Diffusion Models

Particles on a Line

Particles on a line:

- Consider a simple 1D array of compartments.
- Each compartment contains a large number of particles.
- At each time step, particles randomly move to the left or right compartment that they are in.

Observe:

- Each particle individually performs a random walk.
- On average, the number of particles moving into a box is proportional to its neighbors.

Discrete Buckets

$$\dot{z}(x) = \alpha z(x+1) + \alpha z(x-1) - 2\alpha z(x)$$

Infinitesimal Buckets

Buckets of size $dx \to 0$:

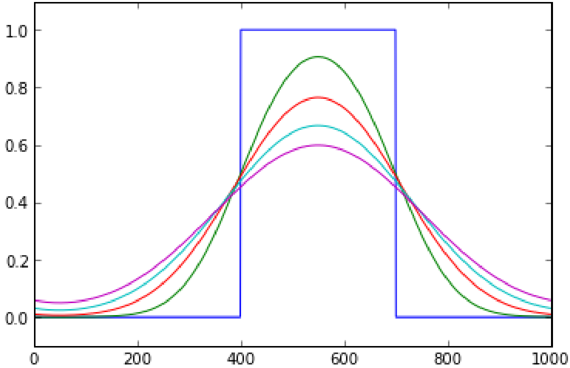
$$\dot{z}(x) = \frac{1}{dx}(\alpha z(x + dx) + \alpha z(x - dx) - 2\alpha z(x))$$

Partial Differential Equations:

$$\frac{\partial z}{\partial t} = \frac{\partial^2 z}{\partial x}$$

Simulation of Diffusion Equation

```
z = zeros(1000); z[400:700] = 1.0
alpha = -0.1
ylim((-.1,1.1))
for i in range(200000):
    if i%40000==0: plot(z)
    dz = alpha * (2*z-roll(z,1)-roll(z,-1))
    z += dz
```



Types of Differential Equations

Boundary Conditions

Initial Value Problem:

- given $z(x, t = 0)$, compute $z(x, t)$ for other times
- discretize x and t , first compute estimate of $\frac{\partial z}{\partial x}$, then $\frac{\partial z}{\partial t}$

Boundary Value Problem:

- given $z(x, y)$ on some curve/boundary $(x, y) \subseteq \mathbb{R}^2$
- alternatively, may also be given the values of partial derivatives
- compute $z(x, y)$ on the interior of that curve

Other Classification

Hyperbolic equation (wave equation):

$$\frac{\partial^2 u}{\partial t^2} = v^2 \frac{\partial^2 u}{\partial x^2}$$

Parabolic equation (diffusion equation:

$$\frac{\partial u}{\partial t^2} = \frac{\partial}{\partial x}(D\frac{\partial u}{\partial x})$$

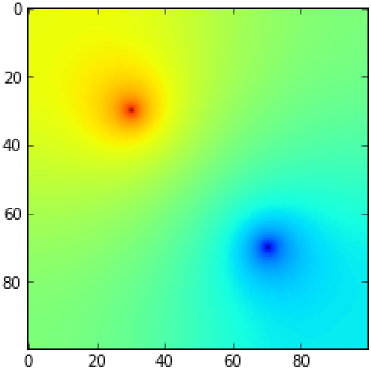
Elliptic equation (Poisson equation):

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \rho(x, y)$$

Relaxation, Diffusion, and Poisson Equations

```
from scipy.ndimage import filters
l = 0.2
u = zeros((100,100))
for t in range(20000):
    u[30,30] = 1; u[-30,-30] = -1
    u = (1-l)*u+l*filters.uniform_filter(u,(3,3))
```

```
imshow(u)
```



This is basically the diffusion equation

- compute the average of the surround
- subtract the value at the center
- if there is a difference, update the function value accordingly to push it closer to the mean
- in addition the the regular diffusion equation, we also fix some boundary conditions

$$\frac{\partial u}{\partial t} = \alpha \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

Diffusion Equation vs Poisson Equation

The limit as $t \to \infty$ of the diffusion equation gives us the Poisson equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \rho(x, y)$$

Therefore, relaxation is:

- a difference equation approximation to the diffusion equation
- a scheme for computing a static solution to the Poisson equation with boundary conditions