An introduction to lightlabsFD

- The lightlabsFDS mission
- Solver
- Interface
- Hardware
- End it

The lightlabsFDS mission

To enable engineers and scientists to characterize optic quickly, simply, and cost-effectively.

Innovate on three fronts to make this a reality:

- Solve for electromagnetic fields in the frequency domain,
- Run simulations from pre-installed scientific software (Ma
- Offload computation to centralized custom-tuned hardwar

Solving electromagnetics in the frequency

- lightlabsFDS: Frequency-Domain Solver

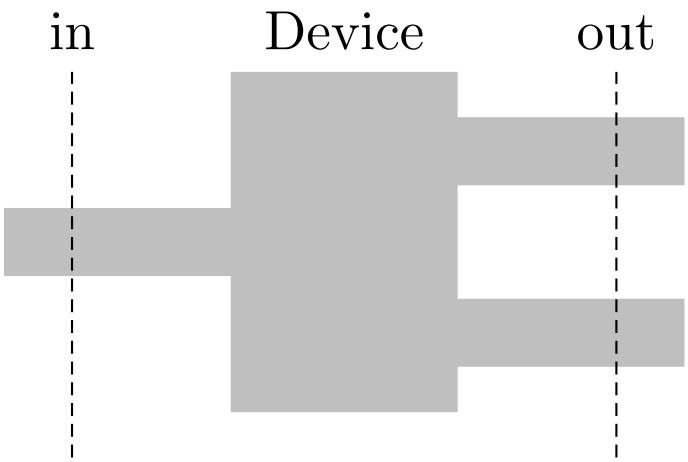
- Solves

$$\nabla \times \mu^{-1} \nabla \times E - \omega^2 \epsilon E = -i\omega J.$$

  - Inputs: frequency (ω), structure (μ, ϵ), and excitation (
  - Outputs: electromagnetic fields (E, H, D, B).

- *Many* practical advantages over existing time-domain solv

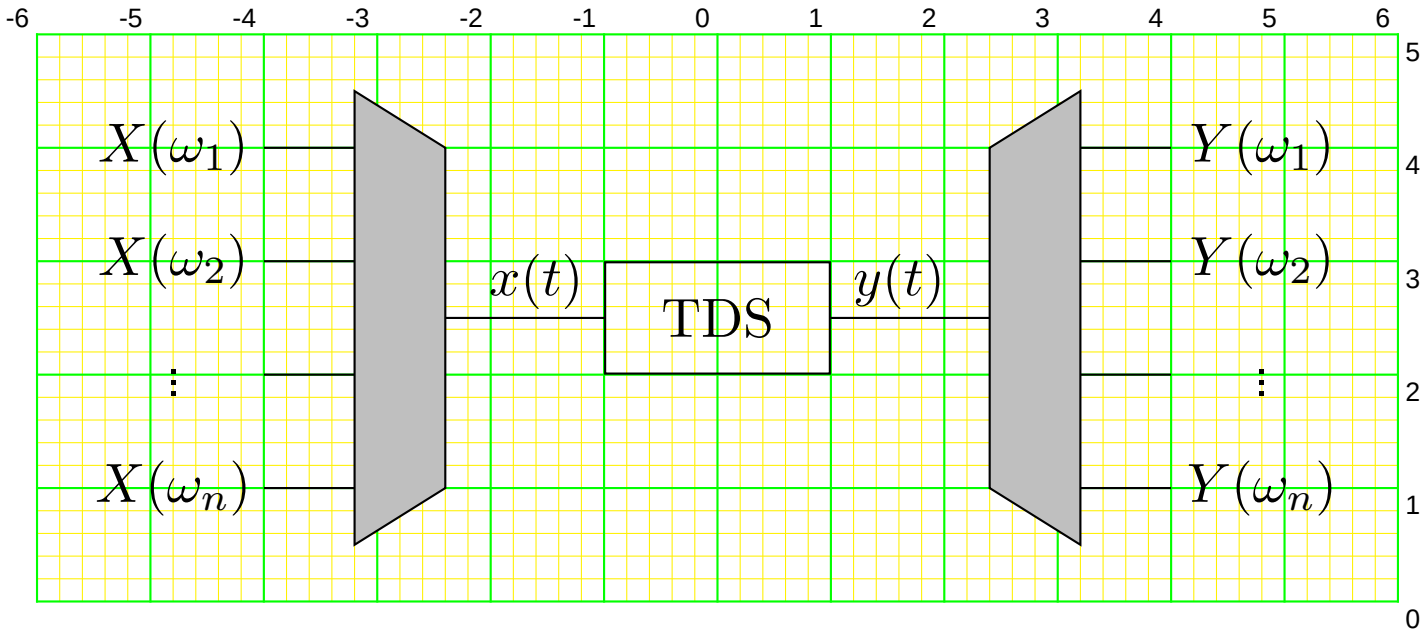
Example



Time-domain issues include

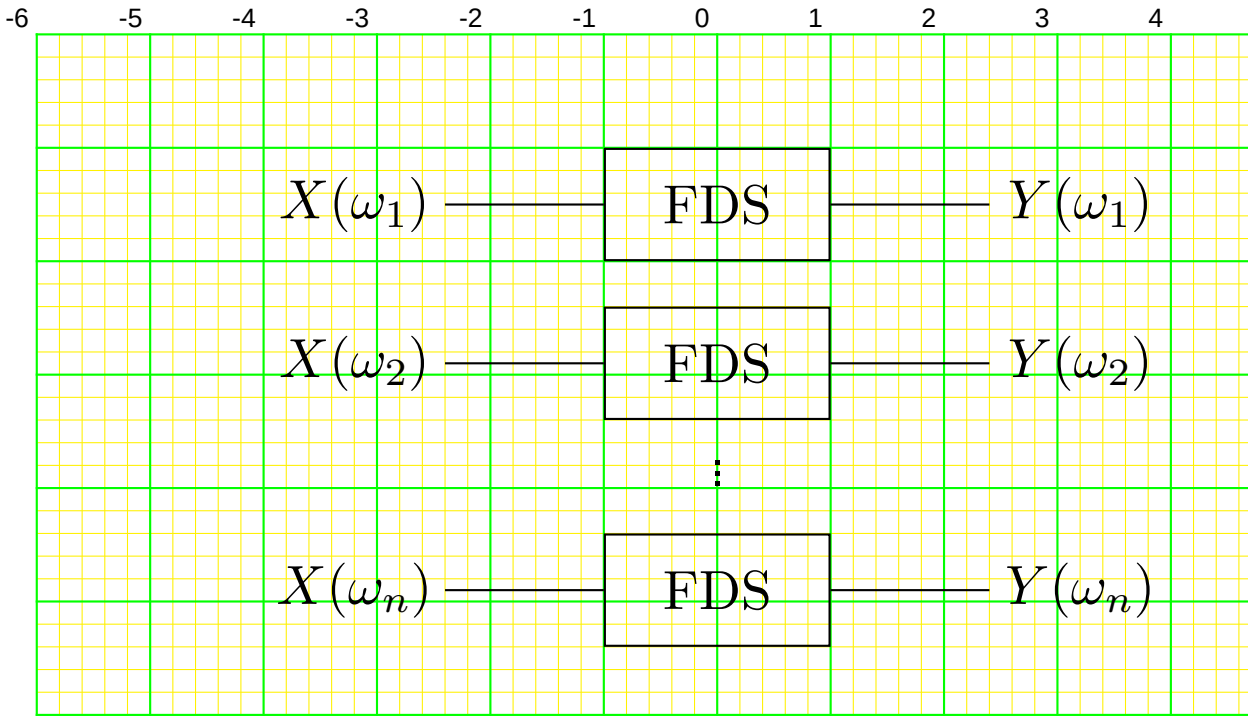
- Input: clean excitation at input requires an auxiliary simul
- Device: approximations required for material dispersion
- Output: overlap integrals must be repeatedly calculated *d*

simulation



- Fundamental problem: trying to use a time-domain solver
  frequency-domain solver.

- Additionally, no method to measure simulation error!



- Allow direct access to frequency-domain data.
- Take care of input and output fields outside of the simula
- Material dispersion explicitly defined.

- Simulation error well defined.

- Frequency-domain solver made possible by correct choice
  linear algebra algorithm.

- See: Wonseok Shin, Shanhui Fan, "Choice of the perfectl
  layer boundary condition for frequency-domain Maxwell's
  solvers", Journal of Computational Physics (January 2012

Interface

We want engineers and scientists to absolutely love usi
at the same time, to forget about it because it just wo

- No installation, just start up Matlab and

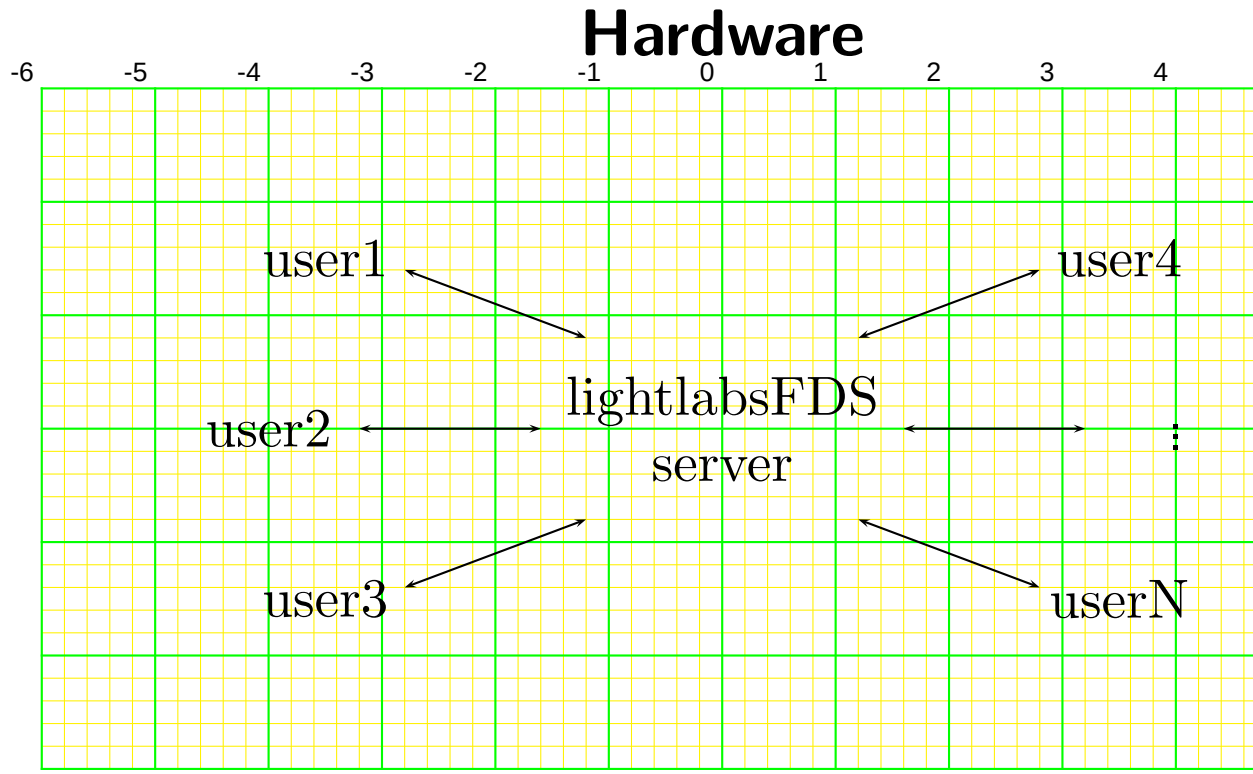
  ```
  >>> [E, H] = fds(omega, epsilon, J); % Done.
  ```

- Helper functions to

  – construct the optical structures (ϵ, μ),
  – define the input excitations (J), and
  – analyze and visualize the output fields (E, H, D, B)

  are all included and open-sourced.

- Additionally, lots of examples and tutorials.



- Centralized, shared, custom-tuned server able to deliver th performance of a large cluster

- Performance achieved via heavily optimized GPU code

- Multiple servers can be clustered with nearly 100% compu
  efficiency.

Current status

- Algorithm: Implemented on GPUs, still optimizing (Jesse)
- Interface: (Wonseok)
- Hardware: Prototype ordered and being built (Jesse)