**Detailed model analysis**

**Thermistor Calibration**

 The calibration offsets used to correct the raw data (as described in the methods section of the paper) are listed in table S1:

|  |  |  |
| --- | --- | --- |
| Sensor ID | Depth and Location on Probe (cm) | Temperature Offset (˚C) |
| 8061 |  +5 (bottom water) | -0.051 |
| 8062 |  -5 (sediment) | -0.051 |
| 8063 |  -10 (sediment) | -0.079 |
| 8064 |  -15 (sediment) | 0.032 |

**Model Details**

As a function of both time and material properties, TD controls a material’s response to transient heat transfer. The governing equation for three-dimensional conduction (the heat diffusion equation, equation S1), demonstrates how this transient term behaves in relation to time-independent conduction and heat generation.

$\frac{∂^{2}T}{∂x^{2}}+\frac{∂^{2}T}{∂y^{2}}+\frac{∂^{2}T}{A^{2}}+\frac{\dot{q}}{k}=\frac{1}{α}\frac{∂T}{∂t}$ (S1)

For conduction through marine sediments, many simplifications can be applied to the general equation. First, conduction is assumed to be one-dimensional; from the bottom water down through the sediment, but not laterally through the sediments. Next, it is assumed that no heat generation occurs, leaving only two terms in equation S1. The result is equation S2, the heat equation for transient conduction in a semi-infinite solid.

$\frac{∂^{2}T}{∂x^{2}}=\frac{1}{α}\frac{∂T}{∂t}$ (S2)

Last, the initial condition *T(x,0) = Ti*  and the interior boundary condition *T(x → ∞ ,t ) = Ti*  are applied, further simplifying equation S2 to equation S3.

$T\left(z,t\right)=\sum\_{i=1}^{n}∆T\_{i}\left(t\right)\*erfc\left(\frac{z}{\sqrt{4\*α\_{z}\*τ\_{i}}}\right)$ (S3)

In equation S3, *T(z,t)* is the temperature at the depth and time of interest: *∆Ti* is the difference in bottom water temperature over time-step *i*, *z* is the depth of interest, *τ* is the time range from the time of interest to the last observed time, and *α* is the TD at the depth of interest. For a series of temperature changes, TD can be calculated using a stepwise approach: this explains the summation term in equation 3, with each *i* term representing a single temperature step. The number of time steps, *n*, is one less than the number of observed data points. Equation 3 allows for downward propagation from the bottom water sensor, resulting in model fits that can be adjusted for a variety of possible TD values. To determine *T(z,t)* at the three depths of interest, the differences between consecutive points in the bottom water temperature sensor’s data were computed in Matlab version R2011a. This gave a *∆Ti* ­ vector with each value corresponding to a 1 minute timestep. The *τ* vector was then calculated, with

$τ\_{i}=t\_{n}-t\_{i}$ (S4)

where *tn* is the time of the last datapoint in the record and *ti* is the time of the current datapoint corresponding to the beginning of the *∆Ti* ­range. A vector was then generated for a range of TD values centered on an estimate of the expected TD values from Thomson (2010). These vectors were supplied to equation S2, generating a *T(z,t)*  matrix for each sensor depth (5 cm, 10 cm, and 15 cm) where the rows correspond to time steps and the columns correspond to TD values. Each column was then differenced with the observed data from that depth, and the standard deviations of those values were plotted as a function of the column’s TD (Figure 3).

**Sensitivity Analysis**

Of the variables in equation S3, the depth of interest (z) was the only parameter where error could be introduced during operation of the probe: if the probe was inserted to a different depth than the ideal, the sensors would be offset from the model depths. The use of a high definition camera during deployment allowed us to mitigate this error by visually inspecting the position of the sediment-bottom water interface along the scale on the probe. Thus we are confident that our deployment depths were within 0.25 cm of the “zero” (top sensor exactly 5 cm below the sediment-water interface). However, the examination of our model where z is varied by up to ±1cm provides a useful sensitivity analysis.

Our model was run with 100 input TDs, spaced evenly between 1x10-7 and 1x10-5 m2/s. This set of inputs was run 8 times for each deployment, with a different depth error introduced in each run. The figures below show these results in terms of the calculated TDs and the percent error between the depth-varied TD results and the “true” results. Here the top (blue diamonds), middle (orange squares), and bottom (green triangles) refer to the three sediment sensors, where top = 5 cm, middle = 10 cm, and bottom = 15 cm below the sediment-water interface for the “true” TD results.

*Deployment 1:*

*Deployment 2:*

The relationship between the TD and the depth error is very clear, with some repeated values on sensors with lower absolute temperature variation. As our deployments were photographically observed to be within 0.25 cm of the target depths, the sensitivity analysis indicates the maximum depth error to be less than 13.03% for deployment 1 and less than 14.98% for deployment 2.

**Matlab Code**

*Separate Depths (Analysis 1)*

close all; clear all; clc;

ll=3600; ul=12200; %Deployment 1

T1=open('Data&Calibration\Dep1\T1.mat'); T1=T1.T1(ll:ul,1);

T2=open('Data&Calibration\Dep1\T2.mat'); T2=T2.T2(ll:ul,1);

T3=open('Data&Calibration\Dep1\T3.mat'); T3=T3.T3(ll:ul,1);

T4=open('Data&Calibration\Dep1\T4.mat'); T4=T4.T4(ll:ul,1);

% ll=2000; ul=10000; %Dep 2

% T1=open('Data&Calibration\Dep2\T1.mat'); T1=T1.T1(ll:ul,1);

% T2=open('Data&Calibration\Dep2\T2.mat'); T2=T2.T2(ll:ul,1);

% T3=open('Data&Calibration\Dep2\T3.mat'); T3=T3.T3(ll:ul,1);

% T4=open('Data&Calibration\Dep2\T4.mat'); T4=T4.T4(ll:ul,1);

n=length(T1);

alpspace=linspace(log10(1E-7),log10(1E-5),100); %Dep 1

%alpspace=linspace(log10(1E-8),log10(1E-5),100); %Dep 2

alp=10.^(alpspace);

z=[.05,.1,.15]';

t=[0:60:(n-1)\*60]';

delta\_T1=diff(T1); delta\_T2=diff(T2); delta\_T3=diff(T3);

mT2=sum(T2)/n; mT3=sum(T3)/n; mT4=sum(T4)/n;

clear Tm2 mTm2 Tf2

for j=1:length(alp)

 for i=1:n-1

 tau=t(i+1)-t(1:i);

 den=sqrt(4\*alp(j).\*tau);

 Tm2(i,j)=erfc(z(1)./den')\*delta\_T1(1:i)+T1(length(T1));

 Tm3(i,j)=erfc(z(2)./den')\*delta\_T1(1:i)+T2(length(T2));

 Tm4(i,j)=erfc(z(3)./den')\*delta\_T1(1:i)+T3(length(T3));

 end

 %Offsetting to account for linear gradient

 mTm2(j)=sum(Tm2(:,j))/length(Tm2(:,j));

 mTm3(j)=sum(Tm3(:,j))/length(Tm3(:,j));

 mTm4(j)=sum(Tm4(:,j))/length(Tm4(:,j));

 Tf2(:,j)=Tm2(:,j)-(mTm2(j)-mT2);

 Tf3(:,j)=Tm3(:,j)-(mTm3(j)-mT3);

 Tf4(:,j)=Tm4(:,j)-(mTm4(j)-mT4);

 %Calculating Standard Deviations

 sd(j,1)=std(T2(2:n)-Tf2(:,j));

 sd(j,2)=std(T3(2:n)-Tf3(:,j));

 sd(j,3)=std(T4(2:n)-Tf4(:,j));

end

I2=find(sd(:,1)==min(sd(:,1)))

I3=find(sd(:,2)==min(sd(:,2)))

I4=find(sd(:,3)==min(sd(:,3)))

diff(1)=alp(I2)

diff(2)=alp(I3)

diff(3)=alp(I4)

figure(1)

subplot(4,1,1); plot(t,T1,'.g','markersize',2)

subplot(4,1,2); hold on;

plot(t(1:length(t)-1),Tf2); plot(t,T2,'.g','markersize',2);

subplot(4,1,3); hold on;

plot(t(1:length(t)-1),Tf3); plot(t,T3,'.g','markersize',2);

subplot(4,1,4); hold on;

plot(t(1:length(t)-1),Tf4); plot(t,T4,'.g','markersize',2);

xlabel('Time'); ylabel('Temperature [C]');

f2= figure(2)

plot(log10(alp),sd,'o-')

xlabel('log(Diffusivity)'); ylabel('Standard Deviation')

legend('5 cm','10 cm','15 cm');

title('Standard Deviation vs. Diffusivity');

figure(3)

plot(diff,z\*100,'.','markersize',20); axis ij

xlabel('Thermal Diffusivity [m^2/s]');

ylabel('Depth [cm]');

title('Diffusivity vs. Depth')

figure(4)

subplot(4,1,1); plot(t,T1,'.r','markersize',3)

subplot(4,1,2); hold on;

plot(t(1:length(t)-1),Tf2(:,I2)); plot(t,T2,'.r','markersize',3);

subplot(4,1,3); hold on;

plot(t(1:length(t)-1),Tf3(:,I3)); plot(t,T3,'.r','markersize',3);

subplot(4,1,4); hold on;

plot(t(1:length(t)-1),Tf4(:,I4)); plot(t,T4,'.r','markersize',3);

legend('Best Model Fit','Observed Data');

xlabel('Time [Seconds]'); ylabel('Temperature [C]');

figure(5)

plot(t,T1,'.r','marker','x','markersize',5); hold on;

plot(t,T2,'.b','marker','x','markersize',5);

plot(t(1:length(t)-1),Tf2(:,I2),'c','linewidth',1.5);

plot(t,T3,'.g','marker','x','markersize',5);

plot(t(1:length(t)-1),Tf3(:,I3),'y','linewidth',1.5);

plot(t,T4,'.k','marker','x','markersize',5);

plot(t(1:length(t)-1),Tf4(:,I4),'m','linewidth',1.5);

%axis([0 8600 4.8 6]);

xlabel('Time [Seconds]'); ylabel('Temperature [C]');

legend('Bottom Water Data', '5cm Data', '5cm Best Fit','10cm Data','10cm Best Fit', '15cm Data','15cm Best Fit')

t=t/3600;

f6 = figure(6)

subplot(3,1,1); plot(t,T1,'.r','marker','o','markersize',3); hold on;

plot(t,T2,'.b','marker','o','markersize',5);

plot(t(1:length(t)-1),Tf2(:,I2),'k','linewidth',1.5);

%title('5 cm Depth','fontsize',16,'Position',[.8,.8,0]);

%legend('5 cm');

%ylim([3.3 3.8]);

subplot(3,1,2); plot(t,T1,'.r','marker','o','markersize',3); hold on;

plot(t,T3,'.b','marker','o','markersize',5);

plot(t(1:length(t)-1),Tf3(:,I3),'k','linewidth',1.5);

ylabel('Temperature [C]')%,'fontsize',16);

%title('10 cm Depth','fontsize',16);

%legend('15 cm');

%ylim([3.3 3.8]);

subplot(3,1,3); plot(t,T1,'.r','marker','o','markersize',3); hold on;

plot(t,T4,'.b','marker','o','markersize',5);

plot(t(1:length(t)-1),Tf4(:,I4),'k','linewidth',1.5);

%title('15 cm Depth','fontsize',16);

xlabel('Time [Hours]')%,'fontsize',16);

%legend('15 cm');

%ylim([3.3 3.8]);

f7 = figure(7)

hold on;

plot(t(1:length(t)-1),Tf2(:,I2),'r','linewidth',1.5);

plot(t(1:length(t)-1),Tf3(:,I3),'b','linewidth',1.5);

plot(t(1:length(t)-1),Tf4(:,I4),'k','linewidth',1.5);

legend('5 cm','10 cm','15 cm');

xlabel('Time [Hours]');

ylabel('Temperature [C]');

% saveas(f2, 'Dep1SD','fig');

% saveas(f2, 'Dep1SD','png');

% saveas(f6, 'Dep1SepDepths','fig');

% saveas(f6, 'Dep1SepDepths','png');

% saveas(f7, 'Dep1ModelFits','fig');

% saveas(f7, 'Dep1ModelFits','png');

*Overall Values (Analysis 2)*

ll=3600; ul=12200; %Deployment 1

T1b=open('Data&Calibration\Dep1\T1.mat'); T1b=T1b.T1(ll:ul,1);

T12=open('Data&Calibration\Dep1\T2.mat'); T12=T12.T2(ll:ul,1);

T13=open('Data&Calibration\Dep1\T3.mat'); T13=T13.T3(ll:ul,1);

T14=open('Data&Calibration\Dep1\T4.mat'); T14=T14.T4(ll:ul,1);

T1=[T12;T13;T14]; n1=length(T1);

T1B=[T1b;T12;T13];

ll=2000; ul=10000; %Dep 2

T2b=open('Data&Calibration\Dep2\T1.mat'); T2b=T2b.T1(ll:ul,1);

T22=open('Data&Calibration\Dep2\T2.mat'); T22=T22.T2(ll:ul,1);

T23=open('Data&Calibration\Dep2\T3.mat'); T23=T23.T3(ll:ul,1);

T24=open('Data&Calibration\Dep2\T4.mat'); T24=T24.T4(ll:ul,1);

T2=[T22;T23;T24]; n2=length(T2);

T2B=[T2b;T22;T23];

alpspace=linspace(log10(1E-8),log10(1E-5),100);

alp=10.^(alpspace);

z=[.05,.1,.15]';

%Deployment 1

t1=[0:60:(n1-1)\*60]';

delta\_T1b=diff(T1B);

mT1=sum(T1)/n1;

for j=1:length(alp)

 for i=1:n1-1

 tau=t1(i+1)-t1(1:i);

 den=sqrt(4\*alp(j).\*tau);

 Tm1(i,j)=erfc(z(1)./den')\*delta\_T1b(1:i)+T1(length(T1));

 end

 %Offsetting to account for linear gradient

 mTm1(j)=sum(Tm1(:,j))/length(Tm1(:,j));

 Tf1(:,j)=Tm1(:,j)-(mTm1(j)-mT1);

 %Calculating Standard Deviations

 sd(j,1)=std(T1(2:n1)-Tf1(:,j));

end

%Deployment 2

t2=[0:60:(n2-1)\*60]';

delta\_T2b=diff(T2B);

mT2=sum(T2)/n2;

for j=1:length(alp)

 for i=1:n2-1

 tau=t2(i+1)-t2(1:i);

 den=sqrt(4\*alp(j).\*tau);

 Tm2(i,j)=erfc(z(1)./den')\*delta\_T2b(1:i)+T2(length(T2));

 end

 %Offsetting to account for linear gradient

 mTm2(j)=sum(Tm2(:,j))/length(Tm2(:,j));

 Tf2(:,j)=Tm2(:,j)-(mTm2(j)-mT2);

 %Calculating Standard Deviations

 sd(j,2)=std(T2(2:n2)-Tf2(:,j));

end

I1=find(sd(:,1)==min(sd(:,1)))

I2=find(sd(:,2)==min(sd(:,2)))

diff(1)=alp(I1)

diff(2)=alp(I2)

figure(1)

plot(log10(alp),sd,'o-')

xlabel('log(Diffusivity)'); ylabel('Standard Deviation')

legend('Deployment 1','Deployment 2');

title('Standard Deviation vs. Diffusivity');

figure(2)

subplot(2,1,1); plot(t1,T1B,'.r','marker','x','markersize',5); hold on;

plot(t1,T1,'.g','marker','x','markersize',5);

plot(t1(1:length(t1)-1),Tf1(:,I1),'m','linewidth',1.5);

subplot(2,1,2); plot(t2,T2B,'.r','marker','x','markersize',5); hold on;

plot(t2,T2,'.b','marker','x','markersize',5);

plot(t2(1:length(t2)-1),Tf2(:,I2),'c','linewidth',1.5);

xlabel('Time [Seconds]'); ylabel('Temperature [C]');

legend('Bottom Water Data', 'Dep 1 Data', 'Dep 1 Best Fit','Dep 2 Data','Dep 2 Best Fit')