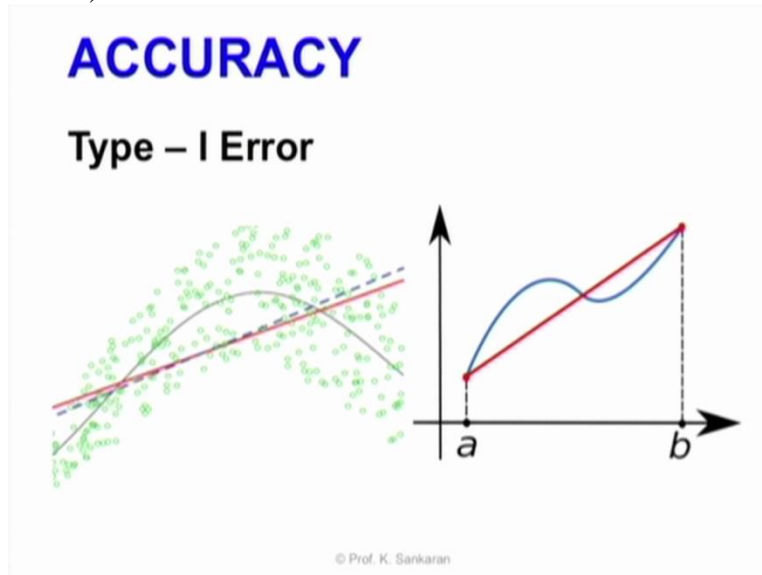
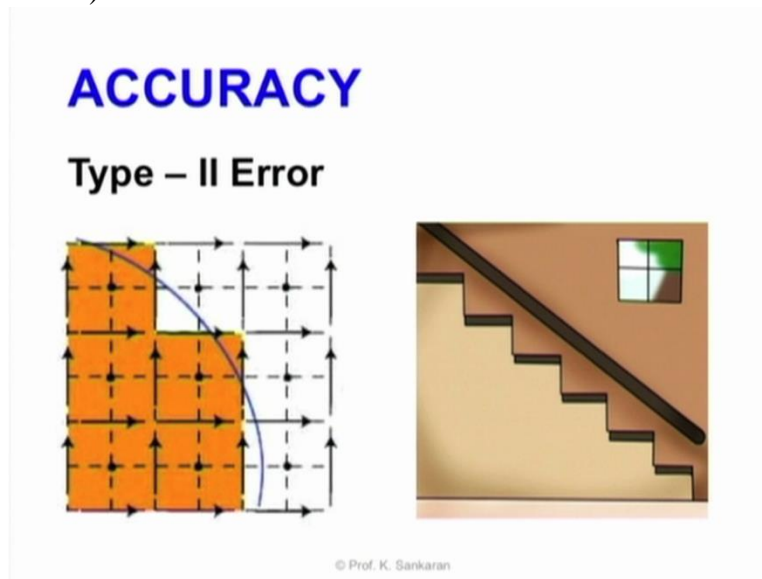


Computational Electromagnetics and Applications
Professor Krish Sankaram
Indian Institute of Technology, Bombay
Summary of Week 2

First we looked at the concept of numerical accuracy from different perspectives
(Refer Slide Time: 00:26)



The first type of error we addressed emerges from linearization of any nonlinear phenomena
(Refer Slide Time: 00:33)

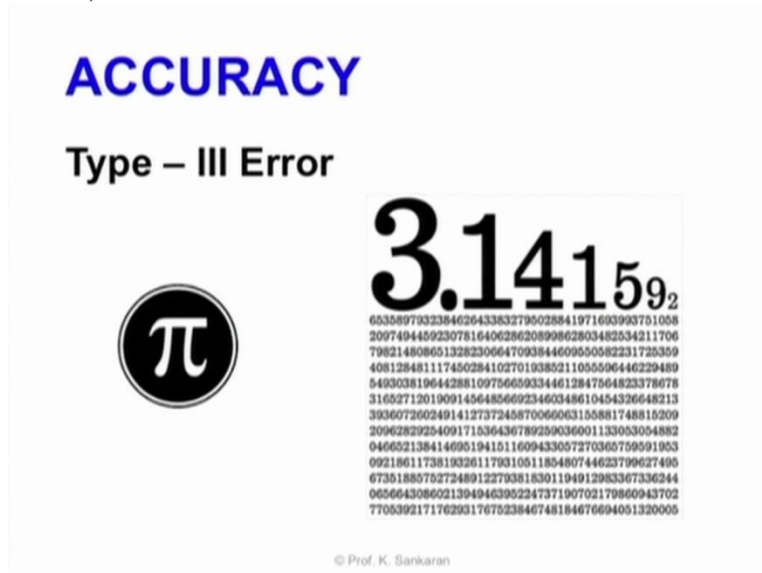


The second type of error is due to spatial discretization. We also called this type of error as the staircasing error in the case of Finite difference method

(Refer Slide Time: 00:45)

ACCURACY

Type – III Error



π

3.14159₂

653597932384629433832795028841971693993701058
209749449230781640628620899862803482534211706
7982148086513282306647093844609550582231725359
408128481117450284102701938521105596446229489
549302819642881097566592344612847564822378678
3165271201909145648566923460348610454326648213
3936072602491412737245870065063155881748815209
2096282925409171536436789259036001133053054882
0466521384146951941511609433057270365799591953
0921861173819326117931051185480744623799627495
673518857527489122793818301194912983367336244
0656643086021394946395234737190702179860943702
7705392171762931767523846748184676694051320005

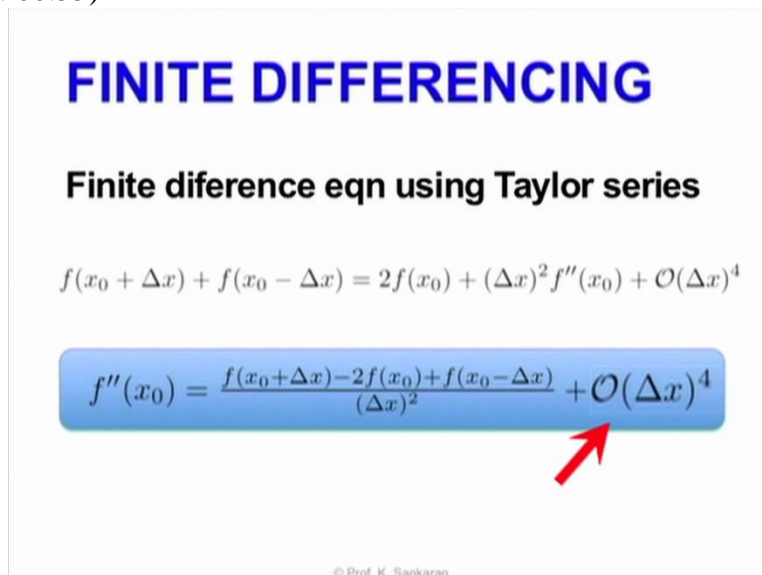
© Prof. K. Sankaran

The third type of error comes from floating point truncation or the round up error in our calculation.

(Refer Slide Time: 00:55)

FINITE DIFFERENCING

Finite difference eqn using Taylor series

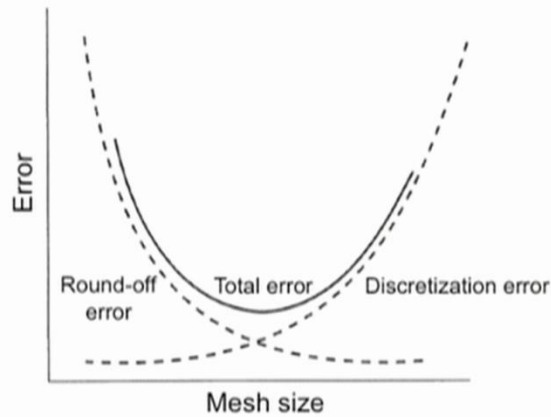
$$f(x_0 + \Delta x) + f(x_0 - \Delta x) = 2f(x_0) + (\Delta x)^2 f''(x_0) + \mathcal{O}(\Delta x)^4$$
$$f''(x_0) = \frac{f(x_0 + \Delta x) - 2f(x_0) + f(x_0 - \Delta x)}{(\Delta x)^2} + \mathcal{O}(\Delta x)^4$$


© Prof. K. Sankaran

We have a similar type of error that we addressed while studying Taylor series approximation. If you remember we talked about the order of truncation error while introducing Taylor series approximation for different Finite difference schemes.

(Refer Slide Time: 01:09)

ACCURACY



© Prof. K. Sankaran

In any modelling exercise the total error is a summation of all these errors we have looked at the graph which illustrates this as the function of spatial discretization.

(Refer Slide Time: 01:23)

DISPERSION

For $\Delta t = \frac{\Delta x}{c}$ $\rightarrow \tilde{k} = k$

$$\Delta t = 0.5 \frac{\Delta x}{c} \rightarrow \frac{\tilde{k} - k}{k} \approx \frac{1}{32} (k \Delta x)^2 = \frac{\pi^2}{8} \left(\frac{\Delta x}{\lambda} \right)^2$$

Error converges quadratically

$$\left(\frac{\Delta x}{\lambda} \right)^2$$

Error control by Δx choice

© Prof. K. Sankaran

We also addressed another type of numerical error related to the face of the numerical solution which is termed as dispersion error.

(Refer Slide Time: 01:38)

STABILITY



© Prof. K. Sankaran

Later we looked at the concept of stability of a numerical scheme.

(Refer Slide Time: 01:43)

STABILITY

LAX EQUIVALENCE THEOREM

STABILITY is necessary and sufficient condition for convergence of a consistent linear FD model

© Prof. K. Sankaran

We briefly introduced the Lax theorem for stability.

(Refer Slide Time: 01:47)

STABILITY

Define an error at time step n as e^n for one independent variable

Then at time step $n + 1$, this error amplifies as,

$$e^{n+1} = ge^n$$

where g is amplification factor

© Prof. K. Sankaran

And also investigated the role of error amplification factor.

(Refer Slide Time: 01:52)

STABILITY

von Neumann Stability Analysis

Decompose round off errors into **Fourier space** and analyze their **time evolution**

© Prof. K. Sankaran

We shortly mentioned about Von Neumann analysis for stability to study the stability of any numerical scheme

(Refer Slide Time: 02:00)

STABILITY

$$|g|^2 = \beta^2 + 1 - \beta^2 = 1$$

Conditionally stable scheme

Stability condition is given by

$$-1 \leq 1 - 2r \sin^2\left(\frac{k\Delta x}{2}\right) \leq 1$$

\approx CFL condition $\frac{\Delta x}{\Delta t} \geq c$

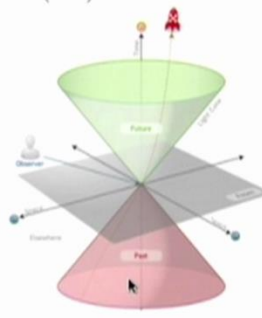

© Prof. K. Sankaran

(Refer Slide Time: 02:05)

STABILITY

For FTCS to be stable, $0 < \frac{2k\Delta t}{(\Delta x)^2} \leq 1$

$\Delta t \leq \frac{(\Delta x)^2}{2k}$



© Prof. K. Sankaran

We derived CFL light condition for forward in time centered in space scheme giving the analogy of the light cone.

(Refer Slide Time: 02:10)

$$\nabla^2 \varphi + k^2 \varphi = 0$$
$$1D \Rightarrow \frac{\partial^2 \varphi}{\partial x^2} + k^2 \varphi = 0$$

$k = \text{eigenvalue}$

Dirichlet BC $\Rightarrow x=0$ $x=a$

$\varphi(x=0)=0$ & $\varphi(x=a)=0$

6 points

As a first exercise we studied one dimensional Helmholtz equation.

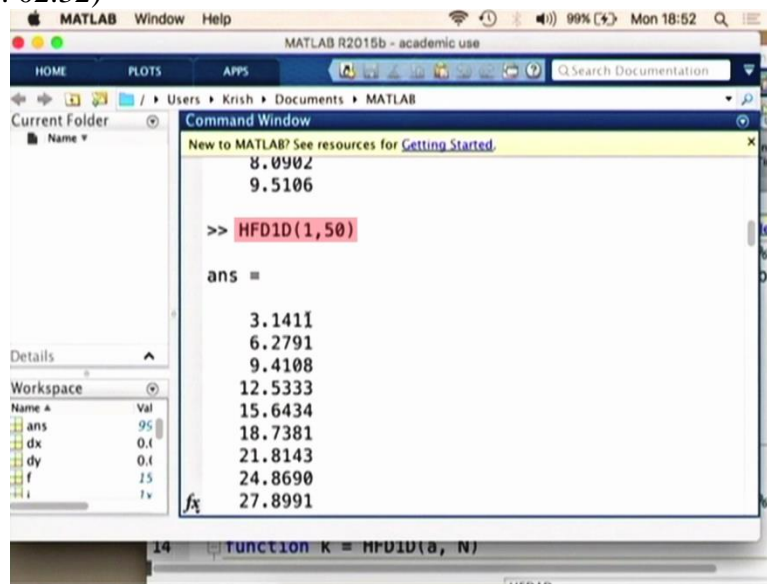
(Refer Slide Time: 02:16)

```
MATLAB R2015b - academic use
HOME PLOTS APPS
Users \Krish \Documents \MATLAB
Current Folder
Workspace
Name Val
ans 2.1
dx 0.1
dy 0.1
f 2.5
i 7x
Command Window
New to MATLAB? See resources for Getting Started.
>> HFD1D(1,1)
ans =
Empty matrix: 0-by-1
>> HFD1D(1,2)
ans =
2.8284
fx >>
```

And calculated the Eigen values by changing various parameters of the simulation.

We also noticed the accuracy of the simulation improved as we refine the spatial discretization which confirms that the solution is converging and the method is stable.

(Refer Slide Time: 02:32)



The image shows a MATLAB R2015b window with the Command Window open. The Command Window displays the following output:

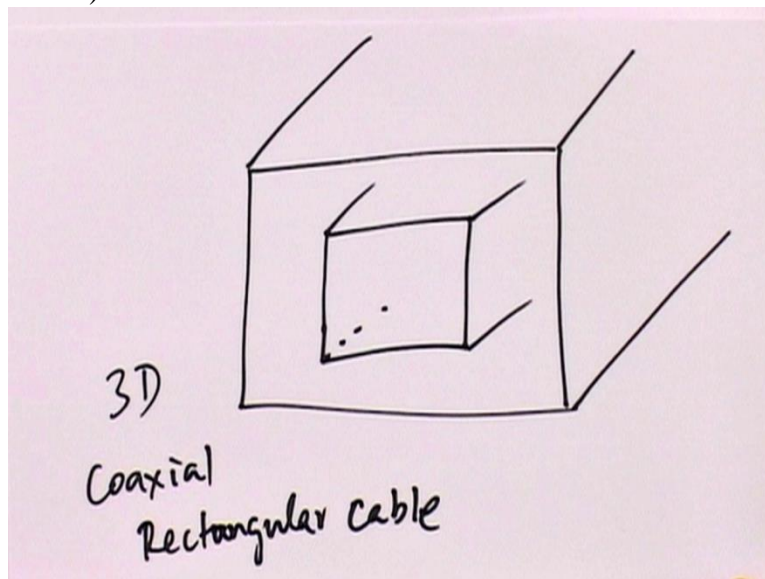
```
New to MATLAB? See resources for Getting Started.  
8.0902  
9.5106  
  
>> HFD1D(1,50)  
  
ans =  
  
3.1411  
6.2791  
9.4108  
12.5333  
15.6434  
18.7381  
21.8143  
24.8690  
27.8991
```

The workspace window shows the following variables:

Name	Val
ans	95
dx	0.0
dy	0.0
f	15
i	1x

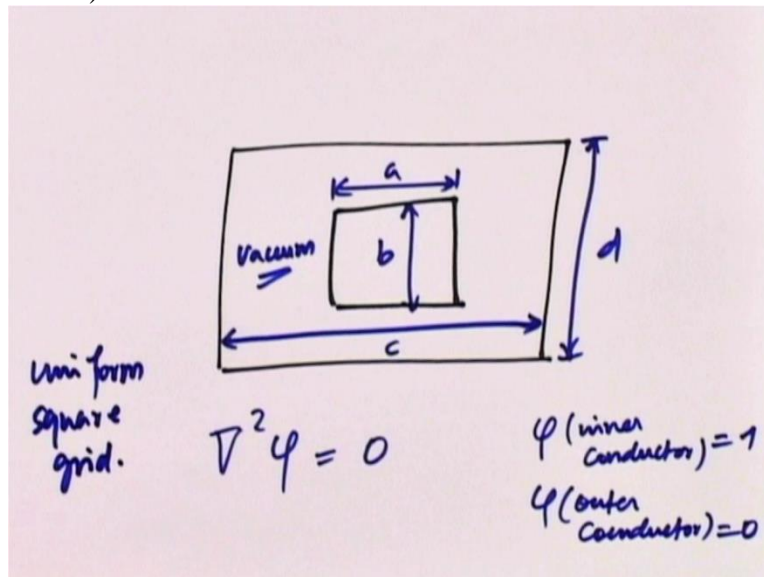
The next exercise is that of a Coaxial Rectangular capacitor

(Refer Slide Time: 02:40)



Where we calculated the capacitance per unit length by solving Laplace equation.

(Refer Slide Time: 02:46)



We used Gauss Seidel iteration method in order to compute the solution of the algebraic finite difference equation. We remarked that the solution converges and the speed of convergent itself greatly varies by the choice of proper iterative method.

(Refer Slide Time: 03:05)

```
% Gauss Seidel iteration
oldcap = 0;
for iter = 1:1000 % Maximum number of iterations|
    f = seidel(f,mask,n,m); % Perform Gauss-Seidel iteration
    cap = gauss(n,m,h,f); % Compute the capacitance
    if (abs(cap-oldcap)/cap < tol)
        break % Stop if change in capacitance is
            % sufficiently small
    else
        oldcap = cap; % Continue until converged
    end
end
```

capacitor Ln 52 Col

We also noticed the accuracy of computed capacitance improved with respect to mesh refinement confirming the convergence of the method.

(Refer Slide Time: 03:15)

```
Command Window
New to MATLAB? See resources for Getting Started.

>> HFD1D(1,100)

ans =

    3.1415
    6.2822
    9.4213
   12.5581
   15.6918
   18.8217
   21.9469
   25.0666
fx
```

We noticed that the forward in time centered in space scheme is inherently unstable.

(Refer Slide Time: 03:23)

STABILITY

For FTCS to be stable, $0 < \frac{2k\Delta t}{(\Delta x)^2} \leq 1$

© Prof. K. Sankaran

And one has to keep the time stepping prohibitively small to simulate any practical problem.

(Refer Slide Time: 03:26)

LAX METHOD

To cure instability

$$u_{n,j} \rightarrow \frac{1}{2}(u_{n+1,j} + u_{n-1,j})$$

Advection equation becomes

$$u_{n,j+1} = \frac{1}{2}(u_{n+1,j} + u_{n-1,j}) - \frac{c\Delta t}{2\Delta x}(u_{n+1,j} + u_{n-1,j})$$

© Prof. K. Sankaran

In order to cure this instability we introduced Lax method. We illustrated the Lax method using an example of advection equation.

(Refer Slide Time: 03:40)

LAX METHOD

To cure instability

$$u_{n,j} \rightarrow \frac{1}{2}(u_{n+1,j} + u_{n-1,j})$$

Advection equation becomes

$$u_{n,j+1} = \frac{1}{2}(u_{n+1,j} + u_{n-1,j}) - \frac{c\Delta t}{2\Delta x}(u_{n+1,j} + u_{n-1,j})$$

© Prof. K. Sankaran

The ((03:41)) of this method lies in taking the average of two neighboring special terms in the forward in time centered in space scheme.

I encourage you to practice the exercises and examples that we have studied in this week and get ready for the next week. So until then Good Bye!