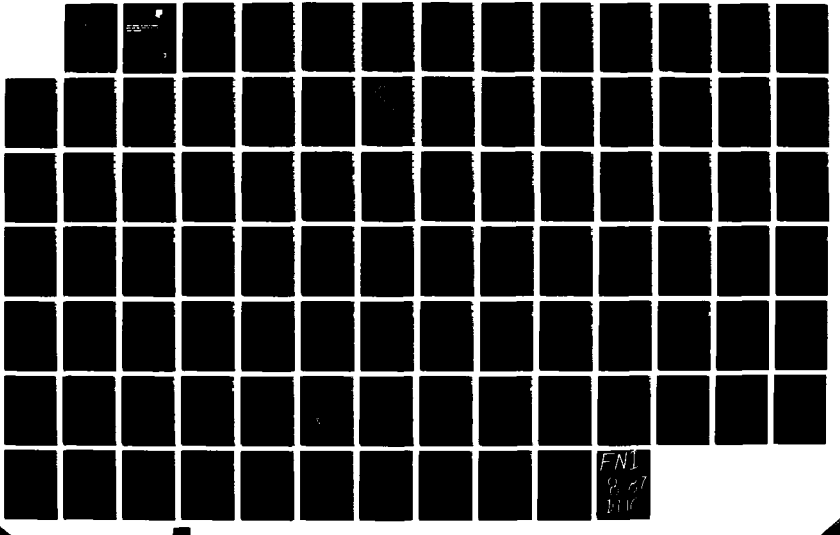
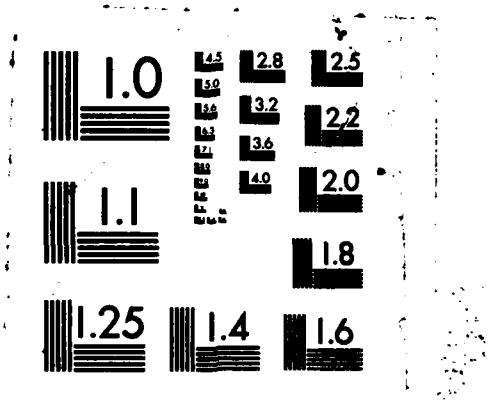
AD-A182 413 USERS MANUAL FOR NEW DEFORMABLE MESH GUN CODE BASED ON 'POISSON GROUP PRO..(U) UTAH UNIV SALT LAKE CITY DEPT OF ELECTRICAL ENGINEERING W W VOLGER MAY 87 1/1

UNCLASSIFIED UTEC-MD-86-070 RADC-TR-87-43 F/G 9/1 NL





(12)

AD-A182 413

DTIC FILE COPY

RADC-TR-87-43
Final Technical Report
May 1987

USERS MANUAL FOR NEW DEFORMABLE MESH GUN CODE BASED ON "POISSON GROUP PROGRAMS"

University of Utah

William W. Vogler

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700

87 006

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A182 413

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS N/A
2a. SECURITY CLASSIFICATION AUTHORITY N/A	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) N/A	5. MONITORING ORGANIZATION REPORT NUMBER(S) RADC-TR-87-43
6a. NAME OF PERFORMING ORGANIZATION University of Utah 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Rome Air Development Center (OCTP)
6c. ADDRESS (City, State, and ZIP Code) Department of Electrical Engineering Salt Lake City UT 84112	7b. ADDRESS (City, State, and ZIP Code) Griffiss AFB NY 13441-5700
8a. NAME OF FUNDING/SPONSORING ORGANIZATION AFOSR 8b. OFFICE SYMBOL (If applicable) NE	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F30602-82-C-0161
8c. ADDRESS (City, State, and ZIP Code) Bolling AFB Wash DC 20332	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 61102F; PROJECT NO. 2305; TASK NO J9; WORK UNIT ACCESSION NO. 16

11. TITLE (Include Security Classification)
USERS MANUAL FOR NEW DEFORMABLE MESH GUN CODE BASED ON "POISSON GROUP PROGRAMS"

12. PERSONAL AUTHOR(S)
William W. Vogler

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM Sep 82 TO Oct 86	May 1987	96

16. SUPPLEMENTARY NOTATION
Research accomplished in conjunction with Air Force Thermionics Engineering Research Program (AFTER) UTEC MD-86-070

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	
09	03		Gun Code; Space-Charge Flow; Traveling Wave Tubes; Deformable Mesh Programs; Electron Flow; Users Manual

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The Users Manual has been written to assist in the implementation of the Computer Code described in Microwave Device and Physical Electronics Laboratory Technical Report AFTER-14, "New Deformable Mesh Gun Code Based on 'POISSON' Group Programs,'" by William E. Vogler dated April 15, 1986. The Fortran 77 source code is available on magnetic tape. The Code and the Manual will be distributed on request to interested parties by the University of Utah or the Rome Air Development Center. (RADC-TR-87-21, "New Deformable Mesh Gun Code Based on 'Poisson Group Programs'" dated February 1987).

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Andrew E. Chrostowski, 1Lt, USAF	22b. TELEPHONE (Include Area Code) (315) 330-4381 22c. OFFICE SYMBOL RADC (OCTP)

DD Form 1473, JUN 86 Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

I. POISSON Group Programs

1.1 Introduction

The POISSON group programs are a set of computer codes that solve both electrostatic and magnetostatic problems using a deformable triangular mesh. This set of codes has been modified to solve the space-charge flow problem.

The first step in solving a problem using the POISSON codes is to construct a non-uniform triangulated mesh which represents a discretization of the problem. The program which generates the mesh is called LATTICE. LATTICE maps equilateral triangles in logical space into a non-uniform triangular mesh in real space.

Logical space consists of an array of regular equilateral triangles composed of three sets of straight lines intersecting each other at 60 degrees as shown in Fig. 1.1. Each vertex of a triangle has associated with it a logical coordinate (K,L) where K is the column order and L is the row order. LATTICE will assign each (K,L) coordinate within the boundary to an (x,y) coordinate in real space. To illustrate this mapping from logical space to real space, the logical diagram for two concentric circles is shown in Fig. 1.2a. For each logical coordinate on the boundary, a physical coordinate, (x,y) must be assigned to it. For the concentric circle problem, the logical coordinates on the boundary are assigned the physical coordinates of a circle. The process of mesh generation results in distorting triangle sides in the logical mesh to fit all region

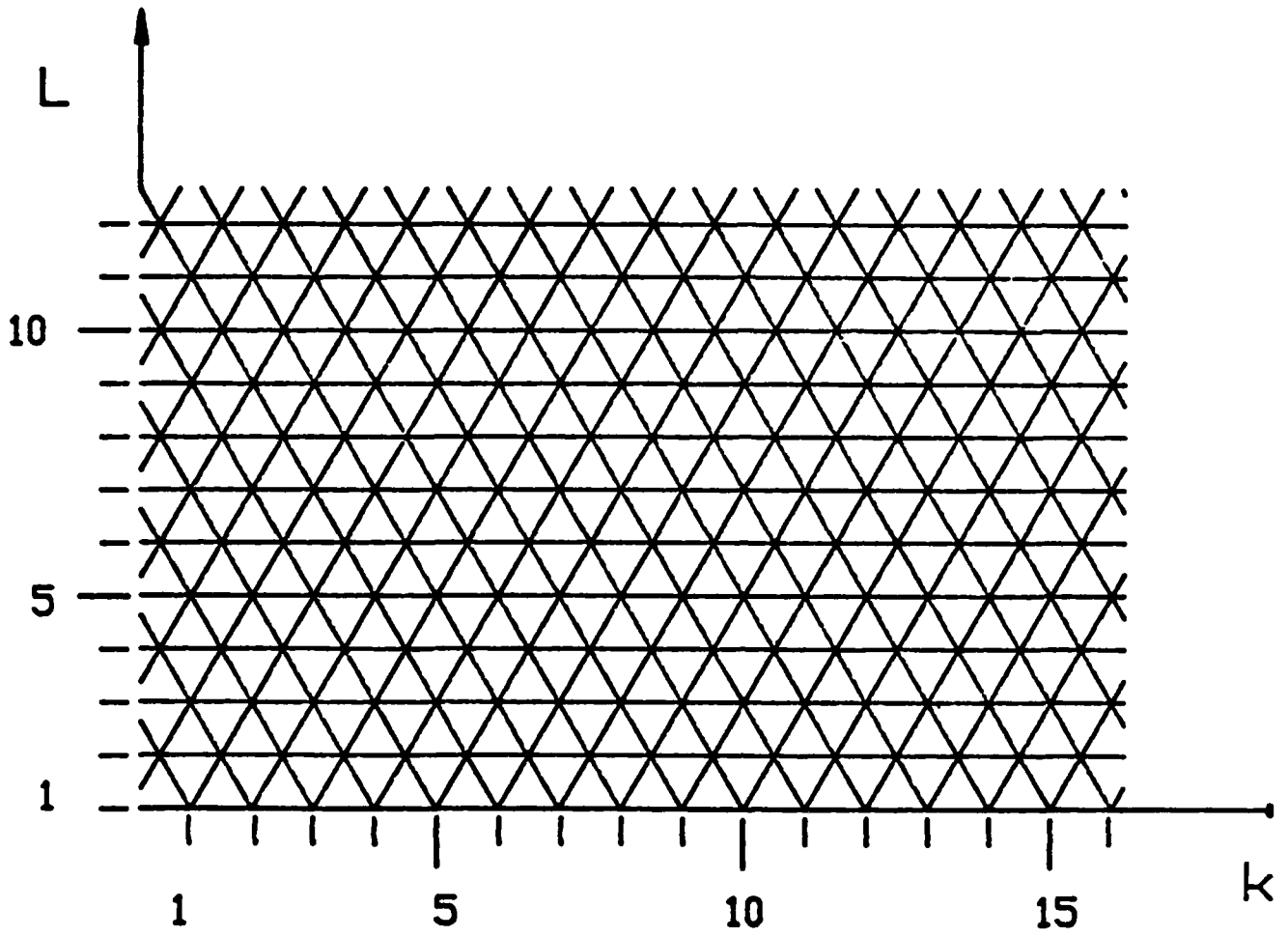
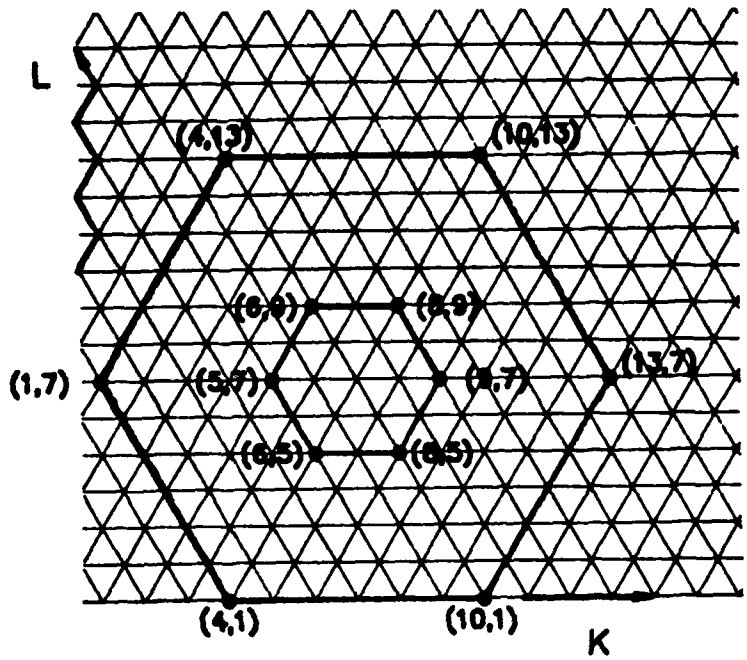
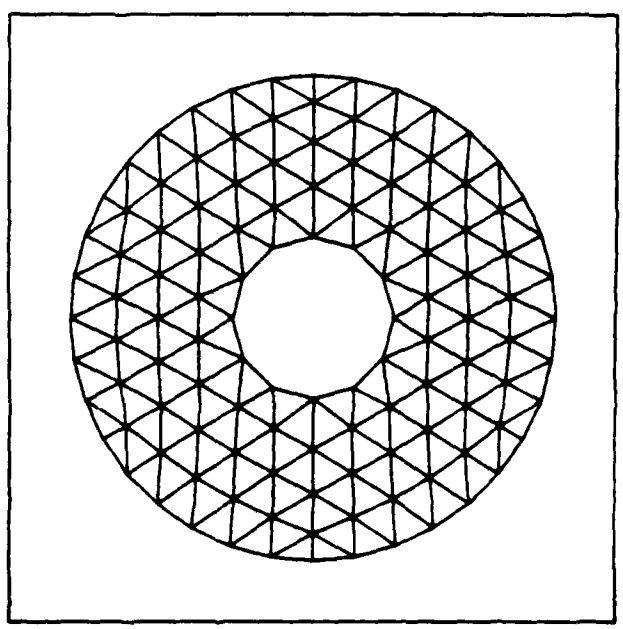


Fig. 1.1 Logic Space



a) Logical diagram.



b) Relaxed mesh.

Fig 1.2 a) Logic diagram for two concentric circles.
b) Relaxed triangulated mesh.

boundaries and determining the location of all interior mesh points. The final result of the mesh generation is shown in Fig. 1.2b. Note that the amount of triangles is conserved from the transformation from logical space to real space, but they are no longer equilateral.

1.2 POISSON Group Programs Used to Solve the Electron Flow Problem

The following is a list of programs with descriptions of their function in solving the space-charge flow problem:

1. AUTOMESH - prepares the input data for LATTICE from physical geometry data describing the problem. By defining the physical region boundaries and specifying the mesh size, AUTOMESH will construct the "logical" path in the mesh and generate (x,y) coordinates on the boundary for straight lines, arcs of circles and segments of hyperbolas.

2. LATTICE - generates an irregular triangular mesh from input data which includes "logical" and physical coordinates describing the boundary and boundary conditions.

3. POISSON - solves, by successive point over-relaxation, Poisson's equation for the scalar potential two-dimensional problems, calculates the derivatives of the

potential (fields and gradients), calculates the stored energy and performs harmonic (multipole) analysis of the potential.

1.3 Units Used In AUTOMESH, LATTICE, and POISSON

All calculations in the POISSON group programs are done in MKS units only. The user may use units other than MKS for length, but must appropriately set the scale factor to the correct value. This will be discussed later.

The following is a list of units used in the program:

length = (meters) or user defined by specifying Con(9) not equal to 1.0.

current = (ampere)

scalar potential V = (volt)

electric field E = (volt/m)

induction B = (tesla)

1.4 Format Free Input Routine

The POISSON group codes have a format-free input subroutine, FREE, which appears in the computer code as follows:

```
Subroutine FREE (N,RAY1,N1,RAY2,N2,RAY3,N3)
```

where RAY1, RAY2, and RAY3 are the array (and/or variable) names which are to be read in and N1, N2, and N3 are the numbers of

values RAY1, RAY2 and RAY3 are to have respectively. The variable N indicates how may names are to be read in, i.e., for N=1, then only RAY1(N1) will be read in.

When the program requires specific data to be read in, or certain variables to be modified, the following command will be executed:

```
CALL FREE (N,RAY1,N1,RAY2,N2,RAY3,N3)
```

The user will then input the data using a specific format.

Special input characters and their functions are described below:

"+" and "-"	indicate the sign of the number (mantissa or exponent). The "+" sign is optional.
"."	The decimal point is required for all floating point numbers.
"E"	The signed integer number following the letter "E" gives the power of ten to which the previous floating point number is raised. No blank before or after the "E" is allowed.
"*"I	Store the number following "I" into the array location (I). Successive input numbers will be stored into array locations (I), (I+1), (I+2), etc.
"R"N	Store the previous input number "N" times into the array locations (I), (I+1), (I+2), . . . through (I+N-1). A blank separating the previous input number and the "R" is optional.
"S"	Skip the rest of the N1, N2, or N3 values requested in the call and go on to the next array or return if the current array is the last array in the argument list.
"C"	Skip the rest of the N1, N2 or N3 values

requested in the call, set N1, N2 or N3 equal to the number of values input into the current array and go on to the next array or return. When using this feature, be sure that the N1, N2 or N3 in the calling statement is a variable and not a constant.

" " and "," — Blank and comma are the only non-numeric characters allowed in the input field, and one of these characters must be used to separate input values.

Comments may follow the last "S", "C" or required number on any input line (card).

An example is given below which illustrates most of the features in the format-free routine. The following is a calling sequence of FORTRAN statements; A(I) and B(I) are dimensioned arrays and K is a single variable.

```
DIMENSION A(5), B(100)
N=100
CALL FREE (3,A,5,B,N,K,1)
```

When the call to FREE is involved, the program requires input data given either from the terminal or input file. The following line is typed on the terminal for the above call to FREE:

```
-3.0,4.0  +5.3E-2 R2 S *20 0.1R10 COMMENTS HERE
```

Once this line has been read in A,B and K take on the following values,

```
A(1)=-3.0
A(2)=4.0
A(3)=0.0053
```

A(4)=0.0053
B(20) through B(29)=0.1
K=13

and the returned value of N is 10.

1.5 Program Flow Chart of the POISSON Group Programs

Fig. 1.3 shows the communication between the POISSON group programs listed in Sec. 1.2.

The boundary generator, AUTOMESH, requires the data file AUTOMESH.DAT which is created by the user. AUTOMESH will create the data file LATTICE.DAT, which is read in by LATTICE.

The mesh generator, LATTICE, requires the data file LATTICE.DAT which is either created by AUTOMESH or the user. LATTICE creates two files:

1. LATTICE.OUT, contains information on the mesh generation.
2. POISSON.DAT, contains the mesh and boundary information that is read by POISSON. LATTICE writes this data file in unformatted output, therefore it cannot be viewed since it is written in internal representation (binary).

The Poisson solver, POISSON, requires the data file POISSON.DAT. POISSON creates the following two files:

1. POISSON.OUT, which is written on unit 90, contains information on the solution of the electron-flow problem.

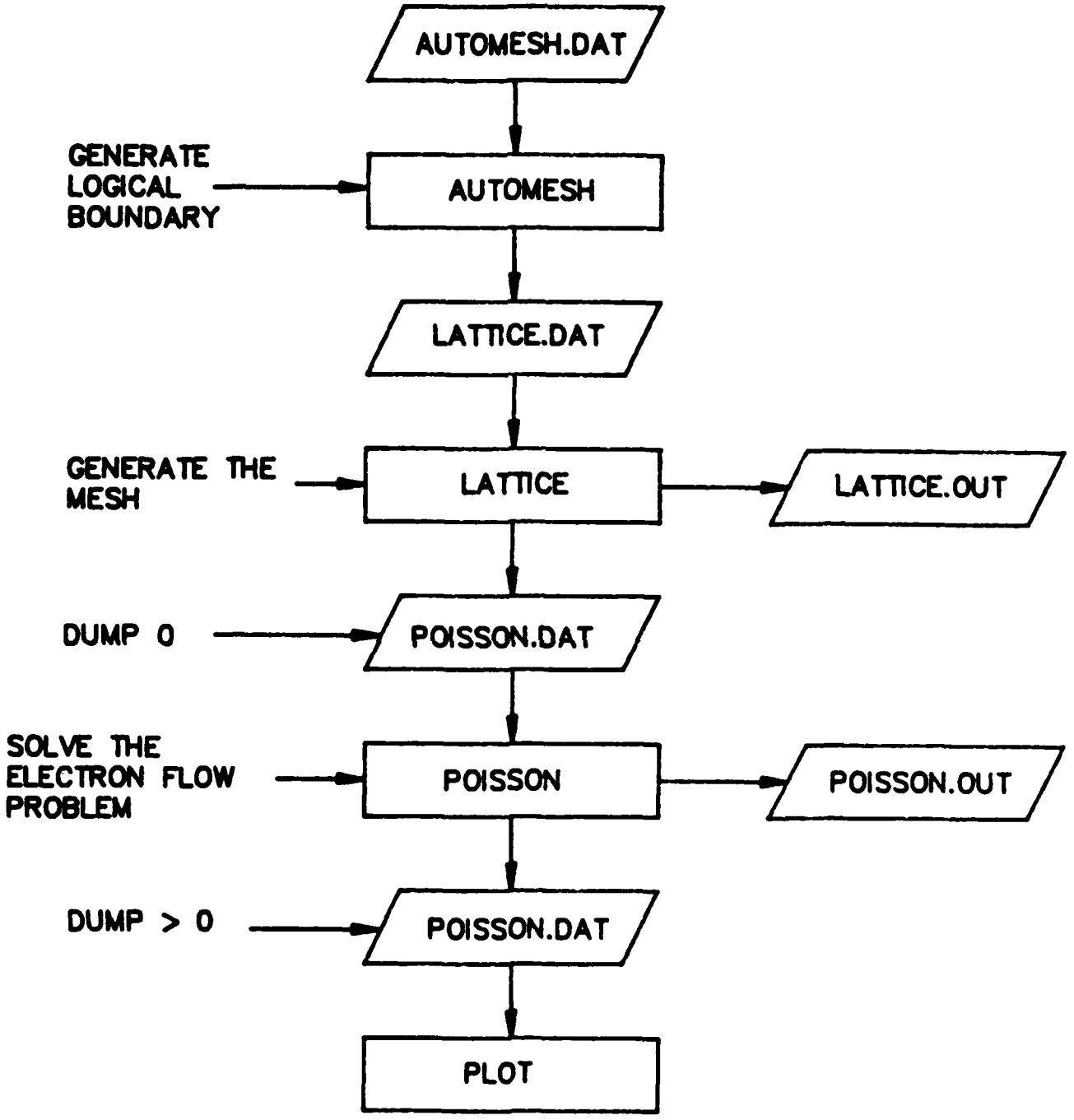


Fig. 1.3 Program Flow Chart of the POISSON Group Programs

2. POSITION.OUT, which is written on unit 91, contains the trajectory information and is optional.

The plotting program, PLOT, requires the data file POISSON.DAT which is read from unit 95.

II. AUTOMESH

The program AUTOMESH finds the path in the logical mesh which best fits the physical boundary. Once AUTOMESH has found this logical path, the physical coordinates (x,y) are assigned to each logical point (K,L). This program only determines the logical coordinates for the physical coordinates on the boundary. The program, LATTICE, determines the physical location of the logical coordinates within the boundary. AUTOMESH can be thought of as a preprocessor for LATTICE.

The algorithm in AUTOMESH which constructs the logical mesh from data describing the boundary of the problem can be visualized as follows:

1. Plot to scale the physical boundary of the problem.
2. Choose a triangle size for the mesh.
3. Overlay the array of identical triangles onto the physical boundary and determine the logical path.

This procedure of constructing the logical mesh is shown in Fig. 2.1.

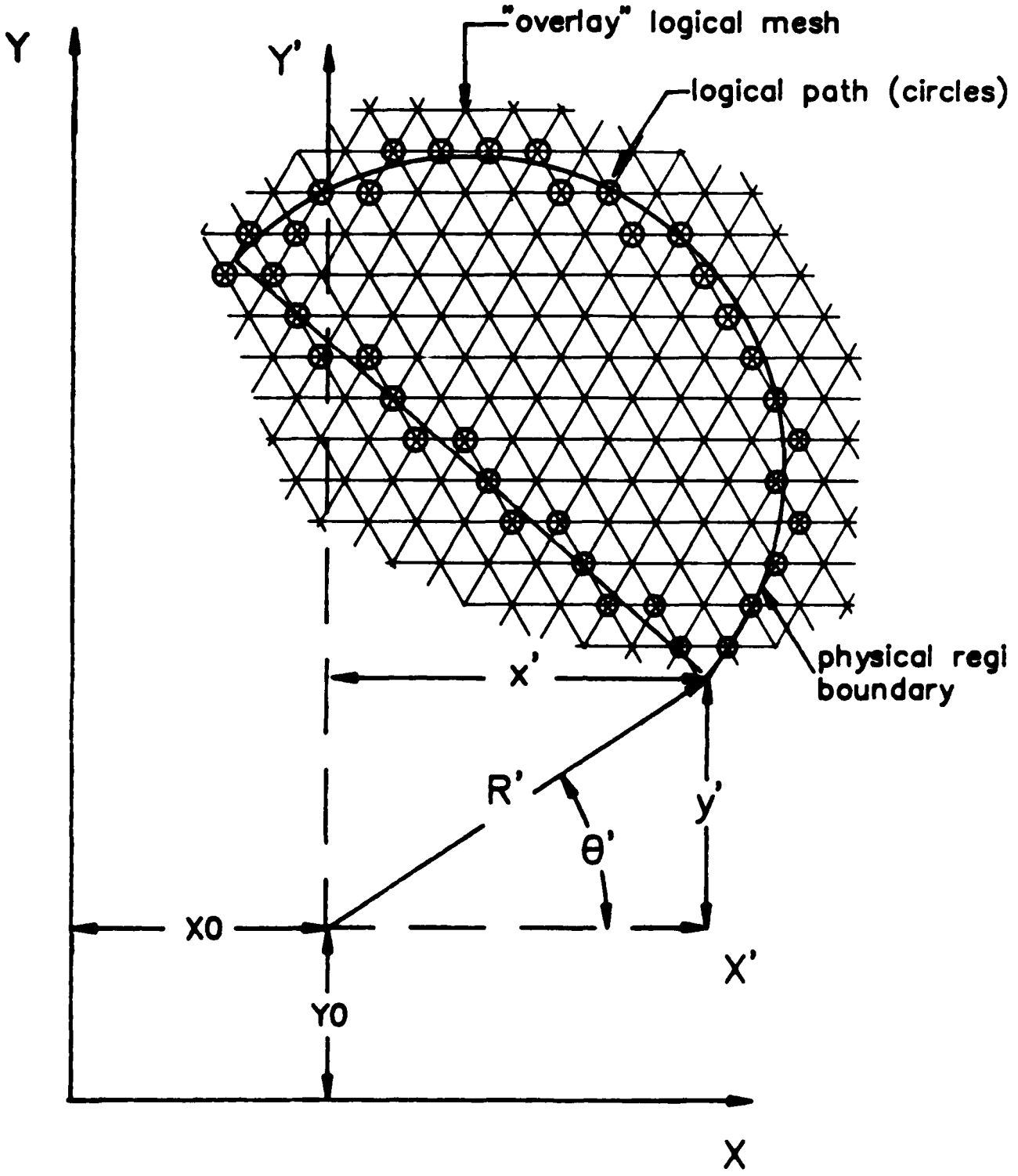


Fig. 2.1 AUTOMESH Coordinate System

2.1 Input Data For AUTOMESH

The input data for AUTOMESH is read by a "pseudo" namelist routine. There are three groups of input data for AUTOMESH which is described below in sections 2.3, 2.4 and 2.5. Data groups 2.4 and 2.5 are repeated for each region. The values of the variables for the first region are preserved for succeeding regions until they are changed in the next region.

2.2 Format of the Input

In the original AUTOMESH program, the input data was read in by the FORTRAN "NAMELIST" routine. Since the HP-1000 does not support NAMELIST, the program was rewritten with a "pseudo" namelist routine. This modified code tries to emulate the NAMELIST feature as closely as possible. The list of variables in the NAMELIST specifies those variables which may or may not have their values defined in the READ statement. Not all variables of the NAMELIST need be read in nor must they be in any order.

The very first item to be input is an integer constant which represents the number of input variables to be read in. The next item or items are the input variables. Each input variables is represented by a pair of FORTRAN variables. The first FORTRAN variable is a character variable which represents the name of the input variable and the second variable is the numerical value of the input variable. Since an unformatted READ statement is used

to read in the pair of variables, the character variable must be included in single quotes and either a blank or a comma must exist between each variable.

As an example, say the following three input variables are to be read in, or changed from their default values: 'NREG=', 'XMIN=', 'YMIN='. The following two lines would be typed in the data file AUTOMESH.DAT:

```
3
'NREG=' 4, 'XMIN=' 2.0, 'YMIN=' 1.0
```

The integer constant 3 in line one indicates that there are three input variables to be read in. The FORTRAN variable "NREG" will be assigned a value of 4 and the FORTRAN variables "XMIN" and "YMIN" will be equal to 2.0 and 1.0 respectively. Note the use of blanks and commas in line two. This is required since each item is a piece of data and must be separated by a blank or comma when using a unformatted input statement. The data items that appear in the input file need not appear in any special order and blanks may be used freely between data items to make the file easier to read. Also note that the three pair of variables in line two do not all have to be on the same line since the execution of an unformatted READ statement is not finished until all the listed variables are assigned values.

2.3 Title Line

The first data line can have anything in columns 2 through 80. For an electron flow problem, column one must be blank. If column one is non-blank, then this data file is for a SUPERFISH problem. If column one is blank, then this data file is for a POISSON problem. SUPERFISH is a separate program which solves for resonant modes in cavities. The characters in columns 2 through 80 are stored for run identification in the printed output.

2.4 Input Variables That Describe The Mesh and Boundary Conditions

The following is a reference list of input variables that describe the physical coordinates of the boundary:

NREG, IREG, MAT, CUR, DEN, ITRI, IBOUND, DX, DY, XMIN, XMAX, YMIN, YMIN, YMAX, NPOINT, IPRINT, XREG1, XREG2, YREG1, YREG2, KREG1, KREG2, KMAX, LREG1, LREG2, LMAX

A complete description of each variable is given below. Default values are shown in parentheses. If the user desires to use a default value, then omit the data items from the input file.

'NREG=' The number of regions to be read in this run (input only once in the data file).

'IREG=' The LATTICE region number. The default value is one for the 1st region data group and is

incremented by one for succeeding region data groups.

'MAT='
(1)

Material code for the region.

=0, all points inside this region are omitted.
=1, air (mu=1)
=2, iron with internal premeability table
=3, iron with input permeability table no. 1
=4, iron with input permeability table no. 2
=5, iron with input permeability table no. 3

'CUR='
(0.0)

The fixed potential value (volts) on the boundary of the region. For magnetic problems, it is the total current (amps) in the region.

'DEN='
(0.0)

The charge density (coloumbs/length**2) in the region.

'ITRI='
(0)

Triangle type generated in the region

=0, equal weight (default)
=1, equilateral
=2, right

'IBOUND='

Region boundary indicator (See LATTICE section for a more complete description)

= -1, fixed potential Dirichlet boundary condition
= 0, Dirichlet condition on the boundary (magnetic)
= 1, Neumann condition on the boundary

'DX='

The width of the triangles in the logical mesh.

'DY='

The height of the triangle in the logical mesh. Default values for ITRI=0, DY=DX*SQRT(3.0)/2.0, or DY=DX if ITRI=2.

'XMIN=' (0.0) — Minimum X value of the problem.

'XMAX=' — Maximum X value of the problem.

'YMIN=' (0.0) — Minimum Y value of the problem.

'YMAX=' — Maximum Y value of the problem.

'XREG1=' — The width of the triangles in the logical mesh will be set to approximately 2.0 * DX1 for X values greater than XREG1.

'XREG2=' — The width of the triangles in the logical mesh will be set to approximately 2.0 * DX1 for X values greater than XREG2.

'YREG1=' — The height of the triangles in the logical mesh will be set to approximately 2.0 * DY = DY1 for Y values greater than YREG1.

'YREG2=' — The height of the triangles in the logical mesh will be set to approximately 2.0 * DY1 for Y values greater than YREG2.

'KREG1=' 'KREG2=' 'KMAX=' 'LREG1=' 'LREG2=' 'LMAX=' — When the "doubling" schemes set up by the XREGn and YREGn variables are not appropriate, regions in the logical mesh with different triangle sizes may be explicitly defined by inputing the variables KREGn and LREGn. With these variables, one defines a logical mesh number, KREGn and/or LREGn, corresponding respecitvely to a physical distance, XREGn and/or YREGn. For example, if XREG1 = 1.0 and XMAX = 2.0, KREG = 21 and KMAX = 61, then DX1 = 0.05 and DX2 = 0.025.

'NPOINT=' — The number of curve end points describing the physical boundary of the region. This is the

number of data items used to describe the boundary.

'IPRINT=' (0) — If IPRINT = 1, then a special diagnostic printout is provided in the logical path finding routine.

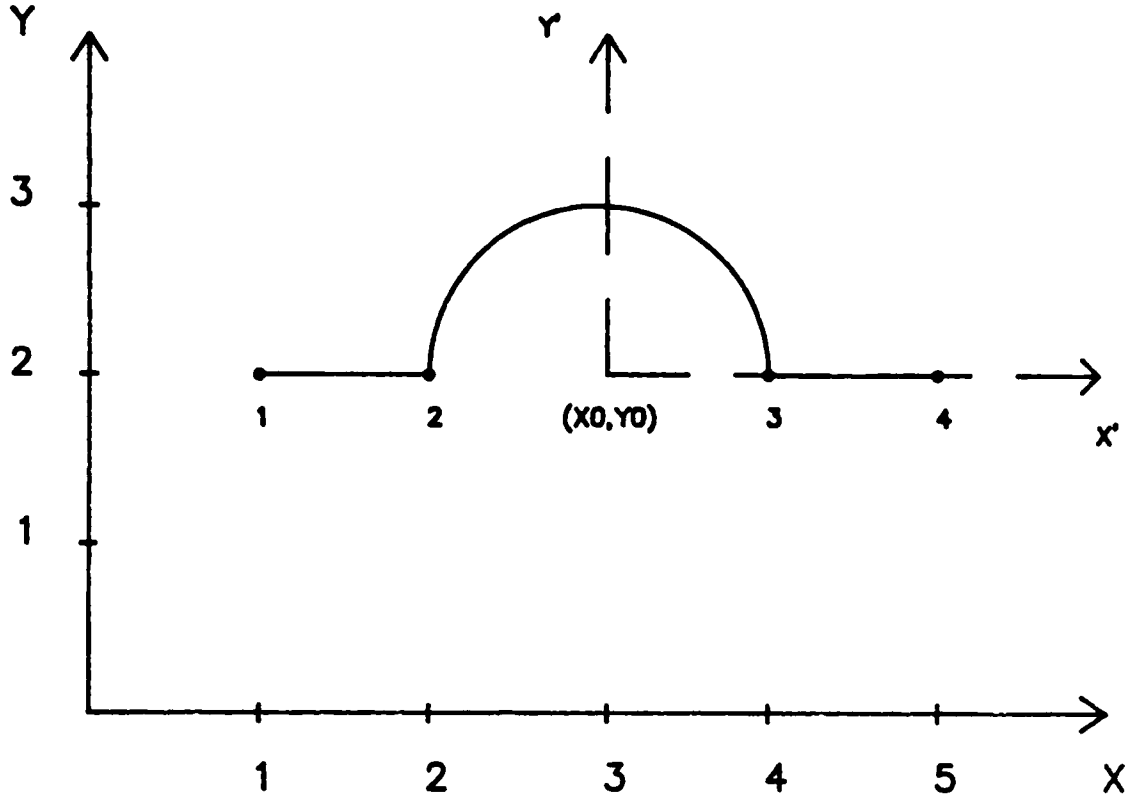
2.5 Input Variables Describing the Physical Coordinates of the Boundary

Each data line gives the coordinate of an end point of a curve on the physical boundary and specifies the type of curve to be drawn from the previous point to the current point. Note the first data line is the starting point of the boundary.

Coordinates for data points may be given as (x,y), (x',y'), (r, theta) or (r', theta') and may be freely intermixed (see Fig. 2.1). When defining an arc of a circle, the center of the circle (XO, YO) must be given and the end point of the arc is determined relative to (XO,YO). An example of an arc is illustrated in Fig. 2.2.

The following is a reference list of input variables that describe the physical coordinates of the boundary:

NT, XO, YO, X, Y, R, THETA, ICATH




```
2
'X=' 1.0, 'Y=' 2.0
2
'X=' 2.0, 'Y=' 2.0
5
'NT=' 2, 'X0=' 3.0, 'Y0=' 2.0, 'X=' 1.0, 'Y=' 0.0
2
'X=' 5.0, 'Y=' 2.0
```

Fig. 2.2 Example of a partial AUTOMESH file that describes two straight line segments and an arc of a circle.

A complete description of each variable is given below:

'NT=' (1)	Specifies the type of curve to be drawn from the previous curve endpoint to the current point. NT = 1, straight line NT = 2, an arc of a circle,(XO,YO) must be at the center of the circle. NT = 3, The hyperbola x'* y'= (r ** 2)/2.0, branch in the first quadrant. The input variable "R" must be set equal to the r in this equation.
'XO=' 'YO=' (0.0)	The displacement of the (x',y') coordinate axes, (0.0) and the center of the circle if "NT" = 2.
'X=' 'Y='	The (X',Y') coordinates of this curve's end point. If NT=2, these values are on the new (X',Y') axis.
'R=' 'THETA='	The (R',theta') coordinates of this curve end point.
'NEW=' (0)	NEW = 1, the points on the logical path for this segment are not allowed to coincide with the logical boundary points of any previous region. NEW = -1, the points on the logical path for this segment are not allowed to coincide with the logical boundary points of any previouys region, except for the starting and ending points.
'ICATH=' (0)	Specifies if this node is to be an emitting node or not. =0 Non-emitting node =1 Emitting node

2.6 Boundary Condition Data Set-Up by AUTOMESH

The boundary condition indicators, CON(21) through CON(24) = NBSUP, NBSLO, NBSRT, NBSLF, are set by AUTOMESH in the LATTICE.DAT "problem-constant" data as follows:

POISSON Problem:

NBSUP = 0
NBSLO = 1
NBSRT = 0
NBSLF = 0

SUPERFISH Problem:

NBSUP = 1
NBSLO = 0
NBSRT = 1
NBSLF = 1

The region special boundary indicators, = C(6) in the LATTICE.DAT data file, are set by AUTOMESH to the following default values:

POISSON Problem:

C(6) = 1, for all regions (Neumann condition).

SUPERFISH Problem:

C(6) = (1) for all regions

NOTE: These default values for C(6) may be over-written by inputing "IBOUND" in the AUTOMESH region input data. See Sec. 3.6 of LATTICE for a description of the special region boundary indicator.

2.7 Sample Input Data for AUTOMESH.DAT for the Problem "MODGUN"

The first step in using AUTOMESH is to determine the coordinates of various corner points of the boundary in the z-r plane. The dimensional diagram of the electrodes for the Pierce gun, MODGUN, is shown in Fig. 2.3. The boundary of MODGUN with it's corner points is shown in Fig. 2.4. The next step is to choose a triangle size for the mesh. This is accomplished by specifying the input variables DX and DY. Many times it is only necessary to specify DX and use the default value of DY, but in this example, MODGUN has some corners which require finer detail than that given by the default value of DY.

Choosing a proper DX and/or DY is important so that AUTOMESH can construct a consistent logical mesh. If DX and/or DY is too large, AUTOMESH will not be able to find a logical path along the boundary. What usually happens when a poor DX and DY has been chosen is that AUTOMESH will assign the same logical coordinate to two different physical coordinates which results in the program being aborted. When choosing values of DX and DY, a good rule of thumb is not to use a value much larger than the smallest difference in "X" and "Y" distances between boundary corners.

Once DX and DY has been chosen, the data file AUTOMESH.DAT is created by the user. AUTOMESH.DAT contains the boundary information that AUTOMESH needs to construct the logical coordinates of the boundary. The data file for the MODGUN

problem is shown in Fig. 2.5. Line number one is the title line, lines two through four are the input variables discussed in Sec. 2.4 and lines five through forty-two are the physical coordinates discussed in Sec. 2.5.

AUTOMESH creates the data file LATTICE.DAT which is used by the program LATTICE to construct the mesh. The LATTICE.DAT file for the problem MODGUN is shown in Fig. 2.6 and the logical diagram constructed by AUTOMESH is shown in Fig. 2.7.

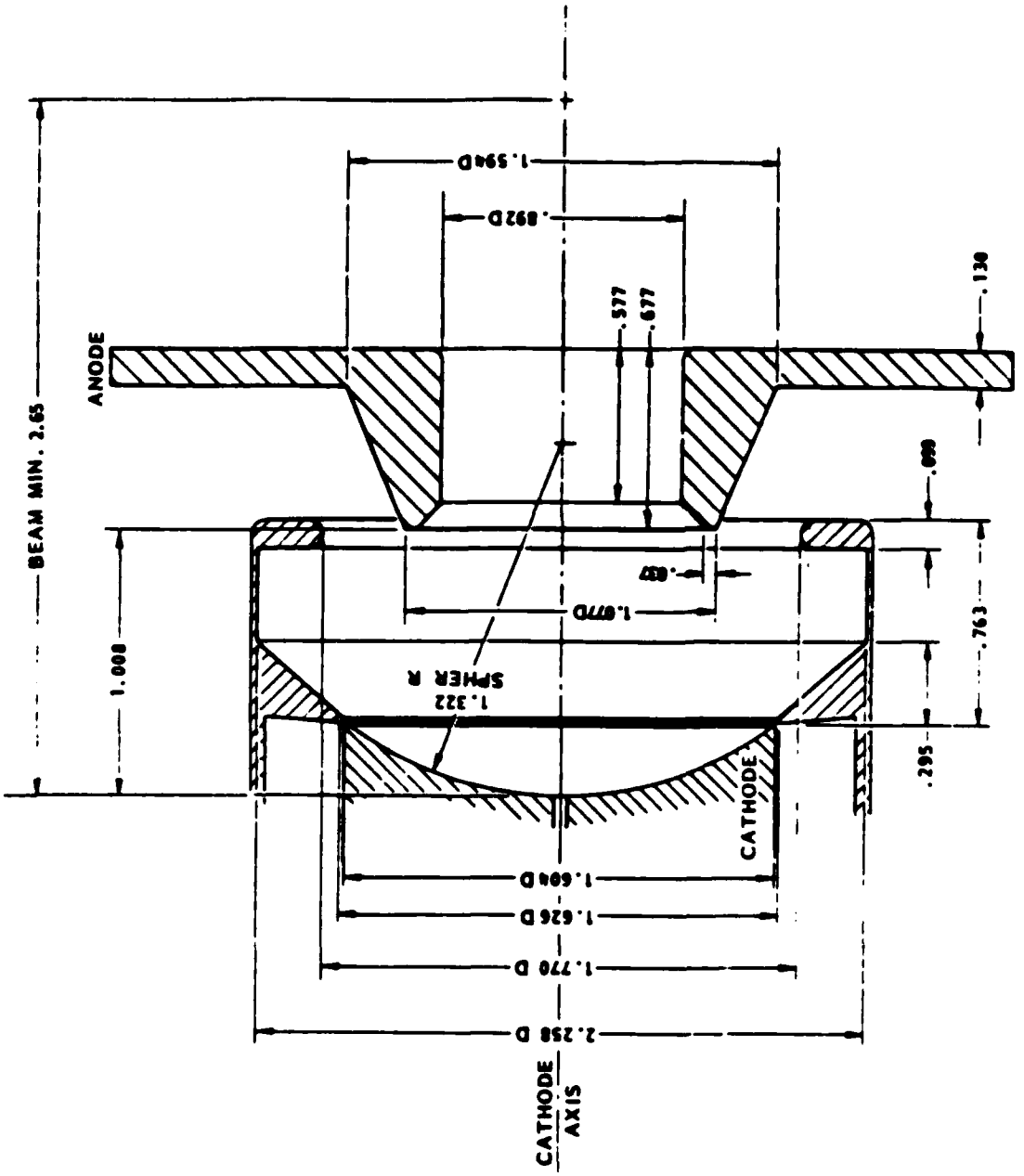


Fig. 2.3 Dimensional Diagram of MODGUN

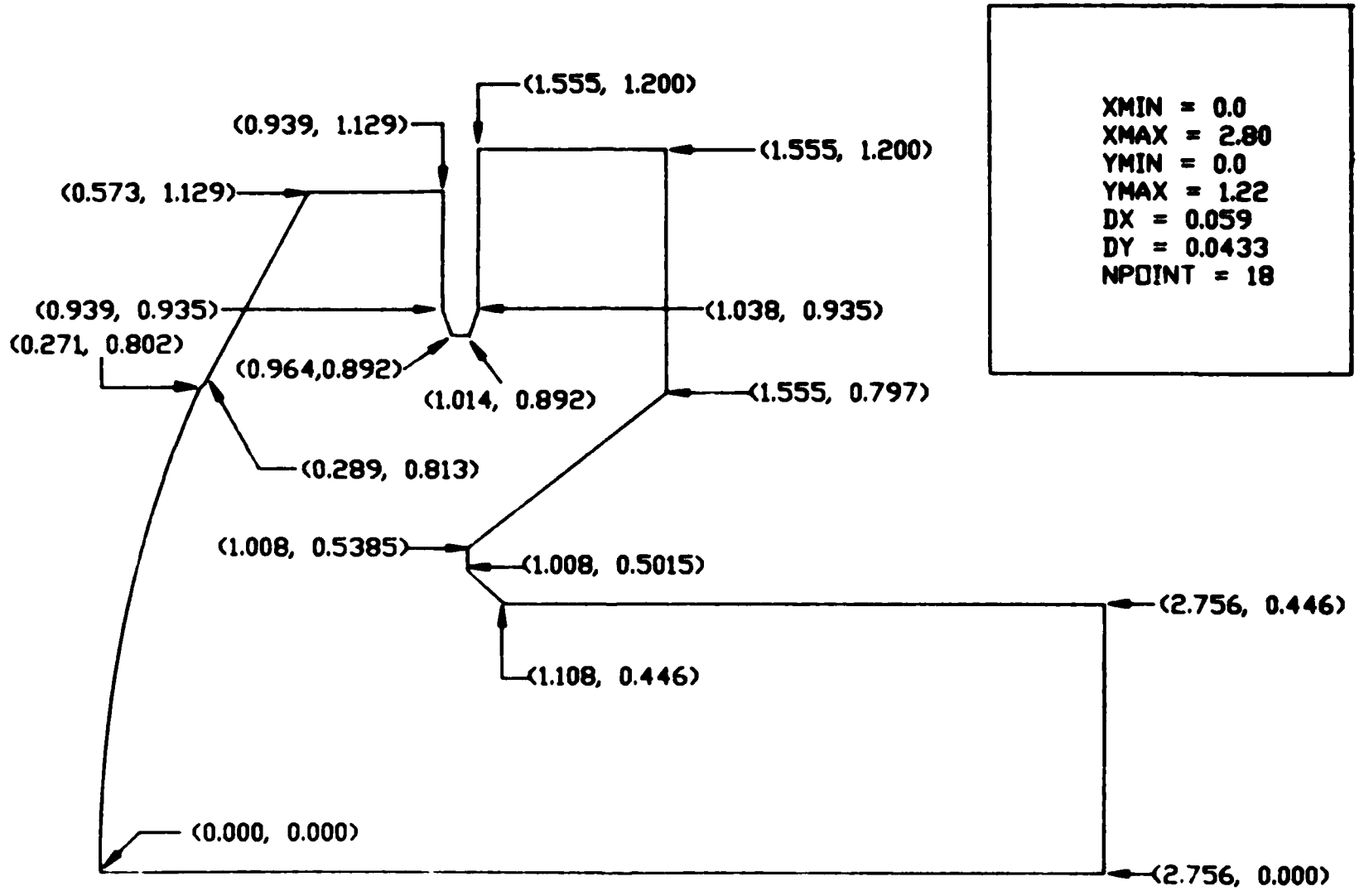


Fig. 2.4 Corner Points of the Physical Boundary of MODGUN

```
 MOD-I GUN
9
'NREG=' 1, 'DX=' 5.9E-2, 'DY=' 4.33E-2, 'XMIN=' 0.0,
'XMAX=' 2.8, 'YMIN=' 0.0, 'YMAX=' 1.22, 'CONV=' 0.0254,
'NPOINT=' 18
3
'X=' 0.0, 'Y=' 0.0, 'ICATH=' 1
2
'X=' 2.7559, 'Y=' 0.0
2
'X=' 2.7559, 'Y=' 0.44600
2
'X=' 1.10800, 'Y=' 0.44600
2
'X=' 1.00800, 'Y=' 0.50150
2
'X=' 1.00800, 'Y=' 0.53850
2
'X=' 1.55500, 'Y=' 0.79700
2
'X=' 1.55500, 'Y=' 1.20000
2
'X=' 1.03790, 'Y=' 1.20000
2
'X=' 1.03790, 'Y=' 0.93500
2
'X=' 1.01400, 'Y=' 0.89170
2
'X=' 0.96400, 'Y=' 0.89170
2
'X=' 0.93890, 'Y=' 0.93500
2
'X=' 0.93890, 'Y=' 1.12900
2
'X=' 0.57290, 'Y=' 1.12900
2
'X=' 0.28890, 'Y=' 0.81300
3
'X=' 0.271058, 'Y=' 0.8020, 'ICATH=' 1
6
'NT=' 2, 'XO=' 1.322, 'YO=' 0.0,'X=' -1.322, 'Y=' 0.0
'ICATH=' 1
```

Fig. 2.5 AUTOMESH.DAT data file for the MODGUN problem.

```
MOD-I GUN
•2  1 •21   0 1 0 0 •9     .0254 SKIP
  I  I       0.0000        0.0000  0  I REGION
  1  1 .000000E+00 .000000E+00    A
 47  1 2.75590     .000000E+00    B
 46  2 2.75590     .445001E-01    B
 47  3 2.75590     .892002E-01    B
 46  4 2.75590     .133800        B
 47  5 2.75590     .178400        B
 46  6 2.75590     .223000        B
 47  7 2.75590     .267601        B
 46  8 2.75590     .312201        B
 47  9 2.75590     .356801        B
 46 10 2.75590     .401401        B
 47 11 2.75590     .446001        B
 19 11 1.18003     .446001        B
 19 12 1.06002     .473755        B
 18 12 1.00001     .501509        B
 18 13 1.00001     .530504        B
 19 14 1.06359     .560046        B
 20 14 1.09817     .581587        B
 20 15 1.14476     .603129        B
 21 15 1.19834     .624671        B
 22 16 1.23592     .646213        B
 23 16 1.28151     .667754        B
 23 17 1.32709     .689296        B
 24 17 1.37267     .710838        B
 25 18 1.41826     .732379        B
 26 18 1.46384     .753921        B
 26 19 1.50942     .775463        B
 27 19 1.56501     .797004        B
 27 29 1.56501     1.20000        B
 18 29 1.03790     1.20000        B
 19 28 1.03790     1.15584        B
 18 27 1.03790     1.11167        B
 19 26 1.03790     1.06750        B
 18 25 1.03790     1.02334        B
 19 24 1.03790     .979172        B
 18 23 1.03790     .935005        B
 18 21 1.01403     .891700        B
 17 21 .964020     .891700        B
 17 22 .938900     .935005        B
 17 27 .938900     1.12900        B
 18 27 .572902     1.12900        B
  7 21 .359900     .892013        B
  6 21 .324409     .852514        B
  6 20 .288910     .813016        B
  5 19 .271059     .802000        K
  5 18 .242754     .763394        K
  4 17 .215822     .723819        K
  4 16 .190341     .683294        K
  3 15 .166344     .641873        K
  3 14 .143863     .599610        K
  3 13 .122926     .556562        K
  3 12 .103562     .512703        K
  2 11 .857949E-01 .468332        K
  2 10 .696492E-01 .423267        K
  2  9 .551457E-01 .377647        K
  2  8 .423033E-01 .331532        K
  1  7 .311391E-01 .284982        K
  2  6 .216675E-01 .238058        K
  1  5 .139012E-01 .190822        K
  1  4 .784993E-02 .143336        K
  1  3 .352192E-02 .956619E-01    K
  1  2 .922910E-03 .478624E-01    K
  1  1 .000000E+00 .000000E+00    K COUN
```

Fig. 2.6 LATTICE.DAT data file produced by AUTOMESH for the problem MODGUN

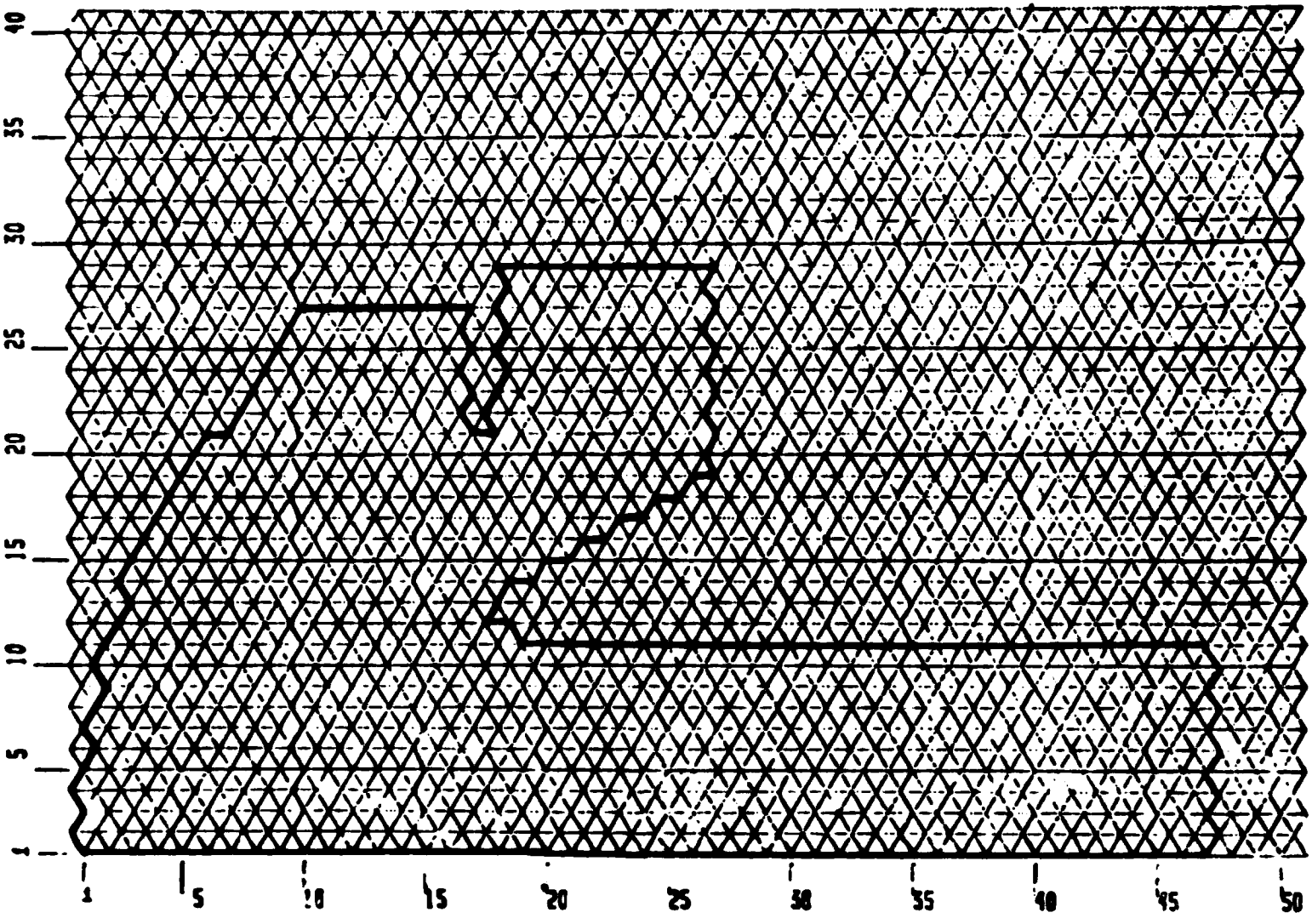


Fig. 2.7 Logical diagram constructed by AUTOMESH.

III. LATTICE (The Mesh Generator)

The program LATTICE generates the non-uniform triangulated mesh from input data describing the boundary. Input data is read by the format-free input routine FREE which is described in Sec. 1.4. The terminology used throughout these notes is as follows: an "air" region means a vacuum region in which the relative permeability equals 1.0, and an "iron" region is a region which contains material for which the relative permeability is a constant but not equal to 1.0, or defined by a linear function or a non-linear table.

3.1 Input Data For LATTICE

There are five groups of input data for LATTICE:

1) The first data line (card) can have anything in columns 2 through 80 and these characters are stored for run identification in the printed output. If column 1 is blank, then this data is for a POISSON problem. If the column is non-blank, then this data is for a SUPERFISH problem. This is shown in Fig. 3.1a.

2) The second data line (or lines) give values for the "problem-constants" (CON's). These are general parameters and options specifying, for example, the number of regions, the boundary conditions, the type of magnetic symmetry, etc. An input value for CON(2) is

a)

```
EXAMPLE LATTICE.DAT FILE
```

b)

```
*2  2 *21  1 1 0 0  *9  1.0000 *138 4  SKIp
```

c)

```
1    1    0.0000    0.0000  0  0  REGION
```

d)

```
 1   1    0.0000    0.0000   B
 10  1    1.0000    0.0000   B
 10  10   1.0000    1.0000   B
 1   10   0.0000    1.0000   B
 1   1    0.0000    0.0000   B  COUNT
```

e)

```
0.1, 0.0,   0.2, 0.25,   0.3,0.5,   0.4, 0.75, 0.5, 1.1
```

B(I) ZB(I)

Fig. 3.1 The five data groups of the LATTICE input data file.
a) Title line.
b) Values of the "problem-constants", CON's.
c) "region-constants", C(I) array.
d) Logical and physical boundary coordinates.
e) Axial magnetic field data (optional).

is only required, the others all having default values. The format of this line is illustrated in Fig. 3.1b and the input list is terminated with the character "S".

3) Data for up to six properties of the region may be specified in the third group of input data. In the program, values are stored in the "C" array and these "region-constants" ("C") specify the following information:

C(1) The region number (integer). The maximum number of regions is 30 and they need not be consecutive.

C(2) The material code for the region (integer).

=0 All points inside this region are omitted from the problem.

=1 air (mu=1)

=2 iron with internal permeability table

=3 iron with input permeability table no. 1

=4 iron with input permeability table no. 2

=5 iron with input permeability table no. 3

C(3) The fixed potential value (volts) on the boundary of the region for electrostatic problems, and for magnetostatic problems it is the total current (amps) in the region.
=0.0 default value (real number)

C(4) The charge density (Coulombs/length**2) of the region for electrostatic problems and the current density (amps/length**2) of the region for magnetostatic problems.

=0.0 default value (real number).

C(5) Triangle type to be generated in the region (integer).

=0 equal weight (default)

=1 equilateral

=2 right

C(6) The region boundary indicator (integer). See section 3.6 for a detailed description of the boundary conditions.

=-1 Fixed potential Dirichlet boundary.

= 0 Magnetic Dirichlet condition on the boundary of this region.

= 1 Neumann condition on the boundary of this region.

The third group of data, "region-constants", is shown in Fig. 3.1c.

4) The fourth group of input data consists of the following

three items:

1. logical coordinates (K,L) of the boundary.
2. physical location (x,y) of the logical coordinates of the boundary.
3. node type

The format of this input data in LATTICE.DAT appears as:

```
K(1) L(1) X(1) Y(1)  B
K(2) L(2) X(2) Y(2)  K
  .    .    .    .   .
  .    .    .    .   .
K(n) L(n) X(n) Y(n)  K Count
```

The input list is terminated with the character "C" to count the number of input values. Comments may follow this "C" since everything else on this line will be ignored. The fourth data group is shown in Fig. 3.1d.

Each logical coordinate must include it's node type. Presently there are three node types which are: boundary, guard and cathode points.

5) The fifth group consists of axial magnetic field data (optional). LATTICE accepts values of B_z measured experimentally along the z-axis from which off axis values of B_z and B_r are calculated.

If the variable NBPTS (CON(138)) is greater than zero, then

the following FORTRAN format free READ statement is executed:

```
READ(91,*)(B(I),ZB(I),I=1,NBPTS)
```

where:

B(I) - value of the magnetic field on the z-axis (in Tesla)

ZB(I) - z-axis location of the magnetic field B(I).

The fifth group of data, the axial magnetic fields, is shown in Fig. 3.1e. B(I) and ZB(I) is located on the data line immediately following the last region and there should be exactly NBPTS pieces of data for both B(I) and ZB(I). Also note that the magnetic field data should extend beyond the boundary by a small amount.

3.1.1 Multiple Regions

Data groups (3) and (4) are repeated for each region. The first region defines the entire problem mesh and must contain the maximum (K,L) values of the mesh. Data for the succeeding regions redefines, or overwrites the region constants for all triangles belonging to this new region. Data for each new region overwrites previously defined values in the same way. This feature is known as "successive region data overwriting". For example, suppose that one wishes to specify the cathode at one voltage and the focus electrode at another. Two additional regions must now be added that specify the cathode at V_1 and the focus electrode at V_2.

Usually each region is closed, i.e., the data for the first and last boundary points are identical. It is also possible to specify data for a "point" region or a "line" region. The purpose of this would be to define the physical coordinates of specific points such as fixed potential values along a portion of a boundary. For point and line regions, the input values for C(2) and C(5) in the "region-constant" data are arbitrary, since this information is used only for closed regions.

3.2 Specifying Logical Coordinates In Logical Space

For two points in logic space, it is necessary that they be on the same "logical lines" in logic space. These four lines shown in Fig. 3.2 are:

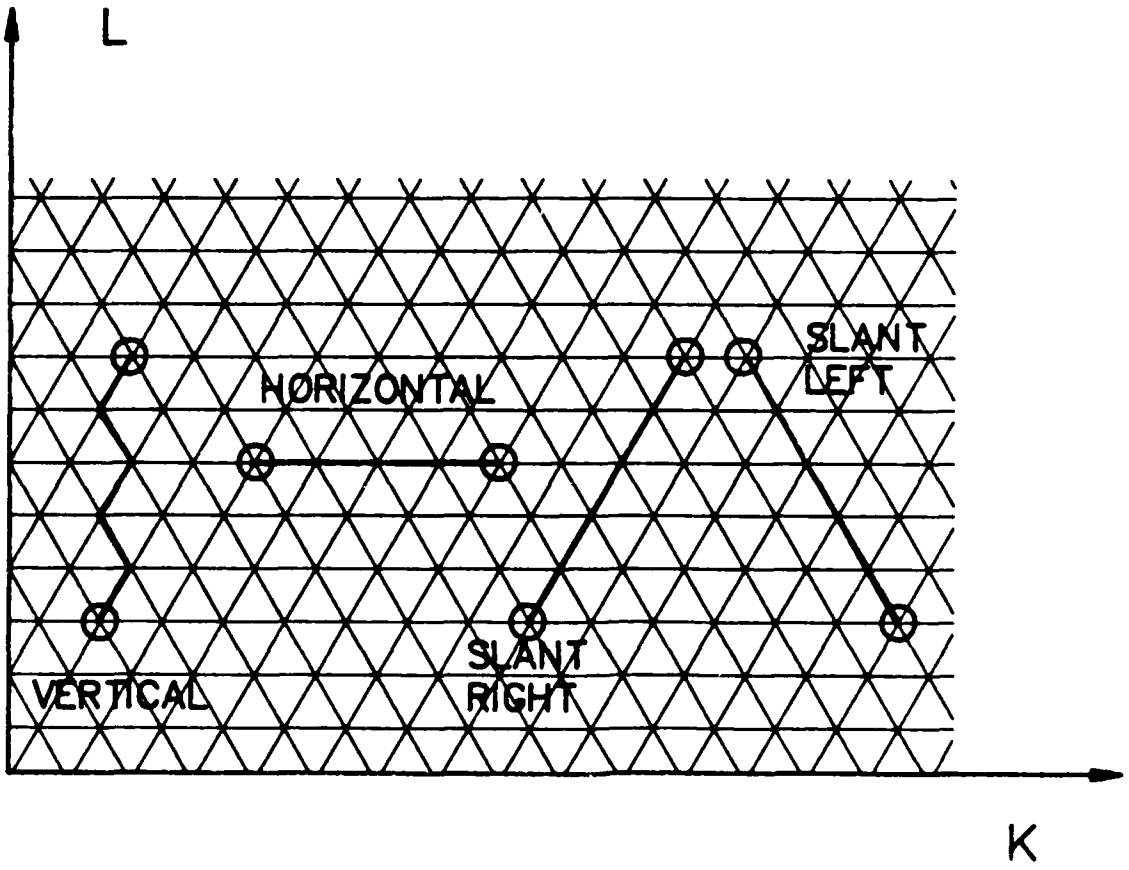


Fig. 3.2 Primary "Logic Lines" on a Equilateral Triangle Logic Mesh

1. vertical
2. horizontal
3. slant right
4. slant left

It is only necessary to specify the end points of the logical line, intermediate points will then be linearly interpolated for (x,y).

3.3 Mesh Generation Resulting In Negative Areas

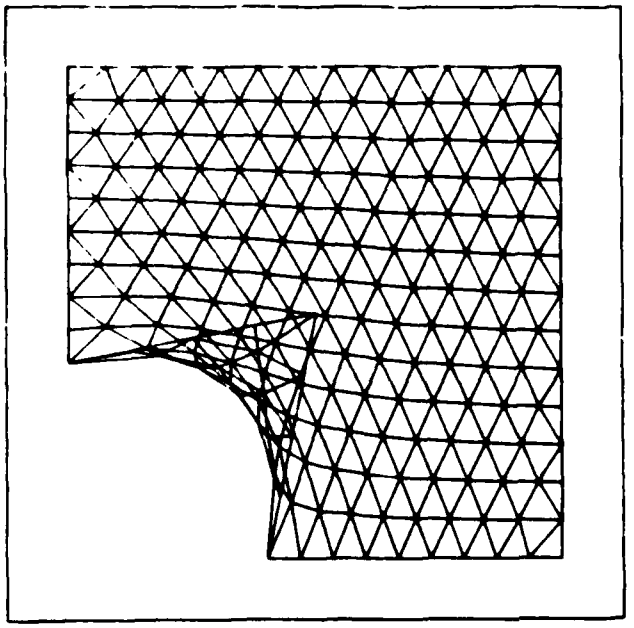
In the generation of the mesh, it is possible to have a negative or zero area triangle. LATTICE computes the area of each triangle in the mesh, and if any of the areas are negative, a flag will be set and the program POISSON will not be able to be executed.

Fig. 3.3a displays a mesh that has some negative area triangles. It appears that the lower left hand corner has folded over onto the mesh. LATTICE will warn the user of negative area triangles by printing the message:

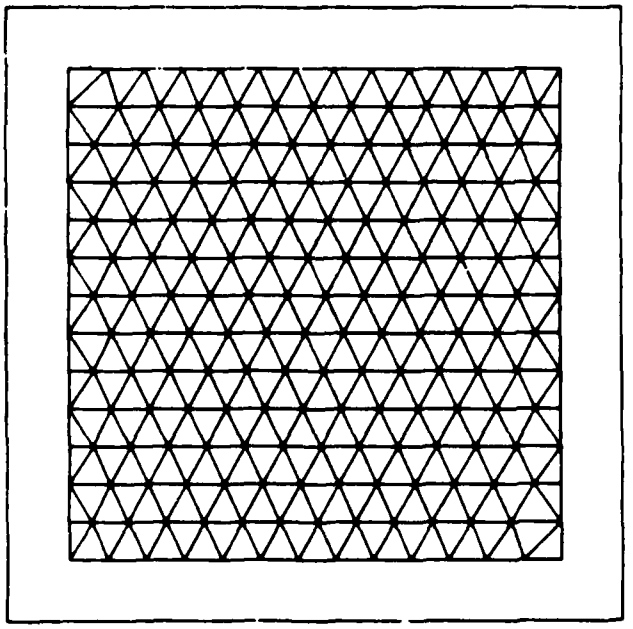
"The triangle of point (K,L) has a negative or zero area"

where the point (K,L) is the logical coordinate of the triangle in logic space.

Usually, the two main probable causes of negative areas are:



(a) A mesh with negative area triangles.



(b) Corrected mesh

Fig. 3.3 (a) An example of a square mesh that has some negative area triangles. (b) Corrected square mesh by specifying the lower left hand corner at (0.0,0.0)

1. physical coordinate (x,y) data error
2. too many mesh points defined in a certain area of the mesh.

For the example of Fig. 3.3a, the cause of the error was case #1. Instead of the lower left corner being specified at (0.0, 0.0), it was specified in the center of the mesh, which resulted in the mesh folding over upon itself. An example of case #2 will be illustrated in the example of MODGUN in Sec. 3.8.

The following procedure can be used to correct the problem of negative areas:

1. Determine the approximate physical location (x,y) of the negative area triangle. This can be determined from the knowledge of the (K,L) coordinate and the LATTICE.DAT data file.

2. Using the plotting program, zoom in on the troubled area of the mesh.

3. After examining the problem, determine what changes must be made to LATTICE.DAT to correct the problem.

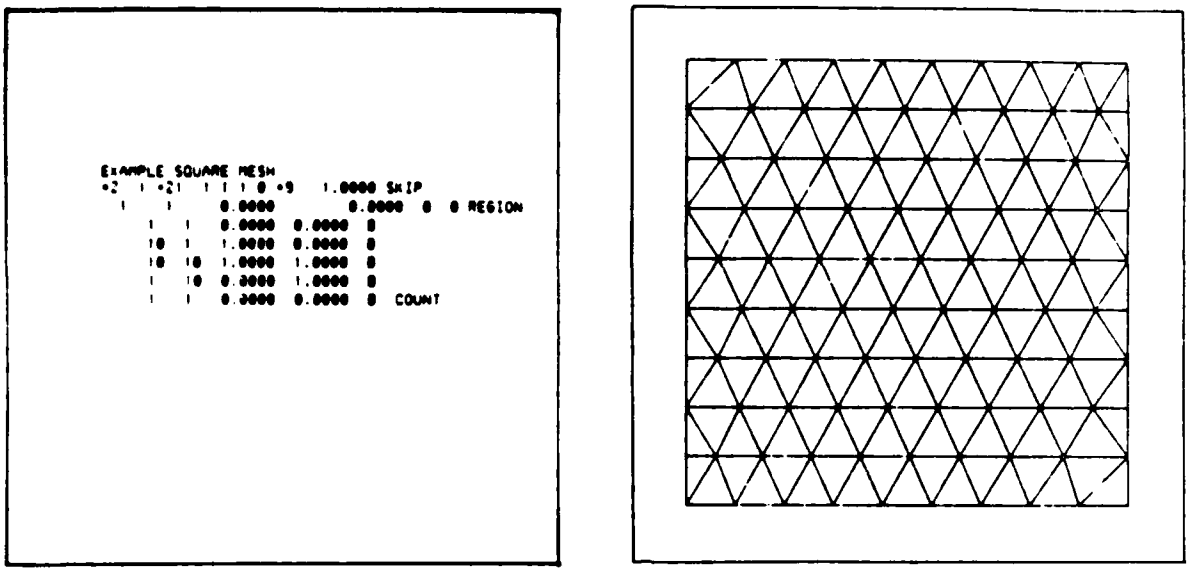
As in the example of Fig. 3.3a, moving the lower left hand corner from the center of the mesh to (0.0, 0.0) corrected the problem. This is shown in Fig. 3.3b.

3.4 Guard Nodes

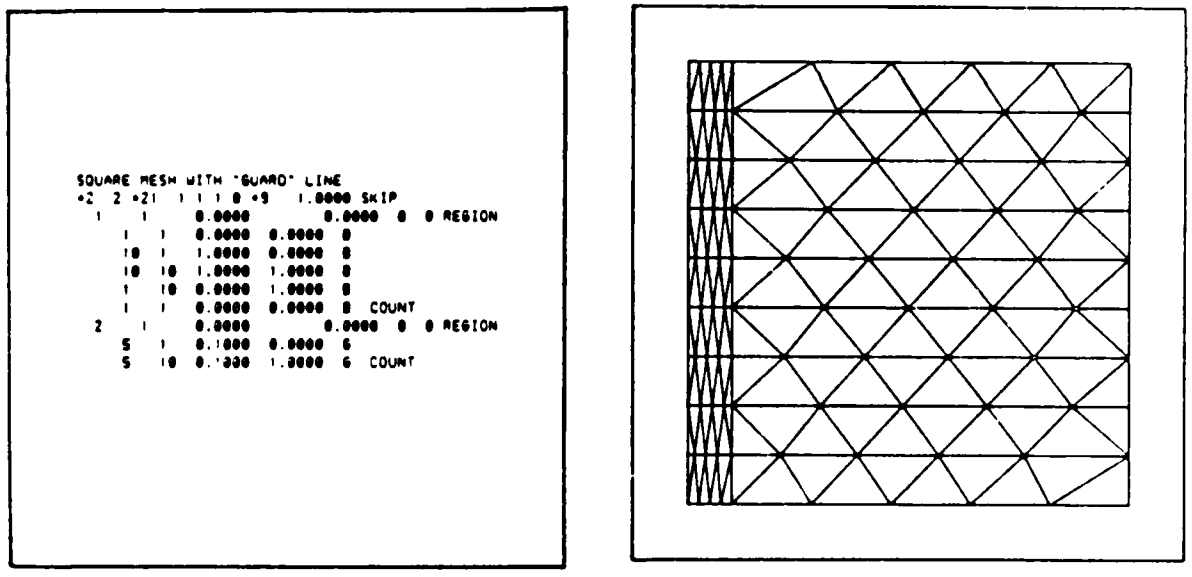
The physical location (x,y) of each mesh point (or node) within the boundary of a problem is determined numerically by a relaxational process. In some instances, the user may want to fix certain mesh points at a predetermined location (x,y) in the mesh. The "guard" node is used to fix the (x,y) coordinate of a node.

The "guard" mesh is a separate region in the LATTICE.DAT file. The "guard" region should be defined after the entire problem mesh of the first region has been defined. For "guard" regions, the input values for C(2)-C(5) in the "region-constant" data are arbitrary and C(6) can either be set to C(6)=0 or 1, but not -1.

An example illustrating the use of "guard" nodes is shown in Fig. 3.4. The "guard" mesh is set up on the X=0.1 line. As mentioned in Sec. 3.2, it is only necessary to specify the end points of the logical line, intermediate points are linearly interpolated for (x,y).



a) Square mesh without "guard" line.



b) Square mesh with "guard" line.

Fig. 3.4 Example of a square mesh with a "guard" line at x = 0.1. a) LATTICE.DAT data file and relaxed mesh without the "guard" line. b) LATTICE.DAT data file and relaxed mesh with the "guard" line.

3.5 Cathode Nodes

The emitting surface consists of a set of elementary planar diodes. Near this surface an approximately rectangular mesh is generated as shown in Fig. 3.5. The spacing "z" of the rectangular mesh is determined by

$$z = (a+b) / (FRT*NAC)$$

where z, a and b are defined in Fig. 3.5. The program parameter FRT (CON(137)) controls the nearness of the nodes, and the program parameter NAC (CON(126)) is the number of nodes in front of the emitting surface.

It has been determined empirically that for curved emitting surfaces being approximated by a set of planar diodes, two criteria should simultaneously be :

1. "The change in the angle of the line of nodes in the rectangular mesh from cathode point to cathode point should be less than 4 degrees".

2. The program parameter FRT which controls the nearness of the nodes should be greater than 8.

The user must be careful that these two conditions are satisfied since the potentials deviate substantially from planar diode theory when they are not close to the emitting surface.

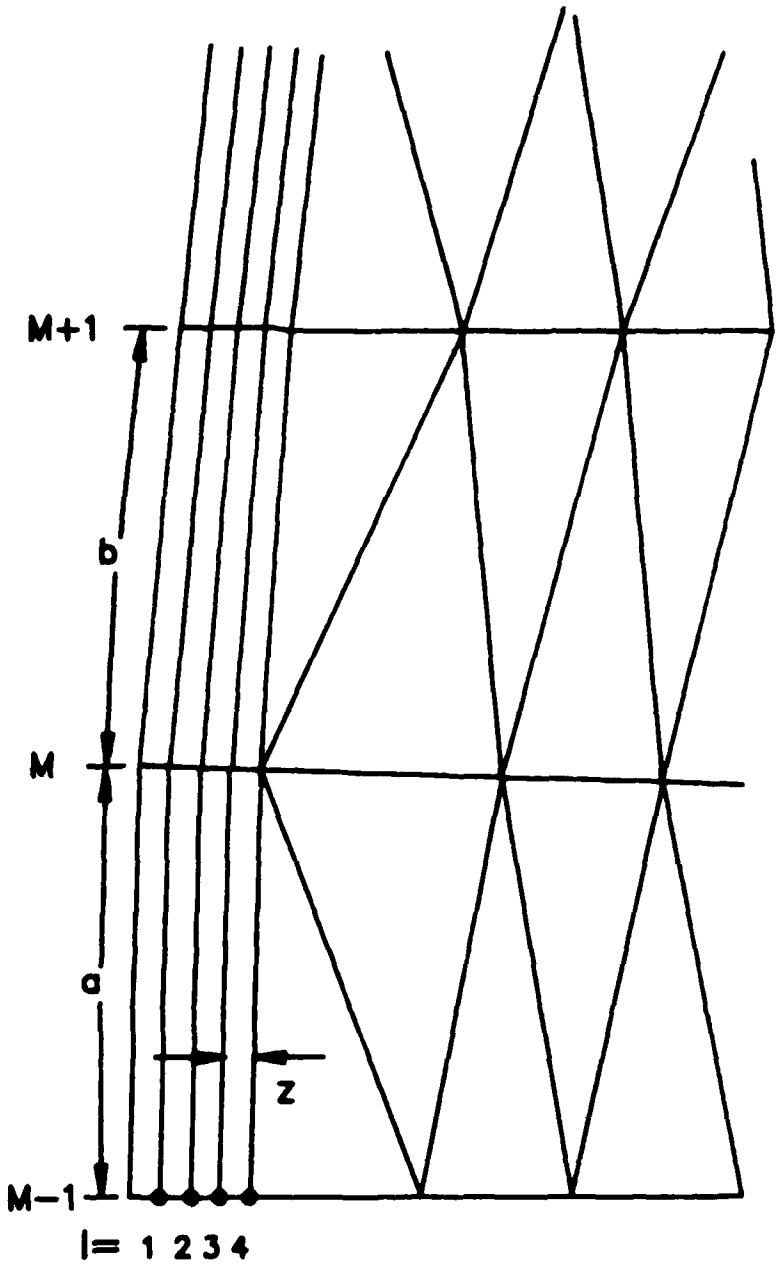


Fig. 3.5 Approximate rectangular mesh near the cathode (NAC=4).

The nodes "M" of Fig. 3.5 are the emitting points of the cathode. They are designated as "K" nodes in the LATTICE.DAT data file.

3.5.1 Horizontal and Vertical "Logic-Line" Cathodes

The program can generate two basic types of cathode mesh in "logic" space:

1. vertical
2. horizontal

The "horizontal" cathode must be specified on the same horizontal logic line, i.e., all cathode nodes must be on the same "L" line in logic space. The NAC nodes associated with the cathode nodes are assigned from bottom to top on the same "K" line as shown in Fig. 3.6a. In cylindrical coordinates, a horizontal cathode on the "z" axis is not allowed since this situation is not physically possible.

The rules for the "vertical" cathode are less stringent than those of the "horizontal" cathode. The "vertical" cathode may be constructed from a vertical, slant right, slant left or a combination of these primary "logic lines" (see Fig. 3.2). The NAC nodes associated with the cathode nodes are assigned from left to right on the same "L" line as shown in Figs. 3.6b, c and d.

When constructing a "vertical logic-line" cathode, it is important that each emitting node, "K", has NAC nodes free to the

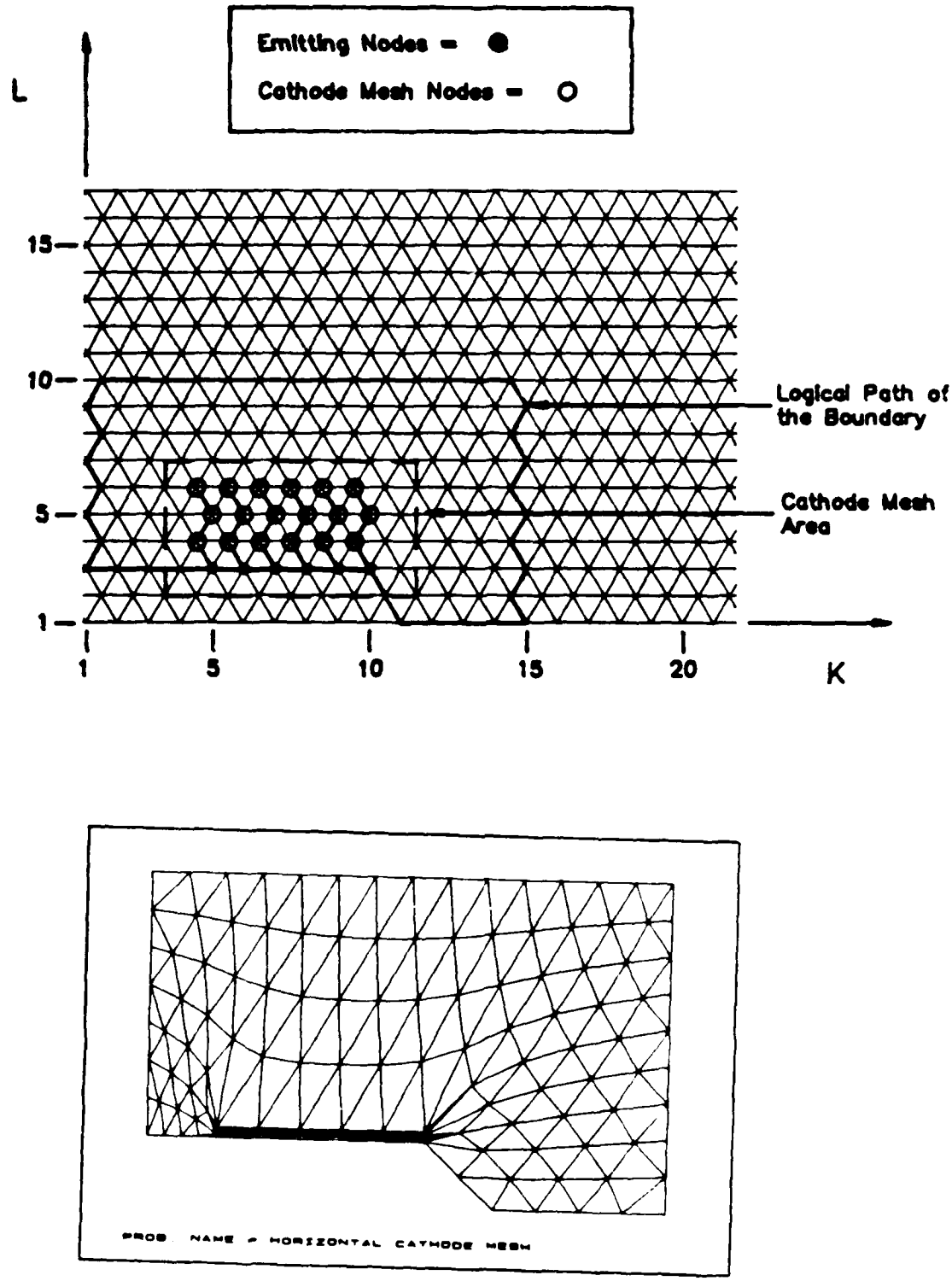


Fig. 3.6 a) Logical space diagram and relaxed triangular mesh of the horizontal cathode.

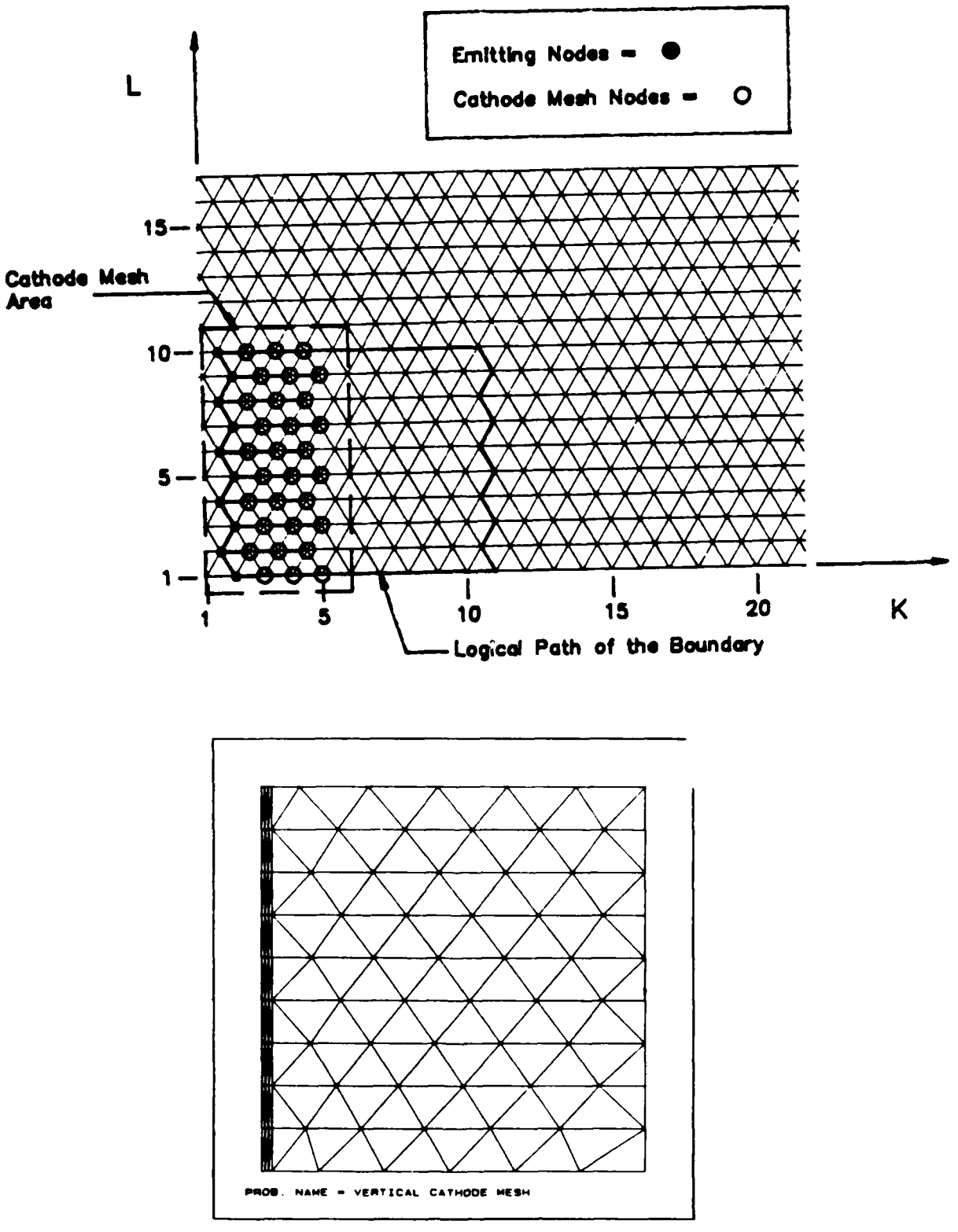
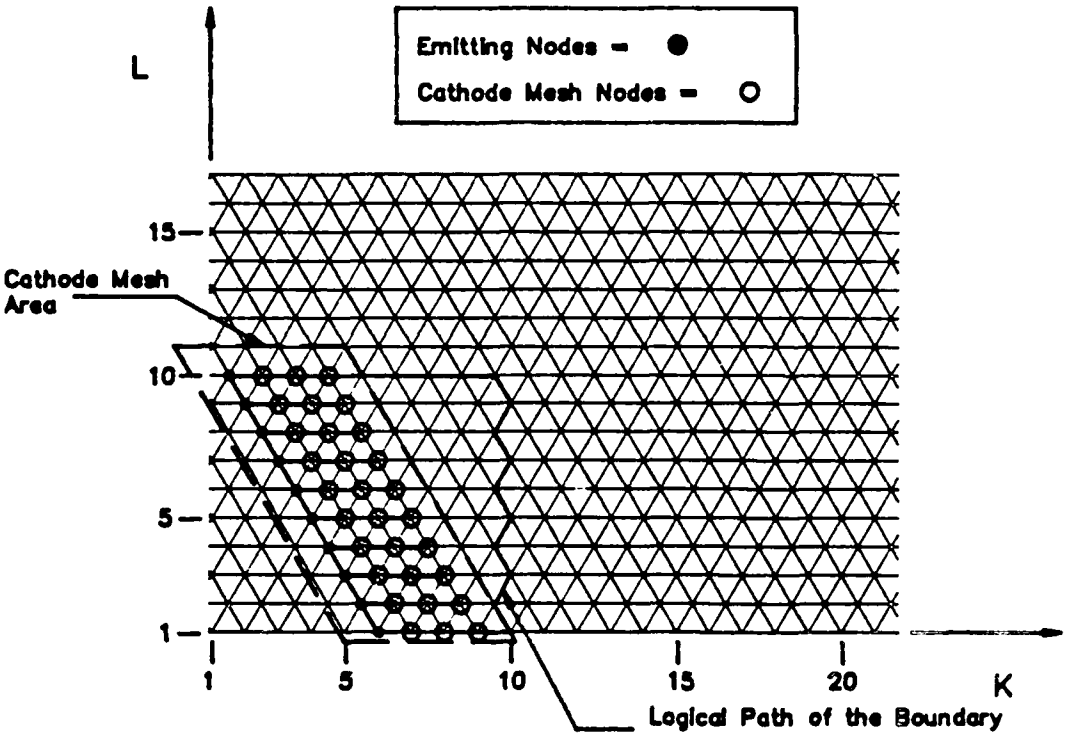


Fig. 3.6 b) Logical space diagram and relaxed triangular mesh of the vertical cathode.



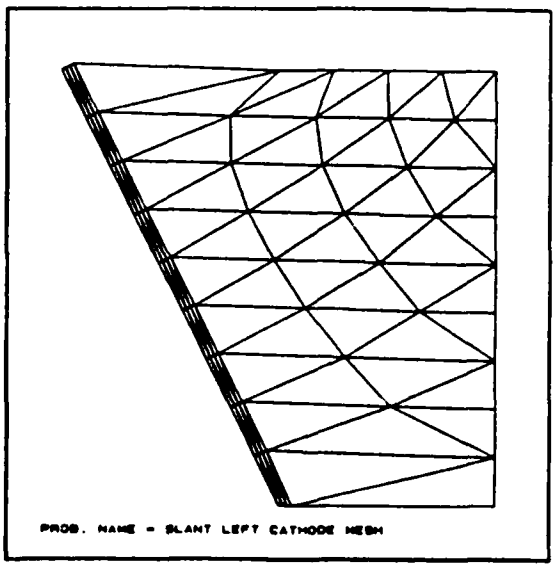
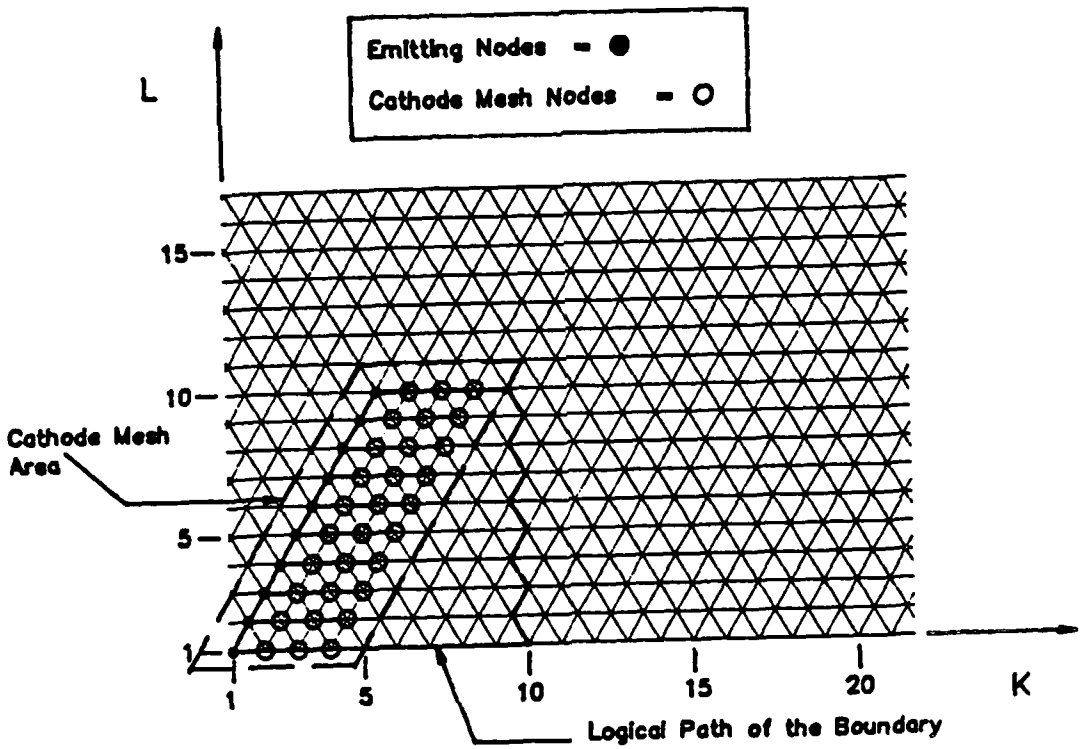


Fig. 3.6 c) Logical space diagram and relaxed triangular mesh of the vertical slant left cathode.



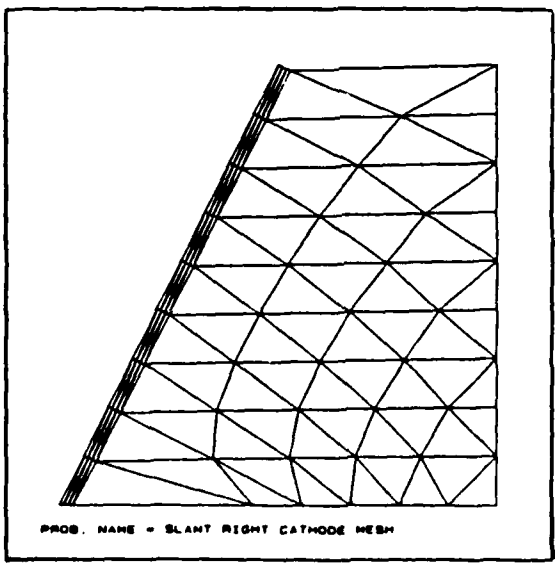
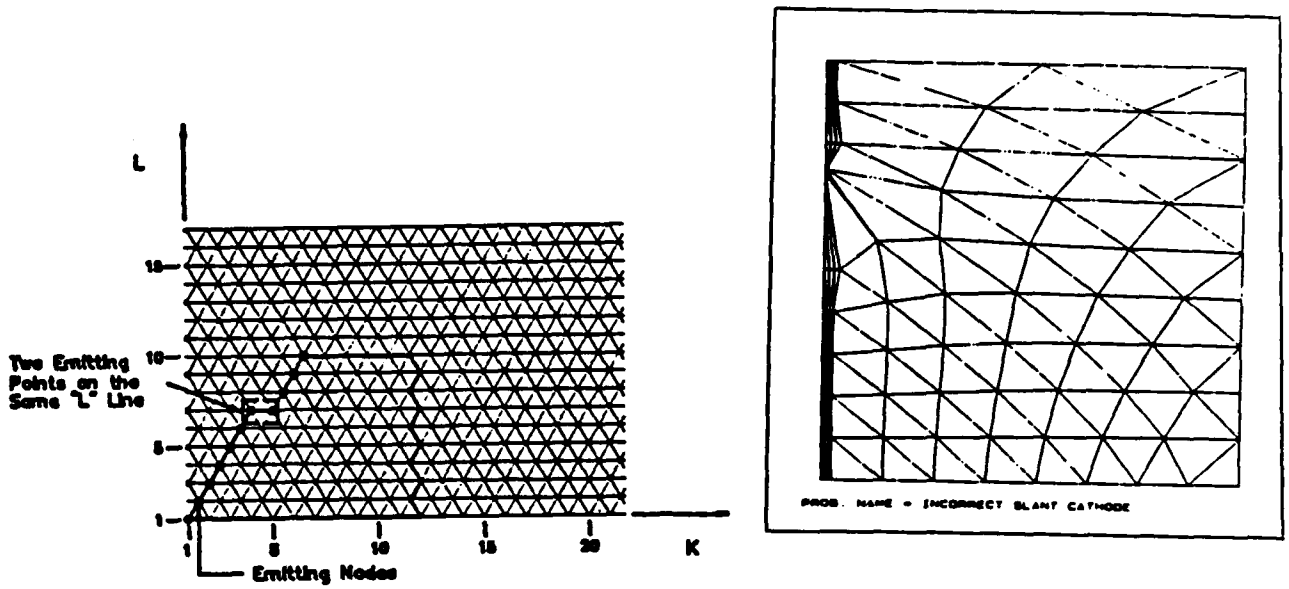
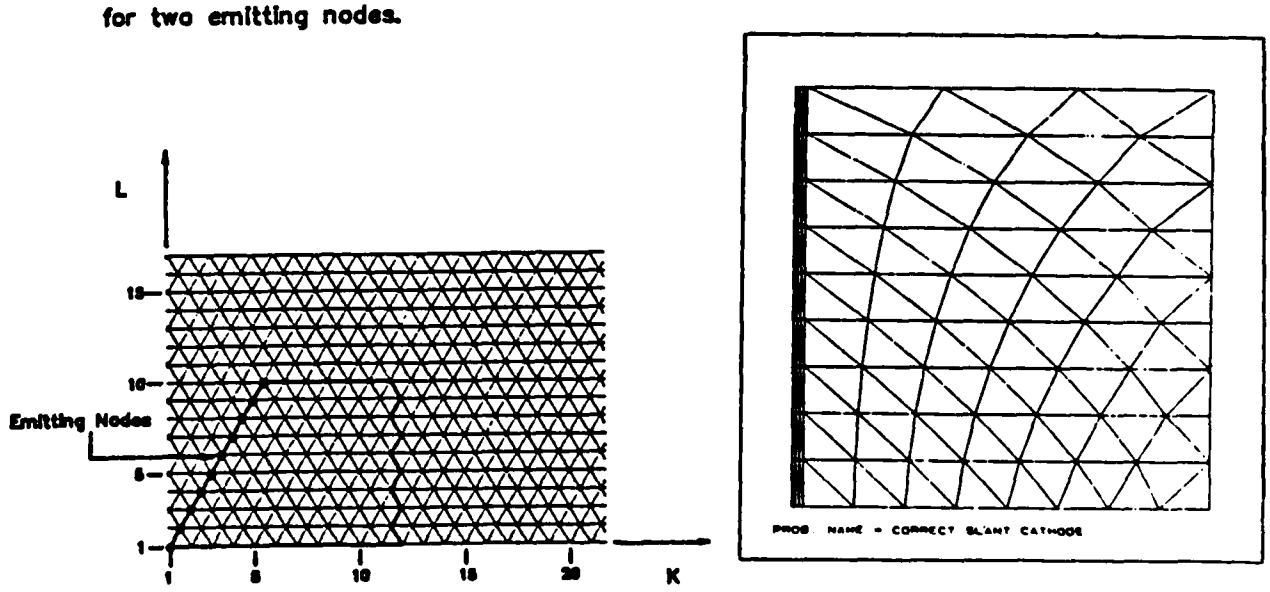


Fig. 3.6 d) Logical space diagram and relaxed triangular mesh of the vertical slant right cahtode.

right of it, i.e., no two emitting nodes may be on the same "L" logic line (or, no two emitting nodes may be on the same "K" line for a "horizontal logic-line" cathode). The program LATTICE does not check for this error, and as a result, negative area mesh will be generated. An example of a "slanted logic-line" cathode that shares the same logic "L" value for two emitting points is shown in Fig. 3.7a, and the corrected mesh is shown in Fig. 3.7b.



(a) Slant–right logic–line cathode that shares the same "L" value for two emitting nodes.



(b) Corrected cathode mesh.

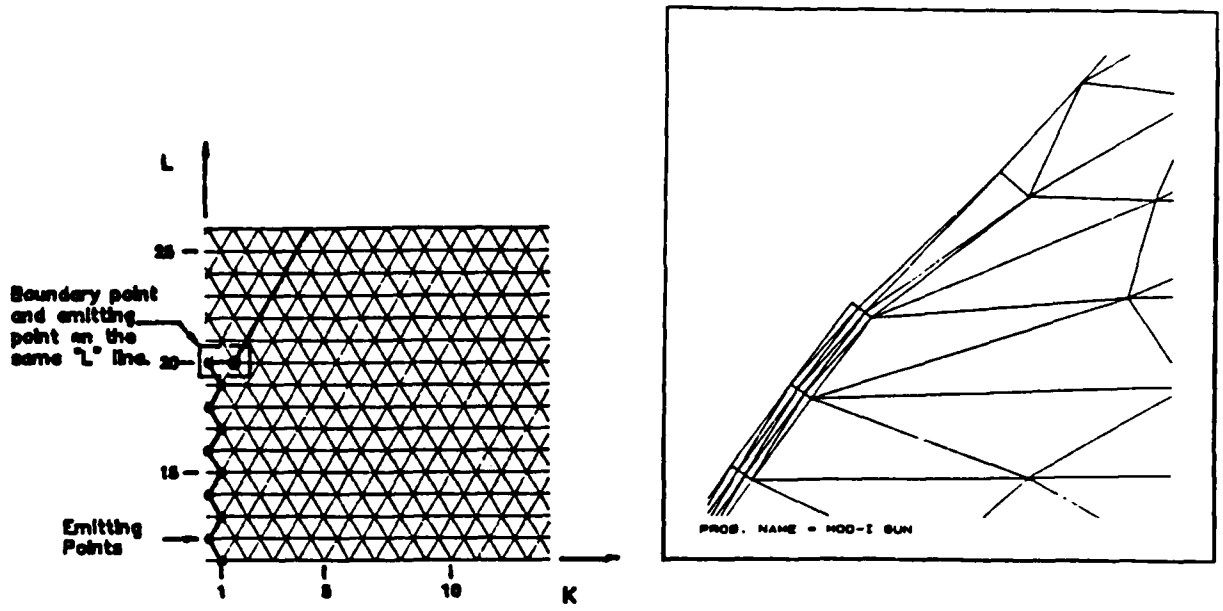
Fig 3.7 Example of an incorrect cathode mesh.
(a) Slant–right cathode sharing the same "L" value for two points.
(b) Corrected cathode such that each emitting node has NAC nodes free to the right of it.

3.5.2 Modifying AUTOMESH Results

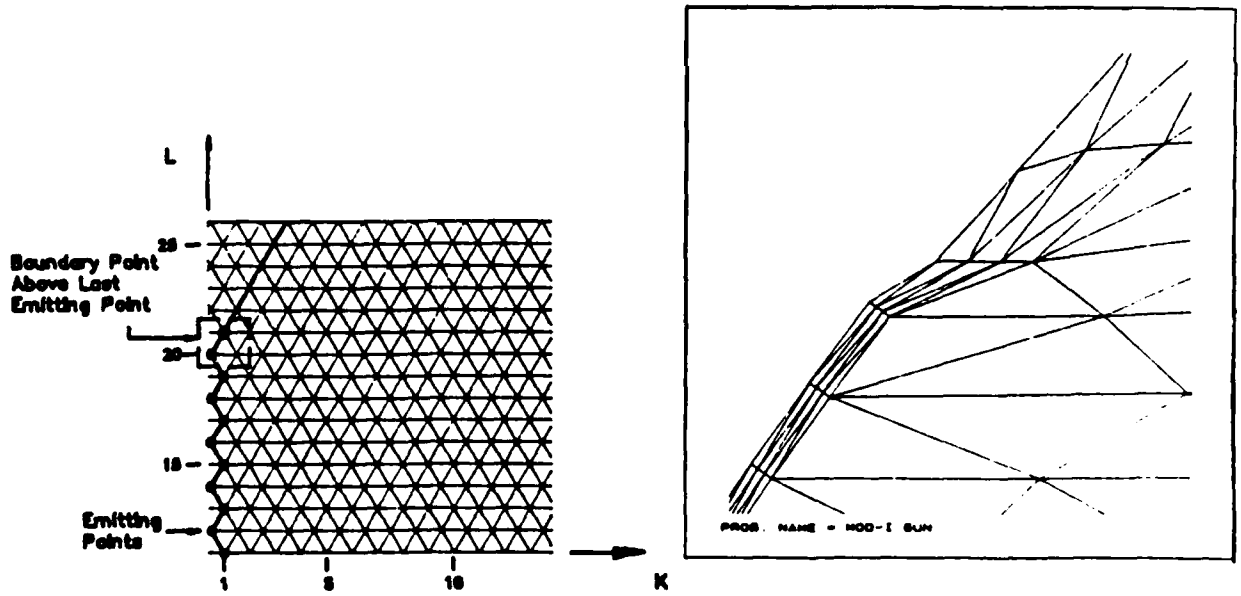
The user must be cautioned that AUTOMESH will not construct a cathode "logic-line" according to the rules given in Sec. 3.5.1. As discussed in Chapter 2, AUTOMESH finds the path in the logical mesh which best fits the physical boundary. This logical path found by AUTOMESH will not necessarily cohere to the rules of Sec. 3.5.1.

Probably the best way to construct a consistent cathode mesh from the output of AUTOMESH is to put all the cathode "K" nodes on one primary logic line. As for the example MODGUN of Sec. 2.7, the cathode could be changed from the "zig-zag" vertical logical line, to the K=1 line. This is shown in Fig. 3.11 of Sec. 3.8.

Another area of concern on the cathode is the last emitting point. Many times AUTOMESH will put a boundary point on the same logical line "L" as the last emitting point. As a result, some boundary points are lost when the cathode mesh is generated by LATTICE because the last emitting point did not have NAC free nodes to the right of it. An example of this type of problem is shown in Fig. 3.8a. This problem was corrected by changing the boundary point from logical coordinate (20,2) to (21,1). The results of the correction is shown in Fig. 3.8b (Note: Fig. 3.8b includes a "guard" mesh).



a) Boundary point on the same logical "L" line as the last emitting point.



b) Corrected mesh by moving the boundary point above the last emitting node.

Fig. 3.8 Modifying incorrect boundary point generation of AUTOMESH

Sec. 3.6 Boundary Conditions

There are two ways that boundary conditions are specified in LATTICE. These two methods, specifying CON(21)-CON(24) and C(6) of the "region-constants", are described below.

Sec. 3.6.1 Specifying CON(21)-CON(24)

The four CON's, CON(21), CON(22), CON(23), and CON(24), determine the boundary conditions on the extreme rectangular logical boundaries of the problem regardless of the value of C(6) for a region part of whose boundary coincides with an extreme rectangular logical boundary. The extreme logical boundaries are: K=1, and K = KMAX, and L = 1 and L = LMAX. (Note: if C(6) = -1, then this is not the case). See Sec. 3.7.1b for the description of these CON's.

Sec. 3.6.2 "Region-Constant" C(6)

C(6), of the "region-constant" C array described in Sec. 3.1, is used for two purposes:

1) to indicate the boundary condition on a boundary of the problem which does not coincide with the extreme logical boundaries (K=1, L=1, K=KMAX, L=LMAX) of the problem.

2) to indicate special fixed potential points.

1) When used for purpose 1, C(6) acts as either a Neumann or Dirichlet boundary indicator:

C(6)=0 Dirichlet condition on the boundary of the particular region (magnetic Dirichlet condition).

C(6)=1 Neumann condition on the boundary of the particular region.

The default values for C(6), which are used unless changed by input data are:

C(6)=1 for all regions.

2) When C(6) is used for purpose (2), it is set to -1 and the fixed potential value is C(3) of the "region-constant" data.

When using feature (2), the region should be either a line or point region that overwrites a portion of the boundary. C(6) should never be set to -1 for the first region, which describes the whole boundary.

When defining the boundary of a problem (first region), C(6) should be set to one, and all Dirichlet boundaries should be added in succeding regions.

Sec. 3.8 gives an example of using both boundary indicators and the use of line regions to define fixed potentials along the

boundary.

Sec. 3.7 "Problem-Constant" Array "CON(I)"

The "CON" array, is an array containing integers and real numbers which are general parameters and options specifying the control of the program. These values are specified or changed from their default values in the second group of input data for LATTICE (see Sec. 3.1).

Sec. 3.7.1 describes the CON(I)'s which controls the mesh generation of LATTICE. Sec. 3.7.2 describes the CON(I)'s that are associated with the electron-flow (gun) problem.

Sec. 3.7.1 Mesh Generation

This section describes the "problem-constants" that can only be specified in the LATTICE program. These CON's are general parameters and options specifying the control of the mesh generation in LATTICE. Default values are given in parentheses.

3.7.1a Parameters defining the basic problem:

CON(2)=NREG	Number of regions in the LATTICE data file.
CON(9)=CONV (1.0)	Conversion factor for the units in the LATTICE data file. CONV = 1.0 for meters CONV = 0.0254 for inches

3.7.1b Boundary Conditions

CON(21)=NBSUP (0) CON(22)=NBSLO (0) CON(23)=NBSRT (0) CON(24)=NBSLF (0)	Indicators for the Neumann boundary conditions on the UPper, LOwer, RighT, and LeFt logical boundaries of the problem region, e.g., NBSUP=1, Neumann boundary on the upper logical problem boundary, NBSUP=0, Dirichlet boundary on the upper logical problem boundary.

3.7.1c Printed Output Options:

CON(32)=IPRINT (0)	IPRINT=-1, print (x,y) coordinates of mesh points. IPRINT=0, nothing

3.7.1d Current Parameter and Options:

CON(70)=ICAL	ICAL=1, use angle formula for calculating the current associated with a point (used when accurate fields near coil boundaries are needed). ICAL=0, use normal area formula.

3.7.1e Over Relaxation Factors

CON(79)=RHOXY (1.6)	The starting over-relaxation factor for the generation of the mesh.
CON(81)=NOTE (1)	NOTE=0, point relaxation order is air points, interface points, then iron points (this point order must be used for

PANDIRA).

NOTE=1, point relaxation order is air and interface points, then iron points.

3.7.1f Convergence Criteria

CON(84)=EPSO
(1.0E-5)

The convergence criterion for the mesh generation.

3.7.1g Conformal Transformation Factors

CON(37)=MAP
(1)

W=Z**MAP/(MAP*RZERO**(MAP-1)) is the conformal transformation used, if any. Program LATTICE automatically transforms the current density. Program POISSON transforms the permeability and calculates fields in both the original and the transformed geometrics.

CON(123)=TNEGC
CON(124)=TPOSC

The total negative and positive current in the transformed geometry (this equals the total negative and positive current in the original geometry).

CON(125)=RZERO
(1.0)

The scaling factor if the transformation
W=Z**MAP/MAP*RZERO**(MAP-1))
is used. Normally RZERO=ABS(ZO) which equals the aperture radius.

Sec. 3.7.2 Space-Charge Flow Parameters

This section describes CON's (125)-(150) which are associated with the electron-gun problem.

CON(126)=NAC (3)	Number of nodes in the rectangular mesh in front of the cathode.
CON(127)=SSSF (0.2)	Space-charge suppression factor (used when CBAR<=0)
CON(128)=CGAMMA (0.2)	CBAR modification parameter, or cathode loading convergence parameter.
CON(129)=FJLIM (1E10)	Maximum current density from temperature limited operation considerations (amps/m^2)
CON(130)=CHILDK (2.334E-6)	Child's Law planar diode constant
CON(131)=VOLTCA (0.0)	Cathode to Anode voltage (this is automatically determined in POISSON).
CON(132)=IGUN	Control flag indicating electron-gun problem. IGUN=0, not an electron-gun problem. IGUN=1, electron gun problem (LATTICE will automatically determine this by the presence of cathode "K" nodes).

CON(133)=ETA (1.759E11)	Electron charge to mass ratio.
CON(135)=MAXDT (400)	Maximum number of time steps allowed in the tracing of each ray.
CON(136)=NODES	Number of emitting nodes of the cathode (determined in POISSON).
CON(137)=FRT (8.0)	Parameter which controls the nearness of the rectangular mesh in front of the cathode.
CON(138)=NBPTS (0)	Number of magnetic field points to be inputed (for NBPTS=0, no magnetic field data is read in).
CON(139)=FRDT (0.12)	A multiplication constant in which the time step may be fractionally adjusted.
CON(140)=MAXCYC (20)	Maximum number of major current cycles to be allowed to run.
CON(141)=ECBAR (5.0E-3)	CBAR convergence tolerance
CON(142)=PGUESS (0)	Initial guess of perveance (amps/volts$^{3/2}$).
CON(143)=MPRINT (1)	Printing options for the results of POISSON.(see 4.4.2)
CON(144)=DCBAR	The change in CBAR between major current cycles. This

	quantity is calculated in POISSON, and is compared with ECBAR to determine convergence.
CON(145)=CURREN (0)	The amount of current emitted from a temperature limited cathode.
CON(146)=INCPLT (1)	Time step increment for plotting and printing trajectories.
CON(147)=ISP (8)	Spiral scan limit of the block type search in the nearest node routine.

Sec. 3.8 Sample LATTICE Data File For The Problem MODGUN.

In Sec. 2.7, AUTOMESH created the data file LATTICE.DAT for the problem MODGUN as shown in Fig. 2.6. A minimum of three modifications have to be made to LATTICE.DAT before the mesh can be generated by LATTICE:

1) Set CON(2) and CON(21)-CON(24) to their correct values. CON(2), which is the number of regions, is set to three. The three regions are: problem region, cathode and focus electrode region, and the anode region. CON(21), CON(22) and CON(23) are set to one since the axis, the extreme top logical L line (LMAX), and the extreme right logical K line (KMAX) are Neumann boundaries.

2) Change the "zig-zag" vertical logic line to a constant K=1 logic line. The "zig-zag" logic line would have worked, but for the sake of an example in constructing a consistent cathode mesh, this modification is illustrated.

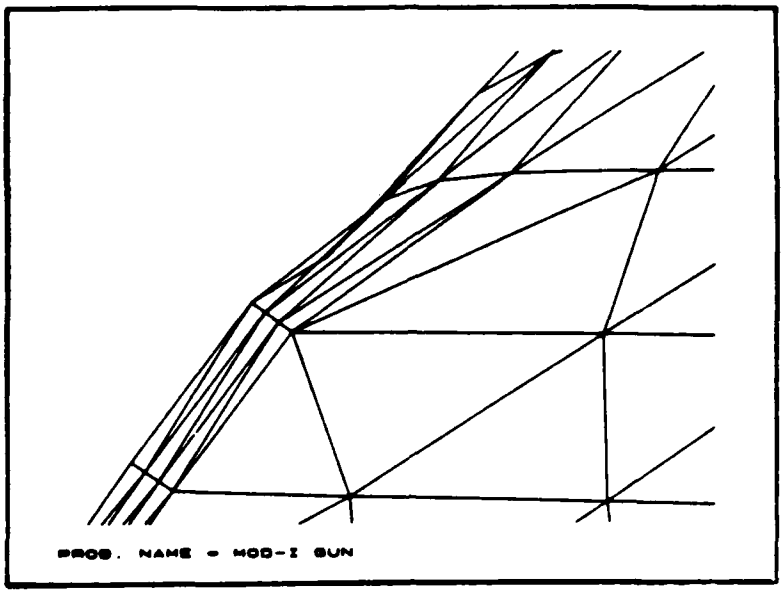
3) Add two additional regions:
   1) cathode and focus electrode at 0 volts
   2) anode at 8000 volts

   Note that IBOUND=C(6) is set to -1 for these two regions.

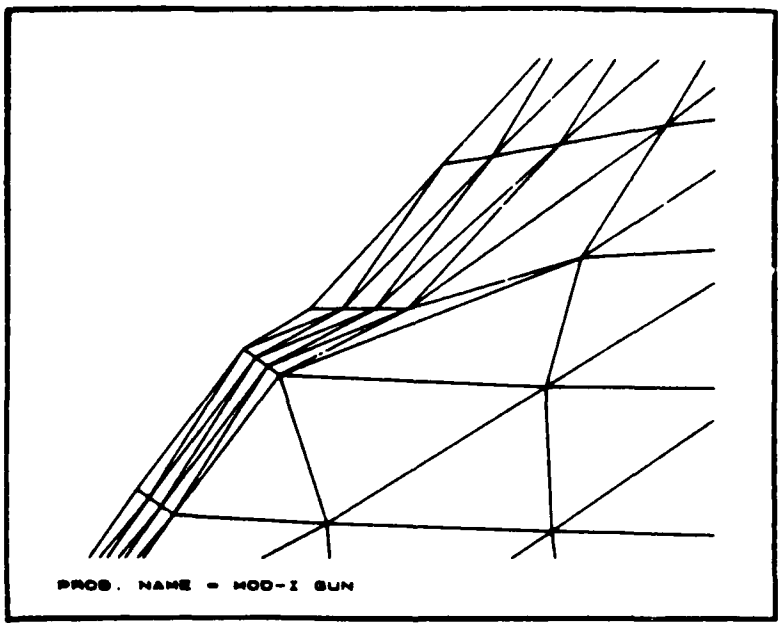
The mesh was generated with these three modifications, and a negative area mesh occurred at the top edge of the cathode as shown in Fig. 3.9a. The folding of the mesh occurred due to the close proximity of the corners and the large nodal density of the cathode mesh. This problem was corrected by using a "guard" line as shown in Fig. 3.9b.

The input data file for LATTICE, including the "guard" regions, is shown in Fig. 3.10 and the logical diagram is shown in Fig. 3.11. The mesh generated for the problem MODGUN is shown in Fig. 3.12.

Note the addition of the two "guard" regions in Fig. 3.11 and Fig 3.12. These "guard" regions were added to illustrate the use of a "guard" mesh in a problem.



a) Negative area mesh



b) Use of a "guard line"

Fig. 3.9 a) Negative area mesh at the top edge of the cathode. b) Use of a "guard" line to correct the problem.

```
MOD-I GUN
•2  6 •21  1 1 1 0 •9    .0254 $KIP
  1  1       0.0000       0.0000   0   1 REGION
    1    1 .000000E+00 .000000E+00     K
   47    1 2.75598     .000000E+00     B
   47   11 2.75598     .446001         B
   18   11 1.10803     .446001         B
   19   12 1.05802     .473755         B
   18   12 1.00801     .501509         B
   18   13 1.00801     .538504         B
   19   14 1.05181     .559204         B
   20   14 1.09561     .579903         B
   20   15 1.13942     .600603         B
   21   15 1.18322     .621302         B
   22   16 1.22702     .642002         B
   23   16 1.27388     .664145         B
   23   17 1.32073     .686288         B
   24   17 1.36759     .708431         B
   25   18 1.41444     .730574         B
   26   18 1.46130     .752717         B
   26   19 1.50815     .774861         B
   27   19 1.55501     .797004         B
   27   29 1.55501     1.20000         B
   18   29 1.03790     1.20000         B
   19   28 1.03790     1.15584         B
   18   27 1.03790     1.11167         B
   19   26 1.03790     1.06750         B
   18   25 1.03790     1.02334         B
   19   24 1.03790     .979172         B
   18   23 1.03790     .935005         B
   18   21 1.01403     .891708         B
   17   21 .964020     .891708         B
   17   22 .938900     .935005         B
   17   27 .938900     1.12900         B
    4   27 .572902     1.12900         B
    1   20 .288910     .813016         B
    1   19 .271059     .802000         K
    1   18 .242754     .763394         K
    1   17 .215822     .723819         K
    1   16 .190341     .683294         K
    1   15 .166344     .641873         K
    1   14 .143863     .599610         K
    1   13 .122926     .556562         K
    1   12 .103562     .512703         K
    1   11 .857949E-01 .468332         K
    1   10 .696492E-01 .423267         K
    1    9 .551457E-01 .377647         K
    1    8 .423033E-01 .331532         K
    1    7 .311391E-01 .284982         K
    1    6 .216675E-01 .238058         K
    1    5 .139012E-01 .190822         K
    1    4 .784993E-02 .143336         K
    1    3 .352192E-02 .956619E-01     K
    1    2 .922918E-03 .478624E-01     K
    1    1 .000000E+00 .000000E+00     K COUN
  2  1       0.0000       0.0000   0   -1 REGION
   18   29 1.03790     1.20000         B
   19   28 1.03790     1.15584         B
   18   27 1.03790     1.11167         B
   19   26 1.03790     1.06750         B
   18   25 1.03790     1.02334         B
   19   24 1.03790     .979172         B
   18   23 1.03790     .935005         B
   18   21 1.01403     .891708         B
   17   21 .964020     .891708         B
   17   22 .938900     .935005         B
   17   27 .938900     1.12900         B
    4   27 .572902     1.12900         B
    1   20 .288910     .813016         B
    1   19 .271059     .802000         K
    1   18 .242754     .763394         K
    1   17 .215822     .723819         K
    1   16 .190341     .683294         K
    1   15 .166344     .641873         K
    1   14 .143863     .599610         K
    1   13 .122926     .556562         K
    1   12 .103562     .512703         K
    1   11 .857949E-01 .468332         K
    1   10 .696492E-01 .423267         K
    1    9 .551457E-01 .377647         K
    1    8 .423033E-01 .331532         K
    1    7 .311391E-01 .284982         K
    1    6 .216675E-01 .238058         K
    1    5 .139012E-01 .190822         K
    1    4 .784993E-02 .143336         K
    1    3 .352192E-02 .956619E-01     K
    1    2 .922918E-03 .478624E-01     K
    1    1 .000000E+00 .000000E+00     K COUN
  3  1       0.0000       0.0000   0   -1 REGION
   47   11 2.75598     .446001         B
   19   11 1.10803     .446001         B
   19   12 1.05802     .473755         B
   18   12 1.00801     .501509         B
   18   13 1.00801     .538504         B
   19   14 1.05181     .559204         B
   20   14 1.09561     .579903         B
   20   15 1.13942     .600603         B
   21   15 1.18322     .621302         B
   22   16 1.22702     .642002         B
   23   16 1.27388     .664145         B
   23   17 1.32073     .686288         B
   24   17 1.36759     .708431         B
   25   18 1.41444     .730574         B
   26   18 1.46130     .752717         B
   26   19 1.50815     .774861         B
   27   19 1.55501     .797004         B
   27   29 1.55501     1.20000         B COUNT
  4  1       0.0000       0.0000   0   0 REGION
    2   20  0.29759      0.81301         6
    3   20  0.3062736    0.81301         6
    4   20  0.3149606    0.81301         6 COUNT
  5  1       0.0000       0.0000   0   0 REGION
   19   11 1.10803     0.44600         6
   19    9 1.10803     0.32210         6
   19    1 1.10803     0.00000         6 COUNT
  6  1       0.0000       0.0000   0   0 REGION
   19    9 1.10803     0.32210         6
   47    9 2.75598     0.26760         6 COUNT
```

Fig. 3.10 Data file LATTICE.DAT for the problem MODGUN.

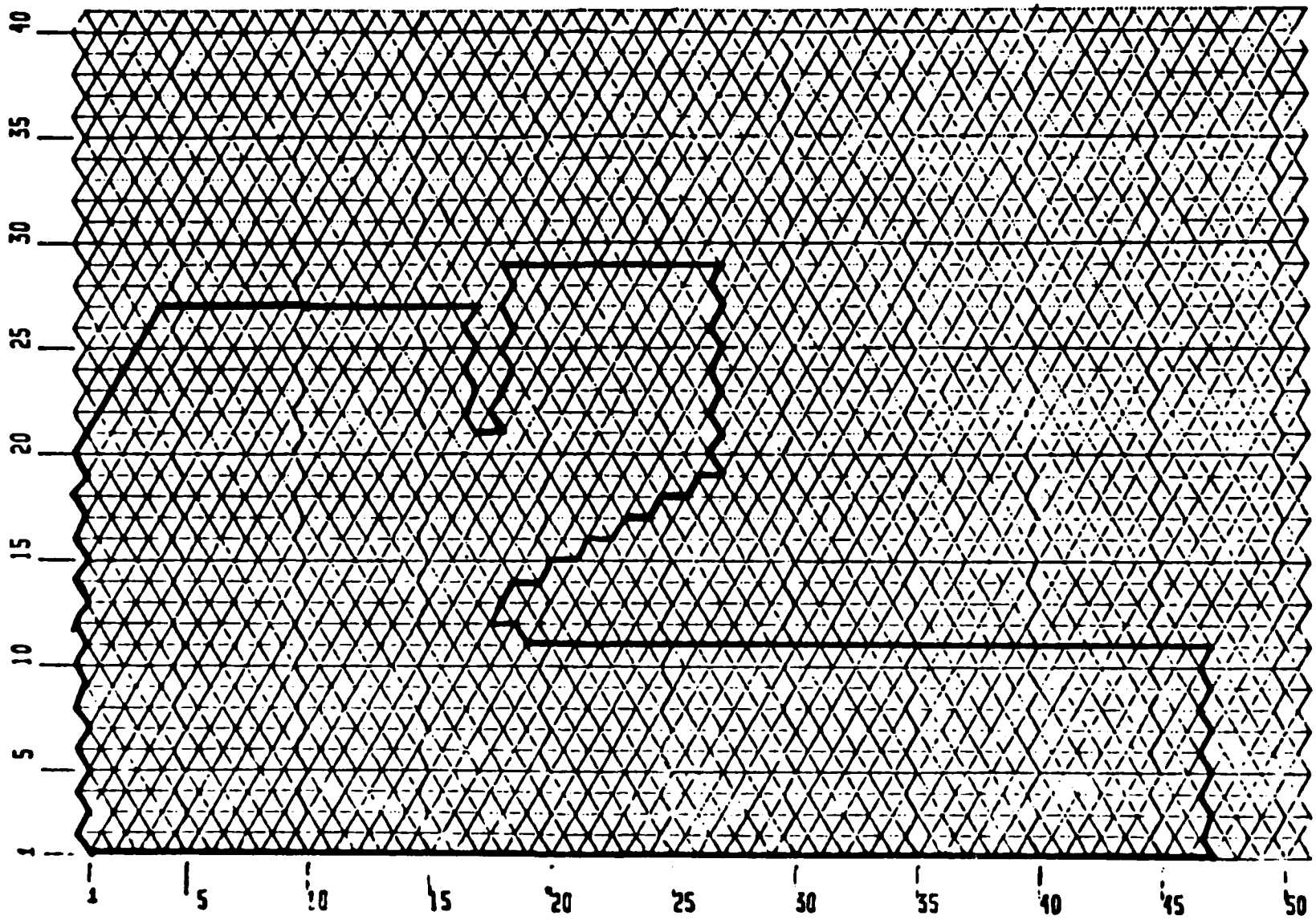


Fig. 3.11 Logical diagram for the problem MODGUN.

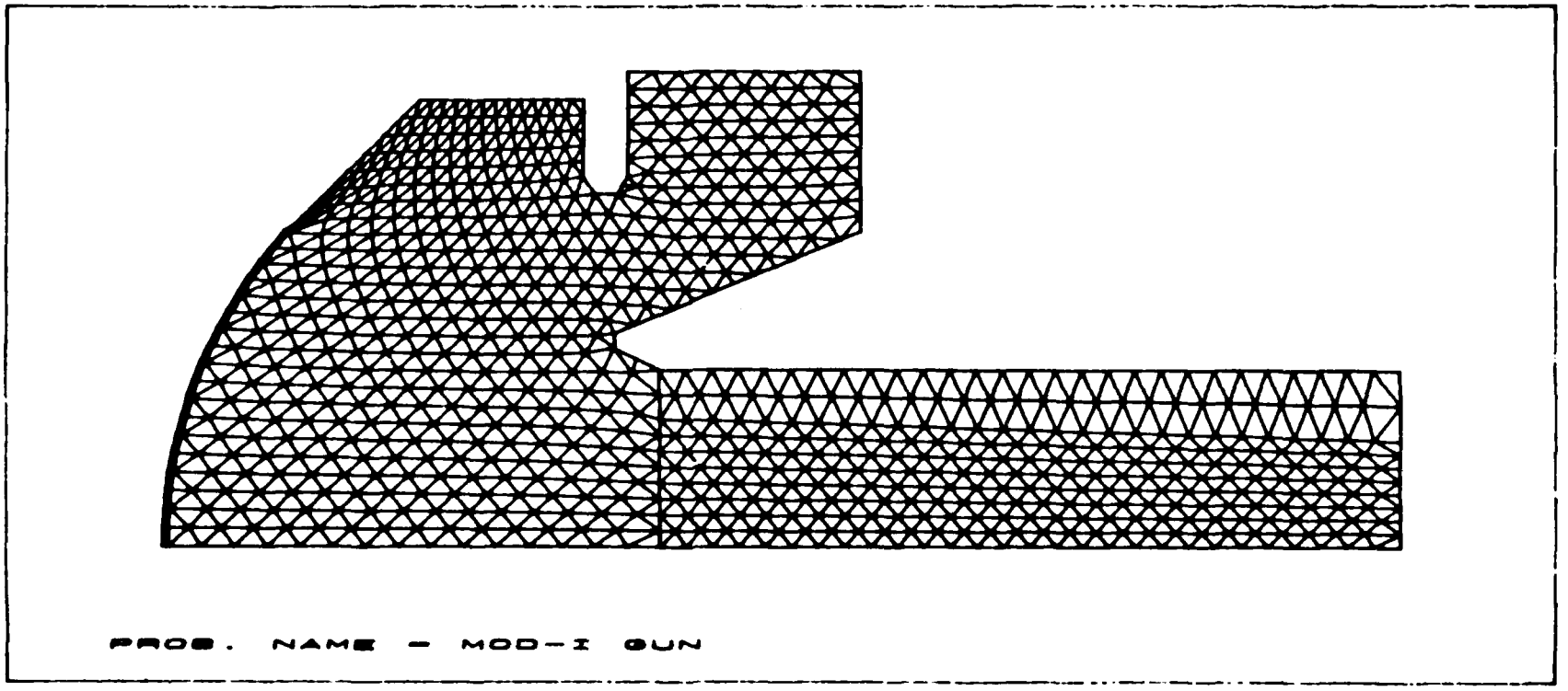


Fig. 3.12 The mesh generated by LATTICE for the problem MODGUN.

IV. POISSON (Solution of the Space-Charge Flow Problem)

The program POISSON solves, by successive point over-relaxation, Poisson's equation in two dimensions. This portion of the manual describes the procedures for solving the space-charge flow problem. For magnetostatic problems, the user should refer to the original manual for instructions to solve this type of problem.

4.1 Input Data for POISSON

There are two groups of input data for the space-charge flow problem and three groups of input data for the electrostatic problem:

1) The first group of input data is an integer number, "NUM", which is the number of the "dump" to be read and processed by POISSON. For space-charge flow and electrostatic problems, the dump number 0 (zero) written by LATTICE is to be used (see Fig. 1.3 of Sec. 1.5). Dump numbers greater than zero are written by POISSON and dumps greater than one are associated with magnetostatic problems.

2) The second group of input data are the input values for the "problem-constants", CON's. These input values can change previously defined CON values read from dump 0, or specify values which are used only in POISSON and

were not input in the LATTICE "problem-constant" data. The format of the second group of input data is read in by the format free input routine and is described in Sec. 1.4. This format is exactly the same as data group (2) of LATTICE.

3) The third group of data is a flag to terminate the POISSON run and is associated with the electrostatic and magnetostatic problem. The last data line, -1, is typed to the terminal to terminate the run.

4.2 POISSON "Problem-Constants" (CON's)

This section describes the CON's which controls the solution of Poisson's equation. Some CON's apply to magnetostatic problems, and the user should refer to the original manual.

4.2.1 Description of the CON's

The following sections describe the CON's used in POISSON. The default values of the CON's are shown in parentheses.

4.2.1a Parameters Defining the Basic Problem

CON(6)=MODE
(-2)

Code for the permeability in iron.
Mode=-2, mu-infinite in iron
Mode=-1, mu-finite but constant using
gamma=1.0/mu=CON(10)=FIXGAM.
Mode=0, mu-finite-variable, using the internal (B,gamma) table or a table input by the user.

CON(7)=STACK (1.0)	Stacking (fill) factor in iron regions.
CON(10)=FIXGAM (0.004)	The value of gamma used in a mu-finite but constant solution (CON(6)=MODE=-1). Also the value to which gamma is initialized for a mu-finite-variable solution.
CON(18)=NPERM (0)	The number of permeability tables to be read in as data by program POISSON.
CON(19)=ICYLIN (1)	Indicator for cylindrical symmetry problem. ICYLIN=1, cylindrical symmetry (r,z) ICYLIN=0, two dimensional (x,y)

4.2.1b Boundary Conditions

CON(20)=INPUTA (0)	The number of fixed potential values to be read in as data by program POISSON.
CON(50)=IHDL (100000)	During the POISSON iteration, the number of cycles between making a quasi-integral Hdl calculation around the Dirichlet boundary. This sometimes speeds the convergence, particularly for non-symmetrical "H" magnets.

4.2.1c Limits, Printed Output and "Dump" Options

CON(30)=MAXCYC (100000)	Maximum number of iteration cycles.
CON(31)=IPRFQ	The cycle print frequency during

(0)

(0) the POISSON iteration. The default value of 0 indicates that the iteration information will be printed only on the first and last cycles. Input values of IPRFQ must be an integer multiple of IVERG=CON(87).

CON(32)IPRINT
(0) of the mesh points

IPRINT=-1, Print (x,y) coordinates (LATTICE only).
IPRINT=0, nothing
IPRINT=1, Print the vector potential array when CON(6)=MODE=0.
IPRINT=2, Print the |B| in iron triangles.
IPRINT=4, Print the B_x,B_y) in iron triangles.
Any combination of these options may be specified by the sum of "IPRINT" values.

CON(35)=NODMP
(0)

Indicator not to write a "dump" at the completion of a POISSON solution.
NODMP=0, Write "dump"
NODMP=1, Do not write "dump"

4.2.1d Edit Options (Fields and Gradients)

CON(38)=XORG
(0.0)
CON(39)=YORG
(0.0)

The origin Z0 in the polynomial expansion for the vector potential. A (x,y)=re[sum Cn*(Z-Z0)**n], the derivatives of which give the field and gradient. Note - for cylindrical symmetry problems, XORG=0, always.

CON(42)=KMIN (1)
CON(43)=KTOP (KMAX)
CON(44)=LMIN (1)
CON(45)=LTOP (1)

The (K,L) limits of the region in which the fields and gradients are to be calculated at each mesh point. See CON(54) through CON(57) for an alternative scheme of defining the region in which the fields and gradients are to be calculated.

CON(46)=ITYPE (2)

A code specifying the problem symmetry and determining which "C_n" term appears in F(z), and whether the "C_n" are real, imaginary, or complex.

ITYPE=1, No symmetry.
ITYPE=2, Midplane symmetry
ITYPE=3, Elliptical quad aperture
ITYPE=4, Symmetrical quad.
ITYPE=5, Skew elliptical quad aperture.
ITYPE=6, Symmetrical "H" magnet or elliptical sextupole aperture.
ITYPE=7, Symmetrical sextupole
ITYPE=8, Elliptical octupole aperture.
ITYPE=9, Symmetrical octupole.

For all of the above symmetry codes, except for ITYPE=1, or =5, field lines are perpendicular to the x-axis. For ITYPE=5, the x-axis is a field line.

For cylindrical symmetry (vector) problems:
ITYPE=1, No symmetry
ITYPE=2, Midplane symmetry, r-axis has field lines perpendicular.

For cylindrical symmetry (scalar) problems,
ITYPE=1, No symmetry
ITYPE=2, Midplane symmetry, r-axis has V lines perpendicular.
ITYPE=3, Midplane symmetry, r-axis is a V=constant line.

CON(47)=W2ND (0.125)

The weight factor for the second nearest neighbors used in determining the "C_n" in the

expansion for A(x,y).

CON(48)=ISECND

Indicator to use second nearest neighbors in determining the "C_n" in the expansion for A(x,y).
ISECND=1, Use first and second neighbors
ISECND=0, Use first neighbors only.

CON(54)=XMIN
(0.0)
CON(55)=XMAX
(0.0)
CON(56)=YMIN
(0.0)
CON(57)=YMAX
(0.0)

The (x,y) limits of a grid over which the fields and gradients are to be calculated. Delta-x and Delta-y in this grid are specified by CON(43)=KTOP and CON(45)=LTOP as follows:

Dx=(XMAX-XMIN)/(KTOP-1) and
Dy=(YMAX-YMIN)/(LTOP-1).

4.2.1e Current Parameters and Options

CON(8)=BDES
(1.0E15)

BDES=|B| at the location KBZERO=CON(40), LBZERO=CON(41). If BDES is not equal to 1.0E15, the current factor, XJFACT=CON(66), will be adjusted so that |B|=BDES within a tolerance XJTOL=CON(67).

CON(40)=KBZERO
(1)
CON(41)=LBZERO
(1)

The (K,L) coordinates specifying the location of BDES for adjusting the current factor.

CON(49)=NFIL
(0)

The number of current filaments to be read in as data by problem POISSON.

CON(66)=XJFACT
(1.0)

The factor by which all current densities (except current filaments) will be scaled in program POISSON. XJFACT=0.0, indicates a scalar potential

problem (no currents). LATTICE will automatically set XJFACT=0.0 when IBOUND=-1.

4.2.1f Over Relaxation Factors

CON(74)=RHOPT1
(1.90)

See RHOAIR=CON(75)

CON(75)=RHOAIR
(1.90)

The over-relaxation factor in POISSON for air and interface points (and iron points during a mu-finite-but constant solution). This factor is automatically optimized during the iteration if the initial value=RHOPT1=CON(74).

CON(77)=RHOFE
(1.0)

The over-relaxation factor in POISSON for iron points during a mu-finite-variable solution.

CON(78)=RHOGAM
(0.08)

The under-relaxation factor in POISSON for gamma during a mu-finite-variable solution.

CON(80)=ISKIP
(1)

The number of cycles between recalculating the gammas during a mu-finite-variable solution.

4.2.1g Convergence Criteria

CON(85)=EPSILA
(5.0E-7)

The convergence criterion for the potential solution of air and interface points (and for iron points during a mu-finite-but constant solution).

CON(86)=EPSILI (5.0E-7)

The convergence criterion for the potential solution of iron points during a mu-finite-variable solution.

CON(87)=IVERG (10)

The number of cycles between making the convergence tests during the POISSON iteration. The default value of 10 should not be altered when using the option to optimize the over-relaxation factor RHOAIR=CON(75).

4.2.1h Parameters for Harmonic Analysis

CON(110)=NTERM (5)

The number of coefficients to be obtained.

CON(111)=NPTC

The number of equidistant points on the arc of a circle with it's center at the origin, at which points the vector potential is to be interpolated. Fourier analysis of the vector potential at these points yields the harmonic coefficients. NPTC should be input equal to (approximately) the number of mesh points adjacent to the arc.

CON(112)=RINT

The radius of the arc for vector potential interpolation and Fourier analysis. RINT should be less than, by at least one mesh space, the radius to the nearest singularity, i.e. the pole or coil.

CON(113)=ANGLE

The extent of interpolation arc included in the problem-angle in degrees.

CON(114)=RNORM

The aperture radius or other harmonic coefficient normalization

radius.

CON(115)=ANGLZ (0.0)	The angle in degrees from the x-axis to where the integration arc begins.

Sec. 4.2.2 Parameters for Space-Charge Flow

This section describes the CON's which are used in POISSON for the space-charge flow problem. The description of the CON's for space-charge flow are described in Sec. 3.7.2.

Sec. 4.2.2a Printing Options for the Results of POISSON

MPRINT (CON(143)), controls the printing options for the results of POISSON. There are two output files where the results are printed:

1) POISSON.OUT
2) POSITION.OUT

1) "POISSON.OUT" contains the printed output of the CON's, cathode data, potential at each mesh point, the charge assigned to each mesh point, current emitted, convergence information, perveance and total beam current.

MPRINT has three levels that may be specified in order to retrieve some or all of the above information. These three levels are:

MPRINT=1 When MPRINT is set to one, the following items are printed:

1) CON's
2) Number of iterations for the potential solution to converge for each current cycle.

3) Total beam current, perveance and DCBAR for each current cycle.

MPRINT=2 In addition to the above items, the following items are printed when MPRINT is set to two:

1) cathode data

2) CBAR and emitted current of each emitting point for each current cycle.

MPRINT=3 In addition to the above items, the following items are printed when MPRINT is set to three:

1) the potential at each mesh point for each current cycle.

2) charge assigned to each mesh point for each current cycle.

2) "POSITION.OUT" contains the trajectory information of the electrons when MPRINT is greater than one. Time increment, z(t), r(t), z-velocity, r-velocity, theta-velocity, perpendicular velocity and two relativistic gammas are printed at INCPLT=CON(146) increments. For problems which have a large amount of time steps, INCPLT should be set accordingly so the plotting arrays do not overflow. A value of INCPLT=3 is typical for problems with many time steps.

Two gammas are calculated by Eq. (5.3) and Eq. (5.25) of the thesis. These two gammas should almost equal each other. If they do not, energy is not being conserved, and the time step is probably too large.

Sec. 4.3 Computation of the Electron Trajectories of MODGUN

In Sec. 3.8, LATTICE created the triangular mesh and created the data file POISSON.DAT (dump 0) for the problem MODGUN. POISSON will read in POISSON.DAT and compute the electron trajectories.

Once POISSON has been executed, the program will prompt on the screen:

INPUT DUMP NUMBER, "NUM"

NUM is the "dump" number to be read in, as discussed in Sec. 4.1. Dump 0 is to be read in by POISSON, therefore the user responds by typing in 0 and then a return.

POISSON will next ask the user to:

INPUT THE CONS YOU WANT TO CHANGE

As discussed in Sec. 4.1, this is the second group of input data. At this point the user may want to change some CON's that were not specified in LATTICE. For example, to get a full diagnostic print out, MPRINT is set to three. The following is typed on the terminal:

*143 3 S

where the format of this line is discussed in Sec. 3.1.

The program will now compute the electron trajectories and write out the plotting data to dump 1.

V. PLOT

Sec. 5.1 The Plotting Program PLOT

The program PLOT, plots the mesh, equipotential lines and electron trajectories. PLOT uses Hewlett Packard's Advanced Graphics Package (AGP) to perform the plotting.

A minimum of 1 input line must be provided for PLOT:

1) The first input line gives values for five parameters for the plot:

"NUM" is the "dump" number to be read. The mesh is written on dumps "0" and "1", and the equipotential lines and electron trajectories are written on dump "1".

"ITRI" is a flag whether or not to plot the triangles of the mesh.

ITRI=0, do not plot the triangles

ITRI=1, plot the triangles

"NPHI" is the number of equipotential lines to be plotted. The program does not plot the minimum or maximum potential values which is usually a boundary.

"INAP" is the indicator to read, on the next data card, the minimum and maximum values (AMIN and AMAX) of the equipotential lines to be plotted.

INAP=0, do not read AMIN and AMAX

INAP=1, read AMIN and AMAX on the next input line

The values plotted are between (AMAX-DELTA) and (AMIN+DELTA) where

DELTA=(AMAX-AMIN)/(NPHI+1)

"IRAYS" is a flag whether or not to plot the electron trajectories.

IRAYS=0, do not plot the trajectories

IRAYS=1, plot the trajectories

2) The second input line is optional and gives values for four parameters for the plot: XMIN, XMAX, YMIN, YMAX.

These four parameters specify the limits for the plot, which may be any part of the problem. If this data is skipped, the entire problem will be plotted.

Other options such as sending the plot to the printer and zooming in on the plot are also available by answering the appropriate questions prompted by PLOT.

Sec. 5.2 Plotting the Results of MODGUN

The triangular mesh may be plotted after LATTICE has been executed. The mesh of MODGUN, Fig. 3.12, was plotted by

specifying the following five parameters:

NUM,	ITRI,	NPHI,	INAP,	IRAYS
0	1	0	0	0

The equipotential lines and electron trajectories may be plotted after POISSON has been executed. For the example problem, MODGUN, five equipotential lines between 2000.0 and 6000.0 volts, and the electron trajectories may be plotted by specifying the following seven parameters:

NUM,	ITRI,	NPHI,	INAP,	IRAYS
1	0	5	1	1

and

AMIN	AMAX
2000.0	6000.0

The plot of the equipotential lines and electron trajectories is shown in Fig. 5.1.

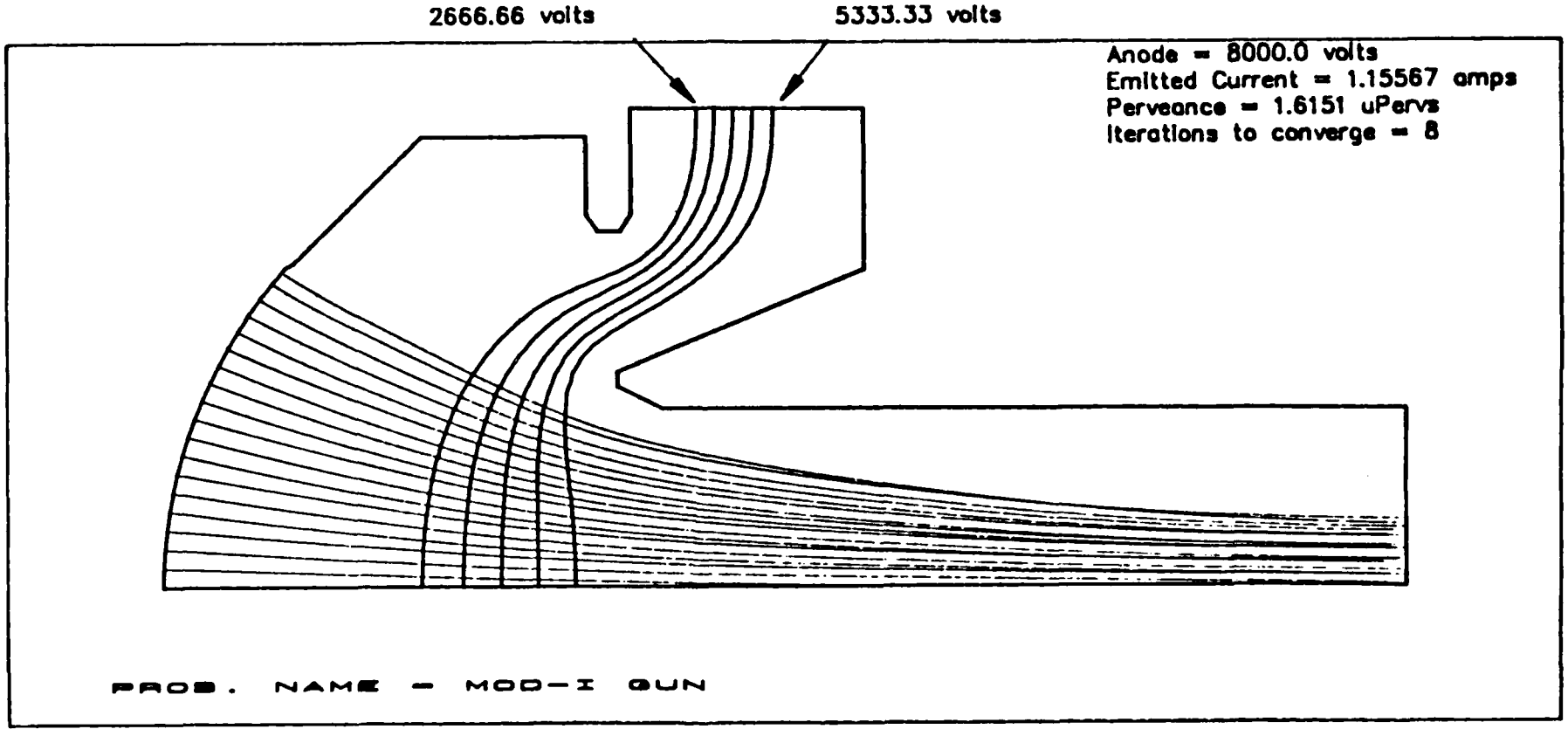


Fig. 5.1 Plot of the equipotential lines and electron trajectories.

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control, Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C^3I systems. The areas of technical competence include communications, command and control, battle management, information processing, surveillance sensors, intelligence data collection and handling, solid state sciences, electromagnetics, and propagation, and electronic, maintainability, and compatibility.

END

8-87

DTIC