Control of chlorine disinfection in off-grid rainwater systems for potable use

ce 291 final project

may 2016

harry durbin

Contents

Figures

INTRODUCTION

Many rural mountain communities in Guatemala do not have a connected water system—some share rainwater cisterns, while others must walk long distances (across hilly and rocky terrain) manually carrying heavy jugs from a well. Occasionally rainwater tanks are constructed by volunteers and placed at local schools, however, these tanks lack a disinfection system so there may be bacteria in the water that can make children sick. Generally water is boiled, but this is inconvenient and still may cause illness if a child drinks raw water before boiling, or washes an open wound. This is concerning because one in five deaths in children under five is caused by diarrhea. This may be effectively reduced using disinfection [5].

In the US, an Ultraviolet (UV) system is commonly used for disinfection, yet requires constant energy and may cost thousands of dollars. The energy supply and cost is not affordable or feasible for these rural mountain communities lacking electricity. Chlorination is more economic for disinfection, costing only $1 per month when added manually [10]. However, one concern is how much chlorine to add to the system: it is important to maintain a chlorine concentration in the range of 0.5 to 5ppm because too much may form cancerous trihalomethanes, while too little may not ensure disinfection [3]. This report explores options to model and systematically control the chlorine disinfection in off-grid rainwater tanks for potable water use to ensure compliant levels.

LITERATURE REVIEW

The following section summarizes research on rainwater disinfection and studies by others.

Rainwater Collection & Disinfection

Two methods may be used to disinfect a rainwater tank:

1. manually adding periodically to the rainwater tank (recommended to dilute a portion of the concentrated chlorine in a separate container)
2. automatically dosing based on flowrate exiting the tank, just prior to use (however the water must first have an approximate 2 minute contact time)

Figure 1. Rainwater harvesting system

Option 1, manual (or automatic) addition to the rainwater tank/cistern, is more economic, but more difficult to control due to varying tank inflows and outflows over time—it is difficult to know precisely when and how much chlorine to add. According to guidelines on rainwater tanks, unscented bleach should be added to the tank each month or as often as each week during a rainy period, with the objective to keep at least 1 ppm in the tank. [6]

Option 2, automatic injection relative to flow out of the tank is more expensive but provides a fully automated system and allows for easier control and modeling. Bleach can be added through a metering pump and set to a fixed value. [2] However, in this method, the pipe length and diameter must be sized to provide sufficient contact time for the chlorine to react and kill bacteria in the water.

PROBLEM INVESTIGATED

The following sections discusses the objective of system modeling, and describes relevant methods and equations.

Objective

The objective is to first investigate options on modeling and controlling chlorine disinfection in rainwater harvesting systems for potable water use, particularly for off-grid, rural communities in developing countries, such as Guatemala. After investigation, a 1-Dimension simulation in Matlab will be developed to model chlorine concentration and control.

System Modelling

The following section describes the system, equations, and frames to create a mathematically model.

ADVECTION - DIFFUSION - REACTION (ADR) EQUATION

The advection-diffusion-reaction (ADR) equation, also known as convection-diffusion-reaction, is a parabolic PDE that can be used to model the evolution of chlorine concentrations. Generally the ADR equation cannot be solved analytically, however techniques have been developed for approximations or exact solutions for some ranges [1]. The ADR equation is of the form:

$$u_t = [D * u_{xx}] + [c * u_t] + [r * \mathrm{u}]$$

Where,

- $u(x,t) = chlorine\ concentration\ at\ point\ x, time\ t$ [ppm]
- $[D * u_{xx}] = diffusion\ term\ (Diffusivity, D = 1e^{-9}\frac{m^2}{s} for\ chlorine)$
- $[c * u_t] = advection/convection\ term\ (c = convective\ velocity)$
- $[r * u] = reaction\ term\ (decay\ constant, r = 1e - 5\ \ [s^{-1}]\ for\ chlorine)$

OPTION #1 – ADDING CHLORINE TO TANK

The first option, to add chlorine directly into the tank is much more complex to model and control as there are variable flows in and out. Due to these variable flows, the domain of the problem is changing over time. Assuming sensors are used to measure the flow in or out of the tank and the tank water level, this method can be used to develop a more precise model of when to add chlorine for off-grid rainwater systems that have constraints of limits energy and finances. The dosing could be done manually or automatically in the tank. If done manually, an electronic display could alert users how much chlorine should be added each week.

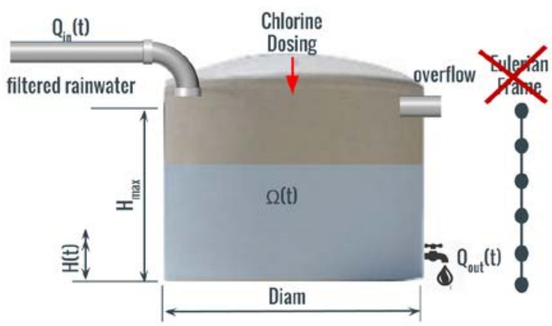

Figure 2. Rainwater tank system schematic in 2-dimensions

A simulation of rainwater levels in a tank using Honduras weather data is presented below. The water surface level varies over time with the constraint that height cannot exceed capacity of the tank, and cannot go below zero. The change in height of the tank can be calculated with this formula:

$$\frac{d}{dt}H(t) = \frac{1}{A(t)}(Q_{in}(t) - Q_{out}(t))$$

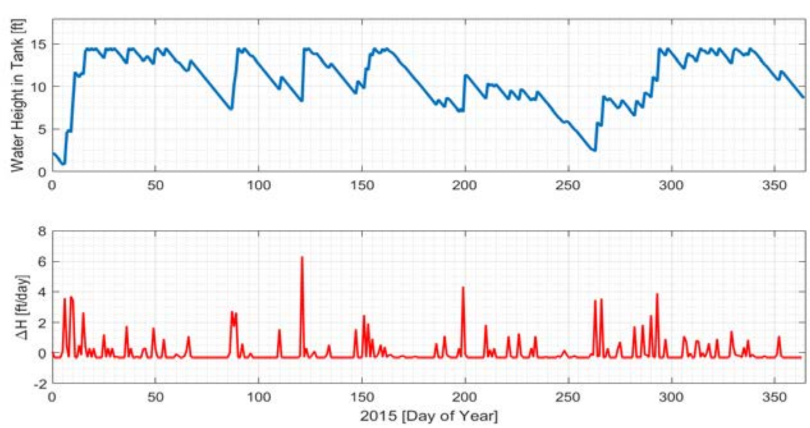

Figure 3. Water surface height variation in rainwater tank in Honduras

Arbitrary Lagrangian Eulerian (ALE) Method

Both Eulerian (fixed in space) and Lagrangian (moving with a particle) coordinate frames have difficulties handling this type of problem with moving boundaries. For this problem, the Arbitrary Lagrangian Eulerian (ALE) Method is a way to handle calculations with a changing domain size by assigning a mesh that moves at an arbitrary velocity [5]—it is a hybrid method of Eulerian and and Lagrangian coordinate frames. Using the ALE method, the domain velocity is set equal to the velocity of the free water surface level moving up and down in the z-axis:

$$v^f(t) = \frac{d}{dt}H(t) = v^d(t)$$

$v^f(t)$ is the free velocity of the water surface

$v^d(t)$ is the domain velocity of the water surface

Meanwhile the fluid velocity, v, is determined from the Navier-Stokes hydrodynamics equations. Flows are caused by momentum of water as it flows into the tank (either falling from above or entering beneath the water surface), and by flows existing the tank. In cylindrical coordinates, the Navier-Stokes equation is:

$$\frac{1}{r} * \frac{\partial(rv_r)}{\partial r} + \frac{1}{r} * \frac{\partial(v_\theta)}{\partial \theta} + \frac{\partial(v_z)}{\partial z}$$

The DCA and Navier-Stokes equations are coupled using the convective velocity, c, to solve and model the concentration evolution in the tank: $c = v - v^d(t)$.

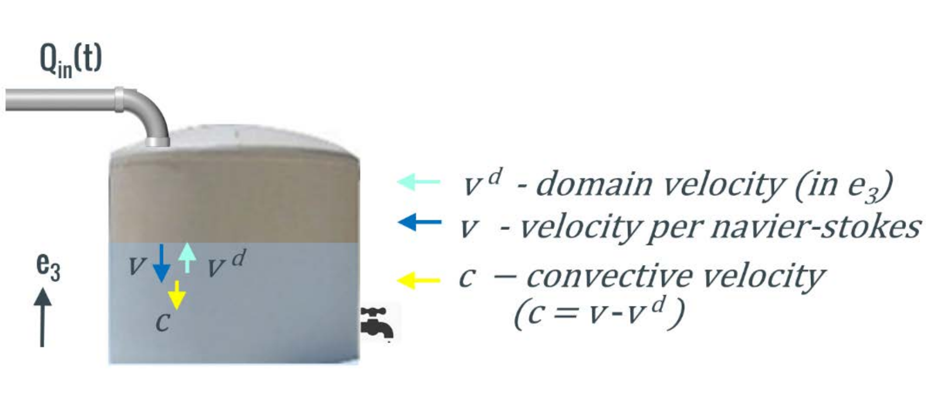

Figure 4. Diagram of convective velocity when using ALE

This method of numerically modelling chlorine concentration in a large water storage tank (with varying in and outflows) was prepared and discussed by Codina in "*Numerical modelling of chlorine concentration in water storage tanks.*" The model was intended for tanks that store water after treatment at a water treatment plant and prior to distribution because the water must have a correct chlorine concentration when it enters the distribution network. This tank system closely resembles a smaller-scale rainwater tank system, with variable inflows and outflows that change the water level over time in the tank. A numerical model was prepared to simulate the evolution of chlorine concentrations over time in the tank, using falling jets on the water surface, and hydrodynamics (modeling with Navier-Stokes equations) coupled with the advection-diffusion-reaction equation) [4]. This research indicates that the ALE method can be used to predict concentration over time in the tank. Results generated by Codina in his numerical model are shown:

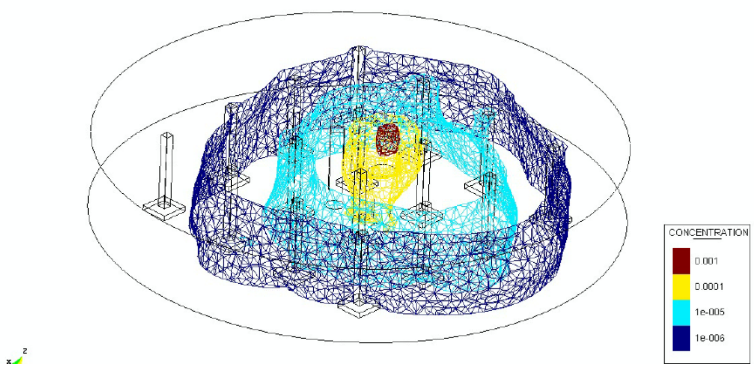

Figure 5. Chlorine modeling in water tank using ALE (Codina)

OPTION #2 – INJECTING CHLORINE IN PIPELINE

The length and the diameter of the pipeline should be sized to ensure sufficient contact time (time for the chlorine to react with organisms). The contact time need is a matter of opinion with different guidelines giving varying minimum requirements. Some guidelines state only 2-5 minutes are needed, while others state 30 minutes are needed. The case of short contact time is much simpler because it means less volume of water will remain stagnant in the pipe over long periods of time. On the other hand, if 30 minutes contact time is desired, water may sit in the pipe for possible 24-48 hours causing the chlorine levels to decay to near zero. A minimum of 0.5 ppm is preferred to ensure disinfection. The flow rate is based on a typical flow from a household sink. An illustration of the pipeline is shown below. The boundary conditions are Dirichlet, with only u_0 being controlled.

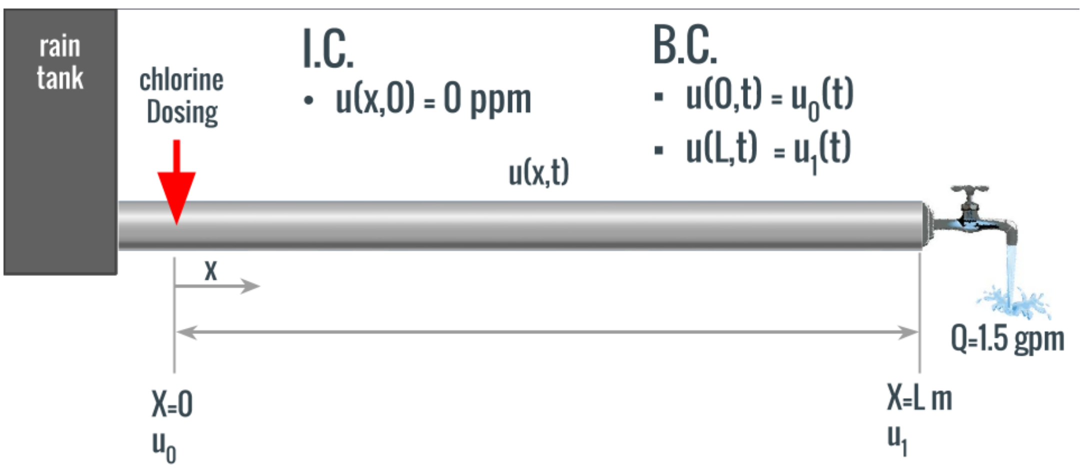

Figure 6. Pipe schematic in 1-dimension

Simulation

A model for this system was developed in Matlab assuming a contact time of 5 minutes. A diameter of 1-1/2" and flow of 1.5 gpm was assumed. A length of 25 meters was set to ensure contact time. Results are presented in figures below:

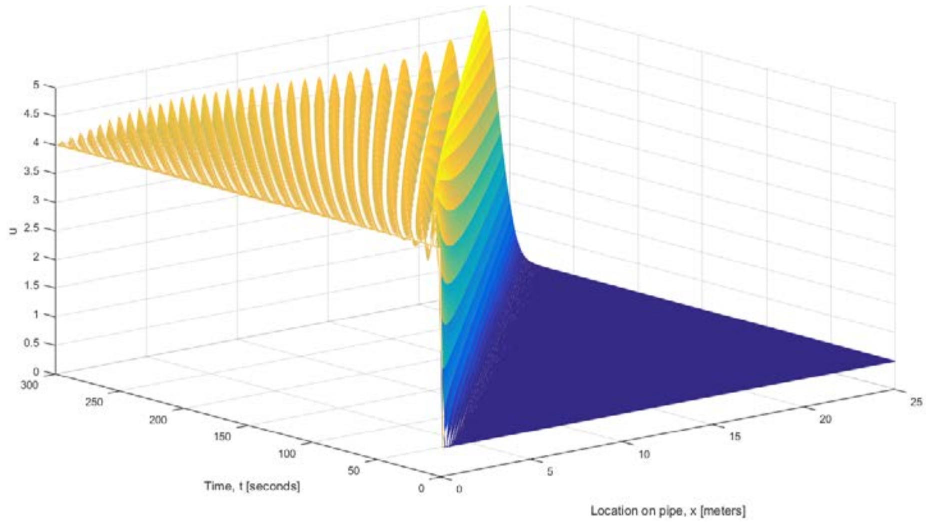

Figure 7. Mesh plot of concentration in pipe over time while dosing (5 minutes)

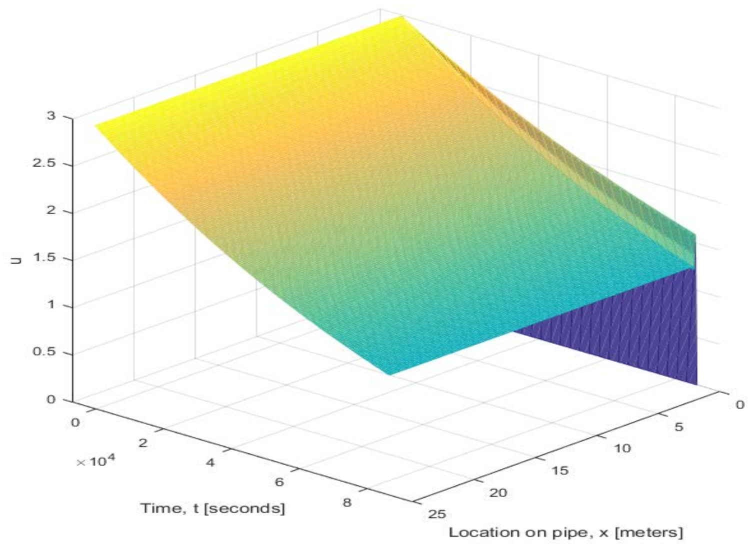

Figure 8. Mesh plot of concentration decay in pipe over time while stagnant (24 hours)

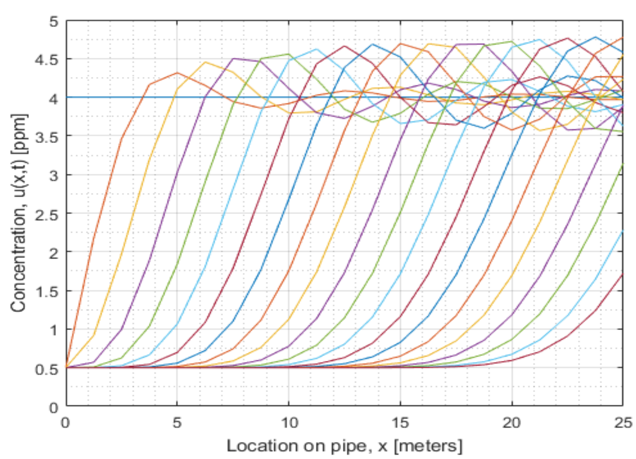

Figure 9. Chlorination breakpoints during dosing

As this model yielded a trivial solution (chlorine could be trivially controlled in this scenario when the piping is so short), a more advanced model was developed based on an assumed flow schedule:

Time Hour	Time Stagnant (hours)	Consumption (gallons)
8 am	-	3
-	4	
12 pm	-	3
-	5	
5 pm	-	3
-	4	
9 pm	-	3
-	11	
8 am	-	3
Total	24 hours	15 gallons

This schedule has average flow of 15 gallons per 24 hours. This average flow was assumed to be constant throughout the day in lieu of a model with on and off, varying flows. Again assuming a 1-1/2" diameter pipe, this flow equates to 0.5 cm/sec. In addition, the length of pipeline was increase to 50 meters. Results are presented below:

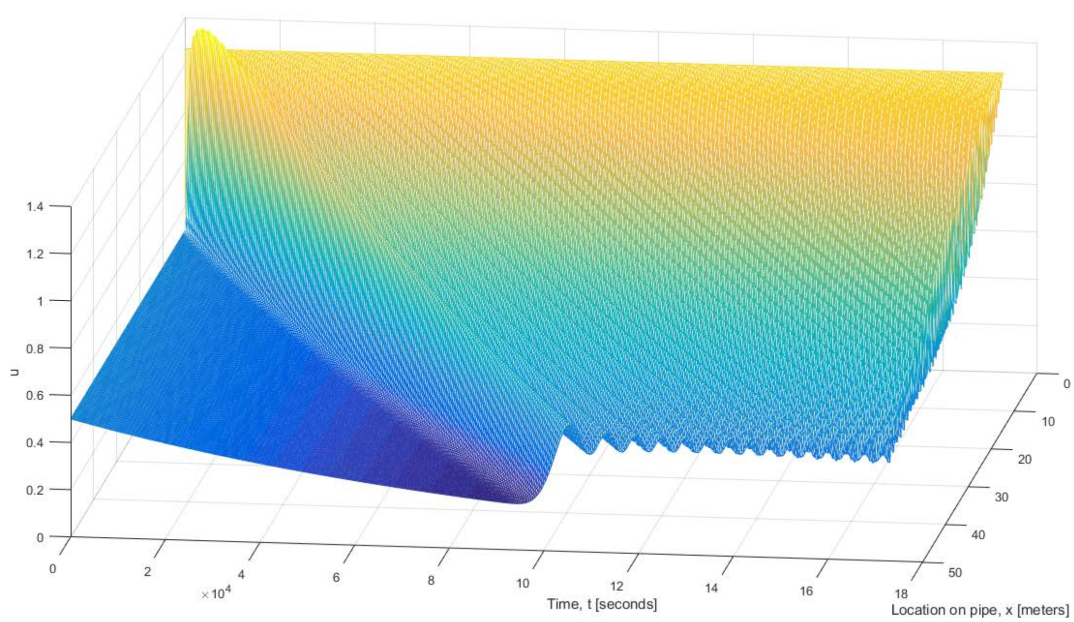

Figure 10. Mesh plot of concentration in pipe over time based on flow schedule (2 days)

Feedback Control

For this more complicated flow scenario, it may be difficult to ensure concentration at the end point is at a desire level. A proportional feedback controller can be used to control the chlorine dosage as indicated in the block diagram below:

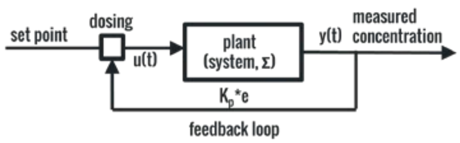

Figure 11. Block diagram of proportional feedback control loop

The equation for the proportional controller is:

$u(t) = k_p * e(t)$

Where,

- $k_p = proportional\ gain$ = 1
- $e = error = [SP - y(t)]$
- $y(t) = measured\ concentration\ at\ faucet$
- $SP = set\ point\ = 0.5\ ppm$

Recursively, the boundary condition chlorine dosage is updated by applying a gain factor equivalent to the proportional gain constant times the error (difference in output from the desire output/set point). This was performed with the results presented below:

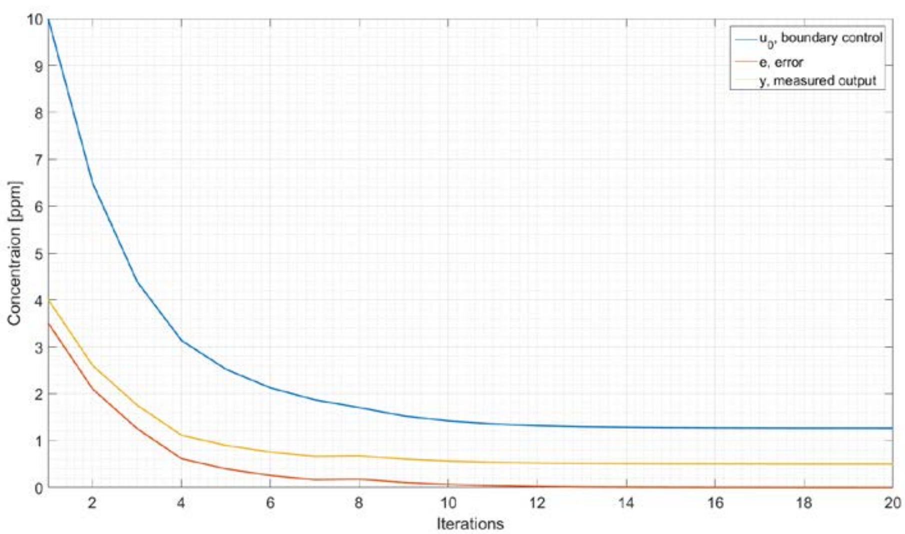

Figure 12. Results of proportional feedback controller for chlorine dosage

CONCLUSION

The investigation into modeling and control of off-grid rainwater tank chlorine disinfection analyzed two options to add chlorine—into the tank or pipeline.

Adding chlorine to the tank is more complex to model because it requires a coupling with hydro-dynamic equations in 2 or 3-dimensions, but is possible as demonstrated by Codina. This method could theoretically be used to create a model, which uses flow meter and level sensor data to precisely predict when chlorine should be added (this could be particularly useful during heavy rain periods).

Adding to pipeline is easier to model and control, but is more expensive, and becomes more complex if long, 30 minute contact times are desired. A simulation was presented for a short contact time of 5 minutes, but has room for improvement by incorporating a non-continuous flow schedule, with periods where the water is stagnant in the pipe long intervals, while chlorine levels decay. A proportional feedback controller is used to control the concentration at a set point of 0.5 ppm.

REFERENCES

[1] Iber, Dagmar. Computational Biology Group (CoBi). Numerical Solution of reaction-diffusion problems.

[2] McPherson. Controlling the additions of Chlorine into a Process, date unknown.

[3] MassDEP. Recommended Action Levels for Free Chlorine for Disinfectants Chlorine and Chloramines, April 2008.

[4] Codina, etal. Numerical modelling of chlorine concentration in water storage tanks, International Center of Numeric Methods in Engineering, 2014.

[5] Kremer, etal. Trickle down: Diffusion of chlorine for drinking water treatment in Kenya. June 2008

[6] Franz, Portial. A guide for rainwater catchment systems in the pacific.

[7] Fisher, Ian. Evaluation of Suitable chlorine bulk-decay models for water distribution systems. Water Research 45, p 4896-4908. 2011.

[8] Monteiro, L. Modeling of chlorine decay in drinking water supply systems using EPANET MSX. Procedia Engineering 70, p. 1192-1200. 2014.

[9] Cunliffe, David. Guidance on the use of rainwater tanks. National Environment Health Forum Monographs. 1998

[10] TWDB—Texas Water Development Board. The Texas Manual on Rainwater Harvesting. Third Edition. 2005

[11] Chrysafinos, Konstantinos. Lagrangian and Moving Mesh Methods for the Convection Diffusion Equations.

[12] Souli and Benson. Arbitrary Lagrangian Eulerian and Fluid Structure Interaction – Numerical SImulation. ISTE Ltd and Wiley. 2010.

APPENDIX: MATLAB CODE

```
%% CE 291 - Control and Optimization of Distributed Parameter Systems
%   Chlorine Disinfection
%   Harrison Durbin, SID 26951511
%   Prof. Gomes
% pipe_calculations_rev_0.m
clear; close all;

%% calculating pipe length from tank to faucet
% two flow scenarios

% scenario 1: ensure 5 minutes of contact time
d_pipe = 1.5; % inch
d_pipe = d_pipe*25.4*1e-3; % m
A_pipe = pi*(d_pipe^2)/4; % m2
t_pipe = 5; % min
t_pipe = t_pipe*60; % sec
Q_pipe = 1.5; % gpm max for bathroom faucets
Q_pipe = Q_pipe * 6.30901966667e-5; % m3/s
v_pipe = Q_pipe/A_pipe; % m/s
L_pipe = v_pipe*t_pipe; % m
L_pipe = L_pipe * 3.28084 % ft
V_pipe = L_pipe * A_pipe % m3
V_pipe = V_pipe * 264 % gal - 264 gal/m3

% scenario 2: assume fixed flow schedule over 24 hours
d_pipe2 = 1.5; % inch
d_pipe2 = d_pipe2*25.4*1e-3; % m
A_pipe2 = pi*(d_pipe2^2)/4; % m2
t_pipe2 = 60*24*3; % min
t_pipe2 = t_pipe2*60; % sec
Q_pipe2 = 15/24/60; % gpm max for bathroom faucets
Q_pipe2 = Q_pipe2 * 6.30901966667e-5; % m3/s
v_pipe2 = Q_pipe2/A_pipe2; % m/s
L_pipe2 = 100; % m
V_pipe2 = L_pipe2 * A_pipe2 % m3
V_pipe2 = V_pipe2* 264 % gal - 264 gal/m3

%%  Proportional controller code
iterations = 20;
y = zeros(iterations,1);
new_u0 = zeros(iterations+1,1);
e = zeros(iterations,1);
global dosage;
dosage = 10; % set initial dosage guess
new_u0(1) = dosage;
Y = pipe_model_rev_0();
K = 1;
set_point = 0.5;
YY = zeros(iterations+1,1);
YY(1) = Y;

for i = (1:iterations)
    y(i) = Y(i);
    e(i) = (y(i)-set_point);
```

```
    new_u0(i+1) = new_u0(i)-K*e(i)+0.001;
    global dosage;
    dosage = new_u0(i+1);
    Y(i+1) = pipe_model_rev_0();
End

%% plot P controller results
plot(new_u0,'LineWidth',2);
hold on
plot(e,'LineWidth',2);
plot(y,'LineWidth',2);
axis([1, 20, 0, 10]);
grid on
grid minor
legend('u_{0}, boundary control', 'e, error', 'y, measured output')
xlabel('Iterations')
ylabel('Concentraion [ppm]')
set(gca,'FontSize',20);
```

```
function [final_u] = pipe_model_rev_0

global dosage;

m = 0;

x0=0;
xf=50;
Nx=201;
t0=0;
tf=60*60*24*2;
Nt=401;
x=linspace(x0,xf,Nx);
t=linspace(t0,tf,Nt);


sol = pdepe(m,@pdex1pde,@pdex1ic,@pdex1bc,x,t);
% Extract the first solution component as u.
u = sol(:,:,1);
final_u = u(end,end);

% A surface plot is often a good way to study a solution.
figure(5);
mesh(x,t,u)
xlabel('Location on pipe, x [meters]')
ylabel('Time, t [seconds]')
zlabel('u')

% A solution profile can also be illuminating.
figure(1);
plot(x,u)
xlabel('Location on pipe, x [meters]')
ylabel('Concentration, u(x,t) [ppm]')
grid minor
grid on

% A solution profile can also be illuminating.
figure(2)
plot(x,u(end,:))
axis([0, 100, 0, 7]);
grid on
grid minor
xlabel('Location on pipe, x [meters]')
ylabel('Time, t [seconds]')

% nFrames = length(t)-2;
%
% % Preallocate movie structure.
% M(1:nFrames) = struct('cdata', [],...
%                          'colormap', []);
% % figure(3)
% G=plot(x,u(10,:),'erase','xor');
% axis([0, 100, 0, 7]);
% xlabel('location along pipe, x [meters]'); ylabel('chlorine conc, u(x,t_k)
[ppm]');
% grid on
```

```
% grid minor
% count=1;
% for k = 2:length(t)
% %     title('');
%     set(G,'xdata',x,'ydata',u(k,:));
%     M(count)=getframe;
%     count=count+1;
% %     pause(0.1);
% end
% numtimes=0;
% fps=1;
% movie(M,numtimes,fps)
% movie2avi(M,'C:\Users\Harrison\Desktop\animation1.avi');

% --------------------------------------------------------------
function [c,f,s] = pdex1pde(x,t,u,DuDx)
% a = 0; % m/s advection
% a = (0.083*100); % m/s advection
a = 5e-4; % m/s
r = (1e-5); % [1/s] reaction rate coefficient
D = (1.25e-9); % [m2/s] diffusivity of chlorine
c = 1;
f = D*DuDx;
s = -a*DuDx-r*u;

% --------------------------------------------------------------
function u0 = pdex1ic(x)
% u0 = 4;
u0 = 0.5;

% --------------------------------------------------------------
function [pl,ql,pr,qr] = pdex1bc(xl,ul,xr,ur,t)
global dosage;
pl = ul-dosage;
% pl = ul-4;
% pl = ul-0;
ql = 0;
pr = 0; %
qr = 1;
```