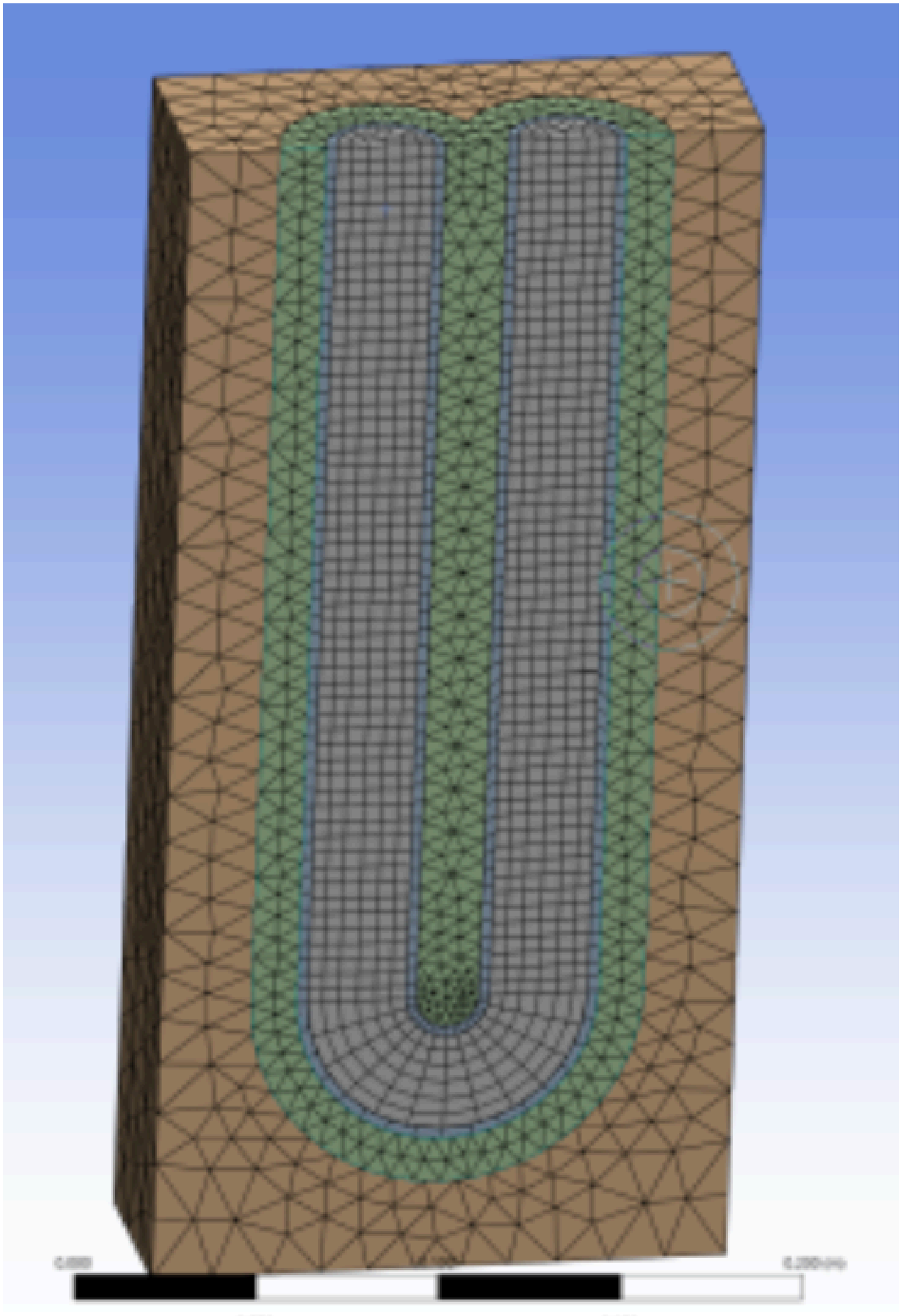
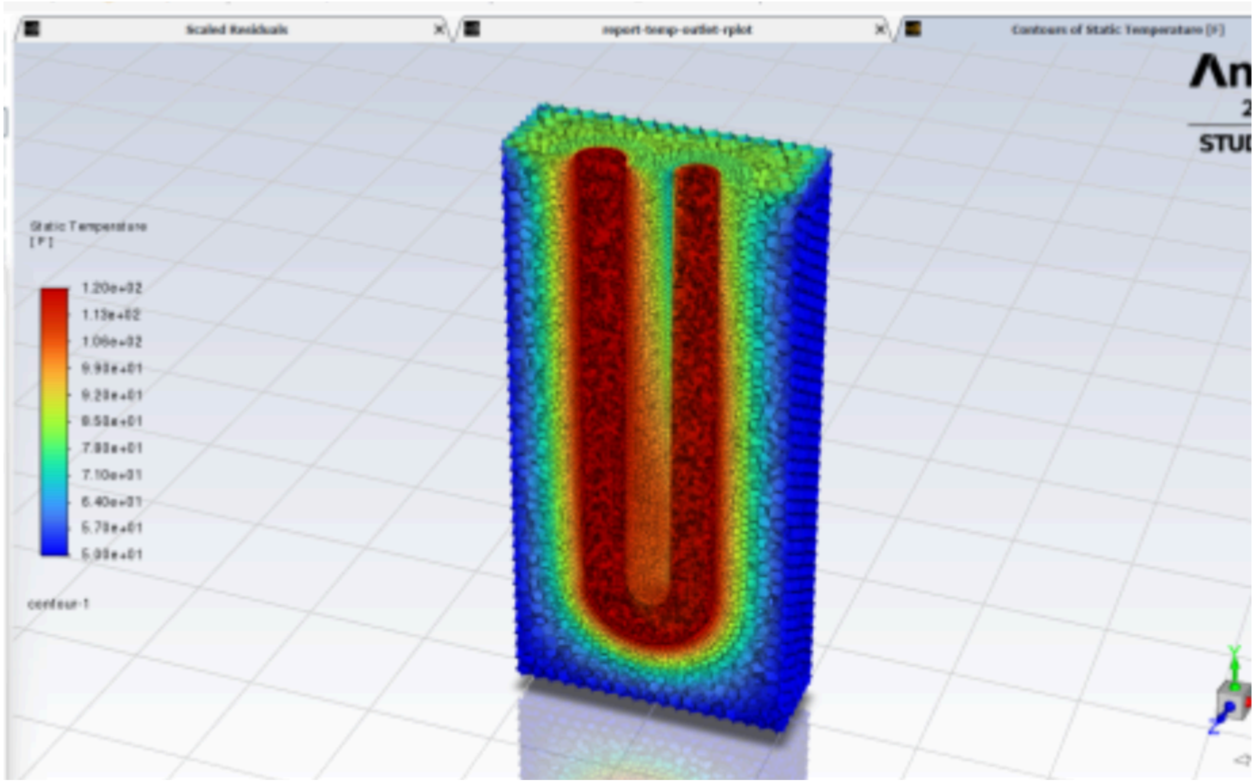
```
## Maintaining Dimensionless Similitude between Models

import CoolProp.CoolProp as CP
import numpy as np
from scipy.optimize import fsolve

Tin = 40 # C = 104 F
Tb_estimate = 37.22222 # Assuming 10 F degree temperature difference

do = 1.315 * 0.0254 # m
di = 1.077 * 0.0254 # m
L_og = 91.44 # m = 300 ft
L_scaled = 6.096 # m = 20 ft

k_pipe = 0.39 # W/m*K
dnom = (do + di)/2

Re_og = 7343 # Assuming barely Turbulent referencing research paper

# Re = (rho*v*D)/mu

mu = CP.PropsSI("VISCOSITY", "T", Tb_estimate+273.15, "Q", 0, "Water") #
Pa*s
rho = CP.PropsSI("D", "T", Tb_estimate+273.15, "Q", 0, "Water") # kg/m^3
Pr = CP.PropsSI("PRANDTL", "T", Tb_estimate+273.15, "Q", 0, "Water")
k = CP.PropsSI("CONDUCTIVITY", "T", Tb_estimate+273.15, "Q", 0, "Water") #
W/m*K
cp = CP.PropsSI("C", "T", Tb_estimate+273.15, "Q", 0, "Water") # J/ kg*K

v_avg = (Re_og*mu)/(rho*di) # m/s
mdot = v_avg*(np.pi/4)*(di**2)*rho # kg/s

Pe = Re_og*Pr

# To calcualte Nusselt Relation use eq 8-71 Second Petukhov equation
modified by Gnielinski
# Equation found in Heat and Mass Transfer Yunus A Cengel

f = (0.790*np.log(Re_og) - 1.64)**(-2) # First Petukhov equation for
smooth tubes
Nu = ((f/8)*(Re_og - 1000)*Pr)/(1+12.7*((f/8)**(0.5))*((Pr**(2/3))-1))

# Time to calculate the internal heat transfer coefficient

hi = (Nu*k)/di # W/m^2*K
```

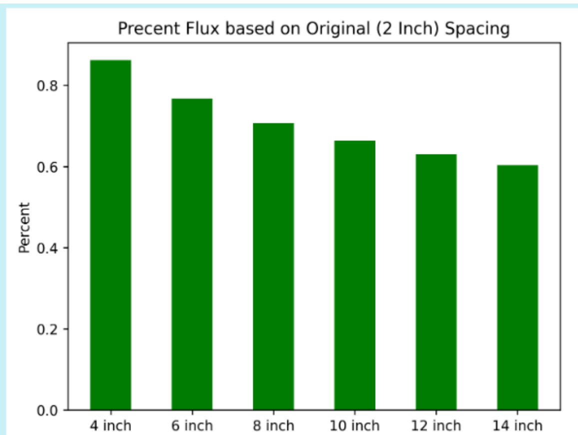
```
# Will proceed to calculate overall heat transfer coefficient U*A with
knowledge from
# Heat and Mass Transfer Yunus A Cengel Pg. 682

U = hi #+ (np.log(do/di))/(2*np.pi*k_pipe*L_og))**(-1) # W/K

# In the above expression the thermal resistance from the wall of the pipe
is
# negligible since the thickness  is small enough
# Above UA value is calculated for the full 150 feet depth of the u-tub.
Need to scale to 10 feet depth.

NTU_og = (U*(np.pi*dnom*L_og))/(mdot*cp)

def equations(vars):
    D, v = vars
    if D <= 0 or v <= 0:
        return [1e6, 1e6]  # prevent non-physical values
    Re_scaled = (rho * v * D) / mu
    mdot_scaled = rho * v * ((np.pi / 4) * D**2)
    f_scaled = (0.790 * np.log(Re_scaled) - 1.64)**(-2)
    Nu_scaled = ((f_scaled/8)*(Re_scaled - 1000)*Pr) / (1 + 12.7 *
(f_scaled/8)**0.5 * (Pr**(2/3) - 1))
    hi_scaled = (Nu_scaled * k) / D
    U_scaled = hi_scaled
    UA_scaled = U_scaled * np.pi * D * L_scaled
    NTU_scaled = UA_scaled / (mdot_scaled * cp)

    return [Re_scaled - Re_og, NTU_scaled - NTU_og]

D_scaled, v_scaled = fsolve(equations, [0.02, 0.4], maxfev=1000)

v = v_scaled
D = D_scaled
Re_scaled = (rho * v * D) / mu
mdot_scaled = rho * v * ((np.pi / 4) * D**2)
f_scaled = (0.790 * np.log(Re_scaled) - 1.64)**(-2)
Nu_scaled = ((f_scaled/8)*(Re_scaled - 1000)*Pr) / (1 + 12.7 *
(f_scaled/8)**0.5 * (Pr**(2/3) - 1))
hi_scaled = (Nu_scaled * k) / D
U_scaled = hi_scaled
UA_scaled = U_scaled * np.pi * D * L_scaled
NTU_scaled = UA_scaled / (mdot_scaled * cp)

print(f"\nOriginal Re: {Re_og:.2f}")
print(f"Original NTU: {NTU_og:.2f}")
```

```
print(f"\nDifference in Re numbers: {Re_scaled - Re_og:.6f} ")
print(f"Difference in NTU numbers: {NTU_scaled - NTU_og:.6f} ")

print(f"\nModified Inner Diameter to: {D:.5f} m")
print(f"\nModified Inlet Velocity to: {v:.5f} m/s")
```





- Temperature distributions from analytical solution based on counter current tubes analysis.
- Temperature contours from a scaled simulation in Ansys fluent.
- Heat flux quantitative results and relative percentage based on spacing.



Heat and Mass Transfer (Çengel & Ghajar)
Chapter 11: Heat Exchangers

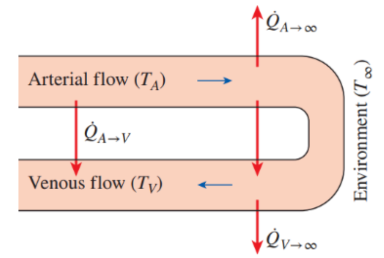


FIGURE 11–37
Simplified extremity model for a countercurrent heat exchanger.

$$
\begin{aligned}
&\dot{m}c_p\frac{dT_A}{dx} + (UA_s)_{AV}(T_A - T_V) + (UA_s)_A(T_A - T_\infty) = 0 \quad \text{Artery}\\
&\dot{m}c_p\frac{dT_V}{dx} + (UA_s)_{AV}(T_A - T_V) - (UA_s)_V(T_V - T_\infty) = 0 \quad \text{Vein}
\end{aligned}
\tag{11–44}
$$

If we solve Eq. 11–44, we get

$$
\frac{T_A - T_\infty}{T_o - T_\infty} = \frac{\sqrt{1+2\left(\frac{N_i}{N_o}\right)}\cosh\left(N_o\sqrt{1+2\left(\frac{N_i}{N_o}\right)}\right)\left(1-\frac{x}{L}\right) + \sinh\left(N_o\sqrt{1+2\left(\frac{N_i}{N_o}\right)}\right)\left(1-\frac{x}{L}\right)}{\sqrt{1+2\left(\frac{N_i}{N_o}\right)}\cosh\left(N_o\sqrt{1+2\left(\frac{N_i}{N_o}\right)}\right) + \sinh\left(N_o\sqrt{1+2\left(\frac{N_i}{N_o}\right)}\right)}
$$

$$
\frac{T_V - T_\infty}{T_o - T_\infty} = \frac{\sqrt{1+2\left(\frac{N_i}{N_o}\right)}\cosh\left(N_o\sqrt{1+2\left(\frac{N_i}{N_o}\right)}\right)\left(1-\frac{x}{L}\right) - \sinh\left(N_o\sqrt{1+2\left(\frac{N_i}{N_o}\right)}\right)\left(1-\frac{x}{L}\right)}{\sqrt{1+2\left(\frac{N_i}{N_o}\right)}\cosh\left(N_o\sqrt{1+2\left(\frac{N_i}{N_o}\right)}\right) + \sinh\left(N_o\sqrt{1+2\left(\frac{N_i}{N_o}\right)}\right)}
$$