DD2370 Computational Methods for Electromagnetics

Finite Difference

Stefano Markidis, KTH Royal Institute of Technology

Motivation

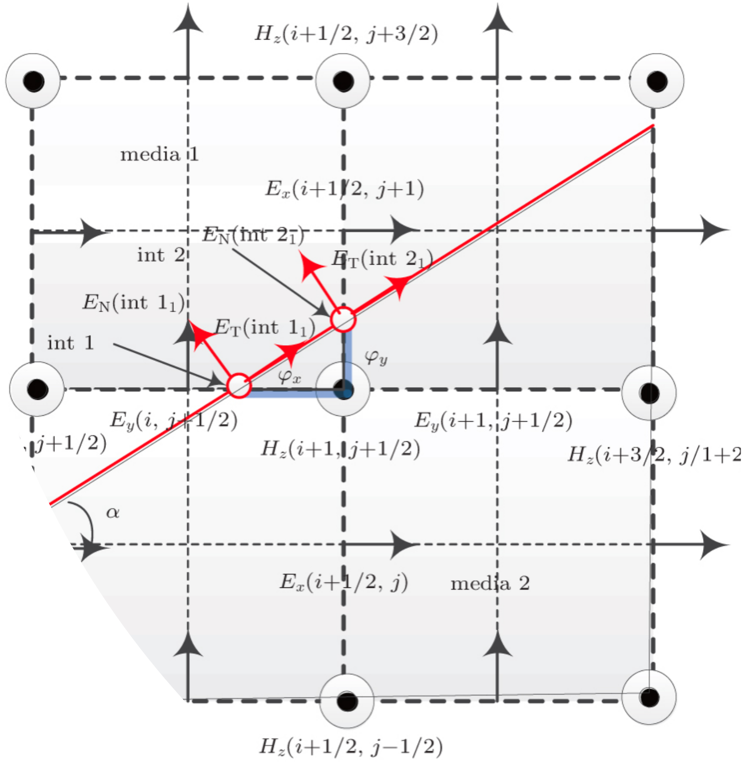
$$\nabla \cdot \mathbf{E} = \frac{\rho}{\varepsilon_0}$$

$$\nabla \cdot \mathbf{B} = 0$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{j} + \frac{1}{c^2}\frac{\partial \mathbf{E}}{\partial t}$$

1. Discretize space and time on finite number of grid points and time steps



2. Substitute continuous derivatives with algebraic equations for each node

Basic Mechanism for Finite Difference: Taylor Expansion

$$f(x+\Delta x) = f(x) + f'(x)\Delta x/1! + f''(x)\Delta x^2/2! + \ldots$$

Finite Difference of First Order – The Building Block

- Want to calculate f'(x) of a function defined only in discrete points …, x - **Δ**x, x, x + **Δ**x
- Three different kinds of approximations:
- **Forward-difference**

$$[f]'(x) = (f(x + \Delta x) - f(x))/\Delta x$$

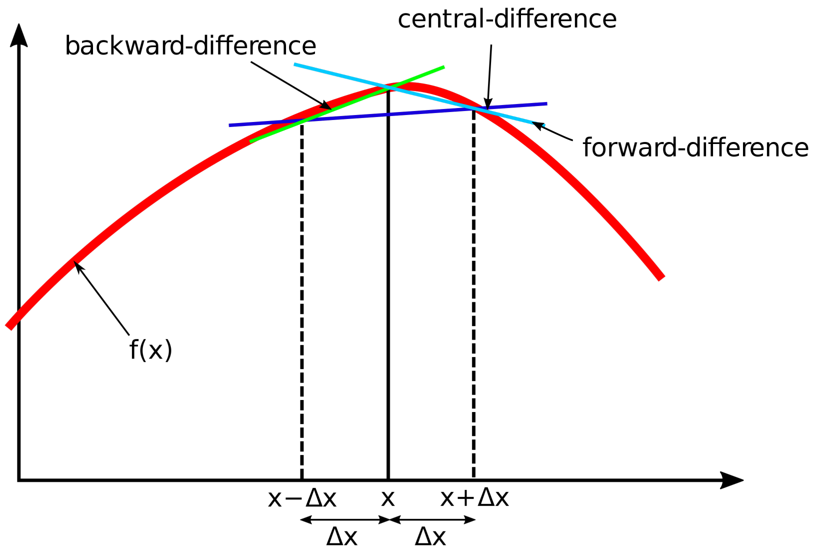
- **Backward-difference**

$$[f]'(x) = (f(x) - f(x - \Delta x))/\Delta x$$

- **Central-difference**

$$[f]'(x) = (f(x + \Delta x)) - f(x - \Delta x))/2\Delta x$$

- What is the best?



How do we Calculate the Truncation Error?

- Taylor expansion series

$$f(x+\Delta x) = f(x) + f'(x)\Delta x/1! + f''(x)\Delta x^2/2! + \ldots$$

- Forward-difference error

Truncation error

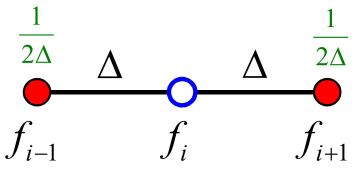
$$f'(x) = (f(x+\Delta x) - f(x))/\Delta x - f''(x)\Delta x/2! + \ldots$$

Forward-difference

What is the accuracy order?

Finite Difference of Second Order Derivative

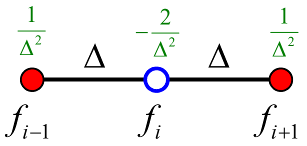
From definition of first order difference we can derive formulation for



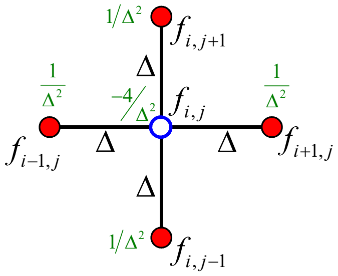
$$\frac{d}{dx} f\left(x_i\right) \cong \frac{f_{i+1} - f_{i-1}}{2\Delta}$$

1D

$$f'' = ((f_{i+1} – f_i)/\mathbf{\Delta} – (f_i + f_{i-1})/ \mathbf{\Delta}) / \mathbf{\Delta}$$



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



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

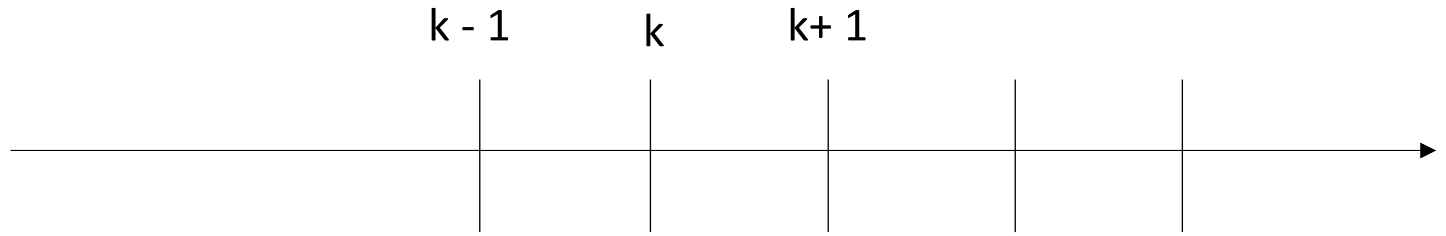
Finite Difference Method

1. We start with a differential equation that we want to solve, e.g.

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

2. We **approximate the derivatives with finite differences**

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



Finite Difference Method

The equation is expanded and we collect common terms

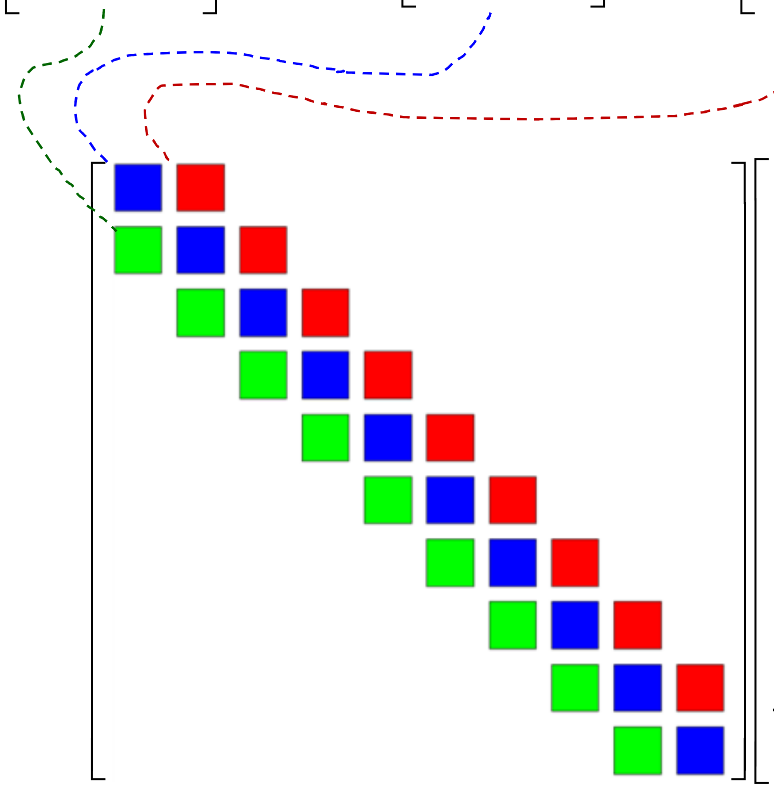
$$\frac{1}{\Delta^2} f\left(k+1\right) - \frac{2}{\Delta^2} f\left(k\right) + \frac{1}{\Delta^2} f\left(k-1\right) - \frac{a\left(k\right)}{2\Delta} f\left(k+1\right) + \frac{a\left(k\right)}{2\Delta} f\left(k-1\right) - b\left(k\right) f\left(k\right) = c\left(k\right)$$

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

Finite Difference Method

4. The final equation is used to populate a matrix equation

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



$$\begin{bmatrix} f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ \vdots \\ f(N-1) \\ f(N) \end{bmatrix} = \begin{bmatrix} c(1) \\ c(2) \\ c(3) \\ c(4) \\ c(5) \\ \vdots \\ c(N-1) \\ c(N) \end{bmatrix}$$

Excursus- Interpretation of Matrix

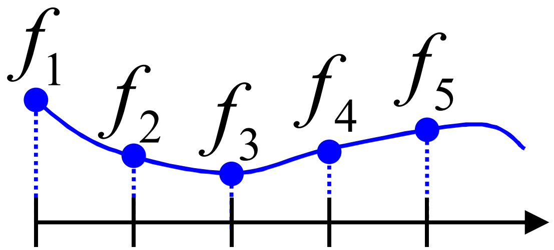
$$a_{11}x + a_{12}y + a_{13}z = b_1$$
$$a_{21}x + a_{22}y + a_{23}z = b_2$$
$$a_{31}x + a_{32}y + a_{33}z = b_3$$

$$\Rightarrow$$

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

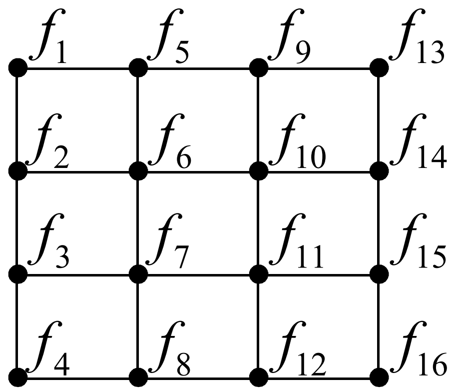
Excursus - Functions are Put Into Column Vectors

1-D System



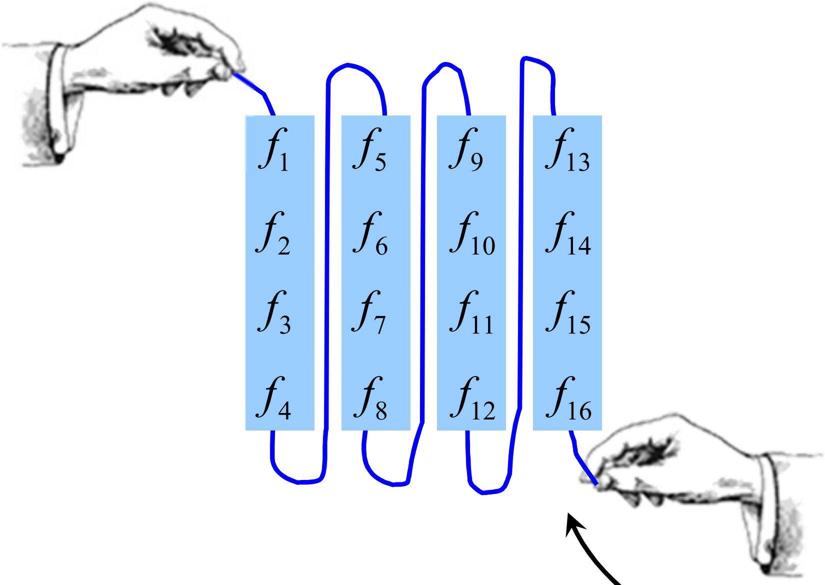
$$\Rightarrow \quad \mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix}$$

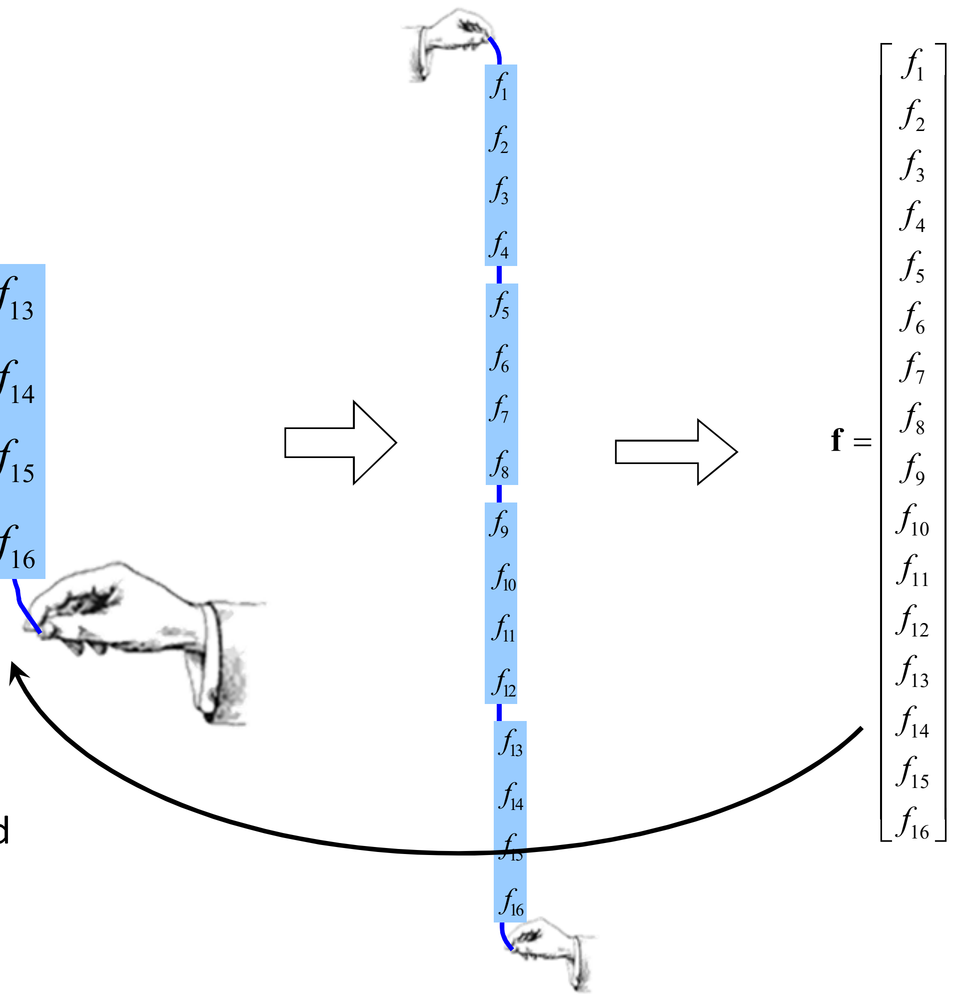
2-D Systems



$$\Rightarrow \quad \mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \\ f_{10} \\ f_{11} \\ f_{12} \\ f_{13} \\ f_{14} \\ f_{15} \\ f_{16} \end{bmatrix}$$

Excursus - Putting Functions into Column Vectors





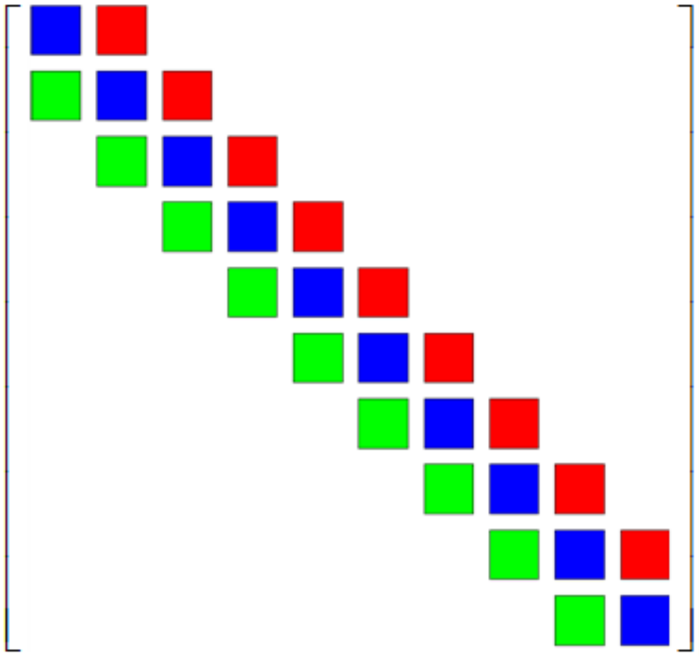
MATLAB ‘`reshape`’ command

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

Solving the Linear System

The matrix equation is solved for the unknown function *f*(*x*)

L



$$\begin{bmatrix} f(1) \\ f(2) \\ f(3) \\ f(4) \\ f(5) \\ \vdots \\ f(N-1) \\ f(N) \end{bmatrix} = \begin{bmatrix} c(1) \\ c(2) \\ c(3) \\ c(4) \\ c(5) \\ \vdots \\ c(N-1) \\ c(N) \end{bmatrix}$$

In Matlab: f = L/c

Other linear solvers can be used, see appendix A of the textbook