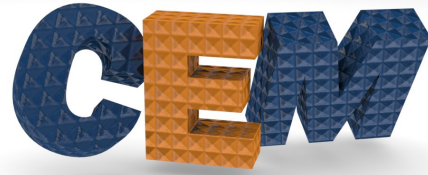


Instructor  
Dr. Raymond Rumpf  
(915) 747-6958  
rrumpf@utep.edu



EE 5337

## Computational Electromagnetics (CEM)

Lecture #10

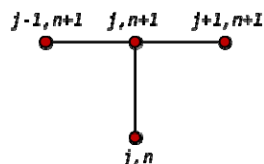
# Finite-Difference Method

◆◆ These notes may contain copyrighted material obtained under fair use rules. Distribution of these materials is strictly prohibited ◆◆

## Outline



- Finite-Difference Approximations
- Finite-Difference Method
- Numerical Boundary Conditions
- Matrix Operators



# Finite Difference Approximations

Lecture 10

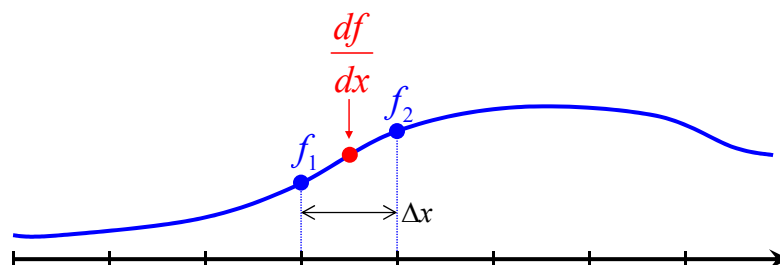
Slide 3

## The Basic Finite-Difference Approximation



$$\frac{df_{1.5}}{dx} \approx \frac{f_2 - f_1}{\Delta x}$$

second-order accurate  
first-order derivative



This is the only finite-difference approximation we will use in this course!

Lecture 10

Slide 4

## Types of Finite-Difference Approximations CEM

<p>Backward Finite-Difference</p>	$\frac{df_2}{dx} \approx \frac{f_2 - f_1}{\Delta x}$	
<p>Central Finite-Difference</p>	$\frac{df_{1.5}}{dx} \approx \frac{f_2 - f_1}{\Delta x}$	
<p>Forward Finite-Difference</p>	$\frac{df_1}{dx} \approx \frac{f_2 - f_1}{\Delta x}$	

Lecture 10Slide 5

## The Generalized Finite-Difference CEM

$$\frac{d^n f}{dx^n} \cong \sum_i a_i f_i$$

$$L\{f\} \cong \sum_i a_i f_i$$

In fact, any linear operation on the function can be approximated as a linear sum of known points of that function.

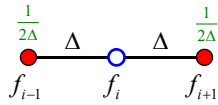
The derivative of any order of a function at any position can be approximated as a linear sum of known points of that function.

Lecture 10Slide 6

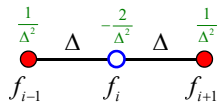
## Finite-Difference “Atoms”



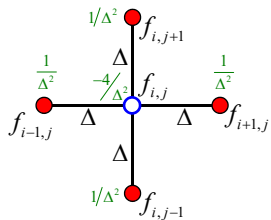
Any finite-difference approximation can be summarized graphically as an “atom.”



$$\frac{d}{dx} f(x_i) \cong \frac{f_{i+1} - f_{i-1}}{2\Delta}$$



$$\frac{d^2}{dx^2} f(x_i) \cong \frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta^2}$$



$$\nabla^2 f(x_i) \cong \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{\Delta^2} + \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{\Delta^2}$$

Lecture 10

Slide 7

# Finite-Difference Method

Lecture 10

Slide 8

## Overview of Our FDM



1. Identify and write the governing equation(s).

$$a(x)\frac{\partial^2}{\partial x^2}f(x) + \gamma b(x)\frac{\partial}{\partial x}f(x) + c(x)f(x) = g(x)$$

2. Write the matrix form of this equation going term-by-term.

$$a(x)\frac{\partial^2}{\partial x^2}f(x) + \gamma b(x)\frac{\partial}{\partial x}f(x) + c(x)f(x) = g(x)$$

$$\begin{array}{cccccccccccc} \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ [A][D_x^2][f] + \gamma[B][D_x][f] + [C][f] = [g] \end{array}$$

3. Put matrix equation in the standard form of  $[L][f] = [g]$ .

$$[L][f] = [g] \quad \text{where} \quad [L] = [A][D_x^2] + \gamma[B][D_x] + [C]$$

$$L = A \cdot D2X + \gamma \cdot B \cdot DX + C;$$

4. Solve  $[L][f] = [g]$ .

$$[f] = [L]^{-1}[g]$$

Lecture 10

Slide 9

## What's the Catch?



$$[L] = [A][D_x^2] + \gamma[B][D_x] + [C]$$

How do we construct these matrices? What do they mean?

$$\gamma \begin{array}{c} [A] \\ [B] \\ [C] \end{array} \begin{array}{c} [D_x] \\ [D_x^2] \end{array}$$

Lecture 10

Slide 10

## Functions Vs. Operations (1 of 2)



$$a(x)\frac{\partial^2}{\partial x^2}f(x) + \gamma b(x)\frac{\partial}{\partial x}f(x) + c(x)f(x) = g(x)$$

### Operations

Everything else in a differential equation is something that operates on a function.

$a(x), b(x), c(x) \equiv$  point-by-point

multiplication on  $f(x)$

$\frac{\partial}{\partial x}, \frac{\partial^2}{\partial x^2} \equiv$  calculates derivatives of  $f(x)$

$\gamma \equiv$  scales entire  $f(x)$

### Functions

The only time functions appear in a differential equation is as the unknown or as the excitation.

$f(x) \equiv$  unknown

$g(x) \equiv$  excitation

Lecture 10

Slide 11

## Functions Vs. Operations (2 of 2)



$$[A][D_x^2][f] + \gamma[B][D_x][f] + [C][f] = [g]$$

### Operations

Operations are always stored in square matrices. Any linear operation can be put into matrix form.

$$[L] = \begin{bmatrix} l_{11} & l_{12} & & l_{1M} \\ l_{21} & l_{22} & \cdots & l_{2M} \\ & \vdots & & \\ l_{M1} & l_{M2} & & l_{MM} \end{bmatrix}$$

### Functions

Functions are stored as column vectors.

$$[f] = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_M \end{bmatrix} \quad [g] = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_M \end{bmatrix}$$

Lecture 10

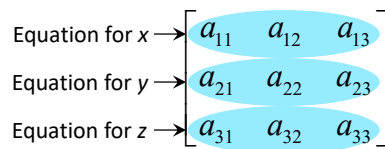
Slide 12

# Interpretation of Matrices

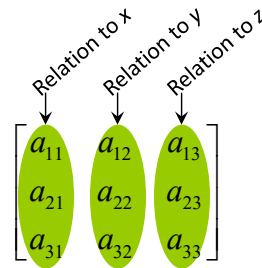


$$\begin{aligned}
 a_{11}x + a_{12}y + a_{13}z &= b_1 \\
 a_{21}x + a_{22}y + a_{23}z &= b_2 \\
 a_{31}x + a_{32}y + a_{33}z &= b_3
 \end{aligned}
 \Rightarrow
 \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}
 \begin{bmatrix} x \\ y \\ z \end{bmatrix}
 =
 \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

EQUATION FOR...



RELATION TO...



From a purely mathematical perspective, this interpretation does not make sense. This interpretation will be highly useful and insightful because of how we derive the equations.

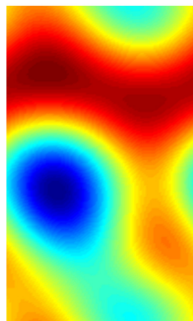
Lecture 10

Slide 13

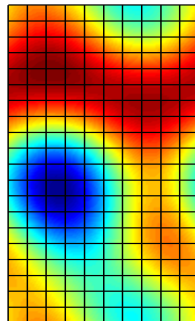
# Representing Functions on a Grid



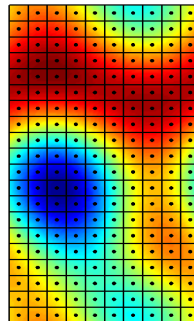
Example physical (continuous) 2D function



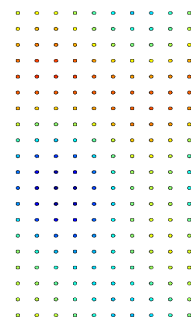
A grid is constructed by dividing space into discrete cells



Function is known only at discrete points




Representation of what is actually stored in memory



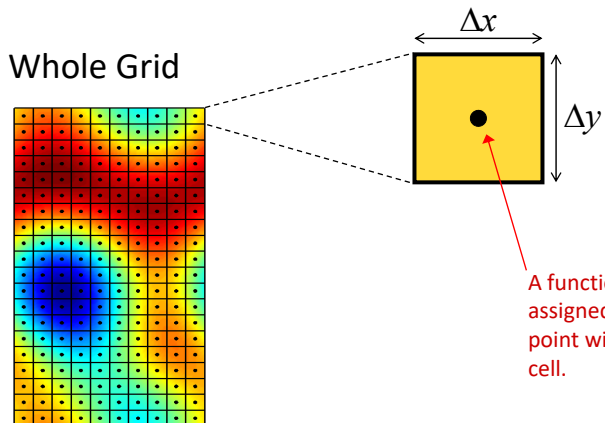
Lecture 10

Slide 14

## Grid Cells



### A Single Grid Cell




A function value is assigned to a specific point within the grid cell.

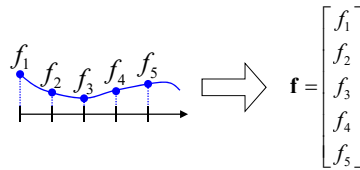
$\Delta x, \Delta y \equiv$  grid resolution parameters

Lecture 10
Slide 15

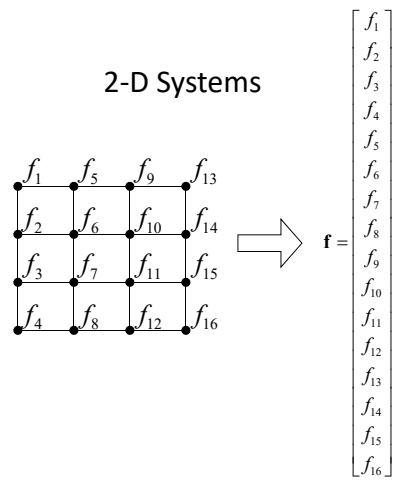
## Functions are Put Into Column Vectors



### 1-D Systems




### 2-D Systems

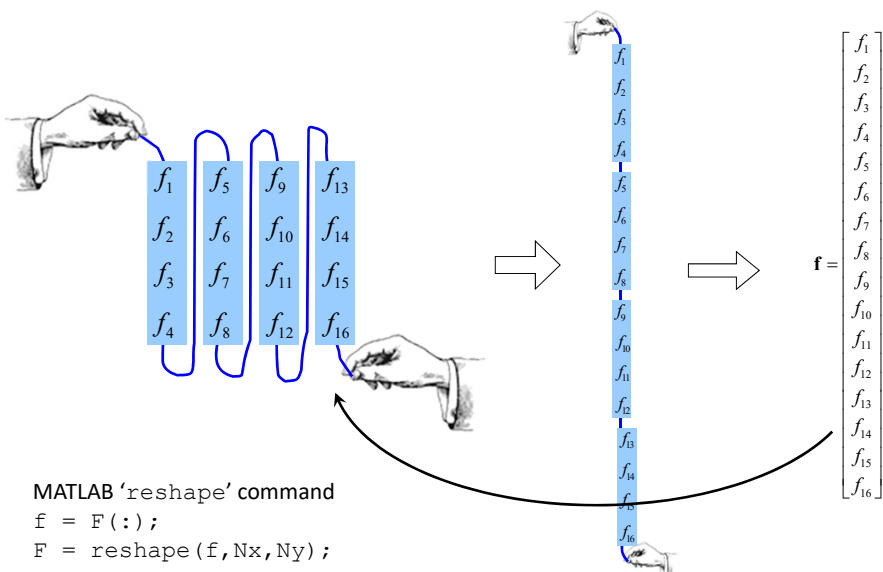


Lecture 10
Slide 16



## Putting Functions into Column Vectors






MATLAB 'reshape' command

```
f = F(:);
F = reshape(f, Nx, Ny);
```

Lecture 10 Slide 17

## Locating Nodes in Column Vectors



1D Grids

Node located at  $n_x \rightarrow m = n_x$

2D Grids

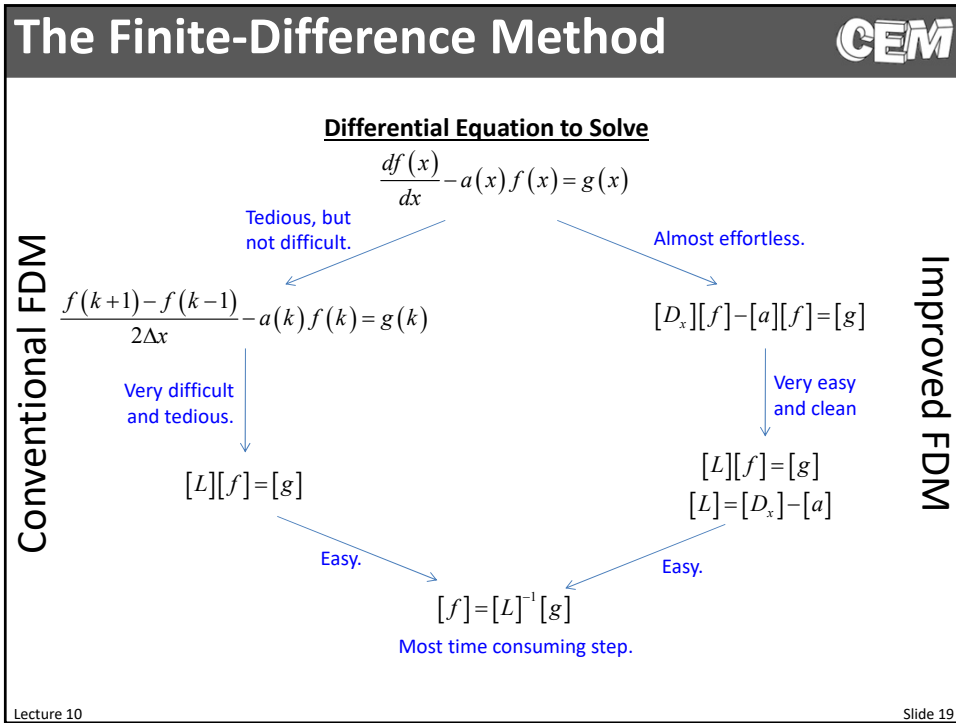
Node located at  $n_x, n_y \rightarrow m = (n_y - 1) * N_x + n_x$

3D Grids

Node located at  $n_x, n_y, n_z$

$$\rightarrow m = (n_z - 1) * N_x * N_y + (n_y - 1) * N_x + n_x$$

Lecture 10 Slide 18



## Conventional FDM (1 of 3) CEM

Step 1 – We start with a differential equation that we wish to solve.

$$\frac{d^2 f}{dx^2} - a \frac{df}{dx} - bf = c$$

Step 2 – We approximate the derivatives with finite differences.

$$\frac{f(k+1) - 2f(k) + f(k-1)}{\Delta^2} - a(k) \frac{f(k+1) - f(k-1)}{2\Delta} - b(k)f(k) = c(k)$$

**IMPORTANT RULE: Every term in a finite-difference equation must exist at the same point.**

Step 3 – The equation is expanded and we collect common terms

$$\frac{1}{\Delta^2} f(k+1) - \frac{2}{\Delta^2} f(k) + \frac{1}{\Delta^2} f(k-1) - \frac{a(k)}{2\Delta} f(k+1) + \frac{a(k)}{2\Delta} f(k-1) - b(k)f(k) = c(k)$$

$$\left[ \frac{1}{\Delta^2} - \frac{a(k)}{2\Delta} \right] f(k+1) - \left[ b(k) + \frac{2}{\Delta^2} \right] f(k) + \left[ \frac{1}{\Delta^2} + \frac{a(k)}{2\Delta} \right] f(k-1) = c(k)$$

Lecture 10 Slide 20



## Improved FDM



We want a very easy way to immediately write differential equations in matrix form. Starting with the same differential equation...

$$\frac{\partial^2 f}{\partial x^2} - a \frac{\partial f}{\partial x} - bf = c$$

We will develop a procedure by which this will be directly written in matrix form without having to explicitly handle any finite-differences.

$$\frac{\partial^2}{\partial x^2} f - a \frac{\partial}{\partial x} f - bf = c$$

$\mathbf{D}_x^{(2)}$ ,  $\mathbf{A}$ ,  $\mathbf{D}_x^{(1)}$ , and  $\mathbf{B}$  are square matrices that perform linear operations on the vector  $\mathbf{f}$ .

$$\mathbf{D}_x^{(2)} \mathbf{f} - \mathbf{A} \mathbf{D}_x^{(1)} \mathbf{f} - \mathbf{B} \mathbf{f} = \mathbf{c} \quad \rightarrow \quad \underbrace{\left[ \mathbf{D}_x^{(2)} - \mathbf{A} \mathbf{D}_x^{(1)} - \mathbf{B} \right]}_{\mathbf{L}=[L]} \mathbf{f} = \mathbf{c}$$

$$\mathbf{f} = \mathbf{L}^{-1} \mathbf{c}$$

Lecture 10

Slide 23

## Locating Nodes in Matrices



$$1\text{D} \rightarrow m \text{ or } n = nx$$

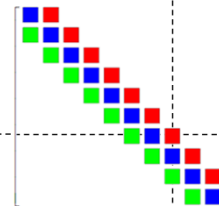
$$2\text{D} \rightarrow m \text{ or } n = (ny-1) * Nx + nx$$

$$3\text{D} \rightarrow m \text{ or } n = (nz-1) * Nx * Ny + (ny-1) * Nx + nx$$

Column  $n$   
Relation to node  $(nx, ny, nz)$

Row  $m$

Equation for node  $(nx, ny, nz)$



Lecture 10

Slide 24

# Numerical Boundary Conditions

Lecture 10

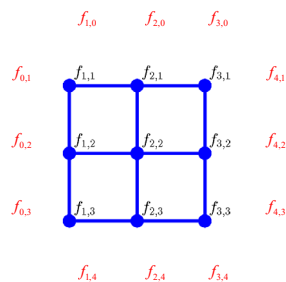
Slide 25

## The Problem at the Boundaries



Suppose it is desired to solve the following differential equation on a  $3 \times 3$  grid.

$$\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} = g(x,y) \quad \rightarrow \quad \frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{\Delta x^2} + \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j-1}}{\Delta y^2} = g_{i,j}$$



The terms in **red** exist outside of the grid.  
How this is handled is called a *boundary condition*.

$$\begin{aligned} \frac{f_{2,1} - 2f_{1,1} + f_{0,1}}{\Delta x^2} + \frac{f_{1,2} - 2f_{1,1} + f_{1,0}}{\Delta y^2} &= g_{1,1} \\ \frac{f_{3,1} - 2f_{2,1} + f_{1,1}}{\Delta x^2} + \frac{f_{2,2} - 2f_{2,1} + f_{2,0}}{\Delta y^2} &= g_{2,1} \\ \frac{f_{4,1} - 2f_{3,1} + f_{2,1}}{\Delta x^2} + \frac{f_{3,2} - 2f_{3,1} + f_{3,0}}{\Delta y^2} &= g_{3,1} \\ \frac{f_{2,2} - 2f_{1,2} + f_{0,2}}{\Delta x^2} + \frac{f_{1,3} - 2f_{1,2} + f_{1,1}}{\Delta y^2} &= g_{1,2} \\ \frac{f_{3,2} - 2f_{2,2} + f_{1,2}}{\Delta x^2} + \frac{f_{2,3} - 2f_{2,2} + f_{2,1}}{\Delta y^2} &= g_{2,2} \\ \frac{f_{4,2} - 2f_{3,2} + f_{2,2}}{\Delta x^2} + \frac{f_{3,3} - 2f_{3,2} + f_{3,1}}{\Delta y^2} &= g_{3,2} \\ \frac{f_{2,3} - 2f_{1,3} + f_{0,3}}{\Delta x^2} + \frac{f_{1,4} - 2f_{1,3} + f_{1,2}}{\Delta y^2} &= g_{1,3} \\ \frac{f_{3,3} - 2f_{2,3} + f_{1,3}}{\Delta x^2} + \frac{f_{2,4} - 2f_{2,3} + f_{2,2}}{\Delta y^2} &= g_{2,3} \\ \frac{f_{4,3} - 2f_{3,3} + f_{2,3}}{\Delta x^2} + \frac{f_{3,4} - 2f_{3,3} + f_{3,2}}{\Delta y^2} &= g_{3,3} \end{aligned}$$

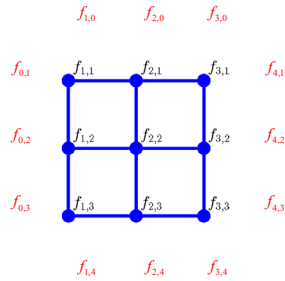
Lecture 10

Slide 26

## Dirichlet Boundary Conditions (1 of 2)



The simplest boundary condition is to assume that all values of  $f(x,y)$  outside of the grid are zero.



$$\begin{aligned} \frac{f_{2,1}-2f_{1,1}+0}{\Delta x^2} + \frac{f_{1,2}-2f_{1,1}+0}{\Delta y^2} &= g_{1,1} \\ \frac{f_{3,1}-2f_{2,1}+f_{1,1}}{\Delta x^2} + \frac{f_{2,2}-2f_{2,1}+0}{\Delta y^2} &= g_{2,1} \\ \frac{0-2f_{3,1}+f_{2,1}}{\Delta x^2} + \frac{f_{3,2}-2f_{3,1}+0}{\Delta y^2} &= g_{3,1} \\ \frac{f_{2,2}-2f_{1,2}+0}{\Delta x^2} + \frac{f_{1,3}-2f_{1,2}+f_{1,1}}{\Delta y^2} &= g_{1,2} \\ \frac{f_{3,2}-2f_{2,2}+f_{1,2}}{\Delta x^2} + \frac{f_{2,3}-2f_{2,2}+f_{2,1}}{\Delta y^2} &= g_{2,2} \\ \frac{0-2f_{3,2}+f_{2,2}}{\Delta x^2} + \frac{f_{3,3}-2f_{3,2}+f_{3,1}}{\Delta y^2} &= g_{3,2} \\ \frac{f_{2,3}-2f_{1,3}+0}{\Delta x^2} + \frac{0-2f_{1,3}+f_{1,2}}{\Delta y^2} &= g_{1,3} \\ \frac{f_{3,3}-2f_{2,3}+f_{1,3}}{\Delta x^2} + \frac{0-2f_{2,3}+f_{2,2}}{\Delta y^2} &= g_{2,3} \\ \frac{0-2f_{3,3}+f_{2,3}}{\Delta x^2} + \frac{0-2f_{3,3}+f_{3,2}}{\Delta y^2} &= g_{3,3} \end{aligned}$$

Lecture 10

Slide 27

## Dirichlet Boundary Conditions (2 of 2)



Dirichlet boundary conditions assume function values from outside of the grid are zero.

$$\begin{aligned} \frac{\partial f(x_1, y_i)}{\partial x} &\cong \frac{f(x_2, y_i) - 0}{2\Delta_x} \\ \frac{\partial f(x_{N_x}, y_i)}{\partial x} &\cong \frac{0 - f(x_{N_x-1}, y_i)}{2\Delta_x} \\ \frac{\partial f(x_i, y_1)}{\partial y} &\cong \frac{f(x_i, y_2) - 0}{2\Delta_y} \\ \frac{\partial f(x_i, y_{N_y})}{\partial y} &\cong \frac{0 - f(x_i, y_{N_y-1})}{2\Delta_y} \\ \frac{\partial^2 f(x_1, y_i)}{\partial x^2} &\cong \frac{f(x_2, y_i) - 2f(x_1, y_i) + 0}{\Delta_x^2} \\ \frac{\partial^2 f(x_{N_x}, y_i)}{\partial x^2} &\cong \frac{0 - 2f(x_{N_x}, y_i) + f(x_{N_x-1}, y_i)}{\Delta_x^2} \\ \frac{\partial^2 f(x_i, y_1)}{\partial y^2} &\cong \frac{f(x_i, y_2) - 2f(x_i, y_1) + 0}{\Delta_y^2} \\ \frac{\partial^2 f(x_i, y_{N_y})}{\partial y^2} &\cong \frac{0 - 2f(x_i, y_{N_y}) + f(x_i, y_{N_y-1})}{\Delta_y^2} \end{aligned}$$

Lecture 10

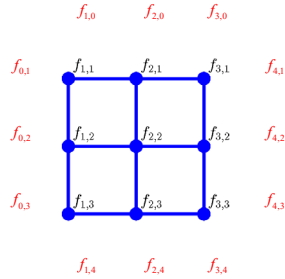
Slide 28

## Periodic Boundary Conditions (1 of 2)



If the function  $f(x,y)$  is periodic, then the values from outside of the grid can be mapped to a value from inside the grid at the other side.

$$f_{0,j} = f_{3,j}, f_{4,j} = f_{1,j}, f_{i,0} = f_{i,3}, \text{ and } f_{i,4} = f_{i,1}$$



$$\frac{f_{2,1} - 2f_{1,1} + f_{3,1}}{\Delta x^2} + \frac{f_{1,2} - 2f_{1,1} + f_{1,3}}{\Delta y^2} = g_{1,1}$$

$$\frac{f_{3,1} - 2f_{2,1} + f_{1,1}}{\Delta x^2} + \frac{f_{2,2} - 2f_{2,1} + f_{2,3}}{\Delta y^2} = g_{2,1}$$

$$\frac{f_{1,1} - 2f_{1,1} + f_{2,1}}{\Delta x^2} + \frac{f_{3,2} - 2f_{3,1} + f_{3,3}}{\Delta y^2} = g_{3,1}$$

$$\frac{f_{2,2} - 2f_{1,2} + f_{3,2}}{\Delta x^2} + \frac{f_{1,3} - 2f_{1,2} + f_{1,1}}{\Delta y^2} = g_{1,2}$$

$$\frac{f_{3,2} - 2f_{2,2} + f_{1,2}}{\Delta x^2} + \frac{f_{2,3} - 2f_{2,2} + f_{2,1}}{\Delta y^2} = g_{2,2}$$

$$\frac{f_{1,2} - 2f_{1,2} + f_{2,2}}{\Delta x^2} + \frac{f_{3,3} - 2f_{3,2} + f_{3,1}}{\Delta y^2} = g_{3,2}$$

$$\frac{f_{2,3} - 2f_{1,3} + f_{3,3}}{\Delta x^2} + \frac{f_{1,1} - 2f_{1,3} + f_{1,2}}{\Delta y^2} = g_{1,3}$$

$$\frac{f_{3,3} - 2f_{2,3} + f_{1,3}}{\Delta x^2} + \frac{f_{2,1} - 2f_{2,3} + f_{2,2}}{\Delta y^2} = g_{2,3}$$

$$\frac{f_{1,3} - 2f_{1,3} + f_{2,3}}{\Delta x^2} + \frac{f_{3,1} - 2f_{3,3} + f_{3,2}}{\Delta y^2} = g_{3,3}$$

Lecture 10

Slide 29

## Periodic Boundary Conditions (2 of 2)



Periodic boundary conditions assume function values from outside of the grid can be taken from the opposite side of the grid.

$$\frac{\partial f(x_1, y_1)}{\partial x} \equiv \frac{f(x_2, y_1) - f(x_{N_x}, y_1)}{2\Delta_x}$$

$$\frac{\partial f(x_{N_x}, y_1)}{\partial x} \equiv \frac{f(x_1, y_1) - f(x_{N_x-1}, y_1)}{2\Delta_x}$$

$$\frac{\partial^2 f(x_1, y_1)}{\partial x^2} \equiv \frac{f(x_2, y_1) - 2f(x_1, y_1) + f(x_{N_x}, y_1)}{\Delta_x^2}$$

$$\frac{\partial^2 f(x_{N_x}, y_1)}{\partial x^2} \equiv \frac{f(x_1, y_1) - 2f(x_{N_x}, y_1) + f(x_{N_x-1}, y_1)}{\Delta_x^2}$$

$$\frac{\partial f(x_i, y_1)}{\partial y} \equiv \frac{f(x_i, y_2) - f(x_i, y_{N_y})}{2\Delta_y}$$

$$\frac{\partial f(x_i, y_{N_y})}{\partial y} \equiv \frac{f(x_i, y_1) - f(x_i, y_{N_y-1})}{2\Delta_y}$$

$$\frac{\partial^2 f(x_i, y_1)}{\partial y^2} \equiv \frac{f(x_i, y_2) - 2f(x_i, y_1) + f(x_i, y_{N_y})}{\Delta_y^2}$$

$$\frac{\partial^2 f(x_i, y_{N_y})}{\partial y^2} \equiv \frac{f(x_i, y_1) - 2f(x_i, y_{N_y}) + f(x_i, y_{N_y-1})}{\Delta_y^2}$$

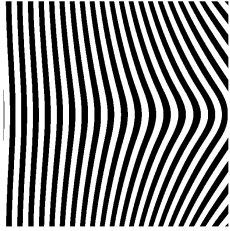
Lecture 10

Slide 30

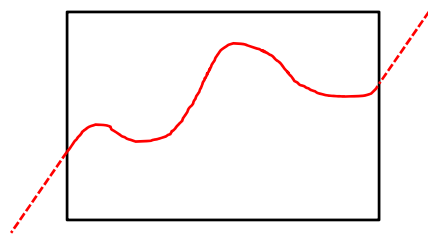
## Neumann Boundary Conditions



We use Neumann boundary conditions for non-periodic functions or functions that are not zero at the boundary. Here the function continues linearly outside of the grid.



Spatially variant grating that is not periodic and not zero at the boundaries.



Here is a 1D function with Neumann boundaries

Lecture 10

Slide 31

## Neumann BC's for 1D Function



The finite-difference approximation for a 1D function is

$$\left. \frac{df}{dx} \right|_i \cong \frac{f_{i+1} - f_{i-1}}{2\Delta_x}$$

At  $i=1$ , we have a problem...

$$\left. \frac{df}{dx} \right|_{i=1} \cong \frac{f_2 - \boxed{f_0}}{2\Delta_x} \quad \text{This term doesn't exist!} \quad \longrightarrow \quad \left. \frac{df}{dx} \right|_{i=1} \cong \frac{f_2 - f_1}{\Delta_x}$$

At  $i=N_x$ , we have another problem...

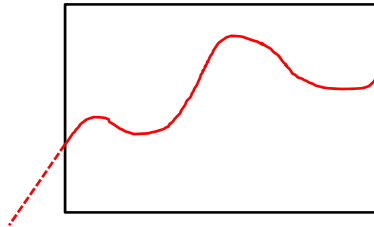
$$\left. \frac{df}{dx} \right|_{i=N_x} \cong \frac{\boxed{f_{N_x+1}} - f_{N_x-1}}{2\Delta_x} \quad \text{This term doesn't exist!} \quad \longrightarrow \quad \left. \frac{df}{dx} \right|_{i=N_x} \cong \frac{f_{N_x} - f_{N_x-1}}{\Delta_x}$$

Lecture 10

Slide 32



## What About the 2nd-Order Derivatives for the Neumann Boundary Condition?



In order for the function to continue in a straight line, the second-order derivative should be set to zero at the boundary.

IMPORTANT: This is NOT Dirichlet boundary conditions. A Dirichlet BC sets the function itself to zero outside of the grid, not the derivative. Here, the 2<sup>nd</sup>-order derivative is set to zero.

$$\left. \frac{d^2 f}{dx^2} \right|_{i=1} \cong \frac{f_2 - f_1 + f_0}{\Delta_x^2} \rightarrow \left. \frac{d^2 f}{dx^2} \right|_{i=1} \cong 0$$

$$\left. \frac{d^2 f}{dx^2} \right|_{i=N_x} \cong \frac{f_{N_x+1} - f_{N_x} + f_{N_x-1}}{\Delta_x^2} \rightarrow \left. \frac{d^2 f}{dx^2} \right|_{i=N_x} \cong 0$$

Lecture 10

Slide 33

## Neumann Boundary Conditions for 2D Functions



Neumann boundary conditions are used when a function should be continuous at the boundary. That is, the first-order derivative is continuous and the second-order derivative is zero.

$$\frac{\partial f(x_1, y_1)}{\partial x} \cong \frac{f(x_2, y_1) - f(x_1, y_1)}{\Delta_x} \qquad \frac{\partial^2 f(x_1, y_1)}{\partial x^2} = 0$$

$$\frac{\partial f(x_{N_x}, y_1)}{\partial x} \cong \frac{f(x_{N_x}, y_1) - f(x_{N_x-1}, y_1)}{\Delta_x} \qquad \frac{\partial^2 f(x_{N_x}, y_1)}{\partial x^2} = 0$$

$$\frac{\partial f(x_i, y_1)}{\partial y} \cong \frac{f(x_i, y_2) - f(x_i, y_1)}{\Delta_y} \qquad \frac{\partial^2 f(x_i, y_1)}{\partial y^2} = 0$$

$$\frac{\partial f(x_i, y_{N_y})}{\partial y} \cong \frac{f(x_i, y_{N_y}) - f(x_i, y_{N_y-1})}{\Delta_y} \qquad \frac{\partial^2 f(x_i, y_{N_y})}{\partial y^2} = 0$$

Lecture 10

Slide 34

# Matrix Operators

Lecture 10

Slide 35

## Origin of Matrix Operators

We start with a governing equation.

We approximate the governing equation with finite-differences and then write the finite-difference equation at each point the grid.

We collect the large set of equations into a single matrix equation.

We construct a grid to store the functions.

We collect the large set of equations into a single matrix equation.

$$\frac{1}{\Delta x^2} \begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix} \begin{bmatrix} f_{1,1} \\ f_{2,1} \\ f_{3,1} \\ f_{1,2} \\ f_{2,2} \\ f_{3,2} \\ f_{1,3} \\ f_{2,3} \\ f_{3,3} \end{bmatrix} = \begin{bmatrix} g_{1,1} \\ g_{2,1} \\ g_{3,1} \\ g_{1,2} \\ g_{2,2} \\ g_{3,2} \\ g_{1,3} \\ g_{2,3} \\ g_{3,3} \end{bmatrix}$$

This matrix calculates the derivative of  $f(x,y)$  and puts the answer in  $g(x,y)$ . This is a matrix operator.

We start with a governing equation.

$$\frac{d^2 f(x,y)}{dx^2} = g(x,y)$$

We approximate the governing equation with finite-differences and then write the finite-difference equation at each point the grid.

$$\begin{aligned} \frac{f_{2,1} - 2f_{1,1} + 0}{\Delta x^2} &= g_{1,1} \\ \frac{f_{3,1} - 2f_{2,1} + f_{1,1}}{\Delta x^2} &= g_{2,1} \\ \frac{0 - 2f_{3,1} + f_{2,1}}{\Delta x^2} &= g_{3,1} \\ \frac{f_{2,2} - 2f_{1,2} + 0}{\Delta x^2} &= g_{1,2} \\ \frac{f_{3,2} - 2f_{2,2} + f_{1,2}}{\Delta x^2} &= g_{2,2} \\ \frac{0 - 2f_{3,2} + f_{2,2}}{\Delta x^2} &= g_{3,2} \\ \frac{f_{2,3} - 2f_{1,3} + 0}{\Delta x^2} &= g_{1,3} \\ \frac{f_{3,3} - 2f_{2,3} + f_{1,3}}{\Delta x^2} &= g_{2,3} \\ \frac{0 - 2f_{3,3} + f_{2,3}}{\Delta x^2} &= g_{3,3} \end{aligned}$$

We construct a grid to store the functions.

This matrix calculates the derivative of  $f(x,y)$  and puts the answer in  $g(x,y)$ . This is a matrix operator.

Lecture 10

Slide 36

## Other Matrix Operators



A square matrix can **always** be constructed to perform **any** linear operation on a function that is stored in a column vector.

$$\frac{d}{dx} f(x) \rightarrow \mathbf{D}_x \mathbf{f} = \underbrace{\left[ \begin{array}{c} ? \\ ? \\ ? \\ \vdots \\ ? \end{array} \right]}_{\mathbf{D}_x} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_N \end{bmatrix}$$

$$\text{FFT}[f(x)] \rightarrow \mathbf{F} \mathbf{f}$$

$$g(x)f(x) \rightarrow \mathbf{G} \mathbf{f}$$

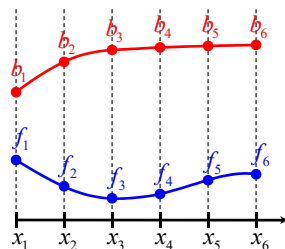
$$\int [f(x)] dx \rightarrow \left[ \int \right] \mathbf{f}$$

$$h(x)*f(x) \rightarrow \mathbf{H} \mathbf{f}$$

Lecture 10

Slide 37

## Point-by-Point Multiplication (1 of 2)



Since we are storing our "functions" in vector form, how do we perform a point-by-point multiplication using a square matrix?

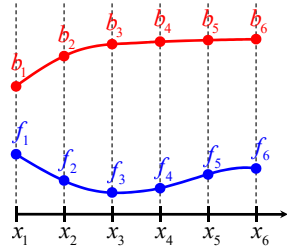
$$b(x)f(x) \rightarrow \mathbf{B} \mathbf{f}$$

$$\underbrace{\left[ \begin{array}{c} ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{array} \right]}_{\mathbf{B}} \underbrace{\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix}}_{\mathbf{f}} = \underbrace{\begin{bmatrix} b_1 f_1 \\ b_2 f_2 \\ b_3 f_3 \\ b_4 f_4 \\ b_5 f_5 \\ b_6 f_6 \end{bmatrix}}_{\mathbf{B} \mathbf{f}}$$

Lecture 10

Slide 38

## Point-by-Point Multiplication (2 of 2) CEM



Since we are storing our "functions" in vector form, how do we perform a point-by-point multiplication using a square matrix?

$$b(x)f(x) \rightarrow \mathbf{Bf}$$

$$\begin{bmatrix} b_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & b_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & b_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & b_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & b_6 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} = \begin{bmatrix} b_1 f_1 \\ b_2 f_2 \\ b_3 f_3 \\ b_4 f_4 \\ b_5 f_5 \\ b_6 f_6 \end{bmatrix}$$

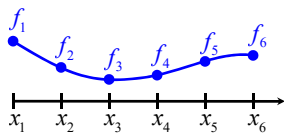
$\mathbf{B} \quad \mathbf{f} \quad \mathbf{Bf}$

Lecture 10

Slide 39

## First-Order Partial Derivative (1 of 2) CEM

How do we construct a square matrix so that when it premultiplies a vector, we get a vector containing the first-order partial derivative?



$$\frac{\partial}{\partial x} f(x) \rightarrow \mathbf{D}_x^{(1)} \mathbf{f}$$

$$\begin{bmatrix} \text{?} \\ \text{?} \\ \text{?} \\ \text{?} \\ \text{?} \\ \text{?} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} = \begin{bmatrix} (f_2 - f_6)/2\Delta_x \\ (f_3 - f_1)/2\Delta_x \\ (f_4 - f_2)/2\Delta_x \\ (f_5 - f_3)/2\Delta_x \\ (f_6 - f_4)/2\Delta_x \\ (f - f_5)/2\Delta_x \end{bmatrix}$$

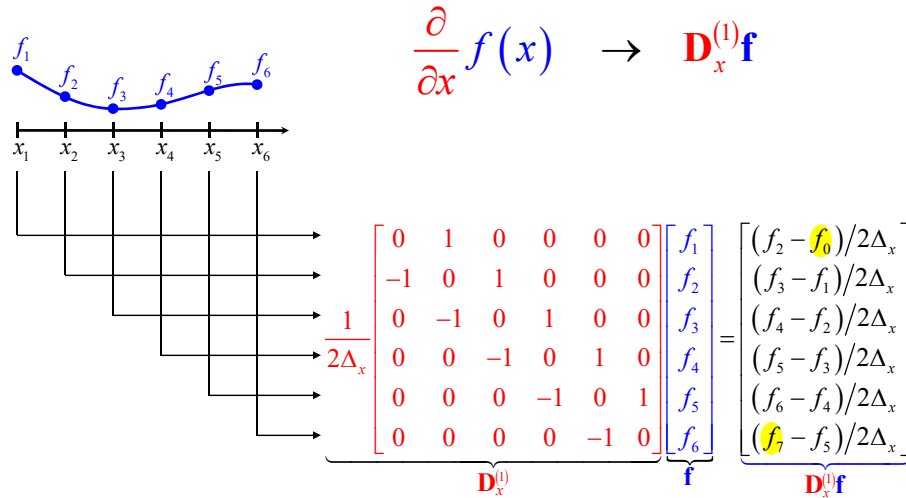
$\mathbf{D}_x^{(1)} \quad \mathbf{f} \quad \mathbf{D}_x^{(1)} \mathbf{f}$

Lecture 10

Slide 40

## First-Order Partial Derivative (2 of 2) CEM

How do we construct a square matrix so that when it premultiplies a vector, we get a vector containing the first-order partial derivative?

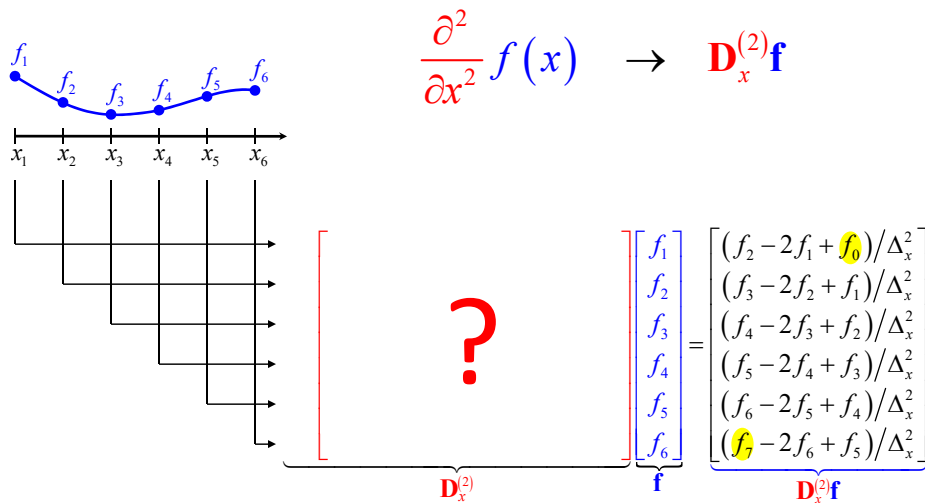


Lecture 10

Slide 41

## Second-Order Partial Derivative (1 of 2) CEM

How do we construct a square matrix so that when it premultiplies a vector, we get a vector containing the second-order partial derivative?



Lecture 10

Slide 42

## Second-Order Partial Derivative (2 of 2) CEM

How do we construct a square matrix so that when it premultiplies a vector, we get a vector containing the second-order partial derivative?

$$\frac{\partial^2}{\partial x^2} f(x) \rightarrow \mathbf{D}_x^{(2)} \mathbf{f}$$

$$\frac{1}{\Delta_x^2} \begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} = \begin{bmatrix} (f_2 - 2f_1 + f_0)/\Delta_x^2 \\ (f_3 - 2f_2 + f_1)/\Delta_x^2 \\ (f_4 - 2f_3 + f_2)/\Delta_x^2 \\ (f_5 - 2f_4 + f_3)/\Delta_x^2 \\ (f_6 - 2f_5 + f_4)/\Delta_x^2 \\ (f_7 - 2f_6 + f_5)/\Delta_x^2 \end{bmatrix}$$

$\mathbf{D}_x^{(2)} \mathbf{f}$

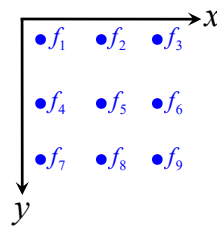
Lecture 10

Slide 43

## What About 2D Grids (1 of 2)? CEM

Two-dimensional grids are a little more difficult

$$\frac{\partial^2}{\partial x^2} f(x, y) \rightarrow \mathbf{D}_x^{(2)} \mathbf{f}$$



$$\frac{1}{\Delta_x^2} \begin{bmatrix} -2 & 1 & & & & & & & \\ 1 & -2 & 1 & & & & & & \\ & & 1 & -2 & 0 & & & & \\ & & & 0 & -2 & 1 & & & \\ & & & & 1 & -2 & 1 & & \\ & & & & & 1 & -2 & 0 & \\ & & & & & & 0 & -2 & 1 \\ & & & & & & & 1 & -2 & 1 \\ & & & & & & & & 1 & -2 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{bmatrix} = \begin{bmatrix} (f_2 - 2f_1 + f_{???})/\Delta_x^2 \\ (f_3 - 2f_2 + f_1)/\Delta_x^2 \\ (f_{???} - 2f_3 + f_2)/\Delta_x^2 \\ (f_5 - 2f_4 + f_{???})/\Delta_x^2 \\ (f_6 - 2f_5 + f_4)/\Delta_x^2 \\ (f_{???} - 2f_6 + f_5)/\Delta_x^2 \\ (f_8 - 2f_7 + f_{???})/\Delta_x^2 \\ (f_9 - 2f_8 + f_7)/\Delta_x^2 \\ (f_{???} - 2f_9 + f_8)/\Delta_x^2 \end{bmatrix}$$

$\mathbf{D}_x^{(2)} \mathbf{f}$

Lecture 10

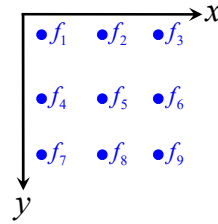
Slide 44

# What About 2D Grids (2 of 2)?



Two-dimensional grids are a little more difficult

$$\frac{\partial^2}{\partial y^2} f(x, y) \rightarrow \mathbf{D}_y^{(2)} \mathbf{f}$$



$$\frac{1}{\Delta_y^2} \begin{bmatrix} -2 & 0 & 0 & 1 & & & & & \\ 0 & -2 & 0 & 0 & 1 & & & & \\ 0 & 0 & -2 & 0 & 0 & 1 & & & \\ 1 & 0 & 0 & -2 & 0 & 0 & 1 & & \\ & 1 & 0 & 0 & -2 & 0 & 0 & 1 & \\ & & 1 & 0 & 0 & -2 & 0 & 0 & 1 \\ & & & 1 & 0 & 0 & -2 & 0 & 0 \\ & & & & 1 & 0 & 0 & -2 & 0 \\ & & & & & 1 & 0 & 0 & -2 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{bmatrix} = \begin{bmatrix} (f_4 - 2f_1 + f_{??})/\Delta_y^2 \\ (f_5 - 2f_2 + f_{??})/\Delta_y^2 \\ (f_6 - 2f_3 + f_{??})/\Delta_y^2 \\ (f_7 - 2f_4 + f_1)/\Delta_y^2 \\ (f_8 - 2f_5 + f_2)/\Delta_y^2 \\ (f_9 - 2f_6 + f_3)/\Delta_y^2 \\ (f_{??} - 2f_7 + f_4)/\Delta_y^2 \\ (f_{??} - 2f_8 + f_5)/\Delta_y^2 \\ (f_{??} - 2f_9 + f_6)/\Delta_y^2 \end{bmatrix}$$

Lecture 10

Slide 45

# Derivative Operators with Dirichlet Boundary Conditions



$$\mathbf{D}_x^{(2)} = \frac{1}{\Delta_x^2} \begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix}$$

Both of these matrices only have numbers along three of their diagonals.

This is called tridiagonal.

This suggests a fast way to construct these matrices.

$\mathbf{D}_x^{(2)}$  has some corrections shown in blue along two of its diagonals.

These matrices contain mostly zeros.

These are called a sparse matrices.

See MATLAB `sparse()` command.

Also see MATLAB `spdiags()` command.

$$\mathbf{D}_y^{(2)} = \frac{1}{\Delta_y^2}$$

$$\begin{bmatrix} -2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & -2 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -2 \end{bmatrix}$$

Lecture 10

Slide 46

## Why Do We Need Separate Matrix Operators for First- and Second-Order Derivatives? CEM

We know that,

$$\frac{\partial^2}{\partial x^2} = \frac{\partial}{\partial x} \frac{\partial}{\partial x} \qquad \frac{\partial^2}{\partial y^2} = \frac{\partial}{\partial y} \frac{\partial}{\partial y}$$

Can we just calculate  $\mathbf{D}^{(2)}$  from  $\mathbf{D}^{(1)}$ ?

$$\mathbf{D}_x^{(2)} \stackrel{?}{=} \mathbf{D}_x^{(1)} \mathbf{D}_x^{(1)} \qquad \mathbf{D}_y^{(2)} \stackrel{?}{=} \mathbf{D}_y^{(1)} \mathbf{D}_y^{(1)}$$

Yes, but this does not make efficient use of the grid. For a 5-point, 1D grid, we have

$$\mathbf{D}_x^{(1)} \mathbf{D}_x^{(1)} = \frac{1}{(2\Delta_x)^2} \begin{bmatrix} -1 & 0 & 1 & 0 & 0 \\ 0 & -2 & 0 & 1 & 0 \\ 1 & 0 & -2 & 0 & 1 \\ 0 & 1 & 0 & -2 & 0 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix}$$

This is not as accurate because it calculates the derivative with poorer grid resolution than is available.

$$\mathbf{D}_x^{(2)} = \frac{1}{\Delta_x^2} \begin{bmatrix} -2 & 1 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 & -2 \end{bmatrix}$$

This matrix operator makes optimal use of the available grid resolution.

Lecture 10

Slide 47

## 2D Derivative Operators for 1D Grids CEM

When  $N_x=1$  and  $N_y>1$

$\mathbf{D}_x = \mathbf{Z}$  zero matrix  
 $\mathbf{D}_y$  is standard for 1D grid

$$\mathbf{D}_x = \begin{bmatrix} 0 & & & 0 \\ & 0 & & \\ & & \ddots & \\ 0 & & & 0 \end{bmatrix}$$

When  $N_x>1$  and  $N_y=1$

$\mathbf{D}_x$  is standard for 1D grid  
 $\mathbf{D}_y = \mathbf{Z}$  zero matrix

$$\mathbf{D}_y = \begin{bmatrix} 0 & & & 0 \\ & 0 & & \\ & & \ddots & \\ 0 & & & 0 \end{bmatrix}$$

Lecture 10

Slide 48



**USE SPARSE MATRICES!!!!!!****CEM****WARNING !!**

The derivative operators will be **EXTREMELY** large matrices.

For a small grid that is just 100×200 points:

Total Number of Points:	20,000
Size of Derivate Operators:	20,000 × 20,000
Total Elements in Matrices:	400,000,000
Memory to Store One Full Matrix:	6 Gb
Memory to Store One Sparse Matrix:	1 Mb

**NEVER AT ANY POINT** should you use **FULL MATRICES** in the finite-difference method. Not even for intermediate steps. **NEVER!**

Lecture 10

Slide 49