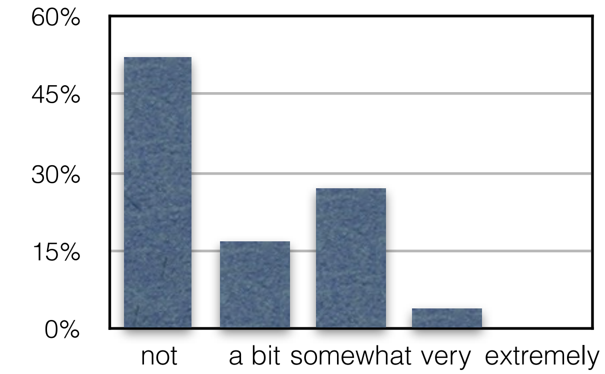
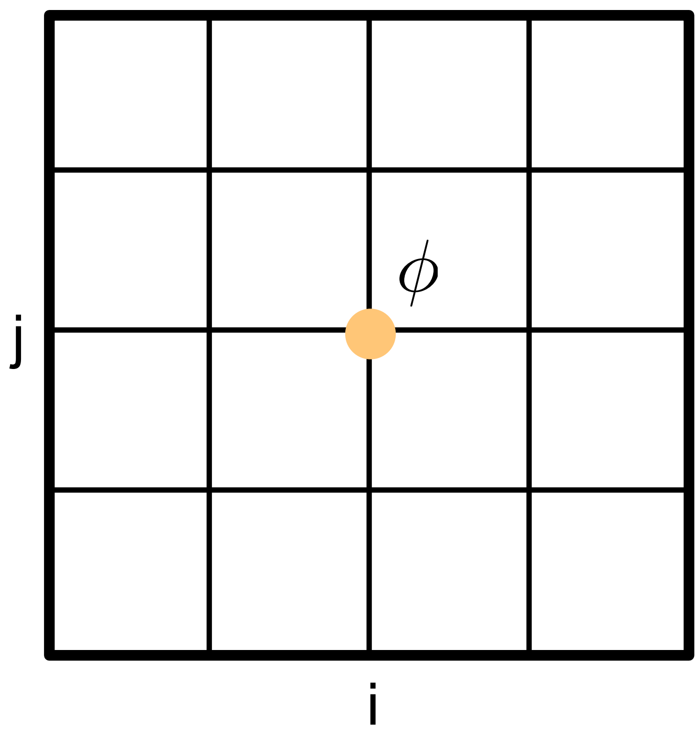
- Muddiest Points from Class 02/13
  - *"In the V-Cycle iteration method, while returning back to the finer mesh, why did we run the loop only till the second to last term and not till the last term?"*
    - the solution variable phi is usually a separate variable and not stored in the solution array eps used in the loop.
    - to not do 2 consecutive Gauss-Seidel steps on the fine mesh (one in the loop, the next at the start of the next V-cycle iteration)
  - *"Just to clarify, are we allowed to use dynamic memory allocation if we are comfortable with it, or are we restricted to using the alternative method presented on slide 2?"*
    - Use dynamic memory allocation if you like. You can also code the methods with recursive function/subroutine calls.
  - *"You mentioned that the V cycle shown was a single iteration. Do you mean that was a single iteration at a single mesh point?"*
    - No it was a single V-cycle iteration for all mesh points.
  - *"since the coarser meshes would make convergence faster, why the V-cycle starts from finest meshes? wouldn't it slow down the iteration?"*
    - If we have a good initial guess for the solution phi, then starting with a fine mesh is more efficient
  - *"What exactly does "p" represent? Is it the number of levels you can move in your coarsening procedure? For example, to coarsen from 32 to 2, would p=5?"*
    - Yes, the levels of meshes one can have up to the coarsest possible mesh with 2 elements
  - *"What are the situations where Gauss-Seidel Multi-grid is preferable to SOR and vice-versa? Or is multi-grid always going to be a better choice?"*
    - Multigrid is preferable in 99.99% of the cases.



Next: Let's revisit meshing

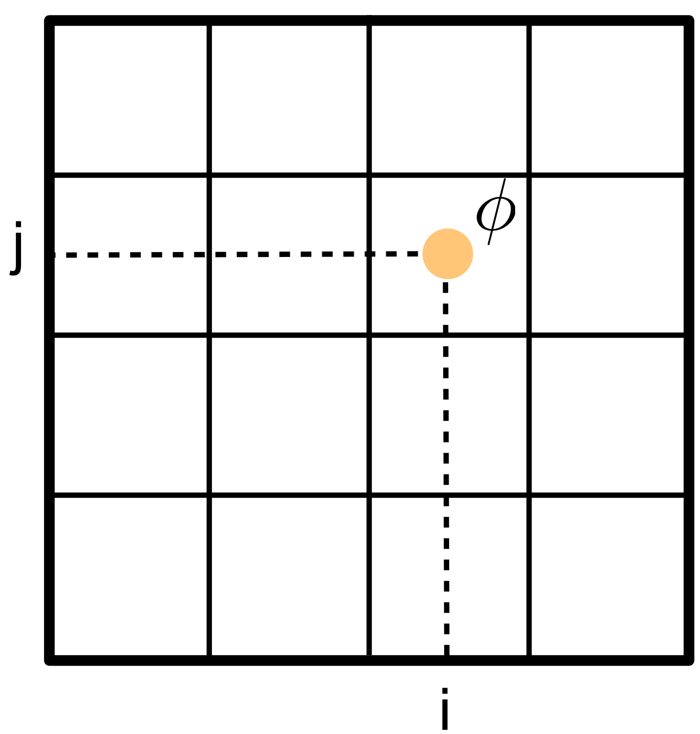
- until now, we have used the following meshes
  - variables are located at the intersection of grid lines

node based mesh



- but, we could also locate variables @ cell centers!

cell centered mesh

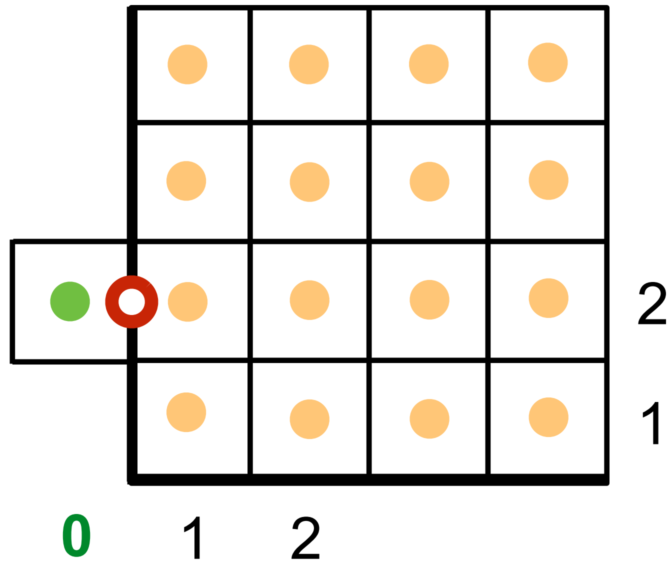


- index i,j refers to cell (element) center

How do cell centered meshes impact boundary conditions?

- **Dirichlet boundary:**
  - there's no longer a variable located on the boundary to set to the given Dirichlet value
  - Trick: add a "virtual" **ghost cell** outside the boundary
  - choose the ghost cells' value such that an interpolation to the boundary location with appropriate order is equal to the Dirichlet value
    - Example: 2nd-order

$$\phi_{bc,j} = \frac{\phi_{0,j} + \phi_{1,j}}{2} \quad \Rightarrow \quad \phi_{0,j} = 2\phi_{bc,j} - \phi_{1,j}$$

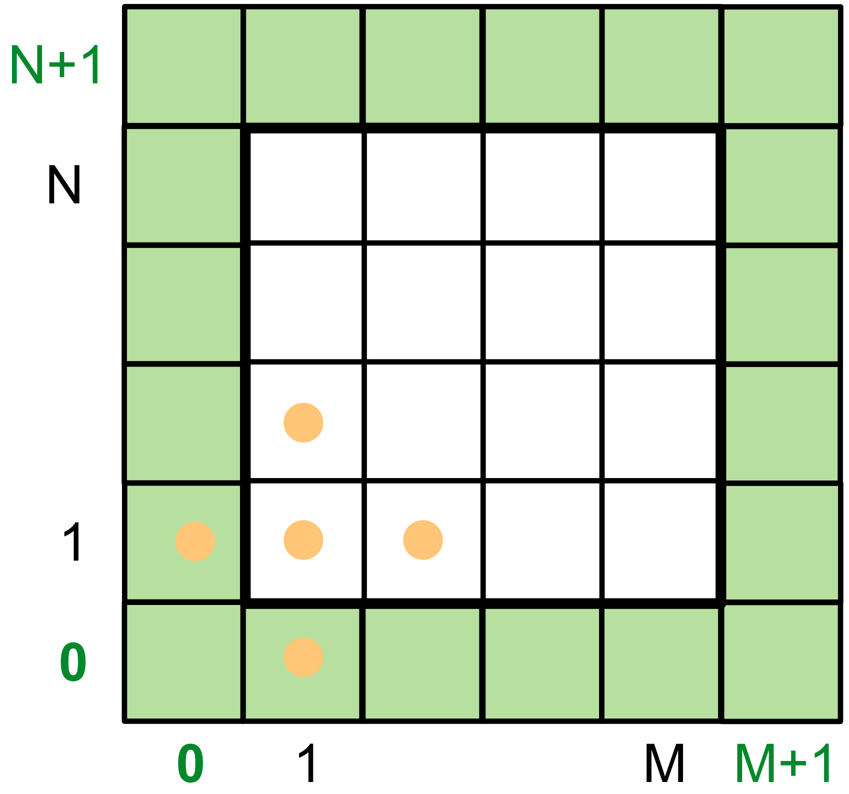


  - extends the mesh by a layer of ghost cells all around

```
phi(0:M+1,0:N+1)
```

  - Benefit: can use regular stencil even adjacent to boundaries with ghost cell values

```
for j=1:N
  for i=1:M
```



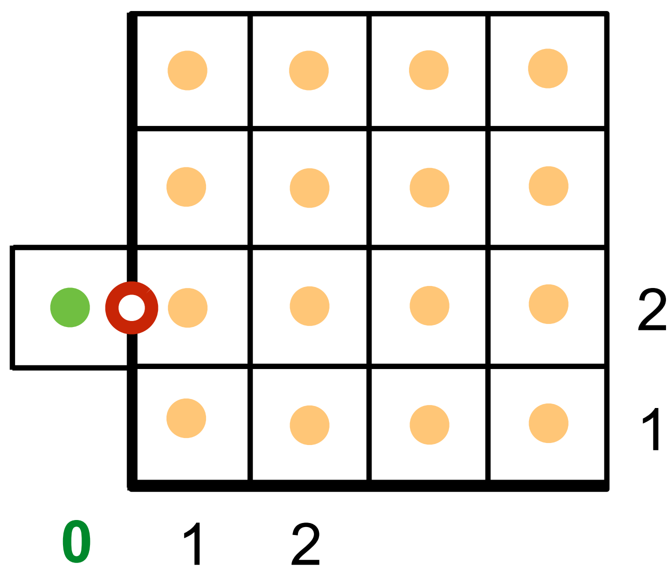
How do cell centered meshes impact boundary conditions?

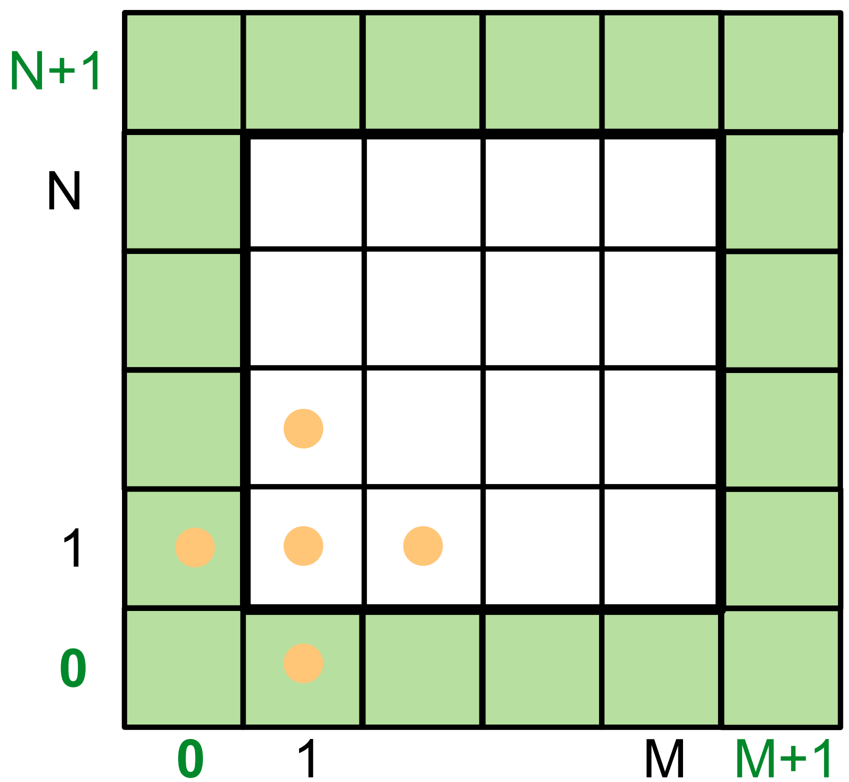
- **Neumann boundary:**
  - Trick: use ghost cell value to calculate derivative on the boundary
    - Example: 2nd-order

$$\left.\frac{\partial \phi}{\partial x}\right|_{bc,j} = \frac{\phi_{1,j} - \phi_{0,j}}{2\,\frac{h}{2}} + O(h^2)$$

$$\Rightarrow \quad \phi_{0,j} = \phi_{1,j} - h\left.\frac{\partial \phi}{\partial x}\right|_{bc}$$

    - this sets the ghost cell value!
  - Again: can use regular stencil even adjacent to boundaries with ghost cell values
  - for higher order, add additional ghost cells



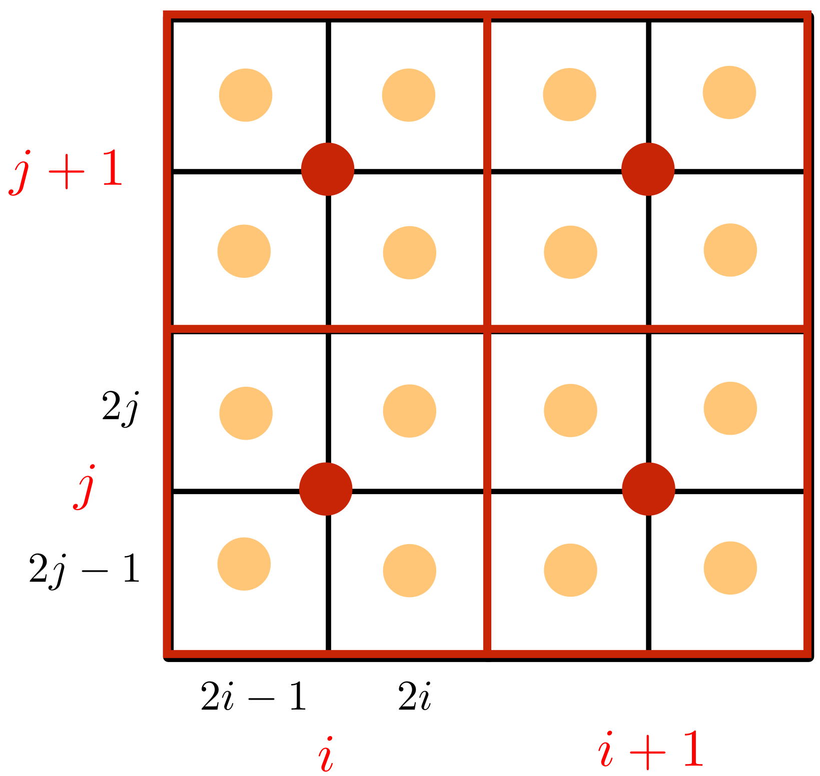


- **Solution procedure for cell centered meshes**
  - update interior cells (j=1:N, i=1:M)
  - after all interior cells are updated, directly calculate ghost cell values with updated interior values

- Small drawback: ghost cells in Gauss-Seidel are not updated and thus may lag one iteration

How do cell centered meshes impact Multigrid methods?

- **Prolongation**



here: i,j are coarse grid indices

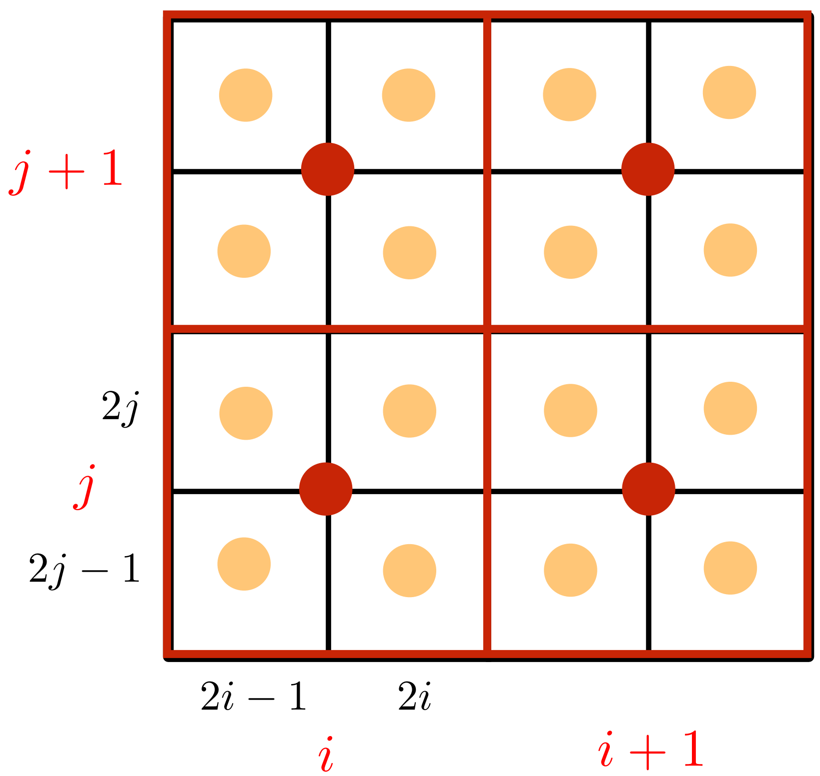
- Option #1: Constant "interpolation"

$$\epsilon^{2h\rightarrow h}_{2i-1:2i,2j-1:2j} = \epsilon^{2h}_{i} \qquad i = 1, 2, \ldots, M^{2h}\,,\ j = 1, 2, \ldots, M^{2h}$$

- Option #2: Bilinear interpolation

How do cell centered meshes impact Multigrid methods?

- **Restriction** (needs to be adjoint of Prolongation)



here: i,j are coarse grid indices

- Option #1: Adjoint to constant "interpolation"

$$r_{i,j}^{h\rightarrow 2h} = \frac{1}{4}\sum_{j'=2j-1}^{2j}\sum_{i'=2i-1}^{2i} r_{i',j'}^{h} \qquad i=1,2,\ldots,M^{2h}\,,\ j=1,2,\ldots,M^{2h}$$

- Option #2: Adjoint to bilinear interpolation

- Finally, a comment on Poisson equation with all Neumann boundaries

$$\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = f(x,y) \qquad \left.\frac{\partial\phi}{\partial n}\right|_{bc} = g(x,y)$$

  - if $\phi(x,y)$ is a solution, so is $\phi(x,y) + const$
  - iterative solution may “drift”
  - this is usually not a problem for convergence checks, since these use the residual

$$r(x,y) = f(x,y) - \left(\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2}\right)$$

  - but, excessive “drift” may cause finite precision problems, since it can lead to differences of large numbers
  - Fix: subtract the mean of ϕ from ϕ after convergence or after some number of iterations

$$\phi_{i,j} \to \phi_{i,j} - \frac{1}{MN}\sum_{j=1}^{N}\sum_{i=1}^{M}\phi_{i,j}$$

- Challenge Question:

Solve $\frac{\partial^2 \varphi}{\partial x^2} = \sin(x)$ on domain $0 \leq x \leq 2\pi$ with bc $\varphi(0) = \varphi(2\pi) = 0$

with second order central differences using Gauss-Seidel and initial guess $\varphi^{(0)} = 0$

Question: Is the exact solution to the PDE $\varphi(x) = -\sin(x)$ the solution to the Gauss-Seidel method after infinitely many iterations?

A: Yes B: No C: No Idea

Show of Hands

Discuss (1-2mins). (also discuss why)

Show of Hands

Second Model Problem: Parabolic Equations

- 1D heat equation

$$\frac{\partial \varphi}{\partial t} = \alpha \frac{\partial^2 \varphi}{\partial x^2} \qquad \text{with} \qquad \varphi = \varphi(x,t)$$

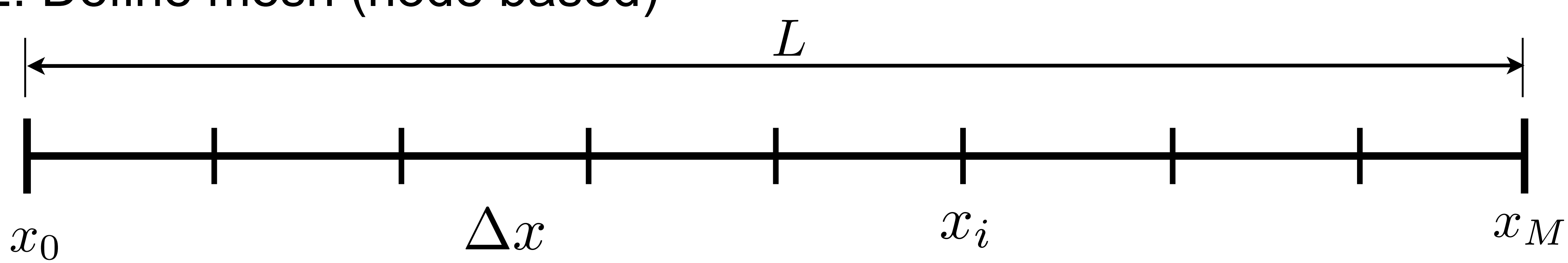
boundary conditions: $\varphi(x=0,t) = \varphi(x=L,t) = 0$

initial condition: $\varphi(x,t=0) = g(x)$

Step 1: Define solution domain

$$0 \le x \le L$$

Step 2: Define mesh (node based)



$$\Delta x = h = \frac{L}{M} \qquad\qquad x_i = ih, \quad i = 0 \dots M$$

Second Model Problem: Parabolic Equations

$$\frac{\partial \varphi}{\partial t} = \alpha \frac{\partial^2 \varphi}{\partial x^2}$$

Step 3: Approximate spatial derivatives

for example: 2nd-order central:

$$\left.\frac{\partial^2 \varphi}{\partial x^2}\right|_i = \frac{\varphi_{i+1} - 2\varphi_i + \varphi_{i-1}}{h^2} + O(\Delta h^2)$$

Step 4: Substitute into PDE

$$\left.\frac{d\varphi}{dt}\right|_i = \frac{\alpha}{h^2}\left(\varphi_{i+1} - 2\varphi_i + \varphi_{i-1}\right)$$ ⇒ now an ODE!

Step 5: Incorporate boundary conditions

$$\varphi(x=0,t) = \varphi(x=L,t) = 0 \quad \Rightarrow \quad \varphi_0 = \varphi_M = 0$$

Second Model Problem: Parabolic Equations

$$\frac{\partial \varphi}{\partial t} = \alpha \frac{\partial^2 \varphi}{\partial x^2}$$

Step 6: Matrix form (only for illustration, never code!)

$$\left.\frac{d\varphi}{dt}\right|_i = \frac{\alpha}{h^2}\left(\varphi_{i+1} - 2\varphi_i + \varphi_{i-1}\right)$$

$$\frac{d}{dt}\begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \vdots \\ \varphi_{M-3} \\ \varphi_{M-2} \\ \varphi_{M-1} \end{bmatrix} = \frac{\alpha}{h^2}\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 1 & -2 & 1 & 0 \\ 0 & \cdots & 0 & 0 & 1 & -2 & 1 \\ 0 & \cdots & 0 & 0 & 0 & 1 & -2 \end{bmatrix}\begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \vdots \\ \varphi_{M-3} \\ \varphi_{M-2} \\ \varphi_{M-1} \end{bmatrix}$$

$$\frac{d\vec{\varphi}}{dt} = A\vec{\varphi}$$ $\Rightarrow$ semi-discrete form $\Rightarrow$ many ODEs

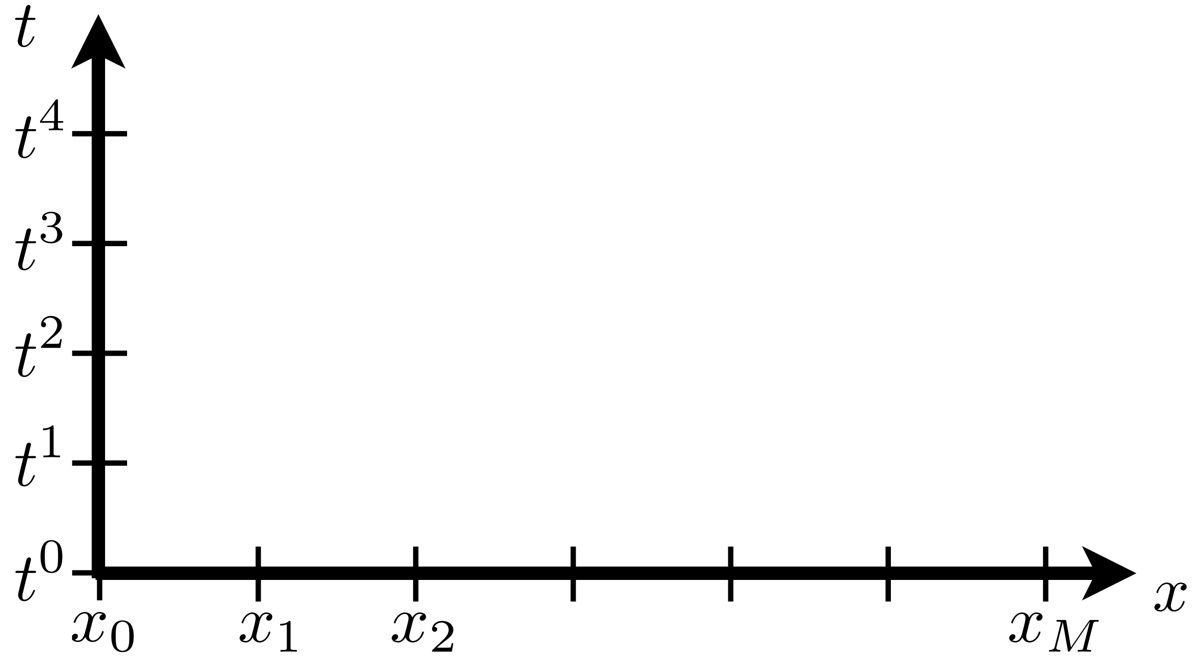
never solve this directly

Second Model Problem: Parabolic Equations

$$\frac{\partial \varphi}{\partial t} = \alpha \frac{\partial^2 \varphi}{\partial x^2}$$

Step 7: Solve (but how?)

- discretize in time: $t^n = n\Delta t\,, \qquad n = 0, 1, 2, \ldots$

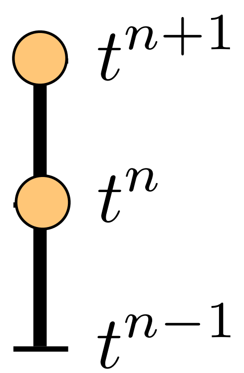


- use finite difference approximation for $\left.\dfrac{d\varphi}{dt}\right|_i$

  - for example: 1st-order forward

$$\left.\frac{d\varphi}{dt}\right|_i^n = \frac{1}{\Delta t}\left(\varphi_i^{n+1} - \varphi_i^n\right)$$

stencil:



Second Model Problem: Parabolic Equations

$$\frac{\partial \varphi}{\partial t} = \alpha \frac{\partial^2 \varphi}{\partial x^2}$$

Step 7: Solve (but how?)

- substitute into ODE

$$\left.\frac{d\varphi}{dt}\right|_i = \frac{\alpha}{h^2}\left(\varphi_{i+1} - 2\varphi_i + \varphi_{i-1}\right) \qquad \left.\frac{d\varphi}{dt}\right|_i^{n} = \frac{1}{\Delta t}\left(\varphi_i^{n+1} - \varphi_i^n\right)$$

$$\frac{\varphi_i^{n+1} - \varphi_i^n}{\Delta t} = \frac{\alpha}{h^2}\left(\varphi_{i+1}^{n} - 2\varphi_i^{n} + \varphi_{i-1}^{n}\right)$$

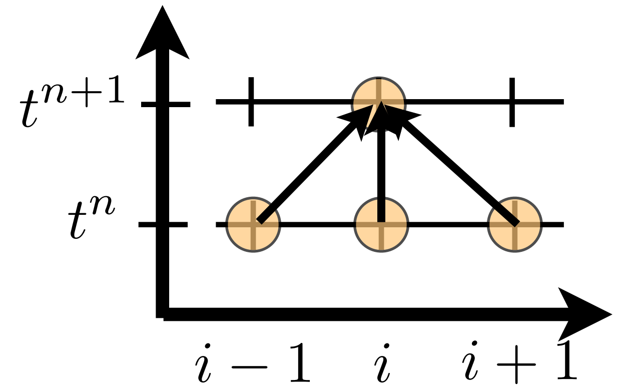
$$\boxed{\varphi_i^{n+1} = \varphi_i^n + \frac{\alpha \Delta t}{h^2}\left(\varphi_{i+1}^n - 2\varphi_i^n + \varphi_{i-1}^n\right)}$$

FTCS <u>F</u>orward <u>T</u>ime <u>C</u>entral <u>S</u>pace

- FTCS expresses a single unknown, φ_i^{n+1}, as a function of only knowns!

⇒ feature of **explicit** methods

solution at t^{n+1} depends only on solution at t^n

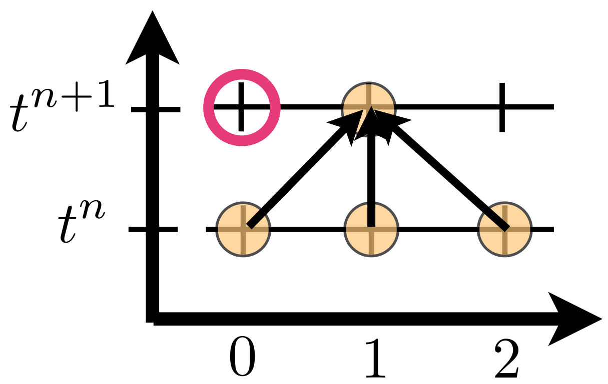


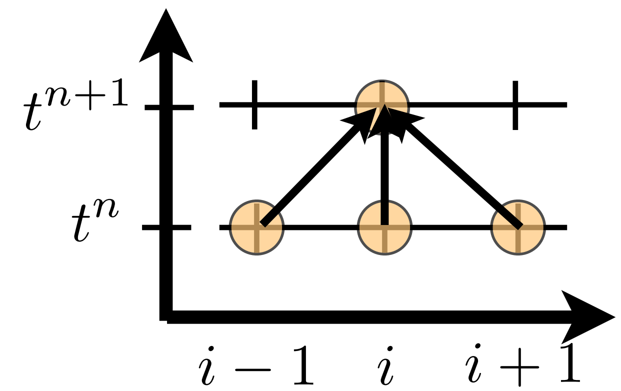
Second Model Problem: Parabolic Equations

$$\frac{\partial \varphi}{\partial t} = \alpha \frac{\partial^2 \varphi}{\partial x^2}$$

Step 7: Solve (but how?)

- BUT: problem at boundary





boundary point (bc) does not influence the solution at same t !

- boundaries lag by one time step
- this violates characteristics of parabolic equations

Second Model Problem: Parabolic Equations

$$\frac{\partial \varphi}{\partial t} = \alpha \frac{\partial^2 \varphi}{\partial x^2}$$

Step 7: Solve (but how?)

- Alternative: use backwards time difference: **Laasonen Method (BTCS)**

$$\left.\frac{d\varphi}{dt}\right|_i^{n+1} = \frac{1}{\Delta t}\left(\varphi_i^{n+1} - \varphi_i^n\right) \qquad \left.\frac{d\varphi}{dt}\right|_i = \frac{\alpha}{h^2}\left(\varphi_{i+1} - 2\varphi_i + \varphi_{i-1}\right)$$

$$\frac{\varphi_i^{n+1} - \varphi_i^n}{\Delta t} = \frac{\alpha}{h^2}\left(\varphi_{i+1}^{n+1} - 2\varphi_i^{n+1} + \varphi_{i-1}^{n+1}\right)$$

Problem: no longer explicit, but now implicit

- gather all $n+1$ terms on left hand side

$$\boxed{\frac{\alpha\Delta t}{h^2}\varphi_{i-1}^{n+1} - \left(1 + 2\frac{\alpha\Delta t}{h^2}\right)\varphi_i^{n+1} + \frac{\alpha\Delta t}{h^2}\varphi_{i+1}^{n+1} = -\varphi_i^n}$$

$$\Rightarrow \quad a_i^n\varphi_{i-1}^{n+1} + b_i^n\varphi_i^{n+1} + c_i^n\varphi_{i+1}^{n+1} = d_i^n$$

$\Rightarrow$ tri-diagonal system

$\Rightarrow$ solve directly using Gauss (see Class 5)

$\Rightarrow$ much more work than FTCS! So, what's the benefit?

$\Rightarrow$ need to discuss accuracy, stability, and consistency

- Definitions:

1. **Consistency**: numerical approximation approaches PDE as $\Delta x, \Delta y, \Delta t \to 0$

2. **Stability**: numerical solution remains bounded

3. **Convergence**: numerical solution approaches PDE solution as $\Delta x, \Delta y, \Delta t \to 0$

turns out if 1. and 2. are true, then 3. is true for linear, well posed initial value problems