COMPUTER PROGRAMS FOR CERTAIN PROBLEMS IN ELECTRIC AND MAGNETIC FIELDS AND ELECTRON STREAMS IN TWO-DIMENSIONAL SYSTEMS

by

R. A. Rao and K. A. Fox

Memorandum ERL-M-255

9 September 1968

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

TABLE OF CONTENTS

ABSTRACT

Computer programs are presented which: (1) solve Laplace's equation with tangential and normal fields specified along an open boundary (the Cauchy problem); (2) solve Laplace's equation with the potential specified on a closed curve (the Dirichlet problem); (3) solve the first order paraxial ray equation for a crossed-field sheet beam in a uniform magnetic field. A chapter is devoted to each problem. Each chapter includes a discussion of the problem, a brief description of the computer programs involved, an explanation of the input variables, the arrangement of the data cards, and a listing of the programs in FORTRAN IV.

ACKNOWLEDGMENT

The authors wish to acknowledge the help and encouragement of Prof. T. Van Duzer without whose help this work would not have materialized. Thanks are also due to Mrs. Gail Venti and Mrs. Lannon Leiman for their cheerful help in debugging the programs. The program on the solution of Laplace's equation for Dirichlet boundary conditions has incorporated a substantial part of the trajectory tracing program written by P. T. Kirstein and J. S. Hornsby at CERN, Geneva to whom the authors are indebted. This work was supported by the U.S. Army Electronics Command, Fort Monmouth, New Jersey, under contract DA 36-039 AMC-02164(E) and DAAB07-67-C-0227.

I. Introduction

In the course of the work on the design of electron guns for crossed-field amplifiers, the authors have had to solve a number of electrostatic and magnetostatic problems as well as problems of space-charge flow by numerical methods. Computer programs were prepared in FORTRAN IV for use in the CDC 6400 at the University of California at Berkeley. Though the project as a whole was motivated by and directed towards the primary objective of synthesizing crossed-field electron guns, some of the programs that were written for this purpose may be used in a wider range of problems in physics and electrical engineering. This report was prepared to make available some of the programs which find use in a wide range of problems.

This report contains the following programs:

(1) Solution of Laplace's equation with the tangential and normal derivatives specified on an open boundary (the Cauchy problem). This program calculates the potentials and fields everywhere in the plane when the tangential and normal derivatives of potential are specified on an open curve. There are four programs listed, each suitable for different kinds of input information. (Two-dimensional rectangular coordinates.)

(2) Solution of Laplace's equation with the potential prescribed on a closed curve (the Dirichlet problem). This program calculates the potentials and fields inside a closed region when the potential is prescribed on the bounding curves. This is suitable for both rectangular and cylindrical geometry and was, for the most part, taken from a larger program written by Kirstein and Hornsby[1] for determining trajectories in electron guns.

(3) Solution of the paraxial ray equation with space charge for a sheet beam. Calculations can include a uniform magnetic field perpendicular to the plane of the trajectories. The first-order paraxial ray equation involves three parameters: the beam thickness, the axial potential, and the axial curvature. When two of these parameters are specified, the third may be determined by solving the paraxial ray equation. Accordingly there are three programs listed here.

The programs are divided into a number of subprograms so that parts of the program may be easily modified or used separately. In subsequent chapters, these programs are discussed in more detail, indicating the type of situations in which they may be useful. The appendices give a brief discussion of the various formats used for input, the arrangement of cards in the program decks and Fortran listings of the source programs.

II. Solution of the Cauchy problem for Laplace's equation in two dimensions

Laplace's equation appears frequently in many areas of physics and electrical engineering. In the Cauchy problem the potential and its normal derivative (or the tangential and normal derivatives of potential) are specified on an open curve. The problem is to determine the potential and its derivatives everywhere in the plane. The problem arises, for example, when one has to design pole pieces to produce a specified variation of the magnetic field along a given curve. Another example is the problem of designing electrodes for electron guns where one is required to produce specified potential and its normal derivative along a curve representing the edge of an electron beam. Take, for example, the problem of designing electrodes to produce a rectilinear sheet beam. We know from the Langmuir[2] flow solution that if we can produce a potential which varies as $x^{4/3}$ and the normal derivative of potential which is zero along the straight line representing the edge of a sheet beam, then we can indeed obtain such a beam. We determine the electrodes by solving Laplace's equation outside the beam edge with the potential and its normal derivative specified on the beam edge. The result is the well-known Pierce[3] gun which is shown in Fig. 1. In more complicated electron gun design problems, the beam may be curved and then one must solve Laplace's equation with Cauchy conditions specified along a curve. The method used to solve this problem has been described by Kirstein.[4] We first transform the problem into a plane where the boundary is one of the axes of the coordinate system. Then we use analytic continuation of a complex potential function to determine the equipotentials in this plane. Finally, we transform the equipotentials back into the original plane of the problem. This is illustrated in Fig. 2.

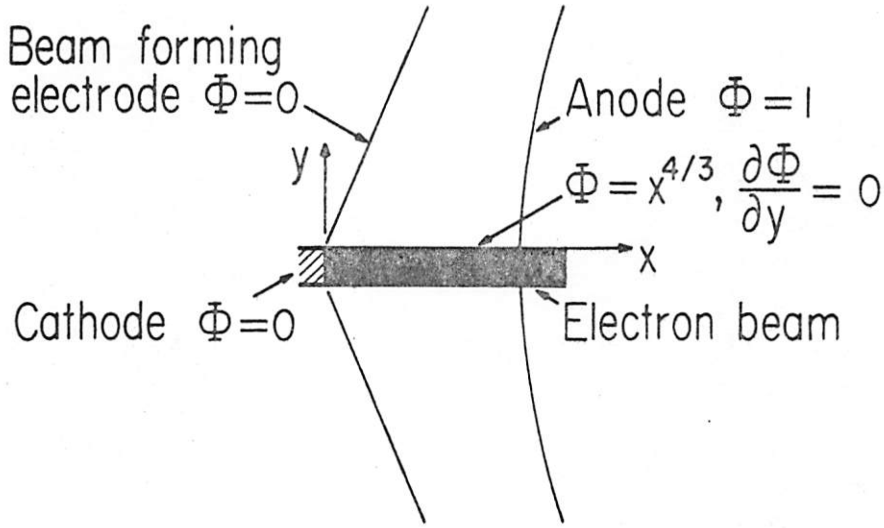

Fig. 1: Pierce Gun

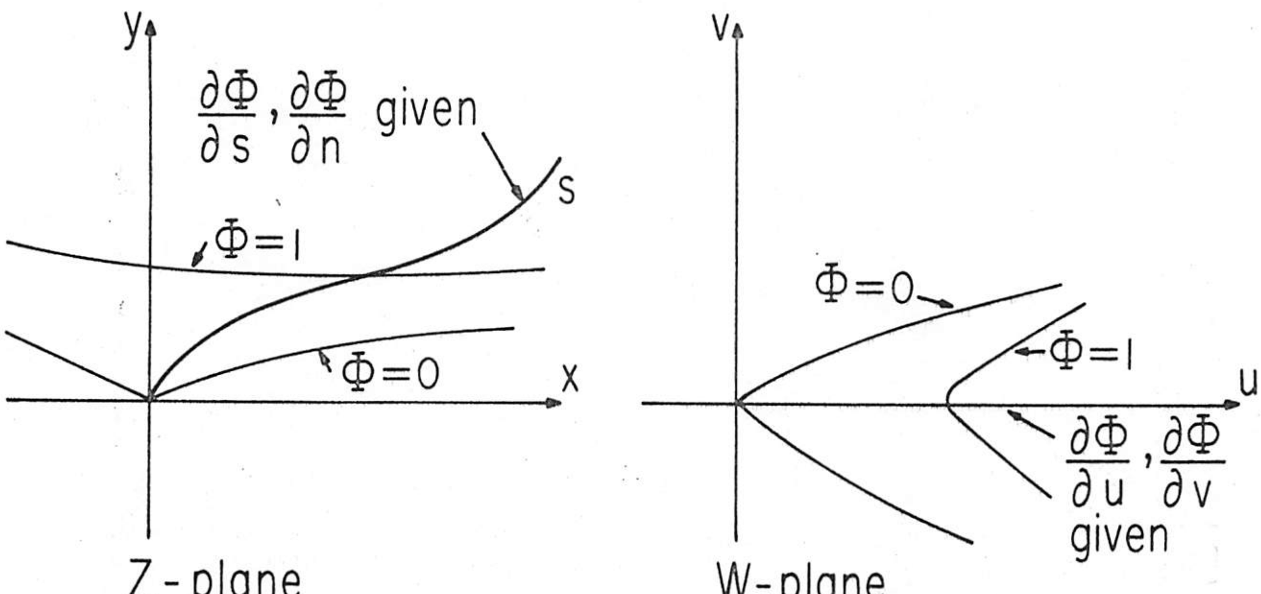

Fig. 2: Cauchy Problem For Curved Boundary

To apply this method, the shape of the curve and the tangential and normal derivatives of potential along this curve must be given as analytic functions of some parameter along the curve. One frequently has to solve a problem in which the boundary conditions are known at a large number of discrete points along the curve. In these programs these quantities are first approximated by polynomials using a least squares fit. Provision is made for polynomials up to degree thirty. It is very easy to change the programs to accommodate higher degree polynomials. In problems where the required accuracy of the fit should be higher along some parts of the boundary than along other parts, weighting functions may be used to achieve this end. The equipotentials calculated by this method are the correct equipotentials to produce fields and potentials corresponding to the polynomial approximations used here. In order to get an estimate of the accuracy of the approximations, the coordinates of the originally prescribed curve, the tangential and normal derivatives of potential along this curve, their respective polynomial approximations and the errors in the polynomial approximation at each of the prescribed points on the given curve are printed out. If the approximations are considered not accurate enough, the accuracies may be increased by modifying the weighting function and increasing the degree of the polynomial. It has been the experience of the authors that the errors can be made sufficiently small for engineering purposes. To check the results, one may solve the inverse problem of calculating the fields on the given curve from the values of potential obtained on a closed boundary enclosing the curve. This can be done by using the programs for the Dirichlet problem discussed in the next chapter. Since we have used only the derivatives of the potential and not the potential itself, the values of the potentials printed out in the output may be wrong by a constant additive constant throughout the plane. In these programs the constant has been chosen such that the potential is zero at S = 0 on the given curve (S is the arc length along the curve). First, a program 'RAODP' is used to obtain the polynomial fit. Then one of two different programs, RA01 and RA02, may be used for the solution of the Cauchy problem. In the following pages short descriptions of these programs along with the explanation of variable names and information about preparing the data cards are given.

Name of the Main Program: RAODP

Source Language: Fortran IV

Purpose: To calculate polynomial approximations using the weighted least-squares method.

Comments: Values of a set of functions and the weights to be used for the polynomial approximations are specified at equal intervals of the independent variable S. Typically, they may be the coordinates and the tangential and normal derivatives of potential along a beam edge specified at equal intervals of arc length along the beam edge. The program uses two code numbers, NCODE1 and NCODE2, both of which may take on values 0 and 1. The following choices are possible.

NCODE1 = 0 NCODE2 = 0, when one function and one set of weights are used.

NCODE1 = 0 NCODE2 = 1, when many functions have to be approximated, but all of them use the same set of weights.

NCODE1 = 1 NCODE2 = 0, when one function has to be approximated and many different polynomial approximations have to be calculated using different weighting functions.

NCODE1 = 1 NCODE2 = 1, when many different functions have to be approximated, and each function uses a different weighting function.

Double precision arithmetic is used for the calculations, but the input and the output have been chosen to have only 8 significant digits. The output will be the polynomial coefficients. The first coefficient is the constant term and the last coefficient is the highest degree term. In addition, the values of the function (given) and the values of the polynomial approximation, and the error in the approximation are printed out at all points along the S axis (independent variable).

Explanation of variable names appearing in the input:

BCD(I) = Any comment (using alphabets and numbers) less than 78 characters in length which one wants to be printed out at the beginning of the output. These comments may be used to facilitate identification and understanding of the output when the program is used many times.

S(1) = The value of the independent variable for the first (left-most) point on the curve.

DS = Increment in S defined by S(I + 1) = S(I) + DS.

SCALE = A scale factor for the independent variable S. Generally SCALE is chosen to be 1.0. However, when the maximum value of S is large, SCALE should be chosen to be greater than 1.0 to avoid overflows (i.e., quantities exceeding 10^{38} during the computation). The criterion for choosing SCALE is to make $(S_{max}/SCALE)^{2M+1}$ less than 10^{38}.

M = Degree of the polynomial. M should be less than or equal to 30.

N = Number of points along the S axis at which the values of the function and the weights are specified. N should be less than 500.

NUM = Number of functions to be approximated at one time. NUM should be less than 10.

NCODE1 = Code number for weights. It should be 0 or 1 as explained in 'comments.'

NCODE2 = Code number for the functions. See 'comments' above.

W(I) = Weights at consecutive points along the S axis. See 'comments' above.

Y(I) = Values of the function at consecutive points along the S axis. See 'comments' above.

The following programs should be in the deck:

RAODP (main program)

WEIGHT

POLDP

CHKDP

DATA CARDS

Column Number		10	20	30		40		50		60	70	80
Card Sequence	Format											
1	13A6	Any comment occupying less than 78 columns for identification purposes										
2	(3F10.5,5I5)	S(1)	DS	SCALE	M	N	NUM	NCODE1	NCODE2			
3	8F10.5	W(1)	W(2)	W(3)	......							
4	5E15.8	Y(1)	Y(2)	Y(3)	......							

Notes: (i) NCODE1 = 0, NCODE2 = 0: One set of weights in sequence 3 followed by one set of Y in sequence 4.

(ii) NCODE1 = 0, NCODE2 = 1: One set of weights followed by several sets of Y.

(iii) NCODE1 = 1, NCODE2 = 0: One set of weights, one set of Y, remaining sets of weights.

(iv) NCODE1 = 1, NCODE2 = 1: One set of weights, one set of Y, repeated several times.

```
      FORTRAN IV PROGRAM RAODP(INPUT,OUTPUT,PUNCH)
C     NCODE1 IS A CODE FOR WEIGHTING. IF NCODE1=0, THE SAME WEIGHTING
C     FUNCTION IS USED FOR ALL CURVES.   NCODE2 IS A CODE FOR CURVE
C     DATA. IF NCODE2=0, ONLY ONE CURVE IS FITTED WITH DIFFERENT WEIGHTS
       DOUBLE PRECISION S,Y,W,SCALE,DS,A
      DIMENSION S(500),Y(500),W(500),A(32),BCD(50)
      READ 5,(BCD(I),I=1,13)
    5 FORMAT(13A6)
      PRINT 10,(BCD(I),I=1,13)
   10 FORMAT(1H1,13A6)
      READ 15,S(1),DS,SCALE,M,N,NUM,NCODE1,NCODE2
   15 FORMAT(3D10.5,5I5)
      PRINT 20,DS,SCALE
   20 FORMAT(6H-DS =  D10.3,13H      SCALE =  D10.3)
      DO 25 I=2,N
   25 S(I)=S(I-1)+DS
      M1=M+1
      DO 60 J=1,NUM
      IF(NCODE1)31,31,32
   31 IF(J-1)32,32,33
   32 CALL WEIGHT(W,S,N)
      PRINT 35,(W(I),I=1,N)
   35 FORMAT(2H-W/(8D15.5))
   33 IF(NCODE2)41,41,42
   41 IF(J-1)42,42,43
   42 READ 40,(Y(I),I=1,N)
   40 FORMAT(5D15.8)
   43 CALL POLDP(S,Y,W,M,N,SCALE,A)
      PRINT 45,J,(A(I),I=1,M1)
   45 FORMAT(2H-A,I1/(4D20.8))
      PUNCH 50,(A(I),I=1,M1)
   50 FORMAT(4D15.8)
      CALL CHKDP(S,Y,M,N,A,DS)
   60 CONTINUE
      STOP
      END
```

```
      SUBROUTINE WEIGHT(W,S,N)
      DOUBLE PRECISION W,S
      DIMENSION S(500),W(500)
C   THIS CALCULATES OR READS THE WEIGHTS AT THE N SAMPLING POINTS
      READ 10,(W(I),I=1,N)
   10 FORMAT(8D10.5)
      RETURN
      END
```

```
      SUBROUTINE POLDP(S,Y,W,M,N,SCALE,A)
C     THIS SUBROUTINE CALCULATES THE POLYNOMIAL COEFFICIENTS FOR THE
C     BEAM EDGE QUANTITIES BY LEAST SQUARES APPROXIMATION.
      DOUBLE PRECISION S,X,Y,W,SUM,V,A,B,SCALE,P,DIVB,FMULTB,SIGMA
      DIMENSION S(500),X(500),Y(500),W(500),SUM(63),V(62),B(32,33),A(32)
      LS=2*M+1
      LB=M+2
      LV=M+1
      DO 4 I=1,N
    4 X(I)=S(I)/SCALE
      DO 6 J=1,LS
    6 SUM(J)=0.
      DO 8 I=1,N
    8 SUM(1)=SUM(1)+W(I)
      DO 10 J=1,LV
   10 V(J)=0.
      DO 16 I=1,N
      P=W(I)
      V(1)=V(1)+Y(I)*W(I)
      DO 13 J=2,LV
      P=X(I)*P
      SUM(J)=SUM(J)+P
   13 V(J)=V(J)+Y(I)*P
      DO 16 J=LB,LS
      P=X(I)*P
   16 SUM(J)=SUM(J)+P
   17 DO 20 I=1,LV
      DO 20 K=1,LV
      J=K+I
   20 B(K,I)=SUM(J-1)
      DO 22 K=1,LV
   22 B(K,LB)=V(K)
   23 DO 31 L=1,LV
      DIVB=B(L,L)
      DO 26 J=L,LB
   26 B(L,J)=B(L,J)/DIVB
      I1=L+1
      IF(I1-LB) 28,33,33
   28 DO 31 I=I1,LV
      FMULTB=B(I,L)
      DO 31 J=L,LB
   31 B(I,J)=B(I,J)-B(L,J)*FMULTB
   33 A(LV)=B(LV,LB)
      I=LV
   35 SIGMA=0.0
      DO 37 J=I,LV
   37 SIGMA=SIGMA+B(I-1,J)*A(J)
      I=I-1
      A(I)=B(I,LB)-SIGMA
   40 IF(I-1) 41,41,35
   41 DO 42 J=1,LV
   42 A(J)=A(J)/SCALE**(J-1)
      RETURN
      END
```

```
      SUBROUTINE CHKDP(S,Y,M,N,A)
      DOUBLE PRECISION S,Y,A,SY,POLY,ERROR
      DIMENSION S(500),Y(500),A(32),POLY(500),ERROR(500)
      M1=M+1
      DO 10 I=1,N
      SY=0.
      DO 5 K=2,M1
    5 SY=SY+A(K)*S(I)**(K-1)
      POLY(I)=SY+A(1)
   10 ERROR(I)=POLY(I)-Y(I)
      PRINT 15
   15 FORMAT(1H1,10X,1HS,25X,1HY,23X,4HPOLY,23X,5HERROR)
      PRINT 20,(S(I),Y(I),POLY(I),ERROR(I),I=1,N)
   20 FORMAT(4D25.8)
      RETURN
      END
```

Name of the main program: RA01

Source Language: FORTRAN IV

Purpose: To calculate equipotentials starting from the polynomial approximations of the tangential and the normal derivatives of the potential along the beam edge. The beam edge is assumed to be straight and is referred to as the U axis.

Comments: The polynomial coefficients for the tangential and the normal derivatives of the potential along the beam edge should be given. Since the polynomials prescribe only the derivatives of the potential, a constant of integration is needed to define the value of an equipotential. This constant is calculated and used in this program when the coordinates and the potential of a point in the U-V plane are given. The output will consist of the value of the equipotential, the coordinates along the equipotential and the slope of the equipotential curve. The value of the equipotential is calculated from equation (19) of Kirstein's paper[4] by taking the real part of W(w). The equipotential contour is found by integrating Kirstein's equation (24). The accuracy of this integration depends in part on the incremental distance for integration DELU. If DELU is chosen too large, the potential will change its value along the curve. The integration step size DELU should then be diminished until the potential variation along the curve is within desired limits.

The following programs should be in the deck:

RA01

EQPOT1

Explanation of variable names in the data and the output:

BCD(I) = Any comment (78 columns long) to help identify the problem and the results.

M = Degree of the polynomial (same for both the tangential and the normal derivative of the potential along the beam edge).

NUM = Number of equipotentials required.

NMAX = The maximum number of points on each equipotential curve.

U1, V1 = Coordinates of a point in the U-V plane.

POT = Potential at the point (U1, V1) in the U-V plane.

DD(I) = Coefficients of the polynomial approximation to the tangential derivative of the potential. The first coefficient is the constant term of the polynomial.

EE(I) = Coefficients of the polynomial approximation to the normal derivative of the potential.

U, V = Coordinates of the starting point on the desired equipotential.

UMIN, UMAX
VMIN, VMAX = Limits of the region in the U-V plane within which the equipotentials are required.

DELU = Incremental distance for integration in the U-V plane. DELU > 0 calculates the equipotential in the direction of increasing U at the starting point and vice versa. DELU may be chosen to be roughly equal to the ratio (length of the equipotential line/NMAX).

DATA CARDS

Column Number		10	20	30	40	50	60	70	80
Card Sequence	Format	Variable							
1	13A6	Any comment occupying less than 78 columns							
2	3I10	M	NUM	NMAX					
3	3F10.5	U1	V1	POT					
4	4E15.8	DD(1)	DD(2)	....					
		....							
5	4E15.8	EE(1)	EE(2)	....					
		....							
6a	7F10.5	U	V	UMIN	UMAX	VMIN	VMAX	DELU	
6b		....							

Note: Each of the data cards 6a, 6b, ... refers to one equipotential. There should be as many of these cards as the number (NUM) of equipotentials required.

```
      FORTRAN IV PROGRAM RA01(INPUT,OUTPUT)
C   THIS IS THE CALLING PROGRAM FOR EQUIPOTENTIAL CALCULATION IN THE U-V
C   PLANE.
      COMPLEX FF,W,CLXPOT
      DIMENSION DD(32),EE(32),FF(32),BCD(15)
      READ 5, (BCD(I),I=1,13)
    5 FORMAT(13A6)
      PRINT 10, (BCD(I),I=1,13)
   10 FORMAT(1H1,13A6)
      READ 15, M,NUM,NMAX
   15 FORMAT(3I10)
      M1=M+1
      READ 20, U1,V1,POT
   20 FORMAT(3F10.5)
      READ 25, (DD(I),I=1,M1)
      READ 25, (EE(I),I=1,M1)
   25 FORMAT(4E15.8)
      DO 30 I=1,M1
   30 FF(I)=CMPLX(DD(I),-EE(I))
      W=CMPLX(U1,V1)
      CLXPOT=(0.,0.)
      DO 35 I=1,M1
   35 CLXPOT=CLXPOT+FF(I)*W**I/FLOAT(I)
      CONST=POT-REAL(CLXPOT)
      DO 70 I=1,NUM
      READ 55,U,V,UMIN,UMAX,VMIN,VMAX,DELU
   55 FORMAT(7F10.5)
      PRINT 60,UMIN,UMAX,VMIN,VMAX,DELU
   60 FORMAT(11H-    UMIN =  F10.3,10H    UMAX =  F10.3,10H    VMIN =  F10.
     X3,10H    VMAX =  F10.3,10H    DELU =  F10.3)
      CALL EQPOT1(M,NMAX,FF,CONST,U,V,UMIN,UMAX,VMIN,VMAX,DELU)
   70 CONTINUE
      STOP
      END
```

```
      SUBROUTINE EQPOT1(M,NMAX,FF,CONST,U,V,UMIN,UMAX,VMIN,VMAX,DELU)
      COMPLEX FF,W,DW,CLXPOT
      DIMENSION FF(32),SGN(2)
      M1=M+1
      DELUM=ABS(DELU)
      SU=DELU/DELUM
      PRINT 10
   10 FORMAT(11H-POTENTIAL 10X,5HDV/DU 11X,1HU 13X,1HV)
      II=1
      LL=1
   15 W=CMPLX(U,V)
      DW=(0.,0.)
      DO 20 J=2,M1
      L=J-1
   20 DW=DW+FF(J)*W**L
      DW=DW+FF(1)
      DV=REAL(DW)/AIMAG(DW)
      DVA=ABS(DV)
      SGN(II)=DV/DVA
      IF (II-1) 21,21,22
   21 II=2
      GO TO 23
   22 II=1
   23 CLXPOT=CMPLX(CONST,0.)
      DO 25 J=1,M1
   25 CLXPOT=CLXPOT+FF(J)*W**J/FLOAT(J)
      POT=REAL(CLXPOT)
      PRINT 30,POT,DV,U,V
   30 FORMAT(4E15.5)
      IF(LL.LE.1) GO TO 35
      SM=SGN(1)*SGN(2)
      IF(SM.LT.0..AND.DVA.GT.1.)SU=-SU
      IF(SM.LT.0..AND.DVA.LE.1.)SV=-SV
      GO TO 40
   35 SV=SU*SGN(1)
   40 LL=LL+1
      IF(DVA.GT.1.) GO TO 45
      U=U+SU*DELUM
      V=V+SU*DELUM*DV
      GO TO 50
   45 V=V+SV*DELU
      U=U+SV*DELU/DV
   50 IF (LL.GT.NMAX) GO TO 55
      IF(U.LT.UMIN.OR.U.GT.UMAX)GO TO 55
      IF(V.LT.VMIN.OR.V.GT.VMAX)GO TO 55
      GO TO 15
   55 RETURN
      END
```

Name of the main program: RA02

Source language: FORTRAN IV

Purpose: To calculate the equipotentials when the polynomial approximations to the coordinates and the tangential and the normal derivatives of the potential along a curved beam edge are given.

Comments: Before this program may be used, the cartesian coordinates X and Y and the tangential and the normal derivatives of potential along the beam edge should be approximated by polynomials in arc length U along the beam edge. Since only the derivatives of the potential are given, a constant of integration is needed to define the value of an equipotential. This constant is calculated in the program when the coordinates (U1, V1) and the potential (POT) of a point in the U-V plane are given. The point may be conveniently chosen on the beam edge in which case V1 = 0. The output will consist of the value of the equipotential, the slope of the equipotential in the U-V plane and the coordinates of the equipotential in the U-V and the X-Y planes.

The following programs should be in the deck:

RA02

EQPOT2

Explanation of variables in the data and the output:

BCD(I) = Any comment (78 columns long) to help identify the problem and the results.

M = Degree of the polynomial (same for all the polynomial approximations).

NUM = Number of equipotentials required.

NMAX = The maximum number of points on each equipotential curve.

U1, V1 = Coordinates of a point in the U-V plane.

POT = Potential of the point (U1, V1) in the U-V plane.

AA(I) = Coefficients of the polynomial approximation to the X-coordinate along the beam edge. The first coefficient is the constant term of the polynomial.

BB(I) = Coefficients of the polynomial approximation to the y-coordinate along the beam edge.

DD(I) = Coefficients of the polynomial approximation to the tangential derivative of the potential along the beam edge.

EE(I) = Coefficients of the polynomial approximation to the normal derivative of the potential along the beam edge.

U, V = Coordinates of the starting point on the desired equipotential.

XMIN, XMAX
YMIN, YMAX = Limits of the region in the U-V plane within which the equipotentials are required.

DELU = Incremental distance for integration in the U-V plane. DELU > 0 calculates the equipotential in the direction of increasing U at the starting point and vice versa. DELU may be chosen to be roughly equal to the ratio (length of the equipotential line/NMAX).

DATA CARDS

Card Sequence	Format	10	20	30	40	50	60	70	80
			variables						
1	13A6	Any comment occupying less than 78 columns							
2	3I10	M	NUM	NMAX					
3	3F10.5	U1	V1	POT					
4	4E15.8	AA(1)	AA(2)		....				
		....							
5	4E15.8	BB(1)	BB(2)		....				
		....							
6	4E15.8	DD(1)	DD(2)		....				
		....							
7	4E15.8	EE(1)	EE(2)		....				
		....							
8a	7F10.5	U	V	XMIN	XMAX	YMIN	YMAX	DELU	
8b	7F10.5	U	V	XMIN	XMAX	YMIN	YMAX	DELU	

Note: Each of the data cards 8a, 8b, ... refers to one equipotential. There should be as many of these cards as the number (NUM) of equipotentials required.

```
      FORTRAN IV PROGRAM RAO2(INPUT,OUTPUT)
C   THIS IS THE CALLING PROGRAM FOR EQUIPOTENTIAL CALCULATION IN U-V AND
C   X-Y PLANES (CURVED BEAM).
      COMPLEX FF,GG,W,CLXPOT
      DIMENSION AA(32),BB(32),DD(32),EE(32),FF(32),GG(32),BCD(15)
      READ 5,(BCD(I),I=1,13)
    5 FORMAT(13A6)
      PRINT 10,(BCD(I),I=1,13)
   10 FORMAT(1H1,13A6)
      READ 15,M,NUM,NMAX
   15 FORMAT(3I10)
      M1=M+1
      READ 20, U1,V1,POT
   20 FORMAT(3F10.5)
      READ 25,(AA(I),I=1,M1)
      READ 25,(BB(I),I=1,M1)
      READ 25,(DD(I),I=1,M1)
      READ 25,(EE(I),I=1,M1)
   25 FORMAT(4E15.8)
      DO 30 I=1,M1
      FF(I)=CMPLX(DD(I),-EE(I))
   30 GG(I)=CMPLX(AA(I),BB(I))
      W=CMPLX(U1,V1)
      CLXPOT=(0.,0.)
      DO 35 I=1,M1
   35 CLXPOT=CLXPOT+FF(I)*W**I/FLOAT(I)
      CONST=POT-REAL(CLXPOT)
      DO 50 I=1,NUM
      READ 40,U,V,XMIN,XMAX,YMIN,YMAX,DELU
   40 FORMAT(7F10.5)
      PRINT 45,XMIN,XMAX,YMIN,YMAX,DELU
   45 FORMAT(11H-    XMIN =    F10.3,10H    XMAX =    F10.3,10H    YMIN =    F10.
     X3,10H    YMAX =    F10.3,10H    DELU =    F10.3)
   50 CALL EQPOT2(M,NMAX,FF,GG,CONST,U,V,XMIN,XMAX,YMIN,YMAX,DELU)
      STOP
      END
```

```
      SUBROUTINE EQPOT2(M,NMAX,FF,GG,CONST,U,V,XMIN,XMAX,YMIN,YMAX,DELU)
      COMPLEX FF,GG,W,Z,DW,CLXPOT
      DIMENSION FF(32),GG(32),SGN(2)
      M1=M+1
      DELUM=ABS(DELU)
      SU=DELU/DELUM
      PRINT 10
   10 FORMAT(11H-POTENTIAL 10X,1HU 13X,1HV 13X,5HDV/DU 11X,1HX 13X,1HY)
      II=1
      LL=1
   15 W=CMPLX(U,V)
      DW=(0.,0.)
      DO 20 J=2,M1
      L=J-1
   20 DW=DW+FF(J)*W**L
      DW=DW+FF(1)
      DV=REAL(DW)/AIMAG(DW)
      DVA=ABS(DV)
      SGN(II)=DV/DVA
      Z=(0.,0.)
      DO 25 J=2,M1
      L=J-1
   25 Z=Z+GG(J)*W**L
      Z=Z+GG(1)
      X=REAL(Z)
      Y=AIMAG(Z)
      IF(II-1)30,30,35
   30 II=2
      GO TO 40
   35 II=1
   40 CLXPOT=CMPLX(CONST,0.)
      DO 45 J=1,M1
   45 CLXPOT=CLXPOT+FF(J)*W**J/FLOAT(J)
      POT=REAL(CLXPOT)
      PRINT 50,POT,U,V,DV,X,Y
   50 FORMAT(6E15.5)
      IF (LL.LE.1) GO TO 55
      SM=SGN(1)*SGN(2)
      IF(SM.LT.0..AND.DVA.GT.1.)SU=-SU
      IF(SM.LT.0..AND.DVA.LE.1.)SV=-SV
      GO TO 60
   55 SV=SU*SGN(1)
   60 LL=LL+1
      IF (DVA.GT.1.) GO TO 65
      U=U+SU*DELUM
      V=V+SU*DELUM*DV
      GO TO 70
   65 V=V+SV*DELUM
      U=U+SV*DELUM/DV
```

```
70 IF(LL.GT.NMAX) GO TO 75
   IF(X.LT.XMIN.OR.X.GT.XMAX) GO TO 75
   IF(Y.LT.YMIN.OR.Y.GT.YMAX) GO TO 75
   GO TO 15
75 RETURN
   END
```

III. Solution of the Dirichlet problem for Laplace's Equation in two dimensions

In electrostatics one often has to determine the potentials and fields in a closed region when the potentials are specified on the boundary enclosing this region. The same problem also appears in the study of the motion of charged particles (for example in electron guns) when the potentials applied on the bounding electrodes are specified and one can assume that the charged particles themselves have a negligible effect on the electric fields. In electromagnetic problems one can sometimes assume that the electric field is curl-free. Here again one has to solve Laplace's equation. The program described here is taken mostly from a larger program written by Kirstein and Hornsby[1] for calculating the electron trajectories in electron guns. Large parts of the original program have been retained here with some changes and additions. The problem may be stated as follows:

Laplace's Equation:

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} = 0 \qquad \text{for rectangular geometry}$$

$$\frac{\partial^2 \varphi}{\partial r^2} + \frac{1}{r}\frac{\partial \varphi}{\partial r} + \frac{\partial^2 \varphi}{\partial z^2} = 0 \qquad \text{for problems with axial symmetry}$$

Boundary conditions:

φ specified on a closed boundary

Figure 3 represents the problem. For the solution of this problem, a net is first superimposed on the region of interest. The differential equation is reduced to a set of difference equations at the mesh points. Then these equations are solved by the method of over-relaxation. The

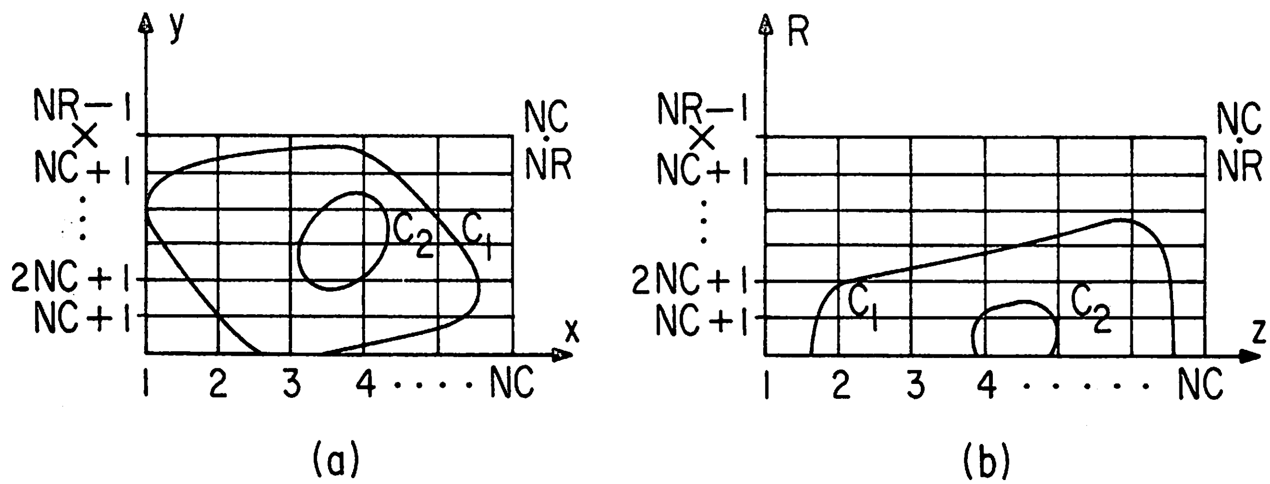

Fig. 3(a) Represents rectangular geometry. The numbering of the mesh points is shown, where the total number of mesh points for this problem is NC•NR. Fig. 3(b) shows a problem in cylindrical geometry, where the z-axis is the axis of symmetry. Typical boundaries along which potentials are specified are shown by C_1 and C_2.

program calculates the optimum relaxation factor by Carre's method[5] at various stages of the iteration process. After each iteration an estimate is made of the maximum error in the potential at the mesh points of the region and when this error is less than the prescribed tolerance set by the user, the program stops the iteration process and prints out the results. The following pages give a brief description of the program with explanation of the variables which appear in the input and information about the preparation of the control cards and data cards. The restrictions on the use of the program and the nature of the output are also described.

Program: Laplace's equation with potentials specified on a closed boundary.

Source Language: FORTRAN IV

Purpose: To calculate potentials and fields at internal points when the potentials are specified on a closed boundary. The program also calculates the potential and its derivatives at prescribed points within the region.

Comments: This is an overlay job which is similar, but not equivalent, to a chain job in the Fortran II system. Overlays are used because at the present time the CDC 6400 at Berkeley with a memory size of $140,000_8$ will not accommodate the entire program at once. The job consists of the main program which calls the overlays, and two primary overlays. At the present time it will not be executed unless the entire job is in binary form. The Fortran source decks for the main program and subroutines must be submitted for compilation with a request for binary decks. Once everything is in binary form, the job may be submitted for execution. A description of the deck setup follows the description of the output format.

Explanation of input variables:

DX = Mesh length in the X-direction

DY = Mesh length in the Y-direction

NSCC = Number of closed curves. The first closed must enclose all the other closed curves.

JOBNUM = Identification number for the particular computer run

KODBAS = Code number for problem; 200 for rectangular geometry and 250 for problems with axial symmetry.

N_i = Number of boundary points in the ith boundary curve

X_{ij}, Y_{ij}, φ_{ij} = The X-coordinate, Y-coordinate (or in axisymmetric case the Z and R coordinates respectively) and potential of the jth point of the ith closed curve. The points on each boundary must be written in clockwise order.

K = This is a code number which is left blank unless the particular boundary point is on the axis of an axi-symmetric problem (R_{ij} = 0) when it is equal to 251.

EF = Accuracy criterion for terminating the relaxation process. For 1% potential accuracy, EF = 1.0.

ANODPT = If the potentials of the boundary points have been normalized in the input and one wants the potentials and their derivatives at the specified points to be multiplied by some denormalizing factor, one should make ANODPT equal to this factor. Otherwise, it should be set to 1.0.

NN = Number of points at which the potentials and their derivatives are required. The potentials at the mesh points are always printed out whether one wants them or not.

X_i, Y_i = Coordinates of these points (i = 1, NN)

φ_i = Values of the equipotentials required

NV = Number of equipotentials required

Restrictions on the use of the program:

1) The closed boundary should be entirely in the first quadrant of the X-Y plane. Parts of the boundary may coincide with the axes.

2) The boundary points are arranged in clockwise order on each bounding curve. The first bounding curve should enclose all the others.

3) The total number of boundary points should be less than 498.

4) NC = number of columns in the net

$$= \text{Integral part of } \left[\frac{X_{max}}{\Delta x} + 0.0001\right] + 2$$

NR = number of rows in the net

$$= \text{Integral part of } \left[\frac{Y_{max}}{\Delta y} + 0.0001\right] + 2$$

$NC \leq 200$, $NR \leq 200$, $NR.NC \leq 3250$

5) Number of equipotentials $NV \leq 15$.

6) Number of points NN at which potentials and their derivatives are required should be less than 50.

7) Number of intersections of any mesh row or column with a boundary should be less than 20. The restriction can be usually circumvented by properly orienting the boundary.

DATA CARDS

Columns	1 15	16 30	31 45	46 50	51 55	56 60		
	DX(E15.6)	DY(E15.6)		NSCC	JOBNUM			
				KODBAS				
				N1				
	X_{11}(E15.6)	Y_{11}(E15.6)	φ_{11}(E15.6)	K				
	X_{12}(")	Y_{12}(")	φ_{12}(")					
		"	"	"				
	"	"	"	"				
	"	"	"	"				
				N2				
	X_{21}(E15.6)	Y_{21}(E15.6)	φ_{21}(E15.6)	K				
	"	"	"	"				
		EF (E15.6)						
	ANODPT(F10.5)							
	NN (I5)							
	X_1(F10.5)	Y_1(F10.5)						
	X_2(")	Y_2(")						
	"	"						
	"	"						
	Φ_1(E15.6)	Φ_2(E15.6)	Φ_3(E15.6)	NV				
	"	"	"					

Output format:

The first block of output is diagnostic printing about the relaxation procedure. It consists of one line of eight numbers for each iteration of the relaxation procedure. The most important of these are the first, second, fourth, sixth and last. The first is the total number of iterations carried out so far; the second is the over-relaxation factor used; the fourth is the mesh point number of the point with the largest absolute change in φ in the last iteration, and the sixth the potential there; the last number is the estimated maximum error in φ (as a fraction of φ). After thirteen iterations, a line of five quantities which are used in the calculation of the optimum relaxation factor for the next 12 iterations are printed out.

The next block of output has the denormalization factor (ANODPT) and the potentials and their derivatives at the prescribed points.

The next block contains the equipotentials. The X and Y coordinates of points on the equipotentials are printed below the value of the equipotential.

Finally the potential matrix is printed out. The numbering of the mesh points is indicated in Fig. 3.

Deck Setup for Laplace's Equation:

a. Job card (see CAL 6400 FORTRAN GUIDE. The field length must be set to the maximum 140,000 octal.)

b. INPUT.[1]

c. 7-8-9 card.[2]

---

[1] This card is punched beginning in column one.

[2] The numbers 7, 8, 9 are all punched in column one.

d. OVERLAY (XFILE, 0, 0)[3]

e. ⟨program MAIN1⟩.

f. OVERLAY (XFILE, 1, 0, C015572)[3]

g. ⟨Subroutines: PART1, MAIN2, MAKTAB, FNDPS, SORT, LOCAT, ININ, CEJE, JEJE, CHECK⟩.

h. OVERLAY (XFILE, 2, 0, C066106).

i. ⟨Subroutines: PART2, MAIN3, USER2, GETCO, DECC, ITRATE, MAIN6, MAIN4, INTPLT, MAIN5, EQUPOT, ZEROIN, TABLE⟩.

j. 7-8-9 card.[4]

k. ⟨Data⟩.

l. 6-7-8-9 card.

---

[3] This card is punched beginning in column seven.

[4] Each binary object deck includes as its last card a 7-8-9 card. This card is included in addition to the 7-8-9 card of the last binary deck in this block.

```
      PROGRAM MAIN1(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3=OUTPUT,TAPE7)
      COMMON DUMOVER(1)
      PRINT 1
    1 FORMAT( 40H1MAIN1 LOADED     NEXT STEP IS OVERLAY       )
      CALL OVERLAY(5HXFILE,1,0,0)
      CALL OVERLAY(5HXFILE,2,0,0)
      STOP
      END
```

```
      PROGRAM PART1
      COMMON DUMOVER(7034)
      CALL MAIN2
      RETURN
      END
```

```
      SUBROUTINE MAIN2
CMAIN2
C     000155I
C     MASTER PROGRAMME
      COMMON C5,C95,NT3,UT3,NT4,HT4,KT4,NT5,NE3,NE4,NE5,NC,NR,
     1KODBAS,DX,DY,JOBNUM,NSW1,NSW2,TIM
      DIMENSION NT3(780),UT3(780),NT4(780),HT4(780,4),KT4(780),
     1NT5(390,2),HL(4),UL(4),XDATA(500),YDATA(500),UDATA(500),
     2NCDATA(500),NBPR(200),NBPC(200),XB(200,20),YB(200,20),
     3UBPR(200,20),UBPC(200,20),KODBPR(200,20),KODBPC(200,20)
      DIMENSION KSCCP(4,4)
      EQUIVALENCE (UWANT,NUWANT)
      EQUIVALENCE(UBPR,KODBPR),(UBPC,KODBPC)
C     TAPES USED IN THIS PROGRAM
      NBII=1
C
      C5=0.0002
      C95=0.9998
      DO 1 I=1,200
      NBPR(I)=0
      NBPC(I)=0
      DO 1 J=1,20
      XB(I,J)=0
      YB(I,J)=0
      UBPR(I,J)=0
    1 UBPC(I,J)=0
      READ 100, DX,DY,NSCC,JOBNUM,KODBAS,NSCCP,NSW1,NSW2,TIM
      PRINT 101,JOBNUM
      IF(NSCCP)3,3,2
    2 READ 104,((KSCCP(I,J),J=1,4),I=1,NSCCP)
    3 KB=0
      DO 11 NCURVE=1,NSCC
      READ 102,NPTS
      KA=KB+1
      KB=KB+NPTS
      READ 107,(XDATA(I),YDATA(I),UDATA(I),NCDATA(I),I=KA,KB)
      DO 23 I=KA,KB
      NEARST=XDATA(I)/DX+0.5
      IF(ABS (XDATA(I)/DX-FLOAT (NEARST))-.00005)20,21,21
   20 XDATA(I)=XDATA(I)+.0001*DX
   21 NEARST=YDATA(I)/DY+0.5
      IF(ABS (YDATA(I)/DY-FLOAT (NEARST))-.00005)22,23,23
   22 YDATA(I)=YDATA(I)+.0001*DY
   23 CONTINUE
      XDATA(KB+1)=XDATA(KA)
      YDATA(KB+1)=YDATA(KA)
      UDATA(KB+1)=UDATA(KA)
      NCDATA(KB+1)=NCDATA(KA)
      DO 10 MP=KA,KB
      LP=MP+1
      CALL FNDBPS(YDATA(MP),YDATA(LP),XDATA(MP),XDATA(LP),UDATA(MP),
     1UDATA(LP),NCDATA(MP),NCDATA(LP),DY,NBPR,XB,UBPR,KODBPR)
      CALL FNDBPS(XDATA(MP),XDATA(LP),YDATA(MP),YDATA(LP),UDATA(MP),
     1UDATA(LP),NCDATA(MP),NCDATA(LP),DX,NBPC,YB,UBPC,KODBPC)
```

```
10 CONTINUE
11 CONTINUE
   NPTS=KB
   XMAX=XDATA(1)
   DO 6 J=2,NPTS
   IF(XDATA(J)-XMAX)6,6,5
 5 XMAX=XDATA(J)
 6 CONTINUE
   NC= INT (XMAX/DX)+2
   YMAX=YDATA(1)
   DO 8 J=2,NPTS
   IF(YDATA(J)-YMAX)8,8,7
 7 YMAX=YDATA(J)
 8 CONTINUE
   NR= INT (YMAX/DY)+2
   CALL SORT(NR,NBPR,XB,UBPR)
   CALL SORT(NC,NBPC,YB,UBPC)
   NE3=0
   NE4=0
   NE5=0
   DO 30 IR=1,NR
   YM=FLOAT (IR-1)*DY
   DO 30 IC=1,NC
   NOFPT=(IR-1)*NC+IC
   XM=FLOAT (IC-1)*DX
   CALL LOCAT(IR,XM,XB,NBPR,UBPR,HL(3),HL(1),UL(3),UL(1),LX,DX)
   CALL LOCAT(IC,YM,YB,NBPC,UBPC,HL(4),HL(2),UL(4),UL(2),LY,DY)
   MU=3*LX+LY+1
   GO TO (50,31,33,31,32,33,33,33,34),MU
50 IF(NSCCP)30,30,51
51 DO 52 I=1,NSCCP
   IF(KSCCP(I,1)-NOFPT)52,53,52
52 CONTINUE
   GO TO 30
53 DO 54 J=1,4
54 HL(J)=0.0
   J=KSCCP(I,2)
   HL(J)=1.0
   J=KSCCP(I,4)
   HL(J)=1.0
   MKR=3
   NUWANT=KSCCP(I,3)
   GO TO 35
31 CALL CEJE(HL,UL,MKR,UWANT)
   GO TO 35
32 CALL JEJE(HL,UL,MKR,UWANT)
   GO TO 35
33 PRINT 103,NOFPT
   STOP
34 CALL ININ(HL,UL,MKR,UWANT)
35 CALL MAKTAB(NOFPT,HL,UWANT,MKR,UWANT)
30 CONTINUE
   IF(KODBAS-200)41,40,40
40 REWIND NBI1
   WRITE (NBI1) NPTS
```

```
      WRITE (NBI1) (XDATA(I),YDATA(I),I=1,NPTS)
      REWIND NBI1
   41 IF(NE4)37,37,36
   36 CALL CHECK(XDATA,YDATA,UDATA,NPTS)
   37 RETURN
  101 FORMAT(49H1 SOLVE ELLIPTIC PARTIAL DIFF. EQUN....JOB NUMBER,I6///)
  102 FORMAT(45X,I5)
  107 FORMAT(3E15.7,I5)
  103 FORMAT(45H0 COMPUTER CANNOT DECIDE WHETHER POINT NUMBER,I5,33H IS
     1INSIDE OR OUTSIDE THE REGION.)
  100 FORMAT(2E15.7,15X,2I5/45X,2I5,2I1,F10.5)
  104 FORMAT(45X,2I5)
      END
```

```
      SUBROUTINE FNDBPS(Y1,Y2,X1,X2,U1,U2,NC1,NC2,DY,NBPR,XB,PHI,KODE)
CFNDBPS
C     0001552
C     TO SET UP TABLE OF MESH BOUNDARY POINTS
      DIMENSION NBPR(200),XB(200,20),PHI(200,20),KODE(200,20)
      IF(Y1-Y2)8,7,8
    7 RETURN
    8 YU=AMAX1(Y1,Y2)
      YL=AMIN1(Y1,Y2)
      MRNL= INT (YL/DY)+2
      MRNU= INT (YU/DY)+1
      IF(MRNU-MRNL)7,9,9
    9 SLOP1=(X2-X1)/(Y2-Y1)
      SLOP2=(U2-U1)/(Y2-Y1)
      DO 13 J=MRNL,MRNU
      L=NBPR(J)+1
      DIFF=FLOAT (J-1)*DY-Y1
      XB(J,L)=X1+DIFF*SLOP1
      IF(NC1+NC2)12,10,12
   10 IF(NC1)11,12,11
   11 KODE(J,L)=NC1
      GO TO 13
   12 PHI(J,L)=U1+DIFF*SLOP2
   13 NBPR(J)=L
      RETURN
      END
```

```
      SUBROUTINE SORT(NR,NBPR,XBPR,UBPR)
CSORT
C     00015S3
C     SORT AND REARRANGE BOUNDARY POINTS AND VALUE
      DIMENSION NBPR(200),XBPR(200,20),UBPR(200,20),F(20),G(20)
C
      DO 25 IR=1,NR
      J=NBPR(IR)
      IF(2*(J/2)-J)1,2,1
    1 PRINT 100,NR,J,(XBPR(IR,K),K=1,J)
      STOP
    2 K=0
    3 N=J
      IF(N-1)25,5,15
    5 K=K+1
      F(K)=XBPR(IR,1)
      G(K)=UBPR(IR,1)
      DO 10 L=1,K
      XBPR(IR,L)=F(L)
   10 UBPR(IR,L)=G(L)
      GO TO 25
   15 XMIN=XBPR(IR,1)
      PHI=UBPR(IR,1)
      DO 22 L=2,N
      J=L-1
      IF(XBPR(IR,L)-XMIN)21,20,20
   20 UBPR(IR,J)=UBPR(IR,L)
      XBPR(IR,J)=XBPR(IR,L)
      GO TO 22
   21 UBPR(IR,J)=PHI
      XBPR(IR,J)=XMIN
      PHI=UBPR(IR,L)
      XMIN=XBPR(IR,L)
   22 CONTINUE
      K=K+1
      F(K)=XMIN
      G(K)=PHI
      GO TO 3
   25 CONTINUE
      RETURN
  100 FORMAT(28H0  MESH ROW OR COLUMN NUMBER,I5,34H  APPEARS TO INTERSEC
     1T BOUNDARY AT,I5,20H  POINTS, THE NUMBER/83H0 OF POINTS BEING ODD
     2AND SO, IMPOSSIBLE. CO-ORDINATES OF ALLEGED INTERSECTIONS ARE//(F1
     35.6))
      END
```

```
      SUBROUTINE LOCAT(IR,XM,XBPR,NBPR,UBPR,HL,HR,UL,UR,LAMDA,DX)          0205
CLOCAT
C     00015S4                                                              0203
C     LOCATION OF MESH-POINT ROW-WISE OR COLUMN-WISE.                      0204
      DIMENSION XBPR(200,20),UBPR(200,20),NBPR(200)                        0206
      HL=1.0                                                               0207
      HR=1.0                                                               0208
      UL=0.0                                                               0209
      UR=0.0                                                               0210
      LAMDA=0                                                              0211
      N=NBPR(IR)                                                           0212
      IF(N)30,30,3                                                         0213
    3 J=1                                                                  0214
    4 IF(XM-XBPR(IR,J)) 10,10,5                                            0215
    5 IF(J-N)6,8,8                                                         0216
    6 J=J+1                                                                0217
      GO TO 4                                                              0218
    8 HR=1.0                                                               0219
      J=J+1                                                                0220
      GO TO 16                                                             0221
   10 HR=AMIN1((XBPR(IR,J)-XM)/DX,1.0)
      IF(HR-1.0)12,13,12                                                   0223
   12 UR=UBPR(IR,J)                                                        0224
   13 IF(J-1)16,15,16                                                      0225
   15 HL=1.0                                                               0226
      GO TO 20                                                             0227
   16 HL=AMIN1((XM-XBPR(IR,J-1))/DX,1.0)
      IF(HL-1.0)18,20,18                                                   0229
   18 UL=UBPR(IR,J-1)                                                      0230
   20 IF((J/2)*2-J)24,22,24                                                0231
   22 LAMDA=2                                                              0232
      RETURN                                                               0233
   24 IF(HL-1.0)28,26,28                                                   0234
   26 IF(HR-1.0)28,30,28                                                   0235
   28 LAMDA=1                                                              0236
   30 RETURN                                                               0237
      END                                                                  0238
```

```
      SUBROUTINE ININ(HL,UL,MKR,UWANT)                                  0241
CININ
C     0001555                                                           0239
C     TREATMENT OF INTERNAL POINTS                                      0240
      DIMENSION HL(4),UL(4)                                             0242
      COMMON C5,C95                                                     0243
      NHNOT1=0                                                          0244
      NFIXU=0                                                           0245
      DO 10 J=1,4                                                       0246
      IF(HL(J)-1.0)5,10,5                                               0247
    5 NHNOT1=NHNOT1+1                                                   0248
      IF(UL(J)) 10,10,6
    6 IF(UL(J)- 1.0E-128) 7,7,10
    7 L=J                                                               0253
      NFIXU=NFIXU+1                                                     0254
   10 CONTINUE                                                          0255
      IF(NHNOT1)14,12,14                                                0256
   12 MKR=1                                                             0257
      RETURN                                                            0258
   14 IF(NFIXU-1)16,25,12                                               0259
   16 I=1                                                               0260
      HMIN=HL(1)                                                        0261
      DO 20 J=2,4                                                       0262
      IF(HL(J)-HMIN)18,20,20                                            0263
   18 HMIN=HL(J)                                                        0264
      I=J                                                               0265
   20 CONTINUE                                                          0266
      IF(HMIN-C5)22,12,12                                               0267
   22 MKR=2                                                             0268
      UWANT=UL(I)                                                       0269
      RETURN                                                            0270
   25 IF(HL(L)-C95)26,26,16                                             0271
   26 IF(HL(L)-C5)27,27,35                                              0272
   27 HMIN=1.0                                                          0273
      DO 30 J=1,4                                                       0274
      IF(J-L)28,30,28                                                   0275
   28 IF(HL(J)-HMIN)29,29,30                                            0276
   29 HMIN=HL(J)                                                        0277
      I=J                                                               0278
   30 CONTINUE                                                          0279
      IF(HMIN-C5)22,31,31                                               0280
   31 MKR=3                                                             0281
      UWANT=UL(L)                                                       0282
      RETURN                                                            0283
   35 MKR=5                                                             0284
      RETURN                                                            0285
      END                                                               0286
```

```
      SUBROUTINE JEJF(HL,UL,MKR,UWANT)                                    0289
CJEJF
C     00015S6                                                             0287
C     TREATMENT OF POINT JUST EXTERIOR IN BOTH DIRECTIONS                 0288
      COMMON C5,C95                                                       0290
      DIMENSION HL(4),UL(4)                                               0291
      NFIXU=0                                                             0292
      DO 10 J=1,4                                                         0293
      IF(HL(J)-1.0)5,10,5                                                 0294
    5 IF(UL(J)) 10,10,6
    6 IF(UL(J)- 1.0E-128) 7,7,10
    7 L=J                                                                 0299
      NFIXU=NFIXU+1                                                       0300
   10 CONTINUE                                                            0301
      IF(NFIXU-1)12,13,11                                                 0302
   11 MKR=1                                                               0303
      DO 31 I=1,4                                                         0304
   31 HL(I)=1.0                                                           0305
      RETURN                                                              0306
   12 L=5                                                                 0307
      GO TO 14                                                            0308
   13 IF(HL(L)-C95)20,20,14                                               0309
   14 HMIN=1.0                                                            0310
      DO 17 J=1,4                                                         0311
      IF(J-L)15,17,15                                                     0312
   15 IF(HL(J)-HMIN)16,17,17                                              0313
   16 HMIN=HL(J)                                                          0314
      I=J                                                                 0315
   17 CONTINUE                                                            0316
   18 UWANT=UL(I)                                                         0317
      MKR=2                                                               0318
      RETURN                                                              0319
   20 IF(HL(L)-C5)25,21,21                                                0320
   21 MKR=5                                                               0321
      RETURN                                                              0322
   25 HMIN=1.0                                                            0323
      DO 28 J=1,4                                                         0324
      IF(J-L)26,28,26                                                     0325
   26 IF(HL(J)-HMIN)27,28,28                                              0326
   27 HMIN=HL(J)                                                          0327
      I=J                                                                 0328
   28 CONTINUE                                                            0329
      IF(HMIN-C95)18,30,30                                                0330
   30 UWANT=UL(L)                                                         0331
      J=MOD(L+1,4)+1
      HL(L)=HL(J)                                                         0333
      HL(J)=0.0                                                           0334
      MKR=3                                                               0335
      RETURN                                                              0336
      END                                                                 0337
```

```
      SUBROUTINE CFJE(HL,UL,MKR,UWANT)
CCFJE
C     0001557
C     TREATMENT OF POINT JUST EXTERIOR/COMPLETELY EXTERIOR
      DIMENSION HL(4),UL(4)
      COMMON C5,C95
      MARK=0
      NHNOT1=0
      NFIXU=0
      DO 10 J=1,4
      IF(HL(J)-1.0)5,10,5
    5 NHNOT1=NHNOT1+1
      IF(UL(J)) 10,10,6
    6 IF(UL(J)- 1.0E-128) 7,7,10
    7 L=J
      NFIXU=NFIXU+1
   10 CONTINUE
      IF(NFIXU-1)15,30,11
   11 MKR=6
      RETURN
   15 L=5
   14 HMIN=1.0
      DO 18 J=1,4
      IF(J-L)16,18,16
   16 IF(HL(J)-HMIN)17,18,18
   17 HMIN=HL(J)
      I=J
   18 CONTINUE
      IF(MARK)22,20,22
   20 UWANT=UL(I)
      MKR=2
      RETURN
   22 IF(HMIN-C95)25,25,24
   24 UWANT=UL(L)
      J=MOD(L+1,4)+1
      HL(L)=HL(J)
      HL(J)=0.0
      MKR=3
      RETURN
   25 MKR=7
      RETURN
   30 IF(NHNOT1-1)35,31,35
   31 IF(HL(L)-C95)33,33,32
   32 MKR=4
      RETURN
   33 IF(HL(L)-C5)24,34,34
   34 MKR=5
      RETURN
   35 IF(HL(L)-C95)36,36,14
   36 IF(HL(L)-C5)37,34,34
   37 MARK=1
      GO TO 14
      END
```

```
      SUBROUTINE CHECK(XDATA,YDATA,UDATA,NPTS)
CCHECK
C     0001558
C     CHECK THAT NEIGHBOURS ARE DEFINED
      COMMON C5,C95,NT3,UT3,NT4,HT4,KT4,NT5,NE3,NE4,NE5,NC,NR
      COMMON KODBAS,DX,DY
      DIMENSION XDATA(500),YDATA(500),UDATA(500),NT3(780),UT3(780),
     1NT4(780),HT4(780,4),KT4(780),NT5(390,2)
C
      DO 30 IN=1,NE4
      DO 20 INN=1,4
      IF(HT4(IN,INN))5,20,5
    5 GO TO (6,7,8,9),INN
    6 N=NT4(IN)+1
      GO TO 10
    7 N=NT4(IN)+NC
      GO TO 10
    8 N=NT4(IN)-1
      GO TO 10
    9 N=NT4(IN)-NC
   10 DO 11 J1=1,NE3
      IF(NT3(J1)-N)11,20,11
   11 CONTINUE
      DO 12 J2=1,NE4
      IF(NT4(J2)-N)12,20,12
   12 CONTINUE
      DO 14 J3=1,NE5
      IF(N-NT5(J3,1))14,20,13
   13 IF(NT5(J3,2)-N)14,20,20
   14 CONTINUE
      DO 15 K=1,4
      IF(HT4(IN,K))15,16,15
   15 CONTINUE
   17 PRINT 100,NT4(IN),N
      STOP
   16 M=NT4(IN)
      YM=FLOAT ((M-1)/NC)*DY
      Q= MOD (M-1,NC)
      XM=Q*DX
      GO TO (21,40,21,40),INN
   21 DO 26 IP=1,NPTS
      IF(ABS (YDATA(IP)-YM)-.003*DY)22,26,26
   22 RHO=(XDATA(IP)-XM)/DX
      IF(INN-1)23,24,23
   23 RHO=-RHO
   24 IF(RHO)26,25,25
   25 IF(1.0-RHO)26,27,27
   26 CONTINUE
      GO TO 17
   40 DO 46 IP=1,NPTS
      IF(ABS (XDATA(IP)-XM)-.003*DX)42,46,46
   42 RHO=(YDATA(IP)-YM)/DY
      IF(INN-2)43,44,43
   43 RHO=-RHO
```

```
 44 IF(RHO)46,45,45
 45 IF(1.0-RHO)46,27,27
 46 CONTINUE
    GO TO 17
 27 IF(C5-RHO)52,52,50
 50 NE3=NE3+1
    NT3(NE3)=N
    UT3(NE3)=UDATA(IP)
    NT4(IN)=0
    GO TO 30
 52 NE3=NE3+1
    NT3(NE3)=N
    UT3(NE3)=UDATA(IP)
    IF(RHO-C95)54,54,53
 53 RHO=1.0
 54 HT4(IN,INN)=RHO
 20 CONTINUE
 30 CONTINUE
    RETURN
100 FORMAT(14H0 POINT NUMBER,I5,27H HAS NEIGHBOUR-POINT NUMBER,I5,10H
   1FOR WHICH/62H NO BOUNDARY VALUE OR FINITE DIFFERENCE EQUATION IS A
   2VAILABLE.)
    END
```

```
      SUBROUTINE MAKTAB(M,HL,UWANT,MKR,KODE)
CMAKTAB
C     0001559
C     WRITE DETAILS OF POINT INTO APPROPRIATE TABLE OR SIGNAL ERROR.
      COMMON C5,C95,NT3,UT3,NT4,HT4,KT4,NT5,NE3,NE4,NE5,NC,NR,KODBAS
      DIMENSION NT3(780),UT3(780),NT4(780),HT4(780,4),KT4(780),
     1NT5(390,2),HL(4)
C
      DO 11 J=1,4
      IF(C5-HL(J))7,7,5
    5 HL(J)=0.0
       GO TO 11
    7 IF(HL(J)-C95)11,11,10
   10 HL(J)=1.0
   11 CONTINUE
       GO TO (15,37,31,26,40,42,44),MKR
   15 DO 17 J=1,4
      IF(HL(J)-1.0)30,17,30
   17 CONTINUE
       IF(NE5)20,21,20
   20 IF(M-NT5(NE5,2)-1)21,25,21
   21 NE5=NE5+1
      IF(NE5-390)23,23,22
   22 PRINT 103
      STOP
   23 NT5(NE5,1)=M
   25 NT5(NE5,2)=M
   26 RETURN
   30 KODE=KODBAS
   31 NE4=NE4+1
      IF(NE4-780)32,32,22
   32 NT4(NE4)=M
      KT4(NE4)=KODE
      DO 35 J=1,4
   35 HT4(NE4,J)=HL(J)
      RETURN
   37 NE3=NE3+1
      IF(NE3-780)38,38,22
   38 NT3(NE3)=M
      UT3(NE3)=UWANT
      RETURN
   40 PRINT 100,M
      STOP
   42 PRINT 101,M
      STOP
   44 PRINT 102,M
      STOP
  100 FORMAT(7H0 POINT,I5,67H NEAR BOUNDARY WITH DERIVATIVE-TYPE BOUNDAR
     1Y CONDITIONS IS SUSPECT./46H BOUNDARY SHOULD BE COINCIDENT WITH ME
     2SH LINE.)
  101 FORMAT(46H0 THERE APPEAR TO BE TWO BOUNDARIES NEAR POINT,I5,42H WI
     1TH DERIVATIVE-TYPE BOUNDARY CONDITIONS./22H THIS IS INADMISSIBLE.)
  102 FORMAT(7H0 POINT,I5,55H NEAR BOUNDARY WITH DERIVATIVE-TYPE BOUNDAR
     1Y CONDITIONS/34H IS TOO CLOSE TO ANOTHER BOUNDARY.)
```

```
103 FORMAT(86H0 THE TABLE OF BOUNDARY, IRREGULAR OR REGULAR POINTS IS
   1FULL. RERUN WITH COARSER MESH.)
    END
```

```
PROGRAM PART2
COMMON DUMOVER(27718)
CALL MAIN3
CALL MAIN6
CALL MAIN3
CALL MAIN4
CALL MAIN5
RETURN
END
```

```
      SUBROUTINE DECC
CDECC
C     00015S11
C     CALCULATE COEFFICIENTS OF DIFFERENCE EQUATIONS
      COMMON C5,C95,NT3,UT3,NT4,HT4,KT4,NT5,NE3,NE4,NE5,NC,NR,KODBAS,
     1DX,DY,JOBNUM,UMAT,NEQU,DUM,NUMPT,DIFCO,X,Y,HL,M,COFFT
      DIMENSION NT3(780),UT3(780),NT4(780),HT4(780,4),KT4(780),
     1NT5(390,2),UMAT(3250),NUMPT(2900),DIFCO(2900,5),
     2HL(4),COFFT(5),DUM(20),JOBNUM(4)
C
      NEQU=0
      J4=1
      K5=1
    4 IF(K5-NE5)6,6,5
    5 IF(J4-NE4)8,8,35
    6 IF(J4-NE4)7,7,20
    7 IF(NT5(K5,1)-NT4(J4))20,20,8
    8 IF(NT4(J4))16,16,9
    9 M=NT4(J4)
      Y=FLOAT ((M-1)/NC)*DY
      Q= MOD (M-1,NC)
      X=Q*DX
      DO 12 I=1,4
   12 HL(I)=HT4(J4,I)
      NEQU=NEQU+1
      CALL GETCO(KT4(J4))
      NUMPT(NEQU)=M
      DO 15 I=1,5
   15 DIFCO(NEQU,I)=COFFT(I)
   16 J4=J4+1
      GO TO 4
   20 NPL=NT5(K5,1)
      NPU=NT5(K5,2)
      DO 23 I=1,4
   23 HL(I)=1.0
      DO 31 M=NPL,NPU
      Y=FLOAT ((M-1)/NC)*DY
      Q= MOD (M-1,NC)
      X=Q*DX
      NEQU=NEQU+1
      CALL GETCO(KODBAS)
      NUMPT(NEQU)=M
      DO 30 I=1,5
   30 DIFCO(NEQU,I)=COFFT(I)
   31 CONTINUE
      K5=K5+1
      GO TO 4
   35 RETURN
      END
```

```
      SUBROUTINE ITRATE(BETA,DMAX,UCORR,ICORR,GNORM)                    0578
CITRATE
C     00015S12                                                          0576
C     THE EXTRAPOLATED-LIEBMANN ITERATION                               0577
      COMMON DUM1,NC,NR,KODBAS,DX,DY,JOBNUM,UMAT,NEQU,DUM2,NUMPT,DIFCO  0579
      DIMENSION DUM1(7025),UMAT(3250),DUM2(20),NUMPT(2900),DIFCO(2900,5) 0580
     1,JOBNUM(4)
      DMAX=0                                                            0581
      GNORM=0                                                           0582
      DO 2 I=1,NEQU                                                     0583
      J=NUMPT(I)                                                        0584
      K=J+NC                                                            0585
      L=J-NC                                                            0586
      RES=DIFCO(I,1)*UMAT(J+1)+DIFCO(I,2)*UMAT(K)                       0587
     1+DIFCO(I,3)*UMAT(J-1)+DIFCO(I,4)*UMAT(L)                          0588
     2+DIFCO(I,5)-UMAT(J)                                               0589
      DELTA=BETA*RES                                                    0590
      UMAT(J)=UMAT(J)+DELTA                                             0591
      GNORM=GNORM+ABSF(DELTA)                                           0592
      IF(ABS (DELTA)-DMAX)2,2,1
    1 DMAX=ABS (DELTA)
      UCORR=UMAT(J)                                                     0595
      ICORR=I                                                           0596
    2 CONTINUE                                                          0597
      RETURN                                                            0598
      END                                                               0599
```

```
      SUBROUTINE USER2                                                  0601
CUSER2
C     00015S18                                                          0600
      RETURN
      END                                                               0604
```

```
      SUBROUTINE GETCO(KODE)
CGETCO
C       00015527                                                       0605
C                                                                      0607
C     CALCULATION OF DIFFERENCE EQUATION COEFFICIENTS FOR POISSONS     0608
C     EQUATION IN CYLINDRICAL COORDINATES WITH AXIAL SYMMETRY (X=Z,Y=R) 0609
C     BASIC CODE IS 250, CODE FOR AXIS IS 251.                         0610
C     FOR POISSONS EQUATION IN RECTANGULAR CO-ORDINATES,               0611
C     USE BASIC CODE 200.                                              0612
C                                                                      0613
      COMMON DUM1,DX,DY,JOBNUM,UMAT,JEQU,DUM2,NLITS,NBFREF,DUM3,NUMPT,
     1DIFCO,X,Y,H,M,C
      DIMENSION H(4),C(5),DUM1(7028),UMAT(3250),DUM2(3),BU(127),JOBNUM(4   0615
     1),DUM3(15),NUMPT(2900),DIFCO(2900,5)
C     TAPES USED IN THIS PROGRAM
      NBI2=2
C                                                                      0616
      R=DX/DY                                                          0617
      IF(KODE-250)6,5,6                                                0618
    6 IF(KODE-251)7,10,7                                               0619
    7 T1=(H(1)*H(3)*R**2)/(H(2)*H(4))                                  0620
      T2=(H(1)+H(3))*(1.+T1)                                           0621
      T3=(H(2)+H(4))*(1.+1./T1)                                        0622
      C(1)=H(3)/T2                                                     0623
      C(2)=H(4)/T3                                                     0624
      C(3)=H(1)/T2                                                     0625
      C(4)=H(2)/T3                                                     0626
      EL=R/(H(2)*H(4))+1./(R*H(1)*H(3))                                0627
      GO TO 15                                                         0628
    5 T1=0.5*DY/Y                                                      0629
      T2=1.0+H(4)*T1                                                   0630
      T1=H(2)*T1                                                       0631
      EL=(T2-T1)*R/(H(2)*H(4))+1.0/(R*H(1)*H(3))                       0632
      C(1)=1.0/(EL*R*(H(1)+H(3)))                                      0633
      C(3)=C(1)/H(3)                                                   0634
      C(1)=C(1)/H(1)                                                   0635
      C(2)=R/(EL*(H(2)+H(4)))                                          0636
      C(4)=(1.0-T1)*C(2)/H(4)                                          0637
      C(2)=T2*C(2)/H(2)                                                0638
      GO TO 15                                                         0639
   10 T1=2.0*R/H(2)**2                                                 0640
      EL=T1+1.0/(R*H(1)*H(3))                                          0641
      T2=1.0/(EL*R*(H(1)+H(3)))                                        0642
      C(1)=T2/H(1)                                                     0643
      C(2)=T1/EL                                                       0644
      C(3)=T2/H(3)                                                     0645
      C(4)=0.0                                                         0646
   15 C(5)=0.0                                                         0647
      IF(NLITS)20,20,21                                                0648
   20 RETURN                                                           0649
   21 IF(M-NBFREF)23,26,26                                             0650
   23 LAM=(NBFREF-M-1)/127+1                                           0651
      LAM1=LAM+1                                                       0652
      DO 24 I=1,LAM1                                                   0653
```

```
   24 BACKSPACE NBI2
      NBFREF=NBFREF-127*LAM
   25 READ (NBI2)     (BU(I),I=1,127)
      GO TO 21
   26 IF(M-NBFREF-126)28,28,27
   27 NBFREF=NBFREF+127
      GO TO 25
   28 LAM=M-NBFREF+1
      C(5)=(-BU(LAM)*DX*DY)/(2.*FL)
      RETURN
      END
```

```
      SUBROUTINE MAIN3
CMAIN3
C     02007 S 11
      COMMON C5,C95,NT3,UT3,NT4,HT4,KT4,NT5,NE3,NE4,NE5,NC,NR,KODBAS,
     1DX,DY,JOBNUM,NSW1,NSW2,TIM,
     2             UMAT,NEQU,NPIA,EWANT,BFINAL,NLITS,NBFREF,DUM,
     3OPM,BETA,BEEPR,EIGEN,NIT,NITP,NUMPT,DIFCO,DUM2,NITSUB
      DIMENSION DUM(9),DUM2(12)
      DIMENSION NT3(780),UT3(780),NT4(780),HT4(780,4),KT4(780),
     1NT5(390,2),UMAT(3250),NUMPT(2900),DIFCO(2900,5),P(12)
C     TAPES USED IN THIS PROGRAM
      NOUT2=7
      GNORML=1.0
      MTAPE=NOUT2
      IF(C5-0.0002)21,72,21
   72 RETURN
   21 CALL DECC
   22 NC12=0
   23 CALL ITRATE(BETA,DMAX,UCORR,ICORR,GNORM)
      NIT=NIT+1
      EMAX=EIGEN*DMAX/(ABS (UCORR)*(1.0-EIGEN))
      NC12=NC12+1
      NITA=NIT-NITSUB
      PRINT 103,NITA,BETA,GNORM,NUMPT(ICORR),DMAX,UCORR,EIGEN,EMAX
      IF(EMAX-EWANT)75,75,35
   75 IF(NIT-NITP-1)776,776,775
  775 IF(OPM)30,76,30
  776 OPM=1.0
   30 WRITE (MTAPE,101)              JOBNUM,DX,DY,NR,NC
      WRITE (MTAPE,102)              BFINAL,EIGEN,BETA,BEEPR,NITA,(UMAT(I),
     1I=1,NPIA)
      WRITE (MTAPE,105)              NE3,NE4,NE5,(NT3(I),UT3(I),I=1,NE3)
      WRITE (MTAPE,106)              (NT4(I),(HT4(I,J),J=1,4),KT4(I),I=1,NE
     14)
      WRITE (MTAPE,107)              (NT5(I,1),NT5(I,2),I=1,NE5)
      WRITE (MTAPE,108)
  108 FORMAT(3HEOF)
      REWIND MTAPE
   76 CONTINUE
   31 CALL USER2
      RETURN
      GO TO 21
   35 IF(NIT-1)36,36,40
   36 BETA=1.375
      IF(BFINAL)22,22,38
   38 BETA=BFINAL
      GO TO 22
   40 IF(BFINAL)45,45,22
   45 P(NC12)=GNORM/GNORML
      IF(NC12-12)47,49,49
   47 GNORML=GNORM
      GO TO 23
   49 IF(NIT-13)50,51,50
   50 EIGEN=P(12)
```

```
      GO TO 58
   51 D10=P(10)-P(11)
      D11=P(11)-P(12)
      IF(D11)56,56,55
   55 IF(D10-D11)50,50,57
   56 IF(D11-D10)50,50,57
   57 EIGEN=P(10)-D10**2/(D10-D11)
      IF(EIGEN-1.0)58,50,50
   58 IF(EIGEN-1.0)59,60,60
   59 BEENW=2.0/(1.0+SQRTF(1.0-(EIGEN+BETA-1.0)**2/(EIGEN*BETA**2)))
      IF(ABS (BEENW-BEEPR)/(2.0-BEENW)-.05)60,65,65
   60 BFINAL=BEENW
      BETA=BFINAL
      EIGEN=BETA-1.0
      GO TO 70
   65 BEEPR=BEENW
      BETA=BEENW-(2.0-BEENW)/4.0
   70 PRINT 104,P(10),P(11),P(12),EIGEN,BEENW
      GO TO 22
  101 FORMAT(I15,2E15.7,2I15)
  102 FORMAT(4E15.7,I15//(1P7E15.7))
  104 FORMAT(//5F15.7//)
  103 FORMAT(I5,1P2E15.7,I5,4F15.7)
  105 FORMAT(///3I15///(I6,1PE15.7))
  106 FORMAT(///(I6,4F10.6,I6))
  107 FORMAT(///(2I6))
      END
```

```
      SUBROUTINE MAIN6
CMAIN6
      COMMON C5,C95,NT3,UT3,NT4,HT4,KT4,NT5,NE3,NE4,NE5,NC,NR,KODBAS,
     1DX,DY,JOBNUM,NSW1,NSW2,TIM,
     2                   UMAT,NEQU,NPIA,EWANT,BFINAL,NLITS,NBFREF,DUM,
     3OPM,BETA,BEEPR,EIGEN,NIT,NITP,NUMPT,DIFCO,DUM2,NITSUB
      DIMENSION DUM(9),DUM2(12)
      DIMENSION NT3(780),UT3(780),NT4(780),HT4(780,4),KT4(780),
     1NT5(390,2),UMAT(3250),NUMPT(2900),DIFCO(2900,5),P(12)
C     TAPES USED IN THIS PROGRAM
      NTAPE=5
C
      NLITS=0
      OPM=1.0
      C5=0.0002001
   71 NPIA=NC*NR
      READ 100,BFINAL,EWANT,INPTMK
      EWANT=EWANT/100.0
      MKCON=0
      IF(INPTMK-1)1,6,6
    1 MKCON=0
      EIGEN=0.95
      NIT=0
      BETA=1.0
      IF(BFINAL-1.0)3,3,2
    2 EIGEN=BFINAL-1.0
    3 DO 4 I=1,NPIA
    4 UMAT(I)=0.0
      GO TO 9
    6 READ (NTAPE,101)         JUNK,WOT,WOT,NRO,NCO
      IF(NR+NC-NRO-NCO)7,8,7
    7 REWIND NTAPE
      GO TO 1
    8 READ (NTAPE,108)         BFINAL,EIGEN,BETA,BEEPR,NIT,JUNK,
     1(UMAT(I),I=1,NPIA)
      REWIND NTAPE
    9 NITSUB=0
      IF(MKCON)16,11,16
   11 NITSUB=NIT
   16 DO 18 I3=1,NE3
      JK=NT3(I3)
   18 UMAT(JK)=UT3(I3)
      RETURN
  100 FORMAT(2E15.7,15X,I5)
  101 FORMAT(I15,2E15.7,2I15)
  108 FORMAT(4E15.7,I15/I10/(7E15.7))
      END
```

```
      SUBROUTINE MAIN4
CMAIN4
C     THIS CHAIN USES INTPLT TO CALCULATE POTENTIAL AND FIELDS ON A
C     PRESCRIBED CURVE AND THEN CALLS CHAIN(4,A4) FOR EQUIPOTENTIAL PLOT
      DIMENSION NT3(780),UT3(780),NT4(780),HT4(780,4),KT4(780),NT5(390,2
     1),UMAT(3250),X(50),Y(50)
      COMMON C5,C95,NT3,UT3,NT4,HT4,KT4,NT5,NE3,NE4,NE5,NC,NR,KODBAS,
     1DZ,DR,JOBPOT,NSW1,NSW2, TIM,UMAT
      READ 5, ANODPT
    5 FORMAT(F10.5)
      PRINT 6, ANODPT
    6 FORMAT(19H-ANODE POTENTIAL = F10.5)
      PRINT 7
    7 FORMAT(42H-POTENTIALS AND FIELDS ON PRESCRIBED CURVE)
      READ 10,NN,(X(I),Y(I),I=1,NN)
   10 FORMAT(I5/(2F10.5))
      PRINT 15
   15 FORMAT(8H-      X11X,1HY9X,3HPHI9X,2HPY9X,2HPX)
      DO 40 J=1,NN
      CALL INTPLT(X(J),Y(J),PHI,PY,PX,OUT)
      IF(OUT)30,30,20
   20 PRINT 25
   25 FORMAT(35H-THIS POINT IS OUTSIDE THE BOUNDARY)
      GO TO 40
   30 PHI=PHI*ANODPT
      PY=PY*ANODPT
      PX=PX*ANODPT
      PRINT 35,X(J),Y(J),PHI,PY,PX
   35 FORMAT(5F12.5)
   40 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INTPLT(ZGIVEN,RGIVEN,VPOT,DVDR,DVDZ,OUT)
CINTPLT
C        11001S14
      DIMENSION NT3(780),UT3(780),NT4(780),HT4(780,4),KT4(780),NT5(390,2
     1),UMAT(3250),M(4),H(4,4),VR(4),VZ(4),V(4)
      COMMON C5,C95,NT3,UT3,NT4,HT4,KT4,NT5,NE3,NE4,NE5,NC,NR,KODBAS,
     1DZ,DR,JOBPOT,NSW1,NSW2, TIM,UMAT
C
C     CALCULATES V,DV/DR,DV/DZ,BYLINEAR INTERPOLATION AT ANY POINT(Z,R).
C     IF POINT IS OUTSIDE THE REGION,SETS(OUT)=1.0.SHOULD WORK FOR ANY
C     REGION,INCLUDING THOSE WITH HOLES,PROVIDED EITHER(A)V IS SPECIFIED
C     EVERY WHERE ON THE BOUNDARY OR(B)DITTO EXCEPT THAT ON THE X-AXIS
C     (=Z-AXIS)DV/DR=0.ACCURACY OF DERIVATES IS LESS NEAR BOUNDARY.MAY
C     WORK FOR POINTS ON BOUNDARY.
C
      IF(ZGIVEN)1,1,7
    1 OUT=1.0
      RETURN
    7 IF(RGIVEN)8,8,4
    8 IF(KODBAS-200)1,1,4
    4 OUT=0.
      Z=ABS (ZGIVEN)
      R=ABS (RGIVEN)
      IL= INT (R/DR)+1
      JL= INT (Z/DZ)+1
      IF(JL-NC)2,50,50
    2 IF(IL-NR)3,50,50
    3 IF(JL-JLP)6,5,6
    5 IF(IL-ILP)6,40,6
    6 M(1)=(IL-1)*NC+JL
      M(2)=M(1)+1
      M(4)=M(1)+NC
      M(3)=M(4)+1
      IND=1
      NTOT=1
      DO 31 K=1,4
      MM=M(K)
      DO 10 JJ=1,NE5
      IF(MM-NT5(JJ,1))10,16,9
    9 IF(NT5(JJ,2)-MM)10,16,16
   10 CONTINUE
      IF(NE4)30,30,11
   11 DO 15 JJ=1,NE4
      IF(NT4(JJ)-MM)15,18,15
   15 CONTINUE
      GO TO 30
   16 DO 17 L=1,4
   17 H(K,L)=1.0
      GO TO 20
   18 DO 19 L=1,4
   19 H(K,L)=HT4(JJ,L)
   20 ML=MM-NC
      MU=MM+NC
      VR(K)=0.0
```

```
      IF(ML)23,23,22
   22 VR(K)=(H(K,4)*(UMAT(MU)-UMAT(MM))/H(K,2)-H(K,2)*(UMAT(ML)-UMAT(MM)
     1)/H(K,4))/((H(K,2)+H(K,4))*DR)
   23 VZ(K)=(H(K,3)*(UMAT(MM+1)-UMAT(MM))/H(K,1)-H(K,1)*(UMAT(MM-1)-UMAT
     1(MM))/H(K,3))/((H(K,1)+H(K,3))*DZ)
      NTOT=NTOT+IND
   30 IND=2*IND
   31 V(K)=UMAT(MM)
   40 X1=R/DR-FLOAT (IL-1)
      X2=Z/DZ-FLOAT (JL-1)
      GO TO(50,51,52,52,53,50,53,54,54,51,50,53,54,52,51,51),NTOT
   50 OUT=1.0
      GO TO 99
   51 XI=X1
      ETA=X2
      K=1
      GO TO 55
   52 XI=1.0-X2
      ETA=X1
      K=2
      GO TO 55
   53 XI=1.0-X1
      ETA=1.0-X2
      K=3
      GO TO 55
   54 XI=X2
      ETA=1.0-X1
      K=4
   55 I1= MOD (K,4)+1
      I2= MOD (K+1,4)+1
      I3= MOD (K+2,4)+1
      GO TO(50,61,61,62,61,50,62,63,61,62,50,63,62,63,63,64),NTOT
   61 IF(XI/H(K,I1)+ETA/H(K,K)-1.0)70,70,50
   70 DVDR=VR(K)
      DVDZ=VZ(K)
      VPOT=(V(I3)-V(K))*XI/H(K,I1)+(V(I1)-V(K))*ETA/H(K,K)+V(K)
      GO TO 99
   62 IF((H(I3,K)-H(K,K))*(XI-1.0)-(ETA-H(I3,K)))50,71,71
   71 DVDR=(1.0-XI)*VR(K)+XI*VR(I3)
      DVDZ=(1.0-XI)*VZ(K)+XI*VZ(I3)
      VPOT=(V(I3)-V(K))*XI+(V(I1)-V(K))*ETA/H(K,K)+V(K)
      GO TO 99
   63 C1=1.0-H(I1,I2)
      C2=1.0-H(I3,I3)
      IF(C1*XI+C2*ETA-C1*C2)50,72,72
   72 DVDR=(VR(I2)-VR(I1))*XI+(VR(I2)-VR(I3))*ETA+VR(I3)+VR(I1)-VR(I2)
      DVDZ=(VZ(I2)-VZ(I1))*XI+(VZ(I2)-VZ(I3))*ETA+VZ(I3)+VZ(I1)-VZ(I2)
      VPOT=(V(I2)-V(I1))*XI+(V(I2)-V(I3))*ETA+V(I3)+V(I1)-V(I2)
      GO TO 99
   64 C3=1.0-XI
      C4=1.0-ETA
      C1=C3*C4
      C2=ETA*C3
      C3=XI*ETA
      C4=XI*C4
```

```
      DVDR=VR(1)*C1+VR(2)*C2+VR(3)*C3+VR(4)*C4
      DVDZ=VZ(1)*C1+VZ(2)*C2+VZ(3)*C3+VZ(4)*C4
      VPOT=V(1)*C1+V(2)*C2+V(3)*C3+V(4)*C4
   99 IPL=IL
      JPL=JL
      IF(RGIVEN)100,101,101
  100 DVDR=-DVDR
  101 RETURN
      END
```

```
      SUBROUTINE MAIN5
CMAIN5
C     02007 S15
C     MAIN PROGRAMME FOR PLOTTING RESULTS OF 02007
C
      COMMON C5,C95,NT3,UT3,NT4,HT4,KT4,NT5,NE3,NE4,NE5,NC,NR
      COMMON KBAS,DX,DY,JOBNUM,NSW1,NSW2,TIM,UMAT,XV,YV,NP,NV,NX,NY
      DIMENSION NT3(780),UT3(780),NT4(780),HT4(780,4),KT4(780),NT5(390,2
     1)
      DIMENSION UMAT(3250),V(15),NP(15),XV(200,15),YV(200,15)
C    TAPES USED IN THIS PROGRAM
      M= 3
      N=7
C
      READ 100,V(1),V(2),V(3),NV,NX,NY
      NVR=NV-3
      IF(NVR)7,7,5
    5 READ 101, (V(I+3),I=1,NVR)
    7 READ (N,102)           JOBNUM,DX,DY,NR,NC
      NPIA=NR*NC
      READ (N,103)           (UMAT(I),I=1,NPIA)
      READ (N,104)           NE3,NE4,NE5,(NT3(I),UT3(I),I=1,NE3)
      READ (N,105)           (NT4(I),(HT4(I,J),J=1,4),KT4(I),I=1,NE4)
      READ (N,106)           (NT5(I,1),NT5(I,2),I=1,NE5)
      IF(NV)26,26,10
   10 CALL EQUPOT(V,NV,NP,XV,YV)
      WRITE (M,107)
      NL=NV
      IF(NV-6)16,16,15
   15 NL=6
   16 WRITE (M,108)            (V(I),I=1,NL)
      CALL TABLE (XV,YV,NP,NL)
      IF(NV-6)26,26,20
   20 NL=NV-6
      IF(NV-12)22,22,21
   21 NL=6
   22 WRITE (M,108)            (V(I+6),I=1,NL)
      CALL TABLE(XV(1,7),YV(1,7),NP(7),NL)
      IF(NV-12)26,26,25
   25 NL=NV-12
      WRITE (M,108)            (V(I+12),I=1,NL)
      CALL TABLE (XV(1,13),YV(1,13),NP(13),NL)
C
C     END OF EQUIPOTENTIAL SECTION
C
   26 PRINT 30
   30 FORMAT(29H-ELEMENTS OF POTENTIAL MATRIX)
      PRINT 35,(UMAT(I),I,I=1,NPIA)
   35 FORMAT(4(F15.8,I5))
  100 FORMAT(3E15.6,3I5)
  101 FORMAT(3E15.6)
  102 FORMAT(I15,2E15.7,2I15)
  103 FORMAT(//(7E15.7))
  104 FORMAT(///3I15///(I6,E15.7))
```

```
105 FORMAT(///(I6,4F10.6,I6))
106 FORMAT(///(2I6))
107 FORMAT(1H1,30X,15H EQUIPOTENTIALS///)
108 FORMAT(///6(F13.6,7X)//)
    STOP
    END
```

```
      SUBROUTINE EQUPOT(V,NV,NP,XV,YV)
CEQUPOT
C     02007 S 12
      COMMON DUM,NC,NR,KBAS,DX,DY,JNUM,NSW1,NSW2,TIM,UMAT
      DIMENSION DUM(7025),UMAT(3250),NP(15),V(15),XV(200,15),YV(200,15)
      DO 2 K=1,NV
    2 NP(K)=0
      NRS=NR-1
      NCS=NC-1
      DO 45 IR=1,NRS
      DO 40 IC=2,NCS
      N=(IR-1)*NC+IC
      IF(ZEROIN(IR,IC))40,5,40
C
C     CURRENT POINT IS IN
C
    5 IF(ZEROIN(IR,IC+1))25,10,25
C
C     RIGHT-HAND POINT IS IN
C
   10 DO 20 K=1,NV
      IF((V(K)-UMAT(N))*(UMAT(N+1)-V(K)))20,15,15
   15 NP(K)=NP(K)+1
      I=NP(K)
      YV(I,K)=FLOAT (IR-1)*DY
      XV(I,K)=(FLOAT (IC-1)+(V(K)-UMAT(N))/(UMAT(N+1)-UMAT(N)))*DX
   20 CONTINUE
   25 IF(ZEROIN(IR+1,IC))40,27,40
C
C     POINT ABOVE IS IN
C
   27 DO 35 K=1,NV
      NN=N+NC
      IF((V(K)-UMAT(N))*(UMAT(NN)-V(K)))35,30,30
   30 NP(K)=NP(K)+1
      I=NP(K)
      XV(I,K)=FLOAT (IC-1)*DX
      YV(I,K)=(FLOAT (IR-1)+(V(K)-UMAT(N))/(UMAT(NN)-UMAT(N)))*DY
   35 CONTINUE
C
   40 CONTINUE
   45 CONTINUE
      RETURN
      END
```

```
      FUNCTION ZEROIN (I,J)
CZEROIN
C     02007 S13
C     ZEROIN IS SET NONZERO UNLESS POINT (I,J) IS IN TABLE 4 OR 5.
C
      COMMON C5,C95,NT3,UT3,NT4,HT4,KT4,NT5,NE3,NE4,NE5,NC,NR
      DIMENSION NT3(780),UT3(780),NT4(780),HT4(780,4),KT4(780),NT5(390,2
     1)
      ZEROIN=0.0
      NP=(I-1)*NC+J
      DO 55 K=1,NE5
      IF(NP-NT5(K,1))6,70,5
    5 IF(NT5(K,2)-NP)55,70,70
   55 CONTINUE
    6 DO 66 K=1,NE4
      IF(NP-NT4(K))66,70,66
   66 CONTINUE
      ZEROIN=1.0
   70 RETURN
      END
```

```
      SUBROUTINE TABLE (X,Y,NP,NV)                                    /391
CTABLE
C     02007 S14                                                       /390
      DIMENSION X(200,15),Y(200,15),NP(15)                            /392
      DIMENSION  BUF(12)
C     TAPES USED IN THIS PROGRAM
      NOUT1= 3
C                                                                     /393
C     NV MUST BE NOT GREATER THAN 6                                   /394
      N=NOUT1
      L=0                                                             /396
    1 L=L+1                                                           /397
      M=0                                                             /398
      NN=1
      DO 65 J=1,NV                                                    /399
      IF(L-NP(J))5,5,3                                                /400
    3 IF(J-1)65,4,65                                                  /401
    4 NN=2
      GO TO 65                                                        /403
    5 M=1                                                             /404
      GO TO (10,20,30,40,50,60),J                                     /405
   10 BUF(1)=X(L,J)
      BUF(2)=Y(L,J)
      NN=3
      GO TO 65                                                        /407
   20 BUF(3)=X(L,J)
      BUF(4)=Y(L,J)
      NN=3
      GO TO 65                                                        /409
   30 BUF(5)=X(L,3)
      BUF(6)=Y(L,3)
      NN=3
      GO TO 65                                                        /411
   40 BUF(7)=X(L,4)
      BUF(8)=Y(L,4)
      NN=3
      GO TO 65                                                        /413
   50 BUF(9)=X(L,5)
      BUF(10)=Y(L,5)
      NN=3
      GO TO 65                                                        /415
   60 BUF(11)=X(L,6)
      BUF(12)=Y(L,6)
      NN=3
   65 CONTINUE                                                        /417
      GO TO (91,80,90),NN
   80 WRITE (N,100)
      GO TO 91
   90 WRITE (N,1000)          (BUF(I),I=1,12)
      DO 92 I=1,12
   92 BUF(I)=0.
   91 IF(M)1,70,1                                                     /418
   70 RETURN                                                          /419
  100 FORMAT(20X)                                                     /420
```

```
1000 FORMAT(12F10.5)                                                    /427
     END
```

IV. Solution of the First-Order Paraxial Ray Equation for a Sheet Beam in a Uniform Magnetic Field.

In this chapter we shall be concerned with the motion of electrons in electric and magnetic fields. The problem involves the solution of the Lorentz force equation, Poisson's equation and the equation of continuity of current in a self-consistent manner. When certain symmetries are assumed, one can solve these equations to obtain certain special exact solutions. However, the design of electron beams involves a number of competing factors and it is seldom possible to satisfy the requirements of the beam with these special solutions. In general it is desirable to use approximate methods in which the effects of the various parameters may be studied and adjustments may be made on the beam specifications. For thin laminar beams the paraxial ray equation[6] is a differential equation which satisfies all the equations of space-charge flow approximately to first order in beam thickness. When the emission velocity is zero, we can write the first order paraxial ray equation for a sheet beam in a uniform magnetic field perpendicular to the plane of the trajectories as

$$2\varphi_0 r'' + \varphi_0' r' + \left(\varphi_0'' + 4k_0^2\varphi_0 + 2k_0 b\sqrt{2\eta\varphi_0} + \eta b^2\right) r = \frac{\pm I}{2\varepsilon_0 w\sqrt{2\eta\varphi_0}} \qquad (4.1)$$

where

φ_0 = axial potential

r = half-thickness of the beam measured along the axis of the beam

k_0 = axial curvature

b = magnetic field

I = total current in the beam

ε_0 = dielectric constant of free space

η = charge to mass ratio of an electron

w = width of the beam in the direction of the magnetic field

The primes indicate differentiation with respect to the arc length s along the axis of the beam. The sign of the beam half-thickness is positive above the axis and negative below the axis. The curvature k_0 is taken to be positive if the center of curvature is reached by moving in the positive r direction from the axis, and negative otherwise. The + sign must be chosen for the right hand side when r is positive and vice versa. We refer to Fig. 4. for an explanation of the symbols. In this report we shall use the following normalizations:

$$s = \frac{\eta J_{yk}}{\varepsilon_0 \omega_c^3} S \tag{4.2}$$

$$\varphi = \frac{\eta J_{yk}^2}{\varepsilon_0^2 \omega_c^4} \Phi \tag{4.3}$$

where J_{yk} is the cathode current density and $\omega_c = \eta b$.

The normalized equation is

$$2\Phi_0 R'' + \Phi_0' R' + (\Phi_0'' + 4k_0^2 \Phi_0 + 2k_0 \sqrt{2\Phi_0} + 1)R = \pm \frac{R_k}{\sqrt{2\Phi_0}} \tag{4.4}$$

where we have used the fact that

$$I = 2r_k w J_{yk} \tag{4.5}$$

where R_k is the half-thickness at the cathode. The lower-case symbols refer to unnormalized values and the upper-case symbols refer to normalized values.

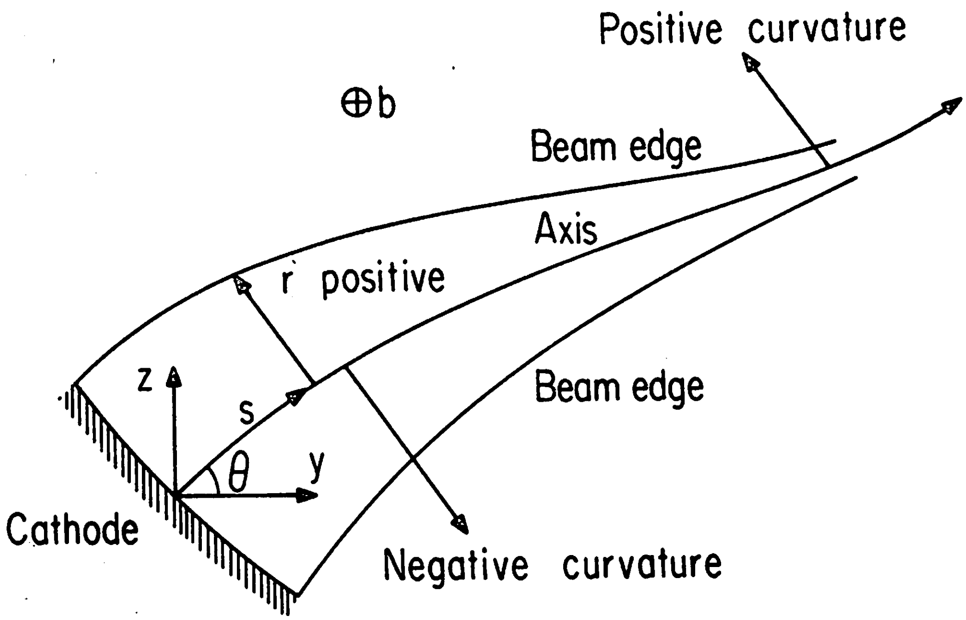

Fig. 4: Typical electron flow for generalized coordinates.

The equation may be solved for one of the parameters Φ_0, R, K_0 when the other two are specified. Therefore we have three different programs to solve Eq. (4.4).

A. Solution for axial and beam-edge potentials.

When the shape of the beam is specified, we know R and K_0 as functions of S. Then the paraxial ray equation may be solved for the potential Φ_0 along the axis. Furthermore we can obtain the potentials and fields along the beam edges. We have the relation:

$$\Phi = \Phi_0 + A_1 R + A_2 R^2 \tag{4.6}$$

The coefficients A_1 and A_2 are given by:

$$A_1 = 2\Phi_0 K_0 + \sqrt{2\Phi_0} + \frac{1}{2}\frac{R_k}{\sqrt{2\Phi_0}} \tag{4.7}$$

$$A_2 = -\frac{1}{2}\left(\Phi_0'' - 2\Phi_0 K_0^2 - K_0\sqrt{2\Phi_0}\right) \tag{4.8}$$

The input quantities are the axial curvature and the half-thickness measured along the axis. Since the equation for Φ_0 is a second-order differential equation we need initial conditions for Φ_0 and Φ_0' at the starting point of the solution. Integration is done by fourth-order Adams-Moulton Predictor-Corrector method. A short description of the program follows.

Name of the Main Program: INPUTP

Source Language: FORTRAN IV

Purpose: To calculate the potential along the axis and the potential and fields along the beam edge using the paraxial ray equation when the axial curvature and beam half-thickness are given.

Comments: The second derivative of the beam half-thickness and the curvature should be specified at equally spaced intervals along the axis. The half-thickness is found by numerical integration involving a Taylor series expansion. The curvature is integrated to obtain the slope of the axis. The beam half-thickness and the slope of the axis are used to calculate the coordinates of the points along the axis and the beam edge. If the code number is NCODE=0, only the parameters along the axis are determined and if NCODE=1, the beam edge parameters are also calculated.

The following programs should be in the deck:

INPUTP (Main Program)

AXIS1

PARAXP

DERIV

INTO

INT

Comment: The subroutines DERIV, INTO, INT are common to both Sections A, B and are listed only once at the end of Section A.

Explanation of input variables:

CARD

I Reads BCD(I), any comment of 78 columns or less in length to identify the computer run.

II Reads L2,NN,NNN,NCODE1 FORMAT(4I10)

L2 An integer which subdivides the intervals DELS at which the axial curvature (C1) and second derivative of half-thickness (R1) are given. (L2 = DELS/DS)

NN Maximum number of points spaced DS along the axis. (NN=(NNN-1)*L2+1)

NNN Number of points at which R1 and C1 are specified.

NCODE1 Equals "0" if only axial parameters are required.

Equals "1" if both axial and beam edge parameters are required.

III Reads RK,DELS,S1(1),S(1),THETA(1),PPSS(1),PPS(1),PP(1) FORMAT(8F10.5).

RK Cathode half-thickness.

DELS Interval at which the R1 and C1 are specified along the axis.

S1(1) Initial value of arc length along the axis.

S(1) Initial value of arc length along beam edge.

THETA(1) Angle between the tangent and the Y-axis at the starting point of the axis.

PPSS(1) Initial value of the second derivative of the potential (Φ_0'') on the axis. If not known, this may be set equal to zero.

PPS(1) Initial value of the first derivative of the potential (Φ_0').

PP(1) Initial value of potential (Φ_0).

IV Reads R(1),RS(1) FORMAT(2E15.8)

R(1) Initial value of beam half-thickness.

RS(1) Initial value of first derivative of beam half-thickness.

V Reads YY(1),ZZ(1),Y(1),Z(1) FORMAT(4F10.5)

YY(1) Initial value of Y-coordinate of beam axis.

ZZ(1) Initial value of Z-coordinate of beam axis.

Y(1) Initial value of Y-coordinate of beam edge.

Z(1) Initial value of Z-coordinate of beam edge.

VI Reads R1(I) (I = 1,NNN) FORMAT(5E15.8)

R1(I) Second derivative (with respect to arc length) of beam half-thickness.

VII Reads C1(I) (I=1,NNN)   FORMAT(5E15.8)

C1(I) Axial curvature.

DATA CARDS

Card Sequence	Format	1–10	11–20	21–30	31–40	41–50	51–60	61–70	71–80
1	13A6	Any comment occupying less than 78 columns							
2	4I10	L2	NN	NNN	NCODE1				
3	8F10.5	RK	DELS	S1(1)	S(1)	THETA(1)	PPSS(1)	PPS(1)	PP(1)
4	4F10.5	YY(1)	ZZ(1)	Y(1)	Z(1)				
5	5E15.8	R1(1)	R1(2)	R1(3)	....				
		....							
6	5E15.8	C1(1)	C1(2)	C1(3)	....				

```
      FORTRAN IV PROGRAM   INPUTP(INPUT,OUTPUT)
C     THIS IS THE MAIN PROGRAM FOR CALCULATION OF POTENTIAL ON THE AXIS
C     USING THE PARAXIAL RAY EQUATION
      DOUBLE PRECISION DS
      COMMON    PHISS,DS,DELS,NNN,NN,L2,NCODE1,S1,C1,R1,SS,C,R,CS,RS,
     1RSS,PP,PPS,PPSS,THETA,YY,ZZ,S,Y,Z,P,PS,PR,RK,L
      DIMENSION S1(100),C1(100),R1(100),SS(800),C(800),R(800),CS(800),
     1RS(800),RSS(800),PP(800),PPS(800),PPSS(800),THETA(800),YY(800),
     2ZZ(800),S(800),Y(800),Z(800),P(800),PS(800),PR(800)
      DIMENSION BCD(13)
      READ 5,(BCD(I),I=1,13)
    5 FORMAT(13A6)
      PRINT 10,(BCD(I),I=1,13)
   10 FORMAT(1H1,13A6)
      READ 15,L2,NN,NNN,NCODE1
   15 FORMAT(4I10)
      PRINT 16,L2,NN,NNN,NCODE1
   16 FORMAT(16H-L2 = DELS/DS = I4,50H NN = MAX NUMBER OF POINTS SPACED
     XDS ALONG AXIS = I4/36H NNN = (TOTAL ARC LENGTH/DELS+1) = I4,10H N
     XCODE1 = I4)
      READ 20,RK,DELS,S1(1),S(1),THETA(1),PPSS(1),PPS(1),PP(1)
   20 FORMAT(8F10.5)
      PRINT 21,S1(1),S(1),THETA(1),PPSS(1),PPS(1),PP(1)
   21 FORMAT(10H- SS(1) = F10.5,5X,7HS(1) =  F10.5,5X,11HTHETA(1) =
     XF10.5/,5X,10HPPSS(1) =  F10.5,5X,9HPPS(1) =  F10.5,5X,8HPP(1) =
     XF10.5)
      DS=DELS/FLOAT(L2)
      PRINT 25,RK,DS
   25 FORMAT(31H-RK = CATHODE HALF-THICKNESS = F5.2,54H          DS = IN
     XCREMENTAL DISTANCE FOR INTEGRATION = D6.3)
      READ 30,R(1),RS(1)
   30 FORMAT(2E15.8)
      READ 35,YY(1),ZZ(1),Y(1),Z(1)
   35 FORMAT(4F10.5)
      READ 40,(R1(J),J=1,NNN)
      READ 40,(C1(J),J=1,NNN)
   40 FORMAT(5E15.8)
      C(1)=C1(1)
      SS(1)=S1(1)
      RSS(1)=R1(1)
      CALL AXIS1
      CALL PARAXP
      STOP
      END
```

```
      SUBROUTINE AXIS1
      DOUBLE PRECISION DS
      COMMON    PHISS,DS,DELS,NNN,NN,L2,NCODE1,S1,C1,R1,SS,C,R,CS,RS,
     1RSS,PP,PPS,PPSS,THETA,YY,ZZ,S,Y,Z,P,PS,PR,RK,L
      DIMENSION S1(100),C1(100),R1(100),SS(800),C(800),R(800),CS(800),
     1RS(800),RSS(800),PP(800),PPS(800),PPSS(800),THETA(800),YY(800),
     2ZZ(800),S(800),Y(800),Z(800),P(800),PS(800),PR(800)
      DIMENSION COST(800),SINT(800)
      COST(1)=COS(THETA(1))
      SINT(1)=SIN(THETA(1))
      N1=NNN-1
      N2=NNN-2
      L3=2*L2+1
      DO 30 J=1,N1
   30 S1(J+1)=S1(J)+DELS
      DO 35 J=1,N2
      RSSS=(R1(J+1)-R1(J))/DELS
      B1=(C1(J+2)-2.*C1(J+1)+C1(J))/(2.*DELS**2)
      B2=(C1(J+1)-C1(J))/DELS-B1*(2.*S1(J)+DELS)
      B3=C1(J)-B1*S1(J)**2-B2*S1(J)
      DO 35 K=1,L2
   20 I=K+L2*(J-1)
      SS(I+1)=SS(I)+DS
      RSS(I+1)=RSS(I)+RSSS*DS
      RS(I+1)=RS(I)+RSS(I)*DS+.5*RSSS*DS**2
      R(I+1)=R(I)+RS(I)*DS+.5*RSS(I)*DS**2
      C(I)=B1*SS(I)**2+B2*SS(I)+B3
      CS(I)=2.*B1*SS(I)+B2
      THETA(I+1)=THETA(I)+B1*(SS(I+1)**3-SS(I)**3)/3.+B2*(SS(I+1)+SS(I))
     X*DS/2.+B3*DS
      COST(I+1)=COS(THETA(I+1))
      SINT(I+1)=SIN(THETA(I+1))
      COSTDS=(COST(I+1)-COST(I))/DS
      SINTDS=(SINT(I+1)-SINT(I))/DS
      YY(I+1)=YY(I)+COST(I)*DS+0.5*COSTDS*DS**2
      ZZ(I+1)=ZZ(I)+SINT(I)*DS+0.5*SINTDS*DS**2
      IF(NCODE1-1)21,31,31
   31 Y(I)=YY(I)-R(I)*SINT(I)
      Z(I)=ZZ(I)+R(I)*COST(I)
      IF(I-1)32,32,34
   32 GO TO 35
   34 DY=Y(I)-Y(I-1)
      DZ=Z(I)-Z(I-1)
      DSS=SQRT(DY**2+DZ**2)
      S(I)=S(I-1)+DSS
   21 IF(J-N2)35,22,35
   22 IF(K-L2)35,23,23
   23 K=K+1
      IF(K-L3)24,36,36
   24 GO TO 20
   35 CONTINUE
   36 IF(NCODE1-1)45,40,40
   40 PRINT 50
   50 FORMAT(8H-       SS7X,5HTHETA6X,2HYY8X,2HZZ8X,1HS9X,1HY9X,1HZ)
      PRINT 55,(SS(I),THETA(I),YY(I),ZZ(I),S(I),Y(I),Z(I),I=1,NN1)
```

```
55 FORMAT(7F10.5)
   GO TO 70
45 PRINT 60
60 FORMAT(8H-      SS7X,5HTHETA6X,2HYY8X,2HZZ)
   PRINT 65,(SS(J),THETA(J),YY(J),ZZ(J),J=1,NN)
65 FORMAT(4F10.5)
70 RETURN
   END
```

```
      SUBROUTINE PARAXP
C     THIS SUBROUTINE SOLVES THE PARAXIAL EQUATION FOR AXIAL POTENTIAL
C     WHEN THE AXIAL CURVATURE AND BEAM THICKNESS ARE GIVEN AS FUNCTIONS
C     OF ARC LENGTH ALONG THE AXIS.
      DOUBLE PRECISION DS,TEMP
      LOGICAL SWPR
      EXTERNAL DERIV
      DIMENSION ZV(2),ZPRIME(2),TEMP(2,8)
      COMMON   PHISS,DS,DELS,NNN,NN,L2,NCODE1,S1,C1,R1,SS,C,R,CS,RS,
     1RSS,PP,PPS,PPSS,THETA,YY,ZZ,S,Y,Z,P,PS,PR,RK,L
      DIMENSION S1(100),C1(100),R1(100),SS(800),C(800),R(800),CS(800),
     1RS(800),RSS(800),PP(800),PPS(800),PPSS(800),THETA(800),YY(800),
     2ZZ(800),S(800),Y(800),Z(800),P(800),PS(800),PR(800)
      MM=2
      T=SS(1)
      ZV(1)=PP(1)
      ZV(2)=PPS(1)
      VV=SQRT(2.*PP(1))
      B1=RK/VV/R(1)
      B2=-2.*PP(1)*RSS(1)/R(1)
      B3=-PPS(1)*RS(1)/R(1)
      B4=-4.*C(1)**2*PP(1)
      B5=-2.*C(1)*VV-1.
      PPSS(1)=B1+B2+B3+B4+B5
      PHISS=PPSS(1)
      CALL INTO(MM,T,DERIV,ZV,ZPRIME,TEMP,DS)
      PRINT 10,ZV(1),ZV(2),ZPRIME(1),ZPRIME(2)
   10 FORMAT( 9H-PP(1) =  F10.6,14H    PPS( 1) = F10.6,14H    PPS( 1) =
     XF10.6,15H    PPSS( 1) = F10.6)
      NN1=NN-1
      DO 15 I=1,NN1
  405 PP(I)=ZV(1)
      PPS(I)=ZV(2)
      VV=SQRT(2.*PP(I))
      B1=RK/VV/R(I)
      B2=-2.*PP(I)*RSS(I)/R(I)
      B3=-PPS(I)*RS(I)/R(I)
      B4=-4.*C(I)**2*PP(I)
      B5=-2.*C(I)*VV-1.
      PPSS(I)=B1+B2+B3+B4+B5
      PHISS=PPSS(I)
      CALL INT(T,DERIV,ZV,ZPRIME,TEMP,SWPR)
      IF(.NOT.SWPR) GO TO 405
      PPSSS=(PPSS(I)-PPSS(I-1))/DS
      A1=2.*PP(I)*C(I)+VV
      A2=-0.5*(PHISS-A1*C(I)-B1)
        P(I)=PP(I)+A1*R(I)+A2*R(I)**2
      PR(I)   =A1+2.*A2*R(I)
      AS1=2.*PP(I)*CS(I)+2.*C(I)*PPS(I)+PPS(I)/VV
      PS(I)    =PPS(I)+AS1*R(I)+A1*RS(I)+2.*A2*R(I)*RS(I)-0.5*(PPSSS-CS(I)
     X*A1-C(I)*AS1+B1*RS(I)/R(I)+RK/R(I)*PPS(I)/VV**3)*R(I)**2
   15 CONTINUE
      PRINT 20
   20 FORMAT(38H1PARAMETERS ALONG THE AXIS OF THE BEAM)
      PRINT 25
```

```
25 FORMAT(9H-        SS11X,1HC12X,2HCS11X,1HR12X,2HRS11X,3HRSS10X,
  X2HPP11X,3HPPS10X,4HPPSS)
   PRINT 30,(SS(I),C(I),CS(I),R(I),RS(I),RSS(I),PP(I),PPS(I),PPSS(I),
  XI=1,NN1)
30 FORMAT(9E13.5)
   IF(NCODE1-1)80,40,40
40 PRINT 45
45 FORMAT(49H1PARAMETERS ALONG THE RIGHT HAND EDGE OF THE BEAM)
   PRINT 50
50 FORMAT(9H-        S11X,1HY12X,1HZ12X,1HP12X,2HPS11X,2HPR)
   PRINT 55,(S(I),Y(I),Z(I),P(I),PS(I),PR(I),I=1,NN1)
55 FORMAT(6E13.5)
80 RETURN
   END
```

```
SUBROUTINE DERIV(T,Z,ZPRIME)
DIMENSION Z(2),ZPRIME(2)
DOUBLE PRECISION DS
COMMON PHISS,I
ZPRIME(1)=Z(2)
ZPRIME(2)=PHISS
BB=T
RETURN
END
```

```
      FORTRAN IV SUBROUTINE INTO(NO,X,DERI ,Y,F,T,HPRO)
      COMMON /INTC/ IPMX,AREF,EMAX,SSSR,HFAC,SWAM,SWEX
      COMMON /INTP/ HPR,XX,N,EUB,ELB,IP,IT,NRKS,SWIN
      DIMENSION Y(1),F(1),T(8,1)
      LOGICAL SWAM,SWEX,SWIN
      INTEGER HFAC
      DOUBLE PRECISION T,HPRO,HPR,XX
      DATA IPMX,AREF,EMAX,SSSR,HFAC,SWAM,SWEX
     $     /1024,1.0,1.0E-6,100.0,2,.TRUE.,.TRUE./
C
      HPR=HPRO
      XX=DBLE(X)
      N=NO
      EUB=EMAX
      ELB=EMAX/SSSR
      IP=1
      IT=0
      NRKS=0
      SWIN=SWEX
      CALL DERI (X,Y,F)
      DO 9 I=1,N
      T(5,I)=DBLE(Y(I))
    9 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE INT(X,DERI ,Y,F,T,SWPR
     X )
      COMMON /INTC/ IPMX,AREF,EMAX,SSSR,HFAC,SWAM,SWEX
      COMMON /INTP/ HPR,XX,N,EUB,ELB,IP,IT,NRKS,SWIN
C
      DIMENSION Y(1),F(1),T(8,1)
      LOGICAL SWAM,SWEX,SWIN
      LOGICAL SWPR
      INTEGER HFAC
      DOUBLE PRECISION T,HPR,XX
      DOUBLE PRECISION D,H
 6000 FORMAT (36H0 CANNOT DECREASE H BECAUSE OF HMIN.    ,1PE16.8,I20)
C
 1    CONTINUE
      SWPR=.FALSE.
      TEST=0.0
      H=HPR/DBLE(FLOAT(IP*24))
      IF ((NRKS .LT. 3) .OR. (.NOT. SWAM)) GO TO 20
C
C     ADAMS-MOULTON STEP.
 100  CONTINUE
      DO 109 I=1,N
      D=DBLE(F(I))
      T(4,I)=D
      Y(I)=SNGL(T(5,I)+H*(
     X 55.0D0*D-59.0D0*T(3,I)+37.0D0*T(2,I)- 9.0D0*T(1,I) ))
 109  CONTINUE
      X=SNGL(XX+24.0D0*H)
      CALL DERI (X,Y,F)
      DO 119 I=1,N
      D=DBLE(F(I))
      D=(     T(5,I)+H*(
     X  9.0D0*D+19.0D0*T(4,I)- 5.0D0*T(3,I)+        T(2,I) ))
      T(6,I)=D
      E=ABS(SNGL(D)-Y(I))/14.0
      TEST=AMAX1(E/AMAX1(AREF,ABS(SNGL(D))),TEST)
 119  CONTINUE
C
      GO TO 300
C
C     ZONNEVELD STEP.
 200  CONTINUE
      DO 209 I=1,N
      D=DBLE(F(I))
      T(4,I)=D
C     1
      Y(I)=SNGL(T(5,I)+H*(
     X 12.0D0*D                                          ))
 209  CONTINUE
      X=SNGL(XX+12.0D0*H)
      CALL DERI (X,Y,F)
      DO 219 I=1,N
      D=DBLE(F(I))
      T(6,I)=D
C     2
```

```
      Y(I)=SNGL(T(5,I)+H*(
     X 12.0D0*D                                              ))
 219  CONTINUE
      CALL DERI (X,Y,F)
      DO 229 I=1,N
      D=DBLE(F(I))
      T(7,I)=D
C     3
      Y(I)=SNGL(T(5,I)+H*(
     X 24.0D0*D                                              ))
 229  CONTINUE
      X=SNGL(XX+24.0D0*H)
      CALL DERI (X,Y,F)
      DO 239 I=1,N
      D=DBLE(F(I))
      T(8,I)=D
C     4
      Y(I)=SNGL(T(5,I)+H*(
     X 3.75D0*T(4,I)+5.25D0*T(6,I)+9.75D0*T(7,I)-0.75D0*D ))
 239  CONTINUE
      X=SNGL(XX+18.0D0*H)
      CALL DERI (X,Y,F)
      DO 249 I=1,N
      D=DBLE(F(I))
      E=ABS(SNGL(H*(
     X -16.0D0*T(4,I)+48.0D0*T(6,I)+48.0D0*T(7,I)+48. D0*T(8,I)
     X -128.0D0*D    )))
C     5
      D=(     T(5,I)+H*(
     X 4.0D0*T(4,I)+ 8.0D0*T(6,I)+ 8.0D0*T(7,I)+ 4.0D0*T(8,I)
     X ))
      T(6,I)=D
      TEST=AMAX1(E/AMAX1(AREF,ABS(SNGL(D))),TEST)
 249  CONTINUE
C
C     BOTH ADAMS-MOULTON AND ZONNEVELD METHODS CONTINUE FROM HERE.
 300  CONTINUE
      X=SNGL(XX+24.0D0*H)
      IF (TEST .LE. EUB) GO TO 310
      IF (IP*HFAC .GT. IPMX) GO TO 309
C
C     REPEAT STEP WITH SMALLER H.
      NRKS=0
      IP=IP*HFAC
      IT=IT*HFAC
      DO 305 I=1,N
      Y(I)=SNGL(T(5,I))
      F(I)=SNGL(T(4,I))
 305  CONTINUE
      GO TO 1
C
C     CANNOT DECREASE H BECAUSE OF HMIN.
 309  CONTINUE
      IF (.NOT. SWIN) GO TO 310
      PRINT 6000, X,IPMX
```

```
      SWIN=.FALSE.
C
 310   CONTINUE
C
C
C      ACCEPT CURRENT STEP.
C
C      XX STILL HAS NOT BEEN CHANGED SINCE ENTRY.
C      YY(XX) IS STILL IN T(5, ).
C      F(YY) IS IN T(4, ).
C
       IT=IT+1
       XX=XX+HPR/DBLE(FLOAT(IP))
       NRKS=MIN0(NRKS+1,4)
       DO 319 I=1,N
       D=T(6,I)
       T(5,I)=D
       Y(I)=SNGL(D)
 319   CONTINUE
       X=SNGL(XX)
      CALL DERI (X,Y,F)
       IF (IT .LT. IP) GO TO 320
C
C      X IS A MULTIPLE OF HPRINT.
       SWPR=.TRUE.
       IT=IT-IP
C
 320   CONTINUE
       IF (TEST .GE. ELB) GO TO 330
       IF (MOD(IP,HFAC)+MOD(IT,HFAC) .NE. 0) GO TO 33
C
C      PROCEED TO NEXT STEP WITH LARGER H, USING ZONNEVELD METHOD.
       NRKS=0
       IP=IP/HFAC
       IT=IT/HFAC
       RETURN
C
C
C      PROCEED TO NEXT STEP WITH SAME H.
 330   CONTINUE
       DO 339 I=1,N
       T(1,I)=T(2,I)
       T(2,I)=T(3,I)
       T(3,I)=T(4,I)
 339   CONTINUE
       RETURN
       END
```

B, Solution for beam thickness

When the shape of the axis is determined by specifying the curvature K_0 of the axis, and the potential Φ_0 is specified along the axis, the paraxial ray equation (4.4) may be solved for the beam half-thickness R. The convention of signs is the same as in section A. The initial conditions R and R′ at the starting point must be specified. A short description of the program follows.

Name of the Program: INPUTR

Source Language: FORTRAN IV

Purpose: To calculate the beam half-width using the paraxial ray equation when the axial potential and the axial curvature are specified.

Comments: The second derivative (with respect to arc length) of the axial potential and the axial curvature should be specified at equally spaced intervals of arc length along the axis. The axial potential is found by integration. The curvature is integrated to obtain the slope of the axis and hence the coordinates of the points along the axis.

The following programs should be in the deck.

INPUTR (Main Program)

PARAXR

DERIV

INTO

INT

Explanation of the input variables:

CARD

I Reads BCD(I), any comment of 78 columns or less in length to identify the computer run.

II Reads L2,NN,NNN FORMAT(3I10)

L2 An integer which subdivides the intervals DELS at which the axial potential and axial curvature are given. (L2=DELS/DS).

NN Maximum number of points spaced DS along the axis (NN=(NNN-1)*L2+1).

NNN Number of points at which second derivative of axial potential (P1) and axial curvature (C1) are specified.

III Reads RK,DELS,S1(1),THETA(1),RS(1),R(1) FORMAT(6F10.5)

RK Cathode half-thickness.

DELS Intervals at which P1 and C1 are specified.

S1(1) Initial value of the arc length along the axis.

THETA(1) Angle between the tangent and the Y-axis at the starting point of the axis.

RS(1) Initial value of the first derivative of the beam half-thickness.

R(1) Initial value of half-thickness.

IV Reads PPS(1),PP(1) FORMAT(2E15.8)

PPS(1) Initial value of the first derivative of axial potential (Φ_0').

PP(1) Initial value of axial potential (Φ_0).

V Reads P1(J) (J=1,NNN) FORMAT(5E15.8)

P1(J) Values of second derivative of axial potential.

VI Reads C1(J) (J=1,NNN) FORMAT(5E15.8)

C1(J) Values of axial curvature.

DATA CARDS

Column Number		10	20	30	40	50	60
Card Sequence	**Format**	**Variables**					
1	13A6	Any comment occupying less than 78 columns					
2	3I10	L2	NN	NNN			
3	6F10.5	RK	DELS	S1(1)	THETA(1)	RS(1)	R(1)
4	5E15.8	P1(1)	P1(2)	P1(3)	P1(4)	P1(5)	
		P1(6)	....				
5	5E15.8	C1(1)	C1(2)	C1(3)	C1(4)	C1(5)	
		C1(6)	....				

```
      FORTRAN IV PROGRAM INPUTR(INPUT,OUTPUT)
C     THIS IS THE MAIN PROGRAM FOR CALCULATION OF BEAM HALF-THICKNESS
C     FROM THE PARAXIAL RAY EQUATION USING AXIAL POTENTIAL AND AXIAL
C     CURVATURE
      DOUBLE PRECISION DS
      COMMON RRSS,DS,DELS,NNN,NN,L2,S1,C1,P1,SS,C,R,CS,RS,RSS,PP,PPS,
     XPPSS,THETA,YY,ZZ,S,Y,Z,RK
      DIMENSION S1(100),C1(100),P1(100),SS(800),C(800),R(800),CS(800),
     1RS(800),RSS(800),PP(800),PPS(800),PPSS(800),THETA(800),YY(800),
     2ZZ(800),S(800),Y(800),Z(800)
      DIMENSION BCD(13)
      READ 5,(BCD(I),I=1,13)
    5 FORMAT(13A6)
      PRINT 10,(BCD(I),I=1,13)
   10 FORMAT(1H1,13A6)
      READ 15,L2,NN,NNN
   15 FORMAT(3I10)
      PRINT 16,L2,NN,NNN
   16 FORMAT(16H-L2 = DELS/DS = I4,50H NN = MAX NUMBER OF POINTS SPACED
     XDS ALONG AXIS = I4/36H NNN = (TOTAL ARC LENGTH/DELS + 1) = I4)
      READ 20,RK,DELS,S1(1),THETA(1),RS(1),R(1)
   20 FORMAT(6F10.5)
      DS=DELS/FLOAT(L2)
      PRINT 25,RK,DS
   25 FORMAT(31H-RK = CATHODE HALF-THICKNESS = F5.2,54H              DS = IN
     XCREMENTAL DISTANCE FOR INTEGRATION = D10.4)
      READ 27,PPS(1),PP(1)
   27 FORMAT(2E15.8)
      READ 30,(P1(J),J=1,NNN)
      READ 30,(C1(J),J=1,NNN)
   30 FORMAT(5E15.8)
      C(1)=C1(1)
      SS(1)=S1(1)
      CALL PARAXR
      STOP
      END
```

```
      SUBROUTINE PARAXR
C       THIS SUBROUTINE SOLVES PARAXIAL RAY EQUATION FOR BEAM HALF
C      THICKNESS AND ALSO CALCULATES THE COORDIANTES OF THE AXIS AND
C      BEAM EDGE
      DOUBLE PRECISION DS,TEMP
      LOGICAL SWPR
      EXTERNAL DERIV
      COMMON RRSS,DS,DELS,NNN,NN,L2,S1,C1,P1,SS,C,R,CS,RS,RSS,PP,PPS,
     XPPSS,THETA,YY,ZZ,S,Y,Z,RK
      DIMENSION S1(100),C1(100),P1(100),SS(800),C(800),R(800),CS(800),
     1RS(800),RSS(800),PP(800),PPS(800),PPSS(800),THETA(800),YY(800),
     2ZZ(800),S(800),Y(800),Z(800)
      DIMENSION COST(800),SINT(800)
      DIMENSION ZV(2),ZPRIME(2),TEMP(8,2)
      COST(1)=COS(THETA(1))
      SINT(1)=SIN(THETA(1))
      N1=NNN-1
      DO 30 J=1,N1
   30 S1(J+1)=S1(J)+DELS
      N2=NNN-2
      N3=NN-L2-1
      PPSS(1)=P1(1)
      DO 35 J=1,N2
      PSSS=(P1(J+1)-P1(J))/DELS
      B1=(C1(J+2)-2.*C1(J+1)+C1(J))/(2.*DELS**2)
      B2=(C1(J+1)-C1(J))/DELS-B1*(2.*S1(J)+DELS)
      B3=C1(J)-B1*S1(J)**2-B2*S1(J)
      DO 35 K=1,L2
      I=K+L2*(J-1)
      SS(I+1)=SS(I)+DS
       C(I)=B1*SS(I)**2+B2*SS(I)+B3
      CS(I)=2.*B1*SS(I)+B2
      PPSS(I+1)=PPSS(I)+PSSS*DS
      PPS(I+1)=PPS(I)+PPSS(I)*DS+.5*PSSS*DS**2
      PP(I+1)=PP(I)+PPS(I)*DS+.5*PPSS(I)*DS**2
      THETA(I+1)=THETA(I)+B1*(SS(I+1)**3-SS(I)**3)/3.+B2*(SS(I+1)+SS(I))
     X*DS/2.+B3*DS
      COST(I+1)=COS(THETA(I+1))
      SINT(I+1)=SIN(THETA(I+1))
      COSTDS=(COST(I+1)-COST(I))/DS
      SINTDS=(SINT(I+1)-SINT(I))/DS
      YY(I+1)=YY(I)+COST(I)*DS+0.5*COSTDS*DS**2
   35 ZZ(I+1)=ZZ(I)+SINT(I)*DS+0.5*SINTDS*DS**2
      MM=2
      T=SS(1)
      ZV(1)=R(1)
      ZV(2)=RS(1)
      VV=SQRT(2.*PP(1))
      F1=RK/VV
      F2=-PPS(1)*RS(1)
      F3=-(PPSS(1)+4.*C(1)**2*PP(1)+2.*C(1)*VV+1.)*R(1)
      RSS(1)=(F1+F2+F3)/(2.*PP(1))
      RRSS=RSS(1)
      CALL INTO(MM,T,DERIV,ZV,ZPRIME,TEMP,DS)
      PRINT 40,ZV(1),ZV(2),ZPRIME(1),ZPRIME(2)
```

```
 40 FORMAT (8H-R(1) = F10.6,13H      RS(1) = F10.6,12H     RS(1) =
   XF10.6,13H    RSS(1) = F10.6)
    DO 45 I=1,N3
    I=I
405 R(I)=ZV(1)
    RS(I)=ZV(2)
    VV=SQRT(2.*PP(I))
    F1=RK/VV
    F2=-PPS(I)*RS(I)
    F3=-(PPSS(I)+4.*C(I)**2*PP(I)+2.*C(I)*VV+1.)*R(I)
    RSS(I)=(F1+F2+F3)/(2.*PP(I))
    RRSS=RSS(I)
    CALL INT(T,DERIV,ZV,ZPRIME,TEMP,SWPR)
    IF(.NOT.SWPR) GO TO 405
    Y(I)=YY(I)-R(I)*SINT(I)
    Z(I)=ZZ(I)+R(I)*COST(I)
 45 CONTINUE
    PRINT 50
 50 FORMAT(8H-      SS7X,5HTHETA6X,2HYY8X,2HZZ8X,1HY9X,1HZ9X,1HR9X,
   X2HPP8X,1HC)
    PRINT 55,(SS(I),THETA(I),YY(I),ZZ(I),Y(I),Z(I),R(I),PP(I),C(I),
   XI=1,N3)
 55 FORMAT(8F10.5,E12.4)
    PRINT 60
 60 FORMAT(9H1        SS11X,2HPP9X,3HPPS8X,4HPPSS7X,1HR11X,2HRS10X,
   X3HRSS)
    PRINT 65,(SS(I),PP(I),PPS(I),PPSS(I),R(I),RS(I),RSS(I),I=1,N3)
 65 FORMAT(7E12.4)
    RETURN
    END
    SUBROUTINE DERIV(T,Z,ZPRIME)
    DIMENSION Z(2),ZPRIME(2)
    DOUBLE PRECISION DS
    COMMON PHISS,I
    ZPRIME(1)=Z(2)
    ZPRIME(2)=PHISS
    BB=T
    RETURN
    END
```

C. Solution for axial curvature

When the beam half-thickness R and the axial potential Φ_0 are specified, the shape of the axis (i.e., the curvature of the axis) may be determined from the paraxial ray equation (4.4). The resulting equation is a simple quadratic equation in K_0 and when it is real, it represents a real solution. A short description of the program follows.

Name of the Main Program: CURVC

Source Language: FORTRAN IV

Purpose: To calculate the axial curvature from the paraxial ray equation when the axial potential and the beam half-thickness are given as functions of arc length along the axis.

Comments: The second derivative of the axial potential and beam half-thickness (with respect to arc length) should be given at equally spaced points along the axis. The initial values of potential and thickness and their first derivatives are also given, and potential and thickness are found as a function of arc length by numerical integration. The equation for the axial curvature is a quadratic equation and is easily solved.

The following programs should be in the deck:

CURVC (Main Program)

PARAXC

All the input quantities are read in the main program.

Explanation of the input variables:

CARD

I Reads BCD(I), any comment of 78 columns or less in length to identify the computer run.

II Reads L1,L,N FORMAT(3I5)

L1 Number of initial point on axis. It must be greater than 1.

L Number which relates 'DS' as a fraction of 'DELS', L=DELS/DS.

N Number of points spaced 'DELS' along axis.

III Reads RS(L2),R(L2),PPS(L2),PP(L2) FORMAT(4E15.8)

NOTE: The numbers L2 and L1 both refer to the same point on the axis. L1 is the initial value for the numbering of points spaced 'DELS,' and L2 is the initial value for the numbering of points spaced 'DS.' Input is specified at intervals of 'DELS' while integration is done and output is given at intervals of 'DS.' L1 and L2 are related by the formula L2=1+L*(L1-1).

RS(L2) Initial value of $\frac{\partial R}{\partial S}$

R(L2) Initial value of R, beam half-thickness

PPS(L2) Initial value of $\frac{\partial \Phi_0}{\partial S}$.

PP(L2) Initial value of Φ_0, beam potential on the axis

IV Reads RK,DS,SS(L2),YY(L2),ZZ(L2),THETA(L2),C(LX) FORMAT(7F10.5)

RK Cathode half-thickness.

DS Incremental distance along axis for integration and for output.

SS(L2) Initial value of arc length along beam axis.

YY(L2) Initial value of y-coordinate of beam axis.

ZZ(L2) Initial value of z-coordinates of beam axis.

THETA(L2) Angle (in radians) beam axis makes with y-axis.

C(LX) Curvature of axis at the point LX=L2-1, which corresponds to the value of arc length SS(L2)-1.

V Reads RSS1(J), J=L1,N (five numbers per card) FORMAT(5E15.8)

RSS1(J) Values of $\frac{\partial^2 R}{\partial S^2}$ spaced DELS along axis.

VI Reads PSS1(J), J L1,N (five numbers per card) FORMAT(5E15.8)

PSS1(J) Values of $\frac{\partial^2 \Phi}{\partial S^2}$ spaced DELS along axis

Column Number		10	20	30	40	50	60	70	80
Card Sequence	Format	Variables							
1	13A6	Any comment occupying less than 78 columns							
2	3I5	L1	L	N					
3	4E15.8	RS(L2)	R(L2)	PPS(L2)	PP(L2)				
4	7F10.5	RK	DS	SS(L2)	YY(L2)	ZZ(L2)	THETA(L2)	C(LX)	
5	5E15.8	RSS(L1)	RSS(L1+1)	....					
		....							
6	5E15.8	PSS1(L2)	PSS1(L1+1)	....					
		....							

```
      PROGRAM CURVC(INPUT,OUTPUT)
C     THIS IS THE MAIN CALLING PROGRAM FOR THE CURVATURE CALCULATION
C     FOR A CROSSED FIELD BEAM WITH A CONSTANT MAGNETIC FIELD
      DIMENSION PSS1(200),PPSS(900),PPS(900),PP(900),RSS1(200),RSS(900),
     1RS(900),R(900),YY(900),ZZ(900),C(900),CS(900),CIM(900),COST(900),
     2SINT(900),THETA(900),BCD(13),SS(800)
      COMMON L1,L2,L,N,LX,PSS1,RSS1,PPSS,PPS,PP,RSS,RS,R,YY,ZZ,C,CS,
     1THETA,RK,DS,SS
      READ 5, (BCD(I),I=1,13)
    5 FORMAT(13A6)
      PRINT 10,(BCD(I),I=1,13)
   10 FORMAT(1H1,13A6)
      READ 15,L1,L,N
   15 FORMAT(3I5)
      L2=1+L*(L1-1)
      READ 16,RS(L2),R(L2),PPS(L2),PP(L2)
   16 FORMAT(4E15.8)
      LX=L2-1
      READ 20,RK,DS,SS(L2),YY(L2),ZZ(L2),THETA(L2),C(LX)
   20 FORMAT(7F10.5)
      READ 25,(RSS1(J),J=L1,N)
      READ 25,(PSS1(J),J=L1,N)
   25 FORMAT(5E15.8)
      PRINT 30,L2,N,RK,DS
   30 FORMAT(44H-L2= NUMBER OF POINTS IN KINO FLOW REGION = I4/
     X47H-N = NUMBER OF POINTS SPACED DELS ALONG AXIS =  I4/
     X31H-RK = CATHODE HALF-THICKNESS = F5.2/
     X40H-DS = INCREMENTAL DISTANCE ALONG AXIS = F6.3)
      CALL PARAXC
      STOP
      END
```

```
      SUBROUTINE PARAXC
      DIMENSION PSS1(200),PPSS(900),PPS(900),PP(900),RSS1(200),RSS(900),
     1RS(900),R(900),YY(900),ZZ(900),C(900),CS(900),CIM(900),COST(900),
     2SINT(900),THETA(900),SS(800)
      COMMON L1,L2,L,N,LX,PSS1,RSS1,PPSS,PPS,PP,RSS,RS,R,YY,ZZ,C,CS,
     1THETA,RK,DS,SS
      COST(L2)=COS(THETA(L2))
      SINT(L2)=SIN(THETA(L2))
      YY(L2-1)=YY(L2)-COST(L2)*DS
      ZZ(L2-1)=ZZ(L2)-SINT(L2)*DS
      PPSS(L2)=PSS1(L1)
      RSS(L2)=RSS1(L1)
      DELS=DS*FLOAT(L)
      N1=N-1
      NN=L2+L*(N-L1)
      NN1=NN-1
      DO 38 J=L1,N1
      RSSS=(RSS1(J+1)-RSS1(J))/DELS
      PSSS=(PSS1(J+1)-PSS1(J))/DELS
      DO 38 I1=1,L
      I=I1+L*(J-1)
      SS(I+1)=SS(I)+DS
      RSS(I+1)=RSS(I)+RSSS*DS
      PPSS(I+1)=PPSS(I)+PSSS*DS
      RS(I+1)=RS(I)+RSS(I)*DS+0.5*RSSS*DS**2
      PPS(I+1)=PPS(I)+PPSS(I)*DS+0.5*PSSS*DS**2
      R(I+1)=R(I)+RS(I)*DS+0.5*RSS(I)*DS**2
   38 PP(I+1)=PP(I)+PPS(I)*DS+0.5*PPSS(I)*DS**2
      DO 40 I=L2,NN1
      VV=SQRT(2.*PP(I))
      G1=4.*PP(I)
      G2=2.*VV
      G3=2.*PP(I)*RSS(I)/R(I)+PPS(I)*RS(I)/R(I)+1.-RK/(R(I)*VV)+PPSS(I)
      G4=G2*G2
      G5=4.*G1*G3
      IF(G4-G5)43,42,42
   42 C(I)=(-G2+SQRT(G4-G5))/(2.*G1)
      CIM(I)=0.
      GO TO 425
   43 C(I)=-G2/(2.*G1)
      CIM(I)=SQRT(G5-G4)/(2.*G1)
  425 CS(I)=(C(I)-C(I-1))/DS
      THETA(I+1)=THETA(I)+C(I)*DS+0.5*CS(I)*DS**2
      COST(I+1)=COS(THETA(I+1))
      SINT(I+1)=SIN(THETA(I+1))
      COSTDS=(COST(I+1)-COST(I))/DS
      SINTDS=(SINT(I+1)-SINT(I))/DS
      YY(I+1)=YY(I)+COST(I)*DS+0.5*COSTDS*DS**2
      ZZ(I+1)=ZZ(I)+SINT(I)*DS+0.5*SINTDS*DS**2
   40 CONTINUE
      PRINT 445
  445 FORMAT(7H1     SS21X,1HR,14X,2HRS,14X,3HRSS,13X,2HPP,12X,3HPPS,11X,
     X4HPPSS)
      PRINT 450,(SS(I),R(I),RS(I),RSS(I),PP(I),PPS(I),PPSS(I),I=L2,NN)
  450 FORMAT(F8.2,F27.4,5F15.4)
```

```
      PRINT 455
  455 FORMAT(103H1 IF IMAGINARY C IS NOT ZERO, THEN SOLUTION NOT PHYSICA
     XL. CURVATURE SET EQUAL TO REAL PART OF SOLUTION.)
      PRINT 45
   45 FORMAT(9H-          SS11X,6HC REAL7X,11HC IMAGINARY8X,5HTHETA10X,
     12HYY13X,2HZZ13X,1HR14X,2HPP)
      PRINT 50,(SS(I),C(I),CIM(I),THETA(I),YY(I),ZZ(I),R(I),PP(I),I=L2,
     1NN1)
   50 FORMAT(8E15.6)
      RETURN
      END
```

REFERENCES

[1] P. T. Kirstein and J. S. Hornsby, "A Fortran Programme for the Numerical Analysis of Curvilinear Electrode Systems," Report No. CERN 63-16, Accelerator Research Division, CERN, Geneva, Switzerland, 25 April 1963.

[2] I. Langmuir, "The Effect of Space Charge and Residual Gases on Thermionic Currents in High Vacuum," Phys. Rev. 2, pp. 450-486 (1913).

[3] J. R. Pierce, "Rectilinear Electron Flow in Beams," Jour. App. Phys. 11, pp. 548-554 (1940).

[4] P. T. Kirstein, "On the Determination of the Electrodes Required to Produce a Given Electric Field Distribution Along a Prescribed Curve," Proc. IRE 46, pp. 1716-1722 (1958).

[5] B. A. Carré, "The Determination of the Optimum Accelerating Factor for Successive Over-Relaxation, "Computer Journal 4, p. 73 (1960).

[6] W. E. Waters, "Paraxial Properties of Crossed-Field Electron Sheet Beams," M. L. Report No. 603, Microwave Laboratory, Stanford University, Stanford (May 1959).