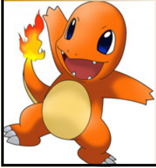
Lecture 6:
The Heat Equation



PDEs

- A differential equation (DE) is any equation that contains a derivative.

$$\text{e.g.} \quad \frac{du}{dx} = x^2 \sec(3u)$$

- A partial differential equation (PDE) contains derivatives with respect to different variables.

$$\text{e.g.} \quad \frac{\partial u}{\partial x} - 4\frac{\partial^2 u}{\partial y^2} = x^2 + 3y$$

- For short, we like to use subscripts for our derivatives.

$$\text{e.g.} \quad u_x - 4u_{yy} = x^2 + 3y$$

- PDEs are very useful in image processing, as we will see over the next few weeks. Today we are going to focus on one important PDE: The Heat Equation.

The 1D Heat Equation

- Suppose we watch heat dissipate through a metal bar of length L.

- Let u(x,t) be the temperature of the bar at position x and time t. The heat dissipates according to the PDE:

$$\boxed{u_t = \frac{\partial}{\partial x}(K\, u_x)}$$

Thermal diffusivity *(conductivity)*

Boundary Conditions

- We have to specify boundary conditions (BCs) at the ends of the bar x=0 and x=L.
- Dirichlet BCs would clamp the ends of the bar at a specific temperature:

$$u(0,t) = a, \qquad u(L,t) = b$$

- Periodic BCs would assume the left and right ends are the same:

$$u(0,t) = u(L,t)$$

- Neumann BCs assume the ends are insulated, so the change over time is zero:

$$u_t(0,t) = u_t(L,t) = 0$$

Isotropic Heat Diffusion

- If the conductivity K is a constant throughout the bar, we can pull it out of the derivative.

$$\boxed{u_t = \frac{\partial}{\partial x}(K\, u_x) = K \frac{\partial}{\partial x}(u_x) = K u_{xx}}$$

- We refer to this an isotropic diffusion, since the heat flows evenly at all points along the bar.

$$u_t = K\, u_{xx}$$

- For example, if the bar was made out of the same material throughout we would expect K to be a constant.

Anisotropic Heat Diffusion

- If K(x) changes throughout the bar, then heat will flow unevenly and we call this anisotropic diffusion.
- Since K is no longer a constant, now we cannot pull the function K out of the derivative.

$$u_t = \frac{\partial}{\partial x}(K\, u_x)$$

- For example, if the bar was made out of different materials we would expect the conductivity to change along the bar.

Divergence

- The <u>divergence</u> operator $\nabla \cdot$ is defined by summing the partial derivatives of respective components:

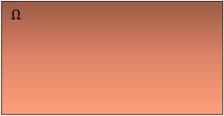
$$\vec{v} = \langle v_1, v_2 \rangle \quad \Longrightarrow \quad \boxed{\nabla \cdot \vec{v} = \frac{\partial v_1}{\partial x} + \frac{\partial v_2}{\partial y}}$$

- <u>Ex</u> Let $\vec{w} = \langle x^2 y, 3x\cos(4y) \rangle$.

  Compute the divergence of $\vec{w}$.

The 2D Heat Equation

- Suppose we keep track of the temperature on a hot plate on some rectangular domain Ω: $u(x,y,t)$.



- The Heat Equation extends to 2D by taking the divergence of K times the gradient.

$$\boxed{u_t = \nabla \cdot (K \nabla u)}$$

Isotropic 2D Heat Equation

- If K is a constant, we can pull it out of the derivatives to get a much simpler PDE.

$$\begin{aligned} u_t &= \nabla \cdot (K\nabla u) \\ &= K\nabla \cdot (\nabla u) \\ &= K\left[\frac{\partial}{\partial x} u_x + \frac{\partial}{\partial y} u_y\right] \\ &= K\left[u_{xx} + u_{yy}\right] \end{aligned}$$

- Recall we defined the Laplacian operator

$$\Delta u = u_{xx} + u_{yy}.$$

- So the Isotropic Heat Equation is

$$\boxed{u_t = K\Delta u}$$

Spatial Discretization

- Last week, we discussed how to approximate spatial derivatives using finite differences.

$$u_x \approx \frac{u(x+1,y)-u(x-1,y)}{2} \qquad u_y \approx \frac{u(x,y+1)-u(x,y-1)}{2}$$

$$u_{xx} \approx u(x+1,y) - 2u(x,y) + u(x-1,y)$$
$$u_{yy} \approx u(x,y+1) - 2u(x,y) + u(x,y-1)$$

- Note this is where the Neumann BCs come in!

$$\frac{\partial u}{\partial \vec{n}} = 0$$

- So we can use this idea to approximate u_{xx} and u_{yy} in the right-hand side our PDE.

$$u_t = K\left[u_{xx} + u_{yy}\right]$$

- But what about the left-hand side?

Temporal Discretization

- We can also discretize our temporal (time) derivative based on some small time step Δt. If we take a backward difference in t we get:

$$u_t(x, y, t + \Delta t) \approx \frac{u(x, y, t + \Delta t) - u(x, y, t)}{\Delta t}$$

- We will evolve the temperature over time. Let $u^n(x, y)$ denote the n-th iteration at time $t = n\,\Delta t$.
- Then we can write our discretization a little more simply as:

$$\boxed{u_t^{n+1} \approx \frac{u^{n+1} - u^n}{\Delta t}}$$

Discretization of the PDE

- So our PDE

$$u_t = K\left[u_{xx} + u_{yy}\right]$$

is discretized as

$$\frac{u^{n+1} - u^n}{\Delta t} = K\left[u_{xx}^n + u_{yy}^n\right]$$

where the spatial derivatives are approximated as

$$u_{xx}^n = u^n(x+1,y) - 2u^n(x,y) + u(x-1,y)$$
$$u_{yy}^n = u^n(x,y+1) - 2u^n(x,y) + u(x,y-1)$$

PDE Iteration

- We want to evolve the equation over time:

$$\boxed{\frac{u^{n+1} - u^n}{\Delta t} = K\left[u_{xx}^n + u_{yy}^n\right]}$$

- If we solve for u^{n+1} we get:

$$\boxed{u^{n+1} = u^n + K\,\Delta t\left[u_{xx}^n + u_{yy}^n\right]}$$

- This tells us how to update our u(x,y,t) at each iteration.

Forward Euler

- Suppose we are given an initial temperature distribution f(x,y).

$$u(x, y, 0) = f(x, y)$$

- We set our initial guess for u=f and evolve it over time.

  Initialize: $u^0 = f$

  Update: $u^{n+1} = u^n + K\,\Delta t\,\left[u_{xx}^n + u_{yy}^n\right]$

- We stop the process when we reach some stopping time T or when u(x,y) no longer seems to be changing.
- This process is known as explicit Forward Euler evolution. It is the simplest way to solve the Heat Equation numerically.

Coding Isotropic Diffusion
(See code on the couse page)

- Let's code this!

  Initialize: $u^0 = f$

  Update: $u^{n+1} = u^n + K\,\Delta t\left[u_{xx}^n + u_{yy}^n\right]$

- Write a Matlab function that takes a 2D matrix f as input and evolves the Heat Equation.

- Let's use the parameters:

  Conductivity K = 1

  Stopping time T=20

  Time step $\Delta t = 0.1$ *(Try playing with this value.)*

Stability

- We say a numerical algorithm is <u>stable</u> if the iterates u^n stay bounded.
  $$\lim_{n\to\infty} u^n(x,y) < C \quad \text{for some constant } C$$
- To make the algorithm fast, we want to make the time step Δt as large as possible.
- But if Δt, then this is no longer a valid approximation of the derivative and the algorithm becomes unstable.
- Von Neumann analysis of the 2D Heat Equation shows Forward Euler is stable as long as
  $$\boxed{K\,\Delta t < \frac{1}{4}}$$

Isotropic Diffusion

- Isotropic heat diffusion looks just like blurring the image with a Gaussian filter.
- In fact, the Gaussian blurred image is a solution of the Heat Equation.

$$u(x, y, t) = G_t * f(x, y)$$

- Blurring is a nice way to smooth the image and get rid of noise. But we don't want to blur the edges of an image.

Anisotropic Diffusion

- So maybe our conductivity value should change throughout the image.

- Ideally we want K(x,y) to be an "edge-stopping" function:

  K(x,y) = 1 in flat regions

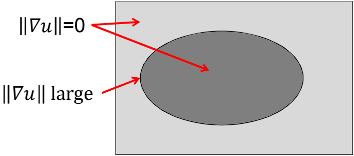
  K(x,y) = 0 on edges

- Then the diffusion will stop at the edges of the image.

$$u_t = \nabla \cdot (K \nabla u)$$

Anisotropic Diffusion

- *(Perona-Malik, 1990)* had the idea to use anisotropic diffusion where the K value is tied to the gradient.
- Recall the norm of the gradient is zero in flat regions and large at edges.



Edge-Stopping Functions

K =

K =

Coding Anisotropic Diffusion

- We evolve the PDE:

$$u_t = \nabla \cdot (K \nabla u) = \frac{\partial}{\partial x}(K u_x) + \frac{\partial}{\partial y}(K u_y)$$

- So our basic algorithm is:

  Initialize: $u^0 = f$

  Update: $\boxed{u^{n+1} = u^n + \Delta t \left[\frac{\partial}{\partial x}(K u_x^n) + \frac{\partial}{\partial y}(K u_x^n)\right]}$

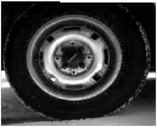
- Recall we derived the formula for u_{xx} by doing a forward then a backward difference: $u_{xx} \approx D_x^-(D_x^+ u)$.
- Now we first calculate the matrix K, multiply this times $D_x^+ u$, and then take a backward difference of the product:

$$\boxed{\frac{\partial}{\partial x}(K u_x) \approx D_x^-(K\, D_x^+ u)}$$

Isotropic vs. Anisotropic

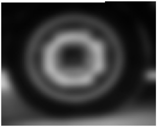
Isotropic $u_t = \Delta u \qquad (K = 1)$

t=0



t=5

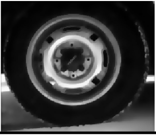
t=20



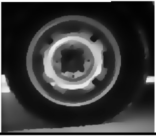
Anisotropic $u_t = \nabla \cdot (K \nabla u) \qquad K = \frac{1}{1 + (\|\nabla u\|/a)^2} \qquad a = 10$

t=0

t=5



t=20



$$K = \frac{1}{1 + (\|\nabla u\| \,/\, a)^2}$$

Original Barbara

a=1

a=2

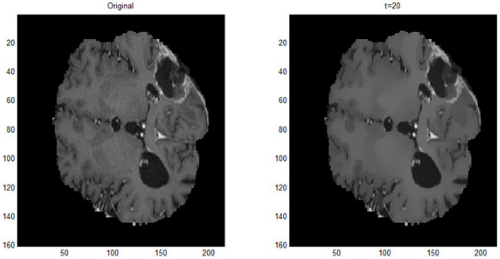
a=5

a=10

a=100

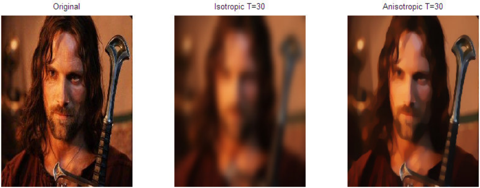
Medical Imaging

- In medical imaging, anisotropic diffusion may help define shapes and objects.



Color Diffusion

- To process a color image, we just run the diffusion on each of the 3 RGB bands.



- Anisotropic diffusion turns photos into cartoons!