



Evaporation Under Intense Energy Deposition

L.L. Loebel and W.G. Wolfer

August 1980

UWFDM-370

FUSION TECHNOLOGY INSTITUTE
UNIVERSITY OF WISCONSIN
MADISON WISCONSIN

Evaporation Under Intense Energy Deposition

L.L. Loebel and W.G. Wolfer

Fusion Technology Institute
University of Wisconsin
1500 Engineering Drive
Madison, WI 53706

<http://fti.neep.wisc.edu>

August 1980

UWFDM-370

Evaporation Under Intense
Energy Deposition

L. L. Loebel and W. G. Wolfer

Fusion Engineering Program
Nuclear Engineering Department
University of Wisconsin
Madison, WI 53706

August 1980

UWFD-370

1. Introduction

Intense heat fluxes on limiters, the plasma chamber wall, and on divertor plates, are encountered in magnetic fusion devices during arcing, when runaway electron beams strike the limiter, and during hard plasma disruptions. The latter are of particular concern in future fusion power reactors, where the thermal energy of the plasma reaches values of the order of 300 MJ. During a hard disruption this energy is deposited within a time estimated to be between 1 to 100 ms on limiters or a portion of the first wall. The resulting energy flux, averaged over the deposition time, may reach values between 10 to 1000 kW/cm². Melting and evaporation of first wall and limiter materials may then occur.

In inertial confinement fusion reactors, evaporation of the first wall may also take place as the x-ray radiation and the debris emanating from the ignited fusion pellet strike the first wall.

It is therefore important to evaluate the amount of material evaporated from an exposed wall or structure for two reasons. First, the evaporated atoms may contaminate the plasma in case of a magnetic fusion device, or induce laser-light breakdown and impair the focussing needed for subsequent pellet implosions. Second, repeated evaporation represents an important erosion mechanism of the first wall in addition to sputtering. This may further limit the ultimate lifetime of the first wall in a fusion reactor.

Evaporation under intense energy fluxes must be distinguished from slow evaporation as it takes place, for example, in high vacuum equipment. In this case, the energy expended in the evaporation process is negligible compared to either the thermal energy stored in the condensed phase, or to the heat conducted into the condensed material. In contrast, under intense energy fluxes,

the energy utilized in the evaporation process is substantial. Therefore, it is necessary to correctly partition the incident energy into the amounts expended for evaporation, for melting, for conduction into the material, for radiation, for heating the vapor, etc. The correct evaluation of the intense evaporation problem calls then, apart from the physical consideration about the kinetics of the evaporation process, for the solution of a moving boundary problem.

In the present paper we shall concentrate on this latter aspect, and not investigate the kinetics of the evaporation process itself. As a consequence, the results presented here represent only an approximation for the amount of material evaporated for a given energy flux and deposition time. Furthermore, as discussed in Section 2, where the slow evaporation process is reviewed, a continuous transition from slow to intense evaporation cannot be made with the present model. A unified treatment of evaporation under any heat flux is presently under investigation, and it will be reported in the near future.

In Section 3, the intense evaporation model of Andrews and Atthey is outlined, and results of its application to a selection of metals are given in Section 4.

2. The Slow Evaporation Process

For the slow evaporation of a solid, the net flux of atoms into the vapor phase is given by the Hertz-Knudsen-Langmuir equation [1]

$$J_v = \alpha_v (P_S - P_0) (2\pi M k T)^{-1/2} . \quad (1)$$

Here, P_0 is the existing partial pressure in the vapor container, P_S the saturation pressure of the vapor, M the mass of the vapor species, and α_v is the

evaporation coefficient. The latter is usually assumed to be close or equal to one.

Net evaporation occurs when $P_0 < P_S$, whereas net condensation takes place when $P_0 > P_S$. The first term in Eq. 1 is therefore the evaporation flux into a vacuum.

Behrisch [2] has employed the vacuum evaporation rate contained in Eq. 1 and computed the amount of material evaporated for a temperature excursion of the first wall following a plasma disruption. It was assumed that the entire incident energy is conducted into the first wall, giving rise to a transient in the surface temperature $T_S(t)$. The evaporation rate was integrated over the duration of the temperature transient utilizing the temperature dependence of the saturation vapor pressure $P_S(T_S(t))$.

This approach in computing the evaporation rate is legitimate only if the rate of energy expended in the evaporation is a small fraction of the incident energy. We can define the validity of this approach in more quantitative terms as follows.

Let the energy flux per unit area and unit time be $W(t)$. If L_V is the latent heat of vaporization (or sublimation when no melt layer forms) per unit mass, ρ the mass density, and Ω the atomic volume, then $L_V \rho \Omega J_V$ is the rate of energy utilized in evaporation. Accordingly, the approach of Behrisch is a good approximation of the evaporation rate if

$$L_V \rho \Omega J_V(t) \ll W(t) \quad . \quad (2)$$

In this case, the energy balance on the surface

$$W(t) = -K \frac{\partial T}{\partial x} \Big|_{\text{surface}} + L_v \rho \Omega J_v(t) \quad (3)$$

(K is the thermal conductivity) can be satisfied by neglecting the second term in Eq. 3. Heat conduction and evaporation can then be treated as uncoupled phenomena.

3. The Intense Evaporation Process

When the inequality (2) does not hold, the heat conduction equation

$$\frac{\partial^2 T}{\partial x^2} = \frac{1}{D} \frac{\partial T}{\partial t} \quad (4)$$

must be solved to satisfy the exact boundary condition of Eq. 3. In Eq. 4,

$$D = K/\rho c \quad (5)$$

is the thermal diffusivity and c the specific heat per unit mass.

By writing

$$\Omega J_v = \frac{ds}{dt} \quad (6)$$

where s(t) is the instantaneous position of the boundary, it becomes obvious that the condition

$$W(t) = -K \frac{\partial T}{\partial x} \Big|_{s(t)} + L_v \rho \frac{ds}{dt} \quad (7)$$

calls for the solution of a moving boundary problem.

Equation 3 or 7 represents an inhomogeneous boundary condition which is

linear in the temperature gradient, and extremely nonlinear in the surface temperature T_s through the dependence of the evaporation flux J_v on the saturation vapor pressure [3]

$$P_s(T) = P_0 \exp(-\Delta H/kT) \quad (8)$$

where ΔH is an activation energy.

To the authors' knowledge, a solution of this problem does not exist in the literature. Therefore, the following approximation is adopted in the present paper.

From Eqs. 1 and 6, we find that net evaporation commences when

$$P_s(T_v) = P_0 \quad . \quad (9)$$

For a given partial vapor pressure P_0 in the plasma chamber, Eq. 9 defines a surface temperature T_v where evaporation begins. We now assume that further heating of the wall does not occur, and that the surface temperature remains at the value of T_v . Note, however, that heat conduction still continues. The boundary condition (6) is therefore replaced by

$$T_s(t) = T_v \quad \text{for } t \geq t_p \quad (10)$$

where t_p is the preheat time required to raise the surface temperature from its initial value T_0 to the boiling or sublimation temperature T_v . Within this approximation, Eq. 10 replaces Eq. 6 and provides a second boundary condition in addition to Eq. 7.

This moving boundary problem was solved by Andrews and Atthey [4] based on a perturbation theory which will be briefly outlined in the following.

First, Andrews and Atthey observe that the latent heat of fusion, L_f , is only a few percent of the latent heat of evaporation. Furthermore the specific heat and the thermal diffusivity of the liquid metal are similar to the values of the solid metal. Therefore, one may in a first approximation neglect the energy expended in melting (or simply include it into L_v), and also treat both melt and solid as one substance with regard to heat conduction.

Second, the Stefan number

$$\epsilon = \bar{c} (T_v - T_0)/L_v, \quad (11)$$

where \bar{c} is an average value of the heat capacity over the temperature range T_0 to T_v , turns out to be a small number for most materials, being of the order of 0.2 or less. It is then possible to expand the temperature

$$T(x,t) = T^{(0)}(x,t) + \epsilon T^{(1)}(x,t) + \dots$$

into a perturbation series in ϵ as well as the boundary velocity ds/dt and the boundary position $s(t)$.

The moving boundary problem is then solved to first-order perturbation in ϵ for a constant energy flux W switched on at time $t = 0$. By introducing the dimensionless variables

$$\zeta = x/\ell \quad (12)$$

for the distance,

$$\xi = s/\ell \quad (13)$$

for the boundary position,

$$\tau = vt/\ell \quad (14)$$

for the time,

$$\theta = T/T_v \quad (15)$$

for the temperature, and

$$\eta = (ds/dt)/v \quad (16)$$

for the surface velocity, a general solution is found. Here, the characteristic length scale is given by

$$\ell = D/v \quad , \quad (17)$$

where

$$v = W/[L_v + \bar{c} (T_v - T_o)]\rho \quad (18)$$

is the asymptotic surface velocity reached for long energy deposition times.

The pre-heat time to reach the surface temperature T_v is in dimensionless units given by [4]

$$\tau_p = \frac{\pi}{4} \frac{\epsilon^2}{(1 + \epsilon)^2} \approx \frac{\pi}{4} \epsilon^2 \quad , \quad (19)$$

and the dimensionless surface velocity is

$$\eta \approx \{1 + \epsilon [\frac{1}{2} \operatorname{erfc} (\sqrt{\tau} / 2) - \exp(-\tau/4)/\sqrt{\pi\tau}]\}^* \quad (20)$$

$$\{\frac{2}{\pi} [1 + \epsilon/\sqrt{\pi\tau}]^{-1/2}\} \arcsin (\sqrt{1 - \pi\epsilon^2/4\tau}) \quad ,$$

valid for $\tau \geq \tau_p$. As noted by Andrews and Atthey, the solution of Eq. 20 has an error of the order of ϵ for times τ close to the preheat time τ_p , but the error reduces to the order of ϵ^2 when τ approaches and exceeds one. For a uniform energy flux W extending over a finite time interval τ , Eq. 20 can be integrated numerically to give the evaporated surface layer thickness ξ or $s = \xi v/D$.

As an example, Figs. 1 and 2 show the dimensionless surface velocity $\eta(\tau)$ and surface position $s(t)$ for values of $\epsilon = 0.167$ and as a function of the time.

4. Representative Results

In this section we present the results for the actual depth evaporated for a few selected metals listed in Table 1. In these calculations it was assumed that evaporation commences at temperatures T_v corresponding to vapor pressures of 10^{-3} , 10^{-1} , and 10 Torr.

Since the density, the specific heat, and the thermal diffusivity are functions of the temperature, average values were obtained from tabulated functions [5]; the averaging was performed over the temperature range given in Table 1.

The boiling temperature for a given pressure P_s was obtained from the equation

$$\log_{10} P_s [\text{Torr}] = A/T[\text{K}] + E$$

where A and E are constants [3] listed in the last two rows of Table 1.

Figures 3 to 8 show the depth of the material evaporated for a given energy flux assumed to be constant for a given deposition time. The straight line

labeled "0" represents the threshold for intense evaporation to occur. This threshold, as well as the lines for a constant depth of evaporated material depend to a minor degree on the pressure P_0 , i.e. on the surface temperature T_v that is reached during the intense evaporation process.

For values of energy fluxes and deposition times below the threshold for intense evaporation, some evaporation does occur. The amount of material removed below the threshold can be estimated according to the slow evaporation process discussed in Section 2.

With the present models for slow and intense evaporation, a discontinuous transition between the two evaporation regimes is suggested. In a unified model, however, a continuous transition between the two regimes is expected, and the threshold must then be defined in terms of a finite, but small, depth of evaporated material.

Acknowledgement

This research was supported in part by the INTOR project. Dr. D. L. Smith from Argonne National Laboratory provided much of the insight with regard to the distinction between slow and intense evaporation.

References

1. J. P. Hirth and G. M. Pound, "Condensation and Evaporation", Vol. 11 of Progr. Mat. Sciences, Pergamon, New York, 1963.
2. R. Behrisch, Nucl. Fusion, 12, (1972) 695.
3. R. E. Honig and D. A. Kramer, RCA Review, 30, (1969) 285.
4. J. G. Andrews and D. R. Atthey, J. Inst. Maths. Applics., 15, (1975) 59.
5. Thermophysical Properties of Matter (TPRC Data Series), Vol. I, IV, X; ed. by Y. S. Jouloukian, Plenum Data Comp., 1979.
6. Handbook of Chemistry and Physics, publ. by The Chemical Rubber Publ. Co., Cleveland, Ohio, 50th Ed., 1963.

Table 1. Materials Parameters

| Material | Al | Fe | Ni | Mo | Ti | W |
|--|-----------|-------------|------------|-------------|-------------|-------------|
| Temperature Range, K | 450 - 933 | 1300 - 2716 | 850 - 1726 | 1400 - 2883 | 1000 - 1953 | 1800 - 3653 |
| Average Specific Heat, (J/gK) | 1.115 | 0.704 | 0.578 | 0.432 | 0.725 | 1.756 |
| Average Density (g/cm ³) | 2.39 | 7.00 | 7.80 | 9.34 | 4.15 | 17.6 |
| Thermal Diffusivity (cm ² /s) | 0.820 | 0.080 | 0.153 | 0.248 | 0.078 | 0.286 |
| Latent Heat of Vaporization (J/g) | 11356 | 6268 | 6316 | 6184 | 8800 | 4490 |
| Constants for Vapor Pressure | | | | | | |
| A | -16764 | -20250 | -21175 | -32535 | -23162 | -43978 |
| E | 9.24 | 9.61 | 9.75 | 9.78 | 9.60 | 10.45 |

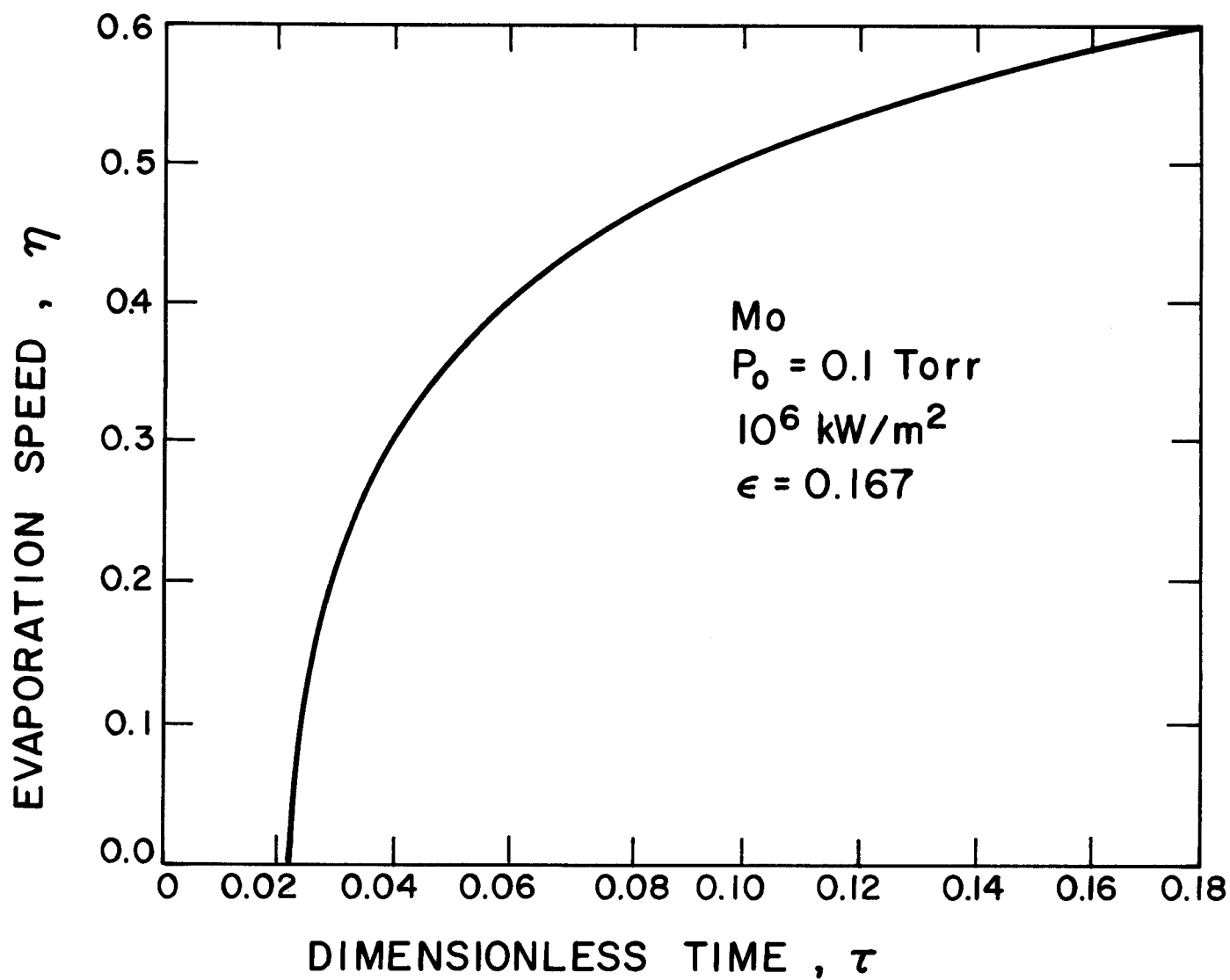


Figure 1. The dimensionless evaporation speed as a function of the dimensionless deposition time.

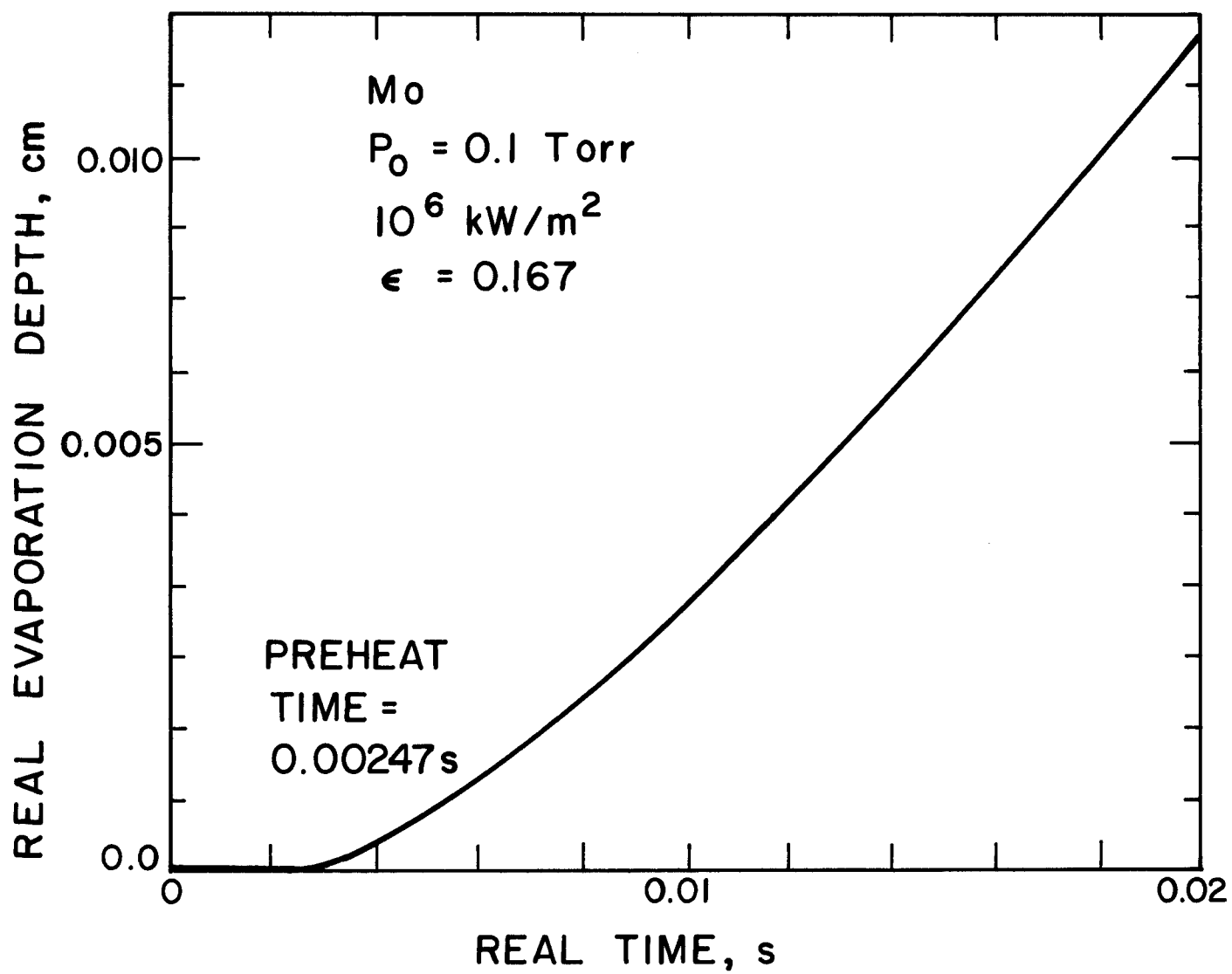


Figure 2. The depth of evaporated material as a function of time.

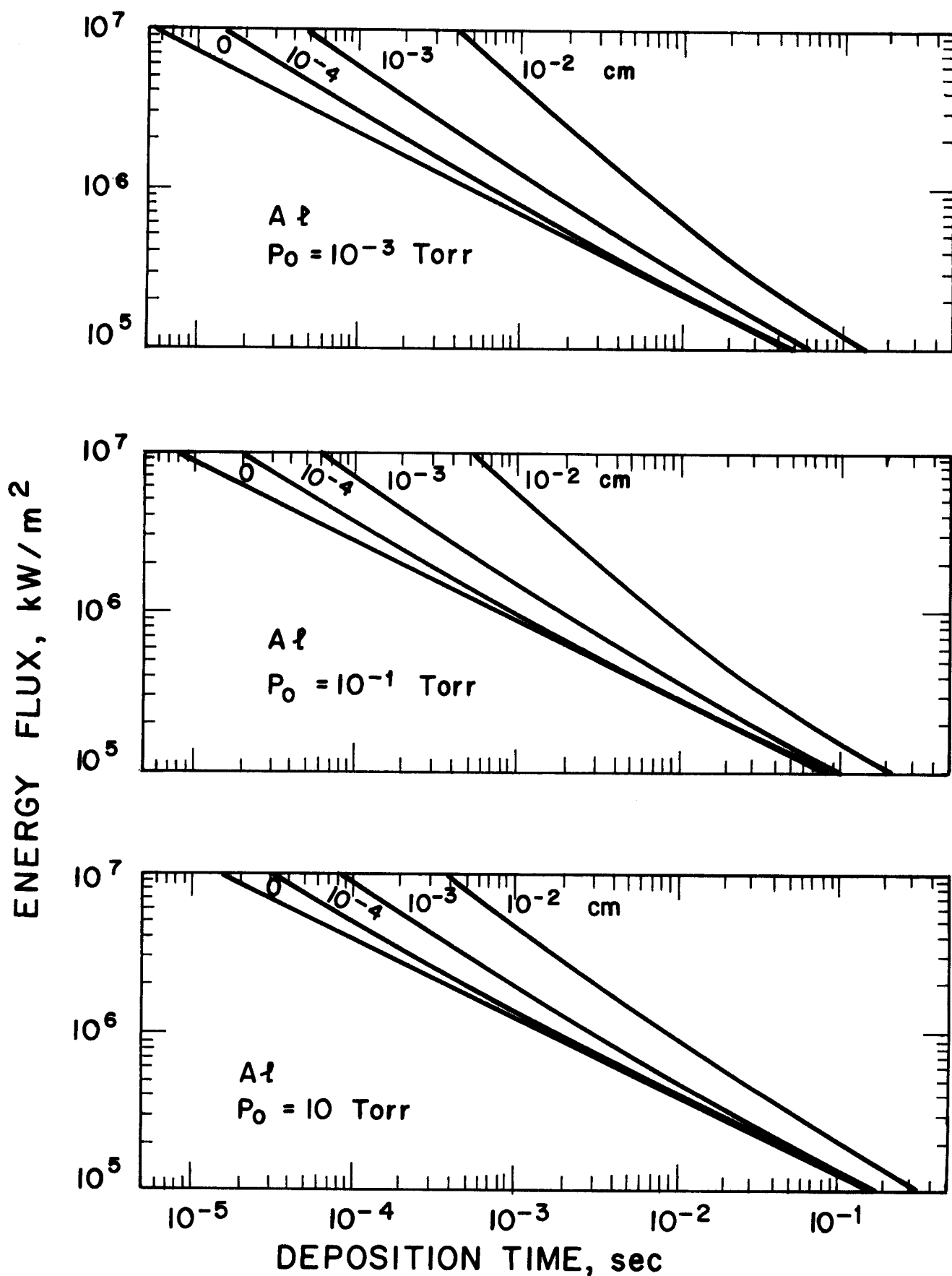
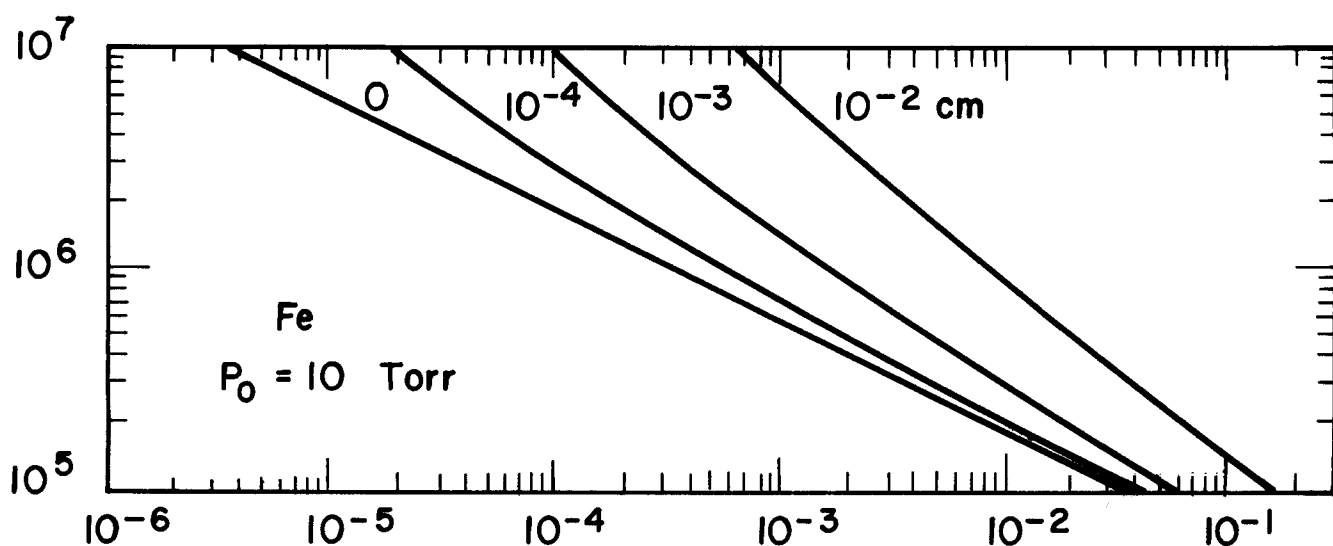
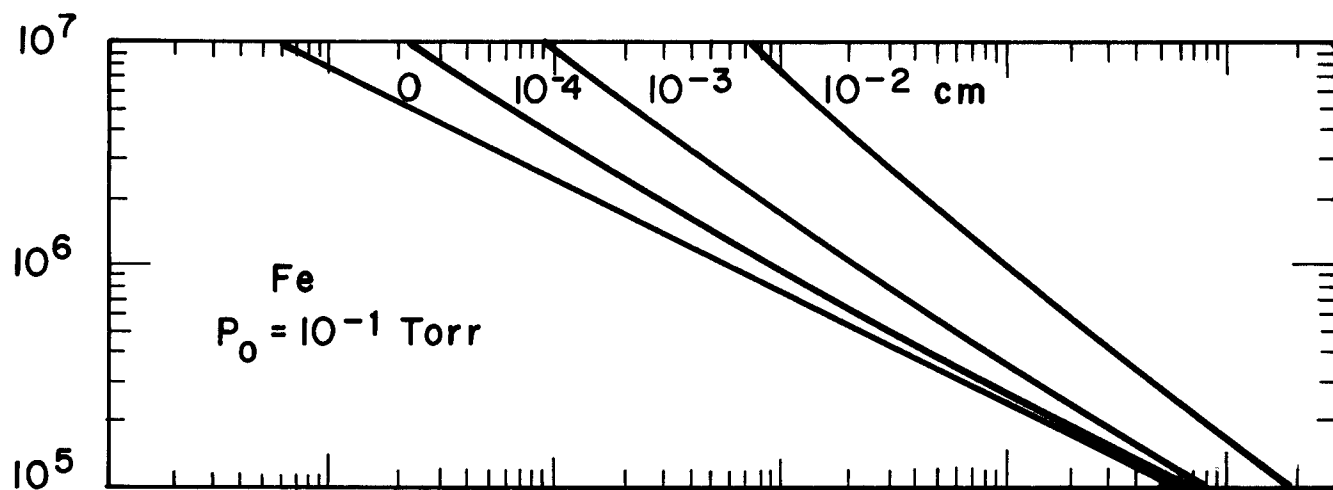
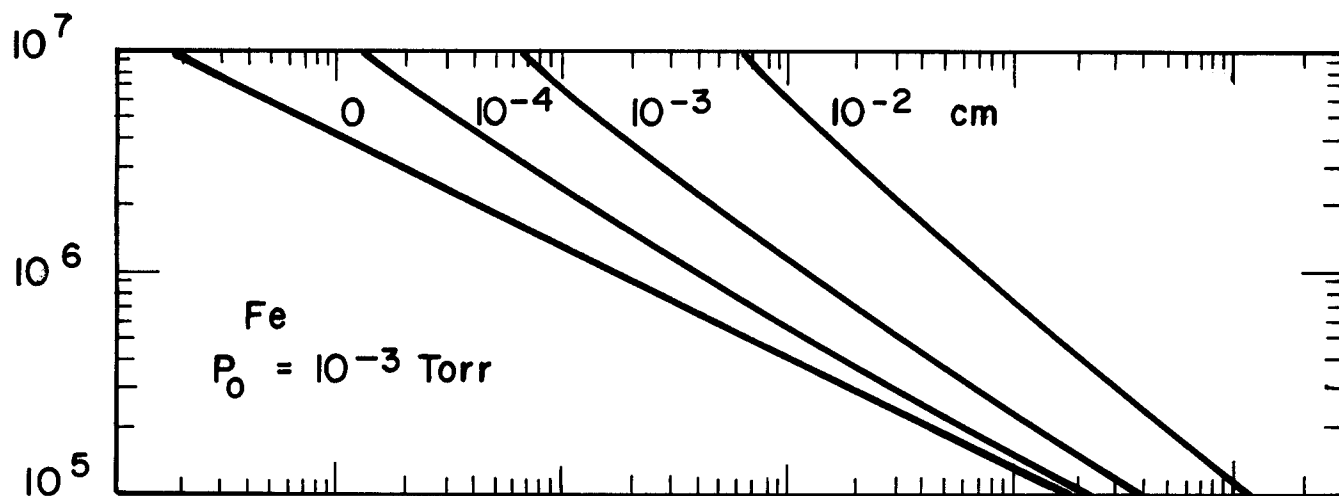


Figure 3. Depth of evaporation for aluminum.



DEPOSITION TIME IN SECONDS

Figure 4. Depth of evaporation for iron.

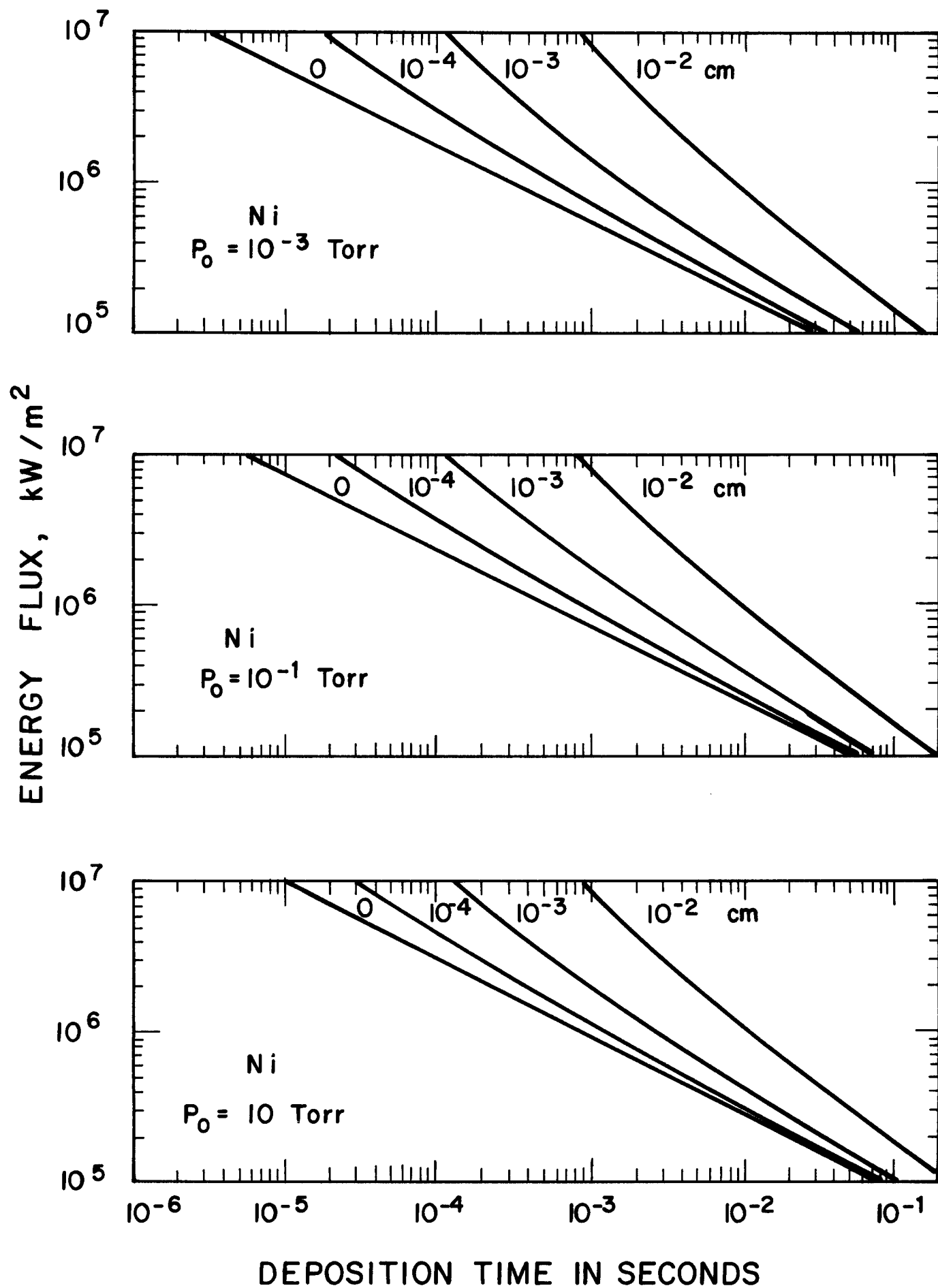


Figure 5. Depth of evaporation for nickel.

ENERGY FLUX, kW/m²

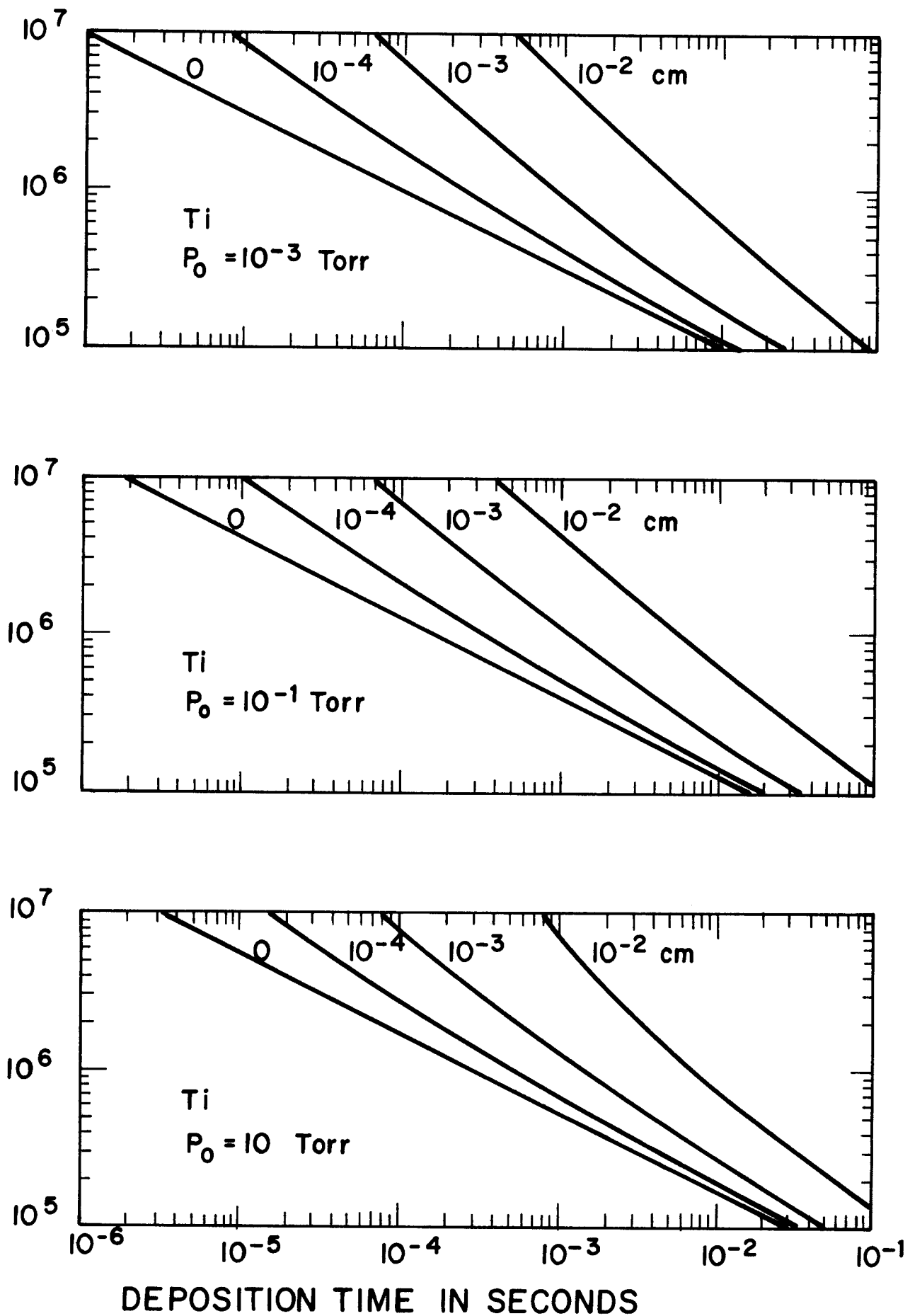


Figure 6. Depth of evaporation for titanium.

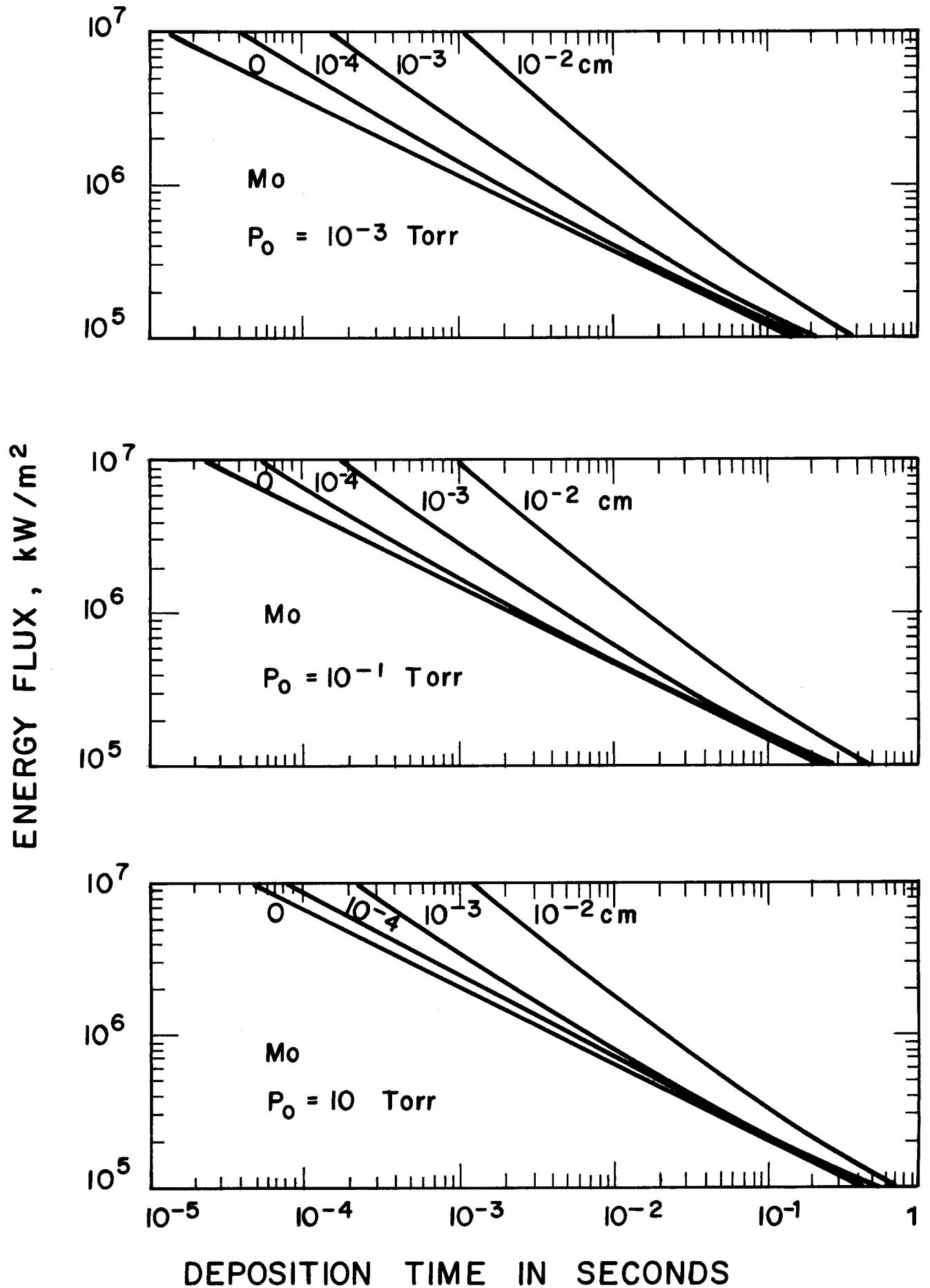


Figure 7. Depth of evaporation for molybdenum.

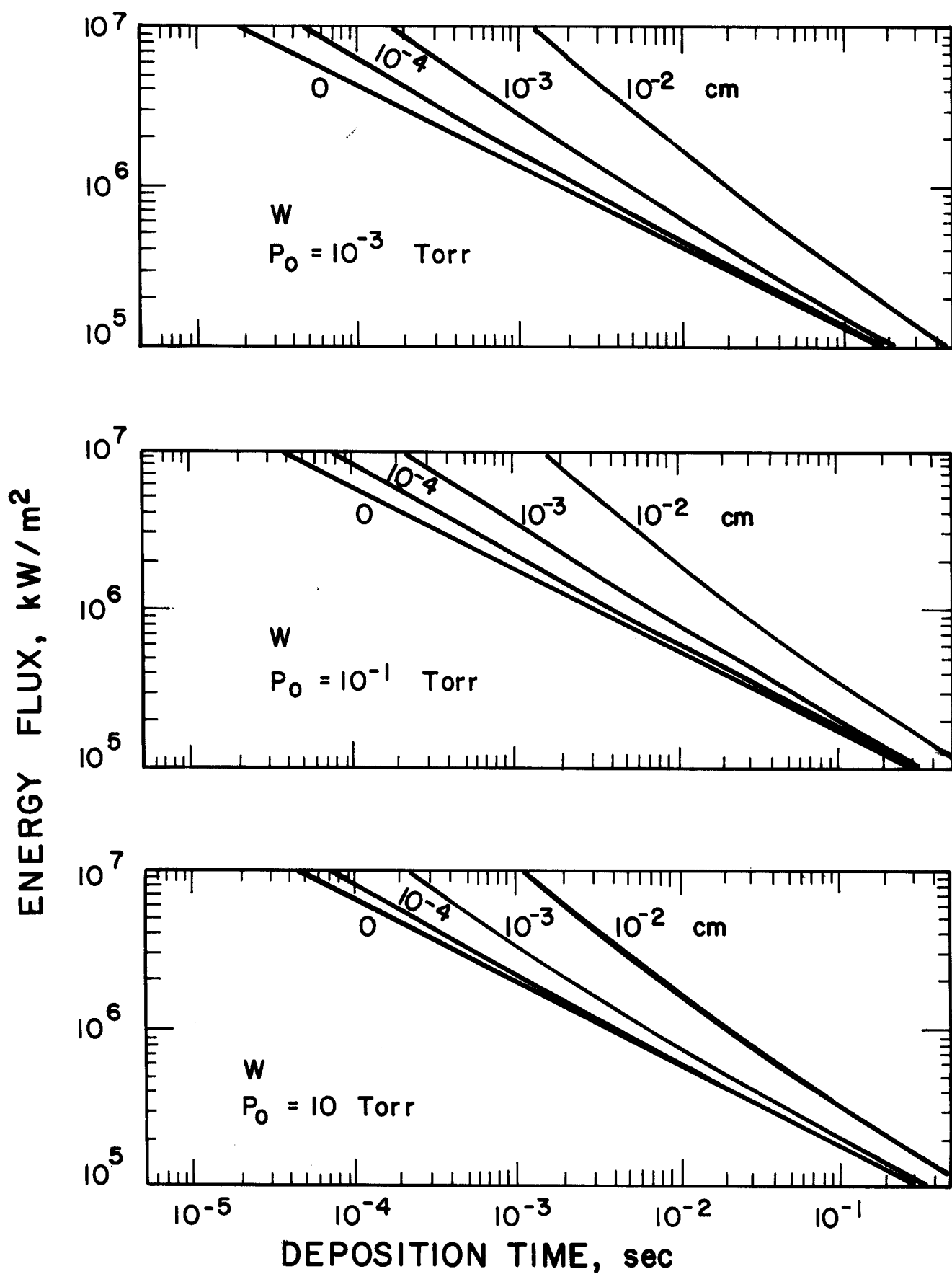


Figure 8. Depth of evaporation for tungsten.

Appendix

Listing of the Computer Program to Construct Figures for Intense Evaporation

```

10  REM :  THIS PROGRAM HAS BEEN WRITTEN USING THE HEWLETT-PACKARD 9845B.
20  REM :  THE OUTPUT OF THE PROGRAM IS A CONTOUR PLOT OF THE LOG(base 10)
30  REM :  OF FLUX vs. THE LOG (base 10) OF TIME, PER EVAPORATION DISTANCE
40  REM :  OF THE CONTAINMENT WALL.
50  !
60  !
70  REM :  SECTION I--DATA ENTRY:
80      INPUT "MATERIAL BEING STUDIED?",Material$
90      INPUT "INITIAL TEMPERATURE? (K)",T0
100     INPUT "PARTIAL PRESSURE OF THE PLASMA? (TORR)",P
110     INPUT "LATENT HEAT OF VAPORIZATION? (cal/gm)",Lvap
120     INPUT "TIME OF ENERGY DEPOSITION? (sec)",Time
130     INPUT "AVERAGE SPECIFIC HEAT? (cal/(gm*K))",C
140     INPUT "AVERAGE DENSITY? (gm/cm^3)",Dens
150     INPUT "AVERAGE THERMAL DIFFUSIVITY? (cm^2/sec)",Diff
160     INPUT "AVERAGE VALUE OF A?",A
170     INPUT "AVERAGE VALUE OF E?",E
180  REM :  SECTION II--DIMENSIONALIZING MATRICES AND INITIALIZING VALUES:
190      DIM Realdistance(2501),Realtime(2501)
200  REM :  The Tx matrices are the values of the time when the evaporation
210      !      distance becomes  $10^{(-x)}$  cm;
220      !      The Tx5 matrices are the values of the time when the
230      !      evaporation distance becomes  $5 \cdot 10^{(-x)}$  cm.
240      DIM Flux_matrix(19),Tpreheat(19),T55(19),T5(19),T45(19),T4(19),T35(19)
250      DIM T3(19),T25(19),T2(19),T1(19)
260  REM :  Initializing Flux matrix values:
270      Count=1
280      FOR I=5 TO 6
290          FOR J=1 TO 9
300              Flux_matrix(Count)=J*10^I
310              Count=Count+1
320          NEXT J
330      NEXT I
340      Flux_matrix(19)=1E7
350  REM :  Initializing Time and Distance matrix values:
360      FOR I=1 TO 2501
370          Realdistance(I)=Realdistance(I)=0
380      NEXT I
390      FOR I=1 TO 19
400          Tpreheat(I)=0
410          T55(I)=T5(I)=T45(I)=T4(I)=T35(I)=T3(I)=T25(I)=T2(I)=T1(I)=0
420      NEXT I
430  REM :  SECTION III--PRE-INTEGRAL CALCULATIONS:
440      FOR J=1 TO 19
450          F=Flux_matrix(J)
460          Count=1
470          Tvap=A/(LGT(P)-E)
480          Epsilon=C*(Tvap-T0)/Lvap
490          V=F*.1/(Dens*(Lvap*4.184)*(1+Epsilon))
500          Tau=Time*V^2/Diff
510          Taupre=PI*Epsilon^2/4
520          Tpreheat(J)=Taupre*Diff/V^2
530          IF Taupre>=Tau THEN GOTO 880
540  REM :  SECTION IV--CALCULATION OF THE INTEGRAL:
550  REM :  PART 1--FROM Tpreheat TO 5*Tpreheat, STEP Tpreheat/100
560  REM :  (The integral is calculated using the trapezoidal rule;
570  REM :  the first portion is divided into 500 pieces.)
580      Taunorm=Taupre
590      GOSUB Compute_eta
600      Old=Eta
610      Sum=0
620      Realdistance(Count)=0
630      Realtime(Count)=Tpreheat(J)

```



```

640      Deltatau=Taupre/100
650      Count=Count+1
660      FOR Taunorm=Taupre+Deltatau TO 5*Taupre STEP Deltatau
670          GOSUB Compute_eta
680          Sum=Sum+Deltatau/2*(Old+Eta)
690          Realdistance(Count)=Diff*Sum/V
700          Realtime(Count)=Taunorm*Diff/V^2
710          Old=Eta
720          Count=Count+1
730      NEXT Taunorm
740  REM : PART 2--The rest of the interval is divided into 2000 pieces
750      Count=501
760      Deltatau=(Tau-5*Taupre)/2000
770      FOR Taunorm=5*Taupre TO Tau STEP Deltatau
780          GOSUB Compute_eta
790          Sum=Sum+Deltatau/2*(Old+Eta)
800          Realdistance(Count)=Diff*Sum/V
810          Realtime(Count)=Taunorm*Diff/V^2
820          Old=Eta
830          Count=Count+1
840      NEXT Taunorm
850      IF Realdistance(2501)=0 THEN GOSUB Last_value
860  REM : PART 3--Pick up the values of Time at the significant distances:
870      GOSUB Time_datapoints
880      NEXT J
890  REM : SECTION V--PRINT OUT AND GRAPH DATA VALUES:
900      CALL Print_all(Flux_matrix(*),Tpreheat(*),T5(*),T55(*),T4(*),T45(*),T3(
*),T35(*),T2(*),T25(*),T1(*))
910      CALL Graph_data(Flux_matrix(*),Tpreheat(*),T4(*),T3(*),T2(*))
920      CALL Print_out(Flux_matrix(*),Tpreheat(*),T4(*),T3(*),T2(*))
930      !
940      !
950  REM : END OF MAIN PROGRAM--SUBROUTINES FOLLOW
960      STOP
970      !
980      !
990      !
1000     !
1010     !
1020     !
1030  REM : PICK UP DESIRED TIME VALUES:
1040  Time_datapoints:
1050      Count=1
1060      IF Realdistance(Count)>=1E-5 THEN GOTO 1100
1070      Count=Count+1
1080      IF Count<Dimension THEN GOTO 1060
1090      GOTO Return
1100      T5(J)=Realtime(Count)
1110      IF Realdistance(Count)>=5E-5 THEN GOTO 1150
1120      Count=Count+1
1130      IF Count<Dimension THEN GOTO 1110
1140      GOTO Return
1150      T55(J)=Realtime(Count)
1160      IF Realdistance(Count)>=1E-4 THEN GOTO 1200
1170      Count=Count+1
1180      IF Count<Dimension THEN GOTO 1160
1190      GOTO Return
1200      T4(J)=Realtime(Count)
1210      IF Realdistance(Count)>=5E-4 THEN GOTO 1250
1220      Count=Count+1
1230      IF Count<Dimension THEN GOTO 1210
1240      GOTO Return
1250      T45(J)=Realtime(Count)
1260      IF Realdistance(Count)>=1E-3 THEN GOTO 1300

```

```

1270      Count=Count+1
1280      IF Count<Dimension THEN GOTO 1260
1290      GOTO Return
1300      T3(J)=Realtime(Count)
1310      IF Realdistance(Count)>=5E-3 THEN GOTO 1350
1320      Count=Count+1
1330      IF Count<Dimension THEN GOTO 1310
1340      GOTO Return
1350      T35(J)=Realtime(Count)
1360      IF Realdistance(Count)>=1E-2 THEN GOTO 1400
1370      Count=Count+1
1380      IF Count<Dimension THEN GOTO 1360
1390      GOTO Return
1400      T2(J)=Realtime(Count)
1410      IF Realdistance(Count)>=5E-2 THEN GOTO 1450
1420      Count=Count+1
1430      IF Count<Dimension THEN GOTO 1410
1440      GOTO Return
1450      T25(J)=Realtime(Count)
1460      IF Realdistance(Count)>=.1 THEN GOTO 1500
1470      Count=Count+1
1480      IF Count<Dimension THEN GOTO 1460
1490      GOTO Return
1500      T1(J)=Realtime(Count)
1510 Return:   RETURN
1520      !
1530      !
1540      !
1550      !
1560 REM :   COMPUTE THE LAST VALUE OF INTEGRAL:
1570 REM :   (Necessary due to computer accuracy limitations.)
1580 Last_value:   !
1590      Taunorm=Tau
1600      GOSUB Compute_eta
1610      Sum=Sum+Deltatau/2*(Old+Eta)
1620      Realdistance(2501)=Diff*Sum/Y
1630      Realtime(2501)=Time
1640      RETURN
1650      !
1660      !
1670      !
1680      !
1690 REM :   COMPUTE VALUE OF ETA:
1700 Compute_eta:   !
1710      W=EXP(-Taunorm/4)
1720      X=SQR(Taunorm*PI)
1730      Y=SQR(1-PI*Epsilon^2/(4*Taunorm))
1740      Z=SQR(Taunorm)/2
1750      CALL Comp_errorfcn(Z,Erfcz)
1760      Eta=2/PI*ASN(Y)*(X/(X+Epsilon))*(1+Epsilon*(Erfcz/2-W))
1770      RETURN
1780      !
1790      !
1800      !
1810      !
1820 REM :   COMPUTE THE NECESSARY ERROR FUNCTION (USED IN COMPUTING ETA):
1830 SUB Comp_errorfcn(X,Erfcx)
1840      OPTION BASE 1
1850      DIM A(5)
1860      P=.3275911
1870      T=1/(1+P*X)
1880      A(1)=.254829592
1890      A(2)=-.284496736
1900      A(3)=1.421413741
1910      A(4)=-1.453152027

```

```

1920      A(5)=1.061405429
1930      Erfcx=0
1940      FOR I=1 TO 5
1950          Erfcx=Erfcx+A(I)*T^I
1960      NEXT I
1970      Erfcx=Erfcx*EXP(-X^2)
1980      SUBEXIT
1990      !
2000      !
2010      !
2020      !
2030      REM : GRAPH THE CONTOUR LOG PLOTS OF FLUX VS. TIME FOR EACH DISTANCE:
2040      SUB Graph_data(F(*),Tpre(*),T4(*),T3(*),T2(*))
2050      PLOTTER IS "GRAPHICS"
2060      GRAPHICS
2070      REM : PART 1--Take the logs (where possible):
2080          FOR I=1 TO 19
2090              F(I)=LGT(F(I))
2100              Tpre(I)=LGT(Tpre(I))
2110              IF T4(I)>0 THEN T4(I)=LGT(T4(I))
2120              IF T3(I)>0 THEN T3(I)=LGT(T3(I))
2130              IF T2(I)>0 THEN T2(I)=LGT(T2(I))
2140          NEXT I
2150      REM : PART 2--Find maximum and minimum values (used for scaling graph):
2160          Tmin=10
2170          Tmax=-10
2180          FOR I=1 TO 19
2190              IF (Tpre(I)<>0) AND (Tpre(I)<Tmin) THEN Tmin=Tpre(I)
2200              IF Tpre(I)>Tmax THEN Tmax=Tpre(I)
2210              IF (T4(I)<>0) AND (T4(I)<Tmin) THEN Tmin=T4(I)
2220              IF T4(I)>Tmax THEN Tmax=T4(I)
2230              IF (T3(I)<>0) AND (T3(I)<Tmin) THEN Tmin=T3(I)
2240              IF T3(I)>Tmax THEN Tmax=T3(I)
2250              IF (T2(I)<>0) AND (T2(I)<Tmin) THEN Tmin=T2(I)
2260              IF T2(I)>Tmax THEN Tmax=T2(I)
2270          NEXT I
2280          Tmin=INT(Tmin)
2290          Tmax=INT(Tmax)
2300      REM : PART 3--Draw and label axes:
2310          FRAME
2320          SCALE 3,7.5,Tmin-1,Tmax+1
2330          CLIP 5,7,Tmin,Tmax
2340          AXES 1,1,5,Tmin
2350          UNCLIP
2360          CSIZE 2.5
2370          FOR X=5 TO 7
2380              MOVE X,Tmin-1/4
2390              LABEL X
2400          NEXT X
2410          FOR Y=Tmin TO Tmax
2420              MOVE 4.75,Y
2430              LABEL Y
2440          NEXT Y
2450          CSIZE 3.5
2460          MOVE 5.5,Tmin-1/2
2470          LABEL "LOG (base 10) of FLUX"
2480          RAD
2490          LDIR PI/2
2500          MOVE 4.50,(Tmin-Tmax)/2-1
2510          LABEL "LOG (base 10) of TIME"
2520          LDIR 0
2530      REM : GRAPH CONTOUR PLOTS WITH VARYING LINE TYPES:
2540          MOVE F(1),Tpre(1)
2550          LINE TYPE 9
2560          FOR I=1 TO 19

```

```

2570     IF Tpre(I)<>0 THEN PLOT F(I),Tpre(I)
2580 NEXT I
2590 !
2600 MOVE F(1),T4(1)
2610 LINE TYPE 4
2620 FOR I=1 TO 19
2630     IF T4(I)<>0 THEN PLOT F(I),T4(I)
2640 NEXT I
2650 !
2660 MOVE F(1),T3(1)
2670 LINE TYPE 3
2680 FOR I=1 TO 19
2690     IF T3(I)<>0 THEN PLOT F(I),T3(I)
2700 NEXT I
2710 !
2720 MOVE F(1),T2(1)
2730 LINE TYPE 1
2740 FOR I=1 TO 19
2750     IF T2(I)<>0 THEN PLOT F(I),T2(I)
2760 NEXT I
2770 !
2780 CSIZE 2.5
2790 LDIR 0
2800 CLIP 3,4.25,Tmin-1,Tmin+3
2810 FRAME
2820 MOVE 3.25,Tmin+2.5
2830 REM : CREATE LEGEND:
2840 LABEL "LEGEND:"
2850 CSIZE 2.25
2860 MOVE 3,Tmin+1.6
2870 LINE TYPE 9
2880 DRAW 3.25,Tmin+1.6
2890 DRAW 3.5,Tmin+1.6
2900 DRAW 3.75,Tmin+1.6
2910 DRAW 4,Tmin+1.6
2920 DRAW 4.25,Tmin+1.6
2930 MOVE 3.25,Tmin+1.7
2940 LINE TYPE 1
2950 LABEL "Preheat time,X = 0"
2960 MOVE 3,Tmin+.9
2970 LINE TYPE 4
2980 DRAW 4.25,Tmin+.9
2990 MOVE 3.25,Tmin+1
3000 LINE TYPE 1
3010 LABEL "X = 1E-4"
3020 MOVE 3,Tmin+.2
3030 LINE TYPE 3
3040 DRAW 4.25,Tmin+.2
3050 MOVE 3.25,Tmin+.3
3060 LINE TYPE 1
3070 LABEL "X = 1E-3"
3080 MOVE 3,Tmin-.5
3090 DRAW 4.25,Tmin-.5
3100 MOVE 3.25,Tmin-.4
3110 LABEL "X = 1E-2"
3120 !
3130 DUMP GRAPHICS
3140 EXIT GRAPHICS
3150 SUBEXIT
3160 !
3170 !

```

```

3180 !
3190 !
3200 REM : PRINT THE NUMERICAL VALUES ON THE GRAPH:
3210 SUB Print_out(F(*),Tpre(*),T4(*),T3(*),T2(*))
3220     FIXED 5
3230     PRINT
3240     PRINT
3250     PRINT
3260     PRINT "DATA VALUES ON GRAPH:"
3270     PRINT " "
3280     PRINT
3290     PRINT "LOG(Flux)          LOG(Tpre)      LOG(T4)      LOG(T3)      LOG(T
2) "
3300     FOR I=1 TO 19
3310         PRINT F(I);TAB(22);Tpre(I);TAB(34);T4(I);TAB(46);T3(I);TAB(58);T2
(I)
3320     NEXT I
3330     PRINT
3340 SUBEXIT
3350 !
3360 !
3370 !
3380 !
3390 REM : PRINT THE COMPLETE NUMERICAL VALUES BEFORE GRAPHING:
3400 SUB Print_all(F(*),Tpre(*),T5(*),T55(*),T4(*),T45(*),T3(*),T35(*),T2(*),T2
5(*),T1(*))
3410     FIXED 7
3420     PRINTER IS 0
3430     PRINT
3440     PRINT
3450     PRINT "COMPLETE DATA VALUES:"
3460     PRINT " "
3470     PRINT
3480     PRINT "          FLUX          PREHEAT TIME      X=1E-5      X=5E-5      X=1E
-4      X=5E-4 "
3490     FOR I=1 TO 19
3500         PRINT F(I);TAB(21);Tpre(I);TAB(33);T5(I);TAB(46);T55(I);TAB(58);T
4(I);TAB(70);T45(I)
3510     NEXT I
3520     PRINT
3530     PRINT "*****"
3540     PRINT
3550     PRINT "          FLUX          X=1E-3      X=5E-3      X=1E-2      X=5E
-2      X=1E-1 "
3560     FOR I=1 TO 19
3570         PRINT F(I);TAB(21);T3(I);TAB(33);T35(I);TAB(46);T2(I);TAB(58);T25
(I);TAB(70);T1(I)
3580     NEXT I
3590     PRINT
3600 SUBEXIT

```