BEGN 221: Mathematical Methods in Bioengineering

Lecture 5

Introduction: PDEs in Linear Space and Time

References

Haberman APDE, Ch. 1.

Haberman APDE, Ch. 4.

Haberman APDE, Ch. 6.

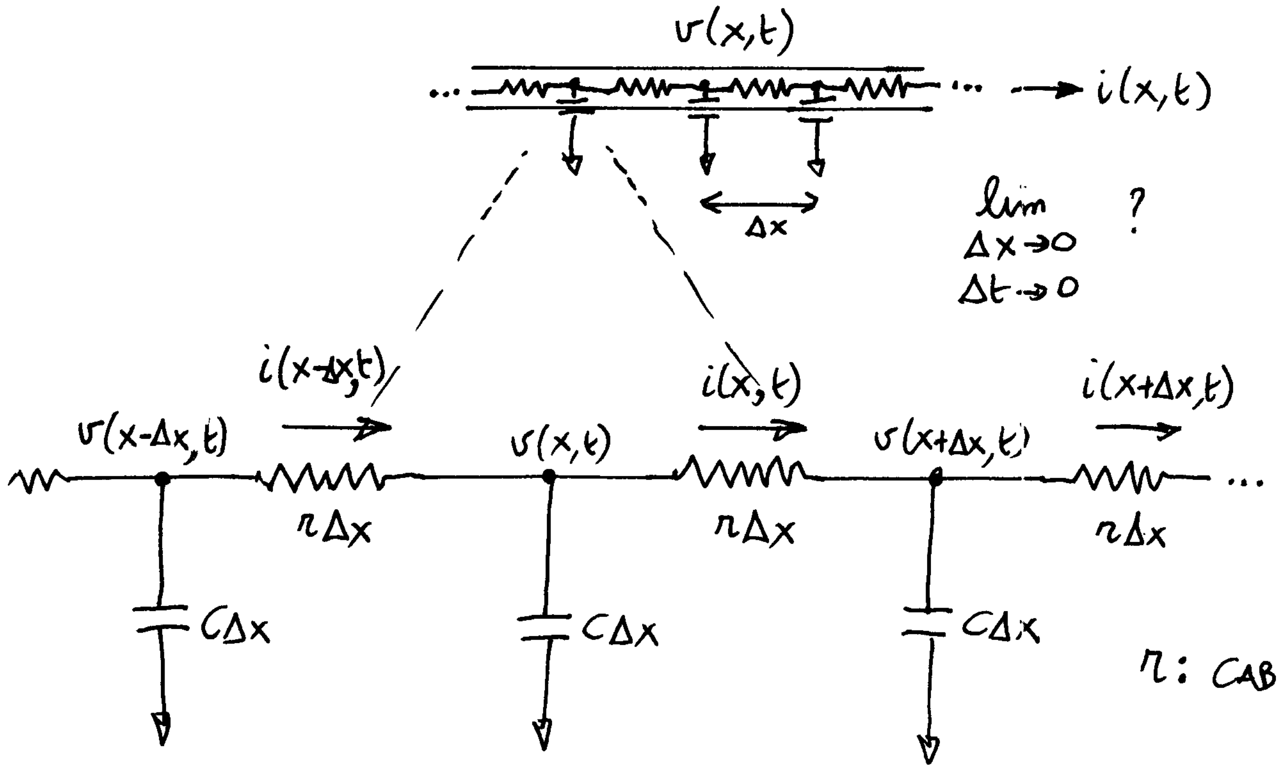
DIFFUSION EQUATION

Also POISSON'S EQUATION, or HEAT EQUATION

From lumped to continuum models

e.g.: PASSIVE CABLE — CONDUCTING WIRE

(non-myelinated axon; dendrite)



r: CABLE RESISTANCE $\left[\frac{\Omega}{m}\right]$

c: CABLE CAPACITANCE $\left[\frac{F}{m}\right]$

$$(c\Delta x)\,\frac{\Delta v(x,t)}{\Delta t} = i(x-\Delta x,t) - i(x,t) \underset{\substack{\lim\\ \Delta x\to 0\\ \Delta t\to 0}}{\Longrightarrow} c\,\frac{\partial v}{\partial t} = -\frac{\partial i}{\partial x}$$

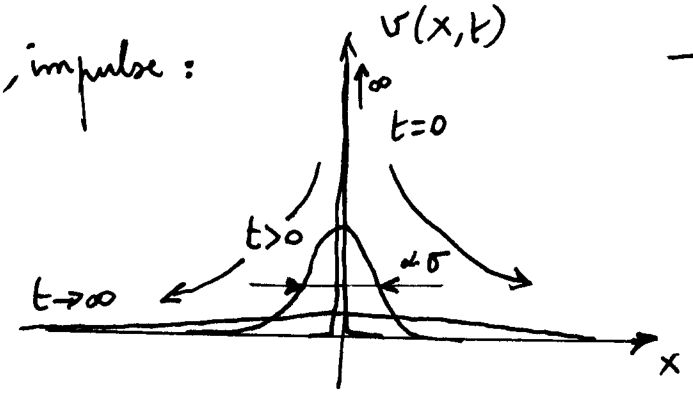
$$(r\Delta x)\, i(x,t) = v(x,t) - v(x+\Delta x,t) \underset{\substack{\lim\\ \Delta x\to 0\\ \Delta t\to 0}}{\Longrightarrow} r\cdot i = -\frac{\partial v}{\partial x}$$

$$\begin{cases} C\,\dfrac{\partial v}{\partial t} = -\dfrac{\partial i}{\partial x} \\ r\, i = -\dfrac{\partial v}{\partial x} \end{cases} \Rightarrow \quad \dfrac{\partial v}{\partial t} = D\,\dfrac{\partial^2 v}{\partial x^2}, \text{ or} \quad \dfrac{\partial i}{\partial t} = D\,\dfrac{\partial^2 i}{\partial x^2}$$

DIFFUSION EQUATION

with $D = \dfrac{1}{r.c}$ DIFFUSIVITY $\left[\dfrac{m^2}{\Omega.F}\right] = \left[\dfrac{m^2}{s}\right]$

e.g., impulse:



→ spatial variance per unit time

$$\sigma \propto \sqrt{Dt}$$

spatial extent of diffusion scales with square root of time

The particular solution (and the solution method!) depends on

INITIAL CONDITIONS e.g. I.C. $v(x,0)$

BOUNDARY CONDITIONS e.g. B.C. $\begin{cases} v(0,t) \\ v(L,t) \end{cases}$

Other physical modalities in bioengineering:
Diffusion of HEAT, or CONCENTRATION of compounds and agents

$C \leftrightarrow$ conservation (divergence) capacity $\qquad v \leftrightarrow$ temperature concentration

(Week 3)

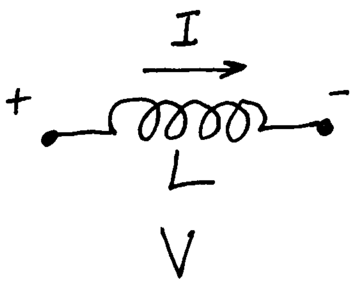
$r \leftrightarrow$ friction (Fick's law) resistance $\qquad i \leftrightarrow$ heat flow flux

WAVE EQUATION

TRANSMISSION LINE — active cable

(model of axon, or 1-D suspended membrane)

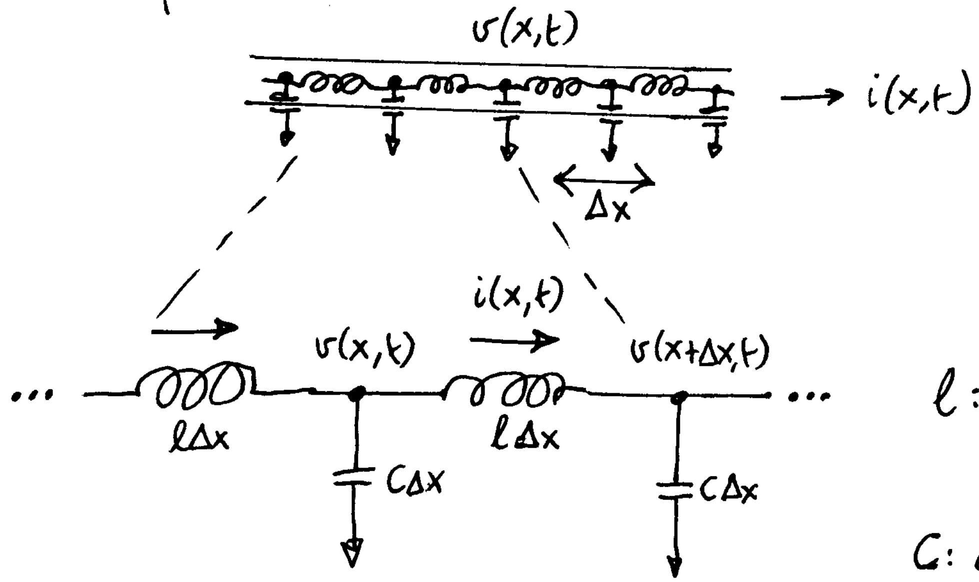
→ replace cable resistance with cable INDUCTANCE :



$$L \frac{dI}{dt} = V$$

L: INDUCTANCE $[H]$

→ lumped model :



l: CABLE INDUCTANCE $\left[\frac{H}{m}\right]$

C: CABLE CAPACITANCE $\left[\frac{F}{m}\right]$

As before: $(C\Delta x)\dfrac{\Delta v(x,t)}{\Delta t} = i(x-\Delta x,t) - i(x,t) \Rightarrow C\dfrac{\partial v}{\partial t} = -\dfrac{\partial i}{\partial x}$

$\lim_{\substack{\Delta x \to 0 \\ \Delta t \to 0}}$

Now : $(l\Delta x)\dfrac{\Delta i(x,t)}{\Delta t} = v(x,t) - v(x+\Delta x,t) \Rightarrow l\dfrac{\partial i}{\partial t} = -\dfrac{\partial v}{\partial x}$

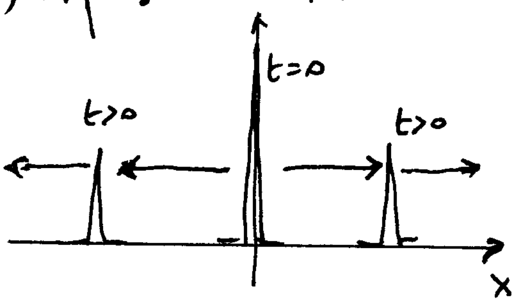
$$\begin{cases} C \dfrac{\partial v}{\partial t} = -\dfrac{\partial i}{\partial x} \\ \ell \dfrac{\partial i}{\partial t} = -\dfrac{\partial v}{\partial x} \end{cases} \Rightarrow \quad \begin{aligned} &\frac{\partial^2 v}{\partial t^2} = c^2 \frac{\partial^2 v}{\partial x^2}, \text{ or} \\ &\frac{\partial^2 i}{\partial t^2} = c^2 \frac{\partial^2 i}{\partial x^2} \end{aligned}$$

WAVE EQUATION

with $c = \dfrac{1}{\sqrt{\ell C}}$ WAVE VELOCITY

$$\left[\frac{m}{\sqrt{H.F}}\right] = \left[\frac{m}{s}\right]$$

e.g., impulse: $v(x,t)$



$x = \pm c.t$

distance traveled is proportional to time, in both directions

The particular solution depends on INITIAL and BOUNDARY CONDITIONS

e.g. initial conditions I.C. $\begin{cases} v(x,0) \\ \dfrac{\partial v}{\partial t}(x,0) \end{cases}$ (need both!)

boundary conditions B.C. $\begin{cases} v(0,t) \\ v(L,t) \end{cases}$

Other physical modalities: vibrating string, sound propagation, ...

$C \leftrightarrow$ stiffness and $v \leftrightarrow$ pressure (Week 9 & 10)

$\ell \leftrightarrow$ mass density $i \leftrightarrow$ velocity

Numerical solution of PDEs with finite differences:

From the continuum back to the lumped model:

$$(c\,\Delta x)\,\frac{v(x,t+\Delta t)-v(x,t)}{\Delta t} = i(x-\Delta x,t) - i(x,t)$$

where

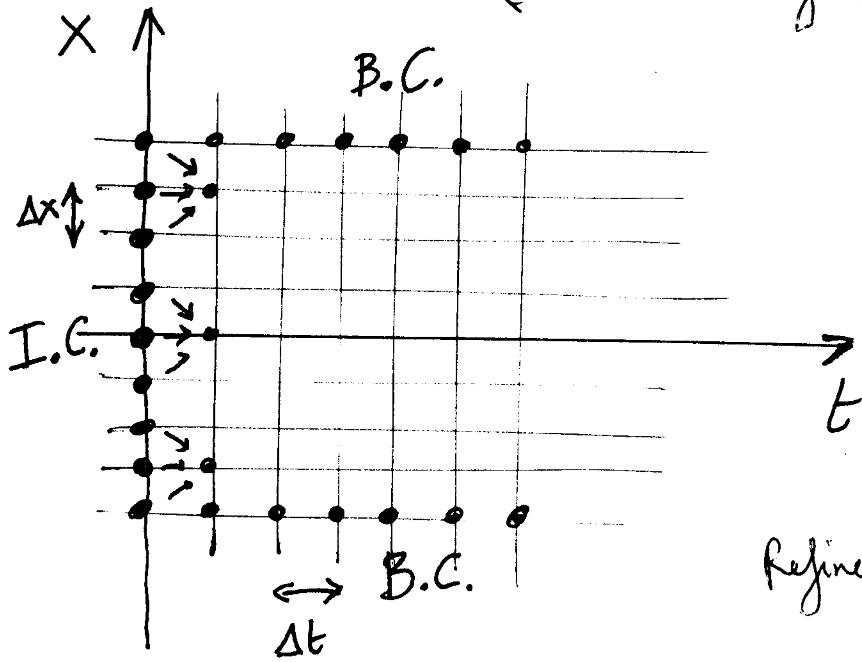
$$(r\,\Delta x)\, i(x,t) = v(x,t) - v(x+\Delta x,t)$$

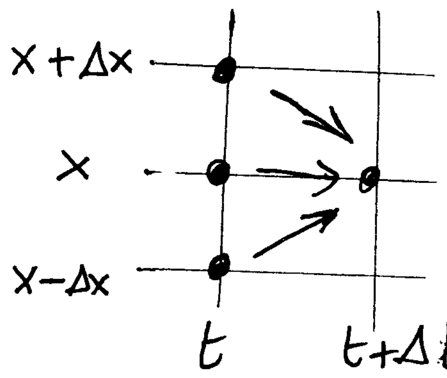
$$(r\,\Delta x)\, i(x-\Delta x,t) = v(x-\Delta x,t) - v(x,t)$$

$$\Rightarrow \quad v(x,t+\Delta t) = v(x,t) + \eta \cdot \Big(v(x+\Delta x,t) - 2\,v(x,t) + v(x-\Delta x,t)\Big)$$

with stepsize $\eta = \frac{1}{r.c}\cdot\frac{\Delta t}{(\Delta x)^2} = D\cdot\frac{\Delta t}{(\Delta x)^2}$

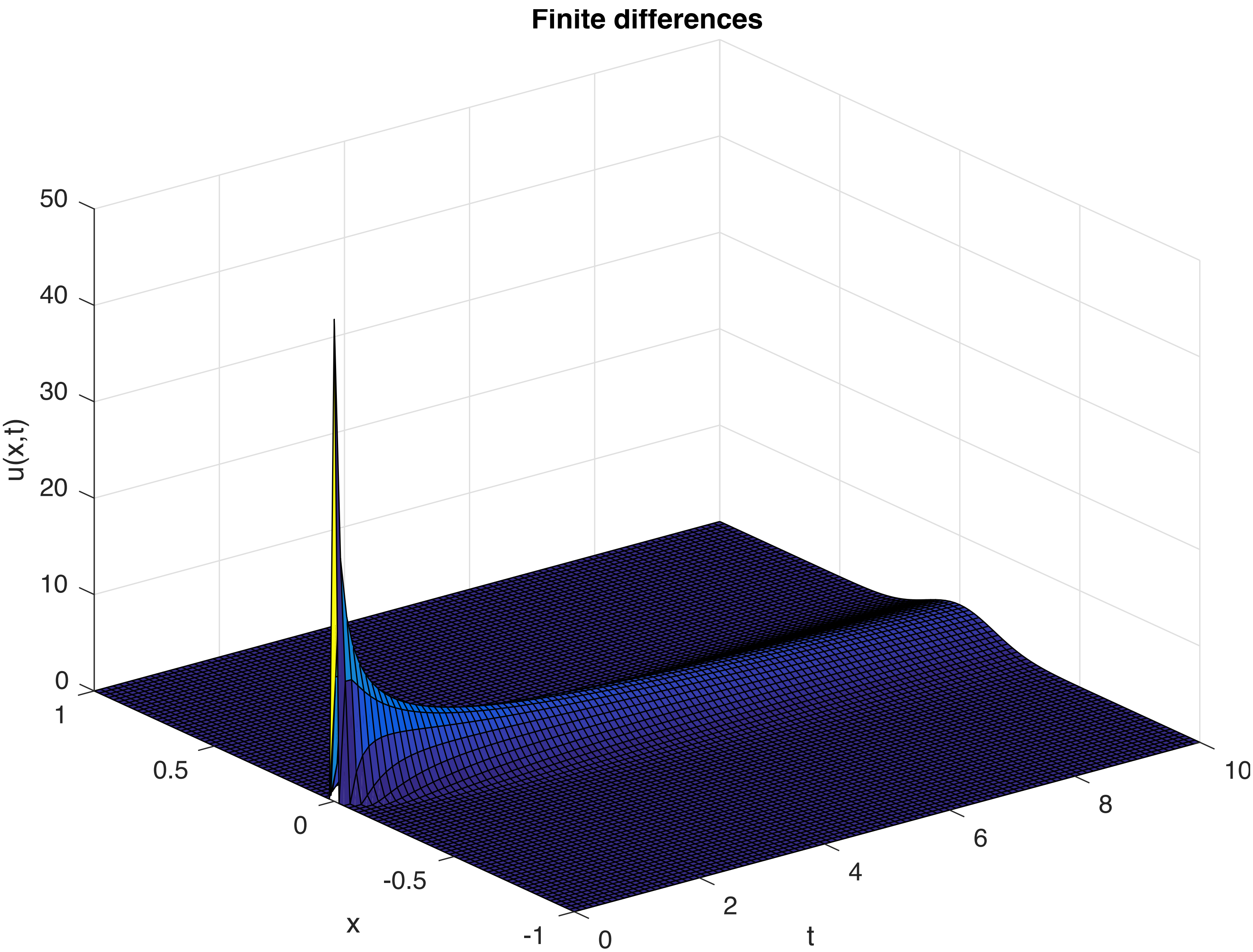
(Euler with $f(v,t) = \frac{D}{(\Delta x)^2}\Big(v(x+\Delta x,t) - 2v(x,t) + v(x-\Delta x,t)\Big)$)

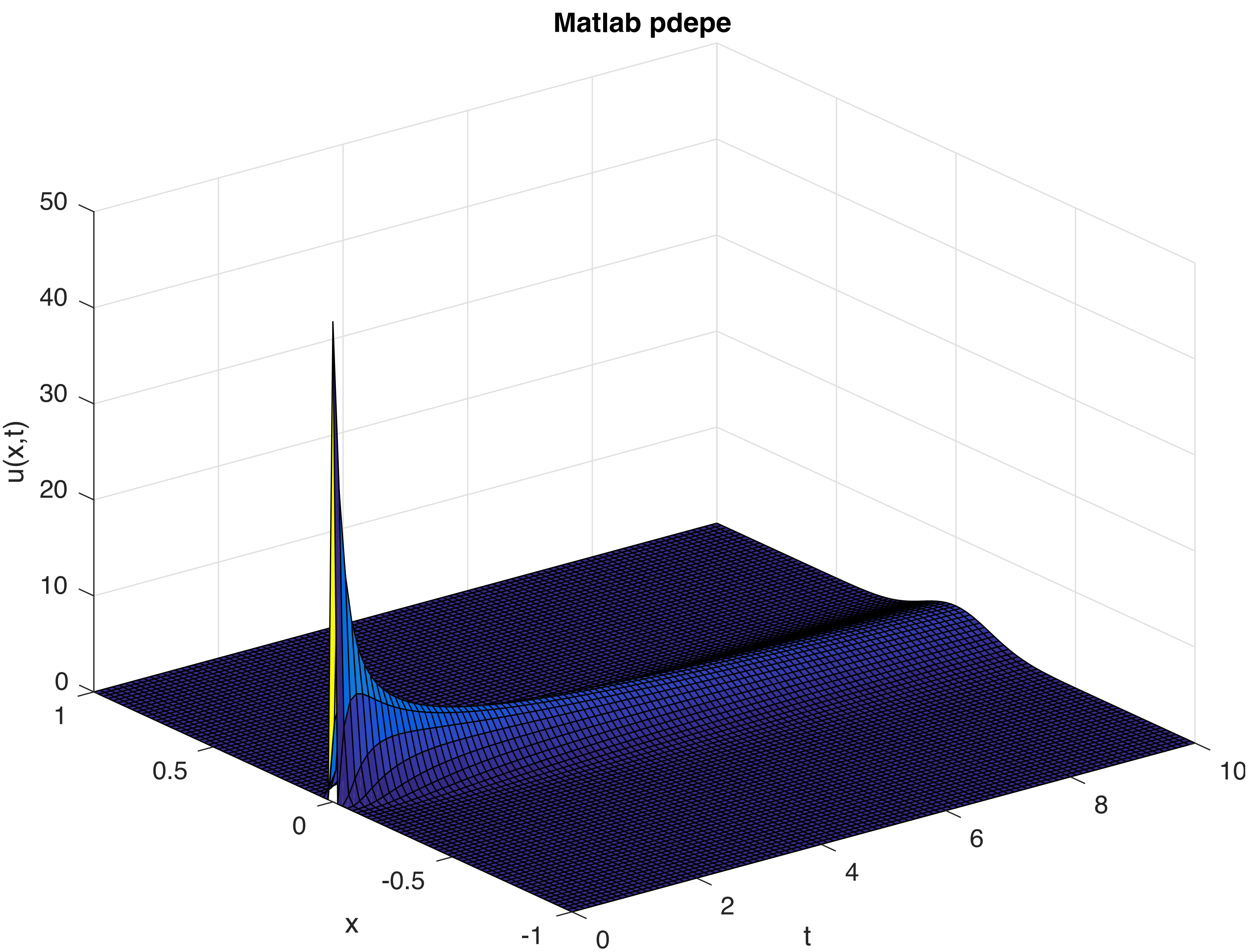




Refinements: Crank-Nicholson rather than Euler in time, etc...

```matlab
function lindiff
%%% Homogeneous PDE: Linear (1-D) Diffusion
%%% BENG 221 example, 9/29/2011, updated 11/4/2017

% diffusion constant
global D
D = 0.001;

% domain
global dx
dx = 0.02; % step size in x dimension
dt = 0.1;  % step size in t dimension
xmesh = -1:0.02:1; % domain in x
tmesh = 0:0.1:10;  % domain in t


% solution using finite differences (see Week 5 class notes)
nx = length(xmesh); % number of points in x dimension
nt = length(tmesh); % number of points in t dimension
stepsize = D * dt / dx^2; % stepsize for numerical integration
sol_fd = zeros(nt, nx);
sol_fd(1, :) = (xmesh == 0)/dx; % initial conditions; delta impulse at center
sol_fd(:, 1) = 0; % left boundary conditions; zero value
sol_fd(:, nx) = 0; % right boundary conditions; zero value
for t = 1:nt-1
    for x = 2:nx-1
        sol_fd(t+1, x) = sol_fd(t, x) + stepsize * ...
            (sol_fd(t, x-1) - 2 * sol_fd(t, x) + sol_fd(t, x+1));
    end
end

figure(1)
surf(tmesh,xmesh,sol_fd')
title('Finite differences')
xlabel('t')
ylabel('x')
zlabel('u(x,t)')


% solution using Matlab's built in "pdepe"
% See: http://www.mathworks.com/help/techdoc/ref/pdepe.html
% Also: https://mse.redwoods.edu/darnold/math55/DEProj/sp02/AbeRichards/paper.pdf
help pdepe
sol_pdepe = pdepe(0,@pdefun,@ic,@bc,xmesh,tmesh);

figure(2)
surf(tmesh,xmesh,sol_pdepe')
title('Matlab pdepe')
xlabel('t')
ylabel('x')
zlabel('u(x,t)')


% function definitions for pdepe:
% --------------------------------------------------------------

function [c, f, s] = pdefun(x, t, u, DuDx)
% PDE coefficients functions

global D
c = 1;
f = D * DuDx; % diffusion
s = 0; % homogeneous, no driving term

% --------------------------------------------------------------

function u0 = ic(x)
% Initial conditions function

global dx
u0 = (x==0)/dx; % delta impulse at center

% --------------------------------------------------------------

function [pl, ql, pr, qr] = bc(xl, ul, xr, ur, t)
% Boundary conditions function

pl = ul; % zero value left boundary condition
ql = 0;  % no flux left boundary condition
pr = ur; % zero value right boundary condition
qr = 0;  % no flux right boundary condition
```


Finite differences
u(x,t)
50
40
30
20
10
0
1
0.5
0
-0.5
-1
x
0
2
4
6
8
10
t


Matlab pdepe
u(x,t)
50
40
30
20
10
0
1
0.5
0
-0.5
-1
x
0
2
4
6
8
10
t