§2. Parabolic PDEs.

I. Difference methods, Σ_T^n, & order of accuracy

$\boxed{u_t = k u_{xx}}$; the classical example of the parabolic PDEs

--- ①

→ We may assume $k = 1 > 0$ for (mathematical) simplicity.

→ It should be noted that the diffusion coefficient k needs to be positive, $k > 0$, in order to have the well-posedness property.

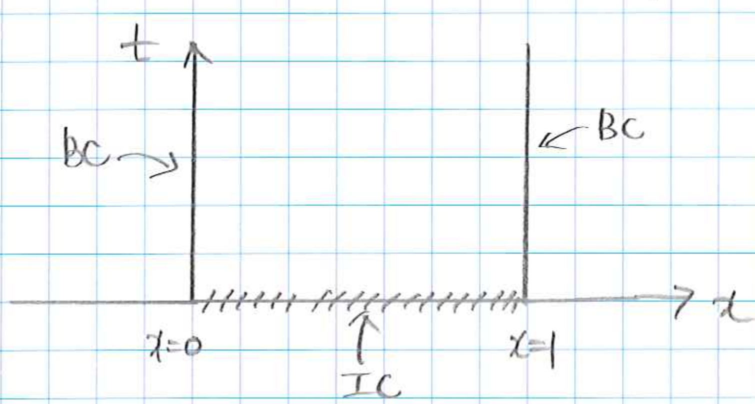
→ If $k < 0$, then the parabolic PDE becomes ill-posed and we do not consider the case.

→ We also need two more side conditions, in addition to the PDE:

(1) IC : $u(x,0) = u^0(x)$, $t^0 = 0$

(2) BC : $\begin{cases} u(0,t) = g_0(t) \\ u(1,t) = g_1(t) \end{cases}$, $\forall t > 0$.

over a bounded domain, e.g., $0 \le x \le 1$.



→ A special case when $\frac{\partial}{\partial t} = 0$:

this becomes a steady-state (or stationary-state) eqn. and it becomes an elliptic PDE.

Therefore, one can consider elliptic PDEs as the limiting case of steady-states from either parabolic or hyperbolic PDEs.

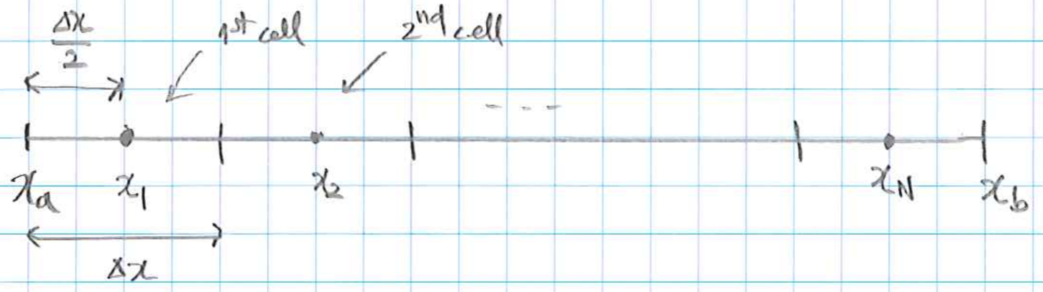
→ Discretize both spatial and temporal grids (x_i, t^n) :

$$\begin{cases} x_i = (i-\frac{1}{2})\Delta x & \text{: cell-centered, } i=1,2,\cdots,N \\ x_i = i\Delta x & \text{: cell-interface, } i=0,1,\cdots,N \\ t^n = n\Delta t, & n=0,1,\cdots M. \end{cases}$$

→ Note that we are going to use the "cell-centered" spatial discretization in our course.

→ If the domain begins, in general, not from $x=0$, but is given as $[x_a, x_b]$, then

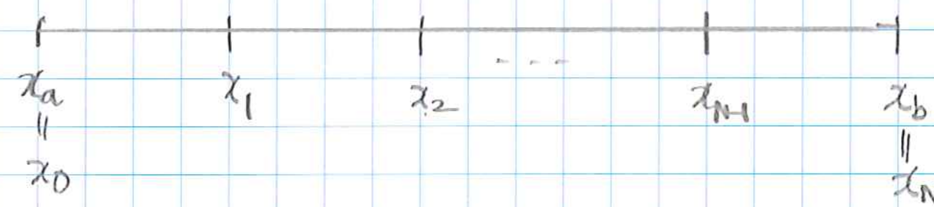
$$x_i = x_a + (i-\tfrac{1}{2})\Delta x, \quad i=1,2,\cdots,N$$



→ In this cell-centered configuration, we obtain "N" discrete point, with $\Delta x = \frac{x_b - x_a}{N}$,
(at which we seek for U_i^n)

in which BCs are to be imposed $\Delta x/2$ distance away from x_1 & x_N, rather than x_1 & x_N directly. (need some care for BCs)

→ Compared to the cell-centered configuration, in the cell-interface configuration we obtain "N+1" discrete points with $\Delta x = \frac{x_b - x_a}{N}$.



In this case, at x_0 & x_N, we do NOT solve for the numerical appns U_0^n & U_N^n, but they should be BCs.

We only seek for numerical solns U_i^n, $\overbrace{i=1,\cdots,N-1}^{(N-1)\text{ pts}}$, using finite difference approximations, using BCs at U_0^n & U_N^n.

→ Let $U_i^n \approx u(x_i, t^n)$ represent the numerical appn at grid pts (x_i, t^n).

Ex. We solve the heat eqn ① using finite difference methods (FDM). For examples,

(i) Forward in time (i.e., Forward Euler)

$$\boxed{\frac{\partial u}{\partial t} \simeq \frac{U_i^{n+1} - U_i^n}{\Delta t}}$$

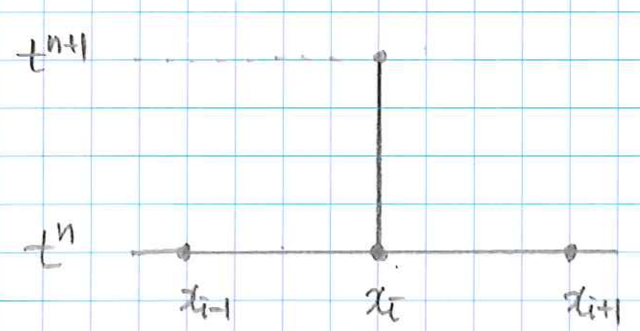
(ii) Centered in space ;

$$\boxed{\frac{\partial^2 u}{\partial x^2} \simeq \frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{\Delta x^2}}$$

⇒ FTCS (Forward in time Centered in Space)

$$\frac{U_i^{n+1} - U_i^n}{\Delta t} = k \, \frac{U_{i+1}^n - 2U_i^n + U_{i-1}^n}{\Delta x^2}, \quad \text{or}$$

$$\boxed{U_i^{n+1} = U_i^n + \frac{k\Delta t}{\Delta x^2}\left(U_{i+1}^n - 2U_i^n + U_{i-1}^n\right)}$$



⇐ stencil

⇒ FTCS :
- ① explicit (i.e., FE temporal discretization)
- ② 1st order in time , $O(\Delta t)$
- ③ 2nd order in space , $O(\Delta x^2)$

Ex. Crank-Nicolson method.

→ Define the 2^{nd} derivative operator $D^2 \approx \frac{\partial^2}{\partial x^2}$.

→ The method is defined by

$$\frac{U_i^{n+1} - U_i^n}{\Delta t} = k\left[D^2 U_i^n + D^2 U_i^{n+1}\right]/2$$

$$= \frac{k}{2\Delta x^2}\left[\left(U_{i+1}^n - 2U_i^n + U_{i-1}^n\right) + \left(U_{i+1}^{n+1} - 2U_i^{n+1} + U_{i-1}^{n+1}\right)\right],$$

or, letting $r = k\frac{\Delta t}{2\Delta x^2}$,

$$\boxed{-rU_{i-1}^{n+1} + (1+2r)U_i^{n+1} - rU_{i+1}^{n+1} = rU_{i-1}^n + (1-2r)U_i^n + rU_{i+1}^n}$$

→ ① implicit (in time)
② 2^{nd} order in time, $O(\Delta t^2)$
③ 2^{nd} order in space, $O(\Delta x^2)$.

→ This yields a tridiagonal system of eqns in matrix form:

$$\begin{bmatrix} (1+2r) & -r & & & 0 \\ -r & (1+2r) & -r & & \\ & \ddots & \ddots & \ddots & \\ & & -r & (1+2r) & -r \\ 0 & & & -r & (1+2r) \end{bmatrix} \begin{bmatrix} U_1^{n+1} \\ U_2^{n+1} \\ \vdots \\ U_{N-1}^{n+1} \\ U_N^{n+1} \end{bmatrix} = \begin{bmatrix} BC_1 + (1-2r)U_1^n + rU_2^n \\ rU_1^n + (1-2r)U_2^n + rU_3^n \\ \vdots \\ rU_{N-2}^n + (1-2r)U_{N-1}^n + rU_N^n \\ rU_{N-1}^n + (1-2r)U_N^n + BC_2 \end{bmatrix}$$

$$\left.\begin{aligned} BC_1 &= g_0(t^n) + g_1(t^{n+1}) \\ BC_2 &= g_1(t^n) + g_1(t^{n+1}) \end{aligned}\right\} \text{BC's using } \left\{\begin{aligned} u(0,t) &= g_0(t), \\ u(1,t) &= g_1(t). \end{aligned}\right.$$

5/2/12

→ Note that the use of the implicit method allows larger Δt in Crank-Nicolson method than in FTCS.

→ We also note that the matrix structure in Crank-Nicolson is similar to the one in the Two-point BVPs.

Ex E_T^n & order of accuracy for FTCS $(K=1)$

$$E_T^{n+1} = \frac{1}{\Delta t}\left[u(x_i, t^n+\Delta t) - u(x_i, t^n)\right]$$

$$- \frac{1}{\Delta x^2}\left[u(x_i-\Delta x, t^n) - 2u(x_i,t^n) + u(x_i+\Delta x, t^n)\right]$$

$$= \frac{1}{\Delta t}\Big[\cancel{u(x_i,t^n)} + \Delta t\, u_t(x_i,t^n) + \frac{\Delta t^2}{2} u_{tt}(x_i,t^n) + O(\Delta t^3)$$

$$- \cancel{u(x_i,t^n)}\Big]$$

$$- \frac{1}{\Delta x^2}\Big[\Big(\cancel{u(x_i,t^n)} - \cancel{\Delta x\, u_x(x_i,t^n)} + \frac{\Delta x^2}{2} u_{xx}(x_i,t^n) + - \cancel{\frac{\Delta x^3}{6} u_{xxx}(x_i,t^n)} + \frac{\Delta x^4}{24} u_{xxxx}(x_i,t^n) + \cdots\Big)$$

$$- \cancel{2u(x_i,t^n)}$$

$$+ \Big(\cancel{u(x_i,t^n)} + \cancel{\Delta x\, u_x(x_i,t^n)} + \frac{\Delta x^2}{2} u_{xx}(x_i,t^n) + + \cancel{\frac{\Delta x^3}{6} u_{xxx}(x_i,t^n)} + \frac{\Delta x^4}{24} u_{xxxx}(x_i,t^n) + \cdots\Big)\Big]$$

0 ←

$$= u_t(x_i,t^n) + \frac{\Delta t}{2} u_{tt}(x_i,t^n) + O(\Delta t^2)$$

$u_t = u_{xx}$

$u_{tt} = (u_t)_{xx}$

$= u_{xxxx}$

$$- u_{xx}(x_i,t^n) - \frac{\Delta x^2}{12} u_{xxxx}(x_i,t^n) + O(\Delta x^4)$$

$$= \left(\frac{\Delta t}{2} - \frac{\Delta x^2}{12}\right) u_{xxxx}(x_i,t^n) + O(\Delta t^2, \Delta x^4)$$

$\Rightarrow E_T^{n+1} = O(\Delta t + \Delta x^2)$

(!) FTCS is 1^{st} order in time &

2^{nd} order in space.

Ex. E_T^n & order of accuracy for Crank-Nicolson.

$\rightarrow$ HW.

2 Method of Lines (MOL)

→ The idea is to discretize the PDE in space alone first, which will result in ODEs

→ This will convert the PDE into an ODE

$$\boxed{u_i'(t) = R(u_i)}, \quad i=1,\cdots,N, \quad --- \quad (2)$$

where $R(u_i)$ is the spatial discretization of the PDE.

→ (2) can be solved using the numerical methods for ODEs for each i.

→ This system of ODEs (note there are N ODEs, $i=1,\cdots,N$) in (2) is

(i) "exact" in time, and
(ii) "approximate" in space,

and is called the "semi-discrete" method.

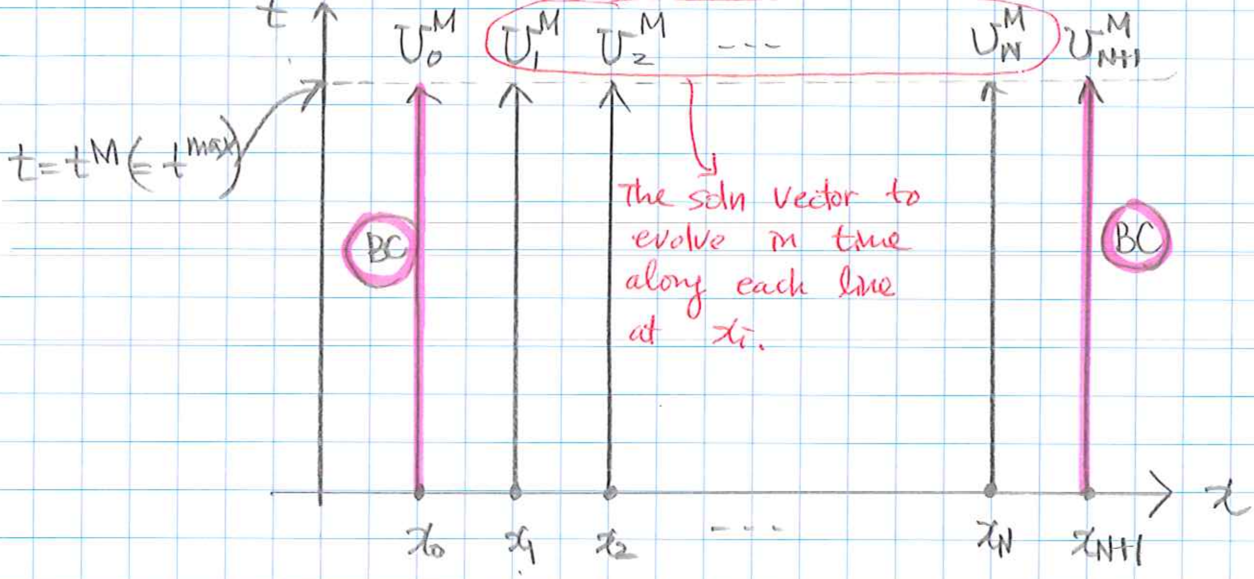
Ex) $u_t = k u_{xx}$ exact!

$$\Rightarrow \frac{dU_i(t)}{dt} = \frac{k}{\Delta x^2}\left(U_{i+1}(t) - 2U_i(t) + U_{i-1}(t)\right), \quad i=1,\cdots,N$$

⇒ often, MOL is useful, since we can write

$$\frac{dU_i(t)}{dt} = R(U_i(t)), \quad i=1,\cdots,N, \quad --- \quad (3)$$

and then we use a proper ODE solver to solve the system given by (3).



→ MOL interpretation is that each $U_i(t)$ is the soln. along the line forward in time at each grid pt. x_i.

→ In a matrix-vector notation, the entire system can be written as

$$\frac{d\underset{\sim}{U}(t)}{dt} = \underset{\sim}{A}\,\underset{\sim}{U}(t) + \underset{\sim}{g}(t), \quad \text{where}$$

$$\underset{\sim}{A} = \frac{k}{\Delta x^2}\begin{bmatrix} -2 & 1 & & & & 0 \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ 0 & & & 1 & -2 & 1 \\ & & & & 1 & -2 \end{bmatrix}_{N\times N}, \quad \underset{\sim}{U}(t) = \begin{bmatrix} U_1(t) \\ U_2(t) \\ \vdots \\ U_N \end{bmatrix}_{N\times 1}$$

$$\underset{\sim}{g}(t) = \frac{k}{\Delta x^2}\begin{bmatrix} g_0(t) \\ 0 \\ \vdots \\ 0 \\ g_1(t) \end{bmatrix}_{N\times 1}, \quad \text{where BCs: } \begin{cases} U_0(t) = g_0(t), \\ U_{N+1}(t) = g_1(t) \end{cases}$$