EPANET WQ Tests | BWFLnet_Sept2021

Prepared by: Yuanfei Wang

Date: 17 June 2022

```
clc
clear
close all
```

Model Setup and Initialisation

Initialisation of EPANET Matlab toolkit and loading of .net file.

```
epanet_path = 'E:\Program Files\MATLAB\R2021b\toolbox\epanet\EPANET-Matlab-Toolkit-master';
net_id = 'BWFLnet_Sept2021';
run( [epanet_path,'\start_toolkit'])
```

```
EPANET-MATLAB Toolkit Paths Loaded.
```

```
net = epanet([net_id,'.inp']);
```

```
EPANET version {20200} loaded (EMT version {v2.2.0}).
Loading File "BWFLnet_Sept2021.inp"...
Input File "BWFLnet_Sept2021.inp" loaded sucessfuly.
```

Load network data and setup general simulation parameters.

```
% Save element count variables
nt = net.getTimeSimulationDuration./3600; % number of time steps in hours
nn = net.NodeJunctionCount;
n0 = net.NodeReservoirCount;
np = net.LinkCount; % count of pipes and valves
D = net.getLinkDiameter';

% Save node index vectors
Reservoir_Idx = net.getNodeReservoirIndex;
Junction_Idx = net.getNodeJunctionIndex;

% Initialise EPANET simulation type
net.setQualityType('Chlorine','mg/L');
net.setNodeSourceQuality(1:nn+n0,zeros(nn+n0,1)); % set all node source quality to zero

% Specify hydraulic and quality time steps (in seconds)
net.setTimeQualityStep(15*60); % 15 minute quality time steps
net.setTimeHydraulicStep(15*60); % 15 minute hydraulic time steps
days = 7; % simulation duration in days; change this variable to change overall simualtion time
net.setTimeSimulationDuration(days*(nt*3600)); % load modified simulation duration from initial
```

Assign chlorine decay parameters.

```
% Link bulk reaction coefficients
lambda = 0.5*ones(np,1); % units of days^-1
net.setLinkBulkReactionCoeff(1:np, -lambda);
net.setOptionsPipeBulkReactionOrder(1);

% Link wall decay reaction coefficients... apply by pipe size for S&G?
theta = ones(np,1); % units of m/day
for k = 1:np
    if D(k) <= 75
        theta(k) = 0.25;
    elseif D(k) > 75 && D(k) <= 150
        theta(k) = 0.175;
    elseif D(k) > 150 && D(k) <= 250
        theta(k) = 0.1;
    else
        theta(k) = 0.05;
    end
end

net.setLinkWallReactionCoeff(1:np, -theta);
net.setOptionsPipeWallReactionOrder(1);

% Initial concentrations at nodes (mg/L)
c0 = zeros(nn+n0,nt);
net.setNodeInitialQuality(net.NodeIndex,c0);

% Assign source chlorine with time varying pattern
cext = ones(n0,4*days*nt);

% Constant source concentration for n0 reservoirs
% cext = cext*0.25;

% Varying source concentration by source element
source_c = [0.15, 0.25];
```

Change the chlorine pattern, whether a spike set or not.

```
% for j = 1:n0
%     cext(j,:) = source_c(j)*cext(j,:);
% end

% Varying source concentration by source element with
% concentration spike (pulse) at t = 12
t = 144;
for i = 1:4*days*nt
    for j = 1:n0
        if i == t
            cext(j,i) = 1*cext(j,i); % Pulse of 1 mg/L
        else
            cext(j,i) = source_c(j)*cext(j,i);
        end
    end
end

base_cext = ones(n0,1);
pattern_cext = cext./(base_cext*ones(1,size(cext,2))); % extend vector over nt columns
```

```
% For loop to assign new patterns to source contrations at reservoirs and
% source type
for i=1:n0
    patternId = sprintf('Res_C_%d',i);
    net.addPattern(patternId,pattern_cext(i,:));
    net.setNodeSourcePatternIndex(net.NodeReservoirIndex(i),net.getPatternIndex(patternId));
    net.setNodeSourceQuality(net.NodeReservoirIndex(i),base_cext(i));
    net.setNodeSourceType(net.NodeReservoirIndex(i),'CONCEN');
end
```

Results from EPANET Simulation

Simulate hydraulic and quality analyses using EPANET's solvers.

```
hydraulic_res = net.getComputedHydraulicTimeSeries;
quality_res = net.getComputedQualityTimeSeries;

% Assign hydraulic results to network elements
h = hydraulic_res.Head(1:nt*days,1:nn).';
q = 1e-3*hydraulic_res.Flow(1:nt*days,:).';

% Assign quality results to network elements
c_nodes = quality_res.NodeQuality';
c_pipes = quality_res.LinkQuality';
```

Graph Creation

Use graph theory to plot network connectivity and spatial coordinates.

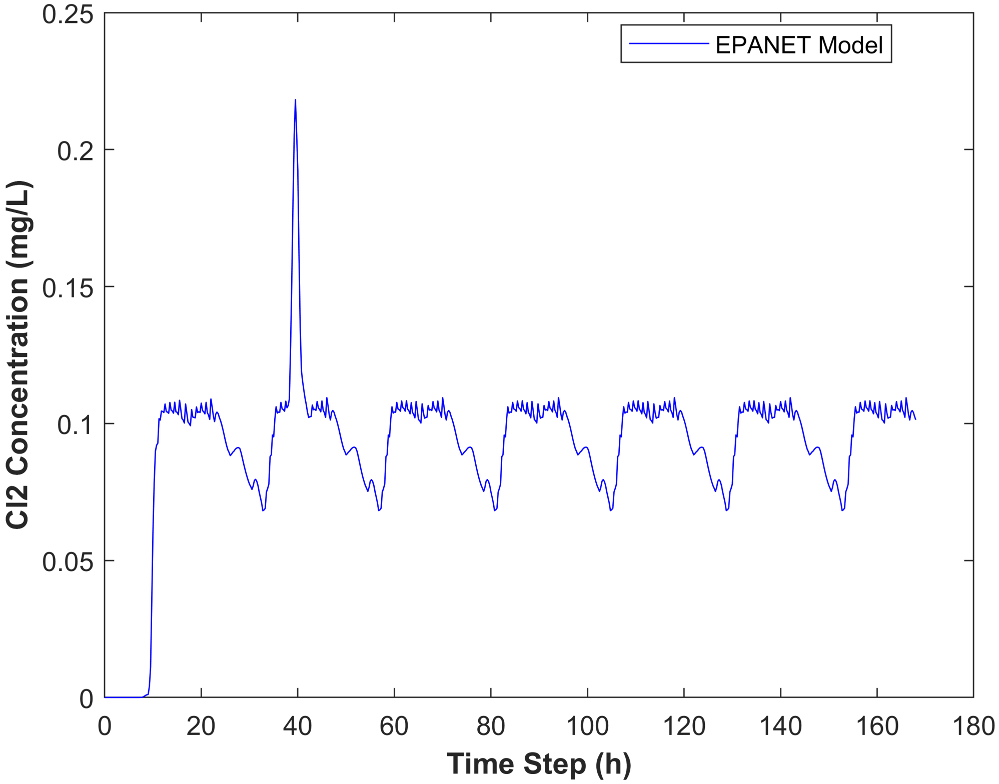
```
% Create A12 and A10 incidence matrices
LinkNodesList = net.getLinkNodesIndex;

A = zeros(np,nn+n0);
for k=1:np
    i = LinkNodesList(k,1);
    j = LinkNodesList(k,2);
    A(k,i) = -1;
    A(k,j) = 1;
end

A12 = A(:,Junction_Idx);
A10 = A(:,Reservoir_Idx);
A12 = sparse(A12);
A10 = sparse(A10);

% Obstain node XY and elevation information
XY = zeros(nn+n0,2);
XY(:,1) = net.getNodeCoordinates{1};
XY(:,2) = net.getNodeCoordinates{2};
elev = double(net.getNodeElevations(Junction_Idx))';
```

```
% Create adjacency matrix
A = [A12,A10];
AdjA = sparse(size(A,2),size(A,2));

for k = 1:size(A,1)
    node_in = find(A(k,:) == -1);
    node_out = find(A(k,:) == 1);
    AdjA(node_in,node_out) = 1;
    AdjA(node_out,node_in) = 1;
end
gr = graph(AdjA);
```

Results Plotting

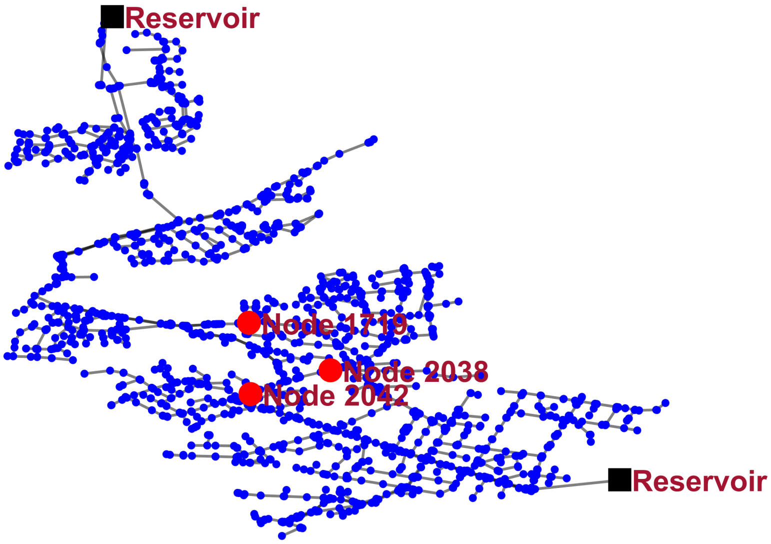
Plot the water age graph.

```
% Randomly select one node
figure,
plot(0:0.25:days*24,c_nodes(3,:)','-b')

xlabel('Time Step (h)','FontWeight','bold')
ylabel('Cl2 Concentration (mg/L)','FontWeight','bold')
legend('EPANET Model','location','best')
```

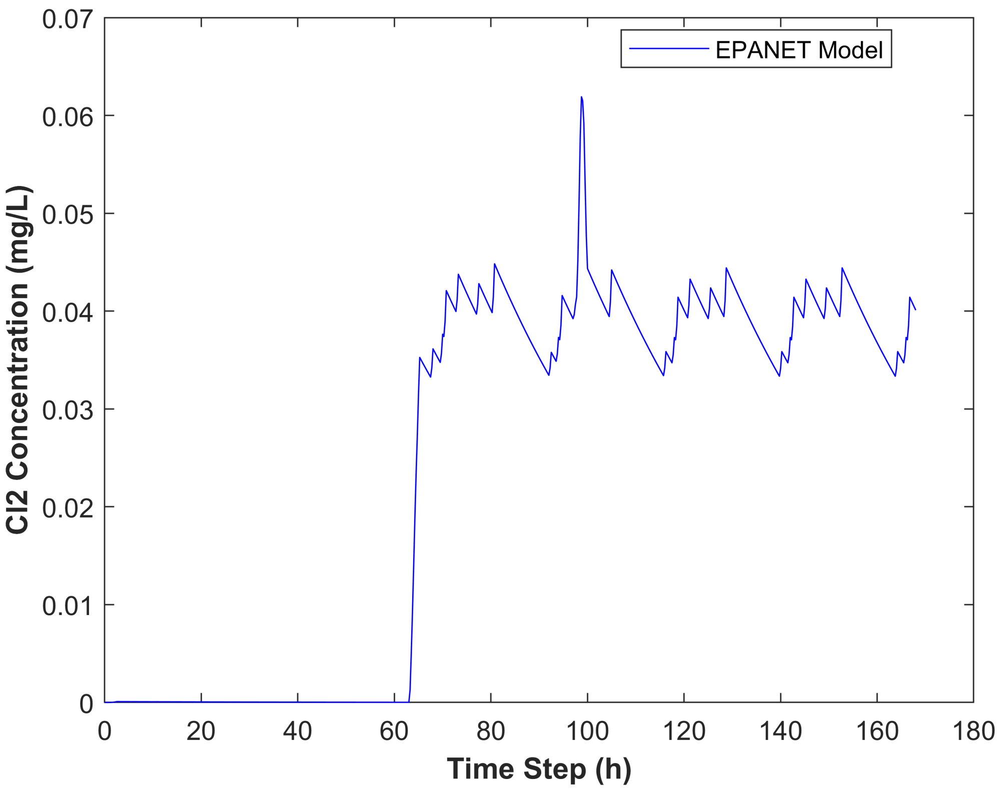


```
% Find the nodes with zero demand
c_limit = 0.01; % if the maximum chlorine concentration is below 0.01, we assume there is no de
Junc_zero = [];
for n = 1:size(c_nodes,1)
```

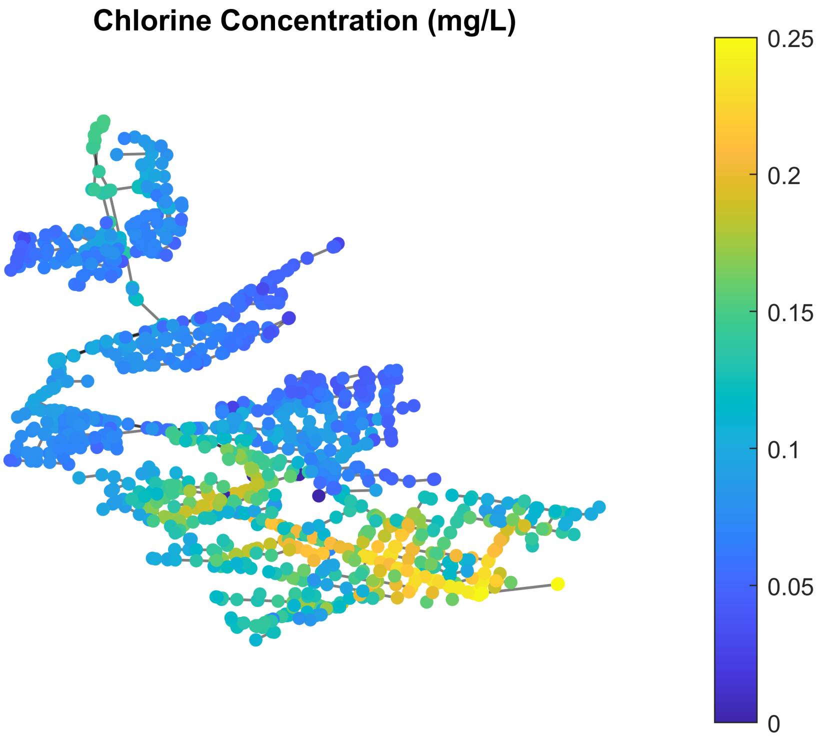
```
    c_max = max(c_nodes(n,:));
    if c_max < c_limit
        Junc_zero = [Junc_zero, n];
    end
end

% Plot of network with zero demand junctions for visualisation
for i = 1:length(Junc_zero)
    Junct_zero_Name{i} = sprintf('Node %d',Junc_zero(i));
end

figure
p1 = plot(gr);
p1.XData = XY(:,1);
p1.YData = XY(:,2);
p1.LineWidth = 1;
p1.EdgeColor = 'k';
p1.MarkerSize = 2;
p1.NodeColor = 'b';
p1.NodeLabel = '';
highlight(p1,Reservoir_Idx,'NodeColor','k','Marker','s','MarkerSize',10);
highlight(p1,Junc_zero,'NodeColor','r','MarkerSize',8);
labelnode(p1,Junc_zero,Junct_zero_Name);
labelnode(p1,Reservoir_Idx,{'Reservoir','Reservoir'});
p1.NodeFontSize = 11;
p1.NodeLabelColor = [0.6350 0.0780 0.1840];
p1.NodeFontWeight = 'bold';
axis('off')
```



```
% Find the longest water age
Junc_age = zeros(size(c_nodes,1),1);
for m = 1:size(c_nodes,1)
    for tt = 1:size(c_nodes,2)
        if c_nodes(m,tt) >= c_limit
            Junc_age(m) = tt;
            break
        else
            Junc_age(m) = 0;
        end
    end
end

max_age = max(Junc_age);
max_nodeIndex = find(Junc_age ==  max_age);

% Plot the node with largest water age
figure,
plot(0:0.25:days*24,c_nodes(max_nodeIndex,:)','-b')

xlabel('Time Step (h)','FontWeight','bold')
ylabel('Cl2 Concentration (mg/L)','FontWeight','bold')
legend('EPANET Model','location','best')
```

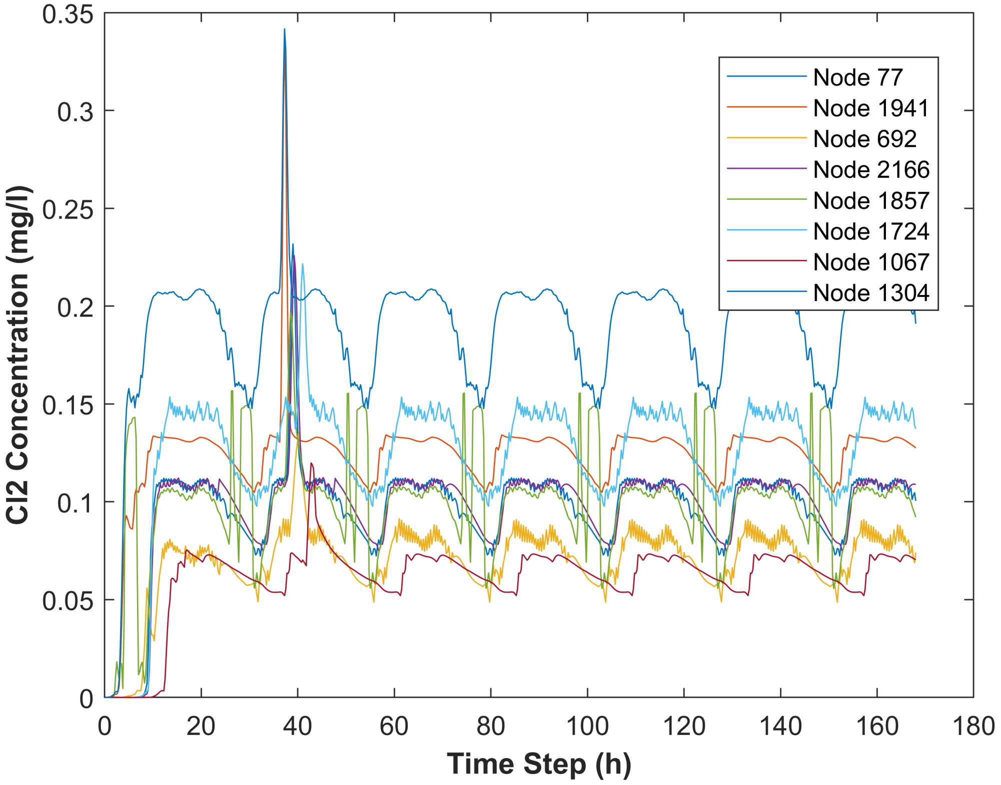


```
% Plot of Cl2 concentration distributed across network
% Select period to show results
i = 225;
figure
p2 = plot(gr);
p2.XData = XY(:,1);
p2.YData = XY(:,2);
p2.LineWidth = 1;
p2.EdgeColor = 'k';
p2.MarkerSize = 4;
p2.NodeLabel = '';

p2.NodeCData = c_nodes(:,i);

colorbar
axis('off')
title('Chlorine Concentration (mg/L)')
```



Plot time series of chlorine concentration for select junctions.

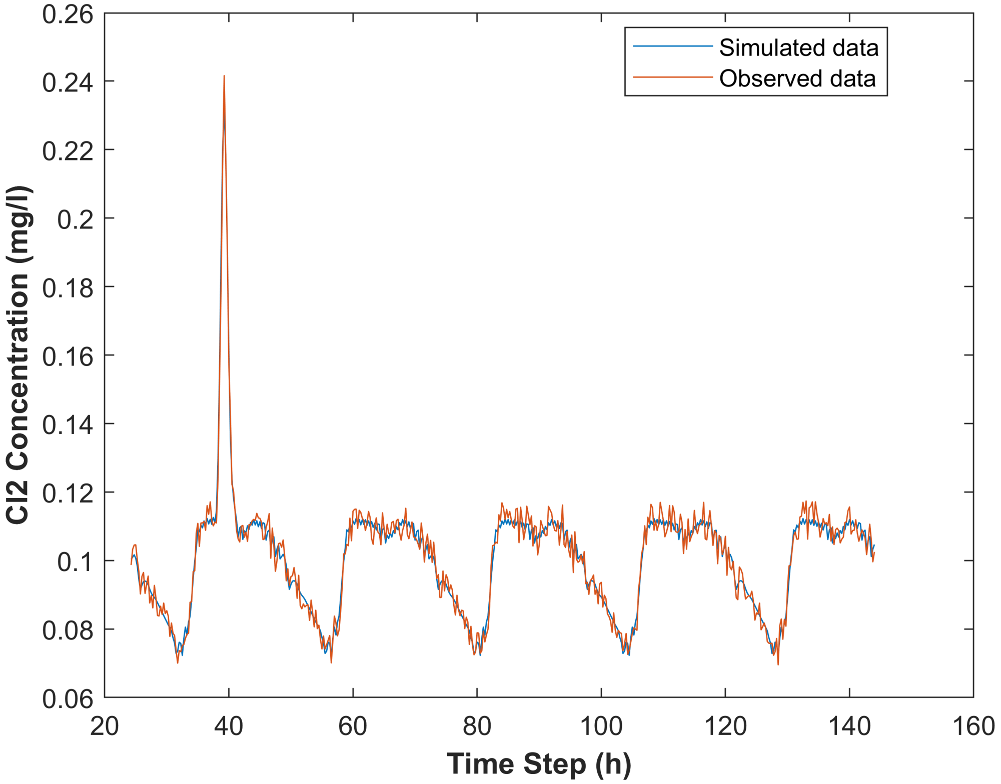
```
% Select junctions below
Junc_select = [77, 1941, 692, 2166, 1857, 1724, 1067, 1304];
for i = 1:length(Junc_select)
    Junct_Name{i} = sprintf('Node %d',Junc_select(i));
end
figure
for ii = 1:length(Junc_select)
        a = Junc_select(ii);
        plot(0:0.25:days*24,c_nodes(a,:)','-')
    hold on
end

legend(Junct_Name,'location','best');
xlabel('Time Step (h)','fontweight','bold')
ylabel('Cl2 Concentration (mg/l)','fontweight','bold')
```



Calibration

Create a random matrix

```
% Create random matrix
qual_sim = c_nodes(Junc_select,97:(days-1)*96); % fisrt day for stabilisation, last day for val
qual_rand = ((rand(size(qual_sim,1),size(qual_sim,2)))+9.5)./10;
qual_obs = qual_sim.*qual_rand;
figure
plot(24.25:0.25:(days-1)*24,qual_sim(1,:)','-')
hold on
plot(24.25:0.25:(days-1)*24,qual_obs(1,:)','-')
legend([{'Simulated data'},{'Observed data'}],'location','best');
xlabel('Time Step (h)','fontweight','bold')
ylabel('Cl2 Concentration (mg/l)','fontweight','bold')
```


Simulated data
Observed data
Cl2 Concentration (mg/l)
0.26
0.24
0.22
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
20
40
60
80
100
120
140
160
Time Step (h)