Program Documentation

Development of the SINDA Temperature Initialization Program

CPD 909

Prepared by
Lockheed Electronics Company, Inc.
Systems and Services Division
Houston, Texas
Contract NAS 9-15800

for

Structures and Mechanics Division

April 1979
PROGRAM DOCUMENTATION

DEVELOPMENT OF THE SINDA TEMPERATURE INITIALIZATION PROGRAM

Job Order 52-309

Prepared by

P. E. Fogerson

D. Raznahan

Approved by

J. M. Allred, Supervisor
Thermal Technology Section

W. J. Reicks, Manager
Applied Mechanics Department

Lockheed Electronics Company, Inc.
Systems and Services Division
Houston, Texas

April 1979
This document describes a program which builds a steady state SINDA model from a transient model for the purpose of obtaining a set of initial temperatures for the correlation of predictions with test data. The new model converts to boundary nodes those diffusion and arithmetic nodes which correspond to measurement locations.
1. INTRODUCTION

Obtaining a good set of initial temperatures is one of the problems which will be encountered in the correlation of the predictions of the Shuttle Thermal Models with the Orbital Flight Test (OFT) data. Of course, for nodes simulating an area which has a thermocouple, the actual data will be used. The rest of the temperature matrix may be obtained in one of several ways. Considering the large number of measurements which are available and the size of the models (also large), the most accurate temperature distribution will be obtained by fixing the temperatures of the nodes with measurements and performing a steady state solution on the rest of the model. This will require converting many diffusion and arithmetic nodes to boundary nodes, which, if done manually, would be tedious at best. Hence, the SINDA Temperature Initialization Program (STIP) was developed. Using the OFT data, a measurement node correlation list, and the SINDA transient thermal model, this program builds a steady state model which has the nodes from the transient model with measured temperatures converted to boundary nodes. After execution of the steady state solution, the temperature data are saved for subsequent input to the transient model.
2. DISCUSSION

The software package which provides the capability of initializing a thermal model from OFT data consists of two main parts: (1) the program and subroutines which build the new steady state model from the old transient model, and (2) two SINDA subroutines; SAVETS, which stores the final conditions from the steady state run, and LOADTS, which is used in the transient model to retrieve the steady state results. The entire process is shown schematically in figure 1.

2.1 SINDA TEMPERATURE INITIALIZATION PROGRAM

The temperature initialization program builds a SINDA thermal model which is designed to compute a set of initial temperatures for an OFT correlation effort. The program requires the following input:

a. A SINDA model for which a set of initial temperatures is desired
b. A list which specifies the relationship of the nodes of the thermal model to the OFT thermocouple data
c. A data tape containing the temperature data for the thermocouples
d. A namelist specifying user option parameters

The diffusion and arithmetic nodes of the original thermal model (item a) which are listed in the measurement node correlation list (item b) are converted into boundary nodes. They are assigned temperatures that were recorded on the OFT data tape (item c) for the corresponding thermocouple as specified in the measurement node correlation list. Since the OFT tape contains the temperature history for the thermocouples, the times at which to take the temperatures and other execution control parameters are specified in the namelist $NAME.$

The input, or original, model must be a SINDA model and be in card image format on a logical unit. The measurement node correlation list must also be on a logical unit. This list is in the same format as that used as input for the Trend Monitoring System (TMS) prediction tape building program and the batch plotting program; i.e., a nine-character thermocouple identification, followed by an integer specifying the number of nodes that correspond to that
Figure 1. - STIP operational flowchart.
thermocouple, and then the SINDA actual node numbers. All data are in free-field format; i.e., separated by commas. The node numbers are related to a particular thermal model by a six-character model name which separates the thermocouple, node correspondence cards. Figure 2 is a measurement-node correlation list.

The thermocouple data tape must be a Computer-Compatible Tape (CCT) of the format normally output by the Orbital Data Reduction Center (ODRC). User input namelist $NAME provides control of the execution. Table I describes the input variables. The variable IMDOUT specifies the output unit for the processed model, and the variables IMODIN, IUNIT, and TMSNDCC provide the program with the logical unit numbers for input of the original thermal model, the CCT from ODRC, and the measurement-node correlation list, respectively.

The variable ITIMEO is an ordered six-word array which determines the time at which the temperatures are to be taken. The input is six two-digit integers which specify year, month, day, hour, minute, and second in GMT. For example, for March 21, 1981, at 33 seconds past 11:45 a.m., input for ITIMEO would be 81, 03, 21, 11, 45, 33.

The six-character name specified by MODEL is used by the program to locate the section of the measurement-node correlation list which is to be used for this run. The list is searched until a card is found which has in its first six characters the same name specified by MODEL. Processing is terminated when a new model name or an end-of-file is encountered.

The variable PRINT controls the amount of printed output. For normal production runs, the default value of FALSE should be used. PRINT = .TRUE. is intended for checkout or diagnostic runs.

The processed model, i.e., the output model which has some nodes converted from diffusion and arithmetic to boundary, has the nodes in the same physical order; however, most multiple node cards will be unpacked. Thus, the node
Figure 2. - Measurement node correlation list.
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Range</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMDOUT</td>
<td>Logical unit number for model output</td>
<td>1-29 except 5 and 6</td>
<td>8</td>
</tr>
<tr>
<td>IMODIN</td>
<td>Logical unit number for model input</td>
<td>1-29 except 6</td>
<td>7</td>
</tr>
<tr>
<td>ITIMEO(I)</td>
<td>Time in Greenwich Mean Time (GMT) at which data is to be taken from CCT</td>
<td>integer (I2)</td>
<td>None</td>
</tr>
<tr>
<td>I = 1, 6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IUNIT</td>
<td>Logical unit number of the CCT</td>
<td>1-29 except 5 and 6</td>
<td>1</td>
</tr>
<tr>
<td>MODEL</td>
<td>Six-character model name that corresponds to name on measurement-node correla-</td>
<td>6H</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td>tion list</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRINT</td>
<td>Printed output control .TRUE. - full output .FALSE. - partial output</td>
<td>.TRUE. or .FALSE.</td>
<td>.FALSE.</td>
</tr>
<tr>
<td>TMSNDC</td>
<td>Logical unit for measurement-node correlation list</td>
<td>1-29 except 6</td>
<td>11</td>
</tr>
</tbody>
</table>
data block might be somewhat longer physically. The relative actual dictionary will be ordered differently from the original model because the new boundary nodes will be shifted to follow the arithmetic nodes.

The remaining card images of the thermal model are unchanged from the original model except as follows:

a. Any references to the converted nodes are deleted from the source data block.

b. The call to the transient execution routine in the execution block is deleted and replaced with these four cards:

\[
\begin{align*}
F & \quad ARLXCA = 0.01 \\
F & \quad DRLXCA = 0.01 \\
STDSTL \\
SAVETS
\end{align*}
\]

c. All logic is deleted from variables 2

d. Any Q reference (heat input) to the converted nodes is deleted from the operations blocks.

The processed model is not expected to execute correctly without user modification, although there is no harm in making the attempt in the same run that does the conversion.

Figure 3 is an example of a runstream for the STIP. It assumes that the user has a secure file (FILE) which has a measurement-node correlation list and a thermal model as existing elements, and that the user wishes to store the processed model into FILE as an element. The ODRC tape is assigned to logical unit 1, the program default value, and a file is cataloged on logical unit 8 for the output model - also the default value. Since the thermal model and the measurement node correlation list are elements in FILE, the respective logical unit number entries in the Namelist IMODIN and TMSNDC are set to 5. Upon completion of namelist input, the measurement-node correlation list followed by the thermal model are made available to the program via the dynamic add. After the execution is completed, the text editor is used to store the processed model into FILE.
Figure 3. - Runstream for the SINDA temperature initialization program.
2.2 SINDA SUBROUTINES

Since the relative order of the nodes differs between the processed model and the original model, it is not possible to use the standard routine for saving temperatures, HSTFLO, and initializing a subsequent run, FLOTMP. Hence, two new routines, SAVETS and LOADTS, were developed to perform these functions.

2.2.1 TEMPERATURE OUTPUT

After the call to the steady state execution routine, STDSTL, the program places a call to SAVETS for the purpose of saving the relative actual node dictionary and the temperatures which have just been calculated. The output of SAVETS is to unit 25, and is formatted in order that it may be examined by the text editor and/or stored in a file as an element. The dictionary is output first in a 10I8 format followed by the temperatures in 10F8.2. The file, unit 25, is rewound before output and terminated by a software EOF.

2.2.2 TEMPERATURE INPUT

Subroutine LOADTS was developed to allow model initialization to a set of temperatures which were stored by SAVETS. The user specifies the logical unit through an argument. The unit may be 5 if a corresponding dynamic add is placed correctly in the SINDA runstream. The routine searches the node dictionary from the logical unit for a correspondence to each node in the current model. If a match is not found, a message is printed but the initialization process is continued. This means that the model which had the SAVETS call does not have to be identical with the one being initialized. Thus SAVETS and LOADTS have applications beyond the STIP. LOADTS required dynamic storage equal to the total number of nodes of the current model (NNTc) plus twice the total number of nodes of the larger of the current model or the input model (NNTi), i.e.,

\[ \text{dynamic storage} = NNT_c + 2 \times \text{MAX}(NNT_c, NNT_i) \]

The calling sequence is:

\[ \text{LOADTS (I)} \]

where I is the logical unit containing the node dictionary and temperatures. The unit I is automatically released by the program.
3. SUMMARY

The STIP was not really expected to result in a model which, with some user
changes, was completely satisfactory for a steady state execution. Perhaps
the area which deserves most scrutiny is the heating rate inputs. The program
deletes those references to converted nodes from the Source and Variable 1
blocks but leaves the other heating inputs unchanged. As this may not be
acceptable for certain conditions, engineering evaluation is required. The
user may also want to adjust the control parameters for the steady state
routine, or change routines.

STIP will require less than 10 sups for normal applications. The runstream
given in figure 3 when applied to a 5513 card image model of the forward
fuselage required 00:03:12.65 sups. The program uses 10,595 words for
instructions and 19,733 data locations, or total core requirements of 30,328.
The program allows only 1200 nodes to be converted to boundary types.