

# Numerical modeling for nanophotonics design

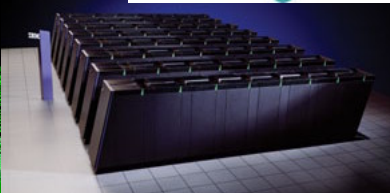
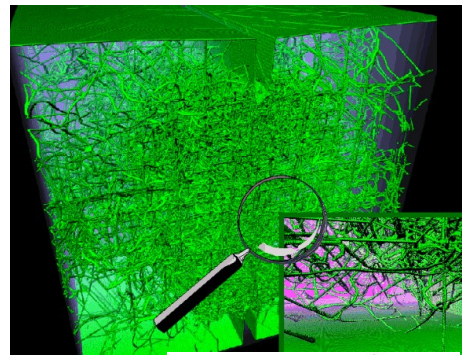
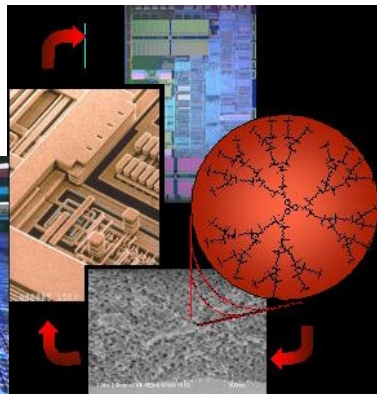
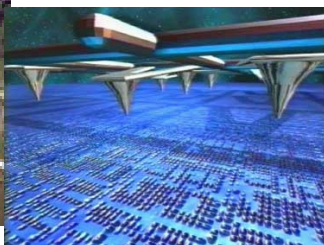
**Geoffrey W. Burr**

*IBM Almaden Research Center  
San Jose, California USA 95120*

- **Background**
  - IBM Almaden Research Center
  - holographic storage
- **Nanophotonics**
  - On-chip optical interconnect as motivation
  - Modeling & simulation tools
- **FDTD – the Finite-Difference Time-Domain technique**
  - Advantages & disadvantages
  - Could it be turned into a *design-optimization* tool?
- **Conclusions**

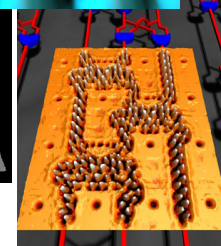
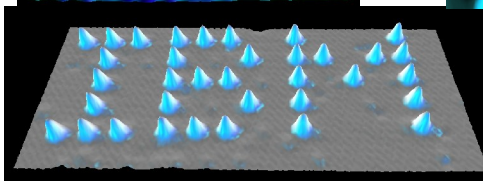
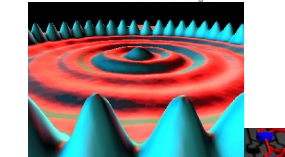
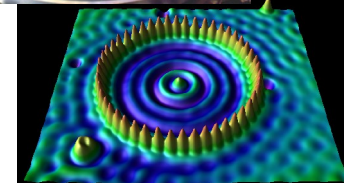
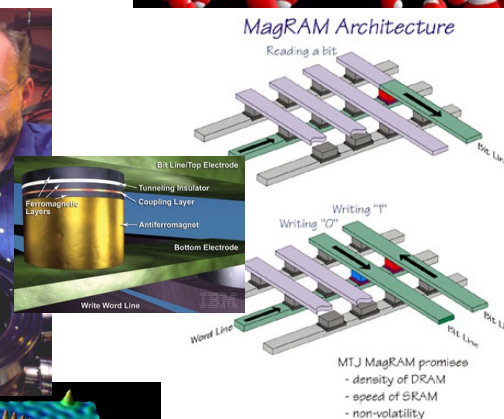
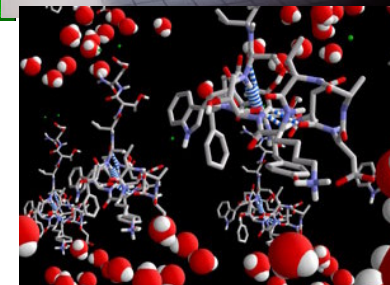
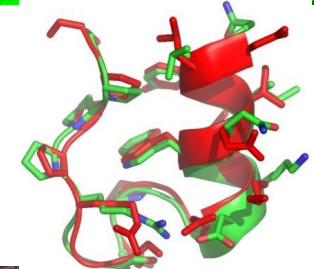
**IBM Almaden Research Center**





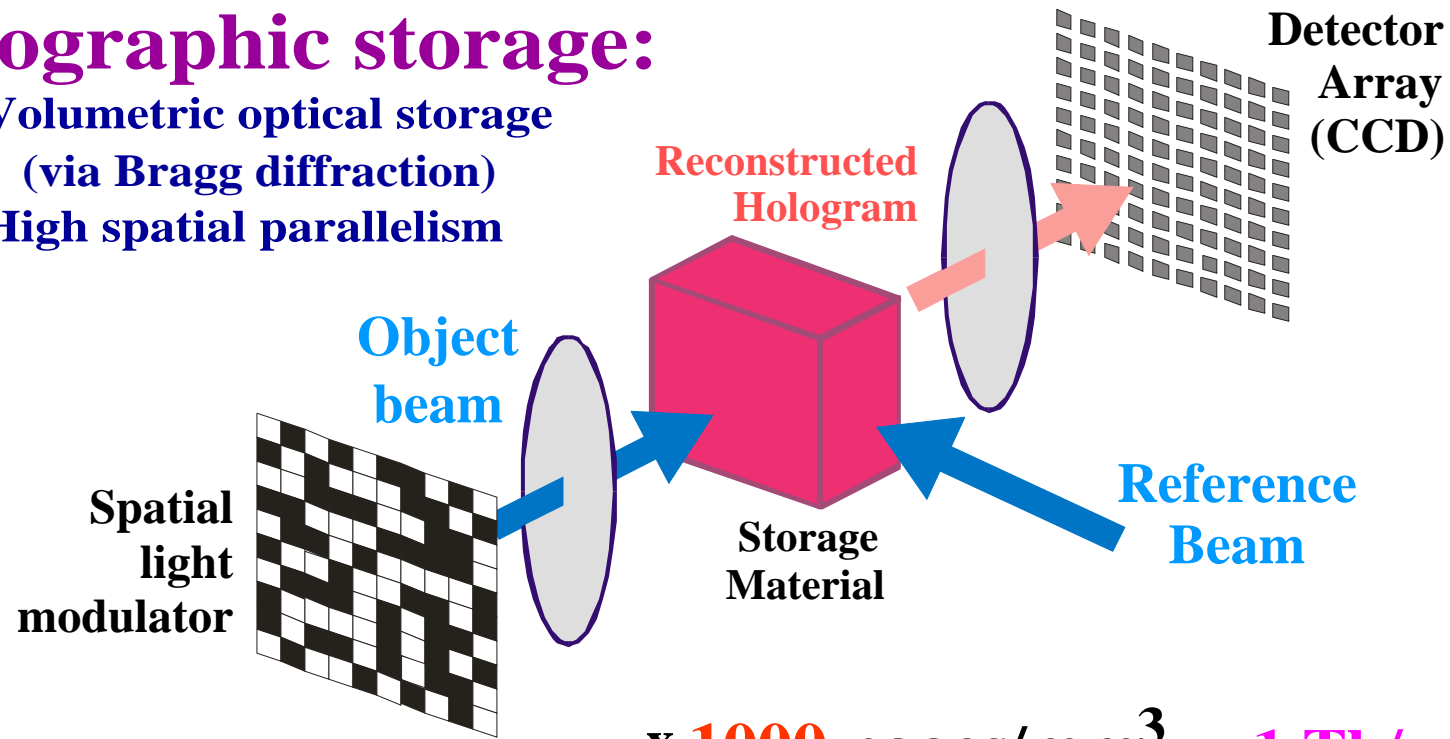
## Science & Technology (~80 researchers)

<b>Chemistry</b>	Dielectric materials, molecular electronics, lithography, polymer science, synthetic development laboratory
<b>Computational science</b>	Chemical kinetics, Molecular simulation, Life sciences simulation
<b>Data storage</b>	Magnetic materials & phenomena, novel recording technology
<b>Exploratory Technology</b>	Molecule cascades, nanophotonics, holographic storage
<b>Magnetism</b>	Magnetoelectronics & Spintronics, Magnetic theory/simulation
<b>Nanoscale science</b>	Atomic-scale microscopy & science, nano-fabrication
<b>Quantum information</b>	Quantum computing, quantum cryptography



# Holographic storage:

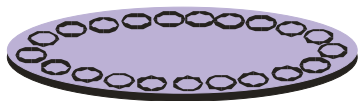
- Volumetric optical storage  
(via Bragg diffraction)
- High spatial parallelism



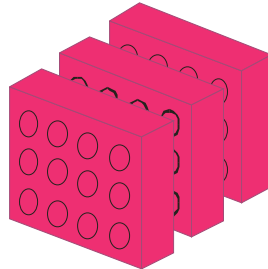
$10^6$  pixels/page  $\rightarrow$

$\times 1000$ pages/mm <sup>3</sup>	=	1 Tb/cm <sup>3</sup>
$\times 1000$ pages/sec	=	1 Gb/sec

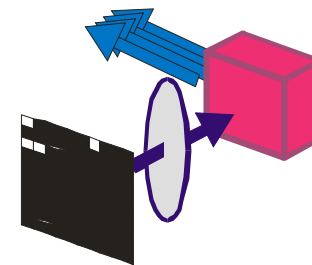
**WORM system**  
(moving media,  
for archival storage)



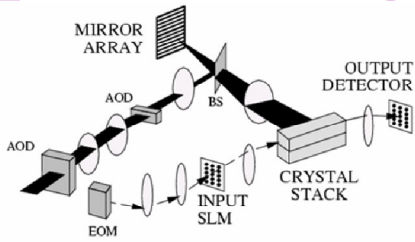
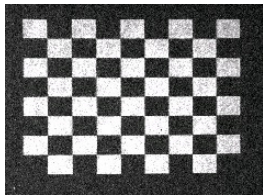
**Read/write system**  
(stationary media  
for low-latency storage)



**Content-addressable storage**  
(for rapidly searching  
massive databases)



# Holographic storage work



## DEMON 2

Holographic data storage at 250 Gbit/sq.in.

**IBM**  
Almaden Research Center

Geoffrey Burr, Hans Coufal, John Hoffnagle, Mike Jefferson, Mark Jurich, Roger Macfarlane, Bob Shelby

**DARPA** Holographic Data Storage Systems Consortium

Holographic data storage at 250 Gbit/sq.in.

**IBM**  
Almaden Research Center

High density, high speed

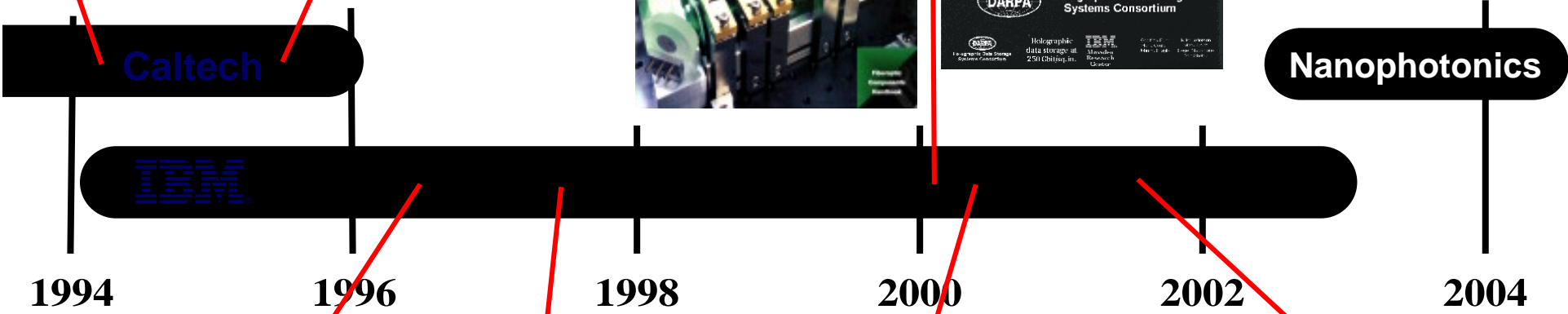
Holographic data storage

150 Gbit/sq.in. + 1 Gbit/sec

(Channel density) (Optical readout rate)

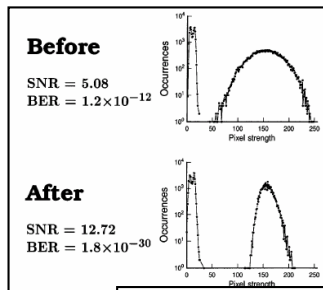
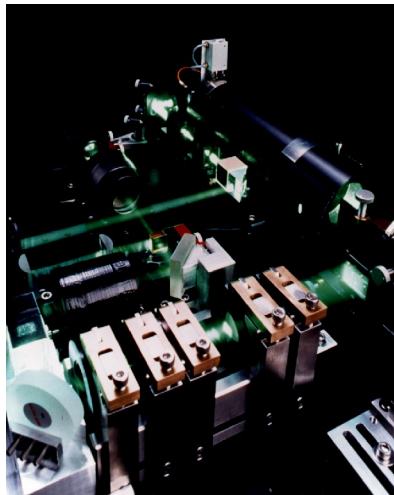
Hans Coufal, Geoffrey Burr, John Hoffnagle, Mike Jefferson, Mark Jurich, Roger Macfarlane, Bob Shelby

**M/#** 16x 10,000 holograms

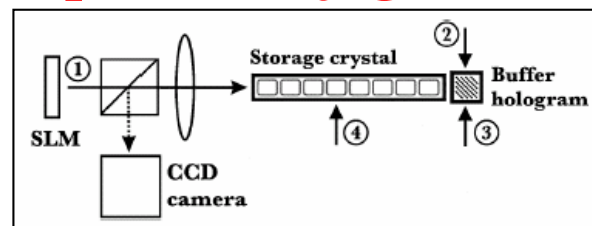


**Nanophotonics**

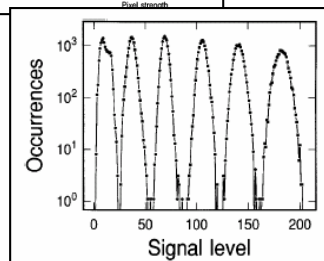
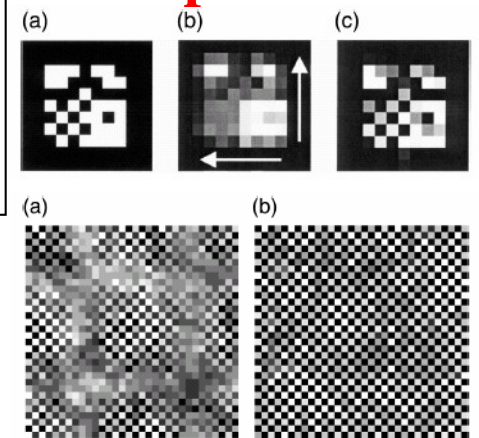
## DEMON 1



## phase-conjugation



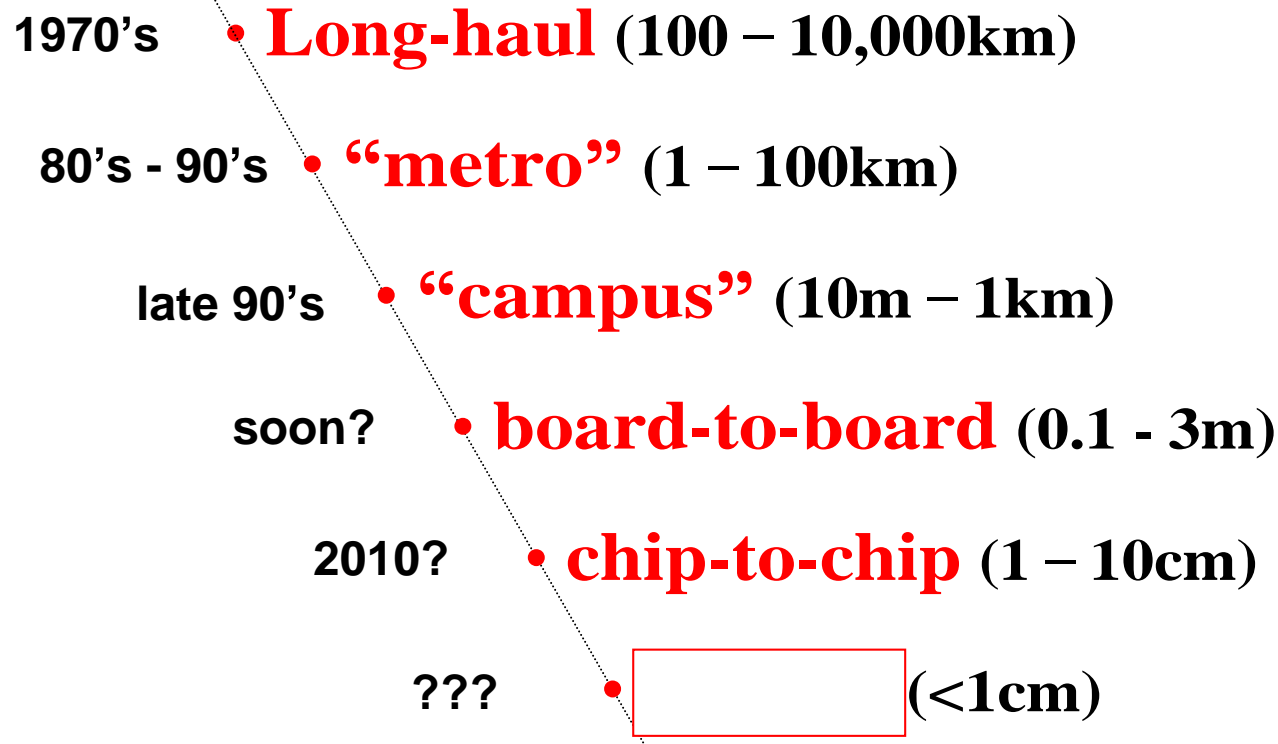
## shift compensation



List of slides

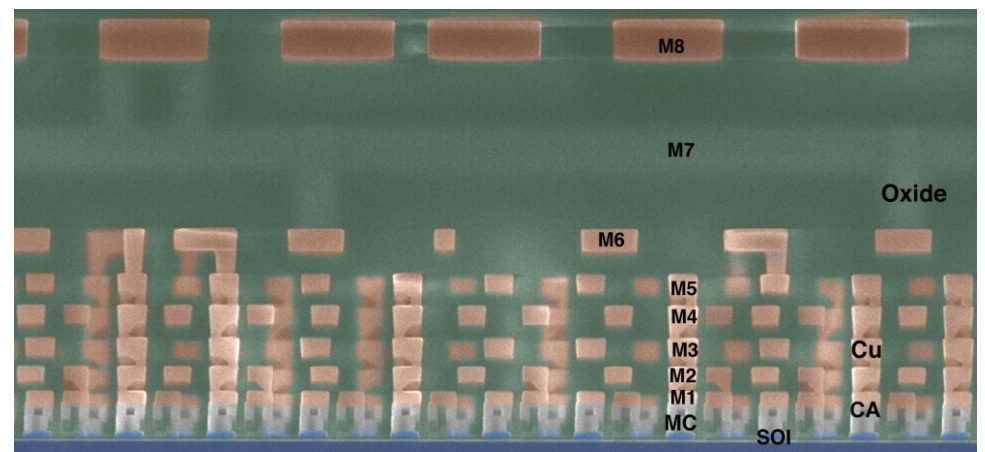
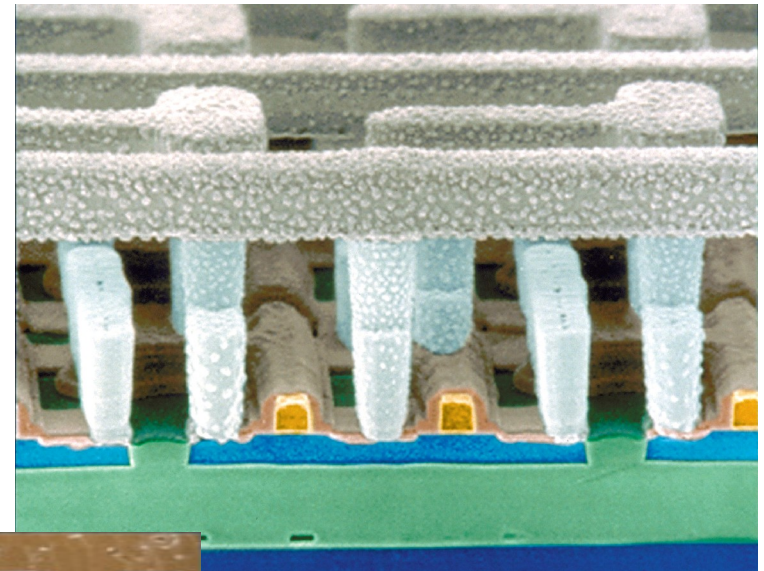
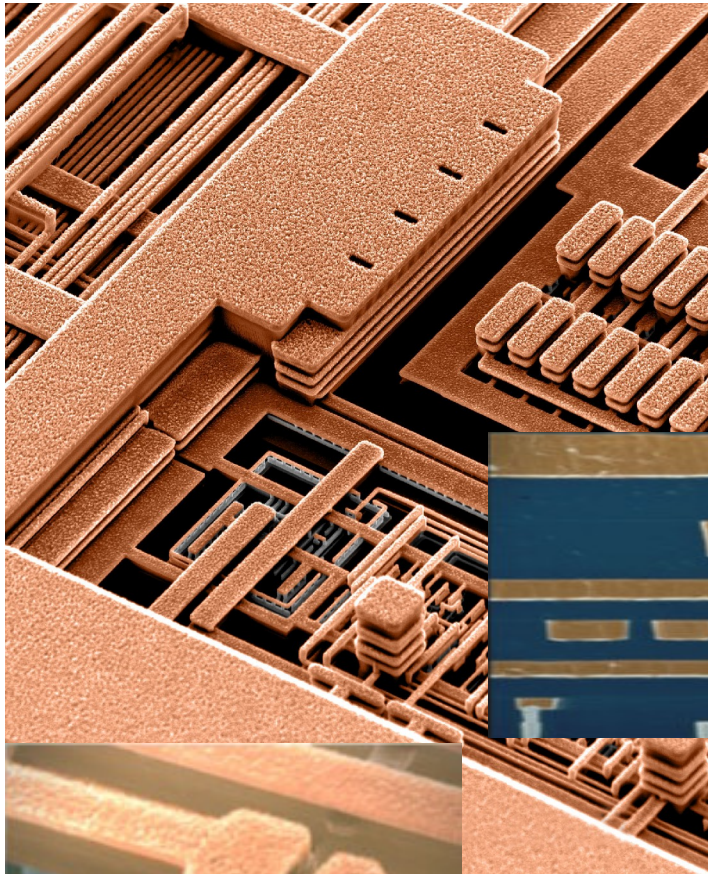
Conclusion

# Progression of optical interconnect



$\lambda \sim 1500\text{nm}$  – where's the nano?

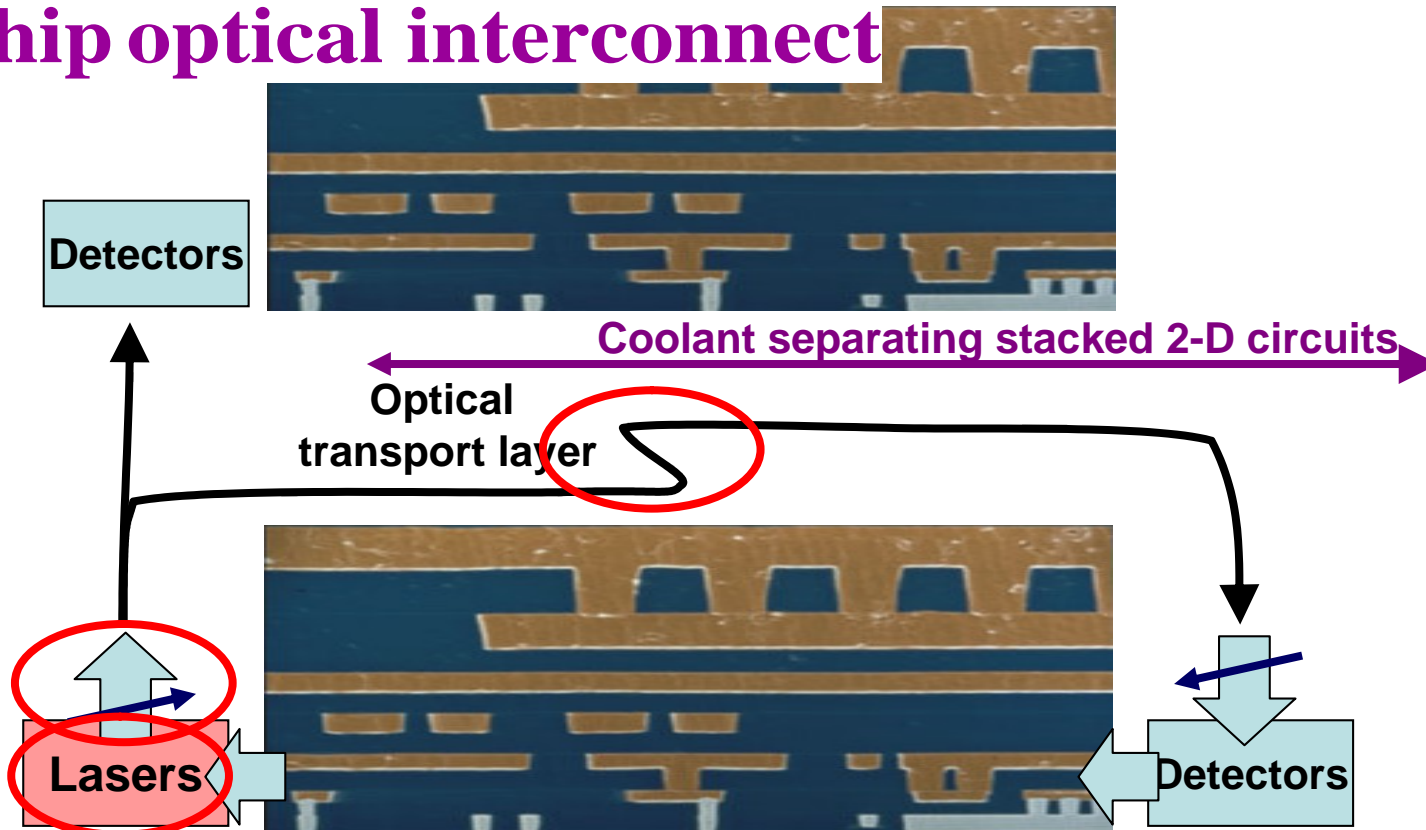
# On-chip optical interconnect



List of slides

Conclusion

# On-chip optical interconnect



## Challenges:

### Light source

- trick silicon into emitting
- flip-chip III-V devices
- light-source off-chip, modulators on-chip

### Optical mux-demux

- get many signals onto same optical path

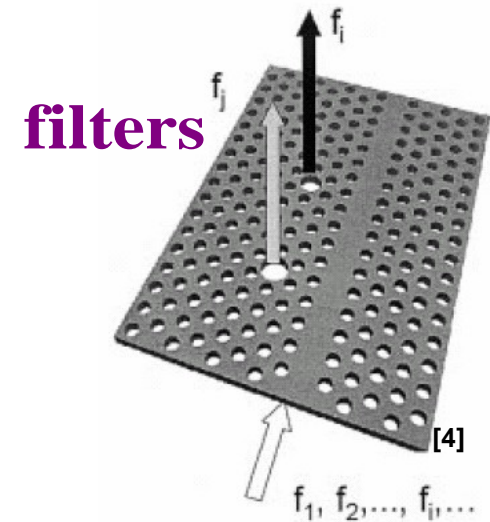
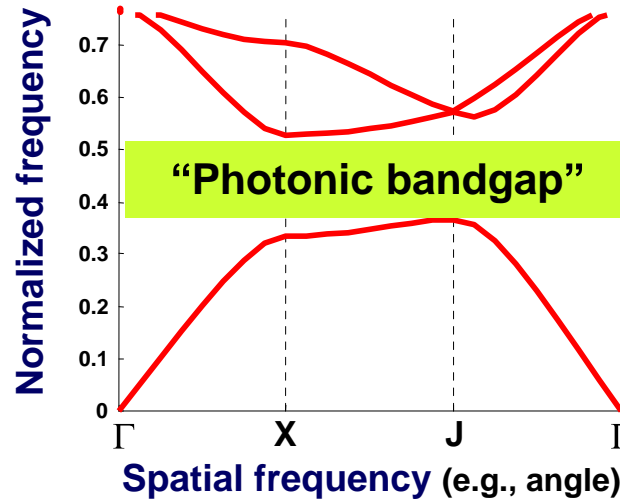
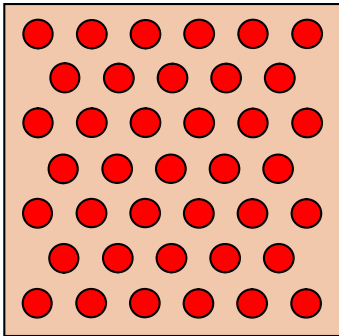
### Optical transport

- passive alignment, sharp bends, low-loss

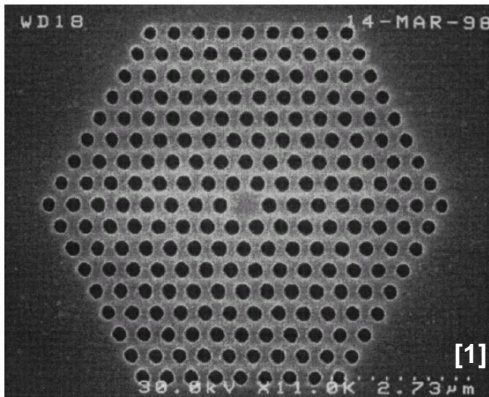
# Nanophotonics:

...the manipulation of light by nanoscale structure

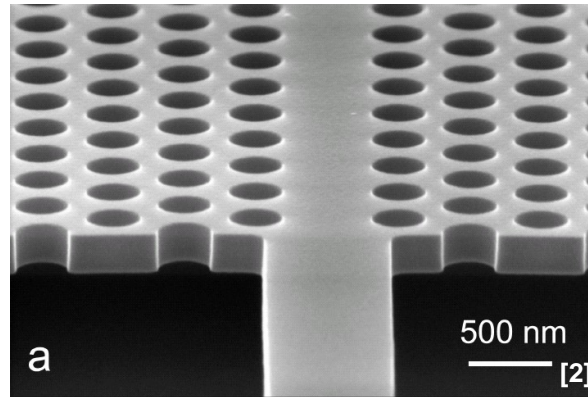
## Photonic crystals



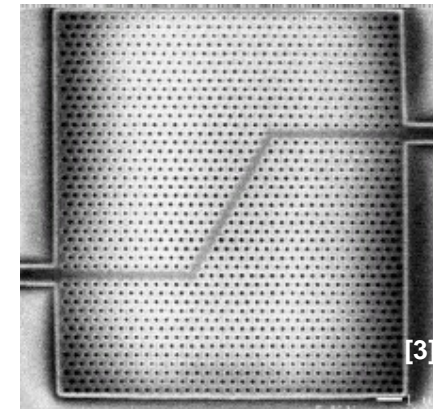
## lasers



## waveguides



## bends



[1] O. Painter, J. Vuckovic, and A. Scherer, *JOSA-B*, **16**(2), 275 (1999).

[2] S. J. McNab, N. Moll, and Y. A. Vlasov, *Optics Express*, **11**(22), 2927 (2003).

[3] E. Chow, S. Y. Lin, J. R. Wendt, S. G. Johnson, and J. D. Joannopoulos, *Optics Letters*, **26**(5), 286 (2001).

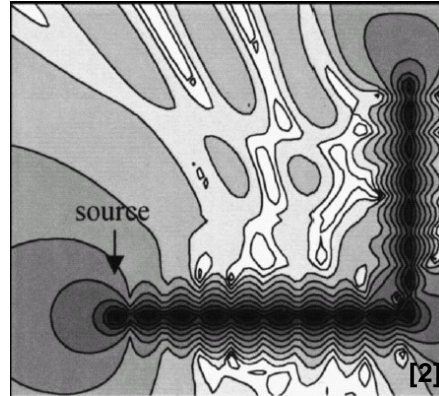
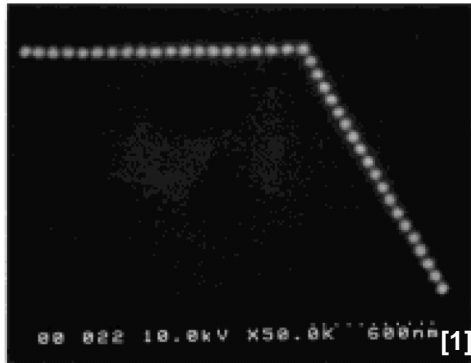
[4] S. Noda, M. Imada, M. Okano, S. Ogawa, M. Mochizuki, and A. Chutinan, *J. Quant. Electr.*, **38**(7), 726 (2002).

# Nanophotonics:

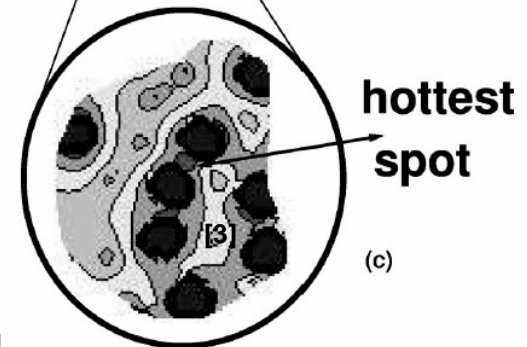
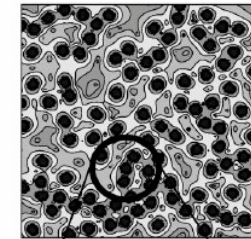
...the manipulation of light by nanoscale structure

## Plasmonics

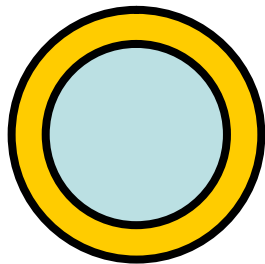
### plasmonic waveguides



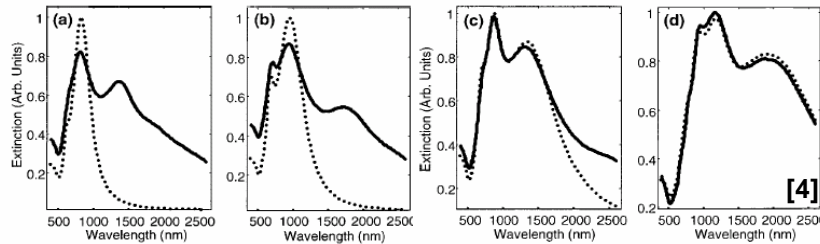
## Surface-enhanced Raman spectroscopy (SERS)



## Plasmon-resonance tuning



gold "nanoshells"



[1] S. A. Maier, M. L. Brongersma, P. G. Kik, S. Meltzer, A. A. G. Requicha, and H. A. Atwater, *Adv. Mater.*, **13**(19), 1501 (2001).

[2] S. A. Maier, M. L. Brongersma, and H. A. Atwater, *Appl. Phys. Lett.*, **78**(1), 16 (2001).

[3] P. Etchegoin, L. F. Cohen, H. Hartigan, R. J. C. Brown, M. J. T. Milton, and J. C. Gallop, *J. Chem. Phys.*, **119**(10), 5281 (2003).

[4] S. J. Oldenburg, J. B. Jackson, S. L. Westcott, and N. J. Halas, *Appl. Phys. Lett.*, **75**(19), 2897 (1999).

# So how to design such nanophotonic devices?

## 1) By trial-and-error?

Not likely, given the cost of fabrication & testing...

## 2) Modular approach

Fast but  
approximate  
methods

Beam-propagation

Frequency-domain

**FD TD** Slow but  
precise  
methods

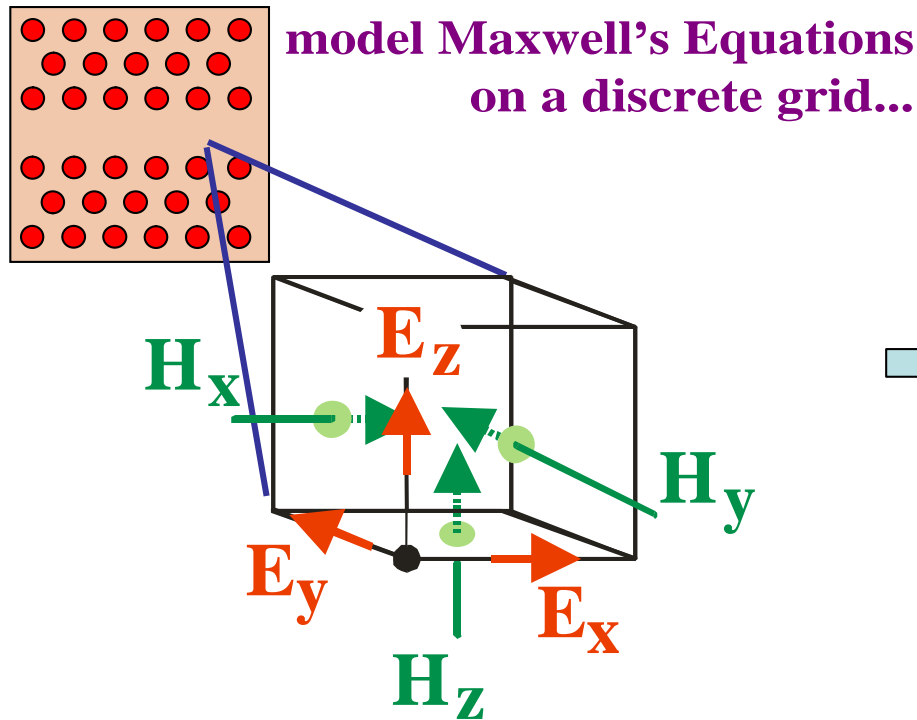
### Beam-propagation

- wave equation rather than Maxwell

### Frequency-domain methods

- can be rigorous, but (in general) ...
- ... assume structure is periodic and non-dispersive
- ... give little input concerning loss

# What is FDTD? “Finite-difference Time-Domain”



$$\mu \frac{\partial \vec{H}}{\partial t} = -\nabla \times \vec{E}$$

$$\epsilon_0 \epsilon_r \frac{\partial \vec{E}}{\partial t} = \nabla \times \vec{H} - \vec{J}_{source} - \sigma \vec{E}$$

$$\vec{E}_y += \frac{\Delta t}{\epsilon_r \epsilon_0} \left( \frac{H_z - H_z^{\text{previous cell}}}{\Delta x} \right)$$

## Why would we need it?

- *design-by-nanofabrication* is too expensive...

➔ need accurate modeling/simulation tools

- **FDTD is rigorous yet flexible (general-purpose tool)**

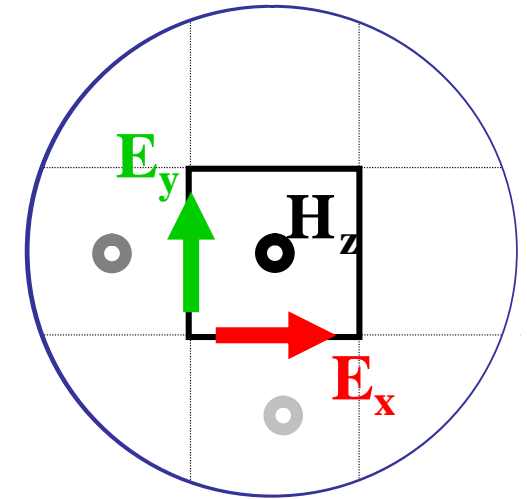
## A brief history

- 1966 – original paper by Yee
- late 70's, 1980s – used for microwave modeling
- early 90's – rapid development
- 1995 (2000) – textbook by Taflove/Hagness

# 'Leapfrog' update

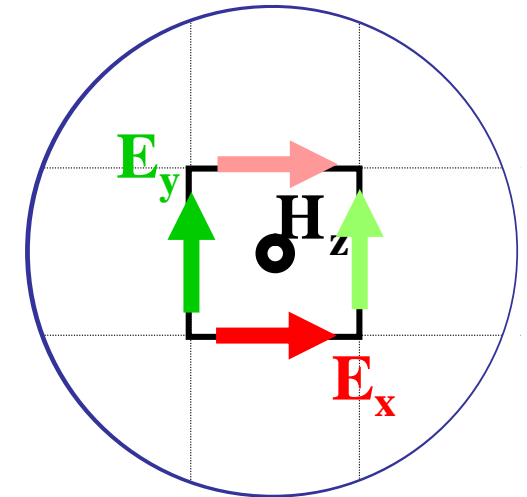
1) at time  $t$ : Update  $\mathbf{E}$  fields everywhere using spatial derivatives of  $\mathbf{H}$

$$\mathbf{E}_x += \frac{\Delta t}{\Delta y \epsilon_r \epsilon_0} \left( \mathbf{H}_z - \mathbf{H}_z^{\text{last } y} \right)$$
$$\mathbf{E}_y += \dots$$



2) at time  $t+0.5$ : Update  $\mathbf{H}$  fields everywhere using spatial derivatives of  $\mathbf{E}$

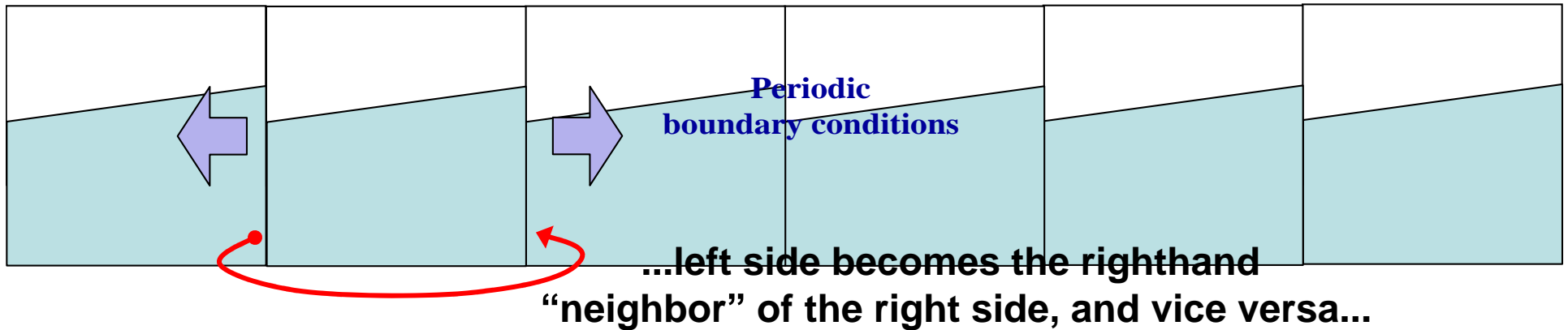
$$\mathbf{H}_z += \frac{\Delta t}{\mu} \left( \frac{\mathbf{E}_x^{\text{next } y} - \mathbf{E}_x}{\Delta y} + \dots \right)$$



- Every cell must get updated
- Accuracy requires  $\Delta x, \Delta y, \Delta z \rightarrow 0$
- Small  $\Delta x, \Delta y, \Delta z$  forces small  $\Delta t$  (Courant stability)
- What happens at edges?  $\rightarrow$  **Boundary conditions**

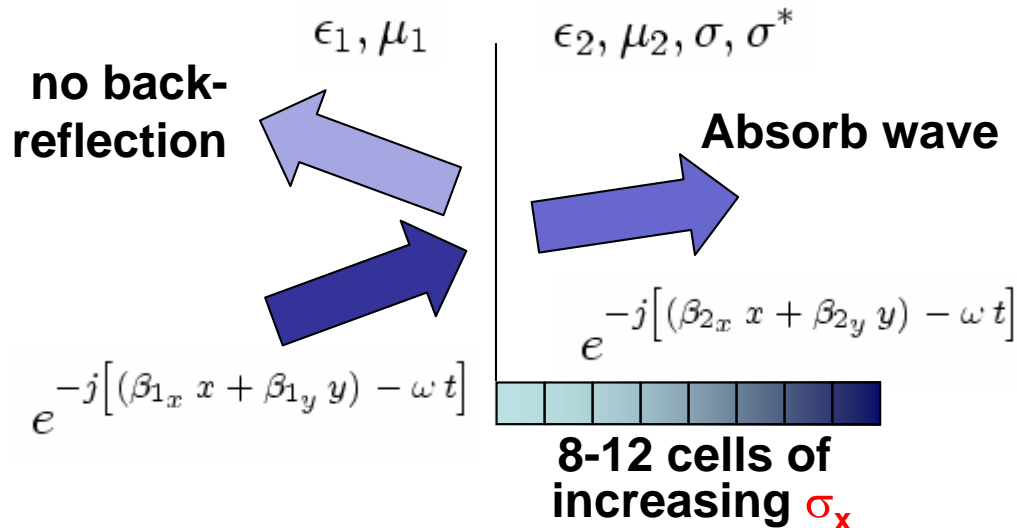
# Boundary conditions

## Infinite but periodic structures



## Isolated structures (by absorbing outgoing radiation)

### "Perfectly Matched Layer" (PML)



Back-reflection

$$\propto \eta_1 \cos \theta - \eta_2 \overset{\sigma_y = 0}{\downarrow} \cos \theta$$

$$\eta_1 = \sqrt{\frac{\mu_1}{\epsilon_1}} \quad \eta_2 = \sqrt{\frac{\mu_2}{\epsilon_2} \frac{(1 + \sigma_x^*/j\omega\mu_2)}{(1 + \sigma_x/j\omega\epsilon_2)}}$$

$$\epsilon_2 \frac{\partial E_x}{\partial t} + \sigma_y E_x = \frac{\partial H_z}{\partial y}$$

$$\epsilon_2 \frac{\partial E_y}{\partial t} + \sigma_x E_y = -\frac{\partial H_z}{\partial x}$$

$$\Rightarrow R(\theta) = \exp(-2 \sigma_x \eta_2 d \cos \theta)$$

# Advantages of Finite-Difference Time-Domain (FDTD)

- **algorithm is rigorous** — potential for arbitrarily high accuracy
- **can handle dispersive material including metals** (surface plasmons, etc.)
- **time-domain simulation**: one simulation can model broad frequency response
- **simple core algorithm + nearest-grid-neighbor dependencies** — amenable to parallelization
- **FDTD can support**
  - **finite or infinitely-periodic structures**
  - **arbitrary spatial arrangements of materials**
  - **input of pulsed, CW, or impulse waveforms**
  - **point-source, plane-wave, or mode-profile wavefronts**
  - **measurement of field, intensity, Poynting vectors**

# Disadvantages of Finite-Difference Time-Domain (FDTD)

- FDTD only becomes accurate as cell-spacing  $\rightarrow$  zero
- small cell spacing also mandates short time-steps (Courant stability)
- finite # of cells — boundary conditions at edges of the simulation are critical
- high frequency resolution requires many timesteps  
(Fourier-transform relationship)
- all simulation cells must be updated each timestep  
— computing just a portion is inefficient

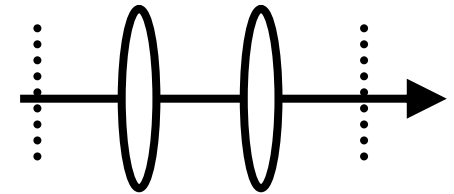
## The sum effect:

“to be confident you will get the right answer,  
3-D FDTD simulations must be large & slow”

But this means FDTD can only be a *design verification* tool,  
never a *design optimization* tool

# Ultimate goal: design optimization

In analogy to lens design software, it would be greatly desirable to have a nanophotonic *design optimization* tool..



A nanophotonics designer would simply:

1. **Build a base design** (“single-hole missing waveguide, ...”)
2. **Specify a performance metric** (“low-loss + wide-bandwidth, ...”)
3. **Establish constraints** (“must use LiNbO<sub>3</sub> at 1550nm, ...”)
4. **Identify free variables** (“hole size, slab thickness, defect type, ...”)
5. **Press the “go” button and wait (but not too long!)  
for the optimized design.**

Later on, one could introduce feedback from fabricated designs, add tolerancing, etc....

But in order for such a design optimization tool to be viable, the modeling of any one design iteration has to be **extremely fast** yet **reasonably accurate**.

## The sum effect:

“to be confident you will get the right answer,  
3-D FDTD simulations must be large & slow”

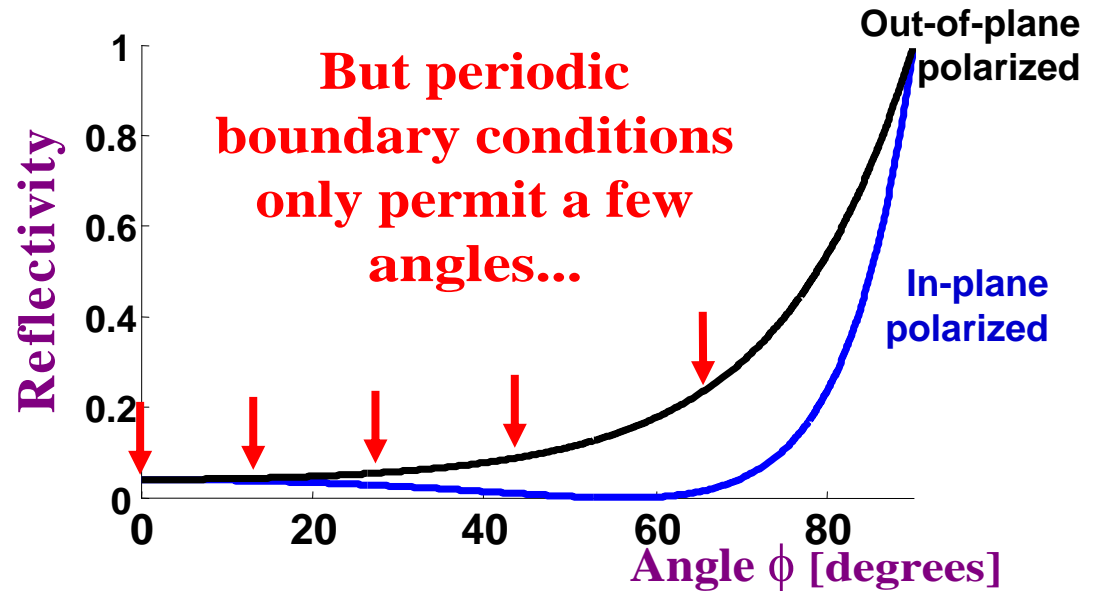
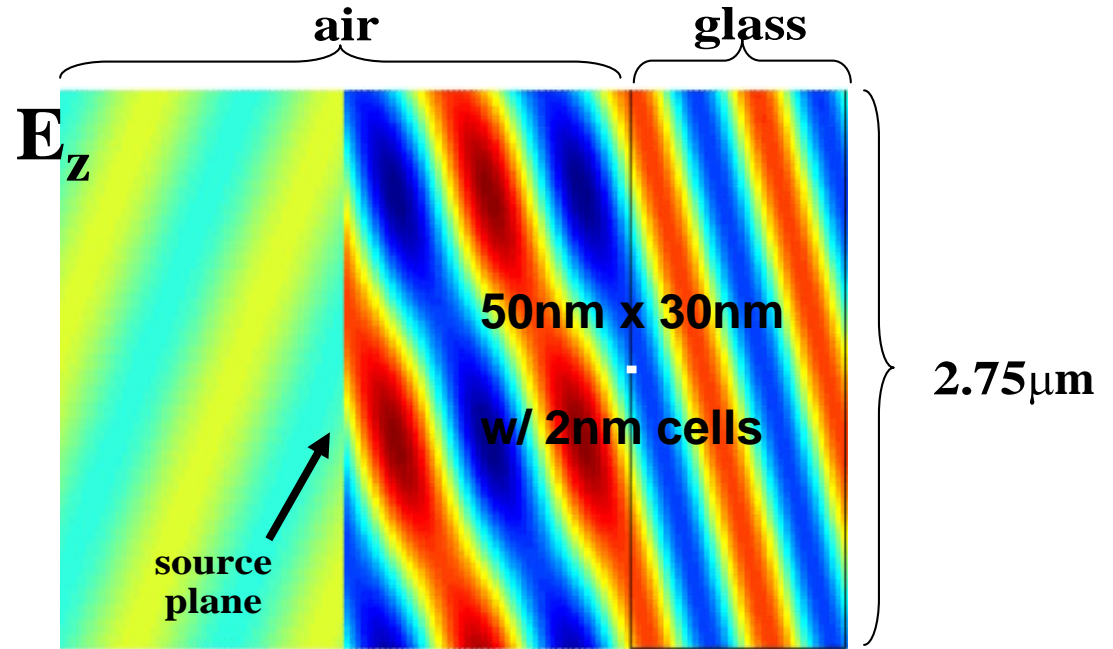
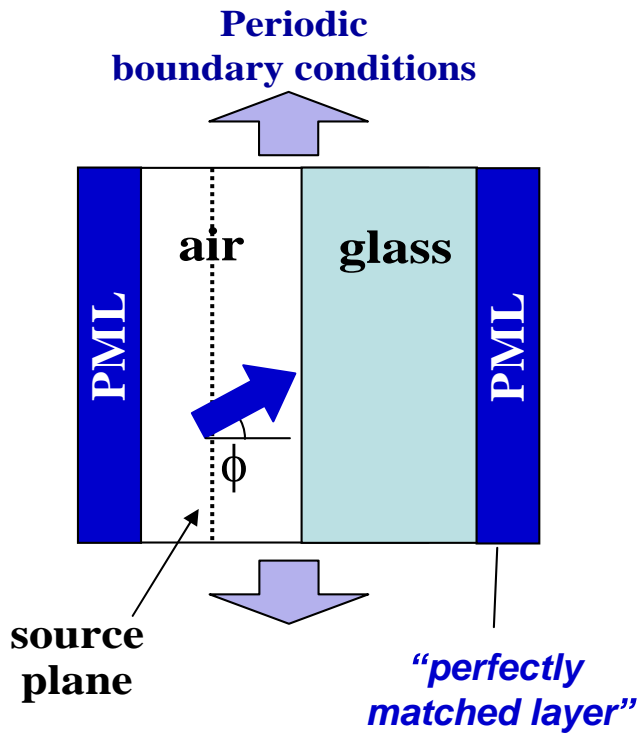
But this means FDTD can only be a *design verification* tool,  
never a *design optimization* tool

How to change this? • error mitigation  
• parallelization  
• careful design of numerical experiment

# Sample simulation

(validation)

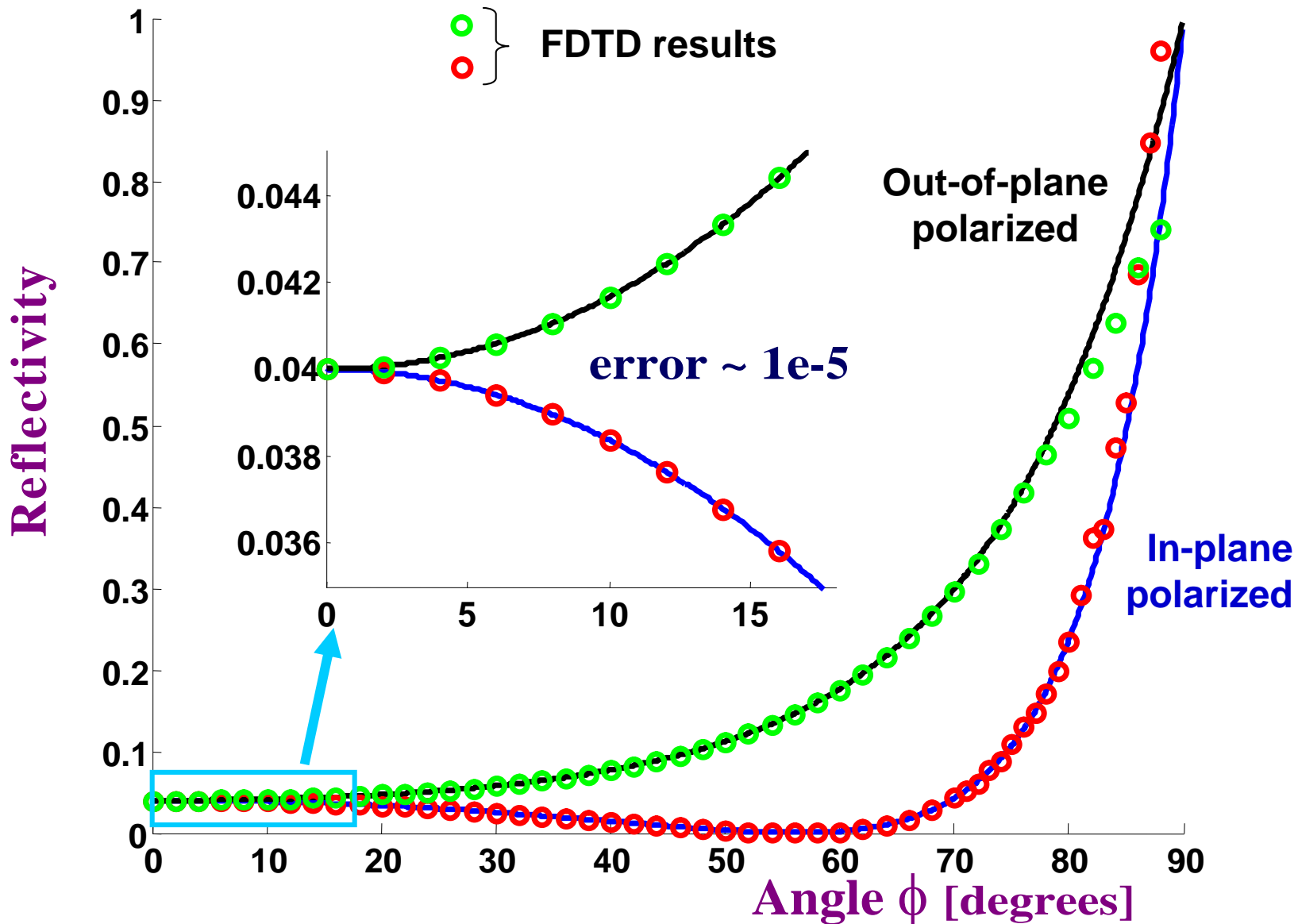
Fresnel reflection coefficients with FDTD



Solution: "Floquet" boundary conditions:

(build 2-part sim: 1 @  $\sin(\omega t)$ , 1 @  $\cos(\omega t)$  – exchange at boundary as  $\exp(\pm j \mathbf{k} \cdot \mathbf{L})$ )

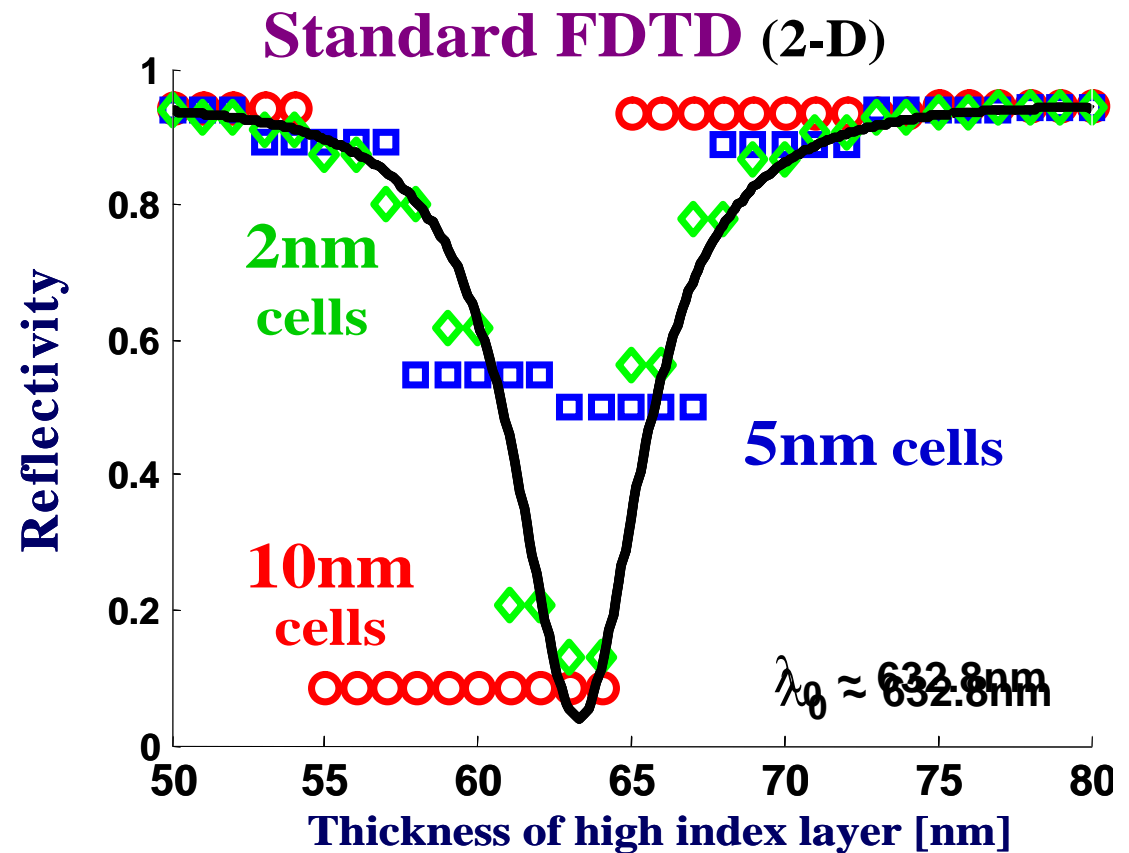
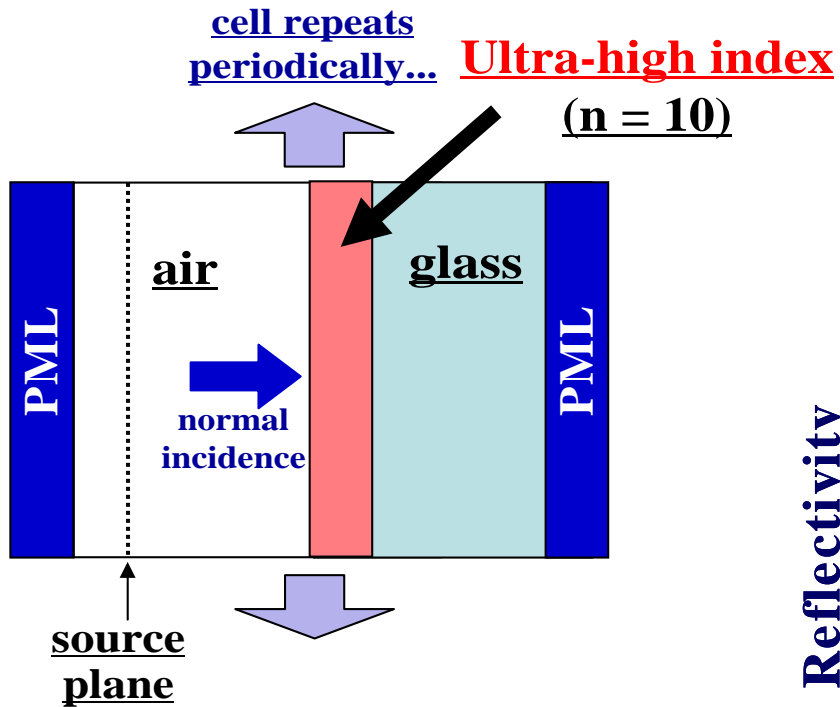
# Fresnel reflection coefficients



➔ **Error analysis: for 1% error, need cell-size  $< \lambda/50$**

# What happens with coarser gridding?

An extreme example: a thin-film of unphysically high index



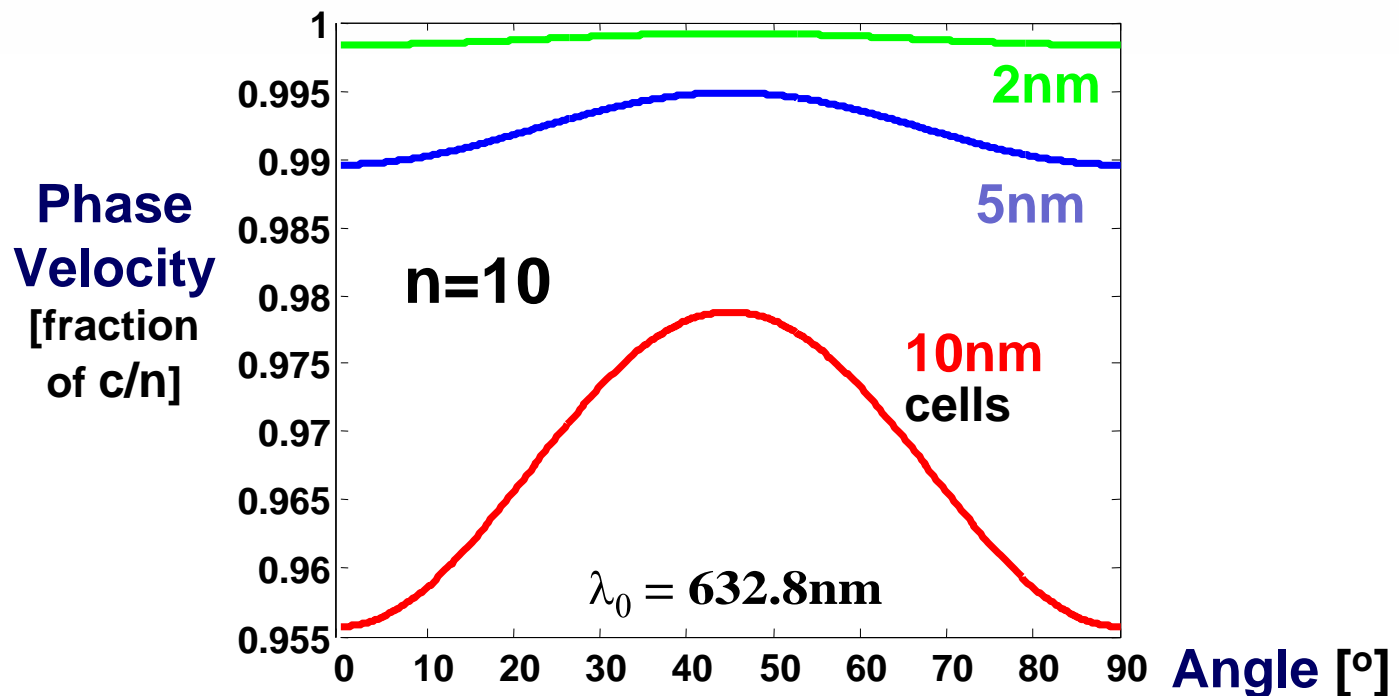
- small spatial features get lost
- get wrong answer AND wrong local minimum

# “Numerical dispersion”

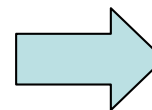
Dispersion relation should be:  $\left(\frac{\omega}{c}\right)^2 = (k_x)^2 + (k_y)^2 + (k_z)^2$

but instead is...

$$\left[\frac{1}{c\Delta t} \sin\left(\frac{\omega\Delta t}{2}\right)\right]^2 = \left[\frac{1}{\Delta x} \sin\left(\frac{\tilde{k}_x\Delta x}{2}\right)\right]^2 + \left[\frac{1}{\Delta y} \sin\left(\frac{\tilde{k}_y\Delta y}{2}\right)\right]^2 + \left[\frac{1}{\Delta z} \sin\left(\frac{\tilde{k}_z\Delta z}{2}\right)\right]^2$$



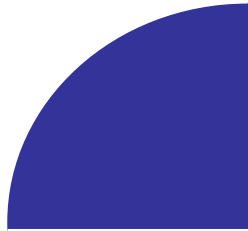
**Partial solution:** when index is high, adjust local  $\mu_0$  and  $\epsilon_0$  (increasing speed of light w/o affecting  $\eta$ )



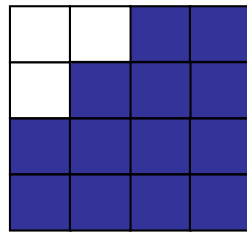
Only an exact fix for one angle & frequency...

# “Staircasing” of small spatial features

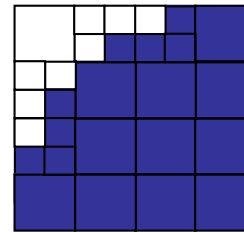
We really want to simulate a smooth interface between regions of  $\epsilon_1$  and  $\epsilon_2$ ...



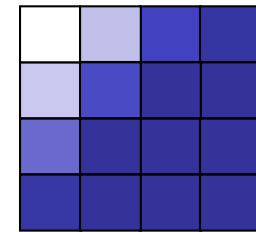
...but the discretization in FDTD gives a “staircase” effect.



Using small grid cells only where we need them sounds ideal, but the logistics are unpleasant...



...but we can get a similar effect by using an *effective*  $\epsilon$  in each cell.



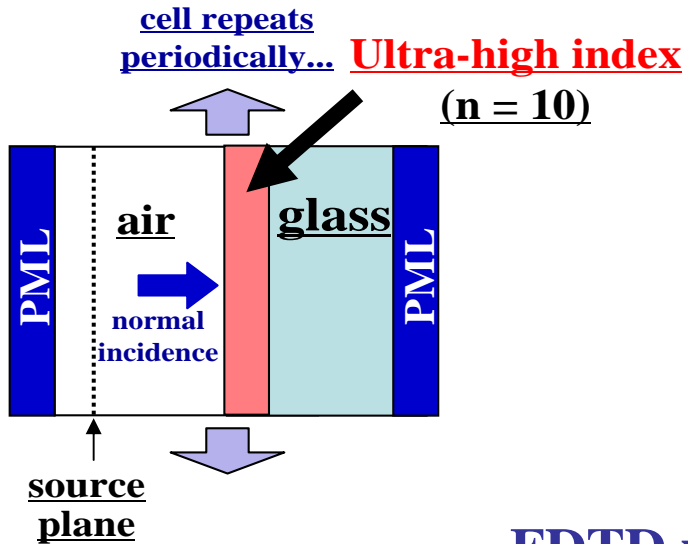
Each E-field component gets its own effective  $\epsilon$ , which makes the most sense in terms of the field-update equations.

$$\mathbf{E}_y += \frac{\Delta t}{\epsilon_r \epsilon_0} \left( \mathbf{H}_z - \mathbf{H}_z^{\text{previous cell}} \right) / \Delta x$$

$$\mathbf{E}_x += \frac{\Delta t}{\epsilon_r \epsilon_0} \left( \mathbf{H}_z - \mathbf{H}_z^{\text{previous cell}} \right) / \Delta y$$

G. W. Burr

# Returning to our artificial example...



FDTD with field-based effective  $\epsilon$

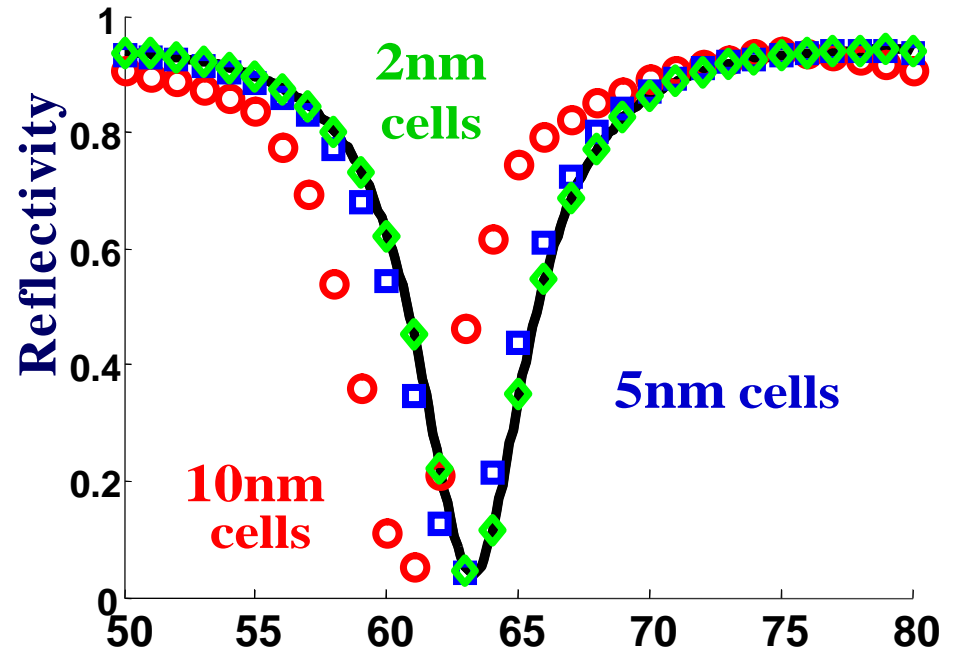
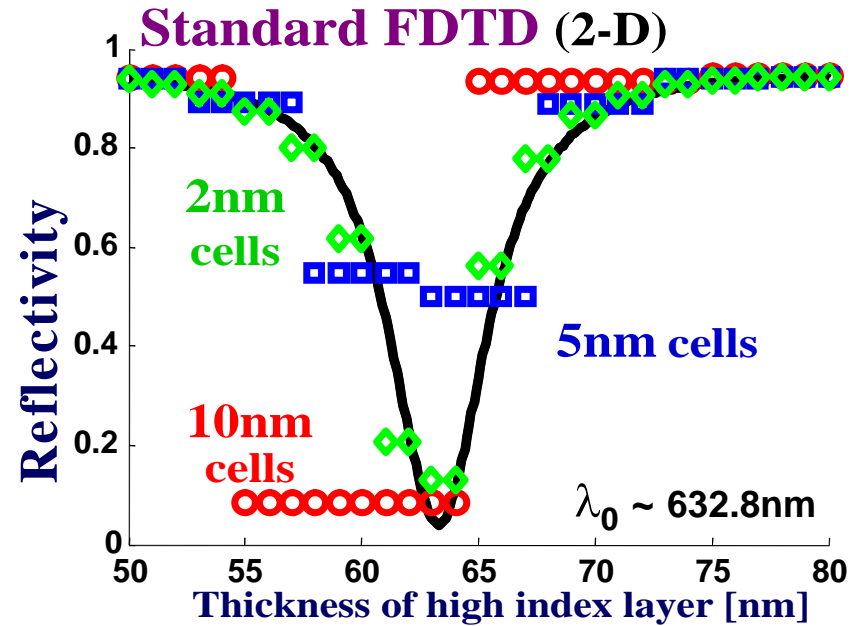
## Despite coarse gridding...

- small spatial features get modeled
- get the correct answer AND the right local minimum

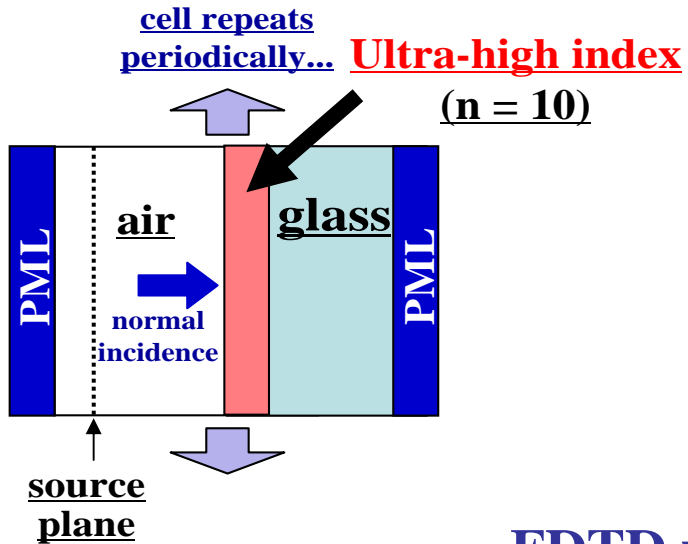
G. W. Burr

List of slides

Conclusion



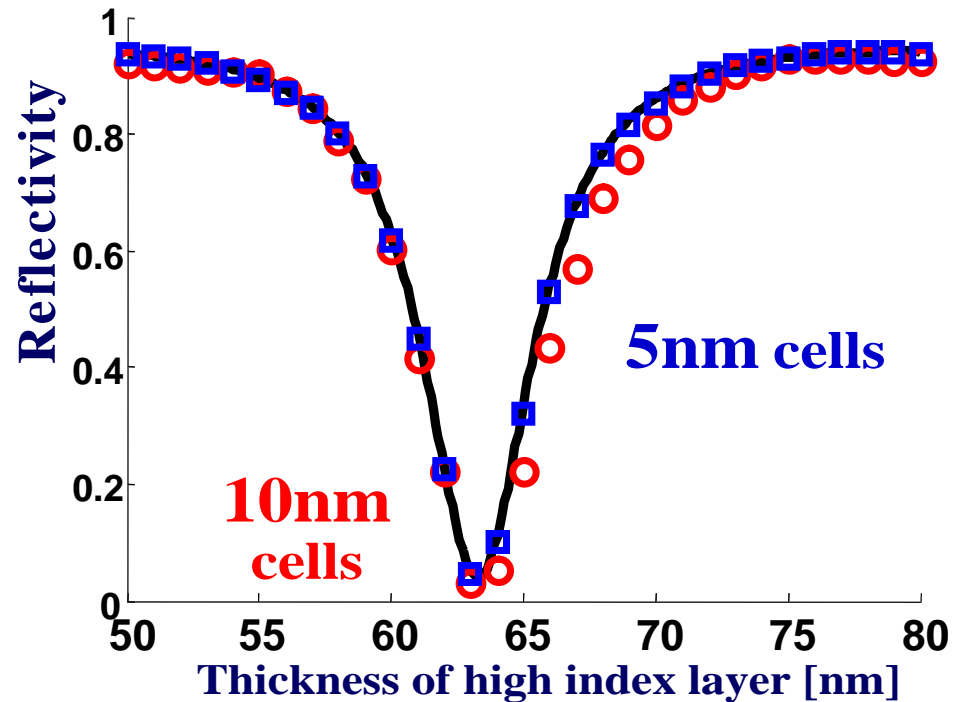
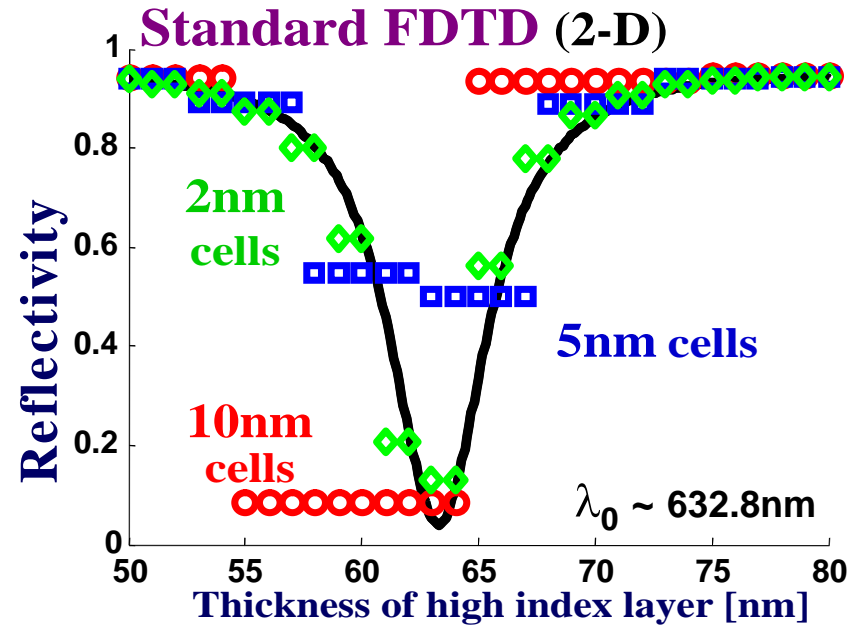
# Returning to our artificial example...



FDTD with field-based effective  $\epsilon$  and  $(\epsilon_0, \mu_0)$  corrected for numerical dispersion →

## Despite coarse gridding...

- small spatial features get modeled
- get the correct answer AND the right local minimum

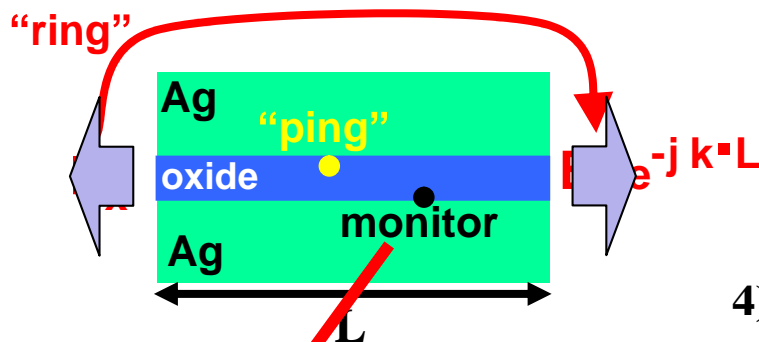


# Measuring frequency response

Example: Mapping out a dispersion diagram...

1) Define unit cell

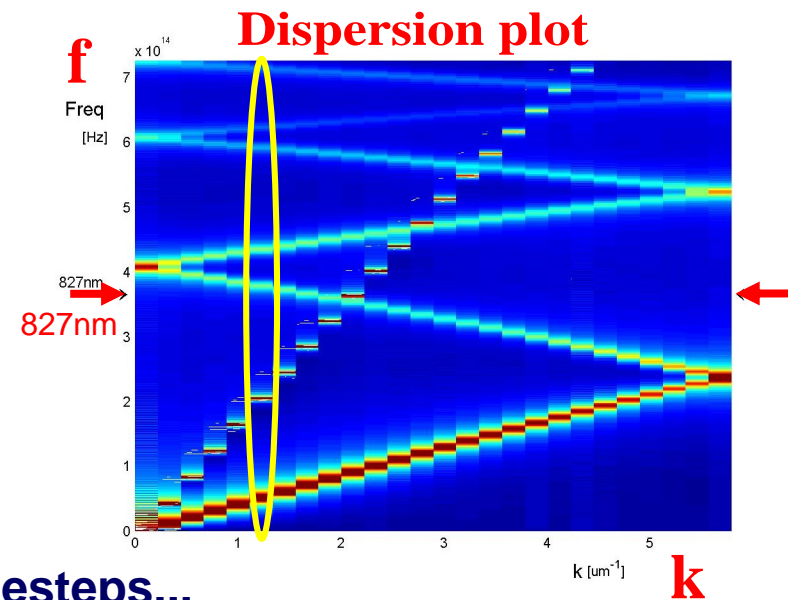
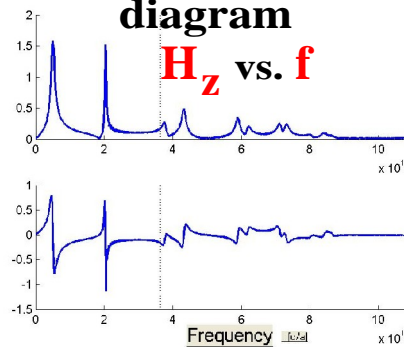
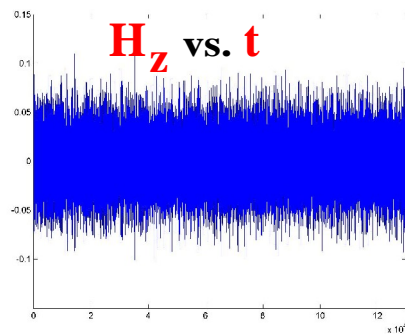
2) Ping it with an impulse



3) Set k-vector at boundaries

4) monitor at non-symmetry point

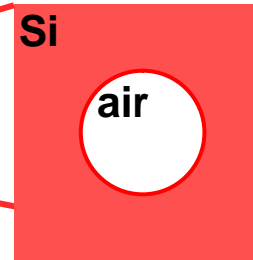
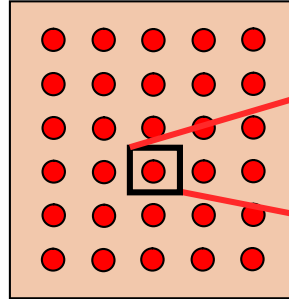
5) FFT becomes a column of the dispersion diagram



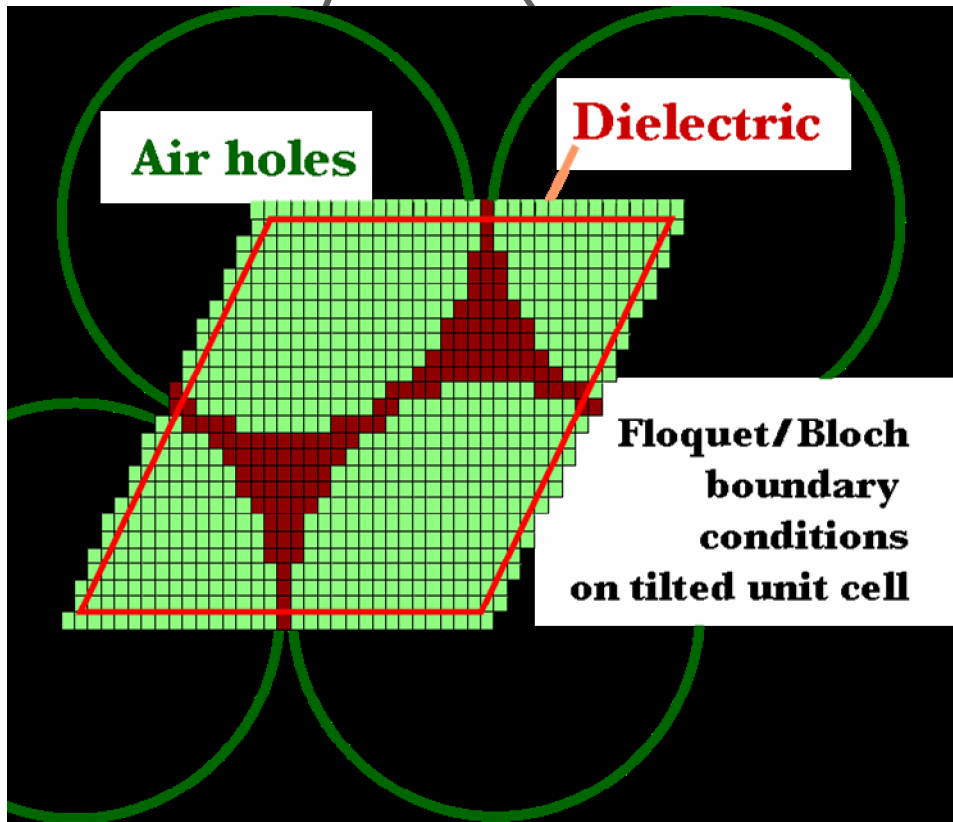
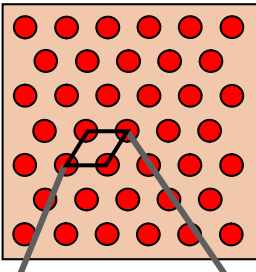
• For good frequency resolution, need lots of timesteps...

# Example: photonic crystal band-diagrams

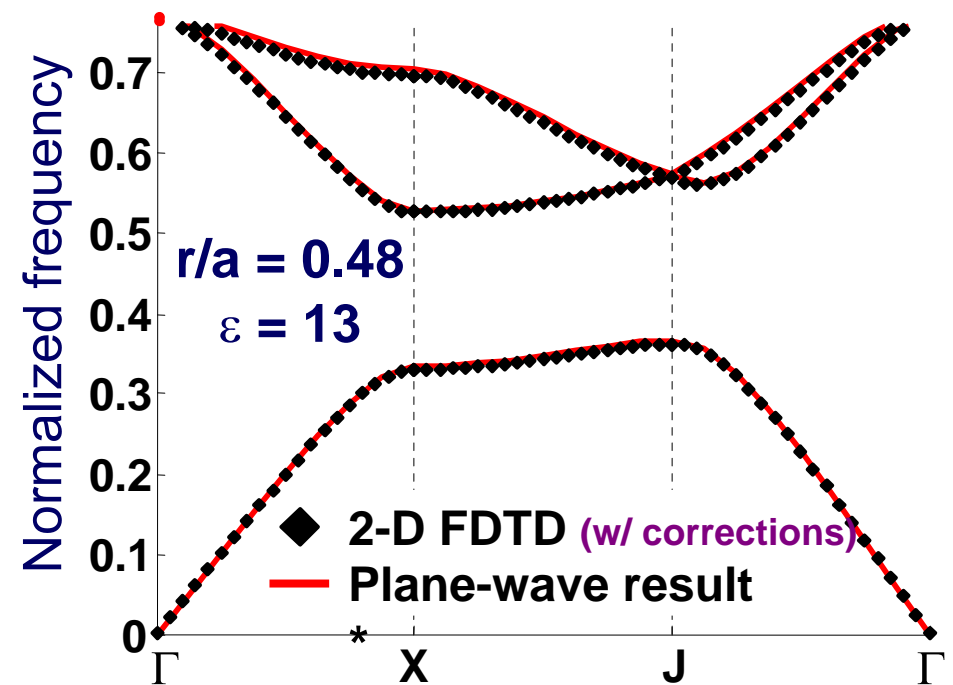
Square Lattice



Triangular Lattice

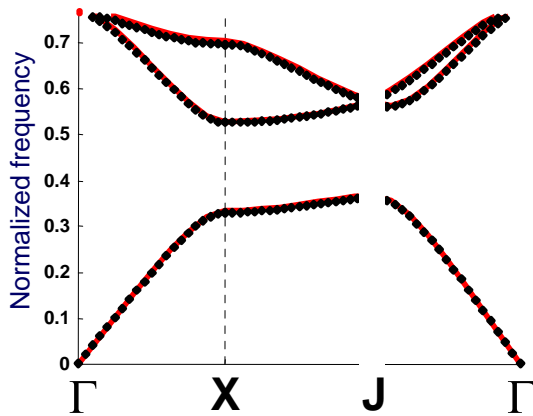
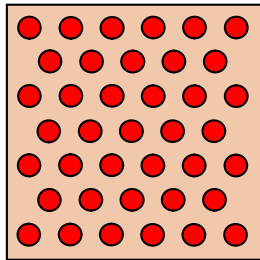


Triangular Lattice – TE bands



\* MIT "Photonic bands" code – thanks to Bob Shelby of IBM Almaden for his help...

# Triangular Lattice

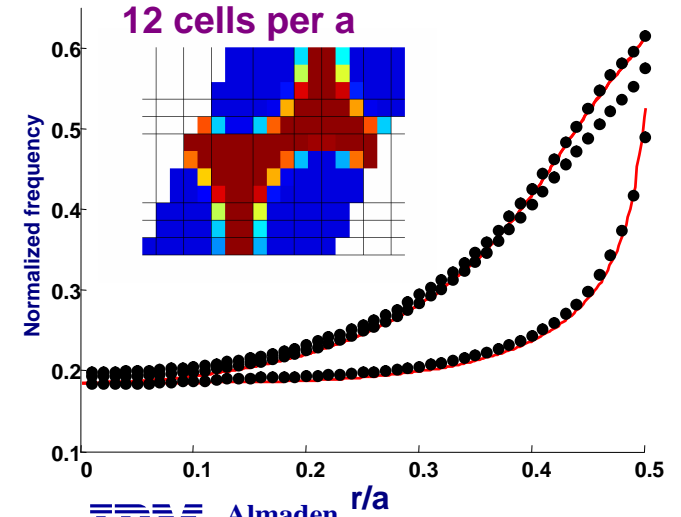
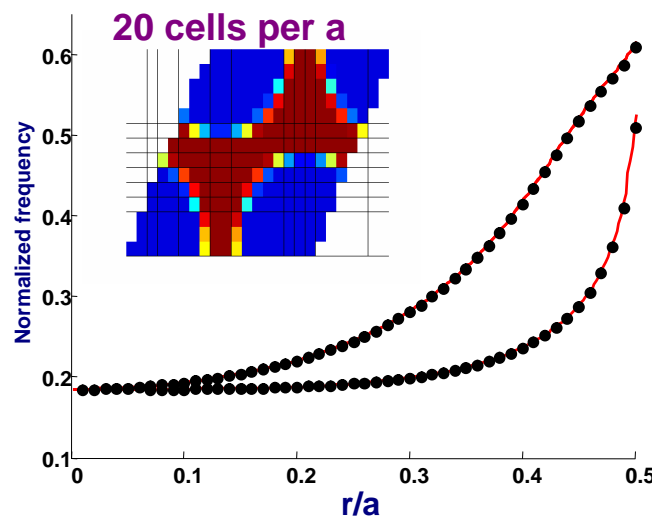
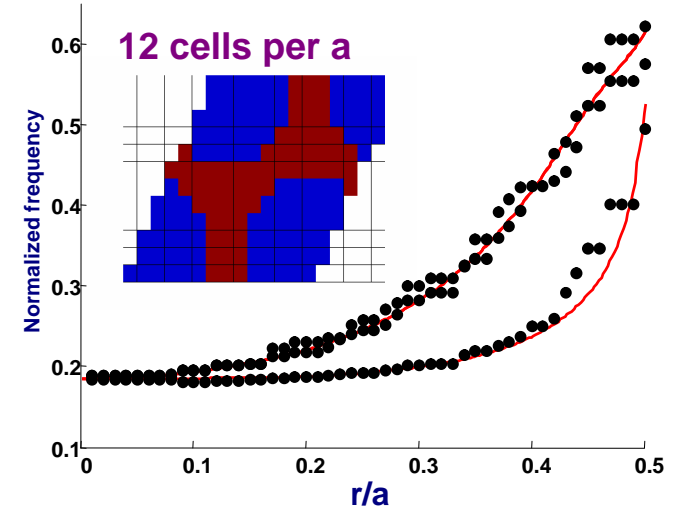
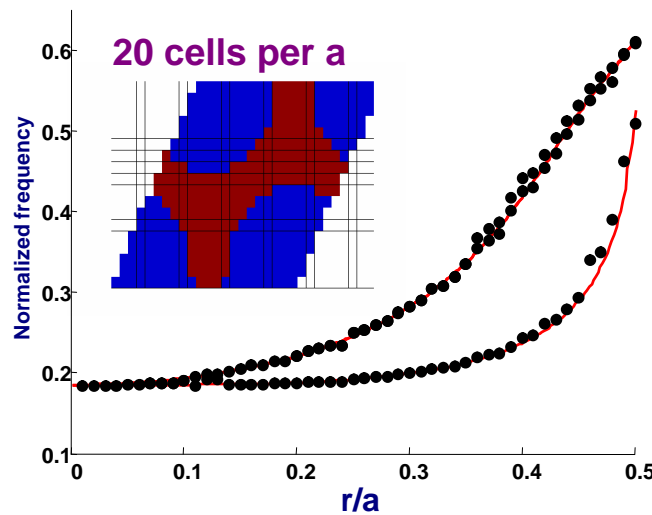


At constant  $r/a = 0.48$

...if we vary  $r/a$  at constant K-vector

improved FDTD

# Standard FDTD

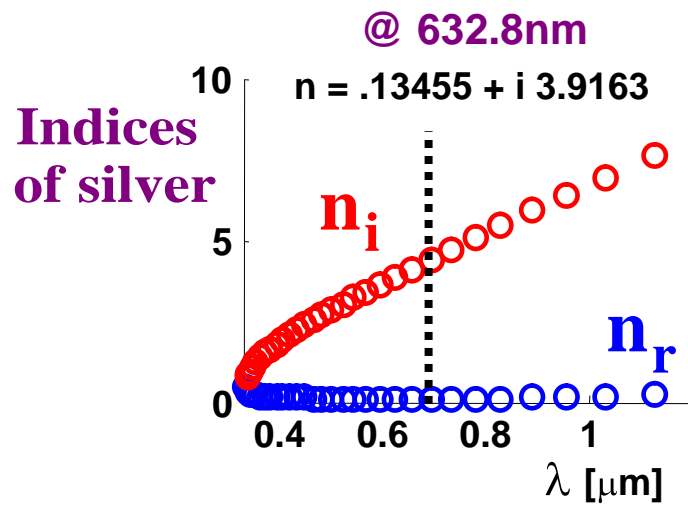


# Metals

**FDTD cannot model materials with  $\text{Re}\{\epsilon\} < 1$**

(such as silver at visible wavelengths)

**except by also modeling material dispersion.**

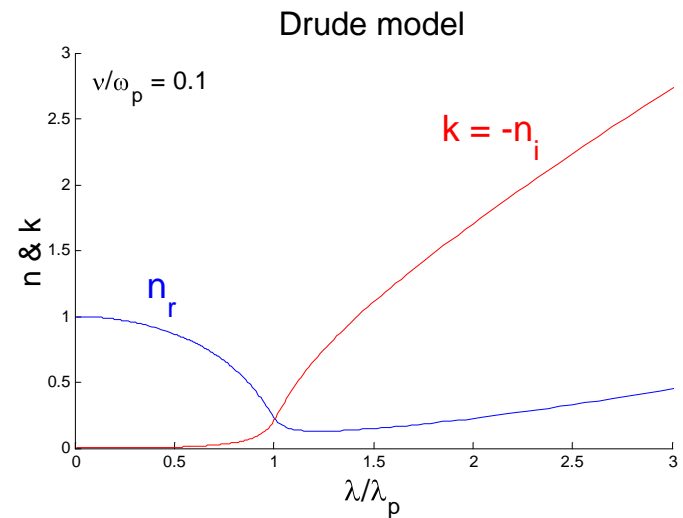


- extra variables & update rules at each metal/dispersive cell

$$\epsilon_\infty \frac{\partial \vec{E}}{\partial t} = \nabla \times \vec{H} - \vec{J}_p$$

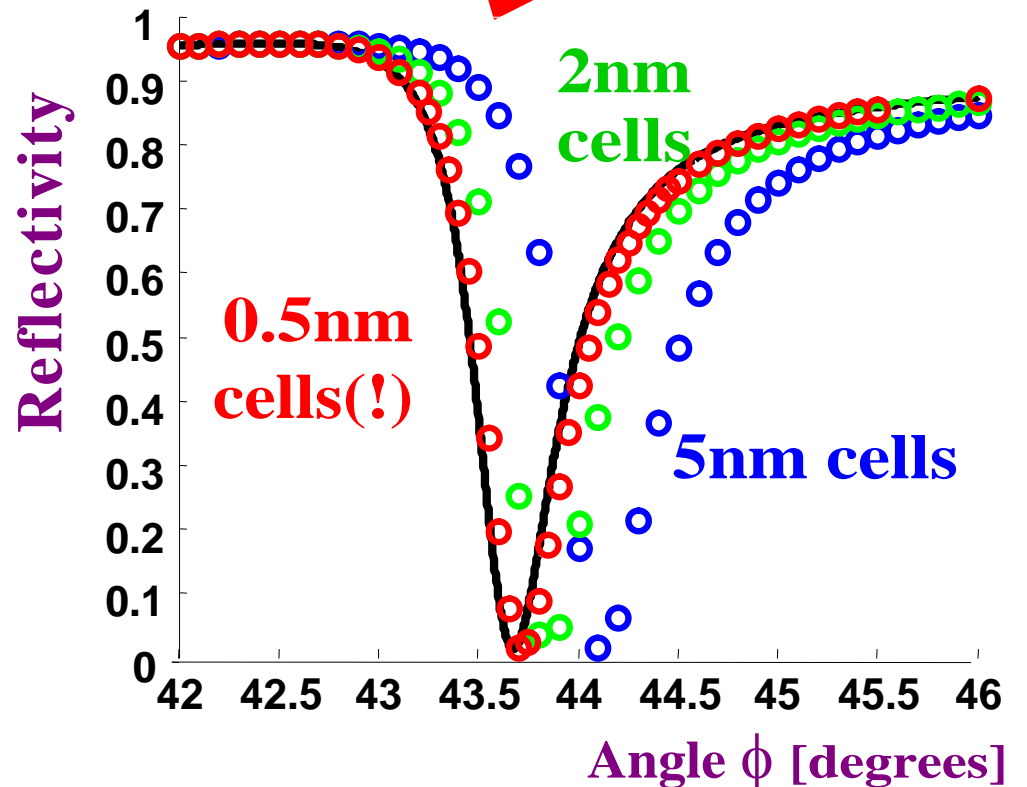
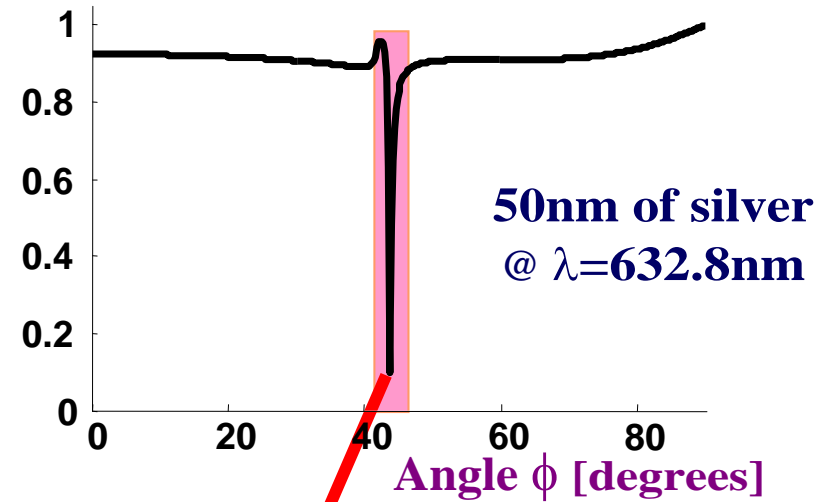
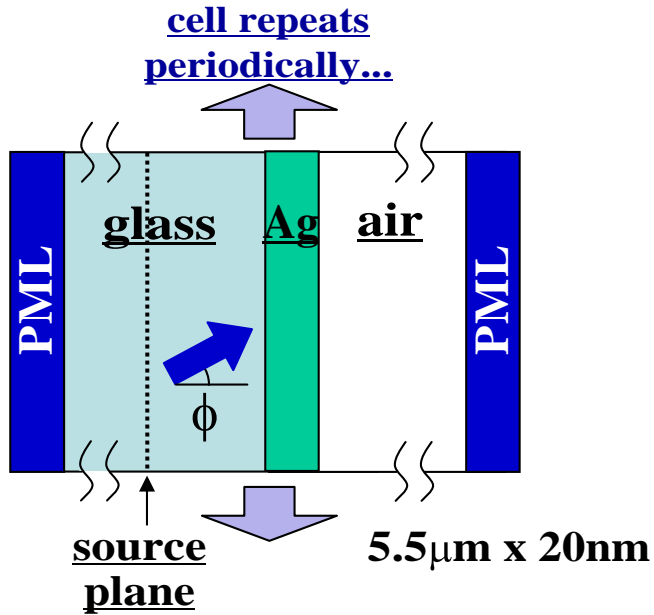
$$\frac{\partial \vec{P}}{\partial t} = \vec{J}_p$$

$$\frac{\partial \vec{J}_p}{\partial t} + \vec{J}_p = f(\vec{E}, \vec{P}, |E|^2, \text{etc.})$$



# Metals - verification

Plasmon resonance is extremely sharp:



## The sum effect:

“to be confident you will get the right answer,  
3-D FDTD simulations must be large & slow”

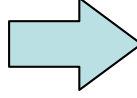
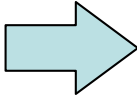
But this means FDTD can only be a *design verification* tool,  
never a *design optimization* tool

How to change this? ✓ error mitigation

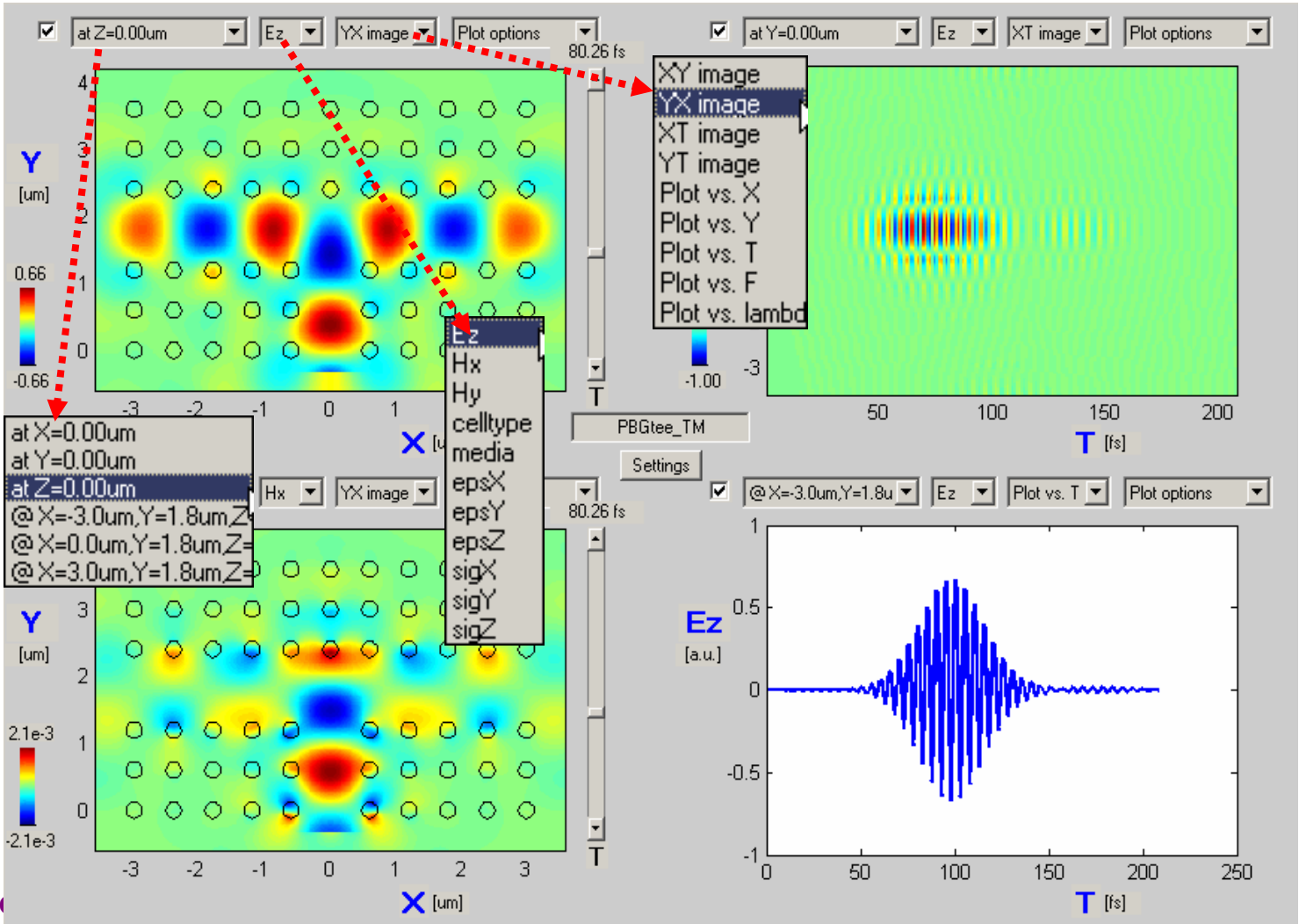
- parallelization
- careful design of numerical experiment

# Parallelized C code

Matlab script

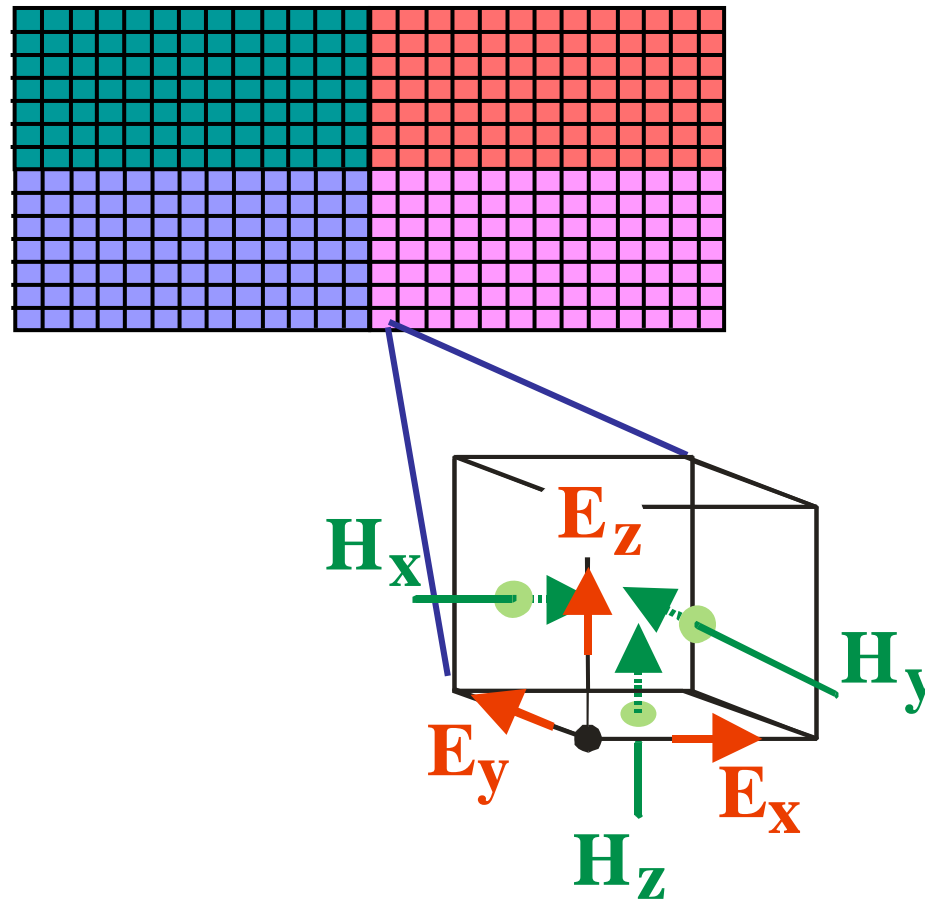


Matlab GUI



# Parallelization using MPI

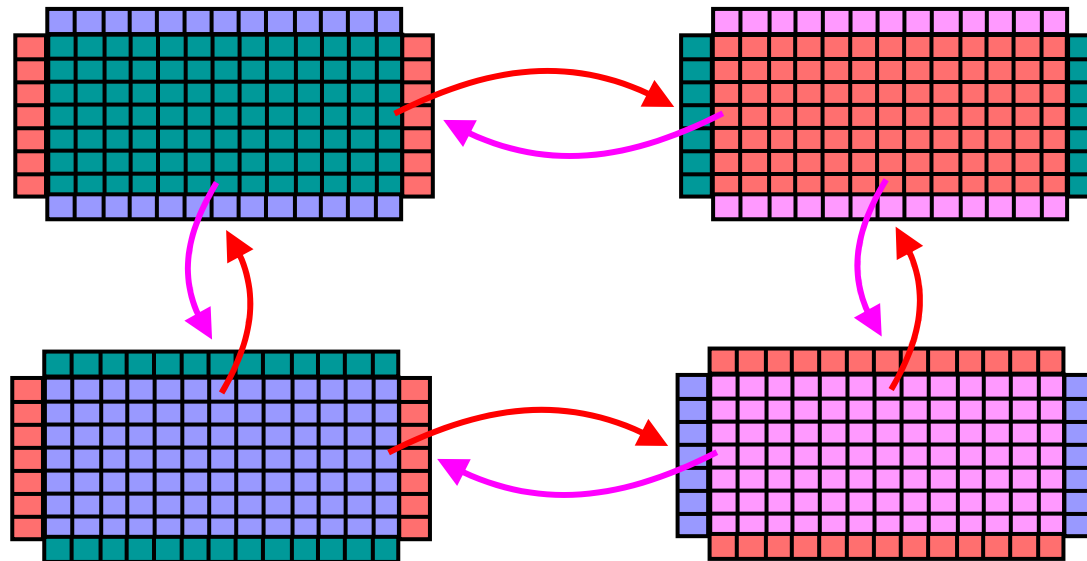
Divide the problem-space among multiple computers



# Parallelization using MPI

Divide the problem-space among multiple computers

Data needed from neighboring computers is also kept locally

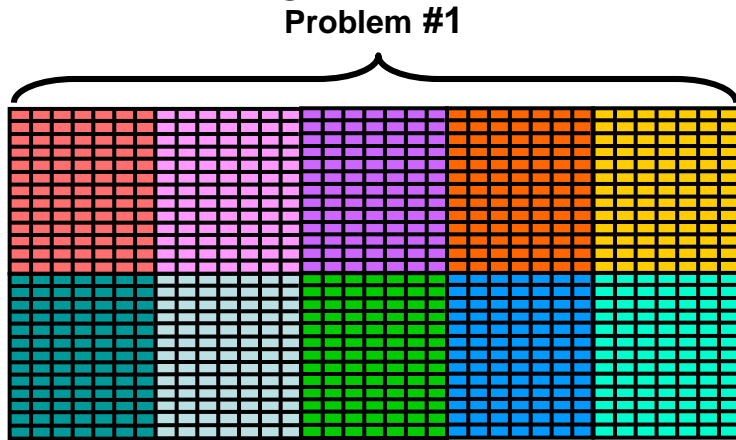


All this redundant data needs to be updated  
once per timestep...

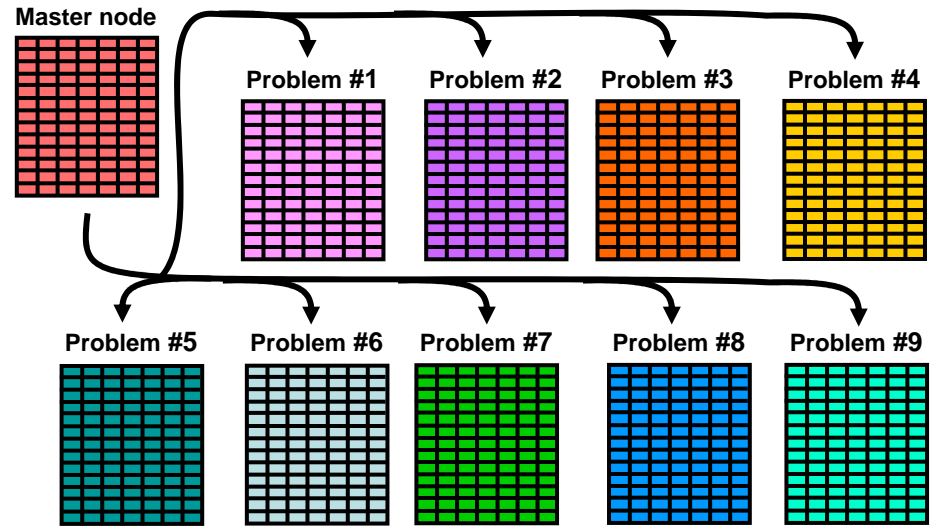
...plus the appropriate boundary conditions observed.

# Two ways to use this parallelism

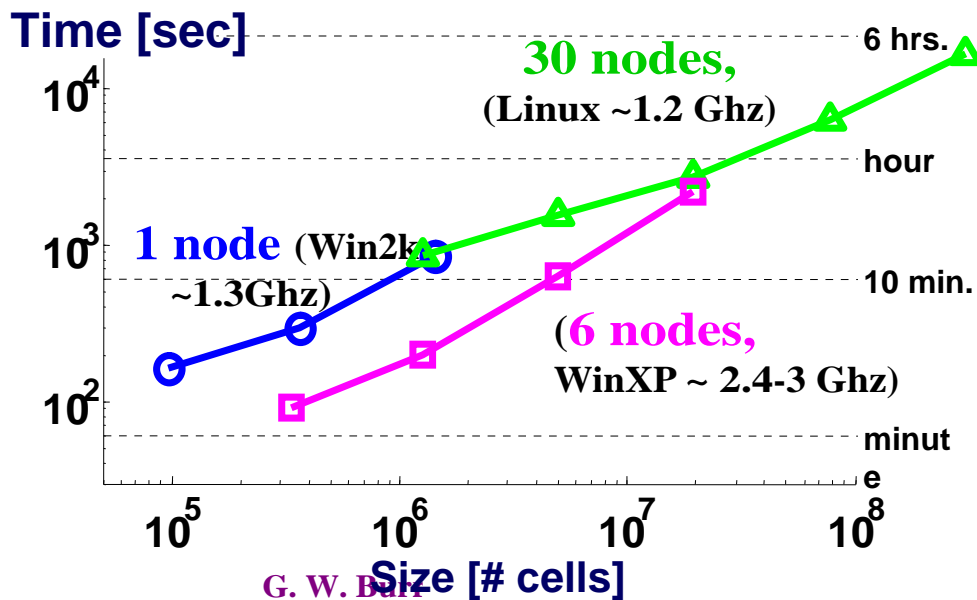
**Distributed mode:** 1 job split amongst 10 machines



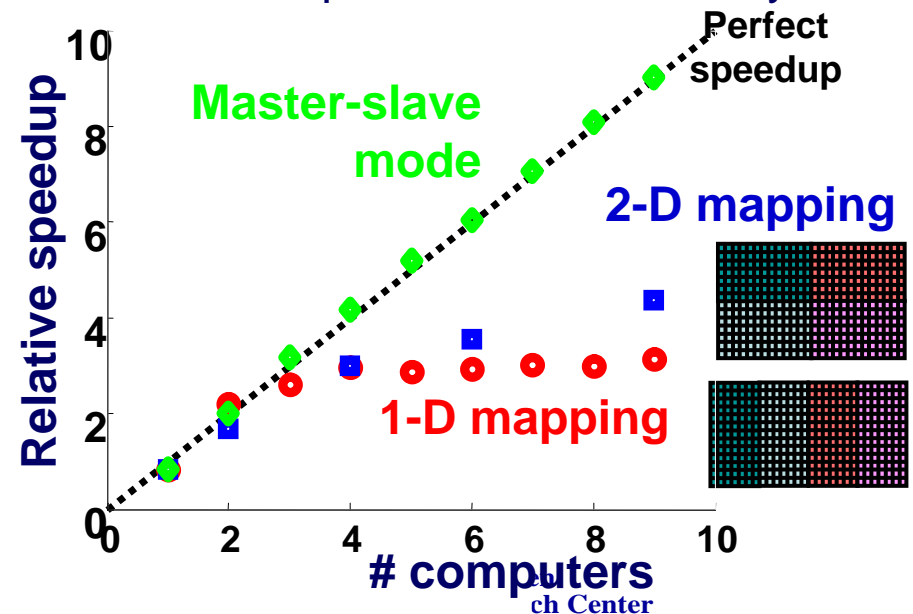
**Master-slave mode:** 9 jobs in parallel



**Distributed mode** can handle really big simulations...



... but **Master-slave mode** uses computers more efficiently



## The sum effect:

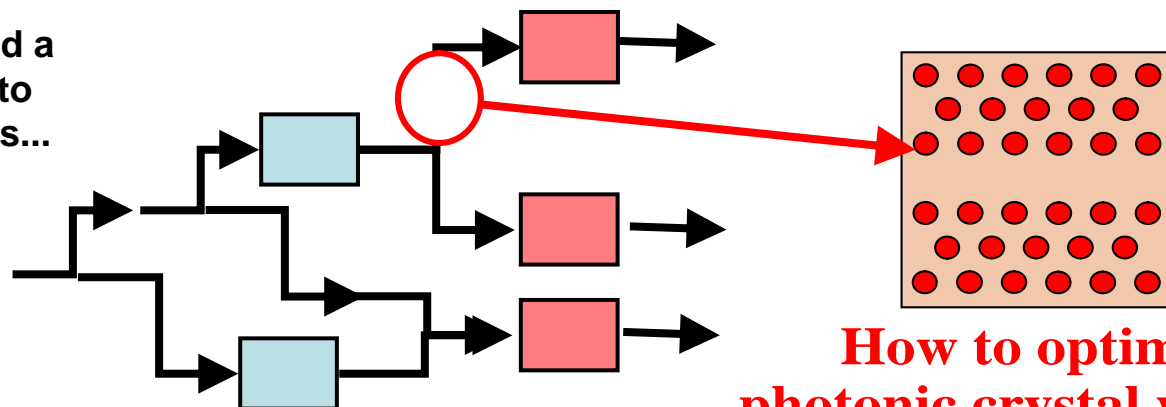
“to be confident you will get the right answer,  
3-D FDTD simulations must be large & slow”

But this means FDTD can only be a *design verification* tool,  
never a *design optimization* tool

- How to change this?
- ✓ error mitigation
  - ✓ parallelization
  - careful design of numerical experiment

# Design of numerical experiment

Always going to need a modular approach to complicated systems...



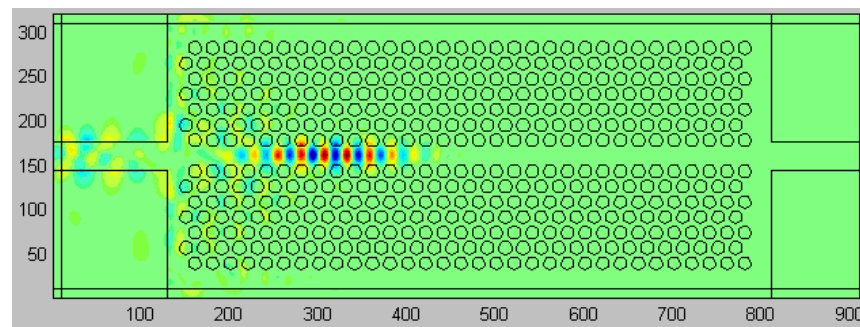
**How to optimize the photonic crystal waveguide?**

I found 3 general methods in the literature for simulating PBG waveguide loss:

**Method #1) Send a pulse in one side, detect it at the other end [1,2, others]**

**Pro:** • only 1-2 simulations needed  
(per design iteration)

**Con:** • simulation is very big  
• measurement combines  
• in-coupling  
• loss-per-cm  
• out-coupling



**920x320x28 (+PMLs)**  
**(takes >10 hours)**

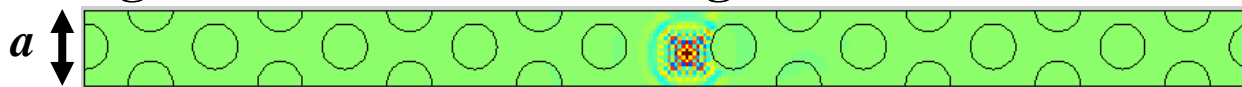
[1] N. Moll and G. L. Bona, Journal of Applied Physics, **93**(9), 4986 (2003).

[2] M. M. Sigalas and E. Chow, Journal of Applied Physics, **93**(12), 10126 (2003).

# Loss in photonic crystal waveguides

**Method #2)** Using periodic boundary conditions, measure loss per cycle and convert to loss-per- $a$  [2]

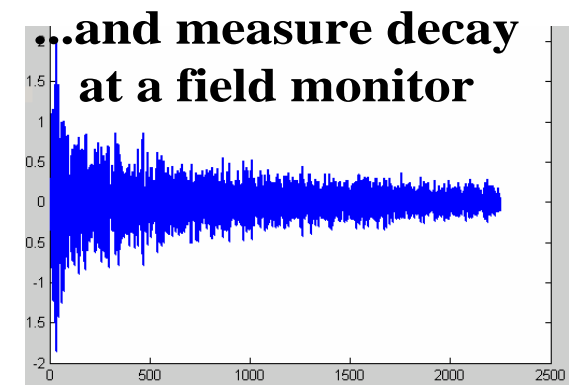
Ping one unit cell of the waveguide...



Possible to even filter to desired frequency

- Pro:**
- same simulations needed anyway to identify defect modes
  - identifies intrinsic loss of PBG waveguide

- Con:**
- need to divide by group-velocity (potential source of significant error)
  - hard to accurately measure low loss
  - still don't know which modes are important (which will get coupled to)



[2] M. M. Sigalas and E. Chow, Journal of Applied Physics, **93**(12), 10126 (2003).

# Loss in photonic crystal waveguides (*continued*)

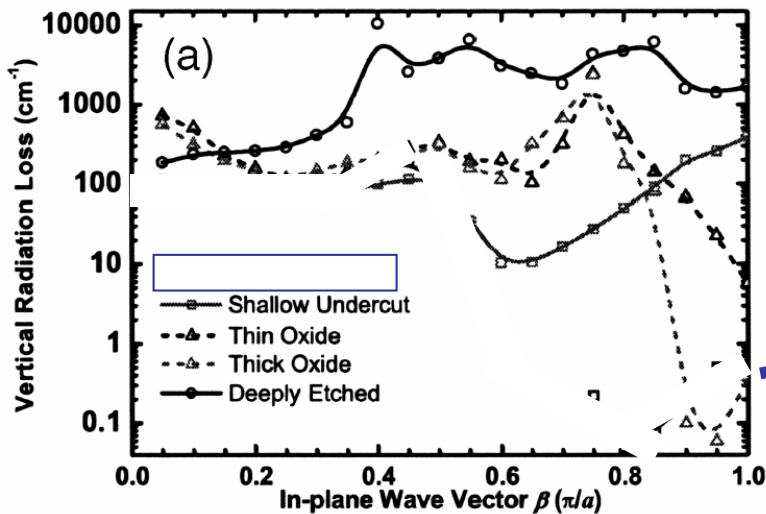
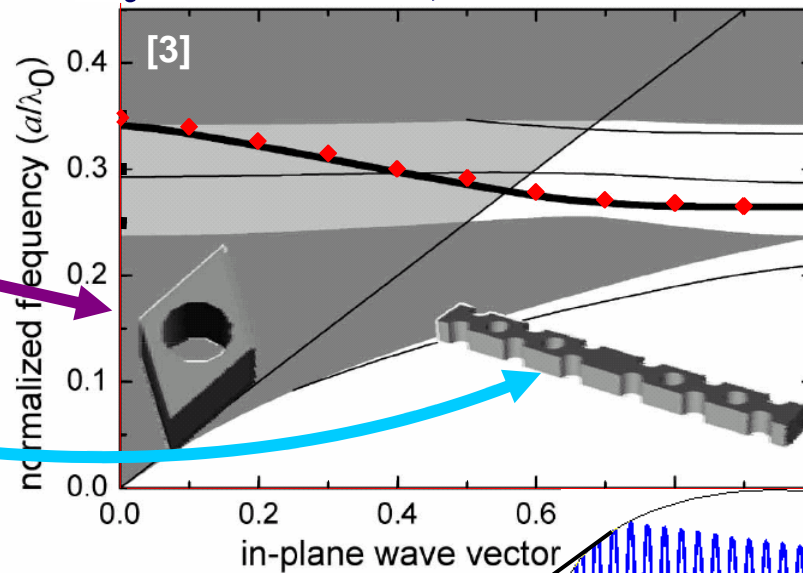
**Method #3) Using periodic boundary conditions, measure**

**loss-per- $a$  directly [2]**

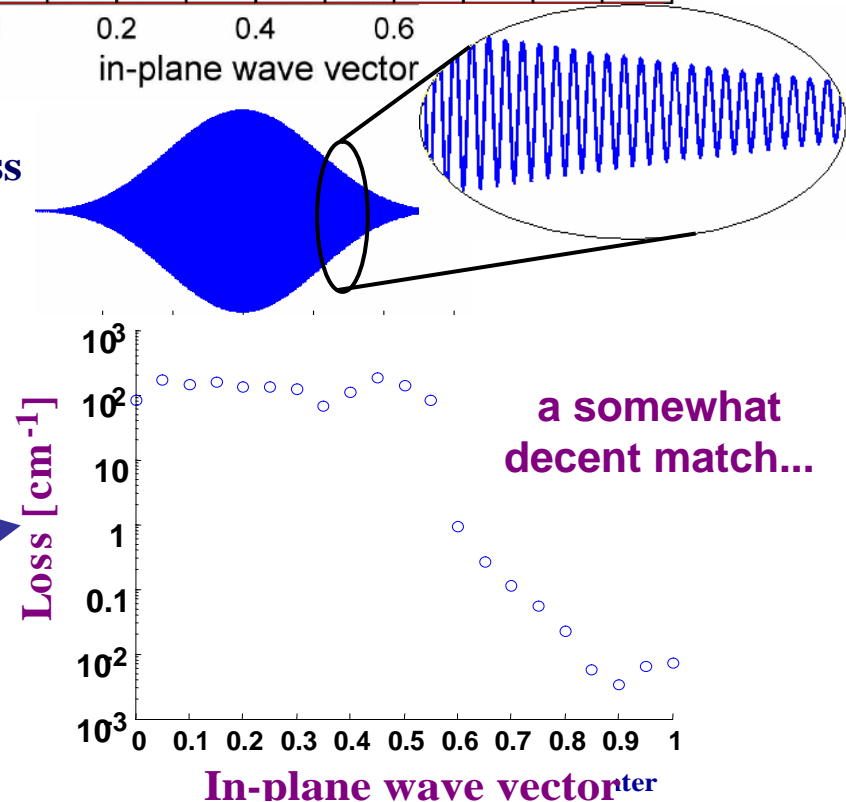
**Step 1: if necessary, identify 3-D PBG diagram using PBG unit cell**

**Step 2: identify waveguide defect mode using waveguide unit cell**

**Step 3: at each  $(\omega, k)$  point, excite this waveguide unit cell with a windowed sinusoid – measure ratio of Poynting vectors out-of-plane to along waveguide as loss**

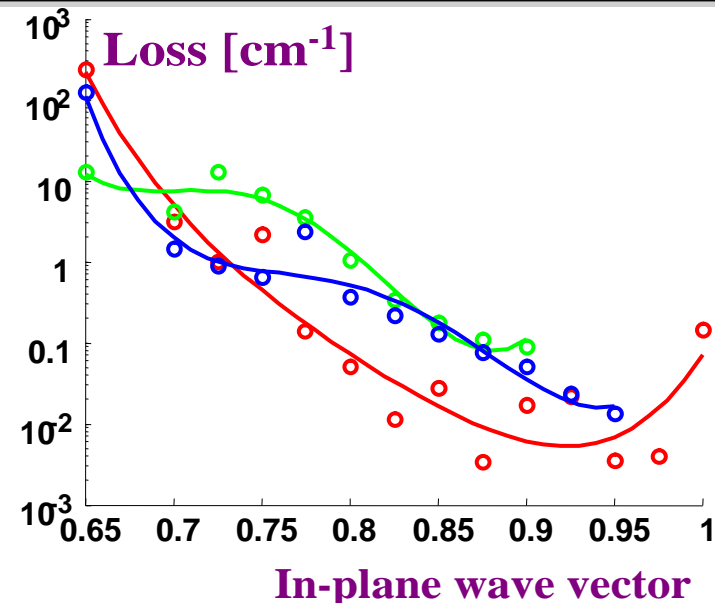
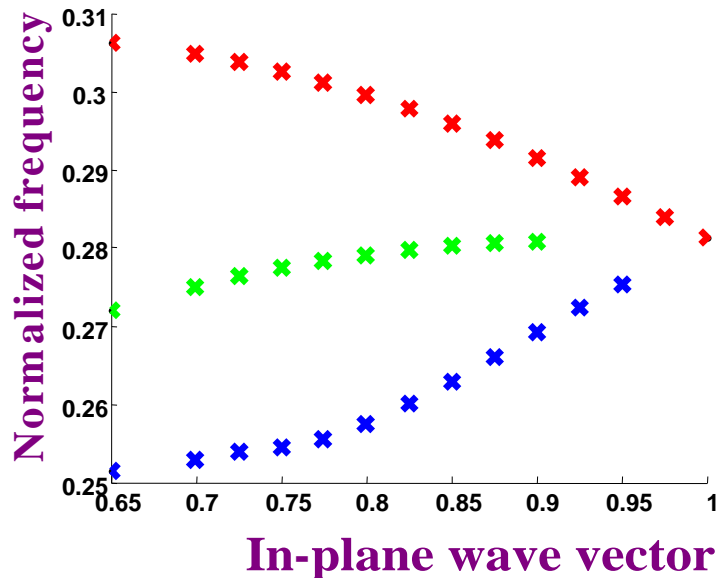
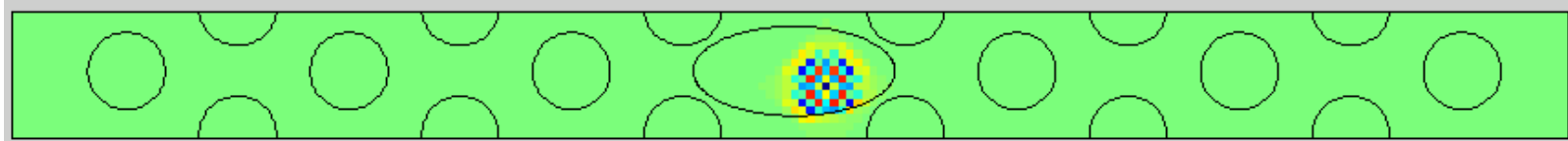


[3] W. Kuang, C. Kim, A. Stapleton, W. J. Kim, and J. D. O'Brien, *Opt. Lett.*, **28**(19), 1781- (2003).



# Loss in photonic crystal waveguides (*continued*)

Now try to extend to elliptical-hole waveguide [1]



**Pro:**

- identifies intrinsic loss of PBG waveguide
- no additional variables needed

**Con:**

- large number of simulations required
- still don't know which modes are important (which will get coupled to)

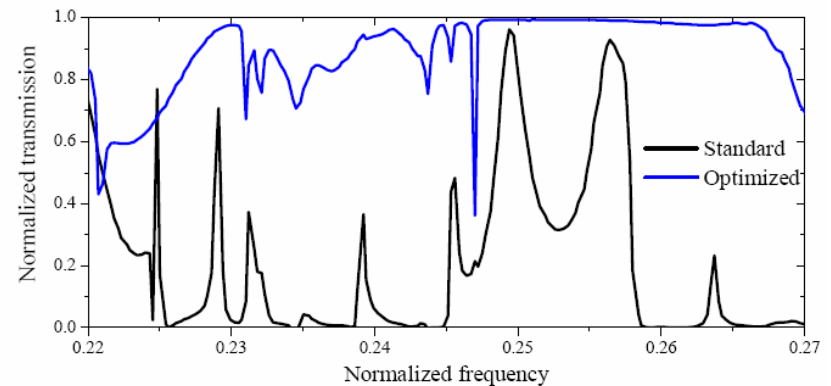
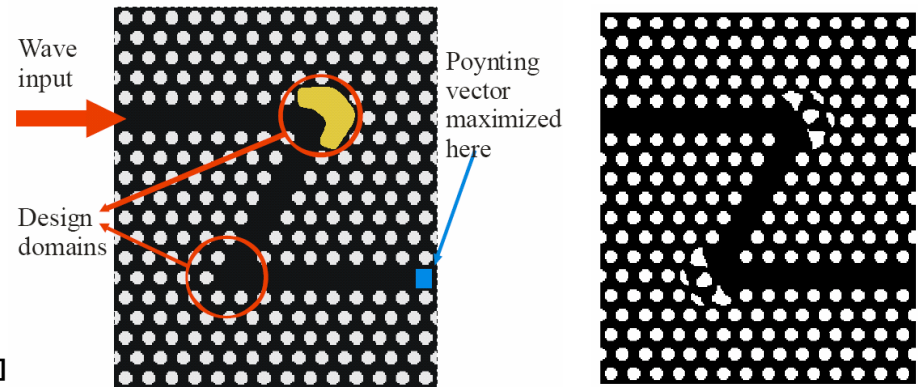
Not clear any of these are suitable to optimize designs with...

**Need Method #4: Get intrinsic loss at all frequencies at once...**

# Adjoint methods

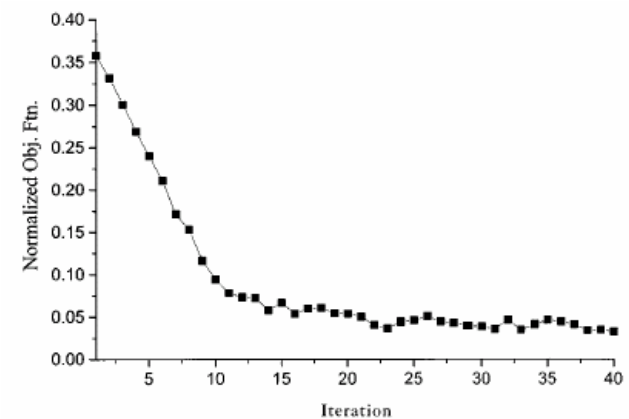
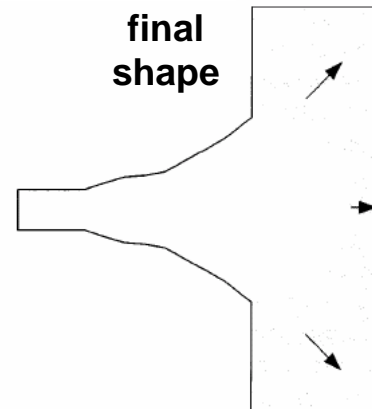
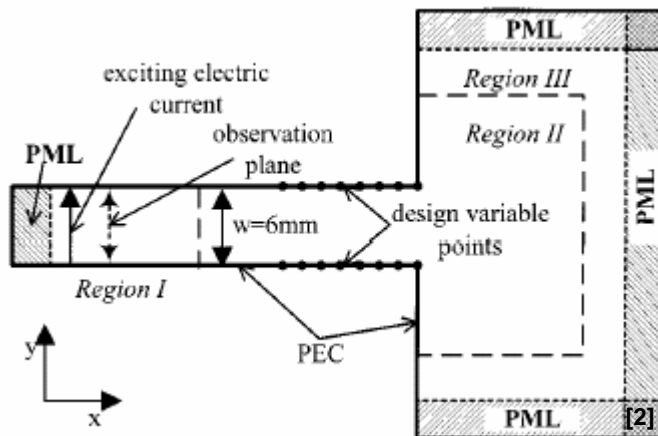
“Topology optimization”

Finite-element analysis (here 2-D) gives derivatives vs. design variables



Recent work showing similar adjoint techniques for

- unstructured FDTD /Finite-element [2]
- regular FDTD [3]



[1] P. I. Borel, A. Harpøth, L. H. Frandsen, M. Kristensen, P. Shi, J. S. Jensen and O. Sigmund, *Opt. Expr.*, **12**(9), 1996 (2004).

[2] Y. S. Chung, C. Cheon, I. H. Park, and S. Y. Hahn, *IEEE Transactions on Magnetics*, **37**(5), 3255, (2001).

[3] N. K. Nikolova, H. W. Tam, and M. H. Bakr, preprint at < [www.ece.mcmaster.ca/faculty/georgieva/papers/AVM\\_FDTDfinalID1750.pdf](http://www.ece.mcmaster.ca/faculty/georgieva/papers/AVM_FDTDfinalID1750.pdf) > 2004

# Conclusions

- The **FDTD** (finite-difference time-domain) algorithm has many advantages as a numerical simulation tool for nanophotonics
- Its disadvantages all boil down to:
  - “to be confident you will get the right answer, simulations must be large & slow”

My motivation is to do *design optimization* with FDTD



- error mitigation
- parallelization
- careful design of numerical experiment

## •Future work

- correct FDTD errors for metals/dispersives
- improve estimation of photonic-crystal waveguide loss
- combine adjoint-methods with FDTD