Exponential Growth - Simple Dynamic System

- differential equation
- dP/dt = rP
- an analytic solution exists so we can write Population as a function of time (integrating both sides)
    - P = P0 * exp(rt)

Numerical Integration

- differential equation
- $dP/dt = rP$
- We can also solve by numerical integration - less precise
- Many techniques are available to do numerical integration
- ODE - is a numerical integration tool in R - it does it for you

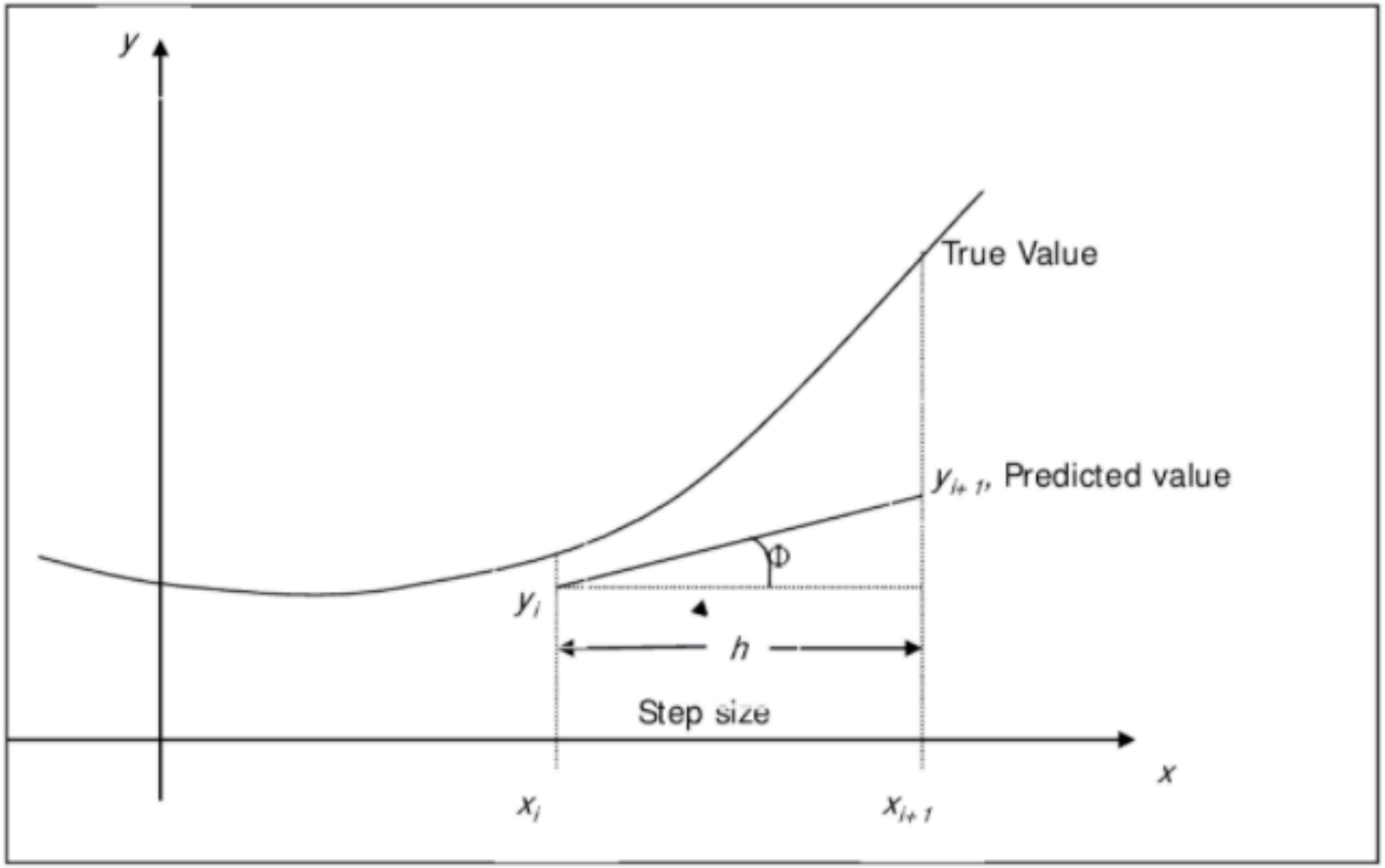
Numerical Integration

- differential equation
- $dP/dt = rP$
- We can also solve by numerical integration - less precise
- Many techniques are available to do numerical integration
- ODE - is a numerical integration tool in R - it does it for you

Basic Concept

$$y_{i+1} = y_i + f(x_i, y_i)h$$

$$h = x_{i+1} - x_i$$



You next value (e.g Population at the next time step), is
Current value + derivative at the value

But since the derivate itself changes with the value, your time steps should be infinitely small -

Numerical Integration Challenges

Dealing with the sensitivity to step size - is the main challenge

Many techniques for dealing this this - some use higher order derivatives etc.
(see see method under *help(ode)*

Which one is the best to use depends on the derivative equation - and the values that you need answer for (e.g what periods of time, for derivatives of time)

Why YOU need to know this

* explains why sometimes solves (like ODE) don't work
* Explains what some of the parameters are for the ODE solver
* Reminds you that you should seek expert help if you are having issues with your solver
* Cautions you about step size when you do integration by hand…

Example: Dynamic Systems

- Diffusion - Ficks Law
- flux (mass/time) =

  $Qx = -D (dC/dx)$

  Diffusivity (D) - depends on characteristics of the fluid (you can imagine how a pollutant might diffuse less rapidly in oil versus water)

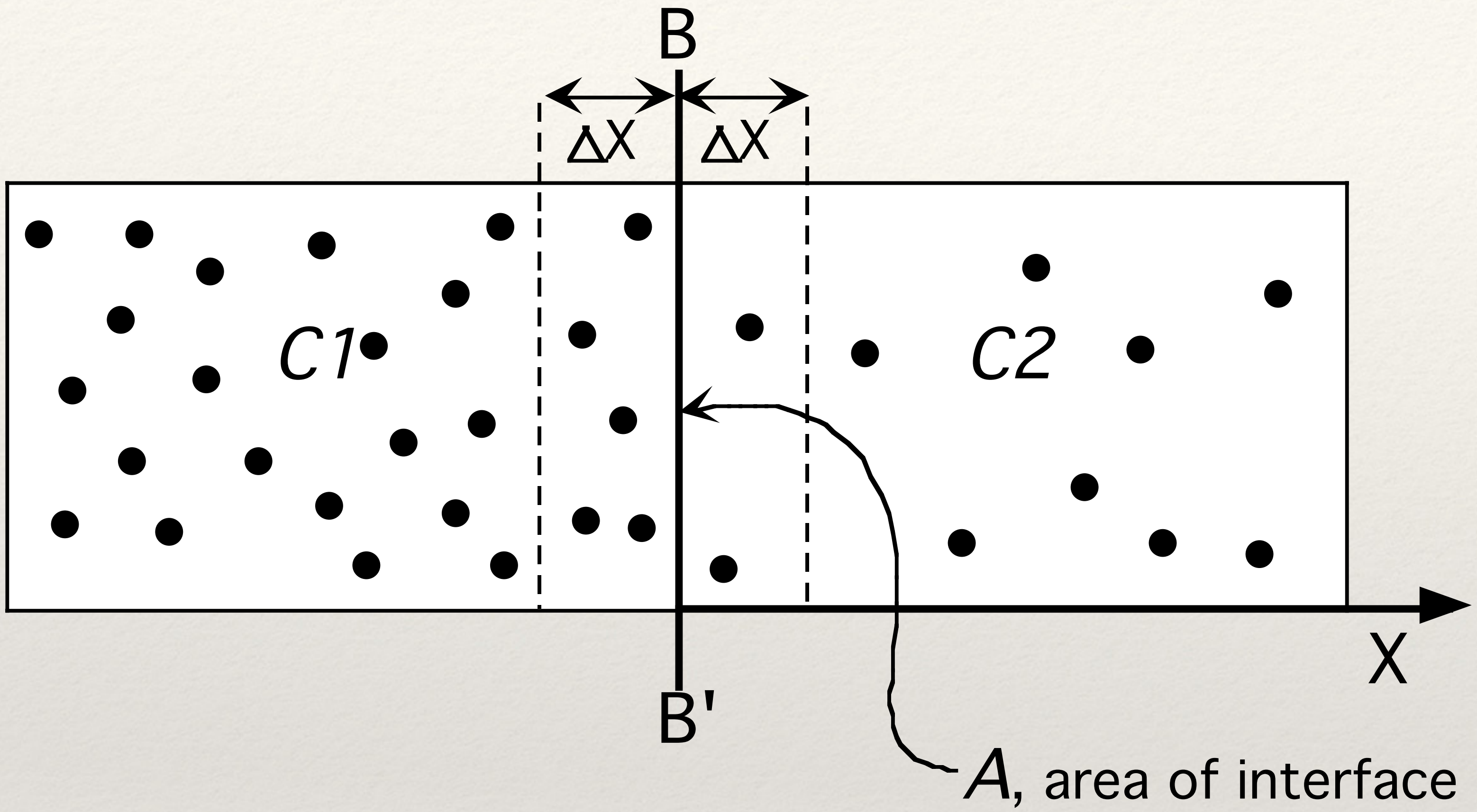
  D is sometimes broken in to D * A (where A is the cross sectional area at the boundary between the two fluids,) and D then is related to fluid type itself

  This is a first order differential equation - changes only with x

  Where x is the direction of diffusion

Diffusion

- Diffusion



How big that delta X is, depends on characteristics of the particles - often represented as a diffusivity (D) term

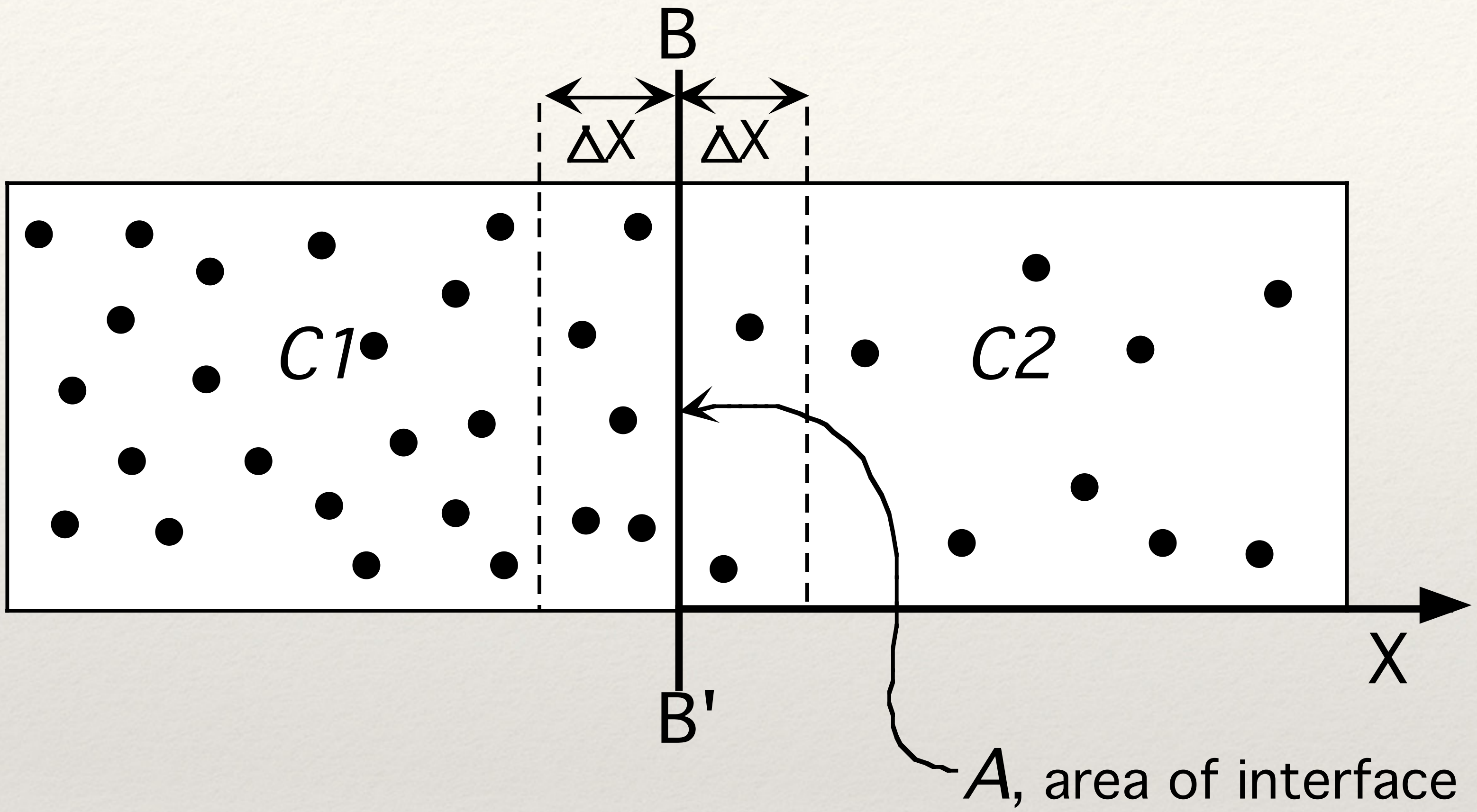
Flux at any instant is and any point along x

$$Qt = 0.5 * D*A * (C2-C1)$$

But as soon as particles move - C1 and C2 will change!!! - so we also need derivatives with respect to time

Diffusion

- Diffusion



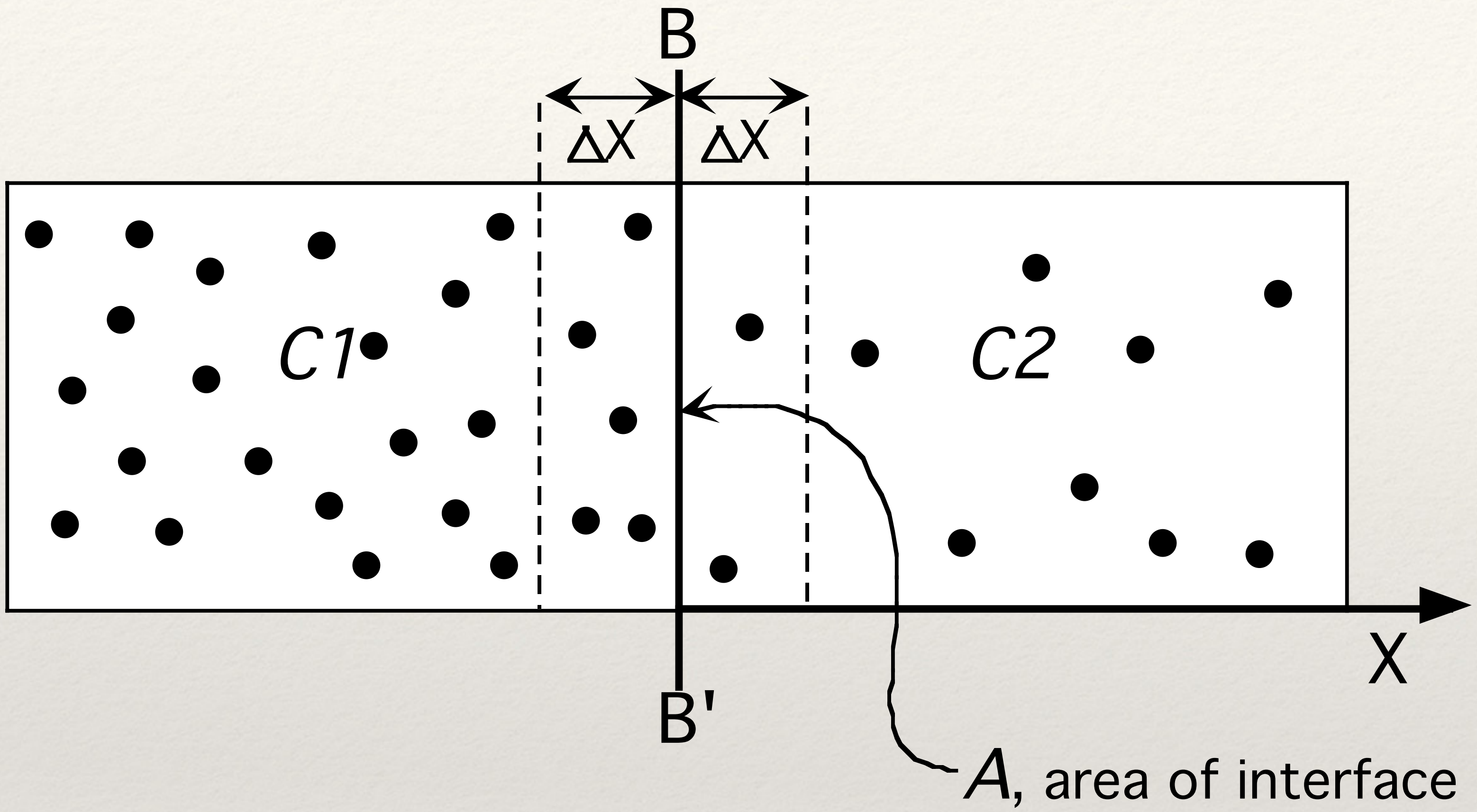
How big that delta X is, depends on characteristics of the particles - often represented as a diffusivity (D) term

Fick's Second Law: $dC/dt = D\, d^2C/dx^2$

This a partial differential equation (derivatives of with respect to t and x), there are solves for this

Diffusion: Numerical Approximation: Difference Equation

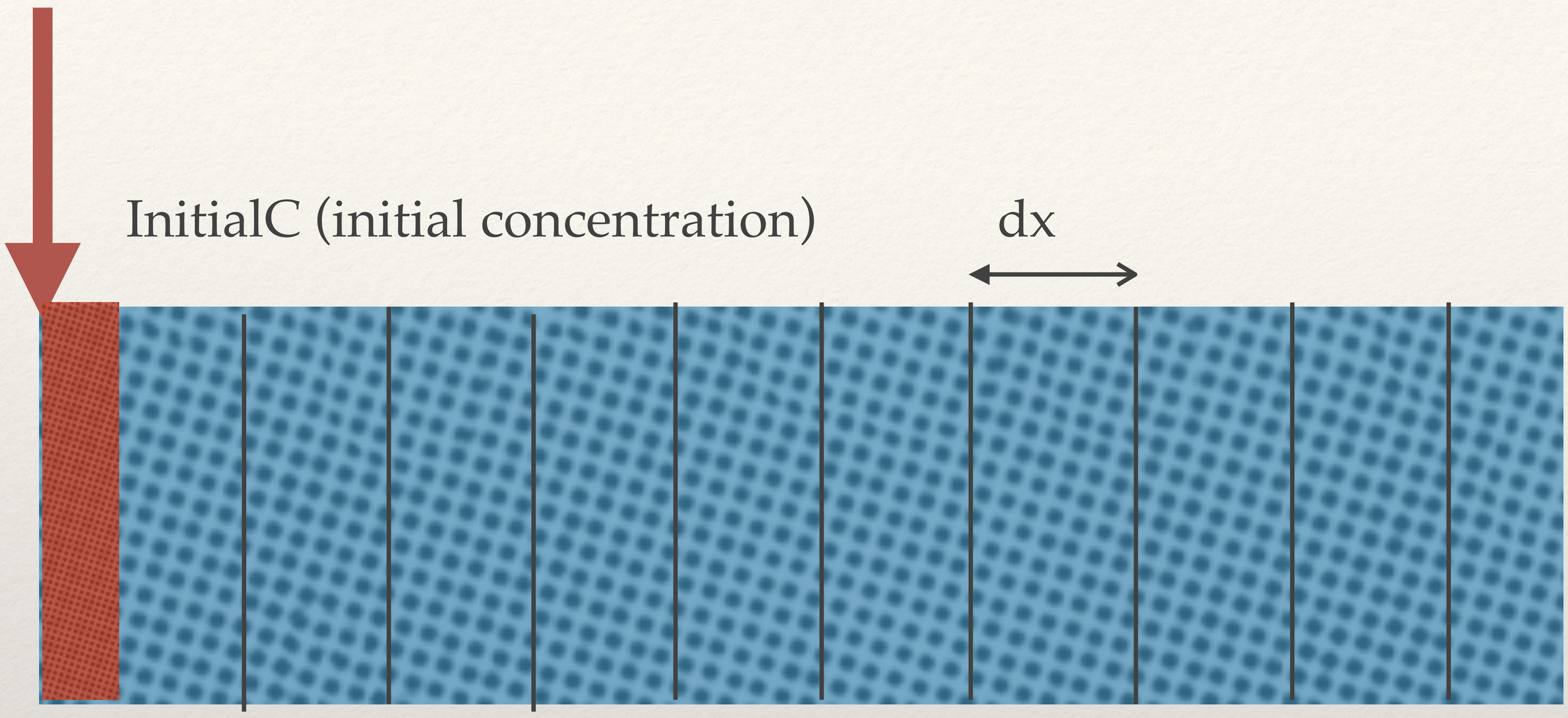
- Diffusion



BUT if we break up x into lots of little boxes, and time into small time chunks

$$Qt = 0.5 * D * A * (C_{x2} - C_{x1})$$

Diffusion as a difference equation

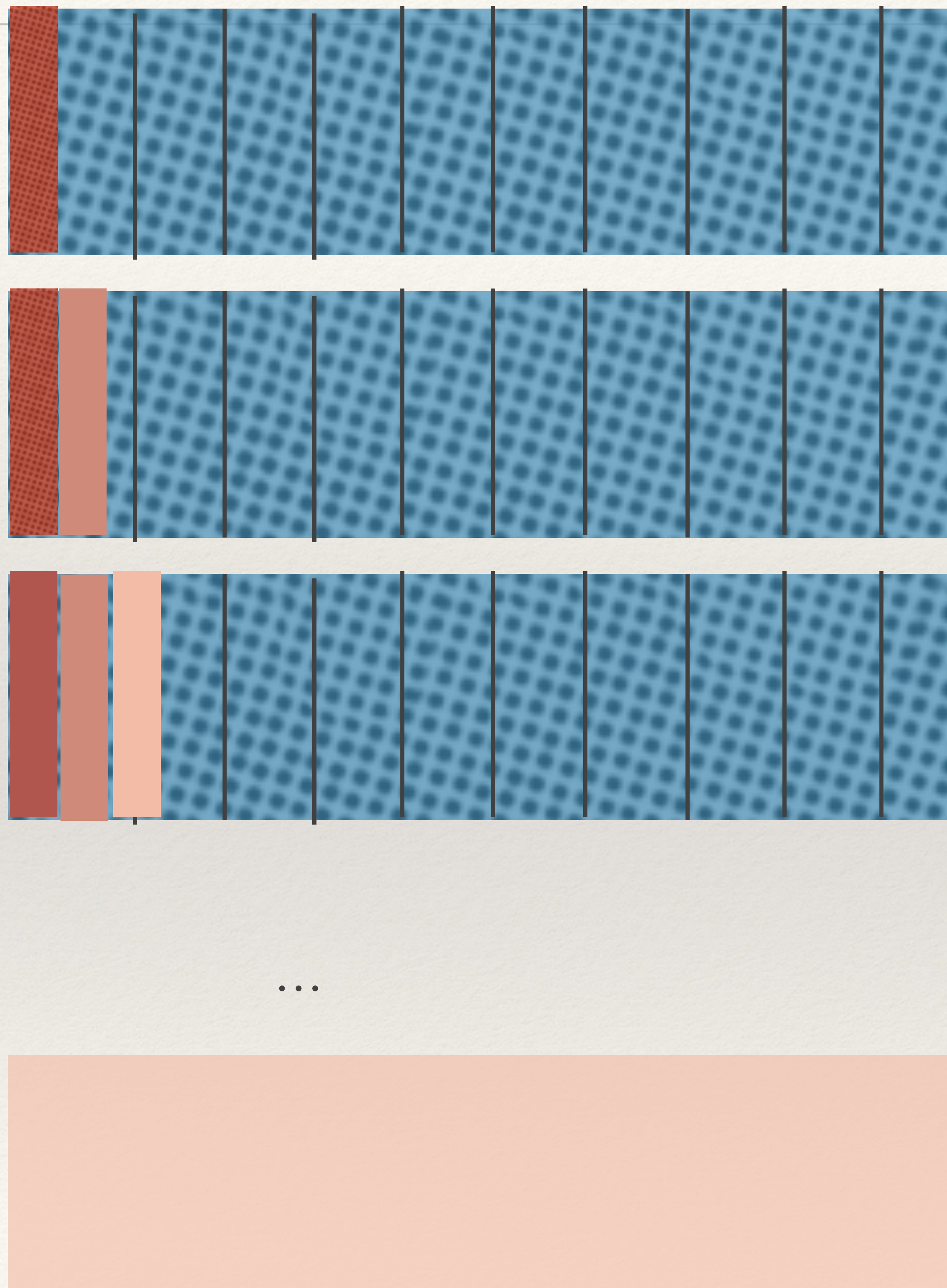


Length = nx * dx

Nx: number of boxes you are breaking space into

dx size of each box

Diffusion Example



Timestep 1

Timestep 2

Timestep 3

State evolves through time (state space trajectory)

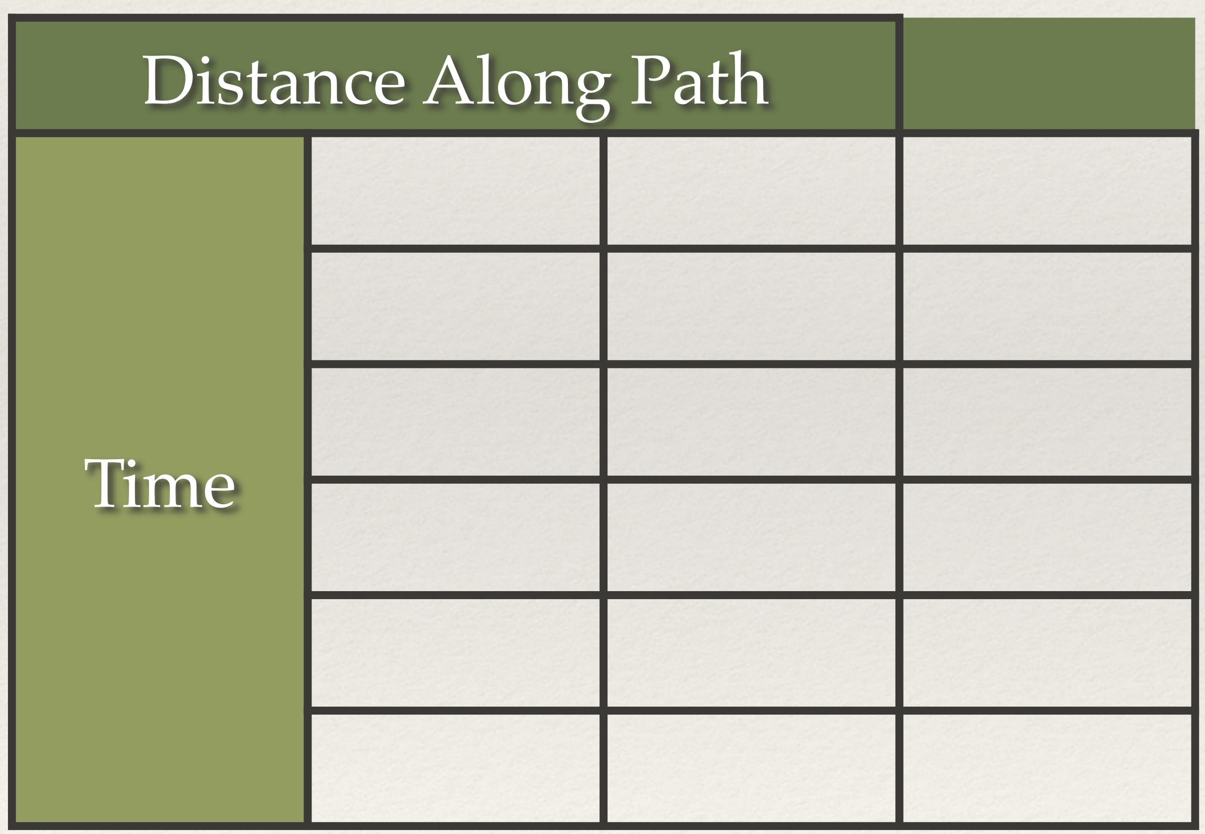
nt - number of time steps

dt - size of time step

Total Simulation time = nt*dt

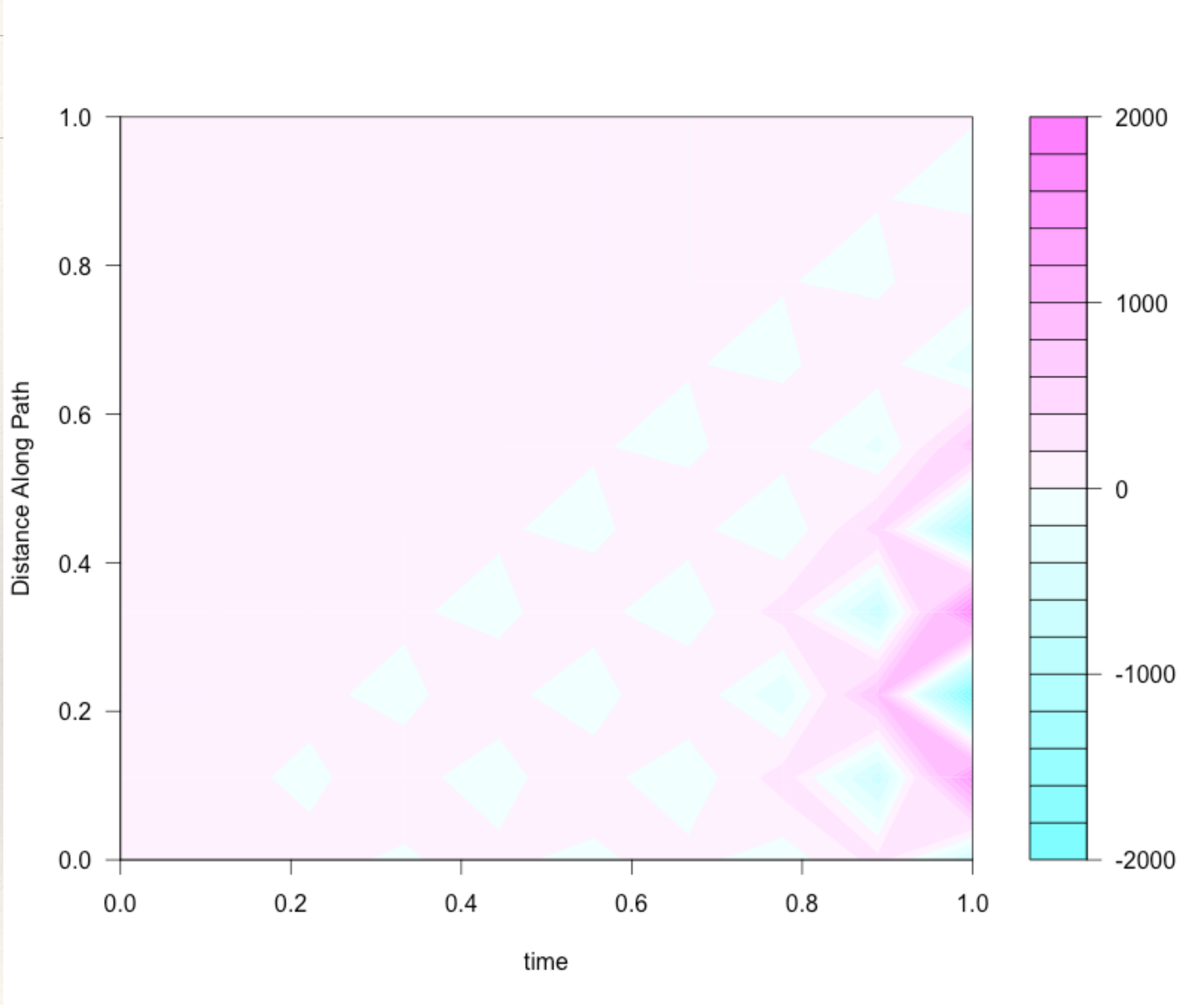
Difference equation in R

- data structure to store the state (Concentration at each point and time)
- 2-d array (rows are time, columns are distance along path)
- use it to track concentration through time



Playing with the diffuse model

- vary the time and space steps - what happens? why?
- vary other parameters (diffusivity, initial concentration)



filled.contour(diff1(10,10,1,10,30,0.6,1)$conc, xlab="time",ylab="Distance Along Path")

Dynamic - Diffusion modeling

- Choosing the appropriate time and space step is important - if either are too large then it is easy to overshoot and create unstable oscillations
- These are a due to using a discrete/difference model (dividing things into units) to model what is actually a continuous process
- Trade-off - computational efficiency vs stability
- Differential equations help us to think through this problem - but often implemented in discrete ways