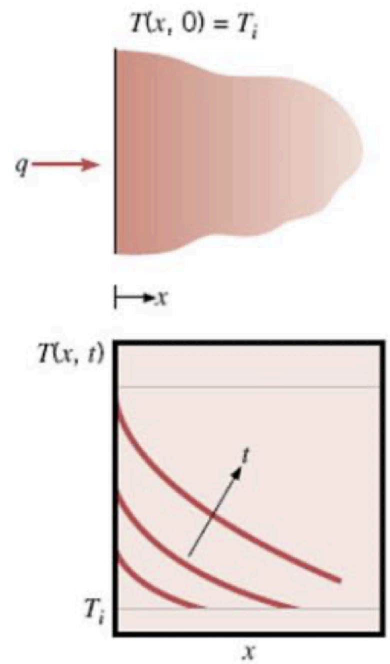
Assignment 3: Mechanical Engineering Case Study – Transient Conduction

Step 1: Problem Identification and Statement

The purpose of this assignment is to calculate the transient temperature distribution within a semi-infinite solid whose surface is subjected to a constant heat flux. More precisely, we need to estimate how the temperature inside the material varies with distance from the heated surface and time, using an analytical solution that includes exponential and error-function components.



[Incropera and DeWitt, Fundamentals of Heat and Mass Transfer, Wiley]

Step 2: Gathering Information and Input/Output Description

Relevant Information:

In heat transfer, Transient conduction refers to instances in which the temperature within a body varies over time. This assignment focusses on transient heat conduction in a semi-infinite solid with a constant surface heat flux q. Under these conditions, the temperature distribution T(x,t) within the solid evolves over both distance (x) from the heated surface and time (t). The primary physical properties governing the problem are:

- ***Initial temperature*** *Ti of the solid (at t=0).*
- ***Thermal conductivity*** *k,* ***density*** *ρ, and* ***specific heat capacity*** *c, which combine to give us the* ***thermal diffusivity :*** *α = k/(ρc)***.**
- ***Constant heat flux*** *q applied at the surface (x=0).*
- *The temperature in the solid, T, at any point (x,t) is given by:*

$$T(x,t) = T_i + \frac{2q\sqrt{\alpha t/\pi}}{k}\exp\left(\frac{-x^2}{4\alpha t}\right) - \frac{qx}{k}\left(1 - \operatorname{erf}\left(\frac{x}{2\sqrt{\alpha t}}\right)\right)$$

Input/output Description (I/O Diagram) :

Input:

1.User input

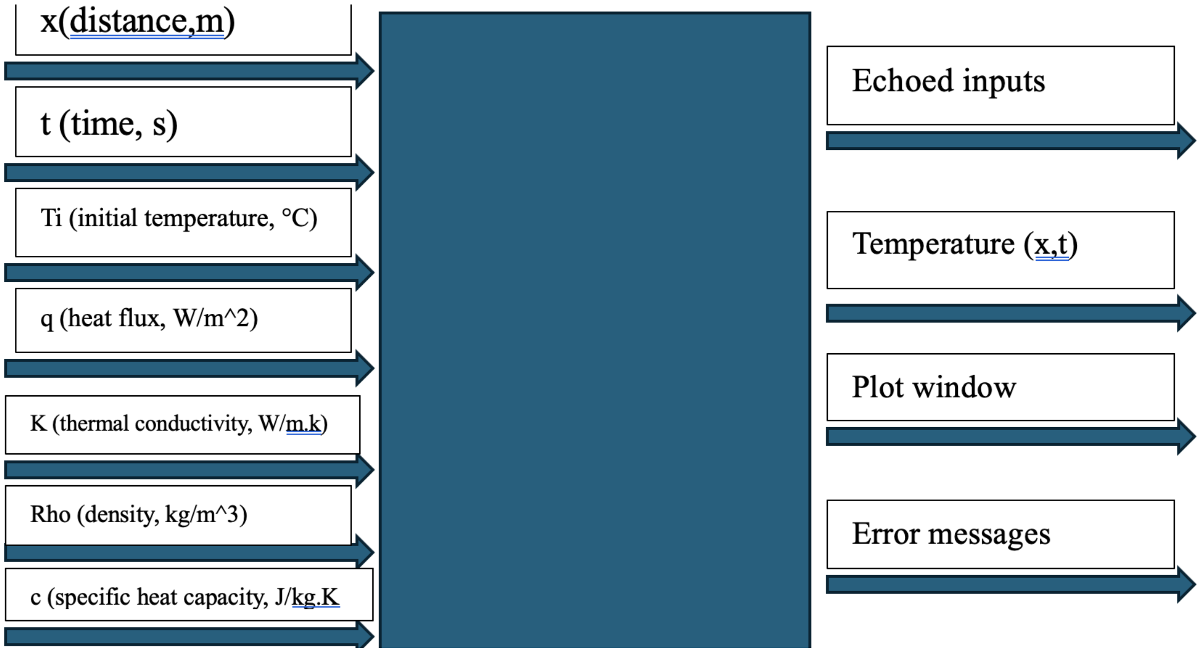
a) x (distance, m)
b) t (time, s)
c) Ti (initial temperature, °C)
d) q (heat flux, W/m²)
e) k (thermal conductivity, W/m·K)
f) ρ (density, kg/m³)
g) c (specific heat capacity, J/kg·K)

Output:

1. Echoed Inputs
2. The temperature T(x,t)
3. Plot Window (T vs. x for t=200,400,600,800 s)

4. Error messages : displaying an error message via error function for invalid input

I/O Diagram:


x(distance,m)
t (time, s)
Ti (initial temperature, °C)
q (heat flux, W/m^2)
K (thermal conductivity, W/m.k)
Rho (density, kg/m^3)
c (specific heat capacity, J/kg.K
Echoed inputs
Temperature (x,t)
Plot window
Error messages

Step 3: Test Cases and algorithm

In this section, we show how to check the algorithm logic and results using manual computations rather than code execution, by having the user inputs and expecting the output.

Test Case	**User Inputs**	**Expected Output**
Case 1/a)	x = 0.02, t = 200, Ti = 20, q = 9000, k = 17.4, ρ = 7900, c = 526	T $\approx$ 28.54
Case 2/b)	x = 0.05, t = 200, Ti = 20, q = 9000, k = 17.4, ρ = 7900, c = 526	T $\approx$ 22.44
Case 3/c)	x = 0.05, t = 800, Ti = 20, q = 9000, k = 17.4, ρ = 7900, c = 526	T $\approx$ 34.05
Case 4/ Handling Invalid Input	User types invalid inputs (., t=0, x=-0.5, letters...).	the program rejects the input and asks for valid input(x must be $\geq$ 0; t, k, ρ, and c must be $\succ$ 0

Algorithm Design:

<u>Pseudo code</u>

BEGIN
// 1. Validate user inputs
REPEAT
PROMPT "Enter x (distance ≥ 0):"
UNTIL x ≥ 0

REPEAT
PROMPT "Enter t (time > 0):"
UNTIL t > 0

PROMPT "Enter Ti (initial temperature):"

REPEAT
PROMPT "Enter q (heat flux > 0):"
UNTIL q > 0

REPEAT
PROMPT "Enter k (thermal conductivity > 0):"
UNTIL k > 0

REPEAT
PROMPT "Enter ρ (density > 0):"
UNTIL ρ > 0

REPEAT
PROMPT "Enter c (specific heat capacity > 0):"
UNTIL c > 0

// 2. Compute thermal diffusivity
α ← k ÷ (ρ × c)

// 3. Compute temperature using the analytical solution
term1 ← (2 × q × √(α × t / π)) ÷ k × EXP(– x² ÷ (4 × α × t))
term2 ← (q × x ÷ k) × [1 – ERF(x ÷ (2 × √(α × t)))]
T ← Ti + term1 – term2

// 4. Display results
DISPLAY all input values with labels
DISPLAY "Computed temperature T =" T

// 5. Plot T vs. x for multiple time values
x_values ← generate evenly spaced points from 0 to 0.1
t_values ← {200, 400, 600, 800}
INITIALIZE new plot
FOR each t_val in t_values DO
FOR each x_i in x_values DO
compute T_i using the same formula
END FOR
ADD curve (x_values, T_values) to plot labeled "t = t_val s"
END FOR
LABEL axes and legend, ENABLE grid

END

Step 4: Code or implementation

```
% Step 1: User Input & handling invalid input
x = input('Enter distance from surface x [m ≥ 0]: ');
while x < 0
   x = input('Invalid. x must be ≥ 0. Enter distance x: ');
end
```

```
t = input('Enter time t [s > 0]: ');
while t <= 0
   t = input('Invalid. t must be > 0. Enter time t: ');
end
```

```
Ti = input('Enter initial temperature Ti [°C]: ');
```

```
q = input('Enter heat flux q [W/m^2 > 0]: ');
while q <= 0
   q = input('Invalid. q must be > 0. Enter heat flux q: ');
end
```

```
k = input('Enter thermal conductivity k [W/(m·K) > 0]: ');
while k <= 0
   k = input('Invalid. k must be > 0. Enter thermal conductivity k: ');
end
```

```
rho = input('Enter density rho [kg/m^3 > 0]: ');
while rho <= 0
   rho = input('Invalid. rho must be > 0. Enter density rho: ');
end
```

```
c = input('Enter specific heat capacity c [J/(kg·K) > 0]: ');
while c <= 0
c = input('Invalid. c must be > 0. Enter specific heat capacity c: ');
end
```

```
% Step 2: Calculate Temperature
alpha = k / (rho * c);  % Thermal diffusivity
term1 = (2 * q * sqrt(alpha * t / pi)) / k * exp(-x^2 / (4 * alpha *
t));
term2 = (q * x / k) * (1 - erf(x / (2 * sqrt(alpha * t))));
T = Ti + term1 - term2;
```

```
% Step 3: show results
fprintf('\n--- Input Parameters ---\n');
fprintf('  x    = %.4f m\n', x);
fprintf('  t    = %.2f s\n', t);
fprintf('  Ti   = %.2f °C\n', Ti);
fprintf('  q    = %.2f W/m^2\n', q);
fprintf('  k    = %.2f W/(m·K)\n', k);
fprintf('  rho  = %.2f kg/m^3\n', rho);
fprintf('  c    = %.2f J/(kg·K)\n', c);
```

```
fprintf('\n--- Computed Temperature ---\n');
fprintf('At x = %.4f m and t = %.2f s, T = %.2f °C\n', x, t, T);
```

```
% Step 4: Plot Temperature graphs
x_vals = linspace(0, 0.1, 100);
t_vals = [200, 400, 600, 800];
```

```
figure; hold on;
for tt = t_vals
   term1_p = (2 * q * sqrt(alpha * tt / pi)) / k .* exp(-x_vals.^2 ./ (4
* alpha * tt));
   term2_p = (q .* x_vals / k) .* (1 - erf(x_vals ./ (2 * sqrt(alpha *
tt))));
   T_p     = Ti + term1_p - term2_p;
   plot(x_vals, T_p, 'DisplayName', sprintf('t = %d s', tt));
end
xlabel('Distance x (m)');
ylabel('Temperature T (°C)');
title('Temperature Distribution vs. Distance at Different Times');
legend('Location', 'northeast');
grid on;
hold off;
```

Step 5: Test and Verification

To test if the program is successful and working, we will run the program and try each test case

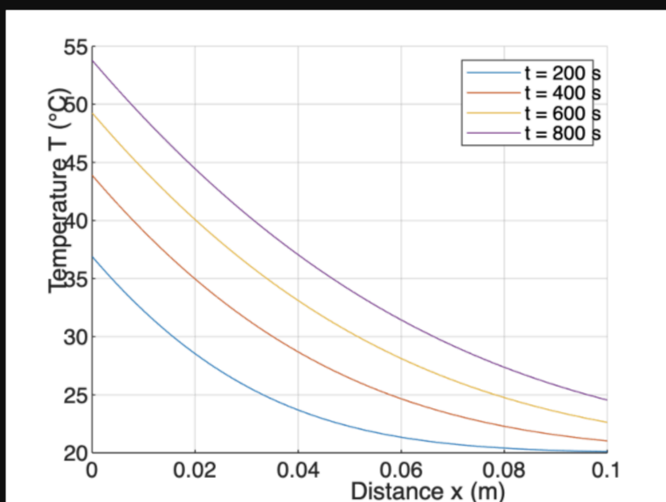
Test Case 1: a

Command Window

New to MATLAB? See resources for Getting Started.

```
Enter distance from surface x [m ≥ 0]:
0.02
Enter time t [s > 0]:
200
Enter initial temperature Ti [°C]:
20
Enter heat flux q [W/m^2 > 0]:
9000
Enter thermal conductivity k [W/(m·K) > 0]:
17.4
Enter density rho [kg/m^3 > 0]:
7900
Enter specific heat capacity c [J/(kg·K) > 0]:
526
>>
```

```
--- Input Parameters ---
  x     = 0.0200 m
  t     = 200.00 s
  Ti    = 20.00 °C
  q     = 9000.00 W/m^2
  k     = 17.40 W/(m·K)
  rho   = 7900.00 kg/m^3
  c     = 526.00 J/(kg·K)
--- Computed Temperature ---
At x = 0.0200 m and t = 200.00 s, T = 28.52 °C
```



No issues

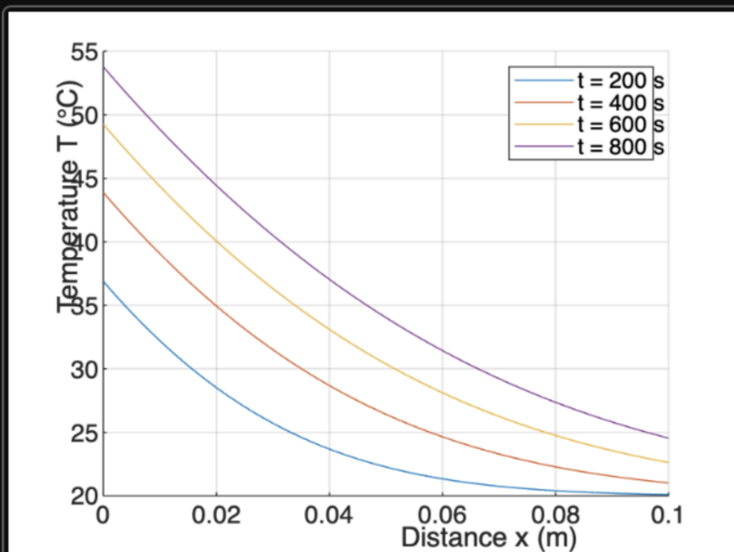
Test Case 2: b

Command Window

New to MATLAB? See resources for Getting Started.

```
Enter distance from surface x [m ≥ 0]:
0.05
Enter time t [s > 0]:
200
Enter initial temperature Ti [°C]:
20
Enter heat flux q [W/m^2 > 0]:
9000
Enter thermal conductivity k [W/(m·K) > 0]:
17.4
Enter density rho [kg/m^3 > 0]:
7900
Enter specific heat capacity c [J/(kg·K) > 0]:
526
>> |
```

```
--- Input Parameters ---
  x    = 0.0500 m
  t    = 200.00 s
  Ti   = 20.00 °C
  q    = 9000.00 W/m^2
  k    = 17.40 W/(m·K)
  rho  = 7900.00 kg/m^3
  c    = 526.00 J/(kg·K)
--- Computed Temperature ---
At x = 0.0500 m and t = 200.00 s, T = 22.27
```



No issues

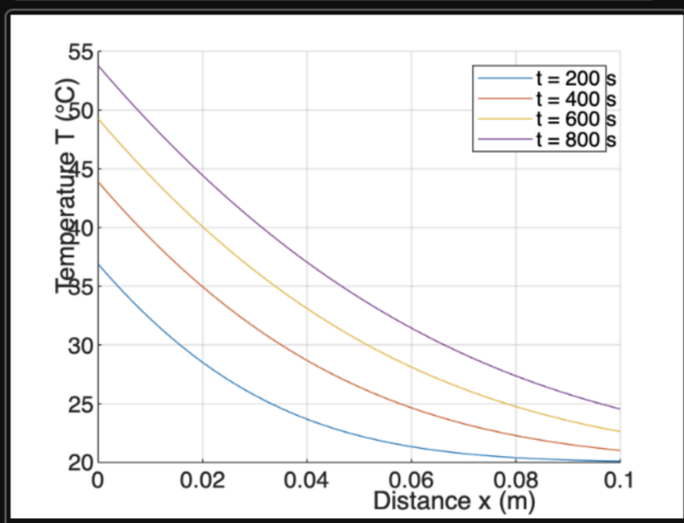
Test Case 3: c

Command Window

New to MATLAB? See resources for Getting Started.

```
Enter distance from surface x [m ≥ 0]:
0.05
Enter time t [s > 0]:
800
Enter initial temperature Ti [°C]:
20
Enter heat flux q [W/m^2 > 0]:
9000
Enter thermal conductivity k [W/(m·K) > 0]:
17.4
Enter density rho [kg/m^3 > 0]:
7900
Enter specific heat capacity c [J/(kg·K) > 0]:
526
>>
```

```
--- Input Parameters ---
  x     = 0.0500 m
  t     = 800.00 s
  Ti    = 20.00 °C
  q     = 9000.00 W/m^2
  k     = 17.40 W/(m·K)
  rho   = 7900.00 kg/m^3
  c     = 526.00 J/(kg·K)
--- Computed Temperature ---
At x = 0.0500 m and t = 800.00 s, T = 34.03
```



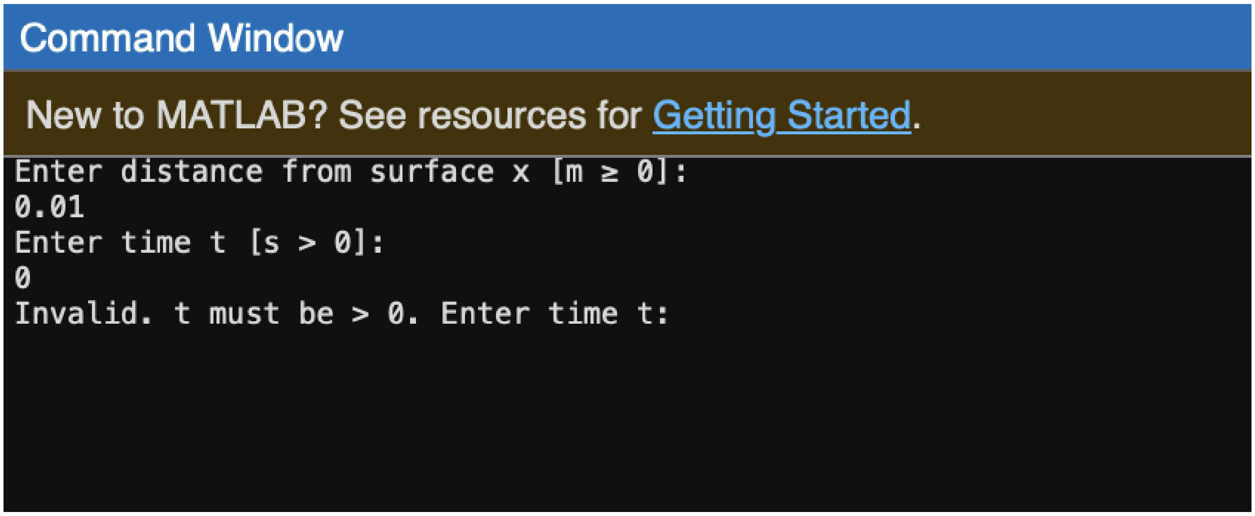
No issues

Test Case 4: handling invalid input

Command Window

New to MATLAB? See resources for Getting Started.

```
Enter distance from surface x [m ≥ 0]:
-0.05
Invalid. x must be ≥ 0. Enter distance x:
```



No issues

Graph verification: The plot shows four different colored curves, illustrating that temperature goes down with increasing distance from the heated surface and goes up as time advances.