FDTD as a nanophotonics design optimization tool

Geoffrey W. Burr

IBM Almaden Research Center
650 Harry Road, San Jose, California 95120

The finite–difference time–domain (FDTD) approach is widely used to simulate the expected performance of photonic crystal, plasmonic, and other nanophotonic devices. Unfortunately, given the computational demands of full 3–D simulations, researchers can seldom bring this modeling tool to bear on more than a few isolated design points. Thus FDTD — as it stands now — is a *verification* rather than a *design optimization* tool.

Over the long term, improvements in computing power will bring structures of current interest within general reach. But this paper addresses the near term: what can be done (given the computing power available today) to make FDTD act more like a design optimization tool?

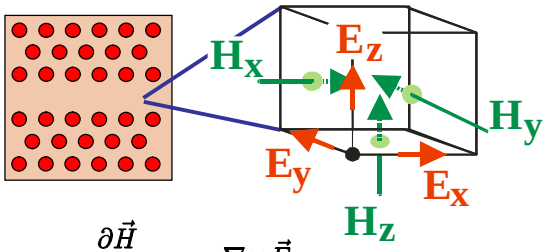
Here I study:

- **the compensation of numerical dispersion and "staircasing" in FDTD, allowing coarser gridding without sacrificing accuracy or simplicity**
- **the impact on accuracy when using FDTD on typical photonic bandgap structures**
- **trying to speed up the simulation of out-of-plane losses in PBG waveguide structures**

1 Background

"Um... so what was FDTD again?"

The Finite-difference Time-Domain (FDTD) algorithm models the propagation of light by discretizing Maxwell's Equations in both time and space coordinates. Although many variants exist, in the basic Yee algorithm[2,3] each grid node contains 3 E-field and 3 H-field components. Both the spatial positions as well as the temporal update of these components are offset ("leap-frogged") to improve the estimation of derivatives.



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

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

$$\Rightarrow \quad E_y \mathrel{+}= \frac{\Delta t}{\varepsilon_r \varepsilon_0} \frac{\left(H_z - H_z^{\text{previous cell}} \right)}{\Delta x}$$

Advantages of FDTD

(for simulating photonic crystal & nanophotonic devices, for instance)

- **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 FDTD *("no pain, no gain")*

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

The combined impact of these factors is that

— for many 3-D structures of interest to the nanophotonics/photonic bandgap community —

an FDTD simulation of high accuracy simply requires too much computer memory and takes too long to run.

Thus it's not surprising that 3-D FDTD is mostly used to *verify* existing nanophotonics designs...

G.W. Burr

IBM Almaden Research Center

PECS-V (2004)
poster Mo-P49

2 Motivation

"OK... so if FDTD is so cumbersome, then what's this poster about anyway?"

Nanophotonics designers need to have accurate 3-D modeling tools
(the alternative is *design-by-nanofabrication* — also known as "trial-and-error": a tremendous waste of time and resources)

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 $LiNbO_3$ 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.

So what can be done to finesse the disadvantages of 3-D FDTD?

We can:

1. **Buy (or borrow) more computing power**
2. **Develop alternative algorithms, sacrificing simplicity and flexibility to get smaller simulations (Frequency-domain methods, higher-order FDTD, etc.)**
3. **Try to be more clever with standard FDTD — in simulation design, and in pre- or post-compensating (where possible) for the inaccuracies of FDTD.**

In this poster, I try to push FDTD closer to a nanophotonics optimization tool through items #1 and #3.

I show

- **a parallelized FDTD tool (demonstrated on up to 60+ nodes, used regularly over 10)**
- **verification against experiment (Fresnel reflection coefficients, plasmon resonances)**
- **improved accuracy in coarsely-gridded FDTD (by reducing the impact of numerical dispersion and staircasing)**
- **initial results on rapid simulation of loss in PBG waveguides**

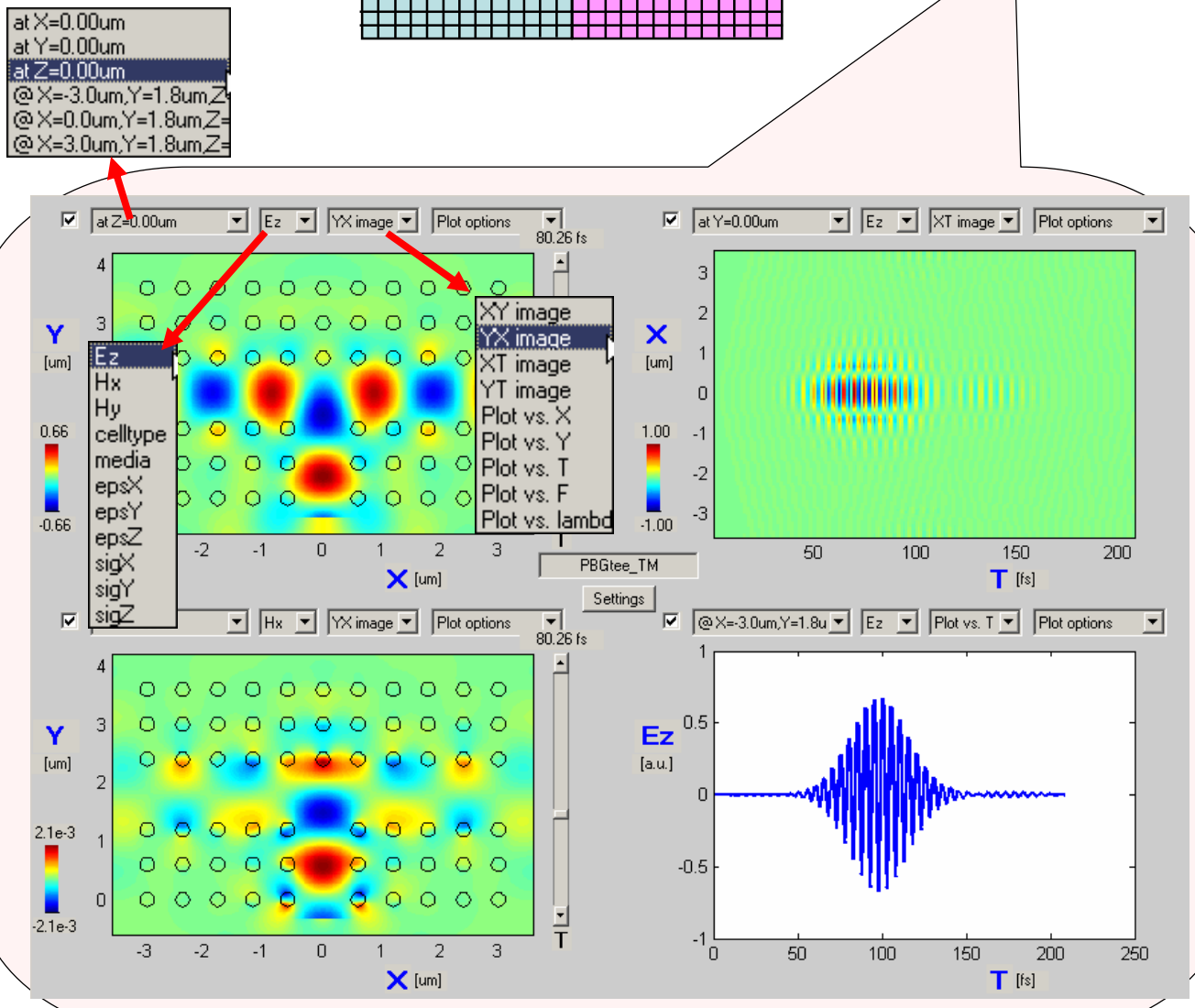
G.W. Burr
IBM Almaden Research Center
PECS-V (2004)
poster Mo-P49

3 FDTD implementation

"Why buy an FDTD tool when you can spend all your time writing one..."

```
Initialize(...);
output_directory = '...';
TerminateWhen(...);

DefineSimpleMaterial(...);
DefineSimpleMaterialByIndex(...);
DefineSimpleMaterialFromFile(...);
DefineDrudeMetal(
DefineDrudeMetalFromFile(...);

SetCenterWavelength(...);
SetSandbox(...);
SetPML(...);

SetWavevector(...);
SetBoundaryConditions(...);

AddSphere(...);
AddCylinder(...);
AddRectangle(...);
AddPolyhedron(...);
AddArbShape(...);

Establish_CW_Waveform(...);
Establish_Pulse_Waveform(...);
Establish_Impulse_Waveform(...);
Establish_windowedCW_Waveform(...);

Excite_PointSource(...);
Excite_ShapedSource(...);
Excite_Infinite_Planewave(...);
Excite_Apertured_Planewave(...);
Excite_Enclosed_Planewave(...);

RequestDataset(...);
AssignDatasetForConvergence(...);
ContinueFrom(...);
RegisterArbStep(...);

GoStructure;
RunFDTD(...);
GoPlot;
```

Matlab script → **Parallelized C++ engine** → **Matlab GUI**



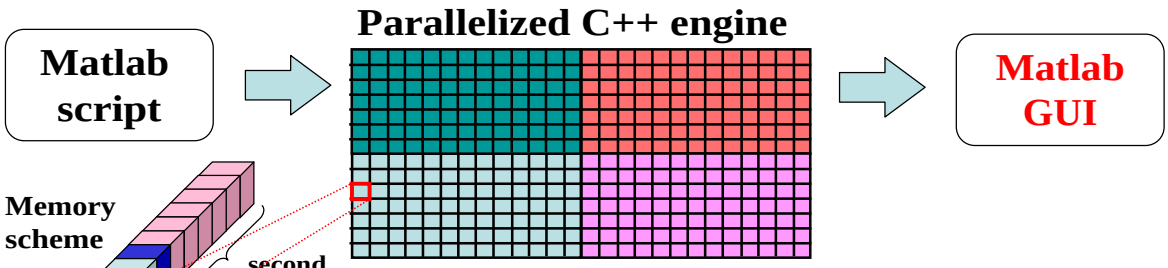
G.W. Burr

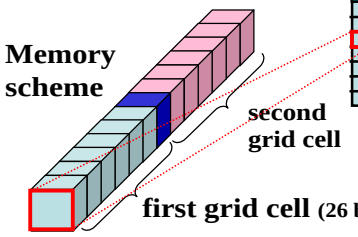
IBM Almaden Research Center

PECS-V (2004)
poster Mo-P49

4 Parallelism

"OK, fine — but won't you need to run FDTD code on more than one computer...?"





Homemade MPI implementation

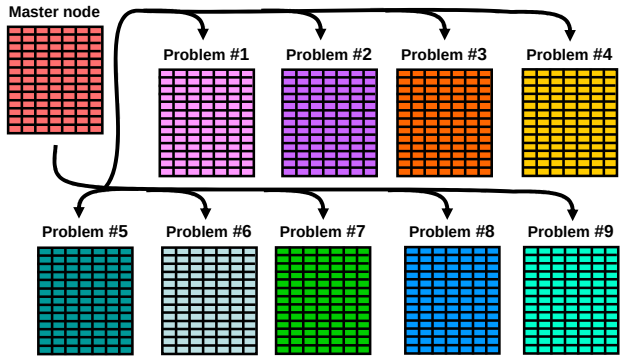
- Each XY point is a separate pointer
- Data for Z spatial dimension (at that *x,y*) is all in one array

Pro:
- Compact, fast
- Designed for FDTD

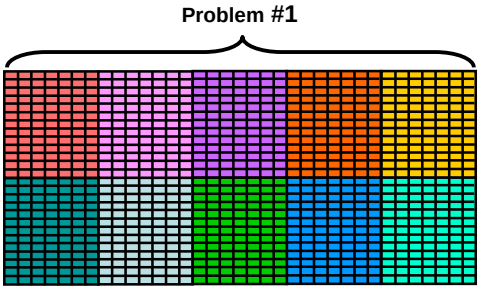
Con:
- Complexity (XY, XZ, YZ)
- Scalability?
- How to vary grid spacing?

Two ways to use this parallelism

Master-slave mode: 9 jobs in parallel



Distributed mode: 1 job split amongst 10 machines



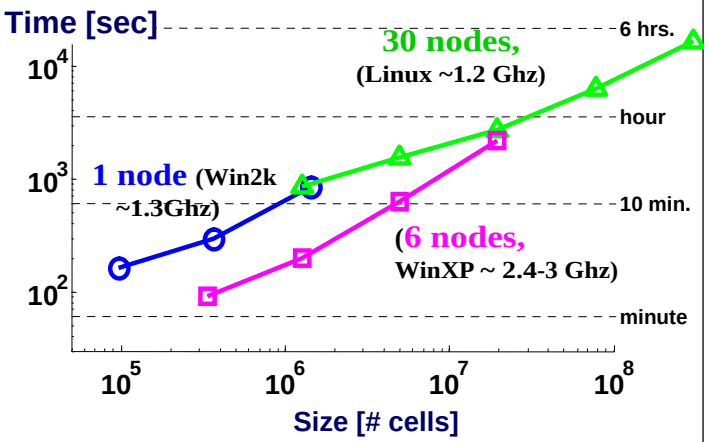
I prefer **Master-slave mode** if

- each simulation fits on 1 machine, and
- I'm going to run multiple jobs anyway (optimization, band diagrams, etc.)

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

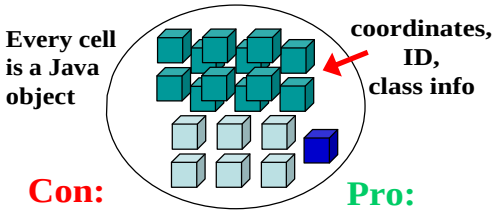
Advantages:

- no redundant grid points
- little network overhead
- less post-execution re-assembly of output files
- easier to combine heterogeneous machines



I've also worked with OptimalGrid

(Experimental grid-computing middleware from IBM Research)



Con:
- Space & time overheads
- Java is slow

Pro:
- Simpler to program
- more scalable
- Variable grid spacing?

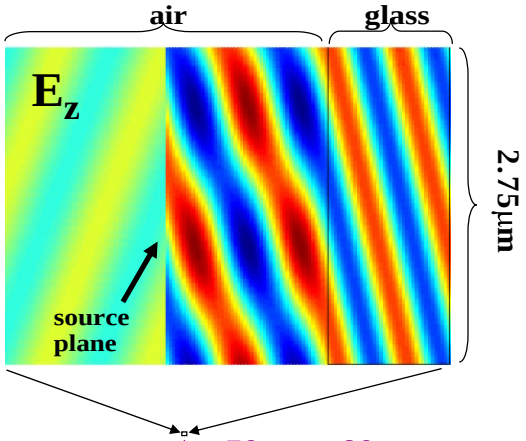
G.W. Burr

IBM Almaden Research Center

PECS-V (2004)
poster Mo-P49

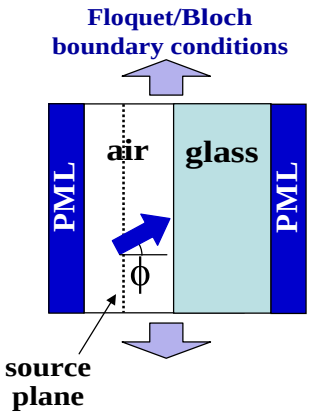
5 Verification against experiment

"How can you know you've written your FDTD code correctly?"



Fresnel reflection coefficients

(with 2-D FDTD)



Coarse simulations make nice pictures...

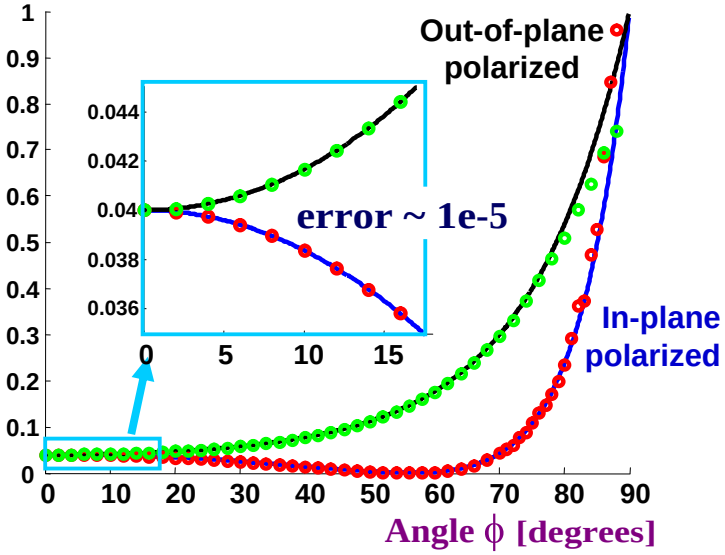
...but small gridding is what produces lower error.

50nm x 30nm w/ 2nm cells

Reflectivity

Analytical result: —— (the "right answer")

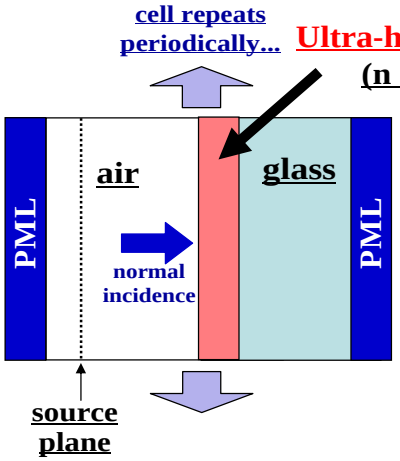
FDTD results: ○ ○

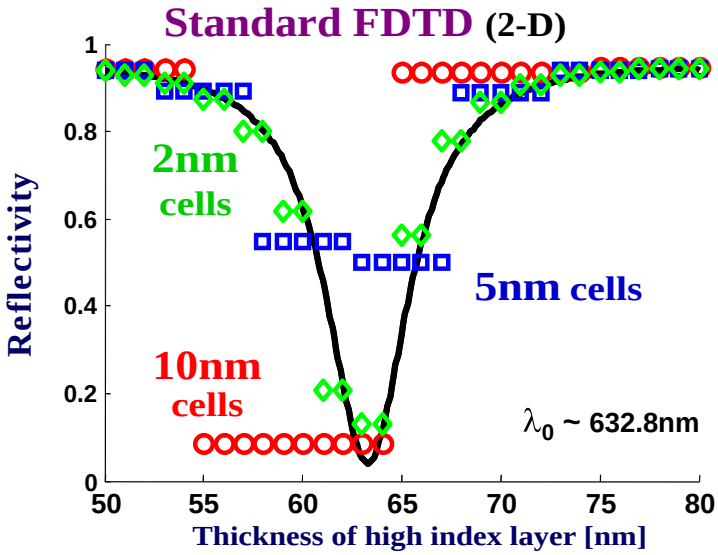


"...Ok, so this is nice as verification but..."

What happens with coarser gridding?

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





With coarse gridding...

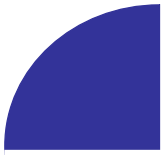
- small spatial features get lost
- get wrong answer AND wrong local minimum

6 Combating sources of error in FDTD

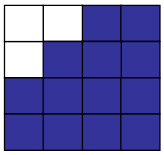
"So coarse gridding is bad – what can you do about it?"

"Staircasing" of small spatial features

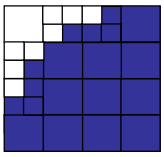
We really want to simulate a smooth interface between regions of ε_1 and ε_2...



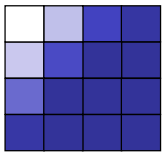
...but the discretization in FDTD gives a "staircase" effect.



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

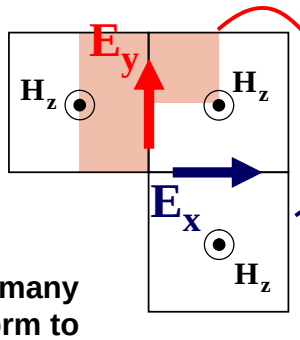


...but we can get a similar effect by using an *effective* ε in each cell.



Indeed, the idea of an effective dielectric constant ε is not new [3-5]. What is done differently* here is that each E-field component gets its own effective ε, weighted over the cell-sized volume around it.

This makes sense in terms of the field-update equations.



$$E_y \mathrel{+}= \frac{\Delta t}{\varepsilon_r \varepsilon_0}\left(\frac{H_z - H_z^{\text{previous cell}}}{\Delta x}\right)$$

$$E_x \mathrel{+}= \frac{\Delta t}{\varepsilon_r \varepsilon_0}\left(\frac{H_z - H_z^{\text{previous cell}}}{\Delta y}\right)$$

And it gives you 3x as many variables to help conform to the desired material boundary...

* Interestingly, the Taflove textbook [3, p.433] summarizes the work of Dey & Mittra [4] as a field-centered approach. But the actual paper [4] describes a cell-centered approach, and later papers of that group [6] support this. There are insufficient details in Chan & Ho [5] to know which was used there.

The wrong answer/local minimum due to numerical dispersion: getting the speed-of-light wrong

The dispersion relation for free space should be:

$$\left(\frac{\omega}{c}\right)^2 = (k_x)^2 + (k_y)^2 + (k_z)^2$$

but instead in the discretized grid you get:

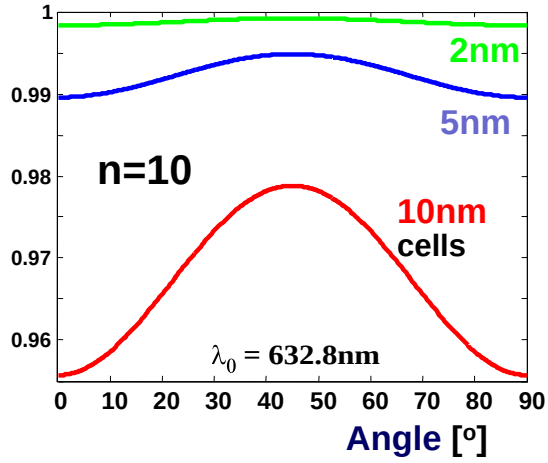
$$\left[\frac{1}{c\Delta t}\sin\left(\frac{\omega \Delta t}{2}\right)\right]^2 = \left[\frac{1}{\Delta x}\sin\left(\frac{\tilde{k_x}\Delta x}{2}\right)\right]^2 + \left[\frac{1}{\Delta y}\sin\left(\frac{\tilde{k_y}\Delta y}{2}\right)\right]^2 + \left[\frac{1}{\Delta z}\sin\left(\frac{\tilde{k_z}\Delta z}{2}\right)\right]^2$$

7 Combating numerical dispersion

"So coarse gridding is bad – what can you do about it?"

The discretization causes wavefronts to propagate at different speeds in different directions within the grid.

This effect gets worse as high index and high frequency decrease the local wavelength.



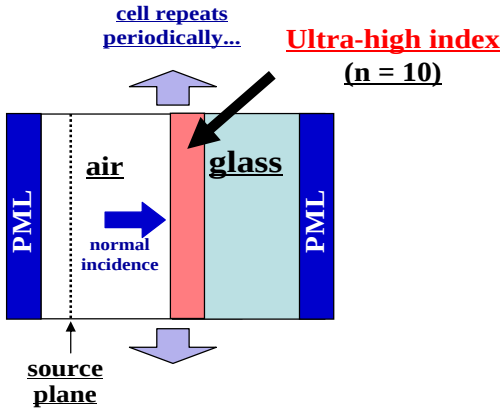
- **Taflove[3] described a way to compensate this exactly at one angle and wavelength by adjusting the global values of both ε_0 and μ_0, fudging c back up to the proper value.**

- **Juntunen[7] extended this to non-cubic lattices, and showed that errors could still be reduced across all angles and a band of frequencies (wavelengths).**

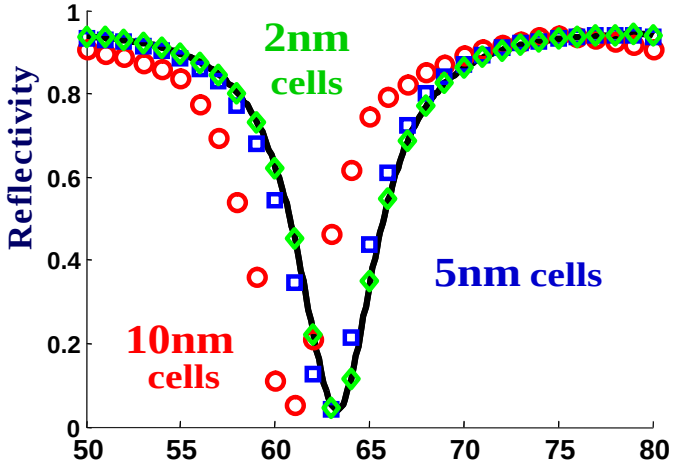
Here I extend this to n≠1 materials, implementing a slightly different local ε_0 and μ_0 in each dielectric media to either:

- **completely compensate numerical dispersion at a particular angle & wavelength**
- **minimize the worst-case numerical dispersion across a band of angles/wavelengths**

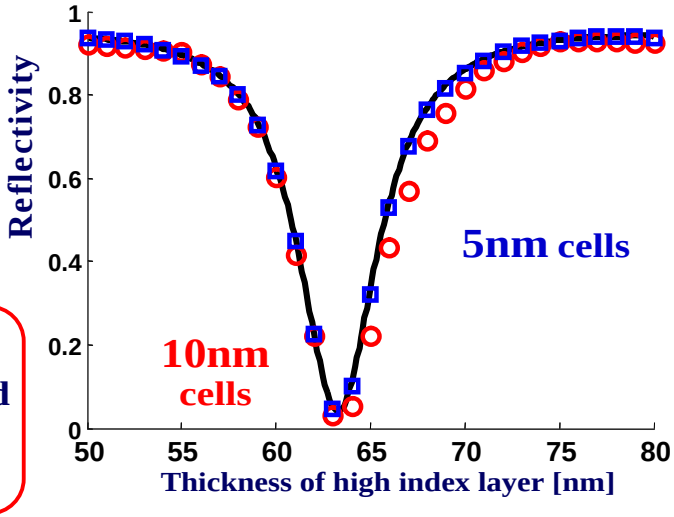
Returning to our artificial example...



FDTD with field-based effective ε:



FDTD with field-based effective ε *and* (ε_0, μ_0) corrected for numerical dispersion



Despite coarse gridding...

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

G.W. Burr

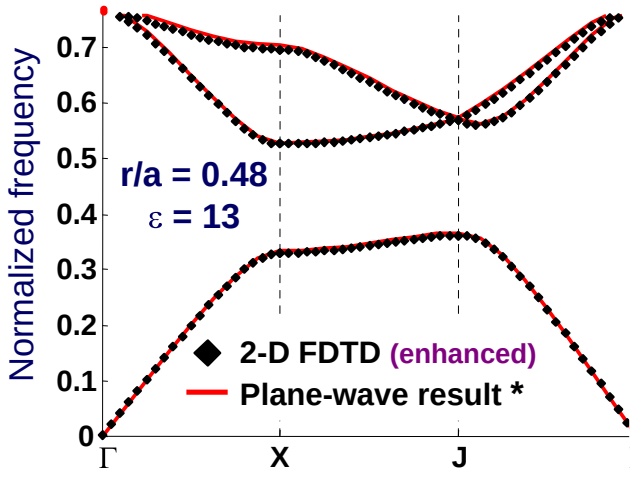
IBM Almaden Research Center

PECS-V (2004) poster Mo-P49

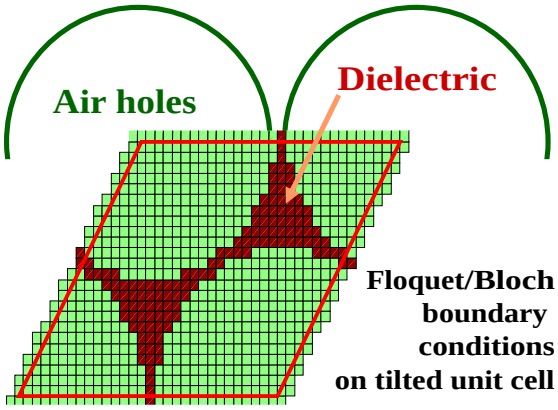
8 Example: PBG dispersion diagrams

"Hey, great — but I don't plan on using material of *n*=10 any time soon..."

Triangular Lattice – TE bands



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

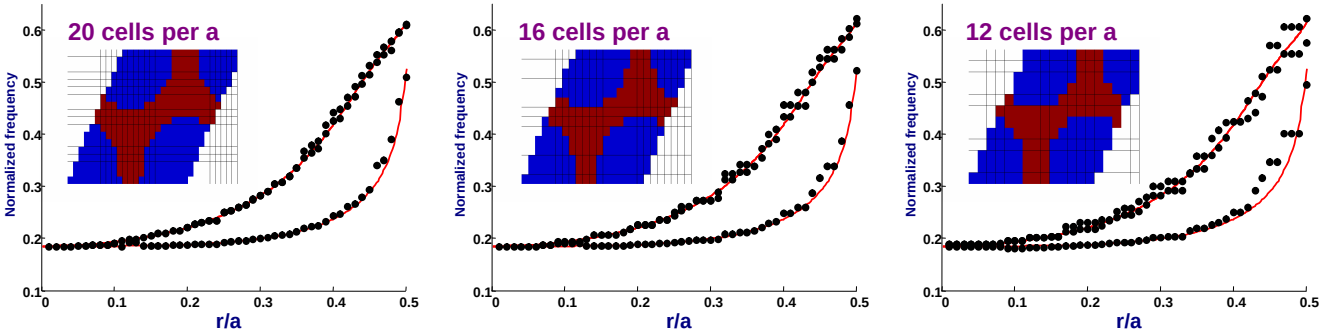


Floquet/Bloch boundary conditions on tilted unit cell

Standard FDTD approach

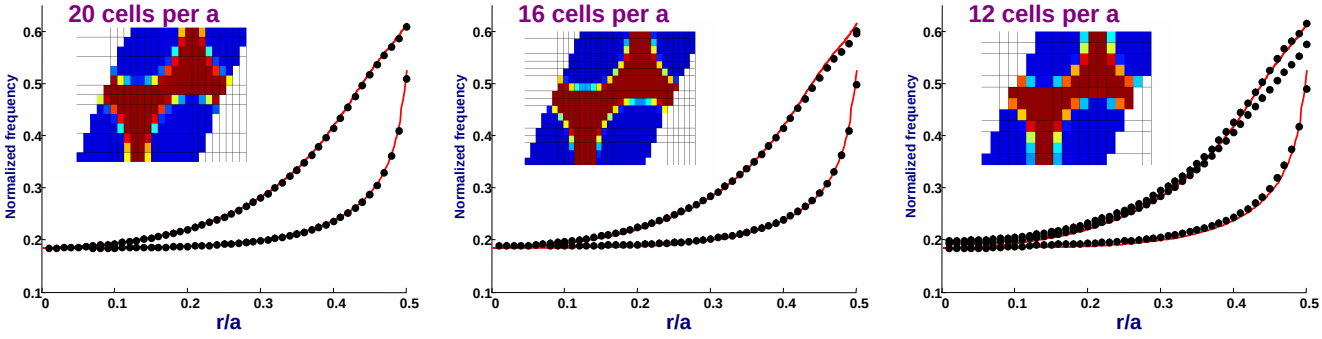
1. **ping with an impulse**
2. **set k-vector at boundaries**
3. **monitor at non-symmetry point**
4. **FFT becomes column of dispersion diagram**

Using the band-diagram at J-point vs. r/a to compare results with standard FDTD...



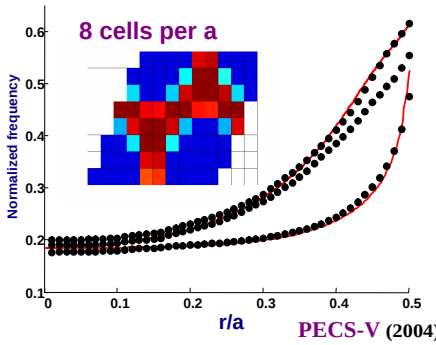
...against the modified FDTD described in this poster

(compensation for numerical dispersion and "staircasing")



Observations:

- **error increases with frequency**
  (smaller wavelength, etc.)
- **modified FDTD permits coarser simulations yet same accuracy**
- **peak-finding algorithm also important**
  (use imaginary part to find double peaks?)



G.W. Burr

IBM Almaden Research Center

PECS-V (2004)
poster Mo-P49

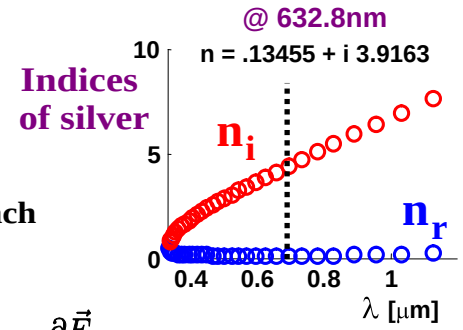
9 Simulating metals

"OK, but what about metals?"

FDTD cannot model materials with Re{ε} < 1

(such as silver at visible wavelengths)

except by also modeling material dispersion.



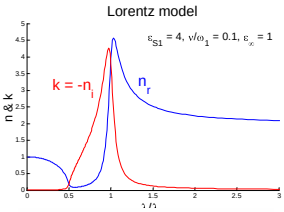
This can be incorporated in several ways — the approach used here is from Ziolkowski et al. [8], a variant of the *auxiliary differential equation* (ADE) approach [3].

Polarization terms are added to and updated at each metal cell. This approach is nice because it can be extended to the Lorentz, Debye, and Drude dispersion models and could lead straightforwardly to nonlinear optics.

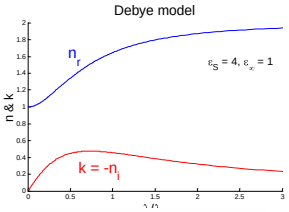
$$\epsilon_\infty \frac{\partial \vec{E}}{\partial t} = \nabla \times \vec{H} - \vec{J}_p$$

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

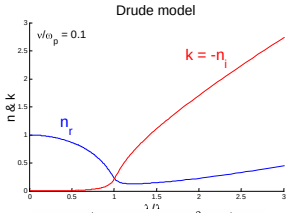
$$\frac{\partial \vec{J}_p}{\partial t} + \vec{J}_p = f\left(\vec{E}, \vec{P}, |E|^2, \text{etc.}\right)$$



$$\vec{D} = \left(\epsilon_\infty + \frac{\epsilon_0 \chi_0 \omega_0^2}{\omega_0^2 - \omega^2 + j\Gamma\omega}\right)\vec{E}$$



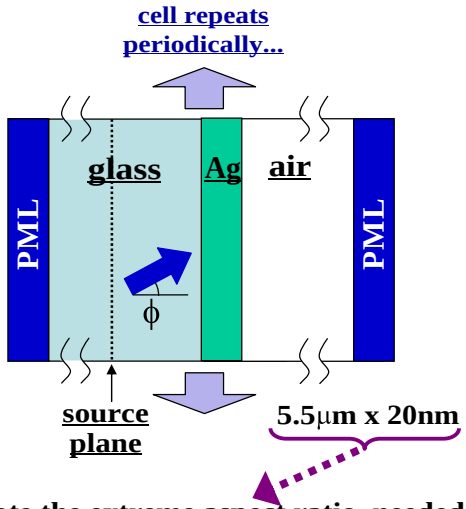
$$\vec{D} = \left(\epsilon_\infty + \frac{\epsilon_0 (\epsilon_s - \epsilon_\infty)}{1 + j\omega\tau}\right)\vec{E}$$



$$\vec{D} = \left(\epsilon_\infty + \frac{\epsilon_0 \omega_p^2}{-\omega^2 + j\nu\omega}\right)\vec{E}$$

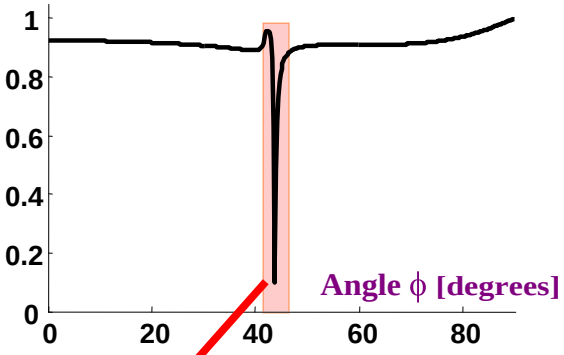
Verification (Drude model):

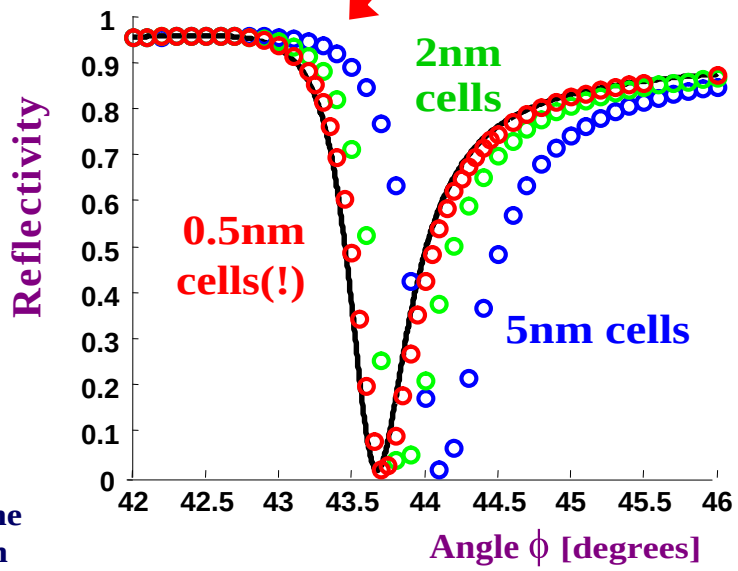
plasmon resonance of a thin silver film



Note the extreme aspect ratio, needed to keep the PMLs away from the silver layer. This is not unique to the treatment of dispersives, but to evanescent waves in general: the same separation is required for the simple glass-to-air interface, once past the critical angle...

Plasmon resonance is extremely sharp





The compensation of the numerical dispersion here — to achieve the same match to the plasmon resonance with coarse gridding — is still underway...

G.W. Burr

IBM Almaden Research Center

PECS-V (2004) poster Mo-P49

10 Loss in PBG waveguides

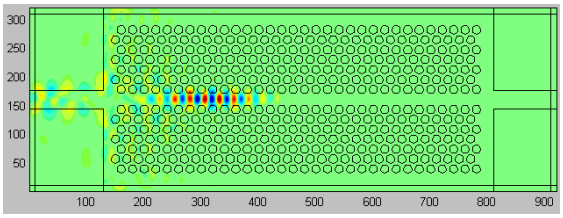
"Fine – let's see something you can optimize..."

Using 3-D FDTD to minimize the expected losses in photonic crystal waveguides seems like a good target for a nanophotonics design tool:

- **it's seemingly intractable given current computing power**
- **it doesn't seem to have been done yet**
- **it would be useful to the field as a whole.**

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

1) Send a pulse in one side, detect it at the other end [3,9,10, *others*]



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

Con: • **simulation is very big**

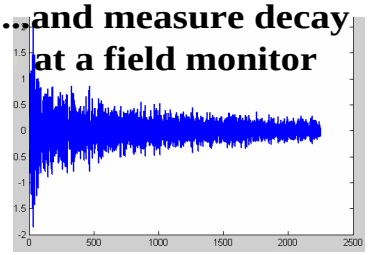
• **measures in-coupling and loss-per-cm together** (plus outcoupling)

920×320×28 (+PMLs)
(takes >10 hours)

2) Using periodic boundary conditions, measure loss per cycle and convert to loss-per-*a* [9]

Ping one unit cell of the waveguide...

...and measure decay at a field monitor



Possible to even filter to desired frequency

Pro: • **same simulations needed anyway to identify defect modes**

• **identifies intrinsic loss of PBG waveguide**

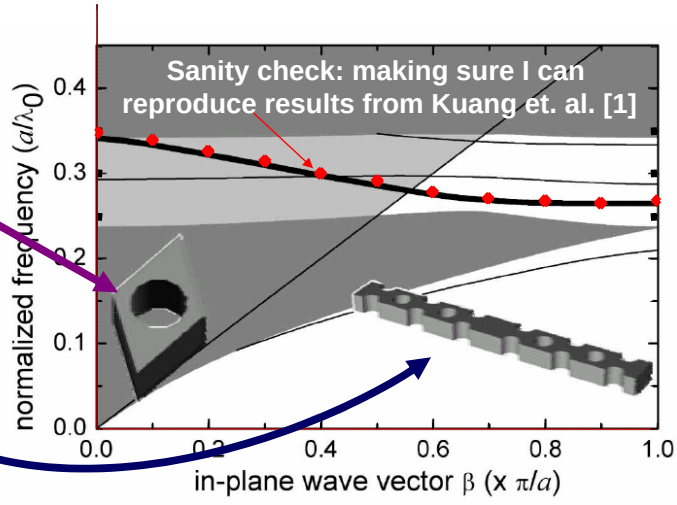
Con: • **don't know which modes are important** (which will get coupled to)

• **need to divide by group-velocity** (potential source of significant error)

3) Using periodic boundary conditions, measure loss-per-*a* directly [1]

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

Step 2: identify waveguide defect mode using waveguide unit cell



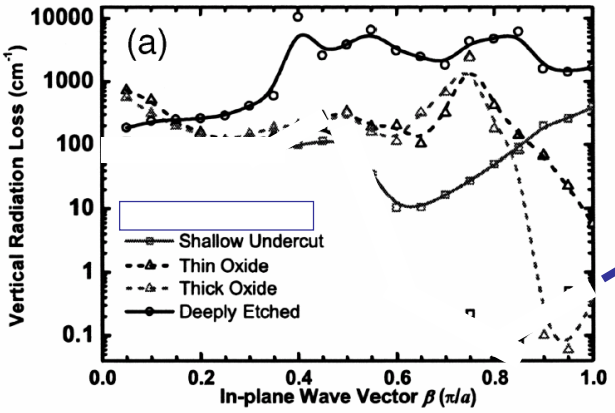
11 Loss in PBG waveguides

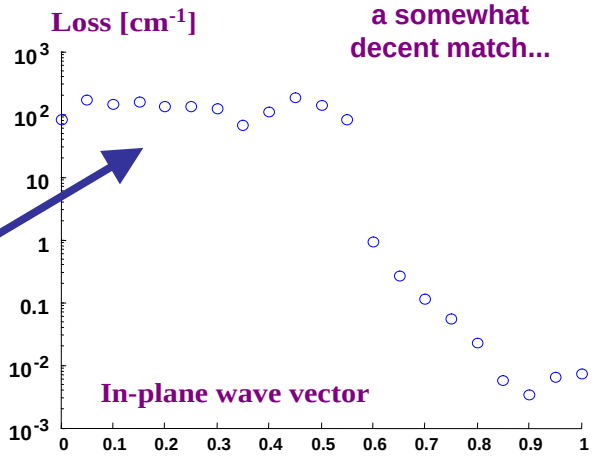
"Fine – let's see something you can optimize..."

3) *continued:* Measuring loss-per-a directly [1]

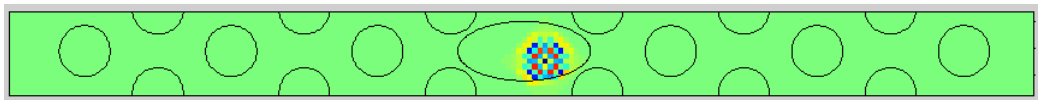
Step 3: at each (ω,k) point, excite same PBG unit cell with windowed sinusoid – measure ratio of Poynting vectors out-of-plane to along waveguide as loss

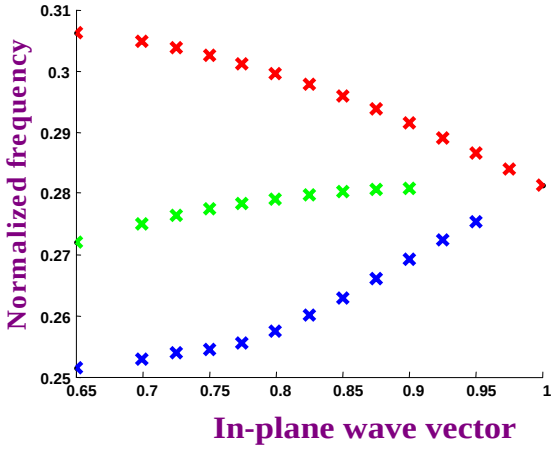
More sanity checking against Kuang [1]

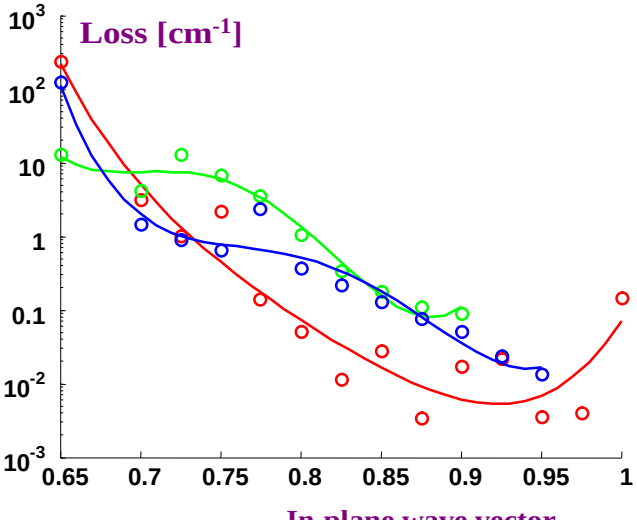




Now try to extend to elliptical-hole waveguide [10]







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

Con:
- still don't know which modes are important (which will get coupled to)
- large number of simulations required
- imprecise ω gives different answer?
- "double-orthogonal" sinusoid not always unidirectional?

Not clear any of these are suitable to optimize designs with... probably need to come up with something else.

12 Conclusions

"Um.. so what did I learn here?"

- **The advantages of FDTD are numerous, while its disadvantages all boil down to:**

"to be confident you will get the right answer, simulations must be large & slow"

•These disadvantages can be finessed in many cases by simple adjustments, including:

- **reducing "stair-casing" by using field-centered effective dielectric constant at interfaces**
- **compensating for numerical dispersion (in each material, for the lowest wavelength of interest)**

•These techniques were shown to significantly improve accuracy at coarse gridding in PBG-relevant applications

•Approaches for simulation of out-of-plane loss in PBG waveguides were discussed (from the point-of-view of implementing nanophotonics design optimization)

Avenues for future work

- **Correction of numerical dispersion (or other errors?) for material-dispersive media such as metals**
- **Improved boundary conditions for metals and other simulations involving evanescent waves**
- **Alternative waveguide-loss-simulation techniques (needs to be fast, with minimal uncertainties, and extendable to studies of fabrication error & surface roughness)**

Acknowledgements

I'd like to thank...

- my colleague Bob Shelby for providing the plane-wave PBG results from the MIT Photonic Bands package.
- Bob Shelby, Bill Risk, Bulent Kurdi and my other colleagues at IBM Almaden, Nikolaj Moll (IBM Zurich), and Annette Grot and Mihail Sigalas (Agilent) for helpful discussions.
- you for your time & interest!

13 References

[1] W. Kuang, C. Kim, A. Stapleton, W. J. Kim, and J. D. O'Brien, "Calculated out-of-plane transmission loss for photonic-crystal slab waveguides," *Optics Letters*, **28**(19), 1781-1783, (2003).

[2] K. S. Yee, "Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media," *IEEE Transactions on Antennas and Propagation*, **14**(3), 302-307, (1966).

[3] A. Taflove and S. C. Hagness, *Computational electrodynamics: the finite-difference time-domain method, 2nd*, Artech House, Boston, (2000).

[4] S. Dey and R. Mittra, "A conformal finite-difference time-domain technique for modeling cylindrical dielectric resonators," *IEEE Transactions on Microwave Theory and Techniques*, **47**(9), 1737-1739, (1999).

[5] C. T. Chan, Q. L. Yu, and K. M. Ho, "Order-N Spectral Method for Electromagnetic-Waves," *Physical Review B*, **51**(23), 16635-16642, (1995).

[6] W. H. Yu and R. Mittra, "A conformal finite difference time domain technique for modeling curved dielectric surfaces," *IEEE Microwave and Wireless Components Letters*, **11**(1), 25-27, (2001).

[7] J. S. Juntunen and T. D. Tsiboukis, "Reduction of numerical dispersion in FDTD method through artificial anisotropy," *IEEE Transactions on Microwave Theory and Techniques*, **48**(4), 582-588, (2000).

[8] R. W. Ziolkowski, "Nonlinear Finite-Difference Time-Domain Modeling of Linear and Nonlinear Corrugated Wave-Guides," *Journal of the Optical Society of America B*, **11**(9), 1565-1575, (1994).

J. B. Judkins and R. W. Ziolkowski, "Finite-Difference Time-Domain Modeling of Nonperfectly Conducting Metallic Thin-Film Gratings," *Journal of the Optical Society of America A*, **12**(9), 1974-1983, (1995).

R. W. Ziolkowski, "The incorporation of microscopic material models into the FDTD approach for ultrafast optical pulse simulations," *IEEE Transactions on Antennas and Propagation*, **45**(3), 375-391, (1997).

J. L. Young and R. O. Nelson, "A summary and systematic analysis of FDTD algorithms for linearly dispersive media," *IEEE Antennas and Propagation Magazine*, **43**(1), 61-77, (2001).

[9] N. Moll and G. L. Bona, "Comparison of three dimensional photonic crystal slab waveguides with two dimensional photonic crystal waveguides: efficient butt coupling into these photonic crystal waveguides," *Journal of Applied Physics*, **93**(9), 4986-4991, (2003).

[10] M. M. Sigalas and E. Chow, "Elliptical air hole waveguides in slab photonic crystals," *Journal of Applied Physics*, **93**(12), 10125-10127, (2003).