LA-3903

257
9-30-68

458

LOS ALAMOS SCIENTIFIC LABORATORY
of the
University of California
LOS ALAMOS • NEW MEXICO

General Purpose Computer Code
for Particle Trajectories and Other Problems

UNITED STATES
ATOMIC ENERGY COMMISSION
CONTRACT W-7405-ENG. 36

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

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

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

This report expresses the opinions of the author or authors and does not necessarily reflect the opinions or views of the Los Alamos Scientific Laboratory.

Printed in the United States of America. Available from
Clearinghouse for Federal Scientific and Technical Information
National Bureau of Standards, U. S. Department of Commerce
Springfield, Virginia 22151

Price: Printed Copy \$3.00; Microfiche \$0.65

MASTER

LOS ALAMOS SCIENTIFIC LABORATORY
of the
University of California
LOS ALAMOS • NEW MEXICO

Report written: March 1968

Report distributed: September 16, 1968

General Purpose Computer Code
for Particle Trajectories and Other Problems

by

H. R. Lewis, Jr.

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

UNIVERSITY OF CALIFORNIA
LOS ALAMOS SCIENTIFIC LABORATORY
(CONTRACT W-7405-ENG-36)
P. O. Box 1663
Los Alamos, New Mexico 87544

IN REPLY
REFER TO:

To: Holders of LA-3903

ERRATA

Page 10. Immediately after statement number 152, insert these two statements:

KCON = 1
KSTEP = 0

Page 13. Replace the statement immediately after statement 25 by this statement:

IF(ISTER)42,42,26

CONTENTS

	Page
Abstract	5
I. Introduction	5
II. Instructions for Using ORB	6
A. Input Format and Some Output	6
B. Further Output	7
C. Necessary Auxiliary Subprograms	7
III. A Sample Problem	8
Acknowledgment	8
References	8
Appendix A. Listings of ORB, RECORD, DER, AXES, and DEQ	9
Appendix B. Listings of Suitable Auxiliary Subprograms for the Sample Problem	19
Appendix C. Data for the Sample Problem	20

**THIS PAGE
WAS INTENTIONALLY
LEFT BLANK**

GENERAL PURPOSE COMPUTER CODE FOR PARTICLE TRAJECTORIES

AND OTHER PROBLEMS

by

H. R. Lewis, Jr.

ABSTRACT

Instructions for using a computer code for computing and graphing the solutions of a general system of not more than six first-order differential equations are presented with an example. The user is allowed very flexible control of the type and format of graphical output by means of data cards. A listing of the code is given in FORTRAN for the CDC 6600.

I. INTRODUCTION

A computer code has been written for easy application to the problem of computing and graphing trajectories of a charged particle moving in a general electromagnetic field or, indeed, to the problem of solving a general system of not more than six first-order differential equations of the form

$$\frac{dy_i}{dt} = f_i(t, y_1, \dots, y_n), \quad i = 1, \dots, n (n \leq 6).$$

The heart of the code is a subroutine called DEQ that uses Nordsieck's method for solving systems of differential equations of this form.^{1,2} To understand the arguments of DEQ as they are used in the trajectory code, reference should be made to the instructions given in Reference 1.

The trajectory code, called ORB, is described here for use on the CDC 6600 computer. The user is allowed very flexible control of the type and format of graphical output (made by the SC-4020 device which produces graphs on 35 mm film). In addition

to t and the y_i , up to three other quantities, completely determined by the user, may be calculated, printed, and graphed. Both two-dimensional and three-dimensional (stereoscopic) graphs can be easily obtained. The input parameters necessary for specifying stereoscopic graphs are defined in the description of the stereoscopic projection subroutine designated by J513A in the program library at the Los Alamos Scientific Laboratory.*

Instructions for using the trajectory code are given in Section II, and a sample problem is described in Section III. The FORTRAN listings of the main program (ORB) and four subroutines (RECORD, DER, AXES, and DEQ) are given in Appendix A. The main program and these four subroutines must always be present and are not to be altered. In Appendix B,

* The trajectory code makes explicit use of several subroutines that are available in the program library for the CDC 6600 at the Los Alamos Scientific Laboratory.

FORTTRAN listings of subprograms intended to be supplied by the user are given for the sample problem. Data for the sample problem are given in Appendix C.

II. INSTRUCTIONS FOR USING ORB

A. Input Format and Some Output

Before describing the input data cards for ORB, we make a few remarks concerning some of the quantities that must be supplied as input. The quantities ERROR and CLIF are arguments of DEQ and are described in Reference 1. The arrays DV, A, SF, FWF, ICHAR, and ICON are arguments of the library stereoscopic projection subroutine designated by J513A and are described in the writeup for that subroutine. The arrays FUN1, FUN2, and FUN3 are the values of the quantities which the user may choose to have ORB evaluate in addition to t and the y_i . They are calculated in function subprograms VAL1, VAL2, and VAL3, respectively (See Section II C). EN is the value of a quantity (for example, the energy) which is calculated by Subroutine ENER (See Section II C). EN can be anything the user desires. The one-dimensional arrays Y, F, and CN have six elements each and they must appear in dimension statements in those auxiliary subprograms which use them. The Y(I) and F(I) are the dependent variables and derivatives, respectively:

$$\frac{dy(I)}{dt} = F(I).$$

(T corresponds to t .) The CN(I) are constants supplied by the user for calculating derivatives, etc. CN(6) sometimes has a special use, as described under item 4 of this section.

The input data cards for ORB are as follows:

(1) HMIN, HMAX, FEE, ERROR, NEQ, MO, M1, M2, M3, IE.
FORMAT (1P4E12.7, 6I1).

HMIN and HMAX are the minimum and maximum step sizes to be used by DEQ. HMIN must be greater than or equal to zero. The direction of integration is determined by the sign of HMAX: HMAX > 0 means increasing T ; HMAX < 0 means decreasing T . FEE is the maximum allowed fractional deviation of EN from its initial value. (If IE = 0, the deviation of EN is checked; otherwise, the deviation is not checked.) ERROR is defined in Reference 1. NEQ is the number of first-order differential equations to be integrated. If MO $\neq$ 0, T is treated mod 2π ; if M1 $\neq$ 0, Y(I) is treated mod 2π ($I = 1, 2, 3$).

(2) (YMIN(I), YMAX(I), I = 1, 10). FORMAT (1P6E12.7).

Minimum and maximum values of the quantities Y(I) ($I=1,6$), T, FUN1, FUN2, and FUN3, respectively. Here and later these quantities [Y(I), T, FUN1, FUN2, and FUN3] are referred to by the following code numbers:

<u>Code Number</u>	<u>Quantity</u>
1	Y(1)
2	Y(2)
3	Y(3)
4	Y(4)
5	Y(5)
6	Y(6)
7	T
8	FUN1
9	FUN2
10	FUN3

Thus, for example, YMIN(8) and YMAX(8) are the minimum and maximum values of FUN1.

If YMIN(I) $\neq$ YMAX(I) ($I = 1,6$), then Y(I) is not allowed to go outside these bounds. If YMIN(I) $\neq$ YMAX(I) ($I = 1, 10$), then these bounds are used to define the graph boundaries when the quantity with code number I is the abscissa or ordinate of a two-dimensional graph. If YMIN(I) = YMAX(I) ($I = 1, 10$), then the graph boundaries for the quantity with code number I are determined by the minimum and maximum values of that quantity which were actually attained during the integration. (In this case, the graph boundaries are automatically chosen to make the graph easily and conveniently readable.) Since, as is described later, the maximum number of points on a given frame of film can be determined by the user, a given graph may consist of more than one frame of film. The graph boundaries are the same on each frame belonging to the same graph. Exception: If T (code number 7) is plotted along the abscissa or ordinate and if YMIN(7) $\neq$ YMAX(7), then the values of T run consecutively from frame to frame if more than one frame is required to complete the graph. If YMIN(7) = YMAX(7), then, as before, the actual minimum and maximum values of T determine the graph boundaries on each of the frames which comprise the graph.

(3) (LABEL(I), I = 1, 10). FORMAT (10A8).

LABEL(I) is an eight-character label for the quantity with code number I. The labels are used on the listings and on the graphs. On stereoscopic graphs, the three axes are drawn and they are labeled 1, 2, and 3. The three LABEL(I) which are associated

with the three axes are indicated at the top of each stereo frame.

(4) IMON, ISTER, ISTB, NF. FORMAT (4I2).

IMON = 0, no two-dimensional graphs
 ≠ 0, two-dimensional graphs
 ISTER ≤ 0, no stereoscopic graphs
 > 0, stereoscopic graphs

If there are stereoscopic graphs, then ISTER (1 ≤ ISTER ≤ 10) equals the number of such graphs. (There may be more than one pair of frames per graph.)

ISTB = 0, CLIF = CN(6) = constant
 ≠ 0, CLIF calculated by subroutine CALLIF (See Section II C.)

CLIF is described in Reference 1.

NF = number of blank frames separating the left and right frames of a stereo pair (NF ≥ 0).

(5) (IAB(J), IORD(J), IBK(J), J = 1, 12).
 FORMAT (36I2).

These specify up to 12 different two-dimensional graphs. IAB(J) and IORD(J) are the code numbers of the quantities used for the abscissa and ordinate, respectively, on the Jth graph. There are IBK(J) × 101 points per frame on the Jth graph. The 101st point on a frame is the same as the 1st point on the succeeding frame of that graph. If IMON = 0, this card is omitted.

(6) ((DV(K,J), K = 1, 3), A(J), SF(J), FWF(J),
 ICHAR(J), ICON(J), J = 1, ISTER).
 FORMAT (1P6E12.7, 2I2).

These quantities are defined in the writeup of J513A; they are required by STEREO (the stereoscopic projection subroutine). If ISTER ≤ 0, these cards are omitted.

(7) (JX(K), JY(K), JZ(K), JBK(K), K = 1, 10).
 FORMAT (40I2).

These specify ISTER (≤ 10) stereoscopic graphs. The quantities used for the X, Y, and Z (1, 2, and 3) coordinates on the Kth graph are those with code numbers JX(K), JY(K), JZ(K), respectively. There are JBK(K) × 101 points per frame for the Kth graph. If ISTER ≤ 0, this card is omitted.

The above cards (items 1 - 7) are common to all problems and are read in only once. The following cards are read in for each problem. To run more than one problem at a time, the data cards are arranged in the following way: The first cards are

those described under items 1 - 7. Immediately following these cards come those described under items 8 and 9 for the first problem; then follow the cards described under items 8 and 9 for the second problem; and so forth.

(8) APLT. FORMAT (8A10/8A10).

The 80 characters on the two cards are written on the listing. The first 54 characters on the first card are written at the bottom of each graph frame.

(9) T, AEND, TMAX, IREV, IWRT, (Y(I), I = 1, 6),
 (CN(I), I = 1, 6).
 FORMAT (1P3E12.7, 2I6/(1P6E12.7)).

T = initial value of T

AEND = interval in T between output points

TMAX = final value of T

< 0, direction of integration reversed at TMAX and problem run to initial value of T

≥ 0, direction of integration not reversed at TMAX

IWRT = 0, no detailed listing of results
 ≠ 0, all results listed

Y(I) (I = 1, 6) are the initial values of Y(I).

CN(I) (I = 1, 6) are constants that may be supplied for use with the auxiliary subprograms (See Section II C).

B. Further Output

In addition to the output already mentioned, all the pertinent input values are listed, along with the initial and final values of EN, the final values of T, H(integration step size) and KSTEP (defined in Reference 1), the minimum and maximum values of H (HMI and HMA), the minimum and maximum values of CLIF (CLIFMI and CLIFMA), and the minimum and maximum values of the ten quantities that have code numbers (XMIN(I) and XMAX(I)). Certain comments are printed out depending on how the problem has progressed. One of these comments mentions a quantity called JUMP; it is defined in Reference 1.

C. Necessary Auxiliary Subprograms

These subroutines, which are supplied by the user, must be present. However, if they are not needed, then they may be dummy routines (i.e., in addition to any necessary dimension statement, consist of the two statements RETURN and END).

(1) Function subprograms VAL1(T,Y,CN), VAL2(T,Y,CN), and VAL3(T,Y,CN) which define FUN1, FUN2, and FUN3, respectively.

(2) SUBROUTINE ONCE.

This subroutine is used to perform a job (read in a table, calculate a constant of the motion, write something, etc.) once only, at the beginning of a set of problems (just after the stereoscopic information is read).

(3) SUBROUTINE REPEAT(T,Y,CN). This is called once for each problem (at the beginning of the problem, just after APLT is written).

(4) SUBROUTINE ENER(T,Y,CN,EN). This calculates the value of EN (for example, the energy).

(5) SUBROUTINE CALF(T,Y,CN,F). This calculates the derivatives F(I).

(6) SUBROUTINE CALLIF(T,Y,CN,CLIF). This calculates CLIF if ISTB $\neq$ 0.

(7) SUBROUTINE EXTRA(T,Y,CN,IX,IY). IX and IY are the code numbers for the abscissa and ordinate, respectively, of the two-dimensional graph currently being drawn when EXTRA is called. EXTRA can be used to plot or write additional information on the graph.

Communication among any of the auxiliary subroutines can be achieved with named common. The only restriction is that the name NC1 may not be used.

III. A SAMPLE PROBLEM

As an example we consider the equations of motion for a charged particle moving in a uniform, time-independent magnetic field when dimensionless canonical variables are used:

$$\frac{dx}{dt} = p_x + \frac{1}{2}y = F(1)$$

$$\frac{dy}{dt} = p_y - \frac{1}{2}x = F(2)$$

$$\frac{dz}{dt} = p_z = F(3)$$

$$\frac{dp_x}{dt} = \frac{1}{2}(p_y - \frac{1}{2}x) = F(4)$$

$$\frac{dp_y}{dt} = -\frac{1}{2}(p_x + \frac{1}{2}y) = F(5)$$

$$\frac{dp_z}{dt} = 0 = F(6)$$

In the subprograms listed in Appendix B, the following correspondence is made:

$$x = Y(1), \quad y = Y(2), \quad z = Y(3),$$

$$p_x = Y(4), \quad p_y = Y(5), \quad p_z = Y(6).$$

Data for this sample problem are given in Appendix C.

ACKNOWLEDGMENT

The author gratefully acknowledges the assistance of K. R. Crandall (Group MP-4) in formulating and writing Subroutine RECORD.

REFERENCES

1. H. R. Lewis, Jr., and E. J. Stovall, Jr., "A FORTRAN Version of Nordsieck's Scheme for the Numerical Integration of Differential Equations," Los Alamos Scientific Laboratory Report LA-3292 (March 1965).
2. H. R. Lewis, Jr., and E. J. Stovall, Jr., "Comments on a Floating-Point Version of Nordsieck's Scheme for the Numerical Integration of Differential Equations," Math. Comp. 21, 157 (1967).

APPENDIX A. Listings of ORB, RECORD, DER, AXES, and DEQ

```

PROGRAM ORB (INPUT,OUTPUT,FILM,TAPE10=INPUT,TAPE9=OUTPUT,
1TAPE12=FILM,TAPE13)
DIMENSION Y(6),F(6),TP(101),RP(101),ZP(101),OP(101),
1DRP(101),DZP(101),DOP(101),CN(6),FUN1(101),FUN2(101),
2FUN3(101)
COMMON/NC1/APLT(16),YMIN(10),YMAX(10),LABEL(10),IAB(12),
1IORD(12),IBK(12),JX(10),JY(10),JZ(10),JBK(10),ICHAR(10),
2ICON(10),DV(3,10),A(10),SF(10),FWF(10),XMIN(10),XMAX(10)
C
SSW 1 UP TO BEGIN. SSW 1 DOWN TO STOP.
TUPI=2.*3.1415926536
CALL ADV(10)
2001 READ(10,1001)HMIN,HMAX,FEE,ERROR,NEQ,MC,M1,M2,M3,IE
1001 FORMAT(1P4E12.7,6I1)
2002 READ(10,1002)(YMIN(I),YMAX(I),I=1,10)
1002 FORMAT(1P6E12.7)
2003 READ(10,1003)(LABEL(I),I=1,10)
1003 FORMAT(10A8)
2004 READ(10,1004)IMDN,ISTER,ISTB,NF
1004 FORMAT(4I2)
IF(IMCN)2005,2006,2005
2005 READ(10,1005)(IAB(I),IORD(I),IBK(I),I=1,12)
1005 FORMAT(36I2)
2006 IF(ISTER)2020,2020,2007
2007 READ(10,1006)((DV(K,J),K=1,3),A(J),SF(J),FWF(J),
1ICHAR(J),ICON(J),J=1,ISTER)
1006 FORMAT(1P6E12.7,2I2)
2010 READ(10,2011)(JX(K),JY(K),JZ(K),JBK(K),K=1,10)
2011 FORMAT(40I2)
C
T IS CURRENT TIME.
C
AEND IS TIME INTERVAL BETWEEN OUTPUT PCINTS.
C
TMAX IS THE TIME AT THE END OF INTEGRATION.
2020 CALL CNCE
HMAXO=HMAX
YMIN7=YMIN(7)
YMAX7=YMAX(7)
2008 READ(10,1007)APLT
1007 FORMAT(8A10/8A10)
1H1ELF,10199998,99999
99998 STOP
99999 HMAX=HMAXO
YMIN(7)=YMIN7
YMAX(7)=YMAX7
2009 READ(10,1008)T,AEND,TMAX,IREV, IWRT,
1(Y(I),I=1,6),(CN(I),I=1,6)
1008 FORMAT(1P3E12.7,2I6/(1P6E12.7))
WRITE(9,300C)APLT
3000 FORMAT(1H1,8A10/1H ,8A10)
CALL REPEAT(T,Y,CN)
WRITE(9,3001)
3001 FORMAT(1H ,5X,1H1,10X,7HYMIN(I),20X,7HYMAX(I))
WRITE(9,3002)(I,YMIN(I),YMAX(I),I=1,10)
3002 FORMAT(5X,12,5X,1PE16.7,11X,E16.7)
WRITE(9,3003)ERROR,FEE,IREV,NEQ,ISTB,IE,HMIN,HMAX,MO,M1,M2,M3,
1AEND,TMAX
3003 FORMAT(1H ,8HERROR = ,1PE12.4,5X,6HFEE = ,
11PE12.4,5X,7HIREV = ,12,5X,6HNEQ = ,11,5X,7HISTB = ,12,
25X,5HIE = ,12/1H ,7HHMIN = ,1PE12.4,
35X,7HHMAX = ,1PE12.4,10X,5HMO = ,12,5X,5HM1 = ,12,5X,5HM2 = ,12,
45X,5HM3 = ,12 /1H ,7HAEND = ,1PE12.4,5X,7HTMAX = ,1PE12.4)
C
EN=TOTAL ENERGY
EN=C.
CALL ENER(T,Y,CN,EN)
WRITE(9,3006)T,EN
3006 FORMAT(1H ,4HT = ,1PE16.7,10X,5HEN = ,E16.7)
WRITE(9,3007)
3007 FORMAT(1H ,4X,1H1,12X,4HY(I),22X,5HCN(I))
WRITE(9,3008)(I,Y(I),CN(I),I=1,6)

```

```

3008 FORMAT(5X,11,5X,1PE16.7,11X,E16.7)
      ENO=EN
      20 TF=T
      TMIX=TMAX-.1*AEND
      M=1
      ASSIGN 951 TO MM
      GO TO 900
900  IF(MO)901,902,901
901  TP(M)=AMOD((AMOD(T,TUPI))+TUPI),TUPI)
      GO TO 903
902  TP(M)=T
903  IF(M1)904,905,904
904  RP(M)=AMOD((AMOD(Y(1),TUPI))+TUPI),TUPI)
      GO TO 906
905  RP(M)=Y(1)
906  IF(M2)907,908,907
907  ZP(M)=AMOD((AMOD(Y(2),TUPI))+TUPI),TUPI)
      GO TO 909
908  ZP(M)=Y(2)
909  IF(M3)910,911,910
910  OP(M)=AMOD((AMOD(Y(3),TUPI))+TUPI),TUPI)
      GO TO 912
911  OP(M)=Y(3)
912  DRP(M)=Y(4)
      DZP(M)=Y(5)
      DOP(M)=Y(6)
      FUN1(M)=VAL1(T,Y,CN)
      FUN2(M)=VAL2(T,Y,CN)
      FUN3(M)=VAL3(T,Y,CN)
      GO TO MM,(951,952,32)
951  K=1
      REWIND 13
      TLIMIT=T +AEND
      N=2
      14 H=HMAX
      15 JUMP=-1
      HMI=1.0E+322
      HMA=0.0
      CLIFMI=1.0E+322
      CLIFMA=0.0
      DO 151 I=1,6
      XMIN(I)=Y(I)
      XMAX(I)=Y(I)
      151 CONTINUE
      XMIN(7)=T
      XMAX(7)=T
      DO 152 I=8,10
      XMIN(I)=1.0E+322
      XMAX(I)=-1.0E+322
      152 CONTINUE
      CALL DER (T,Y,CN,F,KCON,CLIF,ISTB)
      18 CALL SSWTCH(1,K000FX)
      GO TO(19,22),K000FX
      19 WRITE (9,21)
      21 FORMAT(47H8RUN TERMINATED BECAUSE SSW 1 IS DOWN. )
      GO TO 44
      22 CALL DEC(F,T,TLIMIT,Y,ERROR,NEQ,H,HMAX,JUMP,KSTEP,KCON,CLIF)
      99 IF(HMIN-ABS(H))23,23,107
      107 JUMP=-1
      23 IF(JUMP)24,26,28
      24 WRITE (9,25)T,KSTEP,H
      250FORMAT(27H8JUMP = -1.          TIME = 1PE15.7,13H      KSTEP = 16.
      19H      H = E15.7)
      ERRCR=10.0*ERROR
      IF(ERROR-0.1)102,102,101
      102 WRITE (9,103)ERROR
      103 FORMAT(49HMULTIPLY ERROR BY 10 AND CONTINUE. NEW ERROR = 1PE15.7
      1)
      106 T=TP(M)
      Y(1)=RP(M)
      Y(2)=ZP(M)
      Y(3)=OP(M)

```

```

      Y(4)=DRP(M)
      Y(5)=DZP(M)
      Y(6)=DOP(M)
      GO TO 14
101 GO TO 44
26 CALL DER (T,Y,CN,F,KCON,CLIF,ISTB)
      GO TO 18
28 M=M+1
      ASSIGN 952 TO MM
      GO TO 900
952 HMI=AMIN1(HMI,ABS(H))
      HMA=AMAX1(HMA,ABS(H))
      CLIFMI=AMIN1(CLIFMI,CLIF)
      CLIFMA=AMAX1(CLIFMA,CLIF)
      DO 953 I=1,6
      XMIN(I)=AMIN1(XMIN(I),Y(I))
      XMAX(I)=AMAX1(XMAX(I),Y(I))
953 CONTINUE
      XMIN(7)=AMIN1(XMIN(7),T)
      XMAX(7)=AMAX1(XMAX(7),T)
      XMIN(8)=AMIN1(XMIN(8),FUN1(M))
      XMAX(8)=AMAX1(XMAX(8),FUN1(M))
      XMIN(9)=AMIN1(XMIN(9),FUN2(M))
      XMAX(9)=AMAX1(XMAX(9),FUN2(M))
      XMIN(10)=AMIN1(XMIN(10),FUN3(M))
      XMAX(10)=AMAX1(XMAX(10),FUN3(M))
29 IF(101-M)30,30,32
30 WRITE(13)(RP(J),ZP(J),OP(J),DRP(J),DZP(J),DOP(J),
      ITP(J),FUN1(J),FUN2(J),FUN3(J),J=1,M)
31 K=K+1
      M=1
      ASSIGN 32 TO MM
      GO TO 900
32 DO 322 I=1,6
      IF(YMIN(I)-YMAX(I))4000,322,4000
4000 IF(I-3)4001,4001,320
4001 IF(I-2)4002,4003,4004
4002 IF(M1)322,320,322
4003 IF(M2)322,320,322
4004 IF(M3)322,320,322
320 IF(YMIN(I)-Y(I))321,321,33
321 IF(Y(I)-YMAX(I))322,322,33
322 CONTINUE
      GO TO 36
33 WRITE(9,34)T
34 FORMAT(24HEND OF RUNS. TIME = 1PE15.7)
      GO TO 44
36 IF(AEND)38,37,37
37 IF(TLIMIT-TMIX)41,39,39
38 IF(TMIX-TLIMIT)41,39,39
39 WRITE(9,40)
40 FORMAT(19H8INTEGRATION ENDED.)
      GO TO 44
41 IF(IE)82,411,82
411 CALL ENER(T,Y,CN,EN)
      IF(ABS(EN-ENO)-FEE*ENO)82,80,80
80 WRITE(9,81)
810FORMAT(73H8FRACTIONAL ERROR IN TOTAL ENERGY NOT LESS THAN FEE. PR
      IDBLEM TERMINATED.)
      GO TO 44
82 TLIMIT=TF+FLOAT(N)*AEND
      N=N+1
42 GO TO 18
44 WRITE(9,3200)HMI,HMA,CLIFMI,CLIFMA
3200 FORMAT(1H,6HHMI = ,1PE12.4,5X,6HHMA = ,1PE12.4,5X,9HCLIFMI = ,
      11PE12.4,5X,9HCLIFMA = ,1PE12.4///)
      WRITE(9,3004)
3004 FORMAT(1H,5X,1H1,10X,7HXMIN(I),20X,7HXMAX(I))
      WRITE(9,3002)(I,XMIN(I),XMAX(I),I=1,10)
      CALL ENER(T,Y,CN,EN)
      WRITE(9,3009)T,EN,H,KSTEP

```

```

3009 FORMAT(1H ,4HT = ,1PE16.7,10X,5HEN = ,
      1E16.7,10X,4HH = ,E16.7,10X,8HKSTEP = ,112)
      WRITE(9,3010)
3010 FORMAT(1H ,4X,1H1,12X,4HY(1))
      WRITE(9,3011)(I,Y(I),I=1,6)
3011 FORMAT(5X,11,5X,1PE16.7)
      IF(M-1)43,43,46
43 IF(K-1)49,49,45
45 KF=K-1
      MF=101
      GO TO 48
49 CALL SSWTCH(1,K000FX)
      GO TO(52,55),K000FX
46 WRITE(13)(RP(J),ZP(J),OP(J),DRP(J),DZP(J),DOP(J),
      1TP(J),FUN1(J),FUN2(J),FUN3(J),J=1,M)
47 KF=K
      MF=M
48 END FILE 13
      REWIND 13
      CALL RECORD(IWRT,IMON,ISTER,KF,MF,NF,T,Y,CN)
      CALL SSWTCH(1,K000FX)
      GO TO(50,53),K000FX
50 WRITE (9,51)
51 FORMAT(47H8RUN TERMINATED BECAUSE SSW 1 IS DOWN.
52 CALL EXIT
53 IF(IREV)54,55,55
54 IREV=-IREV
      TMAX=TF
      AEND=-AEND
      HMAX=-HMAX
58 T=TF(M)
      DET=YMAX(7)-YMIN(7)
      YMIN(7)=T
      YMAX(7)=T-DET
      Y(1)=RP(M)
      Y(2)=ZP(M)
      Y(3)=OP(M)
      Y(4)=DRP(M)
      Y(5)=DZP(M)
      Y(6)=DOP(M)
      GO TO 20
55 GO TO 2008
      END

```

```

SUBROUTINE RECORD(IWRT,IMON,ISTER,KF,MF,NF,T,Y,CN)
  DIMENSION B(101,10),IXY(4),XYZ(2,3),Y(6),CN(6)
  COMMON/NC1/APLT(16),YMIN(10),YMAX(10),LABEL(10),IAB(12),
  1 IORD(12),IBK(12),JX(10),JY(10),JZ(10),JBK(10),ICHAR(10),
  2 ICON(10),DV(3,10),A(10),SF(10),FWF(10),XMIN(10),XMAX(10)
  DATA(IXY(I),I=1,4)/120,980,50,910/
  L=0
  IF(IWRT)100,9,100
100 WRITE(9,101)APLT
101 FORMAT(1H1,15X,8A10/16X,8A10)
  1 WRITE(9,2)(LABEL(I),I=1,10)
  2 FORMAT(4H0 J,4X,10(A8,3X))
  DO 8 K=1,KF
    IF(KF-K)3,3,4
  3 M=MF
    GO TO 5
  4 M=100
  5 READ(13)((B(I,J),J=1,10),I=1,M)
    DO 7 I=1,M
      L=L+1
      WRITE(9,6)L,(B(I,J),J=1,10)
  6 FORMAT(15,1X,1P10E11.3)
  7 CONTINUE
  8 CONTINUE

```



```

9 REWIND 13
  IF( IMON)10,25,10
10 DO 24 K=1,12
  IF( IAB(K))25,25,11
11 IF( IORD(K))25,25,12
12 IX= IAB(K)
  IY= IORD(K)
  IF( YMAX(IX)-YMIN(IX))13,14,13
13 X1=YMIN(IX)
  X2=YMAX(IX)
  KX=13
  NX=10
  DX=X2-X1
  GO TO 15
14 X1=XMIN(IX)
  X2=XMAX(IX)
  CALL ASCL(4,X1,X2,NX,MX,KX)
  DX=0
15 IF( YMAX(IY)-YMIN(IY))16,17,16
16 Y1=YMIN(IY)
  Y2=YMAX(IY)
  KY=13
  NY=10
  DY=Y2-Y1
  GO TO 18
17 Y1=XMIN(IY)
  Y2=XMAX(IY)
  CALL ASCL(4,Y1,Y2,NY,MY,KY)
  DY=0
18 CALL ADV(1)
  IBS=IBK(K)
  IF( IBS)24,24,185
185 DO 23 K1=1,KF,IBS
  CALL DGA(120,980,50,910,X1,X2,Y2,Y1)
  CALL DLNLN(NX,NY)
  CALL SLLIN(NY,KY)
  CALL SBLIN(NX,KX)
  CALL WLCH(120,980,54,APLT,1)
  CALL WLCH(500,950,8,LABEL(IX),1)
  CALL WLCV(0,500,8,LABEL(IY),1)
  DO 22 IB=1,IBS
  IF(KF-(K1+IB-1))220,19,20
19 M=MF
  GO TO 21
20 M=101
21 REAC(13)((B(I,J),J=1,10),I=1,M)
22 CALL PLOT(M,B(1,IX),1,8(1,IY),1,42,1)
220 CALL EXTRA(T,Y,CN,IX,IY)
  IF(IX-7)222,221,222
221 X1=X1+DX
  X2=X2+DX
222 IF( IY-7)23,223,23
223 Y1=Y1+DY
  Y2=Y2+DY
23 CALL ADV(2)
24 REWIND 13
25 CALL ADV(15)
  IF( ISTER)26,42,26
26 DO 41 K=1,10
  IF( JX(K))41,41,27
27 IX=JX(K)
  IY=JY(K)
  IZ=JZ(K)
  IF( IY)41,41,28
28 IF( IZ)41,41,29
29 XYZ(1,1)=XMIN(IX)
  XYZ(2,1)=XMAX(IX)
  XYZ(1,2)=XMIN(IY)
  XYZ(2,2)=XMAX(IY)
  XYZ(1,3)=XMIN(IZ)
  XYZ(2,3)=XMAX(IZ)
  L=0

```

```

DO 30 I=1,2
DO 30 J=1,2
DO 30 M=1,2
L=L+1
B(L,1)=XYZ(I,1)
B(L,2)=XYZ(J,2)
30 B(L,3)=XYZ(M,3)
CALL STEREO(0,IXY,DV(1,K),A(K),SF(K),FWF(K),8,B(1,1),1,B(1,2),1,
1 B(1,3),1,0,0)
IBS=JBK(K)
IF(IBS)41,41,32
32 DO 40 K1=1,KF,IBS
IEYE=-1
N=0
DO 39 L=1,2
CALL WLCH(100,0,3,3H1=,1)
CALL WLCH(148,0,8,LABEL(IX),1)
CALL WLCH(212,0,5,5H 2=,1)
CALL WLCH(284,0,8,LABEL(IY),1)
CALL WLCH(348,0,5,5H 3=,1)
CALL WLCH(420,0,8,LABEL(IZ),1)
CALL FRAME(15,1005,15,1005)
CALL AXES(XYZ,IEYE)
CALL WLCH(0,1005,54,APLT,1)
DO 36 IB=1,IBS
IF(KF-(K1+IB-1))36,33,34
33 M=MF
GO TO 35
34 M=101
35 REAC(13)((B(I,J),J=1,10),I=1,M)
N=N+1
CALL STEREO(IEYE,1,1,1,1,1,M,B(1,IX),1,B(1,IY),1,B(1,IZ),1,
1 ICHAR(K),ICON(K))
36 CONTINUE
IF(IEYE)37,39,39
37 DO 38 IB=1,N
38 BACKSPACE 13
IEYE=1
39 CALL ADV(NF+1)
REWIND 13
40 CALL ADV(1)
41 CALL ADV(1)
42 CALL WLCH(480,500,3,3HEND,4)
CALL ADV(2)
RETURN
END

```

```

SUBROUTINE DER (T,Y,CN,F,KCON,CLIF,ISTB)
DIMENSION Y(6),CN(6),F(6)
2 CALL CALF(T,Y,CN,F)
IF(KCON)4,3,4
3 RETURN
4 IF(ISTB)6,5,6
5 CLIF=CN(6)
RETURN
6 CALL CALLIF(T,Y,CN,CLIF)
RETURN
END

```

```

SUBROUTINE AXES(XYZ,IE)
DIMENSION XYZ(2,3)
I=1
J=0
K=0
DO 1 M=1,3
CALL STEREO(IE,1,1,1,1,1,2,XYZ(1,1),1,XYZ(1,2),J,XYZ(1,3),K,1,-1)
L=K
K=J
J=I
1 I=L
CALL STEREO(IE,1,1,1,1,1,1,XYZ(2,1),0,XYZ(1,2),0,XYZ(1,3),0,1,0)
CALL STEREO(IE,1,1,1,1,1,1,XYZ(1,1),0,XYZ(2,2),0,XYZ(1,3),0,2,0)
CALL STEREO(IE,1,1,1,1,1,1,XYZ(1,1),0,XYZ(1,2),0,XYZ(2,3),0,3,0)
RETURN
END

```

```

SUBROUTINE DEQ(FF,XX,XLIMIT,YY,ERROR,NEQ,HH,HMAX,JUMP,KKSTEP,
IKCON,CCLIF)
DIMENSION STARTY(20),Y(20),SY(20),SAVEY(20),F(20),FP(20),
IDELTA(20),DALTA(20),A(20),B(20),C(20),D(20),
2AA(20),8B(20),CC(20),DD(20),SF(20),FF(20),YY(20)
NE = NEQ
H=HH
X=XX
KSTEP=KKSTEP
DO 1100 I=1,NE
F(I)=FF(I)
1100 Y(I)=YY(I)
CCLIF=CCLIF
IF(KSTEP-9999999999 1993,991,991
991 KSTEP=28
C TEST FOR TYPE OF ENTRY
993 IF(JUMP)1,998,999
998 GO TO 1A,(1000,11,21,802,803)
C JUMP POS. RESTORE VALUES
999 X=SAVEX
992 JUMP=0
DO 901 I=1,NE
F(I)=SF(I)
901 Y(I)=SAVEY(I)
GO TO 102
C JUMP NEG. INITIALIZE
1 DO 5 I=1,NE
STARTY(I)=Y(I)
A(I)=0.001
B(I)=0.001
C(I)=0.001
5 D(I)=0.001
KSTEP=0
KDELAY=0
KCON=0
T=95./(288.*64.)
U=863./(12.*5040.)
V=95.0/288.0
P=25.0/24.0
Q=35.0/72.0
R=5.0/48.0
S=1.0/120.0
ASSIGN 1000 TO 1A
JUMP=0
GO TO 1101
C BEGIN INTEGRATION STEP
1000 DO 1111 I=1,NE
SF(I)=F(I)
1111 SAVEY(I)=Y(I)

```

```

C      H TOO SMALL, RETURN WITH JUMP NEG.
600 IF(ABS(X+H)-ABS(X))605,601,605
601 JUMP=-1
    GO TO 1101
605 X=X+H
    DO 10 I=1,NE
      Y(I)=Y(I)+H*(F(I)+A(I)+B(I)+C(I)+D(I))
10  FP(I)=F(I)+2.0*A(I)+3.0*B(I)+4.0*C(I)+5.0*D(I)
    ASSIGN 11 TO IA
    GO TO 1101
11  DO 12 I=1,NE
12  SY(I)=Y(I)
    DO 20 I=1,NE
      DELTA (I)=F(I)-FP(I)
20  Y(I)=Y(I)+V*DELTA (I)*H
    ASSIGN 21 TO IA
    KCON=1
    GO TO 1101
21  KCON=0
    DO 30 I=1,NE
      DALTA (I)=F(I)-FP(I)
29  Y(I)=SY(I)+V*DALTA (I)*H
30  CONTINUE
C      TEST FOR STARTING SEQUENCE
31  IF(KSTEP-28)35,40,40
C      APPLY TEST 2 ON ZEROth STEP.
35  IF(KSTEP)50,50,60
C      HALVING TESTS
40  DO 45 I=1,NE
      IF(ABS(DALTA (I))-ERROR/ABS(H))45,45,55
45  CONTINUE
50  IF(V*H*CLIF-0.125)60,60,55
55  X=X-H
C      FAIL TESTS, HALVE H
223 H=H/2.0
    KDELAY=0
    DO 56 I=1,NE
      A(I)=A(I)/2.0
      B(I)=B(I)/4.0
      C(I)=C(I)/8.0
      F(I)=SF(I)
      Y(I)=SAVEY(I)
56  D(I)=D(I)/16.0
    GO TO 1000
C      PASS TESTS, CORRECT A,B,C,D
60  KSTEP=KSTEP+1
    DO 65 I=1,NE
      A(I)=A(I)+3.0*B(I)+6.0*C(I)+10.0*D(I)+P*DALTA (I)
62  B(I)=B(I)+4.0*C(I)+10.0*D(I)+Q*DALTA (I)
64  C(I)=C(I)+5.0*D(I)+R*DALTA (I)
67  D(I)=D(I)+S*DALTA (I)
65  CONTINUE
C      IF IN STARTING SEQUENCE, BRANCH
    IF(KSTEP-24) 70,90,100
700 GO TO (1000,1000,1000,74,1000,1000,1000,78,1000,1000,1000,74,1000,
      1100,1000,86,1000,1000,1000,74,1000,1000,1000),KSTEP
C      4TH, 12TH, 20TH STEP, GO BACK
74  H=-H
    DO 75 I=1,NE
      A(I)=-A(I)
75  C(I)=-C(I)
    GO TO 1000
C      8TH STEP, GO FORWARD
78  H=-H
    DO 79 I=1,NE
      Y(I)=STARTY(I)
      A(I)=-A(I)
79  C(I)=-C(I)
    GO TO 1000
C      16TH STEP, HALVE H, APPLY TEST 1
86  H=H/2.0
    DO 87 I=1,NE

```



```

      A(I)=A(I)/2.0
      B(I)=B(I)/4.0
      C(I)=C(I)/8.0
87  D(I)=D(I)/16.0
      DO 88 I=1,NE
      IF(ABS(DALTA (I))-ERROR/ABS(H))88,88,89
88  CONTINUE
C   PASS TEST  GO FORWARD WITH HALVED H
      GO TO 78
C   FAIL TEST  BEGIN AGAIN WITH HALVED H
89  H=-H
      DO 92 I=1,NE
89  Y(I)=STARTY(I)
      GO TO 1
C   24TH STEP, DOUBLE H, STARTING SEQUENCE ENDS
90  H=H*2.0
      DO 91 I=1,NE
      A(I)=A(I)*2.0
      B(I)=B(I)*4.0
      C(I)=C(I)*8.0
91  D(I)=D(I)*16.0
      GO TO 78
100 KDELAY =KDELAY+1
C   WILL NEXT STEP MOVE PAST XLIMIT
102 IF(ABS(XLIMIT-X)-ABS(H))103,103,110
C   YES.....SAVE X AND Y, INTEGRATE TO XLIMIT, RETURN.
103 ENDF=XLIMIT-X
      DO 105 I=1,NE
      AA(I)=ENDH*A(I)/H
      BB(I)=ENDH**2*B(I)/H**2
      CC(I)=ENDH**3*C(I)/H**3
105 DD(I)=ENDH**4*D(I)/H**4
      SAVEX=X
      DO 800 I=1,NE
      SF(I)=F(I)
800 SAVEY(I)=Y(I)
801 X=XLIMIT
      DO 106 I=1,NE
      Y(I)=Y(I)+ENDH*(F(I)+AA(I)+BB(I)+CC(I)+DD(I))
106 FP(I)=F(I)+2.0*AA(I)+3.0*BB(I)+4.0*CC(I)+5.0*DD(I)
      ASSIGN 802 TO 1A
      GO TO 1101
802 DO 805 I=1,NE
805 SY(I)=Y(I)
      DO 107 I=1,NE
      DELTA (I)=F(I)-FP(I)
107 Y(I)=Y(I)+V*DELTA (I)*ENDH
      ASSIGN 803 TO 1A
      GO TO 1101
803 DO 100 I=1,NE
      DALTA (I)=F(I)-FP(I)
108 Y(I)=SY(I)+V*DALTA (I)*ENDH
      JUMP=1
      GO TO 1101
C   NO.....TEST FOR DOUBLING. IF OK, BEGIN NEXT STEP AFTER DOUBLING
110 IF(ABS(XLIMIT-X)-ABS(2.0*H))1000,1000,111
111 IF(KDELAY-4)1000,120,120
120 IF(ABS(2.0*H)-ABS(HMAX))121,121,1000
121 DO 125 I=1,NE
      IF(ABS(DALTA (I))-ERROR/(128.0*ABS(H))125,125,1000
125 CONTINUE
      IF(V*H*CLIF-0.0625)130,1000,1000
130 CONTINUE
335 H=2.0*H
      DO 135 I=1,NE

```

```
A(I)=2.0*A(I)
B(I)=4.0*B(I)
C(I)=8.0*C(I)
135 D(I)=16.0*D(I)
KDELAY=0
GO TO 1000
1101 NEQ=NE
HH=H
XX=X
KKSTEP=KSTEP
DO 1102 I=1,NE
FF(I)=F(I)
1102 YY(I)=Y(I)
CCLIF=CLIF
RETURN
END
```

APPENDIX B. Listings of Suitable Auxiliary Subprograms for the Sample Problem

```

FUNCTION VAL1(T,Y,CN)
DIMENSION Y(6),CN(6)
VAL1=0.0
RETURN
END

```

```

FUNCTION VAL2(T,Y,CN)
DIMENSION Y(6),CN(6)
VAL2=0.0
RETURN
END

```

```

FUNCTION VAL3(T,Y,CN)
DIMENSION Y(6),CN(6)
VAL3=0.0
RETURN
END

```

```

SUBROUTINE ONCE
RETURN
END

```

```

SUBROUTINE REPEAT(T,Y,CN)
DIMENSION Y(6),CN(6)
RETURN
END

```

```

SUBROUTINE EXTRA(T,Y,CN,IX,IY)
DIMENSION Y(6),CN(6)
RETURN
END

```

```

SUBROUTINE ENER(T,Y,CN,EN)
DIMENSION Y(6),CN(6)
ENERGY OF PARTICLE IN UNIFORM B FIELD IN DIMENSIONLESS CANONICAL VARIABLES
EN=.5*((Y(4)-.5*Y(2))**2+(Y(5)+.5*Y(1))**2+Y(6)**2)
RETURN
END

```

```

SUBROUTINE CALF(T,Y,CN,F)
DIMENSION Y(6),CN(6),F(6)
F(1)=(Y(4)+.5*Y(2))
F(2)=(Y(5)-.5*Y(1))
F(3)=Y(6)
F(4)=.5*(Y(5)-.5*Y(1))
F(5)=-.5*(Y(4)+.5*Y(2))
F(6)=0.
RETURN
END

```

```

SUBROUTINE CALLIF(T,Y,CN,CLIF)
DIMENSION Y(6),CN(6)
RETURN
END

```

APPENDIX C. Data for the Sample Problem

.001	.5	1.	-06 1.	-06600000			
-1.25	1.25	-1.25	1.25	+1.	+1.		
0.	0.	0.	0.	0.	0.		
0.	25.	0.	0.	0.	0.		
0.	0.						
X	Y	Z	PX	PY	PZ	TIME	RENATE BEATA ANNETTE
1 1 0 1							
7 1 2 7 2 2	7 3 2 7 4 2	7 5 2 1 2 2 3	1 2 1 4 2				
1.	1.	1.	0.	1.	1.	42 1	
1 2 3 2							
PARTICLE IN UNIFORM B FIELD. DIMENSIONLESS CAN. VARS.							
TEST OF ORB.							
0.	.125	30.	-1	1			
1.	0.	0.	0.	-.5	.75		
.0	.0	.0	.0	.0	1.		