Nuclear Reactor Thermal-Hydraulics

NUGN520 - Homework

By

GUILLAUME L'HER

Department of Nuclear Engineering
COLORADO SCHOOL OF MINES

Homework submitted for the Nuclear Reactor Thermal-Hydraulics class at the Colorado School of Mines.

SPRING 2017

Table of Contents

CHAPTER 1

Reactor Heat Generation

Several exercises from the book written by M. M. El Wakil [1] are tackled in this homework. The problems in this section relate to the sixth and seventh chapter of the book, covering the subject of heat conduction in reactor elements.

1.1 [6-20] - Variable heat generation

1.1.1 Problem

A cylindrical fuel elment, 0.06 ft in diameter using UO_2 as fuel material, generates 5×10^6 $Btu.h^{-1}.ft^{-3}$ at its surface when that surface is at $2000\ ^\circ F$. The thermal conductivity may be considered constant at $1.25\ Btu.h^{-1}.ft^{-1}.^\circ F^{-1}$. How much hotter would the center of this element operate if the heat generation increases by 1 percent for each 1000 °F temperature rise above surface conditions instead of being uniform at the above value?

1.1.2 Solution

In the uniform cylinder, we know Equation 1.1.

$$T_m = T_s + \frac{q'''R^2}{4k_f} \tag{1.1}$$

Consequently, $T_m = 2900\ ^\circ F$.

Now, if the heat generation depends on the temperature following the given relation, then we can write Equation 1.2.

$$q''' = q_s''' + 10^{-5} q_s''' \Delta T \tag{1.2}$$

For each 1000 $°F$ temperature increase, q"' will gain 1 percent. Hence, we can now solve the Poisson equation for $T' = q_s''' + 10^{-5} q_s''' \Delta T$. The solution is given by Equation 1.3, following the book.

$$T' = AJ_0\left(\sqrt{\frac{10^{-5}q_s'''}{k_f}}r\right) + BY_0\left(\sqrt{\frac{10^{-5}q_s'''}{k_f}}r\right) \tag{1.3}$$

We know two boundary conditions. First and foremost, the temperature needs to be finite at the center of the cylinder, $r = 0$. Thus, considering that $Y_0(0) \rightarrow -\infty$, $B = 0$. Next, we know that at the boundary, $q''' = 5 \times 10^6\ Btu.h^{-1}.ft^{-3}$. Consequently, since we have $T' = q'''$, we have Equation 1.4 at $R = 0.03\ ft$.

$$AJ_0\left(\sqrt{\frac{10^{-5}q_s'''}{k_f}}R\right) = 5 \times 10^6 \tag{1.4}$$

Solving for A, we obtain $A = 5.045 \times 10^6$.

So, we now have Equation 1.5.

$$5.045 \times 10^6 * J_0\left(\sqrt{\frac{10^{-5}q_s'''}{k_f}}r\right) = q_s''' + 10^{-5} q_s''' \Delta T \tag{1.5}$$

At $r = 0$, since $J_0(0) = 1$, we can write Equation 1.6.

$$5.045 \times 10^6 = 5 \times 10^6 + 10^{-5} * 5 \times 10^6 * \Delta T \tag{1.6}$$

Finally, we obtain $\Delta T = T_m - T_s = 906.2\ °F$, hence $T_m = 2906.2\ °F$. If we take into account the given change in heat generation, we obtain a maximum temperature 6.2 $°F$ hotter.

1.2 7-1 - Outer square corner

1.2.1 Problem

Derive the finite difference equation of temperature for a nodal point at the outer square corner of a body with heat generation, having thermal conductivity k, and cooled by a fluid at T_f with a uniform heat transfer coefficient h.

1.2.2 Solution

We can write the heat balance for a node n on a corner, $q_g + q_{xo} + q_{xi} + q_{yo} + q_{yi}$. In our case, $q_{xo} = h(\Delta x)(T_f - T_n)$ and $q_{yo} = h(\Delta y)(T_f - T_n)$.

So, we can write Equation 1.7, considering $\Delta x = \Delta y$.

$$\frac{1}{2}q'''(\Delta x)^2 + k[(T_{n-\Delta x} - T_n) + (T_{n-\Delta y} - T_n)] + 2h\Delta x(T_f - T_n) = 0 \tag{1.7}$$

So, we can divide by k and reorganize.

$$\frac{1}{2k}q'''(\Delta x)^2 + [(T_{n-\Delta x} - T_n) + (T_{n-\Delta y} - T_n)] + 2\frac{h\Delta x}{k}(T_f - T_n) = 0 \tag{1.8}$$

We can recognize the Biot number, $Bi_{\Delta x} = \frac{h\Delta x}{k}$, and $\Delta T_g = \frac{q'''(\Delta x)^2}{k}$.

$$\frac{\Delta T_g}{2} + T_{n-\Delta x} + T_{n-\Delta y} - 2T_n + 2Bi_{\Delta x}T_f - 2Bi_{\Delta x}T_n = 0 \tag{1.9}$$

And so:

$$T_n = \frac{\Delta T_g}{4Bi_{\Delta x} + 4} + \frac{T_{n-\Delta x} + T_{n-\Delta y} + 2Bi_{\Delta x}T_f}{2Bi_{\Delta x} + 2} \tag{1.10}$$

1.3 [G1] - Fine mesh cruciform

1.3.1 Problem

Calculate the temperature in a cruciform with a mesh $\Delta x = \Delta y = 0.1$.

1.3.2 Solution

We can define the origin, taken at the center of the cruciform, to be $T(0,0)$. By symmetry, we know that $T(i,j) = T(-i,j)$ and $T(i,j) = T(i,-j)$. Consequently, the solution will be complete when $T(i,j), i \in [0,11], j \in [0,1]$ are computed. By symmetry, we also know that $T(0,1) = T(1,0)$, so we can get rid of $T(0,1)$ in our computation.

We can approximate the temperature at these discrete points by using Equation 1.11, knowing that $\Delta x = \Delta y$.

$$T_n = \frac{T_{n+\Delta x} + T_{n-\Delta x} + T_{n+\Delta y} + T_{n-\Delta y}}{4} + \frac{\Delta T_g}{2} \tag{1.11}$$

Where, in the case of uniform heat generation:

$\Delta T_g = \frac{(\Delta x)^2 q'''}{2k}$

In our case, we consider $q''' = 1 \times 10^7$ and $k = 19.84$. Consequently, $\Delta T_g = 17.5\ °F$. We also consider a surface temperature of $600\ °F$.

Our goal is to solve the matrix Equation 1.12 for x.

$$A.x = b \tag{1.12}$$

We can calculate a few values in order to construct the matrice A and b, using Equation 1.11. We can obtain:

$$T(0,0) = T(1,0) + \frac{\Delta T_g}{2} \implies T(0,0) - T(1,0) = \frac{\Delta T_g}{2} \tag{1.13}$$

$$T(1,0) = \frac{T(0,0) + T(2,0) + 2T(1,1)}{4} + \frac{\Delta T_g}{2} \implies T(1,0) - \frac{T(0,0)}{4} - \frac{T(2,0)}{4} - \frac{T(1,1)}{2} = \frac{\Delta T_g}{2} \tag{1.14}$$

$$T(1,1) = \frac{2T(1,0) + 2T(2,1)}{4} + \frac{\Delta T_g}{2} \implies T(1,1) - \frac{T(1,0)}{2} - \frac{T(2,1)}{2} = \frac{\Delta T_g}{2} \tag{1.15}$$

$$T(2,0) = \frac{T(1,0) + T(3,0) + 2T(2,1)}{4} + \frac{\Delta T_g}{2} \implies T(2,0) - \frac{T(1,0)}{4} - \frac{T(3,0)}{4} - \frac{T(2,1)}{2} = \frac{\Delta T_g}{2} \tag{1.16}$$

$$\begin{aligned} T(2,1) &= \frac{T(1,1) + T(3,1) + T_S + T(2,0)}{4} + \frac{\Delta T_g}{2} \\ \implies T(2,1) - \frac{T(1,1)}{4} - \frac{T(3,1)}{4} - \frac{T(2,0)}{4} &= \frac{T_S}{4} + \frac{\Delta T_g}{2} \end{aligned} \tag{1.17}$$

$T(i,0)$ and $T(i,1)$ for $i \in [2,10]$ will be identical, given the geometry of the problem.

$$T(11,0) = \frac{T(10,0) + 2T(11,1) + T_S}{4} + \frac{\Delta T_g}{2} \implies T(11,0) - \frac{T(11,1)}{2} - \frac{T(10,0)}{4} = \frac{T_S}{4} + \frac{\Delta T_g}{2} \tag{1.18}$$

$$T(11,1) = \frac{T(11,0) + T(10,1) + 2T_S}{4} + \frac{\Delta T_g}{2} \implies T(11,1) - \frac{T(11,0)}{4} - \frac{T(10,1)}{4} = \frac{T_S + \Delta T_g}{2} \tag{1.19}$$

Consequently, we can write A:

$$A_{0\to 11} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ -0.25 & 1 & -0.5 & -0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & -0.5 & 1 & 0 & -0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & -0.25 & 0 & 1 & -0.5 & -0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & -0.25 & -0.25 & 1 & 0 & -0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & -0.25 & 0 & 1 & -0.5 & -0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & -0.25 & -0.25 & 1 & 0 & -0.25 & 0 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & -0.25 & 0 & 1 & -0.5 & -0.25 & 0 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.25 & -0.25 & 1 & 0 & -0.25 & 0 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.25 & 0 & 1 & -0.5 & -0.25 & 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.25 & -0.25 & 1 & 0 & -0.25 & 0 & 0 & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.25 & 0 & 1 & -0.5 & -0.25 & 0 & \end{bmatrix}$$

$$A_{12\to 22} = \begin{bmatrix}
 & 0 & -0.25 & -0.25 & 1 & 0 & -0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & -0.25 & 0 & 1 & -0.5 & -0.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & -0.25 & -0.25 & 1 & 0 & -0.25 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & -0.25 & 0 & 1 & -0.5 & -0.25 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & -0.25 & -0.25 & 1 & 0 & -0.25 & 0 & 0 & 0 & 0 \\
\dots & 0 & 0 & 0 & 0 & 0 & 0 & -0.25 & 0 & 1 & -0.5 & -0.25 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.25 & -0.25 & 1 & 0 & -0.25 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.25 & 0 & 1 & -0.5 & -0.25 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.25 & -0.25 & 1 & 0 & -0.25 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.25 & 0 & 1 & -0.5 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.25 & -0.25 & 1
\end{bmatrix}$$

And we can also write b:

$$b = \begin{bmatrix} 8.75 \\ 8.75 \\ 8.75 \\ 8.75 \\ 158.75 \\ 8.75 \\ 158.75 \\ 8.75 \\ 158.75 \\ 8.75 \\ 158.75 \\ 8.75 \\ 158.75 \\ 8.75 \\ 158.75 \\ 8.75 \\ 158.75 \\ 8.75 \\ 158.75 \\ 8.75 \\ 158.75 \\ 158.75 \\ 308.75 \end{bmatrix}$$

We can now solve Equation 1.12, using Python. The python script is given in 1.3.2.1. We obtain the following results (Equation 1.20):

$$
\begin{aligned}
T(0,0) &= 723.8\\
T(1,0) &= 715.1\\
T(1,1) &= 703.2\\
T(2,0) &= 695.1\\
T(2,1) &= 673.8\\
T(3,0) &= 682.5\\
T(3,1) &= 662.0\\
T(4,0) &= 675.9\\
T(4,1) &= 656.8\\
T(5,0) &= 672.5\\
T(5,1) &= 654.3\\
T(6,0) &= 670.6\\
T(6,1) &= 652.9\\
T(7,0) &= 668.9\\
T(7,1) &= 651.8\\
T(8,0) &= 666.7\\
T(8,1) &= 650.2\\
T(9,0) &= 662.5\\
T(9,1) &= 647.2\\
T(10,0) &= 654.0\\
T(10,1) &= 641.1\\
T(11,0) &= 636.3\\
T(11,1) &= 628.1
\end{aligned}
\tag{1.20}
$$

1.3.2.1 Python script

```
import numpy as np

m = 23
dx = 0.1
deltax, k, qtriple = (dx/12), 19.84, 1.e7
ts = 600.
dtg = (0.5*deltax**2)*qtriple/k

amat = np.zeros((m,m))
b = np.zeros(m)

amat[0,0] = 1.
amat[0,1] = -1.
b[0] = dtg/2.

amat[1,0] = -0.25
amat[1,1] = 1.
amat[1,2] = -0.5
amat[1,3] = -0.25
b[1] = dtg/2.


amat[2,1] = -0.5
amat[2,2] = 1.
amat[2,4] = -0.5
b[2] = dtg/2.

for i in range(3, m, 2):
    try:
        amat[i,i-2] = -0.25
        amat[i,i] = 1.
        amat[i,i+1] = -0.5
        amat[i,i+2] = -0.25
        b[i] = dtg/2.
    except IndexError:
        break

for i in range(4, m, 2):
    try:
        amat[i,i-2] = -0.25
        amat[i,i-1] = -0.25
        amat[i,i] = 1.
        amat[i,i+2] = -0.25
        b[i] = ts/4. + dtg/2.
    except IndexError:
        break

amat[m-1,m-1] = 1.
amat[m-1,m-2] = -0.25
amat[m-1,m-3] = -0.25
b[m-1] = ts/2. + dtg/2.
```

```
amat[m-2,m-2] = 1.
amat[m-2,m-1] = -0.5
amat[m-2,m-4] = -0.25
b[m-2] = ts/4. + dtg/2.

ysol = np.linalg.solve(amat,b)

print('Temperature solution:')

temp = []
for i in range(12):
    for j in range(2):
        if i == 0 and j == 1:
            continue
        temp.append("T(%s,%s)" % (i, j))

for i,t in enumerate(temp):
    print("%s = %.1f" % (t, ysol[i]))

# check of solution:
if not np.allclose(np.dot(amat, ysol), b):
    print("Solution does not match!")
```

Bibliography

[1] M. M. El-Wakil, *Nuclear Heat Transport*, American Nuclear Society, 1993.