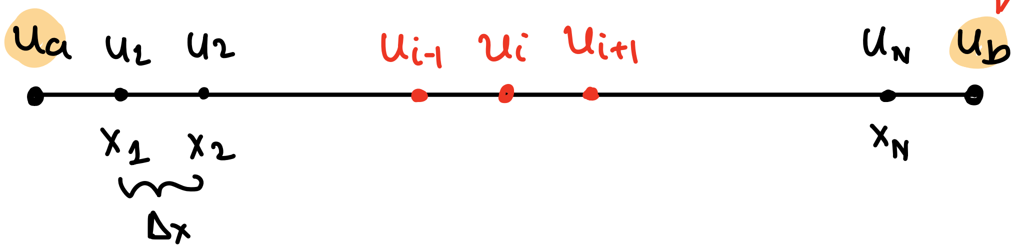
Finite Difference Matrix Operators

Lomax 3.3.2

Consider grid



Boundary condition

Value is known

Appoximation for second derivative at x_i :

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_i \equiv \delta_{xx} u_i = \frac{u_{i-1} - 2u_i + u_{i+1}}{\Delta x^2} + O(\Delta x^2)$$

This is a point difference operator

Point difference operators are not useful for CFD

We need an operator that will take entire grid function and its boundary conditions and result in a derivative approximation, not nessesarily at the nodal points (x_i) of the grid function

we have: $\underbrace{\frac{\partial^?}{\partial x^?} \text{ approximation}}_{\text{all points}} = \text{Differene operator}\left(\underbrace{\text{grid fun}}_{\text{entire grid function}}\right)$

Let's write this in detail:

$$\delta_{xx} u_1 = \frac{1}{\Delta x^2}(u_a - 2u_1 + u_2)$$
$$\delta_{xx} u_2 = \frac{1}{\Delta x^2}(u_1 - 2u_2 + u_3)$$
$$\delta_{xx} u_3 = \frac{1}{\Delta x^2}(u_2 - 2u_3 + u_4)$$
$$\delta_{xx} u_4 = \frac{1}{\Delta x^2}(u_3 - 2u_4 + u_5)$$
$$\vdots$$
$$\delta_{xx} u_N = \frac{1}{\Delta x^2}(u_{N-1} - 2u_N + u_b)$$

That's a lot of equations... also there is a pattern...

we will rearrange...

$$\begin{bmatrix} \delta_{xx} u_1 \\ \delta_{xx} u_2 \\ \delta_{xx} u_3 \\ \delta_{xx} u_4 \\ \vdots \\ \\ \delta_{xx} u_N \end{bmatrix} = \frac{1}{\Delta x^2} \begin{bmatrix} -2 & 1 & 0 & 0 & 0 & & \\ 1 & -2 & 1 & 0 & 0 & & \\ 0 & 1 & -2 & 1 & 0 & & \\ 0 & 0 & 1 & -2 & 1 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & & 1 & -2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ \vdots \\ \\ u_N \end{bmatrix} + \frac{1}{\Delta x^2} \begin{bmatrix} u_a \\ 0 \\ 0 \\ 0 \\ \vdots \\ \\ u_b \end{bmatrix}$$

Banded matrix

Notation: $B(1,-2,1;N)$

vector

BCs go here

$$\delta_{xx} \vec{u} = A\vec{u} + \vec{b}$$

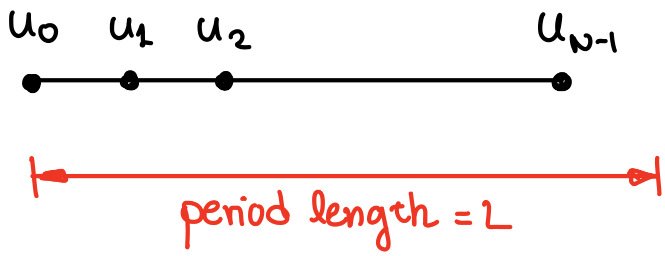
Matrix Difference operator ($N \times N$ matrix)

Matrix A is the discrete second derivative
using our chosen approximation

Do NOT form the matrix and do NOT do a matrix-vector multiplication to estimate $\delta_x \vec{u}$

Matrix for notation and analysis ONLY

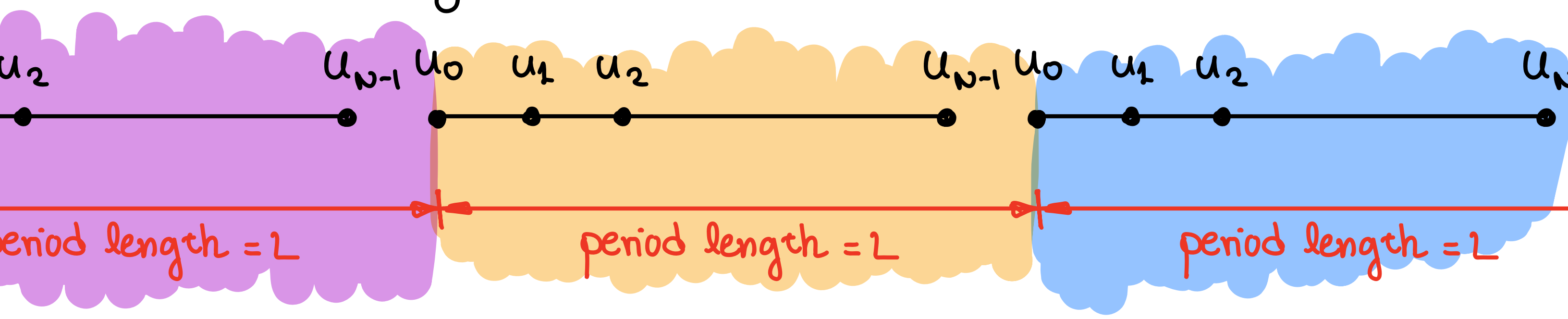
Discrete Periodicity and Boundary Conditions



N points

Grid Spacing: $\Delta x = \frac{L}{N}$

Periodic Boundary Condition means pattern repeats like that:



Spectral Method: method implicitly uses a periodic boundary condition.

Do not try to impose a boundary condition by doing something special at the boundary

Finite Differences: Boundary condition must be imposed

The problem is computing derivatives at the edges of the domain:

e.g. $$\left.\frac{du}{dx}\right|_0 = \frac{u_1 - u_{-1}}{2\Delta x}$$ need this

Option 1:

chage difference at boundaries:

$$\left.\frac{du}{dx}\right|_0 = \frac{u_1 - u_{N-1}}{2\Delta x} \quad \text{and} \quad \left.\frac{du}{dx}\right|_{N-1} = \frac{u_0 - u_{N-2}}{2\Delta x}$$

Option 2: ghost cells

grow the domain left and right using the periodic pattern

Initial array $[u_0 \;\; u_1 \;\; \ldots \;\; u_{N-1}]$

New array $[u_{N-1} \;\; u_0 \;\; u_1 \;\; \ldots \;\; u_{N-1} \;\; u_0]$

ghost cells

number of ghost cells depends on FD stencil!
we need enough ghost cells to be able to apply FD scheme

$[u_{N-1} \;\; u_0 \;\; u_1 \;\; \ldots \;\; u_{N-1} \;\; u_0]$

now derivative at x_0 is $\simeq \dfrac{\overset{u_1}{u_{right}} - \overset{u_{N-1}}{u_{left}}}{2\Delta x}$

Positives of ghost cells: can apply same FD scheme all the way to the boundary

e.g. $\dfrac{u_{i+1} - u_{i-1}}{2\Delta x}$

Negatives of ghost cells: must track indexes since first array index is no longer the left boundary

Circulat Matrices

Consider: • a periodic grid with N points
• second-order approximation of first derivative

$$\delta_x u = \frac{u_{i+1} - u_{i-1}}{2\Delta x}$$

Difference matrix : $\frac{1}{2\Delta x}\begin{bmatrix} 0 & 1 & 0 & & & -1 \\ -1 & 0 & 1 & & \text{zeros} & \\ 0 & -1 & 0 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & \text{zeroes} & & & & \\ 1 & & & & -1 & 0 \end{bmatrix}$

Notation : $A = \frac{1}{2\Delta x} B_p(-1,0,1;N)$

periodic

Important properties:

- $\delta_x u \rightarrow B_p(-1,0,1;N) \longrightarrow$ Pure imaginary eigenvalues
- $\delta_{xx} u \rightarrow B(1,-2,1;N) \longrightarrow$ Pure real eigenvalues

Analytical expressions in Appendix B of Lomax

General FD approximation:

$$\underbrace{\sum_{j=-r}^{s} b_j \frac{\partial^m u}{\partial x^m}\bigg|_{i+j}}_{r+s+1 \text{ points}} - \underbrace{\sum_{j=-p}^{q} a_j u_{i+j}}_{p+q+1 \text{ points}} = \text{error}$$

Implicit | Explicit

Hermitian

Padé

Compact

Stencil width = max(s, q) + max(r, p) + 1

Example: Implicit finite difference

$$\frac{\partial u}{\partial x}\bigg|_{i-1} + 4\frac{\partial u}{\partial x}\bigg|_{i} + \frac{\partial u}{\partial x}\bigg|_{i+1} - \frac{3}{\Delta x}\left(u_{i+1} - u_{i-1}\right) = \frac{\Delta x^4}{120}\frac{\partial^5 u}{\partial x^2} + \cdots$$

Order = 4 stencil width = 3

Matrix Operator: $\frac{1}{6} B(1,4,1)\,\delta_x \vec{u} = \frac{1}{2\Delta x} B(-1,0,1)\,\vec{u} + \vec{b}$

$$\delta_x \vec{u} = 6\,B(1,4,1)^{-1}\left[\frac{1}{2\Delta x} B(-1,0,1)\,\vec{u} + \vec{b}\right]$$

Do NOT compute or form the matrix inverse

- Inverse is a full matrix
- Computing the inverse is VERY expensive
- Use a special linear system solver (much cheper)