function obj=Wetlands\_system\_culverts(number\_pipes,~)

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

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

%Enter data here (English units)

%NW = Number of wetlands connected to a single drain

NW = 4;

%d1, d2, d .. culvert diameters in inches (6",12", 18", 24", 36")

%Initial diameters in inches

g = 32.2;

TW = 0.0;

%Drainage time for 95% of the water volume

load drainage\_time\_input drainage\_time\_input;

% drainage\_time = 3; %Drainage time in hours

depth\_Sediments=0.492; %This is the water depth in feet from the bottom of the wetland until the invert of the 5-way intake %Note that 0.15 m = 0.492 ft

Dt = 100; %Deta time in seconds

diameter\_pipes\_max=24;

diameter\_pipes=[diameter\_pipes\_max diameter\_pipes\_max diameter\_pipes\_max diameter\_pipes\_max]; %Diameter in inches

%NW = Number of Combined Wetlands connected to a single drain

%NSW = Number of SubWetlands for a Combined Wetland

%Find them in a loop.

Done\_search = 0

A{i} = [i; i + 1]

A{i} = 1:i;

for k = 1:NW;

Wetland\_original{k} = zeros(k,2)

Wetland\_search{k} = Wetland\_original{k}

end

%Read data

%If search is completed set Done\_search= 1

for i = 1:NW;

While Done\_search == 0

for i = 1:NW;

NSW = number\_of\_subwetlands(i)

for k = 1:NSW;

compare and start to remove elements. The winners have

ID\_wetland(i,1) = 1

ID\_wetland(i,2) = 5

ID\_donwnstream = 5, -9999,-9999

end

end

end

for k=1:NW

For i = 1:NW

ID\_wetland(i,1) = 1

ID\_wetland(i,2) = 5

ID\_donwnstream = 5, -9999,-9999

A(,2) =

For j= 1,Nw

End

End

End

for j = 1:NW;

d(j) = diameter\_pipes(j); %Diameter in inches

end

k\_compact = 1.25; %constant for pipe cover on top of pipe crown (k\_compact\*d(j));

%Area of wetlands in Acres

Awet(1) = 240.31;

Awet(2) = 161.61;

Awet(3) = 108.73;

Awet(4) = 136.90;

for j = 1:NW;

L(j) = 400; %Pipe length in feet

end

for j = 1:NW;

LS(j) = 0.50; %drop height between the inlet and outlet culvert inverts in feet

end

dr(1) = 0.0;

for j = 2:NW;

dr(j) = 0.0; %in feet (NO DROP)

end

for j = 1:NW;

ke(j) = 0.8; %Culvert entrance loss coefficient (see Federal Highway Administration 2012)

n(j) = 0.009; %n is the roughness coefficient of the culvert barrel (see Federal Highway Administration 2012)

end

%Initial water depths at wetlands above inlet culvert invert

for j = 1:NW;

h\_wet\_init\_depth(j) = 2.0; %measured from 5-way invert

end

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

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

iterations = drainage\_time\_input\*3600/Dt;

for j = 1:NW;

d(j) = d(j)/12; %diameter in feet

end

for j = 1:NW;

cover\_soil(j) = k\_compact\*d(j); % cover measured from culvert inlet in feet

R(j) = d(j)/4;

A(j) = pi()/4\*d(j)^2.0;

Awet(j) = 43560\*Awet(j); %To convert acres to square foot

h\_wet\_init\_depth(j) = h\_wet\_init\_depth(j) + cover\_soil(j)+depth\_Sediments+d(j);

h(j) = h\_wet\_init\_depth(j); %Initial water depths at wetlands above inlet culvert invert

end

for j = 1:NW;

hc(j) = 0.0; %Initial critical depth. This is just to initialize.

end

%Determine intitial flow discharges

k = 1;

while k <iterations;

%%Scenario 2

downst(1) = TW;

downst(2) = h(1);

downst(3) = h(1);

downst(4) = h(2);

%downst(5) = h(3);

for j = 1:NW;

if dr(j) +(hc(j)+d(j))/2.0 > downst(j)

%We are solving for theta

f=@(x) h(j)+LS(j)+ dr(j) -[dr(j) + ([d(j)/2\*(1-cos(x/2))]+d(j))/2.0] - [1/8\*(x-sin(x))\*d(j)^2]^3 /[2.0\*sin(x/2)\*d(j)\*(A(j))^2]\*[1+ke(j)+29\*n(j)^2\*L(j)/R(j)^1.33];

xguess = 0.6\*2\*pi();

else

%We are solving for theta

f=@(x) h(j)+LS(j)+ dr(j) - downst(j) - [1/8\*(x-sin(x))\*d(j)^2]^3 /[2.0\*sin(x/2)\*d(j)\*(A(j))^2]\*[1+ke(j)+29\*n(j)^2\*L(j)/R(j)^1.33] ;

xguess = 0.6\*2\*pi(); %

end

%dc = 0.325(Q/D)2/3 + 0.083D

x=fzero(@(x) f(x),xguess);

Acubictemp = [1/8\*(x-sin(x))\*d(j)^2]^3;

Ttemp = sin(x/2)\*d(j);

hc(j) = d(j)/2\*(1-cos(x/2));

%add n\_pipe for correcting Q;

n\_pipe(j)=number\_pipes(j);

Q(j) = g\*Acubictemp/Ttemp\*n\_pipe(j);

end

temp\_sumQs(1) = Q(2)+Q(3)-Q(1);

temp\_sumQs(2) = Q(4)-Q(2);

temp\_sumQs(3) = -Q(3);

temp\_sumQs(4) = -Q(4);

%temp\_sumQs(5) = -Q(5);

for j = 1:NW;

h(j) = h(j)+Dt/Awet(j)\*temp\_sumQs(j);

end

k = k+1;

% if h(j) > h\_wet\_init\_depth(j); %Maximum water depth in wetland

% h(j) = h\_wet\_init\_depth(j);

% end

if (h(j) < (cover\_soil(j)+d(j)+depth\_Sediments)); %Minimum water depth in wetland

h(j) = (cover\_soil(j)+d(j)+depth\_Sediments);

end

% if ~(mod(k,100))

% %fprintf(fid,'%4.2f\n',x);

% p = h

% end

end

init\_storage = 0.0;

final\_storage = 0.0;

for j = 1:NW;

init\_storage = init\_storage + Awet(j)\* (h\_wet\_init\_depth(j)-cover\_soil(j)-depth\_Sediments-d(j));

final\_storage = (final\_storage + Awet(j)\* (h(j)-cover\_soil(j)-depth\_Sediments-d(j))) ;

% if final\_storage < 0

% final\_storage = 0;

% end

end

% fprintf('The percentage of storage released during the specified drainage time is:');

Ratio\_storage\_released = (init\_storage-final\_storage)/init\_storage;

if Ratio\_storage\_released>1.0

Ratio\_storage\_released=-1;

end

%Maximize

Ratio\_storage\_released =1-Ratio\_storage\_released;

%

% fprintf('The final water depths in the wetlands are (measured from wetland bottom):');

p = h - cover\_soil-d-depth\_Sediments;

%calculate the pipe cost

% fprintf('The cost in the wetlands are :');

for j = 1:NW

CostPipe(j)=(L(j)/10\*(1.8857148201\*diameter\_pipes(j).^2 - 2.5216879005\*diameter\_pipes(j) + 31.4830646471))\*n\_pipe(j);

CostValve(j)=(33.9150692189\*diameter\_pipes(j).^2 - 104.3377332666\*diameter\_pipes(j) + 464.2608919823)\*n\_pipe(j);

end %% in dollar

obj(1)=Ratio\_storage\_released;

obj(2)=(sum(CostPipe)+sum(CostValve))/2.0e7; %% ratio of cost

for j = 1:NW

if h(j)> h\_wet\_init\_depth(j)

obj(1)=100;

obj(2)=100;

end %%keep this constraints for avoiding overflow

end