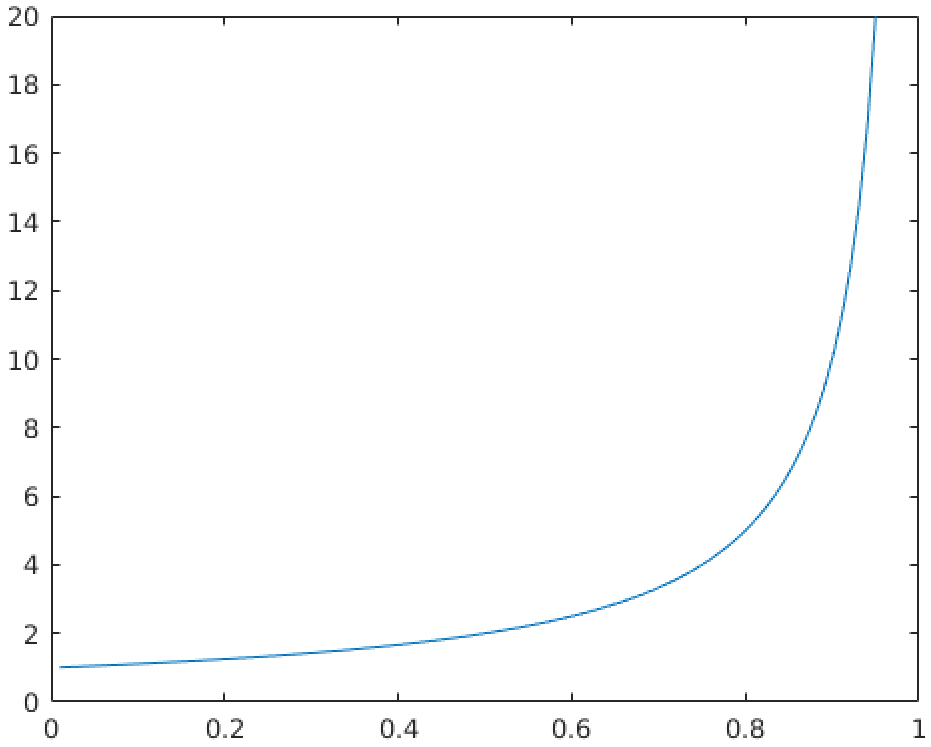
Table of Contents

HW 3 Prob 2a M/M/1

```
clc;clear;
mu = 1;
numLambda = 100;
numTrials = 100;
T = 1e2;
lambda = linspace(0.01,0.95, numLambda);
meanDelay = zeros;
meanTrialDelay = zeros;
```

Theoretical Calculations M/M/1

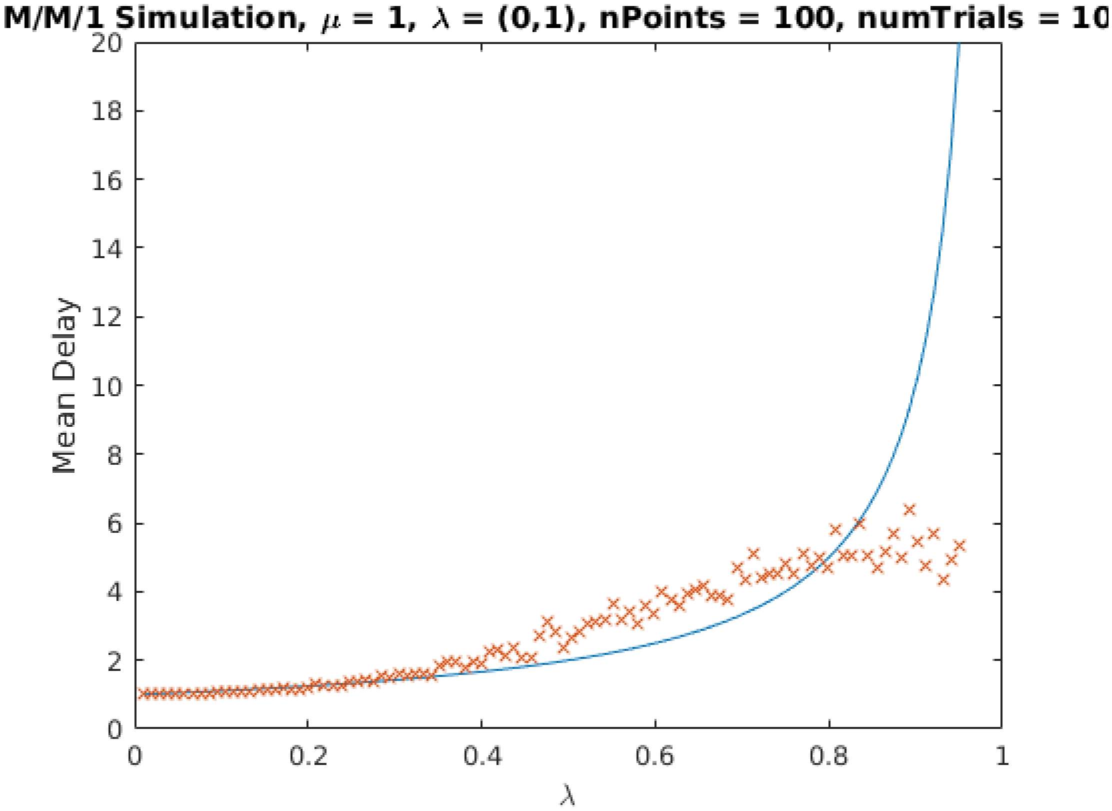
```
expDelay = zeros();
for i = 1:numLambda
    expDelay(i) = inv(mu - lambda(i));
end
plot(lambda,expDelay)
```



Simulation M/M/1

```
for i = 1:numLambda
    meanDelay = zeros;
    for j = 1:numTrials
        A = 1e6;
        S = expRand(lambda(i));
        delayRS = 0;
        Q = zeros; %Array for holding Queue Delays
        qLen = 1;
        for t = 1:T
            %simulate interarrival times as A ~ EXP(lambda) and S ~
 EXP(mu)
            if(A < S) %arrival case
                [Q, qLen, A, S] =
 markovianArrival(Q,qLen,t,lambda(i),mu,S);
            elseif(S < A) %service case
                [Q, qLen, S, delayRS] =
 markovianDeparture(Q,qLen,t,mu,delayRS);
            end
        end
    meanDelay(j) = delayRS/T;
    end
    meanTrialDelay(i) = sum(meanDelay)/numTrials;
end
```

```
% Plot M/M/1
plot(lambda,expDelay)
hold on;
plot(lambda,meanTrialDelay,'x')
title("M/M/1 Simulation, \mu = 1, \lambda = (0,1), nPoints = 100,
 numTrials = 100")
xlabel("\lambda");
ylabel("Mean Delay");
hold off;
```



HW 3 Prob 2b M/D/1

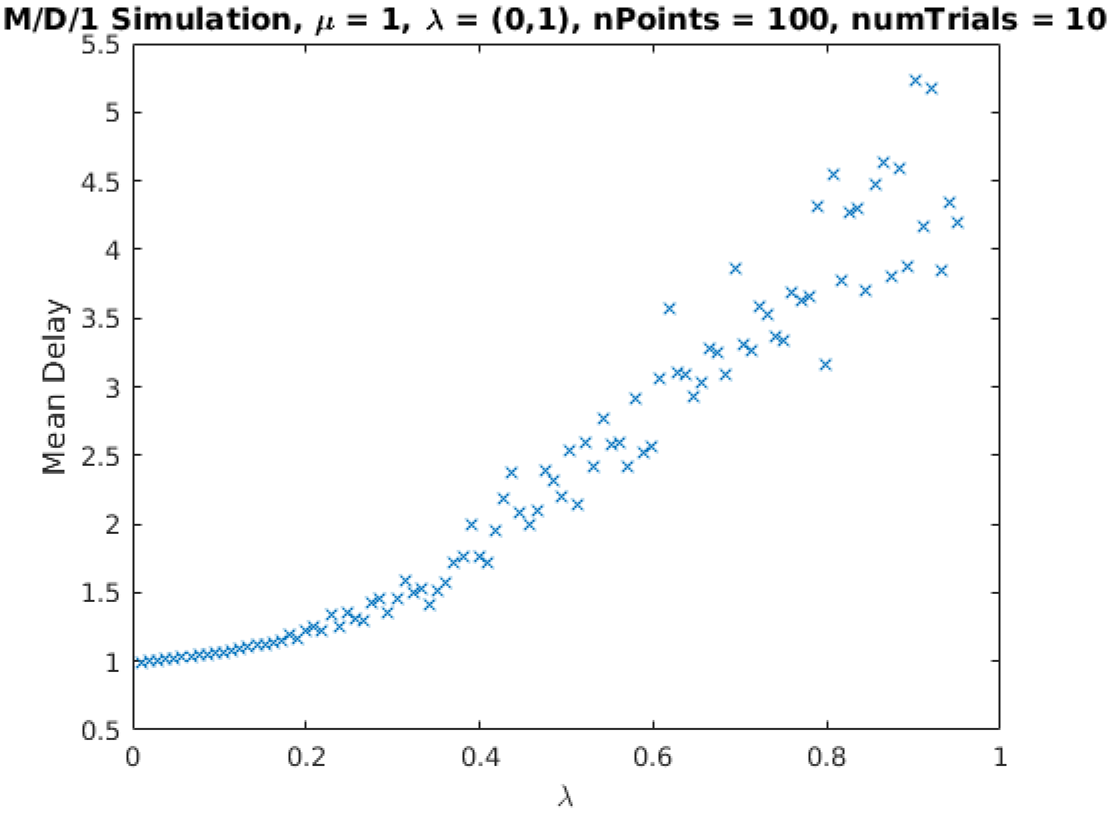
```
clc;clear;
mu = 1;
numLambda = 100;
numTrials = 100;
lambda = linspace(0.01,0.95, numLambda);
meanDelay = zeros();
meanTrialDelay = zeros();
```

Simulation M/D/1

```
T = 1e2;
for i = 1:numLambda
```

```
    for j = 1:numTrials
        A = expRand(lambda(i));
        S = 1e9;
        delayRS = 0;
        Q = zeros; %Array for holding Queue Delays
        qLen = 1;
        for t = 1:T
            %simulate interarrival times as A ~ EXP(lambda) and S ~
 EXP(mu)
            if(A < S) %arrival case
                [Q, qLen, A, S] =
 markovianArrival(Q,qLen,t,lambda(i),mu,S);
            elseif(S < A) %service case
                [Q, qLen, S, delayRS] =
 deterministicDeparture(Q,qLen,t,mu,delayRS);
            end
        end
    meanDelay(i,j) = delayRS/T;
    end
    meanTrialDelay(i) = mean(meanDelay(i,:));
end

%Plot M/D/1
plot(lambda,meanTrialDelay,'x')
title("M/D/1 Simulation, \mu = 1, \lambda = (0,1), nPoints = 100,
 numTrials = 100")
xlabel("\lambda");
ylabel("Mean Delay");
hold off;
```



HW 3 Prob 2c M/G/1

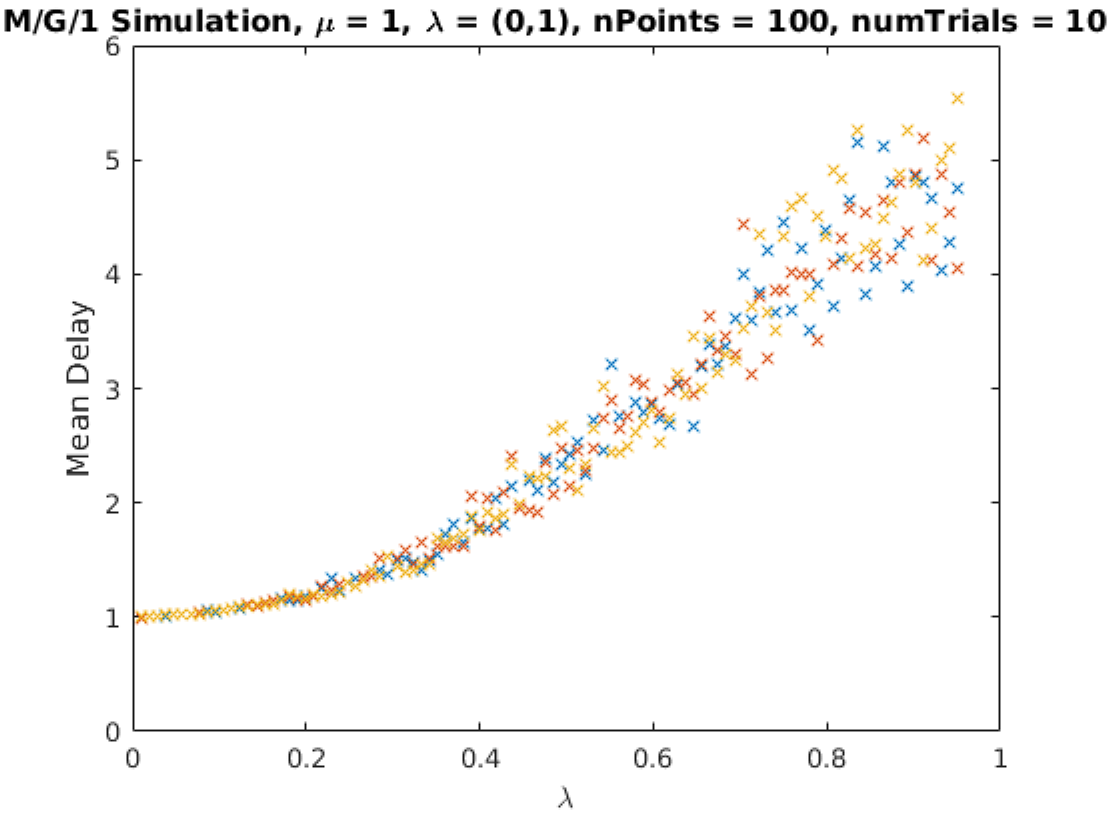
```
clc;clear;
mu = 1;
numLambda = 100;
numTrials = 100;
lambda = linspace(0.01,0.95, numLambda);
meanDelay = zeros();
meanTrialDelay = zeros();
k = 1;

% Simulation M/G/1
T = 1e2;
for p = [0.1, 0.05, 0.01]
    for i = 1:numLambda
        for j = 1:numTrials
            A = expRand(lambda(i));
            S = 1e9;
            delayRS = 0;
            Q = zeros; %Array for holding Queue Delays
            qLen = 1;
            for t = 1:T
                %simulate interarrival times as A ~ EXP(lambda) and S
 ~ EXP(mu)
                if(A < S) %arrival case
```

```
                    [Q, qLen, A, S] =
 markovianArrival(Q,qLen,t,lambda(i),mu,S);
                elseif(S < A) %service case
                    [Q, qLen, S, delayRS] =
 generalDeparture(Q,qLen,t,p,delayRS);
                end
            end
            meanDelay(i,j) = delayRS/T;

        end
        meanTrialDelay(i,k) = sum(meanDelay(i,:))./numTrials;
    end
    k = k+1;
end

%Plot M/D/1
plot(lambda,meanTrialDelay(:,1),'x')
hold on
plot(lambda,meanTrialDelay(:,2),'x')
plot(lambda,meanTrialDelay(:,3),'x')
title("M/G/1 Simulation, \mu = 1, \lambda = (0,1), nPoints = 100,
 numTrials = 100")
xlabel("\lambda");
ylabel("Mean Delay");
hold off;
```



Published with MATLAB® R2017a