



**US Army Corps  
of Engineers®**  
Engineer Research and  
Development Center

*Countermine Phenomenology Program*

## **1-D Thermal Modeling of Layered Materials in Outdoor Environments**

John O. Curtis

February 2006



# 1-D Thermal Modeling of Layered Materials in Outdoor Environments

John O. Curtis

*Environmental Laboratory  
U.S. Army Engineer Research and Development Center  
3909 Halls Ferry Road  
Vicksburg, MS 39180-6199*

Final report

Approved for public release; distribution is unlimited

**ABSTRACT:** This report describes a physics-based, but relatively simple, thermal infrared computational tool that can be used in a personal computer (PC) environment to predict temperature profiles of layered media. The tool is a one-dimensional finite difference simulation code (written in FORTRAN) that is executed through a graphical user interface. Its current utility is in helping to design field tests of airborne mine detection sensor systems and to analyze the results of those tests to better understand the performance of the sensor in different environmental conditions. The tool does not address the separate issues of two- and three-dimensional effects, sensor hardware performance, sensor flight paths, or targeting algorithms.

**DISCLAIMER:** The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products. All product names and trademarks cited are the property of their respective owners. The findings of this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

# Contents

---

Preface .....	v
1—Introduction .....	1
Background.....	1
Objective.....	1
The TSTM/VEGIE Thermal Model.....	2
TSTM.....	2
VEGIE.....	2
Previous Applications.....	3
Code Modifications.....	3
2—Basic Principles of the 1-D Thermal Model .....	4
Model Geometry.....	4
Energy Budget Terms at the Air Interface.....	4
Iterative Finite Difference Solution for Surface Temperatures.....	6
Finite Difference Solution for Energy Flow Within Layers.....	7
Finite Difference Solution for Energy Flow Through Layer Interfaces.....	7
Bottom Boundary Conditions.....	7
3—Executing the 1-D Thermal Code in a PC Environment.....	8
Hardware and Software Requirements.....	8
Input Data File Creation.....	9
Model Execution Using the Graphical User Interface.....	12
Rules-of-Thumb for Selecting Surface Material Properties.....	17
4—Example Simulations .....	19
Determining Moisture-Dependent Soil Properties.....	19
A Simulation of Soil Surface Temperatures.....	21
A Simulation of Land Mine Surface Temperatures.....	22
Landmine-Soil Thermal Contrasts.....	24
5—Summary and Recommendations .....	35
Summary.....	35
Recommendations.....	36
References .....	37
Appendix A: Code Listing .....	A1
Appendix B: Example Input File.....	B1

## List of Figures

---

Figure 1.	Geometry of the TSTM model .....	5
Figure 2.	The TSTM/VEGIE Graphical User Interface.....	13
Figure 3.	Test site thermal and optical properties of soil as a function of volumetric moisture content .....	20
Figure 4.	Predicted and measured soil temperatures for the midwestern U.S. test site (day 193).....	22
Figure 5.	Predicted and measured energy fluxes for the midwestern U.S. test site soil simulation (day 193) .....	23
Figure 6.	Predicted and measured soil temperatures for the midwestern U.S. test site (all days) .....	25
Figure 7.	Differences between predicted and measured soil temperatures for the midwestern U.S. test site (all days) .....	26
Figure 8.	Predicted soil temperature profiles for the midwestern U.S. test site (day 193) .....	27
Figure 9.	Predicted and measured land mine temperatures for the midwestern U.S. test site (day 193).....	28
Figure 10.	Predicted and measured energy fluxes for the midwestern U.S. test site land mine simulation (day 193) .....	29
Figure 11.	Predicted and measured land mine temperatures for the midwestern U.S. test site (all days) .....	30
Figure 12.	Differences between predicted and measured land mine temperatures for the midwestern U.S. test site (all days) .....	31
Figure 13.	Predicted land mine (over soil) temperature profiles for the midwestern U.S. test site (day 193).....	32
Figure 14.	Predicted and measured land mine/soil thermal contrasts (days 193-194) for the midwestern U.S. test site .....	33
Figure 15.	Predicted and measured land mine/soil thermal contrasts (all days) for the midwestern U.S test site.....	34

# Preface

---

This report was prepared as part of the U.S. Army Engineer Research and Development Center (ERDC) Countermining Phenomenology Program, which supports the U.S. Army Research, Development, and Engineering Command (RDECOM) Communications and Electronics Research, Development, and Engineering Center (CERDEC) Night Vision and Electronic Sensors Directorate (NVESD) mine detection sensor test and development programs. The NVESD Technical Monitor for this effort was Dr. Tom Broach, of the NVESD Countermining Division, located at Fort Belvoir, VA. Dr. Larry Lynch, ERDC, Geotechnical and Structures Laboratory, was the manager of the Countermining and Phenomenology Program.

Dr. John Curtis, Environmental Systems Branch (EE-C), Ecosystem Evaluation and Engineering Division (EE), Environmental Laboratory (EL), ERDC, in Vicksburg, MS, conducted this study under the direct supervision of Dr. Rose Kress, Chief of EE-C and Mr. Bruce Sabol, Acting Chief of EE-C and the general supervision of Dr. Dave Tazik, Chief of EE. Dr. Ed Theriot was Director of EL, and Dr. Beth Fleming was Acting Director of EL.

Commander and Executive Director of ERDC was COL James R. Rowan, EN. Director was Dr. James R. Houston.

# 1 Introduction

---

## Background

The U.S. Army Engineer Research and Development Center (ERDC) in Vicksburg, MS, has developed a Countermines Phenomenology Program (CPP) to support the U.S. Army Research, Development, and Engineering Command (RDECOM) Communications and Electronics Research, Development, and Engineering Center (CERDEC) Night Vision and Electronic Sensors Directorate (NVEDS) mine detection sensor test and development efforts. One of the issues being addressed by the CPP is that airborne sensors that attempt to identify ground targets often suffer from unexpected high false alarm rates. ERDC researchers have demonstrated in numerous earlier studies that environmental factors often generate target-like signatures. This is true in both the thermal infrared and radar portions of the electromagnetic spectrum.

Modeling of ground target signatures has historically focused on just the targets themselves, or targets embedded in statistically noisy backgrounds. Little, if any, effort has been made to include realistic natural terrain background signatures in the design and analysis of target detection sensors and algorithms. ERDC believes that physics-based terrain element models need to be included in computational platforms that are capable of modeling the complete sensor detection process. This includes target signatures, background (natural terrain) signatures, atmospheric attenuation of those signatures, sensor hardware and flight path, and targeting algorithms.

A fundamental knowledge of the character of natural terrain and the dynamic processes that alter the properties of the terrain (predominantly season, time-of-day, and weather) are key to the success of the CPP. These models will provide a significant improvement over the current method of treating natural environments as statistical clutter. Instead, the specific geometric and material properties of the terrain can be considered and exploited by the sensor system and algorithm developers.

## Objective

The primary objective of this study is to assemble a physics-based, but relatively simple, thermal infrared computational tool that can be used in a personal computer (PC) environment to help design field tests of airborne mine detection sensor systems and to analyze the results of those tests to better understand the performance of the sensor in different environmental conditions. In particular,

this study focuses on a one-dimensional (1-D) layered media simulation code that will yield first-order understandings of target and background signatures. It does not address the separate issues of two- and three-dimensional effects, sensor hardware performance, sensor flight paths, or targeting algorithms. Those are left for a much more sophisticated computational platform that is currently being developed at ERDC.

## The TSTM/VEGIE Thermal Model

### TSTM

In 1981, two reports were published at ERDC, known then as the Waterways Experiment Station (WES), which dealt with a one-dimensional thermal model for predicting surface temperatures of natural terrain elements. The first of these, entitled “Thermal Modeling of Terrain Surface Elements” (Balick et al. 1981a), described a code named the Terrain Surface Temperature Model (TSTM), that simulated non-vegetation-covered surfaces such as bare ground or concrete slabs and their response to variable weather conditions. The basic assumption of the TSTM model was that each of the layers forming the structure, as well as the environment above the structure, was horizontally uniform. In other words, the only significant heat fluxes would be vertical. Under these conditions, physical temperatures within the structure can be found by solving the one-dimensional heat flow equation:

$$\frac{dT(z,t)}{dt} = \alpha(z) \frac{\partial^2 T(z,t)}{\partial z^2} \quad (1)$$

where  $T$  is the physical temperature of some point at a depth  $z$  below the surface at time  $t$ . The thermal diffusivity of the material at that depth  $\alpha(z)$  is defined as the ratio of the thermal conductivity of the material to the product of the mass density and specific heat of the material:

$$\alpha(z) = \frac{\kappa(z)}{\rho(z)c(z)} \quad (2)$$

Clearly, thermal diffusivity of a material measures the rate at which a change in temperature spreads through that material (Jumikis 1977).

A TSTM simulation is driven by air temperature, solar heat flux, and wind speed variations throughout the course of a day. The surface temperature is controlled by an energy balance that will be discussed in a later section.

### VEGIE

The second WES report dealt with a modification of TSTM that is described in its title: “Inclusion of a Simple Vegetation Layer in Terrain Temperature Models for Thermal Infrared (IR) Signature Prediction,” (Balick et al. 1981b). VEGIE, as the new model was named, simply added a layer of vegetative

material to the bare material through the inclusion of several new input parameters including the foliage cover fraction, an index that characterized the state of the vegetation, the graybody emissivity and solar absorptivity of the foliage, and the foliage height. The same kind of energy balance performed in TSTM is required at both the vegetation surface and the ground surface.

## Previous Applications

TSTM/VEGIE and its many variants have been used extensively in numerous ERDC applications. Early attention focused on simulations to support the ERDC mission of fixed facility camouflage, and publications include the two already referenced. Later the code was adapted to another major ERDC research effort, the Smart Weapon Operability Enhancement (SWOE) Program, and used to generate thermal IR images of targets in natural background settings (Welsh 1994).

## Code Modifications

In its original form, the TSTM/VEGIE model could be executed only on a mainframe computer. Variants of the model, used in a number of unpublished ERDC studies, were later installed on workstations and, finally, on PCs. However, none of these model variants could be called “user-friendly.” They were developed for single users and for specialized applications. One goal of this project, then, was to deliver a “user-friendly” version of the TSTM/VEGIE model that operates in a PC environment and is readily transportable from one platform to another. Data input and execution of the code were simplified through the development of a graphical user interface (GUI). Simulation results can now be readily viewed as Excel charts generated at the same time that the simulation takes place. No separate data analysis needs to be performed.

Another limitation of the original TSTM/VEGIE model was that it simulated only one diurnal cycle, utilizing an input data file that was manually created by the user. Therefore, another goal of this study was to conduct multiple-day simulations using input data files that are primarily derived from field weather station micrologger digital files. A detailed description of how to generate those files follows in a later section.

Existing TSTM/VEGIE model variants were limited to constant value thermal properties for each soil layer and constant value optical properties for the surface layer. However, thermal properties in real materials are not constant values. Among other things, they are certainly a function of moisture content (Ochsner et al. 2001). Clearly, over a period of many weeks, soil moisture conditions can change dramatically. It makes no sense to use single-valued thermal properties of soils to conduct a meaningful simulation of conditions at a test site if soil conditions change significantly during that time. Therefore, this new version of TSTM/VEGIE must allow for moisture-dependent thermal properties.

## 2 Basic Principles of the 1-D Thermal Model

---

### Model Geometry

Figure 1 is a visual representation of the TSTM model mode of operation. Although currently limited to six layers by the dimension statements of the code, theoretically any number of layers of material can be represented by a grid of nodes (equally spaced within each layer) whose spacings and properties are used to solve the finite difference form of Equation 1. The energy balance at the surface and the technique used to solve for temperatures at layer interfaces are discussed in the following sections.

### Energy Budget Terms at the Air Interface

The surface temperature of the simulated material is found at each increment of time by an energy balance that can be written in equation form as:

$$S + I - H - E - X + G = 0.0 \quad (2)$$

or

$$D - X + G = 0.0 \quad (3)$$

where

$S$  = net direct short-wave solar radiative flux density, or insolation, received at the air/solid interface

$I$  = net long-wave irradiance (energy flux density impinging on the air/solid interface) from the sky and clouds

$H$  = sensible heat exchange at the surface (primarily convective)

$E$  = latent heat exchange at the surface (primarily evaporative)

$X$  = graybody emittance (energy flux density radiating from the air/solid interface) due to the physical temperature of the surface

$G$  = energy flux density into the solid surface due to conduction

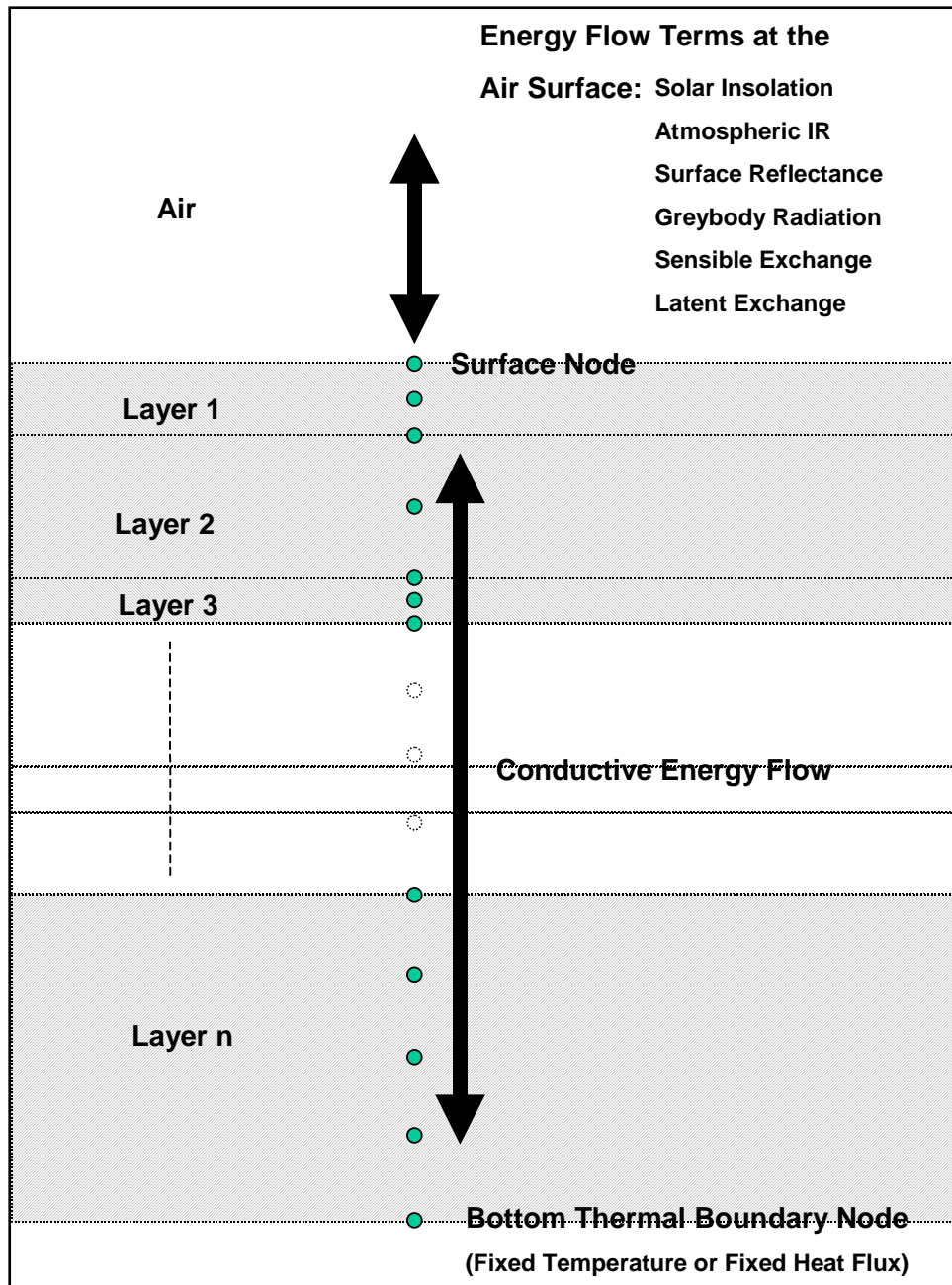


Figure 1. Geometry of the TSTM model

“Short-wave” and “long-wave” are terms used by atmospheric scientists for radiation energy in the 0.15-3.0 micron and 3.0-100 micron wavelength regions, respectively, of the electromagnetic spectrum (Oke 1987).  $I$ ,  $H$ , and  $E$  are all calculated using empirical relationships described in the original TSTM report (Balick et al. 1981a). When short-wave insolation  $S$  is not available as measured data, another empirical relationship referenced in the same report can be used to compute idealized data. This latter technique utilizes the day of the year and the latitude of the test site as controlling factors.

The graybody emittance  $X$  is calculated at each time-step within the simulation using the simple relationship:

$$X = \varepsilon_s \sigma (T_s)^4 \quad (4)$$

where

$\varepsilon_s$  = emissivity of the surface

$\sigma$  = Stephan-Boltzman constant

$T_s$  = current surface temperature predicted by the model

All that remains to define, prior to discussing the numerical solution technique used for these simulations, is the conductive energy flux density  $G$ :

$$G = k \left( \frac{\partial T}{\partial z} \right) = k \frac{(T_1 - T_s)}{\Delta z} \quad (5)$$

where

$k$  = thermal conductivity of the surface layer

$T_1$  = temperature of the first node below the surface

$\Delta z$  = spacing between the surface node and the first node below the surface

## Iterative Finite Difference Solution for Surface Temperatures

Combining Equations 3, 4, and 5, and rearranging terms, one finds that the surface temperature is the root of the following equation:

$$F(T_s) = T_s^4 + \frac{k}{\varepsilon_s \sigma \Delta z} (T_s - T_1) - \frac{D}{\varepsilon_s \sigma} = 0.0 \quad (6)$$

Newton's method is used to find that root:

$$F(T_s)_{new} = F(T_s)_{old} + \left( \frac{\partial F(T_s)}{\partial T_s} \right)_{old} dT_s \quad (7)$$

which provides an estimate for a new surface temperature (by setting  $F(T_s)_{new} = 0$ ):

$$(T_S)_{new} = (T_S)_{old} - \frac{F(T_S)_{old}}{\left(\frac{\partial F(T_S)}{\partial T_S}\right)_{old}} \quad (8)$$

The partial derivative of  $F(T_S)$  comes directly from Equation 6, in which the partial of  $D$  with respect to  $T_S$  is approximated using the previous and current estimates of the surface temperature.

## Finite Difference Solution for Energy Flow Within Layers

For material within each layer of the simulated structure, a central difference form of Equation 1 is used to calculate a new value of temperature at each node  $n$ :

$$(T_n)_{new} = (T_n)_{old} + \frac{\alpha \Delta t}{\Delta z^2} [(T_{n+1})_{old} - 2(T_n)_{old} + (T_{n-1})_{old}] \quad (9)$$

where the “ $n+1$ ” and the “ $n-1$ ” subscripts refer to the nodes immediately below and above the node of interest, respectively.

## Finite Difference Solution for Energy Flow Through Layer Interfaces

The developers of TSTM combined a truncated Taylor series for the temperature of the node adjacent to each side of an interface and the 1-D heat flow equation (Equation 1) to derive a difference expression for the partial derivative of temperature with respect to depth at that interface node looking at the interface from each of the layer materials. Each expression included an estimate for the new interface temperature. Those derivatives multiplied by the thermal conductivity of each layer resulted in two expressions for the conductive heat flux through the interface, one for each of the two adjoining materials. Assuming continuity of the heat flux through the interface, setting the two expressions equal to each other resulted in a lengthy finite difference expression for the interface temperature at the end of the time-step. Details of this derivation can be found in the original TSTM report (Balick et al. 1981a).

## Bottom Boundary Conditions

For all of the simulations conducted during this study, a fixed temperature was chosen as the bottom boundary condition. Selecting either of the other two boundary condition options (a constant heat flux or a constant heat flux combined with a constant temperature radiating surface) results in a finite difference expression that must be evaluated at each time increment. Details can be found in the original TSTM report (Balick et al. 1981a).

# 3 Executing the 1-D Thermal Code in a PC Environment

---

## Hardware and Software Requirements

TSTM/VEGIE is a Fortran code of reasonable size by today's standards. On the author's PC, the source code occupies 45 kilobytes (KB) of disk space, while the executable code occupies only 105 KB of space. In addition to the source code, a user will need a Fortran compiler to facilitate any necessary changes to the code and an Excel spreadsheet software package to execute the GUI and provide visual simulation results in chart and tabular form. The final element of the simulation tools is the input data file, which is described in detail in the next section.

Included on the CD that accompanies this report is a folder labeled "MinGW" that contains the freeware Fortran 77 compiler and other necessary files that were used by the author to conduct the simulations that follow in the next chapter. The "tstm\_files" folder contains the source code, named "tstmforgui.f" (listed in Appendix A), as well as an example input data file ("flw2004 soil.csv," listed in Appendix B) and the resulting output data file ("fort.4").

If it is necessary to make changes to the source code, the user is advised to save a copy of the original source code in a safe place before proceeding. Once that is done, the source code can be opened in any word processing window and the necessary changes made and saved. Then the code must be recompiled. That requires operating in a disk operating system (DOS) command mode. The author's Window's-based PC has a command prompt that opens a DOS window. Once there, the directory needs to be changed to that containing the source code; i.e., by entering the command:

```
cd\tstm_files
```

The source code is compiled and stored as an executable file, named "tstmforgui.exe," in the same folder by entering the command:

```
g77 tstmforgui.f -o tstmforgui
```

Simulation run times will depend upon the PC's speed and memory capabilities as well as the number of nodes simulating the structure and the number of time increments for the simulation. The author's PC has a 2.536-GHz Pentium®4 central processing unit (CPU) and 768 megabytes (MB) of random access memory (RAM). The longest simulation, for which results are shown in the next chapter, included 110 nodes to simulate a land mine over soil and utilized a time increment of 0.0008 minutes. The simulation covered 64 days of weather data and (including 3 days of iterations to achieve simulation stability, took about 24 minutes and 50 seconds of CPU time while occupying 1.7 MB of memory (determined by the size of the code and the size of the specified arrays). Using these numbers to gauge the length of other simulations, one could say that simulation run times should be on the order of 0.11 microseconds/time increment/node.

## Input Data File Creation

Appendix B contains a partial listing of an input data file used to perform one of the simulations described in the next chapter. Line numbers printed on those pages are not part of the data file; they have been added to facilitate the writing of this section.

Input data begins with a single-line description of the simulation. How it reads is the user's choice, but in most cases it will be a description of the test site being simulated. In this case, line 1 reads: "midwestern.test.site.2004." The dots between words facilitate handling of the title within the Excel spreadsheet. The number "11556" was added by a previous execution of the GUI and is not necessary for conducting a simulation.

The bulk of the input file comes from weather station data measured at the test site. Lines 2-27 on page B1 represent only a small portion of field measurement data, the first few lines and the last few lines. This particular file actually contains 11,656 lines of field data. To complete the input data file, the user needs to add a line at the end (line 28) that contains the word "End." It is used by the code to delineate the number of entries and to free the user from counting all of the lines of input data. A few columns of data have to be derived by the user. They will be described in the following paragraphs.

The entries shown on lines 29-46 of Appendix B are not needed to perform a simulation. They represent test site characterization data that has entered into the GUI prior to executing TSTM/VEGIE. These lines were then appended to the original input data file through execution of the GUI. Their only function is to populate data boxes on the GUI when a file is accessed that has been used before. This precludes the necessity of entering all of the GUI data entries by hand each time a new simulation is performed. As the listing on page B1 shows, data file entries must be comma separated (the GUI looks for a comma-separated variable (.csv) input file).

As noted above, lines 2 through 27 in Appendix B are a partial listing of the field measurement data required to execute this code. Most of these data can be

collected at a test site weather station and recorded on a field micrologger. These data will ordinarily be delivered to the user as a spreadsheet or database file (e.g., Excel). It is relatively easy to delete unnecessary columns of data and to add other columns of required data while still in the spreadsheet format.

Field measurement (and complementary) data shown in lines 2 through 27 include the following parameters for each line of data:

- Column 1 Julian day on which the following data were collected
- Column 2 Hour of the day (24-hour clock) on which the following data were collected
- Column 3 Air temperature (deg C) at a known height above the ground
- Column 4 Relative humidity (percent)
- Column 5 Barometric pressure (millibars). The -6999 entries on lines 21-27 dictate to the code that the pressure gauge failed and that a value of 1000 mbars is to be used for that point in time. Any number less than zero would trigger this event.
- Column 6 Solar insolation ( $\text{W}/\text{m}^2$ ). This is the downwelling radiation measured at the weather station by a pyranometer that typically covers the visible and near-infrared portions of the electromagnetic spectrum (400 to 1100 nanometers of wavelength). If these data are not available, the code can be directed to calculate solar insolation on a surface based on the latitude of the test site, time of year, and surface orientation.
- Column 7 Wind speed (m/s) at a known height above the ground
- Column 8 Cloud type (an integer number ranging from 1 to 8). The cloud type index identifies different cloud genera (such as cirrus, stratus, etc.) and triggers correction factors used when the simulation code is directed to generate insolation values (Balick et al. 1981a). In lieu of real data, clear sky conditions are generally identified by cloud types 1 or 2. There is a cloud correction factor for the long-wave irradiance term in the surface energy balance that is also controlled by the cloud type and the percent of cloud cover (to follow). Cloud type must be entered by hand.
- Column 9 Cloud cover (percent). This column of data is also entered by hand and could be significant if insolation is being computed. Otherwise, its effect can be negated by setting all values in this column to zero.
- Column 10 Saturation factor. This is a decimal number ranging from 0.0 to 1.0 that triggers the latent heat exchange calculation. The original documentation for the TSTM code identified this term as the relative saturation of the top surface material but used it only as a weighting factor for controlling the impact of evaporative cooling on the simulation. Furthermore, the original code used the wind speed indicator height above the ground as a factor in both the empirical latent heat exchange and sensible heat exchange functions, but the

source for those functions (Oke 1987) specified that the log height ( $z/\ln(z)$ ) of the wind speed indicator should be used in the formulation, because of an assumed exponential wind speed profile. In other words, this author had some concern about the physics behind the use of this saturation factor. As a result of this concern, log height has been inserted into the sensible and latent heat exchange relationships. Furthermore, the numbers in this data file column are now reasonable approximations to the actual near surface degree of saturation. If near-surface volumetric moisture content data are available (Column 11), then saturation factor values are computed as the ratio of that moisture content to an assumed porosity of the soil (0.40).

- Column 11 Volumetric soil moisture no. 1. Because soil thermal properties are dependent upon moisture content, an attempt was made in this code to track changes in moisture content as a function of depth in the soil. This number represents the moisture content recorded at the shallowest depth on the test site (if those data were collected).
- Column 12 Volumetric soil moisture no. 2. This is the moisture content in the soil at the deepest depth recorded on the test site. Along with a fixed moisture content boundary value at a third depth, the numbers in columns 11 and 12 were used to compute a soil moisture profile at every time step in the simulation.
- Column 13 Physical temperature no. 1. A thermistor or thermocouple temperature measurement made at the same depth in the soil. This number is not used in the simulation, but could be compared to the simulation results as a measure of goodness.
- Column 14 Physical temperature no. 2. Another temperature measurement at another depth. For the simulations reported in this study, the two temperature measurement depths in this data file corresponded to the volumetric moisture measurement depths.
- Column 15 Radiometric temperature no. 1. This column and the next contain surface temperatures measured with a staring radiometer (see the report cover photograph) that can be compared to the surface temperature predictions made by the TSTM/VEGIE code. For this study, the temperatures in column 15 are those of the land mine shown in the cover photograph.
- Column 16 Radiometric temperature no. 2. A second surface temperature that could also be used to verify the simulations. For this study, these temperatures are of the bare soil shown in the cover photograph.

Volumetric soil moisture data are not critical if soil thermal properties that are independent of moisture values are going to be used in the simulation. Physical and radiometric temperatures are also necessary only if one wishes to validate the model simulations at a given test site. With sufficient experience, a model user can generate sensible, if not accurate, results for any test site without

the burden of validating the simulations against real data. Naturally, the optimum situation is one of validated simulations.

## Model Execution Using the Graphical User Interface

Within the “tstm\_files” folder found on the enclosed CD, there exists an Excel spreadsheet called “TSTM simulation results template.xls.” It contains all of the charts that are used to display results for any simulation. To perform a TSTM/VEGIE simulation, the user must open the Excel spreadsheet on his PC, pull down the “Tools” menu, select “Macro,” then select “Macros” from the sub-menu, and then click on “TSTM” in the Macro window that appears. This action executes a Visual Basic (VBA) program that displays the GUI (Figure 2) which, in turn, controls the simulation and display of results. A listing of the TSTM macro can be found in Appendix C.

While the weather data and temperature and moisture measurements described in the previous section form the bulk of the input data file, the user must still provide other parameter values that control the flow of simulation output as well as define thermal and optical properties of the layered structure being simulated. Referring to the highlighted text and buttons shown on Figure 2, the following procedure should be followed to perform a TSTM/VEGIE simulation.

**Select Input File:** Clicking on this button produces a window that helps the user select the .csv file that contains the site description, the weather data and moistures and temperatures described above, and ends with a line containing the word “End.” If a given input data file has not been previously accessed, then additional data must be entered into the GUI input boxes by hand. If, on the other hand, the chosen data file has been previously accessed, then it is possible that the additional data have already been appended to the input data file (lines 29-46 in Appendix B). Those data will be used to automatically populate most of the GUI input data boxes as soon as the input data file is selected. In either case, the macro scans the data file for the range of days, counts the number of input data file line entries, and displays those results in the appropriate input boxes.

**Single-day simulation control:** In this area of the GUI user form, the user is allowed to choose whether he/she wants to do a single-day simulation or a simulation for all of the days listed in the input data file. For the example shown, a single-day simulation was chosen, and that day was 210. Even if a multi-day simulation was selected, the user still has the option of specifying a single day for which results will be displayed.

**Surface properties:** Several parameters are required to properly specify air-soil interface conditions. These include a solar insolation flag, optical properties, and flags that control how latent heat and sensible heat calculations will be carried out within the simulation. For this example, the flag indicating that solar insolation values would be read from the input data file was chosen. If the user had chosen to let TSTM/VEGIE calculate solar insolation values, then he/she

would have to enter the surface element slope, azimuth, and latitude of the test site into the appropriate boxes. A slope of zero degrees is horizontal. The azimuth angle of the surface, in degrees, only has meaning if the slope of the surface is not horizontal. An azimuth angle of zero degrees means that the projection of the surface normal unit vector onto the local horizontal plane at that location points south. Positive angles are clockwise from south. The test site latitude is also expressed in degrees.

**TSTM Simulation Control Panel**

Select Input File: C:\tstm\_files\flw2004 soil.csv (file name)

150 first Julian day, 214 last Julian day, 11656 number of weather parameter data entries, midwestern.test.ste.2004 (file descriptor)

**single-day simulation control**  
 single-day simulation (210 Julian day for single-day simulation and/or for single-day charts and profiles)  
 multi-day simulation

**surface properties**  
 solar insolation from data file (0 deg slope, 0 deg azimuth, 0 site latitude)  
 solar insolation from calculations  
 fixed degree of saturation (slope of emissivity function: -0.0052, intercept of emissivity function: 1, slope of absorptivity function: 0.007, intercept of absorptivity function: 0.4)  
 variable degree of saturation (decimal value of surface degree of saturation: 0)

**layer definitions**

	thickness cm	node spacing cm	diffusivity slope	diffusivity intercept	conductivity slope	conductivity intercept	deltat must be <	
layer 1	2	0.2	0	0.4	0.028	0	0.05	
layer 2	8	0.5	0	0.4	0.028	0	0.3125	
layer 3	40	1	0	0.4	0.028	0	1.25	
layer 4	200	5	0	0.4	0.028	0	31.25	
layer 5	0	0	0	0	0	0		
layer 6	0	0	0	0	0	0		
	250	total thickness						

**moisture profile parameters**  
 50 depth (cm) below which the volumetric soil moisture is fixed  
 10 value (%) of fixed volumetric moisture content  
 (these are not needed if thermal properties are constants)  
 1.5 depth (cm) of column 11 moisture probe  
 4.5 depth (cm) of column 12 moisture probe

**vegetation parameters**  
 0 foliage cover fraction  
 0 stomatal resistance multiplier  
 0 foliage graybody emissivity  
 0 foliage absorptivity  
 0 foliage height (cm)

**miscellaneous simulation controls**  
 30 output print interval (minutes)  
 8 number of 1st day iterations to achieve simulation initial conditions  
 300 height (cm) of wind speed indicator above the ground  
 0.04 simulation time increment (minutes)

**bottom boundary**  
 fixed flux (flux value (cal/cm<sup>2</sup>))  
 fixed temperature (25 temperature (deg C))  
 radiating surface (flux, emissivity 1, shape factor 1, emissivity 2, shape factor 2, surf temp (deg C))

Update deltat's  
Execute TSTM

Figure 2. The TSTM/VEGIE Graphical User Interface

Two numbers are required in this section that represent the slope and intercept of a linear equation that defines the long-wave emissivity of the surface material as a function of surface volumetric moisture content. One method of determining the relationship between optical properties and moisture content is described in Chapter 4. If one does not know how long-wave emissivity varies as a function of moisture content, then he/she can choose emissivity to be a fixed value, in which case the slope should be set equal to 0.0.

Two additional numbers are required that represent the slope and intercept of a linear equation that defines the shortwave absorptivity of the surface material as

a function of surface volumetric moisture content. The same instructions hold for constant values as for the emissivity numbers.

The final option given to the user in the “surface properties” section is to set the flag for allowing the surface to have either a fixed degree of saturation or a variable degree of saturation. Degree of saturation is a parameter used in the calculation of latent heat transfer. The variable condition was described earlier as being related to the availability of near-surface volumetric moisture data (column 10 in the input data file). If a constant value for degree of saturation is chosen, then that value will be used for all of the simulation time-steps.

**Layer definitions:** This is the section of the GUI where the user can define up to six layers of solid material for which the simulation is being performed. In addition to the number of layers chosen, one must enter six numbers that define each layer geometry and the thermal properties of each layer. In this example, there were four material layers. The first number is the layer thickness, in centimeters. The next is the node spacing for that layer, in centimeters. They are followed by the slope and intercept of the linear relationship between thermal diffusivity (units of square centimeters per minute) and volumetric moisture content. The last two numbers are the slope and intercept of the linear relationship between thermal conductivity (units of calories per centimeter-minutes degrees Celsius) and volumetric moisture content. As with the optical properties, constant thermal properties can be defined by setting the slope values to 0.0. The “deltat” entry will be discussed below.

**Bottom boundary:** One of three conditions may be chosen for the bottom boundary of the material column being simulated. One is a constant heat flux that can be specified by the user. The second, which is the one most commonly used, is to specify a constant temperature boundary. For the example shown in Figure 2, the bottom boundary temperature was fixed at 25 °C. This is a very reasonable condition in soils over a time period of a few days, or even weeks. However, temperatures can vary at depths of 2 or 3 m when viewed on a seasonal basis. The final bottom boundary condition is that of a fixed flux lower boundary that faces a fixed-temperature radiating surface beneath the bottom boundary. One could imagine that such a condition might represent a layered medium over a cavity. The input values required for this third boundary condition include the bottom boundary flux, the temperature of the radiating surface, the emissivities of both surfaces, and two shape factors, which are related to the emitting and absorbing efficiencies of those surfaces.

**Moisture profile parameters:** Moving to the upper right area of the control panel, the user is next asked to define parameters that describe how moisture conditions in the materials vary as a function of depth. Those moisture values will be used by the code to calculate moisture-dependent thermal properties for each material layer. If the material is a man-made solid, then there will be no moisture variability, and these parameters are not needed. Furthermore, if the material is porous, but the user chooses to set thermal properties to constant values for the entire simulation, then these parameters are not needed.

If volumetric moisture data are available in the data input file (columns 11 and, possibly, 12), then the following parameters may be used to help track a

realistic moisture profile throughout the layered media. This would be a common need for simulating soils. The parameters include a depth below which the moisture content is fixed, the value of that fixed moisture content, and the depths of the volumetric moisture meters whose data are listed in columns 11 and 12 of the input data file.

**Vegetation parameters:** There are five parameters required to define surface vegetation conditions. If the first number (or only number) is zero, then the vegetation contributions will be skipped. The first parameter is a number between the values of 0.0 and 1.0 that defines the foliage cover fraction and that can be roughly related to the leaf area index. The second number is a multiplier of the stomatal resistance function for stressed plants. A third parameter is the graybody emissivity of the foliage, while the fourth number is the shortwave absorptivity of the foliage. The last parameter is the foliage height, in centimeters. A simulation using the foliage cover option was not conducted for this study. The reader is referred to the original VEGIE report (Balick et al. 1981b) for a more thorough discussion of this option.

**Miscellaneous simulation controls:** The final four parameters that can be specified by the user are found in this area of the GUI, the first of which is the interval at which the user wants to see simulation output results sent to the output file. For the simulation depicted by Figure 2, the user wanted results displayed at half-hour intervals.

The second number specifies how many iterations on the first day of simulation will be performed to achieve something of a steady-state environment. While surface temperatures are very much controlled by the energy balance at the surface, several iterations on the first day's simulation might be required to achieve a repeatable set of temperature-depth profiles for that day. Typically, only a few iterations are required to achieve stability.

Another parameter specified in this area of the GUI is the height above the ground, in centimeters, at which the wind speed measurements were made at the test site weather station. It is the number that is used in both the latent and sensible heat exchange calculations.

The final parameter is the time increment for this simulation, in minutes. Since TSTM/VEGIE functions as an explicit finite difference code, a *stability condition* exists for the time increment that must be satisfied for all material layers. Within each layer, Equation 9 controls the calculation of the next time increment. As long as the coefficient of the bracketed term is less than  $\frac{1}{2}$ , the calculation will not violate the second law of thermodynamics and the results will be stable (Holman 1968). While this is not an airtight proof, consider the following conditions. Let the temperature of the two nodes surrounding the center node in the finite difference calculation be equal and less than that of the center node. While the new temperature of the center node (at the end of the time increment) should be less than its old temperature (at the beginning of the time increment), it should not be less than that of the surrounding nodes. In mathematical notation, let

$$(T_{n-1})_{old} = (T_{n+1})_{old}$$

and

$$(T_n)_{old} = (T_{n+1})_{old} + \Delta T$$

For stability,

$$(T_n)_{new} > (T_n)_{old} - \Delta T$$

or

$$(T_n)_{new} - (T_n)_{old} > -\Delta T$$

If one defines

$$M = \frac{\alpha \Delta t}{(\Delta z)^2}$$

then from Equation 9,

$$M \left[ 2(T_{n+1})_{old} - 2\{(T_{n+1})_{old} + \Delta T\} \right] > -\Delta T$$

or,

$$M < \frac{1}{2}$$

In other words, the stability condition that must be met for each layer of material is:

$$\Delta t < \frac{(\Delta z)^2}{2\alpha} \tag{10}$$

The GUI is set up to calculate a limiting time increment for each layer according to Equation 10. If thermal diffusivity values are defined as being dependent on volumetric moisture, then a value of diffusivity at a moisture content of 40 percent is taken as an upper bound value (assumes diffusivity increases with moisture content). To display the maximum time increments allowed for each layer, the user simply presses the “Update deltat’s” button on the lower right corner of the GUI screen. The numbers shown in the right-hand column of the “layer definitions” area of the screen are then displayed. For the example shown

in Figure 2, the simulation time increment was controlled by the properties of the top soil layer, which required an increment of less than 0.05 minute.

**Execute TSTM:** When the user is satisfied that all simulation parameters have been properly defined, he/she may proceed with the simulation by pressing the “Execute TSTM” button at the lower-right corner of the GUI screen. What will happen immediately is that a small message window will pop up on the display screen that will say “Wait for TSTM to finish executing!” Since there is currently no way to monitor the progress of the simulation (it is proceeding through a macro shell command), the user must watch the color intensity of the message box (it will brighten when the code finishes execution) or watch the taskbar buttons across the bottom of the screen (there will be one for “tstmforgui.exe” that will disappear when the code finishes). The message box forces the macro behind the GUI to pause while the Fortran code executes. Once the simulation has finished, the user must press the “OK” button on the message box.

The next thing that will happen is another message box will appear asking: “Which column contains the measured surface temperature?” The spreadsheet page containing the simulation output values will be in the background. If the input data file contained a column of measured surface temperature data, then the charts generated by this macro can include the measured data as well as the simulated data. For the input data file described earlier, there were two columns of data containing surface measurements. Column 13 (or column “m” as seen in the background spreadsheet page) contained the mine surface measurement, and column 14 contained the soil surface measurement. If no measured surface data exists, then the user should select a column that has no data.

A second message box will then appear asking: “Which column contains the difference between measured and simulated temperatures?” Again, if such data do not exist, then the user can avoid plotting useless data by naming a column without data. For the simulation for which results will be shown in the next chapter, measured and simulated temperature difference results were displayed in column “o” for the mine surface and column “p” for the soil surface. Answering this final question and clicking the “OK” button completes the TSTM/VEGIE simulation. As the macro is currently written (Appendix C) two charts representing simulation results will be sent to the printer. One is the single-day simulation result for surface temperature compared to the measured data and the other is the set of 2-hr snapshots of temperature profiles as a function of depth for the chosen day. Example charts may be viewed in the next chapter.

## Rules of Thumb for Selecting Surface Material Properties

This model can be executed in one of two ways. First of all, one can select material properties for all of the layers of material based on published data and previous experience with the model and simply predict surface temperatures for whatever the input weather parameters may be.

The second method (which is much more realistic for natural materials with thermal properties that are expected to vary with volumetric moisture content) is to select weather extremes for which iterative single-day simulations will be performed to establish the optimum set of properties for each set of weather

conditions. For example, one can choose a day for which site soils would be quite dry near the surface. While allowing the code to use moisture-depth profiles specified by the user (see previous section), the user can determine the optimum values of constant thermal and optical properties that give the best comparison to measured data. Those values can then be assigned to an average soil moisture for that day. The user can then select a wet-soil day and repeat the process. Finally an intermediate soil moisture day can be simulated in the same trial and error manner. The user then will have a crude relationship between each property and soil moisture which, in turn, becomes part of the input data file for a complete multiple-day simulation for that test site. An example of this process is shown in the next chapter.

The following table provides a useful summary of how property value changes for these iterative simulations will change the resulting predicted daytime temperatures of the surface. While the effects of long-wave emissivity and short-wave absorptivity are very sensible, the impact of changes in thermal diffusivity and thermal conductivity are less intuitive. One way to rationalize their effects is to combine the defining relationships for specific heat and thermal diffusivity in the following way. Consider first the relationship that defines how much heat energy  $Q$  is required to raise the temperature of a lump of material (mass  $m$  and volume  $V$ ) by an amount labeled  $\Delta T$  (Ohanian 1985):

$$Q = mc\Delta T = \rho V c \Delta T \quad (11)$$

When combined with Equation 2, which defines thermal diffusivity, one can easily show that

$$\Delta T = \frac{\alpha Q}{\kappa V} \quad (12)$$

In other words, for a given amount of heat energy flowing into a fixed volume of material, an increase in thermal diffusivity will result in an increase in material temperature. The inverse is true for an increase in thermal conductivity.

<b>Table 1 Rules of Thumb for Selecting Surface Material Properties</b>		
<b>Material Property</b>	<b>Physical Description</b>	<b>Effect on Predicted Daytime Temperatures Due to an Increase in the Property Value</b>
Long-wave Emissivity	A measure of the rate at which a surface can radiate IR energy to its surroundings	Decrease
Short-wave Absorptivity	The fraction of incoming solar radiation that is absorbed by the surface material	Increase
Thermal Diffusivity	The ratio of thermal conductivity to volumetric heat capacity. A measure of the speed with which temperature spreads throughout the material.	Increase
Thermal Conductivity	A measure of the rate of heat energy flow through the material to a lower temperature reservoir.	Decrease

## 4 Example Simulations

---

Weather data, volumetric soil moisture values, and radiometric surface temperature measurements were collected at a midwestern test site during the summer of 2004. These data covered a time period of 64 days at a rate of one set of readings every hour for the first 56 days and one set of readings every minute for the next 8 days.

### Determining Moisture-Dependent Soil Properties

Laboratory data clearly show that soil thermal properties are a strong function of moisture content (Ochsner et al. 2001), with a general trend of increasing thermal conductivity, volumetric heat capacity, and thermal diffusivity with increasing volumetric moisture content. Those same measurements also show that the spread in the data is large enough that simple model fits cannot be used for predictive purposes. In fact, other sources argue that thermal diffusivity decreases with increasing moisture content at higher moisture values, because the volumetric heat capacity increases faster than the thermal conductivity (Jumikis 1977).

If a realistic simulation of a layered test site soil is going to be performed over a period of several weeks, during which multiple rain events followed by drying periods occur, then some accounting for a change in thermal (and possibly optical) properties with changing soil moisture content must be made. The approach taken by the author to deal with this dilemma is to conduct at least three single-day simulations for different soil moisture conditions and adjust the surface material properties to best match measured data. Simple model fits to those thermal (and optical) properties plotted against soil moisture content can then be used to define the input data file properties for a multi-week simulation in which the soil moisture conditions were highly variable.

For the following simulations of surface temperatures at a midwestern U.S. test site, single-day calculations were done for day 193 (average measured volumetric moisture content equal to 5.5 percent, peak measured surface temperature equal to 54 deg C), day 178 (7.0 percent, 44 deg C), and day 212 (15.0 percent, 28 deg C). Trial-and-error simulations were performed for each of these days, resulting in a different set of values for both the thermal properties and the optical properties of the surface soil. Those values and the corresponding regression model fits are shown on Figure 3.

It is important to remember that these relationships between thermal and optical properties and volumetric soil moisture content hold for this site and this type of soil. At another test site, or even at another location within this particular test site, a similar single-day simulation exercise might result in a much different set of thermal and optical parameters. In other words, *thermal and optical properties for soils are very likely to be site-dependent.*

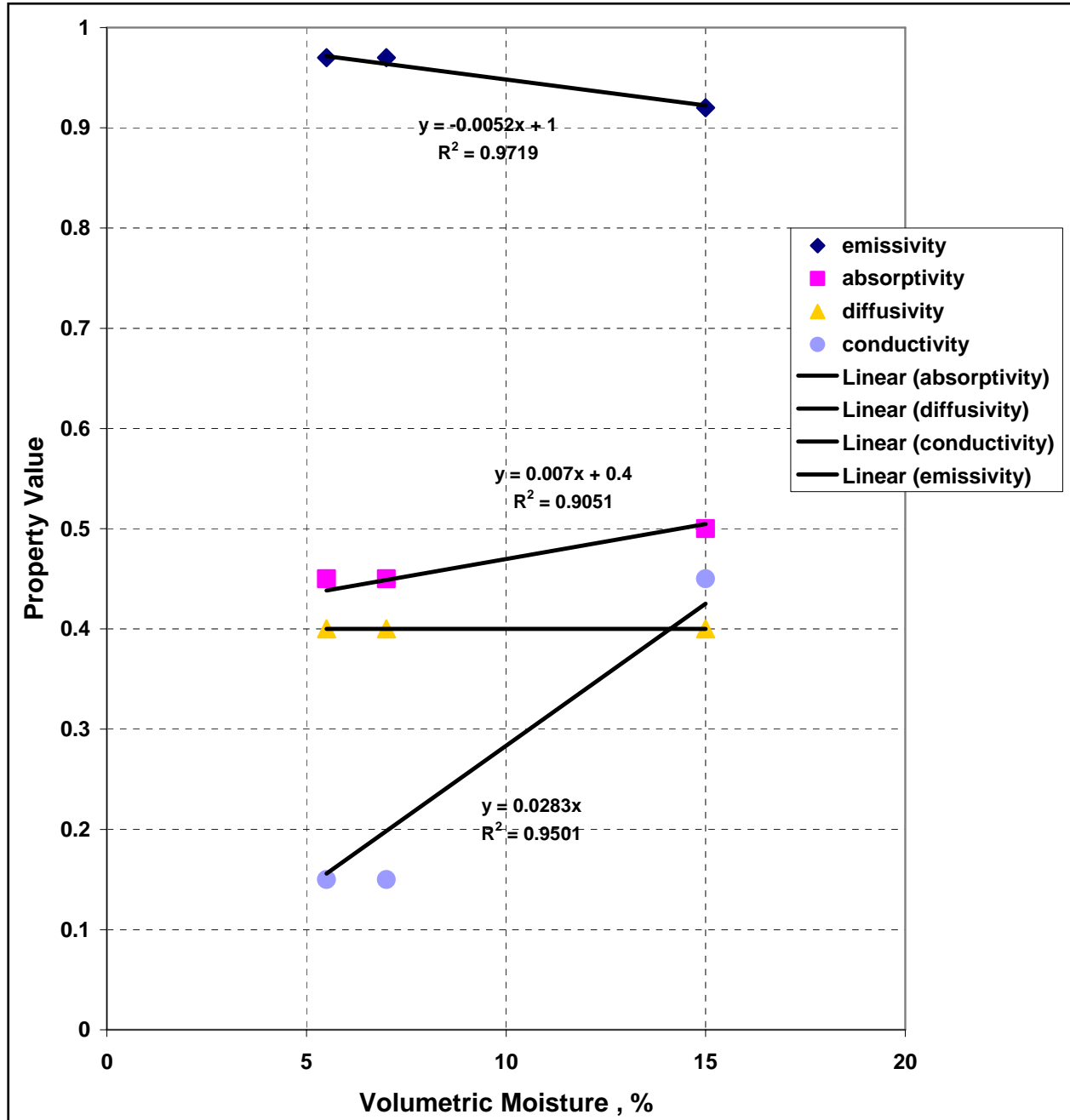


Figure 3. Test site thermal and optical properties of soil as a function of volumetric moisture content

## A Simulation of Soil Surface Temperatures

One goal of this study is to demonstrate that the modified TSTM/VEGIE code can do a reasonably good job of predicting surface temperatures under a variety of weather conditions. Weather data, surface temperature data, and soil moisture data collected at a midwestern U.S. test site during the summer of 2004 were used as input and validation data for two 64-day simulations of a bare soil surface and a metallic land mine surface. In a later section, these simulation results will be used to calculate a predicted thermal contrast between the mine surface and the soil surface.

All of the following charts are self-explanatory, but some commentary may be useful in interpreting the results. Simulation results for any single day could have been chosen for display. In this case, day 193 (one of the hotter, drier days) was chosen, and those results are shown in Figures 4 and 5. Obviously, one can never expect perfect simulation results; however, predicted surface temperatures compare very well with measured data for this day. Predicted results appear to lag the daytime data in a manner that results in predicted morning temperatures that are as much as 4 °C lower than measured data and early evening temperatures that are as much as 5 °C higher than the measured values. The material properties could have been adjusted for this one day to give much smaller differences between predictions and measurements, but that would violate the spirit of this exercise, which was to demonstrate physically sound simulations over a variety of weather conditions using one set of material property definitions.

Figure 6 compares the predicted surface temperature with the measured values for all 64 days of the simulation. In general, the results are quite reasonable, except for extremely wet and overcast conditions. One of the model input parameters that was not measured and used properly was that of percent cloud cover. Those data would have had an impact on simulation results through the long-wave irradiance term in the surface energy budget equation (Equation 2). In addition, there is still some question as to whether or not the latent heat exchange and sensible heat exchange formulations are being used properly. That question remains to be answered through future research.

Figure 7 summarizes the differences between the predicted surface temperatures and the measured data. Although the results appear somewhat noisy, note that the average difference is only 1.1 °C and that the standard deviation of the differences over the entire 64-day period is only 3.7 °C.

The final results shown (Figure 8) for this simulation are the soil temperature profiles for the even hours of day 193. Note that, even though three different layers of soil with the same thermal properties but different node spacings were used for this simulation, the resulting soil temperature profiles are very smooth and clearly show physically correct results such as the thermal inertia of the underlying soil (there is a time lag for temperature change beneath the surface). It would also appear that the assumption of a fixed bottom boundary temperature of 25 °C is not unreasonable, although in hindsight, a fixed temperature of 30 °C, or so,

might have been a better choice. Further note that the zone of active temperature fluctuations is limited to a depth of about 40 cm.

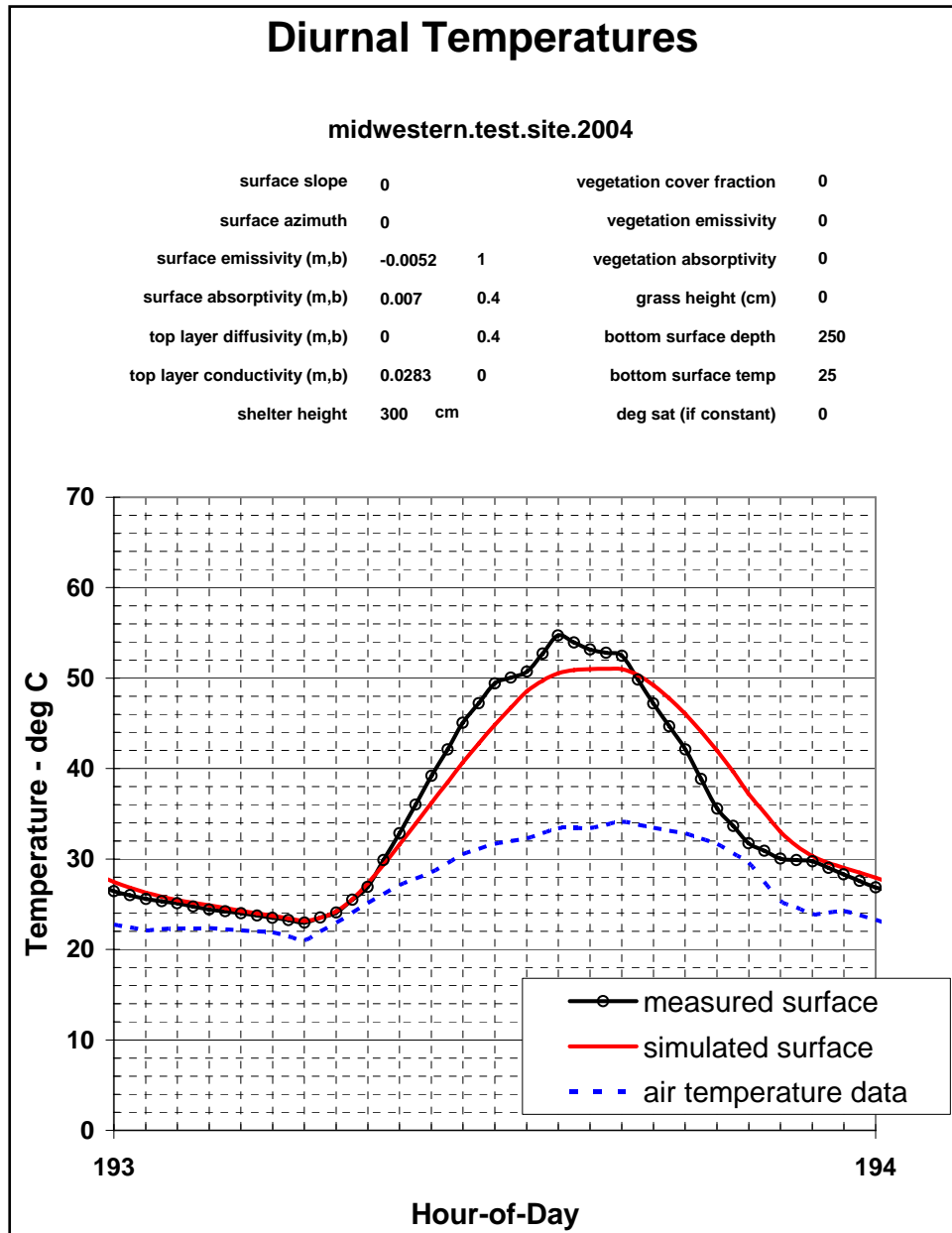


Figure 4. Predicted and measured soil temperatures for the midwestern U.S. test site (day 193)

## A Simulation of Land Mine Surface Temperatures

A second 64-day simulation was performed for a metallic landmine sitting on soil at this test site (see report cover photograph). The geometry of this mine was approximated by a three-layer structure consisting of a 0.33-cm-thick carbon steel jacket filled with dense concrete. The overall thickness was taken to be

about 10 cm. The latent heat exchange term in Equation 2 was turned off for this simulation by setting the saturation factor value at zero for all time-steps. Under those conditions, thermal and optical properties for the painted steel and thermal properties for the concrete were chosen to best predict the measured surface temperatures for days 193 and 212 (extremes in weather conditions). Results are shown in the following charts.

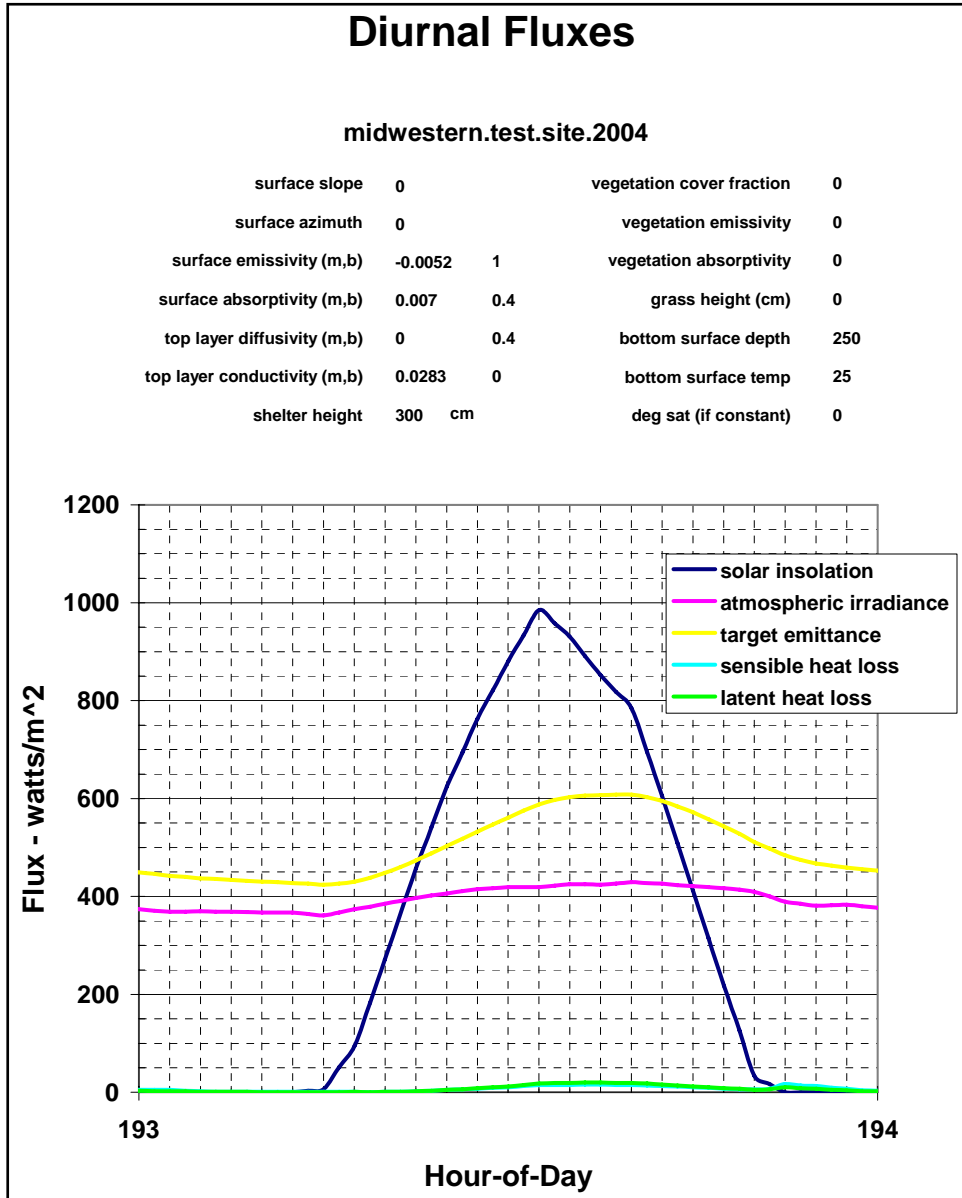


Figure 5. Predicted and measured energy fluxes for the midwestern U.S. test site soil simulation (day 193)

It appears that the land mine simulation was about as good as the soil simulation, with an average difference of  $-1.1\text{ }^{\circ}\text{C}$  and a standard deviation of  $4.9\text{ }^{\circ}\text{C}$ . The most noteworthy result is that of the temperature profiles shown in Figure 13. The profiles within the underlying soil (below 10 cm) have a much different character than those generated for the soil simulation (Figure 8). The

temperature of the soil in contact with the mine is forced to track the temperature of the bottom of the mine. The thermal inertia displayed in Figure 8 is not as evident on this chart. What the bottom boundary temperature should be is certainly left to speculation, and the zone of influence extends a little farther into the underlying soil.

## Landmine-Soil Thermal Contrasts

Of paramount importance to the airborne sensor community is the temperature contrast between man-made targets, such as the land mine, and background materials, such as the soil. Thermal infrared sensors that image the apparent temperatures of objects within their fields of view can easily detect large differences between targets and backgrounds. During the daylight hours, those contrasts can be large and positive (the target is hotter than the background). On the other hand, man-made materials can be cooler than their natural surroundings at night, resulting in a negative thermal contrast between the two.

Figure 13 shows both the predicted and measured thermal contrasts between the land mine and the surrounding soil for only 2 days during the 64-day simulation. In both cases, the positive contrasts are much greater than the negative contrasts, although the simulations produce greater extremes. What is most interesting occurs at the “cross-over times.” Those are the times of day when the contrast polarity switches from positive to negative, or vice-versa. While most IR sensors have the ability to detect fairly small differences in temperature between two objects, such differences can easily be washed out in the array of contrasts at points surrounding the target. Therefore, one wants to avoid using such a sensor to search for targets at times near the cross-overs. For days 193 and 194, the morning cross-over occurs between 8:00 AM and 9:00 AM, while the evening crossovers take place between 8:00 PM and 9:00 PM. For another season of the year, these cross-over times will probably occur at different times. It is noteworthy that the predicted cross-over times take place within an hour of the measured results.

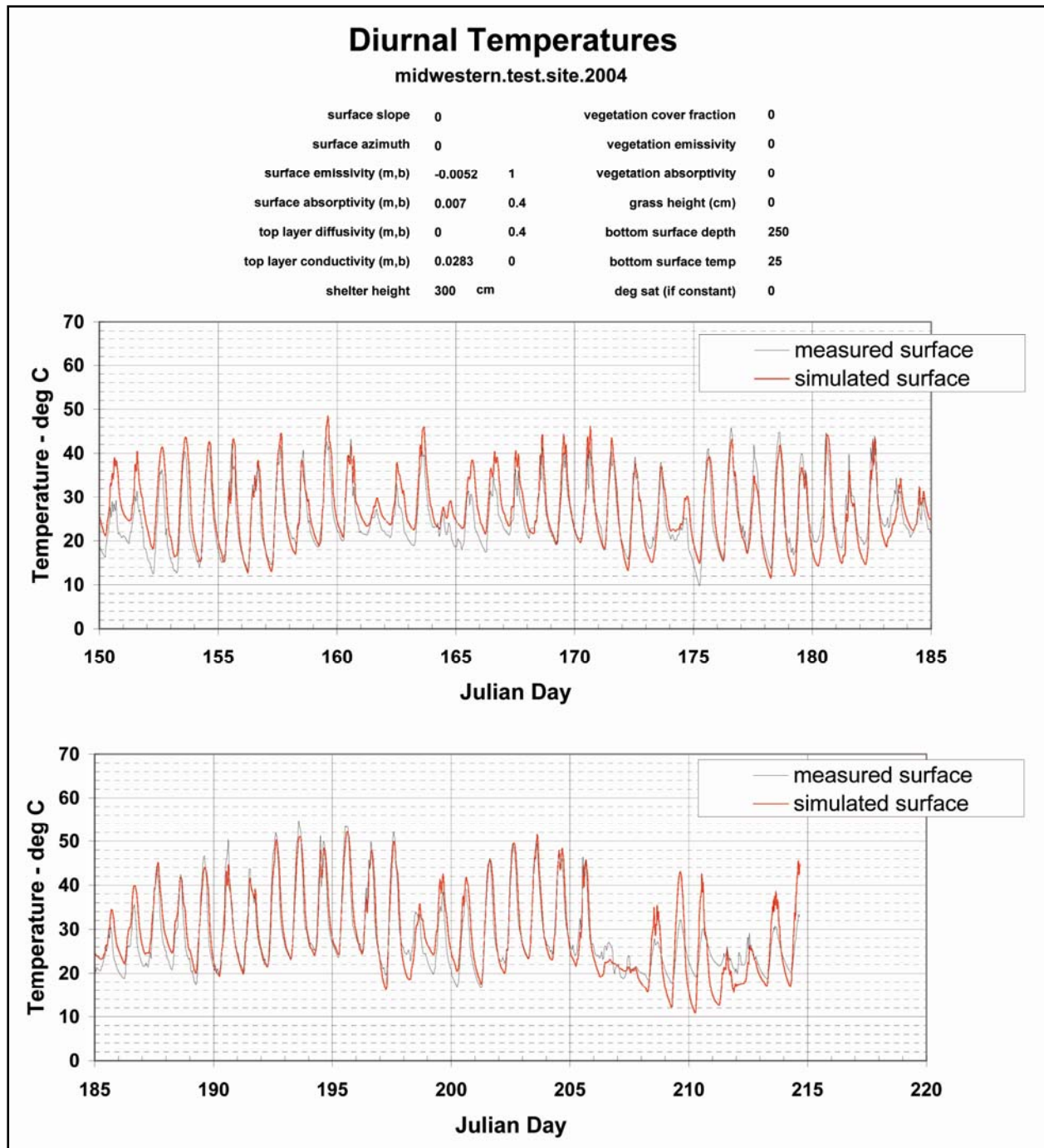


Figure 6. Predicted and measured soil temperatures for the midwestern U.S. test site (all days)

# Predicted - Measured Temperatures

midwestern.test.site.2004

surface slope	0	vegetation cover fraction	0	
surface azimuth	0	vegetation emissivity	0	
surface emissivity (m,b)	-0.0052	1	vegetation absorptivity	0
surface absorptivity (m,b)	0.007	0.4	grass height (cm)	0
top layer diffusivity (m,b)	0	0.4	bottom surface depth	250
top layer conductivity (m,b)	0.0283	0	bottom surface temp	25
shelter height	300	cm	deg sat (if constant)	0
average difference	1.117	population standard deviation	3.71	

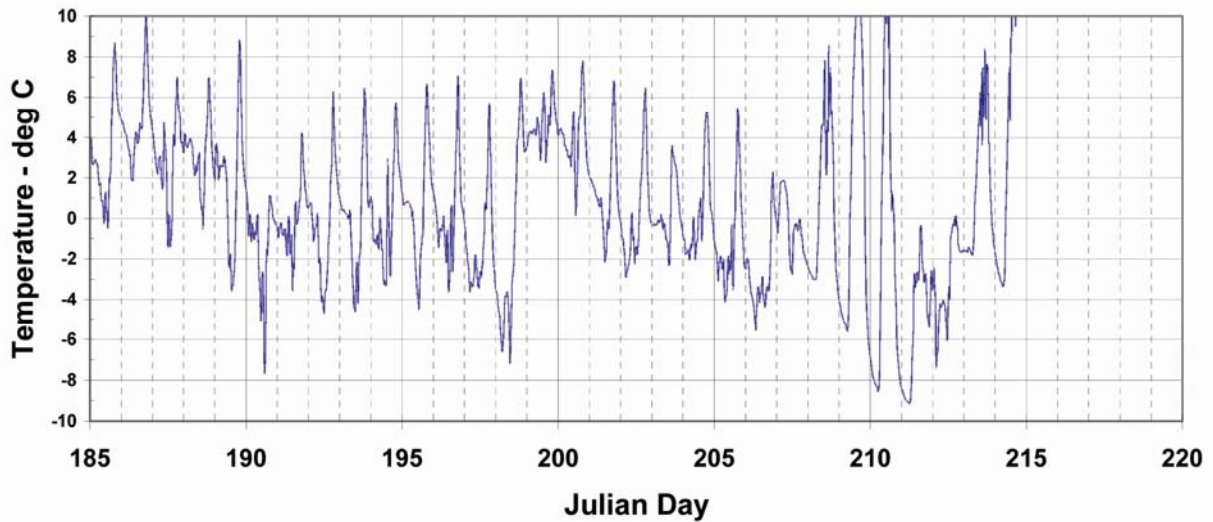
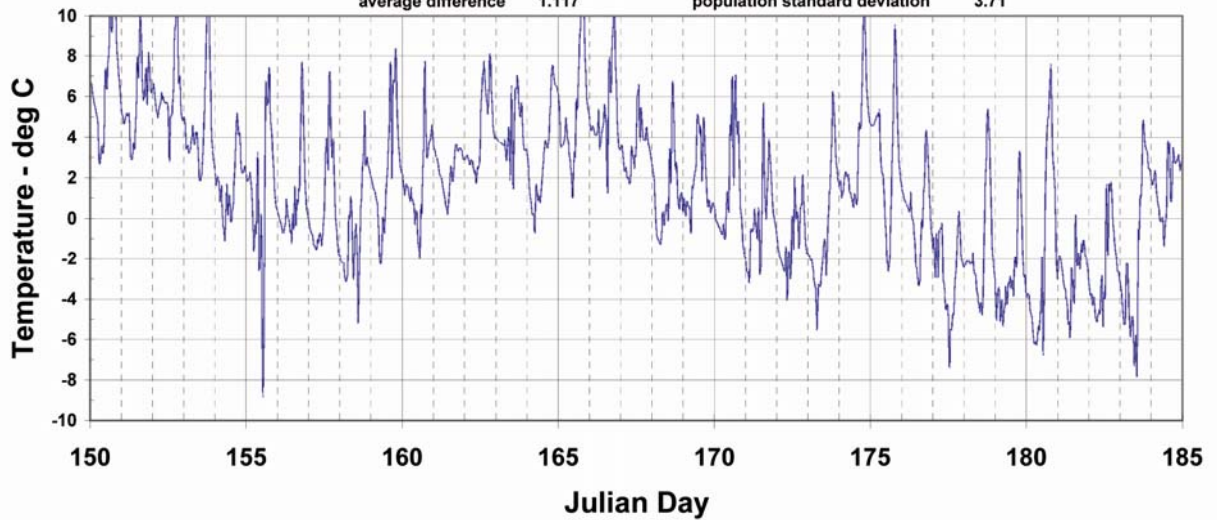


Figure 7. Differences between predicted and measured soil temperatures for the midwestern U.S. test site (all days)

# Day 193 Temperature Profiles

midwestern.test.site.2004

surface slope	0	vegetation cover fraction	0
surface azimuth	0	vegetation emissivity	0
surface emissivity (m,b)	-0.0052 1	vegetation absorptivity	0
surface absorptivity (m,b)	0.007 0.4	grass height (cm)	0
top layer diffusivity (m,b)	0 0.4	bottom surface depth	250
top layer conductivity (m,b)	0.0283 0	bottom surface temp	25
shelter height	300 cm	deg sat (if constant)	0

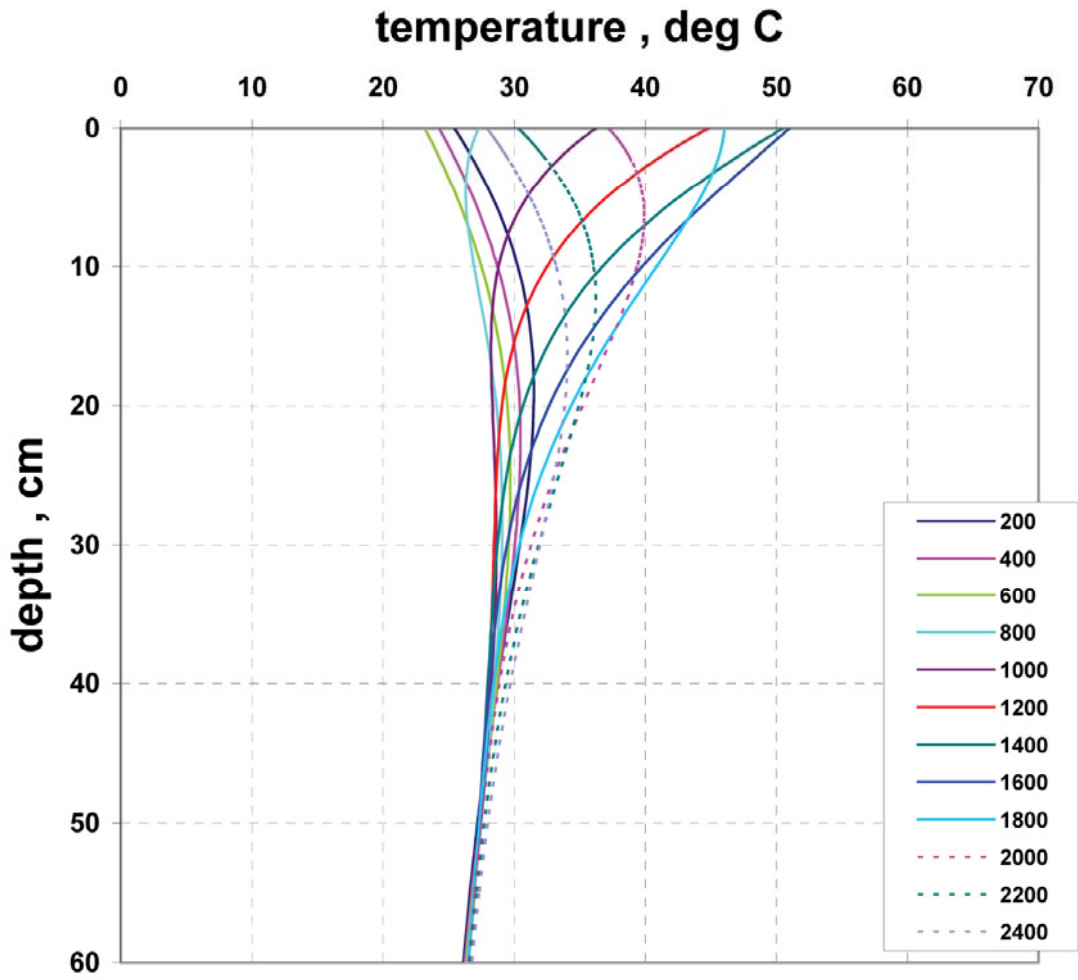


Figure 8. Predicted soil temperature profiles for the midwestern U.S. test site (day 193)

# Diurnal Temperatures

midwestern.test.site.2004

surface slope	0	vegetation cover fraction	0	
surface azimuth	0	vegetation emissivity	0	
surface emissivity (m,b)	0	0.93	vegetation absorptivity	0
surface absorptivity (m,b)	0	0.4	grass height (cm)	0
top layer diffusivity (m,b)	0	7	bottom surface depth	238.6
top layer conductivity (m,b)	0	7	bottom surface temp	25
shelter height	300	deg sat (if constant)	0	

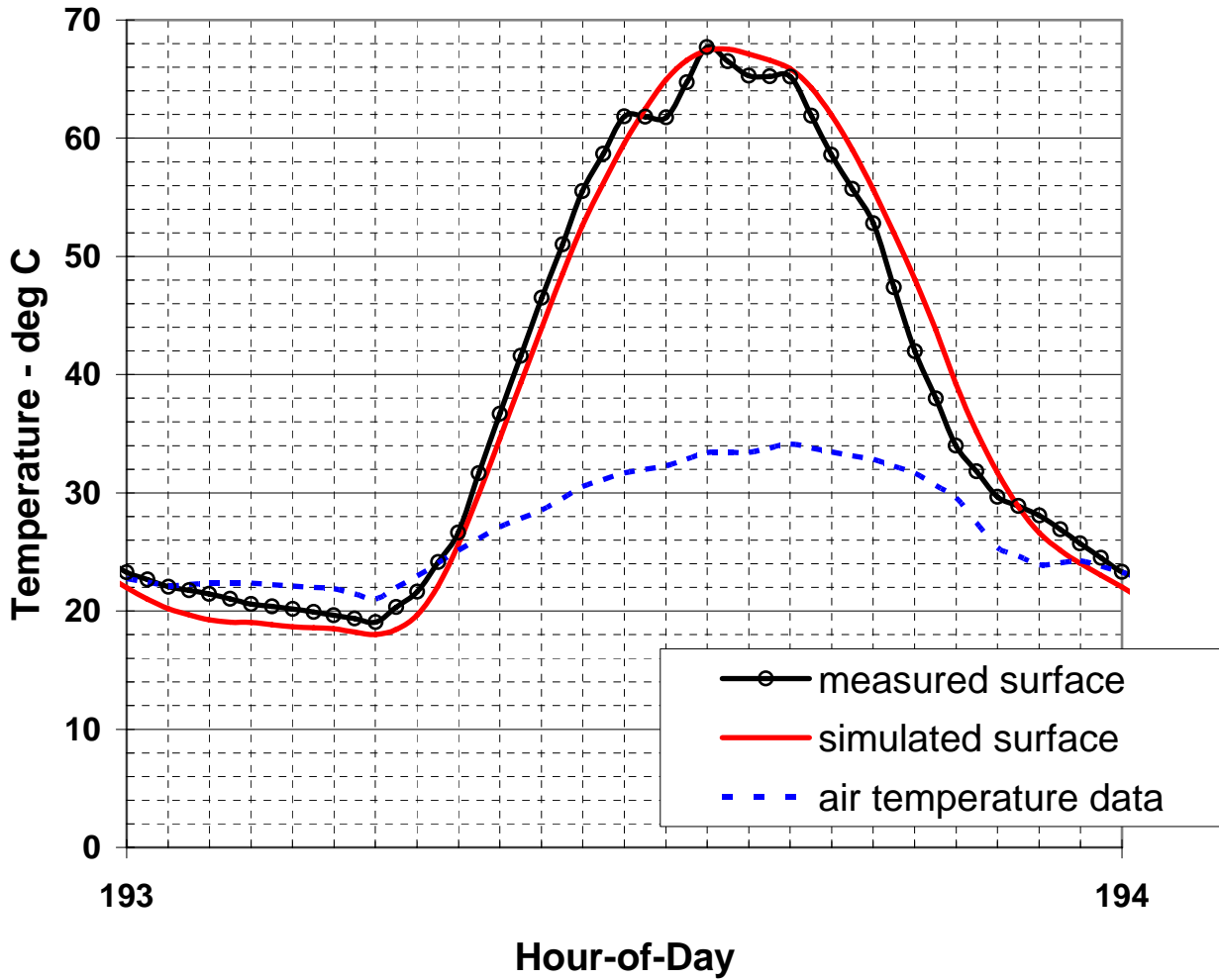


Figure 9. Predicted and measured land mine temperatures for the midwestern U.S. test site (day 193)

# Diurnal Fluxes

midwestern.test.site.2004

surface slope	0	vegetation cover fraction	0	
surface azimuth	0	vegetation emissivity	0	
surface emissivity (m,b)	0	0.93	vegetation absorptivity	0
surface absorptivity (m,b)	0	0.4	grass height (cm)	0
top layer diffusivity (m,b)	0	7	bottom surface depth	238.6
top layer conductivity (m,b)	0	7	bottom surface temp	25
shelter height	300	deg sat (if constant)	0	

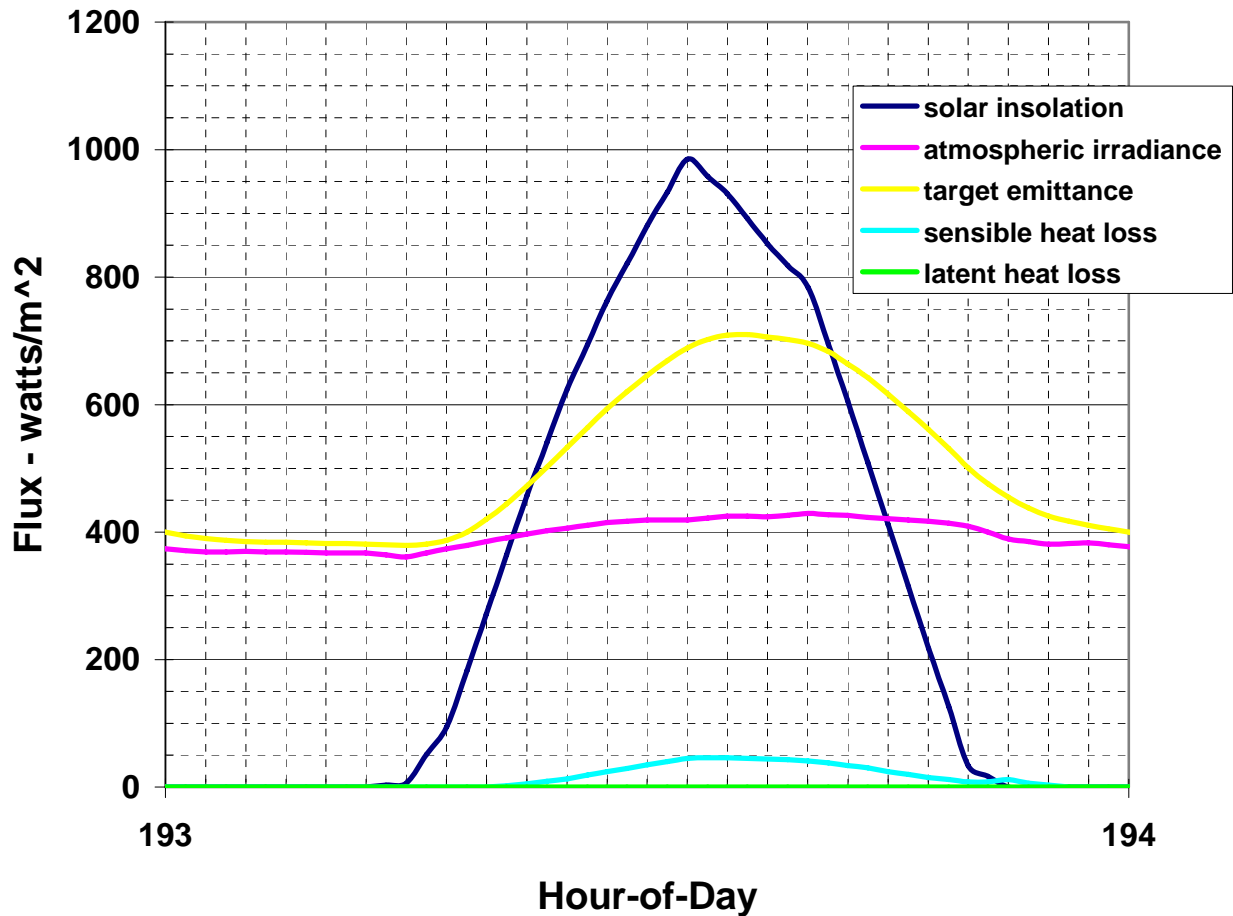


Figure 10. Predicted and measured energy fluxes for the midwestern U.S. test site land mine simulation (day 193)

# Diurnal Temperatures

midwestern.test.site.2004

surface slope	0	vegetation cover fraction	0	
surface azimuth	0	vegetation emissivity	0	
surface emissivity (m,b)	0	0.93	vegetation absorptivity	0
surface absorptivity (m,b)	0	0.4	grass height (cm)	0
top layer diffusivity (m,b)	0	7	bottom surface depth	238.6
top layer conductivity (m,b)	0	7	bottom surface temp	25
shelter height	300	deg sat (if constant)	0	

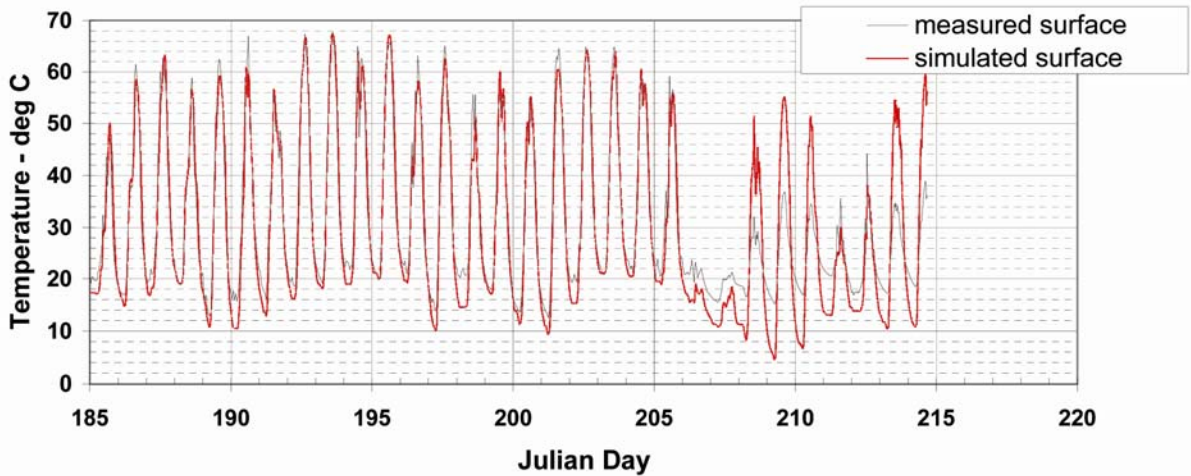
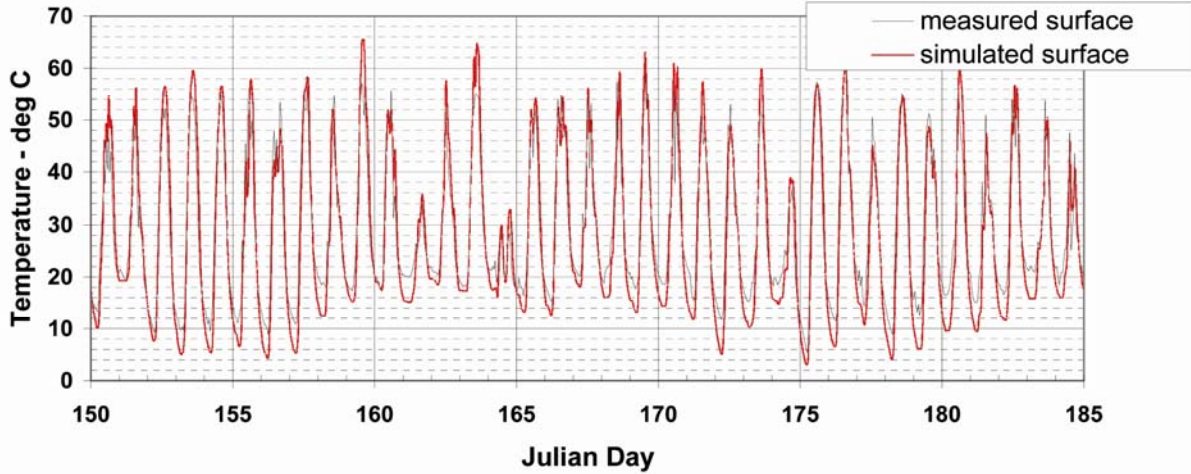


Figure 11. Predicted and measured land mine temperatures for the midwestern U.S. test site (all days)

# Predicted - Measured Temperatures

midwestern.test.site.2004

surface slope	0	vegetation cover fraction	0	
surface azimuth	0	vegetation emissivity	0	
surface emissivity (m,b)	0	0.93	vegetation absorptivity	0
surface absorptivity (m,b)	0	0.4	grass height (cm)	0
top layer diffusivity (m,b)	0	7	bottom surface depth	238.6
top layer conductivity (m,b)	0	7	bottom surface temp	25
shelter height	300	deg sat (if constant)	0	
average difference	-1.088	cm	population standard deviation	4.885

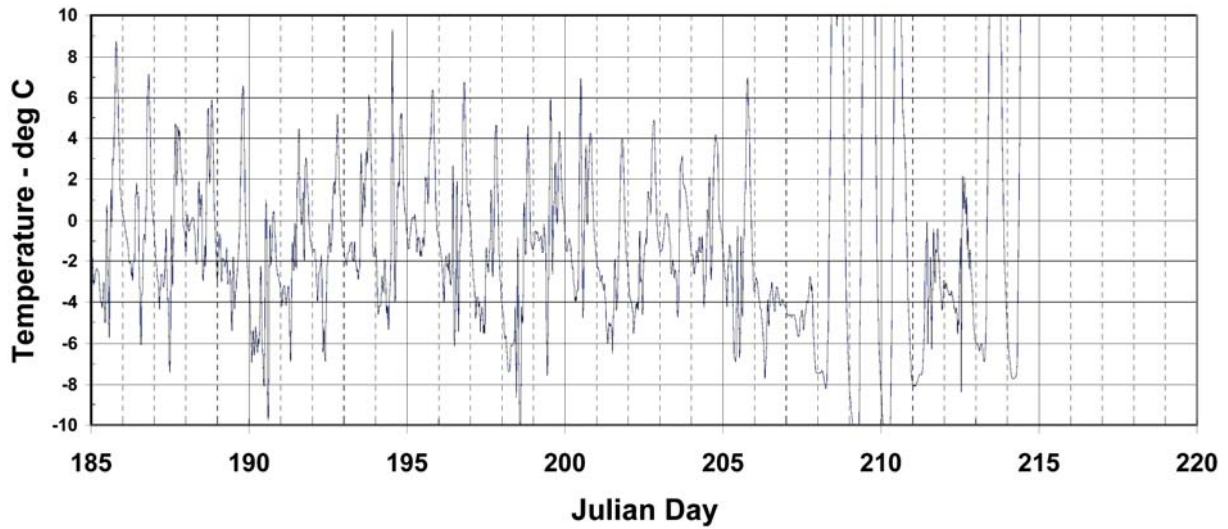
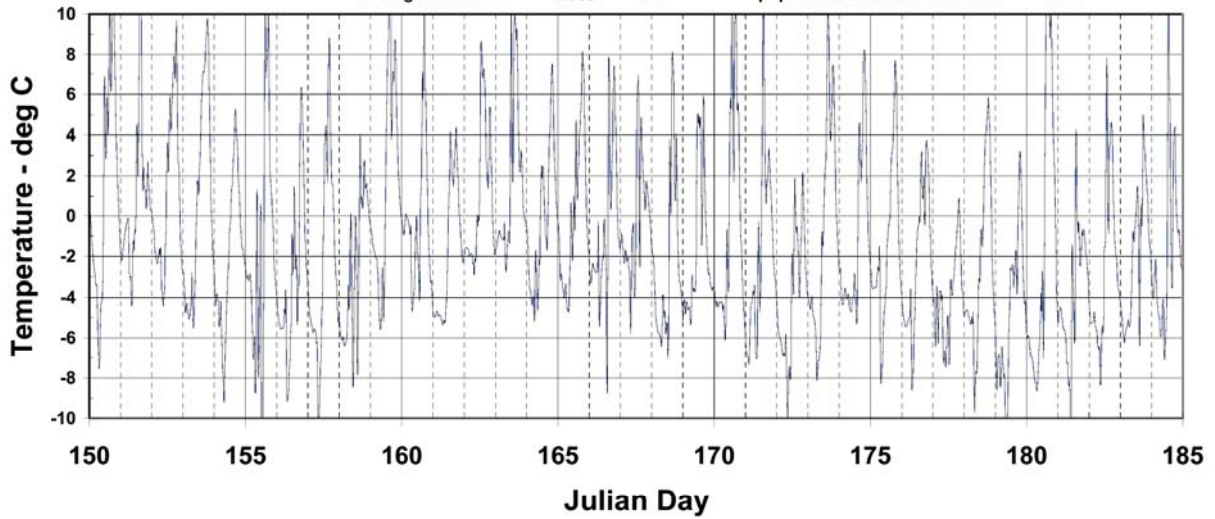


Figure 12. Differences between predicted and measured land mine temperatures for the midwestern U.S. test site (all days) (Continued)

# Day 193 Temperature Profiles

midwestern.test.site.2004

surface slope	0	vegetation cover fraction	0	
surface azimuth	0	vegetation emissivity	0	
surface emissivity (m,b)	0	0.93	vegetation absorptivity	0
surface absorptivity (m,b)	0	0.4	grass height (cm)	0
top layer diffusivity (m,b)	0	7	bottom surface depth	238.6
top layer conductivity (m,b)	0	7	bottom surface temp	25
shelter height	300	deg sat (if constant)	0	

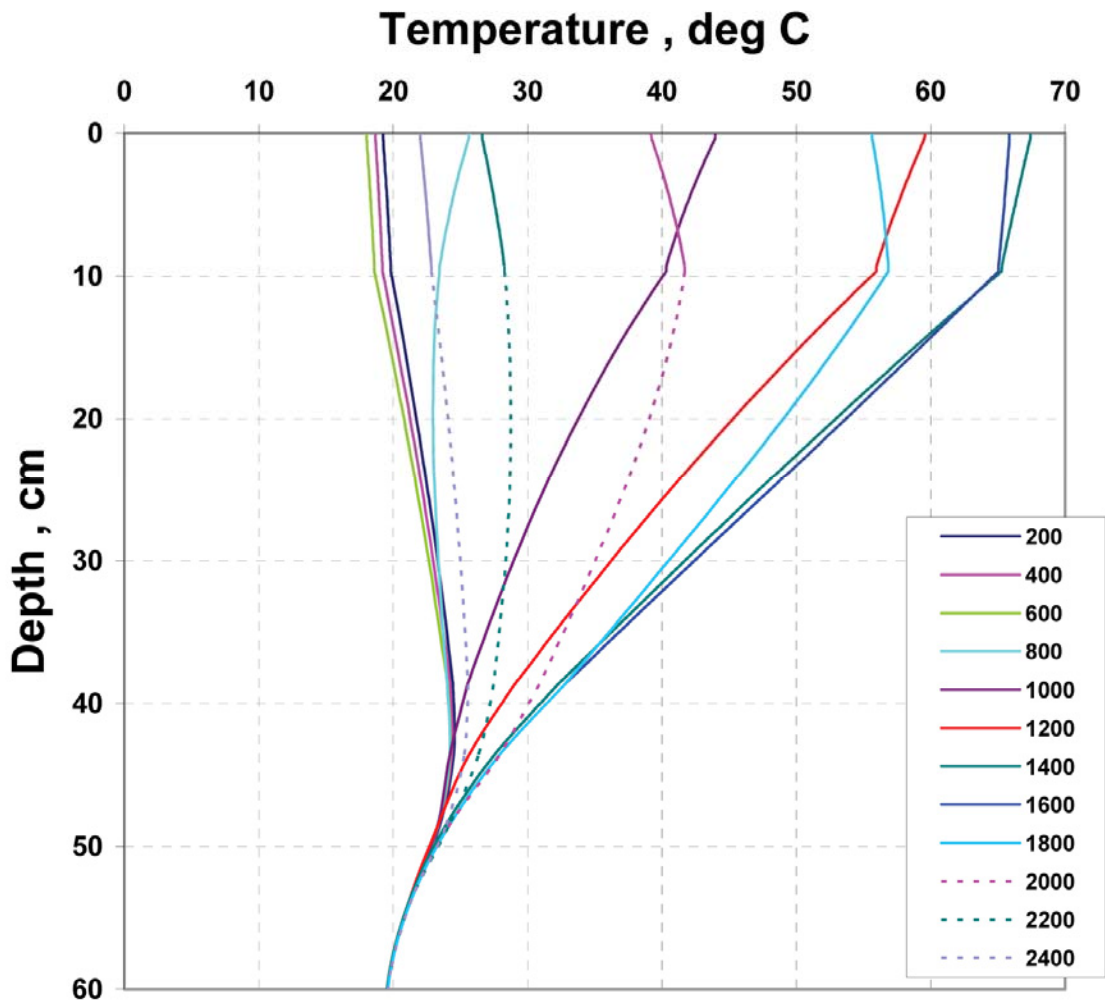


Figure 13. Predicted land mine (over soil) temperature profiles for the midwestern U.S. test site (day 193)

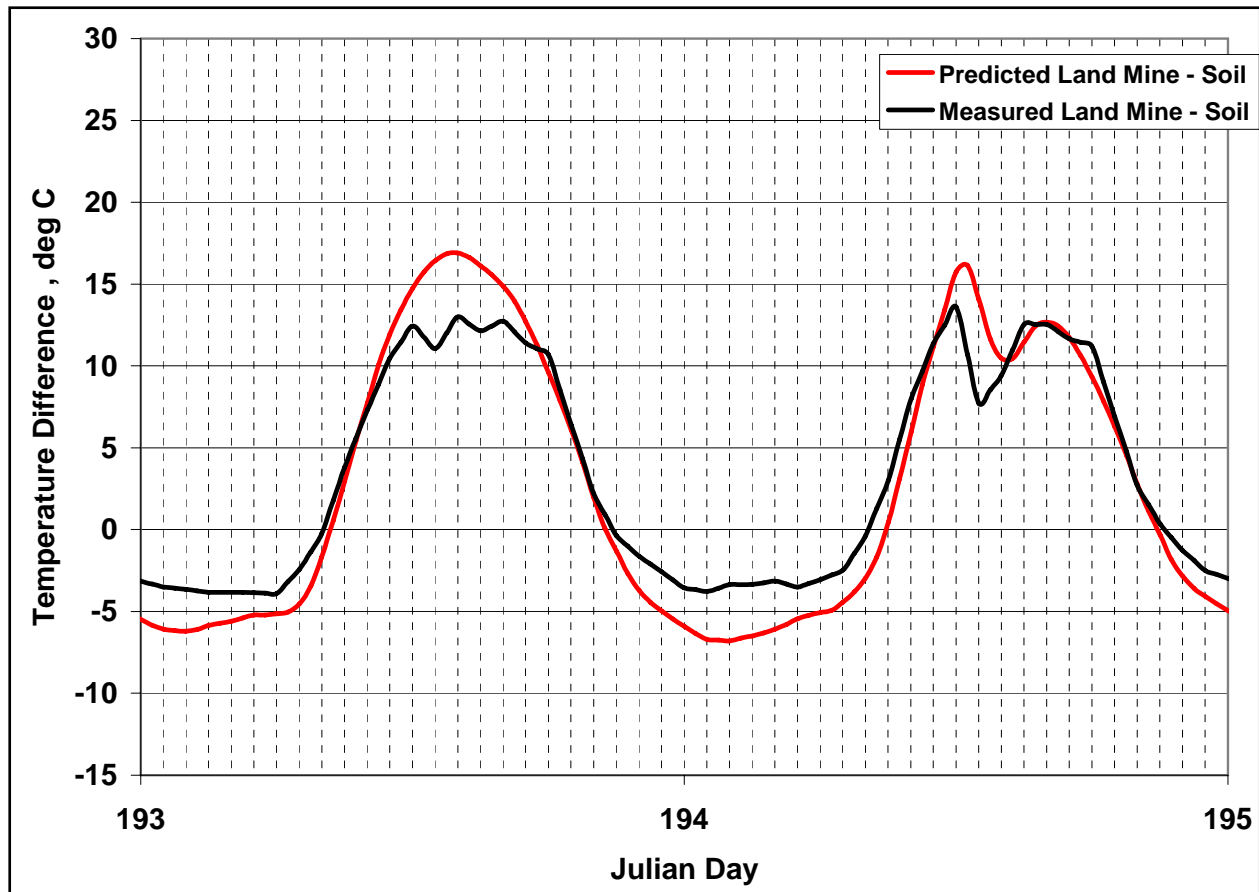


Figure 14. Predicted and measured land mine/soil thermal contrasts (days 193-194) for the midwestern U.S. test site

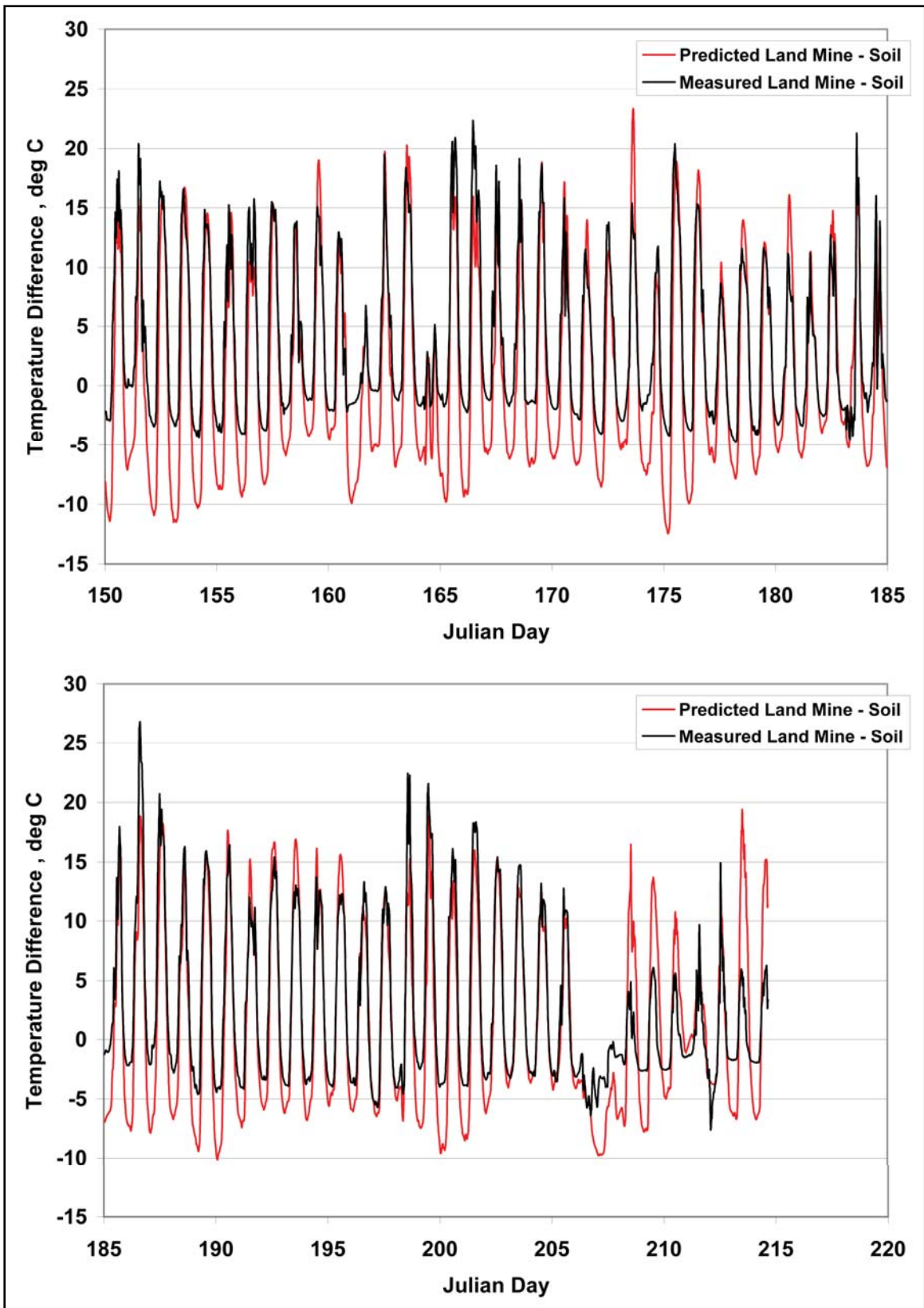


Figure 15. Predicted and measured land mine/soil thermal contrasts (all days) for the midwestern U.S test site

# 5 Summary and Recommendations

---

## Summary

An existing 1-D finite difference surface temperature prediction model was modified to perform multiple-day simulations of natural terrain and man-made surfaces using weather station data as the primary input. The original code was modified in a number of ways, including the ability to exercise the code on a PC, to input thermal and optical material properties that vary with moisture content, and to utilize more realistic latent and sensible heat exchange models. An Excel spreadsheet was developed to help display simulation results in meaningful ways.

The modified code was exercised against 64 days of weather, soil moisture, and surface temperature data collected at a midwestern U.S. test site. Comparisons of measured data with predicted results for bare soil and a metallic land mine were quite favorable. Of particular interest were the daily displays of thermal contrast between the land mine and the bare soil. While the predicted magnitudes of peak thermal contrasts exceeded the measured values, crossover times (times of day when the thermal contrast goes to zero) for both the real data and the simulations were within an hour of each other.

It is anticipated that this code can make reasonable, physics-based predictions for man-made target surface temperatures as well as those of background materials for any test site in the world. Such simulations can be used to anticipate times of day during which airborne sensors can be expected to have difficulty in detecting targets against natural backgrounds. As this code only computes physical temperatures of the surfaces, it cannot be expected to predict sensor performance. That would be a function of the sensitivity of the sensor detectors, the angular resolution of the sensor, and the algorithms that might be used to process measured data.

Furthermore, it must also be recognized that this code cannot simulate two- and three-dimensional effects, which certainly must play a significant part in determining the temperature distribution around a finite three-dimensional target in a large heterogeneous background. Such studies will require a computational test bed that operates on a much larger computer system than the author's PC.

## Recommendations

Several improvements could be made to the code and analysis procedures that would create a more user-friendly environment in which to perform simulations. For example, as the original authors of the TSTM/VEGIE simulation code suggested, one could make the code be implicit in nature, in which the temperatures on the right side of Equation 9 would be written in terms of values at the end of the time-step. The resulting iterative solution would ensure time-step stability.

Another improvement would be to develop an optimization routine for determining material properties. The current process for finding an optimum set of surface material thermal and optical properties is best performed by varying each of the four property values in question (long-wave emissivity, short-wave absorptivity, thermal diffusivity, and thermal conductivity) one at a time and comparing the predicted surface temperatures to measured values. It requires a lot of analyst judgment and the use of the rules of thumb shown in Table 1 in the text. It should be possible to automate that search for an optimum set of values, perhaps using the standard deviation of the difference between the predicted temperatures and measured temperatures as an optimization metric.

# References

---

- Balick, L. K., Link, L. E., Scoggins, R. K, and Solomon, J. L. (1981a). "Thermal modeling of terrain surface elements," Technical Report EL-81-2, U.S. Army Engineer Waterways Experiment Station, Vicksburg, MS.
- Balick, L. K., Scoggins, R. K., and Link, L. E. (1981b). "Inclusion of a simple vegetation layer in terrain temperature models for thermal infrared (IR) signature prediction," Miscellaneous Paper EL-81-4, U.S. Army Engineer Waterways Experiment Station, Vicksburg, MS.
- Holman, J. P. (1968). *Heat Transfer*, 2nd ed., McGraw-Hill Book Company, New York.
- Jumikis, A. R. (1977). *Thermal Geotechnics*, Rutgers University Press, New Brunswick, NJ.
- Ochsner, T. E., Horton, R., and Ren, T. (2001). "A new perspective on soil thermal properties," *Soil Science Society of America Journal* 65, 1641-1647.
- Ohanian, H. C. (1985). *Physics*, W.W. Norton and Company, New York.
- Oke, T. R. (1987). *Boundary Layer Climates*, 2nd ed., Routledge, New York.
- Welsh, J. P. (1994). "Smart weapons operability enhancement (SWOE) joint test and evaluation (JT&E) program, Final report," SWOE Report 94-10, U.S. Army Engineer Cold Regions Research and Engineering Laboratory, Hanover, NH.

# Appendix A

## Code Listing

---

```
1 C
2 C TSTM:VEGIE----- BARE OR VEGETATED SURFACE TEMPERATURE MODEL
3 C-----
4 C
5 C THE FUNCTION OF THIS PROGRAM IS TO PREDICT THE PHYSICAL
6 C TEMPERATURES OF SURFACES EXPOSED TO VARIOUS WEATHER
7 C CONDITIONS
8 C
9 C IT IS A 1-D FINITE DIFFERENCE CODE CURRENTLY LIMITED TO
10 C 6 LAYERS OF MATERIAL AND 150 NODES
11 C
12 C-----
13 C
14 C MAJOR REVISIONS IN SEPT-DEC 2004 BY JOHN CURTIS
15 C REVISIONS INCLUDE MET STATION WEATHER FILES AS INPUT FOR
16 C MULTIPLE-DAY SIMULATIONS AS WELL AS UTILIZING MOISTURE-
17 C DEPENDENT SOIL THERMAL PROPERTIES AND OPTICAL PROPERTIES
18 C
19 C MORE REVISIONS IN JUNE 2005 TO MAKE INPUT COMPATIBLE WITH
20 C A GRAPHICAL USER INTERFACE DEVELOPED BY CURTIS
21 C
22 C-----
23 C PARAMETER (ND=30)
24 C DIMENSION THK(6),DEPTH(150),PROF(2,150),
25 C & XXX(30),YYY(30),LNUM(150)
26 C DIMENSION JD(15000),DT(15000),ATEMP(15000),RELHUM(15000),
27 C & BPRESS(15000),SOLAR(15000),WINDSP(15000),CLDTYPE(15000),
28 C & CLDCOV(15000),DEGSAT(15000),VMOIS1(15000),VMOIS2(15000),
29 C & STEMP1(15000),STEMP2(15000),RTEMP1(15000),RTEMP2(15000)
30 C
31 C ARRAY FOR HOURLY TEMPERATURE PROFILES FOR DAY NSNGLDAY
32 C
33 C DIMENSION TEMPPROF(151,26)
34 C
35 C DIMENSION ALPHM(6),ALPHB(6),SFRQ(6),FKM(6),FKB(6)
36 C DIMENSION TITLE(7),CLABEL(17)
37 C DIMENSION CLR(8),NX(6),ATF(2),FEB(2)
38 C DIMENSION STOR(8,150),RR(6),INTR(7)
39 C REAL KTEMPG,KTEMPA,KTEMPT,LAT,ACL(8),BCL(8),M,KSQ
40 C CHARACTER DATE*8,HEADER*72,AN*1,CLABEL*10
```

```

41 CHARACTER*30 FNAME
42 DATA CLR/0.04,0.08,0.17,0.20,0.22,0.24,0.24,0.25/
43 DATA ACL/82.2,87.1,52.5,39.0,34.7,23.8,11.2,15.4/
44 DATA BCL/.079,.148,.112,.063,.104,.159,-.167,.028/
45 DATA SIGMA,PI,AC,BC/8.12E-11,3.141593,
46 & 17.269,35.86/
47 DATA CC/0.261/
48 DATA LAST,G,KSQ,CP/24,980.0,0.16,0.24/
49 C-----
50 C FORMAT STATEMENTS
51 C
52 90 FORMAT(' botm bndry index=',I3)
53 92 FORMAT(' botm bndry temp=',F6.1,' deg_C')
54 95 FORMAT(' botm bndry heat flux=',F6.1,'cal/cm**2-min')
55 97 FORMAT(5F8.2)
56 120 FORMAT(A8,F6.1)
57 139 FORMAT(1H\|)
58 140 FORMAT(F5.1,F10.1,F6.1,F12.1,F12.1,F13.1)
59 145 FORMAT(F6.1,F7.1)
60 150 FORMAT(F12.1,I12,F11.2)
61 160 FORMAT(3F8.2)
62 170 FORMAT(F9.4,F12.1)
63 180 FORMAT(I7,F6.1,F11.4,F8.2)
64 190 FORMAT(4X,F5.2,F6.1,2X,I8,I9,I11,I9,I9,I8)
65 195 FORMAT(4X,F5.2,1H; ,F6.1,2X,1H; ,I8,
66 & 1H; ,I9,1H; ,I11,1H; ,I9,1H; ,I9,1H; ,I8)
67 200 FORMAT(4F10.4)
68 210 FORMAT(I4,F9.2,F8.2,4F10.4)
69 220 FORMAT(1H1)
70 230 FORMAT(A72)
71 235 FORMAT(' THIS.IS.NOT.A.SINGLE-DAY.SIMULATION'/I5,F5.2/)
72 236 FORMAT(' THIS.IS.A.SINGLE-DAY.SIMULATION.FOR.DAY'/I5,F5.2/)
73 240 FORMAT(4X,F7.2,3X,F5.1,5X,F6.2,10X,F6.2,9X,F7.2)
74 250 FORMAT(6X,F6.1,13X,F4.1,12X,F5.1)
75 260 FORMAT(9X,F8.1,F11.2,F9.2,F9.2,F8.2,F10.2,F8.2)
76 310 FORMAT(1H;,'tot grybdy efectv ground foliage solar')
77 320 FORMAT(1H;,'radnce temp temp temp insol')
78 330 FORMAT('hr (W/m**2) (C) (C) (C) (W/m**2)')
79 340 FORMAT(9X,'----refl-nrefl----refl----nrefl',30(1H-))
80 270 FORMAT(3X,F5.2,5X,I4,2X,I4,3X,F6.1,2X,F6.1,3X,F6.1,3X,F6.1,7X,I4)
81 275 FORMAT(3X,F5.2,1H; ,5X,1H; ,I4,2X,1H; ,
82 & I4,3X,1H; ,F6.1,2X,1H; ,1H; ,F6.1,3X,1H; ,F6.1,3X,1H; ,F6.1,7X,
83 & 1H; ,I4)
84 350 FORMAT(' . . . . . sensbl latent')
85 360 FORMAT(' jday hr jd&hr air surf grybdy solar surf
86 &atms_ir heat heat rad1 rad2 surf-r1 surf-r2')
87 370 FORMAT(' . . . temp temp radnce insol absorp emissn loss loss
88 & temp temp temp temp')
89 380 FORMAT(' . . . deg_C . ',23(1H-),'(W/m**2)',24(1H-))
90 400 FORMAT(2H0 ,F5.2,4(3X,F5.1,' ',F5.2))
91 410 FORMAT(10X,F5.1,1X,F5.2,3X,F5.1,1X,F5.2,3X,F5.1,1X,F5.2,
92 & 3X,F5.1,1X,F5.2)
93 2610 FORMAT(I4,4F8.3,6I6,4F7.2)
94 C
95 C STATEMENT FUNCTIONS FOR USE IN VEGETATION SECTION
96 C

```

```

97      E(T)=RH*6.108*EXP(AC*(T-273.15)/(T-BC))
98      ESAT(T)=6.108*EXP(AC*(T-273.15)/(T-BC))
99      Q(T)=0.622/(PRESS/E(T)-.378)
100     QSAT(T)=0.622/(PRESS/ESAT(T)-.378)
101     C
102     C      FUNCTION STATEMENT FOR ALL LINEAR INTERPOLATION NEEDS
103     C      THIS INCLUDES WEATHER PARAMETERS AS A FUNCTION OF TIME AND
104     C      MOISTURE CONTENT AS A FUNCTION OF DEPTH
105     C
106     C      YVALUE(XVALUE,Y1,Y2)=Y1+(XVALUE-X1)*(Y2-Y1)/(X2-X1)
107     C
108     C      OPEN(2,STATUS='UNKNOWN')
109     C      OPEN(4,STATUS='UNKNOWN')
110     C-----
111     C
112     C      ***** DATA INPUT *****
113     C
114     C-----
115     C
116     C      INITIALIZE-VARIABLES-AND-CONSTANTS
117     C
118     C      BB=-2.4E-4
119     C      IBUG=0
120     C      IEOF=0
121     C      DO 100 I=1,6
122     C      THK(I)=0.
123     C      SFRQ(I)=0.
124     C      ALPHM(I)=0.
125     C      ALPHB(I)=0.
126     C      FKM(I)=0.
127     C      100 FKB(I)=0.
128     C
129     C      DO 101 I=1,26
130     C      DO 101 J=1,151
131     C      101 TEMPPROF(J,I)=0
132     C      JPROF=2
133     C
134     C      DO 102 I=1,150
135     C      DEPTH(I)=0.
136     C      102 LNUM(I)=0
137     C
138     C      SIGF=0.
139     C      STATE=0.
140     C      EPF=0.
141     C      FOLA=0.
142     C      HFOL=0.
143     C-----
144     C
145     C      INPUT SIMULATION TITLE AND NO. OF FIELD WEATHER STATION
146     C      DATA LINES
147     C
148     C      READ(2,*)HEADER,NLDATA
149     C-----
150     C
151     C      INPUT FIELD WEATHER STATION DATA
152     C

```

```

153 C INPUT DATA LINES CONTAIN: JULIAN DAY, 24-HR CLOCK TIME,
154 C AIR TEMP, RELATIVE HUMIDITY, BAR PRESSURE,
155 C SOLAR LOADING, WIND SPEED, CLOUD INDEX, CLOUD COVER PERCENT,
156 C SURFACE DEGREE OF SATURATION (DECIMAL), MOISTURE CONTENT
157 (SHALLOW),
158 C MOISTURE CONTENT (DEEP), SOIL TEMPERATURE (SHALLOW),
159 C SOIL TEMPERATURE (DEEP), RADIOMETRIC TEMPERATURE 1,
160 C RADIOMETRIC TEMPERATURE 2
161 C
162 C IF NSOLAR=1 (A FLAG TO BE READ LATER), THEN THE CODE WILL
163 C GENERATE SOLAR INSOLATION VALUES FOR THE ENTIRE INPUT FILE
164 C
165 C DO 800 I=1,NLDATA
166 C READ(2,*)JD(I),HR24,ATEMP(I),RELHUM(I),BPRESS(I),SOLAR(I),
167 & WINDSP(I),CLDTYPE(I),CLDCOV(I),DEGSAT(I),VMOIS1(I),VMOIS2(I),
168 & STEMP1(I),STEMP2(I),RTEMP1(I),RTEMP2(I)
169 C
170 C A CORRECTION TO PREVENT THE RICHARDSON # FROM BLOWING UP
171 C IF(WINDSP(I).LT..1) WINDSP(I)=.1
172 C
173 C CALCULATE A JULIAN DAY DECIMAL TIME
174 C DT(I)=INT(HR24/100.)+((HR24-INT(HR24/100.)*100.)/60.
175 C
176 C CONVERSION OF TEMPERATURE TO DEG KELVIN
177 C ATEMP(I)=ATEMP(I)+273.15
178 C
179 C CONVERSION OF RELATIVE HUMIDITY TO A DECIMAL VALUE
180 C RELHUM(I)=RELHUM(I)/100.
181 C
182 C CONVERSION OF WIND SPEED TO CM/S
183 C WINDSP(I)=WINDSP(I)*100.
184 C
185 C CONVERSION OF SOLAR LOADING TO (SMALL CAL)/(MIN-CM^2)
186 C SOLAR(I)=SOLAR(I)/697.6
187 C
188 C CORRECTION FOR A BAD PRESSURE GUAGE
189 C IF(BPRESS(I).LT.0.) BPRESS(I)=1000.
190 800 CONTINUE
191 C
192 C SKIP THE LINE OF DATA CONTAINING "END"
193 C READ(2,*)DATE
194 C-----
195 C
196 C READ SINGLE-DAY SIMULATION PARAMETERS
197 C
198 C NSINGLE =0 IF DOING A MULTIPLE-DAY SIMULATION
199 C NSINGLE =1 IF DOING A SINGLE-DAY SIMULATION
200 C NSNGLDAY = THE JULIAN DAY CHOSEN FOR A SINGLE-DAY SIMULATION
201 C
202 C READ(2,*) NSINGLE,NSNGLDAY
203 C NFIRST=1
204 C
205 C IDENTIFY 1ST LINE OF DATA FOR A SINGLE-DAY SIMULATION
206 C
207 C IF(NSINGLE.EQ.0) GO TO 806
208 C DO 805 I=1,NLDATA

```

```

209         IF(JD(I).NE.NSNGLDAY) GO TO 805
210         NFIRST=I
211         GO TO 806
212     805  CONTINUE
213     C-----
214     C
215     C   INPUT SURFACE SLOPE INFO AND SOLAR CALCULATION FLAG
216     C
217     C   SFC SLOPE  SFC AZIMUTH  LATITUDE
218     C   DEG-HORIZ=0 DEG S=0  DEG
219     C
220     806  READ(2,*)NSOLAR,SLOPE,SURFAC,LAT
221         SLOPE=SLOPE*PI/180.0
222         SURFAC=SURFAC*PI/180.
223     C
224     C   COMPUTE SOLAR INSOLATION, IF NECESSARY
225     C
226         IF(NSOLAR.EQ.0) GO TO 88888
227     C
228     C   CALCULATE-INSOLATION-ON-SLOPE-SURFACE
229     C
230         DO 8888 I=1,NLDATA
231     C
232     C   SOLVE-SOLAR-ZENITH
233     C
234         TYME=DT(I)
235         DAY=JD(I)
236         NCLOUD=CLDTYPE(I)
237         PRESS=BPRESS(I)
238         T0=2.0*PI*(DAY-1.0)/365.0
239         DECL=0.006918-0.399912*COS(T0)+0.070257*SIN(T0)
240         &  -0.006758*COS(2.0*T0)+0.000907*SIN(2.0*T0)
241         &  -0.002697*COS(3.0*T0)+0.001480*SIN(3.0*T0)
242         ELF=(LAT/180*PI)
243         TIMER=(TYME/12*PI)+PI
244         IF(TIMER.GT.2.*PI)TIMER=TIMER-2.*PI
245         AA=COS(DECL)*COS(ELF)*COS(TIMER)
246         BB=SIN(DECL)*SIN(ELF)
247         C=AA+BB
248         Z=ACOS(C)
249     C
250     C   SOLVE-SOLAR-AZIMUTH
251     C
252         XNUM=-COS(DECL)*SIN(TIMER)
253         XDNOM=COS(ELF)*SIN(DECL)-SIN(ELF)*COS(TIMER)
254         SAZ=ATAN(XNUM/XDNOM)
255         IF(.NOT.(XNUM.LT.0.0.AND.XDNOM.GT.0.0)) GO TO 99944
256         SAZ=SAZ+PI
257         GO TO 99943
258     99944 IF(.NOT.(XNUM.GT.0.0.AND.XDNOM.GT.0.0)) GO TO 99943
259         SAZ=SAZ-PI
260     99943 CONTINUE
261     C
262     C   CALCULATE-SLOPE-ATMOS-ATTEM-AND-CLOUD-ADJUSTMENTS
263     C
264         SICF=COS(Z)*COS(SLOPE)+SIN(Z)*SIN(SLOPE)

```

```

265      & *COS(SAZ-SURFAC)
266      IF(.NOT.(SICF.LT.0.0.OR.COS(Z).LE.0.0)) GO TO 99941
267      SUN=0.0
268      GO TO 99942
269  99941  M=1/COS(Z)
270      IF(.NOT.(M.GE.0.0)) GO TO 99939
271      TAL=0.02023
272      IF(DAY.GE.92.0 .AND. DAY.LE.152.0)TAL=-0.02290
273          TA=ATEMP(I)
274          RH=RELHUM(I)
275          TD=5352.2/(21.4-ALOG(RH*ESAT(TA)))
276          WATER=EXP(0.07074*(TD-273.15)+TAL)
277          AB=0.271*(WATER*M)**0.303
278          A0=0.085-0.247*ALOG10(PRESS/1000.*1./M)
279          ARG1=((1.-AB)*0.349+(1.-A0)/(1.-A0*0.2)*0.651)
280      GO TO 99940
281  99939  ARG1=1.0
282  99940  QP=2.0*ARG1
283          QO=QP*SICF
284          IF(.NOT.(N CLOUD.EQ.0)) GO TO 99937
285          SUN=QO
286          GO TO 99938
287  99937  CLOUD=CLDCOV(I)
288          ARG2=-(BCL(N CLOUD)-.059)*M
289          CTF=(ACL(N CLOUD)/94.4)*EXP(ARG2)
290          SUN=QO-((CLOUD*CLOUD)*(QO-QO*CTF))
291  99938  CONTINUE
292  99942  SOLAR(I)=SUN
293      8888  CONTINUE
294      88888  CONTINUE
295  C-----
296  C
297  C      INPUT SURFACE OPTICAL PROPERTIES
298  C
299  C      SLOPE AND INTERCEPT OF EMISSIVITY EQUATION
300  C      SLOPE AND INTERCEPT OF ABSORBTIVITY EQUATION
301  C
302  C      IF SOIL IS THE TOP SURFACE, THESE MAY BE FUNCTIONS
303  C      OF SOIL VOLUMETRIC MOISTURE
304  C
305  C      IF ANOTHER MATERIAL IS THE TOP SURFACE,
306  C      THE SLOPE MAY BE SET TO
307  C      ZERO TO YIELD A CONSTANT VALUE OF PARAMETERS
308  C
309      READ(2,*)EPSNM,EPSNB
310      READ(2,*)SMALLAM,SMALLAB
311  C-----
312  C
313  C      INPUT SURFACE DEGREE OF SATURATION FLAG AND VALUE
314  C
315  C      NSATFLAG =0 IF SOIL MOISTURE DATA ARE IN THE INPUT FILE
316  C      NSATFLAG =1 IF THE SURFACE DEGREE OF SAT VALUE IS FIXED
317  C      SATVAL = FIXED SURFACE DEGREE OF SAT VALUE (DECIMAL)
318  C
319      READ(2,*)NSATFLAG,SATVAL
320  C

```

```

321 C      IF NSATFLAG IS NOT ZERO, THEN SET THE DEGSAT VALUE
322 C
323       IF(NSATFLAG.EQ.0) GO TO 99985
324       DO 99984 I=1,NLDATA
325 99984 DEGSAT(I)=SATVAL
326 C-----
327 C
328 C      INPUT-VEGETATION-PARAMETERS
329 C
330 99985 IVEG=0
331       READ(2,*)SIGF,STATE,EPF,FOLA,HFOL
332       IF(SIGF.LE.0.0)GO TO 99969
333       TF=ATEMP(1)
334       IVEG=1
335       EP1=EPF+EPSN-EPF*EPSN
336       Z0=0.131*HFOL**0.997
337       CH0=KSQ/(ALOG(ZASH/Z0)**2)
338       ZDSP=0.701*HFOL**0.979
339       CHH=KSQ/(ALOG((ZASH-ZDSP)/Z0)**2)
340       CHG=(1.-SIGF)*CH0+SIGF*CHH
341       DELTMP=1.
342       QAF=QSAT(TF)
343 C-----
344 C
345 C      INPUT LAYER SPECIFICATIONS
346 C
347 C      THICKNESS VERT. GRID THERMAL DIFF HEAT COND
348 C      CM      SPACE-CM      CM**2/MIN      CAL/MIN-CM-K
349 C
350 99969 CONTINUE
351       TOTTHICK=0.
352       READ(2,*)NOMATL
353       DO 99960 J4=1,6
354       READ(2,*)THK(J4),SFRQ(J4),ALPHM(J4),ALPHB(J4),FKM(J4),FKB(J4)
355       TOTTHICK=TOTTHICK+THK(J4)
356 99960 CONTINUE
357 C-----
358 C
359 C      INPUT BOTTOM BOUNDARY DATA
360 C
361 C      IF IFLUXY LT 0, THERE IS A FIXED HEAT FLUX THROUGH THE
362 C      BOTTOM BOUNDARY
363 C      IF LFLUXY=0, THE BOTTOM BOUNDARY HAS A FIXED TEMPERATURE
364 C      IF IFLUXY GT 0, THERE IS AIRSPACE BENEATH BOTTOM
365 C
366       READ(2,*)LFLUXY
367       IF(.NOT.(LFLUXY.EQ.0)) GO TO 99962
368       READ(2,*)DPRM0
369       TB=0.
370       DPRM0=DPRM0+273.15
371       GO TO 99963
372 99962 IF(.NOT.(LFLUXY.LT.0)) GO TO 99961
373 C
374       READ(2,*)DPRM1
375       TB=FKM(NOMATL)*VMF+FKB(NOMATL)
376       BEP=0.0

```

```

377     BEP=0.
378     BK=0.
379     REP=0.
380     TR=0.0
381     FACTD=0.
382     FACTE=0.
383     RK=0.
384     GO TO 99963
385 99961 READ(2,*)DPRM1,BEP,BK,REP,RK,TR
386     TB=FKM(NOMATL)*VMF+FKB(NOMATL)
387     TR=TR+273.15
388     FACTD=SIGMA*RK*REP*TR**4
389     FACTE=SIGMA*BK*BEP
390 99963 CONTINUE
391 C-----
392 C
393 C     INPUT MOISTURE PROFILE PARAMETERS. THESE STATEMENTS ARE
394 C     USED TO SET THE BOTTOM MOISTURE CONDITION AND THE
395 C     DEPTHS TO BE USED TO ESTABLISH A MOISTURE PROFILE AT
396 C     EACH POINT IN TIME
397 C
398 C     THE FIRST DATA ENTRY IS THE DEPTH (CM) BELOW WHICH THE
399 C     MOISTURE CONTENT IS TAKEN TO BE FIXED; THE SECOND DATA
400 C     ENTRY IS THAT MOISTURE LEVEL (%)
401 C
402 C     THE NEXT LINE OF DATA CONTAINS TWO DEPTHS FOR WHICH
403 C     MOISTURE CONTENT VALUES ARE CONTAINED IN THE INPUT FILE
404 C
405 C     IF THERE IS NO MOISTURE DATA, THE RELATIONSHIPS FOR
406 C     MATERIAL PROPERTIES WILL REFLECT THAT. THE SLOPE WILL BE
407 C     GIVEN AS ZERO, AND THE INTERCEPT WILL BE THE CONSTANT
408 C     PROPERTY VALUE THAT WILL BE USED BY THE PROGRAM
409 C
410     READ(2,*) ZMF,VMF
411     READ(2,*) ZM1,ZM2
412 C-----
413 C
414 C     INPUT SIMULATION CONTROLS
415 C
416 C     TIME STEP (MIN), PRINT FREQ (MIN), NO. OF 1ST DAY
417 C     ITERATIONS, WIND SPEED INDICATOR HEIGHT (CM)
418 C
419     READ(2,*)TFRQ,TPRNT,REPDAY,ZASH
420     IPRNT=TPRNT/TFRQ
421     IPPRNT=60/TFRQ
422     NPRNT=1
423     NPPRNT=1
424 C-----
425 C
426 C     SET INITIAL TOP SURFACE PARAMETERS
427 C
428     X1=ZM1
429     X2=ZM2
430     VMSURF=YVALUE(0.,VMOIS1(1),VMOIS2(1))
431     IF(NSATFLAG.NE.0) VMSURF=SATVAL*0.4
432     EPSN=EPSNM*VMSURF+EPSNB

```

```

433          SMALLA=SMALLAM*VMSURF+SMALLAB
434          FACTA=SIGMA*EPSN
435 C-----
436 C
437 C          ***** PRINT-INPUT-DATA *****
438 C
439 C-----
440 C
441
442          WRITE(4,*) '
443          WRITE(4,230)HEADER
444          WRITE(4,*) '
445          IF(NSINGLE.EQ.0) WRITE(4,235)NSNGLDAY,SATVAL
446          IF(NSINGLE.NE.0) WRITE(4,236)NSNGLDAY,SATVAL
447          WRITE(4,*) '
448          WRITE(4,*) SHLT_HT_CM'
449          WRITE(4,150) ZASH
450          WRITE(4,*) '
451          WRITE(4,*) SURFACE-ORIENTATION-SPECIFICATIONS'
452          WRITE(4,*) sfc_slp sfc_az latitude'
453          WRITE(4,*) deg-hor=0 deg_S=0 deg'
454          WRITE(4,160)SLOPE*180/PI,SURFAC*180.0/PI,LAT
455          WRITE(4,*) '
456          WRITE(4,*) HEAT-FLOW-CACULATION-CONTROLS'
457          WRITE(4,*) no_of no_24 time_stp prn_freq'
458          WRITE(4,*) layers hr_reps min min'
459          WRITE(4,*) <=6'
460          WRITE(4,180)NOMATL,REPDAY,TFRQ,TPRNT
461          WRITE(4,*) '
462          WRITE(4,*) TOP-SURFACE-CONSTANTS'
463          WRITE(4,*) emiss-m emiss-b absrb-m absrb-b'
464          WRITE(4,200)EPSNM,EPSNB,SMALLAM,SMALLAB
465          WRITE(4,*) '
466          WRITE(4,*) MATERIAL-LAYER-SPECIFICATIONS'
467          WRITE(4,*) layer thknss node-sp diff-m diff-b cond-m cond-b'
468          WRITE(4,*) no. cm cm . cm^2/min . cal/min-cm-deg-K'
469 C FOLLOWING ADDED ON 21 Aug 2004 TO HELP MAKE ALL OUTPUT FILES
470 C THE SAME SIZE
471          DO 99956 J4=1,6
472          WRITE(4,210)J4,THK(J4),SFRQ(J4),
473          & ALPHM(J4),ALPHB(J4),FKM(J4),FKB(J4)
474 99956 CONTINUE
475          WRITE(4,*) TOTTHICK
476          WRITE(4,*) BOTTOM_BOUNDARY_THERMAL_CONDITIONS'
477          IF(LFLUXY.NE.0) GO TO 99958
478          WRITE(4,90)LFLUXY
479          WRITE(4,92)DPRM0-273.15
480          WRITE(4,*) '
481          WRITE(4,*) '
482          GO TO 99959
483 99958 IF(LFLUXY.GT.0) GO TO 99957
484          WRITE(4,90)LFLUXY
485          WRITE(4,95)DPRM1
486          WRITE(4,*) '
487          WRITE(4,*) '
488          GO TO 99959

```

```

489 99957 WRITE(4,90)LFLUXY
490     WRITE(4,95)DPRM1
491     WRITE(4,*)' ---BOTTOM SURFACE-----',2H;;;,'SURF_BENTH AIRSP_TEMP'
492     WRITE(4,*)' EMISS GEO_SHAPE EMISS GEO_SHAPE DEG_C'
493     WRITE(4,*)1H;;;,'FACT(0.-1.)',1H;;;,'FACT(0.-1.)'
494     WRITE(4,97)BEP,BK,REP,RK,TR-273.15
495     WRITE(4,*)' '
496 99959 CONTINUE
497     WRITE(4,*)'FIXED_SOIL_MOISTURE_BOUNDARY_CONDITIONS'
498     WRITE(4,*)'depth to fixed moisture (cm)',ZMF
499     WRITE(4,*)'fixed volumetric moisture value (%)',VMF
500     WRITE(4,*)' '
501     WRITE(4,*)'DEPTHS_OF_TWO_MEASURED_VOLUMETRIC_SOIL_MOISTURES_(cm
502 )'
503     WRITE(4,*)ZM1,ZM2
504     WRITE(4,*)' '
505     CALL FLUSH()
506     WRITE(4,*)' VEGETATION_PARAMETERS'
507     WRITE(4,*)' covrg state emiss absorb fol_ht'
508     WRITE(4,*)'(0.0-1.0) . (0.0-1.0) (0.0-1.0) (cm)'
509     WRITE(4,240)SIGF,STATE,EPF,FOLA,HFOL
510     WRITE(4,*)' '
511     WRITE(4,*)' '
512     WRITE(4,*)' '
513     CALL FLUSH()
514 C-----
515 C
516 C   WRITE COLUMN HEADINGS TO OUTPUT FILE
517 C
518     IF(IVEG.GT.0) GO TO 1420
519     WRITE(4,350)
520     WRITE(4,360)
521     WRITE(4,370)
522     WRITE(4,380)
523     CALL FLUSH()
524     GO TO 1425
525 1420 WRITE(4,310)
526     WRITE(4,320)
527     WRITE(4,330)
528     WRITE(4,340)
529     CALL FLUSH()
530 CC-----
531 C
532 C   SET-UP-INITIAL-CONDITIONS
533 C
534 1425  NDT5=NFIRST
535     NSTEP=1
536     TIME=DT(1)
537     DIST=0.
538     IFLAG=0
539     DELT=TFRQ/60.
540     TEMPPROF(1,1)=NSNGLDAY
541 C
542 C   IX=LAYER NUMBER; IY=DEPTH SUBSCRIPT (1 AT SURFACE;
543 C   JMAX AT BOT BNDARY)
544 C

```

```

545     IX=1
546     IY=1
547     GO TO 99913
548 99914 IF(IX.GT.NOMATL) GO TO 99912
549 99913 INTR(IX)=IY
550     IF (SFRQ(IX).LE.0.) SFRQ(IX)=THK(IX)/10.
551     NX(IX)=MAX1(THK(IX)/SFRQ(IX)+.9,1.1)
552     RR(IX)=60.0*DELT/(SFRQ(IX)*SFRQ(IX))
553     LAYERS=0
554     GO TO 99910
555 99911 IF(LAYERS.GT.NX(IX)) GO TO 99909
556 99910 DEPTH(IY)=DIST
557     TEMPPROF(IY+1,1)=DEPTH(IY)
558     LNUM(IY)=IX
559 C
560 C     CALCULATE MOISTURE FOR THIS DEPTH
561 C
562     IF(DEPTH(IY).LT.ZMF) GO TO 2100
563     VOLMOIS=VMF
564     GO TO 99908
565 2100 IF(DEPTH(IY).LT.ZM2) GO TO 2050
566     X1=ZM2
567     X2=ZMF
568     VOLMOIS=YVALUE(DEPTH(IY),VMOIS2(1),VMF)
569     GO TO 99908
570 2050 X1=ZM1
571     X2=ZM2
572     VOLMOIS=YVALUE(DEPTH(IY),VMOIS1(1),VMOIS2(1))
573 99908 CONTINUE
574 C
575 C     RETRIEVE INITIAL MATERIAL PROPERTY VALUES AT EACH DEPTH
576 C
577     VALK=FKM(IX)*VOLMOIS+FKB(IX)
578     VALALPH=ALPHM(IX)*VOLMOIS+ALPHB(IX)
579     STOR(6,IY)=VALK
580     STOR(7,IY)=VALK/VALALPH
581     STOR(8,IY)=VALALPH
582     STOR(4,IY)=0.
583     STOR(2,IY)=STOR(6,IY)
584     STOR(3,IY)=STOR(7,IY)
585     IY=IY+1
586     DIST=DIST+SFRQ(IX)
587     LAYERS=LAYERS+1
588     GO TO 99911
589 99909 IX=IX+1
590     DIST=DIST-SFRQ(IX-1)
591     GO TO 99914
592 99912 JMAX=IY-1
593     INTR(IX)=JMAX
594 C
595 C     SET INITIAL TEMPERATURE PROFILE AS A LINEAR FIT
596 C     BETWEEN THE INITIAL AIR TEMPERATURE AND EITHER
597 C     A FIXED BOTTOM BOUNDARY TEMPERATURE OR A VALUE
598 C     OF 10 DEG C (283.15 K)
599 C
600     YYY(1)=ATEMP(1)

```

```

601      YYY(2)=283.15
602      IF(LFLUXY.EQ.0) YYY(2)=DPRM0
603      DO 2000 I=1,JMAX
604      X1=0.
605      X2=TOTTHICK
606      STOR(1,I)=YVALUE(DEPTH(I),YYY(1),YYY(2))
607      STOR(5,I)=STOR(1,I)
608      2000 CONTINUE
609      C-----
610      C
611      C ***** THIS IS THE SIMULATION CONTROL LOOP *****
612      C
613      C-----
614      c
615      C RUN-HEAT-FLOW-PROGRAM
616      C
617      99919 CONTINUE
618      ASSIGN 99917 TO I99918
619      GO TO 99918
620      C
621      C WRITE SIMULATION RESULTS FOR THIS TIME INCREMENT
622      C TO THE OUTPUT FILE (IF PRINT PARAMETERS ARE MET)
623      C
624      99917 ASSIGN 99915 TO I99916
625      GO TO 99916
626      99915 CONTINUE
627      CALL FLUSH()
628      C
629      C GO TO THE NEXT TIME STEP
630      C
631      GO TO 99919
632      C
633      C TERMINATE THE SIMULATION AND STORE TEMPERATURE PROFILES
634      C
635      99980 CONTINUE
636      WRITE(4,*)'NORMAL TERMINATION'
637      CALL FLUSH()
638      C
639      C OUTPUT THE TEMPERATURE PROFILES
640      C
641      DO 104 I=1,JMAX+1
642      WRITE(4,103) (TEMPPROF(I,J),J=1,26)
643      104 CONTINUE
644      103 FORMAT(26F8.3)
645      CALL FLUSH()
646      STOP
647      C-----
648      C
649      C ***** DO THE CALCULATIONS *****
650      C
651      C-----
652      99918 CONTINUE
653      C
654      NSTEP=NSTEP+1
655      TIME=NSTEP*DELT
656      IF(TIME.LE.24.) GO TO 940

```

```

657     TIME=DELT
658     NSTEP=1
659     NDTS=NDTS+1
660     IF(REPDAY.EQ.0) GO TO 940
661     C
662     C     ITERATION ON THE FIRST DAY
663     C
664     NDTS=NFIRST
665     REPDAY=REPDAY-1
666     940 IF(NDTS.GE.NLDATA) GO TO 99980
667     IF(TIME.LE.DT(NDTS+1)) GO TO 938
668     IF(DT(NDTS+1).LT.DT(NDTS)) GO TO 938
669     C     PREVIOUS LINE A CHECK FOR MIDNIGHT BEING LABELED 0000 HRS
670     NDTS=NDTS+1
671     GO TO 940
672     938 CONTINUE
673     C
674     C     CHECK FOR THE END OF A SINGLE-DAY SIMULATION
675     C
676     IF(NSINGLE.NE.0.AND.JD(NDTS).NE.NSNGLDAY) GO TO 99980
677     C
678     C     CALCULATE ALL TIME-BASED INTERPOLATED PARAMETER VALUES
679     C
680     X1=DT(NDTS)
681     X2=DT(NDTS+1)
682     IF(DT(NDTS+1).LT.DT(NDTS)) X2=DT(NDTS+1)+24.
683     C     PREVIOUS LINE A CORRECTION FOR MIDNIGHT BEING LABELED 0000 HRS
684     AT= YVALUE(TIME,ATEMP(NDTS),ATEMP(NDTS+1))
685     TA=AT
686     CTEMA=AT
687     RH= YVALUE(TIME,RELHUM(NDTS),RELHUM(NDTS+1))
688     PRESS=YVALUE(TIME,BPRESS(NDTS),BPRESS(NDTS+1))
689     FACTH=(1000/PRESS)**.286
690     C     FACTH IS A FACTOR USED IN CALCULATING THE CONVECTION TERM
691     SOL= YVALUE(TIME,SOLAR(NDTS),SOLAR(NDTS+1))
692     BTERM=SOL
693     SPEED=YVALUE(TIME,WINDSP(NDTS),WINDSP(NDTS+1))
694     UA=SPEED
695     CLOUD=YVALUE(TIME,CLDCOV(NDTS),CLDCOV(NDTS+1))
696     WET=YVALUE(TIME,DEGSAT(NDTS),DEGSAT(NDTS+1))
697     VM1=YVALUE(TIME,VMOIS1(NDTS),VMOIS1(NDTS+1))
698     VM2=YVALUE(TIME,VMOIS2(NDTS),VMOIS2(NDTS+1))
699     ST1=YVALUE(TIME,STEMP1(NDTS),STEMP1(NDTS+1))
700     ST2=YVALUE(TIME,STEMP2(NDTS),STEMP2(NDTS+1))
701     RT1=YVALUE(TIME,RTEMP1(NDTS),RTEMP1(NDTS+1))
702     RT2=YVALUE(TIME,RTEMP2(NDTS),RTEMP2(NDTS+1))
703     C
704     C     CALCULATE NEW MOISTURE AND THERMAL PROPERTIES PROFILES
705     C
706     DO 2400 IPROF=1,JMAX
707     IX=LNUM(IPROF)
708     IF(DEPTH(IPROF).LT.ZMF) GO TO 2330
709     VOLMOIS=VMF
710     GO TO 2350
711     2330 IF(DEPTH(IPROF).LT.ZM2) GO TO 2340
712     X1=ZM2

```

```

713         X2=ZMF
714         VOLMOIS=YVALUE(DEPTH(IPROF),VM2,VMF)
715         GO TO 2350
716 2340 X1=ZM1
717         X2=ZM2
718         VOLMOIS=YVALUE(DEPTH(IPROF),VM1,VM2)
719 2350 CONTINUE
720         VALK=FKM(IX)*VOLMOIS+FKB(IX)
721         VALALPH=ALPHM(IX)*VOLMOIS+ALPHB(IX)
722         STOR(1,IPROF)=STOR(5,IPROF)
723         STOR(6,IPROF)=VALK
724         STOR(7,IPROF)=VALK/VALALPH
725         STOR(8,IPROF)=VALALPH
726 2400 CONTINUE
727 C
728 C     SET TOP SURFACE PARAMETERS FOR THIS TIME INCREMENT
729 C
730         X1=ZM1
731         X2=ZM2
732         VMSURF=YVALUE(0.,VM1,VM2)
733         IF(NSATFLAG.NE.0) VMSURF=SATVAL*0.4
734         EPSN=EPSNM*VMSURF+EPSNB
735         SMALLA=SMALLAM*VMSURF+SMALLAB
736         FACTA=SIGMA*EPSN
737 C
738 C     COUNTERS FOR PRINT OUTPUT AND PROFILE OUTPUT
739 C
740         NPRNT=NPRNT+1
741         IF(JD(NDTS).EQ.NSNGLDAY) NPPRNT=NPPRNT+1
742 99907 ZZA=STOR(5,1)
743         ZZB=STOR(5,JMAX)
744         TEML=ZZA
745         TEMR=ZZB
746 C
747 C     CALCULATE-BOUNDARY-CONDITIONS
748 C
749         IF(IVEG.EQ.0)GO TO 930
750         ASSIGN 99905 TO I99800
751         GO TO 99800
752 930 ASSIGN 99905 TO I99906
753         GO TO 99906
754 C
755 C     CALCULATE-UPPER-BOUNDARY-VALUES
756 C
757 99905 IF(IVEG.EQ.0) GO TO 900
758         ASSIGN 99903 TO I99797
759         GO TO 99797
760 900 ASSIGN 99903 TO I99904
761         GO TO 99904
762 C
763 99903 IX=1
764         J=2
765         IMATL=NOMATL
766         IF(NOMATL.NE.1) GO TO 99896
767         IZ=NX(IX)-1
768         IF(IZ.LE.0) GO TO 99902

```

```

769 C
770 C   CALCULATE-INSIDE-MATERIAL-VALUES WHEN THERE IS
771 C     ONLY A SINGLE LAYER OF MATERIAL
772 C
773     ASSIGN 99902 TO I99899
774     GO TO 99899
775 C
776 99896 IF(IMATL.EQ.1) GO TO 99893
777     IZ=NX(IX)-1
778     IF(IZ.LE.0) GO TO 99892
779 C
780 C   CALCULATE-INSIDE MATERIAL-VALUES WHEN THERE IS
781 C     MORE THAN ONE LAYER OF MATERIAL
782 C
783     ASSIGN 99892 TO I99899
784     GO TO 99899
785 C
786 C   CALCULATE-INTERFACE-VALUES
787 C
788 99892 ASSIGN 99894 TO I99890
789     GO TO 99890
790 C
791 C   CALCULATE-INSIDE-MATERIAL-VALUES FOR THE LAST
792 C     LAYER OF MATERIAL
793 C
794 99893 IZ=NX(IX)-1
795     IF(IZ.LE.0) GO TO 99902
796     ASSIGN 99902 TO I99899
797     GO TO 99899
798 99894 IMATL=IMATL-1
799     GO TO 99896
800 C
801 C   CALCULATE-LOWER-BOUNDARY-VALUES
802 C
803 99902 ASSIGN 99883 TO I99886
804     GO TO 99886
805 C
806 99883 GO TO I99918
807 C-----
808 C
809 C   *****   END OF CALCULATIONS FOR THIS TIME STEP   *****
810 C
811 C-----
812 99906 CONTINUE
813 C
814 C   CALCULATE-BOUNDARY-CONDITIONS
815 C
816     X1=ZM1
817     X2=ZM2
818     VMSURF=YVALUE(0.,VM1,VM2)
819     IF(NSATFLAG.NE.0) VMSURF=SATVAL*0.4
820     B = -FKM(1)*VMSURF-FKB(1)
821     T=TIME
822     IF(BTERM.GT.0.0)BTERM=BTERM*SMALLA
823 C
824 C   CALCULATE BOTTOM BOUNDARY HEAT TERMS (APRM,DPRM,BPRM)

```

```

825 C
826     ASSIGN 99880 TO I99881
827     GO TO 99881
828 C
829 C     CALCULATE THE ATMOSPHERIC IR EMISSION (ATERM)
830 C
831 99880 ASSIGN 99878 TO I99879
832     GO TO 99879
833 C
834 C     CALCULATE CONVECTION (HTERM)
835 C
836 99878 ASSIGN 99876 TO I99877
837     GO TO 99877
838 C
839 C     CALCULATE EVAPORATIVE HEAT LOSS (DTERM)
840 C
841 99876 ASSIGN 99874 TO I99875
842     GO TO 99875
843 C
844 99874 D= ATERM + BTERM - HTERM - DTERM
845     GO TO I99906
846 C
847 C-----
848 C     BOTTOM BOUNDARY HEAT TERMS
849 C
850 99881 BPRM=TB
851     IF(.NOT.(TB.EQ.0.0)) GO TO 99872
852     APRM=1.0
853     DPRM=DPRM0
854     GO TO 99873
855 99872 APRM=FACTE*TEMR*TEMR*TEMR
856     DPRM=DPRM1+FACTD
857 99873 GO TO I99881
858 C
859 C-----
860 C     ATMOSPHERIC-INFRARED-EMISSION-ATERM
861 C
862 99879 TAK=TA
863     TAC=(TAK-273.15)
864     EA=6.108*RH*EXP((AC*TAC)/(TAK-BC))
865     ALPHI=(0.61+0.05*SQRT(EA))*(1.0+(CLR(NCLOUD)*(CLOUD**2)))
866     DOWNIR=0.8132E-10*TAK**4*ALPHI
867     ATERM=DOWNIR
868     GO TO I99879
869 C
870 C-----
871 C     CALCULATE-CONVECTION-HTERM
872 C
873 C     TA IS THE AIR TEMPERATURE
874 C     TEML IS THE SURFACE TEMPERATURE
875 C
876 99877 TAK=TA
877     TSK=TEML
878     RHOA=-0.001*0.348*PRESS/TAK
879     1200 THETAZ=TAK*FACTH
880     THETAS=TSK*FACTH

```

```

881      DTHETA=(THETAZ-THETAS)/ZASH
882      DU=SPEED/ZASH
883      THETA V=(THETAZ+THETAS)/2.0
884      RI=G*DTHETA/(THETA V*DU**2)
885      COE1=15.0
886      COE2=1.175
887      EX=.75
888      IF(TSK.GT.TAK)GO TO 31
889      IF(RI.GT.0.2)RI=.19999
890      COE1=5.0
891      COE2=1.0
892      EX=2.0
893      31  HTER=RHOA*KSQ*(ZASH/ALOG(ZASH))**2*DU
894          & *(COE2*(1.0-COE1*RI)**EX)
895      C
896      C      JOHN CURTIS REPLACED ZASH IN 31
897      C      WITH THE LOGARITHMIC HEIGHT (ZASH/ALOG(ZASH))
898      C      BASED ON A REVIEW OF OKE'S FORMULATION.
899      C
900      HTERM=HTER*CP*DTHETA
901      99864 GO TO I99877
902      C
903      C-----
904      C      CALCULATE-EVAPORATIVE-HEAT-LOSS-DTERM
905      C
906      C
907      99875 CONTINUE
908      IF(.NOT.(TEML.GT.TA)) GO TO 99860
909      CTEMA=TA
910      KTEMPA=CTEMA
911      CTEMA=CTEMA-273.15
912      KTEMPG=TEML
913      ES=EXP((AC*(KTEMPG-273.15))/(KTEMPG-BC))*6.1071
914      EA=EXP((AC*CTEMA)/(KTEMPA-BC))*6.1071*RH
915      DG=0.622/PRESS*(EA-ES)*WET/ZASH
916      XL=597.3-0.566*(CTEMA+KTEMPG-273.15)/2.0
917      DTERM=HTER*XL*DG
918      GO TO 99861
919      99860 DTERM=0.0
920      99861 GO TO I99875
921      C-----
922
923      99904 CONTINUE
924      C      CALCULATE-UPPER-BOUNDARY-VALUES
925      C
926      C      T1 IS AN ESTIMATE FOR THE TEMPERATURE OF THE FIRST NODE BELOW
927      C      THE SURFACE AT THE END OF THIS TIME INCREMENT; FOUND USING
928      C      THE 1-D HEAT FLOW EQUATION
929      C
930      T1=STOR(8,1)*RR(1)*(STOR(1,3)-2.*STOR(1,2)+STOR(1,1))+STOR(1,2)
931      III=0
932      830 III=III+1
933      C
934      C      T2 IS F(Ts)/(PARTIAL OF F WRT Ts), WHICH IS THE CHANGE IN Ts
935      C      FROM THE NEWTON METHOD FOR SOLVING F(Ts)=0
936      C

```

```

937 C      F(Ts) IS EQUATION (4) IN THE ORIGINAL REPORT
938 C
939      T2=STOR(5,1)**4*FACTA*SFRQ(1)+STOR(6,1)*STOR(5,1)
940 & -(STOR(6,1)*T1+D*SFRQ(1))
941      T2=T2/(4.*FACTA*SFRQ(1)*STOR(5,1)**3+STOR(6,1)-SFRQ(1)*DDDT)
942      STOR(5,1)=STOR(5,1)-T2
943      TEML=STOR(5,1)
944      ASSIGN 825 TO I99877
945 C
946 C      GET HTERM
947 C
948      GO TO 99877
949      825 ASSIGN 810 TO I99875
950 C
951 C      GET DTERM
952 C
953      GO TO 99875
954      810 DNEW=ATERM+BTERM-HTERM-DTERM
955      IF(ABS(T2).LT.0.005 .OR. III.GT.5)GO TO I99904
956      DDDT=- (DNEW-D)/T2
957      D=DNEW
958      GO TO 830
959 C
960 C-----
961 99899 CONTINUE
962 C      CALCULATE-INSIDE-MATERIAL-VALUES
963 C
964      GO TO 99856
965 99857 IF(IZ.LE.0) GO TO 99855
966 99856 CONTINUE
967      STOR(5,J)=STOR(1,J)+STOR(8,IX)*RR(IX)*(STOR(1,J-1)-2.*STOR(1,J)
968 & +STOR(1,J+1))
969 C      WRITE(4,*)J,IZ,IX,STOR(1,J),STOR(5,J),STOR(8,IX),RR(IX)
970      J=J+1
971      IZ=IZ-1
972      GO TO 99857
973 99855 GO TO I99899
974 C-----
975 C      CALCULATE-INTERFACE-VALUES
976 C
977 99890 CONTINUE
978      BCOEF=STOR(6,J-1)/SFRQ(IX)
979      DCOEF=STOR(6,J+1)/SFRQ(IX+1)
980      CCOEF=BCOEF+DCOEF
981      ACOEF=BCOEF/(2.*STOR(8,IX)*RR(IX))+DCOEF/(2.*STOR(8,IX+1)
982 & *RR(IX+1))
983      STOR(5,J)=STOR(1,J)+(BCOEF*STOR(1,J-1)-CCOEF*STOR(1,J)+DCOEF*
984 & STOR(1,J+2))/ACOEF
985      STOR(5,J+1)=STOR(5,J)
986 C      WRITE(4,*)J,IZ,IX,STOR(1,J),STOR(5,J),STOR(8,IX),RR(IX)
987      IX=IX+1
988      J=J+2
989      GO TO I99890
990 C-----
991 C      CALCULATE-LOWER-BOUNDARY-VALUES
992 99886 IF(LFLUXY.EQ.0) GO TO 880

```

```

993     I=1
994     870 CONTINUE
995     F2=4.0*FACTE*STOR(5,J)**3 - BPRM
996 CCC   F2=4.*APRM-BPRM
997     IF(F2.EQ.0)F2=.000001
998     F2= -(FACTE*SFRQ(IX)*STOR(5,J)**4-BPRM*STOR(5,J)
999     &      +BPRM*STOR(5,J-1)-DPRM*SFRQ(IX))/F2
1000    STOR(5,J)=STOR(5,J) + F2
1001    PRINT *,I,F2
1002    I=I+1
1003    IF(I.LE.3) GO TO 870
1004    880 IF(LFLUXY.EQ.0) STOR(5,J)=STOR(5,J)
1005    GO TO I99886
1006    C-----
1007    99916 CONTINUE
1008    C
1009    C   PRINT-OUTPUT
1010    C
1011    DO 99842 JKX=1,NOMATL+1
1012    IJ=INTR(JKX)
1013    TITLE(JKX)=(STOR(5,IJ)-273.15)
1014    99842 CONTINUE
1015    STEMP=STOR(5,1)-273.15
1016    PRINTHR=AMOD(TIME,24.0)
1017    IF(PRINTHR.EQ.0.)PRINTHR=24
1018    IF(IVEG.EQ.1) GO TO 1110
1019    IGBR=5.67E-8*EPSN*STOR(5,1)**4
1020    ISOL=BTERM/SMALLA*697.6+0.5
1021    IABSOR=ISOL*SMALLA
1022    IATERM=ATERM*697.6
1023    IHTERM=HTERM*697.6+0.5
1024    IDTERM=DTERM*697.6+0.5
1025    IF(NPRNT.LT.IPRNT) GO TO 99844
1026    IF(REPDAY.GT.0) GO TO 99843
1027    WRITE(4,2610)JD(NDTS),PRINTHR,JD(NDTS)+PRINTHR/24.,
1028    &TA-273.15,STEMP,IGBR,
1029    & ISOL,IABSOR,IATERM,IHTERM,IDTERM,RT1,RT2,STEMP-RT1,STEMP-RT2
1030    CALL FLUSH()
1031    c   WRITE(4,*)TAK,TSK,PRESS,RHOA,FACTH,THETAZ,THETAS,DTHETA,DU,
1032    c   & RI,THETA V,KSQ,ZASH
1033    99843 NPRNT=0
1034    99844 CONTINUE
1035    IF(JD(NDTS).NE.NSNGLDAY) GO TO I99916
1036    IF(NPPRNT.LT.IPPRNT) GO TO I99916
1037    IF(REPDAY.GT.0) GO TO 7778
1038    C
1039    C   CAPTURE TEMPERATURE VS DEPTH PROFILES AT HOURLY INTERVALS
1040    C
1041    TEMPPROF(1,JPROF)=PRINTHR
1042    DO 7777 IPROF=1,JMAX
1043    TEMPPROF(IPROF+1,JPROF)=STOR(1,IPROF)-273.15
1044    7777 CONTINUE
1045    JPROF=JPROF+1
1046    7778 NPPRNT=0
1047    GO TO I99916
1048    C

```

```

1049 1110 ASSIGN 1400 TO I1410
1050 GO TO 1410
1051 1400 CONTINUE
1052 WRITE(4,270)PRINTHR,ISURFG+IREFRA,ISURFG,TEFFR-273.15,
1053 & TEFF-273.15,TEML-273.15,TF-273.15,ISOL
1054 CALL FLUSH()
1055 GO TO I99916
1056 C
1057 C-----
1058 99800 CONTINUE
1059 C CALCULATE-BOUNDARY-CONDITIONS-WITH-VEG
1060 T=TIME
1061 C
1062 C ATMOSPHERIC-INFRARED-EMISSION-ATERM
1063 ASSIGN 980 TO I99879
1064 GO TO 99879
1065 C
1066 980 CONTINUE
1067 IF(UA.LT.10.0)UA=10.0
1068 UAF=0.83*SIGF*UA*SQRT(CHH)+(1.-SIGF)*UA
1069 DELTMP=5.
1070 CF=0.01*(1.+30.0/UAF)
1071 DU=(UA-UAF)/ZASH
1072 RS=1/(.05+.0021*(SOL*697.6))
1073 RC=RS*STATE/(7.0*SIGF)
1074 ATF(1)=TF
1075 ASSIGN 1210 TO I950
1076 GO TO 950
1077 1210 CONTINUE
1078 FEB(1)=FENB
1079 NDEX=0
1080 1240 TF=TF+DELTMP
1081 NDEX=NDEX+1
1082 ASSIGN 1220 TO I950
1083 GO TO 950
1084 1220 CONTINUE
1085 FEB(2)=FENB
1086 IF(FEB(1)*FEB(2).LT.0.0) GO TO 1230
1087 IF(ABS(FEB(2)).GT.ABS(FEB(1)))DELTMP=-5.
1088 IF(NDEX.LT.100)GO TO 1240
1089 WRITE(4,*)'FOLIAGE ENERGY BUDGET HAS NOT CROSSED X-AXIS'
1090 WRITE(4,*)'AFTER 100 SEARCH STEPS. CHECK INPUT DATA.'
1091 CALL FLUSH()
1092 STOP
1093 1230 CONTINUE
1094 ATF(2)=TF
1095 1270 SLOPE1=(FEB(2)-FEB(1))/(ATF(2)-ATF(1))
1096 BINT=FEB(1)-SLOPE1*ATF(1)
1097 TF0=-BINT/SLOPE1
1098 IF(ABS(TF-TF0).LE.0.001)GO TO 1260
1099 TF=TF0
1100 ASSIGN 1250 TO I950
1101 GO TO 950
1102 1250 CONTINUE
1103 IF(FENB*FEB(2).GT.0.0)IP=2
1104 IF(FENB*FEB(1).GT.0.0)IP=1

```

```

1105     ATF(IP)=TF
1106     FEB(IP)=FENB
1107     GO TO 1270
1108 1260 GO TO I99800
1109 C-----
1110 C     CALCULATE-UPPER-BOUNDARY-VALUES-FOR-FOLAGE
1111 99797 CONTINUE
1112     DELTMP=5.
1113     ATF(1)=TEML
1114     ASSIGN 1310 TO I1300
1115     GO TO 1300
1116 1310 CONTINUE
1117     FEB(1)=FENB
1118     NDEX=0
1119 1340 TEML=TEML+DELTMP
1120     NDEX=NDEX+1
1121     ASSIGN 1320 TO I1300
1122     GO TO 1300
1123 1320 CONTINUE
1124     FEB(2)=FENB
1125     IF(FEB(1)*FEB(2).LT.0.0) GO TO 1330
1126     IF(ABS(FEB(2)).GT.ABS(FEB(1)))DELTMP=-5.
1127     IF(NDEX.LT.100)GO TO 1340
1128     WRITE(4,*)'GROUND ENERGY BUDGET HAS NOT CROSSED X-AXIS'
1129     WRITE(4,*)'AFTER 100 SEARCH STEPS. CHECK INPUT DATA.'
1130     CALL FLUSH()
1131     STOP
1132 1330 CONTINUE
1133     ATF(2)=TEML
1134 1370 SLOPE1=(FEB(2)-FEB(1))/(ATF(2)-ATF(1))
1135     BINT=FEB(1)-SLOPE1*ATF(1)
1136     TF0=-BINT/SLOPE1
1137     IF(ABS(TEML-TF0).LE.0.001)GO TO 1360
1138     TEML=TF0
1139     ASSIGN 1350 TO I1300
1140     GO TO 1300
1141 1350 CONTINUE
1142     IF(FENB*FEB(2).GT.0.0)IP=2
1143     IF(FENB*FEB(1).GT.0.0)IP=1
1144     ATF(IP)=TEML
1145     FEB(IP)=FENB
1146     GO TO 1370
1147 1360 STOR(5,1)=TEML
1148     GO TO I99797
1149 C-----
1150 C     CALCULATE-ENERGY-BUDGET
1151 950 TAF=(1.-SIGF)*TA+SIGF*(0.3*TA+0.6*TF+0.1*TEML)
1152     DTHETA=(TA-TF)*FACTH/ZASH
1153     THETA=(TA+TF)*FACTH/2.0
1154     RI=G*DTHETA/(THETA*DU**2)
1155     RHOAF=-0.001*.348*PRESS/((TF+TA)/2.)
1156     COE1=15.
1157     COE2=1.175
1158     EX=.75
1159     IF(RI.LE.0.)GO TO 1280
1160     IF(RI.GT.0.2)RI=0.199

```

```

1161      COE1=5
1162      COE2=1.
1163      EX=2.0
1164 1280  CONTINUE
1165      HTER=RHOAF*KSQ*ZASH**2*DU
1166 C & *COE2*(1.-COE1*RI)**EX
1167 C   HSF=1.1*7.*SIGF*CP*CF*UAF*(TF-TAF)*60.
1168      HSF=HTER*CP*DTHETA*60.
1169 C   XL=597.3-0.566*TAF
1170      RA=(ALOG((ZASH-ZDSP)/Z0)*COE2*((1.-COE1*RI)**EX))**2
1171 & /(.16*UA)
1172      RDP=RA/(RS+RA)
1173      QF=RDP*QSAT(TF)+(1.-RDP)*QAF
1174      QAF=(1.-SIGF)*Q(TA)+SIGF*(Q(TA)*0.3+QF*0.6+QG*0.1)
1175      EF=(-RHOAF*CP/0.66)*(ESAT(TF)-E(TA))/(RA+RC)*60.
1176      IF(EF.LT.0.0)EF=0.0
1177      SHRW=FOLA*SOL
1178      XLNGW=EPF*ATERM
1179      TG4=EPF*EPSN/EP1*SIGMA*TEML**4
1180      TF4=(EP1+EPSN)/EP1*EPF*SIGMA*TF**4
1181      FENB=SIGF*(SHRW+XLNGW+TG4-TF4)-HSF-EF
1182      GO TO I950
1183 C-----
1184 C   CALCULATE-ENERGY-BUDGET-FOR-GROUND
1185 1300  CONTINUE
1186      T1=STOR(8,1)*RR(1)*(STOR(1,3)-2.*STOR(1,2)+STOR(1,1))
1187 & +STOR(1,2)
1188      TF4=SIGMA*TF**4
1189      TG4=SIGMA*TEML**4
1190      QG=WET*QSAT(TEML)+(1.-WET)*QAF
1191      RHOAG=0.001*0.348*PRESS/TAF
1192      XL1=597.3-0.566*(TAF+TEML-2.0*273.15)/2.
1193      SG=(1.-SIGF)*SOL
1194      RLU=(1.-SIGF)*(EPSN*TG4+(1.-EPSN)*ATERM)
1195 & +SIGF*(EPSN*TG4+(1.-EPSN)*EPF*TF4)/EP1
1196      RLD=(1.-SIGF)*ATERM+SIGF*(EPF*TF4+(1.-EPF)*EPSN*TG4)/EP1
1197      HSG=RHOAG*CP*CHG*UAF*(TEML-TAF)*60.
1198      ELG=RHOAG*CHG*UAF*(QG-QAF)*60.
1199      FENB=SMALLA*SG-RLU+RLD-HSG-ELG*XL1+(T1-TEML)/SFRQ(1)*STOR(6,1)
1200      GO TO I1300
1201 C-----
1202 C   CALCULATE-RADIANCE-VALUES
1203 1410  CONTINUE
1204      REFRAD=((1.-SIGF)*(1-EPSN)+SIGF*(1-EPF))*DOWNIR*697.6
1205      FOLGB=EPF*5.67E-8*TF**4
1206      GRNDGB=EPSN*5.67E-8*TEML**4
1207      SURFGB=SIGF*FOLGB+(1.-SIGF)*GRNDGB
1208      EEFF=SIGF*EPF+(1.-SIGF)*EPSN
1209      TEFF=(SURFGB/5.67E-8)**.25
1210      ISURFG=SURFGB+.5
1211      TEFFR=((SURFGB+REFRAD)/(5.67E-8))**.25
1212      IREFRA=REFRAD+0.5
1213      ISOL=SOL*697.6+0.5
1214      GO TO I1410
1215 C-----
1216 C

```

```

1217 C      ***** VARIABLE DEFINITIONS *****
1218 C
1219 C-----
1220 C  ALPH(IX)  THERMAL DIFFUSIVITY OF LAYER IX IN CM**2/MIN
1221 C
1222 C  APRM    FACTE*TEMP**3 IN CAL/CM**2-MIN-C
1223 C
1224 C  ATERM    ENERGY CONTRIBUTED BY ATMOSPHERIC IR EMISSION
1225 C          CAL CM**2-MIN
1226 C
1227 C  B        HEAT CONDUCTIVITY OF SURFACE CAL/CM**2-MIN-C
1228 C
1229 C  BBB(J,I) Y INTERCEPT OF LINEAR EQUATION, USED
1230 C          FOR TABLE INTERPOLATION.
1231 C
1232 C  BK       BOTTOM SURFACE GEOMETRIC SHAPE IN FRACTION(0.0-1.0)
1233 C
1234 C  BPRM    HEAT CONDUCTIVITY OF BOTTOM BOUNDARY LAYER
1235 C
1236 C  BTERM    ENERGY CONTRIBUTED BY INSOLATION AFTER ADJUSTMENT USING
1237 C          SURFACE ABSORPTIVITY. IN CAL/CM**2-MIN
1238 C
1239 C  CLOUD    CLOUD COVER IN FRACTION OF 0.1-1.0
1240 C
1241 C  DAY      JULIAN DAY USED IN SOLVING INSOLATION
1242 C
1243 C  DECL     SOLAR DECLINATION ANGLE
1244 C
1245 C  DELT     TIME STEP IN HOURS
1246 C
1247 C  DIST     DEPTH IN CM OF INITIAL SOIL PROFILE AT WHICH
1248 C          CORRESPONDING SOIL TEMPERATURE IN DEGREE C IS
1249 C          INTERPOLATED.(TABLE 5)
1250 C
1251 C  DPRM     HEAT FLUX IN CAL/CM**2-MIN AT BOTTOM BOUDARY OR
1252 C          TEMPERATURE IN RANKINS AT BOTTOM BOUNDARY.
1253 C
1254 C  DPRM0    TEMPERATURE OF BOTTOM MATERIAL IN
1255 C          DEGREE CELSIUS.USED WHEN LFLUXY=0
1256 C
1257 C  DPRM1    HEAT FLUX OF BENEATH BOTTOM MATERIAL,
1258 C          IN CAL/CM**2-MIN, USED WHEN LFLUXY NOT
1259 C          EQUAL 0
1260 C  DTERM    ENERGY LOSS DUE TO EVAPORATION
1261 C
1262 C  DUST     ATMOSPERIC DUST IN POUNDS/CUBIC CENTIMETERS
1263 C          (LBS/CC)USED IN SOLVING INSOLATION.
1264 C
1265 C  ELF      LATITUDE IN RADIANS
1266 C
1267 C  EPSN     EMISSIVITY OF SURFACE MATERIAL
1268 C
1269 C  FACTA    SIGMA*EPSN
1270 C
1271 C  FACTD    FACTD=SIGMA*BK*BEP*TR**4 USED IN BOTTOM BOUNDARY
1272 C          CALCULATION WHEN THERE IS AIRSPACE BENEATH THE BOTTOM

```

```

1273 C
1274 C FACTE FACTE=SIGMA*BK*BEP USED IN BOTTOM BOUNDARY CALCULATION
1275 C WHEN THERE IS AIRSPACE BENEATH THE BOTTOM
1276 C
1277 C FACTH USED IN SOLVING CONVECTION TERM (HTERM)
1278 C (1000.0/PRESS)**0.286
1279 C
1280 C FK(IX) HEAT CONDUCTIVITY OF LAYER IX IN CAL/MIN-CM-K
1281 C
1282 C FMM(J,I) SLOPE OF LINEAR EQUATION,USED FOR TABLE INTERPOLATION
1283 C
1284 C HEADER 72 CHARACTER INPUT VARIABLE USED TO PRINT
1285 C COMMENTS ON OUTPUT.
1286 C
1287 C HTERM ENERGY LOSS OF GAIN DUE TO CONVECTION CAL/CM**2-MIN
1288 C
1289 C IEOF SET FROM 0 TO 1 WHEN AN EOF IS ENCOUNTERED. USED TO
1290 C TERMINATE PROGRAM
1291 C
1292 C IMATL BACKWARD COUNTER OF LAYERS. STARTING WITH THE NUMBER
1293 C OF LAYERS.
1294 C
1295 C INTR(IX) BEGINNING SUB-LAYER DEPTH NUMBER FOR LAYER NUMBER IX
1296 C
1297 C IPRNT BACKWARD COUNTER SET=NPRNT. WHEN EQUAL TO 1 OUTPUT IS
1298 C PRINTED.
1299 C
1300 C ITIME BACKWARD COUNTER INITIALIZE AS TOTAL TIME STEPS IN HOUR
1301 C
1302 C IX LAYER NUMBER STARTING WITH TOP LAYER
1303 C
1304 C IY SUB-LAYER DEPTH NUMBER
1305 C
1306 C JMAX THE TOTAL NUMBER OF SUB-LAYERS
1307 C
1308 C LAT LATITUDE USED IN SOLVING INSOLATION
1309 C!
1310 C LFLUXY INPUT BOTTOM BOUNDARY DATA CONTROL SWITCH. IF=0,THERE
1311 C IS NO HEAT FLUX THROUGH BOTTOM OF MATERIAL,IF NEGATIVE
1312 C THERE IS NO AIR SPACE BENEATH BOTTOM MATERIAL,IF POSIT-
1313 C IVE THERE IS AIR SPACE BENEATH BOTTOM MATERIAL.
1314 C
1315 C LN DUMMY VARIABLE TO READ LINE NUMBER FROM INPUT FILE
1316 C
1317 C M SECANT OF SOLAR ZENITH ANGLE IN RADIANS
1318 C
1319 C INTERPOLATION MODULE.
1320 C
1321 C NLOUD CLOUD TYPE INDEX NUMBER (1-9) USED IN
1322 C SOLVING INSOLATION,INFRARED EMISSION.
1323 C
1324 C NOMATL NUMBER OF MATERIAL LAYERS USED IN SOLVING HEAT FLOW
1325 C
1326 C NPRNT NUMBER OF TIMES OUTPUT TIME PRINT FREQUENCY IS DIVISBL
1327 C BY TIME STEPS. USED TO DETERMINED WHEN TO PRINT OUTPUT.
1328 C

```

1329 C NTABL TABLE NUMBER  
1330 C  
1331 C NX(IX) NUMBER OF SUBLAYER OF EACH LAYER, NX(IX)=THK(IX)/SFRQ(IX)  
1332 C  
1333 C PRESS ATMOSPHERIC PRESSURE IN MILLIBAR(MB)  
1334 C USED IN SOLVING INSOLATION  
1335 C  
1336 C REP EMISSIVITY BENEATH AIRSPACE  
1337 C  
1338 C RH RELATIVE HUMIDITY  
1339 C  
1340 C RHOC(IX) FK(IX)/ALPH(IX) IN CAL/CM\*\*2-K  
1341 C  
1342 C RI RICHARDSON INDEX NUMBER USE IN SOLVING CONVECTION  
1343 C ENERGY LOSS.  
1344 C  
1345 C RK SURFACE BENEATH AIRSPACE GEOMETRIC  
1346 C SHAPE IN FRACTION (0.0 - 1.0)  
1347 C  
1348 C RR(IX) RR(IX)=DELT/SFREQ\*\*2.(PART OF HEAT FLOW EQUATION)  
1349 C  
1350 C  
1351 C SAZ SOLAR AZIMUTH IN RADIANS. SAZ=ATAN(-COS(DECL)\*SIN(TIMER  
1352 C (COS(ELF\*SIN(DECL)-SIN(ELF)\*COS(TIMER)))  
1353 C  
1354 C SFRQ(IX) VERTICAL GRID SPACING IN CM IN EACH LAYER IX IN CM\*\*2/M  
1355 C  
1356 C SICF INSOLATION ADJUSTMENT DUE TO ZENITH ANGLE, SURFACE SLOPE  
1357 C AND SURFACE ASPECT ANGLE. SICF=COS(Z)\*COS(SLOPE)+SIN(Z)  
1358 C SIN(SLOPE)\*COS(SAZ-SURFAC)  
1359 C  
1360 C SIGMA STEFAN-BOLTZMANN CONSTANT  
1361 C 5.67E-8 W/(m\*\*2-K\*\*4), OR  
1362 C 8.12E-11 cal/(min-cm\*\*2-K\*\*4)  
1363 C  
1364 C SLOPE SURFACE SLOPE IN DEGREES WITH HORIZONTAL=0 DEGREE,  
1365 C USED IN SOLVING INSOLATION  
1366 C  
1367 C SMALLA ABSORBTIVITY OF SURFACE MATERIAL  
1368 C  
1369 C SPEED WIND SPEED IN CM/SEC  
1370 C  
1371 C STOR(1,IY) ESTIMATE SUB-LAYER TEMPERATURE IN DEGREE RANKINE  
1372 C  
1373 C STOR(2,IY) FK;HEAT CONDUCTIVITY OF SUB-LAYER IY IN CAL/MIN-CM-K  
1374 C  
1375 C STOR(3,IY) RHOC,FK/ALPH IN CAL/CM\*\*2-K  
1376 C  
1377 C STOR(4,IY) CONSTANT DIMENSIONLESS.  
1378 C  
1379 C STOR(5,IY) INITIAL SOIL TEMPERATURE IN DEGREE RANKINS  
1380 C OF INITIAL SOIL PROFILE  
1381 C  
1382 C STOR(6,IY) SAME AS STOR(2,IY)  
1383 C  
1384 C STOR(7,IY) SAME AS STOR(3,IY)

1385 C  
1386 C SUN CALCULATED INSOLATION VALUE.  
1387 C  
1388 C SURFAC SURFACE AZIMUTH IN DEGREE WITH SOUTH =0 DEGREE,USED  
1389 C IN SOLVING INSOLATION  
1390 C  
1391 C T SAME AS TIME  
1392 C  
1393 C TA AIR TEMPERATURE IN DEGREE RANKINE  
1394 C  
1395 C TAC AIR TEMPERATURE IN DEGREE CELSIUS  
1396 C  
1397 C TAK AIR TEMPERATURE IN DEGREE KELVIN  
1398 C  
1399 C TB THERMAL CONDUCTIVITY OF BOTTOM MATERIAL  
1400 C CAL/CM\*\*2-DEG C-MIN  
1401 C  
1402 C TFRQ TIME STEP IN MINUTES USED IN SOLVING HEAT FLOW  
1403 C  
1404 C THK(IX) LAYER THICKNESS IN CM OF LAYER IX  
1405 C  
1406 C TIME TIME IN HOURS IN WHICH MATERIAL TEMPERATURES  
1407 C ARE ESTIMATED  
1408 C  
1409 C TIMER SUN'S HOUR ANGLE IN RADIANS  
1410 C  
1411 C TOTTIM TOTAL NUMBER OF 24 HOUR REPETITIONS USED IN SOLVING  
1412 C HEAT FLOW  
1413 C  
1414 C TPRNT OUTPUT TIME PRINT FREQUENCY IN MINUTES  
1415 C  
1416 C TR TEMPERATURE OF AIRSPACE BENEATH BOTTOM MATERIAL.  
1417 C  
1418 C TSK MATERIAL SUB-LAYER TEMPERATURE IN DEGREES KELVIN  
1419 C  
1420 C TYME TIME IN HOURS USE INSOLATION CALCULATION  
1421 C  
1422 C VMSURF MOISTURE CONTENT OF SURFACE MATERIAL (DECIMAL)  
1423 C  
1424 C WATER THE AMOUNT OF PRECIPITAL WATER IN MILLIMETERS  
1425 C (MM) USED IN SOLVING INSOLATION.  
1426 C  
1427 C WET DEGREE OF SATURATION OF SURFACE MATERIAL (DECIMAL)  
1428 C  
1429 C XXX(J) DEPTH (IN CENTIMETERS) FOR  
1430 C INITIAL TEMPERATURE PROFILE  
1431 C  
1432 C YYY(J) INITIAL TEMPERATURE PROFILE VALUES, DEG C  
1433 C  
1434 C Z SOLAR ZENITH ANGLE.  $Z=\sin(\text{DECL})*\sin(\text{ELF})+\cos(\text{DECL})*$   
1435 C  $\cos(\text{ELF})*\cos(\text{TIMER})$   
1436 C  
1437 C ZASH HEIGHT OF WIND SPEED INDICATOR (CM)  
1438 C  
1439 C ZZA SURFACE TEMPERATURE OF MATERIAL IN DEGREE RANKINE  
1440 C

```

1441 C   ZZB   BOTTOM LAYER TEMPERATURE OF MATERIAL IN DEGREE RANKINE
1442 C
1443 C   **** NEW PARAMETERS UTILIZED FOR MULTIPLE-DAY SIMULATIONS: ****
1444 C
1445 C   ATEMP   AIR TEMPERATURE , DEG C
1446 C
1447 C   BPRESS   BAROMETRIC PRESSURE , MILLIBARS
1448 C
1449 C   CLDCOV   CLOUD COVER , PERCENT
1450 C
1451 C   CLDTYPE   CLOUD INDEX
1452 C
1453 C   DEGSAT   DEGREE OF SATURATION , DECIMAL
1454 C
1455 C   DEPTH   A VECTOR OF Z VALUES FOR ALL OF THE NODES
1456 C
1457 C   DT       TIME OF DAY IN DECIMAL HOURS
1458 C
1459 C   IPPRNT   NUMBER OF TIME STEPS BETWEEN HOURLY
1460 TEMPERATURE
1461 C   PROFILE OUTPUT ON SELECTED DAY: NSNGLDAY
1462 C
1463 C   IPRNT   NUMBER OF TIME STEPS BETWEEN SIMULATION OUTPUTS
1464 C   = TPRNT/TFRQ
1465 C
1466 C   JD       JULIAN DAY
1467 C
1468 C   LNUM     A VECTOR OF LAYER NUMBERS FOR ALL OF THE NODES
1469 C
1470 C   NDTs     INPUT DATA TIME SUBSCRIPT; VALUE OF 24 MEANS THAT THE
1471 C           CURRENT CALCULATION FALLS BETWEEN THE 24TH AND 25TH
1472 C           DATA STRINGS; USED AS SUBSCRIPT FOR
1473 C           DATA INTERPOLATIONS
1474 C
1475 C   NFIRST   THE INPUT DATA LINE NUMBER CONTAINING THE FIRST
1476 C           LINE OF DATA FOR THE SIMULATION. EQUALS 1 IF THE
1477 C           ENTIRE INPUT DATA FILE IS GOING TO BE USED.
1478 C
1479 C   NLDATA   NUMBER OF LINES OF WEATHER DATA
1480 C
1481 C   NPPRNT   ACCUMULATING NUMBER OF TIME STEPS BETWEEN
1482 C           PROFILE OUTPUTS
1483 C
1484 C   NPRNT   ACCUMULATING NUMBER OF TIME STEPS BETWEEN
1485 C           SIMULATION OUTPUTS
1486 C           RESET TO 1 AFTER EACH SIMULATION OUTPUT
1487 C
1488 C   NSINGLE   A FLAG FOR PERFORMING SINGLE-DAY SIMULATIONS
1489 C           =1 (OR NOT 0) IF DOING A SINGLE-DAY SIMULATION
1490 C           =0 FOR A MULTIDAY SIMULATION
1491 C
1492 C   NSNGLDAY THE JULIAN NUMBER FOR THE DAY CHOSEN FOR A
1493 C           SINGLE-DAY SIMULATION AND/OR 1-HR TEMPERATURE
1494 C           PROFILES
1495 C
1496 C   RELHUM   RELATIVE HUMIDITY , PERCENT

```

```

1497 C
1498 C      RTEMP1          RADIOMETRIC TEMPERATURE OF SOME SURFACE
1499 OBJECT , DEG C
1500 C
1501 C      RTEMP2          RADIOMETRIC TEMPERATURE OF 2ND SURFACE OBJECT
1502 ,      DEG C
1503 C
1504 C      SOLAR          SOLAR LOADING , W/M^2
1505 C
1506 C      STEMP1         SOIL TEMPERATURE AT SHALLOWEST DEPTH , DEG C
1507 C
1508 C      STEMP2         SOIL TEMPERATURE AT NEXT SHALLOWEST DEPTH , DEG
1509 C
1510 C
1511 C      STOR(8,I)      VECTOR OF DIFFUSIVITY VALUES FOR EACH NODE
1512 C
1513 C      VMOIS1         VOLUMETRIC MOISTURE CONTENT AT SHALLOWEST
1514 DEPTH , %
1515 C
1516 C      VMOIS2         VOLUMETRIC MOISTURE CONTENT AT NEXT
1517 SHALLOWEST DEPTH , %
1518 C
1519 C      WINDSP         WIND SPEED , M/S
1520 C
1521      END

```

# Appendix B

## Example Input File

---

```
1 midwestern.test.site.2004,11656,,,,,,,,,,,,,
2 150,0,17.29,95.5,973,0,0.06,1,0,0.32,12.8,16,20.55,23,16.59,19.01
3 150,100,17.44,94.973,0,0.183,1,0,0.32,12.8,15.8,19.86,22.28,16.12,18.28
4 150,200,16.08,96.3,972,0,0.016,1,0,0.318,12.7,15.7,19.2,21.66,14.84,17.71
5 150,300,15.77,96.9,972,0,0.051,1,0,0.318,12.7,15.6,18.72,21.09,14.54,17.35
6 150,400,14.89,98.6,972,0,0.002,1,0,0.318,12.7,15.5,18.19,20.58,14.04,16.89
7 150,500,14.1,98.4,972,0,0,1,0,0.315,12.6,15.4,17.7,20.1,13.52,16.49
8 150,600,13.94,99.2,972,9.27,0,1,0,0.315,12.6,15.3,17.29,19.65,13.42,16.22
9 150,700,16.67,99.4,972,88,0.154,1,0,0.318,12.7,15.3,18.86,19.75,18.73,19.07
10 150,800,20.26,88.8,972,244.9,0.584,1,0,0.32,12.8,15.3,21.64,20.72,25.66,20.86
11 150,900,21.25,87.9,972,295.8,1.256,1,0,0.323,12.9,15.4,24.4,22.53,28.69,21.91
12 150,1000,22.49,80.8,972,367.3,1.679,1,0,0.323,12.9,15.5,25.57,23.51,33.46,23.92
13 150,1100,24.27,73.2,971,931,1.654,1,0,0.323,12.9,15.6,28.24,24.46,42.1,27.51
14 150,1200,23.79,70.8,971,489.2,1.854,1,0,0.32,12.8,15.7,29.27,25.9,37.83,25.31
15 .....
16 .....
17 .....
18 .....
19 .....
20 .....
21 214,1619,32.13,48.09,-6999,710,0.923,1,0,0.19,7.59,14.35,19.45,22.39,35.93,32.91
22 214,1620,32.17,44.25,-6999,702,0.882,1,0,0.19,7.58,14.35,19.44,22.39,35.86,32.91
23 214,1621,32.18,48.19,-6999,694.3,0.48,1,0,0.19,7.58,14.35,19.44,22.39,35.83,32.92
24 214,1622,32.25,46.1,-6999,686.7,0.474,1,0,0.189,7.57,14.35,19.44,22.39,35.79,32.91
25 214,1623,32.35,46.47,-6999,682,0.776,1,0,0.189,7.57,14.34,19.44,22.39,35.77,32.9
26 214,1624,32.45,45.86,-6999,674.2,0.784,1,0,0.189,7.56,14.34,19.43,22.38,35.73,32.88
27 214,1625,32.3,48.61,-6999,669.9,0.982,1,0,0.189,7.56,14.33,19.43,22.38,35.7,32.9
28 End,,,,,,,,,,,,,
29 1,210,0,0,0,0,,,,,,,,,,,,,
30 0,0,0,0,0,0,,,,,,,,,,,,,
31 -0.0052,1,0,0,0,0,,,,,,,,,,,,,
32 0.007,0.4,0,0,0,0,,,,,,,,,,,,,
33 0,0,0,0,0,0,,,,,,,,,,,,,
34 0,0,0,0,0,0,,,,,,,,,,,,,
35 4,0,0,0,0,0,
36 2,0.2,0,0.4,0.0283,0,
37 8,0.5,0,0.4,0.0283,0,
38 40,1,0,0.4,0.0283,0,
39 200,5,0,0.4,0.0283,0,
40 0,0,0,0,0,0,
```

41 0,0,0,0,0,284020000  
42 0,0,0,0,0,  
43 25,0,0,0,0,  
44 50,10,0,0,0,  
45 1.5,4.5,0,0,0,  
46 0.04,30,8,300,0,0,

# Appendix C

## Graphical User Interface Macro

---

```
1 Public ndlines As Integer
2 Public numlines As Integer
3 Public nsingle As Integer
4 Public ndos As Integer
5 Public nbbflag As Integer
6 Public nsolarflag As Integer
7 Public julian1 As Integer
8 Public julian2 As Integer
9 Public filedesc As String
10 Public infilename As String
11 Public status As String
12
13 Private Sub CheckBox1_Click()
14 If CheckBox1.Value = True Then nbbflag = -1
15 End Sub
16
17 Private Sub CheckBox2_Click()
18 If CheckBox2.Value = True Then nbbflag = 0
19 End Sub
20
21 Private Sub CheckBox3_Click()
22 If CheckBox3.Value = True Then nbbflag = 1
23 End Sub
24
25 Private Sub CheckBox4_Click()
26 If CheckBox4.Value = True Then nsolarflag = 0
27 End Sub
28
29 Private Sub CheckBox5_Click()
30 If CheckBox5.Value = True Then nsolarflag = 1
31 End Sub
32
33
34
35 Private Sub fixeddosbox_Click()
36 ndos = 0
37 If fixeddosbox.Value = True Then ndos = 1
38 End Sub
39
40 Private Sub lastjulianbox_Change()
```

```

41
42 End Sub
43
44 Private Sub multidaybox_Click()
45 nsingle = 1
46 If multidaybox.Value = True Then nsingle = 0
47 End Sub
48
49
50 Private Sub singledaybox_Click()
51 nsingle = 0
52 If singledaybox.Value = True Then nsingle = 1
53 End Sub
54
55
56 Private Sub updatebutton_Click()
57 dt1box.Text = space1box.Text ^ 2 / (2 * (diffslp1box.Text * 40 + diffint1box.Text))
58 If nlayerbox.Text > 1 Then dt2box.Text = space2box.Text ^ 2 / (2 * (diffslp2box.Text * 40 +
59 diffint2box.Text))
60 If nlayerbox.Text > 2 Then dt3box.Text = space3box.Text ^ 2 / (2 * (diffslp3box.Text * 40 +
61 diffint3box.Text))
62 If nlayerbox.Text > 3 Then dt4box.Text = space4box.Text ^ 2 / (2 * (diffslp4box.Text * 40 +
63 diffint4box.Text))
64 If nlayerbox.Text > 4 Then dt5box.Text = space5box.Text ^ 2 / (2 * (diffslp5box.Text * 40 +
65 diffint5box.Text))
66 If nlayerbox.Text > 5 Then dt6box.Text = space6box.Text ^ 2 / (2 * (diffslp6box.Text * 40 +
67 diffint6box.Text))
68 totalthickbox.Text = Val(thick1box.Text) + Val(thick2box.Text) + Val(thick3box.Text) +
69 Val(thick4box.Text) + Val(thick5box.Text) + Val(thick6box.Text)
70
71 End Sub
72
73 Private Sub UserForm_Click()
74
75 End Sub
76
77 Private Sub variabledosbox_Click()
78 ndos = 1
79 If variabledosbox.Value = True Then ndos = 0
80 End Sub
81
82 Private Sub inputfilebutton_Click()
83 '
84 ' open a window for selecting the input file
85 '
86
87 Dim irow As Integer
88 Dim th1, th2, th3, th4, th5, th6
89 infile = Application.GetOpenFilename(filefilter:="csv files (*.csv),*.csv", Title:="Input Files")
90 Workbooks.OpenText Filename:=infile, DataType:=xlDelimited, Comma:=True
91 slashnum = InStrRev(infile, "\") + 1
92 infilename = Mid(infile, slashnum)
93 ndlines = Range("a1:aa20000").Find(what:="End").Row - 2
94 numlines = ndlines + 2
95 filedesc = Cells(1, 1)
96 julian1 = Cells(2, 1)

```

```

97 julian2 = Cells(ndlines + 1, 1)
98 datalinesbox.Text = ndlines
99 Cells(1, 2) = ndlines
100 firstjulianbox.Text = julian1
101 lastjulianbox.Text = julian2
102 filenamebox.Text = infile
103 filedescbox.Text = filedesc
104 '
105 'populate the userform with zeros when cells are blank
106 '
107 For i = 1 To 18
108 For j = 1 To 6
109 If IsEmpty(Cells(numlines + i, j)) Then Cells(numlines + i, j) = 0
110 Next j
111 Next i
112 '
113 If Cells(numlines + 1, 1) = 1 Then singledaybox.Value = True Else singledaybox.Value = False
114 If Cells(numlines + 1, 1) = 0 Then multidaybox.Value = True Else multidaybox.Value = False
115 nsngldaybox.Text = Cells(numlines + 1, 2)
116 If Cells(numlines + 2, 1) = 0 Then CheckBox4.Value = True Else CheckBox4.Value = False
117 If Cells(numlines + 2, 1) = 1 Then CheckBox5.Value = True Else CheckBox5.Value = False
118 surfslopebox.Text = Cells(numlines + 2, 2)
119 surfazbox.Text = Cells(numlines + 2, 3)
120 sitelatbox.Text = Cells(numlines + 2, 4)
121 emissslopebox.Text = Cells(numlines + 3, 1)
122 emissintercbox.Text = Cells(numlines + 3, 2)
123 absslopebox.Text = Cells(numlines + 4, 1)
124 absintercbox.Text = Cells(numlines + 4, 2)
125 If Cells(numlines + 5, 1) = 0 Then variabledosbox.Value = True Else variabledosbox.Value =
126 False
127 If Cells(numlines + 5, 1) = 1 Then fixeddosbox.Value = True Else fixeddosbox.Value = False
128 surfsatbox.Text = Cells(numlines + 5, 2)
129 folcoverbox.Text = Cells(numlines + 6, 1)
130 stomatresisbox.Text = Cells(numlines + 6, 2)
131 folemissbox.Text = Cells(numlines + 6, 3)
132 folabsbox.Text = Cells(numlines + 6, 4)
133 folheightbox.Text = Cells(numlines + 6, 5)
134 nlayerbox.Text = Cells(numlines + 7, 1)
135 thick1box.Text = Cells(numlines + 8, 1)
136 th1 = Cells(numlines + 8, 1)
137 space1box.Text = Cells(numlines + 8, 2)
138 diffslp1box.Text = Cells(numlines + 8, 3)
139 diffint1box.Text = Cells(numlines + 8, 4)
140 condslp1box.Text = Cells(numlines + 8, 5)
141 condint1box.Text = Cells(numlines + 8, 6)
142 thick2box.Text = Cells(numlines + 9, 1)
143 th2 = Cells(numlines + 9, 1)
144 space2box.Text = Cells(numlines + 9, 2)
145 diffslp2box.Text = Cells(numlines + 9, 3)
146 diffint2box.Text = Cells(numlines + 9, 4)
147 condslp2box.Text = Cells(numlines + 9, 5)
148 condint2box.Text = Cells(numlines + 9, 6)
149 thick3box.Text = Cells(numlines + 10, 1)
150 th3 = Cells(numlines + 10, 1)
151 space3box.Text = Cells(numlines + 10, 2)
152 diffslp3box.Text = Cells(numlines + 10, 3)

```

```

153 diffint3box.Text = Cells(numlines + 10, 4)
154 conds1p3box.Text = Cells(numlines + 10, 5)
155 condint3box.Text = Cells(numlines + 10, 6)
156 thick4box.Text = Cells(numlines + 11, 1)
157 th4 = Cells(numlines + 11, 1)
158 space4box.Text = Cells(numlines + 11, 2)
159 diffslp4box.Text = Cells(numlines + 11, 3)
160 diffint4box.Text = Cells(numlines + 11, 4)
161 conds1p4box.Text = Cells(numlines + 11, 5)
162 condint4box.Text = Cells(numlines + 11, 6)
163 thick5box.Text = Cells(numlines + 12, 1)
164 th5 = Cells(numlines + 12, 1)
165 space5box.Text = Cells(numlines + 12, 2)
166 diffslp5box.Text = Cells(numlines + 12, 3)
167 diffint5box.Text = Cells(numlines + 12, 4)
168 conds1p5box.Text = Cells(numlines + 12, 5)
169 condint5box.Text = Cells(numlines + 12, 6)
170 thick6box.Text = Cells(numlines + 13, 1)
171 th6 = Cells(numlines + 13, 1)
172 space6box.Text = Cells(numlines + 13, 2)
173 diffslp6box.Text = Cells(numlines + 13, 3)
174 diffint6box.Text = Cells(numlines + 13, 4)
175 conds1p6box.Text = Cells(numlines + 13, 5)
176 condint6box.Text = Cells(numlines + 13, 6)
177 totalthickbox.Text = th1 + th2 + th3 + th4 + th5 + th6
178
179 If Cells(numlines + 14, 1) = -1 Then
180 CheckBox1.Value = True
181 bbfluxbox.Text = Cells(numlines + 15, 1)
182 Else: CheckBox1.Value = False
183 End If
184 If Cells(numlines + 14, 1) = 0 Then
185 CheckBox2.Value = True
186 bbtempbox.Text = Cells(numlines + 15, 1)
187 Else: CheckBox2.Value = False
188 End If
189 If Cells(numlines + 14, 1) = 1 Then
190 CheckBox3.Value = True
191 rbbfluxbox.Text = Cells(numlines + 15, 1)
192 rbbemiss1box.Text = Cells(numlines + 15, 2)
193 rbbf1box.Text = Cells(numlines + 15, 3)
194 rbbemiss2box.Text = Cells(numlines + 15, 4)
195 rbbf2box.Text = Cells(numlines + 15, 5)
196 rbbtempbox.Text = Cells(numlines + 15, 6)
197 Else: CheckBox3.Value = False
198 End If
199 depthfmbbox.Text = Cells(numlines + 16, 1)
200 fixedmoistbox.Text = Cells(numlines + 16, 2)
201 col11depthbox.Text = Cells(numlines + 17, 1)
202 col12depthbox.Text = Cells(numlines + 17, 2)
203 timeincbox.Text = Cells(numlines + 18, 1)
204 outputintbox.Text = Cells(numlines + 18, 2)
205 iterationsbox.Text = Cells(numlines + 18, 3)
206 windheightbox.Text = Cells(numlines + 18, 4)
207 End Sub
208

```

```

209 Private Sub exectstmbutton_Click()
210 '
211 ' load all of the chosenrun parameters into the end of
212 ' the input file, save that file, and execute the fortran code
213 '
214 Cells(numlines + 1, 1) = nsingle
215 Cells(numlines + 1, 2) = nsngldaybox.Text
216 Cells(numlines + 2, 1) = nsolarflag
217 Cells(numlines + 2, 2) = surfslopebox.Text
218 Cells(numlines + 2, 3) = surfazbox.Text
219 Cells(numlines + 2, 4) = sitelatbox.Text
220 Cells(numlines + 3, 1) = emissslopebox.Text
221 Cells(numlines + 3, 2) = emissintercbox.Text
222 Cells(numlines + 4, 1) = absslopebox.Text
223 Cells(numlines + 4, 2) = absintercbox.Text
224 Cells(numlines + 5, 1) = ndos
225 Cells(numlines + 5, 2) = surfsatbox.Text
226 Cells(numlines + 6, 1) = folcoverbox.Text
227 Cells(numlines + 6, 2) = stomatresisbox.Text
228 Cells(numlines + 6, 3) = folemissbox.Text
229 Cells(numlines + 6, 4) = folabsbox.Text
230 Cells(numlines + 6, 5) = folheightbox.Text
231 Cells(numlines + 7, 1) = nlayerbox.Text
232 Cells(numlines + 8, 1) = thick1box.Text
233 Cells(numlines + 8, 2) = space1box.Text
234 Cells(numlines + 8, 3) = diffslp1box.Text
235 Cells(numlines + 8, 4) = diffint1box.Text
236 Cells(numlines + 8, 5) = conds1p1box.Text
237 Cells(numlines + 8, 6) = condint1box.Text
238 Cells(numlines + 9, 1) = thick2box.Text
239 Cells(numlines + 9, 2) = space2box.Text
240 Cells(numlines + 9, 3) = diffslp2box.Text
241 Cells(numlines + 9, 4) = diffint2box.Text
242 Cells(numlines + 9, 5) = conds1p2box.Text
243 Cells(numlines + 9, 6) = condint2box.Text
244 Cells(numlines + 10, 1) = thick3box.Text
245 Cells(numlines + 10, 2) = space3box.Text
246 Cells(numlines + 10, 3) = diffslp3box.Text
247 Cells(numlines + 10, 4) = diffint3box.Text
248 Cells(numlines + 10, 5) = conds1p3box.Text
249 Cells(numlines + 10, 6) = condint3box.Text
250 Cells(numlines + 11, 1) = thick4box.Text
251 Cells(numlines + 11, 2) = space4box.Text
252 Cells(numlines + 11, 3) = diffslp4box.Text
253 Cells(numlines + 11, 4) = diffint4box.Text
254 Cells(numlines + 11, 5) = conds1p4box.Text
255 Cells(numlines + 11, 6) = condint4box.Text
256 Cells(numlines + 12, 1) = thick5box.Text
257 Cells(numlines + 12, 2) = space5box.Text
258 Cells(numlines + 12, 3) = diffslp5box.Text
259 Cells(numlines + 12, 4) = diffint5box.Text
260 Cells(numlines + 12, 5) = conds1p5box.Text
261 Cells(numlines + 12, 6) = condint5box.Text
262 Cells(numlines + 13, 1) = thick6box.Text
263 Cells(numlines + 13, 2) = space6box.Text
264 Cells(numlines + 13, 3) = diffslp6box.Text

```

```

265 Cells(numlines + 13, 4) = diffint6box.Text
266 Cells(numlines + 13, 5) = conds1p6box.Text
267 Cells(numlines + 13, 6) = condint6box.Text
268 Cells(numlines + 13, 7) = totalthickbox.Text
269 Cells(numlines + 14, 1) = nbbflag
270 Cells(numlines + 15, 1) = rbbfluxbox.Text
271 Cells(numlines + 15, 2) = rbbemiss1box.Text
272 Cells(numlines + 15, 3) = rbbf1box.Text
273 Cells(numlines + 15, 4) = rbbemiss2box.Text
274 Cells(numlines + 15, 5) = rbbf2box.Text
275 Cells(numlines + 15, 6) = rbbtempbox.Text
276 If nbbflag = -1 Then Cells(numlines + 15, 1) = bbfluxbox.Text
277 If nbbflag = 0 Then Cells(numlines + 15, 1) = bbtempbox.Text
278 Cells(numlines + 16, 1) = depthfmbbox.Text
279 Cells(numlines + 16, 2) = fixedmoistbox.Text
280 Cells(numlines + 17, 1) = col11depthbox.Text
281 Cells(numlines + 17, 2) = col12depthbox.Text
282 Cells(numlines + 18, 1) = timeincbox.Text
283 Cells(numlines + 18, 2) = outputintbox.Text
284 Cells(numlines + 18, 3) = iterationsbox.Text
285 Cells(numlines + 18, 4) = windheightbox.Text
286 '
287 ' save a new input file for TSTM as "fort.2"
288 '
289 ActiveWorkbook.SaveCopyAs Filename:="fort.2"
290 '
291 ' execute TSTM
292 '
293 Shell ("c:\tstm_files\tstmforgui.exe")
294 '
295 ' wait for the fortran code to finish executing before displaying results
296 '
297 msg = MsgBox("Wait for TSTM to finish executing!", vbOKOnly)
298 '
299 ' hide the TSTM userform
300 '
301 Executing_TSTM.Hide
302 '
303 ' display output
304 '
305 ' begin by importing the results of TSTM simulation
306 ' found in "fort.4" into this Excel file
307 '
308 Windows("TSTM simulation results template.xls").Activate
309 Sheets("simulation output data").Select
310 Range(Cells(1, 1), Cells(3400, 18)).Clear
311 Workbooks.OpenText Filename:= _
312     "fort.4", Origin:= _
313     xlWindows, StartRow:=1, DataType:=xlDelimited, TextQualifier:= _
314     xlDoubleQuote, ConsecutiveDelimiter:=True, Tab:=True, Semicolon:=False, _
315     Comma:=False, Space:=True, Other:=False, FieldInfo:=Array(Array(1, 1), _
316     Array(2, 1))
317 ' transfer the "fort.4" sheet to the output data sheet of this file
318 Range("1:3400").Select
319 Selection.Copy
320 Windows("TSTM simulation results template.xls").Activate

```

```

321 Sheets("simulation output data").Select
322 Range("A1").Select
323 ActiveSheet.Paste
324 ' copy the profiles data to another data sheet
325 '
326 ' determine which lines to cut and paste by counting
327 ' output lines until reaching "NORMAL TERMINATION"
328 '
329 irow = Range("a1:aa3400").Find(what="NORMAL").Row
330
331 Range(Cells(irow + 1, 2), Cells(3400, 28)).Select
332 Selection.Copy
333 Sheets("profiles data").Select
334 Range("a2").Select
335 ActiveSheet.Paste
336 Sheets("simulation output data").Select
337 Range(Cells(irow + 1, 1), Cells(3400, 28)).Clear
338 Range("k50").Select
339
340 copycoln = InputBox("Which column contains the measured surface temperature?", vbOKOnly)
341 colname = copycoln & ":" & copycoln
342 Columns(colname).Select
343 Selection.Copy
344 Columns("r:r").Select
345 ActiveSheet.Paste
346 Range("k50").Select
347 copycoln = InputBox("Which column contains the difference between measured and simulated
348 temperatures?", vbOKOnly)
349 colname = copycoln & ":" & copycoln
350 Columns(colname).Select
351 Selection.Copy
352 Columns("s:s").Select
353 ActiveSheet.Paste
354 '
355 ' calculate the average and population standard deviation of column "s"
356 '
357 Range("t59").Select
358 ActiveCell.FormulaR1C1 = "=average(r[0]c[-1]:r[3400]c[-1])"
359 Selection.NumberFormat = "0.000"
360 Range("t60").Select
361 ActiveCell.FormulaR1C1 = "=stdevp(r[-1]c[-1]:r[3399]c[-1])"
362 Selection.NumberFormat = "0.000"
363 ' print the selected day temperature prediction chart
364 Sheets("temperature chart").Select
365 nsd = nsngldaybox.Text
366 With ActiveChart.Axes(xlCategory)
367 .MinimumScale = nsd
368 .MaximumScale = nsd + 1
369 .MinorUnit = 0.04166666
370 .MajorUnit = 1
371 .Crosses = xlCustom
372 .CrossesAt = 0
373 .ReversePlotOrder = False
374 .ScaleType = xlLinear
375 .DisplayUnit = xlNone
376 End With

```

```

377 Application.CutCopyMode = False
378 ActiveWindow.SelectedSheets.PrintOut Copies:=1, Collate:=True
379 ' print the temperature prediction charts for all days
380 Sheets("temperature chart (2)").Select
381 ndaybegin = firstjulianbox.Text
382 ndaylast = lastjulianbox.Text
383 ndayhalf = ndaybegin + (ndaylast - ndaybegin) / 2 + 1
384 nofiveday = Int((ndayhalf - ndaybegin) / 5) + 1
385 ndayhalf = ndaybegin + 5 * nofiveday
386 ndayend = ndaybegin + 10 * nofiveday
387 With ActiveChart.Axes(xlCategory)
388 .MinimumScale = ndaybegin
389 .MaximumScale = ndayhalf
390 .MinorUnit = 1
391 .MajorUnit = 5
392 .Crosses = xlCustom
393 .CrossesAt = 0
394 .ReversePlotOrder = False
395 .ScaleType = xlLinear
396 .DisplayUnit = xlNone
397 End With
398 ' Application.CutCopyMode = False
399 ' ActiveWindow.SelectedSheets.PrintOut Copies:=1, Collate:=True
400 Sheets("temperature chart (3)").Select
401 With ActiveChart.Axes(xlCategory)
402 .MinimumScale = ndayhalf
403 .MaximumScale = ndayend
404 .MinorUnit = 1
405 .MajorUnit = 5
406 .Crosses = xlCustom
407 .CrossesAt = 0
408 .ReversePlotOrder = False
409 .ScaleType = xlLinear
410 .DisplayUnit = xlNone
411 End With
412 ' print the temperature profiles chart for the selected day
413 Sheets("profiles chart").Select
414 Application.CutCopyMode = False
415 ActiveWindow.SelectedSheets.PrintOut Copies:=1, Collate:=True
416 Sheets("pred-meas").Select
417 With ActiveChart.Axes(xlCategory)
418 .MinimumScale = nsd
419 .MaximumScale = nsd + 1
420 .MinorUnit = 0.04166666
421 .MajorUnit = 1
422 .Crosses = xlCustom
423 .CrossesAt = 0
424 .ReversePlotOrder = False
425 .ScaleType = xlLinear
426 .DisplayUnit = xlNone
427 End With
428 ' print the predicted-measured temperature charts for all days
429 Sheets("pred-meas (2)").Select
430 With ActiveChart.Axes(xlCategory)
431 .MinimumScale = ndaybegin
432 .MaximumScale = ndayhalf

```

```

433     .MinorUnit = 1
434     .MajorUnit = 5
435     .Crosses = xlCustom
436     .CrossesAt = 0
437     .ReversePlotOrder = False
438     .ScaleType = xlLinear
439     .DisplayUnit = xlNone
440     End With
441     Sheets("pred-meas (3)").Select
442     With ActiveChart.Axes(xlCategory)
443         .MinimumScale = ndayhalf
444         .MaximumScale = ndayend
445         .MinorUnit = 1
446         .MajorUnit = 5
447         .Crosses = xlCustom
448         .CrossesAt = 0
449         .ReversePlotOrder = False
450         .ScaleType = xlLinear
451         .DisplayUnit = xlNone
452     End With
453     ' Application.CutCopyMode = False
454     ' ActiveWindow.SelectedSheets.PrintOut Copies:=1, Collate:=True
455     ' print the selected day flux prediction chart
456     Sheets("flux chart").Select
457     With ActiveChart.Axes(xlCategory)
458         .MinimumScale = nsd
459         .MaximumScale = nsd + 1
460         .MinorUnit = 0.04166666
461         .MajorUnit = 1
462         .Crosses = xlCustom
463         .CrossesAt = 0
464         .ReversePlotOrder = False
465         .ScaleType = xlLinear
466         .DisplayUnit = xlNone
467     End With
468     ' Application.CutCopyMode = False
469     ' ActiveWindow.SelectedSheets.PrintOut Copies:=1, Collate:=True
470     ' print the flux charts for all days
471     Sheets("flux chart (2)").Select
472     With ActiveChart.Axes(xlCategory)
473         .MinimumScale = ndaybegin
474         .MaximumScale = ndayhalf
475         .MinorUnit = 1
476         .MajorUnit = 5
477         .Crosses = xlCustom
478         .CrossesAt = 0
479         .ReversePlotOrder = False
480         .ScaleType = xlLinear
481         .DisplayUnit = xlNone
482     End With
483     Sheets("flux chart (3)").Select
484     With ActiveChart.Axes(xlCategory)
485         .MinimumScale = ndayhalf
486         .MaximumScale = ndayend
487         .MinorUnit = 1
488         .MajorUnit = 5

```

```

489     .Crosses = xlCustom
490     .CrossesAt = 0
491     .ReversePlotOrder = False
492     .ScaleType = xlLinear
493     .DisplayUnit = xlNone
494     End With
495 ' return to the 1st day temperature chart
496 Sheets("temperature chart").Select
497 '
498 ' pause to look at single-day temperature results before continuing
499 '
500 '
501 ' choose to save the excel output file and the input file under new names
502 '
503 msg = MsgBox("Do you want to save the output file under a new name?", vbYesNo)
504 If msg = 6 Then
505     filesavename = Application.GetSaveAsFilename(filefilter:="Excel files(*.xls),*.xls", Title:="Save
506 Output File Under a New Name")
507     If filesavename <> False Then ActiveWorkbook.SaveCopyAs Filename:=filesavename
508     Else
509     End If
510     Windows(infilename).Activate
511 msg = MsgBox("Do you want to save the .csv input file under a new name?", vbYesNo)
512 If msg = 6 Then
513     filesavename = Application.GetSaveAsFilename(filefilter:="csv files(*.csv),*.csv", Title:="Save
514 Modified Input File (csv format)")
515     If filesavename <> False Then ActiveWorkbook.SaveCopyAs Filename:=filesavename
516     Else
517     End If
518     Windows("TSTM simulation results template.xls").Activate
519 '
520 '
521 msg = MsgBox("Do you want to do another simulation?", vbYesNo)
522 If msg = 6 Then
523     Windows("fort.4").Activate ' close the file called "fort.4"
524     ActiveWorkbook.Close
525     Windows(infilename).Activate
526     Executing_TSTM.Show ' reveal the TSTM Graphical User Interface
527     Else
528     End If
529 '
530 ' close all files except the output template file
531 '
532     Windows(infilename).Activate
533     ActiveWorkbook.Close
534     Windows("fort.4").Activate
535     ActiveWorkbook.Close
536 End Sub

```

# REPORT DOCUMENTATION PAGE

*Form Approved*  
*OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

<b>1. REPORT DATE (DD-MM-YYYY)</b> February 2006		<b>2. REPORT TYPE</b> Final report		<b>3. DATES COVERED (From - To)</b>	
<b>4. TITLE AND SUBTITLE</b>  1-D Thermal Modeling of Layered Materials in Outdoor Environments				<b>5a. CONTRACT NUMBER</b>	
				<b>5b. GRANT NUMBER</b>	
				<b>5c. PROGRAM ELEMENT NUMBER</b>	
<b>6. AUTHOR(S)</b>  John O. Curtis				<b>5d. PROJECT NUMBER</b>	
				<b>5e. TASK NUMBER</b>	
				<b>5f. WORK UNIT NUMBER</b>	
<b>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</b>  Environmental Laboratory, U.S. Army Engineer Research and Development Center, 3909 Halls Ferry Road, Vicksburg, MS 39180-6199				<b>8. PERFORMING ORGANIZATION REPORT NUMBER</b>  ERDC/EL TR-06-2	
<b>9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)</b> U.S. Army Night Vision and Electronic Sensors Directorate (NVESD), 10221 Burbeck Road, Fort Belvoir, VA 22060-5806				<b>10. SPONSOR/MONITOR'S ACRONYM(S)</b>	
				<b>11. SPONSOR/MONITOR'S REPORT NUMBER(S)</b>	
<b>12. DISTRIBUTION / AVAILABILITY STATEMENT</b> Approved for public release; distribution is unlimited.					
<b>13. SUPPLEMENTARY NOTES</b>					
<b>14. ABSTRACT</b>  This report describes a physics-based, but relatively simple, thermal infrared computational tool that can be used in a personal computer (PC) environment to predict temperature profiles of layered media. The tool is a one-dimensional finite difference simulation code (written in FORTRAN) that is executed through a graphical user interface. Its current utility is in helping to design field tests of airborne mine detection sensor systems and to analyze the results of those tests to better understand the performance of the sensor in different environmental conditions. The tool does not address the separate issues of two- and three-dimensional effects, sensor hardware performance, sensor flight paths, or targeting algorithms.					
<b>15. SUBJECT TERMS</b> Finite difference code Graphical user interface		Heat flow One-dimensional thermal model Temperature model		Terrain surface Thermal signatures	
<b>16. SECURITY CLASSIFICATION OF:</b>			<b>17. LIMITATION OF ABSTRACT</b>	<b>18. NUMBER OF PAGES</b>  92	<b>19a. NAME OF RESPONSIBLE PERSON</b>
<b>a. REPORT</b>  UNCLASSIFIED	<b>b. ABSTRACT</b>  UNCLASSIFIED	<b>c. THIS PAGE</b>  UNCLASSIFIED			<b>19b. TELEPHONE NUMBER (include area code)</b>