ECE 695
Numerical Simulations
Lecture 28: Finite-Difference Time Domain in MEEP

Prof. Peter Bermel
March 27, 2017
Recap: Special Features of MEEP

jdj.mit.edu/meep
nanohub.org/tools/meep

- Arbitrary dimensionality, boundary conditions
- Perfectly matched layers
- Subpixel averaging
- Symmetry and parallelization
- Fully programmable
- Nonlinear and saturable gain media
- Frequency-domain solver
Subpixel Averaging

Symmetry and Parallelization

A. Farjadpour, D. Roundy, A. Rodriguez, M. Ibanescu, P. Bermel, J.D. Joannopoulos, S.G. Johnson,

3/27/2017
Using the Scheme Interface

(set! geometry-lattice (make lattice (size 16 8 no-size)))
(set! geometry (list
    (make block (center 0 0) (size infinity 1)
        (material (make dielectric (epsilon 12))))))
(set! pml-layers (list (make pml (thickness 1.0))))

(set! sources (list
    (make source
        (src (make continuous-src (frequency 0.15))
            (component Ez)
            (center -7 0))))

(set! resolution 10)
(run-until 200
    (at-beginning output-epsilon)
    (at-end output-efield-z))

HDF5 file → plotting program
Outline

• MEEP Interfaces
• MEEP Classes
• Tutorial examples:
  – Waveguide
  – Bent waveguide
  – Multimode ring resonators
  – Isolating individual resonances
  – Kerr nonlinearities
  – Quantifying third-harmonic generation
• Random and correlated random textured structures
MEEP Interfaces

• C/C++ interface
  – Original interface developed
  – Executable compiled with MEEP library
  – Provides tremendous flexibility and speed

• Scheme interface
  – Most widely used interface
  – Built on Scheme and Libctl to make simple problems easy and hard problems soluble
  – Scripted and interpreted at runtime

• nanoHUB GUI
  – Reduces complexity of input at expense of flexibility
MEEP Classes

• Lattice: defines the size of the computational cell, when used
• Material-type: basic materials (3 choices)
  – Medium
  – Perfect-metal
  – Material-function
• Geometric-object: basic structures
  – Block
  – Cylinder
  – Sphere / Ellipsoid
  – Cone
MEEP Classes (Cont’d)

• Symmetry (3 choices)
  – Mirror-symmetry
  – Rotate2-symmetry
  – Rotate4-symmetry

• Pml: key properties
  – Thickness
  – Direction
MEEP Classes (Cont’d)

• Source (3 types):
  – Gaussian-src
  – Continuous-src
  – Custom-src

• Flux-region: used to calculate

\[ P(\omega) = \int dA \cdot [E^*(\omega) \times H(\omega)] \]

• Flux-region properties include:
  – Center
  – Size
  – Direction
  – Weight
Run Functions

- Run-until
- Run-sources
- Run-sources+
- Stop-when-fields-decayed
- (run-k-point T k)
Selected Step functions

• Output-epsilon
• Output-electric-field-z
• Output-total-power
• (output-png component h5topng-options)
Example: Index-Guided Waveguide

(set! geometry-lattice (make lattice (size 16 8 no-size)))
(set! geometry (list (make block (center 0 0) (size infinity 1 infinity) (material (make dielectric (epsilon 12))))))
(set! sources (list (make source (src (make continuous-src (frequency 0.15))) (component Ez) (center -7 0))))
(set! pml-layers (list (make pml (thickness 1.0))))
(set! resolution 10)
(run-until 200 (at-beginning output-epsilon) (at-end output.efield-z))
Example: Index-Guided Waveguide

• Dielectric function is as expected:

• Ez field propagates nicely along waveguide as expected:
Example: Index-Guided Bend

(set! geometry-lattice (make lattice (size 16 16 no-size)))
(set! geometry (list (make block (center -2 -3.5) (size 12 1 infinity) (material (make dielectric (epsilon 12)))) (make block (center 3.5 2) (size 1 12 infinity) (material (make dielectric (epsilon 12))))))
(set! resolution 10)

• Resulting geometry
  (as shown on right):
Example: Index-Guided Bend

(set! pml-layers (list (make pml (thickness 1.0)))
(set! sources (list (make source (src (make continuous-src (wavelength (* 2 (sqrt 12))) (width 20))) (component Ez) (center -7 -3.5) (size 0 1))))
(run-until 200 (at-beginning output-epsilon) (to-appended "ez" (at-every 0.6 output-efield-z)))
Example: Index-Guided Bend

• Can create movie from this (as shown below):
Example: Index-Guided Bend

(define-param no-bend? false)
(set! geometry (if no-bend?
(list (make block (center 0 wvg-ycen)
(size infinity w infinity)
(material (make dielectric (epsilon 12))))))
(list (make block (center (* -0.5 pad) wvg-ycen)
(size (- sx pad) w infinity)
(material (make dielectric (epsilon 12))))
(make block (center wvg-xcen (* 0.5 pad))
(size w (- sy pad) infinity)
(material (make dielectric (epsilon 12)))))
Example: Index-Guided Bend

(define-param nfreq 100)
(define trans ; transmitted flux
  (add-flux fcen df nfreq
    (if no-bend?
      (make flux-region
        (center (- (/ sx 2) 1.5) wvg-ycen) (size 0 (* w 2)))
      (make flux-region
        (center wvg-xcen (- (/ sy 2) 1.5)) (size (* w 2) 0))))
  (define refl ; reflected flux
    (add-flux fcen df nfreq
      (make flux-region
        (center (+ (* -0.5 sx) 1.5) wvg-ycen) (size 0 (* w 2))))))
Example: Index-Guided Bend

Transmission, reflection, and loss spectrum for the bend
Ring Resonators

• Ring resonators are essentially index-guided waveguides bent in on themselves
• Discrete resonant frequencies induced by periodicity
• Free spectral range between modes varies inversely with ring radius
• Radiative losses decay exponentially with ring radius
Ring Resonators

(define-param n 3.4) ; index of waveguide
(define-param w 1) ; width of waveguide
(define-param r 1) ; inner radius of ring
(define-param pad 4) ; padding from waveguide
(define-param dpml 2) ; thickness of PML
(define sxy (* 2 (+ r w pad dpml))) ; cell size
(set! geometry-lattice (make lattice (size sxy sxy no-size)))
(set! geometry (list (make cylinder (center 0 0) (height infinity)
                        (radius (+ r w)) (material (make dielectric (index n)))))
                        (make cylinder (center 0 0) (height infinity)
                        (radius r) (material air))))
(set! pml-layers (list (make pml (thickness dpml)))) (set-param! resolution 10)
Ring Resonators

(define-param fcen 0.15) ; pulse center frequency
(define-param df 0.1) ; pulse width (in frequency)
(set! sources (list (make source (src (make gaussian-src (frequency fcen) (fwidth df))) (component Ez) (center (+ r 0.1) 0)))))

(run-sources+ 300 (at-beginning output-epsilon) (after-sources (harminv Ez (vector3 (+ r 0.1)) fcen df)))
Ring Resonators

Can also access this example on MEEP tool:
https://nanohub.org/tools/meep
Ring Resonators

- Filter diagonalization (harminv) extract resonant frequencies and decay rates:

\[ f(t) = \sum_{k=1}^{N} a_k e^{-j\omega_k t - \Gamma_k t} \]

- Where: \( Q_k = \frac{\omega_k}{2\Gamma_k} \)
- Raw output:

```
harminv0:, frequency, imag. freq., Q, |amp|, amplitude, error
harminv0:, 0.118101575043663, -7.31885828253851e-4, 80.683059081382, 0.00341388964904578, -0.00305022905294175,-0.00153321402956404i, 1.02581433904604e-5
harminv0:, 0.147162555528154, -2.32636643253225e-4, 316.29272471914, 0.0286457663908165, 0.0193127882016469-0.0211564681361413i, 7.32532621851082e-7
harminv0:, 0.175246750722663, -5.22349801171605e-5, 1677.48461212767, 0.00721133215656089, -8.12770506086109e-4-0.00716538314235085i, 1.82066436470489e-7
```
Ring Resonators

• Add the following to ring.ctl:
  (run-until (/ 1 fcen) (at-every (/ 1 fcen 20) output-efield-z))
• Run the following from command line:
  unix% meep fcen=0.118 df=0.01 ring.ctl
  unix% meep fcen=0.147 df=0.01 ring.ctl
  unix% meep fcen=0.175 df=0.01 ring.ctl
  unix% h5topng -RZc dkbluered -C ring-eps-000000.00.h5 ring-ez-* .h5
  unix% convert ring-ez-* .png ring-ez-0.118.gif
Ring Resonators

- End result is to create movies of single ring resonator modes:

\[ \omega = 0.118 \left( \frac{2\pi c}{a} \right) \]
\[ Q = 81 \]

\[ \omega = 0.147 \left( \frac{2\pi c}{a} \right) \]
\[ Q = 316 \]

\[ \omega = 0.175 \left( \frac{2\pi c}{a} \right) \]
\[ Q = 1682 \]
Next Class

• Next time: we will continue finite-difference time domain techniques
• Suggested reference: S. Obayya’s book, Chapter 5, Sections 4-6