Muddiest Points from Class 01/25

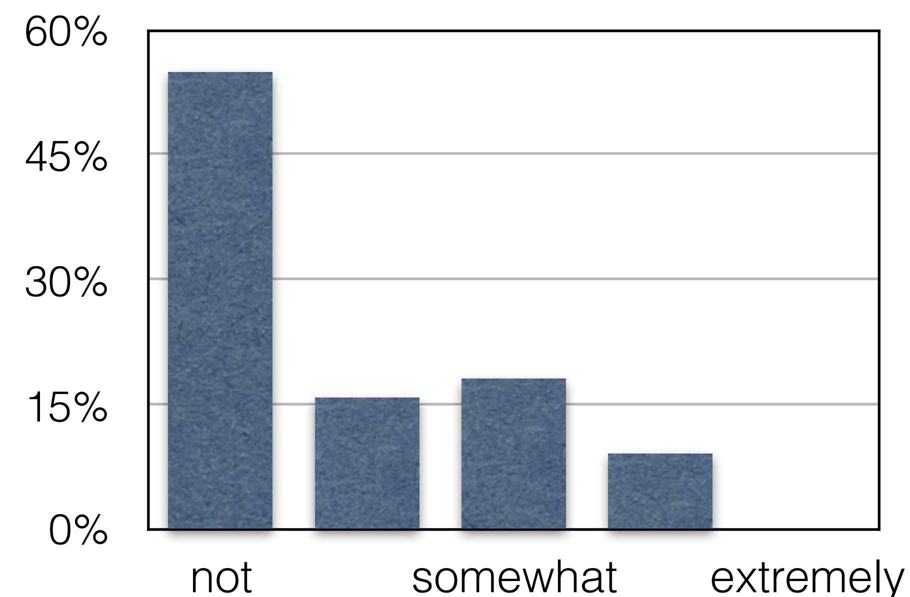
- *"Can PADE schemes be created with a stencil of 4 or more points?"*
  - yes, it can be done with any stencil. However the resulting matrix will not be tri-diagonal and thus more expensive to solve
- *"In a Taylor table, how do you know how many and which columns you can set to zero?"*
  - the number of columns to set to zero is equal to the number of coefficients a_i starting with the left-most
- *"I do not understand how to store the three diagonal rows from the tri diagonal matrix into their own separate vectors. Do you create a loop, or is there a built in function in matlab, fortran, or c/c++ that we use?"*
  - First **allocate** the variable and then use a loop or an implied loop
  - Example for Matlab

```
b = 4*ones(P,1);
b(1) = 1;
b(P) = 1;
```

  - Example for Fortran

```
allocate(b(P))
b(1) = 1
b(2:P-1) = 4
b(P) = 1
```

- *"What does PADE stand for? (used on slide 7, 8, 12, etc)"*
- *"what is "PADE"?"*
  - The name goes back to the French mathematician Henri Padé, who first developed the approach in 1890



- Let's revisit the error of the PADE scheme
  - How to calculate error measures for more than 1 mesh point, i.e. the entire mesh

1. calculate error at each mesh point

$$e_i = f'_{exact} - f'_i \qquad i = 0 \ldots N$$

2. calculate error norms

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

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 $h_2/h_1 = 2$

- Let's revisit the error of the PADE scheme

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

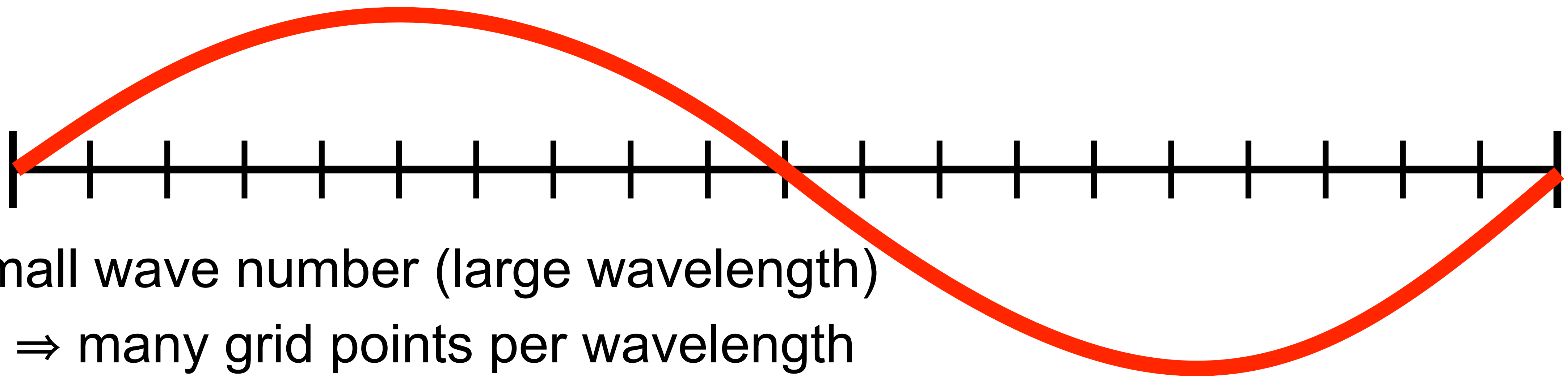
- recall we had a 4-th order formula in the interior and 3-rd order formulas at the boundaries
- if we consider only the infinity-norm, PADE will be order 3
- if we consider the 1-norm, PADE will be approximately order 4
- if we consider the 2-norm, PADE will be approximately order 3.5

Why?

- 1- and 2-norms are averages
- only the 2 boundary points out of N+1 total points are order 3
- however, the 2-norm weighs larger error more due to the square
- there will be some limited “cross-pollution” due to the coupled system

- 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

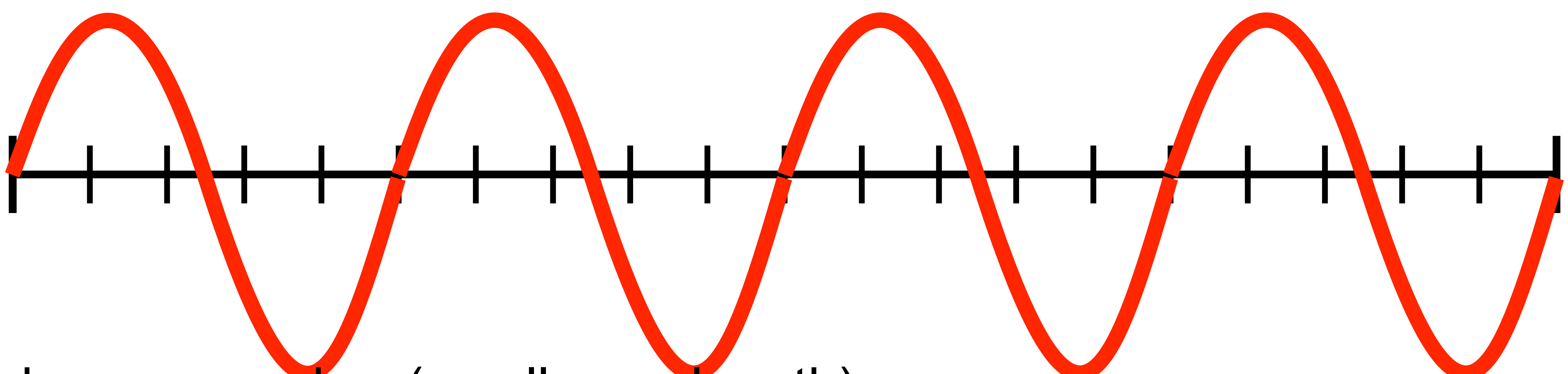
- Examples:



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:

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

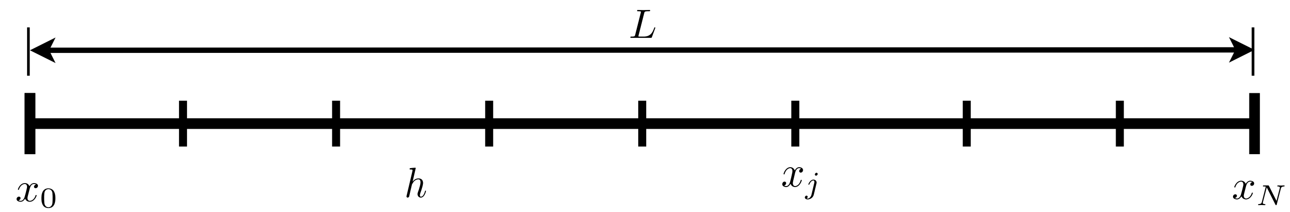
this is the reason why Fourier (spectral) methods are popular:
One can calculate the **exact** derivative in Fourier space!

- 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

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

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



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

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

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

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

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

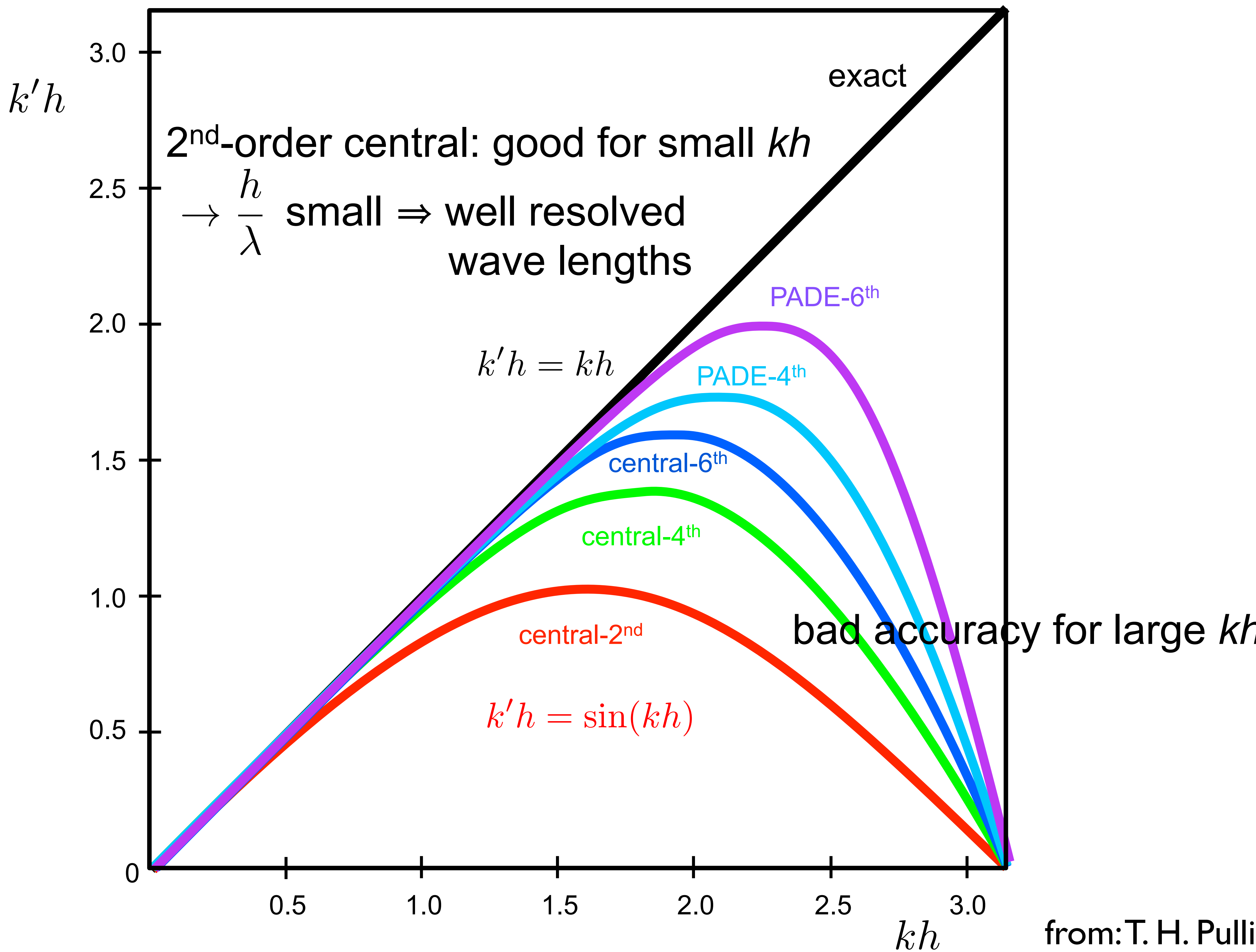
$$k' = \frac{\sin\left(\frac{2\pi n}{N}\right)}{h} = \frac{\sin\left(\frac{2\pi n}{L}\frac{L}{N}\right)}{h} = \frac{\sin(kh)}{h}$$

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

recall: exact derivative

$$f'(x) = ikf(x)$$

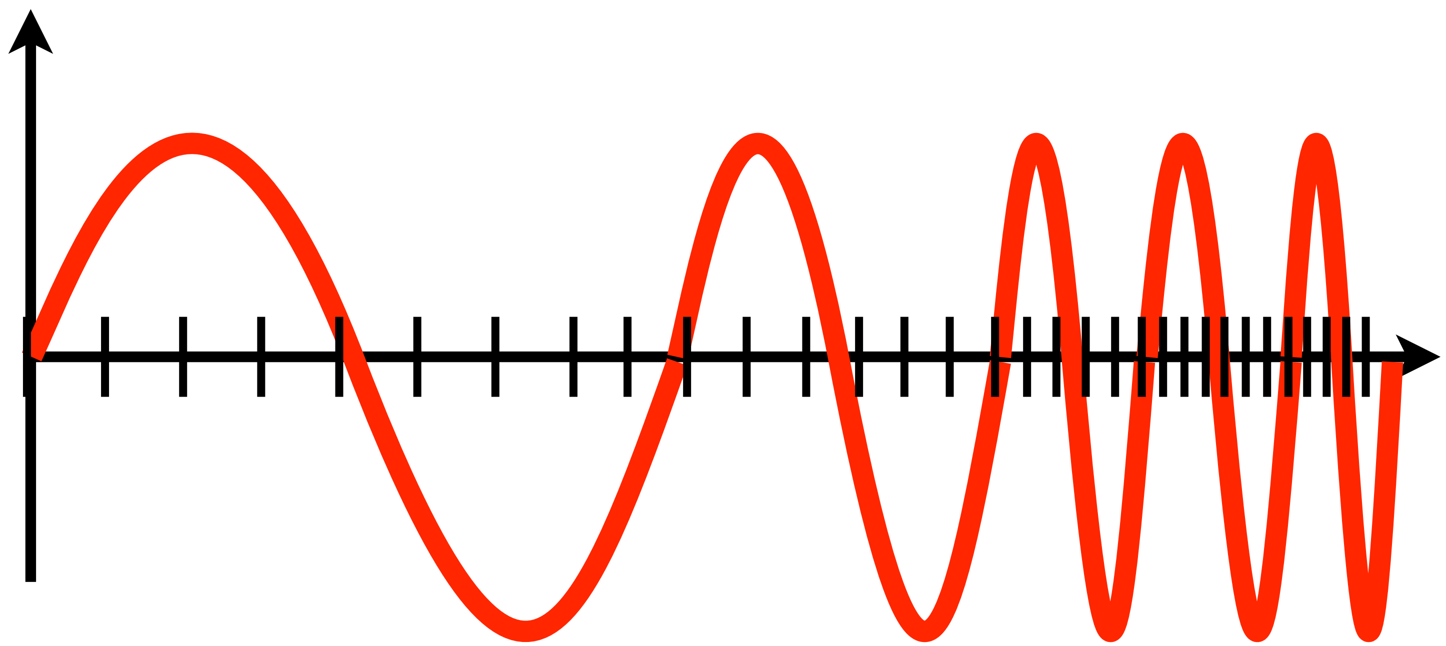
- Plot k'h vs kh: Modified wave number plot



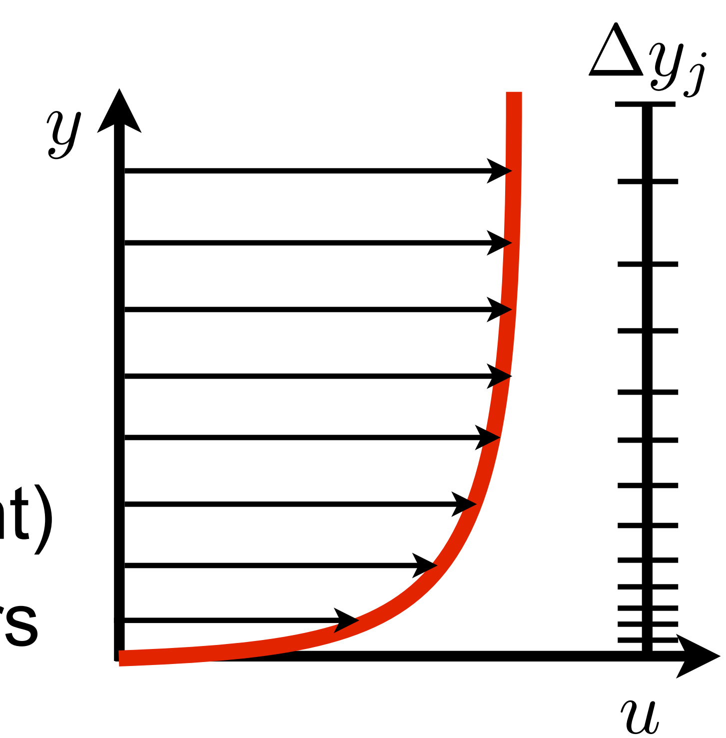
from: T. H. Pulliam

Non-Uniform Grids

- Example Scenario:

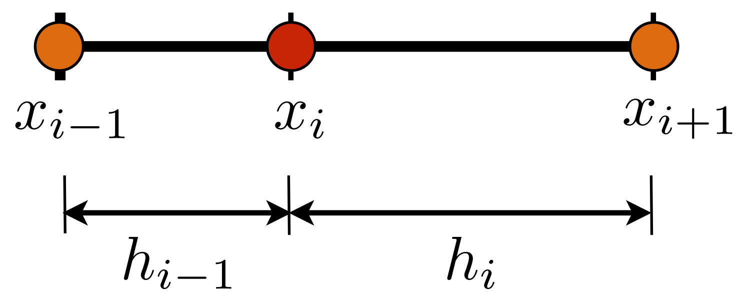


- from modified wave number analysis we know we would like kh to be small
- but not too small for efficiency reasons! (as $h \downarrow \Rightarrow N \uparrow$)
- Strategy: make mesh spacing non-uniform!
  - use large h where k is small
  - use small h where k is large
- Do we know *a-priori* where k will be small or large?
  - often times not $\Rightarrow$ 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:



Taylor Table $f_i'' + a_0 f_i + a_1 f_{i+1} + a_2 f_{i-1} = O(?)$

$$f_{i+1} = f_i + h_i f_i' + \frac{1}{2} h_i^2 f'' + \frac{1}{6} h_i^3 f''' + \dots$$

$$f_{i-1} = f_i - h_{i-1} f_i' + \frac{1}{2} h_{i-1}^2 f'' - \frac{1}{6} h_{i-1}^3 f''' + \dots$$

	f_i	f_i'	f''	f'''
f_i''	0	0	1	0
$a_0 f_i$	a_0	0	0	0
$a_1 f_{i+1}$	a_1	$a_1 h_i$	$\frac{a_1}{2} h_i^2$	$\frac{a_1}{6} h_i^3$
$a_2 f_{i-1}$	a_2	$-a_2 h_{i-1}$	$\frac{a_2}{2} h_{i-1}^2$	$-\frac{a_2}{6} h_{i-1}^3$
	$=0$	$=0$	$=0$	= leading order error

$$a_0 + a_1 + a_2 = 0$$

$$a_1 h_i - a_2 h_{i-1} = 0$$

$$1 + \frac{a_1}{2} h_i^2 - \frac{a_2}{2} h_{i-1}^2 = 0$$

Solve:

$$a_0 = \frac{2}{h_{i-1} h_i}$$

$$a_1 = -\frac{2}{h_i (h_{i-1} + h_i)}$$

$$a_2 = -\frac{2}{h_{i-1} (h_{i-1} + h_i)}$$

Substitute in:

$$f_i'' = \frac{2}{h_i (h_{i-1} + h_i)} f_{i+1} - \frac{2}{h_i h_{i-1}} f_i + \frac{2}{h_{i-1} (h_{i-1} + h_i)} f_{i-1} + \frac{1}{3} (h_{i-1} - h_i) f_i''' + \dots$$

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:

$$f_i'' = \frac{2}{h_i(h_{i-1}+h_i)} f_{i+1} - \frac{2}{h_i h_{i-1}} f_i \quad + \frac{2}{h_{i-1}(h_{i-1}+h_i)} f_{i-1} + \frac{1}{3}(h_{i-1}-h_i) f_i''' + \ldots$$

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

  - this is only 1st order accurate!

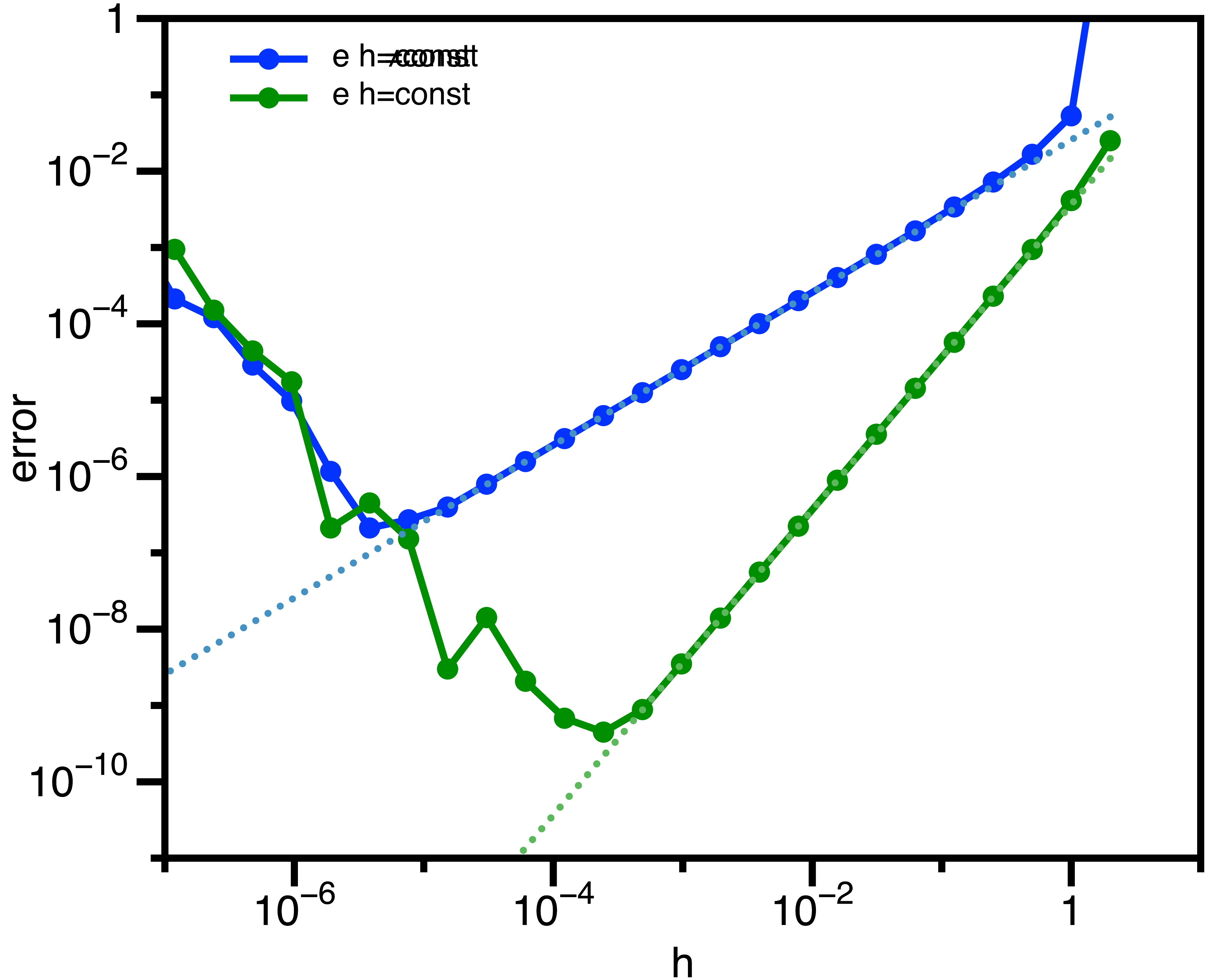
What if h is constant? $h_{i-1} = h_i = h$

$$f_i'' = \frac{2}{h(h+h)} f_{i+1} - \frac{2}{h \cdot h} f_i \quad + \frac{2}{h(h+h)} f_{i-1} + \frac{1}{3}(h-h) f_i''' + \ldots$$

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

  - uniform mesh formula for f” is second order!

Example: $f(x) = \dfrac{1 + x\sin x}{x^3}$ $\qquad f''(x = 3.9) \qquad h = h_i = \dfrac{1}{2}h_{i-1}$



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

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

- How does one decide where to refine the mesh?

Two methods:

Method A (from modified wave number analysis):

1. do Fourier transform on local patches
2. if amplitude of highest wave number > threshold ⇒ refine patch

Method B (from Taylor series analysis)

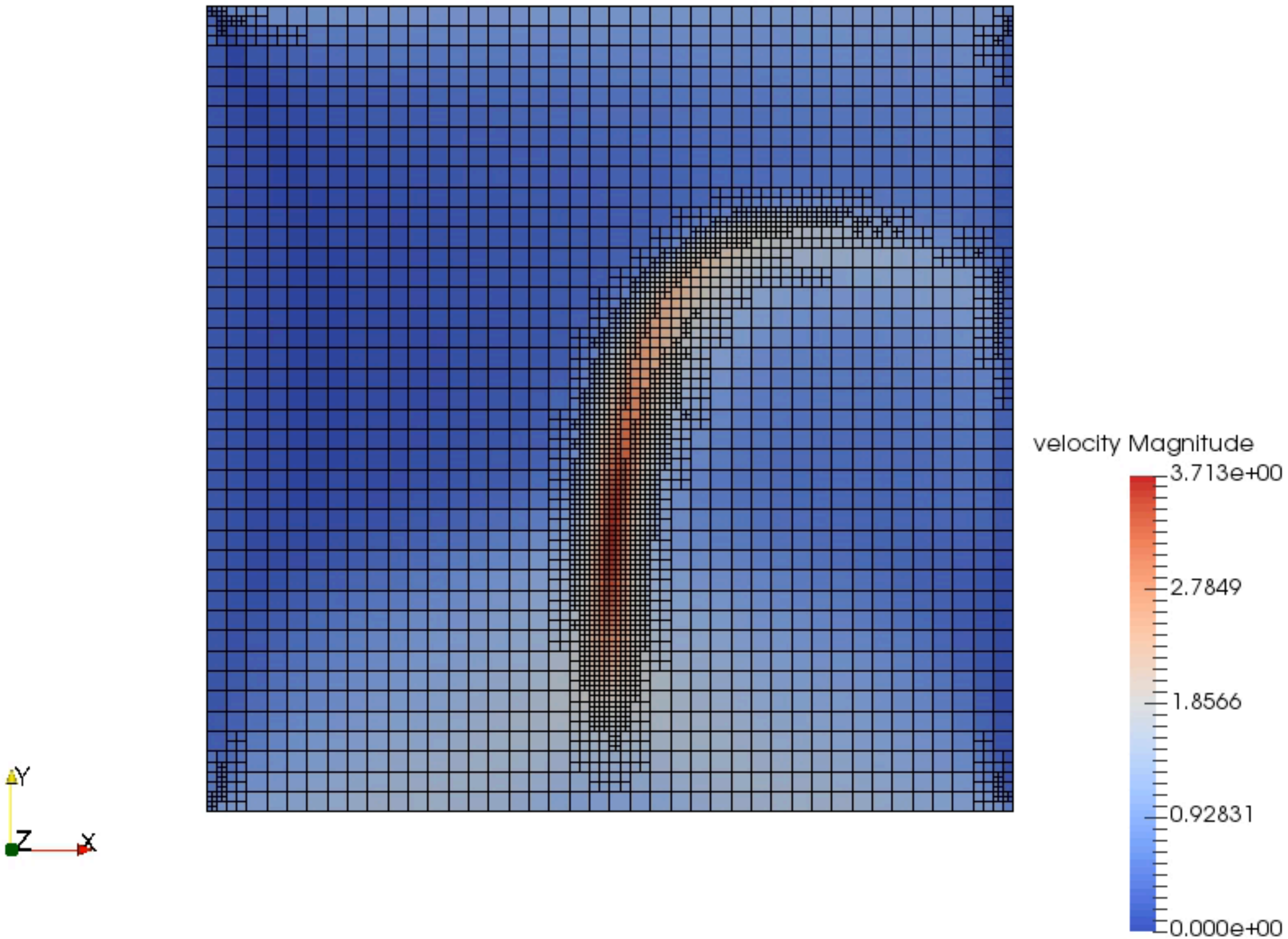
1. estimate leading order error term
2. refine mesh, where estimated error > threshold

Example:

$$f_i'' = \frac{f_i - 2f_{i-1} + f_{i-2}}{h^2} + hf_i''' + \ldots$$

1. estimate leading order error term using finite differences for f_i''': $e_i \approx h_i f_i'''$
2. change h to $h_i \approx \frac{e_{OK}}{f_i'''}$

Method B example (using HW case from later in the semester)

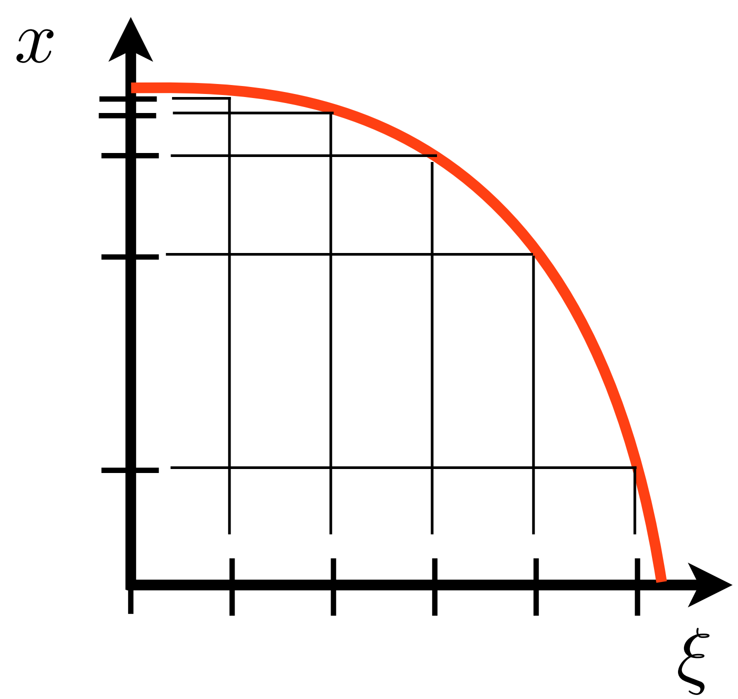


Non-Uniform Grids

- Alternative: **<u>Coordinate transformations</u>**
  - Example:

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

- 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