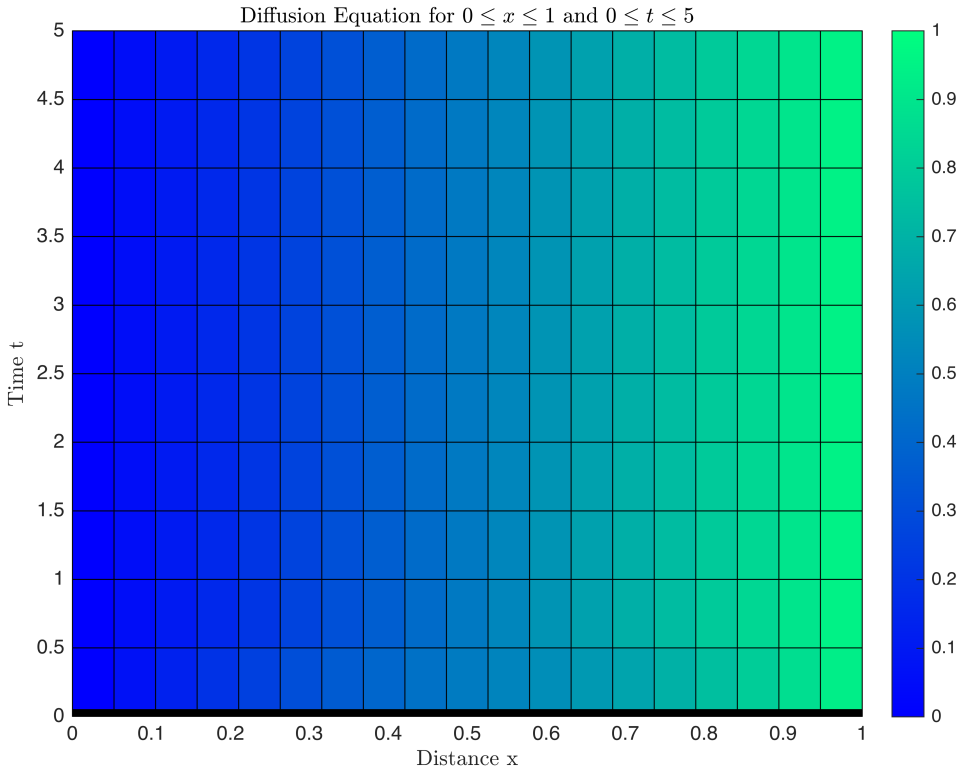
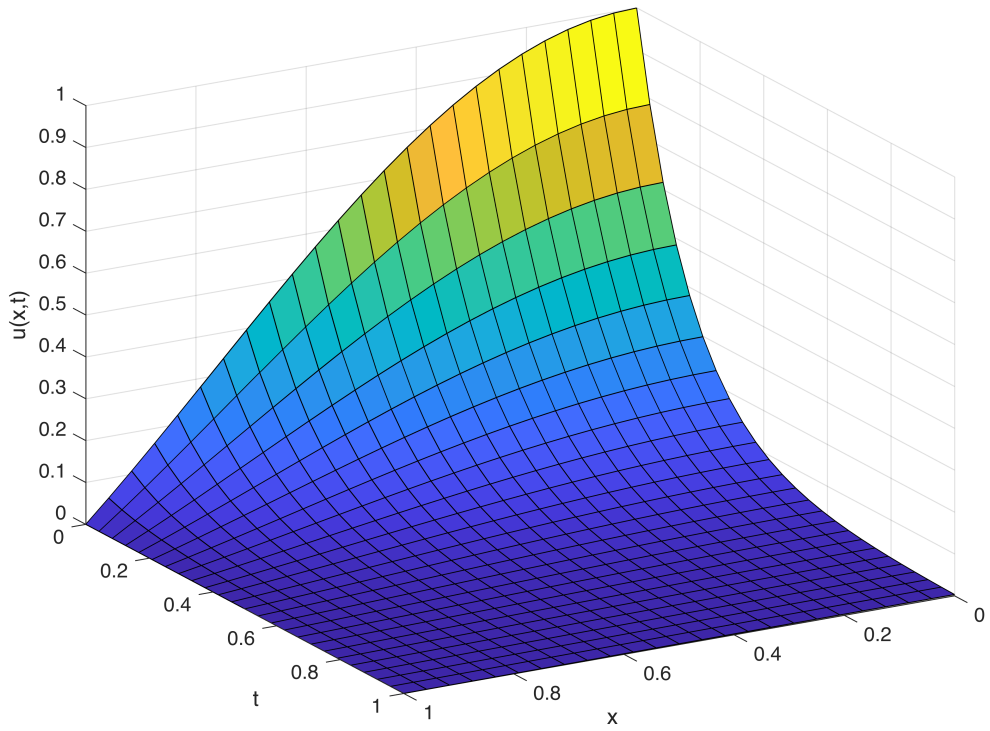
```
% Diffusion Modeling

% preliminary MATLAB code (makes liberal use of MATHWORKS code)

% 1D diffusion (constant D)

% select solution mesh
x = linspace(0,1,20);
t = [linspace(0,0.05,20), linspace(0.5,5,10)];

% implement
m = 0;
sol = pdepe(m,@diffpde,@diffic,@diffbc,x,t);

figure(1)
colormap winter
pcolor(x,t,sol)
colorbar
xlabel('Distance x','interpreter','latex')
ylabel('Time t','interpreter','latex')
title('Diffusion Equation for $0 \le x \le 1$ and $0 \le t \le
5$','interpreter','latex')
```

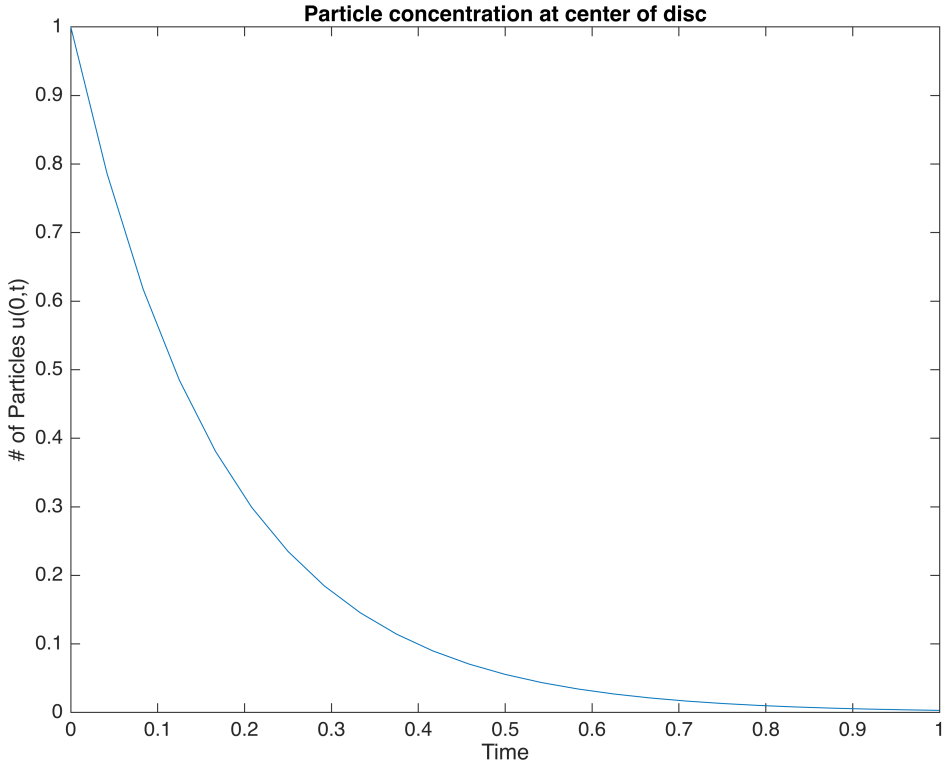



```
% 1D diffusion (cylindrical)
clear all
```

```
x = linspace(0,1,25);
t = linspace(0,1,25);
m = 1;
sol = pdepe(m,@diffcyl,@difficyl,@diffbcyl,x,t);
u = sol(:,:,1);

figure(2)
surf(x,t,u)
xlabel('x')
ylabel('t')
zlabel('u(x,t)')
view([150 25])
```




```
figure(3)
plot(t,sol(:,1))
xlabel('Time')
ylabel('# of Particles u(0,t)')
title('Particle concentration at center of disc')
```



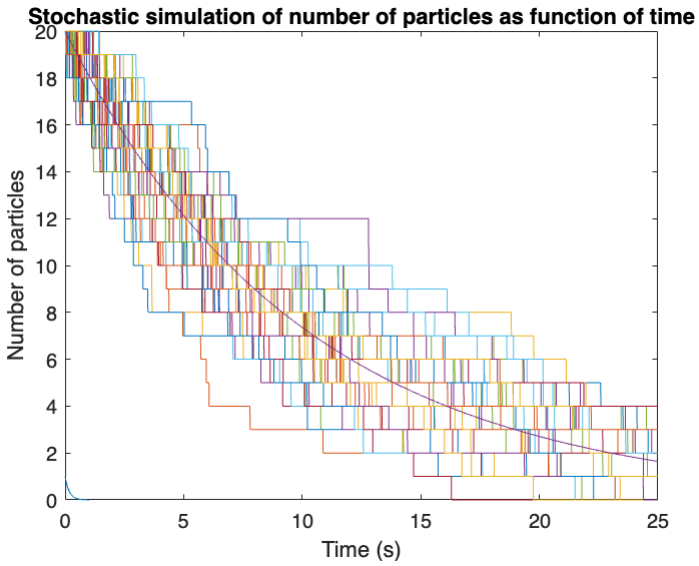
Stochastic simulation: random walk for A decay

```
% stochastic simulation of simple reaction-diffusion decay
clear all
As = zeros(1, 5000);
As(1) = 20;
delt = 0.005; % sec
ts = zeros(1, 5000);
k = 0.1 % sec^-1
```

```
k = 0.1000
```

```
for i = 1:5000
    ts(i) = 0 + delt*(i-1);
end


for i = 1:10
for j = 1:4999
    r = rand;
    if r < As(j)*k*delt
        As(j+1) = As(j)-1;
    else
```

```
        As(j+1) = As(j);
    end
end

hold on
figure(4)
plot(ts, As)
xlabel('Time (s)')
ylabel('Number of particles')
title('Stochastic simulation of number of particles as function of time')
end
plot(ts, 20*exp(-k*ts))
hold off
```




```
function [c, f, s] = diffpde(x, t, u, dudx)
c = 1; % replace with D
f = dudx;
s = 0;
end

% define initial conditions
function u0 = diffic(x)
u0 = 0.5;
end

% define boundary conditions
function [pl, ql, pr, qr] = diffbc(xl, ul, xr, ur, t)
pl = ul;
ql = 0;
pr = ur - 1;
qr = 0;
end
```

```
% defining functions for 1D cylindrical

function [c,f,s] = diffcyl(x,t,u,dudx)
c = 1;
f = dudx;
s = 0;
end
%--------------------------------------------
function u0 = difficyl(x)
n = 2.404825557695773;
u0 = besselj(0,n*x);
end
%--------------------------------------------
function [pl,ql,pr,qr] = diffbcyl(xl,ul,xr,ur,t)
n = 2.404825557695773;
pl = 0; %ignored by solver since m=1
ql = 0; %ignored by solver since m=1
pr = ur-besselj(0,n)*exp(-n^2*t);
qr = 0;
end
%--------------------------------------------
```