UCRL-51806

MAFJ - SOME SIMPLE ADDITIONS TO MAFCO

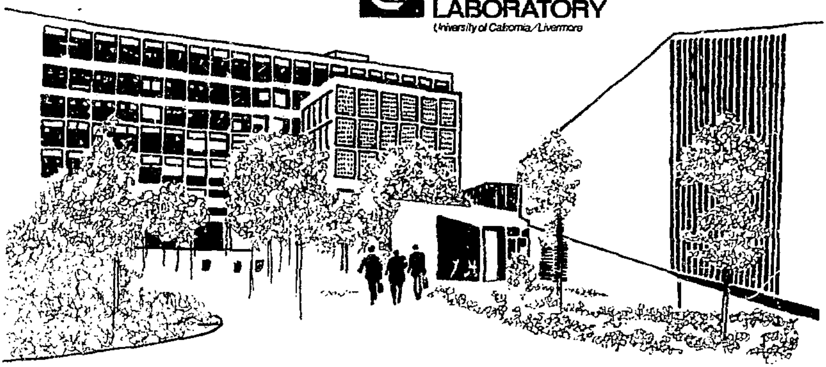
Charles H. Finan, III
Brenda McNamara

January 12, 1976

MASTER

Prepared for U.S. Energy Research & Development Administration under contract No. W-7405-Eng-48

LAWRENCE LIVERMORE LABORATORY
University of California/Livermore



Printed in the United States of America
Available from
National Technical Information Service
U. S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151
Price: Printed Copy $___*; Microfiche $2.25

*Pages	NTIS Selling Price
1-50	$4.00
51-150	$5.45
151-325	$7.60
326-500	$10.60
501-1000	$13.60

Distribution Category, UC-34a

LAWRENCE LIVERMORE LABORATORY
University of California Livermore, California 94550

UCRL-51806

MAFJ - SOME SIMPLE ADDITIONS TO MAFCO

Charles H. Finan, III

Brenda McNamara

MS. date: January 12, 1976

MAFJ - SOME SIMPLE ADDITIONS TO MAFCO

Abstract

MAFJ is a revision of the MAFCO code which generates magnetic fields (B) resulting from collections of current elements. In addition to the original MAFCO's capabilities, MAFJ follows field lines through a region of interest and calculates integrals of functions along these lines. It can also generate three-dimensional grids of B. MAFJ runs from eight to 15 times faster than MAFCO, depending on the types of current elements chosen.

1. Introduction

The version of MAFCO[1] supplied to us by T. Haratani of Lawrence Livermore Laboratory has been updated so that:

1. Coil configurations can be specified in a user subroutine in addition to the regular MAFCO input, eliminating the long card decks otherwise needed (Sec. 2).
2. Integrals of an arbitrary number of functions can be calculated on an array of field lines to study stability and confinement properties of various coil systems[2] (Sec. 3).
3. B can be generated on a three-dimensional grid for input to MCFUSS, a plasma equilibrium and stability code (Sec. 4).

The source code for MAFJ was compiled by CHAT[3] and the object code can be run alone or under the ORDER[4] system on the OCTOPUS or CTR networks at Lawrence Livermore Laboratory. The code runs on the CDC 7600 and is faster than the original MAFCO code by a factor of approximately 15 on calculations involving straight line segments (including general current elements) and by a factor of approximately 8 otherwise. Protection has been added against singularities where a field point falls on a straight line segment. MAFJ will still accept the regular MAFCO input and process it at the accelerated rates without accessing the new options. The user accesses new options by setting flags in the first input card, requesting that new subroutines be called. These subroutines are:

ELCOIL	A user-supplied routine to generate coil current elements.
FLINT	Field line integrator (calls a user-supplied function FOF).
JDRAW	Does contour maps of field line surfaces in x,y plane.

SETXY — Grid generator. Does three dimensional plot of B or sets starting grid for field line calculations.

All arrays in MAFJ have been dimensioned via the PARAMETER* statement, which allows the user to easily increase the size of any needed arrays. Checks have been inserted in the code to insure that the user's input does not exceed the current array sizes. MAFCO's original array sizes are included as comments in the source code (see Appendix).

The following discussion assumes familiarity with the original MAFCO.[1] Sections 2, 3, and 4 describe the three basic changes to MAFCO; Sec. 5 describes how to run the code. The Appendix gives a source listing of MAFJ.

2. Coil Configurations

Upon request, MAFJ will call a user-supplied subroutine named ELCOIL to generate the coil configurations. A sample ELCOIL which generates a set of elliptical coils, all of which lie on a common flux surface, is included in the listing in the Appendix. The generated elements are stored as general current elements and the combined number of these elements and the existing straight line segments and general current elements must be less than or equal to the space available in the general-current-element storage arrays.

3. Field Line Integrals

3.1 GRID SPECIFICATION

The functions to be integrated depend upon the choice of field line, f = f(x,y,B), where (x,y) is the point where the field line intersects the starting plane Z = ZVAL (user-specified) and B is the magnitude of the field. A 20 × 20 array of field lines would involve 400 input cards to MAFCO and considerable work converting cartesian to polar coordinates. To bypass this effort, we have included a separate subroutine, SETXY, which will generate a rectangular grid of QGRID × QGRID points as specified by the user. SETXY will be called by MAFJ at the user's request; the input is described in Sec. 5.

*The PARAMETER statement is described in Ref. 3. Basically, it allows you to make changes throughout the source code by changing only the PARAMETER statement. This would, of course, require recompilation under CHAT.

3.2 METHOD OF INTEGRATION

The most useful line integrals in the study of magnetic fields for plasma containment are $U = \int dl/B$, measuring the average magnetic well properties, $J = \oint dl\sqrt{1 - \nu B}$, the longitudinal invariant for charged particles in the field, and $\Gamma = \oint (Dl/B)P_{||}$, the parallel pressure average defining the pressure balance surfaces. The limits on these integrals may be given Z-surface or, more often, a $|B|$ surface where particles bounce or $P_{||}$ goes to zero. Some coil configurations may produce a very weak well, so the line integral evaluator, FLINT, uses a fourth-order Romberg integrator to display the variations.

MAFCO uses a fixed step length in following each field line; this is good for the Romberg integrator, but the end points for the line integrals will not coincide with a computed point on the line. To maintain the desired accuracy, FLINT interpolates along the line to find the exact end points, fits a quadratic to the integrand near the end point, and calculates an analytic contribution to the integral. The resultant sum (Romberg contribution plus the analytic contribution) is stored in a three-dimensional array, QJIN, which is a function of the (x,y) coordinates of the starting point of the field line and is a function of some arbitrary parameter ν. Hence QJIN = QJIN(x,y,ν).

The FLINT integrator was, of course, tested on a number of analytic functions.

3.3 FUNCTIONS INTEGRATED

The functions to be integrated by FLINT are placed in a user-supplied function, FOF. It receives two arguments, B and M. B is the value of the independent variable and M is the number of the function to be integrated.

Currently, FOF contains two functions to be integrated. These are:

1. The adiabatic invariant J,

$$J \equiv \oint_{b_1(\nu)}^{b_2(\nu)} (1 - \nu B)^{1/2} dl, \tag{1}$$

where b_1 and b_2 are the bounce points $[(1 - \nu B) = 0]$ and ν varies between 0 and B_{min} (found on the center field line).

2. The pressure distribution $\hat{p}_{||}$,

$$\hat{p}_{||} \equiv \int \frac{1}{B}(B_{max} - B)^{5/2} dl, \tag{2}$$

where B_{max} is also found on the center field line.

FLINT receives as input: NNU, the number of ν values to generate and use; ZNORM, the distance (cm) along the Z axis that the calculation is to be followed; INHALF, a switch which tells FLINT that the coil system has symmetry about the Z = 0 plane; and NSYMSW, a switch which tells FLINT that the coil system possesses Baseball-type[5] symmetry. The maximum value of NNU is currently 12 minus the number of functions to be integrated (although this may be changed by varying the PARAMETER IQZ in CLICHE FQ1). The default value is also 12. The default value of ZNORM is 1. INHALF and NSYMSW are set to zero in the code and the input of any nonzero value for them will turn on the specified switch. These switches allow the user to start at the midpoint of his system in the Z plane and calculate out to only one end. The use of INHALF multiplies the resultant values in QJIN by four. The use of NSYMSW allows the user to set a square grid in only one quadrant of the Z = 0 plane and obtain the correct Baseball symmetrization of the integrals. This allows the user to integrate out to only one end. The final input to FLINT is NOFUNCT, an integer (currently set to two) which tells FLINT how many functions are to be integrated.

After all the field lines have been integrated, MAFJ calls the subroutine JDRAW. It makes contour plots of each sheet of the output array, QJIN, in the x-y plane.

4. Three-Dimensional $\underline{B}$ Calculation

The calculation of $\underline{B}$ on the 3-D grid is done in a second version of the subroutine SETXY. This generates the grid, calculates $\underline{B}$, stores the results in families of disk files (BX → BX01, BY → BY01, BZ → BZ01, $|\underline{B}|$ → MODB01), and contours mod $\underline{B}$ as a function of x and y for each z plane (done in subroutine PICTURES). The contour values are chosen on the first Z plane and are then held constant for all remaining planes. This allows the viewing of the mod $\underline{B}$ contours in 3-D. Included in the calculation is a guard grid of zeros around the x,y values of interest. Hence, in a 34 × 34 × 32 system, 32 × 32 × 32 field points are calculated. This guard grid of zeros is used by MCFUSS. Input for this version of SETXY is described in Sec. 5.

5. How to Run MAFJ

5.1 AVAILABILITY

MAFJ is available at Lawrence Livermore Laboratory on the OCTOPUS network from photostore, and may be obtained by the following execute line:

ELF RDSAIL .567025:MLIB:MAFJ / t v

The user will then have the following files:

FMAFJ - Source code (LRLTRAN[3]). Includes all MAFCO subroutines, FLINT (and its associated subroutines), JDRAW, sample ELCOIL, and SETXY.

FSET3D - Source code of SETXY, 3-D version, and its associated subroutines.

ELCOIL - Source code of sample ELCOIL.

BMAFJ, BSET3D, BELCOIL - Compiled (relocatable-binary) versions of above.

MAFJFL - Executable controllee of MAFJ with field-line-integrator version of SETXY.

MAFJ3D - Executable controllee of MAFJ with the 3-D grid of B version of SETXY.

FOF - Source code of the function FOF.

5.2 EXECUTION

The execute line is simple

MAFJFL / t v

for the MAFJ with the field-line-integrator version of SETXY, or

MAFJ3D / t v

for the MAFJ with the three-dimensional-grid-of-B version of SETXY.

The appropriate controllee file must be in the user's private file index, as well as an ASCII file called "INPUT" containing the input to MAFJ.

5.3 INPUT

The input to MAFJ is the same as input to MAFCO (see Ref. 1), except:

1. The first card specifies flags for accessing the new routines available in MAFJ, and
2. Input to the new routines follows the standard MAFCO input. (These routines accept NAMELIST[3] input, i.e., you may specify essentially free-form input assigning values to variable names. For example, PI = 3.1416, MYNAME = "HAROLD", ICOUNT = 1, etc.

5.3.1 First Input Card

Access to the new routines is achieved by setting values on the first input card:

Columns	Program Variable Name	Description
1-5	NOP	Number of problems in this run: integer, right adjusted (same as MAFCO)
6-10	IELIND	Flag to use ELCOIL. If nonzero, MAFJ will CALL ELCOIL with one argument, the value supplied here. In sample ELCOIL, this is the number of coils to be generated; integer, right adjusted.
11-15	ISETIND	To use SETXY, put 1 in col. 15. (Version of SETXY is determined by choice of controllee file.)
16-20	IJIND	To use JDRAW, put 1 in col. 20.
21-25	IFLIND	To use FLINT, 1 in col. 25.
26-30	IGQQ	To output field line plots, put 1 in col. 30. (Not necessary if IFLIND is zero or blank.)
31-35	IQPRISK	To stop printout of MAFCO input, put 1 in col. 35.

5.3.2 Input for SETXY (Integrator Version)

Variable	Type	Use	Default
QGRID	Integer	Grid Size (QGRID = 5, → 5 × 5 grid)	QGRID = 5
QXMAX	Real	Largest x value of grid	If ELCOIL not used, QX(Y)MAX = 1.0
QYMAX	Real	Largest y value of grid	If ELCOIL used, takes value given by ELCOIL
QXMIN	Real	Smallest x value of grid	QXMIN = 0.
QYMIN		Smallest y value of grid	QYMIN = 0.
ZVAL	Real	Initial Z for arr y of field lines	ZVAL = 0.
QSPACE	Real	Δ for field line calculation	QSPACE = .02
QSTOTAL	Real	Total length of field lines (1)	QSTOTAL = 1.5
NSK	Integer	Printout spacing integer (1)	NSK = 1

To obtain default, put F on data card. Total default will yield a 5 × 5 grid from (0., 0., 0.) to (QXMAX, QYMAX, 0.) where QX(Y)MAX come from ELCOIL (if ELCOIL not used QX(Y)MAX = 1.0). See Table 1.

5.3.3 Input to SETXY (3-D Version)

Variable	Type	Use	Default
QXGRID	INTEGER	Specify x grid size (must be even)	QXGRID = 6
QYGRID	INTEGER	Speci.y y grid size	QYGRID = QXGRID
QXGRID			
QZGRID	INTEGER	Specify z grid size	QZGRID = 6
QXMIN	REAL	Minimum x on 3D mesh	QXMIN = 0.
QYMIN	REAL	Minimum y on 3D mesh	QYMIN = 0.
QZMIN	REAL	Minimum z on 3D mesh	QZMIN = 0.
QXMAX	REAL	Maximum x on 3D mesh	QXMAX = 1.
QYMAX	REAL	Maximum y on 3D mesh	QYMAX = 1.
QZMAX	REAL	Maximum z on 3D mesh	QZMAX = 1.
IPR	INTEGER	Stops <u>B</u> printout after 1st z sheet if IPR ≠ 0	IRP = 0 (Full Printout) (Printout of <u>B</u> goes to microfiche)

5.3.4 Input to FLINT

Variable	Type	Definition	Default
NNU	Integer	Number of ν values to generate and use. Maximum etc.	12
ZNORM	Real	Distance (cm) along the Z axis that calculation is to be followed.	1
INHALF	Integer	Switch; a non-zero value tells FLINT that Coil system has symmetry about the z = 0 plane.	0
NSYMSW	Integer	Switch; a non-zero, tells FLINT that the coil system possesses Baseball-type symmetry.	0
NOFUNCT	Integer	Number of functions to be integrated (currently 2).	

5.3.5 Sample Input

Sample input for MAFJFL is shown in Table 1. The first line contains the flags for the new routines. The second line is standard MAFCO input,[1] defining the number of areas, etc. (i.e, it essentially says there are no standard-MAFCO problems to be solved.)

The next four lines are input to the sample ELCOIL routine. The next two lines are the input to SETXY, and the last line is input to FLINT.

Table 1. Sample Input for MAFJFL.

```
   1  25   1   1   1   1
   0   0   0   0   0   0   0   SAMPLE ELLIPTICAL COILS-NO MAFCO INPUT
QEPS=.95 .90 .85 .80 .75 .70 .65 .60 .55 .50 .45 .40 .35 .40 .45 .50
.55 .60 .65 .70 .75 .80 .85 .90 .95 XIL1=.95 .90 .85 .80 .75 .70 .65
.60 .55 .50 .45 .40 .35 .40 .45 .50 .55 .60 .65 .70 .75 .80 .85 .90 .95
QDELTA=13(0.0) 12(90.0) YQZ=.5 ELCOILINPUT
QGRID=10 QXMAX=.1 QYMAX=.25 QXMIN=0. QYMIN=0. QSPACE=.001 QSTOTAL=.65
ZVAL=0.0 SETXYFIELDLINES
ZNORM=1.0 NNU=10 NOFUNCT=2 INHALF=10 FLINTINPUT
```

Sample input to MAFJ3D is given in Table 2. It is the same as that for MAFJFL, except for the last two lines, the input to SETXY.

Table 2. Sample Input for MAFJ3D

```
    1   25    1    1    1    1
    0    0    0    0    0    0    0    SAMPLE ELLIPTICAL COILS-NO MAFCO INPUT
QEPS=.95 .90 .85 .80 .75 .70 .65 .60 .55 .50 .45 .40 .35 .40 .45 .50
.55 .60 .65 .70 .75 .80 .85 .90 .95 XIL1=.95 .90 .85 .80 .75 .70 .65
.60 .55 .50 .45 .40 .35 .40 .45 .50 .55 .60 .65 .70 .75 .80 .85 .90 .95
QDELTA=13(0.0) 12(90.0) YQZ=.5 ELCOILINPUT
QXGRID=10 QYGRID=10 QZGRID=8 QXMAX=.1 QYMAX=.25 QZMAX=1. QXMIN=0.
OYMIN=0.  QZMIN=.5 SETXYCUBEOFBVERSION
```

References

1. W. A. Perkins and J. C. Brown, *MAFCO - A Magnetic Field Code For Handling General Current Elements in Three Dimensions*, Lawreuce Livermore Laboratory, Rept. UCRL-7744-Rev. II (1966).
2. L. S. Hall, B. McNamara, J. K. Boyd, C. H. Finan, D. Fuss, and C. A. Wilgus, *Theoretical Studies of Plasma Confinement in Magnetic Mirrors*, Lawrence Livermore Laboratory, Rept. UCRL-75993, Park II (1974).
3. J. T. Martin, R. G. Zwakenberg, S. V. Solbeck, *LRLTRAN Language Used with the CHAT and STAR Compilers*, Lawrence Livermore Laboratory, Rept. LTSS Chap. 207-Ed. 3 (1973).
4. R. E. Potter, T. E. Fallon, *The ORDER SYSTEM*, Lawrence Livermore Laboratory, Rept. LTSS Chap. 202 (1974).
5. C. Damm, et al., "Baseball Type Magnetic Fields", in *Proc. Third Conference on Plasma Physics and Controlled Nuclear Research*, International Atomic Energy Agency, Vienna (1969).

CB/af/1a