Final Project: Part I

Kian Greene, Cody Kohn, Joseph Sauerbrun, and M. Catherine Yopp

03 May 2022

Abstract

This paper analyzes the application of an explicit finite difference method to find the solutions of two diffusion systems with annular geometries. Although similar in geometry, the difference in the partial differential equations used requires the derivation of two separate finite difference schemes and therefore, different temporal and spatial step sizes to satisfy their respective stability conditions. Chapter 1 explores the heat conduction in fins of various metal materials while Chapter 2 explores the drawdown height of an aquifer. Both systems were able to produce reliable, stable, and expected results.

Chapter 1

Question IV: Transient Heat Conduction in a Fin

1.1 Introduction

The transient temperature distribution within an annular fin in a heating system satisfies the following PDE:

$$\rho C_p \frac{\partial T}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}(kr\frac{\partial T}{\partial r}) - \frac{2h}{w}(T - T_\infty) \tag{1.1}$$

After applying the product rule:

$$\rho C_p \frac{\partial T}{\partial t} = k(\frac{\partial^2 T}{\partial r^2} + \frac{1}{r}\frac{\partial T}{\partial r}) - \frac{2h}{w}(T - T_\infty) \tag{1.2}$$

The above PDE is subject to the following initial and boundary conditions:

$$T(R_{in}, t) = 20 + 40(1 - e^{\frac{-t}{120}}) \tag{1.3}$$

$$hT(R_{out}, t) + k\frac{\partial T}{\partial r}(R_{out}, t) = hT_\infty \tag{1.4}$$

$$T(r, 0) = T_\infty \tag{1.5}$$

Three materials were tested to determine the temperature distribution in the fin after 600 seconds. For each case, the following constants were used: h=50$\frac{W}{m^2 * K}$, w=5mm, $T_\infty = 20°C$, $R_{in} = 2mm$, and $R_{out} = 5mm$. The three materials used, along with their material properties and Fourier number (evaluated at 600 seconds with a characteristic length of 5mm) can be seen in the table below:

Material	$\rho(\frac{kg}{m^3})$	$C_p(\frac{J}{kg*K})$	$k(\frac{W}{m*K})$	Fo
Copper	8933	385	401	2798
Steel	8055	480	15.1	93.73
Bronze	8800	420	52	337.6

Note: The Fourier number is the ratio of conduction to thermal storage and is defined by $Fo = \frac{\alpha t}{L^2}$ where α is the thermal diffusivity, t is a characteristic time, and L is a characteristic length.

1.2 Math Modeling and Analysis Methods

To solve the IBVP (1.2), a finite difference scheme was used. Replacing the first derivatives with a forward difference in time, and the second derivative with a central difference in space, the discretization of the IBVP is as follows:

$$u_j^{n+1} = u_j^n + \frac{\Delta t}{\rho C_p}(k(\frac{u_{j-1}^n - 2u_j^n + u_{j+1}^n}{\Delta r^2}) + \frac{k}{r_j}(\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta r}) - \frac{2h}{w}u_j^n + T_\infty) \quad (1.6)$$

In (1.6), "j" denotes the discretization in space and ranges from 1 to "N", and "n" denotes the discretization in time and ranges from 1 to "M". Updating boundary conditions (1.3), (1.4), and (1.5) using this finite difference scheme yields:

$$u_1^n = 20 + 40(1 - e^{\frac{-t_n}{120}}) \quad (1.7)$$

$$u_{N+1}^n = u_{N-1}^n + \frac{2\Delta r}{k}(hT_\infty - hu_N^n) \quad (1.8)$$

$$u_j^1 = T_\infty \quad (1.9)$$

Simplifying (1.6) by substituting $\lambda = \frac{k}{\rho C_p}\frac{\Delta t}{\Delta r^2}$, $\mu = \frac{k}{\rho C_p}\frac{\Delta t}{\Delta r}$, $\beta = \frac{2h\Delta t}{\rho C_p w}$, and $S = \frac{T_\infty \Delta t}{\rho C_p}$ yields the following finite difference scheme:

$$u_j^{n+1} = (\lambda - \frac{\mu}{2r_j})u_{j-1}^n + (1 - 2\lambda - \beta)u_j^n + (\lambda + \frac{\mu}{2r_j})u_{j+1}^n + S \quad (1.10)$$

It is important to note that the the Robin boundary condition (1.8) requires the use of a "ghost node" due to u_{N+1}^n appearing in the equation. To solve (1.10), an explicit Euler method was used. To ensure a correct solution, the stability criteria needed to be checked.

$$\Delta t_{crit} = \frac{(\Delta r \rho C_p)(\Delta r + h)}{2k} \quad (1.11)$$

With a given Δr, the critical value for Δt could be calculated according to (1.11). The chosen Δt value had to be less than or equal to this value to ensure an accurate solution. Since Δt_{crit} is a function of material properties, it was adjusted to meet the critera for all three materials.

1.3 Results

After the temperature distribution all three materials were calculated, their temperature profiles at 600 seconds were compared. Taking $\Delta r = 1.66 * 10^{-04}$ and $\Delta t = 1.00 * 10^{-04}$ the stability condition was satisfied for all three materials (the smallest Δt_{crit} of the three materials was $1.19 * 10^{-04}$

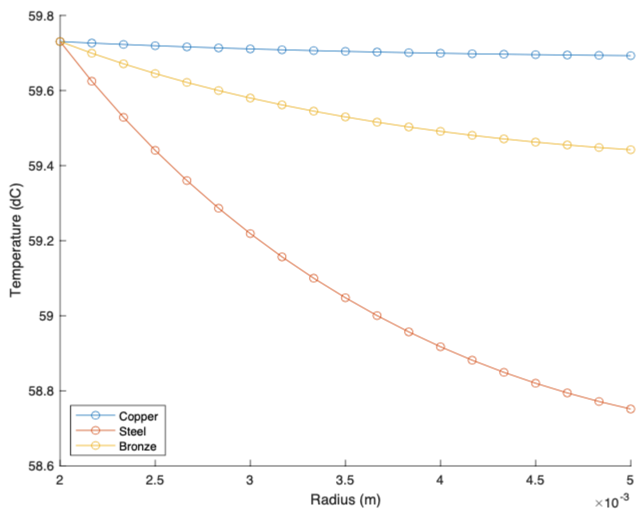

Figure 1.1: Temperature Distribution at 600 Seconds

As seen in Figure 1.1, the three materials have different temperature profiles after 600 seconds. As predicted by Fourier number, Copper has the most uniform temperature distribution and is basically isothermal. Likewise, Steel has the largest temperature gradient between the inner and outer radius, with bronze being between the two.

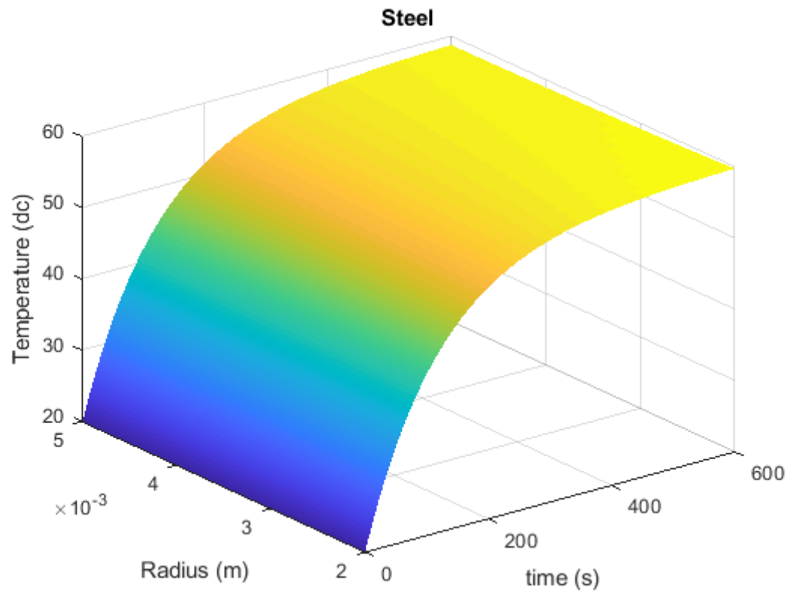

Figure 1.2: Temperature Distribution of Steel

Figure 1.2 shows the numerically solved solution domain to (1.1) for the Steel. To further analyze the temperature distribution, a contour plot was produced as seen in figure 1.3.

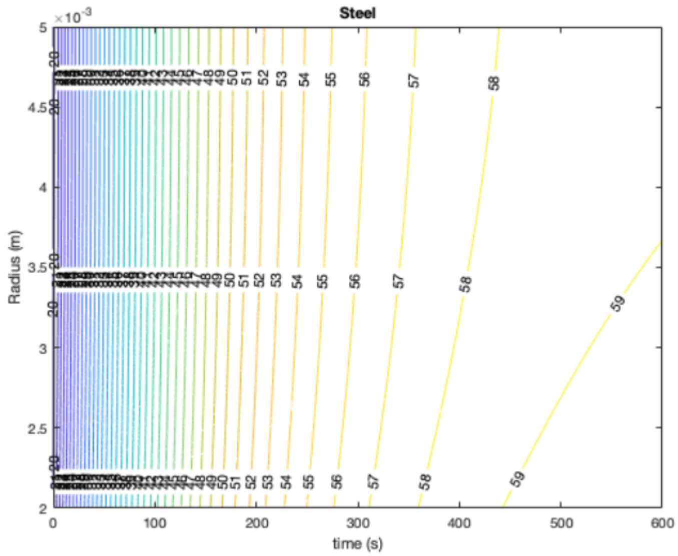

Figure 1.3: Temperature Distribution of Steel

Chapter 2

Question V: The Drawdown Rates of an Aquifer

2.1 Introduction

A one meter diameter well is pumping water out by drawing from an aquifer surrounding it with a 100 meter radius from the center of the well. To model the drawdon of the water level within the aquifer, the diffusion equation can be used in polar coordiantes (Figure 2.1). This specific system has no decay $\beta(r,t)$ and will be analyzed both with and without a source term $s(r,t)$. If $p(t) = 0$, $q(t) = 1$, and the drawdown of the water is given for the entire radius of the aquifer at the initial time, a partial differential system with two Von Neuman boundary conditions and an inital condition generates the system in Figure 2.2.

$$\begin{cases} \frac{\partial u}{\partial t} = D(\frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r}) + \beta(r,t) + s(r,t), r_{in} \leq r \leq r_{out}, t \geq 0 \\ \frac{\partial u}{\partial r}|_{r_{in}} = \alpha(t) \\ p(t)u(r_{out},t) + q(t)\frac{\partial u}{\partial r}|_{r_{out}} = \rho(t) \end{cases}$$

Figure 2.1: The Diffusion Partial differential equation with Von Neuman, Robin, and Initial conditions.

$$\begin{cases} \frac{\partial h}{\partial t} = \frac{T}{S}(\frac{\partial^2 h}{\partial r^2} + \frac{1}{r}\frac{\partial h}{\partial r}) + s(r,t) \\ \frac{\partial h}{\partial r}|_{r=\frac{1}{2}} = \frac{Q}{\pi T} \\ \frac{\partial h}{\partial r}|_{r=100} = 0 \\ h(r,0) = 0 \end{cases}$$

Figure 2.2: The system describing the drawdown water levels within an aquifer.

In the system of Figure 2.2, h, r, and t represent the drawdown height, aquifer radius, and the time in days, respectively while Q represents the daily rate of well pumping, T represent the transmissivity, and S represents the storativity of the aquifer.

This paper will analyze the system with $T = 150m^2/day$, $S = 0.2$, and $Q = 100m^3/day$ in the homogeneous form where $s(r,t) = 0$ and the nonhomogeneous form where $s(r,t) = \frac{w}{S}$ with $w = 0.003m/day$ accounting for precipitation. For both analyses, the drawdown will be found after a 15 day period.

2.2 Math Modeling and Analysis Methods

The method used to analyze the system in Figure 2.2 is a variation of the Explicit Euler finite difference method derived from the use of forward difference in time and central difference in space. Taking into account the source term required to solve the system, Equation (2.1) can be used to calculate the drawdown water level at the internal nodes of the mesh created along the radius of the aquifer where n represents a temporal node and j represents a spatial node.

$$h_j^{n+1} = (\lambda - \frac{\mu}{2r_j})h_{j-1}^n + (1-2\lambda)h_j^n + (\lambda + \frac{\mu}{2r_j})u_{j+1}^n + \Delta t s_j^n \tag{2.1}$$

By accounting for the two Von Neuman boundary conditions, Equations (2.2) and (2.3) can be used to calculate the drawdown height at the radial boundaries of each timestep.

$$h_0^{n+1} = (1-2\lambda)h_0^n + 2\lambda h_1^n - 2\Delta r\alpha(t_n) + \Delta t(\lambda - \frac{\mu}{2r_0}) \tag{2.2}$$

$$h_{N+1}^{n+1} = 2\lambda h_{N-1}^n + (1-2\lambda)h_N^n \tag{2.3}$$

Here, $\lambda = \frac{T\Delta t}{S(\Delta r)^2}$, $\mu = \frac{T\Delta t}{S\Delta r}$, and as can be seen in Figure 2.2, $\alpha(t)$ is held constant at $\frac{Q}{\pi T}$.

2.2.1 Stability

While an explicit method decreases the number of calculations and eliminates the need to solve a matrix equation to find a solution (as necessary for an implicit or Crank-Nicolson method), it can be very unreliable if the Courant-Fredrichs-Lewy (CFL) stability condition is not met. For this system, the CFL condtion is determined by

$$\frac{T}{S}\frac{\Delta t}{(\Delta r)^2} \leq \frac{1}{2}.$$

Using $\Delta t = \frac{15}{M}$ and $r = \frac{99.5}{N}$ where M is the number of temporal steps and N is the number of spatial steps, Equation (2.4) provides a means to calculate the correct number of steps necessary to produce a reliable solution.

$$\frac{N^2}{M} \leq 0.44 \tag{2.4}$$

2.3 Problem Solving Procedure & Solution

For this solution, multiple experiments were run with different values of M and N. It was found that $M = 10,000$ and $N = 60$ produced the most reliable results in consideration of the CFL stability condition and the physics of the situation based on the boundary conditions.

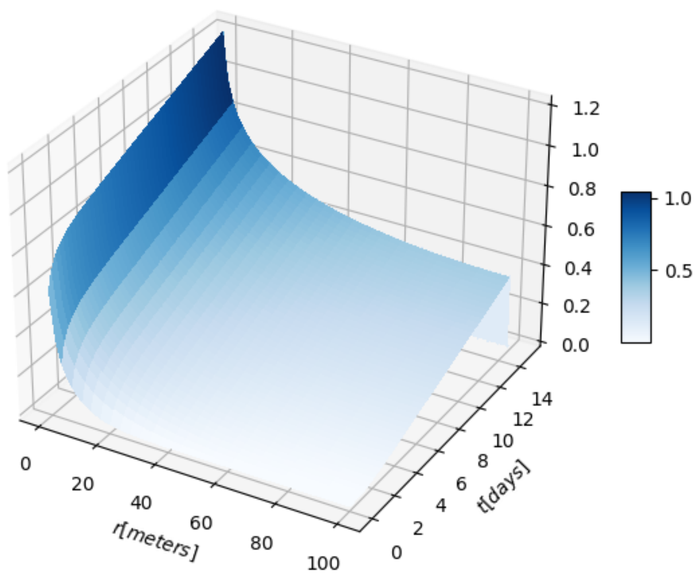

Figure 2.3: The amount of water drawn from the aquifer without a source (homogeneous).

Day	$h(r_{in})$	$h(r_{out})$
0	0.0000	0.0000
2	0.8344	0.0298
3	0.9232	0.0844
4	0.9922	0.1468
6	1.0573	0.2107
8	1.1218	0.2750
9	1.1861	0.3392
10	1.2504	0.4035
12	1.3147	0.4678
14	1.3789	0.5321
15	1.4432	0.5963

Table 2.1: The drawdown height at the inner and outer radius over time for the homogeneous solution.

Both solutions were calculated using the code in Appendix A but the the solution produced in Figure 2.4 had the source term of $s = 0.015$.

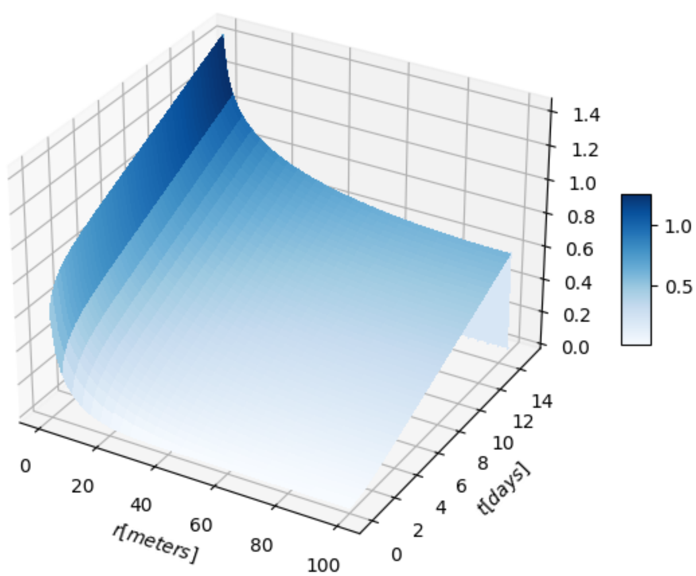

Figure 2.4: The amount of water drawn from the aquifer with a source (homogeneous).

Day	$h(r_{in})$	$h(r_{out})$
0	0.0000	0.0000
2	0.8344	0.0298
3	0.9232	0.0844
4	0.9922	0.1468
6	1.0573	0.2107
8	1.1218	0.2750
9	1.1861	0.3392
10	1.2504	0.4035
12	1.3147	0.4678
14	1.3789	0.5321
15	1.4432	0.5963

Table 2.2: The drawdown height at the inner and outer radius over time for the non-homogeneous solution.

2.4 Discussion & Conclusion

The solution to the system in Figure 2.2 informs of how many meters of water is drawn from the aquifer by day along the radial direction. Figure 2.3 shows that after 15 days, the water level drawn at the the inner radius is approximately $1.2182m$ and $0.3713m$ at the outer radius. Using the same values of N and M, the non-homogeneous solution goes through the same computational process but with the source term set to $s = \frac{w}{S} = 0.015\frac{m}{day}$ instead of zero. Because the source term is constant with respect to time and space, it is expected that over the 15 days, the difference between the drawdown level at every increment along the radius in the non-homogeneous and homogeneous solutions should be $0.225m$. This is confirmed when comparing Table 2.1 and Table 2.2 where the result at the the inner radius is approximately $1.4432m$ which is $.225m$ higher than it was in the homogeneous solution. At the outer radius, the drawdown level in the non-homogeneous solution is $0.5963m$ which is $0.26917m$ higher than that found in the homogeneous solution. These confirmed differences also lend to the reliability of the solution found using the explicit method derived from the forward difference in time and backward difference in space.

Since the rate at which water is being pumped from the well remains the same for both cases, the higher drawdown level in the non-homogeneous system indicates a higher rate at which the well is being refilled by the aquifer.

Appendix A

Code Implementations

Chapter 1: Transient Heat Conduction in a Fin

```
clear;
clc;
close all

h = 50;
w = 0.005;
Tinf = 20;
Rin = 0.002;
Rout = 0.005;

R=Rout;
T=600;

dt = 1e-4
M = T/dt;
t=[0:M]*dt;

N = 18;
dr = (Rout-Rin)/N
r=[0:N]*dr;
r = r+Rin;


rho1 = [8933;8055;8800];
Cp1 = [385;480;420];
k1 = [401;15.1;52];

T = zeros(N+1,M+1,3);

for n = 1:3
```

```
    rho = rho1(n);
    Cp = Cp1(n);
    k = k1(n);

    % Stability criterion ——————————————————————————
    dt_crit = dr*rho*Cp/(2*(k/dr+h))

    u=zeros(N+1,1);
    uu=zeros(N+1,M+1)+Tinf;


    % BC
    for i = 1:M+1
        u(i) = 20+40*(1-exp(-t(i)/120));
    end

    uu(1,1:M+1)=u;

    u = 0;

    % initial condition:
    for i=1:N+1
        u(i) = Tinf;
    end
    uu(1:N+1,1)=u;

    for j=1:M
        t1 = j*dt;
        u = update_pde_expl(u,dr,dt,t1,rho,Cp,k,Rin);
        uu(1:N+1,j+1)=u;
    end

    T(:,:,n) = uu;

end

Copper = T(:,:,1);
Steel = T(:,:,2);
Bronze = T(:,:,3);

% figure
% [tt,rr]=meshgrid(t,r);
% s = surf(tt,rr,Steel);
% s.EdgeColor = 'none';
% title('Steel')
```

```
% xlabel('time (s)')
% ylabel('Radius (m)')
% zlabel('Temperature (dc)')

figure
[tt,rr]=meshgrid(t,r);
contour(tt,rr,Steel,linspace(20,61,42),'ShowText','on')
title('Steel')
xlabel('time (s)')
ylabel('Radius (m)')

figure
hold on
plot(r,Copper(:,end),'o-')
plot(r,Steel(:,end),'o-')
plot(r,Bronze(:,end),'o-')
legend('Copper','Steel','Bronze','location','southwest')
title('600 Seconds')
xlabel('Radius (m)')
ylabel('Temperature (dC)')



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function u2 = update_pde_expl(u1,dr,dt,t,rho,Cp,k,Rin)
h = 50;
w = 0.005;
Tinf = 20;

N=length(u1);
u2=zeros(N,1);

lamda = (k/(rho*Cp))*(dt/(dr*dr));
mu = (k/(rho*Cp))*(dt/dr);
beta = (2*h*dt)/(rho*Cp*w);
S = (Tinf*dt)/(rho*Cp);

% update PDE in the internal nodes
for i=2:N-1
    r = Rin + dr*i; %The radius
    u2(i) = (lamda-(mu/(2*r)))*u1(i-1) + (lamda+(mu/(2*r)
        ))*u1(i+1) + S + (1-2*lamda-beta)*u1(i);
end
% boundary conditions:
```

```
u2(1)= 20+40*(1-exp(-t/120));
u2(N) = u2(N-2) + ((2*dr)/k)*(h*Tinf-h*u2(N-1));
end
```

Chapter 2: The Drawdown Rates of an Aquifer

```
import numpy as np

def ftcs(u1, dr, dt, D, r_in, r_out, \
    alpha, beta, p, q, rho, s):
    N = len(u1)
    u = np.zeros(N)
    mu = D*dt/dr
    nu = D*dt/dr**2

    # Updating the internal spacial nodes
    for i in range(1, N-1):
        rj = r_in + i*dr
        u[i] = (nu-.5*mu/rj)*u1[i-1] + \
            (1 - 2*nu - dt*beta)*u1[i] + \
                (nu + .5*mu/rj)*u1[i+1] + dt*s

    # inserting the boundary conditions
    u[0] = (1 - 2*nu - dt*beta)*u1[0] \
        + 2*nu*u1[1] + dt*s - \
            2*dr*alpha*(nu - .5*mu/r_in)
    u[N-1] = 2*nu*u1[N-2] + \
        (1 - 2*nu - dt*beta - \
            2*dr*p*(nu + .5*mu/r_out)/q)*u1[N-1] \
                + dt*s + 2*dr*rho*(nu + .5*mu/r_out)/q

    return u
##############################################

if __name__ == "__main__":
    T = 150 # Transmissivity m^2/day
    S = .2
    D = T/S
    Q = 100 # daily pump rate m^3/day
    w = .003 # rain added m/day
    source = w/S
    beta = 0
    p = 0
    q = 1
    rho = 0
    alpha = Q/(np.pi*T)
    t0 = 0
    tend = 15 # days
    rin = .5 # m
    rout = 100 # m
```

```
N = 60 # number steps in radial direction
M = 10000# number of time steps
dr = (rout - rin)/N
dt = tend/M

h = np.zeros(N+1)
sol = np.zeros((M+1,N+1))

# The initial condition
for i in range(N+1):
    h[i] = 0
    sol[0,i] = h[i]


# Finding the height along the radius for time period
for n in range(1,M+1):
    h = ftcs(h, dr, dt, D, rin, rout, \
        alpha, beta, p, q, rho, source)
    for i in range(0,N+1):
        sol[n,i] = h[i]
```