NASA Contractor Report 159162

SPAR THERMAL ANALYSIS PROCESSORS
REFERENCE MANUAL, SYSTEM LEVEL 16

(NASA-CR-159162) SPAR THERMAL ANALYSIS PROCESSORS REFERENCE MANUAL, SYSTEM LEVEL 16. VOLUME 1: PROGRAM EXECUTION (Engineering Information Systems, Inc.) 262 p HC A12/MF A01 CSCL 07D G3/25

N80-14212

Unclas 46530

M. B. Marlowe, R. A. Moore, and W. D. Whetstone

ENGINEERING INFORMATION SYSTEMS, INC.
5120 Campbell Avenue, Suite 240
San Jose, California 95130

NASA Contract NAS1-14464
October 1979

NASA
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665

NASA Contractor Report 159162

SPAR THERMAL ANALYSIS PROCESSORS
REFERENCE MANUAL, SYSTEM LEVEL 16

(NASA-CR-159162) SPAR THERMAL ANALYSIS PROCESSORS REFERENCE MANUAL, SYSTEM LEVEL 16. VOLUME 1: PROGRAM EXECUTION (Engineering Information Systems, Inc.) 262 p HC A12/MF A01 CSCL 07D G3/25

N80-14212

Unclas 46530

M. B. Marlowe, R. A. Moore, and W. D. Whetstone

ENGINEERING INFORMATION SYSTEMS, INC.
5120 Campbell Avenue, Suite 240
San Jose, California 95130

NASA Contract NAS1-14464
October 1979

National Aeronautics and
Space Administration

Langley Research Center
Hampton Virginia 23665

Foreword

This report was prepared by Engineering Information Systems, Inc., under NASA Contract NAS1-14464 for the Langley Research Center of the National Aeronautics and Space Administration. The Contracting Officer's Technical Representative was Mr. J. C. Robinson.

The SPAR Thermal Analysis Reference Manual is in five Volumes:

Volume	Page Prefix	Title/Contents
1	TA1	Program Execution. Volume 1 fully defines all commands and the rules of operation of all processors.
2	TA2	Theory. Volume 2 contains discussions of element formulations and other aspects of the theoretical basis of processor functions.
3	TA3	Demonstration Problems. Computed solutions are compared with results from other sources, primarily analytical solutions.
4	TA4	Experimental Thermal Element Capability. Volume 4 describes the facilities provided to enable users to incorporate new thermal element formulations of their own design.
5	TA5	Programmer Reference. Volume 5 contains detailed information on the internal structure of the processors.

Submitted by
Engineering Information Systems, Inc.

W. D. Whetstone
President

Reference Manual

SPAR Thermal Analysis Processors

Volume 1

Program Execution

CONTENTS

Section

FIGURES

Number

Section 1

INTRODUCTION

This volume contains user instructions for performing linear and nonlinear steady state and transient thermal analyses with the SPAR thermal analysis processors TGEO, SSTA, and TRTA. It is assumed that the user is familiar with basic SPAR operatons and basic heat transfer theory. It is recommended that the user read Volume 2 of the SPAR Thermal Analysis Reference Manual (Theory) before attempting to execute the thermal analysis processors.

1.1 OVERVIEW

The SPAR Thermal Analyzer solves the heat equation

$$KT + C\dot{T} = Q + H + R \ ,$$

where

K = Kk + Kh + Kr + Km,
Kk = symmetric conduction matrix,
Kh = symmetric convection matrix,
Kr = symmetric radiation matrix,
Km = asymmetric mass-transport matrix,
C = diagonal capacity matrix,
T = temperature vector,
Q = heat source vector,
H = convection load vector,
R = radiation load vector.

To perform a steady-state or transient thermal analysis, the user must proceed according to the execution sequence described below and illustrated in Figures 1-1 and 1-2.

	See Section #
- Execute TAB/JLOC to define the position coordinates of the nodes.	V1-3.1.5
- Execute AUS/TABLE to create tables of section, material, fluid, and gas properties.	2
- Execute ELD to define all thermal elements. The thermal element repertoire is summerized on Table 1-1, page 1-3.	3, V1-3.2

Volume 1 of the SPAR Reference Manual

- Execute PLTA and PLTB to verify the correctness of the finite element mesh. Re-execute TAB/JLOC and ELD to correct any errors. V1-10

- Execute TGEO to check the element definition tables created by ELD for geometry errors. 4

- To perform steady state analyses:
  (1) Execute AUS/TABLE to create data sets which define thermal excitation quantities. 5.1
  (2) If necessary, execute TAB/JSEQ to create a joint elimination sequence data set. V1-3.1.16
  (3) If a nonlinear analysis is being started, create an initial temperature vector. 5.2.2.2
  (4) Execute SSTA to generate linear or nonlinear steady state solutions. 5.2

- To perform transient analyses:
  (1) Execute AUS/TABLE to create data sets which define time dependent excitation quantities. 6.1
  (2) Execute TRTA to generate transient solutions. 6.2

1.2 UNITS OF PHYSICAL QUANTITIES

The SI units for thermal analysis quantities are listed below.

Quantity	Description	SI Units
T	Temperature	K(degrees Kelvin)
p	Mass density	kg/m**3
c	Specific heat	J/kg K
k	Conductivity	W/m K
h	Convection coefficient	W/m**2 K
e	Emissivity	Dimensionless
V	Viscosity	N/m**2 sec
R	Gas constant	N m/kg K
Q	Source heat rate	W/m**3
F	Flux	W/m**2
P	Pressure	N/m**2
SBCON	Stefan-Boltzmann constant	W/m**2 K**4
G	Acceleration of gravity	N/kg

TABLE 1-1 Thermal Element Repertoire

Name	Type
	Conduction:
K21	2 node line element
K31	3 node area element
K41	4 node area element
K61	6 node volume element
K81	8 node volume element
	Convection to a Known Temperature:
C21	2 node line element
C31	3 node area element
C41	4 node area element
	Fluid-Surface Convective Exchange:
C32	3 node line element
C42	4 node line element
C62	6 node area element
	Mass-Transport:
MT21	2 node line element
	Integrated Mass-Transport, Convective Exchange:
MT42	4 node line element
MT62	6 node area element
	Radiation-Exchange:
R21	2 node line element
R31	3 node area element
R41	4 node area element

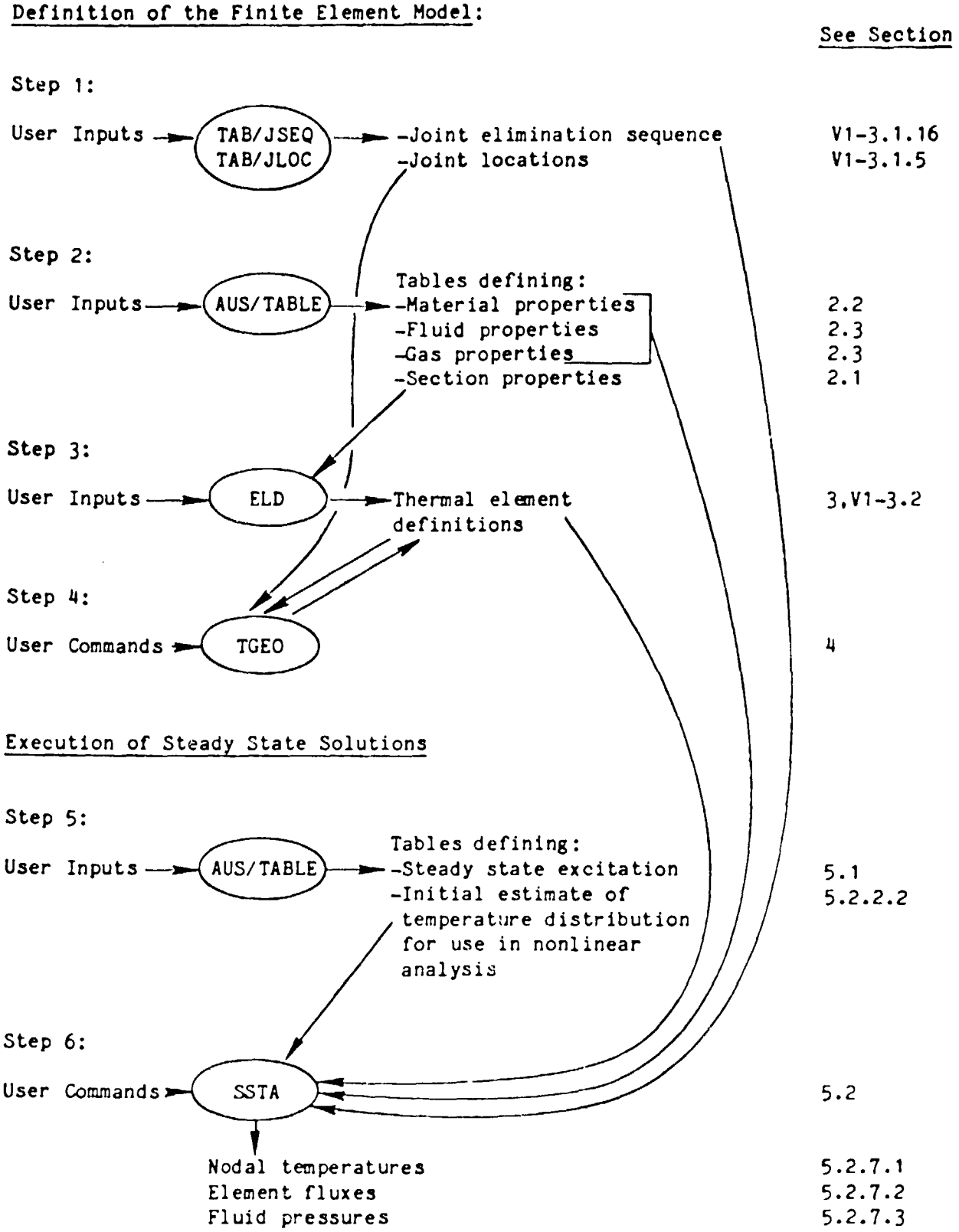


Figure 1-1 Typical Steady State Analysis Data Flow

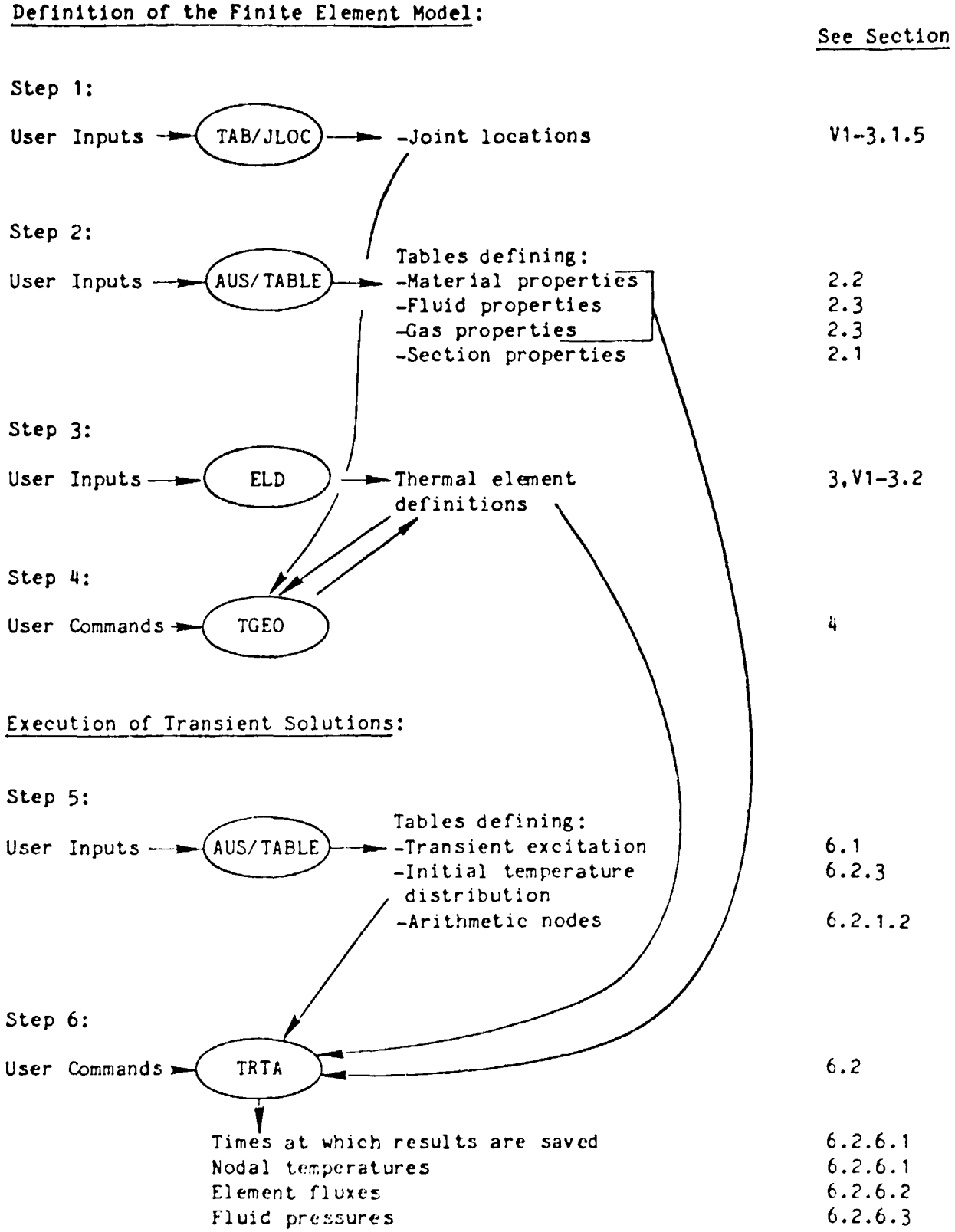


Figure 1-2 Typical Transient Analysis Data Flow

Section 2

CONSTRUCTION OF SOURCE DATA TABLES

The section, material, fluid, and gas property tables referred to by pointers during ELD execution are described in this section.

2.1 SECTION PROPERTIES

The section property tables listed below must be constructed by executing AUS/TABLE prior to executing ELD. Each line contains section properties indicated by ELD pointer NSECT (see Section 3.2). Each table may have any number of lines (NJ).

Data Set Name	NI	Line Contents	Applicable Element Types
K AREA	1	Ak	K21
K THIC	1	Tk	K31 K41
C CIRC	1	C	C21 C32 C42
C62 SECT	2	Tf Ff	C62
MT21 SECT	2	Am Dh	MT21
MT42 SECT	3	Ak C Am	MT42
MT62 SECT	5	Tf Ff Wt Wb Dh	MT62
R CIRC	1	C	R21

In the above:

Ak = conduction area.
Tk = conduction element thickness.
C = convection or radiation perimeter. The total convecting or radiating area is the perimeter times the length.
Tf = fin thickness.
Ff = fin factor. Ff multiplies the fin capacity (MT62), conduction (MT62), and convective exchange (C62,MT62) matrices. It is used to model more than one fin with only one C62 or MT62 element. The normal value of Ff is unity.
Am = flow area, used to compute fluid capacites, heat fluxes, and presure drops.
Dh = hydraulic diameter, used to compute fluid pressure drops. The hydraulic diameter is defined by Dh=4*Am/C, where Am and C are the flow area and wetted perimeter respectively. If Dh for any MT62 element is input as zero (see TGEO RESET control LZERO), it will be computed using the average fin height and the top and bottom wall widths (see the figure for element type MT62 in Section 3.1).
Wt = width of the top wall in contact with the fluid.
Wb = width of the bottom wall in contact with the fluid.

2.2 MATERIAL PROPERTIES

Conduction, convection, and radiation properties may be functions of temperature and/or time. Properties normally are defined as described in Section 2.2.1. An alternate method of defining non-temperature-dependent properties is described in Section 2.2.2.

2.2.1 TEMPERATURE-DEPENDENT MATERIAL PROPERTIES - The XXXX PROP Data Sets

The user must execute AUS/TABLE to create tables of conduction, convection, and radiation properties. The third word of the data set name is a user-assigned integer identifying the material.

Data Set Name	NI	Line Contents	Associated Element Types
COND PROP ncond	9	T p c kxx kyy kzz kxy kzy kzx	K21 K31 K41 K61 K81 C62 MT62
CONV PROP nconv	2	T h	C21 C31 C41
RADI PROP nrad	3	T e r	R21 R31 R41

The symbols used in the above data sets are:

T = temperature corresponding to property data on this line.
p = mass density.
c = specific heat.
kij = conductivities defined with respect to the coordinates of the element reference frames. All non-zero conductivities must be specified. No attempt is made to replace zeroes with default values. For example, to define isotropic conductivities for a two-dimensional analysis, both kxx and kyy must be input.
h = convection coefficient.
e = emissivity.
r = reflectivity.

These tables must be created according to the following rules:

- Temperatures must increase sequentially, line by line.
- Processors SSTA and TRTA compute properties at intermediate temperatures by linear interpolation. Extrapolation beyond the lowest or highest temperature is not allowed.
- The number of temperature points (NJ) may be different for each data set.
- If only one line is given (NJ=1), properties are constant in temperature.

Time-Dependent Properties

If any material properties are functions of time, the user must create, via AUS/TABLE, a data set named PROP TIME:

TABLE(NI=1,NJ=n):PROP TIME:J=1,n: t(1) t(2) t(3) . . . t(n)

Each table of time-dependent properties must contain n blocks, with the data in the k-th block corresponding to time t(k). The times in PROP TIME must increase sequentially. Properties at intermediate times are computed by linear interpolation. Extrapolation below t(1) or beyond t(n) is not allowed. For each table of temperature-dependent properties, the temperatures contained in the first block are used to determine propererties for all blocks.

Any single block XXXX PROP data set is understood to define properties which are constant in time.

2.2.2 NON-TEMPERATURE-DEPENDENT MATERIAL PROPERTIES - The XXXX COEF Data Sets

Conduction, convection, and radiation properties which are not functions of temperature may be input via the XXXX COEF data sets tabulated below. These data sets are specifically suited to problems involving many different materials of a particular type.

Each line contains properties of a specific material. The j-th line defines the properties of material j. There is no limit to the number of lines (NJ) in any data set.

Data Set Name	NI	Line Contents	Associated Element Types
COND COEF	8	p c kxx kyy kzz kxy kyz kzx	K21 K31 K41 K61 K81 C62 MT62
CONV COEF	1	h	C21 C31 C41
RADI COEF	2	e r	R21 R31 R41

The symbols used in the above data sets are the same as those used in the XXXX PROP data sets described in Section 2.2.1.

Time-Dependent Properties

If any material properties are functions of time, the user must create, via AUS/TABLE, a data set named COEF TIME:

TABLE(NI=1,NJ=n):COEF TIME:J=1,n: t(1) t(2) t(3) . . . t(n)

Each table of time-dependent properties must contain n blocks, with the data in the k-th block corresponding to time t(k). The times in COEF TIME must increase sequentially. Properties at intermediate times are computed by linear interpolation. Extrapolation below t(1) or beyond t(n) is not allowed.

Any single block XXXX COEF data set is understood to define properties which are constant in time.

2.2.2 NON-TEMPERATURE-DEPENDENT MATERIAL PROPERTIES - The XXXX COEF Data Sets

Conduction, convection, and radiation properties which are not functions of temperature may be input via the XXXX COEF data sets tabulated below. These data sets are specifically suited to problems involving many different materials of a particular type.

Each line contains properties of a specific material. The j-th line defines the properties of material j. There is no limit to the number of lines (NJ) in any data set.

Data Set Name	NI	Line Contents	Associated Element Types
COND COEF	8	p c kxx kyy kzz kxy kyz kzx	K21 K31 K41 K61 K81 C62 MT62
CONV COEF	1	h	C21 C31 C41
RADI COEF	2	e r	R21 R31 R41

The symbols used in the above data sets are the same as those used in the XXXX PROP data sets described in Section 2.2.1.

Time-Dependent Properties

If any material properties are functions of time, the user must create, via AUS/TABLE, a data set named COEF TIME:

TABLE(NI=1,NJ=n):COEF TIME:J=1,n: t(1) t(2) t(3) . . . t(n)

Each table of time-dependent properties must contain n blocks, with the data in the k-th block corresponding to time t(k). The times in COEF TIME must increase sequentially. Properties at intermediate times are computed by linear interpolation. Extrapolation below t(1) or beyond t(n) is not allowed.

Any single block XXXX COEF data set is understood to define properties which are constant in time.

2.3 FLUID AND GAS PROPERTIES

Fluid and gas properties may be functions of temperature and/or time. The user must execute AUS/TABLE to create the following tables, as applicable. The third word in each data set name is a user-assigned integer which identifies the fluid or gas.

Data Set Name	NI	Line Contents	Associated Element Types
FLUI PROP nflu	9	T p c k h n F m V	C32 C42 C62 MT21 MT42 MT62
GAS PROP ngas	9	T p c k h n F m R	C32 C42 C62 MT21 MT42 MT62

The symbols used in the above data sets are:

T = temperature corresponding to property data on this line.
p = mass density.
c = specific heat.
k = conductivity.
h = convection coefficient.
n = convection coefficient modification parameter:
    For gases h' = h(Tgas)*(Twall/Tgas)**n.
    For fluids h' = h(Tfluid)*(Vwall/Vfluid)**n.
F = friction factor, used to compute pressure drops.
m = friction factor modification parameter:
    For gases F' = F(Tgas)*(Twall/Tgas)**m.
    For fluids F' = F(Tfluid)*(Vwall/Vfluid)**m.
V = fluid viscosity.
R = gas constant, used to compute gas pressure drops.

These tables must be created according to the following rules:

- Temperatures must increase sequentially, line by line.
- Processors SSTA and TRTA compute properties at intermediate temperatures by linear interpolation. Extrapolation beyond the lowest or highest temperature is not allowed.
- The number of temperature points (NJ) may be different for each data set.
- If only one line is given (NJ=1), properties are constant in temperature.

Time-Dependent Properties

To define time-dependent properties, the user must create tables containing n blocks, with the data in the k-th block corresponding to time t(k) in PROP TIME as described in Section 2.2.1.

Single block FLUI PROP and GAS PROP data sets are understood to define properties which are constant in time.

Section 3

ELEMENT DEFINITION

The user must define thermal elements by executing processor ELD. The definition of thermal elements is substantially the same as for structural elements. The MOD and INC commands may be used, and all mesh generation features apply.

For element type Eij, ELD produces the same form of output data sets as for structural elements, i.e. DEF Eij, GD Eij, GTIT Eij, and DIR Eij, permitting the thermal elements to be displayed by the SPAR graphic programs.

For each element type (Eij) and group number (ngrp), ELD produces TED Eij ngrp. Each such data set contains node numbers, section properties, and material property indicators for all elements within the group.

To generate thermal element data sets, the user must execute ELD with the RESET controls described below.

Name	Default Value	Meaning
NUTED	0	Destination library for all TED Eij data sets. If NUTED is non-zero, it identifies the destination library of all TED Eij data sets. ELD will produce TED Eij data sets for all thermal element DEF Eij data sets in library 1, not just those created during the current execution. If NUTED is zero, no TED Eij data sets will be produced.
LRTED	896	Nominal block length of all TED Eij data sets.

Thermal elements are described in Section 3.1. Pointer commands for specifying section, material, fluid, and gas properties are tabulated in Section 3.2. An example ELD runstream is shown in Section 3.3.

3.1 ELEMENT REPERTOIRE

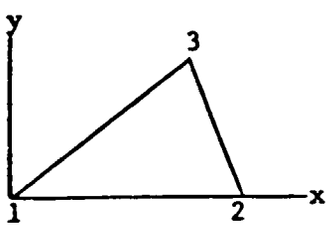
In the following tabulation, each thermal element is described and its associated data sets (section properties, thermal loading, etc.) are identified.

K21 - 2 node, line conduction element. This element represents heat conduction in the x direction and source heat generation. A point conductor and/or source is defined by specifying the same node number at each end or via TGEO execution command ZEROL. The length of a point conductor is set equal to unity.

Associated Data Sets:

Conduction area	- K	AREA
Material properties	- COND	PROP
	- COND	COEF
Source heat rate	- SOUR	K21

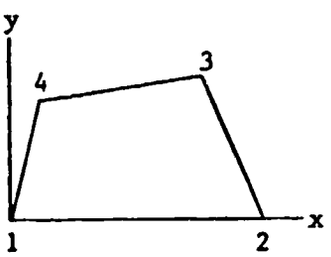
K31 - 3 node, area conduction element. This element represents two-dimensional heat conduction and source heat generation.



Associated Data Sets:

Element thickness	- K	THIC
Material properties	- COND	PROP
	- COND	COEF
Source heat rate	- SOUR	K31

K41 - 4 node, area conduction element. This element represents two-dimensional heat conduction and source heat generation. It should be very nearly flat.

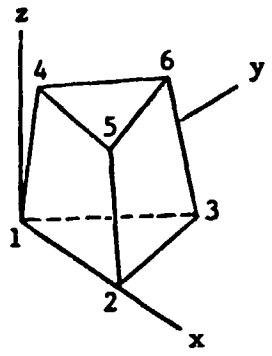


Associated Data Sets:

Element thickness	- K	THIC
Material properties	- COND	PROP
	- COND	COEF
Source heat rate	- SOUR	K41

3.1 ELEMENT REPERTOIRE (continued)

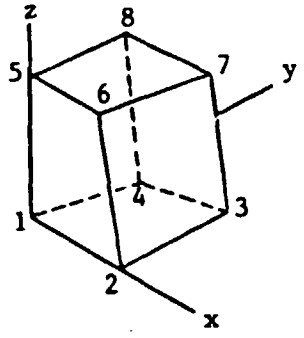
K61 - 6 node, volume conduction element. This element represents three-dimensional heat conduction and source heat generation. All faces should be very nearly flat. Nodes 4-6 must lie above the x-y plane.



Associated Data Sets:

Material properties	- COND PROP
	- COND COEF
Source heat rate	- SOUR K61

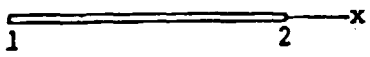
K81 - 8 node, volume conduction element. This element represents three-dimensional heat conduction and source heat generation. All faces should be very nearly flat. Nodes 5-8 must lie above the x-y plane.



Associated Data Sets:

Material properties	- COND PROP
	- COND COEF
Source heat rate	- SOUR K81

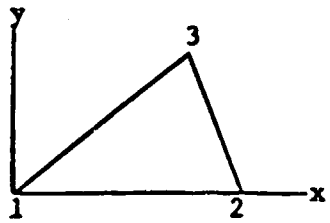
C21 - 2 node, line convection element. This element convects to a known exchange temperature. The convection surface is defined by nodes 1-2 and a width or perimeter. A point convector is defined by specifying the same node number at each end or via TGEO execution command ZEROL. The length of a point convector is set equal to unity.



Associated Data Sets:

Convection perimeter	- C CIRC
Material properties	- CONV PROP
	- CONV COEF
Exchange temperature	- CTEM C21

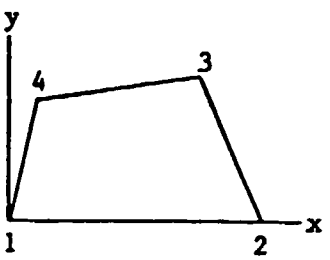
C31 - 3 node, area convection element. This element convects to a known exchange temperature.



Associated Data Sets:

Material properties	- CONV PROP
	- CONV COEF
Exchange temperature	- CTEM C31

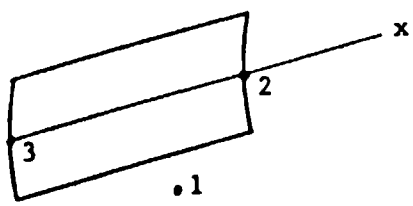
C41 - 4 node, area convection element. This element convects to a known exchange temperature. It should be very nearly flat.



Associated Data Sets:

Material properties	- CONV PROP
	- CONV COEF
Exchange temperature	- CTEM C41

C32 - 3 node, line convection element. This element represents heat transfer between a surface and a fluid or gas. The convection surface is defined by nodes 2-3 and a width or perimeter. The location of fluid node 1 is arbitrary. The fluid temperature is asssumed to be constant in the x direction.

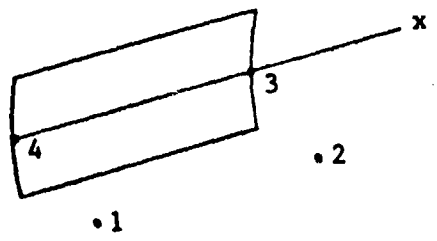


Associated Data Sets:

Convection perimeter	- C CIRC
Fluid/gas properties	- FLUI PROP
	- GAS PROP

3.1 ELEMENT REPERTOIRE (continued)

C42 - 4 node, line convection element. This element represents heat transfer between a surface and a fluid or gas. The convection surface is defined by nodes 3-4 and a width or perimeter. Fluid nodes 1 and 2 are assumed to lie within the planes normal to the x axis at surface nodes 4 and 3 respectively.



Associated Data Sets:

Convection perimeter	- C	CIRC
Fluid/gas properties	- FLUI	PROP
	- GAS	PROP

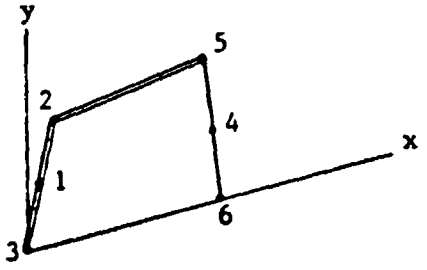
C62 - 6 node, area convection element. This element represents a fin which exchanges heat from both sides with a surrounding fluid or gas. The fin is defined by nodes 3-6-5-2 and should be very nearly flat. For computational purposes, it is assumed that fluid nodes 1 and 4 lie at the midpoints of the lines connecting fin nodes 2-3 and 5-6 respectively.

The average heat flux leaving the fin is given by q=eff*h*(T-Tf), where eff is the fin efficiency, h is the fluid convection coefficient, T is the average fin temperature, and Tf is the average temperature of the fluid in contact with the fin. The fin efficiency normally is unity; however the fin efficiency can be computed from the following formula (see SSTA and TRTA RESET control FINEFF):

$$eff = 2*(\cosh(c)-1)/(c*\sinh(c)) ,$$

where c=SQRT(2*h/(t*kyy))*d, t is the fin thickness, kyy is the fin transverse conductivity, and d is the average fin height.

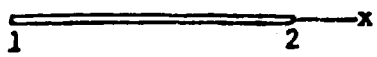
A fin factor is used to model multiple fins (see Section 2.1).



Associated Data Sets:

Section properties	- C62	SECT
Material properties	- COND	PROP
	- COND	COEF
Fluid/gas properties	- FLUI	PROP
	- GAS	PROP

MT21 - 2 node, mass-transport element. This element represents mass-transport and conduction due to fluid or gas flow in the x direction. Pressure drop computations are optional (see SSTA and TRTA RESET control PDROP).



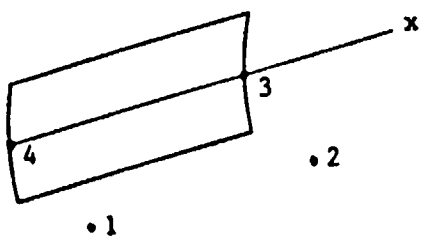
Associated Data Sets:

Flow area	-	MT21 SECT
Fluid/gas properties	-	FLUI PROP
	-	GAS PROP
Mass-transport rate	-	MTR MT21

MT42 - 4 node, integrated element. This element has two surface nodes and two fluid or gas nodes. It is characterized by:

- Conduction between the surface nodes (K21).
- Mass-transport and conduction between the fluid nodes (MT21).
- Convective exchange between the surface and fluid (C42).
- Optional pressure drop computations (see SSTA and TRTA RESET control PDROP).

Fluid flow is in the x direction. It is assumed that fluid nodes 1 and 2 lie within the planes normal to the x axis at surface nodes 4 and 3 respectively.



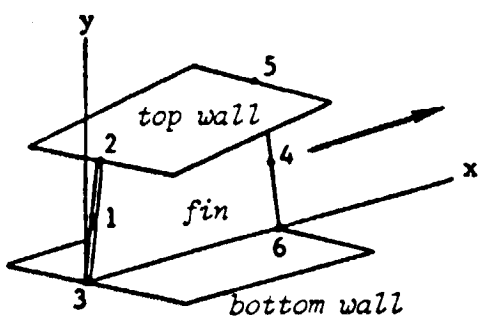
Associated Data Sets:

Section properties	-	MT42 SECT
Material properties	-	COND PROP
	-	COND COEF
Fluid/gas properties	-	FLUI PROP
	-	GAS PROP
Mass-transport rate	-	MTR MT42
Inlet pressure	-	P1 MT42

MT62 - 6 node, integrated element. This element has four surface nodes and two fluid or gas nodes. It consists of two walls connected by a fin and is characterized by:

- Two-dimensional heat conduction in the fin (K41).
- Mass-transport and conduction between the fluid nodes (MT21).
- Convective exchange between the walls and fluid or gas (C42).
- Convective exchange between both sides of the fin and the surrounding fluid or gas (C62). The computation of the fin efficiency is optional as described for element type C62 (see SSTA and TRTA RESET control FINEFF). A fin factor is used to model multiple fins (see Section 2.1).
- Optional pressure drop computations (see SSTA and TRTA RESET control PDROP).

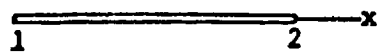
The fin is defined by nodes 3-6-5-2 and should be very nearly flat. The top wall is defined by nodes 2 and 5, and the bottom wall by nodes 3 and 6. Fluid flow is in the direction shown. For computational purposes, fluid nodes 1 and 4 are assumed to lie at the midpoints of the lines connecting nodes 2-3 and 5-6 respectively.



Associated Data Sets:

Section properties	-	MT62 SECT
Material properties	-	COND PROP
	-	COND COEF
Fluid/gas properties	-	FLUI PROP
	-	GAS PROP
Mass-transport rate	-	MTR MT62
Inlet pressure	-	P1 MT62

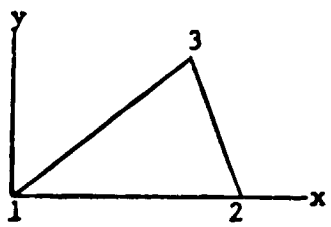
R21 - 2 node, line radiation element. This element exchanges radiant heat energy with other radiation elements as described in the SPAR Thermal Analysis Reference Manual - Volume 2, Theory. The radiation surface area is defined by nodes 1-2 and a width or perimeter. A point radiator is defined by specifying the same node number at each end or via TGEO execution command ZEROL. The length of a point radiator is set equal to unity.



Associated Data Sets:

Radiation perimeter	- R	CIRC
Material properties	- RADI	PROP
	- RADI	COEF
View Factors	- REX	

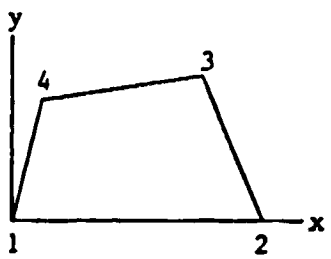
R31 - 3 node, area radiation element. This element exchanges radiant heat energy with other radiation elements as described in the SPAR Thermal Analysis Reference Manual - Volume 2, Theory.



Associated Data Sets:

Material properties	- RADI	PROP
	- RADI	COEF
View Factors	- REX	

R41 - 4 node, area radiation element. This element exchanges radiant heat energy with other radiation elements as described in the SPAR Thermal Analysis Reference Manual - Volume 2, Theory. The element should be very nearly flat.



Associated Data Sets:

Material properties	- RADI	PROP
	- RADI	COEF
View Factors	- REX	

3.2 ELD TABLE POINTER COMMANDS

The following commands must be used to specify the appropriate section and material, fluid, or gas properties associated with each thermal element.

Command	Default Value	Meaning
NSECT	1	Points to a line in the applicable section property table as described in Section 2.1.
MATERIAL NMAT	PROP 1	Insertion of the command MATERIAL=PROP into the ELD input runstream causes the current and any subsequently defined value of NMAT to point to COND PROP nmat, CONV PROP nmat, or RADI PROP nmat as appropriate. These data sets are described in Section 2.2.1. Insertion of the command MATERIAL=COEF into the ELD input runstream causes the current and any subsequently defined value of NMAT to point to a line in COND COEF, CONV COEF, or RADI COEF. These data sets are described in Section 2.2.2.
MEDIUM NMEDIUM	FLUID 0	Insertion of the command MEDIUM=FLUID into the ELD input runstream causes the current and any subsequently defined value of NMEDIUM to point to FLUI PROP nmedium. Insertion of the command MEDIUM=GAS into the ELD input runstream causes the current and any subsequently defined value of NMEDIUM to point to GAS PROP nmedium. Fluid and gas property data sets are described in Section 2.3.

The above commands may be inserted repeatedly, as required, into the ELD input runstream. At the conclusion of the definition of all elements of a given type, all commands will revert to their default values. The use of the above commands is illustrated in Section 3.3.

3.3 EXAMPLE ELD INPUT RUNSTREAM

The following example input runstream illustrates the use of the pointer commands described in Section 3.2.

```
@XQT ELD
 RESET NUTED=2$     Store all TED Eij data sets in library 2.
 K21$               Begin definition of all K21 elements.
 NSECT=6$           Conduction area from line 6 in K AREA.
 NMAT=7$            Conduction properties from COND PROP 7.
 43 62$             K21 element #1.
 21 24$             K21 element #2.
 NMAT=3$            Conduction properties from COND PROP 3.
 73 79$             K21 element #3.
 74 80$             K21 element #4.
 MATERIAL=COEF$     Conduction properties from line 1 of COND COEF.
 NMAT=1$
 NSECT=1$           Conduction area from line 1 of K AREA.
 48 41$             K21 element #5.
 -
 -
$                   All K21 elements defined.
$                   All pointers revert to their default values.
 C41$               Begin definition of all C41 elements.
 MATERIAL=COEF$     Convection properties from line 1 of CONV COEF.
 16 22 21 15$       C41 element #1.
 18 24 22 13$       C41 element #2.
 NMAT=2$            Convection properties from line 2 of CONV COEF.
 30 28 34 40$       C41 element #3.
 MATERIAL=PROP$     Convection properties from CONV PROP 1.
 NMAT=1$
 40 44 52 38$       C41 element #4.
 -
 -
$                   All C41 elements defined.
$                   All pointers revert to their default values.
 MT42$              Begin definition of all MT42 elements.
 NMAT=12$           Conduction properties from COND PROP 12.
 NMEDIUM=1$         Fluid properties from FLUI PROP 1.
 NSECT=2$           Section properties from line 2 of MT42 SECT.
 80 81 100 94$      MT42 element #1.
 81 82 110 100$     MT42 element #2.
 NMEDIUM=2          Fluid properties from FLUI PROP 2.
 6 7 21 18$         MT42 element #3.
 MEDIUM=GAS$        Gas properties from GAS PROP 1.
 NMEDIUM=1$
 NSECT=4$           Section properties from line 4 of MT42 SECT.
 11 12 33 16$       MT42 element #4.
 -
 -
$                   All MT42 elements defined.
$                   All pointers revert to their default values.
$ Other element definitions are similar to the above.
```

Section 4

TGEO - ELEMENT GEOMETRY PROCESSOR

The user must execute processor TGEO after each execution of ELD. TGEO checks the element definition data sets created by ELD for errors and computes element local coordinates. Output consists of the data sets TED GRPS, TED MPD, and TDOF ID which contain element, material property, and degree of freedom definition data required by processors SSTA and TRTA. If geometry errors are detected during execution of TGEO, data set TED GRPS is marked in error.

4.1 RESET CONTROLS

Name	Default Value	Meaning
ELIB	1	Library containing thermal element data sets generated by ELD.
LZERO	1.0E-20	Zero length test parameter. Element boundary lengths must be greater than LZERO (see TGEO execution command ZEROL for zero length bar elements).
THETA	.050	Minimum allowable angle (in radians) between adjacent element edges, used to detect badly shaped elements.
WARP	.100	Excessive warping parameter for four-node area elements. If an element's warpage exceeds WARP an error message is printed and the element's TED Eij ngrp data set is marked in error. Warpage is defined as Z/L, where Z is the distance from the plane of the first three nodes to the fourth node and L is the square root of element's area.
FLAT	.001	Threshold warping parameter for four-node area elements. If an element's warpage (see WARP) exceeds FLAT, a warning message is printed.

ASPECT	20.0	Maximum allowable aspect ratic - defined as the ratio of element edge length to characteristic length. The characteristic length is defined as the square root of the area for a four-node element and the square root of twice the area for a three-node element.
RCH	.0001	(R)t(R) must equal the identity matrix within a tolerance of RCH, where R is the direction cosine matrix of an element local coordinate frame with respect to the global coordinate system.
HEIGHT	.0001	Minimum height ratio for hexahedral elements. The height ratio is defined as H/L, where H is the distance from the element base formed by the "plane" of nodes 1-4 to each of the nodes 5-8, and L is the square root of the area of the element base.
HEXF	.010	Threshold warping parameter for pentahedral and hexahedral element faces (see FLAT).
HEXW	.050	Excessive warping parameter for pentahedral and hexahedral element faces (see WARP).
PRINT	2	Print control parameter for element geometry errors: = 0, no printing. = 1, print error messages. = 2, print error and warning messages.
NERR	5	Maximum number of geometry error and warning messages to be printed for each element type.
FREEZE	0	Unconnected node restraint parameter: = 0, unconnected nodes will cause singular system K and capacity matrices. = 1, unconnected nodes will not affect the solution to either steady-state or transient analyses.
LIST	100	Number of unconnected nodes to list.

4.2 DEFINITION OF ZERO LENGTH ELEMENTS - The ZEROL Command

The ZEROL command, if present, must follow the last RESET control.

Zero length elements are used to model point conductors and/or sources (K21), point convectors (C21), and point radiators (R21). The lengths of zero length elements are set equal to unity by processor TGEO. Zero length elements may be defined in either of two ways:

a) Bar elements defined with the same node number at each end are understood to be zero length elements.

b) The command

ZEROL Eij ngrp element index string

causes the elements listed in the index string to be labeled as zero length elements. If the index string is left blank, all elements within group ngrp are assumed to be zero length elements. Loop limit format is not permitted.

Example:
```
ZEROL   K21   2   4, 5, 9, 12
ZEROL   C21   1
```

The four named elements of type K21, group 2 and all elements of type C21, group 1 are labeled as zero length elements.

4.3 CENTRAL MEMORY REQUIREMENTS

The required data space is approximately

$$2000 + 4*njts ,$$

where njts is the number of joints in the structure.

Section 5

STEADY STATE SOLUTIONS

To generate solutions to steady state problems, the user must:
(1) create source data tables which define thermal excitation as described in Section 5.1, and
(2) execute processor SSTA, as described in Section 5.2, to compute temperature distributions and associated solution data.

5.1 DEFINITION OF STEADY STATE THERMAL EXCITATION

To specify steady state thermal excitation, the user must execute AUS/TABLE to create any of the following data sets which apply to the specific problem to be solved.

Block 1 of each data set corresponds to load case 1, block 2 to load case 2, etc., except for MTR Eij, TEMP NODE, and REX.

For each element excitation table, the second and third words in the data set name are Eij and ngrp, indicating group ngrp of element type Eij. The TOC parameters must be NI=1 and NJ=the number of elements in the group. Line j must contain data corresponding to element j in the indicated group of elements.

Further details are given in Sections 5.1.1 through 5.1.6.

Excitation Type	Data Set Name	NI	Line Contents	Applicable Element Types
Element:				
Source or flux heating	SOUR Eij ngrp	1	Q(W/m**3)	K21 K31 K41 K61 K81
Convection to a known temperature	CTEM Eij ngrp	1	Th(Deg K)	C21 C31 C41
Mass-transport rates	MTR Eij ngrp	1	M(kg/sec)	MT21 MT42 MT62
Inlet pressures	P1 Eij ngrp	1	P(N/m**2)	MT21 MT42 MT62
Nodal:				
Prescribed nodal temperatures	TEMP NODE APPL TEMP	1 1	n Tn(Deg K)	Tn=temperature at node n.
Radiation:				
Exchange factors	REX (see Section 5.1.6)			R21 R31 R41

5.1.1 Source and Flux Heating

Source heat rates are assumed to be uniformly distributed over the volumes of conducting elements K21, K31, K41, K61, and K81. Heat rates need not be defined for all conducting elements; however, if a heat rate is defined for any element within a group, heat rates must be defined for all elements contained within that group. The total heat generated = Q*volume.

Flux heating or cooling of rod and plate elements can be simulated by using source heat rates and conducting elements K21, K31, and K41. Care should be taken to insure that the source heat rates are properly scaled to give the correct heat flux. For example, to specify a line flux (Watts/meter) along the length of a K21 element, the source heat rate must be the prescribed line flux divided by the element area. To specify a flux over the surface of a K31 or K41 element, the source heat rate must be the prescribed surface flux divided by the element thickness.

5.1.2 Convection to a Known Temperature

Convection to a known temperature (e.g. a fluid at a specified temperature) is associated with element types C21, C31, and C41. A convective exchange temperature must be given for every C21, C31, and C41 element in the finite element model. The heat flux leaving the element is given by q=h*(T-Th), where h is the convection coefficient, T is the average element temperature, and Th is the convective exchange temperature.

5.1.3 Mass-Transport Rates

A mass transport rate must be given for every MT21, MT42, and MT62 element. For steady state analyses, each MTR Eij ngrp data set must have only one block.

5.1.4 Inlet Pressures

Inlet pressures are required only when element pressure drops are computed (see SSTA and TRTA RESET control PDROP). The first word in every block of each P1 Eij ngrp data set must be non-zero (see SSTA and TRTA RESET control PZERO).

If an element's inlet pressure is input as zero, it will be set equal to the computed outlet pressure of the of the element with the next lowest index number within the element group. Inlet pressures for successive elements will be defined in this way until a non-zero inlet pressure is encountered.

5.1.5 Prescribed Temperatures

Nodal temperatures may be prescribed at any desired locations. There is no limit to the number of nodes at which temperatures may be specified.

Example: To prescribe the temperatures at nodes 5, 20, and 50 as 100.0, 200.0, and 200.0 for load case 1 and 144.0, 275.0, and 290.0 for load case 2:

```
TABLE(NJ=3):TEMP NODE:J=1,3:              5.   20.   50.
TABLE(NJ=3):APPL TEMP:BLOCK 1:J=1,3:  100.  200.  200.
                      BLOCK 2:J=1,3:  144.  275.  290.
```

An alternate method for prescribing temperatures at a point, along a line, or on a surface is to specify high convection coefficients for element types C21, C31, and C41, as applicable. The appropriate convective exchange temperatures (Section 5.1.2) must be set equal to the desired prescribed temperatures.

To apply this method to a transient analysis, all nodes connected to elements with high convection coefficients must be defined as "arithmetic nodes" (see Section 6.2.1.2) in order to avoid an unneccesarily small integration time step size.

5.1.6 Radiation-Exchange Factors

Exchange factors for diffuse and/or specular radiation exchange must be defined by constructing data sets named REX Rid n, where Rid may be any user defined name and n must be a positive integer which uniquely defines the data set (see RESET Control NREX). TOC parameters NI and NJ and the number of blocks in each data set are arbitrary.

```
Example: TABLE(NI=9,NJ=16): REX BASE 1
         TABLE(NI=5,NJ=23): REX FR11 2
         TABLE(NI=21,NJ=8): REX TUBE 3
```

The contents of each data line must be as follows:

i, string of element number and exchange factor pairs (j,Fij)

where i and j are element numbers and Fij is the exchange factor defined as the amount of radiant heat energy leaving element i that falls on element j. Elements for which exchange factors are not defined radiate to space.

Radiation elements are numbered in the order in which they are defined during execution of processor ELD.

5.2 SSTA - STEADY STATE THERMAL ANALYZER

SSTA generates steady state solutions from the following data:

- Element definition tables from ELD and TGEO.
- Material, fluid, and gas property tables from AUS/TABLE.
- If necessary, a joint elimination sequence data set generated by TAB/JSEQ. A skyline, banded matrix technique is used to factor the K matrix in SSTA; hence, the numbering sequence of choice is that which minimizes the rms bandwidth, otherwise all comments contained in Section 3.1.16 of the SPAR Reference Manual (Volume 1) apply.
- User input RESET controls and execution commands.

Computed nodal temperatures are stored in data set STAT TEMP 1 1 (see Section 5.2.7.1). To avoid a proliferation of data sets named STAT TEMP 1 1, nodal temperatures are always written into an existing STAT TEMP 1 1 data set whenever the existing data set has the same block length as the nodal temperature vector and an equal or greater number of blocks than that specified by RESET control L2.

5.2.1 LINEAR ANALYSIS

SSTA will automatically compute linear solutions whenever there are no radiation elements and no temperature-dependent material, fluid, or gas properties. The solution process is completely automated; SSTA controls the assembly and factoring of the system K matrix, computes thermal load vectors, and solves for nodal temperatures. For a normal linear analysis with one load case, the following SSTA runstream is all that is required to compute nodal temperatures.

```
@XQT SSTA
STOP
```

Any number of load cases may be processed during a single execution. Specific load cases are identified by RESET controls L1 and L2. For example, if each thermal excitation data set contains data for 6 load cases (see Section 5.1), the following SSTA runstream is required to compute solutions for all 6 load cases.

```
@XQT SSTA
RESET L2=6: STOP
```

Factored K matrices are always retained so that solutions may be generated with an existing factored K matrix. For example, suppose that data sets defining 5 new load cases for the above problem have been created. To generate solutions using the factored K matrix produced during the above execution, the following runstream is required.

```
@XQT SSTA
RESET L2=5 START=KFAC: STOP
```

5.2.2 NONLINEAR ANALYSIS

SSTA will automatically perform a nonlinear analysis whenever there are radiation elements or any temperature-dependent material, fluid, or gas properties. Before attempting to execute SSTA in its nonlinear mode of operation, the user should become thoroughly familiar with the contents of this section. The modified Newton method of nonlinear analysis is described in Section 5.2.2.1. The use of RESET controls and execution commands to guide the nonlinear solution process is described in Section 5.2.2.2.

5.2.2.1 NONLINEAR ANALYSIS THEORY

The modified Newton procedure used in SSTA to solve nonlinear problems is described briefly below.

For the nonlinear one-degree-of-freedom equation

$$y(x) = 0 \ ,$$

the Newton method of solution involves successively solving the recurrence equation

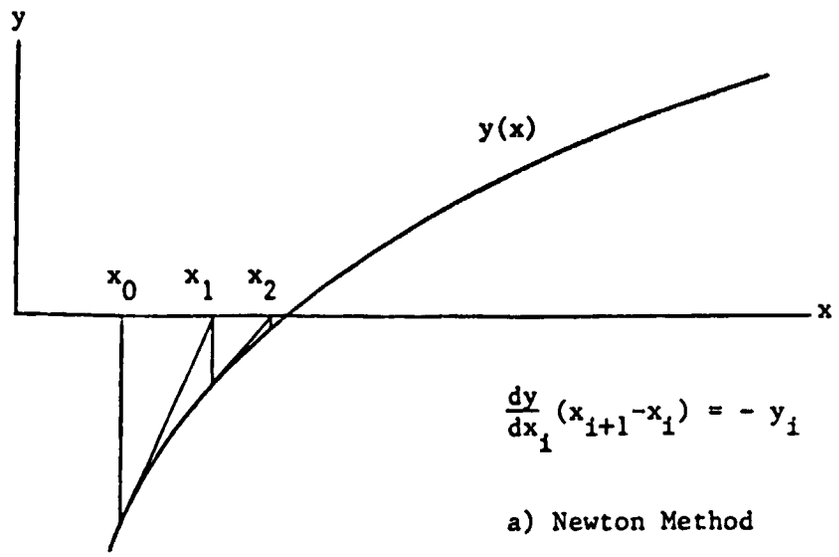
$$\frac{dy}{dx_i}(x_{i+1}-x_i) = - y_i$$

for x_{i+1} until satisfactory convergence is achieved; for example, until the absolute value of $(x_{i+1}-x_i)/x_i$ becomes smaller than a predetermined value. The process is shown graphically on Figure 5-1a.

An alternate approach to that described above is to iterate with a constant value of $\frac{dy}{dx}$ as shown on Figure 5-1b. The recurrence equation for this "modified" Newton procedure is

$$\frac{dy}{dx_0}(x_{i+1}-x_i) = - y_i \ .$$

It is readily seen that the Newton method requires fewer iterations for convergence, while the modified Newton method requires less arithmetic because computing $\frac{dy}{dx}$ is done but once.



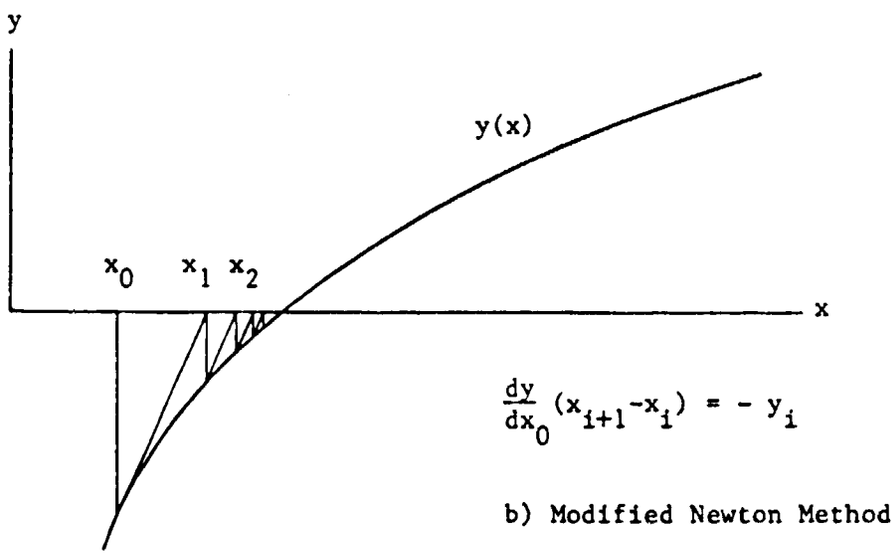
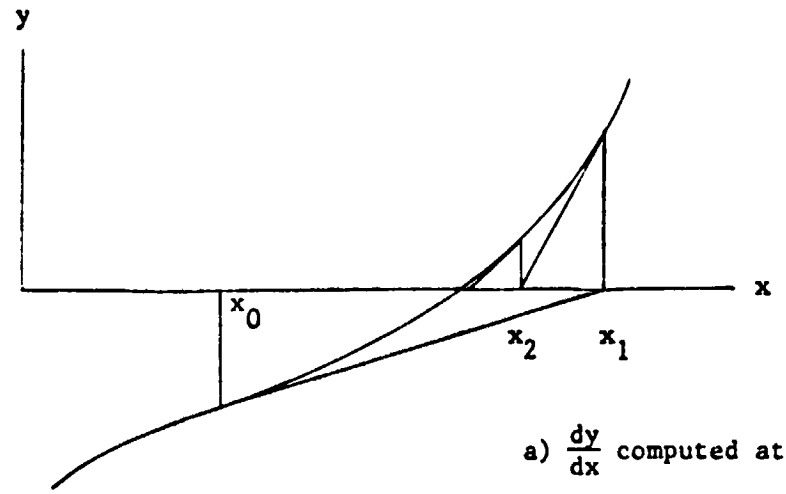
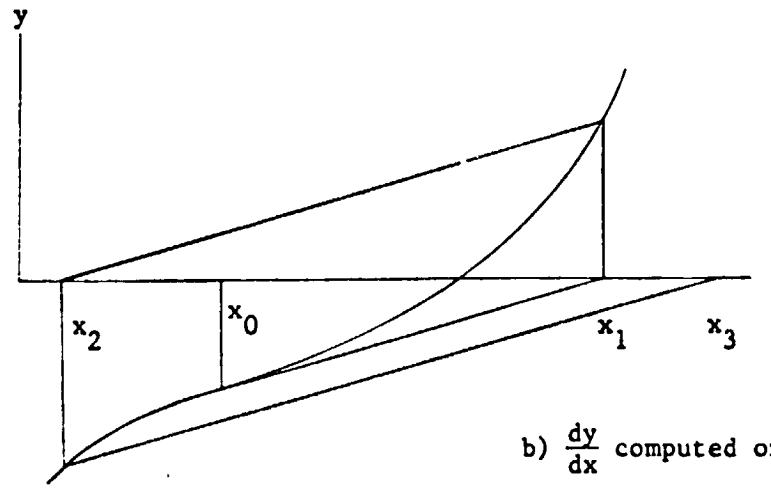


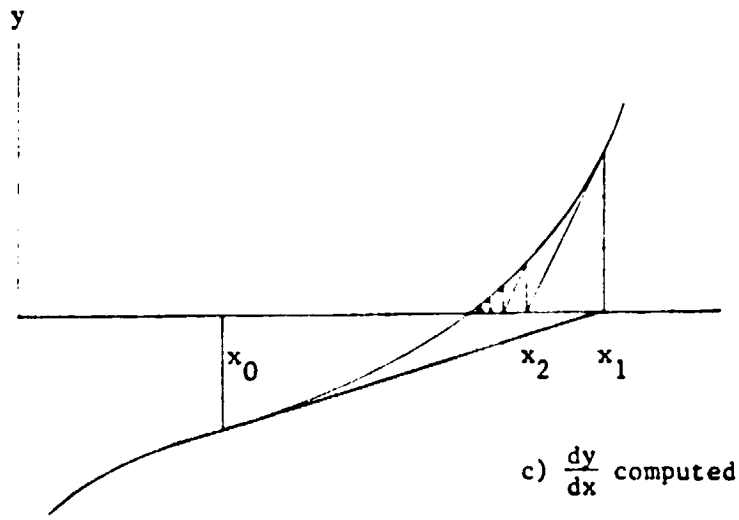
Figure 5-1 Newton and Modified Newton Methods



a) $\frac{dy}{dx}$ computed at each iteration



b) $\frac{dy}{dx}$ computed only at x_0



c) $\frac{dy}{dx}$ computed at x_0 and x_1

Figure 5-2 Modified Newton Method Convergence Characteristics

There is no guarantee that a modified Newton procedure will work for a particular problem. Indeed it may be necessary, depending on the nature of the problem, to compute $\frac{dy}{dx}$ more than once in order to prevent divergence or poor convergence characteristics. Figure 5-2 illustrates such a problem - clearly a modified Newton procedure with $\frac{dy}{dx}$ computed only at x_0 leads to divergence, while computing $\frac{dy}{dx}$ at each iteration does not seem necessary. A reasonable alternative for this problem is to compute $\frac{dy}{dx}$ at x_0 and x_1 and then iterate without recomputing $\frac{dy}{dx}$ until convergence is achieved.

Figure 5.2 also illustrates the importance of the initial estimate for x_0 on the convergence charcteristics of the problem. If x_0 is moved to the right, convergence will occur regardless of when $\frac{dy}{dx}$ is computed.

For the multi-degree-of-freedom system

$$\psi(T) = 0 \ ,$$

the recurrence equation is

$$K(T_{i+1}-T_i) = -\psi_i \ ,$$

where ψ and T are vectors and K is the square derivative matrix $\left[\frac{\partial \psi}{\partial T}\right]$. The method of solution is exactly analogous to that for the one-degree-of-freedom system described above and shown on Figures 5-1 and 5-2. If a system has many degrees of freedom, the computer time required to form and factor the K matrix is usually much greater than the computer time required to solve for T_{i+1}. This makes the modified Newton method especially attractive for large problems.

5.2.2.2 NONLINEAR ANALYSIS STRATEGY

To perform a nonlinear analysis, the user must create an initial temperature vector with which to start the analysis (see Initial Temperature Vector below) and then execute SSTA as required until satisfactory convergence is achieved. During each execution of SSTA, the following sequence of events occurs:

- The DEST library (see RESET control DEST) is searched for a data set named STAT TEMP 1 1. The temperature vector contained within this data set is used to start the execution. If STAT TEMP 1 1 is not found, an error termination results. (Also see the TEMP command).

- Execution continues until (a) convergence occurs, (b) the number of factorings and iterative solutions specified by the user (see RESET controls NFACS and NITER) have been completed, or (c) an error is detected (see Section 5.2.6). If convergence is not achieved, or if an error is detected, the error code of STAT TEMP 1 1 has the value specified by RESET control TERR.

- At the completion of execution, the last computed temperature vector is written into STAT TEMP 1 1.

The instructions given below and on the following pages are meant to serve as introductory guidelines for nonlinear analysis. The execution sequences and specific numerical values assigned to the RESET controls are for purposes of illustration only. Sequences and numerical values used during an actual analysis will depend on many factors, primarily (a) the cost of forming and factoring the K matrix versus the cost for each iterative solution and (b) the convergence characteristics of the problem.

Initial Temperature Vector

Before starting a nonlinear analysis, the user must create a data set named STAT TEMP 1 1 containing estimated nodal temperatures. The estimated temperature distribution used to start the analysis is very important. A good choice may lead to rapid convergence, while a poor choice may lead to slow convergence or divergence. Normally, the best method of creating an initial temperature vector is to perform a linear analysis (RESET PROB=LINEAR), using the TEMP command to specify the temperature at which temperature-dependent properties are computed.

```
@XQT SSTA
RESET PROB=LINEAR
TEMP=temperature at which properties are computed
STOP
```

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                       *
*  As a nonlinear analysis progresses, the user should be very          *
*  careful to insure that, after each execution of SSTA, con-           *
*  verging solutions are saved.  If this is not done, and if a          *
*  diverging solution is written into STAT TEMP 1 1 during any          *
*  subsequent execution of SSTA, the analysis will probably             *
*  have to be restarted from the beginning.  The recommended            *
*  procedure for saving converging solutions is to copy STAT            *
*  TEMP 1 1 from the DEST library into another library.                 *
*                                                                       *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

First Execution

In order to preserve the first computed temperature vector (see Figure 5-2b), the first execution should have one factoring and zero iterations.

```
@XQT SSTA
RESET NFACS=1 NITER=0
STOP
```

Second Execution

To assess the initial convergence characteristics, the second execution should utilize the factored K matrix produced during the first execution, and have zero factorings and 5 iterations.

```
@XQT SSTA
RESET START=KFAC NFACS=0 NITER=5
STOP
```

Restarting After Rapid Convergence

There is no need to factor if convergence is rapid, therefore the next execution should utilize the factored K matrix produced during the previous execution and have ten iterations.

```
@XQT SSTA
RESET START=KFAC NFACS=0 NITER=10
STOP
```

Restarting After Slow Convergence

In order to determine if factoring will improve the convergence characteristics of the problem, executions following slow convergence should have one factoring and 5 iterations.

```
@XQT SSTA
RESET NFACS=1 NITER=5
STOP
```

Restarting after Very Slow Convergence

If convergence is very slow, the following sequence of operations is suggested:

1) Execute SSTA with 2 factorings and zero iterations.
```
@XQT SSTA
RESET NFACS=2 NITER=0: STOP
```

2) Execute SSTA with 1 factoring and 5 iterations.
```
@XQT SSTA
RESET NFACS=1 NITER=5: STOP
```

3) If convergence continues to be very slow, the convergence criteria should probably be made less restrictive.
```
@XQT SSTA
RESET NFACS=0 NITER=10 CONVERGE=xxx CUTOFF=yyy: STOP
```

Restarting After Divergence

Following a divergent execution, the next sequence of operations should be:

1) Copy the STAT TEMP data set that started the divergent execution into the DEST library.

2) Execute SSTA with 2 factorings and zero iterations.
```
@XQT SSTA
RESET NFACS=2 NITER=0: STOP
```

3) Execute SSTA with zero factorings and 5 iterations.
```
@XQT SSTA
RESET NFACS=0 NITER=5: STOP
```

4) If divergence continues to occur despite repeated factorings, the whole nonlinear analysis should be restarted with a more suitable initial temperature vector.

Negative Temperatures

The presence of negative temperatures indicates severe divergence problems or an incorrectly defined initial temperature vector (e.g. temperatures measured with respect to zero Celsius rather than zero Kelvin). Each time a negative temperature is detected, SSTA will print a warning message, but execution will continue unless RESET control NEGTEMP is set to a negative number.

5.2.3 RESET CONTROLS

Name	Default Value	Meaning
SOURCE	1	Input library.
DEST	1	Output library.
ELIB	(SOURCE)	Library containing the TED Eij data sets.
QLIB	(SOURCE)	Thermal excitation library.
KLIB	(DEST)	K matrix library.
L1 L2	1 1	Load cases L1 through L2 will be processed. Any number of load cases may be processed during an execution.
LK	2240	K matrix block length.
KPRT	1	Print control parameter used during factoring of the K matrix: = 0, no printing. = 1, singularity messages will be printed. = 2, negative root messages will be printed. = 3, singularity and negative root messages will be printed.
NSING	0	Number of singularities allowed during factoring of the K matrix before error termination. Singular equations detected during factoring will be removed from the system and will have no effect on the solution process.
SING	1.0E-6	Singularity parameter. If the absolute value of a factored diagonal term is less than or equal to SING times the absolute value of the diagonal term before factoring, the matrix is singular.
ZERO	1.0E-20	Singularity parameter. If the absolute value of a factored diagonal term is less than or equal to ZERO, the matrix is singular.
KBWR	0	Assembly IO access count parameter: = 0, no effect. = 1, an IO access summary will be printed upon completion of assembly of the K matrix. = 2, the number of IO accesses required to assemble the K matrix will be printed. An analysis will not be performed.

PROB	NONLINEAR	If there are no radiation elements and no temperature-dependent material, fluid, or gas properties, a linear analysis will be performed automatically. PROB=LINEAR will cause a linear solution to be computed, even if there are radiation elements or temperature-dependent properties. Radiation effects will be ignored, and properties will be computed at the temperatures contained in an existing STAT TEMP 1 1 data set, or at the temperature defined by the TEMP command.
START		START=KFAC will cause the solution process to be started with an existing factored K matrix resident in library KLIB. If a joint elimination sequence data set (JSEQ BTAB) existed during assembly and factoring of the K matrix, that same JSEQ BTAB data set must reside in library 1 during the current execution. If a TEMP NODE data set (see Section 5.1.5) existed when the K matrix was factored, that same TEMP NODE data set must reside in library QLIB during the current execution.
NFACS	1	Maximum number of factorings allowed during a nonlinear execution.
NITER	0	Maximum number of iterations allowed after each factoring.
CONVERGE CUTOFF	.001 .200	Convergence parameters. Convergence is assumed if ABS(DT/T) is less than or equal to CONVERGE for all nodal temperatures greater than CUTOFF*Tave. DT is the change in nodal temperature, T is the nodal temperature, and Tave is the average of all nodal temperatures.
NEGTEMP	3	Maximum number of negative temperatures to print during each iteration. If NEGTEMP is negative, execution will be terminated when negative temperatures are encountered.
TERR	1	Error code in the table of contents line of each output data set if an error is detected or if convergence is not achieved during a nonlinear execution.

FINEFF	0	Fin efficiency computation parameter for MT62 and C62 elements: = 0, the fin efficiency for all MT62 and C62 elements is 1. = 1, the fin efficiency for each MT62 and C62 element will be computed from the formula given in Section 3.1 - element type C62.
FLUX	0	Heat flux computation parameter (see Section 5.2.7.2): = 0, element fluxes will not be computed. = 1, element fluxes will be computed from the temperatures contained in each block of an existing STAT TEMP data set. = 2, element fluxes will be computed for load cases L1 through L2 after the solution is finished (not recommended for nonlinear analyses).
PDROP	0	Pressure drop and outlet pressure computation parameter for mass-transport elements (see Section 5.2.7.3): = 0, element pressures will not be computed. = 1, element pressures will be computed from the temperatures contained in each block of an existing STAT TEMP data set. = 2, element pressures will be computed for load cases L1 through L2 after the solution is finished (not recommended for nonlinear analyses).
PZERO	1.0E-10	Zero inlet pressure parameter. If an element's inlet pressure is less than PZERO, it will be set equal to the outlet pressure of the element with the next lowest index number within the element group.
G	1.0	Acceleration of gravity (force/mass).
ALPHA		Mass-transport matrix computation parameter (see Section 3 of the SPAR Thermal Analysis Reference Manual, Volume 2 -Theory). If ALPHA is specified, its value will be used to compute the mass-transport matrix for all MT21, MT42, and MT62 elements. If ALPHA is not specified, it will be computed for each element according to the formula given in the above reference.

The following RESET controls are applicable only when the finite element model contains radiation elements.

SBCON	.56697E-7	Stefan-Boltzmann constant, the default value is expressed in SI units.
RLIB		Source library for REX data sets. RLIB must be specified for radiation exchange problems.
NREX	1	Number of REX data sets. Exchange factors will be obtained from data sets REX MASK 1 through REX MASK nrex (see Section 5.1.6). NREX should be 1 for the most efficient computation of radiation load vectors.
RSTEPS	3	Maximum number of steps allowed during computation of the radiation load vector.
RCONV	.01	Radiation load vector convergence parameter. Convergence of the radiation load vector is assumed when QR/QE is less than RCONV, where QR is the sum of the heat reflected by all elements during the current step, and QE is the total heat emmitted by all elements.
RCC	0	Radiation convergence action parameter: = 0, execution will be terminated if convergence is not achieved during the computation of a radiation load vector. = 1, execution will be continued, even if convergence is not achieved. = 2, same as 1 above, except a warning message will be printed each time convergence is not achieved.
RDIST	1	Reflected heat energy distribution parameter: = 0, residual reflected heat energy will be lost to space. = 1, residual reflected heat energy will be distributed to all radiation elements in proportion to the total heat absorbed by each element.

5.2.4 EXECUTION COMMANDS

The execution commands described below, if present, must follow the last RESET control.

5.2.4.1 Initial Temperature Estimate - The TEMP Command

TEMP= T

specifies a uniform temperature T at every node in the finite element model for the purpose of computing initial estimates of temperature-dependent properties during the first execution of a nonlinear analysis, or if RESET control PROB=LINEAR.

If the TEMP command is not used, the DEST library is searched for STAT TEMP 1 1, and the appropriate blocks (see RESET controls L1 and L2) are used as initial temperature vectors. If the data set is not found, an error termination will result.

5.2.4.2 Radiation Heat Flux Computations - The RFLUX Command

RFLUX= lib, list of options (E, H, B, QE, QA, QR)

causes the indicated radiation flux quantities to be computed for all radiation elements from temperatures contained in an existing STAT TEMP 1 1 data set. Data sets containing the flux quantities are described in Section 5.2.7.2. If lib is omitted, each data set will reside in the DEST library. If no options are given, all flux quantities will be computed.

E = Emissive power
H = Incident heat flux (irradiation)
B = Radiosity
QE = Heat emitted
QA = Heat absorbed
QR = Heat reflected

5.2.5 CENTRAL MEMORY REQUIREMENTS

The required data space is approximately

$$3000 + NJTS + LK + L ,$$

where NJTS is the number of joints, LK is the K matrix block length, and L is the greater of LK or 2*NJTS.

5.2.6 ERROR MESSAGES

If a fatal error condition is detected during program execution, the message NERR, N = XXXX, n will be printed, and execution will be terminated. Error messages and their causes are listed below.

NERR	N	Meaning
CORE	n	Allowable data space exceeded, increase core size by n.
KBWR	2	Normal termination when RESET control KBWR=2.
SSTA	0	As indicated by printed message.
SSTA	1	No such execution command.
SSTA	2	Incorrect TEMP command.
SSTA	3	L2 is greater than the number of blocks in the named data set.
SSTA	4	The length of the named data set does not agree with the number of elements in the corresponding TED Eij ngrp data set.
SSTA	5	The wrong skyline vector is in library KLIB.
SSTA	6	The wrong K matrix is in library KLIB.
SSTA	7	The K matrix has the wrong precision parameter.
SSTA	9	RESET control PROB has the wrong image.
SSTA	12	L1 is less than 1 or greater than L2.
SSTA	13	The TEMP command was not utilized, nor is there an existing STAT TEMP data set (see Section 5.2.4).
SSTA	14	The existing STAT TEMP data set is marked in error.
SSTA	15	NJ of data set STAT TEMP is not equal to the number of joints in the finite element model.
SSTA	16	L2 is too large for the existing STAT TEMP data set.
SSTA	17	Data sets TEMP NODE and APPL TEMP are incompatible.
SSTA	18	RESET controls FLUX and PDROP are inconsistent.
SSTA	19	Conflict between RESET controls NFACS and START.
SSTA	20	Incorrect RFLUX command.
SSTA	21	Conflict between RESET control FLUX or PDROP and execution command RFLUX.
SSTA	22	RESET control RLIB must be used with command RFLUX.
TACD	0	As indicated by printed message.

5.2.7 OUTPUT DATA SET CONTENTS

Steady state results are stored in the data sets described in this section. For each data set, block 1 corresponds to load case 1, block 2 to load case 2, etc. Within each block, the j-th line contains data corresponding to the j-th node or element, as appropriate. All data sets reside in the DEST library unless noted otherwise.

5.2.7.1 Nodal Temperatures

Nodal temperatures are stored in the following data set. TOC parameter NJ=the number of nodes in the finite element model.

Data Set Name	NI	Line Contents
STAT TEMP 1 1	1	Nodal temperature

5.2.7.2 Element Heat Fluxes

The data sets tabulated below are generated when RESET control FLUX=1 or FLUX=2. For each data set, TOC parameter NJ=the number of elements in the corresponding TED Eij ngrp data set.

Data Set Name	NI	Line Contents
SFLX K21 ngrp 1	1	Fx
SFLX K31 ngrp 1	2	Fx Fy
SFLX K41 ngrp 1	2	Fx Fy
SFLX K61 ngrp 1	3	Fx Fy Fz
SFLX K81 ngrp 1	3	Fx Fy Fz
SFLX C21 ngrp 1	1	Fc
SFLX C31 ngrp 1	1	Fc
SFLX C41 ngrp 1	1	Fc
SFLX C32 ngrp 1	1	Fs
SFLX C42 ngrp 1	1	Fs
SFLX C62 ngrp 1	2	Fs Eff
SFLX MT21 ngrp 1	2	Fm Fk
SFLX MT42 ngrp 1	4	Fm Fk Fs Fx
SFLX MT62 ngrp 1	8	Fm Fk Fs Fw1 Fw2 Fx Fy Eff
SFLX R21 ngrp 1 *	1	E
SFLX R31 ngrp 1 *	1	E
SFLX R41 ngrp 1 *	1	E

* These data sets are created only when no radiation exchange factors (Section 5.1.6) have been defined.

In the element flux data sets:

Fx, Fy, Fz = conductive fluxes in x, y, and z directions.
Fc = convective flux, positive leaving element.
Fs = surface-fluid convective exchange flux, positive entering fluid.
Eff = fin efficiency.
Fm = fluid mass-transport flux.
Fk = fluid conductive flux.
Fw1, Fw2 = top and bottom wall fluid-surface convective exchange fluxes, positive entering fluid.
E = emissive power.

The following data sets are generated when execution command RFLUX is used. For each data set, TOC parameter NJ=the total number of radiation elements.

Data Set Name	NI	Line Contents
SREX E 1 1	1	Emissive power
SREX E 1 1	1	Incident radiant heat flux (irradiation)
SREX B 1 1	1	Radiosity
SREX QE 1 1	1	Heat emitted
SREX QA 1 1	1	Heat absorbed
SREX QR 1 1	1	Heat reflected

5.2.7.3 Element Fluid Pressures

The data sets tabulated below are generated when RESET control PDROP=1 or PDROP=2. For each data set, TOC parameter NJ=the number of elements in the corresponding TED Eij data set.

Data sets containing element inlet pressures must exist before pressures can be computed (see Section 5.1.5).

Data Set Name	NI	Line Contents
SPD MT21 ngrp 1	4	P1, P2, PDF, PDA
SPD MT42 ngrp 1	4	"
SPD MT62 ngrp 1	4	"

In the above:

P1 = inlet pressure.
P2 = outlet pressure.
PDF = friction pressure drop.
PDA = flow acceleration pressure drop.

5.2.8 K MATRIX DATA SETS

The factored K matrix data sets tabulated below are always stored in order to provide a restart capability for linear and nonlinear analyses.

All K matrix data sets reside in the KLIB library and are always written into existing K matrix data sets having the same name and TOC parameters.

KF POIN, KF SKY, and KF TEMP are the normally encountered K matrix data sets. KAF POIN, KAF SKY, and KAF TEMP result from the inclusion of mass-transport elements in the finite element model.

Data Set Name	NJ	NI*NJ	Data Set Contents
KF SKY nrms mxbw	nk	lk	Factored symmetric K matrix.
KF POIN nrms mxbw	nk	np	Skyline vector of KF SKY.
KF TEMP nrms mxbw	nt	nt	Modified diagonal K matrix terms which account for prescribed nodal temperatures.
KAF SKY nrms mxbw	nk	lk	Factored asymmetric K matrix.
KAF POIN nrms mxbw	nk	np	Skyline vector of KAF SKY.
KAF TEMP nrms mxbw	nt	nt	Modified diagonal K matrix terms which account for prescribed nodal temperatures.

In the above data sets:

nrms = rms off-diagonal bandwidth,
mxbw = maximum off-diagonal bandwidth,
nk = number of K matrix blocks,
np = number of nodes + 3*nk,
lk = K matrix block length,
nt = number of nodes at which temperatures are prescribed.

Section 6

TRANSIENT SOLUTIONS

To generate solutions to transient problems, the user must:
(1) create source data tables which define thermal excitation as described in Section 6.1, and
(2) execute processor TRTA, as described in Section 6.2, to compute temperature distributions and associated solution data.

6.1 DEFINITION OF TRANSIENT THERMAL EXCITATION

To specify transient thermal excitation, the user must execute AUS/TABLE to create data sets having the same name, content, and structure as those described in Section 5.1 (Steady State Thermal Excitation). For steady state excitation, successive blocks in these data sets defined successive static load cases. For transient excitation, successive blocks correspond to specific times, which the user must define by creating, through AUS/TABLE, the XXXX TIME data sets summarized below.

Data Set Name	Associated Excitation Data Sets
SOUR TIME	All SOUR Eij ngrp data sets
CTEM TIME	All CTEM Eij ngrp data sets
MTR TIME	All MTR Eij ngrp and P1 Eij ngrp data sets
TEMP TIME	APPL TEMP

Each XXXX TIME data set must consist of a single block. For each data set, the TOC parameters must be NI=1 and NJ=the number of time points. The number of time points can be different for each XXXX TIME data set.

The times in each XXXX TIME data set must increase sequentially. Processor TRTA computes quantities at intermediate times by linear interpolation. Extrapolation beyond the first or last time is not allowed. The first and last times in each XXXX TIME data set must bracket the anlysis starting and termination times (see RESET controls T1 and T2 or execution command TIME).

For example, the k-th word in SOUR TIME is the time which corresponds to the k-th block in all SOUR Eij data sets. It is mandatory that the number of blocks in each excitation data set (e.g. SOUR Eij ngrp) be equal to the number of times in the associated XXXX TIME data set; with the exception that any single-block data set is assumed to define excitation constant in time.

6.2 TRTA - TRANSIENT THERMAL ANALYZER

TRTA generates transient solutions from the following data:

- Element definition tables from ELD and TGEO.
- Material, fluid, and gas property tables from AUS/TABLE.
- Transient thermal excitation tables from AUS/TABLE.
- User input RESET controls and execution commands.

TRTA permits the user to compute transient solutions in several successive executions, using as initial conditions temperatures computed at any point in time in any previous execution. This allows the user to perform a series of short test executions before being committed to long, expensive computer runs.

Either an explicit or implicit time integration method can be chosen by the user. If the explicit method is chosen (default mode of operation), the integration time step size can be input by the user or computed internally by TRTA. If the implicit method is chosen, the integration time step size must be input by the user.

A typical TRTA runstream for performing a transient analysis using the explicit method is

```
@XQT TRTA
RESET T1= analysis starting time
RESET T2= analysis termination time
TEMP= initial temperature vector
TSAVE= times at which results are stored
STOP
```

A typical TRTA runstream for performing a transient analysis using the implicit method is

```
@XQT TRTA
RESET METHOD=IMPLICIT
RESET DT= integration time step size
RESET T1= analysis starting time
RESET T2= analysis termination time
TEMP= initial temperature vector
TSAVE= times at which results are stored
STOP
```

Computed nodal temperatures are stored in data set TRTA TEMP 1 1 (see Section 6.2.6). Each block in TRTA TEMP 1 1 corresponds to a specific time in data set TRTA TIME 1 1 (see the TSAVE command, Section 6.2.3.3). In order to preserve the results from previous executions, existing TRTA TEMP 1 1 and TRTA TIME 1 1 data sets are disabled each time TRTA is executed.

6.2.1 GUIDELINES FOR TRANSIENT ANALYSIS

In the explicit solution method, the temperature vector at a point in time is expressed as a Taylor series of the temperature vector and its time derivatives at a previous point in time. The implicit solution method is based on the Galerkin or weighted residual method. Both methods are described in Section 4.3 of the SPAR Thermal Analysis Reference Manual, Volume 2 - Theory.

6.2.1.1 THE TIME INTERVAL

For both the explicit and implicit time integration methods, the solutions are carried out within specific regions of time called "time intervals" which have the properties listed below.

1) System K matrices are formed at the beginning of each time interval.

2) The integration time step size is constant within each time interval.

3) Time-dependent material, fluid and gas properties are time averaged over each time interval. Temperature-dependent properties are evaluated at the temperatures computed at the beginning of each time interval.

4) Time-dependent mass-transport rates are averaged over each time interval.

Time intervals should be sized according to a) the time rates of change of material, fluid, and gas properties and mass-transport rates and b) the expected rates of change with respect to temperature of material, fluid, and gas properties.

The normal mode of operation is to let TRTA determine the time intervals from the times contained in data sets PROP TIME, COEF TIME, and MTR TIME (see Sections 2.2 and 6.1). However, if material, fluid, or gas properties are temperature dependent, or if the time points contained in the afore-mentioned data sets are too widely spaced to yield accurate solutions, the user may define additional time intervals with the TIME command as described in Section 6.2.3.2. To avoid unnecessary K matrix computations, the user should use the TIME command with great care because system K matrices will be formed at the times specified by the user in addition to those times contained in PROP TIME, COEF TIME, and MTR TIME not coincident with user defined times.

6.2.1.2 TRANSIENT ANALYSIS STRATEGY

Before attempting to execute TRTA, the user should read Sections 6.2.2 and 6.2.3 of the present volume and Section 4.3 of the SPAR Thermal Analysis Reference Manual, Volume 2 - Theory. It is suggested that the guidelines contained in this section be followed until the user has become sufficiently experienced in the use of TRTA to set up his or her own set of guidelines.

The first execution of TRTA should be

```
@XQT TRTA
RESET PREP=1
RESET T1= analysis starting time
RESET T2= analysis termination time (see below)
TEMP= initial temperature vector (see Section 6.2.3.1)
STOP
```

This will cause TRTA to compute and store, in a data set named NODA LAMB in the DEST library, the conduction+convection+mass-transport+radiation/capacity ratio for each node point in the finite element model. The maximum value (LAMBDA-MAX), the node point at which it occurs, and the critical integration time step size (DT-MAX=1.0/LAMBDA-MAX) will be printed online. Time-dependent material, fluid, and gas properties and time-dependent mass-transport rates will be averaged over the time interval defined by RESET controls T1 and T2. If T2 is not given, properties and mass-transport rates will be evaluated at time T1.

The data contained in NODA LAMB will enable the user to identify those nodes, which because of their very small capacities (or large conductances, etc.), may cause an unnecessarily small DT-MAX. Such nodes are called "arithmetic nodes". To remove the effects of arithmetic node capacities on the computations for DT-MAX, the user must execute AUS/TABLE to create a data set defining all arithmetic nodes:

```
TABLE(NI=1,NJ=n):ARIT NODE:J=1,n: j(1) j(2) j(3) . . . j(n)
```

If the user has created ARIT NODE, TRTA must be re-executed with RESET control PREP=1 so that LAMDA-MAX and DT=MAX will be recomputed from the LAMBDA's at non-arithmetic nodes.

In solving large problems, IO activity can be minimized by making use of the IO access information printed online when RESET control PREP=1. Specific examples are given for the explicit and implicit methods on the following pages.

Explicit Time Integration

During each execution of processor TRTA, the SOURCE library will be searched for ARIT NODE, and if it is found, DT-MAX will be computed only for the non-arithmetic nodes, and the temperatures at arithmetic nodes will be computed from the temperatures at non-arithmetic nodes as described in Section 4.3.2 of the SPAR Thermal Analysis Reference Manual, Volume 2 - Theory.

With the exception of some radiation problems, TRTA will compute stable solutions for all values of DT less than DT-MAX. Experience has shown that TRTA will compute accurate solutions in its default mode of operation of internally computing DT at the beginning of each time interval (see RESET controls IDT and FDT). However, if the user wishes to be sure that accurate solutions will be computed by TRTA, a suitable value for DT must be determined by trial and error. It is suggested that the user execute TRTA a few times over the same time span using different values of DT. Each execution must start with the same initial temperature vector (see the TEMP command, Section 6.2.3.1) and have RESET control IDT=0. Executions should be continued, each with a larger DT, until solutions begin to diverge from the solutions of previous executions.

If the user wishes to take advantage of the automated integration time step size computation capability of TRTA (see RESET controls IDT and FDT), the value for FDT should be the ratio of DT determined above to DT-MAX.

The effect of the number of terms in the Taylor series expansion (see RESET control NTERMS) can be investigated by executing TRTA with an integration time step size larger than that computed above for NTERMS=4,5, etc.

For problems in which there is no ARIT NODE data set, the number of IO accesses required to compute temperatures at each time step is (NTERMS-1)*(NK-1), where NK is the number of element K matrix blocks. When RESET control PREP=1, the K matrix block length (LK), the number of blocks (NK), and the length of the last block (LKLAST) are printed. This information can be used to minimize IO activity. For example, if LK=8620, NK=2, and LKLAST=2342, IO activity would be eliminated if the working data were increased by 2342. If NK were 3, the number of IO accesses per time step would be reduced by 1/3 if the working data space were increased by 1200. In general, if NK is greater than 2, TRTA may have to be executed several times with RESET control PREP=1 in order to determine a suitable data space size.

If an ARIT NODE data set exists, the number of IO accesses per time step is 2*(NTERMS-1)*(NK+NKAN-2), where NKAN is the number of arithmetic node K matrix blocks. If NKAN is too large, the user should redefine the ARIT NODE data set with the arithmetic nodes numbered to minimize the bandwidth of the arithmetic node K matrix.

To guard against runaway situations resulting from extremely small internally computed integration time step sizes, it is suggested that RESET control MXNDT be utilized. MXNDT should have a value slightly higher than the estimated number of integration time steps required for the analysis.

Implicit Time Integration

With BETA=.5, the implicit time integration method will yield stable solutions for all values of DT (except in the case of radiation exchange); however, a value for DT which will yield accurate solutions must be determined by trial and error in the same way as described for the explicit method.

DT must be input by the user; there is no automated computation of the integration time step size for the implicit method.

The integration time step size will be recomputed at each time contained in data sets SOUR TIME, CTEM TIME, and TEMP TIME and at the times specified by RESET control RI. If the computed value of the integration time step size does not agree with the DT input by the user, the K matrix will be refactored. To avoid unnecessary K matrix factorings, the user should insure that excitation times are either coincident with time interval end points or are multiples of DT.

The number of IO accesses required to compute temperatures at each time step is 2*(NK-1), where NK is the number of K matrix blocks. When RESET control PREP=1, the K matrix block size, the number of K matrix blocks, and the number of IO accesses per time step are printed. To minimize the number of blocks, the working data space should be as large as possible. If there are too many K matrix blocks, even with a very large working data space, the bandwidth should be minimized by creating a joint elimination sequence data set by executing TAB/JSEQ.

6.2.2 RESET CONTROLS

Name	Default Value	Meaning
SOURCE	1	Input library.
DEST	1	Output library.
ELIB	(SOURCE)	Library containing the TED Eij data sets.
QLIB	(SOURCE)	Thermal excitation library.
METHOD	EXPLICIT	Time integration method. METHOD=IMPLICIT will cause the implicit time integration method to be used.
T1 T2	0.0	Analysis starting and termination times. T2 must be specified.
DT		Integration time step size. If the explicit time integration method is used, DT can be input by the user, or it can be computed by TRTA (see RESET control IDT). If the implicit time integration method is used, DT must be input by the user. DT will be recomputed at the beginning of each time interval if the time interval size is not a multiple of DT.
MXNDT	1000	Maximum number of integration time steps permitted during TRTA execution.
PRINT	1	Online print control parameter: = 0, no printing. = 1, DT will be printed at at the beginning of each time interval. = 2, same as above, except LAMBDA-MAX and DT-MAX will also be printed (explicit method only).
PREP	0	Procedure parameter: = 0, proceed with analysis. = 1, LAMBDA-MAX, the critical integration time step size (DT-MAX), and an IC activity summary will be printed. An analysis will not be performed.
TERR	1	Error code in the table of contents line of each output data set if a normal termination is not achieved.

The following RESET controls are applicable only if the explicit time integration method is used:

IDT	1	DT computation parameter: = 0, use the DT input with RESET control DT. = 1, compute DT at the beginning of each time interval (see RESET control FDT). If DT was input with RESET control DT, use the smaller of the input DT or the computed DT. = 2, compute and use a new DT at the beginning of each time interval. Ignore any value input with RESET control DT.
FDT	.95	DT computation factor, DT=FDT/LAMBDA-MAX.
NTERMS	3	Number of terms to use in the Taylor series expansion. NTERMS must not be less than 2 nor greater than 10 (5 for radiation problems).
ZCAP	1.0E-20	Singular capacity parameter. If any term in the capacity matrix is less than ZCAP, execution will be terminated, unless RESET control PREP=1.
NCAP	10	Number of singular capacity terms to print.

The following RESET control is applicable only if the implicit time integration method is used:

BETA	.5	Weighted residual parameter (see Section 4.3.3 of the SPAR Thermal Analysis Reference Manual, Volume 2 - Theory). BETA must be greater than zero.

6.2.2 RESET CONTROLS (continued)

FINEFF	0	Fin efficiency computation parameter for MT62 and C62 elements: = 0, the fin efficiency for all MT62 and C62 elements is 1. = 1, the fin efficiency for each MT62 and C62 element will be computed from the formula given in Section 3.1 - element type C62.
FLUX	0	Heat flux computation parameter (see Section 6.2.6.2): = 0, element fluxes will not be computed. = 1, element fluxes will be computed from the temperatures contained in each block of an existing TRTA TEMP data set. = 2, element fluxes will be computed after the solution is finished.
PDROP	0	Pressure drop and outlet pressure computation parameter for mass-transport elements (see Section 6.2.6.3): = 0, element pressures will not be computed. = 1, element pressures will be computed from existing TRTA TEMP and TRTA TIME data sets. = 2, element pressures will be computed after the solution is finished.
PZERO	1.0E-10	Zero inlet pressure parameter. If an element's inlet pressure is less than PZERO, it will be set equal to the outlet pressure of the element with the next lowest index number within the element group.
G	1.0	Acceleration of gravity (force/mass).
ALPHA		Mass-transport matrix computation parameter (see Section 3 of the SPAR Thermal Analysis Reference Manual, Volume 2 -Theory). If ALPHA is specified, its value will be used to compute the mass-transport matrix for all MT21, MT42, and MT62 elements. If ALPHA is not specified, it will be computed for each element according to the formula given in the above reference.

The following RESET controls are applicable only when the finite element model contains radiation elements.

SBCON	.56697E-7	Stefan-Boltzmann constant, the default value is expressed in SI units.
RLIB		Source library for REX data sets. RLIB must be specified for radiation exchange problems.
NREX	1	Number of REX data sets. Exchange factors will be obtained from data sets REX MASK 1 through REX MASK nrex (see Section 5.1.6). NREX should be 1 for the most efficient computation of radiation load vectors.
RSTEPS	3	Maximum number of steps allowed during computation of the radiation load vector.
RCONV	.01	Radiation load vector convergence parameter. Convergence of the radiation load vector is assumed when QR/QE is less than RCONV, where QR is the sum of the heat reflected by all elements during the current step, and QE is the total heat emmitted by all elements.
RCC	0	Radiation convergence action parameter: = 0, execution will be terminated if convergence is not achieved during the computation of a radiation load vector. = 1, execution will be continued, even if convergence is not achieved. = 2, same as 1 above, except a warning message will be printed each time convergence is not achieved.
RDIST	1	Reflected heat energy distribution parameter: = 0, residual reflected heat energy will be lost to space. = 1, residual reflected heat energy will be distributed to all radiation elements in proportion to the total heat absorbed by each element.
RI		Radiation load vectors will be computed at intervals of RI. If RI is not specifed, load vectors will be computed at the beginning of each time interval (see Section 6.2.1.1).

The following RESET controls are applicable if the implicit time integration method is used, or if the explicit method is used and arithmetic nodes are present.

KPRT	1	Print control parameter used during factoring of the K matrix: = 0, no printing. = 1, singularity messages will be printed. = 2, negative root messages will be printed. = 3, singularity and negative root messages will be printed.
NSING	0	Number of singularities allowed during factoring of the K matrix before error termination. Singular equations detected during factoring will be removed from the system and will have no effect on the solution process.
SING	1.0E-6	Singularity parameter. If the absolute value of a factored diagonal term is less than or equal to SING times the absolute value of the diagonal term before factoring, the matrix is singular.
ZERO	1.0E-20	Singularity parameter. If the absolute value of a factored diagonal term is less than or equal to ZERO, the matrix is singular.

The following RESET controls are applicable only when radiation elements are connected to arithmetic nodes:

NFACS	2	Number of refactorings allowed after each load vector computation.
NITER	10	Maximum number of iterations allowed, after each factoring, during the computation of each term in the Taylor series representation of the temperature at each arithmetic node.
ANCONV ANFAC	.01 .01	Convergence parameters. Convergence is assumed if ABS(DTN/TN) is less than or equal to ANCONV for each term in the Taylor series. DTN is the change in the value of the n-th term and TN is the value of the n-th term. Convergence is checked only for those terms whose percentage contribution to an arithmetic node temperature is greater than ANFAC.

ANCC	0	Arithmetic node convergence action parameter: = 0, execution will be terminated if there is no convergence. = 1, execution will continue if there is no convergence.
ANPRT	0	Arithmetic node convergence print parameter: = 0, no printing. = 1, convergence information will be printed.

The following RESET controls are applicable only if the implicit time integration method is used to solve radiation problems:

NFACS	2	Number of refactorings allowed after each load vector computation.
NITER	10	Maximum number of iterations allowed, after each factoring, during the computation of the temperature vector.
CONVERGE CUTOFF	.01 .2	Convergence parameters. Convergence is assumed if ABS(DT/T) is less than or equal to CONVERGE for all nodal temperatures greater than CUTOFF*Tave. DT is the change in nodal temperature, T is the nodal temperature, and Tave is the average of all nodal temperatures.
IMCC	0	Implicit method convergence action parameter: = 0, execution will be terminated if there is no convergence. = 1, execution will continue if there is no convergence.
IMPRT	0	Implicit method convergence print parameter: = 0, no printing. = 1, convergence information will be printed.

6.2.3 EXECUTION COMMANDS

The execution commands described below, if present, must follow the last RESET control.

6.2.3.1 Initial Temperature Distribution - The TEMP Command

The TEMP command specifies an initial temperature distribution in either of two ways:

TEMP= T or TEMP= lib N1 N2 n3 n4 nb

The first command specifies a uniform temperature T at every node in the finite element model. The second command causes the the initial temperature distribution to be obtained from block nb of the named data set. If lib is omitted, the data set is assumed to reside in the SOURCE library. There is no default data set name; N1 must be specified. The default block is the last block in the data set.

If the TEMP command is not used, the DEST library is searched for TRTA TEMP 1 1, and the contents of the last block are used as the initial temperature distribution. If TRTA TEMP 1 1 is not found, an error termination will result.

6.2.3.2 Definition of Time Intervals - The TIME Command

The TIME command allows the user to define time intervals in either of two ways:

TIME= TI or TIME= lib N1 N2 n3 n4

The first command defines a constant time interval size. System K matrices will be formed at time increments of TI starting at the time defined by RESET control T1.

The second command identifies a data set containing a list of times at which system K matrices will be formed. If lib is omitted, it is assumed that the data set resides in the SOURCE library. There is no default data set name; N1 must be given. The times in the data set must bracket the analysis starting and termination times.

6.2.3.3 Definition of Data Retention Times - The TSAVE command

The TSAVE command is used to define the times at which results are to be stored on mass storage:

TSAVE= TS or TSAVE= lib N1 N2 n3 n4

The first command will cause results to be stored at time increments of TS. The second command will cause results to be stored at those times, contained in the named data set, bracketed by the analysis starting and termination times. If lib is omitted, it is assumed that the data set resides in the SOURCE library. There is nc default data set name; N1 must be specified.

The TSAVE command must be present in the input runstream. The times at which results are stored during each execution of TRTA are contained in data set TRTA TIME 1 1 (see Section 6.2.6).

6.2.3.4 Radiation Heat Flux Computations - The RFLUX Command

RFLUX= lib, list of options (E, H, B, QE, QA, QR)

causes the indicated radiation flux quantities to be computed for all radiation elements from temperatures contained in an existing TRTA TEMP 1 1 data set. Data sets containing the flux quantities are described in Section 6.2.6.2. If lib is omitted, each data set will reside in the DEST library. If no options are given, all flux quantities will be computed.

E = Emissive power
H = Incident heat flux (irradiation)
B = Radiosity
QE = Heat emitted
QA = Heat absorbed
QR = Heat reflected

6.2.4 CENTRAL MEMORY REQUIREMENTS

For the explicit method, the required data space is approximately

$$3000 + (4+NTERMS)*NJTS + LR + LK ,$$

where NTERMS is the number of terms in the Taylor series expansion, NJTS is the number of joints, and LR is zero if there are no radiation elements or NJTS if there are radiation elements. LK is the K matrix block length and is set equal to the remaining data space after all other data space requirements are satisfied. Each K matrix block is a sequential list of element conduction, fluid-surface convective exchange, and mass-transport matrices. When arithmetic nodes are present, up to one half of LK will be allocated to each arithmetic node K matrix block.

For the implicit method, the required data space is approximately

$$3000 + 3*NJTS + LK + L$$

where NJTS is the number of joints, and L is the larger of LK, 5*NJTS (no radiation elements), or 7*NJTS (radiation elements present). LK is the assembled K matrix block size and is set equal to one half of the remaining data space after all other data space requirements have been satisfied.

6.2.5 ERROR MESSAGES

If a fatal error condition is detected during program execution, the message NERR, N = XXXX, n will be printed, and execution will be terminated. Error messages and their causes are listed below.

NERR	N	Meaning
CORE	n	Allowable data space exceeded, increase core size by n.
TACD	0	As indicated by printed message.
TRTA	0	As indicated by printed message.
TRTA	1	No such execution command.
TRTA	2	Incorrect TEMP command.
TRTA	3	The named data set has a block length less than 2, has a negative time, has improperly sequenced times, or does not bracket the analysis starting and termination times.
TRTA	4	The length of the named data set does not agree with the number of elements in the corresponding TED Eij ngrp data set.
TRTA	5	The number of blocks in the named data set does not agree with the length of the corresponding XXXX TIME data set.
TRTA	6	Data sets TEMP NODE and APPL TEMP are incompatible.
TRTA	7	The number of blocks in TRTA TEMP does not agree with the length of TRTA TIME.
TRTA	9	RESET control METHOD has the wrong image.
TRTA	10	BETA is less than or equal to zero.
TRTA	11	T2 was not specified or is less than or equal to T1.
TRTA	12	Initial temperature data set has incorrect length.
TRTA	13	Initial temperature data set has too few blocks.
TRTA	14	RESET controls FLUX and PDROP are incompatible.
TRTA	15	NTERMS is less than 2 or greater than 10 (5 if radiation elements are present).
TRTA	16	NJ of data set TRTA TEMP is not equal to the number of joints in the finite element model.
TRTA	17	Incorrect TIME command.
TRTA	18	TI is greater than T2-T1 (see Section 6.2.3.2).
TRTA	19	DT is less than or equal to zero.
TRTA	20	Incorrect RFLUX command.
TRTA	21	Conflict between RESET control FLUX or PDROP and the RFLUX command.
TRTA	22	RESET control RLIB was not used with command RFLUX.
TRTA	23	Incorrect or missing TSAVE command.
TRTA	24	TS is greater then T2-T1 (see Section 6.2.3.3).

6.2.6 OUTPUT DATA SET CONTENTS

Results are stored at the times contained in TRTA TIME 1 1 (see Section 6.2.3.3).

For each of the transient output data sets described below, the k-th block corresponds to the k-th time in TRTA TIME 1 1. Within each block, the j-th line contains data corresponding to the j-th node or element, as appropriate. All data sets reside in the DEST library unless noted otherwise.

6.2.6.1 Nodal Temperatures

Nodal temperatures are stored in the following data set. TOC parameter NJ=the number of nodes in the finite element model.

Data Set Name	NI	Line Contents
TRTA TEMP 1 1	1	Nodal temperature

6.2.6.2 Element Heat Fluxes

The data sets tabulated below are generated when RESET control FLUX=1 or FLUX=2. For each data set, TOC parameter NJ=the number of elements in the corresponding TED Eij ngrp data set.

Data Set Name		NI	Line Contents
TFLX K21	ngrp 1	1	Fx
TFLX K31	ngrp 1	2	Fx Fy
TFLX K41	ngrp 1	2	Fx Fy
TFLX K61	ngrp 1	3	Fx Fy Fz
TFLX K81	ngrp 1	3	Fx Fy Fz
TFLX C21	ngrp 1	1	Fc
TFLX C31	ngrp 1	1	Fc
TFLX C41	ngrp 1	1	Fc
TFLX C32	ngrp 1	1	Fs
TFLX C42	ngrp 1	1	Fs
TFLX C62	ngrp 1	2	Fs Eff
TFLX MT21	ngrp 1	2	Fm Fk
TFLX MT42	ngrp 1	4	Fm Fk Fs Fx
TFLX MT62	ngrp 1	8	Fm Fk Fs Fw1 Fw2 Fx Fy Eff
TFLX R21	ngrp 1 *	1	E
TFLX R31	ngrp 1 *	1	E
TFLX R41	ngrp 1 *	1	E

* These data sets are created only when no radiation exchange factors (Section 5.1.6) have been defined.

In the element flux data sets:

Fx, Fy, Fz = conductive fluxes in x, y, and z directions.
Fc = convective flux, positive leaving element.
Fs = surface-fluid convective exchange flux, positive entering fluid.
Eff = fin efficiency.
Fm = fluid mass-transport flux.
Fk = fluid conductive flux.
Fw1, Fw2 = top and bottom wall fluid-surface convective exchange fluxes, positive entering fluid.
E = emissive power.

The following data sets are generated when execution command RFLUX is used. For each data set, TOC parameter NJ=the total number of radiation elements.

Data Set Name	NI	Line Contents
TREX E 1 1	1	Emissive power
TREX E 1 1	1	Incident radiant heat flux (irradiation)
TREX B 1 1	1	Radiosity
TREX QE 1 1	1	Heat emitted
TREX QA 1 1	1	Heat absorbed
TREX QR 1 1	1	Heat reflected

6.2.6.3 Element Fluid Pressures

The data sets tabulated below are generated when RESET control PDROP=1 or PDROP=2. For each data set, TOC parameter NJ=the number of elements in the corresponding TED Eij data set.

Data sets containing element inlet pressures must exist before pressures can be computed (see Section 5.1.5).

Data Set Name	NI	Line Contents
TPD MT21 ngrp 1	4	P1, P2, PDF, PDA
TPD MT42 ngrp 1	4	"
TPD MT62 ngrp 1	4	"

In the above:

P1 = inlet pressure.
P2 = outlet pressure.
PDF = friction pressure drop.
PDA = flow acceleration pressure drop.

Reference Manual

SPAR Thermal Analysis Processors

Volume 2

Theory

CONTENTS

Section 1

BASIC EQUATIONS AND DEFINITIONS

The pertinent equations of heat transfer for solid and fluid media are tabulated below.

Heat flux at a point on the surface of an anisotropic solid (Reference 1):

$$q = -k_{ij} \frac{\partial T}{\partial x_j} n_i \quad . \tag{1-1}$$

Heat flow in an anisotropic solid (Reference 1):

$$\frac{\partial}{\partial x_i} \left(k_{ij} \frac{\partial T}{\partial x_j}\right) + \phi - \rho c \dot{T} = 0 \quad . \tag{1-2}$$

Heat flux crossing a fluid surface (Reference 1):

$$q = -k \frac{\partial T}{\partial x_i} n_i \quad . \tag{1-3}$$

Heat flow in an incompressible fluid with heat dissipation terms neglected (References 2, 3):

$$k \frac{\partial^2 T}{\partial x_i \partial x_i} - \rho c v_i \frac{\partial T}{\partial x_i} - \rho c T = 0 \quad . \tag{1-4}$$

In the above equations, repeated subscripts are summed from 1 to 3, and

q = surface heat flux,
k_{ij} = conductivity coefficients ($k_{ij} = k_{ji}$),
T = temperature,
x_i = coordinate ($i = 1, 2, 3$),
n_i = components of unit outer normal to the surface,

---

(1) Carslaw, H. S. and J. C. Jaeger, Conduction of Heat in Solids, Oxford Press, 1959.

(2) Knudsen, James G. and Donald L. Katz, Fluid Dynamics and Heat Transfer, McGraw Hill, 1958.

(3) Thornton, Earl A. and Allan R. Wieting, Finite Element Methodology for Thermal Analysis of Convectively Cooled Structures, AIAA Paper No. 77- 187.

ϕ = source heat rate,

ρ = mass density,

c = specific heat,

T = temperature time derivative,

k = fluid conductivity,

v_i = fluid velocity component.

The heat flux leaving a point on the surface bounding a solid or fluid region is assumed to be comprised of convective and radiative parts, i.e.,

$$q = h(T - T_h) + \varepsilon\sigma T^4 - aH \quad . \tag{1-5}$$

In the above equation

h = convection coefficient,

T_h = known convective exchange temperature,

ε = emissivity,

σ = Stefan-Boltzmann constant,

a = absorptivity,

H = incident radiant heat flux (irradiation), computed as described in Section 4.1.

At solid/fluid interfaces, the flux leaving the solid (or entering the fluid) is assumed to be

$$q = h_f (T_{solid} - T_{fluid}), \tag{1-6}$$

where h_f is the fluid convective exchange coefficient (Reference 1).

---

(1) Thornton, Earl A. and Allan R. Wieting, Finite Element Methodology for Thermal Analysis of Convectively Cooled Structures, AIAA Paper No. 77- 187.

Section 2

THE FINITE ELEMENT METHOD

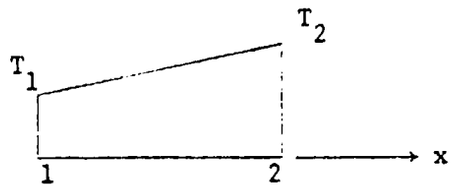
In the finite element method of analysis, a continuum is modeled as an assemblage of discreet elements (typically, straight line elements, triangular and quadrilateral area elements, and pentahedral and hexahedral volume elements) connected to each other at a finite number of node points. The matrix equations of heat transfer are developed for a single element in Section 2.1. Element matrices and vectors are defined in Section 2.2. The assembly process is described in Section 2.3.

2.1 THE MATRIX EQUATIONS FOR A SINGLE FINITE ELEMENT

In the following, the Galerkin method is used to develop the matrix equations of heat transfer for a single element. The method is well known (Reference 1) and will be described here only briefly. Within an element, an approximate solution to the differential equation $L(T,\dot{T}) = 0$ is represented by the finite series

$$T \simeq \sum_{n=1}^{N} f_n T_n \quad , \tag{2-1}$$

where the f_n's are arbitrary functions of the element coordinates, the T_n's are unknown nodal temperatures, and N is the number of element nodes. The temperature distribution in the element is dependent only on the temperatures at the element node points, for example, the linear temperature distribution shown below for a 2-node element is represented by $f_1 = 1-x$ and $f_2 = x$ in the interval $0 \leq x \leq 1$.



Equation 2.1 is substituted into the differential equation to yield the error function

---

(1) Chung, T. J., Finite Element Analysis in Fluid Dynamics, McGraw Hill, 1978.

$$L\left(\sum_{n=1}^{N} f_n T_n, \sum_{n=1}^{N} f_n \dot{T}_n\right) = E(x_1, x_2, x_3, t) \quad ,$$

which is minimized by assuming that it is orthogonal to each of the f functions, i.e.,

$$\int_V L\left(\sum_{n=1}^{N} f_n T_n, \sum_{n=1}^{N} f_n \dot{T}_n\right) f_m \, dv = 0 \quad . \tag{2-2}$$

When the above integration is carried out for all f functions, the result is the matrix equation

$$KT + C\dot{T} = F \quad ,$$

where K and C are NxN matrices, F is a vector of length N, T is the vector of nodal temperatures, and $\dot{T}$ is the vector of nodal temperature time derivatives.

To apply the Galerkin method to the heat transfer problem, we consider the solid region (Equation 1-2) and write Equation 2-2 as

$$\int_{V_s} \{\frac{\partial}{\partial x_i}(k_{ij}\frac{\partial T}{\partial x_j}) + \Phi - \rho c\dot{T}\} f_m \, dv = 0 \quad ,$$

integrate the first term by parts, apply the Divergence Theorem, and use Equations 1-2, 1-5, and 1-6 to obtain

$$\int_{V_s} \{k_{ij}\frac{\partial T}{\partial x_j}\frac{\partial f_m}{\partial x_i} - \Phi f_m + \rho c\dot{T} f_m\} \, dv + \int_{S_{sh}} h(T-T_h) f_m \, dA$$

$$+ \int_{S_{sr}} (\sigma\varepsilon T^4 - aH) f_m \, dA + \int_{S_{sf}} h_f (T - T_{fluid}) f_m \, dA = 0 \quad . \tag{2-3}$$

In the above equation, V_s is the volume of the solid region, S_{sh} are those surfaces where convective exchange temperatures are known, S_{sr} are those surfaces which emit and/or receive radiant heat energy, and S_{sf} are those surfaces in contact with the fluid.

The equation for heat transfer in the fluid region (Equation 1-4) is manipulated in the same fashion as above to yield

$$\int_{V_s} \{k \frac{\partial T}{\partial x_i} \frac{\partial f_m}{\partial x_i} + \rho c v_i \frac{\partial T}{\partial x_i} f_m + \rho c \dot{T} f_m\} \, dv + \int_{S_{fh}} h(T-T_h) f_m \, dA$$

(2-4)

$$+ \int_{S_{fr}} (\sigma \varepsilon T^4 - aH) f_m \, dA + \int_{S_{fs}} h_f (T-T_{solid}) f_m \, dA = 0$$

The integrations are carried out over the fluid volume and surfaces corresponding to those described above for the solid region.

Substitution of the approximate expression for the temperature (Equation 2-1) into Equations 2-3 and 2-4 yields the matrix equation

$$(K_k + K_h + K_r + K_m)T + C\dot{T} = Q + H + R \quad . \tag{2-5}$$

In the above equation

K_k = conduction matrix,
K_h = convection matrix,
K_r = radiation matrix,
K_m = fluid mass-transport matrix,
C = capacity matrix,
Q = source load vector,
H = convection load vector,
R = radiation load vector.

Specific terms in each matrix and vector are determined by matching coefficients. For example, the components of the capacity matrix are

$$C_{mn} = \int_V \rho c f_m f_n \, dv \quad .$$

2.2 ELEMENT MATRIX AND VECTOR DEFINITIONS

The element matrix and vector components given in this section can be deduced directly from Equations 2-3, 2-4, and 2-5. The temperature distribution in each element is assumed to be given by Equation 2-1, i.e.,

$$T = \sum_{n=1}^{N} f_n T_n \quad .$$

The shape functions f_n are defined for each element type in Section 3.

2.2.1 Conduction Matrices

The conduction matrix for a solid element has components

$$K_{mn} = \int_V k_{ij} \frac{\partial f_m}{\partial x_i} \frac{\partial f_n}{\partial x_j} \, dv \quad , \tag{2-6}$$

and the conduction matrix for a fluid element has components

$$K_{mn} = \int_V k \frac{\partial f_m}{\partial x_i} \frac{\partial f_n}{\partial x_i} \, dv \quad . \tag{2-7}$$

2.2.2 Convection Matrices

There are two types of element convection matrices—those which describe convection to a known convective exchange temperature (Equation 1-5) and those which describe convective heat exchange at solid/fluid interfaces (Equation 1-6).

The components of an element matrix defining convection to a known temperature are

$$K_{mn} = \int_S h f_m f_n \, dA \quad , \tag{2-8}$$

where the integration is carried out over that portion of the element's surface where convective exchange temperatures are known.

The components of an element matrix defining solid/fluid convective heat exchange are

$$K_{mn} = \omega \int_S h_f f_m f_n dA \qquad (2\text{-}9)$$

ω = +1, when both m and n are solid or fluid nodes.
= -1, when m and n are different node types.

The integration is carried out over that portion of the element's surface where solid and fluid are in contact.

2.2.3 Radiation Matrix

Element radiation matrices have components

$$K_{mn} = \int_S \varepsilon\sigma f_m f_n T^3 \, dA \quad , \qquad (2\text{-}10)$$

The integration is carried out over that portion of the element's surface which emits radiant heat energy.

Radiation matrices are assumed to be diagonal for steady-state and transient analyses, i.e.,

$$\begin{aligned} K_{mn} &= w_m T_m^3 \quad , m = n \\ &= 0 \quad , m \neq n. \end{aligned} \qquad (2\text{-}11)$$

The w_m's are weighting factors and are determined by requiring that the total radiant heat energy emitted by the element be the same for both diagonal and non-diagonal matrices. The heat emitted by the element is

$$\text{Heat emitted} = \int_S \sigma\varepsilon T^4 \, dA \quad ,$$

which can be written in terms of the radiation matrix by utilizing the relationship

$$\sum_{n=1}^{N} f_n = 1 \, ^* \quad , \qquad (2\text{-}12)$$

---

*This equation can be deduced from Equation 2-1 by noting that if an element's node points all have the same temperature, the temperature everywhere in the element must be the same.

multiplying both sides of Equation 2-10 by T_n, and summing on both m and n to obtain

$$\int_S \varepsilon\sigma T^4 \, dA = \sum_{m=1}^{N} \sum_{n=1}^{N} K_{mn} T_n \quad .$$

Substituting Equation 2-11 into the above equation yields

$$\int_S \varepsilon\sigma T^4 \, dA = \sum_{m=1}^{N} w_m T_m^4 \quad .$$

The only way that this equation can be satisfied for all nodal temperatures T_m is to assume that T^4 varies according to

$$T^4 = f_1 T_1^4 + f_2 T_2^4 + \ldots,$$

which leads to

$$F_{mm} = T_m^3 \int_S \varepsilon\sigma f_m \, dA \quad . \tag{2-13}$$

2.2.4 Mass-Transport Matrix

The mass-transport matrix for a fluid element is an asymmetric matrix with components

$$K_{mn} = \int_V \rho c v_i g_m \frac{\partial g_n}{\partial x_i} \, dv \quad . \tag{2-14}$$

The integration is carried out over the volume of the fluid. For one-dimensional flow in the x direction, the above equation reduces to

$$K_{mn} = \dot{m} \int_0^L c g_m \frac{\partial g_n}{\partial x} \, dx \quad , \tag{2-15}$$

where $\dot{m}$ is the mass-transport rate (constant) defined by

$$\dot{m} = \int_A \rho v dA \quad . \tag{2-16}$$

The integration is carried out over the flow area of the element.

2.2.5 Capacity Matrix

The components of the capacity matrix are

$$C_{mn} = \int_V \rho c f_m f_n \, dv \quad . \tag{2-17}$$

For transient analyses, the capacity matrix is assumed to be diagonal. The components of the diagonal capacity matrix are evaluated by requiring that the heat absorption rate for an element be the same for diagonal and non-diagonal matrices. In a manner analogous to that used to derive the components of a diagonal radiation matrix (Section 2.2.3), the components of the diagonal capacity matrix are found to be

$$C_{mm} = \int_V \rho c f_m \, dv \quad . \tag{2-18}$$

2.2.6 Source Load Vector

The components of the source load vector are

$$Q_m = \int_V \phi f_m \, dv \quad . \tag{2-19}$$

2.2.7 Convection Load Vector

The components of the convection load vector are

$$H_m = \int_S h f_m T_h \, dA \quad . \tag{2-20}$$

The integration is carried out over that portion of the element's surface where the convective exchange temperature is known.

2.2.8 Radiation Load Vector

The components of the radiation load vector are

$$R_m = \int_S a H f_m \, dA \quad . \tag{2-21}$$

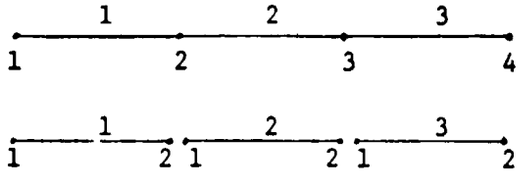
The computation of the irradiation H is described in Section 4.1.

2.3 MATRIX EQUATIONS FOR THE ASSEMBLED SYSTEM OF FINITE ELEMENTS

The global system of equations is (Equation 2-5)

$$KT + C\dot{T} = F \quad , \tag{2-22}$$

where K is the assemblage of element conduction, convection, mass-transport, and radiation matrices, C is the assemblage of element capacity matrices, F is the assemblage of element source, convection, and radiation load vectors, and T is the global temperature vector. The assembly process is described briefly in this section with the aid of the simple finite element model shown below.



complete model

finite elements

The matrix equation for each element is

$$K^i T^i + C^i \dot{T}^i = F^i \quad , \tag{2-23}$$

where the i superscript identifies element i. For example, the equations for element 1 in the above finite element model are

$$K^1_{11} T^1_1 + K^1_{12} T^1_2 + C^1_{11} \dot{T}^1_1 = F^1_1 \tag{2-24}$$

$$K^1_{21} T^1_1 + K^1_{22} T^1_2 + C^1_{22} \dot{T}^1_2 = F^1_2 \quad ,$$

and the equations for element 2 are

$$K^2_{11} T^2_1 + K^2_{12} T^2_2 + C^2_{11} \dot{T}^2_1 = F^2_1 \tag{2-25}$$

$$K^2_{21} T^2_1 + K^2_{22} T^2_2 + C^2_{22} \dot{T}^2_2 = F^2_2 \quad .$$

Whenever two or more elements are joined at a node, the global equation for that node is formed by summing the appropriate element equations (References 1 and 2).

---

(1) Chung, T. J., Finite Element Analysis in Fluid Dynamics, McGraw Hill, 1978.

(2) Tong, P. and J. N. Rossettos, Finite Element Method, MIT Press, 1977.

For example, elements 1 and 2 are joined at node 2; hence, the second of Equations 2-24 must be added to the first of Equations 2-25, i.e.,

$$K_{21}^1 T_1 + (K_{22}^1 + K_{11}^2)T_2 + K_{12}^2 T_3 + (C_{22}^1 + C_{11}^2)T_2 = F_2^1 + F_1^2 \quad . \qquad (2\text{-}26)$$

The above equation has been expressed in terms of temperatures at the global nodes by noting that

$$T_1^1 = T_1 \qquad T_2^1 = T_2 \qquad T_1^2 = T_2 \qquad T_2^2 = T_3 \quad . \qquad (2\text{-}27)$$

The global equation for node 3 is determined in a similar fashion as the global equation for node 2. The global K and C matrices and F vector for the complete finite element model are

$$K = \begin{bmatrix} K_{11}^1 & K_{12}^1 & 0 & 0 \\ K_{21}^1 & K_{22}^1+K_{11}^2 & K_{12}^2 & 0 \\ 0 & K_{21}^2 & K_{22}^2+K_{11}^3 & K_{12}^3 \\ 0 & 0 & K_{21}^3 & K_{22}^3 \end{bmatrix} \qquad (2\text{-}28a)$$

$$C = \begin{bmatrix} C_{11}^1 & & & \\ & C_{22}^1+C_{11}^2 & & \\ & & C_{22}^2+C_{11}^3 & \\ & & & C_{22}^3 \end{bmatrix} \qquad (2\text{-}28b)$$

$$F = \begin{bmatrix} F_1^1 \\ F_2^1+F_1^2 \\ F_2^2+F_1^3 \\ F_2^3 \end{bmatrix} \qquad (2\text{-}28c)$$

Note that terms K_{ij} $(i \neq j)$ occur only when an element connects nodes i and j, e.g. in the above K matrix, there is no K_{41} term because there is no element that connects nodes 1 and 4.

To generalize the assembly process to a system of n elements, each element temperature vector is expressed in terms of the global temperature vector by

$$T^i = D^i T \qquad (2\text{-}29)$$

and the assembled K, C, and F become

$$K = \sum_{i=1}^{n} (D^i)^T K^i D^i \qquad (2\text{-}30)$$

$$C = \sum_{i=1}^{n} (D^i)^T C^i D^i \qquad (2\text{-}31)$$

$$F = \sum_{i=1}^{n} (D^i)^T F^i \quad . \qquad (2\text{-}32)$$

The terms in the distribution matrix D^i are either 1 or 0. For example, the distribution matrices for the 3 element model are

$$D^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad D^2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad D^3 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad .$$

Substitution of the above expressions into Equations 2-30, 2-31, and 2-32 will yield Equations 2-28.

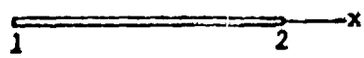
Section 3

ELEMENT FORMULATIONS

Element matrix and load vector formulations for the element types tabulated below are computed from the equations contained in Section 2.2. All formulations are based on the assumptions that geometry parameters (areas, thicknesses, etc.), prescribed thermal excitation quantities (source heat rates, convective exchange temperatures, etc.), and material and/or fluid properties are constant within each element. Temperature-dependent properties are computed at the average element temperature.

Name	Type
	Conduction:
K21	2 node line element
K31	3 node area element
K41	4 node area element
K61	6 node volume element
K81	8 node volume element
	Convection:
C21	2 node line element
C31	3 node area element
C41	4 node area element
	Fluid-Surface Convective Exchange:
C32	3 node line element
C42	4 node line element
C62	6 node area element
	Mass-transport:
MT21	2 node line element
	Integrated Mass-transport, Convective Exchange:
MT42	4 node line element
MT62	6 node area element
	Radiation:
R21	2 node line element
R31	3 node area element
R41	4 node area element

3.1 K21 - 2-NODE LINE CONDUCTION ELEMENT



The assumed temperature distribution is $T = f_1T_1 + f_2T_2$, where $f_1 = 1 - x/L$ and $f_2 = x/L$. Substitution of these expressions into Equations 2-6, 2-15, and 2-16 gives the conduction matrix, capacity matrix, and source load vector:

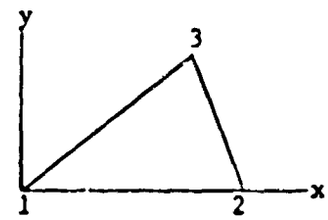
$$K = k_{xx}\frac{A}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

$$C = \frac{1}{2}\rho cL\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$Q = \frac{1}{2}\phi AL\begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad .$$

In the above equations, A is the cross-sectional area, and L is the length.

3.2 K31 - 3-NODE AREA CONDUCTION ELEMENT



The assumed temperature distribution is $T = f_1T_1 + f_2T_2 + f_3T_3$, where

$$f_1 = 1 - ax - (1-d)by$$

$$f_2 = ax - bdy$$

$$f_3 = by$$

The conduction matrix can be evaluated in closed form. From Equation 2-6, the components are

$$K_{11} = At\{a^2k_{xx} + (1-d)^2b^2k_{yy} + 2ab(1-d)k_{xy}\}$$

$$K_{21} = At\{-a^2k_{xx} + (1-d)db^2k_{yy} - ab(1-2d)k_{xy}\}$$

$$K_{22} = At\{a^2k_{xx} + d^2b^2k_{yy} - 2abdk_{xy}\}$$

$$K_{31} = At\{-(1-d)b^2k_{yy} - abk_{xy}\}$$

$$K_{32} = At\{-db^2k_{yy} + abk_{xy}\}$$

$$K_{33} = At\,b^2k_{yy} \quad .$$

In the above equations

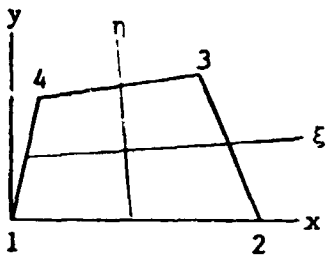
$$a = \frac{1}{x_2} \qquad b = \frac{1}{y_3} \qquad d = \frac{x_3}{x_2} \qquad A = \frac{1}{2}x_2y_3 \text{ (area)} \qquad t = \text{thickness.}$$

From Equations 2-18 and 2-19, the capacity matrix and source load vector are

$$C = \frac{1}{3}\rho cAt\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$$

$$Q = \frac{1}{3}\dot{q}At\begin{bmatrix}1\\ 1\\ 1\end{bmatrix} \quad .$$

3.3 K41 - 4-NODE AREA CONDUCTION ELEMENT



An isoparametric formulation is used to compute the conduction matrix, capacity matrix, and source load vector. The coordinates and temperature are assumed to vary within the element according to

$$x = \sum_{n=1}^{4} f_n x_n \qquad y = \sum_{n=1}^{4} f_n y_n \qquad T = \sum_{n=1}^{4} f_n T_n \quad .$$

The shape functions are

$$f_1 = \frac{1}{4}(1-\xi)(1-\eta) \qquad f_2 = \frac{1}{4}(1+\xi)(1-\eta)$$

$$f_3 = \frac{1}{4}(1+\xi)(1+\eta) \qquad f_4 = \frac{1}{4}(1-\xi)(1+\eta) \quad ,$$

where the quantities ξ and η take values between -1 and 1. Partial derivatives with respect to x and y are

$$\frac{\partial}{\partial x} = \frac{1}{J}\left(\frac{\partial y}{\partial \eta}\frac{\partial}{\partial \xi} - \frac{\partial y}{\partial \xi}\frac{\partial}{\partial \eta}\right) \qquad \frac{\partial}{\partial y} = \frac{1}{J}\left(\frac{\partial x}{\partial \xi}\frac{\partial}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial}{\partial \xi}\right) \quad ,$$

where J is the Jacobian defined by

$$J = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi} \quad .$$

The components of the conduction matrix, capacity matrix, and the source load vector are (Equations 2-6, 2-18, 2-19)

$$K_{mn} = t \int_{-1}^{1}\int_{-1}^{1} \{k_{xx} \frac{\partial f_m}{\partial x} \frac{\partial f_n}{\partial x} + k_{yy} \frac{\partial f_m}{\partial y} \frac{\partial f_n}{\partial y}$$

$$+ k_{xy}(\frac{\partial f_m}{\partial x} \frac{\partial f_n}{\partial y} + \frac{\partial f_m}{\partial y} \frac{\partial f_n}{\partial x})\} J \, d\xi d\eta$$

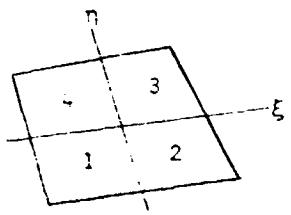
$$C_{mm} = \rho c t \int_{-1}^{1}\int_{-1}^{1} f_m \, J \, d\xi d\eta$$

$$Q_m = \Phi t \int_{-1}^{1}\int_{-1}^{1} f_m \, J \, d\xi d\eta \quad .$$

In the above, t is the thickness, and the indices m and n range from 1 to 4. The above integrals are evaluated using 2x2 Gaussian quadrature. For example, the expression for Q_m is

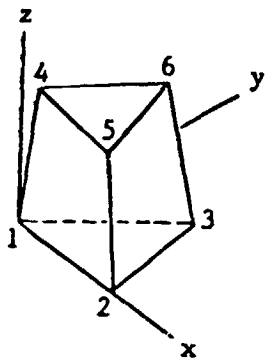
$$Q_m = \Phi t \sum_{I=1}^{4} f_m(I) \, J(I) \quad ,$$

where the f_m's and J are evaluated at the four quadrature points shown below.



1	= -.57735027	= -.57735027
2	= +.57735027	= -.57735027
3	= +.57735027	= +.57735027
4	= =.57735027	= +.57735027

3.4 K61 - 6-NODE VOLUME CONDUCTION ELEMENT



The conduction matrix, capacity matrix, and source load vector for this element are formed by condensing the corresponding K61 matrices and vector. Referring to the figure in Section 3.5, node 4 is collapsed into node 3 and node 8 is collapsed into node 7. Nodes 5, 6, and 7 are then renumbered 4, 5, and 6. The components of K, C, and Q for the K61 element are written below in terms of the corresponding K81 quantities (denoted by a tilde).

$$K_{11} = \tilde{K}_{11} \quad K_{21} = \tilde{K}_{21} \quad K_{22} = \tilde{K}_{22} \quad K_{31} = \tilde{K}_{31} + \tilde{K}_{41}$$

$$K_{32} = \tilde{K}_{32} + \tilde{K}_{42} \quad K_{33} = \tilde{K}_{33} + \tilde{K}_{44} + 2\tilde{K}_{43} \quad K_{41} = \tilde{K}_{51}$$

$$K_{42} = \tilde{K}_{52} \quad K_{43} = \tilde{K}_{53} + \tilde{K}_{54} \quad K_{44} = \tilde{K}_{55} \quad K_{51} = \tilde{K}_{61}$$

$$K_{52} = \tilde{K}_{62} \quad K_{53} = \tilde{K}_{63} + \tilde{K}_{64} \quad K_{54} = \tilde{K}_{65} \quad K_{55} = \tilde{K}_{66}$$

$$K_{61} = \tilde{K}_{71} + \tilde{K}_{81} \quad K_{62} = \tilde{K}_{72} + \tilde{K}_{82} \quad K_{63} = \tilde{K}_{73} + \tilde{K}_{83} + \tilde{K}_{84}$$

$$K_{64} = \tilde{K}_{75} + \tilde{K}_{85} \quad K_{65} = \tilde{K}_{76} + \tilde{K}_{86} \quad K_{66} = \tilde{K}_{77} + \tilde{K}_{88} + 2\tilde{K}_{87}$$

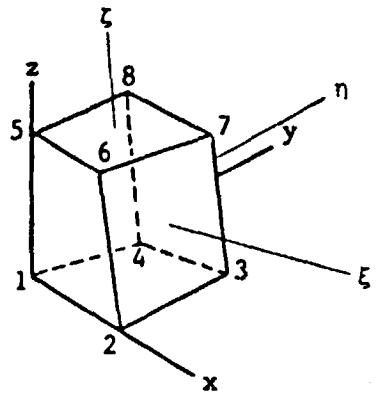
$$C_{11} = \tilde{C}_{11} \quad C_{22} = \tilde{C}_{22} \quad C_{33} = \tilde{C}_{33} + \tilde{C}_{44}$$

$$C_{44} = \tilde{C}_{55} \quad C_{55} = \tilde{C}_{66} \quad C_{66} = \tilde{C}_{77} + \tilde{C}_{88}$$

$$Q_1 = \tilde{Q}_1 \quad Q_2 = \tilde{Q}_2 \quad Q_3 = \tilde{Q}_3 + \tilde{Q}_4$$

$$Q_4 = \tilde{Q}_5 \quad Q_5 = \tilde{Q}_6 \quad Q_6 = \tilde{Q}_7 + \tilde{Q}_8$$

3.5 K81 - 8-NODE VOLUME CONDUCTION ELEMENT



An isoparametric formulation is used to compute the conduction matrix, capacity matrix, and source load vector. The coordinates and temperature are assumed to vary within the element according to

$$x = \sum_{n=1}^{8} f_n x_n \qquad y = \sum_{n=1}^{8} f_n y_n \qquad z = \sum_{n=1}^{8} f_n z_n \qquad T = \sum_{n=1}^{8} f_n T_n \quad ,$$

where the shape functions f_n are

$$f_1 = \frac{1}{8}(1-\xi)(1-\eta)(1-\zeta) \qquad f_2 = \frac{1}{8}(1+\xi)(1-\eta)(1-\zeta)$$

$$f_3 = \frac{1}{8}(1+\xi)(1+\eta)(1-\zeta) \qquad f_4 = \frac{1}{8}(1-\xi)(1+\eta)(1-\zeta)$$

$$f_5 = \frac{1}{8}(1-\xi)(1-\eta)(1+\zeta) \qquad f_6 = \frac{1}{8}(1+\xi)(1-\eta)(1+\zeta)$$

$$f_7 = \frac{1}{8}(1+\xi)(1+\eta)(1+\zeta) \qquad f_8 = \frac{1}{8}(1-\xi)(1+\eta)(1+\zeta) \quad ,$$

and ξ, η, and ζ take values between -1 and 1. The partial derivatives needed to evaluate $\frac{\partial f_m}{\partial x}$, etc., are

$$\frac{\partial \xi}{\partial x} = \frac{1}{J}\left(\frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \zeta} - \frac{\partial z}{\partial \eta}\frac{\partial y}{\partial \zeta}\right) \qquad \frac{\partial \xi}{\partial y} = \frac{1}{J}\left(\frac{\partial z}{\partial \eta}\frac{\partial x}{\partial \zeta} - \frac{\partial x}{\partial \eta}\frac{\partial z}{\partial \zeta}\right) \qquad \frac{\partial \xi}{\partial z} = \frac{1}{J}\left(\frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \zeta} - \frac{\partial y}{\partial \eta}\frac{\partial x}{\partial \zeta}\right)$$

$$\frac{\partial \eta}{\partial x} = \frac{1}{J}\left(\frac{\partial y}{\partial \zeta}\frac{\partial z}{\partial \xi} - \frac{\partial z}{\partial \zeta}\frac{\partial y}{\partial \xi}\right) \qquad \frac{\partial \eta}{\partial y} = \frac{1}{J}\left(\frac{\partial z}{\partial \zeta}\frac{\partial x}{\partial \xi} - \frac{\partial x}{\partial \zeta}\frac{\partial z}{\partial \xi}\right) \qquad \frac{\partial \eta}{\partial z} = \frac{1}{J}\left(\frac{\partial x}{\partial \zeta}\frac{\partial y}{\partial \xi} - \frac{\partial y}{\partial \zeta}\frac{\partial x}{\partial \xi}\right)$$

$$\frac{\partial \zeta}{\partial x} = \frac{1}{J}\left(\frac{\partial y}{\partial \xi}\frac{\partial z}{\partial \eta} - \frac{\partial z}{\partial \xi}\frac{\partial y}{\partial \eta}\right) \qquad \frac{\partial \zeta}{\partial y} = \frac{1}{J}\left(\frac{\partial z}{\partial \xi}\frac{\partial x}{\partial \eta} - \frac{\partial x}{\partial \xi}\frac{\partial z}{\partial \eta}\right) \qquad \frac{\partial \zeta}{\partial z} = \frac{1}{J}\left(\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi}\frac{\partial x}{\partial \eta}\right)$$

and the Jacobian J is

$$J = \frac{\partial x}{\partial \xi}\left(\frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \zeta} - \frac{\partial z}{\partial \eta}\frac{\partial y}{\partial \zeta}\right) + \frac{\partial y}{\partial \xi}\left(\frac{\partial z}{\partial \eta}\frac{\partial x}{\partial \zeta} - \frac{\partial x}{\partial \eta}\frac{\partial z}{\partial \zeta}\right) + \frac{\partial z}{\partial \xi}\left(\frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \zeta} - \frac{\partial y}{\partial \eta}\frac{\partial x}{\partial \zeta}\right) \quad .$$

The components of the conduction matrix, capacity matrix, and the source load vector are (Equations 2-6, 2-18, and 2-19)

$$K_{mn} = t \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \{k_{xx}\frac{\partial f_m}{\partial x}\frac{\partial f_n}{\partial x} + k_{yy}\frac{\partial f_m}{\partial y}\frac{\partial f_n}{\partial y} + k_{zz}\frac{\partial f_m}{\partial z}\frac{\partial f_n}{\partial z}$$

$$+ k_{xy}\left(\frac{\partial f_m}{\partial x}\frac{\partial f_n}{\partial y} + \frac{\partial f_m}{\partial y}\frac{\partial f_n}{\partial x}\right)$$

$$+ k_{yz}\left(\frac{\partial f_m}{\partial y}\frac{\partial f_n}{\partial z} + \frac{\partial f_m}{\partial z}\frac{\partial f_n}{\partial y}\right)$$

$$+ k_{zx}\left(\frac{\partial f_m}{\partial z}\frac{\partial f_n}{\partial x} + \frac{\partial f_m}{\partial x}\frac{\partial f_n}{\partial z}\right)\} J \, d\xi d\eta d\zeta \quad .$$

$$C_{nm} = \rho c \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f_m \, J \, d\xi d\eta d\zeta$$

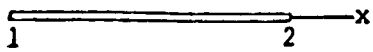
$$Q_m = \Phi \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} f_m \, J \, d\xi d\eta d\zeta \quad .$$

The indices m and n take values from 1 to 8. The above integrals are evaluated numerically using 2x2x2 Gaussian quadrature. For example, the expression for Q_m is

$$Q_m = \Phi \sum_{I=1}^{8} f_m(I) \, J(I) \quad ,$$

where f_m and J are evaluated at the eight quadrature points $\xi, \eta, \zeta = \pm .57735027$.

3.6 C21 - 2-NODE LINE CONVECTION ELEMENT



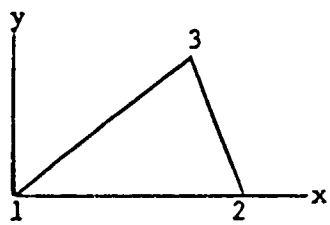
The temperature distribution functions are the same as those used for the K21 element formulation. Equations 2-8 and 2-20 are used to compute the convection matrix and load vector.

$$K = \frac{1}{6} hcL \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$H = \frac{1}{2} hcL\ T_h \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

In the above, c and L are the circumference and length of the convection surface.

3.7 C31 - 3-NODE AREA CONVECTION ELEMENT



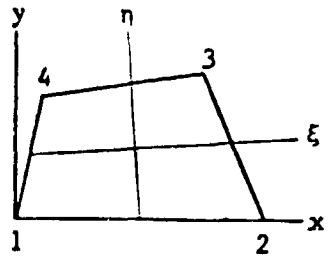
The temperature distribution is the same as that for the K31 element. The convection matrix and load vector are computed from Equations 2-8 and 2-20.

$$K = \frac{1}{12} hA \begin{bmatrix} 2+2d-2d^2 & & \text{sym} \\ 1-2d+2d^2 & 2+2d-2d^2 & \\ 1 & 1 & 2 \end{bmatrix}$$

$$H = \frac{1}{3} hA\ T_h \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

In the above equations, $A = \frac{1}{2} x_2 y_3$ and $d = x_3/x_2$.

3.8 C41 - 4-NODE AREA CONVECTION ELEMENT



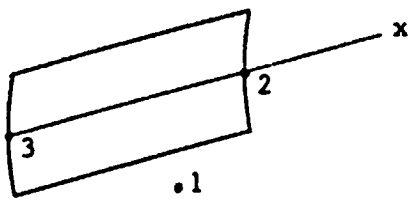
The isoparametric temperature distribution used for the K41 element formulations is used to compute the components of the C41 convection matrix and load vector.

$$K_{mn} = h \int_{-1}^{1} \int_{-1}^{1} f_m f_n \, J \, d\xi d\eta$$

$$H_m = h \, T_h \int_{-1}^{1} \int_{-1}^{1} f_m \, J \, d\xi d\eta$$

The method used to integrate the above equations is identical to that used for the K41 element.

3.9 C32 - 3-NODE FLUID-SURFACE CONVECTIVE EXCHANGE ELEMENT



The temperature distribution is assumed to be $T = f_1 T_1$ in the fluid and $T = f_2 T_2 + f_3 T_3$ on the surface, where $f_1 = 1$, $f_2 = 1 - x/L$, and $f_3 = x/L$. Substitution of these expressions into Equation 2-9 gives

$$K = \frac{1}{6} ch_f L \begin{bmatrix} 6 & -3 & -3 \\ -3 & 2 & 1 \\ -3 & 1 & 2 \end{bmatrix} ,$$

where c is the width of the contact surface, and L is the length of the contact surface. The fluid convection coefficient h_f can be modified to account for a fluid temperature variation at the flow section by either of the formulas (Reference 1)

$$\bar{h}_f = h_f \left(\frac{T_s}{T_f}\right)^n \quad \text{(ideal gas)}$$

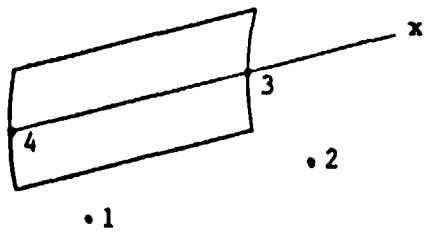
$$\bar{h}_f = h_f \left(\frac{\mu_s}{\mu_f}\right)^n \quad \text{(viscous fluid)} \quad .$$

T_s and T_f are the average surface and fluid (or gas) temperatures, and μ_s and μ_f are fluid viscosities computed at T_s and T_f respectively.

---

(1) Thornton, Earl A., TAP 1: A Finite Element Program for Steady-State Thermal Analysis of Convectively Cooled Structures, NASA CR-145069, 1976.

3.10 C42 - 4-NODE FLUID-SURFACE CONVECTIVE EXCHANGE ELEMENT

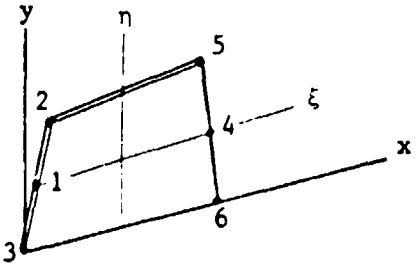


The temperature distribution is assumed to be $T = f_1T_1 + f_2T_2$ in the fluid and $T = f_3T_3 + f_4T_4$ on the surface, where $f_1 = f_4 = 1 - x/L$ and $f_2 = f_3 = x/L$. Substitution of these expressions into Equation 2-9 gives

$$K = \frac{1}{6} ch_f L \begin{bmatrix} 2 & 1 & -1 & -2 \\ 1 & 2 & -2 & -1 \\ -1 & -2 & 2 & 1 \\ -2 & 01 & 1 & 2 \end{bmatrix} ,$$

where c is the width of the contact surface, and L is the length of the contact surface. The fluid convection coefficient h_f can be modified as described for element C32.

3.11 C62 - 6-NODE FLUID-SURFACE CONVECTIVE EXCHANGE FIN ELEMENT



The temperature distribution is assumed to be $T = f_1T_1 + f_4T_4$ in the fluid and $T = f_3T_3 + f_6T_6 + f_5T_5 + f_2T_2$ on the surface. The shape functions are

$$f_1 = \frac{1}{2}(1-\xi) \quad f_2 = \frac{1}{4}(1-\xi)(1+\eta) \quad f_3 = \frac{1}{4}(1-\xi)(1-\eta)$$

$$f_4 = \frac{1}{2}(1+\xi) \quad f_5 = \frac{1}{4}(1+\xi)(1+\eta) \quad f_6 = \frac{1}{4}(1+\xi)(1-\eta) \ .$$

The fin is assumed to exchange heat from both sides with the surrounding fluid; hence the expression for K is (Equation 2-9)

$$K_{mn} = 2eh_f \int_{-1}^{1}\int_{-1}^{1} f_m f_n \, J d\xi d\eta \quad (m,n = 1,4)$$

$$= -2eh_f \int_{-1}^{1}\int_{-1}^{1} f_m f_n \, J d\xi d\eta \quad (m = 2,3,5,6;\ n = 1,4)$$

$$= 2eh_f \int_{-1}^{1}\int_{-1}^{1} f_m f_n \, J d\xi d\eta \quad (m,n = 2,3,5,6) \ .$$

Gaussian quadrature is used to numerically integrate the equations for K as described for element K41.

The fluid convective exchange coefficient h_f can be modified as described for C32 elements. The fin efficiency factor e is defined by (Reference 1)

$$e = \frac{2}{C}\frac{COSH(C)-1}{SINH(C)} \ ,$$

where

$$C = \frac{1}{2}(y_2+y_5)\left(\frac{2h}{tk_{yy}}\right)^{\frac{1}{2}}$$

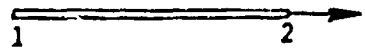
t = fin thickness

k_{yy} = fin conduction coefficient in the y direction.

---

(1) Thornton, Earl A., TAP 1: A Finite Element Program for Steady-State Thermal Analysis of Convectively Cooled Structures, NASA CR-145069, 1976.

3.12 MT21 - 2-NODE MASS TRANSPORT ELEMENT



This element's fluid conduction and capacity matrix formulations are identical to those for K21 elements. For a fluid temperature distribution given by $T = f_1T_1 + f_2T_2$, where $f_1 = 1-x/L$ and $f_2 = x/L$, the mass-transport matrix is (Equations 2-15 and 2-16)

$$K = \frac{1}{2}\dot{m}c \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} .$$

Although this expression gives accurate solutions for most fluid flow problems, it can lead to oscillatory solutions for certain types of problems as described in Reference 1. The mass-transport matrix used in the SPAR Thermal Analyzer is given by (Reference 1).

$$K = \frac{1}{2}\dot{m}c \begin{bmatrix} -1+\alpha & 1-\alpha \\ -1-\alpha & 1+\alpha \end{bmatrix} ,$$

where

$$\alpha \equiv \coth\frac{\gamma}{2} - \frac{2}{\gamma}$$

$$\gamma = \frac{\dot{m}cL}{Ak} .$$

The pressure drop in the fluid is computed from the formula (Reference 2)

$$\Delta p = \frac{1}{8}\left(\frac{\dot{m}}{A}\right)^2 \frac{FL}{D_H(\rho_1+\rho_2)} + \frac{1}{8}\left(\frac{\dot{m}}{A}\right)^2\left(\frac{1}{\rho_2} - \frac{1}{\rho_1}\right) .$$

---

(1) Heinrich, J. C., Huyakorn, P. S., Zienkiewicz, and Mitchell, A. R., "An 'Upwind" Finite Element Scheme for Two-Dimensional Convective Transport Equation", International Journal for Numerical Methods in Engineering, Vol. 11, 1977.

(2) Thornton, Earl A., TAP 1: A Finite Element Program for Steady-State Thermal Analysis of Convectively Cooled Structures, NASA CR-145069, 1976.

In the above equations

$\dot{m}$ = mass transport rate

c = specific heat

A = flow area

L = length

k = fluid conduction coefficient

g = acceleration of gravity

F = friction factor

D_H = hydraulic diameter $\equiv 4 \frac{\text{flow area}}{\text{wetted perimeter}}$

ρ_1 = fluid mass density at inlet temperature

ρ_2 = fluid mass density at outlet temperature .

For fluids, the above equation is solved directly since the mass density is a given function of temperature. For perfect gases, the equation is solved simultaneously with the gas law $p=\rho RT$. The friction factor can be corrected to account for a fluid temperature variation at a flow section by either of the formulas (Reference 1)

$$\bar{F} = F \left(\frac{T_s}{T_f}\right)^m \quad \text{(ideal gas)}$$

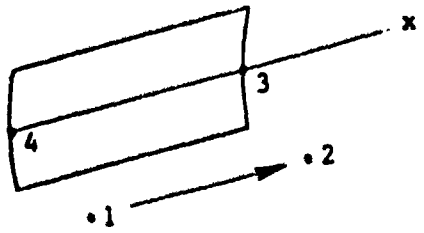
$$\bar{F} = F \left(\frac{\mu_s}{\mu_f}\right)^m \quad \text{(viscous fluid)} \quad .$$

T_s and T_f are the average surface and fluid temperatures, and μ_s and μ_f are fluid viscosities computed at T_s and T_f respectively.

---

(1) Thornton, Earl A., TAP 1: A Finite Element Program for Steady-State Thermal Analysis of Convectively Cooled Structures, NASA CR-145069, 1976.

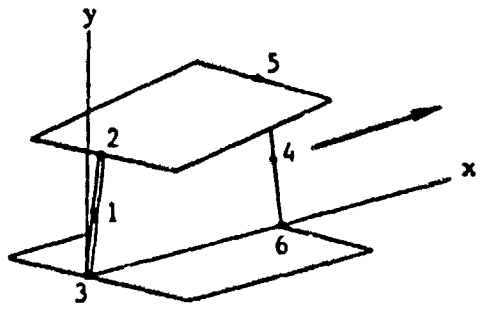
3.13 MT42 - 4-NODE INTEGRATED MASS-TRANSPORT, CONVECTIVE EXCHANGE ELEMENT



This element is characterized by mass-transport and conduction between fluid nodes 1 and 2, conduction between surface nodes 3 and 4, and convective exchange between the fluid and surface. MT42 K matrices are assembled from the element K matrix formulations tabulated below. Pressure drops are computed from the formula given for MT21 elements.

K Matrix	Element Formulation
Conduction (surface)	K21
Conduction (fluid)	MT21
Mass-transport	MT21
Convective exchange	C42

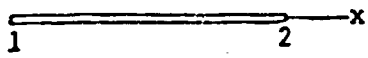
3.14 MT62 - 6-NODE INTEGRATED MASS-TRANSPORT, CONVECTIVE EXCHANGE ELEMENT



This element consists of a fin (nodes 3,6,5,2) and two walls (nodes 3,6 and 2,5) which exchange heat with a surrounding fluid. The element is characterized by mass-transport and conduction in the fluid (nodes 1 and 4), two-dimensional heat conduction in the fin, and convective heat exchange between the fluid and walls and both sides of the fin. Element K matrices are assembled from the element K matrix formulations tabulated below. Pressure drops are computed from the formula given for MT21 elements.

K Matrix	Element Formulation
Conduction (fin)	K41
Conduction (fluid)	MT21
Mass-transport	MT21
Convective exchange (fin)	C62
Convective exchange (walls)	C42

3.15 R21 - 2-NODE LINE RADIATION ELEMENT



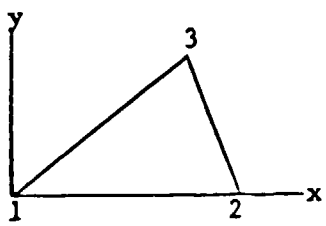
The temperature distribution is the same as that for the K21 element. The radiation matrix and load vector are (Equations 2-13 and 2-21)

$$K = \frac{1}{2} \varepsilon\sigma cL \begin{bmatrix} T_1^3 & 0 \\ 0 & T_2^3 \end{bmatrix} .$$

$$R = \frac{1}{2} aHcL \begin{bmatrix} 1 \\ 1 \end{bmatrix} .$$

In the above equation, c is the radiation surface perimeter and L is the length. The irradiation H is assumed to be constant over the surface and is computed as described in Section 4.1.

3.16 R31 - 3-NODE AREA RADIATION ELEMENT



Equations 2-13 and 2-21 are used to compute the radiation matrix and load vector. The functions f_1, f_2, and f_3 are the same as those used for the K31 element.

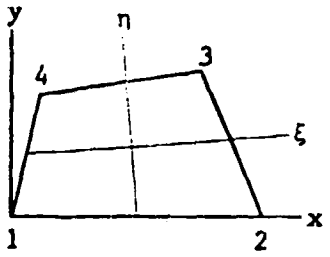
$$K = \frac{1}{3} \varepsilon\sigma A \begin{bmatrix} T_1^3 & 0 & 0 \\ 0 & T_2^3 & 0 \\ 0 & 0 & T_3^3 \end{bmatrix}$$

$$R = \frac{1}{3} aHA \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} ,$$

C-2

where A is the area of the element. The irradiation H is assumed to be constant over the surface and is computed as described in Section 4.1.

3.17 R41 - 4-NODE AREA RADIATION ELEMENT



The components of the diagonal radiation matrix and load vector are given by (Equations 2-13 and 2-21)

$$K_{mm} = \varepsilon\sigma T_m^3 \int_{-1}^{1}\int_{-1}^{1} f_m \, J \, d\xi d\eta$$

$$R_m = aH \int_{-1}^{1}\int_{-1}^{1} f_m \, J \, d\xi d\eta \quad .$$

The functions f_m are the same as those used for the K41 element, and the integrals are evaluated numerically using Gaussian quadrature. The irradiation H is assumed to be constant over the surface and is computed as described in Section 4.1.

Section 4

COMPUTATIONAL METHODS

This section contains descriptions of the computational methods utilized by the SPAR thermal analysis processors SSTA and TRTA. Incident radiant heat fluxes are computed as described in Section 4.1. Sections 4.2 and 4.3 contain descriptions of the steady-state and transient solution methods.

4.1 RADIATION EXCHANGE

This section describes how the incident radiant heat flux (irradiation) is computed for radiation elements. Each R21, R31, and R41 element is assumed to be an opaque surface on which both the emissive power ($\varepsilon\sigma T^4$) and irradiation (H) are uniformly distributed.

For a system of n elements, the irradiation on element i is

$$H_i = \frac{1}{A_i} \sum_{j=1}^{n} B_j A_j F_{ji} \quad , \tag{4-1}$$

where

A_i = surface area of element i.

B_j = radiant heat flux leaving element j (radiosity).

F_{ji} = radiation exchange factor, defined as the fraction of radiant heat leaving element j incident on element i. The exchange factors can represent diffuse (view factor) and/or specular radiation (see Reference 1, Section 5.1). For gray surfaces, $A_i F_{ij} = A_j F_{ji}$.

The radiosity of element i is defined as the sum of the emissive power and the reflected part of the irradiation, i.e.,

$$B_i = \sigma\varepsilon_i T_i^4 + r_i H_i \quad , \tag{4-2}$$

---

(1) Sparrow, E. M. and R. D. Cess, Radiation Heat Transfer, McGraw-Hill, 1978.

where r_i is the reflectivity defined by $a_i+r_i=1$. Substitution of this equation into Equation 4-1 yields

$$H_i = \frac{1}{A_i} \sum_{j=1}^{n} (\sigma \epsilon_j A_j T_j^4 + r_j H_j A_j) F_{ji} \quad . \tag{4-3}$$

The above equation can be written in matrix form as

$$\Gamma = F^T(E + r\Gamma) \quad , \tag{4-4}$$

where

$$\Gamma \equiv \begin{bmatrix} A_1H_1 \\ A_2H_2 \\ - \\ - \\ A_nH_n \end{bmatrix} \quad , \qquad F \equiv \begin{bmatrix} F_{11} & F_{12} & - & - & - \\ F_{21} & F_{22} & - & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & - & F_{nn} \end{bmatrix} \quad ,$$

$$E \equiv \begin{bmatrix} \sigma\epsilon_1 A_1 T_1^4 \\ \sigma\epsilon_2 A_2 T_2^4 \\ - \\ - \\ \sigma\epsilon_n A_n T_2^4 \end{bmatrix} \quad , \qquad r \equiv \begin{bmatrix} r_1 & & & & \\ & r_2 & & & \\ & & . & & \\ & & & . & \\ & & & & r_1 \end{bmatrix} \quad .$$

The computational objective is, given all other quantities in Equation 4-4, to compute all of the Γ_i's. For very small problems, Equation 4-4 can be solved by direct matrix inversion and multiplication; however, for problems involving more than a few hundred elements, the computational costs of standard direct methods becomes prohibitive. Accordingly, an alternative

method has been developed. This method is used in both SSTA and TRTA. It (1) produces solutions accurate to any user prescribed numerical tolerance, (2) is far less expensive to execute than direct standard matrix solution methods, even for small problems, (3) remains highly efficient for extremely large systems, and (4) takes advantage of network "sparsity" (i.e., takes advantage of the fact that radiation exchange between many elements may be non-existent or negligible). The computational procedure considers the radiation exchange process to take place in a sequence of steps as described below. In what follows, the i-th term in Γ_k is the total heat incident on element i at the k-th step, and the i-th term in R_k is the heat reflected by element i during the k-th step.

Step 0: The emissive power for each element is computed and distributed to all elements, so that the initial incident heat vector is

$$\Gamma_0 = F^T E \quad ,$$

and the initial reflected heat vector is

$$R_0 = r\Gamma_0 \quad .$$

Step 1: The heat reflected during step 0 is distributed to all elements so that the incident heat vector becomes

$$\Gamma_1 = \Gamma_0 + F^T R_0 \quad ,$$

and the heat reflected during step 1 is

$$R_1 = rF^T R_0 \quad .$$

-
-
-

Step k: The heat reflected during step k-1 is distributed to all elements so that the incident heat vector becomes

$$\Gamma_k = \Gamma_{k-1} + F^T R_{k-1} \quad ,$$

and the heat reflected during step k is

$$R_k = rF^T R_{k-1} \quad .$$

The procedure is terminated when the sum of the heat reflected by all elements during step k is less than a user specified value.

-
-
-

The above procedure typically converges to highly accurate solutions within 2 to 10 steps. The rapid convergence can be illustrated mathematically by noting that R_k can be expressed in terms of Γ_0 and the k-th powers of the reflectivities and exchange factors by successive back substitution in the above sequence of steps; i.e.,

$$R_k = (rF^T)^k r\Gamma_0 \quad .$$

It can be argued intuitively that each term in R_k rapidly approaches zero as k increases, because for practical problems, the reflectivities are less than unity and the exchange factors are much less than unity.

That the above sequence of steps satisfies Equation 4-4 can be verified by expressing Γ_k in terms of Γ_0 for each step:

Step 1: $\Gamma_1 = \Gamma_0 + F^T r\Gamma_0$, $R_1 = r(\Gamma_1 - \Gamma_0)$

Step 2: $\Gamma_2 = \Gamma_0 + F^T r\Gamma_1$, $R_2 = r(\Gamma_2 - \Gamma_1)$

-

-

-

Step k: $\Gamma_k = \Gamma_0 + F^T r\Gamma_{k-1}$, $R_k = r(\Gamma_k - \Gamma_{k-1})$

The expression for Γ_k can be rewritten as

$$\Gamma_k = \Gamma_0 + F^T r\Gamma_k - F^T R_k$$

which agrees with Equation 4-4 when $R_k = 0$.

4.2 STEADY-STATE SOLUTIONS

The equation for steady-state heat transfer is (Equation 2-5)

$$(K_k + K_h + K_r + K_m)T = Q + H + R \quad , \tag{4-5}$$

where K_h, K_h, K_r are the symmetric conduction, convection, and radiation matrices, K_m is the asymmetric mass-transport matrix, and Q, H, and R are the source, convection, and radiation load vectors.

4.1.1 Solution to the Equation KT = F

Solutions to equations of the form KT = F are required for both linear and nonlinear analyses. The method used in the SPAR thermal analysis processor SSTA involves factoring the K matrix into upper and lower triangular parts and solving the resulting matrix equations by forward and back substitution. The factored K matrix has the form K = LU, where L and U are lower and upper triangular matrices respectively.*

The components of L and U are computed according to the following recurrence relationships.

$$D_j \equiv K_{jj} - \sum_{k=1}^{j-1} L_{jk}U_{ki}$$

$$L_{jj} = \mathrm{sgn}(D_j)\sqrt{|D_j|} \qquad U_{jj} = |L_{jj}|$$

$$L_{ij} = (K_{ij} - \sum_{k=1}^{j-1} L_{ik}U_{kj})/U_{jj} \qquad (i>j)$$

$$U_{ji} = (K_{ji} - \sum_{k=1}^{j-1} L_{jk}U_{ki})/L_{jj} \qquad (i>j) \quad .$$

These relationships are valid for j>1. For j=1, terms involving products of L and U are ignored. After the K matrix is factored, the solution is

---

* All terms above the diagonal in L are zero. All terms below the diagonal in U are zero.

carried out by the forward substitution solution to LZ = F and the back substitution solution to UT = Z. The forward substitution sequence is

$$Z_1 = F_1/L_{11}$$

$$Z_2 = (F_2 - L_{21}Z_1)/L_{22}$$

$$Z_3 = (F_3 - L_{31}Z_1 - L_{32}Z_2)/L_{33}$$

$$\vdots$$

$$Z_n = (F_n - \sum_{k=1}^{n-1} L_{nk}Z_k)/L_{nn} \quad ,$$

and the back substitution sequence is

$$T_n = Z_n/U_{nn}$$

$$\vdots$$

$$T_i = (Z_i - \sum_{k=i+1}^{n} U_{ik}T_k)/U_{ii}$$

$$\vdots$$

$$T_1 = (Z_1 - \sum_{k=2}^{n} U_{1k}T_k)/U_{11} \quad .$$

4.2.2 Nonlinear Analysis

The SPAR thermal analysis processor SSTA utilizes a modified Newton-Raphson technique to solve the nonlinear heat equation. A brief description of the method is presented below.

For a vector Ψ defined by

$$\Psi \equiv (K_k + K_h + K_r + K_m)T - Q - H - R \quad , \qquad (4\text{-}6)$$

the nonlinear steady-state heat equation takes the form (Equation 4-5)

$$\Psi = 0 \quad .$$

The Newton-Raphson method of solution to the above equation involves successively solving the recurrence relationship

$$K(T_{i+1} - T_i) = -\Psi_i \tag{4-7}$$

for $T_{i+1}(i = 0, 1, 2, \ldots)$ until convergence is achieved. In this equation, T_i is the vector of nodal temperatures at step i, Ψ_i is the vector Ψ at step i, and K is the square derivative matrix

$$K \equiv \left[\frac{\partial \Psi}{\partial T}\right] \tag{4-8}$$

at step i. The method is illustrated for a one-degree-of-freedom system on Figure 4-1a.

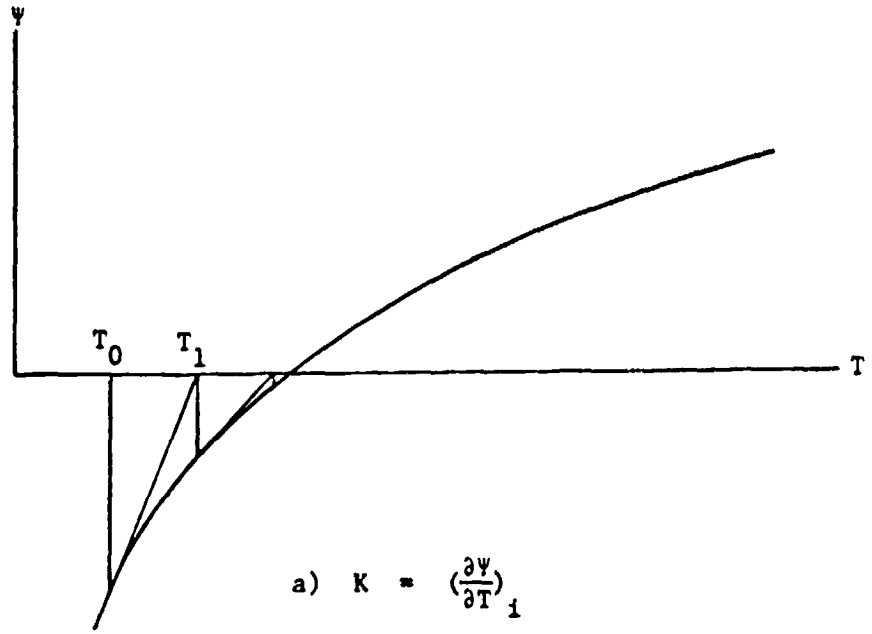
The Newton-Raphson method could prove to be unnecessarily costly because the K matrix is factored at each step. An alternate approach is to iterate more than once using the same K matrix as shown on Figure 4-1b. Although the number of solutions required is greater for the modified method, the number of factorings is less. Since the computer time required to factor the K matrix is generally many times greater than that required for a solution, the modified method of factoring once and iterating many times can prove more economical than the method which requires factoring at each step.

In view of the above discussion, it can be argued that forming the exact K matrix is not necessary in order to obtain accurate solutions. Indeed, as shown on Figure 4-2a, an approximate expression for K should not appreciably affect the convergence characteristics of the modified method and, as shown on Figure 4-2b, can even improve the convergence characteristics.

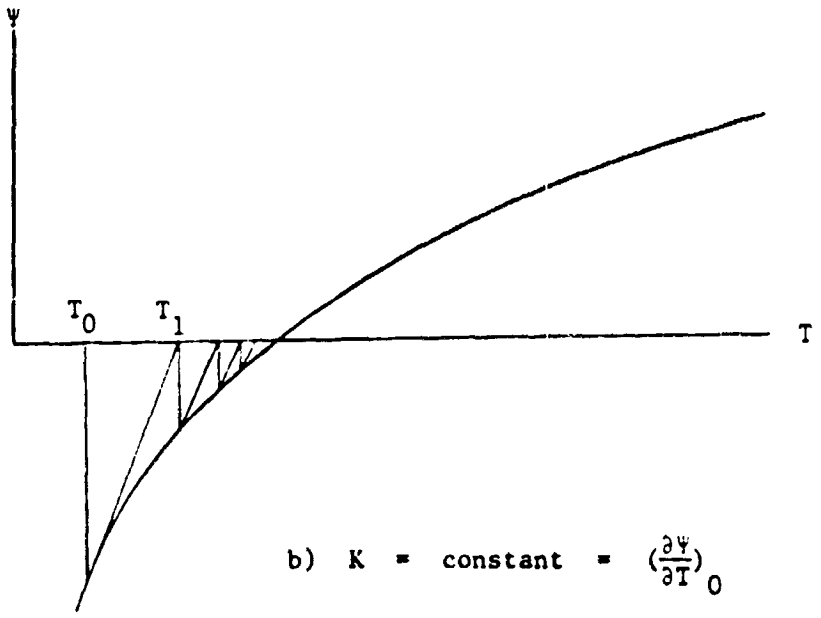
The K matrix is computed by substituting Equation 4-6 into Equation 4-8 and utilizing Equations 2-6 through 2-21, i.e.,

$$K = K_k + K_h + 4K_r + K_m + \Delta K - \left[\frac{\partial H}{\partial T}\right] - \left[\frac{\partial R}{\partial T}\right] .$$

The matrices ΔK and $\left[\frac{\partial H}{\partial T}\right]$ reflect changes in material properties with respect to temperature. In general, they have only a minor effect on the accuracy of K. The matrix $\left[\frac{\partial R}{\partial T}\right]$ is a consequence of radiation exchange. Its terms are of the order of magnitude of terms in K_r multiplied by exchange factors (see Section 4.1). Because of the difficulty and potentially prohibitive costs in



a) $K = \left(\frac{\partial \Psi}{\partial T}\right)_i$



b) $K = \text{constant} = \left(\frac{\partial \Psi}{\partial T}\right)_0$

Figure 4-1. Newton-Raphson and Modified Newton-Raphson Methods for a One-Degree-of-Freedom System.

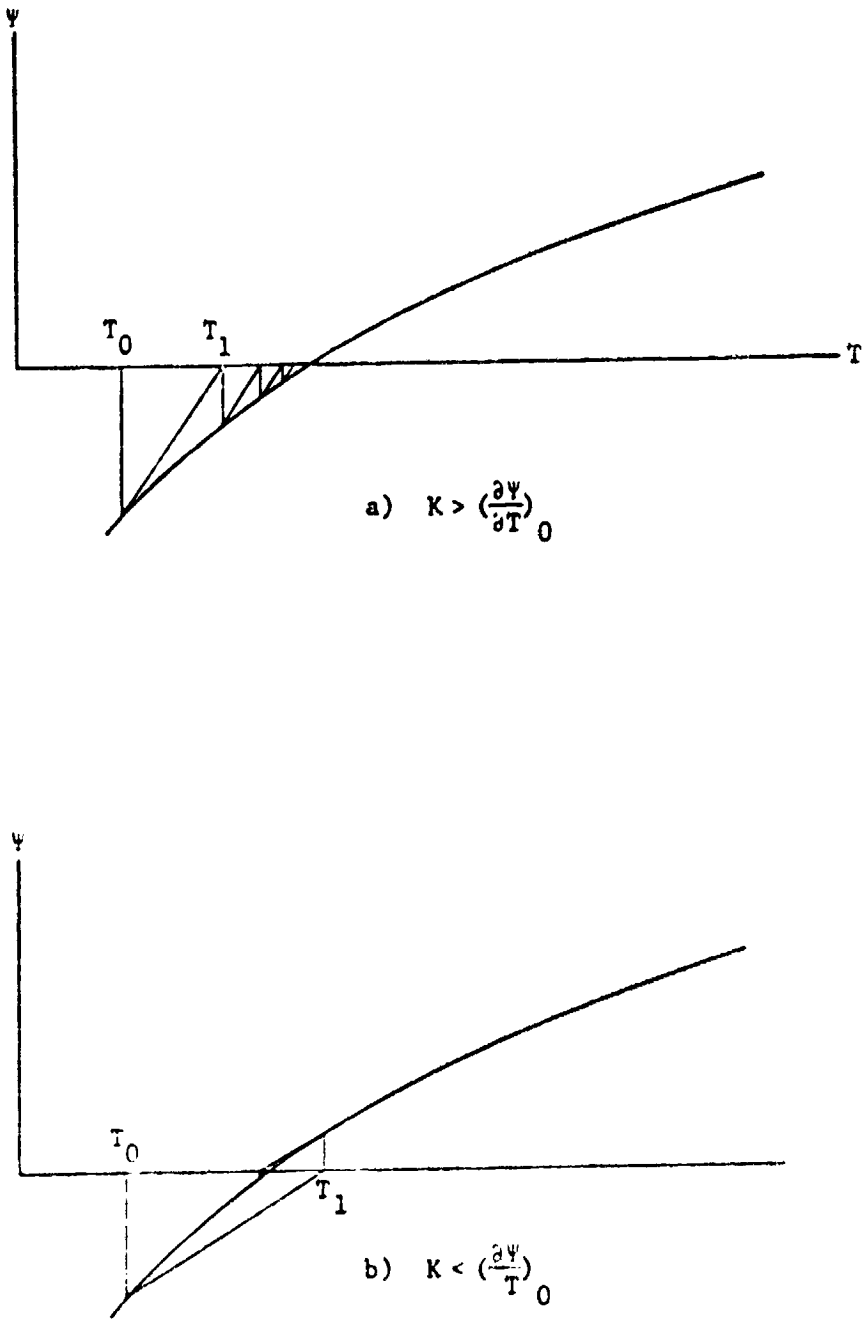


Figure 4-2. Modified Newton-Raphson Method with Approximate Initial K Matrix.

computing $\left[\frac{\partial R}{\partial T}\right]$, and in keeping with the discussion following Equation 4-8, the expression for K used in SSTA is

$$K = K_k + K_n + 4K_r + K_m \quad . \tag{4-9}$$

If material properties are not temperature dependent, and if there is no radiation exchange, the above equation is exact.

An additional simplification is made in the construction of the radiation matrix. In order to avoid integrating the T^3 term over the surface area of each radiation element (Equation 2-10), K_r is assumed to be a diagonal matrix (Equation 2-13).

The SPAR thermal analysis processor SSTA computes nonlinear solutions according to the following sequence of steps. Iteration and convergence control parameters are described in the SPAR Thermal Analysis Reference Manual, Volume 1, Program Execution.

Step 0: Set T_0 = initial temperature vector.
Compute and factor K.
Compute Ψ_0 .
Solve $K\Delta T = \Psi_0$.
Compute $T_1 = T_0 + \Delta T$.
If $|\frac{\Delta T}{T}|$ is less than a user-specified amount for all node points, terminate the procedure; otherwise, go to step 1.

-
-
-

Step i: If factoring is required, compute and factor K; otherwise, use the K matrix from step i-1.
Compute Ψ_i .
Solve $K\Delta T = \Psi_i$.
Compute $T_{i+1} = T_i + \Delta T$.
If $|\frac{\Delta T}{T}|$ is less than a user-specified amount for all node points, terminate the procedure; otherwise, go to step i+1.

-
-
-

4.3 TRANSIENT SOLUTIONS

The explicit solution method is described in Section 4.3.1. The explicit solution process for problems with zero-capacity nodes is described in Section 4.3.2. The implicit solution method is described in Section 4.3.3.

4.3.1 The Explicit Time Integration Method

The transient heat transfer equation is integrated by assuming the temperature vector at time t_{i+1} can be expressed as

$$T_{i+1} = T_i + \dot{T}_i \Delta t + \tfrac{1}{2}\ddot{T}_i \Delta t^2 + \ldots \quad , \tag{4-10}$$

where T_i is the temperature vector at time t_i, and Δt is the time increment. The vector $\dot{T}$ is determined directly from Equation 2-5, i.e.,

$$\dot{T} = -C^{-1}(K_k + K_n + K_r + K_m)T + C^{-1}(Q + H + R) \quad . \tag{4-11}$$

Higher order derivatives are obtained by differentiating the above equation according to the assumptions (1) material, fluid, and gas properties are constant, (2) mass-transport rates are constant, (3) Q and H vary linearly with time, and (4) R is constant.*

$$\begin{aligned} \ddot{T} &= -C^{-1}(K_k + K_h + 4K_r + K_m)\dot{T} + C^{-1}(\dot{Q} + \dot{H}) \\ \dddot{T} &= -C^{-1}(K_k + K_h + 4K_r + K_m)\ddot{T} + 4\dot{K}_r\dot{T} \\ &\vdots \end{aligned} \tag{4-12}$$

A limitation can be placed on the size of Δt by expanding Equation 4-10 in terms of T_i and noting the convergence characteristics. Substituting

---

* Time-dependent properties and mass-transport rates are averaged over regions of time, defined by the user, called "time intervals." Temperature-dependent properties are evaluated at the temperatures computed at the beginning of each time interval. Q, $\dot{Q}$, H, $\dot{H}$, and R are computed at specific times designated by the user.

Equations 4-11 and 4-12 into Equation 4-10 and rearranging terms yields

$$T_{i+1} = \{1 - (\lambda-3\lambda_r)\Delta t + \frac{1}{2}(\lambda-3\lambda_r)\lambda\Delta t^2 - \frac{1}{3!}(\lambda-3\lambda_r)\lambda^2\Delta t^3 + \frac{1}{4!}(\lambda-3\lambda_r)\lambda^3\Delta t^4 - \cdots\}T_i + F\Delta t - \frac{1}{2}(\lambda F-\dot{F})\Delta t^2 + \frac{1}{3!}(\lambda F-\dot{F})\lambda\Delta t^3 - \frac{1}{4!}(\lambda F-\dot{F})\lambda^2\Delta t^4 + \cdots + \text{terms containing } \dot{\lambda}_r, \ddot{\lambda}_r, \dddot{\lambda}_r, \cdots .$$

In the above

$$\lambda \equiv C^{-1}(K_k + K_h + 4K_r + K_m)$$

$$\lambda_r \equiv C^{-1}K_r \qquad (4\text{-}13)$$

$$F \equiv C^{-1}(Q + H + R) \quad .$$

If $\dot{\lambda}_r$, $\ddot{\lambda}_r$, etc., are ignored, the series for T_{i+1} converges when

$$\lambda_{max}\Delta t < 1 \quad , \qquad (4\text{-}14)$$

where λ_{max} is the largest term in the λ matrix.

4.3.2 Arithmetic Nodes

If a node has zero capacity, the temperature derivative at that node cannot be computed from Equation 4-11; instead, it must be computed directly in terms of the temperature derivatives at the other node points in the model. Such a node is called an "arithmetic node." For a system containing arithmetic nodes, the heat transfer equation KT+CT = F, written in partitioned form, is

$$\begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \end{bmatrix} + \begin{bmatrix} C_1 & \\ & 0 \end{bmatrix} \begin{bmatrix} \dot{T}_1 \\ \dot{T}_2 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}$$

where subscripts 1 and 2 refer to non-arithmetic and arithmetic nodes respectively. The above equation can be rewritten as

$$\ddot{T}_1 = -C_1^{-1}K_{11}T_1 - C_1^{-1}K_{12}T_2 + C_1^{-1}F_1 \qquad (4\text{-}15)$$

$$K_{22}T_2 = -K_{21}T_1 + F_2 \quad . \qquad (4\text{-}16)$$

The derivatives $\ddot{T}_1$, etc., are computed by differentiating Equation 4-15 as described for Equation 4-11. Derivatives of T_2 are computed by differentiating Equation 4-16, i.e.,

$$\begin{aligned} K_{22}\dot{T}_2 &= -K_{21}\dot{T}_1 - \dot{K}_{22}T_2 + \dot{F}_2 \\ K_{22}\ddot{T}_2 &= -K_{21}\ddot{T}_1 - 2\dot{K}_{22}\dot{T}_2 - \ddot{K}_{22}T_2 \\ &\vdots \end{aligned} \qquad (4\text{-}17)$$

The derivatives of K_{22} appear in the above equations because of radiation terms (there are no radiation terms in K_{21} since the radiation matrix is diagonal). Equations 4-17 are solved using the modified Newton-Raphson technique described in Section 4.2.2.

4.3.3 The implicit Time Integration Method

The weighted residual method (Reference 1) is used to derive the implicit time integration equations. During each time step, the temperature vector is approximated by

$$T = T_i + (T_{i+1} - T_i)\frac{t}{\Delta t} \quad , \tag{4-18}$$

where T_i is the temperature vector at time t_i, T_{i+1} is the temperature vector at time t_{i+1}, and Δt is the time step size. Substitution of the above equation into Equation 2-5 yields an equation of the form

$$K\{T_i + (T_{i+1}-T_i)\frac{t}{\Delta t}\} + C\{T_{i+1}-T_i\}\frac{t}{\Delta t} - F = E(t) \quad . \tag{4-19}$$

The error function E(t) is minimized by integrating the above equation from 0 to Δt with respect to the weighting function

$$w = 1 + 3(1-2\beta)(1-2\frac{t}{\Delta t}) \quad ,$$

and setting the result equal to zero, i.e.

$$\int_0^{\Delta t} E(t)\, w\, dt = 0 \quad .$$

The resulting matrix equation is

$$\{C + \beta\Delta tK\}T_{i+1} = \{C - (1-\beta)\Delta tK\}T_i + F\Delta t + \dot{F}\beta\Delta t^2 \quad , \tag{4-20}$$

which is solved at successive points in time in accordance with the list of assumptions following Equation 4-11. For different values of the weighted residual parameter β, Equation 4-20 corresponds to the following well known algorithms (Reference 2)

$\beta = 0$, Euler or forward difference
$\beta = 1/2$, Crank-Nicolson
$\beta = 2/3$, Galerkin
$\beta = 1$, Backward difference

---

(1) Chung, T. J., Finite Element Analysis in Fluid Dynamics, McGraw Hill, 1978.

(2) Argyris, J. H., L. Z. Van, and K. J. William, "Higher Order Methods for Transient Diffusion Analysis", Computer Methods in Applied Mechanics and Engineering, Vol. 12, 1977, pp. 243-278.

Equation 4-20 will give stable solutions for all values of Δt when $\beta>0$; however, the accuracy of solutions is not predictable for values of Δt greater than the value given by Equation 4-14.

For problems without radiation effects, the quantities K, F, and $\dot{F}$ in Equation 4-20 are (see the definitions following Equation 2-5)

$$K = K_k + K_h + K_m$$

$$F = Q + H$$

$$\dot{F} = \dot{Q} + \dot{H} \ .$$

For radiation problems, K, F, and $\dot{F}$ are

$$K = K_k + K_h + K_m + \tilde{K}_r T_0^3$$

$$F = Q + H + R + \Psi$$

$$\dot{F} = \dot{Q} + \dot{H}$$

$$\Psi = \tilde{K}_r\{T_0^3(T_i+\beta T_{i+1}-\beta T_i) + T_i^4(.8\beta-.6) + T_i^3T_{i+1}(.4\beta-.4)$$

$$- .2T_i^2T_{i+1}^2 - .4\beta T_iT_{i+1}^3 - T_{i+1}^4(.8\beta-.2)\}$$

T_0 = temperature vector at the time K was formed.

$\tilde{K}_r$ = radiation distribution matrix, $K_r = \tilde{K}_r T^3$.

Equation 4-20 is solved for T_{i+1} at each time step by successive iterations. If convergence does not occur within a user specified number of iterations, the K matrix is reformed with $T_0=T_i$.

Reference Manual

SPAR Thermal Analysis Processors

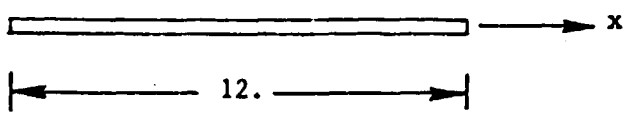
Volume 3

Demonstration Problems

CONTENTS

3.1 Steady-State Convection for a Bar

The bar shown below convects along its length to a constant temperature of 70 degrees. The left end of the bar (node 1) is held at a constant temperature of 250 degrees. The right end (node 13) is insulated. The runstream for the problem is shown on page 1-2.



conduction coefficient =	2.75
cross section area =	0.19635
circumference =	1.5708
convection coefficient =	0.011111

The results computed by SPAR are compared with the analytical results from Reference 1 below.

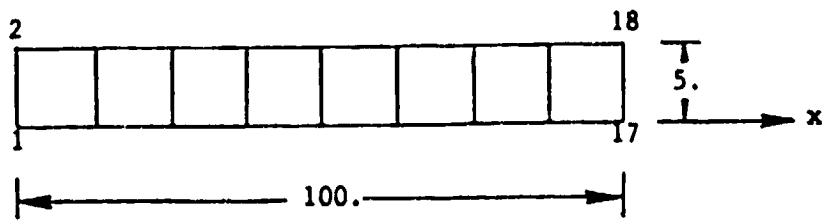
Bar Node	Analytical Solution	SPAR Solution
1	250.0	250.0
2	221.2	221.2
3	197.3	197.3
4	177.6	177.6
5	161.4	161.3
6	148.1	148.0
7	137.4	137.3
8	128.8	128.7
9	122.2	122.1
10	117.2	117.1
11	113.8	113.6
12	111.7	111.6
13	111.1	111.0

(1) Carslaw, H. S. and J. C. Jaeger, "Conduction of Heat in Solids", Oxford Press, 1959, p. 141.

```
@XQT TAB
 START 13
 JLOC: 1 0. 0. 0. 12. 0. 0. 13 1
@XQT AUS
 TABLE(NI=9,NJ=1):   COND PROP 1: I=4: J=1:     2.75
 TABLE(NI=2,NJ=1):   CONV PROP 1: I=2: J=1:     1.1111-2
 TABLE(NI=1,NJ=12): CTEM C21  1:       J=1,12: 70.
 TABLE(NI=1,NJ=1):   K     AREA:       J=1:     .19635
 TABLE(NI=1,NJ=1):   C     CIRC:       J=1:     1.5708
 TABLE(NI=1,NJ=1):   TEMP  NODE:       J=1:     1.
 TABLE(NI=1,NJ=1):   APPL  TEMP:       J=1:     250.
@XQT ELD
 RESET NUTED=1
 K21
   1 2  1 12
 C21
   1 2  1 12
@XQT TGEO
@XQT SSTA
@XQT DCU
 PRINT 1 STAT TEMP
 TOC 1
 STOP
```

3.2 Steady-State Conduction for a Plate

The plate shown below generates heat uniformly over its area. The top and bottom edges are insulated, and the left and right edges are held at constant temperatures by prescribing very high convection coefficients for the C21 elements joining nodes 1,2 and 17,18 respectively. The runstream for the problem is shown on page 2-2.



source heat rate = 4.
thickness = 0.2
conduction coefficient = 20.
temperature at left edge = 100.
temperature at right edge = 200.

The analytical solution is

$$T = 100 + 11x - 0.1x^2$$

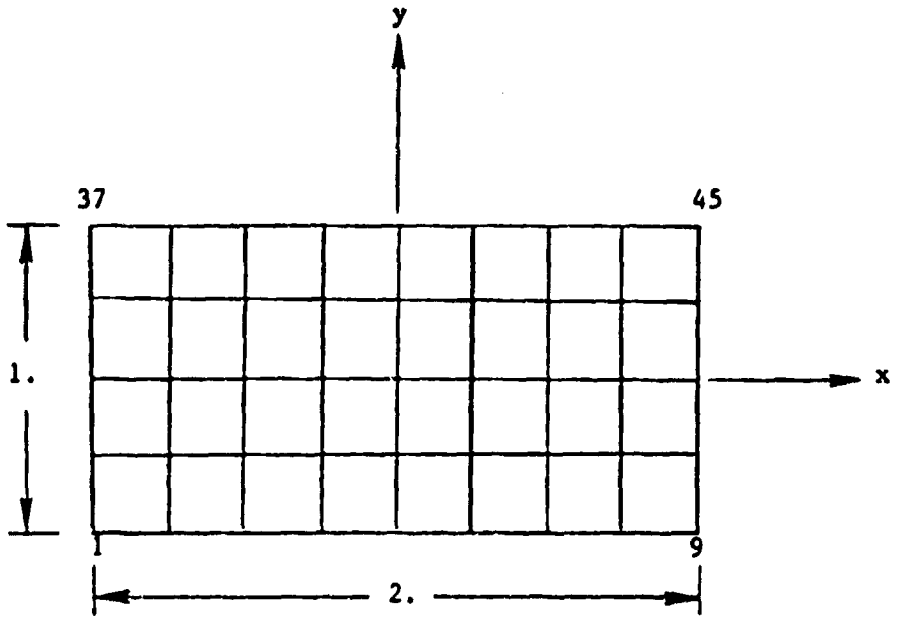
Analytical and computed results are compared below.

Plate Nodes	Analytical Solution	SPAR Solution
1,2	100.0	100.0
3,4	221.9	221.9
5,6	312.5	312.5
7,8	371.9	371.9
9,10	400.0	400.0
11,12	396.9	396.9
13,14	362.5	362.5
15,16	296.9	296.9
17,18	200.0	200.0

```
@XQT TAB
 START 18
 JLOC: 1 0. 0. 0. 100. 0. 0. 9 2 2
       1 0. 5. 0. 100. 5. 0.
@XQT AUS
 TABLE(NI=9,NJ=1): COND PROP 1: I=4: J=1:   20.
 TABLE(NI=2,NJ=1): CONV PROP 1: I=2: J=1:   1.+5
 TABLE(NI=1,NJ=2): CTEM C21  1:      J=1,2: 100. 200.
 TABLE(NI=1,NJ=8): SOUR K41  1:      J=1,8: 4.
 TABLE(NI=1,NJ=1): K    THIC:        J=1:   .2
 TABLE(NI=1,NJ=1): C    CIRC:        J=1:   1.
@XQT ELD
 RESET NUTED=1
 C21
   1 2: 17 18
 K41
   1 3 4 2  1 8
@XQT TGEO
@XQT SSTA
@XQT DCU
 PRINT 1 STAT TEMP
 TOC 1
 STOP
```

3.3 Steady-State Conduction for an Orthotropic Plate

The orthotropic plate shown below generates heat uniformly over its area. All four edges have prescribed temperatures. The runstream for the problem is shown on page 3-2.



conduction coefficient, x-direction = 2.0
conduction coefficient, y-direction = 1.2337
thickness = 1.0
heat generation rate = 1000.0
edge temperature = 0.0

The results computed by SPAR are compared with the analytical results from Reference 1 below.

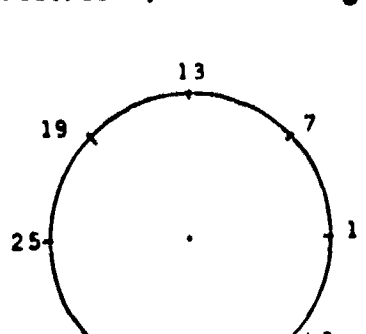
Location	Analytical Solution	SPAR Solution
x=±.5 y=±.25	52.73	54.2
x= .0 y=±.25	63.54	64.8
x=±.5 y= .0	68.62	70.5
x= .0 y= .0	83.72	85.5

(1) Carslaw, H. S. and J. C. Jaeger, "Conduction of Heat in Solids", Oxford Press, 1959, p.170.

```
@XQT TAB
 START 45
 JLOC: 1 -1. -.5  0. 1. -.5  0. 9 1 5
       9 -1.  .5  0. 1.  .5  0.
@XQT AUS
 TABLE(NI=9,NJ=1):  COND PROP 1: I=4 5:  J=1:      2. 1.2336997
 TABLE(NI=1 NJ=32): SOUR K41  1:         J=1,32:  1000.
 TABLE(NI=1,NJ=1):  K    THIC:           J=1:     1.
 TABLE(NI=1,NJ=24): TEMP NODE: DDATA=1.: J=1,10:  1.
                                         J=11,14: 18. 19. 27. 28.
                               DDATA=1.: J=15,24: 36.
 TABLE(NI=1,NJ=24): APPL TEMP:           J=1,24:  0.
@XQT ELD
 RESET NUTED=1
 K41
   1 2 11 10  1 8 4
@XQT TGEO
@XQT SSTA
@XQT DCU
 PRINT 1 STAT TEMP
 TOC 1
 STOP
```

3.4 Steady-State and Transient Convection for Fourier's Ring Problem

The runstream for the problem described below is shown on the next page. For both steady-state and transient analyses, the ring shown convects to a constant temperature. For the steady-state analysis, the temperature at node 25 is prescribed. For the transient analysis, no temperatures are prescribed, and the ring cools from the steady-state temperatures.



radius = 1.
cross sectional area = 1.
conduction coefficient = 2π
convection coefficient = 1.
convective exchange temperature = 0.

The analytical solutions obtained from Reference 1 are compared with the results computed by SPAR below.

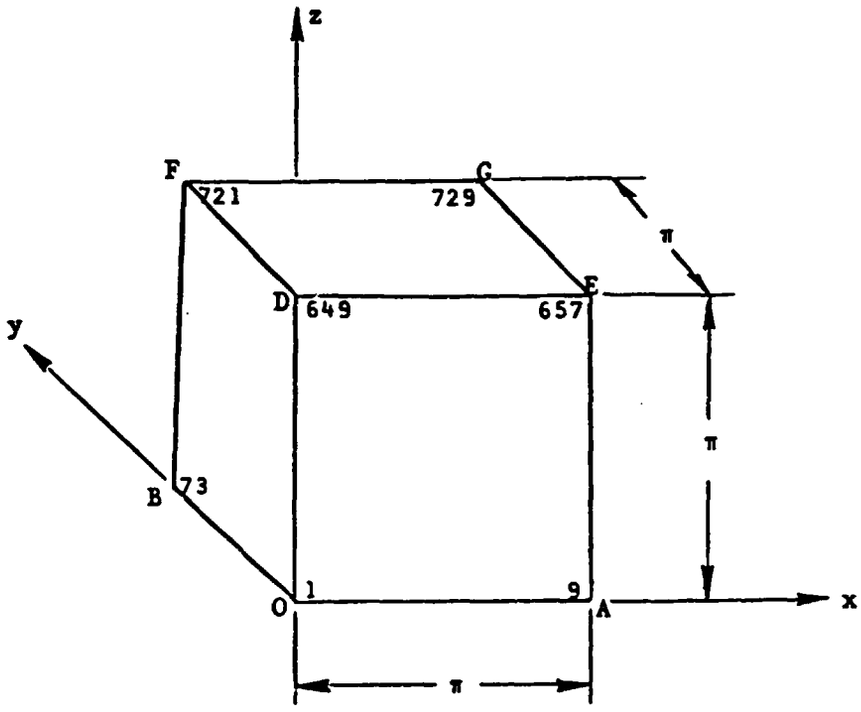
Nodes	Steady-State Solutions		Transient Solutions (t=.05)	
	Analytical	SPAR	Analytical	SPAR
1	8.63	8.63	8.625	8.632
4,46	9.30	9.30		
7,43	11.43	11.43	11.376	11.382
10,40	15.34	15.34		
13,37	21.65	21.65	20.544	20.546
16,34	31.34	31.34		
19,31	45.92	45.92	34.914	34.972
22,28	67.68	67.67		
25	100.00	100.00	43.011	43.150

(1) Carslaw, H. S. and J. C. Jaeger, "Conduction of Heat in Solids", Oxford Press, 1959, pp. 160-161.

```
@XQT TAB
 START 48
 JLOC: FORMAT=2
       1 1. 0. 0. 1. 352.5 0. 48 1
@XQT AUS
 TABLE(NI=9,NJ=1):  COND PROP 1: I=2 3 4: J=1:     1. 1. 6.2831853
 TABLE(NI=2,NJ=1):  CONV PROP 1: I=2:     J=1:     1.
 TABLE(NI=1,NJ=48): CTEM C21  1:          J=1,48: 0.
 TABLE(NI=1,NJ=1):  K    AREA:            J=1:     1.
 TABLE(NI=1,NJ=1):  C    CIRC:            J=1:     6.2831853
 TABLE(NI=1,NJ=1):  TEMP NODE:            J=1:     25.
 TABLE(NI=1,NJ=1):  APPL TEMP:            J=1:     100.
@XQT ELD
 RESET NUTED=1
 K21
   1  2  1 47
   48 1
 C21
   1  2  1 47
   48 1
@XQT TGEO
@XQT SSTA
@XQT DCU
 PRINT 1 STAT TEMP
 DISABLE 1 TEMP NODE
@XQT TRTA
 RESET T2=0.05, DT=0.005
 TEMP=1 STAT TEMP
@XQT DCU
 PRINT 1 TRTA TEMP
 TOC 1
 STOP
```

3.5 Steady-State Solution for a Conducting Solid

A linear steady-state solution was obtained for the cube shown below. Each face of the cube had a prescribed temperature.



conduction coefficient = 1.

temperatures:
face ODFB = 100.
face AEGC = 200.
faces OAED
DEGF
GFBC
OACB = 0.

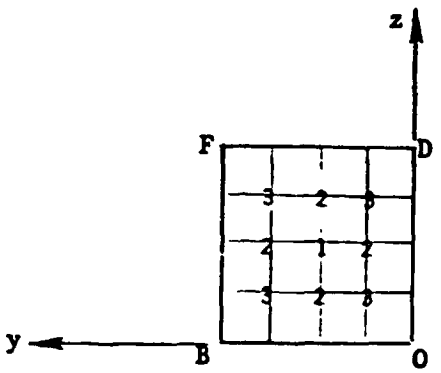
The cube was comprised of K81 conducting elements and each face was modeled with C41 convecting elements.

The results computed by TRTA are compared with the solution obtained from Reference 1 on the next page.

---

(1) Carslaw, H. S. and J. C. Jaeger, "Conduction of Heat in Solids", Oxford Press, 1959, p. 177.

Results are tabulated for nine locations in the x-direction through the solid from face ODFB to face AEGC.



T(x) at face location (1):	Analytical Solution	SPAR Solution
	100.	99.999
	75.051	76.498
	56.015	58.280
	47.210	49.346
	50.	52.333
	65.859	69.150
	96.726	100.90
	143.464	146.25
	200.	200.

T(x) at face location (2):	Analytical Solution	SPAR Solution
	100.	99.999
	66.477	69.421
	44.587	47.111
	35.260	36.637
	36.817	38.180
	49.886	52.097
	78.223	83.076
	128.227	134.06
	200.	200.

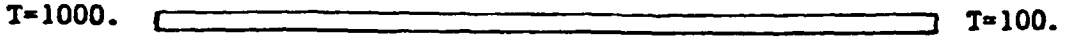
T(x) at face location (3):	Analytical Solution	SPAR Solution
	100.	99.999
	59.725	64.122
	35.944	38.795
	26.477	27.315
	27.192	27.881
	38.031	39.445
	64.049	69.658
	116.082	124.85
	200.	200.

Runstream and data for the solid are shown below.

```
@XQT TAB
 START 729
 JLOC
    1 0. 0.       0.       3.14159 0.       0.       9 1 9
    9 0. 3.14159 0.       3.14159 3.14159 0.
   82 0. 0.        .39270 3.14159 0.        .39270 9 1 9
    9 0. 3.14159  .39270 3.14159 3.14159  .39270
  163 0. 0.        .78540 3.14159 0.        .78540 9 1 9
    9 0. 3.14159  .78540 3.14159 3.14159  .78540
  244 0. 0.       1.17810 3.14159 0.       1.17810 9 1 9
    9 0. 3.14159 1.17810 3.14159 3.14159 1.17810
  325 0. 0.       1.57080 3.14159 0.       1.57080 9 1 9
    9 0. 3.14159 1.57080 3.14159 3.14159 1.57080
  406 0. 0.       1.96350 3.14159 0.       1.96350 9 1 9
    9 0. 3.14159 1.96350 3.14159 3.14159 1.96350
  487 0. 0.       2.35619 3.14159 0.       2.35619 9 1 9
    9 0. 3.14159 2.35619 3.14159 3.14159 2.35619
  568 0. 0.       2.74889 3.14159 0.       2.74889 9 1 9
    9 0. 3.14159 2.74889 3.14159 3.14159 2.74889
  649 0. 0.       3.14159 3.14159 0.       3.14159 9 1 9
    9 0. 3.14159 3.14159 3.14159 3.14159 3.14159
@XQT AUS
 TABLE(NI=9,NJ=1):   COND PROP 1: I=4 5 6: J=1:      1. 1. 1.
 TABLE(NI=2,NJ=1):   CONV PROP 1: I=2:     J=1:      1.+5
 TABLE(NI=1,NJ=64):  CTEM C41  1:          J=1,64:   100.
 TABLE(NI=1,NJ=64):  CTEM C41  2:          J=1,64:   200.
 TABLE(NI=1,NJ=192): CTEM C41  3:          J=1,192:  0.
@XQT ELD
 RESET NUTED=1
 C41
  GROUP 1
   1 82 91 10  1 8 8
  GROUP 2
   9 18 99 90  1 8 8
  GROUP 3
   2   3   84  83   1 6 8
   650 651 660 659  1 6 8
   722 723 642 641  1 6 8
   74  75  66  65   1 6 8
 K81
   1 8 8 8  1 9 81 0 1
@XQT TGEO
@XQT SSTA
 RESET LK=4480
@XQT DCU
 PRINT 1 STAT TEMP
 TOC 1
 STOP
```

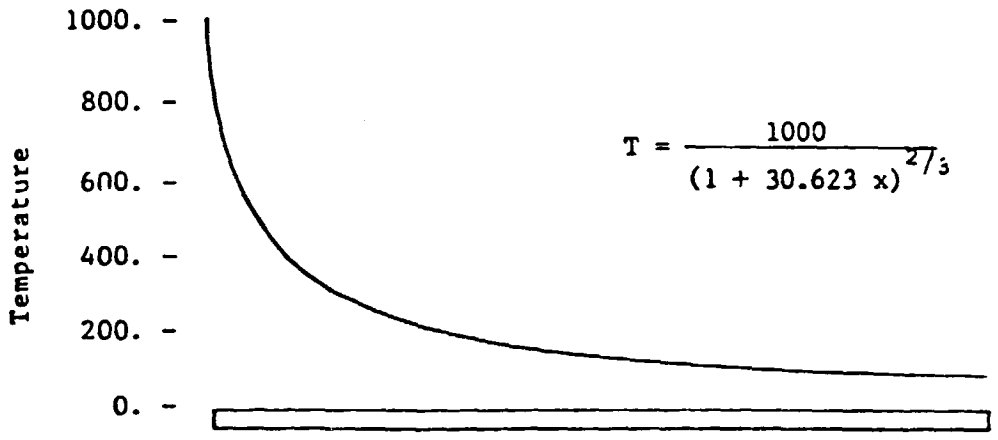
3.6 Steady-State Analysis of a Bar Radiating to Space

The bar shown below has prescribed end temperatures and radiates along its length to space.



length = 1.
area = 1.
circumference = 1.
conduction coefficient = 959739.52
emissivity = 1.
Stefan-Boltzmann constant = 1.

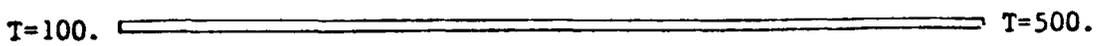
The exact steady-state solution is shown below. Results computed by SSTA are compared with the exact solution on the following page.



Node	SPAR Solution	Analytical Solution	Node	SPAR Solution	Analytical Solution
1	1000.00	1000.00	52	153.69	
2	839.07	836.86	53	151.83	
3	729.80	727.24	54	150.02	
4	650.07	647.64	55	148.27	
5	588.96	586.76	56	146.56	146.45
6	540.38	538.42	57	144.91	
7	5(0.69	498.95	58	143.30	
8	467.54	466.00	59	141.73	
9	439.38	437.99	60	140.21	
10	415.09	413.85	61	138.72	138.63
11	393.90		62	137.28	
12	375.22		63	135.87	
13	358.60		64	134.50	
14	343.71		65	133.16	
15	330.27	329.50	66	131.85	131.78
16	318.07		67	130.58	
17	306.94		68	129.33	
18	296.73		69	128.12	
19	287.33		70	126.93	
20	278.64	278.11	71	125.77	125.71
21	270.58		72	124.64	
22	263.08		73	123.53	
23	256.08		74	122.45	
24	249.52		75	121.39	
25	243.37	242.97	76	120.35	120.30
26	237.58		77	119.33	
27	232.13		78	118.34	
28	226.98		79	117.36	
29	222.10		80	116.41	
30	217.48	217.18	81	115.47	115.43
31	213.09		82	114.56	
32	208.92		83	113.66	
33	204.94		84	112.77	
34	201.15		85	111.91	
35	197.53	197.29	86	111.06	111.03
36	194.07		87	110.23	
37	190.75		88	109.41	
38	187.57		89	108.61	
39	184.53		90	107.82	
40	181.60	181.40	91	107.05	107.03
41	178.79		92	106.29	
42	176.08		93	105.54	
43	173.47		94	104.80	
44	170.96		95	104.08	
45	168.53	168.37	96	103.37	103.36
46	166.20		97	102.68	
47	163.94		98	101.99	
48	161.75		99	101.32	
49	159.64		100	100.65	
50	157.59	157.46	101	100.00	100.00
51	155.61				

3.7 Steady-State Analysis of a Bar with Temperature-Dependent Conductivity

The bar shown is insulated along its length. The end temperatures are prescribed.

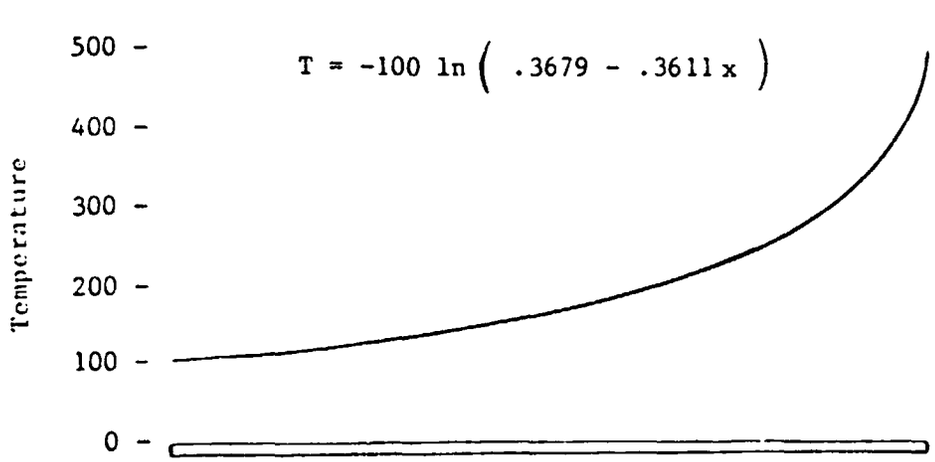


20 segment bar

$$\text{conductivity} = e^{\frac{-T}{100}}$$

length = 1.
area = 1.
circumference = 1.

The exact solution is shown below. These results are compared with SPAR computed results on the following page.

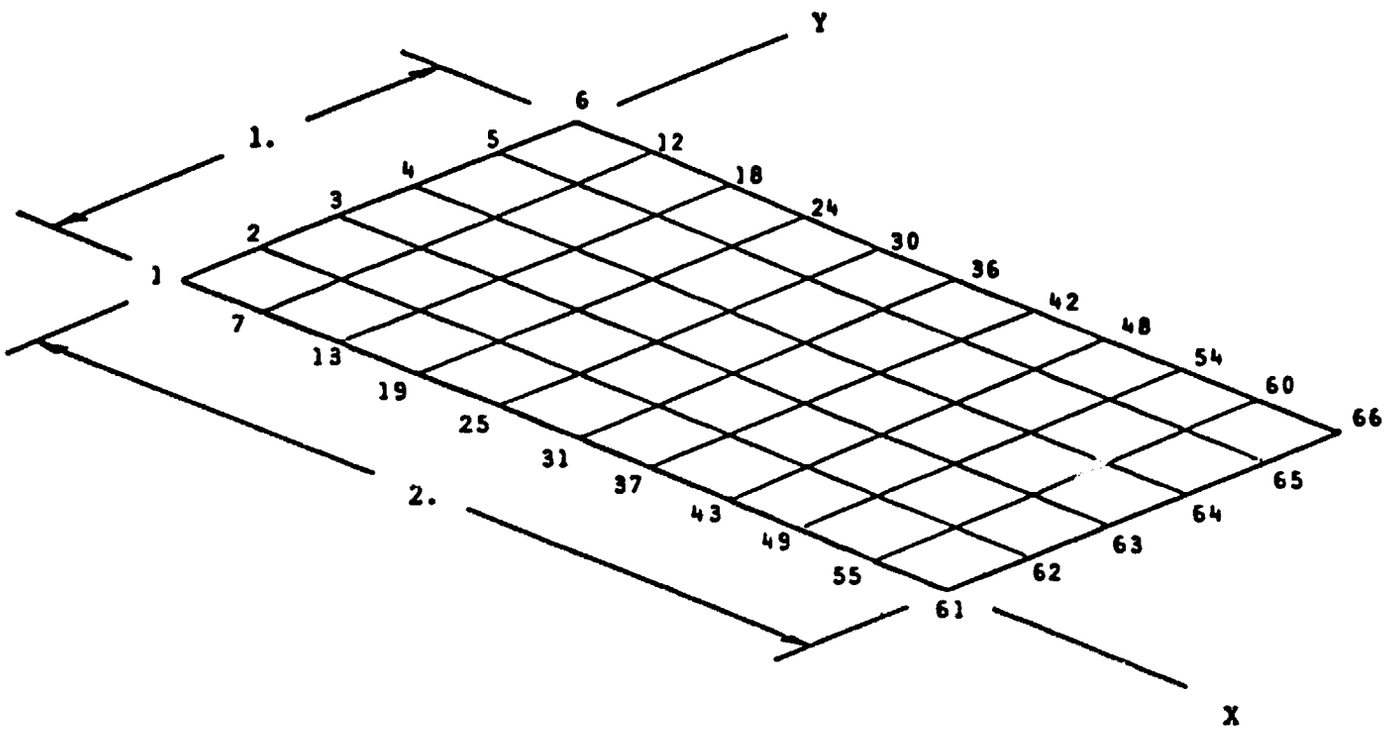


SPAR Solution	Analytical Solution
100.00	100.
105.38	
110.95	110.
116.75	
122.79	122.
129.12	
135.77	135.
142.80	
150.29	150.
158.34	
167.09	167.
176.79	
187.80	189.
200.90	
216.09	216.
233.11	
252.86	254.
277.29	
313.60	315.
365.84	
500.00	500.

```
@XQT TAB
 START 21
 JLOC: 1 0. 0. 0. 1. 0. 0. 21 1
@XQT AUS
 TABLE(NI=9,NJ=7): COND PROP 1: I=1 4: J=1,7:    0. 1.
                                                100. .36788
                                                200. .13534
                                                300. .04979
                                                400. .01832
                                                500. .00674
                                               1000. .000045
 TABLE(NI=1,NJ=1): K    AREA:          J=1:    1.
 TABLE(NI=1,NJ=2): TEMP NODE:          J=1,2: 1. 21.
 TABLE(NI=1,NJ=2): APPL TEMP:          J=1,2: 100. 500.
@XQT ELD
 RESET NUTED=1
 K21
  1 2  1 20
@XQT TGEO
@XQT SSTA
 RESET PROB=LINEAR, DEST=2
 TEMP=250.
@XQT DCU
 COPY 2,3 STAT TEMP
@XQT SSTA
 RESET DEST=2, NFAC=3, NITER=0
@XQT DCU
 COPY 2,3 STAT TEMP
@XQT SSTA
 RESET DEST=2, NFAC=2, NITER=5, START=KFAC
@XQT DCU
 PRINT 2 STAT TEMP
 TOC 1
 TOC 2
 STOP
```

TA3.8 Steady-State Analysis of a Rectangular Plate with Orthotropic, Temperature-Dependent Conductivities

The runstream for the problem described below is shown on page 8-3.



Conductivities: $k_{xx} = e^{-T/200}$, $k_{yy} = 1$

Prescribed edge temperatures: $T(x,0) = 100$

$T(0,y) = 100$

$T(2,y) = 100(4y+1)$

$T(x,1) = 100(x^2+1)$

Heat source distribution: $q = 200y(yx^2-1)e^{-\frac{1}{2}(x^2+1)}$

Analytical solution: $T(x,y) = 100(yx^2+1)$

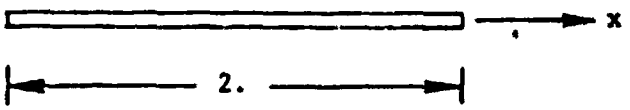
The temperature-dependent conductivities were input as piecewise linear functions of temperature, and the source strengths were computed at the centers of the K41 elements. The results computed by SPAR are compared with the analytical results on the next page.

Plate Node	SPAR Solution	Analytical Solution	Plate Node	SPAR Solution	Analytical Solution
1	100.00	100.	34	160.04	160.
2	100.00	100.	35	179.93	180.
3	100.00	100.	36	200.00	200.
4	100.00	100.	37	100.00	100.
5	100.00	100.	38	128.85	128.8
6	100.00	100.	39	157.63	157.6
7	100.00	100.	40	186.40	186.4
8	100.84	100.8	41	215.29	215.2
9	101.67	101.6	42	244.00	244.
10	102.49	102.4	43	100.00	100.
11	103.27	103.2	44	139.23	139.2
12	104.00	104.	45	178.41	178.4
13	100.00	100.	46	217.63	217.6
14	103.27	103.2	47	256.74	256.8
15	106.54	106.4	48	296.00	296.
16	109.77	109.6	49	100.00	100.
17	112.94	112.8	50	151.21	151.2
18	116.00	116.8	51	202.43	202.4
19	100.00	100.	52	253.55	253.6
20	107.30	107.2	53	304.76	304.
21	114.58	114.4	54	356.00	356.
22	121.82	121.6	55	100.00	100.
23	128.98	128.8	56	164.81	164.8
24	136.00	136.	57	229.58	229.6
25	100.00	100.	58	294.46	294.4
26	112.90	112.8	59	359.26	359.2
27	125.77	125.6	60	424.00	424.
28	138.57	138.4	61	100.00	100.
29	151.30	151.2	62	180.00	180.
30	164.00	164.	63	260.00	260.
31	100.00	100.	64	340.00	340.
32	120.08	120.	65	420.00	420.
33	140.11	140.	66	500.00	500.

```
@XQT TAB
 START 66
 JLOC: 1 0. 0. 0. 1. 0. 6 1 11
       6 2. 0. 0. 2. 1. 0.
@XQT AUS
 TABLE(NI=9,NJ=7):  COND PROP 1: I=1 4 5: J=1,7:   0. 1.000 1.
                                                  100.  .607 1.
                                                  200.  .368 1.
                                                  300.  .223 1.
                                                  400.  .135 1.
                                                  500.  .082 1.
                                                  600.  .018 1.
 TABLE(NI=1,NJ=1):  K    THIC: J=1: 1.
 TABLE(NI=1,NJ=30): TEMP NODE: J=1,30
      1.   7.  13.  19.  25.  31.  37.  43.  49.  55.
     61.  62.  63.  64.  65.  66.  60.  54.  48.  42.
     36.  30.  24.  18.  12.   6.   5.   4.   3.   2.
 TABLE(NI=1,NJ=30): APPL TEMP: J=1,30
    100. 100. 100. 100. 100. 100. 100. 100. 100. 100.
    100. 180. 260. 340. 420. 500. 424. 356. 296. 244.
    200. 164. 136. 116. 104. 100. 100. 100. 100. 100.
 TABLE(NI=1,NJ=50): SOUR K41 1: J=1,50
    -12.1124  -11.9675  -11.6804  -11.2570  -10.7056
    -10.0368   -9.2637   -8.4009   -7.4644   -6.4713
    -36.2283  -34.9344  -32.4235  -28.8425  -24.3968
    -19.3339  -13.9238   -8.4394   -3.1375    1.7576
    -60.1991  -56.6349  -49.8560  -40.5134  -29.4731
    -17.7042   -6.1616    4.3199   13.1049   19.8015
    -84.0253  -77.0975  -64.1851  -46.9964  -27.6915
     -8.5065    8.6010   22.2148   31.5913   36.6512
   -107.7072  -96.3500  -75.6082  -48.9525  -20.5492
      5.6369   26.5878   40.6561   47.6120   48.3730
@XQT ELD
 RESET NUTED=1
 K41
   1 7 8 2  1 10 5
@XQT TGEO
@XQT SSTA
 RESET DEST=2, NFAC=2, NITER=0
 TEMP=100.
@XQT DCU
 COPY 2,3 STAT TEMP
@XQT SSTA
 RESET DEST=2, NFAC=0, NITER=5, START=KFAC
@XQT DCU
 PRINT 2 STAT TEMP
 TOC 1
 TOC 2
 STOP
```

3.9 Transient Analysis of a Convecting Bar

The runstream for the problem described below is on page 9-2. The bar shown has an initial temperature of 100 degrees and convects along its length and at each end to an ambient temperature of zero degrees.



length =	2.
conduction coefficient =	20.
convection coefficient =	40.
area =	1.
circumference =	1.
heat capacity =	1.

The results computed by SPAR at .02 seconds are compared with the results obtained from Reference 1 below.

Bar Node	Analytical Solution	SPAR Solution
1	31.156	30.648
2	44.824	44.455
3	54.192	54.154
4	59.369	59.680
5	60.985	61.445
6	59.369	59.680
7	54.192	54.154
8	44.824	44.455
9	31.156	30.648

(1) Carslaw, H. S. and J. C. Jaeger, "Conduction of Heat in Solids", Oxford Press, 1959, p. 144.

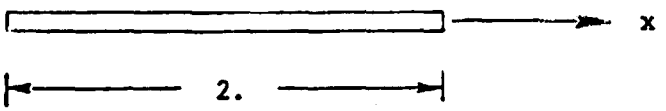
```
@XQT TAB
 START 9
 JLOC: 1 -1. 0. 0. 1. 0. 0. 9 1
@XQT AUS
 TABLE(NI=9,NJ=1): COND PROP 1: I=2 3 4: J=1:    1. 1. 20.
 TABLE(NI=2,NJ=1): CONV PROP 1: I=2:     J=1:    40.
 TABLE(NI=1,NJ=8): CTEM C21  1:          J=1,8: 0.
 TABLE(NI=1,NJ=2): CTEM C21  2:          J=1,2: 0.
 TABLE(NI=1,NJ=9): TRTA TEMP:            J=1,9: 100.
 TABLE(NI=1,NJ=1): K    AREA:            J=1:    1.
 TABLE(NI=1,NJ=1): C    CIRC:            J=1:    1.
@XQT ELD
 RESET NUTED=1
 K21
   1 2  1 8
 C21
  GROUP 1
   1 2  1 8
  GROUP 2
   1 1: 9 9
@XQT TGEO
@XQT TRTA
 RESET T1=0.0, T2=.01, DT=.0005
 TSAVE=.002
@XQT DCU
 PRINT 1 TRTA TIME
 PRINT 1 TRTA TEMP
 TOC 1
 STOP
```

For the implicit method of time integration the TRTA runstream is given below. The value for DT is for purposes of illustration only.

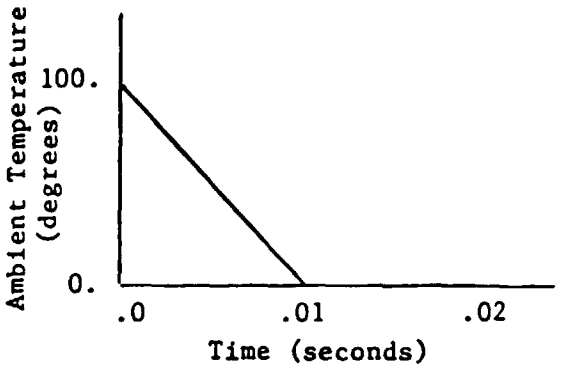
```
@XQT TRTA
 RESET METHOD=IMPLICIT, T1=0.0, T2=.01, DT=.002
 TSAVE=.002
 STOP
```

3.10 Transient Analysis of a Convecting Bar with Time-Dependent Thermal Loading

The bar shown has an initial temperature of 100 degrees. It convects uniformly into a medium whose temperature is given below.



length =	2.
area =	1.
circumference =	1.
convection coefficient =	100.
conduction coefficient =	1.
heat capacity =	1.



The analytical solution is

$$T = 100(1-e^{-100t}) + 100(1-100t) , \quad 0 \leq t \leq .01$$

$$T = 171.83e^{-100t} , \quad .01 \leq t$$

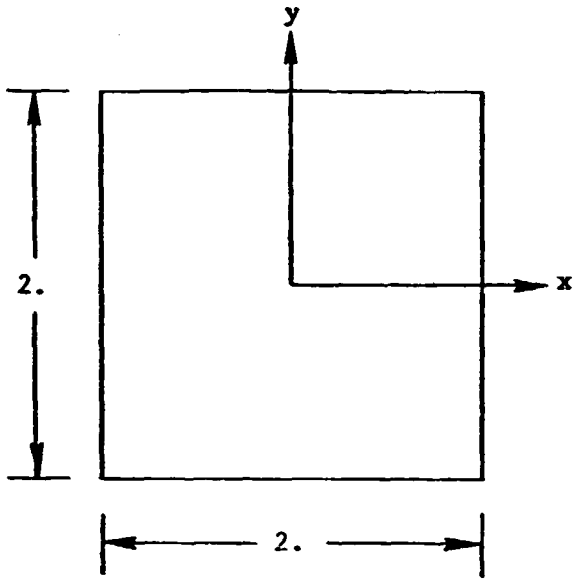
Results computed by SPAR are compared with the analytical solution below.

Time (sec)	Analytical Solution	SPAR Solution
.0	100.	100.
.005	89.35	89.346
.01	63.21	63.211
.015	38.34	38.340
.02	23.25	23.255

```
@XQT TAB
 START 9
 JLOC: 1 -1. 0. 0. 1. 0. 0. 9 1
@XQT AUS
 TABLE(NI=9,NJ=1): COND PROP 1: I=2 3 4: J=1:   1. 1. 1.
 TABLE(NI=2,NJ=1): CONV PROP 1: I=2:     J=1:   100.
 TABLE(NI=1,NJ=1): K    AREA:            J=1:     1.
 TABLE(NI=1,NJ=1): C    CIRC:            J=1:     1.
 TABLE(NI=1,NJ=8): CTEM C21  1: BLOCK 1: J=1,8: 100.
                                BLOCK 2: J=1,8:   0.
                                BLOCK 3: J=1,8:   0.
 TABLE(NI=1,NJ=3): CTEM TIME:            J=1:   .0
                                         J=2:   .01
                                         J=3:   .02

@XQT ELD
 RESET NUTED=1
 K21
   1 2  1 8
 C21
   1 2  1 8
@XQT TGEO
@XQT TRTA
 RESET T1=0.0, T2=.02, DT=.0025, PRINT=2, NTERMS=5
 TSAVE=.005
 TEMP=100.
@XQT DCU
 PRINT 1 TRTA TIME
 PRINT 1 TRTA TEMP
 TOC 1
 STOP
```

3.11 Transient Analysis of an Orthotropic Plate

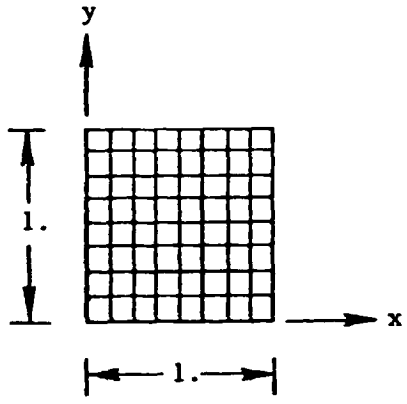
The runstream for the problem described below is shown on page 11-2. The square plate shown has a uniform initial temperature of 100 degrees. Its edges convect to an ambient temperature of zero degrees.



conduction coefficients:
Kxx = 1.
Kyy = 4.

convection coefficient = 20.
thickness = 1.

A one quarter model of the plate is shown below. The edges defined by x,y=0 are insulated, and the edges defined by x,y=1 are the convecting edges.



Results computed by SPAR at t=.01 seconds are compared with the analytical solution obtained from Reference 1 below.

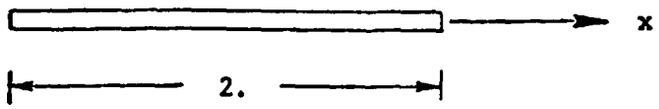
Plate Location	Analytical Solution	SPAR Solution
x=0 y=0	99.911	99.953
x=1 y=0	25.530	24.137
x=0 y=1	42.736	42.679
x=1 y=1	10.920	12.977

```
@XQT TAB
 START 81
 JLOC: 1 0. 0. 0. 1. 0. 0. 9 1 9
       9 0. 1. 0. 1. 1. 0.
@XQT AUS
 TABLE(NI=9,NJ=1):  COND PROP 1: I=2 3 4 5: J=1:    1. 1. 1. 4.
 TABLE(NI=2,NJ=1):  CONV PROP 1: I=2:       J=1:    20.
 TABLE(NI=1,NJ=16): CTEM C21  1:            J=1,16: 0.
 TABLE(NI=1,NJ=1):  C    CIRC:              J=1:    1.
 TABLE(NI=1,NJ=1):  K    THIC:              J=1:    1.
@XQT ELD
 RESET NUTED=1
 C21
   9  18  1 8
   73 74  1 8
 K41
   1 2 11 10  1 8 8
@XQT TGEO
@XQT TRTA
 RESET T1=0.0, T2=.01, DT=.0005, PRINT=2
 TEMP=100.
 TSAVE=.01
@XQT DCU
 PRINT 1 TRTA TIME
 PRINT 1 TRTA TEMP
 TOC 1
 STOP
```

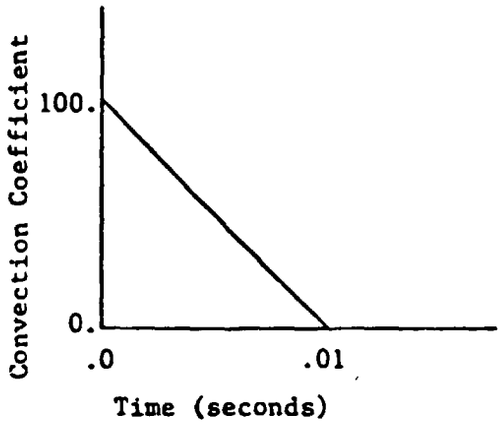
(1) Carslaw, H. S. and J. C. Jaeger, "Conduction of Heat in Solids", Oxford Press, 1959, p. 173.

3.12 Transient Analysis of a Convecting Bar with Time-Dependent Convection Coefficients

The bar shown has an initial temperature of 100 degrees and convects uniformly to an ambient temperature of zero degrees. The convection coefficient is a function of temperature as shown below. The runstream for the problem is shown on page 12-2.



length =	2.
area =	1.
circumference =	1.
conduction coefficient =	1.
heat capacity =	1.



The analytical solution is

$$T = 100e^{-100(t-50t^2)}, \quad 0 \leq t \leq .01$$

$$T = 60.6531, \quad .01 \leq t$$

The results computed by SPAR are compared with the analytical solution below.

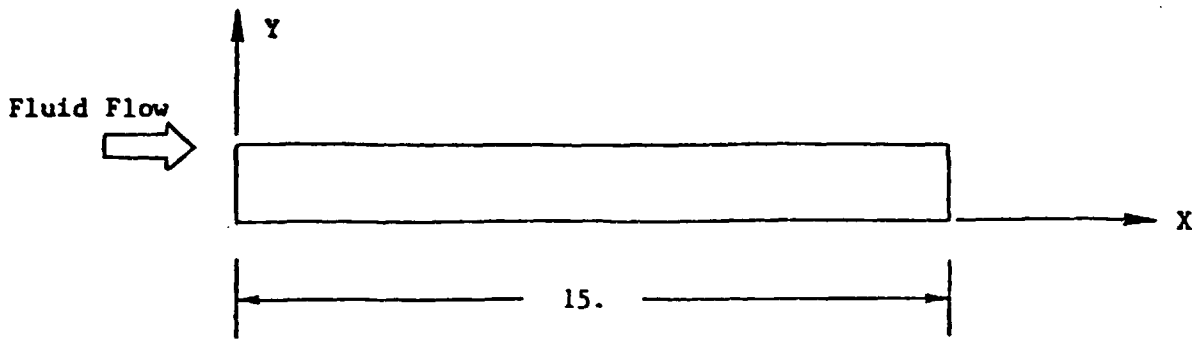
Time	Analytical Solution	SPAR Solution
.005	68.7289	68.729
.01	60.6531	60.653
.0125	60.6531	60.653

```
@XQT TAB
 START 9
 JLOC: 1 -1. 0. 0. 1. 0. 0. 9 1
@XQT AUS
 TABLE(NI=9,NJ=1): COND PROP 1: I=2 3 4: J=1:    1. 1. 1.
 TABLE(NI=1,NJ=1): CONV COEF:   BLOCK 1: J=1:    100.
                                BLOCK 2: J=1:    0.
                                BLOCK 3: J=1:    0.
 TABLE(NI=1,NJ=3): COEF TIME:             J=1:    0.
                                          J=2:    .010
                                          J=3:    1.0
 TABLE(NI=1,NJ=8): CTEM C21  1:           J=1,8: 0.
 TABLE(NI=1,NJ=1): K    AREA:             J=1:    1.
 TABLE(NI=1,NJ=1): C    CIRC:             J=1:    1.
@XQT ELD
 RESET NUTED=1
 K21
   1 2  1 8
 C21
 MATERIAL=COEF
   1 2  1 8
@XQT TGEO
@XQT TRTA
 RESET T1=0.0, T2=.0125, DT=.0025, NTERMS=5
 TEMP=100.
 TSAVE=.005
@XQT DCU
 PRINT 1 TRTA TIME
 PRINT 1 TRTA TEMP
 TOC 1
 STOP
```

```
@XQT TAB
 START 101
 JLOC: 1 0. 0. 0. 1. 0. 0. 101 1
@XQT AUS
 TABLE(NI=9,NJ=1): COND PROP 1: I=4: J=1:    959739.52
 TABLE(NI=3,NJ=1): RADI PROP 1: I=2: J=1:    1.
 TABLE(NI=1,NJ=1): K    AREA:        J=1:    1.
 TABLE(NI=1,NJ=1): R    CIRC:        J=1:    1.
 TABLE(NI=1,NJ=2): TEMP NODE:        J=1,2:  1. 101.
 TABLE(NI=1,NJ=2): APPL TEMP:        J=1,2:  1000. 100.
@XQT ELD
 RESET NUTED=1
 K21
   1 2  1 100
 R21
   1 2  1 100
@XQT TGEO
@XQT SSTA
 RESET PROB=LINEAR
@XQT SSTA
 RESET SBCON=1., NFAC=2, NITER=5
@XQT SSTA
 RESET SBCON=1., NFAC=2, NITER=8, START=KFAC
@XQT DCU
 PRINT 1 STAT TEMP
 TOC 1
 STOP
```

3.13 Steady-State Analysis of a Conducting Pipe

The pipe shown has prescribed temperatures at each end and is insulated along its length. Fluid enters the pipe at a temperature of 50 degrees. Runstreams are shown on pages 13-3 through 13-5 for models defined with various element types and combinations of element types.



pipe length =	15.
pipe conduction area =	2.5
pipe conductivity =	2.
pipe convection perimeter =	8.
pipe-fluid convection coefficient =	0.05
fluid specific heat =	1.
fluid mass flow rate =	5.
pipe temperature (x=0) =	100.
pipe temperature (x=15) =	250.

The analytical solution is given by

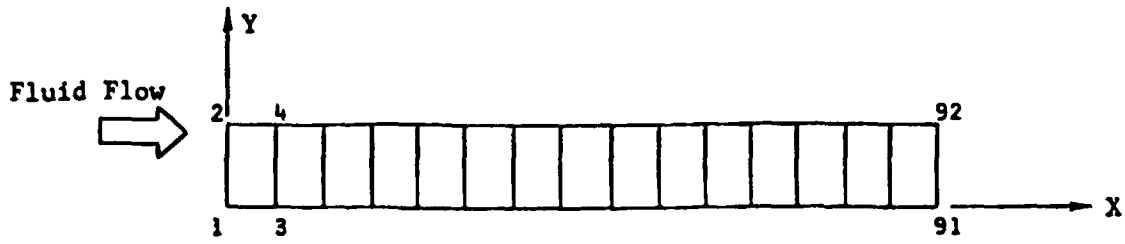
$$T_{pipe} = 4.757\, e^{0.2457x} + 35.009\, e^{-0.3257x} + 60.234$$

$$T_{fluid} = 1.169\, e^{0.2457x} - 11.402\, e^{-0.3257x} + 60.234$$

Results computed by SPAR are compared with the analytical solution below.

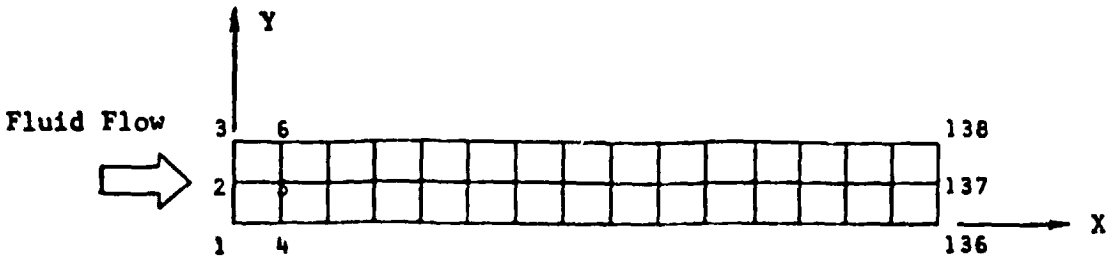
	Pipe Temperatures		Fluid Temperatures	
Pipe Location	Analytical Solution	SPAR Solution	Analytical Solution	SPAR Solution
0.	100.	100.00	50.	50.000
3.	83.353	83.346	58.385	58.414
6.	85.907	85.954	63.724	63.707
9.	105.521	105.49	70.296	70.312
12.	151.678	151.61	82.304	82.258
15.	250.	250.00	106.750	106.72

The finite element model and runstream for the pipe defined with integrated element type MT42 are shown below.



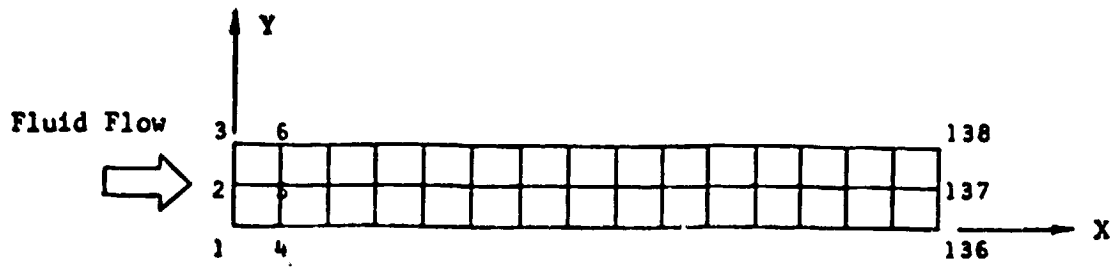

```
@XQT TAB
 START 92
 JLOC: 1 0. 0. 0. 15. 0. 0. 46 2 2
       1 0. 1. 0. 15. 1. 0.
@XQT AUS
 TABLE(NI=9,NJ=1):  COND PROP 1: I=4 5: J=1:    2. 2.
 TABLE(NI=9,NJ=1):  FLUI PROP 1: I=3 5: J=1:    1. .05
 TABLE(NI=3,NJ=1):  MT42 SECT:   I=1 2: J=1:    2.5 8.
 TABLE(NI=1,NJ=45): MTR  MT42 1:        J=1,45: 5.
 TABLE(NI=1,NJ=3):  TEMP NODE:          J=1,3:  2. 1. 91.
 TABLE(NI=1,NJ=3):  APPL TEMP:          J=1,3:  50. 100. 250.
@XQT ELD
 RESET NUTED=1
 MT42
  NMEDIUM=1: 2 4 3 1  1 45
@XQT TGEO
@XQT SSTA
@XQT DCU
 PRINT 1 STAT TEMP
 TOC 1
 STOP
```

The finite element model and runstream for the pipe defined with integrated element type MT62 are shown below.




```
@XQT TAB
 START 138
 JLOC: 1 0. 0. 0. 15. 0. 0. 46 3 2
       2 0. 1. 0. 15. 1. 0.
       2 0. .5 0. 15. .5 0. 46 3
@XQT AUS
 TABLE(NI=9,NJ=1):  COND PROP 1: I=4 5:      J=1:     2. 2.
 TABLE(NI=9,NJ=1):  FLUI PROP 1: I=3 5:      J=1:     1. .05
 TABLE(NI=5,NJ=1):  MT62 SECT:   I=1 2 3 4: J=1:     2.5 1. 3. 3.
 TABLE(NI=1,NJ=45): MTR  MT62 1:             J=1,45:  5.
 TABLE(NI=1,NJ=5):  TEMP NODE:               J=1,5:   2. 1. 3. 136. 138.
 TABLE(NI=1,NJ=5):  APPL TEMP:               J=1,5:   50. 100. 100. 250.  250.
@XQT ELD
 RESET NUTED=1
 MT62
  NMEDIUM=1: 2 1 1 45 2 3 0 1
@XQT TGEO
@XQT SSTA
@XQT DCU
 PRINT 1 STAT TEMP
 TOC 1
 STOP
```

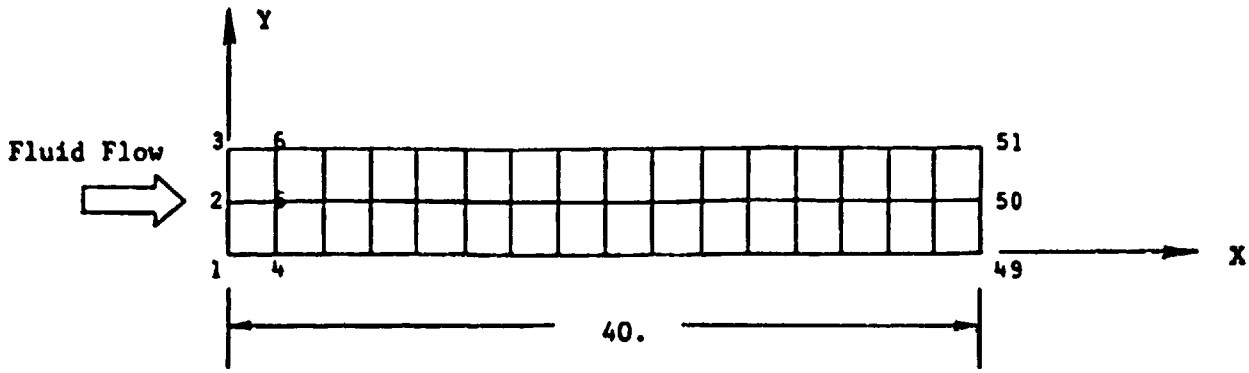
The finite element model and runstream for the pipe defined with a combination of element types C62, C42, K41, and MT21 are shown below.



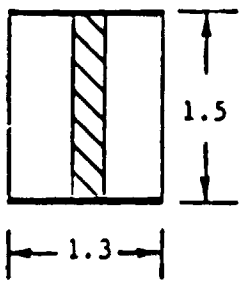

```
@XQT TAB
 START 138
 JLOC: 1 0. 0. 0. 15. 0. 0. 46 3 2
       2 0. 1. 0. 15. 1. 0.
       2 0. .5 0. 15. .5 0. 46 3
@XQT AUS
 TABLE(NI=9,NJ=1):  COND PROP 1: I=4 5:     J=1:    2. 2.
 TABLE(NI=9,NJ=1):  FLUI PROP 1: I=3 5:     J=1:    1. .05
 TABLE(NI=1,NJ=1):  K    THIC:              J=1:    2.5
 TABLE(NI=1,NJ=1):  C    CIRC:              J=1:    3.
 TABLE(NI=2,NJ=1):  C62  SECT:              J=1:    2.5 1.
 TABLE(NI=2,NJ=1):  MT21 SECT:   I=1:       J=1:    4.
 TABLE(NI=1,NJ=45): MTR  MT21 1:            J=1,45: 5.
 TABLE(NI=1,NJ=5):  TEMP NODE:              J=1,5:  2. 1. 3. 136. 138.
 TABLE(NI=1,NJ=5):  APPL TEMP:              J=1,5:  50. 100. 100. 250.  250.
@XQT ELD
 RESET NUTED=1
 C62
  NMEDIUM=1: 2 1 1 45 2 3 0 1
 C42
  NMEDIUM=1: 2 5 6 3  1 45
            2 5 4 1  1 45
 K41
            1 4 6 3  1 45
 MT21
  NMEDIUM=1: 2 5      1 45
@XQT TGEO
@XQT SSTA
@XQT DCU
 PRINT 1 STATIC TEMPERATURES
 TOC 1
 STOP
```

3.14 Steady-State Analysis of a Simple Heat Exchanger

The simple heat exchanger shown below has prescribed temperatures along its top and bottom walls. The fluid convection coefficient is a function of temperature. The runstream for the problem is shown on page 14-3.



Exchanger Cross Section:



exchanger length	= 40.
fin height	= 1.5
fin thickness	= 0.3
effective wall width	= 1.
top wall temperature	= 500.
bottom wall temperature	= 250.
fluid entry temperature	= 50.
fluid mass flow rate	= 50.
conduction coefficient	= 0.05
density	= 60.
specific heat	= 0.8
friction factor	= 0.01
viscosity	= 0.005
fluid convection coefficient (for $0 \leq T \leq 300$)	= 50/(375-T)

The analytical solution is given by

$$T = 50. + 6.25x \qquad 0 \leq x \leq 40$$

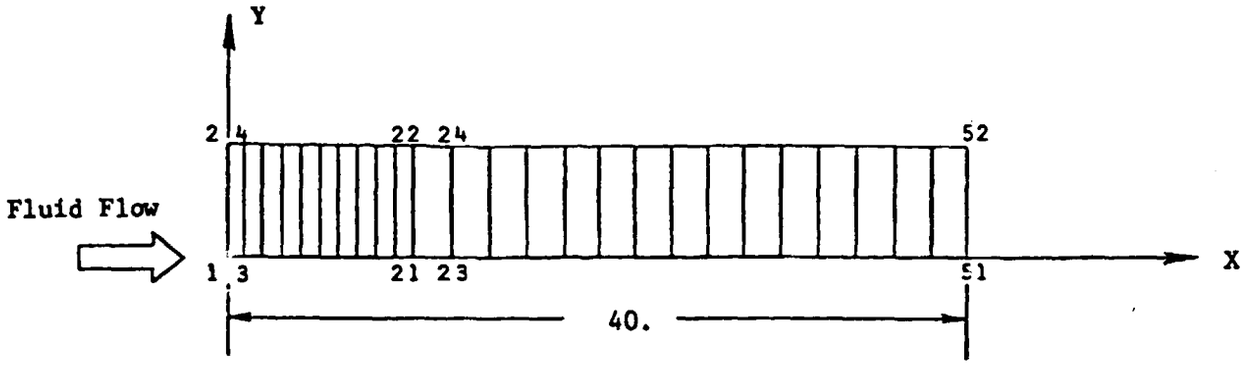
Results computed by SPAR are compared with the analytical solution below.

Exchanger Location	Analytical Solution	SPAR Solution
0.	50.	50.
2.5	65.625	66.007
5.	81.25	81.278
7.5	96.875	97.308
10.	112.5	112.58
12.5	128.125	128.62
15.	143.75	143.89
17.5	159.375	159.94
20.	175.	175.21
22.5	190.625	191.33
25.	206.25	206.54
27.5	221.875	222.72
30.	237.5	237.98
32.5	253.125	254.18
35.	268.75	269.52
37.5	284.375	285.81
40.	300.	301.19

```
@XQT TAB
  START 51
 JLOC: 1 0. 0.    0. 40. 0.    0. 17 3 2
       2 0. 1.5   0. 40. 1.5   0.
       2 0. 0.75  0. 40. 0.75  0. 17 3
@XQT AUS
 TABLE(NI=9,NJ=1):  COND PROP 1: I=4 5  :    J=1:      .05 .05
 TABLE(NI=9,NJ=12): FLUI PROP 1: I=1 3 5:    J=1,12:    50. .8 .1538
                                              75. .8 .1667
                                             100. .8 .1818
                                             125. .8 .2000
                                             150. .8 .2222
                                             175. .8 .2500
                                             200. .8 .2857
                                             225. .8 .3333
                                             250. .8 .4000
                                             275. .8 .5000
                                             300. .8 .6667
                                             400. .8 .6667
 TABLE(NI=5,NJ=1):  MT62 SECT:   I=1 2 3 4: J=1:      .3 1. 1. 1.
 TABLE(NI=1,NJ=16): MTR  MT62 1:             J=1,16:  50.
 TABLE(NI=1,NJ=35): TEMP NODE:               J=1:     2.
                                 DDATA=3.:   J=2,18:  1.
                                 DDATA=3.:   J=19,35: 3.
 TABLE(NI=1,NJ=35): APPL TEMP:               J=1:     50.
                                             J=2,18:  250.
                                             J=19,35: 500.

@XQT ELD
 RESET NUTED=1
 MT62
  NMEDIUM=1: 2 1 1 16 2 3 0 1
@XQT TGEO
@XQT SSTA
 RESET MFAC=1 NITER=5
   TEMP=100.
@XQT DCU
 PRINT 1 STAT TEMP
 TOC 1
 STOP
```

3.15 Steady-State Analysis of a Convectively Heated, Fluid Cooled Pipe with Temperature-Dependent Properties

The runstream for the problem described below is shown on page 15-3. A pipe is convectively heated by a surrounding medium at a temperature of 300 degrees and is cooled by a fluid flowing through the pipe. The fluid enters the pipe at a temperature of 50 degrees.



pipe length =	40.
pipe convection coefficient =	0.03472
pipe conduction area =	1.075
pipe convection perimeter =	6.494
fluid mass flow rate =	10.

The results computed by SPAR are compared with the solution contained in Reference 1 on the next page.

(1) Thornton, E. A., "TAP 1: A Finite Element Program for Steady-State Thermal Analysis of Convectively Cooled Structures", Technical Report 76-M6, School of Engineering, Old Dominion University, Norfolk, Virginia, 1976.

Pipe Node	Reference 1 Temperature	SPAR Temperature
1	50.	50.
2	146.758	146.75
3	53.563	53.563
4	147.163	147.15
5	57.156	57.156
6	148.136	148.13
7	60.566	60.567
8	149.442	149.43
9	64.038	64.039
10	150.943	150.93
11	67.341	67.343
12	152.553	152.54
13	70.721	70.722
14	154.226	154.22
15	73.934	73.936
16	155.930	155.92
17	77.231	77.233
18	157.649	157.64
19	80.361	80.363
20	159.370	159.36
21	83.579	83.582
22	161.089	161.08
23	89.733	89.736
24	164.535	164.53
25	95.811	95.814
26	167.957	167.95
27	101.665	101.67
28	171.388	171.39
29	107.436	107.44
30	174.835	174.83
31	112.965	112.97
32	178.237	178.24
33	118.422	118.43
34	181.575	181.57
35	123.645	123.65
36	184.832	184.83
37	128.809	128.81
38	188.015	188.02
39	133.740	133.74
40	191.113	191.11
41	138.625	138.63
42	194.136	194.14
43	143.279	143.28
44	197.064	197.06
45	147.895	147.90
46	199.883	199.88
47	152.274	152.28
48	202.503	202.50
49	156.604	156.61
50	204.701	204.70
51	160.627	160.63
52	205.757	205.76

1.2 Lists of Processor Subroutines

Comdecks

1 ADDK/14
2 ADDVEC/13
3 ALPHA/14
4 CDPSP/12
5 CEDOF/15
6 CFFC/16
7 CFFF/16
8 CFMDM1/15
9 CFMDM2/15
10 CFPNI/12
11 CHKFPS/13
12 CHKIDS/15
13 CHKRED/14
14 CLC /12
15 CPROP/13
16 CSLV/15
17 CSPDP/12
18 DERIV/15
19 EC21/13
20 EC31/12
21 EC32/13
22 EC41/12
23 EC42/13
24 EC62/13
25 EEPROP/14
26 EFLUX/15
27 ELTEMP/15
28 EPD /14
29 ERK /13
30 ER21/13
31 ER31/12
32 ER41/12
33 EXPFLX/16
34 EXPPD/14
35 EXPRK1/15
36 EXPRK2/15
37 EXPSCQ/16
38 EXPTEK/14
39 EXPTIN/14
40 FACIJ/13
41 FACMD/15
42 FACTOR/15
43 FEF62/14
44 FPROP/16
45 JFIX/13
46 LDIDJS/16
47 LDJS/13
48 LDKL/16
49 LDMP/16
50 LEXP/14
51 LINES/12
52 LOCT/14
53 MOVER/13
54 MPROD1/13
55 MPROD2/13
56 MPROD3/13
57 MPROP/16
58 MPTCHK/14
59 NUEC/15
60 OPN1/16
61 PDNM/14
62 QK21/13
63 QK31/13
64 QK41/13
65 QK61/12
66 QK81/12
67 REVEC/16
68 REXEX/14
69 RFLUX/15
70 SCPROD/13
71 VSUM1/13
72 VSUM2/13
73 WRKB/14
74 XSOL/15

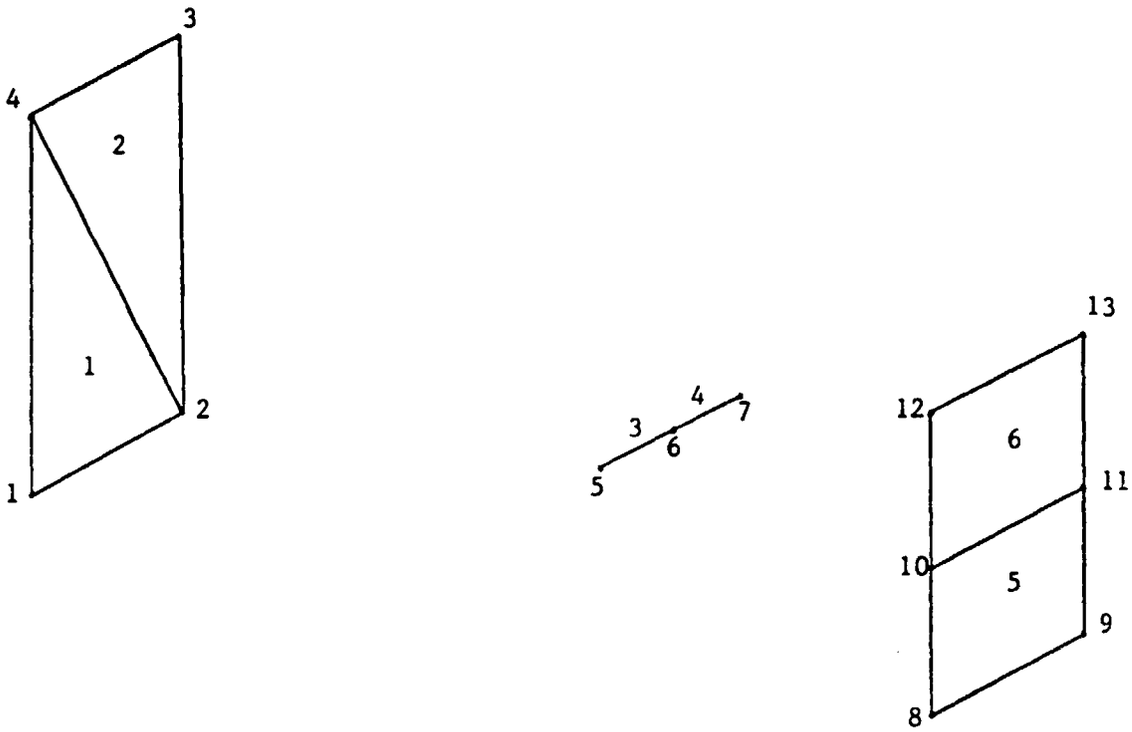
TGEO

1 MPTGEO/12
2 LDTGEO/14
3 READKL/14
4 READZL/14
5 FORMKL/15
6 GOTGEO/14
7 TGEOCS/13
8 EXTGEO/15
9 ELNMAT/14
10 EGREXP/13
11 EGR2/14
12 EGR3/12
13 EGR4/13
14 EGR6/12
15 EGR8/12
16 GE2D/13
17 GE3D/13
18 GEFACE/12
19 GELDG/13
20 GESMRY/13
21 TGEOID/13

```
@XQT TAB
 START 52
 JLOC:  1  0. 0.    0. 10. 0.    0. 11 2 2
        1  0. 1.11 0. 10. 1.11 0.
       23 12. 0.    0. 40. 0.    0. 15 2 2
        1 12. 1.11 0. 40. 1.11 0.
@XQT AUS
 TABLE(NI=9,NJ=4):  COND PROP 1: I=1 4:   J=1,4:   32. 2.208
                                                  212. 2.167
                                                  572. 2.083
                                                  932. 1.833
 TABLE(NI=9,NJ=7):  FLUI PROP 1: I=1 3 5: J=1,7:   32. 1.010 .0561
                                                   50. 1.000 .0584
                                                  100.  .998 .0640
                                                  200. 1.000 .0693
                                                  300. 1.030 .0694
                                                  400. 1.080 .0670
                                                  500. 1.190 .0614
 TABLE(NI=2,NJ=1):  CONV PROP 1: I=2:     J=1:    .03472
 TABLE(NI=1,NJ=1):  C    CIRC:           J=1:    6.494
 TABLE(NI=1,NJ=25): CTEM C21  1:          J=1,25: 300.
 TABLE(NI=3,NJ=1):  MT42 SECT:   I=1 2:   J=1:    1.075 6.494
 TABLE(NI=1,NJ=25): MTR  MT42 1:          J=1,25: 10.
 TABLE(NI=1,NJ=1):  TEMP NODE:            J=1:    1.
 TABLE(NI=1,NJ=1):  APPL TEMP:            J=1:    50.
@XQT ELD
 RESET NUTED=1
 MT42
  NMEDIUM=1: 1 3 4 2   1 25
 C21
            2 4       1 25
@XQT TGEO
@XQT SSTA
 RESET NFAC=1 NITER=5
 TEMP=300.
@XQT DCU
 PRINT 1 STAT TEMP
 TOC 1
 STOP
```

TA3.16 Steady-State and Transient Analysis of a 3-Body Radiation Exchange System

The runstream for the problem described below is shown on page 16-4. A radiation exchange system is comprised of a heater with a constant source heat rate and two plates with dissimilar absorption and reflection characteristics. Node points and elements are numbered as shown.



Radiation properties and exchange factors were chosen so that plate #1 would absorb heat faster than plate #2, and most of the heat incident on plate #2 would be reflected. The emitted and reflected heat not incident on either of the plates or heater is lost to space. Pertinent geometry, material, and excitation information and exchange factors are tabulated on the next page.

	Plate #1	Heater	Plate #2
Radiating Area	10.	2.	8.
Emissivity	0.6	0.85	0.1
Reflectivity	0.4	0.15	0.8
Thickness	0.1		0.1
Conducting Area		0.2	
Conductivity	200.	200.	200.
Mass Density	3000.	3000.	3000.
Specific Heat	0.2	0.2	0.2
Source Heat Rate		22680.	
Element Types	R31,K31	R21,K21	R41,K41

F13 = .04	F14 = .02	F15 = .2	F16 = .1
F23 = .02	F24 = .04	F25 = .1	F26 = .2
F31 = .2	F32 = .1	F35 = .16	F36 = .16
F41 = .1	F42 = .2	F45 = .16	F46 = .16
F51 = .25	F52 = .125	F53 = .04	F54 = .04
F61 = .125	F62 = .25	F63 = .04	F64 = .04

Steady-state analyses were performed for two values of the radiation convergence parameter (RESET control RCONV). The results are compared with the analytical solution below.

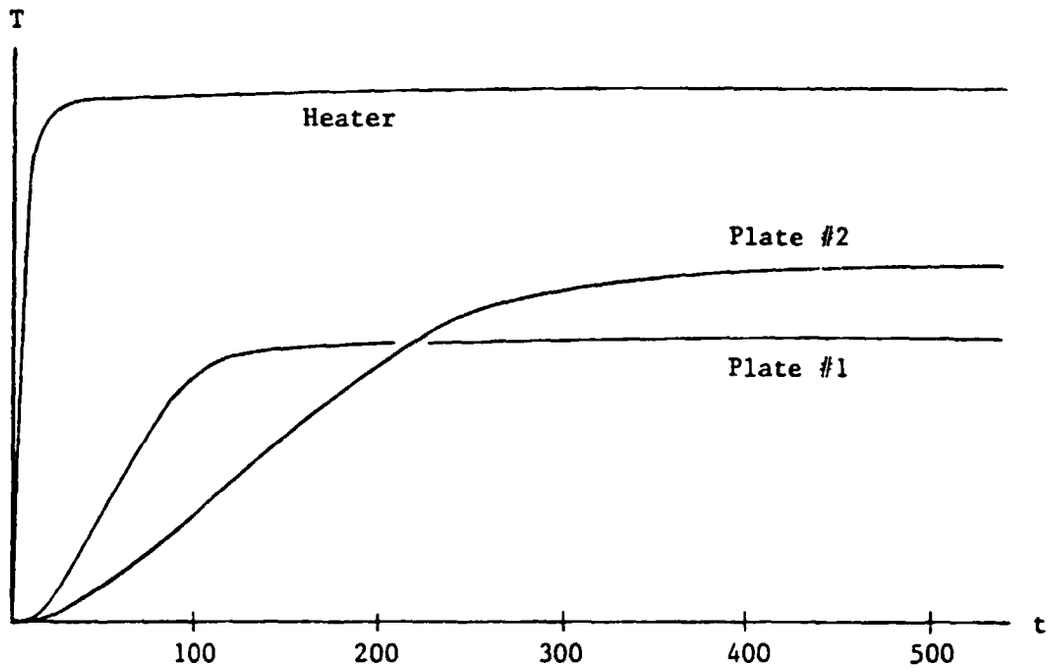
Node	SSTA Solutions RCONV=.01 (default)	SSTA Solutions RCONV=.0001	Analytical Solution
1	300.61	300.06	300.02
2	300.61	300.06	300.02
3	300.61	300.06	300.02
4	300.61	300.06	300.02
5	562.01	561.95	561.94
6	562.01	561.95	561.94
7	562.01	561.95	561.94
8	375.78	375.04	374.98
9	375.78	375.04	374.98
10	375.78	375.04	374.98
11	375.78	375.04	374.98
12	375.78	375.04	374.98
13	375.78	375.04	374.98

A transient analysis was performed using the default value of the radiation convergence parameter. The results are illustrated below. The temperature versus time curves appear to be reasonable, given the radiation properties and exchange factors of the system, i.e.,

1. The heater temperature has a very short rise time and quickly approaches its steady-state value.

2. Plate #1 heats up much faster than plate #2 because it is a more efficient absorber of heat.

3. Because plate #2 reflects most of the heat incident on it, the temperature of plate #2 continues to rise well after the temperature of plate #1 has reached its steady-state value.



```
@XQT TAB
 START 13
 JOINT LOCATIONS
     1  0.0 0.0 -2.5  0.0 2.0 -2.5  2 1
     3  0.0 2.0  2.5  0.0 0.0  2.5  2 1
     5  6.0 0.0  0.0  6.0 2.0  0.0  3 1
     8 10.0 0.0 -2.0 10.0 2.0 -2.0  2 1 3
     2 10.0 0.0  2.0 10.0 2.0  2.0
@XQT AUS
 TABLE(NI=9,NJ=1): COND PROP: I=2 3 4 5: J=1: 3000. .2 200. 200.
 TABLE(NI=1,NJ=1): K    AREA: J=1:    .2
 TABLE(NI=1,NJ=1): K    THIC: J=1:    .1
 TABLE(NI=1,NJ=2): SOUR K21 1:J=1,2: 22680.
 TABLE(NI=2,NJ=3): RADI COEF: J=1,3: .6 .4 .85 .15 .1 .8
 TABLE(NI=1,NJ=1): R    CIRC: J=1:    1.0
 TABLE(NI=9,NJ=6): REX  VF 1: J=1,6: 1. 3. .04  4. .02  5. .2  6. .1
                                     2. 3. .02  4. .04  5. .1  6. .2
                                     3. 1. .2   2. .1   5. .16 6. .16
                                     4. 1. .1   2. .2   5. .16 6. .16
                                     5. 1. .25  2. .125 3. .04 4. .04
                                     6. 1. .125 2. .25  3. .04 4. .04
@XQT ELD
 RESET NUTED=1
 K31: 4 1 2: 2 3 4
 K21: 5 6: 6 7
 K41: 8 9 11 10: 10 11 13 12
 R31: MATERIAL=COEF: NMAT=1: 4 1 2: 2 3 4
 R21: MATERIAL=COEF: NMAT=2: 5 6: 6 7
 R41: MATERIAL=COEF: NMAT=3: 8 9 11 10: 10 11 13 12
@XQT TGEO
@XQT SSTA
 RESET RLIB=1, NFAC=2, NITER=6
 TEMP=400.
@XQT DCU
 PRINT 1 STAT TEMP
@XQT TRTA
 RESET RLIB=1, T1=0., T2=50., RI=10.
 TSAVE=10.
 TEMP=0.
@XQT DCU
 PRINT 1 TRTA TIME: PRINT 1 TRTA TEMP
@XQT TRTA
 RESET RLIB=1, T1=50., T2=200., RI=25.
 TSAVE=25.
@XQT DCU
 PRINT 1 TRTA TIME: PRINT 1 TRTA TEMP
@XQT TRTA
 RESET RLIB=1, T1=200., T2=600., RI=50.
 TSAVE=50.
@XQT DCU
 PRINT 1 TRTA TIME: PRINT 1 TRTA TEMP
 TOC 1
 STOP
```

Reference Manual

SPAR Thermal Analysis Processors

Volume 4

Experimental Thermal Element Capability

CONTENTS

VOLUME TA4

SPAR THERMAL ANALYZER

EXPERIMENTAL ELEMENT CAPABILITY

Section 1 INTRODUCTION

To use the experimental element capability, the user must develop the following subroutines and incorporate them into the indicated SPAR processors:

NAME	PROCESSOR	FUNCTION
EXPTIN	SSTA & TRTA	Used to initialize user defined labeled common blocks at beginning of execution
EXPTEK	SSTA & TRTA	Used to construct element K matrices
EXPSCQ	SSTA & TRTA	Used to construct element thermal load vectors
EXPFLX	SSTA & TRTA	Used to compute element heat fluxes
EXPPD	SSTA & TRTA	Used to compute element fluid pressures.

Calling sequences and other details are discussed in Section 4.

After the above routines have been incorporated into SPAR, the user must execute the steps described below to perform analyses using experimental elements. In the following example it is assumed that there are two types of experimental elements, "KC82" and "R47Z". (Any number of distinct experimental element names are permitted.)

1) Via TAB define all node point locations

2) Via AUS/TABLE define all tables required for the standard elements plus the following tables required for experimental elements (see Section 2).

Data Set Name	Contents
KC82 PROP 1 KC82 PROP 2	Temperature dependent material properties for KC82 elements, materials 1 and 2
KC82 SECT	Section properties for KC82 elements
SOUR KC82 1 1	Source loading for KC82 elements, group 1

```
CTEM KC82 2 1        Convective exchange temperatures for KC82
                     elements, group 2

R47Z PROP 1          Temperature-dependent material properties
                     for R47Z elements, material 1

R47Z SECT            Section properties for R47Z elements
```

3) Via ELD, define standard and experimental elements as follows:

```
@XQT ELD
K21                          Define all K21 elements, usual ELD
 .                           format.
 .
 .
TEXP KC82 2 1 8 1 1 0        Initiate KC82 element definition,
1 3 23 21 2 13 22 11         these elements have 8 node points,
3 5 25 23 4 15 24 13         1 degree of freedom at each node
 .                           point, and 1 flux quantity.
 .
 .
 .
NMAT=2
 .
 .
 .
GROUP=2
NMAT=1
NSECT=2
 .
 .
 .
TEXP R47Z 2 1 4 1 4 1        Initiate R47Z element definition.
57 62 84 83                  These elements have 4 node points,
62 67 85 84                  1 degree of freedom at each node
 .                           point, and 4 flux quantities.
 .                           They are radiating elements.
 .
NSECT=2
56 61 62 57
55 60 61 56
 .
 .
 .
```

4)Processors TGEO, SSTA, and TRTA are executed as usual.

Section 2 EXPERIMENTAL ELEMENT DATA SETS

2-1 Material Property Data Sets

Temperature-dependent material properties are defined for experimental element types and material numbers via the following AUS/TABLE constructed data sets:

N1	N2	N3	Data Line Contents
lname	PROP	nmat	T,P(1),P(2),...,P(NI-1)

lname is the experimental element name, and nmat is the material number defined via the ELD material pointer NMAT. A data set must exist for each element type and material number.

Each line defines properties corresponding to temperature T. Properties at intermediate temperatures are determined by linear interpolation. If only one data line is given (NJ=1), properties are constant.

2-2 Section Property Data Sets

Section properties for experimental elements are input via AUS/TABLE data sets as follows:

N1	N2	Data Line Contents
lname	SECT	S(1), S(2),...,S(NI)

Lname is the experimental element name. Each line corresponds to the ELD section properties pointer NSECT.

A section properties data set must exist for each experimental element type unless the section properties pointer NSECT=0.

2-3 Thermal Excitation Data Sets

The data sets used to define thermal excitation are

N1	N2	N3	N4	Data Line Contents
SOUR	lname	ngrp	iset	source heat rates
CTEM	lname	ngrp	iset	convective exchange temperatures
MTR	lname	ngrp		mass-transport rates
P1	lname	ngrp		inlet pressures

Lname, ngrp, and iset are the element name, element group, and load set respectively. The TOC parameters are NI=number of load quantities associated with the experimental element lname and NJ=number of elements in group ngrp.

For steady-state analyses each block corresponds to one loading case. For transient analyses the k-th block corresponds to time t(k) in the associated time data set SOUR TIME, CTEM TIME, or MTR TIME as appropriate.

2-4 Output Data Sets

The processors SSTA and TRTA generate the data sets

Data Set Name	NWDS	NJ	NI*NJ	Contents
STAT TEMP iset	n*lrt	njts	lrt	steady state solutions
TRTA TEMP iset	n*lrt	njts	lrt	transient solutions
SFLX lname ngrp iset	n*ne*ni	ne	ne*ni	steady state heat fluxes
SPD lname ngrp iset	n*ne*np	ne	ne*np	steady state pressures
TFLX lname ngrp iset	n*ne*ni	ne	ne*ni	transient heat fluxes
TPD lname ngrp iset	n*ne*np	ne	ne*ni	transient pressures

where n = number of load cases for steady state analyses
number of save times for transient analyses
njts = number of node points in structure
lrt = length of solution matrix
= njts*mxndf, where mxndf is the number of freedoms at each node point
ne = number of elements in group ngrp
ni = number of element flux quantities, defined in ELD
np = number of pressure quantities

Solution matrices have the form T(J,I) where I is the node point number and J is the degree of freedom index. Nodal temperatures are always in T(1,I) so that temperature-dependent material properties can be evaluated for standard thermal elements. Quantities represented by T(2,I), T(3,I), etc. are defined by the user.

The number of freedoms at each node point (mxndf) is set equal to the maximum number of freedoms defined during ELD generation of experimental elements.

Section 3 ELD INPUT OF EXPERIMENTAL ELEMENTS

The subprocessor input image for experimental elements is

TEXP lname major minor npts ndf nflux irad imtr npd

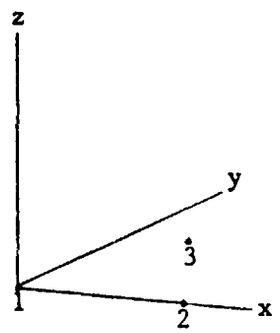
lname = element name. This can be anything except an existing element name or the first word in an existing data set name.
major = element type parameter. It indicates which ELD mesh generation scheme to use for element definition. It is also utilized as an argument in the user written experimental element subroutines to specify whether the element coordinates are globally or locally defined.
- = 0, element type not defined. ELD mesh generation is not used to generate elements. Element coordinates are set equal to the appropriate global coordinates. No geometry checks are performed.
- = 1, 1-dimensional ELD mesh generation can be used to generate elements. Local coordinates are constructed as described below.
- = 2, 2-dimensional ELD mesh generation can be used to generate elements. Local coordinates are constructed as described below.
- = 3, 3-dimensional ELD mesh generation can be used to generate elements. Local coordirates are constructed as described below.

minor = user defined parameter - can be used for any purpose. For example, minor can be used to control print out during the check-out phase of an experimental element formulation.
npts = number of element node points, an element can have any number of node points.
ndf = degrees of freedom at each element node point. ndf can be any number for any element however, the meaning of each degree of freedom must be the same from element to element. The default value is 1.
nflux = number of flux quantities associated with the element. The default value is 0. The flux computation routine EXPFLX is called only for those elements for which nflux >0.
irad = radiation indicator, default value is 0.
0, no radiation effects
1, radiation effects present
imtr = mass-transport rate indicator, default value is 0.
0, no mass-transport
1, mass-tranport element
npd = number of pressure quantities associated with mass-transport elements.

The material properties pointer NMAT indicates the appropriate material properties data set (see Section 2-1). NMAT must be a positive integer.

The section properties pointer NSECT indicates the appropriate section properties line in data set lname SECT (section 2-2). If NSECT=0, it is assumed that no section properties are defined for experimental element type lname.

If major >0, a right handed element coordinate system is constructed for each element as shown below.



The following geometry checks are performed:

1) If nodes 1 and 2 are coincident, an error message is printed unless major= 1 and npts = 2 in which case a warning message is printed.
2) If major = 2 or 3 and node points 1, 2, and 3 are colinear, an error message is printed.
3) If major = 3, no further geometry checks are performed.

Section 4 EXPERIMENTAL ELEMENT SUBROUTINES

```
      SUBROUTINE EXPTIN
C     12/77 MBM
C
C     USER WRITTEN INITIALIZATION ROUTINE
C     EXPTIN IS USED TO INITIALIZE DATA RESIDING IN USER
C     DEFINED LABELED COMMON BLOCKS IN SUBROUTINES EXPTEK,
C     EXPSCQ, AND EXPFLX.  THE SUBROUTINE IS CALLED AT THE
C     BEGINNING OF EXECUTION OF PROCESSORS SSTA AND TRTA.
C     COMMON BLOCKS SHOULD HAVE NAMES
C     CEXPT1, CEXPT2, ETC.
C
      RETURN
      END
```

```
      SUBROUTINE EXPTEK(LNAME,MAJOR,MINOR,NPTS,NDF,SBCON,
     $                  T,PROP,TDMP,SECT,X,TR,
     $                  KTYPE,KFORM,EK)
C     12/77 MBM
      DIMENSION T(NDF,NPTS),PROP(50),TDMP(1),SECT(1),X(3,NPTS),EK(1)
C
C     USER WRITTEN ROUTINE FOR COMPUTING ELEMENT K MATRICES
C     LNAME = ELEMENT NAME                                  (SET IN ELD)
C     MAJOR = ELEMENT TYPE                                  (SET IN ELD)
C             0, NOT DEFINED   (SEE X BELOW)
C             1, 1-DIMENSIONAL
C             2, 2-DIMENSIONAL
C             3, 3-DIMENSIONAL
C     MINOR = USER ID PARAMETER                             (SET IN ELD)
C     NPTS  = NUMBER OF NODE POINTS                         (SET IN ELD)
C     NDF   = DEGREES OF FREEDOM AT EACH NODE POINT         (SET IN ELD)
C     SBCON = STEFAN-BOLTZMANN CONSTANT
C     T     = ELEMENT TEMPERATURE MATRIX
C     PROP  = MATERIAL PROPERTIES VECTOR, FILLED ACCORDING TO
C             THE SEQUENCE:
C             1) TEMP = CHARACTERISTIC TEMPERATURE  COMPUTED
C                FROM ELEMENT TEMPERATURE MATRIX T
C             2) CALL CPROP(TDMP,TEMP) CAUSES PROP TO BE FILLED WITH
C                MATERIAL PROPERTIES EVALUATED AT TEMPERATURE TEMP
C     TDMP  = VECTOR WHICH CONTAINS TEMPERATURE-DEPENDENT MATERIAL
C             PROPERTIES, USED TO FILL PROP ARRAY AS DESCRIBED ABOVE
C     SECT  = ELEMENT SECTION PROPERTIES STRING
C     X     = ELEMENT LOCAL COORDINATE ARRAY IF MAJOR>0
C             ELEMENT GLOBAL COORDINATE ARRAY IF MAJOR=0
C     TR    = MASS-TRANSPORT RATE
C     KTYPE = TYPE OF MATRIX REQUESTED
C             0, NO MATRIX REQUESTED - EXPTEK IS CALLED AT THE
C                BEGINNING OF ASSEMBLY OF EACH ELEMENT TYPE WITH
C                KTYPE = 0 TO ALLOW FOR USER INITIALIZATION OF DATA
C             1, SYMMETRIC CONDUCTION
C             2, SYMMETRIC CONVECTION
C             3, SYMMETRIC RADIATION
C             4, ASYMMETRIC MASS-TRANSPORT
C             5, SYMMETRIC CAPACITANCE
C     KFORM = FORM OF MATRIX RETURNED
C             0, NO MATRIX RETURNED
C             1, FULL MATRIX RETURNED
C     EK    = ELEMENT K MATRIX,  DIMENSIONED EK(NDF*NPTS,NDF*NPTS)
C             1) FULL MATRICES MUST HAVE THE FORM EK(I,J), WITH
C                I AND J SEQUENCED ACCORDING TO NDF*(N-1) + K,
C                WHERE N = 1,...,NPTS AND K = 1,...,NDF
C             2) FOR TRANSIENT ANALYSES, FULL  RADIATION AND
C                CAPACITANCE MATRICES ARE CONVERTED TO DIAGONAL
C                MATRICES BY SUMMING TERMS ROW BY ROW.
C
      KFORM = 0
      RETURN
      END
```

```
      SUBROUTINE EXPSCQ(LNAME,MAJOR,MINOR,NPTS,NDF,
     $                  T,PROP,TDMP,SECT,X,SCQ,
     $                  LTYPE,ITLV,ETLV)
C     12/77 MBM
      DIMENSION T(NDF,NPTS),PROP(50),TDMP(1),SECT(1),X(3,NPTS),
     $          SCQ(1),ETLV(NDF,NPTS)
C
C     USER WRITTEN ROUTINE FOR COMPUTING ELEMENT THERMAL LOAD VECTORS
C     LNAME = ELEMENT NAME                                (SET IN ELD)
C     MAJOR = ELEMENT TYPE                                (SET IN ELD)
C             0, NOT DEFINED   (SEE XBELOW)
C             1, 1-DIMENSIONAL
C             2, 2-DIMENSIONAL
C             3, 3-DIMENSIONAL
C     MINOR = USER ID PARAMETER                           (SET IN ELD)
C     NPTS  = NUMBER OF NODE POINTS                       (SET IN ELD)
C     NDF   = DEGREES OF FREEDOM AT EACH NODE POINT       (SET IN ELD)
C     T     = ELEMENT TEMPERATURE VECTOR
C     PROP  = MATERIAL PROPERTIES VECTOR, FILLED ACCORDING TO
C             THE SEQUENCE:
C             1) TEMP = CHARACTERISTIC TEMPERATURE  COMPUTED
C                FROM ELEMENT TEMPERATURE MATRIX T
C             2) CALL CPROP(TDMP,TEMP) CAUSES PROP TO BE FILLED WITH
C                MATERIAL PROPERTIES EVALUATED AT TEMPERATURE TEMP
C     TDMP  = VECTOR WHICH CONTAINS TEMPERATURE-DEPENDENT MATERIAL
C             PROPERTIES, USED TO FILL PROP ARRAY AS DESCRIBED ABOVE
C     SECT  = ELEMENT SECTION PROPERTIES STRING
C     X     = ELEMENT LOCAL COORDINATE ARRAY IF MAJOR>0
C             ELEMENT GLOBAL COORDINATE ARRAY IF MAJOR=0
C     SCQ   = ELEMENT THERMAL LOADING STRING - CONTAINS SOURCE
C             STRENGTHS OR CONVECTIVE EXCHANGE TEMPERATURES
C             ACCORDING TO VALUE OF LTYPE
C     LTYPE = TYPE OF THERMAL LOAD VECTOR REQUESTED
C             0, NO LOAD VECTOR REQUESTED
C             1, SOURCE
C             2, CONVECTIVE EXCHANGE
C     ITLV  = ELEMENT THERMAL LOAD VECTOR INDICATOR
C             0, NO VECTOR RETURNED
C             1, VECTOR RETURNED
C     ETLV  = ELEMENT THERMAL LOAD VECTOR
C
      ITLV  = 0
      RETURN
      END
```

```
      SUBROUTINE EXPFLX(LNAME,MAJOR,MINOR,NPTS,NDF,SBCON,
     $                  T,PROP,TDMP,SECT,X,CET,TR,
     $                  FLUX,NFLUX)
C     12/77 MBM
      DIMENSION T(NDF,NPTS),PROP(50),TDMP(1),SECT(1),X(3,NPTS),
     $          CET(1),FLUX(NFLUX)
C
C     USER WRITTEN ROUTINE FOR COMPUTING ELEMENT FLUXES
C     LNAME = ELEMENT NAME                                (SET IN ELD)
C     MAJOR = ELEMENT TYPE                                (SET IN ELD)
C             0, NOT DEFINED   (SEE X BELOW)
C             1, 1-DIMENSIONAL
C             2, 2-DIMENSIONAL
C             3, 3-DIMENSIONAL
C     MINOR = USER ID PARAMETER                           (SET IN ELD)
C     NPTS  = NUMBER OF NODE POINTS                       (SET IN ELD)
C     NDF   = DEGREES OF FREEDOM AT EACH NODE POINT       (SET IN ELD)
C     SBCON = STEFAN-BOLTZMANN CONSTANT
C     T     = ELEMENT TEMPERATURE VECTOR
C     PROP  = MATERIAL PROPERTIES VECTOR, FILLED ACCORDING TO
C             THE SEQUENCE:
C             1) TEMP = CHARACTERISTIC TEMPERATURE  COMPUTED
C                FROM ELEMENT TEMPERATURE MATRIX T
C             2) CALL CPROP(TDMP,TEMP) CAUSES PROP TO BE FILLED WITH
C                MATERIAL PROPERTIES EVALUATED AT TEMPERATURE TEMP
C     TDMP  = VECTOR WHICH CONTAINS TEMPERATURE-DEPENDENT MATERIAL
C             PROPERTIES, USED TO FILL PROP ARRAY AS DESCRIBED ABOVE
C     SECT  = ELEMENT SECTION PROPERTIES STRING
C     X     = ELEMENT LOCAL COORDINATE ARRAY IF MAJOR>0
C             ELEMENT GLOBAL COORDINATE ARRAY IF MAJOR=0
C     CET   = CONVECTIVE EXCHANGE TEMPERATURES
C     TR    = MASS-TRANSPORT RATE
C     FLUX  = ELEMENT FLUXES
C     NFLUX = NUMBER OF ELEMENT FLUXES                    (SET IN ELD)
C
      DO 10 I=1,NFLUX
   10 FLUX(I) = 0.0
      RETURN
      END
```

```
      SUBROUTINE EXPPD(LNAME,MAJOR,MINOR,NPTS,NDF,
     $                 T,PROP,TDMP,SECT,TR,
     $                 PRES,NPRES)
C       6/79 MBM
      DIMENSION T(NDF,NPTS),PROP(50),TDMP(1),SECT(1),X(3,NPTS),
     $          PRES(1)
C     LNAME = ELEMENT NAME                                (SET IN ELD)
C     MAJOR = ELEMENT TYPE                                (SET IN ELD)
C             0, NOT DEFINED   (SEE X BELOW)
C             1, 1-DIMENSIONAL
C             2, 2-DIMENSIONAL
C             3, 3-DIMENSIONAL
C     MINOR = USER ID PARAMETER                           (SET IN ELD)
C     NPTS  = NUMBER OF NODE POINTS                       (SET IN ELD)
C     NDF   = DEGREES OF FREEDOM AT EACH NODE POINT       (SET IN ELD)
C     T     = ELEMENT TEMPERATURE VECTOR
C     PROP  = MATERIAL PROPERTIES VECTOR, FILLED ACCORDING TO
C             THE SEQUENCE:
C             1) TEMP = CHARACTERISTIC TEMPERATURE  COMPUTED
C                FROM ELEMENT TEMPERATURE MATRIX T
C             2) CALL CPROP(TDMP,TEMP) CAUSES PROP TO BE FILLED WITH
C                MATERIAL PROPERTIES EVALUATED AT TEMPERATURE TEMP
C     TDMP  = VECTOR WHICH CONTAINS TEMPERATURE-DEPENDENT MATERIAL
C             PROPERTIES, USED TO FILL PROP ARRAY AS DESCRIBED ABOVE
C     SECT  = ELEMENT SECTION PROPERTIES STRING
C     X     = ELEMENT LOCAL COORDINATE ARRAY IF MAJOR>0
C             ELEMENT GLOBAL COORDINATE ARRAY IF MAJOR=0
C     PRES  = ELEMENT PRESSURES
C     NPRES = NUMBER OF ELEMENT PRESSURE QUANTITIES
C
      RETURN
      END
```

Reference Manual

SPAR Thermal Analysis Processors

Volume 5

Programmer Reference

CONTENTS

Section

Section 1 - PROCESSOR CONTENT

1.1 Maps

```
SEG TGEO
 IN T2.XXTGEO
 IN T2.MPTGEO
 IN R2.IMSYS
 IN T1.OPN1
   SEG LDTGEO*,(TGEO)
    IN R2.CREAD
    IN R2.RSET
    IN T2.LDTGEO
    IN T2.READKL
    IN T2.READZL
    IN T2.FORMKL
   SEG GOTGEO*,(TGEO)
    IN T2.GOTGEO
    IN T2.TGEOCS
    IN T2.EXTGEO
    IN T1.LINES
    IN T1.CHKFPS
    IN T1.JFIX
    IN T2.ELNMAT
    IN T1.CFPNI
    IN T2.EGREXP
    IN T2.EGR2
    IN T2.EGR3
    IN T2.EGR4
    IN T2.EGR6
    IN T2.EGR8
    IN T2.GE2D
    IN T2.GE3D
    IN T2.GEFACE
    IN T2.GELDG
    IN T2.GESMRY
    IN T2.TGEOID
      SEG BC*,()
       IN BLANK$COMMON
END
```

```
SEG SSTA
 IN T3.XXSSTA
 IN T3.MPSSTA
 IN R2.IMSYS
 IN T1.OPN1,.LDJS,.LINES,.CFPNI,.CEDOF,.LEXP,.CHKFPS,.JFIX
 IN T1.CHKIDS,.MOVER,.ELTEMP,.MPROP,.LOCT,.FPROP
 IN T1.ALPHA,.CFFC,.FEF62,.EEPROP,.CPROP
    SEG LDSSTA*,(SSTA)
     IN T3.LDSSTA
     IN R2.RSET,.CREAD
     IN T1.NUEC,.EXPTIN
     IN T3.STSSTA
     IN T1.LDKL,.LDMP,.MPTCHK,.LDIDJS
     IN T3.LDKSD
     IN T1.CLC,.CSLV
     IN T3.PRBTYP,.LCCR
     IN T1.CHKRED
     IN T3.SSTACA
    SEG GOSSTA*,(SSTA)
     IN T3.GOSSTA,.EXSSTA,.ASMFAC
     IN T1.REXEX,.ADDVEC
        SEG ASMBL*,(GOSSTA)
         IN T3.ASMBL
         IN T1.WRKB,.ERK,.ADDK
         IN T3.SDKM,.ATKM
        SEG NLTLV*,(GOSSTA)
         IN T1.REVEC
         IN T3.NLTLV,.ADDKLD,.ADDCLD,.ADDRLD,.NLATL
        SEG LTLV*,(GOSSTA)
         IN T3.LTLV,.LATL
         IN T1.CFMDM2
            SEG EK1*,(ASMBL,NLTLV,LTLV)
             IN T3.EK21,.EK31,.EK41
             IN T1.EC21,.EC31,.EC41
             IN T1.ER21,.ER31,.ER41
             IN T3.EMT21,.EMT42,.EMT62
             IN T1.EC32,.EC42,.EC62
             IN T1.QK21,.QK31,.QK41
            SEG EK2*,(ASMBL,NLTLV,LTLV)
             IN T3.EK61,.EK81
             IN T1.DERIV,.QK61,.QK81
            SEG EK3*,(ASMBL,NLTLV,LTLV)
             IN T1.EXPTEK,.EXPRK2,.EXPSCQ
        SEG FACSOL*,(GOSSTA)
         IN T1.FACTOR,.FACIJ,.FACMD,.SCPROD
         IN T3.TSOL
         IN T1.CSPDP,.XSOL,.CDPSP
         IN T3.CSTVEC,.CLSVEC
        SEG FLUX*,(GOSSTA)
         IN T3.SFLUX
         IN T1.EFLUX,.CFFF,.EXPFLX
         IN T3.SMTPD
         IN T1.EPD,.PDNM,.EXPPD
         IN T3.SRFLUX
         IN T1.RFLUX
        SEG BC*,()
         IN BLANK$COMMON
END
```

```
SEG TRTA
IN T4.XXTRTA
IN T4.MPTRTA
IN R2.IMSYS
IN T1.OPN1,.CHKFPS,.JFIX,.CFPNI,.CHKIDS,.LINES,.CEDOF,.WRKB
IN T1.ELTEMP,.MPROP,.LOCT,.FPROP,.ALPHA,.CFFC,.FEF62
IN T1.LEXP,.EEPROP,.CPROP,.MOVER,.SCPROD,.LDJS
   SEG LDTRTA*,(TRTA)
    IN T4.LDTRTA,.IETIM
    IN R2.RSET,.CREAD
    IN T1.NUEC
    IN T1.EXPTIN
    IN T4.STTRTA,.CTCS
    IN T1.LDKL,.LDMP,.MPTCHK,.LDIDJS
    IN T4.CTIMES,.TIMINC,.LDTIME,.LDAND,.MPTIME
    IN T1.CHKRED
    IN T4.TRTACA,.CSLVIM,.CSLVAN,.NANPTS
    IN T1.CLC,.CSLV
   SEG GOTRTA*,(TRTA)
    IN T4.GOTRTA,.EXTRTA,.INTEMP,.WRRDS,.CTI2
    IN T1.REXEX
    IN T4.LTPT,.LOCPT,.NEXTKB,.TPROP,.RDPQ,.WPROP,.JTCOMP
    IN T1.EC21,.EC31,.EC41,.QK21,.QK31,.QK41,.QK61,.QK81
    IN T1.CFMDM1,.CFMDM2,.ADDVEC
    IN T1.FACTOR,.FACIJ,.FACMD,.CSPDP,.XSOL,.CDPSP
       SEG ASMKCR*,(GOTRTA)
        IN T4.ASMKCR,.AMTR,.NADFS,.ADDKAN,.ASMEKS,.JTPACK
        IN T1.ADDK
           SEG EK1*,(ASMKCR)
            IN T4.TEK21,.TEK31,.TEK41,.TEK61,.TEK81
            IN T1.DERIV
            IN T1.ER21,.ER31,.ER41
            IN T4.TEMT21,.TEMT42,.TEMT62
            IN T1.EC32,.EC42,.EC62
           SEG EK2*,(ASMKCR)
            IN T1.EXPTEK,.EXPSCQ,.EXPRK1
       SEG TTLV*,(GOTRTA)
        IN T4.SAVESM,.INVCAP,.DTCOMP,.DTK3
        IN T4.FACKAN,.TTLV,.CSCQ,.CCDT,.FACKIM
        IN T1.REVEC
       SEG TTDS*,(GOTRTA)
        IN T4.TEMPAN,.MODKAN,.SAVET
        IN T4.TTDS,.MTPROD,.RTDT,.TTDSAN,.TTDSIM,.MODKIM
        IN T1.MPROD1,.MPROD2,.MPROD3,.VSUM1,.VSUM2
       SEG FLUX*,(GOTRTA)
        IN T4.TFLUX,.CMPTS,.FTCOEF,.LDTAF
        IN T1.EFLUX,.CFFF,.EXPFLX
        IN T4.TMTPD
        IN T1.EPD,.PDNM,.EXPPD
        IN T4.TRFLUX
        IN T1.RFLUX
       SEG BC*,()
        IN BLANK$COMMON
END
```

SSTA

1 MPSSTA/16
2 LDSSTA/16
3 STSSTA/15
4 LDKSD/15
5 PRBTYP/16
6 LCCR/15
7 SSTACA/15
8 GOSSTA/13
9 EXSSTA/16
10 ASMFAC/15
11 ASMBL/15
12 SDKM/15
13 ATKM/15
14 NLTLV/16
15 ADDKLD/13
16 ADDCLD/13
17 ADDRLD/15
18 NLATL/14
19 LTLV/16
20 LATL/14
21 EK21/13
22 EK31/13
23 EK41/13
24 EMT21/14
25 EMT42/14
26 EMT62/15
27 EK61/12
28 EK81/15
29 TSOL/15
30 CSTVEC/15
31 CLSVEC/13
32 SFLUX/16
33 SMTPD/14
34 SRFLUX/14

TRTA

1 MPTRTA/16
2 LDTRTA/16
3 IETIM/16
4 STTRTA/16
5 CTCS/16
6 CTIMES/16
7 TIMINC/16
8 LDTIME/16
9 LDAND/16
10 MPTIME/16
11 TRTACA/16
12 CSLVIM/16
13 CSLVAN/16
14 NANPTS/16
15 GOTRTA/16
16 EXTRTA/16
17 INTEMP/15
18 WRRDS/15
19 CTI2/16
20 LTPT/16
21 LOCPT/12
22 NEXTKB/16
23 TPROP/16
24 RDPQ/16
25 WPROP/16
26 JTCOMP/15
27 ASMKCR/16
28 AMTR/16
29 NADFS/15
30 ADDKAN/16
31 ASMEKS/16
32 JTPACK/16
33 TEK21/16
34 TEK31/16
35 TEK41/16
36 TEK61/16
37 TEK81/16
38 TEMT21/15
39 TEMT42/15
40 TEMT62/16
41 SAVESM/16
42 INVCAP/16
43 DTCOMP/16
44 DTK3/15
45 FACKAN/16
46 TTLV/16
47 CSCQ/16
48 CCDT/16
49 FACKIM/16
50 TEMPAN/16
51 MODKAN/16
52 SAVET/16
53 TTDS/16
54 MTPROD/14
55 RTDT/16
56 TTDSAN/16
57 TTDSIM/16
58 MODKIM/16
59 TFLUX/16
60 CMPTS/16
61 FTCOEF/16
62 LDTAF/16
63 TMTPD/16
64 TRFLUX/14

1.3 Processor Tables of Contents

Absolute Elements

NAME	VERSION	TYPE	DATE	TIME	SEQ #
TGEO		ABS	21 SEP 79	18:23: 5	1
SSTA		ABS	4 OCT 79	13:43:42	2
TRTA		ABS	13 OCT 79	16:38:42	3

Comdecks

NAME	VERSION	TYPE	DATE	TIME	SEQ #
LINES	12	FOR	5 MAY 77	16:19:36	1
CFPNI	12	FOR	21 SEP 77	8:41: 3	2
LINES		REL	23 MAY 77	14:20:14	3
CFPNI		REL	21 SEP 77	8:45:28	4
MOVER	13	FOR	30 JAN 78	12:35:43	5
MOVER		REL	31 JAN 78	11:25:41	6
CHKFPS	13	FOR	24 APR 78	12: 2:20	7
CHKFPS		REL	24 APR 78	12: 2:23	8
JFIX	13	FOR	24 APR 78	12: 2:28	9
JFIX		REL	24 APR 78	12: 2:30	10
CPROP	13	FOR	12 MAY 78	15: 2: 7	11
CPROP		REL	12 MAY 78	15: 2:23	12
MPROD1	13	FOR	16 JUN 78	13:21: 0	13
MPROD1		REL	16 JUN 78	13:21: 2	14
MPROD2	13	FOR	16 JUN 78	13:21: 5	15
MPROD2		REL	16 JUN 78	13:21: 6	16
MPROD3	13	FOR	16 JUN 78	13:21: 7	17
MPROD3		REL	16 JUN 78	13:21:10	18
SCPROD	13	FOR	16 JUN 78	13:21:15	19
SCPROD		REL	16 JUN 78	13:21:19	20
VSUM1	13	FOR	16 JUN 78	13:21:24	21
VSUM1		REL	16 JUN 78	13:21:25	22
VSUM2	13	FOR	16 JUN 78	13:21:42	23
VSUM2		REL	16 JUN 78	13:21:43	24
TACMPS	13	ELT	19 JUN 78	11:26:31	25
PDNM	14	FOR	17 AUG 78	9:58:25	26
PDNM		REL	17 AUG 78	9:58:54	27
MPTCHK	14	FOR	5 OCT 78	13:47: 1	28
MPTCHK		REL	5 OCT 78	13:47: 2	29
LOCT	14	FOR	3 NOV 78	15:22:19	30
LOCT		REL	3 NOV 78	15:22:39	31
EXPTIN	14	FOR	22 DEC 78	15:17: 2	32
EXPTIN		REL	22 DEC 78	15:17: 4	33
EXPTEK	14	FOR	22 DEC 78	15:17: 6	34
EXPTEK		REL	22 DEC 78	15:17: 8	35
EXPPD	14	FOR	22 DEC 78	15:17:18	36
EXPPD		REL	22 DEC 78	15:17:20	37
LEXP	14	FOR	22 DEC 78	16:26:37	38
LEXP		REL	22 DEC 78	16:26:38	39

EEPROP	14	FOR	22	DEC 78	16:26:38	40
EEPROP		REL	22	DEC 78	16:26:40	41
FEF62	14	FOR	4	JAN 79	10:51:44	42
FEF62		REL	4	JAN 79	10:51:46	43
ALPHA	14	FOR	5	JAN 79	11:37:52	44
ALPHA		REL	5	JAN 79	11:37:53	45
EPD	14	FOR	8	JAN 79	11:25: 2	46
EPD		REL	8	JAN 79	11:25: 4	47
REXEX	14	FOR	25	JAN 79	17:13:28	48
REXEX		REL	25	JAN 79	17:14: 0	49
CHKRED	14	FOR	7	FEB 79	16:42:39	50
CHKRED		REL	7	FEB 79	16:42:41	51
LDJS	13	FOR	18	JAN 78	18:33:12	52
LDJS		REL	19	JAN 78	17:23:31	53
CLC	12	FOR	4	FEB 77	11:16:18	54
CLC		REL	23	MAY 77	14:21:30	55
WRKB	14	FOR	7	MAR 79	8:32:59	56
WRKB		REL	7	MAR 79	8:33: 0	57
EC21	13	FOR	18	JAN 78	18:33:52	58
EC21		REL	19	JAN 78	17:26:27	59
EC31	12	FOR	31	MAR 77	16:30:18	60
EC31		REL	23	MAY 77	14:22:58	61
EC41	12	FOR	5	APR 77	10:54: 1	62
EC41		REL	23	MAY 77	14:23: 2	63
EC32	13	FOR	12	MAY 78	15:21:25	64
EC32		REL	12	MAY 78	15:21:27	65
EC42	13	FOR	12	MAY 78	15:21:29	66
EC42		REL	12	MAY 78	15:21:31	67
EC62	13	FOR	1	JUN 78	11: 4:16	68
EC62		REL	1	JUN 78	11: 4:21	69
ER21	13	FOR	18	JAN 78	18:33:55	70
ER21		REL	19	JAN 78	17:26:32	71
ER31	12	FOR	25	APR 77	11:13:15	72
ER31		REL	23	MAY 77	14:23: 7	73
ER41	12	FOR	25	APR 77	11:15:30	74
ER41		REL	23	MAY 77	14:23:10	75
ERK	13	FOR	24	JAN 78	9:42:21	76
ERK		REL	26	JAN 78	17:50:28	77
ADDK	14	FOR	7	MAR 79	8:32:55	78
ADDK		REL	7	MAR 79	8:32:58	79
FACIJ	13	FOR	12	MAY 78	15:21:58	80
FACIJ		REL	12	MAY 78	15:22: 1	81
CSPDP	12	FOR	18	FEB 77	16:51: 1	82
CSPDP		REL	23	MAY 77	14:27: 9	83
CDPSP	12	FOR	18	FEB 77	16:53: 0	84
CDPSP		REL	23	MAY 77	14:27:30	85
CHKIDS	15	FOR	25	MAY 79	9:48:58	86
CHKIDS		REL	25	MAY 79	9:49: 0	87
ELTEMP	15	FOR	25	MAY 79	9:49:13	88
ELTEMP		REL	25	MAY 79	9:49:16	89
DERIV	15	FOR	25	MAY 79	9:49:18	90
DERIV		REL	25	MAY 79	9:49:24	91
EXPRK2	15	FOR	25	MAY 79	9:49:31	92
EXPRK2		REL	25	MAY 79	9:49:33	93
FACTOR	15	FOR	25	MAY 79	9:49:34	94
FACTOR		REL	25	MAY 79	9:49:38	95

FACMD	15	FOR	25	MAY 79	9:49:39	96
FACMD		REL	25	MAY 79	9:49:45	97
EFLUX	15	FOR	25	MAY 79	9:49:58	98
EFLUX		REL	25	MAY 79	9:50:11	99
RFLUX	15	FOR	25	MAY 79	9:50:12	100
RFLUX		REL	25	MAY 79	9:50:19	101
CEDOF	15	FOR	25	MAY 79	10:38:35	102
CEDOF		REL	25	MAY 79	10:38:38	103
CSLV	15	FOR	25	MAY 79	10:38:46	104
CSLV		REL	25	MAY 79	10:38:50	105
EXPRK1	15	FOR	25	MAY 79	10:39: 3	106
EXPRK1		REL	25	MAY 79	10:39: 7	107
XSOL	15	FOR	25	MAY 79	10:39:25	108
XSOL		REL	25	MAY 79	10:39:29	109
CFMDM1	15	FOR	30	MAY 79	9:13: 3	110
CFMDM1		REL	30	MAY 79	9:13: 4	111
CFMDM2	15	FOR	30	MAY 79	9:13: 8	112
CFMDM2		REL	30	MAY 79	9:13: 9	113
NUEC	15	FOR	4	JUN 79	16:10:50	114
NUEC		REL	4	JUN 79	16:10:52	115
ADDVEC	13	FOR	13	FEB 78	7:30:13	116
ADDVEC		REL	13	FEB 78	7:37:22	117
QK21	13	FOR	31	JAN 78	17:49:45	118
QK21		REL	31	JAN 78	17:54:53	119
QK31	13	FOR	31	JAN 78	17:49:46	120
QK31		REL	31	JAN 78	17:54:56	121
QK41	13	FOR	31	JAN 78	17:49:50	122
QK41		REL	31	JAN 78	17:55: 0	123
QK61	12	FOR	13	OCT 77	10:27:22	124
QK61		REL	30	JAN 78	18:22:17	125
QK81	12	FOR	28	APR 77	13:19:18	126
QK81		REL	30	JAN 78	18:22:34	127
CFFC	16	FOR	4	JUL 79	17:13: 1	128
CFFC		REL	4	JUL 79	17:13: 5	129
CFFF	16	FOR	4	JUL 79	17:13: 7	130
CFFF		REL	4	JUL 79	17:13: 9	131
FPROP	16	FOR	4	JUL 79	17:13:10	132
FPROP		REL	4	JUL 79	17:13:12	133
LDKL	16	FOR	4	JUL 79	17:13:13	134
LDKL		REL	4	JUL 79	17:13:17	135
LDMP	16	FOR	4	JUL 79	17:13:19	136
LDMP		REL	4	JUL 79	17:13:23	137
MPROP	16	FOR	4	JUL 79	17:13:24	138
MPROP		REL	4	JUL 79	17:13:26	139
EXPFLX	16	FOR	4	JUL 79	18: 5:40	140
EXPFLX		REL	4	JUL 79	18: 5:41	141
EXPSCQ	16	FOR	4	JUL 79	18: 5:42	142
EXPSCQ		REL	4	JUL 79	18: 5:42	143
REVEC	16	FOR	4	JUL 79	18: 5:52	144
REVEC		REL	4	JUL 79	18: 5:54	145
LDIDJS	16	FOR	12	SEP 79	10:16:26	146
LDIDJS		REL	12	SEP 79	10:16:29	147
OPN1	16	FOR	21	SEP 79	17:41:35	148
OPN1		REL	21	SEP 79	17:41:37	149

TGEO

NAME	VERSION	TYPE	DATE	TIME	SEQ #
EGR3	12	FOR	16 MAY 77	15:41:34	1
EGR6	12	FOR	16 MAY 77	15:43:28	2
EGR8	12	FOR	20 MAY 77	16:22: 3	3
GEFACE	12	FOR	29 APR 77	17:10: 4	4
EGR3		REL	23 MAY 77	14:29:21	5
EGR6		REL	23 MAY 77	14:29:29	6
EGR8		REL	23 MAY 77	14:29:33	7
GEFACE		REL	19 SEP 77	13:14:52	8
GE2D	13	FOR	18 JAN 78	18:20:51	9
GE3D	13	FOR	18 JAN 78	18:20:53	10
GELDG	13	FOR	18 JAN 78	18:20:54	11
GESMRY	13	FOR	18 JAN 78	18:20:56	12
GE2D		REL	19 JAN 78	17:23:16	13
GE3D		REL	19 JAN 78	17:23:45	14
GELDG		REL	19 JAN 78	17:23:59	15
GESMRY		REL	19 JAN 78	17:24:43	16
TGEOID	13	FOR	26 JAN 78	17:12:59	17
TGEOID		REL	26 JAN 78	17:52: 7	18
EGREXP	13	FOR	16 FEB 78	12:26: 7	19
EGREXP		REL	16 FEB 78	12:31:17	20
TGEOCS	13	FOR	24 APR 78	15: 3:43	21
TGEOCS		REL	24 APR 78	15: 3:46	22
EGR4	13	FOR	1 JUN 78	10:57:12	23
EGR4		REL	1 JUN 78	10:57:21	24
READZL	14	FOR	9 AUG 78	13:50:31	25
READZL		REL	9 AUG 78	13:50:35	26
GOTGEO	14	FOR	9 AUG 78	13:50:49	27
GOTGEO		REL	9 AUG 78	13:50:57	28
READKL	14	FOR	22 DEC 78	14:57:20	29
READKL		REL	22 DEC 78	14:57:22	30
ELNMAT	14	FOR	22 DEC 78	14:57:29	31
ELNMAT		REL	22 DEC 78	14:57:31	32
EGR2	14	FOR	22 DEC 78	14:57:32	33
EGR2		REL	22 DEC 78	14:57:33	34
FORMKL	15	FOR	4 JUN 79	16:19:27	35
FORMKL		REL	4 JUN 79	16:19:30	36
EXTGEO	16	FOR	4 JUL 79	17:13:27	37
EXTGEO		REL	4 JUL 79	17:13:30	38
LDTGEO	16	FOR	12 SEP 79	10:17:33	39
LDTGEO		REL	12 SEP 79	10:17:38	40
MPTGEO	16	FOR	21 SEP 79	17:41:10	41
MPTGEO		REL	21 SEP 79	17:41:11	42
XXTGEO	UNI	FOR	21 SEP 79	17:41:19	43
XXTGEO		REL	21 SEP 79	17:41:20	44
TGEO		MAP	21 SEP 79	18:22:54	45

SSTA

NAME	VERSION	TYPE	DATE	TIME	SEQ #
EK61	12	FOR	13 OCT 77	9: 9:50	1
EK61		REL	13 OCT 77	10:25:12	2
EK21	13	FOR	18 JAN 78	18:33:42	3
EK31	13	FOR	18 JAN 78	18:33:45	4
EK41	13	FOR	18 JAN 78	18:33:49	5
EK21		REL	19 JAN 78	17:26: 4	6
EK31		REL	19 JAN 78	17:26:11	7
EK41		REL	19 JAN 78	17:26:21	8
ADDKLD	13	FOR	26 JAN 78	16:32:57	9
ADDCLD	13	FOR	26 JAN 78	16:32:59	10
ADDKLD		REL	26 JAN 78	17:51:46	11
ADDCLD		REL	26 JAN 78	17:51:50	12
CLSVEC	13	FOR	2 FEB 78	17:45:56	13
CLSVEC		REL	2 FEB 78	17:49: 2	14
GOSSTA	13	FOR	7 APR 78	8:12:30	15
GOSSTA		REL	7 APR 78	9: 4:21	16
LATL	14	FOR	22 DEC 78	15:19:19	17
LATL		REL	22 DEC 78	15:19:21	18
EMT21	14	FOR	4 JAN 79	10:52:27	19
EMT21		REL	4 JAN 79	10:52:29	20
EMT42	14	FOR	4 JAN 79	10:52:30	21
EMT42		REL	4 JAN 79	10:52:31	22
NLATL	14	FOR	7 FEB 79	16:43:10	23
NLATL		REL	7 FEB 79	16:43:12	24
SMTPD	14	FOR	9 FEB 79	12:22:22	25
SMTPD		REL	9 FEB 79	12:22:26	26
SRFLUX	14	FOR	9 FEB 79	12:22:28	27
SRFLUX		REL	9 FEB 79	12:22:30	28
STSSTA	15	FOR	25 MAY 79	10:58:27	29
STSSTA		REL	25 MAY 79	10:58:37	30
LCCR	15	FOR	25 MAY 79	10:58:57	31
LCCR		REL	25 MAY 79	10:59:10	32
SSTACA	15	FOR	25 MAY 79	10:59:13	33
SSTACA		REL	25 MAY 79	10:59:23	34
ASMBL	15	FOR	25 MAY 79	11: 0:14	35
ASMBL		REL	25 MAY 79	11: 0:31	36
EK81	15	FOR	25 MAY 79	11: 0:33	37
EK81		REL	25 MAY 79	11: 1: 2	38
EMT62	15	FOR	25 MAY 79	11: 1: 3	39
EMT62		REL	25 MAY 79	11: 1:22	40
ADDRLD	15	FOR	25 MAY 79	11: 1:24	41
ADDRLD		REL	25 MAY 79	11: 1:41	42
SDKM	15	FOR	25 MAY 79	11: 1:49	43
SDKM		REL	25 MAY 79	11: 1:56	44
TSOL	15	FOR	25 MAY 79	11: 2:32	45
TSOL		REL	25 MAY 79	11: 2:37	46
CSTVEC	15	FOR	25 MAY 79	11: 2:39	47
CSTVEC		REL	25 MAY 79	11: 2:45	48
ASMFAC	15	FOR	25 MAY 79	12:57:23	49
ASMFAC		REL	25 MAY 79	12:57:26	50

LDKSD	15	FOR	25	MAY 79	16:56:17	51
LDKSD		REL	25	MAY 79	16:56:21	52
ATKM	15	FOR	25	MAY 79	17: 0:57	53
ATKM		REL	25	MAY 79	17: 1: 2	54
PRBTYP	16	FOR	4	JUL 79	17:13:31	55
PRBTYP		REL	4	JUL 79	17:13:33	56
NLTLV	16	FOR	4	JUL 79	18: 5:48	57
NLTLV		REL	4	JUL 79	18: 5:51	58
LTLV	16	FOR	4	JUL 79	18: 5:55	59
LTLV		REL	4	JUL 79	18: 5:58	60
SFLUX	16	FOR	4	JUL 79	18: 5:59	61
SFLUX		REL	4	JUL 79	18: 6: 2	62
EXSSTA	16	FOR	6	JUL 79	12:24:45	63
EXSSTA		REL	6	JUL 79	12:24:49	64
XXSSTA	UNI	FOR	21	SEP 79	17:50:36	65
XXSSTA		REL	21	SEP 79	17:50:37	66
MPSSTA	16	FOR	21	SEP 79	17:50:46	67
MPSSTA		REL	21	SEP 79	17:50:47	68
LDSSTA	16	FOR	4	OCT 79	12:51:45	69
LDSSTA		REL	4	OCT 79	12:51:48	70
SSTA		MAP	4	OCT 79	13:43: 9	71

TRTA

NAME	VERSION	TYPE	DATE	TIME	SEQ #
LOCPT	12	FOR	31 AUG 77	12:53:43	1
LOCPT		REL	31 AUG 77	12:55:23	2
MTPROD	14	FOR	22 DEC 78	17:31:59	3
MTPROD		REL	22 DEC 78	17:32: 0	4
TRFLUX	14	FOR	9 FEB 79	12:32: 2	5
TRFLUX		REL	9 FEB 79	12:32: 4	6
WRRDS	15	FOR	28 MAY 79	9:30:46	7
WRRDS		REL	28 MAY 79	9:30:48	8
TEMT21	15	FOR	28 MAY 79	10:39:18	9
TEMT21		REL	28 MAY 79	10:39:20	10
JTCOMP	15	FOR	28 MAY 79	11:38:30	11
JTCOMP		REL	28 MAY 79	11:38:31	12
DTK3	15	FOR	28 MAY 79	12:16:29	13
DTK3		REL	28 MAY 79	12:16:31	14
NADFS	15	FOR	29 MAY 79	17: 7:41	15
NADFS		REL	29 MAY 79	17: 7:41	16
INTEMP	15	FOR	5 JUN 79	13:58:35	17
INTEMP		REL	5 JUN 79	13:58:36	18
TEMT42	15	FOR	11 JUN 79	16:44: 9	19
TEMT42		REL	11 JUN 79	16:44:10	20
WPROP	16	FOR	4 JUL 79	17:12:56	21
WPROP		REL	4 JUL 79	17:12:57	22
RDPQ	16	FOR	4 JUL 79	17:12:58	23
RDPQ		REL	4 JUL 79	17:12:59	24
LDTIME	16	FOR	4 JUL 79	17:13:34	25
LDTIME		REL	4 JUL 79	17:13:39	26
TEK21	16	FOR	4 JUL 79	17:13:40	27
TEK21		REL	4 JUL 79	17:13:42	28
TEK61	16	FOR	4 JUL 79	17:13:48	29
TEK61		REL	4 JUL 79	17:13:50	30
CMPTS	16	FOR	4 JUL 79	17:53:54	31
CMPTS		REL	4 JUL 79	17:53:55	32
LDTAF	16	FOR	4 JUL 79	17:53:56	33
LDTAF		REL	4 JUL 79	17:53:58	34
TMTPD	16	FOR	4 JUL 79	17:54: 0	35
TMTPD		REL	4 JUL 79	17:54: 3	36
TEK31	16	FOR	4 JUL 79	17:54: 7	37
TEK31		REL	4 JUL 79	17:54: 8	38
FTCOEF	16	FOR	4 JUL 79	17:54: 9	39
FTCOEF		REL	4 JUL 79	17:54:11	40
TEK81	16	FOR	4 JUL 79	17:54:17	41
TEK81		REL	4 JUL 79	17:54:19	42
TEK41	16	FOR	4 JUL 79	18: 5:28	43
TEK41		REL	4 JUL 79	18: 5:30	44
CSLVAN	16	FOR	4 JUL 79	18: 6: 3	45
CSLVAN		REL	4 JUL 79	18: 6: 5	46
JTPACK	16	FOR	4 JUL 79	18: 6:13	47
JTPACK		REL	4 JUL 79	18: 6:14	48
TFLUX	16	FOR	4 JUL 79	18: 6:21	49
TFLUX		REL	4 JUL 79	18: 6:25	50

RTDT	16	FOR	6 JUL 79	12:26:33	51
RTDT		REL	6 JUL 79	12:26:35	52
NANPTS	16	FOR	6 JUL 79	18:51:31	53
NANPTS		REL	6 JUL 79	18:51:34	54
ADDKAN	16	FOR	6 JUL 79	18:51:50	55
ADDKAN		REL	6 JUL 79	18:51:51	56
LDAND	16	FOR	12 SEP 79	10:17:22	57
LDAND		REL	12 SEP 79	10:17:26	58
TTDS	16	FOR	12 SEP 79	11:14:52	59
TTDS		REL	12 SEP 79	11:14:58	60
FACKAN	16	FOR	16 SEP 79	12: 9:21	61
FACKAN		REL	16 SEP 79	12: 9:24	62
CTI2	16	FOR	16 SEP 79	12:30:10	63
CTI2		REL	16 SEP 79	12:30:11	64
GOTRTA	16	FOR	16 SEP 79	12:53:35	65
GOTRTA		REL	16 SEP 79	12:53:38	66
ASMEKS	16	FOR	17 SEP 79	12: 2:57	67
ASMEKS		REL	17 SEP 79	12: 3: 0	68
AMTR	16	FOR	17 SEP 79	17:29:41	69
AMTR		REL	17 SEP 79	17:29:43	70
TIMINC	16	FOR	17 SEP 79	17:29:49	71
TIMINC		REL	17 SEP 79	17:29:51	72
MPTIME	16	FOR	18 SEP 79	10: 6:26	73
MPTIME		REL	18 SEP 79	10: 6:28	74
TPROP	16	FOR	18 SEP 79	12:40:52	75
TPROP		REL	18 SEP 79	12:40:54	76
LTPT	16	FOR	18 SEP 79	15: 7:20	77
LTPT		REL	18 SEP 79	15: 7:21	78
SAVET	16	FOR	19 SEP 79	15: 8:54	79
SAVET		REL	19 SEP 79	15: 8:55	80
NEXTKB	16	FOR	19 SEP 79	15: 8:59	81
NEXTKB		REL	19 SEP 79	15: 9: 0	82
CSLVIM	16	FOR	19 SEP 79	17:42:40	83
CSLVIM		REL	19 SEP 79	17:42:41	84
ASMKCR	16	FOR	20 SEP 79	8: 2:47	85
ASMKCR		REL	20 SEP 79	8: 2:50	86
TTDSIM	16	FOR	20 SEP 79	8: 3: 7	87
TTDSIM		REL	20 SEP 79	8: 3: 9	88
SAVESM	16	FOR	20 SEP 79	8: 3:26	89
SAVESM		REL	20 SEP 79	8: 3:28	90
TTLV	16	FOR	20 SEP 79	13:16:43	91
TTLV		REL	20 SEP 79	13:16:46	92
CCDT	16	FOR	20 SEP 79	13:22: 0	93
CCDT		REL	20 SEP 79	13:22: 2	94
TTDSAN	16	FOR	20 SEP 79	15: 5:59	95
TTDSAN		REL	20 SEP 79	15: 6: 7	96
FACKIM	16	FOR	20 SEP 79	15: 6:10	97
FACKIM		REL	20 SEP 79	15: 6:13	98
MODKAN	16	FOR	20 SEP 79	15: 6:18	99
MODKAN		REL	20 SEP 79	15: 6:21	100

MODKIM	16	FOR	20 SEP 79	15: 6:23	101
MODKIM		REL	20 SEP 79	15: 6:26	102
CTCS	16	FOR	21 SEP 79	11:50:12	103
CTCS		REL	21 SEP 79	11:50:15	104
STTRTA	16	FOR	21 SEP 79	11:50:18	105
STTRTA		REL	21 SEP 79	11:50:21	106
TEMPAN	16	FOR	21 SEP 79	17:38: 2	107
TEMPAN		REL	21 SEP 79	17:38: 6	108
XXTRTA	UNI	FOR	21 SEP 79	17:50:42	109
XXTRTA		REL	21 SEP 79	17:50:43	110
MPTRTA	16	FOR	21 SEP 79	17:50:52	111
MPTRTA		REL	21 SEP 79	17:50:53	112
INVCAP	16	FOR	26 SEP 79	13:33:57	113
INVCAP		REL	26 SEP 79	13:34:12	114
EXTRTA	16	FOR	28 SEP 79	14:42:55	115
EXTRTA		REL	28 SEP 79	14:43:11	116
LDTRTA	16	FOR	4 OCT 79	13:30:17	117
LDTRTA		REL	4 OCT 79	13:30:19	118
IETIM	16	FOR	4 OCT 79	13:35:49	119
IETIM		REL	4 OCT 79	13:35:50	120
CSCQ	16	FOR	4 OCT 79	15:37:38	121
CSCQ		REL	4 OCT 79	15:37:40	122
TEMT62	16	FOR	8 OCT 79	11:26:21	123
TEMT62		REL	8 OCT 79	11:26:26	124
DTCOMP	16	FOR	9 OCT 79	11:17:57	125
DTCOMP		REL	9 OCT 79	11:18:12	126
TRTACA	16	FOR	9 OCT 79	11:18:19	127
TRTACA		REL	9 OCT 79	11:18:24	128
CTIMES	16	FOR	13 OCT 79	15:33:13	129
CTIMES		REL	13 OCT 79	15:33:14	130
TRTA		MAP	13 OCT 79	16:38:20	131

Section 2 - FACTORING AND SOLUTION PACKAGE

The factoring and solution package utilized in processors SSTA and TRTA has been designed as a stand alone collection of subroutines.

Labeled Common Blocks: FACPAR
KDATA1
KDATA3
FAC001
FAC002

Subroutines: FACTOR
FACIJ
FACMD
SCPROD
XSOL

Double precision capability is obtained by changing all statement numbers 6600 to comments C6600, and all comments C1108 to statement numbers 1108 in subroutines FACIJ, FACMD, SCPROD, and XSOL.

Labeled Common Blocks:

FACPAR/ NNR,NBR,NPRNT,NBRS,SING,ZERO

- NNR = number of negative diagonal terms detected during factoring.
- NBR = number of singularities detected during factoring.
- NPRNT = print control parameter used during factoring of K matrix.
  - = 0, no printing.
  - = 1, print singularity messages.
  - = 2, print negative root messages.
  - = 3, print singularity and negative root messages.
- NBRS = number of singularities allowed during factoring of the K matrix before error termination.
- SING = singularity parameter. If the absolute value of a factored diagonal term is less than or equal to SING times the absolute value of the diagonal term before factoring, the matrix is considered singular.
- ZERO = singularity parameter. If the absolute value of a factored diagonal term is less than or equal to ZERO, the matrix is considered singular.

KDATA1/ NUK(2),NZK(2),LRK,NRK

NUK(1) = assembled K matrix unit.
(2) = factored K matrix unit.
NZK(1) = sector address of first block of assembled matrix.
(2) = sector address of first block of factored matrix.
(Blocks are written sequentially.)
LRK = K matrix block length.
NRK = number of K matrix blocks.

KDATA3/ KSYM,KDIM

KSYM = 1, symmetric K matrix. Each K matrix block contains only lower triangular terms.
= 2, asymmetric K matrix. Each K matrix block contains both lower and upper triangular terms.
KDIM = number of words in each of the lower and upper triangular parts in the K matrix. The total number of words in each K matrix block (LRK in KDATA1) is given by:
LRK = KDIM*KSYM if single precision.
= 2*KDIM*KSYM if double precision.

FAC001/ I1,I2,J1,J2,KB,KM

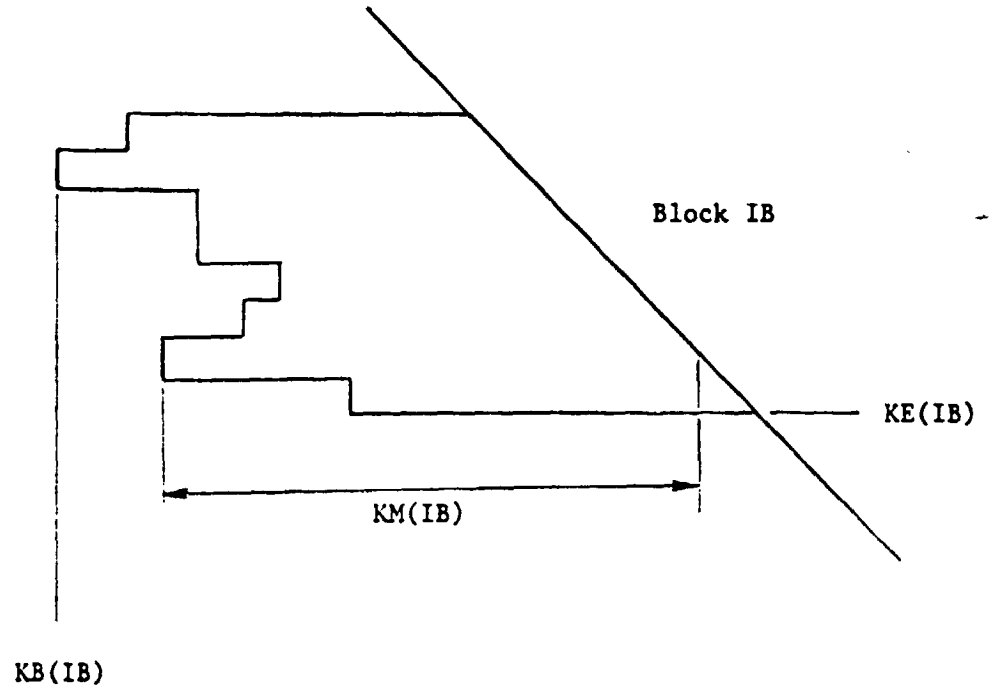
I1 = first equation in block IB.
I2 = last equation in block IB.
J1 = first equation in block JB.
J2 = last equation in block JB.
KB = lowest equation number referenced in block IB.
KM = maximum off-diagonal bandwidth in block IB.

FAC002/ J,KJ1,KJ2,LJ

J = equation number in block JB.
KJ1 = location within block JB of first term of equation J.
KJ2 = location within block JB of last term of equation J.
LJ = KJ2 - KJ1.

Subroutine FACTOR(LC,KE,KB,KM,AI,AJ)
Calls: FACIJ
Function: Controls the decomposition of the K matrix into upper and lower triangular parts.
Singular equations are removed from the system of equations by zeroing out the affected rows and columns and setting the inverse of the diagonal term equal to zero.

LC(I) = location within block of diagonal term of equation I; I=1, ..., number of degrees of freedom. If a factored diagonal term is negative, the corresponding LC term is negative.
KE(IB) = last equation in block IB; IB=1, ..., number of blocks.
KB(IB) = lowest equation referenced in block IB.
KM(IB) = maximum off-diagonal bandwidth in block IB.
AI, AJ = K matrix blocks.



```
Subroutine FACMD(LC,AI,AJ,KDIM,KSYM)
Called by: FACIJ
Calls:     SCPROD
Function:  Factors diagonal terms of the K matrix.
           Checks for singularities and negative diagonal terms.
LC     = skyline matrix, see FACTOR.
AI, AJ = K matrix blocks.
KDIM   = see common block KDATA3.
KSYM   = see common block KDATA3.
```

```
Subroutine FACIJ(LC,AI,AJ,KDIM,KSYM)
Called by: FACTOR
Calls:     FACMD, SCPROD
Function:  Performs factoring operations of all off-diagonal terms.
LC     = skyline vector, see FACTOR.
AI, AJ = K matrix blocks.
KDIM   = see common block KDATA3.
KSYM   = see common block KDATA3.
```

```
Subroutine SCPROD(N,INCA,INCB,A,B,SUM)
Called by: FACIJ, FACMD, XSOL
Function:  Computes the scalar product
           SUM = A(1)*B(1) + A(1+INCA)*B(1+INCB) + ...
N = length of vectors A and B.
```

```
Subroutine XSOL(LC,KE,X,LX,NX,NEQ,A,KDIM,KSYM,NBC)
Calls:     SCPROD
Function:  Solves KX = F using factored K matrix.
LC       = skyline vector, see FACTOR.
KE       = last equation in block vector, see FACTOR.
X(LX,NX) = load vector before solution.
         = solution vector after solution.
LX       = adjusted length (LADJ) of X vector.
NX       = number of solutions.
NEQ      = number of equations.
A        = K matrix block.
KDIM     = see common block KDATA3.
KSYM     = see common block KDATA3.
NBC      = current block in core.
```

Section 3 - LABELED COMMON BLOCK DESCRIPTIONS

Labeled Common Blocks Shared by TGEO, SSTA, and TRTA:

CLINES - see LINES.

EDATA1/ NUE,NLS,LBSIZ
- NUE = element library.
- NLS = number of element types.
- LBSIZ = element data block size.

JDATA1/ NJTS,KAID,KAJS
- NJTS = number of joints (nodes) in finite element model.
- KAID = sector address of TDOF ID.
- KAJS = sector address of JSEQ BTAB.

Labeled Common Blocks of TGEO:

CTGEO1/ KAKL,KAZL,NLZ,NNZ
- KAKL = address in central memory of KL array (see READKL, FORMKL).
- KAZL = address in central memory of LZD vector (see READZL).
- NLZ = length of LZD vector.
- NNZ = number of ZEROL execution commands.

CTGEO2/ NFNP,LFNP,NCMAT
- NFNP = 0, do not restrain unconnected nodes.
  = 1, restrain unconnected nodes.
- LFNP = number of unconnected nodes to list.
- NCMAT = 0, each experimental element must reference an Eij PROP data set.
  = 1, experimental elements may reference Eij COEF data sets or have no material properties defined.

CTGE03/ GEOMP(10)

GEOMP(1) = zero length test parameter. Element boundary lengths must be greater than GEOMP(1).

GEOMP(2) = sine of the minimum allowable angle between adjacent element edges.

GEOMP(3) = threshold warping parameter for four node elements.

GEOMP(4) = excessive warping parameter for four node elements.

GEOMP(5) = maximum allowable aspect ratio - defined as the ratio of element edge length to characteristic length. The characteristic length is defined as SQRT(AREA) for a four node element and SQRT(2.*AREA) for a three node element.

GEOMP(6) = not used.

GEOMP(7) = R(t)R must equal the identity matrix within a tolerance of GEOMP(7), where R is the direction cosine matrix of an element local coordinate frame with respect to the global coordinate system.

GEOMP(8) = minimum height ratio for hexahedral elements. The height ratio is defined as H/L, where H is the distance from the element base formed by the "plane" of nodes 1-4 to each of the nodes 5-8 and L is the square root of the area of the element base.

GEOMP(9) = threshold warping parameter for pentahedral and hexahedral element faces.

GEOMP(10) = excessive warping parameter for pentahedral and hexahedral element faces.

GEO1/ LNAME,NG,NELT,NPTS,LERR

LNAME = element name (4HK21 , etc.).

NG = element group number.

NELT = index number of element within group.

NPTS = number of element node points.

LERR = 0, no element error flag.
= 1, element warning flag.
= 2, element error flag.

GEO2/ TESTGE(10),IERTST(10),NTOTGE(10),ITGE

TESTGE = test parameters (see labeled common CTGE03).

IERTST(I) = 1, warning if element fails test I.
= 2, error if element fails test I.

NTOTGE(I) = number of elements which failed test I.

ITGE = element error status.
= 0, element passed all tests.
= J, element failed test J.

GEO3/ IPRTGE,LIMGE,NPRTGE

IPRTGE = 0, do not print.
= 1, print element error messages.
= 2, print element warning and error messages.

LIMGE = maximum number of messages to be printed for element type.

NPRTGE = number of messages printed for element type.

Labeled Common Blocks Shared by SSTA and TRTA:

ADDK01 - see WRKB

ADDK02 - see WRKB

EDATA2/ MXNDF,MXNPTS,MXKDF
- MXNDF = maximum number of freedoms at any node.
- MXNPTS = maximum number of nodes in any element.
- MXKDF = maximum number of freedoms in any element.

EDATA3/ KLRE,KLMT,LRMT
- KLRE = 0, no radiation elements.
  = 1, radiation elements present.
- KLMT = 0, no mass-transport elements.
  = 1, mass-transport elements present.
- LRMT = maximum of mass-transport elements of any type.

EDATA4/ LNAME,LTYPE,NMATS,NPTS,NISECT,NG,NELT,NMAT
- LNAME = element name.
- LTYPE = KL(5,I) in LDKL.
- NMATS = KL(6,I) in LDKL.
- NPTS = KL(4,I) in LDKL.
- NISECT = KL(3,I) in LDKL.
- NG = KL(2,I) in LDKL.
- NELT = element index number.
- NMAT = element material number.

FAC001 - see Section 2.

FAC002 - see Section 2.

FACPAR - see Section 2.

FDATA1/ NFLUX,NPRFLX,LRFLUX,NRFLUX
- NFLUX = 0, do not compute element fluxes.
  = 1, compute element fluxes from an existing SSTA TEMP or TRTA TEMP data set.
  = 2, compute fluxes after solution is completed.
- NPRFLX = 0, no online print of flux quantities.
  = 1, print flux quantities online (see EFLUX).
- LRFLUX = maximum record length of any SFLX Eij or TFLX Eij data set.
- NRFLUX = number of flux records (corresponding to load cases or times) that can be processed in central memory at one time.

FFCOM1 - see FEF62.

KDATA1 - see Section 2.1.

Labeled Common Blocks Shared by SSTA and TRTA:

ADDK01 - see **UNKB**

ADDK02 - see **UNKB**

EDATA2/ MKDOF,MXNPTS,MXKDF
MXDOF = maximum number of freedoms at any node.
MXNPTS = maximum number of nodes in any element.
MXKDF = maximum number of freedoms in any element.

EDATA3/ KLRE,KLMT,LRMT
KLRE = 0, no radiation elements.
= 1, radiation elements present.
KLMT = 0, no mass-transport elements.
= 1, mass-transport elements present.
LRMT = maximum of mass-transport elements of any type.

EDATA4/ LNAME,LTYPE,NMATS,NPTS,NISECT,NG,NELT,NMAT
LNAME = element name.
LTYPE = KL(5,I) in LDKL.
NMATS = KL(6,I) in LDKL.
NPTS = KL(4,I) in LDKL.
NISECT = KL(3,I) in LDKL.
NG = KL(2,I) in LDKL.
NELT = element index number.
NMAT = element material number.

FAC001 - see Section 2.

FAC002 - see Section 2.

FACPAR - see Section 2.

FDATA1/ NFLUX,NPRFLX,LRFLUX,NRFLUX
NFLUX = 0, do not compute element fluxes.
= 1, compute element fluxes from an existing SSTA TEMP or TRTA TEMP data set.
= 2, compute fluxes after solution is completed.
NPRFLX = 0, no online print of flux quantities.
= 1, print flux quantities online (see EFLUX).
LRFLUX = maximum record length of any SFLX Eij or TFLX Eij data set.
NRFLUX = number of flux records (corresponding to load cases or times) that can be processed in central memory at one time.

FFCOM1 - see FEF62.

KDATA1 - see Section 2.1.

```
KDATA2/ KNAME(2),NBRMS,MXBW,KTYP
        KNAME(1) = first name in assembled K matrix data set.
        KNAME(2) = first name in factored K matrix data set.
        NBRMS    = rms off-diagonal bandwidth.
        MXBW     = maximum off-diagonal bandwidth.
        KTYP     = 0, assemble and factor K matrix.
                 = 4HK   , start analysis with assembled matrix.
                 = 4HKFAC, start analysis with factored matrix.

KDATA3 - see Section 2.1.

KBICOM - see DERIV.

MDATA1 - see LDMP.

MDATA2 - see LDMP.

MTCOM1/ GNEWT,ZEROL,ZEROA
        GNEWT = acceleration of gravity.
        ZEROL = zero length parameter.
        ZEROA = zero area parameter.

MTCOM2 - see ALPHA.

RADCOM/ SBCON
        SBCON = Stefan-Boltzmann constant.

RECOM1/ NUR(2),NREX,LREX,LREXG,NRELS,NWWR
        NUR(1) = REX library.
           (2) = radiation scratch library (REX GEOM, central memory).
        NREX   = number of REX data sets.
        LREX   = maximum length of all REX data sets.
        LREXG  = length of REX GEOM.
        NRELS  = number of radiation elements.
        NWWR   = number of words to store from central memory to make room for
                 radiation exchange computations (TRTA only).

RECOM2/ REI,RCONV,NRITS,NRDIST,NRCC,NRCCS,RCCMX
        REI    = radiation interval size.
        RCONV  = radiation convergence parameter.
        NRITS  = maximum number of steps.
        NRDIST = 0, do not distribute residual reflected heat.
               = 1, distribute residual reflected heat.
        NRCC   = nonconvergence action indicator:
               = 0, fin if no convergence.
               = 1, continue if no convergence.
               = 2, continue and write a message if no convergence.
               = 3, continue, write a message, and save radiation-conduction
                 vector if no convergence.
        NRCCS  = number of nonconverged radiation vectors.
        RCCMX  = maximum ratio of total reflected heat to total emitted heat.
```

[illegible]/ [illegible](2),[illegible],[illegible],KTYP
[illegible](1) = first name is assembled K matrix data set.
[illegible](2) = first name is factored K matrix data set.
[illegible] = rms off-diagonal bandwidth.
[illegible] = maximum off-diagonal bandwidth.
KTYP = 0, assemble and factor K matrix.
= [illegible], start analysis with assembled matrix.
= [illegible], start analysis with factored matrix.

[illegible] - see Section 2.1.

[illegible] - see [illegible].

[illegible] - see LDMP.

[illegible] - see LDMP.

MTCOM1/ GRAVT,ZEROL,ZEROA
GRAVT = acceleration of gravity.
ZEROL = zero length parameter.
ZEROA = zero area parameter.

[illegible] - see ALPHA.

[illegible]/ SBCON
SBCON = Stefan-Boltzmann constant.

[illegible]/ NUR(2),[illegible],LREX,LREXG,NRELS,[illegible]
NUR(1) = REX library.
(2) = radiation scratch library (REX GEOM, central memory).
NREX = number of REX data sets.
LREX = maximum length of all REX data sets.
LREXG = length of REX GEOM.
NRELS = number of radiation elements.
[illegible] = number of words to store from central memory to make room for radiation exchange computations (TRTA only).

RECOM2/ REI,RCONV,NRITS,NRDIST,NRCC,NRCCS,RCOMX
REI = radiation interval size.
RCONV = radiation convergence parameter.
NRITS = maximum number of steps.
NRDIST = 0, do not distribute residual reflected heat.
= 1, distribute residual reflected heat.
NRCC = nonconvergence action indicator:
= 0, fin if no convergence.
= 1, continue if no convergence.
= 2, continue and write a message if no convergence.
= 3, continue, write a message, and save radiation-conduction vector if no convergence.
NRCCS = number of nonconverged radiation vectors.
RCOMX = maximum ratio of total reflected heat to total emitted heat.

```
RECOM3/ NURFLX,NRF(6)
        NURFLX = unit for writing radiation exchange fluxes.
        NRF(I) = 0, do not compute fluxes.
               = 1, compute fluxes.
                 I = 1 - emissive power.
                   = 2 - irradiation.
                   = 3 - heat reflected.
                   = 4 - radiosity.
                   = 5 - heat absorbed.
                   = 6 - heat emitted.

PDATA1/ NPRESS,NPRPRS,LRPRES,NRPRES
        NPRESS = 0, do not compute element pressures.
               = 1, compute element pressures from an existing SSTA TEMP or
                    TRTA TEMP data set.
               = 2, compute pressures after solution is completed.
        NPRPRS = 0, no online printout of pressure quantities.
               = 1, print pressure quantities online (see EPD).
        LRPRES = maximum record length of any SPD Eij or TPD Eij data set.
        NRPRES = number of pressure records (corresponding to load cases or
                 times) that can be processed in central memory at one time.

PDATA2/ PZERO,NPITS,PCONV
        PZERO = zero pressure parameter.
        NPITS = number of iterations allowed during gas pressure computations.
        PCONV = gas pressure convergence parameter.

QDATA1/ NUQ,LRQ,NSOURL,NCONVL,NRADIL
        NUQ    = thermal excitation.
        LRQ    = maximum record length of all thermal excitation data sets.
        NSOURL = 0, no source loading.
               = 1, source loading.
        NCONVL = 0, no convection loading.
               = 1, convection loading.
        NRADIL = 0, no radiation loading.
               = 1, radiation loading.

QDATA2/ NSET,L1,L2,LC1,LC2,NLCS
        NSET     = load set identifier.
        L1, L2   = load cases L1 through L2 are processed.
        LC1, LC2 = load cases LC1 through LC2 are processed simultaneously for
                   linear problems.
        NLCS     = maximum number of load cases that can be processed simul-
                   taneously.

QDATA3/ NTEMPL,NATKA,AKFAC
        NTEMPL = number of prescribed nodal temperatures.
        AKFAC  = multiplication factor for applied temperatures.
```

[illegible]/ [illegible]

NRADFLX = unit for writing radiation exchange fluxes.

NRF(I) = 0, do not compute fluxes.
= 1, compute fluxes.
I = 1 - emissive power.
= 2 - irradiation.
= 3 - heat reflected.
= 4 - radiosity.
= 5 - heat absorbed.
= 6 - heat emitted.

PDATA1/ NPRESS,NPRPRS,LRPRES,NRPRES

NPRESS = 0, do not compute element pressures.
= 1, compute element pressures from an existing SSTA TEMP or TRTA TEMP data set.
= 2, compute pressures after solution is completed.

NPRPRS = 0, no online printout of pressure quantities.
= 1, print pressure quantities online (see EPD).

LRPRES = maximum record length of any SPD Eij or TPD Eij data set.

NRPRES = number of pressure records (corresponding to load cases or times) that can be processed in central memory at one time.

PDATA2/ PZERO,NPITS,PCONV

PZERO = zero pressure parameter.

NPITS = number of iterations allowed during gas pressure computations.

PCONV = gas pressure convergence parameter.

QDATA1/ NSQ,LRQ,NSOURL,NCONVL,NRADIL

NSQ = thermal excitation.

LRQ = maximum record length of all thermal excitation data sets.

NSOURL = 0, no source loading.
= 1, source loading.

NCONVL = 0, no convection loading.
= 1, convection loading.

NRADIL = 0, no radiation loading.
= 1, radiation loading.

QDATA2/ NSET,L1,L2,LC1,LC2,NLCS

NSET = load set identifier.

L1, L2 = load cases L1 through L2 are processed.

LC1, LC2 = load cases LC1 through LC2 are processed simultaneously for linear problems.

NLCS = maximum number of load cases that can be processed simultaneously.

QDATA3/ NTEMPL,NATKA,AKFAC

NTEMPL = number of prescribed nodal temperatures.

AKFAC = multiplication factor for applied temperatures.

SOPCOM/ NDP
NDP = -1, single precision assembly, factoring, and solution.
= -2, double precision assembly, factoring, and solution.

TDATA1/ NUT,NZT,LRT,NRT
NUT = nodal temperatures library unit.
NZT = sector address of nodal temperature vector.
LRT = length of nodal temperature vector.
NRT = number of nodal temperature vectors.

TDATA2/ ITEMP,TEMP1,IERRT
ITEMP = 0, initial temperatures read from existing STAT TEMP or TRTA TEMP data sets.
= 1, initial temperature at all node points is TEMP1.
TEMP1 = initial temperature.
IERRT = error indicator in STAT TEMP or TRTA TEMP data set.

TMPCOM - see LOCT.

TFPCOM/ LASTM,LASTF,NFLUID,FTEMP
LASTM = last material property temperature point referenced.
LASTF = last fluid or gas temperature point referenced.
NFLUID = fluid number if positive and gas number if negative.
FTEMP = fluid or gas temperature.

Labeled Common Blocks of SSTA:

CSSTA1/ N1,N2,N3,N4
N1 = address in central memory of KL (see LDKL).
N2 = address in central memory of MP (see LDMP).
N3 = address in central memory of PROP (see LDMP).
N4 = address in central memory of working data space.

NLCOM1 - see PRBTYP.

NLCOM2/ NITER,NFACS,CONVRG,CUTOFF,NEGTMP
see SSTA RESET controls NITERATIONS, NFACS, CONVERGE, CUTOFF, and NEGTEMP.

SKYVEC - see Section 2.

SSPCOM/ MSP
MSP = -1, single precision assembly, factoring, and solution.
= -2, double precision assembly, factoring, and solution.

TDATA1/ NUT,NZT,LNT,MNT
NUT = nodal temperatures library unit.
NZT = sector address of nodal temperature vector.
LNT = length of nodal temperature vector.
MNT = number of nodal temperature vectors.

TDATA2/ ITEMP,TEMP1,IERMT
ITEMP = 0, initial temperatures read from existing STAT TEMP or TRTA TEMP data sets.
= 1, initial temperature at all node points is TEMP1.
TEMP1 = initial temperature.
IERMT = error indicator in STAT TEMP or TRTA TEMP data set.

TMPCOM - see LOCI.

TFPCOM/ LASTM,LASTF,NFLUID,FTEMP
LASTM = last material property temperature point referenced.
LASTF = last fluid or gas temperature point referenced.
NFLUID = fluid number if positive and gas number if negative.
FTEMP = fluid or gas temperature.

Labeled Common Blocks of SSTA:

CSSTA1/ M1,M2,M3,M4
M1 = address in central memory of KL (see LDKL).
M2 = address in central memory of MP (see LDMP).
M3 = address in central memory of PROP (see LDMP).
M4 = address in central memory of working data space.

NLCOM1 - see PRETYP.

NLCOM2/ NITER,NFACS,CONVRG,CUTOFF,NEGTMP
see SSTA RESET controls NITERATIONS, NFACS, CONVERGE, CUTOFF, and NEGTEMP.

SKYVEC - see Section 2.

Labeled Common Blocks of TRTA:

```
ANCOM1/NUAN,NANPS,NRANS
       NUAN  = ARIT NODE data set unit.
       NANPS = number of arithmetic nodes.
       NRANS = number of arithmetic nodes connected to radiation elements.


CLOCPT - see LOCPT.

CTIME1/ DTI,DTS,MXNDT,NDTI,DT,T1,T2,TI1,TI2
        DTI   = time interval size.
        DTS   = save results at steps of DTS.
        MXNDT = maximum number of integration time steps.
        NDTI  = number of integration time steps per interval.
        DT    = integration time step size.
        T1    = analysis starting time.
        T2    = analysis ending time.
        TI1   = interval starting time.
        TI2   = interval ending time.

CTIME2/ NTERMS,CT(10),CDT(10)
        NTERMS = number of terms in Taylor series.
        CT     = Taylor series coefficients: 1, 1, 1/2, 1/3!, 1/4!, etc.
        CDT(1) = CT(1)
        CDT(2) = CT(2)*DT
        CDT(3) = CT(3)*DT**2

CTIME3/ IDT,NPREP,FDT,NPRDT,DTFAC,DTIP
        IDT   = see RESET control IDT.
        NPREP = see RESET control PREP.
        FDT   = see RESET control FDT.
        NPRDT = see RESET control PRINT.
        DTFAC = parameter used in the recomputation of DT.
        DTIP  = input integration time step size (RESET control DT).

CTIME4 - see TIMINC.

CTIME4/ CTIME,QTIME,QT1(4),QT2(4)
        CTIME  = current time.
        QTIME  = next time to compute load vector.
        QT1(I) = last time at which load vector I was computed.
        QT2(I) = next time to compute load vector I.
                 I = 1, source.
                     2, convection.
                     3, radiation.
                     4, applied temperatures.

CTIME6/ IETR,BETA
         IETR = 0, explicit time integration method.
                1, implicit time integration method after assembly.
                2, implicit time integration method after factoring.
```

Labeled Common Blocks of TRTA:

ARCOM1/MUAN,NARPS,NRARS
MUAN = ARIT NODE data set unit.
NARPS = number of arithmetic nodes.
NRARS = number of arithmetic nodes connected to radiation elements.

CLOCPT - see LOCPT.

CTIME1/ DTI,DTS,MXNDT,NDTI,DT,T1,T2,TI1,TI2
DTI = time interval size.
DTS = save results at steps of DTS.
MXNDT = maximum number of integration time steps.
NDTI = number of integration time steps per interval.
DT = integration time step size.
T1 = analysis starting time.
T2 = analysis ending time.
TI1 = interval starting time.
TI2 = interval ending time.

CTIME2/ NTERMS,CT(10),CDT(10)
NTERMS = number of terms in Taylor series.
CT = Taylor series coefficients: 1, 1, 1/2, 1/3!, 1/4!, etc.
CDT(1) = CT(1)
CDT(2) = CT(2)*DT
CDT(3) = CT(3)*DT**2

CTIME3/ IDT,NPREP,FDT,NPRDT,DTFAC,DTIP
IDT = see RESET control IDT.
NPREP = see RESET control PREP.
FDT = see RESET control FDT.
NPRDT = see RESET control PRINT.
DTFAC = parameter used in the recomputation of DT.
DTIP = input integration time step size (RESET control DT).

CTIME4 - see TIMINC.

CTIME4/ CTIME,QTIME,QT1(4),QT2(4)
CTIME = current time.
QTIME = next time to compute load vector.
QT1(I) = last time at which load vector I was computed.
QT2(I) = next time to compute load vector I.
I = 1, source.
2, convection.
3, radiation.
4, applied temperatures.

CTIME6/ IETR,BETA
IETR = 0, explicit time integration method.
1, implicit time integration method after assembly.
2, implicit time integration method after factoring.

CTIME7/ NUST,IDST(4),NZTIME
NUST = unit of save time vector (execution command TSAVE).
IDST = 4 word name of save time data set.
NZTIME = address of TRTA TIME data set.

CTIME8/ NUKT,IDKT(4),NTIPS,TIFAC
NUKT = unit of time interval vector (execution command TIME).
IDKT = 4 word name of time interval data set.
NTIPS = number of time intervals.
TIFAC = factor used in computation of next time interval
(see subroutine CTI2).

CTRAD1/RKM,NRAERR,NRAPRT,NITER,CONVRG,ANFAC,NFACS,NFAC
RKM = radiation term correction factor in MODKAN and MODKIM.
NRAERR = see RESET controls ANCC and IMCC.
ANRPRT = see RESET controls ANRPRT and IMPRT.
NITER = see RESET control NITER.
CONVRG = see RESET controls ANCONV and CONV.
ANFAC = see RESET controls ANFAC and CUTOFF.
NFACS = see RESET control NFACS.
NFAC = number of factorings during iterations in TTDSAN and TTDSIM.

CTRTA1/ N1,N2,N3,N4,N5,N6,N7,N8,N9
N1 = address in central memory of KL (see LDKL).
N2 = address in central memory of MP (see LDMP).
N3 = address in central memory of PROP (see LDMP).
N4 = address in central memory of interval times (see CTIMES).
N5 = address in central memory of save times (see CTIMES).
N6 = address in central memory of T (see TIMINC).
N7 = address in central memory of ATNP (see LDTIME).
N8 = address in central memory of NPAN (see TRTACA).
N9 = address in central memory of working data space.

CTRTA2 - central memory address computed in TRTACA, see TRTA central memory allocation.

EKCOM1 - see NEXTKB.

EKCOM2 - see NEXTKB.

EKCOM4 - see INVCAP.

MDATA3/ MPTI,MPRT
MPTI = 0, time-average properties over each time interval.
1, evaluate properties at beginning of each time interval.
MPRT = 0, do not print properties.
1, print properties each time they are evaluated.

TDATA3/ NUIT,ITID(5)
NUIT = initial temperature library unit.
ITID = 4 word data set name and record number.

Section 4 - SUBROUTINES COMMON TO MORE THAN ONE PROCESSOR

Subroutine ADDK(LC,KE,A,LM,EK,NDF,KP)
Called by: ASMBL, ADDKAN
Calls: WRKB
Function: Assembles system K matrix from element K matrices.
LC = skyline vector.
KE = last equation in block vector.
A = K matrix buffer (1 block).
LM = element degree of freedom vector.
EK = element K matrix.
NDF = number of element degrees of freedom.
KP(I) = K matrix block to which element K matrix terms in row I are added.
KP(I)=0, after all terms in row I have been added.

Subroutine ADDVEC(A,TA,EA,LM,KDF)
Called by: NLTLV, LTLV, ASMKCR, CSCQ
Function: Adds element contributions to A vector.
A = vector to which element contributions are added.
TA = magnitude of element contributions (source strength, etc.).
EA = element vector.
LM = element degree of freedom vector.
KDF = number of element degrees of freedom.

Subroutine ALPHA(CM,CK,A)
Called by: EMT21, EMT42, EMT62, TEMT21, TEMT42, TEMT62
Function: Computes mass-transport alpha parameter.
Common: MTCOM2/ MTK,ALPH
MTK = 0, set alpha parameter=ALPH.
= 1, compute alpha parameter.
ALPH = see RESET control ALPHA in SSTA and TRTA.
CM = specific heat times mass-transport rate.
CK = k*A/L.
A = alpha parameter.

Subroutine CDPSP(DP,SP,N)
Called by: TSOL, TEMPAN, TTDSAN, TTDSIM
Function: Converts double precision vector to single precision.
DP = double precision vector.
SP = single precision vector.
N = length of DP and SP vectors.

Subroutine CEDOF(JS,NP,LM,NDF,NPTS)
Called by: LDKSD, ASMBL, NLTLV, LTLV, ASMKCR, CSCQ
Function: Computes element degrees of freedom.
JS = joint elimination sequence vector.
NP = element node point vector.
LM = element degree of freedom vector.
NDF = number of element degrees of freedom.
NPTS = number of element node points.

Function CFFC(MP,PROP)
Called by: EFLUX, EMT62, EC32, EC42, EC62, TEMT62, TEC32, TEC42, TEC62
Calls: LOCT
Function: Computes fluid convection coefficient.
MP = see LDMP.
PROP = see LDMP.

Function CFFF(MP,PROP)
Called by: EFD
Calls: LOCT
Function: Computes fluid friction factor.
MP = see LDMP.
PROP = see LDMP.

Subroutine CFMDM1(AF,AD,KDF)
Called by: ASMKCR
Function: Converts a square matrix to a full diagonal matrix.
AF = square matrix.
AD = diagonal matrix.
KDF = dimension of AF and AD.

Subroutine CFMDM2(AF,AD,KDF)
Called by: LTLV, ASMKCR, CSCQ
Function: Converts a square matrix to a diagonal matrix (vector format).
AF = square matrix.
AD = diagonal matrix.
KDF = dimension of AF.

Subroutine CFPNI(A,IA,N)
Function: Converts a floating point string to an integer string.
A = floating point string.
IA = integer string.
N = length of A and IA.

Subroutine CHKFPS(A,N)
Calls: JFIX
Function: Processes floating point string so that integers stored as floating point numbers can be correctly converted to integers by function IFIX.
A = data string.
N = number of words in data string.

Subroutine CHKIDS(IA,N,IA1,IA2)
Function: Checks a string of integers to insure that no two are duplicated and that all are bracketed by IA1 and IA2.
IA = integer string.
N = dimension of IA.
IA1 = lowest allowable integer in IA.
IA2 = highest allowable integer in IA.

Subroutine CHKRED(KL,ED,A,KORE)
Called by: STSSTA, STTRTA
Function: Checks REX data sets for errors and constructs REX GEOM data set.
KL = see LDKL.
ED = element data buffer.
A = working data space.
KORE = data space available.

Subroutine CLC(LC,NP,NPTS)
Called by: LDKSD, CSLVAN
Function: Computes LC.
LC(I) = minimum of existing LC(I) or lowest numbered node point in element.
NP = element node point array.
NPTS = number of element node points.

Subroutine CPROP(P,T)
Called by: EXPTEK, EXPSCQ, EXPFLX, EXPPD
Calls: LOCT
Function: Controls computation of temperature-dependent experimental element material properties.
P = PROP(J), see EEPROP.
T = temperature at which properties are to be evaluated.

Subroutine CSLV(LC,NEQ)
Called by: LDKSD, CSLVAN
Function: Computes skyline vector.
Computes rms and maximum off-diagonal bandwidths.
LC = skyline vector (see Section 2).
NEQ = number of equations.

Subroutine CSPDP(SP,DP,N)
Called by: TSOL, TEMPAN, TTDSAN, TTDSIM
Function: Converts single precision vector to double precision.
SP = single precision vector.
DP = double precision vector.
N = length of SP and DP vectors.

Function DERIV(X,NG)
Called by: EK81, TEK81
Function: Computes shape functions and derivatives at quadrature point NG.
Common: K81COM/ H(8),DHX(8,3)
H(I) = shape function.
DHX(I,J) = JACOBIAN*DH(I)/DX(J).
X = element coordinates.
NG = quadrature point.

Subroutine EC21(CIR,XL,EK)
Called by: ASMBL, NLTLV, LTLV, ASMKCR, CSCQ
Function: Computes convection matrix for C21 elements.
CIR = element circumference.
XL = element length.
EK = element convection matrix.

Subroutine EC31(X,EK)
Called by: ASMBL, NLTLV, LTLV, ASMKCR, CSCQ
Function: Computes convection matrix for C31 elements.
X = element coordinates.
EK = element convection matrix.

Subroutine EC32(AF,XL,EK,ET,MP,PROP)
Called by: ASMBL, NLTLV, ASMKCR, CSCQ
Calls: FPROP, CFFC
Function: Computes convective-exchange matrix for C32 elements.
AF = width of convective-exchange surface.
XL = length of convective-exchange surface.
EK = element convective-exchange matrix.
ET = element temperature vector.
MP = see LDMP.
PROP = see LDMP.

Subroutine EC41(X,EK)
Called by: ASMBL, NLTLV, LTLV, ASMKCR
Function: Computes convection matrix for C41 elements.
X = element coordinates.
EK = element convection matrix.

Subroutine EC42(AF,XL,EK,ET,MP,PROP)
Called by: ASMBL, NLTLV, ASMKCR
Calls: FPROP, CFFC
Function: Computes convective-exchange matrix for C42 elements.
AF = width of convective-exchange surface.
XL = length of convective-exchange surface.
EK = element convective-exchange matrix.
ET = element temperature vector.
MP = see LDMP.
PROP = see LDMP.

Subroutine EC62(SECT,X,EK,ET,MP,PROP)
Called by: ASMBL, NLTLV, EMT62, ASMKCR, TEMT62
Calls: FPROP, CFFC, FEF62
Function: Computes convective-exchange matrix for C62 elements.
SECT = thickness of fin, fin area factor.
X = fin coordinates X(2,4).
EK = element convective-exchange matrix.
ET = element temperature vector.
MP = see LDMP.
PROP = see LDMP.

Subroutine EEPROP(LOC,MP,PROP,J)
Called by: ASMBL, NLTLV, LTLV, SFLUX, SMTPD, ASMKCR, CSCQ, TFLUX, TMTPD
Calls: LOCT
Function: Computes experimental element material properties at the average element temperature.
Computes the location of the appropriate material properties in the the PROP array so that temperature-dependent properties can be referenced by subroutines EXPTEK, EXPSCQ, EXPFLX, and EXPPD.
LOC = KL(19,N) and KL(20,NL), see LDKL.
MP = see LDMP.
PROP = see LDMP.
J = properties locations in PROP array.

Subroutine EFLUX(MP,PROP,SECT,X,ET,TA,FLUX)
Called by: SFLUX, TFLUX
Calls: MPROP, FPROP, CFFC
Function: Computes element fluxes.
MP = see LDMP.
PROP = see LDMP.
SECT = element section properites vector.
X = element coordinates.
ET = element temperature vector.
TA = element exchange temperature or mass-transport rate.
FLUX = element fluxes.

Function ELTEMP(T,NP,ET,NDF,NPTS)
Called by: REVEC, ASMBL, NLTLV, SFLUX, SMTPD, ASMKCR, CSCQ, TFLUX, TMTPD
Function: Computes element nodal temperatures.
T = nodal temperature vector.
NP = element node point vector.
ET = element nodal temperatures.
NDF = number of freedoms at each element node point.
NPTS = number of element nodes.

Subroutine EPD(MP,PROP,SECT,X,ET,TR,P,KERR)
Called by: SMTPD, TMTPD
Calls: FPROP, CFFF, PDNM
Function: Computes element fluid pressures.
MP = see LDMP.
PROP = see LDMP.
SECT = element section properties vector.
X = element coordinates.
ET = element temperature vector.
TR = element mass-transport rate.
P = element fluid pressures.
KERR = error indicator for gas pressures (see PDNM).

Subroutine ERK(EK,ET,KDF)
Called by: ASMBL
Function: Converts element radiation matrices as output by ER21, ER31, and ER41 to a form ready for assembly.
EK = element radiation matrix.
ET = element temperature vector.
KDF = number of element degrees of freedom.

Subroutine ER21(CIR,XL,EK)
Called by: ASMBL, NLTLV, ASMKCR
Function: Computes radiation matrix for R21 elements.
CIR = element circumference.
XL = element length.
EK = element radiation matrix.

Subroutine ER31(X,EK)
Called by: ASMBL, NLTLV, ASMKCR
Function: Computes radiation matrix for R31 elements.
X = element coordinates.
EK = element radiation matrix.

Subroutine ER41(X,EK)
Called by: ASMBL, NLTLV, ASMKCR
Function: Computes radiation matrix for R41 elements.
X = element coordinates.
EK = element radiation matrix.

Subroutine EXPFLX - this subroutine is described in the SPAR Thermal Analysis Reference Manual - Vol. 4, Experimental Element Capability

Subroutine EXPPD - this subroutine is described in the SPAR Thermal Analysis Reference Manual - Vol. 4, Experimental Element Capability

Subroutine EXPRK1(EK,ET,N)
Called by: ASMKCR
Function: Converts diagonal radiation matrix output by EXPTEK and CFMDM2 to a form useable by TRTA.
EK = diagonal element radiation matrix.
ET = element temperature vector.
N = number of element freedoms.

Subroutine EXPRK2(EK,KDF)
Called by: ASMBL
Function: Multiplies experimental element radiation matrix by 4.
EK = element radiation matrix.
KDF = number of element degrees of freedom.

Subroutine EXPSCQ - this subroutine is described in the SPAR Thermal Analysis Reference Manual - Vol. 4, Experimental Element Capability

Subroutine EXPTEK - this subroutine is described in the SPAR Thermal Analysis Reference Manual - Vol. 4, Experimental Element Capability

Subroutine EXPTIN - this subroutine is described in the SPAR Thermal Analysis Reference Manual - Vol. 4, Experimental Element Capability

Subroutine FACIJ - see Section 2.

Subroutine FACMD - see Section 2.

Subroutine FACTOR - see Section 2.

```
Function FEF62(TH,X,MP,PROP)
Called by: EFLUX, EMT62, EC62, TEMT62, TEC62
Calls:     MPROP
Function:  Computes fin efficiency.
Common:    FFCOM1/ NFIN,ZFIN
                   NFIN = 0, fin efficiency is 1.
                          1, compute fin efficiency.
                   ZFIN = zero fin t*h measure.
TH   = fin thickness.
X    = element coordinates.
MP   = see LDMP.
PROP = see LDMP.
```

```
Subroutine FPROP(MP,PROP)
Called by: EFLUX, EPD, EMT21, EC32, EC42, TEMT21, TEC32, TEC42
Calls:     LOCT
Function:  Controls computation of fluid properties.
MP   = see LDKL.
PROP = see LDKL.
```

```
Function JFIX(A)
Function:  Converts an integer input as a floating point number via AUS/TABLE
           to its correct integer value.
A    = original floating point number.
JFIX = integer value of A.
```

```
Subroutine LDIDJS(ID,KORE)
Called by: STSSTA, STTRTA
Function:  Reads degree of freedom and joint elimination sequence data.
ID(N) = number of active freedoms at node N.
KORE  = amount of data space available.
```

Subroutine LDJS(JS)
Called by: REVEC, LDKSD, EXSSTA, TRTACA
Function: Reads joint elimination sequence vector.
JS = joint elimination sequence vector.

Subroutine LDKL(KL,KSPACE)
Called by: STSSTA, STTRTA
Function: Fills KL array from data set TED GRPS.
KL(1,I) = element name (e.g. K21, C41, ...)
KL(2,I) = number of groups.
KL(3,I) = number of section properties.
KL(4,I) = number of node points.
KL(5,I) = element type: 1 - conduction.
2 - convection to a known temperature.
3 - radiation.
4 - mass-transport.
5 - convective exchange.
100 - experimental.
101 - experimental with radiation.
110 - experimental with mass-transport.
111 - experimental with radiation and mass-transport.
KL(6,I) = number of materials.
KL(7,I) = number of freedoms per node point.
KL(8,I) = number of flux quantities.
KL(9,I) = major (experimental elements).
KL(10,I) = minor (experimental elements).
KL(11,I) = number of pressure drop quantities.
KL(12,I) = loading indicator: -1, never any loading.
0, no loading for any groups.
1, loading for some groups.
2, all groups must have loading.
KL(13,I) = number of coordinate quantities.
KL(14,I) = number of elements.
KL(15,I) = not used.
KL(16,I) = not used.
KL(17,I) = MXMATS(1) for experimental elements (see LDMP).
KL(18,I) = MXMATS(2) for experimental elements (see LDMP).
KL(19,I) = LOCMAT(1) for experimental elements (see LDMP).
KL(20,I) = LOCMAT(2) for experimental elements (see LDMP).
KSPACE = data space available.

Subroutine LDMP(KL,MP,PROP,KSPACE,NRR)
Called by: STSSTA, STTRTA
Calls: MPTCHK
Function: Reads material and fluid properties and constructs the MP pointer array.
Common: MDATA1/ NUM,NMP,MBSIZ
NUM = material and fluid properties unit.
NMP = MP dimension - MP(3,NMP) for SSTA.
MP(3,2*NMP) for TRTA.
MBSIZ = PROP dimension.
MDATA2/ MXMAT(2,5),LOCMAT(2,5)
MXMAT(1,MTYPE) = highest temperature-dependent material number referenced.
MXMAT(2,MTYPE) = highest non-temperature-dependent material number referenced.
LOCMAT(1,MTYPE) = location in MP of temperature-dependent properties (NMAT>0) pointer for material type MTYPE.
LOCMAT(2,MTYPE) = location in MP of non-temperature-dependent properties (NMAT<0) for material type MTYPE.
where MTYPE = 1 (conduction).
= 2 (convection).
= 3 (radiation).
= 4 (fluid).
= 5 (gas).

KL = see LDKL.
MP = material or fluid properties pointer (see below).
PROP = properties buffer.
KSPACE = data space available for PROP array.
NRR = analysis type indicator: 1 - steady-state, 2 - transient.

Temperature-dependent properties (NMAT>0):

MLOC = LOCMAT(1,MTYPE).
J = MP(1,MLOC+NMAT), location in PROP of first temperature-dependent property line (from XXXX PROP nmat).
NIP = MP(2,MLOC+NMAT), length of property line.
NJP = MP(3,MLOC+NMAT), number of property lines.
NBLKS = MP(1,MLOC+NMAT+NMP), number of property blocks.
KA = MP(2,MLOC+NMAT+NMP), mass storage sector address of first block of properties.
LASTMP = MP(3,MLOC+NMAT+NMP), last block read from mass storage.

Non-temperature-dependent properties (NMAT<0):

MLOC = LOCMAT(2,MTYPE)
J = MP(1,MLOC)-NIP*(NMAT+1), location in PROP of first property line (from XXXX COEF).
NIP = MP(2,MLOC), length of property line.
NJP = MP(3,MLOC), number of property lines.
NBLKS = MP(1,MLOC+NMP), number of property blocks.
KA = MP(2,MLOC+NMP), mass storage sector address of first block of properties.
LASTMP = MP(3,MLOC+NMP), last block read from mass storage.

Function LEXP(LTYPE.MATCH)
Called by: LDKL, PRBTYP, LCCR, ASMBL, NLTLV, SFLUX, ASMKCR, TFLUX
Function: Determines if an experimental element is a radiation or mass-transport type element.
LTYPE = element type identification number (see LDKL).
MATCH = experimental element identification number to which LTYPE is compared (101 for radiation, 110 for mass-transport).
LEXP = 0, element is not type indicated by MATCH.
= 1, element is type indicated by MATCH.

Function LINES(LAST)
Function: Computes the number of lines that can be transferred from mass storage to central memory.
Common: CLINES/ KA,LR,KAC,NI,NJ,LRC
Input: NI = number of words per line.
NJ = number of lines in mass storage block.
LRC = data space available in central memory.
LAST = last line processed.
Output: KA = relative mass storage sector address.
LR = number of words to read from mass storage.
KAC = relative address in central memory of first data line.
LINES = number of complete data lines that can be read into central memory.

Function LOCT(P,N1)
Called by: MPROP, FPROP, CFFC, CFFF, EEPROP, CPROP
Function: Computes temperature-dependent material and fluid properties by linear interpolation.
Common: TMPCOM/ NIP,NJP,LAST,T,TMP(50)
Input: NIP = property line length.
NJP = number of property lines.
LAST = line number of last call to LOCT.
T = temperature at which to evaluate properties.
P = property buffer.
N1 = point in TMP where properties are written.
Output: TMP(N1) = interpolated properties.
LOCT = current line number.

Subroutine MOVER - see subroutine listing.

Subroutine MPROD1(N,A,B,C)
Called by: TTDS
Function: Computes A(I) = B(I)*C(I), I=1, ..., N.

Subroutine MPROD2(N,A,B,C)
Called by: TTDS
Function: Computes A(I) = -B(I)*C(I), I=1, ..., N.

Subroutine MPROD3(A,X,Y,IA,NV)
Called by: TTDS
Function: The product of the symmetric matrix A and vector X is subtracted from Y. The result is stored in Y.
A = the lower triangular part of a symmetric square matrix.
IA = A matrix degree of freedom vector.
NV = dimension of A and IA.

Subroutine MPROP(MP,PROP,MTYPE)
Called by: FEF62, REVEC, EFLUX, EPD, ASMBL, EMT42, EMT62, NLTLV, LTLV, ASMKCR, CSCQ, TEMT42, TEMT62
Calls: LOCT
Function: Controls computation of material properties.
MP = see LDMP.
PROP = see LDMP.
MTYPE = material type: 1-conduction, 2-convection, 3-radiation.

Subroutine MPTCHK(P,NIP,NJP,IERR)
Called by: LDMP
Function: Checks to insure that multi-line XXXX PROP data sets have sequentially increasing temperatures.
P = material properties buffer.
NIP = NI of XXXX PROP data set.
NJP = NJ of XXXX PROP data set.
IERR = error indicator: 0 - no error, 1 - error.

Function NUEC(NU,N1)
Called by: LDSSTA, LDTRTA
Function: Determines execution command library unit.
NU = default library unit.
N1 = next word to read in IDATA.
NUEC = library unit.

Subroutine PDNM(P,A,B,T1,T2,KERR)
Called by: EPD
Function: Computes gas pressures.
P = element gas pressure quantities.
A = see statement 650 in EPD.
B = see statement 650 in EPD.
T1 = inlet temperature.
T2 = outlet temperature.
KERR = 0, convergence achieved in computing gas pressure quantities.
= 1, no convergence during computation of gas pressure quantities.

Subroutine QK21(A,XL,QK)
Called by: NLTLV, LTLV, CSCQ
Function: Computes unit source heat vector for K21 elements.
A = element area.
XL = element length.
QK = element source heat vector.

Subroutine QK31(TH,X,QK)
Called by: NLTLV, LTLV, CSCQ
Function: Computes unit source heat vector for K31 elements.
TH = element thickness.
X = element coordinates.
QK = element source heat vector.

Subroutine QK41(TH,X,QK)
Called by: NLTLV, LTLV, CSCQ
Function: Computes unit source heat vector for K41 elements.
TH = element thickness.
X = element coordinates.
QK = element source heat vector.

Subroutine QK61(X,QK)
Called by: NLTLV, LTLV, CSCQ
Calls: QK81
Function: Computes unit source heat vector for K61 elements.
X = element coordinates.
QK = element source heat vector.

Subroutine QK81(X,QK)
Called by: NLTLV, LTLV, CSCQ
Function: Computes unit source heat vector for K81 elements.
X = element coordinates.
QK = element source heat vector.

Subroutine REVEC(KL,MP,PROP,T,A,IA,NERR)
Called by: EXSSTA, TTLV
Calls: ELTEMP, MPROP, REXEX
Function: Computes radiation exchange vector.
KL = see LDKL.
MP = see LDMP.
PROP = see LDMP.
T = nodal temperature vector.
A = working data space.
IA = working data space (integers).
NERR = 0, normal return.
1, error return.

Subroutine REXEX - see the subroutine listing.

```
Subroutine RFLUX(KL,MP,PROP,A,LA,RG,NP)
Called by: SRFLUX, TRFLUX
Calls:     ELTEMP, MPROP, REXEX
Function:  Computes radiation exchange fluxes.
KL     = see LDKL.
MP     = see LDMP.
PROP   = see LDMP.
A(N,1) = emissive power of element N.
A(N,2) = irradiation on element N.
A(N,3) = heat reflected by element N.
A(N,4) = radiosity of element N.
A(N,5) = heat absorbed by element N.
A(N,6) = heat emitted by element N.
LA     = number of radiation elements.
RG     = buffer for REX GEOM data set.
NP     = integer equivalent of RG.


Subroutine SCPROD - see Section 2.


Subroutine VSUM1 - see subroutine listing.


Subroutine VSUM2 - see subroutine listing.


Subroutine WRKB(A,IOWR)
Called by: ASMBL, ADDK, TRTACA, ASMKCR
Function:  Writes and reads K matrix blocks during assembly.
Common:    ADDK01/ NBC
                   NBC = current K block in central memory.
           ADDK02/ NCWR,IOW,IOR
                   NCWR = see RESET control KBWR.
                   IOW  = number of K block writes during assembly.
                   IOR  = number of K block reads during assembly.
A    = K matrix block.
IOWR = 0, initialize data, zero out and write each block.
     = 1, write block.
     = 2, read block.


Subroutine XSOL - see Section 2.
```

Section 5 - PROCESSOR TGEO

5.1 TGEO Central Memory Allocation

N1	KL	- element information array	- 20*NLS
N2	LZD	- zero-length element data string	- NLZ
N3	XG	- global coordinates	- 3*NJTS
N4	ID	- maximum number of freedoms vector	- NJTS
N5	ED	- element data buffer	- LBSIZ
N6	NP	- element nodes	- MXNPTS

The blank common locations N1, N2, etc. are computed in GOTGEO.

5.2 TGEO Subroutines

Subroutine LDTGEO
Called by: MPTGEO
Calls: RSET, READER, READKL, READZL, FORMKL
Function: Reads program control information. Initializes data.

Subroutine READKL(KL)
Called by: LDTGEO
Function: Fills portions of the KL array when elements are defined via AUS/TABLE and TGEO rather than ELD. (See Section 5.4).
KL = see LDKL.

Subroutine READZL(LZD)
Called by: LDTGEO
Function: Processes zero-length element data input via execution command ZEROL.
LZD = zero-length data string. The string contains NLZ words in NNZ blocks (common block CTGEO1).
Each block is structured according to:
(1) element name.
(2) element group number.
(3) next block address in LZD.
(4) number of elements.
(5) list of elements.

Subroutine FORMKL(KL,LD,KORE)
Called by: LDTGEO
Function: Fills KL array (see LDKL).
KL = see LDKL.
LD = ELD generated data sets buffer.
KORE = available LD working space.

Subroutine GOTGEO
Called by: MPTGEO
Calls: TGEOCS, EXTGEO
Function: Reads coordinates and controls data space allocation.

```
Subroutine TGEOCS(KL,KORE)
Called by: GOTGEO
Function:  Computes maximum number of freedoms at any node point, checks working
           data space.
KL   = see LDKL.
KORE = available working space.


Subroutine EXTGEO(KL,LZD,XG,ID,ED,NP)
Called by: GOTGEO
Calls:     LINES, CHKFPS, CFPNI, ELNMAT, EGR2, EGR3, EGR4, EGR6, EGR8, EGREXP,
           GESMRY, TGEOID
Function:  Controls element checkout and writes element coordinates into the
           TED Eij data sets.
KL    = see LDKL.
LZD   = see READZL.
XG    = nodal coordinates.
ID(N) = number of active freedoms at node N.
ED    = element data buffer:
          material numbers in ED(I1),
          node points in ED(I2),
          section properties in ED(I3),
          element coordinates in ED(I4).
NP    = element node point vector.


Subroutine ELNMAT(LTYPE,EDM,MXMAT,NBAD)
Called by: EXTGEO
Function:  Fills MXMAT array in MDATA3 common block.
           Checks for illegal material numbers.
LTYPE = element name.
EDM   = material numbers buffer.
MXMAT = see description of MDATA3 common block in LDMP.
NBAD  = material property error counter.


Subroutine EGREXP(XG,NP,XL,MAJOR)
Called by: EXTGEO
Function:  Constructs element coordinates and checks the geometry of experimen-
           tal elements according to the value of MAJOR as described in the SPAR
           Experimental Thermal Element Capability Reference Manual.
XG    = global coordinates (from JLOC BTAB).
NP    = element node point vector.
XL    = element coordinates.
MAJOR = element type parameter.
```

Subroutine EGR2(LZD,XG,NP,XL)
Called by: EXTGEO
Function: Computes the lengths of 2-node elements.
Sets the lengths of "zero-length" elements = 1.
LZD = zero-length data string, see READZL.
XG = global coordinates (from JLOC BTAB).
NP = element node point vector.
XL = element length.

Subroutine EGR3(XG,NP,XL)
Called by: EXTGEO
Calls: GE2D
Function: Controls the geometry checking of 3-node elements.
XG = global coordinates (from JLOC BTAB).
NP = element node point vector.
XL = element coordinates.

Subroutine EGR4(XG,NP,SECT,XL)
Called by: EXTGEO
Calls: GE2D
Function: Controls the geometry checking of 4-node elements.
XG = global coordinates (from JLOC BTAB).
NP = element node point vector.
SECT = element section properties (MT62 elements). The hydraulic diameter of an MT62 element is computed if not defined in data set MT62 SECT.
XL = element coordinates.

Subroutine EGR6(XG,NP,XL)
Called by: EXTGEO
Calls: GE3D
Function: Controls geometry checking of 6-node elements.
XG = global coordinates (from JLOC BTAB).
NP = element node point vector.
XL = element coordinates.

Subroutine EGR8(XG,NP,XL)
Called by: EXTGEO
Calls: GE3D
Function: Controls geometry checking of 8-node elements.
XG = global coordinates (from JLOC BTAB).
NP = element node point vector.
XL = element coordinates.

```
Subroutine GE2D(XG,NODES,NJOINT,XL,R,SAREA,A123,A124,X34)
Called by: EGR3, EGR4, GE3D
Calls:     GELDG, GEFACE
Function:  Checks geometry of 2-dimensional elements.
XG     = global coordinates.
NODES  = number of element node points.
NJOINT = element node point vector.
XL     = element coordinates.
R      = global-to-element coordinate transformation matrix.
         ( XL = R*DXG )
SAREA  = total surface area.
A123   = surface area defined by nodes 1, 2, 3.
A124   = surface area defined by nodes 1, 2, 4.
X34    = out-of-plane position of node 4.


Subroutine GE3D(XG,NJOINT,XL,VOL,R)
Called by: EGR6, EGR8
Calls:     GE2D, GELDG
Function:  Checks geometry of 3-dimensional elements.
XG     = global coordinates.
NJOINT = element node point vector.
XL     = element coordinates.
VOL    = element volume.
R      = global-to-element coordinate transformation matrix.
         ( XL = R*DXG )


Subroutine GEFACE(NODES,X,X34,JT,AREA)
Called by: GE2D
Calls:     GELDG
Function:  Checks geometry of 2-dimensional elements.
NODES = number of element node points.
X     = element coordinates.
X34   = out-of-plane position of node 4.
JT    = element node point vector.
AREA  = surface area.


Subroutine GELDG(KPRT,IE,JT,X,AV,N,NJ)
Called by: GE2D, GE3D, GEFACE
Function:  Sets element error flags and prints error messages.
KPRT = error message print indicator.
IE   = error test number (see labeled common CTGE03).
JT   = element node point vector.
X    = element coordinates.
AV   = surface area or volume.
N    = 2- or 3-dimension indicator.
NJ   = number of element node points.
```

Subroutine GESMRY
Called by: EXTGEO
Function: Prints element error summary.

Subroutine TGEOID(NP,ID)
Called by: EXTGEO
Function : Determines if there are any unconnected node points.
NP = list of unconnected node points.
ID(N) = number of active freedoms at node N.

5.3 Data Sets Generated by TGEO

5.3.1 The TED Eij ngrp Data Sets

Each of these data sets is structured as follows:
(1) material and fluid members (NMATS)
(2) node points (NPTS)
(3) section properties, where applicable (NISECT)
(4) coordinates (NICOOR)
(5) last line(s) referenced in material property table(s) (NMATS)
(6) ELD section properties pointer
(7) free pointer for future use
(8) arithmetic node indicator (set in TRTA subroutine NANPTS)

The data set contents (on exit from TGEO) for each element type are listed below:

```
TED K21  ngrp 0 (NI=9):  nmat,J1,J2,Ak,length,0,nsect,1,0
TED K31  ngrp 0 (NI=15): nmat,J1,J2,J3,Tk,X1,Y1,X2,Y2,X3,Y3,0,nsect,1,0
TED K41  ngrp 0 (NI=18): nmat,J1,J2,J3,J4,Tk,X1,Y1,X2,Y2,X3,Y3,X4,Y4,0,nsect,1,
                         0
TED K61  ngrp 0 (NI=29): nmat,J1,J2,J3,J4,J5,J6,X1,Y1,Z1,X2,Y2,Z2,X3,Y3,Z3,X4,
                         Y4,Z4,X5,Y5,Z5,X6,Y6,Z6,0,nsect,1,0
TED K81  ngrp 0 (NI=37): nmat,J1,J2,J3,J4,J5,J6,J7,J8,X1,Y1,Z1,X2,Y2,Z2,X3,Y3,
                         Z3,X4,Y4,Z4,X5,Y5,Z5,X6,Y6,Z6,X7,Y7,Z7,X8,Y8,Z8,0,
                         nsect,1,0
TED C21  ngrp 0 (NI=9):  nmat,J1,J2,C,length,0,nsect,1,0
TED C31  ngrp 0 (NI=15): nmat,J1,J2,J3,0,X1,Y1,X2,Y2,X3,Y3,0,nsect,1,0
TED C41  ngrp 0 (NI=18): nmat,J1,J2,J3,J4,0,X1,Y1,X2,Y2,X3,Y3,X4,Y4,0,nsect,1,0
TED R21  ngrp 0 (NI=9):  nmat,J1,J2,C,length,0,nsect,1,0
TED R31  ngrp 0 (NI=15): nmat,J1,J2,J3,0,X1,Y1,X2,Y2,X3,Y3,0,nsect,1,0
TED R41  ngrp 0 (NI=18): nmat,J1,J2,J3,J4,0,X1,Y1,X2,Y2,X3,Y3,X4,Y4,0,nsect,1,0
TED MT21 ngrp 0 (NI=12): 0,nfluid,J1,J2,Am,Dh,length,0,0,nsect,1,0
TED MT42 ngrp 0 (NI=15): nmat,nfluid,J1,J2,J3,J4,Ak,C,Am,length(3-4),0,0,nsect,
                         1,0
TED MT62 ngrp 0 (NI=28): nmat,nfluid,J1,J2,J3,J4,J5,J6,Tf,Ff,Wt,Wb,Dh,X3,Y3,X6,
                         Y6,X5,Y5,X2,Y2,length(2-5),length(3-6),0,0,nsect,1,0
TED C32  ngrp 0 (NI=12): 0,nfluid,J1,J2,J3,C,length(2-3),0,0,nsect,1,0
TED C42  ngrp 0 (NI=13): 0,nfluid,J1,J2,J3,J4,C,length(3-4),0,0,nsect,1,0
TED C62  ngrp 0 (NI=23): 0,nfluid,J1,J2,J3,J4,J5,J6,Tf,Ff,X3,Y3,X6,Y6,X5,Y5,X2,
                         Y2,0,0,nsect,1,0
```

5.3.2 The Data Sets TED GRPS, TED MPD, TDOF ID

```
TED  GRPS 0 0: this data set contains the KL array - see LDKL.
TED  MPD  0 0: this data set contains MXMAT(2,5) of labeled common MDATA3 (see
               LDMP).
TDOF ID   0 0: this data set contains the ID array as computed in EXTGEO.
```

5.4 RESET Controls and Execution Commands

The following RESET controls and execution command are referenced in LDTGEO but are not described in Volume 1 of the SPAR Thermal Analysis Reference Manual.

RESET Controls:
PENT - not used.
LBLOCK - element data blcok length (LBSIZ in labeled comman EDATA1).
EXPMAT - see NCMAT in labeled common CTGEO2.
KL - see execution command Eij below.

Execution Command:
When RESET Control KL=ON, the number of groups for each element type can be specified by the command

Eij ngroups

This command is processed by subroutine READKL.

Section 6 - PROCESSOR SSTA

6.1 SSTA Central Memory Allocation

```
KL        - element information array       - 20*NLS
MP        - material properties pointer     - 3*NMP
PROP      - material properties buffer      - MBSIZ

Assembly (ASMBL):
N1  LC,KE - skyline vector                  - MAXO(LRSKY,LADJ(LRT))
N2  A     - K matrix block                  - LRK
N3  JS    - joint elimination sequence      - NJTS
N4  T     - temperature vector              - LRT
    ED    - element data buffer             - LBSIZ
    NP    - element nodes                   - MXNPTS
    LM    - element degrees of freedom      - MXNDF*MXNPTS
    ET    - element temperature vector      - MXNDF*MXNPTS
    EK    - element K matrix                - (MXNDF*MXNPTS)**2
    TR    - mass-transport rates            - LRMT

ID correction (SDKM):
N1  LC,KE
N2  A
N3  JS
N4  ID    - number of freedoms at nodes     - NJTS

Prescribed temperatures (ATKM):
N1  LC,KE
N2  A
N3  JS
N4  NPAT  - prescribed temperature nodes    - NTEMPL
    ATF   - modified diagonal terms         - NTEMPL

Factoring (FACTOR):
N1  LC,KE,KM,KB
N2  A
    A     - K matrix block                  - LRK

Radiation exchange vector (REVEC):
N1  T     - temperature vector
N2  A IA  - see REVEC listing

Thermal load vector (NLTLV,LTLV):
N1  T     - temperature vector
N2  PHI   - thermal load vector             - NLCS*LADJ(LRT)
N3  JS    - joint elimination sequence      - NJTS
N4  ED    - element data buffer             - LBSIZ
N5  NP    - element nodes                   - MXNPTS
N6  LM    - element degrees of freedom      - MXNPTS*MXNDF
N7  ET    - element temperature vector      - MXNPTS*MXNDF
N8  EK    - element K matrix or load vector - (MXNPTS*MXNDF)**2
N9  TA    - element excitation string       - MAXO(NLCS*LRQ,LRMT)
```

Prescribed temperature correction (NLATL,LATL):

N1	T	- temperature vector (NLATL only)	
N2	PHI	- thermal load vector	
N3	JS	- joint elimination sequence	
N4	NPAT	- prescribed temperature nodes	- NTEMPL
N5	AF	- modified diagonal terms	- NTEMPL
N6	AT	- prescribed temperatures	- NTEMPL*NLCS

Solution (TSOL,CSTVEC,CLSVEC):

N1	LC	- skyline vector
N2	T	- load vector, solution vector
N3	JS	- joint elimination sequence

The blank common locations N1, N2, etc. are computed in EXSSTA.

6.2 SSTA Subroutines

Subroutine LDSSTA
Called by: MPSSTA
Calls: RSET, READER, STSSTA
Function: Reads program control information. Initializes data.

Subroutine STSSTA(A,KSPACE)
Called by: LDSSTA
Calls: LDKL, LDMP, LDIDJS, LDKSD, PRBTYP, LCCR, CHKRED, SSTACA
Function: Allocates central memory.
Calls routines for reading element, material property, and thermal excitation data.
Calls skyline computation routines.
A = working data space.
KSPACE = amount of data space available.

Subroutine LDKSD(KL,LC,ED,JS,NP,LM)
Called by: STSSTA
Calls: LDJS, LINES, CFPNI, CEDOF, CLC, CSLV
Function: Controls computation of skyline vector.
Determines if K matrices are present in library for executions that start with an existing assembled or factored K matrix.
KL = see LDKL.
LC(I) = lowest degree of freedom which couples with degree of freedom I after return from subroutine CLC.
= skyline vector after return from subroutine CSLV.
ED = element data buffer with node points in ED(K).
JS = joint elimination sequence vector.
NP = element node point vector.
LM = element degree of freedom vector.

```
Subroutine PRBTYP(KL,MP)
Called by: STSSTA
Function:  Determines type of problem to be solved.
Common:    NLCOM1/ NLTYP
                   NLTYP = 0 - linear problem, no temperature dependent prop-
                               erties or radiation elements.
                         = 4HLINE - forced linear problem, properties wiil be
                                    evaluated according to RESET control TEMP.
                                    Radiation elements are ignored.
                         = 4HNONL - nonlinear problem.
KL = see LDKL.
MP = see LDMP.


Subroutine LCCR(KL,T,KORE)
Called by: STSSTA
Calls:     CHKFPS, CFPNI, CHKIDS
Function:  Checks for existence of STAT TEMP data set when NFLUX=1, NPRESS=1,
           or NURFLX>0.
           Checks for existence of thermal excitation data sets.
           Checks compatibility of TEMP NODE and APPL TEMP data sets.
           Checks central memory size requirements.
           Initializes STAT TEMP data sets.
KL   = see LDKL.
T    = working data space.
KORE = available data space.


Subroutine SSTACA(KORE)
Called by: STSSTA
Function:  Allocates central memory for flux, pressure drop, and radiation
           exchange computations.
KORE = available data space.


Subroutine EXSSTA(KL,MP,PROP,A)
Called by: GOSSTA
Calls:     LDJS, ASMFAC, REVEC, NLTLV, NLATL, TSOL, CSTVEC, LTLV, LATL, CLSVEC,
           SFLUX, SMTPD, RFLUX
Function:  Controls solution process for nonlinear analyses.
           Controls solution process for linear analyses.
           Calls flux and pressure drop computation routines.
KL   = see LDKL.
MP   = see LDMP.
PROP = see LDMP.
A    = working data space.
```

Subroutine ASMFAC(KL,MP,PROP,LC,AK,JS,A)
Called by: EXSSTA
Calls: ASMBL, SDKM, CFPNI, ATKM, FACTOR
Function: Controls assembly and factoring of the K matrix.
KL = see LDKL.
MP = see LDMP.
PROP = see LDMP.
LC = skyline vector.
AK = K matrix buffer.
JS = joint elimination sequence vector.
A = working data space.

Subroutine ASMBL(KL,MP,PROP,LC,KE,A,JS,T,ED,NP,LM,ET,EK,TR)
Called by: ASMFAC
Calls: WRKB, LINES, CFPNI, CEDOF, ELTEMP, MPROP, EK21, EK31, EK41, EK61, EK81, EC21, EC31, EC41, ER21, ER31, ER41, ERK, EMT21, EMT42, EMT62, EC32, EC42, EC62, ADDK, EEPROP, EXPTEK, EXPRK2
Function: Controls formation and assembly of element K matrices.
KL = see LDKL.
MP = see LDMP.
PROP = see LDMP.
LC = skyline vector.
KE = last equation in block vector.
A = K matrix buffer.
JS = joint elimination sequence vector.
T = temperature vector.
ED = element data buffer:
material numbers in ED(I1),
node points in ED(I2),
section properties in ED(I3),
coordinates in ED(I4),
last temperature points in ED(I5).
NP = element node point vector.
LM = element degree of freedom vector.
ET = element nodal temperatures.
EK = element K matrix buffer.
TR = mass-transport buffer.

Subroutine SDKM(LC,KE,JS,ID)
Called by: ASMFAC
Function: Removes singularities due to unrestrained degrees of freedom from the assembled K matrix.
LC = skyline vector.
KE = last equation in block vector.
JS = joint elimination sequence vector.
ID(N) = number of active freedoms at node N.

Subroutine ATKM(LC,KE,A,JS,NPAT,ATF)
Called by: ASMFAC
Function: Modifies the K matrix to account for prescribed nodal temperatures. Each term on the K matrix diagonal corresponding to a prescribed nodal temperature is replaced by its initial value times AKFAC (see RESET control ATFAC and labeled common block QDATA3).
LC = skyline vector.
KE = last equation in block vector.
A = K matrix buffer.
JS = joint elimination sequence vector.
NPAT = vector of node points at which temperatures are prescribed.
ATF = vector of modified diagonal terms.

Subroutine NLTLV(KL,MP,PROP,T,PHI,JS,ED,NP,LM,ET,EK,TA)
Called by: EXSSTA
Calls: LINES, CFPNI, CEDOF, ELTEMP, MPROP, EK21, EK31, EK41, EK61, EK81, ADDKLD, QK21, QK31, QK41, QK61, QK81, ADDLLD, EC21, EC31, EC41, ADDCLD, ER21, ER31, ER41, ADDRLD, EMT21, EMT42, EMT62, EC32, EC42, EC62, EEPROP, EXPTEK, EXPSCQ, ADDVEC
Function: Controls formation of the nonlinear thermal load vector.
KL = see LDKL.
MP = see LDMP.
PROP = see LDMP.
T = temperature vector.
PHI = nonlinear thermal load vector.
JS = joint elimination sequence vector.
ED = element data buffer:
material numbers in ED(I1),
node points in ED(I2),
section properties in ED(I3),
coordinates in ED(I4),
last temperature points in ED(I5).
NP = element node point vector.
LM = element degree of freedom vector.
ET = element nodal temperatures.
EK = element K matrix.
TA = convective exchange temperature, source strengths, or mass-transport rates.

Subroutine ADDKLD(PHI,LM,ET,EK,KDF)
Called by: NLTLV
Function: Incorporates element K matrix contributions into the nonlinear thermal load vector.
PHI = nonlinear thermal load vector.
LM = element degree of freedom vector.
ET = element nodal temperatures.
EK = element conduction matrix.
KDF = number of element degrees of freedom.

Subroutine ADDCLD(PHI,LM,ET,EK,KDF,TA)
Called by: NLTLV
Function: Adds convection element contributions to the nonlinear thermal load vector.
PHI = nonlinear thermal load vector.
LM = element degree of freedom vector.
ET = element temperature vector.
EK = element convection matrix.
KDF = number of element degrees of freedom.
TA = element convective exchange temperature.

Subroutine ADDRLD(PHI,LM,ET,EK,KDF,TA)
Called by: NLTLV
Function: Adds radiation element contributions to the nonlinear thermal load vector.
PHI = nonlinear thermal load vector.
LM = element degree of freedom vector.
ET = element temperature vector.
EK = element radiation matrix.
KDF = number of element degrees of freedom.
TA = radiation exchange temperature.

Subroutine NLATL(T,PHI,JS,NPAT,AF,TA,NDF)
Called by: EXSSTA
Function: Modifies the nonlinear thermal load vector to account for prescribed nodal temperatures. Each term in the thermal load vector corresponding to a prescribed nodal temperature is replaced by the appropriate value of AF multiplied by the prescribed temperature.
T = temperature vector.
PHI = nonlinear thermal load vector.
JS = joint elimination sequence vector.
NPAT = vector of prescribed temperature node points.
AF = ATF vector computed by ATKM.
TA = vector of prescribed nodal temperatures.
NDF = number of freedoms at each node point.

Subroutine LTLV(KL,MP,PROP,T,PHI,LX,JS,ED,NP,LM,ET,Q,TA)
Called by: EXSSTA
Calls: LINES, CFPNI, CEDOF, MPROP, QK21, QK31, QK41, QK61, QK81, EC21, EC31, EC41, ADDVEC, ELTEMP, EEPROP, EXPSCQ
Function: Controls formation of the linear thermal load vector.
KL = see LDKL.
MP = see LDMP.
PROP = see LDMP.
T = temperature vector.
PHI = linear thermal load vector.
LX = adjusted length (LADJ) of T and PHI.
JS = joint elimination sequence vector.
ED = element data buffer:
material numbers in ED(I1),
node points in ED(I2),
section properties in ED(I3),
coordinates in ED(I4).
NP = element node point vector.
LM = element degree of freedom vector.
ET = element temperature vector.
Q = element unit thermal load vector.
TA = source strengths or convective exchange temperatures.

Subroutine LATL(PHI,LX,JS,NPAT,AF,TA,LTA,NDF)
Called by: EXSSTA
Function: Modifies the linear thermal load vector to account for prescribed nodal temperatures. Each term in the thermal load vector corresponding to a prescribed nodal temperature is replaced by the appropriate value of AF multiplied by the prescribed temperature.
PHI = linear load vector.
LX = adjusted length (LADJ) of PHI.
JS = joint elimination sequence vector.
NPAT = vector of prescribed temperature node points.
AF = ATF vector computed by ATKM.
TA = vector of prescribed nodal temperatures.
LTA = adjusted length (LADJ) of TA.
NDF = number of freedoms at each node point.

Subroutine EK21(A,XL,EK)
Called by: ASMBL, NLTLV
Function: Computes conduction matrix for K21 elements.
A = area.
XL = length.
EK = element conduction matrix.

Subroutine EK31(TH,X,EK)
Called by: ASMBL, NLTLV
Function: Computes conduction matrix for K31 elements.
TH = thickness.
X = element coordinates.
EK = element conduction matrix.

Subroutine EK41(TH,X,EK)
Called by: ASMBL, NLTLV
Function: Computes conduction matrix for K41 elements.
TH = thickness.
X = element coordinates.
EK = element conduction matrix.

Subroutine EMT21(SECT,XL,EK,ET,MP,PROP,TR)
Called by: ASMBL, NLTLV
Calls: FPROP, ALPHA
Function: Computes conduction and mass-transport matrices for MT21 elements.
SECT = element conduction area.
XL = element length.
EK = element conduction and mass-transport matrices (summed).
ET = element temperature vector.
MP = see LDMP.
PROP = see LDMP.
TR = element mass-transport rate.

Subroutine EMT42(SECT,XL,EK,ET,MP,PROP,TR)
Called by: ASMBL, NLTLV
Calls: EC42, MPROP, ALPHA
Function: Computes conduction, convective-exchange, and mass-transport matrices for MT42 elements.
SECT = surface conduction area, surface convective-exchange width, fluid conduction area.
XL = length of convective-exchange surface.
EK = element conduction, convective-exchange, and mass-transport matrices (summed).
ET = element temperature vector.
MP = see LDMP.
PROP = see LDMP.
TR = element mass-transport rate.

Subroutine EMT62(SECT,X,EK,ET,MP,PROP,TR)
Called by: ASMBL, NLTLV
Calls: EC62, CFFC, MPROP, EK41, ALPHA
Function: Computes conduction, convective-exchange, and mass-transport matrices for MT62 elements.
SECT = fin thickness, fin area factor, top wall width, bottom wall width.
X = fin coordinates X(2,4), top wall length, bottom wall length.
EK = element conduction, convective-exchange, and mass-transport matrices (summed).
ET = element temperature vector.
MP = see LDMP.
PROP = see LDMP.
TR = element mass-transport rate.

Subroutine EK61(X,EK)
Called by: ASMBL, NLTLV
Calls: EK81
Function: Computes conduction matrix for K61 elements.
X = element coordinates.
EK = element conduction matrix.

Subroutine EK81(X,EK)
Called by: ASMBL, NLTLV, EK61
Calls: DERIV
Function: Computes conduction matrix for K81 elements.
X = element coordinates.
EK = element conduction matrix.

Subroutine TSOL(LC,KE,T,LX,NX)
Called by: EXSSTA
Calls: CSPDP, XSOL, CDPSP
Function: Controls solution process.
LC = skyline vector.
KE = last equation in block vector.
T = thermal load vectors before solution, change in temperature (nonlinear) or temperature (linear) vectors after solution.
LX = adjusted length (LADJ) of T.
NX = number of load cases to process.

Subroutine CSTVEC(T,DT,NDF,JS,CNVR,N)
Called by: EXSSTA
Function: Computes T+DT vector, checks convergence, and checks for negative temperatures.
T = initial temperature vector on entry into CSTVEC, T+DT on exit from CSTVEC.
DT = incremental temperature (solution) vector.
NDF = number of freedoms at each node point.
JS = joint elimination sequence vector.
CNVR = maximum DT/T.
N = node at which maximum DT/T occurs.

Subroutine CLSVEC(T,PHI,LX,NX,JS,NDF)
Called by: EXSSTA
Function: Writes linear solution vectors into nodal temperature vectors.
T = nodal temperature vectors.
PHI = solution vectors.
LX = adjusted length (LADJ) of T and PHI.
NX = number of T and PHI vectors to process.
JS = joint elimination sequence vector.
NDF = number of freedoms at each node point.

Subroutine SFLUX(KL,MP,PROP,T,LX,ED,NP,ET,TA,FLUX)
Called by: EXSSTA
Calls: LINES, CFPNI, ELTEMP, EFLUX, EEPROP, EXPFLX
Function: Controls the computation of steady-state element heat fluxes.
KL = see LDKL.
MP = see LDMP.
PROP = see LDMP.
T = temperature vectors.
LX = adjusted length (LADJ) of T.
ED = element data buffer:
material numbers in ED(I1),
node points in ED(I2),
section properties in ED(I3),
coordinates in ED(I4),
last temperature points in ED(I5).
NP = element node point vector.
ET = element temperature vector.
TA = convective exchange temperatures and mass-transport rates.
FLUX = fluxes.

Subroutine SMTPD(KL,MP,PROP,T,LX,ED,NP,ET,TR,P)
Called by: EXSSTA
Calls: LINES, CFPNI, ELTEMP, EPD, EEPROP, EXPPD
Function: Controls the computation of steady-state fluid pressures.
KL = see LDKL.
MP = see LDMP.
PROP = see LDMP.
T = temperature vectors.
LX = adjusted length (LADJ) of T.
ED = element data buffer:
material numbers in ED(I1),
node points in ED(I2),
section properties in ED(I3),
coordinates in ED(I4),
last temperature point in ED(I5).
NP = element node point vector.
ET = element temperature vector.
TR = mass-transport rates.
P = fluid pressures.

Subroutine SRFLUX(KL,MP,PROP,A)
Called by: EXSSTA
Calls: RFLUX
Function: Controls the computation of steady-state radiation exchange fluxes.
KL = see LDKL.
MP = see LDMP.
PROP = see LDMP.
A = working data space.

6.3 Data Sets Generated by SSTA

The following K matrix data sets are stored when execution control KSAVE (see Section 6.4) is used. The TOC parameters are the same as for the factored K matrix data sets described in Section 5.2.8 of the SPAR Thermal Analaysis Reference Manual - Volume 1.

```
K  SKY  nrms mxbw: assembled symmetric K matrix
K  POIN nrms mxbw: skyline vector of K SKY
KA SKY  nrms mxbw: assembled asymmetric K matrix
KA POIN nrms mxbw: skyline vector of KA SKY
```

The following radiation exchange data set is stored on the unit specified by RESET control RGLIB. The TOC parameters are NI=2+maximum number of nodes in any radiation element and NJ=the number of radiation elements.

```
REX GEOM 0 0: area,material,list of nodes
```

The following radiation exchange data set is stored on the unit specified by RESET control RGLIB when RESET control RCC>2. The TOC parameters are NI=1 and NJ=LRT.

```
REX VEC 0 0: nodal radiation exchange load
```

The following radiation exchange data sets are stored on the unit specified by RESET control RGLIB and RESET control RCC>3. The TOC parameters are NI=1 and NJ=the number of radiation element.

```
REX AB 0 0: element absorptivity
REX Q  0 0: element emitted heat
REX QA 0 0: element absorbed heat
```

6.4 RESET Controls and Execution Commands

The following RESET controls and execution command are referenced in LDSSTA but are not listed in Volume 1 of the SPAR Thermal Analysis Reference Manual.

RESET Controls:

SET - load set identifier.

LBLOCK - element data block length. If LBLOCK is not specified, LBSIZ (see labeled common EDATA1) is set equal to the largest block length of all TED Eij data sets.

START - START=K causes a steady-state analysis to be started with an assembled K matrix.

ATFAC - factor which multiplies the diagonal K matrix terms corresponding to nodes at which nodal temperatures are prescribed (see labeled common block QDATA3 and subroutine ATKM).

ZFIN - see FEF62.

PITER - see NPITS in labeled common block PDATA2.

PCONV - see labeled common block PDATA2.

LZERO - see ZEROL in labeled common MTCOM1.

AZERO - see ZEROA in labeled common MTCOM1.

RGLIB - see Section 6.3 - radiation exchange data sets.

RCC - see Section 6.3 - radiation exchange data sets.

Execution Command:

KSAVE=lib causes the assembled K matrix to be saved on unit lib. If lib is not given, K is saved on the unit specified by RESET control KLIB.

Section 7 - PROCESSOR TRTA

7.1 TRTA Central Memory Allocation (EXTRTA)

KL see labeled common block CTRTA1.
MP "
PROP "
TIMK "
TIMS "
TSCR "
NPAT "
NPAN "

Time-interval computations (ASMKCR,INVCAP,DTCOMP,LDATS):

N1	T	- temperature vector	- LRT
N2	TA	- element mass-transport rates	- 2*LADJ(LRQ)
N13	ED	- element data block	- LBSIZ
N14	NP	- element nodes	- MXNPTS
N15	LM	- element degrees of freedom	- MXKDF
N16	ET	- element temperatures	- MXKDF
N17	EK	- element K matrix	- MXKDF*MXKDF
N18	EC	- element capacity matrix	- MXKDF*MXKDF
N3	LCAN	- arithmetic node skyline vector	- NANPS + 3*NRK
		or implicit method skyline vector	- LRT + 3*NRK
N4	AKAN	- arithmetic node K matrix block	- LRK
		or implicit method K matrix block	- LRK
N5	AK	- element K matrix block (explicit)	- LREK
		or modified diagonal terms (implicit)	- NTEMPL
N6	CI	- inverted capacity matrix (explicit)	- LRT
		or capacity matrix (implicit)	- LRT
N7	CK	- diagonal convection matrix (explicit)	- LRT
	JS	- joint elimination sequence (implicit)	- LRT
N8	RK	- diagonal radiation matrix	- LRT
N9	Q	- thermal load and load rate vectors	- 2*LRT
N10	QAN	- arithmetic node load vector (explicit)	- NDP*NANPS
N11	RKAN	- arithmetic node radiation matrix	- NANPS
		or implicit method radiation matrix	- LRT
N12	JT	- element degreee of freedom vector	- MXKDF

Radiation exchange load vector (REVEC):

N1	T	- temperature vector
N2	RQ	- radiation load vector
N3		- NWWR words of central memory starting at this address are temporarily stored on mass storage if necessary (see TRTACA).

Temperature vector computations (TTDS):

N1	T	- temperature vector
N2	TDOT	- temperature time derivative vectors - LRT*(NTERMS-1)
N3	LCAN	- arithmetic node or implicit method skyline vector
N4	AKAN	- arithmetic node or implicit method K matrix block
N5	AK	- element K matrix block
N6	CI	- inverted capacity matrix (explicit) capacity matrix (implicit)
N7	CK	- diagonal convection matrix
N8	RK	- diagonal radiation matrix
N9	Q	- load and load rate vectors
N10	QAN	- arithmetic node load vector
N11	RKAN	- arithmetic node or implicit method radiation matrix
N12	JT	- element degree of freedom vector

The blank common locations N1, N2, etc. are computed in TRTACA.

7.2 TRTA Subroutines

Subroutine LDTRTA
Called by: MPTRTA
Calls: RSET, READER, ILTIM, STTRTA
Function: Reads program control information. Initializes data.

Subroutine IETIM(IL,RL)
Called by: LDTRTA
Function: Sets up implicit method analysis parameters.
IL = see LDTRTA.
RL = see LDTRTA.

Subroutine STTRTA(A,KSPACE)
Called by: LDTRTA
Calls: LDKL, CTCS, LDMP, CTIMES, TIMINC, LDIDJS, LDTIME, LDAND, MPTIME, CHKRED, TRTACA
Function: Calls routines for reading element, material property, and thermal excitation data.
Allocates central memory.
A = working data space.
KSPACE = amount of data space available.

Subroutine CTCS
Called by: STTRTA
Function: Computes Taylor series coefficients 1/n!.

Subroutine CTIMES(TIME,KORE)
Called by: STTRTA
Function: Computes time interval and save time vectors.
Checks for existence of TRTA TEMP data set when NFLUX=1, NPRESS=1, or NURFLX>0.
TIME = time interval vector (NTIPS), save time vector (NRT).
KORE = available working space.

```
Subroutine TIMINC(T,KSPACE,I2)
Called by: STTRTA
Function:  Checks the XXXX TIME data sets to make sure that times increase
           sequentially and bracket the starting and ending times.
Common:    CTIME4/NTP(10),LTP(10)
           NTP(I) = number of time points in XXXX TIME data set I.
           LTP(I) = last point referenced in XXXX TIME data set I.
T      = vector of times contained in all XXXX TIME data sets.  Listed in
         order of the names contained in dimensioned array N1.
KSPACE = amount of available central memory space.
I2     = length of T vector.


Subroutine LDTIME(KL,ATNP,KORE)
Called by: STTRTA
Calls:     CHKFPS, CFPNI, CHKIDS
Function:  Checks thermal excitation data sets.
KL   = see LDKL.
ATNP = vector of node points at which temperatures are prescribed.
KORE = available central memory data space.


Subroutine LDAND(NPAT,NPAN,CORE)
Called by: STTRTA
Calls:     CHKFPS, CFPNI, CHKIDS
Function:  Reads arithmetic nodes.  Eliminates arithmetic nodes coincident
           with applied temperature nodes.
NPAT = applied temperature nodes.
NPAN = arithmetic nodes.
KORE = available data space.


Subroutine MPTIME(KL,MP)
Called by: STTRTA
Function:  Checks to make sure that the number of blocks in XXXX PROP and XXXX
           COEF data sets agree with the length of the PROP TIME and COEF TIME
           data sets.
KL = see LDKL.
MP = see LDMP.
```

```
Subroutine TRTACA(KL,NPAN,A,KORE)
Called by: STTRTA
Calls:     LDJS, CSLVIM, WRKB, CSLVAN
Function:  Allocates central memory and computes addresses.
KL   = see LDKL.
NPAN = arithmetic node vector.
A    = working data space.
KORE = data space available.
```

```
Subroutine CSLVIM(KL,ED,NP,LM,LC,LCC,KS1,KS2)
Called by: TRTACA
Calls:     CFPNI, CLC, CSLV
Function:  Computes the skyline vector for the implicit method.
KL   = see LDKL.
ED   = element data buffer.
NP   = element node points.
LM   = element degree of freedom vector.
LC   = skyline vector on return.
LCC  = skyline vector.
KS1  = space available for skyline vector and 1 block of K matrix.
KS2  = maximum space available for 1 K matrix block.
```

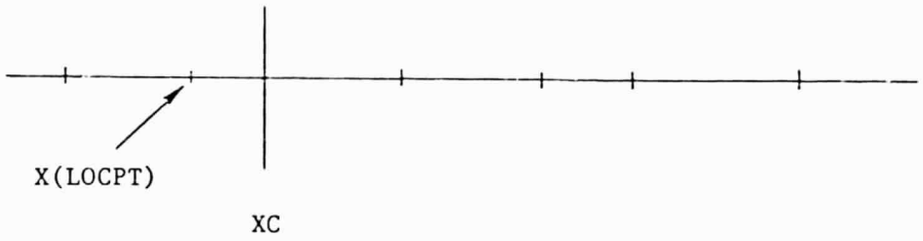
```
Subroutine CLSVAN(KL,NPAN,ED,NP,LM,LC,KORE)
Called by: TRTACA
Calls:     CFPNI, NANPTS, CLC, CSLV
Function:  Computes the arithmetic node skyline vector.
KL   = see LDKL.
NPAN = arithmetic node vector.
ED   = element data buffer.
NP   = element node points.
LM   = element degree of freedom vector.
LC   = arithmetic node skyline vector.
KORE = avaiable data space.
```

```
Function NANPTS(NPAN,NP,NPTS,LM)
Called by: CLSVAN
Function:  Determines the number of arithmetic nodes for an element.
NPAN = arithmetic node vector.
NP   = element node point vector.
NPTS = number of element node points.
LM   = element arithmetic degree of freedom vector.
```

```
Subroutine EXTRTA(KL,MP,PROP,TIMK,TIMS,TSCR,NPAT,NPAN,A)
Called by: GOTRTA
Calls:     INTEMP, WRRDS, SAVESM, CTI2, ASMKCR, INVCAP, DTCOMP, FACKAN, TILV,
           FACKIM, TEMPAN, SAVET, ITDS, TFLUX, TMTPD, TRFLUX
Function:  Controls the solution process for transient analyses.
KL   = see LDKL.
MP   = see LDMP.
PROP = see LDMP.
TIMK = vector of interval times.
TIMS = vector of times at which results are saved.
TSCR = see T in TIMINC.
NPAT = vector of nodes at which temperatures are prescribed.
NPAN = arithmetic node vector.
A    = working data space.


Subroutine INTEMP(T)
Called by: EXTRTA
Function:  Reads or computes the initial temperature vector.
           Writes the initial temperature vector into TRTA TEMP.
T = initial temperature vector.


Subroutine WRRDS(A,IOWR)
Called by: EXTRTA
Function:  Stores that portion of central memory required by REVEC.
Common:    CWRRDS/ MXNWPB,NBLKS,NWPB,NWLB,KADR
                   MXNWPB = maximum number of words that can be transfered
                            with one call to RIO.
                   NBLKS  = number of blocks to write or read.
                   NWPB   = number of words per block.
                   NWLB   = number of words in the last block.
                   KADR   = sector address.
A    = data buffer.
IOWR = 0, initialize data.
     = 1, write NWWR words from central memory.
     = 2, read NWWR words from central memory.


Subroutine CTI2(TIME,TSCR)
Called by: EXTRTA
Calls:     LTPT
Function:  Computes interval end time.
TIME = time interval vector.
TSCR = see T in TIMINC.
```

Function LTPT(TIME,T,R,K)
Called by: CTI2, TPROP, AMTR, TTLV
Calls: LOCPT
Function: Computes point to left of T in TIME array.
TIME = see T in TIMINC.
T = time
R = ratio
K = XXXX TIME data set identifier, see TIMINC.
LTPT = point to left of T.

Subroutine LOCPT(X)
Called by: LTPT, LDTAF
Function: Computes the nearest point in the X array to the left of XC.
Common: CLOCPT/ LX,NST,XC
Input: X = array of values in ascending order.
LX = length of X array.
NST = starting point (usually LOCPT from previous call).
XC = current value of X.
Output: LOCPT = left adjusted point in X array closest to XC.
If LX<2, LOCPT = 0.
If XC<X(1), LOCPT = 1.
If XC>X(LX), LOCPT = LX-1.



```
Subroutine NEXTKB(IA,IOWR)
Called by: ASMKCR, ASMEKS, DTCOMP, TTDS
Function:  Writes or reads element K matrix blocks.
Common:    EKCOM1/ NUK,NZK,LRK,NRK
                   NUK = element K matrix unit.
                   NZK = sector address of first block.
                   LRK = block length.
                   NRK = number of blocks.
Common:    EKCOM2/ NBC,NEKS,LOC,NWK,KTYPE,KDF
                   NBC   = current block in central memory.
                   NEKS  = number of element data records in block.
                   LOC   = current location in block.
                   NWK   = number of words in the current element data record.
                   KTYPE = element matrix type:
                         = 1, square matrix (asymmetric).
                         = 2, lower triangular string (symmetric).
                   KDF   = number of words in element K matrix (JTPACK).
                         = number of element freedoms (JTCOMP).
                           matrices (JTCOMP).
                   KDFAN = number of arithmetic node degrees of freedom.
IA    = element K matrix buffer.
        IA(1) = number of element data records in the block (NEKS).
        Each element data record is structured as follows:
         NWK
         KTYPE
         KDF (number of element freedoms)
         KDFAN
         freedom list
         element matrix
IOWR =  0, initialization.
     =  1, write K matrix block.
     = 10, write last K matrix block.
     =  2, read K matrix block.


Subroutine TPROP(MP,P,TIME,MXMAT,LOCMAT,T1,T2,MT1,MT2)
Called by: ASMKCR, FTCOEF
Calls    : LTPT, RDPQ
Function:  Time averages properties over interval defined by T1 and T2.
MP      = see LDMP.
P       = see PROP in LDMP.
TIME    = see T in TIMINC.
MXMAT   = see LDMP.
LOCMAT  = see LDMP.
T1,T2   = interval starting and ending times.
MT1,MT2 = properties for material types MT1 through MT2 are computed
          (see MTYPE in LDMP).
```

Subrotuine RDPQ(NU,A,LA,KADR,LAST,LOC)
Called by: TPROP, AMTR
Function: Reads time-dependent properties and mass-transport rates.
NU = unit.
A = data buffer.
LA = length of A.
KADR = mass storage sector address.
LAST = last record read.
LOC = record to read.

Subroutine WPROP(P,NI,NJ,MTYPE,NMAT)
Called bt: TPROP
Function: Prints properties online.
P = property array.
NI,NI = P dimensions.
MTYPE = property type (see MTYPE in LDMP).
NMAT = property number.

Function JTCOMP(IA,JT)
Called by: DTCOMP, TTDS
Function: Extracts element data string from the element K matrix block.
IA = element data string.
JT = list of element degrees of freedom.
JTCOMP = location in element K matrix block of element K matrix.

```
Subroutine ASMKCR(KL,MP,PROP,TIME,NPAN,T,EMT,ED,NP,LM,ET,EK,EC,LCAN,AKAN,AK,
                  CI,CK,RK,KP)
Called by: EXTRTA
Calls:     NEXTKB, WRKB, TPROP, LEXT, EXPTEK, AMTR, LINES, CFPNI, CEDOF,
           NADFS, ELTEMP, MPROP, TEK21, TEK31, TEK41, TEK61, TEK81, CFMDM2,
           ADDVEC, ADDK, ADDKAN, ASMEKS, EC21, EC31, EC41, CFMDM1, ER21, ER31,
           ER41, TEMT21, TEMT42, TEMT62, EC32, EC42, EC62, EEPROP, EXPRK1
Function:  Computes the system K and capacity matrices.
KL   = see LDKL.
MP   = see LDMP.
PROP = see LDMP.
TIME = see T in TIMINC.
NPAN = arithmetic node vector.
T    = temperature vector.
EMT  = element mass-transport rates.
ED   = element data buffer:
       material numbers in ED(I1),
       node points in ED(I2),
       section properties in ED(I3),
       coordinates in ED(I4),
       last temperature points in ED(I5).
NP   = element node point vector.
LM   = element degree of freedom vector.
ET   = element nodal temperature vector.
EK   = element K matrix.
EC   = element capacity matrix (square).
LCAN = arithmetic node or implicit method skyline vector.
AKAN = arithmetic node or implicit method K matrix block.
AK   = element K matrix block - comprised of a string of element K matrices,
       (see NEXTKB for block structure).
CI   = diagonal capacity matrix.
CK   = diagonal convection matrix.
RK   = diagonal radiation matrix.
KP   = see ADDK.


Subroutine AMTR(TIME,EMTR,T1,T2)
Called by: ASMKCR
Calls:     LTPT, RDPQ
Function:  Time-averages element mass-transport rates over the time interval.
TIME  = see T in TIMINC.
EMTR  = element mass-transport buffer.
T1,T2 = tie interval end points.
```

Function NADFS(NPAN,NP,NPTS,LM,NDF,FNANP)
Called by: ASMKCR
Function: Computes element arithmetic node vector.
NPAN = arithmetic node vector.
NP = element node point vector.
NPTS = number of element node points.
LM(N,2) = element degree of freedom corresponding to element arithmetic node point N.
LM(N,3) = location in NPAN of element arithmetic node point N.
NDF = number of freedoms per element node point.
FNANP = number of element arithmetic node points (stored as a floating point number in element TED Eij ngrp data set, see Section 5.3).
NADFS = FNANP

Subroutine ADDKAN(LC,AK,LM,EK,KDF,EKAN,NADF,KP)
Called by: ASMKCR
Calls: ADDK
Function: Controls assembly of arithmetic node K matrix.
LC = arithmetic node skyline vector.
AK = arithmetic node K matrix block.
LM = see NADFS.
EK = element K matrix.
KDF = number of element degrees of freedom.
EKAN = element arithmetic node K matrix.
NADF = number of element arithmetic node degrees of freedom.
KP = see KP in ADDK.

Subroutine ASMEKS(AK,EK,LM,KDF,KT)
Called by: ASMKCR
Calls: JTPACK
Function: Inserts element K matrices into the element K matrix block.
AK = element K matrix buffer, see IA in NEXTKB.
EK = element K matrix.
LM = element degree of freedom vector.
KDF = number of element degree of freedoms.
KT = matrix type - 1, square (asymmetric).
2, lower triangular (symmetric).

Function JTPACK(IA,JT,NJT)
Called by: ASMEKS
Function: Inserts element type and degree of freedom data into the element K matrix block.
IA = see NEXTKB.
JT = element degree of freedom vector.
NJT = number of element degrees of freedom.
JTPACK = location in element K matrix block of element K matrix.
= 0, not enough room in element K matrix block.

Subroutine TEK21(A,XL,EK,EC)
Called by: ASMKCR
Function: Computes the conduction and capacity matrices for K21 elements.
XL = element length.
A = element area.
EK = element conduction matrix (square).
EC = element capacity matrix (square).

Subroutine TEK31(TH,X,EK,EC)
Called by: ASMKCR
Function: Computes the conduction and capacity matrices for K31 elements.
TH = element thickness.
X = element coordinates.
EK = element conduction matrix (square).
EC = element capacity matrix (square).

Subroutine TEK41(TH,X,EK,EC).
Called by: ASMKCR
Function: Computes the conduction and capacity matrices for K41 elements.
TH = element thickness.
X = element coordinates.
EK = element conduction matrix (square).
EC = element capacity matrix (square).

Subroutine TEK61(X,EK,EC)
Called by: ASMKCR
Calls: TEK81
Function: Computes the conduction and capacity matrices for K61 elements.
X = element coordinates.
EK = element conduction matrix (square).
EC = element capacity matrix (square).

Subroutine TEK81(X,EK,EC)
Called by: ASMKCR
Function: Computes the conduction and capacity matrices for K81 elements.
X = element coordinates.
EK = element conduction matrix (square).
EC = element capacity matrix (square).

Subroutine TEMT21(A,XL,EK,EC,ET,MP,PROP,TR)
Called by: ASMKCR
Calls: FPROP, ALPHA
Function: Computes the conduction, mass-transport, and capacity matrices for MT21 elements.
A = element area.
XL = element length.
EK = element K matrix (square).
EC = element capacity matrix (square).
ET = element temperature vector.
MP = see LDMP.
PROP = see LDMP.
TR = element mass-transport rate.

Subroutine TEMT42(SECT,XL,EK,EC,ET,MP,PROP,TR)
Called by: ASMKCR
Calls: TEC42, MPROP, ALPHA
Function: Computes the conduction, convective-exchange, mass-transport, and capacity matrices for MT42 elements.
SECT = element section properties.
XL = element length.
EK = element K matrix (square).
EC = element capcity matrix (square).
ET = element temperature vector.
MP = see LDMP.
PROP = see LDMP.
TR = element mass-transport rate.

```
Subroutine TEMT62(SECT,X,EK,EC,ET,MP,PROP,TR)
Called by: ASMKCR
Calls:     TEC62, CFFC, MPROP, TEK41, ALPHA
Function:  Computes the conduction, convective-exchange, mass-transport, and
           capacity matrices for MT62 elements.
SECT = element section properties.
X    = element coordinates.
EK   = element K matrix (square).
EC   = element capacity matrix (square).
ET   = element temperature vector.
MP   = see LDMP.
PROP = see LDMP.
TR   = element mass-transport rate.
```

```
Subroutine SAVESM(A,NSM)
Called by: EXTRTA
Function:  Stores capacity and indicated K matrices on mass storage.  Computes
           mass storage addresses for excitation and temperature derivative
           vectors.
Common:    CSAVE1/ NUCS(7),NZCS(7),LRCS(7)
                   NUCS(N) = library unit.
                   NZCS(N) = vector address.
                   LRCS(N) = record length.
                   where N = 1, capacity matrix.
                           = 2, diagonal convection matrix.
                           = 3, diagonal radiation matrix.
                           = 4, source load vector.
                           = 5, convection load vector.
                           = 6, radiaition load vector.
                           = 7, temperature derivative vector.
A   = data space buffer.
NSM = 0, first call to SAVESM - set up addresses and record lengths.
    > 0, subsequent calls to SAVESM.
```

```
Subroutine INVCAP(NPAT,ID,CI,MXNDF)
Called by: EXTRTA
Function:  Inverts the capacity matrix.
           Zeroes out terms corresponding to unattached node points, applied
           temperatures, and arithmetic nodes.
Common:    EKCOM4/ NCAP,ZCAP
                   NCAP = number of singular capacity terms to print.
                   ZCAP = singular capacity measure.
NPAT  = vector of applied temperature and arithmetic node points.
ID(N) = number of active freedoms at node N.
CI    = capacity matrix on entry, inverted capacity matrix on exit.
MXNDF = number of freedoms at each node point.
```

Subroutine DTCOMP(T,AKD,AK,CI,CK,RK,JT)
Called by: EXTRTA
Calls: NEXTKB, JTCOMP, DTK3
Function: Computes LAMBDA-MAX, etc. when PREP=1. Computes the integration time step size for the explicit method.

T = temperature vector.
AKD = diagonal term vector.
AK = element K matrix buffer.
CI = inverted capacity matrix.
CK = diagonal convection matrix.
RK = diagonal radiation matrix.
JT = element degree of freedom vector (from AK).

Subroutine DTK3(A,AK,LM)
Called by: DTCOMP
Function: Adds the diagonal terms in an element K matrix to the diagonal term vector (AKD in DTCOMP).

A = diagonal term vector.
AK = element K matrix.
LM = element degree of freedom vector.

Subroutine FACKAN(NPAN,T,LC,A1,A2,RK,RKAN)
Called by: EXTRTA
Calls: FACTOR
Function: Controls factoring of the arithmetic node K matrix. Computes the arithmetic node radiation matrix.

NPAN = arithmetic node point vector.
T = temperature vector.
LC = arithmetic node skyline vector.
A1,A2 = arithmetic node K matrix blocks.
RK = radiation matrix.
RKAN = arithmetic node radiation matrix.

```
Subroutine TTLV(KL,MP,PROP,TSCR,NPAT,T,EQ,ED,NP,LM,ET,Q,RB,JS,PHI,NERR)
Called by: EXTRTA
Calls:     LTPT, CSCQ, WRRDS, REVEC
Function:  Computes the thermal load and load rate vectors.
KL   = see LDKL.
MP   = see LDMP.
PROP = see LDMP.
TSCR = see T in TIMINC.
NPAT = vector of nodes at which temperatures are prescribed.
T    = temperature vector.
EQ   = element thermal load and load rate vector.
ED   = element data buffer.
NP   = element node point vector.
LM   = element degree of freedom vector.
ET   = element temperature vector.
Q    = element unit load vector.
RB   = radiation load vector excess data buffer (see WRRDS).
JS   = joint elimination sequence vector.
PHI  = load and load rate vector.
NERR = 0, normal exit.
     > 0, error exit.
```

```
Subroutine CSCQ(KL,MP,PROP,T,EQ,ED,NP,LM,ET,Q,JS,PHI,XLI,R,KQT)
Called by: TTLV
Calls:     CFPNI, CEDOF, ELTEMP, MPROP, QK21, QK31, QK41, QK61, QK81, EC21,
           EC31, EC41, CFMDM2, ADDVEC, EEPROP, EXPSCQ
Function:  Computes the source or convection load and load rate vector.
KL   = see LDKL.
MP   = see LDMP.
PROP = see LDMP.
T    = temperature vector.
EQ   = element group excitation vector buffer.
ED   = element data buffer:
       material number in ED(I1),
       node points in ED(I2),
       section properties in ED(I3),
       coordinates in ED(I4),
       last material number in ED(I5).
NP   = element node point vector.
LM   = element degree of freedom vector.
ET   = element temperature vector.
Q    = element unit load vector.
JS   = joint elimination sequence vector.
PHI  = source or convection load and load rate vector.
XLI  = ratio for computing load rate vector.
R    = ratio for computing load vector at T1 (see TTLV).
KQT  = 1, source load vector.
       2, convection load vector.
```

```
Subroutine CCDT
Called by: EXTRTA
Function:  Computes the integration time step size.
```

```
Subroutine FACKIM(NPAT,T,ID,LC,A1,A2,C,JS,RK,RKIM)
Called by: EXTRTA
Calls:     FACTOR
Function:  Controls factoring of implicit method K matrix.
           Computes initial radiation matrix.
NPAT = prescribed temperature node points.
T    = temperature vector.
ID   = active freedoms vector.
LC   = skyline vector.
A1   = K matrix buffer.
A2   = vector of modified diagonal terms (applied temperatures).
       K matrix buffer during factoring.
C    = capacity matrix.
JS   = joint elimination sequence vector.
RK   = radiation vector.
RKIM = initial radiation matrix.
```

```
Subroutine TEMPAN(NPAN,T,LT,LCAN,AKAN,AK,RK,Q,QAN,RKAN,JT,NERR)
Called by: EXTRTA
Calls:     NEXTKB, JTCOMP, MPROD3, MTPROD, CSPDP, XSOL, CDPSP, MODKAN
Function:  Computes temperatures at arithmetic nodes after computation of
           thermal load vectors.
NPAN = arithmetic node vector.
T    = temperature vector.
LT   = length of T.
LCAN = arithmetic node skyline vector.
AKAN = arithmetic node K matrix block.
AK   = element K matrix block.
RK   = radiation matrix.
Q    = source-convection load vector.
QAN  = arithmetic node load vector.
RKAN = arithmetic node radiation matrix.
JT   = element degree of freedom vector (from AK).
NERR = 0, normal return.
     > 0, error return.
```

```
Subroutine MODKAN(NPAN,T,LC,KE,A1,A2,RK,RKAN)
Called by: TEMPAN, TTDSAN
Calls:     FACTOR
Function:  Modifies and factors the arithmetic node K matrix when the maximum
           number of iterations is reached in TEMPAN or TTDSAN.
NPAN = arithmetic nodes list.
T    = temperature vector.
LC   = skyline vector.
KE   = last equation in a block vector.
A1   = K matrix buffer.
A2   = K matrix buffer.
RK   = radiation matrix.
RKAN = arithmetic node radiation matrix.
```

```
Subroutine SAVET(TIME,T,CTIME,STIME,NSAVE)
Called by: EXTRTA
Function:  Stores temperatures at times specified with the TSAVE command.
TIME  = save time vector (TRTA TIME data set).
T     = temperature vector.
CTIME = current time.
STIME = time at which to save results.
NSAVE = number of temperature vectors stored.
```

```
Subroutine TTDS(NPAT,NPAN,T,LT,LCAN,AKAN,AK,CI,CK,RK,Q,QAN,RKAN,JT,NERR)
Called by: EXTRTA
Calls:     MPROD2, VSUM2, NEXTKB, JTCOMP, MPROD3, MTPROD, RTDT, MPROD1,
           TTDSAN, TTDSIM, VSUM1
Function:  Computes the temperature vector at time t(i+1).
NPAT = vector of node points at which temperatures are prescribed.
NPAN = vector of arithmetic node points.
T    = temperature and temperature time derivative vectors.
LT   = length of T.
LCAN = arithmetic node or implicit method skyline vector.
AKAN = arithmetic node or implicit method K matrix block.
AK   = element K matrix block (explicit method).
CI   = inverted diagonal capacity matrix (explicit method).
       capacity matrix (implicit method).
CK   = diagonal convection matrix (explicit method).
       joint elimination sequence vector (implicit method).
RK   = diagonal radiation matrix.
Q    = load and load rate vector.
QAN  = arithmetic node load vector (explicit method only).
RKAN = arithmetic node or implicit method radiation matrix.
JT   = element degree of freedom vector.
NERR = 0, normal return.
     > 0, error return.
```

Subroutine MTPROD(AKM,TM,T,JT,KDF)
Called by: TTDS
Function: The product of square matrix AKM and vector TM is subtracted from the vector T. The result is stored in T.
JT = AKM matrix degree of freedom vector.
KDF = dimension of AKM and JT.

Subroutine RTDT(T,LT,RK,NM)
Called by: TTDS
Function: Computes the radiation contributions to the temperature time derivative vectors.
T = temperature and temperature time derivative vectors.
LT = length of T.
RK = diagonal radiation matrix.
NM = time derivative indicator: 1-compute first derivative, 2-compute second derivative, etc.

Subroutine TTDSAN(NPAN,T,LT,LCAN,AKAN,AK,RK,Q,QAN,RKAN,JT,NT,NERR)
Called by: TTDS
Calls: NEXTKB, JTCOMP, MPROD3, MTPROD, CSPDP, XSOL, CDPSP, MODKAN
Function: Computes temperature derivatives at arithmetic node points.
NPAN = arithmetic node point vector.
T = temperature and temperature derivative vectors.
LT = length of T.
LCAN = arithmetic node skyline vector.
AKAN = arithmetic node K matrix block.
AK = element K matrix block.
RK = radiation matrix.
Q = load and load rate vectors.
QAN = arithmetic node load vector.
RKAN = arithmetic node radiation matrix.
JT = element degree of freedom vector (from AK).
NT = derivative indicator: 1-temperature vector, 2-first derivative, etc.
NERR = 0, normal return.
> 0, error return.

```
Subroutine TTDSIM(NPAT,TO,T1,TBAR,NDF,LC,AK,ATF,CAP,JS,RK,Q,QD,RKIM,NERR)
Called by: TTDS
Calls:     CSPDP, XSOL, CDPSP, MODKIM
Function:  Computes the temperature vector for the implicit method.
NPAT = vector of applied temperature node points.
TO   = temperature vector at beginning of time step.
T1   = temperature vector at end of time step.
TBAR = BETA*T1 + (1.0-BETA)*TO.
NDF  = number of freedoms at each node point.
LC   = skyline vector.
AK   = K matrix buffer.
ATF  = applied temperature modified diagonal terms.
CAP  = capacity matrix.
JS   = joint elimination sequence vector.
RK   = radiation matrix.
Q    = load vector.
QD   = load rate vector.
RKIM = initial radiation matrix.
NERR = 0, normal return.
     > 0, error return.
```

```
Subroutine MODKIM(NPAT,T,ID,LC,KE,A1,A2,C,JS,RK,RKIM)
Called by: TTDSIM
Calls:     FACTOR
Function:  Modifies and factors implicit method K matrix.
           Computes initial radiation matrix.
NPAT = prescribed temperature node points.
T    = temperature vector.
ID   = active freedoms vector.
LC   = skyline vector.
KE   = last equation in block vector.
A1   = K matrix buffer.
A2   = vector of modified diagonal terms (applied temperatures).
       K matrix buffer during factoring.
C    = capacity matrix.
JS   = joint elimination sequence vector.
RK   = radiation vector.
RKIM = initial radiation matrix.
```

```
Subroutine TFLUX(KL,MP,PROP,TSCR,TIME,T,LX,ED,NP,ET,TA,FLUX)
Called by: EXTRTA
Calls:     LEXP, CMPTS, FTCOEF, LDTAF, LINES, CFPNI, ELTEMP, EFLUX, EEPROP,
           EXPFLX
Function:  Controls the computation of transient element heat fluxes.
KL    = see LDKL.
MP    = see LDMP.
PROP  = see LDMP.
TSCR  = see T in TIMINC.
TIME  = vector of times at which fluxes are to be computed.
T     = temperature vector.
LX    = adjusted length (LADJ) of T.
ED    = element data buffer:
          material numbers in ED(I1),
          node points in ED(I2),
          section properties in ED(I3),
          coordinates in ED(I4),
          last temperature points in ED(I5).
NP    = element node point vector.
ET    = element temperature vector.
TA    = convective-exchange temperatures and mass-transport rates.
FLUX  = fluxes.


Subroutine CMPTS(LOCEXP,MP,NLCS,MPTS)
Called by: TFLUX, TMTPD, TRFLUX
Function:  Determines if material or fluid properties are time-dependent.
LOCEXP = see KL(17,N) in LDKL.
MP     = see LDMP.
NLCS   = number of fluxes (corresponding to different times in TRTA TIME) that
           are computed simultaneously.
MPTS   = 0, properties are not time-dependent.
         1, properties are time-dependent.


Subroutine FTCOEF(LOCEXP,MP,PROP,TSCR,TIME,KFP)
Called by: TFLUX, TMTPD, TRFLUX
Calls:     TPROP
Function:  Controls computation of time-dependent properties.
LOCEXP = see KL(17,N) in LDKL.
MP     = see LDMP.
PROP   = see LDMP.
TSCR   = see T in TIMINC.
TIME   = time at which properties are to be computed.
KFP    = conduction property computation parameter for MT42 and MT62 elements.
       = 0, do not compute conduction properties.
       = 1, compute conduction properties.
```

```
Subroutine LDTAF(TSCR,TIME,TA,LTA,LT,NTA,KA,LDTYP)
Called by: TFLUX, TMTPD
Calls:     LOCPT
Function:  Computes element thermal excitation vector at time TIME by linear
           interpolation.
TSCR  = set T in TIMINC.
TIME  = time at which to compute excitation vector.
TA    = thermal excitation vector.
LTA   = adjusted length (LADJ) of TA.
LT    = length of TA.
NTA   = number of TA vectors.
KA    = sector address of first TA vector.
LDTYP = 2, convective-exchange temperatures.
      = 4, mass-transport rates.
```

```
Subroutine TMTPD(KL,MP,PROP,TSCR,TIME,T,LX,ED,NP,ET,TR,P)
Called by: EXTRTA
Calls:     CMPTS, FTCOEF, LDTAF, LINES, CFPNI, ELTEMP, EPD, EEPROP, EXPPD
Function:  Controls the computation of transient element fluid pressures.
KL    = see LDKL.
MP    = see LDMP.
PROP  = see LDMP.
TSCR  = see T in TIMINC.
TIME  = vector of times at which pressures are to be computed.
T     = temperature vector.
LX    = adjusted length (LADJ) of T.
ED    = element data buffer:
        material numbers in ED(I1),
        node points in ED(I2),
        section properties in ED(I3),
        coordinates in ED(I4),
        last temperature points in ED(I5).
NP    = element node point vector.
ET    = element temperature vector.
TR    = mass-transport rates.
P     = element pressures.
```

```
Subroutine TRFLUX(KL,MP,PROP,TSCR,TIME,A)
Called by: EXTRTA
Calls:     CMPTS, FTCOEF, RFLUX
Function:  Controls the computation of transient radiation-exchange fluxes.
KL    = see LDKL.
MP    = see LDMP.
PROP  = see LDMP.
TSCR  = see T in TIMINC.
TIME  = vector of times at which fluxes are to be computed.
A     = working data space.
```

7.3 Data Sets Generated by TRTA

The following data sets are stored on the unit specified by the RESET control.

RESET	Default	Data Set Name	Contents
KLIB	21	EK TRTA 0 0	Element K matrix string.
NUCAP	0	CAP TRTA 0 0	Capacity matrix.
NUKC	0	KC TRTA 0 0	Diagonal convection matrix.
NUKR	0	KR TRTA 0 0	Diagonal radiation matrix.
NUQS	21	QS TRTA 0 0	Source load and load rate vector.
NUQC	21	QC TRTA 0 0	Convection load and load rate vector.
NUQR	21	QR TRTA 0 0	Radiation load vector.

The following radiation exchange data set is stored on the unit specified by RESET control RGLIB. The TOC parameters are NI=2+maximum number of nodes in any radiation element and NJ=the number of radiation elements.

REX GEOM 0 0: area,material,list of nodes

The following radiation exchange data set is stored on the unit specified by RESET control RGLIB when RESET control RCC>2. The TOC parameters are NI=1 and NJ=LRT.

REX VEC 0 0: nodal radiation load vector

The following radiation exchange data sets are stored on the unit specified by RESET control RGLIB and RESET control RCC>3. The TOC parameters are NI=1 and NJ=the number of radiation elements.

REX AB 0 0: element absorptivity
REX Q 0 0: element emitted heat
REX QA 0 0: element absorbed heat

The following arithmetic node or implicit method K matrix data sets are stored on the unit denoted by RESET control KLIB.

KF SKY nrms mxbw: factored symmetric K matrix
KFA SKY nrms mxbw: factored asymmetric K matrix

The following arithmetic node or implicit method K matrix data sets are stored on the unit specified by execution command kSAVE.

K SKY nrms mxbw: assembled symmetric K matrix
K POIN nrms mxbw: skyline vector of K SKY
KA SKY nrms mxbw: assembled asymmetric K matrix
KA POIN nrms mxbw: skyline vector of KA SKY

7.4 RESET Controls and Execution Commands

The following RESET controls and execution command are referenced in LDTRTA but are not listed in Volume 1 of the SPAR Thermal Analysis Reference Manual.

RESET Controls:

KLIB = K matrix library.
SET = load set identifier.
LEK = element K matrix block length (explicit method only).
NUCAP = see Section 7.3.
NUKC = see Section 7.3.
NUKR = see Section 7.3.
NUQS = see Section 7.3.
NUQC = see Section 7.3.
NUQR = see Section 7.3.
NUTD = see Section 7.3.
DTFAC = see labeled common block CTIME3.
ZFIN = see description of function FEF62.
RGLIB = see Section 7.3 - radiation exchange data sets.
MPTI = see labeled common MDATA3.
MPRT = see labeled common MDATA3.
PITER = see NPITS in labeled common block PDATA2.
PCONV = see labeled common block PDATA2.
LZERO = see ZEROL in labeled common MTCOM1.
AZERO = see ZEROA in labeled common MTCOM1.
RCC = see Section 7.3 - radiation exchange data sets.

Arithmetic node or implicit method RESET controls:

LK = arithmetic node K matrix block length. If the length is not given, it is computed in subroutine CSLVAN or CSLVIM.
KBWR = same as for SSTA, plus the following:
= 1, print K matrices for elements with arithmetic nodes.
= 2, same as 1 above, except stop after printing K matrices.
ANPRT = 0,1 - see Volume 1.
= 2, print the arithmetic node load vector.
RKM = radiation matrix initial approximation factor, see MODKAN and MODKIM.
ATFAC = factor which multiplies the diagonal implicit method K matrix terms corresponding to nodes at which temperatures are prescribed.

Execution command:

KSAVE=lib cause the assembled arithmetic node or implicit method K matrix to be saved on unit lib. If lib is not given, K is saved on the unit specified by RESET control KLIB.