MSUCP-7

# MISTIC GENERAL ORBIT CODE

M.M. Gordon
H.G. Blosser

September 1961
Department of Physics
East Lansing, Michigan

Research Supported in part by U.S. Atomic Energy Commission Contract AT (11-1) - 872

Contents

# Introduction

A computer routine is described which integrates the relativistic equations of motion for particles moving in accelerator type orbits. The routine is designed for use in the study of sector focused cyclotrons; in addition it is adaptable to a considerable range of other orbit tracking problems. The coding is for computers of the basic Illiac type (1024 word fast memory; no auxiliary memory). The program requires 523 memory locations for storage of instructions and computed quantities leaving 501 locations for storage of a table of cosines (typically 25 locations required) and for storage of magnetic field information. The relatively large block of field information (typically 476 locations) allows a carefully detailed description of the field. At the expense of more awkward operation an additional 90 field storage locations can be provided by overwriting portions of the program as described in Appendix F.

While written specifically for the Illiac class of computers the routine could presumably be transcribed to other logically similar computers in a fairly direct way. Of particular interest in this regard would be computers which, like the Illiac, have relatively small memories, since (1) the number of instructions for this routine is considerably smaller than other comparable routines of which we are aware and (2) the storage of the field as tables of Fourier coefficients gives both compact storage and detailed representation of the field features of dominant importance in determining particle trajectories.

In order to use or transcribe this program a moderate familarity with programing procedures for Illiac type computers is necessary. The programing manual for the MISTIC[1].

---

[1]. MISTIC Programing Manual, 2nd Edition, Michigan State University Computer Laboratory Publication (Sept. 1959).

presents such a description in more than adequate detail. In particular, in order to use this program, a general familarity with computer operations (such as is presented in chapters 1, 2 and 3 of the MISTIC manual) and a knowledge of the Decimal Order Input routine (chapter 4 of the MISTIC manual) are adequate. In this report the actual MISTIC order pairs are presented in section VIII - with the exception of this section effort has been made to minimize specific reference to MISTIC language, hopefully thereby making the earlier sections more understandable for those not familiar with the MISTIC.

## I. Equations of Motion

The equations of motion are written in cylindrical coordinates as a set of linear first order coupled equations with the azimuthal angle $\theta$ as the independant variable. The equations of motion in this form are:

(1) $dz/d\theta = rp_z/Q$

(2) $dp_z/d\theta = zr\,\partial B/\partial r - (zp_r/Q)\,\partial B/\partial\theta$

(3) $dr/d\theta = rp_r/Q$

(4) $dp_r/d\theta = Q - rB + (zp_z/Q)\,\partial B/\partial\theta$

(5) $d\phi/d\theta = (\gamma r/Q) - 1$

where z, r, $p_z$ and $p_r$ are the axial and radial positions and momenta of the particle, Q is the azimuthal component of momentum ($Q^2 = \gamma^2 - p_r^2 - p_z^2 - 1$), $\gamma$ is the total energy of the particle in rest mass units, and $\phi$ is the difference in time for the actual particle to reach angle $\theta$ and that required for a fictitious particle moving with uniform angular velocity of $\omega_{rf}$ to reach the same angle, the time difference being measured in units[2] in which the r-f period is $2\pi$. When the "median plane overwrite" is employed, equations (1) and (2), the final term in equation (4), and the $p_z$ factor in Q are dropped.

The equations are exact for motion in the median plane and are accurate to first order terms in z for off median plane motion. The equations would be accurate thru second order terms in z if B in eq. 4 were replaced by $(B - z^2\nabla^2 B/2)$.

The median plane magnetic field is specified in the form:

$$(6)\quad B(r,\theta) = B_o(r) + \sum_{i=1}^{H} [H_i(r)\cos iN\theta + G_i(r)\sin iN\theta] + \sum_{i=1}^{h} [h_i(r)\cos i\theta + g_i(r)\sin i\theta].$$

---

[2] With this choice of units, $\phi$ is also the angular difference in phase between particle and r-f. Note that positive $\phi$ means particle takes longer to reach a given angle and hence is lagging behind r-f.

The $B_o(r)$ and the first sum in (6) are referred to as the "main field", the second sum is referred to as the "imperfection field". The functions $B_o(r)$, $H_1(r)$, etc. are specified by numerical tables giving the values of the function at a sequence of equally spaced radius points ($r_o$, $r_o + \Delta r$, $r_o + 2\Delta r$, ....) with the additional restriction that $r_o$ (the first r at which the functions are given) must be equal to $\eta\Delta r$ where $\eta$ is an integer. The quantities B and $\partial B/\partial\theta$ are determined by evaluating (6) and its $\theta$ derivitive equation at the four nearest radius points to the value desired and performing a four point central polynomial interpolation in these values. The $\partial B/\partial r$ is obtained by differentiation of the polynomial for B. If at any point the orbit lies so close to the edge of the given field information that the central interpolation process is invalid, the computer prints information as described in section VI and stops. The number of sectors, N, which appears in eq. (6) is internally set to be 3 ( to change N, see Appendix B).

The routine will simulate acceleration by increasing $\gamma$ at one or both of two arbitrarily specified azimuths by an amount:

(7) $\Delta\gamma = (1/2)\ (\Delta K/E_o)\ \cos\phi(\theta)$.

The cos $\phi$ in equation (7) is computed accurate to fourth order terms in $\phi$. With minor modification, as described in Appendix E, the routine will omit the cos $\phi$ factor from (7). With a different modification, also described in Appendix E, the energy gain can be distributed over the full circumference rather than being localized at particular azimuths.

The routine will track orbits in either direction with respect to the $\theta$ in eq. (6). Runs designated "forward" go in the negative $\theta$ direction corresponding to the direction of rotation of a positive particle in a positive magnetic field in a right handed coordinate system. For convenience in making phase plots, the signs of $p_r$ and $p_z$ in the equations

are reversed for "backward" runs. As a result of this, if the final point of a forward run is used as the initial point of a backward run, identically the same lines of output will be obtained (except for round off errors) but in inverse order. A backward run is in this sense, then, simply determining where a forward moving particle has come from.

## II. Units, Scaling, and Program Stops

In setting up data and parameters for the routine an appropriate set of units must be employed such that the numerical relationships indicated by equations (1) thru (7) are in correspondence with the equations of motion for the particle being considered. For the case of the f-f cyclotron it is convenient to numerically specify lengths in units of $c/\omega_{rf}$, momenta in units of $m_o c$, and magnetic fields in units of $4\omega_{rf}m_o/q$, where $\omega_{rf}$ is the frequency of the accelerating voltage, $m_o$ and $q$ are the rest mass and charge of the accelerated particle, and c is the velocity of light. The factor four is inserted in the field unit so that the B's, H's, G's, h's and g's describing the field will be numerically less than 1/2 for normal fields as is required by scaling used in the routine.

Units involving $\omega_{rf}$ are usually internally computed by the routine in a manner such that $\omega_{rf}$ is taken to be $eB_o(0)/m_o$. In this case, the field unit becomes simply $4B_o(0)$ which is the scaling of the normally used B's, H's, G's, etc. obtained from the Mistic Pole Code.[3] With this scaling the field will of course necessarily be isochronous close to r=0 (to change to arbitrary $\omega_{rf}$, see Appendix J).

For input, the parameters $q$ and $m_o$ describing the accelerated particle are given as an integer $q'$ ($q = q'e$ where e is the magnitude of the electronic charge) and a fraction $m_o'$ (where $m_o'$ is the mass of the neutral atom in atomic mass units and must be $< 100$). The central magnetic field is input via the pair of quantities $B_o(0)$ in "volts"[4] and a "volts" to weber/$m^2$ conversion factor.

---

3. A Mistic Routine for Conversion of Rectangular Coordinate Field Data to Tables of Fourier Coefficients, H.G. Blosser, MSUCP report (in preperation).

4. An arbitrary unit related to the MSU field measuring equipment. $B_o(0)$ in volts is the 1st number on Pole Code output tapes.

The rest energy of the accelerated particle is internally computed from the charge and mass data ($E_o = m_o'c^2 - q'm_ec^2$). For particles with $E_o < 10^9$ ev, kinetic energies are input in units of $10^9$ ev, if $E_o < 10^{10}$ ev, kinetic energies are in units of $10^{10}$ ev, etc. The quantity $\gamma$ is internally computed by the routine from the kinetic and rest energies.

Tests on the numerical magnitude of various quantities are employed in the code to protect against the possibility of overflow. Principal among these are tests which require r, z, B, $\partial B/\partial r$ and $\partial B/\partial \theta$ to be at all times less than 1/2 in magnitude and a test which requires z to be always both less than r and less than 1/8 in magnitude. A list of program stops and their significance is given in Table I.

Table I. Orbit Code Stops. For each stop the decimal location is given, with "ℓ" or "r" signifying the left or right order, followed by the base 16 location in parenthesis, followed by the stop order and addresses.

"Load B Routine" Stops.

r 071(047) - 0F001: Some $B_o$, $H_i$, $G_i$, $h_i$ or $g_i$ on data tape has magnitude $\geq 1/2$.

ℓ 088(058) - 20058: All $B_o$'s, H's and G's are loaded. Ready for h's and g's if any

r 090(05K) - 243F7: All h's and g's (if any) loaded. Ready for main program.

Main Program Stops

r 001(001) - 243F7: DOI stop-occurs during loading of program tape when all sections except "Parameter Routine" have been loaded.

ℓ 984(3J8) - 2438J: Program tape loaded or run completed - ready for new run.

ℓ 1019(3LS) -SF000: Reader sum check fails (hexadecimal tapes only).

r 396(18N) - 4F037: Memory sum check fails.

ℓ 336(150) - 6601N: $|rp_r/Q| \geq 1$.

r 339(153) - 6601N: $|rp_z/Q| \geq 1$.

r 343(157) - 6600F: $K > E_o$.

ℓ 346(15K) - 6601N: $|(1/4)d\phi/d\theta| \geq 1$.

ℓ 353(161) - 66006: $|z| \geq |r|$

ℓ 355(163) - 66020: $|dp_z/d\theta| \geq 1$.

ℓ 362(16K) - 66020: $|zp_z(\partial B/\partial\theta)/4Q| \geq 1$

r 365(16J) - 66000: $|dp_r/d\theta| \geq 1$.

ℓ 368(170) - KK001: $|r| \geq 1/2$.

r 144(090) - SF001: $|\Sigma B - G_x \sin x\theta| \geq 1/2$.

r 147(093) - SF002: $|\Sigma B + H_x \cos x\theta| \geq 1/2$.

r 153(099) - SF003: $|\Sigma B_\theta + NH_x \sin x\theta| \geq 1/2$.

r 156(09N) - SF004: $|\Sigma B_\theta + NG_x \cos x\theta| \geq 1/2$.

r 059(03S) - 66002: $1+p_r^2+p_z^2 \geq \gamma^2$

## III. Operation of the Routine

Use of the routine begins with loading of magnetic field information, the numerical values being arranged on a data tape the format of which is described in Section IV. A special routine is employed ("Load B Routine") which, after the fields are loaded, is overwritten by the main program. After loading of the main program, parameters giving the initial values of the several variables are loaded as described in section V and the integration proceeds. A basic sixteen Runge-Kutta steps per sector are employed (to change this number see Appendix C.) and the quantities z, $p_z$, r, $p_r$, $\phi/4$, K, and $(1/64)\Sigma r$[5]. are printed out periodically, the number of R-K. steps between prints being given by one of the parameters. The run proceeds in this manner for a number of prints also specified by an input parameter. (Note: the code begins by printing out initial conditions and counts this as 1 print. To obtain new information a minimum of two prints must be requested.)

Immediately after the second and each subsequent print the code performs a sum check of memory locations 55 thru 985 (with locations involving variable addresses excepted) and stops if the sum is improper. On the first run after loading a set of fields, the sum will of necessity be improper and the code will stop after the second print. Restarting with a white switch start will set the sum comparison to the proper value after which no more such stops will normally occur unless fields or code are modified. If such a stop does occur, memory failure is indicated and the program and fields should be reloaded. For the use of maintenance personnel a ten digit sexadecimal word is punched when a sum

---

5. The quantity $(1/64)\Sigma r$ is a scaled (1/64) sum of the values of r on every previous R-K. step, from the beginning of the run. On long runs this quantity will periodically overflow and the values suddenly decrease by minus two. If the proper number of twos, as indicated by the successive jumps, is added to the value printed, the correct sum is obtained.

check stop occurs. This word should be zero; the extent to which it fails to be zero indicates the error in the sum and hence pin points the memory stage involved in the failure.

If several runs are to be made with the same magnetic field it is suggested that after an initial run has been made to set the sum check comparison, a sexadecimal tape of the program and fields be prepared using Mistic library routine X-16S. The fields and program can then be loaded with greatly enhanced speed on subsequent runs and in addition a double check on reader errors is provided. (Tapes produced by X-16S themselves contain a reader sum check; the sum check included in the program provides a second test).

When a given run terminates, the code returns to a position ready to load new parameters and stops. If parameters for a series of runs are on the same data tape the black switch can be placed on ignore and the routine will proceed automatically through the series (see section V for details and limitations on this procedure).

If no axial motion is included in a particular run the median plane overwrite should be employed and is loaded with the white switch prior to loading of the parameters. This overwrite reduces the running time by 40% and in addition prevents round off errors from building up spurious z motion.

## IV. Data Tape Format

The data tape consists of a set of numbers describing the arrangement and characteristics of the field and the accelerated particle, followed by the field information itself. In cases where the quantity of information on the field data tape exceeds the available memory space, the Load B Routine can be instructed to select an arbitrary section of the field thereby avoiding the necessity of having several tapes containing various sections of a given field. The arrangement of the data tape is as follows (in this and following sections the format of the Illiac Decimal Order Input routine is used):

| 00 18 K | Storage Directive |
|---|---|
| 00 F 00 a F: | a is number of entries to be loaded in memory for each of the functions $B_o$, $H_i$, $G_i$,..... |
| 00 F 00 H F: | H is number of harmonics in the main field to be loaded in memory. $H \geq 1$. |
| 00 F 00 h F: | h is number of harmonics in imperfection field to be loaded in memory. $h \geq 0$. |
| 00 dF 00 F: | $B_o$'s, H's, G's appear on data tape as sign plus d decimal digits. |
| 00 F 00 b F: | b is number of entries on the data tape for each of the functions $B_o$, $H_i$, $G_i$, ..... |
| 00 F 00 $\alpha$ F: | $\alpha$ is no. of entries on the data tape which are skipped for each of the functions $B_o$, $H_i$, $G_i$,..... before starting to load the specified a entries. ($b \geq \alpha + a$) |
| 00 F 00 $\beta$ F | $r_{min.} = \beta\Delta r$ is the smallest r value for which the $B_o$'s, H's and G's are loaded in the memory. Since $r_o = \eta\Delta r$ is first r value for which the functions are given on the data tape, $\beta = \alpha + \eta$. Also, always, $\beta$ must be integer greater than or equal to minus one. (Note: $\beta = -1$ is given in DOI format by LL 4095F LL 4095F. |

| | | |
|---|---|---|
| 00 33K | | Storage Directive |
| 00 F 00 | [ξ/10]J: | $B(\text{weber}/m^2) = \xi \cdot B(\text{volts})$ |
| 00 F 00 | $[B_o(0)/10]$J: | $B_o(0)$ is the measured central field in volts ($B_o(0)/10$ is first number on Pole Code output tapes). |
| 00 F 00 | [λ/4]J: | Flutter field (H's, G's, h's and g's but not $B_o$'s) is multiplied by λ. (λ/4 is first number on Isochronous Code output tapes). |
| 00 F 00 | [Δr]J: | Δr is the spacing of B's, H's, etc. in units of inches <u>in the model magnet</u>. |
| 00 F 00 | s J: | s is the scale factor of the model magnet i.e. s=1/6, 1/5, 8.75/64, etc. |
| 00 F 00 | q'F: | the charge of the accelerated particle is q=q'e. |
| 00 F 00 | $[m'_o/100]$J: | $m'_o$ is the mass in atomic mass units of the <u>neutral</u> atom from which the accelerated particle was produced. |
| 24 95 N | | Transfer to load B routine; can be 26 95N if $B_o$'s, H's, G's, etc. are on same tape. |

Field data (each quantity typed as sign plus d decimal digits):

$B_o(\eta\Delta r)$, $B_o([\eta+1]\,\Delta r),\ldots,B_o([\eta+a-1]\,\Delta r)$, $H_1([\eta\Delta r]\,\Delta r),\ldots$

$\ldots,H_1([\eta+a-1]\,\Delta r)$, $G_1(\eta\Delta r),\ldots,G_1([\eta+a-1]\,\Delta r)$, $H_2(\eta\Delta r)\ldots$

$\ldots,G_H([\eta+a-1]\,\Delta r)$.

Following loading of the main field the computer will stop. If the imperfection field is on a different tape this tape is placed under the reader and operation resumed with the black switch. The imperfection field is in the same format as the main field viz:

$h_1(\eta\Delta r)\ldots\ldots,h_1([\eta+a-1]\,\Delta r)$, $g_1(\eta\Delta r)\ldots,g_h([\eta+a-1]\,\Delta r)$.

Following loading of the imperfection field the computer again stops, after which restarting with the black switch will cause the main program to load (if no imperfection field is used, i.e. if h=0, two black switch starts are required to cause the main program to load.) The main program tape as mentioned previously is preset for 48 R-K steps per revolution integration and contains the required table of cosines for this number of R-K steps. This cosine table occupies 25 memory locations - hence in this case, unless the special procedures described in Appendix F have been employed, the total number of B's, H's, G's, h's, and g's loaded in the memory must not exceed 476, i.e., in terms of the parameters, $[2(H+h)+1]\,a \leq 476$.

V. Parameters:

Initial conditions and run parameters are stored at two places in the program the first being in locations 4 thru 16 and the second in locations 986 thru 998. The sequence 4 thru 16 are the actual initial conditions used by the code; these locations are also used for running storage of the various variables and hence are continually changing as the integration process proceeds. The sequence 986 thru 998 are employed as "master" parameters, i.e. at the end of any run (and initially after loading of the complete program tape) 986 thru 998 are copied into 4 thru 16. Table II is a list of various parameters including their location and significance. Also, for each parameter, the value preset in the program tape is included in parentheses.

To run a series of runs the sequence 986 thru 998 is set to the values which apply most often to the sequence; then for the individual runs only those parameters which differ from the master values need be set, the revised values being stored in the sequence 4 thru 16. Note that 986 thru 998 are not normally copied into 4 thru 16 until the end of a run; changes in 986 thru 998 will therefore normally not effect the first run following the change. (Changes in 986 thru 998 can be caused to effect the first run if desired, by using the white switch and DOI, as discussed below, to make the changes with the changes terminated by the special transfer directive 26977N. This termination, causes control to be transferred directly to the routine which copies 986 thru 998 into 4 thru 16; after doing this the computer will stop at the normal end of run stop, ready to pick up additional parameters, if any, in the normal manner.)

Two routines are included in the program for the loading of parameters. These routines can be employed in a variety of fashions to change either the master parameters or the

Table II: RUN PARAMETERS

| Run Location | Master Location | Parameter Function (Preset Value in Parenthesis) |
|---|---|---|
| 004 | 986 | $Z_o$=value of Z at initial $\theta$ in $c/\omega_{rf}$ units. (=$10^{-5}$) |
| 005 | 987 | $p_{z_o}$=value of $p_z$ at initial $\theta$ in $m_o c$ units. (=0) |
| 006 | 988 | $r_o$=value of r at initial $\theta$ in $c/\omega_{rf}$ units. (=.25) |
| 007 | 989 | $p_{r_o}$=value of $p_r$ at initial $\theta$ in $m_o c$ units. (=.085) |
| 008 | 990 | $\phi_o/4$=value of $\phi/4$ at initial $\theta$ in radians. (=0) |
| 009 | 991 | $K_o$=value of K at initial $\theta$. Units of K same as units of $E_o$ below. (=.028) |
| 010 | 992 | [1/64]$\Sigma$r - initial value irrelevant - automatically set to zero by code. |
| 011 | 993 | $\delta$K - K is increased by [$\delta$K/2] cos $\phi$ at gap crossing - units of $\delta$K same as $E_o$. (=.00028) |
| 012 | 994 | $\theta_1 x 2^{-18}$ where $\theta_1$ is integer giving no. of R-K steps at end of which gap one is crossed. Cell must be <u>exactly</u> $\theta_1 x 2^{-18}$ to give gap crossing; hence use DOI integer input to set. Examples: $\theta_1$=4→00 8F 00F, $\theta_1$=0→00 F 00 F, no gap crossing→00 F 00 xF where x[modulo $2^{20}$] ≠ 0. (00 F 00 1F) |
| 013 | 995 | Same as $\theta_1$ except gap two. |
| 014 | 996 | $E_o$-rest energy of particle - normally set by load B routine in units of $10^n$ ev where n is smallest integer which makes $E_o$<1. Examples: protons→$E_o$=.938---, deuterons→ $E_o$=.187---. Preset value may be overwritten if desired, with units arbitrary, except $E_o$<1. Note: see K and $\delta$K descriptions above. |
| 015 | 997 | $[-1+R_p x 2^{-39}]$ where $R_p$ is no. of R-K steps between prints of results. Note: since $R_p$ is integer, use DOI integer input to change; "minus one" implies 80 F 00 $R_p$F. ($R_p$=16) |
| 016 | 998 | $[-1+R_r x 2^{-39}]$ where $R_r$ is no. of prints per run. Value of $R_r$ in these locations is irrelevant when "New Parameter Routine" is in use since that routine independently sets $R_r$ in location 016 as last step before starting run. Note that 998 still is used to set -1 in 016. |

run parameters or both. The first of these routines is specially written to provide for the loading of fractions with minimum tape preparation. This routine also has a special termination feature as described further below which is used to select forward or backward integration of the orbits. The second routine is the standard Decimal Order Input which can of course be used to change the contents of any memory cell following conventional DOI procedure.

After loading of the program tape or after completing a run, the computer stops (if the black switch is not on ignore) on a 24 order in location 984 (3J8), this order being the normal end of run stop. At this point restarting with the white switch will transfer to the DOI whereas restarting with the black switch transfers to the "Special Input Routine". Since the Special Input Routine, as mentioned above, sets a number of orders in the code to provide for either forward or backward integration, it is necessary, if the DOI input is employed, to also go through the special routine even if the routine is used for nothing more than specifying the length of the run and whether it is forward or backward - this is accomplished by terminating the DOI portion of the parameter tape with a transfer to the special routine (26909N). Note that a number of the parameters, i.e. $\theta_1$, $\theta_2$, $R_p$, etc., are of integer type and, since the new parameter routine is designed only for the loading of fractions, it is always necessary when changing these quantities to employ the DOI. When using the DOI, prepare parameters in standard DOI format as described in the Mistic manual.[1].

The Special Input Routine differs from the DOI in that fifth hole characters on the input tape are employed as signals to denote the termination of numbers; the number of digits in any quantity can then be arbitrary. All quantities on the tape except the termination symbol, discussed

in the following paragraph, consist of address-fraction pairs, the address portion of the pair telling where the fraction is to be stored in the memory. In preparing the tape the address is typed first as a decimal integer (at least one and less than 13 digits interpreted modulo 4096) followed by some fifth hole character, this character denoting the end of the address. Next the fraction is typed with or without sign (if sign is omitted the fraction is interpreted as positive) and with optional number of digits (at least one and less than 13) and again with any fifth hole character denoting the termination of the fraction. Always, the fraction is interpreted as having decimal point immediately in front of the first digit regardless of the number of digits typed. The fifth hole character at the end of the fraction signals the routine to store the fraction in the previously loaded address and prepare for a new address-fraction pair. This process is continued for as many pairs as desired.

Finally, when all desired parameters have been set, the run is started by typing a decimal integer specifying the number of prints desired in the run, the integer being followed directly by either an "F" or a "K" depending on whether the run is to be in the Forward or backward direction. Whenever such a group of characters (i.e. one or more digits followed directly by an F or K) appears on the tape, the routine sets the necessary orders in the program, transfers to the proper section of the code, and proceeds with the run. A "4F" for example typed on the tape will cause the code to print four times (one of the times being initial conditions) integrating in the forward direction between prints. Similarly "26K" would give 26 prints with integration in the backward direction.

Table III shows three sample parameter tapes illustrating many of the above features. Each of the three parameter tapes in the figure will accomplish exactly the same set of 4 runs; the optional character of many features of the para-

Table III: Three Sets of Parameters (all accomplishing identical runs) Illustrating Several Format Possibilities.

Example A:

```
00 991K
00 F 00 0320 0000 0000 J
00 993K
00 F 00 0001 4000 0000 J
00 F 00 F
00 48F 00 F
00 997K
80 F 00 48F
26 977N
6 3 7 -064 12K
6 33 7 -054 12F
6 31 7 01 9 033 4F
6 31 7 01 4F
```

Example B:

```
00 994K
00 F 00 F
00 48F 00 F
00 997K
80 F 00 48F
26 909N
991 032 993 00014 1F
6 3 7 -064 12K
6 33 7 -054 12F
6 31 7 01 9 033 4F
6 31 7 01 4F
```

Example C:

```
00 994K
00 F 00 F
00 48F 00 F
00 997K
80 F 00 48F
00 12K
00 F 00 F
00 48F 00 F
00 15K
80 F 00 48F
26 909N
991 032 993 00014
9 032 11 00014
6 3 7 -064 12K
6 33 7 -054 12F
6 31 7 01 9 033 4F
6 31 7 01 4F
```

meter format is illustrated. The assumed runs hypothesized in making up Table III require changing of integer type parameters; all three examples hence begin with DOI format and must be initially loaded with the white switch. (It is good practice to assist the operator by writing "white switch" on the parameter tape at this point.) Examples A and B have a distinct advantage over example C in that each comes immediately to a black switch stop after setting those parameters applicable to all runs - Example A would stop with reader having just read the 26 977N, Example B would stop with the reader having just read the 1F. If this first portion of the tape is marked "master parameters", the operator can be instructed in the event of sum check failure to reload "master parameters", then skip down tape to first uncompleted run and continue (see Appendix A for detailed sample instructions). If parameters are as in Example C such a restart could not be made without completely repeating the first run, since with parameters as in this example no stop will occur until the end of the first run. The Example C arrangement would of course be desirable for a single run since only one operator function is required to load all parameters and start the run. For any of the examples shown the computer will stop (if the black switch is not on ignore) after each of the four runs i.e. with the reader positioned successively at the end of each of the last four lines of the examples in Table III. At each of these stops a black switch restart must be given to continue with the next run. (Again, it is good practice to assist the operator by inserting a length of delays in the tape at these points and writing "black switch" directly on the tape.)

## VI. OUTPUT FORMAT

A series of results of actual orbit code runs is presented in Table IV. The various samples have been selected to illustrate the normal features of the orbit code output format.

Results from five different runs are shown in the figure. A horizontal line has been drawn into the figure between each of the runs as an aid in explanation. Referring to the figure, each of the runs is seen to begin with a special line, which gives descriptive information, followed by a skipped line and then followed by successive similar lines in more or less regular format.

The initial descriptive line in each run consists of (1) a three digit base 16 number giving the number of R-K steps between prints, (2) a decimal fraction giving the value of the rest mass of the accelerated particle, (3) a decimal fraction giving the quantity $2^{-12}/\Delta r$, $\Delta r$ being in cyclotron units and (4) a base 16 number giving the value of the sum check cell - since the setting of the sum check cell varies with different fields and different code configurations, this latter quantity serves as an identification of both the field and the features of the code employed, all-be-it a rather disguised identification. If the initial base 16 number in the descriptive line is preceeded by four x's as in the second and third runs in Table 4, the integration is in the "bacKward" direction.

Following the descriptive information successive lines of fractions are printed giving the values of the run variables at the beginning of the run and at successive azimuths thereafter, as specified by the "number of R-K steps between prints" parameter. The fourth run of the table has full run variable output, the successive fractions in a line being z, $p_z$, r, $p_r$, $\phi/4$, K, and $(1/64)\Sigma r$ all in

Table IV: Sample of Orbit Code Output

```
030  +93821323  +03016911  0000000000

+25000000  +04000000  +00000000  +02800000  +00000000
+26270722  +03466910  +00638749  +02800000  +17754807
N1+L036-34
+27197752  +02799159  +01352221  +02800000  +35541464
+27765054  +02009199  +02169838  +02800000  +53363193


XXXX030  +93821323  +03016911  N1+L036-34

+21500000  +02500000  +00000000  +03200000  +00000000
+21853356  +03043668  +02673477  +03200000  +19143110
+22305406  +03954380  +06199329  +03200000  +38386961
+23800861  +05636582  +12275570  +03200000  +57920228
+38113062  000000002J  0000000004


XXXX030  +93821323  +03016911  N1+L036-34

+27765054  +02009199  +00000000  +02800000
+27197779  +02799108  +00816824  +02800000
LLLLLLLLLF
+26270819  +03466812  +01530514  +02800000
+25000211  +03999870  +02169487  +02800000


030  +93821323  +03016911  N1+L036-36

+00400000  +00000000  +25000000  +02000000  +00000000  +02800000  +00000000

-00151302  -00131173  +25429324  +01453701  +00505935  +02800000  +17864553

-L470802-6
-00351261  +00040919  +25645996  +00840116  +01033395  +02800000  +35743408

+00270764  +00114357  +25660313  +00196957  +01589046  +02800000  +53630399


010  +93821323  +03016911  80L60-6JFN

+04000000  +00000000  +32000000  -03500000  +00000000  +02800000
-00220820  -02868051  +16275272  -01547451  -00380515  +02813864
LLLLLLLLLF
-04538485  -02163811  +25731440  +07805494  -10210726  +02813864
-03941405  +05641545  +32058694  -03998785  +03751615  +02827173
+07804444  +06966362  +14049880  -00841750  +02558834  +02840552
/+15189032///0000000007  0000000005


(((((((((+25000000  0000000007  0000000005
```

the same units as input. It will be noticed in this example that following lines are double spaced from the first. This results since the line of output is exactly the maximum length allowed by the printer and so a carriage return is fed in by the printer in addition to the one on the tape. To prevent this double spacing in the output, the quantity $(1/64)\Sigma r$ is often surpressed as described in Appendix H, in which case the format shown in the fifth run of the table is obtained. For median plane runs the output arrangement is the same, except that the columns for z, and $p_z$ are not printed so the first column is now r, the second $p_r$, etc. Examples 1 and 2 are normal median plane runs while Example 3 is a median plane run with $(1/64)\Sigma r$ surpressed.

In all examples except the second, a short line consisting of a base 16 number is seen to occur between the second and third normal lines of output. This short word, as mentioned previously, indicates a sum check failure and is a 10 character base 16 number showing the negative absolute value of the amount by which the memory sum check fails to be zero. Due to particular characteristics of the printer the first character of this word usually prints during the return stroke of the printer thereby making reading difficult - the sum check failure in the first example is for instance "N1+L036-S4" but then "N" which is printed on top of the first "3" is quite difficult to discern. A sum check failure as in the first example in the figure will, as also mentioned previously, always occur whenever a new code and field are being assembled. The sum failure in the 3rd example reflects modification of the code to surpress the $(1/64)\Sigma r$, the failure in the 4th example reflects the modification to include axial motion and that in the final example again reflects surpression of the $(1/64)\Sigma r$. After each of the sum check failures in this sequence of runs the code has been restarted with the white switch which sets the sum cell to its proper

value so that the failure will not occur again. This is acceptable in this circumstance since each of the failures occurred due to changes deliberately made in the program. If such a failure had occurred other than after the second line of a given run or when no code changes had been made, a computer memory failure would be indicated and proper procedure would be to restart the series of runs beginning with a complete reloading of all tapes. (See sample instructions to operator, Appendix A).

The second of the group of runs shown in the table is seen to consist of four lines of normal output followed by a special fifth line. This special fifth line is printed whenever r exceeds it's allowable limits, either maximum or minimum, and terminates the run. The first quantity in this line is a fraction giving the value of r at the point at which the r interval test failed, the second quantity is a base 16 number telling the number of R-K steps from the last print to the angle at which the r interval tests failed, and the final number is a similar base 16 number, giving the total number of prints performed up to the point where the r interval test failed.

As mentioned in section II, the variable z is required to be at all times less than 1/8 in magnitude. If this test fails the run is also terminated and a line of output as shown at the end of example 5 is printed. This line is identical to that indicating r interval failure except that a "/" proceeds the information (actually a double slant - "//" - is printed but, again due to characteristics of the printer, the first slant prints during the return pass of the printer and so is difficult to discearn). The numerical quantities printed following the "/" are identical with the r interval failure format described in the previous paragraph.

In all of the examples except the fifth the kinetic

energy is seen to remain constant indicating that acceleration has not been included. In Example 5 the energy changes except between the second and third lines of output indicating that in the other three intervals accelerating gaps have been crossed.

The final line in Table IV, not attached to any of the examples, shows the output line which is obtained when the computer is restarted as described in Appendix K following one of the various white switch stops (other than the sum check stop already discussed). Output in this case consists of a series of parenthesis "(((((((" followed by a line of information again identical to that described in the r interval test failure above.

In addition to format features already mentioned, the third run in the table, it will be noticed, is a backward run starting with initial conditions which are identical to the final conditions from the first run of the table. It is seen that the backward run reproduces the results of the forward run to an accuracy of approximately 1 in $10^5$; the magnitude of the disagreement reflects truncation errors in the integration process and is one of the ways available for estimating the effect of such errors. (A second and better way of estimating such errors is to make runs with a greater number of R-K steps per sector using the techniques described in either Appendix C or Appendix D).

## VII. Computation of Running Time

Running time for the Orbit Code involves three principle factors: (1) the time required to carry out the basic integration step, (2) the time required to compute the magnetic field, and (3) the time required for printing out results. In addition to these three factors there is also a highly variable interval of time required for the loading of the program and field. (The variability in loading arises principly from tape handling time when fields and programs are loaded in DOI format. If a hex tape with fields and program is available loading time is fast and relatively uniform.)

In the following list times are given for each of the above segments of the computation. The time required for computation of the magnetic field is of course approximately proportional to the number of harmonics employed and so examples are given with several harmonic numbers in the field.

A. Median Plane Runs

1. Basic integration process----------0.32 sec/R-K step
2. Field computation
   - A. One harmonic----------0.09 sec/R-K step
   - B. Three harmonics----------0.27 sec/R-K step
   - C. Three harmonics plus one bump harmonic----------0.36 sec/R-K step
3. Print time (median plane 5 col. print)---1.10 sec/line

B. Axial Motion Runs

1. Basic Integration Process----------0.54 sec/R-K step
2. Field Computation
   - A. One harmonic----------0.15 sec/R-K step
   - B. Three harmonics----------0.45 sec/R-K step
3. Print time (7 col. print)----------1.55 sec/line

C. Tape Loading Time (hexadecimal program and field tape)

1. 150 field values----------27 sec.
2. 450 field values----------38 sec.

Section VIII. Flow Charts, Temporary Storage Guides, and Order Pairs for the Program

In order to facilitate possible revisions of the code, it is written in blocks using relative addresses within blocks and with an S box[1.] assigned to each block so that inter-block transfers can also be relative using S addresses. As mentioned previously the program is also written in two parts, the first being the "Load B routine" which places the field in memory in the proper arrangement, and the second, which overwrites the Load B routine, consisting of the main program and the parameter loading routines. In the following pages the Load B routine is covered first in three pages, the first page showing the arrangement of the Load B program tape including the S box settings, and the second and third sheets giving the order pairs for the successive major blocks of the Load B routine.

Following the Load B routine similar information is given for the main program including at the beginning a functional flow chart for the routine. In the flow chart solid lines denote functional blocks whereas the dotted lines denote coding blocks. In a number of cases it will be noted that coding blocks include several functional blocks.

Notice in the flow charts that entry in the main loop of the program at the beginning of a run is at the point immediately following the gap crossing test and the gap crossing test is hence first made after completion of the first Runge-Kutta step. An accelerating gap placed at $\theta=0$ will therefore not be crossed until the end of the first full revolution and the initial energy employed in such a run should thus be the value after the initial crossing of the $\theta=0$ gap. Note also in the flow charts that the run termination exit from the main loop occurs immediately after

the last printing of the output; if this printing occurs at the same azimuth as a gap crossing the print will give the after gap crossing energy.

Following the main flow chart, temporary storage guides are given showing the usage of the S3 and S5 temporary storage. S4 locations are also used for temporary storage but with the restriction that nothing of a semipermanent nature can be stored here, i.e. at the end of a block all information in S4 is assumed to be no longer needed and can be overwritten by the next block. Since S4 is used for innumerable purposses no storage breakdown is given for this section. If, in the running of the orbit code, a stop of unexplained character occurs it is usually useful to obtain order pair and fraction post mortums of the S3 and S5 temporary storage locations. From this data a great deal of information can be discerned concerning the particular orbit.

Following the temporary storage guides, pages of order pair lists for the various coding blocks are included and in addition, for some blocks, a detailed flow chart of the block is given.

The program is believed to be completely internally protected against undetected overflow. It will be noted in several places that this protection is often at the expense of operational speed. (Divide orders are used rather than left shifts, etc.) In view of the frequent important decisions based on results from this code, this protection is felt to be a wise precaution.

# PROGRAM TAPE I - "LOAD B PROGRAM"

| Abs. Addr. | Rel. Addr. | Order Pairs | | | | Comments |
|---|---|---|---|---|---|---|
| 999 | | | | | | Library Decimal Order Input Routine |
| . | | | | | | |
| . | | | | | | |
| . | | | | | | |
| 1023 | | | | | | |

## S Box Settings

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 3 | | 00 | F | 00 | 4F | Zero'th | location | black | S3 |
| 4 | | 00 | F | 00 | 25F | " | " | " | S4 |
| 5 | | 00 | F | 00 | 33F | " | " | " | S5 |
| 6 | | 00 | F | 00 | 289F | " | " | " | S6 |
| 7 | | 00 | F | 00 | 129F | " | " | " | S7 |
| 8 | | 00 | F | 00 | 371F | " | " | " | S8 |
| 9 | | 00 | F | 00 | 55F | " | " | " | S9 |
| 10 | | 00 | F | 00 | 410F | " | " | " | SK |
| 11 | | 00 | F | 00 | 435F | " | " | " | SS |
| 12 | | 00 | F | 00 | 64F | " | " | " | SN |
| 13 | | 00 | F | 00 | 112F | " | " | " | SJ |
| 14 | | 00 | F | 00 | 330F | " | " | " | SF |
| 15 | | 00 | F | 00 | 227F | " | " | " | SL |

## Coding Section S9
## New Field Loader Routine

| | | | | | | |
|---|---|---|---|---|---|---|
| 55 | 0 | L5 | 36L | 40 | 17L | |
| 56 | 1 | L5 | 15L | 42 | 3S4 | |
| 57 | 2 | F1 | 15S3 | L0 | 15S3 | |
| 58 | 3 | 40 | S4 | L0 | 16S3 | |
| 59 | 4 | L0 | 16S3 | 40 | 1S4 | |

| Abs. Addr. | Rel. Addr. | Order Pairs | | | | Comments |
|---|---|---|---|---|---|---|
| 60 | 5 | 40 | 6S4 | 19 | 18F | |
| 61 | 6 | L4 | 17S3 | 10 | 18F | |
| 62 | 7 | 42 | 10L | 42 | 27L | |
| 63 | 8 | F1 | 19S3 | 40 | 2S4 | |
| 64 | 9 | F5 | 2S4 | 40 | 2S4 | |
| 65 | 10 | 32 | 11L | 80 | ...F | |
| 66 | 11 | 26 | 9L | 41 | 4S4 | |
| 67 | 12 | L5 | 3S4 | 42 | 19L | |
| 68 | 13 | L5 | 17S3 | 46 | 14L | |
| 69 | 14 | 50 | ...F | 50 | 14L | |
| 70 | 15 | 26 | S7 | SL | SS | |
| 71 | 16 | 36 | 17L | 0F | 1F | |
| 72 | 17 | S5 | F | 22 | 19L | |
| 73 | 18 | L5 | 38L | 74 | 2S5 | Multiply H's and G's by I soch. code scale factor |
| 74 | 19 | 00 | 2F | 40 | ...F | |
| 75 | 20 | L5 | 19L | L0 | 6S4 | |
| 76 | 21 | 42 | 19L | F5 | 4S4 | |
| 77 | 22 | 40 | 4S4 | L0 | 14S3 | |
| 78 | 23 | 36 | 24L | 22 | 14L | |
| 79 | 24 | F1 | 18S3 | L4 | 14S3 | |
| 80 | 25 | L4 | 19S3 | 40 | 5S4 | |
| 81 | 26 | F5 | 5S4 | 40 | 5S4 | |
| 82 | 27 | 32 | 28L | 80 | ...F | |
| 83 | 28 | 26 | 26L | L5 | 37L | |
| 84 | 29 | 40 | 17L | F5 | 3S4 | |
| 85 | 30 | 42 | 3S4 | F5 | S4 | |
| 86 | 31 | 40 | S4 | L3 | S4 | |
| 87 | 32 | 36 | 33L | 22 | 33L | |
| 88 | 33 | 20 | 33L | F5 | 1S4 | |
| 89 | 34 | 40 | 1S4 | 32 | 35L | |

| Abs. Addr. | Rel. Addr. | Order Pairs | | | Comments |
|---|---|---|---|---|---|
| 90 | 35 | 26 | 8L 24 | 999F | |
| 91 | 36 | S5 | F 22 | 19L | |
| 92 | 37 | NO | F NO | F | |
| 93 | 38 | 20 | F 00 | F | |

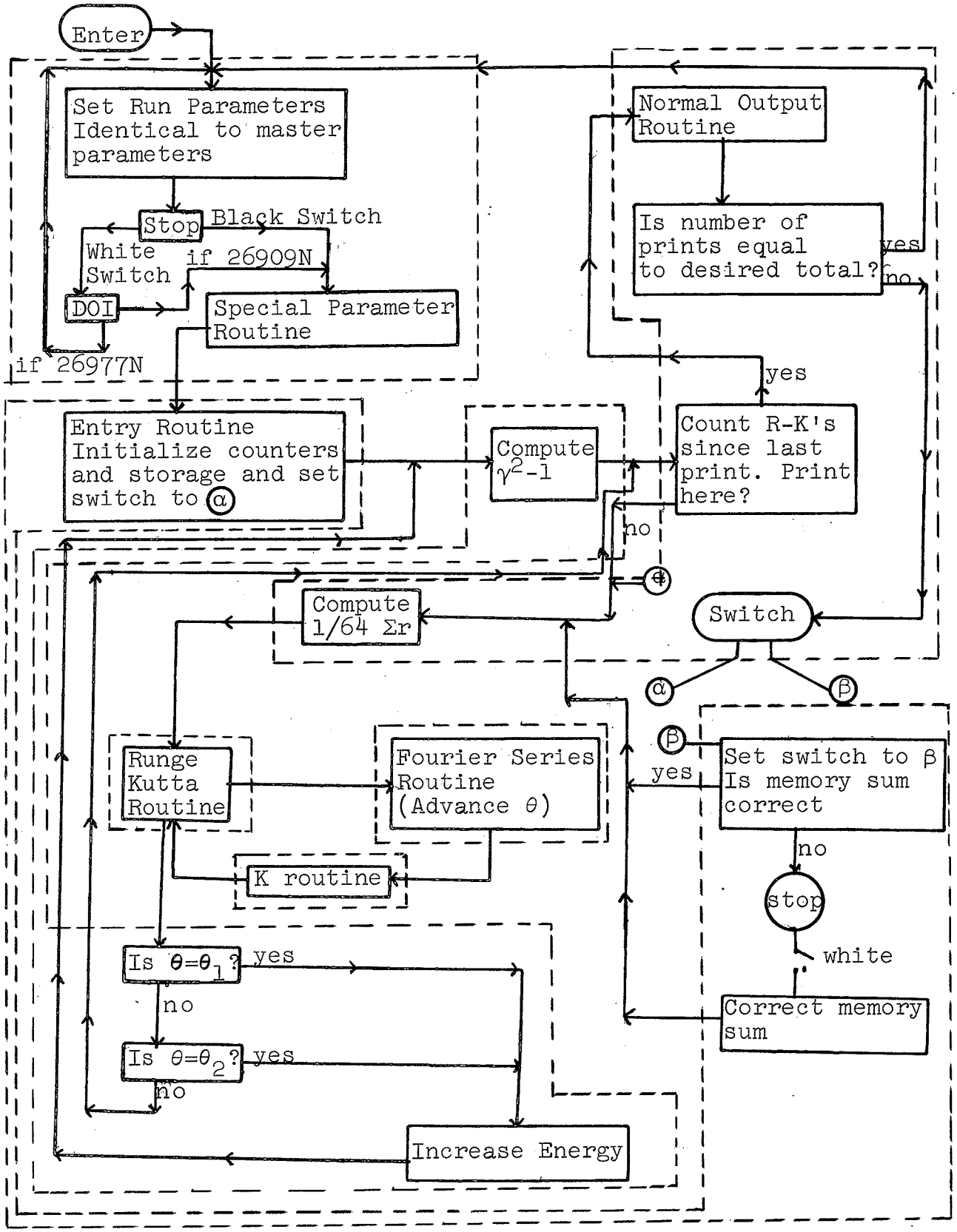
Coding Section 40S9

Compute $E_o$ and $2^{-12}/\Delta r$ Routine

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 95 | 0 | 50 | 6S5 | 7J | 16L | ⑦ | | | .93114 |
| 96 | 1 | 50 | 5S5 | 70 | 17L | ⑥ | | | .00000511 |
| 97 | 2 | S5 | F | 40 | F | | | | $m_o c^2 \times 10^{-5}$ |
| 98 | 3 | S5 | F | 40 | (996)F | | | | $(m_o c^2 \times 10^n) > 1/10$ |
| 99 | 4 | 75 | 18L | 40 | 1F | | | | 00 F 00 10F |
| 100 | 5 | L3 | 1F | 36 | 3L | | | | |
| 101 | 6 | 50 | 3S5 | 75 | 5S5 | ④ | ⑥ | | |
| 102 | 7 | 7J | 1S5 | 40 | 1F | ② | | | |
| 103 | 8 | 50 | 1F | 7J | S5 | | ① | | |
| 104 | 9 | 40 | 1F | 50 | F | | | | |
| 105 | 10 | 7J | 2S5 | 40 | F | ③ | | | |
| 106 | 11 | 50 | F | 7J | 4S5 | | ⑤ | | |
| 107 | 12 | 40 | F | 50 | F | | | | |
| 108 | 13 | 75 | 19L | 66 | 1F | | | | .1282474 |
| 109 | 14 | S5 | F | 40 | (17)F | | | | |
| 110 | 15 | 26 | 55F | 00 | F | | | | |
| 111 | 16 | 40 | F | 00 | 4311 | 4000 | 0000J | | |
| 112 | 17 | 00 | F | 00 | 0000 | 0511 | 0000J | | |
| 113 | 18 | 00 | F | 00 | 10F | | | | |
| 114 | 19 | 00 | F | 00 | 1282 | 4740 | 0000J | | |

24 999N Transfer directive

Main Program Flow Chart



# MAIN PROGRAM ORGANIZATION

Memory layout, main routine:

003 - 015: S box parameters
004 - 024: S3
(004 - 015) Run Parameters
(016 - 024) Field Parameters
025 - 032: S4 Temporary Storage
033 - 054: S5 " "
055 - 063: S9 routine - Library R-1
064 - 111: SN, Interpolate Routine
112 - 128: SJ, Cosine Routine
129 - 226: S7, Fourier Series Routine
227 - 290: SL, Gap Cross and Punch Routine
289 - 329: S6, Runge-Kutta Routine
330 - 371: SF, "K" Routine
371 - 408: S8, Constants and Sum Check and Entry Routines
410 - 434: SK, Cosine Table
435 - 908: SS, B,H,G,h,g Table
909 - 998: New Parameter Routine
999 - 1023: Library Decimal Order Input Routine

## Temporary Storage Guides
## Coding Section S3
## Parameters

| Abs. Addr. | Rel. Addr. | Parameter | Significance |
|---|---|---|---|
| 4 | 0 | z | Described in Table II |
| 5 | 1 | $p_z$ | " " " " |
| 6 | 2 | r | " " " " |
| 7 | 3 | $p_r$ | " " " " |
| 8 | 4 | $\phi/4$ | " " " " |

| Abs. Addr. | Rel. Addr. | Parameter | Significance |
|---|---|---|---|
| 9 | 5 | K | Described in Table II |
| 10 | 6 | $1/64\ \Sigma r$ | " " " " |
| 11 | 7 | $\delta K$ | " " " " |
| 12 | 8 | $\theta_1 \times 2^{-18}$ | " " " " |
| 13 | 9 | $\theta_2 \times 2^{-18}$ | " " " " |
| 14 | 10 | $E_o$ | " " " " |
| 15 | 11 | $-1 + (R_p \times 2^{-39})$ | " " " " |
| 16 | 12 | $-1 + (R_R \times 2^{-39})$ | " " " " |
| 17 | 13 | $2^{-12}/\Delta r$ | Calculated by Load B Program |
| 18 | 14 | $a \times 2^{-39}$ | Described on Page 11 |
| 19 | 15 | $H \times 2^{-39}$ | " " " " |
| 20 | 16 | $h \times 2^{-39}$ | " " " " |
| 21 | 17 | $d \times 2^{-19}$ | " " " " |
| 22 | 18 | $b \times 2^{-39}$ | " " " " |
| 23 | 19 | $\alpha \times 2^{-39}$ | " " " " |
| 24 | 20 | $\beta \times 2^{-39}$ | " " " " |

Coding Section S5

Temporary Storage

| | | | |
|---|---|---|---|
| 033 | 0 | $q_o$ | Runge Kutta Round-off Corrections |
| 034 | 1 | $q_1$ | |
| 035 | 2 | $q_2$ | |
| 036 | 3 | $q_3$ | |
| 037 | 4 | $q_4$ | |
| 038 | 5 | $\theta$ | $\theta \times 2^{19}$=angle in units of R-K steps |
| 039 | 6 | $\theta_c$ | switch to signal $\theta$ advance on alternate Fourier Series cycles |
| 040 | 7 | Count $R_p$ | count R-K's between prints |
| 041 | 8 | Count $R_R$ | " prints per run |
| 042 | 9 | $\gamma^2-1$ | momentum squared |

| Abs. Addr. | Rel. Addr. | Parameter | Significance |
|---|---|---|---|
| 043 | 10 | $B(r_i)$ | magnetic field at grid points |
| 044 | 11 | $B(r_i + 1,\theta)$ | |
| 045 | 12 | $B(r_i + 2,\theta)$ | |
| 046 | 13 | $B(r_i + 3,\theta)$ | |
| 047 | 14 | $B_\theta(r_i,\theta)$ | $\theta$ derivitive of field at grid points |
| 048 | 15 | $B_\theta(r_i + 1,\theta)$ | |
| 049 | 16 | $B_\theta(r_i + 2,\theta)$ | |
| 050 | 17 | $B_\theta(r_i + 3,\theta)$ | |
| 051 | 18 | X | fractional part of $r/\Delta r$ |
| 052 | 19 | $B(r,\theta)$ | field at r, $\theta$. |
| 053 | 20 | $B_r(r)$ | r derivitive of field at $r,\theta$. |
| 054 | 21 | $B_\theta(r)$ | $\theta$ " " " " " " |

PROGRAM TAPE II - "MAIN PROGRAM"

Coding Block S9

Square Root Routine (Library Routine R-1)

| | | | | | | |
|---|---|---|---|---|---|---|
| 55 | 0 | 40 | 1F | K5 | F | |
| 56 | 1 | 42 | 8L | 51 | 1F | |
| 57 | 2 | 10 | 1F | SJ | F | |
| 58 | 3 | 40 | 2F | 50 | F | |
| 59 | 4 | L5 | 1F | 66 | 2F | ½ hang. if square root of negative number |
| 60 | 5 | S5 | F | L0 | 2F | |
| 61 | 6 | 10 | 1F | 36 | 8L | |
| 62 | 7 | L4 | 2F | 26 | 3L | |
| 63 | 8 | L5 | 2F | 22 | ...F | |

Coding Section SN

Interpolation Subroutine

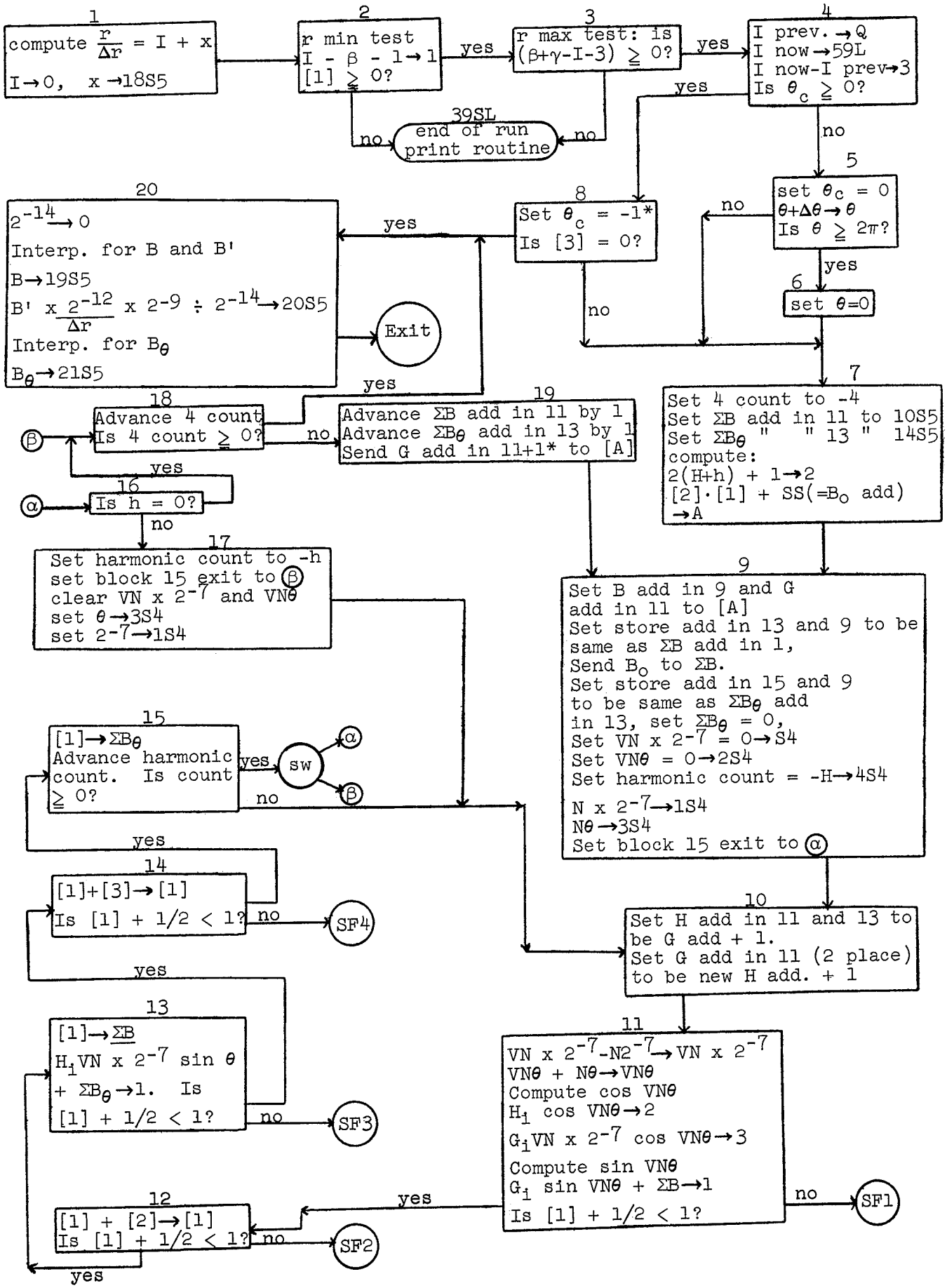
| Abs. Addr. | Rel. Addr. | Order Pairs | | | | Comments |
|---|---|---|---|---|---|---|
| 064 | 0 | 40 | S4 | K5 | F | |
| 065 | 1 | 40 | 1S4 | 10 | 20F | |
| 066 | 2 | 42 | 22L | 42 | 39L | |
| 067 | 3 | F5 | 22L | 42 | 19L | |
| 068 | 4 | 42 | 35L | F5 | 19L | |
| 069 | 5 | 42 | 17L | 42 | 36L | |
| 070 | 6 | F5 | 17L | 42 | 24L | |
| 071 | 7 | 42 | 38L | 09 | 1F | |
| 072 | 8 | L4 | S4 | 40 | 2S4 | |
| 073 | 9 | 10 | 1F | 40 | 3S4 | |
| 074 | 10 | 49 | 4S4 | L5 | S4 | |
| 075 | 11 | 10 | 1F | 40 | 5S4 | |
| 076 | 12 | L4 | 4S4 | 40 | 6S4 | |
| 077 | 13 | L5 | 3S4 | L0 | 4S4 | |
| 078 | 14 | 50 | S4 | 40 | 4S4 | |
| 079 | 15 | 7J | 2S4 | 40 | S4 | |
| 080 | 16 | 50 | S4 | 7J | 47L | |
| 081 | 17 | 40 | S4 | 50 | ...F | |
| 082 | 18 | 79 | 5S4 | 50 | 3S4 | |
| 083 | 19 | 40 | 3S4 | 7J | ...F | |
| 084 | 20 | L4 | 3S4 | 40 | 3S4 | |
| 085 | 21 | 50 | 6S4 | 7J | 4S4 | |
| 086 | 22 | 40 | 5S4 | 50 | ...F | |
| 087 | 23 | 7J | 4S4 | 10 | 1F | |
| 088 | 24 | 40 | 4S4 | 50 | ...F | |

| Abs. Addr. | Rel. Addr. | Order Pairs | | | | Comments |
|---|---|---|---|---|---|---|
| 089 | 25 | 7J | 6S4 | 10 | 1F | |
| 090 | 26 | L0 | 4S4 | 40 | 6S4 | |
| 091 | 27 | 50 | 3S4 | 7J | 5S4 | |
| 092 | 28 | 00 | 2F | 40 | 4S4 | |
| 093 | 29 | 50 | 6S4 | 7J | S4 | |
| 094 | 30 | L4 | 4S4 | 40 | 4S4 | |
| 095 | 31 | L5 | 1S4 | 42 | 46L | |
| 096 | 32 | 36 | 46L | LJ | 2S4 | |
| 097 | 33 | 50 | 6S4 | 40 | 6S4 | |
| 098 | 34 | 7J | 47L | L4 | 3S4 | |
| 099 | 35 | 40 | 1S4 | L5 | ...F | |
| 100 | 36 | 50 | 1S4 | L0 | ...F | |
| 101 | 37 | 40 | 3S4 | 7J | 6S4 | |
| 102 | 38 | 40 | 2S4 | L5 | ...F | |
| 103 | 39 | 50 | 3S4 | L0 | ...F | |
| 104 | 40 | 40 | 1S4 | 7J | 5S4 | |
| 105 | 41 | 40 | 5S4 | 50 | 1S4 | |
| 106 | 42 | 7J | S4 | 10 | 3F | |
| 107 | 43 | L4 | 2S4 | L4 | 5S4 | |
| 108 | 44 | 40 | 2S4 | 49 | S4 | |
| 109 | 45 | L5 | 2S4 | 66 | S4 | |
| 110 | 46 | L5 | 4S4 | 22 | ...F | |
| 111 | 47 | 40 | F | 00 | 1666 | 6666 6667J |

Coding Section SJ
Cosine Routine

| | | | | | | |
|---|---|---|---|---|---|---|
| 112 | 0 | 40 | 14L | K5 | F | |
| 113 | 1 | 42 | 15L | 42 | 16L | |
| 114 | 2 | L5 | 14L | L0 | 1S8 | |
| 115 | 3 | 32 | 2L | 40 | 14L | |
| 116 | 4 | L4 | 1S8 | 40 | F | |

# Fourier Series Routine (S7)
## Flow Chart



| Abs. Addr. | Rel. Addr. | Order Pairs | | | | Comments |
|---|---|---|---|---|---|---|
| 117 | 5 | L4 | 14L | 10 | 1F | |
| 118 | 6 | 40 | 1F | 36 | 8L | |
| 119 | 7 | L5 | F | 22 | 8L | |
| 120 | 8 | L1 | 14L | 40 | F | |
| 121 | 9 | 80 | 2F | L0 | 1S8 | |
| 122 | 10 | 32 | 12L | L5 | 15L | |
| 123 | 11 | L4 | F | 40 | 14L | |
| 124 | 12 | 26 | 14L | L5 | 16L | |
| 125 | 13 | L6 | 1F | 40 | 14L | |
| 126 | 14 | 00 | F | 00 | F | |
| 127 | 15 | L5 | SK | 22 | ... F | |
| 128 | 16 | L1 | SK | 22 | ... F | |

Coding Section S7

Fourier Series Routine

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 129 | 0 | L5 | S4 | L0 | 1S4 | | | |
| 130 | 1 | 40 | S4 | L5 | 2S4 | | | |
| 131 | 2 | L4 | 3S4 | 40 | 2S4 | | | |
| 132 | 3 | 50 | F | 50 | 3L | | | |
| 133 | 4 | 26 | SJ | 40 | F | | | to cos routine |
| 134 | 5 | 50 | F | 7J | ...F | | H | |
| 135 | 6 | 40 | 2F | 50 | F | | | |
| 136 | 7 | 7J | S4 | 40 | F | 11 | | |
| 137 | 8 | 50 | F | (7J) | ...F | | G | |
| 138 | 9 | 40 | 3F | L5 | 2S4 | | | |
| 139 | 10 | L4 | 4S8 | 50 | 10L | | | |
| 140 | 11 | 26 | SJ | 40 | F | | | |
| 141 | 12 | 50 | F | (79) | ...F | | G | |
| 142 | 13 | 50 | 41L | L4 | ...F | | ΣB | left add used in 36L |
| 143 | 14 | 40 | 1F | LL | 1F | | | |

| Abs. Addr. | Rel. Addr. | Order Pairs | | | | | | Comments |
|---|---|---|---|---|---|---|---|---|
| 144 | 15 | 36 | 16L | SF | 1F | 11 | | stop if \|B-G sin $\theta$\| |
| 145 | 16 | L5 | 1F | L4 | 2F | 12 | | $\geq$ 1/2 |
| 146 | 17 | 40 | 1F | LL | 1F | | | |
| 147 | 18 | 36 | 19L | SF | 2F | | | stop if \|B-G sin$\theta$ + |
| 148 | 19 | L5 | 1F | 40 | ...F | | B | H cos\| $\geq$ 1/2 |
| 149 | 20 | 50 | F | 7J | S4 | | | |
| 150 | 21 | 40 | F | 50 | ...F | 13 | H | |
| 151 | 22 | 7J | F | L4 | ...F | | $\Sigma B_\theta$ | |
| 152 | 23 | 40 | 1F | LL | 1F | | | |
| 153 | 24 | 36 | 25L | SF | 3F | | | stop if \|$B_\theta$+NH sin\| |
| 154 | 25 | L5 | 1F | L4 | 3F | | | $\geq$ 1/2 |
| 155 | 26 | 40 | 1F | LL | 1F | 14 | | |
| 156 | 27 | 36 | 28L | SF | 4F | | | stop if \|$B_\theta$+NH sin |
| 157 | 28 | L5 | 1F | 40 | ...F | | $\Sigma B_\theta$ | + NG cos\| $\geq$ 1/2 |
| 158 | 29 | F5 | 4S4 | 40 | 4S4 | 15 | | |
| 159 | 30 | 36 | ...F | F5 | 8L | | | |
| 160 | 31 | 42 | 5L | 42 | 21L | 10 | | |
| 161 | 32 | F5 | 5L | 42 | 8L | | | |
| 162 | 33 | 42 | 12L | 26 | L | | | |
| 163 | 34 | 00 | F | 00 | 3F | | | number of sectors (N from eq. 6) |
| 164 | 35 | L3 | 16S3 | 36 | 41L | 16 | | |
| 165 | 36 | 40 | 4S4 | L5 | 13L | | | |
| 166 | 37 | 46 | 30L | 41 | S4 | | | |
| 167 | 38 | 40 | 2S4 | L5 | 5S5 | 17 | | |
| 168 | 39 | 40 | 3S4 | 19 | (6)F | | | Set by entry interlude if used (=6 in prog. tape) |
| 169 | 40 | 40 | 1S4 | 22 | 30L | | | |
| 170 | 41 | F5 | 5S4 | 40 | 5S4 | 18 | | |
| 171 | 42 | 32 | 88L | F5 | 13L | | | |
| 172 | 43 | 42 | 13L | F5 | 22L | 19 | | |
| 173 | 44 | 42 | 22L | F5 | 8L | | | |

| Abs. Addr. | Rel. Addr. | Order Pairs | | | | | Comments |
|---|---|---|---|---|---|---|---|
| 174 | 45 | 42 | 8L | 42 | 47L | | |
| 175 | 46 | L5 | 13L | 42 | 19L | | |
| 176 | 47 | 42 | 48L | L5 | ...F | $B_o$ | |
| 177 | 48 | 50 | 35L | 40 | ...F | $\Sigma B$ | left add. used in 57L |
| 178 | 49 | L5 | 22L | 42 | 50L | | |
| 179 | 50 | 42 | 28L | 41 | ...F | $\Sigma B_\theta$ | |
| 180 | 51 | 40 | S4 | 40 | 2S4 | | |
| 181 | 52 | L1 | 15S3 | 40 | 4S4 | 9 | |
| 182 | 53 | 50 | 9S8 | L5 | 34L | | |
| 183 | 54 | 00 | (32)F | 40 | 1S4 | | Set by entry interlude if used (=32 in prog. tape) |
| 184 | 55 | 50 | 5S5 | 75 | 34L | | |
| 185 | 56 | 00 | 39F | 40 | 3S4 | | |
| 186 | 57 | L5 | 48L | 46 | 30L | | |
| 187 | 58 | 22 | 30L | 00 | S5 | | |
| 188 | 59 | 00 | F | 00 | F | $\delta$ | |
| 189 | 60 | 50 | 2S3 | 75 | 13S3 | 1 | |
| 190 | 61 | 10 | 27F | 40 | F | | |
| 191 | 62 | L5 | F | F0 | 20S3 | | |
| 192 | 63 | 40 | 1F | 32 | 64L | 2 | if $r < r_o + \Delta r$ |
| 193 | 64 | 26 | 39SL | L5 | 20S3 | | |
| 194 | 65 | F4 | 14S3 | L0 | 8S8 | | |
| 195 | 66 | L0 | F | 32 | 67L | 3 | if $r > r_{max.} - \Delta r$ |
| 196 | 67 | 26 | 39SL | S5 | SS | 1 | rt. add. used in 83L |
| 197 | 68 | 40 | 18S5 | 50 | 59L | | |
| 198 | 69 | L5 | F | 40 | 59L | 4 | |
| 199 | 70 | S0 | F | 40 | 3F | | |
| 200 | 71 | L5 | 6S5 | 36 | 84L | | |
| 201 | 72 | 41 | 6S5 | L5 | 5S5 | | |
| 202 | 73 | L4 | 5S8 | 40 | 5S5 | 5 | |
| 203 | 74 | L0 | 1S8 | 32 | 75L | | |

| Abs. Addr. | Rel. Addr. | Order Pairs | | | | | Comments |
|---|---|---|---|---|---|---|---|
| 204 | 75 | 26 | 76L | 41 | 5S5 | 6 | |
| 205 | 76 | L1 | 8S8 | 40 | 5S4 | | |
| 206 | 77 | L5 | 6S8 | 42 | 13L | | |
| 207 | 78 | L5 | 7S8 | 42 | 22L | | |
| 208 | 79 | L5 | 15S3 | L4 | 16S3 | 7 | |
| 209 | 80 | 80 | 1F | F4 | 9S8 | | |
| 210 | 81 | 40 | 2F | 50 | 2F | | |
| 211 | 82 | 75 | 1F | S5 | F | | |
| 212 | 83 | L4 | 67L | 26 | 45L | | |
| 213 | 84 | F1 | 6S5 | 40 | 6S5 | 8 | |
| 214 | 85 | L3 | 3F | 32 | 88L | | |
| 215 | 86 | 26 | 76L | 00 | F | | |
| 216 | 87 | 00 | F | 00 | F | | |
| 217 | 88 | 00 | F | 19 | 13F | | |
| 218 | 89 | 40 | F | L5 | 18S5 | | |
| 219 | 90 | J0 | 10S5 | 50 | 90L | | |
| 220 | 91 | 26 | SN | 40 | 19S5 | 20 | |
| 221 | 92 | 75 | 13S3 | 10 | (9)F | | Set by entry interlude if used(set = 9 in prog. tape). |
| 222 | 93 | 66 | F | S5 | F | | |
| 223 | 94 | 40 | 20S5 | L5 | 18S5 | | |
| 224 | 95 | 50 | 14S5 | 50 | 95L | | |
| 225 | 96 | 26 | SN | 40 | 21S5 | | |
| 226 | 97 | 26 | SF | 26 | 30S8 | | exit |

Coding Section SL

New Gap Crossing and Punch Routine

| | | | | | | |
|---|---|---|---|---|---|---|
| 227 | 0 | L5 | 5S5 | L0 | 8S3 | |
| 228 | 1 | 40 | F | L3 | F | |
| 229 | 2 | 32 | 5L | L5 | 5S5 | |
| 230 | 3 | L0 | 9S3 | 40 | F | |
| 231 | 4 | L3 | F | 32 | 5L | |

| Abs. Addr. | Rel. Addr. | Order Pairs | | | | Comments |
|---|---|---|---|---|---|---|
| 232 | 5 | 26 | 22L | 50 | 4S3 | to punch routine |
| 233 | 6 | 7J | 4S3 | 40 | F | |
| 234 | 7 | 50 | F | 7J | F | |
| 235 | 8 | 40 | 1F | 50 | 1F | |
| 236 | 9 | 7J | 3S8 | 40 | 1F | |
| 237 | 10 | LJ | 1F | L0 | F | |
| 238 | 11 | 40 | F | 19 | 2F | |
| 239 | 12 | 40 | 1F | 50 | F | |
| 240 | 13 | 7J | F | L0 | 1F | |
| 241 | 14 | 00 | 2F | 40 | F | |
| 242 | 15 | 50 | F | 7J | 7S3 | |
| 243 | 16 | L4 | 5S3 | 40 | 5S3 | |
| 244 | 17 | L5 | 5S3 | 66 | 10S3 | compute $\gamma^2$-1 |
| 245 | 18 | S5 | F | 40 | 1F | |
| 246 | 19 | 00 | 1F | 40 | F | |
| 247 | 20 | 10 | 1F | 7J | 1F | |
| 248 | 21 | L4 | F | 40 | 9S5 | |
| 249 | 22 | L5 | 34L | 42 | 25L | punch routine |
| 250 | 23 | F5 | 7S5 | 40 | 7S5 | |
| 251 | 24 | L0 | 11S3 | 32 | 35L | |
| 252 | 25 | 92 | 515F | L5 | ...F | |
| 253 | 26 | 50 | 8F | 50 | 26L | |
| 254 | 27 | 26 | 46L | F5 | 25L | |
| 255 | 28 | 42 | 25L | L0 | 38L | |
| 256 | 29 | 32 | 25L | 92 | 131F | |
| 257 | 30 | F5 | 8S5 | 40 | 8S5 | |
| 258 | 31 | L0 | 12S3 | 36 | ...F | set by entry routine |
| 259 | 32 | 92 | 143F | 27 | 977F | |
| 260 | 33 | L5 | 33L | 50 | 10S8 | reset 31L |
| 261 | 34 | 42 | 31L | 50 | S3 | 1st word to punch |

| Abs. Addr. | Rel. Addr. | Order Pairs | | | | Comments |
|---|---|---|---|---|---|---|
| 262 | 35 | 41 | 7S5 | L5 | 2S3 | |
| 263 | 36 | 10 | 6F | L4 | 6S3 | |
| 264 | 37 | 40 | 6S3 | 22 | 21S6 | |
| 265 | 38 | 12 | 515F | L5 | 7S3 | last word to punch |
| 266 | 39 | 92 | 515F | L5 | 2S3 | |
| 267 | 40 | 50 | 8F | 50 | 40L | |
| 268 | 41 | 26 | 46L | L5 | 7S5 | |
| 269 | 42 | 82 | 40F | 92 | 967F | r and z interval failure |
| 270 | 43 | L5 | 8S5 | 82 | 40F | punch routine |
| 271 | 44 | 92 | 137F | L5 | 37S6 | |
| 272 | 45 | 42 | 21S6 | 26 | 977F | |

Coding Section 46SL

Convert to decimal and punch routine

| | | | | | | |
|---|---|---|---|---|---|---|
| 273 | 0 | 40 | F | L1 | 16L | |
| 274 | 1 | S4 | L | 46 | 1L | |
| 275 | 2 | 42 | 14L | 36 | 6L | |
| 276 | 3 | L5 | F | 36 | 5L | |
| 277 | 4 | 92 | 706F | 22 | 5L | |
| 278 | 5 | 92 | 642F | 00 | 2F | |
| 279 | 6 | J0 | 8L | 7J | 17L | |
| 280 | 7 | 42 | 8L | 22 | 8L | |
| 281 | 8 | 00 | 63F | 19 | F | Library Routine P-2 |
| 282 | 9 | L6 | F | 10 | 39F | |
| 283 | 10 | 75 | 15L | 00 | 36F | |
| 284 | 11 | 82 | 4F | 10 | 40F | |
| 285 | 12 | L5 | 1L | L0 | 16L | |
| 286 | 13 | 46 | 1L | 36 | 10L | |
| 287 | 14 | 92 | 965F | 22 | F | |
| 288 | 15 | 00 | F | 00 | 10F | |
| 289 | 16 | S4 | 1F | 00 | 1023F | |
| 290 | 17 | 00 | 1F | K9 | 1000F | |

# Coding Section S6
## R-K Routine

Library routine F-1 with following changes:
(1) 0, 1 and 2 relative not used. These cells are overwritten by P-2 routine which proceds F-1 in memory.
(2) Scaling parameter m in 8L and 11L is written into routine rather than being used as a program parameter.
(3) Exit order is permanently set to exit to gap crossing routine. No subroutine type of entry is used.
(4) Enter routine at right of 21L.
(5) Exit to auxiliary subroutine permanently set.

| Abs. Addr. | Rel. Addr. | Order Pairs | | | Comments |
|---|---|---|---|---|---|
| 291 | 2 | 00 F 00 | F | | |
| 292 | 3 | L5 $(q_{i,j})$ F 40 | 1F | From 21 | |
| 293 | 4 | L5 $(k_{i,j})$ F 40 | F | By 20<br>By 18 | |
| 294 | 5 | L0 (1) F L0 | 1F | By 23 | $(k_{i,j}-B_j q_{ij})$ |
| 295 | 6 | 40 3F 50 | $(A_j)$F | By 24 | |
| 296 | 7 | 7J 3F L4 | 3F | | $(k_{ij}-B_j q_{ij})$ |
| 297 | 8 | 10 (m)F 40 | 3F | m=20 | $(A_j+1)2^{-m}$ |
| 298 | 9 | L4 $(y_{ij})$F 40 | $(y_{i,j+1})$F | By 17 | Step $y_i$ |
| 299 | 10 | L5 3F 50 | 2F | | Form $r_{i,j+1}$ |
| 300 | 11 | 00 (m)F 40 | 3F | m=2 | |
| 301 | 12 | 50 $(C_j)$F 7J | F | By 25 | $C_j k_{ij}-k_{ij}$ |
| 302 | 13 | L0 F L4 | 1F | | |
| 303 | 14 | L4 3F L4 | 3F | | $q_{ij}+8r_{i,j+1}$ |
| 304 | 15 | L4 3F 40 | $(q_{i,j+1})$F | By 19 | |
| 305 | 16 | L5 9L L4 | 13L | | |
| 306 | 17 | 42 9L 46 | 9L | | increase all addresses depending on i by 1 until i=n |
| 307 | 18 | L4 39L 46 | 4L | | |
| 308 | 19 | L4 40L 42 | 15L | | |
| 309 | 20 | 46 3L L0 | 37L | | |
| 310 | 21 | 36 3L L5 | (33)L | By 22 | |

| Abs. Addr. | Rel. Addr. | Order Pairs | | | | | Comments |
|---|---|---|---|---|---|---|---|
| 311 | 22 | 42 | 21L | 36 | SL | From 2' By 2 | Increase j from 0 to 3 and then exit |
| 312 | 23 | 46 | 5L | 10 | 10F | | |
| 313 | 24 | L4 | 18L | 42 | 6L | | Adjust addresses which depend on j |
| 314 | 25 | 46 | 12L | 50 | 25L | | Call in auxiliary sub-routine |
| 315 | 26 | 26 | 60S7 | 41 | 2F | | Clear 2F so that it can be used as zero |
| 316 | 27 | L5 | 38L | 26 | 17L | | Start new i cycle |
| 317 | 28 | 40 | F | 00 | F | 1/2 | $C_o$, $C_3$ |
| 318 | 29 | NO | F | 00 | F | -1/2 | $A_o$ |
| 319 | 30 | 40 | F | 00 | 2071 0678 1186 J | | $1/\sqrt{2}\ C_1, A_2$ |
| 320 | 31 | 80 | F | 00 | 2928 9321 8814 J | | $-1/\sqrt{2}\ A_1, C_2$ |
| 321 | 32 | 80 | F | 00 | 1666 6666 6667 J | | $-5/6\ A_3$ |
| 322 | 33 | LJ | 1025F | 06 | 1058L | -11, 1 25, 34L | Expressed in units of $2^{-9}, 2^{-19}, 2^{-29}, 2^{-39}$ |
| 323 | 34 | LJ | 3074F | 06 | 3107L | -9 2, 27 35L | Addresses used to set addresses to refer to the constants $A_j, C_j$ and make the address in 5L,1 or 2 according as $B_j$,2 or 1, and to stop the address in 21, dependent on j, and to stop when positive. |
| 324 | 35 | LF | 2F | 06 | 2084L | -8, 2 26, 36L | |
| 325 | 36 | LJ | 1025F | 07 | 37L | -11, 1 28, 37L | |
| 326 | 37 | 64 | 38F | 00 | 33L | | |
| 327 | 38 | 80 | 4F | 00 | 4F | | |
| 328 | 39 | 00 | 21F | 00 | 21F | | "4" replaced by "6" when median plane overwrite is used. |
| 329 | 40 | 00 | 8F | 00 | 8F | | |

Coding Section SF

"K" Routine

| Abs. Addr. | Rel. Addr. | Order Pairs | | | | Comments |
|---|---|---|---|---|---|---|
| 330 | 0 | 50 | 3S3 | 79 | 3S3 | Note: $' = \frac{d}{d\theta}$ |
| 331 | 1 | 40 | F | 50 | 1S3 | |
| 332 | 2 | 79 | 1S3 | L4 | 1F | |
| 333 | 3 | L4 | 9S5 | 50 | 3L | |
| 334 | 4 | 26 | S9 | 40 | 3S4 | |
| 335 | 5 | 50 | 2S3 | (75) | 3S3 | |
| 336 | 6 | 66 | 3S4 | S5 | F | ÷ hang. if $\lvert r' \rvert \left(= \left\lvert \frac{rp_r}{q} \right\rvert \right) \geq 1$ |
| 337 | 7 | 40 | 5S4 | 7J | 2S8 | |
| 338 | 8 | 40 | 2S4 | 50 | 2S3 | |
| 339 | 9 | (75) | 1S3 | 66 | 3S4 | ÷ hang. if $\lvert Z' \rvert \left(= \left\lvert \frac{rp_z}{q} \right\rvert \right) \geq 1$ |
| 340 | 10 | S5 | F | 40 | 6S4 | |
| 341 | 11 | 7J | 2S8 | 40 | S4 | |
| 342 | 12 | 19 | (8)F | 40 | 7S4 | scaling on $\frac{dB}{d\theta}$ set by entry interlude if used (set = 8 in program tape). |
| 343 | 13 | L5 | 5S3 | 66 | 10S3 | |
| 344 | 14 | 7J | 2S3 | L4 | 2S3 | ÷ hang. if $K > E_o$ |
| 345 | 15 | L0 | 3S4 | 10 | 2F | |
| 346 | 16 | 66 | 3S4 | 7J | 2S8 | ÷ hang. if $\lvert \phi'/4 \rvert \geq 1$ |
| 347 | 17 | 40 | 4S4 | 50 | 5S4 | |
| 348 | 18 | 7J | 21S5 | 40 | 5S4 | |
| 349 | 19 | 50 | 2S3 | 7J | 2S3 | |
| 350 | 20 | 40 | F | 50 | F | |
| 351 | 21 | 7J | 20S5 | L0 | 5S4 | |
| 352 | 22 | 40 | F | L5 | S3 | |
| 353 | 23 | 66 | 2S3 | S5 | F | ÷ hang. if $\lvert Z \rvert \geq r$ |
| 354 | 24 | 40 | 5S4 | (71) | F | |
| 355 | 25 | 66 | 7S4 | 79 | 2S8 | ÷ hang. if $\lvert P'_Z \rvert \geq 1$ |
| 356 | 26 | 40 | 1S4 | 19 | 1F | |
| 357 | 27 | 40 | F | 50 | 2S3 | |
| 358 | 28 | 7J | 19S5 | 40 | 1F | |
| 359 | 29 | 50 | 5S4 | 7J | 6S4 | |

| Abs. Addr. | Rel. Addr. | | Order Pairs | | | Comments |
|---|---|---|---|---|---|---|
| 360 | 30 | 40 | 2F | 50 | 2F | |
| 361 | 31 | 75 | 21S5 | 10 | 2F | |
| 362 | 32 | 66 | 7S4 | S5 | F | ÷hang. if $\left\lvert \frac{ZP_z}{4q} B\theta \right\rvert \geqq 0$ |
| 363 | 33 | 40 | 2F | L1 | 3S4 | |
| 364 | 34 | 10 | 2F | L0 | 2F | |
| 365 | 35 | L4 | 1F | 66 | F | ÷hang. if $\lvert P'_r \rvert \geqq 1$ |
| 366 | 36 | (79) | 2S8 | 40 | 3S4 | |
| 367 | 37 | LL | 2S3 | 32 | 38L | |
| 378 | 38 | KK | 1F | L5 | S3 | stop if $\lvert r \rvert \geqq 1/2$ |
| 369 | 39 | 00 | 2F | 40 | F | |
| 370 | 40 | LL | F | 32 | 26S6 | |
| 371 | 41 | 92 | 455F | 26 | 39SL | test is $\lvert Z \rvert \geqq 1/8$ end run if yes |

Coding Section S8

Constants

| | | | | | | |
|---|---|---|---|---|---|---|
| 371 | 0 | | | | | |
| 372 | 1 | | "$2\pi$" = 2 times no. of R-K steps/rev. x $2^{-19}$ | | | |
| 373 | 2 | | $2^m \Delta\theta_{rad.} = \pi/6$ in prog. tape.[6] | | | |
| 374 | 3 | | 1/3 | | | |
| 375 | 4 | | "$3\pi/2$" = 3/2 times no. of R-K steps/rev. x $2^{-19}$ | | | |
| 376 | 5 | | $2^{-19} = \Delta\theta$ | | | |
| 377 | 6 | 00 | F | 00 | 10S5 | Initial address used in S7 |
| 378 | 7 | 00 | F | 00 | 14S5 | " " " " " |
| 379 | 8 | 4* | | | | |
| 380 | 9 | 0 | | | | |

6. m in 373 must agree with m in 297 and 300. $\Delta\theta_r$ is angular extent of R-K step in radians. See Appendix C. to change.

## Cosing Section 10S8

## New Sum Check and Entry Routine

| Abs. Addr. | Rel. Addr. | Order Pairs | | | | Comments |
|---|---|---|---|---|---|---|
| 381 | 0 | L5 | 15L | 42 | 1L | |
| 382 | 1 | 41 | F | L5 | ...F | |
| 383 | 2 | L4 | F | 40 | F | |
| 384 | 3 | F5 | 1L | 42 | 1L | |
| 385 | 4 | L0 | 18L | 32 | 1L | |
| 386 | 5 | L5 | F | L0 | 5S7 | |
| 387 | 6 | L0 | 8S7 | L0 | 12S7 | |
| 388 | 7 | L0 | 21S7 | L0 | 47S7 | |
| 389 | 8 | L0 | 14SJ | L0 | 59S7 | |
| 390 | 9 | L4 | 20S3 | L4 | 13S3 | |
| 391 | 10 | L4 | 14S3 | L4 | 15S3 | |
| 392 | 11 | L4 | 16S3 | L4 | 19S3 | |
| 393 | 12 | L0 | 940F | L0 | 927F | |
| 394 | 13 | 40 | F | L3 | F | |
| 395 | 14 | 36 | 35SL | 82 | 40F | |
| 396 | 15 | 92 | 131F | 4F | S9 | |
| 397 | 16 | L5 | 19L | L0 | F | |
| 398 | 17 | 40 | 19L | 26 | L | |
| 399 | 18 | N1 | F | L5 | 986F | |
| 400 | 19 | 00 | F | 00 | F | sum check cell |
| 401 | 20 | L5 | 58S7 | 42 | 21L | Entry Routine |
| 402 | 21 | 40 | 59S7 | 41 | ...F | |
| 403 | 22 | F5 | 21L | 42 | 21L | |
| 404 | 23 | L0 | 28L | 32 | 21L | |
| 405 | 24 | L5 | 11S3 | 42 | 7S5 | |
| 406 | 25 | L5 | 25L | 50 | 33SL | |
| 407 | 26 | 42 | 31SL | 41 | 6S3 | |
| 408 | 27 | 92 | 135F | 26 | 17SL | |
| 409 | 28 | N0 | 59S7 | 41 | 9S5 | |

Coding Block SS

Field Storage

| Abs. Addr. | Rel. Addr. | Order Pairs | Comments |
|---|---|---|---|
| 410 | 0 | $B_o(\beta\Delta r)$ | |
| 411 | 1 | $H_1(\beta\Delta r)$ | |
| . | . | . | |
| . | . | . | |
| . | . | . | |

Parameter Routine

| | | | | | | |
|---|---|---|---|---|---|---|
| 909 | 0 | L5 | 1L | 42 | 18L | |
| 910 | 1 | 41 | 3F | 50 | 30L | |
| 911 | 2 | 41 | 2F | 81 | 4F | |
| 912 | 3 | L0 | 22L | 36 | 20L | |
| 913 | 4 | L4 | 21L | 40 | F | |
| 914 | 5 | L5 | 21L | 22 | 9L | |
| 915 | 6 | 50 | F | 74 | 22L | |
| 916 | 7 | 00 | 39F | 40 | F | |
| 917 | 8 | 50 | 1F | 75 | 22L | |
| 918 | 9 | S5 | F | 40 | 1F | |
| 919 | 10 | 91 | 4F | 36 | 23L | |
| 920 | 11 | 50 | 1F | L5 | F | |
| 921 | 12 | 66 | 1F | 10 | 1F | |
| 922 | 13 | L5 | F | L4 | 1F | |
| 923 | 14 | 40 | F | SJ | F | |
| 924 | 15 | 40 | 1F | L1 | 2F | |
| 925 | 16 | 32 | 18L | L1 | F | |
| 926 | 17 | 40 | F | L1 | 1F | |
| 927 | 18 | 40 | 1F | 26 | ...F | |
| 928 | 19 | 50 | F | 79 | F | |

| Abs. Addr. | Rel. Addr. | Order Pairs | | | | Comments |
|---|---|---|---|---|---|---|
| 929 | 20 | 40 | 2F | 22 | 2L | |
| 930 | 21 | 00 | F | 00 | 5F | |
| 931 | 22 | 00 | F | 00 | 10F | |
| 932 | 23 | L0 | 22L | 36 | 25L | |
| 933 | 24 | L4 | 22L | 26 | 6L | |
| 934 | 25 | 40 | S4 | L3 | S4 | |
| 935 | 26 | 36 | 28L | L5 | 51L | |
| 936 | 27 | 42 | 18L | 26 | 11L | |
| 937 | 28 | L5 | 29L | 42 | 18L | |
| 938 | 29 | 26 | 11L | 00 | 35L | |
| 939 | 30 | L5 | 3F | 32 | 32L | |
| 940 | 31 | L5 | 1F | 40 | ...F | |
| 941 | 32 | 26 | 1L | L5 | F | |
| 942 | 33 | 42 | 31L | F1 | 3F | |
| 943 | 34 | 40 | 3F | 26 | 2L | |
| 944 | 35 | L5 | 63L | 40 | 12S7 | |
| 945 | 36 | L5 | 19L | 40 | 8S7 | |
| 946 | 37 | L5 | 64L | 40 | 5SF | |
| 947 | 38 | L5 | 66L | 40 | 36SF | |
| 948 | 39 | L5 | 65L | 40 | 9SF | |
| 949 | 40 | L5 | 67L | 40 | 24SF | |
| 950 | 41 | 92 | 911F | 26 | 48L | |
| 951 | 42 | L5 | 59L | 40 | 12S7 | |
| 952 | 43 | L5 | 57L | 40 | 8S7 | |
| 953 | 44 | L5 | 58L | 40 | 9SF | |
| 954 | 45 | L5 | 60L | 40 | 5SF | |
| 955 | 46 | L5 | 61L | 40 | 36SF | |
| 956 | 47 | L5 | 62L | 40 | 24SF | |
| 957 | 48 | L5 | F | 42 | 12S3 | |
| 958 | 49 | L5 | 11S3 | 00 | 28F | |

| Abs. Addr. | Rel. Addr. | Order Pairs | | | | | Comments |
|---|---|---|---|---|---|---|---|
| 959 | 50 | 82 | 12F | 92 | 967F | | |
| 960 | 51 | L5 | 10S3 | 50 | 42L | | |
| 961 | 52 | 50 | 8F | 50 | 52L | | |
| 962 | 53 | 26 | 46SL | L5 | 13S3 | | |
| 963 | 54 | 50 | 8F | 50 | 54L | | |
| 964 | 55 | 26 | 46SL | L5 | 29S8 | | |
| 965 | 56 | 82 | 40F | 26 | 30S8 | | entry routine |
| 966 | 57 | 50 | F | 7J | F | | |
| 967 | 58 | 75 | 1S3 | 66 | 3S4 | | |
| 968 | 59 | 50 | F | 79 | 77L | | |
| 969 | 60 | 50 | 2S3 | 75 | 3S3 | | |
| 970 | 61 | 79 | 2S8 | 40 | 3S4 | | |
| 971 | 62 | 40 | 5S4 | 71 | F | | |
| 972 | 63 | 50 | F | 7J | F | | |
| 973 | 64 | 50 | 2S3 | 71 | 3S3 | | |
| 974 | 65 | 71 | 1S3 | 66 | 3S4 | | |
| 975 | 66 | 7J | 2S8 | 40 | 3S4 | | |
| 976 | 67 | 40 | 5S4 | 75 | F | | |
| 977 | 68 | L5 | 59L | 42 | 70L | | |
| 978 | 69 | L5 | 69L | 50 | S3 | | |
| 979 | 70 | 42 | 71L | L5 | ...F | | |
| 980 | 71 | 50 | F | 40 | ...F | | |
| 981 | 72 | F5 | 71L | 42 | 71L | | |
| 982 | 73 | F5 | 70L | 42 | 70L | | |
| 983 | 74 | L0 | 76L | 32 | 70L | | |
| 984 | 75 | 24 | L | 26 | 999F | | |
| 985 | 76 | N2 | 71L | L5 | 90L | | |
| 986 | 77 | 00 | F | 00 | 0000 | 1000 0000J | z |
| 987 | 78 | 00 | F | 00 | F | | $p_z$ |
| 988 | 79 | 00 | F | 00 | 2500 | 0000 0000J | r |

| Abs. Addr. | Rel. Addr. | Order Pairs | | | | | | Comments |
|---|---|---|---|---|---|---|---|---|
| 989 | 80 | 00 | F | 00 | 0850 | 0000 | 0000J | $p_r$ |
| 990 | 81 | 00 | F | 00 | F | | | $\phi/4$ |
| 991 | 82 | 00 | F | 00 | 0280 | 0000 | 0000J | K |
| 992 | 83 | 00 | F | 00 | F | | | Not used |
| 993 | 84 | 00 | F | 00 | 0002 | 8000 | 0000J | $\Delta K$ |
| 994 | 85 | 00 | F | 00 | 1F | | | gap 1 |
| 995 | 86 | 00 | F | 00 | 1F | | | gap 2 |
| 996 | 87 | .. | .... | .. | .... | .... | .... | $E_o$ (set by Load B routine) |
| 997 | 88 | 80 | F | 00 | 16F | | | $R_k$=16 |
| 998 | 89 | 80 | F | 00 | F | | | $R_p$ is irrelevant |
| | | | | 26 | 977N | | | final directive |

End Main Program

---

Median Plane Overwrite

| | | | | |
|---|---|---|---|---|
| 331 | 26 | 333F | 00 | F |
| 338 | 40 | 27F | 26 | 343F |
| 347 | 40 | 29F | 22 | 356F |
| 359 | 22 | 363F | 00 | F |
| 364 | 10 | 2F | 50 | F |
| 261 | 42 | 258F | 50 | 6F |
| 135 | 40 | 2F | 22 | 138F |
| 149 | 26 | 158F | 00 | F |
| 219 | 50 | 43F | 50 | 219F |
| 221 | 26 | 330F | 00 | F |
| 327 | 80 | 6F | 00 | 6F |
| | | | 26 | 977N |

Load this tape with white switch after main program tape to set code for median plane runs

## Anti-Median Plane Overwrite

| Abs. Addr. | Rel. Addr. | Order Pairs | | | Comments |
|---|---|---|---|---|---|
| 331 | 40 | F | 50 | 5F | Load this tape with white switch to make axial motion runs with a median plane program tape. |
| 338 | 40 | 27F | 50 | 6F | |
| 347 | 40 | 29F | 50 | 30F | |
| 359 | 50 | 30F | 7J | 31F | |
| 364 | 10 | 2F | L0 | 2F | |
| 261 | 42 | 258F | 50 | 4F | |
| 135 | 40 | 2F | 50 | F | |
| 149 | 50 | F | 7J | 25F | |
| 219 | J0 | 43F | 50 | 219F | |
| 221 | 75 | 17F | 10 | 9F | |
| 327 | 80 | 4F | 00 | 4F | |
| 265 | 12 | 515F | L5 | 10F | |
| | | | 26 | 977N | |

Appendix A: Sample Instructions for Operator

A standard instruction sheet is available for use in Mistic production runs; the following sample instructions are written in format corresponding to this standard sheet. As a precaution against operator error it is in general good policy to have no more than three tapes involved in production runs. Hence runs of the type shown in the first two samples should if possible be made with the preparer of the run present to assist the operator in selecting proper tapes.

1. Single median plane run using DOI input tapes.

| Tape | Start | Stop | Comments |
|---|---|---|---|
| "Load B" | Bootstrap | 24 | Entire tape loads |
| B Parameters | Black Sw. | 24 | Entire tape loads |
| Main field | Black Sw. | 20 | Entire tape loads |
| Bump field | Black Sw. | 24 | Entire tape loads |
| Main Program | Black Sw. | 24 | 80% of tape loads |
| | Black Sw. | 24 | remainder of tape loads |
| Med. Plane Overwrite | White Sw. | 24 | Entire tape loads |
| Run Parameter | xxx Sw. | 4F | Memory sum check must be set after 2nd print. |
| | White Sw. | 24 | End of run |

xxx = white if initial parameters are in DOI format.
= black if initial parameters are in special routine format.

For runs with axial motion included the procedure is the same as above except the Median Plane Overwrite step is omitted.

2. To make Hexadecimal Program Tape.

Make at least one run as in example 1 so as to set sum check cell. At end of this run <u>without</u> clearing memory:

| Tape | Start | Stop | Comments |
|---|---|---|---|
| Rev. x-16S | Boot-no clear | 24 | Part of tape goes in |
| | white Sw. | 24 | More tape loads, much punching, remainder of tape loads |
| Directive tape* | black Sw. | | Sets directive at end of new program tape |
| Output tape from above | bootstrap | 24 or SF | If stop on 24 new tape okay, if stop on SF discard tape and repeat above |

* This tape will read 00K26984NdO where d denotes a single delay.

3. Series of Production Runs using Hexadecimal Program Tape.

In this example, it is assumed that changes in the program are desired (such as perhaps the inclusion of the constant acceleration overwrite, etc.) which cause the sum check to be incorrect. It is also assumed that all DOI parameter changes are applicable to all runs and can hence be on the same tape with the code changes, this tape being marked "Master Parameters". To prepare "Restart" tape see Appendix K.

| Tape | Start | Stop | Comments |
|---|---|---|---|
| Program | Bootstrap | 24 | Entire tape loads. Stop on SF instead of 24 means reader error-adjust reader(?) and reload. |
| Master Param. | White Sw. | 24 | Entire tape loads. |
| Run Param. | Black Sw. | 4F | Memory sum check must be set after 2nd print. |
| Run Param. | White Sw. | 24 | End of run 1. |
| Run Param. | Black Sw. | 24 | End of run 2. Continue to repeat this step to end of parameter tape. |

Notes to operator:

(1) If a 4F stop occurs (other than the one listed above after the second print), memory failure is indicated. In this case mark the "Run parameter" tape at the position which is under the reader, and repeat all of the above steps, except that when placing the run parameter tape under the reader, place it at the mark rather than at the beginning.

(2) If white switch stop other than 4F occurs, mark "Run parameter" tape at the position which is under the reader, bootstrap "restart" tape (no memory clear), reinsert "Run Parameter" tape at marked position and return to repeating of last instruction above.

Appendix B: Instructions for changing the number of machine sectors.

The number of machine sectors is stored as an integer in location 163($0K3_{16}$), and is presently set at 3. If the absolute angular extent of each R-K step is left unchanged (i.e. if the number of R-K's per revolution remains constant) then the number of sectors can be changed to any desired value by simply changing the integer in 163. If it is desired to change the number of sectors and at the same time change the length of the R-K step, as for instance would be required if it was desired to retain the same number of R-K's per sector, than the changes described in Appendix C must be performed in addition to the resetting of 163 to the desired value.

Appendix C: Instructions for changing the angular interval per R-K step.

To change the angular interval per R-K step several constants, the table of cosines, and the location of the field storage must all be changed

The constants to be changed are the following:

| Location | DOI format | |
|---|---|---|
| 372 | 00["2π"]F 00 F | "2π" is integer equal to twice the number of R-K steps per revolution. |
| 373 | 00 F 00[$2^m\Delta\theta$]J | where $\Delta\theta$ is the angular extent of an R-K step in radians. m is presently 2 and for maximum significance $1/2 \leq 2^m\Delta\theta<1$. |
| 297 | 10 m F 40 3F | where m is power of 2 by which $\Delta\theta$ is scaled (see above). |
| 300 | 00 m F 40 3F | same. |
| 375 | 00 "3π/2" F 00 F | "3π/2" is integer equal to 3/2 times the number of R-K's per revolution. (R-K's/ revolution must always be even number.) |

The table of cosines begins in location 410 and must consist successively of the cosines of all integral multiples of $\Delta\theta/2$ from 0 thru $\pi/2$ inclusive ($\Delta\theta/2$ must be such that $\pi/2$ is integral multiple). $\Delta\theta$ as presently set in the program is $\pi/24$ so that the table presently consists of 25 values. Immediately after the table of cosines the field storage begins. If$\Delta\theta$'s smaller than $\pi/24$ are employed, the field storage must be shifted since the table of cosines will in this case require more than 25 locations.

The location of the field storage is set at the very beginning of the "Load B Routine" Program tape, the 8th word on this tape being an integer presently set at 435. In order to shift the field storage, it is necessary to prepare a revised tape with the 435 changed to the desired value - overwriting of the 435 after loading the program will not

suffice since the quantity is used as an "S box".[1.]

The total number of storage locations available for fields is of course decreased by just the amount by which the table of cosines is enlarged.

Appendix D: Use of Fictitious N.

In general as seen from Appendix C, a change in the angular length of the R-K step involves considerable revision of the code. In the special case where it is desired to triple the number of R-K steps per revolution in fields containing no imperfection components, a convenient trick can be employed as follows:

1. Prepare a special field parameter tape with H=1 and h= number of main field harmonics.
2. Prepare a special field data tape with zeros inserted in all $H_1$, $G_1$ positions and with actual main field harmonics typed in the imperfection field position.
3. Change the constant $2^m\Delta\theta$ in location 373 to 1/3 of its previous value.

After making these changes proceed to run in the usual manner. 48 R-K steps will be required to complete a sector and 144 to complete a revolution.

Whenever round off error is suspected of apprieciably influencing the results of normal runs, this simple change by factor three in the number of R-K steps is a convenient procedure for obtaining an estimate of the extent of the error. Due to special features of the gap crossing routine, acceleration should not be included in such runs.

Appendix E: Acceleration Changes

The program, as written, approximates acceleration effects by increasing the kinetic energy of the particle at each of two arbitrary azimuths, the magnitude of the increase varying in the normal way as the cosine of the phase angle between particle and R-F as is indicated by eq. (6), Section I.

In many situations it is desirable to eliminate the phase factor from the acceleration thereby making acceleration constant at each gap (i.e. to use "square wave" acceleration). This is accomplished by the following change:

242: L5 11F 10 1F

To return to normal operation after constant acceleration operation:

242 50 F 7J 11F

The routine can also be caused to approximate betatron type acceleration, (i.e. the energy can be made to increase by the amount $\Delta K/2 \cos \phi$ at the end of every R-K step) by the change:

227: 22 232F L0 12F

Note that in this case $\Delta K$ is the maximum energy gain in 2 R-K steps rather than per turn. If desired this correction and the previous correction can be employed together to give "continuous constant acceleration". To return to normal operation after "continuous acceleration" operation:

227: L5 38F L0 12F

Appendix F: Providing additional field storage by omission of the special Input Routine.

As mentioned in the text, 90 additional field storage locations can be made available by leaving out the special input routine. To accomplish this for forward runs simply remove the program tape from the reader at the black switch stop which occurs near the end of this tape when loading. Substitute a DOI parameter tape and black switch. In this mode of operation all of the parameters in locations 4 thru 16 inclusive must be set on <u>every</u> run, since the master parameter routine is also omitted. The directive 26 401N at the end of the parameter tape will start the run.

To make backward runs in this mode the first of the backward runs must include the following code changes in the parameter tape:

```
137:  50    F  79    F
141:  50    F  7J    F
335:  50   6F  71   7F
339:  71   5F  66  28F
354:  40  30F  75    F
366:  7J 373F  40  28F
```

To revert to forward runs, following backward runs, the first of the forward runs must reset these orders by including in the parameters:

```
137:  50    F  7J    F
141:  50    F  79    F
335:  50   6F  75   7F
339:  75   4F  66  28F
354:  40  30F  71    F
366:  79 373F  40  28F
```

Appendix G: Runs with initial $\theta \neq 0$.

In normal operation, it is assumed that the $\theta=0$ angle of the magnetic field description is the angle at which orbits start i.e. the angle at which the initial conditions apply. If it is desired to start orbits at some other angle with respect to the field the following changes must be made in the program:

```
408:  92  135F  22  215F
215:  26  205F  L5  216F
216:  50  [n]F  46   38F
217:  26  244F  19   13F
```

For forward runs "n" is set at twice the number of R-K steps between $\theta=0$ and the starting azimuth. For backward runs "n" is set to be the complement with respect to twice the number of R-K's per revolution of the forward "n".

The print counter operates independently of $\theta$ by simply counting the R-K steps actually completed. Prints will therefore always be spaced at equal R-K step intervals from the initial $\theta$ irrespective of the value of that $\theta$.

Caution should be exercised in carefully labeling output from runs with $\theta \neq 0$, since the format is not readily distinguishable from other runs. (Actually the contents of the sum check cell do distinguish such runs in a disguised way, since the memory cells changed are included in the sum, and the sum will hence differ by the sum of the changes.)

To revert to normal operation after $\theta \neq 0$ runs reset as follows:

```
408:  92  135F  26  244F
215:  26  205F  00     F
216:  00     F  00     F
217:  00     F  19   13F
```

## Appendix H: Suppressing 1/64Σr in Code Output.

The quantity 1/64Σr was originally included in the routine as a means of obtaining an average orbit radius to use in a later computation of an isochronus correction to the given field. Subsequently the Equilibrium Orbit Code giving the exact average of r has come into use and consequently the 1/64Σr column in the general orbit code output is now seldom used.

This column can be omitted from the output by the following change:

265: 12 515F L5 10F

This change is especially desirable when making axial motion runs since as discussed in Section VII, the output line length for such runs with the 1/64Σr included is such as to cause double spacing by the printer between lines thereby giving an excessively long piece of paper.

To return to normal operation after running without 1/64Σr reset as follows:

265: 12 515F L5 11F.

Appendix J: To run with $\omega_{rf} \neq eB_o(0)/m_o$.

Since $\omega_{rf}$ is implicitly the basis of the system of field, length, and time units used in the code, it is necessary, in order to make runs corresponding to non-isochronus $\omega_{rf}$'s, to numerically rescale the complete field and the unit of length. Let $\omega_o$ be $eB_o(0)/m_o$. To run with $\omega_o \neq \omega_{rf}$ perform the following steps.

(1) In the field parameters change the value of $B_{cen}$ in volts by the factor $\omega_{rf}/\omega_o$. (i.e. the new $B_{cen}$ is now just $\omega_{rf} m_o/e$).

(2) Prepare a new data tape in which all B's, H's etc. are multiplied by the factor $\omega_o/\omega_{rf}$ (i.e. the data on the new tape are now in units of $4\omega_{rf} m_o/e$, as is required by the routine).

After making these changes proceed with the run in the normal manner.

Appendix K: Restarting the routine after a White Switch Stop.

Many of the overflow stops listed in Table I are not programmed in a manner such that operation can continue on to other runs without a special restart. When such a stop occurs in a sequence of runs and it is desired to continue to the next run in the sequence a "reset" tape containing the ten character word 92 0JL 22 10K should be bootstrap loaded with no memory clear. This operation will print a series of ((((( followed by the same series of information as for an r too large or too small stop. The code will reset itself and stop on the normal end of run stop ready for the next parameters of the next run.

Appendix L: To change the Sector Number of the Imperfection Field Fourier Series.

From equation 6, section I it is seen that the imperfection field is implicitly assumed in the program to have only once per revolution periodicity. In some cases it is desirable to study imperfection fields with periodicity of 2 or more per revolution and if the format 6 is employed this requires that zeros must be inserted for the lower order harmonics of the imperfection. The following correction serves to insert a sector number in the imperfection field series in equation 6 in a manner exactly analogous to the sector number appearing in the main field series.

To insert a sector number other than unity in the imperfection field series load the following modification tape with the white switch after loading of the program tape.

```
00 167K
40  27F 50  38F
75 908F S5 ___F
40  28F 26 906F
00 906K
L5 908F 00  32F
40  26F 22 159F
00    F 00   pF    p is number of sectors in imperfection
26 977N                field.
```

This correction uses three locations normally set aside for field storage and therefore when it is used the number of field values must be $\leq 473$.

Appendix M: To Change Scaling of Field Derivitives. "Entry Interlude"

The maximum value of the field derivitives depends sensitively on the characteristics of the particular field. The code is of course protected against overflow of these quantities by stops in the Fourier Series routine - if these stops occure and it is still desired to study the particular field it is necessary to change the scaling of the field derivitives.

To accomplish a change in scaling load the following tape with the white switch after loading of the main program tape.

```
00  33K
L5   6L 42 221F
L0   5L 46 342F
42 168F L5   3L
50  41F L0   6L
46 183F 26 977F
00   1F 00   3F
00   nF 00   mF    derivitives scaled by 2^-n
26  33N
```

The main program, as written, has derivitives scaled by $2^{-7}$. Hence if overflow stops occur and the above modification is employed values of n greater than 7 should be tried. To maintain significance it is desirable to use the smallest value of n which will prevent overflow; a few trial searches should therefore be made to determine the proper value of n for use with a particular field.