Problem Set #5

Due February 28th, 2:50 PM (Submit to Canvas)

Problem 1 - Third Order Finite Difference Derive the central finite difference approximation of $d^3\phi/dx^3$ to order $\mathcal{O}(\Delta x^2)$. Determine what the leading order $\mathcal{O}(\Delta x^2)$ error is.

Central Finite $\rightsquigarrow$ $M+N-1 \rightarrow 3+2-1=4$] $\phi_{i-2}, \phi_{i-1}, \phi_{i+1}, \phi_{i+2}$

$\frac{d^3\phi}{dx^3} \rightarrow M=3 \qquad O(\Delta x^2) \rightarrow N=2$

$$\phi(x+\Delta x) = \phi(x) + \Delta x \frac{d\phi}{dx} + \frac{1}{2}\Delta x^2 \frac{d^2\phi}{dx^2} + \frac{1}{6}\Delta x^3 \frac{d^3\phi}{dx^3} + \frac{1}{24}\Delta x^4 \frac{d^4\phi}{dx^4} + O(\Delta x^5)$$

$$\phi(x-\Delta x) = \phi(x) - \Delta x \frac{d\phi}{dx} + \frac{1}{2}\Delta x^2 \frac{d^2\phi}{dx^2} - \frac{1}{6}\Delta x^3 \frac{d^3\phi}{dx^3} + \frac{1}{24}\Delta x^4 \frac{d^4\phi}{dx^4} + O(\Delta x^5)$$

$$\phi(x+2\Delta x) = \phi(x) + 2\Delta x \frac{d\phi}{dx} + \frac{(2\Delta x)^2}{2} \frac{d^2\phi}{dx^2} + \frac{(2\Delta x)^3}{6} \frac{d^3\phi}{dx^3} + \frac{(2\Delta x)^4}{24} \frac{d^4\phi}{dx^4} + O(\Delta x^5)$$

$$\phi(x-2\Delta x) = \phi(x) - 2\Delta x \frac{d\phi}{dx} + \frac{(2\Delta x)^2}{2} \frac{d^2\phi}{dx^2} - \frac{(2\Delta x)^3}{6} \frac{d^3\phi}{dx^3} + \frac{(2\Delta x)^4}{24} \frac{d^4\phi}{dx^4} + O(\Delta x^5)$$

throw throw keep throw

$$a\,\phi(x+\Delta x) = a\,\phi(x) \boxed{+ a\Delta x \frac{d\phi}{dx}} \boxed{+ a\frac{1}{2}\Delta x^2 \frac{d^2\phi}{dx^2}} \boxed{+ a\frac{1}{6}\Delta x^3 \frac{d^3\phi}{dx^3}} \boxed{+ a\frac{1}{24}\Delta x^4 \frac{d^4\phi}{dx^4}} + O(\Delta x^5)$$

$$b\,\phi(x-\Delta x) = b\,\phi(x) \boxed{- b\Delta x \frac{d\phi}{dx}} \boxed{+ b\frac{1}{2}\Delta x^2 \frac{d^2\phi}{dx^2}} \boxed{- b\frac{1}{6}\Delta x^3 \frac{d^3\phi}{dx^3}} \boxed{+ b\frac{1}{24}\Delta x^4 \frac{d^4\phi}{dx^4}} + O(\Delta x^5)$$

$$c\,\phi(x+2\Delta x) = c\,\phi(x) \boxed{+ c\,2\Delta x \frac{d\phi}{dx}} \boxed{+ c\frac{(2\Delta x)^2}{2} \frac{d^2\phi}{dx^2}} \boxed{+ c\frac{(2\Delta x)^3}{6} \frac{d^3\phi}{dx^3}} \boxed{+ c\frac{(2\Delta x)^4}{24} \frac{d^4\phi}{dx^4}} + O(\Delta x^5)$$

$$d\,\phi(x-2\Delta x) = d\,\phi(x) \boxed{- d\,2\Delta x \frac{d\phi}{dx}} \boxed{+ d\frac{(2\Delta x)^2}{2} \frac{d^2\phi}{dx^2}} \boxed{- d\frac{(2\Delta x)^3}{6} \frac{d^3\phi}{dx^3}} \boxed{+ d\frac{(2\Delta x)^4}{24} \frac{d^4\phi}{dx^4}} + O(\Delta x^5)$$

+

$$a - b + 2c - 2d = 0$$

$$\frac{1}{2}a + \frac{1}{2}b + 2c + 2d = 0$$

$$\frac{1}{6}a - \frac{1}{6}b + \frac{4}{3}c - \frac{4}{3}d = 1$$

$$\frac{1}{24}a + \frac{1}{24}b + \frac{2}{3}c + \frac{2}{3}d = 0$$

$$\left[\begin{array}{cccc|c} 1 & -1 & 2 & -2 & 0 \\ 0.5 & 0.5 & 2 & 2 & 0 \\ \frac{1}{6} & -\frac{1}{6} & \frac{4}{3} & -\frac{4}{3} & 1 \\ \frac{1}{24} & \frac{1}{24} & \frac{2}{3} & \frac{2}{3} & 0 \end{array}\right] \xrightarrow{\text{RREF}} \left[\begin{array}{c|c} 1000 & -1 \\ 0100 & 1 \\ 0010 & 0.5 \\ 0001 & -0.5 \end{array}\right]$$

$a = -1 \qquad c = .5$

$b = 1 \qquad d = -.5$

throw throw keep throw

$$-1\,\phi_{i+1} = -1\,\phi(x+\Delta x) = -1\,\phi(x) + (-1)\Delta x \frac{d\phi}{dx} + (-1)\frac{1}{2}\Delta x^2 \frac{d^2\phi}{dx^2} + (-1)\frac{1}{6}\Delta x^3 \frac{d^3\phi}{dx^3} + (-1)\frac{1}{24}\Delta x^4 \frac{d^4\phi}{dx^4} - \frac{1}{120}\Delta x^5 \frac{d^5\phi}{dx^5}$$

$$1\,\phi_{i-1} = 1\,\phi(x-\Delta x) = 1\,\phi(x) - (1)\Delta x \frac{d\phi}{dx} + (1)\frac{1}{2}\Delta x^2 \frac{d^2\phi}{dx^2} - (1)\frac{1}{6}\Delta x^3 \frac{d^3\phi}{dx^3} + (1)\frac{1}{24}\Delta x^4 \frac{d^4\phi}{dx^4} - \frac{1}{120}\Delta x^5 \frac{d^5\phi}{dx^5}$$

$$0.5\,\phi_{i+2} = 0.5\,\phi(x+2\Delta x) = 0.5\,\phi(x) + (0.5)\,2\Delta x \frac{d\phi}{dx} + (0.5)\frac{(2\Delta x)^2}{2}\frac{d^2\phi}{dx^2} + (0.5)\frac{(2\Delta x)^3}{6}\frac{d^3\phi}{dx^3} + (0.5)\frac{(2\Delta x)^4}{24}\frac{d^4\phi}{dx^4} + \frac{2}{15}\Delta x^5 \frac{d^5\phi}{dx^5}$$

$$-0.5\,\phi_{i-2} = -0.5\,\phi(x-2\Delta x) = -0.5\,\phi(x) - (0.5)\,2\Delta x \frac{d\phi}{dx} + (-0.5)\frac{(2\Delta x)^2}{2}\frac{d^2\phi}{dx^2} - (-0.5)\frac{(2\Delta x)^3}{6}\frac{d^3\phi}{dx^3} + (-0.5)\frac{(2\Delta x)^4}{24}\frac{d^4\phi}{dx^4} + \frac{2}{15}\Delta x^5 \frac{d^5\phi}{dx^5}$$

$+$

$$-\phi_{i+1} + \phi_{i-1} + 0.5\phi_{i+2} - 0.5\phi_{i-2} = \overbrace{\left(-\frac{1}{6} - \frac{1}{6} + \frac{2}{3} + \frac{2}{3}\right)}^{1}\frac{d^3\phi}{dx^3}\Delta x^3 + \underbrace{\left(-\frac{1}{120} - \frac{1}{120} + \frac{2}{15} + \frac{2}{15}\right)}_{0.25}\frac{d^5\phi}{dx^5}\Delta x^5$$

$$-\phi_{i+1} + \phi_{i-1} + 0.5\phi_{i+2} - 0.5\phi_{i-2} = \frac{d^3\phi}{dx^3}\Delta x^3 + 0.25\frac{d^5\phi}{dx^5}\Delta x^5$$

$$\frac{-\phi_{i+1} + \phi_{i-1} + 0.5\phi_{i+2} - 0.5\phi_{i-2} - 0.25\frac{d^5\phi}{dx^5}\Delta x^5}{\Delta x^3} = \frac{d^3\phi}{dx^3}$$

$$\boxed{\frac{d^3\phi}{dx^3} = \frac{-\phi_{i+1} + \phi_{i-1} + 0.5\phi_{i+2} - 0.5\phi_{i-2}}{\Delta x^3} - 0.25\Delta x^2 \frac{d^5\phi}{dx^5}}$$

Problem 2 - Finite Difference to Estimate the Derivative of a Noisy Signal:
We are going to investigate the impact of noise on the finite difference approximation of derivatives. The underlying signal is the function

$$\phi(x) = \frac{1}{2}\left(\sin(x) + \sin(2x)\right),$$

and template files for this problem are available on Canvas.

a. Write a function **noisy_signal** that takes in a noise amplitude **A** and a step resolution **dx** and outputs the base signal, **phi**, and the signal plus noise, **phi_noise**. The range of values that should be output is for $0 \leq x \leq 2\pi$. Create the noise by using the function **rand**. The **rand** function produces a random number uniformly distributed between zero and one each time it is called. For this problem, we want to transform this range from [0, 1] to $[-A, A]$. The inputs into **rand** set the size of the output. *(Hint: the Matlab documentation on* **rand** *gives an example of how to do this.)*

(Function)

Problem 2A Function Code (noisy_signal.m):

```
function [phi, phi_noise] = noisy_signal(A, dx)
    %% Create the domain
```

```
    x = 0:dx:2*pi;  % Assuming you want to create the signal over the interval [0,
2π]

    %% Set the base function
    phi = 0.5 * (sin(x) + sin(2*x));

    %% Add noise to the base function
    noise = rand(size(x)); % First generate noise in the range [0, 1]
    noise = 2 * A * noise; % Scale to range [0, 2A]
    noise = noise - A; % Shift to range [-A, A]
    phi_noise = phi + noise; % Add noise to phi to get the noisy signal
end


%% Testing Function on Command Window and Plotting
%[phi, phi_noise] = noisy_signal(0.5, 0.1);
% x = 0:0.1:2*pi; % range of x is from 0 to 2*pi with steps of 0.1 as per your
function definition

% Plot the base signal
% plot(x, phi, 'b', 'LineWidth', 2);
% hold on; % This command allows you to plot multiple lines on the same figure

% Plot the noisy signal
% plot(x, phi_noise, 'r--', 'LineWidth', 1.5);
% hold off; % This command ends the multiple line plotting
% xlabel('x');
% ylabel('\phi(x)');
% legend('Base Signal', 'Noisy Signal');
% title('Comparison of Base and Noisy Signals');
```

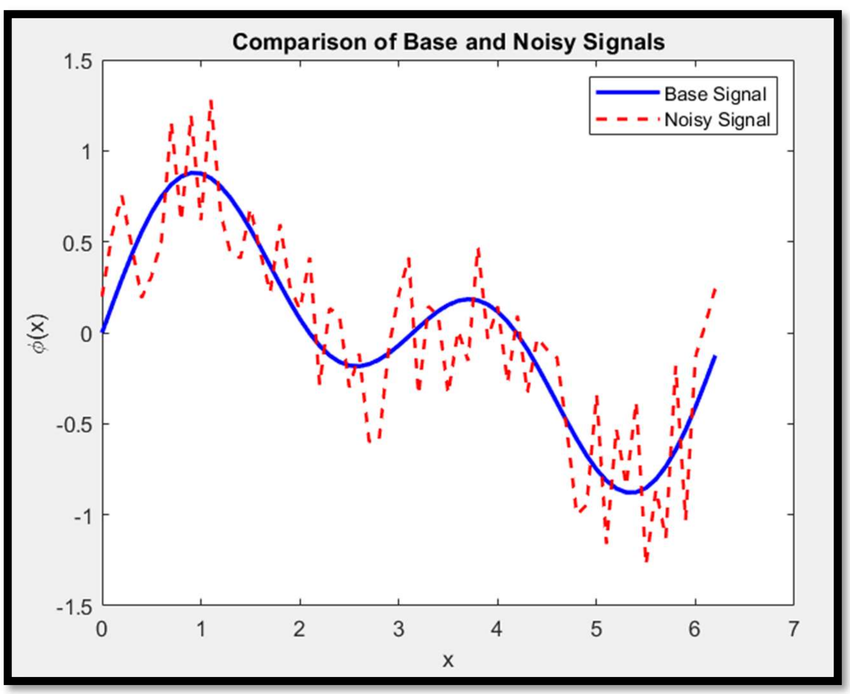


Figure 1: Testing function with graph plot (not necessary for question but just to check if function works). Relationship between phi and x.

b. Write a script that calculates the first derivative of both the base function $\phi(x)$ and the noisy function $\phi_{noise}(x)$ using the second order central difference approximation

$$\left.\frac{d\phi}{dx}\right|_n = \frac{\phi_{n+i} - \phi_{n-i}}{2i\Delta x} + \mathcal{O}\left(\Delta x^2\right),$$

where normally we set $i = 1$. Apply this central difference with $i = 1$ for both ϕ and ϕ_{noise} and plot the result. Also plot the analytic derivative, for comparison. *(Note: there is no calculation performed for points where ϕ_{n-i} or ϕ_{n+i} do not exist.)* Comment on how well the analytic derivative is approximated using the central difference in the presence of noise. Set $\Delta x = \pi/40$, $A = 0.2$.

(Script, Plot, Comments)

Comments:

The central difference method yields a derivative approximation that closely tracks the analytic derivative for the noise-free base signal, as evidenced by the blue line's alignment with the black line. However, the red line representing the derivative of the noisy signal fluctuates widely, reflecting the amplified noise's influence. This highlights the method's sensitivity to noise, which can obscure the true signal characteristics. Despite this limitation, the central difference method remains a valuable tool for computing derivatives, especially when applied to clean, smooth data where it provides accurate estimates of the underlying function's rate of change.

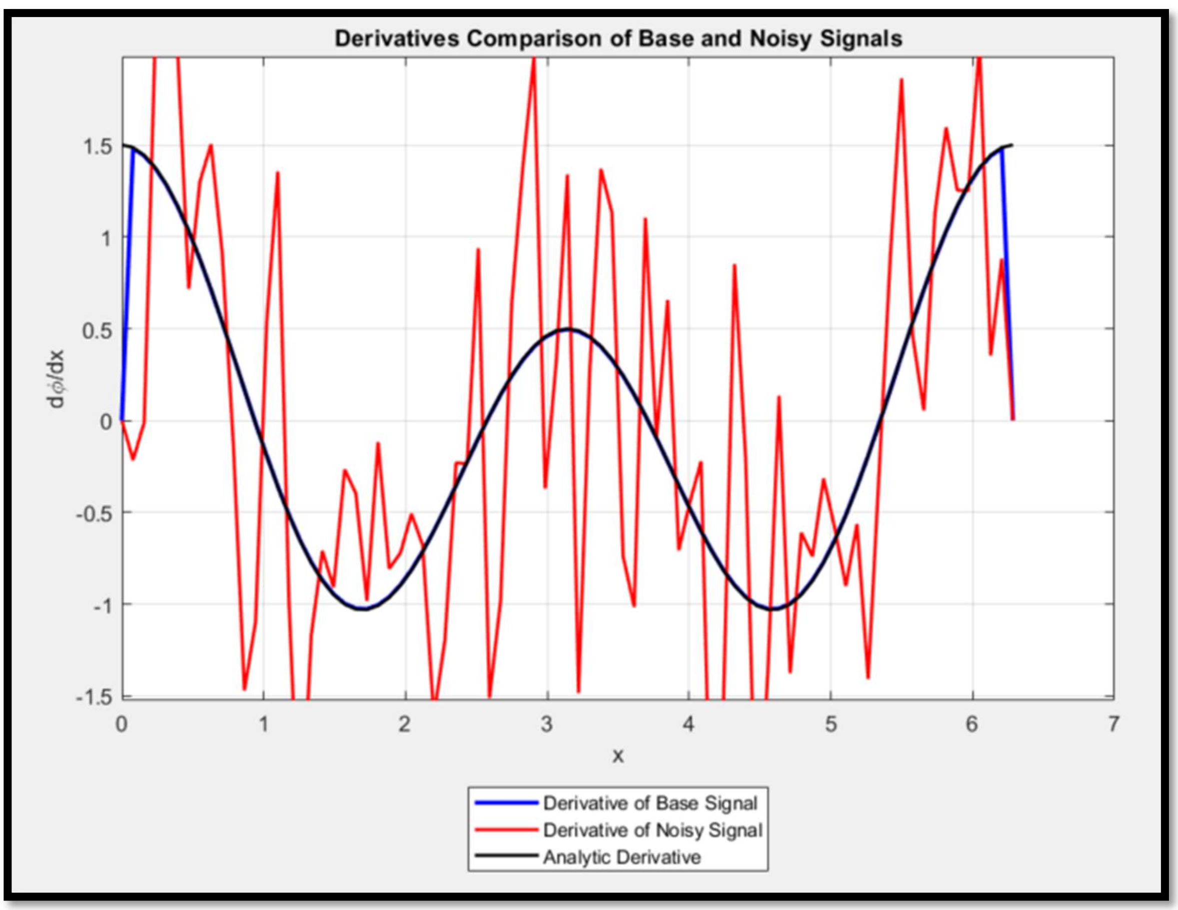


Figure 2: Derivative of Theta vs x using derivative of base signal and derivative of noisy signal. Plot also shows analytic derivative.

Problem 2B Function Code (HW51B.m):

```
% Set the parameters
A = 0.2; % Amplitude of noise
dx = pi/40; % Step size
x = 0:dx:2*pi; % Domain

% Generate the signals
[phi, phi_noise] = noisy_signal(A, dx);

% Initialize derivatives
phi_deriv = zeros(size(x));
phi_noise_deriv = zeros(size(x));

% Compute the central differences for phi and phi_noise
for i = 2:length(x)-1
    phi_deriv(i) = (phi(i+1) - phi(i-1))/(2*dx); % Central difference approximation
of derivative for base signal
    phi_noise_deriv(i) = (phi_noise(i+1) - phi_noise(i-1))/(2*dx); % Central
difference approximation of derivative for noisy signal
end

% Analytic derivative of phi
dphi_dx = 0.5 * (cos(x) + 2*cos(2*x)); % Analytical derivative of the base signal

% Plot the results with adjusted visuals
figure;
plot(x, phi_deriv, 'b', 'LineWidth', 2); % Increased line width and changed to dashed
line for base signal derivative
hold on; % Keep the plot for overlaying the next plots
plot(x, phi_noise_deriv, 'r', 'LineWidth', 1.5); % Noisy signal derivative
plot(x, dphi_dx, 'k', 'LineWidth', 1.5); % Analytic derivative
hold off; % Release the plot hold

% Add grid lines
grid on;

% Adjust the y-axis limits to better view the base signal derivative
ylim([min(phi_deriv)-0.5, max(phi_deriv)+0.5]); % Adjust y-axis limits to base signal
derivative range with some padding

xlabel('x'); % Label for x-axis
ylabel('d\phi/dx'); % Label for y-axis
title('Derivatives Comparison of Base and Noisy Signals'); % Title for the plot

% Create the legend and place it outside the plot area
legend('Derivative of Base Signal', 'Derivative of Noisy Signal', 'Analytic
Derivative', 'Location', 'southoutside'); % Updated legend

% Optionally, you can increase the figure size for better visibility
set(gcf, 'Position', [100, 100, 800, 600]); % Set the figure window size: [left,
bottom, width, height] in pixels
```

c. Calculate the **average** absolute error of both finite difference calculations (using ϕ and ϕ_{noise}), considering all points where the derivative is approximated. How does the error for the noisy data set compare to the value of A (i.e., has the noise amplified or diminished by approximating the derivative)?

(Average Errors, Comments)

```
>> HW5P1C
Average absolute error for phi: 0.002602
Average absolute error for phi_noise: 0.885727
``
```

Comments:

Based on the output provided from running the MATLAB script, the average absolute error for the base function Ø is 0.002602, and for the noisy function Ø_noise it is 0.8857270.885727.

This result demonstrates that the noise in the data set significantly affects the accuracy of the derivative approximation. The error in the noisy data set is considerably larger than the error in the noise-free data set. The noise's amplitude, determined by parameter A, was set to 0.2 in this case, and the error for the noisy data set is more than four times this value. This indicates that the process of computing the derivative has indeed amplified the noise, as the average absolute error is much larger than the noise amplitude itself.

The central difference method is sensitive to noise because it involves subtracting nearby function values. If the function values contain random noise, the subtraction can result in large errors, hence the significant increase in the average absolute error for the noisy signal compared to the clean signal.

Problem 2C Function Code (HW52C.m):

```
clear; close all; clc;

% Set the parameters
A = 0.2; % Amplitude of noise
dx = pi / 40; % Step size
x = 0:dx:2*pi; % Domain including the endpoint for generating signals

% Generate the signals using the noisy_signal function
[phi, phi_noise] = noisy_signal(A, dx);

% Initialize derivatives, leaving out the first and last point
phi_deriv = zeros(1, length(x)-2);
phi_noise_deriv = zeros(1, length(x)-2);

% Compute the central differences for phi and phi_noise
for i = 2:length(x)-1
    phi_deriv(i-1) = (phi(i+1) - phi(i-1)) / (2*dx); % Central difference
approximation of derivative for base signal
    phi_noise_deriv(i-1) = (phi_noise(i+1) - phi_noise(i-1)) / (2*dx); % Central
difference approximation of derivative for noisy signal
end

% Compute the analytical derivative directly
dphi_dx_analytical = 0.5 * (cos(x(2:end-1)) + 2 * cos(2*x(2:end-1)));
```

```
% Calculate the absolute errors
errors_phi = abs(phi_deriv - dphi_dx_analytical);
errors_phi_noise = abs(phi_noise_deriv - dphi_dx_analytical);

% Calculate the average absolute errors
average_error_phi = mean(errors_phi);
average_error_phi_noise = mean(errors_phi_noise);

% Display the results
fprintf('Average absolute error for phi: %f\n', average_error_phi);
fprintf('Average absolute error for phi_noise: %f\n', average_error_phi_noise);

% Function to generate the noisy signal phi_noise based on phi
function [phi, phi_noise] = noisy_signal(A, dx)
    x = 0:dx:2*pi;  % Create the domain over the interval [0, 2π]
    phi = 0.5 * (sin(x) + sin(2*x)); % base signal phi using the given formula
    noise = rand(1, length(x)) * 2 * A - A; % random noise within the specified
amplitude range
    phi_noise = phi + noise; % Add the noise to the base signal to create the noisy
signal phi_noise
end
```

d. Repeat parts **b** and **c** with $i = 1:1:8$ and $A = 0.1$. Plot the average absolute error as a function of the gap size $i\Delta x$. Based on this plot, what is the best gap size for calculating the derivative from the analytic data, and which one is best for the noisy data? How do the results with and without noise compare for the largest gap sizes, and what is the reason for this similarity or difference? How do you think the result for the noisy data will change if the noise amplitude is increased?

(Script, Plot, Comments)

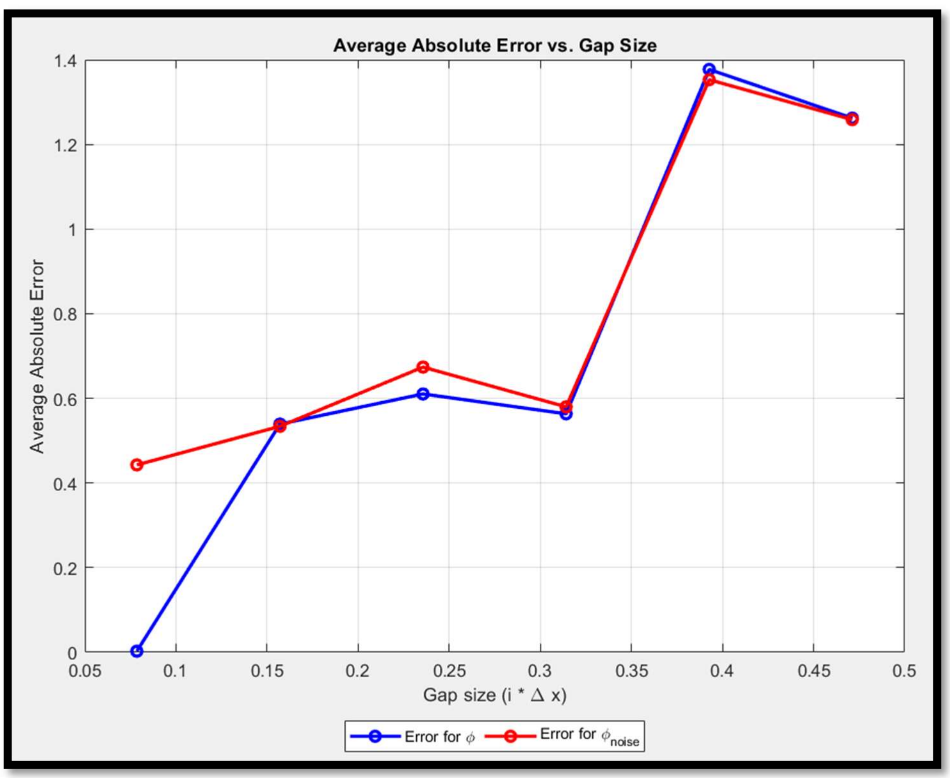


Figure 3: Average Absolute Error vs Gap Size for Part D plot

Comments:

From the plot, it appears that the optimal gap size for calculating the derivative, both for the analytic data and the noisy data, falls within the range corresponding to i = 3 or i = 4 which translates to a gap size of 0.15 to 0.20 on the x-axis. This suggests that a balance between too small and too large a gap size yields the most accurate derivative estimation. Interestingly, the results for clean and noisy data converge at larger gap sizes, showing a significant increase in the average absolute error. This trend indicates that while a larger gap size reduces the sensitivity to noise, it also decreases the precision of the derivative estimate, suggesting a compromise between noise mitigation and accuracy. If the noise amplitude were to increase, we would expect the average absolute error for the noisy data to also increase, particularly at smaller gap sizes where the noise's influence is more pronounced, thus further compromising the derivative's accuracy.

Problem 2D Function Code (HW52D.m):

```
clear; close all; clc;
% Set the parameters
A = 0.1; % Amplitude of noise is now set to 0.1
dx = pi/40; % Step size
gap_multipliers = 1:8; % i values from 1 to 8
errors_phi = zeros(1, length(gap_multipliers));
errors_phi_noise = zeros(1, length(gap_multipliers));

% Loop over each gap size
for i = gap_multipliers
    current_dx = i * dx;
    x = 0:current_dx:2*pi; % Adjust domain for current gap size

    % Generate the signals using the noisy_signal function
    [phi, phi_noise] = noisy_signal(A, current_dx);

    % Compute the central differences for phi and phi_noise
    % Initialize derivatives, considering the gap size
    phi_deriv = zeros(1, length(x)-2*i);
    phi_noise_deriv = zeros(1, length(x)-2*i);

    for k = i+1:length(x)-i
        phi_deriv(k-i) = (phi(k+i) - phi(k-i)) / (2*current_dx); % Central difference
        phi_noise_deriv(k-i) = (phi_noise(k+i) - phi_noise(k-i)) / (2*current_dx); %
Central difference
    end

    % Analytic derivative of phi
    analytic_deriv = 0.5 * (cos(x(i+1:end-i)) + 2*cos(2*x(i+1:end-i)));

    % Calculate the absolute errors for phi and phi_noise
    errors_phi(i) = mean(abs(phi_deriv - analytic_deriv));
    errors_phi_noise(i) = mean(abs(phi_noise_deriv - analytic_deriv));
end

% Plotting the average absolute error as a function of gap size
figure;
```

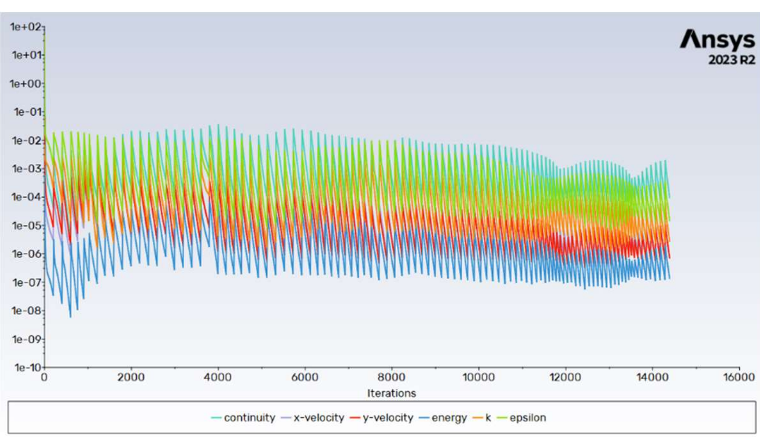
```
plot(gap_multipliers*dx, errors_phi, 'b-o', 'LineWidth', 2);
hold on;
plot(gap_multipliers*dx, errors_phi_noise, 'r-o', 'LineWidth', 2);
hold off;
xlabel('Gap size (i * \Delta x)');
ylabel('Average Absolute Error');
title('Average Absolute Error vs. Gap Size');
% Set the legend below the graph and make it horizontal
legend('Error for \phi', 'Error for \phi_{noise}', 'Location', 'southoutside',
'Orientation', 'horizontal');
grid on;

% Function to generate the noisy signal phi_noise based on phi
function [phi, phi_noise] = noisy_signal(A, dx)
    x = 0:dx:2*pi;  % Create the domain over the interval [0, 2π]
    phi = 0.5 * (sin(x) + sin(2*x)); % Generate the base signal phi
    noise = rand(1, length(x)) * 2 * A - A; % Generate random noise
    phi_noise = phi + noise; % Create the noisy signal phi_noise
end
```

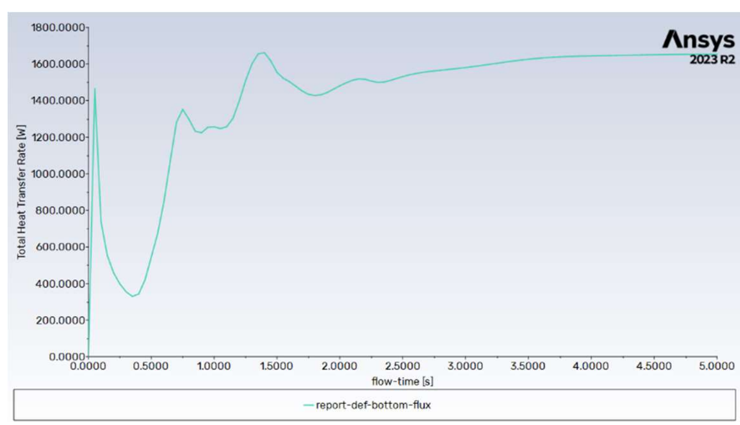
Problem 3 - Convection Cells and Heat Transfer using Ansys:
This week's problem investigates the convection cells that develop when water is heated from below. Starting from the setup studied in class, we are going to look at the output data produced using different thermal diffusivities κ. Using a fixed time step of $\Delta t = 0.05$, take 100 time steps with each time step having a maximum of 200 iterations per step and a reporting interval of 10 iterations. Create the report variables that measure the heat flux through the bottom and top of the domain.

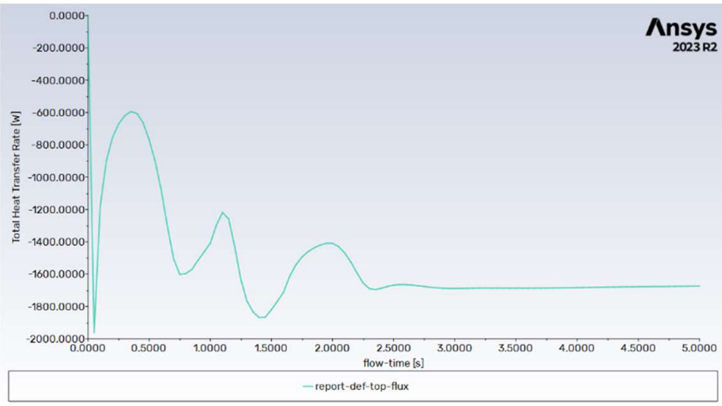
a. Run the system using air as the fluid three times, with a thermal diffusivity of $\kappa_1 = 0.0121$, $\kappa_2 = 0.0242$ and $\kappa_3 = 0.0484$ m^2/s. For each of the simulations:

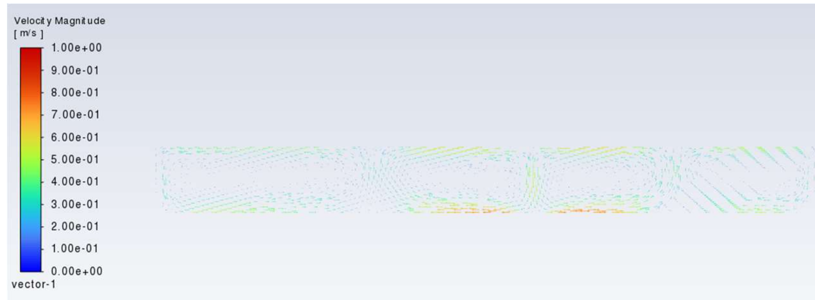
a1. Produce a couple screen shots of the time evolving temperature field.
(Images)

Part a1 Images









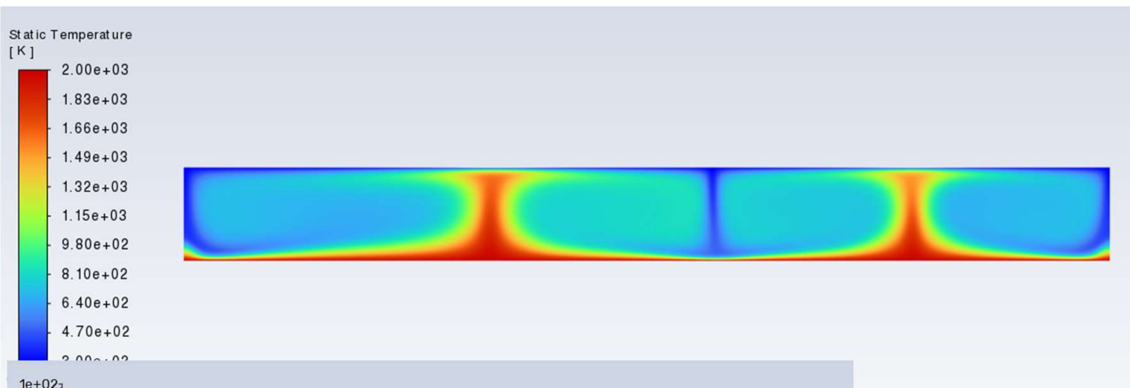
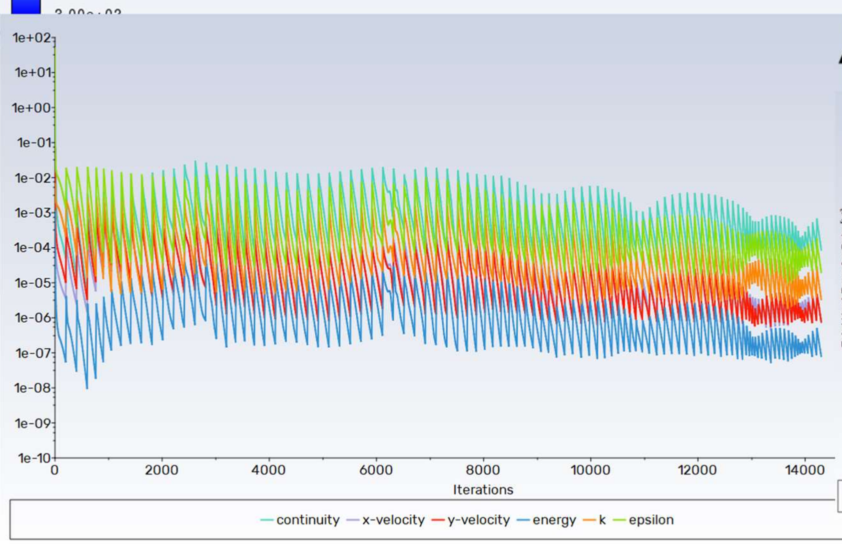
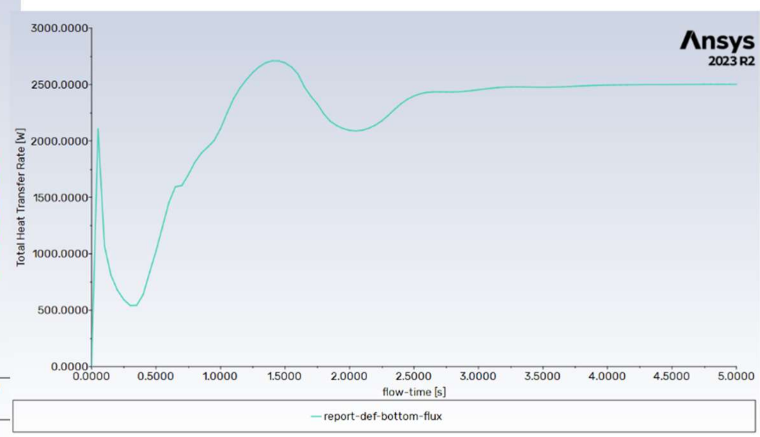
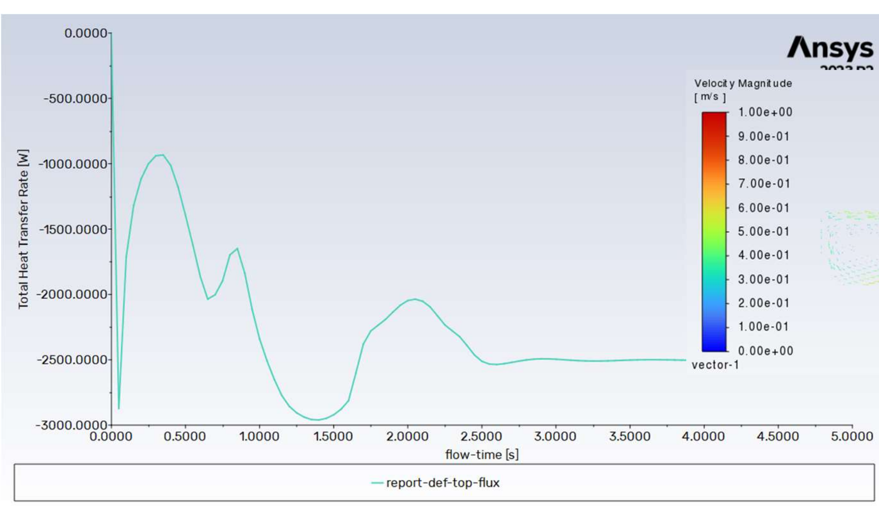


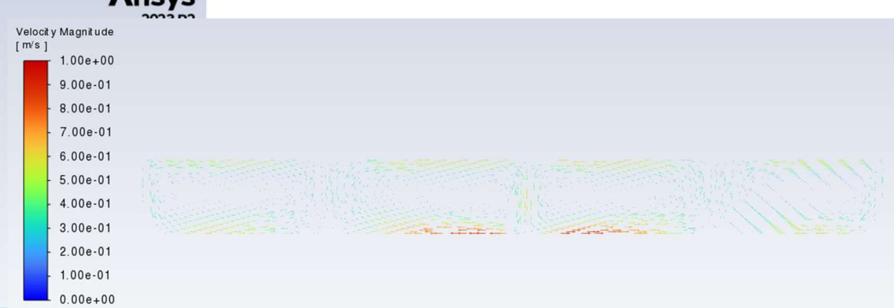
Figure 4: k1 = 0.0121

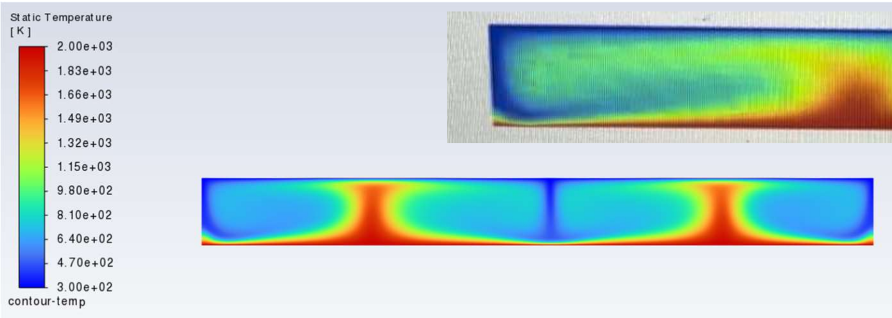
All respective ansys plots











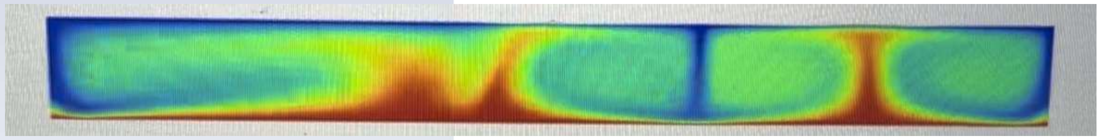
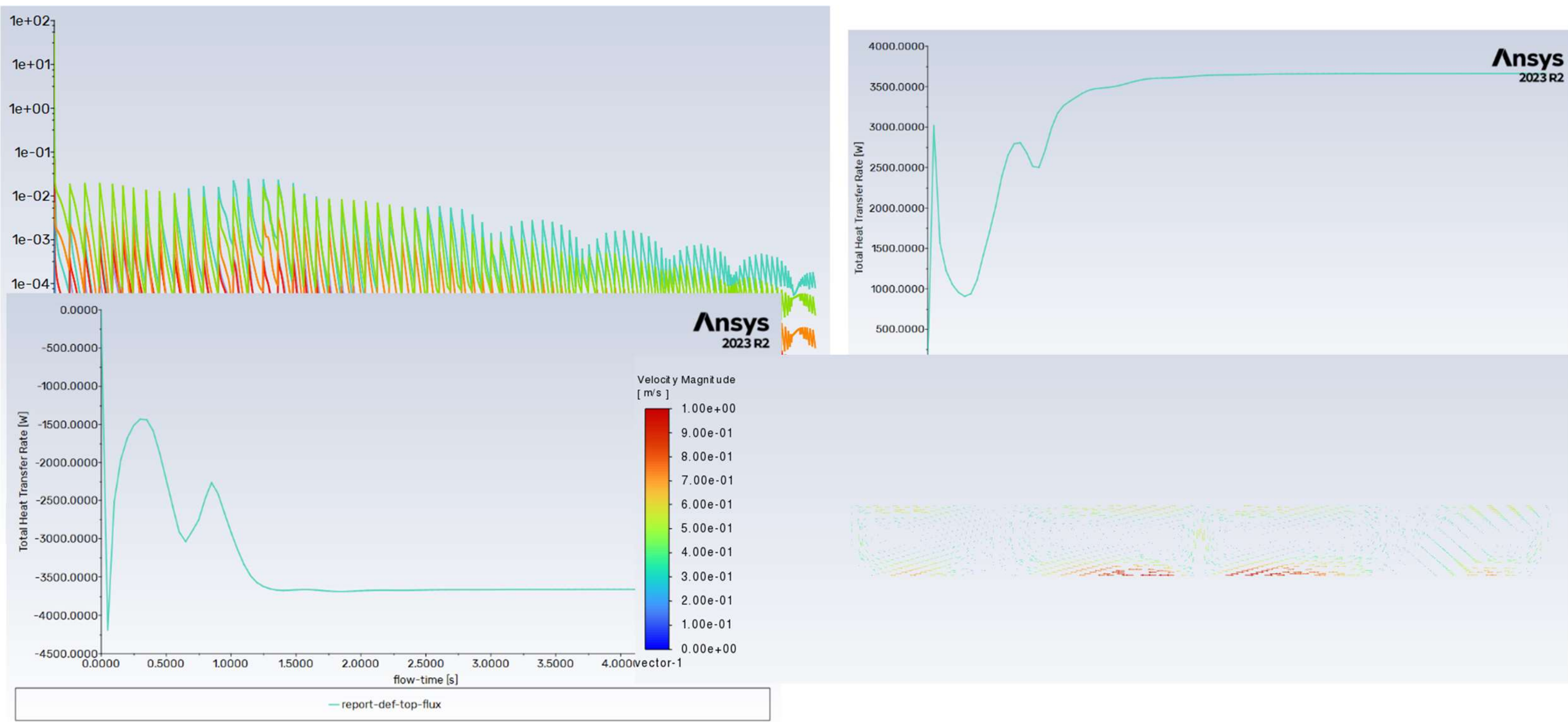
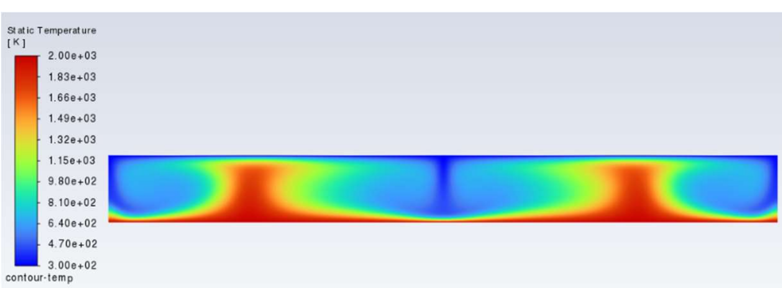


Figure 5: k2 = 0.0242

All respective ansys plots





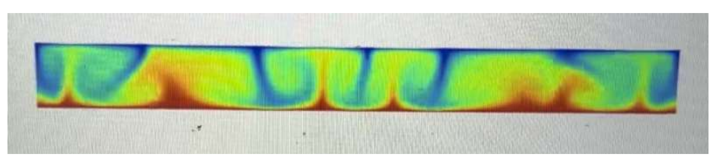


Figure 6: k3 = 0.0484 ; All respective ansys plots

a2. Output the solution data. Use the **`dataConversion.m`** script to create a plot with the heat fluxes through the bottom and top boundaries.

(Figures)

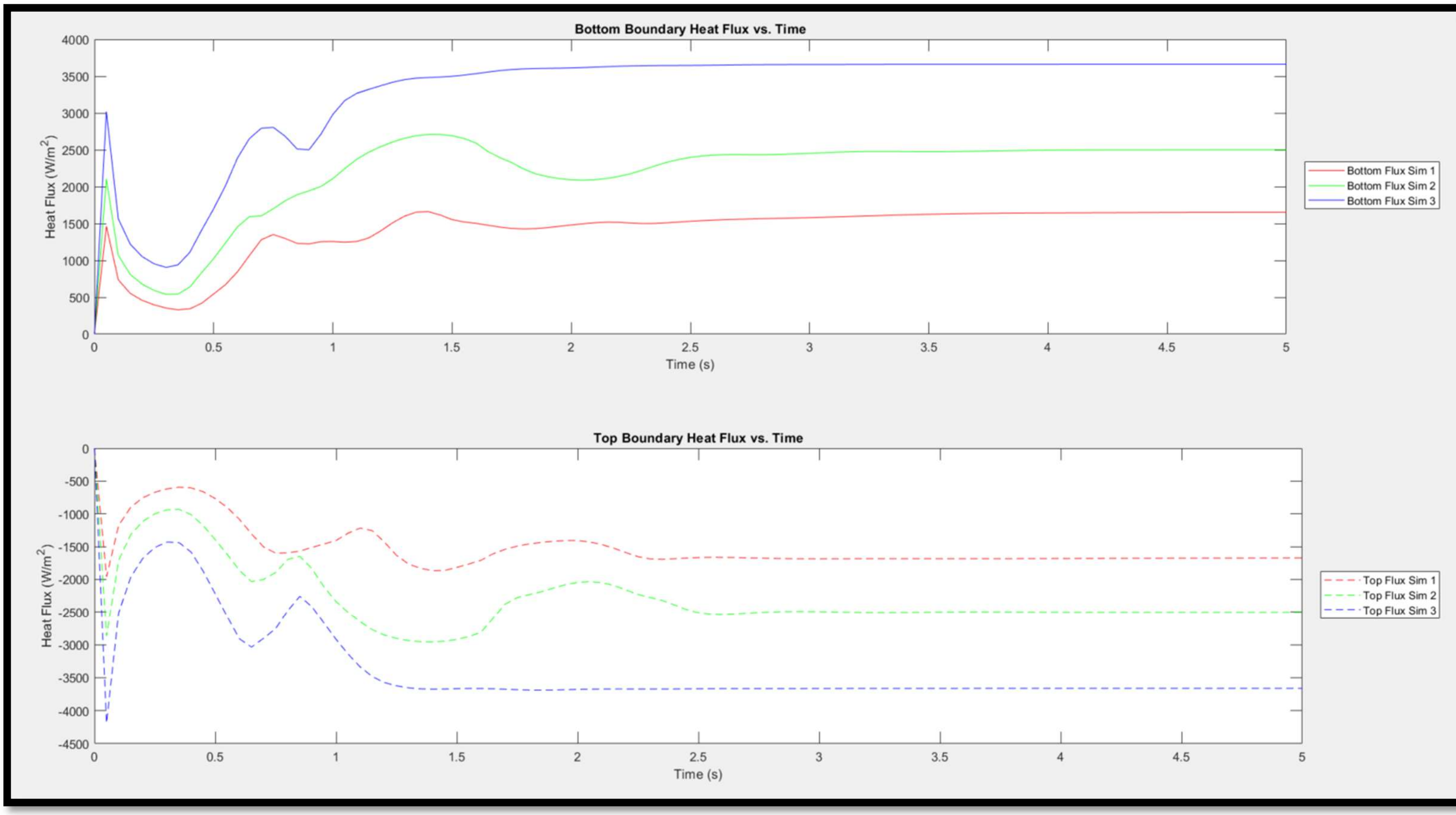


Figure 7: Heat Flux Plots

Heat Flux Scripts (dataConversion.m)

```
clear; close all; clc;

% Define the filenames of the simulation data
txt_N = {'k1report-file-heat-transfer.txt', 'k2report-file-heat-transfer.txt',
'k3report-file-heat-transfer.txt'};

% Create a figure for the combined line plot
figure;

% Colors for each plot for better visibility
colors = ['r', 'g', 'b']; % Assign a color for each file

% Iterate through each simulation
for i = 1:length(txt_N)
    data = importdata(txt_N{i}, ' ', 3); % Load data, skip the first 3 header lines
```

```
    % Time and flux data
    time = data.data(:, 2); % Flow-time
    bottomFlux = data.data(:, 3); % Bottom boundary heat flux
    topFlux = data.data(:, 4); % Top boundary heat flux

    % Plot bottom and top flux vs. time on the same graph with different colors
    subplot(2, 1, 1); % Bottom flux plot
    plot(time, bottomFlux, strcat(colors(i), '-'), 'DisplayName', ['Bottom Flux Sim
', num2str(i)]);
    hold on;

    subplot(2, 1, 2); % Top flux plot
    plot(time, topFlux, strcat(colors(i), '--'), 'DisplayName', ['Top Flux Sim ',
num2str(i)]);
    hold on;
end

% Formatting the bottom flux plot
subplot(2, 1, 1);
title('Bottom Boundary Heat Flux vs. Time');
xlabel('Time (s)');
ylabel('Heat Flux (W/m^2)');
legend('Location', 'eastoutside', 'Orientation', 'vertical');

% Formatting the top flux plot
subplot(2, 1, 2);
title('Top Boundary Heat Flux vs. Time');
xlabel('Time (s)');
ylabel('Heat Flux (W/m^2)');
legend('Location', 'eastoutside', 'Orientation', 'vertical');
```

b. Comment on how changing the fluid diffusivity impacts the heat transfer and temperature fields.

(Comment)

The impact of fluid conductivity on heat transfer is significant, as fluid conductivity directly affects the heat transfer rate between surfaces and the fluid. When a fluid has higher thermal conductivity, it can transfer heat more efficiently. This is essential in many engineering applications, such as cooling systems and heat exchangers, where the goal is often to maximize the rate of heat transfer. Fluids with higher thermal conductivity can carry away more heat from a hot surface or deliver heat more effectively to a cold surface. Conversely, fluids with low thermal conductivity are less effective at transferring heat, which can be desirable in applications where insulation is needed.