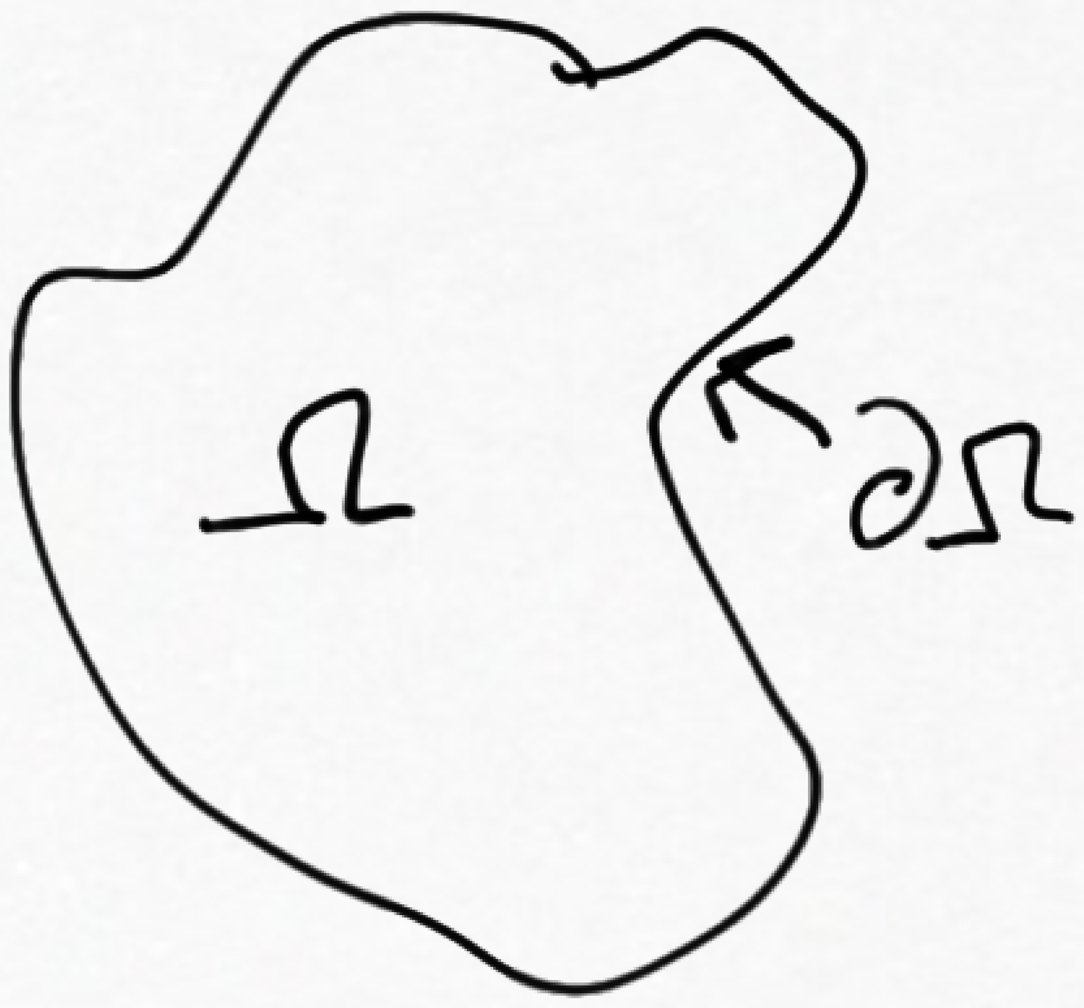
The Heat equation in 2D

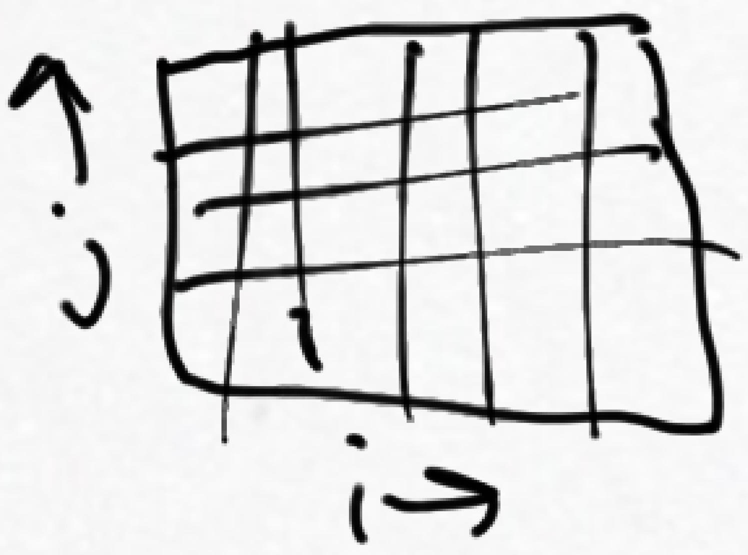
$u_t = u_{xx} + u_{yy}$ $\qquad u = u(x,y,t)$



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

$u(x,y,t) = g(x,y) \quad (x,y) \in \partial\Omega$

5-pt. Laplacian: $\partial_{xx} = \dfrac{U_{i+1,j} - 2U_{ij} + U_{i-1,j}}{(\Delta x)^2} = D_x^2 U$

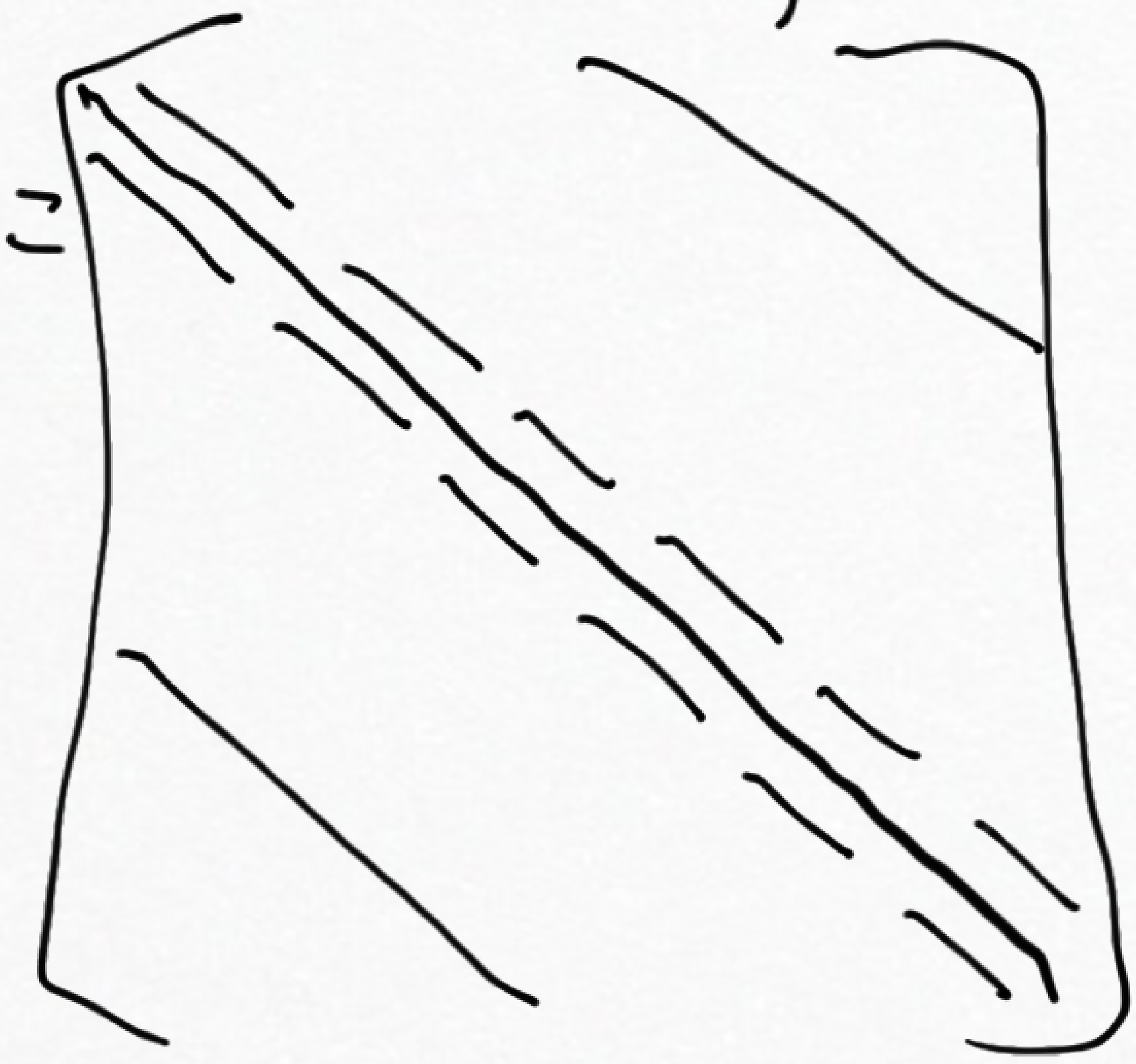


$\partial_{yy} = \dfrac{U_{i,j+1} - 2U_{ij} + U_{i,j-1}}{(\Delta y)^2} = D_y^2 U$

$\Delta x = \Delta y = h$

Semi-discretization: $U_{ij}'(t) = (D_x^2 U)_{ij} + (D_y^2 U)_{ij} = \nabla_h^2 U_{ij}$

$D_x^2 + D_y^2 = \nabla_h^2 =$



(sparsity pattern)

Stiff problem

Use implicit time discretization

Sparse banded matrix

How stiff is it?

Largest eigenvalues: $\approx -\frac{8}{h^2} = \mathcal{O}\left(\frac{1}{h^2}\right)$

Smallest eigenvalues: $\approx -2\pi^2 = \mathcal{O}(1)$

- Apply Trapezoidal method:

$$U^{n+1} = U^n + \frac{K}{2}\left[\nabla_h^2 U^n + \nabla_h^2 U^{n+1}\right]$$

$$\underbrace{\left[I - \frac{K}{2}\nabla_h^2\right]}_{M} U^{n+1} = \left[I + \frac{K}{2}\nabla_h^2\right] U^n$$

solve this linear system at each step.

assume $K \approx h$

Eigenvalues of M: largest are $\mathcal{O}\left(\frac{K}{h^2}\right) \approx \mathcal{O}\left(\frac{1}{h}\right)$

Better-conditioned than the BVP

Iterative methods converge quickly

Locally 1D method
(dimensional splitting)

$$\underline{U'(t) = (\partial_x^2 + \partial_y^2)U(t)} \qquad U(0) = u_0$$

Solution: $e^{t(\partial_x^2+\partial_y^2)}u_0$ $\quad$ $U(t+k) = e^{k(\partial_x^2+\partial_y^2)}U(t)$

$$U(t+k) = U(t) + k(\partial_x^2+\partial_y^2)U(t) + \frac{k^2}{2}(\partial_x^2+\partial_y^2)^2U(t) + \cdots$$

Instead: $\boxed{\begin{aligned} U'(t) &= \partial_x^2 U \\ U'(t) &= \partial_y^2 U \end{aligned}}$ $\Big\}$ Solve these sequentially one after the other

$$U(t+k) = e^{k\partial_y^2}e^{k\partial_x^2}U(t) = \left(I + k\partial_y^2 + \frac{k^2}{2}\partial_y^4 + \cdots\right)\left(I + k\partial_x^2 + \frac{k^2}{2}\partial_x^4 + \cdots\right)U(t)$$

$$= \left(I + k\partial_x^2 + k\partial_y^2 + k^2\partial_y^2\partial_x^2 + \frac{k^2}{2}(\partial_y^4 + \partial_x^4) + \mathcal{O}(k^3)\right)U(t)$$

Do the $\mathcal{O}(k^2)$ terms match?

$$\frac{k^2}{2}(\partial_x^2+\partial_y^2)^2 = \frac{k^2}{2}\left(\partial_x^2\partial_y^2 + \partial_y^2\partial_x^2 + \partial_x^4 + \partial_y^4\right)$$

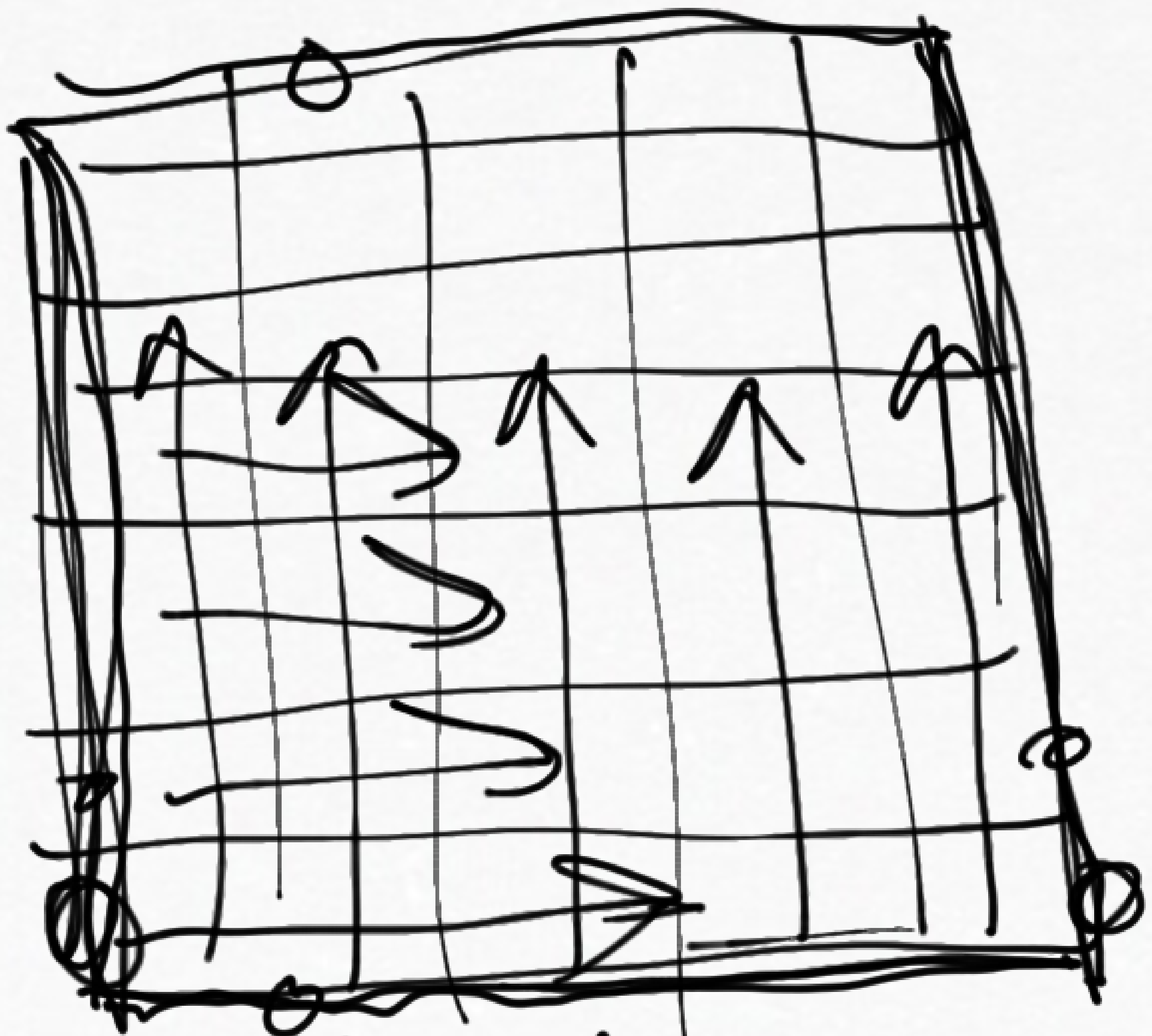
Only if $\partial_x^2\partial_y^2 = \partial_y^2\partial_x^2$. (They do)

This approach is much cheaper because we only solve 1D problems.

If grid $m \times m$

$M: m^2 \times m^2$

But for LOD method, we only use a $m \times m$ tridiagonal matrix.



Algorithm

1. Start with $U^n \approx u(t_n)$
2. Diffuse in x to get U^*
3. Diffuse in y to get U^{n+1}

   $U^{n+1} \approx u(t_{n+1})$

Boundary conditions

What BCs to use for U^*?

Imposed prescribed BCs along bottom + top rows, then solve diffusion equation backward over one step:

$$U_t = -U_{xx}$$

$\boxed{\text{Alternating Direction Implicit (ADI)}}$

$$U^* = U^n + \frac{k}{2}\left(D_y^2 U^n + D_x^2 U^*\right)$$ Implicit only in x

$$U^{n+1} = U^* + \frac{k}{2}\left(D_x^2 U^n + D_y^2 U^{n+1}\right)$$ Implicit only in y

---

$$U^* = \left(I - \frac{k}{2}D_x^2\right)^{-1}\left(I + \frac{k}{2}D_y^2\right)U^n$$

$$U^{n+1} = \left(I - \frac{k}{2}D_y^2\right)^{-1}\left(I + \frac{k}{2}D_x^2\right)U^*$$

$$U^{n+1} = \left(I - \frac{k}{2}D_y^2\right)^{-1}\left(I + \frac{k}{2}D_x^2\right)\left(I - \frac{k}{2}D_x^2\right)^{-1}\left(I + \frac{k}{2}D_y^2\right)U^n$$

$$U^{n+1} = \left(I + \frac{k}{2}D_y^2 + \frac{k^2}{4}D_y^4 + \cdots\right)\left(I + \frac{k}{2}D_x^2\right)\left(I + \frac{k}{2}D_x^2 + \frac{k^2}{4}D_x^4 + \cdots\right)\left(I + \frac{k}{2}D_y^2\right)U^n$$

$$U^{n+1} = \Big[I + \frac{k}{2}\left(2D_y^2 + 2D_x^2\right) + \frac{k^2}{4}\left(2D_y^4 + 2D_x^4 + D_y^2D_x^2 + D_y^2D_x^2 + D_x^2D_y^2 + D_x^2D_y^2\right) + O(k^3)\Big]U^n$$

$$U^{n+1} = \underline{\Big[I + k\left(D_x^2 + D_y^2\right) + \frac{k^2}{2}\left(D_x^4 + D_y^4 + D_x^2D_y^2 + D_y^2D_x^2\right) + O(k^3)\Big]}\,U^n$$

Correct up to $O(k^2)$. (Even if D_x^2, D_y^2 don't commute)