User Manual for *SlugHeat*

A software for processing data collected by a heat-flow measurement system

University of California at Santa Cruz

Earth and Planetary Sciences Department

Written by Kristin Dickerson in March 2023

Contents

1 Introduction

This is a comprehensive user manual for setting up and processing data collected by a heat-flow measurement system using the MATLAB program, *SlugHeat*. This manual corresponds with the 2023 update of *SlugHeat*. ASCII long-files that record all data collected by the heat-flow probe, including multiple penetrations, should first be run through *SlugPen* (a parsing software used to generate individual data files for each penetration separately, which can then be run through *SlugHeat*. To use the parsing software, see SlugPen for code and SlugPen user manual

Processing steps in *SlugHeat* include defining input parameters, calculating equilibrium temperatures of sediment with depth, calculating *in-situ* thermal conductivity values of sediment with depth, and calculating heat flow at the seafloor. *SlugHeat* also includes multiple optional uncertainty assessments including an analysis to test sensitivity of results with varying layer properties and analysis of uncertainty in the linear regressions that are used to calculate heat flow. There are numerous opportunities for inspection and modification of data and results before, during, and after processing. All processing results and input parameters are saved to text files, .mat files, and optional Excel (.xlsx) files.

Included in this manual is an outline of the steps through a general workflow in *SlugHeat* for a single penetration. The example penetration `ExamplePen_SlugHeatTutorial` will be used. The end of this manual also includes details regarding the *SlugHeat* code, including description of all variables, functions, and callbacks.

2 System requirements

2.1 System requirements for running in MATLAB

1. Updated MATLAB software: Version 2023a or later preferred
2. Correct directory in MATLAB Path. This directory must include all of the following functions and folders to run *SlugHeat* correctly:
   (a) Main application (`SlugHeat`)
   (b) All functions (as `.m` files), including:
      - cbValueChange.m
      - ChiSquaredFit.m
      - DiscardSensorsNoHP.m
      - FrictionalDecay.m
      - GetFiles.m
      - HeatFlowAnalysis.m
      - HeatFlowRegression.m
      - HeatPulseDecay.m
      - InitializeProcessing.m
      - InitializeProgram.m
      - PlotCheckboxes.m
      - PlotFrictionalDecay.m
      - PlotHeatFlowAnalysis.m
      - PlotHeatPulseDecay.m
      - PlotHeatFlowRegression.m
      - plotLayout.m
      - PlotTemp.m
      - PlotDepth.m

- PlotTilt.m
- PlotPolynomial.m
- PrintBullardResults.m
- PrintFricResults.m
- PrintHeaderResults.m
- PrintHeatPulseResults.m
- PrintNewPar.m
- PrintParametersResults.m
- PrintSensResults.m
- PrintStatus.m
- ReadCalFile.m
- ReadParFile.m
- ReadPenFile_withPulse.m
- ReadPenText_withPulse.m
- ReadTAPFile.m
- ReadTAPText.m
- ResetAll.m
- SensitivityAnalysis.m
- SplitDecays.m
- TempCorrection.m
- updateLabels.m
- xAlign.m
- xAlignHPD.m

(c) Default input files:

- PAR file SlugHeat22.par

(d) Auxiliary applications (as `.mlapp` files)

- BottomWaterAuxApp.mlapp
- SetParams.mlapp
- SetUpSensAnalysis.mlapp
- UserBWTAuxApp.mlapp
- DiscardData.mlap

(e) Graphics subfolder (`images`), including:

- arrow.png
- slugheatLogo.png
- UCSC-HydrogeologyLogo.jpg
- x.png

(f) inputs subfolder (`inputs`), with files created in *SlugPen* including:

- input text files (.pen and .tap), OR
- input MAT file (.mat)

2.2 System requirements for running compiled code in MATLAB Runtime

1. Compiled software
2. MATLAB Runtime, corresponding with version of compiled software

3 System set up

3.1 Set up for running in MATLAB

1. **Add current directory and all subdirectories to MATLAB path**

To set up the system, you must first ensure that you are in the correct directory, with all functions and subdirectories needed to run *SlugHeat* (listed in section above). It is best to also add these functions and subdirectories to your MATLAB path. There are a few ways to do this once you are in the correct directory and in the MATLAB Command Window, as follows.

To *temporarily* add current directory to MATLAB path:
In MATLAB Command Window:

```
>> addpath(genpath(pwd))
```

To *permanently* add current directory to MATLAB path:
In MATLAB toolbar at top right of application:

(a) Select *'Set Path'* in MATLAB toolbar
(b) Select *'Add with Subfolders...'*
(c) Open current directory
(d) Press *'Save'*

To check that correct directory and all subfolders are in your MATLAB path:
In MATLAB Command Window:

```
>> path
```

2. **Launch *SlugHeat***

Once you ensure all necessary functions and subdirectories are in your current working directory or in your MATLAB path, you can launch the program.

In MATLAB command window:

```
>> SlugHeat
```

3.2 Set up for running compiled code with MATLAB Runtime

1. **Install compiled applications and MATLAB Runtime**

Install *SlugHeat* using the installer found in the software's GitHub repository, here.

Available installers include:

- SlugheatInstaller_macOS-Silicon
- SlugheatInstaller_macOS-Intel
- SlugheatInstaller_Windows
- SlugheatInstaller_Linux

Please follow the instructions provided by the Installer and store the compiled software and MATLAB Runtime on your local drive. This will automatically download the corresponding version of MATLAB Runtime from the web if you do not already have the program on your computer.

2. **Add application and MATLAB Runtime to your computer's path.**

   macOS - Silicon

   To add the compiled software to your path, in Terminal:

```
export PATH="/Applications/slugheat/application/slugheat.app/Contents/MacOS:$PATH"
```

   To add MATLAB Runtime, in Terminal:

```
export DYLD_LIBRARY_PATH="${DYLD_LIBRARY_PATH:+${DYLD_LIBRARY_PATH}:}\
/Applications/MATLAB/MATLAB_Runtime/R2023b/runtime/maca64:
/Applications/MATLAB/MATLAB_Runtime/R2023b/bin/maca64:
/Applications/MATLAB/MATLAB_Runtime/R2023b/sys/osmaca64:
/Applications/MATLAB/MATLAB_Runtime/R2023b/extern/bin/maca64"
```

3. **Launch application.**

   macOS - Silicon

   In Terminal:

```
slugheat
```

Whether using MATLAB command window or MATLAB Runtime, the *SlugHeat* application immediately displays all tabs used for processing, though no data is loaded in or plotted yet (Figure 2).

Command Window:

The command window on the left-hand side of the screen shows all penetration and processing information and controls. This includes input parameters, penetration information, data and plot controls, and processing commands. To close this window, press *Hide toolbar* at top left of screen. To re-open this window, press the *arrow* on the left of screen. *Restart* button will begin *SlugHeat* from beginning, resetting all plots and data to default properties before a penetration is loaded in. Each box on this command window defines data or plot controls, as follows:

Penetration Data:

Defines penetration data to be processed, including the name of the PEN file loaded in. See Figure 6 **(box 1)**

Input Parameters:

Controls parameters for data processing and calibration of temperature sensors, including files where default parameters are obtained. See Figure 6 **(box 2)**

Data Controls:

Controls processing specs such as timing of events that were defined in *SlugPen* (start of penetration, end of penetration, and firing of the calibrated heat pulse), bottom water temperature, and the use of a calibrated heat pulse to determine *in-situ* thermal conductivity. User can modify as needed. See Figure 6 **(boxes 3, 4, & 5)**

Plot Controls:

Controls which plots displayed and which are hidden. See Figure 6 **(box 6)**

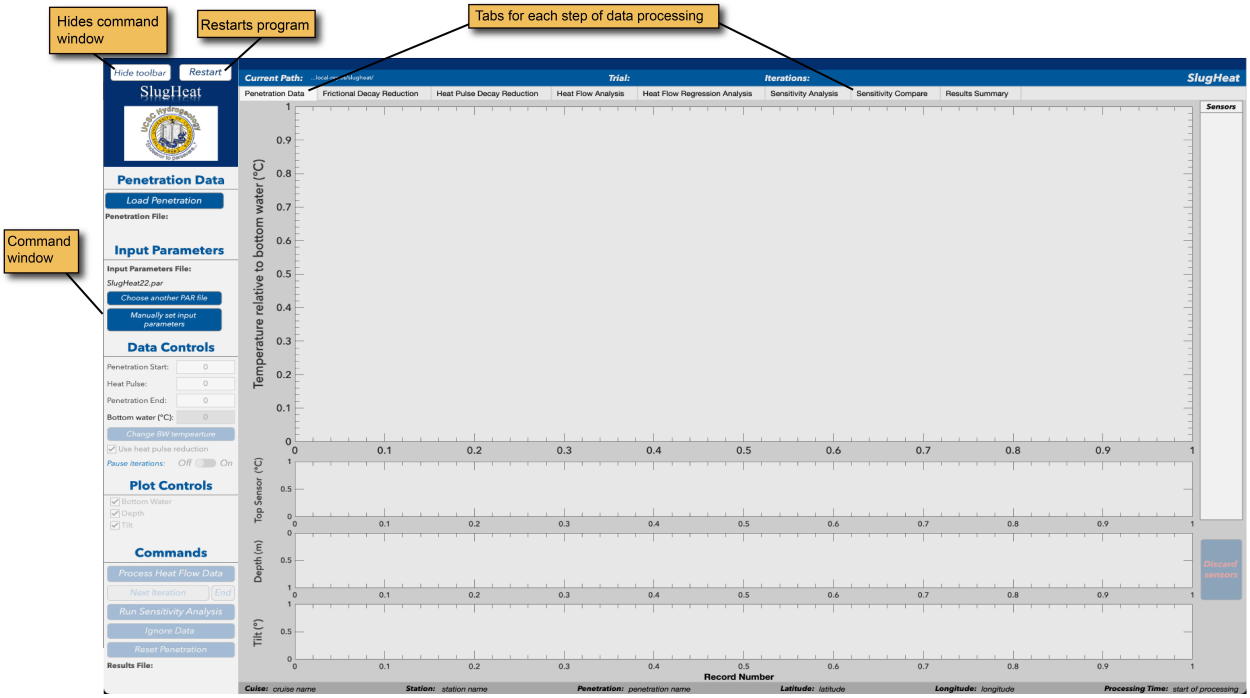


Figure 1: *SlugHeat* application launched. No penetration data is loaded in yet. Left hand side shows command window for data and plot control.

<u>*Commands:*</u>
All commands for processing data (*detailed in subsequent sections*). See Figure 10

Tabs:
The rest of the screen to the right of the command window shows each tab for plotting, inspecting, modifying data, and analyzing heat flow results (Figure 2). Tabs include results from each major step in the heat flow data processing, including: *Penetration Data*, *Frictional Decay Reduction*, *Heat Pulse Decay Reduction*, *Heat Flow Analysis*, *Heat Flow Regression Analysis*, *Sensitivity Analysis*, and *Results Summary*.
The *SlugHeat* application immediately displays all tabs used for processing, though no data is loaded in or plotted yet (Figure 2).

Command Window:
The command window on the left-hand side of the screen shows all penetration and processing information and controls. This includes input parameters, penetration information, data and plot controls, and processing commands. To close this window, press *Hide toolbar* at top left of screen. To re-open this window, press the *arrow* on the left of screen. *Restart* button will begin *SlugHeat* from beginning, resetting all plots and data to default properties before a penetration is loaded in. Each box on this command window defines data or plot controls, as follows:

<u>*Penetration Data:*</u>
Defines penetration data to be processed, including the name of the PEN file loaded in. See Figure 6 **(box 1)**

<u>*Input Parameters:*</u>
Controls parameters for data processing and calibration of temperature sensors, including files where default parameters are obtained. See Figure 6 **(box 2)**

<u>*Data Controls:*</u>
Controls processing specs such as timing of events that were defined in *SlugPen* (start of penetration, end of penetration, and firing of the calibrated heat pulse), bottom water temperature, and the use of a calibrated heat pulse to determine *in-situ* thermal conductivity. User can modify as needed. See Figure 6 **(boxes 3, 4, & 5)**

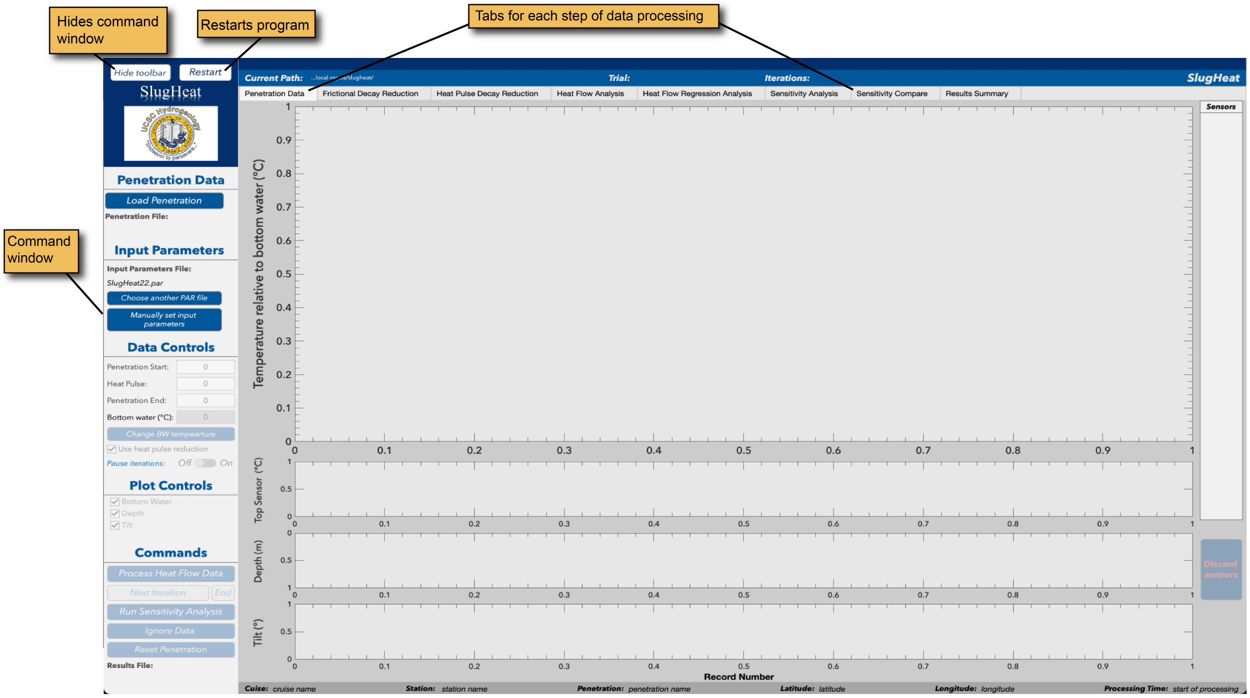


Figure 2: *SlugHeat* application launched. No penetration data is loaded in yet. Left hand side shows command window for data and plot control.

Plot Controls:
Controls which plots displayed and which are hidden. See Figure 6 **(box 6)**
Commands:
All commands for processing data (*detailed in subsequent sections*). See Figure 10

Tabs:
The rest of the screen to the right of the command window shows each tab for plotting, inspecting, modifying data, and analyzing heat flow results (Figure 2). Tabs include results from each major step in the heat flow data processing, including: *Penetration Data*, *Frictional Decay Reduction*, *Heat Pulse Decay Reduction*, *Heat Flow Analysis*, *Heat Flow Regression Analysis*, *Sensitivity Analysis*, and *Results Summary*.

4 Load in penetration data

Next, load in data recorded by the heat-flow probe from an individual penetration. This could be via:

- PEN (*.pen)* file: a penetration (PEN) file is a text file that defines all recorded temperatures as well as timings of significant events during the penetration, such as start and end of the penetration and the firing of a calibrated heat pulse. This might also include metadata such cruise, station, penetration, location, instrument specifications, and power of heat pulse used (in J).

- TAP (*.tap)* file: a temperature and pressure (TAP) text file is a text file that defines all recorded tilt and pressure (i.e., depth) measurements from a single penetration. This often corresponds with a PEN file.

- MAT (*.mat)* file: a MATLAB (MAT) file is a binary file that stores workspace variables. Data from a MAT file consists of all information generally recorded by both a PEN and a TAP file, such as temperatures, tilt, and pressures for an entire penetration. Data stored in this MAT file is already formatted for MATLAB and therefore does not require tedious reading in of text files.

Though PEN, TAP, and MAT files are all created in *SlugPen*, the minimum requirement for processing data in *SlugHeat* is a PEN file or a MAT file. *SlugHeat* will first ask you to select a PEN file that defines the penetration. Corresponding TAP and MAT files for that penetration will also be loaded in automatically if they exist in the same directory. If they do not exist, *SlugHeat* can still process temperature data recorded by a PEN file alone. In this example, penetration `ExamplePen_SlugHeatTutorial` is loaded using PEN, TAP, and MAT files (See Figure 4).

IMPORTANT: If your PEN file was *not* created in the 2022 version of *SlugPen*, you *must* check format of the PEN file. File format has likely changed and must be corrected to match the format required by *SlugHeat*. Any data or metadata that is missing or unknown will be recorded as '-999' in the PEN, MAT, and TAP files. To see example format, see Figure 3

General format for .pen file header:

Station #	**Penetration #**	**'Cruise'**	**Datum**		
Lat	**Lon**	**Av. Depth**	**Av. Tilt**		
Logger ID	**Probe ID**	**Num. Sensors**	**Pulse power (J)**		
	Pen record				
	HP record				
		Cal1	**Cal2**	**Cal3**	**Cal4**	**......**
	Record #	**T1**	**T2**	**T3**	**T4**	**......**

Example .pen file:

```
   0    1 'TutorialCruise' datum
418880 2063970   2234   2.00
  5301    203     11      -999
    13
    77
         2.584  2.634  2.583  2.567  2.548  2.591  2.616  2.589  2.602  2.548  2.594  2.695
     1   2.584  2.633  2.581  2.565  2.545  2.587  2.613  2.588  2.602  2.549  2.594  2.694
     2   2.584  2.633  2.582  2.565  2.545  2.588  2.614  2.589  2.603  2.548  2.594  2.694
     3   2.584  2.633  2.582  2.564  2.545  2.588  2.615  2.589  2.602  2.548  2.594  2.694
     4   2.584  2.634  2.583  2.566  2.546  2.590  2.615  2.589  2.602  2.548  2.594  2.695
     5   2.584  2.634  2.583  2.567  2.547  2.590  2.615  2.590  2.602  2.548  2.594  2.696
     6   2.584  2.634  2.583  2.567  2.548  2.591  2.616  2.589  2.602  2.548  2.594  2.695
     7   2.580  2.629  2.579  2.560  2.543  2.585  2.610  2.582  2.597  2.543  2.587  2.702
```

Figure 3: *SlugHeat* example PEN file formatted correctly. PEN files that do not match this format cannot be loaded into the 2022 version of *SlugHeat*.

Steps for loading in data are as follows:

1. **Press *'Load Penetration'* button** to load in data from a single penetration.

   – Note: If you do not see the pop-up window, you may need to click anywhere on the screen after pressing *'Load Penetration'* button or look on other screens, if applicable. The pop-up window should become visible.

2. **Select a PEN file** in the pop-up window (Figure 4), which defines data from a single penetration.

   – Note: If you are using a PEN file *without* a corresponding MAT file and you wish to include a TAP file as well, the TAP file must be in the same folder as your PEN file.

– Note: If the PEN, TAP, and MAT files were created in the updated 2022 version of *SlugPen*, the power of the heat pulse (J), as recorded directly by the instrument, will be loaded in directly from the PEN

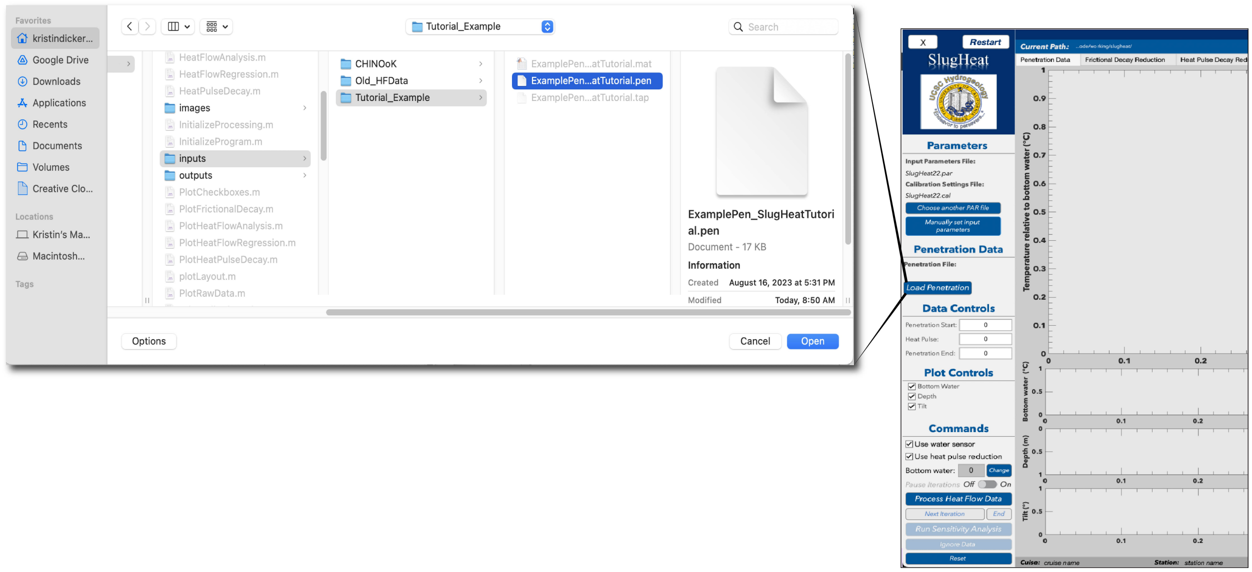


Figure 4: *SlugHeat* example pop-up window to select a PEN file that will be used to load the penetration into the program.

or MAT file. If there is not recorded pulse power in the PEN file (i.e., it is set to -999), the power of the heat pulse will be instead be the default input parameter loaded in by the PAR file.

Once the unprocessed penetration data is loaded in, four time series are plotted on the first tab (*Penetration Data* tab). These time series are shown in Figure 5: **(a)** temperature relative to bottom water; **(b)** temperature recorded by the top (bottom water) sensor that rests atop the data logger; **(c)** depth below seafloor; and **(d)** tilt of the probe relative to vertical. Time is displayed as record numbers, which begin at zero at the start of the penetration and increment with each measurement made by the probe. In plot **(a)**, each line color is a separate temperature sensor (thermistor), which line the probe and therefore vary with depth. Sensors with higher numbers are shallower and sensors with lower numbers are deeper (i.e, T1 is the deepest). The number of sensors could vary with different probe constructions. Data processing to calculate heat flow does not require top sensor, depth, or tilt measurements, but all are useful in data interpretation and are therefore recorded and plotted.

Display and plot controls

Once plotted, all displays are modifiable in the following ways:

(a) **Plot zoom, pan, and save**: Each plot has pan, zoom, and save capabilities. Save options include copying as an image or vector graphic. These capabilities can be used by hovering mouse over the top right corner of each plot. All plots on the *Penetration Data* tab are linked with the same x axis (Record Number), so panning and zooming on one plot will be mirrored on all other plots on this tab. See Figure 5

(b) **Plot display:** Checkboxes in the *Plot Controls* box in the command window indicate which plots are shown. Unchecked plots are hidden from display (Figure 6 **(box 6)**).

(c) **Temperature data display:** Checkboxes in the thermistors legend on the right-hand side of the screen indicate which thermistors' data will be displayed. Data from unchecked thermistors will be hidden from display, but are not removed from data file. Therefore, data from unchecked thermistors will still be used in subsequent calculations unless otherwise noted by user (described in next section). See Figure 5

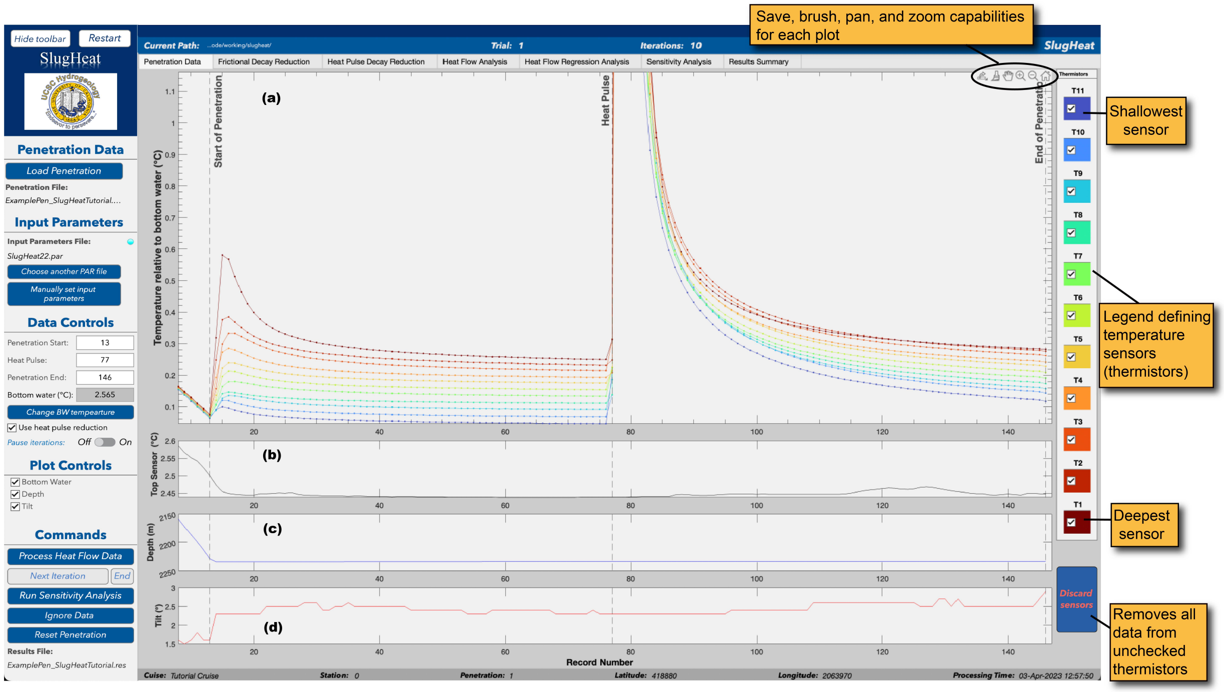


Figure 5: *SlugHeat* example *Penetration Data* tab where data has been loaded in but has not been processed. **(a)** Temperature relative to bottom water ($^{\circ}C$) vs. Record Number; **(b)** Top Sensor temperature ($^{\circ}C$) vs. Record Number; **(c)** Depth (m) vs. Record Number; and **(d)** Tilt ($^{\circ}$) vs. Record Number. Vertical lines indicate timing of start of penetration, heat pulse firing, and end of penetration. Yellow boxes indicate notable plot and data controls on main window. Example penetration used: `ExamplePen_SlugHeatTutorial`

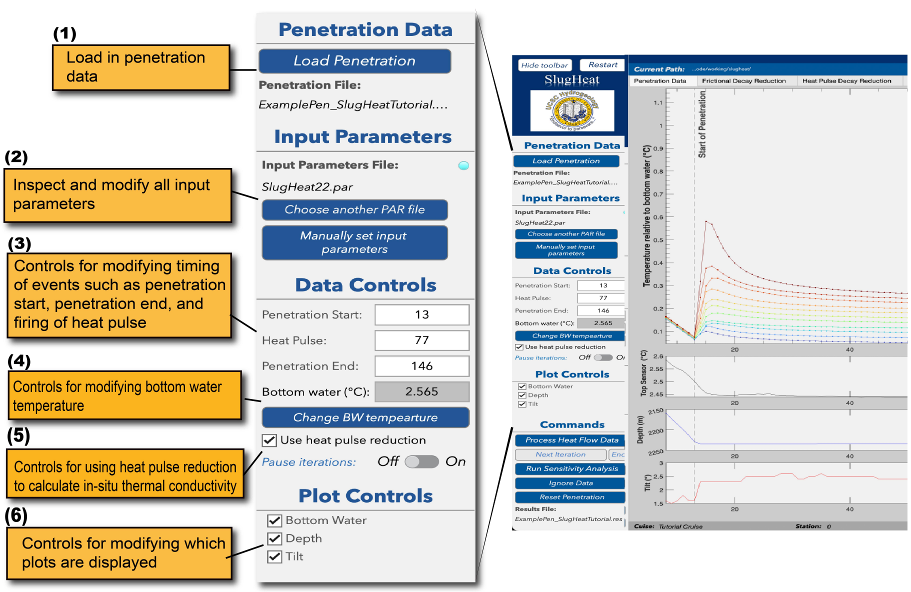


Figure 6: *SlugHeat* controls for modifying data, input parameters, and plot display before processing. Yellow boxes indicate notable pre-processing data and display controls. Example penetration used: `ExamplePen_SlugHeatTutorial`

5 Set input parameters

Once the program is launched and penetration data is loaded in, the input parameters for data processing and calibration of temperature sensors must be defined. To see default parameters and modify as needed, use the *Input Parameters* box on the command window (Figure 6 **(box 2)**).

1. **Check default parameters**

   – The default parameters for processing are read in automatically when the main program is launched using the default parameters text file (PAR file): `SlugHeat.par`

   – To check *all* default input processing parameters and calibration parameters before processing data, either (1) check `SlugHeat.par` in a separate text editor or (2) press *'Manually set input parameters'* button on the *SlugHeat* command window which will launch an auxiliary application within *SlugHeat* to view and adjust all input parameters. See Figure 7.

2. **Adjust input parameters, if needed** (two optional ways to do this):

   (a) **Press *'Manually set input parameters'* button** on top of command window to left of application, launching auxiliary application where individual input parameters can be adjusted manually. **Press *'Update Parameters'*** button to update modified values. See Figure 7.

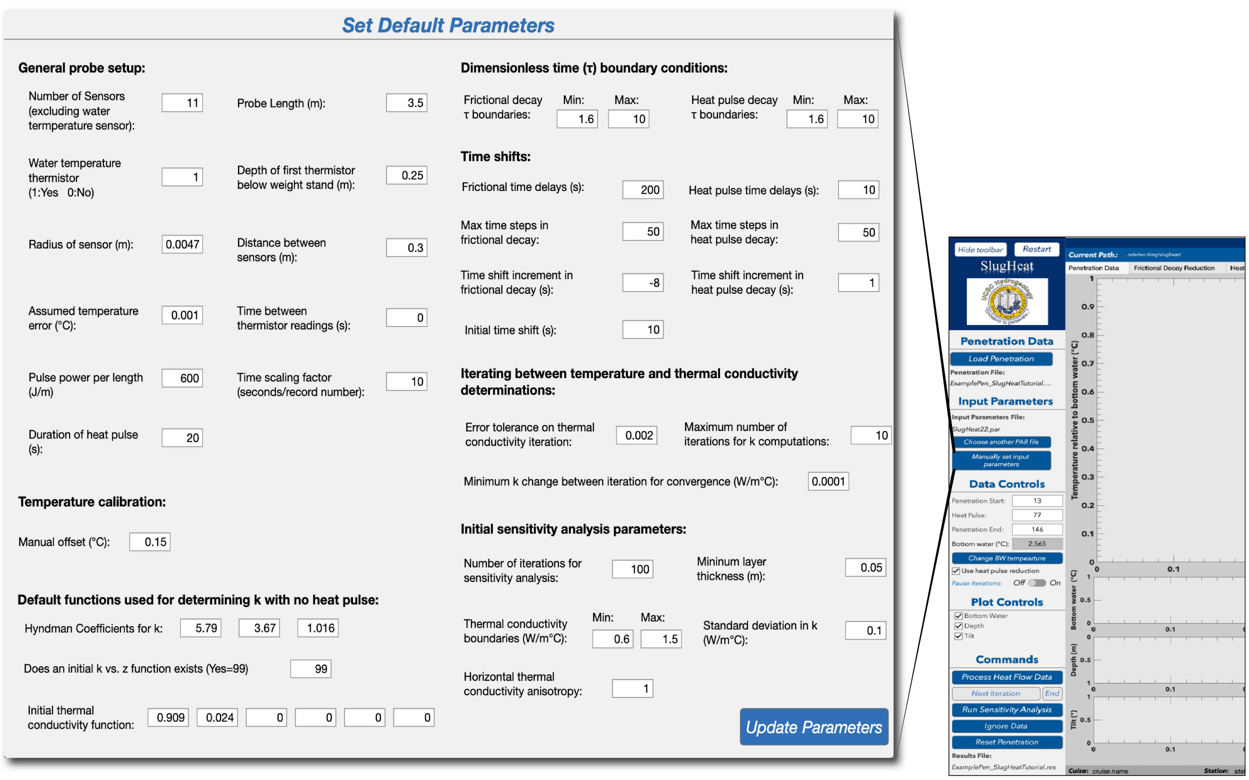


Figure 7: SlugHeat auxiliary application to inspect and modify input parameters

   (b) **Press *'Choose another PAR file'* button** and select another text file to be used instead of the default `SlugHeat.par` text file. This is often a PAR file saved from a previously processed penetration. See Figure 8

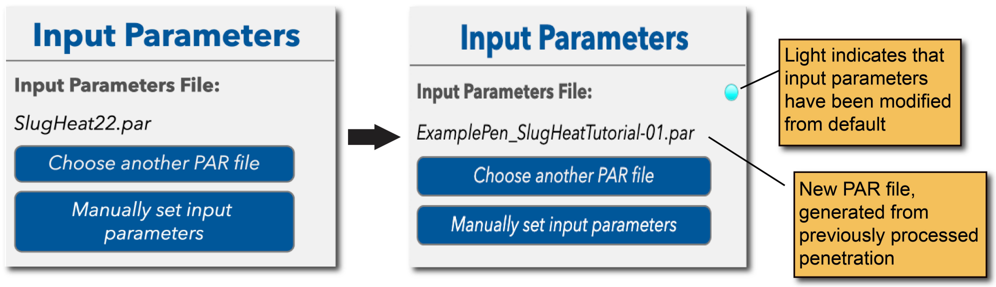


Figure 8: *SlugHeat* option for changing input parameters text file from default `SlugHeat.par` to a PAR file created while processing another penetration (in this example, new input parameters file is `ExamplePen_SlugHeatTutorial_01.par`). The blue light at the top right of second box indicates that parameters have been modified or a new input PAR file has been used.

6 Pre-processing data controls

Along with plots and displays, data are modifiable before processing in the following ways:

(a) **Timing of events:** Using the *Data Controls* box in the command window, timings of significant events, such as start of penetration, heat pulse firing, and end of penetration (shown with dashed vertical lines on the plots to right) can be modified by changing the corresponding record number in the text edit box to a new record number. If changed, plots will update accordingly. These record times and corresponding measurement values will be used in processing calculations. See Figure 6 **(box 3)**

(b) **Bottom water temperature:** All calculations using temperatures recorded by the heat-flow probe are done using temperatures relative to bottom water. When the data is loaded in, bottom water is initially assumed as the average of all temperatures recorded by the sensor that sits atop the data logger and does not penetrate the sediment (referred to as 'top sensor' in this tutorial) during penetration. To modify this value, **press *'Change BW temperature'* button** next to bottom water edit field in the command window under the *Commands* box (Figure 6 **(box 4)**). This will launch an auxiliary application where bottom water can be varied (Figure 9). A new bottom water temperature can be defined using:

- The temperature recorded by a specific sensor at a specific record number, or
- Manually inputting a temperature in the bottom water temperature text edit field at the bottom of this auxiliary application

When bottom water is changed, plots will update and all temperatures will be relative to this new bottom water temperature.

(c) **Sensors used:** Though checkboxes in the thermistors legend initially only control the *display* of data from each sensor, the data from individual sensors can be fully removed from the data set before processing. This is useful if there is a faulty sensor or the probe does not fully penetrate. To do this, **uncheck boxes for the sensors** which you would like to remove and **press *'Discard Sensors'* button** below the thermistors legend (Figure 5). When the data is processed, all measurements made by these sensors will be ignored. Checkboxes on the thermistors legend will be shaded and disabled for removed sensors.

(d) **Ignore the heat pulse decay reduction:** If there is no heat pulse fired, *SlugHeat* will automatically ignore the heat pulse decay reduction. However, if there is a heat pulse and you would like to *ignore* the heat pulse decay reduction anyway, you must **uncheck the *'Use heat pulse reduction'* checkbox** before processing (Figure 6 **(box 5)**. If the heat pulse decay reduction is ignored, *in-situ* thermal conductivity values will not be calculated. Instead, *SlugHeat* will use an assumed thermal conductivity vs. depth function to determine heat flow. This is defined in the default input parameters.

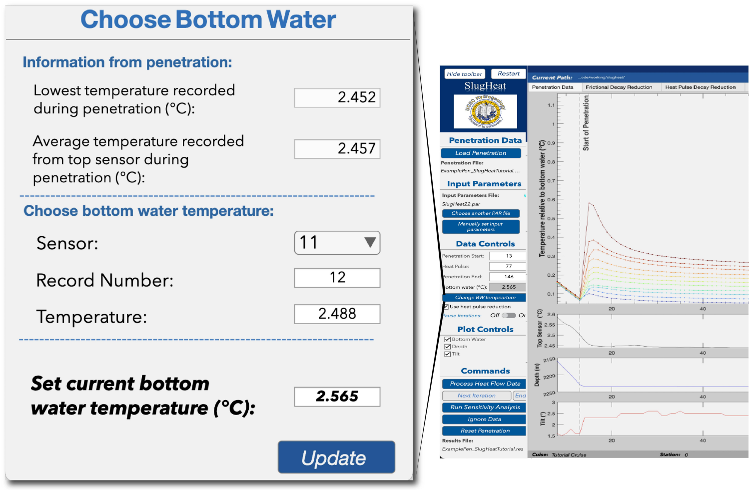


Figure 9: *SlugHeat* auxiliary application for changing bottom water temperature. This includes information from the penetration such as the lowest temperature recorded by the probe during penetration and the average temperature recorded from the top sensor (which is the initially assumed bottom water temperature).

7 Processing to calculate equilibrium temperatures with depth, *in-situ* thermal conductivity values with depth, and heat flow

When input parameters and data have been checked and modified as needed, the data can be processed to determine heat flow. Processing penetration data includes determining equilibrium temperatures of the sediment, *in-situ* thermal conductivity of the sediment, and heat flow. Equilibrium temperatures and thermal conductivity values for each sensor are determined iteratively. Once convergence criteria (specified as a default input parameter) has been reached, iterations end and values are established for each sensor (and therefore at the depths of each sensor in the sediment). These values are then used to calculate heat flow at the seafloor. Processing steps are described briefly below.

1. **Press *'Process Heat Flow Data'* button** in the *Commands* box of the command window (Figure 10 **(box 8)**).

    – **Option to pause between iterations:** *SlugHeat* will immediately begin to iterate between calculations for equilibrium temperatures of the sediment and *in-situ* thermal conductivity of the sediment at the depth of each sensor. Generally, only the final iteration is plotted unless otherwise noted by the user. To pause calculations and plot results from each iteration, **switch the *'Pause Iterations'* control switch on the command window to *'On'***. This will plot results from each iteration to allow mid-processing data inspection. To continue to next iteration, you must **press *'Next Iteration'* button** each time (Figure 10 **(box 9)**). This control switch can be turned on and off at any time. Iterations will continue until convergence criteria is met or a maximum number of iterations is reached. To calculate heat flow using equilibrium temperatures and thermal conductivity values from any iteration, even before convergence criteria has been reached, **press *'End'* button** instead of moving on to the next iteration. Values determined in the latest iteration when this button is pressed will be used to determine heat flow.

2. **Migrate to other tabs**, including *Frictional Decay Reduction* tab, *Heat Pulse Decay Reduction* tab, and *Heat Flow Analysis* tab. These allow investigation into results for each processing step, including:

    (a) ***Frictional Decay Reduction:*** This tab shows results from the frictional decay reduction, where the thermal response following the penetration is extrapolated to infinite time to calculate **equilibrium temperatures** of the sediment at the depth of each sensor. Penetration leads to an increase in temperature due to both an increase in temperature of the sediment relative

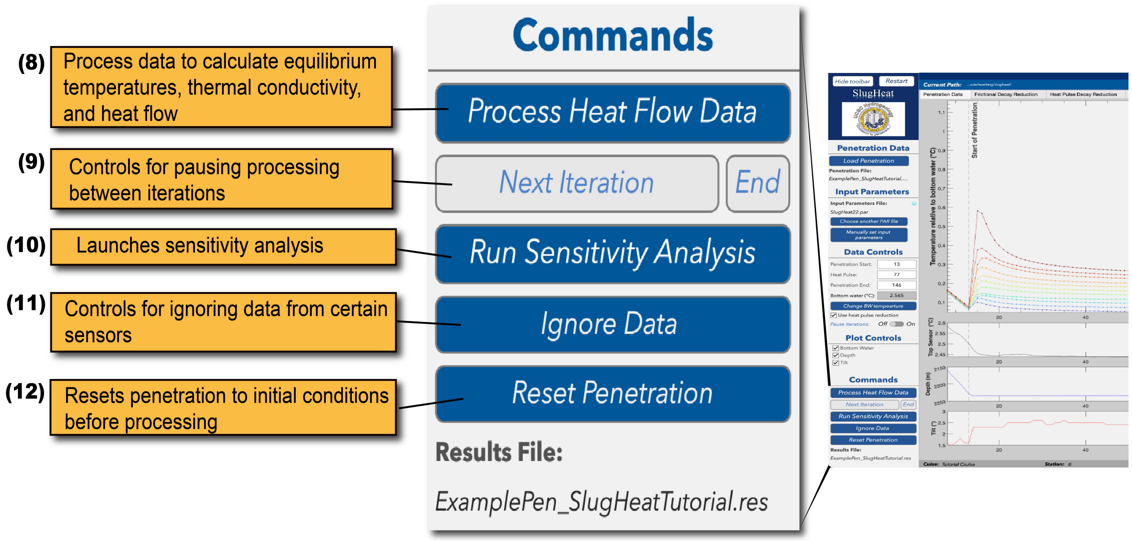


Figure 10: *SlugHeat* processing commands. Yellow boxes indicate notable processing data and display controls. Example penetration used: `ExamplePen_SlugHeatTutorial`

to bottom water as well as a frictional heating caused by the penetration itself. This frictional heating must be removed to determine equilibrium temperatures of the sediment. Each plot represents a step in these calculations. See Figure 11.

(b) ***Heat Pulse Decay Reduction:*** This tab shows results from the heat pulse decay reduction, where the thermal response following the firing of a calibrated heat pulse from the probe into the surrounding sediment is extradpolated to infinite time to calculate ***in-situ* thermal conductivity** of the sediment at the depth of each sensor. Each plot represents a step in these calculations. See Figure 12.

(c) ***Heat Flow Analysis:*** This tab shows results from the heat flow analysis, where the calculated equilibrium temperatures and *in-situ* thermal conductivity values with depth are used to calculate **heat flow**. Each plot represents a step in these calculations. See Figure 13.

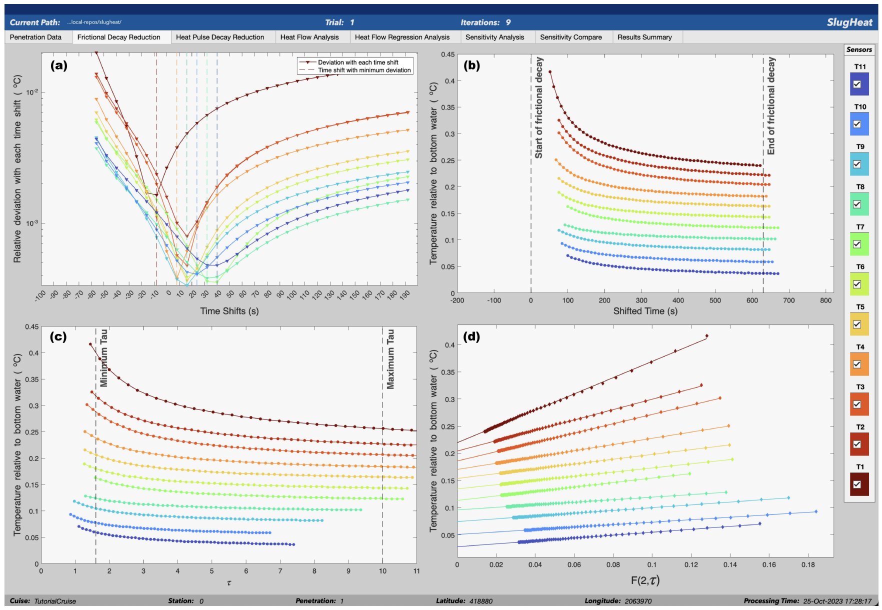


Figure 11: *SlugHeat* example *Frictional Decay Reduction* tab for determining equilibrium temperatures. Frictional heating caused by the penetration is removed by extrapolating temperatures to infinite time. Plots include **(a)** Temperature ($^{\circ}C$) vs. Time (s); **(b)** Temperature ($^{\circ}C$) vs. Dimensional time, τ; **(c)** Temperature ($^{\circ}C$) vs. the cylindrical decay function $F(\alpha, \tau)$; and **(d)** Residual misfit from the linear regression used in the previous plot ($^{\circ}C$) vs. Time Shifts (s). Vertical lines indicate the lowest residual misfit for each sensor.

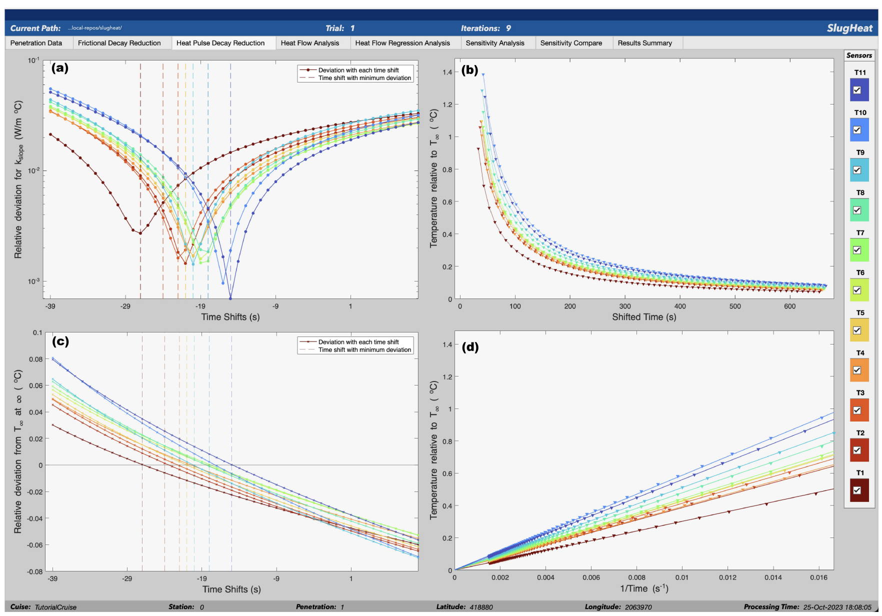


Figure 12: *SlugHeat* example *Heat Pulse Decay Reduction* tab for determining *in-situ* thermal conductivity. This is calculated using the thermal response following a calibrated heat pulse from the probe into the surrounding sediment. Corrected temperatures are temperatures above estimated equilibrium temperatures of the sediment. Plots include **(a)** Corrected temperature ($^{\circ}C$) vs. Time (s); **(b)** Residual misfit from the linear regression used in the previous plot vs. Time shifts (s) ; **(c)** Corrected temperature ($^{\circ}C$) vs. 1/Time (s); and **(d)** The temperature rise above equilibrium temperatures at infinite time (which should be zero) ($^{\circ}C$) vs. Time Shifts (s). Vertical lines in **(b)** and **(d)** indicate the lowest residual misfit for each sensor.

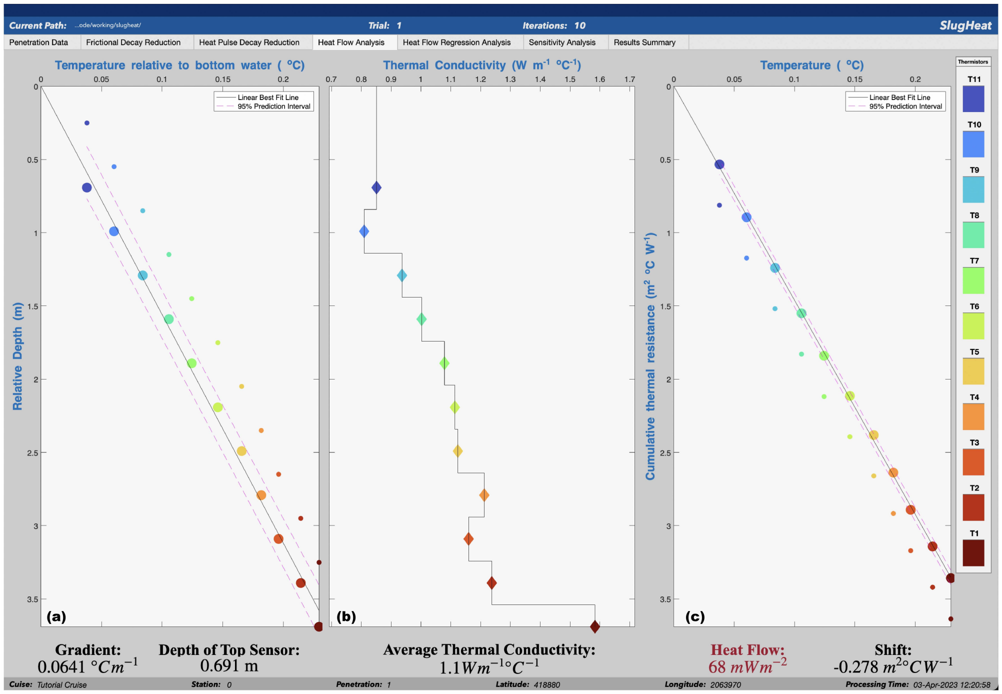


Figure 13: *SlugHeat* example *Heat Flow Analysis* tab for determining heat flow for a particular penetration. This is calculated using the equilibrium temperatures with depth, calculated in the *Frictional Decay Reduction* and the *in-situ* thermal conductivity values with depth, calculated in the *Heat Pulse Decay Reduction*. Plots include **(a)** Depth below seafloor (m) vs. Temperature relative to bottom water ($°C$); **(b)** Depth below seafloor (m) vs. *In-situ* thermal conductivity (W/m$°C$); and **(c)** Cumulative thermal resistance ($m^{2}°C/W$) vs. Temperature relative to bottom water ($°C$). For plots **(a)** and **(c)**, small markers indicate estimated results for each sensor, and large markers indicate results with a depth correction so that temperature relative to bottom water is zero at the seafloor. Solid black lines are linear best fit lines through depth-corrected results. Dashed pink lines indicate 95% confidence interval for the linear best fit line. Key calculations are shown below each plot. The slope of the linear best fit line in **(c)** is heat flow.

8 Post-processing data controls, data analysis, and reprocessing

SlugHeat provides several options for modifying and reprocessing data. This is useful to see effects of certain sensors, various input parameters, and more on heat flow results.

1. **Optionally modify input parameters and/or specify data to be ignored when reprocessed**:

   (a) To modify input parameters, see Section 5.

   (b) To remove results from specific sensors for heat flow determinations, **press the *'Ignore Data'* button** under the *Commands* box of the command window (Figure 10 box **(11)**). This will launch an auxiliary application (Figure 14) where data from individual sensors can be ignored. Equilibrium temperature and *in-situ* thermal conductivity values were calculated for each sensor in previous processing steps (*Frictional Decay Reduction* and *Heat Pulse Decay Reduction*, respectively). In this auxiliary application, either equilibrium temperatures, *in-situ* thermal conductivity values, or both can be ignored for any sensor. Values from unchecked sensors will be ignored in any subsequent processing.

2. **Reprocess**: After checking boxes for data to include and unchecking data to ignore, **press the *'Ignore unchecked data and reprocess for heat flow'* button** to update data to be used for subsequent processing. Heat flow will automatically be recalculated, ignoring all unchecked estimations.

   – Note: If input parameters were changed, all processing calculations will be redone. This includes all iterations between equilibrium temperature and *in-situ* thermal conductivity calculations for all sensors as well as heat flow determinations. If no input parameters were changed but data from certain sensors were ignored using the *Ignore Data* function, this does ***not*** recalculate equilibrium temperatures or *in-situ* thermal conductivity determinations. Instead, values for each sensor from the final iteration from the most recent processing will be used, but values calculated from ignored sensors are simply *unused* when determining heat flow.

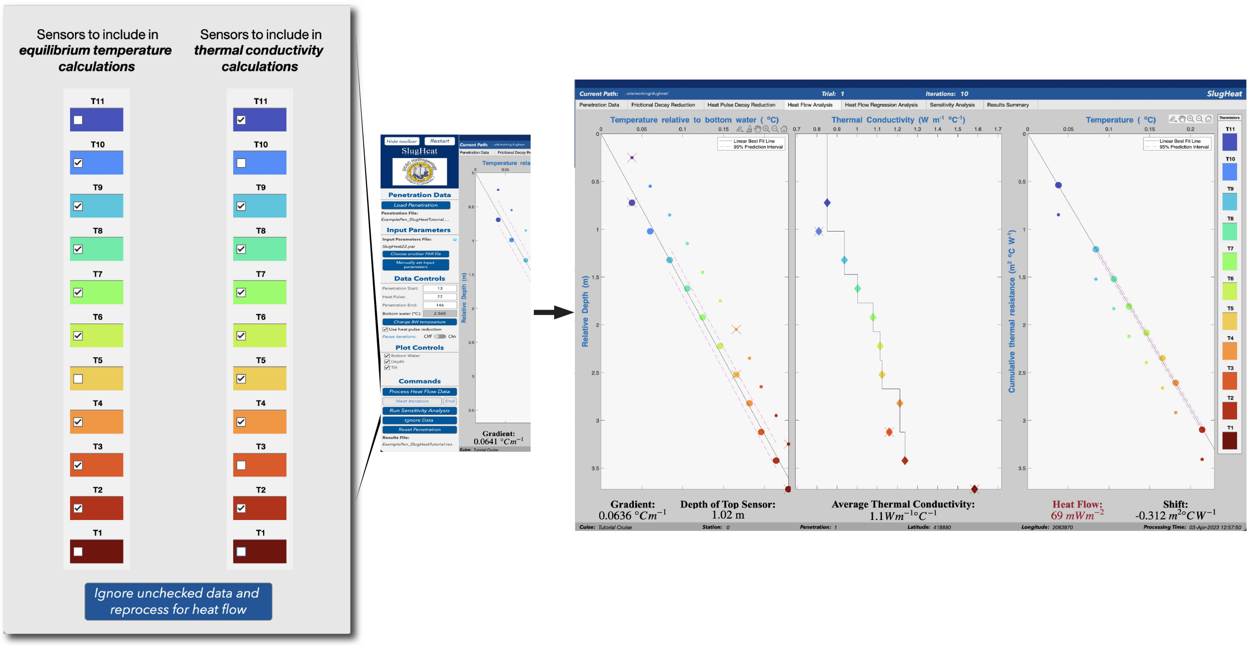


Figure 14: *SlugHeat* auxiliary application for ignoring data from certain sensors in heat flow determinations (left of arrow). This app is launched with the *Ignore Data* button under the *Commands* box in the command window. Plot to right of arrow shows results from a heat flow analysis where data from certain sensors are ignored. All data with an 'X' is not included in heat flow determinations for this analysis. In this example, equilibrium temperatures calculated for sensors T1, T5, and T11 are ignored and *in-situ* thermal conductivity values determined for sensors T1, T3, T10 are ignored. Use of application is detailed in text.

Ignored data will be shown with an 'X' through its corresponding sensor marker on the Depth vs. Temperature and Depth vs. Thermal conductivity plots (Figure 14). A sensor whose thermal conductivity values are ignored cannot be incorporated in the heat flow determination at all, however, a sensor whose equilibrium temperature alone is ignored but whose thermal conductivity value is used can still be incorporated in the heat flow assessments because it will affect the thermal gradient. Therefore, sensors whose equilibrium temperature alone is ignored will still be plotted on the Cumulative thermal resistance vs. Temperature relative to bottom water plots, but sensors whose thermal conductivity is ignored will not be plotted at all on this plot. See Figure 14 for example: T11 equilibrium temperature is ignored but this sensor is still plotted on **(c)**, whereas T10 thermal conductivity is ignored, so this sensor is not plotted on **(c)**.

Ignored data are not lost entirely. To reintroduce previously ignored data, simply **press the *'Ignore Data'* button** again to relaunch the corresponding auxiliary application, and recheck data from sensors you would like to include. This will update which equilibrium temperatures and *in-situ* thermal conductivity values are used in subsequent heat flow determinations, even those previously ignored.

9 Heat flow regression analysis using "scatter"

SlugHeat offers multiple ways to estimate sensitivity of results to various input parameters and calculations.

Firstly, to consider the number of sensors used in all analyses, as significant error could arise when measuring in disturbed sediment, **navigate to the *Heat Flow Regression Analysis* tab**. Uncertainty in the heat flow linear regression (Figure 13c) is assessed using a parameter called "scatter" (Villinger and Davis, 1987):

$$\xi = \sqrt{\frac{\chi^2}{N-1}} \tag{1}$$

where
ξ is "scatter" from the regression model,
χ^2 is based on the minimized misfit to the regression estimate, and
N is the number of sensors used

"Scatter" (ξ) increases with the misfit from the heat flow linear regression, therefore, can be used to assess uncertainty in heat flow. It is analyzed for each heat-flow measurement by systematically removing data from the shallowest subseafloor sensors, to see how this influences values of both ξ and heat flow. Results are presented with plots of ξ vs. N (number of sensors retained) and heat flow vs. N (Figure 15, top plot) as well as normalized ξ vs. N and heat flow uncertainty (σ_{HF}) vs. N (Figure 15, bottom plot).

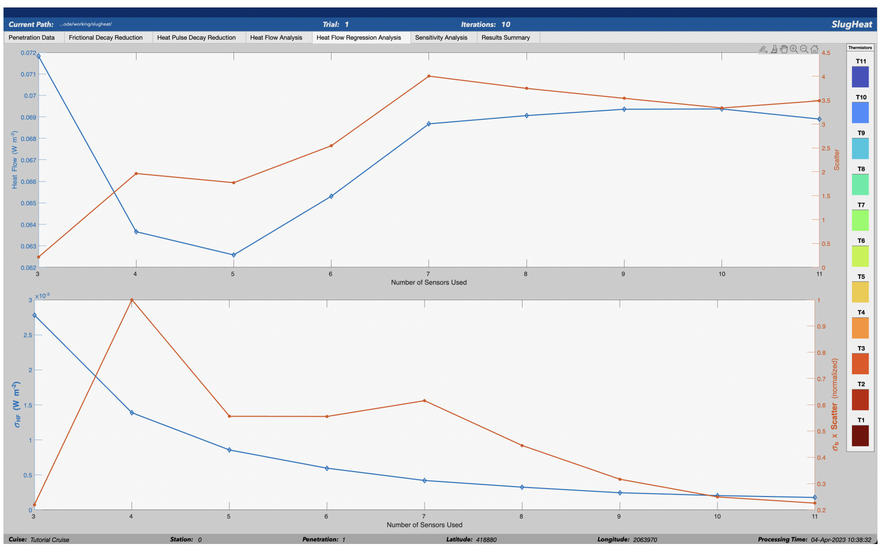


Figure 15: *SlugHeat* example *Heat Flow Regression Analysis* tab. Top plot shows number of sensors used vs. heat flow (blue line) and number of sensors used vs. scatter (ξ) (orange line). Bottom plot shows number of sensors used vs. heat flow uncertainty (σ_{HF}) (blue line) and number of sensors used vs. normalized scatter (orange line).

10 Heat flow Monte Carlo sensitivity analysis

SlugHeat also includes an optiona Monte Carlo sensitivity analysis to assess the impact of varying thermal conductivity and/or the thicknesses of sediment layers for which properties are inferred (Figure 13b). Once thermal conductivity values are determined for each sensor, thermal resistance is calculated based on the assumption that seafloor sediments comprise a series of horizontal layers, with layer boundaries located midway between sensor locations. The sensitivity analysis is based on randomly shifting the boundaries between layers, within limits, and varying the values of thermal conductivity assigned to each layer. With each shift in values, the program recalculates the heat flow. This processes is typically repeated $\geq$100 times (each time is a 'realization'), generating an ensemble of heat flow results for each probe penetration from which multiple metrics are calculated (mean, median, range, standard deviation).

To run the Monte Carlo analysis,

1. **Press the *'Run Sensitivity Analysis'* button** (Figure 10 box **(10)**): This will launch an auxiliary application for generating layer property distributions to be analyzed (Figure 16). Use this auxiliary application to specify and visualize property distributions before running the analysis.

2. **Define properties to be tested:** This includes the number of realizations to be run and layer thickness and thermal conductivity properties to vary each realization. Adjust property distributions using the text edit boxes in the auxiliary application. Properties to vary for *each sensor* include:

   (a) **Layer thickness:** Layer thicknesses are defined by a uniform distribution and is varied such that no layer boundary is closer than a prescribed distance from a sensor. User can define minimum thickness.

   (b) **Thermal conductivity:** Thermal conductivity values can be held constant (select 'None' in *Distribution of Thermal Conductivity Values* options of auxiliary application). If held constant, values calculated in the heat pulse decay reduction will be used (See Section 7) or an assumed function of thermal conductivity vs. depth will be used if no heat pulse reduction was done, and only layer thickness will change each realization. To vary thermal conductivity values with each realization, generate a distribution of values for each sensor to be tested. The default average of each distribution is either the *in-situ* measurement determined using the heat pulse or the value determined by the the assumed depth function, but the user may adjust this as needed for each sensor. Thermal conductivity values are varied to follow either a truncated Gaussian or Gamma distribution with lower and upper limits set based on *a priori* knowledge. For example, thermal conductivity of seafloor sediment can't be lower than that for seawater ($\sim 0.6\ W/m$°C) and is typically bounded at the upper limit based on regional data (perhaps 1.5 - 2.0 W/m°C for most shallow marine sediment types). Distribution and truncation parameters can be set equally for all sensors or set individually, if desired. In the auxiliary application (Figure 16), the text edit boxes in the *Variable thermal conductivity values and layer thickness values* section define these distributions. The two plots on the right-hand side represent a Gaussian distribution (top plot) or Gamma distribution (bottom plot) of thermal conductivity values to be tested and update as the user changes key parameters. The Gaussian distribution plot is highlighted in green because it is currently selected. In this example, sensor 5 is plotted. Varying any parameters on this app will vary those that define sensor 5. Sensor number can be changed with drop-down menu at bottom right-hand corner of the application. Table defines all parameters currently set for all sensors. Alternatively, all sensor parameters can be updated together using the *'Update All Sensors to Current Parameters'* button or the *'Reset All Sensors to Default'* button. If the *'Run Sensitivity Analysis'* button is pressed, the sensitivity analysis will be run using parameters currently defined in this auxiliary application.

3. **Press the *'Run Sensitivity Analysis'* button** once the number of realizations and properties to be varied each realization are defined. This will close the application and begin the sensitivity analysis.

An example sensitivity analysis run using properties defined in Figure 16 is shown in Figure 17. Results displayed include the heat flow standard deviation for each realization of properties (Figure 17 **(a)**), the distribution of possible heat flow values for each sensor (Figure 17 **(b)**), and a heat flow analysis plot for each realization (Figure 17 **(c)**). The ensemble of results from all realizations are then used to calculate a median, mean, and standard deviation, the latter of which is typically considerably larger than inferred from the statistical fit of the initial heat flow analysis plot.

11 Results and outputs

All results are saved to summary tables and output in multiple formats, including MAT files, text files, and even Excel sheets. All plots can be saved individually as needed using the plot controls in the upper right-hand corner of the plot.

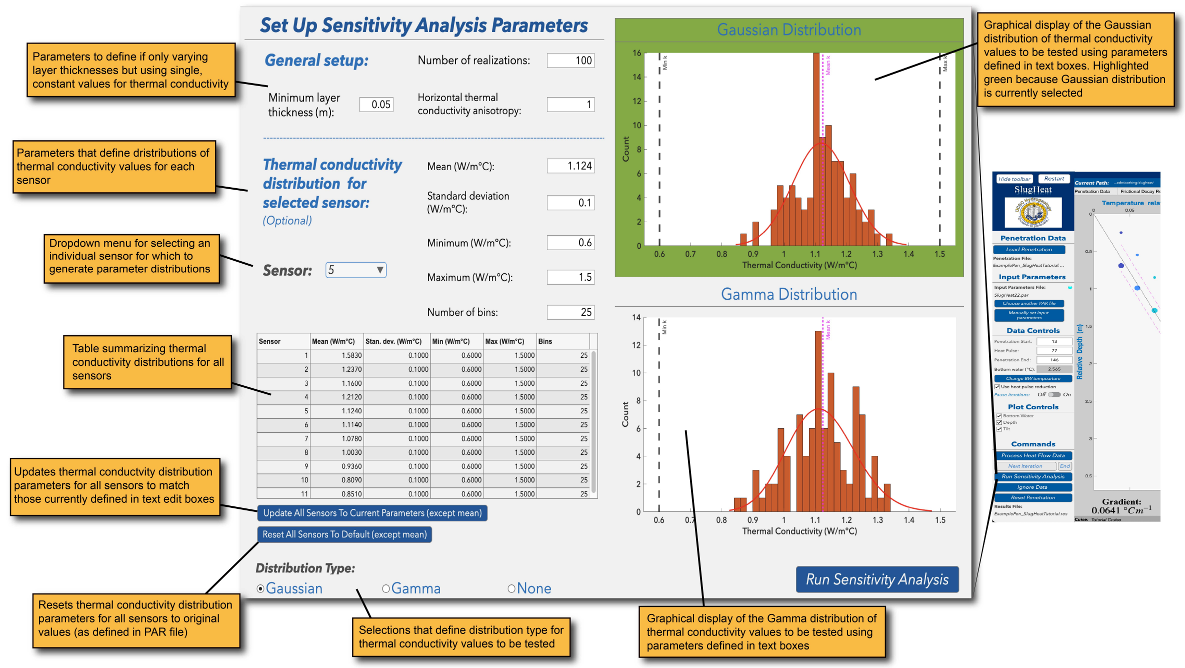


Figure 16: *SlugHeat* auxiliary application for defining parameters to be tested in a sensitivity analysis, including number of realizations to be run, minimum layer thickness, and thermal conductivity distributions. Thermal conductivity distributions for each sensor are displayed on right-hand side of this application, including plots that display distributions and a table that defines all parameters set for all sensors. Details in text. Yellow boxes indicate notable distribution controls and plots. Example penetration used: `ExamplePen_SlugHeatTutorial`

Summary Tables

Key results are summarized on the final tab of *SlugHeat*. To inspect and compare results from all analyses during a single program session (i.e., since the time *SlugHeat* was launched), **navigate to the *Results Summary* tab**. Here, there are two summary tables, which define notable input parameters and results not only from various penetrations but also for each trial of a single penetration. A penetration that is analyzed multiple times with varying input parameters or used/ignored data will have multiple trials. Comparing and contrasting results from different penetrations or different trials can help with heat flow interpretations. The two summary tables shown on this tab include:

(a) **Heat flow results summary table:** Here, results from all heat flow analyses since the start of the session are summarized. This includes key input parameters, such as penetration and trial number, use of a heat pulse, bottom water temperature, sensors used/ignored, and more. Also shown are key results from that penetration and trial, including thermal gradient, average thermal conductivity, and heat flow. (Figure 18)

(b) **Sensitivity analysis results summary table:** Here, results from all sensitivity analyses since the start of the session are summarized. This includes penetration and trial, number of realizations, layer thicknesses and thermal conductivity distribution types, minimum layer thickness, thermal conductivity value range and standard deviation, and more. These property distributions may vary for each sensor. Also summarized are key results of each sensitivity analysis, including initial, final, mean, total range, and standard deviation of heat flow. (Figure 19)

Each results summary table can be saved or cleared. To save the summary table as a MAT file and Excel file in your current directory, **press the *'Save'* button below the summary table**.

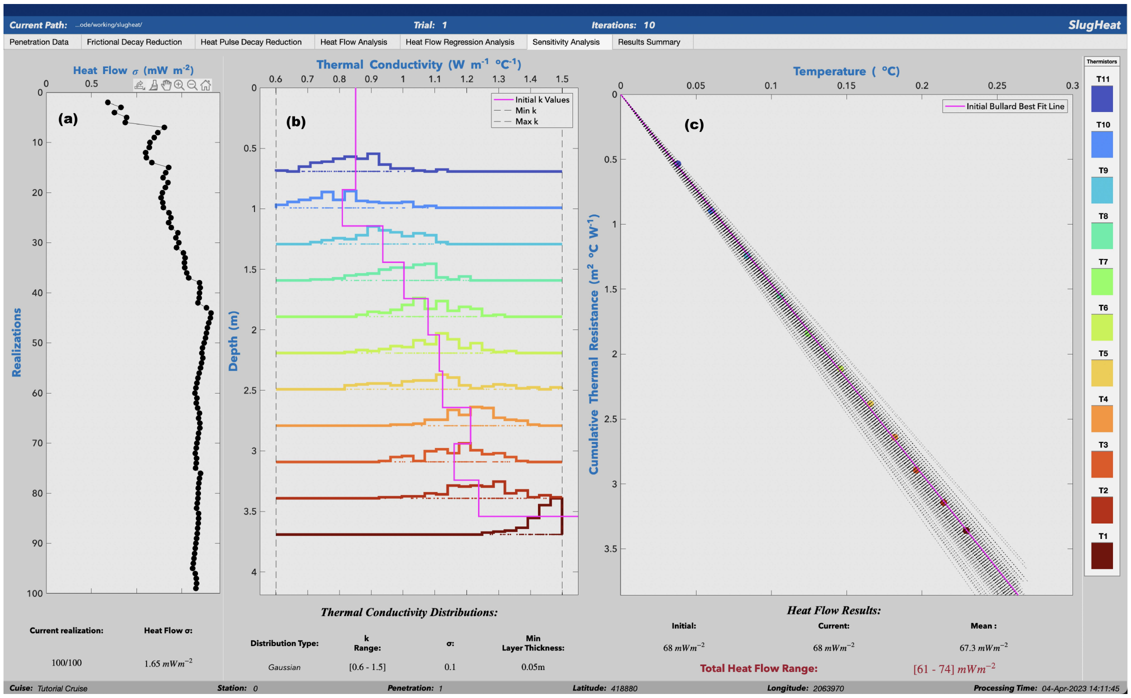
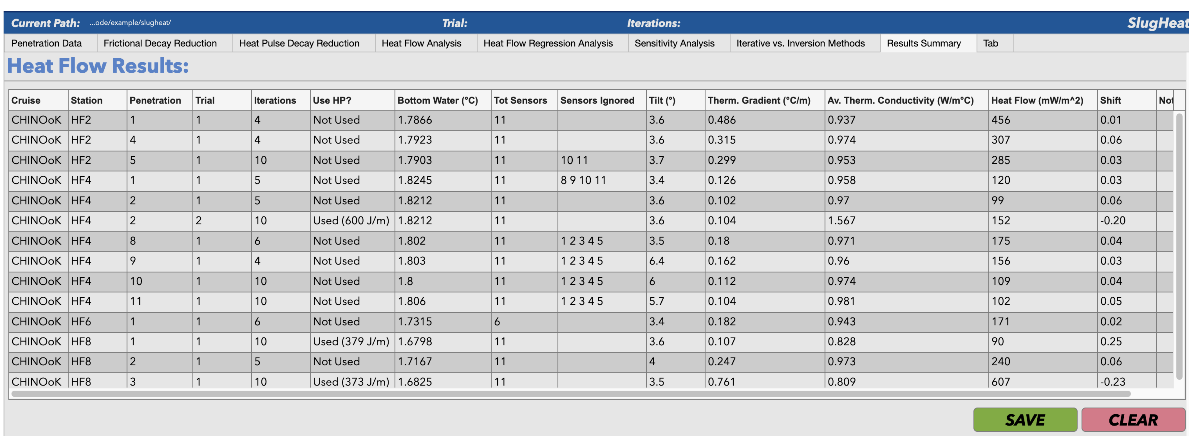


Figure 17: *SlugHeat* example *Sensitivity Analysis* tab with analysis results compiled for all realizations. **(a)** Heat flow uncertainty (σ_{HF}) vs. Realization; **(b)** Thermal conductivity ($W/m°$) vs. Depth below seafloor (m) where the initial thermal conductivity values (used in the original heat flow analysis during processing) are shown with the pink line. Each colored line depicts the distribution of thermal conductivity values for each sensor. Values for each realization are chosen randomly from these distributions. Dashed black lines indicate minimum and maximum values; and **(c)** Cumulative thermal resistance ($m^2/°W$) vs. Temperature relative to bottom water ($°C$) where the pink line represents results from the initial heat flow analysis. Each dashed line is a heat flow regression for a single realization. Key input parameters and results are defined below each plot.



Cruise	Station	Penetration	Trial	Iterations	Use HP?	Bottom Water (°C)	Tot Sensors	Sensors Ignored	Tilt (°)	Therm. Gradient (°C/m)	Av. Therm. Conductivity (W/m°C)	Heat Flow (mW/m^2)	Shift	Not
CHINOoK	HF2	1	1	4	Not Used	1.7866	11		3.6	0.486	0.937	456	0.01	
CHINOoK	HF2	4	1	4	Not Used	1.7923	11		3.6	0.315	0.974	307	0.06	
CHINOoK	HF2	5	1	10	Not Used	1.7903	11	10 11	3.7	0.299	0.953	285	0.03	
CHINOoK	HF4	1	1	5	Not Used	1.8245	11	8 9 10 11	3.4	0.126	0.958	120	0.03	
CHINOoK	HF4	2	1	5	Not Used	1.8212	11		3.6	0.102	0.97	99	0.06	
CHINOoK	HF4	2	2	10	Used (600 J/m)	1.8212	11		3.6	0.104	1.567	152	-0.20	
CHINOoK	HF4	8	1	6	Not Used	1.802	11	1 2 3 4 5	3.5	0.18	0.971	175	0.04	
CHINOoK	HF4	9	1	4	Not Used	1.803	11	1 2 3 4 5	6.4	0.162	0.96	156	0.03	
CHINOoK	HF4	10	1	10	Not Used	1.8	11	1 2 3 4 5	6	0.112	0.974	109	0.04	
CHINOoK	HF4	11	1	10	Not Used	1.806	11	1 2 3 4 5	5.7	0.104	0.981	102	0.05	
CHINOoK	HF6	1	1	6	Not Used	1.7315	6		3.4	0.182	0.943	171	0.02	
CHINOoK	HF8	1	1	10	Used (379 J/m)	1.6798	11		3.6	0.107	0.828	90	0.25	
CHINOoK	HF8	2	1	5	Not Used	1.7167	11		4	0.247	0.973	240	0.06	
CHINOoK	HF8	3	1	10	Used (373 J/m)	1.6825	11		3.5	0.761	0.809	607	-0.23	

Figure 18: *SlugHeat Results Summary* tab example of heat flow summary table.

Text File Outputs

The results summary tables on the final tab of the *SlugHeat* GUI are useful for highlighting key results of each analysis and quickly characterizing influential input parameters, penetration data, etc. However, there are numerous processing parameters and calculation results used in *SlugHeat* that are not shown in these summary tables. The entire catalog of processing parameters and results can be found in an output text file for each analysis. These files are automatically generated and saved to your current directory anytime heat flow data is processed. These are known as RES files. The RES file for the example penetration used in this

Error and Uncertainty Results:

Cruise	Station	Penetration	Trial	Iterations	Thickness Distribution	Min Thickness (m)	k Distribution	k Range (W/m°C)	k Standard Deviation	Initial q (mW/m^2)
CHINOoK	HF8	1	1	100	Uniform	0.05	Normal	[0.7 - 1.5] [0.7 - 1.5] [0.7 - 1.5] [0.7 - 1.5] [0.7…	0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1	90
CHINOoK	HF8	2	1	100	Uniform	0.05	Normal	[0.75 - 1.5] [0.75 - 1.5] [0.75 - 1.5] [0.75 - 1.5…	0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1	240
CHINOoK	HF8	3	1	100	Uniform	0.05	Normal	[0.75 - 1.5] [0.75 - 1.5] [0.75 - 1.5] [0.75 - 1.5…	0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1	607
CHINOoK	HF8	3	1	100	Uniform	0.05	Normal	[0.75 - 0.9] [0.75 - 0.9] [0.75 - 0.9] [0.75 - 0.9…	0.15 0.15 0.15 0.15 0.15 0.15 0.15 0.15 0.1…	607
CHINOoK	HF2	1	1	100	Uniform	0.5	Normal	[0.8 - 1.5] [0.8 - 1.5] [0.8 - 1.5] [0.8 - 1.5] [0.8…	0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1	456
CHINOoK	HF2	1	1	100	Uniform	0.5	Gamma	[0.7 - Inf] [0.8 - Inf] [0.8 - Inf] [0.8 - Inf] [0.8 - I…	0.2 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1	456
CHINOoK	HF4	2	1	100	Uniform	0.05	Normal	[0.7 - Inf] [0.7 - Inf] [0.7 - Inf] [0.7 - Inf] [0.7 - I…	0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2	99
CHINOoK	HF4	2	1	100	Uniform	0.05	Normal	[0.7 - Inf] [0.7 - Inf] [0.7 - Inf] [0.7 - Inf] [0.7 - I…	0.01 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2	99
CHINOoK	HF4	2	1	100	Uniform	0.05	Normal	[0.7 - Inf] [0.7 - Inf] [0.7 - Inf] [0.7 - Inf] [0.7 - I…	0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.0…	99
CHINOoK	HF4	2	1	100	Uniform	0.05	Gamma	[0.7 - Inf] [0.7 - Inf] [0.7 - Inf] [0.7 - Inf] [0.7 - I…	0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1	99
CHINOoK	HF8	4	1	100	Uniform	0.05	Gamma	[0.7 - Inf] [0.7 - Inf] [0.7 - Inf] [0.7 - Inf] [0.7 - I…	0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2	1563
CHINOoK	HF8	4	1	100	Uniform	0.05	Normal	[0.7 - 2] [0.7 - 2] [0.7 - 2] [0.7 - 2] [0.7 - 2] [0.…	0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.01 0.0…	1563

Figure 19: *SlugHeat Results Summary* tab example of sensitivity analysis summary table.

tutorial would be `ExamplePen_SlugHeatTutorial.res`. Each time a penetration is loaded in to *SlugHeat*, a new RES file is created for that penetration and all trials for that penetration are recorded in this single RES file.

In addition to a RES file, a new PAR file will be generated. This not only defines all the input parameters used in this analysis, but it can then be used as an input PAR file in subsequent analyses in lieu of the default `SlugHeat.par` text file, if desired. This can be done by using the *'Choose another PAR file'* button before processing and selecting a PAR file that was generated as an output from another penetration (see Section 5).

To find the RES file for an individual penetration, **look in your current directory** for a subfolder called **outputs/**. In case of difficulty in locating any output files, pop-up windows launched when an analysis is finished will define where all output files are saved.

12 Overview of code

SlugHeat was designed and built using MATLAB's App Designer, a modern interactive development environment. All nested functions and variables are housed in the main application (*SlugHeat.mlapp*). There are multiple auxiliary applications to aid in processing (for example, to collect and record metadata and processing parameters or to build parameter distributions for the sensitivity analysis). The GUI consists of numerous functions, called *callbacks* that are called when certain buttons are pressed by the user. Within each callback, there may be multiple MATLAB functions to be run. All major functions are separated into individual .m files. Here, the general code structure is described, including input and output files and tables describing all callbacks, functions, and variables used in *SlugHeat*.

GUI tabs:

- Penetration Data
- Frictional Equilibration Reduction
- Heat Pulse Decay Reduction
- Heat Flow Analysis
- Heat Flow Regression Analysis
- Sensitivity Analysis
- Sensitivity Analysis Comparison
- Results Summary

Files:

Inputs for each penetration

- .par text file - define input parameters
- .pen file and .tap text files OR .mat file - all data for a single peneration

Outputs for each penetration

- .res text file
- .mat file (2)
- .xlsx file (2)

All code files, callbacks, functions, and variables are described in tables in subsequent pages

Files and Folders

File	Type	Description
BottomWaterAuxApp.mlapp	app	auxiliary GUI for defining bottom water temperature
cbValueChange.m	function	shows or hides data from each sensor on GUI plots based on sensor's checkbox value
ChiSquaredFit.m	function	computes the linear regression and associated uncertainties
DiscardData.mlapp	app	auxiliary GUI for ignoring data from processing
DiscardSensorsNoHP.m	function	removes any unchecked sensors from the dataset when there is no heat pulse. removes Teq values only, as there are no calculated k values
FrictionalDecay.m	function	computes equilibrium temperatures for each sensor
GetFiles.m	function	gets input files (.pen, .tap, and/or .mat files) and creates output .res file
HeatFlowAnalysis.m	function	computes heat flow using the Bullard method
HeatFlowRegression.m	function	computes "scatter" of heat flow and how it varies% with decreasing the number of sensors
HeatPulseDecay.m	function	computes thermal conductivities for each sensor from the thermal response following the calibrated heat pulse
images	folder	houses images needed for the GUI
InitializeProcessing.m	function	initializes the processing of the raw penetration data
InitializeProgram.m	function	initializes SlugHeat
inputs	folder	houses input files
outputs	folder	houses output files. this folder is created if it does not already exist
PlotCheckboxes.m	function	plots all sensor checkbox panels for each tab, along with callbacks for each checkbox to turn sensor lines on and off
PlotFrictionalDecay.m	function	plots results from each step of the frictional equilibration reduction
PlotHeatFlowAnalysis.m	function	plots results from the heat flow analysis
PlotHeatFlowRegression.m	function	plots results from the heat flow regression analysis
PlotHeatPulseDecay.m	function	plots results of the heat pulse decay reduction
plotLayout.m	function	controls visibility of plots on the GUI based on the values of checkboxes for each
PlotPolynomial.m	function	plots a 2nd order polynomial regression for the thermal gradient and Bullard plot.
PlotTemp.m	function	plots the calibrated temperatures relative to bottom water temperature
PlotDepth.m	function	plots the depth data from the tap file
PlotTilt.m	function	plots the tilt data from the tap file
PrintBullardResults.m	function	prints results from heat flow analysis to .res file
PrintFricResults.m	function	prints results of frictional decay processing to .res file
PrintHeaderResults.m	function	prints header of .res file
PrintHeatPulseResults.m	function	prints results of frictional decay processing to .res file
PrintNewPar.m	function	prints all input parameters to a new .par file
PrintParametersResults.m	function	prints all parameters used to .res file
PrintSensResults.m	function	prints results of sensitivity analysis to .res file
PrintStatus.m	function	print out the current status of the program onto an output file

README.md	markdown	program read me file
ReadParFile.m	function	reads in the parameters from .par file
ReadPenFile_withPulse.m	function	reads all penetration data from the .mat file
ReadPenText_withPulse.m	function	reads all penetration data from the .pen text file
ReadTAPFile.m	function	reads all tilt and pressure data from the .mat file
ReadTAPText.m	function	reads all tilt and pressure data from the .tap text file
ResetAll.m	function	resets the axes and all plotting information to conditions prior to reading in a penetration file
SensitivityAnalysis.m	function	runs the heat flow sensitivity analysis
SetParams.mlapp	app	auxiliary GUI for setting input parameters
SetUpSensAnalysis.mlapp	app	auxiliary GUI for setting input parameters and generating parameter distributions for the sensitivity analysis
QuickStartGuide.pdf	pdf	quick start guide for using SlugHeat
SlugHeat.mlapp	app	main GUI for running SlugHeat
SlugHeat22.par	text	default parameters (.par) file automatically loaded in with the start of SlugHeat
SplitDecays.m	function	splits the penetration data into two sets: (a) thermal response following penetration and (b) thermal response following the calibrated heat pulse
TempCorrection.m	function	corrects the raw temperature data with sensor calibration and referencing to bottom water temperature
updateLabels.m	function	updates the layout of the SlugHeat GUI to current conditions
xAlign.m	function	aligns the X axes of each plot on the penetration data tab
xAlignHPD.m	function	aligns the X axes of each plot on the heat pulse decay tab

Callbacks

Callback	Description	Functions called
startupFcn	executes when the figure is created	InitializeProgram
		ReadParFile
		updateLabels
button_ControlWindowExitButtonPushed	executes when button to close controls panel is pushed	xAlign
		xAlignHPD
button_LoadPenPushed	executes when button to load penetration data is pushed	ResetAll
		GetFiles
		ReadPenText_withPulse
		ReadTAPText
		ReadPenFile_withPulse
		ReadTAPFile
		TempCorrection
		SplitDecays
		PlotTemp
		PlotDepth

		PlotTilt PlotCheckboxes plotLayout updateLabels xAlign PrintHeaderResults PrintStatus
checkbox_BottomWaterPlotValueChanged	executes when the bottom water plot checkbox is turned on or off.	plotLayout xAlign
checkbox_DepthPlotValueChanged	executes when the depth plot checkbox is turned on or off	plotLayout xAlign
checkbox_TiltPlotValueChanged	executes when the tilt plot checkbox is turned on or off	plotLayout xAlign
button_SetParamsPushed	executes when button for manually selecting parameters is pressed	SetParams**
button_ProcessButtonPushed	executes when button for manually selecting parameters is pressed	PrintStatus SplitDecays InitializeProcessing FrictionalDecay PlotFrictionalDecay HeatPulseDecay PlotHeatPulseDecay HeatFlowAnalysis PlotHeatFlowAnalysis HeatFlowRegression PlotHeatFlowRegression PrintNewPar PrintParametersResults PrintFricResults PrintHeatPulseResults PrintBullardResults
button_RestartPushed	executes when button for restarting entire program is pressed	ResetAll ReadParFile
edit_PenStartValueChanged	executes when penetration start time text edit field value is changed	
edit_HPValueChanged	executes when heat pulse start time text edit field value is changed	
edit_PenEndValueChanged	executes when penetration end time text edit field value is changed	
figure_MainSizeChanged	executes when the main figure size changes	xAlign xAlignHPD
button_IgnoreDataPushed		DiscardSensorsNoHP

	executes when the user chooses to discard data using the Ignore Data button	DiscardData**
checkbox_TempvDepthValueChanged	executes when user wants to turn off or on Temp vs. Depth plot	
checkbox_TCvDepthValueChanged	executes when user wants to turn off or on Thermal Conductivity vs. Depth plot	
checkbox_TempvCTRValueChanged	executes when user wants to turn off or on Bullard plot	
button_NextIterationPushed	executes when next iteration button is pressed	
button_EndIterationsPushed	executes when end iterations button is pressed	
button_saveresultsPushed	executes when save results button is pressed	
button_clearresultsPushed	executes when clear results button is pressed	
buttongroup_PlotControls_FricSelectionChanged2	executes the user moves to the frictional decay tab on the GUI	plotLayout
checkbox_HFPlotValueChanged	executes the user wants to turn on or off the heat flow "scatter" plot	plotLayout
checkbox_SigmaPlotValueChanged	executes the user wants to turn on or off the normalized heat flow "scatter" plot	plotLayout
button_OpenControlsPanelButtonPushed	executes when the button for bringing back the user controls window is pressed	xAlign xAlignHPD
switch_PauseBetweenIterValueChanged	executes when switch button for pausing between iterations is switched	
tabgroup_MainSelectionChanged	executes when the user moves to another tab in the GUI	
button_resetPushed	executes when the reset penetration button is pressed	PlotTemp
		PlotDepth
		PlotTilt
		PlotCheckboxes
checkbox_UseHPValueChanged	executes when the user manually defines whether to use a heat pulse or ignore it with the Use HP checkbox	
button_SensAnButtonPushed	executes when the Run Sensitivity Analysis button is pressed	SetUpSensAnalysis**
		SensitivityAnalysis
		PrintSensResults
button_saveErrRestultsButtonPushed	executes when the save error and ucnertaitny table button is pressed	
button_clearErrResultsButtonPushed	executes when the clear error and ucnertaitny table button is pressed	
button_NewParFilePushed	executes when the new parameters (.par) file button is pressed	ReadParFile
button_changeBWValueChanged	executes when the cahnge bottom water temperature button is pressed	BottomWaterAuxApp** TempCorrection

		SplitDecays
		PlotTemp
		PlotDepth
		PlotTilt
		PlotCheckboxes
		plotLayout
		updateLabels
		xAlign
edit_bottomwaterValueChanged	executes when the bottom water temperature value is changed	TempCorrection
		SplitDecays
		PlotTemp
		PlotDepth
		PlotTilt
		PlotCheckboxes
		plotLayout
		updateLabels
		xAlign
button_DiscardSensorsPushed	executes when the discard sensors button below the sensor legend on the penetration tab is pressed	PrintStatus
		TempCorrection
buttongroup_PlotControls_HPSelectionChanged	executes when theselected plots to show or hide heat pulse decay reduction tab is changed	plotLayout
checkbox_TiltPlot_2ValueChanged	executes when tilt plot checkbox value is changed	

Variables

Each variable is global and is recognized by the main application, that is it can be called in any callback. Each variable can be called by dot indexing into the app.

*Function that creates the variable

Variable name	Function*	Description	Units
Version	InitializeProgram.m	version of SlugHeat	
Update		update of SlugHeat	
NumberOfColumns		number of columns for .res file	
CurrentPath		current path	
CurrentDateTime		current data and time	
ParFile		full .par file name and path	
ParFilePath		.par file path	
ParFileName		.par file name	

DefaultParFile		default .par file name (*SlugHeat.par*)	
ProgramLogId		.log file ID	
AppPath		path of app if using a compiled app	
AppOutputs		path of directory for storing outputs if using a compiled app	
PenFileName	GetFiles	.pen file name	
PenFilePath		.pen file path	
PenFile		full .pen file name and path	
TAPName		.tap file name	
TAPFileName		.tap file path	
TAPFile		full .tap file name and path	
MATFileName		.mat file name	
MATFile		full .mat file name and path	
LogFileName		.log file name	
LogFile		full .log file name and path	
ResFileName		.res file name	
ResFile		full .res file name and path	
ResFileId		.res file ID	
LogFileId		.log file ID	
S_ParFile	ReadParFile	structure holing parameters from .par file	
NumberOfSensors		total number of sensors excluding bottom water sensor, as defined in .par file	
WaterThermistor		water temperature thermistor? 1:Y 0:N, as defined in .par file	
TimeScalingFactor		seconds per record number, as defined in .par file	s/unit
DeltaTime		time between thermistor readings, as defined in .par file	s
SensorRadius		radius of the sensor, as defined in .par file	m
SensorDistance		distance between sensors, as defined in .par file	m
TempError		assumed temperature error, as defined in .par file	°C
CalibrationCoeffs		calibration coefficients, as defined in .par file	
HyndmanCoeffs		Hyndmann coefficients, as defined in .par file	
FrictionalDelays		frictional time delays, as defined in .par file	s
FricMaxStep		max frictional time step, as defined in .par file	s
TimeInc		time step increment for frictional reduction, as defined in .par file	s
FricTauMin		min τ for frictional reduction, as defined in .par file	

FricTauMax		max τ for frictional reduction, as defined in .par file	
PulseDelays		heat pulse time delays, as defined in .par file	s
kInit		initial thermal conductivities (k) function coefficients, as defined in .par file	
ktype		the first argument of kInit in the .par file. if = 99, use kInit(z), as defined in .par file	
PulsePowerPARFile		pulse power per length, as defined in .par file	J/m
TimeShiftInit		initial time shift, as defined in .par file	s
TimeShiftInc		time step increment for heat pulse reduction, as defined in .par file	s
PulseMaxStep		max heat pulse time step, as defined in .par file	s
kTolerance		k tolerance for defining for convergence, as defined in .par file	W/m°C
PulseTauMin		min τ for heat pulse reduction, as defined in .par file	
PulseTauMax		max τ for heat pulse reduction, as defined in .par file	
HeatPulseLength		duration of heat pulse, as defined in .par file	s
MinTotalkChange		convergence criteria - minimum change of Sigma(k), as defined in .par file	W/m°C
MaxNumberOfIterations		max number of iterations, as defined in .par file	
MaxSAIterations		max number of realizations for sensitivity analysis, as defined in .par file	
Sigmak0		standard deviation in k for sensitivity analysis, as defined in .par file	W/m°C
kMin		min k for sensitivity analysis, as defined in .par file	W/m°C
kMax		max k for sensitivity analysis, as defined in .par file	W/m°C
MinThickness		minimum layer thickness for sensitivity analysis, as defined in .par file	W/m°C
kAnisotropy		k bias in the horizontal direction for the sensitivity analysis, as defined in .par file	
TopSensorDepth		distance between top sensor and bottom of weight stand, as defined in .par file	m
ProbeLength		length of probe, as defined in .par file	m
Offset		manual temperature offset of sensor calibration, as defined in .par file	°C
S_MATFile	ReadPenFile	structure holding data loaded in from .mat file, as defined by .pen or .mat file	
FullExpeditionName		expedition name, as defined by .pen or .mat file	

StationName		station name, as defined by .pen or .mat file	
Penetration		penetration number, as defined by .pen or .mat file	
CruiseName		cruise name, as defined by .pen or .mat file	
Datum		datum, as defined by .pen or .mat file	
Latitude		latitude, as defined by .pen or .mat file	
Longitude		longitude, as defined by .pen or .mat file	
DepthMean		mean depth recorded during penetration, as defined by .pen or .mat file	mbsf
TiltMean		mean tilt recorded during penetration, as defined by .pen or .mat file	°
LoggerId		logger ID, as defined by .pen or .mat file	
ProbeId		probe ID, as defined by .pen or .mat file	
PenetrationRecord		record number for start of penetration, as defined by .pen or .mat file	
HeatPulseRecord		record number for firing of heat pulse, as defined by .pen or .mat file	
EndRecord		record number for end of penetration, as defined by .pen or .mat file	
AllRecords		vector of all record numbers during penetration, as defined by .pen or .mat file	
AllSensorsRawData		all temperature data (excluding bottom water sensor) before sensor calibration, as defined by .pen or .mat file	°C
WaterSensorRawData		bottom water sensor temperature data before sensor calibration, as defined by .pen or .mat file	°C
EqmStartRecord		record number for start of calibration period, as defined by .pen or .mat file	
EqmEndRecord		record number for end of calibration period, as defined by .pen or .mat file	
PulsePower		heat pulse power, as defined by .pen or .mat file	J/m
TAPRecord	ReadTAPFile	all tilt and pressure data record numbers, as defined by .tap file	
Tilt		all tilt data, as defined by .tap file	°
Depth		all depth data, as defined by .tap file	mbsf
BottomWaterTemp	TemperatureCorrection	bottom water temperature	°C
WaterSensorTemp		all bottom water sensor temperature data after sensor calibration	°C
AllSensorsTemp		all temperature data relative to bottom water, excluding bottom water sensor, before sensor calibration	°C
AllSensorsCalibratedTemp		all temperature data relative to bottom water, excluding bottom water sensor, after sensor calibration	°C
FricTime	SplitDecays	times of frictional equilibration	s
FricTemp		temps during frictional equilibration	°C

PulseData		indicator of whether a heat pulse was fired of not (0 = no, 1 = yes)	
PulseTime		times of heat pulse decay	s
PulseTemp		temps during heat pulse decay	°C
S_Lines	PlotRawTemp	structure of all temperature vs. time lines	
AllSensors		all sensors with data plotted	
h_axTempAboveBWT		axis for temperature relative to bottom water plot	
PenCheckboxes	PlotCheckboxes	checkboxes on penetration tab	
FricCheckboxes		checkboxes on frictional equilibration tab	
HPCheckboxes		checkboxes on heat pulse decay tab	
kFunction	InitializeProcessing	assumed thermal conductivity function	
Currentk		current k values for each sensor	W/m°C
CurrentT		current Teq values for each sensor	°C
TChange		change of Teq between each sensor from previous iteration	°C
kChange		change of Teq between each sensor from previous iteration	W/m°C
Iteration		current iteration number	
kInitial		initial thermal conductivity function	
FirstIteration		first iteration number	
TotalIterations		total number of iterations	
AnotherTrial		indicator for a new trial to be run	
RelativeDepths		relative depths of all sensors	mbsf
A		coefficient used for defining k with assumed function	
B		coefficient used for defining k with assumed function	
C		coefficient used for defining k with assumed function	
D		coefficient used for defining k with assumed function	
E		coefficient used for defining k with assumed function	
F		coefficient used for defining k with assumed function	
NumberOfFricUsedPoints	FrictionalDecay	number of data points used from frictional equilibration reduction	
MinimumFricEqTemp		Teq with minimum misfit from linear regression	°C
MinimumFricError		Teq error with minimum misfit from linear regression	°C
MinimumFricDelays		Teq time shift with minimum misfit from linear regression	s
MinimumFricSlope		Teq vs. depth slope with minimum misfit from linear regression	°C/m
HPTooLow		indicator for heat pulse power too low	1= yes, 0 = no

ShiftedTime		shifted time	s
IndexOfMinimums		index for Teq data with minimum misfit from linear regression	
DataTemp		corrected temperature	°C
TimeShifts		time shifts	s
ShiftedTau		shifted τ	
DataFAT		data for cylindrical decay function F(α, τ)	
DataLimits		temperature data limits	
b		temporary variable used for defining linear regression	
a		temporary variable used for defining linear regression	
RMS		residual misfit from linear regression for frictional equilibration reduction	°C
h_axFricTempvTime	PlotFrictionalDecay	axis for frictional equilibration vs time plot	
h_axFricTempvTau		axis for frictional equilibration vs t τ plot	
h_axFricTempvTauPoints		data points for frictional equilibration vs τ plot	
h_axFricTempvTauLines		lines for frictional equilibration vs τ plot	
h_axFricRMSvTimeShift		axis for RMS of frictional equilibration vs time shift plot	
h_axFricRMSvTimeShiftMinDelays		lines for minimums on RMS of frictional equilibration vs time shift plot	
MeankPointAtMinkDiff	HeatPulseDecay	mean k(point) with minimum deviation from k(slope)	W/m°C
kSlopeAtMinkDiff		k(slope) with minimum misfit from linear regression	W/m°C
MeankPointAtZeroInfTemp		mean k(point) with minimum misfit from linear regression based on Teq = T at infinite time	W/m°C
MeankPointAtMinRMS		mean k(point) with minimum misfit from linear regression based on RMS	W/m°C
kSlopeAtZeroInfTemp		mean k(slope) with minimum misfit from linear regression based on Teq = T at infinite time	W/m°C
kSlopeAtMinRMS		mean k(slope) with minimum misfit from linear regression based on RMS	W/m°C
TempAtInf		T relative to Teq at infinity when k(slope) and k(point) difference is minimized	°C
NumberOfUsedPoints		Number of data points used from heat pulse decay reduction	
MinimumPulseDelays		k time shift with minimum misfit from linear regression	s
kError		k error	
HeatPulseTime		time during penetration of heat pulse decay	s
MinkDiffIndex		index for k data with minimum misfit from linear regression	

kSlopeRMS		residual misfit from linear regression for heat pulse decay reduction	W/m°C
TempAtInfinity		T relative to Teq at infinity (all)	°C
OneOverTime		1 / time	1 / s
kSlope		k as determined by k(slope)	W/m°C
h_axHPTempvTime		axis for heat pulse decay vs time plot	
h_axHPRMS		axis for heat pulse decay residual misfits vs time shift plot	
h_axHPRMSLine		lines for heat pulse decay T relative to Teq vs time shift plot	
h_axHPTempvInvTime	PlotHeatPulseDecay	axis for heat pulse decay vs 1 / time plot	
h_axHPTempvInvTimeBestFit		best fit lines for heat pulse decay vs 1 / time plot	
h_axHPTempvTimeShift		axis for heat pulse decay residual misfits vs time shift plot	
h_axHPTempvTimeShiftBestFit		best fit lines for heat pulse decay vs 1 / time plot	
SensorsUsedForBullardFit		sensors with CTR to use for heat flow (Bullard) reduction	
GoodkIndex		index for k to use in heat flow reduction	
CTRToUse		sensors with CTR to use for heat flow (Bullard) reduction	
CTR		cumulative thermal resistance for all sensors	m2/W°C
ShiftedCTR		shifted cumulative thermal resistance for all sensors so that Teq = 0 at seafloor	m2/W°C
ShiftedRelativeDepths		shifted deptsh so that Teq = 0 at seafloor	mbsf
TToUse		sensors with Teq used for heat flow (Bullard) reduction	
kToUse		sensors with k used for heat flow (Bullard) reduction	
Slope		slope of thermal gradient linear best fit	°C/m
Shift	HeatFlowAnalysis	depth shift for all sensors	m
S_BullPlots		structure for holding children of Bullard plot	
HeatFlow		final heat flow value	mW/m2
Averagek		average k value for all sensors	W/m°C
Gradient		thermal gradient	°C/m
HFErr		heat flow error based on linear best fit	mW/m2
HFShift		heat flow shift	mW/m2
HFShiftErr		heat flow shift error	mW/m2
kErr		k error	W/m°C
GradErr		thermal gradient error	°C/m
GradShift		thermal gradient shift	°C/m
GradShiftErr		thermal gradient shift error	°C/m
Sigmaa	HeatFlowRegression	probable uncertainty in the estimates of heat flow	mW/m2

Sigmab		probable uncertainty in the estimates of the depth shift	m
Scatter		scatter	mW/m2
ScatterHeatFlow		heat flow from scatter analysis	mW/m2
HFLine	PlotHeatFlowRegression	heat flow plot in scatter analysis	
ScatterLine		scatter plot in scatter analysis	
SigmaHFLine		standard deviation in heat flow plot in scatter analysis	
SigmaScatterLine		standard deviation in scatter plot in scatter analysis	
Results	SensitivityAnalysis	structure holding all results from sensitivity analysis	
Params		structure holding all parameters from sensitivity analysis	
NewParFileName	New PAR file callback	name of new .par file created with current parameters	
NewParFile		full name and path of of new .par file created with current parameters	
BWPlot	SlugHeat main app	indicator for showing bottom water temperature plot	1= yes, 0 = no
DepthPlot		indicator for showing depth plot	1= yes, 0 = no
TiltPlot		indicator for showing tilt plot	1= yes, 0 = no
TempPlot		indicator for showing sediment temperature plot	1= yes, 0 = no
TempTimePlot		indicator for showing temperature vs. time plot	1= yes, 0 = no
TempTauPlot		indicator for showing temperature vs. τ plot	1= yes, 0 = no
TempBullFuncPlot		indicator for showing temperature vs. $F(\alpha, \tau)$ plot	1= yes, 0 = no
MisfitShiftPlot		indicator for showing residual misfit vs. time shift plot	1= yes, 0 = no
BullTempvDepthPlot		indicator for showing thermal gradient plot	1= yes, 0 = no
TCvDepthPlot		indicator for showing k vs. depth plot	1= yes, 0 = no
TempvCTRPlot		indicator for showing Bullard plot	1= yes, 0 = no
HFPlot		indicator for showing top scatter plot	1= yes, 0 = no
SigmaPlot		indicator for showing bottom scatter plot	1= yes, 0 = no
UseWaterSensor		indicator for using bottom water sensor	1= yes, 0 = no
UseHP		indicator for using a heat pulse	1= yes, 0 = no
NoInitialize		indicator for initializing program	1= yes, 0 = no
Reprocess		indicator for reprocessing in a new trial	1= yes, 0 = no
RunAnalysis		indicator for if the analysis has been run for this trial yet	1= yes, 0 = no
Badk		sensors to ignore k values	
BadT		sensors to ignore Teq values	
PenStartChanged		indicator for penetration start time changed in SlugHeat	1= yes, 0 = no

HeatPulseChanged		indicator for heat pulse time changed in SlugHeat	1= yes, 0 = no
PenEndChanged		indicator for penetration end time changed in SlugHeat	1= yes, 0 = no
Trial		trial number	
Pause		indicator for whether to pause between iterations	1= yes, 0 = no
Converged		indicator for if convergence has been reached	1= yes, 0 = no
TempTimePlotHPPlot		indicator for showing temp vs. time following heat pulse plot	1= yes, 0 = no
MisfitShiftHPPlot		indicator for showing misfit vs time shift following heat pulse plot	1= yes, 0 = no
TempInvTimeHPPlot		indicator for showing temp vs. 1/time following heat pulse plot	1= yes, 0 = no
TempTimeShiftHPPlot		indicator for showing temp vs. time shift following heat pulse plot	1= yes, 0 = no
SetParamsDialogApp		dialog app for setting input parameters	
DiscardDataDialogApp		dialog app for ignoring data	
SetUpSensDialogApp		dialog app for setting up parameters for sensitivity analysis	
BottomWaterApp		dialog app for changing bottom water temperature	
PreviousT		Teq in previous iteration	°C
Previousk		k in previous iteration	W/m°C
LightGrey		color for plot backgrounds	
OrigNumberOfSensors		original number of sensors, before any data is ignored	
IgnoredSensors		ignored sensors	
SensorsToUse		sensors to be used in all processing	
HFData		heat flow results summary data	
ErrData		error and uncertatinty results summary data	
S_FricResults		structure for all frictional decay reduction results	
S_HPResults		structure for all heat pulse decay reduction results	
S_HeatFlowResults		structure for all heat flow reduction results	
S_Results		structure for all results	
S_SensResults		structure for all sensitivity analysis results	
OrigParams		original parameters	
BWChosen		indicator bottom water temperature has been chosen	1= yes, 0 = no
BottomWaterValue		value of bottom water temperature if it's manually chosen by user	°C
NumSensAnalyses		number of sensitivity analyses that have been run for this penetration	

TotkChange		sum of differences in k each iteration for all sensors	W/m°C
kDistribution		distribution of k type for all sensors for sensitivity analysis	1 = Gaussian, 2 = Gamma, 3 = single value
r		distribution of k values for each sensor for sensitivity analysis	W/m°C
Bins		bin size for k distributions for sensitivity analysis	
UseFrictional		indicator for recalculating Teq with new k in sensitivity analysis	1= yes, 0 = no
Meank	SlugHeat main app	mean k for each sensor	W/m°C