CLRC

**Technical Report**
RAL-TR-1998-088

# The EVEREST Post-Processor Module: Version 4.0

J V Ashby R F Fowler and C Greenough

6th January 1999

COUNCIL FOR THE CENTRAL LABORATORY OF THE RESEARCH COUNCILS

Enquiries about copyright, reproduction and requests for additional copies of this report should be addressed to:

The Central Laboratory of the Research Councils
Library and Information Services
Rutherford Appleton Laboratory
Chilton
Didcot
Oxfordshire
OX11 0QX
Tel: 01235 445384 Fax: 01235 446403
E-mail library@rl.ac.uk

**ISSN 1358-6254**

Neither the Council nor the Laboratory accept any responsibility for loss or damage arising from the use of information contained in any of their reports or in any communication about their tests or investigations.

# The EVEREST Post-Processor Module: Version 4.0

JV Ashby, RF Fowler and C Greenough

December 1998

## Abstract

In this report we describe Version 4.0 of the EVEREST Post Processor Module which forms part of the EVEREST suite of programs. The post-processor allows the user to visualise the results of the simulations performed by the Everest Solver module. The input required is the device geometry, mesh, doping and solution variables. Using cut planes through the device, the Post Processor supports contour, isometric, line and vector plots of most physical quantities of interest. Current-voltage characteristics and transient response can also be displayed. This report outlines the methods used and gives full details of the user interface.

The EVEREST suite is one of the products of the ESPRIT project EVEREST (ESPRIT 962E-17, *Three-Dimensional Algorithms for a Robust and Efficient Semiconductor Simulator with Parameter Extraction*). The original authors of the Post-Processor Module were P.A. Mawby, M.S. Towers, G.A. Duffet, J. Zhang and G.J. Huang of University College, Swansea.

A copy of this report can be found at the Department's web site (*http://www.dci.clrc.ac.uk/*) under page *Group.asp?DCICSEMSW* or anonymous ftp server *www.inf.rl.ac.uk* under the directory *pub/mathsoft/publications*

Mathematical Software Group
Department for Computation and Information
Rutherford Appleton Laboratory
Chilton, DIDCOT
Oxfordshire OX11 0QX

# Contents

# 1 Introduction

This manual describes the use of the post-processor module of the three-dimensional semiconductor device simulation suite EVEREST. It also provides an overview of the algorithms used within the software

The post-processor runs interactively on a suitable graphics device (e.g. a workstation running X-windows) and uses GKS as the standard graphics interface. The use of GKS makes the post-processor portable to any system that provides such a library. The public domain software package X-GKS is portable to most UNIX systems and is included in the EVEREST software distribution. Other versions of GKS may be linked in preference to X-GKS on non-UNIX systems or to support other graphics output devices.

The post-processor exists as a stand alone module in the three-dimensional semiconductor simulation suite. The information required by the post-processor is generated by the pre-processor and solver modules. These packages are linked by a data interface consisting of the following family of neutral files:

| | |
|---|---|
| `<filename>.GEO` | (geometric description) |
| `<filename>.MSH` | (mesh) |
| `<filename>.DOP` | (doping profiles) |
| `<filename>.CAT` | (directory of solution cases) |
| `<filename>.PHY` | (physical properties) |
| `<filename>.RES` | (results). |

When the post-processor is run (using the command `pospro`) the user is presented with the command line prompt `Post:` which indicates that one is in the top command level. Typing `help` at this point will give a list of the commands available within this command level. These commands are listed in Appendices A and I, the latter being the internal commands which are available in all command levels.

The current implementation of the post-processor has eight command levels. Most information is read from neutral files in the initial command level (POST). This level also allows selection of the cut plane and variable to be displayed before selecting one of the other command levels (GEOMETRY, CONTOUR, ISOMETRIC, LINE, VECTOR, CHARACTERISTIC or TRANSIENT) to display the data.

Viewing results calculated from the three-dimensional space is not possible on a two-dimensional graphics display, since the results have essentially four degrees of freedom (three spatial, plus the scalar quantity to be plotted). For this reason the results are displayed on planes which are taken as cuts through the device. The current version of the software has the constraint that cut planes must be perpendicular to one of the three major axes. After the cut-plane has been defined, it is treated wholly in terms of two-dimensional data structures. This makes the plotting routines useful for displaying two-dimensional, as well as three-dimensional results on a cut-plane. Having defined the cut plane it is necessary to select the variable to be displayed, at least for the GEOMETRY, ISOMETRIC and LINE command levels. In the present version there is the choice of the three primary variables ($\psi$, $\phi_n$, $\phi_p$), the two doping profiles ($N_A$ and $N_D$), and the eight secondary variables ($n$, $p$, $n_{ie}$, $(N_A - N_D)$, $Net_{Charge}$, $R$, $\mu_p$, $\mu_n$, $J_p$, $J_n$, $J_T$, $E$) calculated internally the post-processor. The names and units used are listed in Table 1.

Once the cut plane and plotting variable are defined the results may be viewed either as a CONTOUR plot or as an ISOMETRIC projection of the surface. In the case of the isometric projection the

| Choice name | Description | Units |
|---|---|---|
| PSI | $\psi$ - potential | $V$ |
| PHIP | $\phi_p$ - hole quasi-Fermi level | $V$ |
| PHIN | $\phi_n$ - electron quasi-Fermi level | $V$ |
| NA | $N_A$ - Acceptor doping | $cm^{-3}$ |
| ND | $N_D$ - Donor doping | $cm^{-3}$ |
| P | $p$ - Hole density | $cm^{-3}$ |
| N | $n$ - Electron density | $cm^{-3}$ |
| NIE | $n_i$ - Intrinsic level | $cm^{-3}$ |
| NETDOP | $\|N_A - N_D\|$ - Nett doping | $cm^{-3}$ |
| NETCHG | $\|N_D - N_A + p - n\|$ - Nett charge density | $cm^{-3}$ |
| REC | $R$ - Recombination rate | $cm^{-3}s^{-1}$ |
| HMOB | $\mu_h$ - Hole mobility | $cm^2V^{-1}s^{-1}$ |
| EMOB | $\mu_n$ - Electron mobility | $cm^2V^{-1}s^{-1}$ |
| JP | $J_p$ - Hole current density | $Acm^{-2}$ |
| JN | $J_n$ - Electron current density | $Acm^{-2}$ |
| JT | $J_p + J_n$ - Total current density | $Acm^{-2}$ |
| E | $E$ - Electric field | $Vcm^{-1}$ |

Table 1: Variables that may be displayed in the Post-processor.

surface may be viewed from any azimuth or elevation angle. The LINE command level allows the user to plot a one dimensional graph along any selected line through the cut plane.

The post-processor also provides the facilities to allow the geometry and mesh of the device to be examined. The three-dimensional mesh is only displayed on the surface of the device, since the full mesh of anything more complex than a cube cannot easily be understood.

## 2 Module Structure

The post-processor module has been designed to be an interactive system. Commands are issued to the program through a command decoder which reads textual input from the default input stream or a specified command file. These commands are then acted upon by the post-processor. There are eight command levels: the main command level (Post) is the entry level and is used to enter the seven lower command levels: Contour, Isometric, Geometry, Line, Vector, Characteristic or Transient.

### 2.1 POST Command Level

This is the main command level and is used to read in data from the neutral files and set up the variables to be viewed. Appendix A lists all the commands available in this level.

### 2.2 CONTOUR Command Level

This command level is used to display contours on a cut-plane taken through the device. To enter this command level you must have read in valid neutral files for the geometry, mesh, and doping. A cut plane and variable to display must also have been selected. Any variable other than doping requires that a solution neutral file has also been read. The user can select the number and range of contour levels, to use solid or line contours, etc. Appendix B lists all the commands available in the contour level.

## 2.3 ISOMETRIC Command Level

This command level is an alternative to the contour level for viewing data on a cut plane and requires the same neutral files to have been read as that level. The level uses two forms of surface plot:

- colour fill type - where each facet (corresponding to an element) is sorted into ascending order of distance from the viewing position,
- line plot - this uses a a full hidden line elimination algorithm.

Both of these types of plot have their own particular merits. The colour fill type plot is suited to raster scan devices since it is possible to re-colour pixels. The line drawing version is obviously better suited to pen plotters. Appendix C lists all the commands available in this level.

## 2.4 LINE Command Level

This command level will display graphs of pre-selected variables along any selected line within the current 2D cut plane. To enter this command level a cut plane and the variable on it must already have been defined, as for the previous two levels. The cut-plane can be rotated through any angle and then a horizontal or a vertical line can be selected. The user is also provided with the facility for plotting multiple graphs on a single axis system or, when there are graphs of different variables, these may be plotted on different axis systems. This provides a very flexible facility for plotting different graph sets. Appendix E lists all the commands available in this level.

## 2.5 GEOMETRY Command Level

This command level will display projections of the device geometry as well as the mesh on the device's surface. The geometry can be viewed as soon as a neutral file with the `.GEO` suffix is produced by the pre-processor. This is useful in checking the input data before a simulation is run. Once the mesh neutral file (`.MSH`) has been generated and read in it is possible to check the surface mesh in this level. Note that refined meshes generated by the Solver module can not currently be displayed in this command level. Appendix D lists all the commands available in this command level.

## 2.6 VECTOR Command Level

This command level is used to display vector variables on a cut-plane taken through the device. The vector variables that may be plotted are hole current ($J_p$), electron current ($J_n$), total current ($J_T$) and electric field ($E$). To enter this command level you must have read in a valid set of neutral files, selected a solution case and defined a cut plane. The user can select the number of vectors to plot and the vector type to use. Note that this command level uses the same scaling as the CONTOUR level, so it is possible to add current or electric field arrows to a previous contour plot. Appendix F lists all the commands available in this level.

## 2.7 CHARACTERISTICS Command Level

This command level is self-contained and no neutral files have to be read before entering it. A results neutral file must be read in which should contain the results from a static simulation using at least two different bias cases. The user is then able to produce current-voltage plots for one or more of the contacts in the device. A wide range of options are available to control the presentation of the graphs. Appendix G lists all the commands available in this command level.

## 2.8 TRANSIENT CHARACTERISTICS Command Level

This command level is very similar to the CHARACTERISTICS command level but is designed for analysis of transient data. No neutral files need be read before entering this level. A results neutral file must be read in which should contain the results from a transient simulation. The user is then able to produce current-time and voltage-time curves for one or more contacts on the device. As with the CHARACTERISTICS level the user has a wide range of options to control the graphical presentation. This level has the ability to perform a time integration of the current arriving at each contact in the transient simulation. This can be useful in analysis of charge generation event simulations. Appendix H lists all the commands available in this command level.

# 3 Algorithm Descriptions

## 3.1 Generating a Cut-plane from a Three-dimensional Mesh

A cut-plane through a three-dimensional object is a plane of intersection as shown in Figure 1. The command for defining the cut-plane has two arguments (see Appendix A):

- The major axis to which the cut-plane is perpendicular
- The distance between the origin and the point at which the plane intersects with the axis.

The present release of the software allows a cut-plane at any position, subject to it being parallel to two of the major axes ($x$, $y$, $z$) and intersecting the device.

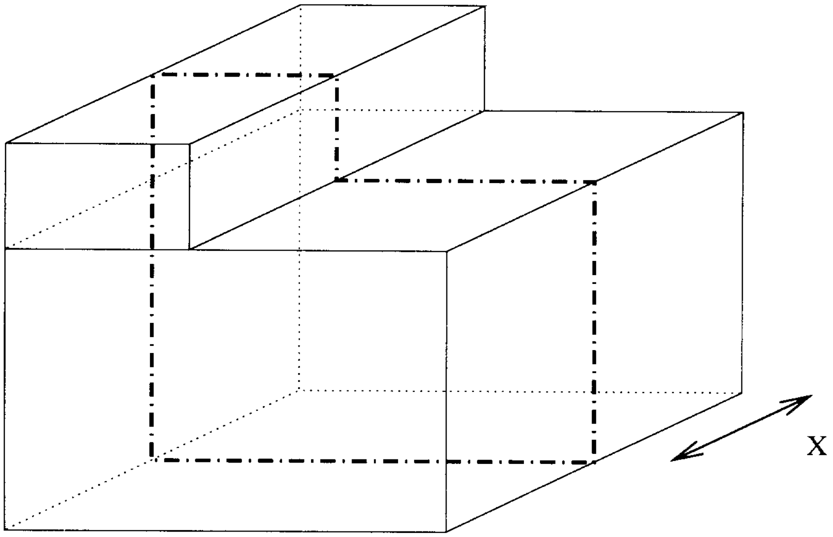


Figure 1: Illustration of a cut-plane parallel to $y$ and $z$ axes at a fixed $x$ value.

Since the cut-plane will not in general be coincident with a plane of nodes in the 3D mesh, it is necessary to search through all edges in the 3D mesh, checking for intersections with the 2D plane. Each such intersection point will be treated as a node in the 2D mesh. The elements in the 2D plane may be either triangular or quadrilateral according to the cross section of the 3D element.

Since the major algorithms for isometric and contour plotting are based on a 2D triangular mesh, it is necessary to convert each quadrilateral element into two triangular ones in the present implementation. This means that quadrilateral elements are drawn as two triangles using mesh display commands.

Physical quantities are calculated first on the nodes in the 3D mesh and then interpolated onto the nodes of the 2D mesh. A simple linear interpolation is used between the two nodes at either end of 3D mesh edge that intersects the 2D plane. This interpolation can introduce some extra noise in the results, particularly for tetrahedral meshes when the cut-plane is not close to a plane of 3D nodes. Another source of noise in the results for coarse meshes occurs when viewing values which are not naturally defined at nodes. In particular electric field and current densities have to be calculated along 3D mesh edges and averaged onto 3D nodes before interpolation to the 2D mesh.

All the graphical operations displaying results use the 2D triangular mesh as the basis for the surface and assume linear variation within each element.

## 3.2 Generation of Contour Maps on Cut-plane

The algorithms used for contouring are fairly conventional in structure. A two-dimensional data structure for triangular elements is defined in terms of an element topology array and an array of nodal co-ordinates. Initially a search is performed on the element topology array to determine those element sides which lie on the boundary. This search is straightforward to implement; the elements are scanned in turn, whilst recording a count of the number of times each side occurs. If at the end the edge has a count of more than one, then it must be internal, otherwise it is part of the boundary. This search gives you the option of plotting the cross-section either with or without the mesh superimposed on the contours. For automatically refined meshes generated by the Solver module the above boundary detection algorithm can sometimes give erroneous results. This is due to the fact that the cut plane may have non-conforming elements where an edge has two elements on one side, but only a single element on the other.

Contours are generated on an element by element basis. For each element the contour heights are scanned in sequence, if a contour is found to cut an element (height of contour is within the vertex height range) then it will cut the element in two places (since linear shape functions are used). The two points of intersection are then joined to give the required contour.

## 3.3 Generation of Isometric Surface Projections

Hidden line (or surface) removal for non-regular three dimensional objects is a complex problem. There are two basic approaches commonly used for the display of such objects:

- Back to front method
- Front to back method

Both of these methods have been implemented in the present software for producing surface plots of results on two-dimensional cut-planes.

The back to front method can only be implemented on raster scan devices, as it relies on the fact that surfaces which have already been drawn can be wholly or partially overdrawn by surfaces drawn subsequently. The basic philosophy is that, given a set of non-interpenetrating surfaces that together form the object of interest, they can be ordered in such a way that, when plotted, the correct hidden surface removal is performed automatically [3]. The main problem is sorting the surfaces into a suitable order. An adaptation of the very general Newell, Newell and Sancha sorting process (see [4]) is used in this program. When all the surfaces have been sorted into the correct front to back order they are then plotted in reverse order (i.e. the hindmost surface is plotted first and the foremost surface last).

The algorithm for hidden-line removal plotting takes the following data as input:

- A collection of triangles specifying a three-dimensional surface
- A collection of three-dimensional co-ordinates specifying nodes of triangles.
- The location (in terms of rotation and elevation) of the viewer

The algorithm uses the input data to plot the wire frame of the surface with hidden-line removal. The three main stages are:

1. Projecting the three-dimensional surface into two-dimensional space without losing depth information
2. Distributing triangles into a *cell* array according to their positions in two-dimensional space
3. Dealing with triangles stored in each *cell* by hidden-line removing and plotting them.

Parallel projection used for efficiency. The projection is made through three-dimensional co-ordinate transformation with the given viewing and elevation angles. After projection, one has the $x$, $y$ co-ordinates in the two-dimensional projected plane as well as the $z$ co-ordinates of the depth.

Distributing three-dimensional triangles into separate cells in two-dimensional space is the most important step for gaining efficiency. The algorithm finds the extent of the three-dimensional surface in the two-dimensional projected plane. The extent of the surface, a rectangle, is then divided into $M \times M$ non-intersecting smaller rectangles of the same size, called it cells, where $M$ is a constant. The algorithm then loops over each triangle to distribute it into those cells which intersect the extent of the triangle.

The distribution of the triangles is made by building a two-dimensional cell array so that each entry, $cell_{i,j}$, corresponds to a particular rectangular cell in two-dimensional space. The value of $cell_{i,j}$ points a linked list of the (index) numbers of the triangles distributed in the cell.

The selection of the constant $M$ is important for the efficiency of the algorithm. A reasonable choice is $M = \sqrt{N}$ where $N$ is the number of triangles in the surface. With such $M \times M$ divisions of the two-dimensional extent of the surface, only a few triangles appear in a single *cell* on average because only $N$ cells exist in the whole two-dimensional extent.

After distributing triangles, each *cell* can be dealt with individually without accessing other cell information. The hidden-line removal proceeds as follows for each cell:

- Clip triangles distributed in the cell to obtain a set $S_1$ of line segments,
- Compute all the intersections of the segments in $S_1$ to obtain a new line segment set $S_2$ where no two line segments innerly intersect each other,
- For each line segment L in $S_2$, check each triangle stored in the cell which includes the mid-point P of L in the two-dimensional projected space to see if P is behind the triangle in its original three-dimensional space from the view direction. If no check fails then line segment L is plotted.

## 3.4 Generating Isometric Projections of the Device Geometry

The geometry command level of the post-processor provides isometric views of the device geometry. The surface mesh can also be superimposed onto the plot if required. The basic principle for obtaining these plots is to draw only those surfaces which can be seen by the viewer. This is determined by comparing the outward normals to the boundary surfaces and the viewer's position as defined by the angles of elevation and rotation. If the outward normal points toward this direction then the surface is assumed visible and will be plotted. The painter's algorithm (or back-to-front method) is then used to plot these surfaces in order to obtain hidden surface removal [3]. Surfaces that are plotted in this manner may be obscured by other surfaces plotted latter.

The mesh information also contains lists of element faces that lie on each surface. It is therefore possible to superimpose the mesh onto each surface as it is plotted. The final output is thus a three-dimensional geometric plot of the device with the mesh plotted on the external surfaces. Refined meshes from the Solver module lack the mesh surface information at present.

# 4 Example set

## 4.1 Introduction

This section presents five examples where the post-processor is used to examine results produced by the other EVEREST modules. For each case a series of commands is listed which can be used to inspect the data along with selected figures to show the output that can be expected from the post-processor. For clarity of presentation all the commands are unabbreviated and written in upper case letters while the parameters associated with them are written in lower case letters. This is not a requirement of the program - free use of upper and lower case letters is allowed.

## 4.2 Diode Containing One-dimensional Effects

### Description

The device has dimensions $10 \times 1 \times 1\mu m$ in $x$, $y$, and $z$ directions and is divided into two blocks of equal size with their interface at $x = 5\mu m$. The blocks are uniformly doped with impurities of acceptor type to $10^{16}cm^{-3}$ in one block, and of donor type to $5 \times 10^{16}cm^{-3}$ in the other. The mesh contains 40, 1 and 1 divisions in $x$, $y$ and $z$ directions which gives 164 nodes and 40 hexahedral elements. The contacts are at $x = 0$ and $x = 10$ and the applied biases are $0V$ at the donor contact and $0V$ and $-500V$ at the acceptor contact.

Figure 2 gives the output for this example, showing the screen split into 4 areas. The four plots are (starting at the top right hand corner and proceeding clockwise):

- The device geometry plot
- An isometric view of the electrostatic potential on the plane $z = 0$ with a bias voltage of $500V$ The mesh is plotted below the surface.
- A contour plot of the same result
- An isometric plot of the potential with contours plotted below the surface.

### Input to the post-processor

```
INPUT pn all
WINDOW 4 2
GEOMETRY
PLOT all
RETURN
SELECT pn 2
PLANE z 0.0
VARIABLE psi
WINDOW 4 1
ISOMETRIC
PLOT axes
PLOT line
PLOT contour
RETURN
```

```
WINDOW 4 3
CONTOUR
PLOT all
RETURN
WINDOW 4 4
ISOMETRIC
PLOT all
RETURN
QUIT
```

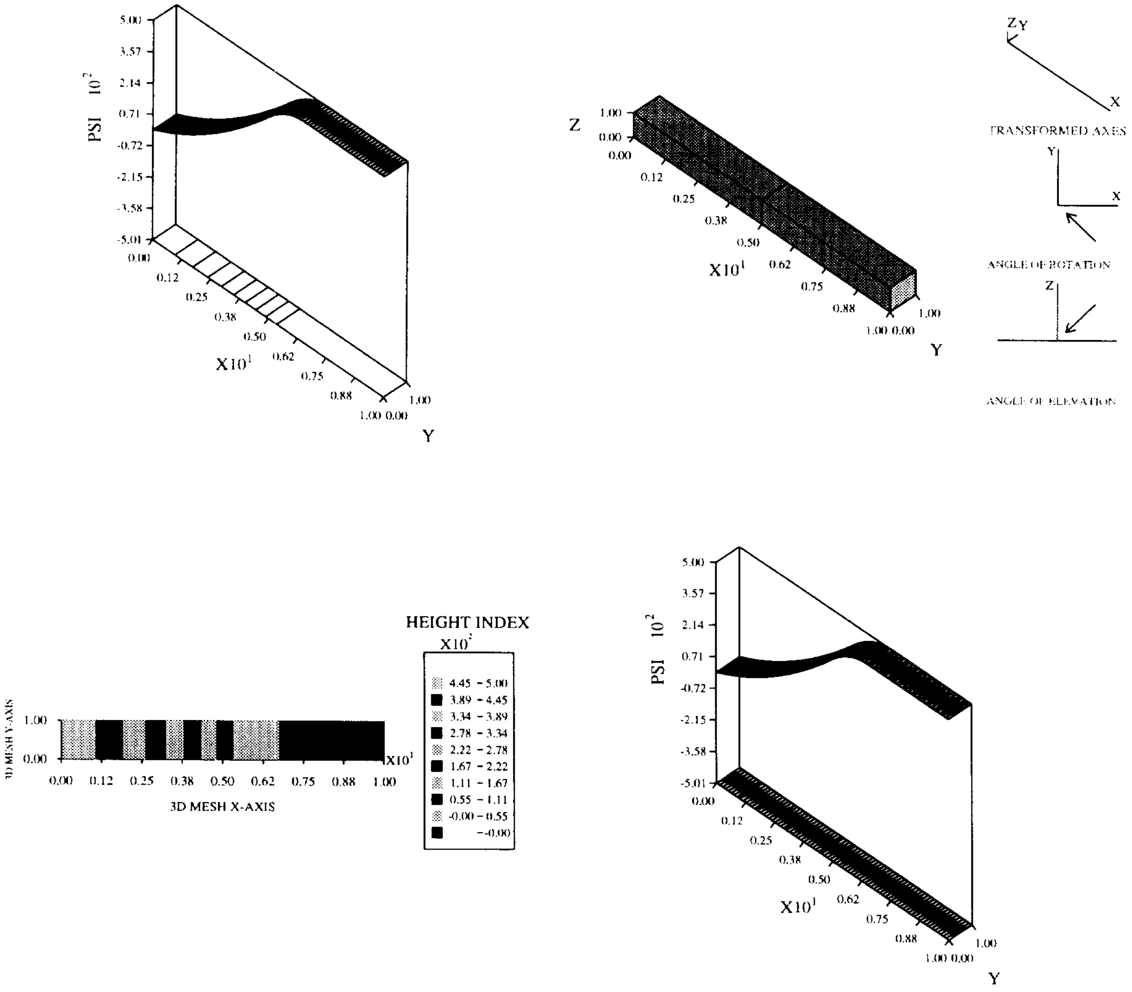


Figure 2: *Four plots of the geometry and potential within an effectively 1D diode.*

### 4.3 Diode containing three-dimensional effects

#### Description

The device consists of two hexahedral blocks of semiconductor of dimensions $1 \times 1 \times 0.5$ and $0.5 \times 0.5 \times 0.5 \mu m$. The larger is uniformly doped with impurity of acceptor type to $10^{15} cm^{-3}$ and the smaller is doped with impurities of donor type to $10^{15} cm^{-3}$. The contacts are at the two faces parallel to the interface.

The mesh contains $4 \times 4 \times 4$ divisions in the smaller block and $8 \times 8 \times 4$ in the larger one. This generates 505 nodes and 320 hexahedral elements. The device is simulated under two bias conditions: zero bias and $10V$ reverse bias.

The geometric description of the device is shown in Figure 3 and the electrostatic potential on the plane $y = 0$ at $10V$ bias is shown in a contour plot in Figure 4 and in an isometric plot in Figure 5. Curves of electrostatic potential plotted along lines of constant $x$ in this plane are shown in Figure 6.

#### Input to the Post-processor

```
INPUT blocks all
DEVICE x11
TITLE 'blocks example'
GEOMETRY
ROTATION 135
ELEVATION 40.0
COLOUR contacts yellow
COLOUR mesh black
PLOT mesh
PLOT triad
RETURN
CASE
SELECT blocks 2
PLANE y 0
VARIABLE psi
CONTOUR
CLEAR
COMMENT 'bias -10v ; plane y=0'
NCONTOURS 10 -10 0
PLOT boundary
PLOT line
PLOT comment
RETURN
ISOMETRIC
CLEAR
COLOUR axes red
MESH hidden
COMMENT 'isometric plot: bias 10v ; plane y=0'
PLOT axes
PLOT line
PLOT mesh
PLOT comment
RETURN
LINE
CLEAR
ADJUST 1.0 0.4
PLOT all
CUT x 0.5
COLOUR curve red
CUT x 0.25
COLOUR curve white
CLEAR
PLOT curves
RETURN
QUIT
```

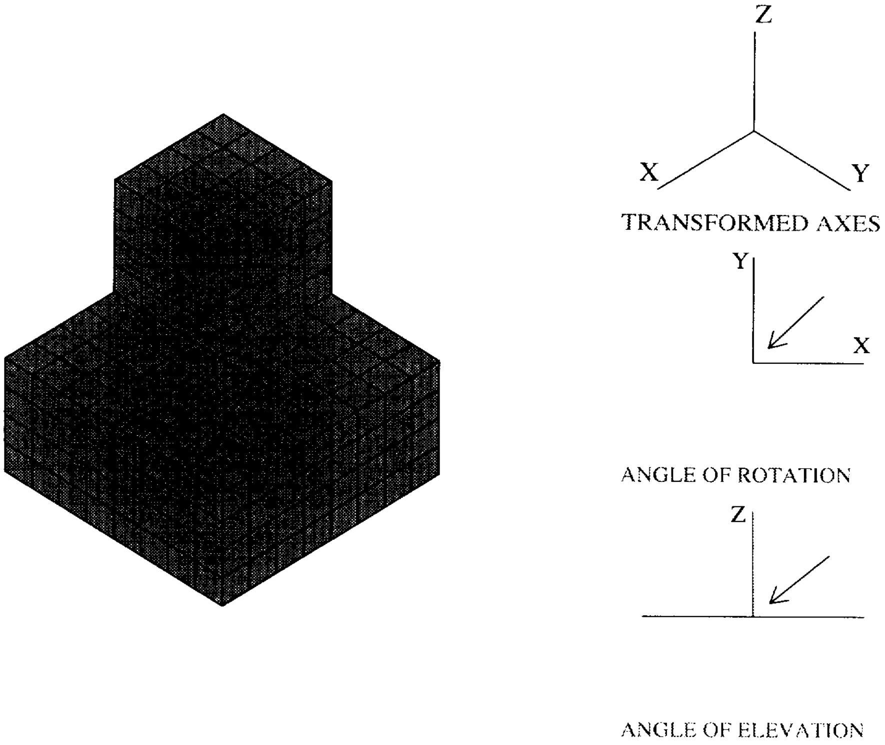


Figure 3: *Geometry of the blocks diode.*

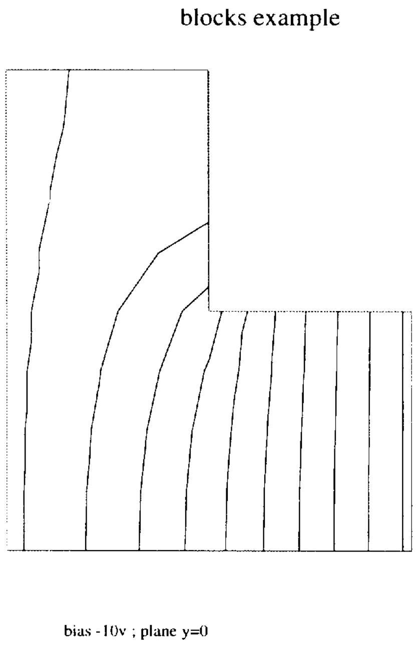


Figure 4: *Contours of potential on the $y = 0$ plane of the blocks diode.*

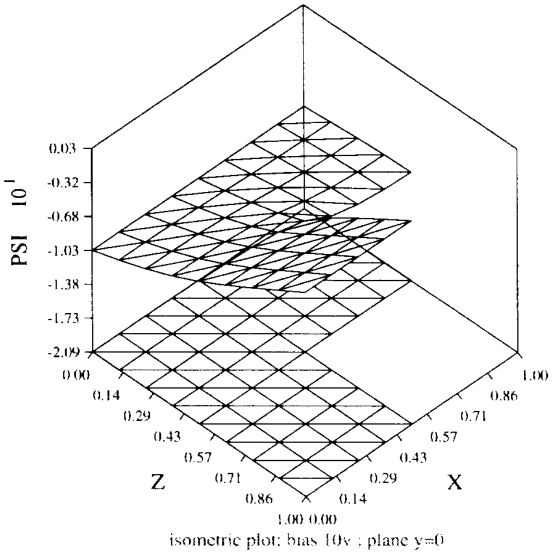


Figure 5: *Isometric plot of potential on the $y = 0$ plane of the blocks diode.*

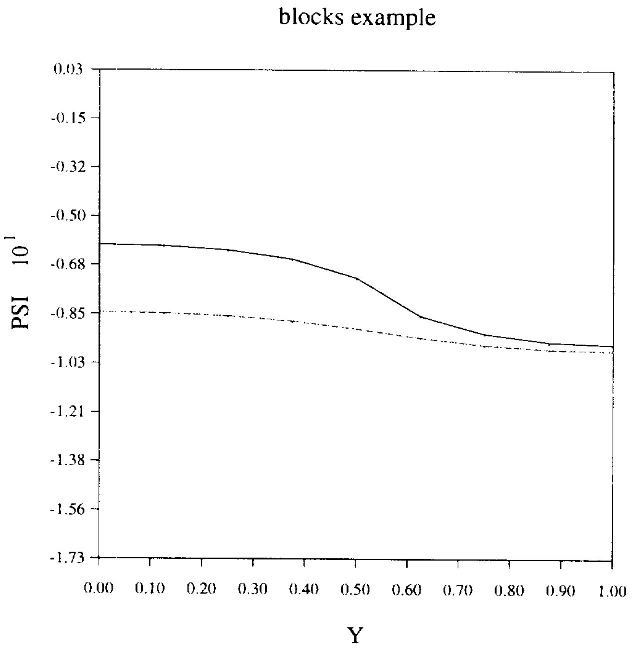


Figure 6: *Line plots of potential in the $y = 0$ cut plane of the blocks diode.*

## 4.4 Corner Diode with Oxide Overlay

### Description

The device consists of a block of semiconductor with an oxide overlay except for one corner, where a contact covering the whole corner and part of the top surface of the oxide is located. The other contact is located on the base of the semiconductor parallel to the oxide-semiconductor interface. The doping profile is generated by first implanting a uniform background doping of donor type at $10^{16}cm^{-3}$ and then Gaussian profiles of peak value $10^{18}cm^{-3}$ and $10^{17}cm^{-3}$ of acceptor type.

A mesh consisting of 476 nodes and 311 hexahedral elements was generated. The device is solved for 3 bias cases: zero bias, $20V$ and $50V$ on the base contact.

A selection of the typical results from this script is shown in Figure 7.

### Input to the Post-processor

```
INPUT corner
TITLE 'corner example 4.4 : isometric plots'
DEVICE
GEOMETRY
COLOUR edges magenta
PLOT line
PLOT axes
PLOT triad
RETURN
SELECT corner 1
```

```
PLANE y 0
VARIABLE netdop log
ISOMETRIC
CLEAR
COMMENT 'net doping on plane y=0'
CONTOUR 10 hidden
ROTATION 120.
PLOT mesh
PLOT axes
PLOT surface
PLOT contour
RETURN
VARIABLE psi
ISOMETRIC
COMMENT 'psi distribution - last plot'
CLEAR
PLOT axes
PLOT line
PLOT contour
PLOT comment
RETURN
LINE
CLEAR
PLOT all
CUT x 2.5
CUT x 5
CUT x 7.5
CUT x 11.
CLEAR
SHOW ltype
LTYPE 1
SHOW curves
SELECT 2
SHOW mtype
MTYPE 3
SELECT 1
MTYPE 4
LTYPE 3
CLEAR
PLOT curves
COMMENT 'Psi curves' 0.4 0.6
PLOT comment
RETURN
QUIT
```

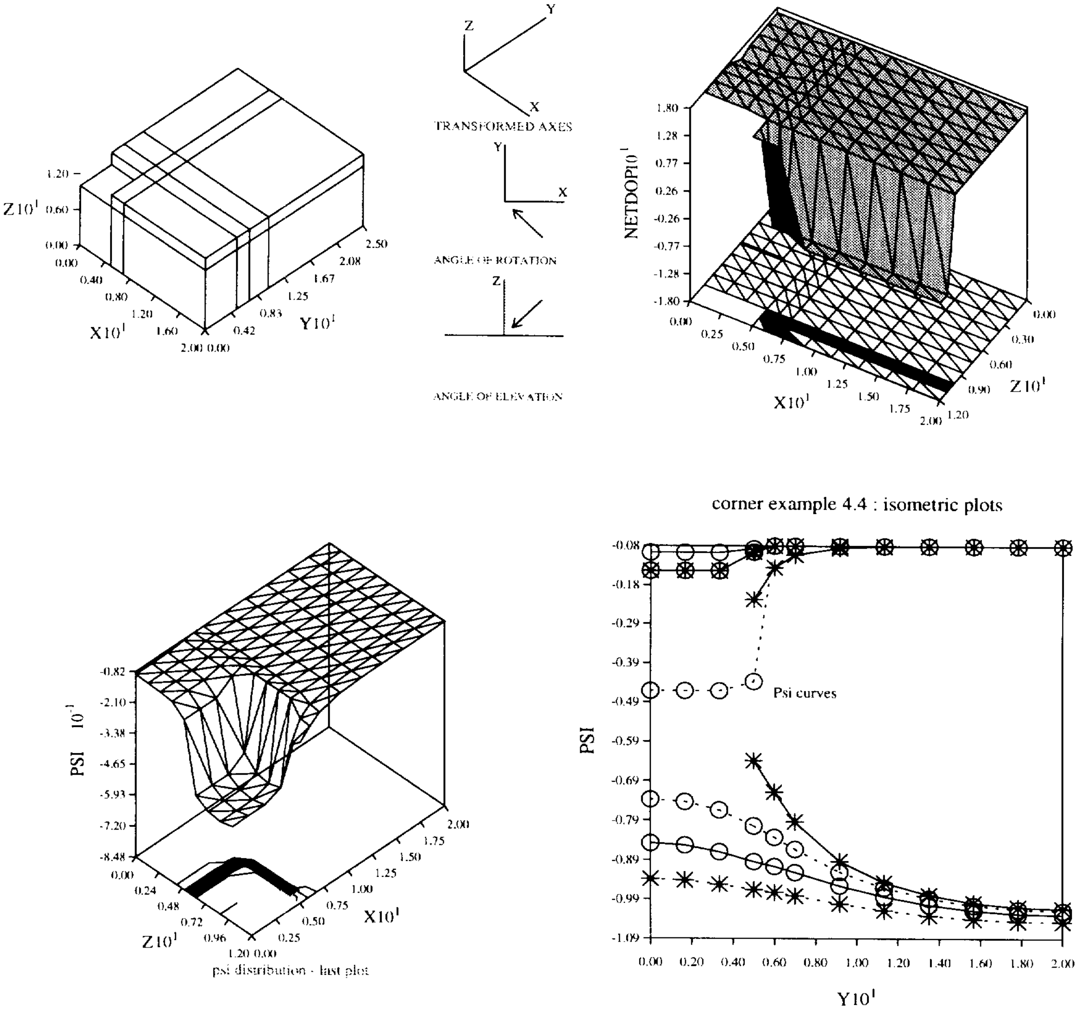


Figure 7: *Four plots of the corner diode results. Top left is geometry plot, top right is nett doping (isometric), bottom left is potential (isometric) and bottom right is potential (line plots).*

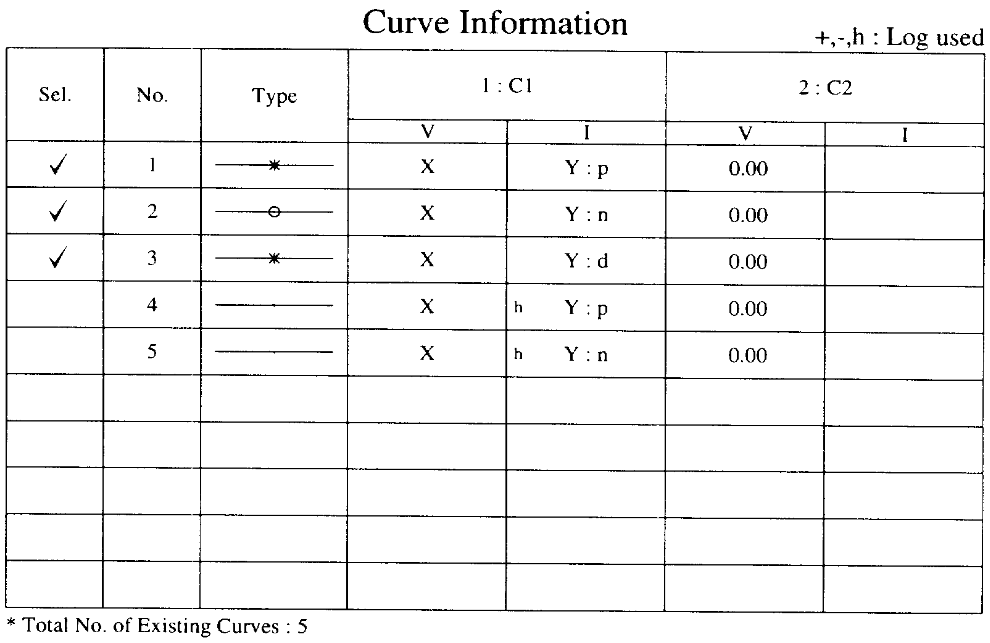
## 4.5 Diode I-V Characteristics

### Description

The device is a simple *p-n* junction used to illustrate diode characteristics. The only input required here is the results file PNCHAR.RES from a set of static simulations of a diode at differing bias points. In this case 9 static cases have been solved between $0.6V$ and $1.0V$ with the device in forward bias. Some typical output screens are shown in Figures 8 and 9. To fully understand the operation of these commands they should be used in the Post-processor.

### Input to the Post-processor

```
CHARACTER
READIN pnchar
DEFINE 1 contx=1 scalx=lin itemx=v conty=1 scaly=lin itemy=ip
DEFINE 1 itemy=in
DEFINE 1 itemy=id
DEFINE 1 itemy=ip scaly=log
DEFINE 1 itemy=in
SHOW curve
LTYPE 2 2
LTYPE 3 2
LTYPE 5 2
MTYPE 1 3
MTYPE 2 4
MTYPE 3 3
SELECT 1
SELECT 2
SELECT 3
CLEAR
PLOT
CLEAR
SHOW curve
SELECT -1
SELECT -2
SELECT -3
SELECT  4
SELECT  5
CLEAR
RANGE manual x 0.5 1.0
RANGE manual y -5 5
PLOT
RETURN
QUIT
```



Curve Information

+,-,h : Log used

| Sel. | No. | Type | 1 : C1 |  | 2 : C2 |  |
|---|---|---|---|---|---|---|
|  |  |  | V | I | V | I |
| √ | 1 | —*— | X | Y : p | 0.00 |  |
| √ | 2 | —o— | X | Y : n | 0.00 |  |
| √ | 3 | —*— | X | Y : d | 0.00 |  |
|  | 4 | — | X | h Y : p | 0.00 |  |
|  | 5 | — | X | h Y : n | 0.00 |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

* Total No. of Existing Curves : 5

Figure 8: *Output from the SHOW command in the characteristic level. The data shows that five curves that have been created at this point and the first three of them are selected for display. In this simulation the voltage only changes on contact C1 while C2 is fixed at* 0V.

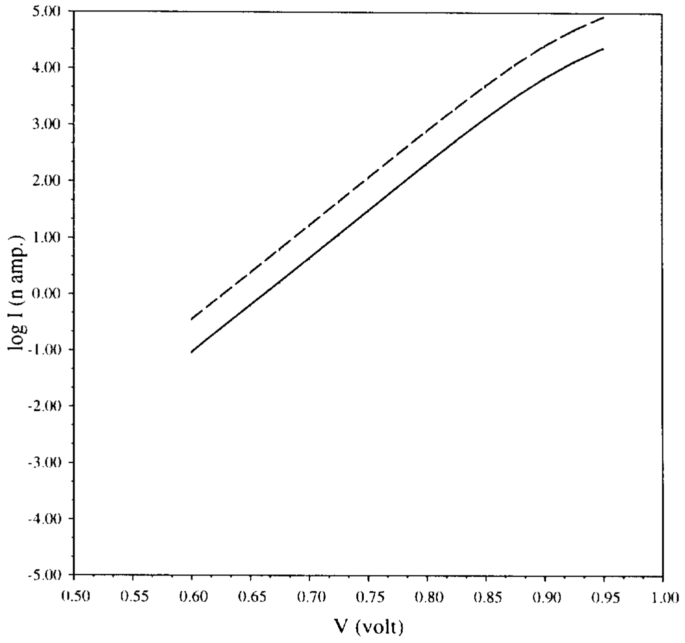


Figure 9: *A plot of the log of hole and electron currents against applied voltage produced by the characteristics command level.*

## 4.6 Diode Transient Characteristics

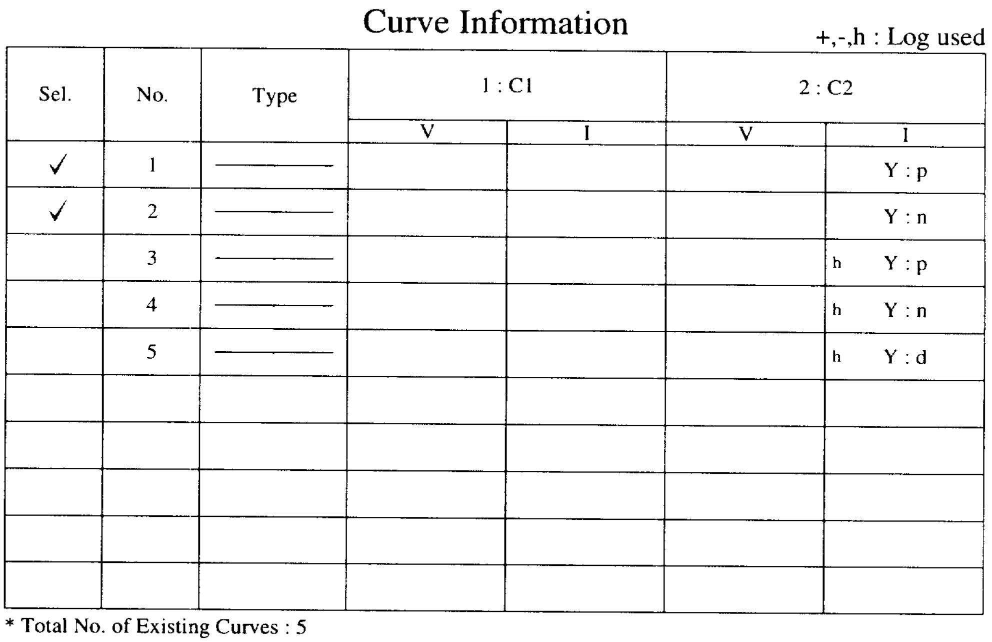
### Description

The device is a simple *p-n* junction used to illustrate the transient characteristics level. Only the file PNTRANS.RES from a transient simulation is required as input. Typical outputs are shown in Figures 10 and 11. To fully understand the function of these commands it is best to enter them into the post-processor and observe the results. The results shown refer to a diode of $10\mu m$ length and $1 \times 1\mu m$ cross section with uniform doping of $10^{18}cm^{-3}$ in either half of the device. Starting from a forward bias of $0.6V$ an additional $0.5V$ step is applied instantaneously at time zero.

### Input to the Post-processor

```
TRANSIENT
READIN pntran
SHOW summary
DEFINE 1 conty=2 itemy=ip scaly=lin
DEFINE itemy=in
DEFINE itemy=ip scaly=log
DEFINE itemy=in
DEFINE itemy=id
CLEAR
SHOW curve
CTYPE 1 blue
CTYPE 2 yellow
CTYPE 3 blue
CTYPE 4 magenta
CTYPE 5 cyan
SELECT 1
SELECT 2
CLEAR
PLOT
RANGE manual t 0  2.0e-8
RANGE manual i 0 0.6e-3
CLEAR
PLOT
PLOT grid
CLEAR
SHOW curve
SELECT -1
SELECT -2
SELECT  3
SELECT  4
SELECT  5
RANGE auto t
RANGE manual i -4 6
```

```
DIVIDE x 15
CLEAR
PLOT all
RETURN
QUIT
```



Curve Information

+,-,h : Log used

| Sel. | No. | Type | 1 : C1 |  | 2 : C2 |  |
|---|---|---|---|---|---|---|
|  |  |  | V | I | V | I |
| ✓ | 1 | ——— |  |  |  | Y : p |
| ✓ | 2 | ——— |  |  |  | Y : n |
|  | 3 | ——— |  |  |  | h Y : p |
|  | 4 | ——— |  |  |  | h Y : n |
|  | 5 | ——— |  |  |  | h Y : d |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

* Total No. of Existing Curves : 5

Figure 10: *Output from the SHOW CURVE command in the transient level. The data shows that five curves that have been created, two of which have been selected for display.*

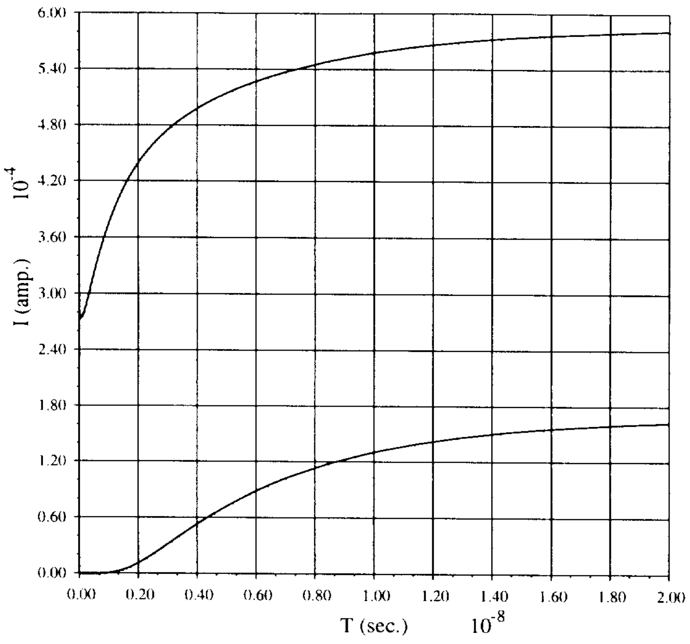


Figure 11: *A plot of the hole and electron currents against time produced by the transient command level.*

## References

[1] *R 5.2.2 (ESPRIT 962(E) Device Modelling Project) Graphics Implementation*, P.A.Mawby, University College Swansea (1987).

[2] *R 5.2.3 (ESPRIT 962(E) Device Modelling Project) Post Processor Implementation*, P.A.Mawby, J.Zhang and G.A.Duffett, University College Swansea (1987).

[3] *High-resolution Computer Graphics Using FORTRAN 77*, I.O.Angell and G.Griffith. Published by Macmillan, London (1987).

[4] *Computer Graphics and Applications*, D.Harris. Published by Chapman and Hall, London (1984).

# A Post-processor Main Commands

| | | |
|---|---|---|
| A.1 | CASE | - to list all the cases in the Results neutral file |
| A.2 | CHARACTER | - to enter characteristics command level |
| A.3 | CONTOUR | - to enter contour plotting command level |
| A.4 | DIFF | - to difference current 2D cut plane with saved data |
| A.5 | DEVICE | - to set the default GKS output device |
| A.6 | DIRECTORY | - to display the contents of the current directory |
| A.7 | GEOMETRY | - to enter geometry/mesh plotting command level |
| A.8 | INPUT | - to select, open and read a family of neutral files |
| A.9 | ISOMETRIC | - to enter isometric plotting command level |
| A.10 | LINE | - to enter line/curve plotting command level |
| A.11 | METAFILE | - to copy GKS output to a METAFILE |
| A.12 | PLANE | - to select a two-dimensional cut plane |
| A.13 | PAUSE | - to pause for a time |
| A.14 | QUIT | - to end program |
| A.15 | SELECT | - to select a case from the Results neutral file |
| A.16 | SAVE | - to save current 2D cut plane values |
| A.17 | TITLE | - to enter a title to be attached to all output |
| A.18 | TRANSIENT | - to enter transient command level |
| A.19 | VARIABLE | - to select a variable to be plotted |
| A.20 | VECTOR | - to enter vector plotting level |
| A.21 | UNZOOM | - to undo effect of ZOOM on the 2D cut plane |
| A.22 | WINDOW | - to split the screen into 1 or 4 plotting areas |
| A.23 | ZOOM | - to zoom in to a sub-region of the present cut plane |

## A.1 CASE - to list all the cases in the Results neutral file

### Syntax

```
CASE
```

### Description

Identifies all the cases in the currently selected neutral file. This is done by listing the $TEXT section of each $CASE block. A number is allocated to each case to allow any one to be chosen using the SELECT command.

### Parameters

No parameters

### Examples

```
CASE
```

## A.2 CHARACTER - to enter characteristic plotting command level

### Syntax

```
CHARacter
```

### Description

Passes control to the characteristic plotting command level.

### Parameters

No parameters

### Examples

```
CHARACTER
char
```

## A.3 CONTOUR - to enter contour plotting command level

### Syntax

```
CONTour
```

### Description

Passes control to the contour plotting command level.

### Parameters

No parameters

### Examples

```
CONTOUR
cont
```

## A.4 DIFF - to difference the current 2D cut plane with saved data

### Syntax

```
DIFF
```

### Description

Allows the difference in a variable between two solution cases to be calculated on a given cut plane. This command can only be used after a previous set of variable values on the chosen cut plane have been saved with the SAVE command.

### Parameters

No parameters.

### Examples

```
select p-n 1
plane z 0
var psi
save
select p-n 2
var psi
DIFF
```

## A.5 DEVICE - to set the default GKS output device

### Syntax

```
DEVice [WORKstation=<choice>]
```

### Description

Sets the type of graphics device being used to a particular workstation. The available devices depend on the version of GKS that is in use. As the system has been ported to several different versions of GKS, not all the devices listed will be available. The choices may be updated as new devices become available. The current default distribution includes X-GKS which supports devices X11 and CGM.

**Parameters**

| | |
|---|---|
| WORKSTATION | optional *choice* : initial = X11<br>Specifies the type of graphics device. Choices are (T4010, T4014, SUNMS, SUNML, SUNCS, SUNCL, SUN8BS, SUN8BL, PSP, PSL, PSPE, PSLE, X11, CGM). |

**Examples**

```
DEV
device x11
```

## A.6 DIRECTORY - to display the contents of the current directory

**Syntax**

```
DIRectory [EXTension=<choice>]
```

**Description**

Lists the names of all the families in the current directory. If the file name extension is specified, then only those files with that extension will be listed.

**Parameters**

| | |
|---|---|
| EXTENSION | optional *choice* : initial = ALL<br>A *choice* of filename extension restricting the class of names to be listed (ALL = all extensions). Choices are (ALL, GEO, MSH, DOP, PHY, RES). |

**Examples**

```
DIR
directory EXT=RES
dir msh
```

## A.7 GEOMETRY - to enter geometry/mesh plotting command level

**Syntax**

```
GEOMetry
```

### Description

Passes control to the geometry/mesh plotting command level.

### Parameters

No parameters

### Examples

```
GEOMETRY
geom
```

## A.8 INPUT - to select, open and read a family of neutral files

### Syntax

```
INPut FILE=<string> [EXTension=<choice>]
```

### Description

Selects, opens and reads a neutral file family. Files with the given file name extension in an existing family will be selected by this command.

### Parameters

| | |
|---|---|
| FILE | required *string*<br>This parameter names the neutral file to be opened. |
| EXTENSION | optional *choice* : initial = ALL<br>A *choice* of filename extension restricting the class of files to be selected (ALL = all extensions). Choices are (ALL, GEO, MSH, DOP, PHY, CAT). |

### Examples

```
INP BIPOLAR ALL
input refine msh
input refine dop
```

## A.9 ISOMETRIC - to enter isometric plotting command level

### Syntax

```
ISOmetric
```

### Description

Passes control to the isometric plotting command level.

### Parameters

No parameters

### Examples

```
ISO
isometric
```

## A.10 LINE - to enter line/curve plotting command level

### Syntax

```
LINE
```

### Description

Passes control to the line/curve plotting command level.

### Parameters

No parameters

### Examples

```
Line
LINE
```

## A.11 METAFILE - to copy GKS output to a METAFILE

### Syntax

```
METAfile [FILE=<string>] [STATus=<choice>]
```

### Description

Copies all GKS output to a metafile so that a hardcopy version can be obtained on another device (e.g. a penplotter). Not supported by some versions of GKS, including X-GKS.

**Parameters**

| | |
|---|---|
| FILE | optional *string* : initial = METFIL.DAT<br>A *choice* of filename for the metafile. |
| STATUS | optional *choice* : initial = OFF<br>Selects the status of the metafile. Choices are (ON, OFF, CLOSE). |

**Examples**

```
META
meta file=plott.met
META plott.met ON
```

## A.12 PAUSE - pause for a time, or until return

**Syntax**

```
PAUse [SECONDS=<integer>]
```

**Description**

Pause execution of program for the given number of seconds. If the number of seconds is less than zero, then the program waits until the return key is pressed. This command is for use in command scripts where it may be useful to wait between plots.

**Parameters**

| | |
|---|---|
| SECONDS | reset *integer* : initial = 1<br>The number of seconds to pause (if positive). |

**Examples**

```
PAUSE
pause 5
paus -1
```

## A.13 PLANE - to select a two-dimensional cut plane

**Syntax**

```
PLANE [AXES=<choice>] [POSition=<real>]
```

### Description

Defines the orientation and position of the cross-sectional plane to be used for plotting. Orientations are restricted to give planes normal to either the X-axis, the Y-axis or the Z-axis. The default case is the Z-axis.

### Parameters

| | |
|---|---|
| AXES | optional *choice* : initial = Z<br>This parameter identifies the orientation to be selected. It identifies to which of the axes the cross-sectional plane is normal. Allowed choices are (X, Y, Z). |
| POSITION | optional *real* : initial = 0.0<br>This parameter gives the value of the intercept of the cross-sectional plane with one of the axes. |

### Examples

```
PLANE X 10
plane AXES=Y,POS=2
plane z .10e-2
plane
```

## A.14 QUIT - to end program

### Syntax

```
QUIT
```

### Description

Terminates program execution.

### Parameters

No parameters

### Examples

```
QUIT
quit
```

## A.15 SAVE - to save data on current cut plane

### Syntax

```
SAVE
```

### Description

Saves the values at nodes on the current cut plane. Hence this command can only be used after a cut plane, and a variable on it, has been defined. The saved values can later be differenced with data from another bias case or transient case. Care should be taken to compare values on the same cut plane and for meaningful variables as only limited sanity checks are performed.

### Parameters

No parameters

### Examples

```
sel p-n 1
plane z 0
var psi
SAVE
```

## A.16 SELECT - to select a case from the Results neutral file

### Syntax

```
SELect FILE=<string> [CHOIce=<integer>]
```

### Description

The command causes a solution case to be read from the given results neutral file. Solution cases are labelled sequentially, and the CASE command can be used to list the available values and their corresponding bias levels.

### Parameters

| | |
|---|---|
| FILE | required *string*<br>This parameter names the neutral file to be used. No extension is required (the extension .RES is automatically used). |
| CHOICE | optional *integer* : initial = 1<br>This parameter identifies the case to be selected. |

### Examples

```
SEL mos 5
select FILE=mos 2
```

## A.17 TITLE - to enter a title to be attached to all output

### Syntax

```
TITle CAPtion=<string>
```

### Description

Defines text to be added to all output.

### Parameters

CAPTION

required *string*
This parameter is assigned the text to be added to output. Strings which include spaces must be enclosed in single quotes.

### Examples

```
TIT 'EVEREST 3-D device simulation program'
tit CAP=Bipolar
```

## A.18 TRANSIENT - to enter transient plotting command level

### Syntax

```
TRANSient
```

### Description

Passes control to the transient plotting command level.

### Parameters

No parameters

### Examples

```
TRANS
transient
```

## A.19 VARIABLE - to select a variable to be plotted

### Syntax

```
VARiable [DIStribution=<choice>] [SCAle=<choice>]
 [LIMIT=<real>]
```

### Description

Causes the chosen variable to be evaluated on the current cut plane. This will fail if no cut plane has been defined or not all required neutral files have been read. Note that the LOG option will take the log of the absolute value.

### Parameters

DISTRIBUTION — optional *choice* : initial = PSI
This parameter selects the variable to be viewed. Allowed choices are (PSI, PHIN, PHIP, NA, ND, NETDOP, P, N, NETCHG, NIE, REC, HMOB, EMOB, JP, JN, JTOT, E).

SCALE — optional *choice* : initial = LIN
This parameter selects the plotting scale. Choices are (LIN, LOG).

LIMIT — optional *real* : initial = 0.0
This parameter specifies the lower limit
of the range of values plotted on a log scale

### Examples

```
VAR PHIN
var na log
variable psi log 0.0
```

## A.20 VECTOR - to enter vector plotting command level

### Syntax

```
VECTor
```

### Description

Passes control to the vector plotting command level.

### Parameters

No parameters

### Examples

```
VECTOR
vect
```

## A.21 UNZOOM - to undo the effect of the ZOOM command

### Syntax

```
UNZOom
```

### Description

This restores the full 2D cut plane after a ZOOM command has been used.

### Parameters

No parameters

### Examples

```
unzoom
```

## A.22 WINDOW - to split screen into 1 or 4 plotting areas

### Syntax

```
WINDow [NUMber=<choice>] [PRESent=<choice>]
```

### Description

Divides the screen into 1 or 4 areas for plotting.

### Parameters

| | |
|---|---|
| NUMBER | optional *choice* : initial = 1<br>This parameter selects either 1 or 4 plotting areas. Choices are (1,4). |
| PRESENT | optional *choice* : initial = 1<br>This parameter selects the area for the subsequent plot. Choices are (1,2,3,4). |

### Examples

```
WINDOW
wind 4 2
wind 1
```

## A.23 ZOOM - to zoom into a sub-region of the cut plane

### Syntax

```
ZOom
```

### Description

Once a cut plane has been defined, it is possible to select a smaller subregion of it to be displayed using this command. A rectangular region of interest is given by a set of four points $x_{min}$, $x_{max}$, $y_{min}$, $y_{max}$. Any elements in the 2D cut plane that have no node in the specified region will not be displayed in subsequent plots. Use the UNZOOM command to undo a ZOOM.

### Parameters

REGION reset *real_list*
Set of four points giving the xmin, xmax, ymin, ymax of the zoom region.

### Examples

```
ZOOM (0 4 0 4)
zoo (22 23 5 7)
```

# B Contour Plotting Commands

## B.1 CLEAR - to clear screen

### Syntax

```
CLear
```

### Description

Clears the screen.

### Parameters

No parameters

### Examples

```
CL
clear
```

## B.2 COMMENT - to define a comment for a contour plot

### Syntax

```
COMMent [CAPtion=<string>]
```

### Description

Defines additional text to be added to the output.

### Parameters

CAPTION optional *string* : initial = ' '
This parameter is assigned the text to be added to output. Strings which include spaces must be enclosed in single quotes.

### Examples

```
COMM 'SYMMETRIC DIODE'
comment CAP=12/12/64
```

## B.3 LCONTOURS - to specify a list of contour values

### Syntax

```
LCONTours LEVels=<rlist>
```

### Description

Specifies which values of the variable to draw contours at. These may be entered in any numerical order as a real list. This is an alternative to NCONTOURS which gives uniformly spaced contour levels.

### Parameters

| | |
|---|---|
| LEVELS | required *real-list*<br>A list of values at which the contours are to be plotted. |

### Examples

```
LCONT (1.0,2.0,3.0,4.0,5.0)
lcontours LEV=(1.0,3.1,0.5)
```

## B.4 NCONTOURS - to specify number and limits of contours

### Syntax

```
NCONTours [NLEVels=<integer>] [FIRST=<real>] [LAST=<real>]
```

### Description

Specifies a given number of contours with values uniformly spaced between the specified limits. If the limits are the same (or unspecified) then the limits will be set to the minimum and maximum values of the variable in the cut plane.

### Parameters

| | |
|---|---|
| NLEVELS | optional *integer* : initial = 10<br>Specifies the number of contours to be plotted. |
| FIRST | optional *real* : initial = 0.0<br>Specifies value of the first contour. |
| LAST | optional *real* : initial = 0.0<br>Specifies value of the last contour. |

### Examples

```
ncont NLEV=3 1 LAST=5.6    - specifies contours 1.0,3.3,5.6
NCONT 5 1.0 5.0            - specifies contours 1,2,3,4,5.
NCONTOURS 10               - 10 contours
```

## B.5 PLOT - to plot output items

### Syntax

```
PLOT [ITem=<choice>]
```

### Description

Plots the requested items on the current output device. PLOT ALL defaults to using solid contour fills.

### Parameters

| | |
|---|---|
| ITEM | optional *choice* : initial = ALL<br>This parameter selects the items required to be plotted. The choices are (ALL, LINE, SOLID, MESH, BOUNDARY, AXES, COMMENT, KEY). |

### Examples

```
PLOT
plot line
plot solid
plot comm
```

## B.6 RETURN - to return to main command level

### Syntax

```
RETurn
```

### Description

Returns control to the main command level.

### Parameters

No parameters

### Examples

```
RET
return
```

# C Isometric Plotting Commands

| | | |
|---|---|---|
| C.1 | ADJUST | - to adjust the height and gap of surface in [0,1] |
| C.2 | AXES | - to specify axes parameters |
| C.3 | CLEAR | - to clear screen |
| C.4 | COLOUR | - to specify colour of output items |
| C.5 | COMMENT | - to define a comment for an isometric plot |
| C.6 | CONTOUR | - to specify level and type of contour plot |
| C.7 | ELEVATION | - to specify the elevation angle of view |
| C.8 | MESH | - to specify type of mesh plot |
| C.9 | PLOT | - to plot output items |
| C.10 | RETURN | - to return to main command level |
| C.11 | ROTATION | - to specify the viewing direction |

## C.1 ADJUST - to adjust the height and gap of the surface in [0,1].

### Syntax

```
ADJust [HEIGHT=<real>] [GAP=<real>]
```

### Description

Adjusts the height of the surface and also the gap between the bottom of the surface and the base. The height limits are defined as [0,1].

### Parameters

HEIGHT

optional *real* : initial = 1.0
Specifies the height of the surface.

GAP

optional *real* : initial = 0.5
Specifies the gap between the surface and the base.

### Examples

```
adjust 0.75
ADJ height=0.75 gap=0.3
adj 0.6 0.4
```

## C.2 AXES - to set axes divisions

### Syntax

```
AXES [DIVX=<integer>] [DIVY=<integer>] [DIVZ=<integer>]
```

### Description

Specifies the number of divisions marked on each axis when the axes are plotted.

### Parameters

DIVX

optional *integer* : initial = 8
Specifies number of divisions on the X axis.

DIVY

optional *integer* : initial = 8
Specifies number of divisions on the Y axis.

DIVZ

optional *integer* : initial = 8
Specifies number of divisions on the Z axis.

### Examples

```
AXES
axes DIVX=12 DIVY=8 16
axes 12 8 16
```

## C.3 CLEAR - to clear screen

### Syntax

```
CLear
```

### Description

Clears the screen.

### Parameters

No parameters

### Examples

```
CL
clear
```

## C.4 COLOUR - to specify colour of output items

### Syntax

```
COLour [ITem=<choice>] [ITCOLour=<choice>]
```

### Description

Selects the colours for the output items.

### Parameters

ITEM

optional *choice* : initial = LINE
Specifies the item for which this colour is selected. Allowable choices are (LINE, TOP, BOTTOM, MESH, CONTOUR, AXES, BACKGROUND, TITLES).

ITCOLOUR

optional *choice* : initial = RED
Specifies the colour of the chosen ITEM. Allowable choices are (BLACK, GREEN, CYAN, BLUE, MAGENTA, RED, YELLOW, WHITE).

### Examples

```
COL lines
colour TOP red
col mesh blue
```

## C.5 COMMENT - to define a comment for an isometric plot

### Syntax

```
COMMent [CAPtion=<string>]
```

### Description

Defines additional text to be added to the output.

### Parameters

| | |
|---|---|
| CAPTION | optional *string* : initial = ' '<br>This parameter is assigned the text to be added to output. Strings which include spaces must be enclosed in single quotes. |

### Examples

```
COMM 'SYMMETRIC DIODE'
comment CAP=12/12/64
```

## C.6 CONTOUR - to specify the level and the type of contour plot

### Syntax

```
CONTour [LEVEL=<integer>] [TYPE=<choice>]
```

### Description

Used when the contours are to be plotted underneath the surface. Hidden line removal may be used to ensure that the surface is not over-plotted. The CONTOUR command is used to specify this as well as to specify the number of contours.

### Parameters

| | |
|---|---|
| LEVEL | optional *integer* : initial = 10<br>Specifies the number of contours to draw. |
| TYPE | optional *choice* : initial = HIDDEN<br>Specifies the type of contour plot. Choices are (HIDDEN, NORMAL). |

### Examples

```
cont LEVEL=8 TYPE=NORMAL
CONTOUR
contour 8 normal
```

## C.7 ELEVATION - to specify the elevation angle of view

### Syntax

```
ELEVation [ANGle=<real>]
```

### Description

Sets the elevation angle of view for the isometric plot.

### Parameters

| | |
|---|---|
| ANGLE | optional *real* : initial = 45.0<br>Sets the elevation angle of view in degrees. |

### Examples

```
ELEV 0.0
elevation
elev 123
```

## C.8 MESH - to specify type of mesh plot

### Syntax

```
MESH [TYPE=<choice>]
```

### Description

Required when the two-dimensional cut-plane mesh is to be plotted underneath the surface. Hidden line removal may be used to ensure that the surface is not over-plotted. The MESH command is used to specify this.

### Parameters

| | |
|---|---|
| TYPE | optional *choice* : initial = HIDDEN<br>Specifies type of plotted mesh. Allowable choices are (HIDDEN, NORMAL). |

### Examples

```
MESH
mesh normal
```

## C.9 PLOT - to plot output items

### Syntax

```
PLOT [ITem=<choice>]
```

### Description

To plot the required items on the current output device.

### Parameters

| | |
|---|---|
| ITEM | optional *choice* : initial = ALL<br>This parameter specifies the item to be plotted. The choices are (ALL, LINE, SURFACE, MESH, CONTOUR, AXES, COMMENT). |

### Examples

```
PLOT
plot mesh
plot CONTOUR
```

## C.10 RETURN - to return to main command level

### Syntax

```
RETurn
```

### Description

Returns control to the main command level.

### Parameters

No parameters

### Examples

```
RET
return
```

## C.11 ROTATION - to specify the viewing direction

### Syntax

```
ROTation [ANGle=<real>]
```

### Description

Sets the azimuthal viewing direction for the isometric plot.

### Parameters

| | |
|---|---|
| ANGLE | optional *real* : initial = 45.0<br>This parameter sets the viewing angle in degrees. |

### Examples

```
rotation 45
ROT 267.
```

# D Geometry/Mesh Plotting Commands

| | | |
|---|---|---|
| D.1 | AXES | - to specify axes parameters |
| D.2 | CLEAR | - to clear screen |
| D.3 | COLOUR | - to specify colour of output items |
| D.4 | COMMENT | - to define a comment for a geometry/mesh plot |
| D.5 | ELEVATION | - to specify the elevation angle of view |
| D.6 | ORIGIN | - to mark the origin onto geometry/mesh plot |
| D.7 | PLOT | - to plot various geometry/mesh plots |
| D.8 | RETURN | - to return to main command level |
| D.9 | ROTATION | - to specify the viewing direction |

## D.1 AXES - to set axes divisions

### Syntax

```
AXES [DIVX=<integer>] [DIVY=<integer>] [DIVZ=<integer>]
```

### Description

Specifies the number of divisions marked on each axis when the axes are plotted.

### Parameters

| | |
|---|---|
| DIVX | optional *integer* : initial = 8<br>Specifies number of divisions on the X axis. |
| DIVY | optional *integer* : initial = 8<br>Specifies number of divisions on the Y axis. |
| DIVZ | optional *integer* : initial = 8<br>Specifies number of divisions on the Z axis. |

### Examples

```
AXES
axes DIVX=10 DIVY=8 12
axes 10 8 12
```

## D.2 CLEAR - to clear screen

### Syntax

```
CLear
```

### Description

Clears the screen.

### Parameters

No parameters

### Examples

```
CL
clear
```

## D.3 COLOUR - to specify colour of output items

### Syntax

```
COLour [ITem=<choice>] [ITCOLour=<choice>]
```

### Description

Changes the default colour for an item in the plot.

### Parameters

| | |
|---|---|
| ITEM | optional *choice* : initial = TITLE<br>Specifies the item for which this colour is selected. The choices are (TITLE, BACKGROUND, MESH, EDGES, CONTACTS, SILICON, OXIDE, AXES). |
| ITCOLOUR | optional *choice* : initial = RED<br>Specifies the colour of the chosen ITEM. The choices are (BLACK, GREEN, CYAN, BLUE, MAGENTA, RED, YELLOW, WHITE). |

### Examples

```
COL contacts
colour
col mesh blue
```

## D.4 COMMENT - to define a comment for a geometry/mesh plot

### Syntax

```
COMMent [CAPtion=<string>]
```

### Description

Defines additional text to be added to the output.

### Parameters

| | |
|---|---|
| CAPTION | optional *string* : initial = ' '<br>This parameter is assigned the text to be added to output. Strings which include spaces must be enclosed in single quotes. |

### Examples

```
COMM 'Three dimensional device plot'
comment CAP=12/12/64
```

## D.5 ELEVATION - to specify the elevation angle of view

### Syntax

```
ELEVation [ANGle=<real>]
```

### Description

Sets the elevation angle of view for the geometry/mesh plot.

### Parameters

| | |
|---|---|
| ANGLE | optional *real* : initial = 45.0<br>Sets the elevation angle of view in degrees. |

### Examples

```
ELEV 0.0
elevation
elev 123
ELEVATION ANGLE=-48.
```

## D.6 ORIGIN - to mark the origin onto the geometry/mesh plot

### Syntax

```
ORIGin
```

### Description

Causes the origin to be marked onto the geometry/mesh plot. This is done only if the origin can be seen.

### Parameters

No parameters

### Examples

```
ORIG
origin
```

## D.7 PLOT - to plot various geometry/mesh plots

### Syntax

```
PLOT [ITem=<choice>]
```

### Description

Draws the requested items on the current output device.

### Parameters

| | |
|---|---|
| ITEM | optional *choice* : initial = SURFACES<br>This parameter selects the type of plot required. The choices are (ALL, LINE, SOLID, MESH, AXES, COMMENT, TRIAD). |

### Examples

```
PLOT
plot mesh
plot LINE
plot TRIAD
```

## D.8 RETURN - to return to main command level

### Syntax

```
RETurn
```

### Description

Returns control to the main command level.

### Parameters

No parameters.

### Examples

```
RET
return
```

## D.9 ROTATION - to specify the viewing direction

### Syntax

```
ROTation [ANGle=<real>]
```

### Description

Sets the azimuthal viewing direction for the geometry/mesh plot.

**Parameters**

| | |
|---|---|
| ANGLE | optional *real* : initial = 45.0<br>This parameter sets the viewing angle in degrees. |

**Examples**

```
rotation 45
ROT 267.
rot -39
```

# E Line Plotting Commands

| | | |
|---|---|---|
| E.1 | ADJUST | - to adjust the height and gap of curve |
| E.2 | CLEAR | - to clear screen |
| E.3 | COLOUR | - to specify colour of output items |
| E.4 | COMMENT | - to define a comment to add to the plot |
| E.5 | CUT | - to create a curve from a line in the 2D plane |
| E.6 | DELETE | - to delete an existing curve |
| E.7 | ERASE | - to clear the curve array |
| E.8 | LTYPE | - to set the line type |
| E.9 | MTYPE | - to set the marker type |
| E.10 | PLOT | - to plot output items |
| E.11 | ROTATION | - to rotate the 2D mesh |
| E.12 | RETURN | - to return to main command level |
| E.13 | SELECT | - to select the current curve |
| E.14 | SHOW | - to display information about items |

## E.1 ADJUST - to adjust the height and gap of the surface

### Syntax

```
ADJust [HEIGHT=<real>] [GAP=<real>]
```

### Description

Adjusts the height of the curve and also the gap between the bottom of the curve and the base. The height limits are defined as [0,1].

### Parameters

HEIGHT — optional *real* : initial = 1.0
Specifies the top of the surface on the screen.

GAP — optional *real* : initial = 0.0
Specifies the gap between the bottom of the surface and the screen base.

### Examples

```
adjust 0.70
ADJ height=0.70 gap=0.35
adj 0.6 0.4
```

## E.2 CLEAR - to clear screen

### Syntax

```
CLear
```

### Description

Clears the screen.

### Parameters

No parameters.

### Examples

```
CL
clear
```

## E.3 COLOUR - to specify colour of output items

### Syntax

```
COLour [ITem=<choice>] [ITCOLour=<choice>]
```

### Description

Changes the default colour for a selected item in the plot.

### Parameters

ITEM

optional *choice* : initial = CURVE
Specifies the item for which this colour is selected. The choices are (MESH, RULER, LINE, CURVE, AXES, TITLE, BACKGROUND).

ITCOLOUR

optional *choice* : initial = GREEN
Specifies the colour of the chosen ITEM. The choices are (BLACK, GREEN, CYAN, BLUE, MAGENTA, RED, YELLOW, WHITE).

### Examples

```
COL curve
colour RULER red
col title blue
```

## E.4 COMMENT - to define a comment to add to the plot

### Syntax

```
COMMent [CAPtion=<string>] [XPOS=<real>] [YPOS=<real>]
```

### Description

Defines additional text to be added to the output. The comment begins at the specified coordinates (XPOS, YPOS). The plot region is [0,1] in $x$ and $y$.

### Parameters

CAPTION — optional *string* : initial = ' '
This parameter is assigned the text to be added to output. Strings which include spaces must be enclosed in single quotes.

XPOS — optional *real* : initial = 0.5
This parameter specifies the X co-ordinate of the comment beginning.

YPOS — optional *real* : initial = 0.0
This parameter specifies the Y co-ordinate of the comment beginning.

### Examples

```
COMM 'Curve number 1'
comm 'I-V curve' 0.75 0.25
comment CAP=12/12/64 xpos=0.75 ypos=0.25
```

## E.5 CUT - to create a curve from a line in the 2D plane

### Syntax

```
CUT [ITem=<choice>] [CONSTant=<real>]
```

### Description

Creates a 1D graph from the 2D cut plane data. The cut line must be parallel with either the current $x$ or $y$ axes, but the 2D cut plane may have previously been rotated by the ROTATION command. If the mesh is currently displayed, the cut line will be drawn on it. Note that curves are retained, even after a return to the POST command level, until the ERASE or DELETE commands are used.

### Parameters

ITEM — optional *choice* : initial = X
Specifies the axis perpendicular to the cutting line. The choices are (X,Y).

CONSTANT — required *real*
Specifies the position of the cut line.

### Examples

```
cut x 0
cut item=Y const=0.5
```

## E.6 DELETE - to delete an existing curve

### Syntax

```
DELete [ITem=<integer>]
```

### Description

Deletes an existing curve from the list of curves currently stored. Remaining curves are renumbered if necessary.

### Parameters

| | |
|---|---|
| ITEM | optional *integer* : initial = 0<br>The number of the curve to be deleted. |

### Examples

```
delete 2
del item=3
```

## E.7 ERASE - to clear the curve array

### Syntax

```
ERase
```

### Description

Erases all curves currently stored.

### Parameters

No parameters.

### Examples

```
erase
ER
```

## E.8 LTYPE - to set the line type

### Syntax

```
LType [ITem=<integer>]
```

### Description

Selects the style of the line to be plotted for the most recently created or selected curve. Type 1 is solid, other are broken lines.

### Parameters

| | |
|---|---|
| ITEM | optional *integer* : initial = 1<br>The style of the line to be plotted. The options are (1, 2, 3, 4). |

### Examples

```
LTYPE
LT item=3
lt 4
```

## E.9 MTYPE - to set the marker type

### Syntax

```
MType [ITem=<integer>]
```

### Description

Selects the marker type to be used for the most recently created or selected curve. There are unresolved problems with GKS polymarkers which means this command does not work for some output devices.

### Parameters

| | |
|---|---|
| ITEM | optional *integer* : initial = 1<br>Determines the style of the marker to use. The options are (1, 2, 3, 4, 5). |

### Examples

```
MTYPE
MT item=1
mt 2
```

## E.10 PLOT - to plot output items

### Syntax

```
PLOT [ITem=<choice>]
```

### Description

Plot the required items on the current output device. Using the parameter ALL does not cause any curves to be plotted; this is used to plot the two-dimensional mesh so that the lines along which the variables are plotted can be determined. The curve plots are only plotted using the CURVE or CURVES parameters.

### Parameters

ITEM

optional *choice* : initial = ALL
This parameter specifies the item to be plotted. The choices are (ALL, MESH, RULER, CURVE, CURVES, AXES, COMMENT).

### Examples

```
PLOT
plot mesh
plot CURVES
PLOT comment
```

## E.11 RETURN - to return to main command level

### Syntax

```
RETurn
```

### Description

Returns control to the main command level.

### Parameters

No parameters.

### Examples

```
RET
return
```

## E.12 ROTATION - to rotate the 2D mesh

### Syntax

```
ROTation [ANGle=<real>]
```

### Description

Rotates the two-dimensional cut plane mesh. This allows curves to be defined along arbitrary cut lines.

### Parameters

| | |
|---|---|
| ANGLE | optional *real* : initial = 0.0<br>This parameter sets the rotation angle in degrees. |

### Examples

```
rotation 15
ROT 245.
```

## E.13 SELECT - to select the current curve

### Syntax

```
SELect [ITem=<integer>]
```

### Description

Makes the chosen curve the current (last) curve. The curve is relabelled as the last curve in the list. After SELECTion the curve can be displayed by PLOT CURVE and the line type, marker type and colour altered by LTYPE, MTYPE and COLOUR commands. Note that other curves may be renumbered after this command - use SHOW CURVES to see the new numbering.

### Parameters

| | |
|---|---|
| ITEM | optional *integer* : initial = 0<br>Determines the current curve number which is to be plotted. |

### Examples

```
select
sel item=1
```

## E.14 SHOW - to display information about items

### Syntax

```
SHow [ITem=<choice>]
```

### Description

Displays information about the requested items. A list of all stored curves with miniature plots (CURVES) can be obtained as well as information about available types of lines and markers and colours. SHOW GRID can be used to add a grid on top of the graph.

### Parameters

| | |
|---|---|
| ITEM | optional *choice* : initial = 'CURVES'<br>The item about which information is required. The options are (CURVES, LTYPE, MTYPE, COLOUR, GRID). |

### Examples

```
SHOW curves
SH item=mtype
sh ltype
```

# F Vector Commands

| | | |
|---|---|---|
| F.1 | CLEAR | - to clear the screen |
| F.2 | COLOUR | - to set the colour of items |
| F.3 | COMMENT | - to specify a comment on the screen |
| F.4 | NVECTOR | - to set the number of vectors in each direction |
| F.5 | PLOT | - to plot items |
| F.6 | RETURN | - to return to main command level |
| F.7 | SVECTOR | - to select vector quantity |
| F.8 | TVECTOR | - to specify vector plot type |

## F.1 CLEAR - to clear the screen

### Syntax

```
CLear
```

### Description

Clears the screen

### Parameters

No parameters

### Examples

```
CL
clear
```

## F.2 COLOUR - to set colour to the required items

### Syntax

```
COLour [ITEM=choice>] [COLour=<choice>]
```

### Description

Set colour of the required items.

### Parameters

| | |
|---|---|
| ITEM | optional *choice* : initial=KEY<br>Specifies the item to which colour is to be set<br>Choices = (KEY,UP,DOWN,AXES,MESH, BACKGROUND,TITLES) |
| COLour | optional *choice* : initial=RED<br>Specifies the colour for the chosen ITEM<br>Choices = (BLACK,GREEN,CYAN,BLUE, MAGENTA,RED,YELLOW,WHITE) |

### Examples

```
COL axes
colour key red
```

## F.3 COMMENT - specify a comment on the screen

### Syntax

```
COMment [CAPtion=<string>]
```

### Description

Specify a comment on the screen

### Parameters

| | |
|---|---|
| CAPtion | optional *string* : initial=' '<br>Specifies the text<br>Strings which include space must be enclosed in single quotes. |

### Examples

```
COM 'Vector Plot'
comment CAP=1/4/90
```

## F.4 NVECTOR - to set the number of vectors in each direction

### Syntax

```
NVECtor [ITEM=<integer>]
```

### Description

Specify the number of vectors in each direction. A uniform square array of vectors is generated so cut planes which do not have a 1:1 aspect ratio will have fewer vectors along the shorter axis.

### Parameters

| | |
|---|---|
| ITEM | optional *integer* : initial=10<br>specifies no. of vectors in each direction. |

### Examples

```
NVEC  10
nvector  10
```

## F.5 PLOT - to plot the required items

### Syntax

```
PLOT [ITEM=<choice>]
```

### Description

Plot the required items. Note that the vector level uses the same scaling as the CONTOUR level. Thus it is possible to add vectors to an existing contour plot using PLOT VECTOR.

### Parameters

| | |
|---|---|
| ITEM | optional *choice* : initial=ALL<br>specifies the item to output<br>Values = (ALL,VECTOR,AXES,<br>MESH,COMMENT) |

### Examples

```
PLOT all
plot ITEM=vector
```

## F.6 RETURN - to return to main command level

### Syntax

```
RETurn
```

### Description

Return to main command level

### Parameters

No parameters

### Examples

```
RET
return
```

## F.7 SVECTOR - to select vector quantity

### Syntax

```
SVECtor [ITEM=<choice>]
```

### Description

Selects the vector quantity to be plotted. This can be the hole or electron current (JP, JN), the total current (JTOT) or the electric field (E).

### Parameters

ITEM — optional *choice* : initial=E
specifies the item to output
Values = (JP,JN,E,JTOT)

### Examples

```
SVECT JP
plot ITEM=jtot
```

## F.8 TVECTOR - to specify vector plot type

### Syntax

```
TVECtor [ITEM=<choice>]
```

### Description

Specify vector plot type. Vectors can be plotted in full or as lines without the vector head.

### Parameters

ITEM — optional *choice* : initial=FULL
specifies the item to output
Values = (LINE,FULL,KEY)

### Examples

```
TVECT FULL
tvector ITEM=key
```

# G I-V Characteristic Commands

| | | |
|---|---|---|
| G.01 | CLEAR | - to clear the screen |
| G.02 | COLOUR | - to set the colour of items |
| G.03 | COMMENT | - to define a comment on the plot |
| G.04 | CTYPE | - to set the colour of a curve |
| G.05 | DEFINE | - to define a curve |
| G.06 | DELETE | - to delete all the selected curves or comments |
| G.07 | DIVIDE | - to set the number of divisions on an axis |
| G.08 | ERASE | - to erase all the existing curves |
| G.09 | LTYPE | - to set line type of a curve |
| G.10 | MTYPE | - to set marker type of a curve |
| G.11 | PLOT | - to plot the required items |
| G.12 | PRINT | - to print the data values readin |
| G.13 | RANGE | - to set range of interest of curves |
| G.14 | READIN | - to read-in a results file |
| G.15 | RETURN | - to return to main command level |
| G.16 | SELECT | - to add or remove a curve to the plot list |
| G.17 | SHOW | - to display information on requested item |
| G.18 | TABLE | - to list curve data to file |
| G.19 | WINDOW | - to set the window to be used |

## G.1 CLEAR - to clear the screen

### Syntax

```
CLear
```

### Description

Clears the screen

### Parameters

None

### Examples

```
CL
clear
```

## G.2 COLOUR - to set the colour of items

### Syntax

```
COLour  [ITem=<choice>] [ITCOLour=<choice>]
```

### Description

To set the default colour of the requested item.

### Parameters

| | |
|---|---|
| ITem | optional *choice* : initial=TITLE<br>Specifies the item for which colour is to be set<br>Choices = (AXES,TITLE,COMMENT, BACKGROUND) |
| ITCOLour | optional *choice* : initial=GREEN<br>Specifies the colour for the chosen ITEM<br>Choices = (BLACK,GREEN,CYAN,BLUE, MAGENTA,RED,YELLOW,WHITE,GREY) |

### Examples

```
COL axes
colour title red
```

## G.3 COMMENT - define a comment on the plot

### Syntax

```
COMment  [CAPtion=<string>] [XPOS=<real>] [YPOS=<real>]
```

### Description

Add a comment beginning at (XPOS,YPOS). For the comments, the graph is taken as 0 to 1 along each axis.

### Parameters

CAPtion

optional *string* : initial=' '
Specifies the text to be added as a comment
Strings which include spaces must be enclosed in quotes

XPOS

optional *real* : initial=0.5
Specifies the $x$ coordinate of the comment position

YPOS

optional *real* : initial=0.5
Specifies the $y$ coordinate of the comment position

### Examples

```
COM 'I-V Curve 1' 0.5 0.5
comment CAP=1/2/90 xpos=0.75 ypos=0.15
```

## G.4 CTYPE - to set the colour of a curve

### Syntax

```
CType  [CURve=<integer>] [ITCOLour=<choice>]
```

### Description

Set colour of a curve. The curve number is as returned by the DEFINE command.

### Parameters

CURve

optional *integer*: initial=0
Specifies the curve to which colour is to be set

ITCOLour

optional *choice* : initial=GREEN
Specifies the colour for the chosen curve
Choices = (BLACK,GREEN,CYAN,BLUE,
MAGENTA,RED,YELLOW,WHITE,GREY)

### Examples

```
CTYPE 1 red
ctype CUR=1 ITCOL=red
```

## G.5 DEFINE - to define a curve

### Syntax

```
DEFine [CURve=<integer>]
       [AXESX=<choice>]   [CONTX=<integer>]
       [SCALX=<choice>]   [ITEMX=<choice>]
       [AXESY=<choice>]   [CONTY=<integer>]
       [SCALY=<choice>]   [ITEMY=<choice>]
```

### Description

Define creates a curve from one of the read-in data sets. Read-in data sets are numbered from 1 upwards. The numbering of contacts can be determined using the SHOW SUMMARY command. The CONTX should be a contact on which the bias is varying. Note that some parameters are redundant.

### Parameters

CURve
optional *integer*: initial=0
Specifies the read-in data set from which this curve is to be defined

AXESX
optional *choice* : initial=X
Specifies it is $x$ coordinate currently dealt with
Choices = (X)

CONTX
optional *integer*: initial=0
Specifies the contact number for $y$ axes

SCALX
optional *choice* : initial=LIN
Specifies the scaling for the $x$ axes
Choices = (LIN,LOG)

ITEMX
optional *choice* : initial=V
Specifies the item for $x$ axes
Choices = (V,IT,IP,IN,ID)

AXESY
optional *choice* : initial=Y
Specifies it is $y$ coordinate currently dealt with
Choices = (Y)

CONTY
optional *integer*: initial=0
Specifies the contact number for $y$ axes

SCALY
optional *choice* : initial=LIN
Specifies the scaling for $y$ axes
Choices = (LIN,LOG)

ITEMY
optional *choice* : initial=IT
Specifies the item for $y$ axes
Choices = (V,IT,IP,IN,ID)

### Examples

```
DEF 1 x 1 lin v y 2 lin in
DEF 2 x 1 lin v y 2 log it
def 2 itemy=ip
def 1 itemy=ip
```

## G.6 DELETE - to delete all the selected curves or comments

### Syntax

```
DELete [ITem=<choice>]
```

### Description

Delete all the selected curves or comments. Un-selected curves may be renumbered after deletion of selected curves.

### Parameters

| | |
|---|---|
| ITem | optional *choice* : initial=CURVE<br>Specifies the item to be deleted<br>Choices=(CURVE,COMMENT) |

### Examples

```
DEL curve
delete item=comment
```

## G.7 DIVIDE - to set the number of divisions on an axis

### Syntax

```
DIVide [AXES=<choice>] [ITEM=<integer>]
```

### Description

Set number of divisions on the axes.

### Parameters

| | |
|---|---|
| AXES | optional *choice* : initial=X<br>Specifies the axes whose number of divisions is to be set<br>Choices = (X,Y) |
| ITEM | optional *integer* : initial=10<br>Specifies number of divisions on the axis |

### Examples

```
DIV x 10
divide axes=y item=15
```

## G.8 ERASE - to erase all the existing curves

### Syntax

```
ERase
```

### Description

Erase all the existing curves from this level.

### Parameters

None

### Examples

```
ER
erase
```

## G.9 LTYPE - to set line type of a curve

### Syntax

```
LType [CURve=<integer>] [ITEM=<integer>]
```

### Description

Sets line type of a curve.

### Parameters

| | |
|---|---|
| CURve | optional *integer*: initial=0<br>Specifies the curve of which line type is to be set |
| ITEM | optional *integer*: initial=1<br>Specifies the line type for the chosen curve, [1:4] |

### Examples

```
LT 1 2
ltype CUR=3 ITEM=4
```

## G.10 MTYPE - to set marker type of a curve

### Syntax

```
MType [CURve=<integer>] [ITEM=<integer>]
```

### Description

Sets marker type of a curve. Types are -4 to -1 and 1 to 4.

### Parameters

| | |
|---|---|
| CURve | optional *integer*: initial=0<br>Specifies the curve of which marker type is to be set |
| ITEM | optional *integer*: initial=1<br>Specifies the marker type for the chosen curve, [-4:4] |

### Examples

```
MT 1 2
mtype CUR=3 ITEM=-2
```

## G.11 PLOT - to plot the requested items

### Syntax

```
PLot [ITEM=<choice>]
```

### Description

Plot the requested items on the current graphics output device. One or more curves must have been defined and selected before using this command.

### Parameters

ITem

optional *choice* : initial=ALL
Specifies the item to output
Choices = (ALL,AXES,CURVE,
COMMENT,TITLE,GRID)

### Examples

```
PL all
plot IT=curve
```

## G.12 PRINT - to print out the data

### Syntax

```
PRint [DAtatset=<integer>] [FIle=<string>] [OPtion=<choice>]
```

### Description

Print out voltage and current data in a readable format. This command lists all the data values (contact name, voltage, displacement, hole, and electron current) for the selected data set. Output can be sent to a file.

### Parameters

DAtaset
retained *integer* : initial=1
Specifies the dataset to output

FIle
retained *string* : initial='*'
Specifies output to screen (*) or file

OPTion
retained *choice* : initial=REPLACE
Specifies if file is to replace or append
Choices=(REPLACE,APPEND)

### Examples

```
PRINT 2
pri data=1 file=tmp app
```

## G.13 RANGE - to set range of interest of curves

### Syntax

```
RANge [MODE=<choice>] [AXES=<choice>]
      [LOW=<real>] [HIGH=<real>]
```

### Description

The axes are normally automatically scaled. If the user wishes to set a different scale the range command can be used with the manual option. The automatic option resets the default behaviour.

### Parameters

MODE
optional *choice* : initial=AUTOMATIC
Specifies the mode
Choices = (AUTOMATIC,MANUAL)

AXES
optional *choice* : initial=X
Specifies the axis to set the range on
Choices = (X,Y)

LOW
optional *real* : initial=-1.0E30
Specifies the lower limit of the range

HIGH
optional *real* : initial= 1.0E30
Specifies the upper limit of the range

### Examples

```
RAN manu y -100.0 100.0
RAN auto x
```

## G.14 READIN - to read-in a results file

### Syntax

```
READIN [ITEM=<string>]
```

### Description

This command is used to read the bias and current data from a results neutral file. Several different neutral files may be read-in to compare results of different simulations, and these will be numbered sequentially starting at one. The DEFINE command must be used to create particular curves from datasets read-in by this command.

### Parameters

| | |
|---|---|
| ITem | required *string*<br>Specifies the results neutral file name to read-in |

### Examples

```
READIN diode
```

## G.15 RETURN - to return to main command level

### Syntax

```
RETurn
```

### Description

Return to main command level

### Parameters

None

### Examples

```
RET
return
```

## G.16 SELECT - to add or remove a curve to the plot list

### Syntax

```
SELect [ITem=<integer>]
```

### Description

Curves created by the DEFINE command will not be drawn by the PLOT command until they have been selected. This requires the number of the curve as reported from the DEFINE command (or obtained via the SHOW CURVES command). All curve numbers are positive, but a negative curve number can be used to de-select a curve from the plotting list. The curve is not deleted and can be re-selected latter.

### Parameters

| | |
|---|---|
| ITem | optional *integer* : initial =1<br>Specifies the curve to select or deselect |

### Examples

```
SEL  1
SEL -2
```

## G.17 SHOW - to display information on the requested item

### Syntax

```
SHow [ITem=<string>] [FIRst=<integer>] [LAst=<integer>]
```

### Description

Display information about the requested item. This command can be used to check what datasets have been read-in and which curves have been defined and selected. Line and marker types and colours can also be viewed.

### Parameters

| | |
|---|---|
| ITem | optional *choice* : initial = GRID<br>Specifies the item to be displayed<br>Choices=(SUMMARY,CURVES,LTYPE,<br>MTYPE,COLOUR,GRID) |
| FIRst | optional *integer*: initial = 1<br>Specifies the first curve to be displayed |
| LAst | optional *integer*: initial = 100<br>Specifies the last curve to be displayed |

### Examples

```
Sh curve 1 5
show grid
```

## G.18 TABLE - to produce a tubular form of curve data

### Syntax

```
TAble [DAtatset=<integer>] [FIle=<string>] [OPtion=<choice>]
```

### Description

Print out voltage and current data of selected curves in a tabular format. Only selected curves with the same $x$ axis will be listed in a format that that can be adapted to other graphing programs, such as gnuplot. Output may be sent to a file.

### Parameters

| | |
|---|---|
| FIle | retained *string* : initial=*<br>Specifies output to screen (*) or file |
| OPTion | retained *choice* : initial=REPLACE<br>Specifies if file is to replace or append<br>Choices=(REPLACE,APPEND) |

### Examples

```
table
tab file=tmp replace
```

## G.19 WINDOW - to set the window to be used

### Syntax

```
WINdow [ITEMN=<integer>] [ITEML=<integer>]
```

### Description

Select the window to be used. This is similar to the WINDOW command in the POST level. Only 1 or 4 windows may be used.

### Parameters

| | |
|---|---|
| ITEMN | optional *integer* : initial = 1<br>Specifies total number of the windows |
| ITEML | optional *integer* : initial = 1<br>Specifies the window to be used |

### Examples

```
WIN 1
window 4 2
```

# H Transient Characteristic Commands

| | | |
|---|---|---|
| H.01 | CLEAR | - to clear the screen |
| H.02 | COLOUR | - to set colour to the required items |
| H.03 | COMMENT | - to define a comment on the plot |
| H.04 | CTYPE | - to set the colour of a curve |
| H.05 | DEFINE | - to define a curve |
| H.06 | DELETE | - to delete all the selected curves or comments |
| H.07 | DIVIDE | - to set number of divisions on an axis |
| H.08 | ERASE | - to erase all the existing curves |
| H.09 | INTEGRATE | - to integrate currents over time |
| H.10 | LTYPE | - to set line type of a curve |
| H.11 | MTYPE | - to set marker type of a curve |
| H.12 | PLOT | - to plot the required items |
| H.13 | PRINT | - to print the data values readin |
| H.14 | RANGE | - to set range of interest of curves |
| H.15 | READIN | - to read in a data set from file |
| H.16 | RETURN | - to return to main command level |
| H.17 | SELECT | - to add or remove a curve to the plot list |
| H.18 | SHOW | - to display information on the required items |
| H.19 | TABLE | - to list curve data to file |
| H.20 | WINDOW | - to set the window to be used |

## H.1 CLEAR - to clear the screen

### Syntax

```
CLear
```

### Description

Clears the screen

### Parameters

None

### Examples

```
CL
clear
```

## H.2 COLOUR - to set the colour of items

### Syntax

```
COLour  [ITem=<choice>] [ITCOLour=<choice>]
```

### Description

To set the default colour of the requested item.

### Parameters

| | |
|---|---|
| ITem | optional *choice* : initial=TITLE<br>Specifies the item for which colour is to be set<br>Choices = (AXES,TITLE,COMMENT, BACKGROUND) |
| ITCOLour | optional *choice* : initial=GREEN<br>Specifies the colour for the chosen ITEM<br>Choices = (BLACK,GREEN,CYAN,BLUE, MAGENTA,RED,YELLOW,WHITE,GREY) |

### Examples

```
COL axes
colour title red
```

## H.3 COMMENT - to define a comment on the plot

### Syntax

```
COMment   [CAPtion=<string>] [XPOS=<real>] [YPOS=<real>]
```

### Description

Add a comment at (XPOS,YPOS) For the comments, the graph is taken as 0 to 1 along each axis.

### Parameters

| | |
|---|---|
| CAPtion | optional *string* : initial=' '<br>Specifies the text to be added as a comment<br>Strings which include spaces must be enclosed in quotes |
| XPOS | optional *real* : initial=0.5<br>Specifies the $x$ coordinate of the comment position |
| YPOS | optional *real* : initial=0.5<br>Specifies the $y$ coordinate of the comment position |

### Examples

```
COM 'I-t Curve 1' 0.5 0.5
comment CAP=1/2/90 xpos=0.75 ypos=0.15
```

## H.4 CTYPE - to the set colour of a curve

### Syntax

```
CType   [CURve=<integer>] [ITCOLour=<choice>]
```

### Description

Set colour of a curve. The curve number is as returned by the DEFINE command.

### Parameters

| | |
|---|---|
| CURve | optional *integer*: initial=0<br>Specifies the curve to which colour is to be set |
| ITCOLour | optional *choice* : initial=GREEN<br>Specifies the colour for the chosen curve<br>Choices = (BLACK,GREEN,CYAN,BLUE,<br>MAGENTA,RED,YELLOW,WHITE,GREY) |

### Examples

```
CTYPE 1 red
ctype CUR=1 ITCOL=red
```

## H.5 DEFINE - to define a curve

### Syntax

```
DEFine [CURve=<integer>] [AXESX=<choice>]
       [SCALX=<choice>]  [ITEMX=<choice>]
       [AXESY=<choice>]  [CONTY=<integer>]
       [SCALY=<choice>]  [ITEMY=<choice>]
```

### Description

Define creates a curve from one of the read-in data sets. Read-in data sets are numbered from 1 upwards. The numbering of contacts can be determined using the SHOW SUMMARY command. Note that some parameters are redundant.

### Parameters

| | |
|---|---|
| CURve | optional *integer*: initial=0<br>Specifies the read-in data set from which this curve is to be defined |
| AXESX | optional *choice* : initial=X<br>Specifies it is $x$ coordinate currently dealt with<br>Choices = (X) |
| SCALX | optional *choice* : initial=LIN<br>Specifies the scaling for $x$ axis<br>Choices = (LIN,LOG) |
| ITEMX | optional *choice* : initial=T<br>Specifies the item for $x$ axis<br>Choices = T |
| AXESY | optional *choice* : initial=Y<br>Specifies it is $y$ coordinate currently dealt with<br>Choices = (Y) |
| CONTY | optional *integer*: initial=0<br>Specifies the contact number for $y$ axis |
| SCALY | optional *choice* : initial=LIN<br>Specifies the scaling for $y$ axis<br>Choices = (LIN,LOG) |
| ITEMY | optional *choice* : initial=IT<br>Specifies the item for $y$ axis<br>Choices = (V,IT,IP,IN,ID) |

### Examples

```
DEF 1 x lin t y 2 lin in
DEF 2 x lin t y 2 log it
def 2 itemy=ip
def 1 itemy=ip
```

## H.6 DELETE - to delete all the selected curves or comments

### Syntax

```
DELete [ITem=<choice>]
```

### Description

Delete all the selected curves or comments. Un-selected curves may be renumbered after deletion of selected curves.

### Parameters

ITem

optional *choice* : initial=CURVE
Specifies the item to be deleted
Choices=(CURVE,COMMENT)

### Examples

```
DEL curve
delete item=comment
```

## H.7 DIVIDE - to set number of divisions on an axis

### Syntax

```
DIVide [AXES=<choice>] [ITEM=<integer>]
```

### Description

Set number of divisions on an axis.

### Parameters

AXES

optional *choice* : initial=X
Specifies the axis whose number of divisions is to be set
Choices = (X,Y)

ITEM

optional *integer* : initial=10
Specifies number of divisions on the axis

### Examples

```
DIV x 10
divide axes=y item=15
```

## H.8 ERASE - to erase all the existing curves

### Syntax

```
ERase
```

### Description

Erase all the existing curves from this level.

### Parameters

None

### Examples

```
ER
erase
```

## H.9 LTYPE - to set the line type of a curve

### Syntax

```
LType [CURve=<integer>] [ITEM=<integer>]
```

### Description

Set line type of a curve.

### Parameters

| | |
|---|---|
| CURve | optional *integer*: initial=0<br>Specifies the curve of which line type is to be set |
| ITEM | optional *integer*: initial=1<br>Specifies the line type for the chosen curve, [1:4] |

### Examples

```
LT 1 2
ltype CUR=3 ITEM=3
```

## H.10 INTEGRATE - to integrate currents at contacts

### Syntax

```
INTegrate [TSTart=<real>] [TEND=<real>] [Units=<choice>]
      [Dataset=<integer>] [MEthod=<choice>]
```

### Description

Integrate all components of current at all contacts over time and report results in tabular form. The limits of integration and the units of charge can be set. By default all data sets are integrated, but this can be restricted to one. The default integration is based on overlapping parabola but the trapezium method can be used instead.

### Parameters

TSTart
retained *real*: initial=0
Specifies the start time for integration

TENd
retained *real*: initial=0
Specifies the end time for integration

UNIts
retained *choice*: initial=ELECTRONS
Specifies the units for charge: Electrons or Coulombs

METHOD
retained *choice*: initial=PARABOLIC
Specifies integration method to use
Choices=(PARABOLIC,TRAPEZIUM,BOTH)

### Examples

```
int 0 1d-7
INTEGRATE 1d-11 5d-9 units=c method=trap
```

## H.11 MTYPE - to set marker type of a curve

### Syntax

```
MType [CURve=<integer>] [ITEM=<integer>]
```

### Description

Set marker type of a curve. Types are -4 to -1 and 1 to 4.

### Parameters

CURve
optional *integer*: initial=0
Specifies the curve of which marker type is to be set

ITEM
optional *integer*: initial=1
Specifies the marker type for the chosen curve

### Examples

```
MT 1 2
mtype CUR=3 ITEM=-2
```

## H.12 PLOT - to plot the requested items

### Syntax

```
PLot [ITEM=<choice>]
```

### Description

Plot the requested items on the current graphics output device. One or more curves must have been defined and selected before using this command.

### Parameters

| | |
|---|---|
| ITem | optional *choice* : initial=ALL<br>Specifies the item to output<br>Choices = (ALL,AXES,CURVE,<br>COMMENT,TITLE,GRID) |

### Examples

```
PL all
plot IT=curve
```

## H.13 PRINT - to print out I-T data

### Syntax

```
PRint [DAtatset=<integer>] [FIle=<string>] [OPtion=<choice>]
```

### Description

Print out times and currents data in a readable format. This command lists all the data values (contact name, time, voltage, displacement, hole, and electron current) for the selected data set. Output can be sent to a file.

### Parameters

DAtaset
retained *integer* : initial=1
Specifies the dataset to output

FIle
retained *string* : initial='*'
Specifies output to screen (*) or file

OPTion
retained *choice* : initial=REPLACE
Specifies if file is to replace or append
Choices=(REPLACE,APPEND)

### Examples

```
PRINT 2
pri data=1 file=tmp app
```

## H.14 RANGE - to set range of interest of curves

### Syntax

```
RANge [MODE=<choice>] [AXES=<choice>]
      [LOW=<real>] [HIGH=<real>]
```

### Description

The axes are normally automatically scaled. If the user wishes to set a different scale the range command can be used with the manual option. The automatic option resets the default behaviour.

### Parameters

MODE
optional *choice* : initial=AUTOMATIC
Specifies the mode
Choices = (AUTOMATIC,MANUAL)

AXES
optional *choice* : initial=T
Specifies the axes
Choices = (T,V,I)

LOW
optional *real* : initial=-1.0E30
Specifies the lower limit of the range

HIGH
optional *real* : initial= 1.0E30
Specifies the upper limit of the range

### Examples

```
RAN auto T
RAN manu I -100.0 100.0
```

## H.15 READIN - to read-in a data set from file

### Syntax

```
READIN [ITEM=<string>]
```

### Description

This command is used to read the time and current data from a results neutral file. Several different neutral files may be read-in to compare results of different simulations, and these will be numbered sequentially starting at one. The DEFINE command must be used to create particular curves from datasets read-in by this command.

### Parameters

| | |
|---|---|
| ITem | required *string*<br>Specifies the results neutral file to read-in |

### Examples

```
READIN diode
```

## H.16 RETURN - to return to main command level

### Syntax

```
RETurn
```

### Description

Return to main command level

### Parameters

None

### Examples

```
RET
return
```

## H.17 SELECT - to add or remove a curve to the plot list

### Syntax

```
SELect [ITem=<integer>]
```

### Description

Curves created by the DEFINE command will not be drawn by the PLOT command until they have been selected. This requires the number of the curve as reported from the DEFINE command (or obtained via the SHOW CURVES command). All curve numbers are positive, but a negative curve number can be used to de-select a curve from the plotting list. The curve is not deleted and can be re-selected latter.

### Parameters

| | |
|---|---|
| ITem | optional *integer* : initial =1<br>Specifies the curve to select or deselect |

### Examples

```
SEL  1
SEL -2
```

## H.18 SHOW - to display information on the required items

### Syntax

```
SHow [ITem=<string>] [FIRst=<integer>] [LAst=<integer>]
```

### Description

Display information about the requested item. This command can be used to check what datasets have been read-in and which curves have been defined and selected. Line and marker types and colours can also be viewed.

### Parameters

| | |
|---|---|
| ITem | optional *choice* : initial = GRID<br>Specifies the item to be displayed<br>Choices=(SUMMARY,CURVES,LTYPE,<br>MTYPE,COLOUR,GRID) |
| FIRst | optional *integer*: initial = 1<br>Specifies the first curve to be displayed |
| LAst | optional *integer*: initial = 100<br>Specifies the last curve to be displayed |

### Examples

```
Sh curve 1 5
show grid
```

## H.19 TABLE - to produce a tubular form of curve data

### Syntax

```
TAble [DAtatset=<integer>] [FIle=<string>] [OPtion=<choice>]
```

### Description

Print out time and current data of selected curves in a tabular format. Only selected curves with the same $x$ axis will be listed in a format that that can be adapted to other graphing programs, such as gnuplot. Output may be sent to a file.

### Parameters

| | |
|---|---|
| FIle | retained *string* : initial='*'<br>Specifies output to screen (*) or file |
| OPTion | retained *choice* : initial=REPLACE<br>Specifies if file is to replace or append<br>Choices=(REPLACE,APPEND) |

### Examples

```
      table
tab file=tmp replace
```

## H.20 WINDOW - to set the window to be used

### Syntax

```
WINdow [ITEMN=<integer>] [ITEML=<integer>]
```

### Description

Set the window to be used. This is similar to the WINDOW command in the POST level. Only 1 or 4 windows may be used.

### Parameters

| | |
|---|---|
| ITEMN | optional *integer* : initial = 1<br>Specifies total number of the windows |
| ITEML | optional *integer* : initial = 1<br>Specifies the window to be used |

### Examples

```
WIN 1
window 4 2
```

# I Internal Commands

| | | |
|---|---|---|
| I.1 | MORE | - to display the contents a file |
| I.2 | CHANGE | - to change working directory |
| I.3 | RENAME | - to rename a file |
| I.4 | COPY | - to copy a file |
| I.5 | RM | - to delete (remove) a file |
| I.6 | LIST | - to provide directory listing |
| I.7 | WRITE | - to provide monitoring of a session |
| I.8 | READ | - to redirect the input stream to read from a file |
| I.9 | SYNTAX | - to provide the syntax of a command |
| I.10 | HELP | - to access HELP system |

## I.1 MORE - to display the contents a file

### Syntax

```
MORe FILe=<string>
```

### Description

Displays the contents of the specified file to the terminal.

### Parameters

| | |
|---|---|
| FILE | Required *string*<br>A string giving the name of the file to be displayed. |

### Examples

```
more file=ANODE
MOR CATHODE
```

## I.2 CHANGE - to change working directory

### Syntax

```
CHAnge DIRectory=<string>
```

### Description

Changes the current working directory.

### Parameters

| | |
|---|---|
| DIRECTORY | Required *string*<br>A string giving the name of the new working directory. |

### Examples

```
change directory=results
CHA MODELS
```

## I.3 RENAME - to rename a file

### Syntax

```
REName FILe1=<string> FILe2=<string>
```

### Description

Renames a given file to a new name.

### Parameters

FILE1 Required *string*
A string giving the current file name.

FILE2 Required *string*
A string giving the new file name.

### Examples

```
RENAME FILE1=RESULT1 FILE2=RESULT.SAVE
ren output1 output2
```

## I.4 COPY - to copy a file

### Syntax

```
COPy FILe1=<string> FILe2=<string>
```

### Description

Copies a given file to a new file.

### Parameters

FILE1 Required *string*
A string giving the source file name.

FILE2 Required *string*
A string giving the destination file name.

### Examples

```
COPY FILE1=RESULT1 FILE2=RESULT.SAVE
cop output1 output2
```

## I.5 RM - to delete (remove) a file

### Syntax

```
RM FILe=<string>
```

### Description

Removes (deletes) the given file from the file system.

### Parameters

FILE Required *string*
A string giving the name of the file to be removed (deleted).

### Examples

```
RM FILE=RESULT
rm output
```

## I.6 LIST - to provide directory listing

### Syntax

```
LISt [FILe=<string>]
```

### Description

Provide a listing of the current or specified directory.

### Parameters

| | |
|---|---|
| FILE | Optional *string* : initial = " "<br>A string giving the name of the file. |

### Examples

```
LIST
lis *.MSH
```

## I.7 WRITE - to provide monitoring of a session

### Syntax

```
WRIte STAte=<choice> [FIle=<string>]
      [PROmpt=<choice>]
```

### Description

Redirects the command decoder echo output to the file specified by the FILE parameter. The information flow is controlled by the STATE parameter. This command can enable the constructions of command files to drive the program in a background mode.

The echoing of the command prompt can be controlled using the PROMPT parameter.

### Parameters

| | |
|---|---|
| STATE | required *choice*<br>Controls the flow of information to the monitoring file It has values NO, OFF or CLOSE. ON switches on monitoring. OFF suspends it but does not close the file and CLOSE ends monitoring and closes the file. |
| FILE | retained *string* : initial = MONITOR<br>Output file name to receive the monitoring stream. |
| PROMPT | reset *choice* : initial = OFF<br>Allows you to select whether the command prompt is echoed in the monitoring file. It has values ON or OFF. |

### Examples

```
WRITE STATE=ON FILE=MONITOR PROMPT=OFF
wri on junk on
```

## I.8 READ - to specify a command input file

### Syntax

```
REad FIle=<string> [,ECHO=<choice>]
```

### Description

Redirects the command decoder to take its input from a file specified by the FILE parameter. However, if FILE is given as TERMINAL, input returns to the standard input stream.

The echoing of the commands being read by the decoder can be controlled using the ECHO parameter.

### Parameters

| | |
|---|---|
| FILE | required *string*<br>Input file name containing program commands. |
| ECHO | reset *choice* : initial = OFF<br>Echo control option. Values can be ON or OFF. |

### Examples

In this example a sequence of commands are read from the file NAIL and ECHOed to the standard output device:

```
Everest: read nail echo=on
```

## I.9 SYNTAX - to provide the syntax of a command

### Syntax

```
SYNtax [COMmand=<string>]
```

### Description

Displays the formal syntax of all the currently defined commands. If the syntax of a specific command name is required then that name is given as a parameter to the command.

### Parameters

| | |
|---|---|
| COMMAND | retained *string* : initial = ALL<br>Specifies the commands name for which the syntax is required. If the syntax of all the currently defined commands is required, then the special command name ALL should be used. |

### Examples

The following example obtains the syntax of all the commands in the DOCUMENT program.

```
Doc: syntax

The commands currently defined are:

SYNtax [COMmand=<string>]
Help [KEY=<string>] [OPTion=<string>]
FILe INput=<string>, OUTput=<string>
PROcess
Quit
TItle TEXT=<string>
AUthor TEXT=<string>
Date TEXT=<string>
OPtions [SORT=<choice>] [CONtents=<choice>]
        [RUNoff=<choice>] [FRont_page=<choice>]

 For further information type: HELP <command name> [<option>],
 where <option> is BRIEF or FULL

 Doc:
```

## I.10 HELP - to access HELP system

### Syntax

```
Help [KEY=<string>] [OPTion=<choice>]
```

### Description

Accesses to the inbuilt HELP system within the command decoder. HELP is one of the internal commands of the command processor and has a companion command SYNTAX.

HELP has two parameters allowing the selection of help on a specific command and the level of help required (SUMMARY, BRIEF, FULL and SYNTAX). If no command name is given summary help is given on all the commands currently defined.

If an ambiguous or invalid command name is given a warning or error message is given.

BRIEF help gives information on the purpose, syntax and the current state of the selected command. A table of command keywords, their type, status and current value (if applicable) is printed.

When the FULL option is used the Help System uses the inbuilt free text retrieval system to access the help data base. This allows the display of the full command description and the searching for specific keywords.

### Parameters

KEY — reset *string* : initial =
Either the global command name SUMMARY, or the specific command name on which help is sought.

OPTION — reset *choice* : initial = BRIEF
The level of help required. This can be SUMMARY, BRIEF, FULL or SYNTAX.

## Examples

```
Everest: help output

Name     : OUTPUT

Purpose  : to specify results file
Syntax   : OUTput FILE=<string> [,REPLACE=<choice>]


Keyword     Type            Status        Current Value
------------------------------------------------------------

FILE        string          required       undefined
REPLACE     choice          reset          replace,NOREPLACE
```