ECE3340
Numerical Methods for Differential Equation Systems

PROF. HAN Q. LE

Note: PPT file is the main outline of the chapter topic – associated Mathematica file(s) contain details and assignments

Introduction

We have seen problems involved differential equations solved with Fourier transform and Laplace transform.

But they are not always applicable.

Why?

- Sometimes, differential equations have to be numerically calculated (solved) directly in the time domain:
  - non-linear system that is not reducible to integrable solution (we talked about this)
  - no analytic response model known, or
  - too complicated for "finesse" approach and "brute force" must be used ← very common, more often than not, especially with complex boundary or initial conditions .
- Functions (entities of interest) and the derivatives are not or cannot be known analytically, and thus must be numerically approximated.
- A method to approximate is finite difference: a numerical approach to calculate differentiation when they are not analytically known or too complicated to calculate by other means.

Note this key concept

A quick example of FD

- Key concept: differentiation vs. finite difference

analytic

$$\frac{df}{dx}\Big|_{x=x_i} = \lim_{h\to 0} \frac{f[x_i+h]-f[x_i]}{h}$$

numerical

$$\frac{df}{dx}\Big|_{x=x_i} \sim \frac{f[x_i+h]-f[x_i]}{h}$$

$$\frac{d^2f}{dx^2}\Big|_{x=x_i} \sim \frac{\frac{f[x_i+h]-f[x_i]}{h} - \frac{f[x_i]-f[x_i-h]}{h}}{h} = \frac{f[x_i+h]-2\,f[x_i]-f[x_i-h]}{h^2}$$

and so on…

- A more general concept: finite difference for any discrete sampling system, regardless whether the variables are continuous or not

$$f[x] \to \{f_0,\ f_1,\ f_2,\ \ldots,\ f_{k-1},\ f_k,\ f_{k+1},\ \ldots\ \}$$

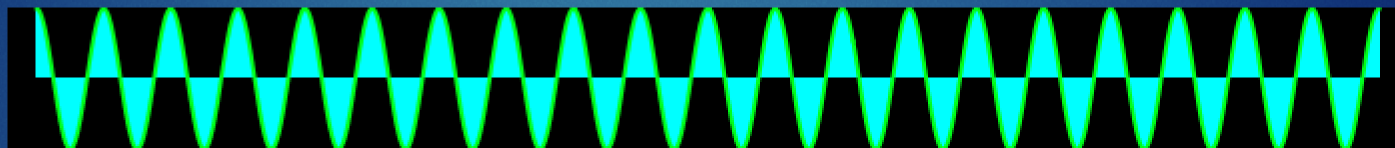
- The computation deals only with values at discrete sampling points: can be used to approximate continuous systems; intrinsically suitable to computer calculation. (remember DFT?)

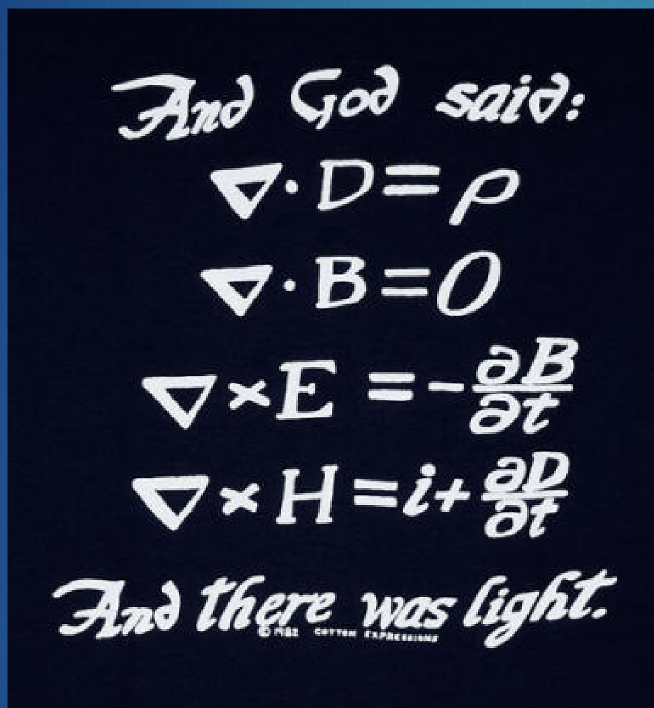
A brief review

- The essence is to “know” how good the approximation is: in other words, an essence of the method is to optimize for accuracy and precision and set upper limits for errors at values of interest.
  - “Exact differentiation” is irrelevant – even if derivatives are known analytically, they can only be computed with some finite precision.
  - As long as it is within error tolerances, FD is just as good.
- Specific FD algorithms have been developed for numerical solutions to various DEs. Example: the very popular Finite-Difference Time-Domain (FDTD) method for EM waves (space-and- time differential equations).

An Introductory example of one-dimensional FDTD application to electrical engineering

Remember this?





$$\nabla \times \mathbf{E} + \frac{\mu}{c}\frac{\partial \mathbf{H}}{\partial t} = 0$$

$$\nabla \times \mathbf{H} - \frac{\epsilon}{c}\frac{\partial \mathbf{E}}{\partial t} = 0$$

This is an example of a system of linear differential equations of rank (dimension) 2 (two unknowns).

$$\nabla\times\mathbf{E}+\frac{\mu}{c}\frac{\partial\mathbf{H}}{\partial t}=0$$
$$\nabla\times\mathbf{H}-\frac{\epsilon}{c}\frac{\partial\mathbf{E}}{\partial t}=0$$

$$\nabla\times\vec{\mathbf{E}} = \left|\begin{pmatrix}\hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{z}}\\ \partial_x & \partial_y & \partial_z\\ E_x & E_y & E_z\end{pmatrix}\right| \to \left|\begin{pmatrix}\hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{z}}\\ \partial_x & \partial_y & \partial_z\\ E_x[z,t] & 0 & 0\end{pmatrix}\right| = \hat{\mathbf{y}}\ \frac{\partial E_x[z,t]}{\partial z}$$

$$\frac{\partial E_x[z,t]}{\partial z}+\frac{\mu}{c}\frac{\partial H_y[z,t]}{\partial t}=0$$

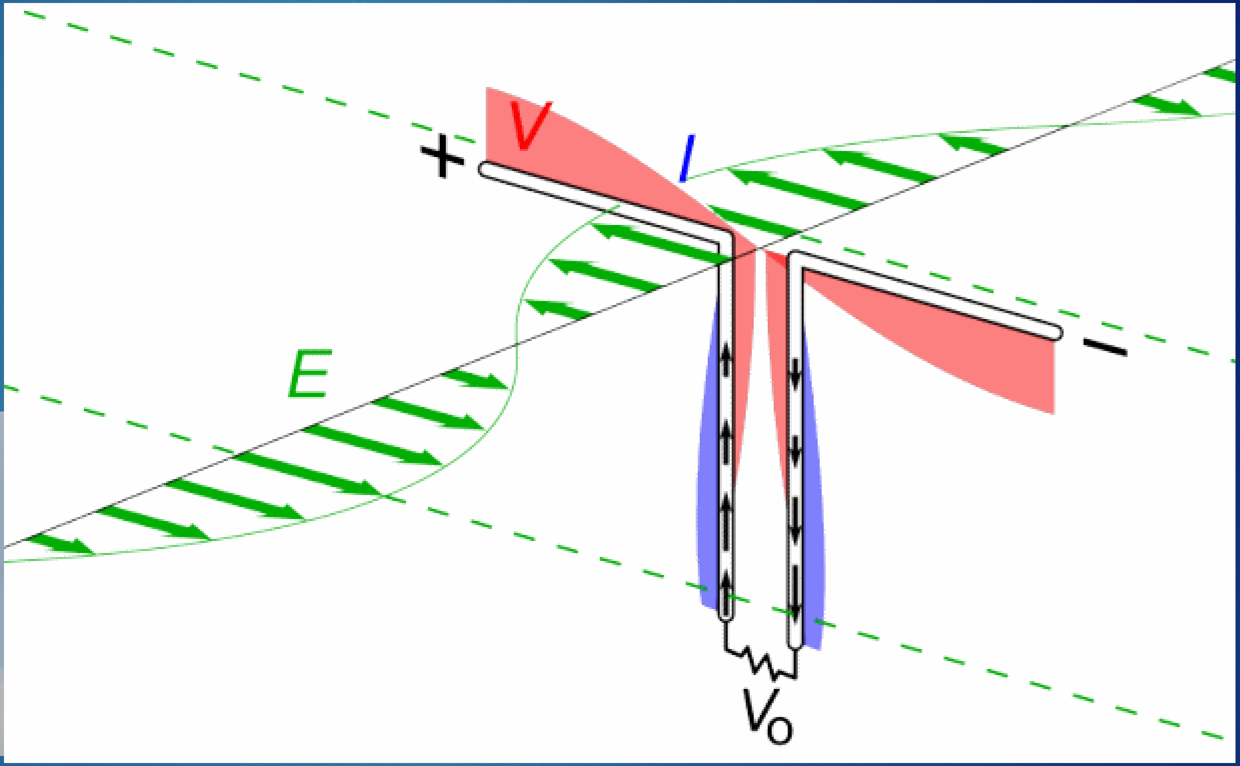
$$\nabla\times\vec{\mathbf{H}} = \left|\begin{pmatrix}\hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{z}}\\ \partial_x & \partial_y & \partial_z\\ H_x & H_y & H_z\end{pmatrix}\right| \to \left|\begin{pmatrix}\hat{\mathbf{x}} & \hat{\mathbf{y}} & \hat{\mathbf{z}}\\ \partial_x & \partial_y & \partial_z\\ 0 & H_y[z,t] & 0\end{pmatrix}\right| = -\hat{\mathbf{x}}\ \frac{\partial H_y[z,t]}{\partial z}$$

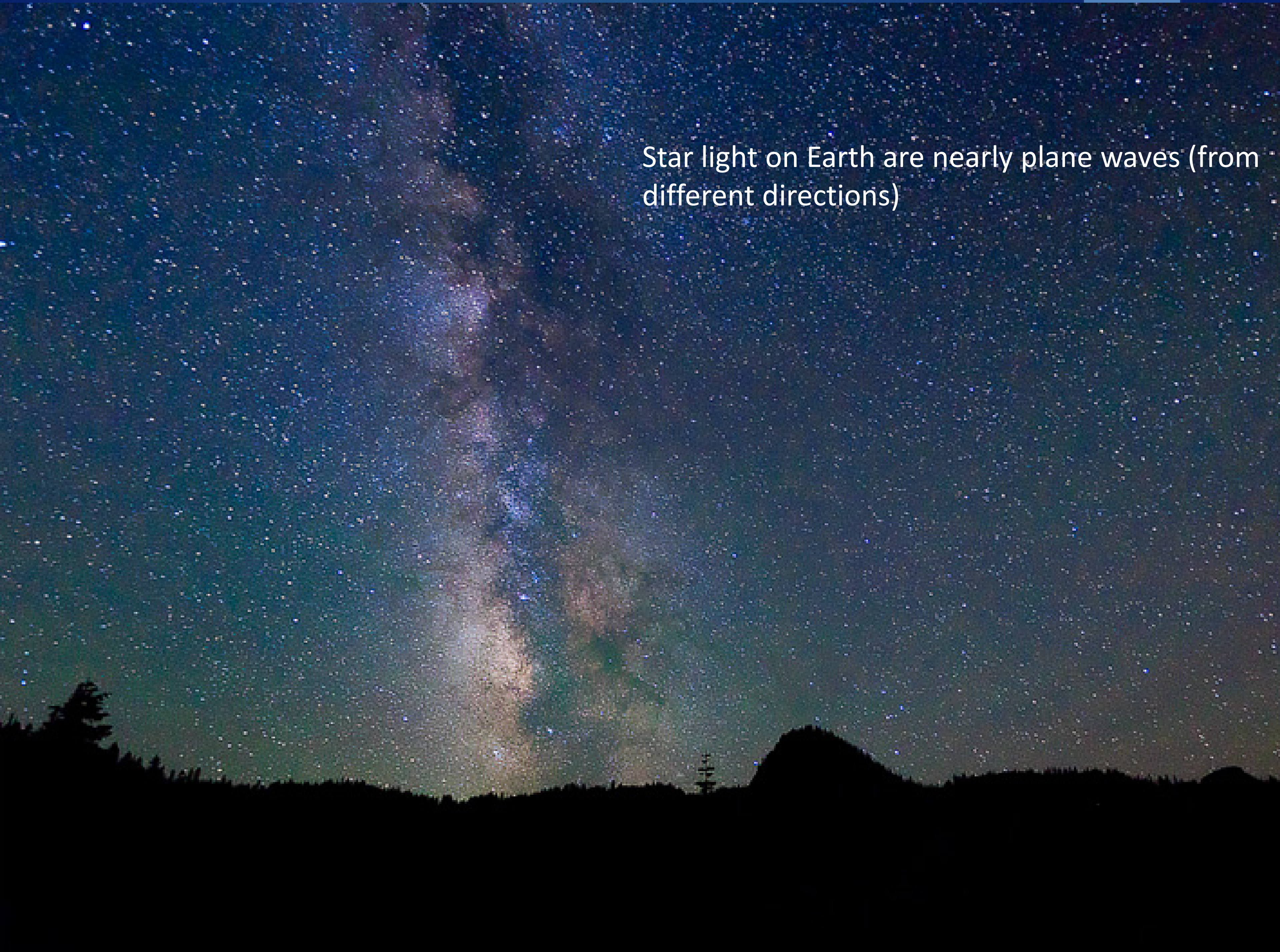
$$\frac{\partial H_y[z,t]}{\partial z}+\frac{\epsilon}{c}\frac{\partial E_x[z,t]}{\partial t}=0$$

Similar concept of state space DEs (but only time variable)

$$\dot{Q}_i = \frac{\partial\bar{H}}{\partial P_i}=0$$
$$\dot{P}_i = -\frac{\partial\bar{H}}{\partial Q_i}=0$$

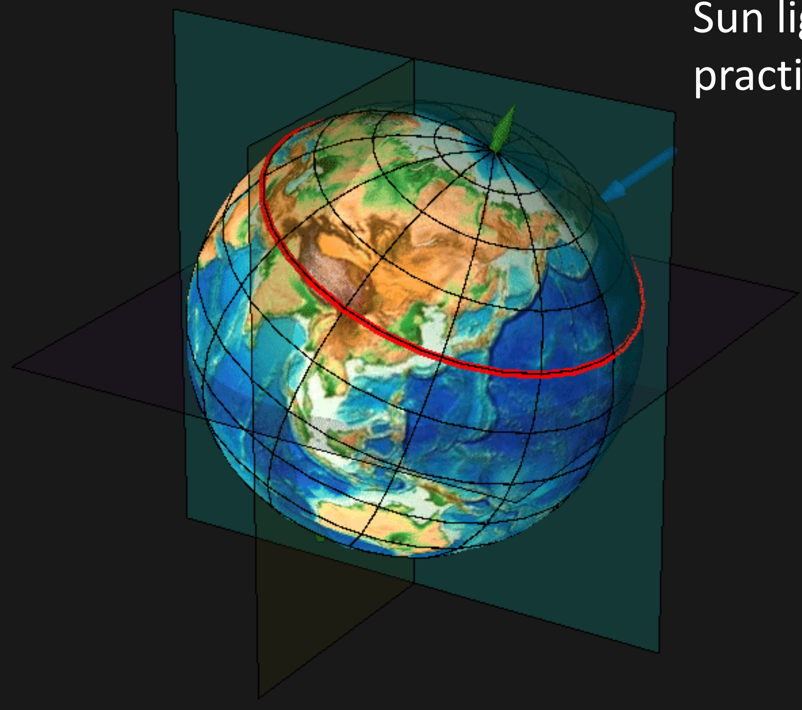
The simplest solution to the Maxwell's equations above describes linearly polarized electromagnetic waves

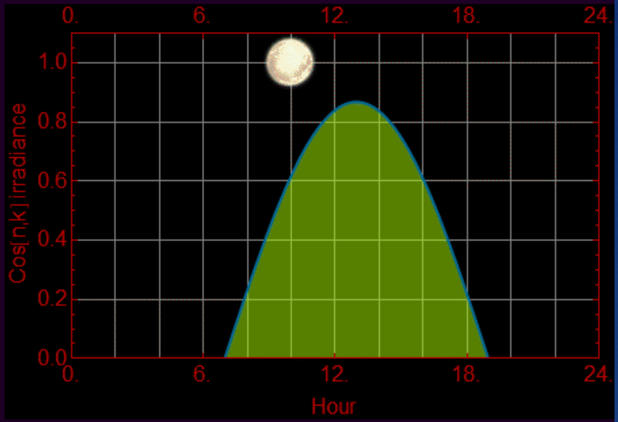


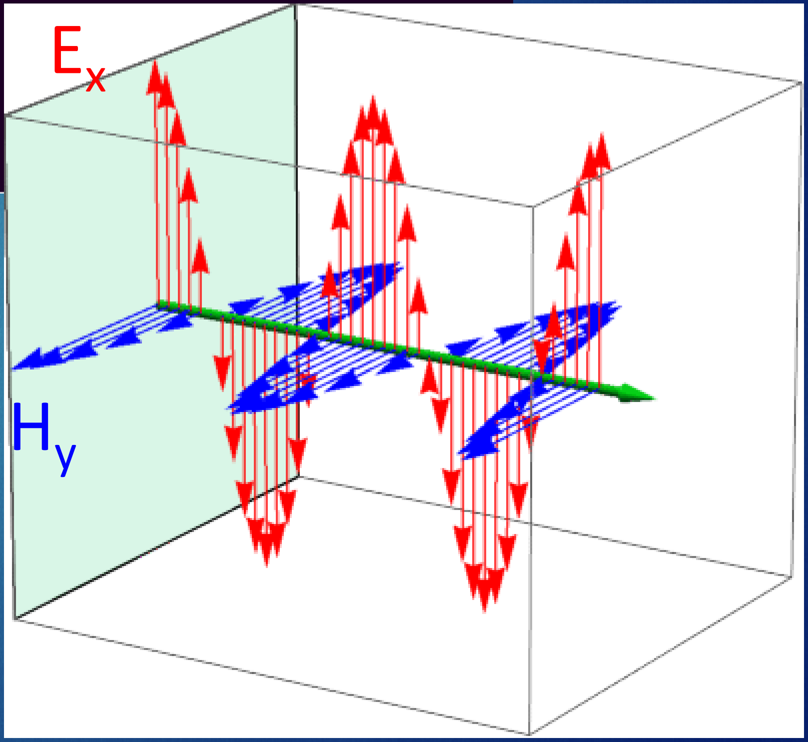


Star light on Earth are nearly plane waves (from different directions)

Sun light on Earth is also practically plane wave





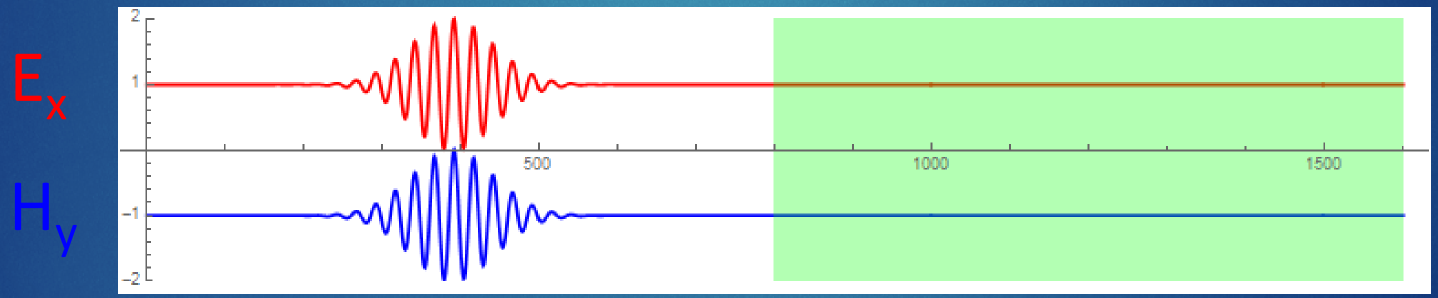


Finite-Difference Time-Domain (FDTD) method is a powerful algorithm to solve Maxwell's equations numerically for problems with complex geometry and media (brute force approach).

While the solution for a continuous infinite plane wave is trivial, the solution for a pulse wave, e. g. a laser light pulse is not always trivial (although exact analytic solutions are known for certain cases).

Below is an illustration of the FDTD method applied to a Gaussian light pulse (exact solution is known, and hence can be use to verify the FDTD result).

(this file can be too big for Window gif player – open with QuickTime)



(The FDTD solution is numerically accurate, but incorrect with regard to physics – but that's the topic for another day – see here)

Overview: the big picture of numerical approximation of differential equations

A problem-centric perspective

Linearization

Fitting,
Interpolation
Approximation

The problem

$f(x_1, x_2 \ldots, t)$, $\mathbf{F}(\mathbf{x}, t)$

$f'(x_1, x_2 \ldots, t)$, $\nabla \mathbf{F}(\mathbf{x}, t)$, $\nabla \times \mathbf{F}(\mathbf{x}, t)$

$f^{(n)}(x_1, x_2 \ldots, t)$, $\nabla^2 \mathbf{F}(\mathbf{x}, t)$, $\nabla \times (\nabla \times \mathbf{F}(\mathbf{x}, t))$

- too complex for analytic solution
- non-linear relationships (not reducible)
- phenomenological (empirical model) with limited or discrete data

Finite difference methods (e. g. FDTD)

Finite element methods (FEM)

- No, we'll not (won't even try) to cover all that…
- Most important:
  - A basic concept of the methods
  - The experience to know what to apply to solve a problem
- Algorithms have been well developed, software are commercially available and well tested – there is no need to develop one's own software unless for learning or highly specialized applications and research. Example:
  - Dedicated FDTD packages for EM waves are available as freeware (no support, no debugging help) and commercially (can be pricey but reliable and low risk of bugs)
  - Similarly for FEM
  - Other functions are available in Mathematica, MATLAB. Codes are also available in Mathematica and MATLAB for FDTD and FEM as well – but may not have friendly UI, or run as fast

An example: data least-square fitting and modeling

(see other chapter)

Interpolation and extrapolation

(see other chapter)

Example: finite element method

(see other chapter)

Finite-difference methods for ordinary differential equation

Numerical methods to O.D.E.

Summary (takeaways)

- Differential equations are involved in a big class of problems in quantitative physical science and electrical engineering.
- Transform approaches (see previous lecture of Fourier, Laplace and other transforms) and associated numerical methods form a big part of solutions.
- Direct numerical methods (brute force) such as finite difference for computing DE solutions are also well developed and form an essential tool for highly complex problems (when all else fail).
  - Complex geometry and boundary for space-time problem (very common in EM waves, which is why FDTD is used).
  - Multi-elements and complex behaviors, e. g. electrons/holes in semiconductor devices (very big software package to simulate device performance).
  - Other engineering: thermal analysis, computational fluid dynamic, …

Learning objectives: ability to apply these numerical methods to ECE problems (easy: circuits; next level: signal processing & control; advanced: EM waves)

Epilog

So, with all these DE solver packages, everything is solved, right? I don't have to learn anything, right?

- Uhm… no.
- These are tools. Real life engineering is to understand problems, think and design solutions, test and implement. Do not be confused between using tools and obtaining solutions.
- Furthermore, numerical tools are to help us gain insight and understanding for the bigger picture of things <- contribution to the growth and maturity of knowledge