1D Diffusion: Predicting Soil Temperature

G582 Fall 2021

Caitlin Sifuentes

**Introduction**

In this exercise we were given varying soil temperatures down to 30ft below the surface. We were also provided with the air temperatures recorded for 279 days after the initial October 2nd measurement. Our hypothesis is that the observed variation in soil temperature is a result of heat diffusion from a seasonally fluctuating surface temperature and a constant heat flux from bedrock below rather than heat advection due to groundwater flow. We assume that the soil is sandy and highly permeable that remained dry during the study so groundwater has no influence on temperature. Our objective was to test our hypothesis by computing the time‐varying solutions to the 1D heat diffusion equation (given some initial and boundary conditions) and comparing those solutions to the observations.

**Derivation of the finite difference scheme**

1. General 1D Diffusion Equation

where T=f(x,t) and D is the thermal diffusivity of dry soil

1. Taylor Series

Forward difference

Backward difference

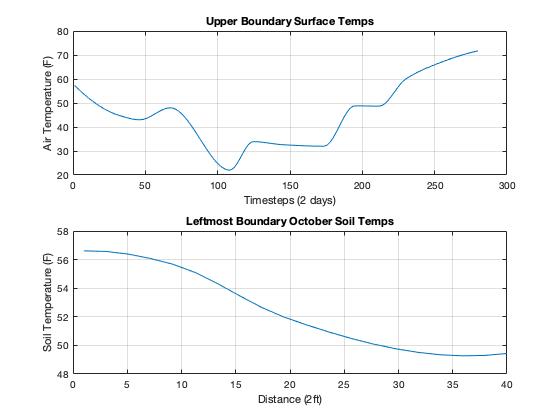
1. Central Differencing Approach

First derivative

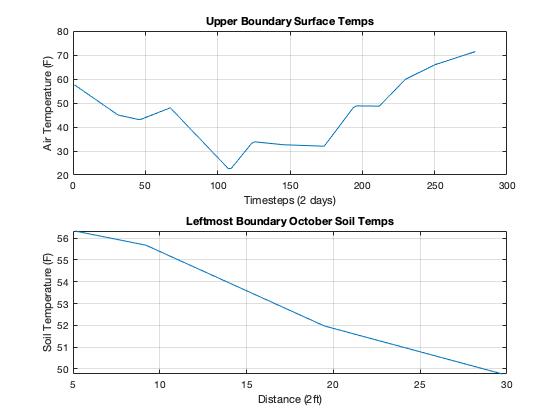
1. Finite Differencing Operator
2. Forward in Time

**Model**

To test the effects of diffusion on the soil temperature we initialize the model using upper and lower boundary conditions. The upper boundary condition for the model is the surface air temperatures collected at intervals for 279 days, and the lower boundary condition is the yearly average temperature of deep caves in this region. The initial condition we used for the leftmost boundary of our model was the October soil temperatures for every depth. The total run time for the model was set to 280 days, total depth was set to 40ft, and our chosen timestep and spatial step were 2. Total depth and run time were chosen to ensure the boundary conditions did not influence the modeled temperatures and the gradient from cell to cell was not too steep at the boundary. By keeping the timesteps and spatial steps at 2, our S value remained under the 0.5 value required to keep the model stable. To refine the resolution of the boundary temperatures, we interpolated surface air temperature data for every timestep in the total model run time and the initial October data for each spatial step in the total model depth. To interpolate the data, we used a cubic method of interpolation to ensure that we were not under or overfitting the data (Figure 1). An example of method that would produce underfit data would be a linear interpolation (Figure 2). Underfitting the data causes the model to inaccurately predict soil temperatures (Figure 6). After the model was initialized with the initial and boundary conditions we used the FTCS 1-D Diffusion equation to diffuse soil temperature across the model domain.

****

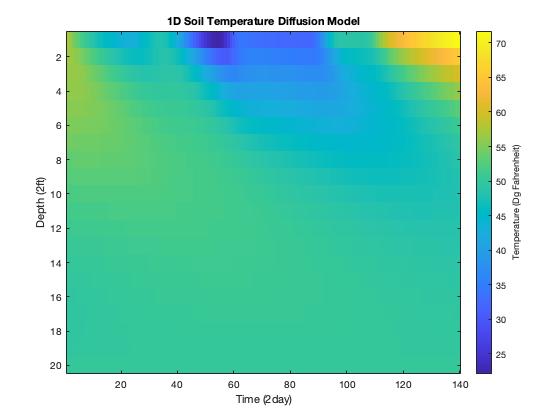
**Figure 1:** Interpolation of initial and boundary condition data using a ‘cubic’ method.

****

**Figure 2:** Example of a linear interpolation of creating underfit initial and boundary condition data.

**Results**

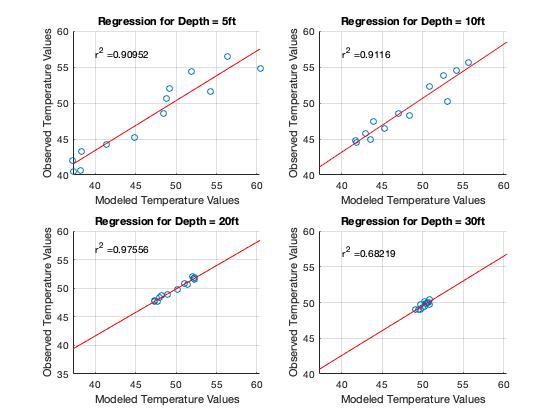
The modeled data shows soil temperature diffusing from varying surface temperatures ranging from 25 to 70 degrees Fahrenheit towards 50 degrees (Figure 3).

****

**Figure 3:** 1D Soil Temperature Diffusion Model. Timesteps and spatial steps are plotted on the x and y-axis.

**Analysis and Discussion**

To test how similar the modeled data was to the observed data we ran a statistical analysis consisting of root mean square error (rmse), Pearson’s correlation coefficient (r), and linear regression through depth. The rmse of the modeled data is 1.9157, meaning that the modeled data matches the observed data within ~2 degrees of error in temperature. When comparing the rmse to the range of standard deviations through the initial observed dataset (1.045-4.278), this value shows that the modeled data’s error is acceptable. In addition to rmse, the r-squared value comparing observed and modeled is 0.9008, showing that the modeled data is a near perfect fit to the observed data. Because of these two statistical tests, I can conclude to accept our hypothesis that the observed variation in soil temperature is a result of heat diffusion from a seasonally fluctuating surface air temperature. When looking at a linear regression between the modeled and observed data through depth, we see that the modeled and observed data show a linear relationship, and that towards 30ft depth both datasets converge towards 50 degrees Fahrenheit (Figure 4).

****

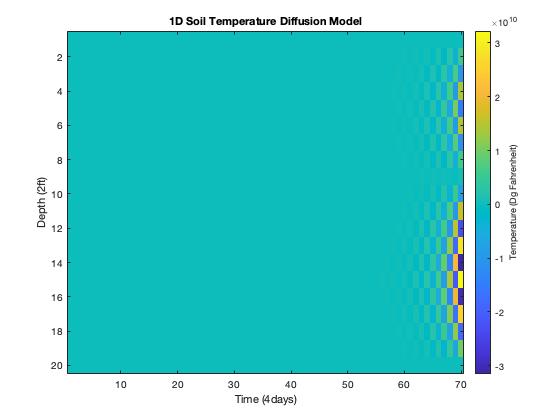
**Figure 4:** Linear regression plots of modeled to observed values of temperature through depth. R-squared values for each depth are displayed on each graph.

**Model Stability**

Model stability in this 1D Diffusion model is dependent on the value of S, which is defined as:

where D is the thermal diffusivity of dry soil, dt is timestep, and dx is spatial step

The model is stable if the value of S remains < 0.5. By keeping both the spatial step and time step values equal to 2 for the model, this keeps S equal to 0.3229, thus satisfying the stability of the model. To make the model unstable, we can change the value of dt to 4 days, causing S to equal 0.6458, making the model unstable (Figure 5).



**Figure 5:** Unstable model output resulting from an S-value equal to 0.6458.

Chart

Description automatically generated

**Figure 6**: Unstable model output resulting from underfit interpolated data.

**Appendix**

%% Exercise 2: 1D Diffusion: Predicting Soil Temperature

% G582

% Caitlin Sifuentes

% 09/01/21

clear

close all

%% Load the data

data = xlsread('SoilTdata.xlsx');

depth = data(2:5,1); % ft

% Observations

dates = data(1,2:14); % month.day

soil\_temp\_ob = data(2:5,2:14); % fahrenheit

% Constant

D = 0.06\*10.764; % ft2/day, thermal diffusivity, constant

% Initial Conditions

soil\_temp = data(2:5,2); % left column IC

soil\_temp\_depth = data(2:5,1);

% Variables

dt = 2; % days, timestep

dx = 2; % ft, space step

totaltime = 280; % total run time

totaldepth = 40; % total distance in ft

s = D\*(dt/(dx^2)); % S needs to be <0.5 to be stable

% Boundary Conditions

air\_temp = data(8,2:14); % fahrenheit , upper boundary

air\_temp\_day = data(7,2:14); % time for air temp measurement

query\_points\_at = (1:1:totaltime/dt);

query\_points\_st = (1:1:totaldepth/dx);

cave\_lowb = zeros(1,totaltime/dt); % initialize lower boundary

cave\_lowb = cave\_lowb+50; % % lower boundary condition, fahrenheit, yearly average temp of deep caves

% Interpolate Boundary Conditions

% use interp1

airtemp\_interp = interp1(air\_temp\_day/dt,air\_temp,query\_points\_at,'pchip'); % sample points, sample values, query points to int thru, using cubic (3rd order)

soiltemp\_interp = interp1(soil\_temp\_depth/dx,soil\_temp,query\_points\_st,'pchip')'; % interp oct soil values for y axis IC

air\_time = linspace(1,totaltime,totaltime/dt);

soil\_depth = linspace(1,totaldepth,totaldepth/dx);

% Plot Interpolated Data

subplot(2,1,1)

plot(air\_time,airtemp\_interp)

title('Upper Boundary Surface Temps')

xlabel('Timesteps (2 days)')

ylabel('Air Temperature (F)')

grid on

subplot(2,1,2)

plot(soil\_depth,soiltemp\_interp)

title('Leftmost Boundary October Soil Temps')

xlabel('Distance (2ft)')

ylabel('Soil Temperature (F)')

grid on

%% Intialize Matrix M

M = zeros(round(totaldepth/dx),round(totaltime/dt));

% Fill Matrix M with Boundary and Initial Conditions

M(:,1) = soiltemp\_interp;

M(1,:) = airtemp\_interp;

M(20,:) = cave\_lowb;

% Run FTCS Finite Diff Eq

for k = 1:totaltime/dt-1

for j = 2:totaldepth/dx-1

M(j,k+1) = s\*M(j+1,k)+(1-2\*s)\*M(j,k)+s\*M(j-1,k);

end

end

figure

imagesc(M)

title('1D Soil Temperature Diffusion Model')

xlabel('Time (2day)')

ylabel('Depth (2ft)')

a = colorbar;

a.Label.String = 'Temperature (Dg Fahrenheit)';

%% Test Statistical Similarity of Model Data and Observed Data

% "The hypothesis will be falsified if model predictions are not

% statistically similar to these data."

% RESIZE MATRIX

observed\_values = data(2:5,2:14); % select observed values from data

modeled\_vals = zeros(length(soil\_temp\_depth),length(air\_temp\_day)); % initialize new matrix

% Resize Modeled Matrix to exact days and depths of Observed to have equal sizes

for c = 1:length(air\_temp\_day)

for r = 1:length(depth)

modeled\_vals(r,c) = M(round(depth(r)/dx),round(air\_temp\_day(c)/dt));

end

end

% STATISTICAL ANALYSIS

dim = size(modeled\_vals);

N = (dim(1)\*dim(2)); % sample size for RMSE

rmse = (sum(sum((observed\_values - modeled\_vals).^2)/N)).^0.5;

stdev = std(modeled\_vals);

rval = corrcoef(modeled\_vals,observed\_values);

rsquare = rval.\*rval;

% Plot a linear regression for every depth

figure

subplot(2,2,1)

scatter(modeled\_vals(1,:),observed\_values(1,:))

c = polyfit(modeled\_vals(1,:),observed\_values(1,:),1);

y\_est = polyval(c,modeled\_vals);

hold on

plot(modeled\_vals,y\_est,'r')

rval1 = corrcoef(modeled\_vals(1,:),observed\_values(1,:));

rsquare1 = rval1.\*rval1;

txt1 = strcat('r^2 = ',num2str(rsquare1(2,1)));

text(40,57,num2str(txt1))

hold off

title('Regression for Depth = 5ft')

xlabel('Modeled Temperature Values')

ylabel('Observed Temperature Values')

grid on

subplot(2,2,2)

scatter(modeled\_vals(2,:),observed\_values(2,:))

c2 = polyfit(modeled\_vals(2,:),observed\_values(2,:),1);

y\_est2 = polyval(c2,modeled\_vals);

hold on

plot(modeled\_vals,y\_est2,'r')

rval2 = corrcoef(modeled\_vals(2,:),observed\_values(2,:));

rsquare2 = rval2.\*rval2;

txt2 = strcat('r^2 = ',num2str(rsquare2(2,1)));

text(40,57,num2str(txt2))

hold off

title('Regression for Depth = 10ft')

xlabel('Modeled Temperature Values')

ylabel('Observed Temperature Values')

grid on

subplot(2,2,3)

scatter(modeled\_vals(3,:),observed\_values(3,:))

c3 = polyfit(modeled\_vals(3,:),observed\_values(3,:),1);

y\_est3 = polyval(c3,modeled\_vals);

hold on

plot(modeled\_vals,y\_est3,'r')

rval3 = corrcoef(modeled\_vals(3,:),observed\_values(3,:));

rsquare3 = rval3.\*rval3;

txt3 = strcat('r^2 = ',num2str(rsquare3(2,1)));

text(40,57,num2str(txt3))

hold off

title('Regression for Depth = 20ft')

xlabel('Modeled Temperature Values')

ylabel('Observed Temperature Values')

grid on

subplot(2,2,4)

scatter(modeled\_vals(4,:),observed\_values(4,:))

c4 = polyfit(modeled\_vals(4,:),observed\_values(4,:),1);

y\_est4 = polyval(c4,modeled\_vals);

hold on

plot(modeled\_vals,y\_est4,'r')

rval4 = corrcoef(modeled\_vals(4,:),observed\_values(4,:));

rsquare4 = rval4.\*rval4;

txt4 = strcat('r^2 = ',num2str(rsquare4(2,1)));

text(40,57,num2str(txt4))

hold off

title('Regression for Depth = 30ft')

xlabel('Modeled Temperature Values')

ylabel('Observed Temperature Values')

grid on