Building Your Second SINDA Model

Steven L. Rickman
NASA Technical Fellow for Passive Thermal
NASA Engineering and Safety Center

Disclaimer

The Systems Improved Numerical Differencing Analyzer (SINDA) computer program or its forebears has been a mainstay of thermal analysis for more than 50 years.

SINDA is offered by a number of different vendors and syntax and features may vary from product to product.

For this lesson, SINDA/FLUINT by Cullimore and Ring Technologies (C&R Technologies®), Inc. is used. However, similar concepts apply to other versions of SINDA. *The use of this product in this lesson should not be construed as an endorsement of one product or another.*

Data used in this lesson are for demonstration purposes and should not be used for design purposes or to replace use of any project-directed data.

Prerequisite

Viewers watching this lesson are assumed to have some familiarity with thermal network modeling and the SINDA input format.

To fully understand the content of this lesson, watching the NESC Academy lesson entitled *Building Your First SINDA Model* is recommended.

Introduction

Engineers rely on a wide variety of modern thermal tools to model thermal problems.

Many of these tools offer graphical “front ends” whereby users formulate their analytical models using a CAD interface.

Behind the scenes, the analysis is performed on a thermal network – this is true whether the analyst uses finite differencing or finite element methodologies.

While graphical front ends are very powerful, understanding the resulting thermal network representation used for a model gives the user the ability to check or even modify a model at a basic level.

Introduction

Additionally, there are instances where an engineer may prefer to develop network models from scratch or use heritage code that does not have a graphical front end.

Some front-end programs output thermal networks in the widely used SINDA format.

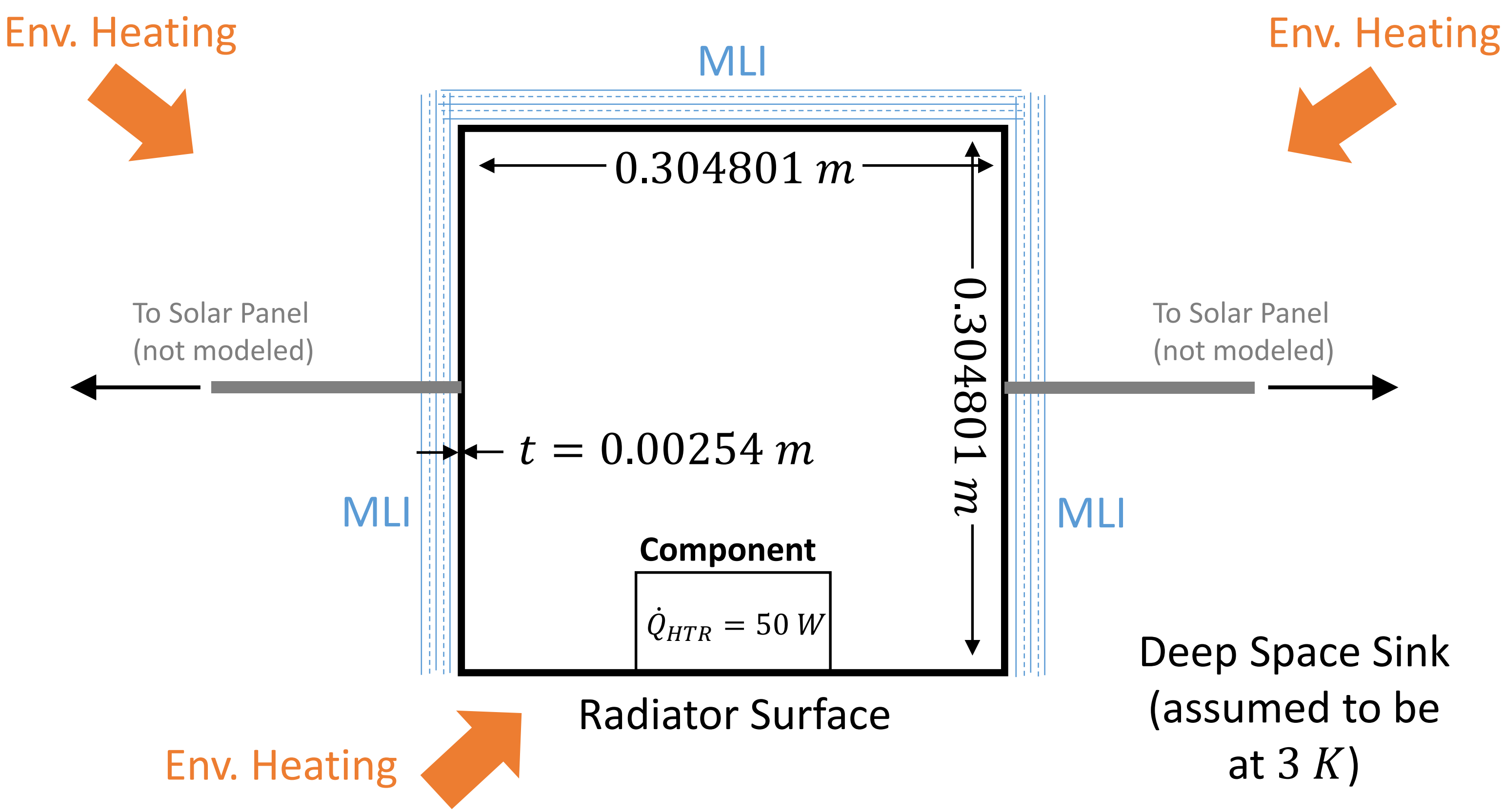
Our lesson will focus on this input format.

Lesson Scope

- Modeling temperature dependent properties;
- Modeling time dependent heating sources;
- Using arrays;
- Using arithmetic nodes;
- Using Variables 1 logic;
- Customizing output.

Problem Statement (1 of 2)

- Consider the simplified spacecraft shown below.



Problem Statement (2 of 2)

- For a 408 km circular orbit about the Earth at $\beta = 0°$ degrees in a fixed, local vertical - local horizontal attitude with one side constantly facing the planet below (i.e., nadir)…

- *Calculate the cyclic steady state temperature profile of each box side, each insulation side, and the component within the box.*

- Assume that the component temperature is maintained between 30 °C and 35 °C using a 50 W thermostatically controlled heater.

Key Dimensions and Properties

Parameter	Value
Box Length	$0.304801\ m\ (1\ ft)$
Box Width	$0.304801\ m\ (1\ ft)$
Box Height	$0.304801\ m\ (1\ ft)$
Box Wall Thickness	$0.00254\ m\ (0.1\ in)$
Box Material (Aluminum 6061 T6)	
Density	$2705.0\ kg/m^3$
Specific Heat	$c_p(T)$
Thermal Conductivity	$k(T)$
Component	
Mass	$0.900\ kg$
Specific Heat	$910\ J/kg \cdot ℃$
Contact Area	$0.010323\ m^2$
Contact Conductance*	$2000\ W/m^2 \cdot ℃$

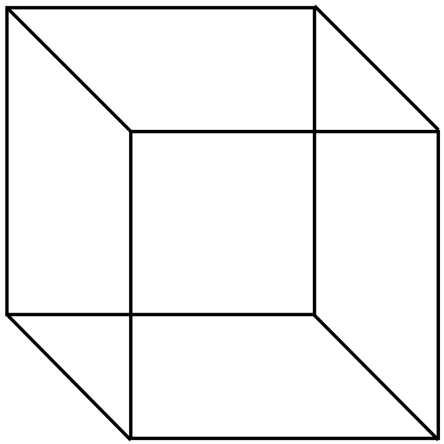
*Representative value from: http://www.thermopedia.com/content/1188

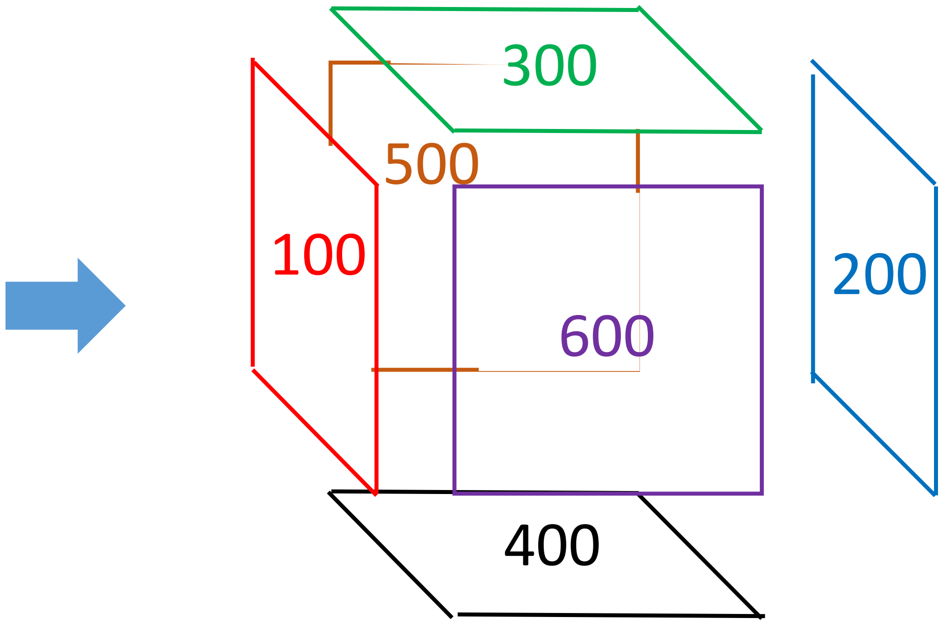
Thermo-Optical Properties

Surface (Use)	Solar Absorptance (α)	Infrared Emittance (ε) or Effective Emissivity (ε^*)
White Paint (Radiator External Surface)	0.27	0.88
Aluminized Polyimide (MLI External Surface)	0.41	0.75
MLI Effective Emissivity	N/A	0.05

Forming the Nodal Network

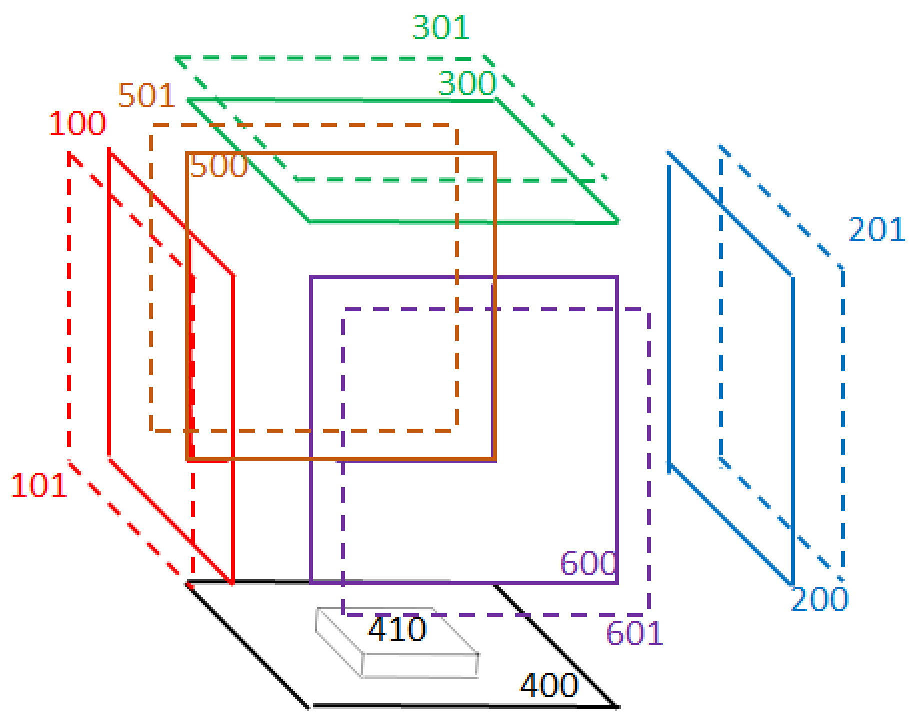
For a finite difference discretization, we'll first model the box as six separate sides for which individual temperatures will be calculated.





Forming the Nodal Network

In addition to the box structure, we'll need to model MLI on five of the six sides plus we'll also need to add a component that is affixed to the exposed radiator surface.



Review of Node Types

There are three* primary types of nodes used in thermal network models:

Diffusion Nodes – represent a finite thermal capacitance – when heat flows into or out of a diffusion node, the temperature changes gradually;

Arithmetic Nodes – represent a “massless” object – when heat flows into or out of an arithmetic node, the temperature changes instantly;

Boundary Nodes – represent an “infinite” thermal capacitance – they are used as heat sources or heat sinks.

*Heater Nodes are a special case of Boundary Nodes and will not be discussed in this lesson.

Review of Conductor Types

In a thermal network model, heat flows from one node to another along pathways called conductors – there are two types:

Linear Conductors -- heat flow for conduction and convection is linear – that is, it is based on the ΔT between the two objects of interest;

Radiation Conductors -- heat flow between two objects is highly non-linear – that is, it is a function of the $\Delta(T^4)$ between the two objects where the temperature is expressed in absolute units.

Assumptions in Forming the Nodal Network (1 of 2)

The box sides and the component are represented by diffusion nodes.

MLI sides are represented by arithmetic nodes.

Space is a boundary node.

The edges that are common to any two sides will indicate a heat transfer path (linear conductor) between those two sides is necessary.

Contact conduction between the component and the radiator is also modeled with a linear conductor.

Each MLI surface and the radiator radiates to deep space (using radiation conductors).

Assumptions in Forming the Nodal Network (2 of 2)

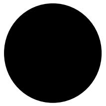
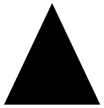
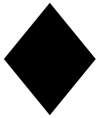
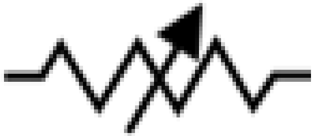
Heat transfer through the MLI is of interest and will be modeled with radiation conductors. Lateral heat transfer in the MLI from one box side to another is negligible so this heat transfer will not be modeled.

No internal radiation is considered.

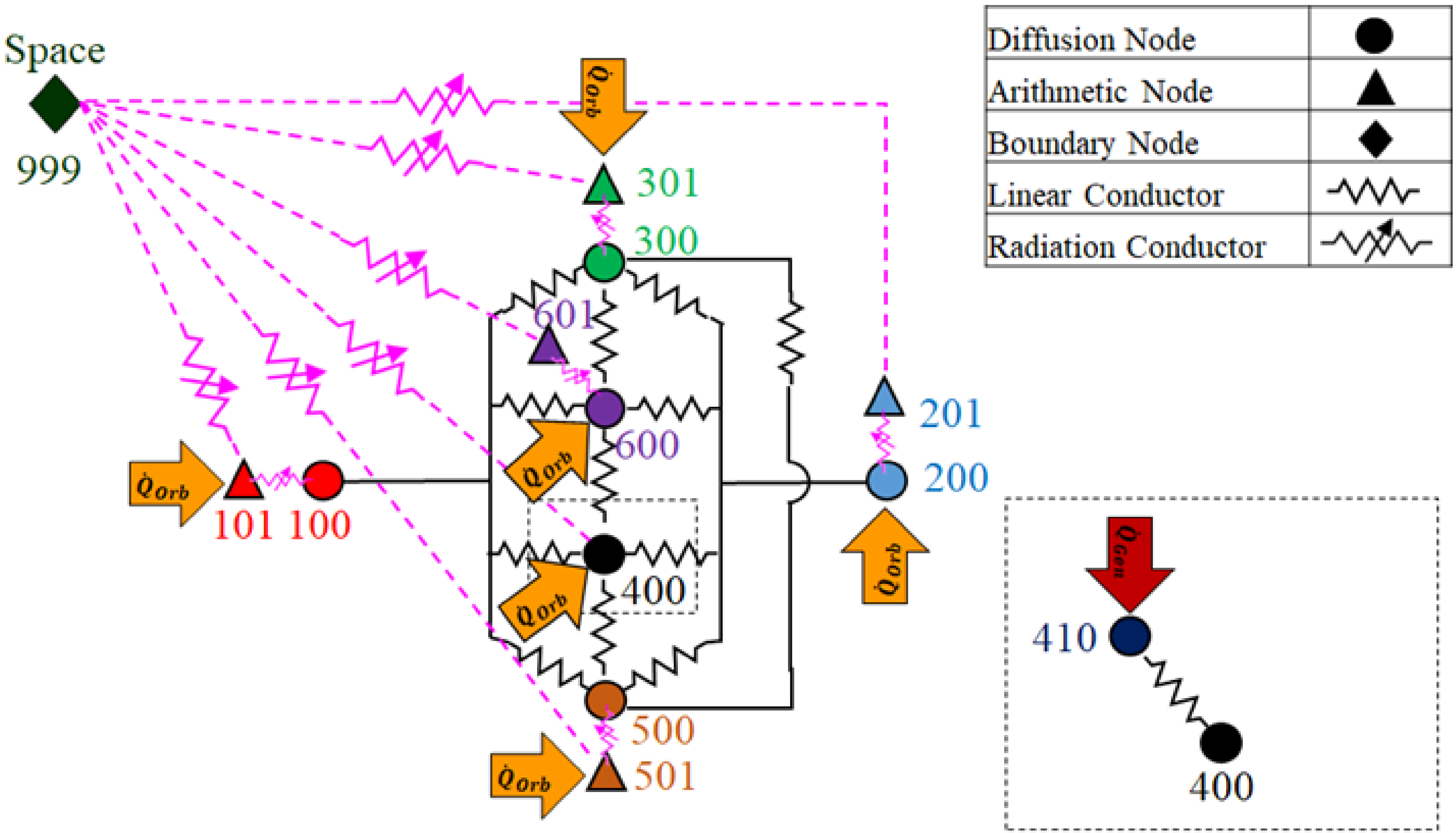
External environmental heating (calculated elsewhere) is applied to each MLI surface and to the exposed radiator surface.

Power from the component heater is applied directly to the node representing the component.

Symbols Used for the Thermal Network in this Example

Diffusion Node	
Arithmetic Node	
Boundary Node	
Linear Conductor	
Radiation Conductor	

Forming the Nodal Network



Thermal Capacitance Calculations

For each diffusion node, we calculate the thermal capacitance, C as follows:

$$C = \rho \cdot l \cdot w \cdot t \cdot c_p(T)$$

Where…

l	is the box side length
w	is the box side width
t	is the box side thickness
ρ	is the aluminum density
$c_p(T)$	is the aluminum specific heat (a function of temperature for this model)

Thermal Capacitance Calculations

For our model, all box sides are the same size so the capacitance becomes:

$$C = \rho \cdot c_p(T) \cdot l \cdot w \cdot t$$

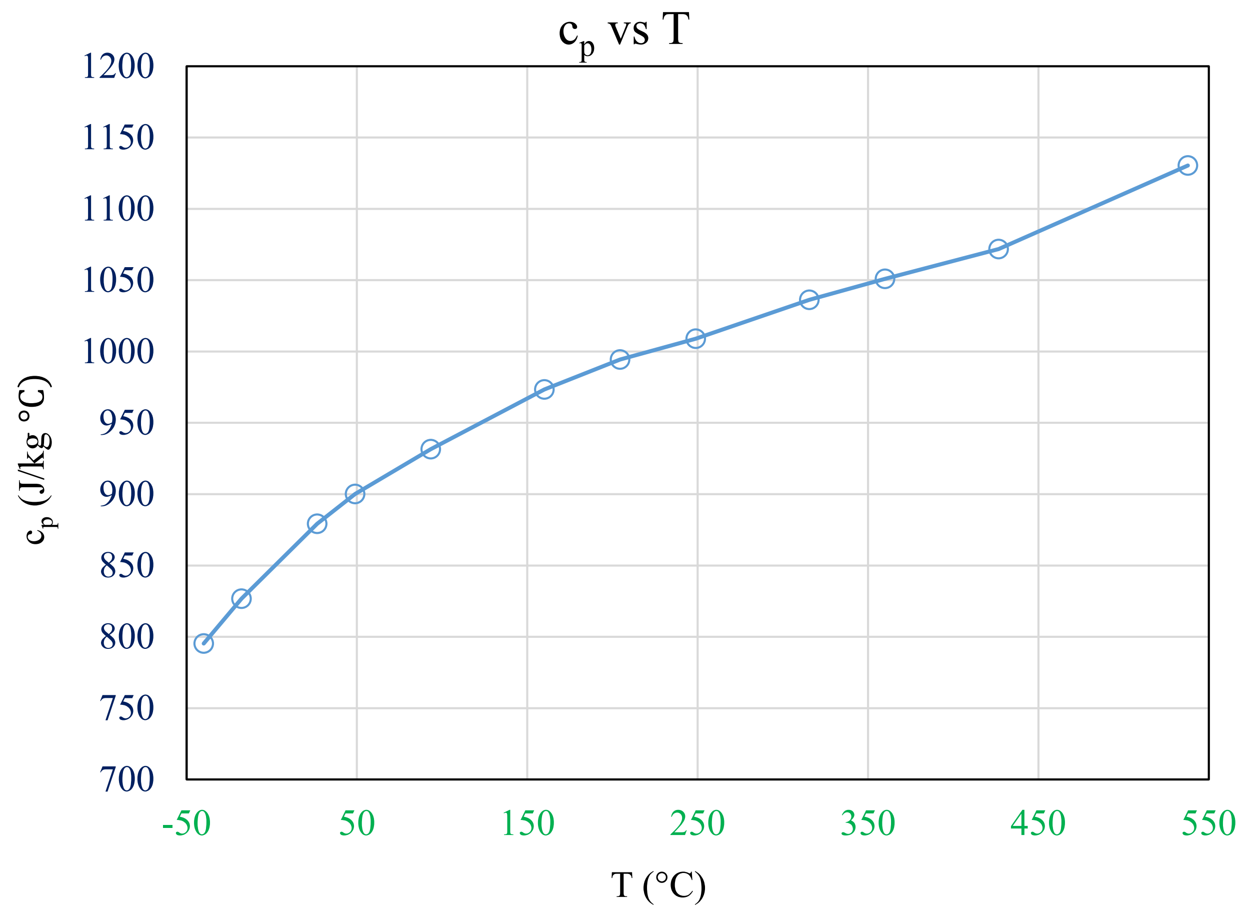
$$C = (2705.\, kg/m^3) \cdot (0.304801\ m) \cdot (0.304801\ m) \cdot (0.00254\ m) \cdot c_p(T)$$

$$C = 0.638313 \cdot c_p(T)$$

Specific Heat vs. Temperature

- The aluminum specific heat as a function of temperature is:

T (°C)	c_p $(J/kg \cdot °C)$
-40.0	795.3
-17.8	826.7
26.7	879.2
48.9	900.2
93.3	931.6
160.0	973.4
204.4	994.4
248.9	1009.0
315.6	1036.2
360.0	1050.9
426.7	1071.8
537.8	1130.4



Thermal Capacitance Calculations

For the component, we are assuming a constant capacitance:

$$C = m \cdot c_p$$

Where…

m is the component mass

c_p is the constant specific heat

The thermal capacitance is:

$$C = m \cdot c_p = 0.900\ kg\ \cdot 910\ J/kg \cdot {}^{\circ}\mathrm{C} = 819\ J/{}^{\circ}\mathrm{C}$$

Thermal Conductance Calculations

For each node, we calculate the thermal conductance, G as follows:

$$G = k(T) \cdot w \cdot t/l$$

Where…

l	is the box side length
w	is the box side width
t	is the box side thickness
$k(T)$	is the aluminum thermal conductivity (a function of temperature for this model)

Thermal Conductance Calculations

For our model, the thermal conductance between all adjacent box nodes is the same, namely:

$$G = k(T) \cdot w \cdot t/l$$

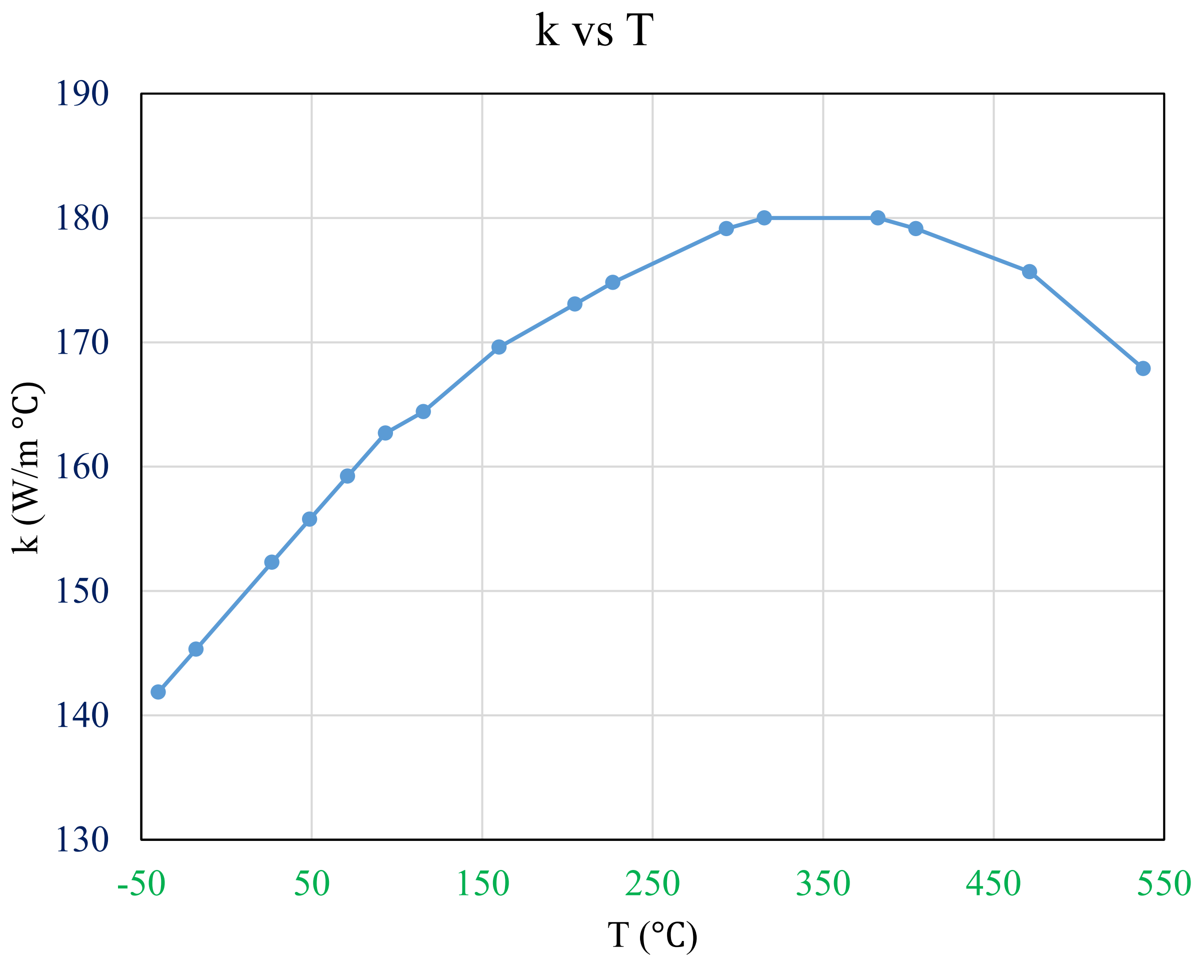
$$G = k(T) \cdot (0.304801\, m) \cdot (0.00254\, m)/(0.304801\, m)$$

$$G = 0.00254 \cdot k(T)$$

Thermal Conductivity vs. Temperature

- For aluminum thermal conductivity as a function of temperature:

T (°C)	k $(W/m \cdot °C)$
-40.0	141.9
-17.8	145.3
26.7	152.3
48.9	155.8
71.1	159.2
93.3	162.7
115.6	164.4
160.0	169.6
204.4	173.1
226.7	174.8
293.3	179.1
315.6	180.0
382.2	180.0
404.4	179.1
471.1	175.7
537.8	167.9



Contact Conductance Calculation

The component in our model is mounted to the radiator surface and we will express the heat transfer in terms of a contact conductance, $G_{contact}$ which takes the form:

$$G_{contact} = h_{contact} \cdot A$$

Where $h_{contact}$ is the contact conductance per unit area and A is the contact area. From our model data:

$$G_{contact} = (2000\ W/m^2 \cdot ℃) \cdot (0.010323\, m^2)$$

$$G = 20.646\ W/℃$$

Radiation Conductance Calculations

We must also calculate the radiation conductance, G_{rad} to account for the radiative heat loss from each box side to deep space:

$$G_{rad} = \varepsilon \cdot B_{ij} \cdot \sigma \cdot l \cdot w$$

Where…

ε is the surface infrared emissivity

B_{ij} interchange factor (assumed to be unity here)

σ is the Stefan-Boltzmann constant ($5.67 \times 10^{-8}\ W/m^2 \cdot K^4$)

l is the box side length

w is the box side width

Radiation Conductance Calculations

Five of the six box sides have an aluminized polyimide external surface with the same radiation conductance to space, namely:

$$G_{rad} = \varepsilon \cdot \sigma \cdot l \cdot w$$
$$G_{rad} = (0.75) \cdot \sigma \cdot (0.304801\ m) \cdot (0.304801\ m)$$
$$G_{rad} = \sigma \cdot 0.069678$$

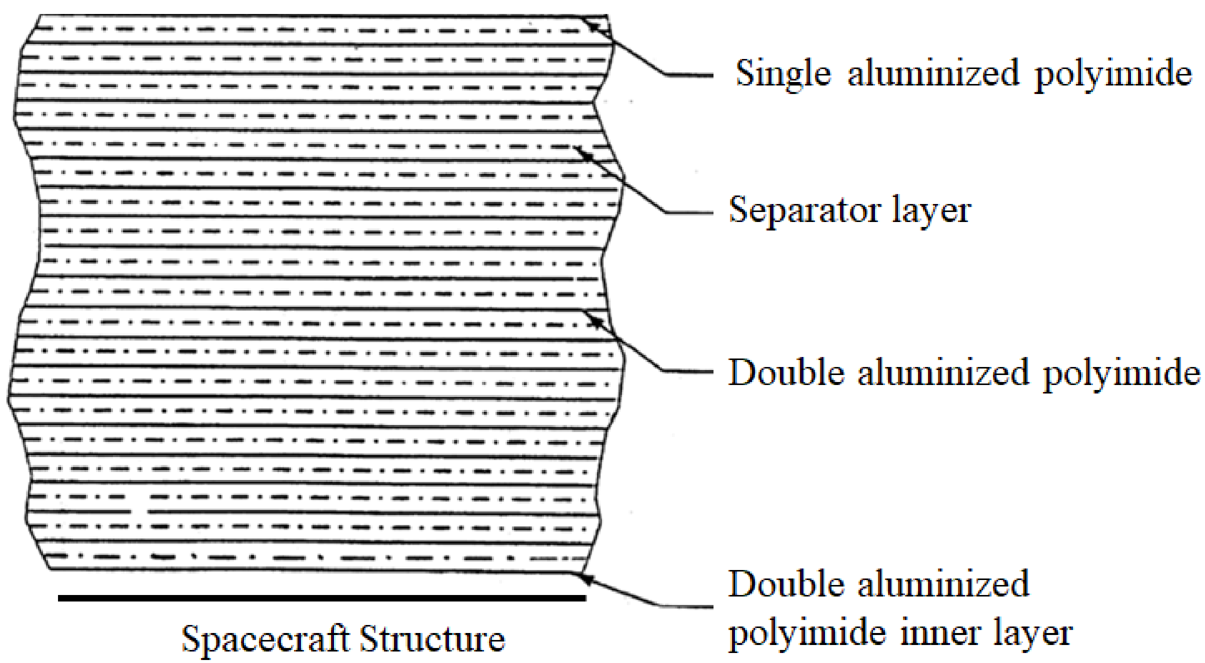
The uninsulated box side (i.e., the radiator) surface has a white paint :

$$G_{rad} = \varepsilon \cdot \sigma \cdot l \cdot w$$
$$G_{rad} = (0.88) \cdot \sigma \cdot (0.304801\ m) \cdot (0.304801\ m)$$
$$G_{rad} = \sigma \cdot 0.081755$$

We will set a variable within SINDA to automatically multiply all radiation conductors by σ.

A Simple MLI Model

Multilayer insulation, or MLI, is a very compact, mass efficient thermal insulation for use in space vacuum



Notional MLI Blanket Cross-Section

A Simple MLI Model

MLI is easily represented as a thermal network by assuming:

Heat transfer through the MLI is via radiation only;

The radiative heat transfer ($\dot{Q}_{MLI}$) for an area (A) is expressed via an effective emissivity (ε^*) through the MLI and the temperature difference from the external surface $\left(T_{surf}\right)$ to the underlying structure (T_{struc}):

$$\dot{Q}_{MLI} = A \cdot \sigma \cdot \varepsilon^* \cdot \left(T_{surf}^4 - T_{struc}^4\right)$$

Where the temperatures are expressed in absolute units (K).

A Simple MLI Model

We'll need to calculate the radiation conductor for heat transfer through the MLI which is based on the effective emissivity:

$$G_{rad} = \varepsilon^* \cdot \sigma \cdot l \cdot w$$

where…

ε^* is the MLI effective emissivity

σ is the Stefan-Boltzmann constant

l is the box side length

w is the box side width

A Simple MLI Model

Using the data provided:

$$G_{rad} = \varepsilon^* \cdot \sigma \cdot l \cdot w$$

$$G_{rad} = 0.05 \cdot \sigma \cdot 0.304801\, m \cdot 0.304801\, m$$

$$G_{rad} = 0.004645 \cdot \sigma$$

Again, we will use a SINDA variable to multiply all radiation conductors by σ.

A Simple MLI Model

MLI can be thought of as very low mass with high heat transfer area and often undergoes a rapid temperature changes when the thermal environment changes;

Since the MLI typically reacts much faster than other components, it is often modeled as having zero mass;

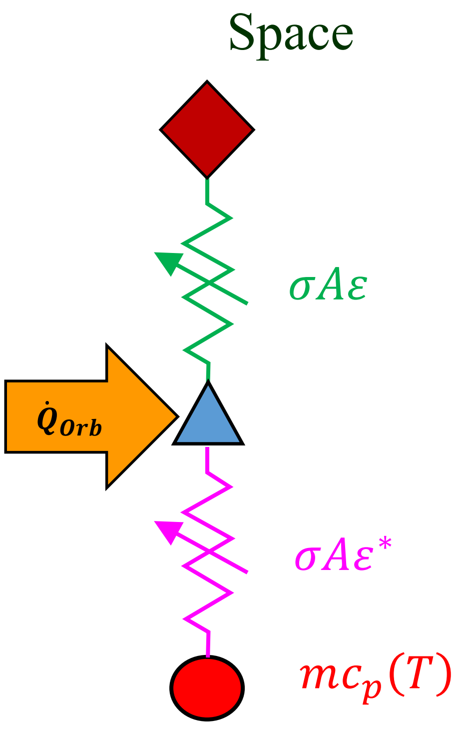
In SINDA, this is modeled by an arithmetic node with the format:

```
N#, Tinit, Any Negative Real Number
```

Where N# and Tinit are the node number and the initial temperature, respectively.

A Simple MLI Model

The simplified representation of MLI, then, is a single arithmetic node and a radiation conductor based on ε^*, which is connected to the underlying structure (which is a diffusion node).



The MLI surface radiates to deep space (with surface emissivity, ε (and *not* ε^*) and can also receive environmental heating.

Using SIV Statements

For our first SINDA model, we assumed that the c_p and k were constant;

For many materials, c_p and/or k vary as a function of temperature;

SINDA provides an input format for, both, nodes and conductors that allows temperature varying c_p and k, respectively;

This is accomplished using the SIV format.

SIV Format for Nodes

For this model, we are allowing the aluminum specific heat to vary as a function of temperature;

To specify a node with a temperature varying capacitance, we use the SIV format:

```
SIV N#, Tinit, A#, Mass
```

Where N# is the node number, Tinit is the initial temperature, Mass is the nodal mass, and A# is an array number for a SINDA doublet array representing the material specific heat as a function of temperature – more about arrays shortly.

SIV Format for Conductors

SIV may also be used to specify a temperature dependent thermal conductivity:

```
SIV #G, N#1, N#2, A#, A/L
```

Where G# is the conductor number, N#1 and N#2 are the node numbers of the nodes connected by the conductor, A/L is the area to length ratio for the conductor, and A# is the SINDA doublet array representing the thermal conductivity as a function of temperature.

The temperature used for thermal conductivity look up is the arithmetic average of the two nodes connected by the conductor.

Modeling Time Varying Heat Sources

For this model, we'll need to subject the spacecraft body to time varying orbital heating;

This heating can be calculated using a variety of available tools.

Additional information on simplified orbital heating calculations may be found in the NESC Academy lesson *Introduction to On-Orbit Thermal Environments (Parts 1 – 4)*.

Modeling Time Varying Heat Sources

For each box side, heating flux is calculated for the solar ($\dot{q}_{solar}(t)$), albedo ($\dot{q}_{albedo}(t)$), and planetary infrared ($\dot{q}_{IR}(t)$) components as a function of time, t;

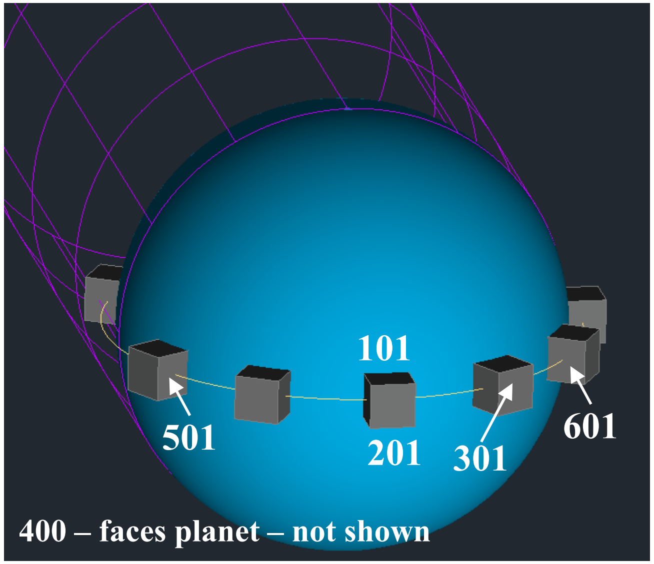
The solar and albedo components are multiplied by a given box side's solar absorptivity (α) and the planetary infrared component is multiplied by the infrared emissivity (ε);

Finally, the components are summed and multiplied by the box side area (A) to obtain the total heating ($\dot{Q}_{total}$):

$$\dot{Q}_{total}(t) = A \cdot [\alpha \cdot \dot{q}_{solar}(t) + \alpha \cdot \dot{q}_{albedo}(t) + \varepsilon \cdot \dot{q}_{IR}(t)]$$

Modeling Time Varying Heat Sources

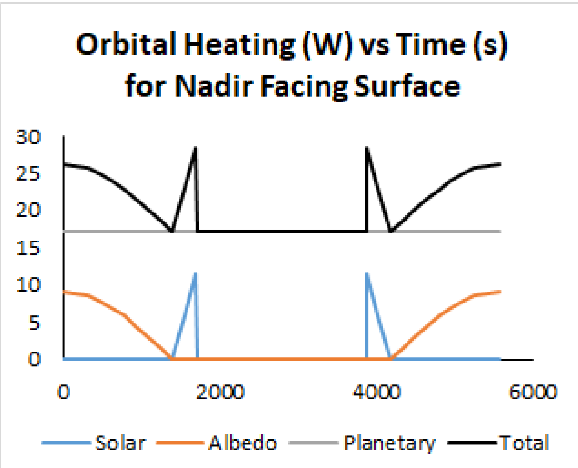
Surface	Direction Facing
101	North
201	South
301	Zenith
400	Nadir
501	Forward
601	Aft

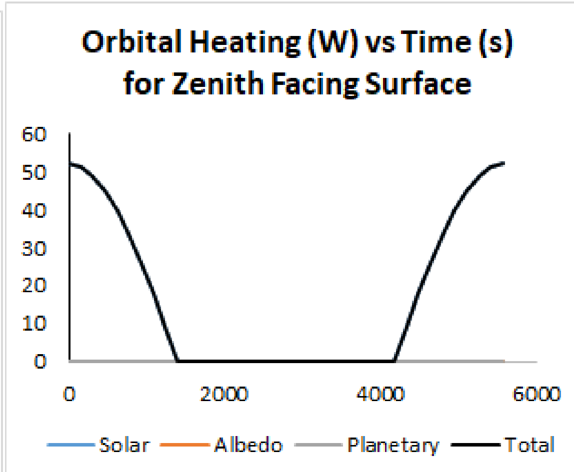


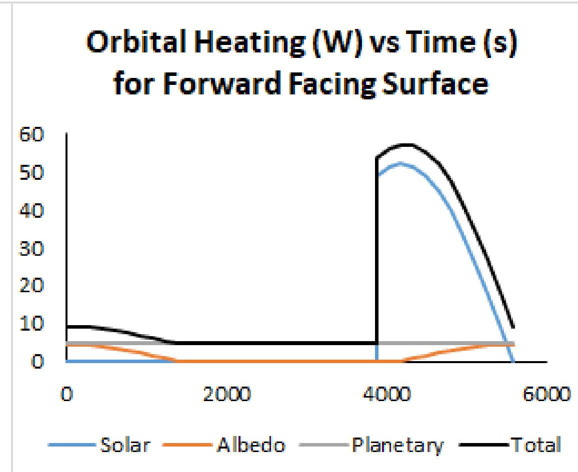
Orbit/Heating Parameter	Value
Altitude (Circular)	$408\ km$
Beta Angle	$0°$
Solar Flux ($\dot{q}_{solar}$)	$1367\ W/m^2$
Albedo Factor	0.3
Planetary Infrared Flux ($\dot{q}_{IR}$)	$236\ W/m^2$

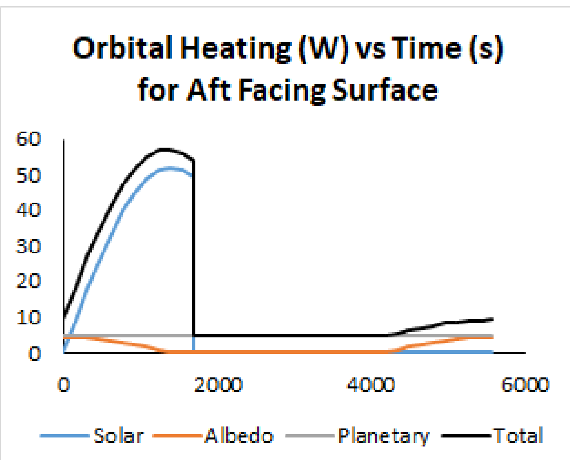
Orbit graphic from Thermal Desktop® by Cullimore and Ring Technologies, Inc. for visualization purposes only.

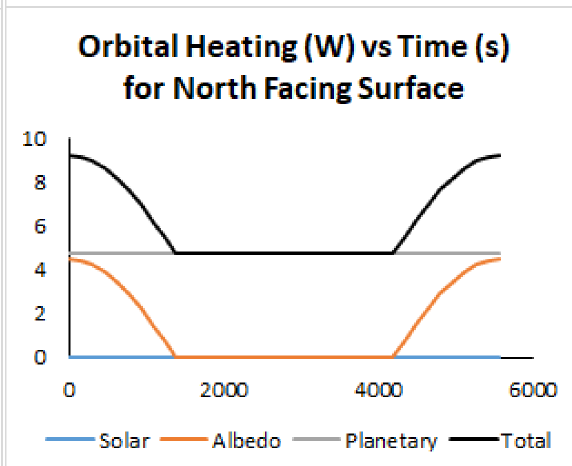
Modeling Time Varying Heat Sources

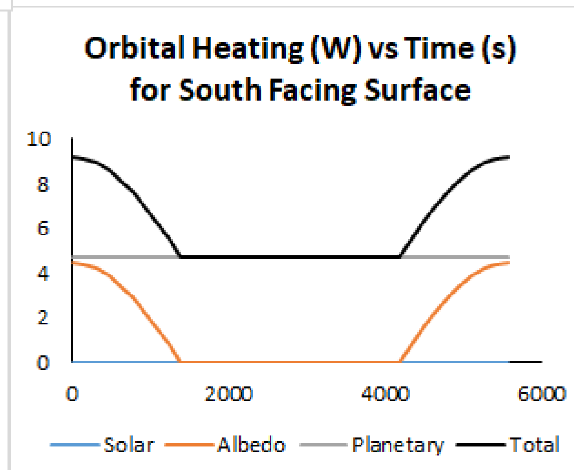












Pre-Defined User Variables

Users can create variables for use in their SINDA calculations;

However, a pre-established list of variables is available without any requirement to create them;

SINDA variables ATEST through ZTEST can be used, as needed by the user:

ATEST – HTEST and OTEST – ZTEST are real numbers

ITEST – NTEST are integers

Using SINDA Subroutines

SINDA has a vast library of subroutines to aid users in model formulation;

For this lesson, we will need two subroutines in our model:

- D11CYL to perform cyclic interpolation of orbital heating.
- THRMST to model a thermostat for a heater.

Using SINDA Subroutines – D11CYL

The D11CYL subroutine is called as follows:

```
CALL D11CYL(PERIOD,X,A#,Y)
```

where `PERIOD` is the period of repetition of `X` and `X` is the X value for which you want to obtain a Y value, `A#` is the SINDA doublet array number containing desired X and Y values, and `Y` is the location where the interpolated result is stored.

From: User's Manual SINDA/FLUINT, General Purpose Thermal/Fluid Network Analyzer, Version 5.8, June 2015, Cullimore and Ring Technologies, Inc.

Using SINDA Subroutines – D11CYL

In the context of orbital heating, if we have an orbit with a period of 5566.3 seconds and we want to pull heating values from array 101 for node 101 at time TIMEN, the interpolation call would appear as:

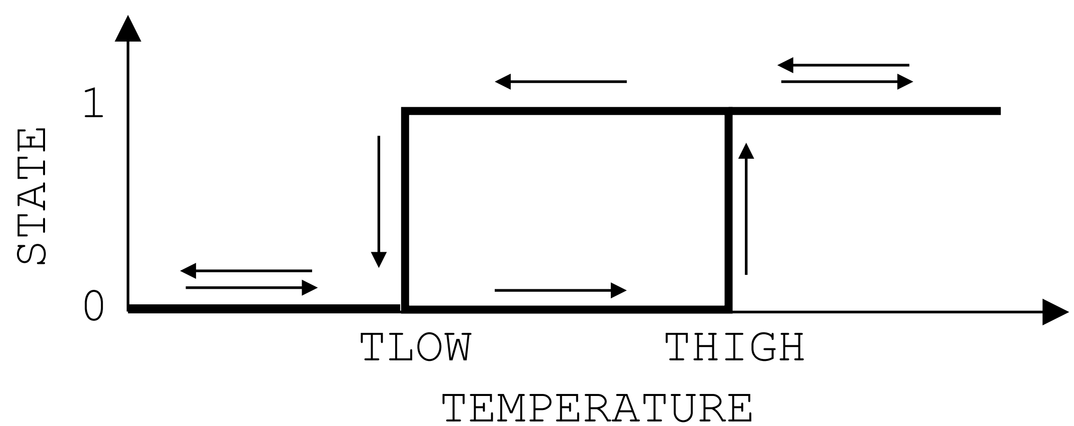
```
CALL D11CYL(5566.3,TIMEN,A101,Q101)
```

Using SINDA Subroutines – THRMST

The THRMST subroutine is called as follows:

```
CALL THRMST(TEMP,TLOW,THIGH,STATE)
```

where `TEMP` is the temperature that is used to control the thermostat, and `TLOW`, `THIGH`, and `STATE` can be ascertained from the diagram below



From: User's Manual SINDA/FLUINT, General Purpose Thermal/Fluid Network Analyzer, Version 5.8, June 2015, Cullimore and Ring Technologies, Inc.

Using SINDA Subroutines – THRMST

If we want heater to cycle between 30 °C and 35 °C based on the temperature of node 410, the subroutine call would appear as:

```
CALL THRMST(T410,30.0,35.0,STEST)
```

However, some additional logic will be required to apply heating to the node – more on this shortly.

Units and ABSZRO

It is important to remember that ***SINDA relies on the user to maintain consistency in units used throughout the model***;

It is critical for users to ensure that the value set of absolute zero (SINDA variable: ABSZRO) is in the units desired for the model;

For our model, we will use SI units so the following assignment must be made in the model in the CONTROL DATA block:

```
ABSZRO = -273.15
```

For model temperatures in °C.

Setting SIGMA

We saw earlier that the form of a radiation conductor (radk) includes the Stefan-Boltzmann constant, σ;

Some analysts prefer to calculate or import radiation conductors from other programs and may choose to include or exclude σ from the radk values;

The SINDA variable `SIGMA` is used to multiply radiation conductors by the Stefan-Boltzmann constant;

The default value of `SIGMA` is 1.0 – in other words, SINDA assumes radiation conductors calculated or imported into SINDA already account for the value of σ.

SINDA Syntax and the Input Deck

For this model, we have decided to calculate our radk inputs and to let SINDA multiply all radiation conductors by σ.

The SINDA parameter for this is SIGMA and in SI units (W/m^2K^4):

```
SIGMA = 5.67E-8
```

and is placed in the CONTROL DATA block.

Assembling the SINDA Input Deck

For our second SINDA model, we will use the following blocks:

```
OPTIONS DATA
NODE DATA
CONDUCTOR DATA
ARRAY DATA
CONTROL DATA
OPERATIONS DATA
VARIABLES 1
OUTPUT CALLS
```

The input deck ends with the last line:

```
END OF DATA
```

OPTIONS DATA

```
HEADER OPTIONS DATA
TITLE   MY SECOND SINDA MODEL
        MODEL  = BOX
        OUTPUT = RESULTS.DAT
        USER1  = TEMPERATURES.DAT
```

Model specification begins with OPTIONS DATA.

Descriptive title.

The model is named BOX.

Model results (and output assigned to logical unit NOUT) will be sent to RESULTS.DAT

Output sent to the USER1 file (logical unit NUSER1) will go to TEMPERATURES.DAT

NODE DATA

```
HEADER NODE DATA, SUB1
C
        SIV 100, 20.0, A1, 0.638313
        SIV 200, 20.0, A1, 0.638313
        SIV 300, 20.0, A1, 0.638313
        SIV 400, 20.0, A1, 0.638313
        SIV 500, 20.0, A1, 0.638313
        SIV 600, 20.0, A1, 0.638313
C
            410, 20.0,      819.0
C
            101, 20.0,     -1.0
            201, 20.0,     -1.0
            301, 20.0,     -1.0
            501, 20.0,     -1.0
            601, 20.0,     -1.0
C
           -999, -270.15,  1.0
C
```

Node Data block header. A submodel name is required.

Six diffusion nodes representing the spacecraft box sides are defined. The mass of each side is 0.638313 kg and the specific heat will be retrieved from array 1.

The component with constant thermal capacitance of 819.0 J/deg C is defined.

Five arithmetic nodes are defined to represent the MLI surface. (MLI for Node 400 is excluded).

A space boundary node is defined.

CONDUCTOR DATA (1 of 2)

```
HEADER CONDUCTOR DATA, SUB1
C
     SIV 100300,   100,  300, A2, 0.00254
     SIV 100400,   100,  400, A2, 0.00254
     SIV 100500,   100,  500, A2, 0.00254
     SIV 100600,   100,  600, A2, 0.00254
     SIV 200300,   200,  300, A2, 0.00254
     SIV 200400,   200,  400, A2, 0.00254
     SIV 200500,   200,  500, A2, 0.00254
     SIV 200600,   200,  600, A2, 0.00254
     SIV 300500,   300,  500, A2, 0.00254
     SIV 300600,   300,  600, A2, 0.00254
     SIV 400500,   400,  500, A2, 0.00254
     SIV 400600,   400,  600, A2, 0.00254
C
         400410,   400,  410,     20.646
```

Conductor Data block header. A submodel name is required.

The spacecraft box side conductors are defined with a temperature dependent thermal conductivity stored in array 2.

The component (410) is connected to the radiator surface (400) using a constant contact conductance.

CONDUCTOR DATA (2 of 2)

```
C
        -100101,   100,  101,        0.004645
        -200201,   200,  201,        0.004645
        -300301,   300,  301,        0.004645
        -500501,   500,  501,        0.004645
        -600601,   600,  601,        0.004645
C
        -999101,   999,  101,        0.069678
        -999201,   999,  201,        0.069678
        -999301,   999,  301,        0.069678
        -999400,   999,  400,        0.081755
        -999501,   999,  501,        0.069678
        -999601,   999,  601,        0.069678
```

The spacecraft box nodes are connected to the MLI surface via radiation conductors. Node 400 is excluded because it is not covered in MLI

Radks to space are calculated for the five MLI surfaces and the radiator surface (400).

Note that the radks presented here do not have the Stefan-Boltzmann constant multiplied into the value. This will be accomplished with the SIGMA variable in CONTROL DATA.

Using Arrays

- Arrays are ways of expressing the variation of one parameter as a function of one (or two) independent parameters.

- For this model, we will limit our use of arrays to those with only one independent variable, x and one dependent variable, y:

$$y = f(x)$$

- We will employ arrays to model temperature dependent specific heat, temperature dependent thermal conductivity, and time varying orbital heating.

Using Arrays

- Consider the following data where x is monotonically increasing:

x	y
x_1	y_1
x_2	y_2
x_3	y_3
$\vdots$	$\vdots$
x_n	y_{n1}

- To represent the variation of y with respect to x, we will use the SINDA doublet array format:

$$Array\ Number = x_1, y_1, x_2, y_2, x_3, y_3, \cdots x_n, y_n$$

- Where $Array\ Number$ is a unique integer identifying the array.

Using Arrays

- The array may be split between lines in any number of ways. The arrays are equivalent to one another.

```
2 =  -40.0, 141.9, -17.8, 145.3,  26.7, 152.3
      48.9, 155.8,  71.1, 159.2,  93.3, 162.7
     115.6, 164.4, 160.0, 169.6, 204.4, 173.1
     226.7, 174.8, 293.3, 179.1, 315.6, 180.0
     382.2, 180.0, 404.4, 179.1, 471.1, 175.7
     537.8, 167.9
```

```
2 =  -40.0, 141.9
     -17.8, 145.3
      26.7, 152.3
      48.9, 155.8
      71.1, 159.2
      93.3, 162.7
     115.6, 164.4
     160.0, 169.6
     204.4, 173.1
     226.7, 174.8
     293.3, 179.1
     315.6, 180.0
     382.2, 180.0
     404.4, 179.1
     471.1, 175.7
     537.8, 167.9
```

ARRAY DATA (1 of 2)

```
HEADER ARRAY DATA, SUB1
 1 = -40.0, 795.3, -17.8, 826.7,  26.7, 879.2
      48.9, 900.2,  93.3, 931.6, 160.0, 973.4
     204.4, 994.4, 248.9,1009.0, 315.6,1036.2
     360.0,1050.9, 426.7,1071.8, 537.8,1130.4
 2 = -40.0, 141.9, -17.8, 145.3,  26.7, 152.3
      48.9, 155.8,  71.1, 159.2,  93.3, 162.7
     115.6, 164.4, 160.0, 169.6, 204.4, 173.1
     226.7, 174.8, 293.3, 179.1, 315.6, 180.0
     382.2, 180.0, 404.4, 179.1, 471.1, 175.7
     537.8, 167.9
```

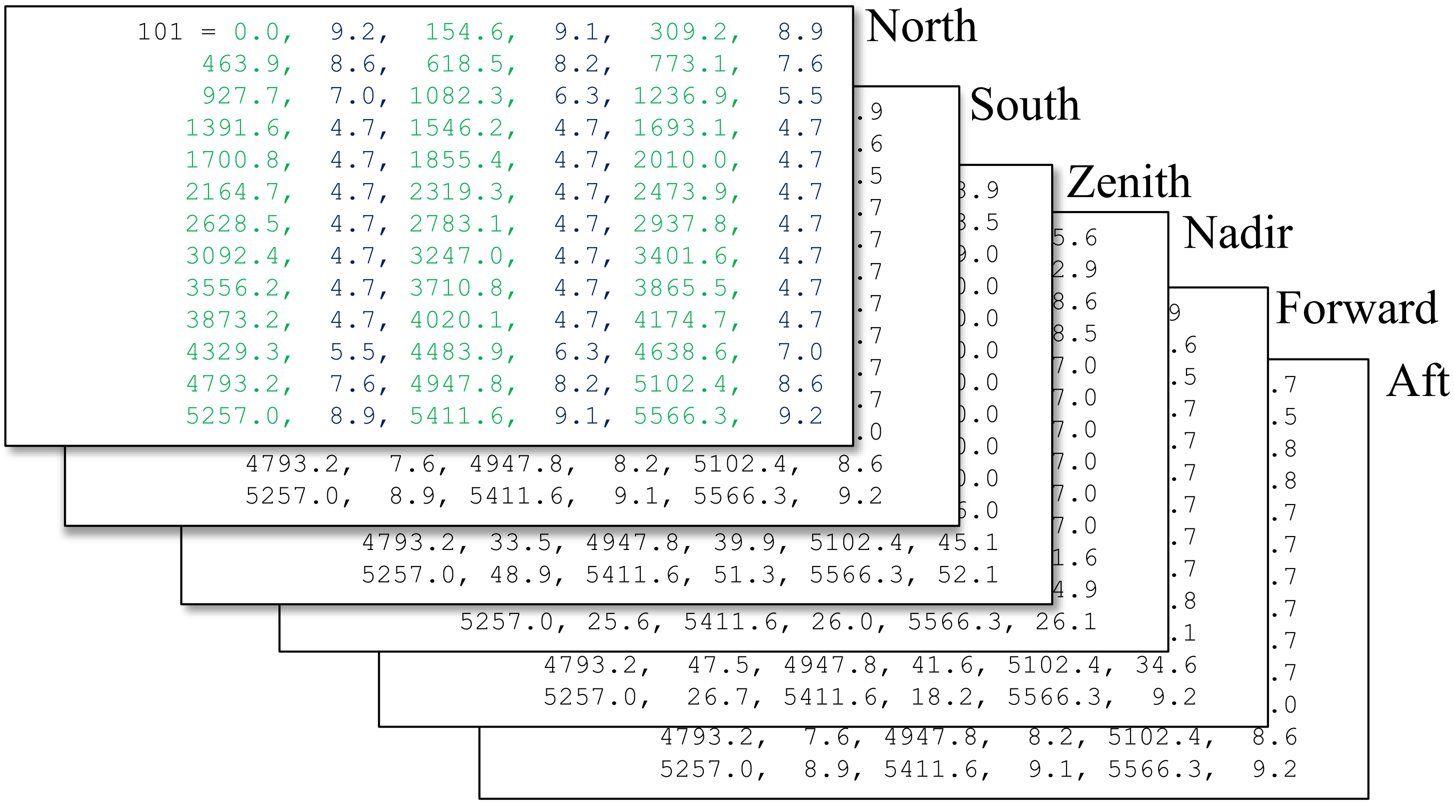
Array Data block header. A submodel name is required.

Array 1 is a T, c_p doublet array.

Array 2 is a T, k doublet array.

Array Data (2 of 2)

The total heating versus time for each box side is represented as a SINDA array – for each time, t a corresponding heating, $\dot{Q}_{total}(t)$:



CONTROL DATA

```
HEADER CONTROL DATA, GLOBAL
          OUTPUT = 5.0
          TIMEND = 5566.3*2.0
          ABSZRO = -273.15
          SIGMA = 5.67E-8
```

Control block header. Note the word GLOBAL is added.

The output interval is set to five seconds.

The model end time is set to twice the orbital period. (But we will reset it elsewhere).

The value of absolute zero, in degrees Celsius, is set.

The Stefan-Boltzmann constant is set in the appropriate units for this analysis and will multiply all radiation conductors.

OPERATIONS DATA

```
HEADER OPERATIONS DATA
BUILD  BOX, SUB1
       ITEST = 0
       CALL TRANSIENT
       TIMEO=0.0
       ITEST = 1
       TIMEND = 5566.3
       CALL TRANSIENT
```

Operations Data block header. Note that no submodel is specified.

Our model will be called BOX and will be composed of a single submodel named SUB1

ITEST is being used as an integer flag to trigger when output will be sent to a file. The output is controlled in the OUTPUT CALLS block in this model. After the first transient solution (to establish a cyclic steady state), the model "old time" solution time is set back to 0.0.

The output flag, ITEST is set to 1 which tells the logic in the OUTPUT CALLS block to send data to file. The model end time is reset for one orbit and a new transient solution is initiated which uses the temperatures from the end of the previous transient run.

Using Variables 1 Logic

One of the things that makes SINDA so powerful is that users have considerable control over the parameters that go into the solution;

As a solution marches through time, we may wish to turn on a heater or apply orbital heating;

In a broader sense, there may be things we want to do, either, before or after a time step.

Using Variables 1 Logic

In this lesson, we're going to introduce a means of affecting model parameters before each time step;

The user interacts with the model using logic that is Fortran-like;

This logic is placed in the Variables 1 block.

VARIABLES 1

Variables 1 block header. Note that a submodel is specified.

```
HEADER VARIABLES 1, SUB1
C
        CALL D11CYL(5566.3,TIMEN,A101,Q101)
        CALL D11CYL(5566.3,TIMEN,A201,Q201)
        CALL D11CYL(5566.3,TIMEN,A301,Q301)
        CALL D11CYL(5566.3,TIMEN,A400,Q400)
        CALL D11CYL(5566.3,TIMEN,A501,Q501)
        CALL D11CYL(5566.3,TIMEN,A601,Q601)
C
        CALL THRMST(T410,30.0,35.0,STEST)
        Q410 = 50.0 * (1.0 - STEST)
C
```

Call to the cyclic interpolation routine D11CYL to take the current time, TIMEN, go to the doublet array A101, and return the heating value into Q101 -- The heating repeats with a period of 5566.3 seconds)

Using the temperature for node 410, determine the thermostat state for an on temperature of 30 °C and an off temperature of 35 °C – Store the thermostat state in the SINDA variable, STEST

Logic to apply 50.0 W to node 410 when the thermostat is on and 0.0 W when it is off

OUTPUT CALLS

Output Calls block header. Note that a submodel is specified.

```
HEADER OUTPUT CALLS, SUB1
      IF(ITEST .EQ. 1) THEN
      WRITE(NUSER1,100) TIMEN, T100, T200, T300, T400, T500, T600,
    +                          T101, T201, T301,       T501, T601,
    +                          T410
      END IF
100   FORMAT(F9.1,3X, 12(F8.1,3X))
```

A FORTRAN format statement is used to format the output.

When the ITEST variable is set to 1, results are written to the USER1 file (logical unit NUSER1) using FORMAT line 100. Note that FORTRAN logic is used and the WRITE statement may continue on subsequent lines as long as a continuation character (+ in this example) is placed in column 6.

Using the INCLUDE Statement

SINDA provides a means of adding data from other files external to the input deck by using the INCLUDE statement.

This allows for a more compact SINDA input file.

For example, if all of the arrays defined for this model are stored in a file named ARRAYS.DAT, they may be added to the model by including the file in the appropriate location in the input deck, as follows:

```
HEADER ARRAY DATA, SUB1
INCLUDE ARRAYS.DAT
```

The SINDA Input Deck

```
HEADER OPTIONS DATA
TITLE  MY SECOND SINDA MODEL
       MODEL  = BOX
       OUTPUT = RESULTS.DAT
       USER1  = TEMPERATURES.DAT
HEADER NODE DATA, SUB1
C
       SIV 100, 20.0, A1, 0.638313
       SIV 200, 20.0, A1, 0.638313
       SIV 300, 20.0, A1, 0.638313
       SIV 400, 20.0, A1, 0.638313
       SIV 500, 20.0, A1, 0.638313
       SIV 600, 20.0, A1, 0.638313
C
           410, 20.0,     819.0
C
           101, 20.0,    -1.0
           201, 20.0,    -1.0
           301, 20.0,    -1.0
           501, 20.0,    -1.0
           601, 20.0,    -1.0
C
          -999, -270.15,  1.0
C
HEADER CONDUCTOR DATA, SUB1
C
     SIV 100300,   100,  300, A2, 0.00254
     SIV 100400,   100,  400, A2, 0.00254
     SIV 100500,   100,  500, A2, 0.00254
     SIV 100600,   100,  600, A2, 0.00254
     SIV 200300,   200,  300, A2, 0.00254
     SIV 200400,   200,  400, A2, 0.00254
     SIV 200500,   200,  500, A2, 0.00254
     SIV 200600,   200,  600, A2, 0.00254
     SIV 300500,   300,  500, A2, 0.00254
     SIV 300600,   300,  600, A2, 0.00254
     SIV 400500,   400,  500, A2, 0.00254
     SIV 400600,   400,  600, A2, 0.00254
C
         400410,   400,  410,    20.646
C
        -100101,   100,  101,      0.004645
        -200201,   200,  201,      0.004645
        -300301,   300,  301,      0.004645
        -500501,   500,  501,      0.004645
        -600601,   600,  601,      0.004645
C
        -999101,   999,  101,      0.069678
        -999201,   999,  201,      0.069678
        -999301,   999,  301,      0.069678
```

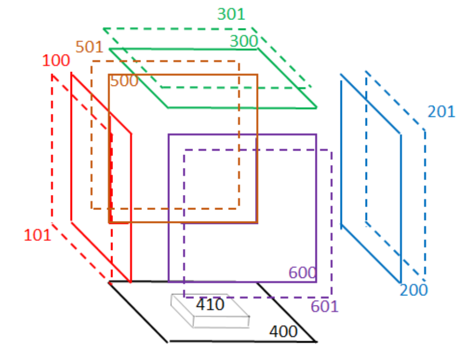
```
        -999400,   999,  400,      0.081755
        -999501,   999,  501,      0.069678
        -999601,   999,  601,      0.069678
HEADER ARRAY DATA, SUB1
INCLUDE ARRAYS.DAT
HEADER CONTROL DATA, GLOBAL
       OUTPUT = 5.0
       TIMEND = 5566.3*2.0
       ABSZRO = -273.15
       SIGMA = 5.67E-8
HEADER OPERATIONS DATA
BUILD  BOX, SUB1
       ITEST = 0
       CALL TRANSIENT
       TIMEO=0.0
       ITEST = 1
       TIMEND = 5566.3
       CALL TRANSIENT
HEADER VARIABLES 1, SUB1
C
C  Determine Orbital Heating Rates
C
       CALL D11CYL(5566.3,TIMEN,A101,Q101)
       CALL D11CYL(5566.3,TIMEN,A201,Q201)
       CALL D11CYL(5566.3,TIMEN,A301,Q301)
       CALL D11CYL(5566.3,TIMEN,A400,Q400)
       CALL D11CYL(5566.3,TIMEN,A501,Q501)
       CALL D11CYL(5566.3,TIMEN,A601,Q601)
C
C  STEST used to identify thermostat on/off state
C
       CALL THRMST (T410,30.0,35.0,STEST)
       Q410 = 50.0 * (1.0 - STEST)
C
HEADER OUTPUT CALLS, SUB1
       IF(ITEST .EQ. 1) THEN
       WRITE(NUSER1,100) TIMEN, T100, T200, T300, T400, T500, T600,
     +                          T101, T201, T301,       T501, T601,
     +                          T410
       END IF
100    FORMAT(F9.1,3X, 12(F8.1,3X))
END OF DATA
```

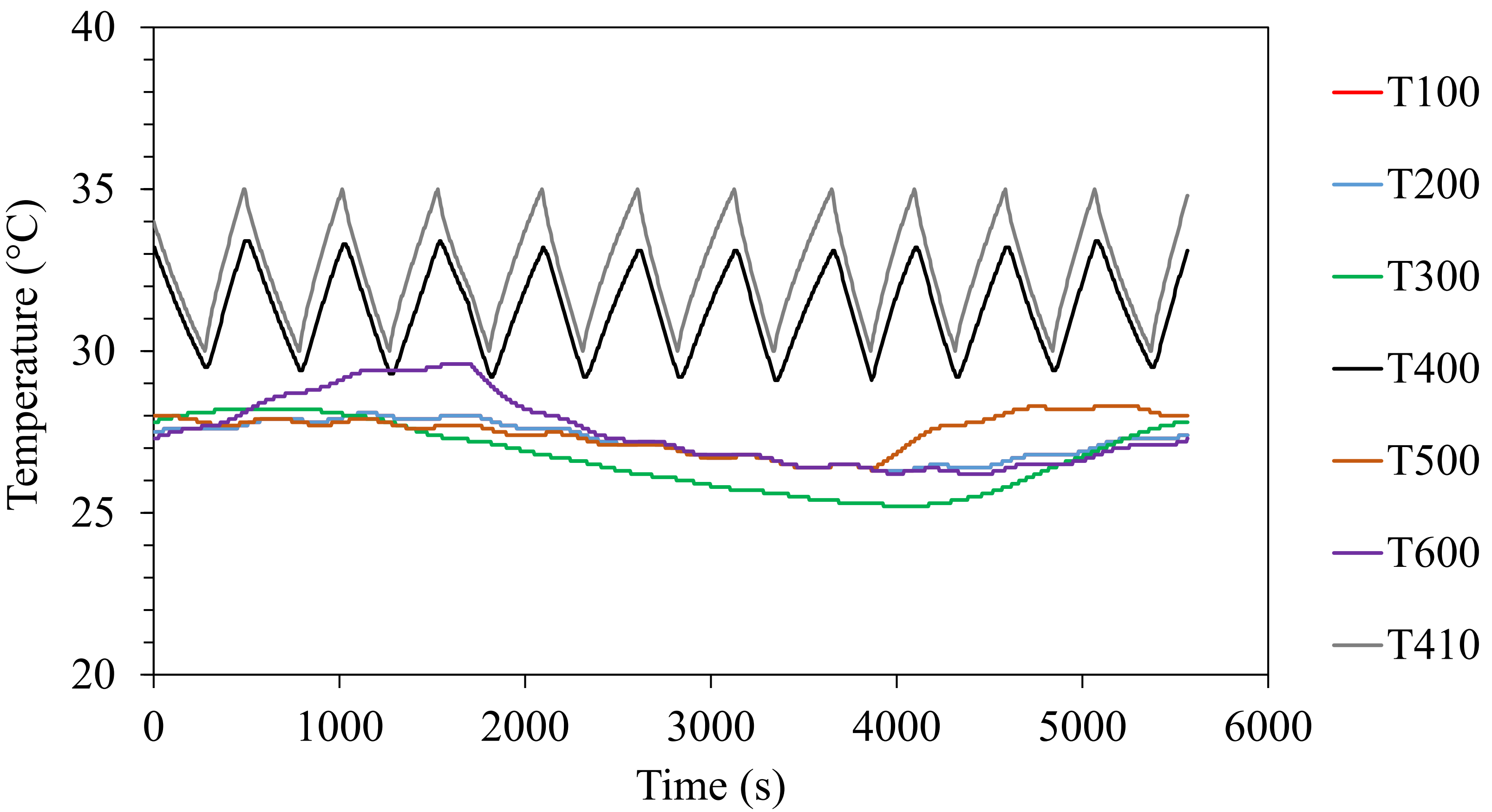
The SINDA Input Deck – ARRAYS.DAT

```
      1 = -40.0, 795.3, -17.8, 826.7,  26.7, 879.2
           48.9, 900.2,  93.3, 931.6, 160.0, 973.4
          204.4, 994.4, 248.9,1009.0, 315.6,1036.2
          360.0,1050.9, 426.7,1071.8, 537.8,1130.4
C
      2 = -40.0, 141.9, -17.8, 145.3,  26.7, 152.3
           48.9, 155.8,  71.1, 159.2,  93.3, 162.7
          115.6, 164.4, 160.0, 169.6, 204.4, 173.1
          226.7, 174.8, 293.3, 179.1, 315.6, 180.0
          382.2, 180.0, 404.4, 179.1, 471.1, 175.7
          537.8, 167.9
C
    101 =   0.0,  9.2,  154.6,  9.1,  309.2,  8.9
          463.9,  8.6,  618.5,  8.2,  773.1,  7.6
          927.7,  7.0, 1082.3,  6.3, 1236.9,  5.5
         1391.6,  4.7, 1546.2,  4.7, 1693.1,  4.7
         1700.8,  4.7, 1855.4,  4.7, 2010.0,  4.7
         2164.7,  4.7, 2319.3,  4.7, 2473.9,  4.7
         2628.5,  4.7, 2783.1,  4.7, 2937.8,  4.7
         3092.4,  4.7, 3247.0,  4.7, 3401.6,  4.7
         3556.2,  4.7, 3710.8,  4.7, 3865.5,  4.7
         3873.2,  4.7, 4020.1,  4.7, 4174.7,  4.7
         4329.3,  5.5, 4483.9,  6.3, 4638.6,  7.0
         4793.2,  7.6, 4947.8,  8.2, 5102.4,  8.6
         5257.0,  8.9, 5411.6,  9.1, 5566.3,  9.2
C
    201 =   0.0,  9.2,  154.6,  9.1,  309.2,  8.9
          463.9,  8.6,  618.5,  8.2,  773.1,  7.6
          927.7,  7.0, 1082.3,  6.3, 1236.9,  5.5
         1391.6,  4.7, 1546.2,  4.7, 1693.1,  4.7
         1700.8,  4.7, 1855.4,  4.7, 2010.0,  4.7
         2164.7,  4.7, 2319.3,  4.7, 2473.9,  4.7
         2628.5,  4.7, 2783.1,  4.7, 2937.8,  4.7
         3092.4,  4.7, 3247.0,  4.7, 3401.6,  4.7
         3556.2,  4.7, 3710.8,  4.7, 3865.5,  4.7
         3873.2,  4.7, 4020.1,  4.7, 4174.7,  4.7
         4329.3,  5.5, 4483.9,  6.3, 4638.6,  7.0
         4793.2,  7.6, 4947.8,  8.2, 5102.4,  8.6
         5257.0,  8.9, 5411.6,  9.1, 5566.3,  9.2
C
    301 =   0.0, 52.1,  154.6, 51.3,  309.2, 48.9
          463.9, 45.1,  618.5, 39.9,  773.1, 33.5
          927.7, 26.0, 1082.3, 17.8, 1236.9,  9.0
         1391.6,  0.0, 1546.2,  0.0, 1693.1,  0.0
         1700.8,  0.0, 1855.4,  0.0, 2010.0,  0.0
         2164.7,  0.0, 2319.3,  0.0, 2473.9,  0.0
         2628.5,  0.0, 2783.1,  0.0, 2937.8,  0.0
         3092.4,  0.0, 3247.0,  0.0, 3401.6,  0.0
```

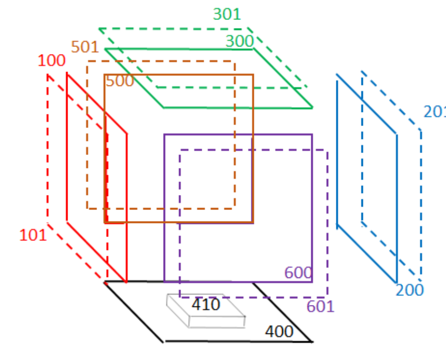
```
         3556.2,  0.0, 3710.8,  0.0, 3865.5,  0.0
         3873.2,  0.0, 4020.1,  0.0, 4174.7,  0.0
         4329.3,  9.0, 4483.9, 17.8, 4638.6, 26.0
         4793.2, 33.5, 4947.8, 39.9, 5102.4, 45.1
         5257.0, 48.9, 5411.6, 51.3, 5566.3, 52.1
C
    400 =   0.0, 26.1,  154.6, 26.0,  309.2, 25.6
          463.9, 24.9,  618.5, 24.0,  773.1, 22.9
          927.7, 21.6, 1082.3, 20.2, 1236.9, 18.6
         1391.6, 17.0, 1546.2, 23.0, 1693.1, 28.5
         1700.8, 17.0, 1855.4, 17.0, 2010.0, 17.0
         2164.7, 17.0, 2319.3, 17.0, 2473.9, 17.0
         2628.5, 17.0, 2783.1, 17.0, 2937.8, 17.0
         3092.4, 17.0, 3247.0, 17.0, 3401.6, 17.0
         3556.2, 17.0, 3710.8, 17.0, 3865.5, 17.0
         3873.2, 28.5, 4020.1, 23.0, 4174.7, 17.0
         4329.3, 18.6, 4483.9, 20.2, 4638.6, 21.6
         4793.2, 22.9, 4947.8, 24.0, 5102.4, 24.9
         5257.0, 25.6, 5411.6, 26.0, 5566.3, 26.1
C
    501 =  0.0,   9.2,  154.6,  9.1,  309.2,   8.9
         463.9,   8.6,  618.5,  8.2,  773.1,   7.6
         927.7,   7.0, 1082.3,  6.3, 1236.9,   5.5
        1391.6,   4.7, 1546.2,  4.7, 1693.1,   4.7
        1700.8,   4.7, 1855.4,  4.7, 2010.0,   4.7
        2164.7,   4.7, 2319.3,  4.7, 2473.9,   4.7
        2628.5,   4.7, 2783.1,  4.7, 2937.8,   4.7
        3092.4,   4.7, 3247.0,  4.7, 3401.6,   4.7
        3556.2,   4.7, 3710.8,  4.7, 3865.5,   4.7
        3873.2,  53.8, 4020.1, 56.0, 4174.7, 56.8
        4329.3,  56.8, 4483.9, 55.2, 4638.6, 52.1
        4793.2,  47.5, 4947.8, 41.6, 5102.4, 34.6
        5257.0,  26.7, 5411.6, 18.2, 5566.3,   9.2
C
     601 =  0.0,  9.2,  154.6, 18.2,  309.2, 26.7
          463.9, 34.6,  618.5, 41.6,  773.1, 47.5
          927.7, 52.1, 1082.3, 55.2, 1236.9, 56.8
         1391.6, 56.8, 1546.2, 56.0, 1693.1, 53.8
         1700.8,  4.7, 1855.4,  4.7, 2010.0,  4.7
         2164.7,  4.7, 2319.3,  4.7, 2473.9,  4.7
         2628.5,  4.7, 2783.1,  4.7, 2937.8,  4.7
         3092.4,  4.7, 3247.0,  4.7, 3401.6,  4.7
         3556.2,  4.7, 3710.8,  4.7, 3865.5,  4.7
         3873.2,  4.7, 4020.1,  4.7, 4174.7,  4.7
         4329.3,  5.5, 4483.9,  6.3, 4638.6,  7.0
         4793.2,  7.6, 4947.8,  8.2, 5102.4,  8.6
         5257.0,  8.9, 5411.6,  9.1, 5566.3,  9.2
```

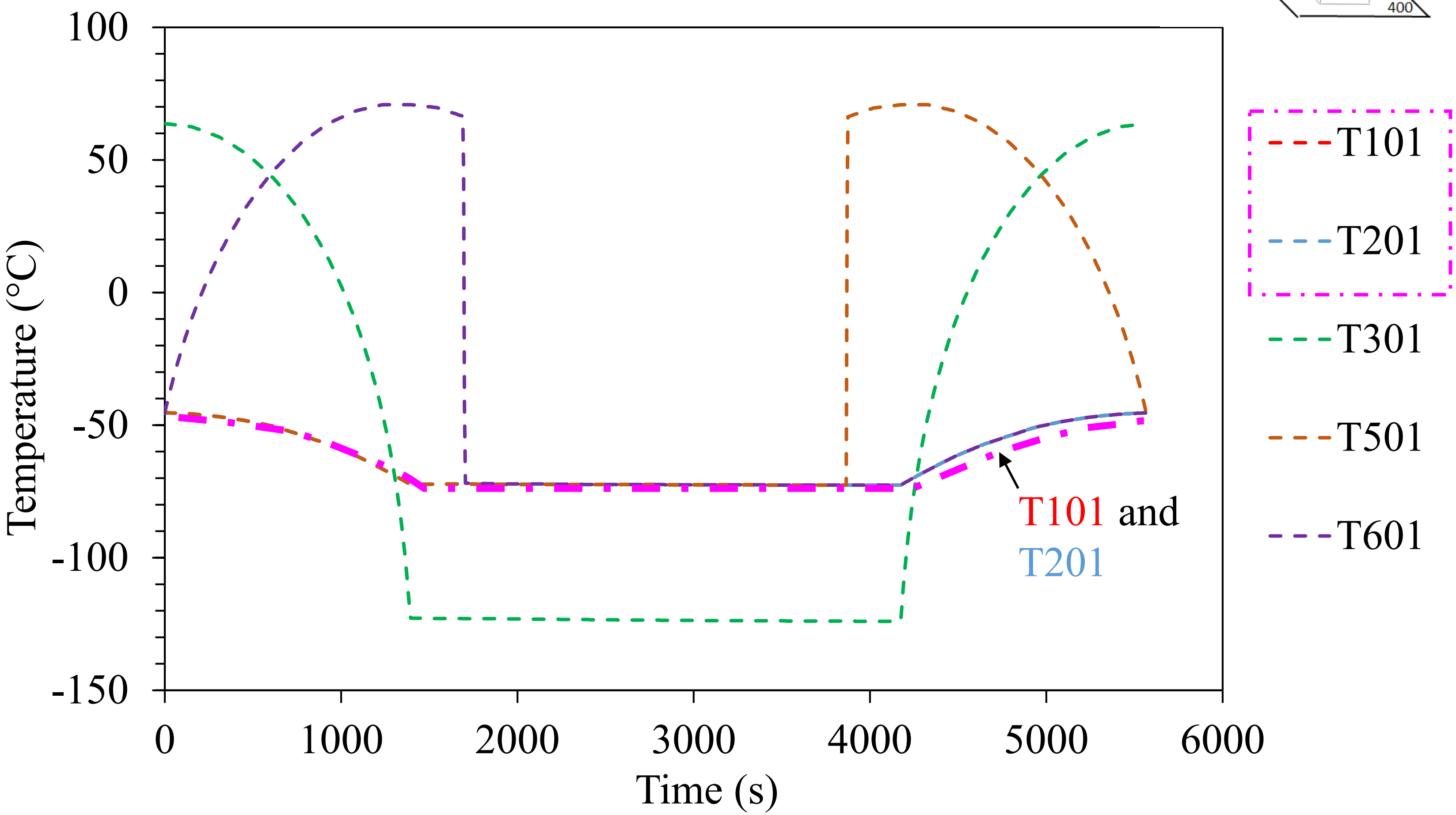
Results – Box/Component Temperatures





Results – MLI Temperatures





Using SIM Statements

In our first SINDA model, we showed a way of expressing multiple nodes or conductors using a single GEN command;

This has the potential to save input time and allow for a more compact input deck;

But GEN cannot be used for materials with temperature dependent properties;

To specify multiple temperature dependent nodes and conductors, the SIM format is required.

Using SIM Statements

The six box nodes required six SIV statements

```
SIV 100, 20.0, A1, 0.638313
SIV 200, 20.0, A1, 0.638313
SIV 300, 20.0, A1, 0.638313
SIV 400, 20.0, A1, 0.638313
SIV 500, 20.0, A1, 0.638313
SIV 600, 20.0, A1, 0.638313
```

However, given that each node is identical, except for the node number, we can abbreviate the definition using the SIM format.

```
SIM N#, #N, Inc. N#, Tinit, A#, Mass
```

The input becomes:

```
SIM 100, 6, 100, 20.0, A1, 0.638313
```

Using SIM Statements

The same is true for the box conductors:

```
SIV 100300,   100,  300, A2, 0.00254
SIV 100400,   100,  400, A2, 0.00254
SIV 100500,   100,  500, A2, 0.00254
SIV 100600,   100,  600, A2, 0.00254
SIV 200300,   200,  300, A2, 0.00254
SIV 200400,   200,  400, A2, 0.00254
SIV 200500,   200,  500, A2, 0.00254
SIV 200600,   200,  600, A2, 0.00254
SIV 300500,   300,  500, A2, 0.00254
SIV 300600,   300,  600, A2, 0.00254
SIV 400500,   400,  500, A2, 0.00254
SIV 400600,   400,  600, A2, 0.00254
```

The SIM format for conductors is:

```
SIM G#, #G, Inc. G#, N#1, Inc. N#1, N#2, Inc. N#2, A#, A/L
```

Using SIM Statements

The modified input becomes:

```
SIM 100300, 4, 100, 100, 0, 300, 100, A2, 0.00254
SIM 200300, 4, 100, 200, 0, 300, 100, A2, 0.00254
SIM 300500, 2, 100, 300, 0, 500, 100, A2, 0.00254
SIM 400500, 2, 100, 400, 0, 500, 100, A2, 0.00254
```

The SINDA Input Deck

```
HEADER OPTIONS DATA
TITLE  MY SECOND SINDA MODEL
       MODEL  = BOX
       OUTPUT = RESULTS.DAT
       USER1  = TEMPERATURES.DAT
HEADER NODE DATA, SUB1
C
C  Box Node Definitions
C
    SIM 100, 6, 100, 20.0, A1, 0.638313
C
           410, 20.0,     819.0
C
    GEN 101, 3,100, 20.0,     -1.0
    GEN 501, 2,100, 20.0,     -1.0
C
          -999, -270.15,  1.0
C
HEADER CONDUCTOR DATA, SUB1
C
C  Box Conductor Definitions
C
    SIM 100300, 4, 100, 100, 0, 300, 100, A2, 0.00254
    SIM 200300, 4, 100, 200, 0, 300, 100, A2, 0.00254
    SIM 300500, 2, 100, 300, 0, 500, 100, A2, 0.00254
    SIM 400500, 2, 100, 400, 0, 500, 100, A2, 0.00254
C
         400410,   400,  410,    20.646
C
    GEN -100101, 3, 100100, 100, 100, 101, 100, 0.004645
    GEN -500501, 2, 100100, 500, 100, 501, 100, 0.004645
C
    GEN -999101, 3,    100, 999,   0, 101, 100, 0.069678
        -999400,            999,          400, 0.081755
    GEN -999501, 2,    100, 999,   0, 501, 100, 0.069678
HEADER ARRAY DATA, SUB1
INCLUDE ARRAYS.DAT
HEADER CONTROL DATA, GLOBAL
       OUTPUT = 5.0
       TIMEND = 5566.3*2.0
       ABSZRO = -273.15
       SIGMA = 5.67E-8
HEADER OPERATIONS DATA
BUILD  BOX, SUB1
       ITEST = 0
```

```
       CALL TRANSIENT
       TIMEO=0.0
       ITEST = 1
       TIMEND = 5566.3
       CALL TRANSIENT
HEADER VARIABLES 1, SUB1
C
       CALL D11CYL(5566.3,TIMEN,A101,Q101)
       CALL D11CYL(5566.3,TIMEN,A201,Q201)
       CALL D11CYL(5566.3,TIMEN,A301,Q301)
       CALL D11CYL(5566.3,TIMEN,A400,Q400)
       CALL D11CYL(5566.3,TIMEN,A501,Q501)
       CALL D11CYL(5566.3,TIMEN,A601,Q601)
C
       CALL THRMST (T410,30.0,35.0,STEST)
       Q410 = 50.0 * (1.0 - STEST)
C
HEADER OUTPUT CALLS, SUB1
       IF(ITEST .EQ. 1) THEN
       WRITE(NUSER1,100) TIMEN, T100, T200, T300, T400, T500, T600,
     +                          T101, T201, T301,       T501, T601,
     +                          T410
       END IF
100    FORMAT(F9.1,3X, 12(F8.1,3X))
END OF DATA
```

Concluding Remarks

In this lesson…

We learned to model temperature dependent properties using the SIV and SIM commands and time dependent heating sources using arrays;

We learned how specify arithmetic nodes and how to use them to represent MLI;

We used logic in VARIABLES 1 to interpolate heating rate arrays and to model a thermostatically controlled heater, and;

We learned how to customize our output using WRITE and FORMAT statements.

Acknowledgements

The author thanks GRC/Ryan Edwards, LaRC/Ruth Amundsen, JSC/Ron Lewis, and JPL/Terry Hendricks for their review and valuable comments on this lesson.

To Contact the Author

Steven L. Rickman
NASA Engineering and Safety Center
NASA Lyndon B. Johnson Space Center
2101 NASA Parkway
Mail Stop: WE
Houston, TX 77058

Email: steven.l.rickman@nasa.gov