Praveen Kumar Yadav

(MTech VLSI Sem 1, EE22M308)

Lab no. : 7

Used version: Matlab 2022

Q.1: Numerical solution of Time dependent Diffusion equation

(a) Describe the formalism to solve time dependent diffusion equation using backward Euler scheme.

Ans:

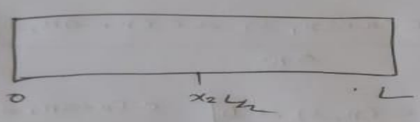
Ans. No - 1 (a)

Time dependent diffusion eqn of particles in a given semiconductor

$$\frac{dc}{dt} = D\frac{\partial^2 c}{\partial x^2} + G - R$$

where D = Diffusion co-efficient
C = concn of particles
G = Generation rate
R = Recombination rate

Take a SC with length L & at x = L/2 diffusion particle are injected.



$\frac{dc}{dt} =$

$$\frac{C(x, t+\Delta t) - C(x, t)}{\Delta t} = D\frac{\left[C(x+\Delta x, t+\Delta t) + C(x-\Delta x, t+\Delta t) - 2C(x, t+\Delta t)\right]}{\Delta x^2}$$

$$c_i(t+\Delta t) - \alpha c_{i+1}(t+\Delta t) - \alpha c_i(t-\Delta t) + 2\alpha c_i(t+\Delta t) = c_i(t)$$

where $\alpha = D\frac{\Delta t}{\Delta x^2}$

$C_1(t+\Delta t) = 0$ boundary condn

$i=2$, $(1+2\alpha) C_2(t+\Delta t) - 2C_1(t+\Delta t) - \alpha C_3(t+\Delta t) = C_2(t)$

$i=3$, $(1+2\alpha) C_3(t+\Delta t) - \alpha C_2(t+\Delta t) - 2C_4(t+\Delta t) = C_3(t)$

⋮

$C_N(t+\Delta t) = 0$ boundary condn

$$
\begin{bmatrix} 1+2\alpha & -\alpha & 0 & 0 \\ -\alpha & 1+2\alpha & -\alpha & 0 \\ 0 & -\alpha & 1+2\alpha & -\alpha \\ \vdots & & & \end{bmatrix}
\begin{bmatrix} C_2(t+\Delta t) \\ C_3(t+\Delta t) \\ C_4(t+\Delta t) \\ \vdots \end{bmatrix}
=
\begin{bmatrix} C_2(t) \\ C_3(t) \\ C_4(t) \\ \vdots \end{bmatrix}
$$

$$
C(t+\Delta t) = H^{-1} C(t)
$$

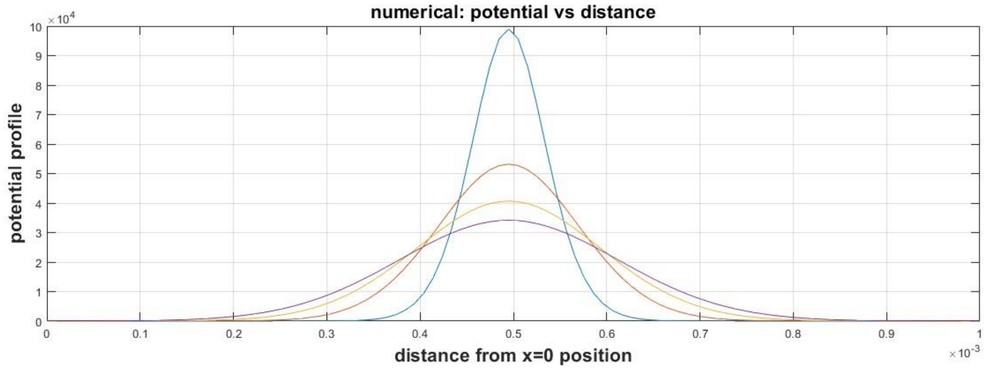
(b) Consider a region of length 10 µm. Assume perfectly absorbing boundary conditions at x=0 and at x=10 µm. At time t=0, assume that particles are injected at x=5 µm is such that the density is 10^6cm-3 (i.e., the injection is a delta function in both space and time). Using the formalism described in (a) explore the evolution of particle density over the specified domain (use D=10-4 cm2/s). Compare with analytical results. Explore the significance of the parameter.

Ans :

Numerical method:

```
clear all;
clc;
D=10^-4; % take D in cm^2/sec
l=10^-3; % take length in cm
time=100*10^-6; %take time 100 micro sec
t1=linspace(0,time,100); % take 100 spacing between time
l1=linspace(0,l,100); % take 100 spacing between length
t=10^-6; % small portion of time
h=10^-5; % small portion of distance
a=D*t/(h^2);
Co=[];
Co = zeros(100,1);
Co(50,1)=10^6;
H=[];
D = [];
for j=1:100
for i =2:100
    H(i,i)=1+2*a;
end
for i=2:99
    H(i,i+1)=-a;
end
for i=1:98
    H(i+1,i)=-a;
end
H(1,1)=1;
H(100,100)=1;
```

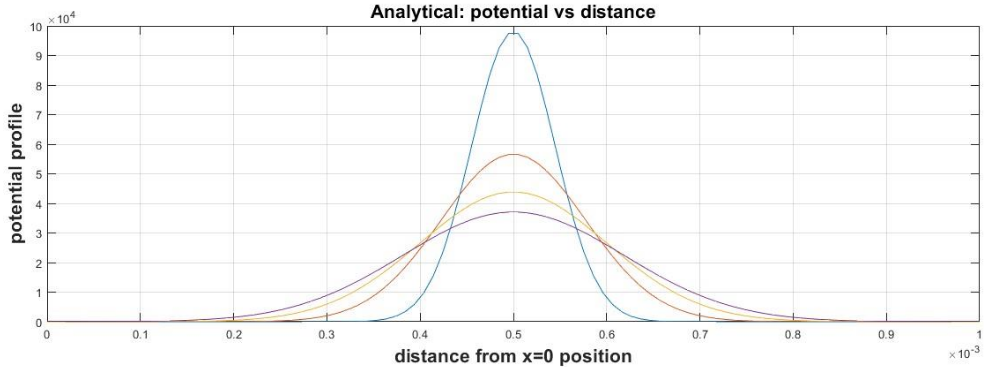
```
Cn=inv(H)*Co;
D=[D,Co];
Co=Cn;

end
plot(l1,D(:,10),l1,D(:,30),l1,D(:,50),l1,D(:,70));
grid on;
xlabel('distance from x=0 position');
ylabel('potential profile');
title('numerical: potential vs distance');
```



Analytical method:

```
clear all;
clc;
D=10^-4; % take D in cm^2/sec
l=10^-3; % take length in cm
time1=10*10^-6; %take time 100 micro sec
time2=30*10^-6;
time3=50*10^-6;
time4=70*10^-6;
% m=linspace(0,time,100); % take 100 spacing between time
k=linspace(0,l,100) ; % take 100 spacing between length
t=10^-6; % small portion of time
h=10^-5; % small portion of distance
Q=1.1*10^1;
p=[];
a1=2*((pi*D*time1)^0.5);
a2=2*((pi*D*time2)^0.5);
a3=2*((pi*D*time3)^0.5);
a4=2*((pi*D*time4)^0.5);
C1=((Q/a1)*exp(-((k-(0.5*l)).^2)/(4*D*time1)));

C2=((Q/a2)*exp(-((k-(0.5*l)).^2)/(4*D*time2)));

C3=((Q/a3)*exp(-((k-(0.5*l)).^2)/(4*D*time3)));

C4=((Q/a4)*exp(-((k-(0.5*l)).^2)/(4*D*time4)));

plot(k,C1,k,C2,k,C3,k,C4);

% for i=1:100
%
% p=[p,C];
%  plot  (p);
% hold on
% end
```

```
grid on
xlabel('distance from x=0 position');
ylabel('potential profile');
title('Analytical: potential vs distance');
```



Ans No – 1 (b)

Significance of the Parameters :

Diffusion co-efficient :- (D) :- Diffusion co-efficient implies that the mass of the substance diffuse through a unit surface in a unit time at a concentration gradient of unity.

Unit :- cm^2/sec.

Flux density (Q) :- rate of movement of particles through a unit area normal to the direction of gradient of particles.

$$Q = 2C(0) \cdot \sqrt{\pi D t}$$

Unit : Cm^{4} Sec.

Q.2: Random Walk simulations:

(a) Discretize the time dependent diffusion equation and arrive at a scheme for solving the time dependent diffusion equation through random walk simulations. For D=10-4 cm2/s and Δx=10 nm, what should be the Δt, the time step in such simulations?

Ans:

2(a)

Random walk:

$$\frac{dc}{dt} = D\frac{\partial^2 c}{\partial x^2}$$

$$\frac{c(x, t+\Delta t) - c(x,t)}{\Delta t} = D\frac{\left(c(x+\Delta x, t) + c(x-\Delta x, t) - 2c(x,t)\right)}{\Delta x^2}$$

$$c(x, t+\Delta t) = \left(1 - 2\frac{D\Delta t}{\Delta x^2}\right) c(x,t) + D\left[\frac{c(x+\Delta x, t) + c(x-\Delta x, t)}{\Delta x^2}\right]$$

Case 1 -

$$1 - 2\frac{D\Delta t}{dx^2} = 0$$

$$\frac{D\Delta t}{\Delta x^2} = 1/2$$

then $c(x, t+\Delta t) = 1/2\left[c(x+\Delta x, t) + c(x-\Delta x, t)\right]$

to be $\Delta t = 1\,\mu sec$. if $\Delta x = 10\,nm$

$$D = \frac{1}{2} \times \frac{100\times10^{-18}}{1\times10^{-6}} = 50\times10^{-12}\ m^2/sec.$$

Case 2 :-

$$1 - 2\frac{D\Delta t}{dx^2} = 1/2$$

then $\frac{D\Delta t}{\Delta x^2} = \frac{1}{4}$ $\Rightarrow$ if $\Delta x = 10nm$, $\Delta t = 1\mu sec$ $D = \frac{1}{4}\times\frac{100\times10^{-18}}{1\times10^{-6}} = 25\times10^{-12}$

then $c(x, t+\Delta t) = \frac{1}{2}c(x,t) + \frac{1}{4}\left(c(x+\Delta x, t) + c(x-\Delta x, t)\right)$

(b) Assume that N=100 particles are released at x=5 µm at t=0. Explore the evolution of particle density profile as a function of time using random walk simulations. Compare with analytical results. Explore the density function for N=1000, and N=10000 particles.

Ans

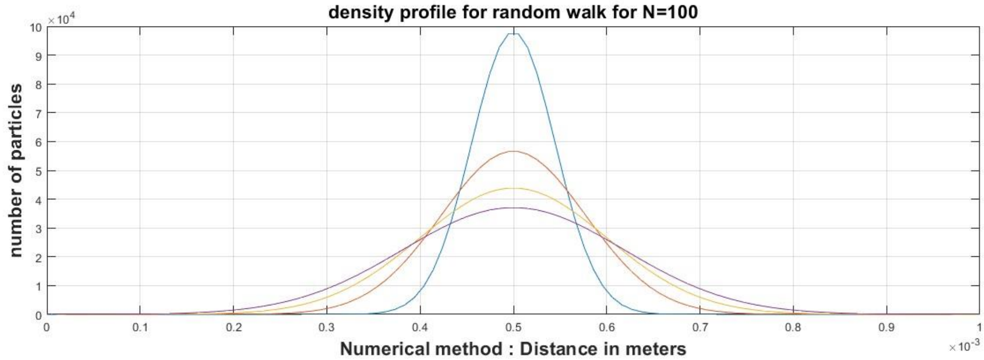
Code:

For N=100

Numerical:

```
clc;
clear all;
```

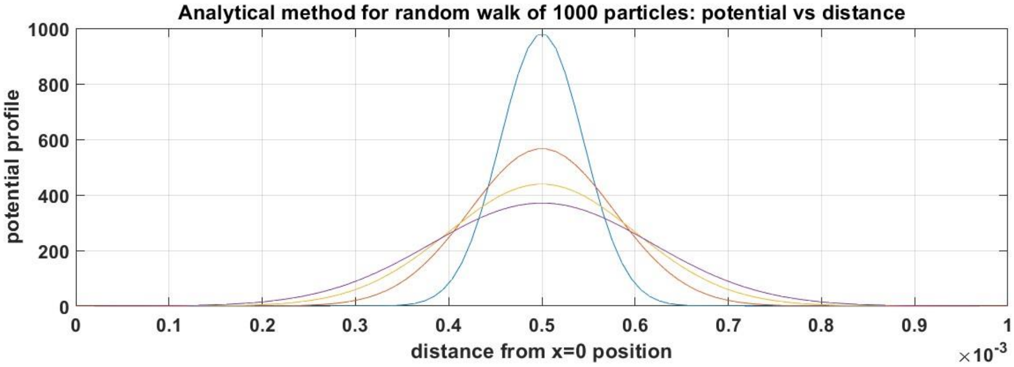
```
l=100*10^-6; % length is 100 micro meters
m=linspace(0,10^-5,1000);
a=1;
c=zeros(1000,1);
c(500,1)=100;
d=[];
hyp=eye(1000);
for i=1:100
% creating jacobian matrix
h=zeros(1000);
    for i=2:999
        h(i,i-1)=0.5;
        h(i,i+1)=0.5;
    end

% creating the for loop for iterations 100times
cf=hyp*(h*c);
d=[d,c];
c=cf;

%plot(m,d(:,10),m,d(:,30),m,d(:,50),m,d(:,70));
grid on
xlabel('Numerical method : Distance in meters')
ylabel('number of particles')
title('density profile for random walk for N=100')
hold on
end
plot(m,d(:,1),m,d(:,30),m,d(:,50),m,d(:,70));
```



ANALYTICAL:

```
clear all;
clc;
D=10^-4; % take D in cm^2/sec
l=10^-3; % take length in cm
time1=10*10^-6; %take time 100 micro sec
time2=30*10^-6;
time3=50*10^-6;
time4=70*10^-6;
% m=linspace(0,time,100); % take 100 spacing between time
k=linspace(0,l,100) ; % take 100 spacing between length
t=10^-6; % small portion of time
h=10^-5; % small portion of distance
Q=0.011;
p=[];
a1=2*((pi*D*time1)^0.5);
a2=2*((pi*D*time2)^0.5);
```

```
a3=2*((pi*D*time3)^0.5);
a4=2*((pi*D*time4)^0.5);
C1=((Q/a1)*exp(-((k-(0.5*l)).^2)/(4*D*time1)));

C2=((Q/a2)*exp(-((k-(0.5*l)).^2)/(4*D*time2)));

C3=((Q/a3)*exp(-((k-(0.5*l)).^2)/(4*D*time3)));

C4=((Q/a4)*exp(-((k-(0.5*l)).^2)/(4*D*time4)));

plot(k,C1,k,C2,k,C3,k,C4);

% for i=1:100
%
% p=[p,C];
%  plot  (p);
% hold on
% end

grid on
xlabel('distance from x=0 position');
ylabel('potential profile');
title('Analytical method for random walk of 100 particles: potential vs distance');
```

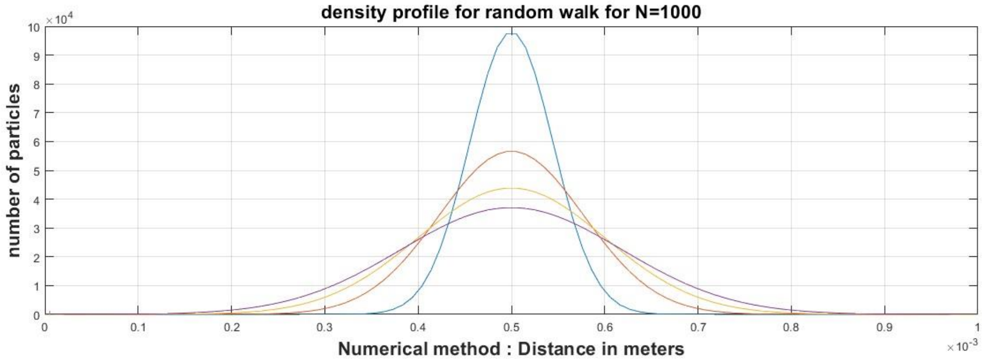


FOR N=1000

NUMERICAL:

```
clc;
clear all;
l=100*10^-6; % length is 100 micro meters
m=linspace(0,10^-5,1000);
a=1;
c=zeros(1000,1);
c(500,1)=1000;
d=[];
hyp=eye(1000);
for i=1:100
% creating jacobian matrix
h=zeros(1000);
    for i=2:999
        h(i,i-1)=0.5;
        h(i,i+1)=0.5;
    end
```

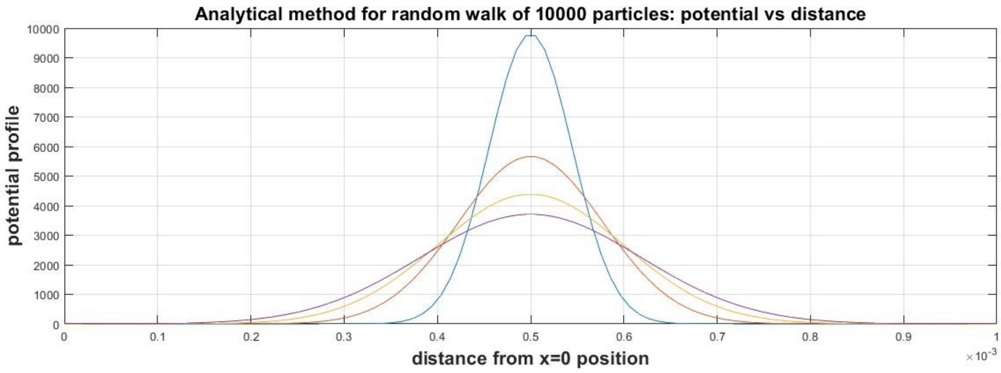
```
% creating the for loop for iterations 100times
cf=hyp*(h*c);
d=[d,c];
c=cf;

%plot(m,d(:,10),m,d(:,30),m,d(:,50),m,d(:,70));
grid on
xlabel('Numerical method : Distance in meters')
ylabel('number of particles')
title('density profile for random walk for N=1000')
hold on
end
plot(m,d(:,1),m,d(:,30),m,d(:,50),m,d(:,70));
```



ANALYTICAL:

```
clear all;
clc;
D=10^-4; % take D in cm^2/sec
l=10^-3; % take length in cm
time1=10*10^-6; %take time 100 micro sec
time2=30*10^-6;
time3=50*10^-6;
time4=70*10^-6;
% m=linspace(0,time,100); % take 100 spacing between time
k=linspace(0,l,100) ; % take 100 spacing between length
t=10^-6; % small portion of time
h=10^-5; % small portion of distance
Q=0.11;
p=[];
a1=2*((pi*D*time1)^0.5);
a2=2*((pi*D*time2)^0.5);
a3=2*((pi*D*time3)^0.5);
a4=2*((pi*D*time4)^0.5);
C1=((Q/a1)*exp(-((k-(0.5*l)).^2)/(4*D*time1)));

C2=((Q/a2)*exp(-((k-(0.5*l)).^2)/(4*D*time2)));

C3=((Q/a3)*exp(-((k-(0.5*l)).^2)/(4*D*time3)));

C4=((Q/a4)*exp(-((k-(0.5*l)).^2)/(4*D*time4)));

plot(k,C1,k,C2,k,C3,k,C4);

% for i=1:100
%
```

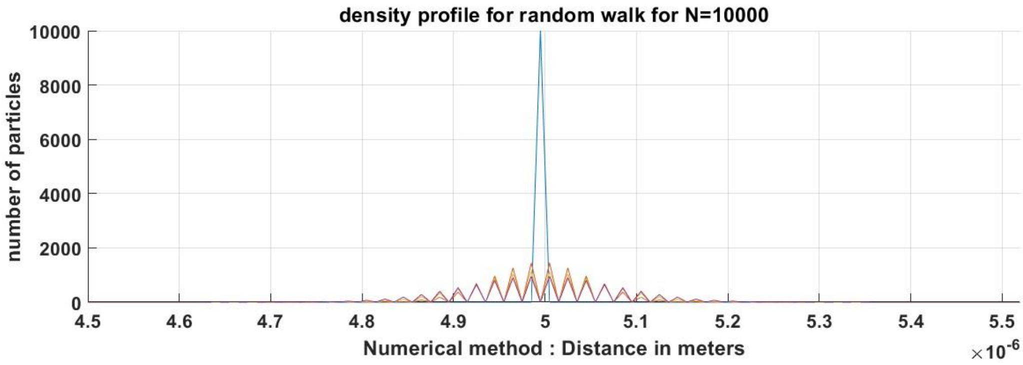
```
% p=[p,C];
%  plot  (p);
% hold on
% end

grid on
xlabel('distance from x=0 position');
ylabel('potential profile');
title('Analytical method for random walk of 1000 particles: potential vs
distance');
```



FOR N=10000

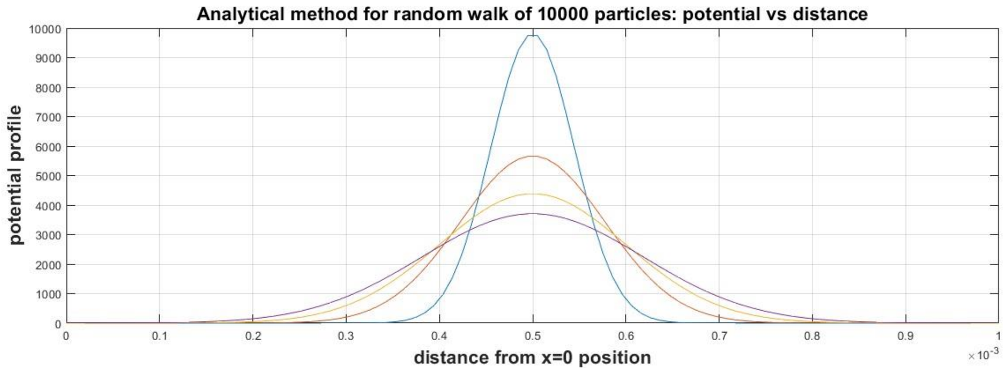
NUMERICAL:

```
clc;
clear all;
l=100*10^-6; % length is 100 micro meters
m=linspace(0,10^-5,1000);
a=1;
c=zeros(1000,1);
c(500,1)=10000;
d=[];
hyp=eye(1000);
for i=1:100
% creating jacobian matrix
h=zeros(1000);
    for i=2:999
        h(i,i-1)=0.5;
        h(i,i+1)=0.5;
    end

% creating the for loop for iterations 100times
cf=hyp*(h*c);
d=[d,c];
c=cf;

%plot(m,d(:,10),m,d(:,30),m,d(:,50),m,d(:,70));
grid on
xlabel('Numerical method : Distance in meters')
ylabel('number of particles')
title('density profile for random walk for N=10000')
hold on
end
plot(m,d(:,1),m,d(:,30),m,d(:,50),m,d(:,70));
```



ANALYTICAL:

```
clear all;
clc;
D=10^-4; % take D in cm^2/sec
l=10^-3; % take length in cm
time1=10*10^-6; %take time 100 micro sec
time2=30*10^-6;
time3=50*10^-6;
time4=70*10^-6;
% m=linspace(0,time,100); % take 100 spacing between time
k=linspace(0,l,100) ; % take 100 spacing between length
t=10^-6; % small portion of time
h=10^-5; % small portion of distance
Q=1.1;
p=[];
a1=2*((pi*D*time1)^0.5);
a2=2*((pi*D*time2)^0.5);
a3=2*((pi*D*time3)^0.5);
a4=2*((pi*D*time4)^0.5);
C1=((Q/a1)*exp(-((k-(0.5*l)).^2)/(4*D*time1)));

C2=((Q/a2)*exp(-((k-(0.5*l)).^2)/(4*D*time2)));

C3=((Q/a3)*exp(-((k-(0.5*l)).^2)/(4*D*time3)));

C4=((Q/a4)*exp(-((k-(0.5*l)).^2)/(4*D*time4)));

plot(k,C1,k,C2,k,C3,k,C4);

% for i=1:100
%
% p=[p,C];
%  plot  (p);
% hold on
% end

grid on
xlabel('distance from x=0 position');
ylabel('potential profile');
title('Analytical method for random walk of 10000 particles: potential vs
distance');
```



Analytical method for random walk of 10000 particles: potential vs distance
potential profile
distance from x=0 position
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
×10-3