Project Report

Name: Akshay Kumar Bhardwaj **Roll no. 214104607**

1). Stochastic BOD

Consider the stochastic BOD model

$$\frac{dB_t}{dt} = -K_1 B_t + s_1 - B_t \sigma \xi_t, \quad B_{t_0} = B_0$$

or in terms of the Wiener process:

$$dB_t = (-K_1 B_t + s_1)\, dt - B_t \sigma dW_t, \quad B_{t_0} = B_0$$

Since the white noise process in this stochastic model is a mathematical approximation of a noise process with a relatively short correlation scale, this SDE has to be interpreted in the Stratonovich sense. Since the Euler scheme can only be used for Itô equations, the model above is rewritten as an Itô SDE:

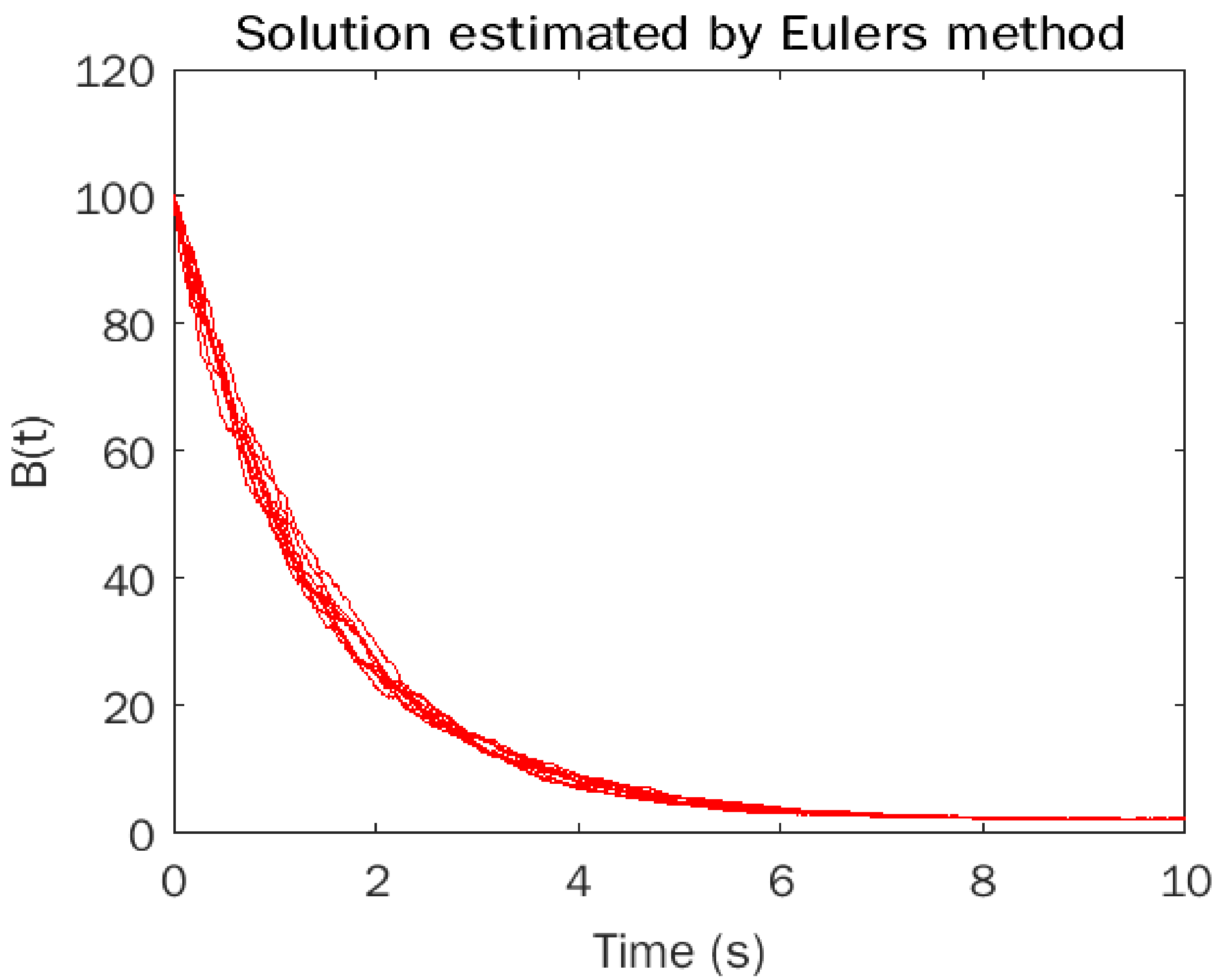
$$dB_t = \left(-K_1 B_t + s_1 + \frac{1}{2} B_t \sigma^2\right) dt - B_t \sigma dW_t, \quad B_{t_0} = B_0$$

Write a MATLAB code to obtain the realizations of B_t using Eular-Maruyama scheme.

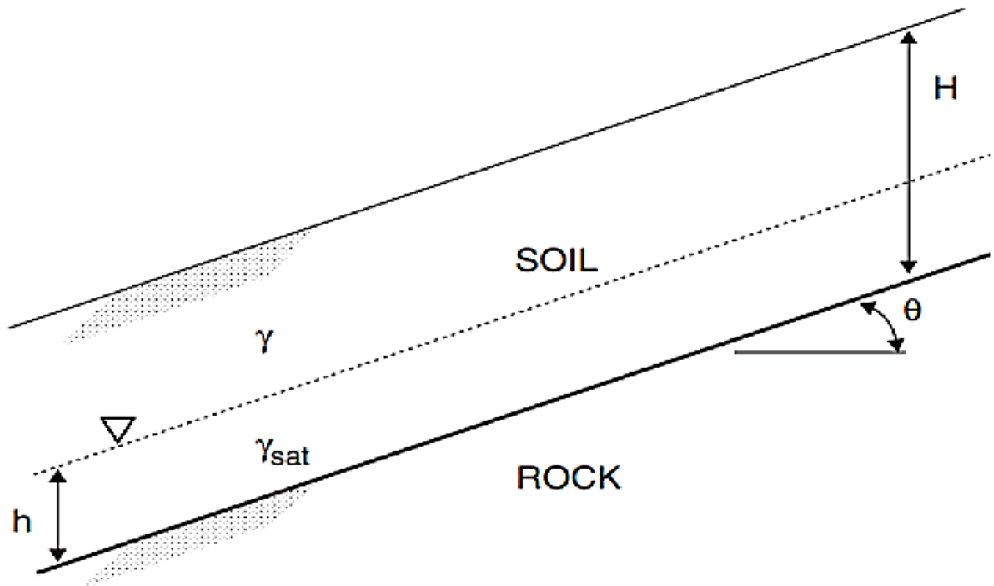
CODE

```
%% the realizations of Bt using Eular-Maruyama scheme
%{
---------------------------------------------------------------------------
*Created by:                 Date:
 Akshay Kumar Bhardwaj      November 2021
*Mail:
 akshay_kumar@iitg.ac.in
*Istitute:
 Indian Institue of Technology, Guwahati
---------------------------------------------------------------------------
%}

numSims = 10;    %10 runs
tBounds = [0 10];   % Time bound
N = 1000;
dt = (tBounds(2) - tBounds(1))/N;
b_init = 100;   % Initial Bt value

pd = makedist('Normal','mu',0,'sigma',sqrt(dt));

c = [0.7, 1.5, 0.06];   % K1, s1, sigma

ts = linspace(tBounds(1), tBounds(2), N); % From t0-->t1 with N points
bs = zeros(1,N); % 1xN Matrix of zeros

bs(1) = b_init;
%% Computing the Process
for j = 1:numSims
    for i = 2:numel(ts)
        t = tBounds(1) + (i-1).*dt;
        x = bs(i-1);
        a = -c(1).*x + c(2) + 0.5*c(3)*c(3).*x;
        b = -c(3).*x;
        dW = random(pd);

        bs(i) = x + a.*dt + b.*dW;
    end

    plot(ts, bs, 'r')
    hold on;
    xlabel('Time (s)');
    ylabel('B(t)');
    title('Solution estimated by Eulers method')
end
```

RESULT/OUTPUT



2). Stochastic Stability of Infinite slopes



$$P = \frac{\left[\gamma\,(H-h) + h\,(\gamma_{sat} - \gamma_w)\right]\cos\theta\tan\phi}{\left[\gamma\,(H-h) + h\gamma_{sat}\right]\sin\theta} - 1$$

H = depth of soil above bedrock
h = height of groundwater table above bedrock,
γ and γ_{sat} = moist unit weight & saturated unit weight of the surficial soil, respectively,
γ_w = unit weight of water (9.81 kN/m^3),
φ = effective stress friction angle,
and θ = slope inclination.

Variable	Description	Distribution	Statistics
H	Depth of soil above bedrock	Uniform	[2,8] m
$h = H \times U$	Height of water table	U is uniform	[0, 1]
ϕ	Effective stress friction angle	Lognormal	mean = 35° cov = 8%
θ	Slope inclination	Lognormal	mean = 20° cov = 5%
γ	Moist unit weight of soil	*	*
γ_{sat}	Saturated unit weight of soil	**	**
γ_w	Unit weight of water	Deterministic	9.81 kN/m^3

* $\gamma = \gamma_w (G_s + 0.2e)/(1 + e)$ (assume degree of saturation = 20% for "moist").
** $\gamma_{sat} = \gamma_w (G_s + e)/(1 + e)$ (degree of saturation = 100%).
Assume specific gravity of solids = G_s = uniformly distributed [2.5, 2.7] and void ratio = e = uniformly distributed [0.3, 0.6].

Write a MATLAB code to perform Monte Carlo simulation in order to estimate the probability of failure

CODE

```
%% Monte Carlo method for estimating the probability of failure of infinite slope
%{
---------------------------------------------------------------------------
*Created by:                  Date:
 Akshay Kumar Bhardwaj        November 2021
*Mail:
 akshay_kumar@iitg.ac.in
*Istitute:
 Indian Institute of Technology, Guwahati
---------------------------------------------------------------------------
%}
%% Initial data and calculations
Ww = 9.81;                          % weight density of water [KN/m^3]
W = @(Gs,e) Ww*(Gs+0.2*(e))/(1+e);  % bulk density of soil [KN/m^3]
Wsat = @(Gs,e) Ww*(Gs+e)/(1+e);     % saturated density of soil [KN/m^3]
h = @(H,U) (H*U);                   % height of water table [m]

P = @(W,Wsat,h,H,thita,phi) ((W*(H-h)+(h*(Wsat-Ww)))*(cos(thita*pi/180))*(tan
(phi*pi/180)))/(((W*(H-h))+(h*Wsat))*sin(thita*pi/180));

% effective stress friction angle (phi): phi ~ logn(mean = 35', var = 0.08');
mu_phi  = 35;
var_phi = .08;
```

```
muphi    = log((mu_phi^2)/sqrt(var_phi+mu_phi^2));
sigmaphi = sqrt(log(var_phi/(mu_phi^2)+1));

% slope inclination (theta): theta ~ logn(mean = 20', var = 0.05');
mu_thita  = 20;
var_thita = .05;
muthita    = log((mu_thita^2)/sqrt(var_thita+mu_thita^2));
sigmathita = sqrt(log(var_thita/(mu_thita^2)+1));

a    = 1;                % threshold level
NSIM = 1000;             % number of monte carlo simulations
b    = zeros(1,NSIM);    % allocating memory for bulk density
c    = zeros(1,NSIM);    % allocating memory for saturated density
d    = zeros(1,NSIM);    % allocating memory for P (factor of safety)
x    = zeros(1,NSIM);    % allocating memory for height of water table


%% MCS using normal computing
fprintf('MONTE CARLO SIMULATION : \n');
tic;
for i = 1:NSIM
    H = unifrnd(2,8);
    U = unifrnd(0,1);
  phi    = lognrnd(muphi,sigmaphi);
  thita    = lognrnd(muthita,sigmathita);
  Gs = unifrnd(2.5,2.7);
  e = unifrnd(0.3,0.6);
  b(i) = W(Gs,e);
  c(i) = Wsat(Gs,e);
  x(i) = h(H,U);
  d(i) = P(W(Gs,e),Wsat(Gs,e),h(H,U),H,thita,phi);
end
toc;   t1 = toc;
[ff1,xx1] = ecdf(d);              % estimate empirical CDF
pf        = mean(d<=a);           % failure probability
var_MCS   = pf*(1-pf)/NSIM;       % variance
std_MCS   = sqrt(var_MCS);
fprintf('Failure probability: %7.8f +- %g \n\n', pf, std_MCS);
```

RESULT/OUTPUT

```
slopemcs
MONTE CARLO SIMULATION :
Elapsed time is 0.455888 seconds.
Failure probability: 0.01400000 +- 0.00371537
```

3).

Autocorrelation function of a sine wave

Compare the autocorrelation functions of a sinusoidal signal $x(t) = A\sin(\omega t + \theta)$, resulting from the ensemble average and the time average. The theoretical autocorrelation function is

$$R_{xx}(\tau) = \frac{A^2}{2}\cos(\omega\tau)$$

For the ensemble average θ is a random variable and t is fixed, and for the time average θ is fixed and t is a time variable.

Write MATLAB code to estimate SAMPLE & ENSEMBLE auto-correlations.

CODE

```
%% the realizations of Bt using Eular-Maruyama scheme
%{
---------------------------------------------------------------------------
*Created by:                   Date:
 Akshay Kumar Bhardwaj         November 2021
*Mail:
 akshay_kumar@iitg.ac.in
*Istitute:
 Indian Institue of Technology, Guwahati
---------------------------------------------------------------------------
%}
A=10;
t=0;
fs=100;
omega=2*pi*2;

%%
theta=rand(1,10000)*2*pi;
x1=A*sin(omega*t+theta);
Rxx1=[];
tp=10;

for tau=-tp:1/fs:tp
tmp=A*sin(omega*(t+tau)+theta);
tmp=mean(x1.*tmp);
Rxx1=[Rxx1 tmp];
end
```

```
tau=-tp:1/fs:tp;
Rxx=A^2/2*cos(omega*tau);

%%
t=0:1/fs:20-1/fs;
x2=A*sin(omega*t);
[Rxx2, tau2]=xcorr(x2,x2,tp*fs,'unbiased');
tau2=tau2/fs;

%%

subplot(2,1,1);
plot(tau,Rxx1,tau,Rxx,'linewidth',1)
xlabel('tau')
ylabel('Autocorrelation')
title('Ensemble Average')
grid on

subplot(2,1,2);
plot(tau2,Rxx2,tau,Rxx,'g','linewidth',1)
xlabel('tau')
ylabel('Autocorrelation')
title('Time Average')
grid on
```

RESULT/OUTPUT