- How to calculate error measures for more than 1 mesh point, i.e. the entire mesh

1. calculate error at each mesh point

$$e_i = f'_i - f'_{exact}(x_i) \qquad\qquad i = 0 \dots N$$

2. calculate error norms

$$L_\infty = \max_{i=0 \dots N} |e_i| \qquad L_1 = \frac{1}{N+1}\sum_{i=0}^{N} |e_i| \qquad L_2 = \sqrt{\frac{1}{N+1}\sum_{i=0}^{N} e_i^2}$$

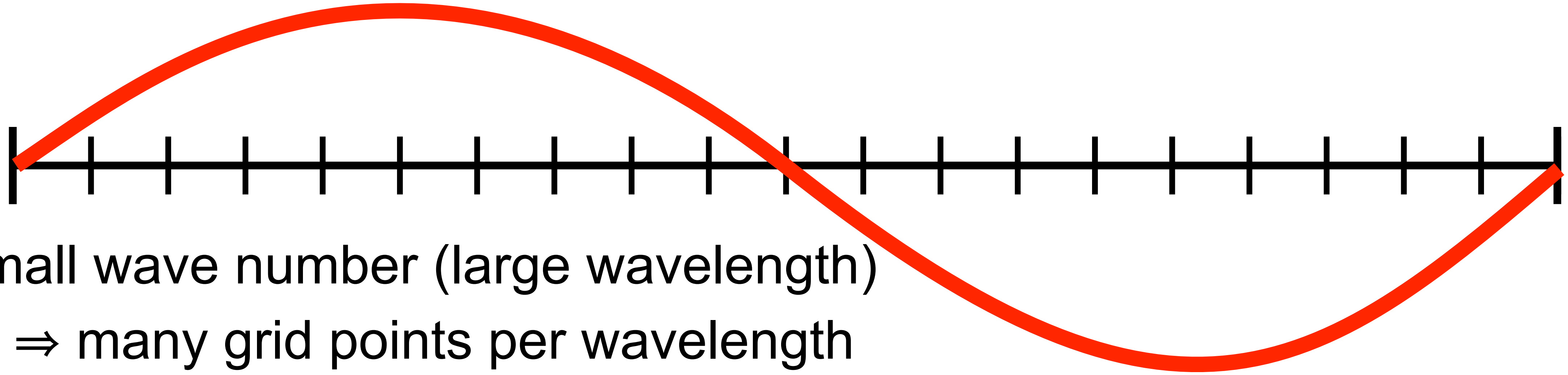
3. calculate observed order of accuracy o by comparing 2 grids with mesh spacing h_1 and h_2 and error norms $L(h_1)$ and $L(h_2)$

$$o = \frac{\log \frac{L(h_1)}{L(h_2)}}{\log \frac{h_1}{h_2}}$$

- usually done for grids with spacing ratios of $h_2/h_1 = 2$

- So far: We can construct finite difference formulas for a given stencil and determine the **formal order of accuracy**
- Recap: Formal order of accuracy tells us by how much the error decreases if we refine the mesh
- That's good, but we would also like to know how good these formulas are for certain classes of special functions
- What special functions? **sinusoidals**
- Why?
  - because sinusoidals are orthogonal basis functions (FFT)
  - can express any function as sum of sinusoidals (discontinuous functions are problematic, though)
  - some CFD methods (spectral methods) are based on sinusoidals

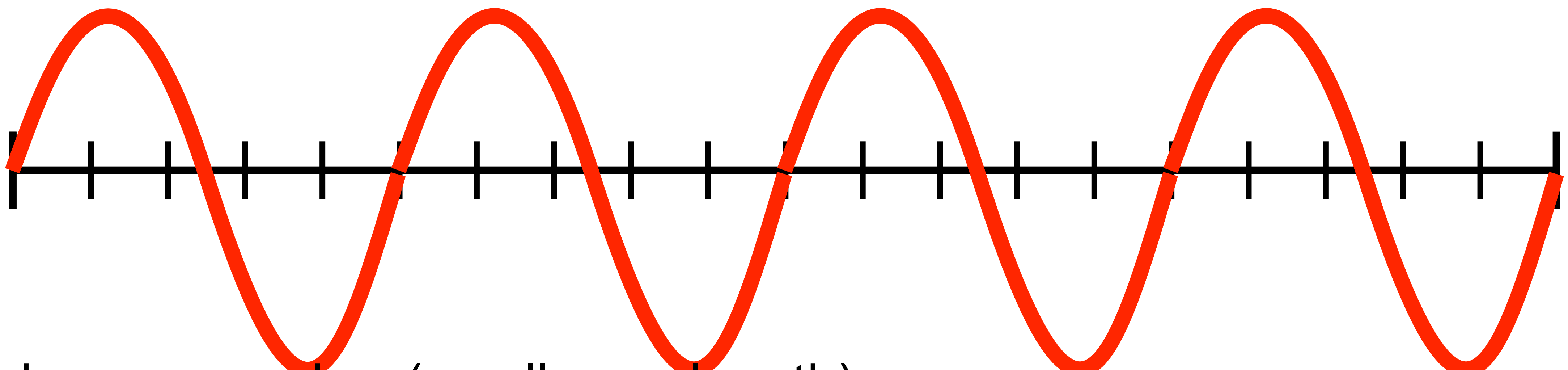
- Examples:



small wave number (large wavelength)

⇒ many grid points per wavelength

⇒well resolved



high wave number (small wavelength)

⇒ few grid points per wavelength

⇒poorly resolved

• Modified Wave Number

- consider a pure harmonic function:

$$f(x) = e^{ikx}$$

i: imaginary number: $i = \sqrt{-1}$

k: wave number: $k = \frac{2\pi}{L} n\,, n = 0,\ 1,\ 2,\ \ldots,\ \frac{N}{2}$

Nyquist frequency

L: domain length

- exact derivative is:

$$f'(x) = ike^{ikx} = ikf(x)$$

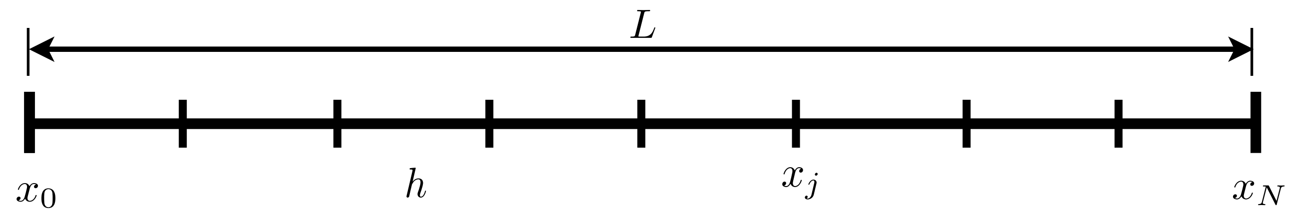
this is the reason why Fourier (spectral) methods are popular:
One can calculate the **exact** derivative in Fourier space!

- but, we are using finite differences! What's the derivative of f(x) then?
- Example: 2nd-order central difference for first derivative
- Note: modified wave number analysis works for higher derivates as well!

- Example: 2nd-order central differences

$$f(x) = e^{ikx} \qquad k = \frac{2\pi}{L} n$$

$$\left.\frac{\partial f}{\partial x}\right|_{x_j} \approx \frac{f_{j+1} - f_{j-1}}{2h}$$



$$h = \frac{L}{N}, \; x_j = \frac{L}{N} j$$

$$f_j = e^{ikx_j} = e^{i\frac{2\pi}{L} n \frac{L}{N} j} = e^{i\frac{2\pi n}{N} j}$$

$$f_{j+1} = e^{i\frac{2\pi n}{N}(j+1)} \qquad\qquad f_{j-1} = e^{i\frac{2\pi n}{N}(j-1)}$$

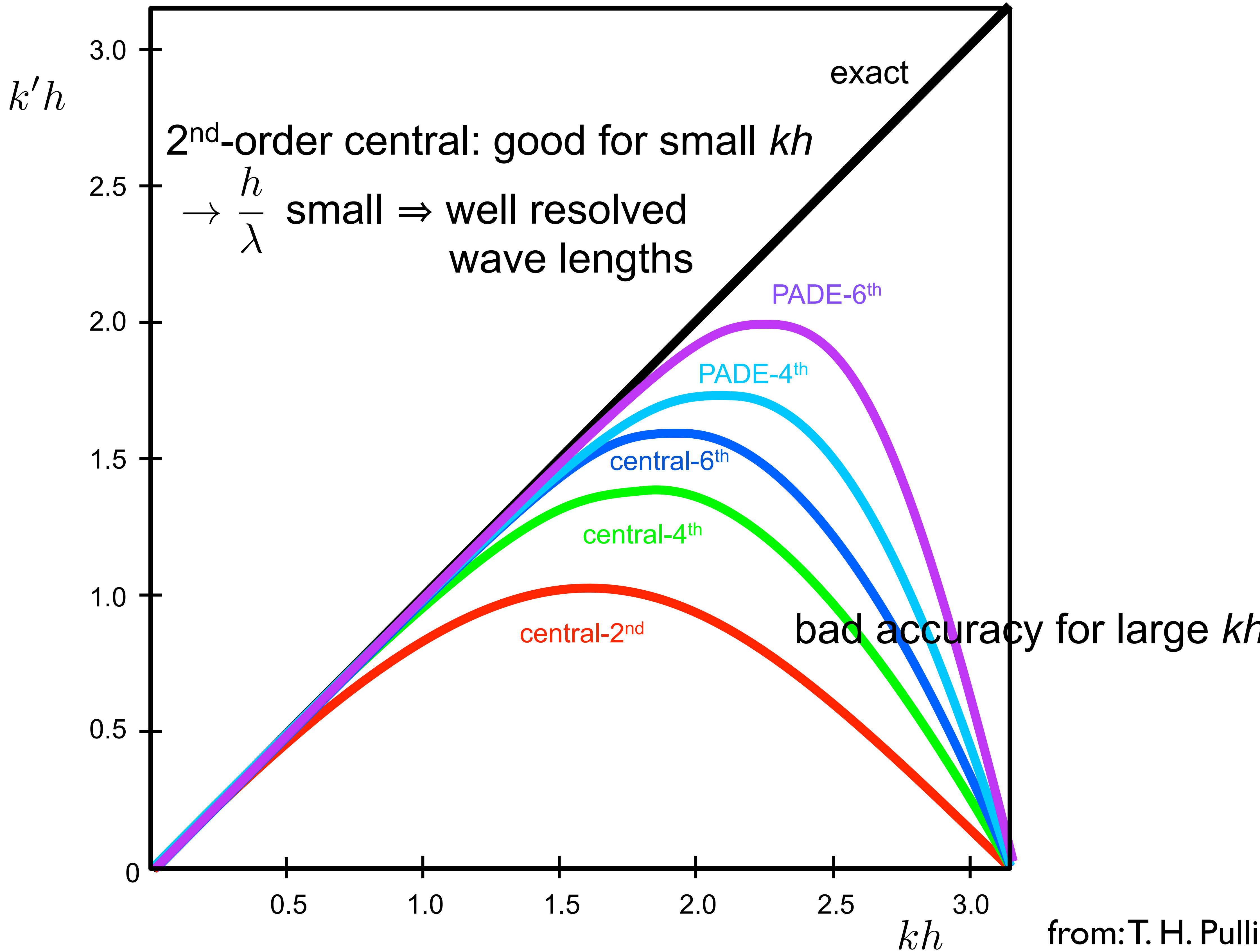
$$\left.\frac{\partial f}{\partial x}\right|_{x_j} \approx \frac{f_{j+1} - f_{j-1}}{2h} = \frac{e^{i\frac{2\pi n}{N}(j+1)} - e^{i\frac{2\pi n}{N}(j-1)}}{2h} = \frac{e^{i\frac{2\pi n}{N}} - e^{-i\frac{2\pi n}{N}}}{2h} e^{i\frac{2\pi n}{N} j}$$

$$= \frac{2i \sin\left(\frac{2\pi n}{N}\right)}{2h} f_j = i\frac{\sin\left(\frac{2\pi n}{N}\right)}{h} f_j = ik' f_j$$

$$k' = \frac{\sin\left(\frac{2\pi n}{N}\right)}{h} = \frac{\sin\left(\frac{2\pi n}{L}\frac{L}{N}\right)}{h} = \frac{\sin(kh)}{h}$$

$$\boxed{\left.\frac{\partial f}{\partial x}\right|_{x_j} \approx ik' f_j \quad \text{with} \quad k' = \frac{\sin(kh)}{h}}$$

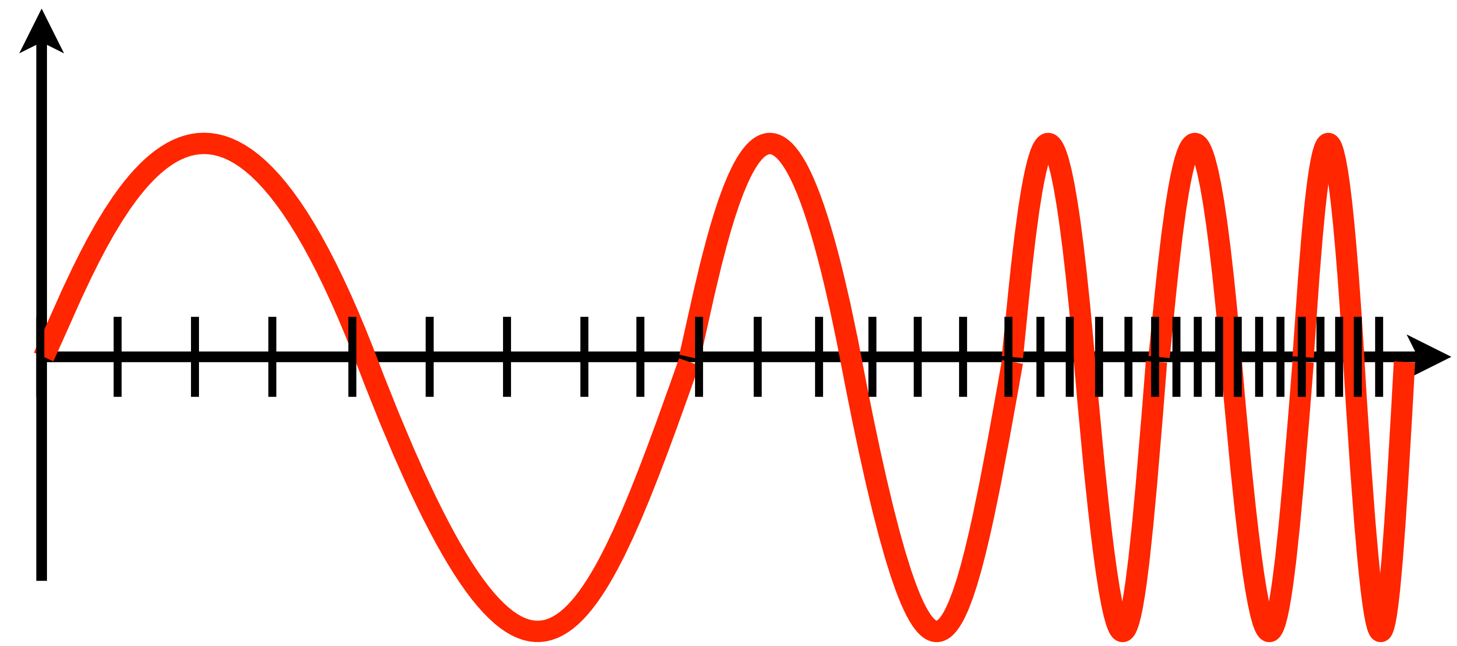
- Plot k'h vs kh: Modified wave number plot



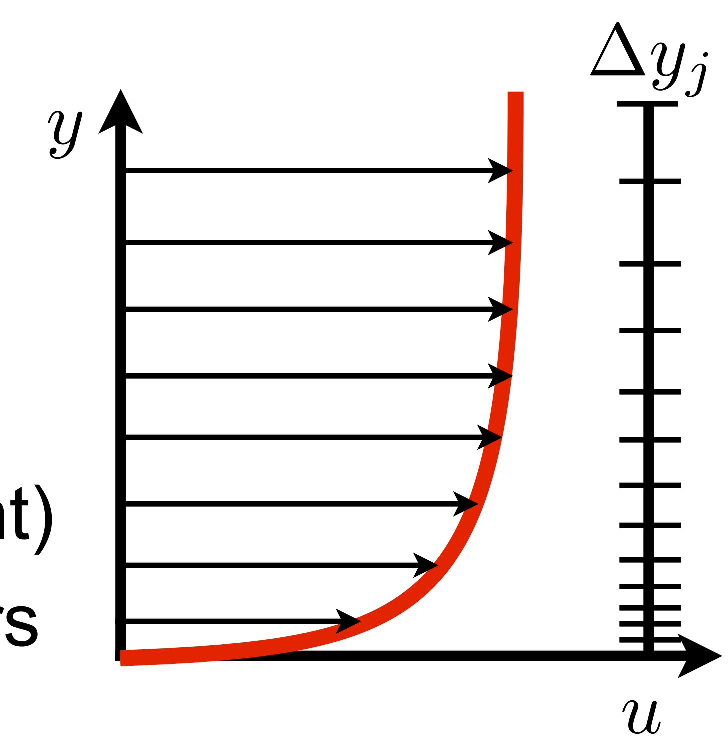
from: T. H. Pulliam

Non-Uniform Grids

- Example Scenario:

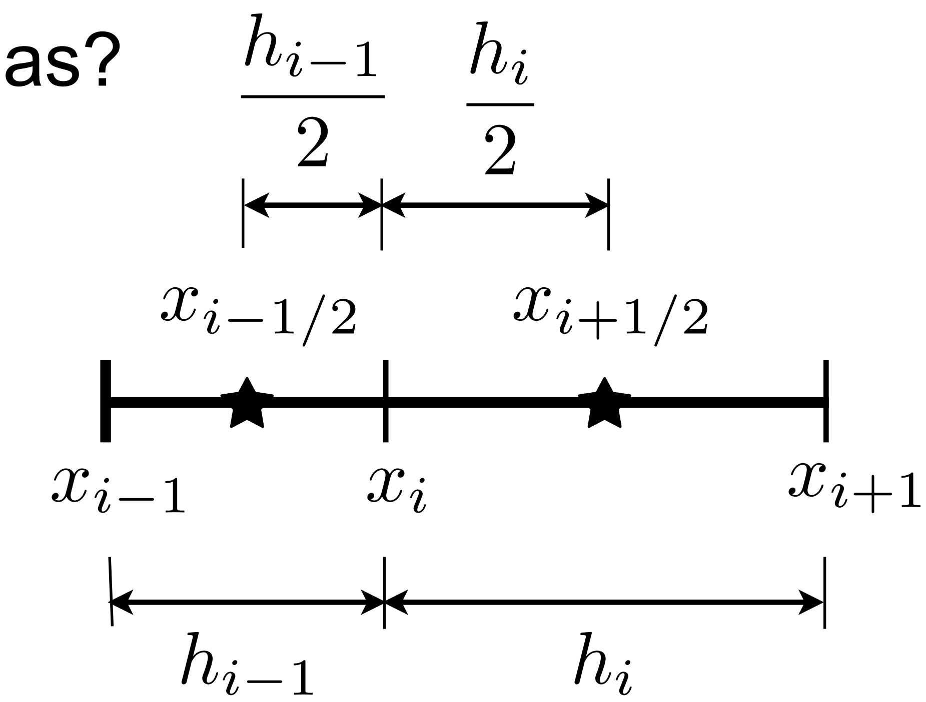


- from modified wave number analysis we know we would like kh to be small
- but not too small for efficiency reasons! (as $h \downarrow \Rightarrow N \uparrow$)
- Strategy: make mesh spacing non-uniform!
  - use large h where k is small
  - use small h where k is large
- Do we know *a-priori* where k will be small or large?
  - often times not ⇒ AMR (**A**daptive **M**esh **R**efinement)
  - but in some cases we do! example: boundary layers



Non-Uniform Grids

- How does this change the finite difference formulas?
  - Example: ‘central’-difference:



$$f_i' = \frac{f_{i+1} - f_{i-1}}{x_{i+1} - x_{i-1}}$$

$$f_i'' = (f_i')' = \frac{f'_{i+1/2} - f'_{i-1/2}}{x_{i+1/2} - x_{i-1/2}} = \frac{f'_{i+1/2} - f'_{i-1/2}}{\frac{1}{2}\left(h_{i-1} + h_i\right)}$$

$$f'_{i+1/2} = \frac{f_{i+1} - f_i}{x_{i+1} - x_i} = \frac{f_{i+1} - f_i}{h_i}$$

$$f'_{i-1/2} = \frac{f_i - f_{i-1}}{x_i - x_{i-1}} = \frac{f_i - f_{i-1}}{h_{i-1}}$$

$$f_i'' = \frac{\frac{f_{i+1} - f_i}{h_i} - \frac{f_i - f_{i-1}}{h_{i-1}}}{\frac{1}{2}\left(h_{i-1} + h_i\right)} = \boxed{\frac{2}{h_{i-1}\left(h_{i-1} + h_i\right)}} f_{i-1} - \boxed{\frac{2}{h_i h_{i-1}}} f_i + \boxed{\frac{2}{h_i\left(h_{i-1} + h_i\right)}} f_{i+1}$$

metric coefficients

Non-Uniform Grids

- Drawback:
  - non-uniform grid approximations tend to be of lower order accuracy
  - Why? because higher order Taylor series terms no longer cancel
- back to example:
  - just derived formula is only 1st order accurate

$$f_i'' = \frac{2}{h_{i-1}\left(h_{i-1}+h_i\right)} f_{i-1} - \frac{2}{h_i h_{i-1}} f_i + \frac{2}{h_i\left(h_{i-1}+h_i\right)} f_{i+1} + O(h)$$

  - but uniform mesh formula for f'' is second order!

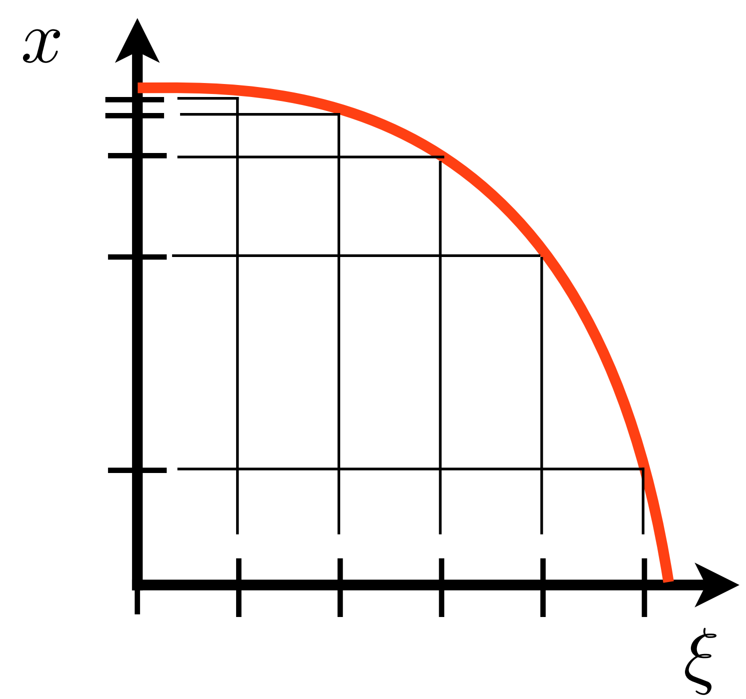
$$f_i'' = \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} + O(h^2)$$

Non-Uniform Grids

- Alternative: **Coordinate transformations**
  - Example:

$$\xi = \arccos(x) \qquad 0 \leq x \leq 1 \quad \rightarrow \quad 0 \leq \xi \leq \frac{\pi}{2}$$

- equal spacing in ξ: $\xi_i = \frac{\pi}{2N} i$ ⇒ non-uniform spacing in x_i



Non-Uniform Grids

- Alternative: **Coordinate transformations**
  - in general:

$$\xi = g(x)$$

- chain rule:

$$\frac{df}{dx} =$$

$$\frac{d^2 f}{dx^2} = \frac{d}{dx}\left(\frac{df}{dx}\right)$$

- use finite difference approximations for uniform meshes for $df/d\xi$, $d^2f/d\xi^2$
- use exact analytical derivatives for g', g'', ..., if g is a known function

First Model Problem

- Poisson Equation in 2D: $\nabla^2 \varphi = f$ or $\dfrac{\partial^2 \varphi}{\partial x^2} + \dfrac{\partial^2 \varphi}{\partial y^2} = f(x,y)$
  - variations of this are

    Laplace equation: $\nabla^2 \varphi = 0$

    Helmholtz equation: $\nabla^2 \varphi + \alpha^2 \varphi = 0$

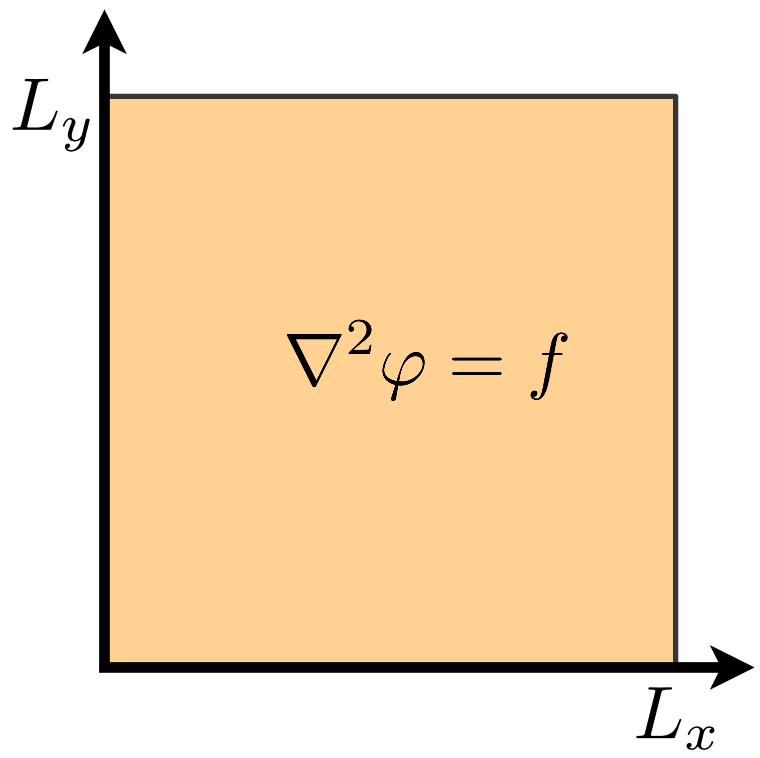
  - elliptic equation!
    - closed solution domain
    - instantaneous propagation of information
    - requires boundary conditions:

$$c_1 \varphi + c_2 \frac{\partial \varphi}{\partial n} = g$$

n: coordinate normal to boundary

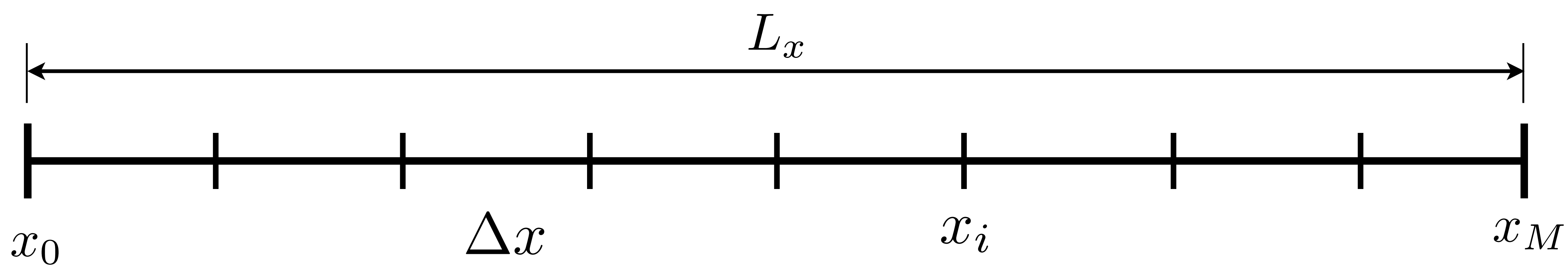
Step 1: Define Solution Domain

$0 \le x \le L_x$ and $0 \le y \le L_y$



Step 2: Define Mesh

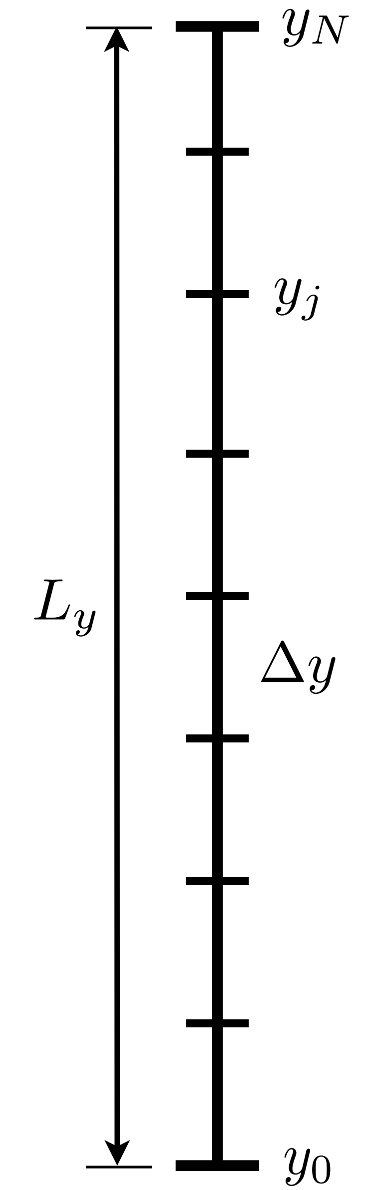
- here: for simplicity: Cartesian, equidistant in each direction
- x-direction:



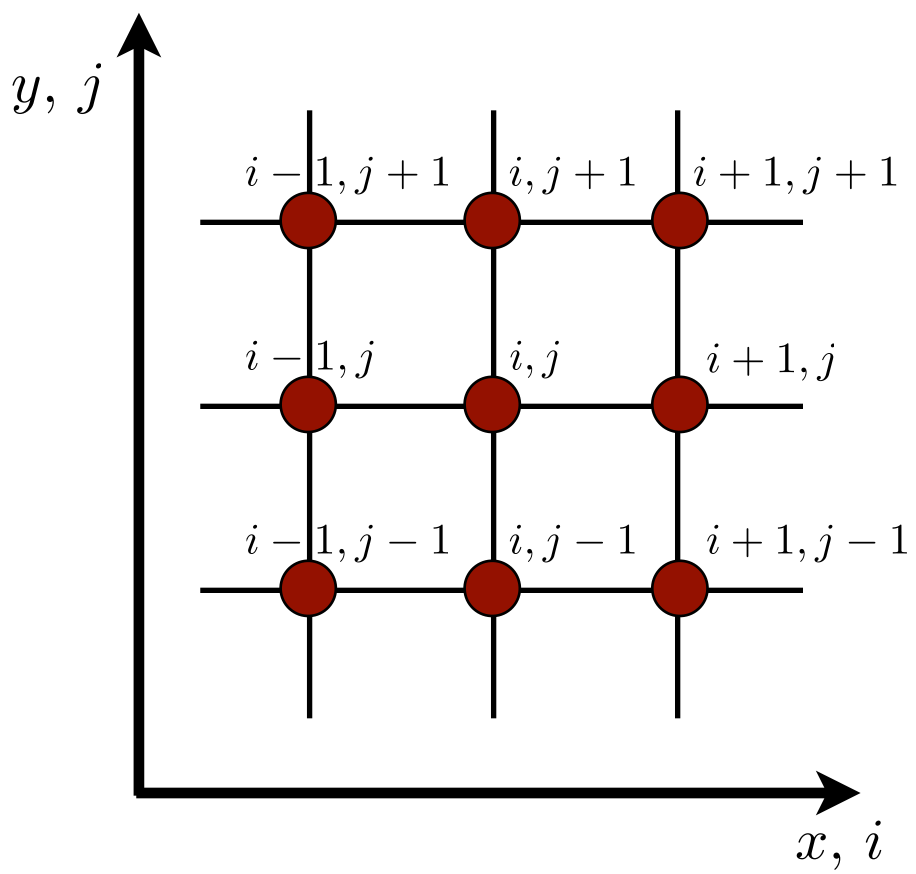
- $M+1$ points: $x_0, x_1, x_2, \ldots x_{M-1}, x_M$
- points at boundary: x_0 and x_M
- M-1 interior points: $x_1, x_2, \ldots x_{M-1}$
- grid spacing: $\Delta x = \dfrac{L_x}{M}$

Step 2: Define Mesh

- y-direction:
  - $N+1$ points: $y_0, y_1, y_2, \ldots y_{N-1}, y_N$
  - points at boundary: y_0 and y_N
  - $N-1$ interior points: $y_1, y_2, \ldots y_{N-1}$
  - grid spacing: $\Delta y = \dfrac{L_y}{N}$



- 2D Mesh:



Step 3: Approximate Spatial Derivatives

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} = f(x, y)$$

- using Taylor tables for 3-point centered stencil:

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

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

Step 4: Substitute into PDE

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

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

$$\frac{\partial^2\varphi}{\partial x^2} + \frac{\partial^2\varphi}{\partial y^2} = f(x,y)$$

$$\frac{\varphi_{i+1,j} - 2\varphi_{i,j} + \varphi_{i-1,j}}{\Delta x^2} + \frac{\varphi_{i,j+1} - 2\varphi_{i,j} + \varphi_{i,j-1}}{\Delta y^2} = f_{i,j}$$

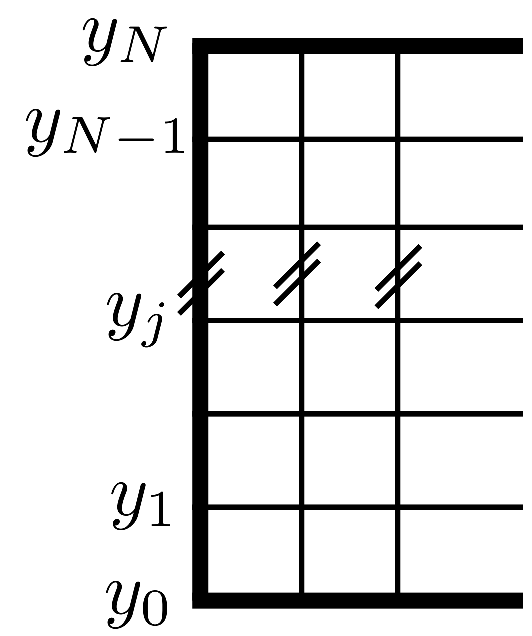
➡ this is now a difference equation and no longer a differential equation!

- let’s assume for simplicity: $\Delta x = \Delta y = \Delta$

$$\boxed{\varphi_{i+1,j} - 4\varphi_{i,j} + \varphi_{i-1,j} + \varphi_{i,j+1} + \varphi_{i,j-1} = \Delta^2 f_{i,j}}$$

  - valid only for interior points, i.e. for $i=1,..,M-1$ and $j=1,..,N-1$
  - **not valid** for boundary points, i.e. for $i=0,M$ or $j=0,N$

Step 5: Incorporate Boundary Conditions



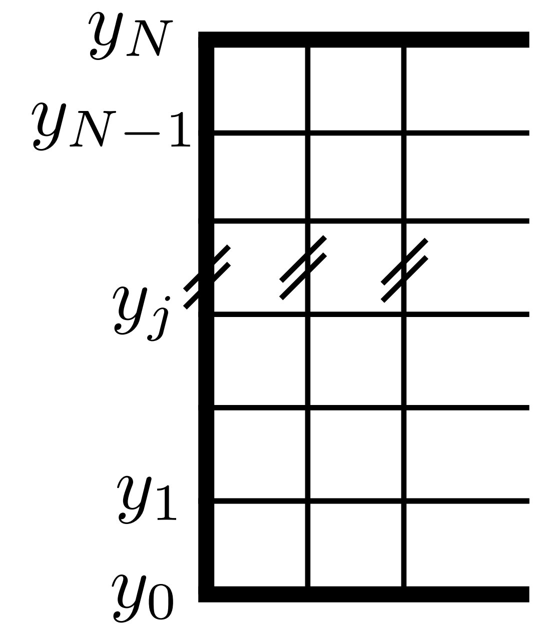
- Example #1:
  - Dirichlet at x_0: $\boxed{\varphi_{0,j} \text{ given:}}$ $j=1, N-1$
    - finite difference formula at $i=1$:

$$\varphi_{2,j} - 4\varphi_{1,j} + \boxed{\varphi_{0,j}} + \varphi_{1,j+1} + \varphi_{1,j-1} = \Delta^2 f_{1,j}$$

$$\varphi_{2,j} - 4\varphi_{1,j} + \varphi_{1,j+1} + \varphi_{1,j-1} = \Delta^2 f_{1,j} - \varphi_{0,j} \qquad \text{for } j = 1,N\text{-}1$$

➡ Dirichlet boundary conditions modify right-hand side only!

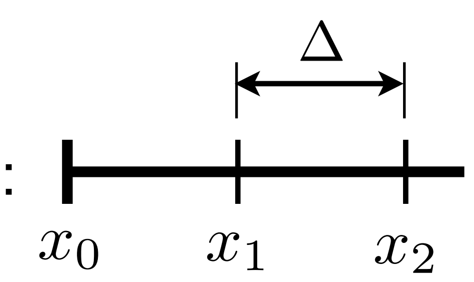
Step 5: Incorporate Boundary Conditions



- Example #2:
  - Neumann at x_0: $$\left.\frac{\partial \varphi}{\partial x}\right|_{x_0} = g(y)$$
    - approximate derivative by one-sided finite difference formula

$$\left.\frac{\partial \varphi}{\partial x}\right|_{0,j} = \frac{-3\varphi_{0,j} + 4\varphi_{1,j} - \varphi_{2,j}}{2\Delta}$$

stencil:



    - solve for $\varphi_{0,j}$

$$\varphi_{0,j} = \frac{1}{3}\left(-2\Delta g_j + 4\varphi_{1,j} - \varphi_{2,j}\right)$$

$$\varphi_{2,j} - 4\varphi_{1,j} + \varphi_{0,j} + \varphi_{1,j+1} + \varphi_{1,j-1} = \Delta^2 f_{1,j}$$

    - substitute into finite difference formula at $i=1$

$$\frac{2}{3}\varphi_{2,j} - \frac{8}{3}\varphi_{1,j} + \varphi_{1,j+1} + \varphi_{1,j-1} = \Delta^2 f_{i,j} + \frac{2}{3}\Delta g_j$$

➡ Neumann boundary conditions modify both sides!