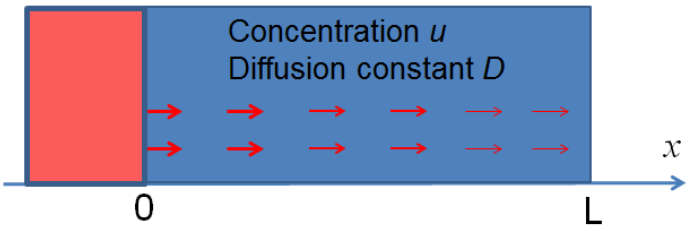

The concentration of molecules follows the diffusion equation $\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2}$ (1)

The initial condition ($t = 0$): $u = 0$ for all $x \geq 0$ (no molecules initially in the medium).
Two boundary conditions:

At $x = 0$: $D\frac{\partial u}{\partial x} = -F_p$ for all $t > 0$ (molecules are produced at a constant rate at left boundary)

At $x = L$: $u = 0$ for all $t > 0$ (molecules are immediately consumed once reaching the right boundary)

In our initial practice, we will use the following settings:

Diffusion coefficient: $D = 10^{-7}$ cm^2/sec ;
Molecules producing rate: $Fp = 4.94 \times 10^{-19}$ mol/cm^2•s ;
Total distance: $L = 0.25$ cm (divide the 0.25 cm distance into 20 steps);
Total time period: $T = 60 \times 3600$ s (divide the 60-hour period into 11 time points).

See the following pages for details on how to use PDEPE to solve this problem.

2) About PDEPE:

PDEPE is a MATLAB program developed for solving the initial-boundary value problems for parabolic-elliptic PDEs in 1D with a general form of

$$c\left(x,t,u,\frac{\partial u}{\partial x}\right)\frac{\partial u}{\partial t}=x^{-m}\frac{\partial}{\partial x}\left(x^{m}f\left(x,t,u,\frac{\partial u}{\partial x}\right)\right)+s\left(x,t,u,\frac{\partial u}{\partial x}\right) \quad (2)$$

sol [t,x,i]= pdepe(m,pdefun,icfun,bcfun,xmesh,tspan)
When we call pdepe, six input arguments (parameters or functions) need to be provided:
m: parameter correspondent to the geometry of the system (slab 0; cylindrical 1; spherical 2)
pdefun: define the format of the pde function (function handle)
icfun: define the initial condition (function handle)
bcfun: define the boundary condition (function handle)
xmesh: define the distance (and step size) for calculation (1D matrix)
tspan: define the time period for calculation (1D matrix)
sol: output argument (solution) is a 3D matrix, including time, distance, and a system of pde functions (usually only have 1).

Before using pdepe, we have to fit the format of our Eq. (1) into the general form of Eq. (2) that can be recognized by the PDEPE.

By comparing Eq. (1) and Eq. (2), we can see that for our equation, m=0 ; $c=1/D$; $f=\partial u/\partial x$; $s=0$.
So, when defining pdefun, we have:

```
function [c, f, s] = pdefun(x,t,u,DuDx)
c = 1/D;
f = DuDx;
s = 0;
```

For initial condition at $t = 0$, $u = 0$ for all $x \geq 0$. Therefore, we have

```
function u0 = icfun(x)
u0=0;
```

According to the two boundary conditions at x=0 and x=L, we have

```
function [pl,ql,pr,qr] = bcfun(xl,ul,xr,ur,t)
pl = Fp;
ql = D;
pr = ur;
qr = 0;
```

We also need to set **xmesh, tspan** by creating two 1D matrix by doing

```
xmesh=linspace(0, L, N);  % L is the total distance along x, N is the steps along x
                          % The suggested setting is linspace(0, 0.25, 20)
tspan=linspace(0, T, M);  % T is the total time period, M is the time points.
                          % The suggested setting is linspace(0, 60*3600, 11)
```

Before starting to program, please look at an example of "**Solve Single PDE**" via the following link. The question in this example (including the equation) is similar to the problem we are solving.
https://www.mathworks.com/help/matlab/math/solve-single-pde.html

3) Suggested programming steps:

In this task, you are encouraged to develop the Matlab code from scratch to solve this diffusion problem using PDEPE, especially if you are familiar with Matlab.

To make the programming faster, you can also develop your code by modifying the code in the example of “Solve Single PDE”.

The following code is from “Solve Single PDE”. You are welcomed to use it. Copy-n-paste the code in a newly generated script, and then save the script (e.g. pdediffusion.m).

```
x = linspace(0,1,20);
t = linspace(0,2,5);
m = 0;

% Solve the equation
sol = pdepe(m,@pdex1pde,@pdex1ic,@pdex1bc,x,t);

% Extract the first solution component from sol.
u = sol(:,:,1);

% Create a surface plot of the solution.
surf(x,t,u)
title('Numerical solution computed with 20 mesh points.')
xlabel('Distance x')
ylabel('Time t')

% Create a line plot of the solution.
Figure;
plot(x,u(end,:),'o',x,exp(-t(end))*sin(pi*x))
title('Solution at t = 2')
legend('Numerical, 20 mesh points','Analytical','Location','South')
xlabel('Distance x')
ylabel('u(x,2)')

% Equation to solve
function [c,f,s] = pdex1pde(x,t,u,DuDx)
c = pi^2;
f = DuDx;
s = 0;
end

% Initial conditions
function u0 = pdex1ic(x)
u0 = sin(pi*x);
end

% Boundary conditions
function [pl,ql,pr,qr] = pdex1bc(xl,ul,xr,ur,t)
pl = ul;
ql = 0;
pr = pi * exp(-t);
qr = 1;
end
```

Then you can start to modify this code (from top to bottom)

a) change xmesh to x = linspace(0, 0.25, 20); % total distance 0.25 cm, # of steps 20
b) change tspan to t = linspace(0, 60*3600, 11); % total time period 60 hours, # of time point 11
c) change plot(x,u(end,:),'o',x,exp(-t(end))*sin(pi*x)) to plot(x,u(end,:));
d) change title('Solution at t = 2') to title('Solution at t=60 hours');
e) delete legend('Numerical, 20 mesh points','Analytical','Location','South');
f) change ylabel('u(x,2)') to ylabel('u(x,60)');
g) change in function pdex1pde: c=1/10^-7; % (c=1/D, $D=10^{-7}$ cm^2/sec).
h) change in initial condition pdex1ic: u0=0;
i) changes in boundary condition pdex1bc: pl=4.94*10^-19; ql=10^-7; pr=ur; qr=0;
% (pl=Fp=4.94 x 10^{-19} mol/cm^2•s; ql=D=10^{-7} cm^2/sec; pr=ur; qr=0).

Save the script, and click **Run**

You can see two figures, as shown in Fig. 1 and Fig. 2 below.
Fig. 1 shows the concentration u as a function of time and distance.
Fig. 2 shows the concentration along the distance at the time of 60 hours.

To see the concentration along the distance at all time points (Fig. 3), replace plot(x,u(end,:)) with plot (x,u), then click Run again.

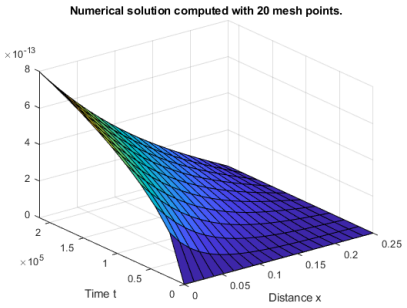

Fig. 1

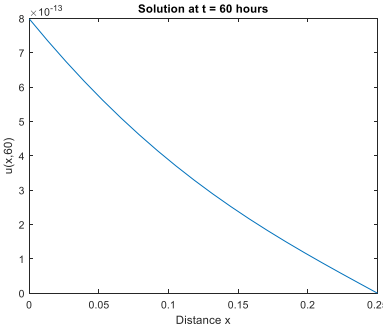

Fig. 2

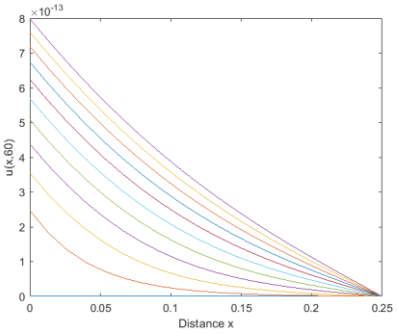

Fig. 3

4) What do we learn from this practice?

1) Each curve (concentration over distance x) follows an exponentially decay.

2) As time goes on, more and more molecules enter the medium.

3) When time is long enough, a steady state is achieved (the change of concentration over time is minimal), and the concentration u along *x* becomes a line (why)?

4) Think about what will change when the diffusion coefficient D is larger?