PLEASE READ THIS PAGE FIRST

A statement of collaboration is mandatory and must be placed in the first page of the report. If no collaboration occurred, simply state "**No collaboration**". This certifies that the person submitting the report ***has not helped anyone or received help from anyone*** in the process of completing the assignment. Collaboration is allowed subject to the following rules.

(1) For each assignment, each person can have maximum of one collaborator. Be aware that a collaborator's collaborator counts as a collaborator. For example, if Alice collaborates with Bob and Bob collaborates with Charles, Charles counts as a collaborator of Alice. All three violate the rule. In other words, collaboration can only be carried out within an isolated "team of two". Please talk to a potential collaborator to ensure that this rule is not violated before establishing the collaboration.

(2) In a legitimate collaboration, each individual must make a meaningful contribution to the collaborative effort. Taking the solution or code from another student without making a reciprocal contribution to it is not allowed. To certify that a collaboration is legitimate, the submitter's contribution to the collaborative effort must be documented in the statement of collaboration.

(3) Even with collaboration, the write-up of the report must be done independently. Verbatim copying of a collaborator's report, in whole or in part, is not allowed.

The statement of collaboration

If no collaboration occurred, simply state "**No collaboration**". If collaboration occurred, provide the name of collaborator (only one allowed), a list of the task(s) on which collaboration occurred, and a description of the contribution by the submitter to the collaborative effort. Example:

Name of collaborator: Joe Smith	
Task(s), specific detail	Contribution to collaborative effort
Problem 1	Worked on Matlab coding together
Problem 3	Worked with collaborator on completing the analytic solution

The statement should be placed in the first page of the report.

A submission without the statement of collaboration will be rejected.

MAE/MSE 502, Fall 2022 Homework #1 (1 point ≈ 1% of the total score for the course.)

A statement of collaboration is required. Please include computer codes (if any) in your work.

Problem 1 (5 points)
(a) For $u(x, t)$ defined on the domain of $0 \le x \le 1$ and $t \ge 0$, solve the PDE

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$$

with the boundary conditions,

(i) $u_x(0,t) = 0$, (ii) $u_x(1,t) = 0$, (iii) $u(x,0) = P(x)$,

where

$$P(x) = 4\,x^3 - 3x^4 + \frac{[1-\cos(2\pi x)]^9}{512} \quad .$$

Plot the solution, $u(x,t)$, as a function of x at t = 0, 0.004, 0.016, and 0.2. Please collect all four curves in a single plot.

(b) Repeat (a) but now solve the system with the first boundary condition changed to

(i) $u(0,t) = 0$.

The 2nd and 3rd boundary conditions remain the same as in Part (a). Plot the solution, $u(x,t)$, as a function of x at t = 0, 0.004, 0.016, and 0.2. Please collect all four curves in a single plot.

(c) Based on the solution for Part (b), plot the "temperature at the right end point" $u(1,t)$ as a function of t over the range of $0 \le t \le 0.1$. Please use a sufficiently small increment of t to ensure that the plot is smooth. The recommended increment is $\Delta t = 0.001$.

Note for Problem 1:

(i) We expect the solution to be expressed as an infinite series. A truncation of the infinite series is needed to numerically compute the series in order to plot the solution. It is your job to determine the appropriate number of terms to keep. As a useful measure, the solution at $t = 0$ should match the given initial state in the 3rd boundary condition. If they do not match, either the solution is wrong or more terms need to be retained in the series. This remark applies to all future homework problems that require the evaluation of an infinite series.

(ii) For the evaluation of the expansion coefficients in the infinite series, there is no need to carry out the integrations analytically. (If you wish to do so, please feel free to use an online integrator or similar software such as Wolfram Alpha. No need to do it by hand.) It is perfectly acceptable (actually, **recommended** for this problem) to evaluate the integrals numerically using Matlab. See Additional Note in the last page for an example of using Matlab to numerically evaluate an integral.

Problem 2 (2 points)
For $u(x,t)$ defined on the domain of $0 \le x \le 1$ and $t \ge 0$, solve the PDE,

$$\frac{\partial u}{\partial t} = e^{-t}\frac{\partial^2 u}{\partial x^2} + 4\pi^2 e^{-t}\, u \quad ,$$

with the boundary conditions,

(i) $u(0,t) = 0$, (ii) $u(1,t) = 0$, (iii) $u(x,0) = \sin(\pi x) + \sin(2\pi x)$.

In addition to the full solution, please also find the steady solution for this problem. We expect a closed-form exact solution with only a finite number of terms and without any unevaluated integrals. This requirement is for both the full solution and the steady solution. Expect a deduction if the requirement is not satisfied.

Problem 3 (2 points)
For $u(x,t)$ defined on the domain of $0 \le x \le \pi$ and $t \ge 0$, solve the PDE,

$$(1+2t)\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - 2t\, u \quad ,$$

with the boundary conditions,

(i) $u_x(0,t) = 0$, (ii) $u_x(\pi,t) = 0$, (iii) $u(x,0) = 1 + \cos(x)$.

For this problem, we expect a closed-form exact solution with only a finite number of terms and without any unevaluated integrals. Expect a deduction if the requirement is not satisfied.

Problem 4 (1 point)
For the heat transfer problem, the 1-D Heat equation in its dimensional form is

$$\frac{\partial u}{\partial \hat{t}} = K\frac{\partial^2 u}{\partial \hat{x}^2}, \quad 0 \le \hat{x} \le L \quad (L \text{ is the length of the "metal rod", in meters) and } \hat{t} \ge 0 , \quad (1)$$

where $\hat{t}$ and $\hat{x}$ are time in seconds and distance in meters, K is thermal diffusivity in m^2/s, and u is temperature. In our class, we usually consider the non-dimensionalized version of (1),

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad 0 \le x \le 1 \;\text{ and }\; t \ge 0 \quad , \qquad (2)$$

where x is related to $\hat{x}$ by $\hat{x} = L\,x$. In (2), time is also non-dimensionalized by $\hat{t} = T\,t$, where T is a certain dimensional time scale, and so on.

(a) In order to claim that the non-dimensionalized system, (1), is equivalent to its dimensional counterpart, (2), the three parameters K, L, and T must satisfy a unique relation. First, find out what this relation is. (For the discussion in part (b) & (c), it is useful to write the relation as $T = f(K, L)$.)

(b) Suppose that the non-dimensionalized Heat equation, (2), is used to model the real world problem of heat transfer along a metal rod that is 1 meter long and made of copper ($K \approx 0.0001$ m^2/s), what would be the actual time, in seconds, that $t = 0.01$ corresponds to in that problem?

(c) Same as (b), but suppose that (2) describes heat transfer along a wooden stick that is 0.3 meter long and made of pine wood ($K \approx 10^{-7}$ m^2/s), what would be the actual time, in seconds, that $t = 0.01$ corresponds to? (We consider $t = 0.01$ because it is about the time when a significant redistribution of temperature begins to take place in the scenario described in Part (d).)

(d) Are the time scales you obtained in (b) and (c) consistent with daily experience? For instance, one can use a long wooden spoon to continuously stir a boiling pot of soup without getting one's hand burned. In contrast, the same practice would make one very uncomfortable if the spoon is made entirely of copper. Note that the time for cooking a pot of soup is about 10 minutes. The length of a big wooden spoon is about a foot, or 0.3 meter.

Additional Note: Using Matlab to numerically evaluate an integral

One of the simplest Matlab functions for this purpose is **trapz**. It uses the trapezoidal method to evaluate an integral. For example, to numerically evaluate

$$I = \int_0^1 \sin(x)dx$$

with $\Delta x = 0.01$, we first construct the discretized arrays of the coordinate points and the values of the integrand at those points. We then call **trapz** with those two arrays as the input to complete the integration. The Matlab code is very simple:

```
x = [0:0.01:1];
y = sin(x);
Integ = trapz(x,y)
```

One can readily verify the outcome with the analytic result of $I = 1 - \cos(1)$.

When using "trapz" or similar functions for numerical integration, it is important to use a sufficient grid resolution to avoid "aliasing errors". It requires a minimum of 5 points (10 points are recommended) to resolve one cycle of a sinusoidal wave. In the example above, 100 grid points are used to resolve less than 1/6 of a cycle, which is sufficient. If the integrand in the above example is changed to y = sin(500 x), the result of numerical integration will be entirely wrong. The remedy would be to also increase the grid resolution. For example, change the grid to x = [0:0.0001:1].