**Problem C**

s = rng;

randn(100);

lambda = 2; mu = 1; Xzero = 1;

T = 1; N = 2^8; dt = 1/N;

dW = sqrt(dt)\*randn(1,N);

W = cumsum(dW);

% problem parameters

% Brownian increments

% discretized Brownian path

Xtrue = Xzero\*exp((lambda-0.5\*mu^2)\*([dt:dt:T])+mu\*W);

plot([0:dt:T],[Xzero,Xtrue],’m-’), hold on

R = 4; Dt = R\*dt; L = N/R;

Xem = zeros(1,L);

Xtemp = Xzero;

for j = 1:L

Winc = sum(dW(R\*(j-1)+1:R\*j));

Xtemp = Xtemp + Dt\*lambda\*Xtemp + mu\*Xtemp\*Winc;

Xem(j) = Xtemp;

end

plot([0:Dt:T],[Xzero,Xem],’r--\*’), hold off

xlabel(’t’,’FontSize’,12)

ylabel(’X’,’FontSize’,16,’Rotation’,0,’HorizontalAlignment’,’right’)

emerr = abs(Xem(end)-Xtrue(end))

EM Strong Convegence

>> s = rng;

randn(100);

lambda=2;mu=1;Xzero=1; T= 1; N = 2^9; dt = T/N;

M= 1000;

Xerr = zeros(M,5);

for s=1:M,

dW = sqrt(dt)\*randn(1,N);

W = cumsum(dW);

Xtrue = Xzero\*exp((lambda-0.5\*mu^2)+mu\*W(end));

for p = 1:5

R = 2^(p-1); Dt = R\*dt; L = N/R;

Xtemp = Xzero;

for j = 1:L

Winc = sum(dW(R\*(j-1)+1:R\*j));

Xtemp = Xtemp + Dt\*lambda\*Xtemp + mu\*Xtemp\*Winc;

end

Xerr(s,p) = abs(Xtemp - Xtrue);

end

end

Dtvals = dt\*(2.^([0:4]));

subplot(221)

loglog(Dtvals,mean(Xerr)), hold on

loglog(Dtvals,(Dtvals.^(.5))), hold off

axis([1e-3 1e-1 1e-4 1])

A = [ones(5,1), log(Dtvals)]; rhs = log(mean(Xerr));

sol = A\rhs; q = sol(2)

resid = norm(A\*sol - rhs)

%% Weak Convergence

randn(’state’,100);

lambda = 2; mu = 0.1;

Xzero = 1;

T = 1;

M = 50000;

Xem = zeros(5,1);

for p = 1:5

Dt = 2^(p-10); L = T/Dt;

Xtemp = Xzero\*ones(M,1);

for j = 1:L

Winc = sqrt(Dt)\*randn(M,1);

% Winc = sqrt(Dt)\*sign(randn(M,1)); %% use for weak E-M %%

Xtemp = Xtemp + Dt\*lambda\*Xtemp + mu\*Xtemp.\*Winc;

end

Xem(p) = mean(Xtemp);

end

Xerr = abs(Xem - exp(lambda));

Dtvals = 2.^([1:5]-10);

subplot(222)

loglog(Dtvals,Xerr,’b\*-’), hold on

loglog(Dtvals,Dtvals,’r--’), hold off

axis([1e-3 1e-1 1e-4 1])

xlabel(’\Delta t’), ylabel(’| E(X(T)) - Sample average of X\_L |’)

title(’emweak.m’,’FontSize’,10)

A = [ones(p,1), log(Dtvals)’]; rhs = log(Xerr);

sol = A\rhs; q = sol(2)

resid = norm(A\*sol - rhs)