARK for hidden-surface-removal. In order of increasing calculational complexity, the first is the comparison of the normal vector of each surface mesh to the view direction. Each mesh will be plotted or not plotted according to whether the dot product of the two vectors is positive or negative. The sub-option of plotting only the "invisible" surfaces is also available.

The second and third options involve projecting the centroid of each surface mesh onto the view direction. This arranges the meshes in an order from farthest to nearest on the view direction, but is not without ambiguity. This arrangement can be used by itself on some terminals by plotting the farthest meshes first so that later meshes "paint over" the earlier meshes and is known as the Painter's Algorithm. Since solid mesh coloring is not a generic option of Version 1.1, this procedure is not directly applicable here.

The second option is to test for overlapping mesh centroids. This means that if the mesh centroid of a farther mesh is inside the projected area of a nearer mesh or if the centroid of a nearer mesh is inside the projected area of the farther mesh, the farther mesh is not plotted. This generally removes less than a perfect number of surface meshes, but may actually enhance interpretation on occasion.

The third option is to test for the overlap of a farther mesh anywhere within a nearer mesh. If any overlap is detected, then the farther mesh is removed. This procedure generally removes more than a perfect number of surfaces, but again may produce the most readily interpretable graph in some cases.

Each of the three options, may be used by themselves or in combination with the others. With any of the options, it often makes a significant difference when a small shift is made in the viewing direction.

All the surface plotting options that were available in Step 5 of Version 1.0 are no longer available in 1.1. Presently streamlines can be plotted on a surface that is indicated by branch lines or streamlines can be plotted without any surface indications at all.

11. Step 5: Posterization Option

A useful convenience feature in many cases is the posterization option in Step 5. When a user has determined the desired view parameters for a particular graph in the usual single frame size, the same graph can be produced as an n by n set of plots automatically in Version 1.1. The resulting set of plots does need to be assembled by hand. Step 1 does not presently have this feature, but Step 5 does now have a no-streamline option by setting `NCONTR = 0`.

APPENDIX D: SPARK EXAMPLE INPUT

RI1 = Step 1: Rectangular Sheet.
RI2 = Step 2: Rectangular Sheet.
RI4 = Step 4: Rectangular Sheet.
RI5 = Step 5: Rectangular Sheet.
TI1 = Step 1: 1/8 Square Plate.
TI2 = Step 2: 1/8 Square Plate.
TI4 = Step 4: 1/8 Square Plate.
TI5 = Step 5: 1/8 Square Plate.
VI1 = Step 1: 9 Sector Torus.
VI2 = Step 2: 9 Sector Torus.
VI4 = Step 4: 9 Sector Torus.
VI5 = Step 5: 9 Sector Torus.

SAMPLE R11

[illegible][illegible]

Table D.RI1(a) Example Step 1 input.

SAMPLE R11

```

      1.0000    0.0    0.0    0.0    0.0    1.0000    0.0
-1.0    0.0    4.5    0.0    4.5
      4      3      0
      1.0000    0.0    0.0    0.0    1.0000    0.0
-1.0    0.0    4.5    0.0    4.5
>>>>END R11 [INPUT FILE FOR STEP 1]

```

```

0061
0062
0063
0064
0065
0066

```

Table D.R11(b) Example Step 1 input.

SAMPLE R12

```

SAMPLE RI2 [INPUT FILE FOR STEP 2]<<<<<<<<<<<<<<<<<<<<<<<<<<<<
2
1      0.0
2      1.0
1
1      1      ODIPOLE-LIKE COIL ON AXIS
2.5    2.0    2.0    0.0    0.0    1.0
0.05   0.05   0.0    0.0    100000.
1
1      -100000.
2
1      -100000.
1      1      0
1      0.000
>>>>>END RI2 [INPUT FILE FOR STEP 2]

```

Table D.R12. Example Step 2 input.

SAMPLE R14

```
SAMPLE R14 [INPUT FILE FOR STEP 4]<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
25 12 1.0000E-5
>>>>END R14 [INPUT FILE FOR STEP 4]
```

0082
0083
0084

Table D.R14. Example Step 4 input.

SAMPLE R15

```

SAMPLE R15 [INPUT FILE FOR STEP 5]<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
0
0
1
2      0
1.0    0.0    0.0    0.0    1.0    0.0
-1.0   0.0    4.5    0.0    4.5
1      4
3      0    0    0      (NOFLD, NOPART, NUMREM, NRPART)
1
1      1    2
0.0    1.0    0.0
1      1.0000
1      13    6
0      0    -1    0
>>>>>END R15 [INPUT FILE FOR STEP 5]

```

Table D.R15. Example Step 5 input.

0101
0102
0103
0104
0105
0106
0107
0108
0109
0110
0111
0112
0113
0114
0115
0116
0117
0118
0119
0120
0121
0122
0123
0124
0125
0126
0127
0128
0129
0130
0131
0132
0133
0134
0135
0136
0137
0138
0139
0140
0141
0142
0143
0144
0145
0146
0147
0148
0149
0150
0151
0152
0153
0154

Table D.T11. Example Step 1 input.

```
SAMPLE T12 [INPUT FILE FOR STEP 2]<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
2
1      0.0
2      1.0
1
1      1      2
0.0      0.0      1.0
1
1      0000000.00      0
2
0
1      1      0
1      +1.000000000
>>>>END T12 [INPUT FILE FOR STEP 2]
```

0155
0156
0157
0158
0159
0160
0161
0162
0163
0164
0165
0166
0167
0168

Table D.TI2. Example Step 2 input.

```
SAMPLE T14 [INPUT FILE FOR STEP 4]<<<<<<<<<<<<<<<<<<<<<<<<<<<<
25 12 1.0000E-5
>>>>END T14 [INPUT FILE FOR STEP 4]
```

0169
0170
0171

Table N.TI4. Example Step 4 input.

```
SAMPLE TIS [INPUT FILE FOR STEP 5]<<<<<<<<<<<<<<<<<<<<<<< 0172  
 0      0      0                                0173  
 2                                         0174  
 1      0      0      0      4                0175  
 2      0      0     -1      4                0176  
 2      1      2                               0177  
 1                                           0178  
 1      1                                       0179  
 4      4                                       0180  
 1                                           0181  
 2      0                                       0182  
 1.0          0.0          0.0          0.0          1.0          0.0        0183  
-1.0         -0.500        0.625         -0.500        0.625              0184  
 0      4                                          0185  
 1      0      0      0             {NOFLD, NOPART, NUMREW, NRPART}    0186  
 1                                           0187  
 1      1      2                               0188  
 0.0          1.0          0.0                    0189  
 1      1.0000                                  0190  
 1      13      6                              0191  
 0      0      0                              0192  
>>>>>END TIS [INPUT FILE FOR STEP 5] 0193
```

Table D.T15. Example Step 5 input.

Table D.VII(a) Example Step 1 input.

57	3.4166	-0.9139	-0.7392	0254			
58	3.4773	-0.6131	-0.7392	0255			
59	3.5175	-0.3677	-0.7392	0256			
60	3.5310	0.0000	-0.7392	0257			
61	3.0305	-1.1030	-0.9959	0258			
62	3.1151	-0.8347	-0.9959	0259			
63	3.1760	-0.5600	-0.9959	0260			
64	3.2127	-0.2811	-0.9959	0261			
65	3.2250	0.0000	-0.9959	0262			
66	2.6778	-0.9747	-1.1325	0263			
67	2.7526	-0.7376	-1.1325	0264			
68	2.8064	-0.4948	-1.1325	0265			
69	2.8389	-0.2484	-1.1325	0266			
70	2.8497	0.0000	-1.1325	0267			
71	2.3025	-0.8381	-1.1325	0268			
72	2.3668	-0.6342	-1.1325	0269			
73	2.4131	-0.4255	-1.1325	0270			
74	2.4410	-0.2136	-1.1325	0271			
75	2.4503	0.0000	-1.1325	0272			
76	1.9499	-0.7097	-0.9959	0273			
77	2.0043	-0.5370	-0.9959	0274			
78	2.0435	-0.3603	-0.9959	0275			
79	2.0671	-0.1808	-0.9959	0276			
80	2.0750	0.0000	-0.9959	0277			
81	1.6524	-0.6051	-0.7392	0278			
82	1.7088	-0.4579	-0.7392	0279			
83	1.7422	-0.3072	-0.7392	0280			
84	1.7623	-0.1542	-0.7392	0281			
85	1.7690	0.0000	-0.7392	0282			
86	1.4747	-0.5368	-0.3933	0283			
87	1.5159	-0.4062	-0.3933	0284			
88	1.5455	-0.2725	-0.3933	0285			
89	1.5634	-0.1368	-0.3933	0286			
90	1.5694	0.0000	-0.3933	0287			
72	1	0	1	0288			
1				0289			
7.6e-7				0290			
1	1	2	7	6	1	0.012700	0291
2	2	3	8	7	1	0.012700	0292
3	3	4	9	8	1	0.012700	0293
4	4	5	10	9	1	0.000127	0294
5	6	7	12	11	1	0.012700	0295
6	7	8	13	12	1	0.012700	0296
7	8	9	14	13	1	0.012700	0297
8	9	10	15	14	1	0.000127	0298
9	11	12	17	16	1	0.012700	0299
10	12	13	18	17	1	0.012700	0300
11	13	14	19	18	1	0.012700	0301
12	14	15	20	19	1	0.000127	0302
13	16	17	22	21	1	0.012700	0303
14	17	18	23	22	1	0.012700	0304
15	18	19	24	23	1	0.012700	0305
16	19	20	25	24	1	0.000127	0306
17	21	22	27	26	1	0.012700	0307
18	22	23	28	27	1	0.012700	0308
19	23	24	29	28	1	0.012700	0309
20	24	25	30	29	1	0.000127	0310
21	26	27	32	31	1	0.012700	0311
22	27	28	33	32	1	0.012700	0312
23	28	29	34	33	1	0.012700	0313

Table D.VII(b) Example Step 1 input.

SAMPLE VII

24	29	30	35	34	1	0.000127	0314
25	31	32	37	36	1	0.012700	0315
26	32	33	38	37	1	0.012700	0316
27	33	34	39	38	1	0.012700	0317
28	34	35	40	39	1	0.000127	0318
29	36	37	42	41	1	0.012700	0319
30	37	38	43	42	1	0.012700	0320
31	38	39	44	43	1	0.012700	0321
32	39	40	45	44	1	0.000127	0322
33	41	42	47	46	1	0.012700	0323
34	42	43	48	47	1	0.012700	0324
35	43	44	49	48	1	0.012700	0325
36	44	45	50	49	1	0.000127	0326
37	46	47	52	51	1	0.012700	0327
38	47	48	53	52	1	0.012700	0328
39	48	49	54	53	1	0.012700	0329
40	49	50	55	54	1	0.000127	0330
41	51	52	57	56	1	0.012700	0331
42	52	53	58	57	1	0.012700	0332
43	53	54	59	58	1	0.012700	0333
44	54	55	60	59	1	0.000127	0334
45	56	57	62	61	1	0.012700	0335
46	57	58	63	62	1	0.012700	0336
47	58	59	64	63	1	0.012700	0337
48	59	60	65	64	1	0.000127	0338
49	61	62	67	66	1	0.012700	0339
50	62	63	68	67	1	0.012700	0340
51	63	64	69	68	1	0.012700	0341
52	64	65	70	69	1	0.000127	0342
53	66	67	72	71	1	0.012700	0343
54	67	68	73	72	1	0.012700	0344
55	68	69	74	73	1	0.012700	0345
56	69	70	75	74	1	0.000127	0346
57	71	72	77	76	1	0.012700	0347
58	72	73	78	77	1	0.012700	0348
59	73	74	79	78	1	0.012700	0349
60	74	75	80	79	1	0.000127	0350
61	76	77	82	81	1	0.012700	0351
62	77	78	83	82	1	0.012700	0352
63	78	79	84	83	1	0.012700	0353
64	79	80	85	84	1	0.000127	0354
65	81	82	87	86	1	0.012700	0355
66	82	83	88	87	1	0.012700	0356
67	83	84	89	88	1	0.012700	0357
68	84	85	90	89	1	0.000127	0358
69	86	87	2	1	1	0.012700	0359
70	87	88	3	2	1	0.012700	0360
71	88	89	4	3	1	0.012700	0361
72	89	90	5	4	1	0.000127	0362
1							0363
1	-4						0364
5	4						0365
4	3						0366
3	2						0367
2	1						0368
0							0369
36							0370
1	1	6					0371
2	6	11					0372
3	11	16					0373

Table D.VII(c) Example Step 1 input.

SAMPLE VII

4	16	21					0374
5	21	26					0375
6	26	31					0376
7	31	36					0377
8	36	41					0378
9	41	46					0379
10	46	51					0380
11	51	56					0381
12	56	61					0382
13	61	66					0383
14	66	71					0384
15	71	76					0385
16	76	81					0386
17	81	86					0387
18	86	1					0388
19	5	10					0389
20	10	15					0390
21	15	20					0391
22	20	25					0392
23	25	30					0393
24	30	35					0394
25	35	40					0395
26	40	45					0396
27	45	50					0397
28	50	55					0398
29	55	60					0399
30	60	65					0400
31	65	70					0401
32	70	75					0402
33	75	80					0403
34	80	85					0404
35	85	90					0405
36	90	5					0406
2							0407
1	0	0	0	9			0408
2	0	0	-1	9			0409
1							0410
1	72	2	1	2			0411
12	2						0412
1	0	0	1	4			0413
2	0	0	69	72			0414
-1	2	1					0415
0.3827	-0.9239	0.0	0.0	0.0	1.0		0416
-1.0	0.0	1.08	-0.5	0.58			0417
2	0	1					0418
0.707	0.707	0.0	-0.5	0.5	0.707		0419
-1.0	0.0	3.6	-3.0	0.6			0420
3	0	1					0421
0.3827	-0.9239	0.0	0.0	0.0	1.0		0422
-1.0	-0.5	3.1	-1.2	2.4			0423
-3	2	1					0424
0.3827	-0.9239	0.0	0.0	0.0	1.0		0425
-1.0	-0.5	1.3	-1.2	0.6			0426
-4	2	1					0427
0.3827	-0.9239	0.0	0.0	0.0	1.0		0428
-1.0	0.0	1.08	-0.5	0.58			0429
6	0	1					0430
0.707	0.707	0.0	-0.5	0.5	0.707		0431
-1.0	0.0	3.6	-3.0	0.6			0432
8	3	0	2	1	2		0433

Table D.VII(d) Example Step 1 input.

SAMPLE VII

0.3827	-0.9239	0.0	0.0	0.0	1.0	0434
-1.0	0.54	1.08	-0.50	0.58		0435
-13	2	1				0436
0.3827	-0.9239	0.0	0.0	0.0	1.0	0437
-1.0	-0.5	1.3	-1.2	0.6		0438
-14	3	0				0439
0.3827	-0.9239	0.0	0.0	0.0	1.0	0440
-1.0	0.54	1.08	-0.50	0.58		0441
-15	2	1				0442
0.3827	-0.9239	0.0	0.0	0.0	1.0	0443
-1.0	-0.5	1.3	-1.2	0.6		0444
16	3	0	2	1	2	0445
0.3827	-0.9239	0.0	0.0	0.0	1.0	0446
-1.0	0.54	1.08	-0.50	0.58		0447
18	3	0	2	1	2	0448
0.3827	-0.9239	0.0	0.0	0.0	1.0	0449
-1.0	0.54	1.08	-0.50	0.58		0450
>>>>>END VII [INPUT FILE FOR STEP 1]						0451

Table D.VII(e) Example Step 1 input.

SAMPLE V12

[illegible]

Table 0.V12(a) Example Step 2 input.

SAMPLE V12

2.6500	2.6500	0.5500	0.5500	1.0000		0512
13	1	1 VERTICAL PLASMA	DISRUPTION; STEP 13 OF 21			0513
0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0514
2.6500	2.6500	0.6000	0.6000	1.0000		0515
14	1	1 VERTICAL PLASMA	DISRUPTION; STEP 14 OF 21			0516
0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0517
2.6500	2.6500	0.6500	0.6500	1.0000		0518
15	1	1 VERTICAL PLASMA	DISRUPTION; STEP 15 OF 21			0519
0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0520
2.6500	2.6500	0.7000	0.7000	1.0000		0521
16	1	1 VERTICAL PLASMA	DISRUPTION; STEP 16 OF 21			0522
0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0523
2.6500	2.6500	0.7500	0.7500	1.0000		0524
17	1	1 VERTICAL PLASMA	DISRUPTION; STEP 17 OF 21			0525
0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0526
2.6500	2.6500	0.8000	0.8000	1.0000		0527
18	1	1 VERTICAL PLASMA	DISRUPTION; STEP 18 OF 21			0528
0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0529
2.6500	2.6500	0.8500	0.8500	1.0000		0530
19	1	1 VERTICAL PLASMA	DISRUPTION; STEP 19 OF 21			0531
0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0532
2.6500	2.6500	0.9000	0.9000	1.0000		0533
20	1	1 VERTICAL PLASMA	DISRUPTION; STEP 20 OF 21			0534
0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0535
2.6500	2.6500	0.9500	0.9500	1.0000		0536
21	1	1 VERTICAL PLASMA	DISRUPTION; STEP 21 OF 21			0537
0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0538
2.6500	2.6500	1.0000	1.0000	1.0000		0539
1						0540
1	3000000.000000					0541
0						0542
2						0543
2	2850000.000000					0544
0						0545
3	2700000.000000					0546
0						0547
4						0548
4	2550000.000000					0549
0						0550
5						0551
5	2400000.000000					0552
0						0553
6						0554
6	2250000.000000					0555
0						0556
7						0557
7	2100000.000000					0558
0						0559
8						0560
8	1950000.000000					0561
0						0562
9						0563
9	1800000.000000					0564
0						0565
10						0566
10	1650000.000000					0567
0						0568
11						0569
11	1500000.000000					0570
						0571

Table D.VI2(h) Example Step 2 input.

SAMPLE VI2

0	0572
12	0573
12 1350000.000000	0574
0	0575
13	0576
13 1200000.000000	0577
0	0578
14	0579
14 1050000.000000	0580
0	0581
15	0582
15 900000.000000	0583
0	0584
16	0585
16 750000.000000	0586
0	0587
17	0588
17 600000.000000	0589
0	0590
18	0591
18 450000.000000	0592
0	0593
19	0594
19 300000.000000	0595
0	0596
20	0597
20 150000.000000	0598
0	0599
21	0600
21 0.000000	0601
0	0602
22	0603
0	0604
0	0605
>>>>END VI2 [INPUT FILE FOR STEP 2]	0606

Table D.VI2(c) Example Step 2 input.

SAMPLE VI 4

```
SAMPLE VI4 [INPUT FILE FOR STEP 4]<<<<<<<<<<<<<<<<<<<<<<<<<<<<
1 100 1.000e-04
>>>>END VI4 [INPUT FILE FOR STEP 4]
```

0607
0608
0609

Table D.VI4. Example Step 4 input.

SAMPLE V15

Table N.VI5. Example Step 5 input.

APPENDIX E: SPARK EXAMPLE GRAPHIC OUTPUT

R01 = Step 1: Rectangular Sheet.

R05 = Step 5: Rectangular Sheet.

T01 = Step 1: 1/8 Square Plate.

T05 = Step 5: 1/8 Square Plate.

V01 = Step 1: 9 Sector Torus.

V05 = Step 5: 9 Sector Torus.

RECTANGULAR SHEET WITH CENTERED HOLE

10/14/87 12:44:47 PST Cray e

NG= 1

. 13	. 14	. 15	. 16
. 9	. 10	. 11	. 12
. 5	. 6	. 7	. 8
. 1	. 2	. 3	. 4

Fig. F.R01.1. Example Step 1 output.

RECTANGULAR SHEET WITH CENTERED HOLE

10/14/87 12:44:47 PST Cray e

NG= 2

B18 S9		B22 S8		B24 S7	
S10					S6
B19		B17		B21	B23
	B12	H1	B20		B15
S11				H1	S5
B13		B11		B16	B14
		H1			
	B3		B7 H1		B10
S12					S4
B4		B2		B6	B9
	B1 S1		B5 S2		B8 S3

Fig. E.R01.2. Example Step 1 output.

RECTANGULAR SHEET WITH CENTERED HOLE

10/14/87 12:44:47 PST Cray e

NG= 3

Fig. F.R01.3. Example Step 1 output.

RECTANGULAR SHEET WITH CENTERED HOLE

10/14/87 12:44:47 PST Cray e

NG= 4

Fig. E.R01.4. Example Step 1 output.

10/14/87 12:33:42 PST Cray e

NG= 1 NP= 0 T= 0.

RECTANGULAR SHEET WITH CENTERED HOLEContour (4 total) increment= $0.424\text{e}+07$ Amperes

Fig. E.R05.1. Example Step 5 output.

10/14/87 12:33:42 PST Cray e

NG= 1 NP= 0 T= 0.150e-02.

RECTANGULAR SHEET WITH CENTERED HOLE

Contour (4 total) increment= 0.424e+07 Amperes

Fig. E.R05.2. Example Step 5 output.

10/14/87 12:33:42 PST Cray e

NG= 1 NP= 0 T= 0.300e-02

RECTANGULAR SHEET WITH CENTERED HOLE

Contour (4 total) increment= 0.424e+07 Amperes

Fig. E.R05.3. Example Step 5 output.

1/8 SQ PLATE:NDIV= 3 SIDE= 0.5000 RHO= 5.200E-07 TK= 6.350E-03

10/08/87 11:02:35 PST Cray e

NG= 1

. 10

. 8 . 9

. 5 . 6 . 7

Fig. E.T01.1 Example Step 1 output.

1/8 SQ PLATE:NDIV= 3 SIDE= 0.5000 RHO= 5.200E-07 TK= 6.350E-03

10/08/87 11:02:35 PST Cray e

NG= 2

Fig. F.T01.2 Example Step 1 output.

1/8 SQ PLATE:NDIV= 3 SIDE= 0.5000 RHQ= 5.200E-07 TK= 6.350E-03

10/08/87 11:02:35 PST Cray e

NG= 3

Fig. E.T01.3. Example Step 1 output.

1/8 SQ PLATE:NDIV= 3 SIDE= 0.5000 RHO= 5.200E-07 TK= 6.350E-03

10/08/87 11:02:35 PST Cray e

NG= 4

Fig. F.T01.4. Example Step 1 output.

10/14/87 12:23:37 PST Cray e

NG= 1 NP= 0 T= 0.

1/8 SQ PLATE:NDIV= 3 SIDE= 0.5000 RHO= 5.200E-07 TK= 6.350E-03

Contour (4 total) increment= 0.114e+06 Amperes

Fig. F.T05.1. Example Step 5 output.

10/14/87 12:23:37 PST Cray e

NG= 1 NP= 0 T= 0.150e-02

1/8 SQ PLATE:NDIV= 3 SIDE= 0.5000 RHO= 5.200E-07 TK= 6.350E-03

Contour (4 total) increment= 0.406e+05 Amperes

Fig. E.T05.2. Example Step 5 output.

10/14/87 12:23:37 PST Cray e NG= 1 NP= 0 T= 0.300e-02.

1/8 SQ PLATE:NDIV= 3 SIDE= 0.5000 RHO= 5.200E-07 TK= 6.350E-03

Contour (4 total) increment= 0.139e+05 Amperes

Fig. E.T05.3. Example Step 5 output.

9 SECTOR TORUS WITH CUT LOOP

10/14/87 12:59:48 PST Cray e

NG= 1

Fig. F.V01.1. Example Step 1 output.

9 SECTOR TORUS WITH CUT LOOP

10/14/87

12:59:48 PST

Cray e

NG= 2

Fig. E.V01.2. Example Step 1 output.

9 SECTOR TORUS WITH CUT LOOP

10/14/87 12:59:48 PST Cray c

NG= 3

Fig. E.V01.3. Example Step 1 output.

9 SECTOR TORUS WITH CUT LOOP

10/14/87 12:59:48 PST Cray e

NG= 4

Fig. E.V01.4. Example Step 1 output.

9 SECTOR TORUS WITH CUT LOOP

10/14/87 12:59:48 PST Cray e

NG= 5

Fig. E.V01.5. Example Step 1 output.

9 SECTOR TORUS WITH CUT LOOP

10/14/87

12:59:48 PST

Cray e

NG= 6

Fig. E.V01.6. Example Step 1 output.

9 SECTOR TORUS WITH CUT LOOP

10/14/87 12:59:48 PST Cray e

NG= 7

Fig. E.V01.7. Example Step 1 output.

9 SECTOR TORUS WITH CUT LOOP

10/14/87 12:59:48 PST Cray e

NG= 8

Fig. E.V01.8. Example Step 1 output.

9 SECTOR TORUS WITH CUT LOOP

10/14/87 12:59:48 PST Cray e

NG= 9

Fig. F.V01.9. Example Step 1 output.

9 SECTOR TORUS WITH CUT LOOP

10/14/87 12:59:48 PST Cray e

NG= 11

Fig. E.V01.11. Example Step 1 output.

9 SECTOR TORUS WITH CUT LOOP

10/14/87 12:59:48 PST Cray e

NG= 12

Fig. E.V01.12. Example Step 1 output.

10/14/87 13:33:32 PST Cray e

NG= 1 NP= 0 I= 0.100e-03

9 SECTOR TORUS WITH CUT LOOP

Local min=-0.945e+05 max= 0.530e+04 Amperes

Contour (19 total) increment= 0.525e+04 Amperes

Global min=-0.876e+06 max= 0.730e+05 Amperes

Fig. E.V05.1(a) Example Step 5 output.

10/14/87 13:33:32 PST Cray e

NG= 2 NP= 0 T= 0 100e-03

9 SECTOR TORUS WITH CUT LOOP

Local min=-0.945e+05 max= 0.530e+04 Amperes

Contour (19 total) increment= 0.525e+04 Amperes

Global min=-0.876e+06 max= 0.730e+05 Amperes

Fig. E.V05.1(b) Example Step 5 output.

10/14/87 13:33:32 PST Cray e

NG= 1 NP= 0 T= 0 100e-02

9 SECTOR TORUS WITH CUT LOOP

Local min=-0.572e+06 max= 0.661e+05 Amperes

Contour (19 total) increment= 0.336e+05 Amperes

Global min=-0.876e+06 max= 0.730e+05 Amperes

Fig. E.V05.2(a) Example Step 5 output.

10/14/87 13:33 32 PSI Cray e

NG= 2 NP= 0 T= 0 100e-02

9 SECTOR TORUS WITH CUT LOOP

Local min=-0.572e+06 max= 0.661e+05 Amperes

Contour (19 total) increment= 0.336e+05 Amperes

Global min=-0.876e+06 max= 0.730e+05 Amperes

Fig. E.V05.2(b) Example Step 5 output.

10/14/87 15:33:32 PST Cray e

NC= 1 NP= 0 T= 0.200e-02

9 SECTOR TORUS WITH CUT LOOP

Local min=-0.767e+06 max= 0.542e+05 Amperes

Contour (19 total) increment= 0.432e+05 Amperes

Global min=-0.876e+06 max= 0.730e+05 Amperes

Fig. E.V05.3(a) Example Step 5 output.

10/14/87 13:33:32 PST Cray e

NG= 2 NP= 0 T= 0.200e-02

9 SECTOR TORUS WITH CUT LOOP

Local min=-0.767e+06 max= 0.542e+05 Amperes

Contour (19 total) increment= 0.432e+05 Amperes

Global min=-0.876e+06 max= 0.730e+05 Amperes

Fig. E.V05.3(b) Example Step 5 output.

10/14/87 15:33 32 PST Cray e

NC= 1 NP= 0 I= 0 300e-02

9 SECTOR TORUS WITH CUT LOOP

Local min=-0.873e+06 max= 0.223e+05 Amperes

Contour (19 total) increment= 0.471e+05 Amperes

Global min=-0.876e+06 max= 0.730e+05 Amperes

Fig. E.V05.4(a) Example Step 5 output.

10/14/87 15:33 32 PSF Cray e

NG= 2 NP= 0 I= 0 300e-02

9 SECTOR TORUS WITH CUT LOOP

Local min=-0.873e+06 max= 0.223e+05 Amperes

Contour (19 total) increment= 0.471e+05 Amperes

Global min=-0.876e+06 max= 0.730e+05 Amperes

Fig. F.V05.4(h) Example Step 5 output.

10/14/87 13:33 32 PST Cray e

NG= 1 NP= 0 I= 0 100e-01

9 SECTOR TORUS WITH CUT LOOP

Local min=-0.279e+05 max= 0.259e+05 Amperes

Contour (19 total) increment= 0.283e+04 Amperes

Global min=-0.876e+06 max= 0.730e+05 Amperes

Fig. E.V05.5(a) Example Step 5 output.

10/14/87 13:33:32 PST Cray e

NG= 2 NP= 0 T= 0.100e-01

9 SECTOR TORUS WITH CUT LOOP

Local min=-0.279e+05 max= 0.259e+05 Amperes

Contour (19 total) increment= 0.283e+04 Amperes

Global min=-0.876e+06 max= 0.730e+05 Amperes

Fig. E.V05.5(b) Example Step 5 output.

APPENDIX F: COIL SET SPECIFICATION PARAMETERS

SPARK uses the Basic Magnetic Field Design Library to calculate magnetic fields from circular coils. These routines were principally written by U.R. Christensen at PPPL, but at present have only been documented in unpublished reports. The conventions used by SPARK are illustrated in Fig. F.1 and are only slightly modified from the original coding. Figure F.1(a) shows an example three-coil coaxial set in the global coordinate system. The local coordinate system for the set is defined by the local origin (O_X, O_Y, O_Z) and the local axis unit vector $\mathbf{n}$ defined by the direction cosines (D_X, D_Y, D_Z) . Most of the example cases for SPARK have local coil set origins identical to the global origin, $(O_X, O_Y, O_Z) = (0.0, 0.0, 0.0)$, and a local axis parallel to the global z-axis, $(D_X, D_Y, D_Z) = (0.0, 0.0, 1.0)$.

Figure F.1(b) shows a cross-sectional view of the coil set along the local axis. The minor cross section of each of the three coils is rectangular. The coil dimensions with respect to the local origin are defined by inner radius A_1 , outer radius A_2 , bottom location on axis Z_1 , and top location on axis Z_2 . Each of the coils in a set can have a different number of turns, but the current per turn in each of the coils must be the same.

Besides rectangular other minor cross sections can be defined using the local coordinates A_1, A_2, Z_1, Z_2 . If $A_1 = A_2$ and $Z_1 = Z_2$, SPARK will use the appropriate formulas for an ideal filament. If $Z_1 = Z_2$, but $A_2 > A_1$, then ideal pancake formulas will be used. If $A_1 = A_2$, but $Z_2 > Z_1$, then ideal solenoid formulas will be used. Semi-infinite ideal solenoids can be defined as described in Section IV. If $A_1 > A_2$, SPARK assumes that the coil has a circular minor cross section with minor radius A_2 and major radius A_1 .

Generally oriented coil sets are defined in essentially the same way except that a local origin and axis must be defined for each coil in the set.

The Basic Magnetic Field Design Library includes a dummy FUNCTION CURDEN (R) to avoid a warning message on loading. This function defines the current density of a coil as a function of coil major radius. SPARK nominally assumes the current density of a coil to be uniform across its cross section and does not call this function. If the product of the number of turns and the current per turn of a coil exceeds 1.0×10^{20} , however, SPARK will expect that the user has supplied a FUNCTION CURDEN (R) when making calculations.

Fig. F.1 (a) Example three-coil coaxial coil set in the global coordinate system. (b) Cross-sectional view of the coil set along the local coordinate axis.

EXTERNAL DISTRIBUTION IN ADDITION TO UC-20

- Dr. Frank J. Paoloni, Univ of Wollongong, AUSTRALIA
- Prof. M.H. Brennan, Univ Sydney, AUSTRALIA
- Plasma Research Lab., Australian Nat. Univ., AUSTRALIA
- Prof. I.R. Jones, Flinders Univ., AUSTRALIA
- Prof. F. Cap, Inst Theo Phys, AUSTRIA
- Prof. M. Heindler, Institut fur Theoretische Physik, AUSTRIA
- M. Goossens, Astronomisch Instituut, BELGIUM
- Ecole Royale Militaire, Lab de Phys Plasmas, BELGIUM
- Commission-European, Dg-XII Fusion Prog, BELGIUM
- Prof. R. Boucique, Laboratorium voor Natuurkunde, BELGIUM
- Dr. P.H. Sakanaka, Instituto Fisica, BRAZIL
- Instituto De Pesquisas Espaciais-INPE, BRAZIL
- Documents Office, Atomic Energy of Canada Limited, CANADA
- Dr. M.P. Bachynski, MPB Technologies, Inc., CANADA
- Dr. H.M. Skarsgard, University of Saskatchewan, CANADA
- Dr. H. Barnard, University of British Columbia, CANADA
- Prof. J. Telchmann, Univ. of Montreal, CANADA
- Prof. S.R. Sreenivasan, University of Calgary, CANADA
- Prof. Tudor W. Johnston, INRS-Energie, CANADA
- Dr. C.R. James, Univ. of Alberta, CANADA
- Dr. Peter Lukac, Komenskaho Univ, CZECHOSLOVAKIA
- The Librarian, Culham Laboratory, ENGLAND
- The Librarian, Rutherford Appleton Laboratory, ENGLAND
- Mrs. S.A. Hutchinson, JET Library, ENGLAND
- C. Mouttet, Lab. de Physique des Milieux Ionises, FRANCE
- J. Radet, CEN/CADARACHE - Bat 506, FRANCE
- Univ. of Ioannina, Library of Physics Dept. GREECE
- Dr. Tom Muai, Academy Bibliographic Ser., HONG KONG
- Preprint Library, Hungarian Academy of Sciences, HUNGARY
- Dr. B. Dasgupta, Saha Inst of Nucl. Phys., INDIA
- Dr. P. Kaw, Institute for Plasma Research, INDIA
- Dr. Philip Rosenau, Israel Inst. Tech, ISRAEL
- Librarian, Int'l Ctr Theo Phys, ITALY
- Prof. G. Rostagni, Univ Di Padova, ITALY
- Miss Clotilde De Palo, Assoc EURATOM-ENEA, ITALY
- Biblioteca, Instituto di Fisica del Plasma, ITALY
- Dr. H. Yamato, Toshiba Res & Dev, JAPAN
- Prof. I. Kawakami, Atomic Energy Res. Institute, JAPAN
- Prof. Kyoji Nishikawa, Univ of Hiroshima, JAPAN
- Direc. Dept. Large Tokamak Res. JAERI, JAPAN
- Prof. Satoshi Itoh, Kyushu University, JAPAN
- Research Info Center, Nagoya University, JAPAN
- Prof. S. Tanaka, Kyoto University, JAPAN
- Library, Kyoto University, JAPAN
- Prof. Nobuyuki Inoue, University of Tokyo, JAPAN
- S. Mori, JAERI, JAPAN
- Librarian, Korea Advanced Energy Res. Institute, KOREA
- Prof. D.I. Choi, Adv. Inst Sci & Tech, KOREA
- Prof. B.S. Lilley, University of Waikato, NEW ZEALAND
- Institute of Plasma Physics, PEOPLE'S REPUBLIC OF CHINA
- Librarian, Institute of Phys., PEOPLE'S REPUBLIC OF CHINA
- Library, Tsing Hua University, PEOPLE'S REPUBLIC OF CHINA
- Z. Li, Southwest Inst. Physics, PEOPLE'S REPUBLIC OF CHINA
- Prof. J.A.C. Cabral, Inst Superior Tecnico, PORTUGAL
- Dr. Octavian Petrus, AL I CUZA University, ROMANIA
- Dr. Johan de Villiers, Fusion Studies, AEC, SO AFRICA
- Prof. M.A. Hellberg, University of Natal, SO AFRICA
- C.I.E.M.A.T., Fusion Div. Library, SPAIN
- Dr. Lennart Stenflo, University of UMEA, SWEDEN
- Library, Royal Inst Tech, SWEDEN
- Prof. Hans Wilhelmson, Chalmers Univ Tech, SWEDEN
- Centre Phys des Plasmas, Ecole Polytech Fed, SWITZERLAND
- Bibliotheek, Fom-Inst Voor Plasma-Fysica, THE NETHERLANDS
- Dr. D.D. Ryutov, Siberian Acad Sci, USSR
- Dr. G.A. Eliseev, Kurchatov Institute, USSR
- Dr. V.A. Glukhikh, Inst Electrophysical Apparatus, USSR
- Dr. V.T. Tolok, Inst. Phys. Tech. USSR
- Dr. L.M. Kovrizhnykh, Institute Gen. Physics, USSR
- Nuclear Res. Establishment, Julich Ltd., W. GERMANY
- Bibliothek, Inst. Fur Plasmeforschung, W. GERMANY
- Dr. K. Schindler, Ruhr Universitat Bochum, W. GERMANY
- ASDEX Reading Rm, IPP/Max-Planck-Institut fur Plasmaphysik, W. GERMANY
- Librarian, Max-Planck Institut, W. GERMANY
- Prof. R.K. Jansev, Inst Phys, YUGOSLAVIA