The Heat Equation in 2D

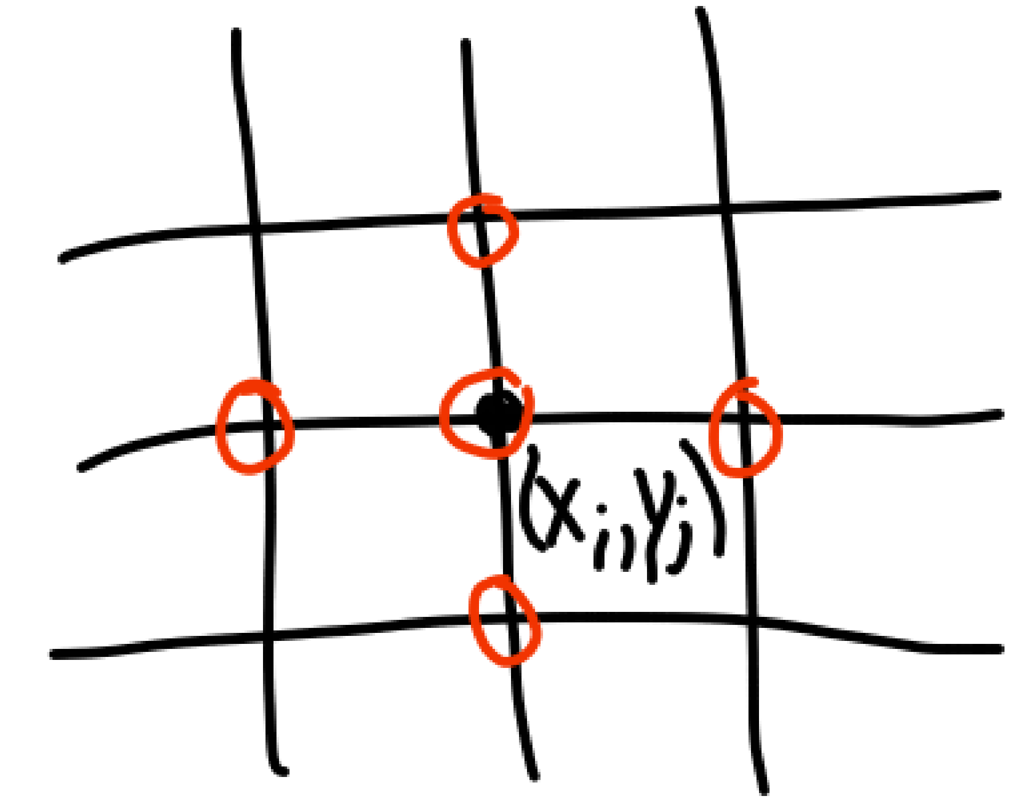
$U_{ij}(t) \approx u(x_i, y_j, t)$

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

$U_{ij}^n \approx u(x_i, y_j, t_n)$

$$u(x,y,t=0) = \eta(x,y) \qquad (x,y) \in \Omega = [0,1] \times [0,1]$$

$$u(x,y,t) = g(t) \text{ for } (x,y) \in \partial\Omega$$



Discretize in space:

$x_i = ih \qquad i,j = 0, 1, \ldots, m+1$

$y_j = jh \qquad h = \frac{1}{m+1}$

$$u_{xx}(x_i, y_j, t) \approx \frac{U_{i+1,j} - 2U_{ij} + U_{i-1,j}}{h^2} = (D_x^2 U)_{ij}$$

$$u_{yy}(x_i, y_j, t) \approx \frac{U_{i,j+1} - 2U_{ij} + U_{i,j-1}}{h^2} = (D_y^2 U)_{ij}$$

We must choose an ordering so we can write all the $U_{ij}(t)$ as a vector $U(t)$.

We have

$$U'(t) = D_x^2 U + D_y^2 U$$

$$= \nabla_h^2 U$$

$$\nabla_h^2 \in \mathbb{R}^{m^2 \times m^2}$$

5-point Laplacian

This is a system of m^2 ODEs.

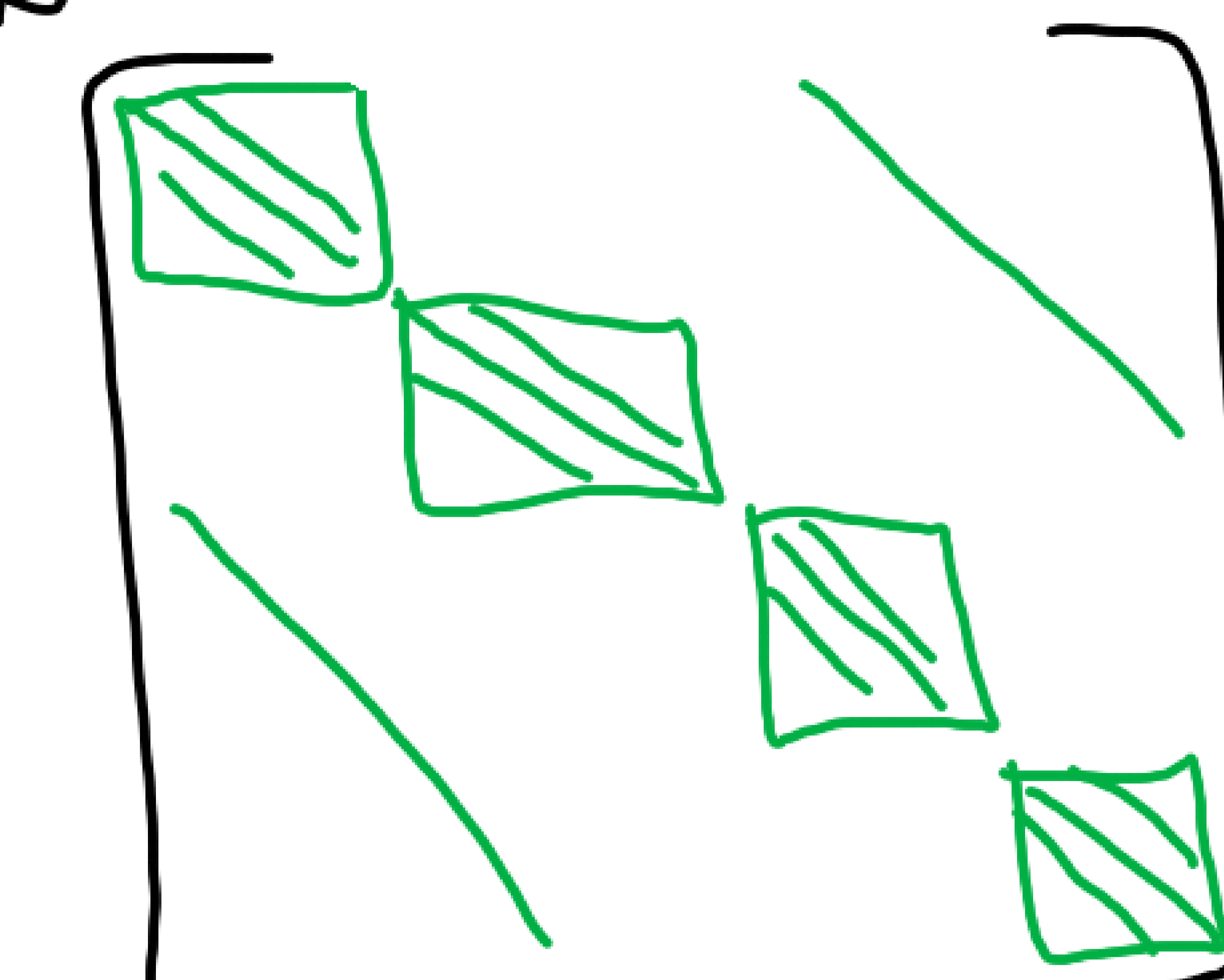
Implicit trapezoidal method in time: (Crank-Nicolson)

$$U^{n+1} = U^n + \frac{k}{2}\left(\nabla_h^2 U^n + \nabla_h^2 U^{n+1}\right)$$

$$\underbrace{\left(I - \tfrac{k}{2}\nabla_h^2\right)}_{A}\underbrace{U^{n+1}}_{x} = \underbrace{\left(I + \tfrac{k}{2}\nabla_h^2\right)U^n}_{b}$$

A is: sparse

With row-wise ordering:



We have to solve a $m^2 \times m^2$ sparse system at each step ($\sim \frac{1}{k}$ times!)

$$A = I - \frac{k}{2}\nabla_h^2$$

Should we use:

- A direct solver (e.g. Gaussian elimination)
- An iterative solver?

We could compute an LU factorization of A and reuse it at each step

cost: $\mathcal{O}(m^4)$

The efficiency of an iterative solver depends on:

- Initial guess
- Condition # of A ($\kappa(A)$)

$$\kappa(A) = \frac{\max|\lambda|}{\min|\lambda|} \quad \lambda \in \sigma(A)$$

Assume $k \propto h$

Eigenvalues of A:

$$\lambda_{p,q} = 1 - \frac{k}{h^2}\left(\cos(p\pi h) + \cos(q\pi h) - 2\right)$$

$p,q = 1, 2, \dots, m$

$$\max_{p,q}|\lambda_{p,q}| \approx \frac{4k}{h^2} \qquad \min_{p,q}|\lambda_{p,q}| \approx 1$$

$\cos(p\pi h) \approx 1 - p^2\pi^2h^2$

$\cos(q\pi h) \approx 1 - q^2\pi^2h^2$

Take $p = q = 1$:

$$\lambda_{p,q} = 1 - \frac{K}{h^2}\left(-2\pi^2h^2\right)$$

$$= 1 + 2\pi^2 K$$

So $K(A) \approx 4\frac{K}{h^2} = \mathcal{O}(1/h)$ as $K, h \to 0$

Recall that $K(\nabla_h^2) = (1/h^2)$.

We have a very good initial guess: U^n

Often iterative methods will converge in 1-2 iterations.

This can be cheaper than using a direct solver!

Dimensional Splitting

Instead of solving

$$u_t = u_{xx} + u_{yy}$$

we solve

$$u_t = u_{xx} \quad (1)$$

$$u_t = u_{yy} \quad (2)$$

in alternating order.

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

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

Solve (1) for U^*, then solve (2) for U^{n+1}.

Cost comparison:

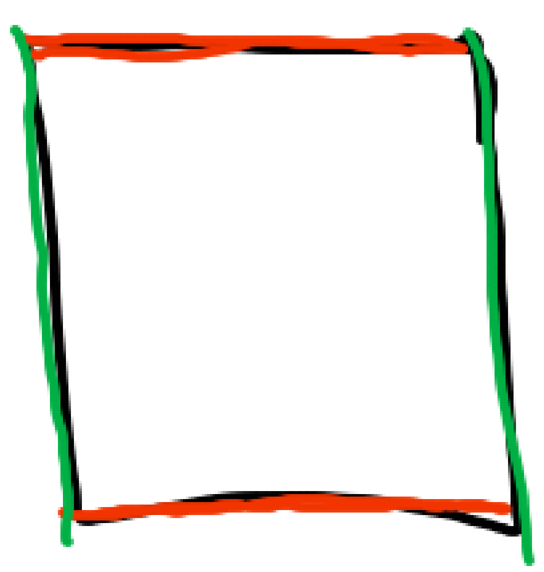
Unsplit method: Solve one sparse system of size m^2 ($\mathcal{O}(m^4)$ work)

Split method: Solve $2m$ tridiagonal systems of size m

Work: $\mathcal{O}(m^2)$

Boundary conditions for U^*:

In (2): We need BCs at top and bottom. Take the BCs at $t=t_n$ and diffuse in the x-direction. (solve $u_t = u_{xx}$)



In (1): we need BCs at left and right edges. Take the boundary values at t_{n+1} and solve

$$u_t = -u_{yy}$$

over one step.

Alternating Direction Implicit (ADI)

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

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

Here $U^* \approx u(x, y, t_n + \frac{k}{2})$.

So we can use $g(t_n + \frac{k}{2})$ for the BCs for U^*.