PROXIMITY EXPOSURE STUDIES IN ELECTRON BEAM LITHOGRAPHY

by

P. R. Deshmukh

Memorandum No. UCB/ERL M84/69

14 September 1984

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

PROXIMITY EXPOSURE STUDIES IN ELECTRON BEAM LITHOGRAPHY

P. R. DESHMUKH

(Central Electronics Engineering Research Institute)
Pilani (Rajasthan) 333031 INDIA

(Report of the work done at University of California, Electronics Research Laboratory, Berkeley, USA during July to September 1984)

Professor in charge
T. VAN DUZER

ABSTRACT

This report contains the description of the various algorithms developed for the determination of electron exposure of resist under different conditions such as electron beam energy, resist (PMMA) thickness, substrate (Si or Pb), etc. Monte Carlo simulation process has been used to obtain the energy deposition function (EDF) curves at different depths in the resist for single electron line source or single electron point source. The actual energy deposition corresponding to the gaussian electron beam of diameter 0.12 μm (typical beam diameter used for writing in LEBES electron beam lithography machine) is obtained after convolution of EDF curves with gaussian beam profile. Total exposure in patterns of 5x9 μm, separated by 0.64 μm (a typical requirement for the fabrication of submicron Josephson junctions) has been obtained assuming that the patterns are composed of long lines. The proximity corrected exposure of patterns is obtained by varying the line charge density according to an iterative solution. The iterative solution for line charge densities is based on the criterion of equalization of exposure within the patterns. Both intrapattern and interpattern proximity corrections have been obtained.

ACKNOWLEDGEMENT

I wish to extend my sincere thanks towards Professor T. Van Duzer for his constant attention during the course of study. Regular discussions with him made it possible to bring the proximity exposure analysis to some acceptable level. I also wish to thank Professor A. R. Neureuther for the fruitful discussions. I wish to thank R. E. Jewett, Herbert Ko, Karen Irwin, Vallath Nandakumar, Paul Bradley, David Petersen, Kevin Clark and Albert Chen. They remained cooperative all the time. I also wish to thank Professor D. J. Angelakos, Dr. Amarjit Singh, and Dr. W. S. Khokle for considering me in this collaborative project on electron beam lithography sponsored by NSF and CSIR. Finally I wish to thank N. K. L. Raja for his homely association.

Deshmukh P. R.

September 18, 1984

LIST OF ALGORITHMS

(1) RANGE.F — A fortran file for computation of electron range in different materials.

(2) RESIS1VAX.F — A fortran file for simulation of electron trajectories using Monte Carlo process.

(3) RESIS2VAX.F — A fortran file for computation of energy deposition function for single electron line source or single electron point source using Monte Carlo process.

(4) RESIS3VAX.F — A fortran file which convolves the energy deposition function for single electron line source obtained through running RESIS2VAX.F with a gaussian beam profile.

(5) LINETOTAL.F — A fortran file which computes the total exposure in a pattern composed of lines. It also generates the file LEXPOSURE for plotting the total exposure in the pattern.

(6) LPROXI1.F — A fortran file which computes the normalized line charge density for each line in a pattern after intraproximity correction.

(7) LPROXI2.F — A fortran file which computes the normalized line charge density for each line in a pattern after intraproximity and interproximity corrections.

(8) READIMPULSE.F — A fortran file which generates the file GENERAL to be used by RUNPLOT to plot delta line EDF

curves stored in the file IMPULSE.

(9) READIMPULSER.F — A fortran file which generates the file GENERAL for plotting EDF curves for delta electron point source stored in the file IMPULSE.

(10) READLINEEDF.F — A fortran file which generates the file GENERAL using the data file LINEEDF to plot the convolved line EDF curves.

LIST OF DATA FILES

(1) FORSCATT — A data file which stores X-Y coordinates of simulated electron trajectories obtained through running RESIS1VAX.F.

(2) IMPULSE — A data file which stores energy deposition function for single electron line source and single electron point source obtained through running RESIS2VAX.F.

(3) LINEEDF — A data file which stores convolved line EDF curves obtained through running RESIS3VAX.F.

(4) JUMPINSERT.C — A graphic file that operates on data file FORSCATT and generates a file TEMP, which in turn can be used by RUNPLOT for plotting the simulated electron trajectories.

(5) LINEDATA — A data file which is required for running LINETOTAL.F. User is required to write the data in this file in the form BEAM POSITION (same as line position), line width, step and dose. Data in

the above form is written automatically by running the programs LPROXI1.F and LPROXI2.F.

(6) RANGEDATA — A data file which is required for running RESIS2VAX.F. It contains the electron range values in microns at 1 KeV intervals from 1 KeV to 30 KeV for Si or Pb.

(1) RANGE.F

This program computes the Bethe electron range in different materials. The method used is discussed by Everhart (J.Appl.Phy.Vol.42,1971,pp 5837-5846). For range, the expression 1 is evaluated.

$$Rb = 9.40\times10^{-12}\times\frac{I^2A}{z\rho}\int_0^{\varepsilon_0}\frac{\varepsilon d\varepsilon}{\ln(\varepsilon)} \qquad (1)$$

Where,

ρ = Density of the material in g/cm .

E = Beam energy in eV.

I = Excitation energy in eV.

a = 1.1658

ε = aE/I.

A = At. Wt.

Z = At. No.

CONST = $9.40\times10^{-12}I^2A/Z$

To get range in PMMA, Si, Au, Pb and Nb, constants like I, CONST, E are to be changed. These constants are included on the comment lines. Before running the file read all comment lines carefully. Figures 1 to 5 show the electron range at different energies in PMMA, Si, Pb, Au, and Nb respectively.

(2) RESIS1VAX.F

This program simulates the electron trajectories in resist-substrate combination. It is also possible to simulate the trajectories in bulk resist or bulk substrate. The Monte Carlo model as described by Howryluk (J.Appl.Phy.Vol.45,No.6,June1974,pp2551-2566) is used for this purpose. The simulation process starts with the calculations of total scattering cross sections for the constituents of PMMA (C,H,O) and substrate atom (Si or Pb) using screened Rutherford scattering formula. If the electron is found to be in resist, then relative scattering probabilities for C, H, O are determined by using the relation,

$$Pi =(ni*\sigma i)/(ni*\sigma i) \quad \text{----} \quad (i = C , H , O)$$

Where,

ni = atomic density of ith type atom.

σi = total scattering cross section for ith type atom.

A random number R1 between 0 and 1 is used to decide which type of atom will act as a scatterer by considering the inequalities, Hydrogen if $R1 < PH$, Carbon if $PH < R1 =< PH+PC$, Oxygen if $R1 > PH+PC$. For each scattering event the azimuthal angles ϑ and φ are calculated using another set of random numbers R2, and R3. The distance traveled by the electron between successive scattering events is calculated from $S = -\lambda(R4)$, where

λ is mean free path given by the relation $\lambda=(\Sigma*\sigma*\sigma^{-1})$ and R4 another random number.

Bethe equation is used to determine the electron energy loss between successive scattering events. This energy loss is attributed to the resist exposure. The electron trajectories are traced till their energy falls near to the excitation energy of PMMA or of the substrate atom, depending upon the electron spatial position. For plotting the trajectories, X and Z coordinates are stored in the data file FORSCATT. While running the program RESIS1VAX.F it will ask the values of beam energy, film thickness and AISI (excitation energy for Si, Pb). If trajectories in bulk PMMA are desired, then a value of film thickness larger than the electron range in PMMA must be specified. If trajectories in bulk substrate are desired then the film thickness should be set to zero.

Figures 6 and 7 show the simulated trajectories in 1.2 μm PMMA on Si and Pb at 20KeV beam energy, respectively.

(3) RESIS2VAX.F

This program computes the energy deposition in the resist using Monte Corlo process as described in the previous section. Electron energy loss between successive scattering events is deposited in the resist cells. Resist film is divided into cell structure. There are 150 cells in X direction and 20 in Z direction. Only four cells in Z direction have been used. Cell dimension in X-direction is 200 A. Cell dimension in Y direction is one fourth of the resist thickness. The energy deposition values for single electron line source and single electron point source are stored in a data file IMPULSE. To run this program one has to prepare a data file RANGEDATA. This file contains the electron range in microns for energy 1 to 30KeV at 1KeV intervals in the format 8F10.4. This has been done to reduce the computational time. If the electron is in the substrate then, its distance from the resist is compared with its range. For range less than the distance the electron is assumed to be stopped. An average of 5000 trajectories is taken for the computation of energy deposition values. Figures 8 and 9 show the delta line and radial energy deposition curves for 20KeV beam energy, with 1.2 μm PMMA on Si respectively. Figures 10 and 11 show the results for the case of Pb substrate.

(4) RESIS3VAX.F

This program does the one dimensional convolution of gaussian beam profile with delta line EDF curves. It takes EDF values from the data file IMPULSE. After convolution the convolved EDF values are stored in a data file named LINEEDF. In the gaussian beam profile the standard deviation, 2o, is taken to be half the beam diameter. The program will ask the value of beam diameter in microns. Figures 12 and 13 show the convolved line EDF curves for 0.12 μm beam diameter at beam energy 20KeV for silicon and lead, respectively.

(5) LINETOTAL.F

This is a program for computation of total exposure in one dimensional pattern. Pattern may be only one beam line or multiple beam lines with a given beam shift. The pattern will be exposed line by line. Line to line gap can be varied and each line may be assigned different normalized line charge density (Ql). The default value of Ql is 1.

Before running this program the user writes the values of line position, pattern width, and line to line gap in μm, and normalized line charge density format 4F6.2 in the data file LINEDATA.

For example, to get the composite exposure for two lines separated by 0.64 μm one specifies the data as

```
-00.32  0.00  0.00  1.00
+00.32  0.00  0.00  1.00
-08.00
```

-08.00 indicate the end of the data.

Another example to get the exposure in a pattern of 4.96 μm line width separated by 0.64 μm, the data file is,

```
-05.28  4.96  0.00  1.00
+00.32  4.96  0.00  1.00
-08.00
```

Figures 14, 15, and 16 show the composite exposure of single line, two lines separated by 1 μm, and three lines separated by 0.5 μm, respectively, for 1.2 μm PMMA on silicon. Beam energy is 20KeV and Beam diameter 0.12 μm for all the cases. Note that the difference between maximum and minimum exposure level of the bottom layer exposure profile decreases as the line separation decreases.

Figures 17 to 19 show the same line exposure profiles for 30KeV beam energy. Note that the energy level difference mentioned above increases because of less proximity exposure.

(6) LPROXI1.F

This program does only the intra-proximity correction and computes the line charge density for each line in a pattern so that the exposure within the pattern is uniform. The Line separation is 0.16 μm. Exposure at the resist-substrate interface is equalized. Solution to the line charge densities is obtained by an iterative process. When this program is run it will ask the number of lines in a pattern. For example, number of lines in a pattern of 4.96 μm will be (4.96/0.16)+1. This program also generates a data file LINEDATA to be used by LINETOTAL.f for computation of exposure in the pattern. Figures 21 and 22 show the total exposure in a pattern before and after intra-proximity correction respectively. Note that edge slope of the exposure profile increases after proximity correction. Pattern definition is also improved. Figure 23 shows inter-proximity effect. In the gap, exposure on both sides increases.

(7) LPROXI2.F

With the help of this program one can determine the normalized line charge density for each line in each pattern after both intra and inter-proximity corrections. This program will ask the total number of lines involved in the patterns. The lines will be spaced by 0.16 μm. It will also ask the pattern to pattern gap. One has to give the value of pattern gap in the units of 0.16 μm. For example, if the pattern width is 4.96 μm and pattern to pattern gap is 0.64 μm, then the total number of lines in the two patterns will be 2*(4.96/0.16 +1), which is equal to 64, and the pattern gap in units of 0.16 μm will be (0.64/0.16), which is equal to 4.

The program will also generate a data file LINEDATA as described in the previous section. After running this program run LINETOTAL.F to get the composite proximity corrected exposure profile. Figure 24 shows the exposure profile for the same pattern of Fig.23, after both types of proximity corrections have been done. Note that exposure profile near the gap also becomes flat.

CONCLUSION

Monte Carlo simulation process has been successfully used for the determination of electron exposure of PMMA on silicon and lead. Energy deposition curves at the resist substrate interface differ appreciably. Energy deposition is more for lead than silicon. This indicates that inter-proximity exposure for lead in case of closely spaced patterns will be more in their separation region. Total exposure (without proximity correction) for 1.2 μm PMMA on Si and Pb in a pattern of 9x5 μm separated by 0.64 μm, shown in Fig.20,27 reveal many things. Level of exposure profile for the bottom layer is much higher than for the top layer. The difference of maximum exposure level of top layer exposure profile and minimum exposure level of bottom layer exposure profile is not large. So the bottom layer exposure profile mimimum in the gap does not go well below the maximum exposure level of top layer exposure profile. Because of this small difference of two exposure levels the pattern separation after development is not guaranteed. The two patterns are likely to immerse into one another. After intra and inter-proximity corrections the profiles are shown in Fig.25 and 27. It can be seen that the difference of the two levels increases. In this case it should be possible to delineate the two patterns without immersing into one another. In Fig.29 extra exposure in the center of each pattern is done without affecting the exposure profile levels near the gap. This technique should in principle enhance the development process. The one dimensional intra and inter-proximity correction algorithm developed using iterative process can be extended to two dimensional case.

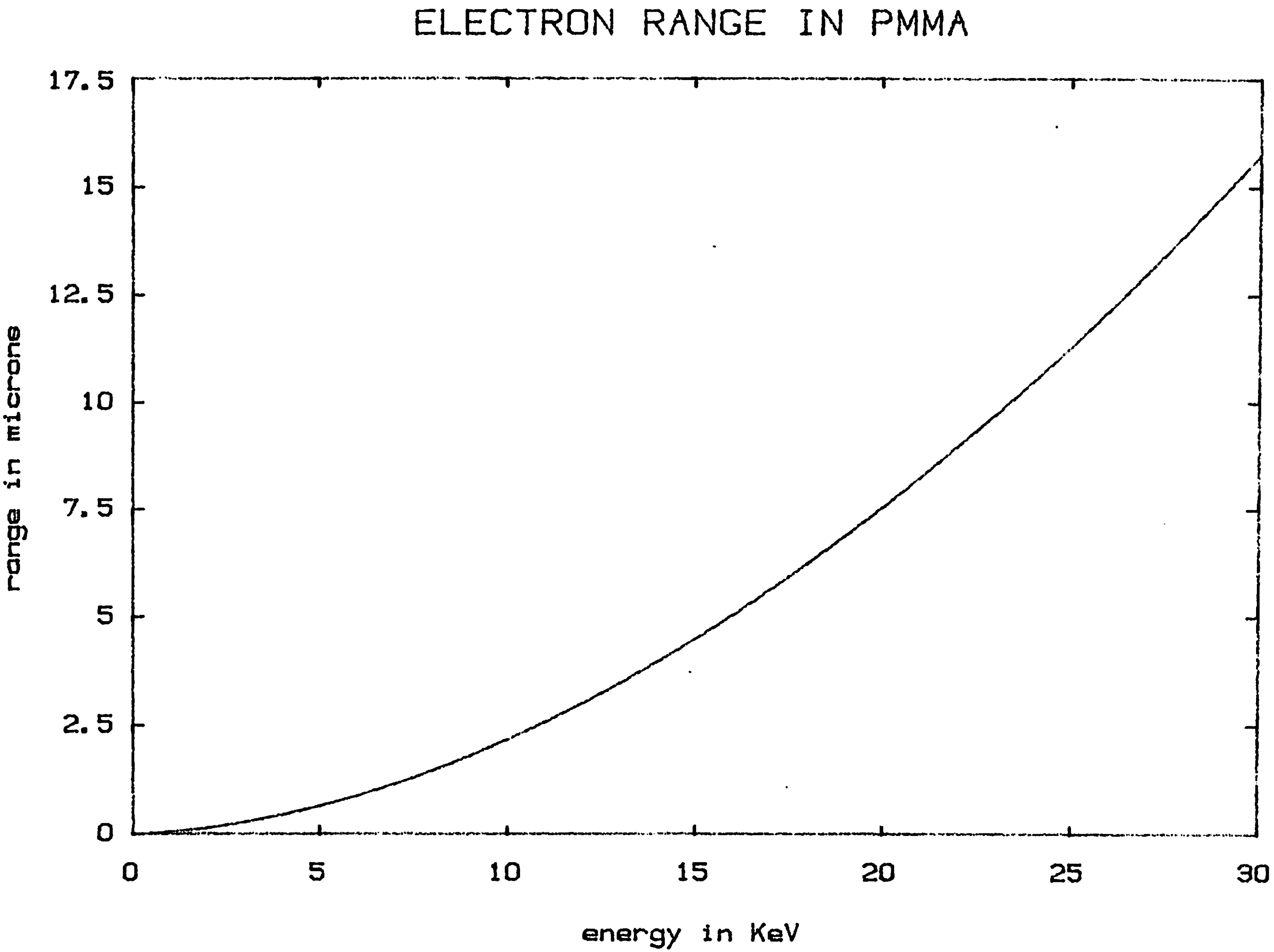

ELECTRON RANGE IN PMMA
17.5
15
12.5
10
7.5
5
2.5
0
range in microns
0
5
10
15
20
25
30
energy in KeV

Fig. 1

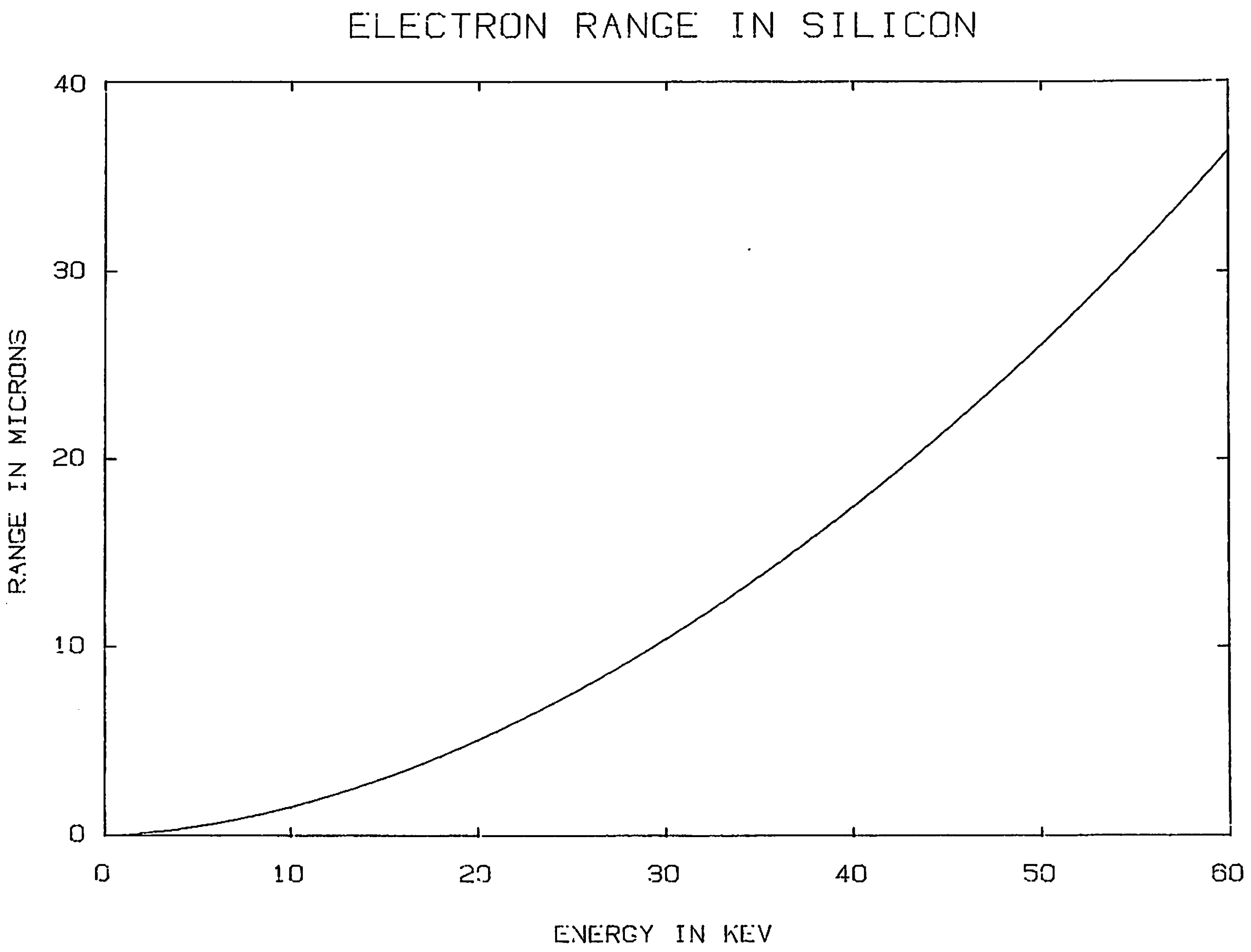

ELECTRON RANGE IN SILICON
RANGE IN MICRONS
0
10
20
30
40
ENERGY IN KEV
0
10
20
30
40
50
60

Fig.2

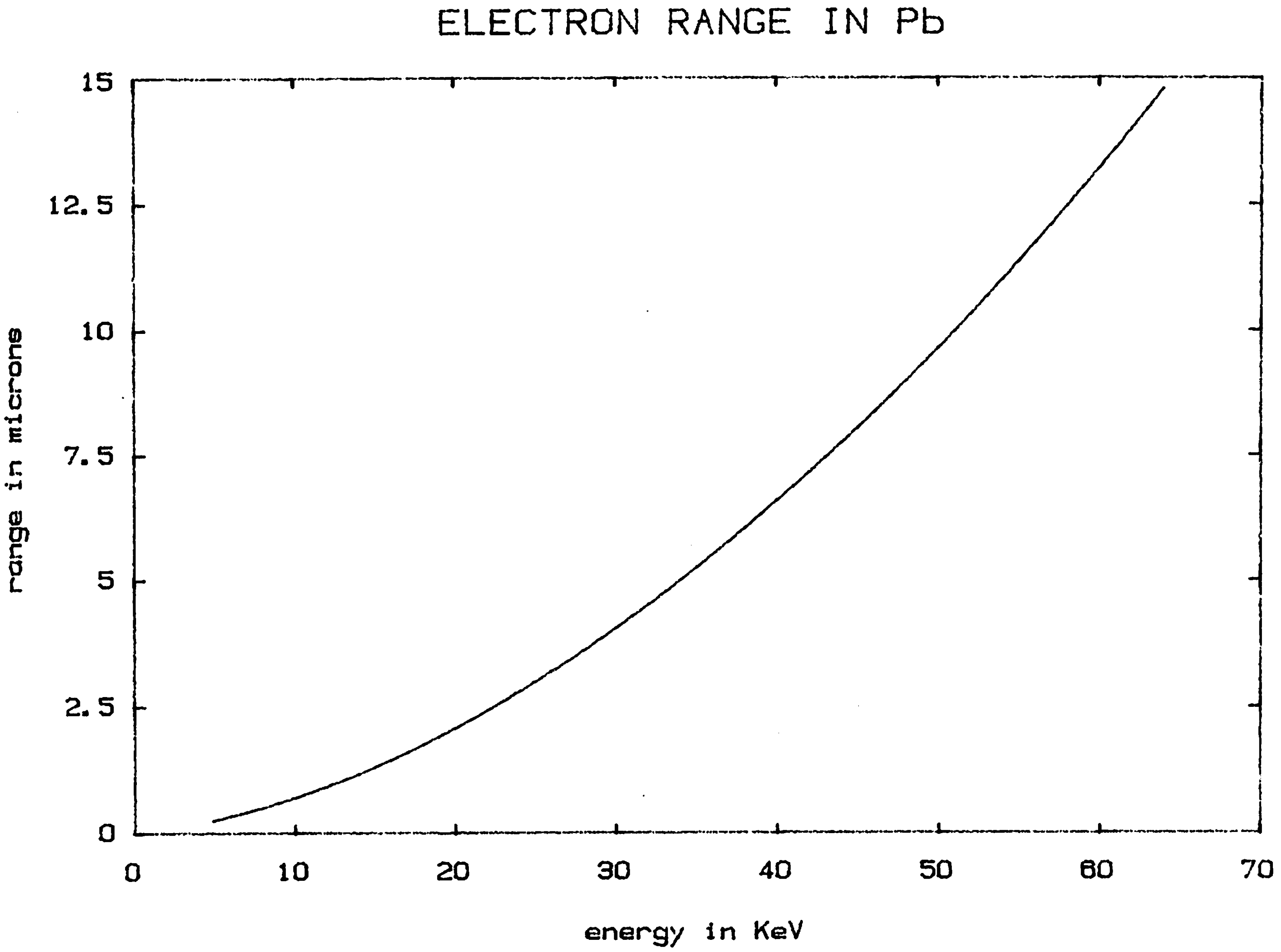

ELECTRON RANGE IN Pb
15
12.5
10
7.5
5
2.5
0
range in microns
0
10
20
30
40
50
60
70
energy in KeV

Fig.3

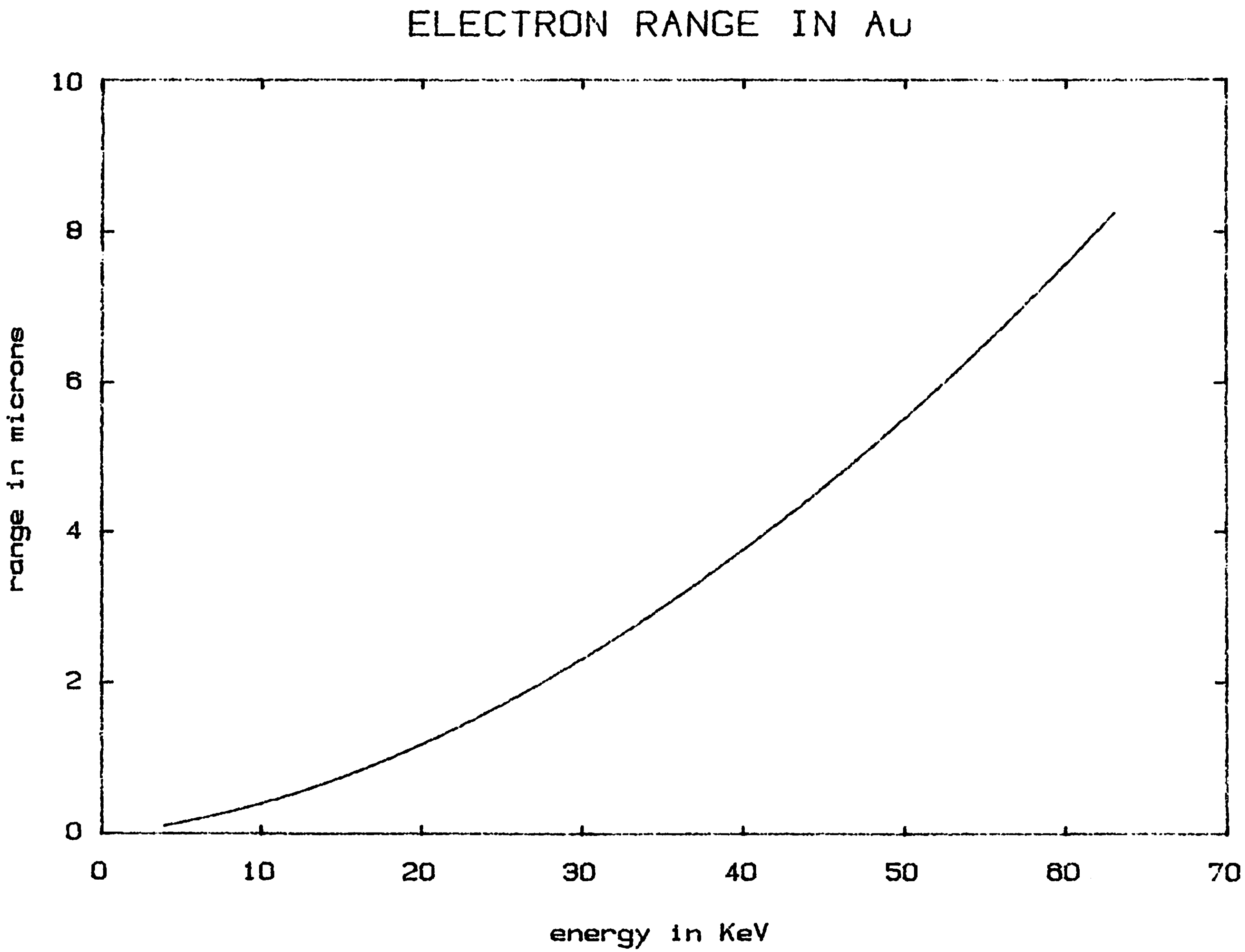

ELECTRON RANGE IN Au
range in microns
0
2
4
6
8
10
0
10
20
30
40
50
60
70
energy in KeV

Fig.4

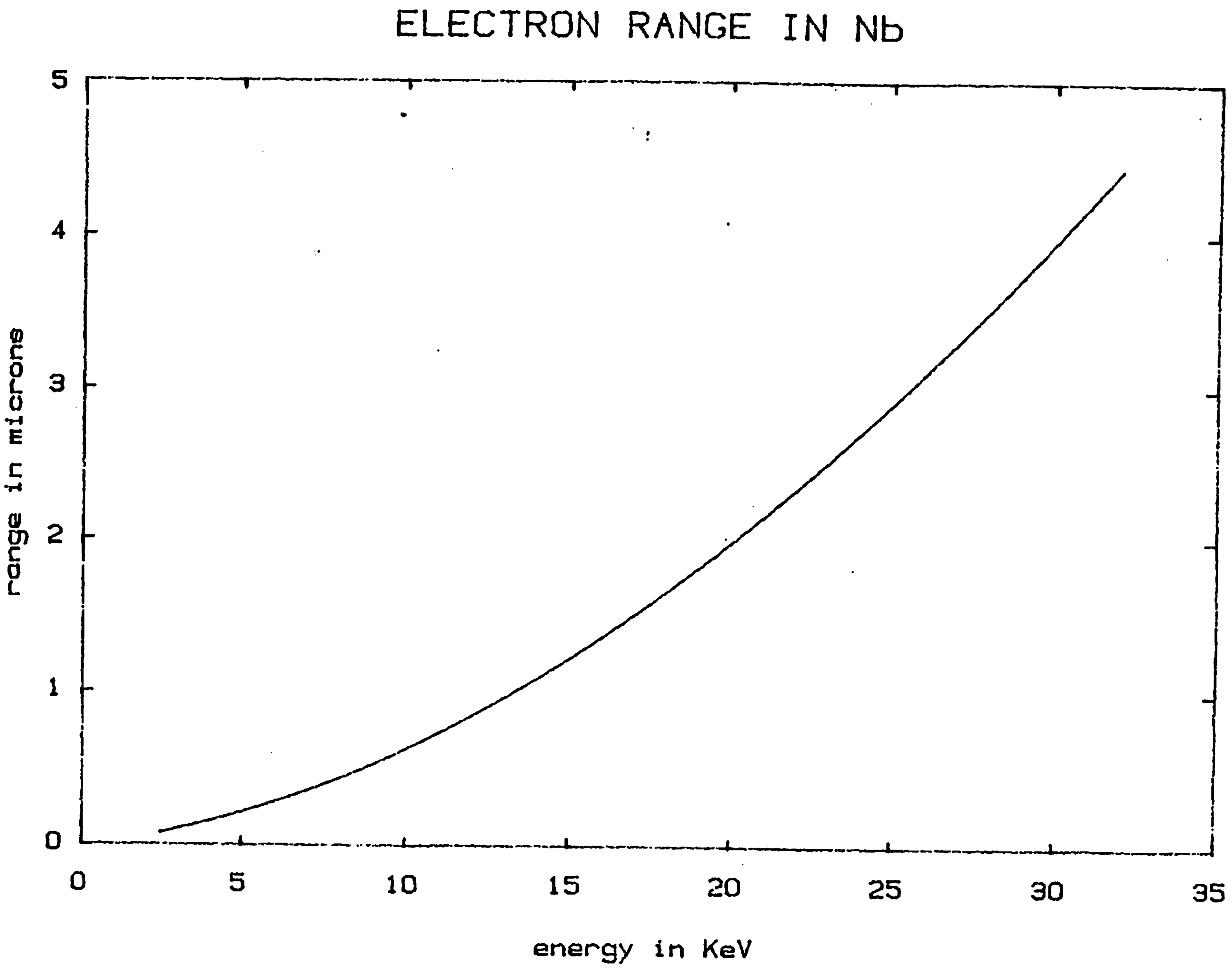

ELECTRON RANGE IN Nb
range in microns
0
1
2
3
4
5
0
5
10
15
20
25
30
35
energy in KeV

Fig.5

Monte Carlo Simulation of Electron Trajectories

(1.2μm PMMA on Silicon, Energy 20KeV)

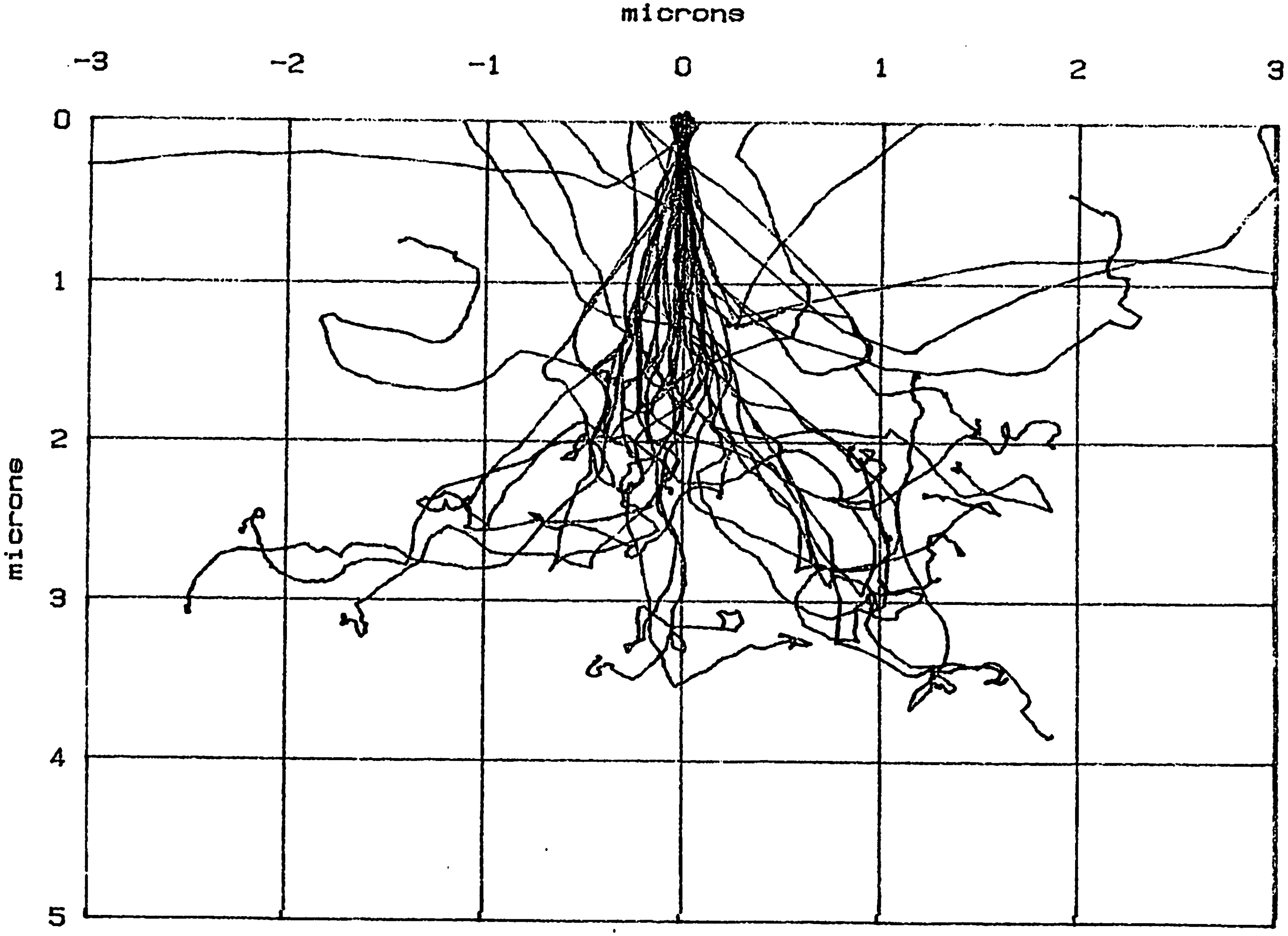
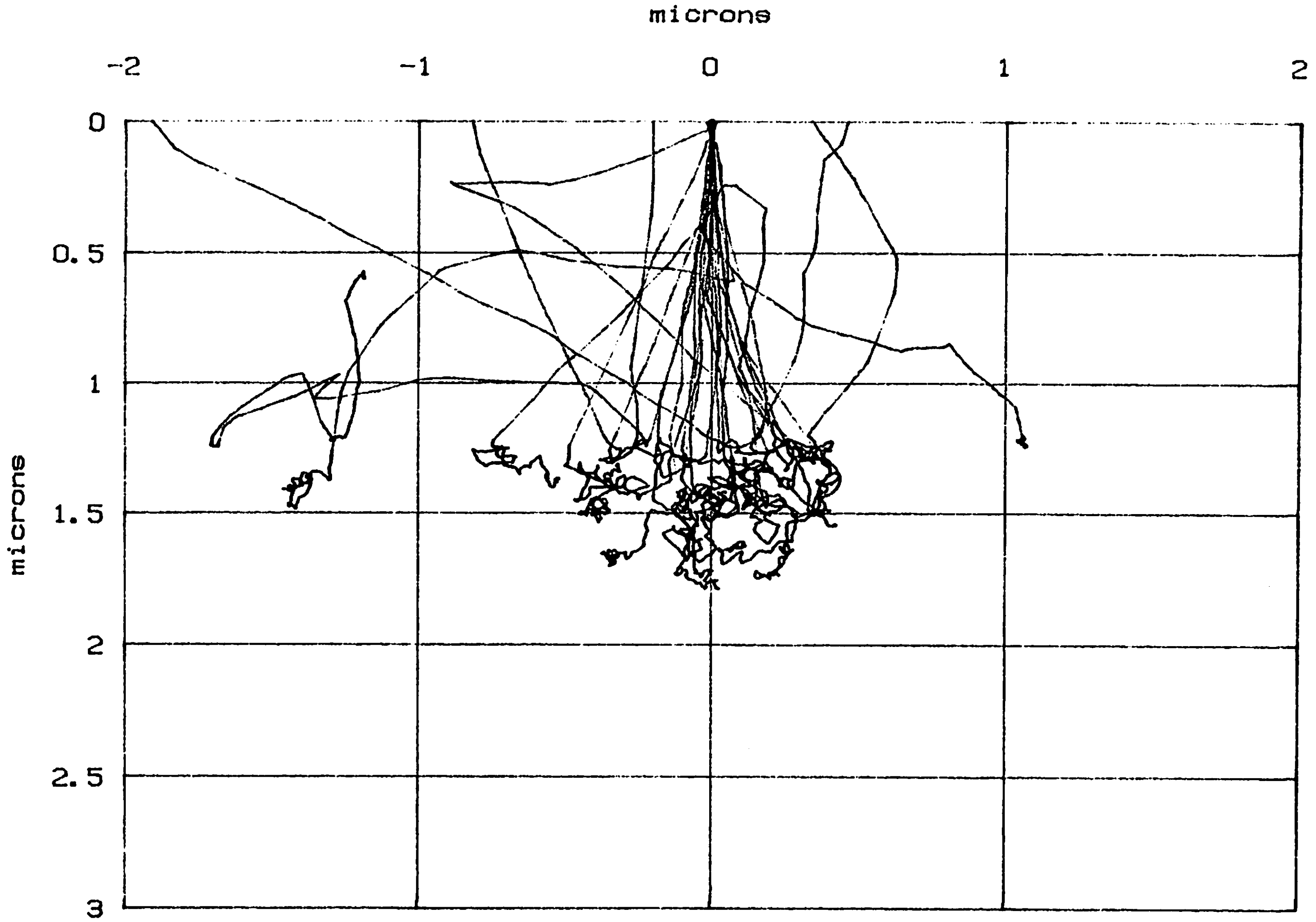


Fig.6



Monte Carlo Simulation of Electron Trajectories

(1.2μm PMMA on LEAD, Energy 20KeV)

Fig.7

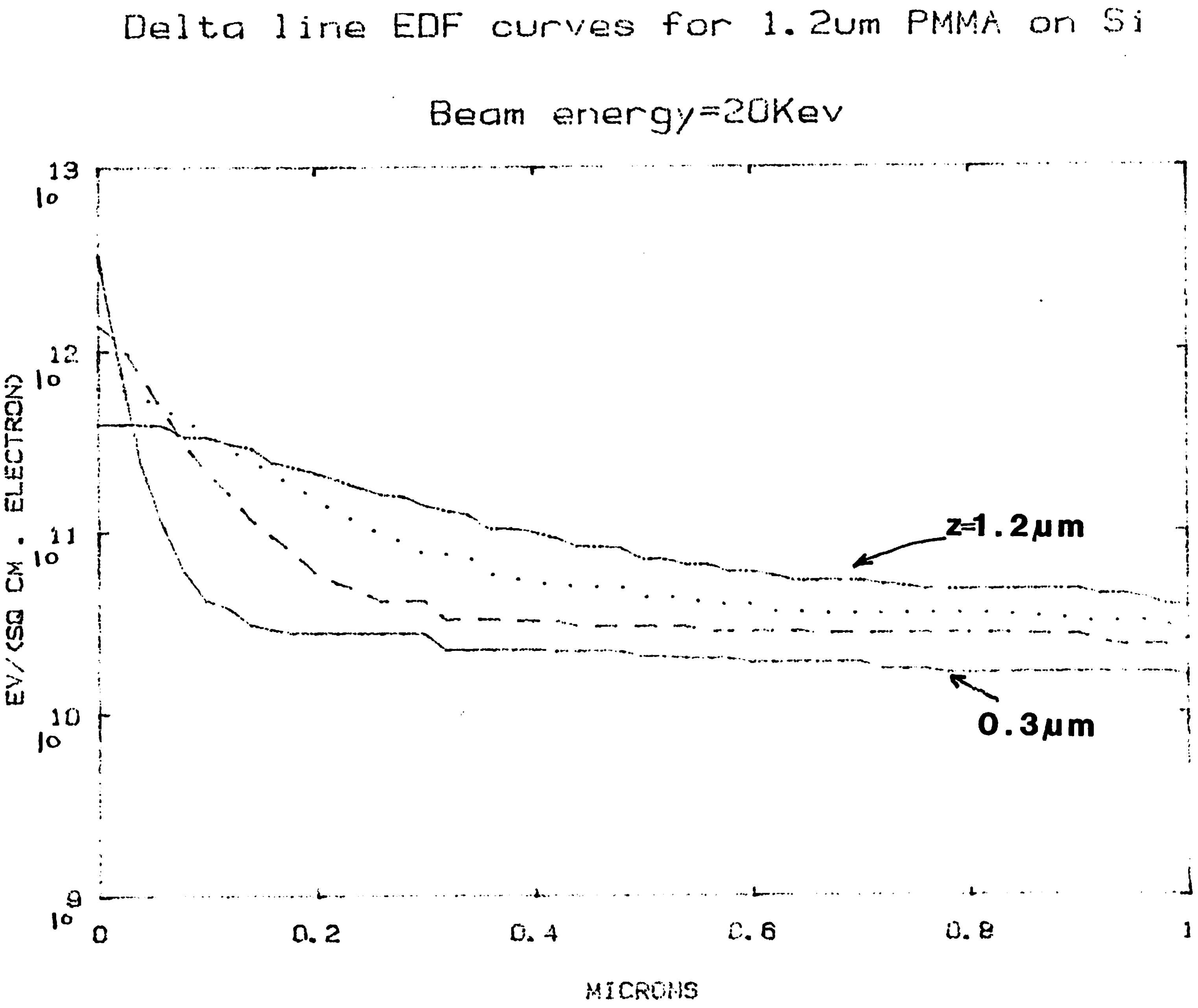

Delta line EDF curves for 1.2um PMMA on Si
Beam energy=20Kev
10^{13}
10^{12}
10^{11}
10^{10}
10^{9}
EV/(SQ CM . ELECTRON)
z=1.2µm
0.3µm
0
0.2
0.4
0.6
0.8
1
MICRONS

Fig.8

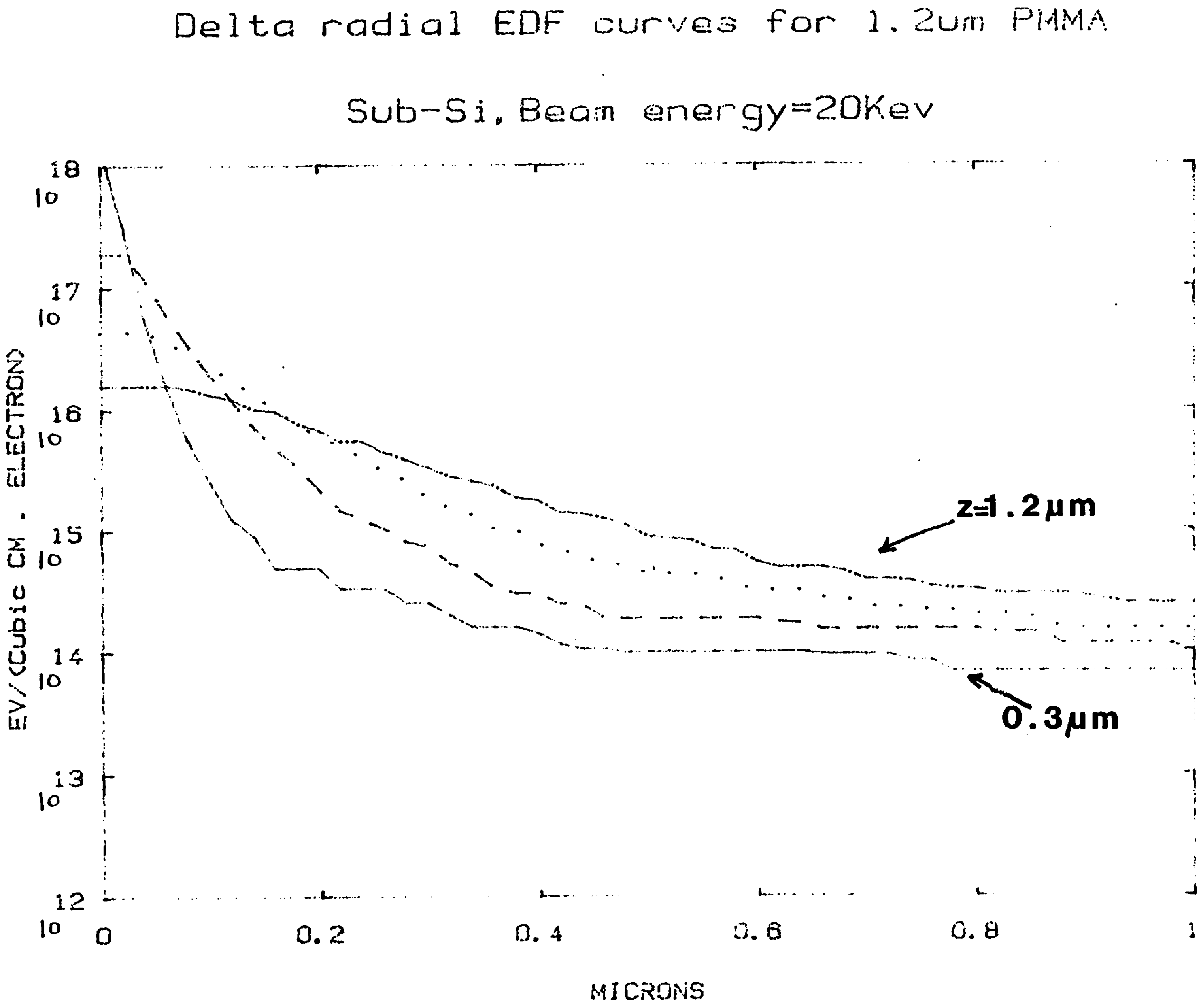

Delta radial EDF curves for 1.2um PMMA
Sub-Si, Beam energy=20Kev
EV/(Cubic CM . ELECTRON)
10^{18}
10^{17}
10^{16}
10^{15}
10^{14}
10^{13}
10^{12}
0
0.2
0.4
0.6
0.8
1
MICRONS
z=1.2µm
0.3µm

Fig.9

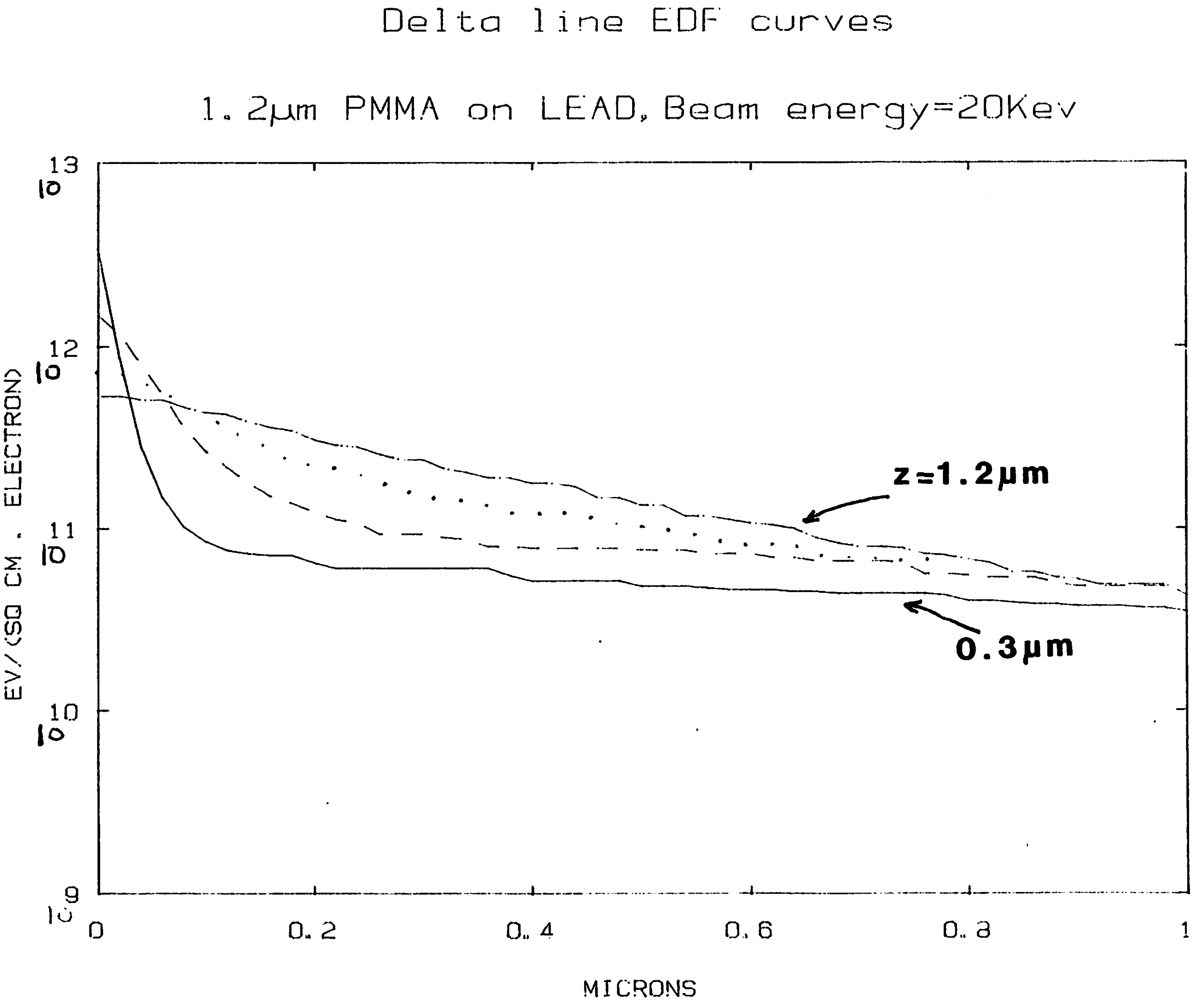

Delta line EDF curves
1.2μm PMMA on LEAD, Beam energy=20Kev
10^{13}
10^{12}
10^{11}
10^{10}
10^{9}
EV/(SQ CM . ELECTRON)
z=1.2μm
0.3μm
0
0.2
0.4
0.6
0.8
1
MICRONS

Fig.10

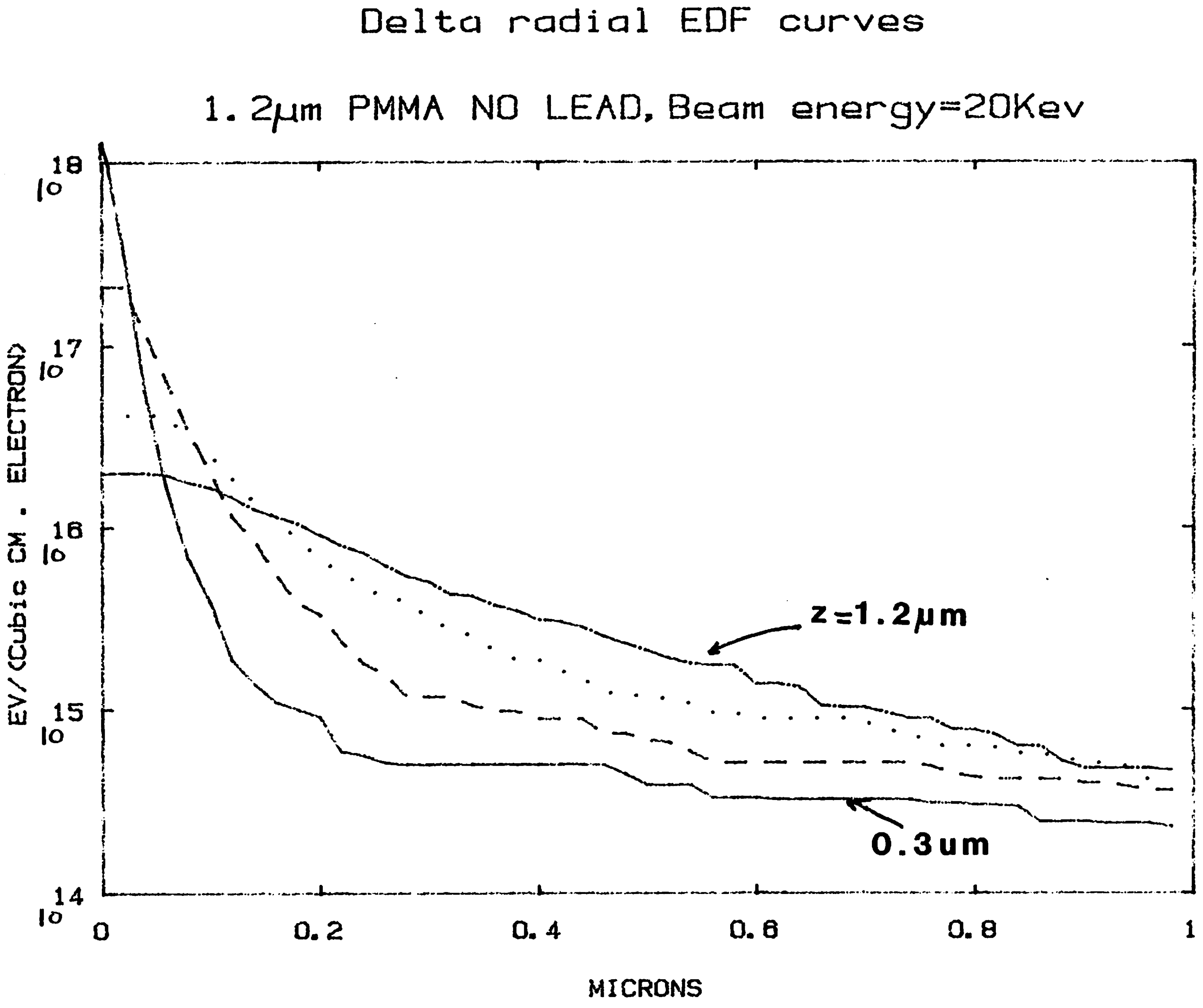
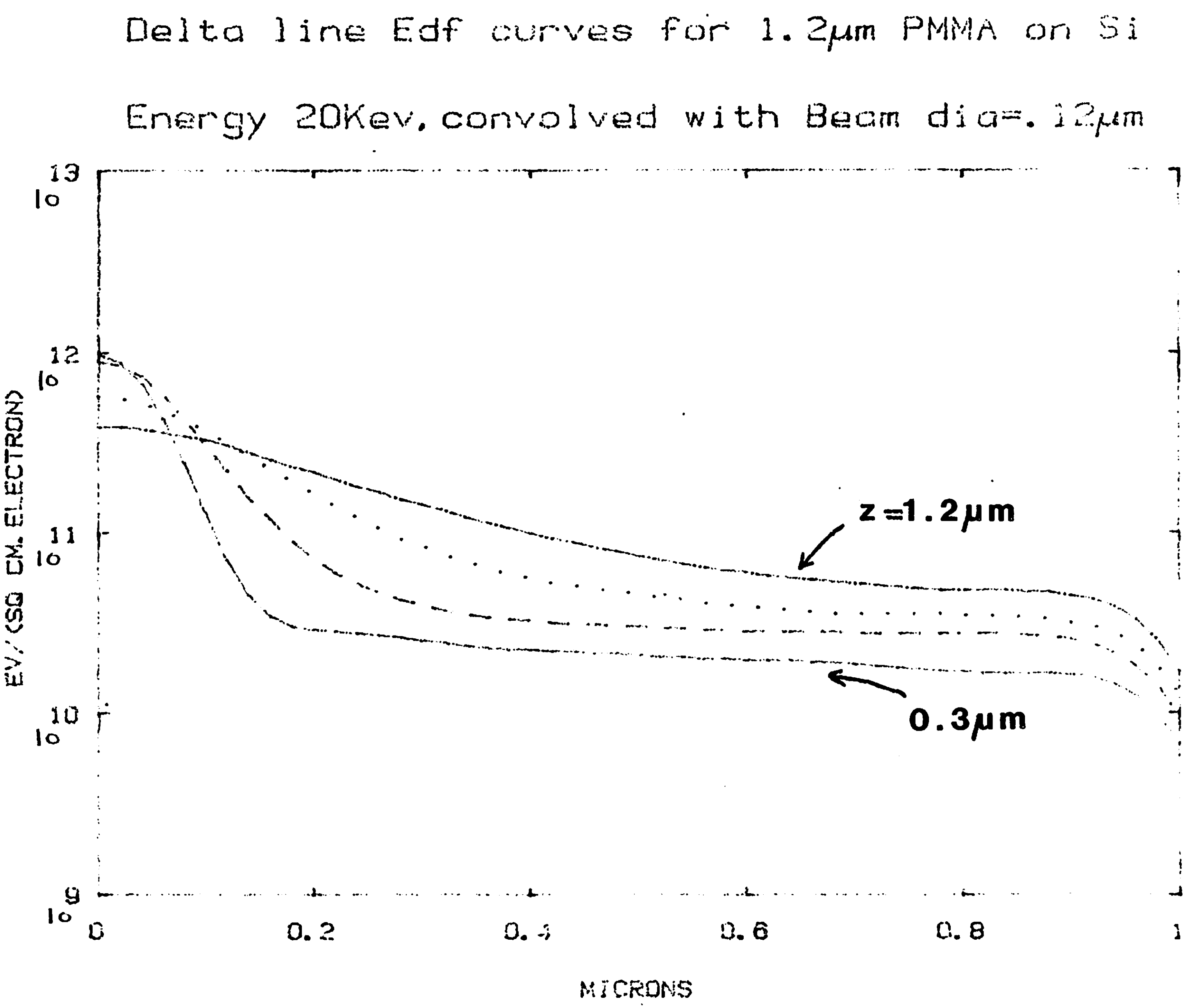

Delta radial EDF curves
1.2μm PMMA NO LEAD, Beam energy=20Kev
EV/(Cubic CM . ELECTRON)
10^{18}
10^{17}
10^{16}
10^{15}
10^{14}
0
0.2
0.4
0.6
0.8
1
MICRONS
z=1.2μm
0.3um

Fig.11



Delta line Edf curves for 1.2μm PMMA on Si
Energy 20Kev, convolved with Beam dia=.12μm
EV/(SQ CM. ELECTRON)
10^13
10^12
10^11
10^10
10^9
0
0.2
0.4
0.6
0.8
1
MICRONS
z=1.2μm
0.3μm

Fig.12

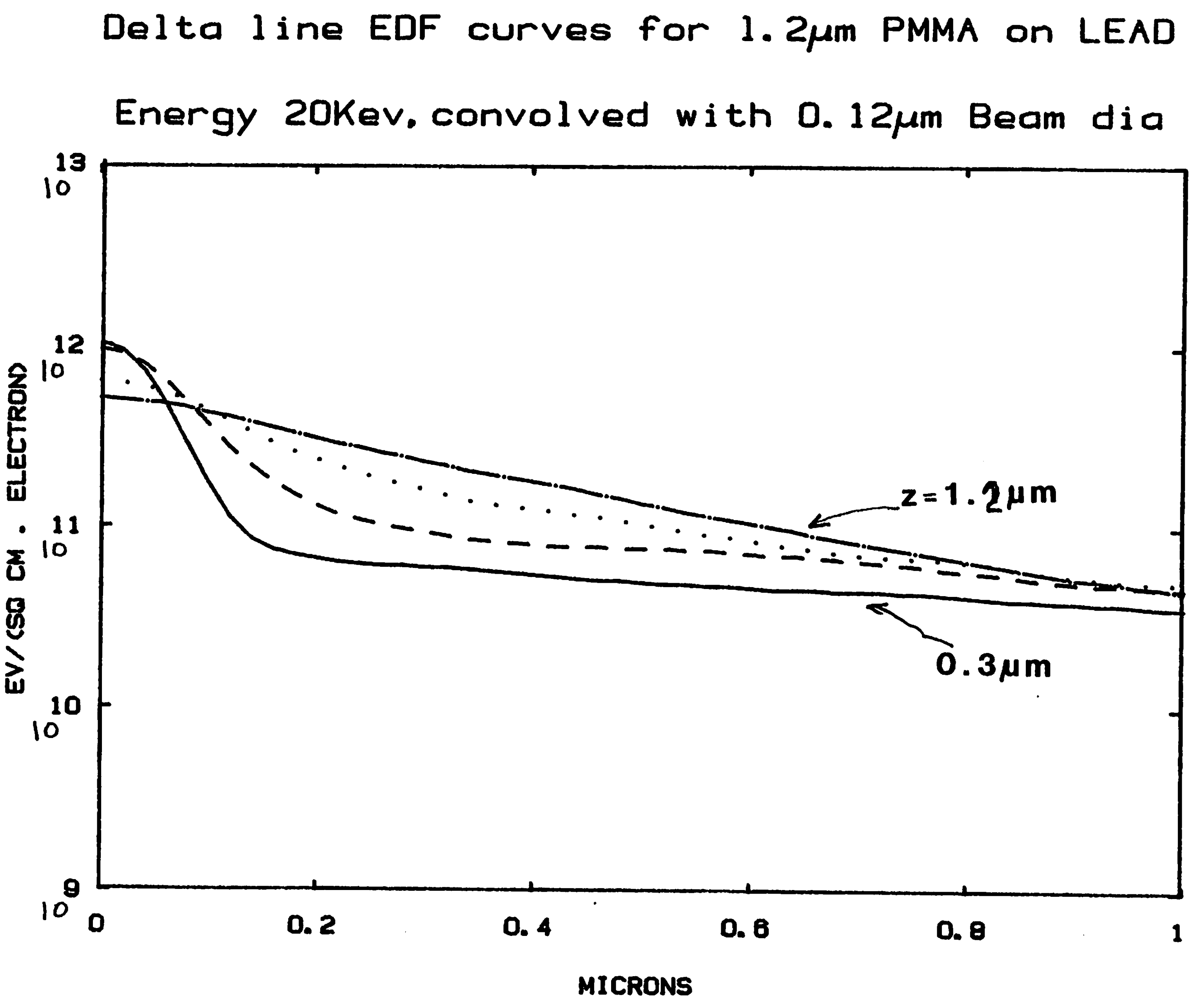

Delta line EDF curves for 1.2μm PMMA on LEAD
Energy 20Kev, convolved with 0.12μm Beam dia
10^{13}
10^{12}
10^{11}
10^{10}
10^{9}
EV/(SQ CM . ELECTRON)
z=1.2μm
0.3μm
0
0.2
0.4
0.6
0.8
1
MICRONS

Fig.13

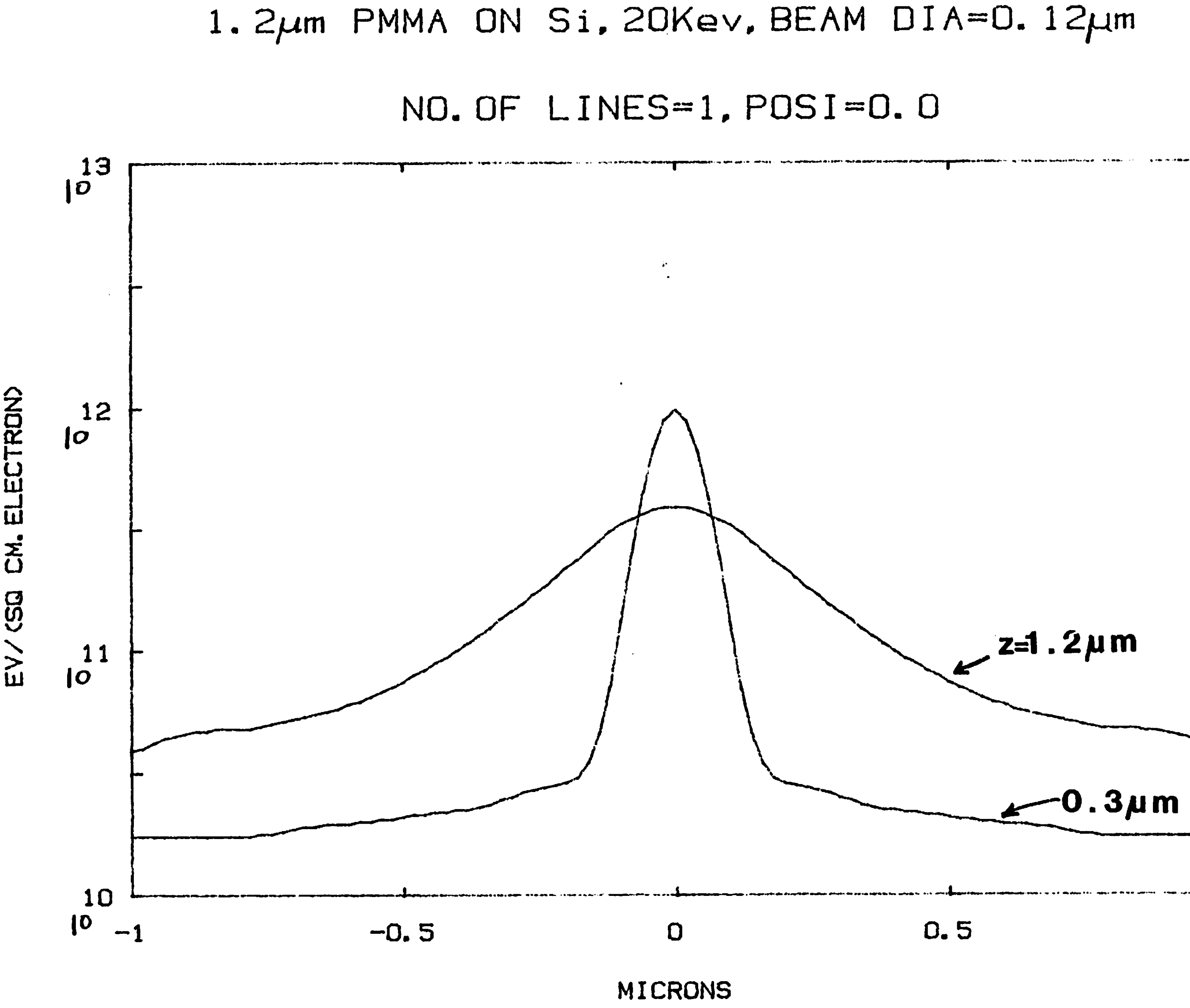

1.2μm PMMA ON Si, 20Kev, BEAM DIA=0.12μm
NO. OF LINES=1, POSI=0.0
EV/(SQ CM. ELECTRON)
10^{13}
10^{12}
10^{11}
10^{10}
-1
-0.5
0
0.5
1
MICRONS
z=1.2μm
0.3μm

Fig.14

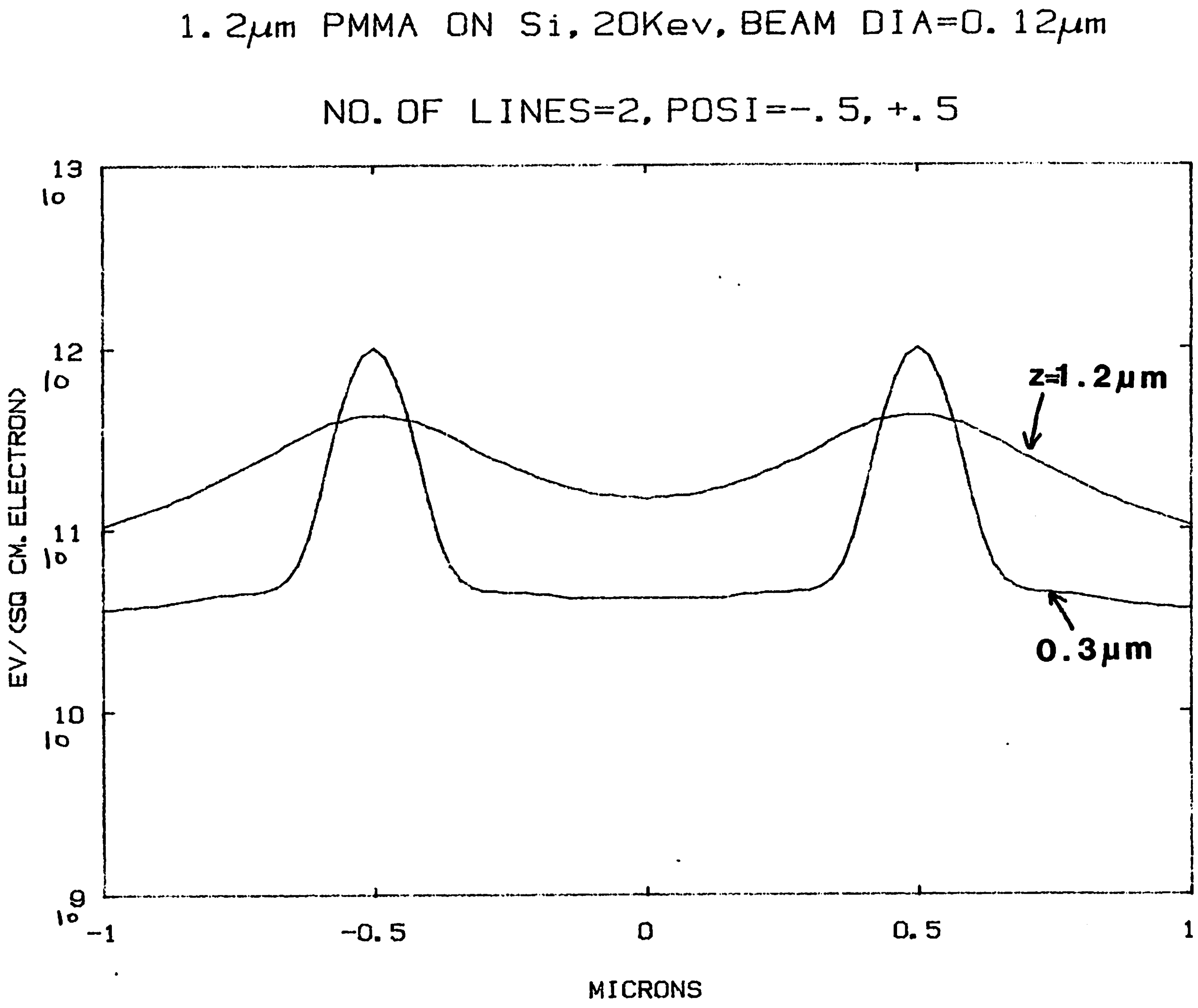

1.2µm PMMA ON Si, 20Kev, BEAM DIA=0.12µm
NO. OF LINES=2, POSI=-.5,+.5
10^{13}
10^{12}
10^{11}
10^{10}
10^{9}
EV/(SQ CM. ELECTRON)
z=1.2µm
0.3µm
-1
-0.5
0
0.5
1
MICRONS

Fig.15

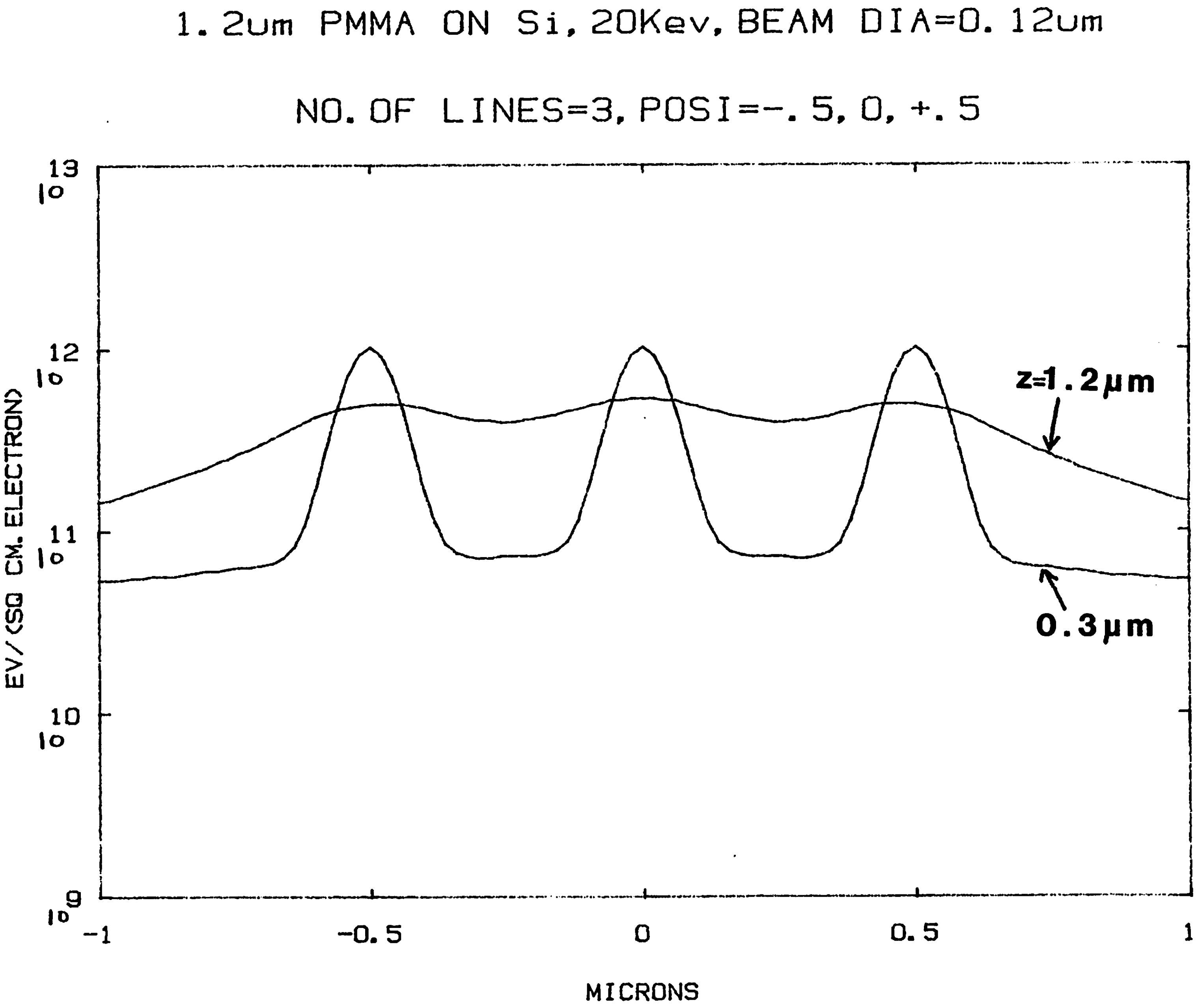

1.2um PMMA ON Si, 20Kev, BEAM DIA=0.12um
NO. OF LINES=3, POSI=-.5, 0, +.5
EV/(SQ CM. ELECTRON)
10^{13}
10^{12}
10^{11}
10^{10}
10^{9}
-1
-0.5
0
0.5
1
MICRONS
z=1.2µm
0.3µm

Fig.16

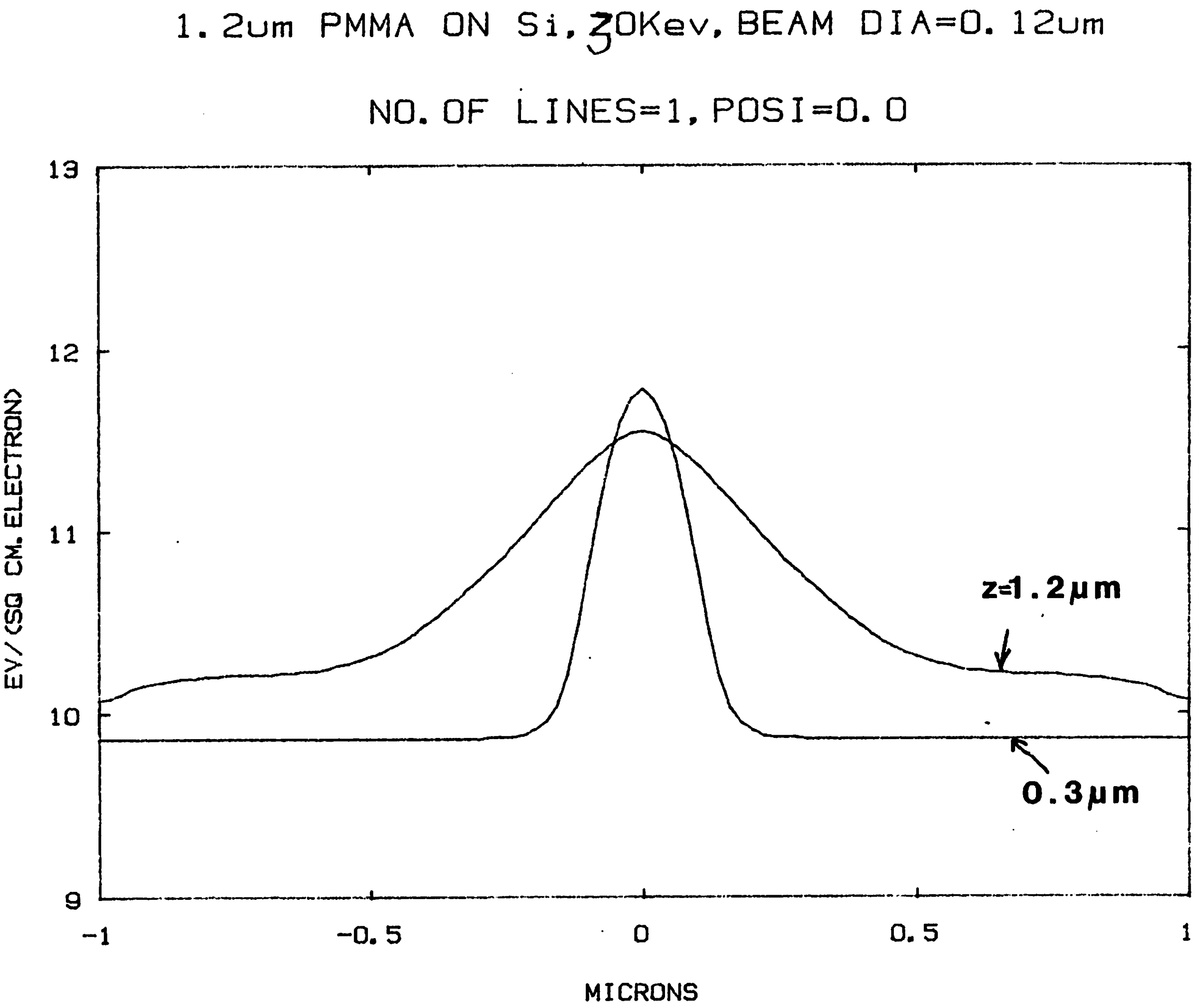

1.2um PMMA ON Si, 30Kev, BEAM DIA=0.12um
NO. OF LINES=1, POSI=0.0
13
12
11
10
9
EV/(SQ CM. ELECTRON)
-1
-0.5
0
0.5
1
MICRONS
z=1.2µm
0.3µm

Fig. 17

1.2μm PMMA ON Si, 30Kev, BEAM DIA=0.12μm

NO. OF LINES=2, POSI=-.5, +.5

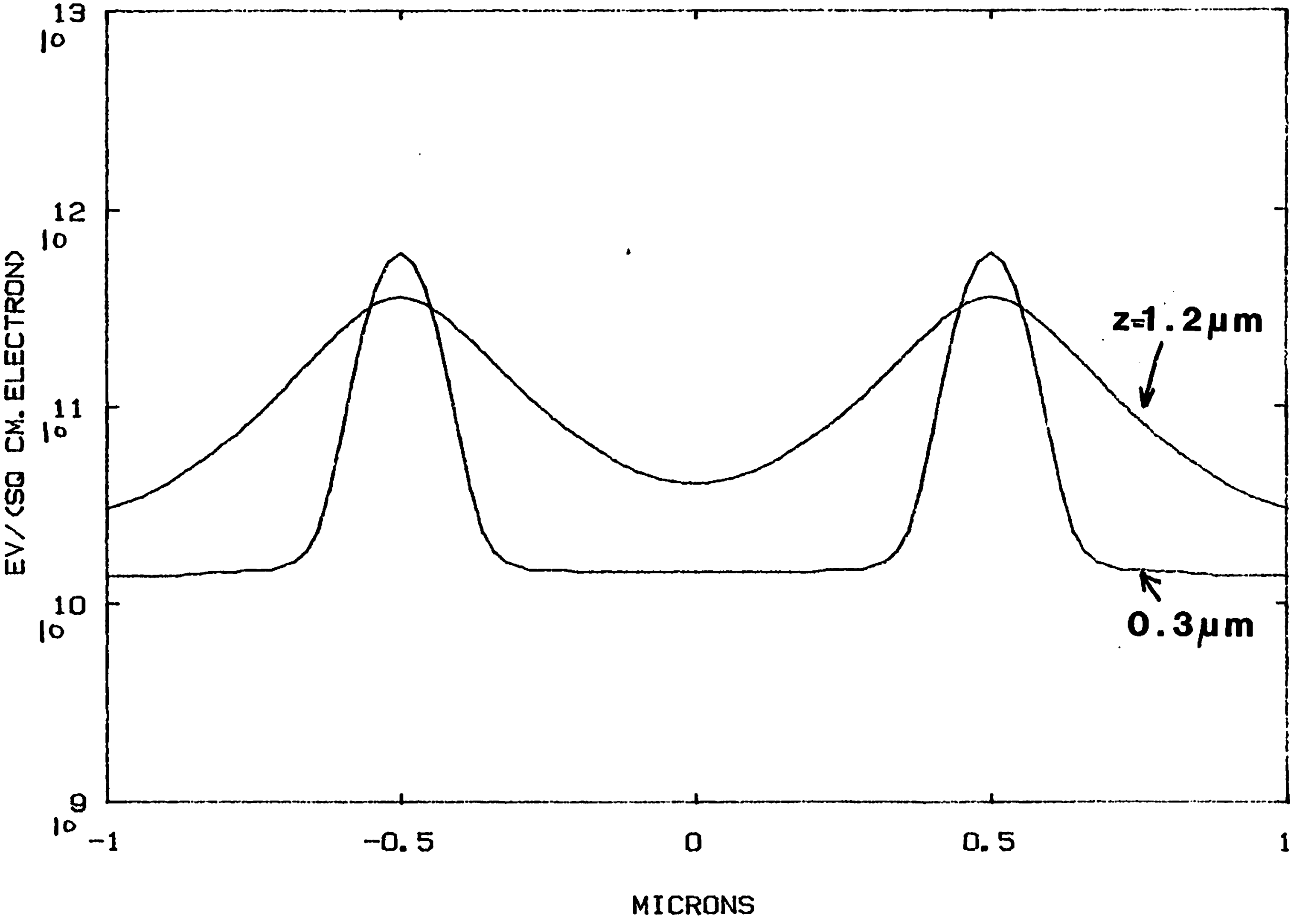


Fig18

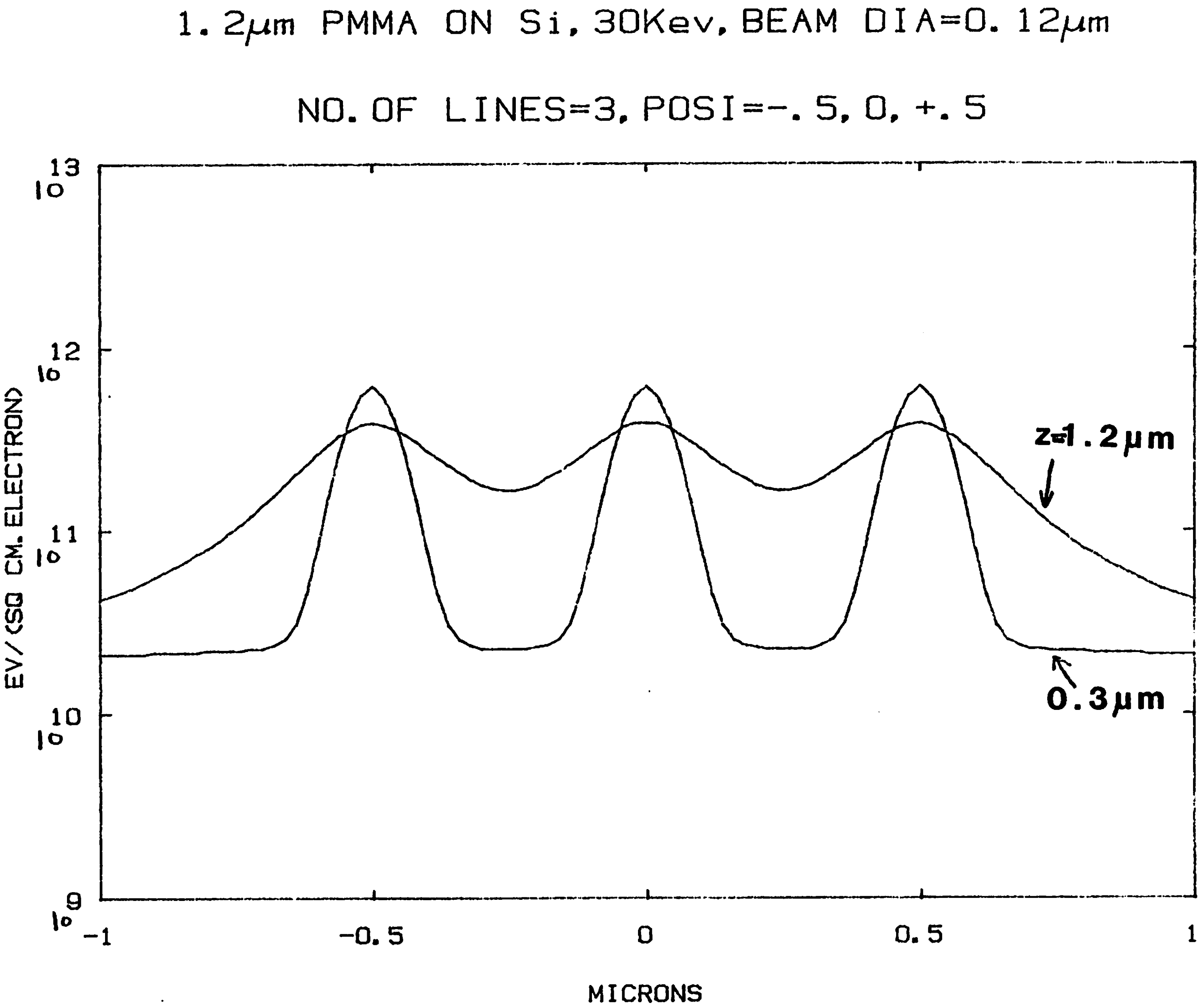

1.2μm PMMA ON Si, 30Kev, BEAM DIA=0.12μm
NO. OF LINES=3, POSI=-.5, 0, +.5
EV/(SQ CM. ELECTRON)
10^{13}
10^{12}
10^{11}
10^{10}
10^{9}
-1
-0.5
0
0.5
1
MICRONS
z=1.2μm
0.3μm

Fig. 19

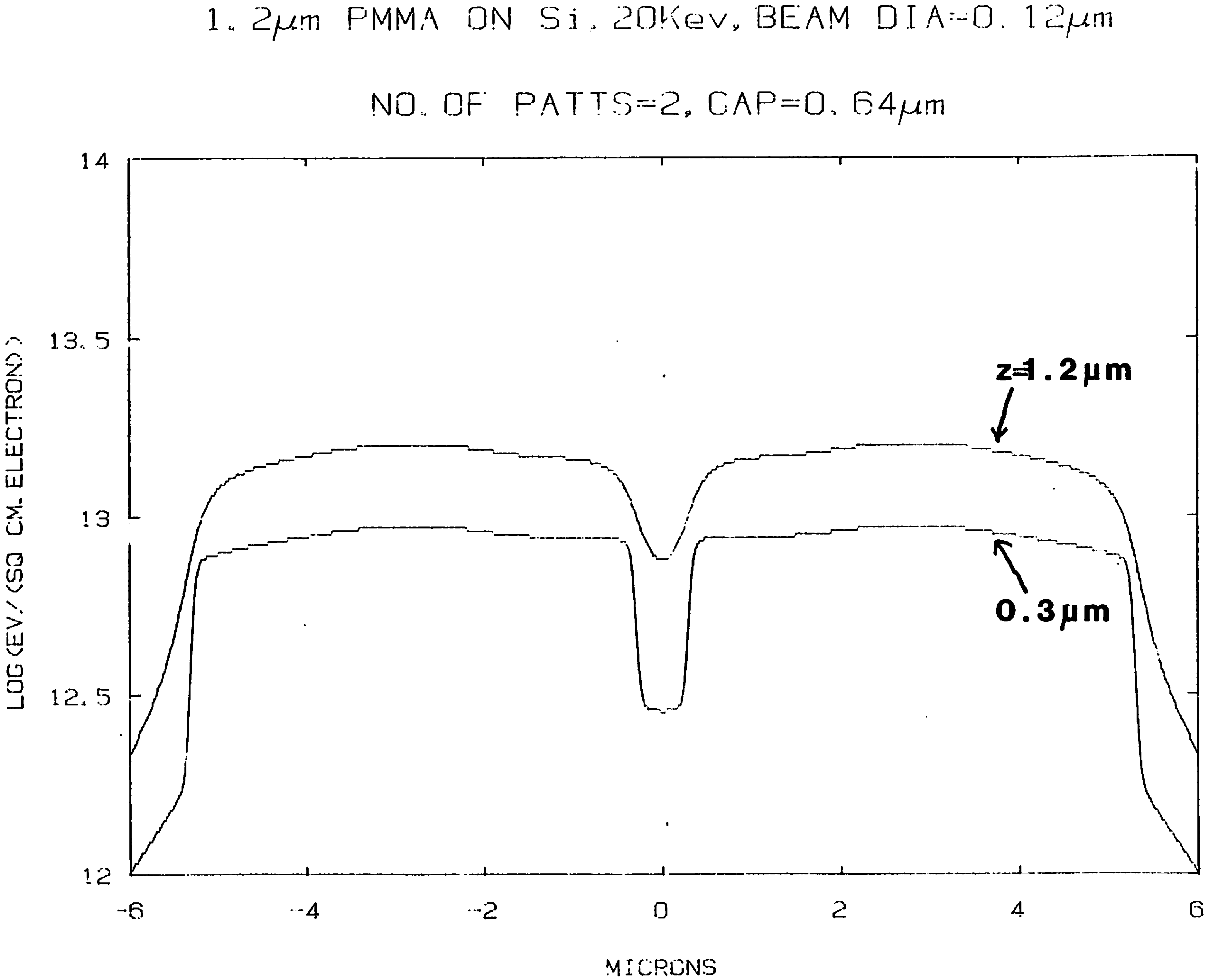

1. 2µm PMMA ON Si, 20Kev, BEAM DIA=0. 12µm
NO. OF PATTS=2, CAP=0. 64µm
LOG (EV./(SQ CM. ELECTRON))
14
13. 5
13
12. 5
12
-6
-4
-2
0
2
4
6
MICRONS
z=1.2µm
0.3µm

Fig.20

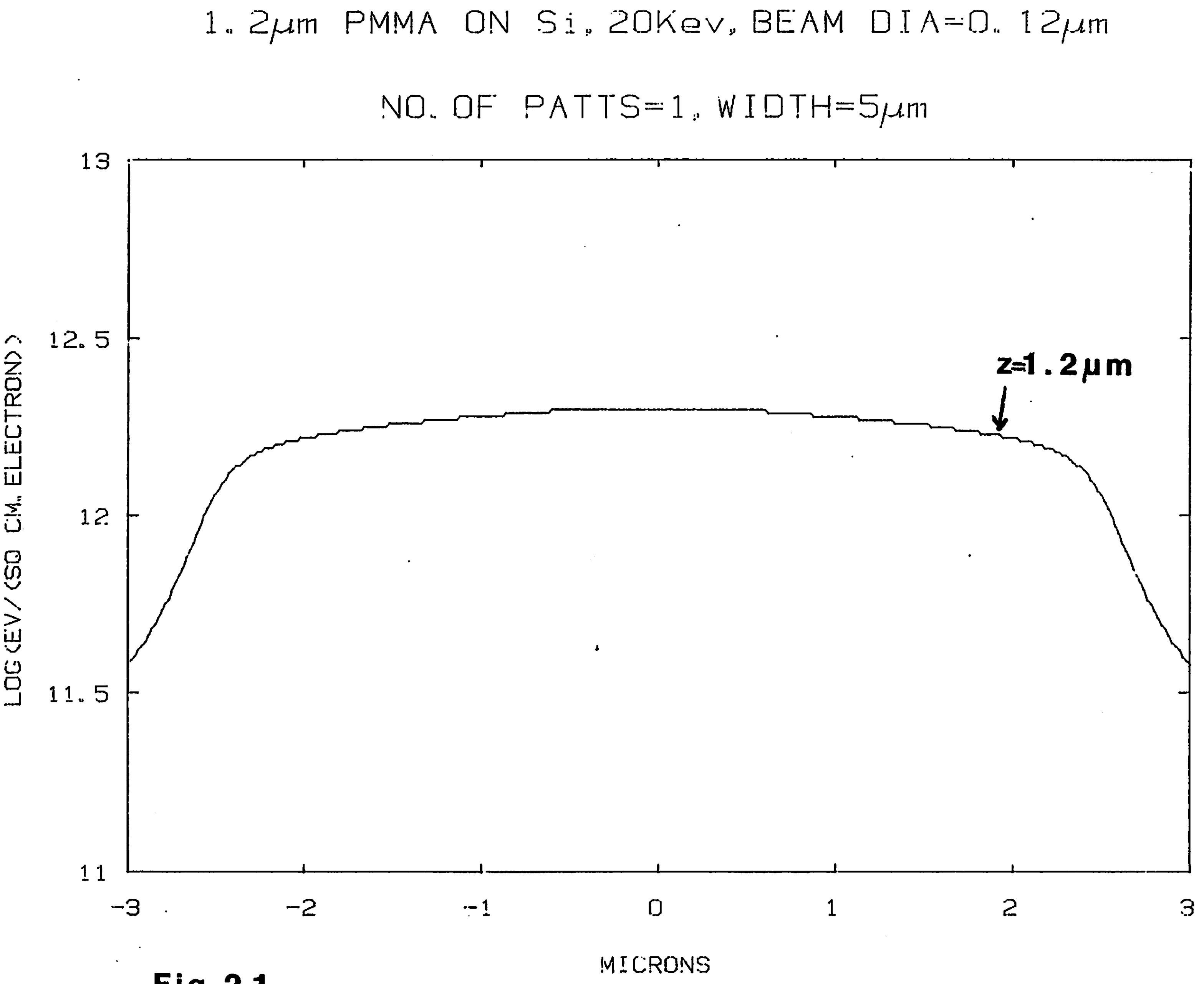

1. 2μm PMMA ON Si, 20Kev, BEAM DIA=0. 12μm
NO. OF PATTS=1, WIDTH=5μm
LOG(EV/(SQ CM. ELECTRON))
13
12.5
12
11.5
11
-3
-2
-1
0
1
2
3
MICRONS
z=1.2μm

Fig.21

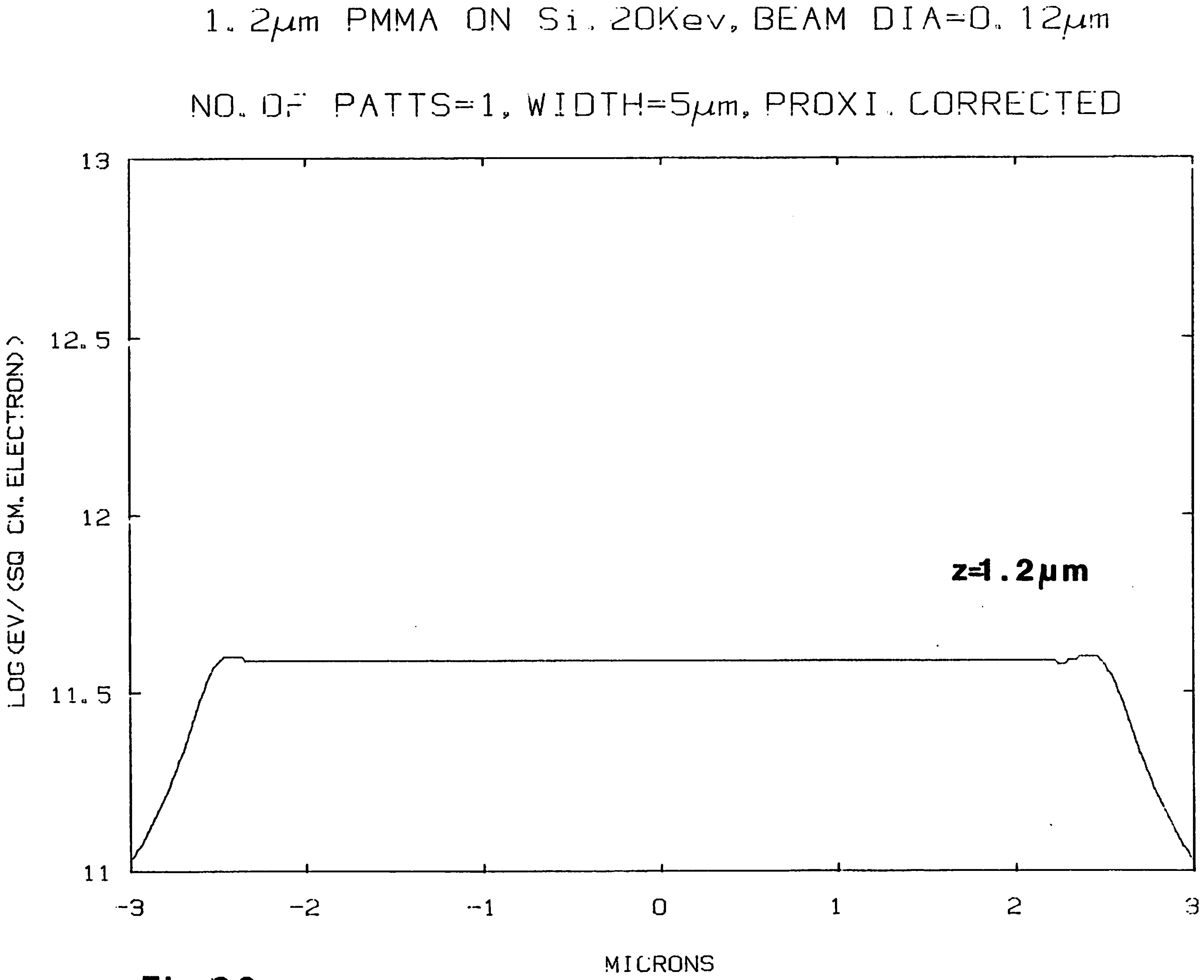

1.2μm PMMA ON Si.20Kev, BEAM DIA=0.12μm
NO. OF PATTS=1, WIDTH=5μm, PROXI. CORRECTED
LOG(EV/(SQ CM. ELECTRON))
13
12.5
12
11.5
11
z=1.2μm
-3
-2
-1
0
1
2
3
MICRONS

Fig.22

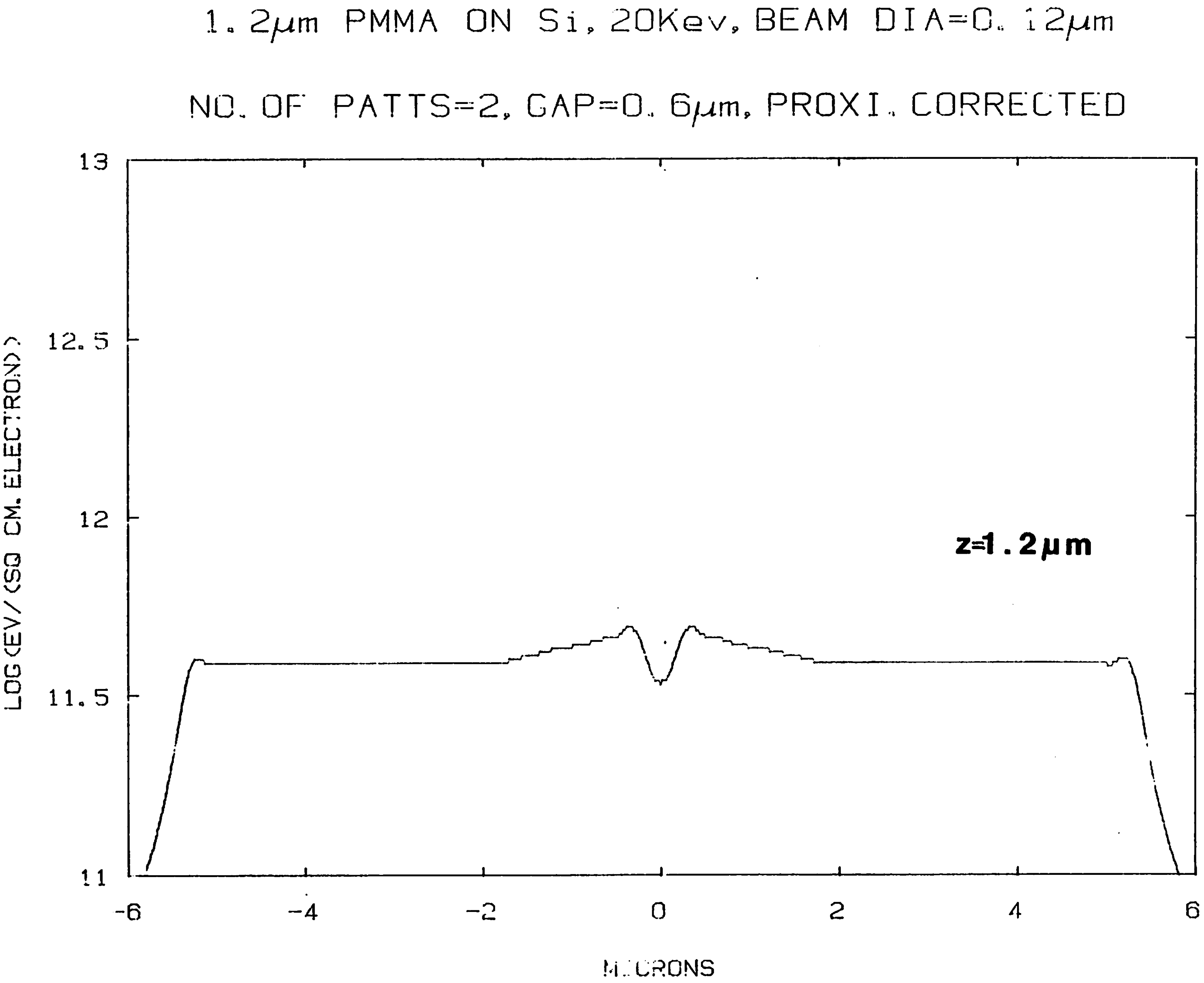


Fig.23 Only intrapattern Proxi. corrected

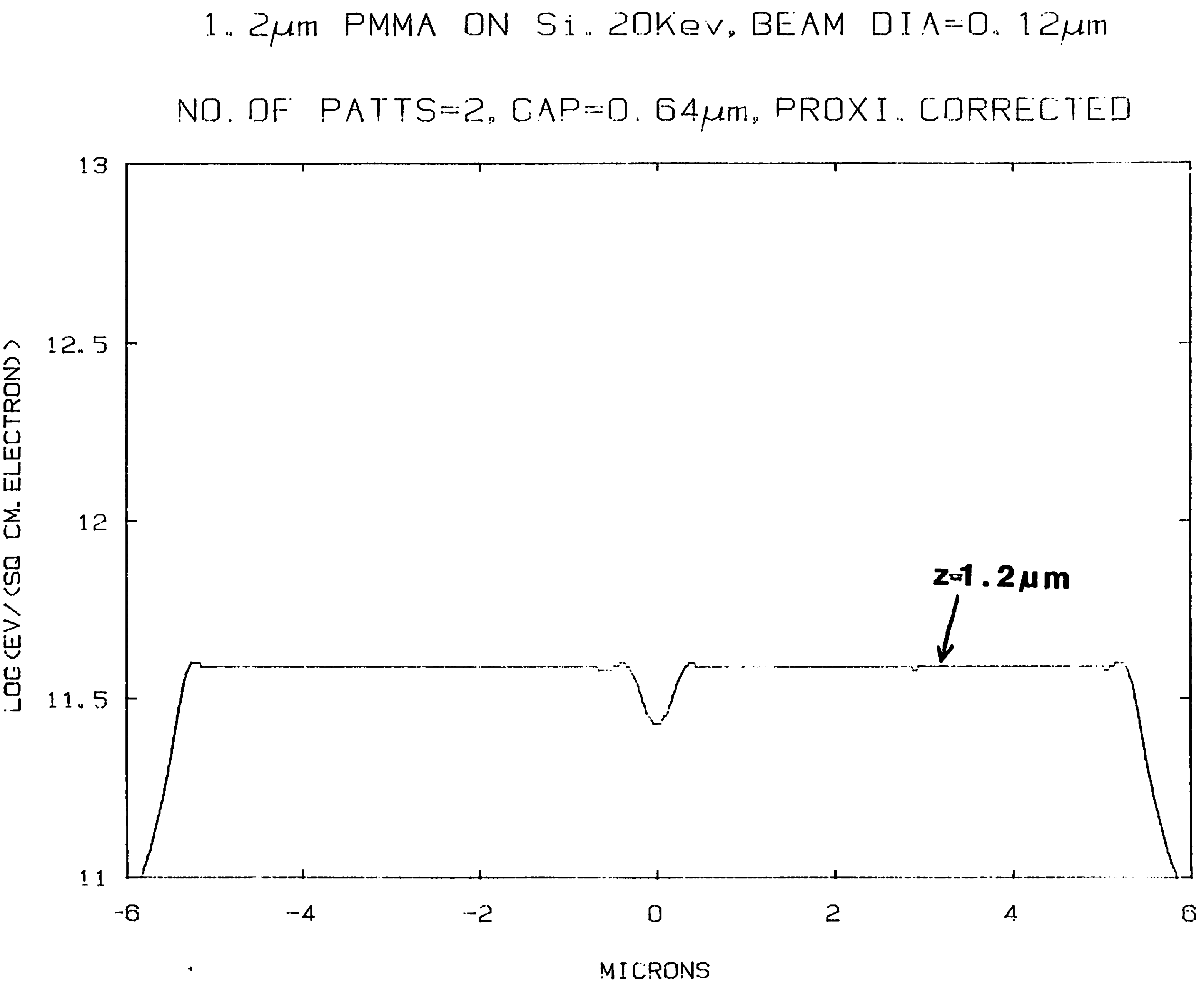

1.2μm PMMA ON Si. 20Kev, BEAM DIA=0.12μm
NO. OF PATTS=2, CAP=0.64μm, PROXI. CORRECTED
13
12.5
12
11.5
11
LOG(EV/(SQ CM. ELECTRON))
-6
-4
-2
0
2
4
6
MICRONS
z=1.2μm

Fig.24

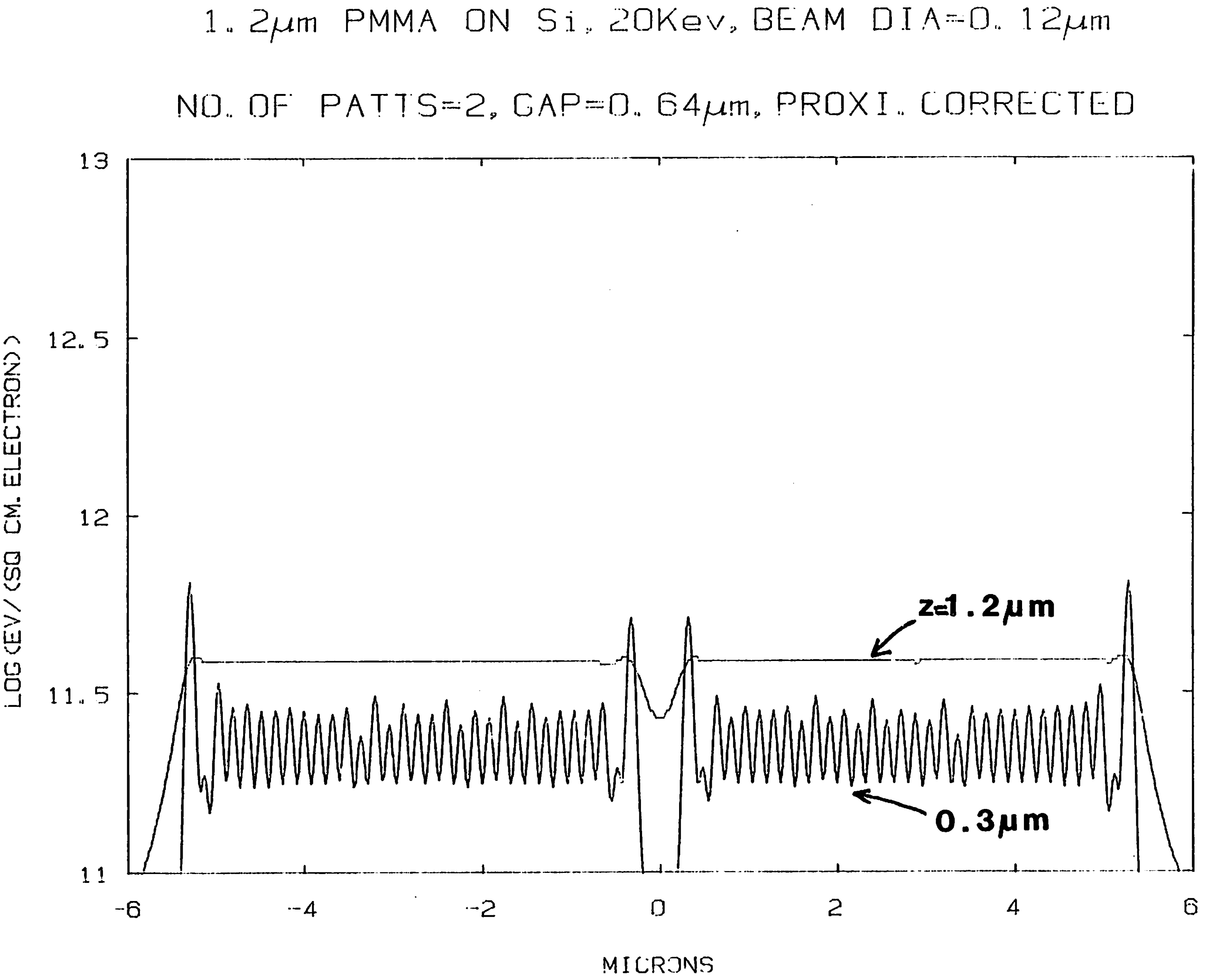

1. 2µm PMMA ON Si, 20Kev, BEAM DIA=0. 12µm
NO. OF PATTS=2, GAP=0. 64µm, PROXI. CORRECTED
13
12. 5
12
11. 5
11
LOG (EV/(SQ CM. ELECTRON))
-6
-4
-2
0
2
4
6
MICRONS
z=1.2µm
0.3µm

Fig.25

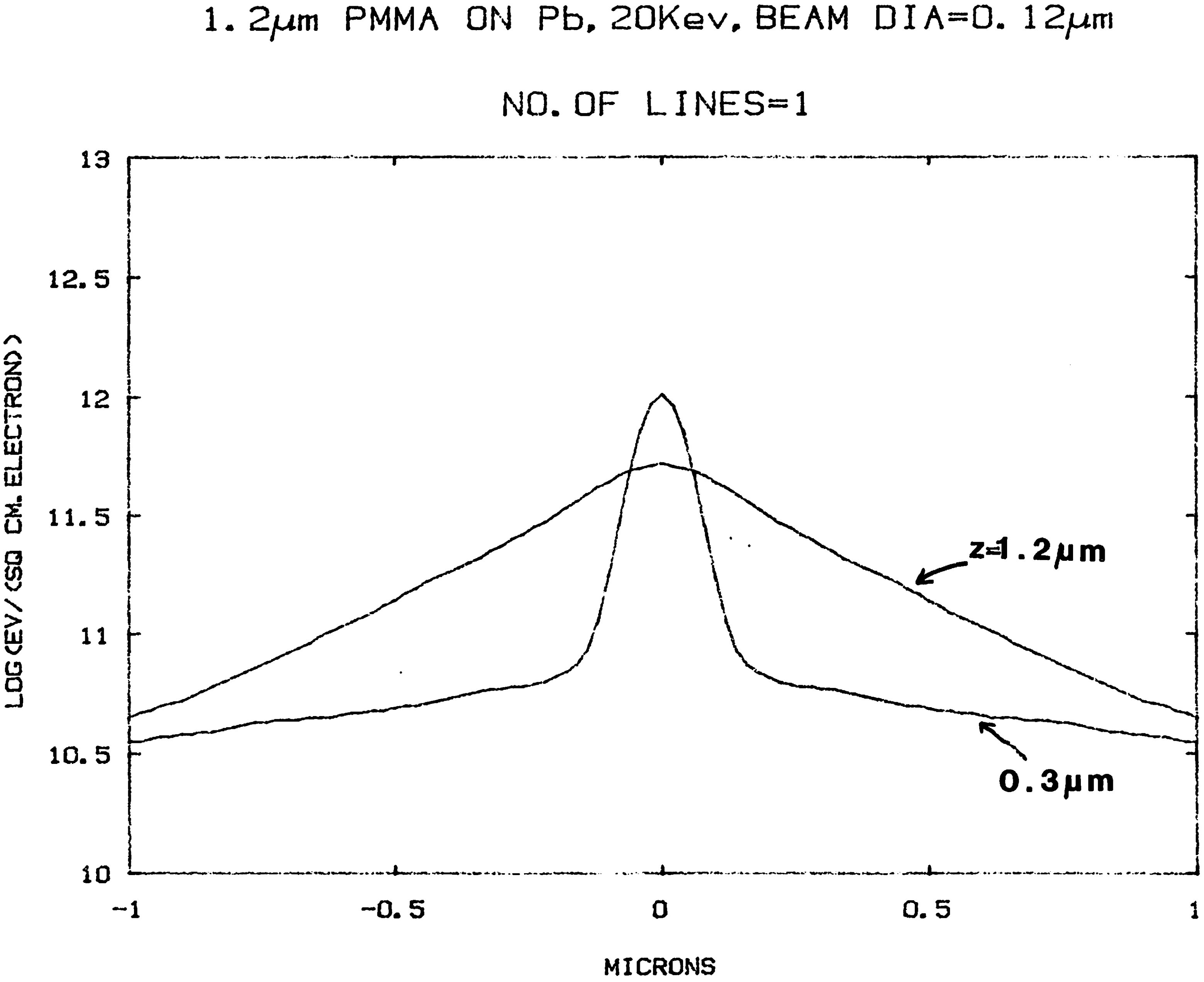

1.2µm PMMA ON Pb, 20Kev, BEAM DIA=0.12µm
NO. OF LINES=1
13
12.5
12
11.5
11
10.5
10
LOG(EV/(SQ CM. ELECTRON))
-1
-0.5
0
0.5
1
MICRONS
z=1.2µm
0.3µm

Fig.26

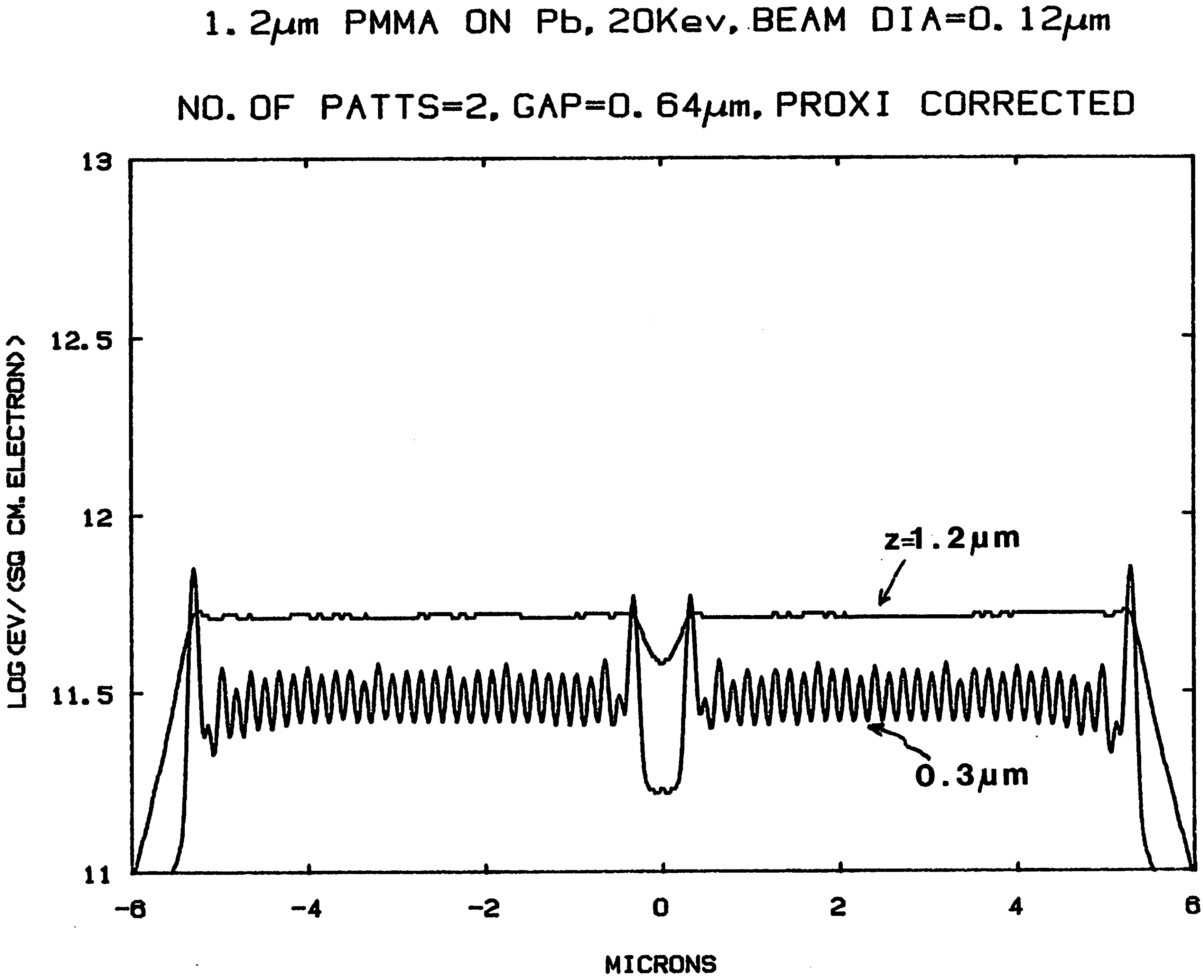

1.2µm PMMA ON Pb, 20Kev, BEAM DIA=0.12µm
NO. OF PATTS=2, GAP=0.64µm, PROXI CORRECTED
LOG(EV/(SQ CM. ELECTRON))
13
12.5
12
11.5
11
-6
-4
-2
0
2
4
6
MICRONS
z=1.2µm
0.3µm

Fig.27

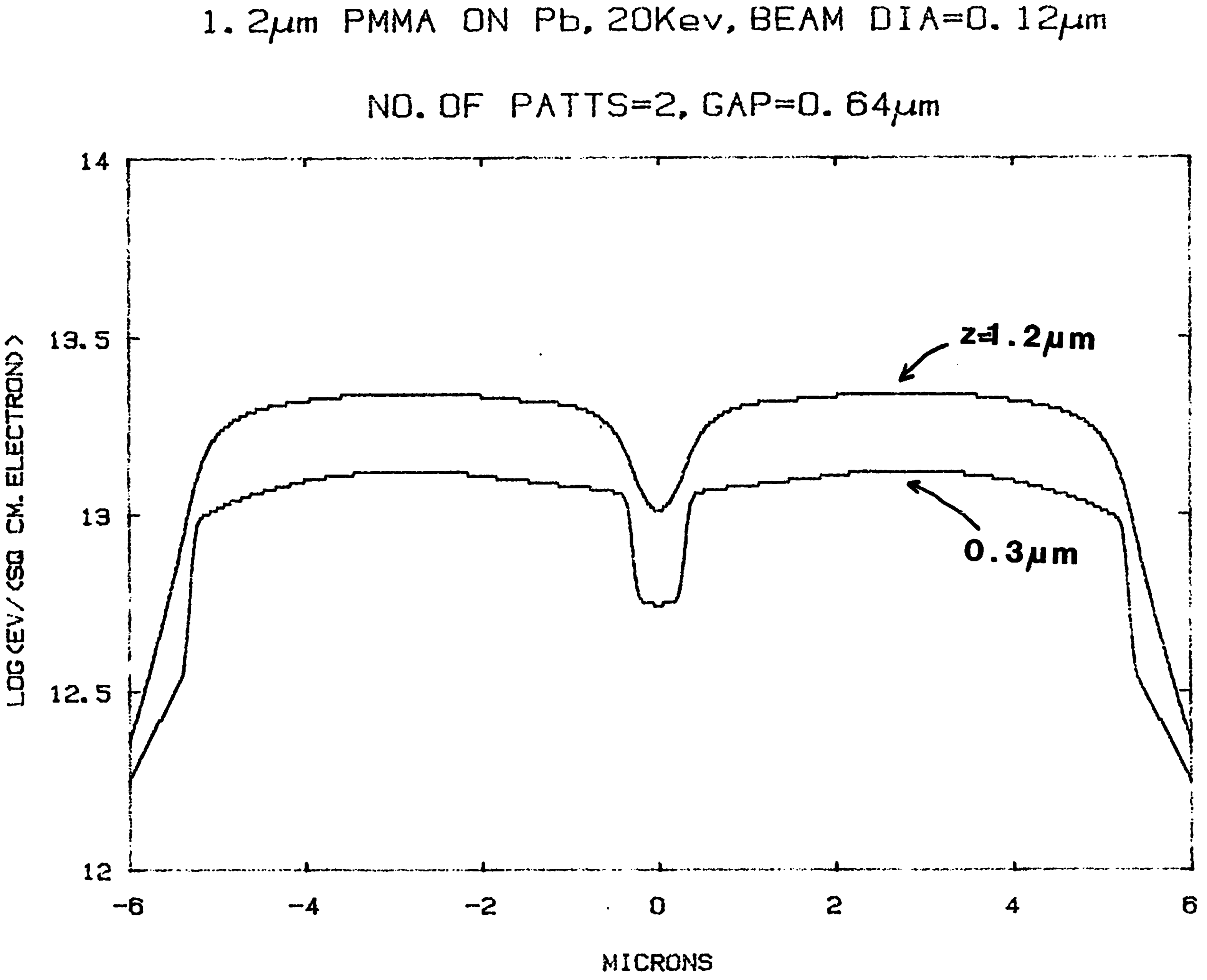

1.2µm PMMA ON Pb, 20Kev, BEAM DIA=0.12µm
NO. OF PATTS=2, GAP=0.64µm
14
13.5
13
12.5
12
LOG(EV/(SQ CM. ELECTRON)
z=1.2µm
0.3µm
-6
-4
-2
0
2
4
6
MICRONS

Fig.28

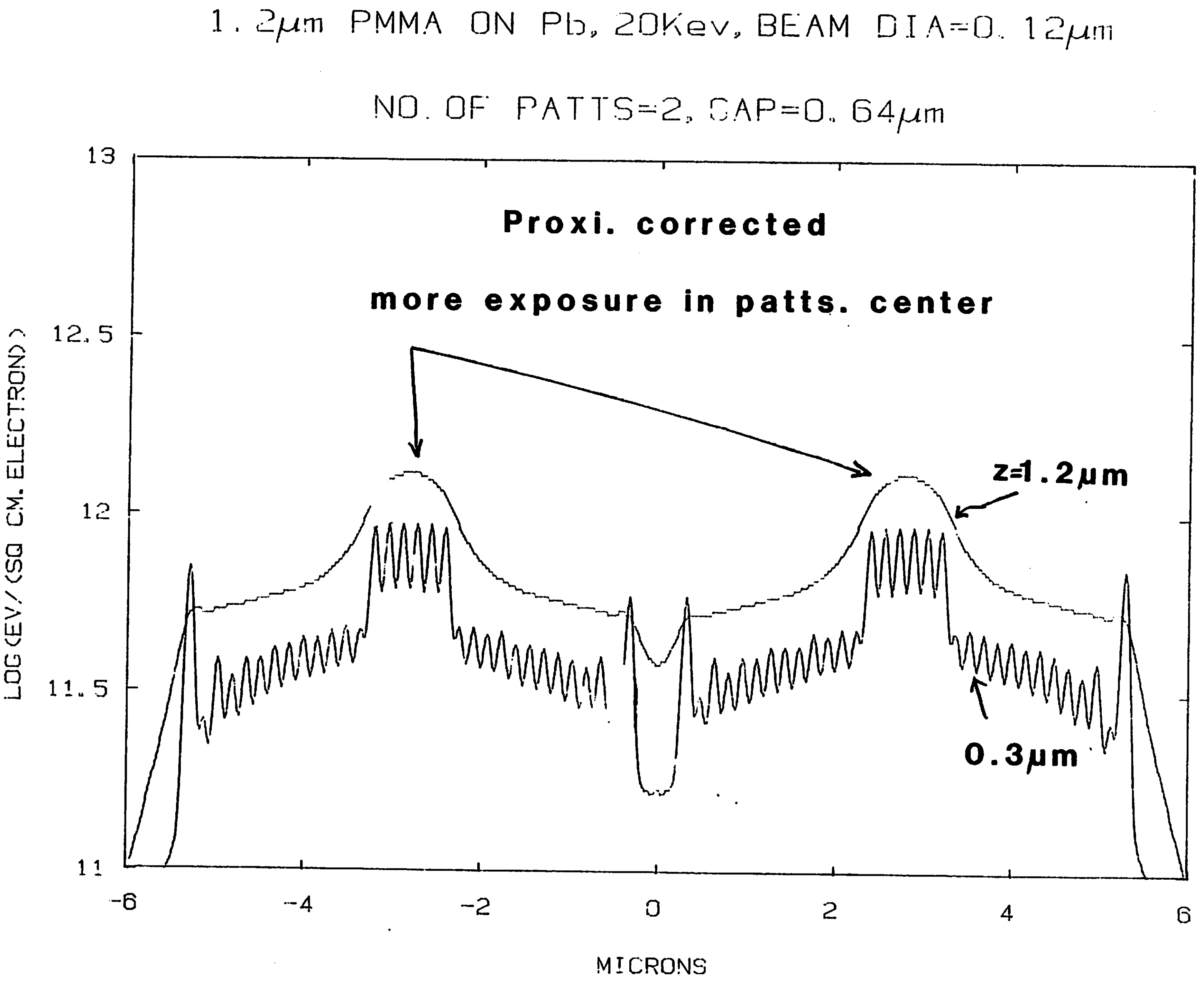

1.2μm PMMA ON Pb, 20Kev, BEAM DIA=0.12μm
NO. OF PATTS=2, GAP=0.64μm
Proxi. corrected
more exposure in patts. center
z=1.2μm
0.3μm
LOG(EV/(SQ CM. ELECTRON))
13
12.5
12
11.5
11
-6
-4
-2
0
2
4
6
MICRONS

Fig.29

APPENDIX

RANGE.F - File listing.

```
c    This routine finds electron range in PMMA,SILICON,GOLD
c    LEAD,NIOBIUM
c    To find range in PMMA do the following changes
c    aisi=65.66,const=7.5e-8,dens=1.18,eev=500.
c    To find range in SILICON do the following changes
c    aisi=172.25,const=5.6e-7,dens=2.33,eev=1000.
c    To find range in GOLD do the following changes
c    aisi=796.695,const=1.48403e-5,dens=19.32,eev=4000.
c    To find range in LEAD do the following changes
c    aisi=825.773,const=1.61224e-5,dens=11.3,eev=5000.
c    To find range in NIOBIUM do the following changes
c    aisi=429.196,const=3.927e-6,dens=8.57,eev=2500.
c    Also write appropriate name of the material in the
c    format NO.1
c    Keep initial energy eev >= 5*aisi
c    Also keep deev >= aisi
     write(6,1)
1    format("     electron range in SILICON—"/
    1" energy in EV         range in CMS"/)
     aconst=1.1658
     aisi=172.25
     const=5.6e-7
```

```
      dens=2.33
      eev=1000.
      deev=1000.
      aconst=1.1658
      do4i=1,60
      eevn=eev*aconst/aisi
      if(eevn.le.5.) go to 5
      j1=eevn
      j=j1-5
      f1=11.267
      e2=5.0
      de2=1.0
      do2k=1,j
      f1=f1+e2/alog(e2)
      e2=e2+1
 2    continue
      e3=eevn-float(j1)
      f1=f1+(e2/alog(e2))*e3
      range=const*f1/dens
      write(6,3)eev,range
 3    format(5x,f8.1,5x,e13.6)
      eev=eev+deev
 4    continue
      go to 7
 5    write(6,6)
 6    format(" initial energy eev is not > 5*aisi")
 7    end
```

RESIS1VAX.F - File listing.

```
C    This file is for simulation of electron trajectories.
     COMMON/PRD/MY,ZZZ,YYY
     COMMON A1,XYZ,TETA,FIE,STEP,EEV,ALPHAS,ITT,EEVL,YFL,IY,IRN
     COMMON/RAN1/IY1,IRN1,YFL1
     COMMON/RAN2/IY2,IRN2,YFL2
     COMMON/RAN3/IY3,IRN3,YFL3
     COMMON/RAN4/IY4,IRN4,YFL4
     COMMON/RAN5/IY5,IRN5,YFL5
     DIMENSION MY(50),ZZZ(50,900),YYY(50,900)
     DIMENSION ZZ(1001),YY(1001),ZZZ1(250000),YYY1(250000)
C     DIMENSION A(20,100)
     DIMENSION A1(3,3),XYZ(3)
     WRITE(6,1)
1    FORMAT(" Give the vlues of EEV1,FILM,AISI in units of"/
    1" KEV,MICRONS,EV (in free format in one line)"/
    2" Remember if sub. is Si then AISI=200."/
    3"         if sub. is Pb then AISI=900.")
     READ(5,*) EEV1,FILM,AISI
     INC=500
     INR=50
     IK=0
     IKK=0
     IY=0
     IRN=0
     DO9I=1,INR
     MY(I)=0
     DO9J=1,INC
     ZZZ(I,J)=0.0
     YYY(I,J)=0.0
9    CONTINUE
     FILM=FILM*1.E-4
     EEV1=EEV1*1.E+3
     NUMI=1
     NUMF=25
     DO34IT=NUMI,NUMF
     EEV=EEV1
     DO10I=1,3
     XYZ(I)=0.0
10   CONTINUE
     DO14I=1,3
     DO13J=1,3
     A1(I,J)=0.0
13   CONTINUE
     A1(I,I)=1.0
14   CONTINUE
     WRITE(6,16)IT
16   FORMAT(//"     ELECTRON TRAJECTORY NUMBER =",I5)
     ITT=1
     YY(ITT)=0.0
     ZZ(ITT)=0.0
50   IF(ZZ(ITT).LT.0.0) GO TO 60
```

```
      IF(ZZ(ITT).LT.FILM) GO TO 55
      IF(EEV.LT.AISI) GO TO 80
      IF(AISI.LT.300.) GO TO 3
      CALL LEAD
      CALL ANGLE
      CALL COORD
      GO TO 4
3     CALL SICON
      CALL ANGLE
      CALL COORD
4     ITT=ITT+1
      ZZ(ITT)=XYZ(3)
      YY(ITT)=XYZ(1)
      IF(ITT.EQ.900) GO TO 100
      GO TO 50
55    IF(EEV.LT.100.) GO TO 90
      CALL PMMA
      CALL ANGLE
      CALL COORD
      ITT=ITT+1
      ZZ(ITT)=XYZ(3)
      YY(ITT)=XYZ(1)
      IF(ITT.EQ.900) GO TO 100
      GO TO 50
60    WRITE(6,65)
65    FORMAT("  ELECTRON HAS COME BACK IN THE VACUUM")
      GO TO 30
```

```
80   WRITE(6,85)
85   FORMAT("  ELECTRON IS IN SUBSTRATE AND ITS ENERGY HAS BECOME",
    1" LESS THAN AISI")
     GO TO 30
90   WRITE(6,95)
95   FORMAT("  ELECTRON IS IN PMMA AND ITS ENERGY HAS BECOME LESS THAN
    1100.0 EV")
     GO TO 30
100  WRITE(6,110)
110  FORMAT(" NUMBER OF SCATTERING EVENTS HAVE BECOME MORE THAN 900")
30   WRITE(6,31)ITT,EEV1,EEV,EEVL,XYZ(3),XYZ(1),IRN3
31   FORMAT(" ITT=",I4," EEV1=",E16.6," EEV=",E16.6," EEVL=",E16.6,
    1" Z=",E16.6," Y=",E16.6," IRN3=",I5/)
     IK=IK+1
     MY(IK)=ITT
     NP=ITT
     IF(NP.GT.500) NP=500
     DO33J=1,NP
     ZZZ(IK,J)=ZZ(J)
     YYY(IK,J)=YY(J)
     IKK=IKK+1
     ZZZ1(IKK)=ZZ(J)*1.E+4
     YYY1(IKK)=YY(J)*1.E+4
33   CONTINUE
34   CONTINUE
     INP=50
     INF=500
```

```
C    REWIND 8
C    WRITE(8) ((MY(I),ZZZ(I,J),YYY(I,J),J=1,INF),I=1,INP)
     WRITE(6,201)(MY(I),I=1,INP)
201  FORMAT(10(I5,5X))
     OPEN(8,FILE='forscatt')
     REWIND 8
     DO203I=1,IKK
     WRITE(8,202)YYY1(I),ZZZ1(I)
202  FORMAT(2F8.4)
203  CONTINUE
     CLOSE(8)
     STOP
     END
     SUBROUTINE ANGLE
     COMMON A1,XYZ,TETA,FIE,STEP,EEV,ALPHAS,ITT,EEVL,YFL,IY,IRN
     COMMON/RAN3/IY3,IRN3,YFL3
     COMMON/RAN4/IY4,IRN4,YFL4
     DIMENSION XYZ(3),A1(3,3)
     CALL RNUM3
     R3=YFL3
     Z1=ALPHAS*ALPHAS
     Z3=R3
     Z2=(Z3*(1.+(2.*Z1))-Z1)/(Z3+Z1)
     TETA=Z2
     CALL RNUM3
     R4=YFL3
     FIE=2.*3.14159*R4
```

```
      RETURN
      END
      SUBROUTINE COORD
      COMMON A1,XYZ,TETA,FIE,STEP,EEV,ALPHAS,ITT,EEVL,YFL,IY,IRN
      DIMENSION XYZ(3),COA(3),CO(3)
      DIMENSION A1(3,3),A2(3,3),A3(3,3)
      STETA=SQRT(1.-TETA*TETA)
      CTETA=TETA
      SFIE=SIN(FIE)
      CFIE=COS(FIE)
      STETA2=STETA*STETA
      CTETA2=CTETA*CTETA
      CFIE2=CFIE*CFIE
      AMODX=SQRT(CTETA2+(STETA2*CFIE2))
      AMODY=SQRT((STETA2*SFIE*CFIE)**2
     1+(CTETA2+(STETA2*CFIE2))**2+(STETA*CTETA*SFIE)**2)
      CO(1)=STETA*CFIE*STEP
      CO(2)=STETA*SFIE*STEP
      CO(3)=CTETA*STEP
      DO53I=1,3
      COA(I)=0.0
      DO54J=1,3
      COA(I)=COA(I)+A1(I,J)*CO(J)
54    CONTINUE
      XYZ(I)=XYZ(I)+COA(I)
53    CONTINUE
      A2(1,1)=CTETA/AMODX
      A2(2,1)=0.
      A2(3,1)=-1.*STETA*CFIE/AMODX
      A2(1,2)=-1.*STETA2*SFIE*CFIE/AMODY
      A2(2,2)=(CTETA2+(STETA2*CFIE2))/AMODY
      A2(3,2)=-1.*STETA*CTETA*SFIE/AMODY
      A2(1,3)=STETA*CFIE
      A2(2,3)=STETA*SFIE
      A2(3,3)=CTETA
      DO55I=1,3
      A3(I,1)=0.0
      A3(I,2)=0.0
      A3(I,3)=0.0
      DO56J=1,3
      A3(I,1)=A3(I,1)+A1(I,J)*A2(J,1)
      A3(I,2)=A3(I,2)+A1(I,J)*A2(J,2)
      A3(I,3)=A3(I,3)+A1(I,J)*A2(J,3)
   56 CONTINUE
   55 CONTINUE
      DO57I=1,3
      DO58J=1,3
      A1(I,J)=A3(I,J)
   58 CONTINUE
   57 CONTINUE
      RETURN
      END
      SUBROUTINE RANDM
      COMMON A1,XYZ,TETA,FIE,STEP,EEV,ALPHAS,ITT,EEVL,YFL,IY,IRN
```

```
DIMENSION A1(3,3),XYZ(3)
DATA IA/12869/,MIC/25843/,M2/16384/,S/.3051758E-04/,IC/6925/
IRN=IRN+1
IY=IY*IA
IF(IY.GT.MIC) IY=(IY-M2)-M2
IY=IY+IC
IF(IY/2.GT.M2) IY=(IY-M2)-M2
IF(IY.LT.0) IY=(IY+M2)+M2
YFL=FLOAT(IY)*S
RETURN
END
SUBROUTINE RNUM1
COMMON/RAN1/IY1,IRN1,YFL1
DATA IA/12869/,MIC/25843/,M2/16384/,S/.3051758E-04/,IC/6925/
IRN1=IRN1+1
IY1=IY1*IA
IF(IY1.GT.MIC)IY1=(IY1-M2)-M2
IY1=IY1+IC
IF(IY1/2.GT.M2) IY1=(IY1-M2)-M2
IF(IY1.LT.0) IY1=(IY1+M2)+M2
YFL1=FLOAT(IY1)*S
RETURN
END
SUBROUTINE RNUM2
COMMON/RAN2/IY2,IRN2,YFL2
DATA IA/12869/,MIC/25843/,M2/16384/,S/.3051758E-04/,IC/6925/
IRN2=IRN2+1
```

```
IY2=IY2*IA
IF(IY2.GT.MIC)IY2=(IY2-M2)-M2
IY2=IY2+IC
IF(IY2/2.GT.M2) IY2=(IY2-M2)-M2
IF(IY2.LT.0) IY2=(IY2+M2)+M2
YFL2=FLOAT(IY2)*S
RETURN
END
SUBROUTINE RNUM3
COMMON/RAN3/IY3,IRN3,YFL3
DATA IA/843314861/,IC/453816693/,MIC/1693666955/,M2/1073741824/
DATA S/.4656613E-9/
IRN3=IRN3+1
IY3=IY3*IA
IF(IY3.GT.MIC)IY3=(IY3-M2)-M2
IY3=IY3+IC
IF(IY3/2.GT.M2) IY3=(IY3-M2)-M2
IF(IY3.LT.0) IY3=(IY3+M2)+M2
YFL3=FLOAT(IY3)*S
RETURN
END
SUBROUTINE RNUM4
COMMON/RAN4/IY4,IRN4,YFL4
DATA IA/12869/,MIC/25843/,M2/16384/,S/.3051758E-04/,IC/6925/
IRN4=IRN4+1
IY4=IY4*IA
IF(IY4.GT.MIC)IY4=(IY4-M2)-M2
```

```
IY4=IY4+IC
IF(IY4/2.GT.M2) IY4=(IY4-M2)-M2
IF(IY4.LT.0) IY4=(IY4+M2)+M2
YFL4=FLOAT(IY4)*S
RETURN
END
SUBROUTINE RNUM5
COMMON/RAN5/IY5,IRN5,YFL5
DATA IA/12869/,MIC/25843/,M2/16384/,S/.3051758E-04/,IC/6925/
IRN5=IRN5+1
IY5=IY5*IA
IF(IY5.GT.MIC)IY5=(IY5-M2)-M2
IY5=IY5+IC
IF(IY5/2.GT.M2) IY5=(IY5-M2)-M2
IF(IY5.LT.0) IY5=(IY5+M2)+M2
YFL5=FLOAT(IY5)*S
RETURN
END
SUBROUTINE PMMA
COMMON A1,XYZ,TETA,FIE,STEP,EEV,ALPHAS,ITT,EEVL,YFL,IY,IRN
COMMON/RAN2/IY2,IRN2,YFL2
COMMON/RAN1/IY1,IRN1,YFL1
COMMON/RAN3/IY3,IRN3,YFL3
DIMENSION XYZ(3),A1(3,3)
DATA DENC/3.54885E22/,DENH/5.67816E22/,DENO/1.41954E22/
DATA CONST/7.49559E-8/,DENS/1.18/
DATA C1/4.23388/,C2/2.33/,C3/4.65999/,C4/3.51761E15/
DATA C5/8.46345E18/,C6/4.03021E17/,C7/1.45087E19/
DATA C8/5.67816E22/,C9/3.54885E22/,ACONST/1.1658/,AIPMMA/65.66/
Z=SQRT(EEV)
ALPHAC=C1/Z
ALPHAH=C2/Z
ALPHAO=C3/Z
VEL=SQRT(C4*EEV)
Z=VEL*VEL
SIGMAC=C5/(Z*(ALPHAC**2)*(ALPHAC**2+1.0))
SIGMAH=C6/(Z*(ALPHAH**2)*(ALPHAH**2+1.0))
SIGMAO=C7/(Z*(ALPHAO**2)*(ALPHAO**2+1.0))
SIGMAC=SIGMAC/Z
SIGMAH=SIGMAH/Z
SIGMAO=SIGMAO/Z
ALAMDA=1.0/((DENC*SIGMAC)+(DENH*SIGMAH)+(DENO*SIGMAO))
CALL RNUM3
R1=YFL3
STEP=-ALAMDA*ALOG(R1)
PH=C8*SIGMAH*ALAMDA
PC=C9*SIGMAC*ALAMDA
CALL RNUM3
R2=YFL3
IF(R2.LT.PH) GO TO 100
Z=PH+PC
IF(R2.LE.Z) GO TO 101
ALPHAS=ALPHAO
NS=8
```

```
    GO TO 103
100 ALPHAS=ALPHAH
    NS=1
    GO TO 103
101 ALPHAS=ALPHAC
    NS=6
103 STEPN=DENS*STEP/CONST
    EEVN=EEV*ACONST/AIPMMA
    EEVI=EEV
    EEVN=EEVN-(STEPN*ALOG(EEVN)/EEVN)
    EEV=AIPMMA*EEVN/ACONST
    EEVL=EEVI-EEV
    RETURN
    END
    SUBROUTINE SICON
    COMMON A1,XYZ,TETA,FIE,STEP,EEV,ALPHAS,ITT,EEVL,YFL,IY,IRN
    COMMON/RAN5/IY5,IRN5,YFL5
    COMMON/RAN3/IY3,IRN3,YFL3
    DIMENSION A1(3,3),XYZ(3)
    DATA CONST/5.59612E-7/,DENS/2.33/,ACONST/1.1658/,AISI/172.253/
    DATA C1/5.61563/,C4/3.51761E15/
    DATA C3/4.23172E19/,C2/4.99878E22/
    ALPHAS=C1/(SQRT(EEV))
    Z=ALPHAS*ALPHAS
    VEL2=C4*EEV
    SIGMAS=C3/(VEL2*Z*(Z+1))
    SIGMAS=SIGMAS/(VEL2)
```

```
ALAMDA=1.0/(C2*SIGMAS)
CALL RNUM3
R5=YFL3
STEP=-ALAMDA*ALOG(R5)
STEPN=DENS*STEP/CONST
EEVI=EEV
EEVN=EEV*ACONST/AISI
EEVN=EEVN-(STEPN*ALOG(EEVN)/EEVN)
EEV=AISI*EEVN/ACONST
EEVL=EEVI-EEV
RETURN
END
SUBROUTINE LEAD
COMMON A1,XYZ,TETA,FIE,STEP,EEV,ALPHAS,ITT,EEVL,YFL,IY,IRN
COMMON/RAN5/IY5,IRN5,YFL5
COMMON/RAN3/IY3,IRN3,YFL3
DIMENSION A1(3,3),XYZ(3)
DATA CONST/1.61224e-5/,DENS/11.3/,ACONST/1.1658/,AISI/825.773/
DATA C1/10.1200/,C4/3.51761E15/
DATA C3/1.37000E21/,C2/3.30000E22/
ALPHAS=C1/(SQRT(EEV))
Z=ALPHAS*ALPHAS
VEL2=C4*EEV
SIGMAS=C3/(VEL2*Z*(Z+1))
SIGMAS=SIGMAS/(VEL2)
ALAMDA=1.0/(C2*SIGMAS)
CALL RNUM3
```

```
R5=YFL3

STEP=-ALAMDA*ALOG(R5)

STEPN=DENS*STEP/CONST

EEVI=EEV

EEVN=EEV*ACONST/AISI

EEVN=EEVN-(STEPN*ALOG(EEVN)/EEVN)

EEV=AISI*EEVN/ACONST

EEVL=EEVI-EEV

RETURN

END

BLOCK DATA RANDOM

COMMON/RAN1/IY1,IRN1,YFL1

COMMON/RAN2/IY2,IRN2,YFL2

COMMON/RAN3/IY3,IRN3,YFL3

COMMON/RAN4/IY4,IRN4,YFL4

COMMON/RAN5/IY5,IRN5,YFL5

DATA IY1,IY2,IY3,IY4,IY5/5*0/

DATA IRN1,IRN2,IRN3,IRN4,IRN5/5*0/

END
```

RESIS2VAX.F - File llisting.

```
C    This file is for energy deposition.
      COMMON/PRD1/INRMAX,INR1,INR2,NUMFF,AMZ,AMXY,AXY,AR
     C,FILM,EEV1,AISI
      COMMON/PRD2/AX,AY
      COMMON/RAN1/IY1,IRN1,YFL1
      COMMON/RAN2/IY2,IRN2,YFL2
      COMMON/RAN3/IY3,IRN3,YFL3
      COMMON/RAN4/IY4,IRN4,YFL4
      COMMON/RAN5/IY5,IRN5,YFL5
      COMMON A1,XYZ,TETA,FIE,STEP,EEV,ALPHAS,ITT,EEVL
      DIMENSION AX(20,150),AY(20,150),AXY(20,150),AR(20,150)
      DIMENSION A1(3,3),XYZ(3)
      WRITE(6,4)
4     FORMAT(" Have you prepared data file RANGEDATA if YES"/
     c" Then GIVE THE VALUES OF ENRGY ,RESIST THICKNESS, AISI"/
     1" IN UNITS OF KEV,MICRONS AND EV,IN FREE FORMAT"/
     2" IN ONE LINE AND SEPARATED BY CAMA AND PRESS RETURN"/
     3" REMEMBER"/
     4" IF SUBSTRATE IS SILICON THEN AISI=200"/
     5" IF SUBSTRATE IS LEAD    THEN AISI=900"/
     6" FOR EXAMPLE:-"/
     7"20.00,1.2,200.0")
      READ(5,*) EEV1,FILM,AISI
5     FORMAT(" BEAM ENERGY=",E13.6," KEV"/
     1" FILM THICKNESS=",E13.6," MICRONS"/
     2" LOWER ENERGY LIMIT=",E13.6," EV"/)
      write(6,5)eev1,film,aisi
      EEV1=EEV1*1.0E+3
      FILM=FILM*1.0E-4
c     (Lower energy limits for Si and Pb are 200 and 900 EV res.)
      AMZ=FILM/4.
      INR1=1
      INR2=4
      INRMAX=20
      NUMFF=5000
      AMXY=0.02E-4
      CALL SEG1
c      CALL SEG2
c      CALL SEG3
c      OPEN(10,FILE="impulse")
c      REWIND 10
c      READ(10,1)((AXY(I,J),J=1,150),I=1,20)
c      READ(10,1)((AR(I,J),J=1,150),I=1,20)
c1      FORMAT(10E13.6)
c      CALL SMOOT(AXY)
c      CALL SMOOT(AR)
c      REWIND 10
c      WRITE(10,1)((AXY(I,J),J=1,150),I=1,20)
c      WRITE(10,1)((AR(I,J),J=1,150),I=1,20)
      STOP
      END
      BLOCK DATA RANDOM
      COMMON/RAN1/IY1,IRN1,YFL1
      COMMON/RAN2/IY2,IRN2,YFL2
      COMMON/RAN3/IY3,IRN3,YFL3
      COMMON/RAN4/IY4,IRN4,YFL4
      COMMON/RAN5/IY5,IRN5,YFL5
      DATA IY1,IY2,IY3,IY4,IY5/5*0/
      DATA IRN1,IRN2,IRN3,IRN4,IRN5/5*0/
      END
      SUBROUTINE SEG1
C     PROGRAM SEG1(5)
      COMMON/PRD1/INRMAX,INR1,INR2,NUMFF,AMZ,AMXY,AXY,AR
     C,FILM,EEV1,AISI
      COMMON/PRD2/AX,AY
      COMMON A1,XYZ,TETA,FIE,STEP,EEV,ALPHAS,ITT,EEVL
      COMMON/RAN1/IY1,IRN1,YFL1
      COMMON/RAN2/IY2,IRN2,YFL2
      COMMON/RAN3/IY3,IRN3,YFL3
      COMMON/RAN4/IY4,IRN4,YFL4
      COMMON/RAN5/IY5,IRN5,YFL5
      DIMENSION A1(3,3),XYZ(3),RSI(30)
      DIMENSION AX(20,150),AY(20,150),AXY(20,150),AR(20,150)
      DIMENSION ZZ(901),YY(901)
      OPEN(9,FILE="rangedata")
      REWIND 9
      READ(9,5)(RSI(I),I=1,30)
5     FORMAT(8(4PF10.4)/8(4PF10.4)/8(4PF10.4)/6(4PF10.4))
```

```
      WRITE(6,5)(RSI(I),I=1,30)
      IRN=0
      NR=150
      NX=150
      NY=150
      NZ=20
      NBS=0
      AMESH=AMXY
      AMRR=AMXY
      DO9I=1,NZ
      DO9J=1,NX
      AX(I,J)=0.00
      AY(I,J)=0.0
      AXY(I,J)=0.0
      AR(I,J)=0.0
    9  CONTINUE
      NUMI=1
      NUMF=NUMFF
      ITOT=0
      DO34IT=NUMI,NUMF
      EEV=EEV1
      EEVL=0.0
      DO10I=1,3
      XYZ(I)=0.0
    10  CONTINUE
      DO14I=1,3
      DO13J=1,3
      A1(I,J)=0.0
13    CONTINUE
      A1(I,I)=1.0
14    CONTINUE
C     WRITE(6,16)IT
C16   FORMAT(//"   ELECTRON TRAJECTORY NUMBER =",I5)
      ITT=1
      YY(ITT)=0.0
      ZZ(ITT)=0.0
50    IF(ZZ(ITT).LT.0.0) GO TO 60
      IF(ZZ(ITT).LT.FILM) GO TO 55
      IRG=INT(EEV/1000.)+1
      DIST=XYZ(3)-FILM
      IF(DIST.GT.RSI(IRG)) GO TO 70
      IF(EEV.LT.AISI) GO TO 80
      IF(AISI.LE.300.) GO TO 4
      CALL LEAD
      CALL ANGLE
      CALL COORD
      GO TO 3
4     CALL SICON
      CALL ANGLE
      CALL COORD
3     ITT=ITT+1
      ZZ(ITT)=XYZ(3)
      YY(ITT)=XYZ(1)
      IF(ITT.EQ.901) GO TO 100
      GO TO 50
55    RAD=SQRT(XYZ(1)*XYZ(1)+XYZ(2)*XYZ(2))
      IMX=1+INT((ABS(XYZ(1))+0.5*AMXY)/AMXY)
      IMY=1+INT((ABS(XYZ(2))+0.5*AMXY)/AMXY)
      IMZ=1+INT((XYZ(3))/AMZ)
      IMR=1+INT((RAD+0.5*AMRR)/AMRR)
      IF(IMX.GT.NX) IMX=NX
      IF(IMY.GT.NY) IMY=NY
      IF(IMR.GT.NR) IMR=NR
      AX(IMZ,IMX)=AX(IMZ,IMX)+EEVL
      AY(IMZ,IMY)=AY(IMZ,IMY)+EEVL
      AR(IMZ,IMR)=AR(IMZ,IMR)+EEVL
      IF(EEV.LT.100.) GO TO 90
      CALL PMMA
      CALL ANGLE
      CALL COORD
      ITT=ITT+1
      ZZ(ITT)=XYZ(3)
      YY(ITT)=XYZ(1)
      IF(ITT.EQ.901) GO TO 100
      GO TO 50
C60   WRITE(6,65)
C65   FORMAT("  ELECTRON HAS COME BACK IN THE VACUUM")
60    NBS=NBS+1
      GO TO 30
C70   WRITE(6,75)
C75   FORMAT(" ELECTRON IS IN SUB. WITH DEPTH MORE THAN THE RANGE")
```

```
70   GO TO 30
C80  WRITE(6,85)
C85  FORMAT("  ELECTRON IS IN SUB. AND ITS ENERGY HAS BECOME LESS
C    1THAN AISI 200 OR 900 EV")
80   GO TO 30
90   WRITE(6,95)
95   FORMAT("  ELECTRON IS IN PMMA AND ITS ENERGY HAS BECOME LESS THAN
    1100.0 EV")
     GO TO 30
100  WRITE(6,110)
110  FORMAT(" NUMBER OF SCATTERING EVENTS HAVE BECOME MORE THAN 501")
C30  WRITE(6,31)ITT,EEV1,EEV,EEVL,XYZ(3),XYZ(2),IY,IRN
C31  FORMAT(" ITT=",I4," EEV1=",E16.6," EEV=",E16.6," EEVL=",E16.6,
C    1" Z=",E16.6," Y=",E16.6," IY=",I5," IRN=",I5/)
30   ITOT=ITOT+ITT
     IF(IT.EQ.NUMF)WRITE(6,31)IT,ITT,IRN3,ITOT
31   FORMAT(" IT=",I5,5X,"ITT=",I5,5X,"IRN3=",I15/
    1"  TOTAL NUMBER OF SCATTERING EVENTS  ="I15//)
34   CONTINUE
     IRN=IRN1+IRN2+IRN3+IRN4+IRN5
     WRITE(6,32)IRN
32   FORMAT(" TOTAL NUMBRE OF RANDOM NUMBERS USED=",I15/)
     DO35I=1,20
     DO35J=1,150
     AXY(I,J)=AX(I,J)+AY(I,J)
35   CONTINUE
     DO36I=1,20
```

```
      AX(I,1)=AX(I,1)/(AMXY*AMZ)
      AY(I,1)=AY(I,1)/(AMXY*AMZ)
      AXY(I,1)=AXY(I,1)/(2.*AMXY*AMZ)
      AR(I,1)=AR(I,1)*4./(3.14*AMRR*AMRR*AMZ)
      AX(I,1)=AX(I,1)/NUMFF
      AY(I,1)=AY(I,1)/NUMFF
      AXY(I,1)=AXY(I,1)/NUMFF
      AR(I,1)=AR(I,1)/NUMFF
      DO36J=2,150
      AXY(I,J)=(AXY(I,J)/(4.*AMZ*AMXY))
      AX(I,J)=(AX(I,J)/(2.*AMZ*AMXY))
      AR(I,J)=(AR(I,J)/(3.14*AMRR*FLOAT(J-1)*AMRR*AMZ))
      AY(I,J)=(AY(I,J)/(2.*AMZ*AMXY))
      AXY(I,J)=AXY(I,J)/NUMFF
      AX(I,J) =AX(I,J) /NUMFF
      AR(I,J) =AR(I,J) /NUMFF
      AY(I,J)=AY(I,J)/NUMFF
   36 CONTINUE
      CALL SMOOT(AXY)
      CALL SMOOT(AR)
      INR1=1
      INR2=4
      WRITE(6,96)EEV1,FILM
   96 FORMAT(" ENERGY DEPOSITION PERPANDICULAR TO DELTA LINE"/
     C"    ENERGY=",E13.6,"    PMMA THICKNESS=",E13.6/)
   37 FORMAT(//" ROW NUMBER I=",I2)
   38 FORMAT(10E13.6)
```

```
     DO41I=INR1,INR2
     WRITE(6,37)I
     WRITE(6,38)(AXY(I,J),J=1,50)
41   CONTINUE
     WRITE(6,97)EEV1,FILM
97   FORMAT(////" ENERGY DEPOSITION FOR SINGLE ELECTRON (RADIAL)"/
    C" ENERGY(EV)=", E13.6,"   PMMA THICKNESS(CM)=",E13.6/)
     DO42I=INR1,INR2
     WRITE(6,37)I
     WRITE(6,38)(AR(I,J),J=1,50)
42   CONTINUE
     WRITE(6,43)NBS,NUMFF
43   FORMAT(" NUMBER OF ELECTRONS CAME BACK INTO VAC=",I5/
    C" TOTAL NUMBER OF ELECTRONS TESTED=",I5/)
     WRITE(6,44)IRN1,IRN2,IRN3,IRN4,IRN5
44   FORMAT(//" IRN1=",I15," IRN2=",I15," IRN3=",I15," IRN4=",I15
    1," IRN5=",I15/)
     OPEN(10,FILE="impulse")
     REWIND 10
     WRITE(10,45)((AXY(I,J),J=1,150),I=1,INRMAX)
     WRITE(10,45)((AR(I,J),J=1,150),I=1,INRMAX)
45   FORMAT(10E13.6)
     CLOSE(10)
     RETURN
     END
     SUBROUTINE ANGLE
     COMMON A1,XYZ,TETA,FIE,STEP,EEV,ALPHAS,ITT,EEVL
```

```
COMMON/RAN1/IY1,IRN1,YFL1
COMMON/RAN2/IY2,IRN2,YFL2
COMMON/RAN3/IY3,IRN3,YFL3
COMMON/RAN4/IY4,IRN4,YFL4
COMMON/RAN5/IY5,IRN5,YFL5
DIMENSION XYZ(3),A1(3,3)
CALL   RNUM3
R3=YFL3
Z1=ALPHAS*ALPHAS
Z3=R3
Z2=(Z3*(1.+(2.*Z1))-Z1)/(Z3+Z1)
TETA=Z2
CALL RNUM3
R4=YFL3
FIE=2.*3.14159*R4
RETURN
END
SUBROUTINE COORD
COMMON A1,XYZ,TETA,FIE,STEP,EEV,ALPHAS,ITT,EEVL
DIMENSION XYZ(3),COA(3),CO(3)
DIMENSION A1(3,3),A2(3,3),A3(3,3)
STETA=SQRT(1.-TETA*TETA)
CTETA=TETA
SFIE=SIN(FIE)
CFIE=COS(FIE)
STETA2=STETA*STETA
CTETA2=CTETA*CTETA
```

```
      CFIE2=CFIE*CFIE
      AMODX=SQRT(CTETA2+(STETA2*CFIE2))
      AMODY=SQRT((STETA2*SFIE*CFIE)**2
     1+(CTETA2+(STETA2*CFIE2))**2+(STETA*CTETA*SFIE)**2)
      CO(1)=STETA*CFIE*STEP
      CO(2)=STETA*SFIE*STEP
      CO(3)=CTETA*STEP
      DO53I=1,3
      COA(I)=0.0
      DO54J=1,3
      COA(I)=COA(I)+A1(I,J)*CO(J)
   54 CONTINUE
      XYZ(I)=XYZ(I)+COA(I)
   53 CONTINUE
      A2(1,1)=CTETA/AMODX
      A2(2,1)=0.
      A2(3,1)=-1.*STETA*CFIE/AMODX
      A2(1,2)=-1.*STETA2*SFIE*CFIE/AMODY
      A2(2,2)=(CTETA2+(STETA2*CFIE2))/AMODY
      A2(3,2)=-1.*STETA*CTETA*SFIE/AMODY
      A2(1,3)=STETA*CFIE
      A2(2,3)=STETA*SFIE
      A2(3,3)=CTETA
      DO55I=1,3
      A3(I,1)=0.0
      A3(I,2)=0.0
      A3(I,3)=0.0
      DO56J=1,3
      A3(I,1)=A3(I,1)+A1(I,J)*A2(J,1)
      A3(I,2)=A3(I,2)+A1(I,J)*A2(J,2)
      A3(I,3)=A3(I,3)+A1(I,J)*A2(J,3)
   56 CONTINUE
   55 CONTINUE
      DO57I=1,3
      DO58J=1,3
      A1(I,J)=A3(I,J)
   58 CONTINUE
   57 CONTINUE
      RETURN
      END
      SUBROUTINE RNUM1
      COMMON/RAN1/IY1,IRN1,YFL1
      DATA IA/12869/,MIC/25843/,M2/16384/,S/.3051758E-04/,IC/6925/
      IRN1=IRN1+1
      IY1=IY1*IA
      IF(IY1.GT.MIC) IY1=(IY1-M2)-M2
      IY1=IY1+IC
      IF(IY1/2.GT.M2) IY1=(IY1-M2)-M2
      IF(IY1.LT.0) IY1=(IY1+M2)+M2
      YFL1=FLOAT(IY1)*S
      RETURN
      END
      SUBROUTINE RNUM2
      COMMON/RAN2/IY2,IRN2,YFL2
```

```
DATA IA/12869/,MIC/25843/,M2/16384/,S/.3051758E-04/,IC/6925/
IRN2=IRN2+1
IY2=IY2*IA
IF(IY2.GT.MIC) IY2=(IY2-M2)-M2
IY2=IY2+IC
IF(IY2/2.GT.M2) IY2=(IY2-M2)-M2
IF(IY2.LT.0) IY2=(IY2+M2)+M2
YFL2=FLOAT(IY2)*S
RETURN
END
SUBROUTINE RNUM3
COMMON/RAN3/IY3,IRN3,YFL3
DATA IA/843314861/,IC/453816693/,MIC/1693666955/,M2/1073741824/
DATA S/.4656613E-9/
IRN3=IRN3+1
IY3=IY3*IA
IF(IY3.GT.MIC) IY3=(IY3-M2)-M2
IY3=IY3+IC
IF(IY3/2.GT.M2) IY3=(IY3-M2)-M2
IF(IY3.LT.0) IY3=(IY3+M2)+M2
YFL3=FLOAT(IY3)*S
RETURN
END
SUBROUTINE RNUM4
COMMON/RAN4/IY4,IRN4,YFL4
DATA IA/12869/,MIC/25843/,M2/16384/,S/.3051758E-04/,IC/6925/
IRN4=IRN4+1
```

```
IY4=IY4*IA

IF(IY4.GT.MIC) IY4=(IY4-M2)-M2

IY4=IY4+IC

IF(IY4/2.GT.M2) IY4=(IY4-M2)-M2

IF(IY4.LT.0) IY4=(IY4+M2)+M2

YFL4=FLOAT(IY4)*S

RETURN

END

SUBROUTINE RNUM5

COMMON/RAN5/IY5,IRN5,YFL5

DATA IA/12869/,MIC/25843/,M2/16384/,S/.3051758E-04/,IC/6925/

IRN5=IRN5+1

IY5=IY5*IA

IF(IY5.GT.MIC) IY5=(IY5-M2)-M2

IY5=IY5+IC

IF(IY5/2.GT.M2) IY5=(IY5-M2)-M2

IF(IY5.LT.0) IY5=(IY5+M2)+M2

YFL5=FLOAT(IY5)*S

RETURN

END

SUBROUTINE PMMA

COMMON A1,XYZ,TETA,FIE,STEP,EEV,ALPHAS,ITT,EEVL

COMMON/RAN1/IY1,IRN1,YFL1

COMMON/RAN2/IY2,IRN2,YFL2

COMMON/RAN3/IY3,IRN3,YFL3

COMMON/RAN4/IY4,IRN4,YFL4

COMMON/RAN5/IY5,IRN5,YFL5
```

```
DIMENSION XYZ(3),A1(3,3)

DATA DENC/3.54885E22/,DENH/5.67816E22/,DENO/1.41954E22/

DATA CONST/7.49559E-8/,DENS/1.18/

DATA C1/4.23388/,C2/2.33/,C3/4.65999/,C4/3.51761E15/

DATA C5/8.46345E18/,C6/4.03021E17/,C7/1.45087E19/

DATA C8/5.67816E22/,C9/3.54885E22/,ACONST/1.1658/,AIPMMA/65.66/

Z=SQRT(EEV)

ALPHAC=C1/Z

ALPHAH=C2/Z

ALPHAO=C3/Z

VEL=SQRT(C4*EEV)

Z=VEL*VEL

SIGMAC=C5/(Z*(ALPHAC**2)*(ALPHAO**2+1.0))

SIGMAH=C6/(Z*(ALPHAH**2)*(ALPHAH**2+1.0))

SIGMAO=C7/(Z*(ALPHAC**2)*(ALPHAO**2+1.0))

SIGMAC=SIGMAC/Z

SIGMAH=SIGMAH/Z

SIGMAO=SIGMAO/Z

ALAMDA=1.0/((DENC*SIGMAC)+(DENH*SIGMAH)+(DENO*SIGMAO))

CALL RNUM3

R1=YFL3

STEP=-ALAMDA*ALOG(R1)

PH=C8*SIGMAH*ALAMDA

PC=C9*SIGMAC*ALAMDA

CALL RNUM3

R2=YFL3

IF(R2.LT.PH) GO TO 100
```

```
    Z=PH+PC
    IF(R2.LE.Z) GO TO 101
    ALPHAS=ALPHAO
    NS=8
    GO TO 103
100 ALPHAS=ALPHAH
    NS=1
    GO TO 103
101 ALPHAS=ALPHAC
    NS=6
103 STEPN=DENS*STEP/CONST
    EEVN=EEV*ACONST/AIPMMA
    EEVI=EEV
    EEVN=EEVN-(STEPN*ALOG(EEVN)/EEVN)
    EEV=AIPMMA*EEVN/ACONST
    EEVL=EEVI-EEV
    RETURN
    END
    SUBROUTINE SICON
    COMMON/RAN1/IY1,IRN1,YFL1
    COMMON/RAN2/IY2,IRN2,YFL2
    COMMON/RAN3/IY3,IRN3,YFL3
    COMMON/RAN4/IY4,IRN4,YFL4
    COMMON/RAN5/IY5,IRN5,YFL5
    COMMON A1,XYZ,TETA,FIE,STEP,EEV,ALPHAS,ITT,EEVL
    DIMENSION A1(3,3),XYZ(3)
    DATA CONST/5.59612E-7/,DENS/2.33/,ACONST/1.1658/,AISI/172.253/
DATA C1/5.61563/,C4/3.51761E15/
DATA C3/4.23172E19/,C2/4.99878E22/
ALPHAS=C1/(SQRT(EEV))
Z=ALPHAS*ALPHAS
VEL2=C4*EEV
SIGMAS=C3/(VEL2*Z*(Z+1))
SIGMAS=SIGMAS/(VEL2)
ALAMDA=1.0/(C2*SIGMAS)
CALL RNUM3
R5=YFL3
STEP=-ALAMDA*ALOG(R5)
STEPN=DENS*STEP/CONST
EEVI=EEV
EEVN=EEV*ACONST/AISI
EEVN=EEVN-(STEPN*ALOG(EEVN)/EEVN)
EEV=AISI*EEVN/ACONST
EEVL=EEVI-EEV
RETURN
END
SUBROUTINE LEAD
COMMON/RAN1/IY1,IRN1,YFL1
COMMON/RAN2/IY2,IRN2,YFL2
COMMON/RAN3/IY3,IRN3,YFL3
COMMON/RAN4/IY4,IRN4,YFL4
COMMON/RAN5/IY5,IRN5,YFL5
COMMON A1,XYZ,TETA,FIE,STEP,EEV,ALPHAS,ITT,EEVL
DIMENSION A1(3,3),XYZ(3)
```

```
DATA CONST/1.61224E-5/,DENS/11.3/,ACONST/1.1658/,AISI/825.773/
DATA C1/10.1200/,C4/3.51761E15/
DATA C3/1.37000E21/,C2/3.30000E22/
ALPHAS=C1/(SQRT(EEV))
Z=ALPHAS*ALPHAS
VEL2=C4*EEV
SIGMAS=C3/(VEL2*Z*(Z+1))
SIGMAS=SIGMAS/(VEL2)
ALAMDA=1.0/(C2*SIGMAS)
CALL RNUM3
R5=YFL3
STEP=-ALAMDA*ALOG(R5)
STEPN=DENS*STEP/CONST
EEVI=EEV
EEVN=EEV*ACONST/AISI
EEVN=EEVN-(STEPN*ALOG(EEVN)/EEVN)
EEV=AISI*EEVN/ACONST
EEVL=EEVI-EEV
RETURN
END
SUBROUTINE SMOOT(AR)
COMMON/PRD1/INRMAX,INR1,INR2,NUMFF,AMZ,AMXY,AXY,AR1
C,FILM,EEV1,AISI
DIMENSION AR1(20,150),AXY(20,150)
DIMENSION AR(20,150)
DIMENSION IT(20)
NOR=INR2
```

```
      DO8I=1,NOR
      IT(I)=0
      IE1=149
4     DO6J=1,IE1
      IF(AR(I,J).LT.AR(I,J+1))GO TO 5
      GO TO 6
5     AMEAN=(AR(I,J)+AR(I,J+1))/2.
      AR(I,J)=AMEAN
      AR(I,J+1)=AMEAN
6     CONTINUE
      IT(I)=IT(I)+1
      IF(IT(I).EQ.2000) GO TO 8
      DO7J=1,IE1
      IF(AR(I,J).LT.AR(I,J+1))GO TO 4
7     CONTINUE
8     CONTINUE
      WRITE(6,9)(IT(I),I=1,NOR)
9     FORMAT(" NO OF ITERATIONS FOR SMOOTHING FOR EACH ROW",/22I6//)
      DO12I=1,NOR
      WRITE(6,10)I
10    FORMAT(" ROW NUMBER=",I4/)
      WRITE(6,11)(AR(I,J),J=1,50)
11    FORMAT(10E13.6)
12    CONTINUE
      RETURN
      END
```

RESIS3VAX.F - File listing

```
C    This file is for convolution.
     COMMON/PRD1/INRMAX,INR1,INR2,NUMFF,AMZ,AMXY,AXY,AR
     COMMON/PRD2/AX,AY
     COMMON/PRD3/EDFL
     DIMENSION EDFL(5,1001)
     DIMENSION AX(20,150),AY(20,150),AXY(20,150),AR(20,150)
     INRMAX=20
     INR1=1
     INR2=4
     NUMFF=5000
     AMZ=0.3E-4
     AMXY=0.02E-4
     OPEN(10,FILE="impulse")
     REWIND 10
     READ(10,1)((AXY(I,J),J=1,150),I=1,20)
     READ(10,1)((AR(I,J),J=1,150),I=1,20)
1    FORMAT(10E13.6)
     CLOSE(10)
C    CALL SEG2
     CALL SEG3
C    CALL SEG4
C    CALL SEG5
     STOP
     END
```

```
      BLOCK DATA RANDOM
      COMMON/PRD3/EDFL
      DIMENSION EDFL(5,1001)
      DATA EDFL/5005*0./
      END
      SUBROUTINE SEG3
C     SEG3 DOES ONE DIMENSIONAL LINE CONVOLUTION WITH GAUSSION BEAM
      COMMON/PRD1/INRMAX,INR1,INR2,NUMFF,AMZ,AMXY,AXY,AR
      COMMON/PRD3/EDFL
      DIMENSION AX(20,150),AY(20,150),AXY(20,150),AR(20,150)
      DIMENSION DELTA(5,501)
      DIMENSION EDF(5,1001),EDFL(5,1001)
      DIMENSION GAUSS(501)
      EQUIVALENCE (EDFL(1,1),EDF(1,1))
      EQUIVALENCE (AXY(1,1),AY(1,1),AX(1,1))
      DATA GAUSS/501*0.0/
      WRITE(6,555)
555   FORMAT(" GIVE THE VALUE OF BEAM DIAMETER IN MICRONS")
      READ(5,*) BEAMD
6     FORMAT(" I=",I5,"   NUMFF=",I5/)
7     FORMAT(10E13.6)
      IR=INR2
      IE1=100
      MIDG=250
      MIDDL=250
      MIDV=500
      BEAMD=BEAMD*1.E-4
      BEAMR=0.5*BEAMD
      CELLX=AMXY
      HCELL=CELLX*0.5
      WEIGHT=1.0
      IG=INT(3.*BEAMR/CELLX)
      I1=MIDG-IG
      I2=MIDG+IG
      X=HCELL
      AREA=0.0
      DO501I=MIDG,I2
      AR1=X/BEAMR
      GAUSS(I)=0.5*ERF(AR1)-AREA
      AREA=AREA+GAUSS(I)
      X=X+CELLX
501   CONTINUE
      GAUSS(MIDG)=2.*GAUSS(MIDG)
      WRITE(6,502)AREA,GAUSS(MIDG)
502   FORMAT(" AREA=",2E13.5)
      IG1=IG+1
      DO503I=1,IG1
      II=MIDG-I+1
      III=MIDG+I-1
      GAUSS(II)=GAUSS(III)
503   CONTINUE
      WRITE(6,504)(GAUSS(I),I=I1,I2)
504   FORMAT(10E13.5)
      J1=IE1+1
```

```
     DO512I=1,IR
     DO511J=1,J1
     K=MIDDL+J-1
     DELTA(I,K)=AXY(I,J)
     K=MIDDL-J+1
     DELTA(I,K)=AXY(I,J)
511  CONTINUE
512  CONTINUE
     DO601I=1,IR
     WRITE(6,6)I,NUMFF
     WRITE(6,7)(DELTA(I,J),J=MIDDL,MIDDL+J1)
601  CONTINUE
     K1=MIDV-IG
     K2=MIDV+IG
     M1=MIDDL-IE1
     M2=MIDDL+IE1
     DO516I=1,IR
     K3=0
     DO515K=K1,K2
     N=K-IE1
     DO513M=M1,M2
     EDF(I,N)=EDF(I,N)+DELTA(I,M)*GAUSS(MIDG-IG+K3)
     N=N+1
513  CONTINUE
     K3=K3+1
515  CONTINUE
516  CONTINUE
```

```
      L1=MIDV
      L2=MIDV+IG+IE1
      DO517I=INR1,2
      WRITE(6,6)I,NUMFF
      WRITE(6,7)(EDF(I,J),J=L1,L2)
  517 CONTINUE
      OPEN(11,FILE="lineedf")
      REWIND 11
      WRITE(11,518)((EDF(I,J),J=1,1001),I=1,IR)
  518 FORMAT(10E13.6)
      CLOSE(11)
      RETURN
      END
      FUNCTION ERF(X)
      DATA TOL,EK1/1.E-4,1.12837/
      IF(X.LE.0.0) GO TO 5
      IF(X.GE.3) GO TO 6
      X2=X*X
      SUM=X
      TERM=X
      I=0
  1   I=I+1
      IF(I.EQ.100) GO TO 2
      SUM1=SUM
      TERM=TERM*X2/(FLOAT(I)+0.5)
      SUM=SUM1+TERM
      IF(TERM.GE.(TOL*SUM1)) GO TO 1
      ERF=(EK1*SUM*EXP(-X*X))
      RETURN
2     WRITE(6,3)I
3     FORMAT(" I=",I3)
      RETURN
5     ERF=0.0
      RETURN
6     ERF=1.0
      RETURN
      END
      SUBROUTINE SMOOT(AR)
      COMMON/PRD1/INRMAX,INR1,INR2,NUMFF,AMZ,AMXY,AXY,AR1
      DIMENSION AR(20,150),AXY(20,150)
      DIMENSION AR1(20,150)
      DIMENSION IT(20)
      NOR=INR2
      IE1=149
      DO8I=1,NOR
      IT(I)=0
4     DO6J=1,IE1
      IF(AR(I,J).LT.AR(I,J+1)) GO TO 5
      GO TO 6
5     AMEAN=(AR(I,J)+AR(I,J+1))/2.
      AR(I,J)=AMEAN
      AR(I,J+1)=AMEAN
6     CONTINUE
      IT(I)=IT(I)+1
      IF(IT(I).EQ.2000) GO TO 8
      DO7J=1,IE1
      IF(AR(I,J).LT.AR(I,J+1)) GO TO 4
7     CONTINUE
8     CONTINUE
      WRITE(6,9)(IT(I),I=1,NOR)
9     FORMAT(//" NO OF ITRATIONS FOR SMOOTHING",/22I6//)
      DO12I=1,NOR
      WRITE(6,10)I
10     FORMAT(" I=",I6)
      WRITE(6,11)(AR(I,J),J=1,50)
11     FORMAT(10E13.6)
12     CONTINUE
      RETURN
      END
```

LINETOTAL.F - File listing

```
c    this program computes total exposure perpandicular
c    to line patterns (patterns will be assummed to be
c    composed of lines seperated by minimum distance=
c    0.02 microns
     real linet, linew,ledf
     dimension linet(4,1001),linew(4,301),ledf(4,1001)
     dimension x(2000),y(2000)
     inr1=1
     inr2=4
     midv=500
     midlw=151
     midlt=501
     k1=0
     amesh=0.02
     open(11,file="lineedf")
     rewind 11
     read(11,5)((ledf(i,j),j=1,1001),i=1,4)
5    format(10e13.6)
     close(11)
     do6i=inr1,inr2
     do6j=1,151
     k=j-1
     linew(i,midlw+k)=ledf(i,midv+k)
     linew(i,midlw-k)=linew(i,midlw+k)
```

```
6   continue
    icount=0
    open(4,file="linedata")
    rewind 4
7   read(4,8)beamp,width,step,dose
8   format(4f6.2)
    if(beamp.lt.-7.) go to 20
    if(dose.eq.0.0) dose=1.0
    icount=icount+1
    j1=midlt+int(beamp/0.02)
    if(icount.eq.1)jmin=j1
    j2=j1+int(width/amesh)
    if(j2.gt.850) go to 25
    nstep=1+int(step/0.02)
    do12i=inr1,inr2,3
    do11j=j1,j2,nstep
    k=0
    do10jj=j-midlw,j+midlw
    k=k+1
    linet(i,jj)=linet(i,jj)+linew(i,k)*dose
10  continue
11  continue
12  continue
    go to 7
20  write(6,21)
21  format("pattern data is less than -7 microns"/)
    go to 30
```

```
25  write(6,26)
26  format(" pattern data is more than +7 microns")
30  continue
    k1=0
    do32i=inr1,inr2,3
    do31j=jmin-midlw,j2+midlw
    k1=k1+1
c   if(j.lt.midlt) l=j-midlt
c   if(j.gt.midlt) l=j-midlt
    l=j-midlt+1
    x(k1)=0.02*float(l)
    y(k1)=linet(i,j)
    if(y(k1).lt.1.e+2) y(k1)=1.e+2
    y(k1)=alog10(y(k1))
31  continue
32  continue
    open(15,file="lexposure")
    rewind 15
    write(15,33)
33  format(".title 1.2\um PMMA ON Pb,20Kev,BEAM DIA=0.12\um"/
   1".title NO.OF PATTS=2,GAP=0.64\um"/
   2".xscale MICRONS'/
   3".yscale LOG(EV/(SQ CM.ELECTRON))"/
   4".yset 11 13"/
   5".xset 6 -6")
    write(15,34)(x(i),y(i),i=1,k1)
34  format(f6.2,5x,f6.2)
```

```
stop
end
```

LPROXI1.F - File listing

```
c    This program does the intra proximity correction in a pattern
c    Pattern will be composed of lines separated by 0.16 microns
     real linet, linew,ledf
     common/prd11/x,y
     common/prd10/linew,linet,dose,midlt,midlw,inr1,inr2,n,ngap
     dimension dose(501)
     dimension linet(4,1001),linew(4,301),ledf(4,1001)
     dimension x(2000),y(2000)
     inr1=1
     inr2=4
     midv=500
     midlw=151
     midlt=501
     k1=0
     amesh=0.02
     open(11,file="lineedf")
     rewind 11
     read(11,5)((ledf(i,j),j=1,1001),i=1,4)
5     format(10e13.6)
     close(11)
     do6i=inr1,inr2
     do6j=1,151
     k=j-1
     linew(i,midlw+k)=ledf(i,midv+k)
```

```
      linew(i,midlw-k)=linew(i,midlw+k)
6     continue
      call lproxi
      stop
      end
      subroutine lproxi
c     This routine computes QL (line charge density)
c     for each line in a pattern to have a uniform
c     exposure .
      common/prd10/linew,linet,dose,midlt,midlw,inr1,inr2,n,ngap
      real line,linew,linet
      dimension linet(4,1001)
      dimension linew(4,301),dose(501)
      dimension line(301)
      dimension a(100,100)
      inr2=4
      write(6,1)
1     format(" give the value of number of lines in the pattern"/)
      read(5,*)n
      m=n+1
      midlw=151
      anorm= linew(inr2,midlw)
      ngap=8
      ng=ngap*(2*n)
      jj=0
      do4j=midlw,midlw+ng,ngap
      line(midlw+jj)=linew(inr2,j)/anorm
      line(midlw-jj)=line(midlw+jj)
      jj=jj+1
4     continue
      kk=midlw+2*n
      do6i=1,n
      do5j=i,n
      k=j-i
      a(i,j)=line(midlw+k)
5     continue
6     continue
      do9i=1,n
      do8j=i,n
      a(j,i)=a(i,j)
8     continue
9     continue
      do11j=1,n
      a(j,m)=1.
      dose(j)=0.0
11    continue
c     write(6,100)((a(i,j),j=1,m),i=1,n)
c00   format(6e13.6/)
      it=1000
      eps=0.001
      do30i=1,it
      switch=0.0
      do28j=1,n
      sum=a(j,m)
      do26k=1,n
      if(k.eq.j) go to 26
      sum=sum-a(j,k)*dose(k)
26    continue
      dnext=sum/a(j,j)
      if(abs(dnext-dose(j)).le.eps) go to 27
      switch=1.0
27    dose(j)=dnext
      if(switch.le.0.) go to 35
28    continue
30    continue
35    write(6,40)i,(j,dose(j),j=1,n)
40    format(" No. of iterations required=",i5//(i5,5x,e13.6))
      open(4,file="linedata")
      rewind 4
      gap=0.64
c     beamp=-2.48
      beamp=-0.16*float((n-1)/2)
      width=0.0
      step=0.16
c     do51j=1,2
c     if(j.eq.2)beamp=gap/2.
      do51i=1,n
c     dose(i)=1.0
      write(4,50)beamp,width,step,dose(i)
      beamp=beamp+step
50    format(3f6.2,f6.2)
```

```
51   continue
     beamp=-8
     write(4,50)beamp
     return
     end
```

LPROXI2.F - File listing

```
c    This program does both intra and inter
c    proximity corrections.
     real linet, linew,ledf
     common/prd11/x,y
     common/prd10/linew,linet,dose,midlt,midlw,inr1,inr2,n,ngap
     dimension dose(501)
     dimension linet(4,1001),linew(4,301),ledf(4,1001)
     dimension x(2000),y(2000)
     inr1=1
     inr2=4
     midv=500
     midlw=151
     midlt=501
     k1=0
     amesh=0.02
     open(11,file="lineedf")
     rewind 11
     read(11,5)((ledf(i,j),j=1,1001),i=1,4)
5    format(10e13.6)
     close(11)
     do6i=inr1,inr2
     do6j=1,151
     k=j-1
     linew(i,midlw+k)=ledf(i,midv+k)
```

```
      linew(i,midlw-k)=linew(i,midlw+k)
6     continue
      call lproxi
      stop
      end
      subroutine lproxi
c     This routine computes QL (line charge density)
c     for each line in a pattern to have a uniform
c     exposure .
      common/prd10/linew,linet,dose,midlt,midlw,inr1,inr2,n,ngap
      real line,linew,linet
      integer pgap
      dimension linet(4,1001)
      dimension linew(4,301),dose(501)
      dimension line(301)
      dimension a(100,100)
      inr2=4
      write(6,1)
1     format(" give the value of numberof lines")
      read(5,*)n
      write(6,2)
2     format(" Give the gap between patterns"/
     1" in unites of 0.16 microns")
      read(5,*)pgap
      m=n+1
      midlw=151
      anorm= linew(inr2,midlw)
      ngap=8
      ng=ngap*(2*n)
      jj=0
      do4j=midlw,midlw+ng,ngap
      line(midlw+jj)=linew(inr2,j)/anorm
      line(midlw-jj)=line(midlw+jj)
      jj=jj+1
4     continue
      kk=midlw+2*n
      pgap=pgap-1
      ii=n/2
      do6i=1,ii
      do6j=i,n
      k=j-i
      if(j.gt.ii) go to 5
      a(i,j)=line(midlw+k)
      go to 6
5     a(i,j)=line(midlw+k+pgap)
6     continue
      do7i=ii+1,n
      do7j=i,n
      k=j-i
      a(i,j)=line(midlw+k)
7     continue
      do9i=1,n
      do8j=i,n
      a(j,i)=a(i,j)
```

```
8    continue
9    continue
     do11j=1,n
     a(j,m)=1.
     dose(j)=0.0
11    continue
c    write(6,100)((a(i,j),j=1,m),i=1,n)
c00  format(6e13.6/)
     it=1000
     eps=0.001
     do30i=1,it
     switch=0.0
     do28j=1,n
     sum=a(j,m)
     do26k=1,n
     if(k.eq.j) go to 26
     sum=sum-a(j,k)*dose(k)
26    continue
     dnext=sum/a(j,j)
     if(abs(dnext-dose(j)).le.eps) go to 27
     switch=1.0
27    dose(j)=dnext
     if(switch.le.0.) go to 35
28    continue
30    continue
35    write(6,40)i,(j,dose(j),j=1,n)
40    format(" No. of iterations required=",i5//(i5,5x,e13.6))
```

```
      open(4,file="linedata")
      rewind 4
      gap=0.16*float(pgap+1)
      beamp=-0.16*float((n/2-1))-gap/2.
      width=0.0
      step=0.16
      ii=n/2
      do51i=1,ii
c     dose(i)=1.0
      write(4,50)beamp,width,step,dose(i)
      beamp=beamp+step
50     format(3f6.2,f6.2)
51     continue
      beamp=(gap/2.)
      do55i=ii+1,n
c     dose(i)=1.0
      write(4,50)beamp,width,step,dose(i)
      beamp=beamp+step
55     continue
      beamp=-8
      write(4,50)beamp
      return
      end
```

READIMPULSE.F - File listing

```
C     This file is for plotting the delta line edf.
      dimension edf(20,150),ax(3000),x(3000)
      inr1=1
      inr2=4
      ie1=149
      open(10,file ="impulse")
      rewind 10
      read(10,4)((edf(i,j),j=1,150),i=1,20)
4     format(10e13.6)
      close(10)
      k=0
      do5i=inr1,inr2
      l=0
      do5j=1,ie1
      k=k+1
      l=l+1
      ax(k)=edf(i,j)
      if(ax(k).lt.20.) ax(k)=20.
      ax(k)=alog10(ax(k))
      x(k)=float(l-1)*0.02
5     continue
      open (12,file="general")
      rewind 12
      write(12,7)
      write(12,6)(x(i),ax(i),i=1,k)
 6    format(f6.2,5x,f6.2)
 7    format(".title Delta line EDF curves"/
     1".title 1.2m PMMA on LEAD,Beam energy=20Kev"/
     1".autoline"/
     2".xscale MICRONS"/
     3".yscale EV/(SQ CM . ELECTRON)"/
     4".yset 13 9"/
     5".xset 1 0")
      close(12)
      end
```

READIMPULSER.F - File listing

```
C    This file is for plloting delta radial edf.
     dimension edf(20,150),ax(3000),x(3000)
     inr1=1
     inr2=4
     ie1=50
     open(10,file ="impulse")
     rewind 10
     read(10,4)((edf(i,j),j=1,150),i=1,20)
     read(10,4)((edf(i,j),j=1,150),i=1,20)
4    format(10e13.6)
     close(10)
     k=0
     do5i=inr1,inr2
     l=0
     do5j=1,ie1
     k=k+1
     l=l+1
     ax(k)=edf(i,j)
     if(ax(k).lt.20.) ax(k)=20.
     ax(k)=alog10(ax(k))
     x(k)=float(l-1)*0.02
5    continue
     open (12,file="general")
     rewind 12
```

```
      write(12,7)
      write(12,6)(x(i),ax(i),i=1,k)
 6    format(f6.2,5x,f6.2)
c     format(f6.2,5x,e13.6)
7     format(".title Delta radial EDF curves"/
     1".title 1.2m PMMA NO LEAD,Beam energy=20Kev"/
     1".autoline"/
     2".xscale MICRONS"/
     3".yscale EV/(Cubic CM . ELECTRON)"/
     4".yset 17 13"/
     5".xset 3 0")
      close(12)
      end
```

READLINEEDF.F - File listing

```
C    This file plots line edf after convolution.
     dimension edf(4,1001),ax(3000),x(3000)
     inr1=1
     inr2=4
     ie1=149
     open(11,file ="lineedf")
     rewind 11
     read(11,4)((edf(i,j),j=1,1001),i=inr1,inr2)
4    format(10e13.6)
     close(11)
     midv=500
     k=0
     do5i=inr1,inr2
     l=0
     do5j=midv,midv+ie1
     k=k+1
     l=l+1
     ax(k)=edf(i,j)
     if(ax(k).lt.20.) ax(k)=20.
     ax(k)=alog10(ax(k))
     x(k)=float(l-1)*0.02
5    continue
     open (12,file="general")
     rewind 12
```

```
      write(12,7)
      write(12,6)(x(i),ax(i),i=1,k)
6     format(f6.2,5x,f6.2)
7     format(".title Delta line EDF curves for 1.2m PMMA on LEAD"/
     1".title  Energy 20Kev,convolved with 0.12m Beam dia"/
     1".autoline"/
     2".xscale MICRONS"/
     3".yscale EV/(SQ CM . ELECTRON)"/
     4".yset 13 9"/
     5".xset 1 0")
      close(12)
      end
```