7/23/18 Lecture Notes: Finite Difference Method for the Heat Equation

- Used to approximate solutions via computer

Calculus Review: Euler's Method

Want to solve: $y' = f(y)$ Pick mesh size Δx.

$y(0) = y_0$ Let $x_n = n\Delta x$

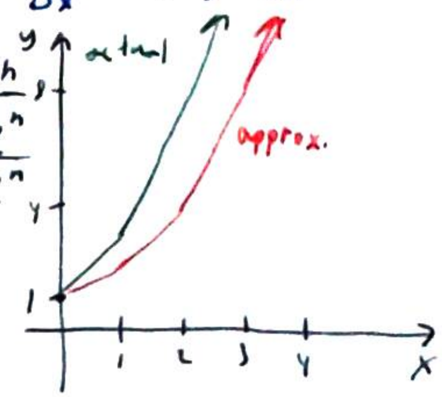
y_n = approx value of $y(x_n)$

Compute y_n by $y_0 = y_0$

$$y_{n+1} = y_n + f(y_n)\Delta x \longleftrightarrow \frac{y_{n+1} - y_n}{\Delta x} = f(y_n) = \text{"}y_n'\text{"}$$

<u>Example</u> $y' = y$, $y(0) = 1$, $\Delta x = 1$

x_n	0	1	2	3	4	n
y_n	1	2	4	8	16	2^n
$f(y_n)$	1	2	4	8	16	2^n



Approx solution 2^{x_n} ~ actual solution e^x

Δx small $\rightarrow$ better approximation, stability

<u>Possible Modifications</u>

Replace asymmetric $\frac{y_{n+1} - y_n}{\Delta x} = y_n'$ with $\frac{y_{n+1} - y_{n-1}}{2\Delta x} = y_n'$ $\rightarrow$ multistep

or $f(y_n)$ with $\frac{f(y_n) + f(y_{n+1})}{2}$ $\rightarrow$ implicit equation

---

Let $u_j = u(j\Delta x)$ $0 \le j \le J$

What is u' at $j\Delta x$? <u>Truncation Error</u> (not roundoff)

Forward difference: $\frac{u_{j+1} - u_j}{\Delta x}$ $O(\Delta x)$ *

Backward difference: $\frac{u_j - u_{j-1}}{\Delta x}$ $O(\Delta x)$

Centered difference: $\frac{u_{j+1} - u_{j-1}}{2\Delta x}$ $O((\Delta x)^2)$

* $f(x) = O(g(x))$ means there exists $C > 0$ such that $f(x) \le C g(x)$ as $x \rightarrow 0$ (or as $x \rightarrow \infty$ in different context)

Finding truncation error: If $u \in C^2$ (Is 2x differentiable with continuous 2nd derivative), then $u(x+h) = u(x) + h\,u'(x) + O(h^2)$, so if $x = j\Delta x$

Forward difference approximation

$$\frac{u_{j+1} - u_j}{\Delta x} = \frac{\cancel{u(x)} + \Delta x\, u'(x) + O(\Delta x^2) - \cancel{u(x)}}{\Delta x}$$

$$= u'(x) + O(\Delta x)$$

- Computations for backward, centered similar

What about 2nd derivatives?

We'll use the "centered second difference" $u''(j\Delta x) \approx \dfrac{u_{j+1} - 2u_j + u_{j-1}}{(\Delta x)^2}$

truncation error $O(\Delta x^2)$

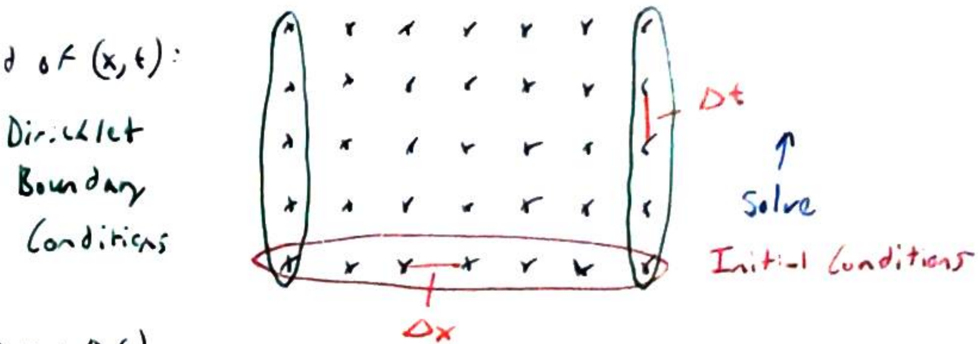
<u>Solving</u> <u>PDEs</u>

Example: $u_t = u_{xx}$ $u(x,0) = \phi(x)$

We can already solve $u_t = u_{xx}$, so why care?

1) Helps develop model for unsolved (and unsolvable) cases.
2) Real-world data is discrete, not continuous.

Need 2D-grid of (x,t):



Let $u_j^n = u(j\Delta x, n\Delta t)$

u_t - forward difference
u_{xx} - centered 2nd difference

$$\left.\begin{array}{l} u_t \text{ - forward difference} \\ u_{xx} \text{ - centered 2nd difference} \end{array}\right\} \longrightarrow \frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2} \Bigg\} \text{ difference equation}$$

If $\Delta t = \Delta x = 1$,

$u_j^{n+1} = u_{j+1}^n - u_j^n + u_{j-1}^n$ ← scheme (explicit)

$*$ ← template

•+1 •−1 •+1

If $\Delta t = \Delta x = 10$,

$$\frac{u_j^{n+1} - u_j^n}{10} = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{100}$$

$$u_j^{n+1} = \frac{u_{j+1}^n}{10} + \frac{8}{10}u_j^n + \frac{u_{j-1}^n}{10}$$

$*$

•$1/10$ •$8/10$ •$1/10$

As a group (but with a grid on each person's page):

a) Take $\Delta t = \Delta x = 10$ and solve

0	.1	2.4	19.5	56	19.5	2.4	.1	0
0	0	1	16	66	16	1	0	0
0	0	0	10	80	10	0	0	0
0	0	0	0	100	0	0	0	0

— Break

b) Take $\Delta t = \Delta x = 1$ and solve

0	1	−3	6	−7	6	−3	1	0
0	0	1	−2	3	−2	1	0	0
0	0	0	1	−1	1	0	0	0
0	0	0	0	1	0	0	0	0

Unstable!
- violates maximum principle

Stability

Consider $u_t = u_{xx}$ on $0 < x < \pi$, $t > 0$ Let $s = \frac{\Delta t}{(\Delta x)^2}$

$u(0,t) = u(\pi,t) = 0$ $t > 0$

$u(x,0) = \phi(x)$

If $1 \le j \le J-1$, use scheme

$j = 0$ or $j = J$, take $u_j^n = 0$

Rewrite scheme: $$\frac{u_j^{n+1}-u_j^n}{\Delta t} = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2} \quad \cdot \Delta t$$

$$u_j^{n+1} - u_j^n = s\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right)$$

$$u_j^{n+1} = s\left(u_{j+1}^n + u_{j-1}^n\right) + \underline{(1-2s)}u_j^n$$

naively speaking, want $1-2s \geq 0 \to$ nice averaging

Problem: "Solutions" blow up when they should decay

Decay seen in $\sum_{n=1}^{\infty} e^{-n^2 t} \sin nx$, since each separated solution decays

Try discrete separation of variables, check decay

Let $u_j^n = X_j T_n \to X_j T_{n+1} = T_n\left[s\left(X_{j+1} + X_{j-1}\right) + (1-2s)X_j\right]$

$$\frac{T_{n+1}}{T_n} = 1 - 2s + s\,\frac{X_{j+1} + X_{j-1}}{X_j} = \xi \quad \text{constant}$$

$T_n = \xi^n T_0$, need $|\xi| \leq 1$ for stability

Solve $1 - 2s + s\,\dfrac{X_{j+1} - X_{j-1}}{X_j} = \xi$ with BC $X_0 = X_J = 0$

Guess* solutions $X_j = \sin(k \cdot j\Delta x) \quad 1 \leq k$

* Can justify with discrete Fourier transform / Fourier series on finite groups

$$s\,\frac{\sin((j+1)k\Delta x) + \sin((j-1)k\Delta x)}{\sin(jk\Delta x)} + 1 - 2s = \xi$$

sine addition formula

$$s\,\frac{2\sin(jk\Delta x)\cos k\Delta x}{\sin jk\Delta x} + 1 - 2s = \xi$$

$$\xi = \xi(k) = 1 - 2s(1 - \cos k\Delta x)$$

$-1 \leq \cos k\Delta x \leq 1$, so

$1 - 4s \leq \xi \leq 1$, stability guaranteed if $\boxed{s \leq 1/2}$

Note: "Solution" is $u_n^j = \sum_k b_k \sin(jk\Delta x)\left[\xi(k)\right]^n$

Neumann Boundary Conditions

Solve $u_t = u_{xx} \quad 0 < x < \pi$

$$u_x(0,t) = g(t) \longrightarrow g^n = g(n\Delta t)$$
$$u_x(\pi,t) = h(t) \longrightarrow h^n = h(n\Delta t)$$

Naive approach: Determine boundary points u_0^n, u_J^n by

$$\frac{u_1^n - u_0^n}{\Delta x} = g^n \qquad \frac{u_J^n - u_{J-1}^n}{\Delta x} = h^n \leftarrow \text{BAD}$$

but this leads to $O(\Delta x)$ error. Instead, introduce u_{-1}^n, u_{J+1}^n, so

$$\frac{u_1^n - u_{-1}^n}{2\Delta x} = g^n \; (*) \qquad \frac{u_{J+1}^n - u_{J-1}^n}{2\Delta x} = h^n \; (**) \leftarrow \text{GOOD}$$

This has $O(\Delta x^2)$ error

Picture