SIMULATION AND MODELING
OF EVAPORATED DEPOSITION PROFILES

by

Chiakang Sung

Memorandum No. UCB/ERL M81/8

17 February 1981

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

SIMULATION AND MODELING OF EVAPORATED DEPOSITION PROFILES

by

Chiakang Sung

I. ABSTRACT

This report presents a program for the simulation of metalization. The simulation is used to investigate metal step coverage for a variety of source and substrate configurations.

The models used for deposition combine analytic and numerical summations. The deposition rate is derived as an analytic function, and the simulation proceeds by summing the depostion through a series of finite time-steps. Simulations have been made to model metalization over steps with the following source-substrate geometries: (1) unidirectional source, (2) dual evaporation sources, (3) hemispherical vapor source, (4) point source with planetary-mounted substrate. The modeling technique has also been extended to deposition processes with elevated substrate temperature.

Research sponsored by the National Science Foundation under Grant ENG77-14660-A01, Hughes Aircraft, INTEL, Signetics, Hewlett-Packard and IBM Corporation.

The following is a reproduction of the authors M.S. Thesis, Plan II.

CONTENTS

II. INTRODUCTION

During the past decade, many investigations were carried out to explore the coverage of evaporated films over steps in substrates [1]-[9]. In particular, Blech derived a model which can be applied to any source geometry and step profile [1]. The simulator described in this report extends Blech's model to surface migration such as would occur in an elevated temperature environment.

The problem of the step coverage is geometric in nature. Every point on the substrate views the source subtending a certain solid angle. The growth rate of the evaporated film at each point is strongly dependent on the associated solid angle. Depending on the surface topology and the source configuration, the shadow effect can occur. As a result of shadowing, the solid angle varies in time and so does the growth rate (Fig. 1). Therefore, after a certain amount of time, some points on the substrate can be completely shadowed by other fast growing points. This can give rise to a crack or a poor step coverage.

The program simulates the growth of a deposited thin film with various source-substrate configurations. Two dimensional surface geometries with arbitrary shape are specified as input. The simulator generates a line-edge profile of the film coverage based on a user-defined source. By comparing the step coverages for various sources, the user can optimize the source-substrate configurations in the deposition apparatus.

III. ASSUMPTIONS

The following assumptions are made in the simulation:

(1) The mean free path of atoms is larger than the distance between source and substrate.

(2) The source-to-substrate distance is large compared to the step height.

(3) The magnitude of film growth rate follows the cosine distribution law, that is, the thickness of the film grows at a rate proportional to $\cos\omega$, where ω is the angle between the vapor stream and surface normal.

(4) The direction in which the film grows is towards the vapor stream.

(5) In case of a cold substrate, the sticking coefficient is assumed to be unity.

(6) In an elevated temperature environment, surface migration occurs on the substrate. It is assumed that the surface migration obeys a random walk law. The migration distance increases with the increase of the substrate temperature.

IV. SOURCE CONFIGURATIONS

The evaporation sources simulated in this program are described as follows:

UNIDIRECTIONAL SOURCE

As shown in Fig. 2, the unshadowed region of the substrate sees the arrival of the vapor streams in one direction only. The growth rate of the deposited film in the shadowed region is equal to zero. According to our assumptions, the growth rate on the substrate can be expressed as:

rate(x,z) = 0; if point (x,z) is shadowed

rate(x,z) = $C\sin\omega\hat{i} + C\cos\omega\hat{k}$; if point (x,z) is unshadowed

where ω is the angle between z-axis and the vapor stream, $\hat{i}$ and $\hat{k}$ are the unit vector in the x and z direction respectively, and C is growth rate of an unshadowed surface normal to the vapor stream.

DUAL EVAPORATION SOURCES

In this type of sources, each point in the unshadowed region views the vapor streams arriving from two different directions (Fig. 3). The growth rate is given as:

rate(x,z) = 0; if point (x,z) is completely shadowed

rate(x,z) = $C\sin\omega_1\hat{i}+ C\cos\omega_1\hat{k}$ or $C\sin\omega_2\hat{i}+ C\cos\omega_2\hat{k}$; if (x,z) is partially shadowed

rate(x,z) = $C(\sin\omega_1+\sin\omega_2)\hat{i} + C(\cos\omega_1+\cos\omega_2)\hat{k}$; if (x,z) is unshadowed

where ω_1 and ω_2 are the incident angles.

HEMISPHERICAL VAPOR SOURCE

The flux of vapor is continuously distributed in a range of directions (Fig. 4). The growth rate can be calculated as:

$$\text{rate}(x,z) = C(\cos\omega_1-\cos\omega_2)\hat{\imath} + C(\sin\omega_2-\sin\omega_1)\hat{k};$$

where ω_1 and ω_2 are the lower and the upper bounds of the incident angles of the vapor streams, respectively.

PLANETARY SOURCE

The planetary evraporation system is shown in Fig. 5. By inspecting this system, one can be convinced that the rotation of the planet along the system central axis has no effect on the deposition rate. For the purpose of simplicity, the growth rate can be calculated by holding the planet stationarily and by rotating only the source along the axis of the planet, as shown in Fig. 6. The growth rate is derived similarly to that of Blech [2]:

$\text{rate}_x(x,z)$

$$= \int \frac{[R^2-r^2-rL\tan(\alpha-\beta)+LW]}{(R^2+W^2)^{0.5}} \frac{[-L\sec^2(\alpha-\beta)]}{[R^2-r^2+L^2-2rL\tan(\alpha-\beta)]^2} \frac{[L\tan(\alpha-\beta)\sin\beta-L\cos\beta]\tan\alpha}{[R^2-(r+L\tan(\alpha-\beta))^2]^{0.5}} d\alpha$$

$\text{rate}_z(x,z)$

$$= \int \frac{[R^2-r^2-rL\tan(\alpha-\beta)+LW]}{(R^2+W^2)^{0.5}} \frac{[-L\sec^2(\alpha-\beta)]}{[R^2-r^2+L^2-2rL\tan(\alpha-\beta)]^2} \frac{[L\tan(\alpha-\beta)\sin\beta-L\cos\beta]}{[R^2-(r+L\tan(\alpha-\beta))^2]^{0.5}} d\alpha$$

where α is the incident angle of the vapor stream, β is the tilt angle of the planet plane, r is the distance between the position of the wafer and the planet axis, R, L, and W are the parameters dependent on the system dimensions as shown in Fig. 5.

CONE SOURCE: A SPECIAL CASE OF PLANETARY SOURCE

If both β and r are zero, the substrate always sees a symmetrical cone source. In this type of configuration, the integral of the above two equations can be evaluated analytically, and expressed in the following simple close form:

$$\text{rate}_x(x,z) = \frac{-R(R^2+LW)}{(R^2+W^2)^{0.5}(R^2+L^2)^2}\left[(1-(\frac{L}{R}\tan\alpha_{max})^2)^{0.5}-(1-(\frac{L}{R}\tan\alpha_{min})^2)^{0.5}\right]$$

$$\text{rate}_z(x,z) = \frac{L(R^2+LW)}{(R^2+W^2)^{0.5}(R^2+L^2)^2}\left[\sin^{-1}(\frac{L}{R}\tan\alpha_{max})-\sin^{-1}(\frac{L}{R}\tan\alpha_{min})\right]$$

V. TEMPERATURE EFFECT

Surface migration on a hot substrate is presumably possible owing to thermal motion of the deposited particles. If a random walk process is assumed, this particle redistribution from any infinitesimal area on the surface can be characterized by a Gaussian function of which the variance depends on temperature. The growth rate for any time interval Δt can be calculated by first calculating the deposition of the material arriving from the source at each surface point (x,z) and then the diffusion to and from the adjacent region. Each point thus sources a Gaussian function, and the sum of the Gaussian functions is the modified deposition rate.

$$\text{rate}'(x,z) = \sum_i \text{rate}(x_i,z_i)e^{\frac{-r^2}{2\sigma^2}}$$

where rate'(x,z) is the growth rate of point (x,z) on the hot substrate, i is summed over all points on the surface, rate(x_i,z_i) is the growth rate of point (x_i,z_i) but on a cold substrate as described before, r is the distance along the surface between (x,z) and (x_i,z_i), and σ is the variance of the Gaussian function. For convenience, the limits at the summation are terminated at the extent of three-variance range.

VI. PROGRAM ORGANIZATION

This program is developed as a portion of SAMPLE [10]. It is similar in structure to the SAMPLE "develop machine" [11], [12]. The string model [13] is also used in this program. Analytic functions are used wherever available in order to attain accuracy and to save computer time. The accuracy of the profile can be improved by reducing the time step or the segment length, at the cost of longer computation time. A typical profile, generated with 10 to 20 advances, composed of 40 to 100 line segments, can give a resonable accuracy.

The modular flowchart of this simulator is shown in Fig. 7. Subroutines PLOT, PUNCH, CHEKER, and DELOOP are those used in SAMPLE, except some slight modifications. The CONTROLLER receives input parameters, which describe the type of source and the substrate step. Then the PROFIL subroutine generates the initial step profile.

The whole processing procedures are now under control of the subroutine DPMAIN. First, subroutine INITLZ initializes various internal parameters. Then subroutine DPMESG echoes the parameters of the deposition configuration, as recognized by the simulator. If no fatal errors occur, subroutine SHADOW determines, on a point-by-point basis, the range of the incident angles of the incoming flux that the substrate views. Called by subroutine ADVNCE, subroutine EVRATE then evaluates the growth rate point by point according to the deposition system and the shadow condition. If hot substrate has been specified, subroutine ADVNCE will call another subroutine DIFF to handle surface migration. The advance of each point of the line-edge profile is performed step by step under the control of subroutine ADVNCE. During advancing, the string segments may become very long which reduces accuracy, or very short which wastes computer time. To avoid this, after each advance, subroutine CHEKER is called to adjust segment length. Subroutine CHEKER may call two other subroutines, ADD and DELETE which respectively performs the task to add and to delete points as the adjustment of the string segment lengths is needed. The purpose of separating ADD and DELETE from CHEKER is to allow the use of more sophisticated algorithms for adding or deleting points, if needed in the future.

As the string advances, spurious loops may form and they can be detected and deleted by calling the subroutine DELOOP. Finally, the output is produced by calling a line-printer-plotting subroutine PLOT and/or a card-punching subroutine PUNCH. Some other subroutines and functions, such as GROTBL, EVALUE, EVINTG, SUM, ANORM, and AMIGR are used only for intermediate calculations. They are self-explanatory with comments on the source code and therefore, they are not discussed here.

VII. USER DOCUMENTATION

Fig. 8 shows a typical output of an asymmetrical step coverage for a planetary rotating source [2]. The plot is obtained by an HP-plotter with data cards produced by the program. The input and output control statements are:

```
SOURCE METAL PLANETARY  (56., 10.)  (4.5, 18., 7.5)  (0.01875)
TIME METAL  0  TO  80,  11
SET CONTOUR 10 (0., 2.) (0.88,  2.)  (0.88,  2.3)  (1.2,  2.85)
(1.5, 3.) (4.5, 3.) (4.8, 2.85) (5.12, 2.3) (5.12, 2.) (6., 2.)
RUN METAL
```

The meaning of these statements is as follows:

```
SOURCE METAL PLANETARY  (56., 10.)  (4.5, 18., 7.5)  (0.01875)
```

specifies the parameters of the planetary metalization system: γ = 56 degrees, β = 10 degrees, r = 4.5 inches, central axis length = 18 inches, planet axis length = 7.5 inches, and deposition rate = 0.01875 microns/second.

```
TIME METAL  0  TO  80,  11
```

tells that the substrate is to be metalized from 0 to 80 seconds. The step profiles should be outputted at 0, 8, 16, 24, 32, 40, 48, 56, 64, 72, and 80 seconds.

```
SET CONTOUR 10 (0., 2.) (0.88,  2.)  (0.88,  2.3)  (1.2,  2.85)
(1.5, 3.) (4.5, 3.) (4.8, 2.85) (5.12, 2.3) (5.12, 2.) (6., 2.)
```

sets the initial profile of metalization. The step profile contains 10 turning points with the coordinates (x_1,z_1) = (0., 2.), ..., and (x_{10},z_{10}) = (6., 2.)

```
RUN METAL
```

this runs the metalization machine to generate the line-edge profiles of the film coverage.

Other deposition sources can be specified in the following formats:

SOURCE METAL UNIDIRECTION (45.) (0.005)

specifies a point source with an incident angle of 45°. The deposition rate is 0.005 microns/second.

SOURCE METAL DUAL (-45., 45.) (0.005)

specifies the metalization sources has dual incident angles of -45° and 45° with deposition rate = 0.005 microns/second.

SOURCE METAL HEMISPHERE (-90., 90.) (0.005)

specifies a hemispherical source with incident angles distributed continuously from -90° to 90°. The depositon rate is 0.005 microns/second.

SOURCE METAL CONE (56.) (18., 7.5) (0.005)

specifies a cone source with parameters: $\gamma = 56^{\circ}$, central axis length = 18", planet axis length = 7.5", and deposition rate = 0.005 microns/second.

However the interface between this program and the top level controller of SAMPLE will not be ready until the very near future.

VIII. SIMULATION RESULTS

Some simulation results are given below which illustrate the capabilities of the program. Fig. 9 shows the deposition under conditions of a typical dual e-beam gun evaporator. This figure illustrates the evolution of the film vacuum interface for 3 equal time steps. Because of the vertical step, a crack develops near the inside corner.

In Fig. 10(a) a hemispherical source is assumed [14], corresponding for example to certain sputtering sources. This figure illustrates the step coverage over a simple square step. The height of the step is of little importance as shown in Fig. 10(b). Fig. 11 shows the rounded top does not eliminate the crack. However the rounding of the inside corner is crucial as shown in Fig. 12.

Fig. 13 illustrates a simulation of double-metalization technique to minimize the shadowing effect [15]. In this simulation, the first layer of metal was deposited by exposing the wafer vertically to the source. The second layer of metal was then deposited, using a planetary configuration. Fig 13(a) shows a metal coverage failure from a single planetary deposition, and Fig. 13(b) shows an improved metal coverage with the double-metalization technique.

Fig. 14 shows an example of metalization, using a hemispherical source, on a hot substrate. Increased mobility of aluminum atoms in the growing film on the hot substrate leads to a reduction of film surface area and a filling of the crack.

Fig. 15(a) shows a symmetrical step coverage, using a certain planetary source. Fig. 15(b) shows the same deposition on a hot substrate. Since surface diffusion tends to fill the unfavorable cracks, depositing the metal on a heated substrate therefore may cause the disappearance of cracks. However too high a substrate temperature may also result in extensive grain boundary grooving which is often undesirable. The technique to improve step coverage but avoid grains is use of different temperatures for two layer metalization [16]. As shown in Fig. 15(c), the first layer of metal was deposited, using a planetary source, in room temperature. Then the second layer was deposited in the same conditions except substrate temperature. By comparing Fig. 15(c) with Fig. 15(a), and (b), one may find the optimum condition for step coverage without the occurrence of surface irregularity or grains.

IX. COMMON BLOCK DOCUMENTATION

This section lists the common blocks of the program in alphabetical order.

/CHKR/

common to CONTROLLER, PROFIL, CHEKER, and DELOOP.

SMINX the minimum x string segment length allowed by CHEKER.

SMINZ the minimum z string segment length allowed by CHEKER.

SMAXX the maximum x string segment length allowed by CHEKER.

SMAXZ the maximum z string segment length allowed by CHEKER.

XZDELT the initial string segment length.

/CRACK/

common to SHADOW, and EVRATE

WI array containing the minimum incident angles of the incoming flux for the 450 possibe string points.

WF array containing the maximum incident angles of the incoming flux for the 450 possible string points.

/DIFUSN/

common to CONTROLLER, ADVNCE, EVRATE, DIFF, ANORM, and AMIGR.

RATE complex array containing the x and z component of the growth rate for the 450 possible string points.

SIGMA a parameter controls the extent of surface migration.

/ETCH1/

ETCH1 is the only common block used in the other parts of SAMPLE program. In this simulator, it is common to the CONTROLLER, PROFIL, DPMAIN, SHADOW, ADVNCE, EVRATE, DIFF, ANORM, AMIGR, CHEKER, DELETE, ADD, DELOOP, PLOT, and PUNCH

XZ complex array containing the x and z positions of the 450 possible deposition string-points.

XMAX the maximum value of x.

ZMAX the maximum value of z.

NPTS the current number of depostion string-points.

CXZL not used.

CXZR not used.

NADCHK not used.

NCKOUT not used.

/METAL/

common to CONTROLLER, DPMAIN, INITLZ, DPMESG, SHADOW, ADVNCE, EVRATE, GROTBL, PLOT, and PUNCH

TETCHR deposition growth rate; minus sign means film grows towards negative-z (upward) direction.

TMOUT time of the last output.

NMOUT total number of the outputs.

ANGLE array containing the minimum and maximum incident angles of the incoming flux.

DELT the time step between advances.

NTOTAL the total number of advances.

/MTFLAG/

common to CONTROLLER, DPMAIN, ADVNCE, DIFF, GAUSSN, ANORM, and AMIGR.

MCOUNT the current number of advances.

MDIFF flag used to call DIFF; 1 means "yes", 0 means "no".

MPLTHP flag used to call PUNCH; 1 means "yes", 0 means "no".

MDLOOP flag used to call DELOOP; 1 means "yes", 0 means "no".

MTYPE flag used to specify different types of sources; 1 means "dual evaporation sources", 2 means "hemispherical vapor source", 3 means "cone source", 4 means "planetary source", 5 means "unidirectional source".

/PLANET/

common to CONTROLLER, INITLZ, DPMESG, EVRATE, and SUM.

DR, DL, DW frequently used internal parameters; to calculate the growth rate in planetary source.

RSL only used in planetary source, the distance between the position of the substrate and the planet rotating axis.

AIW, CSTHET frequently used internal parameters; to calculate the growth rate in planetary source.

/PRFILE/

common to CONTROLLER, and PROFIL

TXZ complex array containing the x and z positions of the 40 possible turning points used to set the initial profile.

NT number of turning points of the initial profile.

/SAVERT/

common to GROTBL, and EVALUE

XP, XM only used in planetary source, array containing the temporary results of growth rate in x direction.

ZP, ZM only used in planetary source, array containing the temporary results of growth rate in z direction.

/SUMCON/

common to INITLZ, and SUM

C1, C2, C4, C5 frequently used internal parameters; to calculate the growth rate in planetary source.

/SYSTEM/

common to CONTROLLER, INITLZ, DPMESG, and EVRATE.

GAMMA only used in planetary source, the angle between the system central axis and the planet rotating axis.

BETA only used in planetary source, the tilt angle of the planet plane.

PHI only used in planetary source, the angle between the position of the substrate and the planet axis.

SW only used in planetary source, the length of the system central axis.

RP only used in planetary source, the length of the planetary rotating axis.

MRSL flag used to control input format in planetary source; 1 means "specify the position of substrate by PHI", 0 means "specify the position of substrate by RSL".

X. EXPERIMENT

In an effort to verify the models used in the simulator, Al evaporations were performed in vacuum at room temperature. The processing sequence is shown in Fig. 16, and also in the appendix. A photolithographic masking pattern (Fig. 17) consisting of varying line widths and spaces (1-50μm) was used in this experiment.

Stripes were delineated in a 0.6μm thermally grown SiO_2 on a <100> Si substrate. The thermal oxidation process was carried out at 1200°C and used dry oxygen-water vapor gases sequentially. After delineation, the Si substrate was preferentially etched to form <111> steps. Then the oxide on top of the substrate was removed. Aluminum layer (0.6-1.0μm) was evaporated on top of the substrate which was mounted on a planetary fixture. The Si steps were oriented either parallel or perpendicular to the Al flux.

Samples were cleaved across silicon steps to examine step angles and deposition profiles. Scanning electron microscope was used in examining cross sections and surface topography. The SEM photographs and corresponding simulation results are shown in Fig. 18.

XI. CONCLUSION

In this report, the problem of step coverage is examined through a simulation program. A good step coverage may be obtained by the optimization of source-substrate configuration in a deposition apparatus, by the use of elevated substrate temperature, and by the control of the substrate step profiles.

ACKNOWLEDGEMENTS

I wish to express my sincere thanks to the persons associated with the SAMPLE program. I wish to thank Professor Andy Neureuther who has encouraged me and given me inspiration in this report. Also thanks are due to Sharad Nandgaonkar for his help and suggestions. I am also deeply grateful to Michael O'Toole, Shankar Subramanian, and John Reynolds for their assistance.

I especially wish to thank my advisor, Professor William G. Oldham, for his support, guidance, and inspiration throughout my graduate year. His enthusiasm and indispensable aid were of great value to me.

Research sponsored by the National Science Foundation grant ENG77-14660-A01.

APPENDIX: PROCESSING SEQUENCE

1. Start with <100>, 1-5 ohms-cm, p-type, 2" diameter Si wafer

2. Wafer Cleaning:

   a. 10 minute dip in piranha etch (H_2SO_4:H_2O_2 5:1)

   b. rinse in deionized water for 2 minutes

   c. 20 second dip in aqueous etch (H_2O:HF 10:1)

   d. rinse in deionized water for 2 minutes

   e. blow dry with N_2

3. Oxide Growth: (thickness: 0.6μm, temperature: 1000°C)

   a. 5 minute push in N_2 at 4.0 cm

   b. 5 minute dry O_2 at 6.5 cm

   c. 100 minute wet O_2 at 2.0cm (water: 97°C)

   d. 10 minute anneal in N_2 at 4.0 cm

   e. 5 minute pull in N_2 at 4.0 cm

4. Standard Clean:

   a. dip in piranha etch for 3 minutes

   b. rinse, blow dry (perform step 2 (d) and (e))

5. Prebake:

   80°C for 15 minutes

6. HMDS Treatment:

   a. 10 minute HMDS vapor bath

   b. 5 minute N_2 flow

7. Photoresist Coating:

a. sqirt photoresist (AZ 1350J) on wafer

b. wait for 10 seconds

c. spin at 6000 rpm for 30 seconds

d. remove wafer after 10 seconds

8. Softbake:

$80^{\circ}C$ in N_2 for 15 minutes

9. Expose: (contact printing)

a. load mask and wafer

b. set automatic timer according to lamp intensity

c. expose

d. unload wafer and mask

10. Develop:

a. dip in H_2O:AZ Developer 1:1 for 1 minute (at $21^{\circ}C$)

b. rinse, blow dry

11. Inspect wafers under microscope and note condition:

note: in case of severe defects, remove photoresist and restart from prebake

12. Postbake:

$120^{\circ}C$ for 20 minutes

13. Oxide Etch:

a. dip in buffered HF ($HF:NH_4F$ 1:4) for 50% overetch

b. rinse, blow dry

note: (i) oxide etchrate: 0.1μm/min

(ii) water sheets off unprotected surface

14. Inspect wafers under microscope:

note: oxide of unprotected region has been etched completely before removing photoresist

15. Photoresist Stripping:

a. 5 minute acetone strip

b. rinse, blow dry

16. Preferential Etch:

a. immerse wafer in preferential etchant of silicon ($NH_2(CH_2)_2NH_2$:$C_6H_4(OH)_2$:H_2O 255ml:45g:120ml) for 3 minutes at 115°C

b. rinse, blow dry

note: (i) use an aspirator for vapor refluxing [17], [18], [19]

(ii) etch ratios for <111>:<110>:<100> 3:30:50

17. Aluminum Evaporation: (pressure: $8x10^{-7}$mmHg, cold substrate)

a. mount wafer on a planetary fixture

b. turn on driving motor

c. deposit aluminum film 0.7-1.0 μm

18. Scribing

19. Inspect dice under scanning electron microscope:

a. step coverage profiles

b. thickness distribution

c. hillocks

REFERENCES

1. I. A. Blech, "Evaporated Film Profiles over Steps in Substrates", Thin Solid Films, 6, 113, (1970)

2. I. A. Blech, D. B. Fraser, S. E. Haszko, "Optimization of Al Step Coverage through Computer Simulation and Scanning Electron Microscopy", J. Vac. Sci. Technol., 15, 13, (1978)

3. T. C. Tisone, J. B. Bindell, "Step Coverage in the Vacuum Deposition of Thin Metal Films", J. Vac. Sci. Technol., 11, 72, (1974)

4. J. B. Bindell, T. C. Tisone, "Step Coverage from an Extended Sputtering Source", Thin Solid Films, 23, 31, (1974)

5. K. H. Behrndt, "Thickness Distribution and Step Coverage in a New Planetary Substrate Holder Geometry", J. Vac. Sci. Technol., 9, 2, (1972)

6. F. Hegner, A. Feuerstein, "Aluminum-Silicon Metallization by Rate Controlled Dual EB-Gun Evaporation", Solid State Technol., 21, 49, (1978)

7. J. S. Logan, F. S. Maddocks, P. D. Davidse, "Metal Edge Coverage and Control of Charge Accumulation in RF Sputtered Insulators", IBM J. Res. & Develop., 14, 182, (1970)

8. T. N. Kennedy, "Sputtered Insulator Film Contouring over Substrate Topography", J. Vac. Sci. Technol., 13, 6, (1976)

9. V. J. Silvestri, V. L. Rideout, V. Maniscalco, "Al Coverage of Surface Steps at SiO_2 Insulated Polycrystalline Si Boundaries: Al Evaporation in Vacuum and Low Pressure Ar", J. Electrochem. Soc., 126, 1335, (1979)

10. W. G. Oldham, S. N. Nandgaonkar, A. R. Neureuther, M. M. O'Toole, "A General Simulator for VLSI Lithography and Etching Processes: Part I --- Application to Projection Lithography", IEEE Trans. Electron Devices, ED-26, 717, (1979)

11. M. M. O'Toole, "Simulation of Optically Formed Image Profiles in Positive Photoresist", Electronic Res. Lab., Department of EECS, U. C. Berkeley, (1979)

12. S. N. Nandgaonkar, "Design of a Simulator Program (SAMPLE) for IC Fabrication", Electronic Res. Lab., Department of EECS, U. C. Berkeley, (1978)

13. R. E. Jewett, P. I. Hagouel, A. R. Neureuther, T. Van Duzer, "Line-Profile Resist Development Simulation Techniques", Polymer Engineering and Science, 17, 6, (1977)

14. W. G. Oldham, C. Sung, A. R. Neureuther, "Simulation of Evaporated Deposition Profiles", Fifteenth Symp. on Electron, Ion and Photon Beam Technology, Extended Abstract No. 93, (May 1979)

15. S. N. Lee, R. A. Kjar, "SOS Island Edge Profiles Following Oxidation", Thirteenth Annual Proceedings Reliability Physics Symposium, (1975)

16. S. S. Baird, "Hot and Cold Substrate Al Evaporation", Processing Seminar, Department of EECS, U. C. Berkeley, (October 1979)

17. P. I. Hagouel, "X-Ray Lithographic Fabrication of Blazed Diffraction Gratings", Ph. D. Dissertation, Department of EECS, U. C. Berkeley, (1976)

18. R. Sud, "Modeling and Characterization of High Resolution Resist for Submicron Integrated Circuits", M. S. Thesis, Department of EECS, U. C. Berkeley, (1977)

19. K. E. Bean, "Anisotropic Etching of Silicon", IEEE Trans. Electron Devices, ED-25, 1185, (1978)

SOURCE CODE

```
      program         metaln

c  *****this version is for f4p compiler on unix pdp 11/70 system*****
c  this program simulates the line edge profile of metalization with
c  various source-substrate configurations

      common/metal/ tetchr,tmout,nmout,angle(2),delt,ntotal
      common/etch1/ xz(450),xmax,zmax,npts,cxzl,cxzr,nadchk,nckout
      common/mtflag/ mcount,mdiff,mplthp,mdloop,mtype
      common/difusn/ rate(450),sigma
      common/prfile/ txz(40),nt
      common/chkr/ sminx,sminz,smaxx,smaxz,xzdelt
      common/system/ gamma,beta,phi,sw,rp,mrsl
      common/planet/ dr,dl,dw,rsl,aiw,csthet
      complex xz,cxzl,cxzr,txz,rate

      xmax=2.0
      zmax=0.8000

      xzdelt=0.05
      sminx=0.2*xzdelt
      sminz=0.2*xzdelt
      smaxx=1.8*xzdelt
      smaxz=1.8*xzdelt

      nt=4
      txz(1)=cmplx(0.,zmax/2.)
      txz(2)=cmplx(xmax/2.,zmax/2.)
      txz(3)=cmplx(xmax/2.,zmax)
      txz(4)=cmplx(xmax,zmax)

      tetchr=-0.005
      tmout=60.
      nmout=3
      ntotal=18/nmout*nmout
      mplthp=1

c  mtype: = 1-dual, 2-hemispherical, 3-cone, 4-planetary,
c  5-unidirectional source
      mtype=2
      mdiff=0
      sigma=1.5*xzdelt
      mdloop=0

      angle(1)=-90.
      angle(2)=90.

      gamma=30.
      beta=0.
      phi=15.
```

```
c  mrsl=0 enables rsl; 1 enables phi
      mrsl=0
      rsl=0.
      sw=25.
      rp=25.

      call profil
      call dpmain

      stop
      end



      subroutine     profil
c  this section creates a piecewise linear profile
      common/etch1/ xz(450),xmax,zmax,npts,cxzl,cxzr,nadchk,nckout
      common/chkr/ sminx,sminz,smaxx,smaxz,xzdelt
      common/prfile/ txz(40),nt
      complex xz,txz,unit
      write(6,100) sminx,sminz,smaxx,smaxz,xzdelt
100   format(1h1,//,3x,6hsminx=,f7.4,/,3x,6hsminz=,f7.4,/,3x,
    &       6hsmaxx=,f7.4,/,3x,6hsmaxz=,f7.4,/,3x,7hxzdelt=,f7.4)
      ntemp=nt-1
      istart=1
      do 1 i=1,ntemp
      unit=(txz(i+1)-txz(i))/cmplx(cabs(txz(i+1)-txz(i)),0.)
c  guard against rounding error, figure 0.5 is arbitrarily chosen
      n=int(cabs(txz(i+1)-txz(i))/xzdelt-0.5)
      if ((istart+n) .ge. 450) go to 5
      xz(istart)=txz(i)
      do 2 j=1,n
      xz(istart+j)=xz(istart+j-1)+unit*cmplx(xzdelt,0.)
2     continue
      istart=istart+n+1
1     continue
      npts=istart
      xz(npts)=txz(nt)
      write(6,20) nt
20    format(//,3x,31htotal number of turning points=,i5)
      do 3 i=1,nt
      ux=real(txz(i))
      uz=aimag(txz(i))
      write(6,30) i,ux,uz
30    format(3x,14hturning point ,i4,4h x= ,f7.4,4h z= ,f7.4,
    &         7hmicrons)
3     continue
      write(6,40) npts
40    format(3x,13htotal points=,i5)
      return
5     write(6,50)
50    format(3x,38(1h*),15htoo many points)
      stop
      end
```

```
        subroutine     dpmain
c  this section controls the logic flow
        common/etch1/ xz(450),xmax,zmax,npts,cxzl,cxzr,nadchk,nckout
        common/metal/ tetchr,tmout,nmout,angle(2),delt,ntotal
        common/mtflag/ mcount,mdiff,mplthp,mdloop,mtype
        common/difusn/ rate(450),sigma
        complex xz,cxzl,cxzr,rate
        call initlz(mtype)
        call dpmesg(mtype)
        if (mdiff .eq. 1) write (6, 2) sigma
2       format(5x,24hsurface diffusion sigma=,f7.4,8h microns,//)
        mcount=0
        nastep=ntotal/nmout
        delt=tmout/float(ntotal)
        nadv=0
c  produce cards of the initial profile for a hp-plotter
        if (mplthp .eq. 1) call punch(0)
        call plot(0)
        do 10 iout=1, nmout
5       call shadow
        call advnce
        nadv=nadv+1
        call cheker
        if (mdloop .eq. 1) call deloop
        if (nadv .lt. nastep) go to 5
        if (mplthp .eq. 1) call punch(iout)
        call plot(iout)
        nadv=0
10      continue
        write(6,15) delt,ntotal,nastep
15      format(5x,31htime interval between advances=,f8.4,8h seconds,
     &      /,5x,25htotal number of advances=,i3,
     &        /,5x,35hnumber of advances between outputs=,i3,//)
        return
        end



        subroutine     initlz(mtype)
c  initialize all the parameters used in the subsequent sections
        common/system/ gamma,beta,phi,sw,rp,mrsl
        common/metal/ tetchr,tmout,nmout,angle(2),delt,ntotal
        common/planet/ dr,dl,dw,rsl,aiw,csthet
        common/sumcon/ c1,c2,c4,c5
c  "detrad" is a constant to convert degrees to radians
        detrad=3.1415926/180.
        if (mtype .ne. 1) go to 2
c  this is dual-directional discrete sources
        angle(1)=detrad*angle(1)
        angle(2)=detrad*angle(2)
        return

2       if (mtype .ne. 2) go to 3
c  this is a hemispherical vapor source
        angle(1)=detrad*angle(1)
        angle(2)=detrad*angle(2)
        return

3       if (mtype .ne. 3) go to 4
c  this is a cone source (beta=0, rsl=0), a special case of planetary
        gamma=detrad*gamma
        dr=sw*sin(gamma)
        dw=sw*cos(gamma)
        dl=dw+rp
        aiw=(dr**2+dl*dw)/(sqrt(dr**2+dw**2)*((dr**2+dl**2)**1.5))
        csthet=dl/((dr**2+dl**2)**0.5)
        angle(1) =-atan(dr/dl)
        angle(2) =-angle(1)
        return

4       if (mtype .ne. 4) go to 5
c  this is a planetary rotating source
        gamma=detrad*gamma
        beta=detrad*beta
        phi=detrad*phi

        dr=sw*sin(gamma)
        dw=sw*cos(gamma)
        if (mrsl .eq. 1) rsl=rp*tan(phi)
        dl=dw+rp
        aiw=(dr**2+dl*dw)/(sqrt(dr**2+dw**2)*((dr**2+dl**2)**1.5))
        csthet=dl/((dr**2+dl**2)**0.5)
        angle(1) =-abs(atan((dr-rsl)/dl))-beta
        angle(2) =atan((dr+rsl)/dl)+beta
c  constants c1,c2,c4,c5 are evaporation system dependent parameters
        c1=(dr**2+dw**2)**0.5
        c2=dr**2-rsl**2
        c4=dl*cos(beta)
        c5=sin(beta)
        call grotbl
        return

5       if (mtype .ne. 5) go to 10
c  this is a unidirectional source
        angle(1)=detrad*angle(1)
        angle(2)=angle(1)
        return

10      write(6,11)
11      format(3x,15(1h*),16hundefined source)
        stop
        end
```

```
      subroutine     dpmesg(mtype)
c  echo-print pertinent evaporation-system messages
      common/system/ gamma,beta,phi,sw,rp,mrsl
      common/planet/ dr,dl,dw,rsl,aiw,csthet
      common/metal/ tetchr,tmout,nmout,angle(2),delt,ntotal
      detrad=3.1415926/180.
      write(6,10)
10    format(////////,20x,38(1h*),15h run machine 6 ,38(1h*))
      write(6,15)
15    format(//,70(1h-),//,20x,9hversion 1,5x,17hseptember 8, 1979,
     &          //,70(1h-))
      write(6,20)
20    format(///,20x,42h--------- system message(runmc6) ---------,/)
      write(6,25) tetchr
25    format(5x,17hdeposition rate= ,f8.5,12h microns/sec,//)
      ang1=angle(1)/detrad
      ang2=angle(2)/detrad
      if (mtype .ne. 5) write(6,30) ang1,ang2
30    format(5x,15hincident angle=,f5.1,2x,f5.1,8h degrees,//)
      if (mtype .eq. 5) write(6,31) ang1
31    format(5x,15hincident angle=,f5.1,8h degrees,//)

c  print pertinent information about the source-substrate configuration
      if (mtype .ne. 1) go to 2
      write(6,35)
35    format(5x,24hdual evaporation sources,//)
      return

2     if (mtype .ne. 2) go to 3
      write(6,40)
40    format(5x,26hhemispherical vapor source,//)
      return

3     if (mtype .ne. 3) go to 4
      write(6,45)
45    format(5x,11hcone source,//)
      betemp=0.
      gatemp=gamma/detrad
      rsl=0.
      write(6,50) sw,rp,rsl,betemp,gatemp
50    format(5x,20hsystem axis length= ,f6.1,3h in,//,
     &         5x,20hplanet axis length= ,f6.1,3h in,//,
     &         5x,15hplanet radius= ,f4.1,3h in,//,
     &         5x,6hbeta= ,f5.1,8h degrees,//,
     &         5x,6hgama= ,f5.1,8h degrees,//)
      if (gamma .eq. 0.) go to 51
      return
51    write(6,52)
52    format(3x,5(1h*),37hfatal error: this is a unidirectional,
     &         22h source, input ignored,5(1h*))
      stop
```

```
4       if (mtype .ne. 4) go to 5
        write(6,55)
55      format(5x,25hplanetary rotating source,//)
        write(6,60) sw,rp
60      format(5x,20hsystem axis length= ,f6.1,3h in,//,
     &         5x,20hplanet asix length= ,f6.1,3h in,/)
        phtemp=phi/detrad
        if (mrsl .eq. 1) write(6,65) phtemp
65      format(5x,5hphi= ,f5.1,8h degrees,/)
        if (mrsl .eq. 0) write(6,70) rsl
70      format(5x,15hplanet radius= ,f4.1,3h in,/)
        betemp=beta/detrad
        gatemp=gamma/detrad
        write(6,75) betemp,gatemp
75      format(5x,6hbeta= ,f5.1,8h degrees,//,
     &         5x,6hgama= ,f5.1,8h degrees,//)
        if ((gamma.eq.0.).and.(beta.eq.0.)) go to 76
        return
76      if ((mrsl.eq.1).and.(phi.eq.0.)) go to 77
        if ((mrsl.eq.0).and.(rsl.eq.0.)) go to 77
        return
77      write(6,78)
78      format(3x,5(1h*),37hfatal error: this is a unidirectional,
     &         22h source, input ignored,5(1h*))
        stop

5       if (mtype .ne. 5) go to 90
        write(5,80)
80      format(5x,21hunidirectional source,//)
c  split incident angle apart so that shadow effect can be detected
        angle(1)=angle(1)-detrad*0.5
        angle(2)=angle(2)+detrad*0.5
        return

90      write(6,85)
85      format(3x,16hsource undefined,38(1h*))
        stop
        end



        subroutine      shadow
c  calculate range of incident angle for every point on the profile;
c  normally wi(i) is negative, and wf(i) is postive
        common/etch1/ xz(450),xmax,zmax,npts,cxzl,cxzr,nadchk,nckout
        common/metal/ tetchr,tmout,nmout,angle(2),delt,ntotal
        common/crack/ wi(450),wf(450)
        complex xz,cxzl,cxzr
c  pi is calculated for guarding against rounding error
        detrad=3.1415926/180.
        pi=detrad*180.
        do 2 i=1,npts
        xi=real(xz(i))
```

```
      zi=aimag(xz(i))
      wi(i)=angle(1)
      wf(i)=angle(2)
      do 2 j=1,npts
      xj=real(xz(j))
      zj=aimag(xz(j))
      if (j .eq. i) go to 2
      temp=(xj-xi)/sqrt((xi-xj)**2+(zi-zj)**2)
      if (abs(temp) .gt. 1.) temp=temp/abs(temp)
      wtemp=asin(temp)
      if (j .gt. i) go to 3
c  calculate initial incident angle, wi(i), measured between z-axis
c  and xz(j) to xz(i); whose upper limit is wf(i)
      if (zj .gt. zi) wtemp=-pi-wtemp
      if (wtemp .gt. wi(i)) wi(i)=wtemp
      if (wi(i) .gt. wf(i)) wi(i)=wf(i)
      go to 2
c  calculate final incident angle, wf(i), measured between z-axis
c  and xz(j) to xz(i); whose lower limit is wi(i)
3     if (zj .gt. zi) wtemp=pi-wtemp
      if (wtemp .lt. wf(i)) wf(i)=wtemp
      if (wf(i) .lt. wi(i)) wf(i)=wi(i)
2     continue
      return
      end



      subroutine     advnce
c  sum up the increment of each point in x- and z- direction for a
c  single time-step
      common/etch1/ xz(450),xmax,zmax,npts,cxzl,cxzr,nadchk,nckout
      common/metal/ tetchr,tmout,nmout,angle(2),delt,ntotal
      common/mtflag/ mcount,mdiff,mplthp,mdloop,mtype
      common/difusn/ rate(450),sigma
      complex xz,cxzl,cxzr,rate
      mcount=mcount+1
      call evrate(mtype)
      if (mdiff .eq. 1) call diff
c  guard against overflow
      if (abs(aimag(rate(1))).lt.1.0e-38) go to 1
c  calculate deposition rate internally for accuracy;despite 'tetchr'
c   was given
      if (mcount.eq.1)tetchr=tetchr/aimag(rate(1))
c  adjust left boundary as growth reference; this implies at the
c  first time-step growth rate at this point is 'tetchr'
      zt1=aimag(rate(1))
      rate(1)=cmplx(0.,zt1)

      do 4 i=1,npts
      xz(i)=xz(i)+cmplx(-tetchr*delt*real(rate(i)),
     &              tetchr*delt*aimag(rate(i)))
4     continue
```

```
      return
1     write(6,20) mcount
20    format(3x,10(1h*),37hnormal rate is too small to evaporate,
     &  1x,7hat adv=,i3,16h advance ignored)
      return
      end


      subroutine      evrate(mtype)
c  evaluate growth rate for various sources
      common/etch1/ xz(450),xmax,zmax,npts,cxzl,cxzr,nadchk,nckout
      common/metal/ tetchr,tmout,nmout,angle(2),delt,ntotal
      common/system/ gamma,beta,phi,sw,rp,mrsl
      common/planet/ dr,dl,dw,rsl,aiw,csthet
      common/difusn/ rate(450),sigma
      common/crack/ wi(450),wf(450)
      complex xz,cxzl,cxzr,rate
      detrad=3.1415926/180.
      if (mtype .ne. 1) go to 2
c  multi-directonal discrete sources
      do 20 i=1,npts
      iflag=0
c  if dual sources come from the same quadrant
      if ((wi(i)*wf(i)).gt.0.) iflag=1
      rate(i)=cmplx(0.,0.)
      if (iflag .eq. 1) go to 21
      if (wi(i) .gt. angle(1)) go to 22
      rate(i)=rate(i)+cmplx(sin(angle(1)),cos(angle(1)))
22    if (wf(i) .lt. angle(2)) go to 20
      rate(i)=rate(i)+cmplx(sin(angle(2)),cos(angle(2)))
      go to 20
21    if((wi(i).ge.0.).and.(wf(i).gt.angle(1)))
     &   rate(i)=cmplx(sin(angle(1)),cos(angle(1)) )
      if((wi(i).le.0.).and.(wi(i).lt.angle(2)))
     &   rate(i)=cmplx(sin(angle(2)),cos(angle(2)) )
      if((wi(i).eq.angle(1)).and.(wf(i).eq.angle(2))) rate(i)=cmplx(
     &   sin(angle(1))+sin(angle(2)),cos(angle(1))+cos(angle(2)) )
20    continue
c  guard against rounding errors, recalculate rate at left boundary;
c  because this is the important reference point
      angtmp=-90.*detrad
      if (angle(1) .eq. angtmp) rate(1)=rate(2)
      return

2     if (mtype .ne. 2) go to 3
c  hemispherical vapor source
      do 30 i=1,npts
      rate(i)=cmplx((cos(wi(i))-cos(wf(i)))/2.,
     &          (sin(wf(i))-sin(wi(i)))/2.)
30    continue
      return

3     if (mtype .ne. 3) go to 4
```

```
c  cone source, or a special case of planetary source
      c=dl/dr
      do 35 i=1,npts
      tem=aiw*csthet
      ctanwf=c*tan(wf(i))
      if (abs(ctanwf) .gt. 1.) ctanwf=ctanwf/abs(ctanwf)
      ctanwi=c*tan(wi(i))
      if (abs(ctanwi) .gt. 1.) ctanwi=ctanwi/abs(ctanwi)
      dz=abs(asin(ctanwf)-asin(ctanwi))
      dx=(1.-(ctanwf)**2)**0.5-(1.-(ctanwi)**2)**0.5
      rate(i)=cmplx(-tem*dx/c,tem*dz)
35    continue
      return

4     if (mtype .ne. 4) go to 5
c  planetary rotating source
c  calculate the integration of etch rate from substending solid angle
      do 40 i=1,npts
      dx2=evalue(wi(i),0)+evalue(wf(i),0)
      dz2=evalue(wi(i),1)+evalue(wf(i),1)
      rate(i)=cmplx(dx2,dz2)
40    continue
      return
5     if (mtype .ne. 5) go to 6
      do 50 i=1,npts
      rate(i)=cmplx(0.,0.)
      if ( (wi(i).eq.angle(1)).and.(wf(i).eq.angle(2)) )
     & rate(i)=cmplx( sin((angle(1)+angle(2))/2.),
     &                cos((angle(1)+angle(2))/2.) )
50    continue
      return

6     write(6,45)
45    format(3x,16hsource undefined,38(1h*))
      stop
      end




      subroutine     grotbl
c  create the growth rate table for planetary rotating source
      common/metal/ tetchr,tmout,nmout,angle(2),delt,ntotal
      common/savert/ xp(16),zp(16),xm(16),zm(16)
c  the data are stored in arrays: xp, zp, xm, and zm
      detrad=3.1415926/180.
      angmax=90.*detrad
      if ((abs(angle(1)).gt.angmax).or.(abs(angle(2)).gt.angmax))
     &     go to 5
c  if the solid angle subtended on the source is 0, then rate is 0
      do 10 i=1,16
      xp(i)=0.
      zp(i)=0.
      xm(i)=0.
      zm(i)=0.
10    continue
      if (angle(1).ge.0.) go to 30
      iang1=-int(angle(1)*10.)+1
      do 20 i=1,iang1
      xm(i)=evintg(-float(i)/10.,0.,0)
      zm(i)=evintg(-float(i)/10.,0.,1)
20    continue
      if (angle(2).le.0.) go to 15
30    iang2=int(angle(2)*10.)+1
      do 25 i=1,iang2
      xp(i)=evintg(0.,float(i)/10.,0)
      zp(i)=evintg(0.,float(i)/10.,1)
25    continue
c  following statements guard against spurious growth rate due to
c  numerical summation errors, so that curves seem smoother
15    if ((angle(1).ge.0.).or.(iang1.le.2)) go to 35
      do 40 i=3,iang1
      if (zm(i).lt.zm(i-1)) xm(i-1)=(xm(i)+xm(i-2))/2.
      if (zm(i).lt.zm(i-1)) zm(i-1)=(zm(i)+zm(i-2))/2.
40    continue
35    if ((angle(2).le.0.).or.(iang2.le.2)) return
      do 45 i=3,iang2
      if (zp(i).lt.zp(i-1)) xp(i-1)=(xp(i)+xp(i-2))/2.
      if (zp(i).lt.zp(i-1)) zp(i-1)=(zp(i)+zp(i-2))/2.
45    continue
      return
5     write(6,50)
50    format(3x,10(1h*),39hfatal error: impossible incident angles)
      stop
      end



      function        evalue(wang,i)
c  evaluate the growth rate by looking up and interpolating
c  the data table created before
      common/savert/ xp(16),zp(16),xm(16),zm(16)
c  i=0 means x-direction, 1 means z-direction
      if (wang .lt. 0.) go to 10
      iang=int(wang*10.)
      if (i .eq. 1) go to 5
      vlumax=xp(iang+1)
      if (iang .eq. 0) vlumin=0.
      if (iang .ge. 1) vlumin=xp(iang)
15    dvalue=abs(vlumax-vlumin)*(abs(wang*10.)-float(iang))
      if (vlumax .gt. 0.) evalue=vlumin+dvalue
      if (vlumax .le. 0.) evalue=vlumin-dvalue
      return
5     vlumax=zp(iang+1)
      if (iang .eq. 0) vlumin=0.
      if (iang .ge. 1) vlumin=zp(iang)
      go to 15
10    iang=int(-wang*10.)
      if (i .eq. 1) go to 20
      vlumax=xm(iang+1)
      if (iang .eq. 0) vlumin=0.
      if (iang .ge. 1) vlumin=xm(iang)
      go to 15
20    vlumax=zm(iang+1)
      if (iang .eq. 0) vlumin=0.
      if (iang .ge. 1) vlumin=zm(iang)
      go to 15
      end
```

```
      function        evintg(begin,end,i)
c  numerical summation is used, accuracy is proportional to n as well
c  as cpu time; figure 3 is arbitrarily chosen
      n=3
      dela=0.
      dn=(end-begin)/float(n)
      l=n-1
      do 3 j=1,l
      a=begin+dn*float(j)
      dela=dela+sum(a,i)
3     continue
      delar=dela+(sum(begin,i)+sum(end,i))/2.
      evintg=delar*dn
      return
      end
```

```
      function        sum(a,i)
c  calculate the growth rate function in x- or z- direction with
c  the given incident angle
      common/planet/ dr,dl,dw,rsl,aiw,csthet
      common/system/ gamma,beta,phi,sw,rp,mrsl
      common/sumcon/ c1,c2,c4,c5
c  i=0 means x-direction, 1 means z-direction; 'a' is incident angle
      c3=dl*tan(a-beta)
      up=(c2-c3*rsl+dl*dw)*(c5*c3-dl*c4)*(-dl/(cos(a-beta))**2)
      bt=c1*((c2+dl**2-2.*c3*rsl)**2)*((abs(c2-c3*rsl*2.-c3**2))
     &             **0.5)
c  neglect any spurious result that may cause overflow
      if (abs(bt) .lt. 1.0e-38) go to 1
      if (i .eq. 0) sum=up*tan(a)/bt
      if (i .eq. 1) sum=up/bt
      return
```

```
1       sum=0.
        return
        end



        subroutine     diff
c  this section handles surface migration resulting from hot substrate
        common/etch1/ xz(450),xmax,zmax,npts,cxzl,cxzr,nadchk,nckout
        common/mtflag/ mcount,mdiff,mplthp,mdloop,mtype
        common/difusn/ rate(450),sigma
        dimension tem(450)
        complex xz,cxzl,cxzr,rate,segmtl,tem,anorm,tnorm
        dev=3.*sigma
        nrange=npts/2
c  adjust rate by considering migration due to adjacent points
c  wherever within 3-sigma range, including both left and right sides
        do 30 i=1,npts
        if ((i.eq.1).or.(i.eq.npts)) go to 32
c  offl and offr define volume cell at local point xz(i)
        offl=cabs(xz(i)-xz(i-1))/2.
        offr=cabs(xz(i)-xz(i+1))/2.
        go to 34
c  adjust boundary points by mirror image
32      if (i.eq.1) offr=cabs(xz(2)-xz(1))/2.
        if (i.eq.1) offl=offr
        if (i.eq.npts) offl=cabs(xz(npts)-xz(npts-1))/2.
        if (i.eq.npts) offr=offl
34      tnorm=cmplx(real(anorm(i)),-aimag(anorm(i)))
        templ=gaussn(offl,sigma)
        tempr=gaussn(offr,sigma)
        tem(i)=rate(i)-tnorm*cmplx(cabs(rate(i)),0.)
     &                *cmplx((templ+tempr),0.)
        segmt=0.
c  calculate distribution from left side (within 3-sigma range)
        do 50 k=1,nrange
        if (i.le.k) indexl=k-i+2
        if (i.le.k) indexr=indexl-1
        if (i.le.k) go to 52
        indexl=i-k
        indexr=indexl+1
52      segmtl=xz(indexl)-xz(indexr)
        segmt=segmt+cabs(segmtl)
        wate=templ-gaussn(segmt,sigma)
        templ=gaussn(segmt,sigma)
        tem(i)=tem(i)+tnorm*cmplx(cabs(rate(indexl)),0.)
     &               *cmplx(wate,0.)
        if (segmt .gt. dev) go to 10
50      continue
1       write (6, 100)
100     format (3x,38(1h*),34hmigration range or sigma too great)
        stop
10      segmt=0.
c  calculate distribution from right side (within 3-sigma range)
```

```
      do 70 k=1,nrange
      if ((i+k).gt.npts) indexr=npts-(i+k-npts)
      if ((i+k).gt.npts) indexl=indexr+1
      if ((i+k).gt.npts) go to 72
      indexr=i+k
      indexl=indexr-1
72    segmtl=xz(indexr)-xz(indexl)
      segmt=segmt+cabs(segmtl)
      wate=tempr-gaussn(segmt,sigma)
      tempr=gaussn(segmt,sigma)
      tem(i)=tem(i)+tnorm*cmplx(cabs(rate(indexr)),0.)
     &             *cmplx(wate,0.)
      if (segmt .gt. dev) go to 30
70    continue
      go to 1
30    continue

      do 12 i=1,npts
      rate(i)=tem(i)
12    continue
      return
      end




      function        gaussn(r,sigma)
c  by using a prestored data table found in a mathematic textbook,
c  this section calculates the value of gaussian function with given
c  'r' and 'sigma'
      common/mtflag/ mcount,mdiff,mplthp,mdloop,mtype
      dimension area(30)
      data area/0.4602,0.4207,0.3821,0.3446,0.3085,0.2743,0.2420,
     &          0.2119,0.1841,0.1587,0.1357,0.1151,0.0968,0.0808,
     &          0.0668,0.0548,0.0446,0.0359,0.0287,0.0228,0.0179,
     &          0.0139,0.0107,0.0082,0.0062,0.0047,0.0035,0.0026,
     &          0.0019,0.0013/
c  calcualte area under standard distribution curve (gaussian curve)
      if (r .lt. 0.) go to 3
      x=r/sigma*10.
      ix=int(x)
      if (ix .eq. 0) go to 2
      if (ix .ge. 30) go to 1
      darea=(area(ix)-area(ix+1))*(1.-x+float(ix))
      gaussn=area(ix+1)+darea
      return
2     gaussn=0.4602+0.0398*(1.-x)
      return
1     gaussn=0.
      return
3     write (6, 200) mcount
200   format (3x,38(1h*),32hfatal error in segmt calculation,
     &   11h at xadv = ,i5)
```

```
        stop
        end



        function         anorm(i)
c  calculate normal unit vector at local point i
        common/etch1/ xz(450),xmax,zmax,npts,cxzl,cxzr,nadchk,nckout
        common/mtflag/ mcount,mdiff,mplthp,mdloop,mtype
        common/difusn/ rate(450),sigma
        complex xz,cxzl,cxzr,rate,anorm,xzl,xzr,xzt
        j=i
        if (j .eq. 1) j=j+1
        if (j .eq. npts) j=j-1
        xzl=xz(j)-xz(j-1)
        xzr=xz(j+1)-xz(j)
        xzt=(xzl+xzr)*cmplx(0.,-1.)
        anorm=xzt/cabs(xzt)
        return
        end




        function         amigr(i)
c  calcualte the normal componet of rate(i)
        common/etch1/ xz(450),xmax,zmax,npts,cxzl,cxzr,nadchk,nckout
        common/mtflag/ mcount,mdiff,mplthp,mdloop,mtype
        common/difusn/ rate(450),sigma
        complex xz,cxzl,cxzr,rate,anorm
        amigr=abs(real(rate(i)*anorm(i)))
        return
        end


        subroutine       cheker
c  adjust string legths of the profile by adding or deleting points
        common/etch1/ xz(450),xmax,zmax,npts,cxzl,cxzr,nadchk,nckout
        common/chkr/ sminx,sminz,smaxx,smaxz,xzdelt
        complex xz,cxzl,cxzr
c  remove those points which are outside of the left or right
c  boundaries
3       segmtr=real(xz(2)-xz(1))
        if (segmtr .gt. 0.) go to 2
        ntmp=npts-1
        do 1 i=2,ntmp
        xz(i)=xz(i+1)
1       continue
        npts=npts-1
        go to 3
2       segmtl=xmax-real(xz(npts))
```

```
      if (segmtl .gt. 0.) go to 4
      npts=npts-1
      go to 2
c  assume each time checker adds less than half of the total points
c  for convenience, figure 'npts/2' is arbitrarily chosen
4     ntemp=npts+npts/2
      do 10 i=2,ntemp
      if(i.gt.npts) go to 6
      segmtx=abs(real(xz(i)-xz(i-1)))
      segmtz=abs(aimag(xz(i)-xz(i-1)))
      if ((segmtx.lt.sminx).and.(segmtz.lt.sminz)) call delete(i)
      if ((segmtx.gt.smaxx).or.(segmtz.gt.smaxz)) call add(i)
10    continue
6     if ((xmax-real(xz(npts))) .gt. smaxx) go to 5
      return
5     npts=npts+1
      xnpts=(real(xz(npts-1))+xmax)/2.
      xz(npts)=cmplx(xnpts,aimag(xz(npts-1)))
      return
      end



      subroutine     delete(i)
c  delete local point xz(i), and update all other points
      common/etch1/ xz(450),xmax,zmax,npts,cxzl,cxzr,nadchk,nckout
      complex xz,cxzl,cxzr
      if (i .eq. npts) go to 16
      iend=npts-1
      do 14 j=i,iend
      xz(j)=xz(j+1)
14    continue
16    npts=npts-1
      return
      end



      subroutine     add(i)
c  add one point between xz(i) and xz(i-1), and update all other points
      common/etch1/ xz(450),xmax,zmax,npts,cxzl,cxzr,nadchk,nckout
      complex xz,cxzl,cxzr
      do 15 k=i,npts
      l=npts+i-k
      if ((l+1) .gt. 450) go to 20
      xz(l+1)=xz(l)
15    continue
      xz(i)=(xz(i+1)+xz(i-1))*cmplx(0.5,0.)
      npts=npts+1
c  skip the next point for checking to avoid possible infinitive loop
      i=i+1
      return
20    write(6,25)
25    format(3x,38(1h*),26hcannot add any more points)
```

```
      end



      subroutine     deloop
c  delete all possible loops (after "develop machine")
c  spurious loops very seldomly exist; for saving computer time, the
c  user should not use it until he finds loops in output;
c  only for completeness, this section is here!
      common/etch1/ xz(450),xmax,zmax,npts,cxzl,cxzr,nadchk,nckout
      common/chkr/ sminx,sminz,smaxx,smaxz,xzdelt
      complex xz,cxzl,cxzr
      nstart=4
      nend=4
      nstep=1
      n=nstart
1     n=n+1
      if (n .ge. (npts-nend)) return
      m=n+nstep
c  n is to the left of m, check whether n,n+1 and m,m+1 intersect
2     m=m+1
      if (m .ge. (npts-nend)) go to 1
      xm=real(xz(m))
      zm=aimag(xz(m))
      xn=real(xz(n))
      zn=aimag(xz(n))
c  check whether m is within minimum range of possible intersection
      if ( (abs(xm-xn).gt.smaxx) .or. (abs(zm-zn).gt.smaxz) ) go to 2
      xmp1=real(xz(m+1))
      zmp1=aimag(xz(m+1))
      xnp1=real(xz(n+1))
      znp1=aimag(xz(n+1))
      if ( (abs(xmp1-xm).lt.1.e-38) .or. (abs(xnp1-xn).lt.1.e-38) )
     &          go to 2
      slopem=(zmp1-zm)/(xmp1-xm)
      slopen=(znp1-zn)/(xnp1-xn)
      rincpm=(xmp1*zm-xm*zmp1)/(xmp1-xm)
      rincpn=(xnp1*zn-xn*znp1)/(xnp1-xn)
      slope=slopen-slopem
      if (abs(slope) .lt. 1.e-38) go to 2
c  n,n+1 and m,m+1 intersect at (xinter,zinter)
      xinter=(rincpm-rincpn)/slope
      zinter=slopen*xinter+rincpn
c  whether (xinter,zinter) is in line segment n,n+1
      if( (xinter.lt.amax1(amin1(xn,xnp1),amin1(xm,xmp1))) .or.
     &     (xinter.gt.amin1(amax1(xn,xnp1),amax1(xm,xmp1))) .or.
     &     (zinter.lt.amax1(amin1(zn,znp1),amin1(zm,zmp1))) .or.
     &     (zinter.gt.amin1(amax1(zn,znp1),amax1(zm,zmp1))) )go to 2
      xz(n+1)=cmplx(xinter,zinter)
c  delte loop and update other points
      jstart=n+2
      jstop=npts-(m-(n+1))
      do 3 j=jstart, jstop
      jj=j+m-n-1
      xz(j)=xz(jj)
3     continue
      npts=jstop
      n=n+1
      go to 1
      end



      subroutine     plot(ioutpt)
c  plot the output profiles (after "develop machine")
      common/etch1/ xz(450),xmax,zmax,npts,cxzl,cxzr,nadchk,nckout
      common/metal/ tetchr,tmout,nmout,angle(2),delt,ntotal
      common/scrach/ x(450),z(450)
      common iplt(123,99)
c  the next statement works on pdp-11 computer, it may not work on
c  the other computers
      logical*1 iplt
      dimension outtim(26),ichar(26)
      complex xz,cxzl,cxzr
      data iblank,iplus,istar,izero/1h ,1h+,1h*,1h0/
      data ichar/1ha,1hb,1hc,1hd,1he,1hf,1hg,1hh,1hi,1hj,1hk,1h1,1hm,
     &  1hn,1ho,1hp,1hq,1hr,1hs,1ht,1hu,1hv,1hw,1hx,1hy,1hz/
      if (ioutpt .gt. 26) go to 99
      if (ioutpt .ne. 0) go to 1
      zt=0.
      zb=zmax
      xr=xmax
      xl=0.
      xaxis=(zmax/xmax)*96.
      if(xaxis .gt. 96.) xaxis=96.
      nxaxis=int(xaxis+3.5)
      do 2 l=1,99
      do 2 k=1,123
      iplt(k,l)=iblank
2     continue
      do 3 k=1,123
      iplt(k,1)=istar
      iplt(k,nxaxis)=istar
3     continue
      do 4 l=2,nxaxis
      iplt(1,l)=istar
      iplt(123,l)=istar
4     continue
      iplt(2,2)=izero

1     if(ioutpt.ne.0)outtim(ioutpt)=tmout/float(nmout)*float(ioutpt)
      do 8 k=1,npts
      tempx=real(xz(k))-xl
      nx=int((tempx/(xr-xl))*120.+2.00001)
      tempz=aimag(xz(k))-zt
      if(tempz.ge.zmax)nzflg=1
      nz=int((tempz/(zb-zt))*xaxis+2.00001)
      if((ioutpt.eq.0).and.(nx.gt.1).and.(tempx.le.xmax).and.(nz.gt.1)
     & .and.(tempz.le.zmax))iplt(nx,nz)=iplus
      if((ioutpt.ne.0).and.(nx.gt.1).and.(tempx.le.xmax).and.(nz.gt.1)
     & .and.(tempz.le.zmax))iplt(nx,nz)=ichar(ioutpt)
 8    continue

      if(ioutpt.ne.nmout) return
      write(6,9) xl,xr,zt,zb,(ichar(iout),outtim(iout),iout=1,nmout)
 9    format(1hx,9hx left = , f8.4,8h microns,/
     &      1x,9hx right =, f8.4,8h microns,/
     &      1x, 9hz top =  , f8.4, 8h microns,/
     &      1x, 11hz bottom = , f6.4,8h microns,
     &      //,1x, 12hsymbol  time,/,
     &      (5x, a1, f7.1,4h sec) )
      write(6,20) ((iplt(i,j),i=1,123),j=1,99)
 20   format( (8x, 123a1) )
      write(6,22) npts
 22   format(8x,13htotal points=,i5)
      return
 99   write(6,98)
 98   format(3x,15(1h*),40hnumber of outputs should be less than 26)
      stop
      end


      subroutine      punch(ioutpt)
c  produce cards for a hp-plotter (after "develop machine")
      common/etch1/ xz(450),xmax,zmax,npts,cxzl,cxzr,nadchk,nckout
      common/metal/ tetchr,tmout,nmout,angle(2),delt,ntotal
      common/scrach/ x(450),z(450)
      complex xz,cxzl,cxzr
      xl=.0
      zb=-zmax
      zt=.0
      if (ioutpt.ne.0) go to 8
      ntemp=nmout+1
      rnout=float(ntemp)
      write(6,1) xl,xmax,zb,zt,rnout
 1    format(/,4(1x,f8.5),/,1x,f8.5)
 8    do 10 i=1,npts
      x(i)=real(xz(i))
      z(i)=-aimag(xz(i))
 10   continue
      rnpts=float(npts)
      write(6,2) rnpts
 2    format(1x,f9.5)
      write(6,3) ((x(i),z(i)),i=1,npts)
 3    format(11(1x,f6.3))
      return
      end
```

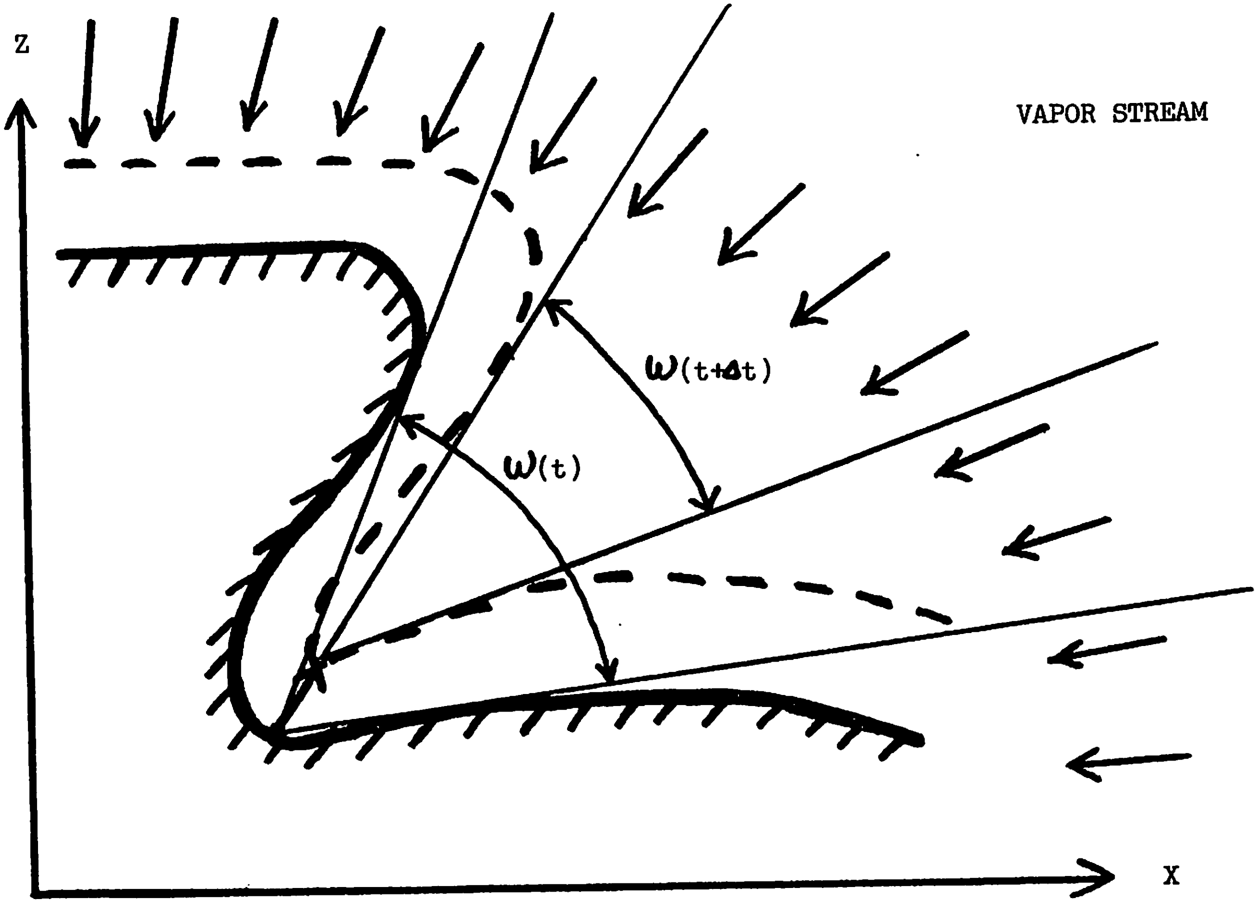


Fig. 1. The solid angle viewed by each point varies in time.

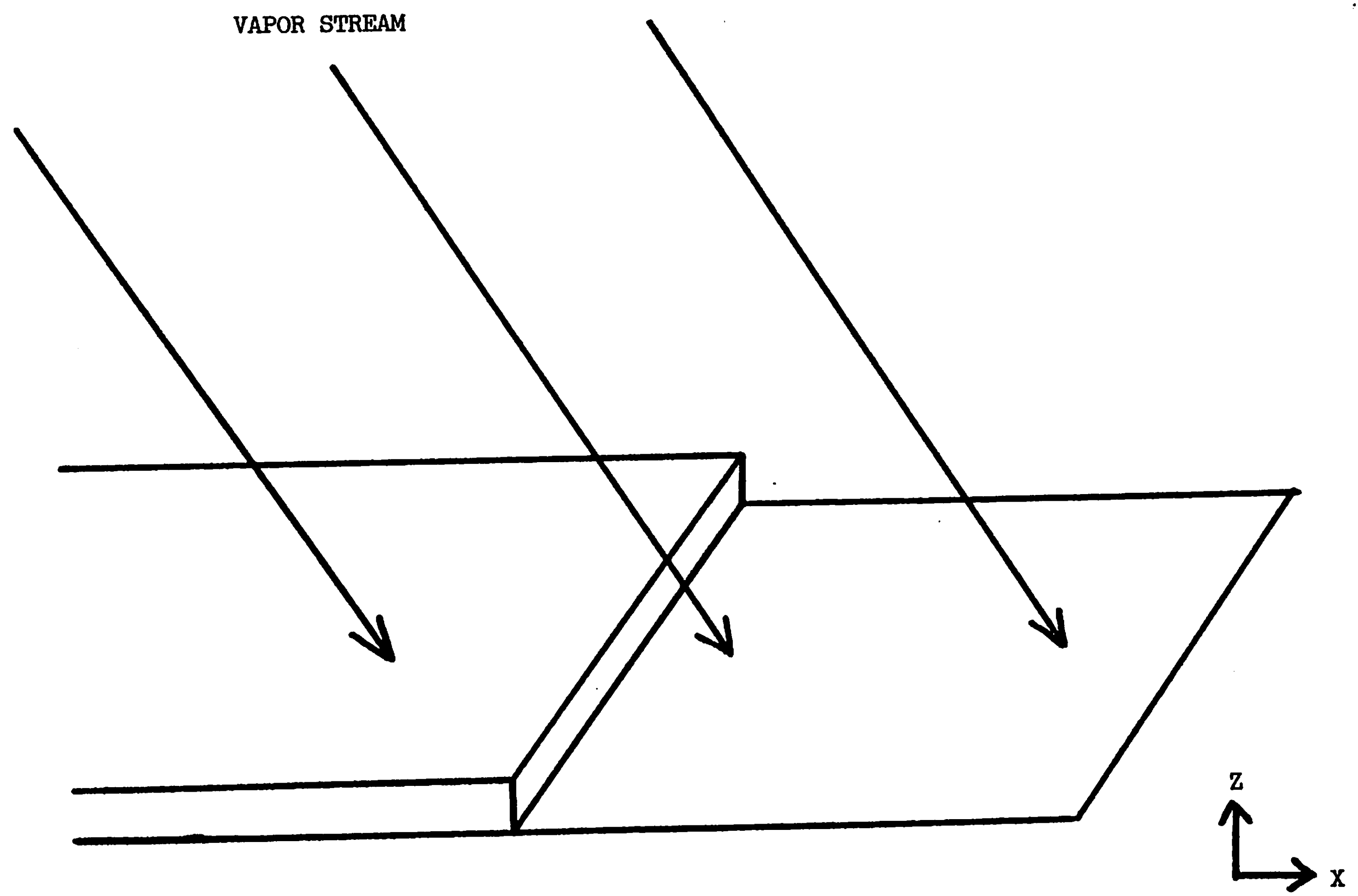


Fig. 2. A step profile with a unidirectional source

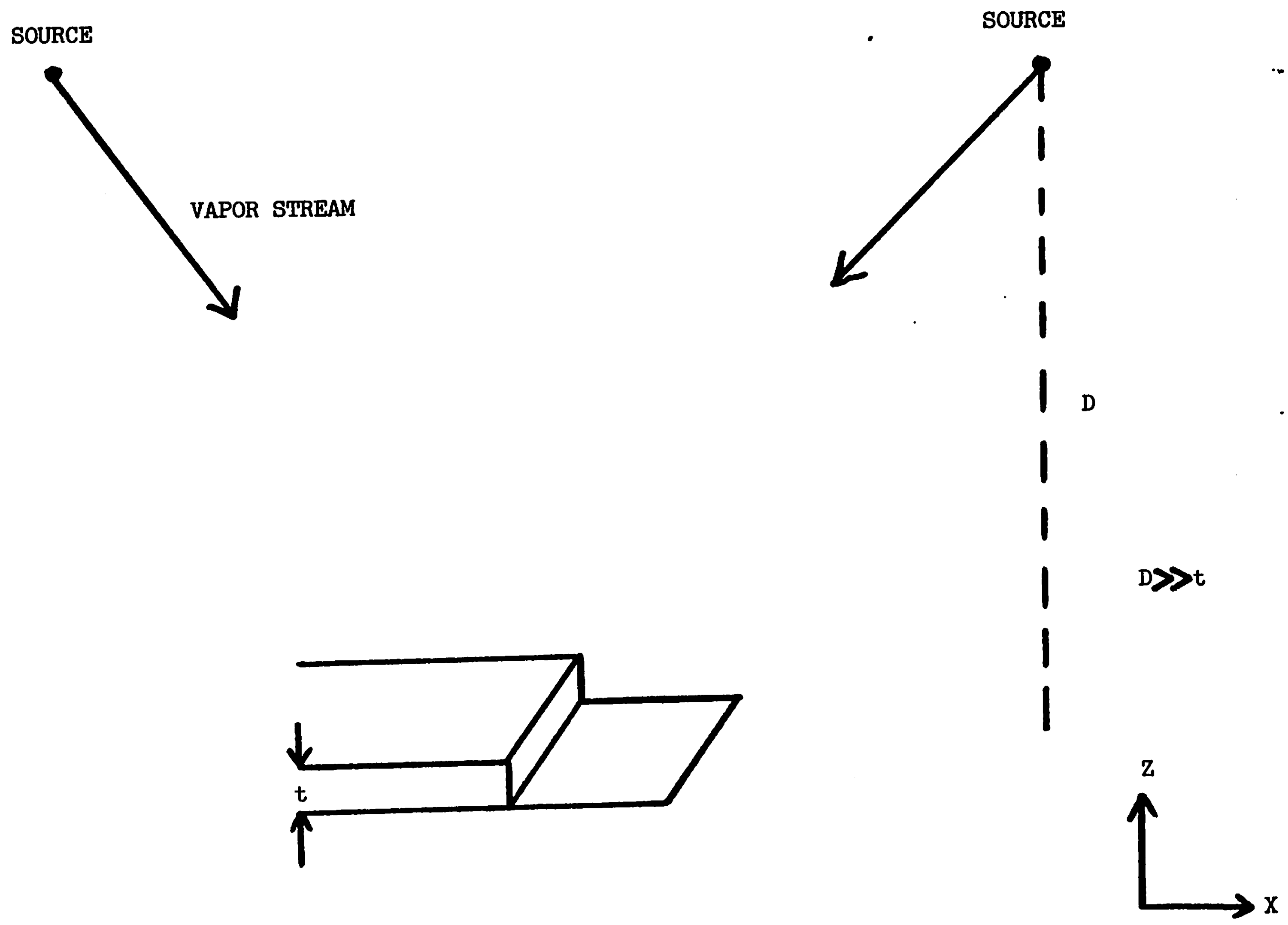


Fig. 3. A step profile with dual evaporation sources

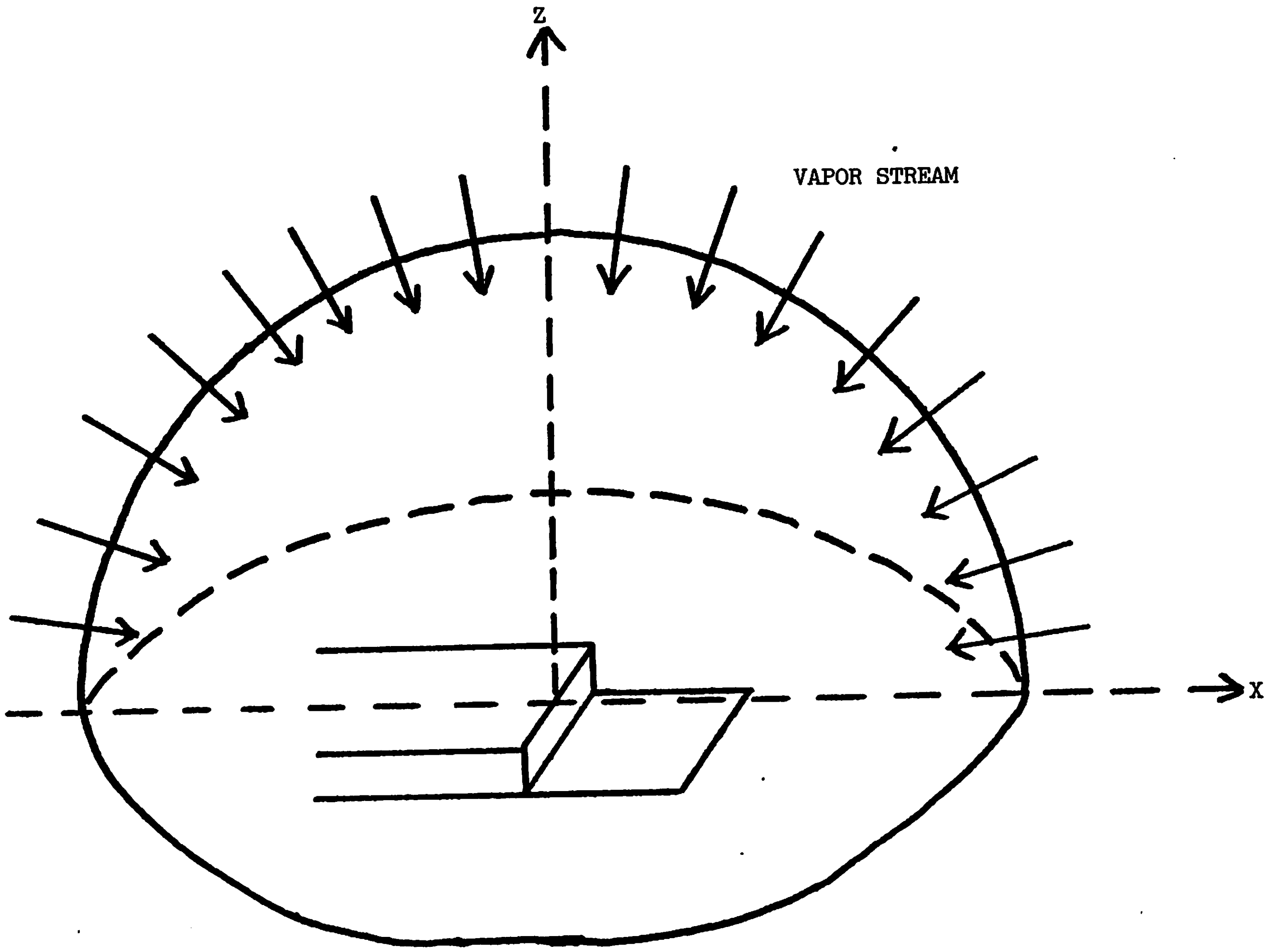


Fig. 4. A step profile with a hemispherical vapor source

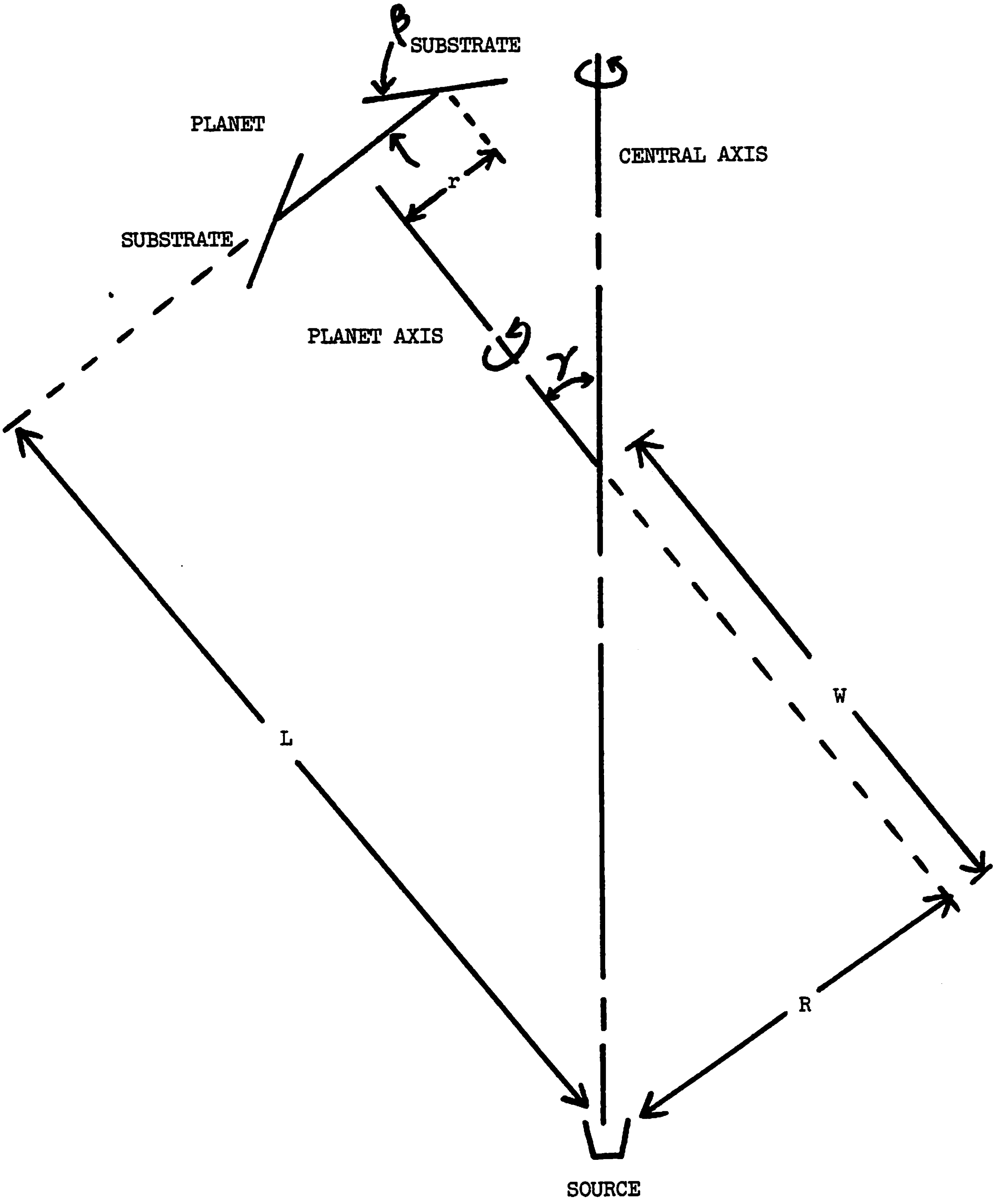


Fig. 5. Schematic planetary evaporator geometry

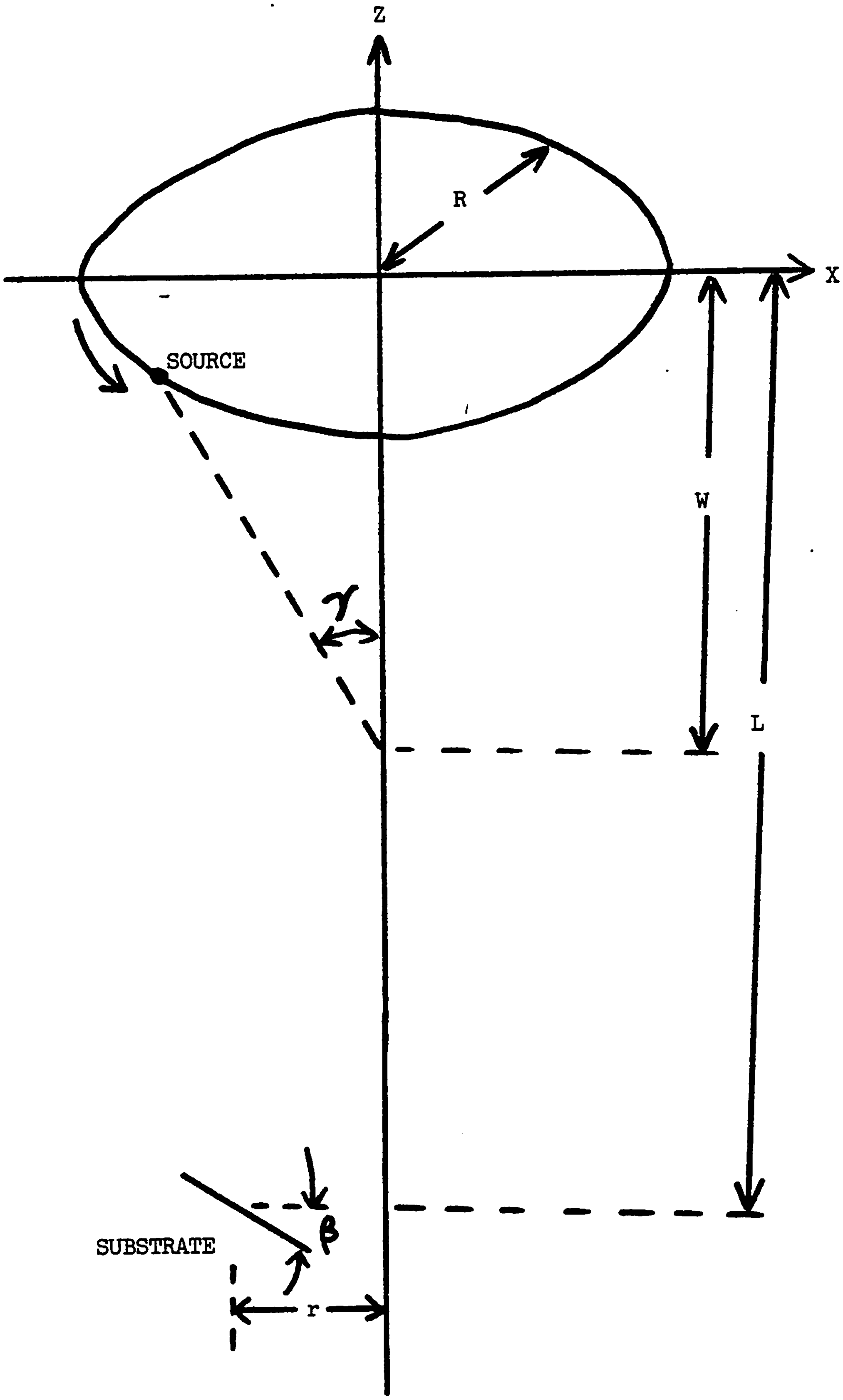


Fig. 6. Geometric relationship of source to substrate in a planetary evaporator

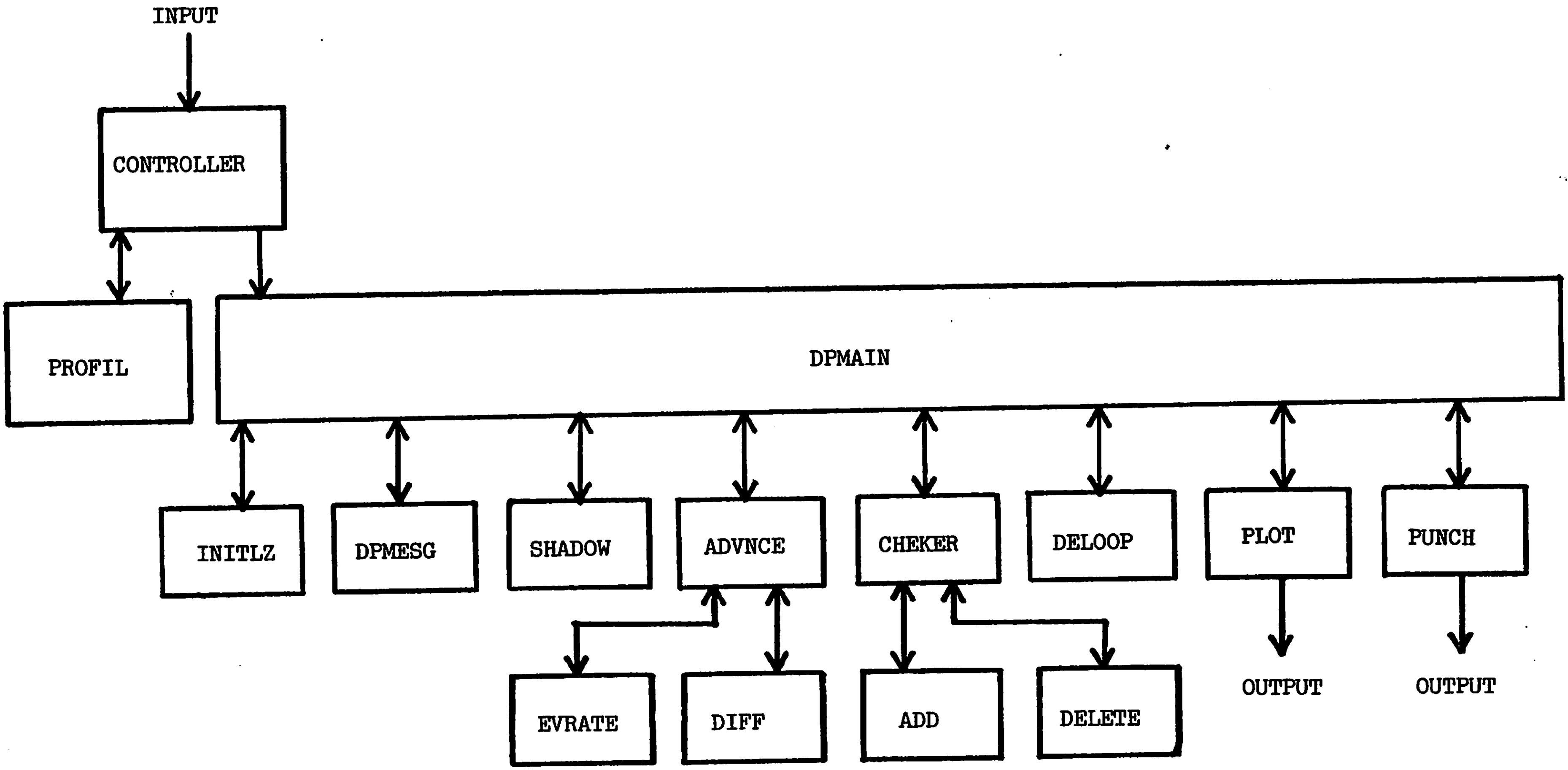


Fig. 7. Modular flowchart of the metalization machine

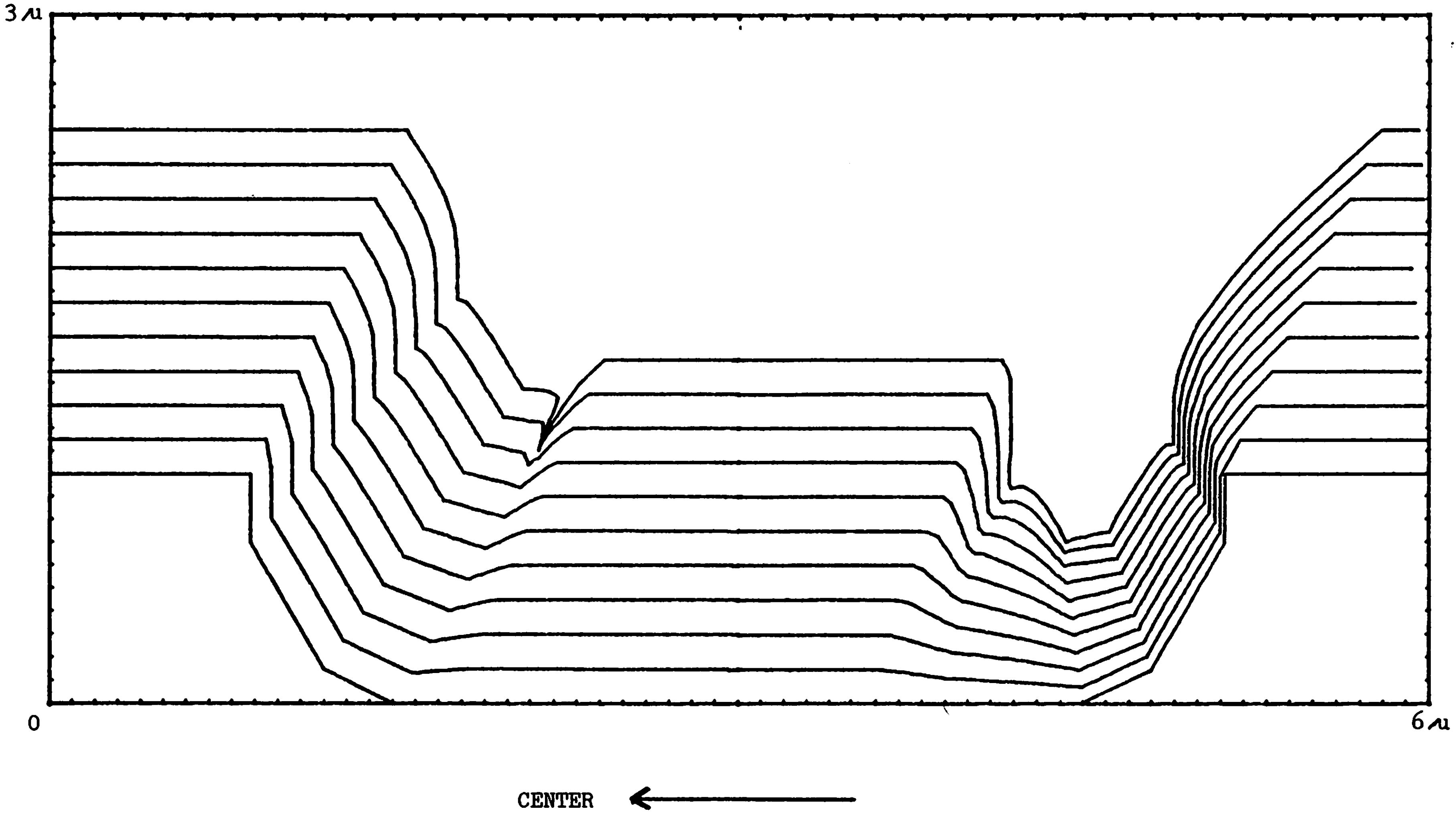


Fig. 8. Simulation result of a typical planetary evaporator

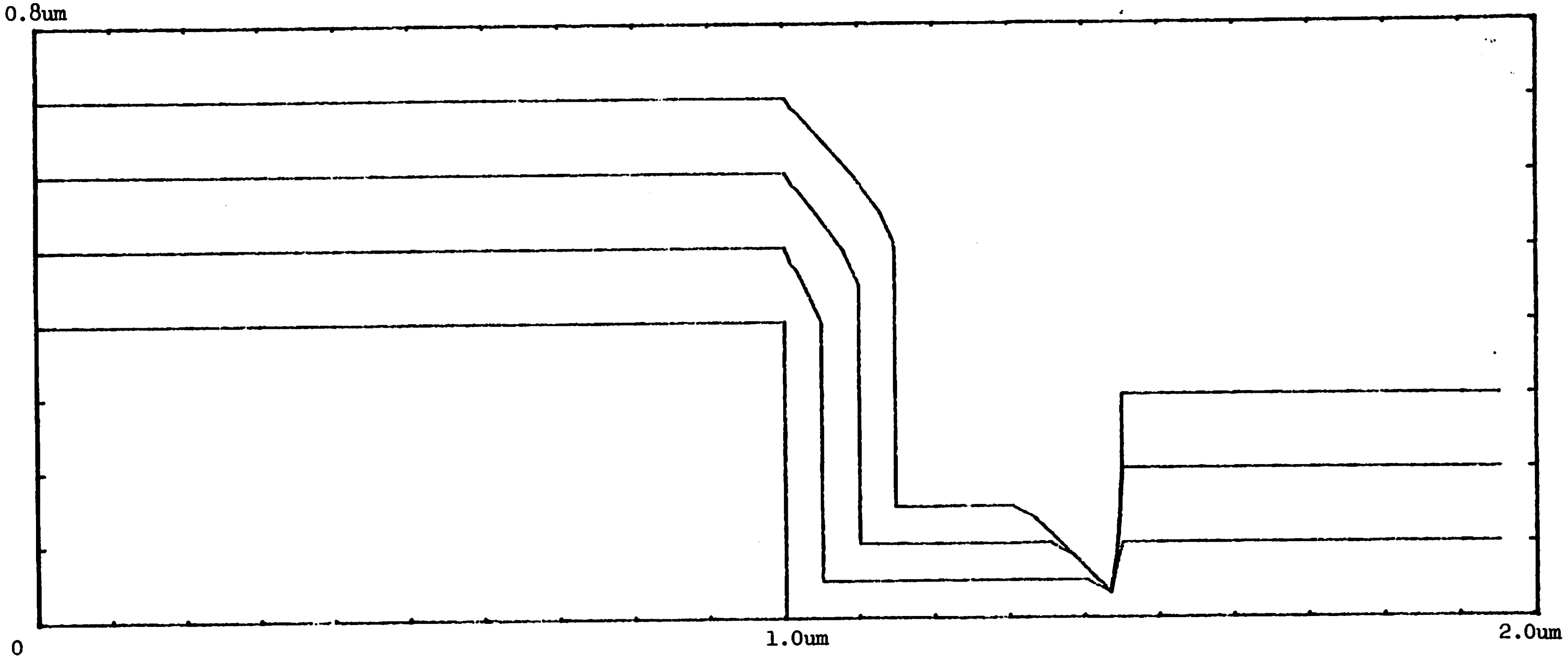


Fig. 9. Simulation result of dual evaporation sources with incident angles = ±45°

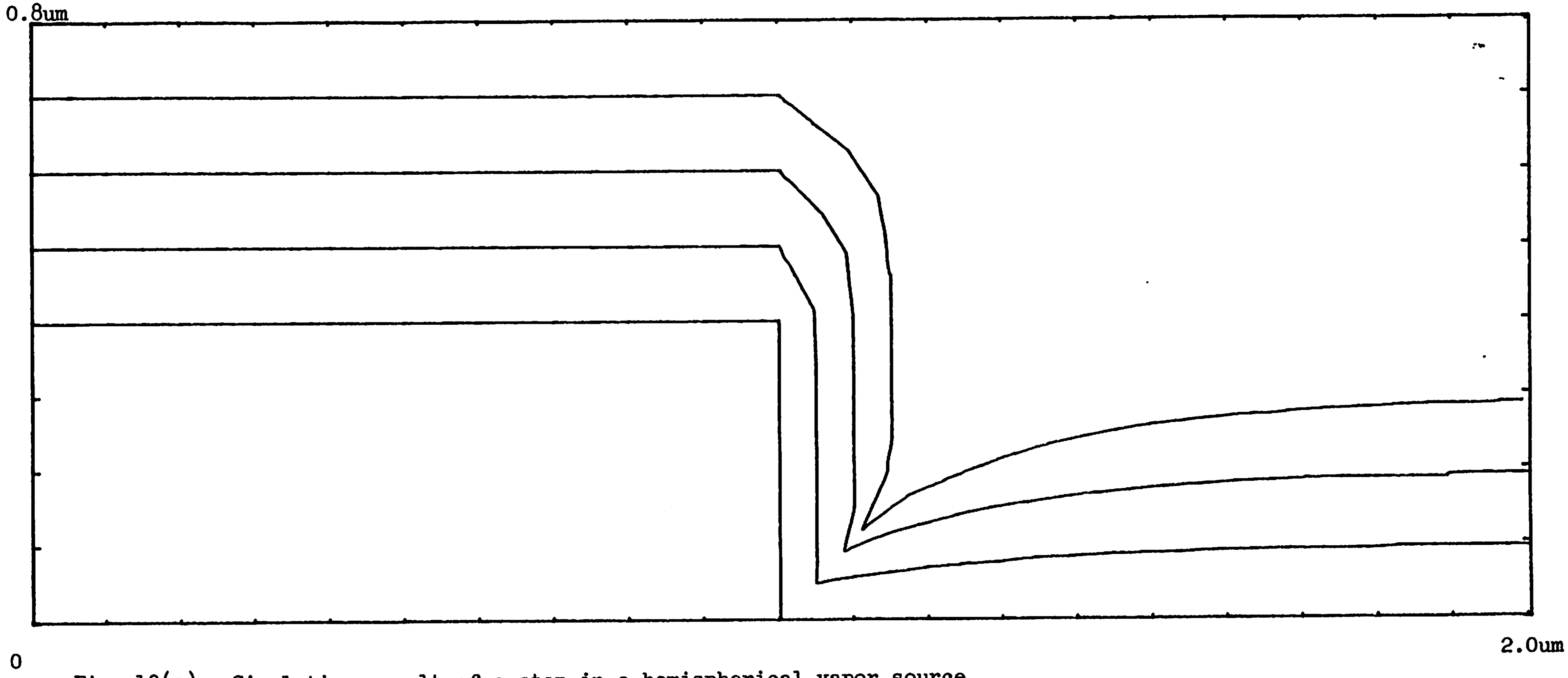


Fig. 10(a). Simulation result of a step in a hemispherical vapor source

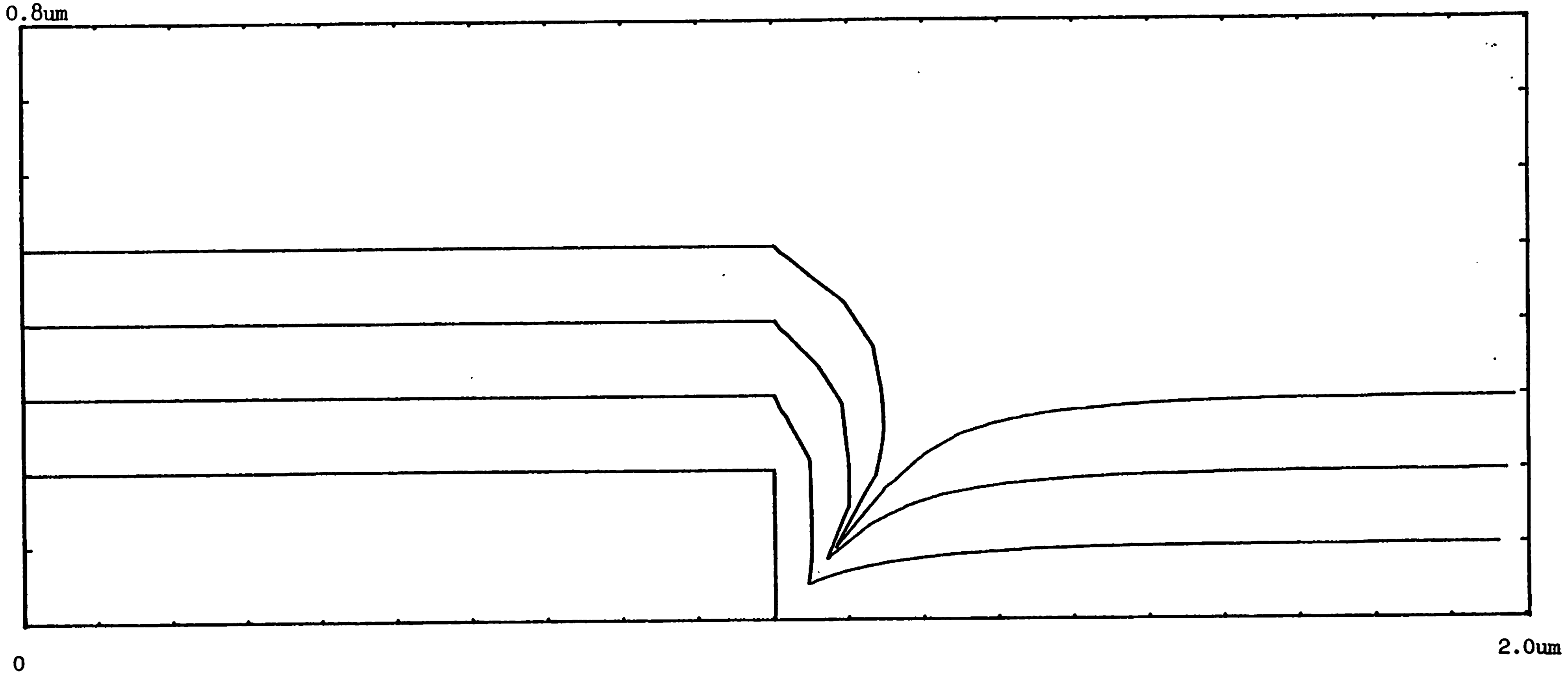


Fig. 10(b). Simulation result of a smaller step in a hemispherical vapor source

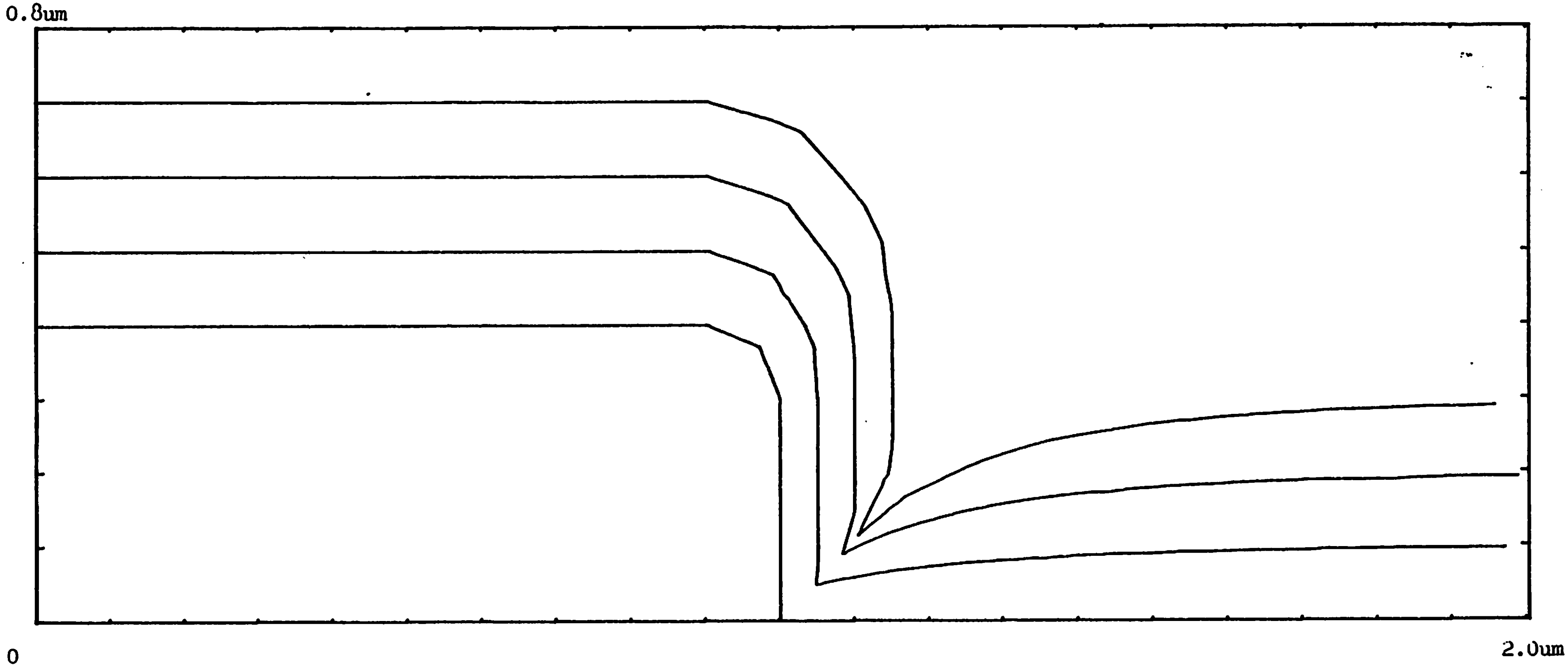


Fig. 11. Simulation result of a step with rounded top in a hemispherical vapor source

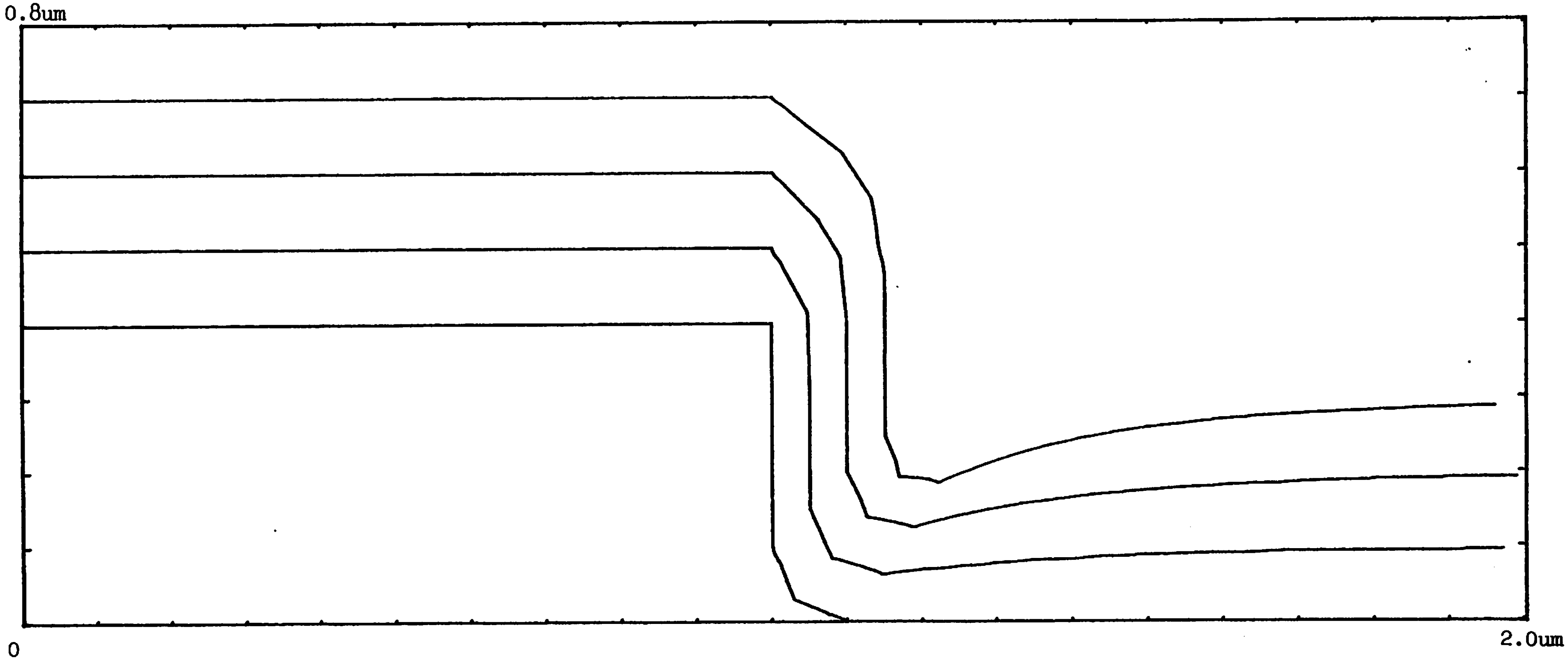


Fig. 12. Simulation result of a step with rounded bottom in a hemispherical vapor source

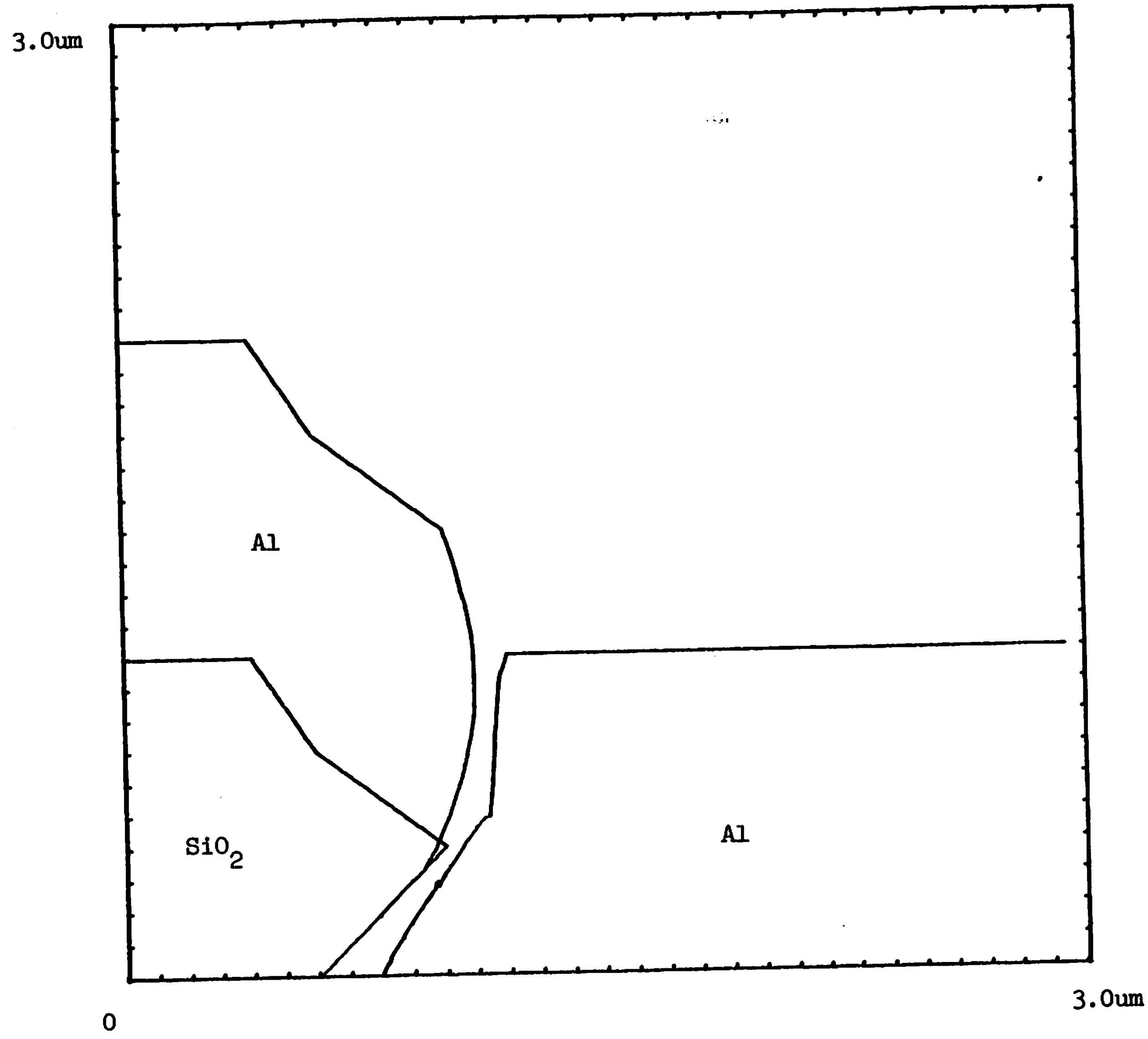


Fig. 13(a) Metal layer is deposited with a planetary source. Discontinuities occur at SiO$_2$ island edge caused by the shadowing effect.

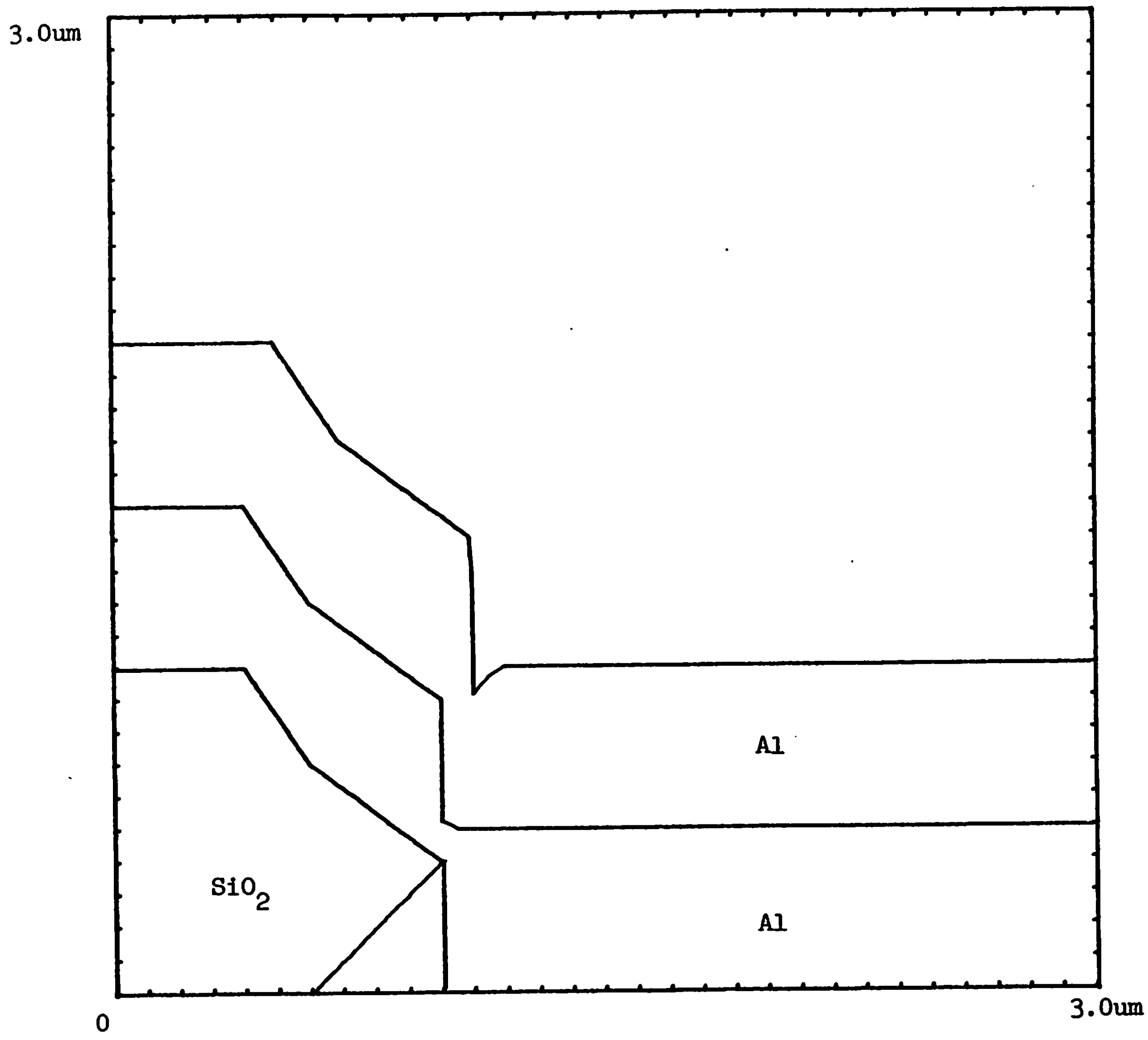


Fig. 13(b) Simulation result of two level metalization. The first level is deposited in a unidirectional source with vertical incidence. The second level is evaporated with a planetary source. A void is formed at the base of the step.

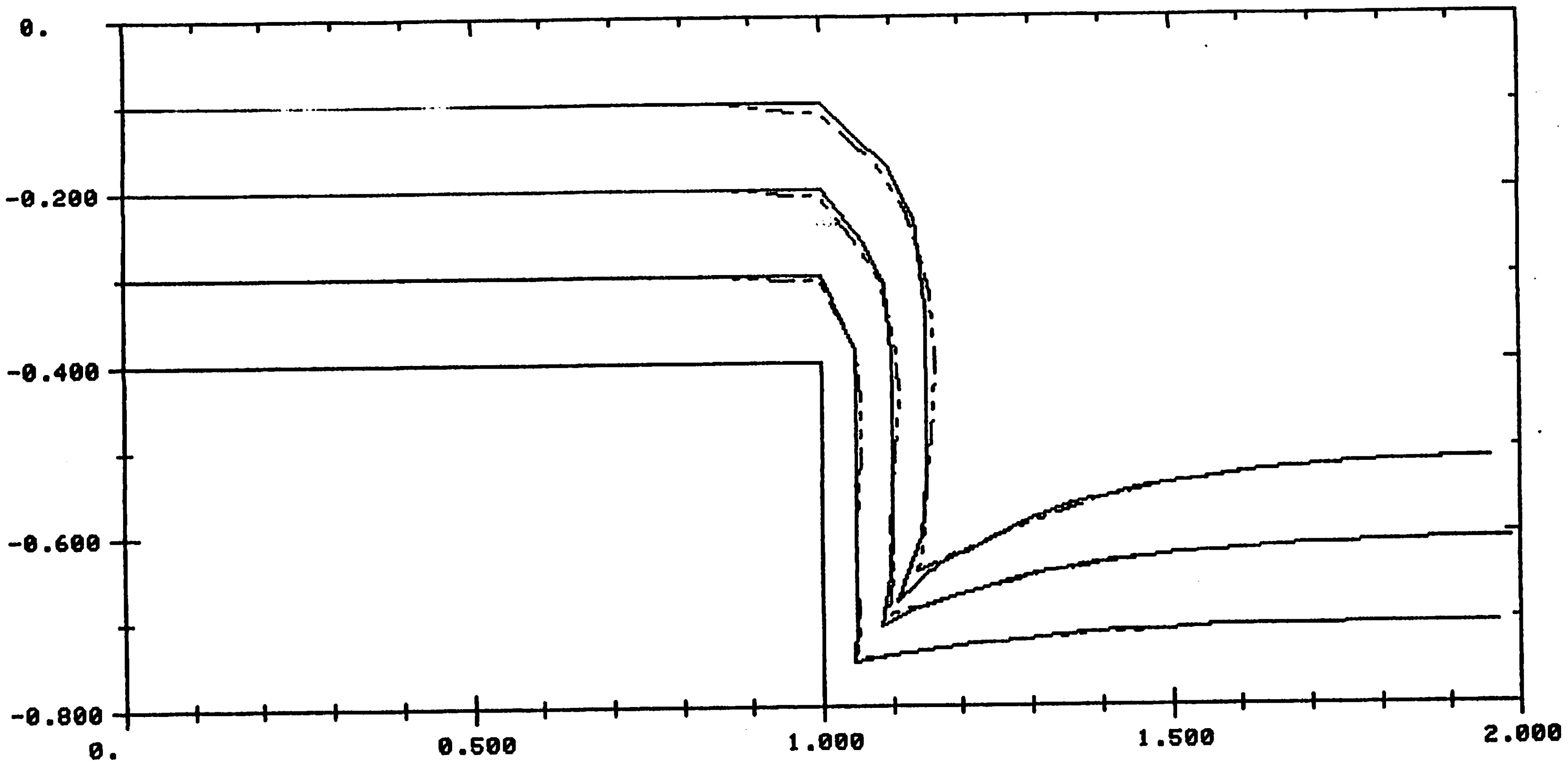


Fig. 14. Simulation result of a step in a hemispherical vapor source. Solid line is the deposition under room temperature. Dash line is the same deposition but with elevated substrate temperature. The crack is filled by material migration.

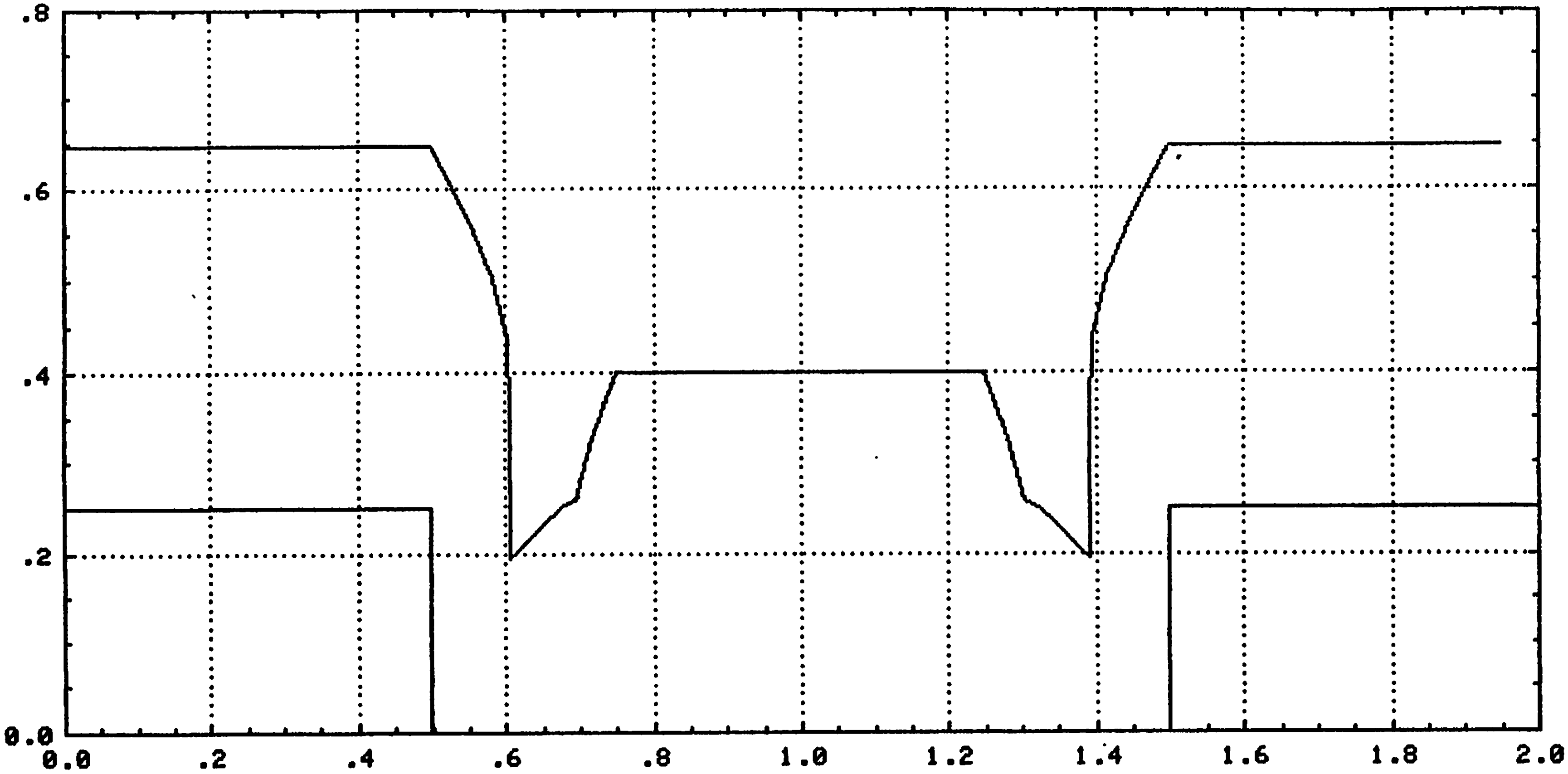


Fig. 15(a) Simulation result of a symmetrical step coverage deposited in room temperature.

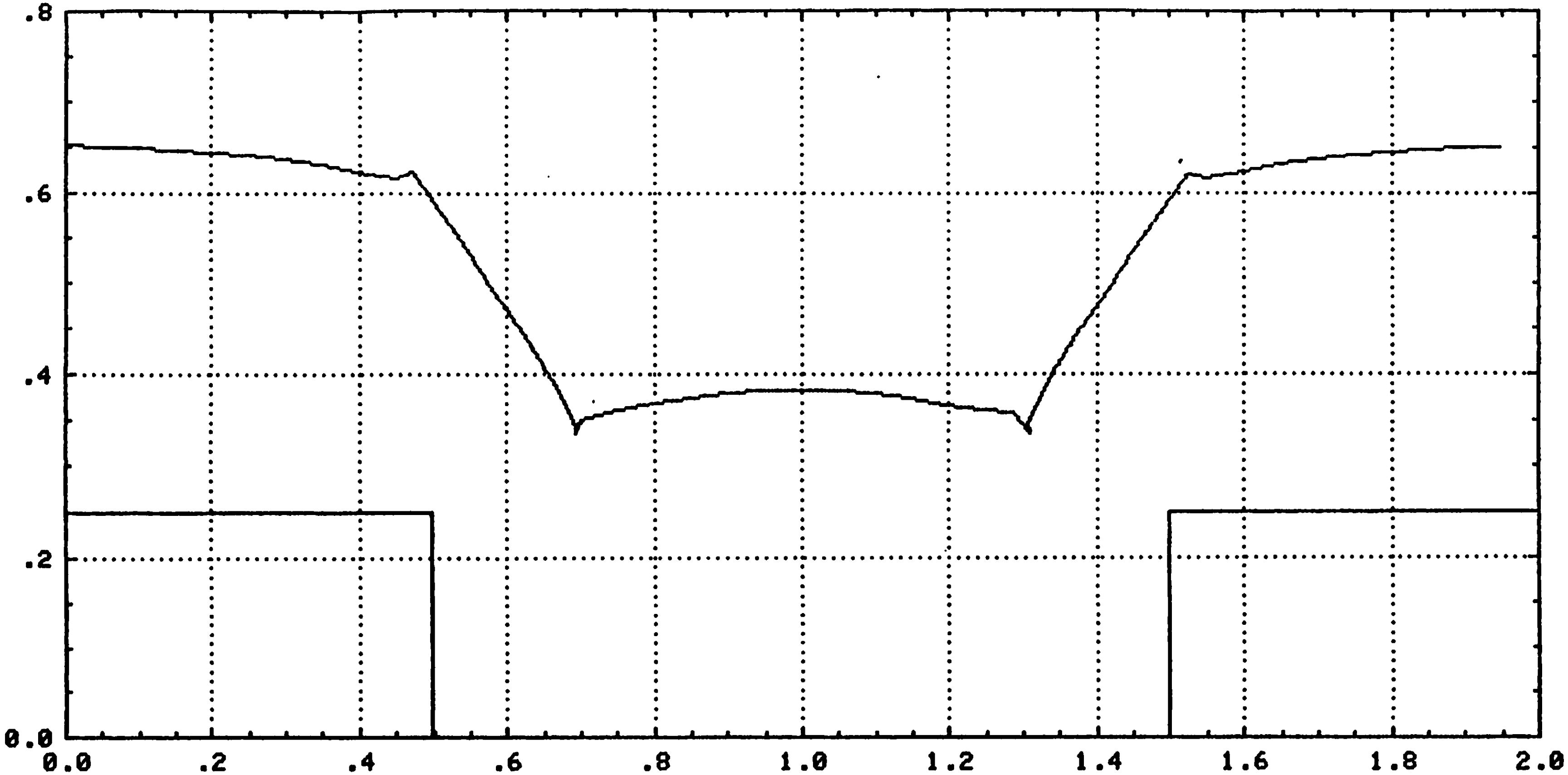


Fig. 15(b) Simulation result of a symmetrical step coverage deposited on a hot substrate.

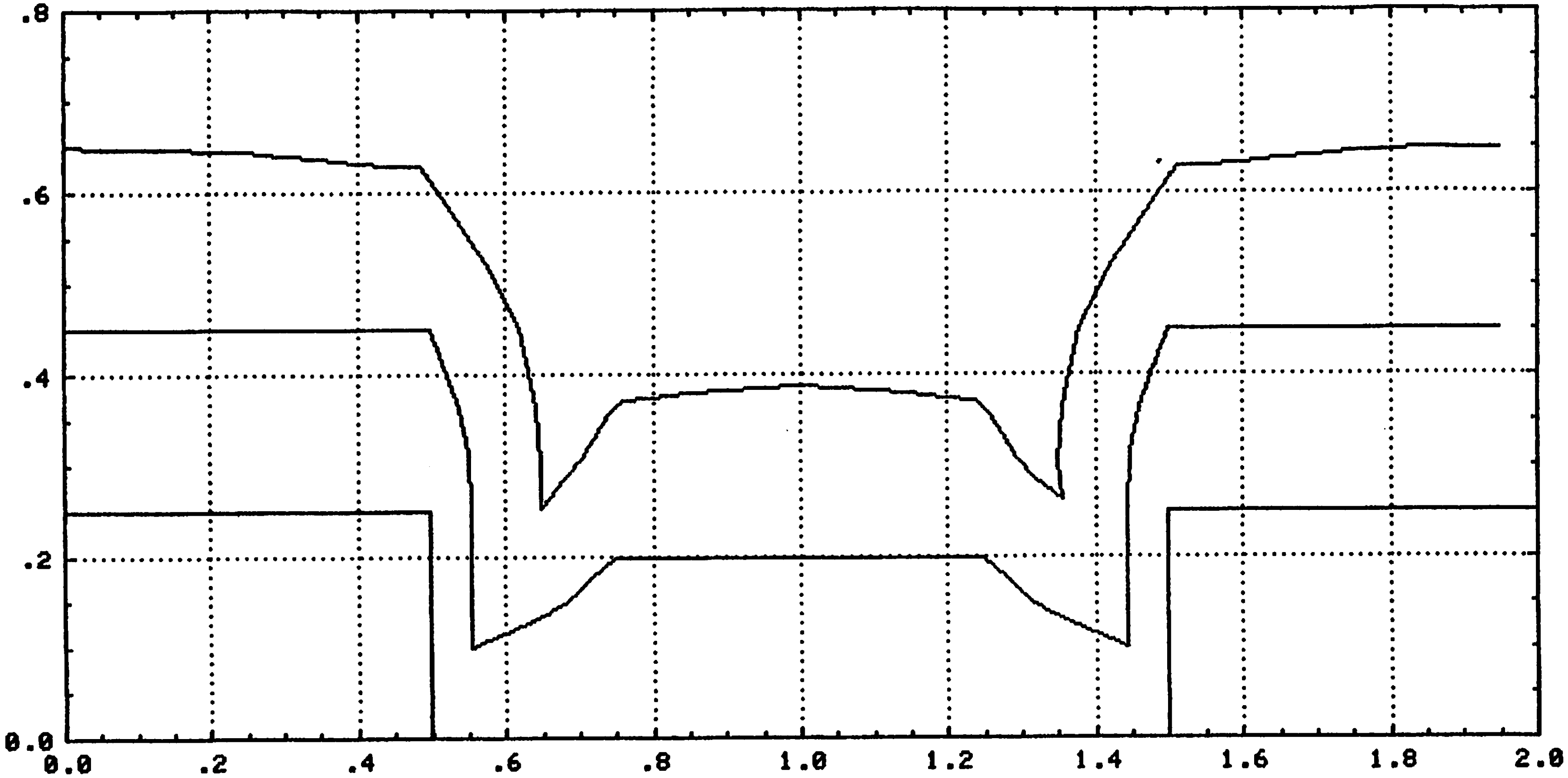
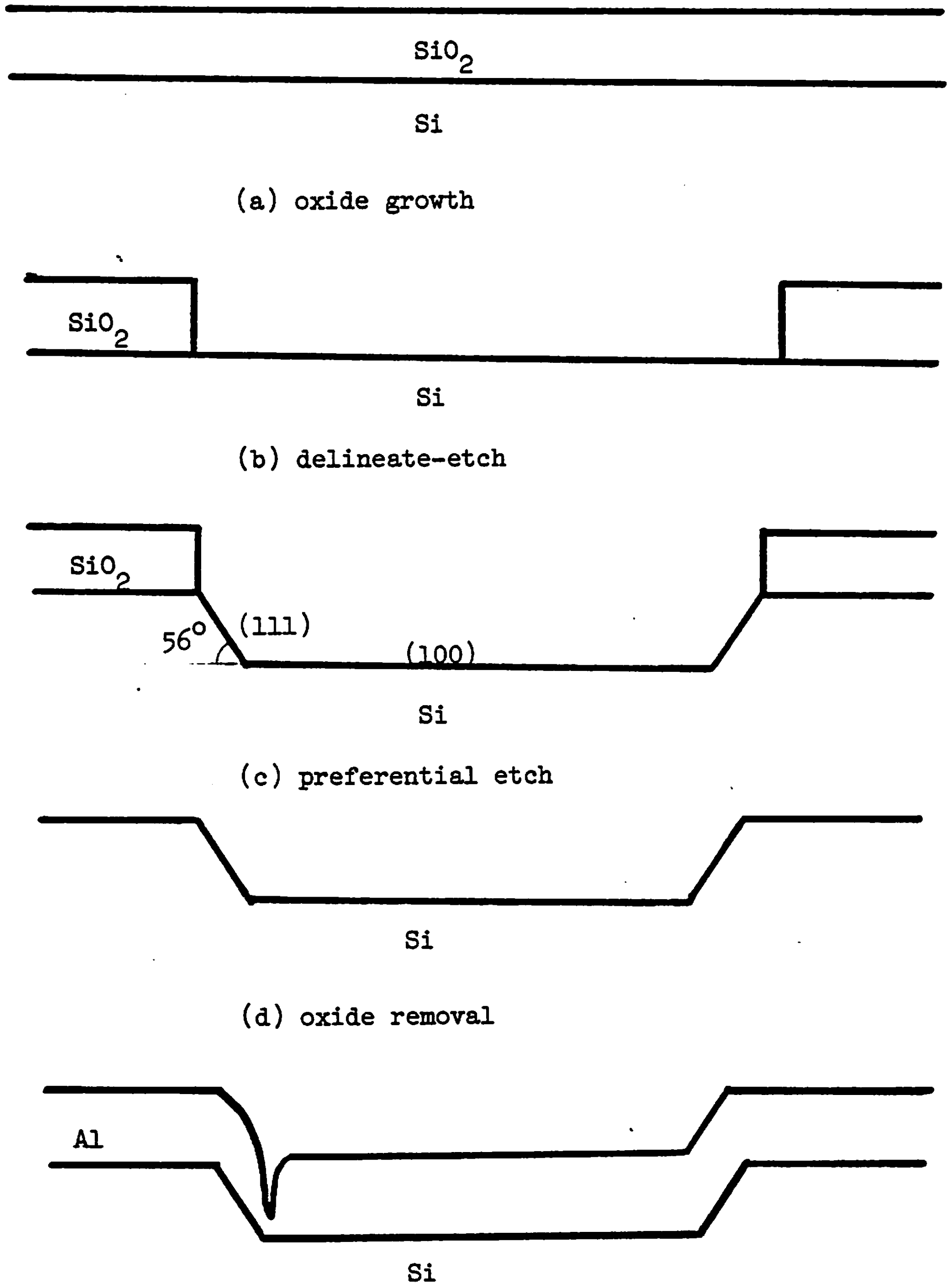


Fig. 15(c) Two layer metalization. The first layer was deposited as in (a), the second layer was deposited as in (b).



(a) oxide growth

(b) delineate-etch

(c) preferential etch

(d) oxide removal

(e) Al evaporation

Fig. 16. Processing Sequence

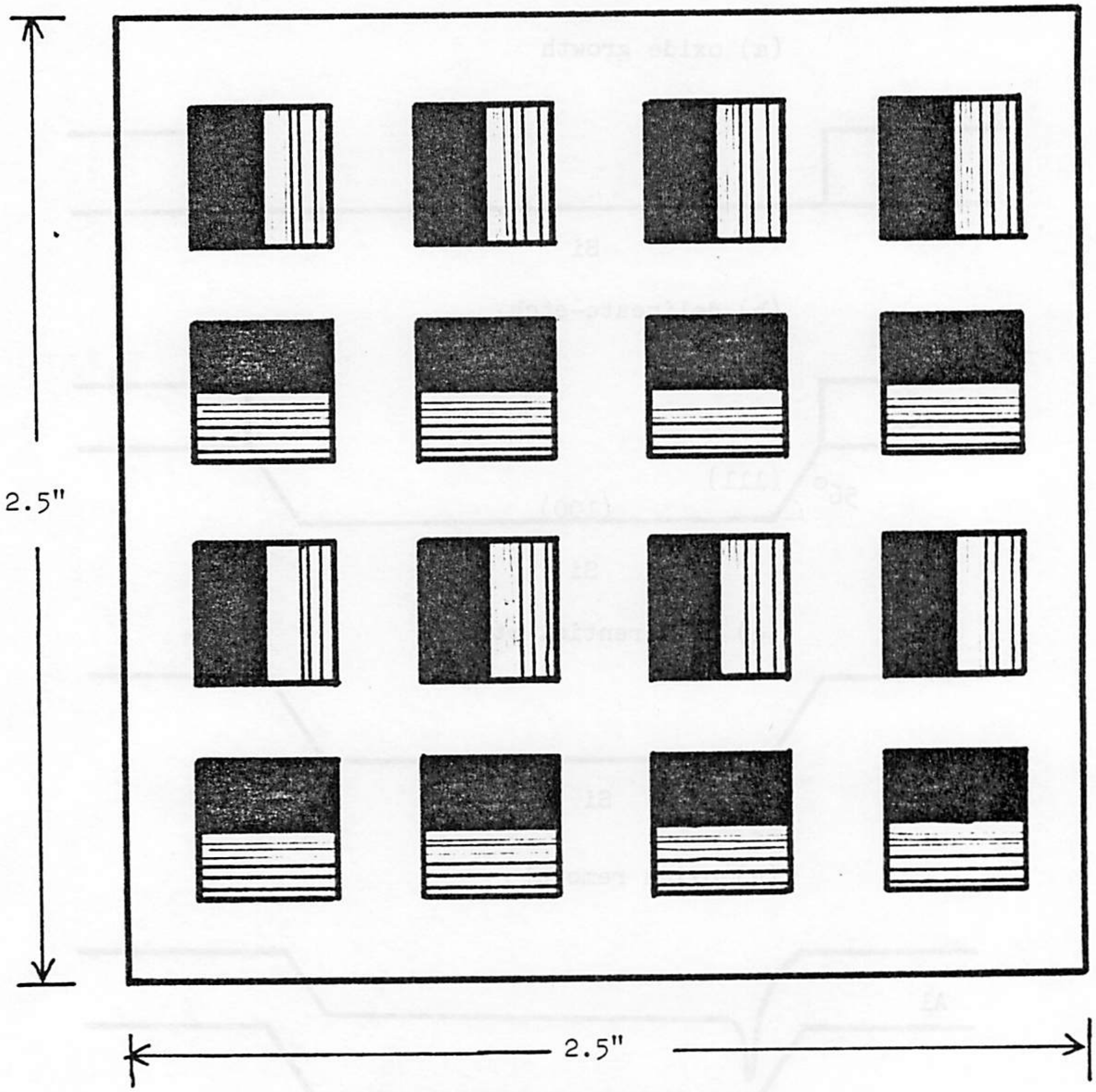


Fig. 17. Photolithographic masking pattern

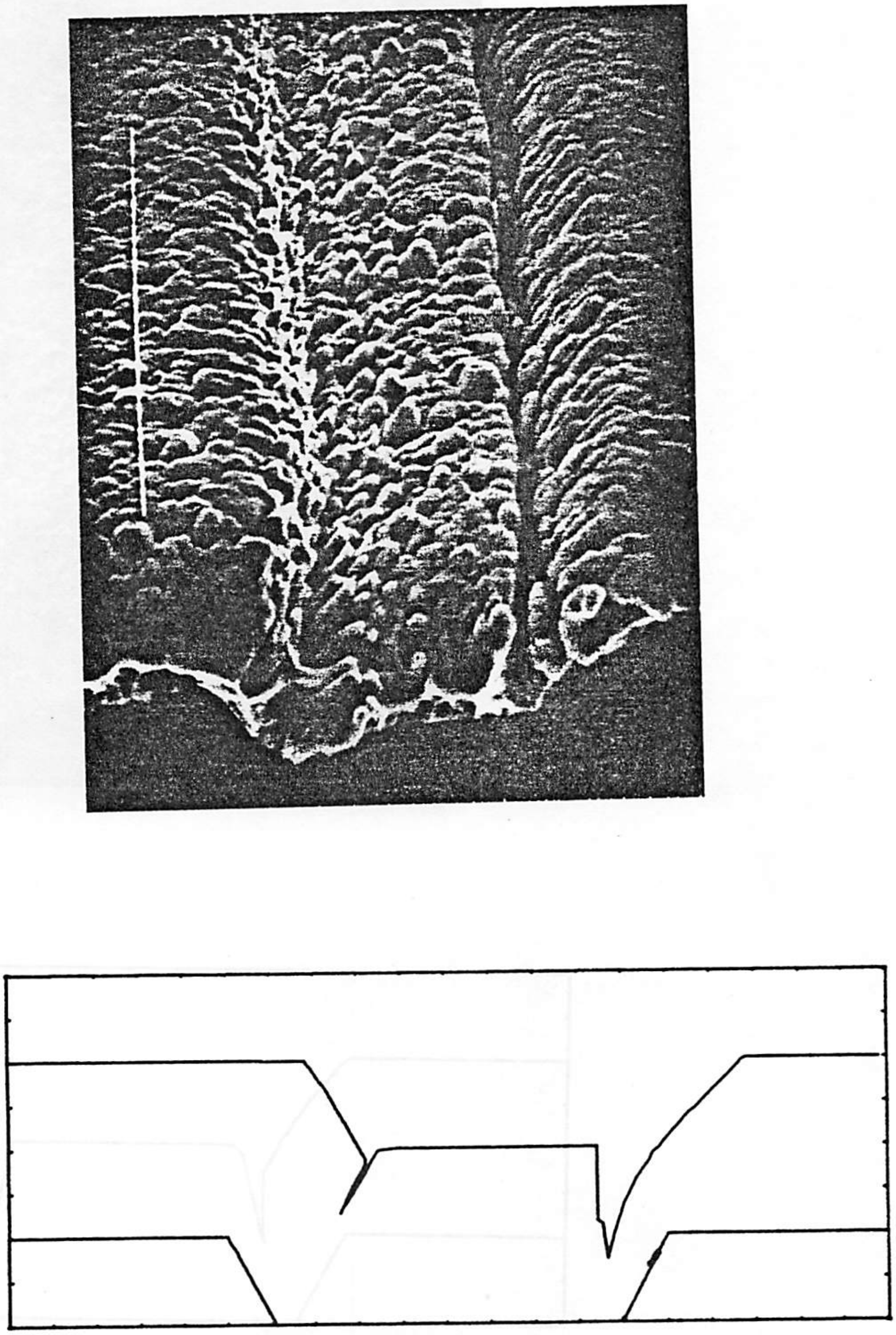


Fig. 18(a) SEM micrograph and computer simulation of asymmetrical aluminum step coverage

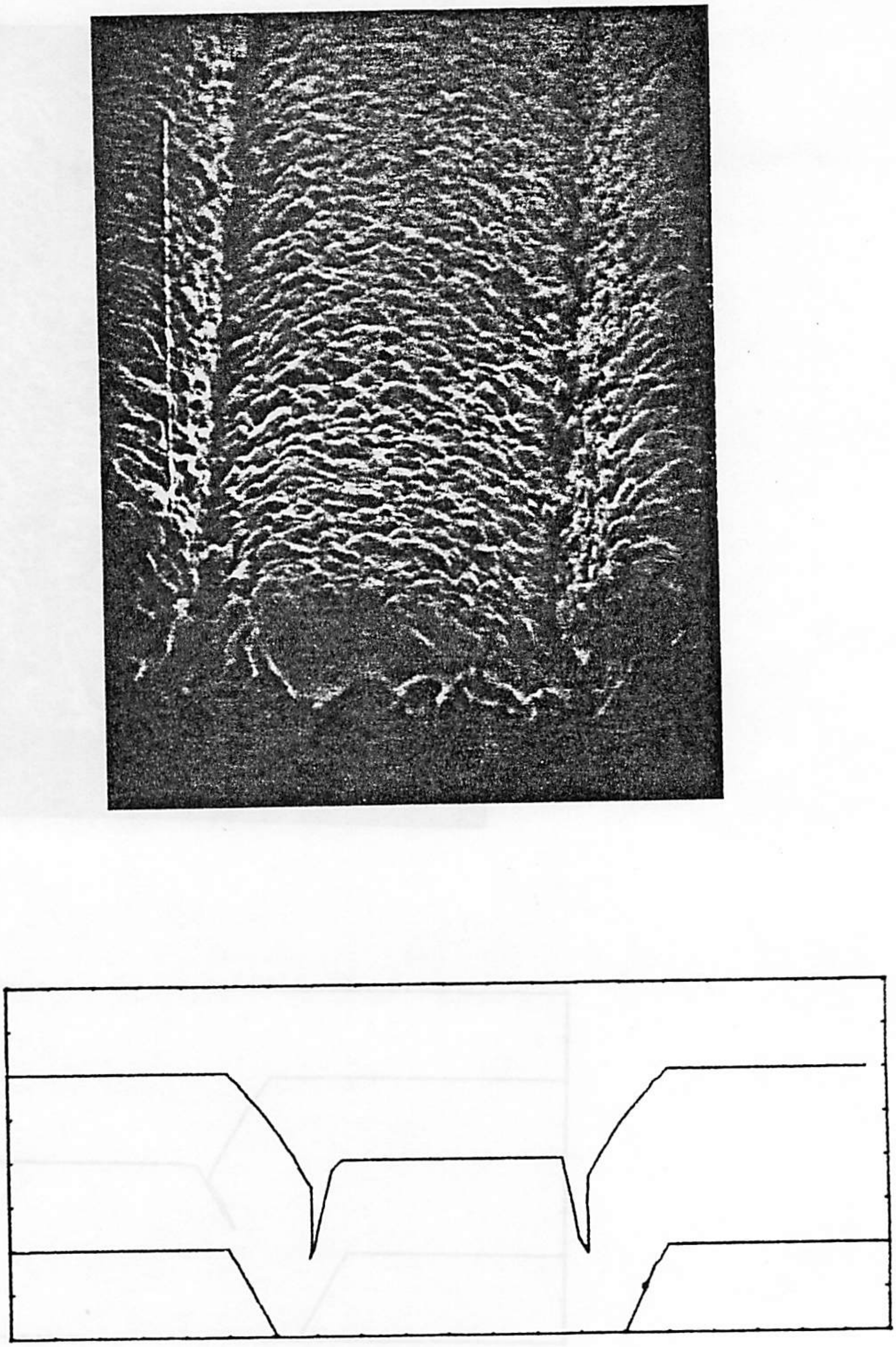


Fig. 18(b) SEM micrograph and computer simulation of symmetrical aluminum step coverage