HFLUX Stream Temperature Solver
[HFLUX]

1. Description

HFLUX is a one-dimensional transient model that calculates stream temperatures with respect to space and time using the mass and energy balance equations for temperature transport in streams. It uses initial spatial and temporal temperature boundary conditions, stream dimension information, discharge data, and meteorological data to calculate stream temperature using a finite difference method.

HFLUX interpolates the input data to the same spatial and temporal scales; this divides space and time into a grid of nodes, and the change from node to node can be computed. The program has two options for solving the governing equations in the model: a Crank-Nicolson method and a second-order Runge-Kutta method. The user may designate solution method depending on their needs. For each time step, HFLUX calculates the heat fluxes from the environment and lateral inflows of water to a node, computes temperature at that node, and then continues to the downstream node and calculates the next temperature. It then uses this information steps to calculate temperature at the next time step.

HFLUX is distributed as a set of MATLAB functions and can be downloaded from http://hydrology.syr.edu/hflux. hflux.m is the main function used in running this model. It calls to hflux_flux.m to calculate heat fluxes, and then applies these heat fluxes to each stream node to adjust the stream temperature.

2. Input data

HFLUX requires initial conditions for temperature in space and time, information about the stream dimensions, the meteorological conditions, streamflow rates, groundwater temperature, and streambed substrate and temperature. These data do not need to be collected at the same spatial or temporal scales; HFLUX uses linear interpolation to refine the resolution of the data in both space and time. The data requirements are:

Table 1. Input data

HFLUX Input Data	HFLUX Variable Name	Description of Data
settings	Numerical values indicating user selection of finite difference solution method, and options for determining shortwave radiation, latent heat, and sensible heat fluxes	
time_mod	Times at which temperatures are computed by the model (min)	
dist_mod	Distances along the reach at which temperatures are computed by the model (m)	
temp_x0_data	time_temp	Times at which temperature measurements are known (min)
	temp_x0	Temperature at the upstream boundary (°C)

temp_t0_data	dist	Distances along the reach where longitudinal temperatures were measured (m)
	temp_t0	Initial temperatures along the stream reach (set to be constant along the reach as the temperature at the upstream boundary at the first time step) (°C)
dim_data	dist_stdim	Distances along the reach where stream dimensions were measured (m)
	area	Cross-sectional area of the stream channel (m^2)
	width	Stream width (m)
	depth	Stream depth (m)
	discharge_stdim	Stream discharge rate at which stream dimensions were measured (m^3/s)
dis_data	dist_dis	Distances along the reach where discharge measurements were made (m)
	discharge	Discharge rates (m^3/s) at times given in time_dis; all locations must have discharge values for all times
time_dis	time_dis	Times at which discharge rates are specified in dis_data (min)
T_L	dist_T_L	Distances along the reach where temperature measurements were made (m)
	T_L	Groundwater temperature (°C)
met_data	year	Year during which measurements were taken (yyyy)
	month	Month during which measurements were taken (m)
	day	Day during which measurements were taken (d)
	hour	Hour at which measurements were taken (0=midnight, 23 =11 pm)
	minute	Minute at which measurements were taken (min)
	time_met	Times at which meteorological data were measured (min)
	solar_rad_in	Solar radiation data (W/m^2)
	air_temp_in	Air temperature (°C)
	rel_hum_in	Relative humidity (%)
	wind_speed_in	Wind speed (m/s)
bed_data1	dist_bed	Distances along the reach where bed temperature measurements were taken (m)
	depth_of_meas	Depth at which streambed temperatures were measured (m)
bed_data2	time_bed	Times at which bed temperatures were measured (min) [times are across the top row of this excel sheet]
	bed_temp	Streambed temperatures (°C)

		[temperatures at each time are listed below time_bed values and correspond to distances in bed_data1]
sed_type	Bed sediment classification: clay, silt, sand, or gravel (a cell array at the same spatial scale as the bed_data)	
shade_data	dist_shade	Distances along the reach where shade values were observed
	shade	Shade values, ranging from 0 to 1 with 0 being no shade and 1 being total shade
	vts	Values for the view to sky coefficient (0 being no view to sky, 1 being total view to sky)
cloud_data	time_cloud	Times at which cloud values were observed (min)
	cl_in	Values for cloud cover (0 to 1, with 0 being clear and 1 being overcast)
site_info	lat	Latitude of the site (positive for northern hemisphere, negative for southern)
	long	Longitude of the site (positive for eastern hemisphere, negative for western)
	t_zone	Indicates the time zone correction factor (East = 5, Central = 6, Mountain = 7, Pacific = 8) (Boyd and Kasper, 2003)
	z	Elevation of the site above sea level

The HFLUX package includes an example template for arranging the input data (example_data.xlsx). hflux_format.m reads the data from the Excel sheet and puts it into a form that is useable by hflux.m (a structure called "input_data").

Note: HFLUX uses MATLAB's built in function for linear interpolation interp1. This function produces NaNs if the data to be resampled (Y) do not start and end on the same or greater values as the x_i *values. If Y extends beyond the extent of the xi data, the interpolated array will stop at the maximum value of* x_i *and ignore the remaining values of Y. If Y does not match up or extend beyond xi, interp1 returns NaNs and hflux.m will not run correctly.*

3. Usage

To execute an example of an HFLUX model run, the following lines of code can be copied and paste directly into MATLAB in the command window. Details on the command syntax and additional options for model runs are outlined below.

```
[data] = hflux_format('example_data.xlsx');
[temp_mod matrix node flux] = hflux(data);
temp = xlsread('example_data.xlsx','temp');
[rel_err me mae mse rmse nrmse] =
hflux_error(data.time_mod,data.dist_mod,temp_mod,data.temp_t0_data(:,1),data.
temp_x0_data(:,1),temp);
[sens] = hflux_sens(data,[-0.01 0.01],[-2 2],[-0.1 0.1],[-0.1 0.1]);
```

3.1 Running hflux.m

An example of HFLUX input data is included as an Excel file (example_data.xlsx). Use the hflux_format.m file to read the data into the MATLAB workspace. This data will be used in the following description of the usage of HFLUX.
The command syntax to read in the data and run the model is:

```
[data] = hflux_format('example_data.xlsx')

[temp_mod matrix node flux] = hflux(data);
```

To suppress the display outputs and user interactions included in HFLUX, use the "output suppression" option included in the "settings" worksheet of the Excel input file. Alternatively, after input data has been read into the MATLAB workspace, manually change the value of data.settings(5) to produce or suppress output:

`data.settings(5)=0;` (to generate graphical output)
`data.settings(5)=1;` (to suppress graphical output)

This switch is particularly useful when HFLUX needs to be run multiple times, such as during a calibration routine or sensitivity analysis.

HFLUX has two options for solving the mass and energy balance equations: a Crank-Nicolson method and a second order Runge-Kutta method. The user can switch between these methods by changing the "solution method" value within the settings worksheet in the Excel input file. Alternatively, after input data has been read into the MATLAB workspace, the user can manually change the value of data.settings (1), similar to as described above for output suppression. A value of 1 uses the Crank-Nicolson method and 2 uses the Runge-Kutta method.

Note that the example dataset is intended for use with the Crank-Nicolson method, given the input spatial and temporal resolution (1 m and 1 min, respectively). The Runge-Kutta method requires use of a Courant time step to eliminate instability. In the case of the example data, the spatial and temporal resolution need to be set at 0.5 m and 0.05 min, respectively.

3.2 Other programs

HFLUX calls to several other m-files to calculate stream temperatures. hflux.m uses the heat fluxes from hflux_flux.m to calculate the source/sink term, R. While the command syntaxes for these associated functions are listed here, the user is not required to interact with these functions; rather, hflux.m controls all calls to them. But, if the user has need to modify the methods used to compute components of the stream heat budget model, these m-files can be easily modified and are annotated in detail to facilitate review and modification of methods.

The other m-files include:

hflux_bed.m: calculates the heat exchange between the water column and the stream bed using a temperature gradient and the thermal conductivity of saturated sediment (Lapham, 1989). The command syntax is

```
[bed] = hflux_bed(sed_type,water_temp,bed_temp,depth_of_meas)
```

hflux_bed_sed.m: converts the sed_type array from type 'cell' to type 'double' and then uses nearest neighbor interpolation to put the sediment array at the same spatial scale as dist_mod. The command syntax is:

```
[sed] = hflux_bed_sed(sed_type,dist_bed,dist_mod)
```

hflux_shortwave_refl.m: uses information about the position of the sun relative to the Earth and the stream and Fresnel's reflectivity to determine how much shortwave radiation is reflected off the surface of the stream at a given time (Boyd and Kasper, 2003; Holbert, 2007; Neumann and Pierson, 1966). The command syntax is:

```
[sol_refl] =
hflux_shortwave_refl(year,month,day,hour,minute,lat,lon,t_zone,t
ime_met,time_mod);
```

hflux_flux.m: calls to a series of m-files that calculate the five major heat fluxes. The command syntax is:

```
[net shortwave longwave atm back land latent sensible bed] =
hflux_flux(settings,solar_rad,air_temp,rel_hum,water_temp,wind_s
peed,z,sed_type,bed_temp,depth_of_meas,shade,vts,cl,sol_refl)
```

The five functions called by hflux_flux.m are as follows:

hflux_shortwave.m: calculates the shortwave radiation reaching the stream. The function has two options for this flux which can be manually selected by changing the "settings" values in the Excel input file, or setting the value of data.settings(2).

eq1=1: Correct the measured shortwave radiation for shading and the reflected shortwave radiation as calculated using hflux_shortwave_refl.m (see upcoming section) (Ouellet et al., 2012; Boyd and Kasper, 2003)

eq1=2: Correct the shortwave radiation for shading and an assumed value for albedo at the stream surface (Magnusson et al., 2012)

The command syntax is:

```
[shortwave] = hflux_shortwave(solar_rad,shade,sol_refl,eq1)
```

hflux_longwave.m: calculates the atmospheric longwave radiation, the back radiation off the stream surface, and the land cover radiation. The sum of these three fluxes is the total longwave radiation to the stream (Westhoff et al., 2007; Boyd and Kasper, 2003; Kustas et al., 1994;

Maidment, 1993). The command syntax is:

```
[longwave atm_rad back_rad land_rad] =
hflux_longwave(air_temp,water_temp,rel_hum,cl,vts)
```

hflux_latent.m: calculates the latent heat flux using one of two methods, which can be selected by changing the "settings" values in the Excel input file, or setting the value of data.settings(3).

eq2=1: Penman equation for open water
eq2=2: Mass transfer method

The command syntax is:

```
[latent] =
hflux_latent(water_temp,air_temp,rel_hum,wind_speed,shortwave,lo
ngwave,z,eq2)
```

hflux_sensible.m: calculates the latent heat flux using one of two methods, which can be selected by changing the "settings" values in the Excel input file, or setting the value of data.settings(4).

eq3=1: calculates the sensible heat flux using Bowen's ratio (Westhoff et al., 2007; Maidment, 1993; Magnusson et al., 2012; Boyd and Kasper, 2003).
eq3=2: direct calculation of sensible heat (Dingman, 1994)

The command syntax is:

```
[sensible] =
hflux_sensible(water_temp,air_temp,rel_hum,wind_speed,z,latent)
```

When using the direct calculation method for sensible heat, the user may want to change the height of vegetation (`z_veg`) and/or the height at which wind speed measurements were taken (`z_met`) to describe their site more accurately. To do this, change the variable within the hflux_sensible m-file.

4. Output Data and Figures

HFLUX outputs several structures of data. They are summarized in Table 2.

Table 2. Output data

<table>
<tr><th>Output Structure</th><th>Output Data</th><th>Description of Data</th></tr>
<tr><td>temp_mod</td><td colspan="2">Modeled stream temperatures through space (rows) and time (columns) (°C)</td></tr>
<tr><td rowspan="2">matrix_data</td><td>a and a_c</td><td>Water budget of the upstream node (future and current time step, respectively)</td></tr>
<tr><td>b and b_c</td><td>Water budget of the current node (future and current time step, respectively)</td></tr>
</table>

	c and c_c	Water budget of the downstream node (future and current time step, respectively)
	d	Energy budget of the current node
	A	Tridiagonal matrix with a on the subdiagonal, b on the diagonal, and c on the superdiagonal
	o,o_c,p,p_c,q,q_c, g,k, and m	Intermediate steps used to calculate b, b_c, and d
node_data	v	Volume of each node (m^3)
	q	Discharge measurement at each node (m^3/min)
	ql	Groundwater discharge at each node (m^3/min)
flux_data	heatflux	Net heat flux to the stream (W/m^2)
	solarflux	Shortwave radiation (W/m^2)
	solar_refl	Reflected shortwave radiation (W/m^2)
	long	Net longwave radiation (W/m^2)
	atmflux	Atmospheric longwave radiation (W/m^2)
	landflux	Landcover radiation (W/m^2)
	backrad	Back radiation from the stream (W/m^2)
	evap	Latent heat flux (W/m^2)
	sensible	Sensible heat flux (W/m^2)
	conduction	Stream bed conduction (W/m^2)

hflux.m also outputs four figures that summarize the results (Figures 1 to 4).

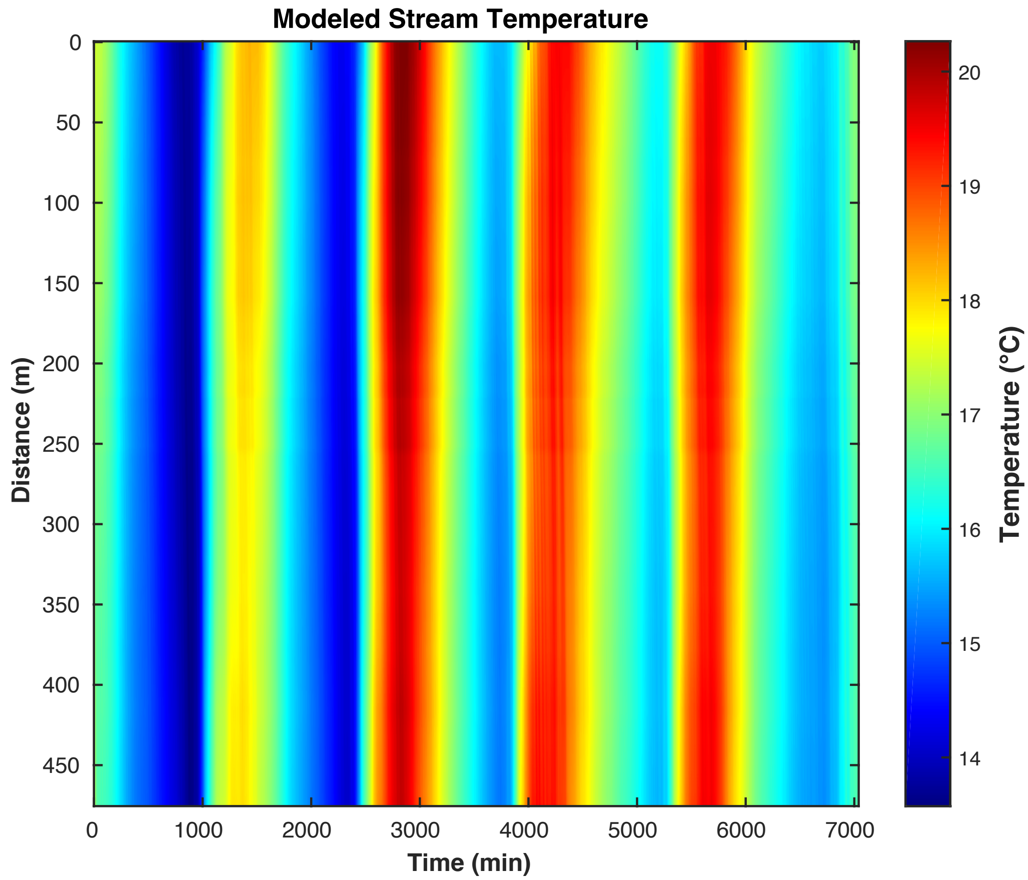

Figure 1. Example of plot of modeled stream temperature results from HFLUX.

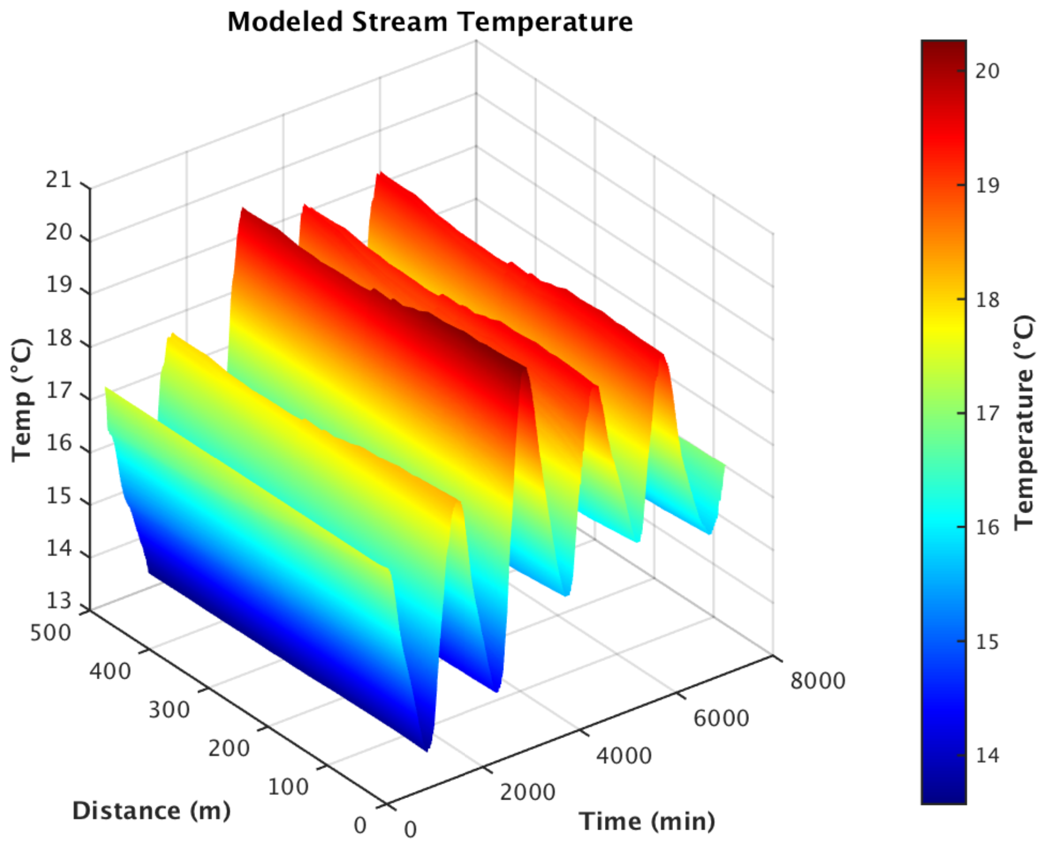

Figure 2. Example of a surface plot of modeled stream temperature from HFLUX.

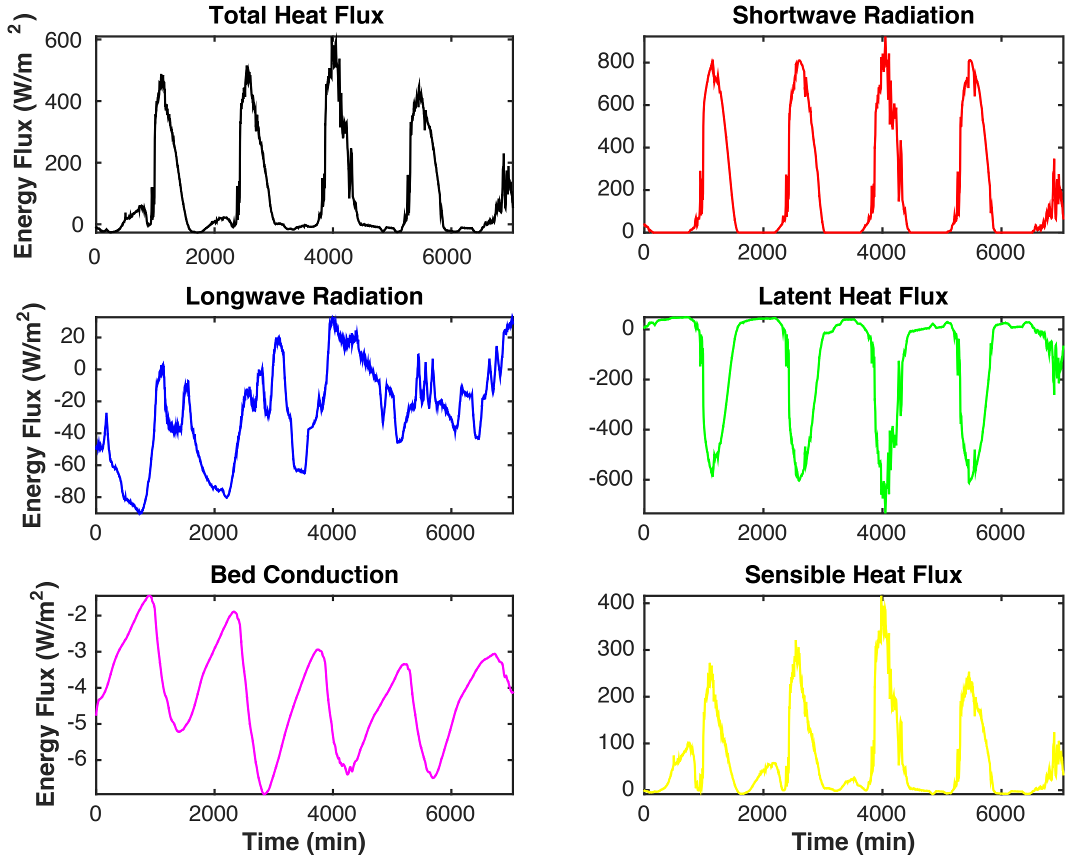

Figure 3. Example of plots of individual heat fluxes from HFLUX.

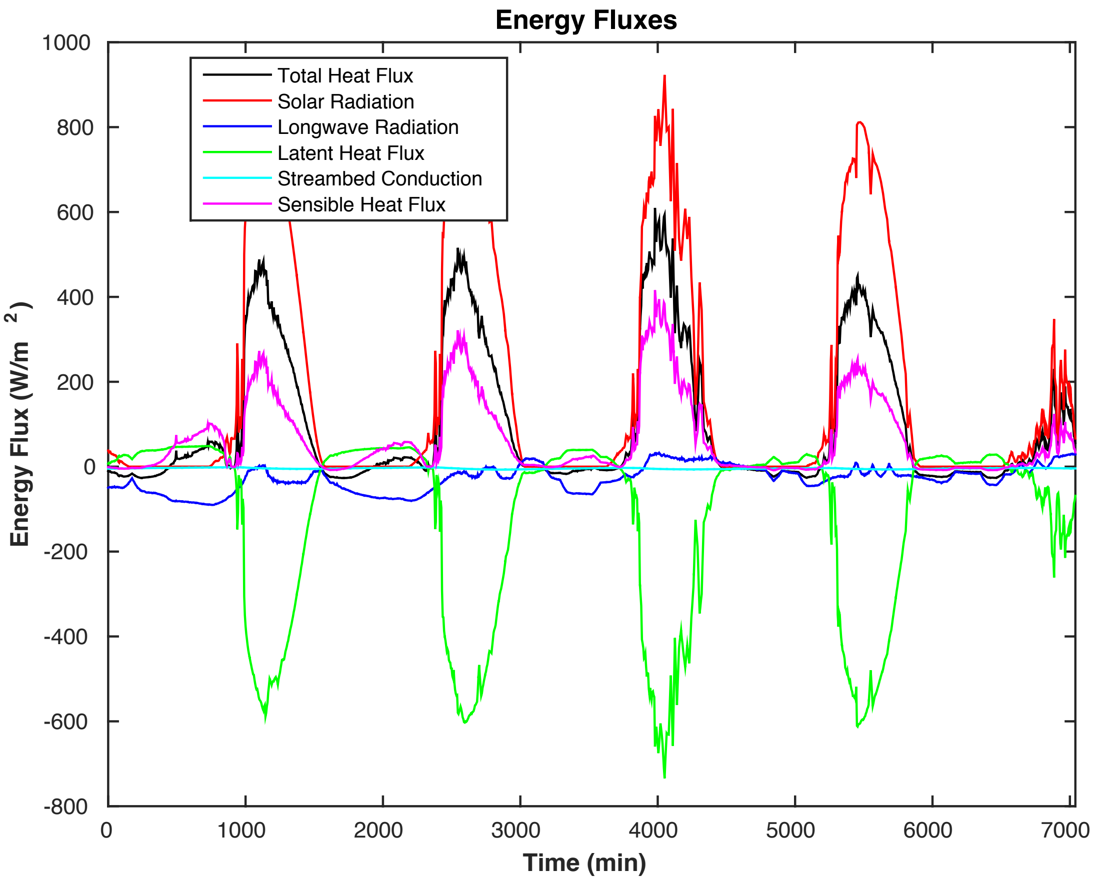

Figure 4. Example of plot comparing heat fluxes from HFLUX.

5. Running hflux_error.m

hflux_error.m is used to calculate various error metrics for the modeled stream temperatures. The program resamples the modeled data back to the spatial and temporal scale of the measured data, and then calculates relative error, mean error, mean absolute error, mean square error, root mean square error, and normalized root mean square error. The command syntax is:

```
[rel_err me mae mse rmse nrmse] =
hflux_error(time_mod,dist_mod,temp_mod,dist_obs,time_obs,temp_ob
s)
```

hflux_error.m returns a plot of the mean measured stream temperature along the reach compared to the mean modeled stream temperature along the reach and a plot of the mean measured stream temperature with respect to time compared to the mean modeled stream temperature with respect to time (Figure 5).

hflux_error.m has an optional unattend switch to suppress user interaction with the program. If using this switch, the command syntax is:

```
[rel_err me mae mse rmse nrmse] =
hflux_error(time_mod,dist_mod,temp_mod,dist_obs,time_obs,temp_ob
s,'unattended')
```

For the example data, use the following syntax:

```
temp=xlsread('example_data.xlsx','temp');

[rel_err me mae mse rmse nrmse]=
hflux_error(data.time_mod,data.dist_mod,temp_mod,
data.temp_t0_data(:,1),data.temp_x0_data(:,1),temp);
```

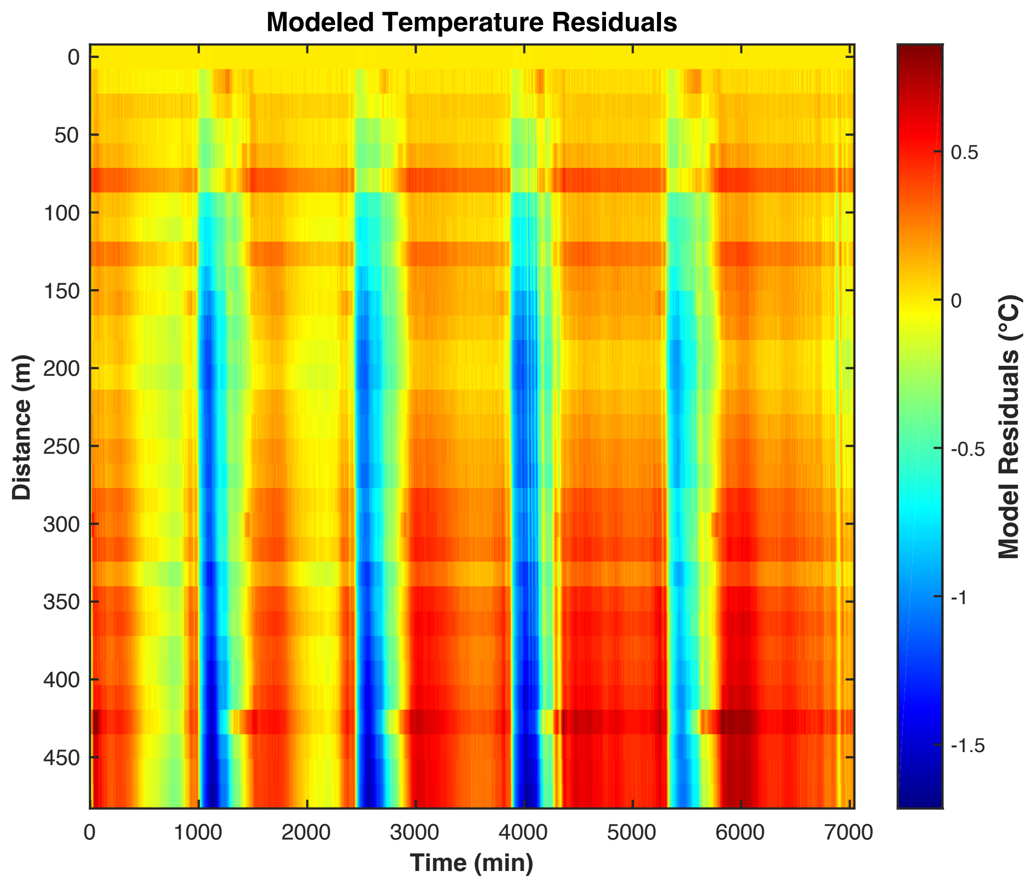

Figure 5. Example of a residual map plot comparing measured and modeled stream temperatures in space and time from hflux_error.m.

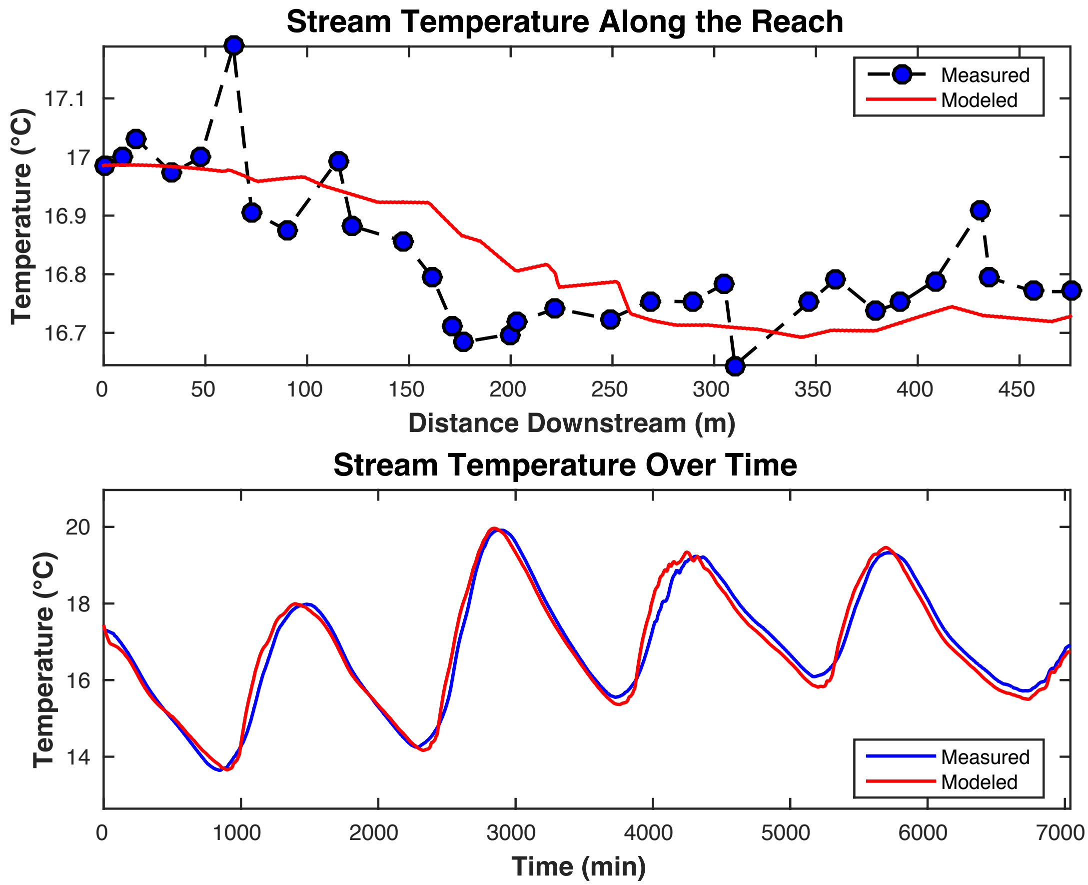

Figure 6. Example of plots comparing average measured and modeled stream temperature from hflux_error.m.

6. Running hflux_sens.m

The HFLUX package includes a sensitivity analysis function that allows the user to test the sensitivity of variables that change longitudinally. These include discharge, the view to sky coefficient, the temperature of lateral inflows, and shading. The function, hflux_sens.m, builds arrays of input data based on a range of values supplied by the user. It then runs hflux.m repeatedly to generate a plot of the high and low values compared to the base values. The command syntax is:

```
[sens] =
hflux_sens(input_data,dis_high_low,T_L_high_low,vts_high_low,sha
de_high_low)
```

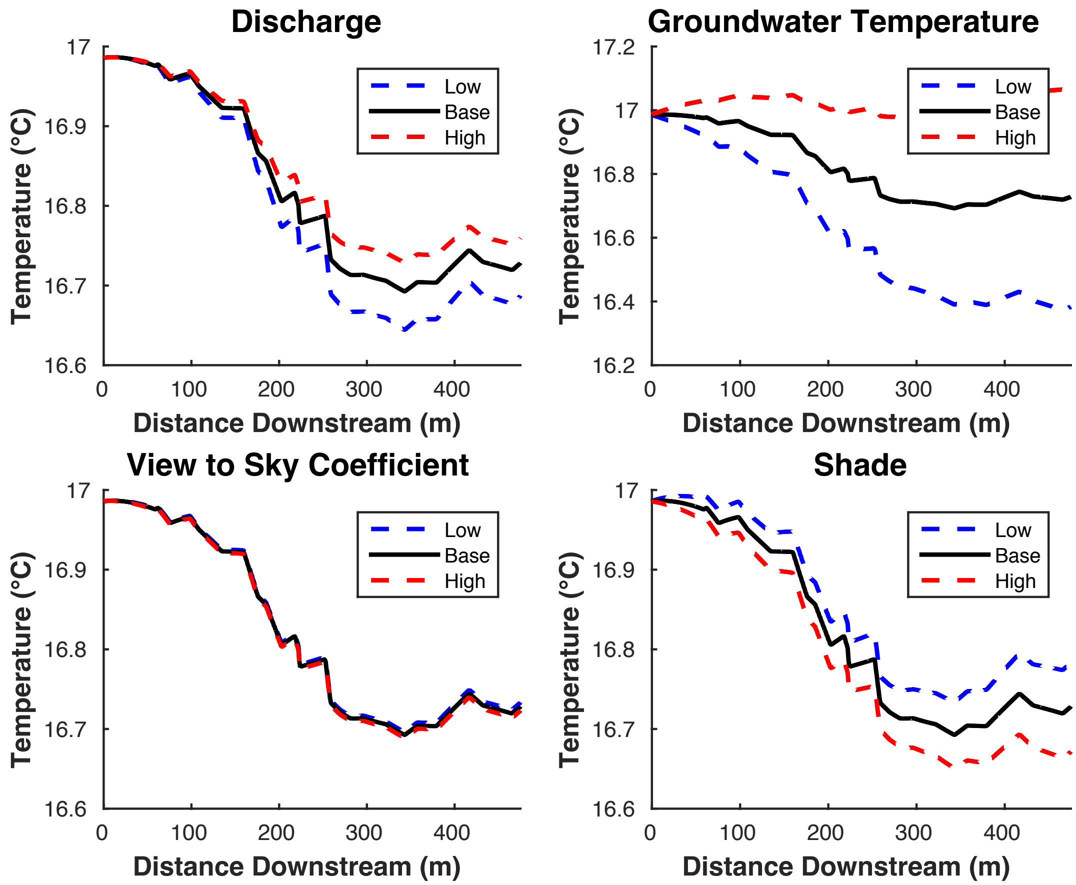

Figure 7. Example of plots displaying results of sensitivity analysis from hflux_sens.m.

where the output is a structure containing the arrays of input data with the high and low and values. `dis_high_low, T_L_high_low, vts_high_low, shade_high_low` are sets of values that the function uses to create the high and low valued arrays. For example, if the user wanted to assess the impact of increasing and decreasing discharge by 0.01 m^3/s, increasing and decreasing groundwater temperature by 2 degrees Celsius, increasing and decreasing the view to sky coefficient by 10%, and increasing and decreasing shade by 10%, the command syntax would look like this:

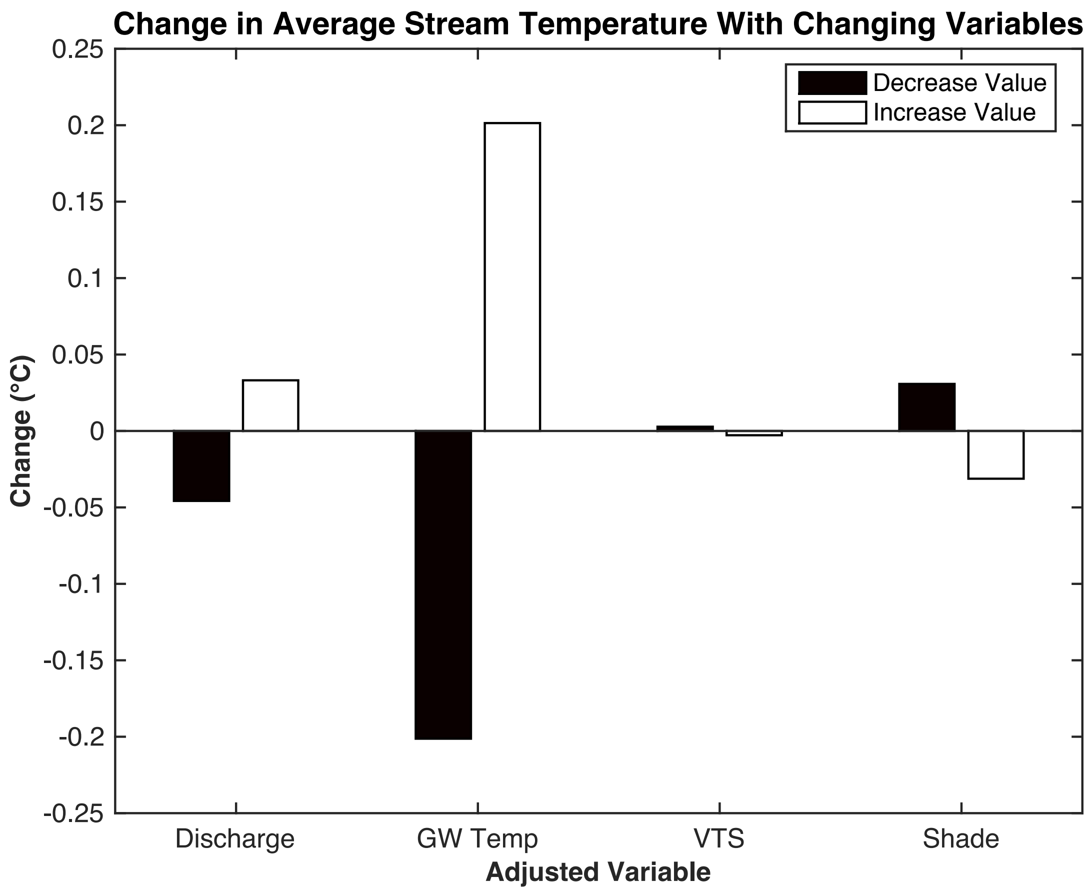

Figure 8. Example of a bar graph showing percent change in average stream temperature based on changes in variables from hflux_sens.m.

```
[sens] = hflux_sens(data,[-0.01 0.01],[-2 2],[-0.1 0.1],[-0.1
0.1])
```

hflux_sens.m generates a plot that shows how changing these variables changes the resulting stream temperatures along the reach (Figure 6) and a bar graph showing the percent change in average stream temperature due to each increase and decrease in value of the specific variables.

7. Authors

The HFLUX Stream Temperature Solver was created by AnneMarie Glose[1*], Laura K. Lautz[1], and Emily Baker[1].

[1] Syracuse University, Department of Earth Sciences, 204 Heroy Geology Lab, Syracuse, NY 13244, USA.

*Contact e-mail address: lklautz@syr.edu

References:

Boyd, M., and Kasper, B., 2003, Analytical methods for dynamic open channel heat and mass Transfer: Methodology for heat source model Version 7.0:.

Dingman, S.L., 1994, Physical Hydrology: Macmillan Publishing Company, New York.

Holbert, K.E., 2007, Solar calculations: , no. 4, p. 1–7.

Kustas, W., Rango, A., and Uijlenhoet, R., 1994, A simple energy budget algorithm for the snowmelt runoff model: Water Resources Research, v. 30, no. 5.

Lapham, W., 1989, Use of Temperature Profiles Beneath Streams to Determine Rates of Vertical Ground-Water Flow and Vertical Hydraulic Conductivity:, 1–44 p.

Loheide, S.P., and Gorelick, S.M., 2006, Quantifying stream-aquifer interactions through the analysis of remotely sensed thermographic profiles and in situ temperature histories.: Environmental science & technology, v. 40, no. 10, p. 3336–41.

Magnusson, J., Jonas, T., and Kirchner, J.W., 2012, Temperature dynamics of a proglacial stream: Identifying dominant energy balance components and inferring spatially integrated hydraulic geometry: Water Resources Research, v. 48, no. 6, p. 1–16, doi: 10.1029/2011WR011378.

Maidment, D.R. (Ed.), 1993, Handbook of Hydrology: McGraw-Hill.

Neumann, G., and Pierson, W., 1966, Principles of Physical Oceanography: Englewood Cliffs.

Ouellet, V., Secretan, Y., St-Hilaire, A., and Morin, J., 2012, Water temperature modelling in a controlled environment: comparative study of heat budget equations: Hydrological Processes,, p. n/a–n/a, doi: 10.1002/hyp.9571.

Recktenwald, G., 2011, Finite-difference approximations to the heat equation: Class Notes, v. 0, no. x.

Westhoff, M.C., Savenije, H.H.G., Luxemburg, W.M.J.., Stelling, G.S., Van de Giesen, N.C., Selker, J.S., Pfister, L., and Uhlenbrook, S., 2007, A distributed stream temperature model using high resolution temperature observations: Hydrology and Earth System Sciences Discussions, v. 4, no. 1, p. 125–149, doi: 10.5194/hessd-4-125-2007.