191
3-4-81
R2436
(1)
Lh. 2359

LBL-12125
UC-21

Lawrence Berkeley Laboratory
UNIVERSITY OF CALIFORNIA

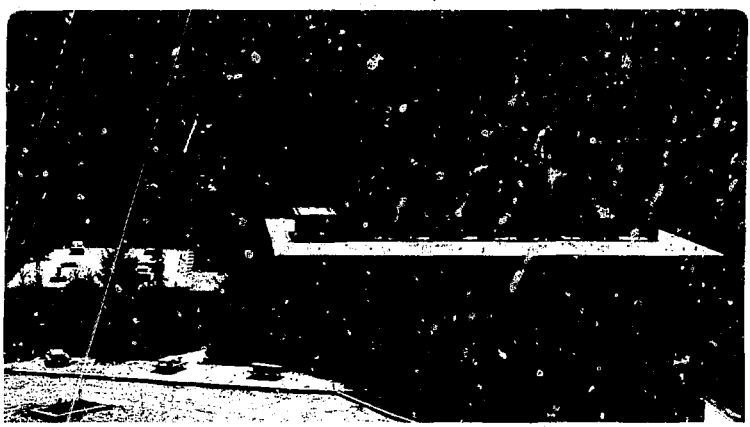
Materials & Molecular Research Division

NUMERICAL SIMULATION OF TRANSIENT, INCONGRUENT VAPORIZATION INDUCED BY HIGH POWER LASER

C.H. Tsai

MASTER

January 1981



Prepared for the U.S. Department of Energy under Contract W-7405-ENG-48

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

LBL-12125

NUMERICAL SIMULATION OF TRANSIENT, INCONGRUENT VAPORIZATION INDUCED BY HIGH POWER LASER

C.H. Tsai

Materials and Molecular Research
Division of the Lawrence Berkeley
Laboratory and the
Department of Nuclear Engineering
University of California Berkeley
Berkeley, California 94720

DISCLAIMER

This work was supported by the Director, Office of Energy Research, Office of Basic Energy Sciences, Materials Sciences Division of the U.S. Department of Energy under contract # W-7405-ENG-48.

January 1981

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

CONTENTS

	page
1. Introduction	1
2. Mathematical Model	4
2.1 Mass Balance Equation	4
2.2 Energy Balance Equation	5
2.3 Diffusive Flux and Heat Flux	7
2.4 Initial and Boundary Conditions	7
2.5 Diffusion and Heat Conduction Equations	10
2.6 Approximate Conservation Equations	11
2.7 Non-dimensionalization of Conservation Equations	13
Appendix A: Numerical Solution	17
Appendix B: Computer Programs and Sample Analyses	36
Appendix C: Thermodynamic Data and Material Properties of UO_2	84
References	89

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Numerical Simulation of Transient, Incongruent Vaporization Induced by High Power Laser

C. H. Tsai

U C Berkeley - Lawrence Berkeley Lab

1. INTRODUCTION

When laser radiation is incident on an absorbing material, all the effects such as phase changes (melting or vaporization), thermal stress, thermal radiation and shock waves from the surface and mass transport in the bulk are associated with the surface temperatures, temperature gradients and composition gradients during the irradiation transient. The laser energy absorption and conduction is considered as a heat transfer process because the laser pulsing time and the characteristic heating time are far longer than the time for electronic relaxation and transfer of energy to the lattice phonos ($\sim 10^{-13}$ sec). In addition to the heat conduction problem, the composition redistribution during the transient due to the incongruent evaporation has to be considered when the target material is a compound.

Although laser surface heating (LSH) technique has been developed for years[1-4] for thermal physical property investigation of liquid phase urania, there has been no any attempt to calculate the ~msec thermal evolution of a solid urania subject to intensive laser impingement. Ohse et al[2] relied mainly on the pyrometric measurement of surface temperature and no surface oxygen depletion has been considered. The only effort they did[5] for temperature profile calculation was the

adoption of Dabby and Paek[6] model assuming a prior known steady state surface temperature, called "evaporation temperature" for the purpose of investigating the influence of the sub-surface temperature profile on the pyrometric measurement of surface temperatures (i.e. the difference in emitted thermal radiation intensity between a uniformly heated sample and the one with non-uniform temperature profile). Bober et al relied on firstly the gasdynamic model interpretation[3], then the pyrometric measurement[7]. The surface oxygen depletion and oxygen-to-uranium ratio profile was calculated based on the "forced congruent evaporation" model by Breitung[8] assuming a prior known steady state, uniform temperature and eventually an asymtotic steady state oxygen-to-uranium ratio profile. The basic requirement for the forced congruent evaporation model is the steady state temperature, and it is not applicable to ~msec transient heating condition because the characteristic time for oxygen diffusion process is in the same order as the temperature evolution and the asymtotic stationary "congruent" evaporation is not obtainable.

In this study, a mathematical model and numerical calculations developed to solve the heat and mass transfer problems specifically for uranum oxide subject to laser irradiaton. It can easly be modified for other heat sources or/and other materials. In the uranium-oxygen system, oxygen is the preferentially vaporizing component, and as a result of the finite mobility of oxygen in the solid, an oxygen deficiency is set up near the surface. Because of the bivariant behavior of uranium oxide, the heat transfer problem and the oxygen diffusion problem are coupled and a numerical method of simultaneously solving the two boundary value problems is studied. The temperature dependence of the

thermal properties and oxygen diffusivity, as well as the highly ablative effect on the surface, leads to considerable non-linearities in both the governing differential equations and the boundary conditions. Based on the earlier work done in this laboratory by Olstad and Olander on Iron[9] and on Zirconium hydride[10], the generality of the problem is expanded and the efficiency of the numerical scheme is improved. The finite difference method, along with some advanced numerical techniques, is found to be an efficient way to solve this problem.

2. MATHEMATICAL MODEL

Consider a semi-infinite slab occupying the region $z \geq 0$, and the slab is irradiated by a pulse of laser beam propagating in z-direction. One-dimensional heat conduction, one-dimensional ablation (no radial liquid movement) with phase change and one-dimensional oxygen diffusion are assumed. The phase change, or fusion, is considered implicitly in the heat capacity term and heat of vaporization term (see Appendix C). Although the laser focal spot size is usually very small, it is still orders of magnitude larger than the characteristic lengths of heat conduction and diffusion in most materials, especially refractory materials like uranium oxide. Thermodynamic equilibrium during transient vaporization due to laser heating is assumed because the decrease in surface stoichiometry can only be quantified based on this assumption. Either penetration of laser radiation (volumetric heating) or the absorbing surface (surface heating) is allowed. This option is designed not only for the different optical properties of the materials subject to laser radiation, but also for the application to other volumetric heating techniques, such as exploding wires, electron beam bombardment heating and neutron pulse heating, which are also proposed for the acquisition of thermochemical data measrements. The vapor plume created by vaporization is assumed to be transparent to the incident laser radiation.

2.1 Mass Balance Equation

Due to the low diffusion coefficient of uranium ions in UO_2

compared with the diffusion coefficient of oxygen ions, the uranium ions are considered to form an immobile lattice through which oxygen can migrate.

The mass balance equation of oxygen in the solid phase is:

$$\frac{\partial C_o}{\partial t} = -\frac{\partial}{\partial z} j_o^d \tag{1}$$

where C_o = mass concentration of oxygen atom in the solid, g/cm^3

j_o^d = oxygen diffusive flux in the solid, g/cm^2-sec.

To account for the possibility of significant ablation from the interphase boundary, we make the following coordinate transformation:

$$\begin{cases} t' = t \\ x = z - vt \end{cases} \tag{2}$$

where z is the coordinate from the original surface, v is the ablation (surface recession) velocity.

After the coordinate transformation, the balance equation of oxygen becomes:

$$\frac{\partial C_o}{\partial t} - v\frac{\partial C_o}{\partial x} = -\frac{\partial}{\partial x} j_o^d \tag{3}$$

Neglecting the prime on t' and rearranging the equation, we have:

$$\frac{\partial C_o}{\partial t} = -\frac{\partial}{\partial x} j_o^d + v\frac{\partial C_o}{\partial x} \tag{4}$$

2.2 Energy Balance Equation

The general energy balance equation for the solid, assuming that oxygen is the only mobile component is:

$$\rho\frac{\partial Q}{\partial t} = -\frac{\partial q}{\partial z} - \frac{\partial}{\partial z}(j_o^d \bar{h}_o) + Q_v \tag{5}$$

where ρ = mass density, g/cm^3

$\hat{U}$ = specific internal energy, J/g

q = heat flux, W/cm^2

$\bar{h}_o$ = partial specific enthalpy of oxygen, J/g

Q_v = volumetric heat source, W/cm^3.

For the heat conduction through a condensed, incompressible medium with mobile oxygen and immobile uranium, the specific internal energy $\hat{U}$ is a function of temperature T and concentration C_o only. Therefore,

$$\frac{\partial \hat{U}}{\partial t} = \left(\frac{\partial \hat{U}}{\partial C_o}\right)_T \left(\frac{\partial C_o}{\partial t}\right) + \left(\frac{\partial \hat{U}}{\partial T}\right)_{C_o} \left(\frac{\partial T}{\partial t}\right) \tag{6}$$

Assuming no mechanical work is done,

$$\hat{U} = \hat{H} = (\bar{h}_o C_o + \bar{h}_U C_U)/\rho \tag{7}$$

where $\bar{h}_U$ = partial specific enthalpy of uranium, J/g

C_U = mass concentration of uranium atom in the solid, g/cm^3

Since $\bar{h}_U$ and C_U are assumed independent of C_o (based on the assumption that uranium atoms are immobile),

$$\left(\frac{\partial \hat{U}}{\partial C_o}\right)_T = \bar{h}_o/\rho \tag{8}$$

Similarly,

$$\left(\frac{\partial \hat{U}}{\partial T}\right)_{C_o} = \left(\frac{\partial \hat{H}}{\partial T}\right)_{C_o} = C_p \tag{9}$$

where C_p is the constant pressure specific heat in J/g-K. Therefore, equation (6) becomes:

$$\frac{\partial \hat{U}}{\partial t} = \frac{\bar{h}_o}{\rho}\frac{\partial C_o}{\partial t} + C_p \frac{\partial T}{\partial t} \tag{10}$$

Substituting eq. (10) into eq. (5) and using the coordinate transformation of eq. (2), we have:

$$\rho C_p \frac{\partial T}{\partial t} - \rho C_p v \frac{\partial T}{\partial x} + \bar{h}_o \frac{\partial C_o}{\partial t} - \bar{h}_o v \frac{\partial C_o}{\partial x} = - \frac{\partial q}{\partial x} - \frac{\partial}{\partial x}(j_o^d \bar{h}_o) + Q_v \tag{11}$$

Multiplying eq. (4) by $\bar{h}_o$ and substituting the result into eq. (11) yields:

$$\rho C_p \frac{\partial T}{\partial t} - \rho C_p v \frac{\partial T}{\partial x} = - \frac{\partial q}{\partial x} - j_o^d \frac{\partial \bar{h}_o}{\partial x} + Q_v \tag{12}$$

2.3 Oxygen Diffusive Flux and Heat Flux

Taking into account the Soret and Dufour effects, the mass and heat fluxes are given by:

$$j_o^d = - D_o \frac{\partial C_o}{\partial x} - \frac{D_o Q^* C_o \partial T}{RT^2 \ \partial x} \tag{13}$$

$$q = - \frac{D_o Q^* \partial C_o}{M_o \ \partial x} - k \frac{\partial T}{\partial x} \tag{14}$$

where D_o = diffusion coefficient of oxygen in uranium oxide, cm^2/sec

Q^* = heat of transport of oxygen in uranium oxide, J/mole

R = gas constant = 8.314 J/K-mole

M_o = atomic weight of oxygen = 16 g/g-atom

k = thermal conductivity, W/cm-K

2.4 Initial and Boundary Conditions

The two balance equations (4) and (12) are coupled through the two fluxes j_o^d and q. In order to solve those two partial differential equations, we need two initial conditions and four boundary conditions.

The initial conditions are:

$$T(x,0) = T_o \quad \text{and} \quad C_o(x,0) = C_o^o \qquad @\ t=0 \tag{15}$$

where T_o is the initial temperature before laser impingement and C_o^o is the initial mass concentration of oxygen.

The boundary conditions are:

(i) at the moving interphase boundary, x = 0:

As a result of preferential vaporization and finite supply rate from the bulk, oxygen is depleted at the surface and a concentration gradient is set up across the inside of the oxide which drives a stream of oxygen atoms towards the surface. The diffusion flux at the surface is given in eq. (13) except all the quantities are evaluated at x=0.

The mass balance for oxygen atoms at the interphase boundary gives:

$$j_o^d = j_o^g + vC_o^s \qquad @\ x=0 \tag{16}$$

where C_o^s is the mass concentration of oxygen atom of the solid at the surface, j_o^g is the total mass flux of oxygen in the gas phase,

$$j_o^g = -\ (3\Phi_{UO_3}+2\Phi_{UO_2}+\Phi_{JO}+\Phi_O+2\Phi_{O_2})M_o \tag{17}$$

and v is the surface recession velocity, which is the ratio of the evaporation mass flux of uranium-bearing species j_U^g and the mass concentration of uranium atom in the solid C_U,

$$v = -\ j_U^g/C_U = (\Phi_{UO_3}+\Phi_{JO_2}+\Phi_{JO}+\Phi_U)M_U/C_U \tag{18}$$

The sign convention is that a flux (mass or heat) is positive if it is in the positive x direction. (Remember that the fluxes are relative to the moving boundary.)

The Langmiur vaporization rate of species i, Φ_i is:

$$\Phi_i = \frac{P_i}{\sqrt{2\pi RT_s}\sqrt{M_i}} \quad \text{mole/cm}^2\text{-sec} \tag{19}$$

where T_s is the surface temperature and P_i is the equilibrium pressure of species i over the solid at the surface composition and temperature. The vacuum vaporization formula of eq. (19) is strictly valid only if the vapor plume in front of the solid is collisionless. However, many theoretical investigations[11-13] have shown that even in a collision-dominated vapor plume, the net vaporization rate is at least 82% of that given by eq. (19) (i.e. the fraction of the forward vaporization flux backscattered to the surface is <18%).

Combining eqs. (13) and (16), we have:

$$-D_o \frac{\partial C_o}{\partial x}\Big|_{x=0} - \frac{D_o Q^* C_o}{RT^2}\frac{\partial T}{\partial x}\Big|_{x=0} = j_o^g + vC_o^s \tag{20}$$

The heat flux in the solid at the interphase boundary is balanced by the heat loss carried by the vaporizing species (ablation), the radiation heat loss and the heat flux input from the laser (for surface absorption only), i.e.,

$$(q)_{x=0} = -\sum_i M_i \Phi_i \Delta H_i^v - \epsilon_t \sigma (T_s^4 - T_b^4) + Q_s \tag{21}$$

where ΔH_i^v = heat of vaorization of species i, J/g

T_b = the ambient temperature (usually room temperature), K

ϵ_t = total hemispherical optical emissivity

σ = Stefan-Boltzmann constant

= 5.6686×10^{-12} J/cm^2-sec-K^4

and Q_s = surface heat source, W/cm^2.

If the laser radiation is considered to penetrate into the solid, the volumetric heat source term Q_v in the energy equation is nonzero and the surface heat source term Q_s in the boundary condition is zero. If the laser irradiation is considered as a surface source, Q_s is nonzero and Q_v is zero.

Combining eqs. (14) and (21), we have:

$$-\frac{D_o Q^* \partial C_o}{M_o \partial x}\Big|_{x=0} - k\frac{\partial T}{\partial x}\Big|_{x=0} = -\sum_i M_i \Phi_i \Delta H_i^v - \epsilon_t \sigma (T_s^4 - T_b^4) + Q_s \qquad (22)$$

The boundary condition eqs. (20) and (22) can be separated to solve for $\partial C_o/\partial x$ and $\partial T/\partial x$ explicitly:

$$\left(\frac{\partial T}{\partial x}\right)_{x=0} = \frac{\sum_i M_i \Phi_i \Delta H_i^v + \epsilon_t \sigma (T_s^4 - T_b^4) - Q_s + \frac{Q^*}{M_o}(j_o^g + v C_o^s)}{k - \frac{D_o (Q^*)^2 C_o^s}{M_o R T_s^2}} \qquad (23)$$

$$(i = UO_3, UO_2, UO, U, O, O_2)$$

$$\left(\frac{\partial C_o}{\partial x}\right)_{x=0} = \frac{\frac{k}{D_o}(j_o^g + v C_o^s) - \frac{Q^* C_o^s}{R T_s^2}\left[\sum_i M_i \Phi_i \Delta H_i^v + \epsilon_t \sigma (T_s^4 - T_b^4) - Q_s\right]}{k - \frac{D_o (Q^*)^2 C_o^s}{M_o R T_s^2}} \qquad (24)$$

(ii) at the far end of the solid, $x = \infty$:

$$T(\infty, t) = T_o \quad \text{and} \quad C_o(\infty, t) = C_o^o \qquad (25)$$

2.5 Oxygen Diffusion and Heat Conduction Equation

Substituting the fluxes, eqs. (13) and (14), into the partial differential equations (4) and (12), we obtain the two boundary value problems to be solved for T and C_o:

$$\frac{\partial C_o}{\partial t} = \frac{\partial}{\partial x}\left(D_o\frac{\partial C_o}{\partial x} + \frac{D_o Q^* C_o \partial T}{RT^2\ \partial x}\right) + v\frac{\partial C_o}{\partial x} \tag{26}$$

$$\frac{\partial T}{\partial t} = \frac{1}{\rho C_p}\frac{\partial}{\partial x}\left(\frac{D_o Q^* \partial C_o}{M_o\ \partial x} + k\frac{\partial T}{\partial x}\right) + \left(\frac{D_o}{\rho C_p}\frac{\partial C_o}{\partial x} + \frac{D_o Q^* C_o \partial T}{\rho C_p RT_2 \partial x}\right)\frac{\partial \bar{h}_o}{\partial x} + v\frac{\partial T}{\partial x} + \frac{Q_v}{\rho C_p} \tag{27}$$

$$\text{I.C.: } T(x,0) = T_o \quad \text{and} \quad C_o(x,0) = C_o^o \quad @\ t=0 \tag{28}$$

$$\text{B.C.: } \left(\frac{\partial T}{\partial x}\right)_{x=0} = \frac{\sum_i M_i \Phi_i \Delta H_i^v + \epsilon_t \sigma(T_s^4 - T_b^4) - Q_s + \frac{Q^*}{M_o}(j_o^g + vC_o^s)}{k - \frac{D_o (Q^*)^2 C_o^s}{M_o R T_s^2}} \tag{29}$$

$$\left(\frac{\partial C_o}{\partial x}\right)_{x=0} = \frac{\frac{k}{D_o}(j_o^g + vC_o^s) - \frac{Q^* C_o^s}{RT_s^2}\left[\sum_i M_i \Phi_i \Delta H_i^v + \epsilon_t \sigma(T_s^4 - T_b^4) - Q_s\right]}{k - \frac{D_o (Q^*)^2 C_o^s}{M_o R T_s^2}} \tag{30}$$

$$T(\infty,t) = T_o \quad \text{and} \quad C_o(\infty,t) = C_o^o \quad @\ x=\infty \tag{31}$$

2.6 Approximate Conservation Equations

In this study, an approxima e solution is treated by assuming that Q^* and $\partial \bar{h}_o/\partial x$ are zero; in other words, the thermal diffusion (Soret and Dufour effects) of oxygen is neglected and the oxygen diffusion does not transport any energy. Also, the ablation heat term $\sum_i M_i \Phi_i \Delta H_i^v$ is approximated by $j_{tot}\Delta H_{vap}$, where j_{tot} is the total vaporization flux $(=\sum_i M_i \Phi_i)$ and ΔH_{vap} is the heat of vaporization from P_{tot}-T relation (P_{tot} is the total vapor pressure of UO_2).

The boundary value problems become:

$$\frac{\partial C_o}{\partial t} = \frac{\partial}{\partial x}\left(D_o\frac{\partial C_o}{\partial x}\right) + v\frac{\partial C_o}{\partial x} \tag{32}$$

$$\frac{\partial T}{\partial t} = \frac{1}{\rho C_p}\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + v\frac{\partial T}{\partial x} + \frac{Q_v}{\rho C_p} \tag{33}$$

I.C.: $T(x,0) = T_o$ and $C_o(x,0) = C_o^o$ (34)

B.C.: $\left(\frac{\partial T}{\partial x}\right)_{x=0} = \frac{1}{k}\left[j_{tot}\Delta H_{vap} + \epsilon_t\sigma(T_s^4 - T_b^4) - Q_s\right]$ (35)

$$\left(\frac{\partial C_o}{\partial x}\right)_{x=0} = \frac{1}{D_o}\left(-j_o^g - vC_o^s\right) \tag{36}$$

$$T(\infty,t) = T_o \quad \text{and} \quad C_o(\infty,t) = C_o^o \tag{37}$$

The validity of these approximations has been tested and found to be acceptable.

In reality, it is the oxygen-to-uranium ratio, O/U, which we are interested in, rather than the oxygen concentration C_o. Let us define the oxygen-to-uranium ratio $r = \frac{C_o/M_o}{C_U/M_U}$. Since the concentration of uranium C_U is assumed constant, we can obtain equations for r simply multiplying the equations involving C_o by $\frac{M_U}{M_o C_U}$:

$$\frac{\partial r}{\partial t} = \frac{\partial}{\partial x}\left(D_o\frac{\partial r}{\partial x}\right) + v\frac{\partial r}{\partial x} \tag{38}$$

$$\frac{\partial T}{\partial t} = \frac{1}{\rho C_p}\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + v\frac{\partial T}{\partial x} + \frac{Q_v}{\rho C_p} \tag{39}$$

I.C.: $T(x,0) = T_o$ and $r(x,0) = r_o$ (40)

B.C.: $\left(\frac{\partial T}{\partial x}\right)_{x=0} = \frac{1}{k_s}\left[j_{tot}\Delta H_{vap} + \epsilon_t\sigma(T_s^4 - T_b^4) - Q_s\right]$ (41)

$$\left(\frac{\partial r}{\partial x}\right)_{x=0} = -\frac{j_o^g M_U}{D_o^s\left(C_U M_o + v r_s\right)} \tag{42}$$

$$T(\infty,t) = T_o \quad \text{and} \quad r(\infty,t) = r_o \tag{43}$$

where $_s$ and r_o are the oxygen-to-uranium ratios of the solid at the surface and in the bulk, respectively.

2.7 Non-dimensionalization of Conservation Equations

Because of the non-linearity of the governing differential equations (38) and (39) and the boundary condition equations (41) and (42), the analytical solutions are not possible. Before the numerical methods are applied, eqs. (38) - (43) have to be non-dimensionalized, because the numerical values in the computations are normally assumed dimensionless and it is obvious that it is meaningless to carry on the operations among the numerical values of different units.

By introducing appropriate characteristic composition $\tilde{r}$, characteristic temperature $\tilde{T}$, characteristic time $\tilde{t}$ and characteristic length $\tilde{x}$, we can define the dimensionless quantities:

$$Y = \frac{r}{\tilde{r}} \; ; \quad \theta = \frac{T}{\tilde{T}} \; ; \quad \tau = \frac{t}{\tilde{t}} \; ; \quad x = \frac{x}{\tilde{x}} \tag{44}$$

In the problem formulated in the beginning of the section, especially for UO_2, the characteristic quantities were taken as:

$$\tilde{r} = r_o \; ; \quad \tilde{T} = T_o \; ; \quad \tilde{t} = t_{pul} \; ; \quad \tilde{x} = \left(\frac{k t_{pul}}{\rho C_p}\right)_m^{\frac{1}{2}} \tag{45}$$

where r_o = initial oxygen-to-uranium ratio

T_o = initial temperature, K

t_{pul} = effective laser power pulse width

$= \int P(t)dt/P_{max} = E_{tot}/P_{max}$

E_{tot} = total laser energy, Joules

P_{max} = maximum laser power, Watts

and "m" means that the quantities are evaluated at the melting temperature.

After the introduction of dimensionless quantities and rearrangement, eqs. (38) - (43) become:

$$\frac{\partial^2\theta}{\partial x^2} = A_1\frac{\partial\theta}{\partial\tau} - A_2\frac{\partial\theta}{\partial x} - A_3\left(\frac{\partial\theta}{\partial x}\right)^2 - A_4 \tag{46}$$

$$\frac{\partial^2 Y}{\partial x^2} = B_1\frac{\partial Y}{\partial\tau} - B_2\frac{\partial Y}{\partial x} - B_3\left(\frac{\partial Y}{\partial x}\right)\left(\frac{\partial\theta}{\partial x}\right) \tag{47}$$

I.C.: $\theta(x,0) = 1$ and $Y(x,0) = 1$ (48)

B.C.: $\left(\frac{\partial\theta}{\partial x}\right)_{x=0} = \eta_T$ (49)

$\left(\frac{\partial Y}{\partial x}\right)_{x=0} = \eta_r$ (50)

$\theta(\infty,\tau) = 1$ and $Y(\infty,\tau) = 1$ (51)

where $A_1 = \left(\frac{k}{\rho C_p}\right)_m / \left(\frac{k}{\rho C_p}\right)$

$$A_2 = v\sqrt{t_{pul}}\left(\frac{k}{\rho C_p}\right)_m^{\frac{1}{2}} / \left(\frac{k}{\rho C_p}\right)$$

$$A_3 = \frac{T_o}{k}\frac{dk}{dT}$$

$$A_4 = \left(\frac{k}{\rho C_p}\right)_m \frac{t_{pul}}{kT_o} Q_v$$

$$B_1 = \left(\frac{k}{\rho C_p}\right)_m / D_o$$

$$B_2 = v\sqrt{t_{pul}}\left(\frac{k}{\rho C_p}\right)_m^{\frac{1}{2}} / D_o$$

$$B_3 = \frac{T_o}{D_o}\frac{dD_o}{dT}$$

η_T = dimensionless surface temperature gradient

$$= \frac{\sqrt{t_{pul}}\left(\frac{k}{\rho C_p}\right)_m^{\frac{1}{2}}}{k_s} T_o\left[j_{tot}\Delta H_{vap} + \epsilon_t\sigma(T_s^4 - T_b^4) - Q_s\right]$$

and η_r = dimensionless surface composition gradient

$$= \frac{-\sqrt{t_{pul}}\left(\frac{k}{\rho C_p}\right)_m^{\frac{1}{2}}}{D_o^s r_o}\left[\frac{j_o^g}{C_U} + v r_s\right]$$

The standard classical reference on the heat transfer analysis is the book by Carslaw and Jaeger[14], in which a number of exact solutions are given for infinite and semi-infinite solids, solid bounded by two parallel planes, and the heat flow in rectangular, cylindrical and spherical geometries, that are subjected to a variety of initial and boundary conditions. In almost all of the problems for which an exact solution is possible, the thermal properties k, ρ and C_p are taken to be constants, independent of both temperature and position. And also the initial and boundary conditions are in linear form. However, in our case, the considerable non-linearities resulting from the temperature dependent thermal properties, the convectivelike term from coordinate transformation, and the nonlinear boundary conditions resulting from the strong temperature dependent ablative and radiation heat loss terms , make exact solution impossible.

The system of eqs. (46) - (51) is solved numerically by the method described in Appendix A. The most commonly used numerical method for solving this kind of problem is the finite difference method, which has been developed for decades and widely used[15-19]. Although it has been found that the finite difference method is not very efficient for highly non-linear problems, it turns out to work quite well with smoothly-varying heat source which drives the transient and with the help of predictor-corrector scheme for constructing a good initial guess for the iteration and the varying time and space increment scheme. A sample calculation for UO_2 is given in Appendix B. The material properties of UO_2 required in the analysis are given in Appendix C and they are permitted to vary with temperature but not with oxygen-to-uranium ratio. The overall effect of the vaporization process is to make the surface of

the urania hypostoichiometric. Although the diffusivity of oxygen in solid UO_{2-x} has not yet been measured, it is almost certainly composition-dependent and larger than that in UO_2. However, the surface is melted very shortly at the initiation of a high energy laser pulse, and the diffusivity of oxygen in <u>liquid</u> UO_{2-x} is the important quantity. Its value is completely unknown but it is doubtful that concentration effects are as important as temperature effects in the liquid range.

APPENDIX A: NUMERICAL SOLUTION OF EQS. (46) - (51)

Reconstructing eqs. (46) - (51):

$$\frac{\partial^2\theta}{\partial x^2} = A_1\frac{\partial\theta}{\partial\tau} - A_2\frac{\partial\theta}{\partial x} - A_3\left(\frac{\partial\theta}{\partial x}\right)^2 - A_4 \tag{A-1}$$

$$\frac{\partial^2 Y}{\partial x^2} = B_1\frac{\partial Y}{\partial\tau} - B_2\frac{\partial Y}{\partial x} - B_3\left(\frac{\partial Y}{\partial x}\right)\left(\frac{\partial\theta}{\partial x}\right) \tag{A-2}$$

$$\theta(x,0) = 1 \quad \text{and} \quad Y(x,0) = 1 \tag{A-3}$$

$$\left(\frac{\partial\theta}{\partial x}\right)_{x=0} = \eta_T \tag{A-4}$$

$$\left(\frac{\partial Y}{\partial x}\right)_{x=0} = \eta_r \tag{A-5}$$

$$\theta(\infty,\tau) = 1 \quad \text{and} \quad Y(\infty,\tau) = 1 \tag{A-6}$$

where $A_1 = \left(\frac{k}{\rho C_p}\right)_m / \left(\frac{k}{\rho C_p}\right)$

$$A_2 = v\sqrt{t_{pul}}\left(\frac{k}{\rho C_p}\right)_m^{1/2} / \left(\frac{k}{\rho C_p}\right)$$

$$A_3 = \frac{T_o}{k}\frac{dk}{dT}$$

$$A_4 = \left(\frac{k}{\rho C_p}\right)_m \frac{t_{pul}}{k T_o} Q_v$$

$$B_1 = \left(\frac{k}{\rho C_p}\right)_m / D_o$$

$$B_2 = v\sqrt{t_{pul}}\left(\frac{k}{\rho C_p}\right)_m^{1/2} / D_o$$

$$B_3 = \frac{T_o}{D_o}\frac{dD_o}{dT}$$

η_T = dimensionless surface temperature gradient

$$= \frac{\sqrt{t_{pul}}\left(\frac{k}{\rho C_p}\right)_m^{1/2}}{k_s} T_o\left[\Phi_{tot}\Delta H_{vap} + \epsilon_t\sigma(T_s^4 - T_b^4) - Q_s\right]$$

and η_r = dimensionless surface composition gradient

$$= \frac{-\sqrt{t_{pul}}\left(\frac{k}{\rho C_p}\right)_m^{1/2}}{D_o^s r_o}\left[\frac{j_o^g}{C_U} + v r_s\right]$$

A.1 Crank-Nicolson Finite Difference Approximation

The Crank-Nicolson finite difference method is an implicit method in which truncated Taylor series expansions are used to approximate the derivatives in the governing differential equations and the space and time derivatives are then replaced by second order correct finite difference representations.

Let $W_{i,n}$ and $U_{i,n}$ denote the approximate solutions of the dimensionless temperature θ and the dimensionless oxygen-to-uranium ratio Y at space (dimensionless) grid x_i (called "grid" hereafter) and time (dimensionless) step τ_n (called "step" hereafter) respectively. Then the Crank-Nicolson method assumes that[17]:

$$\left.\frac{\partial^2 W}{\partial x^2}\right|_{i,n+\frac{1}{2}} = \frac{1}{2}\left(\frac{\partial^2 W_{i,n+1}}{\partial x^2}+\frac{\partial^2 W_{i,n}}{\partial x^2}\right) + O((\Delta\tau)^2)$$

$$= \frac{1}{2}\Delta_x^2(W_{i,n+1}+W_{i,n}) + O((\Delta x)^2+(\Delta\tau)^2) \qquad \text{(A-7)}$$

$$\left.\frac{\partial^2 U}{\partial x^2}\right|_{i,n+\frac{1}{2}} = \frac{1}{2}\left(\frac{\partial^2 U_{i,n+1}}{\partial x^2}+\frac{\partial^2 U_{i,n}}{\partial x^2}\right) + O((\Delta\tau)^2)$$

$$= \frac{1}{2}\Delta_x^2(U_{i,n+1}+U_{i,n}) + O((\Delta x)^2+(\Delta\tau)^2) \qquad \text{(A-8)}$$

$$\left.\frac{\partial W}{\partial \tau}\right|_{i,n+\frac{1}{2}} = \frac{W_{i,n+1}-W_{i,n}}{\Delta\tau} + O((\Delta\tau)^2) \qquad \text{(A-9)}$$

$$\left.\frac{\partial U}{\partial \tau}\right|_{i,n+\frac{1}{2}} = \frac{U_{i,n+1}-U_{i,n}}{\Delta\tau} + O((\Delta\tau)^2) \qquad \text{(A-10)}$$

$$\left.\frac{\partial W}{\partial x}\right|_{i,n+\frac{1}{2}} = \frac{1}{2}\Delta_x(W_{i,n+1}+W_{i,n}) + O((\Delta x)^2) \qquad \text{(A-11)}$$

$$\left.\frac{\partial U}{\partial x}\right|_{i,n+\frac{1}{2}} = \frac{1}{2}\Delta_x(U_{i,n+1}+U_{i,n}) + O((\Delta x)^2) \qquad \text{(A-12)}$$

where $\Delta_x^2 W_{i,n}$ = second order correct centered second

difference of $W_{i,n}$

$\Delta_x^2 U_{i,n}$ = second order correct centered second difference of $U_{i,n}$

$\Delta_x W_{i,n}$ = second order correct centered first difference of $W_{i,n}$

$\Delta_x U_{i,n}$ = second order correct centered first difference of $U_{i,n}$

By utilizing these finite difference operators, the differential equations (A-1) and (A-2) can be approximated by the difference equations: (for $i \geq 2$)

$$\frac{1}{2}\Delta_x^2(W_{i,n+1}+W_{i,n}) = A_1\frac{W_{i,n+1}-W_{i,n}}{\Delta\tau} -A_2[\frac{1}{2}\Delta_x(W_{i,n+1}+W_{i,n})]-A_3[\frac{1}{2}\Delta_x(W_{i,n+1}+W_{i,n})]^2-A_4 \tag{A-13}$$

$$\frac{1}{2}\Delta_x^2(U_{i,n+1}+U_{i,n}) = B_1\frac{U_{i,n+1}-U_{i,n}}{\Delta\tau} -B_2[\frac{1}{2}\Delta_x(U_{i,n+1}+U_{i,n})]-B_3[\frac{1}{2}\Delta_x(W_{i,n+1}+W_{i,n})] [\frac{1}{2}\Delta_x(U_{i,n+1}+U_{i,n})] \tag{A-14}$$

where the coefficients $A_1, A_2, A_3, A_4, B_1, B_2$ and B_3 are, in general, functions of $[x_i, \tau_{n+\frac{1}{2}}, \frac{1}{2}(W_{i,n+1}+W_{i,n})]$ and $\tau_{n+\frac{1}{2}}=\frac{1}{2}(\tau_{n+1}+\tau_n)$.

A.2 Second Order Correct Centered Finite Difference Operators

The finite difference method is an implict method in which a truncated Taylor series expansion is used to approximate the space and time derivatives. These finite difference representations are called finite difference operators; for example, if F_i is the finite difference approximate solution of $f(x)$ at x_i, $\Delta_x F_i$ is the first difference operator of F_i, $\Delta_x^2 F_i$ is the second difference operator of F_i, etc. There are

several ways of expressing the finite difference operators of different degree of order of accuracy, depending on the truncated error from Taylor series expansion. In this section, the second order correct centered finite difference operators are discussed.

For the case of constant increment of x, denoted Δx, the Taylor series expansion of F_{i+1} around F_i is:

$$F_{i+1} = F_i + \Delta x \dot{F}_i + \frac{(\Delta x)^2}{2} \ddot{F}_i + O((\Delta x)^3) \tag{A-15}$$

The Taylor series expansion of F_{i-1} around F_i is:

$$F_{i-1} = F_i - \Delta x \dot{F}_i + \frac{(\Delta x)^2}{2} \ddot{F}_i + O((\Delta x)^3) \tag{A-16}$$

Let's approximate the first derivative $\dot{F}_i$ by the finite difference operator $\Delta x F_i$. Then $\Delta x F_i$ can be solved from the equations:

$$\begin{cases} F_{i+1} = F_i + \Delta x(\Delta_x F_i) + \frac{(\Delta x)^2}{2} \ddot{F}_i & \text{(A-17)} \\ F_{i-1} = F_i - \Delta x(\Delta_x F_i) + \frac{(\Delta x)^2}{2} \ddot{F}_i & \text{(A-18)} \end{cases}$$

These two equations yields:

$$\Delta_x F_i = \frac{F_{i+1} - F_{i-1}}{2\Delta x} \tag{A-19}$$

which is order of $(\Delta x)^3/(\Delta x) = (\Delta x)^2$ correct.

Similarly, the second dfference operator $\Delta_x^2 F_i$ can be solved from the two equations:

$$\begin{cases} F_{i+1} = F_i + \Delta x \dot{F}_i + \frac{(\Delta x)^2}{2}(\Delta_x^2 F_i) + \frac{(\Delta x)^3}{6} \dddot{F}_i & \text{(A-20)} \\ F_{i-1} = F_i - \Delta x \dot{F}_i + \frac{(\Delta x)^2}{2}(\Delta_x^2 F_i) - \frac{(\Delta x)^3}{6} \dddot{F}_i & \text{(A-21)} \end{cases}$$

These two equations yields:

$$\underline{\Delta}_x^2 F_i = \frac{F_{i+1}-2F_i+F_{i-1}}{(\underline{\Delta}x)^2} \tag{A-22}$$

which is order of $(\underline{\Delta}x)^4/(\underline{\Delta}x)^2 = (\underline{\Delta}x)^2$ correct.

Rather than using a constant $\underline{\Delta}x$ as is usually the case, in this program $\underline{\Delta}x$ increases geometrically into the bulk of the solid according to the relation:

$$\underline{\Delta}x_i = \alpha\underline{\Delta}x_{i-1} = \alpha^{i-1}\underline{\Delta}x_1 \tag{A-23}$$

where α = a constant greater than 1, taken to be 1.035

$\underline{\Delta}x_i$ = grid increment following ith grid plane

$\underline{\Delta}x_1$ = the first grid increment at the surface

Then the second order correct differences are considerably more complicated; however, the way of deriving the finite difference operators from the truncated Taylor series expansion is exactly the same.

From the following third order truncated expansions:

$$\begin{cases} F_{i+1} = F_i + \underline{\Delta}x_1\alpha^{i-1}(\underline{\Delta}_x F_i) + \dfrac{(\underline{\Delta}x_1\alpha^{i-1})^2}{2}\ddot{F}_i & \text{(A-24)} \\ F_{i-1} = F_i - \underline{\Delta}x_1\alpha^{i-2}(\underline{\Delta}_x F_i) + \dfrac{(\underline{\Delta}x_1\alpha^{i-2})^2}{2}\ddot{F}_i & \text{(A-25)} \end{cases}$$

Solving for $(\underline{\Delta}_x F_i)$, we have:

$$\underline{\Delta}_x F_i = \frac{1}{\underline{\Delta}x_1\alpha^{i-1}}\left[\frac{-\alpha^2}{1+\alpha}F_{i-1}+(\alpha-1)F_i+\frac{1}{1+\alpha}F_{i+1}\right] \tag{A-26}$$

which is second order correct.

From the following fourth order truncated expansions:

$$F_{i+2} = F_i+(\alpha+1)\underline{\Delta}x_i\dot{F}_i+\frac{(\underline{\Delta}x_i)^2(1+\alpha)^2}{2}(\underline{\Delta}_x^2F_i)+\frac{(\underline{\Delta}x_i)^3(1+\alpha)^3}{6}\dddot{F}_i \tag{A-27}$$

$$F_{i+1} = F_i+\underline{\Delta}x_i\dot{F}_i+\frac{(\underline{\Delta}x_i)^2}{2}(\underline{\Delta}_x^2F_i)+\frac{(\underline{\Delta}x_i)^3}{6}\dddot{F}_i \tag{A-28}$$

$$F_{i-1} = F_i - \frac{\Delta x_i}{\alpha} F_i + \frac{(\Delta x_i)^2}{2\alpha^2}(\Delta_x^2 F_i) - \frac{(\Delta x_i)^3}{6\alpha^3} F_i \ldots \tag{A-29}$$

where $\Delta x_i = \alpha^{i-1}\Delta x_1$.

Solving for $(\Delta_x^2 F_i)$, we have:

$$\Delta_x^2 F_i = \frac{2}{(\Delta x_1 \alpha^{i-1})^2}\left[\frac{\alpha^3(2+\alpha)}{(1+\alpha+\alpha^2)(1+\alpha)} F_{i-1} - \frac{\alpha^2+2\alpha-1}{1+\alpha} F_i + \frac{\alpha^2+\alpha-1}{\alpha(1+\alpha)} F_{i+1} - \frac{\alpha-1}{\alpha(1+\alpha)(1+\alpha+\alpha^2)} F_{i+2}\right] \tag{A-30}$$

Applying these formulae to the variables W and U yields:

$$\Delta_x W_{i,n} = \frac{1}{\Delta x_1 \alpha^{i-1}}\left[\frac{-\alpha^2}{1+\alpha} W_{i-1,n} + (\alpha-1) W_{i,n} + \frac{1}{1+\alpha} W_{i+1,n}\right] \tag{A-31}$$

$$\Delta_x^2 W_{i,n} = \frac{2}{(\Delta x_1 \alpha^{i-1})^2}\left[\frac{\alpha^3(2+\alpha)}{(1+\alpha+\alpha^2)(1+\alpha)} W_{i-1,n} - \frac{\alpha^2+2\alpha-1}{1+\alpha} W_{i,n} + \frac{\alpha^2+\alpha-1}{\alpha(1+\alpha)} W_{i+1,n} - \frac{\alpha-1}{\alpha(1+\alpha)(1+\alpha+\alpha^2)} W_{i+2,n}\right] \tag{A-32}$$

$$\Delta_x U_{i,n} = \frac{1}{\Delta x_1 \alpha^{i-1}}\left[\frac{-\alpha^2}{1+\alpha} U_{i-1,n} + (\alpha-1) U_{i,n} + \frac{1}{1+\alpha} U_{i+1,n}\right] \tag{A-33}$$

$$\Delta_x^2 U_{i,n} = \frac{2}{(\Delta x_1 \alpha^{i-1})^2}\left[\frac{\alpha^3(2+\alpha)}{(1+\alpha+\alpha^2)(1+\alpha)} U_{i-1,n} - \frac{\alpha^2+2\alpha-1}{1+\alpha} U_{i,n} + \frac{\alpha^2+\alpha-1}{\alpha(1+\alpha)} U_{i+1,n} - \frac{\alpha-1}{\alpha(1+\alpha)(1+\alpha+\alpha^2)} U_{i+2,n}\right] \tag{A-34}$$

A.3 Initial and Boundary Conditions

In the notations of the approximate solutions, the finite difference approximation to the initial conditions (A-3) are:

$$W_{i,1} = 1 \quad \text{and} \quad U_{i,1} = 1 \text{ for every } x_i \text{ at } \tau_1 (=0) \tag{A-35}$$

The finite difference approximation to the boundary conditions (A-4) & (A-5) are: (for i=1)

$$W'_{1,n+1} = \eta_T(\tau_{n+1}, W_{1,n+1}, U_{1,n+1}) \tag{A-36}$$

$$U'_{1,n+1} = \eta_r(\tau_{n+1}, W_{1,n+1}, U_{1,n+1}) \tag{A-37}$$

where $W'_{1,n+1}$ = second order correct forward first differrence of $W_{1,n+1}$

$U'_{1,n+1}$ = second order correct forward first difference of $U_{1,n+1}$

(The boundary conditions do not have to be evluated at $\tau_{n+\frac{1}{2}}$ as do the governing equations, because time derivtives re involved in the latters but not in the formers.)

For the case of constant Δx, the second order correct forward difference can be solved from the two truncated equations:

$$\begin{cases} F_{i+2} = F_i + 2\Delta x F'_i + \frac{(2\Delta x)^2}{2}\ddot{F}_i & \text{(A-38)} \\ F_{i+1} = F_i + \Delta x F'_i + \frac{(\Delta x)^2}{2}\ddot{F}_i & \text{(A-39)} \end{cases}$$

here F'_i denotes the second order correct forward first difference instead of exact first derivative.

Eliminating $(\Delta x)^2\ddot{F}_i$ from the above equations, we have:

$$F'_i = \frac{1}{2\Delta x}(-3F_i + 4F_{i+1} - F_{i+2}) \tag{A-40}$$

Similarly, for the case of varying Δx according to eq. (A-23), the second order forward first difference can be solved from the two truncated equations:

$$\begin{cases} F_{i+2} = F_i + (\Delta x_i + \Delta x_{i+1})F'_i + \frac{(\Delta x_i + \Delta x_{i+1})^2}{2}\ddot{F}_i & \text{(A-41)} \\ F_{i+1} = F_i + \Delta x_i F'_i + \frac{(\Delta x_i)^2}{2}\ddot{F}_i & \text{(A-42)} \end{cases}$$

where $\underline{\Delta x}_i = \epsilon^{i-1}\underline{\Delta x}_1$.

The solution comes out to be:

$$F'_i = \frac{1}{\underline{\Delta x}_i}[-\frac{\epsilon+2}{\epsilon+1}F_i + \frac{\epsilon+1}{\epsilon}F_{i+1} - \frac{1}{\epsilon(\epsilon+1)}F_{i+2}] \tag{A-43}$$

For i=1:

$$F'_1 = \frac{1}{\underline{\Delta x}_1}[-\frac{\epsilon+2}{\epsilon+1}F_1 + \frac{\epsilon+1}{\epsilon}F_2 - \frac{1}{\epsilon(\epsilon+1)}F_3] \tag{A-44}$$

Applying these results to the variables W and U yields:

$$W'_{1,n+1} = \frac{1}{\underline{\Delta x}_1}[\frac{-(2+\epsilon)}{1+\epsilon}W_{i,n+1} + \frac{1+\epsilon}{\epsilon}W_{2,n+1} - \frac{1}{\epsilon(1+\epsilon)}W_{3,n+1}] \tag{A-45}$$

$$U'_{1,n+1} = \frac{1}{\underline{\Delta x}_1}[\frac{-(2+\epsilon)}{1+\epsilon}U_{i,n+1} + \frac{1+\epsilon}{\epsilon}U_{2,n+1} - \frac{1}{\epsilon(1+\epsilon)}U_{3,n+1}] \tag{A-46}$$

A.4 Solutions of Finite Difference Equations

Step-by-step method is used to take care of the time (dimensionless) variation. At each step, eqs. (A-13), (A-14), (A-36) and (A-37) comprise a set of nonlinear equations that are to be solved for $W_{i,n+1}$ and $U_{i,n+1}$ for all i. Only a iterative technique can be used to solve a set of nonlinear equations. The Newton-Raphson's method is adopted in this study to carry on the iterations, which is found to be quite powerful and well convergent. The adoption of the predictor-corrector scheme in finding good starting values for iterations also helps to retain both efficency and convergency of the program with reasonable size of time steps.

A.4.1 Predictor-corrector Method

The idea of the predictor-corrector method is that, instead of using the results from previous step as the first guess to start the

iteration, the approximate solution at half of the time step is solved by the simpler linear equations (called "predictor") which needs only the informations at previous step and then the "corrector" is used to obtain an approximate solution for the current step as the first guess for the following iteration of more complicated nonlinear equations.

If the difference equation has the form:

$$\frac{1}{2}\Delta_x^2(F_{i,n+1}+F_{i,n}) - A_1\frac{F_{i,n+1}-F_{i,n}}{\Delta\tau} = \Psi_f[x_i,\tau_{n+\frac{1}{2}},\frac{1}{2}(F_{i,n+1}+F_{i,n}),\frac{1}{2}\Delta_x(F_{i,n+1}+F_{i,n})] \qquad \text{(A-47)}$$

then the predictor is:[18]

$$\frac{1}{2}\Delta_x^2(F_{i,n+\frac{1}{2}}+F_{i,n}) - A_1\frac{F_{i,n+\frac{1}{2}}-F_{i,n}}{\Delta\tau} = \Psi_f(x_i,\tau_{n+\frac{1}{2}},F_{i,n},\Delta_x F_{i,n}) \qquad \text{(A-48)}$$

which becomes a linear algebraic problem, because unknowns only appear in the left hand side with linear forms.

After solving $F_{i,n+\frac{1}{2}}$, it follows with the corrector:[18]

$$\frac{1}{2}\Delta_x^2(F_{i,n+1}+F_{i,n}) - A_1\frac{F_{i,n+1}-F_{i,n}}{\Delta\tau} = \Psi_f(x_i,\tau_{n+\frac{1}{2}},F_{i,n+\frac{1}{2}},\Delta_x F_{i,n+\frac{1}{2}}) \qquad \text{(A-49)}$$

which is also a linear problem. The solution of the corrector, denoted by $F^{(o)}_{i,n+1}$, is then used as the first guess for the iteration:

$$\frac{1}{2}\Delta_x^2(F^{(k)}_{i,n+1}+F_{i,n}) - A_1\frac{F^{(k)}_{i,n+1}-F_{i,n}}{\Delta\tau} = \Psi_f[x_i,\tau_{n+\frac{1}{2}},\frac{1}{2}(F^{(k-1)}_{i,n+1}+F_{i,n}),\frac{1}{2}\Delta_x(F^{(k-1)}_{i,n+1}+F_{i,n})] \qquad \text{(A-50)}$$

where k=1,2,..., with repeated iteration, if necessary, to obtain final solution.

The Crank-Nicolson method with and without predictor-corrector scheme has been tested and compared and the result comes out that the predictor-corrector scheme is very powerful.

A.4.2 Iteration Procedure by Newton-Raphson's Method

Let us multiply eqs. (A-4) & (A-5) by Δx_1 and rewrite as the form:

$$f_1(\vec{W}_{n+1}) = [W'_{1,n+1} - \eta_T(\tau_{n+1}, W_{1,n+1})]\Delta x_1 = 0 \qquad \text{(A-51)}$$

$$g_1(\vec{U}_{n+1}) = [U'_{1,n+1} - \eta_r(\tau_{n+1}, U_{1,n+1})]\Delta x_1 = 0 \qquad \text{(A-52)}$$

Let us also multiply eqs. (A-1) & (A-2) by $2(\Delta x_1)^2$ and rewrite as the form:

$$f_i(\vec{W}_{n+1}) = \{\Delta_x^2(W_{i,n+1}+W_{i,n}) - 2A_1\frac{W_{i,n+1}-W_{i,n}}{\Delta\tau_n} + A_2[\Delta_x(W_{i,n+1}+W_{i,n})] + \frac{A_3}{2}[\Delta_x(W_{i,n+1}+W_{i,n})]^2 + 2A_4\}(\Delta x_1)^2 = 0 \qquad \text{(A-53)}$$

$$g_i(\vec{U}_{n+1}) = \{\Delta_x^2(U_{i,n+1}+U_{i,n}) - 2B_1\frac{U_{i,n+1}-U_{i,n}}{\Delta\tau_n} + B_2[\Delta_x(U_{i,n+1}+U_{i,n})] + \frac{B_3}{2}[\Delta_x(W_{i,n+1}+W_{i,n})][\Delta_x(U_{i,n+1}+U_{i,n})]\}(\Delta x_1)^2 = 0 \qquad \text{(A-54)}$$

where $\vec{W}_{n+1}$ is a vector of $[W_{1,n+1}, W_{2,n+1}, \ldots, W_{m,n+1}]$,

$\vec{U}_{n+1}$ is a vector of $[U_{1,n+1}, U_{2,n+1}, \ldots, U_{m,n+1}]$,

and m is the number of grid beyond which the profiles are essentially constant.

Therefore, we have a set of 2m nonlinear equations to be solved for $\vec{W}_{n+1}$ and $\vec{U}_{n+1}$.

$$\begin{cases} f_1(\vec{W}_{n+1}) = 0 \\ f_2(\vec{W}_{n+1}) = 0 \\ \cdot \\ \cdot \\ \cdot \\ f_m(\vec{W}_{n+1}) = 0 \\ g_1(\vec{U}_{n+1}) = 0 \\ g_2(\vec{U}_{n+1}) = 0 \\ \cdot \\ \cdot \\ \cdot \\ g_m(\vec{U}_{n+1}) = 0 \end{cases} \tag{A-55}$$

To do so, we have to start with an initial guess solutions $\vec{W}^{(o)}_{n+1}$ and $\vec{U}^{(o)}_{n+1}$, from predictor-corrector scheme described in the last section for example, and then find the successive new approximate solutions following the procedure:

(i) For the first iteration of W, $W_{i,n+1}$ and $U_{i,n+1}$ in the coefficient terms of eqs. (A-51) - (A-54) are approximated by $W^{(o)}_{i,n+1}$ and $U^{(o)}_{i,n+1}$ and we can solve the linear equation: [19]

$$\vec{\vec{J}}(\vec{W}^{(o)}_{n+1},\vec{h})(\vec{W}^{(1)}_{n+1}-\vec{W}^{(o)}_{n+1}) + \vec{F}(\vec{W}^{(o)}_{n+1}) = 0 \tag{A-56}$$

where $\vec{\vec{J}}$ is an m x m matrix with elements $\underline{\Delta}_{ij}=\left(\frac{\partial f_i}{\partial W_{j,n+1}}\right)_{\vec{W}^{(o)}_{n+1}}$ if the derivative can be obtained analytically; otherwise, the derivative is approximated by the Steffenson's method, [19]

$$\underline{\Delta}_{ij} = \frac{f_i(\vec{W}^{(o)}_{n+1}+h_j\vec{e}_j)-f_i(\vec{W}^{(o)}_{n+1})}{h_j}$$

$h_j = f_j(\vec{W}^{(o)}_{n+1})$

$\vec{h} = [h_1, h_2, \ldots, h_m]$

and $\vec{F}(\vec{W}^{(o)}_{n+1}) = [f_1(\vec{W}^{(o)}_{n+1}), f_2(\vec{W}^{(o)}_{n+1}), \ldots, f_m(\vec{W}^{(o)}_{n+1})]$

The equation is then solved for $\vec{W}^{(1)}_{n+1}$ by the Gaussian elimination method.

(ii) For the first iteration of U, every $W_{i,n+1}$ is approximated by $W^{(1)}_{i,n+1}$ and $U_{i,n+1}$ in B_1, B_2, B_3 and η_r is approximated by $U^{(o)}_{i,n+1}$, and we can solve the linear equation:

$$\vec{\vec{L}}(\vec{U}^{(o)}_{n+1}, \vec{q})(\vec{U}^{(1)}_{n+1} - \vec{U}^{(o)}_{n+1}) + \vec{G}(\vec{U}^{(o)}_{n+1}) = 0 \qquad \text{(A-57)}$$

where $\vec{\vec{L}}$ is an m x m matrix with elements $\underline{L}_{ij} = \left(\frac{\partial g_i}{\partial U_{j,n+1}}\right)_{\vec{U}^{(o)}_{n+1}}$ if the derivative can be obtained analytically; otherwise, the derivative is approximated by the Steffenson's method,

$$\underline{L}_{ij} = \frac{g_i(\vec{U}^{(o)}_{n+1} + q_j \vec{e}_j) - g_i(\vec{U}^{(o)}_{n+1})}{q_j}$$

$q_j = g_j(\vec{U}^{(o)}_{n+1})$

$\vec{q} = [q_1, q_2, \ldots, q_m]$

and $\vec{G}(\vec{U}^{(o)}_{n+1}) = [g_1(\vec{U}^{(o)}_{n+1}), g_2(\vec{U}^{(o)}_{n+1}), \ldots, g_m(\vec{U}^{(o)}_{n+1})]$

The equation is again solved for $\vec{U}^{(1)}_{n+1}$ by the Gaussian elimination method.

(iii) The approximate solutions $W^{(1)}_{i,n+1}$ and $U^{(1)}_{i,n+1}$ from the first iteration are then used to calculate solutions for the second iteration.

(iv) The process is repeated until the successive iterations are sufficiently close to each other, and then the calculations go on to the next step after determination of the next step size.

A.4.3 Gaussian Elimination Method

Either the predictor-corrector mehtod or Newton-Raphson's method is a kind of linearization process in order to deal with a nonlinear problem. After linearization, another numerical scheme has to be chosen to solve the system of linear equations, and the Gaussian elimination back-substitution is found to be a very easy and efficient way.

In solving eqs. (A-48), (A-49) in section A.4.1 and (A-56), (A-57) in section A.4.2, we are all dealing with a set of equations with the format like:

$$
\begin{aligned}
b_1\delta_1+c_1\delta_2+d_1\delta_3 &= e_1\\
a_2\delta_1+b_2\delta_2+c_2\delta_3+d_2\delta_4 &= e_2\\
a_3\delta_2+b_3\delta_3+c_3\delta_4+d_3\delta_5 &= e_3\\
\cdots\cdots\cdots\cdots\cdots\\
a_i\delta_{i-1}+b_i\delta_i+c_i\delta_{i+1}+d_i\delta_{i+2} &= e_i\\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
a_{m-1}\delta_{m-2}+b'_{m-1}\delta_{m-1}+c'_{m-1}\delta_m &= e_{m-1}\\
a'_m\delta_{m-1}+b'_m\delta_m &= e_m
\end{aligned}
\qquad \text{(A-58)}
$$

where δ_i denotes either the variables $W_{i,n+\frac{1}{2}}$, $U_{i,n+\frac{1}{2}}$ in (A-48), $W_{i,n+1}$, $U_{i,n+1}$ in (A-49) or $(\vec{W}^{(k)}_{n+1}-\vec{W}^{(k-1)}_{n+1})$ in (A-56), $(\vec{U}^{(k)}_{n+1}-\vec{U}^{(k-1)}_{n+1})$ in (A-57), a_i, b_i, c_i, d_i denote either the coefficients in (A-48), (A-49) with eqs. (A-31) - (A-34), and (A-45), (A-46), or the elements of Jacobin matrixes $\vec{\vec{J}}$, $\vec{\vec{L}}$ in (A-56), (A-57), and e_i denotes the known quantities

from either the previous step in eqs. (A-48) and (A-49) or the previous iteration in eqs. (A-56) and (A-57).

Note that the primes in b'_{m-1}, c'_{m-1}, a'_m and b'_m are because we approximated δ_{m+1} and δ_{m+2} by a linear extrapolation of δ_{m-1} and δ_m in order to eliminate the number of unknowns to m,

$$\begin{cases} b'_{m-1} = b_{m-1} - d_{m-1}\alpha \\ c'_{m-1} = c_{m-1} + d_{m-1}(1+\alpha) \\ a'_m = a_m - c_m\alpha - d_m\alpha(1+\alpha) \\ b'_m = b_m + c_m(1+\alpha) + d_m(1+\alpha+\alpha^2) \end{cases} \tag{A-59}$$

The system of equations (A-58) can be put in the matrix form:

$$\vec{\vec{M}}\vec{\Delta} = \vec{E} \tag{A-60}$$

where $\vec{\Delta}$ is a column vector $\delta_1, \delta_2, \ldots, \delta_m$

$\vec{E}$ is a column vector $e_1, e_2, \ldots, e_m$

and $\vec{\vec{M}}$ is an m x m quad-diagonal matrix since all elements are zero except those on the principal diagonal, one below and one and two above the principal diagonal.

$$\begin{bmatrix} \ddots & \ddots & \ddots & 0 \\ & \ddots & \ddots & d \\ & & c & \ddots \\ & b & \ddots & \\ a & \ddots & & \\ 0 & & & \end{bmatrix} \times \begin{bmatrix} \delta_1 \\ \delta_2 \\ \cdot \\ \cdot \\ \cdot \\ \delta_m \end{bmatrix} = \begin{bmatrix} e_1 \\ e_2 \\ \cdot \\ \cdot \\ \cdot \\ e_m \end{bmatrix} \tag{A-61}$$

The idea behind the Gaussian elimination method is to eliminate the

unknows in a systematic way; the first equation can be used to eliminate δ_1 from the second equation, the new second equation used to eliminate δ_2 from the third equation, and so on, until finally, the new last but one equation can be used to eliminate δ_{m-1} from the last equation, giving one equation with only one unknown δ_m. The unknowns δ_i can then be found in turn by back-substitution.(19)

Generally, after i-2 eliminations ($i \geq 2$), we obtain the following two equations for the next elimination:

$$\begin{cases} \alpha_{i-1}\delta_{i-1} + \beta_{i-1}\delta_i + d_{i-1}\delta_{i+1} = S_{i-1} \\ a_i\delta_{i-1} + b_i\delta_i + c_i\delta_{i+1} + d_i\delta_{i+2} = e_i \end{cases} \tag{A-62}$$

where $\alpha_1=b_1$, $\beta_1=c_1$, $S_1=e_1$.

Eliminating δ_{i-1} leads to:

$$(b_i - \frac{a_i\beta_{i-1}}{\alpha_{i-1}})\delta_i + (c_i - \frac{a_i d_{i-1}}{\alpha_{i-1}})\delta_{i+1} + d_i\delta_{i+2} = e_i - \frac{a_i S_{i-1}}{\alpha_{i-1}} \tag{A-63}$$

i.e. $\alpha_i\delta_i + \beta_i\delta_{i+1} + d_i\delta_{i+2} = S_i$ (A-64)

with the recursion relations:

$$\begin{cases} \alpha_i = b_i - \dfrac{a_i\beta_{i-1}}{\alpha_{i-1}} \\ \beta_i = c_i - \dfrac{a_i d_{i-1}}{\alpha_{i-1}} \qquad \text{when } 1 \leq i \leq m-2 \\ S_i = e_i - \dfrac{a_i S_{i-1}}{\alpha_{i-1}} \end{cases} \tag{A-65}$$

After m-3 eliminations, the last three equations are:

$$\begin{cases} \alpha_{m-2}\delta_{m-2} + \beta_{m-2}\delta_{m-1} + d_{m-2}\delta_m = S_{m-2} \\ a_{m-1}\delta_{m-2} + b'_{m-1}\delta_{m-1} + c'_{m-1}\delta_m = e_{m-1} \\ a'_m\delta_{m-1} + b'_m\delta_m \qquad\qquad = e_m \end{cases} \tag{A-66}$$

Eliminating δ_{m-2} from the first two equations yields:

$$\begin{cases} \alpha'_{m-1}\delta_{m-1} + \beta'_{m-1}\delta_m = S_{m-1} \\ a'_m\delta_{m-1} + b'_m\delta_m \qquad = e_m \end{cases} \tag{A-67}$$

where $\alpha'_{m-1} = b'_{m-1} - \dfrac{a_{m-1}\beta_{m-2}}{\alpha_{m-2}}$

$$\beta'_{m-1} = c'_{m-1} - \frac{a_{m-1}d_{m-2}}{\alpha_{m-2}}$$

$$S_{m-1} = e_{m-1} - \frac{a_{m-1}S_{m-2}}{\alpha_{m-2}}$$

Eliminating δ_{m-1} from (A-67) yields:

$$\alpha'_m\delta_m = S'_m \tag{A-68}$$

where $\alpha'_m = b'_m - \dfrac{a'_m\beta'_{m-1}}{\alpha'_{m-1}}$

$$S'_m = e_m - \frac{a'_m S_{m-1}}{\alpha_{m-1}}$$

Therefore, the solutions are:

$$\begin{cases} \delta_m = \dfrac{S'_m}{\alpha'_m} \\ \delta_{m-1} = \dfrac{S_{m-1} - \beta'_{m-1}\delta_m}{\alpha'_{m-1}} \\ \cdot \\ \cdot \\ \cdot \\ \delta_i = \dfrac{1}{\alpha_i}(S_i - \beta_i\delta_{i+1} - d_i\delta_{i+2}) \qquad 1 \le i \le m-1 \end{cases} \tag{A-69}$$

A.5 Determination of the Time Increment

The step-by-step methods as described above could be either one-step methods or multi-step methods, depending upon whether the informations from previous steps are used in taking the next step. The multi-step methods are more efficient in the sense that generally they require fewer evaluations of the difference operators to achieve a given accuracy. The greater efficiency of the multi-step methods is obtained at the cost of requiring special provisions for changing the step size. Considering the fact that the general shape of the solutions could be closely estimated and the powerful iterational method used here, the multi-step step-by-step method is used in this study and the way of determining the step size is described as the following:

Define the ratio of the truncated second order term to the first order term as:

$$d_f = \frac{|(\Delta\tau)^2 \ddot{f}|}{(\Delta\tau)|\dot{f}|} = \frac{\Delta\tau|\ddot{f}|}{|\dot{f}|} \tag{A-70}$$

where f is either T or r, $\dot{f}$ and $\ddot{f}$ are first and second order derivative of f, respectively.

In the problem we are dealing with, the composition changes much slowlier than the temperature, so the time increment is determined by the truncated error ratio for W:

$$d_W = \frac{\left|\Delta\left(\frac{\Delta W}{\Delta\tau}\right)\right|}{\left|\frac{\Delta W}{\Delta\tau}\right|} = \frac{\left|\left(\frac{\Delta W}{\Delta\tau}\right)_{n+1} - \left(\frac{\Delta W}{\Delta\tau}\right)_n\right|}{\left|\left(\frac{\Delta W}{\Delta\tau}\right)_{n+1}\right|} = \frac{\left|\frac{W_{n+1}-W_n}{\Delta\tau_n} - \frac{W_n-W_{n-1}}{\Delta\tau_{n-1}}\right|}{\left|\frac{W_{n+1}-W_n}{\Delta\tau_n}\right|} \tag{A-71}$$

And the criterion is that $\Delta\tau$ is increased when d_W is less than 10%.

while $\Delta\tau$ is decreased when d_W is larger than 10%. A maximum of three times the previous step size is imposed. Both upper and lower bounds for the permissible step size are set considered the convergency and efficiency.

A flow chart of the program is shown in Fig. A.1.

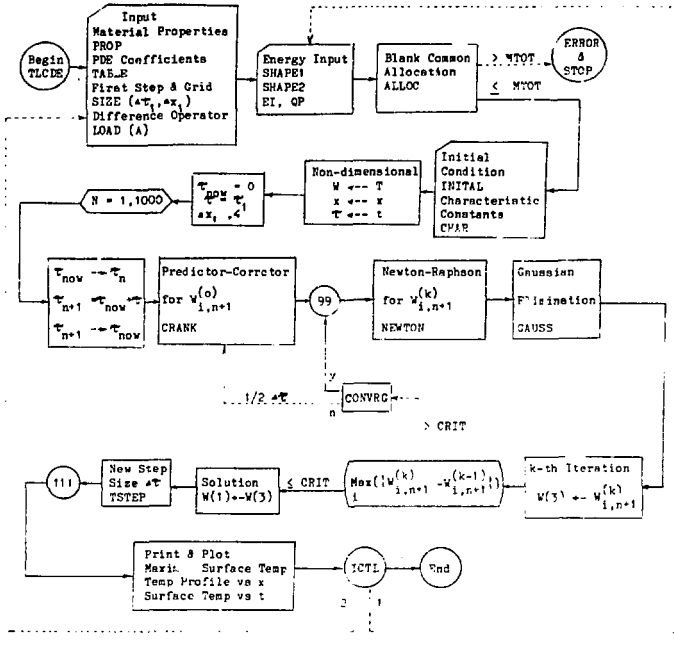


Fig. A.1 The Flow Diagram of the Computer Programs "STAR" and "SURFT"

APPENDIX B: COMPUTER PROGRAMS AND SAMPLE ANALYSES

B.1 Computer Programs:

Three computer programs have been developed in this study. The program "STAR" (Surface Temperature And composition Ratio calculation) is for the materials which vaporize incongruently so that the surface composition changes with time coupled with a temperature transient. In this program, one-dimensional time dependent heat conduction and diffusion equations are solved considering melting, ablating moving boundary, ablation and radiation heat losses. The program "SURFT" (SURFace Temperature calculation) is for the materials which vaporize congruently so that no diffusion process is involved and only one-dimensional heat conduction equation with ablating moving boundary and ablation and radiation heat losses is solved. The program "SURFR" (SURFace composition Ratio calculation) is for the incongruently vaporizing materials in which the surface and bulk temperature is kept constant. Only one-dimensional diffusion equation with ablating moving boundary is solved.

All the programs have been developed based on the numerical scheme described in Appendix A and are coded in FORTRAN IV language. Variable dimensioning is used to make optimum usage of the available storage and flexible capacity controllable by the user. The total blank common block storage MTOT has to be greater than

```
NGMAX*(NPDE*7+11)+LSPUL*(3+NPDE)+NPDE*15
```

An error message will be generated and the run will be aborted if MTOT is set too small.

B.2 Program Listing:

```
STR001        PROGRAM STAR(INPUT,OUTPUT,TAPE5=INPUT,TAPE4=OUTPUT,TAPE99)
STR002  C
STR003  C ##################################################################
STR004  C # THIS PROGRAM CALCULATES THE TRANSIENT HEAT CONDUCTION AND MASS  #
STR005  C # TRANSPORT PROBLEMS OF A COMPOUND SUBJECT TO LASER IRRADIATIONS  #
STR006  C # BY SOLVING COUPLED ONE DIMENSIONAL TIME DEPENDENT HEAT CONDUCTION #
STR007  C # AND DIFFUSION EQUATIONS SIMULTANEOUSLY CONSIDERING MELTING,     #
STR008  C # ABLATING MOVING BOUNDARY AND RADIATION.                          #
STR009  C # THE INCONGRUENT VAPORIZATION IS CONSIDERED.                      #
STR010  C # THIS PROGRAM ALLOWS EITHER THE FINEMATION OF LASER RADIATION    #
STR011  C # (VOLUMETRIC HEATING) OR ABSORBING SURFACE (SURFACE HEATING).    #
STR012  C # THE NUMERICAL METHODS USED ARE CRANK-NICOLSON FINITE DIFFERENCE #
STR013  C # METHOD WITH VARIABLE GRID SPACE INCREMENT AND SELF-ADJUSTABLE   #
STR014  C # TIME INCREMENT, NEWTON RAPHSON AND GAUSSIAN ELIMINATION BACK   #
STR015  C # SUBSTITUTION METHODS.                                            #
STR016  C # THE PREDICTOR-CORRECTOR SCHEME IS USED FOR FINDING BETTER INITIAL #
STR017  C # GUESS AND FASTER CONVERGENCE.                                   #
STR018  C #                                                                 #
STR019  C # DEVELOPED BY C. H. (SAT. JUNE 1980.                             #
STR020  C # UNIVERSITY OF CALIFORNIA - BERKELEY                             #
STR021  C ##################################################################
STR022  C
STR023        COMMON/BLOCK/NTOT
STR024        COMMON/DIM /NPDE,NDMAX,LSPUL
STR025        COMMON/DMAIN/N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13
STR026        COMMON/CHART/XCH,TCH
STR027        COMMON/LASER/EI,DP,N,IPUL,AEFF
STR028        COMMON/STEP /DT1
STR029        COMMON C(8000)
STR030  C
STR031        NTOT=8000
STR032  C
STR033  C     ##########################################
STR034  C     #LAHA        COMMON        STORAGE      ALLOCATION
STR035  C          ARRAY   ------ DESCRIPTION -----   DIMENSION
STR036  C     N1   W       DIMENSIONLESS SOLUTION     NDMAX*3*NPDE
STR037  C     N2   A       DIFFERENTIATOR COEFF       NDMAX*7
STR038  C     N3   SB      NORMALIZED PULSE SHAPE     LSPUL
STR039  C     N4   TT      TIME INDEX FOR SS (DIM)    LSPUL
STR040  C     N5   FF      PEAK POWER DENSITY AT IT   LSPUL
STR041  C     N6   WCH     CHART QUANTITY FOR W (DIM) NPDE
STR042  C     N7   WS      SURFACE W AT INDW          NPDE
STR043  C     N8   WSURF   SURF W AS FUNC OF TT (DIM) LSPUL*NPDE
STR044  C     N9   WX      W PROFILE AS FUNC OF X (DIM) NDMAX*NPDE
STR045  C     N10  X       SPACE GRID FOR PROFILE WX  NDMAX
STR046  C     N11  WO      INITIAL VALUE OF W (DIM)   NPDE
STR047  C     N12  WOLD    SURFACE W AT TOLD (DIMLSS) NPDE
STR048  C     N13  WLAST   WS OF LAST ITERATION       NPDE
STR049  C     ##########################################
STR050  C
STR051  C  INITIATE OR STOP THE PROGRAM
STR052  C  READ NTIN,NDMAX
STR053     99 CALL ICCOL
STR054  C
STR055  C  INPUT THE MATERIAL PROPERTIES AND LASER CHARACTERISTICS
STR056        CALL PROP
STR057  C  GENERATION OF COEFFICIENT TABLES
STR058        CALL TABLE
STR059  C
STR060  C  STEP AND GRID SIZES FOR NUMERICAL STABILITY
STR061        CALL SIZE
STR062  C
STR063        CALL SHAPE1
STR064  C
STR065  C  DEFINE COMMON BLOCK ALLOCATION
STR066        CALL ALLOC
STR067  C
STR068  C  INPUT LASER NORMALIZED PULSE SHAPE
STR069        CALL SHAPE2(C(N3),C(N4),LSPUL)
STR070  C
STR071  C  LOAD CONSTANT VECTORS A'S FOR DIFFERENCE OPERATORS
STR072        CALL LOAD(C(N2),NDMAX)
STR073  C
STR074  C  CALCULATION TO BE ENDED AT TSTOP (IN SEC)
STR075        READ (5,9) TSTOP,ICYCL
STR076     9  FORMAT(2E10.0)
STR077        IF(TSTOP.GT.C(N5-1)) TSTOP=C(N5-1)
STR078  C  DEFAULT OF ICYCL FOR SINGLE PULSE IS 100.,
STR079  C  WHICH IS ASSUMED TO BE INFINITIVE.
STR080        IF(ICYCL.EQ.0.0) ICYCL=100.
STR081  C
STR082  C  INPUT LASER ENERGY
STR083     88 READ (5,8) EI
STR084     8  FORMAT(E10.0)
STR085        IF(EI.EQ.0.0) CALL EXIT
STR086        DP=EI/(IPUL*AEFF)
STR087        DO 1 I=1,LSPUL
STR088        IM=I-1
STR089     1  C(N4+IM)=DP*C(N3+IM)*(1.-R)
STR090  C
STR091  C  LOAD INITIAL CONDITIONS
STR092        CALL INITIAL(C(N1),C(N11),NPDE,NDMAX)
STR093  C  CHARACTERISTIC CONSTANTS FOR NON-DIMENSIONALIZATION
STR094        CALL CHAR(C(N11),C(N6),NPDE)
STR095  C
STR096        TNOW =0.0
STR097        DT   =DT1
STR098        IFLAG=0
STR099        DO 2 I=1,NPDE
STR100        JL=(I-1)*LSPUL
STR101     2  C(N8+JL)=C(N11+I-1)
STR102        TOM  =0.0
STR103        TMAX =0.0
STR104        AMAX =0.0
STR105        MMIN =0
STR106        K    =MMIN
STR107        K    =2
STR108  C
```

```
C  STEP-BY-STEP NUMERICAL CALCULATION
      DO 111 N=1,1000
      TOLD=TNOW
      IF(N.EQ.1) GOTO 11
      DO I I=1,NPDE
      C(NI2+I-1)=C(NI1+I-1)*NOMAX
   3  CALL MMRCAL(C(NI1),TNOW,DT,C(N4),N,C(N2),C(N3),LSPUL,
   11 1                N,IFLAG,DWS,C(N7),C(N13),NPDE,NOMAX)
      TIME=TNOW*TCH
      TS=C(N7)*C(N4)
C
C  TOTAL AMOUNT OF MATERIAL EVAPORATED
      RS=C(N7)*C(N4)
      CALL MFLUX(TIME,TS,RS,AME,AME1,AME2,IFLAD)
      DAMDT=(AME-AMD D)/(DTATCH)
      AMDD=AME
      AMEDT=AMEDT+AMEFF*TCH
      TAME=TAME+AMEDT
      FRAC=AMEDT/TAME
C
C  DETERMINE NEXT TIME STEP SIZE
      CALL TSTEP(DT,DWS,N)
C
C  MAXIMUM SURFACE TEMPERATURE
      IF(TS.LT.TSMAX) GOTO 20
      TSMAX=TS
      RTMAX=RS
      TIMAX=TIME*1.0E3
      LM=N+7
      DO 10 I=1,NPDE
      IM=(I-1)*NOMAX
      DO 10 J=1,LM
      JI=IM+J-1
      JJ=IM+3*J-1
   10 C(N9+JI)=C(N1+JJ)*C(N4+I-1)
C
   20 IF(TIME.LT.C(N4+N-1)) GOTO 111
C
      DO 30 I=1,NPDE
      IM=(I-1)*LSPUL
   30 CINDI(N+K-1)=(C(N12+I-1)+(C(N4+K-1)/TCH-TOLD)*(C(N1+3*I-1)*
     1                NOMAX)-C(N12+I-1))/(TNOW-TOLD)*C(N4+I-1)
      IF(C(N4+N-1).GE.TSTOP) GOTO 40
      K=K+1
  111 CONTINUE
C
   40 KKM
C
C  TEMP AND COMP PROFILE AT THE TIME WHEN SURF TEMP IS MAX
      CALL PRFLE(TSMAX,TIMAX,C(N9),C(N10),LM,NPDE,NOMAX)
      DO 4 I=1,LM
    4 C(N10+I-1)=C(N10+I-1)*1.E4
      IFRM=0
      CALL PLOTTX(C(N10),C(N9),LM,IFRM)
      CALL PLOTRX(C(N10),C(N9*NOMAX),LM,IFRM)
C
C  TOTAL AMOUNT OF MATERIAL EVAPORATED
      IFLAG=1
      CALL MFLUX(TIMAX/1.E-3,TSMAX,RTMAX,AMET,AMET1,AMET2,IFLAD)
      IF(DAMDT.GE.0.0) GOTO 41
      TEND=TIME-AME/DAMDT
      GOTO 42
   41 TEND=TIME
   42 CONTINUE
      IF(TEND.GT.TCYCL) TEND=TCYCL
      TAME1=TAME+AMEEFF*(TEND-TIME)
      TAME2=AMET*TPULSE*EFF
      WRITE (6,103) TAME,FRAC,TAME2
      WRITE (6,104) TCYCL,TEND,TAME1
C
C  SURFACE TEMPERATURE AND COMPOSITION RATIOS
C
   50 WRITE (6,100)
      WRITE (6,101) EI,DP
      WRITE (6,102)
      DO 70 I=1,KM
      IF(I.GT.2) GOTO 60
      IFLAG=IFLAG+1
   60 CALL MFLUX(C(N4+I-1),C(N8+I-1),C(N8+LSPUL+I-1),FLUX,F1,F2,IFLAD)
   70 CONTINUE
C
      DO 5 I=1,KM
    5 C(N4+I-1)=C(N4+I-1)*1.E3
      CALL PLOTTS(C(N4),C(N8),KM,IFRM)
C
  100 FORMAT(1H1,30X,08(1H8))
  101 FORMAT(1H ,13X,46HTHE SURFACE TEMPERATURES AND COMPOSITION RATIOS
     1WITH TIME FOR EI =,E10.3,9H (JOULES)/
     2                 18X,4HDP =,E10.3,8H (W/CM2))
  102 FORMAT(1H ,10X,8HI(HR)/)
  103 FORMAT(1H0,4X,64HTHE TOTAL AMOUNT OF MATERIAL EVAPORATED FROM TIME
     1 INTEGRATION IS,E13.6,4H GRAMS/
     2                 2X,45HFRACTION OF THE MASS OF THE STEP TO TOTAL IS,E13.6,/
     3                 2X,7GHTHE TOTAL AMOUNT OF MATERIAL EVAPORATED FROM MAX SURF
     4 TEMP AND TPUL CALC IS,E13.6,6H GRAMS)
  104 FORMAT(1H0,4X,7HTCYCL =,E13.6,15H SEC AND TEND =,E13.6,4H SEC/
     1                 7X,80HTHE TOTAL AMOUNT OF MATERIAL EVAPORATED TAKING ACCOUN
     2T THE AMOUNT FROM TIME TO TEND IS,E13.6,6H GRAMS)
C
      READ (5,200) ICTL
      IF(ICTL.EQ.0.OR.ICTL.GT.2) IFRM=1
      CALL PLOTRS(C(N4),C(N8+LSPUL),KM,IFRM)
      DO 51 I=1,KM
   51 C(N4+I-1)=C(N4+I-1)/1.E3
      IF(ICTL.EQ.1) GOTO 80
      IF(ICTL.EQ.2) GOTO 99
  200 FORMAT(I1)
      STOP
      END
```

```
TLC001        SUBROUTINE TLCDE
TLC002  C
TLC003  C  READ CONTROL STATEMENT FOR PROBLEM INITIATION
TLC004  C  (NOT PROBLEM NOR MATER DEPENT)
TLC005  C
TLC006        COMMON/DIM  /NPDE,NGMAX,LSPUL
TLC007        DIMENSION HED(4),DATE(2)
TLC008        DIMENSION W3D(2)
TLC009        DATA NOD/3HSTA/,5HSTOP /
TLC010  C
TLC011   10   READ (5,100) MODE,NTYPE,HED,DATE
TLC012        IF(MODE.EQ.NOD(1)) STOP
TLC013        IF(MODE.EQ.NOD(1)) GOTO 20
TLC014        WRITE (6,300)
TLC015        GOTO 10
TLC016  C  READ NO OF PARTIAL DIFFERENTIAL EQUATIONS,
TLC017  C  AND DIMENSION OF MAXIMUM GRID POINTS
TLC018   20   READ (5,101) NPDE,NGMAX
TLC019        IF(NGMAX.GT.0) GOTO 30
TLC020        WRITE (6,301)
TLC021   30   WRITE (6,200) HED,DATE
TLC022        WRITE (6,201) NPDE,NGMAX
TLC023  C
TLC024   100 FORMAT(2A5,5X)/6A5,5X,2A5)
TLC025   101 FORMAT(2I5)
TLC026   200 FORMAT(1H1,5(1H0)/2X,6A5,2X,5(1H0)/1X,5(1H0),12X,2A5,12X,5(1H0)//)
TLC027   201 FORMAT(1H0,40HTHE NUMBER OF DIFFERENTIAL EQUATIONS IS ,I2/
TLC028     1          1X,37HTHE MAXIMUM NUMBER OF GRID PLANES IS ,I5//)
TLC029   300 FORMAT(//46H **ERROR** DATA DECK MUST BEGIN WITH START CARD)
TLC030   301 FORMAT(//46H **ERROR** NO. OF GRID PLANES MUST BE GT ZERO)
TLC031  C
TLC032        RETURN
TLC033        END


ALO001        SUBROUTINE ALLOC
ALO002  C
ALO003  C  DEFINE COMMON BLOCK STORAGE ALLOCATION FOR
ALO004  C  THE MAIN PROGRAM AND SUBROUTINES
ALO005  C
ALO006        COMMON/BLOCK/NTOT
ALO007        COMMON/DIM  /NPDE,NGMAX,LSPUL
ALO008        COMMON/DRAIN/M1,M2,M3,M4,M5,M6,M7,M8,M9,M10,M11,M12,M13
ALO009        COMMON/DCRAN/M1,M2,M3,M4,M5,M6,M7,M8,M9,M10,M11,M12,M13
ALO010        COMMON/DMENT/L1,L2,L3,L4,L5,L6,L7,L8,L9,L10,L11,L12,L13,L14,
ALO011     1                L15,L16,L17
ALO012  C
ALO013  C  PARTITION OF BLANK COMMON FOR MAIN PROGRAM
ALO014        M1 =1
ALO015        M2 =M1 +NGMAX*30*NPDE
ALO016        M3 =M2 +NGMAX*7
ALO017        M4 =M3 +LSPUL
ALO018        M5 =M4 +LSPUL
ALO019        M6 =M5 +LSPUL
ALO020        M7 =M6 +NPDE
ALO021        M8 =M7 +NPDE
ALO022        M9 =M8 +LSPUL*NPDE
ALO023        M10=M9 +NGMAX*NPDE
ALO024        M11=M10+NGMAX
ALO025        M12=M11+NPDE
ALO026        M13=M12+NPDE
ALO027        MLAST=M13+NPDE
ALO028        IF(MLAST.GT.NTOT) CALL ERROR(MLAST-NTOT)
ALO029  C
ALO030  C  PARTITION OF BLANK COMMON FOR SUBROUTINE CRANK
ALO031        M1 =MLAST
ALO032        M2 =M1 +NGMAX*30*NPDE
ALO033        M3 =M2 +NGMAX*NPDE
ALO034        M4 =M3 +NGMAX*NPDE
ALO035        M5 =M4 +NPDE
ALO036        M6 =M5 +NPDE
ALO037        M7 =M6 +NPDE
ALO038        M8 =M7 +NPDE
ALO039        M9 =M8 +NPDE
ALO040        M10=M9 +NPDE
ALO041        M11=M10+NPDE
ALO042        M12=M11+NPDE
ALO043        M13=M12+NGMAX
ALO044        IF(M13.GT.NTOT) CALL ERROR(M13-NTOT)
ALO045  C
ALO046  C  PARTITION OF BLANK COMMON FOR SUBROUTINE NEWTON
ALO047        L1 =MLAST
ALO048        L2 =L1 +NGMAX
ALO049        L3 =L2 +NGMAX
ALO050        L4 =L3 +NGMAX
ALO051        L5 =L4 +NPDE
ALO052        L6 =L5 +NPDE
ALO053        L7 =L6 +NPDE
ALO054        L8 =L7 +NPDE
ALO055        L9 =L8 +NPDE
ALO056        L10=L9 +NPDE
ALO057        L11=L10+NGMAX*NPDE
ALO058        L12=L11+NGMAX*NPDE
ALO059        L13=L12+NPDE
ALO060        L14=L13+NPDE
ALO061        L15=L14+NPDE
ALO062        L16=L15+NPDE
ALO063        L17=L16+NGMAX
ALO064        IF(L17.GT.NTOT) CALL ERROR(L17-NTOT)
ALO065  C
ALO066        RETURN
ALO067        END


SYL001        SUBROUTINE SIZE
SYL002  C
SYL003  C  THIS SUB. CONTAINS SIZES FOR MAXIMUM DIMENSION
SYL004  C  (NOT PARAMETER NOR MAIN DEPENT)
```

```
      C
            COMMON/GRID /DX1,EPS,EPS1,EPS2
            COMMON/ITER /ITMAX,CRIT
            COMMON/STEP /DT1
      C
            READ (5,100) DT1,DX1,EPS
            WRITE (6,200) DT1,DX1,EPS
            READ (5,101) ITMAX,CRIT
            WRITE (6,201) ITMAX,CRIT
      C
        100 FORMAT(2E10.0,F10.0)
        101 FORMAT(I10,E10.0)
        200 FORMAT(1H0,23HTHE NUMERICAL CONSTANTS/
           1          5X,21HTHE FIRST STEP SIZE =,E9.3/
           2          5X,21HTHE FIRST GRID SIZE =,E9.3/
           3          5X,21HGRID INCRE FACTOR   =,F9.3)
        201 FORMAT(5X,21HMAX NO OF ITERATION =,I9/
           1          5X,21HERROR TOLERANCE     =,E9.3)
      C
            RETURN
            END


            SUBROUTINE LOAD(A,NDMAX)
      C
      C LOAD CONSTANT VECTORS FOR DIFFERENCE OPERATORS
      C (NOT PROBLEM NOR MATER DEPENT)
      C
            COMMON/GRID /DX1,EPS,EPS1 EPS2
            DIMENSION A(NDMAX,7)
      C
            EPS1=1.*EPS
            EPS2=1.*EPS*EPS**2
            I=2
            A(I,1)=2.*EPS*(2.*EPS)/(EPS*EPS2)
            A(I,2)=-2.*(EPS**2+2.*EPS-1.)/(EPS**2*EPS1)
            A(I,3)=2.*(EPS**2*EPS-1.)/(EPS1*EPS**3)
            A(I,4)=2.*(1.-EPS)/(EPS**3*EPS1*EPS2)
            A(I,5)=-EPS/EPS1
            A(I,6)=(EPS-1.)/EPS
            A(I,7)=1./(EPS*EPS1)
            DO 1 J=1,4
            DO 1 I=3,NDMAX
        1   A(I,J)=A(I-1,J)/EPS**2
            DO 2 J=5,7
            DO 2 I=3,NDMAX
        2   A(I,J)=A(I-1,J)/EPS
            I=1
            A(I,1)=0.0
            A(I,2)=-(EPS+2.)/EPS1
            A(I,3)=(EPS)/EPS
            A(I,4)=-1./(EPS*EPS1)
            RETURN
            END


            SUBROUTINE SHAPE1
      C
      C READ SDT AND LSPUL
      C
            COMMON/DIM  /M(2),LSPUL
            COMMON/PULS /SDT
      C
            READ (5,100) SDT,LSPUL
            WRITE (6,200) SDT,LSPUL
        100 FORMAT(E10.1,I10)
        200 FORMAT(1H0,25HENP PRINT OUT PARAMETERS/
           1          5X,4HSDT=,E7.2,4H SEC,5X,4HLSPUL=,I3)
      C
            RETURN
            END


            SUBROUTINE SHAPE2(SS,TT,LSPUL)
      C
      C LASER NORMALIZED PULSE SHAPE
      C (NOT PROBLEM NOR MATER DEPENT)
      C
            COMMON/PULS /SDT
            DIMENSION SS(LSPUL),TT(LSPUL)
      C
            DO 1 I=1,LSPUL
        1   TT(I)=SDT*FLOAT(I-1)
            READ (5,101) (SS(I),I=1,LSPUL)
            WRITE (6,201)
            WRITE (6,202)
            WRITE (6,203)
            WRITE (6,204) (I,TT(I),I,SS(I),I=1,LSPUL)
      C
        101 FORMAT(7F10.3)
        201 FORMAT(1H1,55(1H*))
        202 FORMAT(1H ,1X,53HHISTOGRAM OF TEMPORAL SHAPE OF PULSE FROM POWER T
           1RACE)
        203 FORMAT(1H ,55(1H*))
        204 FORMAT(1H ,10X,3HTT(,I2,3H) =,E8.2,5X,3HSS(,I2,3H) =,F6.3)
      C
            RETURN
            END


            FUNCTION SOURCE(TIME,FT,LSPUL)
      C
      C TEMPORAL SHAPE FROM LASER POWER TRACE
      C
            COMMON/PULS /SDT
            DIMENSION FT(LSPUL)
      C
            XI=TIME/SDT
            I=XI
            IF(I.GE.LSPUL) I=LSPUL-1
```

```
SRC011        SOURCE=FF(I+1)*(X(I-1)*(FF(I+2)-FF(I+1))
SRC012        RETURN
SRC013    1   SOURCE=0.0
SRC014        RETURN
SRC015        END


CRG001        SUBROUTINE CONVRG(W,WLAST,NIT,M,MRESET,DT,MPDE,MGMAX)
CRG002  C
CRG003  C  TEST THE CONVERGENCY OF ITERATIONS
CRG004  C  (NOT PROBLEM NOR MATER DEPENT)
CRG005  C
CRG006        DIMENSION W(MGMAX,3,MPDE),WLAST(MPDE)
CRG007  C
CRG008        IF(NIT.EQ.0) GOTO 2
CRG009        DO 10 I=1,MPDE
CRG010        RATIO=ABS(W(1,3,I)-W(1,2,I))/(W(1,2,I)-WLAST(I)))
CRG011        IF(RATIO.GE.1.) GOTO 1
CRG012   10   CONTINUE
CRG013        GOTO 2
CRG014  C  IF NOT CONVERGE, DT WAS TOO LARGE
CRG015  C  DT REDUCED TO HALF
CRG014    1   DT=DT/2.
CRG017        M=MRESET
CRG018        NIT=0
CRG019        RETURN
CRG020  C  IF CONVERGE, (I.3) TRANSFER TO (I.2), ITERATION GOES ON
CRG021    2   CONTINUE
CRG022        DO 20 I=1,MPDE
CRG023   20   WLAST(I)=W(1,2,I)
CRG024        L1=M+2
CRG025        DO 3 I=1,MPDE
CRG026        DO 3 J=1,L1
CRG027        W(J,2,I)=W(J,3,I)
CRG028    3   CONTINUE
CRG029        NIT=NIT+1
CRG030        RETURN
CRG031        END


TSTP01        SUBROUTINE TSTEP(DT,DWS,M)
TSTP02  C
TSTP03  C  DETERMINATION OF TIME INCREMENT BY COMPARING SUCCESIVE TIME
TSTP04  C  DERIVATIVE OF SURFACE TEMPERATURE (NOTE THAT ONLY TEMP IS USED)
TSTP05  C  (NOT PROBLEM NOR MATER DEPENDENT)
TSTP06  C
TSTP07        DTOLD=DT
TSTP08        DWDT=(WS/DTOLD
TSTP09        IF(M.EQ.1) GOTO 1
TSTP10        DWDT=ABS(WDT-DWDT)/ABS(WDT)
TSTP11        DT=DT*0.1/DWDT
TSTP12  C  10% OF D(DWDT) TOLERANCE
TSTP13        IF(DT.GT.1.0*DTOLD) DT=1.0*DTOLD
TSTP14        IF(DT.GT.50.0DT1)   DT=50.0DT1
TSTP15        IF(DT.LT.DT1)       DT=DT1
TSTP16        K1=0
TSTP17   10   K1=K1+1
TSTP18        POWER=10.**K1
TSTP19        IF(AINT(DT*POWER).EQ.0) GOTO 10
TSTP20        DT=AINT(DT*POWER+.5)/POWER
TSTP21   1    DWDT=(WDT
TSTP22        IF(M.GT.1) RETURN
TSTP23        DT1=DT
TSTP24        RETURN
TSTP25        END


EOR001        SUBROUTINE ERROR(N)
EOR002  C
EOR003        WRITE (6,1) N
EOR004    1   FORMAT(//31HERROR##  STORAGE EXCEEDED BY ,I4)
EOR005        STOP
EOR006        END


NCL001        SUBROUTINE NUMCAL(W,TNOW,DT,WCH,M,A,FF,LSPUL,N,IFLAG,
NCL002                  1                   DWS,WS,WLAST,MPDE,MGMAX)
NCL003  C
NCL004  C  NUMERICAL CALCULATION OF PARTIAL DIFFERENCIAL EQUATION AT SOME
NCL005  C  TIME STEP 'TNOW' WITH CRANK-NICOLSON FINITE DIFFERENCE METHOD.
NCL006  C  PREDICTOR-CORRECTOR SCHEME FOR THE FIRST GUESS, NEWTON-RAPHSON
NCL007  C  METHOD FOR ITERATION AND GAUSSIAN ELIMINATION METHOD FOR SOLVING
NCL008  C  LINEARIZED EQUATIONS.
NCL009  C  (NOT PROBLEM NOR MATER DEPENDENT)
NCL010  C
NCL011        COMMON/ITER /ITMAX,CRIT
NCL012        DIMENSION W(MGMAX,3,MPDE)
NCL013        DIMENSION A(MGMAX,7)
NCL014        DIMENSION WS(MPDE),WCH(MPDE)
NCL015        DIMENSION WLAST(MPDE)
NCL016        DIMENSION FF(LSPUL)
NCL017  C
NCL018        TOLD=TNOW
NCL019        MRESET=M
NCL020    1   NIT=0
NCL021        T=TOLD+DT/2.
NCL022        TNOW=TOLD+DT/2.
NCL023  C
NCL024  C  PREDICTOR
NCL025        MFLAG=0
NCL026        CALL CRANK(W,T,DT/2.,WCH,M,A,FF,LSPUL,MFLAG,MPDE,MGMAX)
NCL027  C
NCL028  C  CORRECTOR
NCL029        MFLAG=1
NCL030        CALL CRANK(W,TNOW,DT,WCH,M,A,FF,LSPUL,MFLAG,MPDE,MGMAX)
NCL031  C
NCL032        IF(M.GT.1) GOTO 10
NCL033        DO 11 I=1,MPDE
```

```
MCL034          DO 11 J=1,L1
MCL035       11 W(J,2,I)=W(J,3,I)
MCL036  C
MCL037  C  ITERATIONS
MCL038          NFLAG=2
MCL039       99 CONTINUE
MCL040  C
MCL041          DO 12 I=1,NPDE
MCL042       12 CALL NEWTON(W,TNOW,DT,WCH,NIT,N,A,FF,LBPUL,I,NFLAG,NPDE,NOMAX)
MCL043  C
MCL044          I=NPDE
MCL045      100 IF(ABS(W(1,3,I)-W(1,2,I)).GT.CRIT) GOTO 101
MCL046          IF(I.EQ.1) GOTO 102
MCL047          I=NPDE-1
MCL048          GOTO 100
MCL049  C
MCL050  C  BEFORE FURTHER ITERATION, CONVERGENCY IS CHECKED.
MCL051      101 CALL CONVRG(W,NLAST,NIT,M,NRESET,DT,NPDE,NOMAX)
MCL052          IF(NIT.EQ.0) GOTO 1
MCL053          IF(NIT.GT.ITMAX) GOTO 102
MCL054          GOTO 99
MCL055 -C
MCL056  C  ITERATIONS COMPLETED, W(J,3,I) BECOME FINAL SOLUTIONS OF THE
MCL057  C  CURRENT TIME STEP AND ARE TRANSFERED TO W(J,1,I) AND TO OBT
MCL058  C  TO THE NEXT TIME STEP.
MCL059      102 DWS=W(1,3,I)-W(1,1,I)
MCL060          DO 14 I=1,NPDE
MCL061          DO 13 J=1,M
MCL062          W(J,1,I)=W(J,3,I)
MCL063       13 W(J,2,I)=W(J,3,I)
MCL064          J=1
MCL065       14 WS(I)=W(J,1,I)
MCL066  C
MCL067          RETURN
MCL069          END


CNK001          SUBROUTINE CRANK(W,T,DT,WCH,N,A,FF,LBPUL,NFLAG,NPDE,NOMAX)
CNK002  C
CNK003  C  CRANK-NICOLSON FINITE DIFFERENCE
CNK004  C  GAUSSIAN ELIMINATION BACK-SUBSTITUTION
CNK005  C  NOT PROBLEM NOR MATER DEPENT
CNK006  C
CNK007          COMMON/GRID /DX1,EPS,EPS1,EPS2
CNK008          COMMON/DCRAN/M1,M2,N1,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13
CNK009          COMMON C(1)
CNK010          DIMENSION W(NOMAX,3,NPDE)
CNK011          DIMENSION A(NOMAX,2)
CNK012          DIMENSION WCH(NPDE)
CNK013          DIMENSION FF(LBPUL)
CNK014  C
CNK015  C       ****************************************
CNK016  C       BLANK      COMMON      STORAGE      ALLOCATION
CNK017  C           ARRAY  - -   INSTRUCTION -   -   DIMENSION
```

```
CNK018  C       N1  ALPHA   COEFF OF I FROM ELIM        NOMAX*NPDE
CNK019  C       N2  BETA    COEFF OF I+1 FROM ELIM      NOMAX*NP-PA
CNK020  C       N3  B       RECUR LOW AFTER ELIM        NOMAX*NPDE
CNK021  C       N4  BC      BNDRY BNADIENTS             NPDE
CNK022  C       N5  FLNR    BNDRY LNR CORR TERM         NPDE
CNK023  C       N6  FQ      BNDRY LNR ILNR AFT CORR     NPDE
CNK024  C       N7  DDX     SEC CORR CC FNST DIF        NPDE
CNK025  C       N8  A1      COEFF OF TIME DEN TERM      NPDE
CNK026  C       N9  FNAI    PDE SPACE DEPENT TERMS      NPDE
CNK027. C       N10 WS      SUMF VALUE OF W             NPDE
CNK028  C       N11 WJ      W VALUE AT J-TH GRID        NPDE
CNK029  C       N12 WOS     SOLUTION FROM GAUSS.FIM     NOMAX
CNK030  C       ****************************************
CNK031  C
CNK032          M=NFLAG+1
CNK033          J=1
CNK034          DO 10 I=1,NPDE
CNK035          II=(I-1)*NOMAX
CNK036          C(N2+II)=A(J,1)
CNK037       10 C(N1+II-1)=W(J,M,I)
CNK038          CALL BNDRY(C(N1+1),WCH,T,FF,LBPUL,C(N4),W,C(N5),C(N6),
CNK039         1                NPDE)
CNK040          DO 11 I=1,NPDE
CNK041          IM=I-1
CNK042          II=IM*NOMAX
CNK043          C(N3+II)=A(J,2)-C(N3+IM)*DX1
CNK044          C(N8+IM)=C(N8+IM)*DX1
CNK045       11 C(N3+II)=C(N8+IM)
CNK046       1  J=J+1
CNK047          DO 12 I=1,NPDE
CNK048          IM=I-1
CNK049          C(N7+IM)=(A(J,3)*W(J-1,M,I)+A(J,4)*W(J,M,I)+A(J,7)*W(J+1,M,I))/
CNK050                   DX1
CNK051       12 C(N1+IM)=W(J,M,I)
CNK052          CALL ACALC(C(N1),WCH,T,W,C(N7),C(N8),C(N9),DDD,I,NFLAG,NPDE)
CNK053          DO 13 I=1,NPDE
CNK054          IM=I-1
CNK055          FJ=-A(J,1)*W(J-1,1,I)-A(J,2)*W(J,1,I)-A(J,3)*W(J+1,1,I)-A(J,4)*W(J
CNK056         1   12,1,I)-2.*C(N8+IM)*W(J,1,I)/DT+C(N9+IM)*DX1*DD
CNK057          JI=IM*NOMAX+J-1
CNK058          C(N1+JI)=A(J,2)-C(N2+JI-1)*A(J,1)/C(N1+JI-1)-2.*C(N8+IM)/DT*DX1*DD
CNK059         1        2
CNK060          C(N2+JI)=A(J,3)-A(J-1,4)*A(J,1)/C(N1+JI-1)
CNK061       13 C(N3+JI)=FJ-A(J,1)*C(N3+JI-1)/C(N1+JI-1)
CNK062          IF(J.LT.M) GOTO 1
CNK063          IF(NFLAG.LT.0) GOTO 2
CNK064  C  IN THE BEGINNING OF EACH TIME STEP TO DETERMINE THE NEW M VALUE FOR
CNK065  C  THE CURRENT TIME STEP
CNK066          NFLAG=0
CNK067          DO 14 I=1,NPDE
CNK068          II=(I-1)*NOMAX+J-1
CNK069          W(J,1,I)=W(J,2,I)+DT*(W(J,1,I)-W(J,2,I))/DT
CNK070          IF(W(J,1,I).LT.0.) W(J,1,I)=0.
CNK071          W(J,2,I)=W(J,1,I)
```

```
CMK072          IF(J.EQ.NOMAX-1) GOTO 14
CMK073          CON1=(C(N14,J1)*C(N2,J1)-C(N3,J1)*C(N14,J1-1))*C(N20,J1-1)
CMK074     1        *A(J-1,4)/C(N3,J1)
CMK075          CON2=(C(N2,J1)*2.0*A(J,4))/C(N3,J1)*C(N14,J1-1)*A(J-1,4))/C(N3,J1
CMK076     1        -1)
CMK077          IF(ABS(CON1/CON2).GT.0.002) NFLAG=1
CMK078     14   CONTINUE
CMK079          IF(NFLAG.NE.0) GOTO 1
CMK080          RETURN
CMK081   C  BACK SUBSTITUTION
CMK082     2    L1=N12
CMK083          DO 15 I=1,NPDE
CMK084          I1=(I-1)*NOMAX
CMK085          CALL GAUSS(C(N14,I1),C(N2,I1),C(N3,I1),A(1,4),C(N12),N)
CMK086          DO 3 J=1,L1
CMK087     3    W(J,N11,I)=C(N12,J-1)
CMK088     15   CONTINUE
CMK089          RETURN
CMK090          END

MWT001          SUBROUTINE NEWTON(W,T,DT,WCH,NIT,A,A,FF,LBPUL,IEON,NFLAG,NPDE,
MWT002     1                  NOMAX)
MWT003   C
MWT004   C  NEWTON-RAPHSON METHOD
MWT005   C  (NOT PROBLEM NOR WATER DEPENT)
MWT006   C
MWT007          COMMON/GRID /DX1,EPS,EPS1,EP92
MWT008          COMMON/DREMT/N1,N2,N3,N4,N5,N6,N7,N8,N9,N10,N11,N12,N13,N14,
MWT009     1                N15,N16,N17
MWT010          COMMON C(1)
MWT011          DIMENSION W(NOMAX,3,NPDE)
MWT012          DIMENSION A(NOMAX,7)
MWT013          DIMENSION WCH(NPDE)
MWT014          DIMENSION FF(LBPUL)
MWT015   C
MWT016   C  ##################################################
MWT017   C  BLANK      COMMON      STORAGE        ALLOCATION
MWT018   C        ARRAY      ------ DESCRIPTION ------      DIMENSION
MWT019   C  N1   ALPHA   COEFF OF 1 FROM ELIM          NOMAX
MWT020   C  N2   BETA    COEFF OF 1+1 FROM ELIM        NOMAX
MWT021   C  N3   S       RECUR SUM AFTER ELIM          NOMAX
MWT022   C  N4   BC      BNDRY CONDITIONS              NPDE
MWT023   C  N5   FBC     PENTUNRED BNDRY GRAD          NPDE
MWT024   C  N6   FLNR    DUMMY VARIABLE                NPDE
MWT025   C  N7   FO      DUMMY VARIABLE                NPDE
MWT026   C  N8   FFO     DUMMY VARIABLE                NPDE
MWT027   C  N9   DDX     SEC CORR CE FIRST DIFF        NPDE
MWT028   C  N10  DDXO    DDX OF LAST TIME STEP         NOMAX*NPDE
MWT029   C  N11  D2DXO   LAST SEC CORR CE SEC DIFF     NOMAX*NPDE
MWT030   C  N12  A1      COEFF OF TIME DIX TERM        NPDE
MWT031   C  N13  FMAI    PRC SPACE DIFFERT TERMS       NPDE
MWT032   C  N14  WS      SURF VARIN OF W               NPDE
MWT033   C  N15  W1      W VALUE AT J IN NEXT          NPDE

MWT034   C      N16 WNS    SOLUTION FROM GAUSS ELIM    NOMAX
MWT035   C      ##########################################
MWT036   C
MWT037          IF(NIT.GT.0.OR.IEON.GT.1) GOTO 20
MWT038          DO 9 I=1,NPDE
MWT039          IN=I-1
MWT040          J1=IN*NOMAX
MWT041          C(N10+J1)=0.0
MWT042          C(N11+J1)=0.0
MWT043          DO 10 J=2,M
MWT044          J1=(N*NOMAX+J-1
MWT045          C(N10+J1)=(A(J+5)*W(J-1,1,I)+A(J,6)*W(J,1,I)+A(J,7)*W(J+1,1,I))/
MWT046     1                DX1
MWT047     10   C(N11+J1)=A(J,1)*W(J-1,1,I)+A(J,2)*W(J,1,I)+A(J,3)*W(J+1,1,I)
MWT048     1                +A(J,4)*W(J+2,1,I)
MWT049     9    CONTINUE
MWT050   C
MWT051     20   J=1
MWT052          IEM=IEON-1
MWT053          C(N2)=A(J,3)
MWT054          DO 11 I=1,NPDE
MWT055          IN=I-1
MWT056          IF(I.LE.IEM) GOTO 12
MWT057          C(N14+IN)=W(J,2,I)
MWT058          GOTO 11
MWT059     12   C(N14+IN)=W(J,3,I)
MWT060     11   CONTINUE
MWT061          CALL BNDRY(C(N14),WCH,T,FF,LBPUL,C(N4),U,C(N6),C(N7),NPDE)
MWT062          FJ=A(J,2)*W(J,2,IEON)+A(J,3)*W(J+1,2,IEON)+A(J,4)*W(J+2,2,IEON)-C
MWT063     1        (N4+IEM)*DX1
MWT064          C(N14+IEM)=C(N14+IEM)*FJ
MWT065          CALL BNDRY(C(N14),WCH,T,FF,LBPUL,C(N5),FV,C(N8),C(N9),NPDE)
MWT066          C(N14+IEM)=C(N14+IEM)-FJ
MWT067          C(N1-1+J)=A(J,2)-(C(N5+IEM)-C(N4+IEM))/FJ*DX1
MWT068          C(N3-1+J)=-FJ
MWT069     1    J=J+1
MWT070          DO 13 I=1,NPDE
MWT071          IN=I-1
MWT072          IF(I.LE.IEM) GOTO 14
MWT073          C(N15+IN)=W(J,2,I)*W(J,1,I))/2.
MWT074          C(N9+IN)=(A(J,5)*W(J-1,2,I)+A(J,6)*W(J,2,I)+A(J,7)*W(J+1,2,I))/
MWT075     1                DX1
MWT076          GOTO 13
MWT077     14   C(N15+IN)=W(J,3,I)*W(J,1,I))/2.
MWT078          C(N9+IN)=(A(J,5)*W(J-1,3,I)+A(J,6)*W(J,3,I)+A(J,7)*W(J+1,3,I))/
MWT079     1                DX1
MWT080     13   C(N9+IN)=C(N9+IN)+C(N10+IN*NOMAX+J-1))/2.
MWT081          CALL ACALC(C(N15),WCH,T,V,C(N7),C(N12),C(N13),DPMAI,IEON,NFLAG,
MWT082     1                NPDE)
MWT083          D2DX=A(J,1)*W(J-1,2,IEON)+A(J,2)*W(J,2,IEON)+A(J,3)*W(J+1,2,IEON)+
MWT084     1        A(J,4)*W(J+2,2,IEON)
MWT085          FJ=D2DX+C(N11+IEM*NOMAX+J-1)-2.*C(N12+IEM)*C(N1,2,IEON)-
MWT086     1        W(J,1,IEON))/DT-C(N13+IEM)*DX1**2
MWT087          AK=A(J,1)+A(J,5)*DPMAI*DX1
```

```
MMT088        9R=A(J,2)*A(J,A)*(M*M*BX)-2.*C(M*2*(L+1))/STOBX1**2
MMT089        CR=A(J,3)*A(J,2)*(M*M*BX)
MMT090        C(M1-1+J)=BA-C(M*2)-2*AA/C(M1+J-2)
MMT091        C(M2-1+J)=CK-A(J )*4*BA/C(M1+J-2)
MMT092        C(M3-1+J)=-F.J-AA*C(M3+J-2)/C(M1+J-2)
MMT093        IF(J.LT.M) GOTO 1
MMT094  C  BACK-SUBSTITUTION
MMT095        CALL GAUSS(C(M1),C(M2),C(M3),A(1,4),C(M1),M)
MMT096        L1=M+2
MMT097        DO 2 J=1,L
MMT098     2  W(J,3)=EON)=C(M1+J )*W(J,2,+EON)
MMT099        RETURN
MMT100        END

GUS001        SUBROUTINE GAUSS(ALPHA,BETA,B,D,WOS,M)
GUS002  C
GUS003  C  BACK-SUBSTITUTION AFTER GAUSSIAN ELIMINATION
GUS004  C
GUS005        COMMON/GRID /DX1,EPS,EPS1,EPS2
GUS006        DIMENSION WOS(1)
GUS007        DIMENSION ALPHA(1),BETA(1),B(1),D(1)
GUS008  C
GUS009        AM1=ALPHA(M-1)-D(M-1)*EPS
GUS010        BM1=BETA(M-1)*D(M-1)*EPS1
GUS011        AMM=ALPHA(M)*B(M)*EPS1*D(M)*EPS2
GUS012        BMM=-BETA(M)*EPS-D(M)*EPS*EPS1
GUS013        WOS(M)=(B(M)-BMM*S(M-1))/AM1)/(AMM-BMM*BM1/AM1)
GUS014        WOS(M-1)=(S(M-1)-BM1*WOS(M))/AM1
GUS015        WOS(M+1)=WOS(M+1)*S(M)-WOS(M-1)*EPS
GUS016        WOS(M+2)=WOS(M+1)*B(M)-WOS(M)*EPS
GUS017        L=M-1
GUS018        DO 100 J=2,L
GUS019   100  WOS(M-J)=(S(M-J) BETA(M-J)*WOS(M-J+1)-D(M-J)*WOS(M-J+2))/
GUS020     1          ALPHA(M-J)
GUS021        RETURN
GUS022        END

BDY001        SUBROUTINE BNDRY(WS,WCH,FAU,FF,LSPUL,BC,V,FLNR,FO,NPDE)
BDY002  C
BDY003  C  BOUNDARY CONDITIONS
BDY004  C  (PROBLEM DEPENDENT BUT NOT WATER DEPENT)
BDY005  C
BDY006        COMMON/CHART/XCH,TCH
BDY007        COMMON/LASER/E1,UF,K,TPUL,AEFF
BDY008        COMMON/MATP /TH,HSUB,HFUS,EHIES,A3RM
BDY009        COMMON/NUEW /WH(5,5)
BDY010        DIMENSION WS(NPDE),WCH(NPDE),FLNR(NPDE),FO(NPDE)
BDY011        DIMENSION WCH(NPDE)
BDY012        DIMENSION FF(LSPUL)
BDY013  C
BDY014        TEMP=WS(1)*WCH(1)
BDY015        OU=WS(2)*WCH(2)
BDY016        T=FAUBTCH
BDY017        IFLAG=4
BDY018        CALL RFLUX(T,TEMP,OU,FLUX,FLUX1,FLUX2,IFLAG)
BDY019        HVAP=HSUB
BDY020        IF(TEMP.GE.TH) HVAP=HVAP-HFUS
BDY021        C=XCH/(WCH(1)*COWB(TEMP))
BDY022        OV=FLUX*HVAP*C
BDY023        OR=EMISS*(TEMP**4-298.2**4)*5.6686E-12*C
BDY024        OI=SOURCE(T-FF,LSPUL)*C
BDY025        BC(1)=OV+OR-OI
BDY026        FLNR(1)=OV/WS(1)*OR/WS(1)
BDY027        FO(1)=-OI
BDY028        V=WH(3,2)/WH(1,2)*FLUX2/RHO(TEMP)*TCH/XCH
BDY029        BC(2)=(WH(3,2)*FLUX1/(WH(2,1)*RHO(TEMP))-V*XCH/TCH*OU)*XCH/
BDY030       1(DIF(TEMP)*WCH(2))
BDY031        FLNR(2)=BC(2)/WS(2)
BDY032        FO(2)=0.0
BDY033        RETURN
BDY034        END

ACL001        SUBROUTINE ACALC(NJ,WCH,T,V,BDX,A1,PHAI,BPHAI,IEON,NFLAG,NPDE)
ACL002  C
ACL003  C  COEFFICIENTS IN CONDUCTION AND DIFFUSION EQUATIONS
ACL004  C  (PROBLEM DEPENDENT BUT NOT WATER DEPENT)
ACL005  C
ACL006        COMMON/CHART/XCH,TCH
ACL007        COMMON/DIFFZ/CH,CMEX,EL,ELEX
ACL008        COMMON/MATP /TH,HSUB,HFUS,EHIES,A3RM
ACL009        COMMON/AB  /SCON(500),SA1(500),SA3(500)
ACL010        DIMENSION NJ(NPDE),WCH(NPDE)
ACL011        DIMENSION A1(NPDE),DDX(NPDE),PHAI(NPDE)
ACL012  C
ACL013        TEMP=NJ(1)*WCH(1)
ACL014        OU=NJ(2)*WCH(2)
ACL015        IF(IEON.GT.1) GOTO 2
ACL016        IF(TEMP.LT.300.0.OR.TEMP.GE.TH) GOTO 10
ACL017        X1=TEMP/10.0-29.0
ACL018        I=X1
ACL019        A1(1)=SA(I)+(X1-I)*(SA(I+1)-SA(I))
ACL020        A1(1)=A1(1)*XCH*42/TCH
ACL021        A3=SA3(I)+(X1-I)*(SA3(I+1)-SA3(I))
ACL022        A3=A3*WCH(1)
ACL023        GOTO 20
ACL024   10   A1(1)=SPHT(TEMP)/COND(TEMP)*RHO(TEMP)*XCH**2/TCH
ACL025        A3=0.0
ACL026   20   A2=VPAI(1)
ACL027        A4=0.0
ACL028        PHAI(1)=A2*DDX(1)+A3*DDX(1)**2+A4
ACL029        IF(NFLAG.GT.1) GOTO 100
ACL030    2   CONTINUE
ACL031        IF(TEMP.GE.TH) GOTO 30
ACL032        A3=EL/TEMP**2*WCH(1)
ACL033        GOTO 40
```

```
ACL034      30  A3=EH/TEMP8920WCH(I)
ACL035      40  A1(2)=XCH8920/TCH/DIF(TEMP)
ACL036          A2=V8A1(2)
ACL037          A4=0.0
ACL038          PUA1(2)=A2BDDX(2)+A3DDX(2)BBX(I)+A4
ACL039     100  CONTINUE
ACL040          DPHA1=A2*A3/FLOAT(IEND)*DDX(1)B2.
ACL041          RETURN
ACL042          END


PFL001          SUBROUTINE PRFLE(TS,TIME,NX,X,N,NPDE,NDMAX)
PFL002  C
PFL003  C  WRITE AND PLOT PROFILES NX AT TIME
PFL004  C  (NOT PROBLEM NOR MATER DEPENDENT)
PFL005  C  (FORMAT EASY TO MODIFY IF DESIRED)
PFL006  C
PFL007          COMMON/CHART/XCH,TCH
PFL008          COMMON/GRID /DX1,EPS,EPS1,EPS2
PFL009          COMMON/LASER/EI,DP,R,FBL,AEFF
PFL010          DIMENSION WKINDMA(NPDE)
PFL011          DIMENSION XINGMAX)
PFL012  C
PFL013          WRITE (6,100)
PFL014          WRITE (6,101) TS,TIME,EI,DP
PFL015          WRITE (6,102)
PFL016          WRITE (6,103)
PFL017          DX=DX1BXCH
PFL018          X(1)=0.0
PFL019          DO 20 J=2,N
PFL020       20 X(J)=X(J-1)+EPS8(J-2)BDX
PFL021          WRITE (6,104) (X(J),(WK(J,I),I=2,NPDE),J=1,N)
PFL022  C
PFL023      100 FORMAT(1H1,11X,105(1H8))
PFL024      101 FORMAT(1H0,14X,40HTEMPERATURE AND COMPOSITION PROFILES AT MAXIMUM
            1SURFACE TEMP=F7.1,13H DEG AT TIMAX,F7.4,5H MSEC/
            2                 24X,8HFOR II =F7.3,14H (JOULES) OP =E10.3,8H (W/CM2))
PFL027      102 FORMAT(1H0,11X,105(1H8))
PFL028      103 FORMAT(3X,13HX,CM,4X,7HTEMP(K),5X,4HCOMP(K))
PFL029      104 FORMAT(3(F12E15.4,F9.1,F9.4))
PFL030  C
PFL031          RETURN
PFL032          END


PTS001          SUBROUTINE PLOTTS(X,Y,N,IRR)
PTS002  C
PTS003  C  PLOT SURFACE TEMPERATURE VS TIME
PTS004  C
PTS005          DIMENSION SPECS(30),X(1),Y(1)
PTS006          DIMENSION SIDE(3),SIDY(3)
PTS007          DIMENSION BUFEC(90),BUFYC(90)
PTS008  C
PTS009  C  SET XMIN, YDISE (IN INCHES)
PTS010          SPECS(1)=1.5
PTS011          SPECS(2)=1.5
PTS012  C  SET XRIGHT, XLEFT, YTOP, YBOT
PTS013          SPECS(3)=0.5
PTS014          SPECS(4)=0.0
PTS015          SPECS(5)=3000.
PTS016          SPECS(6)=0.0
PTS017  C  SET XLENGTH, YLENGTH (IN INCHES)
PTS018          SPECS(7)=5.0
PTS019          SPECS(8)=5.0
PTS020  C  SET NO. OF DIVISIONS
PTS021          SPECS(9)=10.
PTS022          SPECS(10)=10.
PTS023  C  SPECIFY IGOL
PTS024          SPECS(11)=1.0
PTS025  C  SPECIFY INTERMEDIATE FILE NO.
PTS026          SPECS(12)=99.
PTS027          CALL AXLTL(SPECS)
PTS028  C  SPECIFY NO. OF DATA POINTS
PTS029          SPECS(13)=FLOAT(N)
PTS030  C  SET XSKIPS, YSKIPS
PTS031          SPECS(14)=1.
PTS032          SPECS(15)=1.
PTS033  C  SET FONTB,FONTN,SPACER,ROTATE,FONTMB
PTS034  C  FONT 2 FOR CHARACTERS OTHER THAN GRAPH TITLE
PTS035          SPECS(17)=0.08
PTS036          SPECS(18)=0.12
PTS037          SPECS(19)=0.04
PTS038          SPECS(20)=0.0
PTS039          SPECS(23)=2.0
PTS040  C  SET ZONES
PTS041          SPECS(24)=0.1
PTS042          SPECS(26)=0.05
PTS043  C  ANNOTATE HORIZONTAL AXIS
PTS044          SPECS(28)=2.0
PTS045          SPECS(29)=2.0
PTS046          SPECS(9)=5.0
PTS047          CALL HORLTB(SPECS)
PTS048  C  ANNOTATE VERTICAL AXIS
PTS049          SPECS(28)=1.0
PTS050          SPECS(29)=4.0
PTS051          SPECS(10)=10.0
PTS052          CALL VERLTL(SPECS)
PTS053  C  CONSTRUCT TITLE FOR X-AXIS
PTS054          CALL TITLE(8HTIME (MSEC),8HSECB)
PTS055  C  CONSTRUCT TITLE FOR Y-AXIS
PTS056          CALL TITLE(23HSURFACE TEMPERATURE (K),8HECB)
PTS057  C  CONSTRUCT ANNOTATION OF GRAPH TITLE
PTS058  C  ALSO USE FONT 2
PTS059  C  SPECIFY XSTART, YSTART (INCHES)
PTS060          SPECS(22)=2.5
PTS061          SPECS(23)=4.0
PTS062          XMT=1.0
PTS063          CALL TITLE(XMTXT,27HSURFACE TEMPERATURE VS TIME,8HECB)
```

```
PT5064          SPEC8(23)=SPEC5(23)-0.3
PT5065          CALL TITLE(GRAINT,18HOR EI -      JOULES,SPEC8)
PT5066  C  PLOT DATA POINTS WITH SOLID LINE
PT5067          CALL PTLIT(X,Y,BUF,BUF,SPEC8)
PT5068  C  PLOT LINES FOR TOP AND RIGHT EDGES OF PLOT AREA
PT5069          SIDEX(1)=SPEC5(4)
PT5070          SIDEX(2)=SPEC5(5)
PT5071          SIDEX(3)=SPEC5(3)
PT5072          SIDEX(4)=SPEC5(5)
PT5073          SIDEY(3)=SPEC5(3)
PT5074          SIDEY(3)=SPEC5(4)
PT5075          SPEC5(13)=3.0
PT5076          CALL SEL1LI(SIDEX,SIDEY,SPEC8)
PT5077  C
PT5078  C  FINISH PLOT IF IT IS THE LAST ONE;
PT5079  C  OTHERWISE, CALL NEXT FRAME
PT5080          IF(IFRM.EQ.1) GOTO 1000
PT5081          CALL NEXTFRM(SPEC5)
PT5082          RETURN
PT5083     1000 CALL ODSEND(SPEC5)
PT5084          RETURN
PT5085          END


PTX001          SUBROUTINE PLOTTA(X,Y,N,IFRM)
PTX002  C
PTX003  C  PLOT TEMPERATURE PROFILE VS SPACE
PTX004  C
PTX005          DIMENSION SPEC5(30),X(1),Y(1)
PTX006          DIMENSION SIDEX(1),SIDEY(3)
PTX007          DIMENSION BUF(500),BUF(500)
PTX008  C
PTX009  C  SET XDIST, YDIST (IN INCHES)
PTX010          SPEC5(1)=1.5
PTX011          SPEC5(2)=1.5
PTX012  C  SET XRIGHT, XLEFT, YTOP, YBOT
PTX013          SPEC5(3)=50.
PTX014          SPEC5(4)=0.0
PTX015          SPEC5(5)=5000.
PTX016          SPEC5(6)=0.0
PTX017  C  SET PLWDTH, TEWDTH (IN INCHES)
PTX018          SPEC5(7)=5.0
PTX019          SPEC5(8)=5.0
PTX020  C  SET NO. OF DIVISIONS
PTX021          SPEC5(9)=10.
PTX022          SPEC5(10)=10.
PTX023  C  SPECIFY TOUL
PTX024          SPEC5(11)=1.0
PTX025  C  SPECIFY INTERMEDIATE FILE NO.
PTX026          SPEC5(12)=99.
PTX027          CALL PRELIT(SPEC5)
PTX028  C  SPECIFY NO. OF DATA POINTS
PTX029          SPEC5(13)=FLOAT(N)
PTX030  C  SET XSKIPS, YSKIPS
PTX031          SPEC8(14)=1.
PTX032          SPEC8(15)=1.
PTX033  C  SET FONTB,FONTH,SPACER,ROTATE,FONTNO
PTX034  C  FONT 2 FOR CHARACTERS OTHER THAN GRAPH TITLE
PTX035          SPEC5(17)=0.08
PTX036          SPEC5(18)=0.12
PTX037          SPEC5(19)=0.04
PTX038          SPEC5(20)=0.0
PTX039          SPEC5(21)=2.0
PTX040  C  SET ZONES
PTX041          SPEC5(24)=0.1
PTX042          SPEC5(25)=0.05
PTX043  C  ANNOTATE HORIZONTAL AXIS
PTX044          SPEC5(28)=0.6
PTX045          SPEC5(29)=2.0
PTX046          SPEC5(9)=5.0
PTX047          CALL MODLIB(SPEC5)
PTX048  C  ANNOTATE VERTICAL AXIS
PTX049          SPEC5(28)=1.0
PTX050          SPEC5(29)=4.0
PTX051          SPEC5(30)=10.0
PTX052          CALL MODLIL(SPEC5)
PTX053  C  CONSTRUCT TITLE FOR X-AXIS
PTX054          CALL TITLE(11HX (MICRONS),SPEC8)
PTX055  C  CONSTRUCT TITLE FOR Y-AXIS
PTX056          CALL TITLE(15HTEMPERATURE (K),SPEC8)
PTX057  C  CONSTRUCT ANNOTATION OF GRAPH TITLE
PTX058  C  ALSO USE FONT 2
PTX059  C  SPECIFY XSTART, YSTART (INCHES)
PTX060          SPEC5(22)=3.0
PTX061          SPEC5(23)=5.0
PTX062          BUF=1.0
PTX063          CALL TITLE(BUF,19HTEMPERATURE PROFILE,SPEC8)
PTX064          SPEC5(23)=SPEC5(23)-0.3
PTX065          CALL TITLE(BUF,19HAT MAX SURFACE TEMP,SPEC8)
PTX066          SPEC5(23)=SPEC5(23)-0.3
PTX067          CALL TITLE(BUF,18HFOR EI =      JOULES,SPEC8)
PTX068  C  PLOT DATA POINTS WITH SOLID LINE
PTX069          CALL PTLIT(X,Y,BUF,BUF,SPEC8)
PTX070  C  PLOT LINES FOR TOP AND RIGHT EDGES OF PLOT AREA
PTX071          SIDEX(1)=SPEC5(4)
PTX072          SIDEY(1)=SPEC5(5)
PTX073          SIDEX(2)=SPEC5(3)
PTX074          SIDEX(3)=SPEC5(5)
PTX075          SIDEY(3)=SPEC5(3)
PTX076          SIDEY(3)=SPEC5(4)
PTX077          SPEC5(13)=3.0
PTX078          CALL SEL1LI(SIDEX,SIDEY,SPEC8)
PTX079  C
PTX080  C  FINISH PLOT IF IT IS THE LAST ONE;
PTX081  C  OTHERWISE, CALL NEXT FRAME
PTX082          IF(IFRM.EQ.1) GOTO 1000
PTX083          CALL NEXTFRM(SPEC5)
PTX084          RETURN
```

```
PTX005    1000 CALL QPSEND(SPECS)
PTX006         RETURN
PTX007         END


PRS001         SUBROUTINE PLOTRX(X,Y,N,IFRM)
PRS002   C
PRS003   C  PLOT SURFACE COMPOSITION VS TIME
PRS004   C
PRS005         DIMENSION SPECS(30),X(1),Y(1)
PRS006         DIMENSION SIDEX(3),SIDEY(3)
PRS007         DIMENSION BUFX(500),BUFY(500)
PRS008   C
PRS009   C  SET XDISP, YDISP (IN INCHES)
PRS010         SPECS(1)=1.5
PRS011         SPECS(2)=1.5
PRS012   C  SET XWIDTH, XLEFT, YTOP, YBOT
PRS013         SPECS(3)=8.5
PRS014         SPECS(4)=5.0
PRS015         SPECS(5)=2.0
PRS016         SPECS(6)=1.5
PRS017   C  SET XLENGTH, YLENGTH (IN INCHES)
PRS018         SPECS(7)=5.0
PRS019         SPECS(8)=5.0
PRS020   C  SET NO. OF DIVISIONS
PRS021         SPEC (9)=10.
PRS022         SPEC (10)=10.
PRS023   C  SPECIFY TOOL
PRS024         SPECS(11)=1.6
PRS025   C  SPECIFY INTERMEDIATE FILE NO.
PRS026         SPECS(12)=99.
PRS027         CALL ARLIL1(SPECS)
PRS028   C  SPECIFY NO. OF DATA POINTS
PRS029         SPECS(13)=FLOAT(N)
PRS030   C  SET XUNITS, YUNITS
PRS031         SPECS(14)=1.
PRS032         SPECS(15)=1.
PRS033   C  SET FONTP,FONTN,SPACER,ROTATE,FONTNO
PRS034   C  FONT 2
PRS035         SPECS(17)=0.08
PRS036         SPECS(18)=0.12
PRS037         SPECS(19)=0.04
PRS038         SPECS(20)=0.0
PRS039         SPECS(21)=2.0
PRS040   C  SET ZONES
PRS041         SPECS(24)=0.1
PRS042         SPECS(26)=0.1
PRS043   C  ANNOTATE HORIZONTAL AXIS
PRS044         SPECS(28)=2.0
PRS045         SPECS(29)=2.0
PRS046         SPECS(9)=5.0
PRS047         CALL HORLIN(SPECS)
PRS048   C  ANNOTATE VERTICAL AXIS
PRS049         SPECS(28)=4.0
PRS050         SPECS(29)=5.0
PRS051         SPECS(10)=5.0
PRS052         CALL VERLIN(SPECS)
PRS053   C  CONSTRUCT TITLE FOR X-AXIS
PRS054         CALL TITLEG(INTIME (NSEC),SPECS)
PRS055   C  CONSTRUCT TITLE FOR Y-AXIS
PRS056         CALL TITLEG(YSURFACE COMPOSITION,SPECS)
PRS057   C  CONSTRUCT ANNOTATION OF GRAPH TITLE
PRS058   C  FONT 2 ALSO
PRS059   C  SPECIFY XSTART, YSTART (INCHES)
PRS060         SPECS(22)=3.0
PRS061         SPECS(23)=3.5
PRS062         RULE=1.0
PRS063         CALL TITLEG(RULE,27HSURFACE COMPOSITION VS TIME,SPECS)
PRS064         SPECS(23)=SPECS(23)-0.3
PRS065         CALL TITLEG(RULE,18HFOR EI =    JOULES,SPECS)
PRS066   C  PLOT DATA POINTS WITH SOLID LINE
PRS067         CALL XYLIL1(X,Y,BUFX,BUFY,SPECS)
PRS068   C  PLOT LINES FOR TOP AND RIGHT EDGES OF PLOT AREA
PRS069         SIDEX(1)=SPECS(4)
PRS070         SIDEY(1)=SPECS(5)
PRS071         SIDEX(2)=SPECS(3)
PRS072         SIDEY(2)=SPECS(5)
PRS073         SIDEX(3)=SPECS(3)
PRS074         SIDEY(3)=SPECS(6)
PRS075         SPECS(13)=3.0
PRS076         CALL SLIL1(SIDEX,SIDEY,SPECS)
PRS077   C  FINISH PLOT IF IT IS THE LAST.
PRS078   C  OTHERWISE, CALL NEXT FRAME
PRS079         IF(IFRM.EQ.1) GOTO 1000
PRS080         CALL NXTFRM(SPECS)
PRS081         RETURN
PRS082    1000 CALL QPSEND(SPECS)
PRS083         RETURN
PRS084         END


PRX001         SUBROUTINE PLOTRX(X,Y,N,IFRM)
PRX002   C
PRX003   C  PLOT COMPOSITION PROFILE VS SPACE
PRX004   C
PRX005         DIMENSION SPECS(30),X(1),Y(1)
PRX006         DIMENSION SIDEX(3),SIDEY(3)
PRX007         DIMENSION BUFX(500),BUFY(500)
PRX008   C
PRX009   C  SET XDISP, YDISP (IN INCHES)
PRX010         SPECS(1)=1.5
PRX011         SPECS(2)=1.5
PRX012   C  SET XWIDTH, XLEFT, YTOP, YBOT
PRX013         SPECS(3)=5.0
PRX014         SPECS(4)=0.0
PRX015         SPECS(5)=2.0
PRX016         SPECS(6)=1.5
PRX017   C  SET XLENGTH, YLENGTH (IN INCHES)
```

```
PRX018          SPECS(7)=5.0
PRX019          SPECS(8)=5.0
PRX020  C  SET NO. OF DIVISIONS
PRX021          SPECS(9)=10.
PRX022          SPECS(10)=10.
PRX023  C  SPECIFY TOOL
PRX024          SPECS(11)=1.0
PRX025  C  SPECIFY INTERMEDIATE FILE NO.
PRX026          SPECS(12)=99.
PRX027          CALL ARLIL(SPECS)
PRX028  C  SPECIFY NO. OF DATA POINTS
PRX029          SPECS(13)=FLOATN
PRX030  C  SET XSKIPS, YSKIPS
PRX031          SPECS(14)=1.
PRX032          SPECS(15)=1.
PRX033  C  SET FONTB,FONTH,SPACER,ROTATE,FONTWD
PRX034  C  FONT 2
PRX035          SPECS(17)=0.08
PRX036          SPECS(18)=0.12
PRX037          SPECS(19)=0.04
PRX038          SPECS(20)=0.0
PRX039          SPECS(21)=2.0
PRX040  C  SET ZONES
PRX041          SPECS(24)=0.1
PRX042          SPECS(26)=0.1
PRX043  C  ANNOTATE HORIZONTAL AXIS
PRX044          SPECS(28)=0.0
PRX045          SPECS(29)=2.0
PRX046          SPECS(31)=5.0
PRX047          CALL ADDL10(SPECS)
PRX048  C  ANNOTATE VERTICAL AXIS
PRX049          SPECS(28)=4.0
PRX050          SPECS(29)=5.0
PRX051          SPECS(31)=5.0
PRX052          CALL ADDL10(SPECS)
PRX053  C  CONSTRUCT TITLE FOR X-AXIS
PRX054          CALL TITLE(6,'MOLE FRACTION',SPECS)
PRX055  C  CONSTRUCT TITLE FOR Y-AXIS
PRX056          CALL TITLE(11,'COMPOSITION',SPECS)
PRX057  C  CONSTRUCT ANNOTATION OF GRAPH TITLE
PRX058  C  FONT 2 ALSO
PRX059  C  SPECIFY XSTART, YSTART (INCHES)
PRX060          SPECS(22)=1.0
PRX061          SPECS(23)=5.0
PRX062          ROTE=1.0
PRX063          CALL TITLE(8,'VPCOMPOSITION PROFILE',SPECS)
PRX064          SPECS(23)=SPECS(23)-0.1
PRX065          CALL TITLE(8,'PHAT MAX SURFACE TEMP',SPECS)
PRX066          SPECS(23)=SPECS(23)-0.1
PRX067          CALL TI (8,'AUDIT COMPUTE',1) =    JOULES,SPECS)
PRX068  C  PLOT DATA POINTS WITH CONTOUR LINE
PRX069          CALL PLOTL(XDATA,YDATA,NPTS,SPECS)
PRX070  C  PLOT LINES FOR TOP AND RIGHT EDGES OF PLOT AREA
PRX071          SIDEX(1)=SPECS(1)
PRX072          SIDEX(2)=SPECS(3)
PRX073          SIDEX(3)=SPECS(4)
PRX074          SIDEX(4)=SPECS(5)
PRX075          SIDEX(3)=SPECS(3)
PRX076          SIDEX(3)=SPECS(4)
PRX077          SPECS(13)=3.0
PRX078          CALL SLLIL(SIDEX,SIDEY,SPECS)
PRX079  C  FINISH PLOT IF IT IS THE LAST.
PRX080  C  OTHERWISE, CALL NEXT FRAME
PRX081          IF(IFRM.EQ.1) GOTO 1000
PRX082          CALL NXTFRM(SPECS)
PRX083          RETURN
PRX084     1000 CALL DOSEND(SPECS)
PRX085          RETURN
PRX086          END

PRP001          SUBROUTINE PROP
PRP002  C
PRP003  C  THE MATERIAL PROPERTIES AND LASER CHARACTERISTICS
PRP004  C  (NOT PROBLEM NOR MATER DEPENDT)
PRP005  C
PRP006          COMMON/DIFFE/EH,EHEX,EL,ELEX
PRP007          COMMON/LASER/ELOP,RATPUL,AEFF
PRP008          COMMON/MATP /TM,HSUB,HFUS,EMISB,A3RM
PRP009          COMMON/MOLW /WA(5,5)
PRP010          DIMENSION WA(2)
PRP011  C
PRP012  C  (MAX NO OF COMPONENTS      = 2
PRP013  C   MAX NO OF ATOMS OF EACH
PRP014  C   COMPONENT IN MOLECULE     = 4)
PRP015  C  (FOR CONGRUENT VAPORIZATION, GIVE
PRP016  C   WA(1) = WA AND WA(2) = 0)
PRP017  C
PRP018  C  MATERIAL PROPERTIES
PRP019          READ (5,100) (WA(I),I=1,2)
PRP020          WRITE (6,200) (WA(I),I=1,2)
PRP021          READ (5,*,31) TM,HSUB,HFUS,EMISB,A3RM
PRP022          WRITE (6,201) TM,HSUB,HFUS,EMISB,A3RM
PRP023          READ (5,102) EH,EHEX,EL,ELEX
PRP024          IF(EH.LO.0.) GOTO 10
PRP025          WRITE (6,202) EH,EHEX,EL,ELEX
PRP026  C
PRP027  C  LASER CHARACTERISTICS
PRP028     10   READ (5,110) RATPUL,AEFF
PRP029          WRITE (6,210) RATPUL,AEFF
PRP030  C
PRP031  C  MOLECULAR WEIGHTS
PRP032          DO 1 M=1,5
PRP033          DO 1 N=1,5
PRP034          WMN=M*FLOAT(N-1)*WA(1)+FLOAT(M-1)*WA(2)
PRP035     1    CONTINUE
PRP036  C
PRP037     100 FORMAT(2F10.0)
```

```
PRF038    101 FORMAT(4F10.0,F10.0)
PRF039    102 FORMAT(2F10.0,I10.0)
PRF040    110 FORMAT(F10.0,2F10.0)
PRF041    200 FORMAT(1H0,2X,3HTHE MATERIAL PROPERTIES/
PRF042      1          5X,25HATOMIC WEIGHT OF COMP 1 =,F9.2/
PRF043      2          5X,25HATOMIC WEIGHT OF COMP 2 =,F9.2)
PRF044    201 FORMAT(1H0,4X,3HIN=,F5.0,5X,5HNSUB=,F5.0,5X,5HNFUG=,F5.0,
PRF045      1          5X,4HMISS=,F5.2,5X,3HMAX(ROOM TEMP)=,E10.3)
PRF046    202 FORMAT(1H0,4X,3HIH=,F7.0,5X,5HEHEX=,E9.2,5X,3HEL=,F7.0,
PRF047      1          5X,5HELEX=,E9.2)
PRF048    210 FORMAT(1H0,2X,9HMODEL CHARACTERISTICS/
PRF049      1          5X,2HN=,F5.3,5X,5HTPUL=,E10.4,4H SEC,
PRF050      2          5X,5HAEFF=,E10.4,4H CM2)
PRF051  C
PRF052          RETURN
PRF053          END


TBL001          SUBROUTINE TABLE
TBL002  C
TBL003  C  TABULATED A1, A3 AND COND BELOW MELTING TEMP
TBL004  C  (PROBLEM DEPENT BUT NOT MATER DEPENT)
TBL005  C
TBL006          COMMON/MATP /TM,HSUB,HFUS,EN155,A3RM
TBL007          COMMON/TAB  /SCOND(300),BA1(300),BA3(300)
TBL008  C
TBL009          IS=AINT(TM-300.)/10.)+2
TBL010          DO 1 I=1,IS
TBL011          TEMP=300.+10.*FLOAT(I-1)
TBL012          SCOND(I)=COND(TEMP)
TBL013  C  SPHT CORRECTED WITH MELTING BY TREATING HFUS AS DELTA FUNCTION
TBL014          ARG=(TEMP-TM)/50.3882
TBL015          IF(ARG.LT.20.) GOTO 11
TBL016          CPF=0.0
TBL017          GOTO 12
TBL018   11    CPF=HFUS/(SQR(2.1415)*850.)*EXP(-ARG)
TBL019   12    SA1(I)=RHO(TEMP)*(SPHT(TEMP)+CPF)/SCOND(I)
TBL020          IF(I.EQ.1) GOTO 13
TBL021          SA3(I)=(SCOND(I)-SCOND(I-1))/(10.*SCOND(I))
TBL022          GOTO 1
TBL023  C  THE VALUE OF A3 AT ROOM TEMP (300 K)
TBL024   13    SA3(1)=A3RM
TBL025    1    CONTINUE
TBL026          RETURN
TBL027          END


ITL001          SUBROUTINE INITAL(N,WO,NPDE,NOMAX)
ITL002  C
ITL003  C  INITIAL CONDITIONS
ITL004  C  (PROBLEM DEPENT BUT NOT MATER DEPENT)
ITL005  C
ITL006          DIMENSION W(NOMAX,3,NPDE)
ITL007          DIMENSION WO(NPDE)
ITL008  C
ITL009          READ (5,100) (WO(I),I=1,NPDE)
ITL010          WRITE (6,200)
ITL011          WRITE (6,201) (I,WO(I),I=1,NPDE)
ITL012          DO 1 K=1,NPDE
ITL013          DO 1 I=1,NOMAX
ITL014          W(I,1,K)=1.0
ITL015          W(I,2,K)=1.0
ITL016          W(I,3,K)=1.0
ITL017    1     CONTINUE
ITL018          WRITE (6,300)
ITL019          WRITE (6,201) (I,W(I,1,I),I=1,NPDE)
ITL020  C
ITL021  100 FORMAT(F10.0)
ITL022  200 FORMAT(1H1,40HTHE INITIAL QUANTITY (WITH DIMENSION) OF/)
ITL023  201 FORMAT(1X,22HPARTIAL DIFFERENTIAL EQN NO:I2,2H =,F10.4)
ITL024  300 FORMAT(1H0,40HTHE INITIAL QUANTITY (DIMENSIONLESS) OF/)
ITL025  C
ITL026          RETURN
ITL027          END


CHR001          SUBROUTINE CHAR(WO,WCH,NPDE)
CHR002  C
CHR003  C  DEFINE THE CHARACTERISTIC TERMS
CHR004  C  (PROBLEM DEPENT AND MATER MATER DEPENT)
CHR005  C
CHR006          COMMON/CHART/XCH,TCH
CHR007          COMMON/LASER/EI,DF,R,TPUL,AEFF
CHR008          COMMON/MATP /TM,HM(4)
CHR009          DIMENSION WCH(NPDE),WO(NPDE)
CHR010  C
CHR011  C  THERMAL DIFFUSIVITY AT MELTING TEMPERATURE
CHR012          TDIFM=COND(TM)/(RHO(TM)*SPHT(TM))
CHR013  C
CHR014          TCH=TPUL
CHR015          XCH=SQRT(TDIFM*TCH)
CHR016          DO 1 I=1,NPDE
CHR017    1     WCH(I)=WO(I)
CHR018          WRITE (6,100)
CHR019          WRITE (6,200) TCH,XCH,(I,WCH(I),I=1,NPDE)
CHR020  C
CHR021  100 FORMAT(1H2,34HTHE CHARACTERISTIC QUANTITIES ARE:/)
CHR022  200 FORMAT(1X,29HCHARACTERISTIC TIME          =,E10.4,4H SEC/
CHR023      1       1X,29HCHARACTERISTIC DISTANCE     =,E10.4,4H  CM/
CHR024      2       (1X,22HCHARAC QUANTITY OF PDE NO:I2,2H =,F10.4))
CHR025          RETURN
CHR026          END


MFX001          SUBROUTINE MFLUX(TIME,T,R,F,F1,F2,IFLAG)
MFX002  C
MFX003  C  CALCULTION OF THE EVAPORATED MASS FLUXES (G/CM2-SEC)
MFX004  C  (PROBLEM AND MATERIAL DEPENDENT)
```

```
MF005  C
MF006        COMMON/HOLM /WM(5,5)
MF007        DIMENSION P(5,5)
MF008  C
MF009        DO 9 N=1,5
MF010        DO 9 M=1,5
MF011     9  P(N,M)=0.
MF012        CALL ROOT(T,R,P(3,1),P(2,1),P(1,2),P(2,2),P(3,2),P(4,2))
MF013        PIOT=0.
MF014        S0=0.
MF015        S1=0.
MF016        S2=0.
MF017        DO 10 N=1,5
MF018        DO 10 M=1,5
MF019        IF(N.EQ.1.AND.M.EQ.1) GOTO 10
MF020        PIOT=PIOT+P(N,M)
MF021        S0=S0+SQRT(WM(N,M))*P(N,M)
MF022        S1=S1+FLOAT(N-1)*P(N,M)/SQRT(WM(N,M))
MF023        S2=S2+FLOAT(M-1)*P(N,M)/SQRT(WM(N,M))
MF024   10   CONTINUE
MF025        C=44.2/SQRT(T)*S0.S2
MF026  C  44.2 - 3.5E229760/4.023E23 - FLUX IN O/CM2-SEC
MF027        F=C*S0
MF028        F1=C*WM(2,1)*S1
MF029        F2=C*WM(1,2)*S2
MF030        IF(IFLAO.LT.1.OR.IFLAO.GT.3) RETURN
MF031        GOTO (1,2,3),IFLAO
MF032  C
MF033    1   CONTINUE
MF034        WRITE (6,100)
MF035        WRITE (6,101) P(4,2),P(3,2),P(2,2),P(1,2),P(3,1),P(2,1),PIOT
MF036        RETURN
MF037  C
MF038    2   CONTINUE
MF039        WRITE (6,200)
MF040  C
MF041    3   CONTINUE
MF042        WRITE (6,300) TIME,T,R,P(4,2),P(3,2),P(2,2),P(1,2),P(3,1),P(2,1),
MF043       1              PIOT,F
MF044  C
MF045  100 FORMAT(1H2,6X,46HTHE EQUILIBRIUM VAPOR PRESSURES AT MAXIMUM SURFAC
MF046       1E TEMPERATURE ARE//
MF047       2          10X,5HP-UO3,5X,5HP-UO2,6X,4HP-UO,7X,3HP-U,6X,4HP-O2,
MF048       3          7X,3HP-U,3X,7HP-TOTAL)
MF049  101 FORMAT(1H ,4X,7(E10.3))
MF050  200 FORMAT(1H ,5X,4HT(K),3X,5HT5(K),6X,3HQ/U,5X,5HP-UO3,5X,5HP-UO2,
MF051       1          4X,4HP-UO,7X,3HP-U,6X,4HP-O2,7X,3HP-O,
MF052       2          3X,7HP-TOTAL,7X,4HFLUX)
MF053  300 FORMAT(1H ,E11.2,F8.0,F9.4,7(E10.3),E11.3)
MF054  C
MF055        RETURN
MF056        END
```

```
UO2001        SUBROUTINE ROOT(TEMP,DU,PO2,PD,PU,PUO,PUO2,PUO3)
UO2002  C
UO2003  C  CALCULATION OF PARTIAL VAPOR PRESSURES OF URANIUM-OXYGEN
UO2004  C  SYSTEM BASED ON BLACKBURN'S MODEL.
UO2005  C  MODIFIED FROM SUBROUTINE DEVELOPED BY BLACKBURN FOR MIXED OXIDE
UO2006  C
UO2007        COMMON/CC/A1,A2,A3,A4,A5,A6,A7
UO2008  C
UO2009        Y=DU-2.0
UO2010        IF(Y.LT.0.) Y=0.0
UO2011        DOH=SQRT(DU)
UO2012        DU2=DU**2
UO2013        TM=3133.0
UO2014  C
UO2015  C  THERMODYNAMIC DATA
UO2016        IF(TEMP.GT.TM) GOTO 60
UO2017        DD=EXP(78260./TEMP-13.6)
UO2018        EE=EXP(16350./TEMP-4.96-54.08*Y)
UO2019        FF=EXP(47000./TEMP-11.25)
UO2020        GG=EXP(25100./TEMP-4.92)
UO2021        CUC=EXP(-187200./TEMP+33.6)
UO2022        CUDC=EXP(-128700./TEMP+27.3)
UO2023        CUO2C=EXP(-72600./TEMP+20.3)
UO2024        CUO3C=EXP(-28500./TEMP+10.5)
UO2025        GOTO 70
UO2026   60   DD=EXP(78260.-1.518TM)/TEMP-12.066)
UO2027        EE=EXP((16500.-1.06*TM)/TEMP-4.19-54.08*Y)
UO2028        FF=EXP((47000.-.7558TM)/TEMP-10.455)
UO2029        GG=EXP((25100.-.7558TM)/TEMP-4.065)
UO2030        CUC=EXP(-167900./TEMP+27.3)
UO2031        CUDC=EXP(-108600./TEMP+20.8)
UO2032        CUO2C=EXP(-51500./TEMP+13.4)
UO2033        CUO3C=EXP(-5100./TEMP+2.8)
UO2034  C
UO2035   70   DEF=DD*EE*FF
UO2036        FD=FF*DD
UO2037        HH=DEF*DU/DU2*(3.0/DU-1.0)
UO2038        PP=DEF/DU*81.54*(3.0/DU-1.0)
UO2039        QQ=DU/DOH*(1.0*EE/DU*2.0*FD/DU*DD/DU*(-EE-FD)
UO2040        RR=DD*FF/DOH*(2.0/DU-1.0)
UO2041        SS=FG/DU*DD*(2.0/DU)*(-DD-FD)
UO2042        TT=(1.0/DOH-DOH)*FF
UO2043        A1=HH
UO2044        A2=PP
UO2045        A3=QQ
UO2046        A4=RR
UO2047        A5=SS
UO2048        A6=TT
UO2049        A7=1.0-DU
UO2050  C
UO2051        G=0.0
UO2052        GTEST=0
UO2053        DO 40 I=1,40
UO2054        G=G+A1*G**4+A2*G**3+A3*G**2+A4*G+A5
```

```
      DO 10 J=1,5
      FM=F(G1)
      FB=FP(G1)
      G5=G1-FM/FB
      G1=G5
      IF(G1.LT.0.0) G1=10.0
   10 CONTINUE
      D2=U1*G1
      G=G1
      U2=1.0/(1.04D2*DD/DU+D2*D2*DD*EE/DU2)
      U4=U2*D2*DD/DU
      U6=U4*D2*EE/DU
      DM=U2*2.0D*43.0*U6
      EREL0=(DU-DM)/DU
      TESTD=ABS(EREL0)
      IF(TESTD.LT.1.0E-6) GOTO 50
      GOTO 20
   50 IF(D.LT.0.0) GOTO 40
      DMCD=DM
      GOTO 100
   20 CONTINUE
   40 PRINT 2
    2 FORMAT(90X,21HROOT DID NOT CONVERGE)
  100 FD=D*4*4
      FD=SQRT(FD)*EXP(-30103./TEMP*8.036)
      DU=U4/FD2*DUC
      FUD=U4/SQRT(FD)*CUDC
      FUD2=U4*SQR(FD)*CUD3C
      RETURN
      END

      FUNCTION F(X)
      COMMON/CC/A1,A2,A3,A4,A5,A6,A7
      F=(((((A1*X+A2)*X+A3)*X+A4)*X+A5)*X+A6)*X+A7
      RETURN
      END

      FUNCTION FP(X)
      COMMON/CC/A1,A2,A3,A4,A5,A6,A7
      FP=(((6.0*A1*X+5.0*A2)*X+4.0*A3)*X+3.0*A4)*X+2.0*A5)*X+A6
      RETURN
      END

      FUNCTION RHO(TEMP)
C
C  DENSITY OF UO2 IN G/CM3
C  GIVEN BY CHASANOV(1973) AND LEIBOWITZ(1974)
C
      DATA A,B,C,D/9.0E-6,6.0E-9,3.0E-12,10.98/
      DATA T1,A1,B1/3133.,10.658,-0.1609E-4/
      IF(TEMP.GE.T1) GOTO 1
      T=TEMP-273.
      RHO=D/(1.+A*T+B*T*T+C*T**3)
      RETURN
    1 RHO=A1+B1*TEMP
      RETURN
      END

      FUNCTION SPHT(TEMP)
C
C  SPECIFIC HEAT OF UO2 IN J/B-K
C  GIVEN BY CHASANOV(1973) AND LEIBOWITZ(1973)
C
      DATA A,C,D,F/24.56,-4.7437E-3,1.6725E-6,6.1981E+5/
      DATA T1/3133./
      T=TEMP
      IF(T.GE.T1) GOTO 1
      CP=B*2.*T*C+3.*D*(T**2)-F*(1./T**2)
      SPHT=4.186*CP/270.
      RETURN
    1 SPHT=0.303
      RETURN
      END

      FUNCTION CONB(TEMP)
C
C  THERMAL CONDUCTIVITY OF UO2 IN W/CM-K
C  GIVEN BY SCHMIDT(1971)
C
      DATA A,B/10.8D,0.0218/
      DATA X,CVD,T0/2.00E-5,299,2050./
      DATA T1,T2/1200.,2860./
      T=TEMP-273.
      IF(T.GE.T1) GOTO 1
      CONB=1./(A+B*T)
      RETURN
    1 IF(T.GE.T2) GOTO 2
      F=(SPHT(TEMP)-X*T-CVD)/CVD
      Y=((T*T0)-(T-T0)*TANH((T-T0)))/2.
      CONB=(1./(A+B*T))*(1.+F)
      RETURN
    2 CONB=0.037
      RETURN
      END

      FUNCTION DIF(TEMP)
C
C  OXYGEN DIFFUSIVITY IN UO2 IN CM2/SEC
C  COEFFICIENTS READ FROM SUBROUTINE PROP
C
      COMMON/DIFFE/EH,EHEX,EL,ELEX
```

```
W02151        IF(TEMP.GE.3133.) GOTO 1
W02152        DIF=LIL*EXP(-EL/TEMP)
W02153        RETURN
W02154    1   DIF=LHK*EXP(-EH/TEMP)
W02155        RETURN
W02156        END
```

B.3 User's Manual

B.3.1 Program Input Data Cards

(1) PROGRAM INITIATION AND TITLE (2(A5,5x),6A5,5x,2A5) - One card, read in subroutine TLCDE.

Column	Variable	Description
1-5	MODE	Problem initiation flag; "START" for initiation and "STOP" for termination
6-10	--	Blank
11-15	MTYPE	Problem type; "SURFT" for heat conduction only and "STAR" for conduction and diffusion
16-20	--	Blank
21-50	HED	Problem title for labeling output (an array of dimension 6)
51-55	--	Blank
56-65	DATE	Date of the run

(2) DIMENSION SPECIFICATION (2I5) - One card, read in subroutine TLCDE.

Column	Variable	Description
1-5	NPDE	Number of partial differential equations to solve; 1 for "SURFT" and 2 for "STAR"
6-10	NGMAX	Estimated maximum number of grid points needed for space variable (normally 300)

(3) MATERIAL PROPERTIES - read in subroutine PROP.

(i) ATOMIC WEIGHTS (2F10.0)

Column	Variable	Description
1-10	WA(1)	Atomic weight of component 1 (g/g-atom)
11-20	WA(2)	Atomic weight of component 2 (g/g-atom); zero or blank for single component materials or congruently vaporizing materials

(ii) OTHER PROPERTIES (4F10.0,E10.0)

Column	Variable	Description
1-10	TM	Melting temperature (K)
11-20	HSUB	Heat of sublimation of solid phase (J/g)
21-30	HFUS	Heat of fusion (J/g)
31-40	EMISS	Total normal optical emissivity
41-50	A3RM	Coefficient A3 (1/k dk/dT) of heat conduction equation at room temperature

(iii) DIFFUSION COEFFICIENT (2(F10.0,E10.0)) - Blank for "SURFT" when no diffusion process is engaged.

Column	Variable	Description
1-10	EH	Diffusion activation energy / Gas constant of the mobile component above some temperature (normally melting point)
11-20	EHEX	Pre-exponential factor of diffusion coefficient of the mobile component above some temperature
21-30	EL	Diffusion activation energy / Gas constant of the mobile component below some temperature
31-40	ELEX	Pre-exponential factor below the temperature

(4) LASER PARAMETERS (F10.0,E10.0,F10.0) - One card, read in subroutine PROP.

Column	Variable	Description
1-10	R	Reflectivity of target material to the laser light
11-20	TPUL	Effective laser pulse width (sec)
21-30	AEFF	Effective surface area (cm^2) of laser exposure spot

(5) CONTROL CARDS FOR NUMERICAL STABILITY - Two cards, read in subroutine SIZE.

(i) STEP AND GRID SIZES (2E10.0,F10.0)

Column	Variable	Description
1-10	DT1	First time step size (dimensionless); also the lower limit of the following step sizes
11-20	DX1	First space grid size (dimensionless); grid sizes are geometrically increasing
21-30	EPS	Geometric factor for increasing grid sizes; a constant greater than 1 (normally 1.035)

(ii) ITERATION CONTROL (I10,E10.0)

Column	Variable	Description
1-10	ITMAX	Maximum number of iterations allowed
11-20	CRIT	Error tolerance for iteration termination

(6) TABULATED LASER PULSE

(i) SIZE AND DIVISION OF TABULATION - One card, read in subroutine SHAPE1.

Column	Variable	Description
1-10	SDT	Step size of tabulation
11-20	LSPUL	Number of divisions of tabulation

(ii) NORMALIZED LASER PULSE SHAPE (7F10.0) - As many cards as needed to specify the tabulated normalized pulse shape (LSPUL/7 or LSPUL/7+1), read in subroutine SHAPE2.

Column	Variable	Description
1-10	SS(I)	Normalized digital pulse shape
...	...	. . .

(7) TERMINATION OF STEP DO LOOP (2E10.0) - One card, read in the main program.

Column	Variable	Description
1-10	TSTOP	Time to stop the time step do loop (sec)
11-20	TCYCL	Time of a cycle for repetitive pulse (sec); Default (if zero or blank) for single pulse source is set to 1.e10 which is supposed to be approximately infinitive

(8) INCIDENT TOTAL ENERGY (F10.0) - One card, read in the main program.

Column	Variable	Description
1-10	EI	Incident total laser energy (J)

(9) INITIAL CONDITIONS (F10.0) - NPDE cards, read in subroutine INITAL.

Column	Variable	Description
1-10	WO(I)	Initial condition for I-th partial differential equation; e.g. WO(1) - initial temperature and WO(2) - initial composition

(10) RESTART OR TERMINATION CARD (I1) -One card, read in the main program.

Column	Variable	Description
1	ICTL	Control character; 0 or blank: STOP 1: same material, another run for different laser energy 2: different material, start from the right beginning

B.3.2 Subroutines to be Supplied by the User

(1) BNDRY: The boundry condition(s) of the problem. Input the surface value(s) of the function(s) and output the surface gradient(s) of the function(s).

(2) ACALC: The coefficients of the partial differential equation(s) and the form of function(s) PHAI (ψ_T in eq. (1) or/and ψ_r in eq. (2)).

(3) TABLE: To tabulate the coefficients of the partial differential equation(s).

(4) INITAL: The initial condition(s).

(5) CHAR: The characteristic quantities for the non-dimensionalization of the boundary value problem.

(6) MFLUX: The calculation of the convective term (due to moving boundry) and the surface heat loss due to surface recession. Also, some printout formats are to be provided.

(7) A function library providing the physical properties, such as RHO (density), SPHT (specific heat), COND (thermal conductivity), DIF (diffusion coefficient), etc.

B.4 Sample Analyses for UO_2 Vaporization

Fig. B.1 shows an example of the input deck of the program "STAR" with total laser energy of 30 Joules and the digital pulse shape measured by the photodiode. The physical properties used in the program are provided from Appendix C.

B.4.1 Program Test with Analytic Solutions

The program was tested by comparing the results with the analytic

Fig. B.1 An Example of The Input Deck for "STAR"

```
start     star      uo2 surf temp and comp calc          12/03/80
  2  300
     16.      238.
   3133.     2234.      275.      0.83  -.190e-2
   5033.  2.22e-02    28550.   .115e01
    0.05 1.9208e-4  4.964e-1
 1.0e-03     .2e-1     1.035
       5   1.0e-4
 1.0e-5        51
    0.0     0.275     0.691     0.851     0.926     0.980     0.996
  1.000     0.990     0.970     0.925     0.890     0.845     0.820
  0.760     0.710     0.670     0.618     0.575     0.525     0.485
  0.440     0.400     0.364     0.330     0.298     0.270     0.238
  0.214     0.190     0.170     0.145     0.126     0.105     0.092
  0.078     0.063     0.050     0.041     0.030     0.025     0.017
  0.010     0.004       0.0       0.0       0.0       0.0       0.0
    0.0       0.0
 0.25e-3
   3.e1
  1400.
   2.00
```

solutions assuming no ablation of the surface (v=0, j=0), no radiation heat loss (ϵ=0) and constant properties ρ, C_p and k (at 4000K) when analytic solutions are possible.

The heat source was assumed either rectangular shape with pulse width equal to t_{pul} or triangular shape with total pulse width equal to $2t_{pul}$ and maximum at t_{peak}.

The test runs assumed total energy of E = 30 Joules, t_{pul} = 1.9208×10^{-4} sec, and $t_{peak} = 0.7 \times 10^{-4}$ sec.

Without ablation and radiation heat loss and with constant properties, the heat conduction can be solved analytically for rectangular and triangular surface heat sources:[14]

(i) Rectangular:

$$T_s(t) - T_o = \frac{2q}{k}\left(\frac{Kt}{\pi}\right)^{1/2} \qquad 0<t<t_{pul}$$

$$= \frac{2q}{k}\left(\frac{K}{\pi}\right)^{1/2}\left[t^{1/2}-(t-t_{pul})^{1/2}\right] \qquad t>t_{pul} \qquad \text{(B-1)}$$

(ii) Triangular:

$$T_s(t) - T_o = \frac{4q}{3k}\left(\frac{K}{\pi}\right)^{1/2}\left[\frac{t^{2/3}}{t_1}\frac{t_2}{t_1}\frac{(t-t_1)^{2/3}}{t_2-t_1}H(t-t_1) + \frac{(t-t_2)^{2/3}}{t_2-t_1}H(t-t_2)\right] \qquad \text{(B-2)}$$

where $H(t-t_i) = 1 \qquad t>t_i$

$\qquad\qquad\qquad = 0 \qquad t<t_i$

where q is the surface heat flux (W/cm^2), k is the thermal conductivity, and K is the thermal diffusivity ($k/\rho C_p$).

Figs. B.2 and B.3 show the results of the calculation. In both of the figures, the solid lines mean the analytic solutions and the dashed lines, which fall almost exactly on the solid lines, supposely are the numerical solutions. This means that the numerical solutions have very good agreement with the analytic solutions at least when vaporization process and radiation are not significant and properties are not temperature dependent.

B.4.2 Numerical Solutions

Figs. B.4, B.5, B.6 and B.7 show the results of the "STAR" computer run in which the incident laser total energy is 30 Joules with the pulse shape shown in Fig. B.8, and the effective laser spot area is 0.4964 cm^2.

In Figs. B.4 and B.6, there also show the results of the "SURFT" computer run which assumed no composition change so that only heat conduction equation is solved and the laser has the same characteristics as

the run in "STAR".

B.4.3 Discussions

The difference of the maximum surface temperature from "STAR" and "SURFT" is about 200K, which is mainly due to the vaporization effect in the ablation terms and the fact that the rate of vaporization is not only a function of temperature but also a function of the composition.

Fig. B.9 shows the maximum surface temperature as a function of the total incident laser energy calculated from both "STAR" and "SURFT". At low energy (or low temperature) range where vaporization effect is relatively unimportant, the two calculations give the consistent results, while at high energy, the discrepancy as much as 200 - 300 K attributes to the ablation effect.

B.5 Other Applications:

The computer program has been designed to deal with (i) thermal conduction problem only ("SURFT") or diffusion process at constant temperature only ("SURFR") or both ("STAR"), (ii) same kind of problems with different materials and (iii) different kind of problems (e.g. different differential equations, boundary conditions, or/and different type of heat sources - laser or ion beam, surface or volumetric source) with different materials.

For instance, the possible applications might be the thermal analysis of the first wall in CTR, the thermal analysis of laser driven micro- explosion, the thermal analysis of electron beam heating, the thermal and mass transport analysis of laser annealing or electron beam

annealing in semiconductor ion implantation and other applications of pulse heating techniques in material processing, etc.

In this report, the thermal analysis of the first wall design is presented and compared with the current design based on approximate analytic solutions.

B.5.1 Problem Formulation

Consider the first wall design of an Inertial Confinement Fusion Reactor with proposed nuclear grade H-451 graphite.[38-40] The heat loading on the first wall has been asaumed consisting of 4000 MJ total with 2% in x-rays, 30% in debris and the remainder in neutrons. The debris pulse has been taken as 500 μs duration which may be treated as a surface heat flux. The x-ray, which occurs long before the debris pulse, is assumed to be uniformly deposited on the first wall over a time of a few microseconds and produce initial temperature of 840K before the debris pulse. The neutron flux which penetrates mostly into the bulk will deposite only a small amount of energy in the first wall and is neglected here[39].

B.5.2 Analytic Solutions

Assuming no ablation, no radiation heat loss and constant properties, the analytic solution for a rectangular shape of 500 μsec duration pulse is given in eq. (B-1) with t_{pul} equal to 500 μs. The thermal conductivity is taken as constant at temperature of 1300K (0.58 W/cm-K for unirradiated graphite and 0.34 W/cm-K for irradiated one), the density is 1.74 g/cm^3, the heat capacity is 1.89 J/g-K for unirradiated and irradiated graphite, and t is the time from the beginning of the debris

pulse.

Figs. B.10 and B.11 show the surface temperatures as a function of time for three sizes of radius (12m, 13m and 14m) for unirradiated and irradiated graphite.

The thermal evaporation rate is approximated by

$$\epsilon_R = 2.14 \times 10^{12}\ T_s^{-1/2} \exp(-92{,}000/T) \qquad \text{mm/sec} \qquad \text{(B-3)}$$

where T_s is the surface temperature given by eq. (B-1).

The total erosion rate per second is obtained by numerically integrating eq. (B-3) assuming one second pulse repetition rate. Fig. B.12 shows the erosion rate per year with first wall radius for unirradiated and irradiated graphite.

B.5.3 Numerical Solutions

The computer code "SURFT" has been applied to this problem with ablation and radiation heat loss and vaiable properties. The properties are given from Ref. 39. Figs. B.13 and B.14 show the surface temperatures as a function of time for three size of radius (12m, 13m and 14m) for unirradiated and irradiated graphite.

The erosion rate is calculated along with the temperature evolution in "SURFT", and the result is shown in Fig. B.15 for unirradiated and irradiated graphite.

B.5.4 Discussions

To compare the numerical solutions with the analytic solutions, Table B.1, B.2, Figs. B.16, B.17, B.18 and B.19 show the maximum surface

temperatures and erosion rate for different reactor radius design.

Given the erosion rate of 5 mm/yr as the design criterion of first wall, the analytic calcualtion by Hopkins[39] gives radius of 12.54 meters for unirradiated H-451 graphite and 14.35 meters for irradiated H-451 garaphite, while the numerical calculation of this work gives radius of 13.38 meters for unirradiated one and 13.35 meters for irradiated one. Figs. B.18 and B.19 show so high sensitivity of the erosion rate to the radius that a difference of 1 meter in radius may result in almost 2 orders of magnitude difference in erosion rate, which leads to 2 orders of magnitude difference in the first wall thickness with same amount of life time. Therefore, it is concluded that the accurate calculation of thermal analysis is necessary.

It is assumed here that the vaporization is the major factor of material erosion and the erosion rate is the design criterion for first wall; however, it has been pointed out that the thermal stress due to the high thermal gradient may impose another design limitation.[41] After all, the purpose of this report is to show a useful application of the computer code developed here and to compare the results with the analytic solutions.

TABLE B.1: Erosion Rate of Unirradiated H-451 Graphite

RADIUS (meters)	T_s^{max} (K) Analytic	T_s^{max} (K) Numerical	$\dot{\epsilon}$ (mm/yr) Analytic	$\dot{\epsilon}$ (mm/yr) Numerical
10	4289	4314	3.98×10^{4}	9.15×10^{4}
12	3223	3766	3.43×10^{1}	1.50×10^{3}
13	2865	3227	0.988	24.7
14	2580	2814	2.94×10^{-2}	0.384
15	2351	2499	9.30×10^{-4}	6.43×10^{-3}
16	2163	2264	3.19×10^{-5}	1.45×10^{-4}
18	1877	1911	5.15×10^{-8}	8.61×10^{-8}
20	1672	1661	1.35×10^{-10}	6.48×10^{-11}

TABLE B.2: Erosion Rate of Irradiated H-451 Graphite

RADIUS (meters)	T_s^{max} (K) Analytic	T_s^{max} (K) Numerical	$\dot{\epsilon}$ (mm/yr) Analytic	$\dot{\epsilon}$ (mm/yr) Numerical
10	5357	4016	2.83×10^{6}	1.32×10^{4}
12	3965	3387	6.93×10^{3}	1.37×10^{2}
13	3497	3110	3.16×10^{2}	11.8
14	3125	2870	14.1	0.985
15	2825	--	0.635	--
16	2580	2466	2.94×10^{-2}	5.01×10^{-3}
18	2207	2168	7.35×10^{-5}	2.94×10^{-5}
20	1939	--	2.46×10^{-7}	--

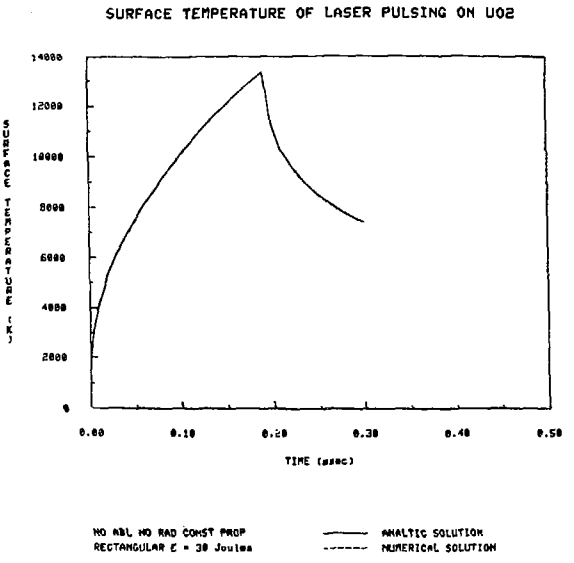


Fig. B.2 The Surface Temperature UO_2 with Time for Rectangular Surface Heat Source with No Ablatron, No Radiation Heat Loss and Constant Properties

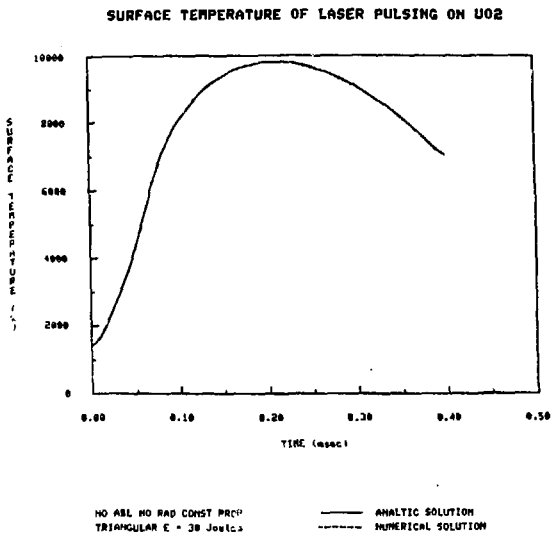


Fig. B.3 The Surface Temperature of UO_2 with Time for Triangular Surface Heat Source with No Ablatron, No Radiation Heat Loss and Constant Properties

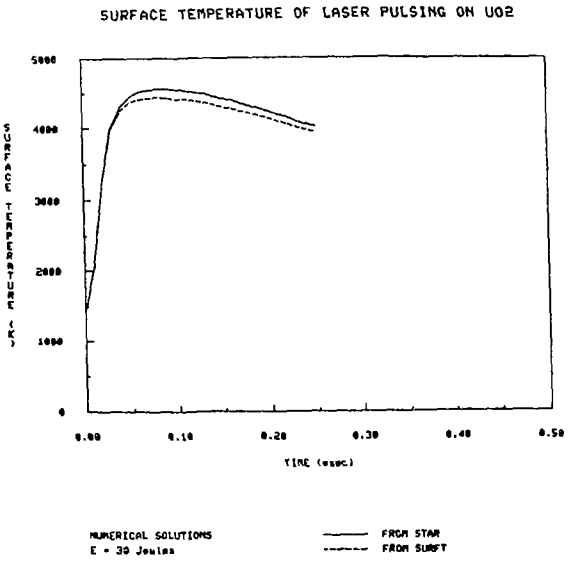


Fig. B.4 The Surface Temperature of UO_2 with Time for Real Laser Pulse with Ablatron, Radiation Heat Loss and Variable Properties

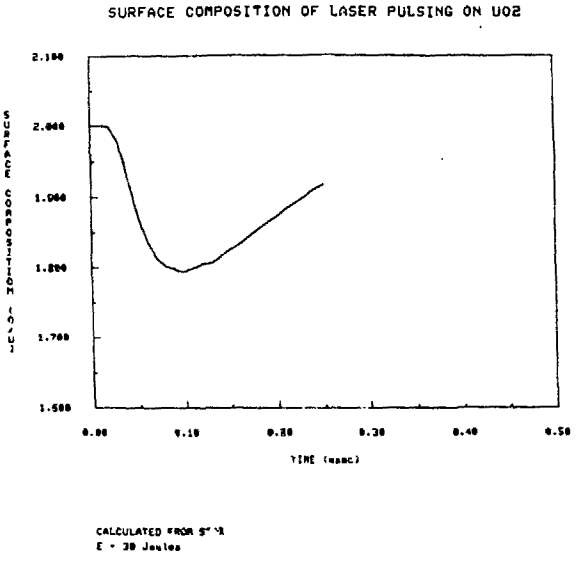


Fig. B.5 The Surface Composition of UO_2 with Time for Real Laser Pule with Incongruent Vaporization

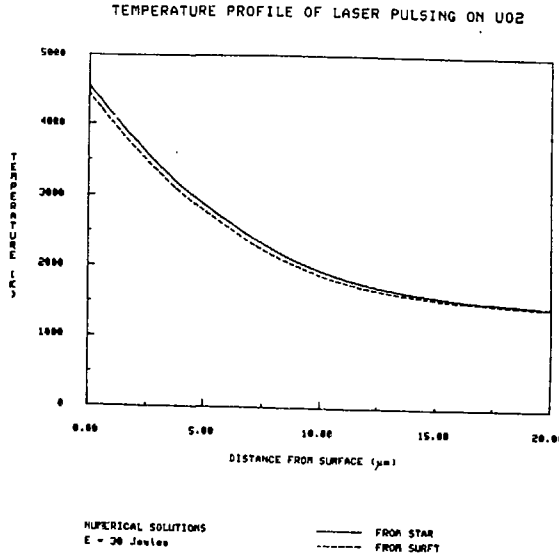


Fig. B.6 The Temperature Profile of UO_2 Subject to Total Energy 30 Joule Laser Surface Heating

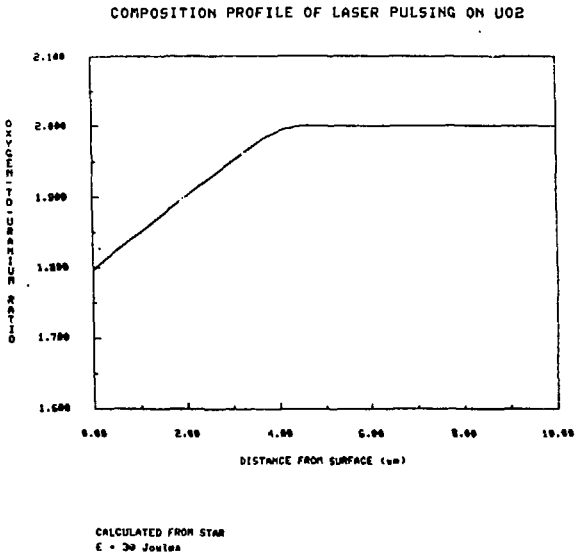


Fig. B.7 The Oxygen-to-Uranium Ratio Profile of UO_2 Subject to Total Energy 30 Joule Laser Surface Heating with Incongruent Vaporization

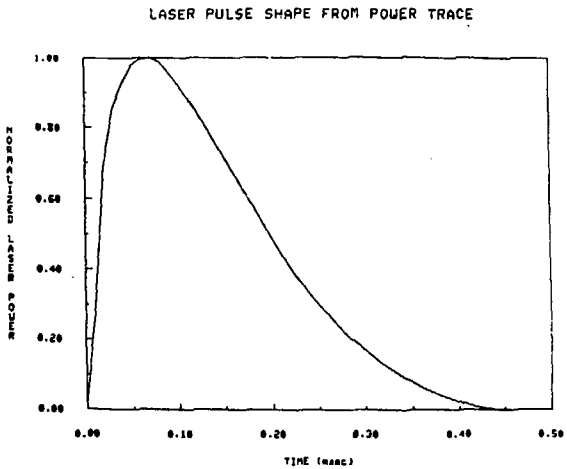


Fig. B.8 The Normalized Laser Pulse Shape With Time

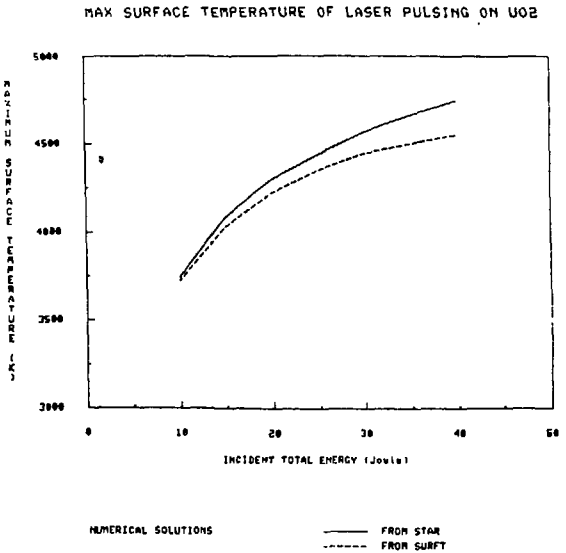


Fig. B.9 The Maximum Surface Temperature of UO_2 vs Incident Total Energy

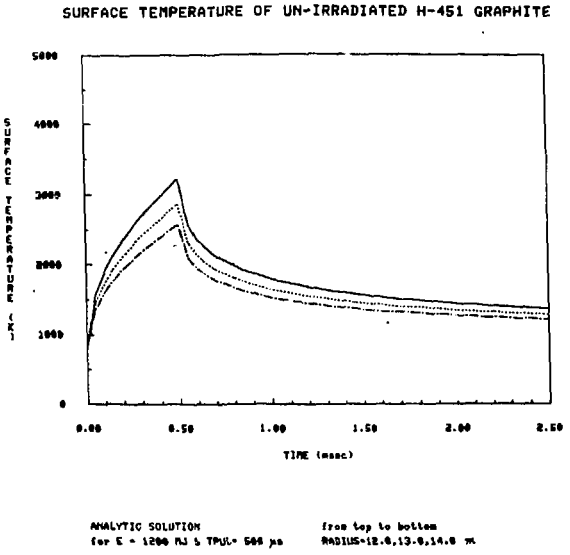


Fig. B.10 The Surface Temperature of Un-irradiated H-451 Graphite With Time for Rectangular Surface Heat Load with No Ablatron, No Radiation Heat Loss and Constant Properties

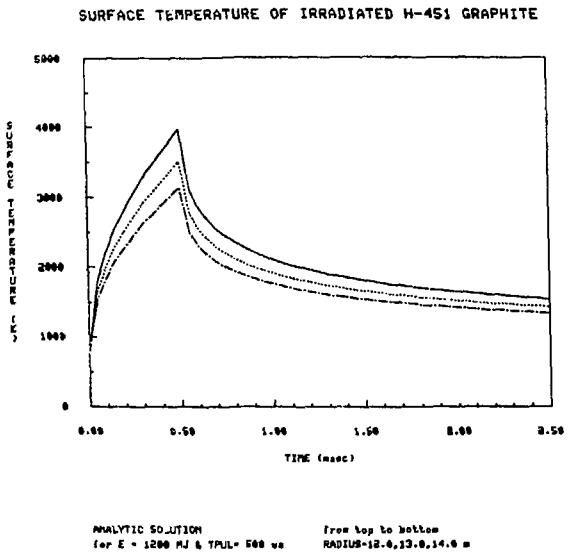


Fig. B.11 The Surface Temperature of Irradiated H-451 Graphite with Time for Rectangular Surface Heat Load with No Ablatron, No Radiation Heat Loss and Constant Properties

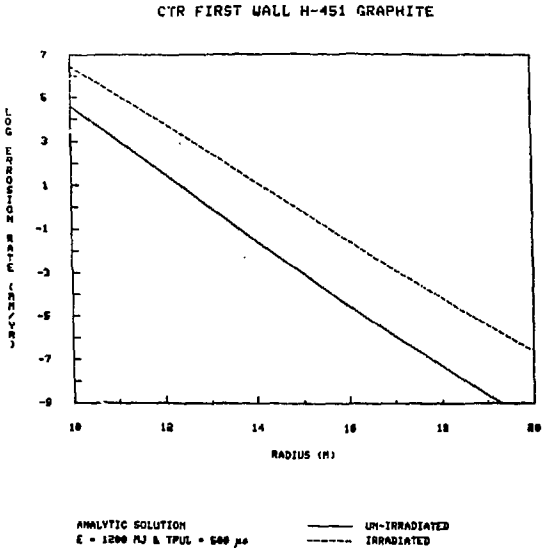


Fig. B.12 The First Wall Erosion Rate with Different Fusion Core Size for Rectangular Heat Load with no Ablatron, No Radiation Heat Loss and Constant Properties

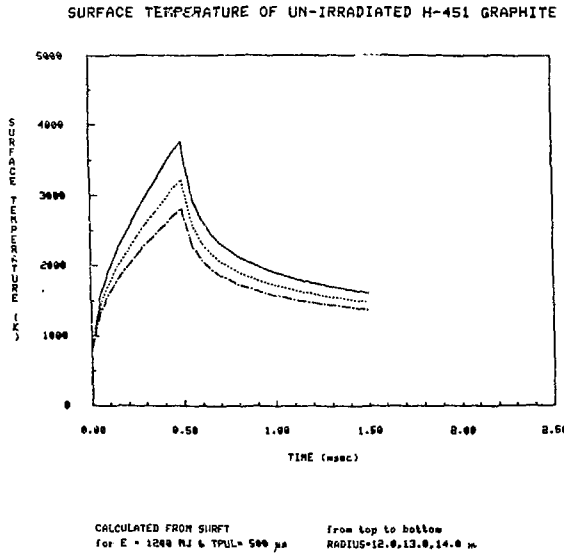


Fig. B.13 The Sur[illegible]emperature of Irradiated H-451 Graphite with Time for [illegible]ngular Surface Heat Load with Ablatron, Radiation Heat Loss and Variable Properties

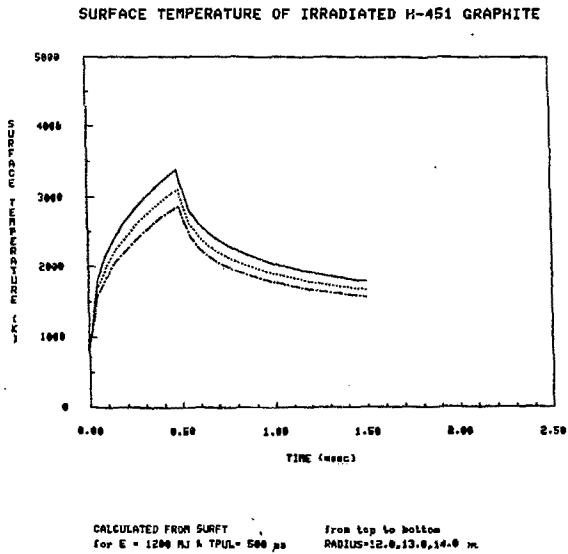


Fig. B.14 The Surface Temperature of Irradiated H-451 Graphite with Time for Rectangular Surface Heat Load with Ablatron, Radiation Heat Loss and Variable Properties

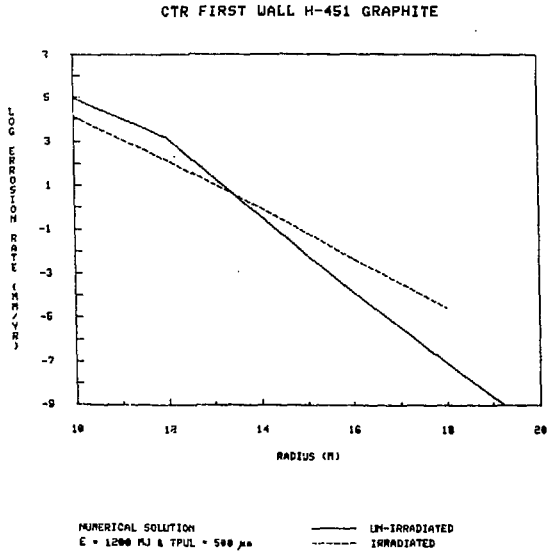


Fig. B.15 The First Wall Erosion Rate with Different Fusion Core Size for Rectangular Heat Load (with Ablatron, Radiation Heat Loss and Variable Properties)

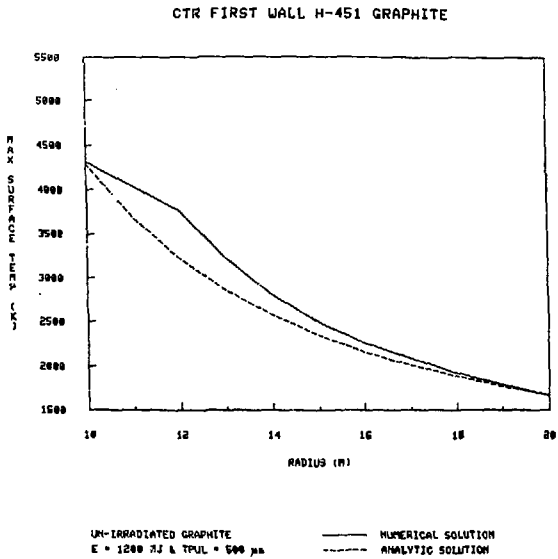


Fig. B.16 Comparison of the Analytic and Numerical Calculation of the Maximum Surface Temperature with Different Fusion Core Size for Un-Irradiated H-451 Graphite

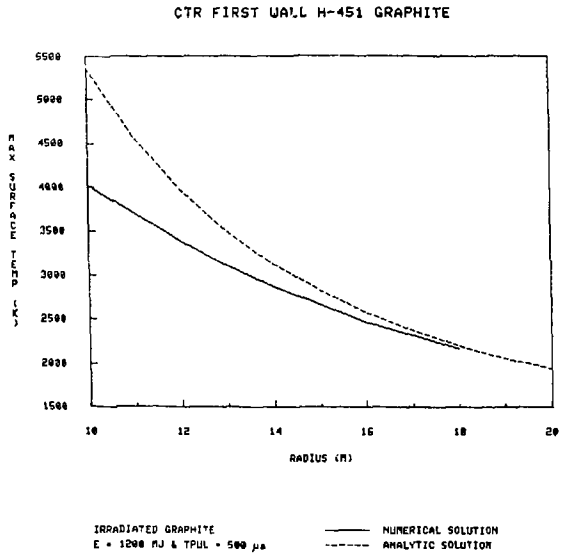


Fig. B.17 The Maximum Surface Temperature with Different Fusion Core Size for Irradiated H-451 Graphite

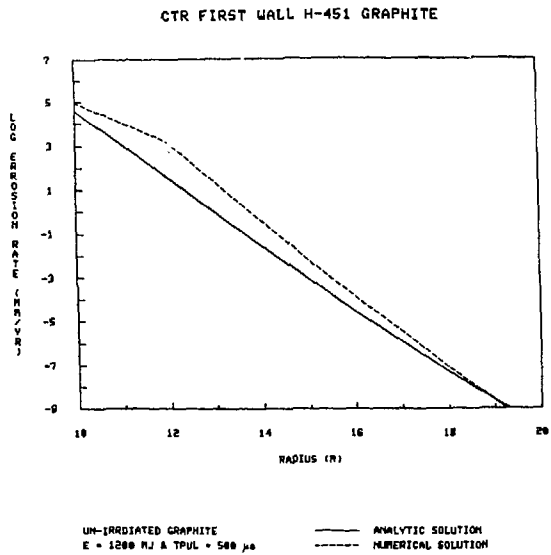


Fig. 3.18 Comparison of the Analytic and Numerical Calculation of the First Wall Erosion Rate with Different Fusion Core Size for Un-Irradiated H-451 Graphite

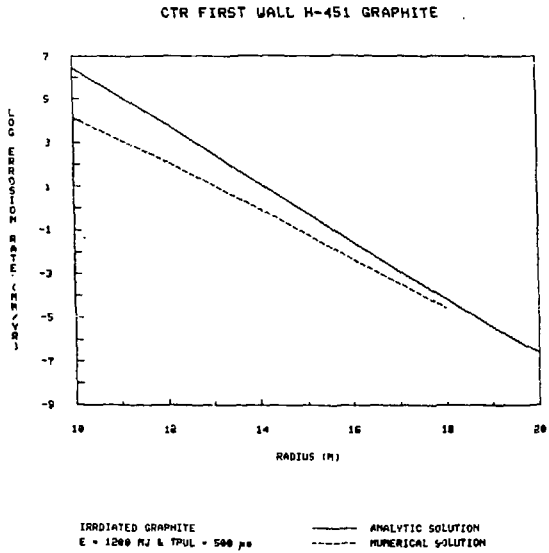


Fig. B.19 Comparison of the Analytic and Numerical Code of the First Wall Erosion Rate with Different Fusion Core Size for Irradiated H-451 Graphite

APPENDIX C: THERMODYNAMIC DATA AND MATERIAL PROPERTIES OF UO_2

C.1 Density

The density of solid UO_2 is given by M. G. Chasanov et al[20] as:

$$\rho = (10.98)/[1+9\times10^{-6}(T-273)+6\times10^{-9}(T-273)^2+3\times10^{-12}(T-273)^3] \quad \text{for } T<3140 \text{ K} \qquad \text{(C-1)}$$

The density of liquid UO_2 is given by L. Leibowitz et al[21] as:

$$\rho = 10.658 - 6.3609\times10^{-4}T \quad \text{for } T>3140 \text{ K} \qquad \text{(C-2)}$$

where ρ is in g/cm^3 and T is in K.

C.2 Specific Heat

The specific heat of solid UO_2 is given by Kerrisk and Clifton[22] as:

$$C_p = \frac{K_1\theta^2 e^{\theta/T}}{T^2(e^{\theta/T}-1)^2} + 2K_2T + \frac{K_3E_D}{RT^2}\, e^{-E_D/RT} \quad \text{for } T<3140 \text{ K} \qquad \text{(C-3)}$$

where θ = 535.285 K

E_D = 1.578×10^5 J/mole

K_1 = 0.2968 J/g-K

K_2 = 1.217×10^{-5} J/g-K^2

K_3 = 8.750×10^{-8} J/g

R = 8.314 J/mole-K

and C_p is in J/g-K and T is in K.

The specific heat of liquid UO_2 is given by Leibowitz[23] as:

$$C_p = 0.503 \text{ J/g-K} \quad \text{for } T>3140 \text{ K} \qquad \text{(C-4)}$$

At the melting point, the enthalpy increment exhibits a discontinuity due to the phase change. This heat of phase transition is

handled by an effective heat capacity term in the vicinity of the melting temperature.[24]

The molar enthalpy of a material at temperature T can be expressed in terms of the heat of fusion and the molar heat capacity at constant pressure C_p as follows:

$$H^o(T) = \begin{cases} H^o_{298} + \int_{298}^{T} C_p dT' & \text{if } T<T_{mp} \\ H^o_{298} + \Delta H_f + \int_{298}^{T} C_p dT' & \text{if } T>T_{mp} \end{cases} \qquad \text{(C-5)}$$

Define an apparent molar heat capacity $\tilde{c}_p$:

$$\tilde{c}_p(T) = C_p(T) + \Delta H_f \delta(T-T_{mp}) \qquad \text{(C-6)}$$

so that

$$H^o(T) = H^o_{298} + \int_{298}^{T} \tilde{c}_p(T')dT' \quad \text{for every T} \qquad \text{(C-7)}$$

Since discontinuities in material properties are not desirable in the numerical solution of the conduction equation, the heat capacity is approximated by a continuous function of temperature; that is, instead of using a delta function to account for the heat of fusion, a Gaussian function of finite width centered at the melting point is used.

Let $C_{pp} \equiv \Delta H_f \delta(T-T_{mp})$ be approximated by a function of Gaussian form:

$$C_{pp} = \frac{\Delta H_f}{\sqrt{\pi}\sigma_1} \exp^{-\left(\frac{T-T_{mp}}{\sigma_1}\right)^2} \qquad \text{(C-8)}$$

where ΔH_f = heat of fusion of UO_2 = 274.4 J/g[23]

T_{mp} = 3140 K[25]

σ_1 = half width of the heat of fusion peak = 50 K

and the apparent specific heat

$$C_p = C_p + C_{pp} \qquad \text{(C-9)}$$

C.3 Thermal Conductivity

The thermal conductivity of solid UO_2 is given by Schmidt[26] as:

$$k = 1/[10.80 + 0.0218(T-273)] \qquad \text{for } T<1473 \text{ K} \qquad \text{(C-10)}$$

$$k = (1+F)/(10.80 + 0.0218\,\theta) \qquad \text{for } 1473 \text{ K}<T<3140 \text{ K} \qquad \text{(C-11)}$$

$$\theta = [T+1777-(T-273)\tanh(T-2323)]/2$$

$$F = [C_p-2\times10^{-5}(T-273)-0.299]/0.299$$

The thermal conductivity of liquid UO_2 is obtained from measured thermal diffusivity of liquid UO_2 (constant value of 2.6×10^{-2} cm^2/sec) by Leibowitz[27]:

$$k = 0.139 - 8.32\times10^{-6}T \qquad \text{for } T>3140 \text{ K} \qquad \text{(C-12)}$$

where k is in W/cm-K, T is in K, and C_p in eq. (C-11) is in J/g-K.

A.4 Vapor Pressure

A computer subroutine following Blackburn's model[28], "ROOT", is used to calculate the partial pressure of each vapor species as a function of both composition and temperature.

A.5 Heat of Vaporization

The heat of vaporization is assumed constant below the melting temperature:

$$\Delta H_{vap} = 2234 \text{ J/g} \qquad \text{for } T<3140 \text{ K} \qquad \text{(C-13)}$$

given by Bogensberger et al[29] in which the averaged vapor pressure curve of Ohse[30] and Tetenbaum and Hunt[31] was fitted to the Claysius-Clapeyton equation.

Given the heat of fusion, 274.4 J/g, from L. Leibowitz, et al.,[32], the heat of vaporization above the melting point is:

$$\Delta H_{vap} = 2234 - 274.4 = 1959.6 \text{ J/g} \qquad \text{for } T>3140 \text{ K} \qquad \text{(C-14)}$$

C.6 Diffusion Coefficient of Oxygen

The diffusion coefficient of oxygen in solid UO_2 is given by J. Belle[33] as:

$$D_o = 1.15 \exp(-28550/T) \qquad \text{for } T<3140 \text{ K} \qquad \text{(C-15)}$$

Since the diffusion coefficient of oxygen in the oxide melt is not available, estimates of upper and lower limits are made in this calculation. An upper estimate can be arrived at with the aid of kinetic gas theory.[34] In a ideal gas the diffusion coefficient is $D = \frac{1}{3} \bar{l} \bar{v}$, where $\bar{l}$ is the mean free path and $\bar{v}$ is the mean velocity of the gas molecules. If the interionic distance in the oxide melt 10 A is introduced for $\bar{l}$, then $D_o = 7.37 \times 10^{-3}$ cm^2/sec at 3140 K and 1.14×10^{-2} cm^2/sec at 7560 K (the estimated critical temperature by Ohse[35]). In this case, D_o is fitted to be approximately

$$(D_o)_{max} = 1.59 \times 10^{-2} \exp(-2516/T) \qquad \text{for } T>3140 \text{ K} \qquad \text{(C-16)}$$

where $(D_o)_{max}$ is in cm^2/sec and T is in K. This sets up an upper limit of diffusion coefficient in the temperature of interest, since the ions in the oxide melt cannot move unimpeded between collisions in the same way as gas molecules. Instead, molecules in

the liquid migrate from one potential ste to an adjacent one, remaining in each site for a certain time. A lower estimate is based upon the fact that the diffusion coefficient in the liquid oxide is higher than that in the solid.

$$(D_o)_{min} = 3.03\times10^{-1} \exp(-24800/T) \quad \text{for } T>3140 \text{ K} \qquad \text{(C-17)}$$

This equation is fitted by taking $D_o = 1.1\times10^{-4}$ cm^2/sec from Belle[33] at 3140 K and $1.14\times10^{-2} cm^2/sec$ at 7560 K.

C.7 Optical Emissivity

The spectral emissivity at λ = 6500 Å is given by Held and Wilder[36] as:

$$\epsilon_{\lambda=6500Å} = 0.83 \qquad \text{(C-18)}$$

which is found insensitive to the temperature over a wide temperature range (450 - 2400 K). It is proposed[21] that $\epsilon_{\lambda=6500Å}$ be constant up to the melting point.

Above the melting point $\epsilon_{\lambda=6500Å}$ is measured by Bober[37] as:

$$\epsilon_{\lambda=6500Å} = 0.81 \text{ to } 0.87 \quad \text{for } 3140 \text{ K}<T<3700 \text{ K} \qquad \text{(C-19)}$$

No data is available for the total hemispherical emissivity of UO_2 with temperature, so an average value of 0.83 is adopted, i.e. $\epsilon_t \approx 0.83$ at all temperatures.

REFERENCES

1. N. Asami, M. Nichikawa and M. Tagnchi, IAEA Symp. on Thermodynamics of Nuclear Materials, Paper IAEA-SM-190/3, Vienna Oct. 1974.

2. R. W. Ohse, P. G. Barrie, H. G. Bogensberger and E. A. Fischer, IAEA Symp. on Thermodynamics of Nuclear Materials, Paper IAEA-SM-190/8, Vienna Oct. 1974.

3. A. Bober, J. U. Karow and K. Schretzmann, IAEA Symp. on Thermodynamics of Nuclear Materials, Paper IAEA-SM-190/34, Vienna Oct. 1974.

4. C. H. Tsai, A. Covington and D. R. Olander, Materials and Molecular Research Division, Lawrence Berkeley Laboratory, LBL-6016, UC-13, TID-4500-R65 (1976) p.188.

5. R. W. Ohse, P. G. Berrie, H. G. Bogensberger and E. A. Fischer, J. Nucl. Mater., 59 (1976) 112-124.

6. F. W. Dabby and U-C Paek, IEEE J. Quan. Elec. Vol. QE-8, No.2, Feb. 1972.

7. H. U. Karow and M. Bober, IAEA Symp. on Thermodynamics of Nuclear Materials, Paper IAEA-SM-236/22, Vienna, 1979.

8. W. Breitung, KFK-2240, Jun. 1976.

9. R. A. Olstad and D. R. Olander, J. Appl. Phys. Vol. 46, No. 4, Apr. 1975, pp. 1499-1518.

10. R. A. Olstad and D. R. Olander, J. Appl. Phys. Vol. 46, No. 4, Apr. 1975, pp. 1509-1518.

11. S. I. Anisimov, Soviet Physics JETP, Vol. 27, No. 1, Jul. 1968, pp. 182-183.

12. Masahide Murakami and Koichi Oshima, 9th Int. Symp. on Rarefied Gas Dynamics, 1974, Vol. II, pp. F.6.1-F.6.9.

13. Tor Ytrehus, 10th Int. Symp. on Rarefied Gas Dynamics, Paper 28, July 1976.

14. H. S. Carslaw and J. C. Jaeger, "Conduction of Heat in Solids" Oxford University Press, London and N.Y. 1959.

15. R. D. Richtmyer and K. W. Morton, "Difference Methods for Initial Value Problems" 2nd edit., Interscience, N.Y. 1967.

16. G. E. Forsythe and W. R. Wason, "Finite Difference Methods for Partial Differential Equations" John Wiley, N.Y. 1960.

17. J. Douglas, Jr. "Survey of Numerical Methods for Parabolic Differential Equations" in Advances in Computers, Vol. 2 Academic Press, N.Y. 1961, pp. 1-54.

18. W. F. Ames, "Nonlinear Partial Differential Equations in Engineering", Academic Press, N.Y. 1965.

19. Germund Dahlquist and Ake Bjerck, "Numerical Methods", Series in Automatic Computation, Prentice-Hall, Inc., N.Y. 1974.

20. M. G. Chasanov, L. Leibowitz and S. D. Gabelnick, J. Nucl. Mater. 49 (1973/74) 129-135.

21. L. Leibowitz et al, Report ANL-CEN-RSD-76-1, Mar. 1976.

22. J. F. Kerrisk and D. G. Clifton, Nucl. Tech. 16 (1972) 531.

23. L. Leibowitz, J. Nucl. Mater., 39 (1971) 115.

24. R. A. Olstad, Ph. D. Thesis, University of California - Berkeley, Dec. 1972.

25. R. E. Latta and R. E. Fryxell, J. Nucl. Mater. 35 (1970) 195-210.

26. H. E. Schmidt, J. Nucl. Mater., 39 (1971) 234.

27. C. Kim, R. A. Blomquist and L. Leibowitz, Report ANL-78-15, 1977, pp. 41-45.

28. P. E. Blackburn, J. Nucl. Mater. 46 (1973) 244-252.

29. H. G. Bogensberger, et al., GfK Report KFK-2272, May 1976.

30. R. W. Ohse, J. Chem. Phys. 44 (1966) 1375.

31. M. Tetenbaum and P. D. Hunt, J. Nucl. Mater., 34 (1970) 86-91.

32. L. Leibowitz, L. W. Mishler and M. G. Chasanov, J. Nucl. Mater., 29 (1969) 356.

33. J. Belle, "Properties of UO_2", 1962, p. 7.

34. W. Breitung, KFK-2091, Jun. 1975.

35. E. A. Fischer, P. R. Kinsman and R. W. Ohse, J. Nucl. Mater., 59 (1976) 125-136.

36. P. C. Held and D. R. Wilder, J. Am. Cer. Soc., 52 (1969) 182-186.

37. M. Bober and H. U. Karow, 7th Symp. on Thermodynamical Properties, NBS Washington, D.C., 1977.

38. G. R. Hopkins, Trans. ANS 22, 163 (1975).

39. C. R. Hopkins, R. J. Price, R. E. Bullock and J. A. Dalessandro, Proc. third topical meeting on the technology of the controlled Nuclear Fusion, Vol 1, CONF-780508 (1978) pp. 324-333.

40. R. W. Conn et al., UWFDM-229, U. of Wisc. 1977.

41. G. R. Hopkins et al., GA-A14894, Mar. 1978.