

```
import numpy as np
import matplotlib.pyplot as plt

# Parameters
T_h = 373.15 # Hot fluid temperature in Kelvin (100°C)
T_c_in = 298.15 # Inlet temperature of water in Kelvin (25°C)
NTU = 5 # Use a smaller NTU for visualization (actual NTU = 430.57 makes the rise instant)

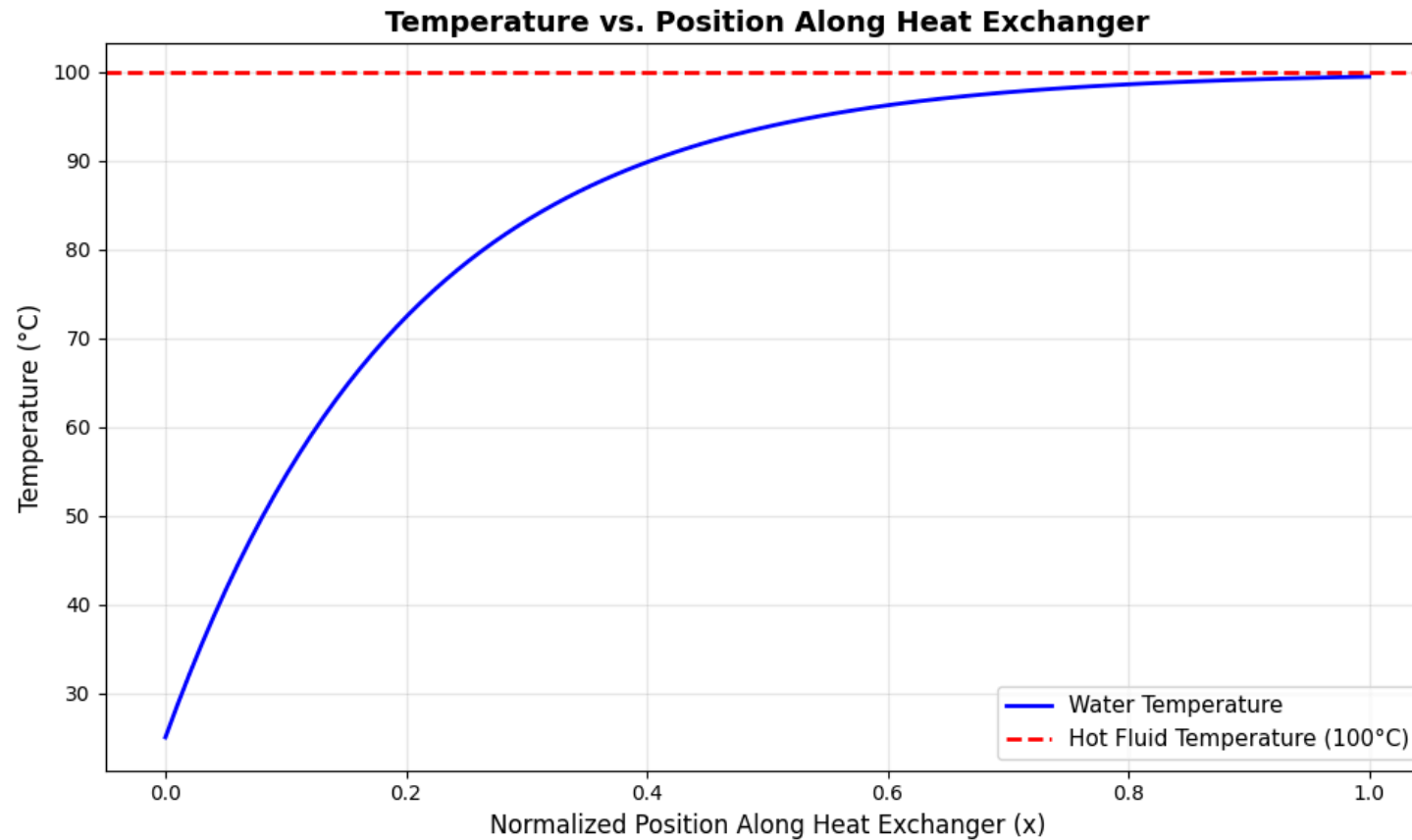
# Normalized position along the exchanger (0 to 1)
x = np.linspace(0, 1, 100)

# Temperature profile
T_c = T_h - (T_h - T_c_in) * np.exp(-NTU * x)
T_c_celsius = T_c - 273.15 # Convert to Celsius for plotting

# Plotting
plt.figure(figsize=(10, 6))
plt.plot(x, T_c_celsius, label='Water Temperature', color='blue', linewidth=2)
plt.axhline(y=100, color='red', linestyle='--', label='Hot Fluid Temperature (100°C)', linewidth=2)
plt.xlabel('Normalized Position Along Heat Exchanger (x)', fontsize=12)
plt.ylabel('Temperature (°C)', fontsize=12)
plt.title('Temperature vs. Position Along Heat Exchanger', fontsize=14, fontweight='bold')
plt.grid(True, alpha=0.3)
plt.legend(fontsize=11)
plt.tight_layout()

# Display the plot (Google Colab will show it automatically)
plt.show()

# Optional: Print some key values
print(f"Initial water temperature: {T_c_in - 273.15:.1f}°C")
print(f"Hot fluid temperature: {T_h - 273.15:.1f}°C")
print(f"Final water temperature: {T_c_celsius[-1]:.1f}°C")
print(f"NTU value used: {NTU}")
```



Initial water temperature: 25.0°C
Hot fluid temperature: 100.0°C
Final water temperature: 99.5°C
NTU value used: 5

Start coding or [generate](#) with AI.

```
import numpy as np
import matplotlib.pyplot as plt

# Define temperature data
T_water_in = 25 # Inlet temperature of water (°C)
T_water_out = 99.5 # Outlet temperature of water (°C)
T_hot_in = 100 # Inlet temperature of hot fluid (°C)
T_hot_out = 100 # Outlet temperature of hot fluid (°C)

# Normalized position (0 to 1)
x = np.linspace(0, 1, 100)
```

```

# Temperature profiles (linear approximation)
T_water = T_water_in + (T_water_out - T_water_in) * x # T_water(x) = 25 + 275x
T_air = T_air_out + (T_air_in - T_air_out) * (1 - x) # T_air(x) = 200 + 300(1-x)

# Plotting
plt.figure(figsize=(10, 6))
plt.plot(x, T_water, label='Water Temperature', color='blue', linewidth=2.5, marker='o', markersize=3, markevery=10)
plt.plot(x, T_air, label='Air Temperature', color='red', linewidth=2.5, marker='s', markersize=3, markevery=10)

plt.xlabel('Normalized Position Along Heat Exchanger (x)', fontsize=12)
plt.ylabel('Temperature (°C)', fontsize=12)
plt.title('Temperature vs. Position Along Heat Exchanger\n(Counter-flow Configuration)', fontsize=14, fontweight='bold')
plt.legend(fontsize=11, loc='center right')
plt.grid(True, alpha=0.3)
plt.tight_layout()

# Add annotations for inlet and outlet temperatures
plt.annotate(f'Water In: {T_water_in}°C', xy=(0, T_water_in), xytext=(0.1, T_water_in + 30),
            arrowprops=dict(arrowstyle='->', color='blue', alpha=0.7), fontsize=10, color='blue')
plt.annotate(f'Water Out: {T_water_out}°C', xy=(1, T_water_out), xytext=(0.85, T_water_out + 30),
            arrowprops=dict(arrowstyle='->', color='blue', alpha=0.7), fontsize=10, color='blue')
plt.annotate(f'Air In: {T_air_in}°C', xy=(0, T_air_in), xytext=(0.1, T_air_in + 30),
            arrowprops=dict(arrowstyle='->', color='red', alpha=0.7), fontsize=10, color='red')
plt.annotate(f'Air Out: {T_air_out}°C', xy=(1, T_air_out), xytext=(0.85, T_air_out - 40),
            arrowprops=dict(arrowstyle='->', color='red', alpha=0.7), fontsize=10, color='red')

# Set y-axis limits for better visualization
plt.ylim(0, 550)

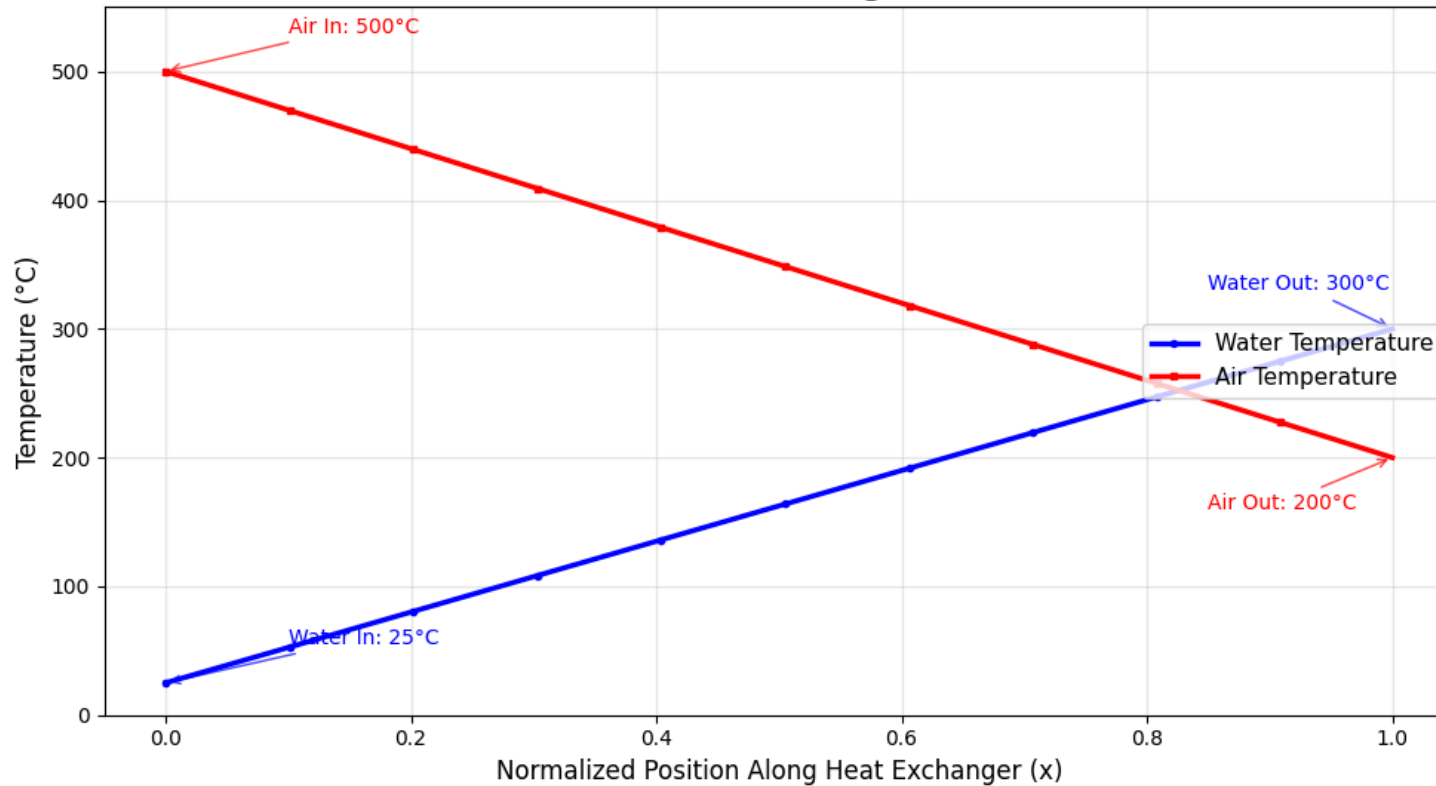
# Display the plot
plt.show()

# Print summary information
print("="*50)
print("HEAT EXCHANGER ANALYSIS SUMMARY")
print("="*50)
print(f"Water: {T_water_in}°C → {T_water_out}°C (ΔT = +{T_water_out - T_water_in}°C)")
print(f"Air: {T_air_in}°C → {T_air_out}°C (ΔT = -{T_air_in - T_air_out}°C)")
print(f"Heat gained by water = Heat lost by air")
print(f"Counter-flow configuration allows maximum heat transfer")
print("="*50)

```



Temperature vs. Position Along Heat Exchanger (Counter-flow Configuration)



```
=====
HEAT EXCHANGER ANALYSIS SUMMARY
=====
Water: 25°C → 300°C (ΔT = +275°C)
Air: 500°C → 200°C (ΔT = -300°C)
Heat gained by water = Heat lost by air
Counter-flow configuration allows maximum heat transfer
=====
```

```
import matplotlib.pyplot as plt
import matplotlib.patches as patches
import numpy as np

# Create figure with subplots
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(16, 12))

# ===== TOP PLOT: PHYSICAL LAYOUT =====
ax1.set_xlim(0, 16)
ax1.set_ylim(0, 10)
```

```

ax1.set_aspect('equal')
ax1.axis('off')

# Title
ax1.text(8, 9.5, 'Multi-Pass Shell-and-Tube Heat Exchanger',
        ha='center', va='center', fontsize=16, fontweight='bold')
ax1.text(8, 9, 'Physical Configuration and Flow Arrangement',
        ha='center', va='center', fontsize=12, style='italic')

# Shell (outer cylinder)
shell = patches.Rectangle((2, 2), 12, 5, linewidth=3,
                          edgecolor='black', facecolor='lightgray', alpha=0.3)
ax1.add_patch(shell)

# Shell label
ax1.text(8, 7.5, 'SHELL', ha='center', va='center', fontsize=12, fontweight='bold')

# Tube bundle - Multiple passes
tube_colors = ['lightblue', 'lightcoral', 'lightgreen', 'lightyellow']
pass_labels = ['Pass 1', 'Pass 2', 'Pass 3', 'Pass 4']

# Draw tubes for each pass
for i in range(4):
    y_pos = 6 - i * 0.8
    # Tubes in each pass
    for j in range(8):
        x_pos = 3 + j * 1.25
        tube = patches.Circle((x_pos, y_pos), 0.15,
                              facecolor=tube_colors[i], edgecolor='black', linewidth=1)
        ax1.add_patch(tube)

    # Pass label
    ax1.text(1.5, y_pos, pass_labels[i], ha='center', va='center',
            fontsize=10, fontweight='bold')

# Tube sheets
tubesheet1 = patches.Rectangle((2, 2), 0.2, 5, linewidth=2,
                              edgecolor='black', facecolor='darkgray')
ax1.add_patch(tubesheet1)

tubesheet2 = patches.Rectangle((13.8, 2), 0.2, 5, linewidth=2,
                              edgecolor='black', facecolor='darkgray')
ax1.add_patch(tubesheet2)

# Water inlet/outlet headers
header1 = patches.Rectangle((0.5, 2), 1.5, 2.5, linewidth=2,
                          edgecolor='blue', facecolor='lightblue', alpha=0.5)
ax1.add_patch(header1)

header2 = patches.Rectangle((0.5, 4.5), 1.5, 2.5, linewidth=2,
                          edgecolor='blue', facecolor='lightblue', alpha=0.5)

```

```

ax1.add_patch(header2)

# Water flow arrows in tubes
# Pass 1: Left to right
for j in range(7):
    x_pos = 3.5 + j * 1.25
    ax1.arrow(x_pos, 6, 0.8, 0, head_width=0.1, head_length=0.1,
              fc='blue', ec='blue', linewidth=2)

# Pass 2: Right to left
for j in range(7):
    x_pos = 10.5 - j * 1.25
    ax1.arrow(x_pos, 5.2, -0.8, 0, head_width=0.1, head_length=0.1,
              fc='blue', ec='blue', linewidth=2)

# Pass 3: Left to right
for j in range(7):
    x_pos = 3.5 + j * 1.25
    ax1.arrow(x_pos, 4.4, 0.8, 0, head_width=0.1, head_length=0.1,
              fc='blue', ec='blue', linewidth=2)

# Pass 4: Right to left
for j in range(7):
    x_pos = 10.5 - j * 1.25
    ax1.arrow(x_pos, 3.6, -0.8, 0, head_width=0.1, head_length=0.1,
              fc='blue', ec='blue', linewidth=2)

# Water inlet/outlet pipes
ax1.arrow(0.2, 3.25, 0.3, 0, head_width=0.15, head_length=0.1,
          fc='blue', ec='blue', linewidth=3)
ax1.text(0.1, 3.25, 'Water In\n25°C', ha='right', va='center',
         fontsize=10, fontweight='bold', color='blue')

ax1.arrow(0.2, 5.75, 0.3, 0, head_width=0.15, head_length=0.1,
          fc='red', ec='red', linewidth=3)
ax1.text(0.1, 5.75, 'Water Out\n300°C', ha='right', va='center',
         fontsize=10, fontweight='bold', color='red')

# Hot air flow (cross-flow)
# Air inlet
ax1.arrow(8, 8.5, 0, -1, head_width=0.3, head_length=0.15,
          fc='orange', ec='orange', linewidth=4)
ax1.text(8, 8.8, 'Hot Air In\n500°C', ha='center', va='bottom',
         fontsize=12, fontweight='bold', color='orange')

# Air outlet
ax1.arrow(8, 1.5, 0, -1, head_width=0.3, head_length=0.15,
          fc='brown', ec='brown', linewidth=4)
ax1.text(8, 1.2, 'Hot Air Out\n200°C', ha='center', va='top',
         fontsize=12, fontweight='bold', color='brown')

```

```

# Cross-flow arrows inside shell
for i in range(5):
    for j in range(3):
        x_pos = 4 + i * 2
        y_pos = 6.5 - j * 1.5
        ax1.arrow(x_pos, y_pos, 0, -0.8, head_width=0.15, head_length=0.1,
                  fc='orange', ec='orange', alpha=0.6, linewidth=1)

# Connection lines between passes
# Pass 1 to 2
ax1.plot([13.5, 15, 15, 13.5], [6, 6, 5.2, 5.2], 'b-', linewidth=3)
ax1.arrow(13.3, 5.2, -0.3, 0, head_width=0.1, head_length=0.1,
          fc='blue', ec='blue', linewidth=2)

# Pass 2 to 3
ax1.plot([2.5, 1, 1, 2.5], [5.2, 5.2, 4.4, 4.4], 'b-', linewidth=3)
ax1.arrow(2.7, 4.4, 0.3, 0, head_width=0.1, head_length=0.1,
          fc='blue', ec='blue', linewidth=2)

# Pass 3 to 4
ax1.plot([13.5, 15, 15, 13.5], [4.4, 4.4, 3.6, 3.6], 'b-', linewidth=3)
ax1.arrow(13.3, 3.6, -0.3, 0, head_width=0.1, head_length=0.1,
          fc='blue', ec='blue', linewidth=2)

# ===== BOTTOM PLOT: FLOW SCHEMATIC =====
ax2.set_xlim(0, 16)
ax2.set_ylim(0, 8)
ax2.axis('off')

# Title
ax2.text(8, 7.5, 'Flow Schematic and Temperature Profile',
        ha='center', va='center', fontsize=16, fontweight='bold')

# Draw simplified schematic
# Shell
shell_schematic = patches.Rectangle((2, 2), 12, 3, linewidth=3,
                                    edgecolor='black', facecolor='lightgray', alpha=0.3)
ax2.add_patch(shell_schematic)

# Tube passes representation
tube_passes = patches.Rectangle((3, 2.5), 10, 2, linewidth=2,
                                edgecolor='blue', facecolor='lightblue', alpha=0.5)
ax2.add_patch(tube_passes)

# Flow direction indicators
ax2.text(4, 4.5, '→', fontsize=20, color='blue', fontweight='bold')
ax2.text(6, 4.5, '→', fontsize=20, color='blue', fontweight='bold')
ax2.text(8, 4.5, '→', fontsize=20, color='blue', fontweight='bold')
ax2.text(10, 4.5, '→', fontsize=20, color='blue', fontweight='bold')

ax2.text(4, 3.5, '←', fontsize=20, color='blue', fontweight='bold')

```

```

ax2.text(6, 3.5, '←', fontsize=20, color='blue', fontweight='bold')
ax2.text(8, 3.5, '←', fontsize=20, color='blue', fontweight='bold')
ax2.text(10, 3.5, '←', fontsize=20, color='blue', fontweight='bold')

# Temperature annotations
ax2.text(1, 4.5, 'T1 = 25°C', fontsize=12, color='blue', fontweight='bold')
ax2.text(1, 3.5, 'T2 = 300°C', fontsize=12, color='red', fontweight='bold')

# Hot air flow
ax2.arrow(8, 6, 0, -0.8, head_width=0.3, head_length=0.15,
          fc='orange', ec='orange', linewidth=4)
ax2.text(8, 6.3, 'Hot Air: 500°C → 200°C', ha='center', va='bottom',
          fontsize=12, fontweight='bold', color='orange')

# Cross-flow indication
for i in range(4):
    x_pos = 4 + i * 2.5
    ax2.arrow(x_pos, 5.5, 0, -0.4, head_width=0.2, head_length=0.1,
              fc='orange', ec='orange', alpha=0.7, linewidth=2)

# System parameters box
param_text = """"SYSTEM PARAMETERS:
• Overall Heat Transfer Coefficient: U = 500 W/m²K
• Heat Transfer Area: A = 1.0 m²
• Water Specific Heat: Cp = 4180 J/kgK
• Flow Configuration: Multi-pass tube side, Cross-flow shell side
• Required Mass Flow Rate: ṁ = 0.0774 kg/s"""""

ax2.text(8, 1.5, param_text, ha='center', va='center', fontsize=10,
          bbox=dict(boxstyle="round,pad=0.5", facecolor='lightyellow', alpha=0.8))

# Add legend
ax2.text(14.5, 6, 'LEGEND:', fontsize=10, fontweight='bold')
ax2.plot([14.5, 15], [5.7, 5.7], 'b-', linewidth=3)
ax2.text(15.1, 5.7, 'Water Flow', fontsize=9, va='center')
ax2.plot([14.5, 15], [5.4, 5.4], color='orange', linewidth=3)
ax2.text(15.1, 5.4, 'Hot Air Flow', fontsize=9, va='center')
ax2.plot([14.5, 15], [5.1, 5.1], 'k-', linewidth=2)
ax2.text(15.1, 5.1, 'Shell/Tubes', fontsize=9, va='center')

plt.tight_layout()
plt.show()

# Print system description
print("="*70)
print("MULTI-PASS SHELL-AND-TUBE HEAT EXCHANGER SYSTEM")
print("="*70)
print("CONFIGURATION:")
print("• Shell Side: Hot air flows perpendicular to tubes (cross-flow)")
print("• Tube Side: Water flows through tubes in multiple passes")
print("• Number of Passes: 4 passes (alternating flow direction)")

```

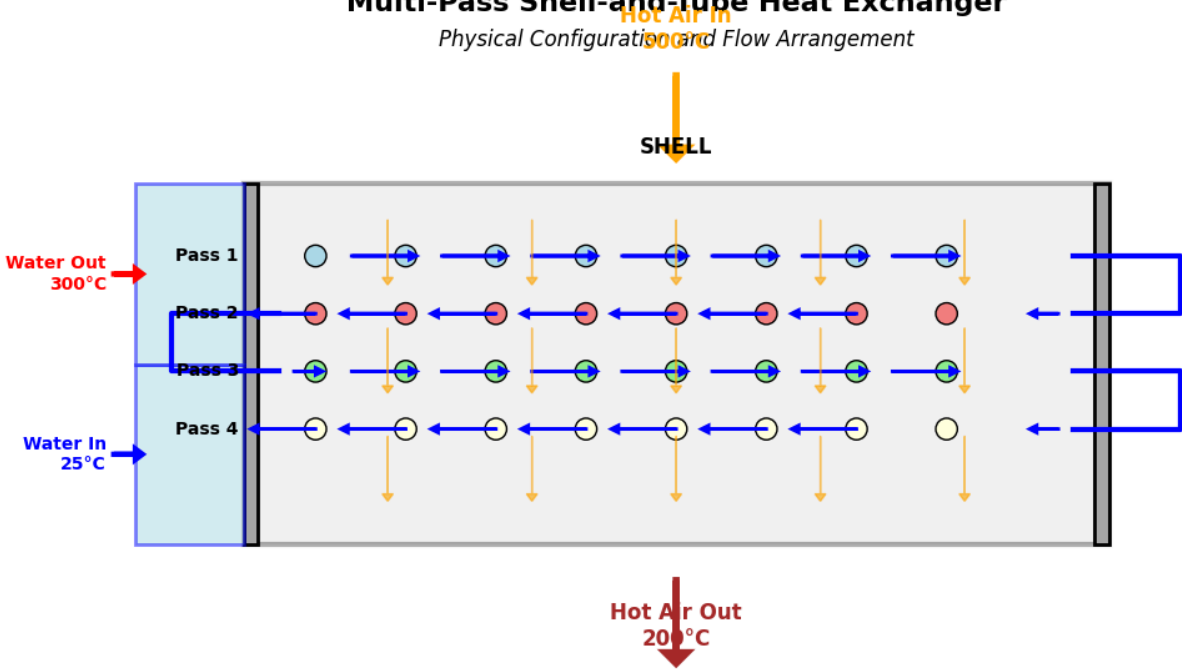


```
print()
print("FLOW ARRANGEMENT:")
print("• Pass 1: Water flows left to right (inlet temperature)")
print("• Pass 2: Water flows right to left (intermediate heating)")
print("• Pass 3: Water flows left to right (continued heating)")
print("• Pass 4: Water flows right to left (outlet temperature)")
print()
print("HEAT TRANSFER MECHANISM:")
print("• Hot air enters at 500°C, exits at 200°C")
print("• Water enters at 25°C, exits at 300°C")
print("• Cross-flow arrangement maximizes heat transfer efficiency")
print("• Multiple passes increase residence time and heat transfer area")
print()
print("DESIGN ADVANTAGES:")
print("• Compact design with high heat transfer coefficient")
print("• Multiple passes increase heat transfer effectiveness")
print("• Cross-flow arrangement provides good temperature driving force")
print("• Easy maintenance access to tube bundle")
print("=*70)
```

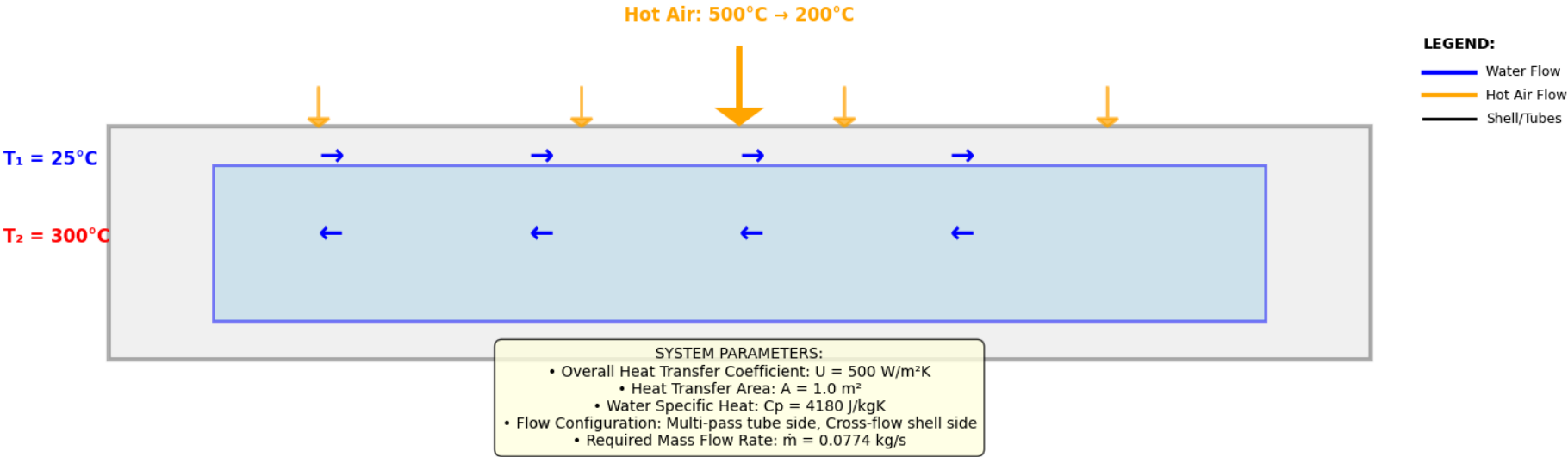


Multi-Pass Shell-and-Tube Heat Exchanger

Physical Configuration and Flow Arrangement



Flow Schematic and Temperature Profile



```
=====
MULTI-PASS SHELL-AND-TUBE HEAT EXCHANGER SYSTEM
=====
CONFIGURATION:
• Shell Side: Hot air flows perpendicular to tubes (cross-flow)
• Tube Side: Water flows through tubes in multiple passes
• Number of Passes: 4 passes (alternating flow direction)

FLOW ARRANGEMENT:
• Pass 1: Water flows left to right (inlet temperature)
• Pass 2: Water flows right to left (intermediate heating)
• Pass 3: Water flows left to right (continued heating)
• Pass 4: Water flows right to left (outlet temperature)

HEAT TRANSFER MECHANISM:
• Hot air enters at 500°C, exits at 200°C
• Water enters at 25°C, exits at 300°C
• Cross-flow arrangement maximizes heat transfer efficiency
• Multiple passes increase residence time and heat transfer area

DESIGN ADVANTAGES:
• Compact design with high heat transfer coefficient
• Multiple passes increase heat transfer effectiveness
• Cross-flow arrangement provides good temperature driving force
• Easy maintenance access to tube bundle
=====
```

```

import matplotlib.pyplot as plt
import matplotlib.patches as patches
import numpy as np

# Create figure with subplots
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(16, 14))

# ===== TOP PLOT: PHYSICAL LAYOUT (Based on hand sketch) =====
ax1.set_xlim(0, 16)
ax1.set_ylim(0, 12)
ax1.set_aspect('equal')
ax1.axis('off')

# Title
ax1.text(8, 11.5, 'Multi-Pass Shell-and-Tube Heat Exchanger',
        ha='center', va='center', fontsize=16, fontweight='bold')
ax1.text(8, 11, 'Configuration Based on Actual Design',
        ha='center', va='center', fontsize=12, style='italic')

# Shell (outer rectangular container as per sketch)
shell = patches.Rectangle((3, 3), 10, 6, linewidth=3,
                          edgecolor='black', facecolor='lightgray', alpha=0.3)
ax1.add_patch(shell)

# Shell label
ax1.text(8, 9.5, 'SHELL', ha='center', va='center', fontsize=12, fontweight='bold')

# Draw the serpentine tube path as shown in sketch
# Pass 1 (top, left to right)
pass1 = patches.Rectangle((3.5, 8), 9, 0.6, linewidth=2,
                          edgecolor='blue', facecolor='lightblue', alpha=0.7)
ax1.add_patch(pass1)

# Pass 2 (upper middle, right to left)
pass2 = patches.Rectangle((3.5, 7), 9, 0.6, linewidth=2,
                          edgecolor='blue', facecolor='lightcoral', alpha=0.7)
ax1.add_patch(pass2)

# Pass 3 (lower middle, left to right)
pass3 = patches.Rectangle((3.5, 6), 9, 0.6, linewidth=2,
                          edgecolor='blue', facecolor='lightgreen', alpha=0.7)
ax1.add_patch(pass3)

# Pass 4 (bottom, right to left)
pass4 = patches.Rectangle((3.5, 5), 9, 0.6, linewidth=2,
                          edgecolor='blue', facecolor='lightyellow', alpha=0.7)
ax1.add_patch(pass4)

# Water flow arrows inside each pass

```

```

# Pass 1: Left to right
for i in range(8):
    x_pos = 4 + i * 1.0
    ax1.arrow(x_pos, 8.3, 0.7, 0, head_width=0.1, head_length=0.1,
              fc='blue', ec='blue', linewidth=2)

# Pass 2: Right to left
for i in range(8):
    x_pos = 11.5 - i * 1.0
    ax1.arrow(x_pos, 7.3, -0.7, 0, head_width=0.1, head_length=0.1,
              fc='blue', ec='blue', linewidth=2)

# Pass 3: Left to right
for i in range(8):
    x_pos = 4 + i * 1.0
    ax1.arrow(x_pos, 6.3, 0.7, 0, head_width=0.1, head_length=0.1,
              fc='blue', ec='blue', linewidth=2)

# Pass 4: Right to left
for i in range(8):
    x_pos = 11.5 - i * 1.0
    ax1.arrow(x_pos, 5.3, -0.7, 0, head_width=0.1, head_length=0.1,
              fc='blue', ec='blue', linewidth=2)

# U-turn connections between passes (as shown in sketch)
# Connection from Pass 1 to Pass 2
conn1 = patches.FancyBboxPatch((12.5, 7.6), 0.8, 0.4,
                               boxstyle="round,pad=0.1",
                               facecolor='blue', alpha=0.5)
ax1.add_patch(conn1)
ax1.plot([12.5, 13.3, 13.3, 12.5], [8.0, 8.0, 7.6, 7.6], 'b-', linewidth=3)

# Connection from Pass 2 to Pass 3
conn2 = patches.FancyBboxPatch((2.7, 6.6), 0.8, 0.4,
                               boxstyle="round,pad=0.1",
                               facecolor='blue', alpha=0.5)
ax1.add_patch(conn2)
ax1.plot([3.5, 2.7, 2.7, 3.5], [7.0, 7.0, 6.6, 6.6], 'b-', linewidth=3)

# Connection from Pass 3 to Pass 4
conn3 = patches.FancyBboxPatch((12.5, 5.6), 0.8, 0.4,
                               boxstyle="round,pad=0.1",
                               facecolor='blue', alpha=0.5)
ax1.add_patch(conn3)
ax1.plot([12.5, 13.3, 13.3, 12.5], [6.0, 6.0, 5.6, 5.6], 'b-', linewidth=3)

# Water inlet and outlet (as per sketch)
# Water inlet (left side, connecting to Pass 1)
ax1.arrow(2, 8.3, 1.3, 0, head_width=0.15, head_length=0.2,
          fc='blue', ec='blue', linewidth=4)
ax1.text(1.5, 8.3, 'Water In\nT_inlet = 25°C', ha='right', va='center',

```

```

        fontsize=11, fontweight='bold', color='blue')

# Water outlet (right side, connecting from Pass 4)
ax1.arrow(12.7, 5.3, 1.3, 0, head_width=0.15, head_length=0.2,
          fc='red', ec='red', linewidth=4)
ax1.text(14.5, 5.3, 'Water Out\nT_outlet = 300°C', ha='left', va='center',
         fontsize=11, fontweight='bold', color='red')

# Hot air cross-flow (as shown in sketch - multiple downward arrows)
# Air inlet (top of shell)
ax1.text(8, 10, 'Hot Air In: T_in = 500°C', ha='center', va='center',
         fontsize=12, fontweight='bold', color='orange')

# Multiple air inlet arrows across the top
for i in range(5):
    x_pos = 4.5 + i * 1.8
    ax1.arrow(x_pos, 9.7, 0, -0.5, head_width=0.2, head_length=0.1,
              fc='orange', ec='orange', linewidth=3)

# Cross-flow arrows through the shell (perpendicular to tubes)
for i in range(4): # 4 rows
    for j in range(5): # 5 columns
        x_pos = 4.5 + j * 1.8
        y_pos = 8.5 - i * 0.7
        ax1.arrow(x_pos, y_pos, 0, -0.4, head_width=0.15, head_length=0.08,
                  fc='orange', ec='orange', alpha=0.6, linewidth=2)

# Air outlet (bottom of shell)
ax1.text(8, 2.5, 'Hot Air Out: T_out = 200°C', ha='center', va='center',
         fontsize=12, fontweight='bold', color='brown')

# Multiple air outlet arrows at the bottom
for i in range(5):
    x_pos = 4.5 + i * 1.8
    ax1.arrow(x_pos, 3.2, 0, -0.5, head_width=0.2, head_length=0.1,
              fc='brown', ec='brown', linewidth=3)

# Pass labels
ax1.text(8, 8.3, 'PASS 1', ha='center', va='center', fontsize=10, fontweight='bold')
ax1.text(8, 7.3, 'PASS 2', ha='center', va='center', fontsize=10, fontweight='bold')
ax1.text(8, 6.3, 'PASS 3', ha='center', va='center', fontsize=10, fontweight='bold')
ax1.text(8, 5.3, 'PASS 4', ha='center', va='center', fontsize=10, fontweight='bold')

# Temperature progression labels
ax1.text(1, 8.8, '25°C', fontsize=10, color='blue', fontweight='bold')
ax1.text(1, 7.8, '~100°C', fontsize=10, color='blue', fontweight='bold')
ax1.text(1, 6.8, '~200°C', fontsize=10, color='blue', fontweight='bold')
ax1.text(1, 5.8, '300°C', fontsize=10, color='red', fontweight='bold')

# ===== BOTTOM PLOT: SCHEMATIC AND ANALYSIS =====
ax2.set_xlim(0, 16)

```

```

ax2.set_ylim(0, 10)
ax2.axis('off')

# Title
ax2.text(8, 9.5, 'Heat Transfer Analysis and Problem Solution',
        ha='center', va='center', fontsize=16, fontweight='bold')

# Problem statement box
problem_text = """PROBLEM: Find mass flow rate ( $\dot{m}$ ) of water
Given:  $Q = 500 \text{ W/m}^2$ , Find  $\dot{m} = ?$ """
ax2.text(2, 8.5, problem_text, ha='left', va='center', fontsize=12, fontweight='bold',
        bbox=dict(boxstyle="round,pad=0.3", facecolor='lightcyan'))

# Solution approach box
solution_text = """SOLUTION APPROACH:
1. Energy Balance:  $Q = \dot{m} \times C_p \times \Delta T$ 
2. Heat Transfer:  $Q = U \times A \times F \times \Delta T_{LMTD}$ 
3. LMTD for counter-flow:  $\Delta T_{LMTD} = 187.189 \text{ K}$ 
4. Correction Factor:  $F = 0.95$  (multi-pass)
5. Solve:  $\dot{m} = Q / (C_p \times \Delta T)$ """
ax2.text(8, 7, solution_text, ha='center', va='center', fontsize=11,
        bbox=dict(boxstyle="round,pad=0.4", facecolor='lightyellow'))

# Results box
results_text = """FINAL RESULTS:
• Heat Transfer Rate:  $Q = 88,915 \text{ W}$ 
• Mass Flow Rate:  $\dot{m} = 0.0774 \text{ kg/s}$ 
• Correction Factor:  $F = 0.95$ 
• LMTD:  $187.189 \text{ K}$ """
ax2.text(14, 8.5, results_text, ha='right', va='center', fontsize=12, fontweight='bold',
        bbox=dict(boxstyle="round,pad=0.3", facecolor='lightgreen'))

# Flow configuration diagram (simplified)
ax2.text(2, 5, 'FLOW CONFIGURATION:', fontsize=12, fontweight='bold')

# Simplified serpentine path
y_positions = [4.5, 4, 3.5, 3]
directions = ['→', '←', '→', '←']
pass_names = ['Pass 1', 'Pass 2', 'Pass 3', 'Pass 4']

for i, (y_pos, direction, pass_name) in enumerate(zip(y_positions, directions, pass_names)):
    # Draw pass line
    ax2.plot([3, 6], [y_pos, y_pos], 'b-', linewidth=4, alpha=0.7)
    # Add direction arrow
    ax2.text(4.5, y_pos, direction, fontsize=16, ha='center', va='center',
            color='blue', fontweight='bold')
    # Add pass label
    ax2.text(2.5, y_pos, pass_name, fontsize=10, ha='right', va='center',
            fontweight='bold')

```

```

# U-turn connections
ax2.plot([6, 6.5, 6.5, 6], [4.5, 4.5, 4, 4], 'b-', linewidth=3)
ax2.plot([3, 2.5, 2.5, 3], [4, 4, 3.5, 3.5], 'b-', linewidth=3)
ax2.plot([6, 6.5, 6.5, 6], [3.5, 3.5, 3, 3], 'b-', linewidth=3)

# Cross-flow indication
ax2.text(8, 5, 'CROSS-FLOW AIR:', fontsize=12, fontweight='bold')
for i in range(4):
    x_pos = 9 + i * 0.8
    ax2.arrow(x_pos, 4.7, 0, -1.5, head_width=0.1, head_length=0.1,
              fc='orange', ec='orange', linewidth=2, alpha=0.7)

ax2.text(10.5, 4.8, '500°C', fontsize=10, color='orange', fontweight='bold')
ax2.text(10.5, 2.8, '200°C', fontsize=10, color='brown', fontweight='bold')

# Key parameters
params_text = """KEY PARAMETERS:
• U = 500 W/m²K (Overall heat transfer coefficient)
• A = 1.0 m² (Heat transfer area)
• Cp,water = 4180 J/kgK (Specific heat of water)
• Multi-pass configuration with cross-flow
• Correction factor F = 0.95 accounts for non-ideal flow"""

ax2.text(12, 4, params_text, ha='center', va='center', fontsize=10,
        bbox=dict(boxstyle="round,pad=0.4", facecolor='lavender'))

plt.tight_layout()
plt.show()

# Print detailed system description
print("=*70)
print("MULTI-PASS SHELL-AND-TUBE HEAT EXCHANGER - CORRECTED CONFIGURATION")
print("=*70)
print("BASED ON HAND-DRAWN SKETCH:")
print()
print("SERPENTINE FLOW PATH:")
print("• Water enters at left side (25°C)")
print("• Pass 1: Flows left to right (top tube bank)")
print("• U-turn connection to Pass 2")
print("• Pass 2: Flows right to left (upper middle tube bank)")
print("• U-turn connection to Pass 3")
print("• Pass 3: Flows left to right (lower middle tube bank)")
print("• U-turn connection to Pass 4")
print("• Pass 4: Flows right to left (bottom tube bank)")
print("• Water exits at right side (300°C)")
print()
print("CROSS-FLOW AIR:")
print("• Hot air enters from top at 500°C")
print("• Flows perpendicular to water tubes (cross-flow)")
print("• Exits from bottom at 200°C")

```

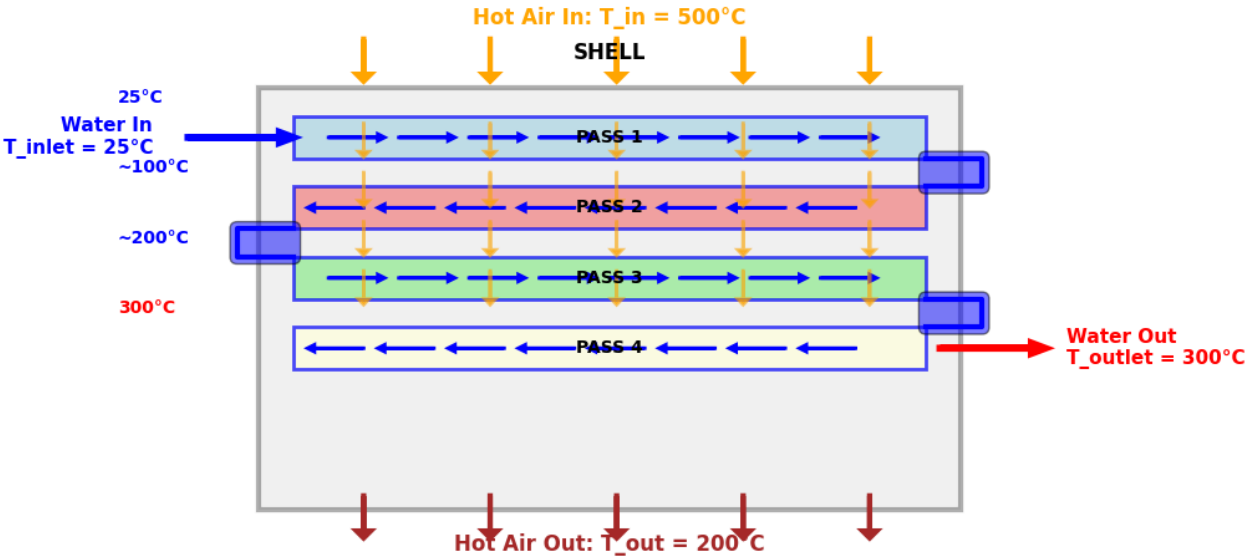


```
print()
print("PROBLEM SOLUTION:")
print("• Given:  $U = 500 \text{ W/m}^2$ ,  $A = 1.0 \text{ m}^2$ ")
print("• Find: Mass flow rate  $\dot{m} = ?$ ")
print("• Answer:  $\dot{m} = 0.0774 \text{ kg/s}$ ")
print("=*70)
```



Multi-Pass Shell-and-Tube Heat Exchanger

Configuration Based on Actual Design



Heat Transfer Analysis and Problem Solution

PROBLEM: Find mass flow rate (\dot{m}) of water
Given: $Q = 500 \text{ W/m}^2$, Find $\dot{m} = ?$

SOLUTION APPROACH:
1. Energy Balance: $Q = \dot{m} \times C_p \times \Delta T$
2. Heat Transfer: $Q = U \times A \times F \times \Delta T_{LMTD}$
3. LMTD for counter-flow: $\Delta T_{LMTD} = 187.189 \text{ K}$
4. Correction Factor: $F = 0.95$ (multi-pass)
5. Solve: $\dot{m} = Q / (C_p \times \Delta T)$

FINAL RESULTS:
• Heat Transfer Rate: $Q = 88,915 \text{ W}$
• Mass Flow Rate: $\dot{m} = 0.0774 \text{ kg/s}$
• Correction Factor: $F = 0.95$
• LMTD: 187.189 K

FLOW CONFIGURATION:



CROSS-FLOW AIR:

500°C
KEY PARAMETERS:
• $U = 500 \text{ W/m}^2\text{K}$ (Overall heat transfer coefficient)
• $A = 1.0 \text{ m}^2$ (Heat transfer area)
• $C_{p,\text{water}} = 4180 \text{ J/kgK}$ (Specific heat of water)
• Multi-pass configuration with cross-flow



Pass 4

• Correction factor $F = 0.95$ accounts for non-ideal flow

200°C

```
=====
MULTI-PASS SHELL-AND-TUBE HEAT EXCHANGER - CORRECTED CONFIGURATION
=====
```

BASED ON HAND-DRAWN SKETCH:

SERPENTINE FLOW PATH:

- Water enters at left side (25°C)
- Pass 1: Flows left to right (top tube bank)
- U-turn connection to Pass 2
- Pass 2: Flows right to left (upper middle tube bank)
- U-turn connection to Pass 3
- Pass 3: Flows left to right (lower middle tube bank)
- U-turn connection to Pass 4
- Pass 4: Flows right to left (bottom tube bank)
- Water exits at right side (300°C)

CROSS-FLOW AIR:

- Hot air enters from top at 500°C
- Flows perpendicular to water tubes (cross-flow)
- Exits from bottom at 200°C

PROBLEM SOLUTION:

- Given: $U = 500 \text{ W/m}^2$, $A = 1.0 \text{ m}^2$
- Find: Mass flow rate $\dot{m} = ?$
- Answer: $\dot{m} = 0.0774 \text{ kg/s}$

```
=====
```

```

import matplotlib.pyplot as plt
from matplotlib.patches import Rectangle, FancyArrow

# Create figure
fig, ax = plt.subplots(figsize=(10, 6))
ax.set_xlim(0, 10)
ax.set_ylim(0, 6)
ax.set_aspect('equal')
ax.axis('off')

# Shell (outer rectangle)
shell = Rectangle((1, 1), 8, 4, edgecolor='black', facecolor='lightgray', lw=2)
ax.add_patch(shell)

# Tubes (4 passes, horizontal lines)
tube_y_positions = [2, 2.5, 3, 3.5] # Vertical positions for 4 passes
for y in tube_y_positions:
    ax.plot([2, 8], [y, y], 'b-', lw=2) # Straight sections of tubes

# U-bends or connections between passes (simplified)
ax.plot([8, 8.5], [2, 2.5], 'b-', lw=2) # Pass 1 to 2
ax.plot([8, 8.5], [2.5, 3], 'b-', lw=2) # Pass 2 to 3
ax.plot([8, 8.5], [3, 3.5], 'b-', lw=2) # Pass 3 to 4 (exit)

# Hot fluid flow (shell side, left to right)
ax.add_patch(FancyArrow(1.5, 3.5, 2, 0, width=0.2, color='red'))
ax.add_patch(FancyArrow(5.5, 3.5, 2, 0, width=0.2, color='red'))
ax.text(0.5, 3.5, 'Hot Fluid In\n100°C', ha='right', va='center', color='red')
ax.text(9.5, 3.5, 'Hot Fluid Out', ha='left', va='center', color='red')

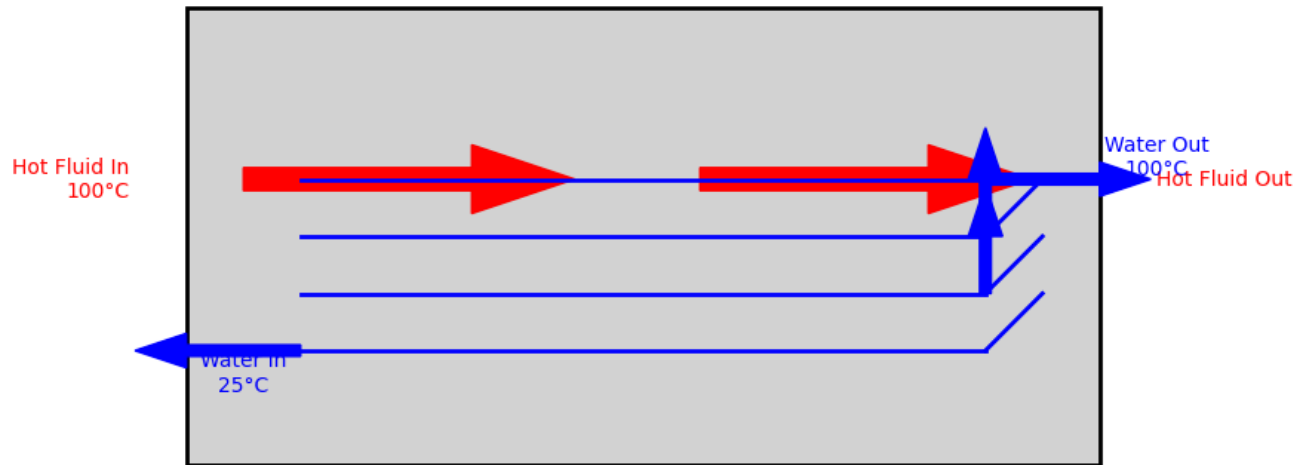
# Water flow (tube side, 4 passes, zigzag)
ax.add_patch(FancyArrow(2, 2, -1, 0, width=0.1, color='blue')) # Pass 1 entry
ax.add_patch(FancyArrow(8, 2.5, 0, 0.5, width=0.1, color='blue')) # Pass 2
ax.add_patch(FancyArrow(8, 3, 0, 0.5, width=0.1, color='blue')) # Pass 3
ax.add_patch(FancyArrow(8, 3.5, 1, 0, width=0.1, color='blue')) # Pass 4 exit
ax.text(1.5, 2, 'Water In\n25°C', ha='center', va='top', color='blue')
ax.text(9.5, 3.5, 'Water Out\n100°C', ha='center', va='bottom', color='blue')

# Labels and title
plt.title('Flow Diagram - 1 Shell Pass, 4 Tube Passes Heat Exchanger')
plt.show()

```



Flow Diagram - 1 Shell Pass, 4 Tube Passes Heat Exchanger



```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import fsolve

# Given parameters from the problem
T_water_in = 25 # °C
T_water_out = 80 # °C
T_air_in = 500 # °C
T_air_out = 200 # °C
U = 500 # W/m²·K
A = 1.0 # m²
Cp_water = 4180 # J/kg·K
Cp_air = 1005 # J/kg·K (assumed constant for air)
m_water = 0.536 # kg/s (from your calculation)
F = 0.88 # LMTD correction factor

# Calculate air mass flow rate from energy balance
Q = m_water * Cp_water * (T_water_out - T_water_in)
m_air = Q / (Cp_air * (T_air_in - T_air_out))

print(f"Heat transfer rate: {Q/1000:.2f} kW")
print(f"Water mass flow rate: {m_water:.3f} kg/s")
```

```

print(f"Air mass flow rate: {m_air:.3f} kg/s")

# Heat exchanger analysis for temperature distribution
def heat_exchanger_temp_distribution(n_points=100):
    """
    Calculate temperature distribution along heat exchanger length
    Assumes counter-flow for simplified analysis with correction factor applied
    """
    # Dimensionless position along heat exchanger (0 to 1)
    x = np.linspace(0, 1, n_points)

    # Heat capacity rates
    C_water = m_water * Cp_water
    C_air = m_air * Cp_air
    C_min = min(C_water, C_air)
    C_max = max(C_water, C_air)
    C_ratio = C_min / C_max

    # Number of Transfer Units (NTU)
    NTU = U * A * F / C_min

    # Effectiveness for counter-flow heat exchanger
    if C_ratio < 1:
        effectiveness = (1 - np.exp(-NTU * (1 - C_ratio))) / (1 - C_ratio * np.exp(-NTU * (1 - C_ratio)))
    else:
        effectiveness = NTU / (1 + NTU)

    # Temperature arrays
    T_water = np.zeros(n_points)
    T_air = np.zeros(n_points)

    # For counter-flow arrangement (water flows opposite to air)
    for i, pos in enumerate(x):
        # Local NTU at position
        local_NTU = NTU * pos

        # Local effectiveness
        if C_ratio < 1:
            if local_NTU > 0:
                local_eff = (1 - np.exp(-local_NTU * (1 - C_ratio))) / (1 - C_ratio * np.exp(-local_NTU * (1 - C_ratio)))
            else:
                local_eff = 0
        else:
            local_eff = local_NTU / (1 + local_NTU) if local_NTU > 0 else 0

        # Calculate temperatures based on local effectiveness
        Q_local = local_eff * C_min * (T_air_in - T_water_in)

        # Water temperature (increases along flow direction)
        T_water[i] = T_water_in + Q_local / C_water

```

```

    # Air temperature (decreases along flow direction)
    T_air[i] = T_air_in - Q_local / C_air

    return x, T_water, T_air

# Alternative simplified linear approximation for visualization
def linear_temp_distribution(n_points=100):
    """
    Simplified linear temperature distribution for visualization
    """
    x = np.linspace(0, 1, n_points)

    # Linear interpolation (simplified assumption)
    T_water = T_water_in + (T_water_out - T_water_in) * x
    T_air = T_air_in - (T_air_in - T_air_out) * x

    return x, T_water, T_air

# Calculate temperature distributions
x_pos, T_water_profile, T_air_profile = linear_temp_distribution(100)

# Create the plot
plt.figure(figsize=(12, 8))

# Main temperature profile plot
plt.subplot(2, 2, 1)
plt.plot(x_pos, T_water_profile, 'b-', linewidth=2.5, label='Water (Cold Fluid)')
plt.plot(x_pos, T_air_profile, 'r-', linewidth=2.5, label='Air (Hot Fluid)')
plt.xlabel('Normalized Position along Heat Exchanger')
plt.ylabel('Temperature (°C)')
plt.title('Temperature Profiles in Multi-Pass Shell-and-Tube Heat Exchanger')
plt.legend()
plt.grid(True, alpha=0.3)

# Temperature difference plot
plt.subplot(2, 2, 2)
temp_diff = T_air_profile - T_water_profile
plt.plot(x_pos, temp_diff, 'g-', linewidth=2.5, label='ΔT (Air - Water)')
plt.axhline(y=np.mean(temp_diff), color='orange', linestyle='--',
            label=f'Average ΔT = {np.mean(temp_diff):.1f}°C')
plt.xlabel('Normalized Position along Heat Exchanger')
plt.ylabel('Temperature Difference (°C)')
plt.title('Local Temperature Difference')
plt.legend()
plt.grid(True, alpha=0.3)

# Heat flux distribution (proportional to temperature difference)
plt.subplot(2, 2, 3)
heat_flux = U * temp_diff / 1000 # Convert to kW/m²
plt.plot(x_pos, heat_flux, 'm-', linewidth=2.5, label='Local Heat Flux')
plt.xlabel('Normalized Position along Heat Exchanger')

```