ECE 595, Section 10
Numerical Simulations
Lecture 36: MEEP Tutorial II

Prof. Peter Bermel

April 10, 2013

Recap from Monday

- MEEP Interfaces
- MEEP Classes
- Tutorial examples:
  - Waveguide
  - Bent waveguide

Outline

- Recap from Monday
- Examples
  - Multimode ring resonators
  - Isolating individual resonances
  - Kerr nonlinearities
  - Quantifying third-harmonic generation

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

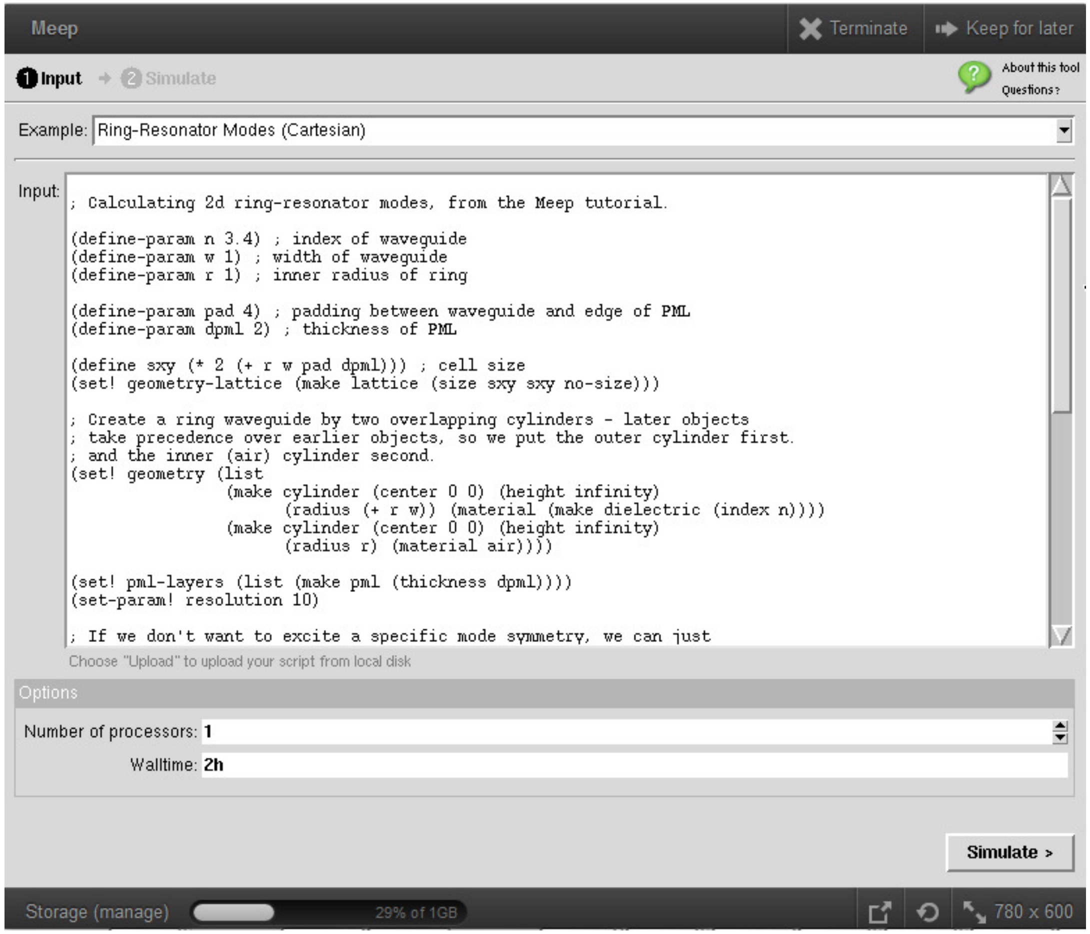
Ring Resonators

```
(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: $Q_k = \omega_k / 2\Gamma_k$
- 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

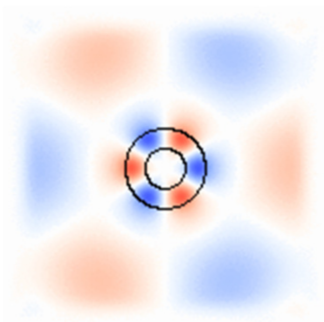
unix% meep fcen=0.175 df=0.01 ring.ctl

unix% h5topng -RZc dkbluered -C ring-eps-000000.00.h5 ring-ez-*.h5

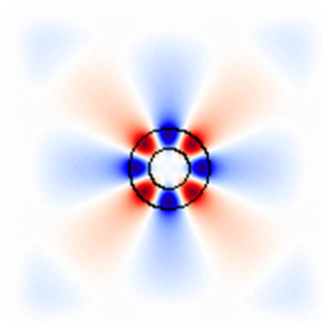
unix% convert ring-ez-*.png ring-ez-0.118.gif

Ring Resonators

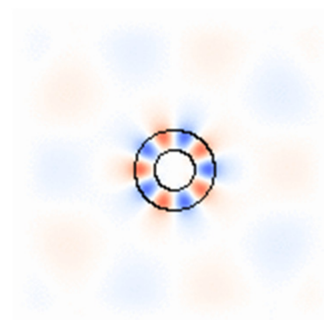
- End result is to create movies of single ring resonator modes:



$\omega = 0.118\ (2\pi c/a)$
$Q = 81$



$\omega = 0.147\ (2\pi c/a)$
$Q = 316$



$\omega = 0.175\ (2\pi c/a)$
$Q = 1682$

Kerr Nonlinearities

- FDTD can simulate Kerr nonlinear media, where $n = n_o + k|\boldsymbol{E}|^2$
- Physically, four-wave mixing will result from this. Two key processes:
  - Sum/difference frequency generation
  - Third-harmonic generation
- Relative rates depend on field strengths, input profile overlaps, and output density of modes

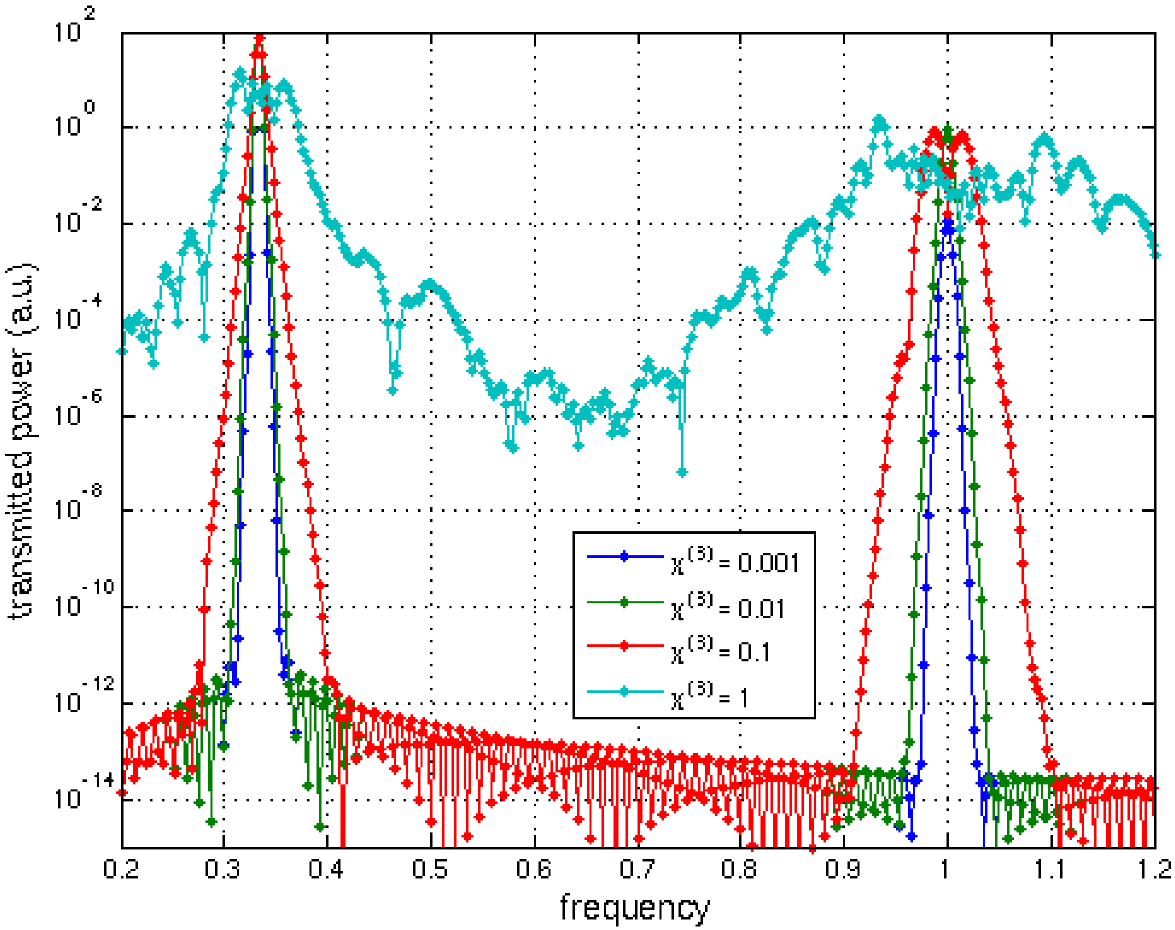
Kerr Nonlinearities

```
(define-param sz 100) ; size of cell in z direction
(define-param fcen (/ 1 3)) ; center frequency of source
(define-param df (/ fcen 20)) ; frequency width of source
(define-param amp 1.0) ; amplitude of source
(define-param k 1e-2) ; Kerr susceptibility
(define-param dpml 1.0) ; PML layer thickness
(set-param! dimensions 1)
(set! geometry-lattice (make lattice (size no-size no-size sz)))
(set! pml-layers (list (make pml (thickness dpml))))
(set-param! resolution 20)
(set! default-material (make dielectric (index 1) (chi3 k)))
```

Kerr Nonlinearities

```
(set! sources (list (make source (src (make gaussian-src
(frequency fcen) (fwidth df))) (component Ex) (center 0 0 (+ (*
-0.5 sz) dpml)) (amplitude amp)))) ; frequency range for flux
calculation

(define-param nfreq 400)

(define-param fmin (/ fcen 2))

(define-param fmax (* fcen 4))

(define trans ; transmitted flux (add-flux (* 0.5 (+ fmin fmax (-
fmax fmin) nfreq (make flux-region (center 0 0 (- (* 0.5 sz)
dpml 0.5)))))
(run-sources+ (stop-when-fields-decayed 50 Ex (vector3 0 0 (-
(* 0.5 sz) dpml 0.5)) 1e-6))

(display-fluxes trans)
```

Kerr Nonlinearities



Third harmonic generation observed; modulation instability for strong nonlinearities

Kerr Nonlinearities

- To quantify THG – add the following to our ctl file:

```
(define trans1 (add-flux fcen 0 1 (make flux-region (center 0 0
(- (* 0.5 sz) dpml 0.5)))))
(define trans3 (add-flux (* 3 fcen) 0 1 (make flux-region
(center 0 0 (- (* 0.5 sz) dpml 0.5)))))
(print "harmonics:, " k ", " amp ", " (first (get-fluxes trans1)) ",
" (first (get-fluxes trans3)) "\n")
```

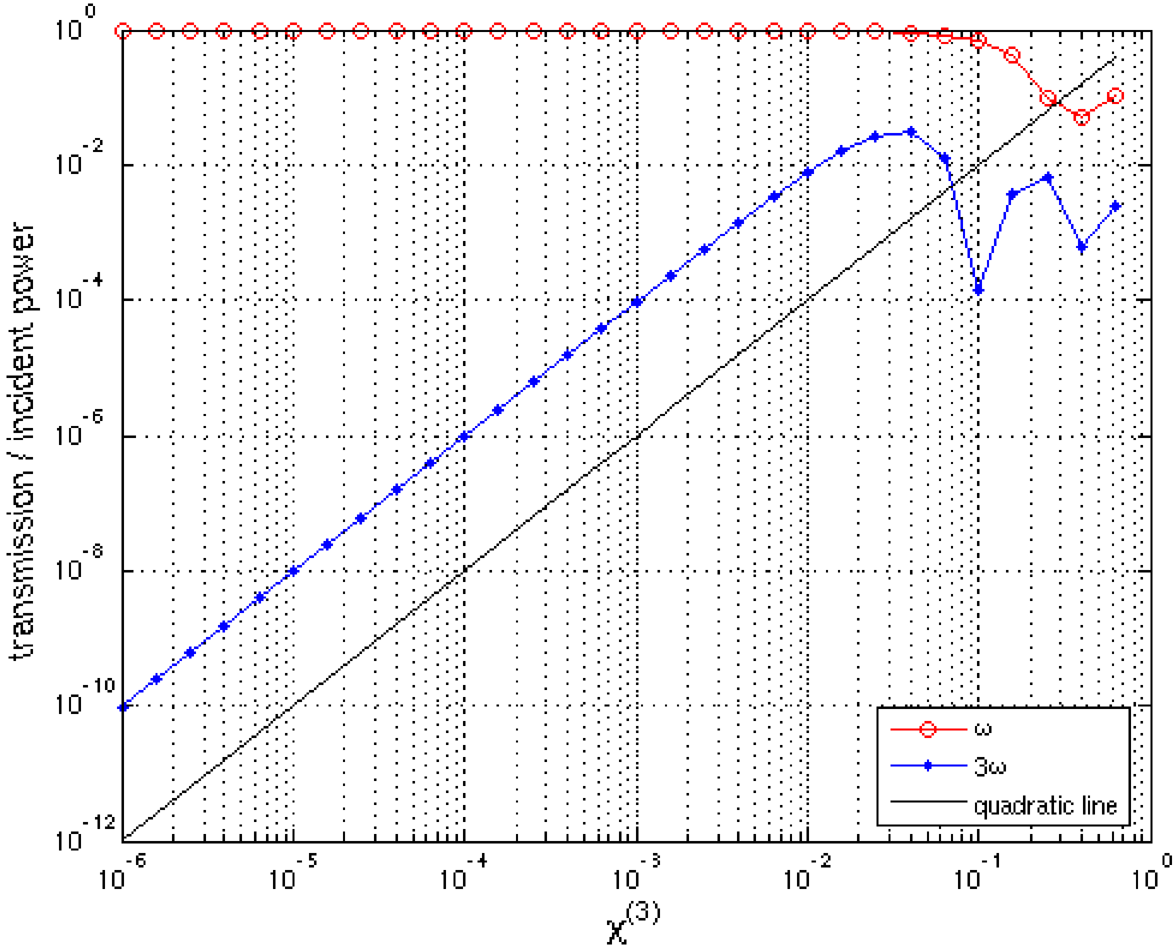
- From command line:

```
unix% (for logk in `seq -6 0.2 0`; do meep k="(expt 10 $logk)"
3rd-harm-1d.ctl |grep harmonics:; done) | tee harmonics.dat
```

- Resulting output:

```
harmonics:, 0, 1.0, 112.62889036581, 1.20863942821229e-16
```

Kerr Nonlinearities



Third harmonic generation rate scales quadratically with nonlinearity

Next Class

- Is on Friday, April 10
- Next time: we will discuss using finite-difference time domain software: MEEP
- Suggested reference: MEEP tutorial, http://jdj.mit.edu/wiki/index.php/Meep_Tutorial