19/7/16

The Heat Equation

$$\frac{\partial u}{\partial t} \pm \nabla^2 u = 0 \qquad \text{Later}$$

Preliminary

Taylor's theorem

$$f(x+\Delta x) = f(x) + \Delta x\, f'(x) + \frac{(\Delta x)^2}{2!} f''(x) + \frac{(\Delta x)^3}{3!} f'''(x) + \ldots$$

$$f(x-\Delta x) = f(x) - \Delta x\, f'(x) + \frac{(\Delta x)^2}{2!} f''(x) - \frac{(\Delta x)^3}{3!} f'''(x) + \ldots$$

$$f'(x) = \frac{f(x+\Delta x) - f(x)}{\Delta x} + \frac{\Delta x}{2!} f''(x) + \text{HOT} \qquad \text{FD}$$

$$\text{or} \quad \frac{f(x) - f(x-\Delta x)}{\Delta x} + O(\Delta x) \qquad \text{BD}$$

$$\frac{f(x+\Delta x) - f(x-\Delta x)}{2\Delta x} + O((\Delta x)^2) \quad - \text{C}$$

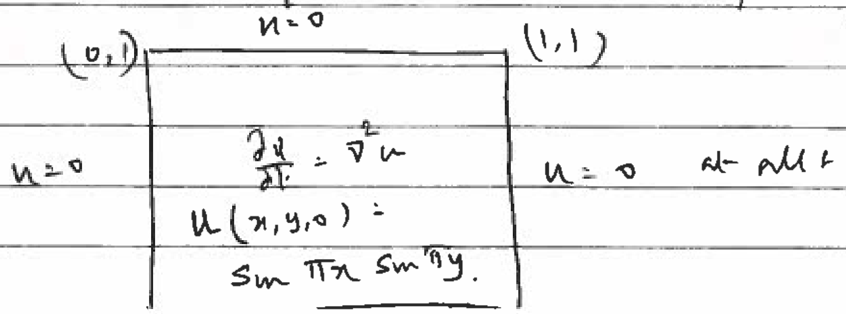
$$f''(x) = \frac{f(x+\Delta x) - 2f(x) + f(x-\Delta x)}{(\Delta x)^2} + O((\Delta x)^2) \quad -$$

Heat Equation

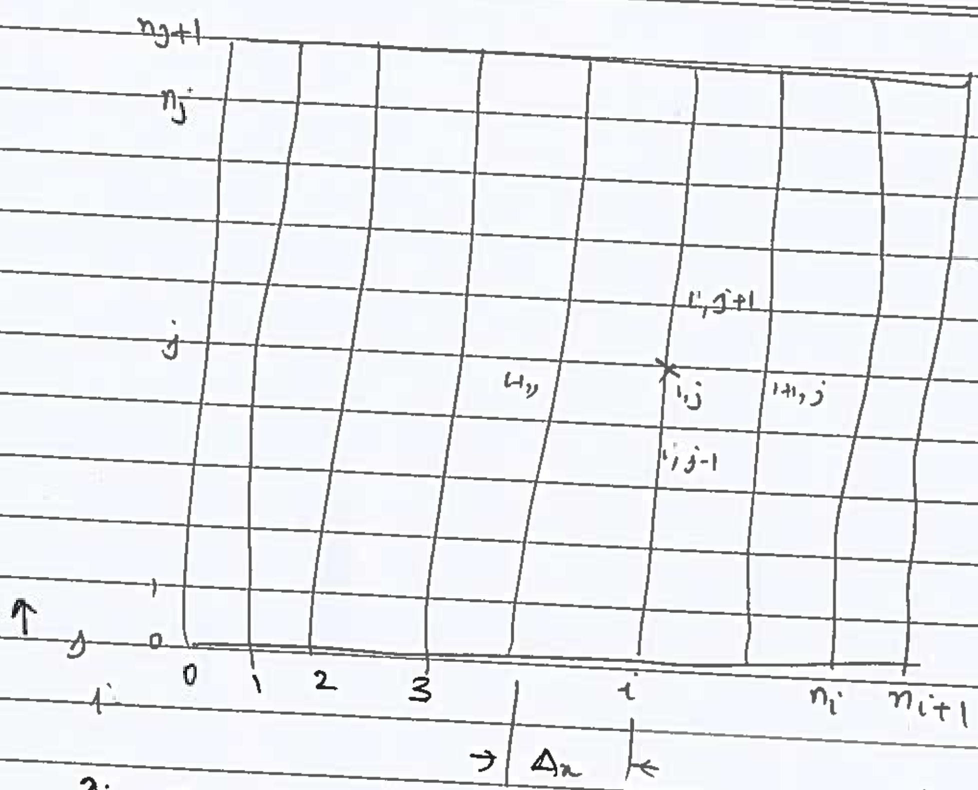
Non dimensionalised

$$\frac{\partial u}{\partial t} = \nabla^2 u \qquad \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

On a unit square with the following BCs



Next step – Discretise



x: 0 ; Δ

$$\Delta x = \frac{1}{n_i+1} \; ; \quad \Delta y = \frac{1}{n_j+1}.$$

$$x_i = i\,\Delta x \qquad y_j = j\,\Delta y \qquad t = n\Delta t$$

$$u(x_i, y_j, t_n) = u_{i,j}^{n}$$

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2}$$

$$\left.\frac{\partial^2 u}{\partial y^2}\right|_{i,j} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{(\Delta y)^2}$$

$$\frac{\partial u}{\partial t} = \frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t}$$

$$\frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} = \frac{u_{i+1,j} - 2u_{i,j} - u_{i-1,j}}{(\Delta x)^2} + \frac{u_{i,j+1} - 2u_{i,}}{(\Delta y)^2}$$

P.S. SWATHI

Question :- At what time do you evaluate the RHS

If we do at n, — Explicit.

at $n+1$ — Implicit.

Explicit

Let $\alpha = \frac{\Delta t}{(\Delta x)^2}$; $\beta = \frac{\Delta t}{(\Delta y)^2}$

$$u_{i,j}^{n+1} = \alpha \left[u_{i-1,j}^{n} + u_{i+1,j}^{n} \right] + \beta \left[u_{i,j-1}^{n} + u_{i,j+1}^{n} \right]$$

$$+ \left(1 - 2\alpha - 2\beta\right) u_{i,j}^{n}$$

Now $0 < 1 - 2\alpha - 2\beta < 1$ Stability condition

< 0 — Flip - flop.

> 1 $\rightarrow$ exponentially grow.

$$\Rightarrow 2\Delta t \left[\frac{1}{(\Delta x)^2} + \frac{1}{(\Delta y)^2} \right] < 1$$

$$\text{or} \quad \Delta t < \frac{1}{2} \frac{(\Delta x)^2 (\Delta y)^2}{\Delta x^2 + \Delta y^2}$$

Can step in time explicitly

Aside : What happens when $n = 1$

(A)

$$A\bar{U}^{n+1} = B\bar{U}^{n}$$

$\bar{U}$ — a vector $u_{1,1}, u_{1,2}, \ldots \; u_{ni,nj}$

A — Matrix $ni\,nj \times ni\,nj$

$\& B$

HW $\rightarrow$ Find $A \& B$.

(b) If you want second order in time
leapfrog

$$\left.\frac{\partial u}{\partial t}\right|_{i,j} = \frac{u_{i,j}^{n+1} - u_{i,j}^{n-1}}{2\Delta t}$$

$$u^{n+1} = \overline{u^{n-1}} + 2\Delta t\,[RHS]$$

$$\overline{u^{n}} = u_n + \gamma\left[\overline{u^{n-1}} - 2u_n + u_{n+1}\right]$$

$\gamma \rightarrow$ Robert - Asselin Filter $\gamma = 0.01$

(c) Crank - Nicolson scheme

$$u^{n+1} = \alpha\, u^{n+1}_{imp} + (1-\alpha)\, u^{n+1}_{explc}$$

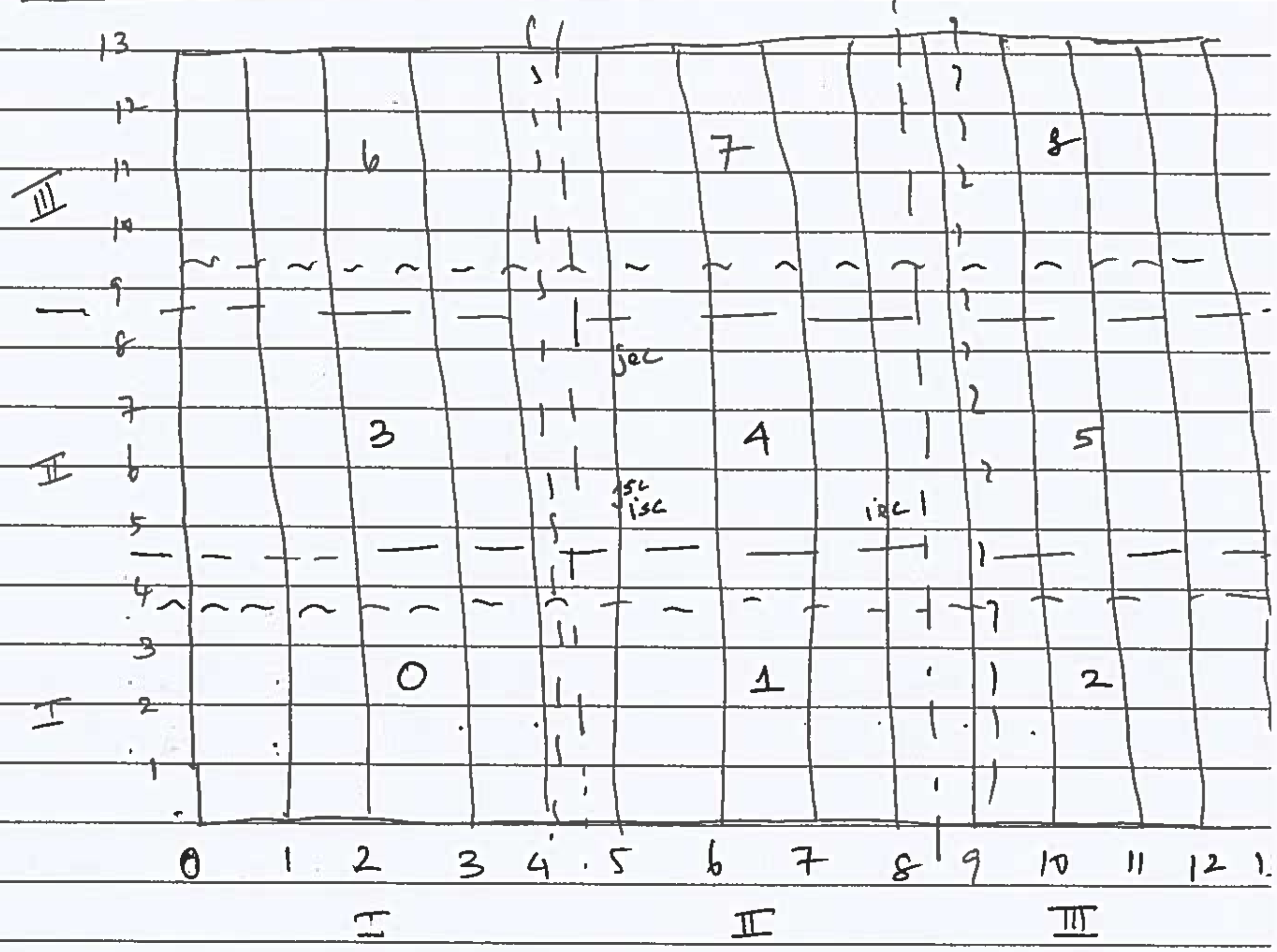
$$\alpha = 1/2$$

Coding : Fortran 90, Modules, Dynamic Array Pointer
Data structure - Type
Loops - i, inner, j - outer
Netcdf

classmate

Date
Page

Parallelisation



$n_i = 12$ $n_j = 12$ 144/9 16 → 4×4

3×3

Numbering global

4

pe numbering

x halo depends on differencing — 1 for CD, >1 for higher Diff

$isd = isc - xhalo$, $ied = iec + xhalo$

$jsd = jsc - yhalo$; $jed = jec + yhalo$

but isd, ied, jsd + jed are in different processors.

At each time we have to synchronise and update the halos by communication with the neighbours. (Need to keep a table of neighbours)

Parallel processing

→ Communication & Computation
↓ slow ↓ fast

To keep a balance between the two.

We will see how to do this with Athena map.

Parallelisation steps

To communicate we use the MPI library.
(Balaji has already told you !)

1) Domain Layout — Given npes ; factor it for x & y ;
Allow load imbalance or not

2) Domain decomposition with global numbering

1) At any pe — 2nd isd, ied, jsd, jed & neighbours.

Nice type called domain 2D which will keep all this data in place

3) Get domain components.

4) Compute — update in time on each pe

5) Update the halos ; go to 4.

I/O → Each PE writes its own output file.
combine at the end.

DDT Demo

Codes — final-20. etc.

FMS based code

↓

Does everything for you.

3 main modules → mpp-mod
mpp-domains-mod
mpp-io-mod.

Parallel programming } Michael Quinn (coded in C)

Lawrence Liver More (~~[illegible]~~)

LLNL (Blaise Barney)

Numerical Analysis: Dale A Durran