$u_{xx} = \lambda u$ depends on sign of λ

$$e^{\pm\sqrt{\lambda}\, x} \qquad \lambda \geq 0$$

$$\cos(\sqrt{-\lambda}\, x) \qquad \sin(\sqrt{-\lambda}\, x) \qquad \lambda < 0$$

But to set $u(0) = 0$, $u(1) = 0$

only $\lambda < 0$ works with $u = \sin(k\pi x)$

eigenfunction $\qquad \lambda = -k^2\pi^2$

$$\underbrace{e^{-k^2\pi^2 t}\sin(k\pi x),}_{\text{solution of heat equation.}}$$

$k=1$ $\qquad$ $k=2$ $\qquad$ $k=3$

decay: $-\pi^2$ $\qquad$ $-4\pi^2$ $\qquad$ $-9\pi^2$

A $\quad u = \sum_{k=1}^{n} c_k e^{-k^2\pi^2 t}\sin(k\pi x)$ solves PDE, BC's

with initial cond $\sum_{k=1}^{n} c_k \sin(k\pi x)$.

Morally, one would like to start with any u_0, and write

$$u_0 = \sum_{k=1}^{\infty} a_k \sin(k\pi x)$$

the sum to ∞ makes this subtle.

What does "=" mean?

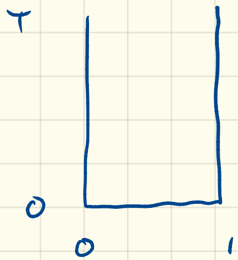
One hopes

$$u = \sum_{k=1}^{\infty} a_k e^{-k^2\pi^2 t} \sin(k\pi x)$$ solves the PDE.

Finding conditions to justify this procedure is the domain of Fourier analysis, which is too far afield.

Maximum principle for heat equation:

"under the forward flow in time heat can't concentrate"



$\Omega = [0,1] \times [0,T]$

$\partial\Omega$ is boundary

$\partial\Omega^*$ is boundary except for

$\{t = T,\ x \in (0,1)\}$

Weak maximum principle:

If $u_t - u_{xx} \leq 0$ then $\max_{\Omega} u = \max_{\partial\Omega^*} u$.

Cor: if $u_t - u_{xx} \geq 0$ then $\min_{\Omega} u = \min_{\partial\Omega^*} u$.

Cor: if $u_t - u_{xx} = 0$, u achieves both its max and min on $\partial\Omega^*$

Cor: $u_t - u_{xx} = f$, $u|_{t=0} = u_0$ + dirichlet BC's has at most one solution; $v = u_1 - u_2$ has $v_t - v_{xx} = 0$, $v|_{\partial\Omega^*} = 0$.

Pf: We first show the property holds if $u_t - u_{xx} < 0$ everywhere in interior.

At a point in $\Omega \setminus \partial\Omega^+$ where a max is achieved,

$u_t \geqslant 0$

← uses not at $t=0$

$u_x = 0$

$u_{xx} \leqslant 0.$

] uses not on space boundary

So $u_t - u_{xx} \geqslant 0$ at this point

But no such point exists.

Now suppose only $u_t - u_{xx} \leqslant 0$.

Let $v_\varepsilon = u - \varepsilon t$

So $(v_\varepsilon)_t - (v_\varepsilon)_{xx} = -\varepsilon + u_t - u_{xx} < 0.$

So v_ε achieves its max on $\partial\Omega^+$.

$$\max_{x\in\Omega} u \leq \max_{x\in\Omega} v_\varepsilon + \varepsilon T \leq \max_{x\in\partial^+\Omega} v_\varepsilon + \varepsilon T$$

$$\leq \max_{x\in\partial^+\Omega} u + \varepsilon T.$$ Now let $\varepsilon \to 0$.

Energy

$$E(t) = \frac{1}{2}\int_0^1 |u_x|^2 \, dx$$

HW

$$\frac{d}{dt}E(t) = \int_0^1 u_x \, u_{xt} \, dx$$

$$= \int_0^1 \partial_x (u_x u_t) - u_{xx} u_t \, dx$$

$$= \int_0^1 \partial_x (u_x u_t) - (u_t)^2 \, dx$$

$$= u_x u_t \Big|_0^1 - \int_0^1 (u_t)^2 dx$$

Homogeneous Neumann $\Rightarrow \frac{d}{dt}E(t) \leq 0$

Homogeneous Dirichlet $\Rightarrow \frac{d}{dt}E(t) \leq 0$

Solution becomes "smoother"

If $E(t) = 0$ at some point, $E(t) \equiv 0$.

Exercise: Show that there is at most one solution (C^2, say, in domain).

$$u_t = u_{xx}$$

$$u(0,x) = u_0$$

$$u(t,0) = b_0(t)$$
$$u(t,1) = b_1(t)$$

$$\left[\begin{bmatrix} \lambda & 1 \\ 0 & \lambda \end{bmatrix} t\right]^n = \begin{bmatrix} \lambda^n & n\lambda^{n-1} \\ 0 & \lambda^n \end{bmatrix} t^n$$

$$\sum \frac{\lambda^n t^n}{n!} = e^{\lambda t} \qquad \sum_{n=0}^{\infty} \frac{n \lambda^{n-1} t^n}{n!} = \sum_{n=1}^{\infty} \frac{\lambda^{n-1}}{(n-1)!} t^n$$

$$= t \sum_{n=1}^{\infty} \frac{(t\lambda)^{n-1}}{(n-1)!}$$

$$= t e^{\lambda t}$$

$$\boxed{e^{tA} = \begin{bmatrix} e^{\lambda t} & te^{\lambda t} \\ 0 & e^{\lambda t} \end{bmatrix}}$$

Explicit Method for heat equation

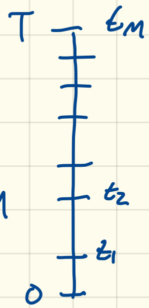
$$u_t = u_{xx} + f(x,t)$$

$u(0,t) = 0$ $\qquad$ $x \longleftrightarrow t$ swap

$u(\ell,t) = 0$ $\qquad$ ℓ instead of 1.

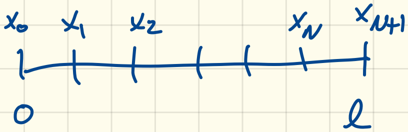
$u(x,0) = g(x)$ $\qquad$ $0 \le x \le \ell$.

Introduce a grid of sample points on our domain



M intervals

$k = T/M$



$M+1$ sample times,

M unknown, one initial

$N+1$ intervals

$N+2$ sample points

N unknowns at $x_1, \dots, x_N$

Introduce approximations for the derivatives.

We've spent a lot of time thinking about discritizing time derivatives. Let's hold off on those. Instead, how about the space derivatives?

$$u_x(x_i) = \frac{u(x_i+h) - u(x_i)}{h} + O(h)$$

$$u_x(x_i) = \frac{u(x_i) - u(x_i-h)}{h} + O(h)$$

$$u_{xx}(x_i) = \frac{u(x_i+h) - 2u(x_i) + u(x_i-h)}{h^2} + ?$$

$$u(x_i+h) = u(x_i) + u_x(x_i)h + \frac{1}{2}u_{xx}(x_i)h^2 + \frac{1}{6}u_{xxx}(x_i)h^3 +$$

$$u(x_i-h) = u(x_i) - u_x(x_i)h + \frac{1}{2}u_{xx}(x_i)h^2 - \frac{1}{6}u_{xxx}(x_i)h^3 +$$

$$u(x_i+h) - 2u(x_i) + u(x_i-h) = u_{xx}(x_i)h^2 + O(h^4) \qquad O(h^4)$$

$$u_{xx}(x_i) = \frac{(\quad\quad)}{h^2} + O(h^2)$$

vanish as $h \to 0$

$$u_t(x_i, t_j) = \frac{u(x_i, t_j + k) - u(x_i, t_j)}{k} + O(k)$$

$$u_{i,j} \approx u(x_i, t_j)$$

$$\frac{u_{i,j+1} - u_{i,j}}{k} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + f(x_i, t_j)$$

Local truncation error: subsitute true solution

$$O(k) + O(h^2)$$

$$f(x_i, t_i) = u_t - u_{xx} \quad \text{at } (x_i, t_i).$$

$$\hookrightarrow = \frac{\quad}{k} + O(k) \qquad \rightarrow = \frac{\quad}{h^2} + O(h^2)$$

It will be helpful to write this as

$$u_{i,j+1} = u_{i,j} + \left(\frac{k}{h^2}\right)\left[u_{i+1,j} - 2u_{i,j} + u_{i,j}\right] + k\,\underbrace{f(x_i, t_j)}_{f_{i,j}}$$

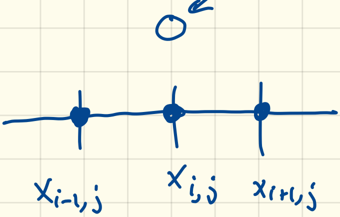
with understanding that $u_{0,j} = 0$, $u_{N+1,j} = 0$

so above holds $1 \le i \le N$

$$0 \le j \le M-1$$

$\frac{k}{h^2} : \lambda$

$x_{i,j+1}$ depends on



This diagram is known as a stencil.

might be 0
$i=1$

might be 0
$i=N$

$$u_{i,j+1} = \lambda u_{i-1,j} + (1-2\lambda) u_{i,j} + \lambda u_{i+1,j} + f_{ij}$$

$$\underbrace{\begin{bmatrix} u_{1,j+1} \\ \vdots \\ \vdots \\ u_{N,j+1} \end{bmatrix}}_{\vec{u}_{j+1}} = \underbrace{\begin{bmatrix} (1-2\lambda) & \lambda & & & \\ \lambda & 1-2\lambda & \lambda & & \\ & \lambda & 1-2\lambda & \lambda & \\ & & & \ddots & \lambda \\ & & & \lambda & (1-2\lambda) \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} u_{1,j} \\ \vdots \\ \vdots \\ u_{N,j} \end{bmatrix}}_{\vec{u}_j} + \underbrace{\begin{bmatrix} f_{1,j} \\ \vdots \\ \vdots \\ f_{N,j} \end{bmatrix}}_{\vec{f}_j}$$

$$\vec{u}_{j+1} = A\vec{u}_j + f_j \qquad \vec{u}_0 = \vec{g} \qquad g_i = u_0(x_i)$$

So this gives us a compact way to express the operations of solving this equation.

You wouldn't want to build A as a full matrix for a big problem though: it's mostly 0's

Ax $\quad O(n^2)$ operations vs

Ax $\quad O(n)$ operations if A is tridiagonal.

Matlab: use sparse matrices

sparse(m,n) ~ zeros(m,n)

b\A will detect A is banded.

Python: need to hold its hand

scipy.linalg.solve_banded

$$(l, u)$$

bands below

above

$$\begin{bmatrix} * & a_{01} & \cdots & \cdots & a_{n-1,n} \\ a_{00} & a_{11} & \cdots & \cdots & a_{nn} \\ a_{10} & a_{21} & \cdots & a_{n,n-1} & * \end{bmatrix} \qquad A = \begin{bmatrix} a_{00} & a_{01} & 0 & \cdots & \\ a_{10} & a_{11} & a_{12} & \cdots & \\ & \ddots & \ddots & \ddots & \\ & & & a_{n,n-1} & a_{nn} \end{bmatrix}$$

$*$ is ignored.

$$(1,1) \qquad \begin{bmatrix} * & \lambda & \cdots & \cdots & \lambda \\ 1-2\lambda & 1-2\lambda & \cdots & \cdots & 1-2\lambda \\ \lambda & \cdots & \cdots & \lambda & * \end{bmatrix}$$

super easy to construct.

Ad

```
scipy.sparse.spdiags(Ad, (1,0,-1), N,N)
```