

DOT/FAA/AR-04/51

Office of Aviation Research
Washington, D.C. 20591

THERMOD, an Enhanced Thermal Model for Determining Aircraft Operational Temperatures

December 2004

Final Report

This document is available to the U.S. public
through the National Technical Information
Service (NTIS), Springfield, Virginia 22161.



U.S. Department of Transportation
Federal Aviation Administration

NOTICE

This document is disseminated under the sponsorship of the U.S. Department of Transportation in the interest of information exchange. The United States Government assumes no liability for the contents or use thereof. The United States Government does not endorse products or manufacturers. Trade or manufacturer's names appear herein solely because they are considered essential to the objective of this report. This document does not constitute FAA certification policy. Consult your local FAA aircraft certification office as to its use.

This report is available at the Federal Aviation Administration William J. Hughes Technical Center's Full-Text Technical Reports page: actlibrary.tc.faa.gov in Adobe Acrobat portable document format (PDF).

Technical Report Documentation Page

1. Report No. DOT/FAA/AR-04/51	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle THERMOD, AN ENHANCED THERMAL MODEL FOR DETERMINING AIRCRAFT OPERATIONAL TEMPERATURES		5. Report Date December 2004	
		6. Performing Organization Code	
7. Author(s) Nathan Govindarajoo, Ph.D, PE.		8. Performing Organization Report No.	
9. Performing Organization Name and Address 1889 Old Dixie Highway, #203 Vero Beach, FL 32960		10. Work Unit No. (TRAIS)	
		11. Contract or Grant No.	
12. Sponsoring Agency Name and Address U.S. Department of Transportation Federal Aviation Administration Office of Aviation Research Washington, DC 20591		13. Type of Report and Period Covered Final Report	
		14. Sponsoring Agency Code ACE-120, AGATE	
15. Supplementary Notes The FAA William J. Hughes Technical Center Technical Monitor was Peter Shyprykevich.			
16. Abstract An enhanced version of a thermal analysis computer program called THERMOD was developed for determining the maximum operating limit (MOL) temperatures of general aviation aircraft. The project was undertaken under a Federal Aviation Administration-AGATE sponsorship. This report is the second of two reports prepared in conjunction with this project. Enhancements included program debugging and corrections as well as development of an alternate implicit finite difference method, which is used in transient analysis. Numerical validation of THERMOD was undertaken with respect to the finite element methods. Good correlation was found. The validated THERMOD can be used to determine the MOL temperatures of a typical aircraft that has a low wing configuration.			
17. Key Words Maximum operating limit temperature, Thermal analysis, Conduction, Convection, Radiation, Steady state, Transient state		18. Distribution Statement This document is available to the public through the National Technical Information Service (NTIS), Springfield, Virginia 22161.	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 282	22. Price

TABLE OF CONTENTS

	Page
EXECUTIVE SUMMARY	ix
1. INTRODUCTION	1-1
1.1 Background	1-1
1.2 THERMOD Geometry and Input Data	1-3
1.3 Project Objectives	1-7
2. DEBUGGING AND PROGRAM CORRECTIONS	2-1
2.1 BUG 1	2-1
2.2 BUG 2	2-1
2.3 BUG 3	2-2
2.4 Algorithm Development for IBFDM	2-3
3. NUMERICAL VALIDATION OF THERMOD	3-1
3.1 Sample Problem 1	3-2
3.2 Sample Problem 2	3-17
3.3 Sample Problem 3	3-23
3.4 Sample Problem 4	3-39
3.5 Sample Problem 5	3-56
3.6 Sample Problem 6	3-74
4. SAMPLE PROBLEM	4-1
4.1 Input Data	4-1
4.2 Input File	4-5
4.3 Output Files	4-7
5. SUMMARY	5-1
6. REFERENCES	6-1
APPENDIX A—THERMOD CODE	

LIST OF FIGURES

Figure	Page
1-1 A Schematic Presentation of THERMOD	1-2
1-2 Overall Model Geometry and Typical Temperature Profiles at Critical Locations	1-4
1-3 Input Thermal Properties for THERMOD Model	1-5
1-4 A Typical Flight Profile During Transient Cooling	1-6
2-1 Control Volumes for Developing Thermal Equations	2-4
3-1 Contour Plot of the FEM Output of Sample Problem 1	3-16
3-2 Contour Plot of the FEM Output of Sample Problem 2	3-22
3-3 Contour Plot of the FEM Output of Sample Problem 3	3-37
3-4 Contour Plot of the FEM Output of Sample Problem 4	3-54
3-5 Contour Plot of the FEM Output of Sample Problem 5	3-72
3-6 Contour Plot of the FEM Output of Sample Problem 6	3-89

LIST OF TABLES

Table	Page
3-1 THERMOD Input File (input.dat) of Sample Problem 1	3-2
3-2 THERMOD Output File (summary.dat) of Sample Problem 1	3-4
3-3 THERMOD-Simulated Temperatures (°F) of the Right Wing of Sample Problem 1	3-7
3-4 Finite Element Method Input File of Sample Problem 1	3-8
3-5 Truncated FEM Output File of Sample Problem 1	3-16
3-6 Finite Element Method-Simulated Temperatures (°F) of the Right Wing of Sample Problem 1	3-17
3-7 THERMOD Output File (summary.dat) for Sample Problem 2	3-18
3-8 THERMOD-Simulated Temperatures (°F) of the Right Wing of Sample Problem 2	3-21
3-9 Truncated FEM Output File of Sample Problem 2	3-22
3-10 Finite Element Method-Simulated Temperatures (°F) of the Right Wing of Sample Problem 2	3-23

3-11	THERMOD Input File (input.dat) of Sample Problem 3	3-24
3-12	THERMOD Output File (summary.dat) for Sample Problem 3	3-25
3-13	THERMOD-Simulated Temperatures (°F) of the Right Wing of Sample Problem 3	3-29
3-14	Finite Element Method Input File of Sample Problem 3	3-29
3-15	Truncated FEM Output File of Sample Problem 3	3-38
3-16	Finite Element Method-Simulated Temperatures (°F) of the Right Wing of Sample Problem 3	3-39
3-17	THERMOD Input File (input.dat) of Sample Problem 4	3-40
3-18	THERMOD Output file (summary.dat) for Sample Problem 4	3-41
3-19	THERMOD-Simulated Temperatures (°F) of the Right Wing of Sample Problem 4	3-45
3-20	Finite Element Method Input File of Sample Problem 4	3-45
3-21	Truncated FEM Output File of Sample Problem 4	3-55
3-22	Finite Element Method-Simulated Temperatures (°F) of the Right Wing of Sample Problem 4	3-56
3-23	THERMOD Input File (input.dat) of Sample Problem 5	3-57
3-24	THERMOD Output File (summary.dat) for Sample Problem 5	3-58
3-25	THERMOD-Simulated Temperatures (°F) of the Right Wing of Sample Problem 5	3-62
3-26	Finite Element Method Input File of Sample Problem 5	3-63
3-27	Truncated FEM Output File of Sample Problem 5	3-73
3-28	Finite Element Method-Simulated Temperatures (°F) of the Right Wing of Sample Problem 5	3-74
3-29	THERMOD Input File (input.dat) of Sample Problem 6	3-74
3-30	THERMOD Output File (summary.dat) for Sample Problem 6	3-76
3-31	THERMOD-Simulated Temperatures (°F) of the Right Wing of Sample Problem 6	3-79

3-32	Finite Element Method Input File of Sample Problem 6	3-80
3-33	Truncated FEM Output File of Sample Problem 6	3-90
3-34	Finite Element Method-Simulated Temperatures (°F) of the Right Wing of Sample Problem 6	3-91
4-1	Temperature and Associated Radiation Data	4-1
4-2	Thickness of Thermal Elements of the Wing	4-2
4-3	Thickness of Thermal Elements of the Fuselage Side	4-2
4-4	Thickness of the Thermal Elements of the Floor	4-2
4-5	Thickness of the Thermal Elements of the Roof	4-3
4-6	Overall Dimensions of the Thermal Model	4-3
4-7	Absorptivity and Emissivity Properties of the Exterior Surfaces of the Wing and Fuselage	4-3
4-8	Thermal Properties of Various Solid Materials	4-3
4-9	Thermal Properties of Air	4-4
4-10	Miscellaneous Properties	4-4
4-11	Input File (input.dat) for the Sample Problem	4-6
4-12	Sample Problem Output Data File (summary.dat)	4-7
4-13	Sample Problem Output Data File (transient.dat)	4-11

LIST OF ACRONYMS

EFFDM	Explicit forward finite difference method
FAA	Federal Aviation Administration
FEM	Finite element method
IBFDM	Implicit backward finite difference method
MOL	Maximum operating limit
SPC	

EXECUTIVE SUMMARY

Composite aircraft structural elements are unfavorably affected by an increase in temperature due to exposure to the thermal environment. For design purposes, the synergistic effects of extreme ambient temperatures and the accompanying solar radiation should be taken into consideration in determining the maximum operating limit (MOL) temperatures experienced by the structural elements. Allowable design properties of composite materials may then be generated based on these MOL temperatures. THERMOD is a computer model that was developed for determining the MOL temperatures for aircraft under various paint schemes. The selected low-wing geometry is best suited to general aviation aircraft.

In determining the MOL temperatures, THERMOD considers the effects of radiation, convection, and conduction. Radiation includes direct solar radiation, infrared sky radiation, and their reflections from the tarmac. Infrared emission from the surrounding structures, including the tarmac, wings and the fuselage and its interaction among these structures, are also considered. Convection due to the wind, as well as turbulent convection within the cabin, is also simulated, as are one-dimensional conduction through element thickness. The model incorporates the effect of fillets at the wing-fuselage junction. Due consideration is also given to the greenhouse effect within the cabin, which allows for a realistic modeling of the thermal environment within the cabin. The effect of shade underneath the wing is also modeled.

The soaked temperatures of an aircraft are predicted based on steady-state assumptions, giving conservative steady-state temperatures. Because limit loads generally occur in flight conditions, a transient (unsteady-state) thermal analysis is used to simulate the thermal conditions while the aircraft executes the following maneuvers: taxi, takeoff, climb, and cruise. These maneuvers are cooling effects that in addition to intentional cooling of the cabin through opening the door simulate a more realistic thermal environment for predicting the MOL temperature.

THERMOD formulates a total of 67 independent equations within a nonlinear system of equations. The nonlinearity is introduced through radiation effects and through convective properties with the cabin, which are modeled as temperature-dependent. This system of equations is solved using the Newton-Raphson iteration technique.

This report is the third of four reports prepared in conjunction with this project. The fourth report is entitled "THERMOD User's Manual," and it will serve as an instructive guide for users who wish to use THERMOD for determining aircraft MOL temperatures. The first two reports validated THERMOD analyses with test data and studied the sensitivity to input variables.

The enhanced version of THERMOD was developed for simulating the MOL temperatures of general aviation aircraft. Enhancements included program debugging and corrections as well as development of an alternate implicit finite difference method. This method for solving the transient problem is unconditionally stable over all time and spatial domain as opposed to the existing explicit forward finite difference method. Numerical validation of THERMOD was undertaken with respect to the finite element methods. Good correlation was found.

1. INTRODUCTION.

An enhanced version of an original thermal analysis computer program called THERMOD [1] has been developed for simulating maximum operating limit (MOL) temperatures for general aviation aircrafts. This project was undertaken under a Federal Aviation Administration (FAA)-AGATE sponsorship.

This report is the third of four reports prepared in conjunction with this project. The fourth report, "THERMOD User's Manual," serves as a guide for users who wish to use THERMOD for determining aircraft MOL temperatures [2]. The first two reports validated THERMOD analyses with test data and studied the sensitivity to input variables [3 and 4].

To help the reader get acquainted with THERMOD, background information on pertinent thermal aspects of the program and input data requirements are presented. This is followed by a statement of the project objectives. A more complete description of THERMOD and the development of model equations can be found in the original THERMOD document [1].

1.1 BACKGROUND.

Structural components strength and sharpness of composite airframe are unfavorably affected by an increase in their temperatures due to exposure to the thermal environment. For design purposes, the combined effects of extreme ambient temperature and the accompanying solar radiation should be taken into consideration in determining the MOL temperatures experienced by the structural elements. Design properties of composite materials may then be generated based on these MOL temperatures. THERMOD was developed for determining the MOL temperatures of low-wing aircraft under various paint schemes.

In determining the MOL temperatures, THERMOD considers the effects of radiation, convection, and conduction. Radiation includes direct solar radiation, infrared sky radiation, and their reflections from the tarmac. Infrared emission from the surrounding elements, including the tarmac, wings, fuselage, and its interaction among these elements, are also considered. Convection due to the wind as well as turbulent convection within the cabin is also simulated, as are one-dimensional conduction through element thickness. The model incorporates the effect of fillets at the wing-fuselage junction. Due consideration is also given to the greenhouse effect within the cabin that allows for a realistic modeling of the thermal environment within the cabin. The effect of shade underneath the wing is also modeled.

In addition to the above factors, the following assumptions are made in THERMOD: (1) nonparticipating medium (air); (2) discretized space and time domains; (3) discretized elements being isothermal, opaque, diffuse, gray and characterized by uniform radiosity, irradiation, and material properties; (4) nonopaque materials such as windows and windshields are considered transparent with associated transmissivity values; and (5) constant material properties with respect to time (and, hence, temperature). These assumptions are necessary in simplifying the complexities involved in a three-dimensional thermal problem being addressed.

The soaked temperatures of an aircraft are predicted based on steady-state assumptions, giving conservative steady-state temperatures. Because limit loads generally occur in flight conditions,

a transient (unsteady-state) thermal analysis is used to simulate the thermal conditions while the aircraft executes the following maneuvers: taxi, takeoff, climb, and cruise. These maneuvers are cooling effects that in addition to intentional cooling of the cabin through opening the door simulate realistic thermal environment for predicting the MOL temperature.

THERMOD formulates a total of 67 independent equations within a nonlinear system of equations. The nonlinearity is introduced through radiation effects and through convective properties with the cabin, which are modeled as temperature-dependent. This system of equations is solved using the Newton-Raphson iteration technique [5].

As previously noted, THERMOD considers all three heat transfer mechanisms (convection, conduction, and radiation) normally associated with an aircraft parked in the open. Figure 1-1 is a schematic representation of a thermal environment showing short wavelength solar radiation, long wavelength sky (infrared) radiation, convective heat transfer due to the wind, and one-dimensional conduction (through the thickness). THERMOD also considers reflected solar energy between the fuselage and wing, diffused solar energy reflected from the tarmac, and infrared reflections. The fillets and greenhouse effects are considered as well. Because THERMOD simulates a total of 67 independent equations, 67 unknowns are needed. These unknowns are 53 temperatures and 14 radiosites. The 53 temperatures (T_1 through T_{53}) and their locations are noted in figure 1-1.

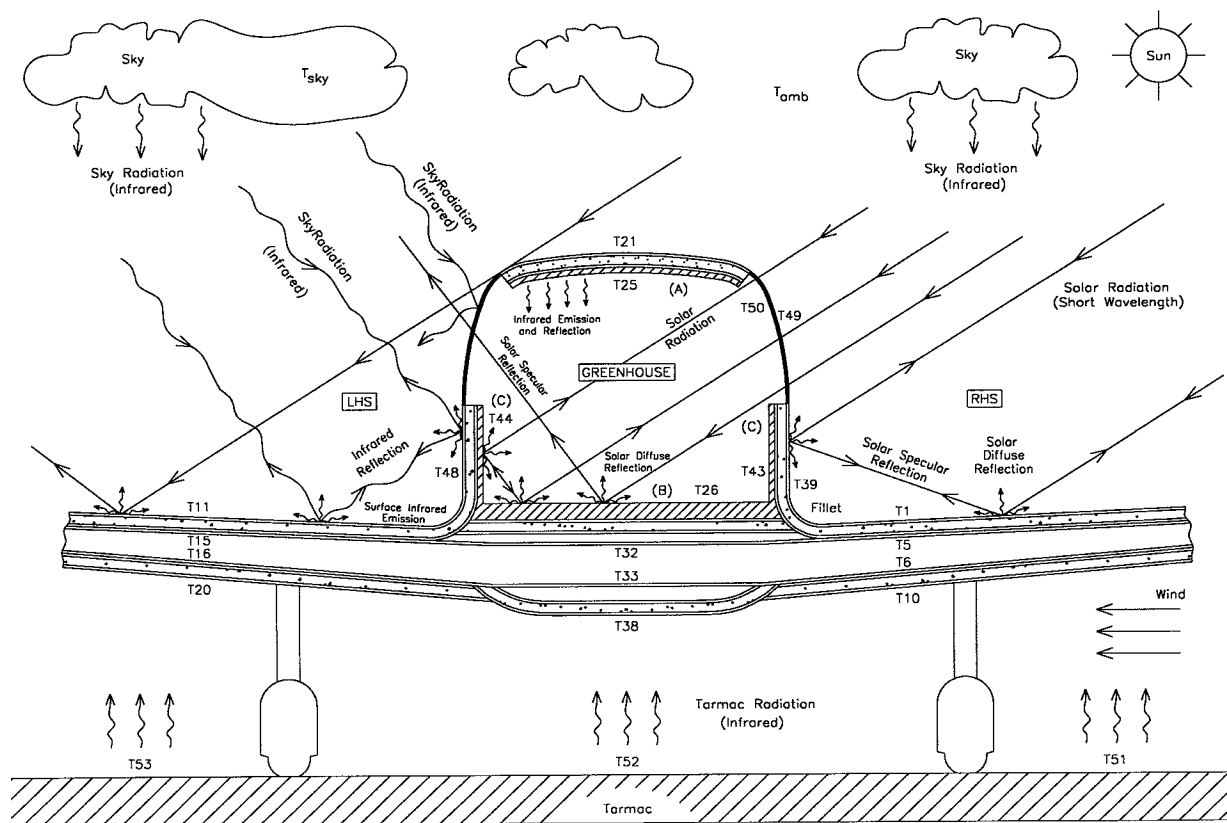


FIGURE 1-1. A SCHEMATIC PRESENTATION OF THERMOD

Note that not all the unknown temperatures are shown. T2, for example, is located immediately below T1 at the skin-core interface (see figure 1-2 for further clarity). Similarly, T27 is located immediately below T26 at the insulatory material B-composite floor interface.

The geometry of the aircraft shown is low wing with sandwich construction for skins. The sandwich can be either foam or honeycomb. Nonsandwich skins are also admissible if the thickness of the core is assumed to be very small.

A typical THERMOD analysis begins at the steady-state phase and continues on to the transient phase; a phase in which cooling is introduced due to aircraft maneuvers. At the point of application of limit or gust load, the temperatures at all 53 locations are noted. Of these 53 temperatures, 9 temperatures are considered nonstructural. These nonstructural temperatures are the three tarmac temperatures (T51, T52, and T53); four insulatory material temperatures (T25, T26, T43, and T44); and two transparent material temperatures (T49 and T50), leaving 44 temperatures to be considered structural.

THERMOD repeats its analysis over different time periods, as requested. The maximum temperature over these time periods is then reported as the MOL temperature. This MOL temperature may be located at the surface (depending on how dark the paint is), inside the cabin, within the floor space, or located anywhere else in the aircraft.

1.2 THERMOD GEOMETRY AND INPUT DATA.

Figure 1-1 showed a typical cross section of a small general aviation aircraft. A simplified rendering of the model is shown in figure 1-2. THERMOD is built on this simplified model. Figure 1-2 also shows typical sections at critical locations. In addition to the overall geometry as indicated in figure 1-2, a set of input data characterizing each layer, as shown in figure 1-3, is also needed. The layer properties include thickness, density, thermal conductivity, and specific heat capacity. To consider surface effects, all exposed aircraft surfaces are assigned their respective absorptivity and emissivity values. These values are dependent on paint color and surface texture. The tarmac is also considered a surface, whose surface and other thermal properties are required as well. THERMOD models the greenhouse and fillet effects. Transmissivity and fillet color information is needed to consider these two effects. The degree of greenhouse effect is also dependent on the percentage of transparent material. This information is input in the form of transparent surface area in relation to the overall cabin surface area. To simulate convective coefficients, the kinematic viscosity and Prandtl number are needed. Because the aircraft is left out in the open, it is subjected to the environment. The climatic data of the environment, for each time period, includes ambient temperature, sky temperature, solar radiation, and wind speed.

The final piece of information concerns the cooling that takes place when the aircraft executes the following maneuvers: taxi, takeoff, climb, and cruise. This data was furnished through a flight profile that is unique to a particular aircraft and is part of the input. The flight profile provides information on typical aircraft maneuvering speed with respect to time, from which, the time-varying convective coefficient is determined. This coefficient is then used in the subsequent transient cooling process. An example flight profile is shown in figure 1-4.

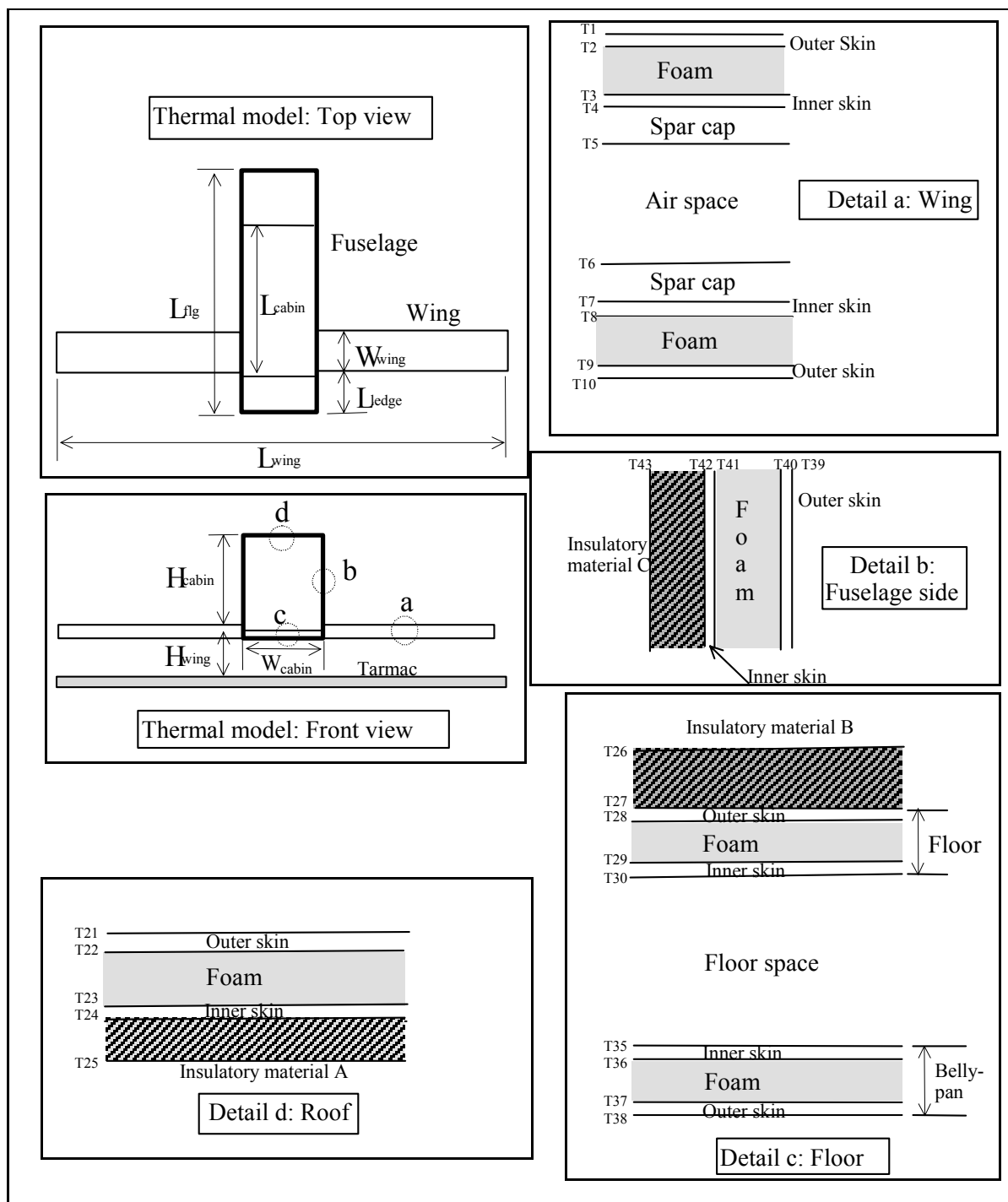


FIGURE 1-2. OVERALL MODEL GEOMETRY AND TYPICAL TEMPERATURE PROFILES AT CRITICAL LOCATIONS

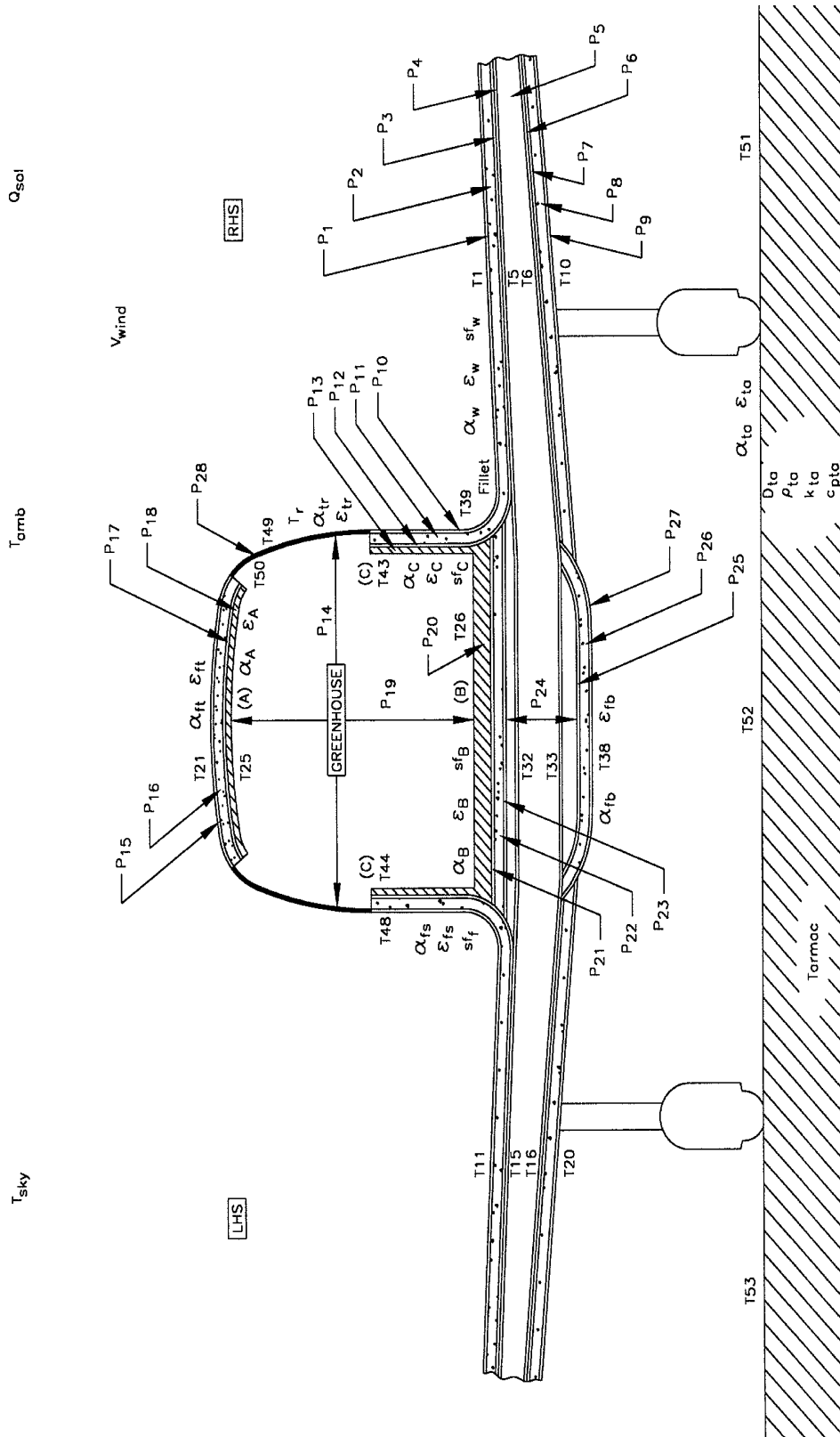


FIGURE 1-3. INPUT THERMAL PROPERTIES FOR THE THERMOD MODEL

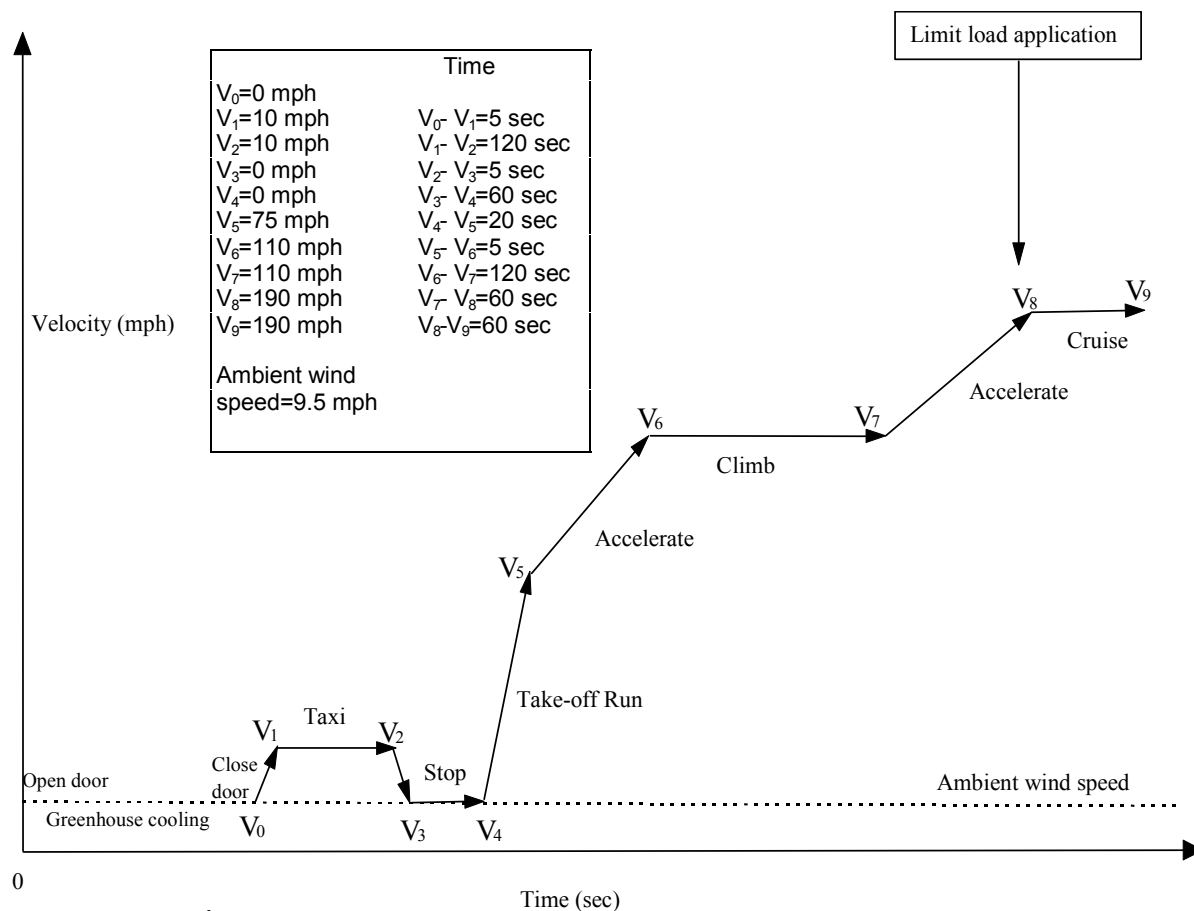


FIGURE 1-4. A TYPICAL FLIGHT PROFILE DURING TRANSIENT COOLING

Fuselage is modeled as a box and treated as an enclosure in modeling the greenhouse effect. The fuselage roof is treated as a surface with an equivalent transmissivity, allowing solar radiation to enter the cabin. The insulatory materials (A), (B), and (C) are simplified representations of cabin roof interior, cabin floor, and cabin side interiors.

A total of 28 distinct layers are considered. Each layer, labeled P_1 to P_{28} in figure 1-3, is assigned four properties, i.e., thickness, density, thermal conductivity, and specific heat capacity. The tarmac is regarded as a special layer and also characterized by these four properties, as indicated in figure 1-3. The eight exposed surfaces are assigned absorptivity (α) or emissivity (ϵ) properties, depending on their surface texture and color. The wing, fuselage, and the interior (B) and (C) surfaces are assigned scatter factors (sf). The aircraft is subjected to climatic factors that include solar radiation (Q_{sol}), wind (V_{wind}), ambient temperature (T_{amb}), and sky temperature (T_{sky}).

Cooling is assumed to occur as soon as the door is opened. The greenhouse cooling duration is a function of the soaked steady-state ambient temperature of the cabin and of the tolerable ambient temperature desired before closing the door. This tolerable temperature, arbitrarily selected, is

input into THERMOD as *ambcabt*. The MOL temperature is determined at the point when limit load is applied. This moment in time is located at 395 seconds from point V_0 , as shown in figure 1-4. The ambient wind speed is added to the flight speed to give the total relative wind speed. THERMOD uses this relative wind speed at selected times to determine convective coefficient, which is a required input in the cooling process.

1.3 PROJECT OBJECTIVES.

The objectives of the project were to

- Debug, identify, and correct the existing errors in THERMOD.
- develop and validate an alternate transient solution process based on the implicit backward finite difference method (IBFDM). The existing transient model in THERMOD is based on the explicit forward finite difference method (EFFDM), which is not unconditionally stable. The solution to some problems may not converge. An IBFDM, however, guarantees convergence by virtue of it being unconditionally stable across all time and spatial domain.
- undertake numerical validation of THERMOD, with respect to the finite element methods (FEM). The three modes of heat transfer, i.e., conduction, convection, and radiation, were investigated. In addition, both the steady-state analysis and the transient analysis were considered.
- solve a sample aircraft problem. The problem entailed developing input data sets and determining the MOL temperature of a typical small aircraft.

2. DEBUGGING AND PROGRAM CORRECTIONS.

Several bugs were detected in the older THERMOD program [1]. This section discusses these bugs and presents remedial actions that were taken to correct them.

2.1 BUG 1.

BUG 1 caused occasional program nonconvergence. The problem was traced to subroutine updradio. This subroutine updates radiosity at each time step of the transient analysis.

Radiosity is defined as the total energy leaving a surface, which includes both infrared reflection and emission. There are a total of 14 radiosity functions in THERMOD. Two of these radiosity functions were found to be in error. The infrared emission terms in these functions were inadvertently left out. The subroutine is called from either subroutine trnsien1 or trnsien2, depending on whether the EFFDM or IBFDM is selected. Subroutine updradio in turn calls subroutines ludcmp and lubksb.

The two radiosities affected were J_{12} and J_{14} . In both functions, the infrared emission effect of the tarmac were inadvertently left out, which are shown below:

- function J_{12} : $\text{sig} * t(51) ** 4 / ((1 - \text{emista}) / \text{emista})$
- function J_{14} : $\text{sig} * t(53) ** 4 / ((1 - \text{emista}) / \text{emista})$

These errors have been fixed by including the terms in the appropriate locations. Simulation runs of sample problems were found to converge.

2.2 BUG 2.

BUG 2 occasionally produced negative temperatures in some of the 53 temperatures that defined THERMOD, which is not necessarily an error. In the solution process, the nonlinear solution routine determines the temperature values that satisfy user input tolerances for temperatures or thermal functions. As long as one of the tolerances are satisfied, the solution is said to have converged, regardless of whether there are any negative temperatures found as part of the solution. Mathematically, the solution is correct.

The variable-based convergence criterion must satisfy the following condition:

$$\frac{\sum_{j=1}^{67} \text{abs}(X_{j_{i+1}} - X_{j_i})}{\sum_{j=1}^{67} X_{j_{i+1}}} \leq \text{tolx} \quad (2-1)$$

where:

$X_{j_{i+1}}$ = The current value of the unknown variable X_j
 X_{j_i} = The previous value of the unknown variable X_j

The function-based convergence criterion must satisfy the following condition:

$$\frac{\sum_{j=1}^{67} \text{abs}(fn_{j_{i+1}} - fn_{j_i})}{\sum_{j=1}^{67} fn_{j_{i+1}}} \leq \text{tol}f \quad (2-2)$$

where:

$fn_{j_{i+1}}$ = The current value of the function fn_j
 fn_{j_i} = The previous value of the function fn_j

In the older version of THERMOD, when either of these two criteria were satisfied, the solution was completed. Because of this approach, both the tolerances were not satisfied at the same time. In the current version of THERMOD, both criteria must be satisfied simultaneously before the program exits. This is enforced through the statement

“if(errf.le.tolf.and.errx.le.tolx) goto 20”

This dual enforcement removed the problem of negative temperatures for sample problems considered.

2.3 BUG 3.

BUG 3 prevented solution convergence for certain laminate thickness or when too small a time step was used. This problem occurs only in the transient phase of THERMOD. This problem was caused by the EFFDM. This method is not unconditionally stable and is the only method found in the original THERMOD program. An alternate finite difference method IBFDM, was developed to solve this problem. Simulation studies demonstrated that IBFDM has effectively eliminated this problem.

The IBFDM is unconditionally stable over all spatial and time domain [6]. This stability is attributed to the fact that, unlike EFFDM, IBFDM considers the transient problem as a system of equations to be solved in a simultaneous manner. This imparts stability to the solution process and allows the use of less refined spatial discretization and larger time steps. The advantages of IBFDM over EFFDM come, however, at the expense of more complex programming and bookkeeping procedures. In THERMOD, for example, the development of the IBFDM necessitated the rearrangement of the transient equations, their assembly into a set of simultaneous equations, their nonlinear solution using the Newton-Raphson iteration technique [5], and the retrieval of the solved transient temperatures. These solution steps require a substantial increase in computing time. However, this increase in computing time may be offset by the use of less refined spatial discretization and larger time steps, exploiting the inherent stability of IBFDM.

The following section discusses the development of the algorithm for IBFDM followed by an example problem substantiating IBFDM with respect to the existing EFFDM. In addition, an example problem will be presented that will illustrate the inherent stability of IBFDM over EFFDM.

2.4 ALGORITHM DEVELOPMENT FOR IBFDM.

IBFDM in THERMOD can be conceptually viewed as a procedure that recognizes the fact that the temperature of a point in space at a certain instance in time is dependent on the temperatures of all other points that are directly associated with that point. The temperature of each of these other points in turn has its own dependence on temperatures and other points associated with it, and so on. One then notices a chain-like link being formed among all transient temperatures of a system. A unique solution for this link exists for every time step, and only a system of simultaneous equations will provide this solution. THERMOD sets up these equations, assembles them, solves them, and retrieves the transient temperatures for each time step requested.

The governing equation of the transient heat transfer process is based on the formula

$$\dot{E}_{in} - \dot{E}_{out} = \Delta \dot{E} \quad (2-3)$$

Where $\dot{E}_{in}$ = rate of incoming energy; $\dot{E}_{out}$ = rate of outgoing energy; and $\Delta \dot{E}$ = rate of energy gain or energy lost. A positive $\Delta \dot{E}$ indicates energy is gained, while a negative $\Delta \dot{E}$ indicates energy is lost. A zero $\Delta \dot{E}$ implies a steady-state system where there is no gain or loss of energy in the system. Note that equation 2-3 does not include the effect of energy generation, which is certainly absent in the case of an aircraft parked in the open.

The implementation of the IBFDM transient programming rests on setting up equation 2-3 for each one of the control volumes defined in the system and solving the resulting set of simultaneous equations.

The procedure can be demonstrated by developing the equation for control volume 3 shown in figure 2-1. This control volume may be viewed to be within the wing of an aircraft where the surface effects are absent. The only mode of heat transfer in this region is conduction. Only through thickness, one-dimensional conduction is assumed. Along-wing span and along-wing width conduction were neglected in THERMOD to produce conservative temperatures.

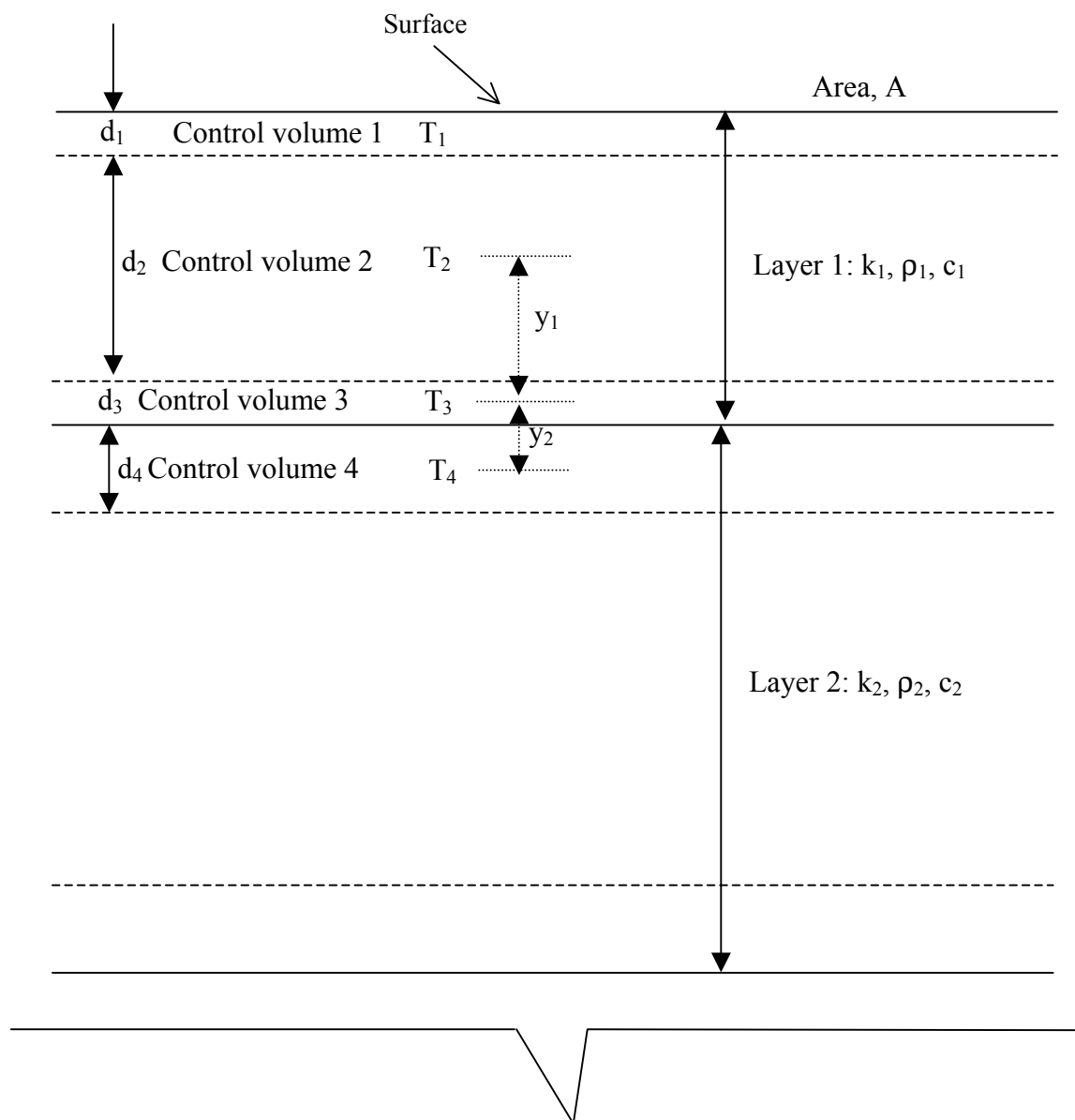


FIGURE 2-1. CONTROL VOLUMES FOR DEVELOPING THERMAL EQUATIONS

In developing the equations, it is assumed that the reader has a basic understanding of the principles of heat transfer. For an overview of the mathematical derivation of the three modes of heat transfer, i.e., conduction, convection, and radiation, the reader is referred to reference 1.

From figure 2-1, for control volume 3 with cross-sectional area A

$$\dot{E}_{in} = k_a \frac{T_2^{i+1} - T_3^{i+1}}{y_1} A \quad \text{where } y_1 = \frac{1}{2} (d_2 + d_3)$$

$$k_a = \frac{\left(\frac{d_2}{2} + \frac{d_3}{2}\right)}{\frac{\frac{d_2}{2}}{k_1} + \frac{\frac{d_3}{2}}{k_1}}$$

$$\dot{E}_{out} = k_b \frac{T_3^{i+1} - T_4^{i+1}}{y_2} A \quad \text{where } y_2 = \frac{1}{2} (d_3 + d_4)$$

$$k_b = \frac{\left(\frac{d_3}{2} + \frac{d_4}{2}\right)}{\frac{\frac{d_3}{2}}{k_1} + \frac{\frac{d_4}{2}}{k_2}}$$

$$\Delta \dot{E} = m c_1 \frac{T_3^{i+1} - T_3^i}{\Delta t}$$

where: m = mass of the control volume = $\rho_1 A d_3$
 ρ_1 = density of the control volume
 c_1 = specific heat capacity of the control volume
 Δt = time step

In the above equations, T^{i+1} represents new temperature, which is an unknown quantity, and T^i represents the previous temperature, which is a known quantity.

Substituting the above expressions for $\dot{E}_{in}$, $\dot{E}_{out}$, and $\Delta \dot{E}$ into equation 2-3 gives

$$k_a \frac{T_2^{i+1} - T_3^{i+1}}{y_1} A - k_b \frac{T_3^{i+1} - T_4^{i+1}}{y_2} A = \rho_1 A d_3 c_1 \frac{T_3^{i+1} - T_3^i}{\Delta t}$$

Eliminating A, and rearranging, one obtains:

$$a_1 T_2^{i+1} + a_2 T_3^{i+1} + a_3 T_4^{i+1} = a_4 T_3^i \quad (2-4)$$

where:

$$a_1 = \frac{k_a}{y_1}$$

$$a_2 = \left(-\frac{k_a}{y_1} - \frac{k_b}{y_2} - \frac{\rho_1 d_3 c_1}{\Delta t} \right)$$

$$a_3 = \frac{k_b}{y_2}$$

$$a_4 = -\frac{\rho_1 d_3 c_1}{\Delta t}$$

Note that in equation 2-4, a_1 , a_2 , a_3 , a_4 , and T^i are known quantities. The unknowns are T_2^{i+1} , T_3^{i+1} , and T_4^{i+1} . It is observed that equation 2-4 is implicit, in that there are more than one unknown. To solve this equation, one needs to create another equation representing the control volume adjacent to the current one. This equation would contain other unknown temperatures. Thus, one needs to develop a series of equations representing all control volumes that were used to define the thermal system, before a unique solution for the set of new temperatures, T^{i+1} could be found.

The difference between IBFDM and EFFDM can be observed by studying the derivation of equation 2-4. Had $\dot{E}_{in}$ and $\dot{E}_{out}$ been defined in terms of the previous temperatures, the only unknown in equation 2-4 would have been T_3^{i+1} , and this would have rendered the equation explicit. Because $\dot{E}_{in}$ and $\dot{E}_{out}$ were expressed in terms of the new temperatures, which are unknown quantities, equation 2-4 was rendered implicit.

In THERMOD, a total of 39 control volumes were created, resulting in 117 unknown temperatures, T^{i+1} . Note that each control volume was designed to have three temperatures, as illustrated in the derivation of the equations above. Twenty-seven of the control volumes were within the structure where the only mode of heat transfer was by means of conduction, as was illustrated in the above sample equations. The remaining twelve control volumes represented surface elements where convection and radiation effects, in addition to conduction effects, were considered. The 117 equations were assembled and subsequently solved using the nonlinear Newton-Raphson iteration procedure. A description of this procedure is found in references 3 and 4. Note that the nonlinear solution procedure is required for two reasons. First, infrared radiation, a capability modeled in THERMOD, follows the Stefan-Boltzman Law, which is based on a nonlinear formulation, where radiation $R = \sigma T^4$. In this formulation, the temperature, T , is an unknown quantity. Second, the convection coefficient within the cabin was modeled to be dependent on temperatures, which again, are an unknown quantity.

The transient solution is initiated by treating the temperatures obtained from the steady-state analysis as the initial set of previous temperatures. The solved new temperatures are then treated as the previous temperatures in the next time step. The analysis is repeated until a solution satisfying a required set of tolerance in temperatures and functions is obtained. Once solved, the 117 temperatures are retrieved, from which only the 53 temperatures defining THERMOD are reported. (See figures 1-1 and 1-2 to locate these temperatures.)

As indicated, THERMOD is a thermal system with 67 unknowns, where 53 of the unknowns are temperatures and 14 are radiosity. Radiosity is defined as the radiative flux that leaves a surface, and it includes both the reflected irradiation as well as emission [6]. Because THERMOD depends on temperatures, it follows that radiosity will change with time in a transient environment. This change in radiosity is implemented in THERMOD by updating its value using the new temperatures determined at the end of each time step.

The various subroutines, their definitions, and the program flow chart addressing IBFDM are contained in appendix A.

The development of IBFDM required extensive rearrangement of the transient equations, substantial bookkeeping, a nonlinear solution of the resulting simultaneous equations, and data retrieval. These extra efforts were expected to increase the computing resources. This was indeed the case. The amount of computing time spent was vastly different for the previous two problems. The EFFDM solution process took less than 1 minute, while the IBFDM took about 90 minutes.

As was previously noted, the primary purpose of IBFDM was to overcome probable instability problems that may arise using EFFDM. The substantial increase in the computing effort of IBFDM, however, seems to be a worthwhile price to pay, given that a potential stability problem was solved and satisfactory results are expected. It is also noted that because IBFDM is inherently stable regardless of the time step duration, the computing effort may be substantially reduced by increasing the time step until accuracy of the solution is not affected.

3. NUMERICAL VALIDATION OF THERMOD.

This section presents the numerical validation of THERMOD with respect to FEMs. A total of eight problems were formulated and solved using the THERMOD program and then the FEM programs. The results from both methods of analysis were then compared.

It was beyond the scope of this project to develop a complete FEM thermal model of an aircraft. The FEM sample problems were, thus, kept as simple as possible and were yet able to capture the three fundamental modes of heat transfer mechanisms, i.e., conduction, convection, and radiation. The problems, where possible, addressed each transport mechanism individually first, then progressed into more complex thermal systems involving a combination of these mechanisms.

In developing sample input files for THERMOD, all 15 sets of data defining the entire thermal problem of an aircraft must be generated and used in all sample problems. This is because, unlike the FEM methods, THERMOD was not developed as a general-purpose thermal program. It was designed to solve a very specific thermal problem associated with an aircraft subjected to the thermal environment. While the solution process in itself is rather complex, the inputs to the program were designed to be simple and straightforward. Because of this constraint, aircraft problems in its entirety will be solved in THERMOD during this validation exercise. The FEM methods, however, are not subject to this constraint.

Of the eight validation problems solved, six involved steady-state analysis. The other two involved both steady-state and transient analyses. The FEM models of the six steady-state problems were analyzed using UAI/NASTRAN software. The FEM models of the two combined steady-state and transient problems were analyzed using MSC/NASTRAN software. The primary reason for using two different FEM packages was that the thermal module of the UAI/NASTRAN package was found to be deficient in the transient analysis. It lacked a time-dependent convective coefficient, which is needed in analyzing transient cooling, where air is forced over the aircraft while maneuvering according to various flight profile information. The MSC/NASTRAN thermal module was found to have this capability.

To have a better understanding of the sample problems presented in this section, it is helpful to be familiar with the THERMOD User's Manual [2], which gives a complete description of input data sets, variable definitions, formats, and other general information. Only information directly relevant to the validation problems will be discussed in detail in this section, while the discussion of other data will be kept to a minimum.

It is assumed that the reader have a working knowledge of the FEM methods, sufficient to decipher NASTRAN input data decks. The relevant FEM information pertaining to NASTRAN can be found in the following documents: UAI/NASTRAN User's Reference Manual [7], UAI/NASTRAN User's Guide [8], MSC/NASTRAN Quick Reference Guide [9], MSC/NASTRAN User's Guide [10], MSC/NASTRAN Command Reference Guide [11], and MSC/NASTRAN Thermal Analysis User's Guide [12].

3.1 SAMPLE PROBLEM 1.

The purpose of sample problem 1 was to illustrate the conduction mechanism of the heat transfer process. Table 3-1 shows the THERMOD input file, input.dat, of sample problem 1. As previously noted, a total of 15 sets of data must always be used to complete a successful run of THERMOD. These sets of data define a thermal problem of a complete aircraft.

TABLE 3-1. THERMOD INPUT FILE (input.dat) OF SAMPLE PROBLEM 1

```
198e-6,0.704
1, 0.020,125,0.300,0.30
2, 0.375,4.4,0.021,0.24
3, 0.020,125,0.300,0.30
4, 0.350,125,0.300,0.30
5, 10.70,0.067,0.016,0.24
6, 0.350,125,0.300,0.30
7, 0.020,125,0.300,0.30
8, 0.375,4.4,0.021,0.24
9, 0.020,125,0.300,0.30
10,0.030,125,0.300,0.30
11,0.375,4.4,0.021,0.24
12,0.020,125,0.300,0.30
13,0.250,10.,0.020,0.24
14,48.00,0.067,0.016,0.24
15,0.020,125,0.300,0.30
16,0.375,4.4,0.021,0.24
17,0.020,125,0.300,0.30
18,0.250,10.,0.020,0.24
19,30.00,0.067,0.016,0.24
20,5.000,10.,0.020,0.24
21,0.020,125,0.300,0.30
22,0.375,4.4,0.021,0.24
23,0.020,125,0.300,0.30
24,2.500,0.067,0.016,0.24
25,0.020,125,0.300,0.30
26,0.375,4.4,0.021,0.24
27,0.020,125,0.300,0.30
28,0.250,130,0.800,0.20
3.5,34.0,2.5,5.5,22.0,11.0,3.0,4.0,30,24,28
0.70,0.9,0.9,2.0,150,0.80,0.22
0.3,0.9,0.3,0.9,0.3,0.9,0.1,0.9
3,100
0.9,0.9,0.9,0.9
0.95,0.47,0.3,0.8,0.95,0.47,0.95,0.47
9
0,10,10,0,0,75,110,110,190,190
5,120,5,60,20,5,120,60,60
```

TABLE 3-1. THERMOD INPUT FILE (input.dat) OF SAMPLE PROBLEM 1 (Continued)

7
60 570.67 459.67 14 330 10 10 0.36
75 573.67 459.67 14 355 10 10 0.31
90 578.67 459.67 14 355 10 12 0.33
75 581.67 459.67 14 330 10 10 0.31
60 582.67 459.67 14 291 10 10 0.36
45 583.67 459.67 14 231 10 10 0.44
30 582.67 459.67 14 160 10 10 0.57
2,100,180,1
10,1.0,1.0
10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,
10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,
10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,
10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,
10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,
10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,
10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,

For comparison purposes with the FEM method, the focus was on the right wing of the aircraft. (The right wing is indicated by RHS in figure 1-1.) As illustrated in the input.dat file of table 3-1, the right wing was modeled as follows:

- Layer 1: 0.02-in.-thick top outer fiber glass skin
- Layer 2: 0.375-in.-thick top foam core
- Layer 3: 0.02-in.-thick top inner fiber glass skin
- Layer 4: 0.35-in.-thick top fiber glass spar cap
- Layer 5: 10.70-in.-deep air space between the top and bottom spar caps
- Layer 6: 0.35-in.-thick bottom fiber glass spar cap
- Layer 7: 0.02-in.-thick bottom inner fiber glass skin
- Layer 8: 0.375-in.-thick bottom foam core
- Layer 9: 0.02-in.-thick bottom outer fiber glass skin

The summary.dat file of table 3-2 summarizes the results of the THERMOD analysis. As observed in table 3-2, a total of seven time periods were involved in the THERMOD analysis. Steady-state temperature results obtained for time period 3 was chosen for comparison purposes with the FEM method. A total of 10 temperature locations were identified on the right wing, as illustrated in figure 1-1. These 10 temperatures from the top to the bottom are T1, T2.....T10. Their values were extracted from table 3-2 and summarized in table 3-3. The results indicated that the top surface of the wing, because of its exposure to the sun, was hotter than the lower surface, which was more sheltered. There was a significant difference in temperature between the upper cap (T5) and the lower cap (T6). This was attributed to the 10.70 in. of air (which has low thermal conductivity) that separates them.

TABLE 3-2. THERMOD OUTPUT FILE (summary.dat) OF SAMPLE PROBLEM 1

```

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
THERMOD:  A THERMAL PROGRAM FOR AIRCRAFTS
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
  
```

SUMMARY OF STEADY-STATE AND TRANSIENT ANALYSIS

(Temperatures are shown for all time periods)

TEMPERATURES AT THE END OF STEADY STATE ANALYSIS

	TEMPERATURES IN DEGREES FAHRENHEIT						
	1	2	3	4	5	6	7
1	124.0	132.5	136.1	134.0	126.4	115.2	102.8
2	124.0	132.5	136.1	134.0	126.4	115.2	102.8
3	123.7	132.2	135.8	133.7	126.2	115.3	103.0
4	123.7	132.2	135.8	133.7	126.2	115.3	103.0
5	123.7	132.2	135.8	133.7	126.2	115.3	103.0
6	114.4	120.2	124.7	124.6	121.1	116.3	110.5
7	114.3	120.2	124.7	124.6	121.1	116.4	110.5
8	114.3	120.2	124.7	124.6	121.1	116.4	110.5
9	114.1	119.9	124.4	124.3	121.0	116.4	110.7
10	114.1	119.9	124.4	124.3	121.0	116.4	110.7
11	122.1	131.4	135.9	132.9	124.8	113.5	101.4
12	122.1	131.4	135.9	132.9	124.8	113.5	101.4
13	121.9	131.1	135.6	132.7	124.7	113.6	101.6
14	121.9	131.1	135.6	132.7	124.7	113.6	101.6
15	121.9	131.1	135.6	132.7	124.7	113.6	101.6
16	114.3	120.2	124.7	124.6	121.1	116.3	110.5
17	114.3	120.2	124.7	124.6	121.1	116.3	110.5
18	114.3	120.2	124.7	124.6	121.1	116.3	110.5
19	114.1	119.9	124.4	124.3	121.0	116.4	110.7
20	114.1	119.9	124.4	124.3	121.0	116.4	110.7
21	205.8	225.9	231.9	222.2	200.7	169.4	134.2
22	205.8	225.8	231.8	222.1	200.7	169.5	134.2
23	207.4	216.5	217.2	213.6	202.4	180.2	143.1
24	207.4	216.4	217.2	213.6	202.4	180.2	143.1
25	208.4	209.9	206.9	207.6	203.6	187.7	149.4
26	198.9	198.2	195.5	196.8	195.4	183.4	146.2
27	150.5	153.0	154.4	155.5	153.6	146.5	127.6
28	150.4	153.0	154.4	155.4	153.6	146.5	127.6

TABLE 3-2. THERMOD OUTPUT FILE (summary.dat) OF SAMPLE
 PROBLEM 1 (Continued)

29	147.0	149.8	151.5	152.5	150.6	143.9	126.3
30	147.0	149.8	151.5	152.5	150.6	143.9	126.3
31	131.8	135.6	138.7	139.6	137.5	132.4	120.5
32	131.8	135.6	138.7	139.6	137.5	132.4	120.5
33	131.8	135.6	138.7	139.6	137.5	132.4	120.5
34	131.8	135.6	138.7	139.6	137.5	132.4	120.5
35	116.7	121.5	125.8	126.6	124.5	120.8	114.7
36	116.7	121.5	125.8	126.6	124.4	120.8	114.7
37	113.2	118.3	122.9	123.7	121.5	118.2	113.3
38	113.2	118.3	122.9	123.7	121.4	118.2	113.3
39	199.5	179.7	148.8	180.0	196.6	194.7	173.9
40	199.5	179.7	148.9	180.1	196.6	194.6	173.9
41	197.6	186.6	168.7	186.2	194.5	188.4	162.3
42	197.6	186.6	168.8	186.2	194.5	188.4	162.3
43	196.2	191.4	182.7	190.5	193.1	184.1	154.2
44	191.8	188.5	182.6	187.8	189.2	180.4	151.1
45	169.9	170.8	168.7	171.5	170.2	161.4	138.9
46	169.8	170.7	168.7	171.5	170.0	161.3	138.8
47	138.6	145.3	148.8	148.2	142.9	134.1	121.3
48	138.4	145.2	148.7	148.0	142.7	133.9	121.2
49	145.6	148.6	149.7	150.4	147.7	139.7	123.0
50	147.6	150.5	151.3	152.1	149.5	141.3	123.9
51	113.6	120.8	125.1	123.7	118.1	110.0	100.6
52	108.2	115.1	119.2	118.0	112.7	105.1	96.2
53	113.6	120.8	125.1	123.7	118.1	110.0	100.6
Maximum Structural Temperature for Period 1 = 207.37 Occurring at Location 24							
Maximum Structural Temperature for Period 2 = 225.86 Occurring at Location 21							
Maximum Structural Temperature for Period 3 = 231.88 Occurring at Location 21							
Maximum Structural Temperature for Period 4 = 222.15 Occurring at Location 21							
Maximum Structural Temperature for Period 5 = 202.43 Occurring at Location 24							
Maximum Structural Temperature for Period 6 = 194.68 Occurring at Location 39							
Maximum Structural Temperature for Period 7 = 173.92 Occurring at Location 39							
Maximum Structural Temperature Over All 7 Periods =231.88 Occurring at Location 21 at Period 3							

TABLE 3-2. THERMOD OUTPUT FILE (summary.dat) OF SAMPLE
 PROBLEM 1 (Continued)

TEMPERATURES AT THE END OF TRANSIENT ANALYSIS							
TEMPERATURES IN DEGREES FAHRENHEIT							
	1	2	3	4	5	6	7
1	112.9	116.8	121.7	123.8	123.4	122.5	119.8
2	113.2	117.3	122.2	124.1	123.5	122.3	119.2
3	122.9	131.1	134.8	133.0	126.0	115.7	104.1
4	123.2	131.5	135.2	133.3	126.1	115.6	103.7
5	123.2	131.6	135.2	133.3	126.0	115.4	103.5
6	114.6	120.5	125.0	124.9	121.3	116.5	110.6
7	114.3	120.1	124.6	124.6	121.3	116.7	111.0
8	114.3	120.0	124.5	124.6	121.3	116.9	111.3
9	112.7	116.6	121.6	123.8	123.6	123.1	120.9
10	112.7	116.4	121.4	123.7	123.7	123.3	121.2
11	112.7	116.8	121.7	123.8	123.3	122.4	119.7
12	113.0	117.2	122.1	124.0	123.3	122.1	119.1
13	121.3	130.1	134.6	132.1	124.6	114.2	102.9
14	121.5	130.5	135.0	132.3	124.6	114.0	102.4
15	121.5	130.6	135.0	132.3	124.6	113.8	102.2
16	114.5	120.4	124.9	124.8	121.2	116.4	110.5
17	114.3	120.1	124.6	124.6	121.2	116.6	111.0
18	114.2	120.0	124.5	124.6	121.3	116.8	111.2
19	112.7	116.6	121.6	123.8	123.6	123.1	120.9
20	112.7	116.4	121.4	123.7	123.7	123.3	121.2
21	125.4	131.3	136.7	137.5	134.8	130.9	124.7
22	127.4	133.3	138.7	139.3	136.4	132.3	125.2
23	181.2	187.5	192.2	187.4	180.5	171.4	140.3
24	183.1	189.2	193.8	188.9	182.0	172.9	140.9
25	200.7	202.6	205.1	200.1	195.7	189.2	147.8
26	185.7	185.5	187.8	185.4	184.4	181.0	145.5
27	150.7	153.2	154.6	155.7	153.8	146.7	127.7
28	149.5	152.1	153.7	154.7	152.8	145.8	127.3
29	146.5	149.3	151.2	152.1	150.2	143.7	126.3
30	145.7	148.6	150.4	151.4	149.5	143.1	126.0
31	131.5	135.1	138.2	139.3	137.7	133.2	121.9
32	131.5	135.1	138.2	139.3	137.7	133.2	121.9
33	131.5	135.1	138.2	139.3	137.7	133.2	121.9
34	131.5	135.1	138.2	139.3	137.7	133.2	121.9
35	117.2	121.6	125.9	127.3	125.8	123.2	117.7
36	116.4	120.7	125.2	126.5	125.2	122.7	117.5
37	112.1	115.7	120.6	123.1	123.4	123.4	121.5
38	112.0	115.5	120.5	123.0	123.3	123.4	121.6
39	121.2	122.3	123.8	129.2	131.3	131.5	127.8
40	123.2	124.0	125.1	130.7	133.0	133.1	128.7

TABLE 3-2. THERMOD OUTPUT FILE (summary.dat) OF SAMPLE
 PROBLEM 1 (Continued)

41	172.1	166.9	160.2	168.7	173.2	172.1	150.2
42	173.6	168.3	161.6	170.0	174.5	173.3	150.8
43	183.1	179.5	176.6	179.9	182.2	181.5	152.3
44	179.2	177.0	176.6	177.6	178.7	178.0	149.3
45	162.3	161.7	161.5	163.9	164.5	162.3	141.2
46	160.8	160.3	160.1	162.6	163.2	161.0	140.6
47	121.5	123.0	125.1	129.8	131.5	131.4	127.2
48	120.0	121.6	123.8	128.6	130.3	130.3	126.7
49	123.6	126.8	131.0	132.5	132.0	130.4	123.1
50	127.8	130.8	134.8	136.0	135.3	133.2	123.9
51	169.1	187.2	193.0	184.0	164.8	137.5	109.0
52	157.9	174.6	180.0	172.0	154.5	129.7	103.8
53	169.1	187.2	193.0	184.0	164.8	137.5	109.0
Maximum Structural Temperature for Period 1 = 183.07 Occurring at Location 24							
Maximum Structural Temperature for Period 2 = 189.17 Occurring at Location 24							
Maximum Structural Temperature for Period 3 = 193.83 Occurring at Location 24							
Maximum Structural Temperature for Period 4 = 188.86 Occurring at Location 24							
Maximum Structural Temperature for Period 5 = 181.98 Occurring at Location 24							
Maximum Structural Temperature for Period 6 = 173.32 Occurring at Location 42							
Maximum Structural Temperature for Period 7 = 150.75 Occurring at Location 42							
Maximum Structural Temperature Over All 7 Periods =193.83 Occurring at Location 24 at Period 3							

TABLE 3-3. THERMOD-SIMULATED TEMPERATURES (°F) OF THE RIGHT WING OF
 SAMPLE PROBLEM 1

T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
136.1	136.1	135.8	135.8	135.8	124.7	124.7	124.7	124.4	124.4

An FEM model of the wing was developed based on the information used in the THERMOD model. One-half of the wing was modeled as a 17-ft-long structure with layer properties as indicated above. A total of 153 CQUAD4 elements, each 0.001 ft thick, was employed, resulting in a total of 324 grid points (nodes). Material properties, representing conductivity, were assigned using MAT4 cards. PSHELL cards were used to assign the thickness and material properties for each of the element.

Boundary conditions were imposed along the outer nodes of the model. Recall that the objective of this sample problem was to validate the conductive aspect of THERMOD. The FEM boundary conditions were then set such that all top surface nodes assumed the value of T1 (136.1°F), and all bottom surface nodes assumed the value of T10 (124.4°F), whose values were obtained from the earlier THERMOD analysis. The outer left and right most edges of the wing were assigned boundary temperatures of 124.4°F, thus completely encapsulating the FEM model. The temperature profile across the wing depth is now entirely due to conduction. Table 3-4 illustrates the FEM input file.

TABLE 3-4. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE PROBLEM 1

```

ID D:\Natha, MSC/N
APP HEAT
SOL 1
TIME 10000
CEND
  TITLE = CONDUCTION IN RECTANGULAR ADIABATIC PLATE (t=0.001ft)
  SUBTITLE = 10.7 in gap filled by air; k=0.016BTU/hr/ft/degF
  SPC = 1
  THERMAL= ALL
  FLUX = ALL
  SPCF = ALL
BEGIN BULK
$ *****
$   Written by : MSC/NASTRAN for Windows
$   Version   : 6.00
$   Translator : UAI/NASTRAN
$   From Model : D:\Thermod_Validation\FEM\conduction_only.MOD
$   Date      : Mon Mar 06 11:27:09 2000
$ *****
$
PARAM, K6ROT, 100.
PARAM, MAXRATIO, 1.E+8
CORD2C      1      0      0.      0.      0.      0.      0.      1.+MSC/NC1
+MSC/NC1    1.      0.      1.
CORD2S      2      0      0.      0.      0.      0.      0.      1.+MSC/NC2
+MSC/NC2    1.      0.      1.
$ MSC/NASTRAN for Windows Constraint Set 1 : steady-state_bound-temp
SPC          1      1      1 124.4
SPC          1     20      1 124.4
SPC          1     21      1 124.4
SPC          1     22      1 124.4
SPC          1     23      1 124.4
SPC          1     24      1 124.4
SPC          1     25      1 124.4
SPC          1     26      1 124.4
SPC          1     27      1 124.4
SPC          1     28      1 124.4
SPC          1     29      1 124.4
SPC          1     30      1 124.4
SPC          1     31      1 124.4
SPC          1     32      1 124.4
SPC          1     33      1 124.4
SPC          1     34      1 124.4
SPC          1     35      1 124.4
SPC          1     36      1 124.4
SPC          1     37      1 124.4

```

TABLE 3-4. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 1 (Continued)

SPC	1	55	1	124.4
SPC	1	56	1	124.4
SPC	1	73	1	124.4
SPC	1	109	1	124.4
SPC	1	128	1	124.4
SPC	1	145	1	124.4
SPC	1	146	1	124.4
SPC	1	164	1	124.4
SPC	1	181	1	124.4
SPC	1	182	1	124.4
SPC	1	199	1	124.4
SPC	1	200	1	124.4
SPC	1	201	1	136.1
SPC	1	202	1	136.1
SPC	1	203	1	136.1
SPC	1	204	1	136.1
SPC	1	205	1	136.1
SPC	1	206	1	136.1
SPC	1	207	1	136.1
SPC	1	208	1	136.1
SPC	1	209	1	136.1
SPC	1	210	1	136.1
SPC	1	211	1	136.1
SPC	1	212	1	136.1
SPC	1	213	1	136.1
SPC	1	214	1	136.1
SPC	1	215	1	136.1
SPC	1	216	1	136.1
SPC	1	235	1	124.4
SPC	1	253	1	124.4
SPC	1	254	1	124.4
SPC	1	307	1	124.4
SPC	1	308	1	124.4
\$ MSC/NASTRAN for Windows Property 1 : 0.02 in glass_outer_layer				
PSHELL	1	1	0.001	1 0.
\$ MSC/NASTRAN for Windows Property 2 : 3/8 in foam				
PSHELL	2	2	0.001	2 0.
\$ MSC/NASTRAN for Windows Property 3 : 0.02 in inner glass				
PSHELL	3	1	0.001	1 0.
\$ MSC/NASTRAN for Windows Property 4 : 0.350 in cap				
PSHELL	4	1	0.001	1 0.
\$ MSC/NASTRAN for Windows Property 5 : 10.70 in air space				
PSHELL	5	3	0.001	3 0.
\$ MSC/NASTRAN for Windows Property 6 : 0.350 in cap				
PSHELL	6	1	0.001	1 0.
\$ MSC/NASTRAN for Windows Property 7 : 0.02 in inner glass				
PSHELL	7	1	0.001	1 0.
\$ MSC/NASTRAN for Windows Property 8 : 3/8 in foam				
PSHELL	8	2	0.001	2 0.
\$ MSC/NASTRAN for Windows Property 9 : 0.02 in glass_outer_layer				
PSHELL	9	1	0.001	1 0.
\$ MSC/NASTRAN for Windows Material 1 : glass				
MAT4	1	0.3	1.	
\$ MSC/NASTRAN for Windows Material 2 : foam				
MAT4	2	0.021	1.	
\$ MSC/NASTRAN for Windows Material 3 : air				
MAT4	3	0.016	1.	

TABLE 3-4. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 1 (Continued)

GRID	1	0	0.	0.	0.	0
GRID	3	0	1.1.667E-3	0.	0.	0
GRID	5	0	3.1.667E-3	0.	0.	0
GRID	7	0	5.1.667E-3	0.	0.	0
GRID	9	0	7.1.667E-3	0.	0.	0
GRID	10	0	8.1.667E-3	0.	0.	0
GRID	11	0	9.1.667E-3	0.	0.	0
GRID	12	0	10.1.667E-3	0.	0.	0
GRID	14	0	12.1.667E-3	0.	0.	0
GRID	15	0	13.1.667E-3	0.	0.	0
GRID	18	0	16.1.667E-3	0.	0.	0
GRID	20	0	17.	0.	0.	0
GRID	21	0	16.	0.	0.	0
GRID	22	0	15.	0.	0.	0
GRID	23	0	14.	0.	0.	0
GRID	24	0	13.	0.	0.	0
GRID	25	0	12.	0.	0.	0
GRID	26	0	11.	0.	0.	0
GRID	27	0	10.	0.	0.	0
GRID	28	0	9.	0.	0.	0
GRID	29	0	8.	0.	0.	0
GRID	30	0	7.	0.	0.	0
GRID	31	0	6.	0.	0.	0
GRID	32	0	5.	0.	0.	0
GRID	33	0	4.	0.	0.	0
GRID	34	0	3.	0.	0.	0
GRID	35	0	2.	0.	0.	0
GRID	36	0	1.	0.	0.	0
GRID	37	0	0.1.667E-3	0.	0.	0
GRID	41	0	3.0.032917	0.	0.	0
GRID	44	0	6.0.032917	0.	0.	0
GRID	45	0	7.0.032917	0.	0.	0
GRID	47	0	9.0.032917	0.	0.	0
GRID	48	0	10.0.032917	0.	0.	0
GRID	50	0	12.0.032917	0.	0.	0
GRID	51	0	13.0.032917	0.	0.	0
GRID	52	0	14.0.032917	0.	0.	0
GRID	53	0	15.0.032917	0.	0.	0
GRID	54	0	16.0.032917	0.	0.	0
GRID	55	0	17.0.032917	0.	0.	0
GRID	56	0	17.1.667E-3	0.	0.	0
GRID	58	0	15.1.667E-3	0.	0.	0
GRID	59	0	14.1.667E-3	0.	0.	0
GRID	62	0	11.1.667E-3	0.	0.	0
GRID	67	0	6.1.667E-3	0.	0.	0
GRID	69	0	4.1.667E-3	0.	0.	0
GRID	71	0	2.1.667E-3	0.	0.	0
GRID	73	0	0.0.032917	0.	0.	0
GRID	75	0	1.0.034583	0.	0.	0
GRID	76	0	2.0.034583	0.	0.	0
GRID	77	0	3.0.034583	0.	0.	0
GRID	78	0	4.0.034583	0.	0.	0
GRID	79	0	5.0.034583	0.	0.	0
GRID	80	0	6.0.034583	0.	0.	0
GRID	83	0	9.0.034583	0.	0.	0
GRID	85	0	11.0.034583	0.	0.	0
GRID	87	0	13.0.034583	0.	0.	0

TABLE 3-4. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 1 (Continued)

GRID	88	0	14.0.034583	0.	0
GRID	98	0	11.0.032917	0.	0
GRID	101	0	8.0.032917	0.	0
GRID	104	0	5.0.032917	0.	0
GRID	105	0	4.0.032917	0.	0
GRID	107	0	2.0.032917	0.	0
GRID	108	0	1.0.032917	0.	0
GRID	109	0	0.0.034583	0.	0
GRID	112	0	2. 0.06375	0.	0
GRID	115	0	5. 0.06375	0.	0
GRID	116	0	6. 0.06375	0.	0
GRID	118	0	8. 0.06375	0.	0
GRID	120	0	10. 0.06375	0.	0
GRID	121	0	11. 0.06375	0.	0
GRID	122	0	12. 0.06375	0.	0
GRID	123	0	13. 0.06375	0.	0
GRID	124	0	14. 0.06375	0.	0
GRID	125	0	15. 0.06375	0.	0
GRID	128	0	17.0.034583	0.	0
GRID	129	0	16.0.034583	0.	0
GRID	130	0	15.0.034583	0.	0
GRID	133	0	12.0.034583	0.	0
GRID	135	0	10.0.034583	0.	0
GRID	137	0	8.0.034583	0.	0
GRID	138	0	7.0.034583	0.	0
GRID	145	0	0. 0.06375	0.	0
GRID	146	0	0. 0.95542	0.	0
GRID	147	0	1. 0.95542	0.	0
GRID	150	0	4. 0.95542	0.	0
GRID	152	0	6. 0.95542	0.	0
GRID	160	0	14. 0.95542	0.	0
GRID	161	0	15. 0.95542	0.	0
GRID	164	0	17. 0.06375	0.	0
GRID	165	0	16. 0.06375	0.	0
GRID	172	0	9. 0.06375	0.	0
GRID	174	0	7. 0.06375	0.	0
GRID	177	0	4. 0.06375	0.	0
GRID	178	0	3. 0.06375	0.	0
GRID	180	0	1. 0.06375	0.	0
GRID	181	0	0. 1.01917	0.	0
GRID	182	0	0. 1.0175	0.	0
GRID	183	0	1. 1.0175	0.	0
GRID	184	0	2. 1.0175	0.	0
GRID	188	0	6. 1.0175	0.	0
GRID	193	0	11. 1.0175	0.	0
GRID	195	0	13. 1.0175	0.	0
GRID	196	0	14. 1.0175	0.	0
GRID	197	0	15. 1.0175	0.	0
GRID	198	0	16. 1.0175	0.	0
GRID	199	0	17. 1.0175	0.	0
GRID	200	0	17. 1.01917	0.	0
GRID	201	0	16. 1.01917	0.	0
GRID	202	0	15. 1.01917	0.	0
GRID	203	0	14. 1.01917	0.	0
GRID	204	0	13. 1.01917	0.	0
GRID	205	0	12. 1.01917	0.	0
GRID	206	0	11. 1.01917	0.	0

TABLE 3-4. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 1 (Continued)

GRID	207	0	10.	1.01917	0.	0
GRID	208	0	9.	1.01917	0.	0
GRID	209	0	8.	1.01917	0.	0
GRID	210	0	7.	1.01917	0.	0
GRID	211	0	6.	1.01917	0.	0
GRID	212	0	5.	1.01917	0.	0
GRID	213	0	4.	1.01917	0.	0
GRID	214	0	3.	1.01917	0.	0
GRID	215	0	2.	1.01917	0.	0
GRID	216	0	1.	1.01917	0.	0
GRID	219	0	1.	0.98625	0.	0
GRID	220	0	2.	0.98625	0.	0
GRID	221	0	3.	0.98625	0.	0
GRID	222	0	4.	0.98625	0.	0
GRID	223	0	5.	0.98625	0.	0
GRID	226	0	8.	0.98625	0.	0
GRID	227	0	9.	0.98625	0.	0
GRID	228	0	10.	0.98625	0.	0
GRID	229	0	11.	0.98625	0.	0
GRID	231	0	13.	0.98625	0.	0
GRID	232	0	14.	0.98625	0.	0
GRID	235	0	17.	0.98625	0.	0
GRID	241	0	12.	1.0175	0.	0
GRID	243	0	10.	1.0175	0.	0
GRID	244	0	9.	1.0175	0.	0
GRID	245	0	8.	1.0175	0.	0
GRID	246	0	7.	1.0175	0.	0
GRID	248	0	5.	1.0175	0.	0
GRID	249	0	4.	1.0175	0.	0
GRID	250	0	3.	1.0175	0.	0
GRID	253	0	0.	0.98625	0.	0
GRID	254	0	0.	0.98458	0.	0
GRID	257	0	3.	0.98458	0.	0
GRID	258	0	4.	0.98458	0.	0
GRID	259	0	5.	0.98458	0.	0
GRID	260	0	6.	0.98458	0.	0
GRID	264	0	10.	0.98458	0.	0
GRID	267	0	13.	0.98458	0.	0
GRID	268	0	14.	0.98458	0.	0
GRID	269	0	15.	0.98458	0.	0
GRID	270	0	16.	0.98458	0.	0
GRID	273	0	16.	0.98625	0.	0
GRID	274	0	15.	0.98625	0.	0
GRID	277	0	12.	0.98625	0.	0
GRID	282	0	7.	0.98625	0.	0
GRID	283	0	6.	0.98625	0.	0
GRID	292	0	2.	0.95542	0.	0
GRID	293	0	3.	0.95542	0.	0
GRID	295	0	5.	0.95542	0.	0
GRID	297	0	7.	0.95542	0.	0
GRID	298	0	8.	0.95542	0.	0
GRID	299	0	9.	0.95542	0.	0
GRID	300	0	10.	0.95542	0.	0
GRID	301	0	11.	0.95542	0.	0
GRID	302	0	12.	0.95542	0.	0
GRID	303	0	13.	0.95542	0.	0
GRID	306	0	16.	0.95542	0.	0

TABLE 3-4. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 1 (Continued)

GRID	307	0	17.	0.95542	0.	0
GRID	308	0	17.	0.98458	0.	0
GRID	313	0	12.	0.98458	0.	0
GRID	314	0	11.	0.98458	0.	0
GRID	316	0	9.	0.98458	0.	0
GRID	317	0	8.	0.98458	0.	0
GRID	318	0	7.	0.98458	0.	0
GRID	323	0	2.	0.98458	0.	0
GRID	324	0	1.	0.98458	0.	0
CQUAD4	1	9	1	37	3	36
CQUAD4	2	9	36	3	71	35
CQUAD4	3	9	35	71	5	34
CQUAD4	4	9	34	5	69	33
CQUAD4	5	9	33	69	7	32
CQUAD4	6	9	32	7	67	31
CQUAD4	7	9	31	67	9	30
CQUAD4	8	9	30	9	10	29
CQUAD4	9	9	29	10	11	28
CQUAD4	10	9	28	11	12	27
CQUAD4	11	9	27	12	62	26
CQUAD4	12	9	26	62	14	25
CQUAD4	13	9	25	14	15	24
CQUAD4	14	9	24	15	59	23
CQUAD4	15	9	23	59	58	22
CQUAD4	16	9	22	58	18	21
CQUAD4	17	9	21	18	56	20
CQUAD4	18	8	37	73	108	3
CQUAD4	19	8	3	108	107	71
CQUAD4	20	8	71	107	41	5
CQUAD4	21	8	5	41	105	69
CQUAD4	22	8	69	105	104	7
CQUAD4	23	8	7	104	44	67
CQUAD4	24	8	67	44	45	9
CQUAD4	25	8	9	45	101	10
CQUAD4	26	8	10	101	47	11
CQUAD4	27	8	11	47	48	12
CQUAD4	28	8	12	48	98	62
CQUAD4	29	8	62	98	50	14
CQUAD4	30	8	14	50	51	15
CQUAD4	31	8	15	51	52	59
CQUAD4	32	8	59	52	53	58
CQUAD4	33	8	58	53	54	18
CQUAD4	34	8	18	54	55	56
CQUAD4	35	7	73	109	75	108
CQUAD4	36	7	108	75	76	107
CQUAD4	37	7	107	76	77	41
CQUAD4	38	7	41	77	78	105
CQUAD4	39	7	105	78	79	104
CQUAD4	40	7	104	79	80	44
CQUAD4	41	7	44	80	138	45
CQUAD4	42	7	45	138	137	101
CQUAD4	43	7	101	137	83	47
CQUAD4	44	7	47	83	135	48
CQUAD4	45	7	48	135	85	98
CQUAD4	46	7	98	85	133	50
CQUAD4	47	7	50	133	87	51
CQUAD4	48	7	51	87	88	52

TABLE 3-4. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 1 (Continued)

CQUAD4	49	7	52	88	130	53
CQUAD4	50	7	53	130	129	54
CQUAD4	51	7	54	129	128	55
CQUAD4	52	6	109	145	180	75
CQUAD4	53	6	75	180	112	76
CQUAD4	54	6	76	112	178	77
CQUAD4	55	6	77	178	177	78
CQUAD4	56	6	78	177	115	79
CQUAD4	57	6	79	115	116	80
CQUAD4	58	6	80	116	174	138
CQUAD4	59	6	138	174	118	137
CQUAD4	60	6	137	118	172	83
CQUAD4	61	6	83	172	120	135
CQUAD4	62	6	135	120	121	85
CQUAD4	63	6	85	121	122	133
CQUAD4	64	6	133	122	123	87
CQUAD4	65	6	87	123	124	88
CQUAD4	66	6	88	124	125	130
CQUAD4	67	6	130	125	165	129
CQUAD4	68	6	129	165	164	128
CQUAD4	69	5	145	146	147	180
CQUAD4	70	5	180	147	292	112
CQUAD4	71	5	112	292	293	178
CQUAD4	72	5	178	293	150	177
CQUAD4	73	5	177	150	295	115
CQUAD4	74	5	115	295	152	116
CQUAD4	75	5	116	152	297	174
CQUAD4	76	5	174	297	298	118
CQUAD4	77	5	118	298	299	172
CQUAD4	78	5	172	299	300	120
CQUAD4	79	5	120	300	301	121
CQUAD4	80	5	121	301	302	122
CQUAD4	81	5	122	302	303	123
CQUAD4	82	5	123	303	160	124
CQUAD4	83	5	124	160	161	125
CQUAD4	84	5	125	161	306	165
CQUAD4	85	5	165	306	307	164
CQUAD4	86	1	182	181	216	183
CQUAD4	87	1	183	216	215	184
CQUAD4	88	1	184	215	214	250
CQUAD4	89	1	250	214	213	249
CQUAD4	90	1	249	213	212	248
CQUAD4	91	1	248	212	211	188
CQUAD4	92	1	188	211	210	246
CQUAD4	93	1	246	210	209	245
CQUAD4	94	1	245	209	208	244
CQUAD4	95	1	244	208	207	243
CQUAD4	96	1	243	207	206	193
CQUAD4	97	1	193	206	205	241
CQUAD4	98	1	241	205	204	195
CQUAD4	99	1	195	204	203	196
CQUAD4	100	1	196	203	202	197
CQUAD4	101	1	197	202	201	198
CQUAD4	102	1	198	201	200	199
CQUAD4	103	2	253	182	183	219
CQUAD4	104	2	219	183	184	220
CQUAD4	105	8	220	184	250	221

TABLE 3-4. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 1 (Continued)

CQUAD4	106	2	221	250	249	222
CQUAD4	107	2	222	249	248	223
CQUAD4	108	2	223	248	188	283
CQUAD4	109	2	283	188	246	282
CQUAD4	110	2	282	246	245	226
CQUAD4	111	2	226	245	244	227
CQUAD4	112	2	227	244	243	228
CQUAD4	113	2	228	243	193	229
CQUAD4	114	2	229	193	241	277
CQUAD4	115	2	277	241	195	231
CQUAD4	116	2	231	195	196	232
CQUAD4	117	2	232	196	197	274
CQUAD4	118	2	274	197	198	273
CQUAD4	119	2	273	198	199	235
CQUAD4	120	3	254	253	219	324
CQUAD4	121	3	324	219	220	323
CQUAD4	122	3	323	220	221	257
CQUAD4	123	3	257	221	222	258
CQUAD4	124	3	258	222	223	259
CQUAD4	125	3	259	223	283	260
CQUAD4	126	3	260	283	282	318
CQUAD4	127	3	318	282	226	317
CQUAD4	128	3	317	226	227	316
CQUAD4	129	3	316	227	228	264
CQUAD4	130	3	264	228	229	314
CQUAD4	131	3	314	229	277	313
CQUAD4	132	3	313	277	231	267
CQUAD4	133	3	267	231	232	268
CQUAD4	134	3	268	232	274	269
CQUAD4	135	3	269	274	273	270
CQUAD4	136	3	270	273	235	308
CQUAD4	137	4	146	254	324	147
CQUAD4	138	4	147	324	323	292
CQUAD4	139	4	292	323	257	293
CQUAD4	140	4	293	257	258	150
CQUAD4	141	4	150	258	259	295
CQUAD4	142	4	295	259	260	152
CQUAD4	143	4	152	260	318	297
CQUAD4	144	4	297	318	317	298
CQUAD4	145	4	298	317	316	299
CQUAD4	146	4	299	316	264	300
CQUAD4	147	4	300	264	314	301
CQUAD4	148	4	301	314	313	302
CQUAD4	149	4	302	313	267	303
CQUAD4	150	4	303	267	268	160
CQUAD4	151	4	160	268	269	161
CQUAD4	152	4	161	269	270	306
CQUAD4	153	4	306	270	308	307
ENDDATA						

The results of the one-half FEM wing model analysis are presented in figure 3-1 in the form of a contour plot. The contours along the outer nodes are in agreement with the imposed boundary conditions. The contours of the inner layers appear to be indicative of the temperature profile of the THERMOD analysis.

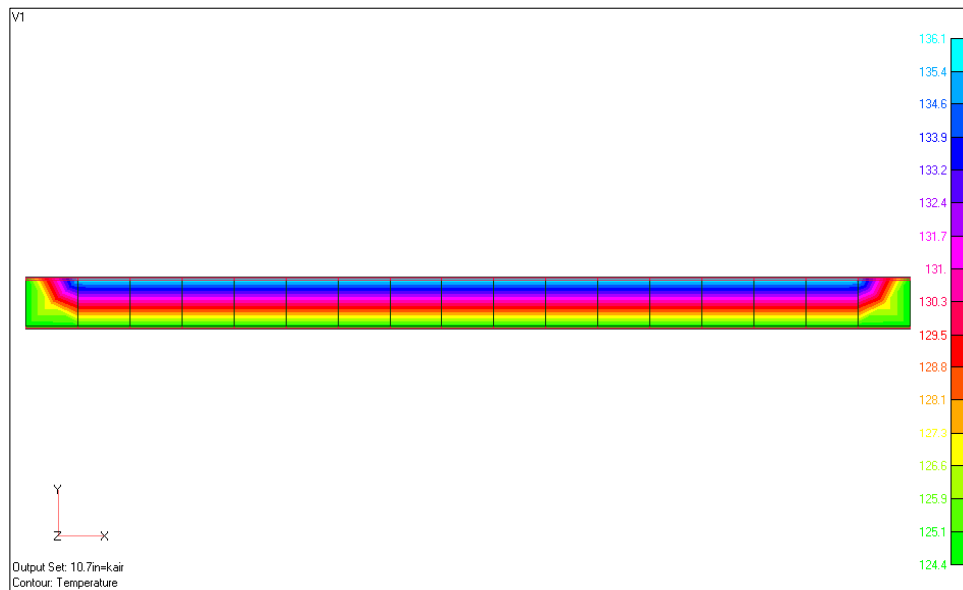


FIGURE 3-1. CONTOUR PLOT OF THE FEM OUTPUT OF SAMPLE PROBLEM 1

The truncated FEM output file is shown in table 3-5. The nodes, from top to bottom of the wing, located at the mid-span of the model are indicated by nodes 209, 245, 226, 317, 298, 118, 137, 101, 10, and 29. The temperatures of these nodes were extracted from this file and are summarized in table 3-6. Mid-span was chosen because it was sufficiently far away from any edge effects that might be present.

TABLE 3-5. TRUNCATED FEM OUTPUT FILE OF SAMPLE PROBLEM 1

1	CONDUCTION IN RECTANGULAR ADIABATIC PLATE (T=0.001FT)					MARCH	6, 2000	UAI/NASTRAN VERSION 20.1	PAGE				
16	10.7 IN GAP FILLED BY AIR; K=0.016BTU/HR/FT/DEGF												
0													
	T E M P E R A T U R E V E C T O R												
POINT ID.	TYPE	ID	VALUE	ID+1	VALUE	ID+2	VALUE	ID+3	VALUE	ID+4	VALUE	ID+5	VALUE
1	TEMP	1.244000E+02											
3	TEMP	1.244011E+02											
5	TEMP	1.244011E+02											
7	TEMP	1.244011E+02											
9	TEMP	1.244011E+02		1.244011E+02		1.244011E+02		1.244011E+02					
14	TEMP	1.244011E+02		1.244011E+02									
18	TEMP	1.244011E+02											
20	TEMP	1.244000E+02		1.244000E+02		1.244000E+02		1.244000E+02		1.244000E+02		1.244000E+02	
26	TEMP	1.244000E+02		1.244000E+02		1.244000E+02		1.244000E+02		1.244000E+02		1.244000E+02	
32	TEMP	1.244000E+02		1.244000E+02		1.244000E+02		1.244000E+02		1.244000E+02		1.244000E+02	
41	TEMP	1.246966E+02											
44	TEMP	1.246966E+02		1.246966E+02									
47	TEMP	1.246966E+02		1.246966E+02									
50	TEMP	1.246966E+02		1.246966E+02		1.246966E+02		1.246961E+02		1.246825E+02		1.244000E+02	
56	TEMP	1.244000E+02											
58	TEMP	1.244011E+02		1.244011E+02									
62	TEMP	1.244011E+02											
67	TEMP	1.244011E+02											
69	TEMP	1.244011E+02											
71	TEMP	1.244011E+02											
73	TEMP	1.244000E+02											
75	TEMP	1.246836E+02		1.246972E+02		1.246977E+02		1.246977E+02		1.246977E+02		1.246977E+02	
83	TEMP	1.246977E+02											
85	TEMP	1.246977E+02											
87	TEMP	1.246977E+02		1.246977E+02									

Temperatures from the THERMOD run is shown in table 3-7, from which the steady-state temperatures T1, T2.....T10 (top to bottom of wing) were extracted and tabulated in table 3-8 for time period 3.

TABLE 3-7. THERMOD OUTPUT FILE (summary.dat) FOR SAMPLE PROBLEM 2

```

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
THERMOD:  A THERMAL PROGRAM FOR AIRCRAFTS
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

SUMMARY OF STEADY-STATE AND TRANSIENT ANALYSIS

(Temperatures are shown for all time periods)

TEMPERATURES AT THE END OF STEADY STATE ANALYSIS

      TEMPERATURES IN DEGREES FAHRENHEIT
      1      2      3      4      5      6      7

1 123.7 132.2 135.9 133.8 126.3 115.2 102.9
2 123.7 132.2 135.9 133.8 126.3 115.2 103.0
3 122.7 131.0 134.7 132.8 125.7 115.4 103.8
4 122.7 131.0 134.7 132.8 125.7 115.4 103.8
5 122.7 130.9 134.6 132.7 125.7 115.4 103.8
6 115.4 121.5 125.9 125.6 121.7 116.2 109.7
7 115.3 121.4 125.8 125.5 121.6 116.2 109.8
8 115.3 121.4 125.8 125.5 121.6 116.2 109.8
9 114.3 120.2 124.6 124.6 121.1 116.4 110.6
10 114.3 120.2 124.6 124.6 121.1 116.4 110.6
11 122.0 131.2 135.6 132.8 124.7 113.6 101.6
12 122.0 131.2 135.6 132.8 124.7 113.6 101.6
13 121.1 130.0 134.5 131.9 124.3 113.8 102.5
14 121.1 130.0 134.5 131.9 124.3 113.8 102.5
15 121.1 129.9 134.4 131.8 124.3 113.9 102.6
16 115.1 121.4 125.9 125.5 121.5 116.0 109.5
17 115.1 121.3 125.8 125.4 121.4 116.0 109.6
18 115.1 121.3 125.8 125.4 121.4 116.0 109.6
19 114.3 120.1 124.6 124.5 121.1 116.3 110.5
20 114.3 120.1 124.6 124.5 121.1 116.3 110.5
21 205.8 225.9 231.9 222.2 200.7 169.4 134.2
22 205.8 225.8 231.8 222.1 200.7 169.5 134.2
23 207.4 216.5 217.2 213.6 202.4 180.2 143.1
24 207.4 216.4 217.2 213.5 202.4 180.2 143.1
25 208.4 209.9 206.9 207.6 203.6 187.7 149.4
  
```

TABLE 3-7. THERMOD OUTPUT FILE (summary.dat) FOR SAMPLE
 PROBLEM 2 (Continued)

26	198.9	198.2	195.5	196.8	195.4	183.4	146.2
27	150.5	153.0	154.4	155.5	153.6	146.5	127.6
28	150.4	153.0	154.4	155.4	153.6	146.5	127.6
29	147.0	149.8	151.5	152.5	150.6	143.9	126.3
30	147.0	149.8	151.5	152.5	150.6	143.9	126.3
31	131.8	135.6	138.7	139.6	137.5	132.4	120.5
32	131.8	135.6	138.7	139.6	137.5	132.4	120.5
33	131.8	135.6	138.7	139.6	137.5	132.4	120.5
34	131.8	135.6	138.7	139.6	137.5	132.4	120.5
35	116.7	121.5	125.8	126.6	124.5	120.8	114.7
36	116.7	121.5	125.8	126.6	124.4	120.8	114.7
37	113.2	118.3	122.9	123.7	121.5	118.2	113.3
38	113.2	118.3	122.9	123.7	121.4	118.2	113.3
39	199.5	179.7	148.8	180.0	196.6	194.7	173.9
40	199.5	179.7	148.9	180.1	196.6	194.6	173.9
41	197.6	186.6	168.7	186.2	194.5	188.4	162.3
42	197.6	186.6	168.8	186.2	194.5	188.4	162.3
43	196.2	191.4	182.7	190.5	193.1	184.1	154.2
44	191.8	188.5	182.6	187.8	189.2	180.4	151.1
45	169.9	170.8	168.7	171.5	170.1	161.4	138.9
46	169.8	170.7	168.7	171.5	170.0	161.3	138.8
47	138.6	145.3	148.8	148.2	142.9	134.1	121.3
48	138.4	145.2	148.7	148.0	142.7	133.9	121.2
49	145.6	148.6	149.7	150.4	147.7	139.7	123.0
50	147.6	150.5	151.3	152.1	149.5	141.3	123.9
51	113.6	120.9	125.1	123.7	118.1	110.0	100.6
52	108.2	115.1	119.2	118.0	112.7	105.1	96.2
53	113.6	120.9	125.1	123.7	118.1	110.0	100.6
Maximum Structural Temperature for Period 1 = 207.37 Occurring at Location 24							
Maximum Structural Temperature for Period 2 = 225.86 Occurring at Location 21							
Maximum Structural Temperature for Period 3 = 231.88 Occurring at Location 21							
Maximum Structural Temperature for Period 4 = 222.15 Occurring at Location 21							
Maximum Structural Temperature for Period 5 = 202.43 Occurring at Location 24							
Maximum Structural Temperature for Period 6 = 194.68 Occurring at Location 39							
Maximum Structural Temperature for Period 7 = 173.93 Occurring at Location 39							
Maximum Structural Temperature Over All 7 Periods =231.88 Occurring at Location 21 at Period 3							

TABLE 3-7. THERMOD OUTPUT FILE (summary.dat) FOR SAMPLE
 PROBLEM 2 (Continued)

TEMPERATURES AT THE END OF TRANSIENT ANALYSIS							
TEMPERATURES IN DEGREES FAHRENHEIT							
	1	2	3	4	5	6	7
1	112.8	116.8	121.7	123.8	123.4	122.5	119.8
2	113.2	117.3	122.1	124.1	123.5	122.3	119.3
3	122.0	129.9	133.7	132.1	125.5	115.9	104.9
4	122.3	130.3	134.1	132.3	125.6	115.7	104.5
5	122.3	130.3	134.1	132.3	125.5	115.5	104.3
6	115.5	121.7	126.1	125.8	121.9	116.4	109.8
7	115.3	121.3	125.7	125.5	121.8	116.6	110.2
8	115.2	121.2	125.6	125.5	121.9	116.7	110.5
9	112.8	116.7	121.6	123.8	123.6	123.1	120.8
10	112.7	116.5	121.5	123.8	123.7	123.3	121.2
11	112.7	116.7	121.6	123.7	123.3	122.4	119.7
12	113.0	117.2	122.1	124.0	123.3	122.1	119.2
13	120.5	129.1	133.6	131.3	124.2	114.4	103.8
14	120.7	129.4	133.9	131.5	124.3	114.2	103.3
15	120.8	129.4	133.9	131.5	124.2	114.1	103.1
16	115.3	121.6	126.0	125.6	121.6	116.1	109.6
17	115.0	121.2	125.7	125.4	121.6	116.3	110.1
18	115.0	121.1	125.6	125.4	121.7	116.5	110.4
19	112.8	116.6	121.6	123.8	123.6	123.1	120.8
20	112.7	116.5	121.5	123.8	123.7	123.3	121.2
21	125.4	131.3	136.7	137.5	134.8	130.9	124.7
22	127.4	133.3	138.7	139.3	136.4	132.3	125.2
23	181.2	187.5	192.2	187.4	180.5	171.4	140.3
24	183.1	189.2	193.8	188.9	182.0	172.9	140.9
25	200.7	202.6	205.1	200.1	195.7	189.2	147.8
26	185.7	185.5	187.8	185.4	184.4	181.0	145.5
27	150.7	153.2	154.6	155.7	153.8	146.7	127.7
28	149.5	152.1	153.7	154.7	152.8	145.8	127.3
29	146.5	149.3	151.2	152.1	150.2	143.7	126.3
30	145.7	148.6	150.4	151.4	149.5	143.1	126.0
31	131.5	135.1	138.2	139.3	137.7	133.2	121.9
32	131.5	135.1	138.2	139.3	137.7	133.2	121.9
33	131.5	135.1	138.2	139.3	137.7	133.2	121.9
34	131.5	135.1	138.2	139.3	137.7	133.2	121.9
35	117.2	121.6	125.9	127.3	125.8	123.2	117.7
36	116.4	120.7	125.2	126.5	125.2	122.7	117.5
37	112.1	115.7	120.6	123.1	123.4	123.4	121.5
38	112.0	115.5	120.5	123.0	123.3	123.4	121.6
39	121.2	122.3	123.8	129.2	131.3	131.5	127.8
40	123.2	124.0	125.1	130.7	133.0	133.1	128.7
41	172.1	166.9	160.2	168.7	173.2	172.1	150.2

TABLE 3-7. THERMOD OUTPUT FILE (summary.dat) FOR SAMPLE
 PROBLEM 2 (Continued)

42	173.6	168.3	161.5	170.0	174.5	173.3	150.8
43	183.1	179.5	176.6	179.9	182.2	181.5	152.3
44	179.2	177.0	176.6	177.6	178.7	178.0	149.3
45	162.3	161.7	161.5	163.9	164.5	162.3	141.2
46	160.8	160.3	160.1	162.6	163.2	161.0	140.6
47	121.5	123.0	125.1	129.8	131.5	131.4	127.2
48	120.0	121.6	123.8	128.6	130.3	130.3	126.7
49	123.6	126.8	131.0	132.5	132.0	130.4	123.1
50	127.8	130.8	134.8	136.0	135.3	133.2	123.9
51	169.1	187.2	193.0	184.0	164.8	137.5	109.0
52	157.8	174.5	180.0	172.0	154.5	129.7	103.8
53	169.1	187.2	193.0	184.0	164.8	137.5	109.0
Maximum Structural Temperature for Period 1 = 183.07 Occurring at Location 24							
Maximum Structural Temperature for Period 2 = 189.17 Occurring at Location 24							
Maximum Structural Temperature for Period 3 = 193.83 Occurring at Location 24							
Maximum Structural Temperature for Period 4 = 188.85 Occurring at Location 24							
Maximum Structural Temperature for Period 5 = 181.98 Occurring at Location 24							
Maximum Structural Temperature for Period 6 = 173.32 Occurring at Location 42							
Maximum Structural Temperature for Period 7 = 150.76 Occurring at Location 42							
Maximum Structural Temperature Over All 7 Periods =193.83 Occurring at Location 24 at Period 3							

TABLE 3-8. THERMOD-SIMULATED TEMPERATURES (°F) OF THE RIGHT WING OF
 SAMPLE PROBLEM 2

T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
135.9	135.9	134.7	134.7	134.6	125.9	125.8	125.8	124.6	124.6

The results of the one-half FEM wing model analysis are presented in figure 3-2 in the form of a contour plot. The contours along the outer nodes are in agreement with the imposed boundary conditions while the contours of the inner layers appear to be indicative of the temperature profile of the THERMOD analysis.



FIGURE 3-2. CONTOUR PLOT OF THE FEM OUTPUT OF SAMPLE PROBLEM 2

The truncated FEM output file is shown in table 3-9. The nodes, from top to bottom of the wing, located at mid-span of the model are indicated by nodes 209, 245, 226, 317, 298, 118, 137, 101, 10, and 29. The temperatures of these nodes were extracted from this file and are reproduced in table 3-10. Mid-span was chosen because it was sufficiently far away from any edge effects that might be present.

TABLE 3-9. TRUNCATED FEM OUTPUT FILE OF SAMPLE PROBLEM 2

1	CONDUCTION IN RECTANGULAR ADIABATIC PLATE (T=0.001FT)					MARCH	6, 2000	UAI/NASTRAN VERSION 20.1	PAGE				
16	10.7 IN GAP FILLED BY OIL; K=0.081BTU/HR/FT/DEGF												
0													
			T E M P E R A T U R E V E C T O R										
POINT ID.	TYPE	ID	VALUE	ID+1	VALUE	ID+2	VALUE	ID+3	VALUE	ID+4	VALUE	ID+5	VALUE
1	TEMP		1.246000E+02										
3	TEMP		1.246039E+02										
5	TEMP		1.246044E+02										
7	TEMP		1.246044E+02										
9	TEMP		1.246044E+02	1.246044E+02		1.246044E+02		1.246044E+02					
14	TEMP		1.246044E+02	1.246044E+02									
18	TEMP		1.246039E+02										
20	TEMP		1.246000E+02	1.246000E+02		1.246000E+02		1.246000E+02		1.246000E+02		1.246000E+02	
26	TEMP		1.246000E+02	1.246000E+02		1.246000E+02		1.246000E+02		1.246000E+02		1.246000E+02	
32	TEMP		1.246000E+02	1.246000E+02		1.246000E+02		1.246000E+02		1.246000E+02		1.246000E+02	
41	TEMP		1.257877E+02										
44	TEMP		1.257885E+02	1.257885E+02									
47	TEMP		1.257885E+02	1.257885E+02									
50	TEMP		1.257885E+02	1.257885E+02		1.257877E+02		1.257777E+02		1.256599E+02		1.246000E+02	
56	TEMP		1.246000E+02										
58	TEMP		1.246044E+02	1.246044E+02									
62	TEMP		1.246044E+02										
67	TEMP		1.246044E+02										
69	TEMP		1.246044E+02										
71	TEMP		1.246044E+02										
73	TEMP		1.246000E+02										
75	TEMP		1.256639E+02	1.257821E+02		1.257921E+02		1.257929E+02		1.257929E+02		1.257929E+02	
83	TEMP		1.257929E+02										
85	TEMP		1.257929E+02										
87	TEMP		1.257929E+02	1.257921E+02									

TABLE 3-9. TRUNCATED FEM OUTPUT FILE OF SAMPLE PROBLEM 2 (Continued)

	98	TEMP	1.257885E+02																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																										
--	----	------	--------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

When compared, the inner temperatures of tables 3-8 and 3-10 show excellent agreement, again validating the conduction capability of the THERMOD program.

TABLE 3-10. FINITE ELEMENT METHOD-SIMULATED TEMPERATURES (°F) OF THE RIGHT WING OF SAMPLE PROBLEM 2

T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
135.9	135.9	134.7	134.7	134.6	125.9	125.8	125.8	124.6	124.6

3.3 SAMPLE PROBLEM 3.

Sample problem 3, isolated and studied the effects of surface convection. Convection was introduced into the system by surface airflow. Depending on the ambient and surface temperatures, heat may either be added or removed from the system.

As in sample problems 1 and 2, a complete aircraft problem was solved using the THERMOD program. A particular item of interest was that the absorptivity and emissivity of all external surfaces, i.e., the fuselage, wing, and tarmac surfaces, were assigned 1.0 and 0.0 values, respectively. Such surfaces absorb all heat and do not emit any energy (no radiation). The only way for surface heat transfer to take place was through conduction and convection. Table 3-11 shows the THERMOD input file for this sample problem.

TABLE 3-11. THERMOD INPUT FILE (input.dat) OF SAMPLE PROBLEM 3

```
198e-6,0.704
1, 0.020,125,0.300,0.30
2, 0.375,4.4,0.021,0.24
3, 0.020,125,0.300,0.30
4, 0.350,125,0.300,0.30
5, 10.70,0.067,0.016,0.24
6, 0.350,125,0.300,0.30
7, 0.020,125,0.300,0.30
8, 0.375,4.4,0.021,0.24
9, 0.020,125,0.300,0.30
10,0.030,125,0.300,0.30
11,0.375,4.4,0.021,0.24
12,0.020,125,0.300,0.30
13,0.250,10.,0.020,0.24
14,48.00,0.067,0.016,0.24
15,0.020,125,0.300,0.30
16,0.375,4.4,0.021,0.24
17,0.020,125,0.300,0.30
18,0.250,10.,0.020,0.24
19,30.00,0.067,0.016,0.24
20,5.000,10.,0.020,0.24
21,0.020,125,0.300,0.30
22,0.375,4.4,0.021,0.24
23,0.020,125,0.300,0.30
24,2.500,0.067,0.016,0.24
25,0.020,125,0.300,0.30
26,0.375,4.4,0.021,0.24
27,0.020,125,0.300,0.30
28,0.250,130,0.800,0.20
3.5,34.0,2.5,5.5,22.0,11.0,3.0,4.0,30,24,28
0.70,1.0,0.0,2.0,150,0.80,0.22
0.3,0.9,0.3,0.9,0.3,0.9,0.1,0.9
3,100
0.9,0.9,0.9,0.9
1.0,0.0,1.0,0.0,1.0,0.0,1.0,0.0
9
0,10,10,0,0,75,110,110,190,190
5,120,5,60,20,5,120,60,60
7
60 570.67 459.67 14 330 10 10 0.36
```

TABLE 3-11. THERMOD INPUT FILE (input.dat) OF SAMPLE PROBLEM 3 (Continued)

75	573.67	459.67	14	355	10	10	0.31
90	578.67	459.67	14	355	10	12	0.33
75	581.67	459.67	14	330	10	10	0.31
60	582.67	459.67	14	291	10	10	0.36
45	583.67	459.67	14	231	10	10	0.44
30	582.67	459.67	14	160	10	10	0.57
2,100,180,1							
10,1.0,1.0							
10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,							
10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,							
10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,							
10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,							
10,10,10,10,10,10,10,10							

Temperatures from the THERMOD run is shown in table 3-12, from which the steady-state temperatures T1, T2.....T10 (top to bottom of wing) for time period 3 were extracted and tabulated in table 3-13.

TABLE 3-12. THERMOD OUTPUT FILE (summary.dat) FOR SAMPLE PROBLEM 3

\$							
THERMOD: A THERMAL PROGRAM FOR AIRCRAFTS							
\$							
SUMMARY OF STEADY-STATE AND TRANSIENT ANALYSIS							
(Temperatures are shown for all time periods)							
TEMPERATURES AT THE END OF STEADY STATE ANALYSIS							
TEMPERATURES IN DEGREES FAHRENHEIT							
	1	2	3	4	5	6	7
1	321.1	366.1	380.0	356.4	308.3	244.1	181.8
2	321.1	366.1	380.0	356.3	308.3	244.1	181.8
3	315.9	359.8	373.5	350.5	303.7	241.1	180.3
4	315.8	359.8	373.4	350.5	303.6	241.1	180.3
5	315.5	359.4	373.0	350.1	303.3	240.9	180.2
6	119.2	123.9	129.2	131.2	130.3	128.7	125.3
7	118.9	123.5	128.8	130.8	130.0	128.5	125.2
8	118.9	123.5	128.8	130.8	129.9	128.5	125.2

TABLE 3-12. THERMOD OUTPUT FILE (summary.dat) FOR SAMPLE
 PROBLEM 3 (Continued)

9	113.6	117.2	122.3	124.9	125.3	125.5	123.7
10	113.6	117.1	122.3	124.9	125.3	125.5	123.7
11	321.1	366.1	380.0	356.4	308.3	244.1	181.8
12	321.1	366.1	380.0	356.3	308.3	244.1	181.8
13	315.9	359.8	373.5	350.5	303.7	241.1	180.3
14	315.8	359.8	373.4	350.5	303.6	241.1	180.3
15	315.5	359.4	373.0	350.1	303.3	240.9	180.2
16	119.2	123.9	129.2	131.2	130.3	128.7	125.3
17	118.9	123.5	128.8	130.8	130.0	128.5	125.2
18	118.9	123.5	128.8	130.8	129.9	128.5	125.2
19	113.6	117.2	122.3	124.9	125.3	125.5	123.7
20	113.6	117.1	122.3	124.9	125.3	125.5	123.7
21	277.2	308.5	317.8	302.4	269.0	222.0	171.4
22	277.1	308.3	317.6	302.2	268.9	221.9	171.4
23	244.4	257.4	258.7	253.9	238.6	210.2	166.7
24	244.3	257.2	258.5	253.7	238.5	210.2	166.7
25	221.4	221.6	217.3	219.9	217.3	202.0	163.4
26	194.1	189.2	184.0	189.0	192.2	184.4	150.1
27	147.9	147.4	147.9	151.8	153.7	150.8	135.0
28	147.9	147.4	147.9	151.8	153.7	150.8	135.0
29	144.6	144.4	145.3	149.1	151.0	148.4	134.0
30	144.6	144.4	145.3	149.1	151.0	148.4	134.0
31	130.2	131.3	134.0	137.5	138.9	137.9	129.2
32	130.2	131.3	134.0	137.5	138.9	137.9	129.2
33	130.2	131.3	134.0	137.5	138.9	137.9	129.2
34	130.2	131.3	134.0	137.5	138.9	137.9	129.2
35	115.7	118.3	122.7	125.8	126.9	127.4	124.5
36	115.7	118.3	122.7	125.8	126.9	127.4	124.5
37	112.4	115.3	120.1	123.1	124.2	125.0	123.5
38	112.4	115.3	120.1	123.1	124.2	125.0	123.5
39	210.7	173.9	128.8	177.2	210.2	219.4	200.8
40	210.7	174.0	129.0	177.3	210.2	219.3	200.6
41	200.6	178.8	152.2	180.6	199.7	201.5	178.1
42	200.6	178.8	152.3	180.6	199.6	201.4	178.0
43	193.6	182.2	168.6	183.0	192.3	189.0	162.3
44	186.0	177.5	168.6	178.6	185.6	182.2	156.6
45	161.4	156.6	152.3	160.0	165.0	163.1	145.6
46	161.3	156.5	152.2	159.9	164.9	163.0	145.5
47	126.1	126.8	129.0	133.4	135.6	135.7	129.8
48	125.9	126.6	128.8	133.2	135.4	135.6	129.7
49	126.2	124.5	123.5	127.4	129.7	127.1	114.5
50	128.7	126.7	125.5	129.6	132.0	129.2	115.7
51	188.8	207.4	215.7	208.8	191.6	168.5	144.8
52	183.9	201.5	209.6	203.3	187.3	165.7	143.4
53	188.8	207.4	215.7	208.8	191.6	168.5	144.8

TABLE 3-12. THERMOD OUTPUT FILE (summary.dat) FOR SAMPLE
 PROBLEM 3 (Continued)

Maximum Structural Temperature for Period	1 = 321.12	Occurring					
at Location 1							
Maximum Structural Temperature for Period	2 = 366.11	Occurring					
at Location 1							
Maximum Structural Temperature for Period	3 = 380.01	Occurring					
at Location 1							
Maximum Structural Temperature for Period	4 = 356.36	Occurring					
at Location 1							
Maximum Structural Temperature for Period	5 = 308.29	Occurring					
at Location 1							
Maximum Structural Temperature for Period	6 = 244.09	Occurring					
at Location 1							
Maximum Structural Temperature for Period	7 = 200.77	Occurring					
at Location 39							
Maximum Structural Temperature Over All	7 Periods =380.01						
Occurring at Location 1 at Period 3							
TEMPERATURES AT THE END OF TRANSIENT ANALYSIS							
TEMPERATURES IN DEGREES FAHRENHEIT							
	1	2	3	4	5	6	7
1	129.1	135.7	141.4	142.2	138.9	134.3	128.1
2	134.7	142.5	148.5	148.5	143.9	137.6	129.6
3	301.7	342.8	355.8	334.6	291.1	233.0	176.4
4	306.4	348.5	361.8	340.0	295.3	235.7	177.7
5	306.1	348.2	361.4	339.7	295.1	235.5	177.6
6	124.1	129.7	135.2	136.6	134.5	131.5	126.7
7	118.8	123.3	128.7	130.7	129.9	128.4	125.2
8	118.6	123.1	128.4	130.4	129.7	128.3	125.1
9	111.4	114.5	119.6	122.5	123.4	124.3	123.1
10	111.2	114.3	119.3	122.3	123.2	124.1	123.1
11	129.1	135.7	141.4	142.2	138.9	134.3	128.1
12	134.7	142.5	148.5	148.5	143.9	137.6	129.6
13	301.7	342.8	355.8	334.6	291.1	233.0	176.4
14	306.4	348.5	361.8	340.0	295.3	235.7	177.7
15	306.1	348.2	361.4	339.7	295.1	235.5	177.6
16	124.1	129.7	135.2	136.6	134.5	131.5	126.7
17	118.8	123.3	128.7	130.7	129.9	128.4	125.2
18	118.6	123.1	128.4	130.4	129.7	128.3	125.1
19	111.4	114.5	119.6	122.5	123.4	124.3	123.1
20	111.2	114.3	119.3	122.3	123.2	124.1	123.1
21	130.8	137.4	142.9	143.7	140.4	135.8	128.8
22	133.5	140.4	145.8	146.4	142.7	137.7	129.7
23	203.9	216.1	216.0	215.1	202.6	188.7	155.1

TABLE 3-12. THERMOD OUTPUT FILE (summary.dat) FOR SAMPLE
 PROBLEM 3 (Continued)

24	205.8	217.8	217.4	216.7	204.1	190.2	155.8
25	210.8	216.5	210.1	213.7	205.9	197.6	157.6
26	186.1	186.3	181.1	186.4	185.3	182.1	149.5
27	148.1	147.5	148.0	151.9	153.9	150.9	135.1
28	147.0	146.6	147.2	151.1	153.0	150.2	134.8
29	144.2	144.1	145.1	148.9	150.7	148.2	133.9
30	143.4	143.4	144.4	148.2	150.0	147.6	133.6
31	129.7	131.0	133.7	137.1	138.6	137.6	129.1
32	129.7	131.0	133.7	137.1	138.6	137.6	129.1
33	129.7	131.0	133.7	137.1	138.6	137.6	129.1
34	129.7	131.0	133.7	137.1	138.6	137.6	129.1
35	116.0	118.5	122.9	126.0	127.2	127.6	124.6
36	115.2	117.8	122.3	125.4	126.5	127.0	124.4
37	111.3	114.3	119.2	122.2	123.2	124.2	123.1
38	111.2	114.1	119.1	122.1	123.1	124.1	123.1
39	119.2	119.2	120.3	126.8	130.2	131.8	129.2
40	121.4	120.9	121.3	128.3	132.0	133.7	130.5
41	174.2	164.5	149.6	167.4	176.6	178.9	159.7
42	175.8	166.0	150.9	168.8	178.0	180.2	160.4
43	185.3	179.9	167.2	180.8	185.2	186.1	160.0
44	178.6	175.6	167.2	176.9	179.3	179.9	154.7
45	159.1	156.3	150.9	159.8	163.2	163.9	146.6
46	157.5	154.8	149.6	158.4	161.9	162.6	145.9
47	119.0	119.6	121.3	127.1	129.9	131.4	128.6
48	117.6	118.3	120.3	125.9	128.8	130.2	127.9
49	116.9	118.7	121.3	124.6	125.7	125.5	119.8
50	120.3	121.9	123.7	127.3	128.2	127.8	120.4
51	306.1	348.1	361.3	339.6	295.0	235.5	177.6
52	288.9	327.5	340.0	320.5	279.9	225.7	172.8
53	306.1	348.1	361.3	339.6	295.0	235.5	177.6
Maximum Structural Temperature for Period					1 = 306.45	Occurring	
at Location 4							
Maximum Structural Temperature for Period					2 = 348.51	Occurring	
at Location 4							
Maximum Structural Temperature for Period					3 = 361.78	Occurring	
at Location 4							
Maximum Structural Temperature for Period					4 = 339.99	Occurring	
at Location 4							
Maximum Structural Temperature for Period					5 = 295.35	Occurring	
at Location 4							
Maximum Structural Temperature for Period					6 = 235.71	Occurring	
at Location 4							
Maximum Structural Temperature for Period					7 = 177.71	Occurring	
at Location 4							
Maximum Structural Temperature Over All					7 Periods = 361.78		
Occurring at Location 4 at Period 3							

TABLE 3-13. THERMOD-SIMULATED TEMPERATURES (°F) OF THE RIGHT WING OF
SAMPLE PROBLEM 3

T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
380.0	380.0	373.5	373.4	373.0	129.2	128.8	128.8	122.3	122.3

As in sample problems 1 and 2, a one-half wing structure was modeled using the UAI/NASTRAN FEM program. Boundary conditions were imposed based on the THERMOD-simulated data for period 3. The wing top was constrained to 380.0°F, and the wing sides were constrained to 124.4°F, which happens to be the highest ambient temperature (time period 6 of table 3-11). Because the ratio of the wing length to its depth is sufficiently high, edge effects due to the wing sides (and the boundary temperature used) are insignificant, as the FEM results show. The bottom wing surface was free of any constraint, except that it was subjected to convection, with a film coefficient $h = 1.3433 \text{ (Btu/hr)/(ft}^2\text{°F)}$. This film coefficient was calculated internally by the THERMOD program and is based on the theory of airflow on a flat surface and a wind speed of 14 ft/sec.

The FEM input file is shown in table 3-14. The boundary conditions were imposed by the SPC cards. An ambient temperature of 119.0°F (pertaining to period 3 of the THERMOD input) was applied through the scalar point, SPOINT, bulk data. The boundary elements, CHBDY, through which convection is realized, were modeled using the LINE elements connecting the nodes at the wing bottom. Property cards PHBDY and material cards MAT4 simulated the film coefficient and other element properties.

TABLE 3-14. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE PROBLEM 3

```

ID D:\Natha, MSC/N
APP HEAT
SOL 1
TIME 10000
CEND
  TITLE = CONDUCTION & CONVECTION IN RECT. ADIABATIC PLATE (t=0.001ft)
  SUBTITLE = 10.7 in air gap; k=0.016BTU/hr/ft/degF; h=1.3433BTU/hr/ft^2/degF
  SPC = 1
  THERMAL= ALL
  FLUX = ALL
  SPCF = ALL
BEGIN BULK
$ *****
$   Written by : MSC/NASTRAN for Windows
$   Version   : 6.00
$   Translator : UAI/NASTRAN
$   From Model : D:\Thermod_Validation\FEM\conduction&convection.MOD
$   Date      : Tue Mar 07 08:27:39 2000
$ *****
$
PARAM, K6ROT, 100.
PARAM, MAXRATIO, 1.E+8
CORD2C      1      0      0.      0.      0.      0.      0.      1.+MSC/NC1
+MSC/NC1    1.      0.      1.
CORD2S      2      0      0.      0.      0.      0.      0.      1.+MSC/NC2
+MSC/NC2    1.      0.      1.

```

TABLE 3-14. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 3 (Continued)

```
$ MSC/NASTRAN for Windows Constraint Set 1 : bound_temp&ambient constr
SPC      1      37      1 124.4
SPC      1      55      1 124.4
SPC      1      56      1 124.4
SPC      1      73      1 124.4
SPC      1     109      1 124.4
SPC      1     128      1 124.4
SPC      1     145      1 124.4
SPC      1     146      1 124.4
SPC      1     164      1 124.4
SPC      1     181      1 124.4
SPC      1     182      1 124.4
SPC      1     199      1 124.4
SPC      1     200      1 124.4
SPC      1     201      1 380.0
SPC      1     202      1 380.0
SPC      1     203      1 380.0
SPC      1     204      1 380.0
SPC      1     205      1 380.0
SPC      1     206      1 380.0
SPC      1     207      1 380.0
SPC      1     208      1 380.0
SPC      1     209      1 380.0
SPC      1     210      1 380.0
SPC      1     211      1 380.0
SPC      1     212      1 380.0
SPC      1     213      1 380.0
SPC      1     214      1 380.0
SPC      1     215      1 380.0
SPC      1     216      1 380.0
SPC      1     235      1 124.4
SPC      1     253      1 124.4
SPC      1     254      1 124.4
SPC      1     307      1 124.4
SPC      1     308      1 124.4
spc,1,500,1,119.0
spoint,500
chbdy,1000,100,line,1,36,,,,500,500
chbdy,1001,100,line,36,35,,,,500,500
chbdy,1002,100,line,35,34,,,,500,500
chbdy,1003,100,line,34,33,,,,500,500
chbdy,1004,100,line,33,32,,,,500,500
chbdy,1005,100,line,32,31,,,,500,500
chbdy,1006,100,line,31,30,,,,500,500
chbdy,1007,100,line,30,29,,,,500,500
chbdy,1008,100,line,29,28,,,,500,500
chbdy,1009,100,line,28,27,,,,500,500
chbdy,1010,100,line,27,26,,,,500,500
chbdy,1011,100,line,26,25,,,,500,500
chbdy,1012,100,line,25,24,,,,500,500
chbdy,1013,100,line,24,23,,,,500,500
chbdy,1014,100,line,23,22,,,,500,500
chbdy,1015,100,line,22,21,,,,500,500
chbdy,1016,100,line,21,20,,,,500,500
phbdy,100,200,0.001
mat4,200,1.3433
$mat4,200,0.0001
```

TABLE 3-14. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 3 (Continued)

\$							
\$ MSC/NASTRAN	for	Windows	Property 1 :	0.02 in glass_outer_layer			
PSHELL	1	1	0.001	1	1		0.
\$ MSC/NASTRAN	for	Windows	Property 2 :	3/8 in foam			
PSHELL	2	2	0.001	2	2		0.
\$ MSC/NASTRAN	for	Windows	Property 3 :	0.02 in inner glass			
PSHELL	3	1	0.001	1	1		0.
\$ MSC/NASTRAN	for	Windows	Property 4 :	0.350 in cap			
PSHELL	4	1	0.001	1	1		0.
\$ MSC/NASTRAN	for	Windows	Property 5 :	10.70 in air space			
PSHELL	5	3	0.001	3	3		0.
\$ MSC/NASTRAN	for	Windows	Property 6 :	0.350 in cap			
PSHELL	6	1	0.001	1	1		0.
\$ MSC/NASTRAN	for	Windows	Property 7 :	0.02 in inner glass			
PSHELL	7	1	0.001	1	1		0.
\$ MSC/NASTRAN	for	Windows	Property 8 :	3/8 in foam			
PSHELL	8	2	0.001	2	2		0.
\$ MSC/NASTRAN	for	Windows	Property 9 :	0.02 in glass_outer_layer			
PSHELL	9	1	0.001	1	1		0.
\$ MSC/NASTRAN	for	Windows	Property 10 :	10.70 in oil			
PSHELL	10	4	0.001	4	4		0.
\$ MSC/NASTRAN	for	Windows	Material 1 :	glass			
MAT4	1	0.3	1.				
\$ MSC/NASTRAN	for	Windows	Material 2 :	foam			
MAT4	2	0.021	1.				
\$ MSC/NASTRAN	for	Windows	Material 3 :	air			
MAT4	3	0.016	1.				
\$ MSC/NASTRAN	for	Windows	Material 4 :	oil			
MAT4	4	0.081	1.				
GRID	1	0	0.	0.	0.	0	
GRID	3	0	1.1.667E-3	0.	0.	0	
GRID	5	0	3.1.667E-3	0.	0.	0	
GRID	7	0	5.1.667E-3	0.	0.	0	
GRID	9	0	7.1.667E-3	0.	0.	0	
GRID	10	0	8.1.667E-3	0.	0.	0	
GRID	11	0	9.1.667E-3	0.	0.	0	
GRID	12	0	10.1.667E-3	0.	0.	0	
GRID	14	0	12.1.667E-3	0.	0.	0	
GRID	15	0	13.1.667E-3	0.	0.	0	
GRID	18	0	16.1.667E-3	0.	0.	0	
GRID	20	0	17.	0.	0.	0	
GRID	21	0	16.	0.	0.	0	
GRID	22	0	15.	0.	0.	0	
GRID	23	0	14.	0.	0.	0	
GRID	24	0	13.	0.	0.	0	
GRID	25	0	12.	0.	0.	0	
GRID	26	0	11.	0.	0.	0	
GRID	27	0	10.	0.	0.	0	
GRID	28	0	9.	0.	0.	0	
GRID	29	0	8.	0.	0.	0	
GRID	30	0	7.	0.	0.	0	
GRID	31	0	6.	0.	0.	0	
GRID	32	0	5.	0.	0.	0	
GRID	33	0	4.	0.	0.	0	
GRID	34	0	3.	0.	0.	0	
GRID	35	0	2.	0.	0.	0	
GRID	36	0	1.	0.	0.	0	

TABLE 3-14. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 3 (Continued)

GRID	37	0	0.1.667E-3	0.	0
GRID	41	0	3.0.032917	0.	0
GRID	44	0	6.0.032917	0.	0
GRID	45	0	7.0.032917	0.	0
GRID	47	0	9.0.032917	0.	0
GRID	48	0	10.0.032917	0.	0
GRID	50	0	12.0.032917	0.	0
GRID	51	0	13.0.032917	0.	0
GRID	52	0	14.0.032917	0.	0
GRID	53	0	15.0.032917	0.	0
GRID	54	0	16.0.032917	0.	0
GRID	55	0	17.0.032917	0.	0
GRID	56	0	17.1.667E-3	0.	0
GRID	58	0	15.1.667E-3	0.	0
GRID	59	0	14.1.667E-3	0.	0
GRID	62	0	11.1.667E-3	0.	0
GRID	67	0	6.1.667E-3	0.	0
GRID	69	0	4.1.667E-3	0.	0
GRID	71	0	2.1.667E-3	0.	0
GRID	73	0	0.0.032917	0.	0
GRID	75	0	1.0.034583	0.	0
GRID	76	0	2.0.034583	0.	0
GRID	77	0	3.0.034583	0.	0
GRID	78	0	4.0.034583	0.	0
GRID	79	0	5.0.034583	0.	0
GRID	80	0	6.0.034583	0.	0
GRID	83	0	9.0.034583	0.	0
GRID	85	0	11.0.034583	0.	0
GRID	87	0	13.0.034583	0.	0
GRID	88	0	14.0.034583	0.	0
GRID	98	0	11.0.032917	0.	0
GRID	101	0	8.0.032917	0.	0
GRID	104	0	5.0.032917	0.	0
GRID	105	0	4.0.032917	0.	0
GRID	107	0	2.0.032917	0.	0
GRID	108	0	1.0.032917	0.	0
GRID	109	0	0.0.034583	0.	0
GRID	112	0	2. 0.06375	0.	0
GRID	115	0	5. 0.06375	0.	0
GRID	116	0	6. 0.06375	0.	0
GRID	118	0	8. 0.06375	0.	0
GRID	120	0	10. 0.06375	0.	0
GRID	121	0	11. 0.06375	0.	0
GRID	122	0	12. 0.06375	0.	0
GRID	123	0	13. 0.06375	0.	0
GRID	124	0	14. 0.06375	0.	0
GRID	125	0	15. 0.06375	0.	0
GRID	128	0	17.0.034583	0.	0
GRID	129	0	16.0.034583	0.	0
GRID	130	0	15.0.034583	0.	0
GRID	133	0	12.0.034583	0.	0
GRID	135	0	10.0.034583	0.	0
GRID	137	0	8.0.034583	0.	0
GRID	138	0	7.0.034583	0.	0
GRID	145	0	0. 0.06375	0.	0
GRID	146	0	0. 0.95542	0.	0
GRID	147	0	1. 0.95542	0.	0

TABLE 3-14. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 3 (Continued)

GRID	150	0	4.	0.95542	0.	0
GRID	152	0	6.	0.95542	0.	0
GRID	160	0	14.	0.95542	0.	0
GRID	161	0	15.	0.95542	0.	0
GRID	164	0	17.	0.06375	0.	0
GRID	165	0	16.	0.06375	0.	0
GRID	172	0	9.	0.06375	0.	0
GRID	174	0	7.	0.06375	0.	0
GRID	177	0	4.	0.06375	0.	0
GRID	178	0	3.	0.06375	0.	0
GRID	180	0	1.	0.06375	0.	0
GRID	181	0	0.	1.01917	0.	0
GRID	182	0	0.	1.0175	0.	0
GRID	183	0	1.	1.0175	0.	0
GRID	184	0	2.	1.0175	0.	0
GRID	188	0	6.	1.0175	0.	0
GRID	193	0	11.	1.0175	0.	0
GRID	195	0	13.	1.0175	0.	0
GRID	196	0	14.	1.0175	0.	0
GRID	197	0	15.	1.0175	0.	0
GRID	198	0	16.	1.0175	0.	0
GRID	199	0	17.	1.0175	0.	0
GRID	200	0	17.	1.01917	0.	0
GRID	201	0	16.	1.01917	0.	0
GRID	202	0	15.	1.01917	0.	0
GRID	203	0	14.	1.01917	0.	0
GRID	204	0	13.	1.01917	0.	0
GRID	205	0	12.	1.01917	0.	0
GRID	206	0	11.	1.01917	0.	0
GRID	207	0	10.	1.01917	0.	0
GRID	208	0	9.	1.01917	0.	0
GRID	209	0	8.	1.01917	0.	0
GRID	210	0	7.	1.01917	0.	0
GRID	211	0	6.	1.01917	0.	0
GRID	212	0	5.	1.01917	0.	0
GRID	213	0	4.	1.01917	0.	0
GRID	214	0	3.	1.01917	0.	0
GRID	215	0	2.	1.01917	0.	0
GRID	216	0	1.	1.01917	0.	0
GRID	219	0	1.	0.98625	0.	0
GRID	220	0	2.	0.98625	0.	0
GRID	221	0	3.	0.98625	0.	0
GRID	222	0	4.	0.98625	0.	0
GRID	223	0	5.	0.98625	0.	0
GRID	226	0	8.	0.98625	0.	0
GRID	227	0	9.	0.98625	0.	0
GRID	228	0	10.	0.98625	0.	0
GRID	229	0	11.	0.98625	0.	0
GRID	231	0	13.	0.98625	0.	0
GRID	232	0	14.	0.98625	0.	0
GRID	235	0	17.	0.98625	0.	0
GRID	241	0	12.	1.0175	0.	0
GRID	243	0	10.	1.0175	0.	0
GRID	244	0	9.	1.0175	0.	0
GRID	245	0	8.	1.0175	0.	0
GRID	246	0	7.	1.0175	0.	0
GRID	248	0	5.	1.0175	0.	0

TABLE 3-14. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 3 (Continued)

GRID	249	0	4.	1.0175	0.	0
GRID	250	0	3.	1.0175	0.	0
GRID	253	0	0.	0.98625	0.	0
GRID	254	0	0.	0.98458	0.	0
GRID	257	0	3.	0.98458	0.	0
GRID	258	0	4.	0.98458	0.	0
GRID	259	0	5.	0.98458	0.	0
GRID	260	0	6.	0.98458	0.	0
GRID	264	0	10.	0.98458	0.	0
GRID	267	0	13.	0.98458	0.	0
GRID	268	0	14.	0.98458	0.	0
GRID	269	0	15.	0.98458	0.	0
GRID	270	0	16.	0.98458	0.	0
GRID	273	0	16.	0.98625	0.	0
GRID	274	0	15.	0.98625	0.	0
GRID	277	0	12.	0.98625	0.	0
GRID	282	0	7.	0.98625	0.	0
GRID	283	0	6.	0.98625	0.	0
GRID	292	0	2.	0.95542	0.	0
GRID	293	0	3.	0.95542	0.	0
GRID	295	0	5.	0.95542	0.	0
GRID	297	0	7.	0.95542	0.	0
GRID	298	0	8.	0.95542	0.	0
GRID	299	0	9.	0.95542	0.	0
GRID	300	0	10.	0.95542	0.	0
GRID	301	0	11.	0.95542	0.	0
GRID	302	0	12.	0.95542	0.	0
GRID	303	0	13.	0.95542	0.	0
GRID	306	0	16.	0.95542	0.	0
GRID	307	0	17.	0.95542	0.	0
GRID	308	0	17.	0.98458	0.	0
GRID	313	0	12.	0.98458	0.	0
GRID	314	0	11.	0.98458	0.	0
GRID	316	0	9.	0.98458	0.	0
GRID	317	0	8.	0.98458	0.	0
GRID	318	0	7.	0.98458	0.	0
GRID	323	0	2.	0.98458	0.	0
GRID	324	0	1.	0.98458	0.	0
CQUAD4	1	9	1	37	3	36
CQUAD4	2	9	36	3	71	35
CQUAD4	3	9	35	71	5	34
CQUAD4	4	9	34	5	69	33
CQUAD4	5	9	33	69	7	32
CQUAD4	6	9	32	7	67	31
CQUAD4	7	9	31	67	9	30
CQUAD4	8	9	30	9	10	29
CQUAD4	9	9	29	10	11	28
CQUAD4	10	9	28	11	12	27
CQUAD4	11	9	27	12	62	26
CQUAD4	12	9	26	62	14	25
CQUAD4	13	9	25	14	15	24
CQUAD4	14	9	24	15	59	23
CQUAD4	15	9	23	59	58	22
CQUAD4	16	9	22	58	18	21
CQUAD4	17	9	21	18	56	20
CQUAD4	18	8	37	73	108	3
CQUAD4	19	8	3	108	107	71

TABLE 3-14. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 3 (Continued)

CQUAD4	20	8	71	107	41	5
CQUAD4	21	8	5	41	105	69
CQUAD4	22	8	69	105	104	7
CQUAD4	23	8	7	104	44	67
CQUAD4	24	8	67	44	45	9
CQUAD4	25	8	9	45	101	10
CQUAD4	26	8	10	101	47	11
CQUAD4	27	8	11	47	48	12
CQUAD4	28	8	12	48	98	62
CQUAD4	29	8	62	98	50	14
CQUAD4	30	8	14	50	51	15
CQUAD4	31	8	15	51	52	59
CQUAD4	32	8	59	52	53	58
CQUAD4	33	8	58	53	54	18
CQUAD4	34	8	18	54	55	56
CQUAD4	35	7	73	109	75	108
CQUAD4	36	7	108	75	76	107
CQUAD4	37	7	107	76	77	41
CQUAD4	38	7	41	77	78	105
CQUAD4	39	7	105	78	79	104
CQUAD4	40	7	104	79	80	44
CQUAD4	41	7	44	80	138	45
CQUAD4	42	7	45	138	137	101
CQUAD4	43	7	101	137	83	47
CQUAD4	44	7	47	83	135	48
CQUAD4	45	7	48	135	85	98
CQUAD4	46	7	98	85	133	50
CQUAD4	47	7	50	133	87	51
CQUAD4	48	7	51	87	88	52
CQUAD4	49	7	52	88	130	53
CQUAD4	50	7	53	130	129	54
CQUAD4	51	7	54	129	128	55
CQUAD4	52	6	109	145	180	75
CQUAD4	53	6	75	180	112	76
CQUAD4	54	6	76	112	178	77
CQUAD4	55	6	77	178	177	78
CQUAD4	56	6	78	177	115	79
CQUAD4	57	6	79	115	116	80
CQUAD4	58	6	80	116	174	138
CQUAD4	59	6	138	174	118	137
CQUAD4	60	6	137	118	172	83
CQUAD4	61	6	83	172	120	135
CQUAD4	62	6	135	120	121	85
CQUAD4	63	6	85	121	122	133
CQUAD4	64	6	133	122	123	87
CQUAD4	65	6	87	123	124	88
CQUAD4	66	6	88	124	125	130
CQUAD4	67	6	130	125	165	129
CQUAD4	68	6	129	165	164	128
CQUAD4	69	5	145	146	147	180
CQUAD4	70	5	180	147	292	112
CQUAD4	71	5	112	292	293	178
CQUAD4	72	5	178	293	150	177
CQUAD4	73	5	177	150	295	115
CQUAD4	74	5	115	295	152	116
CQUAD4	75	5	116	152	297	174
CQUAD4	76	5	174	297	298	118

TABLE 3-14. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 3 (Continued)

CQUAD4	77	5	118	298	299	172
CQUAD4	78	5	172	299	300	120
CQUAD4	79	5	120	300	301	121
CQUAD4	80	5	121	301	302	122
CQUAD4	81	5	122	302	303	123
CQUAD4	82	5	123	303	160	124
CQUAD4	83	5	124	160	161	125
CQUAD4	84	5	125	161	306	165
CQUAD4	85	5	165	306	307	164
CQUAD4	86	1	182	181	216	183
CQUAD4	87	1	183	216	215	184
CQUAD4	88	1	184	215	214	250
CQUAD4	89	1	250	214	213	249
CQUAD4	90	1	249	213	212	248
CQUAD4	91	1	248	212	211	188
CQUAD4	92	1	188	211	210	246
CQUAD4	93	1	246	210	209	245
CQUAD4	94	1	245	209	208	244
CQUAD4	95	1	244	208	207	243
CQUAD4	96	1	243	207	206	193
CQUAD4	97	1	193	206	205	241
CQUAD4	98	1	241	205	204	195
CQUAD4	99	1	195	204	203	196
CQUAD4	100	1	196	203	202	197
CQUAD4	101	1	197	202	201	198
CQUAD4	102	1	198	201	200	199
CQUAD4	103	2	253	182	183	219
CQUAD4	104	2	219	183	184	220
CQUAD4	105	8	220	184	250	221
CQUAD4	106	2	221	250	249	222
CQUAD4	107	2	222	249	248	223
CQUAD4	108	2	223	248	188	283
CQUAD4	109	2	283	188	246	282
CQUAD4	110	2	282	246	245	226
CQUAD4	111	2	226	245	244	227
CQUAD4	112	2	227	244	243	228
CQUAD4	113	2	228	243	193	229
CQUAD4	114	2	229	193	241	277
CQUAD4	115	2	277	241	195	231
CQUAD4	116	2	231	195	196	232
CQUAD4	117	2	232	196	197	274
CQUAD4	118	2	274	197	198	273
CQUAD4	119	2	273	198	199	235
CQUAD4	120	3	254	253	219	324
CQUAD4	121	3	324	219	220	323
CQUAD4	122	3	323	220	221	257
CQUAD4	123	3	257	221	222	258
CQUAD4	124	3	258	222	223	259
CQUAD4	125	3	259	223	283	260
CQUAD4	126	3	260	283	282	318
CQUAD4	127	3	318	282	226	317
CQUAD4	128	3	317	226	227	316
CQUAD4	129	3	316	227	228	264
CQUAD4	130	3	264	228	229	314
CQUAD4	131	3	314	229	277	313
CQUAD4	132	3	313	277	231	267
CQUAD4	133	3	267	231	232	268

TABLE 3-14. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 3 (Continued)

CQUAD4	134	3	268	232	274	269
CQUAD4	135	3	269	274	273	270
CQUAD4	136	3	270	273	235	308
CQUAD4	137	4	146	254	324	147
CQUAD4	138	4	147	324	323	292
CQUAD4	139	4	292	323	257	293
CQUAD4	140	4	293	257	258	150
CQUAD4	141	4	150	258	259	295
CQUAD4	142	4	295	259	260	152
CQUAD4	143	4	152	260	318	297
CQUAD4	144	4	297	318	317	298
CQUAD4	145	4	298	317	316	299
CQUAD4	146	4	299	316	264	300
CQUAD4	147	4	300	264	314	301
CQUAD4	148	4	301	314	313	302
CQUAD4	149	4	302	313	267	303
CQUAD4	150	4	303	267	268	160
CQUAD4	151	4	160	268	269	161
CQUAD4	152	4	161	269	270	306
CQUAD4	153	4	306	270	308	307
ENDDATA						

The results of the one-half FEM wing model analysis are presented in figure 3-3 in the form of a contour plot. The contours along the outer nodes are in agreement with the imposed boundary conditions, while the contours of the inner layers appear to be indicative of the temperature profile of the THERMOD analysis. Also note the surface boundary conditions enforced on the top and sides of the wing.

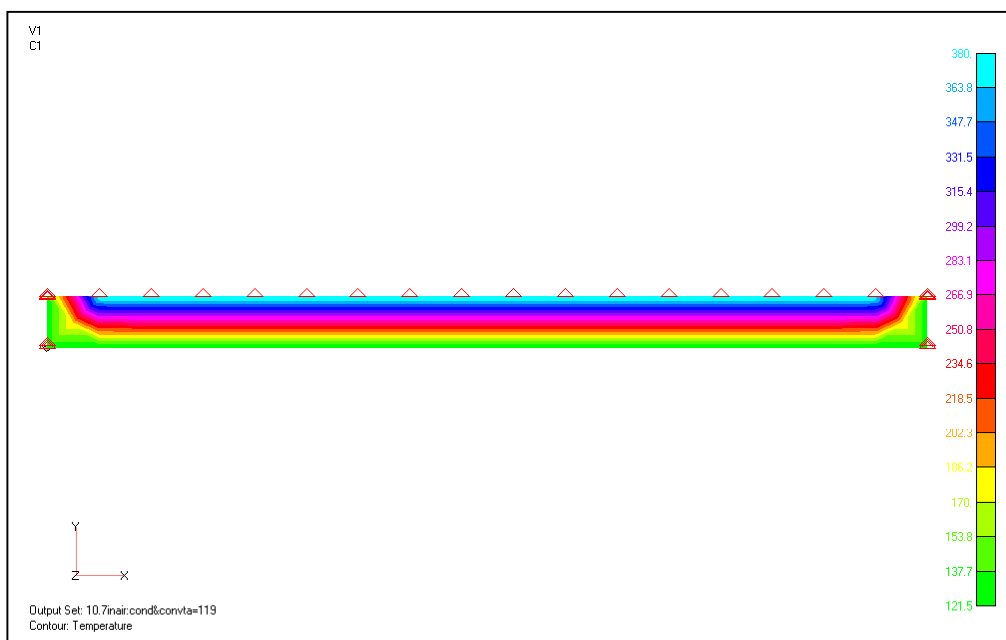


FIGURE 3-3. CONTOUR PLOT OF THE FEM OUTPUT OF SAMPLE PROBLEM 3

The truncated FEM output file is shown in table 3-15. The nodes, from top to bottom of the wing, located at mid-span of the model are indicated by nodes 209, 245, 226, 317, 298, 118, 137, 101, 10, and 29. The temperatures of these nodes were extracted from this file and are reproduced in table 3-16. Mid-span was chosen because it was sufficiently far away from any edge effects that might be present.

TABLE 3-15. TRUNCATED FEM OUTPUT FILE OF SAMPLE PROBLEM 3

1	CONDUCTION & CONVECTION IN RECT. ADIABATIC PLATE (T=0.001FT)					AUGUST 15, 2000	UAI/NASTRAN VERSION 20.1		PAGE
16	10.7 IN AIR GAP; K=0.016BTU/HR/FT/DEGF; H=1.3433BTU/HR/FT^2/DEGF								
0									
	T E M P E R A T U R E V E C T O R								
	POINT ID.	TYPE	ID VALUE	ID+1 VALUE	ID+2 VALUE	ID+3 VALUE	ID+4 VALUE	ID+5 VALUE	
	1	TEMP	1.243670E+02						
	3	TEMP	1.215624E+02						
	5	TEMP	1.222429E+02						
	7	TEMP	1.222786E+02						
	9	TEMP	1.222806E+02	1.222807E+02	1.222807E+02	1.222806E+02			
	14	TEMP	1.222786E+02	1.222896E+02					
	18	TEMP	1.215624E+02						
	20	TEMP	1.243670E+02	1.215388E+02	1.224165E+02	1.222185E+02	1.222653E+02	1.222543E+02	
	26	TEMP	1.222569E+02	1.222563E+02	1.222564E+02	1.222564E+02	1.222563E+02	1.222569E+02	
	32	TEMP	1.222543E+02	1.222653E+02	1.222185E+02	1.224165E+02	1.215388E+02	1.244000E+02	
	41	TEMP	1.287578E+02						
	44	TEMP	1.287905E+02	1.287900E+02					
	47	TEMP	1.287901E+02	1.287900E+02					
	50	TEMP	1.287883E+02	1.287975E+02	1.287578E+02	1.289130E+02	1.278674E+02	1.244000E+02	
	56	TEMP	1.244000E+02						
	58	TEMP	1.224406E+02	1.222429E+02					
	62	TEMP	1.222812E+02						
	67	TEMP	1.222812E+02						
	69	TEMP	1.222896E+02						
	71	TEMP	1.224406E+02						
	73	TEMP	1.244000E+02						
	75	TEMP	1.278909E+02	1.289371E+02	1.287821E+02	1.288218E+02	1.288126E+02	1.288148E+02	
	83	TEMP	1.288144E+02						
	85	TEMP	1.288148E+02						
	87	TEMP	1.288218E+02	1.287821E+02					
	98	TEMP	1.287905E+02						
	101	TEMP	1.287901E+02						
	104	TEMP	1.287883E+02	1.287975E+02					
	107	TEMP	1.289130E+02	1.278674E+02	1.244000E+02				
	112	TEMP	1.293606E+02						
	115	TEMP	1.292379E+02	1.292401E+02					
	118	TEMP	1.292397E+02						
	120	TEMP	1.292395E+02	1.292401E+02	1.292379E+02	1.292470E+02	1.292076E+02	1.293606E+02	
	128	TEMP	1.244000E+02	1.278909E+02	1.289371E+02				
	133	TEMP	1.288126E+02						
	135	TEMP	1.288143E+02						
	137	TEMP	1.288144E+02	1.288143E+02					
	145	TEMP	1.244000E+02	1.244000E+02	3.669625E+02				
	150	TEMP	3.730168E+02						
	152	TEMP	3.730167E+02						
	160	TEMP	3.730125E+02	3.728751E+02					
	164	TEMP	1.244000E+02	1.283041E+02					
	172	TEMP	1.292397E+02						
	174	TEMP	1.292395E+02						
	177	TEMP	1.292470E+02	1.292076E+02					
	180	TEMP	1.283041E+02	1.244000E+02	1.244000E+02	3.799533E+02	3.799752E+02		
	188	TEMP	3.799756E+02						
	193	TEMP	3.799756E+02						
	195	TEMP	3.799756E+02	3.799756E+02	3.799752E+02	3.799533E+02	1.244000E+02	1.244000E+02	
	201	TEMP	3.800000E+02	3.800000E+02	3.800000E+02	3.800000E+02	3.800000E+02	3.800000E+02	
1	CONDUCTION & CONVECTION IN RECT. ADIABATIC PLATE (T=0.001FT)					AUGUST 15, 2000	UAI/NASTRAN VERSION 20.1		PAGE
17	10.7 IN AIR GAP; K=0.016BTU/HR/FT/DEGF; H=1.3433BTU/HR/FT^2/DEGF								
0									
	T E M P E R A T U R E V E C T O R								
	POINT ID.	TYPE	ID VALUE	ID+1 VALUE	ID+2 VALUE	ID+3 VALUE	ID+4 VALUE	ID+5 VALUE	
	207	TEMP	3.800000E+02	3.800000E+02	3.800000E+02	3.800000E+02	3.800000E+02	3.800000E+02	
	213	TEMP	3.800000E+02	3.800000E+02	3.800000E+02	3.800000E+02			
	219	TEMP	3.676886E+02	3.733311E+02	3.734622E+02	3.734663E+02	3.734662E+02		
	226	TEMP	3.734662E+02	3.734662E+02	3.734662E+02	3.734663E+02			
	231	TEMP	3.734663E+02	3.734622E+02					
	235	TEMP	1.244000E+02						
	241	TEMP	3.799756E+02						
	243	TEMP	3.799756E+02	3.799756E+02	3.799756E+02	3.799756E+02			
	248	TEMP	3.799756E+02	3.799756E+02	3.799756E+02				
	253	TEMP	1.244000E+02	1.244000E+02					
	257	TEMP	3.734379E+02	3.734420E+02	3.734419E+02	3.734419E+02			
	264	TEMP	3.734419E+02						

TABLE 3-15. TRUNCATED FEM OUTPUT FILE OF SAMPLE PROBLEM 3 (Continued)

267	TEMP	3.734420E+02	3.734379E+02	3.733062E+02	3.676435E+02				
273	TEMP	3.676886E+02	3.733311E+02						
277	TEMP	3.734662E+02							
282	TEMP	3.734602E+02	3.734663E+02						
292	TEMP	3.728751E+02	3.730125E+02						
295	TEMP	3.730167E+02							
297	TEMP	3.730167E+02	3.730167E+02	3.730167E+02	3.730167E+02	3.730167E+02	3.730167E+02		
303	TEMP	3.730168E+02							
306	TEMP	3.669625E+02	1.244000E+02	1.244000E+02					
313	TEMP	3.734419E+02	3.734419E+02						
316	TEMP	3.734419E+02	3.734419E+02	3.734419E+02					
323	TEMP	3.733062E+02	3.676435E+02						
500	TEMP	1.190000E+02							

When compared, the temperatures of tables 3-13 and 3-16 show excellent agreement, validating the conduction and convection capabilities of the THERMOD program.

TABLE 3-16. FINITE ELEMENT METHOD-SIMULATED TEMPERATURES (°F) OF THE RIGHT WING OF SAMPLE PROBLEM 3

T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
380.0	380.0	373.5	373.4	373.0	129.2	128.8	128.8	122.3	122.3

3.4 SAMPLE PROBLEM 4.

The primary purpose of sample problem 4 was to study the effects of solar radiation on an aircraft. As in sample problems 1 and 2, THERMOD was set up to solve a complete aircraft problem, while the FEM method was confined to the wing model.

An aircraft parked in the open is subjected to three types of radiation: direct solar radiation, diffused solar radiation, and infrared radiation.

- The direct solar radiation rays are in the visible regions of the energy spectrum, have a fixed incident angle, are of short wavelengths, and are characterized by high energy levels.
- The diffused solar radiation is also in the visible region of the energy spectrum but does not have a fixed incident angle. It is diffused in nature.
- The infrared radiation falls in the nonvisible region of the energy spectrum. It is of long wavelength and characterized by low energy level.
- The direct solar radiation type is explored in sample problem 4.

In addition, all external surfaces, except the wing surface, were assigned absorptivity and emissivity values of 1.0 and 0.0, respectively. The values for the wing surface were 0.9 and 0.0, respectively. A 0.0 emissivity value prevents the emission of infrared energy, thereby removing its influence on the thermal system. For the same reason, the sky radiation was assigned an absolute temperature of 0.0. The conduction and convection heat transfer modes, however, were left intact, as in sample problem 3. The THERMOD input file, input.dat, is illustrated in table 3-17.

TABLE 3-17. THERMOD INPUT FILE (input.dat) OF SAMPLE PROBLEM 4

```

198e-6,0.704
1, 0.020,125,0.300,0.30
2, 0.375,4.4,0.021,0.24
3, 0.020,125,0.300,0.30
4, 0.350,125,0.300,0.30
5, 10.70,0.067,0.016,0.24
6, 0.350,125,0.300,0.30
7, 0.020,125,0.300,0.30
8, 0.375,4.4,0.021,0.24
9, 0.020,125,0.300,0.30
10,0.030,125,0.300,0.30
11,0.375,4.4,0.021,0.24
12,0.020,125,0.300,0.30
13,0.250,10.,0.020,0.24
14,48.00,0.067,0.016,0.24
15,0.020,125,0.300,0.30
16,0.375,4.4,0.021,0.24
17,0.020,125,0.300,0.30
18,0.250,10.,0.020,0.24
19,30.00,0.067,0.016,0.24
20,5.000,10.,0.020,0.24
21,0.020,125,0.300,0.30
22,0.375,4.4,0.021,0.24
23,0.020,125,0.300,0.30
24,2.500,0.067,0.016,0.24
25,0.020,125,0.300,0.30
26,0.375,4.4,0.021,0.24
27,0.020,125,0.300,0.30
28,0.250,130,0.800,0.20
3.5,34.0,2.5,5.5,22.0,11.0,3.0,4.0,30,24,28
0.70,1.0,0.0,2.0,150,0.80,0.22
0.3,0.9,0.3,0.9,0.3,0.9,0.1,0.9
3,100
0.9,0.9,0.0,0.0
1.0,0.0,0.9,0.0,1.0,0.0,1.0,0.0
9
0,10,10,0,0,75,110,110,190,190
5,120,5,60,20,5,120,60,60
7
60 570.67 450.00 14 330 10 10 0.36
75 573.67 0.0000 14 355 10 10 0.31
90 578.67 0.0000 14 355 10 12 0.33
75 581.67 0.0000 14 330 10 10 0.31
60 582.67 0.0000 14 291 10 10 0.36
45 583.67 0.0000 14 231 10 10 0.44
30 582.67 0.0000 14 160 10 10 0.57
  
```

TABLE 3-17. THERMOD INPUT FILE (input.dat) OF SAMPLE PROBLEM 4 (Continued)

```
2,100,180,1
10,1.0,1.0
10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,
10,10,10,10,10,10,10,10,10,10,10,10,10,10,
10,10,10,10,10,10,10,10,10,10,10,10,10,10,
10,10,10,10,10,10,10,10,10,10,10,10,10,10,
10,10,10,10,10,10,10,10,10,10,10,10,10,10,
```

Temperatures from the THERMOD run is shown in table 3-18, from which the steady-state temperatures T1, T2.....T10 (top to bottom of wing) for time period 3 were extracted and tabulated in table 3-19.

TABLE 3-18. THERMOD OUTPUT FILE (summary.dat) FOR SAMPLE PROBLEM 4

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
THERMOD:  A THERMAL PROGRAM FOR AIRCRAFTS
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

SUMMARY OF STEADY-STATE AND TRANSIENT ANALYSIS

(Temperatures are shown for all time periods)

TEMPERATURES AT THE END OF STEADY STATE ANALYSIS

      TEMPERATURES IN DEGREES FAHRENHEIT
      1      2      3      4      5      6      7

1  300.1  340.9  353.9  332.9  289.8  232.1  175.9
2  300.1  340.9  353.9  332.9  289.7  232.1  175.9
3  295.4  335.2  348.0  327.6  285.6  229.4  174.6
4  295.4  335.2  348.0  327.6  285.6  229.4  174.6
5  295.1  334.8  347.6  327.3  285.3  229.2  174.5
6  118.4  122.9  128.2  130.3  129.5  128.2  125.1
7  118.1  122.5  127.8  129.9  129.3  128.1  125.0
8  118.1  122.5  127.8  129.9  129.3  128.1  125.0
9  113.4  116.9  122.0  124.6  125.1  125.4  123.7
10 113.4  116.8  121.9  124.6  125.1  125.3  123.7
11 300.1  340.9  353.9  332.9  289.8  232.1  175.9
12 300.1  340.9  353.9  332.9  289.7  232.1  175.9
13 295.4  335.2  348.0  327.6  285.6  229.4  174.6
14 295.4  335.2  348.0  327.6  285.6  229.4  174.6
```

TABLE 3-18. THERMOD OUTPUT FILE (summary.dat) FOR SAMPLE
 PROBLEM 4 (Continued)

15	295.1	334.8	347.6	327.3	285.3	229.2	174.5
16	118.4	122.9	128.2	130.3	129.5	128.2	125.1
17	118.1	122.5	127.8	129.9	129.3	128.1	125.0
18	118.1	122.5	127.8	129.9	129.3	128.1	125.0
19	113.4	116.9	122.0	124.6	125.1	125.4	123.7
20	113.4	116.8	121.9	124.6	125.1	125.3	123.7
21	277.3	306.8	316.0	300.7	267.3	220.2	169.5
22	277.2	306.6	315.8	300.5	267.2	220.2	169.5
23	244.7	251.8	252.7	248.1	232.9	204.4	160.4
24	244.6	251.6	252.5	247.9	232.8	204.3	160.4
25	221.9	213.2	208.4	211.3	208.8	193.3	154.0
26	194.0	172.3	166.4	172.1	175.6	167.8	132.5
27	147.9	139.9	140.0	144.2	146.4	143.5	127.2
28	147.9	139.9	140.0	144.2	146.4	143.4	127.2
29	144.6	137.6	138.2	142.2	144.3	141.7	126.8
30	144.6	137.5	138.2	142.2	144.3	141.7	126.8
31	130.1	127.4	129.9	133.5	135.1	134.1	125.2
32	130.1	127.4	129.9	133.5	135.1	134.1	125.2
33	130.1	127.4	129.9	133.5	135.1	134.1	125.2
34	130.1	127.4	129.9	133.5	135.1	134.1	125.2
35	115.7	117.3	121.7	124.8	126.0	126.5	123.5
36	115.7	117.3	121.7	124.8	126.0	126.5	123.5
37	112.4	115.0	119.8	122.8	123.9	124.7	123.2
38	112.4	115.0	119.8	122.8	123.9	124.7	123.2
39	219.3	175.9	125.8	178.9	215.2	225.3	205.4
40	219.2	175.9	125.9	178.9	215.1	225.2	205.2
41	205.0	171.8	142.0	173.4	194.6	197.1	173.1
42	204.9	171.8	142.1	173.4	194.5	197.0	173.0
43	194.9	168.9	153.4	169.6	180.1	177.3	150.4
44	186.3	163.3	153.4	164.4	172.1	169.1	143.4
45	161.6	147.1	142.1	150.5	156.0	154.3	136.7
46	161.4	147.0	142.0	150.4	155.9	154.2	136.7
47	126.1	123.9	125.9	130.5	132.9	133.1	127.1
48	125.9	123.8	125.8	130.4	132.8	133.0	127.1
49	124.3	92.5	91.1	95.6	98.2	95.5	81.8
50	126.8	95.1	93.4	98.1	100.9	97.9	83.4
51	188.8	207.4	215.7	208.8	191.6	168.5	144.8
52	183.9	201.5	209.6	203.3	187.3	165.7	143.4
53	188.8	207.4	215.7	208.8	191.6	168.5	144.8
Maximum Structural Temperature for Period 1 = 300.11 Occurring at Location 1							
Maximum Structural Temperature for Period 2 = 340.91 Occurring at Location 1							
Maximum Structural Temperature for Period 3 = 353.91 Occurring at Location 1							

TABLE 3-18. THERMOD OUTPUT FILE (summary.dat) FOR SAMPLE
 PROBLEM 4 (Continued)

Maximum Structural Temperature for Period	4	=	332.93	Occurring			
at Location	1						
Maximum Structural Temperature for Period	5	=	289.76	Occurring			
at Location	1						
Maximum Structural Temperature for Period	6	=	232.09	Occurring			
at Location	1						
Maximum Structural Temperature for Period	7	=	205.42	Occurring			
at Location	39						
Maximum Structural Temperature Over All	7	Periods	=353.91				
Occurring at Location	1	at Period	3				
TEMPERATURES AT THE END OF TRANSIENT ANALYSIS							
TEMPERATURES IN DEGREES FAHRENHEIT							
	1	2	3	4	5	6	7
1	127.3	133.5	139.2	140.1	137.3	133.3	127.6
2	132.4	139.6	145.5	145.8	141.8	136.2	129.0
3	282.6	319.9	332.1	313.4	274.3	222.1	171.0
4	286.9	325.1	337.5	318.2	278.1	224.5	172.2
5	286.6	324.7	337.2	317.9	277.9	224.4	172.2
6	122.7	128.1	133.6	135.1	133.4	130.7	126.3
7	118.0	122.4	127.7	129.8	129.2	128.0	125.0
8	117.8	122.2	127.5	129.6	129.0	127.9	124.9
9	111.4	114.5	119.5	122.4	123.4	124.2	123.1
10	111.2	114.2	119.3	122.2	123.2	124.1	123.1
11	127.3	133.5	139.2	140.1	137.3	133.3	127.6
12	132.4	139.6	145.5	145.8	141.8	136.2	129.0
13	282.6	319.9	332.1	313.4	274.3	222.1	171.0
14	286.9	325.1	337.5	318.2	278.1	224.5	172.2
15	286.6	324.7	337.2	317.9	277.9	224.4	172.2
16	122.7	128.1	133.6	135.1	133.4	130.7	126.3
17	118.0	122.4	127.7	129.8	129.2	128.0	125.0
18	117.8	122.2	127.5	129.6	129.0	127.9	124.9
19	111.4	114.5	119.5	122.4	123.4	124.2	123.1
20	111.2	114.2	119.3	122.2	123.2	124.1	123.1
21	130.8	137.1	142.6	143.4	140.3	135.5	128.5
22	133.6	139.9	145.2	145.9	142.5	137.2	129.2
23	204.0	207.7	207.2	206.6	198.8	180.2	146.1
24	205.9	209.0	208.1	207.7	200.0	181.2	146.3
25	210.6	196.6	189.5	193.7	192.8	177.7	137.0
26	186.0	169.5	163.5	169.5	173.2	165.7	132.1
27	148.1	140.0	140.1	144.3	146.5	143.5	127.2
28	147.0	139.3	139.6	143.7	145.8	143.0	127.1
29	144.2	137.3	138.0	142.1	144.1	141.6	126.8

TABLE 3-18. THERMOD OUTPUT FILE (summary.dat) FOR SAMPLE
 PROBLEM 4 (Continued)

30	143.4	136.8	137.5	141.6	143.6	141.1	126.7
31	129.7	127.1	129.7	133.3	134.9	133.9	125.1
32	129.7	127.1	129.7	133.3	134.9	133.9	125.1
33	129.7	127.1	129.7	133.3	134.9	133.9	125.1
34	129.7	127.1	129.7	133.3	134.9	133.9	125.1
35	116.0	117.5	121.9	125.0	126.2	126.6	123.6
36	115.2	116.9	121.4	124.5	125.6	126.2	123.5
37	111.3	114.2	119.2	122.2	123.2	124.2	123.0
38	111.2	114.1	119.1	122.1	123.1	124.1	123.0
39	119.9	119.2	119.9	126.8	130.5	132.1	129.4
40	122.1	120.6	120.6	128.0	132.2	133.8	130.5
41	176.5	157.0	140.1	159.8	172.7	173.0	153.5
42	178.1	158.2	141.0	160.8	173.8	174.0	153.9
43	186.5	166.5	152.0	167.4	177.3	174.3	148.1
44	178.8	161.5	152.0	162.8	170.1	166.9	141.7
45	159.6	147.2	141.0	150.7	157.1	155.7	138.3
46	158.0	146.1	140.1	149.7	156.0	154.7	137.9
47	119.5	119.2	120.6	126.7	129.9	131.3	128.4
48	118.1	118.2	119.9	125.8	128.9	130.4	127.9
49	116.2	106.1	108.7	112.2	113.7	113.2	107.3
50	119.5	107.0	108.8	112.6	114.3	113.3	105.7
51	306.1	348.1	361.3	339.6	295.0	235.5	177.6
52	288.9	327.5	340.0	320.5	279.9	225.7	172.8
53	306.1	348.1	361.3	339.6	295.0	235.5	177.6
Maximum Structural Temperature for Period 1 = 286.91 Occurring at Location 4							
Maximum Structural Temperature for Period 2 = 325.06 Occurring at Location 4							
Maximum Structural Temperature for Period 3 = 337.51 Occurring at Location 4							
Maximum Structural Temperature for Period 4 = 318.20 Occurring at Location 4							
Maximum Structural Temperature for Period 5 = 278.12 Occurring at Location 4							
Maximum Structural Temperature for Period 6 = 224.54 Occurring at Location 4							
Maximum Structural Temperature for Period 7 = 172.24 Occurring at Location 4							
Maximum Structural Temperature Over All 7 Periods = 337.51 Occurring at Location 4 at Period 3							

TABLE 3-19. THERMOD-SIMULATED TEMPERATURES (°F) OF THE RIGHT WING OF
SAMPLE PROBLEM 4

T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
353.9	353.9	348.0	348.0	347.6	128.2	127.8	127.8	122.0	121.9

As in sample problems 1 through 3, a one-half wing structure was modeled using the UAI/NASTRAN FEM program. Two types of loads were applied to the model. These loads, based on period 3 of the THERMOD input file, were direct solar radiation of 355 Btu/hr/ft² and convection with a film coefficient of $h = 1.3433 \text{ (Btu/hr)/(ft}^2\text{°F)}$. The film coefficient was calculated internally by the THERMOD program and was based on the theory of airflow on a flat surface and a wind speed of 14 ft/sec. The radiation load was applied perpendicular to the top surface only, while both the top and bottom surfaces were subjected to convection forces. The ambient temperature was maintained at 119°F (578.67°R).

The FEM input file is shown in table 3-20. The ambient temperature of 119°F was imposed through the SPC case control card. The ambient temperature boundary condition was scalar in nature and implemented through the SPOINT bulk data. The boundary elements, CHBDY, through which solar radiation and convection effects were applied, were modeled using the LINE boundary shape type, connecting the boundary nodes along the wing top and wing bottom. The SPACE designation in the CHBDY boundary element indicates that this element emits infrared radiation to a black body, in which case, no VIEW bulk data were required. Property cards PHBDY and the material cards MAT4, define absorptivity, emissivity, film coefficient, and conductivity values. Because of the involvement of infrared radiation, through emission, this sample problem was rendered nonlinear. This required the Stefan-Boltzman constant and an initial set of temperatures, which were implemented through the PARAM and TEMPERATURE statements, respectively.

TABLE 3-20. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE PROBLEM 4

```
ID D:\Natha, MSC/N
APP HEAT
SOL 3
TIME 10000
CEND
  TITLE = direct_radiation_conduction_noemission_convection (t=0.001ft)
  SUBTITLE = 10.7 in gap filled by air; k=0.016BTU/hr/ft/degF
    load=1
    SPC = 1
    temperature(estimate)=2
    THERMAL= ALL
    FLUX = ALL
    SPCF = ALL
BEGIN BULK
$ *****
$   Written by : MSC/NASTRAN for Windows
$   Version   : 6.00
$   Translator : UAI/NASTRAN
$   From Model : D:\Thermod_Validation\FEM\conduction_only.MOD
$   Date      : Mon Mar 06 11:27:09 2000
$ *****
```

TABLE 3-20. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 4 (Continued)

```

$
PARAM,K6ROT,100.
PARAM,MAXRATIO,1.E+8
param,sigma,1.714e-9
param,tabs,0.0
param,maxit,10
CORD2C      1      0      0.      0.      0.      0.      0.      1.+MSC/NC1
+MSC/NC1    1.      0.      1.
CORD2S      2      0      0.      0.      0.      0.      0.      1.+MSC/NC2
+MSC/NC2    1.      0.      1.
$
$ MSC/NASTRAN for Windows Property 1 : 0.02 in glass_outer_layer
PSHELL      1      1      0.001      1      1      0.
$ MSC/NASTRAN for Windows Property 2 : 3/8 in foam
PSHELL      2      2      0.001      2      2      0.
$ MSC/NASTRAN for Windows Property 3 : 0.02 in inner glass
PSHELL      3      1      0.001      1      1      0.
$ MSC/NASTRAN for Windows Property 4 : 0.350 in cap
PSHELL      4      1      0.001      1      1      0.
$ MSC/NASTRAN for Windows Property 5 : 10.70 in air space
PSHELL      5      3      0.001      3      3      0.
$ MSC/NASTRAN for Windows Property 6 : 0.350 in cap
PSHELL      6      1      0.001      1      1      0.
$ MSC/NASTRAN for Windows Property 7 : 0.02 in inner glass
PSHELL      7      1      0.001      1      1      0.
$ MSC/NASTRAN for Windows Property 8 : 3/8 in foam
PSHELL      8      2      0.001      2      2      0.
$ MSC/NASTRAN for Windows Property 9 : 0.02 in glass_outer_layer
PSHELL      9      1      0.001      1      1      0.
$ MSC/NASTRAN for Windows Material 1 : glass
MAT4        1      0.3      1.
$ MSC/NASTRAN for Windows Material 2 : foam
MAT4        2      0.021      1.
$ MSC/NASTRAN for Windows Material 3 : air
MAT4        3      0.016      1.
$
chbdy,1001,100,line,200,201,,,space,500,500,,,0.0,1.0,0.0
chbdy,1002,100,line,201,202,,,space,500,500,,,0.0,1.0,0.0
chbdy,1003,100,line,202,203,,,space,500,500,,,0.0,1.0,0.0
chbdy,1004,100,line,203,204,,,space,500,500,,,0.0,1.0,0.0
chbdy,1005,100,line,204,205,,,space,500,500,,,0.0,1.0,0.0
chbdy,1006,100,line,205,206,,,space,500,500,,,0.0,1.0,0.0
chbdy,1007,100,line,206,207,,,space,500,500,,,0.0,1.0,0.0
chbdy,1008,100,line,207,208,,,space,500,500,,,0.0,1.0,0.0
chbdy,1009,100,line,208,209,,,space,500,500,,,0.0,1.0,0.0
chbdy,1010,100,line,209,210,,,space,500,500,,,0.0,1.0,0.0
chbdy,1011,100,line,210,211,,,space,500,500,,,0.0,1.0,0.0
chbdy,1012,100,line,211,212,,,space,500,500,,,0.0,1.0,0.0
chbdy,1013,100,line,212,213,,,space,500,500,,,0.0,1.0,0.0
chbdy,1014,100,line,213,214,,,space,500,500,,,0.0,1.0,0.0
chbdy,1015,100,line,214,215,,,space,500,500,,,0.0,1.0,0.0
chbdy,1016,100,line,215,216,,,space,500,500,,,0.0,1.0,0.0
chbdy,1017,100,line,216,181,,,space,500,500,,,0.0,1.0,0.0
phbdy,100,200,0.001,0.0,0.9
mat4,200,1.3433
spoint,500
spc,1,500,1,578.67

```

TABLE 3-20. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 4 (Continued)

```
$
$ Direct radiation load from a distant source: load case 1
qvect,1,355.0,0.0,-1.0,0.0,1001,thru,1017
$
$ view ids
$view,1,yes,yes,1,1,0.0
$
chbdy,2001,300,line,20,21,,space,500,500,,0.0,-1.0,0.0
chbdy,2002,300,line,21,22,,space,500,500,,0.0,-1.0,0.0
chbdy,2003,300,line,22,23,,space,500,500,,0.0,-1.0,0.0
chbdy,2004,300,line,23,24,,space,500,500,,0.0,-1.0,0.0
chbdy,2005,300,line,24,25,,space,500,500,,0.0,-1.0,0.0
chbdy,2006,300,line,25,26,,space,500,500,,0.0,-1.0,0.0
chbdy,2007,300,line,26,27,,space,500,500,,0.0,-1.0,0.0
chbdy,2008,300,line,27,28,,space,500,500,,0.0,-1.0,0.0
chbdy,2009,300,line,28,29,,space,500,500,,0.0,-1.0,0.0
chbdy,2010,300,line,29,30,,space,500,500,,0.0,-1.0,0.0
chbdy,2011,300,line,30,31,,space,500,500,,0.0,-1.0,0.0
chbdy,2012,300,line,31,32,,space,500,500,,0.0,-1.0,0.0
chbdy,2013,300,line,32,33,,space,500,500,,0.0,-1.0,0.0
chbdy,2014,300,line,33,34,,space,500,500,,0.0,-1.0,0.0
chbdy,2015,300,line,34,35,,space,500,500,,0.0,-1.0,0.0
chbdy,2016,300,line,35,36,,space,500,500,,0.0,-1.0,0.0
chbdy,2017,300,line,36,1,,space,500,500,,0.0,-1.0,0.0
phbdy,300,400,0.001,0.0,0.9
mat4,400,1.3433
$
temp,2,1,578.67,3,578.67,5,578.67
temp,2,7,578.67,9,578.67,10,578.67
temp,2,11,578.67,12,578.67,14,578.67
temp,2,15,578.67,18,578.67,20,578.67
temp,2,21,578.67,22,578.67,23,578.67
temp,2,24,578.67,25,578.67,26,578.67
temp,2,27,578.67,28,578.67,29,578.67
temp,2,30,578.67,31,578.67,32,578.67
temp,2,33,578.67,34,578.67,35,578.67
temp,2,36,578.67,37,578.67,41,578.67
temp,2,44,578.67,45,578.67,47,578.67
temp,2,48,578.67,50,578.67,51,578.67
temp,2,52,578.67,53,578.67,54,578.67
temp,2,55,578.67,56,578.67,58,578.67
temp,2,59,578.67,62,578.67,67,578.67
temp,2,69,578.67,71,578.67,73,578.67
temp,2,75,578.67,76,578.67,77,578.67
temp,2,78,578.67,79,578.67,80,578.67
temp,2,83,578.67,85,578.67,87,578.67
temp,2,88,578.67,98,578.67,101,578.67
temp,2,104,578.67,105,578.67,107,578.67
temp,2,108,578.67,109,578.67,112,578.67
temp,2,115,578.67,116,578.67,118,578.67
temp,2,120,578.67,121,578.67,122,578.67
temp,2,123,578.67,124,578.67,125,578.67
temp,2,128,578.67,129,578.67,130,578.67
temp,2,133,578.67,135,578.67,137,578.67
temp,2,138,578.67,145,578.67,146,578.67
temp,2,147,578.67,150,578.67,152,578.67
temp,2,160,578.67,161,578.67,164,578.67
```

TABLE 3-20. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 4 (Continued)

temp, 2, 165, 578.67, 172, 578.67, 174, 578.67					
temp, 2, 177, 578.67, 178, 578.67, 180, 578.67					
temp, 2, 181, 578.67, 182, 578.67, 183, 578.67					
temp, 2, 184, 578.67, 188, 578.67, 193, 578.67					
temp, 2, 195, 578.67, 196, 578.67, 197, 578.67					
temp, 2, 198, 578.67, 199, 578.67, 200, 578.67					
temp, 2, 201, 578.67, 202, 578.67, 203, 578.67					
temp, 2, 204, 578.67, 205, 578.67, 206, 578.67					
temp, 2, 207, 578.67, 208, 578.67, 209, 578.67					
temp, 2, 210, 578.67, 211, 578.67, 212, 578.67					
temp, 2, 213, 578.67, 214, 578.67, 215, 578.67					
temp, 2, 216, 578.67, 219, 578.67, 220, 578.67					
temp, 2, 221, 578.67, 222, 578.67, 223, 578.67					
temp, 2, 226, 578.67, 227, 578.67, 228, 578.67					
temp, 2, 229, 578.67, 231, 578.67, 232, 578.67					
temp, 2, 235, 578.67, 241, 578.67, 243, 578.67					
temp, 2, 244, 578.67, 245, 578.67, 246, 578.67					
temp, 2, 248, 578.67, 249, 578.67, 250, 578.67					
temp, 2, 253, 578.67, 254, 578.67, 257, 578.67					
temp, 2, 258, 578.67, 259, 578.67, 260, 578.67					
temp, 2, 264, 578.67, 267, 578.67, 268, 578.67					
temp, 2, 269, 578.67, 270, 578.67, 273, 578.67					
temp, 2, 274, 578.67, 277, 578.67, 282, 578.67					
temp, 2, 283, 578.67, 292, 578.67, 293, 578.67					
temp, 2, 295, 578.67, 297, 578.67, 298, 578.67					
temp, 2, 299, 578.67, 300, 578.67, 301, 578.67					
temp, 2, 302, 578.67, 303, 578.67, 306, 578.67					
temp, 2, 307, 578.67, 308, 578.67, 313, 578.67					
temp, 2, 314, 578.67, 316, 578.67, 317, 578.67					
temp, 2, 318, 578.67, 323, 578.67, 324, 578.67					
temp, 2, 500, 578.67					
\$					
GRID	1	0	0.	0.	0.
GRID	3	0	1.1.667E-3	0.	0
GRID	5	0	3.1.667E-3	0.	0
GRID	7	0	5.1.667E-3	0.	0
GRID	9	0	7.1.667E-3	0.	0
GRID	10	0	8.1.667E-3	0.	0
GRID	11	0	9.1.667E-3	0.	0
GRID	12	0	10.1.667E-3	0.	0
GRID	14	0	12.1.667E-3	0.	0
GRID	15	0	13.1.667E-3	0.	0
GRID	18	0	16.1.667E-3	0.	0
GRID	20	0	17.	0.	0
GRID	21	0	16.	0.	0
GRID	22	0	15.	0.	0
GRID	23	0	14.	0.	0
GRID	24	0	13.	0.	0
GRID	25	0	12.	0.	0
GRID	26	0	11.	0.	0
GRID	27	0	10.	0.	0
GRID	28	0	9.	0.	0
GRID	29	0	8.	0.	0
GRID	30	0	7.	0.	0
GRID	31	0	6.	0.	0
GRID	32	0	5.	0.	0
GRID	33	0	4.	0.	0

TABLE 3-20. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 4 (Continued)

GRID	34	0	3.	0.	0.	0
GRID	35	0	2.	0.	0.	0
GRID	36	0	1.	0.	0.	0
GRID	37	0	0.1.667E-3	0.	0.	0
GRID	41	0	3.0.032917	0.	0.	0
GRID	44	0	6.0.032917	0.	0.	0
GRID	45	0	7.0.032917	0.	0.	0
GRID	47	0	9.0.032917	0.	0.	0
GRID	48	0	10.0.032917	0.	0.	0
GRID	50	0	12.0.032917	0.	0.	0
GRID	51	0	13.0.032917	0.	0.	0
GRID	52	0	14.0.032917	0.	0.	0
GRID	53	0	15.0.032917	0.	0.	0
GRID	54	0	16.0.032917	0.	0.	0
GRID	55	0	17.0.032917	0.	0.	0
GRID	56	0	17.1.667E-3	0.	0.	0
GRID	58	0	15.1.667E-3	0.	0.	0
GRID	59	0	14.1.667E-3	0.	0.	0
GRID	62	0	11.1.667E-3	0.	0.	0
GRID	67	0	6.1.667E-3	0.	0.	0
GRID	69	0	4.1.667E-3	0.	0.	0
GRID	71	0	2.1.667E-3	0.	0.	0
GRID	73	0	0.0.032917	0.	0.	0
GRID	75	0	1.0.034583	0.	0.	0
GRID	76	0	2.0.034583	0.	0.	0
GRID	77	0	3.0.034583	0.	0.	0
GRID	78	0	4.0.034583	0.	0.	0
GRID	79	0	5.0.034583	0.	0.	0
GRID	80	0	6.0.034583	0.	0.	0
GRID	83	0	9.0.034583	0.	0.	0
GRID	85	0	11.0.034583	0.	0.	0
GRID	87	0	13.0.034583	0.	0.	0
GRID	88	0	14.0.034583	0.	0.	0
GRID	98	0	11.0.032917	0.	0.	0
GRID	101	0	8.0.032917	0.	0.	0
GRID	104	0	5.0.032917	0.	0.	0
GRID	105	0	4.0.032917	0.	0.	0
GRID	107	0	2.0.032917	0.	0.	0
GRID	108	0	1.0.032917	0.	0.	0
GRID	109	0	0.0.034583	0.	0.	0
GRID	112	0	2. 0.06375	0.	0.	0
GRID	115	0	5. 0.06375	0.	0.	0
GRID	116	0	6. 0.06375	0.	0.	0
GRID	118	0	8. 0.06375	0.	0.	0
GRID	120	0	10. 0.06375	0.	0.	0
GRID	121	0	11. 0.06375	0.	0.	0
GRID	122	0	12. 0.06375	0.	0.	0
GRID	123	0	13. 0.06375	0.	0.	0
GRID	124	0	14. 0.06375	0.	0.	0
GRID	125	0	15. 0.06375	0.	0.	0
GRID	128	0	17.0.034583	0.	0.	0
GRID	129	0	16.0.034583	0.	0.	0
GRID	130	0	15.0.034583	0.	0.	0
GRID	133	0	12.0.034583	0.	0.	0
GRID	135	0	10.0.034583	0.	0.	0
GRID	137	0	8.0.034583	0.	0.	0
GRID	138	0	7.0.034583	0.	0.	0

TABLE 3-20. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 4 (Continued)

GRID	145	0	0.	0.06375	0.	0
GRID	146	0	0.	0.95542	0.	0
GRID	147	0	1.	0.95542	0.	0
GRID	150	0	4.	0.95542	0.	0
GRID	152	0	6.	0.95542	0.	0
GRID	160	0	14.	0.95542	0.	0
GRID	161	0	15.	0.95542	0.	0
GRID	164	0	17.	0.06375	0.	0
GRID	165	0	16.	0.06375	0.	0
GRID	172	0	9.	0.06375	0.	0
GRID	174	0	7.	0.06375	0.	0
GRID	177	0	4.	0.06375	0.	0
GRID	178	0	3.	0.06375	0.	0
GRID	180	0	1.	0.06375	0.	0
GRID	181	0	0.	1.01917	0.	0
GRID	182	0	0.	1.0175	0.	0
GRID	183	0	1.	1.0175	0.	0
GRID	184	0	2.	1.0175	0.	0
GRID	188	0	6.	1.0175	0.	0
GRID	193	0	11.	1.0175	0.	0
GRID	195	0	13.	1.0175	0.	0
GRID	196	0	14.	1.0175	0.	0
GRID	197	0	15.	1.0175	0.	0
GRID	198	0	16.	1.0175	0.	0
GRID	199	0	17.	1.0175	0.	0
GRID	200	0	17.	1.01917	0.	0
GRID	201	0	16.	1.01917	0.	0
GRID	202	0	15.	1.01917	0.	0
GRID	203	0	14.	1.01917	0.	0
GRID	204	0	13.	1.01917	0.	0
GRID	205	0	12.	1.01917	0.	0
GRID	206	0	11.	1.01917	0.	0
GRID	207	0	10.	1.01917	0.	0
GRID	208	0	9.	1.01917	0.	0
GRID	209	0	8.	1.01917	0.	0
GRID	210	0	7.	1.01917	0.	0
GRID	211	0	6.	1.01917	0.	0
GRID	212	0	5.	1.01917	0.	0
GRID	213	0	4.	1.01917	0.	0
GRID	214	0	3.	1.01917	0.	0
GRID	215	0	2.	1.01917	0.	0
GRID	216	0	1.	1.01917	0.	0
GRID	219	0	1.	0.98625	0.	0
GRID	220	0	2.	0.98625	0.	0
GRID	221	0	3.	0.98625	0.	0
GRID	222	0	4.	0.98625	0.	0
GRID	223	0	5.	0.98625	0.	0
GRID	226	0	8.	0.98625	0.	0
GRID	227	0	9.	0.98625	0.	0
GRID	228	0	10.	0.98625	0.	0
GRID	229	0	11.	0.98625	0.	0
GRID	231	0	13.	0.98625	0.	0
GRID	232	0	14.	0.98625	0.	0
GRID	235	0	17.	0.98625	0.	0
GRID	241	0	12.	1.0175	0.	0
GRID	243	0	10.	1.0175	0.	0
GRID	244	0	9.	1.0175	0.	0

TABLE 3-20. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 4 (Continued)

GRID	245	0	8.	1.0175	0.	0
GRID	246	0	7.	1.0175	0.	0
GRID	248	0	5.	1.0175	0.	0
GRID	249	0	4.	1.0175	0.	0
GRID	250	0	3.	1.0175	0.	0
GRID	253	0	0.	0.98625	0.	0
GRID	254	0	0.	0.98458	0.	0
GRID	257	0	3.	0.98458	0.	0
GRID	258	0	4.	0.98458	0.	0
GRID	259	0	5.	0.98458	0.	0
GRID	260	0	6.	0.98458	0.	0
GRID	264	0	10.	0.98458	0.	0
GRID	267	0	13.	0.98458	0.	0
GRID	268	0	14.	0.98458	0.	0
GRID	269	0	15.	0.98458	0.	0
GRID	270	0	16.	0.98458	0.	0
GRID	273	0	16.	0.98625	0.	0
GRID	274	0	15.	0.98625	0.	0
GRID	277	0	12.	0.98625	0.	0
GRID	282	0	7.	0.98625	0.	0
GRID	283	0	6.	0.98625	0.	0
GRID	292	0	2.	0.95542	0.	0
GRID	293	0	3.	0.95542	0.	0
GRID	295	0	5.	0.95542	0.	0
GRID	297	0	7.	0.95542	0.	0
GRID	298	0	8.	0.95542	0.	0
GRID	299	0	9.	0.95542	0.	0
GRID	300	0	10.	0.95542	0.	0
GRID	301	0	11.	0.95542	0.	0
GRID	302	0	12.	0.95542	0.	0
GRID	303	0	13.	0.95542	0.	0
GRID	306	0	16.	0.95542	0.	0
GRID	307	0	17.	0.95542	0.	0
GRID	308	0	17.	0.98458	0.	0
GRID	313	0	12.	0.98458	0.	0
GRID	314	0	11.	0.98458	0.	0
GRID	316	0	9.	0.98458	0.	0
GRID	317	0	8.	0.98458	0.	0
GRID	318	0	7.	0.98458	0.	0
GRID	323	0	2.	0.98458	0.	0
GRID	324	0	1.	0.98458	0.	0
CQUAD4	1	9	1	37	3	36
CQUAD4	2	9	36	3	71	35
CQUAD4	3	9	35	71	5	34
CQUAD4	4	9	34	5	69	33
CQUAD4	5	9	33	69	7	32
CQUAD4	6	9	32	7	67	31
CQUAD4	7	9	31	67	9	30
CQUAD4	8	9	30	9	10	29
CQUAD4	9	9	29	10	11	28
CQUAD4	10	9	28	11	12	27
CQUAD4	11	9	27	12	62	26
CQUAD4	12	9	26	62	14	25
CQUAD4	13	9	25	14	15	24
CQUAD4	14	9	24	15	59	23
CQUAD4	15	9	23	59	58	22
CQUAD4	16	9	22	58	18	21

TABLE 3-20. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 4 (Continued)

CQUAD4	17	9	21	18	56	20
CQUAD4	18	8	37	73	108	3
CQUAD4	19	8	3	108	107	71
CQUAD4	20	8	71	107	41	5
CQUAD4	21	8	5	41	105	69
CQUAD4	22	8	69	105	104	7
CQUAD4	23	8	7	104	44	67
CQUAD4	24	8	67	44	45	9
CQUAD4	25	8	9	45	101	10
CQUAD4	26	8	10	101	47	11
CQUAD4	27	8	11	47	48	12
CQUAD4	28	8	12	48	98	62
CQUAD4	29	8	62	98	50	14
CQUAD4	30	8	14	50	51	15
CQUAD4	31	8	15	51	52	59
CQUAD4	32	8	59	52	53	58
CQUAD4	33	8	58	53	54	18
CQUAD4	34	8	18	54	55	56
CQUAD4	35	7	73	109	75	108
CQUAD4	36	7	108	75	76	107
CQUAD4	37	7	107	76	77	41
CQUAD4	38	7	41	77	78	105
CQUAD4	39	7	105	78	79	104
CQUAD4	40	7	104	79	80	44
CQUAD4	41	7	44	80	138	45
CQUAD4	42	7	45	138	137	101
CQUAD4	43	7	101	137	83	47
CQUAD4	44	7	47	83	135	48
CQUAD4	45	7	48	135	85	98
CQUAD4	46	7	98	85	133	50
CQUAD4	47	7	50	133	87	51
CQUAD4	48	7	51	87	88	52
CQUAD4	49	7	52	88	130	53
CQUAD4	50	7	53	130	129	54
CQUAD4	51	7	54	129	128	55
CQUAD4	52	6	109	145	180	75
CQUAD4	53	6	75	180	112	76
CQUAD4	54	6	76	112	178	77
CQUAD4	55	6	77	178	177	78
CQUAD4	56	6	78	177	115	79
CQUAD4	57	6	79	115	116	80
CQUAD4	58	6	80	116	174	138
CQUAD4	59	6	138	174	118	137
CQUAD4	60	6	137	118	172	83
CQUAD4	61	6	83	172	120	135
CQUAD4	62	6	135	120	121	85
CQUAD4	63	6	85	121	122	133
CQUAD4	64	6	133	122	123	87
CQUAD4	65	6	87	123	124	88
CQUAD4	66	6	88	124	125	130
CQUAD4	67	6	130	125	165	129
CQUAD4	68	6	129	165	164	128
CQUAD4	69	5	145	146	147	180
CQUAD4	70	5	180	147	292	112
CQUAD4	71	5	112	292	293	178
CQUAD4	72	5	178	293	150	177
CQUAD4	73	5	177	150	295	115

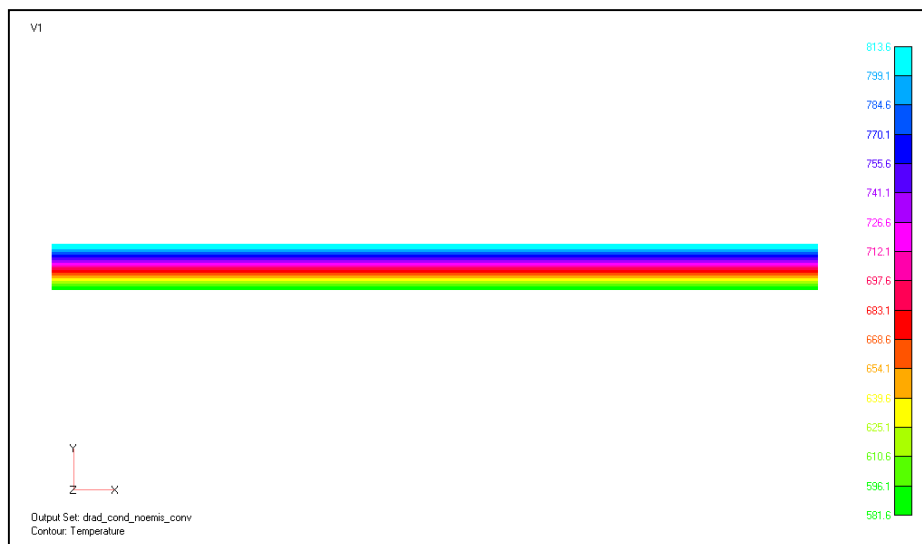
TABLE 3-20. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 4 (Continued)

CQUAD4	74	5	115	295	152	116
CQUAD4	75	5	116	152	297	174
CQUAD4	76	5	174	297	298	118
CQUAD4	77	5	118	298	299	172
CQUAD4	78	5	172	299	300	120
CQUAD4	79	5	120	300	301	121
CQUAD4	80	5	121	301	302	122
CQUAD4	81	5	122	302	303	123
CQUAD4	82	5	123	303	160	124
CQUAD4	83	5	124	160	161	125
CQUAD4	84	5	125	161	306	165
CQUAD4	85	5	165	306	307	164
CQUAD4	86	1	182	181	216	183
CQUAD4	87	1	183	216	215	184
CQUAD4	88	1	184	215	214	250
CQUAD4	89	1	250	214	213	249
CQUAD4	90	1	249	213	212	248
CQUAD4	91	1	248	212	211	188
CQUAD4	92	1	188	211	210	246
CQUAD4	93	1	246	210	209	245
CQUAD4	94	1	245	209	208	244
CQUAD4	95	1	244	208	207	243
CQUAD4	96	1	243	207	206	193
CQUAD4	97	1	193	206	205	241
CQUAD4	98	1	241	205	204	195
CQUAD4	99	1	195	204	203	196
CQUAD4	100	1	196	203	202	197
CQUAD4	101	1	197	202	201	198
CQUAD4	102	1	198	201	200	199
CQUAD4	103	2	253	182	183	219
CQUAD4	104	2	219	183	184	220
CQUAD4	105	8	220	184	250	221
CQUAD4	106	2	221	250	249	222
CQUAD4	107	2	222	249	248	223
CQUAD4	108	2	223	248	188	283
CQUAD4	109	2	283	188	246	282
CQUAD4	110	2	282	246	245	226
CQUAD4	111	2	226	245	244	227
CQUAD4	112	2	227	244	243	228
CQUAD4	113	2	228	243	193	229
CQUAD4	114	2	229	193	241	277
CQUAD4	115	2	277	241	195	231
CQUAD4	116	2	231	195	196	232
CQUAD4	117	2	232	196	197	274
CQUAD4	118	2	274	197	198	273
CQUAD4	119	2	273	198	199	235
CQUAD4	120	3	254	253	219	324
CQUAD4	121	3	324	219	220	323
CQUAD4	122	3	323	220	221	257
CQUAD4	123	3	257	221	222	258
CQUAD4	124	3	258	222	223	259
CQUAD4	125	3	259	223	283	260
CQUAD4	126	3	260	283	282	318
CQUAD4	127	3	318	282	226	317
CQUAD4	128	3	317	226	227	316
CQUAD4	129	3	316	227	228	264
CQUAD4	130	3	264	228	229	314

TABLE 3-20. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 4 (Continued)

CQUAD4	131	3	314	229	277	313
CQUAD4	132	3	313	277	231	267
CQUAD4	133	3	267	231	232	268
CQUAD4	134	3	268	232	274	269
CQUAD4	135	3	269	274	273	270
CQUAD4	136	3	270	273	235	308
CQUAD4	137	4	146	254	324	147
CQUAD4	138	4	147	324	323	292
CQUAD4	139	4	292	323	257	293
CQUAD4	140	4	293	257	258	150
CQUAD4	141	4	150	258	259	295
CQUAD4	142	4	295	259	260	152
CQUAD4	143	4	152	260	318	297
CQUAD4	144	4	297	318	317	298
CQUAD4	145	4	298	317	316	299
CQUAD4	146	4	299	316	264	300
CQUAD4	147	4	300	264	314	301
CQUAD4	148	4	301	314	313	302
CQUAD4	149	4	302	313	267	303
CQUAD4	150	4	303	267	268	160
CQUAD4	151	4	160	268	269	161
CQUAD4	152	4	161	269	270	306
CQUAD4	153	4	306	270	308	307
ENDDATA						

The results of the one-half FEM wing model analysis are presented in figure 3-4 in the form of a contour plot. The contours appear to be indicative of the temperature profile of the THERMOD analysis. The top surface, which was subjected to direct solar radiation, is certainly much hotter than the bottom surface.



Note that the temperatures are in absolute terms ($^{\circ}\text{R}$).

FIGURE 3-4. CONTOUR PLOT OF THE FEM OUTPUT OF SAMPLE PROBLEM 4

The truncated FEM output file is shown in table 3-21. The nodes, from top to bottom of the wing, located at mid-span of the model are indicated by nodes 209, 245, 226, 317, 298, 118, 137, 101, 10, and 29. The temperatures of these nodes were extracted from this file and are reproduced in table 3-22. Mid-span was chosen because it was sufficiently far away from any edge effects that might be present. Note that temperatures in table 3-22 were reduced from absolute temperatures ($^{\circ}\text{R}$) to $^{\circ}\text{F}$.

TABLE 3-21. TRUNCATED FEM OUTPUT FILE OF SAMPLE PROBLEM 4

1	DIRECT_RADIATION_CONDUCTION_NOEMISSION_CONVECTION (T=0.001FT						MARCH 15, 2000	UAI/NASTRAN VERSION 20.1	PAGE
18	10.7 IN GAP FILLED BY AIR; K=0.016BTU/HR/FT/DEGF								
0									
	T E M P E R A T U R E V E C T O R								
	POINT ID.	TYPE	ID VALUE	ID+1 VALUE	ID+2 VALUE	ID+3 VALUE	ID+4 VALUE	ID+5 VALUE	
	1	TEMP	5.816010E+02						
	3	TEMP	5.816228E+02						
	5	TEMP	5.816228E+02						
	7	TEMP	5.816228E+02						
	9	TEMP	5.816228E+02	5.816228E+02	5.816228E+02	5.816228E+02			
	14	TEMP	5.816228E+02	5.816228E+02					
	18	TEMP	5.816228E+02						
	20	TEMP	5.816010E+02	5.816010E+02	5.816010E+02	5.816010E+02	5.816010E+02	5.816010E+02	
	26	TEMP	5.816010E+02	5.816010E+02	5.816010E+02	5.816010E+02	5.816010E+02	5.816010E+02	
	32	TEMP	5.816010E+02	5.816010E+02	5.816010E+02	5.816010E+02	5.816010E+02	5.816228E+02	
	41	TEMP	5.874816E+02						
	44	TEMP	5.874816E+02	5.874816E+02					
	47	TEMP	5.874816E+02	5.874816E+02					
	50	TEMP	5.874816E+02	5.874816E+02	5.874816E+02	5.874816E+02	5.874816E+02	5.874816E+02	
	56	TEMP	5.816228E+02						
	58	TEMP	5.816228E+02	5.816228E+02					
	62	TEMP	5.816228E+02						
	67	TEMP	5.816228E+02						
	69	TEMP	5.816228E+02						
	71	TEMP	5.816228E+02						
	73	TEMP	5.874816E+02						
	75	TEMP	5.875035E+02	5.875035E+02	5.875035E+02	5.875035E+02	5.875035E+02	5.875035E+02	
	83	TEMP	5.875035E+02						
	85	TEMP	5.875035E+02						
	87	TEMP	5.875035E+02	5.875035E+02					
	98	TEMP	5.874816E+02						
	101	TEMP	5.874816E+02						
	104	TEMP	5.874816E+02	5.874816E+02					
	107	TEMP	5.874816E+02	5.874816E+02	5.875035E+02				
	112	TEMP	5.878863E+02						
	115	TEMP	5.878863E+02	5.878863E+02					
	118	TEMP	5.878863E+02						
	120	TEMP	5.878863E+02	5.878863E+02	5.878863E+02	5.878863E+02	5.878863E+02	5.878863E+02	
	128	TEMP	5.875035E+02	5.875035E+02	5.875035E+02				
	133	TEMP	5.875035E+02						
	135	TEMP	5.875035E+02						
	137	TEMP	5.875035E+02	5.875035E+02					
	145	TEMP	5.878863E+02	8.073008E+02	8.073008E+02				
	150	TEMP	8.073008E+02						
	152	TEMP	8.073008E+02						
	160	TEMP	8.073008E+02	8.073008E+02					
	164	TEMP	5.878863E+02	5.878863E+02					
	172	TEMP	5.878863E+02						
	174	TEMP	5.878863E+02						
	177	TEMP	5.878863E+02	5.878863E+02					
	180	TEMP	5.878863E+02	8.135643E+02	8.135643E+02	8.135643E+02			
	188	TEMP	8.135643E+02						
	193	TEMP	8.135643E+02						
	195	TEMP	8.135643E+02	8.135643E+02	8.135643E+02	8.135643E+02	8.135643E+02	8.135862E+02	
	201	TEMP	8.135862E+02	8.135862E+02	8.135862E+02	8.135862E+02	8.135862E+02	8.135862E+02	
1	DIRECT_RADIATION_CONDUCTION_NOEMISSION_CONVECTION (T=0.001FT						MARCH 15, 2000	UAI/NASTRAN VERSION 20.1	PAGE
19	10.7 IN GAP FILLED BY AIR; K=0.016BTU/HR/FT/DEGF								
0									
	T E M P E R A T U R E V E C T O R								
	POINT ID.	TYPE	ID VALUE	ID+1 VALUE	ID+2 VALUE	ID+3 VALUE	ID+4 VALUE	ID+5 VALUE	
	207	TEMP	8.135862E+02	8.135862E+02	8.135862E+02	8.135862E+02	8.135862E+02	8.135862E+02	
	213	TEMP	8.135862E+02	8.135862E+02	8.135862E+02	8.135862E+02			
	219	TEMP	8.077054E+02	8.077054E+02	8.077054E+02	8.077054E+02	8.077054E+02		
	226	TEMP	8.077054E+02	8.077054E+02	8.077054E+02	8.077054E+02			
	231	TEMP	8.077054E+02	8.077054E+02					
	235	TEMP	8.077054E+02						
	241	TEMP	8.135643E+02						
	243	TEMP	8.135643E+02	8.135643E+02	8.135643E+02	8.135643E+02			
	248	TEMP	8.135643E+02	8.135643E+02					
	253	TEMP	8.077054E+02	8.076835E+02					
	257	TEMP	8.076835E+02	8.076835E+02	8.076835E+02	8.076835E+02			
	264	TEMP	8.076835E+02						
	267	TEMP	8.076835E+02	8.076835E+02	8.076835E+02	8.076835E+02			
	273	TEMP	8.077054E+02	8.077054E+02					

TABLE 3-21. TRUNCATED FEM OUTPUT FILE OF SAMPLE PROBLEM 4 (Continued)

277	TEMP	8.077054E+02						
282	TEMP	8.077054E+02	8.077054E+02					
292	TEMP	8.073008E+02	8.073008E+02					
295	TEMP	8.073008E+02						
297	TEMP	8.073008E+02	8.073008E+02	8.073008E+02	8.073008E+02	8.073008E+02	8.073008E+02	
303	TEMP	8.073008E+02						
306	TEMP	8.073008E+02	8.073008E+02	8.076835E+02				
313	TEMP	8.076835E+02	8.076835E+02					
316	TEMP	8.076835E+02	8.076835E+02	8.076835E+02				
323	TEMP	8.076835E+02	8.076835E+02					
500	TEMP	5.786700E+02						

When compared, the temperatures of tables 3-19 and 3-22 show excellent agreement, validating the radiation, conduction, and convection capabilities of the THERMOD program.

TABLE 3-22. FINITE ELEMENT METHOD-SIMULATED TEMPERATURES (°F) OF THE RIGHT WING OF SAMPLE PROBLEM 4

T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
353.9	353.9	348.0	348.0	347.6	128.2	127.8	127.8	122.0	121.9

3.5 SAMPLE PROBLEM 5.

In sample problem 4, the combined effects of conduction, convection, and direct solar radiation were studied. The wing surfaces, in addition to other external surfaces however, were assigned 0.0 emissivity values, meaning that the surfaces were rendered incapable of emitting infrared radiation. The purpose of sample problem 5 was to study and validate the effects of infrared emission. This was done by removing the emission restriction on the wing surfaces of sample problem 4, by changing it from 0.0 to 0.9. This action was expected to result in lower wing temperatures than in sample problem 4. As in sample problems 1 through 4, THERMOD was set up to solve a complete aircraft problem, while the FEM method was confined to the wing model.

To ensure consistency with the FEM wing model, the wing in the THERMOD model was isolated from infrared radiation effects from the fuselage and tarmac surfaces by assigning emissivity values of 0.0. As in sample problem 4, the sky radiation was assigned an absolute temperature of 0.0 to preclude infrared radiation from the sky. The conduction and convection heat transfer modes, however, were left intact, as in sample problem 4.

Despite the above setup, particularly the efforts to isolate the wing, a preliminary comparative study of the THERMOD and FEM results showed significant differences in wing temperatures. This discrepancy was resolved by further isolating the wing from the tarmac effects by increasing the height of the wing from the tarmac to 100.0 ft from its current 2.5 ft. In addition, the infrared reflection effects from the fuselage surface were reduced by decreasing the cabin height from 4 ft to 0.1 ft.

Despite these additions, temperature differences still existed between the THERMOD and FEM wing models, particularly on elements on the wing top surface. The temperatures on the wing bottom surface were found to be in excellent agreement, however. This led to the believe that the fillet effect, which has been modeled as an integral part of THERMOD, could be a plausible cause of this problem. The fillet effect is defined as the effect found in the fuselage-wing

junctions, usually covered by fairing, that serves to increase the local temperatures because the surface is concave. A concave surface is viewed as a partial cavity, and in thermal processes, it is the nature of cavities to increase temperatures regardless of the paint color. In THERMOD, the fillet was modeled as a quadrant. Its temperature-increasing effect was confined to direct solar radiation only. This explains why only the top wing surface temperatures were different in the THERMOD and FEM models.

The fillet effect was removed from the wing top by assigning a 0.0 value in the second field of input data set 6, in essence eliminating the fillet effect. See the THERMOD User's Reference Manual for further details on the input requirements for fillet effects. With the removal of the fillet effect, there was excellent agreement in the temperatures of the THERMOD and FEM wing models. The THERMOD input file, input.dat, is illustrated in table 3-23.

TABLE 3-23. THERMOD INPUT FILE (input.dat) OF SAMPLE PROBLEM 5

```
198e-6,0.704
1, 0.020,125,0.300,0.30
2, 0.375,4.4,0.021,0.24
3, 0.020,125,0.300,0.30
4, 0.350,125,0.300,0.30
5, 10.70,0.067,0.016,0.24
6, 0.350,125,0.300,0.30
7, 0.020,125,0.300,0.30
8, 0.375,4.4,0.021,0.24
9, 0.020,125,0.300,0.30
10,0.030,125,0.300,0.30
11,0.375,4.4,0.021,0.24
12,0.020,125,0.300,0.30
13,0.250,10.,0.020,0.24
14,48.00,0.067,0.016,0.24
15,0.020,125,0.300,0.30
16,0.375,4.4,0.021,0.24
17,0.020,125,0.300,0.30
18,0.250,10.,0.020,0.24
19,30.00,0.067,0.016,0.24
20,5.000,10.,0.020,0.24
21,0.020,125,0.300,0.30
22,0.375,4.4,0.021,0.24
23,0.020,125,0.300,0.30
24,2.500,0.067,0.016,0.24
25,0.020,125,0.300,0.30
26,0.375,4.4,0.021,0.24
27,0.020,125,0.300,0.30
```


TABLE 3-24. THERMOD OUTPUT FILE (summary.dat) FOR SAMPLE
 PROBLEM 5 (Continued)

TEMPERATURES IN DEGREES FAHRENHEIT							
	1	2	3	4	5	6	7
1	146.4	166.2	172.3	162.4	141.2	111.0	79.4
2	146.4	166.2	172.3	162.4	141.2	111.0	79.4
3	143.7	163.1	169.1	159.5	138.8	109.4	78.6
4	143.7	163.1	169.1	159.5	138.8	109.4	78.6
5	143.5	162.9	168.9	159.3	138.7	109.3	78.5
6	43.4	45.9	49.1	50.6	50.5	50.0	48.3
7	43.2	45.7	48.9	50.4	50.3	49.9	48.3
8	43.2	45.7	48.9	50.4	50.3	49.9	48.3
9	40.5	42.5	45.7	47.5	47.9	48.3	47.5
10	40.5	42.5	45.7	47.5	47.9	48.3	47.5
11	146.4	166.2	172.3	162.4	141.2	111.0	79.4
12	146.4	166.2	172.3	162.4	141.2	111.0	79.4
13	143.7	163.1	169.1	159.5	138.8	109.4	78.6
14	143.7	163.1	169.1	159.5	138.8	109.4	78.6
15	143.5	162.9	168.9	159.3	138.7	109.3	78.5
16	43.4	45.9	49.1	50.6	50.5	50.0	48.3
17	43.2	45.7	48.9	50.4	50.3	49.9	48.3
18	43.2	45.7	48.9	50.4	50.3	49.9	48.3
19	40.5	42.5	45.7	47.5	47.9	48.3	47.5
20	40.5	42.5	45.7	47.5	47.9	48.3	47.5
21	273.7	307.6	319.4	301.1	265.1	215.9	166.4
22	273.5	307.4	319.2	300.9	265.0	215.8	166.4
23	232.5	254.2	264.3	249.5	225.7	189.9	149.9
24	232.4	254.0	264.1	249.3	225.5	189.8	149.9
25	203.6	216.8	225.7	213.3	198.0	171.7	138.4
26	161.0	169.1	177.3	167.7	158.3	140.2	115.4
27	133.2	138.5	144.9	142.3	138.7	131.2	119.6
28	133.2	138.5	144.9	142.3	138.7	131.2	119.6
29	131.2	136.3	142.6	140.5	137.3	130.5	119.9
30	131.2	136.3	142.6	140.5	137.3	130.5	119.9
31	122.5	126.7	132.4	132.5	131.1	127.7	121.2
32	122.5	126.7	132.4	132.5	131.1	127.7	121.2
33	122.5	126.7	132.4	132.5	131.1	127.7	121.2
34	122.5	126.7	132.4	132.5	131.1	127.7	121.2
35	113.8	117.1	122.3	124.6	125.0	124.9	122.6
36	113.8	117.1	122.3	124.6	125.0	124.9	122.6
37	111.8	114.9	120.0	122.8	123.6	124.3	122.9
38	111.8	114.9	120.0	122.8	123.6	124.3	122.9
39	214.6	176.2	128.7	178.9	212.9	221.3	202.5
40	214.4	176.2	128.8	178.8	212.8	221.1	202.3
41	189.0	172.6	151.6	173.4	187.0	183.5	163.2
42	188.9	172.6	151.7	173.3	186.9	183.4	163.0
43	171.1	170.1	167.7	169.5	168.8	157.0	135.6

TABLE 3-24. THERMOD OUTPUT FILE (summary.dat) FOR SAMPLE
 PROBLEM 5 (Continued)

44	161.2	164.1	167.7	163.8	160.0	148.1	129.4
45	144.7	147.6	151.7	150.1	147.8	140.2	127.3
46	144.7	147.5	151.6	150.0	147.8	140.2	127.3
47	121.1	124.1	128.8	130.4	130.4	128.9	124.3
48	121.0	123.9	128.7	130.3	130.3	128.8	124.3
49	88.4	92.9	98.4	95.3	91.9	84.7	74.4
50	90.9	95.5	101.1	97.8	94.1	86.4	75.6
51	188.8	207.4	215.7	208.8	191.6	168.5	144.8
52	183.9	201.5	209.6	203.3	187.3	165.7	143.4
53	188.8	207.4	215.7	208.8	191.6	168.5	144.8
Maximum Structural Temperature for Period 1 = 273.69 Occurring at Location 21							
Maximum Structural Temperature for Period 2 = 307.55 Occurring at Location 21							
Maximum Structural Temperature for Period 3 = 319.43 Occurring at Location 21							
Maximum Structural Temperature for Period 4 = 301.07 Occurring at Location 21							
Maximum Structural Temperature for Period 5 = 265.15 Occurring at Location 21							
Maximum Structural Temperature for Period 6 = 221.30 Occurring at Location 39							
Maximum Structural Temperature for Period 7 = 202.48 Occurring at Location 39							
Maximum Structural Temperature Over All 7 Periods =319.43 Occurring at Location 21 at Period 3							
TEMPERATURES AT THE END OF TRANSIENT ANALYSIS							
TEMPERATURES IN DEGREES FAHRENHEIT							
	1	2	3	4	5	6	7
1	114.4	119.2	124.3	126.0	124.8	122.8	119.0
2	115.2	120.3	125.5	126.9	125.2	122.4	117.9
3	141.2	159.5	165.4	156.6	137.5	110.5	81.8
4	142.0	160.7	166.7	157.5	137.9	110.2	80.7
5	142.7	161.7	167.7	158.3	138.1	109.8	79.6
6	45.1	47.7	51.0	52.4	52.4	52.0	50.2
7	46.7	49.2	52.5	54.1	54.1	53.7	52.0
8	48.4	50.9	54.3	55.9	55.9	55.6	53.9
9	103.2	106.0	110.8	113.6	114.5	115.4	114.4
10	104.8	107.7	112.5	115.4	116.3	117.3	116.3
11	114.4	119.2	124.3	126.0	124.8	122.8	119.0
12	115.2	120.3	125.5	126.9	125.2	122.4	117.9

TABLE 3-24. THERMOD OUTPUT FILE (summary.dat) FOR SAMPLE
 PROBLEM 5 (Continued)

13	141.2	159.5	165.4	156.6	137.5	110.5	81.8
14	142.0	160.7	166.7	157.5	137.9	110.2	80.7
15	142.7	161.7	167.7	158.3	138.1	109.8	79.6
16	45.1	47.7	51.0	52.4	52.4	52.0	50.2
17	46.7	49.2	52.5	54.1	54.1	53.7	52.0
18	48.4	50.9	54.3	55.9	55.9	55.6	53.9
19	103.2	106.0	110.8	113.6	114.5	115.4	114.4
20	104.8	107.7	112.5	115.4	116.3	117.3	116.3
21	124.8	130.4	136.0	137.2	135.0	131.8	126.7
22	126.6	132.4	138.1	139.0	136.5	132.7	126.9
23	180.2	193.0	200.8	192.9	180.0	159.6	135.1
24	181.6	194.5	202.3	194.2	181.1	160.2	135.2
25	175.0	185.3	192.9	184.5	173.4	154.2	129.7
26	150.4	157.7	164.8	158.5	151.3	137.2	117.3
27	133.3	138.6	145.0	142.4	138.7	131.2	119.6
28	132.8	138.0	144.4	141.9	138.5	131.1	119.6
29	131.2	136.3	142.6	140.5	137.3	130.5	119.9
30	131.2	136.3	142.6	140.5	137.3	130.5	119.9
31	122.5	126.6	132.3	132.5	131.1	127.7	121.2
32	122.5	126.6	132.3	132.5	131.1	127.7	121.2
33	122.5	126.6	132.3	132.5	131.1	127.7	121.2
34	122.5	126.6	132.3	132.5	131.1	127.7	121.2
35	113.7	117.0	122.1	124.5	125.0	124.9	122.6
36	113.7	116.9	122.0	124.5	125.0	124.9	122.6
37	111.2	114.2	119.2	122.2	123.1	124.1	123.0
38	111.1	114.1	119.1	122.1	123.1	124.0	123.0
39	116.0	117.1	119.8	124.8	127.3	128.6	126.6
40	117.3	118.2	120.5	125.8	128.3	129.4	127.0
41	153.0	149.4	144.5	152.4	156.1	151.2	137.7
42	154.0	150.4	145.5	153.2	156.8	151.7	137.7
43	153.1	155.0	156.3	156.8	155.0	145.8	129.8
44	149.5	152.6	156.3	154.7	151.8	142.8	128.0
45	141.2	143.0	145.5	146.4	145.5	139.4	127.8
46	140.3	142.1	144.5	145.5	144.8	139.0	127.7
47	116.4	117.7	120.5	125.3	127.5	128.5	126.3
48	115.6	116.9	119.8	124.6	126.9	128.1	126.3
49	105.5	108.7	113.7	115.5	115.7	115.1	112.5
50	106.3	109.7	114.9	116.3	116.1	114.9	111.5
51	306.1	348.1	361.3	339.6	295.0	235.5	177.6
52	288.9	327.5	340.0	320.5	279.9	225.7	172.8
53	306.1	348.1	361.3	339.6	295.0	235.5	177.6
Maximum Structural Temperature for Period					1	= 181.56	Occurring
at Location 24							
Maximum Structural Temperature for Period					2	= 194.50	Occurring
at Location 24							

TABLE 3-24. THERMOD OUTPUT FILE (summary.dat) FOR SAMPLE
PROBLEM 5 (Continued)

Maximum Structural Temperature for Period at Location 24	3 = 202.33	Occurring
Maximum Structural Temperature for Period at Location 24	4 = 194.16	Occurring
Maximum Structural Temperature for Period at Location 24	5 = 181.07	Occurring
Maximum Structural Temperature for Period at Location 24	6 = 160.21	Occurring
Maximum Structural Temperature for Period at Location 42	7 = 137.74	Occurring
Maximum Structural Temperature Over All Occurring at Location 24 at Period 3	7 Periods = 202.33	

TABLE 3-25. THERMOD-SIMULATED TEMPERATURES (°F) OF THE RIGHT WING OF
SAMPLE PROBLEM 5

T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
172.3	172.3	169.1	169.1	168.9	49.1	48.9	48.9	45.7	45.7

As in sample problems 1 through 4, a one-half wing structure was modeled using the UAI/NASTRAN FEM program. Two types of loads were applied to the model. These loads, based on period 3 of the THERMOD input file, were direct solar radiation of 355 Btu/hr/ft² and convection with a film coefficient of $h = 1.3433 \text{ (Btu/hr)/(ft}^2\text{°F)}$. This film coefficient was calculated internally by the THERMOD program and was based on the theory of airflow on a flat surface and a wind speed of 14 ft/sec. The radiation load was applied perpendicular to the top surface only, while both the top and bottom surfaces were subjected to convection forces. The ambient temperature was maintained at 119°F (578.67°R). Unlike sample problem 4, the wings were assigned a nonzero emissivity value of 0.9, allowing the wings to emit infrared radiation, thereby, reducing the wing temperatures.

The FEM input file is shown in table 3-26. The ambient temperature of 119°F was imposed through the SPC case control card. The ambient temperature boundary condition was scalar in nature and was implemented through the SPOINT bulk data. The boundary elements, CHBDY, through which solar radiation and convection effects were applied, were modeled using the LINE boundary shape type, connecting the boundary nodes along the wing top and bottom. The SPACE designation in the CHBDY boundary element indicates that this element emits infrared radiation to a black body, in which case, no VIEW bulk data was required. Property cards PHBDY and the material cards MAT4, define absorptivity, emissivity, film coefficient, and conductivity values. Note that unlike sample problem 4 where wings were prevented from emitting radiation, an emissivity value of 0.9 was assigned to the wing in sample problems, as indicated in the PHBDY card. Because of the involvement of infrared radiation, through emission, this sample problem was rendered nonlinear. This required the Stefan-Boltzman constant, σ , and an initial set of temperatures, which were implemented through the PARAM and TEMPERATURE statements, respectively.

TABLE 3-26. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE PROBLEM 5

```

ID D:\Natha, MSC/N
APP HEAT
SOL 3
TIME 10000
CEND
  TITLE = direct_radiation_conduction_emission_convection (t=0.001ft)
  SUBTITLE = 10.7 in gap filled by air; k=0.016BTU/hr/ft/degF
    load=1
    SPC = 1
    temperature(estimate)=2
    THERMAL= ALL
    FLUX = ALL
    SPCF = ALL
BEGIN BULK
$ *****
$   Written by : MSC/NASTRAN for Windows
$   Version   : 6.00
$   Translator : UAI/NASTRAN
$   From Model : D:\Thermod_Validation\FEM\conduction_only.MOD
$   Date      : Mon Mar 06 11:27:09 2000
$ *****
$
PARAM, K6ROT, 100.
PARAM, MAXRATIO, 1.E+8
param, sigma, 1.714e-9
param, tabs, 0.0
param, maxit, 10
CORD2C      1      0      0.      0.      0.      0.      0.      1.+MSC/NC1
+MSC/NC1     1.      0.      1.
CORD2S      2      0      0.      0.      0.      0.      0.      1.+MSC/NC2
+MSC/NC2     1.      0.      1.
$
$ MSC/NASTRAN for Windows Property 1 : 0.02 in glass_outer_layer
PSHELL      1      1      0.001      1      1      0.
$ MSC/NASTRAN for Windows Property 2 : 3/8 in foam
PSHELL      2      2      0.001      2      2      0.
$ MSC/NASTRAN for Windows Property 3 : 0.02 in inner glass
PSHELL      3      1      0.001      1      1      0.
$ MSC/NASTRAN for Windows Property 4 : 0.350 in cap
PSHELL      4      1      0.001      1      1      0.
$ MSC/NASTRAN for Windows Property 5 : 10.70 in air space
PSHELL      5      3      0.001      3      3      0.
$ MSC/NASTRAN for Windows Property 6 : 0.350 in cap
PSHELL      6      1      0.001      1      1      0.
$ MSC/NASTRAN for Windows Property 7 : 0.02 in inner glass
PSHELL      7      1      0.001      1      1      0.
$ MSC/NASTRAN for Windows Property 8 : 3/8 in foam
PSHELL      8      2      0.001      2      2      0.
$ MSC/NASTRAN for Windows Property 9 : 0.02 in glass_outer_layer
PSHELL      9      1      0.001      1      1      0.
$ MSC/NASTRAN for Windows Material 1 : glass
MAT4         1      0.3      1.
$ MSC/NASTRAN for Windows Material 2 : foam
MAT4         2      0.021      1.

```

TABLE 3-26. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 5 (Continued)

```
$ MSC/NASTRAN for Windows Material 3 : air
MAT4          3    0.016    1.
$
chbdy,1001,100,line,200,201,,,space,500,500,,,0.0,1.0,0.0
chbdy,1002,100,line,201,202,,,space,500,500,,,0.0,1.0,0.0
chbdy,1003,100,line,202,203,,,space,500,500,,,0.0,1.0,0.0
chbdy,1004,100,line,203,204,,,space,500,500,,,0.0,1.0,0.0
chbdy,1005,100,line,204,205,,,space,500,500,,,0.0,1.0,0.0
chbdy,1006,100,line,205,206,,,space,500,500,,,0.0,1.0,0.0
chbdy,1007,100,line,206,207,,,space,500,500,,,0.0,1.0,0.0
chbdy,1008,100,line,207,208,,,space,500,500,,,0.0,1.0,0.0
chbdy,1009,100,line,208,209,,,space,500,500,,,0.0,1.0,0.0
chbdy,1010,100,line,209,210,,,space,500,500,,,0.0,1.0,0.0
chbdy,1011,100,line,210,211,,,space,500,500,,,0.0,1.0,0.0
chbdy,1012,100,line,211,212,,,space,500,500,,,0.0,1.0,0.0
chbdy,1013,100,line,212,213,,,space,500,500,,,0.0,1.0,0.0
chbdy,1014,100,line,213,214,,,space,500,500,,,0.0,1.0,0.0
chbdy,1015,100,line,214,215,,,space,500,500,,,0.0,1.0,0.0
chbdy,1016,100,line,215,216,,,space,500,500,,,0.0,1.0,0.0
chbdy,1017,100,line,216,181,,,space,500,500,,,0.0,1.0,0.0
phbdy,100,200,0.001,0.9,0.9
mat4,200,1.3433
spoint,500
spc,1,500,1,578.67
$
$ Direct radiation load from a distant source: load case 1
qvect,1,355.0,0.0,-1.0,0.0,1001,thru,1017
$
$ view ids
$view,1,yes,yes,1,1,0.0
$
chbdy,2001,300,line,20,21,,,space,500,500,,,0.0,-1.0,0.0
chbdy,2002,300,line,21,22,,,space,500,500,,,0.0,-1.0,0.0
chbdy,2003,300,line,22,23,,,space,500,500,,,0.0,-1.0,0.0
chbdy,2004,300,line,23,24,,,space,500,500,,,0.0,-1.0,0.0
chbdy,2005,300,line,24,25,,,space,500,500,,,0.0,-1.0,0.0
chbdy,2006,300,line,25,26,,,space,500,500,,,0.0,-1.0,0.0
chbdy,2007,300,line,26,27,,,space,500,500,,,0.0,-1.0,0.0
chbdy,2008,300,line,27,28,,,space,500,500,,,0.0,-1.0,0.0
chbdy,2009,300,line,28,29,,,space,500,500,,,0.0,-1.0,0.0
chbdy,2010,300,line,29,30,,,space,500,500,,,0.0,-1.0,0.0
chbdy,2011,300,line,30,31,,,space,500,500,,,0.0,-1.0,0.0
chbdy,2012,300,line,31,32,,,space,500,500,,,0.0,-1.0,0.0
chbdy,2013,300,line,32,33,,,space,500,500,,,0.0,-1.0,0.0
chbdy,2014,300,line,33,34,,,space,500,500,,,0.0,-1.0,0.0
chbdy,2015,300,line,34,35,,,space,500,500,,,0.0,-1.0,0.0
chbdy,2016,300,line,35,36,,,space,500,500,,,0.0,-1.0,0.0
chbdy,2017,300,line,36,1,,,space,500,500,,,0.0,-1.0,0.0
phbdy,300,400,0.001,0.9,0.9
mat4,400,1.3433
$
temp,2,1,578.67,3,578.67,5,578.67
temp,2,7,578.67,9,578.67,10,578.67
temp,2,11,578.67,12,578.67,14,578.67
temp,2,15,578.67,18,578.67,20,578.67
temp,2,21,578.67,22,578.67,23,578.67
temp,2,24,578.67,25,578.67,26,578.67
```

TABLE 3-26. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 5 (Continued)

```
temp, 2, 27, 578.67, 28, 578.67, 29, 578.67
temp, 2, 30, 578.67, 31, 578.67, 32, 578.67
temp, 2, 33, 578.67, 34, 578.67, 35, 578.67
temp, 2, 36, 578.67, 37, 578.67, 41, 578.67
temp, 2, 44, 578.67, 45, 578.67, 47, 578.67
temp, 2, 48, 578.67, 50, 578.67, 51, 578.67
temp, 2, 52, 578.67, 53, 578.67, 54, 578.67
temp, 2, 55, 578.67, 56, 578.67, 58, 578.67
temp, 2, 59, 578.67, 62, 578.67, 67, 578.67
temp, 2, 69, 578.67, 71, 578.67, 73, 578.67
temp, 2, 75, 578.67, 76, 578.67, 77, 578.67
temp, 2, 78, 578.67, 79, 578.67, 80, 578.67
temp, 2, 83, 578.67, 85, 578.67, 87, 578.67
temp, 2, 88, 578.67, 98, 578.67, 101, 578.67
temp, 2, 104, 578.67, 105, 578.67, 107, 578.67
temp, 2, 108, 578.67, 109, 578.67, 112, 578.67
temp, 2, 115, 578.67, 116, 578.67, 118, 578.67
temp, 2, 120, 578.67, 121, 578.67, 122, 578.67
temp, 2, 123, 578.67, 124, 578.67, 125, 578.67
temp, 2, 128, 578.67, 129, 578.67, 130, 578.67
temp, 2, 133, 578.67, 135, 578.67, 137, 578.67
temp, 2, 138, 578.67, 145, 578.67, 146, 578.67
temp, 2, 147, 578.67, 150, 578.67, 152, 578.67
temp, 2, 160, 578.67, 161, 578.67, 164, 578.67
temp, 2, 165, 578.67, 172, 578.67, 174, 578.67
temp, 2, 177, 578.67, 178, 578.67, 180, 578.67
temp, 2, 181, 578.67, 182, 578.67, 183, 578.67
temp, 2, 184, 578.67, 188, 578.67, 193, 578.67
temp, 2, 195, 578.67, 196, 578.67, 197, 578.67
temp, 2, 198, 578.67, 199, 578.67, 200, 578.67
temp, 2, 201, 578.67, 202, 578.67, 203, 578.67
temp, 2, 204, 578.67, 205, 578.67, 206, 578.67
temp, 2, 207, 578.67, 208, 578.67, 209, 578.67
temp, 2, 210, 578.67, 211, 578.67, 212, 578.67
temp, 2, 213, 578.67, 214, 578.67, 215, 578.67
temp, 2, 216, 578.67, 219, 578.67, 220, 578.67
temp, 2, 221, 578.67, 222, 578.67, 223, 578.67
temp, 2, 226, 578.67, 227, 578.67, 228, 578.67
temp, 2, 229, 578.67, 231, 578.67, 232, 578.67
temp, 2, 235, 578.67, 241, 578.67, 243, 578.67
temp, 2, 244, 578.67, 245, 578.67, 246, 578.67
temp, 2, 248, 578.67, 249, 578.67, 250, 578.67
temp, 2, 253, 578.67, 254, 578.67, 257, 578.67
temp, 2, 258, 578.67, 259, 578.67, 260, 578.67
temp, 2, 264, 578.67, 267, 578.67, 268, 578.67
temp, 2, 269, 578.67, 270, 578.67, 273, 578.67
temp, 2, 274, 578.67, 277, 578.67, 282, 578.67
temp, 2, 283, 578.67, 292, 578.67, 293, 578.67
temp, 2, 295, 578.67, 297, 578.67, 298, 578.67
temp, 2, 299, 578.67, 300, 578.67, 301, 578.67
temp, 2, 302, 578.67, 303, 578.67, 306, 578.67
temp, 2, 307, 578.67, 308, 578.67, 313, 578.67
temp, 2, 314, 578.67, 316, 578.67, 317, 578.67
temp, 2, 318, 578.67, 323, 578.67, 324, 578.67
temp, 2, 500, 578.67
$
```

TABLE 3-26. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 5 (Continued)

GRID	1	0	0.	0.	0.	0
GRID	3	0	1.1.667E-3	0.	0.	0
GRID	5	0	3.1.667E-3	0.	0.	0
GRID	7	0	5.1.667E-3	0.	0.	0
GRID	9	0	7.1.667E-3	0.	0.	0
GRID	10	0	8.1.667E-3	0.	0.	0
GRID	11	0	9.1.667E-3	0.	0.	0
GRID	12	0	10.1.667E-3	0.	0.	0
GRID	14	0	12.1.667E-3	0.	0.	0
GRID	15	0	13.1.667E-3	0.	0.	0
GRID	18	0	16.1.667E-3	0.	0.	0
GRID	20	0	17.	0.	0.	0
GRID	21	0	16.	0.	0.	0
GRID	22	0	15.	0.	0.	0
GRID	23	0	14.	0.	0.	0
GRID	24	0	13.	0.	0.	0
GRID	25	0	12.	0.	0.	0
GRID	26	0	11.	0.	0.	0
GRID	27	0	10.	0.	0.	0
GRID	28	0	9.	0.	0.	0
GRID	29	0	8.	0.	0.	0
GRID	30	0	7.	0.	0.	0
GRID	31	0	6.	0.	0.	0
GRID	32	0	5.	0.	0.	0
GRID	33	0	4.	0.	0.	0
GRID	34	0	3.	0.	0.	0
GRID	35	0	2.	0.	0.	0
GRID	36	0	1.	0.	0.	0
GRID	37	0	0.1.667E-3	0.	0.	0
GRID	41	0	3.0.032917	0.	0.	0
GRID	44	0	6.0.032917	0.	0.	0
GRID	45	0	7.0.032917	0.	0.	0
GRID	47	0	9.0.032917	0.	0.	0
GRID	48	0	10.0.032917	0.	0.	0
GRID	50	0	12.0.032917	0.	0.	0
GRID	51	0	13.0.032917	0.	0.	0
GRID	52	0	14.0.032917	0.	0.	0
GRID	53	0	15.0.032917	0.	0.	0
GRID	54	0	16.0.032917	0.	0.	0
GRID	55	0	17.0.032917	0.	0.	0
GRID	56	0	17.1.667E-3	0.	0.	0
GRID	58	0	15.1.667E-3	0.	0.	0
GRID	59	0	14.1.667E-3	0.	0.	0
GRID	62	0	11.1.667E-3	0.	0.	0
GRID	67	0	6.1.667E-3	0.	0.	0
GRID	69	0	4.1.667E-3	0.	0.	0
GRID	71	0	2.1.667E-3	0.	0.	0
GRID	73	0	0.0.032917	0.	0.	0
GRID	75	0	1.0.034583	0.	0.	0
GRID	76	0	2.0.034583	0.	0.	0
GRID	77	0	3.0.034583	0.	0.	0
GRID	78	0	4.0.034583	0.	0.	0
GRID	79	0	5.0.034583	0.	0.	0
GRID	80	0	6.0.034583	0.	0.	0
GRID	83	0	9.0.034583	0.	0.	0
GRID	85	0	11.0.034583	0.	0.	0
GRID	87	0	13.0.034583	0.	0.	0

TABLE 3-26. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 5 (Continued)

GRID	88	0	14.0.034583	0.	0
GRID	98	0	11.0.032917	0.	0
GRID	101	0	8.0.032917	0.	0
GRID	104	0	5.0.032917	0.	0
GRID	105	0	4.0.032917	0.	0
GRID	107	0	2.0.032917	0.	0
GRID	108	0	1.0.032917	0.	0
GRID	109	0	0.0.034583	0.	0
GRID	112	0	2. 0.06375	0.	0
GRID	115	0	5. 0.06375	0.	0
GRID	116	0	6. 0.06375	0.	0
GRID	118	0	8. 0.06375	0.	0
GRID	120	0	10. 0.06375	0.	0
GRID	121	0	11. 0.06375	0.	0
GRID	122	0	12. 0.06375	0.	0
GRID	123	0	13. 0.06375	0.	0
GRID	124	0	14. 0.06375	0.	0
GRID	125	0	15. 0.06375	0.	0
GRID	128	0	17.0.034583	0.	0
GRID	129	0	16.0.034583	0.	0
GRID	130	0	15.0.034583	0.	0
GRID	133	0	12.0.034583	0.	0
GRID	135	0	10.0.034583	0.	0
GRID	137	0	8.0.034583	0.	0
GRID	138	0	7.0.034583	0.	0
GRID	145	0	0. 0.06375	0.	0
GRID	146	0	0. 0.95542	0.	0
GRID	147	0	1. 0.95542	0.	0
GRID	150	0	4. 0.95542	0.	0
GRID	152	0	6. 0.95542	0.	0
GRID	160	0	14. 0.95542	0.	0
GRID	161	0	15. 0.95542	0.	0
GRID	164	0	17. 0.06375	0.	0
GRID	165	0	16. 0.06375	0.	0
GRID	172	0	9. 0.06375	0.	0
GRID	174	0	7. 0.06375	0.	0
GRID	177	0	4. 0.06375	0.	0
GRID	178	0	3. 0.06375	0.	0
GRID	180	0	1. 0.06375	0.	0
GRID	181	0	0. 1.01917	0.	0
GRID	182	0	0. 1.0175	0.	0
GRID	183	0	1. 1.0175	0.	0
GRID	184	0	2. 1.0175	0.	0
GRID	188	0	6. 1.0175	0.	0
GRID	193	0	11. 1.0175	0.	0
GRID	195	0	13. 1.0175	0.	0
GRID	196	0	14. 1.0175	0.	0
GRID	197	0	15. 1.0175	0.	0
GRID	198	0	16. 1.0175	0.	0
GRID	199	0	17. 1.0175	0.	0
GRID	200	0	17. 1.01917	0.	0
GRID	201	0	16. 1.01917	0.	0
GRID	202	0	15. 1.01917	0.	0
GRID	203	0	14. 1.01917	0.	0
GRID	204	0	13. 1.01917	0.	0
GRID	205	0	12. 1.01917	0.	0
GRID	206	0	11. 1.01917	0.	0

TABLE 3-26. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 5 (Continued)

GRID	207	0	10.	1.01917	0.	0
GRID	208	0	9.	1.01917	0.	0
GRID	209	0	8.	1.01917	0.	0
GRID	210	0	7.	1.01917	0.	0
GRID	211	0	6.	1.01917	0.	0
GRID	212	0	5.	1.01917	0.	0
GRID	213	0	4.	1.01917	0.	0
GRID	214	0	3.	1.01917	0.	0
GRID	215	0	2.	1.01917	0.	0
GRID	216	0	1.	1.01917	0.	0
GRID	219	0	1.	0.98625	0.	0
GRID	220	0	2.	0.98625	0.	0
GRID	221	0	3.	0.98625	0.	0
GRID	222	0	4.	0.98625	0.	0
GRID	223	0	5.	0.98625	0.	0
GRID	226	0	8.	0.98625	0.	0
GRID	227	0	9.	0.98625	0.	0
GRID	228	0	10.	0.98625	0.	0
GRID	229	0	11.	0.98625	0.	0
GRID	231	0	13.	0.98625	0.	0
GRID	232	0	14.	0.98625	0.	0
GRID	235	0	17.	0.98625	0.	0
GRID	241	0	12.	1.0175	0.	0
GRID	243	0	10.	1.0175	0.	0
GRID	244	0	9.	1.0175	0.	0
GRID	245	0	8.	1.0175	0.	0
GRID	246	0	7.	1.0175	0.	0
GRID	248	0	5.	1.0175	0.	0
GRID	249	0	4.	1.0175	0.	0
GRID	250	0	3.	1.0175	0.	0
GRID	253	0	0.	0.98625	0.	0
GRID	254	0	0.	0.98458	0.	0
GRID	257	0	3.	0.98458	0.	0
GRID	258	0	4.	0.98458	0.	0
GRID	259	0	5.	0.98458	0.	0
GRID	260	0	6.	0.98458	0.	0
GRID	264	0	10.	0.98458	0.	0
GRID	267	0	13.	0.98458	0.	0
GRID	268	0	14.	0.98458	0.	0
GRID	269	0	15.	0.98458	0.	0
GRID	270	0	16.	0.98458	0.	0
GRID	273	0	16.	0.98625	0.	0
GRID	274	0	15.	0.98625	0.	0
GRID	277	0	12.	0.98625	0.	0
GRID	282	0	7.	0.98625	0.	0
GRID	283	0	6.	0.98625	0.	0
GRID	292	0	2.	0.95542	0.	0
GRID	293	0	3.	0.95542	0.	0
GRID	295	0	5.	0.95542	0.	0
GRID	297	0	7.	0.95542	0.	0
GRID	298	0	8.	0.95542	0.	0
GRID	299	0	9.	0.95542	0.	0
GRID	300	0	10.	0.95542	0.	0
GRID	301	0	11.	0.95542	0.	0
GRID	302	0	12.	0.95542	0.	0
GRID	303	0	13.	0.95542	0.	0
GRID	306	0	16.	0.95542	0.	0

TABLE 3-26. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 5 (Continued)

GRID	307	0	17.	0.95542	0.	0
GRID	308	0	17.	0.98458	0.	0
GRID	313	0	12.	0.98458	0.	0
GRID	314	0	11.	0.98458	0.	0
GRID	316	0	9.	0.98458	0.	0
GRID	317	0	8.	0.98458	0.	0
GRID	318	0	7.	0.98458	0.	0
GRID	323	0	2.	0.98458	0.	0
GRID	324	0	1.	0.98458	0.	0
CQUAD4	1	9	1	37	3	36
CQUAD4	2	9	36	3	71	35
CQUAD4	3	9	35	71	5	34
CQUAD4	4	9	34	5	69	33
CQUAD4	5	9	33	69	7	32
CQUAD4	6	9	32	7	67	31
CQUAD4	7	9	31	67	9	30
CQUAD4	8	9	30	9	10	29
CQUAD4	9	9	29	10	11	28
CQUAD4	10	9	28	11	12	27
CQUAD4	11	9	27	12	62	26
CQUAD4	12	9	26	62	14	25
CQUAD4	13	9	25	14	15	24
CQUAD4	14	9	24	15	59	23
CQUAD4	15	9	23	59	58	22
CQUAD4	16	9	22	58	18	21
CQUAD4	17	9	21	18	56	20
CQUAD4	18	8	37	73	108	3
CQUAD4	19	8	3	108	107	71
CQUAD4	20	8	71	107	41	5
CQUAD4	21	8	5	41	105	69
CQUAD4	22	8	69	105	104	7
CQUAD4	23	8	7	104	44	67
CQUAD4	24	8	67	44	45	9
CQUAD4	25	8	9	45	101	10
CQUAD4	26	8	10	101	47	11
CQUAD4	27	8	11	47	48	12
CQUAD4	28	8	12	48	98	62
CQUAD4	29	8	62	98	50	14
CQUAD4	30	8	14	50	51	15
CQUAD4	31	8	15	51	52	59
CQUAD4	32	8	59	52	53	58
CQUAD4	33	8	58	53	54	18
CQUAD4	34	8	18	54	55	56
CQUAD4	35	7	73	109	75	108
CQUAD4	36	7	108	75	76	107
CQUAD4	37	7	107	76	77	41
CQUAD4	38	7	41	77	78	105
CQUAD4	39	7	105	78	79	104
CQUAD4	40	7	104	79	80	44
CQUAD4	41	7	44	80	138	45
CQUAD4	42	7	45	138	137	101
CQUAD4	43	7	101	137	83	47
CQUAD4	44	7	47	83	135	48
CQUAD4	45	7	48	135	85	98
CQUAD4	46	7	98	85	133	50
CQUAD4	47	7	50	133	87	51
CQUAD4	48	7	51	87	88	52

TABLE 3-26. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 5 (Continued)

CQUAD4	49	7	52	88	130	53
CQUAD4	50	7	53	130	129	54
CQUAD4	51	7	54	129	128	55
CQUAD4	52	6	109	145	180	75
CQUAD4	53	6	75	180	112	76
CQUAD4	54	6	76	112	178	77
CQUAD4	55	6	77	178	177	78
CQUAD4	56	6	78	177	115	79
CQUAD4	57	6	79	115	116	80
CQUAD4	58	6	80	116	174	138
CQUAD4	59	6	138	174	118	137
CQUAD4	60	6	137	118	172	83
CQUAD4	61	6	83	172	120	135
CQUAD4	62	6	135	120	121	85
CQUAD4	63	6	85	121	122	133
CQUAD4	64	6	133	122	123	87
CQUAD4	65	6	87	123	124	88
CQUAD4	66	6	88	124	125	130
CQUAD4	67	6	130	125	165	129
CQUAD4	68	6	129	165	164	128
CQUAD4	69	5	145	146	147	180
CQUAD4	70	5	180	147	292	112
CQUAD4	71	5	112	292	293	178
CQUAD4	72	5	178	293	150	177
CQUAD4	73	5	177	150	295	115
CQUAD4	74	5	115	295	152	116
CQUAD4	75	5	116	152	297	174
CQUAD4	76	5	174	297	298	118
CQUAD4	77	5	118	298	299	172
CQUAD4	78	5	172	299	300	120
CQUAD4	79	5	120	300	301	121
CQUAD4	80	5	121	301	302	122
CQUAD4	81	5	122	302	303	123
CQUAD4	82	5	123	303	160	124
CQUAD4	83	5	124	160	161	125
CQUAD4	84	5	125	161	306	165
CQUAD4	85	5	165	306	307	164
CQUAD4	86	1	182	181	216	183
CQUAD4	87	1	183	216	215	184
CQUAD4	88	1	184	215	214	250
CQUAD4	89	1	250	214	213	249
CQUAD4	90	1	249	213	212	248
CQUAD4	91	1	248	212	211	188
CQUAD4	92	1	188	211	210	246
CQUAD4	93	1	246	210	209	245
CQUAD4	94	1	245	209	208	244
CQUAD4	95	1	244	208	207	243
CQUAD4	96	1	243	207	206	193
CQUAD4	97	1	193	206	205	241
CQUAD4	98	1	241	205	204	195
CQUAD4	99	1	195	204	203	196
CQUAD4	100	1	196	203	202	197
CQUAD4	101	1	197	202	201	198
CQUAD4	102	1	198	201	200	199
CQUAD4	103	2	253	182	183	219
CQUAD4	104	2	219	183	184	220
CQUAD4	105	8	220	184	250	221

TABLE 3-26. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 5 (Continued)

CQUAD4	106	2	221	250	249	222
CQUAD4	107	2	222	249	248	223
CQUAD4	108	2	223	248	188	283
CQUAD4	109	2	283	188	246	282
CQUAD4	110	2	282	246	245	226
CQUAD4	111	2	226	245	244	227
CQUAD4	112	2	227	244	243	228
CQUAD4	113	2	228	243	193	229
CQUAD4	114	2	229	193	241	277
CQUAD4	115	2	277	241	195	231
CQUAD4	116	2	231	195	196	232
CQUAD4	117	2	232	196	197	274
CQUAD4	118	2	274	197	198	273
CQUAD4	119	2	273	198	199	235
CQUAD4	120	3	254	253	219	324
CQUAD4	121	3	324	219	220	323
CQUAD4	122	3	323	220	221	257
CQUAD4	123	3	257	221	222	258
CQUAD4	124	3	258	222	223	259
CQUAD4	125	3	259	223	283	260
CQUAD4	126	3	260	283	282	318
CQUAD4	127	3	318	282	226	317
CQUAD4	128	3	317	226	227	316
CQUAD4	129	3	316	227	228	264
CQUAD4	130	3	264	228	229	314
CQUAD4	131	3	314	229	277	313
CQUAD4	132	3	313	277	231	267
CQUAD4	133	3	267	231	232	268
CQUAD4	134	3	268	232	274	269
CQUAD4	135	3	269	274	273	270
CQUAD4	136	3	270	273	235	308
CQUAD4	137	4	146	254	324	147
CQUAD4	138	4	147	324	323	292
CQUAD4	139	4	292	323	257	293
CQUAD4	140	4	293	257	258	150
CQUAD4	141	4	150	258	259	295
CQUAD4	142	4	295	259	260	152
CQUAD4	143	4	152	260	318	297
CQUAD4	144	4	297	318	317	298
CQUAD4	145	4	298	317	316	299
CQUAD4	146	4	299	316	264	300
CQUAD4	147	4	300	264	314	301
CQUAD4	148	4	301	314	313	302
CQUAD4	149	4	302	313	267	303
CQUAD4	150	4	303	267	268	160
CQUAD4	151	4	160	268	269	161
CQUAD4	152	4	161	269	270	306
CQUAD4	153	4	306	270	308	307
ENDDATA						

The results of the one-half FEM wing model analysis are presented in figure 3-5 in the form of a contour plot. The contours appear to be indicative of the temperature profile of the THERMOD analysis. The top surface, which was subjected to direct solar radiation, is certainly much hotter than the bottom surface. The temperatures are significantly lower than those in figure 3-4 of sample problem 4 because in sample problem 5 the wing surfaces were modeled to emit infrared radiation energy.



Note that the temperatures are in absolute terms (°R).

FIGURE 3-5. CONTOUR PLOT OF THE FEM OUTPUT OF SAMPLE PROBLEM 5

The truncated FEM output file is shown in table 3-27. The nodes, from top to bottom of the wing, located at mid-span of the model, are indicated by nodes 209, 245, 226, 317, 298, 118, 137, 101, 10, and 29. The temperatures of these nodes were extracted from this file and are reproduced in table 3-28. Mid-span was chosen because it was sufficiently far away from any edge effects that might be present. Note that the temperatures in table 3-28 were reduced from absolute temperatures (°R) to °F.

When compared, the temperatures of tables 3-25 and 3-28 show excellent agreement, validating the radiation, conduction and convection, capabilities of the THERMOD program, including infrared emission.

TABLE 3-27. TRUNCATED FEM OUTPUT FILE OF SAMPLE PROBLEM 5

1	DIRECT_RADIATION_CONDUCTION_EMISSION_CONVECTION (T=0.001FT)					AUGUST 17, 2000	UAI/NASTRAN VERSION 20.1	PAGE
18	10.7 IN GAP FILLED BY AIR; K=0.016BTU/HR/FT/DEGF							
0								
	T E M P E R A T U R E V E C T O R							
	POINT ID.	TYPE	ID VALUE	ID+1 VALUE	ID+2 VALUE	ID+3 VALUE	ID+4 VALUE	ID+5 VALUE
	1	TEMP	5.055055E+02					
	3	TEMP	5.055174E+02					
	5	TEMP	5.055174E+02					
	7	TEMP	5.055174E+02					
	9	TEMP	5.055174E+02	5.055174E+02	5.055174E+02	5.055174E+02		
	14	TEMP	5.055174E+02	5.055174E+02				
	18	TEMP	5.055174E+02					
	20	TEMP	5.055055E+02	5.055055E+02	5.055055E+02	5.055055E+02	5.055055E+02	5.055055E+02
	26	TEMP	5.055055E+02	5.055055E+02	5.055055E+02	5.055055E+02	5.055055E+02	5.055055E+02
	32	TEMP	5.055055E+02	5.055055E+02	5.055055E+02	5.055055E+02	5.055055E+02	5.055174E+02
	41	TEMP	5.087106E+02					
	44	TEMP	5.087106E+02	5.087106E+02				
	47	TEMP	5.087106E+02	5.087106E+02				
	50	TEMP	5.087106E+02	5.087106E+02	5.087106E+02	5.087106E+02	5.087106E+02	5.087106E+02
	56	TEMP	5.055174E+02					
	58	TEMP	5.055174E+02	5.055174E+02				
	62	TEMP	5.055174E+02					
	67	TEMP	5.055174E+02					
	69	TEMP	5.055174E+02					
	71	TEMP	5.055174E+02					
	73	TEMP	5.087106E+02					
	75	TEMP	5.087225E+02	5.087225E+02	5.087225E+02	5.087225E+02	5.087225E+02	5.087225E+02
	83	TEMP	5.087225E+02					
	85	TEMP	5.087225E+02					
	87	TEMP	5.087225E+02	5.087225E+02				
	98	TEMP	5.087106E+02					
	101	TEMP	5.087106E+02					
	104	TEMP	5.087106E+02	5.087106E+02				
	107	TEMP	5.087106E+02	5.087106E+02	5.087225E+02			
	112	TEMP	5.089311E+02					
	115	TEMP	5.089311E+02	5.089311E+02				
	118	TEMP	5.089311E+02					
	120	TEMP	5.089311E+02	5.089311E+02	5.089311E+02	5.089311E+02	5.089311E+02	5.089311E+02
	128	TEMP	5.087225E+02	5.087225E+02	5.087225E+02			
	133	TEMP	5.087225E+02					
	135	TEMP	5.087225E+02					
	137	TEMP	5.087225E+02	5.087225E+02				
	145	TEMP	5.089311E+02	6.285157E+02	6.285157E+02			
	150	TEMP	6.285157E+02					
	152	TEMP	6.285157E+02					
	160	TEMP	6.285157E+02	6.285157E+02				
	164	TEMP	5.089311E+02	5.089311E+02				
	172	TEMP	5.089311E+02					
	174	TEMP	5.089311E+02					
	177	TEMP	5.089311E+02	5.089311E+02				
	180	TEMP	5.089311E+02	6.319413E+02	6.319294E+02	6.319294E+02	6.319294E+02	
	188	TEMP	6.319294E+02					
	193	TEMP	6.319294E+02					
	195	TEMP	6.319294E+02	6.319294E+02	6.319294E+02	6.319294E+02	6.319294E+02	6.319413E+02
	201	TEMP	6.319413E+02	6.319413E+02	6.319413E+02	6.319413E+02	6.319413E+02	6.319413E+02
1	DIRECT_RADIATION_CONDUCTION_EMISSION_CONVECTION (T=0.001FT)					AUGUST 17, 2000	UAI/NASTRAN VERSION 20.1	PAGE
19	10.7 IN GAP FILLED BY AIR; K=0.016BTU/HR/FT/DEGF							
0								
	T E M P E R A T U R E V E C T O R							
	POINT ID.	TYPE	ID VALUE	ID+1 VALUE	ID+2 VALUE	ID+3 VALUE	ID+4 VALUE	ID+5 VALUE
	207	TEMP	6.319413E+02	6.319413E+02	6.319413E+02	6.319413E+02	6.319413E+02	6.319413E+02
	213	TEMP	6.319413E+02	6.319413E+02	6.319413E+02	6.319413E+02		
	219	TEMP	6.287362E+02	6.287362E+02	6.287362E+02	6.287362E+02	6.287362E+02	
	226	TEMP	6.287362E+02	6.287362E+02	6.287362E+02	6.287362E+02		
	231	TEMP	6.287362E+02	6.287362E+02				
	235	TEMP	6.287362E+02					
	241	TEMP	6.319294E+02					
	243	TEMP	6.319294E+02	6.319294E+02	6.319294E+02	6.319294E+02		
	248	TEMP	6.319294E+02	6.319294E+02	6.319294E+02			
	253	TEMP	6.287362E+02	6.287242E+02				
	257	TEMP	6.287242E+02	6.287242E+02	6.287242E+02	6.287242E+02		
	264	TEMP	6.287242E+02					
	267	TEMP	6.287242E+02	6.287242E+02	6.287242E+02	6.287242E+02		
	273	TEMP	6.287362E+02	6.287362E+02				
	277	TEMP	6.287362E+02					
	282	TEMP	6.287362E+02	6.287362E+02				
	292	TEMP	6.285157E+02					
	295	TEMP	6.285157E+02					
	297	TEMP	6.285157E+02	6.285157E+02	6.285157E+02	6.285157E+02	6.285157E+02	6.285157E+02
	303	TEMP	6.285157E+02					
	306	TEMP	6.285157E+02	6.285157E+02	6.287242E+02			
	313	TEMP	6.287242E+02	6.287242E+02				
	316	TEMP	6.287242E+02	6.287242E+02	6.287242E+02			
	323	TEMP	6.287242E+02	6.287242E+02				
	500	TEMP	5.786700E+02					

TABLE 3-28. FINITE ELEMENT METHOD-SIMULATED TEMPERATURES (°F) OF THE RIGHT WING OF SAMPLE PROBLEM 5

T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
172.3	172.3	169.1	169.1	168.9	49.3	49.1	49.0	45.9	45.8

3.6 SAMPLE PROBLEM 6.

In sample problem 5, the effects of infrared emission, in addition to the combined effects of conduction, convection, and direct solar radiation, were studied. The effects of infrared emission were carried out by removing the emission restriction on the wing surfaces of sample problem 4, by changing it from 0.0 to 0.9. The only external source of heat originated from direct solar radiation. Sample problem 6 incorporated the effects of external infrared radiation from the sky, in addition to the direct solar radiation. As in sample problems 1 through 5, THERMOD was set up to solve a complete aircraft problem, while the FEM method was confined to the wing model.

To ensure consistency with the FEM wing model, the wing in the THERMOD model was isolated from infrared radiation effects from the fuselage and tarmac surfaces by assigning emissivity values of 0.0. Unlike sample problem 5, the sky radiation was assigned an absolute temperature of 459.67°R, so as to incorporate the infrared radiation from the sky. The conduction and convection heat transfer modes were left intact, as in sample problem 5.

As in sample problem 5, the wings were further isolated from the tarmac effects by arbitrarily placing the wing at a height of 100.0 ft from the tarmac. In addition, the infrared reflection effects from the fuselage surface were reduced by maintaining a cabin height of 0.1 ft.

As in sample problem 5, the fillet effect was removed from the wing top by assigning a 0.0 value in the second field of input data set 6. See the THERMOD User's Reference Manual for further details on the input requirements for fillet effects. The THERMOD input file, input.dat, is illustrated in table 3-29.

TABLE 3-29. THERMOD INPUT FILE (input.dat) OF SAMPLE PROBLEM 6

198e-6,0.704
1, 0.020,125,0.300,0.30
2, 0.375,4.4,0.021,0.24
3, 0.020,125,0.300,0.30
4, 0.350,125,0.300,0.30
5, 10.70,0.067,0.016,0.24
6, 0.350,125,0.300,0.30
7, 0.020,125,0.300,0.30
8, 0.375,4.4,0.021,0.24
9, 0.020,125,0.300,0.30
10,0.030,125,0.300,0.30
11,0.375,4.4,0.021,0.24
12,0.020,125,0.300,0.30

TABLE 3-29. THERMOD INPUT FILE (input.dat) OF SAMPLE
 PROBLEM 6 (Continued)

13,0.250,10.,0.020,0.24
14,48.00,0.067,0.016,0.24
15,0.020,125,0.300,0.30
16,0.375,4.4,0.021,0.24
17,0.020,125,0.300,0.30
18,0.250,10.,0.020,0.24
19,30.00,0.067,0.016,0.24
20,5.000,10.,0.020,0.24
21,0.020,125,0.300,0.30
22,0.375,4.4,0.021,0.24
23,0.020,125,0.300,0.30
24,2.500,0.067,0.016,0.24
25,0.020,125,0.300,0.30
26,0.375,4.4,0.021,0.24
27,0.020,125,0.300,0.30
28,0.250,130,0.800,0.20
3.5,34.0,100.0,5.5,22.0,11.0,0.1,4.0,30,24,28
0.70,1.0,0.0,2.0,150,0.80,0.22
0.3,0.9,0.3,0.9,0.3,0.9,0.1,0.9
3,0.0
0.9,0.9,0.0,0.0
1.0,0.0,0.9,0.9,1.0,0.0,1.0,0.0
9
0,10,10,0,0,75,110,110,190,190
5,120,5,60,20,5,120,60,60
7
60 570.67 459.67 14 330 10 10 0.36
75 573.67 459.67 14 355 10 10 0.31
90 578.67 459.67 14 355 10 12 0.33
75 581.67 459.67 14 330 10 10 0.31
60 582.67 459.67 14 291 10 10 0.36
45 583.67 459.67 14 231 10 10 0.44
30 582.67 459.67 14 160 10 10 0.57
1,0.02,180,10
10,1.0,1.0
10,10,10,10,10,10,10,10,10,10,10,10,10,10,10,
10,10,10,10,10,10,10,10,10,10,10,10,10,10,
10,10,10,10,10,10,10,10,10,10,10,10,10,10,
10,10,10,10,10,10,10,10,10,10,10,10,10,10,
10,10,10,10,10,10,10,10,10,10,10,10,10,10,
10,10,10,10,10,10,10,10,10,10,10,10,10,10,
10,10,10,10,10,10,10,10,10,10,10,10,10,10,

Temperatures from the THERMOD run is shown in table 3-30, from which the steady-state temperatures T1, T2.....T10 (top to bottom of wing) for time period 3 were extracted and tabulated in table 3-31.

TABLE 3-30. THERMOD OUTPUT FILE (summary.dat) FOR SAMPLE PROBLEM 6

```

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
THERMOD:  A THERMAL PROGRAM FOR AIRCRAFTS
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
  
```

SUMMARY OF STEADY-STATE AND TRANSIENT ANALYSIS

(Temperatures are shown for all time periods)

TEMPERATURES AT THE END OF STEADY STATE ANALYSIS

	TEMPERATURES IN DEGREES FAHRENHEIT						
	1	2	3	4	5	6	7
1	171.1	189.7	195.5	186.1	166.2	137.9	108.3
2	171.1	189.7	195.5	186.1	166.2	137.9	108.3
3	169.0	187.2	192.8	183.8	164.3	136.8	107.9
4	168.9	187.1	192.8	183.8	164.3	136.8	107.9
5	168.8	187.0	192.7	183.6	164.2	136.7	107.9
6	88.8	91.1	94.1	95.4	95.2	94.8	93.3
7	88.7	91.0	93.9	95.2	95.1	94.8	93.3
8	88.7	90.9	93.9	95.2	95.1	94.7	93.2
9	86.5	88.4	91.2	92.9	93.3	93.6	92.9
10	86.5	88.4	91.2	92.8	93.3	93.6	92.9
11	171.1	189.7	195.5	186.1	166.2	137.9	108.3
12	171.1	189.7	195.5	186.1	166.2	137.9	108.3
13	168.9	187.2	192.8	183.8	164.3	136.8	107.9
14	168.9	187.1	192.8	183.8	164.3	136.8	107.9
15	168.8	187.0	192.7	183.6	164.2	136.7	107.9
16	88.8	91.1	94.1	95.4	95.2	94.8	93.3
17	88.7	91.0	93.9	95.2	95.1	94.8	93.3
18	88.7	90.9	93.9	95.2	95.1	94.7	93.2
19	86.5	88.4	91.2	92.9	93.3	93.6	92.9
20	86.5	88.4	91.2	92.8	93.3	93.6	92.9
21	276.7	310.4	322.3	304.0	268.2	219.2	169.9
22	276.5	310.3	322.1	303.8	268.1	219.1	169.9
23	242.5	264.0	274.0	259.4	236.0	200.8	161.5
24	242.4	263.8	273.8	259.2	235.8	200.7	161.5
25	218.6	231.4	240.1	228.1	213.4	187.9	155.7
26	185.0	192.9	200.7	191.5	182.5	165.1	141.4
27	143.9	149.0	155.3	152.9	149.4	142.2	131.2
28	143.9	149.0	155.3	152.9	149.4	142.2	131.2
29	140.9	145.9	152.0	150.1	147.1	140.6	130.4
30	140.9	145.9	152.0	150.1	147.1	140.6	130.4

TABLE 3-30. THERMOD OUTPUT FILE (summary.dat) FOR SAMPLE
 PROBLEM 6 (Continued)

31	128.1	132.2	137.8	138.0	136.7	133.5	127.2
32	128.1	132.2	137.8	138.0	136.7	133.5	127.2
33	128.1	132.2	137.8	138.0	136.7	133.5	127.2
34	128.1	132.2	137.8	138.0	136.7	133.5	127.2
35	115.2	118.5	123.6	125.9	126.4	126.3	124.0
36	115.2	118.5	123.6	125.9	126.4	126.3	124.0
37	112.3	115.3	120.4	123.2	124.0	124.7	123.3
38	112.2	115.3	120.4	123.2	124.0	124.7	123.3
39	218.4	180.1	132.7	182.8	216.8	225.1	206.3
40	218.4	180.2	132.9	182.9	216.7	225.0	206.2
41	202.0	186.0	165.4	186.7	200.0	196.5	176.2
42	202.0	186.0	165.5	186.8	200.0	196.4	176.1
43	190.5	190.1	188.2	189.5	188.3	176.4	155.1
44	181.2	184.4	188.2	184.1	180.0	168.0	149.3
45	158.2	161.3	165.5	163.7	161.3	153.5	140.6
46	158.0	161.2	165.4	163.6	161.2	153.5	140.6
47	125.1	128.2	132.9	134.5	134.5	132.8	128.3
48	124.9	128.0	132.7	134.3	134.3	132.7	128.2
49	123.9	128.0	133.1	130.2	127.0	120.3	110.7
50	126.1	130.5	135.7	132.6	129.1	121.8	111.6
51	188.8	207.4	215.7	208.8	191.6	168.5	144.8
52	183.9	201.5	209.6	203.3	187.3	165.7	143.4
53	188.8	207.4	215.7	208.8	191.6	168.5	144.8
Maximum Structural Temperature for Period 1 = 276.67 Occurring at Location 21							
Maximum Structural Temperature for Period 2 = 310.45 Occurring at Location 21							
Maximum Structural Temperature for Period 3 = 322.29 Occurring at Location 21							
Maximum Structural Temperature for Period 4 = 304.01 Occurring at Location 21							
Maximum Structural Temperature for Period 5 = 268.19 Occurring at Location 21							
Maximum Structural Temperature for Period 6 = 225.14 Occurring at Location 39							
Maximum Structural Temperature for Period 7 = 206.35 Occurring at Location 39							
Maximum Structural Temperature Over All 7 Periods = 322.29 Occurring at Location 21 at Period 3							

TABLE 3-30. THERMOD OUTPUT FILE (summary.dat) FOR SAMPLE
 PROBLEM 6 (Continued)

TEMPERATURES AT THE END OF TRANSIENT ANALYSIS

TEMPERATURES IN DEGREES FAHRENHEIT							
	1	2	3	4	5	6	7
1	117.0	121.7	126.8	128.5	127.3	125.4	121.6
2	118.4	123.4	128.6	130.0	128.3	125.7	121.2
3	164.7	181.8	187.5	179.2	161.2	135.7	109.0
4	166.1	183.7	189.3	180.7	162.3	136.0	108.7
5	167.4	185.3	191.0	182.1	163.1	136.2	108.4
6	89.4	91.7	94.7	96.0	95.9	95.4	93.9
7	89.8	92.1	95.1	96.5	96.4	96.1	94.6
8	90.4	92.7	95.8	97.2	97.1	96.8	95.3
9	108.2	111.1	115.8	118.6	119.6	120.5	119.5
10	108.8	111.6	116.4	119.3	120.2	121.2	120.2
11	117.0	121.7	126.8	128.5	127.3	125.4	121.6
12	118.4	123.4	128.6	130.0	128.3	125.7	121.2
13	164.7	181.8	187.5	179.2	161.2	135.7	109.0
14	166.1	183.6	189.3	180.7	162.3	136.0	108.7
15	167.4	185.3	191.0	182.1	163.1	136.2	108.4
16	89.4	91.7	94.7	96.0	95.9	95.4	93.9
17	89.8	92.1	95.1	96.5	96.4	96.1	94.6
18	90.4	92.7	95.8	97.2	97.1	96.8	95.3
19	108.2	111.1	115.8	118.6	119.6	120.5	119.5
20	108.8	111.6	116.4	119.3	120.2	121.2	120.2
21	125.0	130.7	136.2	137.4	135.3	132.0	126.9
22	127.1	132.9	138.5	139.5	137.0	133.2	127.5
23	189.3	201.7	208.5	201.7	189.3	169.4	145.5
24	191.0	203.5	210.4	203.4	190.7	170.4	146.0
25	188.7	198.6	205.0	197.8	187.4	168.9	145.2
26	167.6	174.4	180.2	175.2	168.6	155.4	137.2
27	144.1	149.2	155.5	153.0	149.5	142.3	131.1
28	143.2	148.3	154.5	152.3	148.9	142.0	131.1
29	140.9	145.9	152.0	150.2	147.1	140.6	130.5
30	140.9	145.8	151.9	150.1	147.0	140.6	130.4
31	127.8	131.9	137.5	137.8	136.6	133.5	127.2
32	127.8	131.9	137.5	137.8	136.6	133.5	127.2
33	127.8	131.9	137.5	137.8	136.6	133.5	127.2
34	127.8	131.9	137.5	137.8	136.6	133.5	127.2
35	114.7	117.9	123.1	125.6	126.1	126.3	124.0
36	114.6	117.8	123.0	125.5	126.1	126.3	124.0
37	111.2	114.2	119.2	122.2	123.2	124.1	123.1
38	111.1	114.1	119.1	122.1	123.1	124.1	123.0
39	116.3	117.5	120.0	125.2	127.6	128.9	126.9
40	117.9	118.9	121.2	126.4	128.9	130.0	127.7
41	163.4	159.9	154.8	162.9	166.6	161.7	148.3

TABLE 3-30. THERMOD OUTPUT FILE (summary.dat) FOR SAMPLE
 PROBLEM 6 (Continued)

42	164.8	161.4	156.2	164.2	167.7	162.5	148.8
43	166.2	168.2	169.0	170.1	168.3	159.2	143.7
44	162.8	166.0	169.0	168.0	165.3	156.4	142.0
45	152.2	154.1	156.2	157.5	156.6	150.4	139.0
46	150.9	152.7	154.8	156.3	155.6	149.6	138.6
47	117.0	118.3	121.2	125.9	128.1	129.1	127.0
48	115.9	117.2	120.0	124.9	127.2	128.4	126.6
49	113.1	116.3	121.1	123.1	123.3	122.7	120.0
50	115.2	118.6	123.5	125.0	124.9	123.6	120.3
51	306.1	348.1	361.3	339.6	295.0	235.5	177.6
52	288.9	327.5	340.0	320.5	279.9	225.7	172.8
53	306.1	348.1	361.3	339.6	295.0	235.5	177.6
Maximum Structural Temperature for Period 1 = 191.02 Occurring at Location 24							
Maximum Structural Temperature for Period 2 = 203.55 Occurring at Location 24							
Maximum Structural Temperature for Period 3 = 210.37 Occurring at Location 24							
Maximum Structural Temperature for Period 4 = 203.36 Occurring at Location 24							
Maximum Structural Temperature for Period 5 = 190.72 Occurring at Location 24							
Maximum Structural Temperature for Period 6 = 170.41 Occurring at Location 24							
Maximum Structural Temperature for Period 7 = 148.79 Occurring at Location 42							
Maximum Structural Temperature Over All 7 Periods =210.37 Occurring at Location 24 at Period 3							

TABLE 3-31. THERMOD-SIMULATED TEMPERATURES (°F) OF THE RIGHT WING OF
 SAMPLE PROBLEM 6

T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
195.5	195.5	192.8	192.8	192.7	94.1	93.9	93.9	91.2	91.2

As in the previous sample problems, a one-half wing structure was modeled using the UAI/NASTRAN FEM program. Instead of just two types of loads that were applied as in sample problem 5, three types of loads were applied to the model in sample problem 6. These loads, based on period 3 of the THERMOD input file, were direct solar radiation of 355 Btu/hr/ft², convection with a film coefficient of $h = 1.3433 \text{ (Btu/hr)/(ft}^2\text{°F)}$, and infrared sky radiation, based on a sky temperature of 0°F (459.67°R). The film coefficient was calculated internally by the THERMOD program and was based on the theory of airflow on a flat surface and a wind speed of 14 ft/sec. The solar radiation load was applied perpendicular to the top surface only, while both the top and bottom surfaces were subjected to the infrared sky radiation. Both the top and bottom surfaces were subjected to the convection loads. The ambient

temperature was maintained at 119°F (578.67°R). As in sample problem 5, the wings were assigned a nonzero emissivity value of 0.9, allowing the wings to emit and absorb infrared radiation. The effect of infrared radiation of the tarmac was neglected.

The FEM input file is shown in table 3-32. The ambient temperature of 119°F was imposed through the SPC case control card. This ambient temperature boundary condition was scalar in nature and was implemented through the SPOINT bulk data. The boundary elements, CHBDY, through which solar radiation and convection effects were applied, were modeled using the LINE boundary shape type, connecting the boundary nodes along the wing top and bottom. The SPACE designation in the CHBDY boundary element indicates that this element emits infrared radiation to a black body, in which case, no VIEW bulk data was required. Property cards PHBDY and the material cards MAT4, define absorptivity, emissivity, film coefficient, and conductivity values. Note that as in problem 5, the wing was assigned an emissivity of 0.9, as indicated in the PHBDY card. Load case 10 combines all three load cases: solar radiation on the top surface (indicated by QVECT) and the infrared sky radiation on the top and bottom surfaces (indicated by QBDY1). Because of the involvement of infrared radiation, through emission, this sample problem was rendered nonlinear. This required the Stefan-Boltzman constant, σ , and an initial set of temperatures, which were implemented through the PARAM and TEMPERATURE statements, respectively.

TABLE 3-32. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE PROBLEM 6

```
ID D:\Natha, MSC/N
APP HEAT
SOL 3
TIME 10000
CEND
  TITLE = dradiation_infrd_conduction_emission_convection (t=0.001ft)
  SUBTITLE = 10.7 in gap filled by air; k=0.016BTU/hr/ft/degF
  load = 10
  SPC = 1
  temperature(estimate)=2
  THERMAL= ALL
  FLUX = ALL
  SPCF = ALL
BEGIN BULK
$ *****
$   Written by : MSC/NASTRAN for Windows
$   Version    : 6.00
$   Translator  : UAI/NASTRAN
$   From Model : D:\Thermod_Validation\FEM\conduction_only.MOD
$   Date       : Mon Mar 06 11:27:09 2000
$ *****
$
PARAM, K6ROT, 100.
PARAM, MAXRATIO, 1.E+8
param, sigma, 1.714e-9
param, tabs, 0.0
param, maxit, 10
CORD2C          1          0          0.          0.          0.          0.          0.
1.+MSC/NC1
```

TABLE 3-32. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 6 (Continued)

+MSC/NC1	1.	0.	1.						
CORD2S	2		0	0.	0.	0.	0.	0.	0.
1.+MSC/NC2									
+MSC/NC2	1.	0.	1.						
\$									
\$ MSC/NASTRAN for Windows Property 1 : 0.02 in glass_outer_layer									
PSHELL	1	1	0.001	1		1			0.
\$ MSC/NASTRAN for Windows Property 2 : 3/8 in foam									
PSHELL	2	2	0.001	2		2			0.
\$ MSC/NASTRAN for Windows Property 3 : 0.02 in inner glass									
PSHELL	3	1	0.001	1		1			0.
\$ MSC/NASTRAN for Windows Property 4 : 0.350 in cap									
PSHELL	4	1	0.001	1		1			0.
\$ MSC/NASTRAN for Windows Property 5 : 10.70 in air space									
PSHELL	5	3	0.001	3		3			0.
\$ MSC/NASTRAN for Windows Property 6 : 0.350 in cap									
PSHELL	6	1	0.001	1		1			0.
\$ MSC/NASTRAN for Windows Property 7 : 0.02 in inner glass									
PSHELL	7	1	0.001	1		1			0.
\$ MSC/NASTRAN for Windows Property 8 : 3/8 in foam									
PSHELL	8	2	0.001	2		2			0.
\$ MSC/NASTRAN for Windows Property 9 : 0.02 in glass_outer_layer									
PSHELL	9	1	0.001	1		1			0.
\$ MSC/NASTRAN for Windows Material 1 : glass									
MAT4	1	0.3	1.						
\$ MSC/NASTRAN for Windows Material 2 : foam									
MAT4	2	0.021	1.						
\$ MSC/NASTRAN for Windows Material 3 : air									
MAT4	3	0.016	1.						
\$									
chbdy,1001,100,line,200,201,,,space,500,500,,,0.0,1.0,0.0									
chbdy,1002,100,line,201,202,,,space,500,500,,,0.0,1.0,0.0									
chbdy,1003,100,line,202,203,,,space,500,500,,,0.0,1.0,0.0									
chbdy,1004,100,line,203,204,,,space,500,500,,,0.0,1.0,0.0									
chbdy,1005,100,line,204,205,,,space,500,500,,,0.0,1.0,0.0									
chbdy,1006,100,line,205,206,,,space,500,500,,,0.0,1.0,0.0									
chbdy,1007,100,line,206,207,,,space,500,500,,,0.0,1.0,0.0									
chbdy,1008,100,line,207,208,,,space,500,500,,,0.0,1.0,0.0									
chbdy,1009,100,line,208,209,,,space,500,500,,,0.0,1.0,0.0									
chbdy,1010,100,line,209,210,,,space,500,500,,,0.0,1.0,0.0									
chbdy,1011,100,line,210,211,,,space,500,500,,,0.0,1.0,0.0									
chbdy,1012,100,line,211,212,,,space,500,500,,,0.0,1.0,0.0									
chbdy,1013,100,line,212,213,,,space,500,500,,,0.0,1.0,0.0									
chbdy,1014,100,line,213,214,,,space,500,500,,,0.0,1.0,0.0									
chbdy,1015,100,line,214,215,,,space,500,500,,,0.0,1.0,0.0									
chbdy,1016,100,line,215,216,,,space,500,500,,,0.0,1.0,0.0									
chbdy,1017,100,line,216,181,,,space,500,500,,,0.0,1.0,0.0									
phbdy,100,200,0.001,0.9,0.9									
mat4,200,1.3433									
spoint,500									
spc,1,500,1,578.67									
\$									
\$ Direct radiation load from a distant source: load case 1:									
qvect,1,355.0,0.0,-1.0,0.0,1001,thru,1017									
\$									
\$ Infrared radiation from the sky: load case 2 (applied to the top)									
\$ and load case 3 (applied to the bottom)									

TABLE 3-32. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 6 (Continued)

```
$ Q0=sigma*(T^4)=1.714e-9*(459.67)^4
$ =76.5236BTU/hr/ft^2: assuming sky temp.=0degF=459.67 degR.
Qbdy1,2,76.5236,1001,thru,1017
qbdy1,3,76.5236,2001,thru,2017
$ combine the above three load cases 1, 2 and 3 into load case 10:
load,10,1.0,1.0,1.0,1,1.0,2,1.0,3
$
chbdy,2001,300,line,20,21,, ,space,500,500,, ,0.0,-1.0,0.0
chbdy,2002,300,line,21,22,, ,space,500,500,, ,0.0,-1.0,0.0
chbdy,2003,300,line,22,23,, ,space,500,500,, ,0.0,-1.0,0.0
chbdy,2004,300,line,23,24,, ,space,500,500,, ,0.0,-1.0,0.0
chbdy,2005,300,line,24,25,, ,space,500,500,, ,0.0,-1.0,0.0
chbdy,2006,300,line,25,26,, ,space,500,500,, ,0.0,-1.0,0.0
chbdy,2007,300,line,26,27,, ,space,500,500,, ,0.0,-1.0,0.0
chbdy,2008,300,line,27,28,, ,space,500,500,, ,0.0,-1.0,0.0
chbdy,2009,300,line,28,29,, ,space,500,500,, ,0.0,-1.0,0.0
chbdy,2010,300,line,29,30,, ,space,500,500,, ,0.0,-1.0,0.0
chbdy,2011,300,line,30,31,, ,space,500,500,, ,0.0,-1.0,0.0
chbdy,2012,300,line,31,32,, ,space,500,500,, ,0.0,-1.0,0.0
chbdy,2013,300,line,32,33,, ,space,500,500,, ,0.0,-1.0,0.0
chbdy,2014,300,line,33,34,, ,space,500,500,, ,0.0,-1.0,0.0
chbdy,2015,300,line,34,35,, ,space,500,500,, ,0.0,-1.0,0.0
chbdy,2016,300,line,35,36,, ,space,500,500,, ,0.0,-1.0,0.0
chbdy,2017,300,line,36,1,, ,space,500,500,, ,0.0,-1.0,0.0
phbdy,300,400,0.001,0.9,0.9
mat4,400,1.3433
$
temp,2,1,578.67,3,578.67,5,578.67
temp,2,7,578.67,9,578.67,10,578.67
temp,2,11,578.67,12,578.67,14,578.67
temp,2,15,578.67,18,578.67,20,578.67
temp,2,21,578.67,22,578.67,23,578.67
temp,2,24,578.67,25,578.67,26,578.67
temp,2,27,578.67,28,578.67,29,578.67
temp,2,30,578.67,31,578.67,32,578.67
temp,2,33,578.67,34,578.67,35,578.67
temp,2,36,578.67,37,578.67,41,578.67
temp,2,44,578.67,45,578.67,47,578.67
temp,2,48,578.67,50,578.67,51,578.67
temp,2,52,578.67,53,578.67,54,578.67
temp,2,55,578.67,56,578.67,58,578.67
temp,2,59,578.67,62,578.67,67,578.67
temp,2,69,578.67,71,578.67,73,578.67
temp,2,75,578.67,76,578.67,77,578.67
temp,2,78,578.67,79,578.67,80,578.67
temp,2,83,578.67,85,578.67,87,578.67
temp,2,88,578.67,98,578.67,101,578.67
temp,2,104,578.67,105,578.67,107,578.67
temp,2,108,578.67,109,578.67,112,578.67
temp,2,115,578.67,116,578.67,118,578.67
temp,2,120,578.67,121,578.67,122,578.67
temp,2,123,578.67,124,578.67,125,578.67
temp,2,128,578.67,129,578.67,130,578.67
temp,2,133,578.67,135,578.67,137,578.67
temp,2,138,578.67,145,578.67,146,578.67
temp,2,147,578.67,150,578.67,152,578.67
temp,2,160,578.67,161,578.67,164,578.67
```

TABLE 3-32. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 6 (Continued)

temp, 2, 165, 578.67, 172, 578.67, 174, 578.67					
temp, 2, 177, 578.67, 178, 578.67, 180, 578.67					
temp, 2, 181, 578.67, 182, 578.67, 183, 578.67					
temp, 2, 184, 578.67, 188, 578.67, 193, 578.67					
temp, 2, 195, 578.67, 196, 578.67, 197, 578.67					
temp, 2, 198, 578.67, 199, 578.67, 200, 578.67					
temp, 2, 201, 578.67, 202, 578.67, 203, 578.67					
temp, 2, 204, 578.67, 205, 578.67, 206, 578.67					
temp, 2, 207, 578.67, 208, 578.67, 209, 578.67					
temp, 2, 210, 578.67, 211, 578.67, 212, 578.67					
temp, 2, 213, 578.67, 214, 578.67, 215, 578.67					
temp, 2, 216, 578.67, 219, 578.67, 220, 578.67					
temp, 2, 221, 578.67, 222, 578.67, 223, 578.67					
temp, 2, 226, 578.67, 227, 578.67, 228, 578.67					
temp, 2, 229, 578.67, 231, 578.67, 232, 578.67					
temp, 2, 235, 578.67, 241, 578.67, 243, 578.67					
temp, 2, 244, 578.67, 245, 578.67, 246, 578.67					
temp, 2, 248, 578.67, 249, 578.67, 250, 578.67					
temp, 2, 253, 578.67, 254, 578.67, 257, 578.67					
temp, 2, 258, 578.67, 259, 578.67, 260, 578.67					
temp, 2, 264, 578.67, 267, 578.67, 268, 578.67					
temp, 2, 269, 578.67, 270, 578.67, 273, 578.67					
temp, 2, 274, 578.67, 277, 578.67, 282, 578.67					
temp, 2, 283, 578.67, 292, 578.67, 293, 578.67					
temp, 2, 295, 578.67, 297, 578.67, 298, 578.67					
temp, 2, 299, 578.67, 300, 578.67, 301, 578.67					
temp, 2, 302, 578.67, 303, 578.67, 306, 578.67					
temp, 2, 307, 578.67, 308, 578.67, 313, 578.67					
temp, 2, 314, 578.67, 316, 578.67, 317, 578.67					
temp, 2, 318, 578.67, 323, 578.67, 324, 578.67					
temp, 2, 500, 578.67					
\$					
GRID	1	0	0.	0.	0.
GRID	3	0	1.1.667E-3	0.	0
GRID	5	0	3.1.667E-3	0.	0
GRID	7	0	5.1.667E-3	0.	0
GRID	9	0	7.1.667E-3	0.	0
GRID	10	0	8.1.667E-3	0.	0
GRID	11	0	9.1.667E-3	0.	0
GRID	12	0	10.1.667E-3	0.	0
GRID	14	0	12.1.667E-3	0.	0
GRID	15	0	13.1.667E-3	0.	0
GRID	18	0	16.1.667E-3	0.	0
GRID	20	0	17.	0.	0
GRID	21	0	16.	0.	0
GRID	22	0	15.	0.	0
GRID	23	0	14.	0.	0
GRID	24	0	13.	0.	0
GRID	25	0	12.	0.	0
GRID	26	0	11.	0.	0
GRID	27	0	10.	0.	0
GRID	28	0	9.	0.	0
GRID	29	0	8.	0.	0
GRID	30	0	7.	0.	0
GRID	31	0	6.	0.	0
GRID	32	0	5.	0.	0
GRID	33	0	4.	0.	0

TABLE 3-32. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 6 (Continued)

GRID	34	0	3.	0.	0.	0
GRID	35	0	2.	0.	0.	0
GRID	36	0	1.	0.	0.	0
GRID	37	0	0.1.667E-3	0.	0.	0
GRID	41	0	3.0.032917	0.	0.	0
GRID	44	0	6.0.032917	0.	0.	0
GRID	45	0	7.0.032917	0.	0.	0
GRID	47	0	9.0.032917	0.	0.	0
GRID	48	0	10.0.032917	0.	0.	0
GRID	50	0	12.0.032917	0.	0.	0
GRID	51	0	13.0.032917	0.	0.	0
GRID	52	0	14.0.032917	0.	0.	0
GRID	53	0	15.0.032917	0.	0.	0
GRID	54	0	16.0.032917	0.	0.	0
GRID	55	0	17.0.032917	0.	0.	0
GRID	56	0	17.1.667E-3	0.	0.	0
GRID	58	0	15.1.667E-3	0.	0.	0
GRID	59	0	14.1.667E-3	0.	0.	0
GRID	62	0	11.1.667E-3	0.	0.	0
GRID	67	0	6.1.667E-3	0.	0.	0
GRID	69	0	4.1.667E-3	0.	0.	0
GRID	71	0	2.1.667E-3	0.	0.	0
GRID	73	0	0.0.032917	0.	0.	0
GRID	75	0	1.0.034583	0.	0.	0
GRID	76	0	2.0.034583	0.	0.	0
GRID	77	0	3.0.034583	0.	0.	0
GRID	78	0	4.0.034583	0.	0.	0
GRID	79	0	5.0.034583	0.	0.	0
GRID	80	0	6.0.034583	0.	0.	0
GRID	83	0	9.0.034583	0.	0.	0
GRID	85	0	11.0.034583	0.	0.	0
GRID	87	0	13.0.034583	0.	0.	0
GRID	88	0	14.0.034583	0.	0.	0
GRID	98	0	11.0.032917	0.	0.	0
GRID	101	0	8.0.032917	0.	0.	0
GRID	104	0	5.0.032917	0.	0.	0
GRID	105	0	4.0.032917	0.	0.	0
GRID	107	0	2.0.032917	0.	0.	0
GRID	108	0	1.0.032917	0.	0.	0
GRID	109	0	0.0.034583	0.	0.	0
GRID	112	0	2. 0.06375	0.	0.	0
GRID	115	0	5. 0.06375	0.	0.	0
GRID	116	0	6. 0.06375	0.	0.	0
GRID	118	0	8. 0.06375	0.	0.	0
GRID	120	0	10. 0.06375	0.	0.	0
GRID	121	0	11. 0.06375	0.	0.	0
GRID	122	0	12. 0.06375	0.	0.	0
GRID	123	0	13. 0.06375	0.	0.	0
GRID	124	0	14. 0.06375	0.	0.	0
GRID	125	0	15. 0.06375	0.	0.	0
GRID	128	0	17.0.034583	0.	0.	0
GRID	129	0	16.0.034583	0.	0.	0
GRID	130	0	15.0.034583	0.	0.	0
GRID	133	0	12.0.034583	0.	0.	0
GRID	135	0	10.0.034583	0.	0.	0
GRID	137	0	8.0.034583	0.	0.	0
GRID	138	0	7.0.034583	0.	0.	0

TABLE 3-32. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 6 (Continued)

GRID	145	0	0.	0.06375	0.	0
GRID	146	0	0.	0.95542	0.	0
GRID	147	0	1.	0.95542	0.	0
GRID	150	0	4.	0.95542	0.	0
GRID	152	0	6.	0.95542	0.	0
GRID	160	0	14.	0.95542	0.	0
GRID	161	0	15.	0.95542	0.	0
GRID	164	0	17.	0.06375	0.	0
GRID	165	0	16.	0.06375	0.	0
GRID	172	0	9.	0.06375	0.	0
GRID	174	0	7.	0.06375	0.	0
GRID	177	0	4.	0.06375	0.	0
GRID	178	0	3.	0.06375	0.	0
GRID	180	0	1.	0.06375	0.	0
GRID	181	0	0.	1.01917	0.	0
GRID	182	0	0.	1.0175	0.	0
GRID	183	0	1.	1.0175	0.	0
GRID	184	0	2.	1.0175	0.	0
GRID	188	0	6.	1.0175	0.	0
GRID	193	0	11.	1.0175	0.	0
GRID	195	0	13.	1.0175	0.	0
GRID	196	0	14.	1.0175	0.	0
GRID	197	0	15.	1.0175	0.	0
GRID	198	0	16.	1.0175	0.	0
GRID	199	0	17.	1.0175	0.	0
GRID	200	0	17.	1.01917	0.	0
GRID	201	0	16.	1.01917	0.	0
GRID	202	0	15.	1.01917	0.	0
GRID	203	0	14.	1.01917	0.	0
GRID	204	0	13.	1.01917	0.	0
GRID	205	0	12.	1.01917	0.	0
GRID	206	0	11.	1.01917	0.	0
GRID	207	0	10.	1.01917	0.	0
GRID	208	0	9.	1.01917	0.	0
GRID	209	0	8.	1.01917	0.	0
GRID	210	0	7.	1.01917	0.	0
GRID	211	0	6.	1.01917	0.	0
GRID	212	0	5.	1.01917	0.	0
GRID	213	0	4.	1.01917	0.	0
GRID	214	0	3.	1.01917	0.	0
GRID	215	0	2.	1.01917	0.	0
GRID	216	0	1.	1.01917	0.	0
GRID	219	0	1.	0.98625	0.	0
GRID	220	0	2.	0.98625	0.	0
GRID	221	0	3.	0.98625	0.	0
GRID	222	0	4.	0.98625	0.	0
GRID	223	0	5.	0.98625	0.	0
GRID	226	0	8.	0.98625	0.	0
GRID	227	0	9.	0.98625	0.	0
GRID	228	0	10.	0.98625	0.	0
GRID	229	0	11.	0.98625	0.	0
GRID	231	0	13.	0.98625	0.	0
GRID	232	0	14.	0.98625	0.	0
GRID	235	0	17.	0.98625	0.	0
GRID	241	0	12.	1.0175	0.	0
GRID	243	0	10.	1.0175	0.	0
GRID	244	0	9.	1.0175	0.	0

TABLE 3-32. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 6 (Continued)

GRID	245	0	8.	1.0175	0.	0
GRID	246	0	7.	1.0175	0.	0
GRID	248	0	5.	1.0175	0.	0
GRID	249	0	4.	1.0175	0.	0
GRID	250	0	3.	1.0175	0.	0
GRID	253	0	0.	0.98625	0.	0
GRID	254	0	0.	0.98458	0.	0
GRID	257	0	3.	0.98458	0.	0
GRID	258	0	4.	0.98458	0.	0
GRID	259	0	5.	0.98458	0.	0
GRID	260	0	6.	0.98458	0.	0
GRID	264	0	10.	0.98458	0.	0
GRID	267	0	13.	0.98458	0.	0
GRID	268	0	14.	0.98458	0.	0
GRID	269	0	15.	0.98458	0.	0
GRID	270	0	16.	0.98458	0.	0
GRID	273	0	16.	0.98625	0.	0
GRID	274	0	15.	0.98625	0.	0
GRID	277	0	12.	0.98625	0.	0
GRID	282	0	7.	0.98625	0.	0
GRID	283	0	6.	0.98625	0.	0
GRID	292	0	2.	0.95542	0.	0
GRID	293	0	3.	0.95542	0.	0
GRID	295	0	5.	0.95542	0.	0
GRID	297	0	7.	0.95542	0.	0
GRID	298	0	8.	0.95542	0.	0
GRID	299	0	9.	0.95542	0.	0
GRID	300	0	10.	0.95542	0.	0
GRID	301	0	11.	0.95542	0.	0
GRID	302	0	12.	0.95542	0.	0
GRID	303	0	13.	0.95542	0.	0
GRID	306	0	16.	0.95542	0.	0
GRID	307	0	17.	0.95542	0.	0
GRID	308	0	17.	0.98458	0.	0
GRID	313	0	12.	0.98458	0.	0
GRID	314	0	11.	0.98458	0.	0
GRID	316	0	9.	0.98458	0.	0
GRID	317	0	8.	0.98458	0.	0
GRID	318	0	7.	0.98458	0.	0
GRID	323	0	2.	0.98458	0.	0
GRID	324	0	1.	0.98458	0.	0
CQUAD4	1	9	1	37	3	36
CQUAD4	2	9	36	3	71	35
CQUAD4	3	9	35	71	5	34
CQUAD4	4	9	34	5	69	33
CQUAD4	5	9	33	69	7	32
CQUAD4	6	9	32	7	67	31
CQUAD4	7	9	31	67	9	30
CQUAD4	8	9	30	9	10	29
CQUAD4	9	9	29	10	11	28
CQUAD4	10	9	28	11	12	27
CQUAD4	11	9	27	12	62	26
CQUAD4	12	9	26	62	14	25
CQUAD4	13	9	25	14	15	24
CQUAD4	14	9	24	15	59	23
CQUAD4	15	9	23	59	58	22
CQUAD4	16	9	22	58	18	21

TABLE 3-32. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 6 (Continued)

CQUAD4	17	9	21	18	56	20
CQUAD4	18	8	37	73	108	3
CQUAD4	19	8	3	108	107	71
CQUAD4	20	8	71	107	41	5
CQUAD4	21	8	5	41	105	69
CQUAD4	22	8	69	105	104	7
CQUAD4	23	8	7	104	44	67
CQUAD4	24	8	67	44	45	9
CQUAD4	25	8	9	45	101	10
CQUAD4	26	8	10	101	47	11
CQUAD4	27	8	11	47	48	12
CQUAD4	28	8	12	48	98	62
CQUAD4	29	8	62	98	50	14
CQUAD4	30	8	14	50	51	15
CQUAD4	31	8	15	51	52	59
CQUAD4	32	8	59	52	53	58
CQUAD4	33	8	58	53	54	18
CQUAD4	34	8	18	54	55	56
CQUAD4	35	7	73	109	75	108
CQUAD4	36	7	108	75	76	107
CQUAD4	37	7	107	76	77	41
CQUAD4	38	7	41	77	78	105
CQUAD4	39	7	105	78	79	104
CQUAD4	40	7	104	79	80	44
CQUAD4	41	7	44	80	138	45
CQUAD4	42	7	45	138	137	101
CQUAD4	43	7	101	137	83	47
CQUAD4	44	7	47	83	135	48
CQUAD4	45	7	48	135	85	98
CQUAD4	46	7	98	85	133	50
CQUAD4	47	7	50	133	87	51
CQUAD4	48	7	51	87	88	52
CQUAD4	49	7	52	88	130	53
CQUAD4	50	7	53	130	129	54
CQUAD4	51	7	54	129	128	55
CQUAD4	52	6	109	145	180	75
CQUAD4	53	6	75	180	112	76
CQUAD4	54	6	76	112	178	77
CQUAD4	55	6	77	178	177	78
CQUAD4	56	6	78	177	115	79
CQUAD4	57	6	79	115	116	80
CQUAD4	58	6	80	116	174	138
CQUAD4	59	6	138	174	118	137
CQUAD4	60	6	137	118	172	83
CQUAD4	61	6	83	172	120	135
CQUAD4	62	6	135	120	121	85
CQUAD4	63	6	85	121	122	133
CQUAD4	64	6	133	122	123	87
CQUAD4	65	6	87	123	124	88
CQUAD4	66	6	88	124	125	130
CQUAD4	67	6	130	125	165	129
CQUAD4	68	6	129	165	164	128
CQUAD4	69	5	145	146	147	180
CQUAD4	70	5	180	147	292	112
CQUAD4	71	5	112	292	293	178
CQUAD4	72	5	178	293	150	177
CQUAD4	73	5	177	150	295	115

TABLE 3-32. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE
 PROBLEM 6 (Continued)

CQUAD4	74	5	115	295	152	116
CQUAD4	75	5	116	152	297	174
CQUAD4	76	5	174	297	298	118
CQUAD4	77	5	118	298	299	172
CQUAD4	78	5	172	299	300	120
CQUAD4	79	5	120	300	301	121
CQUAD4	80	5	121	301	302	122
CQUAD4	81	5	122	302	303	123
CQUAD4	82	5	123	303	160	124
CQUAD4	83	5	124	160	161	125
CQUAD4	84	5	125	161	306	165
CQUAD4	85	5	165	306	307	164
CQUAD4	86	1	182	181	216	183
CQUAD4	87	1	183	216	215	184
CQUAD4	88	1	184	215	214	250
CQUAD4	89	1	250	214	213	249
CQUAD4	90	1	249	213	212	248
CQUAD4	91	1	248	212	211	188
CQUAD4	92	1	188	211	210	246
CQUAD4	93	1	246	210	209	245
CQUAD4	94	1	245	209	208	244
CQUAD4	95	1	244	208	207	243
CQUAD4	96	1	243	207	206	193
CQUAD4	97	1	193	206	205	241
CQUAD4	98	1	241	205	204	195
CQUAD4	99	1	195	204	203	196
CQUAD4	100	1	196	203	202	197
CQUAD4	101	1	197	202	201	198
CQUAD4	102	1	198	201	200	199
CQUAD4	103	2	253	182	183	219
CQUAD4	104	2	219	183	184	220
CQUAD4	105	8	220	184	250	221
CQUAD4	106	2	221	250	249	222
CQUAD4	107	2	222	249	248	223
CQUAD4	108	2	223	248	188	283
CQUAD4	109	2	283	188	246	282
CQUAD4	110	2	282	246	245	226
CQUAD4	111	2	226	245	244	227
CQUAD4	112	2	227	244	243	228
CQUAD4	113	2	228	243	193	229
CQUAD4	114	2	229	193	241	277
CQUAD4	115	2	277	241	195	231
CQUAD4	116	2	231	195	196	232
CQUAD4	117	2	232	196	197	274
CQUAD4	118	2	274	197	198	273
CQUAD4	119	2	273	198	199	235
CQUAD4	120	3	254	253	219	324
CQUAD4	121	3	324	219	220	323
CQUAD4	122	3	323	220	221	257
CQUAD4	123	3	257	221	222	258
CQUAD4	124	3	258	222	223	259
CQUAD4	125	3	259	223	283	260
CQUAD4	126	3	260	283	282	318
CQUAD4	127	3	318	282	226	317
CQUAD4	128	3	317	226	227	316
CQUAD4	129	3	316	227	228	264
CQUAD4	130	3	264	228	229	314

TABLE 3-32. FINITE ELEMENT METHOD INPUT FILE OF SAMPLE PROBLEM 6 (Continued)

CQUAD4	131	3	314	229	277	313
CQUAD4	132	3	313	277	231	267
CQUAD4	133	3	267	231	232	268
CQUAD4	134	3	268	232	274	269
CQUAD4	135	3	269	274	273	270
CQUAD4	136	3	270	273	235	308
CQUAD4	137	4	146	254	324	147
CQUAD4	138	4	147	324	323	292
CQUAD4	139	4	292	323	257	293
CQUAD4	140	4	293	257	258	150
CQUAD4	141	4	150	258	259	295
CQUAD4	142	4	295	259	260	152
CQUAD4	143	4	152	260	318	297
CQUAD4	144	4	297	318	317	298
CQUAD4	145	4	298	317	316	299
CQUAD4	146	4	299	316	264	300
CQUAD4	147	4	300	264	314	301
CQUAD4	148	4	301	314	313	302
CQUAD4	149	4	302	313	267	303
CQUAD4	150	4	303	267	268	160
CQUAD4	151	4	160	268	269	161
CQUAD4	152	4	161	269	270	306
CQUAD4	153	4	306	270	308	307
ENDDATA						

The results of the one-half FEM wing model analysis are presented in figure 3-6 in the form of a contour plot. The contours appear to be indicative of the temperature profile of the THERMOD analysis. The top surface, which was subjected to both direct solar radiation and infrared sky radiation, is certainly much hotter than the bottom surface, which was subjected to the infrared sky radiation only. The temperatures are higher than those found in figure 3-5 of sample problem 5 because of the inclusion of both the direct solar radiation and infrared sky radiation.



Note that the temperatures are in absolute terms ($^{\circ}\text{R}$).

FIGURE 3-6. CONTOUR PLOT OF THE FEM OUTPUT OF SAMPLE PROBLEM 6

The truncated FEM output file is shown in table 3-33. The nodes, from top to bottom of the wing, located at mid-span of the model are indicated by nodes 209, 245, 226, 317, 298, 118, 137, 101, 10, and 29. The temperatures of these nodes were extracted from this file and are reproduced in table 3-34. Mid-span was chosen because it was sufficiently far away from any edge effects that might be present. Note that the temperatures in table 3-34 were reduced from absolute temperatures (°R) to °F.

TABLE 3-33. TRUNCATED FEM OUTPUT FILE OF SAMPLE PROBLEM 6

1	DRADIATION_INFRD_CONDUCTION_EMISSION_CONVECTION (T=0.001FT)				AUGUST 21, 2000		UAI/NASTRAN VERSION 20.1		PAGE
18	10.7 IN GAP FILLED BY AIR; K=0.016BTU/HR/FT/DEGF								
0									
	T E M P E R A T U R E V E C T O R								
	POINT ID.	TYPE	ID VALUE	ID+1 VALUE	ID+2 VALUE	ID+3 VALUE	ID+4 VALUE	ID+5 VALUE	
	1	TEMP	5.397126E+02						
	3	TEMP	5.397238E+02						
	5	TEMP	5.397238E+02						
	7	TEMP	5.397238E+02						
	9	TEMP	5.397238E+02						
	14	TEMP	5.397238E+02	5.397238E+02	5.397238E+02	5.397238E+02			
	18	TEMP	5.397238E+02						
	20	TEMP	5.397126E+02	5.397126E+02	5.397126E+02	5.397126E+02	5.397126E+02	5.397126E+02	
	26	TEMP	5.397126E+02	5.397126E+02	5.397126E+02	5.397126E+02	5.397126E+02	5.397126E+02	
	32	TEMP	5.397126E+02	5.397126E+02	5.397126E+02	5.397126E+02	5.397126E+02	5.397238E+02	
	41	TEMP	5.427075E+02						
	44	TEMP	5.427075E+02	5.427075E+02					
	47	TEMP	5.427075E+02	5.427075E+02					
	50	TEMP	5.427075E+02	5.427075E+02	5.427075E+02	5.427075E+02	5.427075E+02	5.427075E+02	
	56	TEMP	5.397238E+02						
	58	TEMP	5.397238E+02	5.397238E+02					
	62	TEMP	5.397238E+02						
	67	TEMP	5.397238E+02						
	69	TEMP	5.397238E+02						
	71	TEMP	5.397238E+02						
	73	TEMP	5.427075E+02						
	75	TEMP	5.427186E+02	5.427186E+02	5.427186E+02	5.427186E+02	5.427186E+02	5.427186E+02	
	83	TEMP	5.427186E+02						
	85	TEMP	5.427186E+02						
	87	TEMP	5.427186E+02	5.427186E+02					
	98	TEMP	5.427075E+02						
	101	TEMP	5.427075E+02						
	104	TEMP	5.427075E+02	5.427075E+02					
	107	TEMP	5.427075E+02	5.427075E+02	5.427186E+02				
	112	TEMP	5.429136E+02						
	115	TEMP	5.429136E+02	5.429136E+02					
	118	TEMP	5.429136E+02						
	120	TEMP	5.429136E+02	5.429136E+02	5.429136E+02	5.429136E+02	5.429136E+02	5.429136E+02	
	128	TEMP	5.427186E+02	5.427186E+02	5.427186E+02				
	133	TEMP	5.427186E+02						
	135	TEMP	5.427186E+02						
	137	TEMP	5.427186E+02	5.427186E+02					
	145	TEMP	5.429136E+02	6.546570E+02	6.546570E+02				
	150	TEMP	6.546570E+02						
	152	TEMP	6.546570E+02						
	160	TEMP	6.546570E+02	6.546570E+02					
	164	TEMP	5.429136E+02	5.429136E+02					
	172	TEMP	5.429136E+02						
	174	TEMP	5.429136E+02						
	177	TEMP	5.429136E+02	5.429136E+02					
	180	TEMP	5.429136E+02	6.578580E+02	6.578469E+02	6.578469E+02	6.578469E+02		
	188	TEMP	6.578469E+02						
	193	TEMP	6.578469E+02						
	195	TEMP	6.578469E+02	6.578469E+02	6.578469E+02	6.578469E+02	6.578469E+02	6.578580E+02	
	201	TEMP	6.578580E+02	6.578580E+02	6.578580E+02	6.578580E+02	6.578580E+02	6.578580E+02	
1	DRADIATION_INFRD_CONDUCTION_EMISSION_CONVECTION (T=0.001FT)				AUGUST 21, 2000		UAI/NASTRAN VERSION 20.1		PAGE
19	10.7 IN GAP FILLED BY AIR; K=0.016BTU/HR/FT/DEGF								
0									
	T E M P E R A T U R E V E C T O R								
	POINT ID.	TYPE	ID VALUE	ID+1 VALUE	ID+2 VALUE	ID+3 VALUE	ID+4 VALUE	ID+5 VALUE	
	207	TEMP	6.578580E+02	6.578580E+02	6.578580E+02	6.578580E+02	6.578580E+02	6.578580E+02	
	213	TEMP	6.578580E+02	6.578580E+02	6.578580E+02	6.578580E+02			
	219	TEMP	6.548631E+02	6.548631E+02	6.548631E+02	6.548631E+02	6.548631E+02		
	226	TEMP	6.548631E+02	6.548631E+02	6.548631E+02	6.548631E+02			
	231	TEMP	6.548631E+02	6.548631E+02					
	235	TEMP	6.548631E+02						
	241	TEMP	6.578469E+02						
	243	TEMP	6.578469E+02	6.578469E+02	6.578469E+02	6.578469E+02			
	248	TEMP	6.578469E+02	6.578469E+02	6.578469E+02				
	253	TEMP	6.548631E+02	6.548631E+02	6.548519E+02				
	257	TEMP	6.548519E+02	6.548519E+02	6.548519E+02	6.548519E+02			
	264	TEMP	6.548519E+02						
	267	TEMP	6.548519E+02	6.548519E+02	6.548519E+02	6.548519E+02			
	273	TEMP	6.548631E+02	6.548631E+02					

TABLE 3-33. TRUNCATED FEM OUTPUT FILE OF SAMPLE PROBLEM 6 (Continued)

277	TEMP	6.548631E+02						
282	TEMP	6.548631E+02	6.548631E+02					
292	TEMP	6.546570E+02	6.546570E+02					
295	TEMP	6.546570E+02						
297	TEMP	6.546570E+02	6.546570E+02	6.546570E+02	6.546570E+02	6.546570E+02	6.546570E+02	
303	TEMP	6.546570E+02						
306	TEMP	6.546570E+02	6.546570E+02	6.548519E+02				
313	TEMP	6.548519E+02	6.548519E+02					
316	TEMP	6.548519E+02	6.548519E+02	6.548519E+02				
323	TEMP	6.548519E+02	6.548519E+02					
500	TEMP	5.786700E+02						

When compared, the temperatures of tables 3-31 and 3-34 show good agreement at the wing top. However, there is a significant difference in the bottom wing temperatures. This is due to the fact that the wing bottom in the FEM model was subjected to the sky radiation only, while the wing bottom in the THERMOD model was subjected to both the sky radiation and the tarmac radiation. In addition, the THERMOD program used control volumes and view factors in modeling the radiation effects, while the sample FEM model radiated heat to space, whereby the temperature-increasing effects of multiple reflections within an enclosure was neglected. These are the reasons why the wing bottom temperatures of the FEM model are lower than the THERMOD model, as comparison of tables 3-31 and 3-34 shows. In any case, sample problem 6 validates the radiation, conduction, and convection capabilities of THERMOD program, including infrared emission and absorption.

TABLE 3-34. FINITE ELEMENT METHOD-SIMULATED TEMPERATURES (°F) OF THE RIGHT WING OF SAMPLE PROBLEM 6

T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
198.2	198.2	195.2	195.2	195.0	83.2	83.0	83.0	80.0	80.0

4. SAMPLE PROBLEM.

This section presents a complete thermal solution for an aircraft. The objective was to determine the MOL temperature of the aircraft painted with a particular paint scheme and subjected to a given thermal environment. The solution involved both the steady-state and the transient analysis procedures. The transient analysis will be dependent on the type of flight profile information provided. Hypothetical values were used for all required thermal input data.

4.1 INPUT DATA.

THERMOD requires one input file, input.dat. Once invoked (by double-clicking THERMOD), THERMOD automatically reads in the input.dat file and reads out two output files. These output files are summary.dat and transient.dat. The summary.dat file summarizes the temperature results and prints the maximum operating temperatures for steady state and transient state. The transient.dat file prints the transient temperatures with respect to time.

A complete description of the input file, input variables, their definitions, and formats is located in the THERMOD User's Manual [2]. Refer to figures 1-1 through 1-4 for a graphical description of the problem.

Numerical values for the geometric and thermal properties for the hypothetical problem are presented in the following.

The environmental data for the sample problem is based on table 4-1. An analysis was performed for each hour noted in the table. This is important because the maximum ambient temperature could lag the maximum solar radiation, as indicated in the table. To account for infrared sky radiation, an effective sky temperature of 0°F was assumed. This temperature can vary from about -50°F under a cold, clear sky to +50°F under warm, cloudy conditions [6]. Because a desert environment is likely to be simulated during an extremely hot day, it is reasonable to assume a cold, clear sky that would allow maximum solar radiation to penetrate through the atmosphere. Thus, an effective sky temperature of 0°F was assumed.

TABLE 4-1. TEMPERATURE AND ASSOCIATED RADIATION DATA

Hour	Angle* (degrees)	Ambient Temperature (°F)	Wind (ft/sec)	Solar Radiation (Btu/hr/ft ²)
1100	60	106	10	300
1200	75	109	11	325
1300	90	114	14	325
1400	75	117	13	300
1500	60	120	12	261
1600	45	119	12	201
1700	30	118	10	130

*Angle refers to the sun's incident angle at different times. To determine the infrared radiation from the sky, an effective sky temperature of 0°F was assumed.

The dimensional schematics of the model are given in figure 1-2. The dimensions and thermal property values of various elements of the hypothetical problem are given in tables 4-2 through 4-10.

TABLE 4-2. THICKNESS OF THERMAL ELEMENTS OF THE WING

Thermal Element	Thickness (in)
Outer skin	0.03
Foam	0.375
Inner skin	0.02
Spar cap	0.30
Air space	12.0
Spar cap	0.30
Inner skin	0.02
Foam	0.375
Outer skin	0.03

Use figure 1-2 (Detail a: Wing) in conjunction with this table.

TABLE 4-3. THICKNESS OF THERMAL ELEMENTS OF THE FUSELAGE SIDE

Thermal Element	Thickness (in)
Outer skin	0.03
Foam	0.375
Inner skin	0.02
Insulatory material C	0.25

Use figure 1-2 (Detail b: Fuselage side) in conjunction with this table.

TABLE 4-4. THICKNESS OF THE THERMAL ELEMENTS OF THE FLOOR

Thermal Element	Thickness (in)
Insulatory material B	5.0
Outer skin	0.03
Foam	0.375
Inner skin	0.02
Floor space	3.0
Inner skin	0.02
Foam	0.375
Outer skin	0.03

Use figure 1-2 (Detail c: Floor) in conjunction with this table. The insulatory material B is a symbolic representation of the seat, the carpet, the middle console, and other possible items, all combined and treated as one material. This assumption is necessary for THERMOD models conduction in one dimension only (through-thickness). In addition, THERMOD considers all discretized elements as being isothermal, characterized by uniform radiosity, irradiation, and material properties. The thickness of insulatory material B is regarded as the weighted average (in terms of surface area) of the items just described.

TABLE 4-5. THICKNESS OF THE THERMAL ELEMENTS OF THE ROOF

Thermal Element	Thickness (in)
Outer skin	0.03
Foam	0.375
Inner skin	0.02
Insulatory material A	0.25

Use figure 1-2 (Detail d: Roof) in conjunction with this table.

TABLE 4-6. OVERALL DIMENSIONS OF THE THERMAL MODEL

Dimension	Length (ft)
L_{wing}	35.0
W_{wing}	3.5
H_{wing}	2.8
L_{cabin}	11.0
W_{cabin}	4.0
H_{cabin}	2.5
L_{flg}	22.0
L_{ledge}	5.5

Greenhouse effect within the cabin is modeled with the following dimensions:
 $W_{cabin} = 4.0$ ft; $H_{cabin} = 2.5$ ft; $L_{cabin} = 11.0$ ft.

TABLE 4-7. ABSORPTIVITY AND EMISSIVITY PROPERTIES OF THE EXTERIOR SURFACES OF THE WING AND FUSELAGE

Thermal Element	Absorptivity	Emissivity
Wing	0.45	0.80
Fuselage sides	0.65	0.90
Fuselage top	0.50	0.85
Fuselage bottom	0.60	0.85

TABLE 4-8. THERMAL PROPERTIES OF VARIOUS SOLID MATERIALS

Thermal Element	Density (lb/ft ³)	Conductivity (Btu/hr/ft/°F)	Specific Heat Capacity (Btu/lb/°F)
Divinycell HT70 foam [13]	4.4	0.021	0.24
E-Glass Composite [14]	125	0.30	0.30
Fuselage Interior Material (A, B, C as in figure 1-1)	10	0.020	0.24
Plexiglas [6]	130	0.80	0.20
Tarmac [6]	150	0.80	0.22

Note: The numbers in brackets refer to the references from which the corresponding data were obtained.

TABLE 4-9. THERMAL PROPERTIES OF AIR

Thermal Element	Density (lb/ft ³)	Conductivity (Btu/hr/ft/°F)	Specific Heat Capacity (Btu/lb/°F)	Kinematic Viscosity (ft ² /sec)	Prandtl Number
Air [6]	0.067	0.016	0.24	198e-6	0.704

Note: The number in brackets refer to the references from which the corresponding data were obtained. Air properties are based on a temperature of 150°F.

TABLE 4-10. MISCELLANEOUS PROPERTIES

Item	Values
Tarmac surface: asphalt [6]	Absorptivity = 0.9; emissivity = 0.9
Tarmac depth	2.0 ft of concrete
Insulatory material A and C [15]	Absorptivity = 0.5; emissivity = 0.9
Insulatory material B [15]	Absorptivity = 0.6; emissivity = 0.9
Plexiglas [6 and 16]	Absorptivity = 0.2; emissivity = 0.9; transmissivity = 0.70; thickness = 0.25 in.
Cabin scatter factor	Sides = 0.9; floor = 0.9
Exterior scatter factor [6]	Wing = 0.9; fuselage = 0.9

Note: The numbers in brackets refer to the references from which the corresponding data were obtained.

Note that the E-glass values in table 4-8 are based on typical 7781 epoxy-based glass composite [14]. Materials A and C could represent some typical composite panels made of felt core, sandwiched between layers of glass cloth. The insulatory material B is a symbolic representation of the seat, carpet, and other possible items, all combined and treated as one material. This assumption is necessary for THERMOD models conduction in one dimension only (through-thickness). In addition, THERMOD considers all discretized elements as being isothermal, characterized by uniform radiosity, irradiation, and material properties. The thermal properties of insulatory materials A, B, and C are simply assumed as shown.

The thermal values used for Plexiglas in table 4-8 are typical values for this material [6]. The tarmac base material is conservatively assumed to be concrete, allowing for substantial conduction of heat from the unsheltered tarmac region to the sheltered tarmac region underneath the wing. The tarmac, however, is assumed to be paved with an asphalt surface, thereby, absorbing the most heat from the sun (solar absorptivity = 0.9; see table 4-10) and emitting the most infrared radiation (emissivity = 0.9; see table 4-10) than would a typical concrete-paved tarmac.

In table 4-10, a tarmac depth of 2.0 ft was assumed: judged thick enough to conduct substantial heat from the unsheltered tarmac to the sheltered tarmac underneath the wing. The absorptivity of the insulatory materials A, B, and C may vary from 0.1 to 0.7 [15], depending on the texture and its composition. Their emissivities are generally high [15], therefore, a value of 0.9 was

assumed for all of them. Depending on the thickness and tint of the Plexiglas, its transmissivity may vary. FAA Title 14 Code of Federal Regulation 23.775 (d) stipulates that the transmissivity of the windshield and side windows must not be less than 0.70 when the pilot is seated in the normal flight position. Based on this stipulation, a transmissivity value of 0.70 was assumed. A Plexiglas thickness of 0.25 in. was assumed in the analysis.

The scatter factor for the cabin and exterior (wings and fuselage) surfaces are as shown in table 4-10. The scatter factor is defined as the fraction of the reflected solar radiation that is diffuse; the rest is specular (focus). Generally, the cabin does not have highly polished surfaces, therefore, a 0.9 scatter factor is justified. The exterior surface is, generally, not highly polished and non-mirror-like. Reference 6 states that for most engineering applications, a dull surface may be assumed. A scatter factor of 0.9, which produces a dull surface, is assumed for the exterior surface. This means 10% of the reflected solar radiation is specular.

Cooling takes place when the door is opened and the full convective forces that have always been present outside (due to wind, see table 4-1) is assumed to be applied within the cabin until an ambient cabin temperature of 140°F is reached. This ambient temperature is calculated by taking the weighted average surface temperatures of T25, T26, T43, T44, and T50 at each time step of the transient phase. When the ambient temperature reaches 140°F, the door is closed, and the aircraft begins taxiing as indicated in figure 1-4.

Based on the above information, THERMOD determines the steady-state temperature of the model. The results are then used as a starting base in a second phase of analysis, in which cooling is introduced. This phase of the analysis is called the transient analysis. To undertake this analysis, the flight profile information of figure 1-3 is used. The flight profile provides information on relative wind velocity, which is then used to determine the instantaneous convective coefficient. The higher the coefficient the sooner the temperature reaches a steady-state during the transient cooling phase. At the end of the transient analysis (at the point of application of the limit or gust load), the 44 structural temperatures, are compared to each other, and the maximum of these structural temperatures is then used as the MOL temperature for a given paint scheme. It should be emphasized that THERMOD was developed in this manner so that no special remedial measures need to be taken to address temperatures in heat-soaked deep locations. These and all other temperatures are addressed as individual unknowns within a system of nonlinear equations, and the maximum of these unknowns (structural temperatures) is reported as the MOL temperature for a given paint scheme.

4.2 INPUT FILE.

The input file, input.dat, for the sample problem is presented in table 4-11. A total of 15 sets of data are required to complete the model description. Most of these data sets were developed based on the information provided above. Some information, not described above, was also included in these sets, which can be found in the THERMOD User's Manual [2].

TABLE 4-11. INPUT FILE (input.dat) FOR THE SAMPLE PROBLEM

```

198e-6,0.704
1, 0.030,125,0.300,0.30
2, 0.375,4.4,0.021,0.24
3, 0.020,125,0.300,0.30
4, 0.300,125,0.300,0.30
5, 12.0,0.067,0.016,0.24
6, 0.300,125,0.300,0.30
7, 0.020,125,0.300,0.30
8, 0.375,4.4,0.021,0.24
9, 0.030,125,0.300,0.30
10,0.030,125,0.300,0.30
11,0.375,4.4,0.021,0.24
12,0.020,125,0.300,0.30
13,0.250,10.,0.020,0.24
14,48.00,0.067,0.016,0.24
15,0.030,125,0.300,0.30
16,0.375,4.4,0.021,0.24
17,0.020,125,0.300,0.30
18,0.250,10.,0.020,0.24
19,30.00,0.067,0.016,0.24
20,5.000,10.,0.020,0.24
21,0.030,125,0.300,0.30
22,0.375,4.4,0.021,0.24
23,0.020,125,0.300,0.30
24,3.00,0.067,0.016,0.24
25,0.020,125,0.300,0.30
26,0.375,4.4,0.021,0.24
27,0.030,125,0.300,0.30
28,0.250,130,0.800,0.20
3.5,35.0,2.8,5.5,22.0,11.0,2.5,4.0,30,24,28
0.70,0.9,0.9,2.0,150,0.80,0.22
0.5,0.9,0.6,0.9,0.5,0.9,0.2,0.9
3,100
0.9,0.9,0.9,0.9
0.65,0.90,0.45,0.80,0.50,0.85,0.60,0.85
9
0,10,10,0,0,75,110,110,190,190
5,120,5,60,20,5,120,60,60
7
60 565.67 459.67 10 300 10 10 0.36
75 568.67 459.67 11 325 10 10 0.31
90 573.67 459.67 14 325 10 12 0.33
75 576.67 459.67 13 300 10 10 0.31
60 579.67 459.67 12 261 10 10 0.36
45 578.67 459.67 12 201 10 10 0.44
30 577.67 459.67 10 130 10 10 0.57
1,0.02,140,100
10,1.0,1.0
  
```


TABLE 4-12. SAMPLE PROBLEM OUTPUT DATA FILE (summary.dat) (Continued)

10	111.4	117.2	119.6	119.9	117.9	111.2	103.9
11	134.1	145.9	148.3	144.1	133.8	116.2	98.1
12	134.1	145.9	148.3	144.1	133.8	116.2	98.1
13	133.6	145.2	147.7	143.5	133.5	116.1	98.2
14	133.6	145.2	147.7	143.5	133.5	116.1	98.2
15	133.6	145.2	147.6	143.5	133.4	116.1	98.2
16	111.9	117.8	120.3	120.5	118.3	111.3	103.7
17	111.9	117.8	120.2	120.5	118.3	111.3	103.7
18	111.9	117.8	120.2	120.5	118.3	111.3	103.7
19	111.4	117.1	119.6	119.9	117.9	111.2	103.9
20	111.4	117.1	119.6	119.9	117.9	111.2	103.9
21	135.8	145.0	144.3	141.2	132.7	115.4	92.8
22	135.9	145.1	144.4	141.3	132.8	115.6	92.8
23	166.2	169.4	165.9	164.3	160.0	143.1	109.0
24	166.3	169.5	166.0	164.4	160.1	143.2	109.1
25	187.5	186.5	181.0	180.5	179.1	162.5	120.4
26	195.0	193.5	188.7	187.4	186.5	170.6	125.1
27	148.4	150.7	149.8	149.4	147.9	137.0	110.9
28	148.4	150.7	149.8	149.3	147.9	137.0	110.9
29	145.1	147.6	147.0	146.6	145.1	134.6	109.9
30	145.1	147.6	147.0	146.6	145.1	134.6	109.9
31	127.6	131.5	132.4	132.3	130.6	122.0	104.5
32	127.6	131.5	132.4	132.3	130.6	122.0	104.5
33	127.6	131.5	132.4	132.3	130.6	122.0	104.5
34	127.6	131.5	132.4	132.3	130.6	122.0	104.5
35	110.1	115.5	117.8	118.1	116.2	109.4	99.2
36	110.1	115.5	117.8	118.1	116.2	109.4	99.2
37	106.8	112.4	115.0	115.4	113.4	107.0	98.2
38	106.8	112.4	115.0	115.3	113.4	107.0	98.2
39	159.2	147.2	126.8	143.9	153.5	147.2	130.2
40	159.3	147.3	127.0	144.0	153.6	147.3	130.2
41	174.0	166.3	152.3	161.8	167.1	157.2	129.8
42	174.1	166.4	152.4	161.8	167.2	157.2	129.8
43	184.4	179.7	170.2	174.3	176.7	164.2	129.6
44	187.2	180.7	170.1	175.4	179.5	169.1	135.0
45	160.5	158.6	152.2	155.4	156.8	146.9	121.3
46	160.4	158.5	152.1	155.3	156.6	146.8	121.2
47	122.3	126.9	126.4	126.7	124.1	115.0	101.6
48	122.1	126.7	126.3	126.6	123.9	114.9	101.4
49	149.8	150.9	146.4	147.1	146.2	134.9	113.9
50	151.7	152.6	148.0	148.7	147.8	136.3	114.4
51	109.7	116.7	118.1	117.3	113.0	102.9	90.6
52	104.2	111.0	112.9	112.1	108.0	98.6	86.5
53	109.7	116.7	118.1	117.3	113.0	102.9	90.6

Maximum Structural Temperature for Period 1 = 174.10 Occurring
 at Location 42

TABLE 4-12. SAMPLE PROBLEM OUTPUT DATA FILE (summary.dat) (Continued)

Maximum Structural Temperature for Period	2	=	169.52	Occurring			
at Location	24						
Maximum Structural Temperature for Period	3	=	165.98	Occurring			
at Location	24						
Maximum Structural Temperature for Period	4	=	164.41	Occurring			
at Location	24						
Maximum Structural Temperature for Period	5	=	167.20	Occurring			
at Location	42						
Maximum Structural Temperature for Period	6	=	157.23	Occurring			
at Location	42						
Maximum Structural Temperature for Period	7	=	130.19	Occurring			
at Location	39						
Maximum Structural Temperature Over All	7	Periods	=174.10				
Occurring at Location	42	at Period	1				
TEMPERATURES AT THE END OF TRANSIENT ANALYSIS							
TEMPERATURES IN DEGREES FAHRENHEIT							
	1	2	3	4	5	6	7
1	108.5	112.3	117.2	119.5	121.4	118.9	116.5
2	109.3	113.3	118.1	120.3	121.8	118.9	116.0
3	135.1	145.5	147.0	144.1	135.3	118.9	101.7
4	135.9	146.4	147.9	144.8	135.7	118.9	101.2
5	136.5	147.2	148.6	145.4	136.0	118.9	100.9
6	112.1	117.9	120.3	120.6	118.4	111.4	104.4
7	111.9	117.8	120.2	120.5	118.4	111.6	104.6
8	111.8	117.6	120.1	120.5	118.5	111.8	104.9
9	107.4	110.9	115.6	118.2	120.4	118.4	116.5
10	107.3	110.7	115.5	118.1	120.4	118.6	116.9
11	108.4	112.2	117.2	119.5	121.3	118.8	116.4
12	109.2	113.2	118.1	120.2	121.6	118.7	115.9
13	131.7	142.5	145.2	141.7	132.4	116.2	99.7
14	132.4	143.4	146.0	142.4	132.7	116.1	99.2
15	132.9	144.1	146.6	142.9	132.9	116.1	98.8
16	111.9	117.8	120.3	120.5	118.3	111.3	104.3
17	111.8	117.7	120.2	120.4	118.3	111.5	104.5
18	111.7	117.5	120.0	120.4	118.4	111.7	104.9
19	107.4	110.9	115.6	118.2	120.4	118.4	116.5
20	107.3	110.7	115.5	118.1	120.4	118.6	116.9
21	108.6	112.6	117.5	119.5	121.0	118.3	115.3
22	109.5	113.4	118.2	120.2	121.5	118.7	115.3
23	135.3	138.6	140.9	139.5	136.8	132.6	114.4
24	136.4	139.7	141.8	140.4	137.6	133.3	114.5
25	148.7	150.0	150.5	149.3	147.4	143.0	119.3
26	153.3	154.1	154.9	153.5	152.3	147.9	122.5

TABLE 4-12. SAMPLE PROBLEM OUTPUT DATA FILE (summary.dat) (Continued)

27	148.8	151.0	150.1	149.6	148.2	137.2	110.9
28	147.8	150.1	149.3	148.8	147.4	136.6	110.8
29	145.1	147.7	147.1	146.7	145.2	134.6	110.0
30	145.1	147.6	147.0	146.6	145.1	134.6	110.0
31	128.1	131.7	132.9	133.3	132.4	124.5	108.4
32	128.1	131.7	132.9	133.3	132.4	124.5	108.4
33	128.1	131.7	132.9	133.3	132.4	124.5	108.4
34	128.1	131.7	132.9	133.3	132.4	124.5	108.4
35	111.1	115.8	118.9	120.0	119.6	114.4	106.8
36	111.0	115.6	118.8	120.0	119.6	114.5	107.0
37	107.1	110.6	115.3	117.8	119.9	117.8	115.7
38	107.0	110.5	115.2	117.7	119.9	118.0	116.0
39	109.1	111.4	114.9	118.6	121.7	120.2	117.9
40	110.0	112.2	115.5	119.2	122.3	120.8	118.1
41	137.3	136.0	133.9	137.5	139.3	137.7	123.5
42	138.3	136.9	134.7	138.2	140.0	138.3	123.7
43	144.9	144.1	142.7	144.6	145.6	143.0	124.5
44	147.5	145.2	142.7	145.7	148.0	146.6	128.2
45	136.2	135.6	134.5	137.1	138.6	137.5	123.2
46	135.1	134.6	133.8	136.4	137.9	136.8	122.9
47	109.9	112.1	115.5	119.2	122.2	120.7	118.0
48	109.0	111.4	114.9	118.6	121.7	120.2	117.8
49	112.2	115.2	119.2	121.4	123.3	121.1	117.0
50	114.3	117.3	121.0	123.0	124.7	122.3	117.2
51	166.3	183.2	182.1	174.9	157.7	128.5	98.0
52	155.1	170.7	170.4	163.9	148.2	121.6	93.1
53	166.3	183.2	182.1	174.9	157.7	128.5	98.0
Maximum Structural Temperature for Period 1 = 148.77 Occurring at Location 27							
Maximum Structural Temperature for Period 2 = 151.00 Occurring at Location 27							
Maximum Structural Temperature for Period 3 = 150.07 Occurring at Location 27							
Maximum Structural Temperature for Period 4 = 149.62 Occurring at Location 27							
Maximum Structural Temperature for Period 5 = 148.17 Occurring at Location 27							
Maximum Structural Temperature for Period 6 = 138.33 Occurring at Location 42							
Maximum Structural Temperature for Period 7 = 123.67 Occurring at Location 42							
Maximum Structural Temperature Over All 7 Periods =151.00 Occurring at Location 27 at Period 2							

The MOL temperature at the end of the steady-state analysis was 174.1°F. The MOL temperature at the end of the transient analysis was 151.0°F. Thus, the final MOL temperature for this aircraft was 151.0°F.

TABLE 4-13. SAMPLE PROBLEM OUTPUT DATA FILE (transient.dat)

```

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
THERMOD:  A THERMAL PROGRAM FOR AIRCRAFTS
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

TRANSIENT TEMPERATURES FOR PERIOD:          1

COOLING WHILE DOOR IS OPENED:

time(sec)      1      2      3      4      .      .      .      51      52      53
0.0000    138.1  138.1  137.5  137.5      .      .      .    109.7  104.2  109.7
2.0000    138.1  138.1  137.5  137.5      .      .      .    109.7  104.2  109.7
4.0000    138.1  138.1  137.6  137.5      .      .      .    109.7  104.2  109.7
6.0000    138.1  138.1  137.6  137.5      .      .      .    109.7  104.2  109.7
8.0000    138.1  138.1  137.6  137.5      .      .      .    109.7  104.2  109.7
10.0000   138.1  138.1  137.6  137.5      .      .      .    109.7  104.2  109.7
.          .      .      .      .      .      .      .      .      .      .
.          .      .      .      .      .      .      .      .      .      .
386.0000   138.1  138.1  137.6  137.5      .      .      .    109.7  104.2  109.7
388.0000   138.1  138.1  137.6  137.5      .      .      .    109.7  104.2  109.7
390.0000   138.1  138.1  137.6  137.5      .      .      .    109.7  104.2  109.7

COOLING DURING AIRCRAFT MANEUVERS:

time(sec)      1      2      3      4      .      .      .      51      52      53
392.0000   138.1  138.1  137.6  137.5      .      .      .    119.1  112.6  119.0
394.0000   138.0  138.1  137.6  137.5      .      .      .    145.5  136.3  145.5
396.0000   137.9  138.0  137.6  137.5      .      .      .    157.2  146.8  157.2
398.0000   137.8  137.9  137.6  137.5      .      .      .    162.3  151.4  162.3
400.0000   137.6  137.8  137.6  137.5      .      .      .    164.6  153.5  164.6
.          .      .      .      .      .      .      .      .      .      .
.          .      .      .      .      .      .      .      .      .      .
.          .      .      .      .      .      .      .      .      .      .
840.0000   108.5  109.3  135.2  135.9      .      .      .    166.3  155.1  166.3
842.0000   108.5  109.3  135.1  135.9      .      .      .    166.3  155.1  166.3
844.0000   108.5  109.3  135.1  135.9      .      .      .    166.3  155.1  166.3
846.0000   108.5  109.3  135.1  135.9      .      .      .    166.3  155.1  166.3

TRANSIENT TEMPERATURES FOR PERIOD:          2

COOLING WHILE DOOR IS OPENED:

time(sec)      1      2      3      4      .      .      .      51      52      53
.          .      .      .      .      .      .      .      .      .      .
.          .      .      .      .      .      .      .      .      .      .
.          .      .      .      .      .      .      .      .      .      .

COOLING DURING AIRCRAFT MANEUVERS:

time(sec)      1      2      3      4      .      .      .      51      52      53
.          .      .      .      .      .      .      .      .      .      .
.          .      .      .      .      .      .      .      .      .      .
.          .      .      .      .      .      .      .      .      .      .

```

TABLE 4-13. SAMPLE PROBLEM OUTPUT DATA FILE (transient.dat) (Continued)

TRANSIENT TEMPERATURES FOR PERIOD:										3
COOLING WHILE DOOR IS OPENED:										
time(sec)	1	2	3	4	.	.	.	51	52	53
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
COOLING DURING AIRCRAFT MANEUVERS:										
time(sec)	1	2	3	4	.	.	.	51	52	53
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
TRANSIENT TEMPERATURES FOR PERIOD:										4
COOLING WHILE DOOR IS OPENED:										
time(sec)	1	2	3	4	.	.	.	51	52	53
0.0000	146.9	146.9	146.3	146.3	.	.	.	117.3	112.1	117.3
2.0000	146.9	146.9	146.3	146.3	.	.	.	117.3	112.1	117.3
4.0000	146.9	146.9	146.3	146.3	.	.	.	117.3	112.1	117.3
6.0000	146.9	146.9	146.3	146.3	.	.	.	117.3	112.1	117.3
8.0000	146.9	146.9	146.3	146.3	.	.	.	117.3	112.1	117.3
10.0000	146.9	146.9	146.3	146.3	.	.	.	117.3	112.1	117.3
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
336.0000	146.9	146.9	146.3	146.3	.	.	.	117.3	112.1	117.3
338.0000	146.9	146.9	146.3	146.3	.	.	.	117.3	112.1	117.3
340.0000	146.9	146.9	146.3	146.3	.	.	.	117.3	112.1	117.3
342.0000	146.9	146.9	146.3	146.3	.	.	.	117.3	112.1	117.3
344.0000	146.9	146.9	146.3	146.3	.	.	.	117.3	112.1	117.3
COOLING DURING AIRCRAFT MANEUVERS:										
time(sec)	1	2	3	4	.	.	.	51	52	53
346.0000	146.9	146.9	146.3	146.3	.	.	.	143.5	135.7	143.5
348.0000	146.8	146.8	146.3	146.3	.	.	.	161.1	151.4	161.1
350.0000	146.6	146.7	146.3	146.3	.	.	.	168.8	158.4	168.8
352.0000	146.5	146.6	146.3	146.3	.	.	.	172.2	161.5	172.2
354.0000	146.4	146.5	146.3	146.3	.	.	.	173.7	162.8	173.7
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
790.0000	119.5	120.3	144.2	144.9	145.5	120.6	120.5	174.9	163.9	174.9
792.0000	119.5	120.3	144.1	144.9	145.5	120.6	120.5	174.9	163.9	174.9
794.0000	119.5	120.3	144.1	144.9	145.4	120.6	120.5	174.9	163.9	174.9
796.0000	119.5	120.3	144.1	144.8	145.4	120.6	120.5	174.9	163.9	174.9
798.0000	119.5	120.3	144.1	144.8	145.4	120.6	120.5	174.9	163.9	174.9

TABLE 4-13. SAMPLE PROBLEM OUTPUT DATA FILE (transient.dat) (Continued)

TRANSIENT TEMPERATURES FOR PERIOD:										5
COOLING WHILE DOOR IS OPENED:										
time(sec)	1	2	3	4	.	.	.	51	52	53
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
COOLING DURING AIRCRAFT MANEUVERS:										
time(sec)	1	2	3	4	.	.	.	51	52	53
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
TRANSIENT TEMPERATURES FOR PERIOD:										6
COOLING WHILE DOOR IS OPENED:										
time(sec)	1	2	3	4	.	.	.	51	52	53
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
COOLING DURING AIRCRAFT MANEUVERS:										
time(sec)	1	2	3	4	.	.	.	51	52	53
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
TRANSIENT TEMPERATURES FOR PERIOD:										7
COOLING WHILE DOOR IS OPENED:										
time(sec)	1	2	3	4	.	.	.	51	52	53
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
COOLING DURING AIRCRAFT MANEUVERS:										
time(sec)	1	2	3	4	.	.	.	51	52	53
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.

Note: This file was truncated. Transient data was written for each time period for all 53 temperatures. Interior cooling took place when the door was opened. Once cabin ambient temperature of 140°F was reached, the door was closed, and the aircraft was subjected to further cooling while maneuvering according to the input flight profile.

5. SUMMARY.

A thermal analysis was performed on a hypothetical aircraft. The wing was painted with a paint color containing the following thermal properties: absorptivity = 0.45 and emissivity = 0.80. The fuselage was painted as follows: (1) sides: absorptivity = 0.65 and emissivity = 0.90, (2) top: absorptivity = 0.50 and emissivity = 0.85, and (3) bottom: absorptivity = 0.60 and emissivity = 0.85.

Table 4-12 shows all 53 steady-state temperatures for different times of the day. For example, at 1100 hours (time period 1), temperature T1 is 138.1°F, and at 1300 hours (time period 2), it is 149.1°F. The hottest structural temperature is T42 (174.1°F), occurring at 1100 hours (time period 1). If only steady-state results were used in the Federal Aviation Administration certification, the maximum operating limit (MOL) temperature for this aircraft should at least be 175°F. However, THERMOD was modeled to consider transient analysis, and as a result, the benefits associated with this cooling phase should be considered.

Table 4-12 also shows the results of the transient analysis. The transient analysis was performed using the flight profile of figure 1-4 and the steady-state temperatures of table 4-12 as the starting base. In general, the transient temperatures of table 4-12 are lower than steady-state temperatures. The hottest transient structural temperature is T27 (151.0°F), occurring at 1200 hours (time period 2). This temperature is practically the same as its steady-state counterpart (T27 = 150.7°F). This is due to the fact that T27 was located underneath the 5-inch-thick insulatory material (material B, representing the seats, carpet, and other related items), which acts as an effective barrier to thermal changes above it. Note that in contrast, surfaces that are exposed to convective forces (such as those surfaces represented by T1, T11, T25, T26, T39, T43, T44, and T49) undergo rapid cooling.

A few important observations from tables 4-12 and 4-13 should be noted.

1. A total of 53 temperatures were simulated, out of which 44 temperatures were considered structural and 9 were nonstructural. The structural temperatures were T1-T24, T27-T38, T39-T42, and T45-T48. The nonstructural temperatures were
 - a. insulatory material temperatures (T25, T26, T43, and T44);
 - b. Plexiglas temperatures (T49 and T50); and
 - c. tarmac temperatures (T51, T52, and T53).
2. Transient temperature results were simulated based on the flight profile information in figure 1-4.
3. To obtain conservative results, THERMOD assumed the ground to possess thermal characteristics similar to that of the tarmac throughout the entire flight profile, even while the plane was in the air.
4. In some cases, the transient temperatures of table 4-12 are higher than the corresponding steady-state temperatures, even though the transient phase is thought to have a cooling effect on the plane. Generally, during the transient phase, the aircraft is expected to cool

down due to convective heat transfer from the air flowing over (it until equilibrium is achieved).

For the purpose of computing the transient phase temperatures, THERMOD does not take airplane altitude into account, but instead, compensates by changing the air flow temperature over the airplane to equal the ambient temperatures prescribed for that flight profile. The convective heat transfer coefficients vary according to the time and speed set for each flight segment by the user. The user also sets the ambient conditions when they specify the “time of the day”, “sun inclination angle”, “relative wind speed” and “solar radiation” etc. This means, that the airplane would in fact never leave the ground, but its flight profile such as shown in figure 1-4 would be simulated by changing the ambient conditions and the convective coefficients for each specified flight segment.

Thus, in some cases, when the user has set the time of the day as early morning or late evenings, and depending on the user’s other ‘weather condition’ inputs, it’s sometimes possible that the airplane’s exterior is cooler (due to relatively cooler steady state temperatures) than the relatively warmer surroundings (which are warmer due to the user specified inputs of solar radiation, sun angle etc.). And thus, for calculating transient phase temperatures, when THERMOD moves this relatively warmer air over the airplane, it raises the temperature on the airplane’s exterior.

5. The tarmac temperatures (T51, T52, and T53) show significant increase during the transient phase. In the steady-state phase, these temperatures were lower because they represent regions that were directly under the wing, which provided some measure of shade from direct sunlight. However, because the plane is in motion, as in the transient phase, these temperatures were essentially the exposed tarmac temperatures.
6. For a given paint scheme, THERMOD simulated the temperatures at each time period. The 44 structural temperatures in the transient phase were compared with each other over seven time periods (1100, 1200, 1300, 1400, 1500, 1600, and 1700 hours), and the maximum structural temperature obtained from this comparison was used as the MOL temperature. This MOL temperature may be located at the surface (depending on how dark the paint is), inside the cabin, within the floor space, or elsewhere. It is the maximum structural temperature experienced anywhere in the aircraft.

From this project, it was observed that this hypothetical aircraft with paint absorptivity and emissivity values as noted above had a MOL temperature of 174.1°F at the end of the steady-state analysis. The MOL temperature at the end of the transient analysis was 151.0°F. Thus, the transient analysis reduced the maximum soaked temperature by 23.1°F. The final MOL temperature for this aircraft was 151.0°F. This temperature occurred on the floor surface underneath the insulatory material (material B of figure 1-1) covering the floor.

6. REFERENCES.

1. Govindarajoo, N., "THERMOD: A Thermal Model for Predicting Aircraft MOL Temperatures," Report No. REK0002A.DOC, Cirrus Design Corporation, Duluth, MN, October 1996.
2. Govindarajoo, N., "THERMOD, An Enhanced Thermal Model for Determining Aircraft Operational Temperatures: THERMOD User's Manual," FAA report DOT/FAA/AR-04/52, December 2004.
3. Miller, L. Scott, Waltner, J., Mazmudar, T., Merchant, M., and Tomblin, J., "THERMOD Composite Airframe Temperature Prediction Tool Evaluation, Validation, and Enhancement With Initial Steady-State Temperature Data," FAA report DOT/FAA/AR-04/30, September 2004.
4. Miller, L.S. "Impact of Aircraft Operation on Composite Airframe Temperatures," AGATE-WP3.3-033051-122, October 2001 (also available at <http://www.niar.wichita.edu/agate>).
5. Press, W.H. et al., "Numerical Recipes in FORTRAN: The Art of Scientific Computing," second edition, Cambridge Univ. Press, New York, NY, 1994.
6. Incropera, F.P. and Dewitt, D.P., "Fundamentals of Heat and Mass Transfer," third edition, John Wiley & Sons, New York, NY, 1990.
7. UAI/NASTRAN 20.1, "User's Reference Manual," Universal Analytics, Inc., 3625 Del Almo Blvd., Suite 370, Torrance, CA, 1999.
8. UAI/NASTRAN 20.1, "User's Guide," Universal Analytics, Inc., 3625 Del Almo Blvd., Suite 370, Torrance, CA, 1999.
9. MScNASTRAN 70.5, "Quick Reference Guide," MacNeal-Schwendler Corp., 815 Colorado Boulevard, Los Angeles, CA, 1998.
10. MScNASTRAN 70.5, "User's Guide," MacNeal-Schwendler Corp., 815 Colorado Boulevard, Los Angeles, CA, 1998.
11. MScNASTRAN 70.5, "Command Reference Guide," MacNeal-Schwendler Corp., 815 Colorado Boulevard, Los Angeles, CA, 1998.
12. MScNASTRAN 68, "Thermal Analysis User's Guide," MacNeal-Schwendler Corp., 815 Colorado Boulevard, Los Angeles, CA, 1994.
13. Technical Brochure, Divinycell International Inc., DIAD Group, 315 Seahawk Drive, DeSoto, TX 75115.
14. Reinhart, T.J., Chairman, "Engineered Materials Handbook Vol. 1, Composites," ASM International, OH, 1987.

15. Gubareff, G.G. et al., "Thermal Radiation Properties Survey: A Review of Literature," second edition, Honeywell Research Center, Minneapolis, MN, 1960.
16. Gebhart, B., "Heat Transfer," second edition, McGraw-Hill Book Co., 1971.

APPENDIX A—THERMOD CODE

THERMOD is a FORTRAN-based program developed using the DIGITAL VISUAL FORTRAN 6.0 compiler, an IBM-compatible Pentium II PC, and Windows 98. THERMOD is noninteractive and runs on either the DOS or Windows platform. A computer with 64 MB or more of RAM and with at least 500 MB of hard disk space is recommended. A typical analysis may take from as little as 30 seconds to a few hours, depending on whether the explicit forward finite difference method (EFFDM) or the implicit backward finite difference method (IBFDM), respectively, is selected. It also depends on the time step used in the transient analysis.

THERMOD requires one input file, input.dat. Once invoked (by double-clicking THERMOD), THERMOD automatically reads in the input.dat file and reads out two output files, which are summary.dat and transient.dat. The summary.dat file summarizes the temperature results and prints the maximum operating temperatures for steady state and transient state. The transient.dat file prints the transient temperatures with respect to time. Details on these files and instructions on how to use THERMOD can be found in THERMOD User's Manual [A-1].

THERMOD simulates the service temperatures of an aircraft by setting up and solving a set of 67 equations in a nonlinear system of equations to produce 53 steady-state temperatures and 14 steady-state radiosity. The solution is performed using the Newton-Raphson iteration technique [A-2]. This is followed by a transient analysis, based on either the EFFDM or the IBFDM. Figure A-1 shows a flowchart of THERMOD. The following paragraphs give a brief description of THERMOD, and section A.1 lists the definitions of the routines. Section A.2 presents the THERMOD code in its entity.

MAIN is the main controlling program that calls subroutine READIN. READIN reads in the input file containing the relevant information to run THERMOD. MAIN next calls the SOLUTION subroutine, which controls both the steady-state and transient analyses. SOLUTION performs this function by calling 11 subroutines as shown in figure A-1. THERMOD was programmed to simulate the temperatures of an aircraft for a particular paint scheme chosen. The subroutines VIEWFACT, HCONVECT, QREFCAB, TARMMOD, INFRQSKY, and SBFILLET are called by SOLUTION only once. SOLUTION also calls subroutines QREFFW, QREFTW, NEWTRAPH, and TRANSIEN.

The steady-state analysis phase is controlled by the NEWTRAPH subroutine, which calls THERMODL, LUDCMP, and LUBKSB subroutines. The THERMODL subroutine forms the 67 equations needed in THERMODL. This is the heart of the steady-state analysis. THERMODL also calls JACOBIAN to formulate the Jacobians needed in the Newton-Raphson Iteration solution procedure [A-2]. Having formulated the equations using the THERMODL and JACOBIAN subroutines, NEWTRAPH next calls the LUDCMP subroutine, which decomposes the Jacobian matrix. This is followed by another call by NEWTRAPH to LUBKSB where backsubstitution is performed to find the solution for the unknowns.

The transient analysis phase is controlled either by the TRNSIEN1 or TRNSIEN2 subroutines, where the EFFDM method is controlled by TRNSIEN1 and the IBFDM method is controlled by TRNSIEN2. TRNSIEN1 calls LHSWNG1, RHSWNG1, FUSLGR1, FUSLGRF1, TARMAC1,

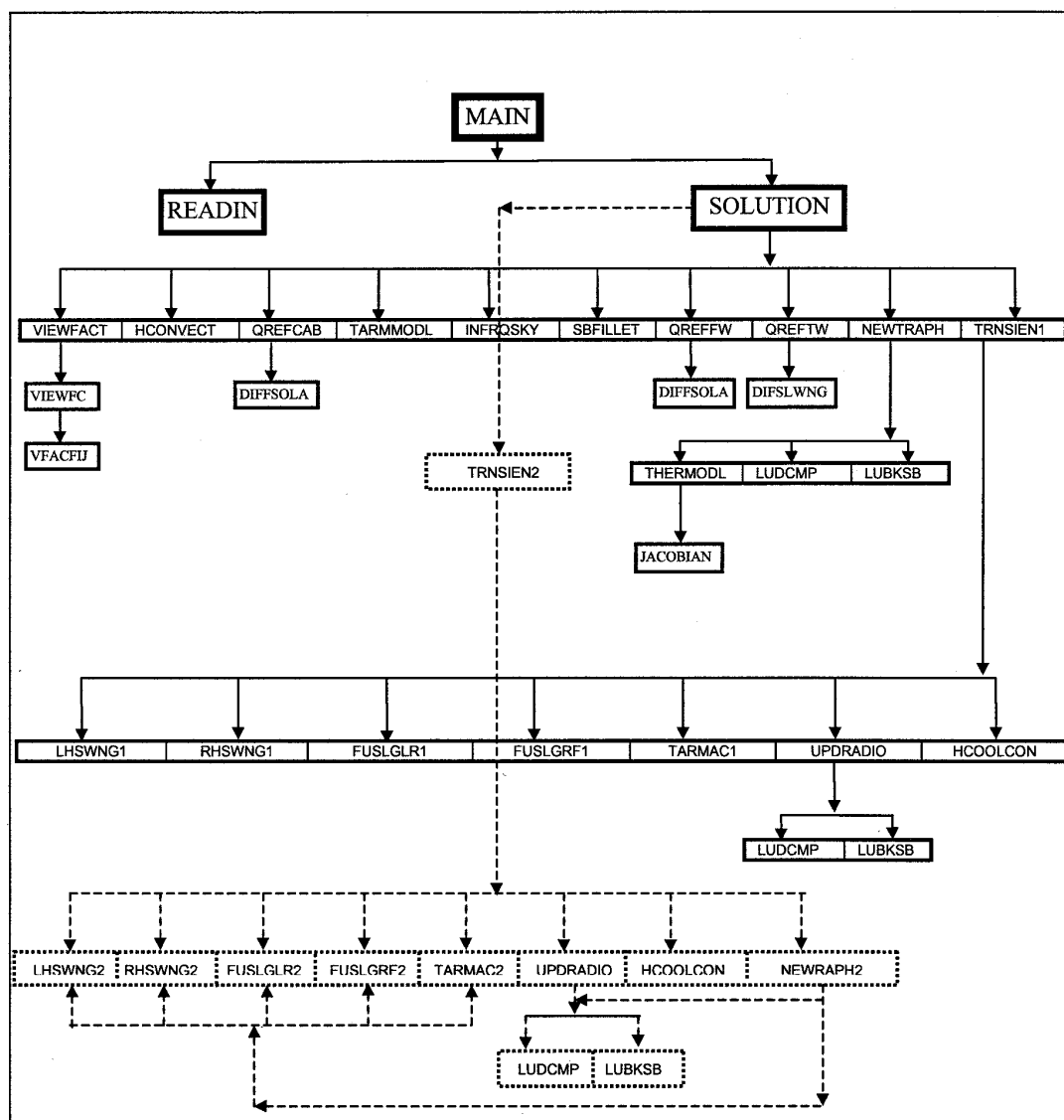
UPRADIO, and HCOOLCON subroutines. These subroutines together with LUDCMP and LUBKSB complete the EFFDM transient analysis. The inclusion of the UPRADIO subroutine, which updates the radiosity for each time increment, is necessary because without it, numerical overflow or other numerical difficulties will result. TRNSIEN2 calls LHSWNG2, RHSWNG2, FUSLGR2, FUSLGRF2, TARMAC2, UPRADIO, and HCOOLCON and NEWGRAPH2 subroutines. The UPDRADIO calls LUDCMP and LUBKSB to update the radiosity. The NEWGRAPH2 subroutine calls LHSWNG2, RHSWNG2, FUSLGR2, FUSLGRF2, TARMAC2, LUDCMP, and LUBKSB to complete the IBFDM finite difference analysis.

A.1 DEFINITION OF SUBROUTINES.

MAIN	The main controlling program.
READIN	Reads in the input data.
SOLUTION	Controls both the steady-state and transient analyses.
VIEWFACT	The controlling routine for determining view factors for aligned parallel rectangles and for perpendicular rectangles with a common edge.
VIEWFC	Determines the view factors for aligned parallel rectangles, and is the controlling routine for determining the view factors for perpendicular rectangles with a common edge.
VFACFIJ	Determines the view factors for perpendicular rectangles with a common edge.
HCONVECT	Determines the steady-state convective coefficients for the fuselage, wing, and tarmac.
QREFCAB	Determines the intercepted solar radiation in the cabin, and is the controlling routine for determining the diffuse solar radiation that is absorbed in the cabin by treating the cabin as a partial cavity.
DIFFSOLA	Determines the diffuse solar radiation that is absorbed in a six-sided rectangular enclosure by treating it as a partial cavity in which multiple reflections and absorptions are considered.
TARMMODL	Determines the infrared emission from the tarmac as well as the tarmac temperature.
INFRQSKY	Determines the infrared emission from the sky.
SBFILLET	Determines the view factors of fillets by assuming them to be a quadrant.
QREFFW	Determines the intercepted solar radiation at the fuselage-wing junction, and is the controlling routine for determining the diffuse solar radiation that is absorbed at this junction by treating the junction as a partial cavity.

QREFTW	Determines the intercepted solar radiation at the tarmac-wing bottom region, and is the controlling routine for determining the diffuse solar radiation that is absorbed in this region by treating the region as a partial cavity.
DIFSLWNG	Determines the diffuse solar radiation that is absorbed at the tarmac-wing bottom region by representing the region as a six-sided rectangular enclosure and by treating it as a partial cavity in which multiple reflections and absorptions are considered. The region is divided into three smaller rectangular boxes, representing the LHS, RHS, and the MIDDLE of the wing.
NEWTRAPH	The controlling routine for the steady-state analysis based on the Newton-Raphson iteration technique.
THERMODL	Formulates the 67 heat transfer equations.
LUDCMP	Decomposes a matrix by the LU decomposition technique.
LUBKSB	Determines the solution of the unknown by backward substitution.
JACOBIAN	Determines the Jacobian for the 67 equations as a requirement of the Newton-Raphson iteration technique.
TRNSIEN1	The controlling routine for the EFFDM transient analysis.
LHSWNG1	Performs EFFDM transient analysis of the left wing (the wing exposed to the solar radiation).
RHSWNG1	Performs EFFDM transient analysis of the right wing (the wing that is under the shade of the fuselage).
FUSLGRL1	Performs EFFDM transient analysis of the fuselage from left to right.
FUSLGRF1	Performs EFFDM transient analysis of the fuselage from top to bottom (including the Plexiglas cover).
TARMAC1	Performs EFFDM transient analysis of the tarmac.
UPRADIO	Updates radiosity (all 14) at each time increment of the transient analysis.
HCOOLCON	Determines the convective coefficient at each time increment for the fuselage surface and the wing surface based on a given flight profile.
TRNSIEN2	The controlling routine for the IBFDM transient analysis.
LHSWNG2	Performs IBFDM transient analysis of the left wing (the wing exposed to the solar radiation).
RHSWNG2	Performs IBFDM transient analysis of the right wing (the wing that is under the shade of the fuselage).

- FUSLGLR2 Performs IBFDM transient analysis of the fuselage from left to right.
- FUSLGRF2 Performs IBFDM transient analysis of the fuselage from top to bottom (including the Plexiglas cover).
- TARMAC2 Performs IBFDM transient analysis of the tarmac.
- NEWGRAPH2 The controlling subroutine for the nonlinear solution of the IBFDM transient problem. The nonlinear solution is provided by the Newton-Raphson iteration technique.



Solid lines represent the original THERMOD program. The dotted lines represent additions to the original THERMOD program.

FIGURE A-1. FLOWCHART OF THE THERMOD MODEL

A-2. THERMOD PROGRAM CODE.

program main

c This is the main controlling program.

c This program determines the service temperatures of an aircraft.

c There are 67 equations with 67 unknowns to be solved in a non-linear

c system of equations.

c

c Program developed by Nathan Govindarajoo, PhD., PE., Consulting Engineer.

c Program belongs to FAA and AGATE.

c

```

write(6,*)'$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$'
write(6,*)'WELCOME TO THERMOD: A THERMAL PROGRAM FOR AIRCRAFTS'
write(6,*)'$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$'
  write(6,*) '
  write(6,*)'INPUT DATA BEING READ'
call readin
write(6,*)'INPUT DATA SUCCESSFULLY READ'
  write(6,*) '
  write(6,*)'SOLUTION IN PROGRESS'
call solution
write(6,*) '
  write(6,*)'THERMOD SUCCESSFULLY COMPLETED'
end

```

c

subroutine readin

c purpose: reads in the data

c calls: none

c called from: main

```

common/c1/n,ntrial,tolt,tolfn
common/c2/t(200),fn(200),jacob(200,200)
common/c3/th(100),k(100)
common/c4/widthwng,lengtwng,hwng,locwng,lplane,lfuselg,hfuselg,
&      wfuselg
common/c5/trnsm,absorbta,emista,tarnd,densta,conducta,shcta
common/c6/absorbf,emisf,absorbw,emisw,absorbfl,emisfl,absorbfi,
&emisfi,absorbpq,emispq,absorftp,emisrftp,absorfbt,emisrftb,
&absorfbf,emisfb
common/c7/nperiods,envtdata(100,20)
common/c8/icode,nconstrn,idtemp(100),tempcons(100)
common/c9/scatffi,scatffw,scatff
common/c10/aircondc,thdiffus,visckine,prandtl
common/c11/icodef11,percent,nparts
common/c12/a1,a2,a3,a4,a5,a6(100),a7(100)
common/c13/nintervl,vel(100),timeintv(100)
common/c14/thicknes(100),density(100),conductv(100),sheatcap(100)
common/c15/ibegabsf,iendabsf,ibegemsw,iendemsw,ibegabsw,iendabsw,
&      ibegemsw,iendemsw
common/c16/ifindiff,delttime,amcabtmp,intvprnt
common/store1/absorbff,emisff,absorbww,emisww,absrftpp,emirftpp,
&      absrbfbb,emisfbb

```

```
real k,lengthwng,locwng,lplane,lfuselg
parameter(small=0.00001,smallth=0.0001,glassk=1.0)
open(unit=8,file='input.dat',status='old')
```

c

c Variables:

```
c absorbf=absorbtivity of exterior fuselage side
c emisf=emissivity of exterior fuselage side
c absorbfl=absorbtivity of floor of cabin (fuselage interior)
c emisfl=emissivity of floor of cabin (fuselage interior)
c absorbfi=absorbtivity of side of cabin (fuselage interior)
c emisfi=emissivity of side of cabin (fuselage interior)
c absorbpg=absorbtivity of plexiglass
c emispg=emissivity of plexiglass
c absorftp=absorbtivity of the top of roof top of fuselage exterior
c emisrft=emissivity of the top of roof top of fuselage exterior
c absorbt=absorbtivity of the bottom of roof top in fuselage interior
c emisrbt=emissivity of the bottom of roof top in fuselage interior
c absorbfb=absorbtivity of the bottom of fuselage exterior (ie above tarmac)
c emisfb=emissivity of the bottom of fuselage exterior (ie above tarmac)
c n=number of equations=67
c ifindiff=indicator for the type of finite difference method in transient analysis:
c   =1 means explicit finite difference method
c   =2 means implicit finite difference method
c ntrial=maximum number of iterations allowed in the non-linear solution
c tolt=temperature tolerance in the solution
c toln=function tolerance in the solution
c t(=temperature (deg F or Rankine)
c fn(=equations/or functions
c jacob(,)=Jacobian
c th(=element thickness (ft)
c k(=element conductivity ((BTU/hr/ft/deg F)
c widthwng=wing width (ft)
c lengthwng=total wing length (tip-to-tip) (ft)
c hwng=wing height from ground (ft)
c locwng=leading edge wing location from the fuselage front (ft)
c lplane=entire plane length (ft)
c lfuselg=fuselage length (ft)
c hfuselg=height of fuselage (ft)
c wfuselg=width of fuselage (ft)
c trns=transmissivity of plexiglass
c absorbta=tarmac absorbtivity
c emista=tarmac emissivity
c tarnd=tarmac depth (overall) (ft)
c densta=tarmac density (overall) (lb/ft^3)
c conducta=tarmac conductivity (overall) (BTU/hr/ft/deg F)
c shcta=tarmac specific heat capacity (overall) (BTU/lbm/deg F)
c nperiods=number of time intervals (say from the morning till the evening)
c envtda(,1)=sun's incident angle (degrees)
c envtda(,2)=ambient temperature (Rankine)
c envtda(,3)=effective sky temperature (Rankine)
c envtda(,4)=wind speed (ft/s)
```

c envtdata(,5)=total (both direct and diffused) solar radiation (BTU/hr/sq.ft)
 c envtdata(,6)=area (sq. ft) of plexiglass perpendicular to the sun, & seen
 c by the sun
 c envtdata(,7)=area (sq. ft) of plexiglass horizontal to the sun, & seen
 c by the sun
 c envtdata(,8)=fraction of sun that gets into the fuselage (in terms of
 c the projected area)
 c icode=1, means, only one analysis is done knowing the paint scheme
 c icode=2, means, analysis is done for all paint schemes, satisfying
 c some temperature requirements
 c nconstrn=number of temperatures that must be satisfied above
 c idtemp(,)=which temperature
 c tempcons(,)=the value of the temperature requirement to be satisfied
 c scatffl=fraction of the light reflected from the floor
 c that is diffused
 c scatffi=fraction of the light reflected from the interior fuselage side
 c that is diffused
 c scatfw=fraction of the light reflected from the wing that is diffused
 c scatff=fraction of the light reflected from the exterior fuselage side
 c that is diffused
 c aircondc=conductivity of air (BTU/hr/ft/deg F)
 c thdiffus=air thermal diffusivity (ft²/sec)
 c visckine=air kinematic viscosity (ft²/sec)
 c prandtl=Prandtl number
 c icodef11=1: the fillet is completely painted with paint that is the same as
 c that used on the wing
 c icodef11=2: the fillet is completely painted with paint that is the same as
 c that used on the fuselage
 c icodef11=3: the fuselage/wing paint boundary falls on the fillet; need
 c to assign the percentage of the fillet occupied by the wing paint
 c percent=percentage when icodef11=3
 c nparts=determines absorptivity and emissivity increments (1/nparts)
 c a1=total area (sq. ft) of the plexiglass
 c a2=total area (sq. ft) of the fuselage top half
 c a3=a1/(a1+a2)*lfuselg*wfuselg
 c a4=a2/(a1+a2)*lfuselg*wfuselg
 c a5(,)=area (sq. ft) of plexiglass that intercepts diffuse reflected solar
 c rays from the tarmac
 c a6(,)=same as envtdata(,6)
 c a7(,)=same as envtdata(,7)
 c nintervl=number of intervals in the flight profile used in cooling phase
 c vel(,)=speed of aircraft during cooling phase (mph)
 c timeintv(,)=time interval during each portion of the flight profile (sec)
 c thicknes(,)=element thickness (in)
 c density(,)=element density(lb/ft³)
 c conductv(,)=element thermal conductivity (BTU/hr/ft/deg F)
 c sheatcap(,)=element specific heat capacity (BTU/lbm/deg F)
 c ibegabsf=fuselage absorbtivity pointer on the beginning location of
 c the paint matrix
 c iendabsf=fuselage absorbtivity pointer on the ending location of
 c the paint matrix

```

c ibegemsf=fuselage emissivity pointer on the beginning location of
c   the paint matrix
c iendemsf=fuselage emissivity pointer on the ending location of
c   the paint matrix
c ibegabsw=wing absorbtivity pointer on the beginning location of
c   the paint matrix
c iendabsw=wing absorbtivity pointer on the ending location of
c   the paint matrix
c ibegemsw=wing emissivity pointer on the beginning location of
c   the paint matrix
c iendemsw=wing emissivity pointer on the ending location of
c   the paint matrix
c deltime=time increment in the transient phase (sec)
c amcabtmp=ambient cabin temperature that is tolerable before egressing
c   and taxing (deg F)
c intvprnt=print interval in the transient phase
c
  read(8,*)visckine,prandtl
  do 10 i=1,28
c j is number of the element: it is just a dummy; not used anywhere
    read(8,*)j,thicknes(i),density(i),conductv(i),sheatcap(i)
10  thicknes(i)=thicknes(i)/12.
    read(8,*)widthwng, lengtwng, hwng, locwng, lplane, lfuselg, hfuselg,
    & wfuselg, a1, a2, a5
    read(8,*)trnsm, absorbta, emista, tarmd, densta, conducta, shcta
    read(8,*)absorbfl, emisfl, absorbfi, emisfi, absorfbt, emisrftb,
    & absorbpg, emispg
    if(absorbfl.eq.1.0)absorbfl=1.-small
    if(absorbfl.eq.0.0)absorbfl=small
    if(emisfl.eq.1.0)emisfl=1.-small
    if(emisfl.eq.0.0)emisfl=small
    if(absorbfi.eq.1.0)absorbfi=1.-small
    if(absorbfi.eq.0.0)absorbfi=small
    if(emisfi.eq.1.0)emisfi=1.-small
    if(emisfi.eq.0.0)emisfi=small
    if(absorfbt.eq.1.0)absorfbt=1.-small
    if(absorfbt.eq.0.0)absorfbt=small
    if(emisrftb.eq.1.0)emisrftb=1.-small
    if(emisrftb.eq.0.0)emisrftb=small
    if(absorbpg.eq.1.0)absorbpg=1.-small
    if(absorbpg.eq.0.0)absorbpg=small
    if(emispg.eq.1.0)emispg=1.-small
    if(emispg.eq.0.0)emispg=small
    read(8,*)icodef11,percent
    read(8,*)scatffl,scatffi,scatfw,scatff
c   read(8,*)icode
    icode=1
    if(icode.eq.1)then
      read(8,*)absorbf,emisf,absorbw,emisw,absorftp,emisrftp,
    & absorbfb,emisfb
      if(absorbf.eq.1.0)absorbf=1.-small

```

```

if(absorbf.eq.0.0)absorbf=small
if(emisf.eq.1.0)emisf=1.-small
if(emisf.eq.0.0)emisf=small
if(absorbw.eq.1.0)absorbw=1.-small
if(absorbw.eq.0.0)absorbw=small
if(emisw.eq.1.0)emisw=1.-small
if(emisw.eq.0.0)emisw=small
if(absorftp.eq.1.0)absorftp=1.-small
if(absorftp.eq.0.0)absorftp=small
if(emisrftp.eq.1.0)emisrftp=1.-small
if(emisrftp.eq.0.0)emisrftp=small
if(absorbfbb.eq.1.0)absorbfbb=1.-small
if(absorbfbb.eq.0.0)absorbfbb=small
if(emisfbb.eq.1.0)emisfbb=1.-small
if(emisfbb.eq.0.0)emisfbb=small
elseif(icode.eq.2)then
  read(8,*)nparts
  read(8,*)ibegabsf,iendabsf
  read(8,*)ibegemsf,iendemsf
  read(8,*)ibegabsw,iendabsw
  read(8,*)ibegemsw,iendemsw
  read(8,*)nconstrn
  read(8,*)(idtemp(i),tempcons(i),i=1,nconstrn)
endif
read(8,*)nintervl
read(8,*)(vel(i),i=1,nintervl+1)
read(8,*)(timeintv(i),i=1,nintervl)
read(8,*)nperiods
do 20 i=1,nperiods
20  read(8,*)(envtdata(i,j),j=1,8)
    n=67
    read(8,*)ifindiff,delttime,amcabtmp,intvpnt
    amcabtmp=amcabtmp+459.67
    read(8,*)ntrial,tolt,tolfn
c initial trial for the aircraft temperatures and radiosities
  read(8,*)(t(i),i=1,n)
c because some of these properties will be destroyed depending on
c icode, they will be stored for later retrieval
  absorbff=absorbf
  emisff=emisf
  absorbww=absorbw
  emisww=emisw
  absrftpp=absorftp
  emirftpp=emisrftp
  absrbfbb=absorbfbb
  emisfbb=emisfbb
c
c reference some of the values as follows:
c
c air conductivity & thermal diffusivity within the cabin
c conductv(14) same as conductv(19)

```

aircondc=conductv(19)

thdiffus=conductv(19)/(density(19)*sheatcap(19))*(1./3600)

c bec. the model was developed piece-meal, need to reorder things

c as follows (for internal representation):

c first thickness:

th(1)=thicknes(1)

th(2)=thicknes(2)

th(3)=thicknes(3)

th(4)=thicknes(4)

th(29)=thicknes(5)

th(30)=thicknes(6)

th(5)=thicknes(7)

th(6)=thicknes(8)

th(7)=thicknes(9)

th(8)=thicknes(10)

th(9)=thicknes(11)

th(10)=thicknes(12)

th(23)=thicknes(13)

th(11)=thicknes(14)

th(27)=thicknes(15)

th(26)=thicknes(16)

th(25)=thicknes(17)

th(28)=thicknes(18)

th(24)=thicknes(19)

th(21)=thicknes(20)

th(20)=thicknes(21)

th(19)=thicknes(22)

th(18)=thicknes(23)

th(31)=thicknes(24)

th(14)=thicknes(25)

th(13)=thicknes(26)

th(12)=thicknes(27)

th(22)=thicknes(28)

c spar inside the cabin is assumed not to be in direct contact with the

c floor or belly-pan. its temperature is taken to be the average of the

c floor bottom and top of belly-pan. because of this, and the fact that

c floor bottom temp. is usually higher than the belly-pan top, no bulk

c or turbulent air motion exists. also radiation effects

c will be assumed to be negligible; therefore, temperatures:

c t(31), t(32), t(33) and t(32) will all be assumed to be the same.

c ie equal to the average of the floor bottom and top of belly-pan.

c therefore the floor space (space between the floor bottom and belly-pan

c top is assumed=th(31), and all others (th(17); th(16); th(32); and

c th(15) is assumed very small;= smallth. numerically ok.

th(17)=smallth

th(16)=smallth

th(32)=smallth

th(15)=smallth

c now conductivity:

k(1)=conductv(1)

k(2)=conductv(2)

```
k(3)=conductv(3)
k(4)=conductv(4)
k(29)=conductv(5)
k(30)=conductv(6)
k(5)=conductv(7)
k(6)=conductv(8)
k(7)=conductv(9)
k(8)=conductv(10)
k(9)=conductv(11)
k(10)=conductv(12)
k(23)=conductv(13)
k(11)=conductv(14)
k(27)=conductv(15)
k(26)=conductv(16)
k(25)=conductv(17)
k(28)=conductv(18)
k(24)=conductv(19)
k(21)=conductv(20)
k(20)=conductv(21)
k(19)=conductv(22)
k(18)=conductv(23)
k(31)=conductv(24)
k(14)=conductv(25)
k(13)=conductv(26)
k(12)=conductv(27)
k(22)=conductv(28)
```

c because of the statements made above, k(17); k(16); k(32) and
 c k(15) are taken as somewhat higher than glass (1.0); too high, say 1500
 c will cause numerical overflow during the solution process:
 c this and the above thicknesses (smallth) will make k()/th() very large:
 c thereby making the resistance th()/k() negligible, which is what we want.

```
k(17)=glassk
k(16)=glassk
k(32)=glassk
k(15)=glassk
close(8)
return
end
```

c

subroutine solution

c purpose: controls the solution of the problem

c calls: viewfact,hconvect,qrefcab,tarmmodl,infrqsky,sbfillet,qreffw,

c qreffw,newtraph,transien,

c called from: main

```
common/c1/n,ntrial,tolt,tolfn
common/c2/t(200),fn(200),jacob(200,200)
common/c4/widthwng,lengtwng,hwng,locwng,lplane,lfuselg,hfuselg,
& wfuselg
common/c5/trnsm,absorbta,emista,tarmd,densta,conducta,shcta
common/c6/absorbfi,emisf,absorbw,emisw,absorbfl,emisfl,absorbfi,
& emisfi,absorbpg,emisp,absorftp,emisrftp,absorfbt,emisrbt,
```

```

&absorbfb,emisfb
common/c7/nperiods,envtdata(100,20)
common/c8/icode,nconstrn,idtemp(100),tempcons(100)
common/c11/icodef11,percent,nparts
common/c12/a1,a2,a3,a4,a5,a6(100),a7(100)
common/c15/ibegabsf,iendabsf,ibegemsw,iendemsw,ibegabsf,iendabsf,
&      ibegemsw,iendemsw
common/c16/iffindiff,delttime,amcabttmp,intvprnt
common/convect/hf(100),hw(100),ht(100)
common/tarmacqt/qqtarm(100),tttarm(100)
common/qskyinfr/qtsky(100)
common/fillet/f11w,f11f
common/result/temp(200),temptrn(200)
dimension tinitial(100)
dimension tempall(100,100),itemp1(2,100),tempall1(100),
&location(100),tempp(100),tempp1(100)
      dimension itemp1tr(2,100),temptr(100,100),temptr1(100),
&locattr(100)
character*1 indicat(150,150)/22500*'N'/
character*1 indicat1(150,150)/22500*'N'/
parameter(small=0.00001)
parameter(tarmleng=100.0)
real lfuselg,lplane
open(unit=9,file='summary.dat',status='unknown')
open(unit=10,file='transient.dat',status='unknown')
c  open(unit=12,file='outconv',status='unknown')
c  open(unit=13,file='odiffsol',status='unknown')
c  open(unit=14,file='out2.dat',status='unknown')
c  open(unit=15,file='out3.dat',status='unknown')
c  open(unit=16,file='transin3.out',status='unknown')
c  open(unit=17,file='transin4.out',status='unknown')
c Variables:
c hf=fuselage convective coefficient
c hw=wing convective coefficient
c ht=tarmac convective coefficient
c qqtarm=tarmac emission
c tttarm=tarmac temperature
c qqsky=sky emission
c f11w=fillet view factor (for wing)
c f11f=fillet view factor (for fuselage)
c tinitial=initial guessed temperatures
c indicat=indicator for the paint matrix
c tarmleng=tarmac length for finding hw
c
c store the initial guessed temperatures
c
  do 80 i=1,n
    80  tinitial(i)=t(i)
c
c store the various areas relating the fuselage roof and
c the plexiglass:

```

```

c
a3=a1/(a1+a2)*lfuselg*wfuselg
a4=a2/(a1+a2)*lfuselg*wfuselg
do 85 j=1,nperiods
  a6(j)=envtdata(j,6)
85  a7(j)=envtdata(j,7)
c
c find view factors
  call viewfact
c find convection coefficients (hf) for the fuselage (4 sides) &
c for the left wing (hw) (top & bottom) and the
c right wing (hw) (top & bottom).
c hf for all four fuselage sides are the same
c hw for all the wing sides are the same
c
c find hf for the fuselage:
  call hconvect(lplane,hf)
c find hw for the wings
  call hconvect(widthwng,hw)
c now find ht for the tarmac
  call hconvect(tarmleng,ht)
c find reflected solar rays (qrefl) within the cabin
  call qrefcab
c find tarmac temperature and qtarm (tarmac radiation)
  call tarmmodl(absorbta,emista,qqtarm,ttarm)
c find the infrared sky radiation
  call infrqsky(qqsky)
c find the modification factors for absorbtivity & emissivity due to fillets
  call sbfillet(icodef11,percent,f11w,f11f)
  if(icode.eq.2)del=1./nparts
c
c for the option chosen determine the temperatures; or paint scheme for
c a given set of temperatures chosen
c
c icode=1: just one run with temperature output for a given
c set of paint scheme:
c
  write(10,*)'$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$'
  write(10,*)'THERMOD: A THERMAL PROGRAM FOR AIRCRAFTS'
  write(10,*)'$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$'
  if(icode.eq.1)then
    itotitr=0
    absinfrf=emisf*(1./(1.-f11f*(1.-emisf)))
    eemisf=emisf*(1.-f11f)/(1.-f11f*(1.-emisf))
    rff=absinfrf/eemisf
    call qreffw(absorbf,absorbw)
    call qreftw(absorbw)
    absinfrw=emisw*(1./(1.-f11w*(1.-emisw)))
    eemisw=emisw*(1.-f11w)/(1.-f11w*(1.-emisw))
    rfw=absinfrw/eemisw
  c
    write(6,*)'absorbf,emisf,absorbw,emisw'

```

```

c      write(6,*)absorbf,emisf,absorbw,emisw
do 10 i=1,nperiods
write(6,*)' '
      write(6,*)'STEADY STATE ANALYSIS BEGUN FOR PERIOD',i
      itotitr=itotitr+1
      do 90 itemp=1,n
90      t(itemp)=tinitial(itemp)
      theta=envtdata(i,1)
      ta=envtdata(i,2)
      qsky=qqsky(i)
      qdir=envtdata(i,5)
      hhf=hf(i)
      hhw=hw(i)
      hht=ht(i)
      qtarm=qqtarm(i)
      ttarm=tttarm(i)
c find airplane temperature
      call newtraph(i,kkk,theta,ta,qsky,hhf,hhw,hht,qdir,
&      qtarm,ttarm,absorbf,emisf,absorbw,emisw,
&      eemisf,eemisw,rff,rfw)
95      format(1x,i1,1x,i4,70(1x,f4.0))
      write(6,*)'STEADY STATE ANALYSIS COMPLETED FOR PERIOD',i
c      write(6,95)i,kkk,(temp(j),j=1,n)
      itemp1(1,i)=i
c      itemp1(2,i)=kkk
      do 97 j=1,n
97      tempall(j,i)=temp(j)
      write(6,*)'TRANSIENT ANALYSIS BEGUN FOR PERIOD',i
      if(ifindiff.eq.1)call trnsien1(itotitr,i,theta,ta,qsky,
&      hhf,hhw,hht,qdir,qtarm,ttarm,absorbf,emisf,absorbw,
&      emisw,eemisf,eemisw,rff,rfw)
      if(ifindiff.eq.2)call trnsien2(itotitr,i,theta,ta,qsky,
&      hhf,hhw,hht,qdir,qtarm,ttarm,absorbf,emisf,absorbw,
&      emisw,eemisf,eemisw,rff,rfw)
      write(6,*)'TRANSIENT ANALYSIS COMPLETED FOR PERIOD',i
      itemp1tr(1,i)=i
      do 197 j=1,n
197      temptr(j,i)=temptrn(j)
10      continue
c find the maximum temperatures
      write(9,*)'$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$'
      write(9,*)'THERMOD: A THERMAL PROGRAM FOR AIRCRAFTS'
      write(9,*)'$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$'
      write(9,*)' '
      write(9,*)' '
      write(9,*)' '
      write(9,*)'SUMMARY OF STEADY-STATE AND TRANSIENT ANALYSIS'
      write(9,*)' '
      write(9,*)' '
      write(9,*)' '
c      write(9,*)'absorbf,emisf,absorbw,emisw'

```

```

c      write(9,*)absorbf,emisf,absorbw,emisw
      tempall2=0.0
      tempr2=0.0
      do 120 i=1,nperiods
        tempall1(i)=0.0
        tempr1(i)=0.0
c Temperatures T25, T26, T43, T44 (all insulatory surfaces); T51, T52 and
c T53 (tarmac surfaces); and T49 and T50 (plexiglass material) are
c considered non-structural.
        do 125 j=1,48
          if(j.ne.25.and.j.ne.26.and.j.ne.43.and.j.ne.44)then
            if(tempall(j,i).gt.tempall1(i))then
              tempall1(i)=tempall(j,i)
              location(i)=j
            endif
          endif
          if(j.ne.25.and.j.ne.26.and.j.ne.43.and.j.ne.44)then
            if(tempr(j,i).gt.tempr1(i))then
              tempr1(i)=tempr(j,i)
              locattr(i)=j
            endif
          endif
125      continue
          if(tempall1(i).gt.tempall2)then
            tempall2=tempall1(i)
            locfinal=location(i)
            iperiodf=i
          endif
          if(tempr1(i).gt.tempr2)then
            tempr2=tempr1(i)
            locfintr=locattr(i)
            iperiotr=i
          endif
120      continue
c print the temperatures
c      do 100 j=1,2
c 100      write(9,101)(itemp1(j,i),i=1,nperiods)
c steady-state output
      write(9,*)(Temperatures are shown for all time periods)'
      write(9,*)' '
      write(9,*)' '
      write(9,*)TEMPERATURES AT THE END OF STEADY STATE ANALYSIS'
      write(9,*)' '
250      write(9,250)
      format(8x,'TEMPERATURES IN DEGREES FAHRENHEIT')
      write(9,101)(itemp1(1,i),i=1,nperiods)
101      format(4x,100(1x,i5))
      write(9,*)' '
      do 99 j=1,53
99      write(9,98)j,(tempall(j,i),i=1,nperiods)
98      format(1x,i3,100(1x,f5.1))
  
```

```

    write(9,*)"
        do 130 i=1,nperiods
c          write(6,*)'maximum structural temperature for period',i,'=',
c      &      tempall1(i),' occurring at location',location(i)
          write(9,135)i,tempall1(i),location(i)
135      format('Maximum Structural Temperature for Period',1x,i3,1x,
      &      ' ',1x,f6.2,1x,'Occurring at Location',1x,i3)
130      continue
c          write(6,*)'maximum structural temperature over all periods=',
c      &      tempall2,' occurring at location ',locfinal,' and at ',
c      &      'period ',iperiodf
    write(9,*)' '
        write(9,136)nperiods,tempall2,locfinal,iperiodf
136      format('Maximum Structural Temperature Over All',1x,i3,
      &      1x,'Periods',1x,'=',f6.2,1x,'Occurring at Location',1x,i3,
      &      1x,'at Period',1x,i3)
c transient analysis output
    write(9,*)' '
    write(9,*)' '
    write(9,*)'TEMPERATURES AT THE END OF TRANSIENT ANALYSIS'
    write(9,*)' '
    write(9,250)
    write(9,101)(itemp1tr(1,i),i=1,nperiods)
    write(9,*)' '
        do 299 j=1,53
299      write(9,98)j,(temptr(j,i),i=1,nperiods)
    write(9,*)"
        do 1130 i=1,nperiods
c          write(6,*)'maximum structural temperature for period',i,'=',
c      &      temptr1(i),' occurring at location',locattr(i)
          write(9,137)i,temptr1(i),locattr(i)
137      format('Maximum Structural Temperature for Period',1x,i3,1x,
      &      ' ',1x,f6.2,1x,'Occurring at Location',1x,i3)
1130      continue
c          write(6,*)'maximum structural temperature over all',nperiods,
c          'periods=',
c      &      temptr2,' occurring at location ',locfintr,' and at ',
c      &      'period ',iperiotr
    write(9,*)' '
        write(9,138)nperiods,temptr2,locfintr,iperiotr
138      format('Maximum Structural Temperature Over All',1x,i3,
      &      1x,'Periods',1x,'=',f6.2,1x,'Occurring at Location',1x,i3,
      &      1x,'at Period',1x,i3)
c
c
c
c icode=2: this is the "mother" of all iterations.  for a given paint
c on the fuselage, this option iterates for all paint combination on the
c wing.  this process is repeated for the next fuselage paint.      note
c that the fuselage paint are the same for the sides, top and bottom.
c this is how it has been programmed. can be modified if requested.

```

c this option outputs a matrix of paint schemes that are permissible for
 c the entire plane without violating a set of temperature
 c requirements requested by the user.

```

c
  elseif(icode.eq.2)then
    itotitr=0
    do 440 iabsorbf=ibegabsf,iendabsf
      absorbf=(iabsorbf-1)*del
      if(iabsorbf.eq.1)absorbf=small
      if(iabsorbf.eq.nparts+1)absorbf=1.-small
      absorftp=absorbf
      absorbfb=absorbf
      do 460 iemisf=ibegemsf,iendemsf
        isumm=(iabsorbf-1)*(nparts+1)+iemisf
        emisf=(iemisf-1)*del
        if(iemisf.eq.1)emisf=small
        if(iemisf.eq.nparts+1)emisf=1.-small
        emisrftp=emisf
        emisfb=emisf
        absinfrf=emisf*(1./(1.-f11f*(1.-emisf)))
        eemisf=emisf*(1.-f11f)/(1.-f11f*(1.-emisf))
        rff=absinfrf/eemisf
        do 470 iabsorbw=ibegabsw,iendabsw
          absorbw=(iabsorbw-1)*del
          if(iabsorbw.eq.1)absorbw=small
          if(iabsorbw.eq.nparts+1)absorbw=1.-small
          call qreffw(absorbf,absorbw)
          call qreftw(absorbw)
          do 450 iemisw=ibegemsw,iendemsw
            isum=(iabsorbw-1)*(nparts+1)+iemisw
            emisw=(iemisw-1)*del
            if(iemisw.eq.1)emisw=small
            if(iemisw.eq.nparts+1)emisw=1.-small
            absinfrw=emisw*(1./(1.-f11w*(1.-emisw)))
            eemisw=emisw*(1.-f11w)/(1.-f11w*(1.-emisw))
            rfw=absinfrw/eemisw
            itot=0
            itot1=0
            do 425 iconstrn=1,nconstrn
              tempp(iconstrn)=0.0
            do 426 iconstrn=1,nconstrn
              tempp1(iconstrn)=0.0
            do 410 i=1,nperiods
              itotitr=itotitr+1
              do 490 itemp=1,n
                t(itemp)=tinitial(itemp)
                theta=envtdata(i,1)
                ta=envtdata(i,2)
                qsky=qsky(i)
                qdir=envtdata(i,5)
                hhf=hf(i)

```

```

      hhw=hw(i)
      hht=ht(i)
      qtarm=qqarm(i)
      ttarm=ttarm(i)
c find airplane temperature
      call newtraph(i,kkk,theta,ta,tsky,hhf,hhw,hht,qdir,
&      qtarm,ttarm,absorbf,emisf,absorbw,emisw,
&      eemisf,eemisw,rff,rfw)
      write(9,*)'STEADY STATE ANALYSIS'
      write(9,*)'Total # of Analysis (per period)=' ,itotitr
      write(9,*)'absorbf,emisf,absorbw,emisw,period,iteratns'
      write(9,*)absorbf,emisf,absorbw,emisw,i,kkk
      write(9,96)(ii,ii=1,n)
96      format(7x,70(1x,i4))
      write(9,95)i,kkk,(temp(ii),ii=1,n)
      write(6,*)'STEADY STATE ANALYSIS'
      write(6,*)'Total # of Analysis (per period)=' ,itotitr
      write(6,*)'absorbf,emisf,absorbw,emisw,period,iteratns'
      write(6,*)absorbf,emisf,absorbw,emisw,i,kkk
      write(6,96)(ii,ii=1,n)
      write(6,95)i,kkk,(temp(ii),ii=1,n)
      do 430 iconstrn=1,nconstrn
        if(temp(idtemp(iconstrn)).gt.tempp(iconstrn))
&          tempp(iconstrn)=temp(idtemp(iconstrn))
430      continue
      if(ifindiff.eq.1)call trnsien1(itotitr,i,theta,ta,tsky,
&      hhf,hhw,hht,qdir,qtarm,ttarm,absorbf,emisf,absorbw,
&      emisw,eemisf,eemisw,rff,rfw)
      if(ifindiff.eq.2)call trnsien2(itotitr,i,theta,ta,tsky,
&      hhf,hhw,hht,qdir,qtarm,ttarm,absorbf,emisf,absorbw,
&      emisw,eemisf,eemisw,rff,rfw)
      write(6,*)'TRANSIENT ANALYSIS'
      write(9,*)'TRANSIENT ANALYSIS'
      write(6,95)i,kkk,(temptrn(ii),ii=1,n)
      write(9,95)i,kkk,(temptrn(ii),ii=1,n)
      do 431 iconstrn=1,nconstrn
        if(temptrn(idtemp(iconstrn)).gt.tempp1(iconstrn))
&          tempp1(iconstrn)=temptrn(idtemp(iconstrn))
431      continue
410      continue
      do 435 iconstrn=1,nconstrn
435      if(tempp(iconstrn).le.temppcons(iconstrn))
&        itot=itot+1
      if(itot.eq.nconstrn)then
        indicat(isumm,isum)='Y'
        write(9,*)'passed the steady-state temp. constraints'
        write(9,*)'absorbf,emisf,absorbw,emisw'
        write(9,*)'absorbf,emisf,absorbw,emisw'
        write(9,96)(ii,ii=1,n)
        write(9,199)(temp(i),i=1,67)
199      format(7x,70(1x,f4.0))

```

```

endif
do 436 iconstrn=1,nconstrn
436   if(temp1(iconstrn).le.tempcons(iconstrn))
    &       itot1=itot1+1
    if(itot1.eq.nconstrn)then
      indicat1(isumm,isum)="Y"
      write(9,*)'passed the transient temp. constraints'
      write(9,*)'absorbf,emisf,absorbw,emisw'
      write(9,*)'absorbf,emisf,absorbw,emisw'
      write(9,96)(ii,ii=1,n)
      write(9,199)(temptrn(i),i=1,67)
    endif
450   continue
470   continue
460   continue
440   continue
      nmatrix=(nparts+1)**2
      write(9,*)'STEADY STATE ANALYSIS ONLY'
      do 420 isumm=1,nmatrix
        write(9,495)(indicat(isumm,isum),isum=1,nmatrix)
420   continue
      write(9,*)'AT THE END OF TRANSIENT ANALYSIS'
      do 421 isumm=1,nmatrix
        write(9,495)(indicat1(isumm,isum),isum=1,nmatrix)
421   continue
495   format(150a1)
      endif
      close(9)
      close(10)
c   close(12)
c   close(13)
c   close(14)
c   close(15)
c   close(16)
c   close(17)
      return
      end
c
c
      subroutine hconvect(l,h)
c purpose: determines the convective coefficient
c calls: none
c called from: solution
      common/c7/nperiods,envtdata(100,20)
      common/c10/k,thdiffus,visckine,prandtl
      real l,llamn,nusllamn,k
      dimension h(100)
      parameter(rellamn=5.e5)
c
c Variables:
c h: convective coefficient

```

```

c l=length
c nusllamn=Nusselt number in the laminar zone
c rellamn=Reynold's number in the laminar zone
c
  do 10 i=1,nperiods
    v=envtdata(i,4)
    rel=v*l/visckine
    if(rel.le.rellamn)then
      nusllamn=0.664*rel**(1./2)*prandtl**(1./3)
      h(i)=k*nusllamn/l
c    write(12,*)'rel,nusllamn,h',rel,nusllamn,h(i)
    elseif(rel.gt.rellamn)then
      llamn=rellamn*visckine/v
      nusllamn=0.664*rellamn**(1./2)*prandtl**(1./3)
      hlamn=k*nusllamn/llamn
      hturb=0.0296*k/(l-llamn)*prandtl**(1./3)*(v/visckine)**(4./5)
      &      *(5./4)*(l**(4./5)-llamn**(4./5))
      h(i)=1./l*(llamn*hlamn+(l-llamn)*hturb)
c    write(12,*)'rel,llamn,hlamn,hturb,h',
c    & rel,llamn,hlamn,hturb,h(i)
    endif
  10 continue
end

c
c
  subroutine viewfact
c purpose: determines the view factor (controlling routine)
c calls: viewfc
c called from: solution
  common/c4/widthwng,lengtwng,hwng,locwng,lplane,lfuselg,hfuselg,
  &      wfuselg
  common/vf1/fbwa(6,6),fbwb(6,6)
  common/vf2/fa11,fa12,fa13,fa14,fa15,fa16,fa21,fa22,fa23,fa24,
  &      fa25,fa26,fa31,fa32,fa33,fa34,fa35,fa36,fa41,fa42,
  &      fa43,fa44,fa45,fa46,fa51,fa52,fa53,fa54,fa55,fa56,
  &      fa61,fa62,fa63,fa64,fa65,fa66
  common/vf4/fa57,fa68
  common/vf5/fc11,fc12,fc13,fc14,fc15,fc16,fc21,fc22,fc23,fc24,
  &      fc25,fc26,fc31,fc32,fc33,fc34,fc35,fc36,fc41,fc42,
  &      fc43,fc44,fc45,fc46,fc51,fc52,fc53,fc54,fc55,fc56,
  &      fc61,fc62,fc63,fc64,fc65,fc66
  dimension f(6,6)
  real lengtwng,l,locwng,lfuselg,lplane

c
c Variables:
c fbwa(,)=view factors for the wing bottom (LHS & RHS)
c fbwb(,)=view factors for the wing bottom (CENTER)
c fa11....fa66=view factors for the top fuselage-wing junction
c fa57,fa68=view factors for face 5 & 7; and 6 & 8, where 7 is
c      the fuselage face front of the wing and 8 is the
c      fuselage face aft of the wing.

```

cfc11....fc66=view factors for the fuselage inside (cabin)

c

c view factors for wingbottom

c LHS & RHS:

x=widthwng

y=(lengtwnw-wfuselg)/2.

z=hwng

call viewfc(x,y,z,fbwa)

c CENTER (below fuselage):

x=widthwng

y=wfuselg

z=hwng

call viewfc(x,y,z,fbwb)

c

c view factors for the top fuselage-wing junction

x=widthwng

y=(lengtwnw-wfuselg)/2

l=hfuselg

call viewfc(x,y,l,f)

fa11=f(1,1)

fa12=f(1,2)

fa13=f(1,3)

fa14=f(1,4)

fa15=f(1,5)

fa16=f(1,6)

fa21=f(2,1)

fa22=f(2,2)

fa23=f(2,3)

fa24=f(2,4)

fa25=f(2,5)

fa26=f(2,6)

fa31=f(3,1)

fa32=f(3,2)

fa33=f(3,3)

fa34=f(3,4)

fa35=f(3,5)

fa36=f(3,6)

fa41=f(4,1)

fa42=f(4,2)

fa43=f(4,3)

fa44=f(4,4)

fa45=f(4,5)

fa46=f(4,6)

fa51=f(5,1)

fa52=f(5,2)

fa53=f(5,3)

fa54=f(5,4)

fa55=f(5,5)

fa56=f(5,6)

fa61=f(6,1)

fa62=f(6,2)

```

fa63=f(6,3)
fa64=f(6,4)
fa65=f(6,5)
fa66=f(6,6)
c view factors for face 5 & 7; and 6 & 8, where 7 is the fuselage face
c front of the wing and 8 is the fuselage face aft of the wing.
c find fa57:
  x=hfuselg
  y=(lengtwng-wfuselg)/2
  l=locwng
  call viewfc(x,y,l,f)
  fa57=f(1,2)
c   write(9,*)'fa57=',fa57
c find fa68
  x=hfuselg
  y=(lengtwng-wfuselg)/2
  l=lplane-locwng-widthwng
  call viewfc(x,y,l,f)
  fa68=f(1,2)
c   write(9,*)'fa68=',fa68
c
c find fij's for the inside of the fuselage.
c that is find fc11..fc16; fc21..fc26; fc31..fc36; fc41..fc46
c fc51..fc56; fc61..fc66. note fc11,fc22,fc33,fc44,fc55 & fc66
c are zeroes.
  x=lfuselg
  y=wfuselg
  l=hfuselg
  call viewfc(x,y,l,f)
  fc11=f(1,1)
  fc12=f(1,2)
  fc13=f(1,3)
  fc14=f(1,4)
  fc15=f(1,5)
  fc16=f(1,6)
  fc21=f(2,1)
  fc22=f(2,2)
  fc23=f(2,3)
  fc24=f(2,4)
  fc25=f(2,5)
  fc26=f(2,6)
  fc31=f(3,1)
  fc32=f(3,2)
  fc33=f(3,3)
  fc34=f(3,4)
  fc35=f(3,5)
  fc36=f(3,6)
  fc41=f(4,1)
  fc42=f(4,2)
  fc43=f(4,3)
  fc44=f(4,4)

```

```

fc45=f(4,5)
fc46=f(4,6)
fc51=f(5,1)
fc52=f(5,2)
fc53=f(5,3)
fc54=f(5,4)
fc55=f(5,5)
fc56=f(5,6)
fc61=f(6,1)
fc62=f(6,2)
fc63=f(6,3)
fc64=f(6,4)
fc65=f(6,5)
fc66=f(6,6)
return
end

```

c

subroutine viewfc(x,y,l,f)

c purpose: determines the view factors for: aligned parallel rectangles

c and for: perpendicular rectangles with a common edge

c calls: vfactij

c called from: viewfact

dimension f(6,6)

parameter(pi=3.141593)

real l

a1=x*y

a2=x*l

a3=a1

a4=a2

a5=l*y

a6=a5

c find fij for aligned parallel rectangles (fig 13.4) Frank P. Incropera

c "fundamentals of heat and mass transfer"

do 10 i=1,3

if(i.eq.1)then

xb=x/l

yb=y/l

elseif(i.eq.2)then

xb=x/y

yb=l/y

elseif(i.eq.3)then

xb=y/x

yb=l/x

endif

a=alog((((1+xb**2)*(1+yb**2))/(1+xb**2+yb**2))**.5)

b=xb*(1+yb**2)**.5*atan(xb/(1+yb**2)**.5)

c=yb*(1+xb**2)**.5*atan(yb/(1+xb**2)**.5)

d=xb*atan(xb)

e=yb*atan(yb)

c if(i.eq.1)write(9,*)'a,b,c,d,e',a,b,c,d,e

fij=2/(pi*xb*yb)*(a+b+c-d-e)

```

      if(i.eq.1)f(1,3)=fij
      if(i.eq.2)f(2,4)=fij
      if(i.eq.3)f(5,6)=fij
10  continue
      f(3,1)=f(1,3)
      f(4,2)=f(2,4)
      f(6,5)=f(5,6)
      do 20 i=1,6
20  f(i,i)=0.

```

c find fij for perpendicular rectangles with a common edge

c (fig 13.6) Frank P. Incropera

c "fundamentals of heat and mass transfer"

c

c go about x-axis

```

      w=y/x
      h=l/x
      call vfacfij(w,h,fij)
      f(1,2)=fij
      f(3,4)=f(1,2)
      f(2,1)=f(1,2)*a1/a2
      f(4,3)=f(2,1)
      call vfacfij(h,w,fij)
      f(2,3)=fij
      f(4,1)=f(2,3)
      f(3,2)=f(2,3)*a2/a3
      f(1,4)=f(3,2)

```

c go about y-axis

```

      w=x/y
      h=l/y
      call vfacfij(w,h,fij)
      f(1,5)=fij
      f(3,6)=f(1,5)
      f(5,1)=f(1,5)*a1/a5
      f(6,3)=f(5,1)
      call vfacfij(h,w,fij)
      f(5,3)=fij
      f(6,1)=f(5,3)
      f(3,5)=f(5,3)*a5/a3
      f(1,6)=f(3,5)

```

c go about l-axis

```

      w=x/l
      h=y/l
      call vfacfij(w,h,fij)
      f(4,5)=fij
      f(2,6)=f(4,5)
      f(5,4)=f(4,5)*a4/a5
      f(6,2)=f(5,4)
      call vfacfij(h,w,fij)
      f(5,2)=fij
      f(6,4)=f(5,2)

```

```

f(2,5)=f(5,2)*a5/a2
f(4,6)=f(2,5)
c do 50 i=1,6
c 50 write(9,*)(f(i,j),j=1,6)
return
end
c
subroutine vfacfij(w,h,fij)
c purpose: find fij for perpendicular rectangles with a common edge
c (fig 13.6) Frank P. Incropera
c "fundamentals of heat and mass transfer"
c calls: none
c called from: viewfc
parameter(pi=3.141593)
a=w*atan(1/w)
b=h*atan(1/h)
c=(h**2+w**2)**.5*atan(1/(h**2+w**2)**.5)
d=((1+w**2)*(1+h**2))/(1+w**2+h**2)
e=((w**2*(1+w**2+h**2))/((1+w**2)*(w**2+h**2)))*w**2
f=((h**2*(1+h**2+w**2))/((1+h**2)*(h**2+w**2)))*h**2
c write(9,10)a,b,c,d,e,f
c 10 format(6f10.5)
fij=1/(pi*w)*(a+b-c+.25*alog(d*e*f))
return
end
c
c
subroutine newtraph(iperiod,k,theta,ta,tsky,hhf,hhw,hht,qdir,
& qtarm,ttarm,absorbf,emisf,absorbw,emisw,
& eemisf,eemisw,rff,rfw)
c purpose: this subroutine calls: linear solvers; ludcmp and lubksb.
c given an initial guess x for a root in n dimensions, take
c ntrial Newton-Raphson steps to improve the root. stop if the
c root converges in either summed absolute variable increments tol x or summed
c absolute function values tolf.
c calls: thermmdl,ludcmp,lubksb
c called from: solution
common/c1/n,ntrial,tolx,tolf
common/c2/x(200),fvec(200),jacob(200,200)
common/sc1/indx(200),p(200)
common/result/temp(200),temptrn(200)
c
c Variables:
c ipower=exponent of 10. to ensure solution converges in positive numbers
c
c ipower=1
50 do 14 k=1,ntrial
c user subroutine supplies function values at x in fvec and
c Jacobian matrix in jacob.
call thermmdl(iperiod,k,theta,ta,tsky,hhf,hhw,hht,qdir,
& qtarm,ttarm,absorbf,emisf,absorbw,emisw,

```

```

&          eemisf,eemism,rff,rw)
c check function convergence
  errf=0.
  do 11 i=1,n
    errf=errf+abs(fvec(i))
11  continue
c  if(errf.le.tolf) goto 20
c right-hand side of linear equations
  do 12 i=1,n
    p(i)=-fvec(i)
12  continue
c solve linear eqns. using LU decomposition
  call ludcmp
c forward and back substitution
  call lubksb
c check root convergence; and update solution
  errx=0.
  do 13 i=1,n
    errx=errx+abs(p(i))
    x(i)=x(i)+p(i)
13  continue
  write(14,*)'k(trial#),errx,errf,k,errx,errf
  if(errf.le.tolf.and.errx.le.tolx) goto 20
14  continue
20  loopindc=0
  do 30 i=1,n
    if(x(i).lt.0.0)then
      write(6,*)'i=',i,x(i)
      x(i)=10**ipower
      loopindc=1
    endif
30  continue
  if(loopindc.eq.1)then
    ipower=ipower+1
    goto 50
  endif
c rearrange all temperatures and radiosity in a neat manner as in
c Fig 1. of the report
  temp(1)=x(1)
  temp(2)=x(2)
  temp(3)=x(3)
  temp(4)=x(4)
  temp(5)=x(62)
  temp(6)=x(63)
  temp(7)=x(5)
  temp(8)=x(6)
  temp(9)=x(7)
  temp(10)=x(8)
  temp(11)=x(9)
  temp(12)=x(10)
  temp(13)=x(11)

```

temp(14)=x(12)
temp(15)=x(64)
temp(16)=x(65)
temp(17)=x(13)
temp(18)=x(14)
temp(19)=x(15)
temp(20)=x(16)
temp(21)=x(59)
temp(22)=x(58)
temp(23)=x(57)
temp(24)=x(60)
temp(25)=x(61)
temp(26)=x(35)
temp(27)=x(34)
temp(28)=x(33)
temp(29)=x(32)
temp(30)=x(31)
temp(31)=(x(31)+x(28))/2.
temp(32)=(x(31)+x(28))/2.
temp(33)=(x(31)+x(28))/2.
temp(34)=(x(31)+x(28))/2.
temp(35)=x(28)
temp(36)=x(27)
temp(37)=x(26)
temp(38)=x(25)
temp(39)=x(17)
temp(40)=x(18)
temp(41)=x(19)
temp(42)=x(20)
temp(43)=x(38)
temp(44)=x(39)
temp(45)=x(21)
temp(46)=x(22)
temp(47)=x(23)
temp(48)=x(24)
temp(49)=x(37)
temp(50)=x(36)
temp(51)=x(51)
temp(52)=x(52)
temp(53)=x(53)
temp(54)=x(40)
temp(55)=x(41)
temp(56)=x(42)
temp(57)=x(43)
temp(58)=x(44)
temp(59)=x(45)
temp(60)=x(46)
temp(61)=x(47)
temp(62)=x(48)
temp(63)=x(49)
temp(64)=x(50)

```
temp(65)=x(54)
temp(66)=x(55)
temp(67)=x(56)
do 100 i=1,53
100 temp(i)=temp(i)-459.67
return
end
```

c

```
subroutine thermodl(iperiod,ik,theta,ta,tsky,hhf,hhw,hht,qdir,
&                  qtarm,ttarm,absorbf,emisf,absorbw,emisw,
&                  eemisf,eemisw,rff,rfw)
```

c purpose: calculates and return the function value for each function fn

c and the jacobian values jacob; all at each new t values for each

c new newton-raphson trials. core of the thermal model

c called from: newtraph

c calls:jacobian

```
common/c1/n,ntrial,tolt,tolfn
common/c2/t(200),fn(200),jacob(200,200)
common/c3/th(100),k(100)
common/c4/widthwng,lengtwng,hwng,locwng,lplane,lfuselg,hfuselg,
&          wfuselg
common/c5/trnsm,absorbta,emista,tarmd,densta,conducta,shcta
common/c6/absorbff,emisff,absorbww,emisww,absorbfl,emisfl,absorbfi
&          emisfi,absorbpq,emispg,absorftp,emisrftp,absorfbt,emisrftb
&          absorfbf,emisfb
common/c7/nperiods,envtdata(100,20)
common/c10/aircondc,thdiffus,visckine,prandtl
common/c12/a1,a2,a3,a4,a5,a6(100),a7(100)
common/reflecq/qreflin(6,100),qscabsin(6,100),qreflflwr(6,100),
&          qreflflwl(6,100),qscabflwr(6,100),qscabflwl(6,100),qabstwa1(6,100),
&          qabstwa2(6,100),qabstwb(6,100)
common/vf1/fbwa(6,6),fbwb(6,6)
common/vf2/fa11,fa12,fa13,fa14,fa15,fa16,fa21,fa22,fa23,fa24,
&          fa25,fa26,fa31,fa32,fa33,fa34,fa35,fa36,fa41,fa42,
&          fa43,fa44,fa45,fa46,fa51,fa52,fa53,fa54,fa55,fa56,
&          fa61,fa62,fa63,fa64,fa65,fa66
common/vf4/fa57,fa68
common/vf5/fc11,fc12,fc13,fc14,fc15,fc16,fc21,fc22,fc23,fc24,
&          fc25,fc26,fc31,fc32,fc33,fc34,fc35,fc36,fc41,fc42,
&          fc43,fc44,fc45,fc46,fc51,fc52,fc53,fc54,fc55,fc56,
&          fc61,fc62,fc63,fc64,fc65,fc66
common/factarm/fact
dimension jj(20)
real jj,jacob,k,nusselt1,nusselt2
real lengtwng,locwng,lfuselg
parameter(fpi=3.141593/180.)
parameter(sig=0.1714e-8)
parameter(gravity=32.2)
data fact/0.3/
```

c

c Variables:

```

c nusselt1=Nusselt number inside the fuselage (vertical)
c nusselt2=Nusselt number inside the fuselage (horizontal)
c sig=Stefan-Boltzmann constant
c fact=allows for modification of temperature of the tarmac shade
c underneath the wing
c qreflin(,)=intercepted solar radiation (both direct and specular)
c      in the cabin
c qscabsin(,)=absorbed diffuse solar radiation in the cabin
c qreflwr(,)=intercepted solar radiation (both direct and specular)
c      by right wing
c qreflwl(,)=intercepted solar radiation (both direct and specular)
c      by left wing
c qscabfwr(,)=absorbed diffuse solar radiation by the right wing
c qscabfwl(,)=absorbed diffuse solar radiation by the left wing
c qabstwa1(,)=intercepted solar radiation (direct)
c      by the tarmac underneath the wing (LHS)
c qabstwa2(,)=intercepted solar radiation (direct)
c      by the tarmac underneath the wing      (RHS)
c qabstwb(,)=intercepted solar radiation (direct)
c      by the tarmac underneath the wing(CENTER)
c
c input t(from 40 to 50) into jj(1 to 11): jj=radiosities
  do 10 i=1,11
    10  jj(i)=t(39+i)
c input t(from 54 to 56) into jj(12 to 14): jj=radiosities
  do 15 i=1,3
    15  jj(11+i)=t(53+i)
c note t(51) to t(53) are for temperatures of tarmac under the wing
c
c find the values for each functions, fn, using the current t values
c
c left wing (conduction): Eqns. 1,2,3,4,5,6,62&63
c
  fn(1)=-k(1)/th(1)*(t(1)-t(2))+k(2)/th(2)*(t(2)-t(3))
  fn(2)=-k(2)/th(2)*(t(2)-t(3))+k(3)/th(3)*(t(3)-t(4))
  fn(3)=-k(3)/th(3)*(t(3)-t(4))+k(4)/th(4)*(t(4)-t(62))
  fn(4)=-k(4)/th(4)*(t(4)-t(62))+k(29)/th(29)*(t(62)-t(63))
  fn(62)=-k(29)/th(29)*(t(62)-t(63))+k(30)/th(30)*(t(63)-t(5))
  fn(63)=-k(30)/th(30)*(t(63)-t(5))+k(5)/th(5)*(t(5)-t(6))
  fn(5)=-k(5)/th(5)*(t(5)-t(6))+k(6)/th(6)*(t(6)-t(7))
  fn(6)=-k(6)/th(6)*(t(6)-t(7))+k(7)/th(7)*(t(7)-t(8))
c
c right wing (conduction): Eqns. 7,8,9,10,11,12,64&65
c
  fn(7)=-k(1)/th(1)*(t(9)-t(10))+k(2)/th(2)*(t(10)-t(11))
  fn(8)=-k(2)/th(2)*(t(10)-t(11))+k(3)/th(3)*(t(11)-t(12))
  fn(9)=-k(3)/th(3)*(t(11)-t(12))+k(4)/th(4)*(t(12)-t(64))
  fn(10)=-k(4)/th(4)*(t(12)-t(64))+k(29)/th(29)*(t(64)-t(65))
  fn(64)=-k(29)/th(29)*(t(64)-t(65))+k(30)/th(30)*(t(65)-t(13))
  fn(65)=-k(30)/th(30)*(t(65)-t(13))+k(5)/th(5)*(t(13)-t(14))
  fn(11)=-k(5)/th(5)*(t(13)-t(14))+k(6)/th(6)*(t(14)-t(15))

```

```

    fn(12)=-k(6)/th(6)*(t(14)-t(15))+k(7)/th(7)*(t(15)-t(16))
c
c left fuselage (conduction).  Eqns 13,14,17
  do 40 i=13,14
    fn(i)=-k(i-5)*((t(i+4)-t(i+5))/th(i-5))
    &      +k(i-4)*((t(i+5)-t(i+6))/th(i-4))
  40 continue
c Eqn. 17 recently modified
  fn(17)=-k(10)/th(10)*(t(19)-t(20))
  &      +k(23)/th(23)*(t(20)-t(38))
c right fuselage (conduction).  Eqns 15,16,18
  ii=0
  do 50 i=15,16
    fn(i)=-k(10-ii)*((t(i+6)-t(i+7))/th(10-ii))
    &      +k(9-ii)*((t(i+7)-t(i+8))/th(9-ii))
  ii=ii+1
  50 continue
c Eqn. 18 recently modified
  fn(18)=-k(23)/th(23)*(t(39)-t(21))
  &      +k(10)/th(10)*(t(21)-t(22))
c
c Conduction Eqns. 32,33,34,35,36,37,38,39,40,66&67. These eqns model
c the spar-box and floor section within the cabin. Green house effect
c is part of this addition. This effect is considered through
c JJ8 to JJ11 around the four sides of the cabin interior
c
c Eqns. 32,33&34:
c
  do 53 j=1,3
    fn(31+j)=-k(12+j)/th(12+j)*(t(26+j)-t(25+j))
    &      +k(11+j)/th(11+j)*(t(25+j)-t(24+j))
  53 continue
c
c Eqns. 37,38,39&40:
c
  do 55 j=1,4
    fn(36+j)=-k(17+j)/th(17+j)*(t(31+j)-t(30+j))
    &      +k(16+j)/th(16+j)*(t(30+j)-t(29+j))
  55 continue
c
c Eqns. 35,36,66&67:
c
  fn(35)=-k(32)/th(32)*(t(67)-t(29))+k(15)/th(15)*(t(29)-t(28))
  fn(36)=-k(31)/th(31)*(t(66)-t(67))+k(32)/th(32)*(t(67)-t(29))
  fn(66)=-k(16)/th(16)*(t(30)-t(66))+k(31)/th(31)*(t(66)-t(67))
  fn(67)=-k(17)/th(17)*(t(31)-t(30))+k(16)/th(16)*(t(30)-t(66))
c
c wing bottom, facing tarmac. there are a total of 12 eqns. to consider
c within three sections of the wing: LHS(section: 1); RHS (section 2)
c MIDDLE (section 3, ie underneath the fuselage). these eqns. are:
c LHS(section 1): fn19(t8),fn20(j2),fn51(t51),fn54(j12);

```

c RHS(section 2): fn21(t16),fn22(j4),fn53(t53),fn56(j14);
 c MIDDLE(section 3): fn31(t25),fn46(j7),fn52(t52),fn55(j13).
 c simple two-dimensional conduction from the unsheltered tarmac
 c surface to the sheltered surface is considered. a tarmac
 c thickness tarmd is assumed. also the bottom of the sheltered
 c tarmac is considered adiabatic (insulated) so that no q
 c is assumed to sip into the ground (being conservative).
 c also for the depth, tarmd, the temperature of the unsheltered
 c tarmac is assumed constant. note, qtarm and ttarm are known,
 c for they are unsheltered tarmac temperatures.

c
 qlwng=(qlwng+0.5*qtarm+0.5*(1.-emista)*qsky
 qlwng=(lengtwn-g-wfuselg)/2.

c
 c LHS(section 1): fn19(t8),fn20(j2),fn51(t51),fn54(j12):

c
 qnetsky=(sig*t(8)**4-jj(2))/((1.-emisw)/emisw)
 rad32skt=jj(14)*fbwa(4,1)+qskytarm*fbwa(4,2)+
 & qskytarm*fbwa(4,5)+qskytarm*fbwa(4,6)
 rad12skt=jj(13)*fbwb(4,1)+rad32skt*fbwb(4,2)+
 & qskytarm*fbwb(4,5)+qskytarm*fbwb(4,6)
 fn(19)=qabstwa1(3,iperiod)-qnetsky-hhw*(t(8)-ta)+
 & k(7)/th(7)*(t(7)-t(8))
 fn(20)=-qnetsky+
 & (jj(2)-jj(12))/(1./fbwa(3,1))+
 & (jj(2)-rad12skt)/(1./fbwa(3,2))+
 & (jj(2)-qskytarm)/(1./fbwa(3,4))+
 & (jj(2)-qskytarm)/(1./fbwa(3,5))+
 & (jj(2)-qskytarm)/(1./fbwa(3,6))
 qnetsky=(sig*t(51)**4-jj(12))/((1.-emista)/emista)
 fn(51)=(qabstwa1(1,iperiod)-qnetsky-hht*(t(51)-ta))*
 & qlwng*widthwng+
 & 2.*(conducta/(fact*widthwng)*(ttarm-t(51)))*tarmd*rlwng+
 & conducta/(fact*widthwng)*(ttarm-t(51))*tarmd*widthwng-
 & conducta/(.5*wfuselg+.5*rlwng)*(t(51)-t(52))*tarmd*widthwng
 rad32sk=qsky*fbwa(4,2)+jj(4)*fbwa(4,3)+qsky*fbwa(4,5)+
 & qsky*fbwa(4,6)
 rad12sk=rad32sk*fbwb(4,2)+jj(7)*fbwb(4,3)+
 & qsky*fbwb(4,5)+qsky*fbwb(4,6)
 fn(54)=-qnetsky+
 & (jj(12)-rad12sk)/(1./fbwa(1,2))+
 & (jj(12)-jj(2))/(1./fbwa(1,3))+
 & (jj(12)-qsky)/(1./fbwa(1,4))+
 & (jj(12)-qsky)/(1./fbwa(1,5))+
 & (jj(12)-qsky)/(1./fbwa(1,6))

c
 c RHS(section 2): fn21(t16),fn22(j4),fn53(t53),fn56(j14):

c
 qnetsky=(sig*t(16)**4-jj(4))/((1.-emisw)/emisw)
 rad34skt=jj(12)*fbwa(2,1)+qskytarm*fbwa(2,4)
 & +qskytarm*fbwa(2,5)+qskytarm*fbwa(2,6)

```

rad24skt=jj(13)*fbwb(2,1)+rad34skt*fbwb(2,4)
&      +qskytarm*fbwb(2,5)+qskytarm*fbwb(2,6)
fn(21)=qabstwa2(3,iperiod)-qnetsky-hhw*(t(16)-ta)+
&      k(7)/th(7)*(t(15)-t(16))
fn(22)=-qnetsky+
&      (jj(4)-jj(14))/(1./fbwa(3,1))+
&      (jj(4)-qskytarm)/(1./fbwa(3,2))+
&      (jj(4)-rad24skt)/(1./fbwa(3,4))+
&      (jj(4)-qskytarm)/(1./fbwa(3,5))+
&      (jj(4)-qskytarm)/(1./fbwa(3,6))
qnetsky=(sig*t(53)**4-jj(14))/((1.-emista)/emista)
fn(53)=(qabstwa2(1,iperiod)-qnetsky-hht*(t(53)-ta))*
&      rlwng*widthwng+
&      2.*(conducta/(fact*widthwng)*(ttarm-t(53)))*tarmd*rlwng+
&      conducta/(fact*widthwng)*(ttarm-t(53))*tarmd*widthwng-
&      conducta/(.5*wfuselg+.5*rlwng)*(t(53)-t(52))*tarmd*widthwng
rad34sk=jj(2)*fbwa(2,3)+qsky*fbwa(2,4)+qsky*fbwa(2,5)+
&      qsky*fbwa(2,6)
rad24sk=jj(7)*fbwb(2,3)+rad34sk*fbwb(2,4)+
&      qsky*fbwb(2,5)+qsky*fbwb(2,6)
fn(56)=-qnetsky+
&      (jj(14)-qsky)/(1./fbwa(1,2))+
&      (jj(14)-jj(4))/(1./fbwa(1,3))+
&      (jj(14)-rad24sk)/(1./fbwa(1,4))+
&      (jj(14)-qsky)/(1./fbwa(1,5))+
&      (jj(14)-qsky)/(1./fbwa(1,6))

```

c

c MIDDLE(section 3): fn31(t25),fn46(j7),fn52(t52),fn55(j13).

c

```

qnetsky=(sig*t(25)**4-jj(7))/((1.-emisfb)/emisfb)
fn(31)=qabstwb(3,iperiod)-qnetsky-hhf*(t(25)-ta)+
&      k(12)/th(12)*(t(26)-t(25))
fn(46)=-qnetsky+
&      (jj(7)-jj(13))/(1./fbwb(3,1))+
&      (jj(7)-rad32skt)/(1./fbwb(3,2))+
&      (jj(7)-rad34skt)/(1./fbwb(3,4))+
&      (jj(7)-qskytarm)/(1./fbwb(3,5))+
&      (jj(7)-qskytarm)/(1./fbwb(3,6))
qnetsky=(sig*t(52)**4-jj(13))/((1.-emista)/emista)
fn(52)=(qabstwb(1,iperiod)-qnetsky-hht*(t(52)-ta))*
&      wfuselg*widthwng+
&      2.*(conducta/(fact*widthwng)*(ttarm-t(52)))*tarmd*wfuselg+
&      conducta/(.5*wfuselg+.5*rlwng)*(t(51)-t(52))*tarmd*widthwng+
&      conducta/(.5*wfuselg+.5*rlwng)*(t(53)-t(52))*tarmd*widthwng
fn(55)=-qnetsky+
&      (jj(13)-rad32sk)/(1./fbwb(1,2))+
&      (jj(13)-jj(7))/(1./fbwb(1,3))+
&      (jj(13)-rad34sk)/(1./fbwb(1,4))+
&      (jj(13)-qsky)/(1./fbwb(1,5))+
&      (jj(13)-qsky)/(1./fbwb(1,6))

```

c

c equations for the fuselage-wing junction facing the sky
 c there are a total of 8 eqns. (4 for left half and 4 for right half)
 c consider the wings first (a total of 4 eqns: 2 on each side)
 c eqns. 23, 24, 25 & 26.
 c
 c when ii=1 (eqn. 23 & 24), left hand side; when ii=2 (eqn. 25 & 26),
 c right hand side
 do 80 ii=1,2
 qnetsky=(sig*t(8*(ii-1)+1)**4-rfw*jj(1+(ii-1)*2))/
 & ((1.-rfw*eemisw)/eemisw)
 if(ii.eq.1)fn(23+(ii-1)*2)=absorbw*qreflflw(1,iperiod)*
 & sin(fpi*theta)+qscabfwr(1,iperiod)-qnetsky-hhw*
 & (t(8*(ii-1)+1)-ta)-k(1)*((t(8*(ii-1)+1)-t(8*(ii-1)+2))/th(1))
 if(ii.eq.2)fn(23+(ii-1)*2)=absorbw*qreflflw(1,iperiod)*
 & sin(fpi*theta)+qscabfwr(1,iperiod)-qnetsky-hhw*
 & (t(8*(ii-1)+1)-ta)-k(1)*((t(8*(ii-1)+1)-t(8*(ii-1)+2))/th(1))
 fn(24+(ii-1)*2)=-qnetsky
 & +(((jj(1+(ii-1)*2)-jj(5+(ii-1))))/(1./fa12))
 & +((jj(1+(ii-1)*2)-qsky)/(1./fa13))
 & +((jj(1+(ii-1)*2)-qsky)/(1./fa14))
 & +((jj(1+(ii-1)*2)-qsky-fa57*(emisf*sig*t(17+(ii-1)*7)
 & **4+(1-emisf)*(qsky+.5*qtarm+.5*(1.-emista)*qsky)))/(1./fa15))
 & +((jj(1+(ii-1)*2)-qsky-fa68*(emisf*sig*t(17+(ii-1)*7)
 & **4+(1-emisf)*(qsky+.5*qtarm+.5*(1.-emista)*qsky)))/(1./fa16)))

80 continue

c now consider the 4 eqns. of the above related to the fuselage.
 c 2 eqns. on the left hand side (eqns. 27 & 28) and 2 eqns. on
 c the right hand side (eqns. 29 & 30).

c
 c when ii=1, left hand side; when ii=2, right hand side

do 90 ii=1,2
 qnetsky=(sig*t(17+(ii-1)*7)**4-rff*jj(5+(ii-1)))/
 & ((1.-rff*eemisf)/eemisf)
 if(ii.eq.1)fn(27+(ii-1)*2)=absorbw*qreflflw(2,iperiod)*
 & cos(fpi*theta)+qscabfwr(2,iperiod)-qnetsky-hhf*
 & (t(17+(ii-1)*7)-ta)+
 & (-1)**ii*k(8)*((t(17+(ii-1)*6)-t(18+(ii-1)*6))/th(8))
 if(ii.eq.2)fn(27+(ii-1)*2)=qscabfwr(2,iperiod)-qnetsky-hhf*
 & (t(17+(ii-1)*7)-ta)+
 & (-1)**ii*k(8)*((t(17+(ii-1)*6)-t(18+(ii-1)*6))/th(8))
 fn(28+(ii-1)*2)=-qnetsky
 & +((jj(5+(ii-1))-jj(1+(ii-1)*2))/(1/fa21))
 & +((jj(5+(ii-1))-qsky)/(1/fa23))
 & +((jj(5+(ii-1))-qsky-.5*qtarm-.5*(1.-emista)*qsky)/(1/fa24))
 & +((jj(5+(ii-1))-qsky-.5*qtarm-.5*(1.-emista)*qsky)/(1/fa25))
 & +((jj(5+(ii-1))-qsky-.5*qtarm-.5*(1.-emista)*qsky)/(1/fa26))

90 continue

c Recent modification has added a total of 9 eqns within the cabin.
 c These eqns. are : 4 temperatures within the cabin (Eqn. 41,42,44,45);
 c 1 temperature outside plexiglass (Eqn. 43); and 4 JJ's within
 c the cabin (Eqns. 47,48,49,50).

c

c In order to consider the effects of the roof cover, five
 c equations (eqn. 57, 58, 59, 60 & 61) have been added, giving temperatures
 c t57, t58, t59, t60 & t61. note that in these equations, areas are
 c kept to indicate their comparative influence.
 c also note that eqns. 42 & 43 have been modified.
 c surface 4 (combined roof and glass bottom) is treated as one
 c unit having a uniform temperature, tprf; uniform radiosity JJ9; and
 c uniform emissivity; in order to obtain a uniform qnet within the cabin.

c

c equation 58 (conduction):

$$\begin{aligned} \text{fn}(58) &= k(27)/th(27)*(t(59)-t(58))*a2 \\ &\& -k(26)/th(26)*(t(58)-t(57))*a2 \end{aligned}$$

c equation 57 (conduction):

$$\begin{aligned} \text{fn}(57) &= k(26)/th(26)*(t(58)-t(57))*a2 \\ &\& -k(25)/th(25)*(t(57)-t(60))*a2 \end{aligned}$$

c equation 60 (conduction):

$$\begin{aligned} \text{fn}(60) &= k(25)/th(25)*(t(57)-t(60))*a2 \\ &\& -k(28)/th(28)*(t(60)-t(61))*a2 \end{aligned}$$

c

c top of plexiglass and roof top facing the sky & tarmac.
 c eqn 43 (t37(plexiglass) & eqn 59 (t59 (roof top))

c

$$\begin{aligned} \text{fn}(43) &= -\text{emispg}*\text{sig}*t(37)**4*a1 \\ &\& -\text{hhf}*(t(37)-\text{ta})*a1 \\ &\& -k(22)/th(22)*(t(37)-t(36))*a1 \\ &\& +\text{emispg}*qsky*a1 \\ &\& +\text{emispg}*.5*(1.-\text{emista})*qsky*a5 \\ &\& +\text{emispg}*.5*qtarm*a5 \\ &\& +.5*\text{absorbpg}*qdir*\cos(\text{fpi}*\text{theta})*a6(\text{iperiod}) \\ &\& +.5*\text{absorbpg}*qdir*\sin(\text{fpi}*\text{theta})*a7(\text{iperiod}) \\ &\& +.5*\text{absorbpg}*.5*(1.-\text{absorbta})*qdir*\sin(\text{fpi}*\text{theta})*a5 \\ &\& +.5*\text{absorbpg}*qreflin(4,\text{iperiod})*\sin(\text{fpi}*\text{theta})*a3 \\ &\& +.5*\text{absorbpg}*qscabsin(4,\text{iperiod})*\text{lfuselg}*w\text{fuselg} \\ &\& *a3/(\text{absorbpg}*a3+\text{absorfbt}*a4) \end{aligned}$$

c

$$\begin{aligned} \text{fn}(59) &= -\text{emisrftp}*\text{sig}*t(59)**4*a2 \\ &\& -\text{hhf}*(t(59)-\text{ta})*a2 \\ &\& -k(27)/th(27)*(t(59)-t(58))*a2 \\ &\& +\text{emisrftp}*qsky*a2 \\ &\& +\text{absorftp}*qdir*\sin(\text{fpi}*\text{theta})*a2 \end{aligned}$$

c

c find the weighted average temperature of the ends, tend

$$\begin{aligned} \text{aa1} &= \text{lfuselg}*w\text{fuselg} \\ \text{aa2} &= \text{a1} \\ \text{aa3} &= \text{lfuselg}*h\text{fuselg} \\ \text{aa4} &= \text{aa3} \\ \text{aa5} &= \text{a2} \\ \text{atot} &= \text{aa1}+\text{aa2}+\text{aa3}+\text{aa4}+\text{aa5} \\ \text{cc1} &= \text{aa1}/\text{atot} \\ \text{cc2} &= \text{aa2}/\text{atot} \end{aligned}$$

```
cc3=aa3/atot
cc4=aa4/atot
cc5=aa5/atot
tend=cc1*t(35)+cc2*t(36)+cc3*t(38)+cc4*t(39)+cc5*t(61)
```

c
 c for eqns: 42 (bottom of plexiglass facing the cabin floor below);
 c 41 (floor of cabin inside (surface B)); 44 (left hand side of cabin
 c inside (surface C)); and 45 (right hand side of cabin inside
 c (surface A)): the effective convection coefficient must be found.
 c these coeffs. are coupled with the surface temperatures. therefore
 c need to read in thermal diffusivity (thdiffus); kinematic viscosity
 c (visckine); gravity (gravity); and Prandtl # (prandtl).
 c from these, Rayleigh number (rayleigh); and Nusselt # (nusselt)
 c must be found within the system of equation. keff=k * nusselt.

c
 c find the weighted plexiglass and roof bottom temperatures:

```
c1=a1/(a1+a2)
c2=a2/(a1+a2)
tpgrf=c1*t(36)+c2*t(61)
```

c
 rayleig1=gravity*1./((tpgrf+t(35))/2.)*(t(35)-tpgrf)*th(24)**3/
 & (thdiffus*visckine)
 if(t(35).lt.0.0.or.tpgrf.lt.0.0)then
 nusselt1=1.0
 else
 if(rayleig1.le.3e5)then
 nusselt1=1.0
 elseif(rayleig1.gt.3e5.and.rayleig1.le.7e9)then
 nusselt1=0.069*rayleig1**(1./3.)*prandtl**.074
 elseif(rayleig1.gt.7e9)then
 nusselt1=0.069*7e9**(1./3.)*prandtl**.074
 c this is to ensure a reasonable nusselt1 number. we do not want
 c a runaway solution during the Newton-Raphson solution of the
 c system of non-linear equations.
 c write(9,*)'Rayleigh number exceeded 7e9 for vertical'
 endif
 endif
 c write(9,*)'vertical',iperiod,ik,t(35),tpgrf,rayleig1,nusselt1
 rayleig2=gravity*1./((t(38)+t(39))/2.)*abs(t(38)-t(39))*
 & th(11)**3/(thdiffus*visckine)
 if(t(38).lt.0.0.or.t(39).lt.0.0)then
 nusselt2=1.0
 else
 if(rayleig2.le.(1e3*(0.2+prandtl)/prandtl))then
 nusselt2=1.0
 elseif(rayleig2.gt.(1e3*(0.2+prandtl)/prandtl).and.rayleig2.lt.
 & 1.0e9)then
 nusselt2=0.18*((prandtl/(0.2+prandtl))*rayleig2)**0.29
 elseif(rayleig2.gt.1.0e9)then
 nusselt2=0.18*((prandtl/(0.2+prandtl))*1.0e9)**0.29
 c this is to ensure a reasonable nusselt1 number. we do not want

c a runaway solution during the Newton-Raphson solution of the
 c system of non-linear equations.

c write(9,*)'Rayleigh number exceeded 1e9 for horizontal'
 endif

 endif

c write(9,*)'horizontal',iperiod,ik,t(38),t(39),rayleig2,nusselt2

c

c bottom of plexiglass facing the cabin floor below.

c eqns: eqn 42 (t36);& eqn 48 (JJ9)

c the plexiglass and roof bottom are treated as "one" unit,

c (surface 4), for determining radiosity JJ9 and qnet

 emispg=emispgrf/(a1+a2)*emispg+a2/(a1+a2)*emisrft

 qnet=(sig*tpgrf**4-jj(9))/((1.-emispg)/emispg)

 qpg=emispg*sig*t(36)**4-emisp/(a3*emispg+a4*emisrft)*

 & (a3*emispg*sig*t(36)**4+a4*emisrft*sig*t(61)**4-qnet*(a3+a4))

 fn(42)=-qpg*a3

 & -nusselt1*k(24)/th(24)*(tpgrf-t(35))*a3

 & +k(22)/th(22)*(t(37)-t(36))*a1

 & +.5*absorbpq*mdir*cos(fpi*theta)*a6(iperiod)

 & +.5*absorbpq*mdir*sin(fpi*theta)*a7(iperiod)

 & +.5*absorbpq*.5*(1.-absorbta)*mdir*sin(fpi*theta)*a5

 & +.5*absorbpq*qreflin(4,iperiod)*sin(fpi*theta)*a3

 & +.5*absorbpq*qscabsin(4,iperiod)*lfuselg*wfuselg

 & *a3/(absorbpq*a3+absorfbt*a4)

c fn48:

 fn(48)=-qnet+(jj(9)-jj(11))/(1./fc31)+(jj(9)-jj(10))/(1./fc32)

 & +(jj(9)-jj(8))/(1./fc34)+(jj(9)-sig*tend**4)/(1./fc35)

 & +(jj(9)-sig*tend**4)/(1./fc36)

c

c equation 61 (t61) (roof bottom surface):

 qrft=emisrft*sig*t(61)**4-emisrft/(a3*emispg+a4*emisrft)*

 & (a3*emispg*sig*t(36)**4+a4*emisrft*sig*t(61)**4-qnet*(a3+a4))

 fn(61)=-qrft*a4

 & -nusselt1*k(24)/th(24)*(tpgrf-t(35))*a4

 & +k(28)/th(28)*(t(60)-t(61))*a2

 & +absorfbt*qreflin(4,iperiod)*sin(fpi*theta)*a4

 & +absorfbt*qscabsin(4,iperiod)*lfuselg*wfuselg

 & *a4/(absorbpq*a3+absorfbt*a4)

c

c left hand side of cabin inside (surface C)

c eqn 44 (t38);& eqn 47 (JJ8)

 qnet=(sig*t(38)**4-jj(8))/((1.-emisfi)/emisfi)

 fn(44)=-qnet-nusselt2*k(11)/th(11)*(t(38)-t(39))

 & +k(23)/th(23)*(t(20)-t(38))

 & +absorbfq*qreflin(3,iperiod)*cos(fpi*theta)

 & +qscabsin(3,iperiod)

 fn(47)=-qnet+(jj(8)-jj(11))/(1./fc41)+(jj(8)-jj(10))/(1./fc42)

 & +(jj(8)-jj(9))/(1./fc43)+(jj(8)-sig*tend**4)/(1./fc45)

 & +(jj(8)-sig*tend**4)/(1./fc46)

c right hand side of cabin inside (surface A)

c eqn 45 (t39);& eqn 49 (JJ10)

```

qnet=(sig*t(39)**4-jj(10))/((1.-emisfi)/emisfi)
fn(45)=-qnet-k(23)/th(23)*(t(39)-t(21))
& +nusselt2*k(11)/th(11)*(t(38)-t(39))
& +absorbfi*qreflin(1,iperiod)*cos(fpi*theta)
& +qscabsin(1,iperiod)
fn(49)=-qnet+(jj(10)-jj(11))/(1./fc21)+(jj(10)-jj(9))/(1./fc23)
& +(jj(10)-jj(8))/(1./fc24)+(jj(10)-sig*tend**4)/(1./fc25)
& +(jj(10)-sig*tend**4)/(1./fc26)
c floor of cabin inside (surface B)
c eqn 41 (t35):& eqn 50 (JJ11)
qnet=(sig*t(35)**4-jj(11))/((1.-emisfl)/emisfl)
fn(41)=-qnet-k(21)/th(21)*(t(35)-t(34))
& +nusselt1*k(24)/th(24)*(tpgrf-t(35))
& +absorbfl*qreflin(2,iperiod)*sin(fpi*theta)
& +qscabsin(2,iperiod)
fn(50)=-qnet+(jj(11)-jj(10))/(1./fc12)+(jj(11)-jj(9))/(1./fc13)
& +(jj(11)-jj(8))/(1./fc14)+(jj(11)-sig*tend**4)/(1./fc15)
& +(jj(11)-sig*tend**4)/(1./fc16)
c
c now find the jacobian values for the current t values
c
  call jacobian(hhf,hhw,hht,emisf,emisw,eemisf,eemisw,rff,
&              rfw,nusselt1,nusselt2,c1,c2,cc1,cc2,cc3,
&              cc4,cc5,tend,tpgrf,emispgrf,rlwng)
c
c just a dummy for using ik, (otherwise compiler warning)
c ik maybe necessary for Rayleigh check, in case during debugging
  dummyik=ik
  return
  end
c
  subroutine jacobian(hhf,hhw,hht,emisf,emisw,eemisf,eemisw,rff,
&                    rfw,nusselt1,nusselt2,c1,c2,cc1,cc2,cc3,
&                    cc4,cc5,tend,tpgrf,emispgrf,rlwng)
c purpose: calculates and return the jacobian values
c at each new t values
c called from: thermodl
c calls:none
  common/c1/n,ntrial,tolt,tolfn
  common/c2/t(200),fn(200),jacob(200,200)
  common/c3/th(100),k(100)
  common/c4/widthwng,lengthwng,hwng,locwng,lplane,lfuselg,hfuselg,
&          wfuselg
  common/c5/trnsm,absorbta,emista,tarnd,densta,conducta,shcta
  common/c6/absorbff,emisff,absorbww,emisww,absorbfl,emisfl,absorbfi
& ,emisfi,absorbpq,emispq,absorftp,emisrftp,absorfbt,emisrftb
& ,absorfbf,emisfb
  common/c7/nperiods,envtdata(100,20)
  common/c10/aircondc,thdiffus,visckine,prandtl
  common/c12/a1,a2,a3,a4,a5,a6(100),a7(100)
  common/vf1/fbwa(6,6),fbwb(6,6)

```

```

common/vf2/fa11,fa12,fa13,fa14,fa15,fa16,fa21,fa22,fa23,fa24,
&      fa25,fa26,fa31,fa32,fa33,fa34,fa35,fa36,fa41,fa42,
&      fa43,fa44,fa45,fa46,fa51,fa52,fa53,fa54,fa55,fa56,
&      fa61,fa62,fa63,fa64,fa65,fa66
common/vf4/fa57,fa68
common/vf5/fc11,fc12,fc13,fc14,fc15,fc16,fc21,fc22,fc23,fc24,
&      fc25,fc26,fc31,fc32,fc33,fc34,fc35,fc36,fc41,fc42,
&      fc43,fc44,fc45,fc46,fc51,fc52,fc53,fc54,fc55,fc56,
&      fc61,fc62,fc63,fc64,fc65,fc66
common/facttarm/fact
real jacob,k,nusselt1,nusselt2
parameter(sig=0.1714e-8)
parameter(gravity=32.2)
c
c initialize: (most jacobian coefficients are zero)
do 110 i=1,n
do 110 j=1,n
jacob(i,j)=0.
110 continue
c
c left-wing conduction (eqn. 1 & 2)
c
do 120 i=1,2
jacob(i,i)=-k(i)/th(i)
jacob(i,i+1)=k(i)/th(i)+k(i+1)/th(i+1)
jacob(i,i+2)=-k(i+1)/th(i+1)
120 continue
c
c left-wing conduction (eqn. 5 & 6)
c
do 125 i=5,6
jacob(i,i)=-k(i)/th(i)
jacob(i,i+1)=k(i)/th(i)+k(i+1)/th(i+1)
jacob(i,i+2)=-k(i+1)/th(i+1)
125 continue
c
c left-wing conduction (eqn. 3,4,62&63)
c
jacob(3,3)=-k(3)/th(3)
jacob(3,4)=k(3)/th(3)+k(4)/th(4)
jacob(3,62)=-k(4)/th(4)
c
jacob(4,4)=-k(4)/th(4)
jacob(4,62)=k(4)/th(4)+k(29)/th(29)
jacob(4,63)=-k(29)/th(29)
c
jacob(62,5)=-k(30)/th(30)
jacob(62,62)=-k(29)/th(29)
jacob(62,63)=k(29)/th(29)+k(30)/th(30)
c
jacob(63,5)=k(30)/th(30)+k(5)/th(5)

```

```

  jacob(63,6)=-k(5)/th(5)
  jacob(63,63)=-k(30)/th(30)
c
c right-wing conduction (eqn. 7 & 8)
c
  do 130 i=1,2
    jacob(i+6,i+8)=-k(i)/th(i)
    jacob(i+6,i+9)=k(i)/th(i)+k(i+1)/th(i+1)
    jacob(i+6,i+10)=-k(i+1)/th(i+1)
  130 continue
c
c right-wing conduction (eqn. 11 & 12)
c
  do 135 i=5,6
    jacob(i+6,i+8)=-k(i)/th(i)
    jacob(i+6,i+9)=k(i)/th(i)+k(i+1)/th(i+1)
    jacob(i+6,i+10)=-k(i+1)/th(i+1)
  135 continue
c
c right-wing conduction (eqn. 9,10,64&65)
c
  jacob(9,11)=-k(3)/th(3)
  jacob(9,12)=k(3)/th(3)+k(4)/th(4)
  jacob(9,64)=-k(4)/th(4)
c
  jacob(10,12)=-k(4)/th(4)
  jacob(10,64)=k(4)/th(4)+k(29)/th(29)
  jacob(10,65)=-k(29)/th(29)
c
  jacob(64,13)=-k(30)/th(30)
  jacob(64,64)=-k(29)/th(29)
  jacob(64,65)=k(29)/th(29)+k(30)/th(30)
c
  jacob(65,13)=k(30)/th(30)+k(5)/th(5)
  jacob(65,14)=-k(5)/th(5)
  jacob(65,65)=-k(30)/th(30)
c
c left fuselage conduction (eqn. 13 to 14)
  do 140 i=1,2
    jacob(i+12,i+16)=-k(i+7)/th(i+7)
    jacob(i+12,i+17)=k(i+7)/th(i+7)+k(i+8)/th(i+8)
    jacob(i+12,i+18)=-k(i+8)/th(i+8)
  140 continue
c add eqn. 17, recently modified
  jacob(17,19)=-k(10)/th(10)
  jacob(17,20)=k(10)/th(10)+k(23)/th(23)
  jacob(17,38)=-k(23)/th(23)
c right fuselage conduction (eqn. 15 to 16)
  do 150 i=1,2
    jacob(i+14,i+20)=-k(11-i)/th(11-i)
    jacob(i+14,i+21)=k(11-i)/th(11-i)+k(10-i)/th(10-i)

```

```

      jacob(i+14,i+22)=-k(10-i)/th(10-i)
150 continue
c add eqn. 18, recently modified
      jacob(18,21)=k(23)/th(23)+k(10)/th(10)
      jacob(18,22)=-k(10)/th(10)
      jacob(18,39)=-k(23)/th(23)
c fuselage-wing junction
c wing portion (i=1 implies left hand side. i=2 implies right hand side)
c (eqns. 23 to 26)
      do 180 i=1,2
        jacob(23+(i-1)*2,1+(i-1)*8)=-((4*sig*t(1+(i-1)*8)**3)/
&      ((1-rfw*eemisw)/eemisw)-hhw-k(1)/th(1)
        jacob(23+(i-1)*2,2+(i-1)*8)=k(1)/th(1)
        jacob(23+(i-1)*2,40+(i-1)*2)=1.*rfw/((1.-rfw*eemisw)/eemisw)
        jacob(24+(i-1)*2,1+(i-1)*8)=-((4*sig*t(1+(i-1)*8)**3)/
&      ((1-rfw*eemisw)/eemisw)
        jacob(24+(i-1)*2,17+(i-1)*7)=-((4*fa57*emisf*sig*t(17+(i-1)*7)**3
&      /(1/fa15))-((4*fa68*emisf*sig*t(17+(i-1)*7)**3/(1/fa16)))
        jacob(24+(i-1)*2,40+(i-1)*2)=1.*rfw/((1.-rfw*eemisw)/eemisw)
&      +(1/(1/fa12)+1/(1/fa13)+1/(1/fa14)+1/(1/fa15)+1/
&      (1/fa16))
        jacob(24+(i-1)*2,44+(i-1))=-1/(1/fa12)
180 continue
c fuselage portion (i=1 implies left hand side. i=2 implies right hand side)
c (eqns. 27 to 30)
      do 190 i=1,2
        jacob(27+(i-1)*2,17+(i-1)*7)=-((4*sig*t(17+(i-1)*7)**3)/
&      ((1-rff*eemisf)/eemisf)-hhf-k(8)/th(8)
        jacob(27+(i-1)*2,18+(i-1)*5)=k(8)/th(8)
        jacob(27+(i-1)*2,44+(i-1))=1.*rff/((1.-rff*eemisf)/eemisf)
        jacob(28+(i-1)*2,17+(i-1)*7)=-((4*sig*t(17+(i-1)*7)**3)/
&      ((1-rff*eemisf)/eemisf)
        jacob(28+(i-1)*2,40+(i-1)*2)=-1/(1/fa21)
        jacob(28+(i-1)*2,44+(i-1))=1.*rff/((1.-rff*eemisf)/eemisf)
&      +(1/(1/fa21)+1/(1/fa23)+1/(1/fa24)+1/(1/fa25)+1/
&      (1/fa26))
190 continue
c
c Eqns. 32...40; & 66,67 for conduction in the middle section of fuselage
c (through: bottom skin cover; through the spar; through the
c the upper skin of the spar box cap)
c
c Eqns. 32,33 & 34:
c
      do 195 i=1,3
        jacob(31+i,24+i)=-k(11+i)/th(11+i)
        jacob(31+i,25+i)=k(12+i)/th(12+i)+k(11+i)/th(11+i)
        jacob(31+i,26+i)=-k(12+i)/th(12+i)
195 continue
c
c Eqns. 37,38,39 & 40:

```

```

c
  do 200 i=6,9
    jacob(31+i,24+i)=-k(11+i)/th(11+i)
    jacob(31+i,25+i)=k(12+i)/th(12+i)+k(11+i)/th(11+i)
    jacob(31+i,26+i)=-k(12+i)/th(12+i)
  200 continue
c
c Eqns. 35,36,66&67:
c
  jacob(35,28)=-k(15)/th(15)
  jacob(35,29)=k(32)/th(32)+k(15)/th(15)
  jacob(35,67)=-k(32)/th(32)
c
  jacob(36,29)=-k(32)/th(32)
  jacob(36,66)=-k(31)/th(31)
  jacob(36,67)=k(31)/th(31)+k(32)/th(32)
c
  jacob(66,30)=-k(16)/th(16)
  jacob(66,66)=k(16)/th(16)+k(31)/th(31)
  jacob(66,67)=-k(31)/th(31)
c
  jacob(67,30)=k(17)/th(17)+k(16)/th(16)
  jacob(67,31)=-k(17)/th(17)
  jacob(67,66)=-k(16)/th(16)
c
c add jacobian 43 for top of plexiglass:
  jacob(43,36)=k(22)/th(22)*a1
  jacob(43,37)=-4*emispg*sig*t(37)**3*a1
  &          -hhf*a1
  &          -k(22)/th(22)*a1
c add jacobian 59 for top of roof cover:
  jacob(59,58)=k(27)/th(27)*a2
  jacob(59,59)=-4*emisrftp*sig*t(59)**3*a2
  &          -hhf*a2
  &          -k(27)/th(27)*a2
c
c jacobian 57, 58 & 60 for internal conduction of roof cover:
c jacobian(57; 57, 58, 60):
  jacob(57,57)=-k(26)/th(26)*a2
  &          -k(25)/th(25)*a2
  jacob(57,58)=k(26)/th(26)*a2
  jacob(57,60)=k(25)/th(25)*a2
c jacobian(58: 57,58,59):
  jacob(58,57)=k(26)/th(26)*a2
  jacob(58,58)=-k(27)/th(27)*a2
  &          -k(26)/th(26)*a2
  jacob(58,59)=k(27)/th(27)*a2
c jacobian(60: 57, 60, 61)
  jacob(60,57)=k(25)/th(25)*a2
  jacob(60,60)=-k(25)/th(25)*a2
  &          -k(28)/th(28)*a2

```

$$\text{jacob}(60,61)=k(28)/th(28)*a2$$

c

c for eqns: 42 (bottom of plexiglass facing the cabin floor below);
 c 41 (floor of cabin inside (surface B)); 44 (left hand side of cabin
 c inside (surface C)); and 45 (right hand side of cabin inside
 c (surface A)); the effective convection coefficient must be found.
 c these coeffs. are coupled with the surface temperatures. therefore
 c need to read in thermal diffusivity (thdiffus); kinematic viscosity
 c (visckine); gravity (gravity); and Prandtl # (prandtl).
 c from these, Rayleigh number (rayleigh); and Nusselt # (nusselt)
 c must be found within the system of equation. $keff=k * nusselt$.

c

c

c add jacobians 42, 61 & 48 (J9) for bottom of plexiglass

c and the bottom of roof cover (all inside cabin)

c jacobian 42:

```

if(nusselt1.eq.1.0)then
  jacob(42,35)=nusselt1*k(24)/th(24)*a3
  jacob(42,36)=-((4*emisp*sig*t(36)**3-
    &      emisp/(a3*emisp+a4*emisrftb)*
    &      (4*a3*emisp*sig*t(36)**3-(a3+a4)/((1.-emispgrf)/emispgrf)*
    &      (4*sig*tpgrf**3*c1)))a3
    &      -nusselt1*k(24)/th(24)*c1*a3
    &      -k(22)/th(22)*a1
  jacob(42,37)=k(22)/th(22)*a1
  jacob(42,48)=emisp/(a3*emisp+a4*emisrftb)*
    &      (a3+a4)/((1.-emispgrf)/emispgrf)*a3
  jacob(42,61)=-((emisp/(a3*emisp+a4*emisrftb)*
    &      (4*a4*emisrftb*sig*t(61)**3-(a3+a4)/((1.-emispgrf)/emispgrf)*
    &      (4*sig*tpgrf**3*c2)))a3
    &      -nusselt1*k(24)/th(24)*c2*a3
else
  value1=-0.069*(gravity/(.5*thdiffus*visckine))**(1./3.)
  &      *prandtl**0.074*th(24)
  partl35=(t(35)+tpgrf)**(-1./3.)*(4./3.)*(t(35)-tpgrf)**(1./3.)
  &      -(t(35)-tpgrf)**(4./3.)*(1./3.)*(t(35)+tpgrf)**(-4./3.)
  facnusel=-value1*partl35
  jacob(42,35)=facnusel*k(24)/th(24)*a3
  partl36=-(t(35)+tpgrf)**(-1./3.)*(4./3.)*(t(35)-tpgrf)**(1./3.)
  &      -(t(35)-tpgrf)**(4./3.)*(1./3.)*(t(35)+tpgrf)**(-4./3.)
  facnusel=value1*partl36
  jacob(42,36)=-((4*emisp*sig*t(36)**3-
    &      emisp/(a3*emisp+a4*emisrftb)*
    &      (4*a3*emisp*sig*t(36)**3-(a3+a4)/((1.-emispgrf)/emispgrf)*
    &      (4*sig*tpgrf**3*c1)))a3
    &      -facnusel*k(24)/th(24)*c1*a3
    &      -k(22)/th(22)*a1
  jacob(42,37)=k(22)/th(22)*a1
  jacob(42,48)=emisp/(a3*emisp+a4*emisrftb)*
    &      (a3+a4)/((1.-emispgrf)/emispgrf)*a3
  partl61=-(t(35)+tpgrf)**(-1./3.)*(4./3.)*(t(35)-tpgrf)**(1./3.)

```

```

&      -(t(35)-tpgrf)**(4./3.)*(1./3.)*(t(35)+tpgrf)**(-4./3.)
facnuser=value1*partl61
jacob(42,61)=-(-emispgr/(a3*emispgr+a4*emisrftb)*
& (4*a4*emisrftb*sig*t(61)**3-(a3+a4)/((1.-emispgrf)/emispgrf)*
& (4*sig*tpgrf**3*c2)))a3
&      -facnuser*k(24)/th(24)*c2*a3
endif

c
c  jacobian 61:
if(nusselt1.eq.1.0)then
jacob(61,35)=nusselt1*k(24)/th(24)*a4
jacob(61,36)=-(-emisrftb/(a3*emispgr+a4*emisrftb)*
& (4*a3*emispgr*sig*t(36)**3-(a3+a4)/((1.-emispgrf)/emispgrf)*
& (4*sig*tpgrf**3*c1)))a4
&      -nusselt1*k(24)/th(24)*c1*a4
jacob(61,48)=emisrftb/(a3*emispgr+a4*emisrftb)*
&      (a3+a4)/((1.-emispgrf)/emispgrf)*a4
jacob(61,60)=k(28)/th(28)*a2
jacob(61,61)=-(-4*emisrftb*sig*t(61)**3-
&      emisrftb/(a3*emispgr+a4*emisrftb)*
& (4*a4*emisrftb*sig*t(61)**3-(a3+a4)/((1.-emispgrf)/emispgrf)*
& (4*sig*tpgrf**3*c2)))a4
&      -nusselt1*k(24)/th(24)*c2*a4
&      -k(28)/th(28)*a2
else
facnuser=-value1*partl35
jacob(61,35)=facnuser*k(24)/th(24)*a4
facnuser=value1*partl36
jacob(61,36)=-(-emisrftb/(a3*emispgr+a4*emisrftb)*
& (4*a3*emispgr*sig*t(36)**3-(a3+a4)/((1.-emispgrf)/emispgrf)*
& (4*sig*tpgrf**3*c1)))a4
&      -facnuser*k(24)/th(24)*c1*a4
jacob(61,48)=emisrftb/(a3*emispgr+a4*emisrftb)*
&      (a3+a4)/((1.-emispgrf)/emispgrf)*a4
jacob(61,60)=k(28)/th(28)*a2
facnuser=value1*partl61
jacob(61,61)=-(-4*emisrftb*sig*t(61)**3-
&      emisrftb/(a3*emispgr+a4*emisrftb)*
& (4*a4*emisrftb*sig*t(61)**3-(a3+a4)/((1.-emispgrf)/emispgrf)*
& (4*sig*tpgrf**3*c2)))a4
&      -facnuser*k(24)/th(24)*c2*a4
&      -k(28)/th(28)*a2
endif

c
c  jacobian 48:
jacob(48,35)=-4*sig*tend**3*cc1/(1./fc35)
&      -4*sig*tend**3*cc1/(1./fc36)
jacob(48,36)=-4*sig*(tpgrf)**3*c1/
&      ((1.-emispgrf)/emispgrf)
&      -4*sig*tend**3*cc2/(1./fc35)
&      -4*sig*tend**3*cc2/(1./fc36)

```

```

jacob(48,38)=-4*sig*tend**3*cc3/(1./fc35)
&
-4*sig*tend**3*cc3/(1./fc36)
jacob(48,39)=-4*sig*tend**3*cc4/(1./fc35)
&
-4*sig*tend**3*cc4/(1./fc36)
jacob(48,61)=-4*sig*(tpgrf)**3*c2/
&
((1.-emispgrf)/emispgrf)
&
-4*sig*tend**3*cc5/(1./fc35)
&
-4*sig*tend**3*cc5/(1./fc36)
jacob(48,47)=-1./(1/fc34)
jacob(48,48)=1./((1.-emispgrf)/emispgrf)
&
+(1/(1/fc31)+1/(1/fc32)+1/(1/fc34)+1/(1/fc35)+1/
&
(1/fc36))
jacob(48,49)=-1./(1/fc32)
jacob(48,50)=-1./(1/fc31)
c add eqns. 44 & 47 (left hand side of cabin interior (surface C))
c jacobian 44:
if(nusselt2.eq.1.0)then
jacob(44,20)=k(23)/th(23)
jacob(44,38)=-4*sig*t(38)**3/((1.-emisfi)/emisfi)
&
-nusselt2*k(11)/th(11)-k(23)/th(23)
jacob(44,39)=nusselt2*k(11)/th(11)
jacob(44,47)=1./((1.-emisfi)/emisfi)
else
jacob(44,20)=k(23)/th(23)
value2=0.18*(prandtl/(0.2+prandtl)*gravity/(.5*thdiffus*visckine)
&
*th(11)**3)**0.29
ft38t391=(t(38)+t(39))**(-0.29)*(t(38)-t(39))
&
*0.29*(abs(t(38)-t(39)))**(-0.71)+(abs(t(38)-t(39)))**0.29
&
*((t(38)-t(39))*(-0.29)*(t(38)+t(39))**(-1.29)
&
+(t(38)+t(39))**(-0.29))
facnusel=value2*ft38t391
jacob(44,38)=-4*sig*t(38)**3/((1.-emisfi)/emisfi)
&
-facnusel*k(11)/th(11)-k(23)/th(23)
ft38t392=(t(38)+t(39))**(-0.29)*(t(38)-t(39))
&
*(-0.29)*(abs(t(38)-t(39)))**(-0.71)+(abs(t(38)-t(39)))**0.29
&
*((t(38)-t(39))*(-0.29)*(t(38)+t(39))**(-1.29)
&
+(-1.)*(t(38)+t(39))**(-0.29))
facnusel=-value2*ft38t392
jacob(44,39)=facnusel*k(11)/th(11)
jacob(44,47)=1./((1.-emisfi)/emisfi)
endif
c
c jacobian 47:
jacob(47,35)=-4*sig*tend**3*cc1/(1./fc45)
&
-4*sig*tend**3*cc1/(1./fc46)
jacob(47,36)=-4*sig*tend**3*cc2/(1./fc45)
&
-4*sig*tend**3*cc2/(1./fc46)
jacob(47,38)=-4*sig*t(38)**3/((1.-emisfi)/emisfi)
&
-4*sig*tend**3*cc3/(1./fc45)
&
-4*sig*tend**3*cc3/(1./fc46)
jacob(47,39)=-4*sig*tend**3*cc4/(1./fc45)

```

```

&      -4*sig*tend**3*cc4/(1./fc46)
jacob(47,61)=-4*sig*tend**3*cc5/(1./fc45)
&      -4*sig*tend**3*cc5/(1./fc46)
jacob(47,47)=1./((1.-emisfi)/emisfi)
&      +(1/(1/fc41)+1/(1/fc42)+1/(1/fc43)+1/(1/fc45)+1/
&      (1/fc46))
jacob(47,48)=-1./(1/fc43)
jacob(47,49)=-1./(1/fc42)
jacob(47,50)=-1./(1/fc41)
c add eqns. 45 & 49 (left hand side of cabin interior (surface C))
c jacobian 45:
  if(nusselt2.eq.1.0)then
    jacob(45,21)=k(23)/th(23)
    jacob(45,38)=nusselt2*k(11)/th(11)
    jacob(45,39)=-4*sig*t(39)**3/((1.-emisfi)/emisfi)
    &      -nusselt2*k(11)/th(11)-k(23)/th(23)
    jacob(45,49)=1./((1.-emisfi)/emisfi)
  else
    jacob(45,21)=k(23)/th(23)
    facnusel=value2*ft38t391
    jacob(45,38)=facnusel*k(11)/th(11)
    facnusel=-value2*ft38t392
    jacob(45,39)=-4*sig*t(39)**3/((1.-emisfi)/emisfi)
    &      -facnusel*k(11)/th(11)-k(23)/th(23)
    jacob(45,49)=1./((1.-emisfi)/emisfi)
  endif
c
c
c jacobian 49:
  jacob(49,35)=-4*sig*tend**3*cc1/(1./fc25)
  &      -4*sig*tend**3*cc1/(1./fc26)
  jacob(49,36)=-4*sig*tend**3*cc2/(1./fc25)
  &      -4*sig*tend**3*cc2/(1./fc26)
  jacob(49,38)=-4*sig*tend**3*cc3/(1./fc25)
  &      -4*sig*tend**3*cc3/(1./fc26)
  jacob(49,39)=-4*sig*t(39)**3/((1.-emisfi)/emisfi)
  &      -4*sig*tend**3*cc4/(1./fc25)
  &      -4*sig*tend**3*cc4/(1./fc26)
  jacob(49,61)=-4*sig*tend**3*cc5/(1./fc25)
  &      -4*sig*tend**3*cc5/(1./fc26)
  jacob(49,47)=-1./(1/fc24)
  jacob(49,48)=-1./(1/fc23)
  jacob(49,49)=1./((1.-emisfi)/emisfi)
  &      +(1/(1/fc21)+1/(1/fc23)+1/(1/fc24)+1/(1/fc25)+1/
  &      (1/fc26))
  jacob(49,50)=-1./(1/fc21)
c add eqns. 41 & 50 (floor (top of spar cap) of cabin interior
c (surface C))
c jacobian 41:
  if(nusselt1.eq.1.0)then
    jacob(41,34)=k(21)/th(21)

```

```

jacob(41,35)=-4*sig*t(35)**3/((1.-emisfl)/emisfl)
&      -nusselt1*k(24)/th(24)-k(21)/th(21)
jacob(41,36)=nusselt1*k(24)/th(24)*c1
jacob(41,50)=1./((1.-emisfl)/emisfl)
jacob(41,61)=nusselt1*k(24)/th(24)*c2
else
jacob(41,34)=k(21)/th(21)
facnusel=-value1*partl35
jacob(41,35)=-4*sig*t(35)**3/((1.-emisfl)/emisfl)
&      -facnusel*k(24)/th(24)-k(21)/th(21)
facnusel=value1*partl36
jacob(41,36)=facnusel*k(24)/th(24)*c1
jacob(41,50)=1./((1.-emisfl)/emisfl)
facnusel=value1*partl61
jacob(41,61)=facnusel*k(24)/th(24)*c2
endif
c
c jacobian 50:
jacob(50,35)=-4*sig*t(35)**3/((1.-emisfl)/emisfl)
&      -4*sig*tend**3*cc1/(1./fc15)
&      -4*sig*tend**3*cc1/(1./fc16)
jacob(50,36)=-4*sig*tend**3*cc2/(1./fc15)
&      -4*sig*tend**3*cc2/(1./fc16)
jacob(50,38)=-4*sig*tend**3*cc3/(1./fc15)
&      -4*sig*tend**3*cc3/(1./fc16)
jacob(50,39)=-4*sig*tend**3*cc4/(1./fc15)
&      -4*sig*tend**3*cc4/(1./fc16)
jacob(50,61)=-4*sig*tend**3*cc5/(1./fc15)
&      -4*sig*tend**3*cc5/(1./fc16)
jacob(50,47)=-1./(1/fc14)
jacob(50,48)=-1./(1/fc13)
jacob(50,49)=-1./(1/fc12)
jacob(50,50)=1./((1.-emisfl)/emisfl)
&      +(1/(1/fc12)+1/(1/fc13)+1/(1/fc14)+1/(1/fc15)+1/
&      (1/fc16))
c
c
c wing bottom, facing tarmac. there are a total of 12 eqns. to consider
c within three sections of the wing: LHS(section: 1); RHS (section 2)
c MIDDLE (section 3, ie underneath the fuselage). these eqns. are:
c LHS(section 1): fn19(t8),fn20(j2),fn51(t51),fn54(j12);
c RHS(section 2): fn21(t16),fn22(j4),fn53(t53),fn56(j14);
c MIDDLE(section 3): fn31(t25),fn46(j7),fn52(t52),fn55(j13).
c simple two-dimensional conduction from the unsheltered tarmac
c surface to the sheltered surface is considered. a tarmac
c thickness tarmd is assumed. also the bottom of the sheltered
c tarmac is considered adiabatic (insulated) so that no q
c is assumed to sip into the ground (being conservative).
c also for the depth, tarmd, the temperature of the unsheltered
c tarmac is assumed constant. note, qtarm and ttarm are known,
c for they are unsheltered tarmac temperatures.

```

c

c

c LHS(section 1): jacobian: 19;20;51&54;

```

  jacob(19,7)=k(7)/th(7)
  jacob(19,8)=-(4*sig*t(8)**3)/((1-emisw)/emisw)
&      -hhw-k(7)/th(7)
  jacob(19,41)=1./((1.-emisw)/emisw)
  jacob(20,8)=-(4*sig*t(8)**3)/((1-emisw)/emisw)
  jacob(20,41)=1./((1.-emisw)/emisw)
&      +1/(1/fbwa(3,1))+1/(1/fbwa(3,2))+1/(1/fbwa(3,4))
&      +1/(1/fbwa(3,5))+1/(1/fbwa(3,6))
  jacob(20,54)=-1/(1/fbwa(3,1))
  jacob(20,55)=-fbwb(4,1)/(1/fbwa(3,2))
  jacob(20,56)=-fbwa(4,1)*fbwb(4,2)/(1/fbwa(3,2))
  jacob(51,51)=(-4*sig*t(51)**3)/((1.-emista)/emista)-hht)
&      *rlwng*widthwng
&      -2*conducta/(fact*widthwng)*tarmd*rlwng
&      -conducta/(fact*widthwng)*tarmd*widthwng
&      -conducta/(0.5*wfuselg+0.5*rlwng)*tarmd*widthwng
  jacob(51,52)=conducta/(0.5*wfuselg+0.5*rlwng)*tarmd*widthwng
  jacob(51,54)=1./((1.-emista)/emista)*rlwng*widthwng
  jacob(54,51)=-4*sig*t(51)**3)/((1.-emista)/emista)
  jacob(54,43)=-fbwa(4,3)*fbwb(4,2)/(1/fbwa(1,2))
  jacob(54,46)=-fbwb(4,3)/(1/fbwa(1,2))
  jacob(54,54)=1./((1.-emista)/emista)
&      +1/(1/fbwa(1,2))+1/(1/fbwa(1,3))+1/(1/fbwa(1,4))
&      +1/(1/fbwa(1,5))+1/(1/fbwa(1,6))
  jacob(54,41)=-1/(1/fbwa(1,3))

```

c

c RHS(section 2): jacobian: 21;22;53&56;

```

  jacob(21,15)=k(7)/th(7)
  jacob(21,16)=-(4*sig*t(16)**3)/((1-emisw)/emisw)
&      -hhw-k(7)/th(7)
  jacob(21,43)=1./((1.-emisw)/emisw)
  jacob(22,16)=-(4*sig*t(16)**3)/((1-emisw)/emisw)
  jacob(22,43)=1./((1.-emisw)/emisw)
&      +1/(1/fbwa(3,1))+1/(1/fbwa(3,2))+1/(1/fbwa(3,4))
&      +1/(1/fbwa(3,5))+1/(1/fbwa(3,6))
  jacob(22,54)=-fbwa(2,1)*fbwb(2,4)/(1/fbwa(3,4))
  jacob(22,55)=-fbwb(2,1)/(1/fbwa(3,4))
  jacob(22,56)=-1/(1/fbwa(3,1))
  jacob(53,52)=conducta/(0.5*wfuselg+0.5*rlwng)*tarmd*widthwng
  jacob(53,53)=(-4*sig*t(53)**3)/((1.-emista)/emista)-hht)
&      *rlwng*widthwng
&      -2*conducta/(fact*widthwng)*tarmd*rlwng
&      -conducta/(fact*widthwng)*tarmd*widthwng
&      -conducta/(0.5*wfuselg+0.5*rlwng)*tarmd*widthwng
  jacob(53,56)=1./((1.-emista)/emista)*rlwng*widthwng
  jacob(56,53)=-4*sig*t(53)**3)/((1.-emista)/emista)
  jacob(56,41)=-fbwa(2,3)*fbwb(2,4)/(1/fbwa(1,4))
  jacob(56,43)=-1/(1/fbwa(1,3))

```

```

      jacob(56,46)=-fbwb(2,3)/(1/fbwa(1,4))
      jacob(56,56)=1/((1-emista)/emista)
&      +1/(1/fbwa(1,2))+1/(1/fbwa(1,3))+1/(1/fbwa(1,4))
&      +1/(1/fbwa(1,5))+1/(1/fbwa(1,6))
c
c MIDDLE(section 3): jacobian: 31;46;52&55:
c
      jacob(31,25)=-(4*sig*t(25)**3)/((1-emisfb)/emisfb)
&      -hhf-k(12)/th(12)
      jacob(31,26)=k(12)/th(12)
      jacob(31,46)=1/((1.-emisfb)/emisfb)
      jacob(46,25)=-(4*sig*t(25)**3)/((1-emisfb)/emisfb)
      jacob(46,46)=1/((1.-emisfb)/emisfb)
&      +1/(1/fbwb(3,1))+1/(1/fbwb(3,2))+1/(1/fbwb(3,4))
&      +1/(1/fbwb(3,5))+1/(1/fbwb(3,6))
      jacob(46,54)=-fbwa(2,1)/(1/fbwb(3,4))
      jacob(46,55)=-1/(1/fbwb(3,1))
      jacob(46,56)=-fbwa(4,1)/(1/fbwb(3,2))
      jacob(52,51)=conducta/(0.5*wfuselg+0.5*rlwng)*tarmd*widthwng
      jacob(52,52)=(-4*sig*t(52)**3)/((1.-emista)/emista)-hht)
&      *wfuselg*widthwng
&      -2*conducta/(fact*widthwng)*tarmd*wfuselg
&      -conducta/(0.5*wfuselg+0.5*rlwng)*tarmd*widthwng
&      -conducta/(0.5*wfuselg+0.5*rlwng)*tarmd*widthwng
      jacob(52,53)=conducta/(0.5*wfuselg+0.5*rlwng)*tarmd*widthwng
      jacob(52,55)=1/((1.-emista)/emista)*wfuselg*widthwng
      jacob(55,52)=-4*sig*t(52)**3/((1.-emista)/emista)
      jacob(55,41)=-fbwa(2,3)/(1/fbwb(1,4))
      jacob(55,43)=-fbwa(4,3)/(1/fbwb(1,2))
      jacob(55,46)=-1/(1/fbwb(1,3))
      jacob(55,55)=1/((1-emista)/emista)
&      +1/(1/fbwb(1,2))+1/(1/fbwb(1,3))+1/(1/fbwb(1,4))
&      +1/(1/fbwb(1,5))+1/(1/fbwb(1,6))
      return
      end
c
      subroutine tarmmodl(absorbta,emista,qtarm,t)
c purpose: determine the tarmac emission and tarmac temperature
c called from: solution
c calls: none
      common/c7/nperiods,envtdata(100,20)
      common/convect/hf(100),hw(100),ht(100)
      parameter(fpi=3.141593/180.)
      parameter(sig=0.1714e-8)
      parameter(ntrial=1000,tolt=0.0001,tolf=0.0001)
      real jacob
      dimension qtarm(100),t(100)
c assume no heat loss through the ground
c (ie ground is adiabatic/insulated).
      do 20 i=1,nperiods
        theta=envtdata(i,1)

```

```

ta=envtdata(i,2)
tsky=envtdata(i,3)
qdir=envtdata(i,5)
h=ht(i)
t(i)=100.
do 10 k=1,ntrial
  f=absorbta*qdir*sin(fpi*theta)-emista*sig*(t(i)**4-tsky**4)
&  -h*(t(i)-ta)
  errf=abs(f)
  if(errf.le.tolf)then
    qtarm(i)=emista*sig*t(i)**4
    goto 20
  endif
  p=-f
  jacob=-4*emista*sig*t(i)**3-h
  delt=p/jacob
  t(i)=t(i)+delt
  errt=abs(delt)
  if(errt.le.tolt)then
    qtarm(i)=emista*sig*t(i)**4
    goto 20
  endif
10 continue
20 continue
return
end

c
  subroutine infrqsky(qqsky)
c purpose: determines the sky radiation
c called from: solution
c calls: none
  common/c7/nperiods,envtdata(100,20)
  dimension qqsky(100)
  parameter(sig=0.1714e-8)
  do 10 i=1,nperiods
    tsky=envtdata(i,3)
10  qqsky(i)=sig*tsky**4
  return
end

c
  subroutine ludcmp
  common/c1/n,ntrial,tolx,tolf
  common/c2/x(200),fvec(200),jacob(200,200)
  common/sc1/indx(200),p(200)
  real jacob
  parameter(tiny=1.0e-20)
c purpose: given jacob(1:n,1:n), with physical dimension 100 by 100,
c this routine replaces it by the LU decomposition of a rowwise
c permutation of itself. jacob is also output, arranged as in eqn(2.3.14)
c of Numerical Recipes in Fortran (William H. press et.al).
c indx(1:n) is an output vector that records the row permutation

```

```

c affected by the partial pivot; d is output as plus minus 1 depending on
c whether the number of row interchanges was even or odd, respectively.
c this routine is used in combination with lubksb to solve linear equations
c or invert a matrix. note: tiny=a small number (for ill-conditioned pivot)
c called from: newtraph, updradio
c calls: none
      dimension vv(200)
c vv stores the implicit scaling of each row
c no row interchanges yet
      d=1.
c loop over rows to get the implicit scaling information
      do 12 i=1,n
        aamax=0.
        do 11 j=1,n
          if(abs(jacob(i,j)).gt.aamax)aamax=abs(jacob(i,j))
11      continue
        if(aamax.eq.0)pause 'singular matrix in ludcmp'
c no nonzero largest element
c save the scaling
        vv(i)=1./aamax
12      continue
c this is the loop over columns of Crout's method; this is
c equation (2.3.12), except for i=j
        do 19 j=1,n
          do 14 i=1,j-1
            sum=jacob(i,j)
            do 13 k=1,i-1
              sum=sum-jacob(i,k)*jacob(k,j)
13          continue
            jacob(i,j)=sum
14          continue
c initialize for the search for largest pivot element
          aamax=0.
c this is i=j of eqn.(2.3.12) and i=j+1...n of eqn.(2.3.13)
          do 16 i=j,n
            sum=jacob(i,j)
            do 15 k=1,j-1
              sum=sum-jacob(i,k)*jacob(k,j)
15          continue
            jacob(i,j)=sum
c figure of merit for the pivot; is it better than the best so far?
            dum=vv(i)*abs(sum)
            if (dum.ge.aamax)then
              imax=i
              aamax=dum
            endif
16          continue
c do we need to interchange rows? yes, do so
            if(j.ne.imax)then
              do 17 k=1,n
                dum=jacob(imax,k)

```

```

      jacob(imax,k)=jacob(j,k)
      jacob(j,k)=dum
17    continue
c ...and change the parity of d; also interchange the scale factor
      d=-d
      vv(imax)=vv(j)
    endif
    indx(j)=imax
    if(jacob(j,j).eq.0)jacob(j,j)=tiny
c if the pivot element is zero the matrix is singular (at least
c to the precision of the algorithm). for some applications on singular
c matrices, it is desirable to substitute tiny for zero.
c now, finally divide by the pivot element
      if(j.ne.n)then
        dum=1./jacob(j,j)
        do 18 i=j+1,n
          jacob(i,j)=jacob(i,j)*dum
18      continue
        endif
c go back for the next column in the reduction
19  continue
    return
  end
c
  subroutine lubksb
    common/c1/n,ntrial,tolx,tolf
    common/c2/x(200),fvec(200),jacob(200,200)
    common/sc1/indx(200),b(200)
    real jacob
c purpose: solves the set of n linear equations a.x=b.
c here jacob is input, not as the
c matrix jacob but rather as its LU decomposition, determined by the routine
c ludcmp. indx is input as the permutation vector returned by ludcmp.
c b(1:n) is input as the right-hand side vector b, and returns with the
c solution vector. jacob and index are not modified by this routine and can
c be left in place for successive calls with different right-hand sides b.
c this routine takes into account the possibility that b will begin with many
c zero elements, so it is efficient for use in matrix inversion (if needed).
c called from: newtraph,updradio
c calls: none
      ii=0
c when ii is set to a positive value, it will become the index
c of the first nonvanishing element of b. we now do the forward substitution,
c eqn. (2.3.6). the only new wrinkle is to unscramble the permutation as we
c go.
      do 12 i=1,n
        ll=indx(i)
        sum=b(ll)
        b(ll)=b(i)
        if(ii.ne.0)then
          do 11 j=ii,i-1

```

```

      sum=sum-jacob(i,j)*b(j)
11    continue
      elseif(sum.ne.0)then
c a nonzero element was encountered, so from now on we will have
c to do the sums in the loop above.
      ii=i
      endif
      b(i)=sum
12    continue
c now we do the backsubstitution, eqn. (2.3.7)
      do 14 i=n,1,-1
        sum=b(i)
        do 13 j=i+1,n
          sum=sum-jacob(i,j)*b(j)
13      continue
c store a component of the solution vector x.
        b(i)=sum/jacob(i,i)
14    continue
      return
      end

```

subroutine qrefcab

c purpose: determines the intercepted solar radiation and the absorbed

c diffuse radiation in the cabin. cabin treated as a partial

c cavity

c called from: solution

c calls: diffsola

```

      common/c4/widthwng, lengtwng, hwng, locwng, lplane, lfuselg, hfuselg,
&          wfuselg
      common/c5/trnsm, absorbta, emista, tarmd, densta, conducta, shcta
      common/c6/absorbff, emisff, absorbbw, emisww, absorbfl, emisfl, absorbfi
&          emisfi, absorbpg, emispg, absorftp, emisrftp, absorfbt, emisrft
&          absorfb, emisfb
      common/c7/nperiods, envtdata(100,20)
      common/c9/scatffi, scatfi, scatfw, scatff
      common/c12/a1,a2,a3,a4,a5,a6(100),a7(100)
      common/reflecq/qreflin(6,100),qscabsin(6,100),qreflfwr(6,100),
&          qreflflwr(6,100),qscabfwr(6,100),qscabfwl(6,100),qabstwa1(6,100),
&          qabstwa2(6,100),qabstwb(6,100)
      common/vf5/fc11,fc12,fc13,fc14,fc15,fc16,fc21,fc22,fc23,fc24,
&          fc25,fc26,fc31,fc32,fc33,fc34,fc35,fc36,fc41,fc42,
&          fc43,fc44,fc45,fc46,fc51,fc52,fc53,fc54,fc55,fc56,
&          fc61,fc62,fc63,fc64,fc65,fc66
      dimension qscat(6,100)
      dimension f(6,6),absorb(6),ttrnsm(6,100)
      parameter(fpi=3.141593/180.)
      real lfuselg

```

c

c Variables:

c qreflin(,)=intercepted solar radiation (both direct and specular)

c in the cabin

```

c qscabsin(,)=absorbed diffuse solar radiation in the cabin
c
c initialize
  do 9 j=1,nperiods
    do 9 i=1,6
      ttrnsm(i,j)=0.0
      qreflin(i,j)=0.0
      qscat(i,j)=0.0
9
c find qrefl for surface A (right hand side of cabin interior)
c right hand side means as you seated normally and face the wind shield
  do 10 i=1,nperiods
    theta=fpi*envtdata(i,1)
    qdir=envtdata(i,5)
    fopndirs=envtdata(i,8)
    if(theta.ge.0.0.and.theta.le.atan(hfuselg/(2*wfuselg)))then
      qreflin(1,i)=qdir*trnsm*fopndirs
      qscat(1,i)=scatffi*(1.-absorbfi)*qreflin(1,i)*cos(theta)
    elseif(theta.gt.atan(hfuselg/(2*wfuselg)).and.theta.lt.
& atan(hfuselg/wfuselg))then
      qreflin(1,i)=qdir*trnsm*fopndirs*(1.+(1.-scatffi)*(1.-absorbfi)
& *(1.-scatffi)**2*(1.-absorbfi)**2)
      qscat(1,i)=scatffi*(1.-absorbfi)*qreflin(1,i)*cos(theta)
    elseif(theta.eq.atan(hfuselg/wfuselg))then
      qreflin(1,i)=qdir*trnsm*fopndirs
      qscat(1,i)=scatffi*(1.-absorbfi)*qreflin(1,i)*cos(theta)
    elseif(theta.gt.atan(hfuselg/wfuselg).and.theta.le.(fpi*90))
& then
      qreflin(1,i)=qdir*trnsm*fopndirs*(1.+(1.-scatffi)*(1.-absorbfi))
      qscat(1,i)=scatffi*(1.-absorbfi)*qreflin(1,i)*cos(theta)
    endif
10 continue
c find qrefl for surface B (floor of cabin interior)
  do 20 i=1,nperiods
    theta=fpi*envtdata(i,1)
    qdir=envtdata(i,5)
    fopndirs=envtdata(i,8)
    if(theta.ge.0.0.and.theta.le.atan(hfuselg/(2*wfuselg)))then
      qreflin(2,i)=qdir*trnsm*fopndirs*(1.-scatffi)**2*
& (1.-absorbfi)**2
      qscat(2,i)=scatffi*(1.-absorbfi)*qreflin(2,i)*sin(theta)
    elseif(theta.gt.atan(hfuselg/(2*wfuselg)).and.theta.lt.
& atan(hfuselg/wfuselg))then
      qreflin(2,i)=qdir*trnsm*fopndirs*((1.-scatffi)*(1.-absorbfi)
& +(1.-scatffi)**2*(1.-absorbfi)**2)
      qscat(2,i)=scatffi*(1.-absorbfi)*qreflin(2,i)*sin(theta)
    elseif(theta.eq.atan(hfuselg/wfuselg))then
      qreflin(2,i)=qdir*trnsm*fopndirs*(1.-scatffi)*(1.-absorbfi)
      qscat(2,i)=scatffi*(1.-absorbfi)*qreflin(2,i)*sin(theta)
    elseif(theta.gt.atan(hfuselg/wfuselg).and.theta.le.(fpi*90))
& then
      qreflin(2,i)=qdir*trnsm*fopndirs*(1.+(1.-scatffi)*(1.-absorbfi))

```

```

    qscat(2,i)=scatffi*(1.-absorbfi)*qreflin(2,i)*sin(theta)
  endif
20 continue
c find qrefl for surface C (left hand side of cabin interior)
do 30 i=1,nperiods
  theta=fpi*envtdata(i,1)
  qdir=envtdata(i,5)
  fopndirs=envtdata(i,8)
  if(theta.ge.0.0.and.theta.le.atan(hfuselg/(2*wfuselg)))then
    qreflin(3,i)=qdir*trnsm*fopndirs*(1.-scatffi)*(1.-absorbfi)
    qscat(3,i)=scatffi*(1.-absorbfi)*qreflin(3,i)*cos(theta)
  elseif(theta.gt.atan(hfuselg/(2*wfuselg)).and.theta.lt.
& atan(hfuselg/wfuselg))then
    qreflin(3,i)=qdir*trnsm*fopndirs*((1.-scatffi)*(1.-absorbfi)+
& (1.-scatffi)*(1.-absorbfi)*(1.-scatffi)*(1.-absorbfi))
    qscat(3,i)=scatffi*(1.-absorbfi)*qreflin(3,i)*cos(theta)
  elseif(theta.ge.atan(hfuselg/wfuselg).and.theta.lt.
& atan(2*hfuselg/wfuselg))then
    qreflin(3,i)=qdir*trnsm*fopndirs*(1.-scatffi)*(1.-absorbfi)
& *(1.-scatffi)*(1.-absorbfi)
    qscat(3,i)=scatffi*(1.-absorbfi)*qreflin(3,i)*cos(theta)
  elseif(theta.ge.atan(2*hfuselg/wfuselg).and.theta.le.(fpi*90))
& then
    qreflin(3,i)=0.0
    qscat(3,i)=0.0
  endif
30 continue
c find qrefl for plexiglass surface on the top of cabin interior
do 40 i=1,nperiods
  theta=fpi*envtdata(i,1)
  qdir=envtdata(i,5)
  fopndirs=envtdata(i,8)
  if(theta.ge.0.0.and.theta.le.atan(hfuselg/(2*wfuselg)))then
    qreflin(4,i)=qdir*trnsm*fopndirs*(1.-scatffi)*(1.-absorbfi)
& *(1.-scatffi)**4*(1.-absorbfi)**4
    qscat(4,i)=0.0
  elseif(theta.gt.atan(hfuselg/(2*wfuselg)).and.theta.lt.
& atan(hfuselg/wfuselg))then
    qreflin(4,i)=qdir*trnsm*fopndirs*((1.-scatffi)*(1.-absorbfi)*
& (1.-scatffi)**2*(1.-absorbfi)**2+
& (1.-scatffi)*(1.-absorbfi)*(1.-scatffi)**3*(1.-absorbfi)**3)
    qscat(4,i)=0.0
  elseif(theta.eq.atan(hfuselg/wfuselg))then
    qreflin(4,i)=qdir*trnsm*fopndirs*(1.-scatffi)*(1.-absorbfi)*
& (1.-scatffi)**2*(1.-absorbfi)**2
    qscat(4,i)=0.0
  elseif(theta.gt.atan(hfuselg/wfuselg).and.theta.lt.
& atan(2*hfuselg/wfuselg))then
    qreflin(4,i)=qdir*trnsm*fopndirs*((1.-scatffi)*(1.-absorbfi)*
& (1.-scatffi)*(1.-absorbfi)+
& (1.-scatffi)*(1.-absorbfi)*(1.-scatffi)**2*(1.-absorbfi)**2)

```

```

    qscat(4,i)=0.0
  elseif(theta.eq.atan(2*hfuselg/wfuselg))then
    qreflin(4,i)=qdir*trnsm*fopndirs*(1.-scatffi)*(1.-absorbfi)
  &      *(1.-scatffi)*(1.-absorbfi)
    qscat(4,i)=0.0
  elseif(theta.gt.atan(2*hfuselg/wfuselg).and.theta.le.(fpi*90))
  & then
    qreflin(4,i)=qdir*trnsm*fopndirs*((1.-scatffi)*(1.-absorbfi)+
  &      (1.-scatffi)*(1.-absorbfi)*(1.-scatffi)*(1.-absorbfi))
    qscat(4,i)=0.0
  endif
40  continue
c
c qscat(4,i) is modified to consider direct solar rays reflected from
c the tarmac as diffused solar rays, which subsequently hit the glass
  do 50 i=1,nperiods
    theta=fpi*envtdata(i,1)
    qdir=envtdata(i,5)
    fopndifs=a5/(a1+a2)
50  qscat(4,i)=trnsm*fopndifs*.5*(1.-absorbta)*qdir*sin(theta)
c
c now find the total scattered energy absorbed at surface A(RHS-inside)
c , surface B(floor-bottom), surface C(LHS-inside)
c and the plexiglass for each period. note: surface A=1; surface B=2;
c surface C=3; and inside plexiglass surface=4. Note: surfaces
c 5 & 6 are assumed to be ideal black bodies. they absorb all scattered
c rays coming from other surfaces and reflect none (ie qscat=0.0). note
c that no direct rays actually falls onto these 2 surfaces
c because of the way the plane is positioned (ie the fuselage is-
c perpendicular to the rays)
    f(1,1)=fc22
    f(1,2)=fc21
    f(1,3)=fc24
    f(1,4)=fc23
    f(1,5)=fc25
    f(1,6)=fc26
    f(2,1)=fc12
    f(2,2)=fc11
    f(2,3)=fc14
    f(2,4)=fc13
    f(2,5)=fc15
    f(2,6)=fc16
    f(3,1)=fc42
    f(3,2)=fc41
    f(3,3)=fc44
    f(3,4)=fc43
    f(3,5)=fc45
    f(3,6)=fc46
    f(4,1)=fc32
    f(4,2)=fc31
    f(4,3)=fc34

```

```

f(4,4)=fc33
f(4,5)=fc35
f(4,6)=fc36
f(5,1)=fc52
f(5,2)=fc51
f(5,3)=fc54
f(5,4)=fc53
f(5,5)=fc55
f(5,6)=fc56
f(6,1)=fc62
f(6,2)=fc61
f(6,3)=fc64
f(6,4)=fc63
f(6,5)=fc65
f(6,6)=fc66
absorb(1)=absorbfi
absorb(2)=absorbfl
absorb(3)=absorbfi
absorb(4)=a1/(a1+a2)*absorbpg+a2/(a1+a2)*absorbft
absorb(5)=1.0
absorb(6)=1.0
do 60 i=1,nperiods
  fopndifs=a1/(a1+a2)
c note: fopndifs=fraction open that allows diffused solar ray to get
c out of the cabin. this value is the same as the fraction of
c infrared sky radiation entering the cabin.
  60  ttrnsm(4,i)=trnsm*fopndifs
    call diffsola(f,absorb,ttrnsm,qscat,qscabsin)
    return
  end
c
c
  subroutine diffsola(f,absorb,trnsm,e,qscabs)
c purpose: determines the diffuse solar radiation that is absorbed by the
c           surfaces. multiple absorption and reflections considered.
c called from: qrefcab and qreffw
c calls: none
    common/c7/nperiods,envtdata(100,20)
    dimension f(6,6),absorb(6),trnsm(6,100),e(6,100),qscabs(6,100),
    &          sumdq(6),edum(6)
    real intcepq(6)
    parameter(qismall=1e-23,toldq=0.00001)
c
    do 150 j=1,nperiods
c initialize
      do 20 i=1,6
        edum(i)=e(i,j)
20      qscabs(i,j)=qismall
      do 110 iter=1,1000
        do 40 i=1,6
          intcepq(i)=0.0

```

```

      do 40 ii=1,6
40      intcepq(i)=intcepq(i)+f(i,ii)*edum(ii)
      do 50 i=1,6
        sumdq(i)=absorb(i)*intcepq(i)
        qscabs(i,j)=qscabs(i,j)+sumdq(i)
        edum(i)=(1.-absorb(i)-trnsn(i,j))*intcepq(i)
        if(edum(i).lt.0.0)edum(i)=0.0
50      continue
        if(sumdq(1)/qscabs(1,j).le.toldq.and.
          & sumdq(2)/qscabs(2,j).le.toldq.and.
          & sumdq(3)/qscabs(3,j).le.toldq.and.
          & sumdq(4)/qscabs(4,j).le.toldq.and.
          & sumdq(5)/qscabs(5,j).le.toldq.and.
          & sumdq(6)/qscabs(6,j).le.toldq)goto 150
110 continue
150 continue
c    do 170 j=1,nperiods
c    write(13,*)'period,e(i,j)'
c    write(13,*)'j=',j,(e(i,j),i=1,6)
c    write(13,*)'period,qscabs(i,j)'
c 170 write(13,*)'j=',j,(qscabs(i,j),i=1,6)
      return
      end
c
      subroutine qreffw(absorbf,absorbw)
c purpose: determines the intercepted solar radiation and the absorbed
c           diffuse radiation at the fuselage/wing junction.
c called from: solution
c calls: diffsola
      common/c4/widthwng,lengtwng,hwng,locwng,lplane,lfuselg,hfuselg,
      & wfuselg
      common/c5/trnsn,absorbta,emista,tarnd,densta,conducta,shcta
      common/vf4/fa57,fa68
      common/c7/nperiods,envtdata(100,20)
      common/c9/scatffi,scatfi,scatfw,scatff
      common/reflecq/qreflin(6,100),qscabsin(6,100),qreffwr(6,100),
      & qreffwl(6,100),qscabfwr(6,100),qscabfwl(6,100),qabstwa1(6,100),
      & qabstwa2(6,100),qabstwb(6,100)
      common/vf2/fa11,fa12,fa13,fa14,fa15,fa16,fa21,fa22,fa23,fa24,
      & fa25,fa26,fa31,fa32,fa33,fa34,fa35,fa36,fa41,fa42,
      & fa43,fa44,fa45,fa46,fa51,fa52,fa53,fa54,fa55,fa56,
      & fa61,fa62,fa63,fa64,fa65,fa66
      common/fillet/f11w,f11f
      dimension qscatfwr(6,100),qscatfwl(6,100)
      dimension f(6,6),absorb(6),ttrnsn(6,100),tempab(6,100)
      real lengtwng
      parameter(fpi=3.141593/180.)
c
c Variables:
c qreffwr(,)=intercepted solar radiation (both direct and specular)
c           by right wing

```

```

c qreflflw(,)=intercepted solar radiation (both direct and specular)
c      by left wing
c qscabfwr(,)=absorbed diffuse solar radiation by the right wing
c qscabflw(,)=absorbed diffuse solar radiation by the left wing
c
c initialize
  do 10 j=1,nperiods
    do 10 i=1,6
      ttrnsm(i,j)=0.0
      qscatfwr(i,j)=0.0
      qscatflw(i,j)=0.0
      qreflflwr(i,j)=0.0
      qreflflw(i,j)=0.0
      qscabfwr(i,j)=0.0
10    qscabflw(i,j)=0.0
c reference f(i,j)'s
  f(1,1)=fa11
  f(1,2)=fa12
  f(1,3)=fa13
  f(1,4)=fa14
  f(1,5)=fa15
  f(1,6)=fa16
  f(2,1)=fa21
  f(2,2)=fa22
  f(2,3)=fa23
  f(2,4)=fa24
  f(2,5)=fa25
  f(2,6)=fa26
  f(3,1)=fa31
  f(3,2)=fa32
  f(3,3)=fa33
  f(3,4)=fa34
  f(3,5)=fa35
  f(3,6)=fa36
  f(4,1)=fa41
  f(4,2)=fa42
  f(4,3)=fa43
  f(4,4)=fa44
  f(4,5)=fa45
  f(4,6)=fa46
  f(5,1)=fa51
  f(5,2)=fa52
  f(5,3)=fa53
  f(5,4)=fa54
  f(5,5)=fa55
  f(5,6)=fa56
  f(6,1)=fa61
  f(6,2)=fa62
  f(6,3)=fa63
  f(6,4)=fa64
  f(6,5)=fa65

```

```

    f(6,6)=fa66
    absorb(3)=0.0
    absorb(4)=0.0
    absorb(5)=0.0
    absorb(6)=0.0
    do 15 j=1,nperiods
      do 15 i=3,6
15      ttrnsm(i,j)=1.0
      fdiffsw=1./(1.-f11w*(1.-absorbw))
      fdiffsf=1./(1.-f11f*(1.-absorbf))
      absorb(1)=absorbw*fdiffsw
      absorb(2)=absorbf*fdiffsf
c find qreflwr, qreflwl, qscabfwr and qscabfwl:
c
c LHS of fuselage:
c
c diffused solar interaction between fuselage/wing only
c
    do 20 i=1,nperiods
      theta=fpi*envtdata(i,1)
      qdir=envtdata(i,5)
      qreflwl(1,i)=(1.+(1.-scatff)*(1.-absorbf))*qdir
      qscatfwl(1,i)=scatfw*(1.-absorbw)*qreflwl(1,i)*sin(theta)
      qreflwl(2,i)=(1.+(1.-scatfw)*(1.-absorbw))*qdir
      qscatfwl(2,i)=scatff*(1.-absorbf)*qreflwl(2,i)*cos(theta)
20    continue
    call diffsola(f,absorb,ttrnsm,qscatfwl,qscabfwl)
c store temporarily
c   write(13,*)'Partial diffuse solar fuselage/wing only-LHS'
    do 25 j=1,nperiods
c   write(13,*)(qscabfwl(i,j),i=1,6)
    do 25 i=1,6
25      tempab(i,j)=qscabfwl(i,j)
c
c now consider diffused solar reflection from the tarmac subsequently
c absorbed by the fuselage/wing
c
c reinitialize
    do 125 j=1,nperiods
      do 125 i=1,6
125      qscatfwl(i,j)=0.0
      do 135 j=1,nperiods
        theta=fpi*envtdata(j,1)
        qdir=envtdata(j,5)
        qtemp=.5*(1.-absorbta)*qdir*sin(theta)
        qscatfwl(4,j)=qtemp
        qscatfwl(5,j)=qtemp
135      qscatfwl(6,j)=qtemp
        f(1,4)=0.0
        f(1,5)=0.0
        f(1,6)=0.0

```

```

    call diffsola(f,absorb,ttrnsm,qscatfwl,qscabfwl)
c   write(13,*)'Partial diffuse solar reflected from tarmac-LHS'
    do 145 j=1,nperiods
c   write(13,*)(qscabfwl(i,j),i=1,6)
        do 145 i=1,6
145   tempab(i,j)=tempab(i,j)+qscabfwl(i,j)
c replace f(1,4),f(1,5),f(1,6):
        f(1,4)=fa14
        f(1,5)=fa15
        f(1,6)=fa16
c
c now consider diffused reflection from the fuselage (both direct ray
c as well as from the tarmac) that fall onto surface 5 & 6
c
c reinitialize
        do 126 j=1,nperiods
            do 126 i=1,6
126   qscatfwl(i,j)=0.0
            do 136 j=1,nperiods
                theta=fpi*envtdata(j,1)
                qdir=envtdata(j,5)
                qscatfwl(5,j)=fa57*((1.-absorbf)*(.5*(1.-absorbta)*qdir*
&               sin(theta))+scatff*(1.-absorbf)*qdir*cos(theta))
                qscatfwl(6,j)=fa68*((1.-absorbf)*(.5*(1.-absorbta)*qdir*
&               sin(theta))+scatff*(1.-absorbf)*qdir*cos(theta))
136   continue
            f(2,5)=0.0
            f(2,6)=0.0
            call diffsola(f,absorb,ttrnsm,qscatfwl,qscabfwl)
            write(13,*)'Partial diffuse solar reflected from the fuselage',
&               '(both direct ray as well as from the tarmac) that',
&               'fall onto surface 5 & 6 -LHS'
            do 146 j=1,nperiods
                write(13,*)(qscabfwl(i,j),i=1,6)
            do 146 i=1,6
146   qscabfwl(i,j)=tempab(i,j)+qscabfwl(i,j)
c replace f(2,5),f(2,6):
        f(2,5)=fa25
        f(2,6)=fa26
c
c RHS of fuselage:
c
c diffused solar interaction between fuselage/wing only
c
        do 30 i=1,nperiods
            theta=fpi*envtdata(i,1)
            qdir=envtdata(i,5)
            if(theta.ge.atan(hfuselg/(0.5*lengthwng)))then
                qreflwr(1,i)=qdir
                qscatfwr(1,i)=scatfw*(1.-absorbw)*qreflwr(1,i)*sin(theta)
            endif

```

```

      qreffwr(2,i)=0.0
      qscatfwr(2,i)=0.0
30  continue
      call diff sola(f,absorb,ttrnsm,qscatfwr,qscabfwr)
c store temporarily
      write(13,*)'Partial diffuse solar fuselage/wing only-RHS'
      do 325 j=1,nperiods
        write(13,*)(qscabfwr(i,j),i=1,6)
        do 325 i=1,6
          325   tempab(i,j)=qscabfwr(i,j)
c
c now consider diffused solar reflection from the tarmac subsequently
c absorbed by the fuselage/wing
c
c reinitialize
      do 225 j=1,nperiods
        do 225 i=1,6
          225   qscatfwr(i,j)=0.0
          do 235 j=1,nperiods
            theta=fpi*envtdata(j,1)
            qdir=envtdata(j,5)
            qtemp=.5*(1.-absorbta)*qdir*sin(theta)
            qscatfwr(4,j)=qtemp
            qscatfwr(5,j)=qtemp
          235   qscatfwr(6,j)=qtemp
            f(1,4)=0.0
            f(1,5)=0.0
            f(1,6)=0.0
            call diff sola(f,absorb,ttrnsm,qscatfwr,qscabfwr)
            write(13,*)'Partial diffuse solar reflected from tarmac-RHS'
            do 245 j=1,nperiods
              write(13,*)(qscabfwr(i,j),i=1,6)
              do 245 i=1,6
                245   tempab(i,j)=tempab(i,j)+qscabfwr(i,j)
c replace f(1,4),f(1,5),f(1,6)
            f(1,4)=fa14
            f(1,5)=fa15
            f(1,6)=fa16
c
c now consider diffused reflection from the fuselage (originating
c from the tarmac) that fall onto surface 5 & 6
c
c reinitialize
      do 226 j=1,nperiods
        do 226 i=1,6
          226   qscatfwr(i,j)=0.0
          do 236 j=1,nperiods
            theta=fpi*envtdata(j,1)
            qdir=envtdata(j,5)
            qscatfwr(5,j)=fa57*(1.-absorbf)*(1.-absorbta)*qdir*
            & sin(theta)

```

```

      qscatfwr(6,j)=fa68*(1.-absorbf)*(1.-absorbta)*qdir*
&      sin(theta))
236 continue
      f(2,5)=0.0
      f(2,6)=0.0
      call diffsola(f,absorb,trnsm,qscatfwr,qscabfwr)
      write(13,*)'Partial diffuse solar reflected from the fuselage',
&      '(originating from the tarmac) that',
&      'fall onto surface 5 & 6 -LHS'
      do 246 j=1,nperiods
        write(13,*)(qscabfwr(i,j),i=1,6)
        do 246 i=1,6
246   qscabfwr(i,j)=tempab(i,j)+qscabfwr(i,j)
      return
    end
  c
  subroutine qreftw(absorbw)
c purpose: determines the intercepted solar radiation and the absorbed
c           diffuse radiation between the wing bottom and the tarmac.
c called from: solution
c calls: difslwng
      common/c5/trnsm,absorbta,emista,tarnd,densta,conducta,shcta
      common/c6/absorbff,emisff,absorbww,emisww,absorbfl,emisfl,absorbfi
&      ,emisfi,absorbpg,emispg,absorftp,emisrftp,absorfbt,emisrftb
&      ,absorfbf,emisfb
      common/c7/nperiods,envtdata(100,20)
      common/vf1/fbwaa(6,6),fbwbb(6,6)
      common/reflecq/qreflin(6,100),qscabsin(6,100),qreflfwr(6,100),
&      qreflflw(6,100),qscabfwr(6,100),qscabflw(6,100),qabstwa1(6,100),
&      qabstwa2(6,100),qabstwb(6,100)
      dimension fbwa1(6,6),fbwa2(6,6),fbwb(6,6),
&      e1(6,100),e2(6,100),e3(6,100)
      dimension absorb1(6),absorb2(6),absorb3(6),trnsm1(6),
&      trnsm2(6),trnsm3(6)
      parameter(fpi=3.141593/180.)
  c
  c Variables:
  c qabstwa1(,)=intercepted solar radiation (direct)
  c           by the tarmac underneath the wing (LHS)
  c qabstwa2(,)=intercepted solar radiation (direct)
  c           by the tarmac underneath the wing (RHS)
  c qabstwb(,)=intercepted solar radiation (direct)
  c           by the tarmac underneath the wing(CENTER)
  c
  c initialize the diffused solar energy entering each surface
      do 10 j=1,nperiods
        do 10 i=1,6
          e1(i,j)=0.0
          e2(i,j)=0.0
10      e3(i,j)=0.0
  c initialize/reference absorb, trnsm & fbw

```

```

do 20 i=1,6
  absorb1(i)=0.0
  absorb2(i)=0.0
  absorb3(i)=0.0
  trnsm1(i)=1.0
  trnsm2(i)=1.0
  trnsm3(i)=1.0
  do 20 j=1,6
    fbwa1(i,j)=fbwaa(i,j)
    fbwa2(i,j)=fbwaa(i,j)
    fbwb(i,j)=fbwbb(i,j)
20 continue
c find diffused solar energy entering each surface, e(i,j)'s
do 30 i=1,nperiods
  theta=fpi*envtdata(i,1)
  qdir=envtdata(i,5)
  qqscat=0.5*(1.-absorbta)*qdir*sin(theta)
  e1(4,i)=qqscat
  e1(5,i)=qqscat
  e1(6,i)=qqscat
  e2(2,i)=qqscat
  e2(5,i)=qqscat
  e2(6,i)=qqscat
  e3(5,i)=qqscat
  e3(6,i)=qqscat
30 continue
c input absorb
  absorb1(1)=absorbta
  absorb2(1)=absorbta
  absorb3(1)=absorbta
c input trnsm
  trnsm1(1)=0.0
  trnsm1(3)=0.0
  trnsm2(1)=0.0
  trnsm2(3)=0.0
  trnsm3(1)=0.0
  trnsm3(3)=0.0
c some of the reflected diffused solar radiation does not affect
c surface 1 right away: because  $f_{14}=f_{41}*a_4/a_1$ , therefore if  $f_{41}=0$ ,  $f_{14}$ 
c should also = 0.
c  fbwa1(4,1)=0.0
c  fbwa1(5,1)=0.0
c  fbwa1(6,1)=0.0
c  fbwa2(2,1)=0.0
c  fbwa2(5,1)=0.0
c  fbwa2(6,1)=0.0
c  fbwb(5,1)=0.0
c  fbwb(6,1)=0.0
  fbwa1(1,4)=0.0
  fbwa1(1,5)=0.0
  fbwa1(1,6)=0.0

```

```

fbwa2(1,2)=0.0
fbwa2(1,5)=0.0
fbwa2(1,6)=0.0
fbwb(1,5)=0.0
fbwb(1,6)=0.0
c input absorb that changes
  absorb1(3)=absorbw
  absorb2(3)=absorbw
  absorb3(3)=absorbfb
  call difslwng(fbwa1,fbwa2,fbwb,absorb1,absorb2,absorb3,
&trnsm1,trnsm2,trnsm3,e1,e2,e3,qabstwa1,qabstwa2,qabstwb)
  return
end
c
  subroutine difslwng(fbwa1,fbwa2,fbwb,absorb1,absorb2,absorb3,
&trnsm1,trnsm2,trnsm3,e1,e2,e3,qabstwa1,qabstwa2,qabstwb)
c purpose: determines the diffuse solar radiation that is absorbed by the
c       wing bottom and tarmac surfaces. multiple absorption and
c       reflections considered. region divided into 3 portions
c called from: qreftw
c calls: none
  common/c7/nperiods,envtdata(100,20)
  dimension fbwa1(6,6),fbwa2(6,6),fbwb(6,6),
&          absorb1(6),absorb2(6),absorb3(6),
&          trnsm1(6),trnsm2(6),trnsm3(6),
&          e1(6,100),e2(6,100),e3(6,100),
&          qabstwa1(6,100),qabstwa2(6,100),qabstwb(6,100)
  dimension sumdq1(6),sumdq2(6),sumdq3(6),
&          edum1(6),edum2(6),edum3(6),
&          trnsmt1(6),trnsmt2(6),trnsmt3(6)
  real intcepq1(6),intcepq2(6),intcepq3(6)
  parameter(qismall=1e-23,toldq=0.00001)
c
  do 150 j=1,nperiods
c initialize
  do 20 i=1,6
    edum1(i)=e1(i,j)
    edum2(i)=e2(i,j)
    edum3(i)=e3(i,j)
    qabstwa1(i,j)=qismall
    qabstwa2(i,j)=qismall
    qabstwb(i,j)=qismall
20  continue
  igo1=0
  igo2=0
  igo3=0
  do 110 iter=1,1000
    do 40 i=1,6
      intcepq1(i)=0.0
      intcepq2(i)=0.0
      intcepq3(i)=0.0

```

```

    do 40 ii=1,6
      intcepq1(ii)=intcepq1(ii)+fbwa1(ii)*edum1(ii)
      intcepq2(ii)=intcepq2(ii)+fbwa2(ii)*edum2(ii)
      intcepq3(ii)=intcepq3(ii)+fbwb(ii)*edum3(ii)
40    continue
    trnsmt1(2)=trns1(2)*intcepq1(2)
    trnsmt2(4)=trns2(4)*intcepq2(4)
    trnsmt3(4)=trns3(4)*intcepq3(4)
    trnsmt3(2)=trns3(2)*intcepq3(2)
c
    do 50 i=1,6
c LHS (Section 1):
    if(igo1.eq.0)then
      sumdq1(i)=absorb1(i)*intcepq1(i)
      qabstwa1(i,j)=qabstwa1(i,j)+sumdq1(i)
      if(i.eq.2)then
        edum1(2)=trnsmt3(4)
      else
        edum1(i)=(1.-absorb1(i)-trns1(i))*intcepq1(i)
      endif
    endif
c RHS (section 2):
    if(igo2.eq.0)then
      sumdq2(i)=absorb2(i)*intcepq2(i)
      qabstwa2(i,j)=qabstwa2(i,j)+sumdq2(i)
      if(i.eq.4)then
        edum2(4)=trnsmt3(2)
      else
        edum2(i)=(1.-absorb2(i)-trns2(i))*intcepq2(i)
      endif
    endif
c CENTER (underneath the fuselage; Section 3):
    if(igo3.eq.0)then
      sumdq3(i)=absorb3(i)*intcepq3(i)
      qabstwb(i,j)=qabstwb(i,j)+sumdq3(i)
      if(i.eq.2)then
        edum3(2)=trnsmt2(4)
      elseif(i.eq.4)then
        edum3(4)=trnsmt1(2)
      else
        edum3(i)=(1.-absorb3(i)-trns3(i))*intcepq3(i)
      endif
    endif
50    continue
c LHS (Section 1):
    if(sumdq1(1)/qabstwa1(1,j).le.toldq.and.
&    sumdq1(2)/qabstwa1(2,j).le.toldq.and.
&    sumdq1(3)/qabstwa1(3,j).le.toldq.and.
&    sumdq1(4)/qabstwa1(4,j).le.toldq.and.
&    sumdq1(5)/qabstwa1(5,j).le.toldq.and.
&    sumdq1(6)/qabstwa1(6,j).le.toldq)igo1=1

```

c RHS (section 2):

```
      if(sumdq2(1)/qabstwa2(1,j).le.toldq.and.
&      sumdq2(2)/qabstwa2(2,j).le.toldq.and.
&      sumdq2(3)/qabstwa2(3,j).le.toldq.and.
&      sumdq2(4)/qabstwa2(4,j).le.toldq.and.
&      sumdq2(5)/qabstwa2(5,j).le.toldq.and.
&      sumdq2(6)/qabstwa2(6,j).le.toldq)igo2=1
```

c CENTER (underneath the fuselage; Section 3):

```
      if(sumdq3(1)/qabstwb(1,j).le.toldq.and.
&      sumdq3(2)/qabstwb(2,j).le.toldq.and.
&      sumdq3(3)/qabstwb(3,j).le.toldq.and.
&      sumdq3(4)/qabstwb(4,j).le.toldq.and.
&      sumdq3(5)/qabstwb(5,j).le.toldq.and.
&      sumdq3(6)/qabstwb(6,j).le.toldq)igo3=1
      if(igo1.eq.1.and.igo2.eq.1.and.igo3.eq.1)goto 150
```

110 continue

150 continue

c

```
      write(13,*)'LHS (Section 1): '
      do 170 j=1,nperiods
        write(13,*)'period,e1(i,j)'
        write(13,*)'j=',j,(e1(i,j),i=1,6)
        write(13,*)'period,qabstwa1(i,j)'
170    write(13,*)'j=',j,(qabstwa1(i,j),i=1,6)
        write(13,*)'RHS (section 2):'
        do 180 j=1,nperiods
          write(13,*)'period,e2(i,j)'
          write(13,*)'j=',j,(e2(i,j),i=1,6)
          write(13,*)'period,qabstwa2(i,j)'
180    write(13,*)'j=',j,(qabstwa2(i,j),i=1,6)
          write(13,*)'CENTER (underneath the fuselage; Section 3): '
          do 190 j=1,nperiods
            write(13,*)'period,e3(i,j)'
            write(13,*)'j=',j,(e3(i,j),i=1,6)
            write(13,*)'period,qabstwb(i,j)'
190    write(13,*)'j=',j,(qabstwb(i,j),i=1,6)
          return
        end
```

c

subroutine sbfillet(icodef11,percent,f11w,f11f)

c purpose: determines the view factors for a portion of a quadrant surface to

c be used for determining the effective absorbtivities

c and emissivities

c called from: solution

c calls: none

real l

parameter(fpi=3.141592654/180, r=1.0,l=1000.)

c

c Variables:

c icodef11=1: the normal scenario where the wing paint extend beyond

c the fillet-fuselage junction. the wing will

```

c      be assigned an (infrared & diffused solar) absorbtivity,
c      and emissivity modified by the fillet factor,
c      the fuselage will have f11f=0.0, implying that the flat
c      absorbtivity and emissivity (normal) will be used throughout
c      the fuselage.
c
c icodef11=2: the abnormal scenario where the fuselage paint extend
c      beyond the fillet-wing junction, and into the wing to
c      some extend. the fuselage will
c      be assigned an (infrared & diffused solar) absorbtivity,
c      and emissivity modified by the fillet factor, whereas the
c      wing will have f11w=0.0, implying that the flat absorbtivity
c      and emissivity (normal) will be used throughout the wing.
c
c icodef11=3: the wing-paint/fuselage paint junction is within the fillet
c      region. you need the percentage of the fillet occupied by the
c      wing paint (percentage range from 0 to 100%).
c
c r &l=just dummies
c
  if(icodef11.eq.1)percent=100.
  if(icodef11.eq.2)percent=0.0
  theta=.9*percent
c wing region (region a):
c initialize:
  f11=0.0
  if(theta.eq.0)goto 10
  w1=theta*fpi*r
  a1=w1*l
  w2=2.*r*sin(fpi*theta/2.)
  a2=w2*l
  f21=1.
  f12=a2/a1*f21
  f11=1.-f12
10 f11w=f11
c write(9,*)'wing region, theta, f11w',theta,f11w
c fuselage region (region b):
c initialize
  f11=0.0
  if(theta.eq.90.)goto 20
  theta1=90.-theta
  w1=theta1*fpi*r
  a1=w1*l
  w2=2.*r*sin(fpi*theta1/2.)
  a2=w2*l
  f21=1.
  f12=a2/a1*f21
  f11=1.-f12
20 f11f=f11
c write(9,*)'fuselage region, theta, f11f',theta,f11f
  return

```

end

```

subroutine trnsien1(itotitr,iperiod,theta,ta,qsky,
  &      hhfo,hhwo,hhto,qdir,qtarm,ttarm,absorbf,emisf,absorbw,
  &      emisw,eemisf,eemisw,rff,rfw)
c purpose: controls the transient analysis
c called from: solution
c calls: lhwng1, rhswng1, fuslgr1, fuslgrf1, tarmac1, updradio
  common/c1/n,ntrial,tolx,tolf
  common/c4/widthwng,lengthwng,hwng,locwng,lplane,lfuselg,hfuselg,
  &      wfuselg
  common/c7/nperiods,envtdata(100,20)
  common/c12/a1,a2,a3,a4,a5,a6(100),a7(100)
  common/c13/nintervl,vel(100),timeintv(100)
  common/c16/iffindiff,delttime,amcabtmp,intvprnt
  common/temp1/t1(1000)
  common/temp2/t2(1000)
  common/temp3/t3(1000)
  common/temp4/t4(1000),tpg1,tpg2,tpg3
  common/temp5/t5(1000)
  common/radiosit/radiosj(14)
  common/data1/ddelttime
  common/result/temp(200),temptrn(200)
  dimension tvel(100),tottime(100)
  real lfuselg,lplane
  parameter(fhhc=1.0)
  parameter(sunf=0.5,sunf2=0.5)
  parameter(temptoln=0.000001)
c
c Variables:
c t1()...t5()=temperatures
c radiosj()=updated radiosity
c deltime=time increment in transient analysis (sec)
c temp=steady state temperature
c temptrn=transient state temperature
c amcabtmp=ambient cab temperature (Rankine)
c fhhc=factor for convection in cabin
c temptoln=difference between the current ambient cabin temp and the
c      previous one: to prevent infinite loop
c intvprnt=intervals for printing purposes
c sunf=assume there is a fifty percent chance that a spot will be under
c      direct sun throughout the cooling phase.
c sunf2=only fifty percent of the sun gets into the cabin. or take it
c      the average between nothing getting through to all of the direct rays
c      getting through.
c
  if(itotitr.eq.1)then
    ddelttime=delttime/3600
c convert the velocity from mph to ft/sec
  do 50 i=1,nintervl+1
50   tvel(i)=vel(i)*5280./3600

```

```

c find the total time from taxi till the application of limit load & beyond
    tottime(1)=0.0
    do 60 i=1,nintervl
60    tottime(i+1)=tottime(i)+timeintv(i)
    endif
c    if(iperiod.ne.3)return
c
c formulate transient cooling:
c
c initial cooling when the door is opened
c
    icool=1
c sunfact is assumed to be always=1, during initial cooling
    sunfact=1.0
    if(itotitr.le.10*nperiods)then
    do 15 i=1,2
    write(13+i,*)'Total # of analysis=',itotitr
    write(13+i,*)'TRANSIENT ANALYSIS-Temperatures'
15  write(13+i,*)'cooling when door is opened, iperiod=',iperiod
c    write(16,*)'Total # of analysis=',itotitr
c    write(16,*)'TRANSIENT ANALYSIS-Radiosity'
c    write(16,*)'cooling when door is opened, iperiod=',iperiod
    write(10,*)"
        write(10,*)"
    write(10,*)'TRANSIENT TEMPERATURES FOR PERIOD:',iperiod
    write(10,*)"
    write(10,*)'COOLING WHILE DOOR IS OPENED:'
    write(10,*)"
    write(10,90)(i,i=1,53)
    write(14,90)(i,i=1,20)
    write(15,90)(i,i=21,53)
c    write(16,90)(i,i=54,67)
90  format(1x,'time(sec)',1x,500(1x,i5))
    endif
    do 75 i=1,n
75  temptrn(i)=temp(i)
    iprint=1
    nprn=0
    istep=0
c find convective coefficient in the cabin; assume it to be a function of:
    hhc=fhhc*(hhfo+hhwo+hhto)/3.
    if(itotitr.eq.1)then
c find weights for tend and tambcab
    aa1=lfuselg*wfuselg
    aa2=a1
    aa3=lfuselg*hfuselg
    aa4=aa3
    aa5=a2
    atot=aa1+aa2+aa3+aa4+aa5
    cc1=aa1/atot
    cc2=aa2/atot
  
```

```

cc3=aa3/atot
cc4=aa4/atot
cc5=aa5/atot
aa6=hfuselg*wfuselg
aa7=aa6
atot1=aa1+aa2+aa3+aa4+aa5+aa6+aa7
dd1=aa1/atot1
dd2=aa2/atot1
dd3=aa3/atot1
dd4=aa4/atot1
dd5=aa5/atot1
dd6=aa6/atot1
dd7=aa7/atot1
endif
tempert2=1000.0
10  istep=istep+1
    call lhwng1(itotitr,iperiod,istep,theta,ta,hhwo,rfw,absorbw,
    &          emisw,eemisw,sunfact)
    call rhwng1(itotitr,iperiod,istep,theta,ta,hhwo,rfw,absorbw,
    &          emisw,eemisw,sunfact)
    call fuslgr1(itotitr,icool,iperiod,istep,theta,ta,hhfo,hhc,rff,
    &          absorbf,eemisf,sunfact,sunfact2)
    call fuslgrf1(itotitr,icool,iperiod,istep,theta,ta,hhfo,hhc,qdir,
    &          qsky,qtarm,sunfact2)
    call tarmac1(itotitr,icool,iperiod,istep,ta,ttarm,hhto)
    call updradio(itotitr,iperiod,istep,qsky,qtarm,emisf,
    &          emisw,eemisf,eemisw,rff,rfw)
    if(iprint.eq.istep)then
      nprn=1+nprn
      iprint=nprn*intvprnt+1
      if(itotitr.le.10*nperiods)then
        write(10,80)(istep-1)*delttime,(temptrn(i),i=1,53)
        write(14,80)(istep-1)*delttime,(temptrn(i),i=1,20)
        write(15,80)(istep-1)*delttime,(temptrn(i),i=21,53)
c      write(16,80)(istep-1)*delttime,(temptrn(i),i=54,67)
80    format(f10.4,2x,500(1x,f5.1))
      endif
    endif
    tend=cc1*t4(16)+cc2*tpg3+cc3*t3(12)+cc4*t3(16)+cc5*t4(12)
    tempert1=tempert2
    tambcab=dd1*t4(16)+dd2*tpg3+dd3*t3(12)+dd4*t3(16)+dd5*t4(12)
    &      +dd6*tend+dd7*tend
    tempert2=tamcab
    write(14,*)'istep,t1,t2, absdiff',istep,tempert1,tempert2,
    &      abs(tempert1-tempert2)
    if(tamcab.le.amcabttmp)goto 100
    goto 10
c
c cooling during taxi, stop, take-off, climb & cruise
c
100 continue

```

```

    if(itotitr.le.10*nperiods)then
      do 16 i=1,3
16  write(13+i,*)'cooling during taxi, stop, take-off, climb & cruise,
    &iperiod=',iperiod
        write(10,*)"
        write(10,*)'COOLING DURING AIRCRAFT MANEUVERS:'
        write(10,*)"
        write(10,90)(i,i=1,53)
        write(14,90)(i,i=1,20)
        write(15,90)(i,i=21,53)
c    write(16,90)(i,i=54,67)
      endif
      icool=2
      sunfact=sunf
      sunfact2=sunf2
      vnatural=envtdata(iperiod,4)
      v1=tvel(1)
      time1=0.0
20  istep=istep+1
c find the convective coefficients for the fuselage and the wing, hhf & hhw
      time1=time1+delttime
      if(time1.gt.tottime(nintervl+1))goto 120
        do 130 j=1,nintervl
          if(time1.gt.tottime(j).and.time1.le.tottime(j+1))then
            jj=j
            goto 140
          endif
130  continue
140  v2=tvel(jj)+(tvel(jj+1)-tvel(jj))/timeintv(jj)*(time1-tottime(jj))
      &                                     + vnatural
      v=(v1+v2)/2.
      v1=v2
      call hcoolcon(itotitr,lplane,v,hhf)
      call hcoolcon(itotitr,widthwng,v,hhw)
c
      call lhwng1(itotitr,iperiod,istep,theta,ta,hhw,rw,absorbw,
      &            emisw,eemisw,sunfact)
      call rhwng1(itotitr,iperiod,istep,theta,ta,hhw,rw,absorbw,
      &            emisw,eemisw,sunfact)
      call fuslgr1(itotitr,icool,iperiod,istep,theta,ta,hhf,hhc,rff,
      &            absorbf,eemisf,sunfact,sunfact2)
      call fuslgrf1(itotitr,icool,iperiod,istep,theta,ta,hhf,hhc,qdir,
      &            qsky,qtarm,sunfact2)
      call tarmac1(itotitr,icool,iperiod,istep,ta,ttarm,hhto)
      call updradio(itotitr,iperiod,istep,qsky,qtarm,emisf,
      &            emisw,eemisf,eemisw,rff,rw)
      if(iprint.eq.istep)then
        nprn=1+nprn
        iprint=nprn*intvpnt+1
        if(itotitr.le.10*nperiods)then
          write(10,80)(istep-1)*delttime,(temptrn(i),i=1,53)

```

```

      write(14,80)(istep-1)*delttime,(temptrn(i),i=1,20)
      write(15,80)(istep-1)*delttime,(temptrn(i),i=21,53)
c      write(16,80)(istep-1)*delttime,(temptrn(i),i=54,67)
      endif
      endif
      goto 20
c replace n: note n was changed to 14 in the updradio subroutine
120 n=67
      return
      end
c
c
      subroutine trnsien2(itotitr,iperiod,theta,ta,qsky,
&      hhfo,hhwo,hhto,qdir,qtarm,ttarm,absorbf,emisf,absorbw,
&      emisw,eemisf,eemism,rff,rfw)
c purpose: controls the transient analysis
c called from: solution
c calls: lhswn2, rhswn2, fuslgr2, fuslgrf2, tarmac2, updradio
      common/c1/n,ntrial,tolx,tolf
      common/c4/widthwng,lengthwng,hwng,locwng,lplane,lfuselg,hfuselg,
&      wfuselg
      common/c7/nperiods,envtdata(100,20)
      common/c12/a1,a2,a3,a4,a5,a6(100),a7(100)
      common/c13/nintervl,vel(100),timeintv(100)
      common/c16/iffindiff,delttime,amcabtmp,intvprnt
      common/temp1/t1(1000)
      common/temp2/t2(1000)
      common/temp3/t3(1000)
      common/temp4/t4(1000),tpg1,tpg2,tpg3
      common/temp5/t5(1000)
      common/radiosit/radiosj(14)
      common/data1/ddelttime
      common/result/temp(200),temptrn(200)
      dimension tvel(100),tottime(100)
      real lfuselg,lplane
      parameter(fhhc=1.0)
      parameter(sunf=0.5,sunf2=0.5)
      parameter(temptoln=0.01)
c      parameter(temptoln=0.000001)
c
c Variables:
c t1()...t5()=temperatures
c radiosj()=updated radiosity
c deltime=time increment in transient analysis (sec)
c temp=steady state temperature
c temptrn=transient state temperature
c amcabtmp=ambient cab temperature (Rankine)
c fhhc=factor for convection in cabin
c temptoln=difference between the current ambient cabin temp and the
c      previous one: to prevent infinte loop
c intvprnt=intervals for printing purposes

```

```

c sunf=assume there is a fifty percent chance that a spot will be under
c   direct sun throughout the cooling phase.
c sunf2=only fifty percent of the sun gets into the cabin. or take it
c   the average between nothing getting through to all of the direct rays
c   getting through.
c
  if(itotitr.eq.1)then
    ddelttime=delttime/3600
c convert the velocity from mph to ft/sec
    do 50 i=1,nintervl+1
      50   tvel(i)=vel(i)*5280./3600
c find the total time from taxi till the application of limit load & beyond
    tottime(1)=0.0
    do 60 i=1,nintervl
      60   tottime(i+1)=tottime(i)+timeintv(i)
    endif
c   if(iperiod.ne.3)return
c
c formulate transient cooling:
c
c initial cooling when the door is opened
c
  icool=1
c sunfact is assumed to be always=1, during initial cooling
  sunfact=1.0
  if(itotitr.le.10*nperiods)then
    do 15 i=1,2
      write(13+i,*)'Total # of analysis=',itotitr
      write(13+i,*)'TRANSIENT ANALYSIS-Temperatures'
      15   write(13+i,*)'cooling when door is opened, iperiod=',iperiod
      write(16,*)'Total # of analysis=',itotitr
      write(16,*)'TRANSIENT ANALYSIS-Radiosity'
      write(16,*)'cooling when door is opened, iperiod=',iperiod
      write(10,*)"
      write(10,*)"
      write(10,*)'TRANSIENT TEMPERATURES FOR PERIOD:',iperiod
      write(10,*)"
      write(10,*)'COOLING WHILE DOOR IS OPENED:'
      write(10,*)"
      write(10,90)(i,i=1,53)
      write(14,90)(i,i=1,20)
      write(15,90)(i,i=21,53)
      write(16,90)(i,i=54,67)
      90   format(1x,'time(sec)',1x,500(1x,i5))
    endif
    do 75 i=1,n
      75   temptrn(i)=temp(i)
      iprint=1
      nprn=0
      istep=0
c find convective coefficient in the cabin; assume it to be a function of:

```

```

    hhc=fhhc*(hhfo+hhwo+hhto)/3.
    if(itotitr.eq.1)then
c find weights for tend and tambcab
    aa1=lfuselg*wfuselg
    aa2=a1
    aa3=lfuselg*hfuselg
    aa4=aa3
    aa5=a2
    atot=aa1+aa2+aa3+aa4+aa5
    cc1=aa1/atot
    cc2=aa2/atot
    cc3=aa3/atot
    cc4=aa4/atot
    cc5=aa5/atot
    aa6=hfuselg*wfuselg
    aa7=aa6
    atot1=aa1+aa2+aa3+aa4+aa5+aa6+aa7
    dd1=aa1/atot1
    dd2=aa2/atot1
    dd3=aa3/atot1
    dd4=aa4/atot1
    dd5=aa5/atot1
    dd6=aa6/atot1
    dd7=aa7/atot1
    endif
    tempert2=1000.0
10    istep=istep+1
    call newraph2(itotitr,iperiod,istep,theta,ta,hhwo,rff,absorbw,
    &               emisw,eemisw,sunfact,icool,hhfo,hhc,rff,absorbf,
    &               eemisf,sunfact2,qdir,qsky,qtarm,ttarm,hhto)
    call updradio(itotitr,iperiod,istep,qsky,qtarm,emisf,
    &               emisw,eemisf,eemisw,rff,rff)
    if(iprint.eq.istep)then
        nprn=1+nprn
        iprint=nprn*intvprnt+1
        if(itotitr.le.10*nperiods)then
            write(10,80)(istep-1)*delttime,(temptrn(i),i=1,53)
            write(14,80)(istep-1)*delttime,(temptrn(i),i=1,20)
            write(15,80)(istep-1)*delttime,(temptrn(i),i=21,53)
            write(16,80)(istep-1)*delttime,(temptrn(i),i=54,67)
80        format(f10.4,2x,500(1x,f5.1))
        endif
    endif
    tend=cc1*t4(16)+cc2*tpg3+cc3*t3(12)+cc4*t3(16)+cc5*t4(12)
    tempert1=tempert2
    tambcab=dd1*t4(16)+dd2*tpg3+dd3*t3(12)+dd4*t3(16)+dd5*t4(12)
    &      +dd6*tend+dd7*tend
    tempert2=tambcab
    write(14,*)'tempert1,tempert2, absdiff',tempert1,tempert2,
    &      abs(tempert1-tempert2)
    if(tambcab.le.amcabtmp)goto 100
  
```

```

      goto 10

c
c cooling during taxi, stop, take-off, climb & cruise
c
100 continue
    if(itotitr.le.10*nperiods)then
      do 16 i=1,3
16  write(13+i,*)'cooling during taxi, stop, take-off, climb & cruise,
    &iperiod=',iperiod
      write(10,*)"
      write(10,*)'COOLING DURING AIRCRAFT MANEUVERS:'
      write(10,*)"
      write(10,90)(i,i=1,53)
      write(14,90)(i,i=1,20)
      write(15,90)(i,i=21,53)
      write(16,90)(i,i=54,67)
    endif
    icool=2
    sunfact=sunf
    sunfact2=sunf2
    vnatural=envtdata(iperiod,4)
    v1=tvel(1)
    time1=0.0
20  istep=istep+1
c find the convective coefficients for the fuselage and the wing, hhf & hhw
    time1=time1+delttime
    if(time1.gt.tottime(nintervl+1))goto 120
      do 130 j=1,nintervl
        if(time1.gt.tottime(j).and.time1.le.tottime(j+1))then
          jj=j
          goto 140
        endif
130 continue
140 v2=tvel(jj)+(tvel(jj+1)-tvel(jj))/timeintv(jj)*(time1-tottime(jj))
    &                                     + vnatural
      v=(v1+v2)/2.
      v1=v2
      call hcoolcon(itotitr,lplane,v,hhf)
      call hcoolcon(itotitr,widthwng,v,hhw)
c
    call newraph2(itotitr,iperiod,istep,theta,ta,hhw,r fw,absorbw,
    &               emisw,eemisw,sunfact,icool,hhf,hhc,rff,absorbf,
    &               eemisf,sunfact2,qdir,qsky,qtarm,ttarm,hhto)
    call updradio(itotitr,iperiod,istep,qsky,qtarm,emisf,
    &               emisw,eemisf,eemisw,rff,r fw)
    if(iprint.eq.istep)then
      nprn=1+nprn
      iprint=nprn*intvpnt+1
      if(itotitr.le.10*nperiods)then
        write(10,80)(istep-1)*delttime,(temptrn(i),i=1,53)

```

```

      write(14,80)(istep-1)*delttime,(temptrn(i),i=1,20)
      write(15,80)(istep-1)*delttime,(temptrn(i),i=21,53)
      write(16,80)(istep-1)*delttime,(temptrn(i),i=54,67)
    endif
  endif
  goto 20
c restore n to 67: note n was changed to 120 in the newraph2 subroutine and
c to 14 in the updradio subroutine
  120 n=67
    return
  end
c
c
  subroutine newraph2(itotitr,iperiod,istep,theta,ta,hhwo,rfw,
&      absorbw,emisw,eemisw,sunfact,icool,hhfo,hhc,rff,
&      absorbf,eemisf,sunfact2,qdir,tsky,qtarm,ttarm,hhto)
c
c purpose: this subroutine calls: linear solvers; ludcmp and lubksb.
c given an initial guess x for a root in n dimensions, take
c ntrial Newton-Raphson steps to improve the root. stop if the
c root converges in either summed absolute variable increments tolz or summed
c absolute function values tolf.
c calls: lhswn2,rhswn2,fuslgr2,fuslgrf2,ludcmp,tarmac2,lubksb
c called from: trnsien2
  common/c1/n,ntrial,tolz,tozf
  common/c2/x(200),fvec(200),jacob(200,200)
  common/sc1/indx(200),p(200)
  real jacob
c
c change n to 120 to reflect the 120 equations related to the 120 temperatures to be
c found for each time step, istep for the implicit finite difference method.
c
  n=120
c
c Variables:
c isolfoun=indicator of whether solution is found or not: =1 means found;
c =0 means not found yet.
c ipower=exponent of 10. to ensure solution converges in positive numbers
c
  isolfoun=0
  ipower=1
  50 do 14 k=1,ntrial
c user subroutines supply function values at x in fvec and
c Jacobian matrix in jacob.
c
c initialize jacobian jacob(120,120)
c note: most jacobian values are 0's. also note that the local jacobians
c tjacob in subroutines below are also initialized to 0's. this initialization
c is redundant but simplifies programming.
c
  do 70 ijacob=1,n

```

```

do 70 jjacob=1,n
70 jacob(ijacob,jjacob)=0.0
c
  call lhwng2(k,itotitr,iperiod,istep,theta,ta,hhwo,rfw,absorbw,
&      emisw,eemisw,sunfact,isoloun)
  call rhwng2(k,itotitr,iperiod,istep,theta,ta,hhwo,rfw,absorbw,
&      emisw,eemisw,sunfact,isoloun)
  call fusglr2(k,itotitr,icool,iperiod,istep,theta,ta,hhfo,hhc,rff,
&      absorbf,eemisf,sunfact,sunfact2,isoloun)
  call fusgrf2(k,itotitr,icool,iperiod,istep,theta,ta,hhfo,hhc,qdir
&      ,qsky,qtarm,sunfact2,isoloun)
  call tarmac2(k,itotitr,icool,iperiod,istep,ta,ttarm,hhto,isoloun)
c return to trnsien2 for the first step (istep=1) where only initialization has
c taken place in the above called subroutines. note that the local temperature
c variables have assumed the steady state temperature tt(i) (ie x(i)) in the
c the above subroutines. after step 1, the steady state temperature values are
c all destroyed, and instead filled up with the new temperature values, newt(i).
c to retrieve the steady state temperatures, find them in /result/temp(i),
c where temp(i) is in deg Fahrenheit, and not in absolute Rankine.
  write(14,*)'istep,trial number=',istep,k
c    do 170 ijk=1,n
c      write(14,*)'row=',ijk
c 170 write(14,*)(jacob(ijk,jjj),jjj=1,n)
  if(istep.eq.1)return
c check function convergence
  errf=0.
  do 11 i=1,n
    errf=errf+abs(fvec(i))
  11 continue
c  if(errf.le.tolf) goto 20
c right-hand side of linear equations
  do 12 i=1,n
    p(i)=-fvec(i)
  12 continue
c solve linear eqns. using LU decomposition
  call ludcmp
c forward and back substitution
  call lubksb
c check root convergence; and update solution
  errx=0.
  do 13 i=1,n
    errx=errx+abs(p(i))
    x(i)=x(i)+p(i)
  13 continue
  write(14,*)'k(trial#),errx,errf',k,errx,errf
  if(errf.le.tolf.and.errx.le.tolx) goto 20
  14 continue
20 loopindc=0
  do 30 i=1,n
    if(x(i).lt.0.0)then
      write(6,*)'i=',i,x(i)

```

```

      x(i)=10**ipower
      loopindc=1
    endif
30  continue
    if(loopindc.eq.1)then
      ipower=ipower+1
      goto 50
    endif
c call the subroutines one more time, indicating a solution has been found
c for the time step istep. this call also updates the previous temperatures
c with the solved new (current) temperatures in these called subroutines.
    isolfoun=1
    call lhwng2(k,itotitr,iperiod,istep,theta,ta,hhwo,rfw,absorbw,
&      emisw,eemisw,sunfact,isolfoun)
    call rhwng2(k,itotitr,iperiod,istep,theta,ta,hhwo,rfw,absorbw,
&      emisw,eemisw,sunfact,isolfoun)
    call fuslgr2(k,itotitr,icool,iperiod,istep,theta,ta,hhfo,hhc,rff,
&      absorbf,eemisf,sunfact,sunfact2,isolfoun)
    call fuslgrf2(k,itotitr,icool,iperiod,istep,theta,ta,hhfo,hhc,qdir
&      ,qsky,qtarm,sunfact2,isolfoun)
    call tarmac2(k,itotitr,icool,iperiod,istep,ta,ttarm,hhto,isolfoun)
    return
  end
c
c
c
  subroutine lhwng1(itotitr,iperiod,istep,theta,ta,hhw,rfw,
&      absorbw,emisw,eemisw,sunfact)
c purpose: transient analysis of the left wing (the wing that is not under
c      shade)
c called from: trnsien1
c calls: none
    common/c2/tt(200),fn(200),jacob(200,200)
    common/c3/tth(100),kk(100)
    common/c6/absorbff,emisff,absorbww,emisww,absorbfl,emisfl,absorbfi
&      emisfi,absorbpg,emispq,absorftp,emisrftp,absorfbt,emisrftb
&      ,absorbfb,emisfb
    common/c10/aircondc,thdiffus,visckine,prandtl
    common/c14/thicknes(100),density(100),conductv(100),sheatcap(100)
    common/reflecq/qreflin(6,100),qscabsin(6,100),qreflfr(6,100),
&      qreflfrl(6,100),qscabfwr(6,100),qscabfwr(6,100),qabstwa1(6,100),
&      qabstwa2(6,100),qabstwb(6,100)
    common/temp1/t(1000)
    common/data1/delttime
    common/radiosit/jj(14)
    common/result/temp(200),temptrn(200)
    dimension newt(100),k(100),th(100),dens(100),shc(100)
    real kk,k,jj,newt
    parameter(sig=.1714e-8)
    parameter(fpi=3.141593/180.)
    parameter(nlayer=9)

```

```

parameter(lend=3*nlayer)
parameter(layair=5)
parameter(f=0.2)
c
c Variables:
c layair=the layer (or element) that is air space in the wing
c f=fraction of an element thickness
c
  if(itotitr.eq.1.and.istep.eq.1)then
c reference thickness (th):
  th(1)=tth(1)
  th(2)=tth(2)
  th(3)=tth(3)
  th(4)=tth(4)
  th(5)=tth(29)
  th(6)=tth(30)
  th(7)=tth(5)
  th(8)=tth(6)
  th(9)=tth(7)
c reference conductivity (k):
  k(1)=kk(1)
  k(2)=kk(2)
  k(3)=kk(3)
  k(4)=kk(4)
  k(5)=kk(29)
  k(6)=kk(30)
  k(7)=kk(5)
  k(8)=kk(6)
  k(9)=kk(7)
c reference density (dens):
  dens(1)=density(1)
  dens(2)=density(2)
  dens(3)=density(3)
  dens(4)=density(4)
  dens(6)=density(6)
  dens(7)=density(7)
  dens(8)=density(8)
  dens(9)=density(9)
c reference specific heat capacity (shc):
  shc(1)=sheatcap(1)
  shc(2)=sheatcap(2)
  shc(3)=sheatcap(3)
  shc(4)=sheatcap(4)
  shc(6)=sheatcap(6)
  shc(7)=sheatcap(7)
  shc(8)=sheatcap(8)
  shc(9)=sheatcap(9)
  endif
  if(istep.eq.1)then
c reference the initial temp. (t):
  t(1)=tt(1)

```

```

t(2)=(tt(1)+tt(2))/2.
t(3)=tt(2)
t(4)=tt(2)
t(5)=(tt(2)+tt(3))/2.
t(6)=tt(3)
t(7)=tt(3)
t(8)=(tt(3)+tt(4))/2.
t(9)=tt(4)
t(10)=tt(4)
t(11)=(tt(4)+tt(62))/2.
t(12)=tt(62)
t(16)=tt(63)
t(17)=(tt(63)+tt(5))/2.
t(18)=tt(5)
t(19)=tt(5)
t(20)=(tt(5)+tt(6))/2.
t(21)=tt(6)
t(22)=tt(6)
t(23)=(tt(6)+tt(7))/2.
t(24)=tt(7)
t(25)=tt(7)
t(26)=(tt(7)+tt(8))/2.
t(27)=tt(8)
c reference the initial radiosity (jj)
  jj(1)=tt(40)
  jj(2)=tt(41)
  return
endif
c
c formulate transient cooling:
c
c upper wing surface temp (t1), as well as internal temp t2 and t3:
  qnetsky=(sig*t(1)**4-rfw*jj(1))/((1.-rfw*eemisw)/eemisw)
  newt(1)=t(1)+detime/(dens(1)*shc(1)*f/2*th(1))*
&    ((absorbw*qreflflw(1,iperiod)*sin(fpi*theta)+
&    qscabflw(1,iperiod))*sunfact+
&    (absorbw*qreflflwr(1,iperiod)*sin(fpi*theta)+
&    qscabflwr(1,iperiod))*(1-sunfact)
&    -qnetsky-hhw*(t(1)-ta)-
&    (t(1)-t(2))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
  newt(2)=t(2)+detime/(dens(1)*shc(1)*(1-f)*th(1))*
&    ((t(1)-t(2))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-
&    (t(2)-t(3))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
  newt(3)=t(3)+detime/(dens(1)*shc(1)*f/2*th(1))*
&    ((t(2)-t(3))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-
&    (t(3)-t(4))/(f/4*th(1)/k(1)+f/4*th(2)/k(2)))
c internal nodes (all conduction): t4-t24:
  do 20 i=1,nlayer
    if(i.eq.layair)goto 20
    if(i.eq.1.or.i.eq.nlayer)goto 20
    ii=(i-1)*3

```

c beginning element within layer i: t4,t7,t10,t13,t16,t19,t22:

```

if(i.eq.layair+1)then
  newt(ii+1)=t(ii+1)+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((t(ii-3)-t(ii+1))/(f/4*th(i-2)/k(i-2)+th(i-1)/k(i-1)+
& f/4*th(i)/k(i))-
& (t(ii+1)-t(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
else
  newt(ii+1)=t(ii+1)+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((t(ii)-t(ii+1))/(f/4*th(i-1)/k(i-1)+f/4*th(i)/k(i))-
& (t(ii+1)-t(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
endif

```

c middle element within layer i: t5,t8,t11,t14,t17,t20,t23:

```

newt(ii+2)=t(ii+2)+delttime/(dens(i)*shc(i)*(1-f)*th(i))*
& ((t(ii+1)-t(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (t(ii+2)-t(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))

```

c ending element within layer i: t6,t9,t12,t15,t18,t21,t24:

```

if(i.eq.layair-1)then
  newt(ii+3)=t(ii+3)+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((t(ii+2)-t(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (t(ii+3)-t(ii+7))/(f/4*th(i)/k(i)+th(i+1)/k(i+1)+
& f/4*th(i+2)/k(i+2)))
else
  newt(ii+3)=t(ii+3)+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((t(ii+2)-t(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (t(ii+3)-t(ii+4))/(f/4*th(i)/k(i)+f/4*th(i+1)/k(i+1)))
endif

```

20 continue

c lower wing surface temp (t27), as well as internal temp t25 and t26:

```

newt(lend-2)=t(lend-2)+
& deltime/(dens(nlayer)*shc(nlayer)*f/2*th(nlayer))*
& ((t(lend-3)-t(lend-2))/
& (f/4*th(nlayer-1)/k(nlayer-1)+f/4*th(nlayer)/k(nlayer))-
& (t(lend-2)-t(lend-1))/
& (f/4*th(nlayer)/k(nlayer)+(1-f)/2*th(nlayer)/k(nlayer)))
newt(lend-1)=t(lend-1)+
& deltime/(dens(nlayer)*shc(nlayer)*(1-f)*th(nlayer))*
& ((t(lend-2)-t(lend-1))/
& (f/4*th(nlayer)/k(nlayer)+(1-f)/2*th(nlayer)/k(nlayer))-
& (t(lend-1)-t(lend))/
& (f/4*th(nlayer)/k(nlayer)+(1-f)/2*th(nlayer)/k(nlayer)))
qnetsky=(sig*t(lend)**4-jj(2))/((1.-emisw)/emisw)
newt(lend)=t(lend)+
& deltime/(dens(nlayer)*shc(nlayer)*f/2*th(nlayer))*
& (qabstwa1(3,iperiod)-qnetsky-hhw*(t(lend)-ta)+
& (t(lend-1)-t(lend))/
& (f/4*th(nlayer)/k(nlayer)+(1-f)/2*th(nlayer)/k(nlayer)))

```

do 60 i=1,lend

60 t(i)=newt(i)

temptrn(1)=t(1)-459.67

temptrn(2)=(t(3)+t(4))/2.-459.67

temptrn(3)=(t(6)+t(7))/2.-459.67

```

temptrn(4)=(t(9)+t(10))/2.-459.67
temptrn(5)=t(12)-459.67
temptrn(6)=t(16)-459.67
temptrn(7)=(t(18)+t(19))/2.-459.67
temptrn(8)=(t(21)+t(22))/2.-459.67
temptrn(9)=(t(24)+t(25))/2.-459.67
temptrn(10)=t(27)-459.67
return
end
c
c
  subroutine rhswng1(itotitr,iperiod,istep,theta,ta,hhw,rfw,
    &               absorbw,emisw,eemisw,sunfact)
c purpose: transient analysis of the right wing (the wing that is under
c          shade)
c called from: trnsien1
c calls: none
  common/c2/tt(200),fn(200),jacob(200,200)
  common/c3/tth(100),kk(100)
  common/c6/absorbff,emisff,absorbww,emisww,absorbfl,emisfl,absorbfi
    &.emisfi,absorbpg,emispg,absorftp,emisrftp,absorfbt,emisrftb
    &.absorfbf,emisfb
  common/c10/aircondc,thdiffus,visckine,prandtl
  common/c14/thicknes(100),density(100),conductv(100),sheatcap(100)
  common/reflecq/qreflin(6,100),qscabsin(6,100),qreflwr(6,100),
    &qreflwl(6,100),qscabfwr(6,100),qscabfwl(6,100),qabstwa1(6,100),
    &qabstwa2(6,100),qabstwb(6,100)
  common/temp2/t(1000)
  common/data1/delttime
  common/radiosit/jj(14)
  common/result/temp(200),temptrn(200)
  dimension newt(100),k(100),th(100),dens(100),shc(100)
  real kk,k,jj,newt
  parameter(sig=.1714e-8)
  parameter(fpi=3.141593/180.)
  parameter(nlayer=9)
  parameter(lend=3*nlayer)
  parameter(layair=5)
  parameter(f=0.2)
c
c Variables:
c layair=the layer (or element) that is air space in the wing
c f=fraction of an element thickness
c
  if(itotitr.eq.1.and.istep.eq.1)then
c reference thickness (th):
  th(1)=tth(1)
  th(2)=tth(2)
  th(3)=tth(3)
  th(4)=tth(4)
  th(5)=tth(29)

```

th(6)=tth(30)

th(7)=tth(5)

th(8)=tth(6)

th(9)=tth(7)

c reference conductivity (k):

k(1)=kk(1)

k(2)=kk(2)

k(3)=kk(3)

k(4)=kk(4)

k(5)=kk(29)

k(6)=kk(30)

k(7)=kk(5)

k(8)=kk(6)

k(9)=kk(7)

c reference density (dens):

dens(1)=density(1)

dens(2)=density(2)

dens(3)=density(3)

dens(4)=density(4)

dens(6)=density(6)

dens(7)=density(7)

dens(8)=density(8)

dens(9)=density(9)

c reference specific heat capacity (shc):

shc(1)=sheatcap(1)

shc(2)=sheatcap(2)

shc(3)=sheatcap(3)

shc(4)=sheatcap(4)

shc(6)=sheatcap(6)

shc(7)=sheatcap(7)

shc(8)=sheatcap(8)

shc(9)=sheatcap(9)

endif

if(istep.eq.1)then

c reference the initial temp. (t):

t(1)=tt(9)

t(2)=(tt(9)+tt(10))/2.

t(3)=tt(10)

t(4)=tt(10)

t(5)=(tt(10)+tt(11))/2.

t(6)=tt(11)

t(7)=tt(11)

t(8)=(tt(11)+tt(12))/2.

t(9)=tt(12)

t(10)=tt(12)

t(11)=(tt(12)+tt(64))/2.

t(12)=tt(64)

t(16)=tt(65)

t(17)=(tt(65)+tt(13))/2.

t(18)=tt(13)

t(19)=tt(13)

```

t(20)=(tt(13)+tt(14))/2.
t(21)=tt(14)
t(22)=tt(14)
t(23)=(tt(14)+tt(15))/2.
t(24)=tt(15)
t(25)=tt(15)
t(26)=(tt(15)+tt(16))/2.
t(27)=tt(16)
c reference the initial radiosity (jj):
  jj(3)=tt(42)
  jj(4)=tt(43)
  return
endif
c
c formulate transient cooling:
c
c upper wing surface temp (t1), as well as internal temp t2 and t3:
  qnetsky=(sig*t(1)**4-rfw*jj(3))/((1.-rfw*eemisw)/eemisw)
  newt(1)=t(1)+delttime/(dens(1)*shc(1)*f/2*th(1))*
&    ((absorbw*qreflfrw(1,iperiod)*sin(fpi*theta)+
&    qscabfwr(1,iperiod))*sunfact+
&    (absorbw*qreflfrw(1,iperiod)*sin(fpi*theta)+
&    qscabfwr(1,iperiod))*(1-sunfact)
&    -qnetsky-hhw*(t(1)-ta)-
&    (t(1)-t(2))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
  newt(2)=t(2)+delttime/(dens(1)*shc(1)*(1-f)*th(1))*
&    ((t(1)-t(2))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-
&    (t(2)-t(3))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
  newt(3)=t(3)+delttime/(dens(1)*shc(1)*f/2*th(1))*
&    ((t(2)-t(3))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-
&    (t(3)-t(4))/(f/4*th(1)/k(1)+f/4*th(2)/k(2)))
c internal nodes (all conduction): t4-t24:
  do 20 i=1,nlayer
    if(i.eq.layair)goto 20
    if(i.eq.1.or.i.eq.nlayer)goto 20
    ii=(i-1)*3
c beginning element within layer i: t4,t7,t10,t13,t16,t19,t22:
    if(i.eq.layair+1)then
      newt(ii+1)=t(ii+1)+delttime/(dens(i)*shc(i)*f/2*th(i))*
&    ((t(ii-3)-t(ii+1))/(f/4*th(i-2)/k(i-2)+th(i-1)/k(i-1)+
&    f/4*th(i)/k(i))-
&    (t(ii+1)-t(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
    else
      newt(ii+1)=t(ii+1)+delttime/(dens(i)*shc(i)*f/2*th(i))*
&    ((t(ii)-t(ii+1))/(f/4*th(i-1)/k(i-1)+f/4*th(i)/k(i))-
&    (t(ii+1)-t(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
    endif
c middle element within layer i: t5,t8,t11,t14,t17,t20,t23:
    newt(ii+2)=t(ii+2)+delttime/(dens(i)*shc(i)*(1-f)*th(i))*
&    ((t(ii+1)-t(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
&    (t(ii+2)-t(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))

```

```

c ending element within layer i: t6,t9,t12,t15,t18,t21,t24:
  if(i.eq.layair-1)then
    newt(ii+3)=t(ii+3)+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((t(ii+2)-t(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (t(ii+3)-t(ii+7))/(f/4*th(i)/k(i)+th(i+1)/k(i+1)+
& f/4*th(i+2)/k(i+2)))
    else
    newt(ii+3)=t(ii+3)+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((t(ii+2)-t(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (t(ii+3)-t(ii+4))/(f/4*th(i)/k(i)+f/4*th(i+1)/k(i+1)))
    endif
20 continue
c lower wing surface temp (t27), as well as internal temp t25 and t26:
  newt(lend-2)=t(lend-2)+
& deltime/(dens(nlayer)*shc(nlayer)*f/2*th(nlayer))*
& ((t(lend-3)-t(lend-2))/
& (f/4*th(nlayer-1)/k(nlayer-1)+f/4*th(nlayer)/k(nlayer))-
& (t(lend-2)-t(lend-1))/
& (f/4*th(nlayer)/k(nlayer)+(1-f)/2*th(nlayer)/k(nlayer)))
  newt(lend-1)=t(lend-1)+
& deltime/(dens(nlayer)*shc(nlayer)*(1-f)*th(nlayer))*
& ((t(lend-2)-t(lend-1))/
& (f/4*th(nlayer)/k(nlayer)+(1-f)/2*th(nlayer)/k(nlayer))-
& (t(lend-1)-t(lend))/
& (f/4*th(nlayer)/k(nlayer)+(1-f)/2*th(nlayer)/k(nlayer)))
  qnetsky=(sig*t(lend)**4-jj(4))/((1.-emisw)/emisw)
  newt(lend)=t(lend)+
& deltime/(dens(nlayer)*shc(nlayer)*f/2*th(nlayer))*
& (qabstwa2(3,iperiod)-qnetsky-hhw*(t(lend)-ta)+
& (t(lend-1)-t(lend))/
& (f/4*th(nlayer)/k(nlayer)+(1-f)/2*th(nlayer)/k(nlayer)))
  do 60 i=1,lend
60 t(i)=newt(i)
  temptrn(11)=t(1)-459.67
  temptrn(12)=(t(3)+t(4))/2.-459.67
  temptrn(13)=(t(6)+t(7))/2.-459.67
  temptrn(14)=(t(9)+t(10))/2.-459.67
  temptrn(15)=t(12)-459.67
  temptrn(16)=t(16)-459.67
  temptrn(17)=(t(18)+t(19))/2.-459.67
  temptrn(18)=(t(21)+t(22))/2.-459.67
  temptrn(19)=(t(24)+t(25))/2.-459.67
  temptrn(20)=t(27)-459.67
  return
end
c
c
  subroutine fusglr1(itotitr,icool,iperiod,istep,theta,ta,hhf,hhc,
& rff,absorbf,eemisf,sunfact,sunfact2)
c purpose: transient analysis of the fuselage (left to right)
c called from: trnsien1

```

c calls: none

```
common/c2/tt(200),fn(200),jacob(200,200)
common/c3/tth(100),kk(100)
common/c6/absorbff,emisff,absorbwww,emisww,absorbfl,emisfl,absorbfi
&,emisfi,absorbpg,emispg,absorftp,emisrftp,absorfbt,emisrftb
&,absorbfb,emisfb
common/c10/aircondc,thdiffus,visckine,prandtl
common/c14/thicknes(100),density(100),conductv(100),sheatcap(100)
common/reflecq/qreflin(6,100),qscabsin(6,100),qreflfwr(6,100),
&qreflflwr(6,100),qscabfwr(6,100),qscabfwr(6,100),qabstwa1(6,100),
&qabstwa2(6,100),qabstwb(6,100)
common/temp3/t(1000)
common/data1/delttime
common/radiosit/jj(14)
common/result/temp(200),temptrn(200)
dimension newt(100),k(100),th(100),dens(100),shc(100)
real kk,k,jj,newt,nusselt2
parameter(sig=.1714e-8,gravity=32.2)
parameter(fpi=3.141593/180.)
parameter(nlayer=9)
parameter(lend=3*nlayer)
parameter(layair=5)
parameter(f=0.2)
```

c

c Variables:

c layair=the layer (or element) that is air space (the fuselage width)

c f=fraction of an element thickness

c

if(itotitr.eq.1.and.istep.eq.1)then

c reference thickness (th):

```
th(1)=tth(8)
th(2)=tth(9)
th(3)=tth(10)
th(4)=tth(23)
th(5)=tth(11)
th(6)=tth(23)
th(7)=tth(10)
th(8)=tth(9)
th(9)=tth(8)
```

c reference conductivity (k):

```
k(1)=kk(8)
k(2)=kk(9)
k(3)=kk(10)
k(4)=kk(23)
k(5)=kk(11)
k(6)=kk(23)
k(7)=kk(10)
k(8)=kk(9)
k(9)=kk(8)
```

c reference density (dens):

```
dens(1)=density(10)
```

```
dens(2)=density(11)
dens(3)=density(12)
dens(4)=density(13)
dens(6)=density(13)
dens(7)=density(12)
dens(8)=density(11)
dens(9)=density(10)
```

c reference specific heat capacity (shc):

```
shc(1)=sheatcap(10)
shc(2)=sheatcap(11)
shc(3)=sheatcap(12)
shc(4)=sheatcap(13)
shc(6)=sheatcap(13)
shc(7)=sheatcap(12)
shc(8)=sheatcap(11)
shc(9)=sheatcap(10)
endif
```

```
if(istep.eq.1)then
```

c reference the initial temp. (t):

```
t(1)=tt(17)
t(2)=(tt(17)+tt(18))/2.
t(3)=tt(18)
t(4)=tt(18)
t(5)=(tt(18)+tt(19))/2.
t(6)=tt(19)
t(7)=tt(19)
t(8)=(tt(19)+tt(20))/2.
t(9)=tt(20)
t(10)=tt(20)
t(11)=(tt(20)+tt(38))/2.
t(12)=tt(38)
t(16)=tt(39)
t(17)=(tt(39)+tt(21))/2.
t(18)=tt(21)
t(19)=tt(21)
t(20)=(tt(21)+tt(22))/2.
t(21)=tt(22)
t(22)=tt(22)
t(23)=(tt(22)+tt(23))/2.
t(24)=tt(23)
t(25)=tt(23)
t(26)=(tt(23)+tt(24))/2.
t(27)=tt(24)
```

c reference the initial radiosity (jj)

```
jj(5)=tt(44)
jj(6)=tt(45)
jj(8)=tt(47)
jj(10)=tt(49)
return
```

```
endif
```

```
if(icool.eq.2)then
```

```

rayleig2=gravity*1./((t(12)+t(16))/2.)*abs(t(12)-t(16))*
& th(layair)**3/(thdiffus*visckine)
if(rayleig2.le.(1e3*(0.2+prandtl)/prandtl))then
  nusselt2=1.0
elseif(rayleig2.gt.(1e3*(0.2+prandtl)/prandtl).and.rayleig2.lt.
& 1.0e9)then
  nusselt2=0.18*((prandtl/(0.2+prandtl))*rayleig2)**0.29
endif
endif
c
c formulate transient cooling:
c
c LHS surface temp (t1), as well as internal temp t2 and t3:
  qnetsky=(sig*t(1)**4-rff*jj(5))/((1.-rff*eemisf)/eemisf)
  newt(1)=t(1)+delttime/(dens(1)*shc(1)*f/2*th(1))*
& ((absorbfi*qreffwl(2,iperiod)*cos(fpi*theta)+
& qscabfwr(2,iperiod))*sunfact+
& qscabfwr(2,iperiod)*(1-sunfact)
& -qnetsky-hhf*(t(1)-ta)-
& (t(1)-t(2))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
  newt(2)=t(2)+delttime/(dens(1)*shc(1)*(1-f)*th(1))*
& ((t(1)-t(2))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-
& (t(2)-t(3))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
  newt(3)=t(3)+delttime/(dens(1)*shc(1)*f/2*th(1))*
& ((t(2)-t(3))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-
& (t(3)-t(4))/(f/4*th(1)/k(1)+f/4*th(2)/k(2)))
c internal nodes (all conduction): t4-t24:
  do 20 i=1,nlayer
    if(i.eq.layair)goto 20
    if(i.eq.1.or.i.eq.nlayer)goto 20
    ii=(i-1)*3
c beginning element within layer i: t4,t7,t10,t16,t19,t22:
    if(i.eq.layair+1)then
      qnetcab=(sig*t(ii+1)**4-jj(10))/((1.-emisfi)/emisfi)
      if(icool.eq.1)then
        newt(ii+1)=t(ii+1)+delttime/(dens(i)*shc(i)*f/2*th(i))*
& (-t(ii+1)-t(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& qnetcab+absorbfi*qreflin(1,iperiod)*cos(fpi*theta)+
& qscabsin(1,iperiod)-hhc*(t(ii+1)-ta))
      elseif(icool.eq.2)then
        newt(ii+1)=t(ii+1)+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((t(ii-3)-t(ii+1))/(f/4*th(i-2)/k(i-2)+
& th(i-1)/(nusselt2*k(i-1))+f/4*th(i)/k(i))-
& (t(ii+1)-t(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& qnetcab+
& (absorbfi*qreflin(1,iperiod)*cos(fpi*theta)+
& qscabsin(1,iperiod))*sunfact2)
c & (absorbfi*qreflin(3,iperiod)*cos(fpi*theta)+
c & qscabsin(3,iperiod))*(1-sunfact))
      endif
    else

```

```

    newt(ii+1)=t(ii+1)+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((t(ii)-t(ii+1))/(f/4*th(i-1)/k(i-1)+f/4*th(i)/k(i))-
& (t(ii+1)-t(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
  endif
c middle element within layer i: t5,t8,t14,t17,t20,t23:
  newt(ii+2)=t(ii+2)+delttime/(dens(i)*shc(i)*(1-f)*th(i))*
& ((t(ii+1)-t(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (t(ii+2)-t(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
c ending element within layer i: t6,t9,t12,t18,t21,t24:
  if(i.eq.layair-1)then
    qnetcab=(sig*t(ii+3)**4-jj(8))/((1.-emisfi)/emisfi)
    if(icool.eq.1)then
      newt(ii+3)=t(ii+3)+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((t(ii+2)-t(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& qnetcab+absorbfi*qreflin(3,iperiod)*cos(fpi*theta)+
& qscabsin(3,iperiod)-hhc*(t(ii+3)-ta))
    elseif(icool.eq.2)then
      newt(ii+3)=t(ii+3)+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((t(ii+2)-t(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (t(ii+3)-t(ii+7))/(f/4*th(i)/k(i)+
& th(i+1)/(nusselt2*k(i+1))+f/4*th(i+2)/k(i+2))-
& qnetcab+
& (absorbfi*qreflin(3,iperiod)*cos(fpi*theta)+
& qscabsin(3,iperiod))*sunfact2)
c & (absorbfi*qreflin(1,iperiod)*cos(fpi*theta)+
c & qscabsin(1,iperiod))*(1-sunfact))
    endif
  else
    newt(ii+3)=t(ii+3)+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((t(ii+2)-t(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (t(ii+3)-t(ii+4))/(f/4*th(i)/k(i)+f/4*th(i+1)/k(i+1)))
  endif
20 continue
c RHS fuselage surface temp (t27), as well as internal temp t25 and t26:
  newt(lend-2)=t(lend-2)+
& deltime/(dens(nlayer)*shc(nlayer)*f/2*th(nlayer))*
& ((t(lend-3)-t(lend-2))/
& (f/4*th(nlayer-1)/k(nlayer-1)+f/4*th(nlayer)/k(nlayer))-
& (t(lend-2)-t(lend-1))/
& (f/4*th(nlayer)/k(nlayer)+(1-f)/2*th(nlayer)/k(nlayer)))
  newt(lend-1)=t(lend-1)+
& deltime/(dens(nlayer)*shc(nlayer)*(1-f)*th(nlayer))*
& ((t(lend-2)-t(lend-1))/
& (f/4*th(nlayer)/k(nlayer)+(1-f)/2*th(nlayer)/k(nlayer))-
& (t(lend-1)-t(lend))/
& (f/4*th(nlayer)/k(nlayer)+(1-f)/2*th(nlayer)/k(nlayer)))
  qnetsky=(sig*t(lend)**4-rff*jj(6))/((1.-rff*eemisf)/eemisf)
  newt(lend)=t(lend)+
& deltime/(dens(nlayer)*shc(nlayer)*f/2*th(nlayer))*
& (qscabfwr(2,iperiod)*sunfact+
& (absorbfi*qreffwl(2,iperiod)*cos(fpi*theta)+

```

```

&      qscabfwl(2,iperiod))*(1-sunfact)
&      -qnetsky-hhf*(t(lend)-ta)+
&      (t(lend-1)-t(lend))/
&      (f/4*th(nlayer)/k(nlayer)+(1-f)/2*th(nlayer)/k(nlayer)))
do 60 i=1,lend
60  t(i)=newt(i)
    temptrn(39)=t(1)-459.67
    temptrn(40)=(t(3)+t(4))/2.-459.67
    temptrn(41)=(t(6)+t(7))/2.-459.67
    temptrn(42)=(t(9)+t(10))/2.-459.67
    temptrn(43)=t(12)-459.67
    temptrn(44)=t(16)-459.67
    temptrn(45)=(t(18)+t(19))/2.-459.67
    temptrn(46)=(t(21)+t(22))/2.-459.67
    temptrn(47)=(t(24)+t(25))/2.-459.67
    temptrn(48)=t(27)-459.67
    return
end
c
c
  subroutine fuslgrf1(itotitr,icool,iperiod,istep,theta,ta,hhf,hhc,
&      qdir,qsky,qtarm,sunfact2)
c purpose: transient analysis of the fuselage (top to bottom)
c called from: trnsien1
c calls: none
  common/c2/tt(200),fn(200),jacob(200,200)
  common/c3/tth(100),kk(100)
  common/c4/widthwng,lengtwng,hwng,locwng,lplane,lfuselg,hfuselg,
&      wfuselg
  common/c6/absorbff,emisff,absorbww,emisww,absorbfl,emisfl,absorbfi
&      ,emisfi,absorbpg,emispq,absorftp,emisrftp,absorfbt,emisrftb
&      ,absorfbf,emisfb
  common/c5/trnsm,absorbta,emista,tarnd,densta,conducta,shcta
  common/c10/aircondc,thdiffus,visckine,prandtl
  common/c12/a1,a2,a3,a4,a5,a6(100),a7(100)
  common/c14/thicknes(100),density(100),conductv(100),sheatcap(100)
  common/reflecq/qreflin(6,100),qscabsin(6,100),qreflwr(6,100),
&      qreflflwr(6,100),qscabfwr(6,100),qscabfwl(6,100),qabstwa1(6,100),
&      qabstwa2(6,100),qabstwb(6,100)
  common/temp4/t(1000),tpg1,tpg2,tpg3
  common/data1/delttime
  common/radiosit/jj(14)
  common/result/temp(200),temptrn(200)
  dimension newt(100),k(100),th(100),dens(100),shc(100)
  real kk,k,jj,newt,nusselt1,lfuselg,kpg,newtpg1,newtpg2,newtpg3
  parameter(sig=.1714e-8,gravity=32.2)
  parameter(fpi=3.141593/180.)
  parameter(nlayer=13)
  parameter(lend=3*nlayer)
  parameter(layair1=5,layair2=10)
  parameter(f=0.2)

```

c

c Variables:

c layair1=the layer (or element) that is air space (the fuselage height)

c layair2=the layer (or element) that is air space (in between the floor

c and belly-pan)

c f=fraction of an element thickness

c

if(itotitr.eq.1.and. istep.eq.1)then

c reference thickness (th):

th(1)=tth(27)

th(2)=tth(26)

th(3)=tth(25)

th(4)=tth(28)

th(5)=tth(24)

th(6)=tth(21)

th(7)=tth(20)

th(8)=tth(19)

th(9)=tth(18)

th(10)=tth(31)

th(11)=tth(14)

th(12)=tth(13)

th(13)=tth(12)

c reference conductivity (k):

k(1)=kk(27)

k(2)=kk(26)

k(3)=kk(25)

k(4)=kk(28)

k(5)=kk(24)

k(6)=kk(21)

k(7)=kk(20)

k(8)=kk(19)

k(9)=kk(18)

k(10)=kk(31)

k(11)=kk(14)

k(12)=kk(13)

k(13)=kk(12)

c reference density (dens):

dens(1)=density(15)

dens(2)=density(16)

dens(3)=density(17)

dens(4)=density(18)

dens(6)=density(20)

dens(7)=density(21)

dens(8)=density(22)

dens(9)=density(23)

dens(11)=density(25)

dens(12)=density(26)

dens(13)=density(27)

c reference specific heat capacity:

shc(1)=sheatcap(15)

shc(2)=sheatcap(16)

```
shc(3)=sheatcap(17)
shc(4)=sheatcap(18)
shc(6)=sheatcap(20)
shc(7)=sheatcap(21)
shc(8)=sheatcap(22)
shc(9)=sheatcap(23)
shc(11)=sheatcap(25)
shc(12)=sheatcap(26)
shc(13)=sheatcap(27)
```

c also reference plexiglass thickness, conductivity, density

c and specific heat capacity:

```
thpg=tt(22)
kpg=kk(22)
denspg=density(28)
shcp=shcp(28)
```

c also find these values:

```
c1=a1/(a1+a2)
c2=a2/(a1+a2)
emispgrf=c1*emisp+c2*emisrft
endif
if(istep.eq.1)then
```

c reference the initial temp. (t):

```
t(1)=tt(59)
t(2)=(tt(59)+tt(58))/2.
t(3)=tt(58)
t(4)=tt(58)
t(5)=(tt(58)+tt(57))/2.
t(6)=tt(57)
t(7)=tt(57)
t(8)=(tt(57)+tt(60))/2.
t(9)=tt(60)
t(10)=tt(60)
t(11)=(tt(60)+tt(61))/2.
t(12)=tt(61)
t(16)=tt(35)
t(17)=(tt(35)+tt(34))/2.
t(18)=tt(34)
t(19)=tt(34)
t(20)=(tt(34)+tt(33))/2.
t(21)=tt(33)
t(22)=tt(33)
t(23)=(tt(33)+tt(32))/2.
t(24)=tt(32)
t(25)=tt(32)
t(26)=(tt(32)+tt(31))/2.
t(27)=tt(31)
t(31)=tt(28)
t(32)=(tt(28)+tt(27))/2.
t(33)=tt(27)
t(34)=tt(27)
t(35)=(tt(27)+tt(26))/2.
```

```

t(36)=tt(26)
t(37)=tt(26)
t(38)=(tt(26)+tt(25))/2.
t(39)=tt(25)
c also reference the initial plexiglass temp.
  tpg1=tt(37)
  tpg2=(tt(37)+tt(36))/2.
  tpg3=tt(36)
c reference the initial radiosity
  jj(7)=tt(46)
  jj(9)=tt(48)
  jj(11)=tt(50)
  return
endif
tpgrf=c1*tpg3+c2*t(12)
qnet=(sig*tpgrf**4-jj(9))/((1.-emispgrf)/emispgrf)
c
  if(icool.eq.2)then
    rayleig1=gravity*1./((tpgrf+t(16))/2.)*(t(16)-tpgrf)*
&    th(5)**3/(thdiffus*visckine)
    if(rayleig1.le.3e5)then
      nusselt1=1.0
    elseif(rayleig1.gt.3e5.and.rayleig1.le.7e9)then
      nusselt1=0.069*rayleig1**(1./3.)*prandtl**.074
    endif
  endif
c
c formulate transient cooling:
c
c first for the only three temperatures of plexiglass
c
c
  newtpg1=tpg1+delttime/(a1*denspg*shcp* f/2*thpg)*
& (emisp* qsky*a1+emisp*.5*(1.-emista)*qsky*a5
& +emisp*.5*qtarm*a5
& +f/2*absorbpg*qdir*cos(fpi*theta)*a6(iperiod)
& +f/2*absorbpg*qdir*sin(fpi*theta)*a7(iperiod)
& +f/2*absorbpg*.5*(1.-absorbta)*qdir*sin(fpi*theta)*a5
& +f/2*absorbpg*qreflin(4,iperiod)*sin(fpi*theta)*a3
& +f/2*absorbpg*qscabsin(4,iperiod)*lfuselg*wfuselg
& *a3/(absorbpg*a3+absorfbt*a4)
& -emisp*sig*tpg1**4*a1
& -hhf*(tpg1-ta)*a1
& -(tpg1-tpg2)/(f/4*thpg/kpg+(1-f)/2*thpg/kpg)*a1)
  newtpg2=tpg2+delttime/(a1*denspg*shcp* (1-f)*thpg)*
& ((tpg1-tpg2)/(f/4*thpg/kpg+(1-f)/2*thpg/kpg)*a1
& -(tpg2-tpg3)/((1-f)/2*thpg/kpg+f/4*thpg/kpg)*a1
& +(1-f)*absorbpg*qdir*cos(fpi*theta)*a6(iperiod)
& +(1-f)*absorbpg*qdir*sin(fpi*theta)*a7(iperiod)
& +(1-f)*absorbpg*.5*(1.-absorbta)*qdir*sin(fpi*theta)*a5
& +(1-f)*absorbpg*qreflin(4,iperiod)*sin(fpi*theta)*a3

```

```

& +(1-f)*absorbpq*qsabsin(4,iperiod)*lfuselg*wfuselg
& *a3/(absorbpq*a3+absorfbt*a4))
  qpg=emispg*sig*tpg3**4-emispg/(a3*emispg+a4*emisrftb)*
& (a3*emispg*sig*tpg3**4+a4*emisrftb*sig*t(12)**4-qnet*(a3+a4))
  if(icool.eq.1)then
    newtpg3=tpg3+delttime/(a1*denspg*shcpg*f/2*thpg)*
& ((tpg2-tpg3)/((1-f)/2*thpg/kpg+f/4*thpg/kpg)*a1
& -hhc*(tpg3-ta)*a1
& -qpg*a3
& +f/2*absorbpq*qdir*cos(fpi*theta)*a6(iperiod)
& +f/2*absorbpq*qdir*sin(fpi*theta)*a7(iperiod)
& +f/2*absorbpq*.5*(1.-absorbta)*qdir*sin(fpi*theta)*a5
& +f/2*absorbpq*qreflin(4,iperiod)*sin(fpi*theta)*a3
& +f/2*absorbpq*qsabsin(4,iperiod)*lfuselg*wfuselg
& *a3/(absorbpq*a3+absorfbt*a4))
  elseif(icool.eq.2)then
    newtpg3=tpg3+delttime/(a1*denspg*shcpg*f/2*thpg)*
& ((tpg2-tpg3)/((1-f)/2*thpg/kpg+f/4*thpg/kpg)*a1
& -(tpgrf-t(16))/(th(5)/(nusselt1*k(5)))*a3
& -qpg*a3
& +f/2*absorbpq*qdir*cos(fpi*theta)*a6(iperiod)
& +f/2*absorbpq*qdir*sin(fpi*theta)*a7(iperiod)
& +f/2*absorbpq*.5*(1.-absorbta)*qdir*sin(fpi*theta)*a5
& +f/2*absorbpq*qreflin(4,iperiod)*sin(fpi*theta)*a3
& +f/2*absorbpq*qsabsin(4,iperiod)*lfuselg*wfuselg
& *a3/(absorbpq*a3+absorfbt*a4))
  endif
c
c next for the roof structure and the rest
c
c first three temperatures of roof structure:
  qnetsky=(emisrftb*sig*t(1)**4-emisrftb*qsky)
  newt(1)=t(1)+delttime/(dens(1)*shc(1)*f/2*th(1))*
& (absorfbt*qdir*sin(fpi*theta)-qnetsky-hhf*(t(1)-ta)-
& (t(1)-t(2))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
  newt(2)=t(2)+delttime/(dens(1)*shc(1)*(1-f)*th(1))*
& ((t(1)-t(2))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-
& (t(2)-t(3))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
  newt(3)=t(3)+delttime/(dens(1)*shc(1)*f/2*th(1))*
& ((t(2)-t(3))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-
& (t(3)-t(4))/(f/4*th(1)/k(1)+f/4*th(2)/k(2)))
c internal nodes: t4-t36:
  do 20 i=1,nlayer
    if(i.eq.layair1.or.i.eq.layair2)goto 20
    if(i.eq.1.or.i.eq.nlayer)goto 20
    ii=(i-1)*3
c beginning element within layer i: t4,t7,t10,t16,t19,t22,t25,t31,t34:
    if(i.eq.layair1+1)then
      qnetcab=(sig*t(ii+1)**4-jj(11))/((1.-emisfl)/emisfl)
      if(icool.eq.1)then
        newt(ii+1)=t(ii+1)+delttime/(dens(i)*shc(i)*f/2*th(i))*

```

```

&      (- (t(ii+1)-t(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
&      hhc*(t(ii+1)-ta)-
&      qnetcab+absorbfl*qreflin(2,iperiod)*sin(fpi*theta)+
&      qscabsin(2,iperiod))
      elseif(icool.eq.2)then
        newt(ii+1)=t(ii+1)+delttime/(dens(i)*shc(i)*f/2*th(i))*
&      ((tpgrf-t(ii+1))/(th(i-1)/(nusselt1*k(i-1)))-
&      (t(ii+1)-t(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
&      qnetcab+
&      (absorbfl*qreflin(2,iperiod)*sin(fpi*theta)+
&      qscabsin(2,iperiod))*sunfact2)
      endif
      elseif(i.eq.layair2+1)then
        newt(ii+1)=t(ii+1)+delttime/(dens(i)*shc(i)*f/2*th(i))*
&      ((t(ii-3)-t(ii+1))/(f/4*th(i-2)/k(i-2)+th(i-1)/k(i-1)+
&      f/4*th(i)/k(i))-
&      (t(ii+1)-t(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
      else
        newt(ii+1)=t(ii+1)+delttime/(dens(i)*shc(i)*f/2*th(i))*
&      ((t(ii)-t(ii+1))/(f/4*th(i-1)/k(i-1)+f/4*th(i)/k(i))-
&      (t(ii+1)-t(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
      endif
c middle element within layer i: t5,t8,t11,t17,t20,t23,t26,t32,t35:
      newt(ii+2)=t(ii+2)+delttime/(dens(i)*shc(i)*(1-f)*th(i))*
&      ((t(ii+1)-t(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
&      (t(ii+2)-t(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
c ending element within layer i: t6,t9,t12,t18,t21,t24,t27,t33,t36:
      if(i.eq.layair1-1)then
        qrftb=emisrftb*sig*t(ii+3)**4-emisrftb/(a3*emispg+a4*emisrftb)*
&      (a3*emispg*sig*tpg3**4+a4*emisrftb*sig*t(ii+3)**4-qnet*(a3+a4))
        if(icool.eq.1)then
          newt(ii+3)=t(ii+3)+delttime/(a2*dens(i)*shc(i)*f/2*th(i))*
&      ((t(ii+2)-t(ii+3))/((1-f)/2*th(i)/k(i)+f/4*th(i)/k(i))*a2-
&      hhc*(t(ii+3)-ta)*a2
&      -qrftb*a4
&      +absorbft*qreflin(4,iperiod)*sin(fpi*theta)*a4
&      +absorbft*qscabsin(4,iperiod)*lfuselg*wfuselg*a4/
&      (absorbpg*a3+absorbft*a4))
        elseif(icool.eq.2)then
          newt(ii+3)=t(ii+3)+delttime/(a2*dens(i)*shc(i)*f/2*th(i))*
&      ((t(ii+2)-t(ii+3))/((1-f)/2*th(i)/k(i)+f/4*th(i)/k(i))*a2-
&      (tpgrf-t(ii+7))/(th(i+1)/(nusselt1*k(i+1)))*a4
&      -qrftb*a4
&      +(absorbft*qreflin(4,iperiod)*sin(fpi*theta)*a4
&      +absorbft*qscabsin(4,iperiod)*lfuselg*wfuselg*a4/
&      (absorbpg*a3+absorbft*a4))*sunfact2)
        endif
      elseif(i.eq.layair2-1)then
        newt(ii+3)=t(ii+3)+delttime/(dens(i)*shc(i)*f/2*th(i))*
&      ((t(ii+2)-t(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
&      (t(ii+3)-t(ii+7))/(f/4*th(i)/k(i)+th(i+1)/k(i+1))+

```

```

&                                f/4*th(i+2)/k(i+2)))
    else
      newt(ii+3)=t(ii+3)+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((t(ii+2)-t(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (t(ii+3)-t(ii+4))/(f/4*th(i)/k(i)+f/4*th(i+1)/k(i+1)))
    endif
20  continue
c belly-pan temp (t39), as well as internal temp t37 and t38:
  newt(lend-2)=t(lend-2)+
&   deltime/(dens(nlayer)*shc(nlayer)*f/2*th(nlayer))*
&   ((t(lend-3)-t(lend-2))/
&   (f/4*th(nlayer-1)/k(nlayer-1)+f/4*th(nlayer)/k(nlayer))-
&   (t(lend-2)-t(lend-1))/
&   (f/4*th(nlayer)/k(nlayer)+(1-f)/2*th(nlayer)/k(nlayer)))
  newt(lend-1)=t(lend-1)+
&   deltime/(dens(nlayer)*shc(nlayer)*(1-f)*th(nlayer))*
&   ((t(lend-2)-t(lend-1))/
&   (f/4*th(nlayer)/k(nlayer)+(1-f)/2*th(nlayer)/k(nlayer))-
&   (t(lend-1)-t(lend))/
&   (f/4*th(nlayer)/k(nlayer)+(1-f)/2*th(nlayer)/k(nlayer)))
  qnetsky=(sig*t(lend)**4-jj(7))/((1.-emisfb)/emisfb)
  newt(lend)=t(lend)+
&   deltime/(dens(nlayer)*shc(nlayer)*f/2*th(nlayer))*
&   (qabstwb(3,iperiod)-qnetsky-hhf*(t(lend)-ta)+
&   (t(lend-1)-t(lend))/
&   (f/4*th(nlayer)/k(nlayer)+(1-f)/2*th(nlayer)/k(nlayer)))
  tpg1=newtpg1
  tpg2=newtpg2
  tpg3=newtpg3
  do 60 i=1,lend
60  t(i)=newt(i)
  temptrn(21)=t(1)-459.67
  temptrn(22)=(t(3)+t(4))/2.-459.67
  temptrn(23)=(t(6)+t(7))/2.-459.67
  temptrn(24)=(t(9)+t(10))/2.-459.67
  temptrn(25)=t(12)-459.67
  temptrn(26)=t(16)-459.67
  temptrn(27)=(t(18)+t(19))/2.-459.67
  temptrn(28)=(t(21)+t(22))/2.-459.67
  temptrn(29)=(t(24)+t(25))/2.-459.67
  temptrn(30)=t(27)-459.67
  temptrn(31)=(t(27)+t(31))/2.-459.67
  temptrn(32)=(t(27)+t(31))/2.-459.67
  temptrn(33)=(t(27)+t(31))/2.-459.67
  temptrn(34)=(t(27)+t(31))/2.-459.67
  temptrn(35)=t(31)-459.67
  temptrn(36)=(t(33)+t(34))/2.-459.67
  temptrn(37)=(t(36)+t(37))/2.-459.67
  temptrn(38)=t(39)-459.67
  temptrn(49)=tpg1-459.67
  temptrn(50)=tpg3-459.67

```

```

    return
  end
c
  subroutine tarmac1(itotitr,icool,iperiod,istep,ta,ttarm,hht)
c purpose: transient analysis of the tarmac
c called from: trnsien1
c calls: none
  common/c4/widthwng,lengthwng,hwng,locwng,lplane,lfuselg,hfuselg,
    & wfuselg
  common/c5/trnsm,absorbta,emista,tarmd,densta,conducta,shcta
  common/c2/tt(200),fn(200),jacob(200,200)
  common/c3/tth(100),kk(100)
  common/c14/thicknes(100),density(100),conductv(100),sheatcap(100)
  common/reflecq/qreflin(6,100),qscabsin(6,100),qreflfwr(6,100),
    &qreflflw(6,100),qscabfwr(6,100),qscabfwr(6,100),qabstwa1(6,100),
    &qabstwa2(6,100),qabstwb(6,100)
  common/temp5/t(1000)
  common/data1/delttime
  common/radiosit/jj(14)
  common/facttarm/fact
  common/result/temp(200),temptrn(200)
  dimension newt(100)
  real jj,newt,lengthwng
  parameter(sig=.1714e-8)
  parameter(fpi=3.141593/180.)
  parameter(afact=1.0,bfact=0.00001)
c
c Variables:
c afact=cooling when door is opened: no motion: therefore tarmac temperature
c   underneath the wing remains very constant
c bfact=a small number to force the tarmac temperature underneath the wing
c   to be the same as the unsheltered tarmac temperature: in effect when
c   the aircraft is in motion
c
  if(itotitr.eq.1.and.istep.eq.1)then
    rlwng=(lengthwng-wfuselg)/2.
  endif
  if(istep.eq.1)then
c reference the initial temp.:
    t(51)=tt(51)
    t(52)=tt(52)
    t(53)=tt(53)
c reference the initial radiosity
    jj(12)=tt(54)
    jj(13)=tt(55)
    jj(14)=tt(56)
    return
  endif
  if(icool.eq.1)fact1=afact
  if(icool.eq.2)fact1=bfact
c

```

c formulate transient cooling:

c

c LHS Wing/Tarmac t(51):

```

  qnetsky=(sig*t(51)**4-jj(12))/((1.-emista)/emista)
  newt(51)=t(51)+deltim/(densta*rlwng*widthwng*tarmd*shcta)*
& ((qabstwa1(1,iperiod)-qnetsky-hht*(t(51)-ta))*rlwng*widthwng+
& 2.*(conducta/(fact1*fact*widthwng)*(ttarm-t(51)))*tarmd*rlwng+
& conducta/(fact1*fact*widthwng)*(ttarm-t(51))*tarmd*widthwng-
& conducta/(.5*wfuselg+.5*rlwng)*(t(51)-t(52))*tarmd*widthwng)

```

c RHS Wing/Tarmac t(53):

```

  qnetsky=(sig*t(53)**4-jj(14))/((1.-emista)/emista)
  newt(53)=t(53)+deltim/(densta*rlwng*widthwng*tarmd*shcta)*
& ((qabstwa2(1,iperiod)-qnetsky-hht*(t(53)-ta))*rlwng*widthwng+
& 2.*(conducta/(fact1*fact*widthwng)*(ttarm-t(53)))*tarmd*rlwng+
& conducta/(fact1*fact*widthwng)*(ttarm-t(53))*tarmd*widthwng-
& conducta/(.5*wfuselg+.5*rlwng)*(t(53)-t(52))*tarmd*widthwng)

```

c MIDDLE Wing/Tarmac t(52):

```

  qnetsky=(sig*t(52)**4-jj(13))/((1.-emista)/emista)
  newt(52)=t(52)+deltim/(densta*wfuselg*widthwng*tarmd*shcta)*
& ((qabstw(1,iperiod)-qnetsky-hht*(t(52)-ta))*wfuselg*widthwng+
& 2.*(conducta/(fact1*fact*widthwng)*(ttarm-t(51)))*tarmd*wfuselg
& +conducta/(.5*wfuselg+.5*rlwng)*(t(51)-t(52))*tarmd*widthwng+
& conducta/(.5*wfuselg+.5*rlwng)*(t(53)-t(52))*tarmd*widthwng)
  t(51)=newt(51)
  t(52)=newt(52)
  t(53)=newt(53)
  temptrn(51)=t(51)-459.67
  temptrn(52)=t(52)-459.67
  temptrn(53)=t(53)-459.67
  return
end

```

c

```

  subroutine updradio(itotitr,iperiod,istep,qskey,qtarm,emisf,
& emisw,eemisf,eemisw,rff,rfw)

```

c purpose: updates the radiosity (14 of them) during the cooling phase

c ie:a*radiosity=tempfact

c called from: trnsien1 and trnsien2

c calls: ludcmp,lubksb

```

  common/c1/n,ntrial,tolx,tolf
  common/c2/x(200),fvec(200),a(200,200)
  common/sc1/indx(200),b(200)
  common/c4/widthwng,lengtwnng,hwng,locwng,lplane,lfuselg,hfuselg,
& wfuselg
  common/c5/trnsm,absorbta,emista,tarmd,densta,conducta,shcta
  common/c6/absorbff,emisff,absorbww,emisww,absorbfl,emisfl,absorbfi
& ,emisfi,absorbpbg,emispbg,absorftp,emisrftp,absorfbt,emisrftb
& ,absorfbf,emisfb
  common/c7/nperiods,envtdata(100,20)
  common/c12/a1,a2,a3,a4,a5,a6(100),a7(100)
  common/temp1/t1(1000)
  common/temp2/t2(1000)

```

```

common/temp3/t3(1000)
common/temp4/t4(1000),tpg1,tpg2,tpg3
common/temp5/t5(1000)
common/radiosit/radiosit(14)
common/vf1/fbwa(6,6),fbwb(6,6)
common/vf2/fa11,fa12,fa13,fa14,fa15,fa16,fa21,fa22,fa23,fa24,
&      fa25,fa26,fa31,fa32,fa33,fa34,fa35,fa36,fa41,fa42,
&      fa43,fa44,fa45,fa46,fa51,fa52,fa53,fa54,fa55,fa56,
&      fa61,fa62,fa63,fa64,fa65,fa66
common/vf4/fa57,fa68
common/vf5/fc11,fc12,fc13,fc14,fc15,fc16,fc21,fc22,fc23,fc24,
&      fc25,fc26,fc31,fc32,fc33,fc34,fc35,fc36,fc41,fc42,
&      fc43,fc44,fc45,fc46,fc51,fc52,fc53,fc54,fc55,fc56,
&      fc61,fc62,fc63,fc64,fc65,fc66
common/result/temp(200),temptrn(200)
dimension t(100),aa(14,14),iindx(14)
dimension tempfact(14)
parameter(sig=.1714e-8)
real lfuselg
c
c Variables:
c tempfact()=updated term that is a function of the previous temperature
c aa(,)=matrix relating radiosity to tempfact; to be inverted
c radiosity()=updated radiosity
c
  if(iperiod.eq.1.and.istep.eq.1)then
c find the a matrix relating radiosity jj to the temperature term tempfact
c because the a matrix is a constant, you need to find it and its
c inverse only once for a given paint scheme
c
  n=14
c initialize a(i,j) to zeroes initially because most of the coefficients are
c zeroes
  do 15 i=1,n
    do 15 j=1,n
      a(i,j)=0.0
  15 continue
c
c LHS Wing/Fuselage :::Wing
c J1:::a(1,1),a(1,5),tempfact(1); from fn(24):
  a(1,1)=1/(1/fa12)+1/(1/fa13)+1/(1/fa14)+1/(1/fa15)+1/(1/fa16)
  &      +rfw/((1-rfw*eemisw)/eemisw)
  a(1,5)=-1/(1/fa12)
c LHS Wing/Fuselage :::Fuselage
c J5:::a(5,1),a(5,5),tempfact(5); from fn(28):
  a(5,1)=-1/(1/fa21)
  a(5,5)=1/(1/fa21)+1/(1/fa23)+1/(1/fa24)+1/(1/fa25)+1/(1/fa26)
  &      +rff/((1-rff*eemisf)/eemisf)
c RHS Wing/Fuselage :::Wing
c J3:::a(3,3),a(3,6),tempfact(3); from fn(26):
  a(3,3)=1/(1/fa12)+1/(1/fa13)+1/(1/fa14)+1/(1/fa15)+1/(1/fa16)

```

```

&      +rfw/((1-rfw*eemisw)/eemisw)
a(3,6)=-1/(1/fa12)
c RHS Wing/Fuselage :::Fuselage
c J6:::a(6,3),a(6,6),tempfact(6); from fn(30):
  a(6,3)=-1/(1/fa21)
  a(6,6)=1/(1/fa21)+1/(1/fa23)+1/(1/fa24)+1/(1/fa25)+1/(1/fa26)
  &      +rff/((1-rff*eemisf)/eemisf)
c LHS Wing/Tarmac :::Wing
c J2:::a(2,2),a(2,12),a(2,13),a(2,14),tempfact(2); from fn(20):
  a(2,2)=1/(1/fbwa(3,1))+1/(1/fbwa(3,2))+1/(1/fbwa(3,4))+
  &      1/(1/fbwa(3,5))+1/(1/fbwa(3,6))+1/((1-emisw)/emisw)
  a(2,12)=-1/(1/fbwa(3,1))
  a(2,13)=-fbwb(4,1)/(1/fbwa(3,2))
  a(2,14)=-fbwa(4,1)*fbwb(4,2)/(1/fbwa(3,2))
c RHS Wing/Tarmac :::Wing
c J4:::a(4,4),a(4,12),a(4,13),a(4,14),tempfact(4); from fn(22):
  a(4,4)=1/(1/fbwa(3,1))+1/(1/fbwa(3,2))+1/(1/fbwa(3,4))+
  &      1/(1/fbwa(3,5))+1/(1/fbwa(3,6))+1/((1-emisw)/emisw)
  a(4,12)=-fbwa(2,1)*fbwb(2,4)/(1/fbwa(3,4))
  a(4,13)=-fbwb(2,1)/(1/fbwa(3,4))
  a(4,14)=-1/(1/fbwa(3,1))
c MIDDLE Wing/Tarmac :::Wing
c J7:::a(7,7),a(7,12),a(7,13),a(7,14),tempfact(7); from fn(46):
  a(7,7)=1/(1/fbwb(3,1))+1/(1/fbwb(3,2))+1/(1/fbwb(3,4))+
  &      1/(1/fbwb(3,5))+1/(1/fbwb(3,6))+1/((1-emisfb)/emisfb)
  a(7,12)=-fbwa(2,1)/(1/fbwb(3,4))
  a(7,13)=-1/(1/fbwb(3,1))
  a(7,14)=-fbwa(4,1)/(1/fbwb(3,2))
c LHS Wing/Tarmac :::Tarmac
c J12:::a(12,2),a(12,4),a(12,7),a(12,12),tempfact(12); from fn(54):
  a(12,2)=-1/(1/fbwa(1,3))
  a(12,4)=-fbwa(4,3)*fbwb(4,2)/(1/fbwa(1,2))
  a(12,7)=-fbwb(4,3)/(1/fbwa(1,2))
  a(12,12)=1/(1/fbwa(1,2))+1/(1/fbwa(1,3))+1/(1/fbwa(1,4))+
  &      1/(1/fbwa(1,5))+1/(1/fbwa(1,6))+1/((1-emista)/emista)
c RHS Wing/Tarmac :::Tarmac
c J14:::a(14,2),a(14,4),a(14,7),a(14,14),tempfact(14); from fn(56):
  a(14,2)=-fbwa(2,3)*fbwb(2,4)/(1/fbwa(1,4))
  a(14,4)=-1/(1/fbwa(1,3))
  a(14,7)=-fbwb(2,3)/(1/fbwa(1,4))
  a(14,14)=1/(1/fbwa(1,2))+1/(1/fbwa(1,3))+1/(1/fbwa(1,4))+
  &      1/(1/fbwa(1,5))+1/(1/fbwa(1,6))+1/((1-emista)/emista)
c MIDDLE Wing/Tarmac :::Tarmac
c J13:::a(13,2),a(13,4),a(13,7),a(13,13),tempfact(13); from fn(55):
  a(13,2)=-fbwa(2,3)/(1/fbwb(1,4))
  a(13,4)=-fbwa(4,3)/(1/fbwb(1,2))
  a(13,7)=-1/(1/fbwb(1,3))
  a(13,13)=1/(1/fbwb(1,2))+1/(1/fbwb(1,3))+1/(1/fbwb(1,4))+
  &      1/(1/fbwb(1,5))+1/(1/fbwb(1,6))+1/((1-emista)/emista)
c FUSELAGE INTERIOR (CABIN)
c LHS Fuselage Interior (Cabin)

```

```

c J8:::a(8,8),a(8,9),a(8,10),a(8,11),tempfact(8); from fn(47):
  a(8,8)=1/(1/fc41)+1/(1/fc42)+1/(1/fc43)+1/(1/fc45)+1/(1/fc46)
  &      +1/((1-emisfi)/emisfi)
  a(8,9)=-1/(1/fc43)
  a(8,10)=-1/(1/fc42)
  a(8,11)=-1/(1/fc41)
c RHS Fuselage Interior (Cabin)
c J10:::a(10,8),a(10,9),a(10,10),a(10,11),tempfact(10); from fn(49):
  a(10,8)=-1/(1/fc24)
  a(10,9)=-1/(1/fc23)
  a(10,10)=1/(1/fc21)+1/(1/fc23)+1/(1/fc24)+1/(1/fc25)+1/(1/fc26)
  &      +1/((1-emisfi)/emisfi)
  a(10,11)=-1/(1/fc21)
c BOTTOM (floor) Fuselage Interior (Cabin)
c J11:::a(11,8),a(11,9),a(11,10),a(11,11),tempfact(11); from fn(50):
  a(11,8)=-1/(1/fc14)
  a(11,9)=-1/(1/fc13)
  a(11,10)=-1/(1/fc12)
  a(11,11)=1/(1/fc12)+1/(1/fc13)+1/(1/fc14)+1/(1/fc15)+1/(1/fc16)
  &      +1/((1-emisfi)/emisfi)
c TOP (combined roof and plexiglass structure) Fuselage Interior (Cabin)
c J9:::a(9,8),a(9,9),a(9,10),a(9,11),tempfact(9); from fn(48):
  c1=a1/(a1+a2)
  c2=a2/(a1+a2)
  emispgf=c1*emispgr+c2*emisrft
  a(9,8)=-1/(1/fc34)
  a(9,9)=1/(1/fc31)+1/(1/fc32)+1/(1/fc34)+1/(1/fc35)+1/(1/fc36)
  &      +1/((1-emispgrf)/emispgrf)
  a(9,10)=-1/(1/fc32)
  a(9,11)=-1/(1/fc31)
c   if(itotitr.le.nperiods)write(17,*)'matrix a'
c   do 10 i=1,14
c 10   if(itotitr.le.nperiods)write(17,80)(a(i,j),j=1,14)
  80   format(2x,500(1x,e10.3))
  call ludcmp
c   if(itotitr.le.nperiods)write(17,*)'decomposed matrix a'
c   do 20 i=1,14
c 20   if(itotitr.le.nperiods)write(17,80)(a(i,j),j=1,14)
c replace a(i,j) with aa(i,j), bec. a(i,j) gets changed in the steady-state
c analysis phase. same thing for index(i)
  do 25 i=1,14
    iindx(i)=indx(i)
    do 25 j=1,14
      25 aa(i,j)=a(i,j)
  return
endif

c
c find the updated temperature term tempfact(i)
c ie A*J=T, where T is the tempfact. then find updated J. A is constant
c first relate t(i) to t1(i),t2(i),t3(i),tpg3,t4(i) and t5(i) obtained from
c transient analysis

```

```

t(1)=t1(1)
t(8)=t1(27)
t(9)=t2(1)
t(16)=t2(27)
t(17)=t3(1)
t(38)=t3(12)
t(39)=t3(16)
t(24)=t3(27)
t(36)=tpg3
t(61)=t4(12)
t(35)=t4(16)
t(25)=t4(39)
t(51)=t5(51)
t(52)=t5(52)
t(53)=t5(53)
c LHS Wing/Fuselage :::Wing
c J1:::a(1,1),a(1,5),tempfact(1); from fn(24):
  q1=qsky+fa57*(emisf*sig*t(17)**4
  & +(1-emisf)*(qsky+.5*qtarm+.5*(1.-emista)*qsky))
  q2=qsky+fa68*(emisf*sig*t(17)**4
  & +(1-emisf)*(qsky+.5*qtarm+.5*(1.-emista)*qsky))
  tempfact(1)=qsky/(1/fa13)+qsky/(1/fa14)+q1/(1/fa15)+q2/(1/fa16)
  & +sig*t(1)**4/((1-rfw*eemisw)/eemisw)
c LHS Wing/Fuselage :::Fuselage
c J5:::a(5,1),a(5,5),tempfact(5); from fn(28):
  q1=qsky+.5*qtarm+.5*(1-emista)*qsky
  tempfact(5)=qsky/(1/fa23)+q1/(1/fa24)+q1/(1/fa25)+q1/(1/fa26)
  & +sig*t(17)**4/((1-rff*eemisf)/eemisf)
c RHS Wing/Fuselage :::Wing
c J3:::a(3,3),a(3,6),tempfact(3); from fn(26):
  q1=qsky+fa57*(emisf*sig*t(24)**4
  & +(1-emisf)*(qsky+.5*qtarm+.5*(1.-emista)*qsky))
  q2=qsky+fa68*(emisf*sig*t(24)**4
  & +(1-emisf)*(qsky+.5*qtarm+.5*(1.-emista)*qsky))
  tempfact(3)=qsky/(1/fa13)+qsky/(1/fa14)+q1/(1/fa15)+q2/(1/fa16)
  & +sig*t(9)**4/((1-rfw*eemisw)/eemisw)
c RHS Wing/Fuselage :::Fuselage
c J6:::a(6,3),a(6,6),tempfact(6); from fn(30):
  q1=qsky+.5*qtarm+.5*(1-emista)*qsky
  tempfact(6)=qsky/(1/fa23)+q1/(1/fa24)+q1/(1/fa25)+q1/(1/fa26)
  & +sig*t(24)**4/((1-rff*eemisf)/eemisf)
c LHS Wing/Tarmac :::Wing
c J2:::a(2,2),a(2,12),a(2,13),a(2,14),tempfact(2); from fn(20):
c find qskytarm first
  qskytarm=qsky+0.5*qtarm+0.5*(1.-emista)*qsky
  tempfact(2)=1/(1/fbwa(3,2))*
  & ((qskytarm*fbwa(4,2)+qskytarm*fbwa(4,5)+qskytarm*fbwa(4,6))*
  & fbwb(4,2)+qskytarm*fbwb(4,5)+qskytarm*fbwb(4,6))+
  & qskytarm/(1/fbwa(3,4))+qskytarm/(1/fbwa(3,5))+
  & qskytarm/(1/fbwa(3,6))+sig*t(8)**4/((1-emisw)/emisw)
c RHS Wing/Tarmac :::Wing

```

```

c J4::a(4,4),a(4,12),a(4,13),a(4,14),tempfact(4); from fn(22):
  tempfact(4)=qskytarm/(1/fbwa(3,2))+1/(1/fbwa(3,4))*
  & ((qskytarm*fbwa(2,4)+qskytarm*fbwa(2,5)+qskytarm*fbwa(2,6))*
  & fbwb(2,4)+qskytarm*fbwb(2,5)+qskytarm*fbwb(2,6))+
  & qskytarm/(1/fbwa(3,5))+
  & qskytarm/(1/fbwa(3,6))+sig*t(16)**4/((1-emisw)/emisw)
c MIDDLE Wing/Tarmac :::Wing
c J7::a(7,7),a(7,12),a(7,13),a(7,14),tempfact(7); from fn(46):
  tempfact(7)=1/(1/fbwb(3,2))*
  & (qskytarm*fbwa(4,2)+qskytarm*fbwa(4,5)+qskytarm*fbwa(4,6))+
  & 1/(1/fbwb(3,4))*
  & (qskytarm*fbwa(2,4)+qskytarm*fbwa(2,5)+qskytarm*fbwa(2,6))+
  & qskytarm/(1/fbwb(3,5))+
  & qskytarm/(1/fbwb(3,6))+
  & sig*t(25)**4/((1-emisfb)/emisfb)
c LHS Wing/Tarmac :::Tarmac
c J12::a(12,2),a(12,4),a(12,7),a(12,12),tempfact(12); from fn(54):
  tempfact(12)=1/(1/fbwa(1,2))*
  & ((qsky*fbwa(4,2)+qsky*fbwa(4,5)+qsky*fbwa(4,6))*
  & fbwb(4,2)+qsky*fbwb(4,5)+qsky*fbwb(4,6))+
  & qsky/(1/fbwa(1,4))+qsky/(1/fbwa(1,5))+
  & qsky/(1/fbwa(1,6))+sig*t(51)**4/((1-emista)/emista)
c RHS Wing/Tarmac :::Tarmac
c J14::a(14,2),a(14,4),a(14,7),a(14,14),tempfact(14); from fn(56):
  tempfact(14)=qsky/(1/fbwa(1,2))+1/(1/fbwa(1,4))*
  & ((qsky*fbwa(2,4)+qsky*fbwa(2,5)+qsky*fbwa(2,6))*
  & fbwb(2,4)+qsky*fbwb(2,5)+qsky*fbwb(2,6))+
  & qsky/(1/fbwa(1,5))+
  & qsky/(1/fbwa(1,6))+sig*t(53)**4/((1-emista)/emista)
c MIDDLE Wing/Tarmac :::Tarmac
c J13::a(13,2),a(13,4),a(13,7),a(13,13),tempfact(13); from fn(55):
  tempfact(13)=1/(1/fbwb(1,2))*
  & (qsky*fbwa(4,2)+qsky*fbwa(4,5)+qsky*fbwa(4,6))+
  & 1/(1/fbwb(1,4))*
  & (qsky*fbwa(2,4)+qsky*fbwa(2,5)+qsky*fbwa(2,6))+
  & qsky/(1/fbwb(1,5))+
  & qsky/(1/fbwb(1,6))+
  & sig*t(52)**4/((1-emista)/emista)
c FUSELAGE INTERIOR (CABIN)
c find the weighted average temperatures tend and tpgrf first.
  aa1=lfuselg*wfuselg
  aa2=a1
  aa3=lfuselg*hfuselg
  aa4=aa3
  aa5=a2
  atot=aa1+aa2+aa3+aa4+aa5
  cc1=aa1/atot
  cc2=aa2/atot
  cc3=aa3/atot
  cc4=aa4/atot
  cc5=aa5/atot

```

```

tend=cc1*t(35)+cc2*t(36)+cc3*t(38)+cc4*t(39)+cc5*t(61)
tpgrf=c1*t(36)+c2*t(61)
c LHS Fuselage Interior (Cabin)
c J8:::a(8,8),a(8,9),a(8,10),a(8,11),tempfact(8); from fn(47):
tempfact(8)=sig*tend**4/(1/fc45)+sig*tend**4/(1/fc46)
&
+sig*t(38)**4/((1-emisfi)/emisfi)
c RHS Fuselage Interior (Cabin)
c J10:::a(10,8),a(10,9),a(10,10),a(10,11),tempfact(10); from fn(49):
tempfact(10)=sig*tend**4/(1/fc25)+sig*tend**4/(1/fc26)
&
+sig*t(39)**4/((1-emisfi)/emisfi)
c BOTTOM (floor) Fuselage Interior (Cabin)
c J11:::a(11,8),a(11,9),a(11,10),a(11,11),tempfact(11); from fn(50):
tempfact(11)=sig*tend**4/(1/fc15)+sig*tend**4/(1/fc16)
&
+sig*t(35)**4/((1-emisfl)/emisfl)
c TOP (combined roof and plexiglass structure) Fuselage Interior (Cabin)
c J9:::a(9,8),a(9,9),a(9,10),a(9,11),tempfact(9); from fn(48):
tempfact(9)=sig*tend**4/(1/fc35)+sig*tend**4/(1/fc36)
&
+sig*tpgrf**4/((1-emispgrf)/emispgrf)
c
c now find the updated radiosity
c first replace tempfact(i) with a column vector b(i) to be
c input into lubksb
do 30 i=1,14
30 b(i)=tempfact(i)
c if(itotitr.eq.1)write(17,80)(b(i),i=1,14)
c put aa(i,j) into a(i,j); because a(i,j) has changed in the steady-state
c phase. same thing for indx(i)
do 50 i=1,14
indx(i)=iindx(i)
do 50 j=1,14
50 a(i,j)=aa(i,j)
c input n (n was 67)
n=14
call lubksb
c lubksb now returns b(i) which is the updated radiosity
c for clarity transfer b(i) to radiosj(i)
do 40 i=1,14
40 radiosj(i)=b(i)
c if(itotitr.eq.1)write(17,80)(radiosj(i),i=1,14)
do 45 i=1,14
45 temptrn(53+i)=radiosj(i)
return
end
c
subroutine hcoolcon(itotitr,l,v,h)
c purpose: determines the convective coefficient according to the flight
c speed
c called from: trnsien1 and trnsien2
c calls: none
common/c7/nperiods,envtdata(100,20)
common/c10/k,thdiffus,visckine,prandtl

```

```

    real l,llamn,nusllamn,k
    parameter(rellamn=5.e5)
c
c Variables:
c nusllamn=Nusselt number in the laminar zone
c rellamn=Reynold's number for laminar flow
c
    rel=v*l/visckine
c    if(itotitr.le.nperiods)write(12,*)'coolingneg,l,v,h',l,v,h
    if(rel.le.rellamn)then
        nusllamn=0.664*rel**(1./2)*prandtl**(1./3)
        h=k*nusllamn/l
    elseif(rel.gt.rellamn)then
        llamn=rellamn*visckine/v
        nusllamn=0.664*rellamn**(1./2)*prandtl**(1./3)
        hlamn=k*nusllamn/llamn
        hturb=0.0296*k/(l-llamn)*prandtl**(1./3)*(v/visckine)**(4./5)
&        *(5./4)*(l**(4./5)-llamn**(4./5))
        h=1./l*(llamn*hlamn+(l-llamn)*hturb)
    endif
c    if(itotitr.le.nperiods)write(12,*)'cooling,v,h',v,h
    return
end
c
c
c
    subroutine lhwng2(ithtrial,itotitr,iperiod,istep,theta,ta,hhw,
&        rfw,absorbw,emisw,eemisw,sunfact,isoloun)
c purpose: transient analysis of the left wing (the wing that is not under
c        shade)
c called from: newraph2
c calls: none
    common/c2/tt(200),ffn(200),jacob(200,200)
    common/c3/tth(100),kk(100)
    common/c6/absorbff,emisff,absorbww,emisww,absorbfl,emisfl,absorbfi
&    ,emisfi,absorbpg,emispq,absorftp,emisrftp,absorfbt,emisrftb
&    ,absorfbf,emisfb
    common/c10/aircondc,thdiffus,visckine,prandtl
    common/c14/thicknes(100),density(100),conductv(100),sheatcap(100)
    common/reflecq/qreflin(6,100),qscabsin(6,100),qreflfr(6,100),
&    qreflfrl(6,100),qscabfwr(6,100),qscabfwr(6,100),qabstwa1(6,100),
&    qabstwa2(6,100),qabstwb(6,100)
    common/temp1/t(1000)
    common/data1/delttime
    common/radiosit/jj(14)
    common/result/temp(200),temptrn(200)
    dimension newt(100),k(100),th(100),dens(100),shc(100),fn(100)
    dimension tjacob(27,27)
    real kk,k,jj,newt,jacob
    parameter(sig=.1714e-8)
    parameter(fpi=3.141593/180.)

```

```
parameter(f=0.2)
c
c Variables:
c f=fraction of an element thickness
c
c Note:
c A total of 9 layers are used to model the wing. the fifth layer is usually either
c air or fuel or a combination of both
c
  if(itotitr.eq.1.and.istep.eq.1)then
c reference thickness (th):
  th(1)=tth(1)
  th(2)=tth(2)
  th(3)=tth(3)
  th(4)=tth(4)
  th(5)=tth(29)
  th(6)=tth(30)
  th(7)=tth(5)
  th(8)=tth(6)
  th(9)=tth(7)
c reference conductivity (k):
  k(1)=kk(1)
  k(2)=kk(2)
  k(3)=kk(3)
  k(4)=kk(4)
  k(5)=kk(29)
  k(6)=kk(30)
  k(7)=kk(5)
  k(8)=kk(6)
  k(9)=kk(7)
c reference density (dens):
  dens(1)=density(1)
  dens(2)=density(2)
  dens(3)=density(3)
  dens(4)=density(4)
  dens(5)=density(5)
  dens(6)=density(6)
  dens(7)=density(7)
  dens(8)=density(8)
  dens(9)=density(9)
c reference specific heat capacity (shc):
  shc(1)=sheatcap(1)
  shc(2)=sheatcap(2)
  shc(3)=sheatcap(3)
  shc(4)=sheatcap(4)
  shc(5)=sheatcap(5)
  shc(6)=sheatcap(6)
  shc(7)=sheatcap(7)
  shc(8)=sheatcap(8)
  shc(9)=sheatcap(9)
endif
```

```

    if(istep.eq.1)then
      c reference the initial temp. (t): each layer is partitioned into 3 control volumes.
      c so the total number of temperatures = (nlayer=9)*3=27. note that after step 1, ie at
      c step 2 and beyond the steady state temperature tt(i) is replaced with newt(i). so
      c all steady state temperatures are lost. they can be retrieved, however from
      c the /result/temp(i) variable in the form of degrees Farenheit.
      t(1)=tt(1)
      t(2)=(tt(1)+tt(2))/2.
      t(3)=tt(2)
      t(4)=tt(2)
      t(5)=(tt(2)+tt(3))/2.
      t(6)=tt(3)
      t(7)=tt(3)
      t(8)=(tt(3)+tt(4))/2.
      t(9)=tt(4)
      t(10)=tt(4)
      t(11)=(tt(4)+tt(62))/2.
      t(12)=tt(62)
      t(13)=tt(62)
      t(14)=(tt(62)+tt(63))/2.
      t(15)=tt(63)
      t(16)=tt(63)
      t(17)=(tt(63)+tt(5))/2.
      t(18)=tt(5)
      t(19)=tt(5)
      t(20)=(tt(5)+tt(6))/2.
      t(21)=tt(6)
      t(22)=tt(6)
      t(23)=(tt(6)+tt(7))/2.
      t(24)=tt(7)
      t(25)=tt(7)
      t(26)=(tt(7)+tt(8))/2.
      t(27)=tt(8)
      c for the nonlinear solution of the new transient temperatures newt, one needs an
      c initial guess for the set of newt's. what better guess than the steady state
      c temperatures (as above) prior to the start of the transient analysis
      do 10 i=1,27
      10 newt(i)=t(i)
      c reference the initial radiosity (jj)
      jj(1)=tt(40)
      jj(2)=tt(41)
      return
    endif

  c
  c formulate transient cooling: develop the function ffn(i), which is the fvec
  c in the non-linear transient solution process.
  c
  c first need to replace newt(i) with the incoming tt(i), where tt(i) is
  c the solution from newraph2 subroutine for step=istep and at iteration=ithtrial.
  c note that the original tt(i), ie the steady state temperatures are all destroyed
  c after step 1, ie at step 2 and greater.

```

```

c
  if(ithtrial.eq.1)go to 65
    do 50 i=1,27
      newt(i)=tt(i)
50  continue
c
65  if(isolfoun.eq.1)then
    if(ithtrial.eq.1)then
      do 55 i=1,27
        newt(i)=tt(i)
55  continue
    endif
    do 60 i=1,27
60  t(i)=newt(i)
    temptrn(1)=t(1)-459.67
    temptrn(2)=(t(3)+t(4))/2.-459.67
    temptrn(3)=(t(6)+t(7))/2.-459.67
    temptrn(4)=(t(9)+t(10))/2.-459.67
    temptrn(5)=(t(12)+t(13))/2.-459.67
    temptrn(6)=(t(15)+t(16))/2.-459.67
    temptrn(7)=(t(18)+t(19))/2.-459.67
    temptrn(8)=(t(21)+t(22))/2.-459.67
    temptrn(9)=(t(24)+t(25))/2.-459.67
    temptrn(10)=t(27)-459.67
      return
    endif
c
c upper wing surface temp (newt1), as well as internal temp newt2 and newt3: and fn(1),
c fn(2) and fn(3).
c
  qnetsky=(sig*newt(1)**4-rfw*jj(1))/((1.-rfw*eemisw)/eemisw)
  fn(1)=-newt(1)+t(1)+delttime/(dens(1)*shc(1)*f/2*th(1))*
& ((absorbw*qreflflw(1,iperiod)*sin(fpi*theta)+
& qscabflw(1,iperiod))*sunfact+
& (absorbw*qreflflwr(1,iperiod)*sin(fpi*theta)+
& qscabflwr(1,iperiod))*(1-sunfact)
& -qnetsky-hhw*(newt(1)-ta)-
& (newt(1)-newt(2))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
  fn(2)=-newt(2)+t(2)+delttime/(dens(1)*shc(1)*(1-f)*th(1))*
& ((newt(1)-newt(2))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-
& (newt(2)-newt(3))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
  fn(3)=-newt(3)+t(3)+delttime/(dens(1)*shc(1)*f/2*th(1))*
& ((newt(2)-newt(3))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-
& (newt(3)-newt(4))/(f/4*th(1)/k(1)+f/4*th(2)/k(2)))
c
c internal nodes (all conduction): newt4-newt24: fn(4)-fn(24): layers 2 to 8.
c
  do 20 i=2,8
    ii=(i-1)*3
c
c beginning element within layer i: newt4,newt7,newt10,newt13,newt16,newt19,newt22:

```

c

```

      fn(ii+1)=
& -newt(ii+1)+t(ii+1)+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((newt(ii)-newt(ii+1))/(f/4*th(i-1)/k(i-1)+f/4*th(i)/k(i))-
& (newt(ii+1)-newt(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))

```

c

c middle element within layer i: newt5,newt8,newt11,newt14,newt17,newt20,newt23:

c

```

      fn(ii+2)=
& -newt(ii+2)+t(ii+2)+delttime/(dens(i)*shc(i)*(1-f)*th(i))*
& ((newt(ii+1)-newt(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (newt(ii+2)-newt(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))

```

c

c ending element within layer i: newt6,newt9,newt12,newt15,newt18,newt21,newt24:

c

```

      fn(ii+3)=
& -newt(ii+3)+t(ii+3)+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((newt(ii+2)-newt(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (newt(ii+3)-newt(ii+4))/(f/4*th(i)/k(i)+f/4*th(i+1)/k(i+1)))

```

20 continue

c

c lower wing surface temp newt27, as well as internal temp newt25 and newt26: fn(27),
 c fn(26) and fn(25).

c

```

      fn(25)=-newt(25)+t(25)+
& deltime/(dens(9)*shc(9)*f/2*th(9))*
& ((newt(24)-newt(25))/
& (f/4*th(8)/k(8)+f/4*th(9)/k(9))-
& (newt(25)-newt(26))/
& (f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
      fn(26)=-newt(26)+t(26)+
& deltime/(dens(9)*shc(9)*(1-f)*th(9))*
& ((newt(25)-newt(26))/
& (f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9))-
& (newt(26)-newt(27))/
& (f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
      qnetsky=(sig*newt(27)**4-jj(2))/((1.-emisw)/emisw)
      fn(27)=-newt(27)+t(27)+
& deltime/(dens(9)*shc(9)*f/2*th(9))*
& (qabstwa1(3,iperiod)-qnetsky-hhw*(newt(27)-ta)+
& (newt(26)-newt(27))/
& (f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))

```

c

c transfer the fn(i) into the global ffn(i), which is used together with jacobian
 c to find the p(i) and then the new tt(i) in newraph2 subroutine.

c

```

      do 30 i=1,27
      30 ffn(i)=fn(i)

```

c

c now find the jacobian, global jacob(120,120) from the partial derivative of fn(i).
 c find tjacob(27,27) first. then put it into the global jacob(120,120).

```

c
c first initialize tjacob(27,27) to 0.0.
c note: most of the tjacob(27,27) are 0.0
c
  do 70 itj=1,27
    do 70 jtj=1,27
      70 tjacob(itj,jtj)=0.0
c
c first find tjacob(1,),tjacob(2,),tjacob(3,) for the upper wing surface functions
c fn(1),fn(2),fn(3)
c
c tjacob(1,1),tjacob(1,2):
c
  tjacob(1,1)=-1+delttime/(dens(1)*shc(1)*f/2*th(1))*
&    (-4*sig*newt(1)**3)/((1.-rfw*eemisw)/eemisw)-hhw*(1)-
&    (1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
  tjacob(1,2)=delttime/(dens(1)*shc(1)*f/2*th(1))*
&    (-(-1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
c
c tjacob(2,1),tjacob(2,2),tjacob(2,3):
c
  tjacob(2,1)=delttime/(dens(1)*shc(1)*(1-f)*th(1))*
&    ((1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
  tjacob(2,2)=-1+delttime/(dens(1)*shc(1)*(1-f)*th(1))*
&    ((-1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-
&    (1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
  tjacob(2,3)=delttime/(dens(1)*shc(1)*(1-f)*th(1))*
&    (-(-1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
c
c tjacob(3,2),tjacob(3,3),tjacob(3,4):
c
  tjacob(3,2)=delttime/(dens(1)*shc(1)*f/2*th(1))*
&    ((1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
  tjacob(3,3)=-1+delttime/(dens(1)*shc(1)*f/2*th(1))*
&    ((-1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-
&    (1)/(f/4*th(1)/k(1)+f/4*th(2)/k(2)))
  tjacob(3,4)=delttime/(dens(1)*shc(1)*f/2*th(1))*
&    (-(-1)/(f/4*th(1)/k(1)+f/4*th(2)/k(2)))
c
c now find jacobians for the internal conduction layers: tjacob(4,) to tjacob(24,)
c for functions fn(4) to fn (24):
c
  do 120 i=2,8
    ii=(i-1)*3
c
c find tjacob(4,),tjacob(7,),tjacob(10,),tjacob(13,),tjacob(16,),tjacob(19,)
c and tjacob(22,) for beginning element within
c layer i: from fn(4),fn(7),fn(10),fn(13),fn(16),fn(19) and fn(22):
c
  tjacob(ii+1,ii)=delttime/(dens(i)*shc(i)*f/2*th(i))*
&    ((1)/(f/4*th(i-1)/k(i-1)+f/4*th(i)/k(i)))

```

```

tjacob(ii+1,ii+1)=-1+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((-1)/(f/4*th(i)/k(i-1)+f/4*th(i)/k(i))-
& (1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
tjacob(ii+1,ii+2)=delttime/(dens(i)*shc(i)*f/2*th(i))*
& (-(-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
c
c find tjacob(5,),tjacob(8,),tjacob(11,),tjacob(14,),tjacob(17,),tjacob(20,)
c and tjacob(23,) for middle element within
c layer i: from fn(5),fn(8),fn(11),fn(14),fn(17),fn(20) and fn(23):
c
tjacob(ii+2,ii+1)=delttime/(dens(i)*shc(i)*(1-f)*th(i))*
& ((1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
tjacob(ii+2,ii+2)=-1+delttime/(dens(i)*shc(i)*(1-f)*th(i))*
& ((-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
tjacob(ii+2,ii+3)=delttime/(dens(i)*shc(i)*(1-f)*th(i))*
& (-(-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
c
c find tjacob(6,),tjacob(9,),tjacob(12,),tjacob(15,),tjacob(18,),tjacob(21,)
c and tjacob(24,) for ending element within
c layer i: from fn(6),fn(9),fn(12),fn(15),fn(18),fn(21) and fn(24):
c
tjacob(ii+3,ii+2)=delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
tjacob(ii+3,ii+3)=-1+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (1)/(f/4*th(i)/k(i)+f/4*th(i+1)/k(i+1)))
tjacob(ii+3,ii+4)=delttime/(dens(i)*shc(i)*f/2*th(i))*
& (-(-1)/(f/4*th(i)/k(i)+f/4*th(i+1)/k(i+1)))
120 continue
c
c now find tjacob(25,),tjacob(26,),tjacob(27,) for the lower wing surface functions
c fn(25),fn(26) and fn(27)
c
c tjacob(25,24),tjacob(25,25) and tjacob(25,26):
c
tjacob(25,24)=delttime/(dens(9)*shc(9)*f/2*th(9))*
& ((1)/(f/4*th(8)/k(8)+f/4*th(9)/k(9)))
tjacob(25,25)=-1+delttime/(dens(9)*shc(9)*f/2*th(9))*
& ((-1)/(f/4*th(8)/k(8)+f/4*th(9)/k(9))-
& (1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
tjacob(25,26)=delttime/(dens(9)*shc(9)*f/2*th(9))*
& (-(-1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
c
c tjacob(26,25),tjacob(26,26) and tjacob(26,27):
c
tjacob(26,25)=delttime/(dens(9)*shc(9)*(1-f)*th(9))*
& ((1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
tjacob(26,26)=-1+delttime/(dens(9)*shc(9)*(1-f)*th(9))*
& ((-1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9))-
& (1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))

```

```

tjacob(26,27)=delttime/(dens(9)*shc(9)*(1-f)*th(9))*
&      (-(-1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
c
c tjacob(27,26) and tjacob(27,27):
c
tjacob(27,26)=delttime/(dens(9)*shc(9)*f/2*th(9))*
&      ((1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
tjacob(27,27)=-1+delttime/(dens(9)*shc(9)*f/2*th(9))*
&      (-(-4*sig*newt(27)**3)/((1.-emisw)/emisw)-hhw*(1)+
&      (-1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
c
c jacobian analysis is done. now put tjacob(i,j) into the global jacobian matrix
c jacob(ijacob,jjacob), where ijacob starts and ends at 1 and 27,
c and jjacob starts and ends at 1 and 27. note: most jacob values are 0's
c
do 130 i=1,27
do 130 j=1,27
jacob(i,j)=tjacob(i,j)
130 continue
return
end
c
c
subroutine rhswng2(ithtrial,itotitr,iperiod,istep,theta,ta,hhw,
&      rfw,absorbw,emisw,eemisw,sunfact,isoloun)
c purpose: transient analysis of the right wing (the wing that is under
c      shade)
c called from: newraph2
c calls: none
common/c2/tt(200),ffn(200),jacob(200,200)
common/c3/tth(100),kk(100)
common/c6/absorbff,emisff,absorbww,emisww,absorbfl,emisfl,absorbfi
&      emisfi,absorbpg,emisp,absorftp,emisrftp,absorfbt,emisrft
&      absorbfb,emisfb
common/c10/aircondc,thdiffus,visckine,prandtl
common/c14/thicknes(100),density(100),conductv(100),sheatcap(100)
common/reflecq/qreflin(6,100),qscabsin(6,100),qreflfwr(6,100),
&      qreflflwr(6,100),qscabfwr(6,100),qscabfwl(6,100),qabstwa1(6,100),
&      qabstwa2(6,100),qabstwb(6,100)
common/temp2/t(1000)
common/data1/delttime
common/radiosit/jj(14)
common/result/temp(200),temptrn(200)
dimension newt(100),k(100),th(100),dens(100),shc(100),fn(100)
dimension tjacob(27,27)
real kk,k,jj,newt,jacob
parameter(sig=.1714e-8)
parameter(fpi=3.141593/180.)
parameter(f=0.2)
c

```

c Variables:

c f=fraction of an element thickness

c

c Note:

c A total of 9 layers are used to model the wing. the fifth layer is usually either

c air or fuel or a combination of both

c

if(itotitr.eq.1.and.istep.eq.1)then

c reference thickness (th):

th(1)=tth(1)

th(2)=tth(2)

th(3)=tth(3)

th(4)=tth(4)

th(5)=tth(29)

th(6)=tth(30)

th(7)=tth(5)

th(8)=tth(6)

th(9)=tth(7)

c reference conductivity (k):

k(1)=kk(1)

k(2)=kk(2)

k(3)=kk(3)

k(4)=kk(4)

k(5)=kk(29)

k(6)=kk(30)

k(7)=kk(5)

k(8)=kk(6)

k(9)=kk(7)

c reference density (dens):

dens(1)=density(1)

dens(2)=density(2)

dens(3)=density(3)

dens(4)=density(4)

dens(5)=density(5)

dens(6)=density(6)

dens(7)=density(7)

dens(8)=density(8)

dens(9)=density(9)

c reference specific heat capacity (shc):

shc(1)=sheatcap(1)

shc(2)=sheatcap(2)

shc(3)=sheatcap(3)

shc(4)=sheatcap(4)

shc(5)=sheatcap(5)

shc(6)=sheatcap(6)

shc(7)=sheatcap(7)

shc(8)=sheatcap(8)

shc(9)=sheatcap(9)

endif

if(istep.eq.1)then

c reference the initial temp. (t): each layer is partitioned into 3 control volumes.

c so the total number of temperatures = (nlayer=9)*3=27. note that after step 1, ie at
 c step 2 and beyond the steady state temperature tt(i) is replaced with newt(i). so
 c all steady state temperatures are lost. they can be retrieved, however from
 c the /result/temp(i) variable in the form of degrees Fahrenheit.

```
t(1)=tt(9)
t(2)=(tt(9)+tt(10))/2.
t(3)=tt(10)
t(4)=tt(10)
t(5)=(tt(10)+tt(11))/2.
t(6)=tt(11)
t(7)=tt(11)
t(8)=(tt(11)+tt(12))/2.
t(9)=tt(12)
t(10)=tt(12)
t(11)=(tt(12)+tt(64))/2.
t(12)=tt(64)
t(13)=tt(64)
t(14)=(tt(64)+tt(65))/2.
t(15)=tt(65)
t(16)=tt(65)
t(17)=(tt(65)+tt(13))/2.
t(18)=tt(13)
t(19)=tt(13)
t(20)=(tt(13)+tt(14))/2.
t(21)=tt(14)
t(22)=tt(14)
t(23)=(tt(14)+tt(15))/2.
t(24)=tt(15)
t(25)=tt(15)
t(26)=(tt(15)+tt(16))/2.
t(27)=tt(16)
```

c for the nonlinear solution of the new transient temperatures newt, one needs an
 c initial guess for the set of newt's. what better guess than the steady state
 c temperatures (as above) prior to the start of the transient analysis

```
do 10 i=1,27
```

```
10 newt(i)=t(i)
```

c reference the initial radiosity (jj):

```
jj(3)=tt(42)
```

```
jj(4)=tt(43)
```

```
return
```

```
endif
```

c

c formulate transient cooling: develop the function fn(i), which is the fvec
 c in the non-linear transient solution process.

c

c first need to replace newt(i) with the incoming tt(i), where tt(i) is
 c the solution from newraph2 subroutine for step=istep and at iteration=ithtrial.
 c note that the original tt(i), ie the steady state temperatures are all destroyed
 c after step 1, ie at step 2 and greater.

c

```
if(ithtrial.eq.1)go to 65
```

```

    do 50 i=1,27
    newt(i)=tt(27+i)
50  continue
c
65  if(isolfoun.eq.1)then
    if(ithtrial.eq.1)then
        do 55 i=1,27
        newt(i)=tt(27+i)
55  continue
    endif
    do 60 i=1,27
60  t(i)=newt(i)
    temptrn(11)=t(1)-459.67
    temptrn(12)=(t(3)+t(4))/2.-459.67
    temptrn(13)=(t(6)+t(7))/2.-459.67
    temptrn(14)=(t(9)+t(10))/2.-459.67
    temptrn(15)=(t(12)+t(13))/2.-459.67
    temptrn(16)=(t(15)+t(16))/2.-459.67
    temptrn(17)=(t(18)+t(19))/2.-459.67
    temptrn(18)=(t(21)+t(22))/2.-459.67
    temptrn(19)=(t(24)+t(25))/2.-459.67
    temptrn(20)=t(27)-459.67
    return
  endif
c
c upper wing surface temp (newt1), as well as internal temp newt2 and newt3: and fn(1),
c fn(2) and fn(3).
c
    qnetsky=(sig*newt(1)**4-rfw*jj(3))/((1.-rfw*eemisw)/eemisw)
    fn(1)=-newt(1)+t(1)+delttime/(dens(1)*shc(1)*f/2*th(1))*
&      ((absorbw*qreflfr(1,iperiod)*sin(fpi*theta)+
&      qscabfwr(1,iperiod))*sunfact+
&      (absorbw*qreflfr(1,iperiod)*sin(fpi*theta)+
&      qscabfwr(1,iperiod))*(1-sunfact)
&      -qnetsky-hhw*(newt(1)-ta)-
&      (newt(1)-newt(2))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
    fn(2)=-newt(2)+t(2)+delttime/(dens(1)*shc(1)*(1-f)*th(1))*
&      ((newt(1)-newt(2))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-
&      (newt(2)-newt(3))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
    fn(3)=-newt(3)+t(3)+delttime/(dens(1)*shc(1)*f/2*th(1))*
&      ((newt(2)-newt(3))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-
&      (newt(3)-newt(4))/(f/4*th(1)/k(1)+f/4*th(2)/k(2)))
c
c internal nodes (all conduction): newt4-newt24: fn(4)-fn(24): layers 2 to 8.
c
    do 20 i=2,8
    ii=(i-1)*3
c
c beginning element within layer i: newt4,newt7,newt10,newt13,newt16,newt19,newt22:
c
    fn(ii+1)=

```

```

& -newt(ii+1)+t(ii+1)+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((newt(ii)-newt(ii+1))/(f/4*th(i-1)/k(i-1)+f/4*th(i)/k(i))-
& (newt(ii+1)-newt(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
c
c middle element within layer i: newt5,newt8,newt11,newt14,newt17,newt20,newt23:
c
  fn(ii+2)=
& -newt(ii+2)+t(ii+2)+delttime/(dens(i)*shc(i)*(1-f)*th(i))*
& ((newt(ii+1)-newt(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (newt(ii+2)-newt(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
c
c ending element within layer i: newt6,newt9,newt12,newt15,newt18,newt21,newt24:
c
  fn(ii+3)=
& -newt(ii+3)+t(ii+3)+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((newt(ii+2)-newt(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (newt(ii+3)-newt(ii+4))/(f/4*th(i)/k(i)+f/4*th(i+1)/k(i+1)))
20 continue
c
c lower wing surface temp newt27, as well as internal temp newt25 and newt26: fn(27),
c fn(26) and fn(25).
c
  fn(25)=-newt(25)+t(25)+
& deltime/(dens(9)*shc(9)*f/2*th(9))*
& ((newt(24)-newt(25))/
& (f/4*th(8)/k(8)+f/4*th(9)/k(9))-
& (newt(25)-newt(26))/
& (f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
  fn(26)=-newt(26)+t(26)+
& deltime/(dens(9)*shc(9)*(1-f)*th(9))*
& ((newt(25)-newt(26))/
& (f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9))-
& (newt(26)-newt(27))/
& (f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
  qnetsky=(sig*newt(27)**4-jj(4))/((1.-emisw)/emisw)
  fn(27)=-newt(27)+t(27)+
& deltime/(dens(9)*shc(9)*f/2*th(9))*
& (qabstwa2(3,iperiod)-qnetsky-hhw*(newt(27)-ta)+
& (newt(26)-newt(27))/
& (f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
c
c transfer the fn(i) into the global ffn(i), which is used together with jacobian
c to find the p(i) and then the new tt(i) in newraph2 subroutine.
c
  do 30 i=1,27
  30 ffn(27+i)=fn(i)
c
c now find the jacobian, global jacob(120,120) from the partial derivative of fn(i).
c find tjacob(27,27) first. then put it into the global jacob(120,120).
c
c first initialize tjacob(27,27) to 0.0.

```

c note: most of the tjacob(27,27) are 0.0

c

do 70 itj=1,27

do 70 jtj=1,27

70 tjacob(itj,jtj)=0.0

c

c first find tjacob(1,),tjacob(2,),tjacob(3,) for the upper wing surface functions

c fn(1),fn(2),fn(3)

c

c tjacob(1,1),tjacob(1,2):

c

tjacob(1,1)=-1+delttime/(dens(1)*shc(1)*f/2*th(1))*

& ((-4*sig*newt(1)**3)/((1.-rfw*eemisw)/eemisw)-hhw*(1)-

& (1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))

tjacob(1,2)=delttime/(dens(1)*shc(1)*f/2*th(1))*

& ((-1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))

c

c tjacob(2,1),tjacob(2,2),tjacob(2,3):

c

tjacob(2,1)=delttime/(dens(1)*shc(1)*(1-f)*th(1))*

& ((1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))

tjacob(2,2)=-1+delttime/(dens(1)*shc(1)*(1-f)*th(1))*

& ((-1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-

& (1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))

tjacob(2,3)=delttime/(dens(1)*shc(1)*(1-f)*th(1))*

& ((-1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))

c

c tjacob(3,2),tjacob(3,3),tjacob(3,4):

c

tjacob(3,2)=delttime/(dens(1)*shc(1)*f/2*th(1))*

& ((1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))

tjacob(3,3)=-1+delttime/(dens(1)*shc(1)*f/2*th(1))*

& ((-1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-

& (1)/(f/4*th(1)/k(1)+f/4*th(2)/k(2)))

tjacob(3,4)=delttime/(dens(1)*shc(1)*f/2*th(1))*

& ((-1)/(f/4*th(1)/k(1)+f/4*th(2)/k(2)))

c

c now find jacobians for the internal conduction layers: tjacob(4,) to tjacob(24,)

c for functions fn(4) to fn (24):

c

do 120 i=2,8

ii=(i-1)*3

c

c find tjacob(4,),tjacob(7,),tjacob(10,),tjacob(13,),tjacob(16,),tjacob(19,)

c and tjacob(22,) for beginning element within

c layer i: from fn(4),fn(7),fn(10),fn(13),fn(16),fn(19) and fn(22):

c

tjacob(ii+1,ii)=delttime/(dens(i)*shc(i)*f/2*th(i))*

& ((1)/(f/4*th(i-1)/k(i-1)+f/4*th(i)/k(i)))

tjacob(ii+1,ii+1)=-1+delttime/(dens(i)*shc(i)*f/2*th(i))*

& ((-1)/(f/4*th(i-1)/k(i-1)+f/4*th(i)/k(i))-

```

&      (1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
tjacob(ii+1,ii+2)=delttime/(dens(i)*shc(i)*f/2*th(i))*
&      (-(-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
c
c find tjacob(5,),tjacob(8,),tjacob(11,),tjacob(14,),tjacob(17,),tjacob(20,)
c and tjacob(23,) for middle element within
c layer i: from fn(5),fn(8),fn(11),fn(14),fn(17),fn(20) and fn(23):
c
tjacob(ii+2,ii+1)=delttime/(dens(i)*shc(i)*(1-f)*th(i))*
&      ((1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
tjacob(ii+2,ii+2)=-1+delttime/(dens(i)*shc(i)*(1-f)*th(i))*
&      ((-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
&      (1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
tjacob(ii+2,ii+3)=delttime/(dens(i)*shc(i)*(1-f)*th(i))*
&      (-(-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
c
c find tjacob(6,),tjacob(9,),tjacob(12,),tjacob(15,),tjacob(18,),tjacob(21,)
c and tjacob(24,) for ending element within
c layer i: from fn(6),fn(9),fn(12),fn(15),fn(18),fn(21) and fn(24):
c
tjacob(ii+3,ii+2)=delttime/(dens(i)*shc(i)*f/2*th(i))*
&      ((1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
tjacob(ii+3,ii+3)=-1+delttime/(dens(i)*shc(i)*f/2*th(i))*
&      ((-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
&      (1)/(f/4*th(i)/k(i)+f/4*th(i+1)/k(i+1)))
tjacob(ii+3,ii+4)=delttime/(dens(i)*shc(i)*f/2*th(i))*
&      (-(-1)/(f/4*th(i)/k(i)+f/4*th(i+1)/k(i+1)))
120 continue
c
c now find tjacob(25,),tjacob(26,),tjacob(27,) for the lower wing surface functions
c fn(25),fn(26) and fn(27)
c
c tjacob(25,24),tjacob(25,25) and tjacob(25,26):
c
tjacob(25,24)=delttime/(dens(9)*shc(9)*f/2*th(9))*
&      ((1)/(f/4*th(8)/k(8)+f/4*th(9)/k(9)))
tjacob(25,25)=-1+delttime/(dens(9)*shc(9)*f/2*th(9))*
&      ((-1)/(f/4*th(8)/k(8)+f/4*th(9)/k(9))-
&      (1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
tjacob(25,26)=delttime/(dens(9)*shc(9)*f/2*th(9))*
&      (-(-1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
c
c tjacob(26,25),tjacob(26,26) and tjacob(26,27):
c
tjacob(26,25)=delttime/(dens(9)*shc(9)*(1-f)*th(9))*
&      ((1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
tjacob(26,26)=-1+delttime/(dens(9)*shc(9)*(1-f)*th(9))*
&      ((-1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9))-
&      (1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))

tjacob(26,27)=delttime/(dens(9)*shc(9)*(1-f)*th(9))*

```

```

&      (-(-1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
c
c tjacob(27,26) and tjacob(27,27):
c
  tjacob(27,26)=delttime/(dens(9)*shc(9)*f/2*th(9))*
&      ((1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
  tjacob(27,27)=-1+delttime/(dens(9)*shc(9)*f/2*th(9))*
&      (-4*sig*newt(27)**3)/((1.-emisw)/emisw)-hhw+
&      (-1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
c
c jacobian analysis is done. now put tjacob(i,j) into the global jacobian matrix
c jacob(ijacob,jjacob), where ijacob starts and ends at 28 and 54,
c and jjacob starts and ends at 28 and 54. note: most jacob values are 0's
c
  do 130 i=1,27
    do 130 j=1,27
      jacob(27+i,27+j)=tjacob(i,j)
130 continue
  return
end
c
c
  subroutine fuslgr2(ithttrial,itotitr,icool,ipperiod,istep,theta,ta,
&      hhf,hhc,rff,absorbfb,eemisf,sunfact,sunfact2,isolfoun)
c purpose: transient analysis of the fuselage (left to right)
c called from: newraph2
c calls: none
  common/c2/tt(200),ffn(200),jacob(200,200)
  common/c3/tth(100),kk(100)
  common/c6/absorbff,emisff,absorbwww,emisww,absorbfl,emisfl,absorbfi
& ,emisfi,absorbpg,emisp,absorftp,emisrftp,absorfbt,emisrbt
& ,absorbfb,emisfb
  common/c10/aircondc,thdiffus,visckine,prandtl
  common/c14/thicknes(100),density(100),conductv(100),sheatcap(100)
  common/reflecq/qreflin(6,100),qscabsin(6,100),qreflfr(6,100),
& qreflfl(6,100),qscabfwr(6,100),qscabfwr(6,100),qabstwa1(6,100),
& qabstwa2(6,100),qabstwb(6,100)
  common/temp3/t(1000)
  common/data1/delttime
  common/radiosit/jj(14)
  common/result/temp(200),temptrn(200)
  dimension newt(100),k(100),th(100),dens(100),shc(100),fn(100)
  dimension tfn(100),tjacob(27,27),ttjacob(27,27)
  real kk,k,jj,newt,nusselt2,jacob
  parameter(sig=.1714e-8,gravity=32.2)
  parameter(fpi=3.141593/180.)
  parameter(f=0.2)
c
c Variables:
c f=fraction of an element thickness
c

```

c Note:

c A total of 9 layers are used to model the fuselage cabin (widthwise).

c the fifth layer is air space (maybe in turbulence). the fifth layer is

c not modelled because of its complexity. any reference to the 13,14 and 15

c control volume is only a dummy.

c

if(itotitr.eq.1.and.istep.eq.1)then

c reference thickness (th):

th(1)=tth(8)

th(2)=tth(9)

th(3)=tth(10)

th(4)=tth(23)

th(5)=tth(11)

th(6)=tth(23)

th(7)=tth(10)

th(8)=tth(9)

th(9)=tth(8)

c reference conductivity (k):

k(1)=kk(8)

k(2)=kk(9)

k(3)=kk(10)

k(4)=kk(23)

k(5)=kk(11)

k(6)=kk(23)

k(7)=kk(10)

k(8)=kk(9)

k(9)=kk(8)

c reference density (dens):

dens(1)=density(10)

dens(2)=density(11)

dens(3)=density(12)

dens(4)=density(13)

dens(5)=density(14)

dens(6)=density(13)

dens(7)=density(12)

dens(8)=density(11)

dens(9)=density(10)

c reference specific heat capacity (shc):

shc(1)=sheatcap(10)

shc(2)=sheatcap(11)

shc(3)=sheatcap(12)

shc(4)=sheatcap(13)

shc(5)=sheatcap(14)

shc(6)=sheatcap(13)

shc(7)=sheatcap(12)

shc(8)=sheatcap(11)

shc(9)=sheatcap(10)

endif

if(istep.eq.1)then

c reference the initial temp. (t): each layer is partitioned into 3 control volumes.

c so the total number of temperatures = (nlayer=9)*3=27. note that after step 1, ie at

c step 2 and beyond the steady state temperature tt(i) is replaced with newt(i). so
 c all steady state temperatures are lost. they can be retrieved, however from
 c the /result/temp(i) variable in the form of degrees Fahrenheit.

```
t(1)=tt(17)
t(2)=(tt(17)+tt(18))/2.
t(3)=tt(18)
t(4)=tt(18)
t(5)=(tt(18)+tt(19))/2.
t(6)=tt(19)
t(7)=tt(19)
t(8)=(tt(19)+tt(20))/2.
t(9)=tt(20)
t(10)=tt(20)
t(11)=(tt(20)+tt(38))/2.
t(12)=tt(38)
t(13)=tt(38)
t(14)=(tt(38)+tt(39))/2.
t(15)=tt(39)
t(16)=tt(39)
t(17)=(tt(39)+tt(21))/2.
t(18)=tt(21)
t(19)=tt(21)
t(20)=(tt(21)+tt(22))/2.
t(21)=tt(22)
t(22)=tt(22)
t(23)=(tt(22)+tt(23))/2.
t(24)=tt(23)
t(25)=tt(23)
t(26)=(tt(23)+tt(24))/2.
t(27)=tt(24)
```

c for the nonlinear solution of the new transient temperatures newt, one needs an
 c initial guess for the set of newt's. what better guess than the steady state
 c temperatures (as above) prior to the start of the transient analysis

```
do 10 i=1,27
```

```
10 newt(i)=t(i)
```

c reference the initial radiosity (jj)

```
jj(5)=tt(44)
jj(6)=tt(45)
jj(8)=tt(47)
jj(10)=tt(49)
return
endif
```

c

c formulate transient cooling: develop the function ffn(i), which is the fvec
 c in the non-linear transient solution process.

c

c first need to replace newt(i) with the incoming tt(i), where tt(i) is
 c the solution from newraph2 subroutine for step=istep and at iteration=ithtrial.
 c note that the original tt(i), ie the steady state temperatures are all destroyed
 c after step 1, ie at step 2 and greater.
 c note that the internal layer (cabin air) is not modelled due to its complexity.

c result should be ok. so there are (27-3) control volumes = 24, rearranged as c shown here and at the end of this subroutine.

c

```
if(ithtrial.eq.1)go to 65
  isum=0
  do 50 i=1,27
    if(i.eq.13.or.i.eq.14.or.i.eq.15)goto 50
    isum=isum+1
    newt(i)=tt(54+isum)
```

50 continue

c

```
65 if(isolfoun.eq.1)then
  if(ithtrial.eq.1)then
    isum=0
    do 55 i=1,27
      if(i.eq.13.or.i.eq.14.or.i.eq.15)goto 55
      isum=isum+1
      newt(i)=tt(54+isum)
```

55 continue

endif

do 60 i=1,27

60 t(i)=newt(i)

temptrn(39)=t(1)-459.67

temptrn(40)=(t(3)+t(4))/2.-459.67

temptrn(41)=(t(6)+t(7))/2.-459.67

temptrn(42)=(t(9)+t(10))/2.-459.67

temptrn(43)=t(12)-459.67

temptrn(44)=t(16)-459.67

temptrn(45)=(t(18)+t(19))/2.-459.67

temptrn(46)=(t(21)+t(22))/2.-459.67

temptrn(47)=(t(24)+t(25))/2.-459.67

temptrn(48)=t(27)-459.67

return

endif

c

c find the rayleigh and nusselt number within the cabin when it is closed (icool=2)

c and perhaps turbulence is present. these numbers are required to find the convective

c coefficient within the cabin. note that when the cabin door is opened (icool=1)

c the convective coefficient is hhc, coming from trnsien2 subroutine.

c

```
if(icool.eq.2)then
```

```
  rayleig2=gravity*1./((newt(12)+newt(16))/2.)*abs(newt(12)-newt(16))*
```

```
& th(5)**3/(thdiffus*visckine)
```

```
  if(rayleig2.le.(1e3*(0.2+prandtl)/prandtl))then
```

```
    nusselt2=1.0
```

```
  elseif(rayleig2.gt.(1e3*(0.2+prandtl)/prandtl).and.rayleig2.lt.
```

```
& 1.0e9)then
```

```
    nusselt2=0.18*((prandtl/(0.2+prandtl))*rayleig2)**0.29
```

```
  endif
```

```
endif
```

c

c formulate transient cooling:

c

c LHS surface temp of fuselage (newt1), as well as internal temp newt2 and newt3: as well

c as functions fn(1), fn(2) and fn(3).

c

```

qnetsky=(sig*newt(1)**4-rff*jj(5))/((1.-rff*eemisf)/eemisf)
fn(1)=-newt(1)+t(1)+delttime/(dens(1)*shc(1)*f/2*th(1))*
& ((absorbf*qreffwl(2,iperiod)*cos(fpi*theta)+
& qscabfwr(2,iperiod))*sunfact+
& qscabfwr(2,iperiod)*(1-sunfact)
& -qnetsky-hhf*(newt(1)-ta)-
& (newt(1)-newt(2))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
fn(2)=-newt(2)+t(2)+delttime/(dens(1)*shc(1)*(1-f)*th(1))*
& ((newt(1)-newt(2))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-
& (newt(2)-newt(3))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
fn(3)=-newt(3)+t(3)+delttime/(dens(1)*shc(1)*f/2*th(1))*
& ((newt(2)-newt(3))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-
& (newt(3)-newt(4))/(f/4*th(1)/k(1)+f/4*th(2)/k(2)))

```

c

c internal nodes::: for layers 2 to 8

c internal nodes::: beginning of control volume: newt(4,7,10,16,19,22): same as fn()

c middle of control volume: newt(5,8,11,17,20,23): same as fn()

c ending of control volume: newt(6,9,12,18,21,24): same as fn()

c newt(13,14 and 16) for cabin air neglected: ie layer 5

c

```

do 20 i=2,8
if(i.eq.5)goto 20
ii=(i-1)*3

```

c

c beginning element within layer 2,3,4,6,7,8:

c newt4,newt7,newt10,newt16,newt19,newt22: fn(4),fn(7),fn(10),fn(16),fn(19),fn(22)

c

```

if(i.eq.6)then
  qnetcab=(sig*newt(ii+1)**4-jj(10))/((1.-emisfi)/emisfi)
  if(icool.eq.1)then
    fn(ii+1)=
& -newt(ii+1)+t(ii+1)+delttime/(dens(i)*shc(i)*f/2*th(i))*
& (-newt(ii+1)-newt(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& qnetcab+absorbf*qreflin(1,iperiod)*cos(fpi*theta)+
& qscabsin(1,iperiod)-hhc*(newt(ii+1)-ta))
  elseif(icool.eq.2)then
    fn(ii+1)=
& -newt(ii+1)+t(ii+1)+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((newt(ii-3)-newt(ii+1))/(f/4*th(i-2)/k(i-2)+
& th(i-1)/(nusselt2*k(i-1))+f/4*th(i)/k(i))-
& (newt(ii+1)-newt(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& qnetcab+
& (absorbf*qreflin(1,iperiod)*cos(fpi*theta)+
& qscabsin(1,iperiod))*sunfact2)
c & (absorbf*qreflin(3,iperiod)*cos(fpi*theta)+
c & qscabsin(3,iperiod))*(1-sunfact))

```

```

    endif
  else
    fn(ii+1)=
    &   -newt(ii+1)+t(ii+1)+delttime/(dens(i)*shc(i)*f/2*th(i))*
    &   ((newt(ii)-newt(ii+1))/(f/4*th(i-1)/k(i-1)+f/4*th(i)/k(i))-
    &   (newt(ii+1)-newt(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
    endif
c
c middle element within layer 2,3,4,6,7,8:
c newt5,newt8,newt11,newt17,newt20,newt23: fn(5),fn(8),fn(11),fn(17),fn(20),fn(23)
c
    fn(ii+2)=
    &   -newt(ii+2)+t(ii+2)+delttime/(dens(i)*shc(i)*(1-f)*th(i))*
    &   ((newt(ii+1)-newt(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
    &   (newt(ii+2)-newt(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
c
c ending element within layer 2,3,4,6,7,8:
c newt6,newt9,newt12,newt18,newt21,newt24: fn(6),fn(9),fn(12),fn(18),fn(21),fn(24)
c
    if(i.eq.4)then
      qnetcab=(sig*newt(ii+3)**4-jj(8))/((1.-emisfi)/emisfi)
      if(icool.eq.1)then
        fn(ii+3)=
        &   -newt(ii+3)+t(ii+3)+delttime/(dens(i)*shc(i)*f/2*th(i))*
        &   ((newt(ii+2)-newt(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
        &   qnetcab+absorbfi*qreflin(3,iperiod)*cos(fpi*theta)+
        &   qscabsin(3,iperiod)-hhc*(newt(ii+3)-ta))
        elseif(icool.eq.2)then
          fn(ii+3)=
          &   -newt(ii+3)+t(ii+3)+delttime/(dens(i)*shc(i)*f/2*th(i))*
          &   ((newt(ii+2)-newt(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
          &   (newt(ii+3)-newt(ii+7))/(f/4*th(i)/k(i)+
          &   th(i+1)/(nusselt2*k(i+1))+f/4*th(i+2)/k(i+2))-
          &   qnetcab+
          &   (absorbfi*qreflin(3,iperiod)*cos(fpi*theta)+
          &   qscabsin(3,iperiod))*sunfact2)
c    &   (absorbfi*qreflin(1,iperiod)*cos(fpi*theta)+
c    &   qscabsin(1,iperiod))*(1-sunfact))
        endif
      else
        fn(ii+3)=
        &   -newt(ii+3)+t(ii+3)+delttime/(dens(i)*shc(i)*f/2*th(i))*
        &   ((newt(ii+2)-newt(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
        &   (newt(ii+3)-newt(ii+4))/(f/4*th(i)/k(i)+f/4*th(i+1)/k(i+1)))
        endif
20    continue
c
c RHS surface temp of fuselage (newt27), as well as internal temp
c newt25 and newt26: functions fn(27), fn(25) and fn(26).
c
    fn(25)=-newt(25)+t(25)+

```

```

&   deltime/(dens(9)*shc(9)*f/2*th(9))*
&   ((newt(24)-newt(25))/
&   (f/4*th(8)/k(8)+f/4*th(9)/k(9))-
&   (newt(25)-newt(26))/
&   (f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
  fn(26)=-newt(26)+t(26)+
&   deltime/(dens(9)*shc(9)*(1-f)*th(9))*
&   ((newt(25)-newt(26))/
&   (f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9))-
&   (newt(26)-newt(27))/
&   (f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
  qnetsky=(sig*newt(27)**4-rff*jj(6))/((1.-rff*eemisf)/eemisf)
  fn(27)=-newt(27)+t(27)+
&   deltime/(dens(9)*shc(9)*f/2*th(9))*
&   (qscabfwr(2,iperiod)*sunfact+
&   (absorbf*qreflflw(2,iperiod)*cos(fpi*theta)+
&   qscabfwr(2,iperiod))*(1-sunfact)
&   -qnetsky-hhf*(newt(27)-ta)+
&   (newt(26)-newt(27))/
&   (f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))

```

c

c transfer fn(i) into tfn(i) so that fn(13),fn(14) and fn(15) are removed
 c and thus tfn(i) begins at tfn(1) and ends at tfn(24). fn(13),fn(14) and fn(15)
 c are dummies for the layer 5 (air in the cabin).

c

```

  isum=0
  do 100 i=1,27
    if(i.eq.13.or.i.eq.14.or.i.eq.15)goto 100
    isum=isum+1
    tfn(isum)=fn(i)

```

```

100 continue

```

c

c transfer the tfn(i) into the global ffn(i), which is used together with jacobian
 c to find the p(i) and then the new tt(i) in newraph2 subroutine.

c

```

  do 30 i=1,24
  30 ffn(54+i)=tfn(i)

```

c

c now find the jacobian, global jacob(120,120) from the partial derivative of fn(i).
 c find tjacob(27,27) first. then put it into the global jacob(55..78,55..78).
 c note: tjacob(13,),tjacob(14,) and tjacob(15,) are just dummies to be removed later.

c

c first initialize tjacob(27,27) to 0.0.
 c note: most of the tjacob(27,27) are 0.0

c

```

  do 70 itj=1,27
  do 70 jtj=1,27
  70 tjacob(itj,jtj)=0.0

```

c

c first find tjacob(1,),tjacob(2,),tjacob(3,) for the LHS first layer functions
 c fn(1),fn(2),fn(3)

c

c tjacob(1,1) and tjacob(1,2):

c

```
tjacob(1,1)=-1+delttime/(dens(1)*shc(1)*f/2*th(1))*
&    (-4*sig*newt(1)**3)/((1.-rff*eemisf)/eemisf)-hhf*(1)-
&    (1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
tjacob(1,2)=delttime/(dens(1)*shc(1)*f/2*th(1))*
&    (-(-1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
```

c

c tjacob(2,1),tjacob(2,2) and tjacob(2,3):

c

```
tjacob(2,1)=delttime/(dens(1)*shc(1)*(1-f)*th(1))*
&    ((1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
tjacob(2,2)=-1+delttime/(dens(1)*shc(1)*(1-f)*th(1))*
&    ((-1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-
&    (1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
tjacob(2,3)=delttime/(dens(1)*shc(1)*(1-f)*th(1))*
&    (-(-1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
```

c

c tjacob(3,2),tjacob(3,3) and tjacob(3,4):

c

```
tjacob(3,2)=delttime/(dens(1)*shc(1)*f/2*th(1))*
&    ((1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
tjacob(3,3)=-1+delttime/(dens(1)*shc(1)*f/2*th(1))*
&    ((-1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-
&    (1)/(f/4*th(1)/k(1)+f/4*th(2)/k(2)))
tjacob(3,4)=delttime/(dens(1)*shc(1)*f/2*th(1))*
&    (-(-1)/(f/4*th(1)/k(1)+f/4*th(2)/k(2)))
```

c

c now find jacobians for the internal conduction layers: tjacob(4,) to tjacob(24,)

c for functions fn(4) to fn (24):

c

```
do 120 i=2,8
if(i.eq.5)goto 120
ii=(i-1)*3
```

c

c find tjacob(4,),tjacob(7,),tjacob(10,),tjacob(16,),tjacob(19,)

c and tjacob(22,) for beginning element within

c layer i: from fn(4),fn(7),fn(10),fn(16),fn(19) and fn(22):

c note: tjacob(13) is skipped

```
if(i.eq.6)then
  if(icool.eq.1)then
```

c tjacob(16,16),tjacob(16,17):

```
tjacob(ii+1,ii+1)=-1+delttime/(dens(i)*shc(i)*f/2*th(i))*
&    (-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
&    (4*sig*newt(ii+1)**3)/((1.-emisfi)/emisfi)-hhc*(1))
tjacob(ii+1,ii+2)=delttime/(dens(i)*shc(i)*f/2*th(i))*
&    (-(-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
elseif(icool.eq.2)then
```

c tjacob(16,12),tjacob(16,16),tjacob(16,17):

```
if(nusselt2.eq.1.0)then
```

```

tjacob(ii+1,ii-3)=delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((1)/(f/4*th(i-2)/k(i-2)+
& th(i-1)/(nusselt2*k(i-1))+f/4*th(i)/k(i)))
tjacob(ii+1,ii+1)=-1+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((-1)/(f/4*th(i-2)/k(i-2)+
& th(i-1)/(nusselt2*k(i-1))+f/4*th(i)/k(i))-
& (1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (4*sig*newt(ii+1)**3)/((1.-emisfi)/emisfi))
tjacob(ii+1,ii+2)=delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
else
const=0.18*( prandtl/(0.2+prandtl)*gravity*th(5)**3/
& (0.5*(thdiffus*visckine)) )**0.29
dndt12=const*((newt(12)+newt(16))**(-0.29)*(0.29)*
& abs(newt(12)-newt(16))**(-0.71)*(1.0)) +
& (abs(newt(12)-newt(16))**0.29*(-0.29)*
& (newt(12)+newt(16))**(-1.29)*(1.0)) )
dndt16=const*((newt(12)+newt(16))**(-0.29)*(0.29)*
& abs(newt(12)-newt(16))**(-0.71)*(-1.0)) +
& (abs(newt(12)-newt(16))**0.29*(-0.29)*
& (newt(12)+newt(16))**(-1.29)*(1.0)) )
dcdt12=th(i-1)/k(i-1)*(-1.0)*nusselt2**(-2.0)*dndt12
dcdt16=th(i-1)/k(i-1)*(-1.0)*nusselt2**(-2.0)*dndt16
abcd=(f/4*th(i-2)/k(i-2)+
& th(i-1)/(nusselt2*k(i-1))+f/4*th(i)/k(i))
dydt12=abcd**(-1.0)*(1.0) +
& (newt(12)-newt(16))*( (-1.0)*abcd**(-2.0)*dcdt12 )
dydt16=abcd**(-1.0)*(-1.0) +
& (newt(12)-newt(16))*( (-1.0)*abcd**(-2.0)*dcdt16 )
c
tjacob(ii+1,ii-3)=delttime/(dens(i)*shc(i)*f/2*th(i))*
& (dydt12)
tjacob(ii+1,ii+1)=-1+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ( dydt16-
& (1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (4*sig*newt(ii+1)**3)/((1.-emisfi)/emisfi) )
tjacob(ii+1,ii+2)=delttime/(dens(i)*shc(i)*f/2*th(i))*
& ( (-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)) )
endif
endif
else
tjacob(ii+1,ii)=delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((1)/(f/4*th(i-1)/k(i-1))+f/4*th(i)/k(i)))
tjacob(ii+1,ii+1)=-1+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((-1)/(f/4*th(i-1)/k(i-1))+f/4*th(i)/k(i))-
& (1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
tjacob(ii+1,ii+2)=delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
endif
endif
c
c find tjacob(5,),tjacob(8,),tjacob(11,),tjacob(17,),tjacob(20,)
```

c and tjacob(23,) for middle element within
 c layer i: from fn(5),fn(8),fn(11),fn(17),fn(20) and fn(23):
 c note: tjacob(14) is skipped

```
c
  tjacob(ii+2,ii+1)=deltme/(dens(i)*shc(i)*(1-f)*th(i))*
& ((1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
  tjacob(ii+2,ii+2)=-1+deltme/(dens(i)*shc(i)*(1-f)*th(i))*
& ((-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
  tjacob(ii+2,ii+3)=deltme/(dens(i)*shc(i)*(1-f)*th(i))*
& ((-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
```

c
 c find tjacob(6,),tjacob(9,),tjacob(12,),tjacob(18,),tjacob(21,)
 c and tjacob(24,) for ending element within
 c layer i: from fn(6),fn(9),fn(12),fn(18),fn(21) and fn(24):
 c note: tjacob(15) is skipped

```
c
  if(i.eq.4)then
    if(icool.eq.1)then
c tjacob(12,11),tjacob(12,12):
  tjacob(ii+3,ii+2)=deltme/(dens(i)*shc(i)*f/2*th(i))*
& ((1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
  tjacob(ii+3,ii+3)=-1+deltme/(dens(i)*shc(i)*f/2*th(i))*
& ((-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (4*sig*newt(ii+3)**3)/((1.-emisfi)/emisfi)-hhc*(1))
  elseif(icool.eq.2)then
c tjacob(12,11),tjacob(12,12), tjacob(12,16):
  if(nusselt2.eq.1.0)then
    tjacob(ii+3,ii+2)=deltme/(dens(i)*shc(i)*f/2*th(i))*
& ((1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
    tjacob(ii+3,ii+3)=-1+deltme/(dens(i)*shc(i)*f/2*th(i))*
& ((-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (1)/(f/4*th(i)/k(i)+
& th(i+1)/(nusselt2*k(i+1))+f/4*th(i+2)/k(i+2))-
& (4*sig*newt(ii+3)**3)/((1.-emisfi)/emisfi)))
    tjacob(ii+3,ii+7)=deltme/(dens(i)*shc(i)*f/2*th(i))*
& ((-1)/(f/4*th(i)/k(i)+
& th(i+1)/(nusselt2*k(i+1))+f/4*th(i+2)/k(i+2))))
  else
    const=0.18*( prandtl/(0.2+prandtl)*gravity*th(5)**3/
& (0.5*(thdiffus*visckine)))**0.29
    dndt12=const*((newt(12)+newt(16))**(-0.29)*(0.29)*
& abs(newt(12)-newt(16))**(-0.71)*(1.0)) +
& (abs(newt(12)-newt(16))**0.29*(-0.29)*
& (newt(12)+newt(16))**(-1.29)*(1.0)))
    dndt16=const*((newt(12)+newt(16))**(-0.29)*(0.29)*
& abs(newt(12)-newt(16))**(-0.71)*(-1.0)) +
& (abs(newt(12)-newt(16))**0.29*(-0.29)*
& (newt(12)+newt(16))**(-1.29)*(1.0)))
    dcdt12=th(i+1)/k(i+1)*(-1.0)*nusselt2**(-2.0)*dndt12
    dcdt16=th(i+1)/k(i+1)*(-1.0)*nusselt2**(-2.0)*dndt16
```

```

    abcd=(f/4*th(i)/k(i)+
&      th(i+1)/(nusselt2*k(i+1))+f/4*th(i+2)/k(i+2))
    dydt12=abcd**(-1.0)*(1.0) +
&      (newt(12)-newt(16))*( (-1.0)*abcd**(-2.0)*dcdt12 )
    dydt16=abcd**(-1.0)*(-1.0) +
&      (newt(12)-newt(16))*( (-1.0)*abcd**(-2.0)*dcdt16 )
c
    tjacob(ii+3,ii+2)=delttime/(dens(i)*shc(i)*f/2*th(i))*
&      ( (1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)) )
    tjacob(ii+3,ii+3)=-1+delttime/(dens(i)*shc(i)*f/2*th(i))*
&      ( (-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
&      (dydt12)-
&      (4*sig*newt(ii+3)**3)/((1.-emisfi)/emisfi) )
    tjacob(ii+3,ii+7)=delttime/(dens(i)*shc(i)*f/2*th(i))*
&      (-dydt16))
    endif
    endif
    else
    tjacob(ii+3,ii+2)=delttime/(dens(i)*shc(i)*f/2*th(i))*
&      ((1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
    tjacob(ii+3,ii+3)=-1+delttime/(dens(i)*shc(i)*f/2*th(i))*
&      ((-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
&      (1)/(f/4*th(i)/k(i)+f/4*th(i+1)/k(i+1)))
    tjacob(ii+3,ii+4)=delttime/(dens(i)*shc(i)*f/2*th(i))*
&      (-(-1)/(f/4*th(i)/k(i)+f/4*th(i+1)/k(i+1)))
    endif
120 continue
c
c
c now find tjacob(25,),tjacob(26,),tjacob(27,) for the RHS last layer surface functions
c fn(25),fn(26) and fn(27)
c
c tjacob(25,24),tjacob(25,25) and tjacob(25,26):
c
    tjacob(25,24)=delttime/(dens(9)*shc(9)*f/2*th(9))*
&      ((1)/(f/4*th(8)/k(8)+f/4*th(9)/k(9)))
    tjacob(25,25)=-1+delttime/(dens(9)*shc(9)*f/2*th(9))*
&      ((-1)/(f/4*th(8)/k(8)+f/4*th(9)/k(9))-
&      (1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
    tjacob(25,26)=delttime/(dens(9)*shc(9)*f/2*th(9))*
&      (-(-1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
c
c tjacob(26,25),tjacob(26,26) and tjacob(26,27):
c
    tjacob(26,25)=delttime/(dens(9)*shc(9)*(1-f)*th(9))*
&      ((1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
    tjacob(26,26)=-1+delttime/(dens(9)*shc(9)*(1-f)*th(9))*
&      ((-1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9))-
&      (1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
    tjacob(26,27)=delttime/(dens(9)*shc(9)*(1-f)*th(9))*
&      (-(-1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))

```

```

c
c tjacob(27,26) and tjacob(27,27): outer most control volume
c
  tjacob(27,26)=delttime/(dens(9)*shc(9)*f/2*th(9))*
&    ((1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
  tjacob(27,27)=-1+delttime/(dens(9)*shc(9)*f/2*th(9))*
&    (-4*sig*newt(27)**3)/((1.-rff*eemisf)/eemisf)-hhf*(1)+
&    (-1)/(f/4*th(9)/k(9)+(1-f)/2*th(9)/k(9)))
c
c jacobian analysis is done. now put tjacob(i,j) into the global jacobian matrix
c jacob(ijacob,jjacob), where ijacob starts and ends at 55 and 78,
c and jjacob starts and ends at 55 and 78. note: most jacob values are 0's
c
c
c first transfer jacob(27,27) into ttjacob(24,24) so that rows and columns
c 13, 14 and 15 are removed.
c
  isum1=0
  do 140 i=1,27
    if(i.eq.13.or.i.eq.14.or.i.eq.15)goto 140
    isum1=isum1+1
    isum2=0
    do 125 j=1,27
      if(j.eq.13.or.j.eq.14.or.j.eq.15)goto 125
      isum2=isum2+1
      ttjacob(isum1,isum2)=tjacob(i,j)
125  continue
140  continue
c
c now put ttjacob into the global jacob
c
  do 130 i=1,24
    do 130 j=1,24
      jacob(54+i,54+j)=ttjacob(i,j)
130  continue
  return
end
c
c
  subroutine fuslgrf2(ithttrial,itotitr,icool,iperiod,istep,theta,ta,
&    hhf,hhc,qdir,qsky,qtarm,sunfact2,isolfour)
c purpose: transient analysis of the fuselage (top to bottom)
c called from: newraph2
c calls: none
  common/c2/tt(200),ffn(200),jacob(200,200)
  common/c3/tth(100),kk(100)
  common/c4/widthwng,lengtwng,hwng,locwng,lplane,lfuselg,hfuselg,
&    wfuselg
  common/c6/absorbff,emisff,absorbwww,emisww,absorbfl,emisfl,absorbfi
&    emisfi,absorbpg,emisp,absorftp,emisrftp,absorfbt,emisrbt
&    absorbfb,emisfb

```

```
common/c5/trnsm,absorbta,emista,tarnd,densta,conducta,shcta
common/c10/aircondc,thdiffus,visckine,prandtl
common/c12/a1,a2,a3,a4,a5,a6(100),a7(100)
common/c14/thicknes(100),density(100),conductv(100),sheatcap(100)
common/reflecq/qreflin(6,100),qscabsin(6,100),qreflwr(6,100),
&qreflwl(6,100),qscabfwr(6,100),qscabfwl(6,100),qabstwa1(6,100),
&qabstwa2(6,100),qabstwb(6,100)
common/temp4/t(1000),tpg1,tpg2,tpg3
common/data1/delttime
common/radiosit/jj(14)
common/result/temp(200),temptrn(200)
dimension newt(100),k(100),th(100),dens(100),shc(100),fn(100)
dimension tfn(100),tjacob(42,42),ttjacob(42,42)
real kk,k,jj,newt,nusselt1,lfuselg,kpg,newtpg1,newtpg2,newtpg3
real jacob
parameter(sig=.1714e-8,gravity=32.2)
parameter(fpi=3.141593/180.)
parameter(f=0.2)
```

c

c Variables:

c f=fraction of an element thickness

c

c Note:

c A total of 13 layers are used to model the fuselage cabin (heightwise),

c the space between floor bottom and belly-pan top, and the belly itself.

c the fifth layer is air space (maybe in turbulence). the fifth layer is

c not modelled because of its complexity. any reference to the 13,14 and 15

c control volume is only a dummy.

c the tenth layer (or element) is air space (in between the floor bottom

c and belly-pan top). it may be filled with fuel or just air

c or both. this layer is modelled as conduction element using control volumes

c 28, 29 and 30.

if(itotitr.eq.1.and. istep.eq.1)then

c reference thickness (th):

th(1)=tth(27)

th(2)=tth(26)

th(3)=tth(25)

th(4)=tth(28)

th(5)=tth(24)

th(6)=tth(21)

th(7)=tth(20)

th(8)=tth(19)

th(9)=tth(18)

th(10)=tth(31)

th(11)=tth(14)

th(12)=tth(13)

th(13)=tth(12)

c reference conductivity (k):

k(1)=kk(27)

k(2)=kk(26)

k(3)=kk(25)

k(4)=kk(28)
k(5)=kk(24)
k(6)=kk(21)
k(7)=kk(20)
k(8)=kk(19)
k(9)=kk(18)
k(10)=kk(31)
k(11)=kk(14)
k(12)=kk(13)
k(13)=kk(12)

c reference density (dens):

dens(1)=density(15)
dens(2)=density(16)
dens(3)=density(17)
dens(4)=density(18)
dens(5)=density(19)
dens(6)=density(20)
dens(7)=density(21)
dens(8)=density(22)
dens(9)=density(23)
dens(10)=density(24)
dens(11)=density(25)
dens(12)=density(26)
dens(13)=density(27)

c reference specific heat capacity:

shc(1)=sheatcap(15)
shc(2)=sheatcap(16)
shc(3)=sheatcap(17)
shc(4)=sheatcap(18)
shc(5)=sheatcap(19)
shc(6)=sheatcap(20)
shc(7)=sheatcap(21)
shc(8)=sheatcap(22)
shc(9)=sheatcap(23)
shc(10)=sheatcap(24)
shc(11)=sheatcap(25)
shc(12)=sheatcap(26)
shc(13)=sheatcap(27)

c also reference plexiglass thickness, conductivity, density

c and specific heat capacity:

thpg=tth(22)
kpg=kk(22)
denspg=density(28)
shcpg=sheatcap(28)

c also find these values:

c1=a1/(a1+a2)
c2=a2/(a1+a2)
emispgrf=c1*emispg+c2*emisrft
endif

if(istep.eq.1)then

c reference the initial temp. (t):

```

t(1)=tt(59)
t(2)=(tt(59)+tt(58))/2.
t(3)=tt(58)
t(4)=tt(58)
t(5)=(tt(58)+tt(57))/2.
t(6)=tt(57)
t(7)=tt(57)
t(8)=(tt(57)+tt(60))/2.
t(9)=tt(60)
t(10)=tt(60)
t(11)=(tt(60)+tt(61))/2.
t(12)=tt(61)
t(13)=tt(61)
t(14)=(tt(61)+tt(35))/2.
t(15)=tt(35)
t(16)=tt(35)
t(17)=(tt(35)+tt(34))/2.
t(18)=tt(34)
t(19)=tt(34)
t(20)=(tt(34)+tt(33))/2.
t(21)=tt(33)
t(22)=tt(33)
t(23)=(tt(33)+tt(32))/2.
t(24)=tt(32)
t(25)=tt(32)
t(26)=(tt(32)+tt(31))/2.
t(27)=tt(31)
t(28)=tt(31)
t(29)=(tt(31)+tt(28))/2.
t(30)=tt(28)
t(31)=tt(28)
t(32)=(tt(28)+tt(27))/2.
t(33)=tt(27)
t(34)=tt(27)
t(35)=(tt(27)+tt(26))/2.
t(36)=tt(26)
t(37)=tt(26)
t(38)=(tt(26)+tt(25))/2.
t(39)=tt(25)

```

c also reference the initial plexiglass temp.

```

tpg1=tt(37)
tpg2=(tt(37)+tt(36))/2.
tpg3=tt(36)

```

c for the nonlinear solution of the new transient temperatures newt, one needs an
 c initial guess for the set of newt's. what better guess than the steady state
 c temperatures (as above) prior to the start of the transient analysis. also
 c assign initial guesses for newtpg's.

```

do 10 i=1,39
10 newt(i)=t(i)
newtpg1=tpg1
newtpg2=tpg2

```

```

    newtpg3=tpg3
c reference the initial radiosity
    jj(7)=tt(46)
    jj(9)=tt(48)
    jj(11)=tt(50)
    return
  endif
c
c formulate transient cooling: develop the function ffn(i), which is the fvec
c in the non-linear transient solution process.
c
c first need to replace newt(i) with the incoming tt(i), where tt(i) is
c the solution from newraph2 subroutine for step=istep and at iteration=ithtrial.
c note that the original tt(i), ie the steady state temperatures are all destroyed
c after step 1, ie at step 2 and greater.
c note that the internal layer (cabin air) is not modelled due to its complexity.
c result should be ok. so there are (39-3) control volumes = 36, plus the
c three plexiglass control volumes with temperatures tpg1,tpg2 and tpg3.
c the plexiglass fvec are placed at the end of
c fn ie: fn(40), fn(41) and fn(42).
c
  if(ithtrial.eq.1)go to 65
    isum=0
    do 50 i=1,39
      if(i.eq.13.or.i.eq.14.or.i.eq.15)goto 50
      isum=isum+1
      newt(i)=tt(78+isum)
50  continue
    newtpg1=tt(78+36+1)
    newtpg2=tt(78+36+2)
    newtpg3=tt(78+36+3)
c
65  if(isolfoun.eq.1)then
    if(ithtrial.eq.1)then
      isum=0
      do 55 i=1,39
        if(i.eq.13.or.i.eq.14.or.i.eq.15)goto 55
        isum=isum+1
        newt(i)=tt(78+isum)
55  continue
      newtpg1=tt(78+36+1)
      newtpg2=tt(78+36+2)
      newtpg3=tt(78+36+3)
    endif
    do 60 i=1,39
60   t(i)=newt(i)
      tpg1=newtpg1
      tpg2=newtpg2
      tpg3=newtpg3
      temptrn(21)=t(1)-459.67
      temptrn(22)=(t(3)+t(4))/2.-459.67

```

```

temptrn(23)=(t(6)+t(7))/2.-459.67
temptrn(24)=(t(9)+t(10))/2.-459.67
temptrn(25)=t(12)-459.67
temptrn(26)=t(16)-459.67
temptrn(27)=(t(18)+t(19))/2.-459.67
temptrn(28)=(t(21)+t(22))/2.-459.67
temptrn(29)=(t(24)+t(25))/2.-459.67
temptrn(30)=(t(27)+t(28))/2.-459.67
temptrn(31)=(t(27)+t(31))/2.-459.67
temptrn(32)=(t(27)+t(31))/2.-459.67
temptrn(33)=(t(27)+t(31))/2.-459.67
temptrn(34)=(t(27)+t(31))/2.-459.67
temptrn(35)=(t(30)+t(31))/2.-459.67
temptrn(36)=(t(33)+t(34))/2.-459.67
temptrn(37)=(t(36)+t(37))/2.-459.67
temptrn(38)=t(39)-459.67
temptrn(49)=tpg1-459.67
temptrn(50)=tpg3-459.67
return
endif

c
tpgrf=c1*newtpg3+c2*newt(12)
qnet=(sig*tpgrf**4-jj(9))/((1.-emispgrf)/emispgrf)

c
if(icool.eq.2)then
  rayleig1=gravity*1./((tpgrf+newt(16))/2.)*(newt(16)-tpgrf)*
&   th(5)**3/(thdiffus*visckine)
  if(rayleig1.le.3e5)then
    nusselt1=1.0
  elseif(rayleig1.gt.3e5.and.rayleig1.le.7e9)then
    nusselt1=0.069*rayleig1**(1./3.)*prandtl**.074
  endif
endif

c
c formulate transient cooling:
c
c first for the only three temperatures of plexiglass:
c fn(40),fn(41),fn(42): newtpg1,newtpg2,newtpg3.
c note: newtpg1=outermost plexiglas temperature
c note: newtpg3=innermost plexiglas temperature
c note: newtpg2=middle plexiglas temperature
c regard newtpg1,newtpg2 and newtpg3 as newt(40),newt(41) and newt(42) in the
c jacobian tjacob matrix.
c
c
  fn(40)=-newtpg1+tpg1+delttime/(a1*denspg*shcp*g*f/2*thpg)*
& (emisp*gsky*a1+emisp*.5*(1.-emista)*qsky*a5
&   +emisp*.5*qtarm*a5
&   +f/2*absorbpg*qdir*cos(fpi*theta)*a6(iperiod)
&   +f/2*absorbpg*qdir*sin(fpi*theta)*a7(iperiod)
&   +f/2*absorbpg*.5*(1.-absorbta)*qdir*sin(fpi*theta)*a5

```

```

& +f/2*absorbpg*qreflin(4,iperiod)*sin(fpi*theta)*a3
& +f/2*absorbpg*qscabsin(4,iperiod)*lfuselg*wfuselg
& *a3/(absorbpg*a3+absorfbt*a4)
& -emispg*sig*newtpg1**4*a1
& -hhf*(newtpg1-ta)*a1
& -(newtpg1-newtpg2)/(f/4*thpg/kpg+(1-f)/2*thpg/kpg)*a1)
  fn(41)=-newtpg2+tpg2+delttime/(a1*denspg*shcpg*(1-f)*thpg)*
& ((newtpg1-newtpg2)/(f/4*thpg/kpg+(1-f)/2*thpg/kpg)*a1
& -(newtpg2-newtpg3)/((1-f)/2*thpg/kpg+f/4*thpg/kpg)*a1
& +(1-f)*absorbpg*qdir*cos(fpi*theta)*a6(iperiod)
& +(1-f)*absorbpg*qdir*sin(fpi*theta)*a7(iperiod)
& +(1-f)*absorbpg*.5*(1.-absorbta)*qdir*sin(fpi*theta)*a5
& +(1-f)*absorbpg*qreflin(4,iperiod)*sin(fpi*theta)*a3
& +(1-f)*absorbpg*qscabsin(4,iperiod)*lfuselg*wfuselg
& *a3/(absorbpg*a3+absorfbt*a4))
  qpg=emispg*sig*newtpg3**4-emispg/(a3*emispg+a4*emisrftb)*
& (a3*emispg*sig*newtpg3**4+a4*emisrftb*sig*newt(12)**4-qnet*
& (a3+a4))
  if(icool.eq.1)then
    fn(42)=-newtpg3+tpg3+delttime/(a1*denspg*shcpg*f/2*thpg)*
& ((newtpg2-newtpg3)/((1-f)/2*thpg/kpg+f/4*thpg/kpg)*a1
& -hhc*(newtpg3-ta)*a1
& -qpg*a3
& +f/2*absorbpg*qdir*cos(fpi*theta)*a6(iperiod)
& +f/2*absorbpg*qdir*sin(fpi*theta)*a7(iperiod)
& +f/2*absorbpg*.5*(1.-absorbta)*qdir*sin(fpi*theta)*a5
& +f/2*absorbpg*qreflin(4,iperiod)*sin(fpi*theta)*a3
& +f/2*absorbpg*qscabsin(4,iperiod)*lfuselg*wfuselg
& *a3/(absorbpg*a3+absorfbt*a4))
  elseif(icool.eq.2)then
c there is a problem here. correct the tpg2-tpg3 resistance
    fn(42)=-newtpg3+tpg3+delttime/(a1*denspg*shcpg*f/2*thpg)*
& ((newtpg2-newtpg3)/((1-f)/2*thpg/kpg+f/4*thpg/kpg)*a1
& -(tpgrf-newt(16))/(th(5)/(nusselt1*k(5)))*a3
& -qpg*a3
& +f/2*absorbpg*qdir*cos(fpi*theta)*a6(iperiod)
& +f/2*absorbpg*qdir*sin(fpi*theta)*a7(iperiod)
& +f/2*absorbpg*.5*(1.-absorbta)*qdir*sin(fpi*theta)*a5
& +f/2*absorbpg*qreflin(4,iperiod)*sin(fpi*theta)*a3
& +f/2*absorbpg*qscabsin(4,iperiod)*lfuselg*wfuselg
& *a3/(absorbpg*a3+absorfbt*a4))
  endif
c
c next for the roof structure and the rest:
c
c first three temperatures of roof structure: fn(1),fn(2,fn(3):
c newt(1),newt(2),newt(3).
c
  qnetsky=(emisrftb*sig*newt(1)**4-emisrftb*qsky)
  fn(1)=-newt(1)+t(1)+delttime/(dens(1)*shc(1)*f/2*th(1))*
& (absorftb*qdir*sin(fpi*theta)-qnetsky-hhf*(newt(1)-ta)-

```

```

&      (newt(1)-newt(2))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
fn(2)=-newt(2)+t(2)+delttime/(dens(1)*shc(1)*(1-f)*th(1))*
&      ((newt(1)-newt(2))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-
&      (newt(2)-newt(3))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
fn(3)=-newt(3)+t(3)+delttime/(dens(1)*shc(1)*f/2*th(1))*
&      ((newt(2)-newt(3))/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-
&      (newt(3)-newt(4))/(f/4*th(1)/k(1)+f/4*th(2)/k(2)))
c
c internal nodes::: for layers 2 to 12: newt4 to newt36: also fn(4) to fn(36)
c where newt(13,14 and 16) and fn(13),fn(14) and fn(15) for cabin air (layer 5) neglected
c internal nodes: beginning of control volume: newt(4,7,10,16,19,22,25,28,31,34): fn()
c      middle of control volume: newt(5,8,11,17,20,23,26,29,32,35): fn()
c      ending of control volume: newt(6,9,12,18,21,24,27,30,33,36): fn()
c note: newt13,newt14 and newt15 are dummies: and so are fn(13),fn(14),fn(15).
c
      do 20 i=2,12
      if(i.eq.5)goto 20
      ii=(i-1)*3
c
c beginning element within layer 2,3,4,6,7,8,9,10,11 and 12:
c newt4,newt7,newt10,newt16,newt19,newt22,newt25,newt28,newt31,newt34:
c fn(4),fn(7),fn(10),fn(16),fn(19),fn(22),fn(25),fn(28),fn(31),fn(34):
c
      if(i.eq.6)then
        qnetcab=(sig*newt(ii+1)**4-jj(11))/((1.-emisfl)/emisfl)
        if(icool.eq.1)then
          fn(ii+1)=-newt(ii+1)+t(ii+1)+delttime/(dens(i)*shc(i)*f/2*th(i))*
&      (- (newt(ii+1)-newt(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
&      hhc*(newt(ii+1)-ta)-
&      qnetcab+absorbfl*qreflin(2,iperiod)*sin(fpi*theta)+
&      qscabsin(2,iperiod))
        elseif(icool.eq.2)then
          fn(ii+1)=-newt(ii+1)+t(ii+1)+delttime/(dens(i)*shc(i)*f/2*th(i))*
&      ((tpgrf-newt(ii+1))/(th(i-1)/(nusselt1*k(i-1))))-
&      (newt(ii+1)-newt(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
&      qnetcab+
&      (absorbfl*qreflin(2,iperiod)*sin(fpi*theta)+
&      qscabsin(2,iperiod))*sunfact2)
        endif
      else
        fn(ii+1)=-newt(ii+1)+t(ii+1)+delttime/(dens(i)*shc(i)*f/2*th(i))*
&      ((newt(ii)-newt(ii+1))/(f/4*th(i-1)/k(i-1)+f/4*th(i)/k(i))-
&      (newt(ii+1)-newt(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
      endif
c
c middle element within layer 2,3,4,6,7,8,9,10,11 and 12:
c newt5,newt8,newt11,newt17,newt20,newt23,newt26,newt29,newt32,newt35:
c fn(5),fn(8),fn(11),fn(17),fn(20),fn(23),fn(26),fn(29),fn(32),fn(35):
c
c234567812345678123456781234567812345678123456781234567812345678123456781234
5678

```

```

      fn(ii+2)=-newt(ii+2)+t(ii+2)+delttime/(dens(i)*shc(i)*(1-f)*th(i))*
& ((newt(ii+1)-newt(ii+2))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (newt(ii+2)-newt(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
c
c ending element within layer 2,3,4,6,7,8,9,10,11 and 12:
c newt6,newt9,newt12,newt18,newt21,newt24,newt27,newt30,newt33,newt36:
c fn(6),fn(9),fn(12),fn(18),fn(21),fn(24),fn(27),fn(30),fn(33),fn(36):
c
  if(i.eq.4)then
    qrfbt=emisrft*sig*newt(ii+3)**4-emisrft/(a3*emisp+g+a4*emisrft)*
& (a3*emisp*sig*newtpg3**4+a4*emisrft*sig*newt(ii+3)**4-qnet*
& (a3+a4))
    if(icool.eq.1)then
c2345678123456781234567812345678123456781234567812345678123456781234
5678
      fn(ii+3)=-newt(ii+3)+t(ii+3)+
& deltime/(a2*dens(i)*shc(i)*f/2*th(i))*
& ((newt(ii+2)-newt(ii+3))/((1-f)/2*th(i)/k(i)+f/4*th(i)/k(i))*a2-
& hhc*(newt(ii+3)-ta)*a2
& -qrfbt*a4
& +absorfbt*qreflin(4,iperiod)*sin(fpi*theta)*a4
& +absorfbt*qscabsin(4,iperiod)*lfuselg*wfuselg*a4/
& (absorbp+g*a3+absorfbt*a4))
    elseif(icool.eq.2)then
      fn(ii+3)=-newt(ii+3)+t(ii+3)+
& deltime/(a2*dens(i)*shc(i)*f/2*th(i))*
& ((newt(ii+2)-newt(ii+3))/((1-f)/2*th(i)/k(i)+f/4*th(i)/k(i))*a2-
& (tpgrf-newt(ii+7))/(th(i+1)/(nusselt1*k(i+1))))*a4
& -qrfbt*a4
& +(absorfbt*qreflin(4,iperiod)*sin(fpi*theta)*a4
& +absorfbt*qscabsin(4,iperiod)*lfuselg*wfuselg*a4/
& (absorbp+g*a3+absorfbt*a4))*sunfact2)
    endif
  else
    fn(ii+3)=-newt(ii+3)+t(ii+3)+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((newt(ii+2)-newt(ii+3))/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (newt(ii+3)-newt(ii+4))/(f/4*th(i)/k(i)+f/4*th(i+1)/k(i+1)))
    endif
20  continue
c
c belly-pan external temp (t39), as well as internal temp t37 and t38:
c
  fn(37)=-newt(37)+t(37)+
& deltime/(dens(13)*shc(13)*f/2*th(13))*
& ((newt(36)-newt(37))/
& (f/4*th(12)/k(12)+f/4*th(13)/k(13))-
& (newt(37)-newt(38))/
& (f/4*th(13)/k(13)+(1-f)/2*th(13)/k(13)))
  fn(38)=-newt(38)+t(38)+
& deltime/(dens(13)*shc(13)*(1-f)*th(13))*
& ((newt(37)-newt(38))/

```

```

&      (f/4*th(13)/k(13)+(1-f)/2*th(13)/k(13))-
&      (newt(38)-newt(39))/
&      (f/4*th(13)/k(13)+(1-f)/2*th(13)/k(13)))
      qnetsky=(sig*newt(39)**4-jj(7))/((1.-emisfb)/emisfb)
      fn(39)=-newt(39)+t(39)+
&      deltime/(dens(13)*shc(13)*f/2*th(13))*
&      (qabstwb(3,iperiod)-qnetsky-hhf*(newt(39)-ta)+
&      (newt(38)-newt(39))/
&      (f/4*th(13)/k(13)+(1-f)/2*th(13)/k(13)))
c
c transfer fn(i) into tfn(i) so that fn(13),fn(14) and fn(15) are removed
c and thus tfn(i) begins at tfn(1) and ends at tfn(39).  fn(13),fn(14) and fn(15)
c are dummies for the layer 5 (air in the cabin).  and fn(40),fn(41),fn(42)
c are equations for plexiglass
c
      isum=0
      do 100 i=1,42
        if(i.eq.13.or.i.eq.14.or.i.eq.15)goto 100
        isum=isum+1
        tfn(isum)=fn(i)
100  continue
c
c transfer the tfn(i) into the global ffn(i), which is used together with jacobian
c to find the p(i) and then the new tt(i) in newraph2 subroutine.
c
      do 30 i=1,39
        30 ffn(78+i)=tfn(i)
c
c now find the jacobian, global jacob(120,120) from the partial derivative of fn(i).
c find tjacob(42,42) first.  then put it into the global jacob(79..117,79..117).
c note: tjacob(13,),tjacob(14,) and tjacob(15,) are just dummies to be removed later.
c
c first initialize tjacob(42,42) to 0.0.
c note: most of the tjacob(42,42) are 0.0
c
      do 70 itj=1,42
        do 70 jtj=1,42
          70  tjacob(itj,jtj)=0.0
c
c first find tjacob for the only three temperatures of plexiglass:
c tjacob(40,),tjacob(41,),tjacob(42,)
c for fn(40),fn(41),fn(42): newtpg1,newtpg2,newtpg3.
c note: newtpg1=outermost plexiglas temperature
c note: newtpg3=innermost plexiglas temperature
c note: newtpg2=middle plexiglas temperature
c regard newtpg1,newtpg2 and newtpg3 as newt(40),newt(41) and newt(42) in the
c jacobian tjacob matrix.
c
c jacobians tjacob(40,40) and tjacob(40,41):
c
      tjacob(40,40)=-1+deltime/(a1*denspg*shcpg*f/2*thpg)*

```

```

& (-4*emisp*sig*newtpg1**3*a1
& -hhf*(1)*a1
& -(1)/(f/4*thpg/kpg+(1-f)/2*thpg/kpg)*a1)
  tjacob(40,41)=delttime/(a1*denspg*shcp* f/2*thpg)*
& (-(-1)/(f/4*thpg/kpg+(1-f)/2*thpg/kpg)*a1)
c
c jacobians tjacob(41,40),tjacob(41,41), tjacob(41,42):
c
  tjacob(41,40)=delttime/(a1*denspg*shcp*(1-f)*thpg)*
& ((1)/(f/4*thpg/kpg+(1-f)/2*thpg/kpg)*a1)
  tjacob(41,41)=-1+delttime/(a1*denspg*shcp*(1-f)*thpg)*
& ((-1)/(f/4*thpg/kpg+(1-f)/2*thpg/kpg)*a1)
& -(1)/((1-f)/2*thpg/kpg+f/4*thpg/kpg)*a1)
  tjacob(41,42)=delttime/(a1*denspg*shcp*(1-f)*thpg)*
& (-(-1)/((1-f)/2*thpg/kpg+f/4*thpg/kpg)*a1)
c
  if(icool.eq.1)then
c jacobians tjacob(42,12), tjacob(42,41) tjacob(42,42):
  tjacob(42,12)=delttime/(a1*denspg*shcp* f/2*thpg)*
& (-(-emisp/(a3*emisp+a4*emisrft)*
& (4*a4*emisrft*sig*newt(12)**3-(4*sig*(c1*newtpg3+
& c2*newt(12))**3*(c1*(1)))/((1.-emispgrf)/emispgrf)*(a3+a4))))*a3)
  tjacob(42,41)=delttime/(a1*denspg*shcp* f/2*thpg)*
& ((1)/((1-f)/2*thpg/kpg+f/4*thpg/kpg)*a1)
  tjacob(42,42)=-1+delttime/(a1*denspg*shcp* f/2*thpg)*
& ((-1)/((1-f)/2*thpg/kpg+f/4*thpg/kpg)*a1)
& -hhc*(1)*a1
& -(4*emisp*sig*newtpg3**3-emisp/(a3*emisp+a4*emisrft)*
& (4*a3*emisp*sig*newtpg3**3-(4*sig*(c1*newtpg3+
& c2*newt(12))**3*(c1*(1)))/((1.-emispgrf)/emispgrf)*(a3+a4))))*a3)
  elseif(icool.eq.2)then
c jacobians tjacob(42,12),tjacob(42,16),tjacob(42,41) tjacob(42,42):
  if(nusselt1.eq.1.0)then
    tjacob(42,12)=delttime/(a1*denspg*shcp* f/2*thpg)*
& ( -(c2)/(th(5)/(nusselt1*k(5)))*a3
& -(-emisp/(a3*emisp+a4*emisrft)*
& (4*a4*emisrft*sig*newt(12)**3-(4*sig*tpgrf**3*c2)/
& ((1.-emispgrf)/emispgrf)*(a3+a4))))*a3 )
    tjacob(42,16)=delttime/(a1*denspg*shcp* f/2*thpg)*
& ( -(-1)/(th(5)/(nusselt1*k(5)))*a3 )
    tjacob(42,41)=delttime/(a1*denspg*shcp* f/2*thpg)*
& ( (1)/((1-f)/2*thpg/kpg+f/4*thpg/kpg)*a1 )
    tjacob(42,42)=-1+delttime/(a1*denspg*shcp* f/2*thpg)*
& ( (-1)/((1-f)/2*thpg/kpg+f/4*thpg/kpg)*a1
& -(c1)/(th(5)/(nusselt1*k(5)))*a3
& -(4*emisp*sig*newtpg3**3-emisp/(a3*emisp+a4*emisrft)*
& (4*a3*emisp*sig*newtpg3**3-(4*sig*tpgrf**3*c1)/
& ((1.-emispgrf)/emispgrf)*(a3+a4))))*a3 )
  else
    const=0.069*prandtl**.074*(gravity*th(5)**3/
& (0.5*(thdiffus*visckine)))*(1./3.)

```

```

dndt12=const*( (tpgrf+newt(16))**(-1./3.)*
&      ((1./3.)*(newt(16)-tpgrf)**(-2./3.)*(-c2)) +
&      (newt(16)-tpgrf)**(1./3.)*
&      ((-1./3.)*(tpgrf+newt(16))**(-4./3.)*(c2)) )
dndt16=const*( (tpgrf+newt(16))**(-1./3.)*
&      ((1./3.)*(newt(16)-tpgrf)**(-2./3.)*(1)) +
&      (newt(16)-tpgrf)**(1./3.)*
&      ((-1./3.)*(tpgrf+newt(16))**(-4./3.)*(1)) )
dndt42=const*( (tpgrf+newt(16))**(-1./3.)*
&      ((1./3.)*(newt(16)-tpgrf)**(-2./3.)*(-c1)) +
&      (newt(16)-tpgrf)**(1./3.)*
&      ((-1./3.)*(tpgrf+newt(16))**(-4./3.)*(c1)) )
dcdt12=(-1.0)*(th(5)/k(5))*nusselt1**(-2.0)*dndt12
dcdt16=(-1.0)*(th(5)/k(5))*nusselt1**(-2.0)*dndt16
dcdt42=(-1.0)*(th(5)/k(5))*nusselt1**(-2.0)*dndt42
abcd=(th(5)/(nusselt1*k(5)))
dydt12=abcd**(-1.0)*(c2) +
&      (tpgrf-newt(16))*((-1.0)*abcd**(-2.0)*dcdt12)
dydt16=abcd**(-1.0)*(-1.0) +
&      (tpgrf-newt(16))*((-1.0)*abcd**(-2.0)*dcdt16)
dydt42=abcd**(-1.0)*(c1) +
&      (tpgrf-newt(16))*((-1.0)*abcd**(-2.0)*dcdt42)
c
tjacob(42,12)=delttime/(a1*denspg*shcp*g*f/2*thpg)*
&      ( -(dydt12)*a3
&      -(emisp/(a3*emisp+a4*emisrftb))*
&      (4*a4*emisrftb*sig*newt(12)**3-(4*sig*tpgrf**3*c2)/
&      ((1.-emispgrf)/emispgrf)*(a3+a4)))a3 )
tjacob(42,16)=delttime/(a1*denspg*shcp*g*f/2*thpg)*
&      ( -(dydt16)*a3 )
tjacob(42,41)=delttime/(a1*denspg*shcp*g*f/2*thpg)*
&      ( (1)/((1-f)/2*thpg/kpg+f/4*thpg/kpg)*a1 )
tjacob(42,42)=-1+delttime/(a1*denspg*shcp*g*f/2*thpg)*
&      ( (-1)/((1-f)/2*thpg/kpg+f/4*thpg/kpg)*a1
&      -(dydt42)*a3
&      -(4*emisp*sig*newtpg3**3-emisp/(a3*emisp+a4*emisrftb))*
&      (4*a3*emisp*sig*newtpg3**3-(4*sig*tpgrf**3*c1)/
&      ((1.-emispgrf)/emispgrf)*(a3+a4)))a3 )
endif
endif
c
c jacobians for the roof structure tjacob(1,),tjacob(2,),tjacob(3,):
c from the three temperatures and functions of roof structure: fn(1),fn(2,fn(3):
c newt(1),newt(2),newt(3).
c
c
c
c tjacob(1,1) and tjacob(1,2):
c
tjacob(1,1)=-1+delttime/(dens(1)*shc(1)*f/2*th(1))*
&      (-(4*emisrftb*sig*newt(1)**3)-hhf*(1)-
&      (1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))

```

```

    tjacob(1,2)=delttime/(dens(1)*shc(1)*f/2*th(1))*
    &    (-(-1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
c
c tjacob(2,1), tjacob(2,2) and tjacob(2,3):
c
    tjacob(2,1)=delttime/(dens(1)*shc(1)*(1-f)*th(1))*
    &    ((1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
    tjacob(2,2)=-1+delttime/(dens(1)*shc(1)*(1-f)*th(1))*
    &    ((-1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-
    &    (1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
    tjacob(2,3)=delttime/(dens(1)*shc(1)*(1-f)*th(1))*
    &    (-(-1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
c
c tjacob(3,2), tjacob(3,3) and tjacob(3,4):
c
    tjacob(3,2)=delttime/(dens(1)*shc(1)*f/2*th(1))*
    &    ((1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1)))
    tjacob(3,3)=-1+delttime/(dens(1)*shc(1)*f/2*th(1))*
    &    ((-1)/(f/4*th(1)/k(1)+(1-f)/2*th(1)/k(1))-
    &    (1)/(f/4*th(1)/k(1)+f/4*th(2)/k(2)))
    tjacob(3,4)=delttime/(dens(1)*shc(1)*f/2*th(1))*
    &    (-(-1)/(f/4*th(1)/k(1)+f/4*th(2)/k(2)))
c
c now find jacobians tjacob(4) to tjacob(36) within layers 2 to 12.
c note that tjacob(13,) tjacob(14,) and tjacob(15,) are skipped and are just dummies.
c
    do 120 i=2,12
    if(i.eq.5)goto 120
    ii=(i-1)*3
c
c find tjacob(4),tjacob(7),tjacob(10),tjacob(16),tjacob(19),tjacob(22)
c and tjacob(25),tjacob(28),tjacob(31) and tjacob(34) for:
c beginning element within layer 2,3,4,6,7,8,9,10,11 and 12:
c newt4,newt7,newt10,newt16,newt19,newt22,newt25,newt28,newt31,newt34:
c fn(4),fn(7),fn(10),fn(16),fn(19),fn(22),fn(25),fn(28),fn(31),fn(34):
c
    if(i.eq.6)then
    if(icool.eq.1)then
c tjacob(16,16),tjacob(16,17):
    tjacob(ii+1,ii+1)=-1+delttime/(dens(i)*shc(i)*f/2*th(i))*
    &    (-(-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
    &    hhc*(1)-
    &    (4*sig*newt(ii+1)**3)/((1.-emisfl)/emisfl))
    tjacob(ii+1,ii+2)=delttime/(dens(i)*shc(i)*f/2*th(i))*
    &    (-(-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
    elseif(icool.eq.2)then
c tjacob(16,12),tjacob(16,16),tjacob(16,17),tjacob(16,42)
    if(nusselt1.eq.1.0)then
    tjacob(ii+1,ii-3)=delttime/(dens(i)*shc(i)*f/2*th(i))*
    &    ( (c2)/(th(i-1)/(nusselt1*k(i-1))) )
    tjacob(ii+1,ii+1)=-1+delttime/(dens(i)*shc(i)*f/2*th(i))*

```

```

&      ( (-1)/(th(i-1)/(nusselt1*k(i-1)))-
&      (1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
&      (4*sig*newt(ii+1)**3)/((1.-emisfl)/emisfl) )
tjacob(ii+1,ii+2)=delttime/(dens(i)*shc(i)*f/2*th(i))*
&      ( -(-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)) )
tjacob(ii+1,42)=delttime/(dens(i)*shc(i)*f/2*th(i))*
&      ( (c1)/(th(i-1)/(nusselt1*k(i-1))) )
else
const=0.069*prandtl**.074*(gravity*th(5)**3/
&      (0.5*(thdiffus*visckine)))**(1./3.)
dndt12=const*( (tpgrf+newt(16))**(-1./3.)*
&      ((1./3.)*(newt(16)-tpgrf)**(-2./3.)*(c2)) +
&      (newt(16)-tpgrf)**(1./3.)*
&      ((-1./3.)*(tpgrf+newt(16))**(-4./3.)*(c2)) )
dndt16=const*( (tpgrf+newt(16))**(-1./3.)*
&      ((1./3.)*(newt(16)-tpgrf)**(-2./3.)*(1)) +
&      (newt(16)-tpgrf)**(1./3.)*
&      ((-1./3.)*(tpgrf+newt(16))**(-4./3.)*(1)) )
dndt42=const*( (tpgrf+newt(16))**(-1./3.)*
&      ((1./3.)*(newt(16)-tpgrf)**(-2./3.)*(c1)) +
&      (newt(16)-tpgrf)**(1./3.)*
&      ((-1./3.)*(tpgrf+newt(16))**(-4./3.)*(c1)) )
dcdt12=(-1.0)*(th(5)/k(5))*nusselt1**(-2.0)*dndt12
dcdt16=(-1.0)*(th(5)/k(5))*nusselt1**(-2.0)*dndt16
dcdt42=(-1.0)*(th(5)/k(5))*nusselt1**(-2.0)*dndt42
abcd=(th(5)/(nusselt1*k(5)))
dydt12=abcd**(-1.0)*(c2) +
&      (tpgrf-newt(16))*((-1.0)*abcd**(-2.0)*dcdt12)
dydt16=abcd**(-1.0)*(-1.0) +
&      (tpgrf-newt(16))*((-1.0)*abcd**(-2.0)*dcdt16)
dydt42=abcd**(-1.0)*(c1) +
&      (tpgrf-newt(16))*((-1.0)*abcd**(-2.0)*dcdt42)
c
tjacob(ii+1,ii-3)=delttime/(dens(i)*shc(i)*f/2*th(i))*
&      ( (dydt12) )
tjacob(ii+1,ii+1)=-1+delttime/(dens(i)*shc(i)*f/2*th(i))*
&      ( (dydt16)-
&      (1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
&      (4*sig*newt(ii+1)**3)/((1.-emisfl)/emisfl) )
tjacob(ii+1,ii+2)=delttime/(dens(i)*shc(i)*f/2*th(i))*
&      ( -(-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)) )
tjacob(ii+1,42)=delttime/(dens(i)*shc(i)*f/2*th(i))*
&      ( (dydt42) )
endif
endif
else
tjacob(ii+1,ii)=delttime/(dens(i)*shc(i)*f/2*th(i))*
&      ((1)/(f/4*th(i-1)/k(i-1)+f/4*th(i)/k(i)))
tjacob(ii+1,ii+1)=-1+delttime/(dens(i)*shc(i)*f/2*th(i))*
&      ((-1)/(f/4*th(i-1)/k(i-1)+f/4*th(i)/k(i))-
&      (1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))

```

```

    tjacob(ii+1,ii+2)=delttime/(dens(i)*shc(i)*f/2*th(i))*
    &    ((-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
    endif
c
c find tjacob(5),tjacob(8),tjacob(11),tjacob(17),tjacob(20),tjacob(23)
c and tjacob(26),tjacob(29),tjacob(32) and tjacob(35) for:
c middle element within layer 2,3,4,6,7,8,9,10,11 and 12:
c newt5,newt8,newt11,newt17,newt20,newt23,newt26,newt29,newt32,newt35:
c fn(5),fn(8),fn(11),fn(17),fn(20),fn(23),fn(26),fn(29),fn(32),fn(35):
c
    tjacob(ii+2,ii+1)=delttime/(dens(i)*shc(i)*(1-f)*th(i))*
    &    ((1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
    tjacob(ii+2,ii+2)=-1+delttime/(dens(i)*shc(i)*(1-f)*th(i))*
    &    ((-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
    &    (1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
    tjacob(ii+2,ii+3)=delttime/(dens(i)*shc(i)*(1-f)*th(i))*
    &    ((-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
c
c find tjacob(6),tjacob(9),tjacob(12),tjacob(18),tjacob(21),tjacob(24)
c and tjacob(27),tjacob(30),tjacob(33) and tjacob(36) for:
c ending element within layer 2,3,4,6,7,8,9,10,11 and 12:
c newt6,newt9,newt12,newt18,newt21,newt24,newt27,newt30,newt33,newt36:
c fn(6),fn(9),fn(12),fn(18),fn(21),fn(24),fn(27),fn(30),fn(33),fn(36):
c
    if(i.eq.4)then
        if(icool.eq.1)then
c tjacob(12,11),tjacob(12,12),tjacob(12,42)
            tjacob(ii+3,ii+2)=delttime/(a2*dens(i)*shc(i)*f/2*th(i))*
            &    ((1)/((1-f)/2*th(i)/k(i)+f/4*th(i)/k(i))*a2)
            tjacob(ii+3,ii+3)=-1+delttime/(a2*dens(i)*shc(i)*f/2*th(i))*
            &    ((-1)/((1-f)/2*th(i)/k(i)+f/4*th(i)/k(i))*a2-
            &    hhc*(1)*a2
            &    -(4*emisrftb*sig*newt(ii+3)**3-emisrftb/(a3*emispgr+a4*emisrftb))*
            &    (4*a4*emisrftb*sig*newt(ii+3)**3-(4*sig*tpgrf**3*(c2*(1))))/
            &    ((1.-emispgrf)/emispgrf)*(a3+a4))*a4)
            tjacob(ii+3,42)=delttime/(a2*dens(i)*shc(i)*f/2*th(i))*
            &    (-(-emisrftb/(a3*emispgr+a4*emisrftb))*
            &    (4*a3*emispgr*sig*newtpg3**3-(4*sig*tpgrf**3*(c1*(1))))/
            &    ((1.-emispgrf)/emispgrf)*(a3+a4))*a4)
            elseif(icool.eq.2)then
c tjacob(12,11),tjacob(12,12),tjacob(12,16),tjacob(12,42)
                if(nusselt1.eq.1.0)then
                    tjacob(ii+3,ii+2)=delttime/(a2*dens(i)*shc(i)*f/2*th(i))*
                    &    ( (1)/((1-f)/2*th(i)/k(i)+f/4*th(i)/k(i))*a2 )
                    tjacob(ii+3,ii+3)=-1+delttime/(a2*dens(i)*shc(i)*f/2*th(i))*
                    &    ( (-1)/((1-f)/2*th(i)/k(i)+f/4*th(i)/k(i))*a2-
                    &    (c2)/(th(i+1)/(nusselt1*k(i+1))))*a4
                    &    -(4*emisrftb*sig*newt(ii+3)**3-emisrftb/(a3*emispgr+a4*emisrftb))*
                    &    (4*a4*emisrftb*sig*newt(ii+3)**3-(4*sig*tpgrf**3*c2)/
                    &    ((1.-emispgrf)/emispgrf)*(a3+a4))*a4 )
                    tjacob(ii+3,ii+7)=delttime/(a2*dens(i)*shc(i)*f/2*th(i))*

```

```

&    ( -(-1)/(th(i+1)/(nusselt1*k(i+1)))3*a4 )
tjacob(ii+3,42)=delttime/(a2*dens(i)*shc(i)*f/2*th(i))*
&    ( -(c1)/(th(i+1)/(nusselt1*k(i+1)))3*a4
&    -(-emisrft/(a3*emispgr+a4*emisrft))*
&    (4*a3*emispgr*sig*newtpg33-(4*sig*tpgrf3*c1)/
&    ((1.-emispgrf)/emispgrf)*(a3+a4)))3*a4 )
else
  const=0.069*prandtl0.74*(gravity*th(5)3/
&    (0.5*(thdiffus*visckine)))1/3
  dndt12=const*( (tpgrf+newt(16))-1/3)*
&    ((1/3)*(newt(16)-tpgrf)-2/3*(c2)) +
&    (newt(16)-tpgrf)1/3*
&    ((-1/3)*(tpgrf+newt(16))-4/3*(c2)) )
  dndt16=const*( (tpgrf+newt(16))-1/3)*
&    ((1/3)*(newt(16)-tpgrf)-2/3*(1)) +
&    (newt(16)-tpgrf)1/3*
&    ((-1/3)*(tpgrf+newt(16))-4/3*(1)) )
  dndt42=const*( (tpgrf+newt(16))-1/3)*
&    ((1/3)*(newt(16)-tpgrf)-2/3*(c1)) +
&    (newt(16)-tpgrf)1/3*
&    ((-1/3)*(tpgrf+newt(16))-4/3*(c1)) )
  dcdt12=(-1.0)*(th(5)/k(5))*nusselt1-2.0*dndt12
  dcdt16=(-1.0)*(th(5)/k(5))*nusselt1-2.0*dndt16
  dcdt42=(-1.0)*(th(5)/k(5))*nusselt1-2.0*dndt42
  abcd=(th(5)/(nusselt1*k(5)))
  dydt12=abcd-1.0*(c2) +
&    (tpgrf-newt(16))*((-1.0)*abcd-2.0*dcdt12)
  dydt16=abcd-1.0*(c1) +
&    (tpgrf-newt(16))*((-1.0)*abcd-2.0*dcdt16)
  dydt42=abcd-1.0*(c1) +
&    (tpgrf-newt(16))*((-1.0)*abcd-2.0*dcdt42)
c
  tjacob(ii+3,ii+2)=delttime/(a2*dens(i)*shc(i)*f/2*th(i))*
&    ( (1)/((1-f)/2*th(i)/k(i)+f/4*th(i)/k(i))*a2 )
  tjacob(ii+3,ii+3)=-1+delttime/(a2*dens(i)*shc(i)*f/2*th(i))*
&    ( (-1)/((1-f)/2*th(i)/k(i)+f/4*th(i)/k(i))*a2-
&    (dydt12)*a4
&    -(4*emisrft*sig*newt(ii+3)3-emisrft/(a3*emispgr+a4*emisrft))*
&    (4*a4*emisrft*sig*newt(ii+3)3-(4*sig*tpgrf3*c2)/
&    ((1.-emispgrf)/emispgrf)*(a3+a4)))3*a4 )
  tjacob(ii+3,ii+7)=delttime/(a2*dens(i)*shc(i)*f/2*th(i))*
&    ( -(dydt16)*a4 )
  tjacob(ii+3,42)=delttime/(a2*dens(i)*shc(i)*f/2*th(i))*
&    ( -(dydt42)*a4
&    -(-emisrft/(a3*emispgr+a4*emisrft))*
&    (4*a3*emispgr*sig*newtpg33-(4*sig*tpgrf3*c1)/
&    ((1.-emispgrf)/emispgrf)*(a3+a4)))3*a4 )
  endif
  endif
else
  tjacob(ii+3,ii+2)=delttime/(dens(i)*shc(i)*f/2*th(i))*

```

```

& ((1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i)))
  tjacob(ii+3,ii+3)=-1+delttime/(dens(i)*shc(i)*f/2*th(i))*
& ((-1)/(f/4*th(i)/k(i)+(1-f)/2*th(i)/k(i))-
& (1)/(f/4*th(i)/k(i)+f/4*th(i+1)/k(i+1)))
  tjacob(ii+3,ii+4)=delttime/(dens(i)*shc(i)*f/2*th(i))*
& (-(-1)/(f/4*th(i)/k(i)+f/4*th(i+1)/k(i+1)))
endif
120 continue
c
c belly-pan external temp (t39), as well as internal temp t37 and t38:
c
c find tjacob(37,),tjacob(38,) and tjacob(39,) for:
c belly-pan elements within layer 13:
c newt37,newt38 and newt39:
c from fn(37),fn(38) and fn(39):
c
c
c tjacob(37,36) and tjacob(37,37) and tjacob(37,38):
c
  tjacob(37,36)=delttime/(dens(13)*shc(13)*f/2*th(13))*
& ( (1)/(f/4*th(12)/k(12)+f/4*th(13)/k(13)) )
  tjacob(37,37)=-1+delttime/(dens(13)*shc(13)*f/2*th(13))*
& ( (-1)/(f/4*th(12)/k(12)+f/4*th(13)/k(13))-
& (1)/(f/4*th(13)/k(13)+(1-f)/2*th(13)/k(13)) )
  tjacob(37,38)=delttime/(dens(13)*shc(13)*f/2*th(13))*
& ( (-1)/(f/4*th(13)/k(13)+(1-f)/2*th(13)/k(13)) )
c
c tjacob(38,37) and tjacob(38,38) and tjacob(38,39):
c
  tjacob(38,37)=delttime/(dens(13)*shc(13)*(1-f)*th(13))*
& ( (1)/(f/4*th(13)/k(13)+(1-f)/2*th(13)/k(13)) )
  tjacob(38,38)=-1+delttime/(dens(13)*shc(13)*(1-f)*th(13))*
& ( (-1)/(f/4*th(13)/k(13)+(1-f)/2*th(13)/k(13))-
& (1)/(f/4*th(13)/k(13)+(1-f)/2*th(13)/k(13)) )
  tjacob(38,39)=delttime/(dens(13)*shc(13)*(1-f)*th(13))*
& ( (-1)/(f/4*th(13)/k(13)+(1-f)/2*th(13)/k(13)) )
c
c tjacob(39,38) and tjacob(39,39):
c
  tjacob(39,38)=delttime/(dens(13)*shc(13)*f/2*th(13))*
& ( (1)/(f/4*th(13)/k(13)+(1-f)/2*th(13)/k(13)) )
  tjacob(39,39)=-1+delttime/(dens(13)*shc(13)*f/2*th(13))*
& ( -(4*sig*newt(39)**3)/((1.-emisfb)/emisfb)
& -hhf*(1)+
& (-1)/(f/4*th(13)/k(13)+(1-f)/2*th(13)/k(13)) )
c
c
c jacobian analysis is done. now put tjacob(i,j) into the global jacobian matrix
c jacob(ijacob,jjacob), where ijacob starts and ends at 79 and 117,
c and jjacob starts and ends at 79 and 117. note: most jacob values are 0's
c

```

```

c
c first transfer tjacob(42,42) into ttjacob(39,39) so that rows and columns
c 13, 14 and 15 are removed.
c
  isum1=0
  do 140 i=1,42
    if(i.eq.13.or.i.eq.14.or.i.eq.15)goto 140
    isum1=isum1+1
    isum2=0
    do 125 j=1,42
      if(j.eq.13.or.j.eq.14.or.j.eq.15)goto 125
      isum2=isum2+1
      ttjacob(isum1,isum2)=tjacob(i,j)
125  continue
140  continue
c
c now put ttjacob into the global jacob
c
  do 130 i=1,39
    do 130 j=1,39
      jacob(78+i,78+j)=ttjacob(i,j)
130 continue
  return
  end
c
c
  subroutine tarmac2(ithtrial,itotitr,icool,iperiod,istep,ta,ttarm,
    & hht,isoloun)
c purpose: transient analysis of the tarmac
c called from: newraph2
c calls: none
  common/c4/widthwng,lengtwnng,hwng,locwng,lplane,lfuselg,hfuselg,
    & wfuselg
  common/c5/trnsm,absorbta,emista,tarnd,densta,conducta,shcta
  common/c2/tt(200),ffn(200),jacob(200,200)
  common/c3/tth(100),kk(100)
  common/c14/thicknes(100),density(100),conductv(100),sheatcap(100)
  common/reflecq/qreflin(6,100),qscabsin(6,100),qreflfwr(6,100),
    &qreflflw(6,100),qscabfwr(6,100),qscabfwl(6,100),qabstwa1(6,100),
    &qabstwa2(6,100),qabstwb(6,100)
  common/temp5/t(1000)
  common/data1/delttime
  common/radiosit/jj(14)
  common/facttarm/fact
  common/result/temp(200),temptrn(200)
  dimension newt(100),fn(100)
  real jj,newt,lengtwnng,jacob
  parameter(sig=.1714e-8)
  parameter(fpi=3.141593/180.)
  parameter(afact=1.0,bfact=0.00001)
c

```

```

c Variables:
c afact=cooling when door is opened: no motion: therefore tarmac temperature
c   underneath the wing remains very constant
c bfact=a small number to force the tarmac temperature underneath the wing
c   to be the same as the unsheltered tarmac temperature: in effect when
c   the aircraft is in motion
c
  if(itotitr.eq.1.and.istep.eq.1)then
    rlwng=(lengtwng-wfuselg)/2.
  endif
  if(istep.eq.1)then
c reference the initial temp.:
    t(51)=tt(51)
    t(52)=tt(52)
    t(53)=tt(53)
c make the initial guess for the new temperature newt(i)
    newt(51)=t(51)
    newt(52)=t(52)
    newt(53)=t(53)
c reference the initial radiosity
    jj(12)=tt(54)
    jj(13)=tt(55)
    jj(14)=tt(56)
    return
  endif
  if(ithtrial.eq.1)goto 65
    newt(51)=tt(117+1)
    newt(52)=tt(117+2)
    newt(53)=tt(117+3)
65  if(isolfoun.eq.1)then
    if(ithtrial.eq.1)then
      newt(51)=tt(117+1)
      newt(52)=tt(117+2)
      newt(53)=tt(117+3)
    endif
    t(51)=newt(51)
    t(52)=newt(52)
    t(53)=newt(53)
    temprn(51)=t(51)-459.67
    temprn(52)=t(52)-459.67
    temprn(53)=t(53)-459.67
    return
  endif
c
  if(icool.eq.1)fact1=afact
  if(icool.eq.2)fact1=bfact
c
c formulate transient cooling:
c
c LHS Wing/Tarmac t(51): global temperature tt(118): global fvec ffn(118)
c

```

```

    qnetsky=(sig*newt(51)**4-jj(12))/((1.-emista)/emista)
    fn(51)=
    & -newt(51)+t(51)+delttime/(densta*rlwng*widthwng*tarmd*shcta)*
    & ((qabstwa1(1,iperiod)-qnetsky-hht*(newt(51)-ta))*rlwng*widthwng+
    & 2.*(conducta/(fact1*fact*widthwng)*(ttarm-newt(51)))*tarmd*rlwng+
    & conducta/(fact1*fact*widthwng)*(ttarm-newt(51))*tarmd*widthwng-
    & conducta/(.5*wfuselg+.5*rlwng)*(newt(51)-newt(52))*tarmd*widthwng)
c
c RHS Wing/Tarmac t(53): global temperature tt(120): global fvec ffn(120)
c
    qnetsky=(sig*newt(53)**4-jj(14))/((1.-emista)/emista)
    fn(53)=
    & -newt(53)+t(53)+delttime/(densta*rlwng*widthwng*tarmd*shcta)*
    & ((qabstwa2(1,iperiod)-qnetsky-hht*(newt(53)-ta))*rlwng*widthwng+
    & 2.*(conducta/(fact1*fact*widthwng)*(ttarm-newt(53)))*tarmd*rlwng+
    & conducta/(fact1*fact*widthwng)*(ttarm-newt(53))*tarmd*widthwng-
    & conducta/(.5*wfuselg+.5*rlwng)*(newt(53)-newt(52))*tarmd*widthwng)
c
c MIDDLE Wing/Tarmac t(52): global temperature tt(119): global fvec ffn(119)
c
    qnetsky=(sig*newt(52)**4-jj(13))/((1.-emista)/emista)
    fn(52)=
    & -newt(52)+t(52)+delttime/(densta*wfuselg*widthwng*tarmd*shcta)*
    & ((qabstwb(1,iperiod)-qnetsky-hht*(newt(52)-ta))*wfuselg*widthwng+
    & 2.*(conducta/(fact1*fact*widthwng)*(ttarm-newt(51)))*tarmd*
    & wfuselg+
    & conducta/(.5*wfuselg+.5*rlwng)*(newt(51)-newt(52))*tarmd*widthwng+
    & conducta/(.5*wfuselg+.5*rlwng)*(newt(53)-newt(52))*tarmd*widthwng)
c
c transfer fn(51),fn(52),fn(53) into the global vector ffn(118),ffn(119) and ffn(120)
c
    ffn(118)=fn(51)
    ffn(119)=fn(52)
    ffn(120)=fn(53)
c
c find the jacobians, tjacb and put it into the global jacobian(120,120)
c
c now find tjacb151 and tjacb152
c for LHS Wing/Tarmac t(51): global temperature tt(118): global fvec ffn(118)
c
    tjacb151=-1+delttime/(densta*rlwng*widthwng*tarmd*shcta)*
    & ((-4*sig*newt(51)**3)/((1.-emista)-hht*(1))*rlwng*
    & widthwng+
    & 2.*(conducta/(fact1*fact*widthwng)*(-1))*tarmd*rlwng+
    & conducta/(fact1*fact*widthwng)*(-1)*tarmd*widthwng-
    & conducta/(.5*wfuselg+.5*rlwng)*(1)*tarmd*widthwng)
    tjacb152=delttime/(densta*rlwng*widthwng*tarmd*shcta)*
    & (-conducta/(.5*wfuselg+.5*rlwng)*(-1)*tarmd*widthwng)
c

```

c now find tjacb251,tjacb252,tjacb253

c MIDDLE Wing/Tarmac t(52): global temperature tt(119): global fvec ffn(119)

c

```
tjacb251=deltme/(densta*wfuselg*widthwng*tarnd*shcta)*
& ((2.*(conducta/(fact1*fact*widthwng)*(-1)))*tarnd*wfuselg
& +conducta/(.5*wfuselg+.5*rlwng)*(1)*tarnd*widthwng)
tjacb252=-1+deltme/(densta*wfuselg*widthwng*tarnd*shcta)*
& ((-4*sig*newt(52)**3)/((1.-emista)/emista)-hht*(1))*
& wfuselg*widthwng
& +conducta/(.5*wfuselg+.5*rlwng)*(-1)*tarnd*widthwng+
& conducta/(.5*wfuselg+.5*rlwng)*(-1)*tarnd*widthwng)
tjacb253=deltme/(densta*wfuselg*widthwng*tarnd*shcta)*
& (conducta/(.5*wfuselg+.5*rlwng)*(1)*tarnd*widthwng)
```

c

c now find tjacb352 and tjacb353

c RHS Wing/Tarmac t(53): global temperature tt(120): global fvec ffn(120)

c

```
tjacb352=deltme/(densta*rlwng*widthwng*tarnd*shcta)*
& (-conducta/(.5*wfuselg+.5*rlwng)*(-1)*tarnd*widthwng)
tjacb353=-1+deltme/(densta*rlwng*widthwng*tarnd*shcta)*
& ((-4*sig*newt(53)**3)/((1.-emista)/emista)-hht*(1))*
& rlwng*widthwng+
& 2.*(conducta/(fact1*fact*widthwng)*(-1))*tarnd*rlwng+
& conducta/(fact1*fact*widthwng)*(-1)*tarnd*widthwng-
& conducta/(.5*wfuselg+.5*rlwng)*(1)*tarnd*widthwng)
```

c

c now put these jacobians into the global jacobian jacob(120,120)

c the jacobians are: for ffn(118):jacob(118,118);jacob(118,119);

c for ffn(119):jacob(119,118);jacob(119,119);jacob(119,120)

c for ffn(120):jacob(120,119);jacob(120,120)

c

```
jacob(118,118)=tjacb151
jacob(118,119)=tjacb152
jacob(119,118)=tjacb251
jacob(119,119)=tjacb252
jacob(119,120)=tjacb253
jacob(120,119)=tjacb352
jacob(120,120)=tjacb353
return
end
```

c

A.3 REFERENCES.

- A-1. Govindarajoo, N., "THERMOD, An Enhanced Thermal Model for Determining Aircraft Operational Temperatures User's Manual," FAA Report to be published.
- A-2. MSC.NASTRAN 70.5, "Quick Reference Guide," MacNeal-Schwendler Corp., 815 Colorado Boulevard, Los Angeles, CA, 1998.