2-7-75

01-1158

LA-5832-MS
Informal Report

UC-20
Reporting Date: December 1974
Issued: January 1975

Heat Transfer Model for Composite First Wall Materials in a Pulsed High-Beta Controlled Thermonuclear Reactor (CTR)

by

Jefferson W. Tester
C. C. Herrick

los alamos
scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87544

An Affirmative Action/Equal Opportunity Employer

UNITED STATES
ATOMIC ENERGY COMMISSION
CONTRACT W-7405-ENG. 36

In the interest of prompt distribution, this LAMS report was not edited by the Technical Information staff.

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22151
Price: Printed Copy $4.00 Microfiche $2.25

HEAT TRANSFER MODEL FOR COMPOSITE FIRST WALL MATERIALS IN A PULSED HIGH-BETA CONTROLLED THERMONUCLEAR REACTOR (CTR)

by

Jefferson W. Tester and C. C. Herrick

ABSTRACT

A computer model has been constructed to predict temperature and time excursions for radial composite walls currently under consideration for pulsed high-beta Z-pinch machines. The effects of incident flux, internal heat distribution functions, thermal properties, and material dimensions have been examined for a Nb/Al_2O_3 composite to establish the feasibility of the model.

I. INTRODUCTION AND SCOPE

In a previous report,[1] a preliminary treatment of first wall heat transfer and chemical stability effects was presented. For homogeneous materials such as Nb, Al_2O_3, BeO, or BN temperature excursions and/or chemical reactivity with molecular or atomic hydrogen became prohibitive, indicating that a composite first wall might present a feasible alternative. Prediction of thermodynamic equilibrium, kinetic, thermal stressing, and radiation damage effects require first-hand knowledge of anticipated temperature-time profiles for composite wall materials intended for use in pulsed, high-beta, controlled thermonuclear reactors (CTR's) where heat fluxes on the order of 1 kW/cm^2 or more are possible. Furthermore, estimates of maximum operating temperatures for the molten lithium blanket are useful in establishing the effectiveness of proposed CTR's in producing high temperature heat sources for direct or indirect energy production.

II. DESCRIPTION OF THE MODEL

A. Basic Geometry

Due to the large radius of curvature (30 m) and torus diameter (~ 1 m) a rectangular coordinate system was used for the model. Figure 1 illustrates schematically how a Z-pinch prototype might be

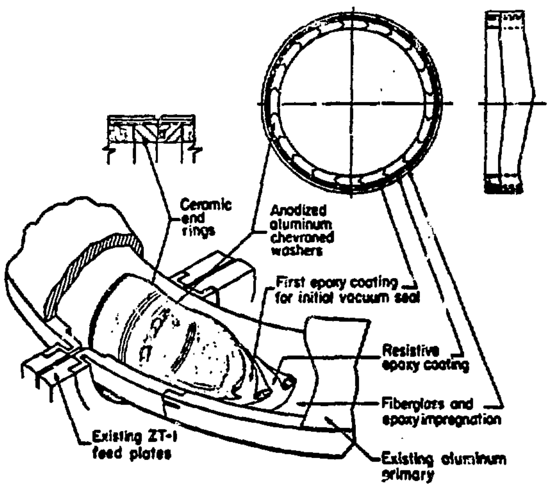

Fig. 1. Schematic of prototype Z-pinch design.[2]

designed.[2] The major feature of interest is the radial arrangement of the composite first wall. In the prototype design the conductor (material 1) is an aluminum washer separated by thin layers of anodized aluminum which can be conceptually thought of as the insulator (material 2). Figures 2A and 2B schematically represent the geometry used in the model. The grid has I2 points in the x-direction and J

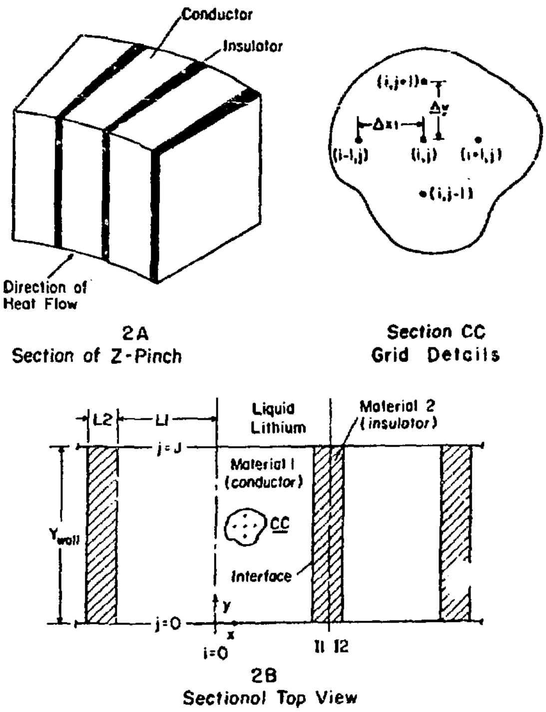

Fig. 2. Geometry employed for finite difference grid. I2 x J points having Δy spacing in the y-direction and $\Delta x1(\Delta x2)$spacing in the x-direction for materials 1(2).

points in the y-direction with the point at I1 on the interface between materials 1 and 2.

A time-dependent heat flux impinges on the inner surface of the composite [i=0, ..., I1, ... I2; j=0], and a liquid metal (lithium)/metal conduction temperature dependent heat transfer resistance exists on the outer surface [i=0, ..., I1, ..., I2; j=J]. The two center lines (-·-) define mirror symmetry planes in each material and can be represented by a zero flux $[-k\frac{\partial T}{\partial x} = 0]$ condition.

B. Design Criteria

Heat enters the first wall via several sources, including:

1. Bremsstrahlung radiation,
2. n-γ reactions within the wall, and
3. direct neutron deposition energy.

In a preliminary report, Burnett, Ellis, Oliphant, and Ribe[3] demonstrated that most of the energy deposited (> 85%) was Bremsstrahlung energy. In our model, the total heat absorbed is divided

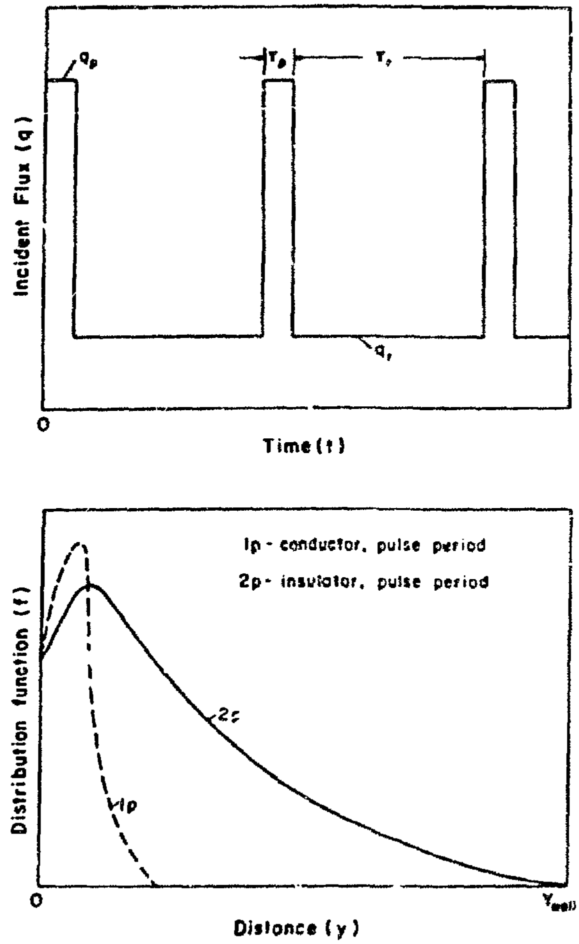

Fig. 3. Incident heat flux q and heat distribution functions $f = H(y)/q_p \Delta y$ expressed as a fraction of the pulse heat flux q_p (arbitrary scales).

into two quantities:

1. An incident flux which is deposited at the surface y = 0.
2. A distributed heat source function $H = f(y)$ representing the energy absorbed as a function of distance into the wall from the point y = 0 to the extent of the wall y = Ywall.

Consequently, for a two-component composite, there would be four H functions corresponding to each material in the pulse and rest mode. In Fig. 3, we present idealizations of these heat distribution and incident flux functions used in the current approach.

Only distribution ($H(y)$) curves for the pulse period are shown in Fig. 3, since negligible values for the rest period are anticipated when heat transfer to the wall will be primarily by radiation and convection from the expanding plasma. As a first approximation, one might assume that $H(y)/q = 0$ during the rest period for both materials, indicating that all of the heat is deposited on the inside surface of the wall. Nevertheless, in implementing the model, the user is free to select any heat distribution function that is appropriate. For example, for our Nb/Al_2O_3 composite both rest and pulse H functions are set to zero for Nb, and a finite H used only for the pulse mode in Al_2O_3 (see Ref. 3). In general, the insulator (ceramic) would be expected to have a much wider distribution function than the conductor (metal) as is illustrated in Fig. 3.

The square wave function idealization for q is somewhat of an over-simplification of the actual case which might show an exponential increase and decrease of heat flux during the cycle.[4] However, at this stage, a square wave functionality should be adequate. Actual values for the incident heat flux q may be determined by design limitations of the materials used in the first wall. For example, the magnitude of q can be partially controlled by changing the amount of first wall surface area for a given amount of heat produced during the cycle.

C. Governing Equations and Boundary Conditions

The following partial differential equation (PDE) applicable to unsteady state, two-dimensional heat conduction was used for both materials.

$$\alpha_i \left[\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right] + \frac{H_i\ (y)}{\rho_i\ C_{p_i}} = \frac{\partial T}{\partial t} \quad . \tag{1}$$

$i = 1,2$ (for both materials) .

An ambient temperature (T_B) equal to the bulk lithium temperature is assumed for the initial condition at $t = 0$. Four boundary conditions are applied to positions specified on Fig. 2b:

1. Incident heat flux at the inside surface (see Fig. 3)

at $y = 0$ ($j = 0$), all x

$$-ki\left(\frac{\partial T}{\partial y}\right) = q_i(t)^{\dagger} \quad . \tag{2}$$

2. Temperature dependent flux with contact resistance at the outside surface

at $y = 0$ ($j = 0$), all x

$$-ki\left(\frac{\partial T}{\partial y}\right) = h\ (T-T_B)^{\dagger} \tag{3}$$

where h is an effective heat transfer coefficient applying to the molten lithium blanket and any solid liners that might be used.

3. Continuous flux and temperature at the interface

at $x = L1/2$ ($i = I1$), all y

$$k1\left(\frac{\partial T}{\partial x}\right) = k2\left(\frac{\partial T}{\partial x}\right) \quad . \tag{4}$$

4. Zero flux condition at centerlines of materials 1 and 2 via symmetry

at $x = 0$: ($i = 0$), $\left(\frac{\partial T}{\partial x}\right) = 0$ (5)

at $x = \frac{(L1 + L2)}{2}$ ($i = I2$) $\left(\frac{\partial T}{\partial x}\right) = 0$ (6)

In solving Eq. (1) to generate temperature profiles as functions of time, a dimensionless temperature u was defined as

$$u = \frac{T-T_B}{T_B} \quad , \tag{7}$$

and finite difference equations were written to approximate the PDE. Appendix A contains a tabular presentation of these equations. A detailed description of the finite difference formulation of the boundary conditions is presented in Appendix B. An Alternating Direction Implicit (ADI) scheme was used to solve the system of equations (see Appendix C). The advantages of an implicit rather than explicit scheme should be useful in conserving machine time and in adding to the flexibility of the code.

---

†In the expression ki or q_i the i = 1 or 2 depending on what material it is.

The tridiagonal algorithm and a complete listing of the Madcap V code are presented in Appendixes D and E.

III. LIMITATIONS AND APPLICATIONS OF THE MODEL

Several features of the model have been kept general; for example, various wall sizes can be used with any two materials. If the repeating thicknesses in the x-direction, L1 and L2, become much smaller than the thicknees of the wall in the y-direction Y_w, the code reverts to a unidirectional (y only) calculation of temperature profiles with area average physical properties used. Any combination of incident heat flux and internal heat generation terms can be used. The outside boundary condition (all x, y= Ywall at j=J) is temperature dependent in order that an effective heat transfer coefficient can be used which combines the resistances of a liquid lithium boundary layer and any metallic and/or ceramic backing material that might be present.

The interface condition (at i=I1) can be specified by either of two procedures (see Appendix B):

1. Criteria of continuous flux at the boundary

$$-k1 \left(\frac{\partial T}{\partial x}\right) = -k2 \left(\frac{\partial T}{\partial x}\right) . \quad (4)$$

2. Criteria of continuous flux and an operable PDE at the boundary.

In using the code, large time steps should be avoided since they can cause inaccuracies as well as instabilities because of the pulsed boundary condition and the interface between materials 1 and 2. At least 10 time steps for each pulse comprise the upper limit, i.e., for a 10 ms (10^{-3} s) pulse Δt would be 1 ms. Since the rest period is usually much longer than the pulse period, e.g., 90 ms compared to 10 ms, a larger Δt could be used during this period if conserving computation time became important.

IV. PRELIMINARY RESULTS AND DISCUSSION

The main purpose of this section is to discuss preliminary results which demonstrate the feasibility of applying our heat transfer model to CTR applications.

A. Choice of a test system

A niobium (Nb) - alumina (Al_2O_3) radial composite was selected since it is currently under consideration as a first wall composite material,[3] and because its thermal properties are representative of typical metallic conductors and ceramic insulators that might be considered at a later time. Present Z-pinch design estimates will require an insulating capacity between 1 to 3 kV/cm which will control the relative dimensions of insulator (2) to conductor (1).[2] Although actual sizes have not been specified for a real operating system, a prototype experimental design utilizing anodized aluminium washers (0.0254 cm thick Al with approximately 0.0005 cm of anodized coating) is currently under construction by Phillips and associates.[2] A large scale-up from these dimensions is anticipated for future experiments and consequently a test geometry with about 1 cm width of conductor to 0.1 cm of insulator with an overall wall thickness of 1 cm was selected. Total heat flux loads on the first wall during the pulse period are expected to be the range of 0.1 to 10 kW/cm^2 consisting mainly of Bremsstrahlung and n-γ energy. Niobium, due to its high mass number, will absorb most of the plasma energy within a very thin layer (~0.01mm).[3] Alumina, on the other hand, will absorb the energy continuously with a distribution function given in Fig. 4. As suggested by Burnett et al.[3] an average electron temperature of 25 keV was selected to define the heat generation function. During the rest period, approximately 10% of the instantaneous pulse heat flux will impinge on the inside surface of the wall with no distribution within the wall (H(y) = 0). As a first approximation a constant value was used during the entire rest period (see Fig. 3). In order to meet the Lawson criterion a 10% duty cycle corresponding to a 0.01 s pulse and a 0.09 s rest period has been employed for the test case. A range of outside surface (y = Ywall, Fig. 2) heat transfer coefficients from h = 0.14 to 14 cal/cm^2 s K were utilized to approximate the thermal resistance anticipated from a niobium (Nb)/ boron nitride (BN) protective liner and a molten lithium boundary layer. Average values for material properties were selected at approximately 800°C, and these are tabulated in Table I for several first wall material possibilities.

A summary of the system parameters investigated is presented in Table II. Again, we would like to emphasize that our purpose at this stage was to

TABLE I

MATERIAL PROPERTIES(*)

Conductors (1)	k cal/(cm^2 s K/cm)	ρ g/cm^3	C_p cal/gK	$\alpha = k/\rho C_p$ cm^2/s
Niobium, Nb	0.158	8.57	0.0736	0.250
Molybdenum, Mo	0.350	10.20	0.0630	0.545
Insulators (2)				
Alumina, α-Al_2O_3	0.034	3.96	0.198	0.0434
Beryllia, BeO	0.835	3.00	0.50	0.0557

k-thermal conductivity ρ-density C_p-heat capacity α-thermal diffusivity

(*) Data based on information taken at ~800°C from

1. "Perry's Handbook for Chemical Engineers," 4th Ed., McGraw-Hill N.Y., (1965).
2. "Handbook of Chemistry and Physics," Chemical Rubber Publ., N.Y., 41st Ed. (1962).
3. "Thermal Physical Properties of Matter," Vols. 1-2 Eds. Touloukian, Powell, Ho, and Klemens, Plenum Publ. Corp., N.Y. (1970).

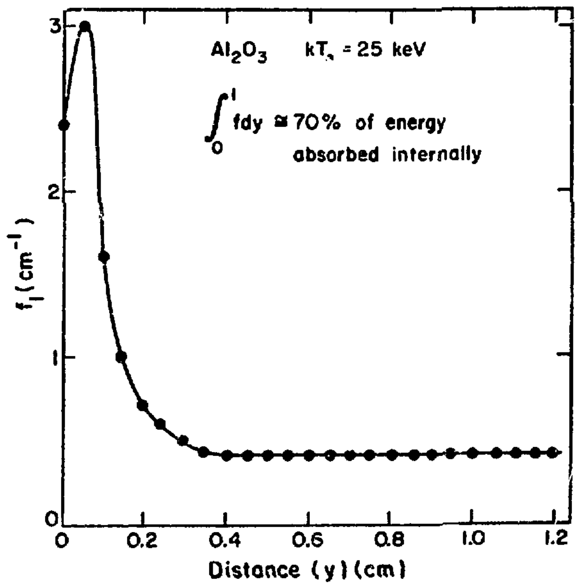

Fig. 4. Heat distribution function for Al_2O_3 for pulse period (original data Ref 3 kT_e-electron temperature of the plasma).

demonstrate calculational feasibility rather than propose a definitive design.

B. Temperature-Time Excursions for a Nb/Al_2O_3 Composite

Table III (A and B) provides a complete summary of the test runs made. The effects of heat flux, heat transfer coefficient, time step, and grid size parameters were all examined.

A typical temperature-time excursion for seven consecutive pulses (for complete parameter specification see Table III, Run 1) is presented in Fig. 5. Several features of the graph are apparent.

1. There are no inherent instabilities in the ADI solution.
2. The outside surface temperatures, $\Delta T(0,J)$, $\Delta T(I1,J)$, and $\Delta T(I2,J)$, do not increase due to the large value of h = 14 cal/cm^2 s K used.
3. The interface $\Delta T(I1,0)$ is between the maximum excursion in the Al_2O_3 layer ($\Delta T(0,0)$) and the minimum in Nb layer ($\Delta T(0,0)$).
4. The inside surface temperature for either material Nb or Al_2O_3 does not relax to what its initial level was before the pulse, hence there is a continuous increase in ΔT which should approach steady-state conditions after a temperature profile of sufficient magnitude has been established

TABLE II

SYSTEM PARAMETERS INVESTIGATED

1. Duty cycle $\tau_p = .01$ s $\tau_r = .09$ s

2. Incident heat flux

   q_i (pulse period) 0.1-1.0 kw/cm^2 (~23.82 - 238.2 cal/cm^2 s)

   q_i (rest period) .01-.1 kw/cm^2 (~2.382 - 23.82 cal/cm^2 s)

3. Heat distribution/generation function H(y)

   separate functions for insulator (2) and conductor (1) during pulse and rest mode utilized

4. Heat transfer coefficient h = .14-14 cal/cm^2 s K

   outside surface-combined resistance of backing material and liquid lithium

5. Bulk temperature T_B = 600°C[a]

6. geometrical parameters

   wall thickness Ywall = 1 cm

   Composite: conductor thickness L1 = .01-1 cm

   Composite: insulator thickness L2 = .0005 - .1 cm

7. Equation solution parameters

grid sizes	$\Delta x1$ = .0005 - .05 cm
	$\Delta x2$ = .0005 - .005 cm
	Δy = .01 - .02 cm
time steps	Δt = 10 - 2000 μs (10^{-6} s)

[a]Really arbitrary, material limitations will set the upper bound.

to conduct away the total energy deposited during the pulse and rest periods.

A series of temperature profiles are presented in Fig. 6 for the conditions of Run 5 (Table III). In this case, heat was deposited on the inside surface of the Nb layer during both pulse and rest periods and on the inside surface of the Al_2O_3 layer during the rest period. The heat distribution function given in Fig. 4 was used for Al_2O_3 during the pulse period. One can see a marked reduction in the temperature excursion of the Al_2O_3 layer caused by distributing the heat. All three profiles, at the center lines of materials 1 and 2 and the interface, are uniform in shape and magnitude for the three times given. This effect is also illustrated by comparing Fig. 7b with Fig. 8 which have identical conditions, except in Fig. 8 no heat distribution was used (H(y)'s = 0).

The magnitude of the outside surface effective heat transfer coefficient has a significant effect on predicted temperature-time excursions (see Figs. 7a and 7b). With h = 0.14 cal/cm^2 s K to approximate anticipated thermal resistances, the outside wall temperature has increased by > 60K over the bulk lithium value in 30 pulses. This ΔT will, of course, continue to increase until steady-state conditions are reached.

TABLE III

TABLE III (SECTION A)

SUMMARY OF RESULTS FOR COMPOSITE/PULSED CASE[a]

Run	Conductor (1)	Insulator (2)	Geometry: L1 cm	Geometry: L2 cm	Geometry: Ywall cm	Grid Size: Δx1 cm	Grid Size: Δx2 cm	Grid Size: Δy cm	Time Step Δt μs	Heat Transfer Coeff. Outside Surface h cal/cm^2 s K	Total Incident Flux: q_i Rest Period kW/cm^2	Total Incident Flux: q_i Pulse Period kW/cm^2
	Niobium	Alumina										
1	Nb	Al_2O_3	1.0	0.1	1.0	0.05	0.005	0.02	1000	14	0.01	1.0
2	Nb	Al_2O_3	0.01	0.0005	1.0	0.0005	0.00005	0.02	1000	14	0.01	1.0
3	Nb	Al_2O_3	1.0	0.1	1.0	0.05	0.005	0.02	1000	14	0.01	1.0
4	Nb	Al_2O_3	1.0	0.1	1.0	0.05	0.005	0.02	1000	14	0.1	1.0
5+9	Nb	Al_2O_3	1.0	0.1	1.0	0.05	0.005	0.02	1000	0.14	0.1	1.0
6	Nb	Al_2O_3	1.0	0.1	1.0	0.05	0.005	0.02	100	0.14	0.1	1.0
7	Nb	Al_2O_3	1.0	0.1	1.0	0.025	0.0025	0.01	200	0.14	0.1	1.0
8+10	Nb	Al_2O_3	1.0	0.1	1.0	0.05	0.005	0.02	1000	0.14	0.1	1.0
	Molybdenum	Beryllia										
	Mo	BeO	1.0	0.1	1.0	0.05	0.005	0.02	1000	0.14	0.1	1.0

[a] Conditions fixed for all runs: $\tau_p = 0.01$ s $\tau_r = 0.09$ s.

TABLE III (SECTION B)

SUMMARY OF RESULTS FOR COMPOSITE/PULSED CASE[a]

Run	Heat Distribution Functions Utilized[c]: Conductor (1) Pulse Period Hp1(y)	Conductor (1) Rest Period Hr1(y)	Insulator (2) Pulse Period Hp2(y)	Insulator (2) Rest Period Hp2(y)	Steady State Temperature Excursions $\Delta T(x,y,t=\infty)$[b]: Inside Surface (Plasma Side) Conductor $\Delta T(x=0, y=0, t=\infty)$ K	Inside Surface (Plasma Side) Interface $\Delta T(x=l1, y=0, t=\infty)$ K	Inside Surface (Plasma Side) Insulator $\Delta T(x=l2, y=0, t=\infty)$ K	Outside Surface Average $\Delta T(<x>, y=Ywall, t=\infty)$ K	Comments
1	0	0	0	0	370	460	490	~ 0	
2	0	0	0	0	260	260	260	~ 0	unidirectional (y only)
3	0	0	0	0	250	320	380	~ 0	
4	0	0	Hp2(y)	0	360	351	348	~ 0	
5+9	0	0	Hp2(y)	0	650	640	640	300	
6	0	0	Hp2(y)	0	650	640	640	300[d]	
7	0	0	Hp2(y)	0	650	640	640	300[d]	
8+10	0	0	0	0	600	695	810	300	
	0	0	Hp2(y)	0					

[a] Refer to nomenclature section (Appendix F) and Figs. 1-2.

[b] Extrapolated to ∞ time.

[c] Refer to section IIC and Figs. 3-4.

[d] Equivalent to run 5.

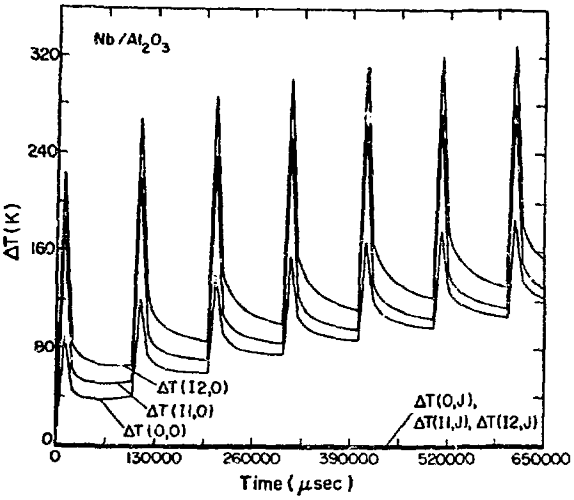

Fig. 5. Temperature-time excursions for a Nb (1cm)/Al_2O_3 (0.1 cm) composite at six locations. For parameter specifications see Table III, Run 1, and see Fig. 2 for geometrical grid locations.

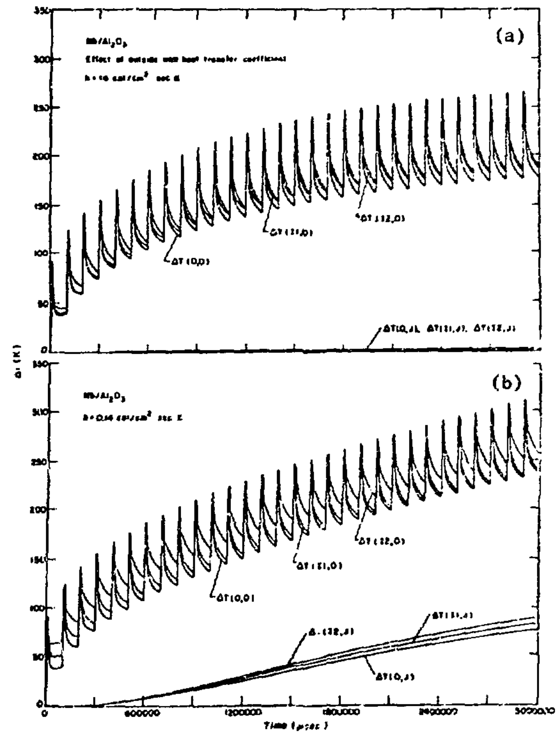

Fig. 7. Effect of outside wall heat transfer coefficient h on temperature-time excursion for a Nb/Al_2O_3 composite. For parameter specifications see Table III, Runs 4(7a), 5(7b).

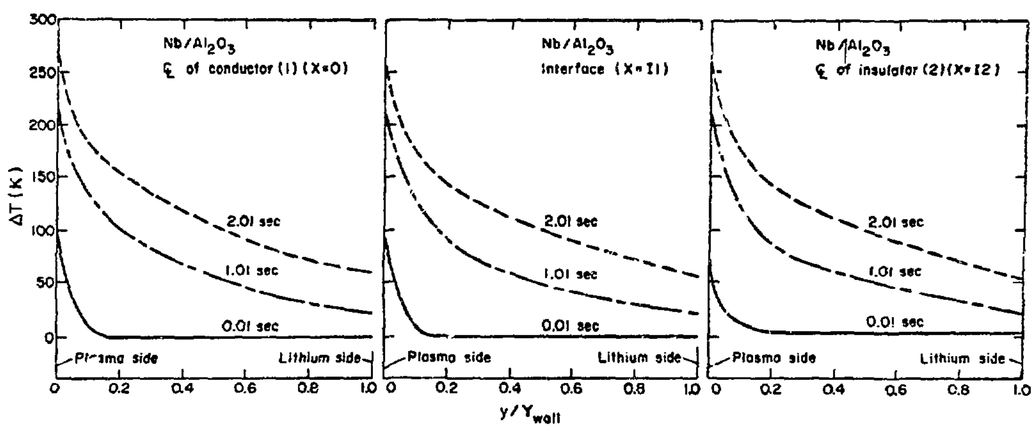

Fig. 6. Approximate temperature profiles T = f(y) at various times (2.01 s - 21 pulses, 1.01 s - 11 pulses, 0.01 s - 1 pulse). For parameter specifications see Table III, Run 5 and see Fig. 7b for temperature-time excursion.

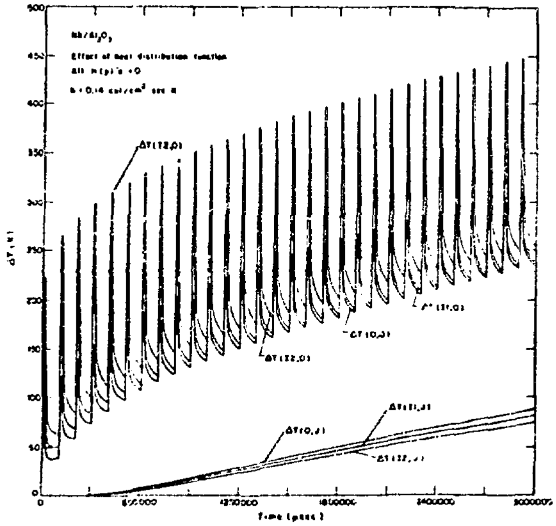

Fig. 8 Effect of heat distribution function on the temperature-time excursion of an Nb/Al_2O_3 composite. For parameter specifications see Table III, Run 8.

C. Approach to Steady State

As steady state is reached, the temperature profile at any position along the composite will stabilize except in the vicinity of the inside surface where it is continuously pulsed. This behavior was observed in a preliminary study of heat transfer effects.[1] Because the thermal time constant $\tau_w = Y_w^2/\alpha$ is large compared to a cycle time of 0.1 s, e.g., for a 1-cm wall $\tau_w(Al_2O_3) \cong 23$ s and $\tau_w(Nb) \cong 6$ s and because an additional thermal resistance is imposed by the low h = .14 cal/cm² s K on the outside surface, successive pulsing will cause ΔT to increase at any point in the wall. A crude estimate of the maximum ΔT anticipated is given by superimposing both the ΔT_a equivalent to steady-state heat transfer through the wall and the ΔT_h caused by thermal contact resistance at the outside surface onto the ΔT_p caused by the pulse itself. For instance, at the center line of the conductor (0,0), an estimate of $\Delta T^{\infty}_{0,0}$ at steady state is given by,

$$\Delta T^{\infty}_{0,0} \cong \Delta T_a + \Delta T_h + \Delta T_p \quad ,$$

where
$$\Delta T_a = \frac{\text{(net heat transferred/time)}}{kl/Y_w}$$

$$= \frac{(q_p \tau_p + q_r \tau_r)\, Y_w}{(\tau_p + \tau_r)\, kl}$$

ΔT_r = temperature rise after the 1st pulse at (0.0)

$$\Delta T_h = \frac{\text{(net heat transferred/time)}}{h}$$

$$= \frac{(q_p \tau_p + q_r \tau_r)}{(\tau_p + \tau_r) h} \quad .$$

For the case of a 1 kW/cm² (238.2 cal/s cm²) pulse and a .1 kW/cm² (23.82 cal/s cm²) heat dump,

$$\Delta T_a = 287 \text{ K}$$
$$\Delta T_p \cong 90 \text{ K}$$
$$\Delta T_h = 333 \text{ K} \quad .$$

Therefore,

$$\Delta T^{\infty}_{0,0} \cong 710 \text{ K} \quad .$$

From Table III, one can see that excursions of 650 K are typical for these conditions (Runs 5,6,and 7).

D. Prototype Geometry - Effective Undirectional Heat Transport

Run 2 attempted to simulate conditions similar to those expected in the prototype Z-pinch reactor (Fig. 1). The widths of Nb and Al_2O_3 in the x-direction, .01 cm for Nb and .0005 cm for Al_2O_3, are very small compared to the thickness of the wall in the y-direction, 1 cm. Consequently, conduction in the x-direction is fast and can be neglected relative to that in the y-direction and the code performs a unidirectional ADI solution to the PDE using area average properties. In Fig. 9, temperature-time excursions are presented for the case with h = .14 cal/cm² s K.

E. Convergence and Stability of the Method - Effect of Grid Size and Time Step

Convergence of the ADI technique was checked with Runs 6 and 7 by reducing the grid sizes, $\Delta x1$ from .05 to .025 cm and $\Delta x2$ from .005 to .0025 cm and Δy from .02 to .01 cm, and time step Δt from

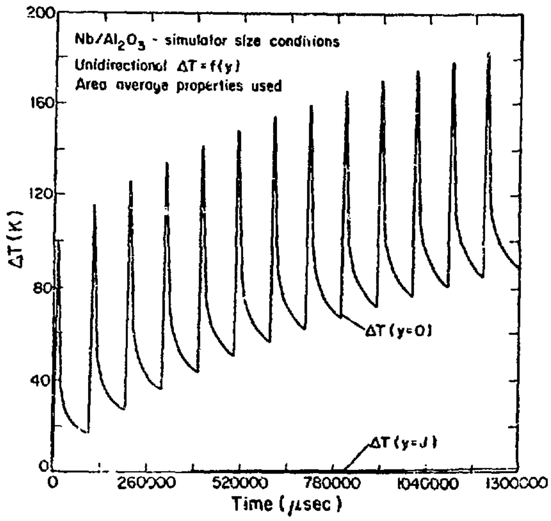

Fig. 9 Temperature-time excursion for a Nb/Al_2O_3 composite having similar dimensions to the prototype Z-pinch (Fig. 1). For parameter specifications see Table III, Run 2.

1000 to 200 μ s. Temperature profiles varied by no more than 5% at equivalent grid locations. Furthermore, when the composite was reduced to a single component, e.g., Nb, and a two-dimensional ADI solution was run, x-direction variation of ΔT was less than 0.1% and the temperature-time excursions were consistent with previous data accumulated for unidirectional heat flow using an explicit method.[1]

Although the ADI technique, as applied to rectangular two-dimensional problems, should be unconditionally stable regardless of the choices of Δt, Δx, and Δy,[9] our specific application of the ADI technique did result in instabilities as mentioned in Sec. III. The pulsed heat flux and interface condition were probably responsible for this since when they were removed from the problem by using a single component and continuous flux boundary, Δt could be selected independently of Δx and Δy. Certain improvements to the stability of the ADI procedure are obtained if the grid system is converted to a half-interval system with the interface containing $\Delta x1/2$ and $\Delta x2/2$ parts of materials 1 and 2.

F. Concluding Remarks

The computer model for heat flow in radial composite CTR first wall materials should provide a useful tool for establishing temperature excursions and profiles which are necessary in evaluating the mechanical and chemical behavior of any proposed materials.

V. RECOMMENDATIONS

1. Additional materials should be examined, including, ZrO_2, BeO, and other insulating oxides as well as Ta, Zr, Mo, and other conducting metals.
2. Having established anticipated temperature-time excursions, other properties such as chemical stability, radiation damage including void and helium bubble growth, thermal stressing, and other aspects of materials compatability should be considered.[1,5,6]
3. By selecting a range of thermal properties, dimensions, incident fluxes, and heat distribution functions, generalized thermal history charts applicable to pulsed-high-beta machines could easily be generated for use in preliminary design work.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the assistance given by J. A. Phillips of CTR-3; R. L. Bivins, M. Stein, and R. D. Richtymer of LASL, and D. G. Wilson of ORNL.

APPENDIX A

FINITE DIFFERENCE EQUATION FORMALISM

Tables A-1 and A-2 list the difference equations utilized by the code. Both sequences of sweeping x first and then y, and vice versa, are presented. In addition, two different equations applying at the interface between materials 1 and 2 are included. A complete description of the nomenclature employed is given in Appendix F and a partial one below for Tables A-1 and A-2. Tridiagonal matrix coefficients are easily determined by recalling that a would be the coefficient of the i-1 term, b the i term, and c the i + 1 term and d the remaining terms. (See Appendix D.)

Nomenclature for Tables A-1 and A-2

$A1 = \alpha 1\Delta t/(\Delta x1)^2$ - material 1

$A2 = \alpha 2\Delta t/(\Delta x2)^2$ - material 2

$B1 = \alpha 1\Delta t/(\Delta y)^2$ - material 1

$B2 = \alpha 1\Delta t/(\Delta y)^2$ - material 2

$C1 = H1/\rho 1 C_p 1 T_B$ = heat distribution function (f(y)) for material 1

$C2 = H2/\rho 2 C_p 2 T_B$ = heat distribution function (g(y)) for material 2

$$E = \frac{k2\Delta x2}{k1\Delta x1}$$

$$F = [k2\Delta x1/k1\Delta x2]$$

$$G = \frac{k2\Delta x2\alpha 1}{k1\Delta x1\alpha 2}$$

$$\Phi = \left[C1 + \left(\frac{k2\Delta x2\alpha 1}{k1\Delta x1\alpha 2}\right) C2\right] \Big/ \left[1 + \frac{k2\Delta x2\alpha 1}{k1\Delta x1\alpha 2}\right]$$

$$\xi = \alpha 1 \left[1 + \frac{k2\Delta x2}{k1\Delta x1}\right] \Big/ \left[1 + \frac{k2\Delta x2\alpha 1}{k1\Delta x1\alpha 2}\right]$$

$$\delta u_{yy} = u_{I1,j-1} - 2u_{I1,j} + u_{I1,j+1}$$

APPENDIX B

FINITE DIFFERENCE EQUATIONS APPLYING AS BOUNDARY CONDITIONS AT THE INTERFACE BETWEEN MATERIALS 1 AND 2

I. CONTINUOUS FLUX AND TEMPERATURE AT THE INTERFACE

Both temperature and heat flux must be continuous at an interface assumed to be in good thermal contact. Using the nomenclature adopted in this report, this is equivalent to saying that

(1) u^*_{I1} is continuous

and

(2)
$$k1 \frac{(u^*_{I1,j} - u^*_{I1-1,j})}{\Delta x1} = k2 \frac{(u^*_{I1+1,j} - u^*_{I1,j})}{\Delta x2} \quad . \qquad (8)$$

Equation (8) can be used directly in the tridiagonal matrix since only the terms $u^*_{I1-1,j}$, $u^*_{I1,j}$, $u^*_{I1+1,j}$ are involved. Therefore, by rearranging Eq. (8), the coefficients a_{I1}, b_{I1}, c_{I1}, and d_{I1} can be specified as:

TABLE A-1

DIFFERENCE EQUATIONS FOR COMPOSITE (X-FIRST)

Difference Equation	Condition	Range	Comments
<u>begin x-sweep</u>			
1. $u^*_{1,j} = u^*_{0,j}$	left boundary material 1	$j = 1,\ldots,J-1$ $i = 1,2$	Symmetry (no flux)
2. $u^*_{i,j} - u_{i,j} = \frac{A1}{2}(u^*_{i+1,j} - 2u^*_{i,j} + u^*_{i,j-1})$ $+ \Delta t C1 + \frac{B1}{2}(u_{i,j+1} - 2u_{i,j} + u_{i,j-1})$	material 1	$j = 1,\ldots,J-1$ $i = 1,\ldots,I1-1$	PDE, implicit x
3a. $(u^*_{I1,j} - u^*_{I1-1,j})\frac{k1}{\Delta x1} = (u^*_{I1+1,j} - u^*_{I1,j})\frac{k2}{\Delta x2}$	Interface	$j = 1,\ldots,J-1$	a. Cont. flux
3b. $u^*_{I1,j} = u_{I1,j} + \Phi\Delta t + \xi u_{yy}\frac{\Delta t}{2\Delta y^2}$ $+ \frac{A1\,(u^*_{I1-1,j} + (1+F)\,u^*_{I1,j} + (F)\,u^*_{I1+1,j})}{2(1+G)}$	Interface	$j = 1,\ldots,J-1$ $i = I1$	b. Cont. flux and PDE apply
4. $(u^*_{i,j} - u_{i,j}) = \frac{A2}{2}(u^*_{i+1,j} - 2u^*_{i,j} + u^*_{i-1,j})$ $+ \Delta t C2 + \frac{B2}{2}(u_{i,j+1} - 2u_{i,j} + u_{i,j-1})$	material 2	$j = 1,\ldots,J-1$ $i = I1+1,\ldots,I2-1$	PDE, implicit x
5. $u^*_{I2,j} = u^*_{I2-1,j}$	right boundary material 2	$j = 1,\ldots,J-1$ $i = I2, I2-1$	symmetry (no flux)
<u>begin y-sweep (no heat source term)</u>			
6. $km(u^{**}_{i,1} - u^{**}_{i,0}) = \Delta y q^*_m/T_B$	material 1 or 2 $m = 1,2$	$i = 1,\ldots,I1-1,$ $I1 + 1,\ldots I2-1$ $j = 0,1$	inside boundary (incident fixed heat flux) ($q^* \cong q_r$ rest time) ($q^* = q_p$ pulse time)
7a. $u^{**}_{i,j} - u^*_{i,j} = \frac{Am}{2}(u^*_{i+1,j} - 2u^*_{i,j}$ $+ u^*_{i-1,j}) + \frac{Bm}{2}(u^{**}_{i,j+1} - 2u^{**}_{i,j}$ $+ u^{**}_{i,j-1})$	$m = 1,2$	$i = 1,\ldots,I1-1,$ $I1 + 1,\ldots,I2-1$ $j = 1,\ldots,J$	materials 1 or 2 excluding interface and right & left boundaries.
7b. $u^{**}_{I1,j} = u^*_{I1,j} + \frac{A1}{(1+G)}(u^*_{I1-1,j}$ $- (1+F)\,u_{I1,j} + (F)\,u_{I1+1,j})$ $+ \frac{\xi\Delta t}{2\Delta y^2}(u^{**}_{I1,j+1} - 2u^{**}_{I1,j} + u^{**}_{I1,j-1})$	interface	$i = I1$ $j = 1,\ldots,J$	PDE implicit y applies at interface if Eq. (3b) is used
8. $-km\,(u^{**}_{i,J} - u^{**}_{i,J-1}) = \Delta yh\,(u^{**}_{i,J})$	$m = 1,2$	$i = 1,\ldots,I1-1,$ $I1+1,\ldots,I2-1$ $j = J-1,J$	outside boundary (temp. dependent flow with liq. metal heat transfer coeff.)

TABLE A-2

DIFFERENCE EQUATIONS FOR COMPOSITE (Y-FIRST)

Difference equation	Conditions	Range	Comments
begin y-sweep			
1. $km\,(u^*_{i,1} - u^*_{i,0}) = \Delta y\; q_m/T_B$	materials 1 or 2 m=1,2	i=1,...I1-i, I1+1, ...,I2-1 j=1,0	inside boundary (incident fixed heat flux) (q^* =qr for rest time) (q^* =qp for pulse period)
2a. $u^*_{i,j} = u_{i,j} = \frac{Am}{2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j})$ $+ \Delta t\, Cm + \frac{Bm}{2}(u^*_{i,j+1} - 2u^*_{i,j} + u^*_{i,j-1})$	m = 1,2	i=1,...,I1-1,I1+1, ...,I2-1 j=1,...,J	material 1 or 2 (excluding interface and left boundaries)
2b. $u^*_{I1,j} = u_{I1,j} + \frac{CI + GC2}{(1 + G)}\Delta t$ $+ \frac{A1}{(1+G)}(u_{I1-1,j} - (1+F)u_{I1,j} + (F)u_{I1+1,j})$ $+ \frac{\xi\Delta t}{2\Delta y^2}(u^*_{I1,j-1} - 2u^*_{I1,j} + u^*_{I1,j+1})$	interface	i = I1	applies at interface if Eq. (6b) is used
3. $km\,(u^*_{i,J} - u^*_{i,J-1}) = \Delta y h(u^*_{i,J})$	m = 1,2	i = 1,...I1-1, I1+1 ...,I2-1 j = J-1,J	outside boundary (temperature dependent flux with liquid metal heat transfer coeff.)
begin x-sweep (no heat source term)			
4. $u^{**}_{i,j} = u^{**}_{0,j}$	material 1 left boundary	j = 1,...,J-1 i = 0,1	symmetry (no flux)
5. $u^{**}_{i,j} - u^*_{i,j} = \frac{A1}{2}(u^{**}_{i+1,j} - 2u^{**}_{i,j}$ $+ u^{**}_{i-1,j}) + \frac{B1}{2}(u^*_{i,j+1} - 2u^*_{i,j} + u^*_{i,j-1})$	material 1	j = 1,...,J-1 i = 1,...,I1-1	PDE, implicit x
6a. $(u^{**}_{I1,j} - u^{**}_{I1-1,j})\frac{k1}{\Delta x1} = (u^{**}_{I1+1,j}$ $- u^{**}_{I1,j})\frac{k_2}{\Delta x2}$	interface	j = 1,...,J-1 i = I1	a. continuous flux
6b. $u^{**}_{I1,j} = u^*_{I1,j} + \frac{\xi\delta u^*_{yy}\Delta t}{2\Delta y^2} + \frac{A1}{2\,(1+G)}$ $(u^{**}_{I1-1,j} + (1+F)\,u^{**}_{I1,j} + (F)\,u^{**}_{I1+1,j})$		j=1,...,J-1 i = I1	b. continuous flux and PDE
7. $u^{**}_{i,j} - u^*_{i,j} = \frac{A2}{2}(u^{**}_{i+1,j} - 2u^{**}_{i,j}$ $+ u^{**}_{i-1,j}) + \frac{B2}{2}(u^*_{i,j+1} - 2u^*_{i,j} + u^*_{i,j-1})$	material 2	j = 1,...,J-1 i = I1+1,...I2-1	PDE, implicit x
8. $u^{**}_{I2,j} = u^{**}_{I2-1,j}$	material 2 right boundary	j = 1,...,J-1 i = I2-1,I2	symmetry (no flux)

$$a_{I1} = -1$$

$$b_{I1} = 1 + \frac{k2\Delta x1}{k1\Delta x2}$$

$$c_{I1} = -\frac{k2\Delta x1}{k1\Delta x2}$$

$$d_{I1} = 0 \quad . \tag{9}$$

The stability and convergence of the ADI procedure appeared to depend on the choice of $\Delta x1$ and $\Delta x2$ for a given k1 and k2. If values of $\Delta x2$ were selected such that

$$\frac{k1}{\Delta x1} \cong \frac{k2}{\Delta x2} \quad , \tag{10}$$

the ADI technique was convergent and stable. Consequently, an alternate form of the interface condition was developed to keep the PDE itself continuous at the interface.

II. CONTINUOUS FLUX AND TEMPERATURE WITH MODIFIED PDE AT THE INTERFACE

By utilizing the technique suggested by Carnahan, Luther, and Wilkes,[7] one can develop appropriate finite difference equations for the boundary between material 1 and 2 for our case. Following the conventions of the model, the dimensionless temperature at position I1-1 in material 1 can be approximated by a Taylor expansion as

$$u_{I1-1,j} \cong u_{I1,j} - \Delta x1 \left(\frac{\partial u}{\partial x}\right)_{I1^-} + \frac{(\Delta x1)^2}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)_{I1^-} + \ldots \tag{11}$$

by solving Eq. (11) for $(\partial^2 u/\partial x^2)_{I1^-}$, one gets

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{I1^-} \cong \frac{2}{(\Delta x1)^2}\left[u_{I1-1,j} - u_{I1,j} + \Delta x1 \left(\frac{\partial u}{\partial x}\right)_{I1^-}\right] \quad . \tag{12}$$

Using the finite difference equation for $(\partial^2 u/\partial y^2)$ and $\partial u/\partial t$

$$(\partial^2 u/\partial y^2) \cong \frac{1}{\Delta y^2}\left[u_{I1,j+1} - 2u_{I1,j} + u_{I1,j-1}\right] \tag{13}$$

$$(\partial u/\partial t) \cong \frac{1}{\Delta t}\left[u^*_{I1,j} - u_{I1,j}\right]$$

$$u^* \text{ at new time } t + \Delta t \quad . \tag{14}$$

Likewise for material 2, Eqs. (11), (12), (13), and (14) can be rewritten as,

$$u_{I1+1,j} \cong u_{I1,j} + \Delta x2 \left(\frac{\partial u}{\partial x}\right)_{I1^+} + \frac{(\Delta x2)^2}{2}\left(\frac{\partial^2 u}{\partial x^2}\right)_{I1^+} \tag{15}$$

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{I1^+} \cong \frac{2}{(\Delta x2)^2}\left[u_{I1+1,j} - u_{I1,j} - \Delta x2 \left(\frac{\partial u}{\partial x}\right)_{I1^+}\right] \tag{16}$$

$$\left(\frac{\partial^2 u}{\partial y^2}\right) \cong \frac{1}{\Delta y^2}\left[u_{I1,j+1} - 2u_{I1,j} + u_{I1,j-1}\right] \tag{17}$$

$$\left(\frac{\partial u}{\partial t}\right) = \frac{1}{\Delta t}\left(u^*_{I1,j} - u_{I1,j}\right) \quad . \tag{18}$$

By substituting into the differential equation,

$$\alpha\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + C = \frac{\partial u}{\partial t} \quad ,$$

one can develop an expression for $\partial u/\partial t$ at the interface. For medium 1, using Eqs. (12), (13), and (14)

$$\alpha 1\left[\frac{2}{(\Delta x1)^2}\left(u_{I1-1,j} - u_{I1,j} + \Delta x1 \left(\frac{\partial u}{\partial x}\right)_{I1^-}\right) + \frac{1}{\Delta y^2}\left(u_{I1,j+1} - 2u_{I1,j} + u_{I1,j-1}\right)\right] + C1 = \left(u^*_{I1,j} - u_{I1,j}\right)\Big/ \Delta t \quad . \tag{19}$$

Solving for $(\partial u/\partial x)_{I1^-}$, by defining

$$\delta u_{yy} \equiv u_{I1,j-1} - 2u_{I1,j} + u_{I1,\,j+1} \quad ,$$

Eq. (19) becomes

$$\Delta x1 \left(\frac{\partial u}{\partial x}\right)_{I1^-} = \frac{(\Delta x1)^2}{2\alpha 1 \Delta t}\left(u^*_{I1,j} - u_{I1,j}\right) - \frac{(\Delta x1)^2\, C1}{2\alpha 1} - \frac{(\Delta x1)^2}{2(\Delta y)^2}\,\delta u_{yy} + u_{I1,j} - u_{I1-1,j} \quad . \tag{20}$$

Similarly for medium 2, using Eqs. (16), (17), and (18)

$$-\Delta x2 \left(\frac{\partial u}{\partial x}\right)_{I1^+} = \frac{(\Delta x2)^2}{2\alpha 2 \Delta t}\left(u^*_{I1,j} - u_{I1,j}\right) - \frac{(\Delta x2)^2}{2\alpha 2}\, C2 - \frac{(\Delta x2)^2}{2(\Delta y)^2}\,\delta u_{yy} + u_{I1,j} - u_{I1+1,j} \quad . \tag{21}$$

Applying the interface condition of continuous flux, viz,

$$k1\left(\frac{\partial u}{\partial x}\right)_{I1^-} = k2\left(\frac{\partial u}{\partial x}\right)_{I1^+} \quad . \tag{22}$$

We can use Eqs. (20), (21), and (22) to eliminate $\left(\frac{\partial u}{\partial x}\right)_{I1^-}$ and $\left(\frac{\partial u}{\partial x}\right)_{I1^+}$ by just rearranging Eqs. (20 and (21).

$$k1\left(\frac{\partial u}{\partial x}\right)_{I1^-} = \frac{k1\Delta x1}{2\alpha 1\Delta t}\left(u^*_{I1,j} - u_{I1,j}\right) - \frac{k1\Delta x1 C1}{2\alpha 1} - \frac{k1\Delta x1}{2(\Delta y)^2}\,\delta u_{yy} + \frac{k1}{\Delta x1}\left(u_{I1,j} - u_{I1-1,j}\right) \tag{23}$$

$$k2\left(\frac{\partial u}{\partial x}\right)_{I1^+} = -\frac{k2\Delta x2}{2\alpha 2\Delta t}\left(u^*_{I1,j} - u_{I1,j}\right) + \frac{k2\Delta x2 C2}{2\alpha 2} + \frac{k2\Delta x2}{2(\Delta y)^2}\,\delta u_{yy} - \frac{k2}{\Delta x2}\left(u_{I1,j} - u_{I1+1,j}\right) \quad . \tag{24}$$

Equations (23) and (24) can be used to solve for $u^*_{I1,j}$.

$$\begin{aligned}\left[\frac{k1\Delta x1}{2\alpha 1\Delta t} + \frac{k2\Delta x2}{2\alpha 2\Delta t}\right] u^*_{I1,j} &= \left[\frac{k1\Delta x1}{2\alpha 1\Delta t} + \frac{k2\Delta x2}{2\alpha 2\Delta t}\right] u_{I1,j} \\ &+ \left[\frac{k1\Delta x1 C1}{2\alpha 1} + \frac{k2\Delta x2 C2}{2\alpha 2}\right] \\ &+ \left[\frac{k1\Delta x1}{2(\Delta y)^2} + \frac{k2\Delta x2}{2(\Delta y)^2}\right]\delta u_{yy} \\ &- \frac{k1}{\Delta x1}\left[u_{I1,j} - u_{I1-1,j}\right] \\ &- \frac{k2}{\Delta x2}\left[u_{I1,j} - u_{I1+1,j}\right] \quad .\end{aligned} \tag{25}$$

By simplifying Eq. (25),

$$u^*_{I1,j} = u_{I1,j} + C\Delta t + \frac{\alpha\Delta t\,\delta u_{yy}}{\Delta y^2} + \left[\frac{u_{I1-1,j} - u_{I1,j}\left[1 + \frac{k2\Delta x1}{k1\Delta x2}\right] + u_{I1+1,j}\left[\frac{k2\Delta x1}{k1\Delta x2}\right]}{\frac{(\Delta x1)^2}{2\alpha 1\Delta t}\left[1 + \frac{k2\Delta x2}{k1\Delta x1}\,\frac{\alpha 1}{\alpha 2}\right]}\right] \tag{26}$$

where

$$\Phi \equiv \left[C1 + \frac{k2\Delta x2\,\alpha1}{k1\Delta x1\,\alpha2}C2\right] \Bigg/ \left[1 + \frac{k2\Delta x2\,\alpha1}{k1\Delta x1\,\alpha2}\right] \quad (27)$$

$$\xi = \alpha1\left[1 + \frac{k2\Delta x2}{k1\Delta x1}\right] \Bigg/ \left[1 + \frac{k2\Delta x2\,\alpha1}{k1\Delta x1\,\alpha2}\right] . \quad (28)$$

Equation (26) is similar to the explicit difference equation presented by Arpaci.[2]

For the case of no heat generation, $C1 = C2 = 0$; $\Delta x1 = \Delta x2 = \Delta x$; and only one direction dependence for u, i.e., $\delta u_{yy} = 0$, u^* becomes

$$u^*_{I1,j} = u_{I1,j} + \frac{2\alpha1\Delta t}{\Delta x^2}\,\frac{\left[u_{I1-1,j} - u_{I1,j}\left(1 + \frac{k2}{k1}\right) + u_{I1+1,j}\left(\frac{k2}{k1}\right)\right]}{\left[1 + \frac{k2\alpha1}{k1\alpha2}\right]} . \quad (29)$$

By multiplying the numerator and denominator of the second term on the right-hand side of Eq. (29) by k1/k2 and rearranging, one gets,

$$u^*_{I1,j} = u_{I1,j} + \frac{2\alpha1\Delta t}{\Delta x^2}\,\frac{\left[u_{I1+1,j} - u_{I1,j}\left(1 + \frac{k1}{k2}\right) + u_{I1-1,j}\left(\frac{k1}{k2}\right)\right]}{\left[\frac{k1}{k2} + \frac{\alpha1}{\alpha2}\right]} , \quad (30)$$

which corresponds to Eq. (7.67) presented by Carnahan et al.[7] on page 463. If both materials are the same, $\alpha1 = \alpha2 = \alpha$; $k1 = k2 = k$ and,

$$u^*_{I1,j} = u_{I1,j} + \frac{\alpha\Delta t}{\Delta x^2}\left(u_{I1+1,j} - 2u_{I1,j} + u_{I1-1,j}\right) , \quad (31)$$

which is in standard explicit form for a homogeneous system.

Using implicit formulation in order to implement this algorithm in the current ADI code, one can show that

$$u^*_{I1,j} = u_{I1,j} + \Phi^*\Delta t + \frac{\delta u_{yy}\,(\Delta t/2)\,\xi^*}{\Delta y^2}$$

$$+ \frac{\alpha1\Delta t/2}{(\Delta x1)^2}\,\frac{\left[u^*_{I1-1,j} - u^*_{I1,j}\left(1 + \frac{k2\Delta x1}{k1\Delta x2}\right) + u^*_{I1+1,j}\left(\frac{k2\Delta x1}{k1\Delta x2}\right)\right]}{\left[1 + \frac{k2\Delta x2\,\alpha1}{k1\Delta x1\,\alpha2}\right]} \quad (32)$$

with $\Phi = \dot{\psi}^*$, $\xi^* = \xi$.

(Note that again the heat source Φ^* is put in with full Δt, and $\Delta t/2$ is used for other time intervals.)

To determine the coefficients for the tridiagonal matrix, viz., a_{I1}, b_{I1}, c_{I1}, d_{I1}, we define the following quantities.

$$E \equiv \frac{k2\Delta x2}{k1\Delta x1} ; \quad F \equiv \frac{k2\Delta x1}{k1\Delta x2} ; \quad G \equiv \frac{k2\Delta x2\,\alpha1}{k1\Delta x1\,\alpha2} . \quad (33)$$

Note that $\delta u_{yy} = u_{I1,j-1} - 2u_{I1,j} + u_{I1,j+1}$ is defined at the old time t rather than $t + \Delta t$.

The first three terms on the right-hand side of Eq. (32) are used to specify d_{I1}, while the fourth term specifies a_{I1}, b_{I1}, and c_{I1}, along with the left-hand side of Eq. (32). Consequently,

$$a_{I1} = \frac{-2\alpha1\Delta t/2}{(\Delta x1)^2(1 + G)} \quad (34)$$

$$b_{I1} = 1 + \frac{2\alpha1\Delta t/2\,(1 + F)}{(\Delta x1)^2\,(1 + G)} \quad (35)$$

$$c_{I1} = \frac{-2\alpha1\Delta t/2(F)}{(\Delta x1)^2\,(1 + G)} \quad (36)$$

$$d_{I1} = u_{I1,j} + \frac{\Delta t\ (C1 + GC2)}{(1 + G)} + \frac{\Delta t \alpha 1\ (1 + E)}{2(1 + G)\Delta y^2}\left[u_{I1,j-1} - 2u_{I1,j} + u_{I1,j-1}\right] \qquad (37)$$

(in the Madcap code $\alpha 1$ = D1 and $\alpha 2$ = D2) .

In the ADI scheme, we also need an equation to allow us to implicitly calculate $u_{I1,j}$ at the interface when sweeping in the y-direction. Since Eq. (25) is an equivalent form of the PDE applying at i = I1 (interface), it can be rewritten implicit in y and explicit in x. Equation (26) thus can be restructured as

$$u^*_{I1,j} = u_{I1,j} + \Phi\Delta t + \frac{\xi\Delta t}{2\Delta y^2}\left[u^*_{I1,j-1} - 2u^*_{I1,j} + u^*_{I1,j+1}\right]$$

$$+ \frac{2\alpha 1\Delta t/2}{(\Delta x1)^2}\left[\frac{u_{I1-1,j} - u_{I1,j}\left[1 + \frac{k2\Delta x1}{k1\Delta x2}\right] + u_{I1+1,j}\left[\frac{k2\Delta x1}{k1\Delta x2}\right]}{\left[1 + \frac{k2\Delta x2\,\alpha 1}{k1\Delta x1\,\alpha 2}\right]}\right] , \qquad (38)$$

which is similar to Eq. (32). Again we can solve for the tridiagonal coefficients using Eq. (33) to define terms.

$$u^*_{I1,j} = u_{I1,j} + \frac{\Delta t\left[C1 + GC2\right]}{(1 + G)}$$

$$+ \frac{\Delta t\,\alpha 1}{(\Delta x1)^2\ (1 + G)}\left[u_{I1-1,j} - (1 + F)\,u_{I1,j} + (F)u_{I1+1,j}\right]$$

$$+ \frac{\xi\Delta t}{2\Delta y^2}\left[u^*_{I1,j-1} - 2u^*_{I1,j} + u^*_{I1,j+1}\right] \qquad (39)$$

$$\xi = \frac{\alpha 1\ (1 + E)}{(1 + G)} \qquad (40)$$

$$a_{I1} = -\frac{\xi\Delta t}{2\Delta y^2} = -\frac{\alpha 1\ (1 + E)\Delta t}{(1 + G)(2\Delta y^2)} \qquad (41)$$

$$b_{I1} = 1 + \frac{\xi\Delta t}{\Delta y^2} = 1 + \frac{\alpha 1(1 + E)\Delta t}{(1 + G)\Delta y^2} \qquad (42)$$

$$c_{I1} = -\frac{\xi\Delta t}{2\Delta y^2} = -\frac{\alpha 1(1 + E)\ \Delta t}{(1 + G)(2\Delta y^2)} \qquad (43)$$

$$d_{I1} = \frac{\Delta t\,\alpha 1}{(\Delta x1)^2\ (1 + G)}\left[u_{I1-1,j} - (1 + F)\,u_{I1,j} + (F)u_{I1+1,j}\right] + u_{I1,j} + \frac{\Delta t\ (C1 + GC2)}{(1 + G)} \qquad (44)$$

APPENDIX C

ALTERNATING DIRECTION IMPLICIT METHOD (ADI)

The implementation of the ADI method as discussed in Appendix A has been considered by numerous authors (7,9,10,11), and consequently only a brief discussion is included here. The ADI technique when applied to a rectangular grid network avoids the step size limitations of an explicit method and also uses a tridiagonal coefficient matrix for rapid calculation of the temperature grid at any time step. The basic concept is to use two difference equations, each applied at half Δt steps.

Each difference equation is implicit in either the x or y direction. For example, solving the two-dimensional elliptic equation

$$\alpha \left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right] = \frac{\partial u}{\partial t} \tag{45}$$

would involve iterations using difference equations of the following form for an (i,j) grid. The x-sweep [implicit in x] is written as

$$\frac{u^*_{i,j} - u_{i,j}}{\Delta t/2} = \frac{(u^*_{i-1,j} - 2u^*_{i,j} + u^*_{i+1,j})}{\Delta x^2} + \frac{(u_{i,j-1} - 2u_{i,j} + u_{i,j+1})}{\Delta y^2} , \tag{46}$$

and the y-sweep [implicit in y] as

$$\frac{u^{**}_{i,j} - u^*_{i,j}}{\Delta t/2} = \frac{u^*_{i-1,j} - 2u^*_{i,j} + u^*_{i+1,j}}{\Delta x^2} + \frac{(u^{**}_{i,j-1} - 2u^{**}_{i,j} + u^{**}_{i,j+1})}{\Delta y^2} , \tag{47}$$

where

$u_{i,j}$ = value of $u_{i,j}$ at time t

$u^*_{i,j}$ = value of $u_{i,j}$ at $t + \Delta t/2$ (half time step)

$u^{**}_{i,j}$ = value of $u_{i,j}$ at $t + \Delta t$ (full time step).

Richtymer and Morton[3] have demonstrated that this form of the ADI method is unconditionally stable regardless of the choice of Δx, Δy, or Δt. Our particular problem has three additional complications:

(1) A heat source term C is present [Eq. (1)].

(2) An interface between two materials is present.

(3) The inside boundary condition is time dependent (pulsed flux).

All of the above can induce instabilities and/or inadequate convergence unless the difference equations applying at the interface and boundaries are properly formulated. (See Appendix B.) Consistency for the difference equations has been demonstrated if the heat source term is introduced at the full time step, i.e., $C\Delta t$ is introduced in either the x or y sweep and not at both half-time steps.[5] Systematic errors due to this procedure were eliminated by altering the sweeping sequence to xyyxxyyx ... .

APPENDIX D

FORMULATION OF THE TRIDIAGONAL ALGORITHM

The ADI technique inherently generates equations for each grid point involving 3 adjacent terms in the u matrix.

$$u_{i-1,j},\ u_{i,j},\ u_{i+1,j}$$

or

$$u_{i,j-1},\ u_{i,j},\ u_{i,j+1}\ . \tag{48}$$

The coefficients a,b,c refer to i-1 (j-1), i(j), and i+1(j+1) terms, respectively, while d refers to the remaining terms. Furthermore the a,b,c coefficients would be for terms involving the new time step either u^* or u^{**} (see Table I). Thus, the tridiagonal matrix can be represented as

$$\begin{bmatrix} b_0Z_0 & c_0Z_1 & & & \\ a_1Z_0 & b_1Z_1 & c_1Z_2 & & \\ \cdots & \cdots & \cdots & & \\ & a_iZ_{i-1} & b_iZ_i & c_iZ_{i+1} & \\ & & & & \\ & & & a_nZ_{n-1} & b_nZ_n \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ \cdots \\ d_i \\ \vdots \\ d_n \end{bmatrix} . \tag{49}$$

[Z] refers either to $u_{i,j}$, j fixed, or $u_{i,j}$, i fixed. The algorithm for solving the tridiagonal matrix is relatively straightforward. The matrix is swept from top to bottom and then from bottom to top to solve for [Z]. The following flow sheet depicts this procedure.[7]

TRIDIAGONAL PROCEDURE

START

$$\beta_0 = b_0 \qquad \gamma_0 = \frac{d_0}{\beta_0}$$

for i = 1 to N

$$\beta_i = b_i - \frac{(a_i\, c_{i-1})}{\beta_{i-1}}$$

$$\gamma_i = \frac{(d_i - a_i\, \gamma_{i-1})}{\beta_i}$$

$$Z_N = \gamma_N$$

for j = N-1 to 0

$$Z_j = \gamma_j - \frac{(c_j Z_{j+1})}{\beta_j}$$

STOP

APPENDIX E

MADCAP V LISTING

```
26 Jul 73 0926+41
                          Rec 01 Page 01
                  |  |  |  |  |  |  |  |  |  |  |
"CTR COMPOSITE HEAT FLOW MODEL"
"ALTERNATING DIRECTION IMPLICIT METHOD USED"
"Pulsed case"
"Isotropic and homogeneous properties assumed for each material"
"modified Code with continous interface condition"
"Variable Specification"
     "T = temperature, °C"
     "T_B = bulk lithium temperature, °C"
     "Cp = heat capacity, cal/g°C"
     "p = density, g/cm^3"
     "h = heat transfer coefficient, cal/cm^2sec°C"
     "k = thermal conductivity, cal/cm sec°C"
     "D = thermal diffusivity = k/PC_p, cm^2/sec"
     "Tp = burn time for pulse, Micro-sec"
     "Tr = test time, Micro-sec"
     "Δx1 = x-step size in Material 1"
     "Δx2 = x-step size in Material 2"
     "Δy = y-step size"
     "Δt = step size for time"
     "Time = actual time, sec"
     "Tprint = interval between prints Micro-sec"
     "Y_w = Wall thickness, cm"
     "L1 = size of material 1 element, cm"
     "L2 = size of material 2 element, cm"
     "sub or postscripts 1 and 2 refer to two different Materials"
   "  sub or postscript 3 refers to average value at interface"
```

```
                     |    |    |    |    :    |    |    |    |    |    |
01,01b          "Differential Equation (Rectangular coordinates)"

01,01c               "D(d²u/dx²+d²u/dy²)+C(y) = du/dt"

01,01d          "Dimensionless parameters"

01,01e               "u = (T-T_B)/T_B"

01,01f               "A = DΔt/Δx²"

01,020               "B = DΔt/Δy²"

01,021               "CΔt = HΔt/ρCp"

01,022               "QΔY/R = incident heat flux"

01,023          "where:"

01,024          "Postscripts 1 and 2 refer to two different materials"

01,025          "postscripts r and p refer to rest and burn periods"

01,026          "For example,"

01,027          " H is the internal heat generation term, it can take on"

01,028          "values: Hr1(y),Hp1(y),Hr2(y),Hp2(y)"

01,029          "Likewise for Q: Qr1,Qp1,Qr2,Qp2"

01,02A          "u* = dimensionless temperature at 1/2 time step"

01,02b          "u** = dimensionless temperature at complete time step"
```

26 Jul 73 0926+48

Rec 02 Page 01

```
02,001    "sense 1 - on for trial data set"
02,002    "sense 2 - on for print out at each"
02,003    "sense 3 - on set generation terms to zero"
02,004    "sense 4 - on to set up plots"
02,005    "sense 5 - on to terminate the iteration"
02,006    "sense 6 - on to terminate iteration and initial plotting"
02,007    "sense 7 - on ask for new print interval"
02,008    "sense 8 - on to use old interface condition at I1"
02,009    "          -k1(du/dx) = -k2(du/dx) in finite difference form"
02,00a    "    off to use rodified interface condition at I1"
02,00b    "          Continuous flux and PDE apply"
02,00c    "If cont. flux and PDE are used at the interface then the"
02,00d    "interface is included in the y sweep"
02,00e    "sense 5 - on to use harmonic mean for interface,"
02,00f    "    off for arithmetic area average"
02,010    u,ue,uee  0 to 110,0 to 110
02,011    Z,a,b,c,d  0 to 110
02,012    Gr1,Gr2,Gp1,Gp2,Hr1,Hr2,Hp1,Hp2  0 to 110
02,013    Ω,Y  0 to 110
02,014    "Array assignment for plots"
02,015    "ΔT = T-T_L, °C"
02,016    "ΔT1 = inside surface (plasma) temp. rise for material 1 at (i=1,j=0)"
02,017    "ΔT2 = inside surface (plasma) temp. rise for material 2 at (i=I2-1,j=0)"
02,018    "ΔT3 = inside surface (plasma) temp. rise at interface (i=I1,j=0)"
02,019    "ΔT4 = outside surface (lithium) temp. rise for material 1 (i=1,j=J)"
02,01a    "ΔT5 = outside surface (lithium) temp. rise for material 2 (i=I2-1,j=J)"
02,01b    "ΔT6 = outside surface (lithium) temp. rise at interface (i=I1,j=J)"
```

```
02,01c    ΔT1,ΔT2,ΔT3,ΔT4,ΔT5,ΔT6,t_0 to 500
02,01d    Cal_0 to 2000
02,01e    Yaxis_0 to 500
02,01f    D629_0 to 10
02,020    v = 0
02,021    t_0,ΔT1_0,ΔT2_0,ΔT3_0,ΔT4_0,ΔT5_0,ΔT6_0 = 0
02,022    (200 characters) Comp1,Comp2
02,023    for i = 0 to 110
02,024        a_i,b_i,c_i,d_i,Z_i = 0
02,025        ΔT1_i,ΔT2_i,ΔT3_i,ΔT4_i,ΔT5_i,ΔT6_i = 0
02,026        for j = 0 to 110
02,027            u_i,j,u*_i,j,u**_i,j = 0
02,028    fo is "            Temperature Profiles (T-T_B , °C)        "
```

```
26 Jul 73 0926+55
                               Rec 03 Page 01

                     |   |   |   |   |   |   |   |   |   |   |
03,001               f1 is "i =     0         1         I1/2      I1        (I1+I2)/2    I2-1
03,002               f2 is " j"
03,003               Tstop = 1000000
03,004               if sense 1 is on         "Trial data set"
03,005                    read console by " Δt = x    Δx1 = x    Δx2 = x    Δy = x ":Δt,Δx1,Δx2,Δy
03,006                    read console by "L1 = x L2 = x Ywall = x": L1,L2,Y_w
03,007                    read console by " k1= x   k2 = x ":k1,k2
03,008                    read console by "Cp1= x   Cp2= x ":Cp1,Cp2
03,009                    read console by " p1= x  p2 = x ":p1,p2
03,00a                    read console by " D1= x  D2 = x ":D1,D2
03,00b                    read console by " h = x":h
03,00c                    read console by "Qr1 = x Qr2 = x "
03,00d   cont.                 "Qp1 = x Qp2 = x": Qr1,Qr2,Qp1,Qp2
03,00e                    read console by " COMP1 = x    COMP2 = x ":Comp1,Comp2
03,00f                    read console by "Tprint(micro-sec) = x": Tprint
03,010                    if sense 4 is on
03,011                         read console by "Tstop=x": Tstop
03,012               otherwise              "input data"
03,013                    Δt=1000; Δx1=.05; Δx2=.005; Δy=.02
03,014                    L1 = 1 ; L2 = .1 ; Y_w = 1
03,015                    k1=.158; k2=.034
03,016                    Cp1=.0731; Cp2=.178
03,017                    p1=8.57; p2=3.96
03,018                    D1=.25; D2=.0434
03,019                    h = .14
03,01a                    Qr1 = 23.82 ; Qr2 = 23.82
03,01b                    Qp1=238.2; Qp2=238.2
03,01c                    Comp1 = "Nb"
```

```
03,01d            Comp2 = "Al2O3   small h = .14 cal/cm^2sec°C "

03,01e            Tprint = 10000

03,01f            Tstop = 3000000

03,020      T_s = 600             "°C"

03,021      T_p = 10000           "Micro-sec"

03,022      T_r = 90000           "Micro-sec"

03,023      Ip = [(T_p/Δt)]

03,024      It = [((T_p+T_r)/Δt)]

03,025      I1 = [(L1/(2Δx1) + .5)]
03,026      I2 = I1 + [(L2/(2Δx2) + .5)]
03,027      J = [(Y_w/Δy + .5)]

03,028      Δt = .000001 Δt            "conversion to sec from micro sec"

03,029      A1 = D1(Δt/Δx1^2)

03,02a      A2 = D2(Δt/Δx2^2)

03,02b      B1 = D1(Δt/Δy^2)

03,02c      B2 = D2(Δt/Δy^2)
03,02d      if sense 9 is on          "harmonic mean"
03,02e           k3 = (2k1*k2)/(k1+k2)
```

```
otherwise            "arithmetic area average"
     k3 = (k1×Δx1+k2×Δx2)/(Δx1+Δx2)
Index = 1
Delta = [(.000001 × Tprint/Δt + .5)]
"internal heat generation functions as arrays"
if sense 3 is off
     read card by "(d10)5": Points,Fract1,Fract2,Fract3,Fract4
     new card
     for i = 0 to Points
          read card by "(d10)5": y_i,Qr1_i,Qp1_i,Qr2_i,Qp2_i
          new card
     Degree = 2
     i0 = 1
     for j = 0 to J
          ȳ = j(Δy)
          for i = i0 to Points
               if y_i ≥ ȳ
                    execute lagran(j,i,Degree,ȳ)
                    i0 = i
                    exit from loop
               otherwise: loop back
otherwise
     Fract1 = 1
     Fract2 = 1
     Fract3 = 1
     Fract4 = 1
     for j = 0 to J
```

```
                 |    |    |    |    |    |    |    |    |    |    |
                          Hr1j,Hp1j,Hr2j,Hp2j = 0

             "Conversion from percent absorption to heat,cal/cm3sec "
             for j = 0 to J
                  Hr1j = Qr1*Hr1j/ΔY
                  Hr2j = Qr2*Hr2j/ΔY
                  Hp1j = Qp1*Hp1j/ΔY
                  Hp2j = Qp2*Hp2j/ΔY
             Qr1 = Fract1*Qr1
             Qr2 = Fract2*Qr2
             Qp1 = Fract3*Qp1
             Qp2 = Fract4*Qp2
             for i = 0 to I2            "Initial condition ui,j = 0"
                  for j = 0 to J
                       ui,j,u*i,j,u**i,j = 0
             Time = 0
             "begin. of iterations for each time period At as n=1 to infinity"
             "Code will proceed with one of two algorithms"
             "    1 - if x and y Profiles are important, 2-D ADI"
             "    is used with entire heat source added at one"
             "    half time step, and iteration sequence altered"
             "    as XYYXXYYX in sweeping x and y arrays,"
             "    2 - if composite has very small x dimensions,"
             "    i.e. if L1 and L2 are small compared to the"
             "    thermal diffusion depths, only the y direction"
             "    is used in the code, and a unidirectional ADI"
             "    is run with average property values used"
             "Test for parabolic (2D) or unidirectional dependence"
```

26 Jul 73 0937?1b

Rac PS Page 01

```
                 1    1    1    1    1    1    1    1    1    1    1
05.001      Tum1 = (L1/2)^2/D1
05.002      Tum2 = (L2/2)^2/D2
05.003      Tvr1 = r_w^2/D1 [illegible]
05.004      Tvr2 = r_w^2/D2
05.005      if sense A is off or (n1=2) and (D1=D2) and (ks1=ks2)
05.006          index = 2J-1        "includes interface in computation"
05.007      otherwise
05.008          index = J1          "excludes interface"
05.009      for n = 2 to infinity
05.00a          if Model(1) = 1          "Parabolic AD? (2D) x and y directions"
05.00b              if index ≤ 1?        "Counter to determine if in pulse or rest mode"
05.00c                  q1 = Qp1
05.00d                  q2 = Qp2
05.00e                  q3 = ([illegible])/([illegible])
05.00f              otherwise
05.010                  q1 = Qr1
05.011                  q2 = Qr2
05.012                  q3 = ([illegible])/([illegible])
05.013              if n is even              "sweep x first"
05.014                  execute sweep(0)
05.015                  for j = 1 to J-1
05.016                      if index ≤ 1?
05.017                          C1 = Rp1_j/(p1*cp1*T_p)     "pulse period"
05.018                          C2 = Rp2_j/(p2*cp2*T_p)
05.019                      otherwise
05.01a                          C1 = Rr1_j/(p1*cp1*T_r)          "rest period"
```

```
05.01b            Cz = [illegible]
05.01c        for i = 1 to [illegible]
05.01d            execute eqtvo([illegible])
05.01e        execute [illegible]
05.01f        for i = [illegible] to [illegible]
05.020            execute eqtvo([illegible])
05.021        execute [illegible]
05.022        execute [illegible]
05.023        for i = 0 to [illegible]
05.024            [illegible]
05.025    for j = 1 to [illegible]                  "begin y sweep"
05.026        if i < [illegible]                    "material 1"
05.027        if i = [illegible]           "interface"
05.028        if i > [illegible]                        "material 2"
05.029        execute [illegible]
05.02a        for j = 1 to [illegible]
05.02b            if i = [illegible]
05.02c                execute [illegible]
05.02d            otherwise
05.02e                execute [illegible]
05.02f        execute [illegible]
```

```
                    execute std(J;a,b,c,d,2)
                    for j = 0 to J
                         w*o_i,j = z_j
          otherwise          "[illegible]"
               for i = 1 to I2-1 + 1 + iomit
                    if i < I1: A=A1;B=B1;K=K1;q=q1        "material 1"
                    if i = I1: q=q3;K=K3          "interface"
                    if i > I1: A=A2;B=B2;K=K2;q=q2          "material 2"
                    execute eqsix(0,Δy,q,K,T_B)
                    for j = 1 to J-1
                         if Index ≤ Ip
                              C1 = Rp1_j/(p1×Cp1×T_B)
                              C2 = Rp2_j/(p2×Cp2×T_B)
                         otherwise
                              C1 = Rr1_j/(p1×Cp1×T_B)
                              C2 = Rr2_j/(p2×Cp2×T_B)
                         if i < I1: C = C1
                         otherwise: C = C2
                         if i ≠ I1
                              execute eqtwo(i,j,B,A,C,Δt,0)
                         otherwise
                              execute eqseven(I1,j,K1,K2,D1,D2,C1,C2,ΔX1,ΔX2,Δy,Δt,1)
                    execute eqeight(J,Δy,h,x)
                    execute std(J,a,b,c,d,2)
                    for j = 0 to J
                         w*_i,j = z_j
               for j = 1 to J-1          "begin of x sweep"
```

```
                  execute eqone(0)
                  for i = 1 to I1-1
                       execute eqfour(i,j,A1,B1,0,0,1)
                  execute eqthree(I1,j,K1,K2,D1,D2,0,0,Δx1,Δx2,ΔY,Δt,1)
                  for i = I1+1 to I2-1
                       execute eqfour(i,j,A2,B2,0,C,1)
                  execute eqfive(I2)
                  execute std(I2;a,b,c,d,2)
                  for i = 0 to I2
                       u**_i,j = z_i
       "Missing values for u at (i=0,I1,I2; j=0,J)are assigned via B.C.'s"
       "These are not used in the computation of u(x,y,t)"
       u**_0,0 = u**_1,0
       u**_0,J = u**_1,J
       u**_I2,0 = u**_I2-1,0
       u**_I2,J = u**_I2-1,J
       if k2 ≠ k1            "interface values at j=0,J"
            Phi = (k2*Δx1)/(k1*Δx2)
            u**_I1,0 = ((Phi)u**_I1+1,0 + u**_I1-1,0)/(1 + Phi)
```

```
24 Jul 73 0837:30
                          Sec 07 Page 01

07.001                    u**_{ii,j} = ((phi_i u**_{ii+1,j} + u**_{ii-1,j})/(1 + 2phi)
07.002              for i = 0 to II
07.003                  for j = 0 to J
07.004                      u_{i,j} = u**_{i,j}
07.005          otherwise            "Unidirectional [illegible] dependence"
07.006              A = (A1(L1)+A2(L2))/(L1+L2)        "average properties"
07.007              B = (B1(L1)+B2(L2))/(L1+L2)
07.008              K = (K1(L1)+K2(L2))/(L1+L2)
07.009              if index ≤ 2P
07.00a                  q1 = Q_p1
07.00b                  q2 = Q_p2
07.00c              otherwise
07.00d                  q1 = Q_r1
07.00e                  q2 = Q_r2
07.00f              q = (q1(L1)+q2(L2))/(L1+L2)
07.010              execute eqsix(0,Δy,q,K,T_s)
07.011              for j = 1 to J-1
07.012                  if index < 3P
07.013                      C1 = Hp1_j/(p1*Cp1*T_s)
07.014                      C2 = Hp2_j/(p2*Cp2*T_s)
07.015                  otherwise
07.016                      C1 = Hr1_j/(p1*Cp1*T_n)
07.017                      C2 = Hr2_j/(p2*Cp2*T_n)
07.018                  C = (C1(L1)+C2(L2))/(L1+L2)
07.019                  execute eqtwo(1,j,2B,0,C,Δt,0)
07.01a              execute eqeight(J,Δy,h,K)
07.01b              execute std(J,a,h,c,d,2)
```

```
26 Jul 73 0937•35
                         Sec 07 Page 02

07,01c            for i = 0 to I2
07,01d                 for j = 0 to J
07,01e                      u_i,j = T_j
07,01f       if index = It: index=1
07,020       otherwise: index=index+1           "end of dt period"
07,021       if sense 2 is on; interval = 0
07,022       otherwise: interval = 10
07,023       if (n0-1) = interval(mod delta)
07,024                                                    "OUTPUT"
07,025            Time = (n0-1)dt
07,026            if sense b is on        "set up arrays for plotting"
07,027                 v = v+1
07,028                 t_v = Time
07,029                 ΔT1_v = u_0,0*T_B
07,02a                 ΔT2_v = u_I2,0*T_B
07,02b                 ΔT3_v = u_I1,0*T_B
07,02c                 ΔT4_v = u_0,J*T_B
07,02d                 ΔT5_v = u_I2,J*T_B
07,02e                 ΔT6_v = u_I1,J*T_B
07,02f            if [(100000_0 Time)] = [(Tprint)]
07,030                 new page
07,031                 print: date
07,032                 skip 2 lines
07,033                 print: "CTR COMPOSITE FIRST WALL"
07,034                 skip 2 lines
07,035                 if model() = 1
07,036                      Print: "Two dimensional ADI (x and y)"
07,037                 otherwise
07,038                      Print: "Unidirectional ADI (y only)"
07,039                      Print: "average property values used"
```

```
26 Jul 73 0927:40
                                   Rec 08 Page 01
                     |    |    |    |    |    |    |    |    |    |    |
08.001                              skip 1 line
08.002                              print: "heat generation functions for pulse and rest mode"
08.003                              for j = 0 to J
08.004                                  print by "j=xx  yj=x.xx  Hr1=x.xx+ee  "
08.005   cont.                                "Hr2=x.xx+ee  Hp1=x.xx+ee  "
08.006   cont.                                "Hp2=x.xx+ee": j,j*dy,Hr1j,Hr2j,Hp1j,Hp2j
08.007                          new page
08.008                          print: date
08.009                          skip 4 lines
08.00a                          print: "CTR COMPOSITE FIRST WALL"
08.00b                          skip 2 lines
08.00c                          if model(1) = 1
08.00d                              print: "Two dimensional ADI (x and y)"
08.00e                          otherwise
08.00f                              print: "Unidirectional ADI (y only)"
08.010                              print: "average property values used"
08.011                          skip 1 line
08.012                          if sense 8 is on
08.013                              print: "cont. flux interface condition"
08.014                          otherwise
08.015                              print: "cont. flux and PDE at interface"
08.016                          if sense 9 is on
08.017                              print: "harmonic mean for k at interface"
08.018                          otherwise
08.019                              print: "arithmetic area average for k at interface"
08.01a                          print by " conductor(1) =  x       insulator(2) = x":Comp1,Comp2
08.01b                          skip 1 line
08.01c                          print: "incident flux = k(dT/dy)y = 0"
```

```
26 Jul 73 0927+45
                                  Rec 08 Page 02

08.01d                            print by "    pulse period qp1=x4.x4 qp2=x4.x4"
08.01e   cont.                        "cal/sec cm^2": Qp1,Qp2
08.01f                            print by "    rest period qr1=x4.x4 qr2=x4.x4"
08.020   cont.                        "cal/sec cm^2": Qr1,Qr2
08.021                            skip 1 line
08.022                            print by "wall thickness (y direction) = x3.x4 cm":Yw
08.023                            print by "element size material 1 (x direction) = x3.x4 cm":L1
08.024                            print by "element size material 2 (x direction) = x3.x6 cm":L2
08.025                            skip 1 line
08.026                            print by "Δt = x5 micro-sec ":[(1000000Δt + .5)]
08.027                            print by "Δx1 = x4.x5 cm ":Δx1
08.028                            print by "Δx2 = x4.x5 cm ":Δx2
08.029                            print by "Δy  = x4.x5 cm ":Δy
08.02a                            skip 1 line
08.02b                            print by "D1Δt/Δx1^2=x4.x5": A1
08.02c                            print by "D2Δt/Δx2^2=x4.x5": A2
08.02d                            print by "D1Δt/Δy^2 =x4.x5": B1
```

```
26 Jul 73 0927+49
                                  Rec 09 Page 01
                 |   |   |   |   |   |   |   |   |   |   |
09,001                            print by "D2Δt/Δy² =x4.x5": β2
09,002                            skip 1 line
09,003                            print by "Pulse time = x.x5 sec  rest time = x.x5 sec":
09,004   cont.                        Tp/1000000,Tr/1000000
09,005                            skip 1 line
09,006                            print by "for material 1 k= x3.x5 cal/sec cm°C  "
09,007   cont.                        "Cp= x3.x5 cal/g°C  ρ= x3.x4 g/cm³  "
09,008   cont.                        "D1= x3.x5 cm²/sec": k1,Cp1,ρ1,D1
09,009                            skip 1 line
09,00a                            print by "for material 2 k= x3.x5 cal/sec cm°C  "
09,00b   cont.                        "Cp= x3.x5 cal/g °C  ρ= x3.x4 g/cm³  "
09,00c   cont.                        "D2= x3.x5 cm²/sec": k2,Cp2,ρ2,D2
09,00d                            skip 2 lines
09,00e                            print by "  h = x3.x4     effective heat transfer coeff,  cal/cm²sec°C ":h
09,00f                            skip 1 line
09,010                            print by "Grid Size = [x(material 1) = x3 points, "
09,011   cont.                        "x(material 2) = x3 points] by [y = x3 points]":I1,(I2-I1),J
09,012                            new page
09,013                            print by "Time= x Micro-sec ": 1000000 Time
09,014                            skip 2 lines
09,015                            print : f0
09,016                            print: f1
09,017                            print: f2
09,018                            skip 2 lines
09,019                            I76 = [(I1/2)]
09,01a                            I77 = [((I1+I2)/2)]
```

```
                         |    |    |    |    |    |    |    |    |    |    |
09.01b                            for j = 0 to J
09.01c                                 print by "xxx..(..sx.xh+ee)7..": j,
09.01d   cont.                              u_0,j*T_B, u_1,j*T_B, u_I76,j*T_B,
09.01e   cont.                              u_I1,j*T_B, u_I77,j*T_B, u_I2-1,j*T_B,
09.01f   cont.                              u_I2,j*T_B
09.020                            if sense 7 is on
09.021                                 read console by "Tprint=x": Tprint
09.022                                 Delta = [(.000001*Tprint/Δt+.5)]
09.023                            if sense 6 is on
09.024                                 Tstop = 0
09.025                       if 1000000*Time > Tstop and sense 4 is on
09.026                            "plotting routine"
09.027                            for m = 1 to 6
09.028                                 for j = 1 to w
09.029                                      if m = 1: Yaxis_j = ΔT1_j
09.02a                                      if m = 2: Yaxis_j = ΔT2_j
09.02b                                      if m = 3: Yaxis_j = ΔT3_j
09.02c                                      if m = 4: Yaxis_j = ΔT4_j
09.02d                                      if m = 5: Yaxis_j = ΔT5_j
09.02e                                      if m = 6: Yaxis_j = ΔT6_j
09.02f                                 D629_m = MAX_n=1 to w(Yaxis_n)
09.030                            Tmax = MAX_n = 1 to 6(D629_n)
```

26 Jul 73 0927+59

Rec 0a Page 01

```
S629 = "  TIME MICRO-SEC"
T629 = " T-T_n (C) "
U629 = "COMPOSITE-PULSED CASE"
execute cprime(Ca1,2000,1,12)
execute csymbol(,1,9.5,,14,U629,0)
execute csymbol(,1,9.2,,14,Comp1,0)
execute csymbol(,1,9.0,,14,Comp2,0)
if Tmax <500 : Tmax = 500
otherwise: Tmax = 1000
Tstop = 1000000*Time
execute cscaler(0,Tstop,0,Tmax,0,10,0,10)
execute cplot(0,Tmax,3)
execute cplot(Tstop,Tmax,2)
execute cplot(Tstop,0,2)
execute caxis(0,0,0,10,S629,17)
execute caxis(0,0,90,-10,T629,-10)
for n = 1 to 6
     Symb = n
     for m = 0 to w
          if n = 1: Yaxis_m = ΔT1_m
          if n = 2: Yaxis_m = ΔT2_m
          if n = 3: Yaxis_m = ΔT3_m
          if n = 4: Yaxis_m = ΔT4_m
          if n = 5: Yaxis_m = ΔT5_m
          if n = 6: Yaxis_m = ΔT6_m
     Q629 = 3
     for m = 0 to w
```

```
0a,01c                                   execute cplot(1000000t_m,Yaxis_m,q629)
0a,01d                                   q629 = 2
0a,01e                              execute chump((Ystop+.02),Yaxis_u,,14,Symb,0,0)
0a,01f                         execute cempty(1,4)
0a,020                         stop #1
0a,021                    if sense 5 is on, stop
0a,022               "Procedures"
0a,023               "eqone thru eqeight generate coefficients for the tridiagonal matrix"
0a,024               "    eqone - left hand boundary-material 1 (x-direction)"
0a,025               "    eqtwo - material 1 or 2, PDE at even At/2"
0a,026               "    eqthree - interface condition at I1 (x-direction)"
0a,027               "    eqfour - material 1 or 2, PDE at odd At/2"
0a,028               "    eqfive - right hand boundary-material 2 (x-direction)"
0a,029               "    eqsix - inside(plasma) side boundary (y-direction)"
0a,02a               "    eqseven - interface condition (PDE) for y sweep"
0a,02b               "    eqeight - outside, liquid metal heat transfer coeff, (y direction)"
0a,02c               "Std solves the tridiagonal matrix"
0a,02d               "lagran generates a lagrangian interpolation polynominal for"
0a,02e               "estimating discrete values of the heat generation term"
```

26 Jul 73 0928+06

Rec Ob Page 01

```
          |    |    |    |    |    |    |    |    |    |    |
Ob,001          "Model() determines with a 2-D or unidirectional solution"
Ob,002          "will be used"
Ob,003   (...   eqone(n;all)
Ob,004          a_n=0;b_n=1;c_n=-1;d_n=0
Ob,005   ...)
Ob,006   (...   eqtwo(n,m,r,s,C,Δt,Test;all)
Ob,007          (array)u
Ob,008          if Test = 1
Ob,009               x = (u_{n,m+1}-2u_{n,m}+u_{n,m-1})
Ob,00a               k* = n
Ob,00b          otherwise
Ob,00c               x = (u_{n+1,m}-2u_{n,m}+u_{n-1,m})
Ob,00d               k* = n
Ob,00e          a_{k*} = -r/2
Ob,00f          b_{k*} = 1+r
Ob,010          c_{k*} = -r/2
Ob,011          d_{k*} = C×Δt+(s/2)(x)+u_{n,m}
Ob,012   ...)
Ob,013   (...   eqthree(I1,j,k1,k2,D1,D2,C1,C2,Δx1,Δx2,Δy,Δt,Test2;all)
Ob,014          (array)u,u*
Ob,015          if sense 8 is on        "continuous flux at interface"
Ob,016               a_{I1} = -1
Ob,017               b_{I1} = 1+(k2×Δx1)/(k1×Δx2)
Ob,018               c_{I1} = -(k2×Δx1)/(k1×Δx2)
Ob,019               d_{I1} = 0
Ob,01a          otherwise          "continuous flux and PDE apply at interface"
```

```
ob,01b          E = (K2×Δx2)/(K1×Δx1)
ob,01c          F = (K2×Δx1)/(K1×Δx2)
ob,01d          G = (K2×Δx2×D1)/(K1×Δx1×D2)
ob,01e          a_I1 = -(D1×Δt)/(Δx1^2(1+G))
ob,01f          b_I1 = 1 + (D1×Δt(1+F))/(Δx1^2(1+G))
ob,020          c_I1 = -(D1×Δt×F)/(Δx1^2(1+G))
ob,021          if Test2 = 0
ob,022               H = D1(1+E)×(Δt/2)×(u_I1,j-1 -2u_I1,j +u_I1,j+1)/(Δy^2)
ob,023               H2 = u_I1,j
ob,024          otherwise
ob,025               H = D1(1+E)×(Δt/2)×(u*_I1,j-1 -2u*_I1,j +u*_I1,j+1)/(Δy^2)
ob,026               H2 = u*_I1,j
ob,027          d_I1 = H2+(Δt(C1+G×C2)+H)/(1+G)
ob,028    ...)
ob,029    (...   eqfour(n,m,r,s,C,Δt,Test:all)
ob,02a           (array)u*
ob,02b           if Test = 1
ob,02c               x = (u*_n,m+1 -2u*_n,m +u*_n,m-1)
ob,02d               k* = n
ob,02e           otherwise
ob,02f               x = (u*_n+1,m -2u*_n,m +u*_n-1,m)
ob,030               k* = m
ob,031           a_k* = -r/2
ob,032           b_k* = 1+r
```

```
24 Jul 73 0928+16
                              Rec 0b Page 03
                    |    |    |    |    |    |    |    |    |    |    |
0b,033              c_ke = -r/2
0b,034              d_ke = CHAt*s(x)/2+ue_B,n
0b,035     ...)
0b,036     (...     eqfive(I2;all)
0b,037              a_I2 = -1;b_I2=1;c_I2=0;d_I2=0
0b,038     ...)
0b,039     (...     eqsix(n,Ay,q,k,T_B;all)
0b,03a              a_n = 0; b_n = 1; c_n = -1; d_n = (q*Ay)/(k*T_B)


24 Jul 73 0928+16
                              Rec 0c Page 01
                    |    |    |    |    |    |    |    |    |    |    |
0c,001     ...)
0c,002     (...     eqseven(I1,J,K1,X2,D1,D2,C1,C2,Ax1,Ax2,Ay,At,Test3;all)
0c,003              (array)u,ue
0c,004              E = (K2*Ax2)/(K1*Ax1)
0c,005              F = (K2*Ax1)/(K1*Ax2)
0c,006              G = (K2*Ax2*D1)/(K1*Ax1*D2)
0c,007              H = D1(1+E)/(1+G)
0c,008              P = (At*D1)/(Ax1^2(1+G))
0c,009              if Test3 = 1
0c,00a                   H = P(u_I1-1,J-(1+F)u_I1,J+(F)u_I1+1,J)
0c,00b                   H2 = u_I1,J
0c,00c              otherwise
0c,00d                   H = P(ue_I1-1,J-(1+F)ue_I1,J+(F)ue_I1+1,J)
0c,00e                   H2 = ue_I1,J
0c,00f              a_I1 = -(H*At)/(2Ay^2)
0c,010              b_I1 = 1+(H*At)/(Ay^2)
0c,011              c_I1 = -(H*At)/(2Ay^2)
0c,012              d_I1 = H2+H*At(C1+G*C2)/(1+G)
0c,013     ...)
0c,014     (...     eqeight(n,Ay,h,k;all)
0c,015              a_n = -1;b_n=1+Ay*h/k;c_n=0;d_n=0
0c,016     ...)
0c,017     (...     std(n;a,b,c,d,Z)          "Tridiagonal matrix algorithm"
0c,018              (array)a,b,c,d,Z
0c,019              B,G_0 to 110
```

```
0c,01a          for m = 0 to 110
0c,01b               B_m,G_m = 0
0c,01c          B_0 = b_0
0c,01d          G_0 = d_0/B_0
0c,01e          for m = 1 to n
0c,01f               B_m = b_m-a_m c_{m-1}/B_{m-1}
0c,020               G_m = (d_m-a_m G_{m-1})/B_m
0c,021          Z_n = G_n
0c,022          for m = n-1,n-2,...,0
0c,023               Z_m = G_m-c_m Z_{m+1}/B_m
0c,024    ...)
0c,025    (...  lagran(j,i,Degree,ȳ;k,l))
0c,026          (array)Hp1,Hp2,Hr1,Hr2,Gp1,Gp2,Gr1,Gr2,z,y
0c,027          for k = 1 to 4
0c,028               if k=1: z=Gp1
0c,029               if k=2: z=Gp2
0c,02a               if k=3: z=Gr1
0c,02b               if k=4: z=Gr2
0c,02c               c* = 1
0c,02d               for j* = i-1 to i+Degree-1
0c,02e                    if ȳ = y_{j*}
0c,02f                         Hp1_j = Gp1_{j*}
0c,030                         Hp2_j = Gp2_{j*}
0c,031                         Hr1_j = Gr1_{j*}
0c,032                         Hr2_j = Gr2_{j*}
0c,033                         exit from procedure
0c,034                    otherwise: c*=c*(ȳ-y_{j*})
0c,035               z̄ = 0
0c,036               for i* = i-1 to i+Degree-1
0c,037                    t* = c*z_{i*}/(ȳ - y_{i*})
0c,038                    for j* = i-1 to i+Degree-1
0c,039                         if i* = j*: loop back
0c,03a                         t* = t*/(y_{i*}-y_{j*})
0c,03b                    z̄ = z̄+t*
0c,03c               if k=1: Hp1_j = z̄
0c,03d               if k=2: Hp2_j = z̄
```

```
26 Jul 73 0928+29
                                  Rec 0d Page 01
                  |    |    |    |    |    |    |    |    |    |    |
0d,001                 if k=3; Mr1_j = Ē

0d,002                 if k=4; Mr2_j = S̄

0d,003     ...)
0d,004     (...    model(none; all)
0d,005             Tbarx = (Tux1+Tux2)/2
0d,006             Tbary = (Tuy1 + Tuy2)/2
0d,007             if Tbarx < .01Tbary
0d,008                 model() = 0          "Unidirectional"
0d,009             otherwise
0d,00a                 model() = 1          "2-dimensional"
0d,00b     ...)
           $
```

APPENDIX F

NOMENCLATURE

Variable Specification

$A1 = \alpha 1 \Delta t/(\Delta x1)^2$

$A2 = \alpha 2 \Delta t/(\Delta x2)^2$

$B1 = \alpha 1 \Delta t/(\Delta y)^2$

$B2 = \alpha 2 \Delta t/(\Delta y)^2$

$C1 = H1/\rho 1 C_p 1 T_B$

$C2 = H2/\rho 1 C_p 1 T_B$

C_p = heat capacity, cal/g°C

$C(y) = H(y)/\rho C_p T_B$ designated as C1 or C2

D or α = thermal diffusivity = $k/\rho C_p$, cm^2/s

Δy = step size in both materials (y direction)

$\Delta x1$ = step size in material 1 (x direction)

$\Delta x2$ = step size in material 2 (x direction)

Δt = full time step

$\Delta T = T - T_B$ K or °C

$F = k2 \Delta x1/k1 \Delta x2$

h = heat transfer coefficient (liquid lithium, cal/cm^2 s °C)

H(y) = heat generation rate, cal/s cm^3, designated as Hr1, Hr2, Hp1, Hp2

I1 = number of grid pts in x-direction material 1

I2 = number of grid pts in x-direction material 2

J = number of grid pts in x-direction material 2

k = thermal conductivity, cal/s cm°C

L1 = size of element in material 1

L2 = size of element in material 2

ρ = density, g/cm^3

q or q_1 = incident flux on the inside surface

T = temperature, K or °C

T_B = bulk Lithium temp., K or °C

τ_p = burn time for pulse, μ s or m s

τ_r = rest time, μ s of m s

u = dimensionless temperature = $(T-T_B)/T_B$

$u^*_{i,j}$ = dimensionless temp. 1/2 time interval

$u^{**}_{i,j}$ = dimensionless temp. full-time interval

Subscripts or postscripts

1-material 1

2-material 2

r-rest period

p-pulse (burn) period

REFERENCES

1. J. W. Tester, R. C. Feber, and C. C. Herrick, "Heat Transfer and Chemical Stability Calculations for Controlled Thermonuclear Reactors (CTR)," Los Alamos Scientific Laboratory report LA-5328 MS (August 1973).
2. J. A. Phillips, private communication (July 1973).
3. S. C. Burnett, W. R. Ellis, T. A. Oliphant, and F. L. Ribe, "A Reference Theta Pinch Reactor (RTPR)," LA-5121-MS (December 1972).
4. T. A. Oliphant, private communication. (August 1973).
5. W. V. Green and F. L. Ribe, Los Alamos Scientific Laboratory report. Private communication (1972).
6. L. C. Ianniello (Ed.), "Fusion Reactor First Wall Materials." A'C, WASH 1206 (April 1972).
7. B. Carnahan, H. A. Luther, and J. A. Wilkes, "Applied Numerical Methods," Wiley, N. Y. (1969).
8. V. S. Arpaci, "Conduction Heat Transfer," Addison-Wesley, Reading, Mass. (1966), p. 511.
9. R. D. Richtmyer and K. W. Morton, "Difference Methods for Initial Value Problems," 2nd Ed. Wiley (Interscience), New York (1967).
10. D. W. Peaceman and H. H. Rachford, Jr., "The Numerical Solution of Parabolic and Elliptic Differential Equations," J. Soc. Indust. Appl. Math. 3 (1), 28 (1955).
11. J. Douglas, Jr., "On the Numerical Integration of $\partial^2 u / \partial x^2 + \partial^2 u/\partial y^2 = \partial u/\partial t$ by Implicit Methods," J. Soc. Indust. Appl. Math. 3 (1), 42 (1955).

10:268(40)