



A USER'S GUIDE TO SYNCH

A. A. Garren and A. S. Kenney
Lawrence Berkeley Laboratory, Berkeley, California 94720

E. D. Courant
Brookhaven National Laboratory, Upton, New York 11973

and

M. J. Syphers
Fermi National Accelerator Laboratory, Batavia, Illinois 60510

June 1985



A USER'S GUIDE TO SYNCH

A.A. Garren A.S. Kenney
 LBL

E.D. Courant
 BNL

M.J. Syphers
 FNAL

June, 1985

A USER'S GUIDE TO SYNCH
Table of Contents

I. INTRODUCTION	4
A. Applications	4
B. SYNCH Language	5
C. History	5
D. Documentation	6
II. COMPUTER ENVIRONMENT	
A. Program Organization	8
B. Program Control -- CDC, VAX	9
C. Input, Output	10
III. CALCULATIONS PERFORMED BY SYNCH	
A. Matrix Definition and Operations	13
B. Betatron Function Calculations	14
C. Particle Beam Calculations	15
D. Closed Orbit Calculations	15
E. Particle Tracking	16
F. Non-linear Transformation Calculations	16
G. Element Misalignment Calculations	19
H. Orbit Correction Calculations	19
I. Straight Section Design	20
J. Fitting Routines	20
List of SYNCH program commands by topic	21

IV. SYNCH STATEMENTS AND COMMANDS	
A. SYNCH Statements	
1) Conventions	23
2) Format	24
B. SYNCH Program Commands	25
A detailed list of all SYNCH program commands in alphabetical order.	
C. Miscellaneous Features	95
V. SYNCH MATHEMATICAL FORMULATION	
A. Primary Element Definitions	100
B. Beamline	111
C. Betatron Functions	112
D. Particle Beam Envelopes	119
E. Closed Orbits	121
F. Particle Tracking	123
G. Misalignments	124
VI. REFERENCES	127
VII. EXAMPLES OF SYNCH RUNS	129
Status of VAX Conversion	

I. INTRODUCTION

INTRODUCTION

SYNCH is a computer program for use in the design and analysis of synchrotrons, storage rings, and beamlines. It has a large repertoire of commands that can be accessed in a flexible way through input statements. Their input and the results of the calculations they invoke are saved in an internal database, so that this information may be shared by other related statements. SYNCH is the first accelerator program to organize its input in the form of a language. The statements, which resemble sentences, provide a natural way of describing lattices and asking the program to give information about them. The organization of the program is modular, so that it has been possible to add new capabilities progressively over the years. In this sense SYNCH may be considered to be an open ended system.

A. Applications

The lattice description is made by statements describing the beamline elements -- drifts, dipoles, quadrupoles, sextupoles, other non-linear elements, etc. -- and beamlines composed of these elements. Beamline statements may refer to the superperiods, cells, transport lines, or substructures of these. Each element-defining statement results in the calculation of the corresponding linear transfer matrix. These matrices can be manipulated in various ways; for example, they can be multiplied together to give matrices corresponding to particular beamlines.

Having generated the lattice of a complete ring, superperiod, cell, or transport line, one may obtain linear properties, especially the betatron functions, of the corresponding beamline; closed orbits and linear properties corresponding to momentum deviations and/or magnet misalignments can be determined; and particles with arbitrary initial conditions can be tracked repetitively around the machine. One may also calculate emittances, damping

times, and other properties of electron rings.

To design machines, the program adjusts the lengths or strengths of certain elements in order to obtain specified properties. For example, one may adjust the gradients of the quadrupoles of a cell to obtain desired phase advances, or adjust several quadrupoles or drifts of an insertion to fit it to the adjacent cells and give a low beta value at its center.

B. SYNCH Language

The input for a SYNCH run consists of a user determined sequence of statements, each containing a label or name, a command (operation, definition), and data. The data may be numerical or the names of previously defined statements. Most statements either define parts of a machine, invoke calculations, or both.

In order to scan systems over ranges of parameters, a set of statements can be placed together in a subroutine-like block that may be called sequentially with different parameter values. These blocks are also used by the fitting commands.

C. History

The first versions of SYNCH were programmed for IBM 704, 7044, and 709 computers by James Eusebio and Garren in 1964-5, primarily for use in the 200 BeV Design Study. These did not contain the general fitting programs, but instead partly analytic routines were included for designing long straight sections of the type used in the Fermilab Main Ring (and later in the Tevatron). In 1966-7 the program was rewritten for the CDC 6600 by Garren and Kenney and later adapted to run on the CDC 7600.

With the aid of Barry Miller of IBM, the CDC 7600 version was converted to run on IBM 360 computers in 1968. New facilities were subsequently added to the CDC version by Garren and Kenney, such as a general fitting routine that uses the CERN program MINUIT, and calculations of electron beam properties. Other capabilities were implemented by Courant, such as polarization effects and

thin-lens multipole calculations. The IBM version was also updated from time to time, with the help of Barbara Wu, Alec King and others at SLAC. In 1979 the current IBM version was implemented to run on a Fujitsu computer by K. Chiba of the INS laboratory of the University of Tokyo. This same version was compiled on a VAX 11-780 by Walter Trecziak of the University of Wisconsin. This version is currently being updated by Kenney to correspond to the present CDC version.

D. Documentation

The first versions of the program were documented in a Berkeley Internal Report by J. Eusebio, A. Garren, and A. Kenney, SYNCH -- A Computer System for Synchrotron Design and Orbit Analysis, UCID-10153, 1965. Since then, this document has been informally updated and supplemented with addenda and notes from time to time.

The present document is based on the above fragmentary documentation, as well as on various extended comments in the program itself. The present document has been written mainly by Syphers, with advice from the other authors.

AAG
MJS

Batavia
April, 1985

II. COMPUTER ENVIRONMENT

COMPUTER ENVIRONMENT

A. Program Organization

SYNCH has been organized in a most flexible way. The user builds the synchrotron structure by a series of modular input statements that correspond to simple verbal statements descriptive of the accelerator. These statements comprise a special-purpose language.

Each SYNCH statement contains three components -- a name, or label; a command; and a set of data. The name is an arbitrary set of characters which can be used as data in subsequent statements to refer to the element defined by the statement. The command is a particular set of characters that characterize the nature of the elements described and of the calculations to be performed. The data completes the specification of the element and provides input parameters for the command. The statement, together with quantities calculated, such as matrices, are stored for use by subsequent statements.

The SYNCH statements are executed sequentially as entered in the input file. Any label or name that is referred to in a SYNCH statement (with the exception of the BML statement; see below) must have been previously defined. If a user defined SYNCH subroutine is to be executed, the subroutine must occur in the input prior to a CALL to it.

A SYNCH subroutine is a sequence of SYNCH statements linked together by being inserted between a SUB statement and an END statement. By including statements in the subroutine to increment data, the effect of a FORTRAN DO loop is achieved.

The beamline (BML) statement has as data a sequence of element names representing a segment of a synchrotron. In this statement, the elements are to be listed in beam order from left to right, i.e., in the order in which the beam

traverses the elements. Note that beam order is opposite to the order in which the corresponding matrices are usually written for hand computation. The beamline statements merely create "lists" of elements; they are not "executable" statements. Hence, a beamline may be defined prior to the statements defining the beamline's individual elements.

B. Program Control

1) Using the CDC Version at FNAL

The CDC version of program SYNCH is run as a batch job and may be executed by typing the following Job Control Language commands into the input file to be submitted to SYNCH:

```
--- any other JCL commands or information ---
```

```
ATTACH,SYNCH/UN=LIBR  
SYNCH  
/EOS
```

```
--- SYNCH input statements go here ---
```

```
/EOP
```

If the above information exists in an input file named SYNCHDAT, for example, then the following command will submit the job to the CDC:

```
SUBMIT,SYNCHDAT
```

Here, the output from the program will be directed to the CDC line printer. Naturally, by defining one's own output files and by using proper options for the submit command, the output of the program could be routed to a disk file which could then be examined from a user terminal and/or routed to a line printer.

2) Using the VAX Version at FNAL

The VAX version of SYNCH may be run interactively by creating the following DCL command file, called SYNCH.COM in this example:

```
$ assign SYNCH.DAT for002
$ assign SYNCH.OUT for003
$ run synch
```

The compiled executable file SYNCH.EXE must exist in the user's area. Copies of SYNCH.EXE may be obtained from other users of the program via DECnet.

The file SYNCH.DAT contains only the input statements beginning with a RUN statement and ending with a STOP statement. The output from the run is sent to SYNCH.OUT.

The program is then run by typing @SYNCH <return> while in DCL.

C. Input, Output

1) The Input File:

SYNCH input is of 80-column format. The data is read from the file as "cards", one line at a time. All input data must be located in proper fields, properly justified. The rules governing the format of the input statements are described in Section IV, SYNCH STATEMENTS AND COMMANDS. No special characters (such as TAB's) are supported, and extraneous blank lines cannot be dealt with at the present time.

Any valid SYNCH statements may be included in the input file. The first statement found in the file must be a RUN statement. The end of a SYNCH program is designated by a FIN statement. Any number of SYNCH programs (RUN --> FIN) may be placed sequentially in a single input file. After the last program of the input file, a STOP statement must be present.

2) The Output File:

The output generated by SYNCH includes the input statements found in the input file, re-formatted, and any output produced by the various SYNCH statements. The output generated by a particular SYNCH statement is sent to the output file, or output device, immediately upon execution of the SYNCH statement. The echoing of the input statements may be turned off at any point during the SYNCH program by inserting a NCPY statement. To re-enable echoing of the input statements, the COPY statement is used.

III. CALCULATIONS PERFORMED BY SYNCH

CALCULATIONS PERFORMED BY SYNCH

This section describes what calculations may be performed by SYNCH and more importantly which SYNCH commands are necessary in order to perform certain types of calculations. Detailed descriptions of individual SYNCH statements are left to Section IV, SYNCH STATEMENTS AND COMMANDS. Details concerning the mathematical methods employed by the program, such as descriptions of beamline element matrices, may be found in Section V, SYNCH MATHEMATICAL FORMULATION.

A. Matrix Definition and Operations

SYNCH allows the definition of many standard beamline elements which act on the particle beam. Linear transfer matrices are calculated for these elements upon execution of their defining statements. The **MAG** statement defines a magnet which could be strictly bending, strictly focusing, or a combination of both. This magnet is thought of as bending in the horizontal plane. A vertically bending magnet may be defined through the use of the **MAGV** statement. The **DRF** statement is used to define a drift region. Other elements include the sextupole magnet (**SXTP**), the n-pole thin lens (**NPOL**), and the delta-function dipole kick using the **KICK** statement. The **MAG**, **MAGV**, **DRF**, and **KICK** statements generate matrices which act on a particle state vector containing a particle's position and slope in each plane and the particle's momentum error, dp/p . The effects of the **SXTP** and **NPOL** statements on these vectors are not linear and are handled by separate subroutines in SYNCH.

The user may also define linear or non-linear elements by use of the **CRD**, **MAT**, and **MAP** statements.

Once defined, various operations may be performed upon the matrices of the linear elements. In particular, **MMM** multiplies two or more matrices and stores the result in a newly defined matrix; **INV** determines the inverse of a matrix; **ROT** performs a rotational transformation upon a given matrix (The **ROTZ** command, on the other hand, generates a rotation matrix.); **EQU** equates two matrices; **REF** generates the matrix of the reflection of a beamline, and the ****** command allows the repeated multiplication of a matrix with itself. At any time, the user may print out the elements of a matrix by invoking the **WMA** command.

A particular sequence of beamline elements, which may contain linear and non-linear elements, may be created using the **BML** statement. This statement may be used recursively, i.e., a **BML** statement may contain as part of its defining sequence the names of other **BML** statements.

It should be pointed out that when defining various beamline elements, the units used are quite arbitrary, but should be consistent throughout the entire **SYNCH** input. For instance, one could enter all lengths in meters, all field strengths in kilogauss, kilogauss/meter, and so forth. In this case, the betatron functions that are calculated by **SYNCH** will be in units of meters, etc.

B. Betatron Function Calculations

One of the primary uses of the program **SYNCH** is to calculate the linear betatron functions which describe the amplitude and phase of the transverse betatron oscillations throughout an accelerator and/or beamline. The formal language of betatron motion used throughout **SYNCH** and the rest of this manual is the language of Courant and Snyder (Reference 3). The reader is referred also to Section V-C, Betatron Functions, for further information.

The values of the various betatron functions corresponding to a particular matrix may be obtained using the **BETA** statement. These values may be printed out by use of the **WBE** command. Given an initial set of betatron functions, which may be input directly by the user or obtained from a previously defined matrix, the betatron functions may be tracked through a beamline by using the **TRKB** command.

Perhaps the most commonly used SYNCH betatron function calculation is that of the CYC statement. CYC generates the betatron functions at each point of a periodic section of a circular accelerator, printing out the values of the functions in the process. A similar command, CYA, also saves the cyclic matrices used in the calculation for use later in the SYNCH run.

C. Particle Beam Calculations

Using the betatron functions generated as in the CYC statement, SYNCH may be used to compute beam envelopes with the CYAE command. The user must input beam emittances and the momentum offset of the beam, dp/p. The beam emittances are input with the BVAL statement.

For electron machines, emittance calculations may be performed using the CYEM command. Here, electron integrals, rf output, and electron emittance factors are computed and printed out.

D. Closed Orbit Calculations

Another important feature of SYNCH is the ability to compute the closed orbit through a circular accelerator for a particle of any momentum. The accelerator may consist of linear and/or non-linear elements. The user must input an initial guess for the closed orbit at the point in question using a PVEC statement. Then, with the FXPT statement, the closed orbit at that point is found and propagated through the entire accelerator. The transfer matrices of the beamline elements are then linearized about this closed orbit and the betatron functions from these new matrices are printed out along with the closed orbit.

The FXPT calculation uses 7x7 matrices and hence allows for the study of horizontal-vertical coupling of the betatron motion. The single-turn transfer matrix for the point in question is thus printed out as well as the eigenvalues and eigenvectors of the 4x4 submatrix representing x-y motion. The eigenvectors may be tracked through the accelerator.

E. Particle Tracking

In a rather straightforward manner, a particle whose initial state is specified, may be tracked repeatedly through a previously defined beamline using the TRK statement. The initial trajectory and dp/p of the particle is input with the PVEC statement. The trajectory of the particle may be output at all points in the beamline each transit or at specified beamline locations every so many transits. The beamline may consist of linear and/or non-linear elements.

F. Non-linear Transformation Calculations

As was stated previously, the user may define non-linear transformations to be used in the SYNCH program via the MAP statement. The SYNCH code has 10 dummy subroutines labeled MAP, MAP1, MAP2, ..., MAP9. Any or all of these subroutines may be changed by the user and SYNCH recompiled.

To access one of the MAP subroutines, the user simply defines a beamline element with the MAP statement as dictated in Section IV. Then, whenever SYNCH encounters one of these beamline elements, the current particle state vector and any other parameters specified by the user in the MAP statement will be passed to the respective MAP subroutine and the subroutine will be executed.

Example: Writing a Non-linear Subroutine Named MAPk

The MAP subroutine names and arguments are fixed and are

MAPk(V,PAR)

where

k is blank or an integer, $0 < k < 10$

V is a subscripted variable of dimension 7 which contains elements of a particle state vector $(x, x', y, y', -ds, dp/p, 1)$ on which transformations will be made. At the end of the routine, the transformed values will replace the original ones in the array V

PAR is an array containing parameters needed for the transformation. The dimension depends on the transformation. The input variable m in the MAP statement will correspond to the dimension.

As an example, consider the following non-linear transformation:

$$X = x$$

$$X' = x' - (3Ax^2 + 2Bxy + Cy^2)$$

$$Y = y$$

$$Y' = y' - (Bx^2 + 2Cxy + 3Dy^2)$$

$$-dS = -ds$$

$$dP/P = dp/p$$

$$1 = 1$$

where A, B, C, and D are four parameters associated with the transformation. We arbitrarily take k=6 and give the MAP statement the name XMPL.

A MAP subroutine accomplishing the above transformation follows:

```
      SUBROUTINE MAP6(V,PAR)
      DIMENSION V(7),W(7),PAR(4)
C      Compute the transformed values...
      W(1) = V(1)
      W(2) = V(2) - (3.*PAR(1)*V(1)*V(1) + 2.*PAR(2)*V(1)*V(3)
1      + PAR(3)*V(3)*V(3) )
      W(3) = V(3)
      W(4) = V(4) - (PAR(2)*V(1)*V(1) + 2.*PAR(3)*V(1)*V(3)
1      + 3.*PAR(4)*V(3)*V(3) )
      W(5) = V(5)
      W(6) = V(6)
      W(7) = V(7)
C      Put transformed values in V array...
      DO 2 L=1,7
2      V(L) = W(L)
      RETURN
      END
```

Having written this routine, the user may name the MAP statement and input values of A, B, C, and D by including the following line in the input file

```
      XMPL MAP6 4 A B C D
```

and substituting the appropriate values for A, B, C, and D.

The parameters associated with a MAP statement may be altered by a REPL or an INCR statement.

G. Element Misalignment Calculations

The effects of magnet misalignments on the closed orbit of the accelerator may be studied using SYNCH. There are two possible approaches to the problem. One approach involves the **MAGS** command, which defines a magnet with transverse misalignments. The **BMIS** command (Begin MISalignment mode) must first be issued and the **EMIS** command will end the misalignment mode. In this mode, the new closed orbit due to the misalignments is calculated using a **CYC** statement.

The second approach requires the use of the **MOVE** statement. Here, rotational misalignments about the s-axis as well as transverse misalignments may be invoked. This method uses the **FXPT** statement to find the new closed orbit and may be used to study coupling effects of the transverse motion in each plane.

For more information concerning these methods, see Section V-G.

As aids for misalignment calculations, the **LIST** statement and the **RAND** statement were created. The **LIST** statement allows the user to generate a list of elements which can be used to substitute successive calls for a particular transfer matrix. The **RAND** statement generates a random number uniformly distributed on the interval $[-1/2, 1/2)$.

H. Orbit Correction Calculations

Using the **ORBC** statement, the correction of the closed orbit through an accelerator with field errors may be calculated. The name of a **FXPT** statement is used to define the initial orbit, while the name of a **CYC** statement provides necessary betatron functions. The common beamline used in the **CYC** and **FXPT** statements contains the name of the elements at which the displacements are measured and the name of the elements which are used for corrections. The optimized correction element strengths are calculated and printed out.

I. Straight Section Design

Four SYNCH statements allow the user to design matching straight section insertions to be incorporated in a previously defined beamline. The first command, STRN, designs a normal matching straight section of a specified phase advance. The second command, STRP, designs a straight section of the form FD-0-FD or DF-0-DF which has a total phase advance of π radians. The third command, STR2, designs a Collins straight section of desired phase advance. STR4, the fourth command, designs a four-element antisymmetric straight section.

In each command, the betatron functions at the ends of the insertion are either specified explicitly by the user, or obtained from a specified matrix representative of a certain point in the beamline.

J. Fitting Routines

Several fitting routines exist in the program SYNCH allowing one to generate beamline elements which create desired values of betatron functions. For instance, FITQ varies the parameters of specified magnets to create specific horizontal and vertical phase advances for a particular portion of a beamline. Likewise, FITB may be used to fit other betatron functions to desired values. The SOLV command is more general in that it can vary any number of parameters subject to constraints in order to generate desired values of several betatron functions at once.

Each of the above mentioned fitting routines makes use of the routine MINUIT (see Reference 5). Direct use of all valid MINUIT commands may be made by using the SMIN statement in the SYNCH input.

A complete list of all SYNCH commands organized by topic is given on the following page.

SYNCH Program Commands by Topic

(p) = generates output

Program Control	RUN, FIN, STOP, DELE, ACT, SIZE
SYNCH Subroutines	SUB, END, CALL, REPL, INCR, MESH(p)
Mathematical Operations	=, SUM, CALC (COS,SIN,SQRT,1/X,etc.), RAND
Beamline, Element Definitions	DRF, MAG, MAGV, ROTZ, KICK, SXTF, MAP, NPOL, BML
General Matrix/Vector Definitions	CRD, MAT, VEC
Matrix Operations	MMM, EQU, INV, REF, **, ROT, INV2, MXV, WMA(p)
Betatron Function Calculations	BETA, WBE(p), CYC(p), CYA(p), TRKB(p)
Particle Beam Calculations	CYAE(p), CYEM(p), BVAL
Fitting Routines	FITB(p), FITQ(p), SOLV(p), SMIN(p)
Closed Orbit Determination and Particle Tracking	PVEC, FXPT(p), TRK(p)
Element Misalignments	BMIS, EMIS, MAGS, SHF7, MOVE, LIST
Orbit Correction	ORBC
Straight Section Design	STRN(p), STRP(p), STR2(p), STR4(p)
Output Statements (p)	C, P, TAB, PTAB, PRNT, COPY, NCPY

IV. SYNCH STATEMENTS AND COMMANDS

A. SYNCH STATEMENTS

Conventions

A SYNCH program is made up of a set of SYNCH statements beginning with a RUN statement and ending with a FIN statement. The statements in between are composed of various parts: a label (or name), a command, and assorted input parameters which depend upon the nature of the statement. The purpose of this section is to describe the statement associated with each SYNCH command. The descriptions will include a Statement Diagram and a short paragraph about the command and its uses. Some examples are included in the descriptions. However, for examples of the more complex SYNCH commands, the user should refer to Section VII. The list of commands found in this section is arranged in alphabetical order. For a cross-reference of commands organized by categories, refer to Section III.

In the pages that follow, each description of a SYNCH command is preceded by a SYNCH Statement Diagram. The conventions used in these diagrams to describe various types of elements, variables, and commands are as follows:

SYNCH Statement Diagram Conventions

1) Each part of a SYNCH statement will be left or right adjusted in its respective field.

2) All commands, and all labels other than those of "=" statements will be written in capital letters.

3) Labels and data corresponding to symbolic floating point numbers (see next page) will be written in lower case and followed by a period.

4) Integer variables will be written in lower case and begin with the letters i, j, k, l, m, or n.

(Note that SYNCH distinguishes between upper and lower case characters.)

General SYNCH Statement Format

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
 LBL | |CMD | mm| nn|----      ---      ---      ---      data      ---      ---      ---      ---
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

| -- field not used

COLUMNS

- 1 -- used for the following commands or options: C, ., P, and - .
- 2-5 LBL -- label or name (LEFT ADJUSTED) of the element or variable generated by the SYNCH command; this name may be used as data for future commands involving that element or variable. A few symbols do have special meanings (such as *, /, -, #, and +) and should be used with caution.
- 8-11 CMD -- the name of the particular SYNCH command; LEFT ADJUSTED.
- 13-15, mm,nn -- integers, up to 3 digits, used in conjunction with some of
17-19 the commands; must be RIGHT ADJUSTED.
- 21-80 data -- input data for the particular command; the data is floating point, alphanumeric, or integer depending upon the command:
- Floating Point -- numerical floating point numbers, entered in 10-column fields 21-30, 31-40, etc.
- Symbolic Floating Point -- label of a floating point parameter previously defined by an "=" command; used in the same 10-column data fields (LEFT ADJUSTED) which normally would contain numerical input.
- Alphanumeric -- names or labels (LEFT ADJUSTED) in 5-column fields 21-25, 26-30, 31-35, etc.
- Integer -- these are RIGHT ADJUSTED in 5-column fields 21-25, 26-30, 31-35, etc.

B. SYNCH PROGRAM COMMANDS

ACT -- Activate Execution of Certain SYNCH Statement(s)

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
      ACT      m      n A1      A2      A3      ...      Am
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

m = number of elements to be activated
n = (see below)
A1,A2,...,Am = names of elements to be activated

Activates statements which have been deactivated using DELE statement; if n=0, then the data field contains m elements which are to be activated; if n=1, then the data field contains only one name (A1, say) which is the name of the first of m consecutive statements to be activated, beginning with A1.

See also, DELE

See also, use of "-" in column 1, Section IV-C, Miscellaneous Features.

COMMANDS

BETA -- Betatron Function Definition

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
BE      BETA      m      M
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

BE is made equal to the m-th betatron function of the matrix M, where m may be one of the values given in the table below:

m = 1	mux	m = 11	muy
2	betax	12	betay
3	alphax	13	alphay
4	gammax	14	gammay
5	x-dispersion	15	y-dispersion
6	slope of x-dispersion	16	slope of y-dispersion

The variable BE may be referred to in future SYNCH statements.

COMMANDS

BMIS -- Begin Misalignment Mode

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
      BMIS
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

 Begins the magnet misalignment mode; magnets defined by a MAG statement after a BMIS statement and before an EMIS statement will have transfer matrix elements $M13 = M23 = 0$.

 See also: EMIS,

 MAGS,

 Section V-G, Misalignments.

COMMANDS

BML -- Beam Line Definition

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAME  BML      m      A1   A2   A3   A4   ...  Ak
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

NAME = arbitrary name
 m = reflection option (see below)
 A1,A2,...,Ak = names of elements

Defines a beamline made up of primary elements or other beamlines A1,A2,...,Ak entered in order as "encountered by the beam". BML statements may occur in the input file before the elements making up the beamline are individually defined -- the BML command merely sets up a list of elements, it does not perform any calculations.

Blank fields are ignored. The beamline definition may be continued on successive lines.

N repetitions of a set of elements may be obtained by placing bbN(b on the left of the set and b)bbb on the right, where b = blank space (i.e., parentheses may only occur in "4" or "9" columns).

Reflection is obtained by setting m = -1, in which case all of the elements will occur in reverse order. Note, however, that SYNCH will not reflect any of the constituent elements, so all of these should be self-reflecting -- magnets, drifts, or symmetric MMM's. As an example, in the following code...

```

.OBB  BML           0   B   B
OBB  MMM           .OBB
BBO  REF           OBB
*C   BML           OBB  QD  OBB  QF
C*   BML      -1   *C

```

COMMANDS

the last line produces the beamline C* which is equivalent to QF OBB QD OBB in beam order, whereas the correct reversed beamline is QF BBO QD BBO.

Another Example -- The following recursive set of BML's ...

```

CELL  BML      QD      2( B1      ) QF      2( B1      )
STSS  BML      0      QDL  LS   QFL  0
SPRD  BML      STSS   5( CELL      )

```

is equivalent to...

```

SPRD  BML      0      QDL  LS   QFL  0      QD  B1  B1  QF  B1  B1  QD
        B1  B1  QF  B1  B1  QD  B1  B1  QF  B1  B1  QD
        B1  B1  QF  B1  B1  QD  B1  B1  QF  B1  B1

```

where QD, B1, 0, etc. are previously defined MAG's, BML's, MMM's, DRF's, etc.

Thus, if a CYC statement is performed on SPRD, no matter which method was used to define it, output will be generated for each of the 35 elements listed above.

COMMANDS

BVAL -- Particle Beam Definition

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAM    BVAL    m    n pt.        aw.        epsx.        epsy.        epsl.        sigl.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

```

NAM      = arbitrary name
m        = kinematics calculation option (see below)
n        = print option
pt       = momentum or kinetic energy of particles (see below)
aw       = atomic weight (multiples of proton mass; see below)
epsx     = horizontal emittance
epsy     = vertical emittance
epsl     = longitudinal emittance
sigl     = bunch length

```

The emittances of a particle beam is entered and saved for use with CYAE and CYEM statements. If the kinematics option is selected (pt not 0), then the input emittances are taken to be normalized emittances. If m = 0, then pt represents the particle momentum (GeV/c). If m is not zero (or blank), then pt represents the particle kinetic energy (GeV). If pt = 0, then the values input for the emittances are taken to be unnormalized and no kinematics calculations are performed. The atomic weight is in units of the proton rest mass. However, if aw = 0 (or blank), the electron mass is assumed.

For use in the CYAE statement, the value of dp/p needs to be determined. In this case, the user should use sigl = 1, and then dp/p will be given by the value for epsl. (epsl = sigp x sigl)

The final values for the vertical, horizontal, and longitudinal emittances and sigp (or dp/p) are saved for future use in the CYEM (CYAE) statement.

COMMANDS

C -- Comment

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
C      ----                --- text ---
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

Causes any text entered in columns 2 through 80 to be printed.

COMMANDS

CALC -- Calculator Simulator

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAM  CALC  m      cmd1 cmd2 cmd3 ...  cmdk
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

Simulates a Reverse Polish calculator; commands are entered as 4-character alphanumeric data; printing is controlled by m:

```

m = 0  -- no print
m = 1  -- print stack at end
m = 2  -- print stack every step

```

Allowable commands are:

+	-	*	/	1/X	SQRT	X*X	Y**X
EXP	LN	SIN	COS	TAN	ASIN	ACOS	ATAN
ABS	EEX	CHS	X-Y	X=Y	P-R	R-P	X-0
PI	RDN	RUP	ENTR				
STO	RCL	CLX	LSTX				
XLEY	XGTY						

The final result of the calculation is stored in NAM. The stack is preserved -- it may be picked up by later CALC statements.

When issuing a STO or RCL command to and from memory, one must issue the name of a SYNCH "=" statement to be used as the storage buffer.

Example:

```

lhct = 22.
lb = 20.
ldr1 = 0.5
ldr2 = 0.0
      CALC      RCL lhct RCL lb - RCL ldr1 - STO ldr2

```

COMMANDS

In this example, the lengths lb and ldrl are subtracted from lhct and the result is stored as ldr2.

CALL -- Call for Execution of a SYNCH Subroutine

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
      CALL      m      SBR
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

The SYNCH subroutine SBR is executed m times; if m is omitted, m is taken to be 1.

COPY -- Print Out SYNCH Input Statements

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
      COPY
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

Re-enables echoing of SYNCH input statements as they are read in from the input file. The command NCPY is used to suppress this echoing. "COPY" is the default mode.

COMMANDS

CRD -- Define Transfer Matrix

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
R      CRD          Rx11.   Rx12.   Rx13.   Rx21.   Rx22.   Rx23.
          Ry11.   Ry12.   Ry13.   Ry21.   Ry22.   Ry23.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

Forms the matrix R representing the horizontal (Rx) and vertical (Ry) transfer matrices with the elements given above; the third row is taken to be (0, 0, 1).

COMMANDS

CYA -- Cycle Matrix Product and Save Cycled Matrices

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAME  CYA          n  A
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

NAME = arbitrary name
n = number of superperiods in the synchrotron
A = name of previously defined BML consisting of the
 k elements B1,B2,...,Bk

n is optional

Same as CYC (see below) with some exceptions. Cycled matrices are computed for the beamline BML and betatron functions computed. The k new matrices and lattice functions are saved for future use. If N is the first character of NAME, then the names of the cycled matrices are of the form Nj where j is an integer; as an example, if the following SYNCH statements were used,

```

      L      BML      A      B      C      D
TST  CYA          L

```

then four cycled matrices bearing the names T1, T2, T3, and T4 would be generated and would be equal to ABCD, BCDA, CDAB, and DABC respectively; the matrix TST could then be referred to in a later SYNCH statement as well, TST being equivalent to T1. (The authors recommend using a single character to name a CYA statement.)

The chromaticity calculation found in CYC is not supported in CYA.

COMMANDS

CYAE -- Beam Envelope Calculation

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAM    CYAE    m      BL    BEAM efact.    epxco.    epyco.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

```

NAME   = arbitrary name
m      = (see below)
BL     = name of previously defined beamline (BML statement)
BEAM   = name of BVAL or CYEM statement giving emittances, dp/p
efact  = eps/eps0
epxco  = closed orbit equivalent emittance (horizontal)
epyc0  = closed orbit equivalent emittance (vertical)

```

Using betatron functions, beam envelopes along the beamline BL are calculated from the previously determined emittances ϵ_{sx} , ϵ_{sy} and dp/p ; $efact$ is the ratio of the emittance to be used by the CYAE command to the emittance obtained from the BVAL statement; the displacements due to the betatron motion and due to the momentum deviation are thus multiplied by \sqrt{efact} ; the betatron and momentum displacements may be added together algebraically ($m=1$) or in quadrature ($m=0$).

COMMANDS

CYC -- Cycle Matrix Product

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAME  CYC          n A          SX1  SX2  cx.          cy.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

NAME = arbitrary name
n = number of superperiods in the synchrotron
A = name of previously defined BML
SX1,SX2 = name of previously defined sextupoles
cx,cy = desired horizontal and vertical chromaticities

n is optional

This command creates k new matrices representing the cyclic products of B_1, B_2, \dots, B_k ; betatron functions (beta, alpha, cumulative phase advance/ 2π (*)) and the cumulative path lengths s are computed at the end of each element and printed out, as well as the tune of the entire synchrotron (n superperiods), the machine radius, the total bend angle of the machine, and the transition gamma; the horizontal and vertical chromaticities of the machine due to magnets which are explicitly used in the BML statement are calculated and printed (i.e., if none of B_1, B_2, \dots, B_k are MAG's, then no chromaticities will be computed).

This command also provides an option for adjusting the chromaticity of the accelerator by varying the strengths of two families of sextupoles SX1 and SX2. The beamline A may contain more than two families of sextupoles, but only the two specified will be used in the computation. The printout of the CYC command will list the chromaticities before the recomputations of parameters of SX1 and SX2. To verify the computation, another statement involving the beamline A (or,

*) -- If any of the matrices B_1, \dots, B_k represent elements with phase advance greater than 2π , the integer part of the cumulative phase advance/ 2π will not be correct.

COMMANDS

at least the sextupoles which were varied) must be run. (See SXTP for discussion of sextupole element definition.)

CYEM -- Electron Beam Emittance Calculation

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
  NAM   CYEM   m   n BL           energy.   radius.   freq.     volt.     kappa.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

```

  NAM       = arbitrary name
  m         = output option (see below)
  n         = number of superperiods (see below)
  BL        = name of beamline (BML) or FXPT statement
  energy    = electron beam energy (GeV)
  radius    = machine radius (meters)
  freq      = rf frequency (MHz)
  volt      = rf voltage (MV/turn)
  kappa     = coupling coefficient [sqrt(epy/epx0)]

```

Calculates electron integrals, rf output, and electron emittance factors.

If BL is the name of a FXPT statement, then the beamline and n are taken to be those of the FXPT statement. This procedure is used for the study of the effect of perturbations due to misalignments. For this case, output options include:

```

  m = 0,1   --   print only at start of beam line
  m = 2,3,4 --   print at every element
  m = even  --   betatron part of sigma matrix
  m = odd   --   total sigma matrix
  m = 2,3   --   sigma matrix long output
  m = 4     --   Aij, Bij long output

  m positive prints Rii, Tij ,

```

COMMANDS

where $R_{ii} = \sqrt{S_{ii}}$
 $T_{ij} = \arctan(S_{ij}/S_{ii} - S_{jj})/2$
 $S_{ij} = \sigma(i,j)$

m negative prints $S_{ij} = \sigma(i,j)$ for all elements
 κ is the coupling coefficient $\sqrt{\epsilon_{py}/\epsilon_{px0}}$; if $\kappa=0$ and BL is a FXPT statement, then κ is set to $1.0E-08$.

DELE -- Delete Execution of Certain SYNCH Statements

```
-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
      DELE      m      n A1      A2      A3      ...      Am
-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

m = number of elements to be deleted
 n = (see below)
 A_1, A_2, \dots, A_m = names of elements to be deleted

Deactivates execution of specified SYNCH statements; if $n=0$, then the data field contains m elements which are to be deactivated; if $n=1$, then the data field contains only one name (A_1 , say) which is the name of the first of m consecutive statements to be deactivated, beginning with A_1 ; input data is stored when a statement is deactivated; an entire subroutine CANNOT be deactivated; a CALL to a subroutine can be deactivated; RUN and FIN should never be inoperative;

See also, ACT

See also, use of "-" in column 1, Section IV-C, Miscellaneous Features.

COMMANDS

DRF -- Drift Space Definition

METHOD 1:

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAME DRF          length.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

NAME = arbitrary name
length = length of the drift region

Defines a drift region of length "length" designated by NAME.

METHOD 2:

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
DNAM DRF   m   ldnam.   ltot.   D1   D2   D3   ...   Dm
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

DNAM = arbitrary name
ldnam = adjustable length of the drift region DNAM
ltot = total length of the drifts DNAM, D1,D2,...,Dm

This use of the DRF statement is used to adjust the length of the drift DNAM to maintain the sum of a set of drifts and/or magnets at constant length. If ltot IS NOT equal to 0.0, ldnam is adjusted to make

$$ldnam + 1D1 + 1D2 + 1D3 + \dots + 1Dm = ltot$$

If ltot IS equal to 0.0, ltot is stored as

$$ltot = ldnam + 1D1 + 1D2 + 1D3 + \dots + 1Dm$$

If DRF is used in this fashion within a SYNCH subroutine, then on subsequent calls lnam will be adjusted to maintain constant ltot.

COMMANDS

EMIS -- End Misalignment Mode

```
-----1-----2-----3-----4-----5-----6-----7-----8
      EMIS
-----1-----2-----3-----4-----5-----6-----7-----8
```

Turns off the magnet misalignment mode.

See also, BMIS

See also, MAGS

END -- Completion of Definition of SYNCH Subroutine

```
-----1-----2-----3-----4-----5-----6-----7-----8
      END
-----1-----2-----3-----4-----5-----6-----7-----8
```

The statements between a SUB statement and an END statement constitute a SYNCH subroutine.

COMMANDS

EQU -- Equate Matrices

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
P      EQU      Q
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

The matrix P is equated to the matrix Q

FIN -- End of Run

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
      FIN
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

Signals the end of a run; a further set of data, beginning with RUN, may follow this command.

COMMANDS

FITB -- Vary Two Parameters to Fit Two Betatron Functions to Specified Values

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAME FITB   m   n SBR  MTR  V1   V2       il   i2des1.    des2.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

m,n = specify which betatron functions to fit (see table below)
 SBR = SYNCH subroutine containing MTR
 MTR = matrix from which to extract betatron functions
 V1,V2 = elements containing parameters to be varied
 il,i2 = parameters of V1 and V2 which are to be varied
 des1,des2 = desired values of the betatron functions to which to fit

FITB iterates through subroutine SBR, which contains matrix MTR (an MMM statement) to obtain desired values des1,des2 of betatron functions m, n, respectively; the betatron functions to be fitted are specified by:

m or n	function	m or n	function
2	betax	12	betay
3	alphax	13	alphay
5	dispersionx	15	dispersiony
6	slope of dispx	16	slope of dispy
7	sqrt(betax)	17	spqrt(betay)

il,i2 refer to the i-th parameters of the SYNCH statements named V1 and V2; see example, next page.

COMMANDS

FITB Example: Fit beta functions at ends of a standard FODO cell...

```

SBR  SUB
gd   =          -90.
gf   =          90.
QD   MAG          2.      gd   5000.
QF   MAG          2.      gf   5000.
B    MAG          28.     0.   5000.      25.      $
CELL MMM          B      QD   B          QF
END
C
betx =          99.
bety =          30.
      FITB  2  12 SBR  CELL gf  gd      1      lbetx  bety

```

The above code would iterate through subroutine SBR, varying gd and gf until the values of the horizontal and vertical beta functions associated with matrix CELL were 99.0 and 30.0 respectively. One could also use the following FITB statement to perform the same task...

```

      FITB  2  12 SBR  CELL QF  QD      2      2betx  bety

```

This method would only vary the matrix elements of QF and QD and the values of the variables gf and gd would not be changed.

COMMANDS

FITQ -- Vary Two Parameters to Fit the Betatron Phases to Desired Values

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
      FITQ           SBR  MTR  V1   V2       i1   i2des1.     des2.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

```

SBR      = SYNCH subroutine containing MTR
MTR      = matrix from which to extract phase advances (MMM)
V1,V2    = elements containing parameters to be varied
i1,i2    = parameters of V1 and V2 to be varied
des1,des2 = desired values of tunes (phase advances/2pi)

```

Similar to FITB, FITQ iterates through subroutine SBR, which contains matrix MTR, to obtain desired values des1,des2 of the phase advances/2pi through the elements composing MTR; if MTR is defined through a CYA or CYC statement, the values des1,des2 represent the tunes through n superperiods (see CYA, CYC); note that only the fractional parts of the tunes can be derived from MTR, thus FITQ will fit to the nearest tunes with the fractional parts specified.

Example:

```

      SBR  SUB
      gd   =          -90
      gf   =          90
      QD   MAG      2.      gd      5000.
      QF   MAG      2.      gf      5000.
      CELL MMM      .C
      END
C
      nux  =          .4
      nuy  =          .4
      FITQ SBR  CELL gd   gf      1   lnux   nuy

```

COMMANDS

FXPT -- Closed Orbit Calculation

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAM  FXPT   m   n P   BLIN  nscl isav  itr iflg itap
      e1.    e2.    dl.    d2.    d3.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

```

NAM      = arbitrary name
m        = output option (see below)
n        = number of superperiods
P        = initial guess of closed orbit (name of PVEC statement)
BLIN     = beamline
nscl     = number of superperiods to close the orbit
isav     = save option (see below)
itr      = number of iterations to find closed orbit (default = 25)
iflg     = ray trace option (see below)
itap     = 0 (do not write to tape) or 1 (do write to tape)
e1,e2    = convergence criteria for displacements,slopes
dl,d2    = displacements from ref. ray of neighbor rays for finding M4
d3       = displacement in dp/p for calculating linearized 3x3 matrices

```

Given a beamline BLIN which may contain non-linear elements, the FXPT command calculates and prints the positions through BLIN of the closed orbit for particles with momentum error dp/p . The beamline may be made up of any linear transformations, SXTP, NPOL, and MOVE elements and user defined MAP elements. The second card of the FXPT statement is ignored if no MAP statements are encountered in the beamline.

Each non-linear element is linearized in a neighborhood about the closed orbit and the betatron functions of the resulting linear system are calculated and printed. A by-product of the closed orbit calculations is a 4x4 matrix, M4, representing the beamline. M4 acts on column vectors of the form $\{X-X_e, X'-X'_e,$

COMMANDS

$Y-Y_e, Y'-Y'_e$ }, where $\{X_e, X'_e, Y_e, Y'_e\}$ is the closed orbit. This matrix as well as its eigenvalues and eigenvectors are printed.

As a first guess for the closed orbit, one starts with the particle state vector P (defined by a FVEC statement) which contains dp/p as its 6-th element. The particle is tracked through BLIN until the i -th iteration where

$$\begin{array}{ll} \text{abs}(X_i - X_{i-1}) < e1 & \text{abs}(X'_i - X'_{i-1}) < e2 \\ \text{abs}(Y_i - Y_{i-1}) < e1 & \text{abs}(Y'_i - Y'_{i-1}) < e2 \end{array}$$

or until the maximum number of iterations, itr , is reached (default: $itr=25$).

OUTPUT OPTIONS:

The integer m determines the output provided by the FXPT statement:

Option

0	$m = 0$	matrix, eigenvalues, closed orbit, eta-orbit at location 0
1	$m = 1$	option 0 and eigenvectors at position 0
2	$m = 2$	option 1 and track closed orbit vector through BLIN
3	$m = 3$	option 1 and track eigenvectors
4	$m = 4$	options 2 and 3 together

The integer n is the number of superperiods upon which the fixed point orbit is to close; if n is negative, then the reflection of BLIN is added to BLIN to form the complete superperiod used in the closed orbit determination -- this feature will not work if $nscl$ is greater than 1;

if $isav=1$, the calculated equilibrium orbit is saved in vector P -- if $isav=0$, it is not saved; the value of $iflg$ determines the total number of rays that are traced through BLIN:

$iflg = 0$	closed orbit ray only
$iflg = 1$... plus 9 neighboring rays
$iflg = 2$ plus 4 rays of linearized equations

COMMANDS

The values d1 and d2 determine the displacement from the reference ray of the neighboring rays used to calculate M4; d3 is the displacement in dp/p for calculating linearized 3x3 matrices.

FXPT recalculates matrices for the elements to correspond to the new closed orbit calculated. In particular, the magnetic rigidity brho, the 3-rd parameter of the MAG elements, will be multiplied by dp/p, the 6-th element of P (a PVEC statement).

To calculate closed orbits, etc. as functions of dp/p, the FXPT statement should be put in a SYNCH subroutine with an INCR of the 6-th PVEC parameter. The MAG statements should not be included in the subroutine (or at least should not be modified by it).

COMMANDS

INCR -- Increment an Input Parameter

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAME INCR  m      Q      TYP  R
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

Increment the m-th parameter of type TYP in statement labeled Q by an amount R; the type of parameter incremented is specified by:

TYP = F	floating point number (default)
= I	integer
= KA	integer corresponding to parameter mm
= KB	integer corresponding to parameter nn

R must correspond to TYP: if TYP = F, R must be a floating point number, otherwise R must be an integer; F may be positive or negative.

Example:

```

  QD  MAG      2.0      -50.0      brho      b0      $
      INCR  2      QD  F      -0.05

```

Here, the second parameter (the gradient) of the MAG statement QD is incremented from -50.0 to -50.05.

INCR changes only the input data, not a corresponding matrix. The matrix will be updated only if it is re-executed, such as in a SYNCH subroutine.

COMMANDS

INV -- Invert a Matrix

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
P      INV      F
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

The transport matrix P is defined to be the inverse of the transport matrix F; i.e., the 3x3 portion of P representing horizontal motion is equal to the inverse of the 3x3 portion of F which represents horizontal motion, etc.

See also, "Symbolic Entry for Inverse of a Matrix," MISCELLANEOUS COMMANDS.

INV2 -- Rotate a Magnet 180 degrees and Reflect It

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAME  INV2      A
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

A is the name of a previously defined magnet; this statement is equivalent to the following example:

A	MAGV	len	grad	brho	b0	e1	e2
K	ROT	A	180.0				
NAME	REF	K					

COMMANDS

KICK -- Dipole Kicker Magnet or Field Error Definition

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAM  KICK  m    C          0.0    brho.    bk.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

Defines a kick in the center of a DRF or creates a field error for a MAG element; the kick or field error deflects the beam vertically upward (+y) if m=1, horizontally inside (-x) if m=2; brho is the magnetic rigidity and bk is the field strength of the kicker; the kick angle is $\theta = (L)(bk)/(brho)$, where L is the length of element C; if L=0, then $\theta = (bk)/(brho)$.

C must be a 7x7 matrix.

If C represents a zero-length drift region, then the kick is taken to be a delta-function kick, changing only the slope of the particle's trajectory. If C represents a drift region of length L, then KICK turns the matrix for C into the matrix for a zero-gradient magnet, neglecting any edge focusing effects. If C represents a magnet of length L, then KICK converts the matrix of C to represent a magnet with a field error dB/B, where dB = bk.

COMMANDS

LIST -- Define Set of Elements

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
Q      LIST                A1   A2   A3   ...   Ak
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

Generates a list of elements A1,...,Ak which will replace the element Q on successive calls for that element. This command is sometimes useful when generating an accelerator lattice made up of misaligned elements.

Example:

```

QF      MAG                L          K          1.
QD      MAG                L          -K         1.
QF1     MOVE               QF         x1         xp1         y1         yp1
QF2     MOVE               QF         x2         xp2         y2         yp2
|       |                 |         |         |         |         |
QF25    MOVE               QF         x25        xp25        y25        yp25
QD1     MOVE               QD         x26        xp26        dy26       yp26
|       |                 |         |         |         |         |
QD25    MOVE               QD         x50        xp50        y50        yp50
C
.C      BML                QFL     .B      QDL     .B
.SP     BML                25( .C      ) .SS
C
QFL     LIST               QF1    QF2    QF3    QF4    QF5    QF6    QF7    QF8    QF9    QF10
          QF11             ...    QF25
QDL     LIST               QD1    QD2    QD3    QD4    QD5    QD6    QD7    QD8    QD9    QD10
          QD11             ...    QD25
C
FXPT    2    1 PV     .SP          1    0    10    0

```

COMMANDS

In this example, the beamline .SP will be made up of 25 standard cells and a long straight section. When the FXPT command is implemented, the beamline will use the list of misaligned quadrupoles in place of the elements QFL and QDL when calculating the closed orbit.

COMMANDS

MAG -- Magnet Definition

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAME  MAG          length.  gradient.  brho.     b0.       el.       e2.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

The transfer matrix is calculated for a magnet of the specified length, gradient, magnetic rigidity Brho, reference bend field Bo, and entrance and exit angles el, e2 (degrees); gradient = dBy/dR, evaluated at Ro, the reference orbit.

If a \$ is present in either field for el or e2 (LEFT ADJUSTED), then the magnet is taken to be a parallel-faced rectangular bending magnet which bends the beam by an amount theta -- in this case, el and e2 are calculated by the program to be $e1 = e2 = \theta/2$.

Positive entrance and exit angles represent magnet faces with normal vectors pointing away from the center of curvature.

The three quantities B' (the gradient), Brho, and Bo can all be scaled together, leading to other possible formats for the MAG statement. Hence, one could also enter the second through fourth magnet parameters as multiples of Brho. That is,

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAME  MAG          length.  K.         1.0       1/rho.    el.       e2.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

Here, $K = \text{gradient}/\text{brho} = (1 / \text{focal length})(1 / \text{magnet length})$ of the equivalent "thin lens" quadrupole.

COMMANDS

Likewise, one may also define the magnet through multiples of $B\rho/\text{length}$. The input parameters then become

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAME  MAG           length.  q.           length.  theta.   el.      e2.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

Here, the length of the magnet is entered twice, but using the bend angle θ as an input parameter may be useful in some instances. The parameter q in this case is the "thin lens" quad strength $q = K \times \text{length} = (1 / \text{focal length})$.

Yet another acceptable format is to enter parameters in terms of B_0 ,

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAME  MAG           length.  k.           rho.     w.       el.      e2.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

where $k = (dBy/dR)/B_0$ (dBy/dR evaluated at R_0), ρ = radius of curvature, and w = relative horizontal curvature = curvature $\times \rho$ (typically = 1.0); w is positive if the center of curvature is in the negative x direction.

COMMANDS

MAGS -- Magnet Definition with Errors

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAM    MAGS    m      MG      kxin.    kxout.    kyin.    kyout.    db.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

NAM = arbitrary name
 m = see below
 MG = previously defined magnet
 kxin = radial displacement of magnet axis at magnet entrance
 kxout = radial displacement of magnet axis at magnet exit
 kyin = vertical displacement of magnet axis at magnet entrance
 kyout = vertical displacement of magnet axis at magnet exit
 db = dB/B or dB (see below)

Given the previously defined magnet MG, new transfer matrix for a new magnet, NAM, is calculated; NAM has the same magnet properties as MG but has relative field error dB/B and misalignments; if m=1, then db = dB/B and the MAG statement defining MG should be of the form

```

MG      MAG      length      k      rho      1.0

```

If m=2, then db = dB and the MAG statement defining MG should be of the form

```

MG      MAG      length      gradient      brho      b0

```

The MAGS command is used in conjunction with the BMIS and EMIS statements. The resulting closed orbit is found using a CYC statement.

See also, BMIS, EMIS

See also, MOVE

See also, Section V-G, Magnet Misalignments

COMMANDS

MAGV -- Define a Vertically Bending Magnet

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAME  MAGV          length.  gradient.  brho.      b0.        e1.        e2.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

Same input definition as MAG (see above); the matrix formed for NAME is equivalent to:

R+	ROTZ	90.0					
R-	ROTZ	-90.0					
B1	MAG	len	grad	brho	b0		
B1V	MMM	R+	B1	R-			

MAP -- Non-linear Transformation Definition

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAM  MAPk  m      PM1  PM2  PM3  ...  PMm
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

The MAP statement allows the user to use non-linear transformations, some of which are already encoded in the SYNCH code, others of which the user may install. The element NAM operates on the particle state vector

$$V = (x, x', y, y', -ds, dp/p, l)$$

through the subroutine MAPk (k=1,2,...,9) with the parameters PM1, PM2, ..., PMm being passed on to the subroutine.

COMMANDS

For more detailed information on the use of the MAP command, see Section III-F, Non-linear Transformation Calculations.

MAT -- General Matrix Definition

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAM   MAT      m    n All.    A21.    A31.    A41.    A51.    A61.
      |         |         |         |         |         |         |         |
      A71.     ...     Am1.
      A12.     A22.     A32.    A42.     ...
      ...     ...     Am2.
      |         |         |
      A1n.     A2n.     A3n.    A4n.     A5n.     A6n.
      A7n.     ...     Amn.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

Defines a matrix NAM, m x n in size; elements All,...,Amn are floating point numbers; the first row of a new column of the matrix must begin on a new card.

COMMANDS

MESH -- Loop Through Subroutine Variables

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAME  MESH   k      SBR
          m1     A1      min1.    max1.    incl.
          m2     A2      min2.    max2.    inc2.
          |      |      |
          mk     Ak      mink.    maxk.    inck.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

```

k           = number of variables
SBR         = name of subroutine containing elements A1,..,Ak
m1,..,mk    = position of floating point variable in defining
              statement of element A1,..,Ak
A1,..,Ak    = name of element containing the variable
min1,..,mink = minimum value of variable
max1,..,maxk = maximum value of variable
incl,..,inck = increment of the variable

```

The MESH command loops through the subroutine SBR for all values of each of the k variables between the minimums and maximums provided; the variables are incremented by an amount inc each iteration; the mesh is formed by a series of loops: the first variable forms the outer loop, while the k-th variable forms the inner loop.

See example, next page.

COMMANDS

EXAMPLE:

```

SR    SUB
QF    MAG      2.0      0.5      1.0
QD    MAG      2.0     -0.05     1.0
.C    BML      QF      0      QD      QD      0      QF
C     CYC      .C
      END
      MESH    2      SR
                2      QF      0.5      1.0      0.1
                2      QD     -1.0     -0.5     0.1

```

The above code will produce a CYC print-out of the cell C for all of the combinations of the gradients of the magnets QF and QD from 0.5 to 1.0 kG/m for QF and -1.0 to -0.5 kG/m for QD.

MMM -- Matrix Multiplication

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAME  MMM      m      Q1      Q2      Q3      Q4      ...      Qm
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

NAME = arbitrary name
|m| = number of matrices to be multiplied
Q1,Q2,...,Qm = names of previously defined elements or beamlines

m is optional

Multiplies the matrices Q1, Q2,...,Qm together and stores the result in the matrix NAME. If m is negative, then the list of elements or beamlines will represent one-half of a symmetrically reflected beamline and the matrix for this total beamline will be stored in NAME.

COMMANDS

MOVE -- Perform an Element Misalignment

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAM  MOVE          A      VC
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

Defines a new element, NAM which represents the element A which has been misaligned according to the vector VC which contains

$$VC = (hx, dhx, hy, dhy, hs, \text{theta-s})$$

where hx, hy, and hs are translations along the x, y, and s axes (in meters), and dhx, dhy (radians) are rotations in the x-s and y-s planes, and theta-s is the rotation about the s-axis (radians). VC must be defined using a VEC statement.

When using the MOVE statement, closed orbit calculations are to be made using the FXPT command. (If magnet misalignments are studied using the BMIS, MAGS, etc. statements, then the closed orbit calculations are done using a CYC command. This method does not allow for coupling between the horizontal and vertical motion. The MOVE command does allow for coupling.)

Alternate format for the MOVE statement, not using a VEC statement:

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAM  MOVE          A      hx.      hy.      hs.      theta-s.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

COMMANDS

MXV -- Matrix-Vector Multiplication

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
VF      MXV      M      VI
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

M = previously defined matrix
VI = previously defined vector (result of PVEC statement)
VF = resulting 7-component vector

The 7-component vector VF is defined to be the result of the matrix M operating on the 7-component vector VI. If M is not 7x7, it is converted to 7x7 before the multiplication is carried out.

NCPY -- Prevent Print Out of SYNCH Input Statements

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
      NCPY
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

Disables echoing of SYNCH input statements upon execution. To re-enable echoing, use COPY statement.

COMMANDS

NPOL -- Define N-pole Magnet Matrix

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAM  NPOL  n   j length.  coeff.  brho.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

```

      NAM      = arbitrary name
      n        = order of perturbation in Hamiltonian; m = n-1
      j        = 0, 1 for normal, skew n-pole thin lens
      length   = effective length of element
      coeff    = Taylor series expansion coefficient of median-plane
                 expression of field
      brho     = magnetic rigidity

```

Defines a thin lens n-pole magnet. The magnetic field is taken to be of the form

$$j=0: \quad B = B_y + iB_x = [(coeff)m / m!] z^{*m} \qquad z = x + iy$$

$$j=1: \quad B = B_x - iB_y = [(coeff)m / m!] z^{*m} .$$

Elements corresponding to various values of N are:

n = 2	quadropole	n = 3	Sextupole
n = 4	octupole	n = 5	Decapole
etc. ...			

COMMANDS

ORBC -- Perform Closed Orbit Correction Calculation

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
      ORBC   m   n FX   CY   MON   COR   deltam.   deltax.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

m = 0, 1 for horizontal, vertical correction
 n = print option (see below)
 FX = name of FXPT statement defining initial orbit
 CY = name of CYC statement with identical beamline
 MON = name of elements in beamline where displacements are
 assumed to be measured
 COR = name of correction elements in beamline
 deltam = assumed measurement error (deltam/2)
 deltax = tolerable range of final residual orbit errors

The ORBC command calculates optimized correction element strengths necessary to correct a given closed orbit through an accelerator with field errors.

The FXPT statement used must have a 1 in column 55. This tells the program to write the closed orbit information to Tape 12.

The CYC statement used should have $m = 12$ or $m = -12$ (i.e., right justified to column 15). Again, this writes betatron function information to tape.

There should, of course, be many MON's and COR's throughout the beamline.

The maximum number of iterations will be $\text{abs}(n)$. If n is negative, all intermediate iterations as well as the matrix relating orbit displacements to correction elements is printed out. If n is positive, only first and last iterations are printed and matrix print is suppressed.

COMMANDS

P -- Restore Page

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
P      ----                ----      text                ----                ----
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

Causes the next line of print to begin on a new page; the text (in columns 2 through 80) is optional -- if present, it will be printed at the top of the new page.

COMMANDS

PRNT -- Print Element Parameter

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
      PRNT   m   n A1   A2   A3   ...   Ak
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

Prints out the m-th parameter of type n of each element or variable listed, A1,A2,A3,...,Ak.

Valid values of n:

n = 1	floating point number
2	alphanumeric characters
3	integer
4	(not used)
5	internally calculated floating point number

Example:

```
QF   MAG           len      k           brho
     PRNT   2   1 QF
```

The PRNT statement will cause the second parameter of QF, k, to be printed.

COMMANDS

PTAB -- Print Tables Generated by TAB Statement

```
-----1-----2-----3-----4-----5-----6-----7-----8
      PTAB  m      T1  T2  T3  ...  Tm
-----1-----2-----3-----4-----5-----6-----7-----8
```

m = number of tables to be printed (<11)

T1,...,Tm = name of TAB statement that generated a table

Prints the tables that have been generated by a TAB statement.

See also, TAB

COMMANDS

PVEC -- Define a Particle Vector

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
PV      PVEC      m      n  x1.      xpl.      y1.      ypl.      -ds1.      dp/p1.
          |          |          |          |          |          |          |
          x2.      xp2.      y2.      yp2.      -ds2.      dp/p2.
          |          |          |          |          |          |
          xm.      spm.      ym.      ypm.      -dsm.      dp/pm.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

Define m number of particle vectors; if m is left blank, it is taken to be 1; if n = 1 or blank, the 7-th component of the vector will be 1; if n=0, the 7-th component of the vector will be 0.

NOTE: When incrementing with the INCR statement, it must be remembered that 7 values are stored for each vector. Therefore, if one wished to increment the first parameter of the second vector of the statement PV above, the INCR statement would ask to change the 8-th parameter of PV.

RAND -- Generate Random Number

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
R      RAND      m      n  x.      y.      z.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

The floating point number R is set equal to $x + g*y$, where g is a uniformly distributed random number between -1/2 and +1/2. If m and n are positive, then the random number generator is run $m*n$ times. (If $m=n=0$, the random number generator is run once.)

COMMANDS

If $0 < z < 1$, then the random number is reset with seed value z . (Both of these features operate BEFORE the number R is computed.)

REF -- Matrix Reflection

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
R      REF      P
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

Produces a new set of matrices R which represents the reflection of P ; i.e., if P represents horizontal and vertical transfer matrices through a sequence of elements A_1, A_2, \dots, A_n , then R represents the transfer matrices through the sequence $A_n, A_{n-1}, \dots, A_2, A_1$; i.e. if the horizontal (or vertical) portion of P is represented by

$$P_x = \begin{pmatrix} a & b & e \\ c & d & f \\ 0 & 0 & 1 \end{pmatrix}, \quad \text{then} \quad R_x = \begin{pmatrix} d & b & -de+bf \\ c & a & -ce+af \\ 0 & 0 & 1 \end{pmatrix}.$$

COMMANDS

REPL -- Replace One Parameter With Another

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAME  REPL   m      Q      TYP  R
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

Replace the m-th parameter of type TYP in statement labeled Q with the value R; the type of the parameter replaced is specified by:

TYP = F	floating point number (default)
= H	symbolic floating point number
= I	integer
= KA	integer corresponding to parameter mm
= KB	integer corresponding to parameter nn

R must correspond to TYP: if TYP = F, R must be a floating point number; if TYP = H, R must be a symbolic floating point number; otherwise R must be an integer; F may be positive or negative.

Example:

```

QD      MAG      2.0      -50.0      brho      b0      $
      REPL      2      QD      F      -50.05

```

Here, the second parameter (gradient = -50.0) of the MAG statement QD is replaced by -50.05.

REPL changes only the input data, not a corresponding matrix. The matrix will be updated only if it is re-executed, such as in a SYNCH subroutine.

COMMANDS

ROT -- Perform a Rotation Transformation

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAME  ROT          R          theta.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

Perform a rotation transformation of theta degrees on the matrix R and store the result as matrix NAME.

See also, ROTZ.

COMMANDS

ROTZ -- Define a Matrix Representing a Rotation about the z-axis

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
R      ROTZ      theta.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

R represents a rotation about the z-axis of theta degrees; if x and y are the original coordinates and xl and yl are the coordinates after the rotation of coordinates, then

$$\begin{aligned}
 x_l &= x \cos(\theta) + y \sin(\theta), & x_l' &= x' \cos(\theta) + y' \sin(\theta) \\
 y_l &= -x \sin(\theta) + y \cos(\theta), & y_l' &= -x' \sin(\theta) + y' \cos(\theta)
 \end{aligned}$$

and R is given by

$$R = \begin{pmatrix}
 \cos(\theta) & 0 & \sin(\theta) & 0 & 0 & 0 & 0 \\
 0 & \cos(\theta) & 0 & \sin(\theta) & 0 & 0 & 0 \\
 -\sin(\theta) & 0 & \cos(\theta) & 0 & 0 & 0 & 0 \\
 0 & -\sin(\theta) & 0 & \cos(\theta) & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{pmatrix}$$

NOTE: In order to obtain proper results when using the ROTZ instruction, matrices must be declared as 7x7's (see statement SIZE).

COMMANDS

RUN -- Start of Run

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAME  RUN          ---          ----          ---          text          ---          ----          ---
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

Signals the start of run NAME where NAME is the name or number of the run (up to four alphanumeric characters); arbitrary remarks may be written in the data region (columns 21-80); NAME, RUN, and the text will be printed on the output.

SHF7 -- Define a 7x7 Shift Matrix

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
SM    SHF7          R17          R27          R37          R47          R57          R67
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

Defines a 7x7 shift matrix, SM, of the form:

$$\text{SM} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \text{R17} \\ 0 & 1 & 0 & 0 & 0 & 0 & \text{R27} \\ 0 & 0 & 1 & 0 & 0 & 0 & \text{R37} \\ 0 & 0 & 0 & 1 & 0 & 0 & \text{R47} \\ 0 & 0 & 0 & 0 & 1 & 0 & \text{R57} \\ 0 & 0 & 0 & 0 & 0 & 1 & \text{R67} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

COMMANDS

SIZE -- Define Size of Matrices

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
      SIZE      k
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
  
```

Defines matrices used in SYNCH as either 3x3 pairs (k=3) or as 7x7's (k=7); as most matrices used will be 3x3 pairs, this is the default; some statements require 7x7's, such as ROTZ, for example; if 3x3 pairs are used, they are assumed to operate on particle state vectors of the form

$$\begin{pmatrix} x \\ x' \\ d \end{pmatrix}, \begin{pmatrix} y \\ y' \\ d \end{pmatrix}$$

where x = the particle's displacement, x' = the slope of the particle's trajectory, and $d = dp/p$; if 7x7 matrices are used, they are assumed to operate on particle state vectors of the form

$$\begin{pmatrix} x \\ x' \\ y \\ y' \\ h \\ d \\ l \end{pmatrix}$$

where $h = -ds = -(\text{change in path length from reference orbit})$; 7x7 matrices are most often used in SYNCH for calculations of closed orbits due to magnet misalignments or for off-momentum particles.

COMMANDS

SMIN -- MINUIT Access from SYNCH

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAME  SMIN    m      STAT1
                        STAT2
                        |
                        STATm
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

This command allows the user to make calls to MINUIT; upon execution of the SMIN statement, the m MINUIT commands, STAT1,...,STATm, are performed; input has the same structure as MINUIT commands, but shifted to start in column 21, with five words of floating point data per command.

EXAMPLE:

```

FIT    SMIN    4      PRINTOUT  2.
                        SEEK      500.
                        SIMPLEX   1500.      0.1
                        END RETURN

```

COMMANDS

SOLV -- Betatron Function Fitting Using MINUIT

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
SOLV   m     n  SBR  TB    ips1 ips2  itr  itol
      BNM1 OPR1  jp11 jp12 jp13 jp14 jp15 jp16 bfit1.  sigma1.
      BNM2 OPR2  jp21 jp22 jp23 jp24 jp25 jp26 bfit2.  sigma2.
      |      |      |      |      |      |      |      |
      BNMm OPRm  jpm1 jpm2 jpm3 jpm4 jpm5 jpm6 bfitm.  sigmam.
      VR11 VR12 VR13 VR14 VR15  ipm1 bdlol.  bdhil.  stpsz1.
      VR21 VR22 VR23 VR24 VR25  ipm2 bdlo2.  bdhi2.  stpsz2.
      |      |      |      |      |      |      |      |
      VRn1 VRn2 VRn3 VRn4 VRn5  ipmn bdlon.  bdhin.  stpszn.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

m = number of constraint lines
 n = number of variable lines
 SBR = subroutine in which TB statement is located
 TB = name of TRKB statement found in subroutine SBR
 ips1, ips2 = start, end positions to be used in TRKB statement
 itr = maximum number of calls to MINUIT (default = 1000)
 itol = exponent of desired tolerance on FCN (see FCN below)
 BNM = names of betatron function constraints (see table below)
 jp = location of BNM in TRKB statement
 OPR = "SUM", "DIF", or "blank" (see below)
 bfit = desired value of betatron function
 sigma = desired tolerance on betatron function being fitted
 VR = name of SYNCH variable to be used in fitting process
 ipm = parameter of VR's to be varied in the fitting process
 bdlo = lower bound of the ipm-th parameter of VR's
 bdhi = upper bound of the ipm-th parameter of VR's
 stpsz = step size used when varying ipm-th parameter of VR's

COMMANDS

The ipm-th parameters of the SYNCH variables VR are varied through various MINUIT least-squares minimization routines in an attempt to fit the betatron functions BNM to their desired values, bfit; the functions BNM correspond to locations jp in a beamline defined in a TRKB statement, TB; a "fit" has been accomplished when the value of the function

$$FCN = \text{sum} \{ [(value - bfit) / itol]**2 \} / m$$

is below 10^{**itol} ; if the value of FCN does not reach 10^{**itol} within itr calls to MINUIT, execution of the SOLV statement is halted; the default MINUIT commands for the SOLV statement are:

```
PRINTOUT      0.
MINIMIZE      1000.      1.0
END RETURN
```

Valid BNM's are given by the list below:

QX,QY	phase advance/2pi
BX,BY	beta function
AX,AY	alpha function
GX,GY	gamma function
X ,Y	momentum dispersion
DX,DY	slope of X,Y
S	cumulative path length
THET	cumulative bending angle
AXAY	equivalent to AX, X, and AY

The OPR parameter allows one to fit to the difference or sum of the betatron functions at different locations if desired. If OPR equals "SUM" or "DIF" the sum or differences of the values of the last five betatron functions with the first one on the constraint line are made to be equal to bfit.

See the example of a SOLV routine in Section VII, "EXAMPLES OF SYNCH RUNS."

COMMANDS

STOP -- End of Job

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
      STOP
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

Signals the end of the entire job, or set of runs

See also, FIN

COMMANDS

STRN -- Design a Matching Straight Section

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAM  STRN  m      POS  MQ
      mu.    lt.    grad.  rho.
      x-beta. y-beta. x-alpha. y-alpha.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

Designs a matching straight section with phase advance μ to be inserted at the point in the accelerator whose betatron functions are given by the matrix POS, or by the values entered on line 3 (depending upon m ; see below); the previously defined magnet MQ is given a gradient, grad, and radius of curvature, rho, and an initial guess as to its required length, lt. The length of the quadrupole is varied until the desired phase advance is obtained. The final magnet length and necessary drift length of the straight section is printed out.

If $m = 1$, the betatron functions are given by the matrix POS and line 3 should be omitted. If $m = 2$, the betatron functions are read in from line 3.

The transfer matrix representing the final straight section is stored in NAM.

COMMANDS

STRP -- Design a pi-type Straight Section

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
Q      STRP  m   n CELL      bo.      bq.      aper.      ltot.      lsep.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

Calculates the parameters and transfer matrices for a pi-type straight section which may be inserted into a previously defined cell, CELL. The straight section may be either DFDF or FDFD. The values of the betatron functions alpha and beta at the beginning and end of the pi-SS match the corresponding quantities at the points of insertion in the cell. The type of straight section to be generated is determined by m: m = 1 gives DFDF; m = 2 gives FDFD.

In addition to CELL, the user must input bo, the magnetic field strength at the magnet center line; bq, the magnetic field strength at the pole faces of the quadrupole magnet; aper, the aperture of the vacuum chamber through the cell (semi-diameter); ltot, the total length of the insertion; and lsep, the distance separating the F and D (or D and F) magnets.

The integer n determines the point in the cell where the straight section is inserted, as shown in the diagram below:

Position:	1	2	3	4	5	6	7	8	9	10	11
Matrix:	QS	QD	QT	QF	QS	QS	QD	QT	QF	QS	QS
Length:	lend	lmag	lsep	12	ldrf	12	lsep	lmag	lend	lend	lend

(lend = ldrf/2)

COMMANDS

Output includes: betatron functions, magnet lengths, m , n , drift lengths, maximum aperture of the insertion, maximum drift length in the insertion, all input data. The matrices for the quadrupoles (QF and QD), the matrices for the drift spaces of lengths $ldr/2$ and $lsep$ (QS, QT, respectively), and the total insertion matrix (Q) are computed and stored.

COMMANDS

STR2 -- Design a Collins Straight Section

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
Q      STR2  m      CELL      bo.      bq.      aper.      mu/2pi.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

Calculates the parameters and transfer matrices for a Collins straight section which may be inserted into a previously defined cell, CELL. The values of the betatron functions alpha and beta at the beginning and end of the CSS match the corresponding quantities at the point of insertion in the cell. The type of CSS to be generated and the point of insertion is specified by m: if m = 1, the CSS will be inserted at cell position 4 (see below) and it will be of the form DF; if m = 2, the CSS will be inserted at cell position 10 and it will be of the form FD. The quantities bo, bq, and aper have the same meanings as in STRP. The desired value of the phase advance, mu, is entered in units of 2pi radians.

Position:	1	2	3	4	5	6	7
Matrix:	QE	QD	QS	QS	QF	QE	
Length:	lend	lmag	ldrift	lmag	lend		

STR2 computes and prints out: the betatron functions; the input data; the drift lengths; the total SS length; the maximum aperture, Aq; GRAD = Bq/Aq. The following matrices are calculated and stored: QF, QD, QS, QE, and Q (the total straight section matrix).

COMMANDS

STR4 -- Design a Four-element Antisymmetric Straight Section

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
K      STR4          P      Q      MG          mu/2pi.  lend.      lmag.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

Calculates the parameters and matrices for a straight section which will replace or be inserted into a previously defined structure. The name of the new straight section is K, a one character symbol. P is a previously defined matrix which gives the initial position of replacement K. Q is an element previously defined by a MAG statement. The outer elements KOF and KOD have the same magnet parameters as Q, except that their length, lmag, is given as input for K. The end drift length of the straight section is given by lend. MG is a previously defined MAG element. The inner elements KIF and KID will have the same magnet parameters as MG, except that their length, l2, is calculated internally.

Position:	1	2	3	4	5	6	7	8	9	10	11
Matrix:	KE	KOF	KT	KID	KS	KS	KIF	KT	KOD	KE	
Length:	lend	lmag	a	l2	L	L	l2	a	lmag	lend	

K will replace a section of previously defined structure starting at P and running to the next anti-homologous point. (In a synchrotron periodic structure, if the profile parameter k is an odd function of the path length S about some point S_0 , then S_0+S and S_0-S are anti-homologous points if $k(S_0-S) = -k(S_0+S)$.)

The following elements are generated and saved: K (entire insertion matrix); KIF, KID (inner elements); KOF, KOD (outer elements); KE (end drift space); KT (drift space between inner and outer magnets); and KS (half the

COMMANDS

inner drift space). The lengths of all elements are printed out as well as the betatron functions.

SUB -- Begin Definition of a SYNCH Subroutine

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
SBR   SUB
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

The statements following this card are taken to be the defining sequence of a SYNCH subroutine named SBR. the subroutine SBR may only contain the following SYNCH statements:

DRF, MAG, MMM, MXV, REF, SHF, INV, **, SUM, =, CYA, CYC, BML, CRD, MAGV, EQUL, TRKB, VEC, SHF7, MOVE, TAB, CALC, END.

See also, END

COMMANDS

TAB -- Tablulate Values to be Printed

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAM  TAB    m    n S1    S2    S3    ...  Sm
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

NAM = arbitrary name
 m = number of columns to be printed (<11)
 n = print option (see below)
 S1,...,Sm = name of element to be printed

The quantities S1,...,Sm are saved until a PTAB statement is encountered; These quantities MUST be defined scalar quantities or CYC outputs as described as follows:

If Si refers to a CYC statement, then S is the first letter of the CYC name and is followed by a number from 1 to 6; the element name should really consist of only one letter; the number refers to the CYC output as follows:

1	betax max		4	nux
2	betay max		5	nuy
3	x-dispersion		6	transition gamma

If n=0, then CYC output is printed as usual; if n=1, then CYC output is omitted (assuming that at least one of the elements Si refer to a CYC statement).

The table being saved can hold up to 100 entries; thus, when this limit is reached, the table will automatically be printed.

No more than 10 TAB statements may be operative at one time.

See also, PTAB

COMMANDS

TRK -- Track Particles Through Linear or Non-linear Transformations

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAME TRK      m      n V      BLIN MAT  V0      ntr  iof isav imod  ipr  ipl
                xmax.    dxmax.    ymax.    dymax.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

```

m              = number of particles; these are stored in V
n              = input option (see below)
V              = name of PVEC statement containing initial particle state
BLIN           = name of the beam line through which to track
MAT            = name of matrix which gives betas at print positions
V0             = name of reference particle vector (PVEC statement)
ntr            = number of transits of BLIN to be made by each particle
iof            = output frequency (see below)
isav           = save tracked vectors (if =1)
imod           = tracking mode (see below)
ipr            = print option (ipr=0: print;  ipr=-1: noprint)
ipl            = plot option (ipl=1: plot;   ipl=-1,0: noplot)
xmax,...,dymax = scales for phase plots (mm, mrad)
                (note: second card must always be used)

```

The m particles are tracked through the beamline BLIN, which may contain non-linear elements, ntr times. The actual particle vector being tracked is given by $U = V0 + V$. The frequency of the output is dependent upon iof :

$iof = 00001$ implies output at each position of the beamline each transit

$iof = 0r00p$ implies output at the p -th position every r -th transit.

The vectors $V0$ and V may be input in different formats according to n :

COMMANDS

```

n = 0      V = { x, dx, y, dy, -ds, dp/p } (meters,radians)
n = 1      V = { x, dx, y, dy, -ds, dp/p } (mm,mrad, dp/p in o/oo)
n = 2      V = { ex, psix, ey, psiy } (mm-mrad, degrees)

```

If imod=0,1 then will track with stored matrices and MAPS; if imod=2 then will track saving only the cumulative matrix in the storage of TRK; if imod=3 then will not save matrix upon tracking.

TRKB -- Track Betatron Functions Through a Beamline

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
TB      TRKB      m      n BLN  MTR      ips1 ips2
                nux0.      betax0.      alphax0.      gammax0.      Dx0.      D'x0.
                nuy0.      betay0.      alphay0.      gammay0.      Dy0.      D'y0.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

```

m,n = specify how many elements of beamline BLN may be used in the
      TRKB statement (from element ips1 to element ips2)
BLN = name of beamline (BML) through which to track betatron functions
MTR = name of matrix from which to extract initial betatron functions
ips1 = begin tracking at ips1-st element of beamline BLN
ips2 = end tracking at ips2-nd element of beamline BLN

```

Obtains betatron functions from matrix MTR and tracks them through the specified portion of beamline BLN, printing the values of the functions at the end of each element of BLN; if the "MTR" field is left blank, the initial values of the betatron functions are obtained from the second and third lines of the TRKB statement as shown above; if MTR is specified, the second and third lines are ignored and may be left blank; every TRKB statement must consist of three lines, however, whether or not MTR is specified; it should be pointed out that a BML statement is recursive (see BML) so that if the beamline BLN is composed of many other BML's, then the total number of elements in BLN may be quite different than the number of elements specified in the BLN-BML statement itself.

COMMANDS

VEC -- Vector Definition

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
NAM   VEC      m      v1.      v2.      v3.      ...      vm.
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

Defines the m -column vector with components v_1, v_2, \dots, v_m .

COMMANDS

WBE -- Write Betatron Functions of Matrices

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
      WBE      m      n A1      A2      A3      ...  Am
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8

```

The betatron functions for the m previously defined transfer matrices labeled $A1, A2, A3, \dots, Am$ are printed out; each matrix can be represented as

$$A = \begin{pmatrix} \cos(\mu) + (\alpha)\sin(\mu) & (\beta)\sin(\mu) & A13 \\ -[(1+\alpha^2)/\beta]\sin(\mu) & \cos(\mu) - (\alpha)\sin(\mu) & A23 \\ 0 & 0 & 1 \end{pmatrix}$$

If $n = 0$, the WBE statement will print β , α , $\mu/2\pi$, D , D' , and W for each plane, where

$$\begin{aligned}
 D &= (1/2)[A13 + (\text{inv}(A))13] / [1 - \cos(\mu)], \\
 D' &= (1/2)[A23 + (\text{inv}(A))23] / [1 - \cos(\mu)],
 \end{aligned}$$

and

$$W = \text{sqrt}(\beta).$$

If $|\cos(\mu)| > 1$, then all betatron functions will be printed as 0.0 .

If $n = 1$, then μ will be determined from

$$\mu = \begin{cases} \arccos(\text{trace}/2), & \text{if } |\text{trace}| < 2 \\ \text{arccosh}(\text{trace}/2), & \text{if } |\text{trace}| > 2 \end{cases}$$

and $\text{trace}/2 = (A11 + A22)/2$ will be printed out in place of $W = \text{sqrt}(\beta)$.

COMMANDS

WMA -- Write Matrices

```
-----1-----2-----3-----4-----5-----6-----7-----8
      WMA      m      A1   A2   A3   ...   Am
-----1-----2-----3-----4-----5-----6-----7-----8
```

The previously defined pairs of transfer matrices (or 7x7 matrices, if SIZE 7 has been issued) labeled A1,A2,A3,...,Am are printed out.

COMMANDS

= -- Equate to Floating Point Number

USE 1:

```

-----1-----2-----3-----4-----5-----6-----7-----8
NAME  =          float.
-----1-----2-----3-----4-----5-----6-----7-----8

NAME    = arbitrary name
float   = floating point number
    
```

The variable NAME is given the value "float," which is a floating point number; NAME may now be substituted for a floating point variable in any SYNCH input statement and the value float will be used; use of the symbolic floating point may not be carried more than one step -- see examples below.

Example: acceptable

```

g      =          1.5
F      MAC          2.4      g      36.8      1.0
    
```

Example: NOT ACCEPTABLE

```

g      =          1.5
h      =          g
F      MAG          2.4      g      36.8      1.0
    
```

COMMANDS

USE 2: Arithmetic Operations

```
-----1-----2-----3-----4-----5-----6-----7-----8
C      =          operand1. operator  operand2.
-----1-----2-----3-----4-----5-----6-----7-----8
```

C = arbitrary name
operand1,2 = floating point numbers, or names of scalars or matrices
operator = identifies operation to be performed

operator may be +, -, *, /, or ** indicating operation of addition, subtraction, multiplication, division, or raising to a power, respectively; if operand1,2 are symbolic, they must have been previously defined; operand1,2 must both be of the same type -- scalars (but could be one symbolic and one a floating point number), or matrices; matrix operations work only for operator = +, -, *.

** -- Matrix to a Power

```
-----1-----2-----3-----4-----5-----6-----7-----8
B      **      k      A
-----1-----2-----3-----4-----5-----6-----7-----8
```

The transfer matrix B is defined to be the transfer matrix A raised to the k-th power.

COMMANDS

. -- Comment

```
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
.      - - - - -          - - - - -          text          - - - - -          - - - - -
-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5-----+-----6-----+-----7-----+-----8
```

A period in column one causes any text entered in columns 2 through 80 to be printed.

MISCELLANEOUS

C. Miscellaneous Features

Symbolic Entry for the Negative of a Symbolic Floating Point Number:

If a Symbolic Floating Point variable (V, say) has been previously defined, the negative of that variable can be used in a SYNCH statement by denoting it by -V.

Example:

```
grd   =          10.0
QF    MAG        len      grd      brho      b0      $
QD    MAG        len      -grd     brho      b0      $
```

Here, magnet QF will have gradient 10.0 and magnet QD will have gradient -10.0 .

MISCELLANEOUS

Symbolic Entry for Inverse of Matrix:

If a matrix has been previously defined, its inverse may be used in a computation by preceding its name with a slash (/).

Example:

A	MAG	.9	280.	13373.	0.
B	MAG	.9	-280.	13373.	0.
P	MMM	A	/B		

Here, the matrix multiplication will be $P = \text{inv}(B) \times A$.

MISCELLANEOUS

Internally defined matrices:

a) Unit matrix:

$$(I) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

b) Symplectic matrix:

$$(S) = \begin{pmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The parentheses must be included when referring to these matrices.

MISCELLANEOUS

Inputing Initially Deactivated SYNCH Statements

A SYNCH statement can be initially placed in the deactivated state (see DELE) by having a dash (-) in column one of the statement line. The statement will not be executed unless it has been activated by an ACT statement (see ACT). Whole SYNCH subroutines can be bypassed until needed by placing a "-" in front of the CALL statement, for example.

V. SYNCH MATHEMATICAL FORMULATION

SYNCH MATHEMATICAL FORMULATION

A. Primary Element Definitions

The coordinate system employed by the program SYNCH is a curvilinear system of right-handed orthogonal coordinates x-y-s where s = the longitudinal position along the reference orbit, x = the horizontal (radial) displacement from the reference orbit, and y = the vertical displacement from the reference orbit. (The reference orbit itself follows an isomagnetic line through each element.) The trajectory, or state of a particle traversing through this coordinate system is represented by the vectors $X = (x, x', dp/p)$, and $Y = (y, y', dp/p)$, where $x' = dx/ds$, $y' = dy/ds$ and dp/p is the momentum error with respect to the reference particle. Most accelerator or beamline elements can be represented as a pair of 3x3 matrices R_x and R_y which act on the above particle state vectors:

$$X_{out} = R_x X_{in}$$

$$Y_{out} = R_y Y_{in}$$

where the matrices are of the form

$$R = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ 0 & 0 & 1 \end{pmatrix}.$$

$$[\det(R) = 1]$$

For some types of calculations one may wish to investigate coupling between horizontal and vertical motion. For these cases, a 7x7 matrix is used to describe each beamline element. This matrix, M, acts on the particle state vector $X = (x, x', y, y', -ds, dp/p, 1)$:

$$X_{out} = M X_{in} .$$

The matrix M for a non-rotated magnet, for example, would be of the form

$$M = \begin{pmatrix}
 Rx11 & Rx12 & 0 & 0 & 0 & Rx13 & 0 \\
 Rx21 & Rx22 & 0 & 0 & 0 & Rx23 & 0 \\
 0 & 0 & Ry11 & Ry12 & 0 & Ry13 & 0 \\
 0 & 0 & Ry21 & Ry22 & 0 & Ry23 & 0 \\
 M51 & M52 & M53 & M54 & 1 & M56 & 0 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{pmatrix}$$

(A 7x7 matrix is used in order to facilitate magnet misalignment calculations -- see Section V-G.) Bending magnets are assumed to bend in the horizontal plane. If the matrix for a magnet which bends in the vertical plane is desired, then a rotational transformation of coordinates is made before entering and upon exiting the magnet. (See below.)

For simplicity, the 3x3 matrices for the various types of elements will be introduced under the assumption that the appropriate 7x7 matrices can be derived if necessary.

Linear Elements

DRIFT REGION -- length L

$$Rx = \begin{pmatrix} 1 & L & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad Ry = \begin{pmatrix} 1 & L & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

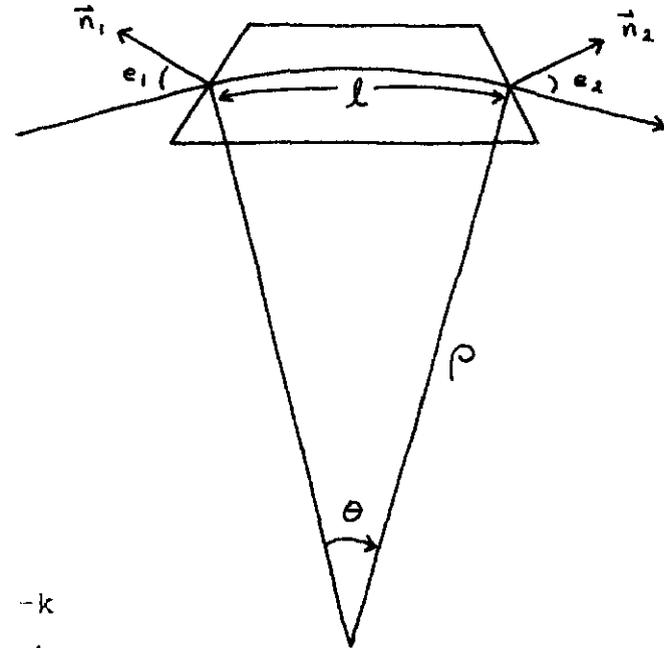
MAGNETS --

$$B' = (dB/dR) \text{ at } R_0$$

$$B_0 = (B_y) \text{ at } R_0$$

$$k = B'/B_0$$

where R_0 is the reference orbit



$$k_x = k + 1/\rho$$

$$K_x = |k_x/\rho|$$

$$\Phi_x = \sqrt{K_x} l$$

$$k_y = -k$$

$$K_y = |k_y/\rho|$$

$$\Phi_y = \sqrt{K_y} l$$

The transport matrices for motion through a magnet are given by

$$R_x = E_{2x} T_x E_{1x}$$

$$R_y = E_{2y} T_y E_{2y}$$

where E_1 and E_2 are edge-focusing matrices. The T matrices for various magnets as calculated by SYNCH are shown below.

HORIZONTALLY (RADIALLY) FOCUSING COMBINED FUNCTION MAGNET:

$$\begin{aligned}
 T_x &= \begin{pmatrix} \cos \Phi_x & \frac{1}{\sqrt{K_x}} \sin \Phi_x & \frac{1}{K_x \rho} (1 - \cos \Phi_x) \\ (-\sqrt{K_x}) \sin \Phi_x & \cos \Phi_x & \frac{1}{\sqrt{K_x \rho}} \sin \Phi_x \\ 0 & 0 & 1 \end{pmatrix} \\
 T_y &= \begin{pmatrix} \cosh \Phi_y & \frac{1}{\sqrt{K_y}} \sinh \Phi_y & 0 \\ \sqrt{K_y} \sinh \Phi_y & \cosh \Phi_y & 0 \\ 0 & 0 & 1 \end{pmatrix}
 \end{aligned}$$

VERTICALLY FOCUSING COMBINED FUNCTION MAGNET:

$$\begin{aligned}
 T_x &= \begin{pmatrix} \cosh \Phi_x & \frac{1}{\sqrt{K_x}} \sinh \Phi_x & \frac{1}{K_x \rho} (\cosh \Phi_x - 1) \\ \sqrt{K_x} \sinh \Phi_x & \cosh \Phi_x & \frac{1}{\sqrt{K_x \rho}} \sinh \Phi_x \\ 0 & 0 & 1 \end{pmatrix} \\
 T_y &= \begin{pmatrix} \cos \Phi_y & \frac{1}{\sqrt{K_y}} \sin \Phi_y & 0 \\ (-\sqrt{K_y}) \sin \Phi_y & \cos \Phi_y & 0 \\ 0 & 0 & 1 \end{pmatrix}
 \end{aligned}$$

HORIZONTALLY BENDING WEDGE MAGNET ($k = 0$) :

$$T_x = \begin{pmatrix} \cos \frac{l}{\rho} & \rho \sin \frac{l}{\rho} & \rho(1 - \cos \frac{l}{\rho}) \\ -\frac{1}{\rho} \sin \frac{l}{\rho} & \cos \frac{l}{\rho} & \sin \frac{l}{\rho} \\ 0 & 0 & 1 \end{pmatrix}$$

$$T_y = \begin{pmatrix} 1 & l & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Note: $R_x = T_x$; $R_y = T_y$ ($e_1 = e_2 = 0$)

For rectangular magnets, $e_1 = e_2 = \theta/2$

HORIZONTALLY (RADIALLY) FOCUSING QUADRUPOLE MAGNET :

$$T_x = \begin{pmatrix} \cos \Phi_x & \frac{1}{\sqrt{K_x}} \sin \Phi_x & 0 \\ (-\sqrt{K_x} \sin \Phi_x & \cos \Phi_x & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$T_y = \begin{pmatrix} \cosh \Phi_y & \frac{1}{\sqrt{K_y}} \sinh \Phi_y & 0 \\ (\sqrt{K_y} \sinh \Phi_y & \cosh \Phi_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

VERTICALLY FOCUSING QUADRUPOLE MAGNET :

$$T_x = \begin{pmatrix} \cosh \Phi_x & \frac{1}{\sqrt{K_x}} \sinh \Phi_x & 0 \\ (\sqrt{K_x} \sinh \Phi_x & \cosh \Phi_x & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$T_y = \begin{pmatrix} \cos \Phi_y & \frac{1}{\sqrt{K_y}} \sin \Phi_y & 0 \\ (-\sqrt{K_y} \sin \Phi_y & \cos \Phi_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

EDGE FOCUSING MATRICES :

$$E1 = \begin{pmatrix} 1 & 0 & 0 \\ \pm \frac{1}{\rho} \tan(e1) & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$E2 = \begin{pmatrix} 1 & 0 & 0 \\ \pm \frac{1}{\rho} \tan(e2) & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

where + is used for x,
- is used for y.

Thus, the total magnet transfer matrix is given by ...

$$\begin{aligned} R_x &= E2_x T_x E1_x \\ R_y &= E2_y T_y E1_y \end{aligned}$$

ROTATED MAGNET :

The matrix for a rotated magnet is obtained by altering the coordinate system through the magnet via a rotational transformation. To perform this calculation, 7x7 matrices are used. In the 7x7 mode, a rotation of the coordinates of θ degrees about the s-axis is represented by

$$R_{\theta} = \begin{pmatrix} \cos\theta & 0 & \sin\theta & 0 & 0 & 0 & 0 \\ 0 & \cos\theta & 0 & \sin\theta & 0 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 & 0 & 0 & 0 \\ 0 & -\sin\theta & 0 & \cos\theta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} .$$

If R_{θ} represents a rotation of θ degrees and $R_{-\theta}$ represents a rotation of $-\theta$ degrees, then the transfer matrix of a rotated magnet is given by $M = R_{-\theta} M_0 R_{\theta}$, where M_0 is the original magnet matrix. Thus, a vertically bending magnet, for example, would be represented by the matrix

$$M_v = R_{-90} M_h R_{90} .$$

KICKER MAGNET :

The effect of the KICK command is to introduce a delta-function kick in the center of a previously defined zero-length drift region, or produce a field error in a previously defined magnet. If the element was defined as a zero-length drift, then the specified slope (x' or y') will be changed by the amount θ . The corresponding matrix (shown here for vertical deflection) would be

$$R = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \theta \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

If the original element was a magnet, then the KICK statement will alter the transport matrix to represent a magnet with a field error of dB/B. This is accomplished by changing the R17 and R27 elements of the transport matrix from 0 to $(L \cdot L/2 \rho)(dB/B)$ for R17 and from 0 to $(L/\rho)(dB/B)$ for R27.

On the other hand, if the element was previously defined as a drift region of length L, then the KICK statement turns the drift into a zero-gradient magnet, neglecting any edge focusing. The transport matrix for this new element is

$$\begin{array}{l}
 \text{horizontal} \\
 \text{kick:} \\
 R =
 \end{array}
 \begin{pmatrix}
 1 & L & 0 & 0 & 0 & L \theta/2 & -L \theta/2 \\
 0 & 1 & 0 & 0 & 0 & \theta & -\theta \\
 0 & 0 & 1 & L & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 -\theta & -L \theta/2 & 0 & 0 & 1 & L \theta^2/3 & -L \theta^2/6 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{pmatrix}$$

$$\begin{array}{l}
 \text{vertical} \\
 \text{kick:} \\
 R =
 \end{array}
 \begin{pmatrix}
 1 & L & 0 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & L & 0 & -L \theta/2 & L \theta/2 \\
 0 & 0 & 0 & 1 & 0 & -\theta & \theta \\
 0 & 0 & \theta & L \theta/2 & 1 & L \theta^2/3 & -L \theta^2/6 \\
 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1
 \end{pmatrix}$$

Non-linear Elements

Several types of accelerator elements cannot be represented as matrices which act on (x, x') , (y, y') in a linear fashion. SYNCH supports many of the common non-linear elements and the user may define his/her own subroutines to

deal with elements which SYNCH does not support (see MAP ROUTINES below).

SEXTUPOLE MAGNET :

The effect of the sextupole on reference ray V is to replace x' and y' with $x'-P$ and $y'+Q$, where

$$V = (x, x', y, y', ds, dp/p, 1)$$

and

$$P = \bar{S} (x^2 - y^2) / 2$$

$$Q = \bar{S}xy$$

$$\bar{S} = S / (1 + dp/p)$$

$$S = LB'' / B\rho .$$

The effect of the sextupole on rays near the reference ray, V , is given by the following 7x7 matrix:

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ (-\bar{S}x & 1 & \bar{S}y & 0 & 0 & P & P \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \bar{S}y & 0 & \bar{S}x & 1 & 0 & -Q & -Q \\ -P & 0 & Q & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} .$$

N-POLE MAGNET :

The effect of a thin-lens multipole kick of n-th order can be accommodated by SYNCH using the NPOL statement. The user may input the order of the multipole, n, the effective length of the element, L, and its Taylor series expansion coefficient, C_m , where $m = n - 1$. The magnetic field is then computed to be

$$B = B_x + i*B_y = (C_m/m!)z^{*m}, \quad \text{where } z = x + i*y.$$

or

$$B = B_y - i*B_x = (C_m/m!)z^{*m}, \quad \text{for skewed elements.}$$

A particle being tracked through such an element is then subject to this field over the distance L.

MAP ROUTINES :

In order to track particles through non-linear synchrotron elements not explicitly supported by SYNCH, the user may insert into SYNCH up to ten non-linear transformation subroutines. These subroutines are named MAP, MAP1, MAP2, ..., MAP9. These subroutines are merely dummy routines. All routines MAP through MAP9 may be replaced by the user's own subroutines. For more detailed information concerning the implementation of a user written MAP subroutine, see Section III-F, Non-linear Transformation Calculations.

B. Beamline

A beamline is defined as a particular sequence of accelerator elements (drift regions, magnets, etc.). In the SYNCH beamline statement (BML), the elements are listed in the order in which they are encountered by the beam. This order is the reverse of the order in which one would write the matrix multiplication on paper. The program, of course, sequentially multiplies the beamline matrices "on the left."

C. Betatron Functions

Given an accelerator made up of p periodic sections (superperiods), each section being composed of N elements, the 2×2 transfer matrices corresponding to one passage through one superperiod may be calculated (neglecting coupling) by:

$$\begin{aligned} M_0^x &= R_x(N)R_x(N-1)R_x(N-2) \dots R_x(2)R_x(1) , \\ M_0^y &= R_y(N)R_y(N-1)R_y(N-2) \dots R_y(2)R_y(1) . \end{aligned}$$

Here, the matrices are taken to act on the vectors (x, x') or (y, y') . These matrices are submatrices of the corresponding 3×3 matrices used earlier in this section. The elements of these submatrices may be parameterized as follows:

$$M = \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} \cos\mu + \alpha \sin\mu & \beta \sin\mu \\ -[(1 + \alpha^2)/\beta] \sin\mu & \cos\mu - \alpha \sin\mu \end{pmatrix}$$

where α , β , and μ are the betatron functions as defined by Courant and Snyder (see Reference 3). Thus, the values of the betatron functions at point 0 of the accelerator (i.e., at the entrance to element 1, exit of element N) are given by

$$\begin{aligned} \mu &= \arccos[(a + d)/2] \\ \beta &= b/\sin\mu \\ \alpha &= (a - d)/(2\sin\mu) . \end{aligned}$$

The parameter μ represents the phase advance of the betatron oscillation for one passage through the superperiod and can be denoted as $\mu = 2\pi\nu/p$, where p is the number of superperiods in the machine and ν is the tune of the accelerator. The amplitude of the betatron oscillation is proportional to $\sqrt{\beta}$. The slope of the function β is given by $d\beta/ds = -2\alpha$. The betatron oscillations which a particle undergoes while traversing the accelerator may be expressed as

$$\begin{aligned}x(s) &= A_x \sqrt{\beta_x(s)} \cos[\psi_x(s) + \phi_x] \\y(s) &= A_y \sqrt{\beta_y(s)} \cos[\psi_y(s) + \phi_y]\end{aligned}$$

where A_x , A_y , ϕ_x , and ϕ_y are arbitrary constants, and $\psi_x(s)$, $\psi_y(s)$ are the horizontal and vertical phase advances of the betatron oscillations from the arbitrary $s=0$ point in the accelerator.

To calculate the values of α , β , and ψ for all points in the machine, SYNCH computes the matrices representing a single pass through a superperiod starting at the end of each element. Once the single-pass transfer matrix, M_0 , for point 0 has been found, the values of $\beta(i)$ and $\alpha(i)$ at the end of the i -th element can be found from the matrix

$$M_i = R(i) M_{i-1} R^{-1}(i), \quad i = 1, 2, \dots, N.$$

Here, $R(i)$ is the transfer matrix for element i . The phase advance through the i -th element is given by

$$\mu(i) = \arctan\{R_{12}(i) / [\beta(i-1)R_{11}(i) - \alpha(i-1)R_{12}(i)]\}$$

Thus, the phase advance from point 0 to the end of the i -th element is given by

$$\psi(i) = \sum_{k=1}^i \mu(k).$$

The tunes of the accelerator are thus given by $\nu_x = p\psi_x/2\pi$, $\nu_y = p\psi_y/2\pi$.

The third row and column of each 3x3 transfer matrix is used to compute the closed orbit for an off-momentum particle. If M is the matrix for one complete revolution (i.e., through p superperiods), then the particle state vector $X_{eq} = (x, x', dp/p)$ representing a closed orbit must satisfy $X_{eq} = M X_{eq}$, from which

$$X_{eq} = (dp/p) \begin{bmatrix} (M_{13} + M^{-1}_{13}) / (2 - M_{11} - M_{22}) , \\ (M_{23} + M^{-1}_{23}) / (2 - M_{11} - M_{22}) , \\ 1 \qquad \qquad \qquad 1 \end{bmatrix} .$$

The dispersion function, $D(s)$, is given by

$$\begin{aligned} X_{eq} &= (dp/p) D_x & Y_{eq} &= (dp/p) D_y \\ X'_{eq} &= (dp/p) D'_x & Y'_{eq} &= (dp/p) D'_y \end{aligned}$$

Thus,

$$\begin{aligned} D &= (M_{13} + M^{-1}_{13}) / (2 - M_{11} - M_{22}) \\ D' &= (M_{23} + M^{-1}_{23}) / (2 - M_{11} - M_{22}) . \end{aligned}$$

Hence, the dispersion function can be calculated at the end of each element in the accelerator using the 3x3 matrices corresponding to the matrices M_i described earlier.

It must be stressed that in the above treatment of betatron function calculations, coupling between horizontal and vertical motion has been completely ignored. This is true for all SYNCH computations of β and α functions. However, accurate values for the machine tunes and dispersion functions may be obtained by using 7x7 matrices and performing a FXPT calculation. This method is further discussed under Closed Orbit Calculations later in this section.

Other machine parameters calculated by SYNCH are the transition energy and the natural chromaticities. The transition energy is that energy of the particle beam for which the period of revolution about the machine is independent of particle momentum. This energy is given by $E = \gamma_t E_0$, where E_0 is the rest mass of the orbiting particles. The parameter γ_t , called the transition gamma, is found from the relationship

$$\gamma_t = 1/\sqrt{\alpha},$$

where here, α represents the momentum compaction factor which is defined as

$$\alpha = (dC/C)/(dp/p) \quad , \quad C = \text{the machine's circumference.}$$

By tracking the vector $(Dx, D'x, Dy, D'y, 0, 1, 0)$ through a superperiod of ideal length S_0 , where D represents the momentum dispersion function, the change in path length for a particle with $dp/p = 1$ can be found. The resulting vector at the end of the superperiod will be $(Dx, D'x, Dy, D'y, -dL, 1, 0)$. Hence, the value of the transition gamma will be given by

$$\gamma_t = \sqrt{S_0 / dL} .$$

Since it could occur that $\alpha < 0$, the complex root of γ_t^2 is taken. This root is displayed at the end of the CYC output.

Chromaticity is defined as the change in the tune of the accelerator per unit change in dp/p . It can be shown (see Reference 3) that due to errors, k , in the field gradient function $K = B'/B\rho$ (i.e., $K = K_0 + k$), the change in tune can be found by

$$\Delta \nu = \frac{1}{4\pi} \int k \beta \, ds .$$

If $K = K_0(1 - dp/p)$, then the chromaticity can be written as

$$\xi_\alpha = - \frac{1}{4\pi} \int K \beta \, ds . \quad (\text{due to quadrupoles})$$

Likewise, an expression for the chromaticity due to sextupoles can be obtained:

$$\xi_s = \frac{1}{4\pi} \int K' Dx \beta ds . \quad (\text{due to sextupoles })$$

In a similar fashion, chromatic effects due to the edge focusing of the bending magnets can be taken into account. SYNCH attempts to calculate the total machine chromaticity due to these three effects and displays the result at the end of the CYC output. The chromaticities computed are due only to magnets which are explicitly listed in the beamline used in the CYC statement.

Another method for studying the machine's chromatic properties would be to perform a FXPT calculation for various values of dp/p and look at how the total machine tunes behave.

Tracking Betatron Functions Through a Beamline

If the 2x2 matrix M represents the passage through one periodic section of the accelerator, then M may be written as

$M = I \cos \mu + J \sin \mu$,
 where I is the identity matrix, and J is given by

$$J = \begin{pmatrix} \alpha & \beta \\ -\gamma & -\alpha \end{pmatrix}.$$

Let R represent the transfer matrix from point 0 to point 1, J_0 the above matrix with betatron functions evaluated at point 0, and J the above matrix with betatron functions evaluated at point 1. Then the two J-matrices are related by

$$J = R J_0 R^{-1}.$$

If R has the matrix elements

$$R = \begin{pmatrix} a & b \\ c & d \end{pmatrix},$$

then the betatron functions at point 1 in terms of the betatron functions at point 0 are given by the relations

$$\alpha = (ad + bc)\alpha_0 - ac\beta_0 - bd\gamma_0$$

$$\beta = a^2\beta_0 - 2ab\alpha_0 + b^2\gamma_0$$

$$\gamma = d^2\gamma_0 - 2cd\alpha_0 + c^2\beta_0$$

$$\psi = \psi_0 + \arctan\{ b / [a\beta_0 - b\alpha_0] \}$$

Using these relationships, the betatron functions can be tracked through a beamline given their initial values.

Dispersion functions are tracked through a beamline using the relations

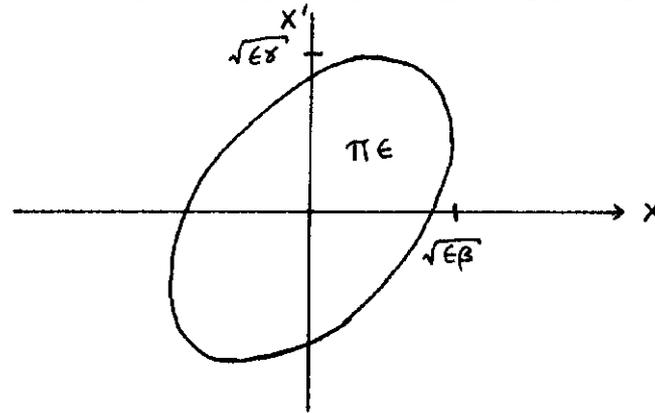
$$D = a D_o + b D'_o + e$$

$$D' = c D_o + d D'_o + f$$

where e and f are the 1-3 and 2-3 elements of the 3x3 transfer matrices.

D. Particle Beam Envelopes

Particle beam envelopes are calculated by SYNCH using betatron functions computed by the program and beam emittances input by the user. The beam emittance is defined as the area of the x' - x phase space ellipse which contains some certain fraction of the beam (99%, say). The user may input his/her own favorite value of the beam emittance for the machine in question. If β and γ are the familiar Courant-Snyder functions (see Reference 3) at a particular longitudinal location in the accelerator, then the phase space ellipse representing the beam will have the form shown in the figure below:



Here, the area is represented by $\pi\epsilon$, where ϵ is the beam emittance. The beam size, σ , as measured from the reference orbit, is thus given by

$$\sigma = \sqrt{\epsilon\beta}$$

If the beam has an average momentum error of dp/p , then this ellipse will be translated along the x -axis by an amount $D(dp/p)$ and translated along the x' -axis by an amount $D'(dp/p)$. D and D' are the dispersion function and its slope. The closed orbit may not lie on the reference orbit, if magnet misalignments and errors are taken into account, and so the beam ellipse may be displaced even further. The total beam envelope displayed by the CYAE command

is given either by

$$\begin{aligned}\sigma &= \sqrt{\epsilon\beta} \left[1 + \sqrt{\epsilon_{co}/\epsilon} \right] + |D dp/p| \\ \sigma' &= \sqrt{\epsilon\gamma} \left[1 + \sqrt{\epsilon_{co}/\epsilon} \right] + |D' dp/p|\end{aligned}$$

or by

$$\begin{aligned}\sigma &= \sqrt{\epsilon\beta + (D dp/p)^2} + \sqrt{\epsilon_{co}\beta} \\ \sigma' &= \sqrt{\epsilon\gamma + (D' dp/p)^2} + \sqrt{\epsilon_{co}\gamma}\end{aligned}$$

depending upon the option chosen. Here, ϵ_{co} is the equivalent emittance representing the closed orbit:

$$\epsilon_{co} = (X_{co})^2 / \beta .$$

The calculation is performed for both planes.

For electron machines, the CYEM statement may be used for determination of beam emittances, rf integrals, etc. (See CYEM in Section IV-B.) For these calculations, SYNCH follows the conventions of Morton (Reference 4). The user is referred to this paper for more information.

E. Closed Orbit Calculations

To calculate the closed orbit around an accelerator composed of various linear and non-linear elements, SYNCH begins by tracking an initial "first guess" particle state vector through one complete revolution. This initial vector will be denoted as V_0 , while the particle state vector after one revolution will be called V_1 . Next, all of the transfer matrices of the accelerator elements are linearized about this initial single-turn trajectory, generating new transfer matrices, R . For details of how the various element matrices are modified on this step, the reader is referred to Reference 2.

One may now track a particle vector, X , in the neighborhood of V (i.e., $X = V + Z$, where $Z \ll V$) around the machine. Let Z_0 be the initial difference vector ($X_0 - V_0$) which is input by the user. After one revolution, the vector Z_1 , the difference vector from V_1 , will be given by $Z_1 = T Z_0$, where T is the linearized single-turn transfer matrix found by $T = R_N R_{N-1} \dots R_2 R_1$.

For the trajectory to be a closed orbit,

$$X_1 = V_1 + Z_1 = X_0 = V_0 + Z_0$$

or,

$$X_0 = V_1 + T Z_0 = V_1 + T (X_0 - V_0)$$

or,

$$X_0 = -(T + I)^{-1} (V_1 - V_0) \equiv X_{eq}$$

where I is the identity matrix. Hence, the vector X_{eq} may be used as a new "best guess" (V_0) for the closed orbit, and the entire operation described above may be repeated n number of times until $|X_{eq}(n) - X_{eq}(n-1)| = Z_0(n)$ is less than some tolerance.

Once the final vector X_{eq} is found, the closed orbit throughout the machine is given by the last tracking of the particle. Also, the betatron functions about this closed orbit may be extracted from the matrices T and R_i in the

manner described earlier in this section. Again, the computations of β and α functions ignore coupling between horizontal and vertical motion.

Generalized momentum dispersion functions may also be extracted from T and R_i , since $T D = D$, where $D = (x, x', y, y', 0, 1, 1)$. This computation of D gives the change in the closed orbit from X_{eq} per unit dp/p .

The program also calculates eigenvalues and eigenvectors of the 4x4 submatrix M , the single-turn matrix which operates on (x, x', y, y') . The eigenvalues and eigenvectors will contain any coupling information and hence will provide the user with accurate values for the "eigen" tunes of the machine. The eigenvectors may be tracked around the accelerator as an output option of the FXPT statement.

F. Particle Tracking

The procedure for particle tracking with the SYNCH program is rather straight forward. The options allowed by the TRK command allows one to track a particle through a beamline consisting only of linear elements, or to track a particle through a beamline consisting of linear and non-linear elements. In the first case, the single-turn transfer matrix corresponding to the location of interest is computed and used in the calculation. In the second case, the trajectory of the particle through each individual sub-beamline (previously defined BML or MMM) and non-linear element obviously must be computed upon every passage.

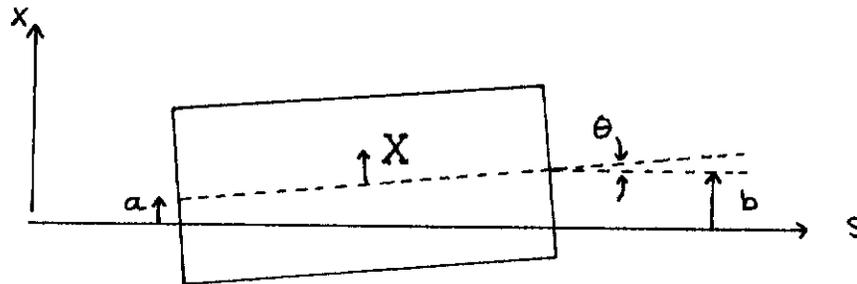
G. Magnet Misalignment Calculations

SYNCH provides two methods of generating magnet misalignments. One method employs the MAGS command which must be used in conjunction with the BMIS and EMIS commands. In this procedure, the user may specify transverse misalignments, but not rotations about the s-axis. If this method is used, then the orbit distortions brought about by the misalignments is shown in the CYC output in the columns usually containing the momentum dispersion functions. The second method employs the MOVE command. This command allows both x and y translations as well as rotational misalignments about the s-axis. In this procedure, the FXPT statement is used to compute the new closed orbit of the machine. Historically, the BMIS procedure was developed first for SYNCH and thus will be discussed first below. The second procedure, developed in more recent years, is perhaps the one more commonly used today.

The BMIS Method

One way to perform magnet misalignment calculations, is to think of the element matrices as acting on the particle state vector $(x, x', 1)$ [or $(y, y', 1)$ for the vertical plane]. The diagram below suggests that the coordinates relative to the element's centerline must be transformed in the following manner when entering or exiting a misaligned element:

entrance to	$x \Rightarrow x - a$
misaligned	
element :	$x' \Rightarrow x' - \theta$
exit from	$x \Rightarrow x + b$
misaligned	
element :	$x' \Rightarrow x' + \theta$



Thus, if the coordinates are transformed upon entrance to and exit from the magnet by the "shift" matrices

$$S_i = \begin{pmatrix} 1 & 0 & -a \\ 0 & 1 & -\theta \\ 0 & 0 & 1 \end{pmatrix}, \quad S_o = \begin{pmatrix} 1 & 0 & b \\ 0 & 1 & \theta \\ 0 & 0 & 1 \end{pmatrix},$$

then the transfer matrix for the misaligned magnet is given by $M = S_o M_o S_i$, where M_o is the original 3x3 matrix representing the aligned magnet with $M_o(1,3)$ and $M_o(2,3)$ set to zero.

In this manner, the same procedure used to calculate the closed orbit for an off-momentum particle in an aligned accelerator may be followed to calculate the closed orbit in the misaligned accelerator. Hence, CYC is used.

The MOVE Method

The other procedure for studying the effects of misaligned elements is to invoke the MOVE command and calculate the new closed orbit using FXPT. This method requires the use of 7x7 matrices and allows the study of horizontal and vertical coupling. Again, shift matrices are employed as well as rotation matrices to handle rotations about the s-axis. The appropriate 7x7 shift

matrices would be of the form

$$S_i = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -a_x \\ 0 & 1 & 0 & 0 & 0 & 0 & -\theta_x \\ 0 & 0 & 1 & 0 & 0 & 0 & -a_y \\ 0 & 0 & 0 & 1 & 0 & 0 & -\theta_y \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \text{etc.}$$

VI. REFERENCES

REFERENCES

- 1) A.A. Garren, A.S. Kenney, J.W. Eusebio; SYNCH -- A Computer System for Synchrotron Design and Orbit Analysis; LBL Internal Report UCID--10153; 1965. Original program documentation. Mostly hand-written notes.
- 2) A.A. Garren; SYNCH Closed Orbit and Related Calculations; PEP Technical Memo 49; 1977. A discussion of the methods used in SYNCH to calculate closed orbit information.
- 3) E.D. Courant, H.S. Snyder; Theory of the Alternating Gradient Synchrotron; Annals of Physics, Vol. 3, No. 1; 1958.
- 4) P.L. Morton; Effects of Transverse Coupling in the SLAC Storage Ring"; SLAC-PUB-863, 1971 Particle Accelerator Conference, Chicago, IL.
- 5) F. James, M. Roos; MINUIT -- Function Minimization and Error Analysis; CERN Computer Center Program Library, D506; 1967, revised 1983. CERN write-up referring to FORTRAN 77 and FORTRAN 4 implementation of MINUIT.
- 6) Proceedings from the 1981 - 1984 Summer Schools on High Energy Particle Accelerators. Provide many excellent articles, tutorials, problems in accelerator physics.
- 7) Theoretical Aspects of the Behaviour of Beams in Accelerators and Storage Rings; Proceedings of the First Course of the International School of Particle Accelerators, November, 1976; CERN, 1977.

VII. EXAMPLES OF SYNCH RUNS

EXAMPLES OF SYNCH RUNS

This section provides sample runs of SYNCH programs. The individual programs have been run on a DEC VAX11/780. Both the input file and the output file produced by the SYNCH program are presented.

- A. Standard Tevatron Cell
- B. Fitting of Tunes in a Phase Trombone
- C. Misalignment Calculation

```

TEVC  RUN          Tevatron Standard Cell
C
C          This particular SYNCH run calculates the betatron
C          functions through a Tevatron standard FODO cell.
C
C-----
C
C          Lengths in meters, field strengths in kG, kG/m, etc.
C
C          Magnetic Rigidity at 1 TeV ...
C
BRHO  =           33387.702
C
C          Bend field strength ...
C
BY    =           44.27664
C
C          Quadrupole gradients ...
C
GF    =           760.32056
GD    =          -760.32056
C
C          Magnet lengths ...
C
BL    =           6.1214
GL    =           1.67894
C
C
C          Drift Definitions
C
D      DRF          0.2794
DD     DRF          2.29616
DDD    DRF          0.4445
C
C          Magnet Definitions
C
B      MAG          BL          0.0          BRHO          BY          s
GF     MAG          GL          GF          BRHO
GD     MAG          GL          GD          BRHO
C
C          Standard Cell Beamline Definition
C
HC     BML          DD  B  D  B  D  B  D  B  D  B  DDD
C
CELL  BML          HC  GF  HC  GD
C
P          Computation of Betatron Functions
C
CYC    CELL
C
P          Define half-cell matrices in order to print out betatron functions
C          only at the ends of the quadrupoles...
C
MHCF   MMM          HC  GF
MHCD   MMM          HC  GD
C
C          Define a new "beamline" made up of the above two matrices.  Print
C          out betatron functions at the ends of the quadrupoles...
C

```

C
CL

BML
CYC

MHCF MHCD
CL

C
C

STOP
FIN

```

=====
*
*      This particular SYNCH run calculates the betatron
*      functions through a Tevatron standard FODO cell.
*
*-----
*
*      Lengths in meters, field strengths in kG, kG/m, etc.
*
*      Magnetic Rigidity at 1 TeV ...
*** BRHO =      0  0 // 33387.702
*
*      Bend field strength ...
*** BY   =      0  0 // 44.27664
*
*      Quadrupole gradients ...
*** QF   =      0  0 // 760.32056
*** QD   =      0  0 // -760.32056
*
*      Magnet lengths ...
*** BL   =      0  0 // 6.1214
*** QL   =      0  0 // 1.67894
*
*
*      Drift Definitions
*** O    DRF   0  0 // 0.2794
*** OO   DRF   0  0 // 2.29616
*** OOO  DRF   0  0 // 0.4445
*
*      Magnet Definitions
*** B    MAG   0  0 // BL      0.0      BRHO    BY      *
*** QF   MAG   0  0 // QL      QF      BRHO
*** QD   MAG   0  0 // QL      QD      BRHO
*
*      Standard Cell Beamline Definition
*** HC   BML   0  0 // OO  B  O  B  O  B  O  B  OOO
*
*** CELL BML   0  0 // HC  QF  HC  QD
*
=====

```

*
*
Computation of Betatron Functions

***	CYC	0	0	//	CELL								
POS	S(M)	NUX	NUY	BETAX(M)	BETAY(M)	XEG(M)	YEG(M)	ZEG(M)	ALPHAX	ALPHAY	DXEG	DYEG	
0		0.0000	0.00000	0.00000	29.04507	97.80421	2.26773	0.00000	0.00000	-0.58154	1.87141	0.04412	0.00000
1	00	2.2962	0.01201	0.00391	31.95858	89.45278	2.36903	0.00000	0.00000	-0.68733	1.76572	0.04412	0.00000
2	B	8.4176	0.03869	0.01627	42.09965	69.55521	2.66393	0.00000	0.02039	-0.96935	1.48467	0.05223	0.00000
3	0	8.6970	0.03974	0.01691	42.64492	68.72917	2.67852	0.00000	0.02039	-0.98222	1.47180	0.05223	0.00000
4	B	14.8184	0.05966	0.03317	56.39634	52.43252	3.02311	0.00000	0.04350	-1.26425	1.19036	0.06035	0.00000
5	0	15.0978	0.06044	0.03403	57.10640	51.77095	3.03997	0.00000	0.04350	-1.27712	1.17748	0.06035	0.00000
6	B	21.2192	0.07540	0.05575	74.46816	39.07971	3.43425	0.00000	0.06975	-1.55915	0.89571	0.06847	0.00000
7	0	21.4986	0.07600	0.05690	75.34301	38.58278	3.45338	0.00000	0.06975	-1.57202	0.88283	0.06847	0.00000
8	B	27.6200	0.08744	0.08594	96.31512	29.50044	3.89735	0.00000	0.09955	-1.85405	0.60083	0.07659	0.00000
9	000	28.0645	0.08817	0.08836	97.97247	28.97542	3.93140	0.00000	0.09955	-1.87453	0.58032	0.07659	0.00000
10	0F	29.7434	0.09087	0.09768	97.97247	28.97530	3.93310	0.00000	0.09955	1.87453	-0.58024	-0.07457	0.00000
11	00	32.0396	0.09477	0.10972	89.60695	31.88317	3.76189	0.00000	0.09955	1.76874	-0.68617	-0.07457	0.00000
12	B	38.1610	0.10711	0.13646	69.67928	42.01003	3.33029	0.00000	0.12830	1.48671	-0.96811	-0.06645	0.00000
13	0	38.4404	0.10776	0.13751	68.85210	42.55461	3.31172	0.00000	0.12830	1.47384	-0.98100	-0.06645	0.00000
14	B	44.5618	0.12398	0.15747	52.53477	56.28955	2.92982	0.00000	0.15360	1.19181	-1.26268	-0.05833	0.00000
15	0	44.8412	0.12483	0.15826	51.87238	56.99873	2.91352	0.00000	0.15360	1.17894	-1.27556	-0.05833	0.00000
16	B	50.9626	0.14652	0.17325	39.16540	74.33796	2.58131	0.00000	0.17587	0.89691	-1.55690	-0.05021	0.00000
17	0	51.2420	0.14766	0.17384	38.66780	75.21155	2.56728	0.00000	0.17587	0.88404	-1.56977	-0.05021	0.00000
18	B	57.3634	0.17663	0.18531	57.57116	96.15030	2.28476	0.00000	0.19553	0.60201	-1.85069	-0.04209	0.00000
19	000	57.8079	0.17904	0.18604	29.04507	97.80466	2.26605	0.00000	0.19553	0.58154	-1.87115	-0.04209	0.00000
20	0D	59.4868	0.18835	0.18874	29.04507	97.80421	2.26773	0.00000	0.19553	-0.58154	1.87141	0.04412	0.00000
CIRCUMFERENCE = 59.4868 M													
RADIUS = 9.4676 M													
(DS/S)/(DP/P) = 0.0032870													
THETX = 0.06494248 RAD													
THETY = 0.00000000 RAD													
TGAM=(17.44210, 0.00000)													
NUX = 0.18835													
NUY = 0.18874													
DNUX/(DP/P) = -0.21240													
DNUY/(DP/P) = -0.21294													
MAXIMA	---	BETX(10) =	97.97247	BETY(19) =	97.80466	XEG(10) =	3.93310	YEG(20) =	0.00000				
MINIMA	---	BETX(19) =	29.04507	BETY(10) =	28.97530	XEG(19) =	2.26605	YEG(20) =	0.00000				

```

*
*       Define half-cell matrices in order to print out betatron functions
*       only at the ends of the quadrupoles...
*
***  MHCF  MMM   0   0 //  HC   GF
***  MHCD  MMM   0   0 //  HC   GD
*
*       Define a new "beamline" made up of the above two matrices.  Print
*       out betatron functions at the ends of the quadrupoles...
*
***  CL    BML   0   0 //  MHCF MHCD
***  CYC   0   0 //  CL

```

POS	S(M)	NUX	NUY	BETAX(M)	BETAY(M)	XEQ(M)	YEQ(M)	ZEQ (M)	ALPHAX	ALPHAY	DXEQ	DYEQ
0	0.0000	0.00000	0.00000	29.04507	97.80421	2.26773	0.00000	0.00000	-0.58154	1.87141	0.04412	0.00000
1 MHCF	29.7434	0.09087	0.09768	97.97247	28.97530	3.93310	0.00000	0.09955	1.87453	-0.58024	-0.07457	0.00000
2 MHCD	59.4868	0.18835	0.18874	29.04507	97.80421	2.26773	0.00000	0.19553	-0.58154	1.87141	0.04412	0.00000

```

CIRCUMFERENCE = 59.4868 M          THETX = 0.06494248 RAD      NUX = 0.18835      DNUX/(DP/P) = 0.00000
RADIUS = 9.4676 M                THETY = 0.00000000 RAD      NUY = 0.18874      DNUY/(DP/P) = 0.00000
(DS/S)/(DP/P) = 0.0032870        TGAM=( 17.44210, 0.00000)

```

```

MAXIMA --- BETX( 1) = 97.97247    BETY( 2) = 97.80421    XEQ( 1) = 3.93310    YEQ( 2) = 0.00000
MINIMA --- BETX( 2) = 29.04507    BETY( 1) = 28.97530    XEQ( 2) = 2.26773    YEQ( 2) = 0.00000

```

```

*
*

```

SYNCH Example to Illustrate Fitting

In this example, the phase advance through a "phase trombone" is fitted to specified values. The phase trombone consists of 4 standard-sized cells with full bending, but the quadrupoles are powered separately. The idea is to match the Courant-Snyder parameters at the beginning and end of the trombone to those of the standard cell but to make the phase advance through the trombone any value one wishes.

Phase Trombone:



Adjust quad power supplies A,B,C,D,E to make phase advances from middle of QD to middle of QD desired values.

Lengths in meters, Fields in Tesla, Tesla/meter

```

BRHO = 66712. B
BO = 6.0
RHO = 11118. B
RHI = 1. / RHO

```

60-degree Standard Cells:

```

MUX = .166666667
MUY = .166666667

BR = 1.0
KF = .003
KD = -.003
LB = 16.6
LQ = 2.5

```

(Note: LQ is the half-quadrupole length)

Bending Magnets ...

```

B   MAG      LB      O.      BR      RHI

```

Standard Cell Quadrupole Magnets ...

```

SRC  SUB
QF   MAG      LQ      KF      BR
QD   MAG      LQ      KD      BR
CL   MMM      C

```

END

O DRF 0.5
OO DRF 2.5
OOC DRF 5.0

Beamlines

Standard Cell

.BB BML B O B O B O B
.OBO BML OOC BB OOC
BB BB
OBO MMM .OBO
FD BML QF .OBO QD
DF BML QD .OBO QF
C BML DF FD

Phase Trombone

.TR BML QD OBO QFA QFA OBO QDB QDB OBO QFC
QFC OBO QDD QDD OBO QFE QFE
OBO QDD QDD OBO QFC QFC
OBO QDB QDB OBO QFA QFA OBO QD

Fit the Cells ...

FITQ SRC CL KF KD 1 IMUX MUY
CELL CYC C

Fit Phase Trombone ...

Determine Desired Phase Advance Through
the Trombone...

MUXT = .95
MUYT = .95

Trombone Quadrupole Magnets

KAA = KF
KBB = KD
KCC = KF
KDD = KD
KEE = KF

TRMB SUB
QFA MAG LQ KAA BR
QDB MAG LQ KBB BR
QFC MAG LQ KCC BR
QDD MAG LQ KDD BR
QFE MAG LQ KEE BR

```

TTR  TRKB  0 30 TR
      0 30
      115.95190 0.      0.0086243 2.24716 0.0
      344.95041 0.      0.0028990 0.0 0.0

```

END

(NOTE: The last two lines in the TRKB statement could have been left blank and a CL could have been appropriately placed in line 1. The program would have then retrieved the values of the Courant-Snyder parameters from the matrix CL.)

```

SDLV  5  1 TRMB TTR      3000  -2      1
      AX      15      0.0      .0001
      AY      15      0.0      .0001
      DX      15      0.0      .0001
      GX      30      MUXT      .001
      QY      30      MUXT      .001
      KAA KBB KCC KDD KEE  1 -.004  +.004  .00001

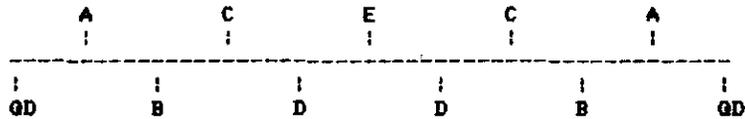
```

FIN
STOP

SYNCH Example to Illustrate Fitting

In this example, the phase advance through a "phase trombone" is fitted to specified values. The phase trombone consists of 4 standard-sized cells with full bending, but the quadrupoles are powered separately. The idea is to match the Courant-Snyder parameters at the beginning and end of the trombone to those of the standard cell but to make the phase advance through the trombone any value one wishes.

Phase Trombone:



Adjust quad power supplies A,B,C,D,E to make phase advances from middle of GD to middle of GD desired values.

Lengths in meters. Fields in Tesla. Tesla/meter

```

*** BRHO = 0 0 // 66712.8
*** BO   = 0 0 // 6.0
*** RHO  = 0 0 // 11118.8
*** RHI  = 0 0 // 1. / RHO
  
```

60-degree Standard Cells:

```

*** MUX = 0 0 // .166666667
*** MUY = 0 0 // .166666667
*** BR  = 0 0 // 1.0
*** KF  = 0 0 // .003
*** KD  = 0 0 // -.003
*** LB  = 0 0 // 16.6
*** LG  = 0 0 // 2.5
  
```

(Note: LG is the half-quadrupole length)

Bending Magnets ...

```

*** B   MAG 0 0 // LB 0. BR RHI
  
```

Standard Cell Quadrupole Magnets ...

```

*** SRC SUB 0 0 //
  
```

```
.....  
*** QF   MAC   0   0 //  LQ       KF       BR  
*** QD   MAC   0   0 //  LQ       KD       BR  
*** CL   MMM   0   0 //  C  
***     END   0   0 //  
.....
```

```
*  
*  
*  
*** O    DRF   0   0 //  0.5  
*** OO   DRF   0   0 //  2.5  
*** OOC  DRF   0   0 //  5.0  
*
```

* Beamlines

* Standard Cell

```

*** .BB BML 0 0 // B O B O B O B
*** .OBO BML 0 0 // OOC .BB OOC
*** BB MMM 0 0 // .BB
*** OBO MMM 0 0 // .OBO
*** FD BML 0 0 // GF .OBO QD
*** DF BML 0 0 // QD .OBO GF
*** C BML 0 0 // DF FD

```

* Phase Trombone

```

*** .TR BML 0 0 // QD OBO QFA QFA OBO QDB QDB OBO QFC
* // QFC OBO QDD QDD OBO QFE QFE
* // OBO QDD QDD OBO QFC QFC
* // OBO QDB QDB OBO QFA QFA OBO QD

```

Fit the Cells ...

*** *
 *
 *** FIT@ 0 0 // SRC CL KF KD 1 IMUX MUY

PARAMETER REPLACEMENTS MADE BY FITTING

1 OF KF = 0.002034033492 1 OF KD = -0.002034156078

*** CELL CYC 0 0 // C

POS	S(M)	NUX	NUY	BETAX(M)	BETAY(M)	XEQ(M)	YEQ(M)	ZEQ(M)	ALPHAX	ALPHAY	DXEQ	DYEQ
0	0.0000	0.00000	0.00000	115.95190	344.95041	2.24716	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
1 QD	2.5000	0.00342	0.00116	117.48644	340.60149	2.26146	0.00000	0.00000	-0.61641	1.73219	0.01145	0.00000
2 DDC	7.5000	0.01001	0.00356	123.94423	323.57321	2.31872	0.00000	0.00000	-0.67514	1.67346	0.01145	0.00000
3 B	24.1000	0.02946	0.01248	149.59536	271.25075	2.52121	0.00000	0.00361	-0.87010	1.47849	0.01294	0.00000
4 O	24.6000	0.02999	0.01277	150.46840	269.77519	2.52769	0.00000	0.00361	-0.87598	1.47262	0.01294	0.00000
5 B	41.2000	0.04595	0.02352	182.78703	224.12080	2.75496	0.00000	0.00755	-1.07093	1.27765	0.01444	0.00000
6 O	41.7000	0.04639	0.02388	183.86089	222.84610	2.76217	0.00000	0.00755	-1.07680	1.27177	0.01444	0.00000
7 B	58.3000	0.05945	0.03695	222.84674	183.85978	3.01422	0.00000	0.01186	-1.27174	1.07680	0.01593	0.00000
8 O	58.8000	0.05981	0.03738	224.12142	182.78592	3.02218	0.00000	0.01186	-1.27762	1.07093	0.01593	0.00000
9 B	75.4000	0.07056	0.05334	269.77413	150.46768	3.29901	0.00000	0.01657	-1.47255	0.87595	0.01742	0.00000
10 O	75.9000	0.07086	0.05387	271.24961	149.59466	3.30772	0.00000	0.01657	-1.47842	0.87008	0.01742	0.00000
11 B	92.5000	0.07978	0.07332	323.56875	123.94449	3.60932	0.00000	0.02174	-1.67334	0.67511	0.01891	0.00000
12 DDC	97.5000	0.08217	0.07992	340.59570	117.48703	3.70389	0.00000	0.02174	-1.73206	0.61638	0.01891	0.00000
13 QF	100.0000	0.08333	0.08333	344.94429	115.95257	3.72756	0.00000	0.02174	0.00000	0.00000	0.00000	0.00000
14 QF	102.5000	0.08449	0.08675	340.59570	117.48703	3.70389	0.00000	0.02174	1.73206	-0.61638	-0.01891	0.00000
15 DDC	107.5000	0.08689	0.09335	323.56875	123.94449	3.60932	0.00000	0.02174	1.67334	-0.67511	-0.01891	0.00000
16 B	124.1000	0.09581	0.11280	271.24961	149.59466	3.30772	0.00000	0.02690	1.47842	-0.87008	-0.01742	0.00000
17 O	124.6000	0.09611	0.11333	269.77413	150.46768	3.29901	0.00000	0.02690	1.47255	-0.87595	-0.01742	0.00000
18 B	141.2000	0.10686	0.12928	224.12142	182.78592	3.02218	0.00000	0.03161	1.27762	-1.07093	-0.01593	0.00000
19 O	141.7000	0.10721	0.12972	222.84674	183.85978	3.01422	0.00000	0.03161	1.27174	-1.07680	-0.01593	0.00000
20 B	158.3000	0.12028	0.14279	183.86089	222.84610	2.76217	0.00000	0.03592	1.07680	-1.27177	-0.01444	0.00000
21 O	158.8000	0.12072	0.14314	182.78703	224.12080	2.75496	0.00000	0.03592	1.07093	-1.27765	-0.01444	0.00000
22 B	175.4000	0.13667	0.15389	150.46840	269.77519	2.52769	0.00000	0.03986	0.87598	-1.47262	-0.01294	0.00000
23 O	175.9000	0.13720	0.15419	149.59536	271.25075	2.52121	0.00000	0.03986	0.87010	-1.47849	-0.01294	0.00000
24 B	192.5000	0.15665	0.16311	123.94423	323.57321	2.31872	0.00000	0.04347	0.67514	-1.67346	-0.01145	0.00000
25 DDC	197.5000	0.16325	0.16551	117.48644	340.60149	2.26146	0.00000	0.04347	0.61641	-1.73219	-0.01145	0.00000
26 QD	200.0000	0.16667	0.16667	115.95190	344.95041	2.24716	0.00000	0.04347	0.00000	0.00000	0.00000	0.00000

CIRCUMFERENCE = 200.0000 M
 RADIUS = 31.8310 M
 (DS/S)/(DP/P) = 0.0002174

THETX = 0.01492967 RAD NUX = 0.16667 DNUX/(DP/P) = -0.18367
 THETY = 0.00000000 RAD NUY = 0.16667 DNUY/(DP/P) = -0.18370
 TQAM=(67.82970, 0.00000)

MAXIMA --- BETX(13) = 344.94429 BETY(1) = 344.95041 XEQ(13) = 3.72756 YEQ(26) = 0.00000
 MINIMA --- BETX(1) = 115.95190 BETY(13) = 115.95257 XEQ(1) = 2.24716 YEQ(26) = 0.00000

```

*
*
*           Fit Phase Trombone ...
*
*           Determine Desired Phase Advance Through
*           the Trombone...
*
*** MUXT = 0 0 // .95
*** MUYT = 0 0 // .95
*
*           Trombone Quadrupole Magnets
*
*** KAA = 0 0 // KF
*** KBB = 0 0 // KD
*** KCC = 0 0 // KF
*** KDD = 0 0 // KD
*** KEE = 0 0 // KF
*
*
*** TRMB SUB 0 0 //
-----
*** GFA MAG 0 0 // LG      KAA      BR
*** GDB MAG 0 0 // LG      KBB      BR
*** GFC MAG 0 0 // LG      KCC      BR
*** GDD MAG 0 0 // LG      KDD      BR
*** GFE MAG 0 0 // LG      KEE      BR
*** TTR TRKB 0 30 // .TR      0 30
*
*           //      115.95190 0.      0.0086243 2.24716 0.0
*           //      344.95041 0.      0.0028990 0.0 0.0
***      END 0 0 //
-----

```

```

*
*           (NOTE: The last two lines in the TRKB statement
*           could have been left blank and a CL could
*           have been appropriately placed in line 1.
*           The program would have then retrieved the
*           values of the Courant-Snyder parameters
*           from the matrix CL.)
*
*
*

```

```

***      SOLV 5 1 // TRMB TTR      3000 -2      1
*
*           // AX      15      0.0      .0001
*           // AY      15      0.0      .0001
*           // DX      15      0.0      .0001
*           // GX      30      MUYT      .001
*           // GY      30      MUYT      .001
*           // KAA KBB KCC KDD KEE 1 -.004 +.004 .00001
-----

```

--- SOLV --- BETA-FUNCTION FITTING

FIRST CALL TO TRMB

PDS	S	GX	BX	AX	QX	X	DX	GY	BY	AY	QY	Y	DY
0	0.00000	0.00000	115.95190	0.00000	0.00862	2.24716	0.00000	0.00000	344.95041	0.00000	0.00290	0.00000	0.00000
1	GD	2.50000	0.00342	117.48644	-0.61642	0.01175	2.26146	0.01145	0.00116	340.60149	1.73219	0.01175	0.00000

2	OBO	97.50000	0.08217	340.59604	-1.73206	0.01174	3.70388	0.01891	0.07992	117.48734	0.61638	0.01175	0.00000	0.00000
3	GFA	100.00000	0.08333	344.94463	0.00000	0.00290	3.72755	0.00000	0.08333	115.95290	0.00000	0.00862	0.00000	0.00000
4	GFA	102.50000	0.08449	340.59605	1.73206	0.01174	3.70388	-0.01891	0.08675	117.48739	-0.61639	0.01175	0.00000	0.00000
5	OBO	197.50000	0.16325	117.48678	0.61642	0.01175	2.26146	-0.01145	0.16551	340.60433	-1.73221	0.01175	0.00000	0.00000
6	QDB	200.00000	0.16667	115.95224	0.00000	0.00862	2.24716	0.00000	0.16667	344.95331	0.00000	0.00290	0.00000	0.00000
7	QDB	202.50000	0.17008	117.48680	-0.61642	0.01175	2.26146	0.01145	0.16782	340.60438	1.73220	0.01175	0.00000	0.00000
8	OBO	297.50000	0.24884	340.59705	-1.73206	0.01174	3.70389	0.01891	0.24658	117.48835	0.61639	0.01175	0.00000	0.00000
9	GFC	300.00000	0.25000	344.94565	0.00000	0.00290	3.72756	0.00000	0.25000	115.95388	0.00000	0.00862	0.00000	0.00000
10	GFC	302.50000	0.25116	340.59705	1.73206	0.01174	3.70389	-0.01891	0.25342	117.48835	-0.61639	0.01175	0.00000	0.00000
11	OBO	397.50000	0.32992	117.48680	0.61642	0.01175	2.26147	-0.01145	0.33217	340.60438	-1.73220	0.01175	0.00000	0.00000
12	QDD	400.00000	0.33333	115.95224	0.00000	0.00862	2.24717	0.00000	0.33333	344.95331	0.00000	0.00290	0.00000	0.00000
13	QDD	402.50000	0.33675	117.48678	-0.61642	0.01175	2.26147	0.01145	0.33449	340.60433	1.73221	0.01175	0.00000	0.00000
14	OBO	497.50000	0.41551	340.59605	-1.73206	0.01174	3.70390	0.01891	0.41325	117.48739	0.61639	0.01175	0.00000	0.00000
15	GFE	500.00000	0.41667	344.94463	0.00000	0.00290	3.72757	0.00000	0.41666	115.95290	0.00000	0.00862	0.00000	0.00000
16	GFE	502.50000	0.41782	340.59604	1.73206	0.01174	3.70390	-0.01891	0.42008	117.48734	-0.61638	0.01175	0.00000	0.00000
17	OBO	597.50000	0.49658	117.48644	0.61642	0.01175	2.26147	-0.01145	0.49884	340.60149	-1.73219	0.01175	0.00000	0.00000
18	QDD	600.00000	0.50000	115.95190	0.00000	0.00862	2.24717	0.00000	0.50000	344.95041	0.00000	0.00290	0.00000	0.00000
19	QDD	602.50000	0.50342	117.48644	-0.61642	0.01175	2.26147	0.01145	0.50116	340.60149	1.73219	0.01175	0.00000	0.00000
20	OBO	697.50000	0.58217	340.59604	-1.73206	0.01174	3.70390	0.01891	0.57991	117.48734	0.61638	0.01175	0.00000	0.00000
21	GFC	700.00000	0.58333	344.94463	0.00000	0.00290	3.72757	0.00000	0.58333	115.95290	0.00000	0.00862	0.00000	0.00000
22	GFC	702.50000	0.58449	340.59605	1.73206	0.01174	3.70390	-0.01891	0.58675	117.48739	-0.61639	0.01175	0.00000	0.00000
23	OBO	797.50000	0.66325	117.48678	0.61642	0.01175	2.26147	-0.01145	0.66551	340.60433	-1.73221	0.01175	0.00000	0.00000
24	QDB	800.00000	0.66667	115.95224	0.00000	0.00862	2.24717	0.00000	0.66666	344.95331	0.00000	0.00290	0.00000	0.00000
25	QDB	802.50000	0.67008	117.48680	-0.61642	0.01175	2.26147	0.01145	0.66782	340.60438	1.73220	0.01175	0.00000	0.00000
26	OBO	897.50000	0.74884	340.59705	-1.73206	0.01174	3.70389	0.01891	0.74658	117.48835	0.61639	0.01175	0.00000	0.00000
27	GFA	900.00000	0.75000	344.94565	0.00000	0.00290	3.72756	0.00000	0.75000	115.95388	0.00000	0.00862	0.00000	0.00000
28	GFA	902.50000	0.75116	340.59705	1.73206	0.01174	3.70389	-0.01891	0.75341	117.48835	-0.61639	0.01175	0.00000	0.00000
29	OBO	997.50000	0.82992	117.48680	0.61642	0.01175	2.26146	-0.01145	0.83217	340.60438	-1.73220	0.01175	0.00000	0.00000
30	QD	1000.00000	0.83333	115.95224	0.00000	0.00862	2.24716	0.00000	0.83333	344.95331	0.00000	0.00290	0.00000	0.00000

CONSTRAINT	FUNCTION	POSITION	VALUE	FIT VALUE	ERROR	TOLERANCE
1	AX	15	0.0000017	0.0000000	0.0000017	0.0001000
2	AY	15	0.0000049	0.0000000	0.0000049	0.0001000
3	DX	15	0.0000000	0.0000000	0.0000000	0.0001000
4	QX	30	0.8333318	0.9500000	-0.1166682	0.0010000
5	QY	30	0.8333291	0.9500000	-0.1166709	0.0010000

FCN = SUM((ERROR/TOL.)**2) / 5 = 0.5444715E+04

 * D506 MINUITS *
 * VERSION 1.75 *
 * DATA BLOCK NO. 1 *

 1 KAA 0.002034 0.000010 -0.4000E-02 0.4000E-02
 2 KBB -0.002034 0.000010 -0.4000E-02 0.4000E-02
 3 KCC 0.002034 0.000010 -0.4000E-02 0.4000E-02
 4 KDD -0.002034 0.000010 -0.4000E-02 0.4000E-02
 5 KEE 0.002034 0.000010 -0.4000E-02 0.4000E-02

FCN VALUE	CALLS	TIME	EDM	INT. EXT.	PARAMETER	VALUE	ERROR	INTERN. VALUE	INT. STEP SIZE
0.5444715E+04	1	0.111	0.00E+00	1 1	CAA	1 0.20340E-02	0.10000E-04	0.53345E+00	0.29034E-02
				2 2	KBB	1 -0.20342E-02	0.10000E-04	-0.53349E+00	0.29035E-02
				3 3	KCC	1 0.20340E-02	0.10000E-04	0.53345E+00	0.29034E-02
				4 4	KDD	1 -0.20342E-02	0.10000E-04	-0.53349E+00	0.29035E-02
				5 5	KEE	1 0.20340E-02	0.10000E-04	0.53345E+00	0.29034E-02

 *** 1***PRINTOUT 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000

 *** 2***MINIMIZE 3000.00000 0.01000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000

 START SIMPLEX MINIMIZATION CONVERGENCE CRITERION -- ESTIMATED DISTANCE TO MINIMUM (EDM) .LT. 0.10E-01
 SIMPLEX MINIMIZATION HAS CONVERGED

FCN VALUE	CALLS	TIME	EDM	INT. EXT.	PARAMETER	VALUE	ERROR	INTERN. VALUE	INT. STEP SIZE
0.4379707E+04	466	0.585	0.51E-02	1 1	CAA	1 0.18750E-02	0.13976E-05	0.48788E+00	0.00000E+00
				2 2	KBB	1 -0.21361E-02	0.13874E-05	-0.56336E+00	0.24251E-04
				3 3	KCC	1 0.21808E-02	0.55557E-05	0.57664E+00	0.10509E-03
				4 4	KDD	1 -0.22100E-02	0.30757E-05	-0.58536E+00	0.59502E-04
				5 5	KEE	1 0.20167E-02	0.10851E-05	0.52841E+00	0.20432E-04

ERRORS CORRESPOND TO FUNCTION CHANGE OF 1.0000
 START MIGRAD MINIMIZATION. CONVERGENCE CRITERIA -- ESTIMATED DISTANCE TO MINIMUM (EDM) .LT. 0.10E-06
 OR EDM .LT. 0.10E-01 AND FRACTIONAL CHANGE IN VARIANCE MATRIX .LT. 0.40E-01
 COVARIANCE MATRIX IS NOT POSITIVE-DEFINITE.
 MIGRAD MINIMIZATION HAS CONVERGED

FCN VALUE	CALLS	TIME	EDM	INT. EXT.	PARAMETER	VALUE	ERROR	INTERN. VALUE	INT. STEP SIZE
0.9240698E-05	2012	2.252	0.54E-04	1 1	CAA	1 0.22388E-02	0.92511E-05	0.59403E+00	-0.55649E-05
				2 2	KBB	1 -0.21540E-02	0.44797E-05	-0.56866E+00	-0.28945E-07
				3 3	KCC	1 0.22722E-02	0.75090E-05	0.60413E+00	0.68678E-05
				4 4	KDD	1 -0.25637E-02	0.13381E-04	-0.69569E+00	-0.10980E-04
				5 5	KEE	1 0.24012E-02	0.83303E-05	0.64388E+00	0.10276E-04

ERRORS CORRESPOND TO FUNCTION CHANGE OF 1.0000

 *** 3***END RETURN 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000

LAST CALL TO TRMB

PDS	S	GX	BX	AX	OX	X	DX	GY	BY	AY	OY	Y	DY
0	0.00000	0.00000	115.95190	0.00000	0.00862	2.24716	0.00000	0.00000	344.75041	0.00000	0.00290	0.00000	0.00000
1	GD	2.50000	0.00342	117.48644	-0.61642	0.01175	2.26146	0.01145	0.00116	340.60149	1.73219	0.01175	0.00000
2	OBO	97.50000	0.08217	340.59604	-1.73206	0.01174	3.70388	0.01891	0.07992	117.48734	0.61638	0.01175	0.00000

3	GFA	100.00000	0.08333	344.50521	0.17569	0.00299	3.72518-0.00190	0.08333	116.10200	-0.05966	0.00864	0.00000	0.00000
4	GFA	102.50000	0.08450	338.85549	2.07365	0.01564	3.69441-0.02269	0.08674	118.08952	-0.73905	0.01309	0.00000	0.00000
5	DBO	197.50000	0.17841	86.01394	0.58779	0.01564	1.89355-0.01522	0.16110	376.67879	-1.98294	0.01309	0.00000	0.00000
6	QDB	200.00000	0.18310	84.30995	0.09687	0.01197	1.86816-0.00511	0.16215	381.53786	0.04804	0.00263	0.00000	0.00000
7	QDB	202.50000	0.18781	85.03654	-0.38881	0.01354	1.86795 0.00494	0.16320	376.20270	2.07644	0.01412	0.00000	0.00000
8	DBO	297.50000	0.29312	281.07394	-1.67470	0.01354	2.69182 0.01240	0.24088	109.10308	0.73513	0.01412	0.00000	0.00000
9	GFC	300.00000	0.29453	285.47988	-0.07933	0.00352	2.70366-0.00294	0.24457	107.03792	0.09484	0.00943	0.00000	0.00000
10	GFC	302.50000	0.29593	281.85971	1.52053	0.01175	2.67716-0.01824	0.24827	108.14569	-0.54004	0.01194	0.00000	0.00000
11	DBO	397.50000	0.39220	98.99968	0.40427	0.01175	1.29891-0.01078	0.33379	318.54539	-1.67469	0.01194	0.00000	0.00000
12	QDD	400.00000	0.39624	98.62527	-0.25370	0.01079	1.28232-0.00252	0.33503	321.82713	0.36901	0.00353	0.00000	0.00000
13	QDD	402.50000	0.40023	101.56392	-0.92803	0.01833	1.28630 0.00571	0.33628	314.89454	2.38920	0.02130	0.00000	0.00000
14	DBO	497.50000	0.47411	443.26076	-2.66871	0.01832	2.18282 0.01317	0.46743	53.20889	0.36539	0.02130	0.00000	0.00000
15	GFE	500.00000	0.47500	449.96612	0.00000	0.00222	2.19930 0.00000	0.47500	52.29996	0.00000	0.01912	0.00000	0.00000
16	GFE	502.50000	0.47589	443.26077	2.66871	0.01832	2.18282-0.01317	0.48256	53.20889	-0.36539	0.02130	0.00000	0.00000
17	DBO	597.50000	0.54977	101.56394	0.92803	0.01833	1.28618-0.00571	0.61372	314.89447	-2.38920	0.02130	0.00000	0.00000
18	QDD	600.00000	0.55376	98.62529	0.25370	0.01079	1.28220 0.00251	0.61497	321.82706	-0.36901	0.00353	0.00000	0.00000
19	QDD	602.50000	0.55780	98.99971	-0.40427	0.01175	1.29878 0.01077	0.61621	318.54532	1.67469	0.01194	0.00000	0.00000
20	DBO	697.50000	0.65407	281.85978	-1.52054	0.01175	2.67676 0.01824	0.70172	108.14568	0.54004	0.01194	0.00000	0.00000
21	GFC	700.00000	0.65547	285.47995	0.07933	0.00352	2.70326 0.00294	0.70543	107.03791	-0.09484	0.00943	0.00000	0.00000
22	GFC	702.50000	0.65688	281.07401	1.67470	0.01354	2.69142-0.01240	0.70912	109.10307	-0.73513	0.01412	0.00000	0.00000
23	DBO	797.50000	0.76219	85.03653	0.38881	0.01354	1.86772-0.00494	0.78680	376.20276	-2.07644	0.01412	0.00000	0.00000
24	QDB	800.00000	0.76690	84.30994	-0.09687	0.01197	1.86793 0.00511	0.78785	381.53793	-0.04804	0.00263	0.00000	0.00000
25	QDB	802.50000	0.77159	86.01393	-0.58779	0.01564	1.89332 0.01522	0.78889	376.67885	1.98294	0.01309	0.00000	0.00000
26	DBO	897.50000	0.86550	338.85538	-2.07365	0.01564	3.69410 0.02269	0.86326	118.08954	0.73905	0.01309	0.00000	0.00000
27	GFA	900.00000	0.86667	344.50510	-0.17569	0.00299	3.72487 0.00190	0.86667	116.10202	0.05966	0.00864	0.00000	0.00000
28	GFA	902.50000	0.86783	340.59593	1.73206	0.01174	3.70358-0.01891	0.87008	117.48737	-0.61638	0.01175	0.00000	0.00000
29	DBO	997.50000	0.94658	117.48643	0.61641	0.01175	2.26141-0.01145	0.94884	340.60151	-1.73219	0.01175	0.00000	0.00000
30	GD	1000.00000	0.95000	115.95189	0.00000	0.00862	2.24712 0.00000	0.95000	344.95044	0.00000	0.00290	0.00000	0.00000

CONSTRAINT	FUNCTION	POSITION	VALUE	FIT VALUE	ERROR	TOLERANCE
1	AX	15	-0.0000002	0.0000000	-0.0000002	0.0001000
2	AY	15	0.0000001	0.0000000	0.0000001	0.0001000
3	DX	15	-0.0000006	0.0000000	-0.0000006	0.0001000
4	GX	30	0.9500004	0.9500000	0.0000004	0.0010000
5	GY	30	0.9499982	0.9500000	-0.0000018	0.0010000

FCN = SUM((ERROR/TOL.)**2) / 5 = 0.9240698E-05

*
*
*

***	FIN	0	0	// CORE USE SUMMARY	MAXIMUM	USED	UNUSED
				STORE (ELEMENT STORAGE)	17280 (LMAX)	2837	14443
				INFF (ELEMENT DEFINITIONS)	720 (MAX)	157	563

END OF SYNCH RUN TRMB

MSMG RUN

Calculation of Closed Orbit for a Synchrotron with
a Misaligned Element.

The calculation will be performed using two different
methods -- 1) using BMIS and CYC
2) using MOVE and FXPT

Lengths in meters, field strengths in kG, kG/m, etc.

Magnetic Rigidity at 1 TeV ...

BRHO = 33387.702

Bend field strength ...

BY = 44.622553

Quadrupole gradients ...

GF = 760.32056

GD = -760.32056

Magnet lengths ...

BL = 6.1214

GL = 1.67894

Magnet definitions ...

Magnet Definitions

MDFN	SUB						
B	MAG	BL	0.0	BRHO	BY	*	
GF	MAG	GL	GF	BRHO			
GD	MAG	GL	GD	BRHO			
	END						

Drift Definitions

D	DRF	0.2794
DD	DRF	2.29616
DDD	DRF	0.4445

Standard Cell Beamline Definition

HC	BML	DD	B	D	B	D	B	D	B	DDD
CELL	BML	HC	GF	HC	GD					

----- METHOD 1:

Begin the Magnet Misalignment Mode...

BMIS

```

C      CALL 1      MDFN
C      ARC  MMM      76( CELL      )
C
C      Define half-cell matrices in order to print out betatron functions
C      only at the ends of the quadrupoles...
C
MHC  MMM      HC
MHCF MMM      HC  GF
MHCD MMM      HC  GD
C
C      Define a new "beamline" made up of the above two matrices.
C
C      BML      MHCF MHCD
C
C      Define the misaligned element and a corresponding
C      "misaligned" standard cell
C
MSG  MAGS  2      GF      0.0      0.0      -.010      -.010      0.0
C
MSC  BML      MHC  MSG  MHCD
C
C      Define a ring made up of the above standard cells with one
C      quadrupole being misaligned.
C
RNG  BML      10( C      ) MSC  9( C      ) ARC
C
C      Calculate the Closed Orbit using CYC.      (In BMIS mode, the
C      closed orbit will appear in the columns marked XEG, YEG, in
C      the CYC output.)      Note that in this CYC output, each line
C      will correspond to the end of a quadrupole.      Also, the "tunes"
C      at the end of the CYC output will be incorrect because the
C      phase advance through the matrix ARC is greater than 2pi.
C      See Section IV -- CYC.
C
CYC      RNG
C
EMIS
P
C      ----- METHOD 2:
C
C      CALL 1      MDFN
C
MHC  MMM      HC
MHCF MMM      HC  GF
MHCD MMM      HC  GD
ARC  MMM      76( CELL      )
MSC  BML      MHC  GF  MHCD
RNG  BML      10( C      ) MSC  9( C      ) ARC
C
C      Re-define a misaligned quadrupole magnet using the MOVE command...
C
C
VC  VEC  6      dx      dxp      dy      dyp      ds      dphi
GF  MOVE      GF  VC      0.0      0.0      -0.010      0.0      0.0      0.0

```



```

*
*          ----- METHOD 1:
*
*          Begin the Magnet Misalignment Mode...
***      BMIS
*
***      CALL 1 0 // MDFN
*
***      ARC  MMM  0 0 // 76( CELL  )
*
*          Define half-cell matrices in order to print out betatron functions
*          only at the ends of the quadrupoles...
***      MHC  MMM  0 0 // HC
***      MHCF  MMM  0 0 // HC  GF
***      MHCD  MMM  0 0 // HC  GD
*
*          Define a new "beamline" made up of the above two matrices.
***      C    BML  0 0 // MHCF MHCD
*
*          Define the misaligned element and a corresponding
*          "misaligned" standard cell
***      MSG  MAGS  2 0 // GF          0.0      0.0      -.010      -.010      0.0
***      MSC  BML  0 0 // MHC  MSG  MHCD
*
*          Define a ring made up of the above standard cells with one
*          quadrupole being misaligned.
***      RNG  BML  0 0 // 10( C      ) MSC  9( C      ) ARC
*
*          Calculate the Closed Orbit using CYC.      (In BMIS mode, the
*          closed orbit will appear in the columns marked XEQ, YEQ, in
*          the CYC output.)      Note that in this CYC output, each line
*          will correspond to the end of a quadrupole.      Also, the "tunes"
*          at the end of the CYC output will be incorrect because the
*          phase advance through the matrix ARC is greater than 2pi.
*          See Section IV -- CYC.
***      CYC  0 0 // RNG

```

POS	S	PSIX	BETAX	ALPHAX	XEQ	DXEQ	PSIY	BETAY	ALPHAY	YEQ	DYEQ
0	0.0000	0.00000	29.045075	-0.58154	0.000000	0.00000	0.00000	97.801567	1.87136	0.02419-0.00060	
1	MHCF 29.7434	0.09087	97.972458	1.87453	0.000000	0.00000	0.09769	28.974208	-0.58022	0.00663-0.00033	
2	MHCD 59.4868	0.18835	29.045075	-0.58154	0.000000	0.00000	0.18875	97.801567	1.87136	-0.00313-0.00022	
3	MHCF 89.2302	0.27922	97.972458	1.87453	0.000000	0.00000	0.28643	28.974208	-0.58022	-0.00997-0.00059	
4	MHCD 118.9736	0.37669	29.045075	-0.58154	0.000000	0.00000	0.37749	97.801567	1.87136	-0.02654 0.00043	
5	MHCF 148.7170	0.46756	97.972458	1.87453	0.000000	0.00000	0.47518	28.974208	-0.58022	-0.01412-0.00011	
6	MHCD 178.4604	0.56504	29.045075	-0.58154	0.000000	0.00000	0.56624	97.801567	1.87136	-0.01680 0.00054	
7	MHCF 208.2038	0.65591	97.972458	1.87453	0.000000	0.00000	0.66393	28.974208	-0.58022	-0.00063 0.00050	
8	MHCD 237.9472	0.75339	29.045075	-0.58154	0.000000	0.00000	0.75499	97.801567	1.87136	0.01393-0.00002	

9	MHCF	267.6906	0.84426	97.972458	1.87453	0.000000	0.00000	0.85267	28.974208	-0.58022	0.01365	0.00049
10	MHCD	297.4340	0.94173	29.045075	-0.58154	0.000000	0.00000	0.94374	97.801567	1.87136	0.02726	-0.00056
11	MHCF	327.1774	1.03260	97.972458	1.87453	0.000000	0.00000	1.04142	28.974208	-0.58022	0.01088	-0.00014
12	MHCD	356.9208	1.13008	29.045075	-0.58154	0.000000	0.00000	1.13248	97.801567	1.87136	0.00654	-0.00040
13	MHCF	386.6642	1.22095	97.972458	1.87453	0.000000	0.00000	1.23017	28.974208	-0.58022	-0.00548	-0.00059
14	MHCD	416.4076	1.31843	29.045075	-0.58154	0.000000	0.00000	1.32123	97.801567	1.87136	-0.02235	0.00026
15	MHCF	446.1510	1.40930	97.972458	1.87453	0.000000	0.00000	1.41891	28.974208	-0.58022	-0.01499	-0.00031
16	MHCD	475.8944	1.50677	29.045075	-0.58154	0.000000	0.00000	1.50998	97.801567	1.87136	-0.02332	0.00060
17	MHCF	505.6378	1.59764	97.972458	1.87453	0.000000	0.00000	1.60766	28.974208	-0.58022	-0.00578	0.00036
18	MHCD	535.3812	1.69512	29.045075	-0.58154	0.000000	0.00000	1.69872	97.801567	1.87136	0.00484	0.00018
19	MHCF	565.1246	1.78599	97.972458	1.87453	0.000000	0.00000	1.79641	28.974208	-0.58022	0.01065	0.00058
20	MHCD	594.8680	1.88347	29.045075	-0.58154	0.000000	0.00000	1.88747	97.801567	1.87136	0.02695	-0.00046
21	MHC	622.9325	1.97164	97.972458	-1.87453	0.000000	0.00000	1.97583	28.974337	0.58030	0.01410	-0.00046
22	MSQ	624.6114	1.97434	97.972458	1.87453	0.000000	0.00000	1.98516	28.974208	-0.58022	0.01410	0.00046
23	MHCD	654.3548	2.07181	29.045075	-0.58154	0.000000	0.00000	2.07622	97.801567	1.87136	0.02685	-0.00058
24	MHCF	684.0982	2.16269	97.972458	1.87453	0.000000	0.00000	2.17390	28.974208	-0.58022	0.01002	-0.00018
25	MHCD	713.8416	2.26016	29.045075	-0.58154	0.000000	0.00000	2.26496	97.801567	1.87136	0.00438	-0.00036
26	MHCF	743.5850	2.35103	97.972458	1.87453	0.000000	0.00000	2.36265	28.974208	-0.58022	-0.00658	-0.00060
27	MHCD	773.3284	2.44851	29.045075	-0.58154	0.000000	0.00000	2.45371	97.801567	1.87136	-0.02357	0.00031
28	MHCF	803.0718	2.53938	97.972458	1.87453	0.000000	0.00000	2.55140	28.974208	-0.58022	-0.01496	-0.00026
29	MHCD	832.8152	2.63685	29.045075	-0.58154	0.000000	0.00000	2.64246	97.801567	1.87136	-0.02207	0.00059
30	MHCF	862.5586	2.72773	97.972458	1.87453	0.000000	0.00000	2.74014	28.974208	-0.58022	-0.00465	0.00040
31	MHCD	892.3020	2.82520	29.045075	-0.58154	0.000000	0.00000	2.83121	97.801567	1.87136	0.00699	0.00014
32	MHCF	922.0454	2.91607	97.972458	1.87453	0.000000	0.00000	2.92889	28.974208	-0.58022	0.01146	0.00056
33	MHCD	951.7888	3.01355	29.045075	-0.58154	0.000000	0.00000	3.01995	97.801567	1.87136	0.02732	-0.00049
34	MHCF	981.5322	3.10442	97.972458	1.87453	0.000000	0.00000	3.11764	28.974208	-0.58022	0.01326	0.00002
35	MHCD	1011.2756	3.20190	29.045075	-0.58154	0.000000	0.00000	3.20870	97.801567	1.87136	0.01352	-0.00050
36	MHCF	1041.0190	3.29277	97.972458	1.87453	0.000000	0.00000	3.30638	28.974208	-0.58022	-0.00151	-0.00055
37	MHCD	1070.7624	3.39024	29.045075	-0.58154	0.000000	0.00000	3.39745	97.801567	1.87136	-0.01717	0.00011
38	MHCF	1100.5058	3.48111	97.972458	1.87453	0.000000	0.00000	3.49513	28.974208	-0.58022	-0.01439	-0.00043
39	MHCD	1130.2492	3.57859	29.045075	-0.58154	0.000000	0.00000	3.58619	97.801567	1.87136	-0.02641	0.00059
40	MHCF	1159.9926	3.66946	97.972458	1.87453	0.000000	0.00000	3.68388	28.974208	-0.58022	-0.00930	0.00022
41	MHCD	1189.7360	3.76694	29.045075	-0.58154	0.000000	0.00000	3.77494	97.801567	1.87136	-0.00266	0.00033
42	ARC	5710.7328	4.08129	29.045075	-0.58154	0.000000	0.00000	4.11972	97.801567	1.87136	0.02419	-0.00060

POS S PSIX BETAX ALPHAX XEQ DXEG PSIY BETAY ALPHAY YEG DYEG
R= 908.89135 THETA= 6.28318525 QX= 4.08129 QY= 4.11972

MAXIMA BETAX(38)= 97.97246 XEQ(42)= 0.00000 BETAY(42)= 97.80157 YEG(33)= 0.02732
MINIMA BETAX(2)= 29.04507 XEQ(42)= 0.00000 BETAY(3)= 28.97421 YEG(7)= -0.00063
CONTRIBUTIONS TO CHROMATICITY (DNU/(DP/P)) FROM MAGNETS EXPLICITLY IN CHR X = 0.0000, CHR Y = 0.0000

*
EMIS

*
*
*
*

----- METHOD 2:

*** CALL 1 0 // MDFN

*
*** MHC MMM 0 0 // HC
*** MHCF MMM 0 0 // HC GF
*** MHCD MMM 0 0 // HC GD
*** ARC MMM 0 0 // 76(CELL)
*** MSC BML 0 0 // MHC GFM MHCD
*** RNG BML 0 0 // 10(C) MSC 9(C) ARC

* Re-define a misaligned quadrupole magnet using the MOVE command...

*
* dx dyp dy dyp ds dphi
*** VC VEC 6 0 // 0.0 0.0 -0.010 0.0 0.0 0.0
*** GFM MOVE 0 0 // GF VC

* Look at the magnet matrices ...

*** WMA 2 0 // GF GFM

TRANSFER MATRICES

R(I, J)

ELEMENT	X	DX/DS	Y	DY/DS	-DS	DP/P	I
GF	0.96807535	1.66103516	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
	-0.03782588	0.96807535	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
	0.00000000	0.00000000	1.03226803	1.69696015	0.00000000	0.00000000	0.00000000
	0.00000000	0.00000000	0.03864398	1.03226803	0.00000000	0.00000000	0.00000000
	0.00000000	0.00000000	0.00000000	0.00000000	1.00000000	0.00000000	0.00000000
	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	1.00000000	0.00000000
	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	1.00000000

LENGTH = 1.67894000 THETA = 0.00000000

GFM	0.96807535	1.66103516	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
	-0.03782588	0.96807535	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
	0.00000000	0.00000000	1.03226803	1.69696015	0.00000000	0.00000000	0.0032268
	0.00000000	0.00000000	0.03864398	1.03226803	0.00000000	0.00000000	0.00038644
	0.00000000	0.00000000	0.00000000	0.00000000	1.00000000	0.00000000	0.00000000
	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	1.00000000	0.00000000
	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	1.00000000

LENGTH = 1.67894000 THETA = 0.00000000

*
*
*
*
*
*

Now the closed orbit may be calculated using FXPT...

Provide initial guess of closed orbit for the FXPT routine:

*** * PV PVEC 0 (x , x ' , y , y ' , ds , dp/p)
0 // 0 . 0 . 0 . 0 . 0 . 0 .

*** *
S FXPT 2 1 // PV RNC 1 0 10 0
* //

CALCULATION OF THE EQUILIBRIUM ORBIT AND BETATRON FUNCTIONS OF S

INITIAL REFERENCE RAY DEFINED BY PV

X = 0.00000000 DX = 0.00000000 Y = 0.00000000 DY = 0.00000000 DS = 0.00000000 DP/P = 0.00000000 1.00000000
 ITERATION = 1 X0= 0.00000000 DX0= 0.00000000 Y0= 0.02419259 DY0= -0.00059762
 ITERATION = 2 X0= 0.00000000 DX0= 0.00000000 Y0= 0.02419259 DY0= -0.00059762
 2

7X7 MATRIX FOR S

0.58808639	14.19863142	0.00000000	0.00000000	0.00000000	0.31012632	0.00000000
-0.02252253	1.15665138	0.00000000	0.00000000	0.00000000	0.04450932	0.00000000
0.00000000	0.00000000	2.00878975	66.82241806	0.00000000	-0.03722790	0.01552896
0.00000000	0.00000000	-0.03145116	-0.54841109	0.00000000	0.00039429	-0.00016447
-0.03316015	-0.27326333	0.00000000	0.00000000	1.00000000	-18.97781347	0.00000000
0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	1.00000000	0.00000000
0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	1.00000000

EIGENVALUES OF THE 4X4 SUBMATRIX

OX... LMD1 = (0.87236889 0.48884816), C(1) = 1.00000000, MU(1) = 0.51076891 RAD, G(1) = 0.08129140
 LMD1 = (0.87236889 -0.48884816), C(2) = 1.00000000, MU(2) = -0.51076891 RAD, G(2) = 0.91870860
 OY... LMD3 = (0.73018933 0.68324486), C(3) = 1.00000000, MU(3) = 0.75219731 RAD, G(3) = 0.11971592
 LMD3 = (0.73018933 -0.68324486), C(4) = 1.00000000, MU(4) = -0.75219731 RAD, G(4) = 0.88028408

EIGENVALUE = (0.87236889, 0.48884816), EIGENVECTOR = (5.38934827, 0.00000000)
 (0.10790458, 0.18555119)
 (0.00000000, 0.00000000)
 (0.00000000, 0.00000000)

EIGENVALUE = (0.87236889, -0.48884816), EIGENVECTOR = (5.38934827, 0.00000000)
 (0.10790458, -0.18555119)
 (0.00000000, 0.00000000)
 (0.00000000, 0.00000000)

EIGENVALUE = (0.73018933, 0.68324486), EIGENVECTOR = (0.00000000, 0.00000000)
 (0.00000000, 0.00000000)
 (9.88946749, 0.00000000)
 (-0.18922807, 0.10111768)

EIGENVALUE = (0.73018933, -0.68324486), EIGENVECTOR = (0.00000000, 0.00000000)
 (0.00000000, 0.00000000)
 (9.88946749, 0.00000000)
 (-0.18922807, -0.10111768)

	X	DX	Y	DY	DS	DP/P
EQ ORBIT	0.00000000	0.00000000	0.02419259	-0.00059762	0.00000000	1.00000000
ET ORBIT	2.28545231	0.04446080	-0.05799741	0.00143268	0.00000000	0.00000000

EIGENVECTORS 1 AND 3 IN POLAR COORDINATES

POS		X1 X3		DX1 DX3		Y1 Y3		DY1 DY3
0		5.389348	0.000000	0.214645	1.044064	0.000000	0.000000	0.000000
		0.000000	0.000000	0.000000	0.000000	9.889467	0.000000	0.214551 2.650829

BETATRON FUNCTIONS OF S

POS	S (M)	QX	QY	BX (M)	BY (M)	AX	AY	EX (M)	EXP	EY (M)	EXP	XCO (MM)	DXCO (MR)	YCO (MM)	DYCO (MR)
0	0.00	0.00	0.00	29.05	97.80	-0.58	1.87	2.28545	0.04446	-0.057997	0.001433	0.0000	0.0000	24.1926	-0.5976
1	MHCF 29.74	0.09	0.10	97.97	28.97	1.87	-0.58	3.96383	-0.07515	-0.015905	0.000794	0.0000	0.0000	6.6343	-0.3312
2	MHCD 59.49	0.19	0.19	29.05	97.80	-0.58	1.87	2.28545	0.04446	0.007499	0.000527	0.0000	0.0000	-3.1283	-0.2200
3	MHCF 89.23	0.28	0.29	97.97	28.97	1.87	-0.58	3.96383	-0.07515	0.023906	0.001405	0.0000	0.0000	-9.9718	-0.5861
4	MHCD 118.97	0.38	0.38	29.05	97.80	-0.58	1.87	2.28545	0.04446	0.063628	-0.001037	0.0000	0.0000	-26.5415	0.4324
5	MHCF 148.72	0.47	0.48	97.97	28.97	1.87	-0.58	3.96383	-0.07515	0.033855	0.000261	0.0000	0.0000	-14.1218	-0.1089
6	MHCD 178.46	0.57	0.57	29.05	97.80	-0.58	1.87	2.28545	0.04446	0.040277	-0.001306	0.0000	0.0000	-16.8008	0.5447
7	MHCF 208.20	0.66	0.66	97.97	28.97	1.87	-0.58	3.96383	-0.07515	0.001515	-0.001209	0.0000	0.0000	-0.6318	0.5044
8	MHCD 237.95	0.75	0.75	29.05	97.80	-0.58	1.87	2.28545	0.04446	-0.033386	0.000056	0.0000	0.0000	13.9263	-0.0234
9	MHCF 267.69	0.84	0.85	97.97	28.97	1.87	-0.58	3.96383	-0.07515	-0.032717	-0.001169	0.0000	0.0000	13.6475	0.4876
10	MHCD 297.43	0.94	0.94	29.05	97.80	-0.58	1.87	2.28545	0.04446	-0.065345	0.001348	0.0000	0.0000	27.2576	-0.5623
11	MHCF 327.18	1.03	1.04	97.97	28.97	1.87	-0.58	3.96383	-0.07515	-0.026081	0.000331	0.0000	0.0000	10.8792	-0.1382
12	MHCD 356.92	1.13	1.13	29.05	97.80	-0.58	1.87	2.28545	0.04446	-0.015680	0.000956	0.0000	0.0000	6.5405	-0.3988
13	MHCF 386.66	1.22	1.23	97.97	28.97	1.87	-0.58	3.96383	-0.07515	0.013134	0.001418	0.0000	0.0000	-5.4787	-0.5914
14	MHCD 416.41	1.32	1.32	29.05	97.80	-0.58	1.87	2.28545	0.04446	0.053572	-0.000630	0.0000	0.0000	-22.3466	0.2629
15	MHCF 446.15	1.41	1.42	97.97	28.97	1.87	-0.58	3.96383	-0.07515	0.035943	0.000733	0.0000	0.0000	-14.9929	-0.3058
16	MHCD 475.89	1.51	1.51	29.05	97.80	-0.58	1.87	2.28545	0.04446	0.055905	-0.001429	0.0000	0.0000	-23.3198	0.5961
17	MHCF 505.64	1.60	1.61	97.97	28.97	1.87	-0.58	3.96383	-0.07515	0.013854	-0.000867	0.0000	0.0000	-5.7790	0.3618
18	MHCD 535.38	1.70	1.70	29.05	97.80	-0.58	1.87	2.28545	0.04446	-0.011595	-0.000443	0.0000	0.0000	4.8366	0.1847
19	MHCF 565.12	1.79	1.80	97.97	28.97	1.87	-0.58	3.96383	-0.07515	-0.025540	-0.001384	0.0000	0.0000	10.6537	0.5775
20	MHCD 594.87	1.88	1.89	29.05	97.80	-0.58	1.87	2.28545	0.04446	-0.064611	0.001097	0.0000	0.0000	26.9514	-0.4574
21	MHCF 624.93	1.97	1.98	97.97	28.97	-1.87	0.58	3.96211	0.07719	-0.033805	0.001099	0.0000	0.0000	14.1012	-0.4583
22	QFM 624.61	1.97	1.99	97.97	28.97	1.87	-0.58	3.96383	-0.07515	-0.033805	-0.001099	0.0000	0.0000	14.1012	0.4583
23	MHCD 654.35	2.07	2.08	29.05	97.80	-0.58	1.87	2.28545	0.04446	-0.064370	0.001383	0.0000	0.0000	26.8507	-0.5767
24	MHCF 684.10	2.16	2.17	97.97	28.97	1.87	-0.58	3.96383	-0.07515	-0.024015	0.000442	0.0000	0.0000	10.0175	-0.1844
25	MHCD 713.84	2.26	2.26	29.05	97.80	-0.58	1.87	2.28545	0.04446	-0.010489	0.000867	0.0000	0.0000	4.3753	-0.3618
26	MHCF 743.58	2.35	2.36	97.97	28.97	1.87	-0.58	3.96383	-0.07515	0.015773	0.001431	0.0000	0.0000	-6.5794	-0.5968
27	MHCD 773.33	2.45	2.45	29.05	97.80	-0.58	1.87	2.28545	0.04446	0.056494	-0.000731	0.0000	0.0000	-23.5655	0.3050
28	MHCF 803.07	2.54	2.55	97.97	28.97	1.87	-0.58	3.96383	-0.07515	0.035858	0.000632	0.0000	0.0000	-14.9577	-0.2636
29	MHCD 832.82	2.64	2.64	29.05	97.80	-0.58	1.87	2.28545	0.04446	0.052908	-0.001416	0.0000	0.0000	-22.0698	0.5909
30	MHCF 862.56	2.73	2.74	97.97	28.97	1.87	-0.58	3.96383	-0.07515	0.011152	-0.000956	0.0000	0.0000	-4.6518	0.3988
31	MHCD 892.30	2.83	2.83	29.05	97.80	-0.58	1.87	2.28545	0.04446	-0.016767	-0.000332	0.0000	0.0000	6.9940	0.1386
32	MHCF 922.05	2.92	2.93	97.97	28.97	1.87	-0.58	3.96383	-0.07515	-0.027485	-0.001350	0.0000	0.0000	11.4648	0.5631
33	MHCD 951.79	3.01	3.02	29.05	97.80	-0.58	1.87	2.28545	0.04446	-0.065498	0.001167	0.0000	0.0000	27.3214	-0.4868
34	MHCF 981.53	3.10	3.12	97.97	28.97	1.87	-0.58	3.96383	-0.07515	-0.031789	-0.000058	0.0000	0.0000	13.2604	0.0240
35	MHCD1011.28	3.20	3.21	29.05	97.80	-0.58	1.87	2.28545	0.04446	-0.032413	0.001209	0.0000	0.0000	13.5207	-0.5041
36	MHCF1041.02	3.29	3.31	97.97	28.97	1.87	-0.58	3.96383	-0.07515	0.003615	0.001307	0.0000	0.0000	-1.5081	-0.5451
37	MHCD1070.76	3.39	3.40	29.05	97.80	-0.58	1.87	2.28545	0.04446	0.041160	-0.000260	0.0000	0.0000	-17.1692	0.1083
38	MHCF1100.51	3.48	3.50	97.97	28.97	1.87	-0.58	3.96383	-0.07515	0.034504	0.001039	0.0000	0.0000	-14.3927	-0.4333
39	MHCD1130.25	3.58	3.59	29.05	97.80	-0.58	1.87	2.28545	0.04446	0.063319	-0.001403	0.0000	0.0000	-26.4124	0.5854

Status of VAX Conversion -- June, 1985

At the time of this writing, the VAX version of SYNCH is in the process of being upgraded to have the identical features of the current CDC version. The differences between the two versions as they stand today are described below.

The CDC version allows the names of statements and Symbolic Floating Point data to be expressed as five alphanumeric characters. At present, the VAX version will only support four-character names.

The following SYNCH commands do not exist in the current VAX version:

BVAL NPOL ORBC RAND

The following SYNCH commands have either different options or different formats in the current VAX version:

CYC:

The CYC statement does not support the chromaticity correction option in the current VAX version.