Argonne
NATIONAL LABORATORY

ANL-06/10

On a Thermal Analysis of a Second Stripper for Rare Isotope Accelerator

---

Nuclear Engineering and Physics Divisions

About Argonne National Laboratory
Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at 9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne, see www.anl.gov.

Availability of This Report
This report is available, at no cost, at http://www.osti.gov/bridge. It is also available on paper to the U.S. Department of Energy and its contractors, for a processing fee, from:

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
phone (865) 576-8401
fax (865) 576-5728
reports@adonis.osti.gov

Disclaimer
This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor UChicago Argonne, LLC, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, Argonne National Laboratory, or UChicago Argonne, LLC.

ANL-06/10

On a Thermal Analysis of a Second Stripper for Rare Isotope Accelerator

---

by
Yoichi Momozaki, Nuclear Engineering Division and Jerry Nolen, Physics Division
Argonne National Laboratory

July 2008

Table of contents

ANL-06/10

On a Thermal Analysis of a Second Stripper for RIA

Yoichi Momozaki and Jerry Nolen

Abstract

This memo summarizes simple calculations and results of the thermal analysis on the second stripper to be used in the driver linac of Rare Isotope Accelerator (RIA). Both liquid (Sodium) and solid (Titanium and Vanadium) stripper concepts were considered. These calculations were intended to provide basic information to evaluate the feasibility of liquid (thick film) and solid (rotating wheel) second strippers. Nuclear physics calculations to estimate the volumetric heat generation in the stripper material were performed by "*LISE for Excel*". In the thermal calculations, the strippers were modeled as a thin 2D plate with uniform heat generation within the beam spot. Then, temperature distributions were computed by assuming that the heat spreads conductively in the plate in radial direction without radiative heat losses to surroundings.

Model

A schematic of the second stripper is shown in Figure 1. The stripper was modeled as a moving slab at velocity *v*, which could be a liquid or a solid. An incident U beam at the charge state of 72+, flux of 4 particle μA, and energy of 85 MeV/nucleon passes through the stripper material and deposits some energy as heat within the stripper. Expected beam diameter was 1 mm.

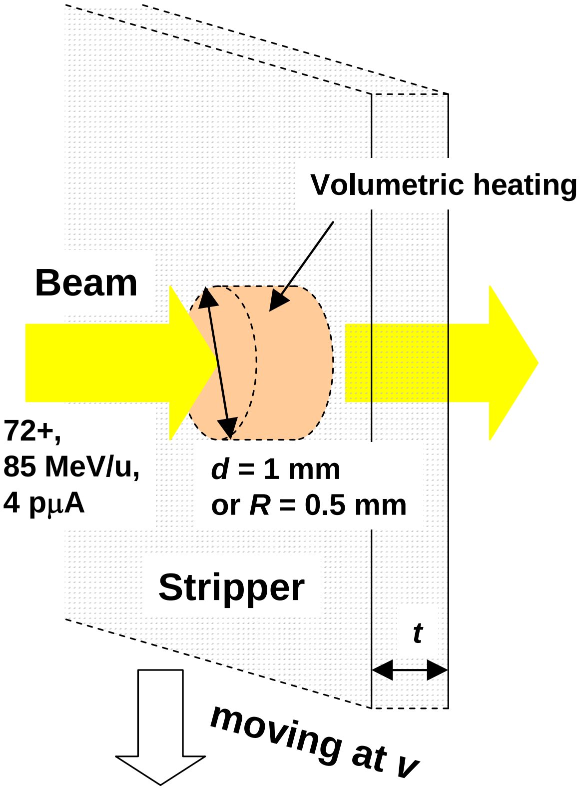

Figure 1. Schematic of Second Stripper.

Choice of Stripper Material

To experimentally determine the best stripper material at around 80 MeV/u, the series of experiments to determine the charge state distribution as a function of material variation and thickness was performed at National Superconducting Cyclotron Laboratory (NSCL) in Michigan State University (MSU). To simulate a 80 MeV/u, U beam, Bi beam at 80 MeV/u and the charge state of 63+ was used in the experiments. Various stripper materials with different dimensions including Be 13.2 mg/cm^2, C 16.8 mg/cm^2, Al 10.8 mg/cm^2, Al 15.5 mg/cm^2, V 14.0 mg/cm^2, Cu 8.4 mg/cm^2, and Cu 17.9 mg/cm^2 were tested. Results are summarized in Table 1, Figure 2, and Figure 3.

Table 1. Summary of Charge Distribution.

Q	Be 13.2 mg/cm^2 fractions	C 16.8 mg/cm^2 fractions	Al 10.8 mg/cm^2 fractions	Al 15.5 mg/cm^2 fractions	V 14.0 mg/cm^2 fractions	Cu 8.4 mg/cm^2 fractions	Cu 17.9 mg/cm^2 fractions
83		0.000166432	0.000138825	0.000291122	4.07711E-05	3.55096E-05	4.19519E-05
82	0.000781956	0.016127091	0.011693697	0.012104976	0.002219475	0.001717779	0.001384412
81	0.050323568	0.390985875	0.259024419	0.255559266	0.078203996	0.059727247	0.051983045
80	0.193424653	0.397093588	0.37251853	0.3709073	0.231629331	0.200163425	0.185592045
79	0.311737494	0.153659036	0.24092467	0.241536573	0.306225504	0.294056822	0.289109949
78	0.272114096	0.036574167	0.089083214	0.092108926	0.2281583	0.252602564	0.259436984
77	0.13419233	0.004958026	0.022542115	0.022964242	0.109213508	0.132073138	0.146067652
76	0.037425903	0.000435785	0.003647416	0.004080031	0.035111513	0.045862265	0.054109523
75			0.000420334	0.000447565	0.00792882	0.011896409	0.012274438
74			6.77958E-06		0.001268782	0.001864843	
centroid	78.64364119	80.18031477	79.77937883	79.76743748	78.80487979	78.61366781	78.53088057
Atomic number	4	6	13	13	23	29	29

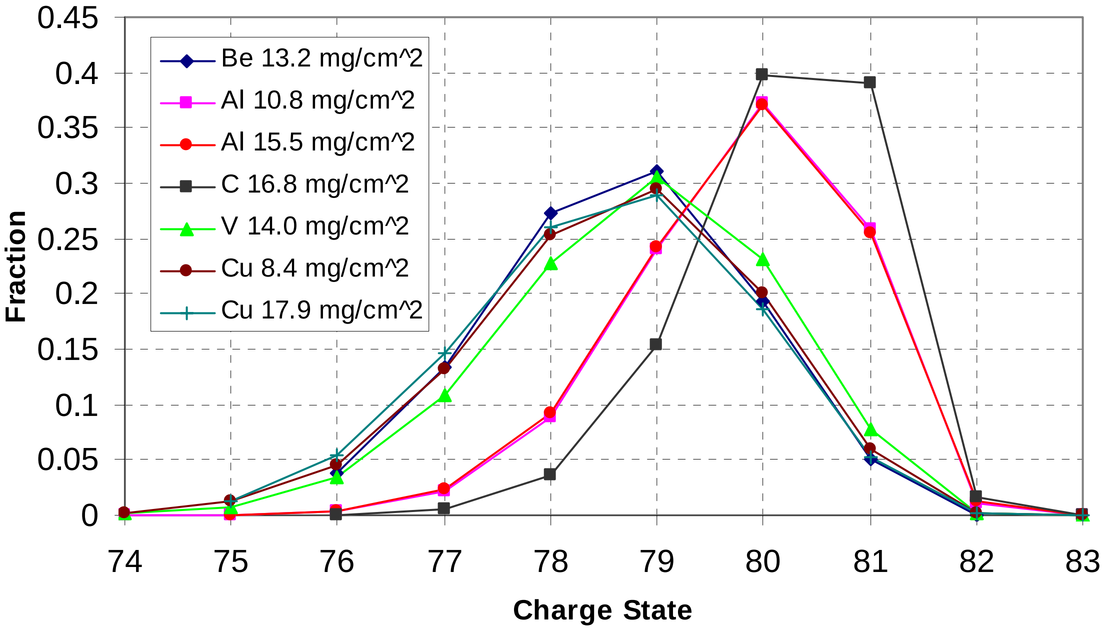

Figure 2. Charge Distribution for Various Materials with different thicknesses (for Al and Cu).

From these results, it appeared that a material with the atomic number ranging from 6 to 23 would possess good stripping characteristics (Figure 3). However, since carbon does not retain the good dimensional uniformity, it was not selected as a candidate for the stripper material. Other metallic materials, including Mg, Al, Ca, and Sc, are either chemically not very stable, having low melting point, or not readily available, and were not considered suitable candidates. These considerations left Na as a candidate for the liquid stripper and Ti and V as candidates for the solid stripper.

Also, the obtained charge distributions for Al 10.8 mg/cm^2 and 15.5 mg/cm^2 were almost identical and those for Cu 8.4 mg/cm^2 and 17.9 mg/cm^2 were very similar (Figure 2), indicating that the minimum thickness to reach equilibrium charge state distribution would have been approximately 10 mg/cm^2 for the present case.

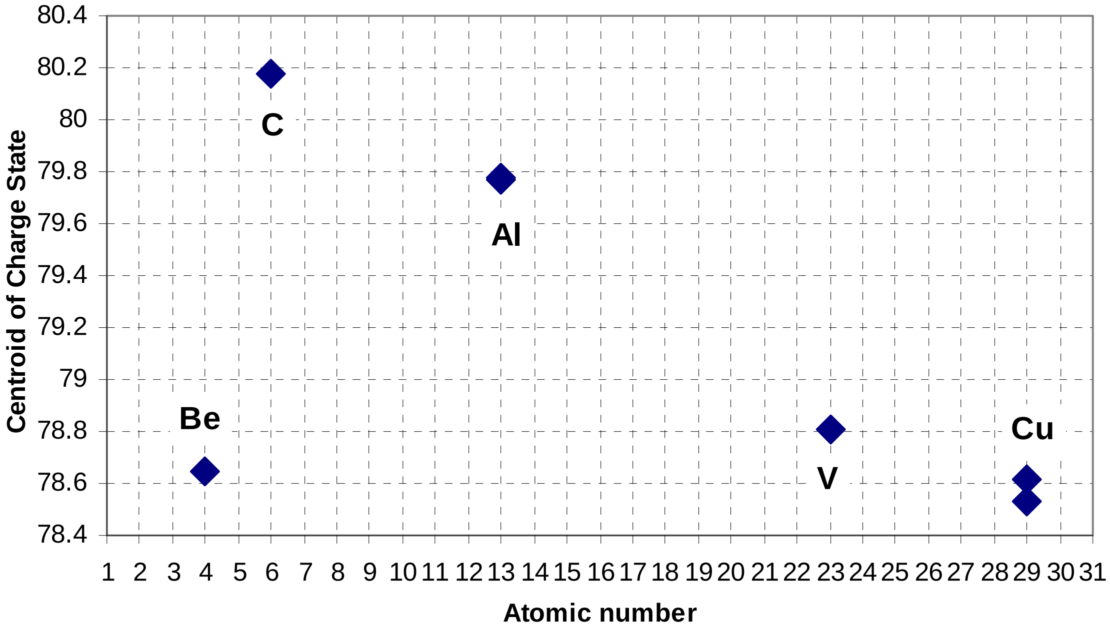

Figure 3. Change in Centroid of Charge Distribution as a Function of Atomic Number.

Nuclear Physics Calculations

To evaluate the volumetric heat generation in the stripper material, the function "EnergyLossInMatter_option(*Z_p*, *A_p*, *E*, *Z_t*, Δ*t*, *OPT*)" in *LISE for Excel* was used. This function requires 6 arguments; *Z_p* is the atomic number of an incident beam, *A_p* is the atomic mass number of an incident beam, *E* is the energy of an incident beam in MeV/nucleon, *Z_t* is the atomic number of a target material, Δ*t* is the thickness of a target in mg/cm^2, and *OPT* is a flag to specify which model is used in the calculation (i.e. 0 = Hubert, 1 = Ziegler, 2 = ATIMA). *OPT* = 2 was used in this study. The thickness, Δ*t*, is related to the actual physical distance Δ*x* in meters as,

$$\Delta t = 100\rho\Delta x\,, \tag{1}$$

where ρ is the density of the target in kg/m^3. The function, "EnergyLossInMatter_option" returns the value of the incident beam energy after penetrating through the target with a thickness of Δ*t* specified in the arguments shown in Figure 4.

Then, the volumetric heat generation, Q''', in the stripper was calculated as,

$$Q'''(x) = \frac{Q''}{E(0)} \frac{E(x-\Delta x) - E(x+\Delta x)}{2\Delta x}\,, \tag{2}$$

where Q'' is the initial energy flux density of the incident beam. The initial energy flux density, Q'' in W/m^2 was calculated as,

$$Q'' = \frac{E(0) \times A\_p \times i_B}{\pi R^2}, \tag{3}$$

where $E(0)$ is the energy of an incident beam in MeV/nucleon, i_B is the beam flux in particle μA, and R is the radius of the beam spot in meter. For the case of the incident U beam at the flux of 4 particle μA and energy of 85 MeV/nucleon, the initial energy flux is 80920 W (~81 kW) and the energy flux density is $Q'' = 1.03 \times 10^{11}$ W/m^2. Several results are presented in the following figures.

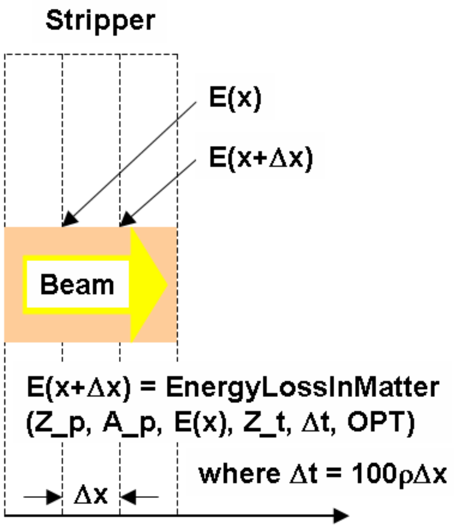

Figure 4. Calculating Energy Loss in the Target.

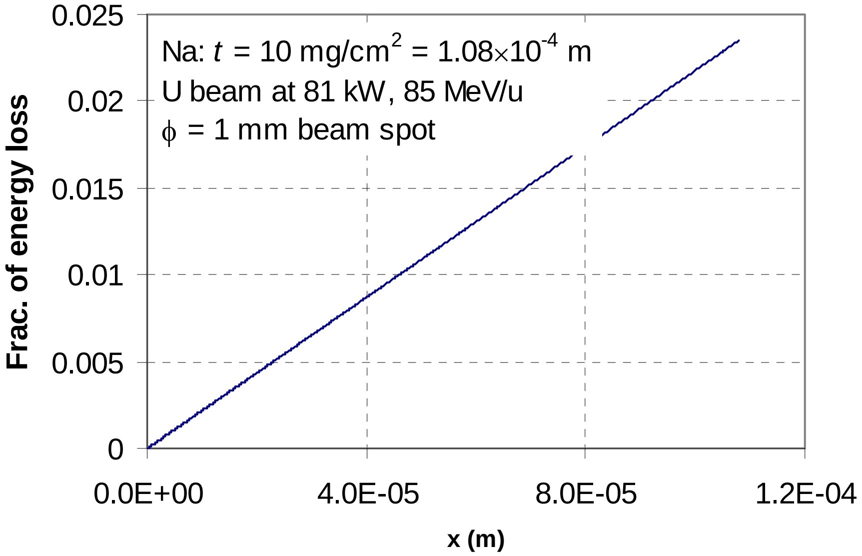

Figure 5. Beam Energy Loss Profile in the Na 2nd Stripper.

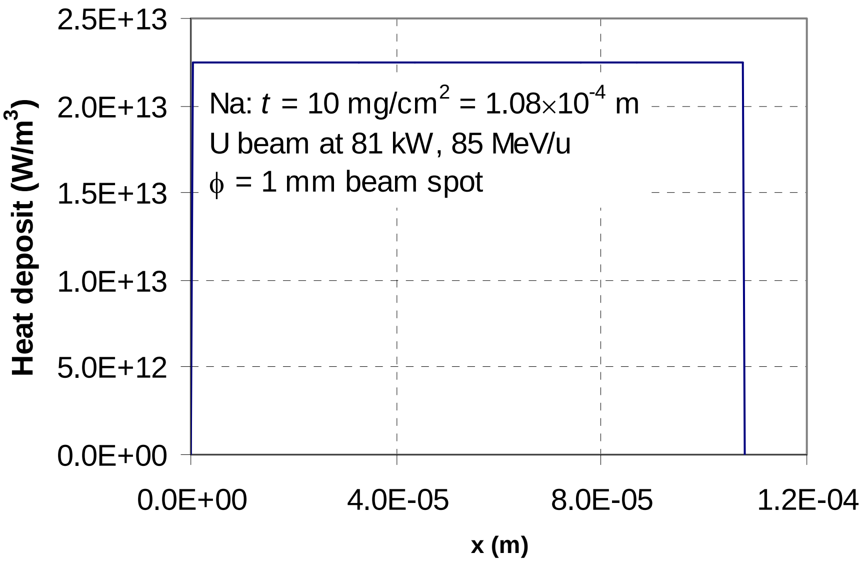

Figure 6. Volumetric Heat Generation in the Na 2nd Stripper.

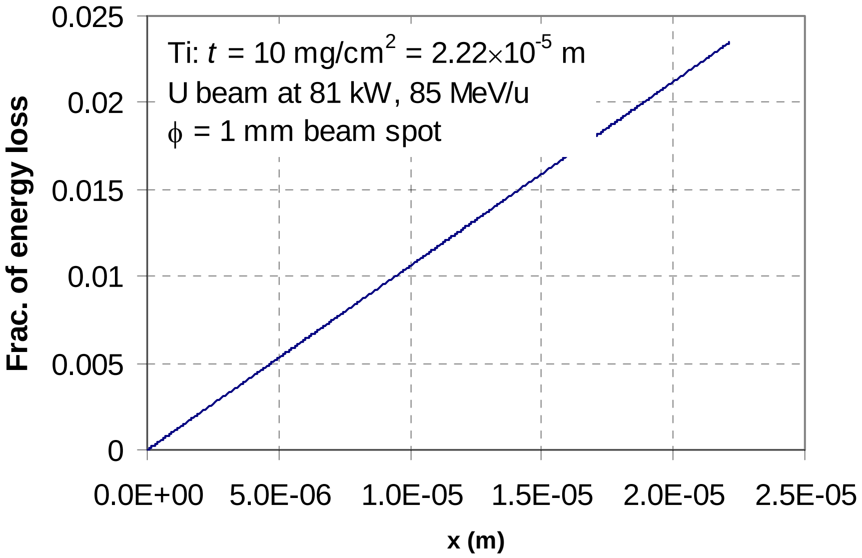

Figure 7. Beam Energy Loss Profile in the Ti 2nd Stripper.

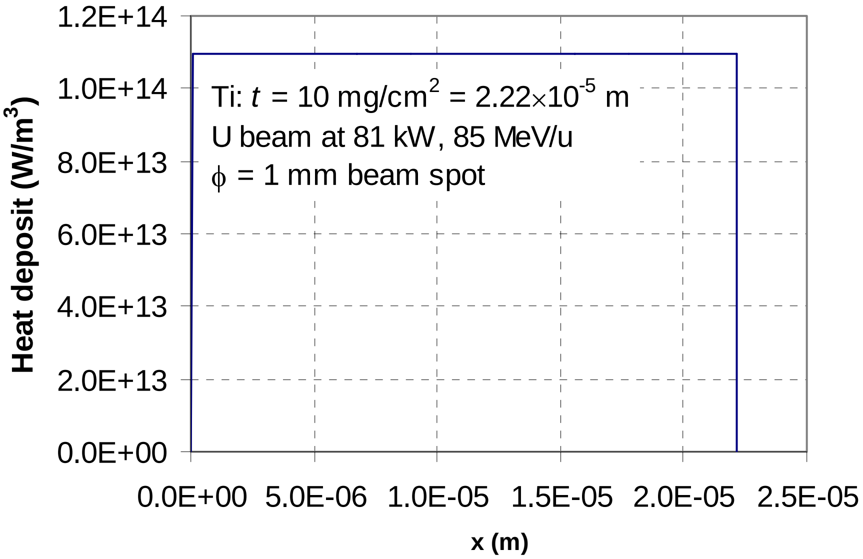

Figure 8. Volumetric Heat Generation in the Ti 2nd Stripper.

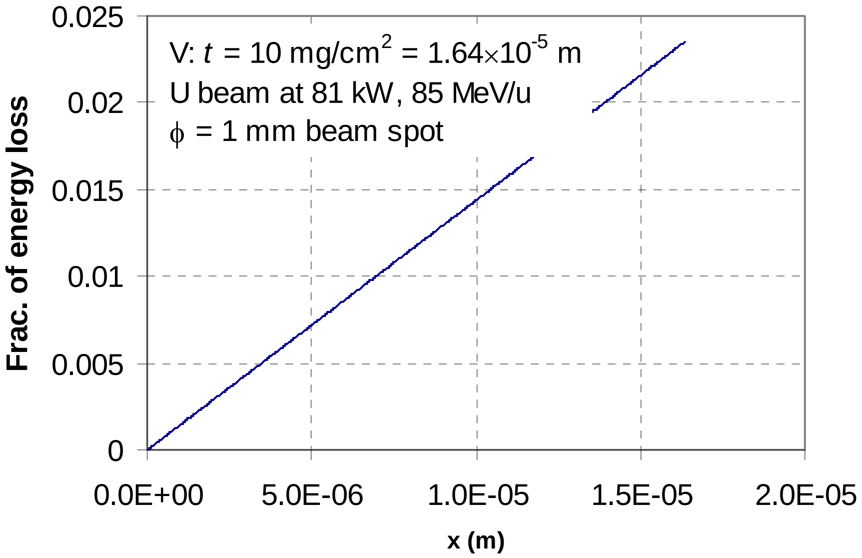

Figure 9. Beam Energy Loss Profile in the V 2nd Stripper.

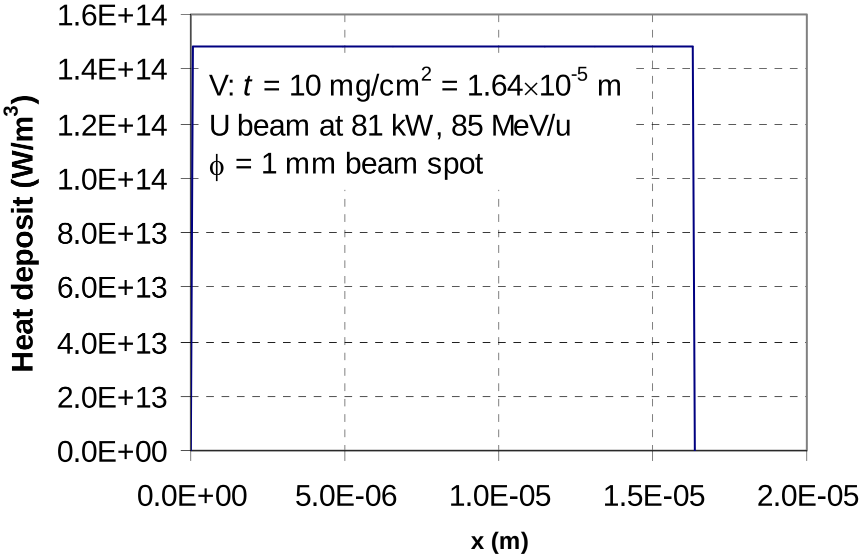

Figure 10. Volumetric Heat Generation in the V 2nd Stripper.

These figures suggest that this problem may be treated as a 2-D problem since the profile of the volumetric heat generation in the stripper material is uniform across the thickness of the material and the material is very thin. The fraction of the energy loss in the stripper materials at 10 mg/cm^2 thickness for a U beam at 85 MeV/u and 81 kW over a spot diameter of 1 mm is 0.0235 for all cases and corresponding energy loss, Q is 1900 W.

Thermal Calculations

Introducing some simplifications into this problem made it possible to use an analytical solution for a heat conduction problem of a moving, solid body. Since the stripper is very thin, the problem may be treated as a 2-D problem (Pittaway, 1964). In this analysis, a heat source that had Gaussian distribution was moving at the velocity, v m/s on a thin stripper material. The stripper had a constant, uniform, background temperature of T_∞. For simplicity, it was assumed that no heat transfer from the surfaces of the stripper took place. Then, the temperature distribution in the thin stripper material is given as (Pittaway, 1964),

$$T(a,b,m) = T_\infty + \frac{a^2 Q}{4\pi k t}\int_0^\infty \frac{1}{1+a^2 z}\exp\left[-\frac{(az+m)^2+b^2}{1+a^2 z}\right]dz\,, \tag{4}$$

where $a = \dfrac{4k}{\rho C_p v R}$, $b = \dfrac{y}{R}$, $m = \dfrac{x}{R}$. In this coordinate, the origin (x=0, y=0) is the center of the beam spot.

The integration in Eq. (4) was numerically performed using a Fortran code that implemented Romberg's method (Press et. al., 1992). Source codes and input file are attached in Appendix. Several results are presented in the following figures.

Sodium

The initial background temperature for Na was taken to be 381 K that is 10 K above the melting point of Na. Results of the thermal calculations while changing the Na film velocity from 10 m/s to 100 m/s are presented in Figures 11-16. Figures 11-13 present the spatial temperature distribution of the Na film along the beam line. Figures 14-16 show the 2D spatial temperature distribution of the film near the beam spot. These figures show that for a once-through Na stripper system, the calculated maximum temperatures of the Na film are ~1800 K, ~680 K, and ~530 K for Na film velocity of 10 m/s, 50 m/s, and 100 m/s, respectively.

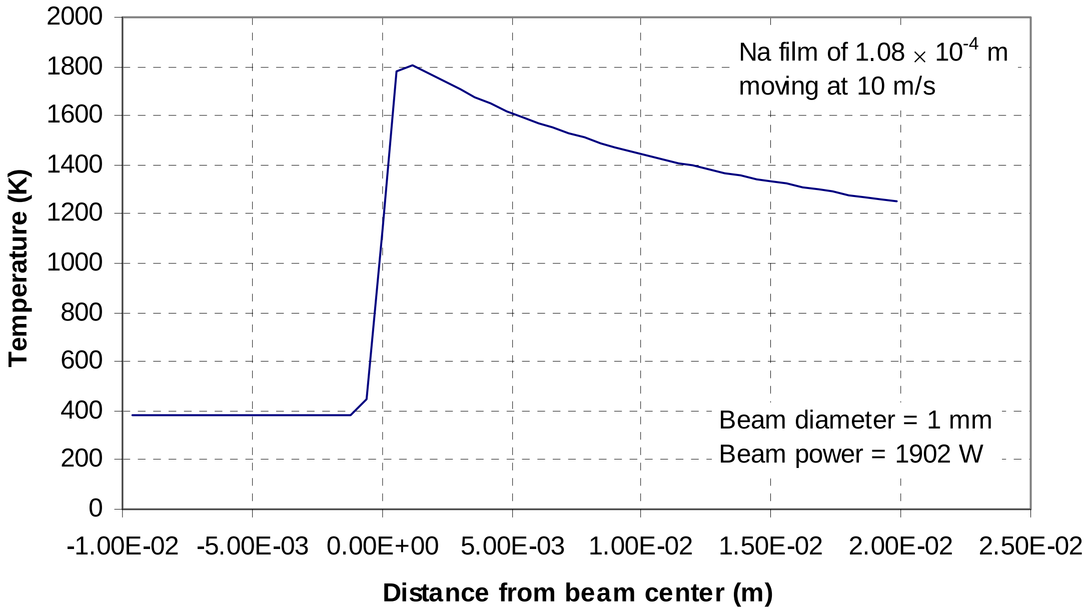

Figure 11. Temperature Distribution at Center (Na at 10 m/s).

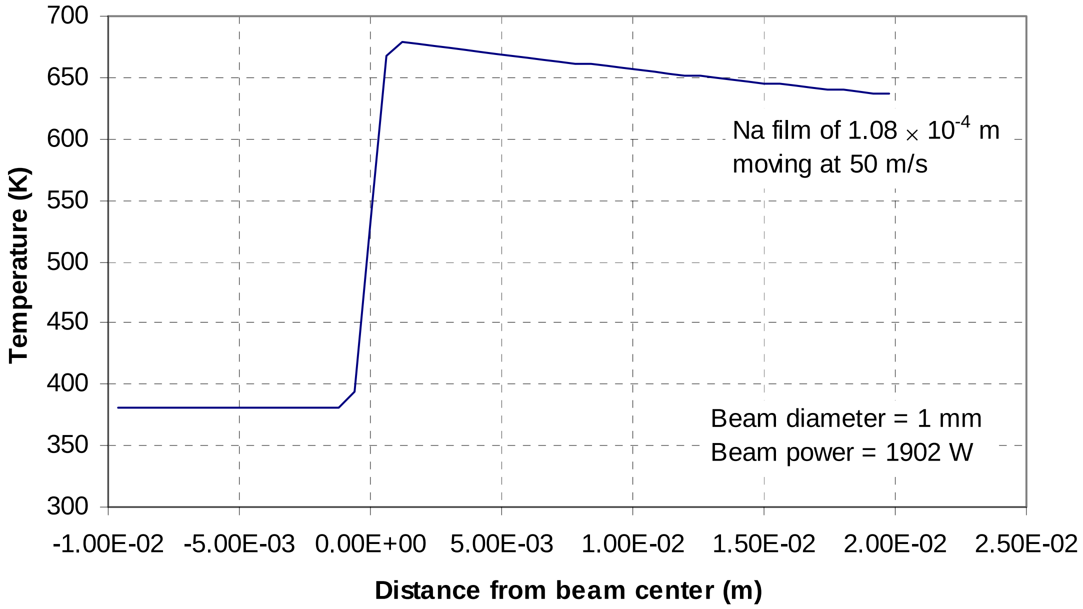

Figure 12. Temperature Distribution at Center (Na at 50 m/s).

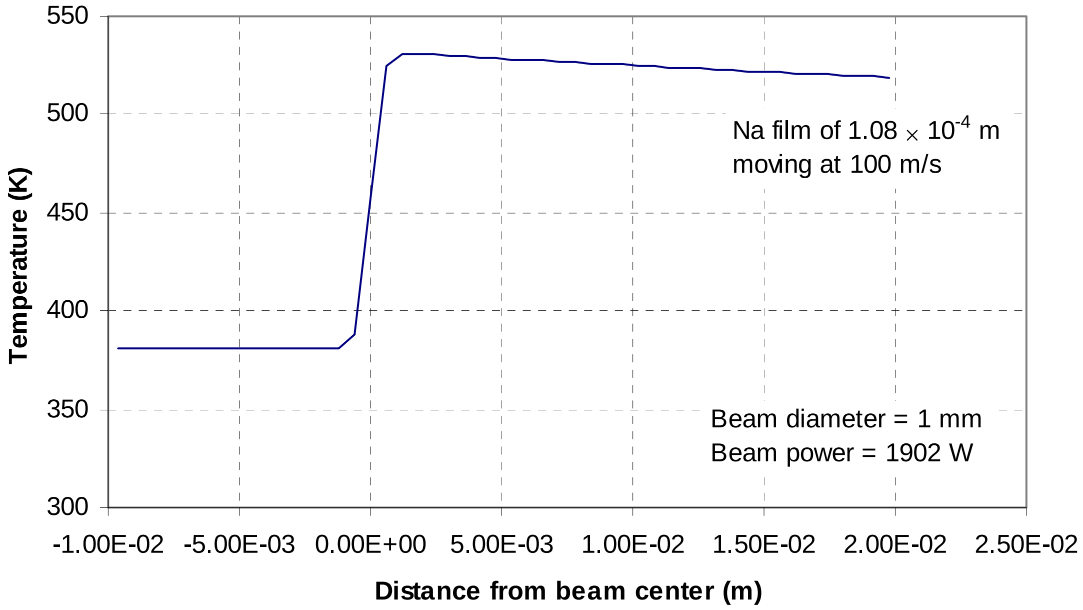

Figure 13. Temperature Distribution at Center (Na at 100 m/s).

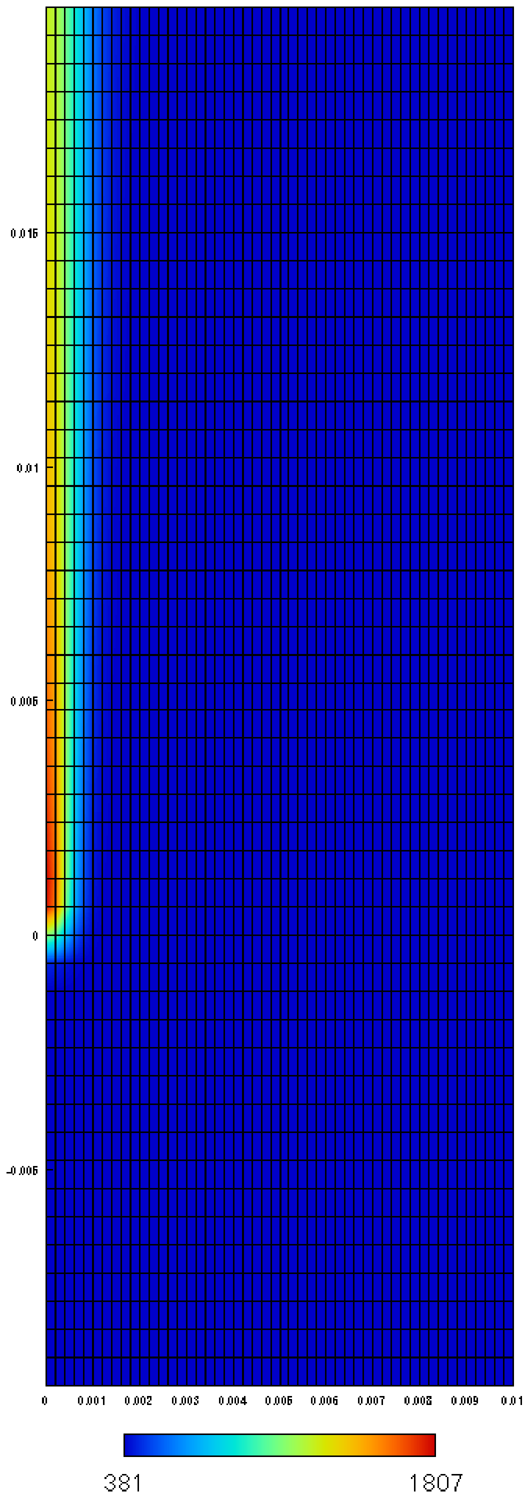

Figure 14. 2D Spatial Temperature Distribution (Na at 10 m/s).

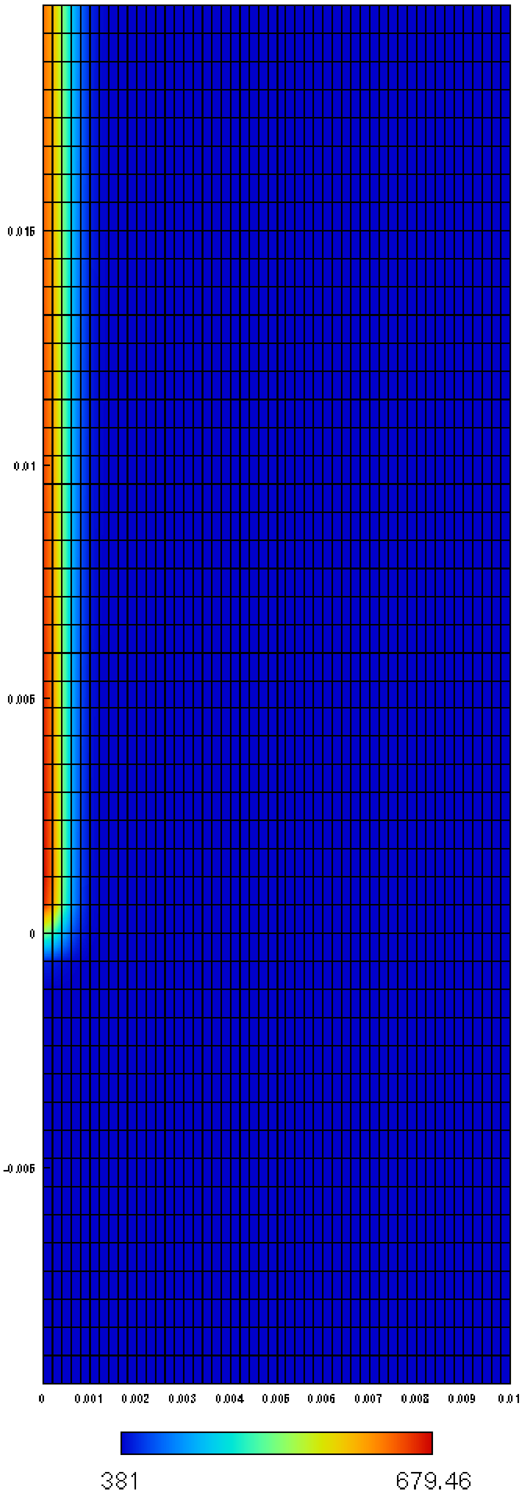

Figure 15. 2D Spatial Temperature Distribution (Na at 50 m/s).

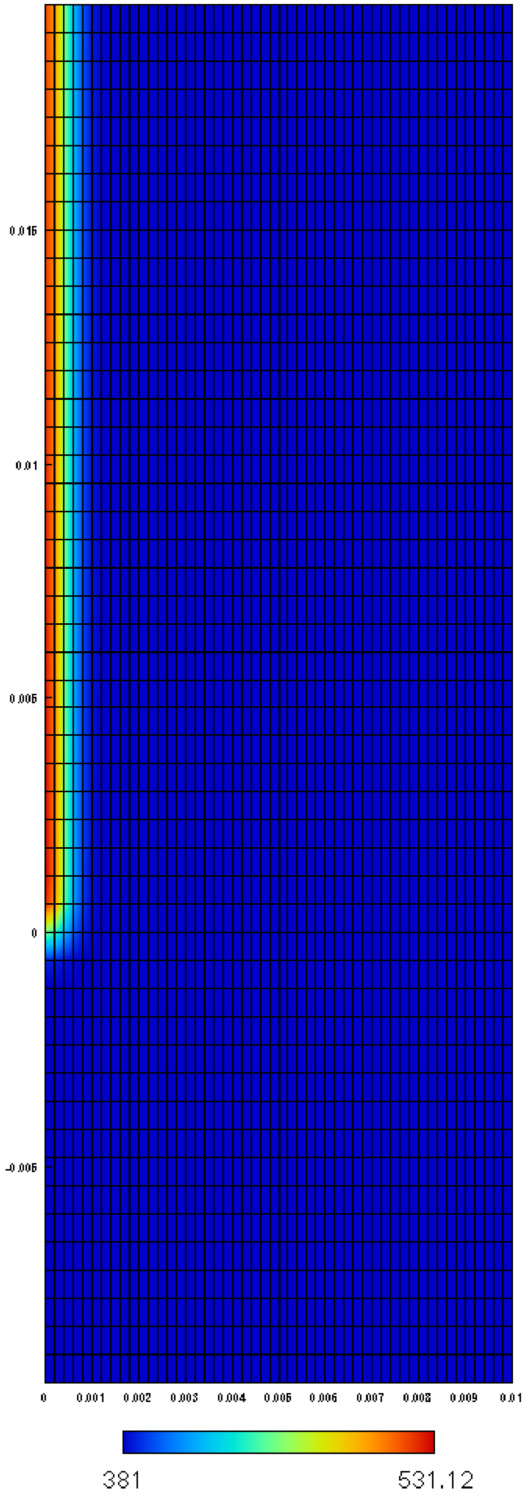

Figure 16. 2D Spatial Temperature Distribution (Na at 100 m/s).

Figures 14-16 indicate that the area where the calculated Na temperature becomes high is relatively narrow. The width of the high temperature area is ~x1-x2 beam diameter. Although liquid stripper concept does not suffer from thermal damaging due to melting, it is prone to release high pressure vapor depending on the vapor pressure of the stripper material. Next figure shows the saturation vapor pressure of Na.

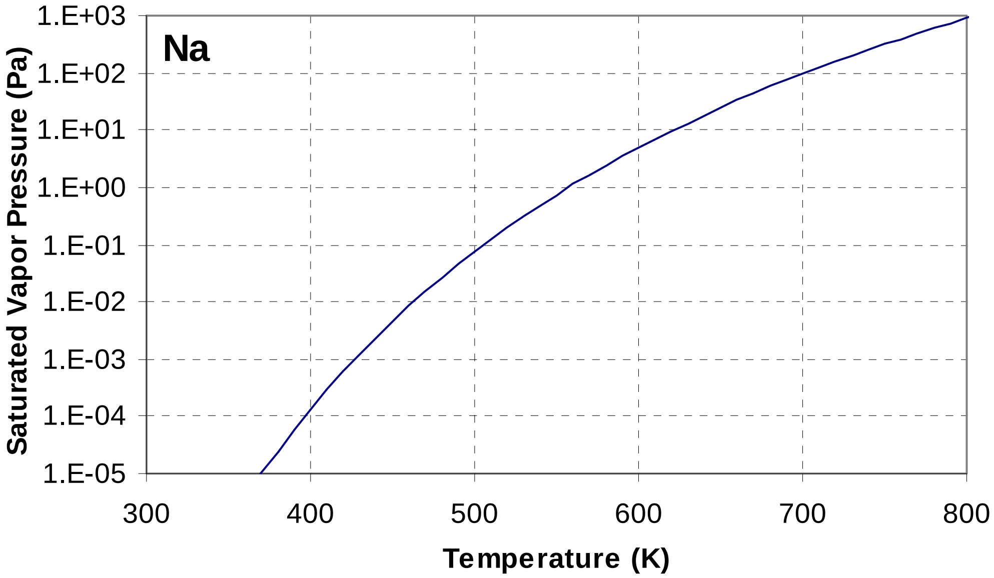

Figure 17. Na saturation vapor pressure as a function of temperature.

This figure shows that the values of the vapor pressure of Na are ~100 Pa and ~0.5 Pa at 680 K and 530 K, respectively. These vapor pressures seem to be too high and may not be acceptable for linac operation.

Titanium and Vanadium

The initial background temperature for Ti and V was room temperature (298 K). An initial conceptual design for the moving Ti and V stripper was a rotating disk. As the first design, it was assumed that the diameter of the disk was ~0.3 m and rotation of 3000 rpm from engineering view point. Thickness must have met the nuclear physical requirement (~15-20 micrometers). The beam spot was located near the edge of the disk. The circumference of the disk was therefore ~1 m and the beam spot moved at 50 m/s on the stripper foil. Figures 18, 19 present the spatial temperature distribution of the disk along the beam line, assuming linear beam spot movement.

Table 2. Properties of Ti and V.

	Z	density (kg/m^3)	k (W/m-K)	Cp (J/K-kg)		MP (K)	T at 1E-6 Torr (K)	T at 1E-5 Torr (K)	T at 1E-4 Torr (K)	T at 1E-3 Torr (K)	T at 1 Torr (K)	BP (K)
Ti	22	4507	21.9	520		1941	1500	1600	1715	1850	2450	3560
V	23	6110	31	489		2183				1950	2550	3680

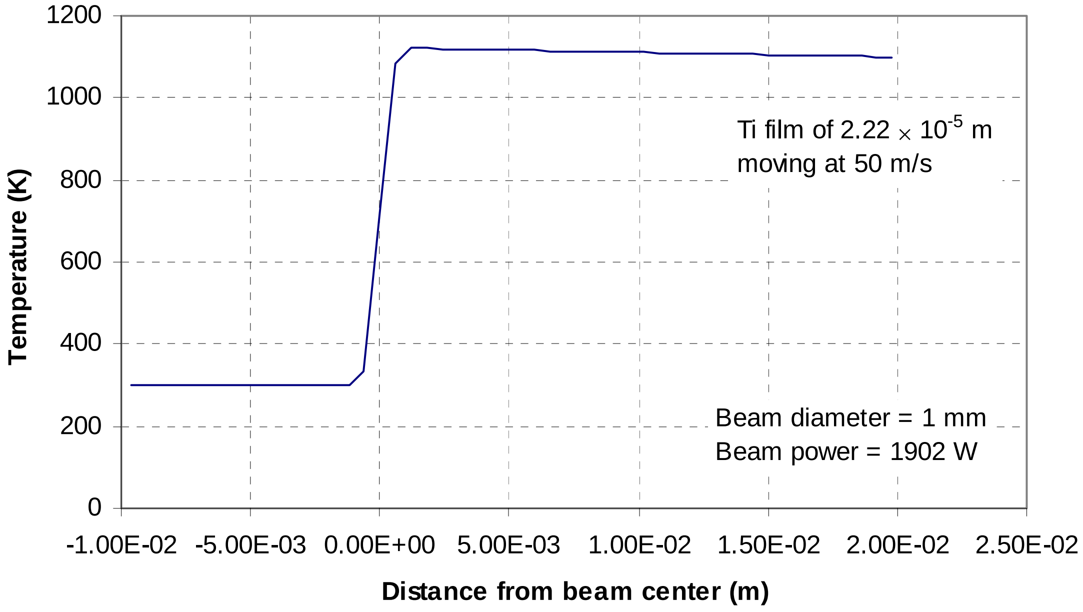

Figure 18. Temperature Distribution at Center (Ti at 50 m/s).

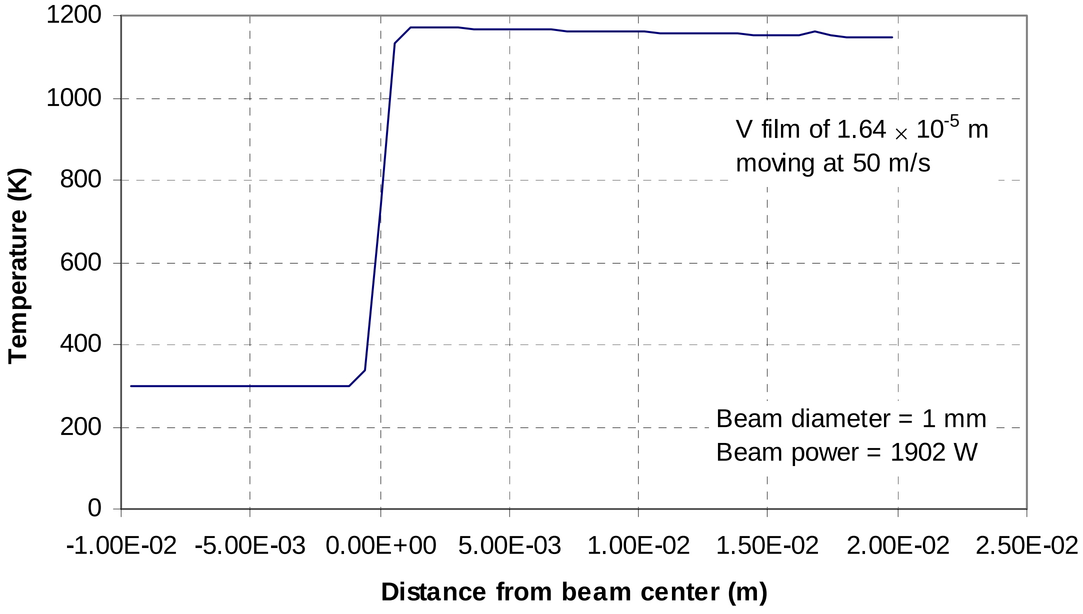

Figure 19. Temperature Distribution at Center (V at 50 m/s).

These figures show that the calculated maximum temperatures of the stripper disk after the first beam shot are ~1120 K and ~1170 K for Ti and V at 50 m/s, respectively. Instantaneous temperature jumps after first beam shot are ~820 K and ~870 K for Ti and

V, respectively. Next figures show 2D temperature distribution of the stripper moving at 50 m/s near the beam spot after the first beam shot.

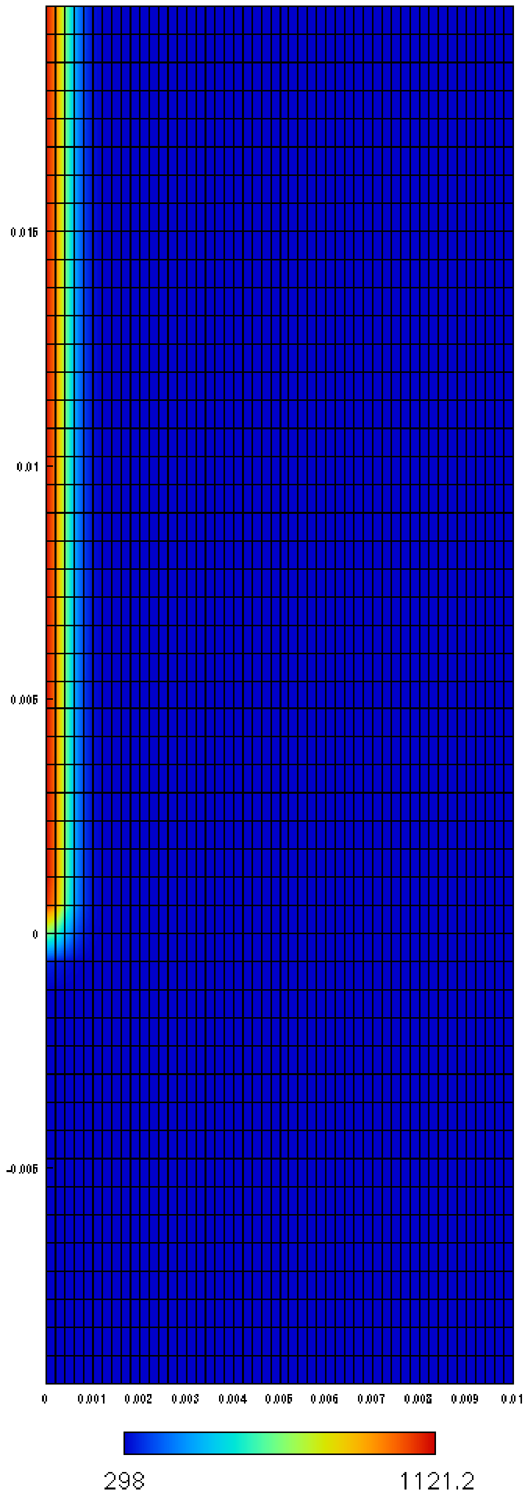

Figure 20. 2D Spatial Temperature Distribution (Ti at 50 m/s).

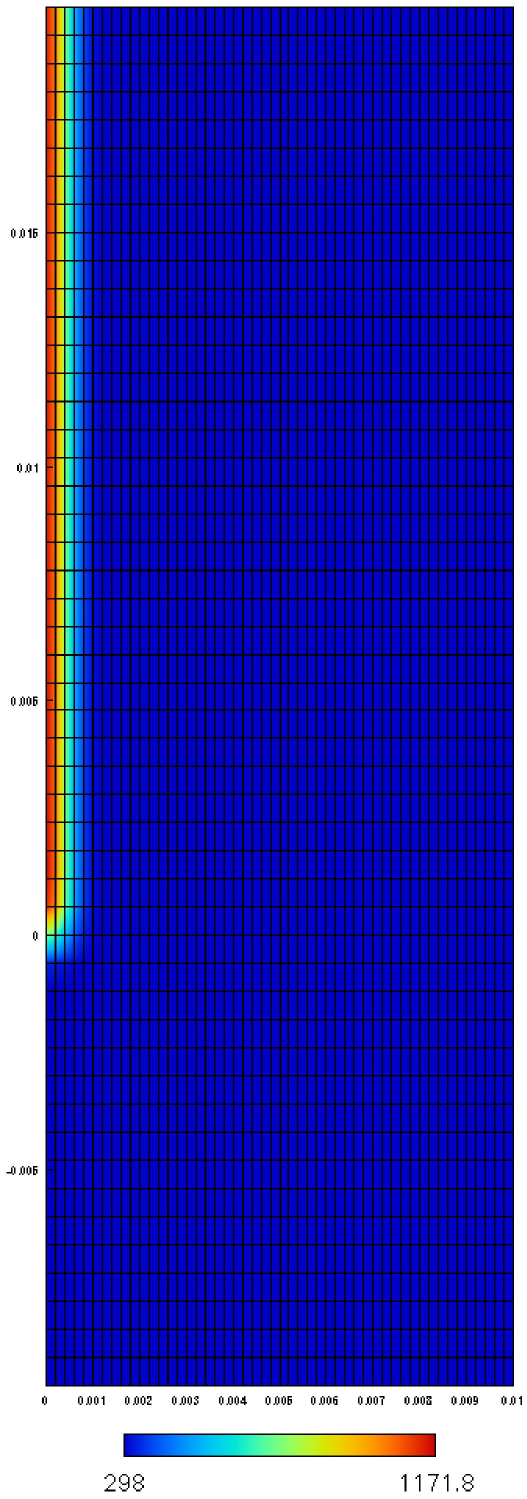

Figure 21. 2D Spatial Temperature Distribution (V at 50 m/s).

Again, these figures indicate that the area where temperature becomes high is relatively narrow. The width of the high temperature area in the proximity of the beam spot is limited within ~x1-x2 beam diameter. For the rotating disk stripper, the same location will eventually be irradiated by the beam again. For example, if the disk is not wobbled in radial direction, the same spot on the disk will be beamed again after one rotation. The next figures show how thermal energy spreads in the disk by conduction when no radiation heat loss to the outside was assumed. It must be noted that if no wobbling was assumed, the second initial temperature for the beam spot would be ~700 K, which is ~400 K higher than the first initial temperature. The width of the heated area became ~4-5 mm after one rotation due to spreading of thermal energy by conduction through the disk. This means that if the disk is shifted in radial direction by more than ~2-3 mm after one rotation, the beam will hit the place that has not been thermally affected by the beam. Similar results and observations were also obtained for the V disk case.

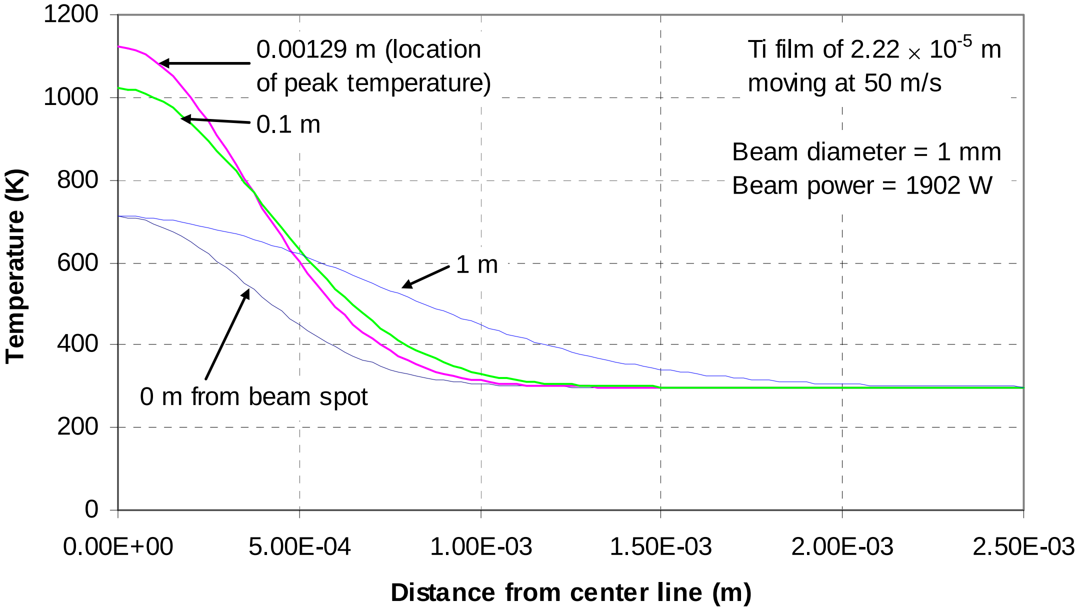

Figure 22. Temperature distribution across the stripper perpendicular to the direction of motion for Ti at 50 m/s.

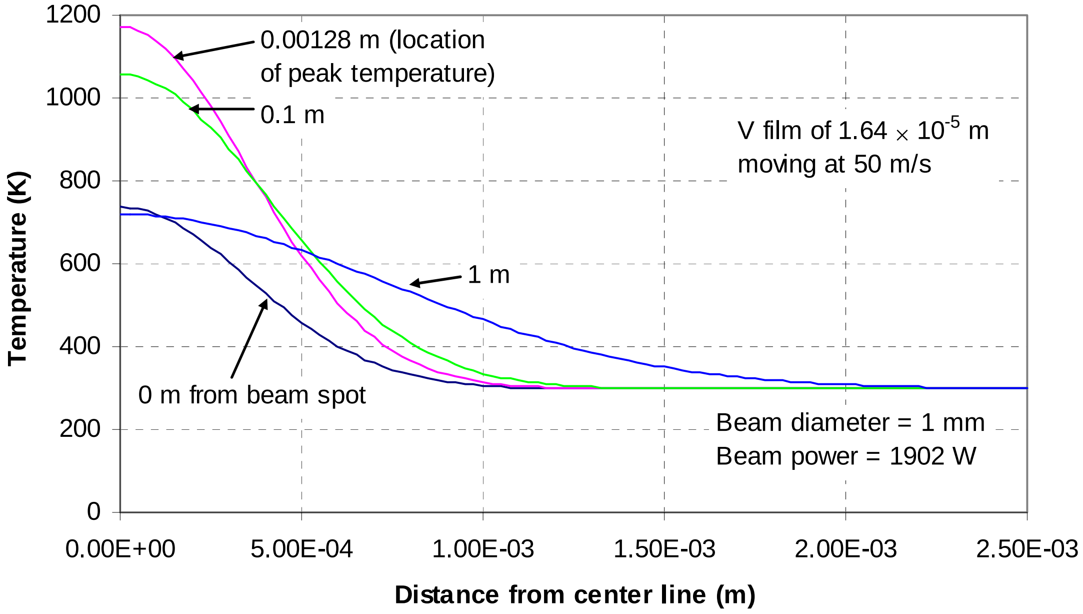

Figure 23. Temperature distribution across the stripper perpendicular to the direction of motion for V at 50 m/s.

Eventually, the radiation heat loss from the surface of the disk must balance the thermal energy deposition from the beam that is ~2000 W. For a simple estimation, the stripper disk was assumed to be a disk whose circumference with a certain width was held at a constant, uniform temperature held in vacuum whose surrounding temperature was the room temperature. The disk may be blackened to increase the emissivity to enhance radiative heat transfer. The enhanced emissivity of 0.8 was used in the following calculations. Both sides of the disk were assumed to contribute radiation.

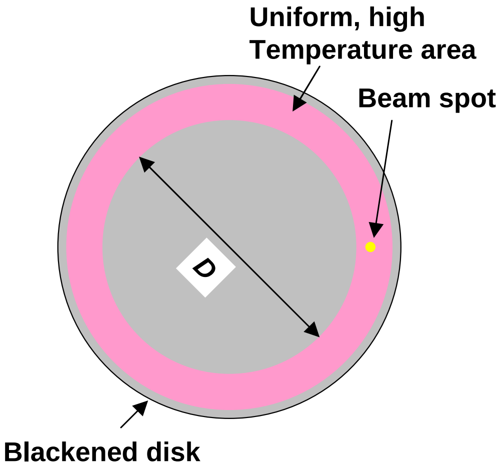

Figure 24. Modeled rotating disk stripper for radiation analysis.

The amount of heat removed by radiation was calculated as,

$$Q_{rad}(T) = A\sigma\varepsilon F(T^4 - T_{BG}{}^4), \tag{5}$$

where A is the surface area, σ is Stephan-Boltzmann constant (= 5.67×10^{-8} W/m^2-K^4), ε is an emissivity of the wall (0.8 for all cases), F is a view factor (taken to be 1 for all cases), and T_{BG} is the background temperature (taken to be 300 K for all cases).

Table 3. Required surface temperature to radiatively remove 2000 W.

inner radius (m)	0.15	0.15	0.15	0.15	0.3	0.3	0.3
width of hot area (m)	0.01	0.025	0.05	0.075	0.01	0.025	0.05
emissivity	0.8	0.8	0.8	0.8	0.8	0.8	0.8
Background (K)	300	300	300	300	300	300	300
radiation (W)	2000	2000	2000	2000	2000	2000	2000
temperature required (K)	**1227.7**	**966.3**	**799.8**	**712.4**	**1037.5**	**822.3**	**688.0**

For linac operation, low vapor pressure in the cavity is desirable (~10^{-4} Pa). For Ti, upper limit temperature was set at 1500 K and 1600 K for V. Since instantaneous temperature increases are ~820 K and ~870 K for Ti and V, respectively, the highest background temperature for both cases needs to be < ~700 K. One more important point is that the beam line must cross after one rotation when the direction of shifting changes at the edges of the hot area (see Figure 25). This means that the point at which the beam line crosses experiences beam shot twice after only one rotation (~1 m after first beam shot for a disk with ~0.3 m diameter). Therefore, an additional ~700 K temperature increase has to be added. As a result, it is not possible to maintain the temperature at the beam cross point below 1500 K for Ti or 1600 K for V, unless disk size and disk rotation velocity are appropriately adjusted, which may create engineering issues.

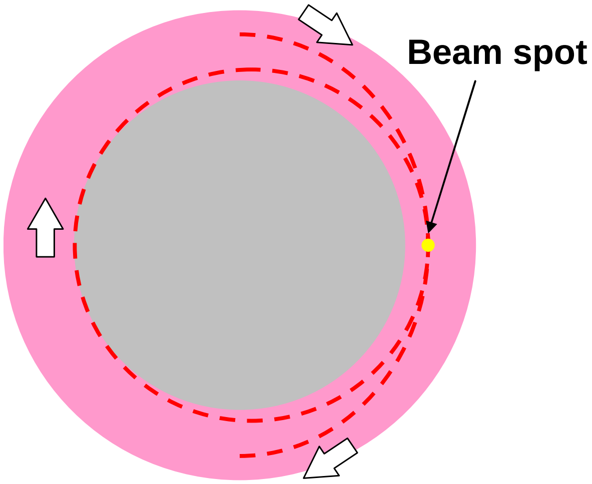

Figure 25. Beam path when direction of shifting changes.

It must be noted that since the average temperature of the hot area will be higher than the uniform background temperature calculated in the above table due to heating from beam irradiation, this estimation is expected to be somewhat conservative. It is apparent that the width of the hot area needs to be > ~0.05 m, unless the disk size is much larger. Since it is desirable to shift the beam line by ~2-3 mm after each rotation to avoid producing hot spots, 10-20 beam lines can fit within the width of the hot area.

References

http://dnr080.jinr.ru/lise/lise.html (LISE++).

Pittaway, L. G., (1964) "The temperature distributions in thin foil and semi-infinite targets bombarded by an electron beam," *Brit. J. Appl. Phys.*, **v** 15, pp967-982.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., (1992) *Numerical Recipes in FORTRAN: The Art of Scientific Computing, Second Edition*, Cambridge University Press.

http://heliumclub.usc.edu/Refs/Vapor.htm (vapor pressure data for Ti).

http://www.puretechinc.com/data_sheets/vanadium.htm (vapor pressure data for V).

Appendix

Fortran program xTFXR

```
      PROGRAM xTFXR
C     Calculating maximum temperature within thin film.
C     Assuming uniform temperature in the film.
C     Maximum temperature occurs in the trace of the center of the
C     beam spot.

      IMPLICIT NONE

      CHARACTER*20 filename
c      CHARACTER*8 CRUN
      CHARACTER*1 ONEDFLAG

      INTEGER i,j,counter
c      INTEGER NRUN
      INTEGER NXgrid,NYgrid

      DOUBLE PRECISION K,RO,CP
      DOUBLE PRECISION Tmax0,T0
      DOUBLE PRECISION t
      DOUBLE PRECISION v
      DOUBLE PRECISION W

      DOUBLE PRECISION Qt
      DOUBLE PRECISION R

      DOUBLE PRECISION HY,WX

      DOUBLE PRECISION KAP,A,B

      DOUBLE PRECISION m,x,y
      DOUBLE PRECISION tx,tmax
      DOUBLE PRECISION Qmax

      DOUBLE PRECISION db,dm

      EXTERNAL temp,mmax
      DOUBLE PRECISION temp,mmax

      call WELCOME()


C     input name for output file
      write(*,*)"type output filename"
      read(*,*)filename

C     specify type of output
      write(*,*)"1D distribution along flow direction, y or n?"
      read(*,*)ONEDFLAG

C     read parameters from input.dat file
      call READINP(K,RO,CP,Tmax0,T0,t,v,W,Qt,R,HY,WX,NYgrid,NXgrid)
```

```
C     calculate constants
      KAP=K/(RO*CP)
      A  =4.0d0*KAP/(v*R)
      B  =W/R

      OPEN(UNIT=10,FILE=filename)
C     print header in output file
      write(UNIT=10,FMT=*)"target properties"
      write(UNIT=10,FMT=*)"thermal conductivity",K," W/m-K"
      write(UNIT=10,FMT=*)"density",RO," kg/m^3"
      write(UNIT=10,FMT=*)"specific heat",CP," J/K-kg"
      write(UNIT=10,FMT=*)"initial film temperature",T0," K"
      write(UNIT=10,FMT=*)"film thickness",t," m"
      write(UNIT=10,FMT=*)"film velocity",v," m/s"
c      write(UNIT=10,FMT=*)"location relative to beam spot",W," m"
      write(UNIT=10,FMT=*)
      write(UNIT=10,FMT=*)"beam properties"
      write(UNIT=10,FMT=*)"power",Qt," W"
      write(UNIT=10,FMT=*)"beam radius",R," m"

write(UNIT=10,FMT=*)"********************************************"
      write(UNIT=10,FMT=*)"maximum temperature in target"
      write(UNIT=10,FMT=*)"location (m), max. temp (K)"

C     calculate maximum temperature
      m   =mmax(A)
      tmax=temp(K,RO,CP,T0,t,v,W,Qt,R,A,0.0d0,m)
C     calculate beam spot location for maximum temperature
      x=-m*R
      write(*,FMT=100)x,"m ",tmax,"K "
      write(UNIT=10,FMT=101)x,"     ",tmax


write(UNIT=10,FMT=*)"********************************************"
C     calculate maximum power
      Qmax = (Tmax0-T0)/(tmax-T0) * Qt
      write(*,*)"maximum allowable power",Qmax," W"
      write(UNIT=10,FMT=*)"maximum allowable power",Qmax," W"


write(UNIT=10,FMT=*)"********************************************"
      write(UNIT=10,FMT=*)"2D temperature distribution"
      write(UNIT=10,FMT=*)"Number of grid ",NXgrid," x ",NYgrid
      write(UNIT=10,FMT=*)"X (m), Y (m), T (K)"

      db=WX/dble(NXgrid)
      dm=HY/dble(NYgrid)
C     calculate temperature profile
C     sweep vertically
      counter=0
      do j=NYgrid/3,-(NYgrid*2)/3,-1
          y=-dble(j)*dm
          m=-y/R

          write(*,*)"calculating at ",y,"..."
C     sweep horizontally
```

```
          if(ONEDFLAG.eq."y")then
              x=0.0d0
              b=x/R
              tx=temp(K,RO,CP,T0,t,v,W,Qt,R,A,b,m)

              write(UNIT=10,FMT=102)x,"   ",y," ",tx
          else
              do i=0,NXgrid,1
                  x=dble(i)*db
                  b=x/R
                  tx=temp(K,RO,CP,T0,t,v,W,Qt,R,A,b,m)

                  write(UNIT=10,FMT=102)x,"     ",y," ",tx

c                  write(*,*)y,"    ",tx
c                  write(UNIT=10,FMT=102)x,", ",y,", ",tx
              enddo
          endif
      enddo

      CLOSE(UNIT=10)
100   FORMAT(e12.6,a4,e12.6,a2)
101   FORMAT(e12.6,a4,e12.6)
102   FORMAT(e12.6,a1,e12.6,a1,e12.6)
103   FORMAT(i8.8)

      END
```

Fortran subroutine WELCOME

```
      SUBROUTINE welcome()
C     Calculating maximum temperature of thin film X-ray source.
C     Assuming uniform temperature in the film.
C     Maximum temperature occurs on the trace of the center of the
C     beam spot.

      IMPLICIT NONE

      WRITE(*,*)"                                                  "
      WRITE(*,*)"       2nd stripper thermal analysis code         "
      WRITE(*,*)"                                                  "
      WRITE(*,*)"                 Momozaki, Yoichi                 "
      WRITE(*,*)"                                                  "
      WRITE(*,*)"                    ver. 2.0                      "
      WRITE(*,*)"                    2005/05/12                    "
      WRITE(*,*)"                                                  "
      WRITE(*,*)"++++++++++++++++++++++++++++++++++++++++++++++++++"
      WRITE(*,*)"+                                                +"
      WRITE(*,*)"+                                                +"
      WRITE(*,*)"++++++++++++++++++++++++++++++++++++++++++++++++++"
      WRITE(*,*)"                                                  "
      WRITE(*,*)"  Provide input.dat file to specify various       "
      WRITE(*,*)"properties of system and parameters.              "
      WRITE(*,*)"                                                  "
      WRITE(*,*)"                                                  "
```

```
      PAUSE "hit return key to proceed"
C      WRITE(*,*)"Computation started..."
      WRITE(*,*)
      END
```

Fortran subroutine READINP

```
      SUBROUTINE
READINP(K,RO,CP,Tmax,T0,t,v,W,Qt,R,HY,WX,NYgrid,NXgrid)

      IMPLICIT NONE

      INTEGER NXgrid,NYgrid

      DOUBLE PRECISION K,RO,CP
      DOUBLE PRECISION Tmax,T0
      DOUBLE PRECISION t
      DOUBLE PRECISION v
      DOUBLE PRECISION W

      DOUBLE PRECISION Qt
      DOUBLE PRECISION R

      DOUBLE PRECISION HY,WX

      OPEN(UNIT=101,FILE="input.dat",ACTION="READ")
      READ(UNIT=101,FMT=*)
      READ(UNIT=101,FMT=*)
      READ(UNIT=101,FMT=*)K,RO,CP
      READ(UNIT=101,FMT=*)
      READ(UNIT=101,FMT=*)Tmax
      READ(UNIT=101,FMT=*)
      READ(UNIT=101,FMT=*)T0
      READ(UNIT=101,FMT=*)
      READ(UNIT=101,FMT=*)t
      READ(UNIT=101,FMT=*)
      READ(UNIT=101,FMT=*)v
      READ(UNIT=101,FMT=*)
      READ(UNIT=101,FMT=*)W
      READ(UNIT=101,FMT=*)
      READ(UNIT=101,FMT=*)Qt
      READ(UNIT=101,FMT=*)
      READ(UNIT=101,FMT=*)R
      READ(UNIT=101,FMT=*)
      READ(UNIT=101,FMT=*)HY,WX
      READ(UNIT=101,FMT=*)
      READ(UNIT=101,FMT=*)NYgrid,NXgrid

      CLOSE(UNIT=101)

      return
      END
```

Fortran functions MMAX, INTEG2, FUNCT2, DZRIDDR2

```
      DOUBLE PRECISION FUNCTION mmax(A)
      IMPLICIT NONE

C     Compute mmax in double precision.
      DOUBLE PRECISION xacc
      PARAMETER(xacc=1.0d-3)
c      PARAMETER(xacc=1.0d-6)

      DOUBLE PRECISION A,B
      DOUBLE PRECISION LL,UL
      DOUBLE PRECISION test

      EXTERNAL integ2,dzriddr2
      DOUBLE PRECISION integ2,dzriddr2

      B = 0.0d0
      UL= 0.0d0
      LL=-1.0d0

      test=0.0d0
      do while(test.le.0.0d0)
          test=integ2(A,B,LL)
            if(test.le.0.0d0)LL=2.0d0*LL
      enddo

      mmax=dzriddr2(integ2,A,B,LL,UL,xacc)

      RETURN
      END
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      DOUBLE PRECISION FUNCTION integ2(A,B,m)
      IMPLICIT NONE

      INTENT(IN):: A,B,m

      DOUBLE PRECISION A,B
      DOUBLE PRECISION m

      DOUBLE PRECISION s

      DOUBLE PRECISION LL
      DOUBLE PRECISION UL

      EXTERNAL dmidrcp
      EXTERNAL funct2
C     function (funct) to be integrated must be in a form:
C     f(A,m,z) and integration is performed as:
C     Integral f(A,m,z) dz from 0 to Infinity.
C     s = result.
      DOUBLE PRECISION funct2

C     lower limit of integration, LL,
C     upper limit of integration, UL.
      LL=0.0d0
```

```
      UL=1.0d0
C     Improper Romberg Integration (0 -> INF) using dmidrcp routine.
C     UL is not used in calculation.
      call dqromo(funct2,LL,UL,s,A,B,m,dmidrcp)

      integ2=s
      return
      END
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      DOUBLE PRECISION FUNCTION funct2(A,B,m,z)
      IMPLICIT NONE

      INTENT(IN):: A,B,m,z

      DOUBLE PRECISION A,B,m,z

      funct2= -(m+A*z)/(1.0d0+A*A*z)**2.0d0
     &       *dexp( -( (A*z+m)*(A*z+m) + B*B )/(1.0d0+A*A*z) )

      return
      END
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      FUNCTION dzriddr2(func,A,B,x1,x2,xacc)
      IMPLICIT NONE
      INTEGER MAXIT
      DOUBLE PRECISION dzriddr2,x1,x2,xacc,func,UNUSED
      DOUBLE PRECISION A,B
      PARAMETER (MAXIT=60,UNUSED=-1.11d30)
      EXTERNAL func
CU    USES func
      INTEGER j
      DOUBLE PRECISION fh,fl,fm,fnew,s,xh,xl,xm,xnew
      fl=func(A,B,x1)
      fh=func(A,B,x2)
      if( (fl.gt.0.0d0.and.fh.lt.0.0d0).or.
     &    (fl.lt.0.0d0.and.fh.gt.0.0d0))then
        xl=x1
        xh=x2
        dzriddr2=UNUSED
        do 11 j=1,MAXIT
          xm=0.5d0*(xl+xh)
          fm=func(A,B,xm)
          s=dsqrt(fm**2.0d0-fl*fh)
          if(s.eq.0.0d0)return
          xnew=xm+(xm-xl)*(dsign(1.0d0,fl-fh)*fm/s)
          if (dabs(xnew-dzriddr2).le.xacc) return
          dzriddr2=xnew
          fnew=func(A,B,dzriddr2)
          if (fnew.eq.0.0d0) return
          if(dsign(fm,fnew).ne.fm) then
            xl=xm
            fl=fm
            xh=dzriddr2
            fh=fnew
          else if(dsign(fl,fnew).ne.fl) then
            xh=dzriddr2
            fh=fnew
```

```
          else if(dsign(fh,fnew).ne.fh) then
            xl=dzriddr2
            fl=fnew
          else
            pause 'never get here in zriddr'
          endif
          if(dabs(xh-xl).le.xacc) return
11      continue
        pause 'dzriddr2 exceed maximum iterations'
      else if (fl.eq.0.0d0) then
        dzriddr2=x1
      else if (fh.eq.0.0d0) then
        dzriddr2=x2
      else
        pause 'root must be bracketed in dzriddr2'
      endif
      return
      END
```

Fortran function TEMP

```
      DOUBLE PRECISION FUNCTION temp(K,RO,CP,T0,t,v,W,Qt,R,A,B,m)
      IMPLICIT NONE

      DOUBLE PRECISION K,RO,CP
      DOUBLE PRECISION T0
      DOUBLE PRECISION t
      DOUBLE PRECISION v
      DOUBLE PRECISION W
      DOUBLE PRECISION Qt
      DOUBLE PRECISION R
      DOUBLE PRECISION A,B,m

      DOUBLE PRECISION dummy

      EXTERNAL integ
      DOUBLE PRECISION integ

      DOUBLE PRECISION PI
      PARAMETER (PI=3.141592653589793d0)

      dummy = RO
      dummy = CP
      dummy = v
      dummy = W
      dummy = R

      temp= T0 + A*A*Qt/(4.0d0*PI*K*t) * integ(A,B,m)

      END
```

Fortran function INTEG, subroutines DQROMO, POLINT, DMIDRCP, DMIDEXP

```
      DOUBLE PRECISION FUNCTION integ(A,B,m)
```

```
      IMPLICIT NONE

      INTENT(IN):: A,B,m

      DOUBLE PRECISION A,B
      DOUBLE PRECISION m

      DOUBLE PRECISION s

      DOUBLE PRECISION LL
      DOUBLE PRECISION UL

      EXTERNAL dmidexp,dmidrcp
      EXTERNAL funct
C     function (funct) to be integrated must be in a form:
C     f(A,m,z) and integration is performed as:
C     Integral f(A,m,z) dz from 0 to Infinity.
C     s = result.
      DOUBLE PRECISION funct

C     lower limit of integration, LL,
C     upper limit of integration, UL.
      LL=0.0d0
      UL=1.0d0
C     Improper Romberg Integration (0 -> INF) using dmidrcp routine.
C     UL is not used in calculation.
      call dqromo(funct,LL,UL,s,A,B,m,dmidrcp)
c      call dqromo(funct,LL,UL,s,A,B,m,dmidexp)

      integ=s
      return
      END
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      SUBROUTINE dqromo(func,ll,ul,ss,A,B,m,
     &choose)
      IMPLICIT NONE

      INTENT(IN):: ll,ul
      INTENT(OUT):: ss
      INTENT(IN):: A,B,m

      INTEGER JMAX,JMAXP,K,KM
      INTEGER flag
      DOUBLE PRECISION ll,ul,func,ss,EPS
      DOUBLE PRECISION ssold,dssold
      EXTERNAL func,choose
      PARAMETER (EPS=1.d-9,JMAX=20,JMAXP=JMAX+1)
C      PARAMETER (EPS=1.d-10,JMAX=20,JMAXP=JMAX+1,K=5,KM=K-1)
CU    USES polint
      INTEGER j
      DOUBLE PRECISION dss,h(JMAXP),s(JMAXP)

      DOUBLE PRECISION A,B,m

c      write(*,*)"in DQROMO"
C     initial values
      K=5
```

```
      KM=K-1

      do j=1,JMAXP
        h(j)=0.0d0
        s(j)=0.0d0
      enddo

      h(1)=1.0d0
      flag=0
      do 11 j=1,JMAX
        call choose(func,ll,ul,s(j),j,A,B,m)

        if( (K.lt.10).and.(j.ge.K).and.(s(j).eq.0.0d0) )then
          K=K+1
          KM=K-1
        endif

c       write(*,*)j,K,s(j)
        if (j.ge.K) then
          call polint(h(j-KM),s(j-KM),K,0.0d0,ss,dss)
c       write(*,*)ss,dss
c           if (dabs(dss).le.EPS*dabs(ss)) return
          if(flag.eq.1)then
            if( (ss.eq.ssold).and.(dss.eq.dssold) )return
            if(ssold.ne.0.0d0)then
                if( dabs( (ss-ssold)/ssold ).lt.1.0d-5 )return
            endif
          endif
          ssold=ss
          dssold=dss
          flag=1
        endif
        s(j+1)=s(j)
        h(j+1)=h(j)/9.0d0
11    continue
C       pause 'too many steps in qromo'
      write(*,*)'no convergence in qromo'
      write(*,*)'solution in qromo is not accurate'
      do j=1,JMAXP
        h(j)=0.0d0
        s(j)=0.0d0
      enddo
      return
      END
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      SUBROUTINE polint(xa,ya,n,x,y,dy)
      IMPLICIT NONE

      INTENT(IN):: xa,ya,n,x
      INTENT(OUT):: y,dy

      INTEGER n,NMAX
      DOUBLE PRECISION dy,x,y,xa(n),ya(n)
      PARAMETER (NMAX=100)
      INTEGER i,m,ns
      DOUBLE PRECISION den,dif,dift,ho,hp,w,c(NMAX),d(NMAX)
```

```
      do i=1,NMAX
        c(NMAX)=0.0d0
        d(NMAX)=0.0d0
      enddo

      ns=1
      dif=dabs(x-xa(1))
      do 11 i=1,n
        dift=dabs(x-xa(i))
        if (dift.lt.dif) then
          ns=i
          dif=dift
        endif
        c(i)=ya(i)
        d(i)=ya(i)
11    continue
      y=ya(ns)
      ns=ns-1
      do 13 m=1,n-1
        do 12 i=1,n-m
          ho=xa(i)-x
          hp=xa(i+m)-x
          w=c(i+1)-d(i)

          den=ho-hp
          if(den.eq.0.0d0)pause 'failure in polint'
          den=w/den
          d(i)=hp*den
          c(i)=ho*den
12      continue
        if (2*ns.lt.n-m)then
          dy=c(ns+1)
        else
          dy=d(ns)
          ns=ns-1
        endif
        y=y+dy
13    continue
      do i=1,NMAX
        c(NMAX)=0.0d0
        d(NMAX)=0.0d0
      enddo
      return
      END
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      SUBROUTINE dmidrcp(funk,ll,ul,s,n,AA,BB,m)
      IMPLICIT NONE

      INTENT(IN):: ll,ul
      INTENT(OUT):: s
      INTENT(IN):: n,AA,BB,m

      INTEGER n
      DOUBLE PRECISION ul,ll,s,funk

      EXTERNAL funk
      INTEGER it,j
```

```
      DOUBLE PRECISION ddel,del,sum,tnm,x,func,a,b,c,d
      DOUBLE PRECISION dum
      DOUBLE PRECISION AA,BB,m
      PARAMETER(c=1.0d+0,d=5.0d-2)
C     x -> 1/(1+cx)**d
         func(AA,BB,m,x)
     &=  funk(AA,BB,m,(1.0d0/(x**(1.0d0/d)) - 1.0d0)/c)
     & / ( c*d*x**(1.0d0/d+1.0d0) )

      b=1.0d0/( (1.0d0+c*ll)**d )
      a=0.0d0

      dum=ul

      if (n.eq.1) then
        s= (b-a)
     &    *func(AA,BB,m,0.5d0*(a+b))
      else
        it=3**(n-2)
        tnm=dble(it)
        del=(b-a)/(3.0d0*tnm)
        ddel=del+del
        x=ll+0.5d0*del
        sum=0.0d0
        do 11 j=1,it
          sum=sum+func(AA,BB,m,x)
          x=x+ddel
          sum=sum+func(AA,BB,m,x)
          x=x+del
11      continue
        s=(s+(b-a)*sum/tnm)/3.0d0
      endif
      return
      END
cccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      SUBROUTINE dmidexp(funk,ll,ul,s,n,AA,BB,m)
      IMPLICIT NONE

      INTENT(IN):: ll,ul
      INTENT(OUT):: s
      INTENT(IN):: n,AA,BB,m

      INTEGER n
      DOUBLE PRECISION ul,ll,s,funk

      EXTERNAL funk
      INTEGER it,j
      DOUBLE PRECISION ddel,del,sum,tnm,x,func,a,b
      DOUBLE PRECISION dum
      DOUBLE PRECISION AA,BB,m

C     x -> exp(-x)
       func(AA,BB,m,x)
     &=funk(AA,BB,m,-dlog(x))/x

      dum=ul
```

```
      b=dexp(-ll)
      a=0.0d0

      if (n.eq.1) then
        s= (b-a)
     &    *func(AA,BB,m,0.5d0*(a+b))
      else
        it=3**(n-2)
        tnm=dble(it)
        del=(b-a)/(3.0d0*tnm)
        ddel=del+del
        x=ll+0.5d0*del
        sum=0.0d0
        do 11 j=1,it
          sum=sum+func(AA,BB,m,x)
          x=x+ddel
          sum=sum+func(AA,BB,m,x)
          x=x+del
11      continue
        s=(s+(b-a)*sum/tnm)/3.0d0
      endif
      return
      END
```

Fortran function FUNCT

```
      DOUBLE PRECISION FUNCTION funct(A,B,m,z)
      IMPLICIT NONE

      INTENT(IN):: A,B,m,z

      DOUBLE PRECISION A,B,m,z

      funct= 1.0d0/(1.0d0+A*A*z)
     &      *dexp( -( (A*z+m)*(A*z+m) + B*B )/(1.0d0+A*A*z) )

      return
      END
```

Fortran input file INPUT.DAT

```
input file for xTFXR
target:     thermal conductivity (W/m-k), density (kg/m^3), specific
heat (J/K-kg)
83.2d0,                            925.3d0,          1421.5d0
maximum allowable temperature (K)
500.0d0
initial temperature (K)
381.0d0
film thickness (m)
1.08d-4
film velocity (m/s)
50.0d0
location relative to beam spot (m)
```

```
0.0d0
beam power (W)
1902.0d0
beam spot radius (m)
0.5d-3
area of calculation H, W (m)
30.0d-3,    10.0d-3
number of grid H, W (-)
50,   50
```

Argonne
NATIONAL LABORATORY

Nuclear Engineering Division / Physics Division
Argonne National Laboratory
9700 South Cass Avenue, Bldg. 208
Argonne, IL 60439-4842

www.anl.gov

UChicago ▶
Argonne LLC

A U.S. Department of Energy laboratory
managed by UChicago Argonne, LLC