Appendix A

GAAS PHOTONIC CRYSTALS

This chapter describes initial attemps of fabricating GaAs photonic crystal devices for coupling Yb^3+ ions in YVO_4 with optical transition at 984 nm [96]. This only shows the starting point of the fabrication optimization and parameters or procedures described here will be greatly improved in the future. The plan of fabrication of GaAs devices is as following. They are first fabricated on GaAs-AlGaAs-GaAs substrate then undercut to be released from the substrate. We use a nanomanipulator to pick up a device and transfer it deterministically on YVO_4 substrate.

GaAs periodic photonic crystal band diagram

In this simulation, we simulate only a unit lattice of a periodic PhC and see if there are forbidden modes (bandgap) exist so the PhC will reflect those mode propagating inside acting like a mirror. We sweep parameters (width, height, hole period and hole radius) with initial guess chosen based on the strategy descriped in Quan and Loncar [99]. The goal is to find a parameter set that gives a wide bandgap around wavelength 980 nm with GaAs layer height around 240nm which corresponds to the top layer height of GaAs/AlGaAs/GaAs samples we had for tests. The PhC refractive index is set to GaAs and the surrounding medium including holes in PhC are air. The optimized parameters are (width, height, hole period and hole radius) = (1, 0.7, 1, 0.25) that corresponds to (343 nm, 240 nm, 343 nm, 86 nm) if set height to 240 nm. The band diagram with these parameters is shown in figure A.1. The second 3D simulation is to check that the partial periodic PhC acts as a mirror. In this configuration ,shown in figure A.2, PhC with 3 holes is sandwiching a defect at the center. A light source at the center generates field around the wavelength of the bandgap center. A harmonic inversion function recognizes the resonance created with the structure,thus supporting the result of the previous bandgap simulation.

A.1 GaAs photonic crystal fabrication

The first attempt of GaAs photonic crystal was based on the parameters described in the previous section mainly to optimize dry etching recipe and device transfer. The initial dry etching recipe of ICP-RIE etcher is based on Ar:$SiCl_4$=3:10 flow rate. With varying RF/ICP power, the etched sidewall remained rough.

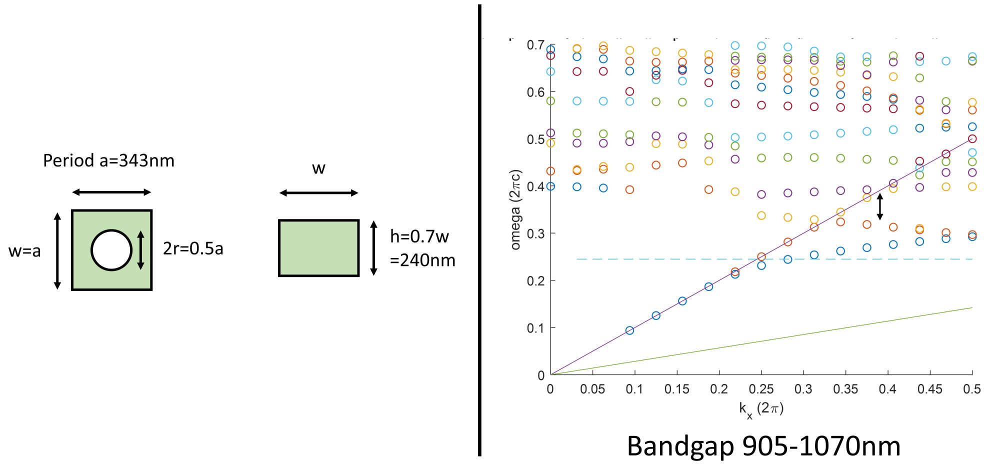

Figure A.1: 3D periodic photonic crystal bandgap simulation.

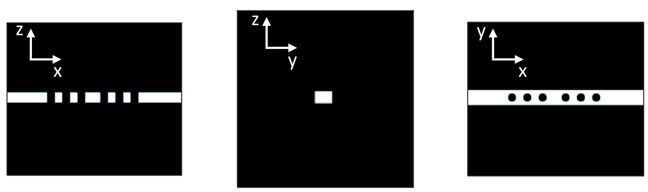

Figure A.2: 3D photonic crystal simulation with defect at the center

AlGaAs undercut

The undercut procedure for 100 nm thickness AlGaAs is shown in the following table **??**.

Photonic crystal transfer by nanomanipulator

Transferring smaller structures or single photonic devices to desired location can be performed using a nanomanipulator. In the FIB/SEM system, this methods allows for device examination in microscopic level and transfer at the same time. In this method, platinum is deposited where we want to hold and is welded to the probe. In order to avoid deposition on the device itself, We patterned two lines along with the PhC and cut and transfer a larger area including the device with FIB as shown in A.3.

Procedure	time (s)
Dip in citric acid	60
Dip in 3.5% HF	3
Dip in water	10
Repeat dipping in HF then water twice	9 total HF dip time
Dip in citric acid	60
Put still wet sample to IPA, gently take out (never blow dry)	

Table A.1: a-Si recipe

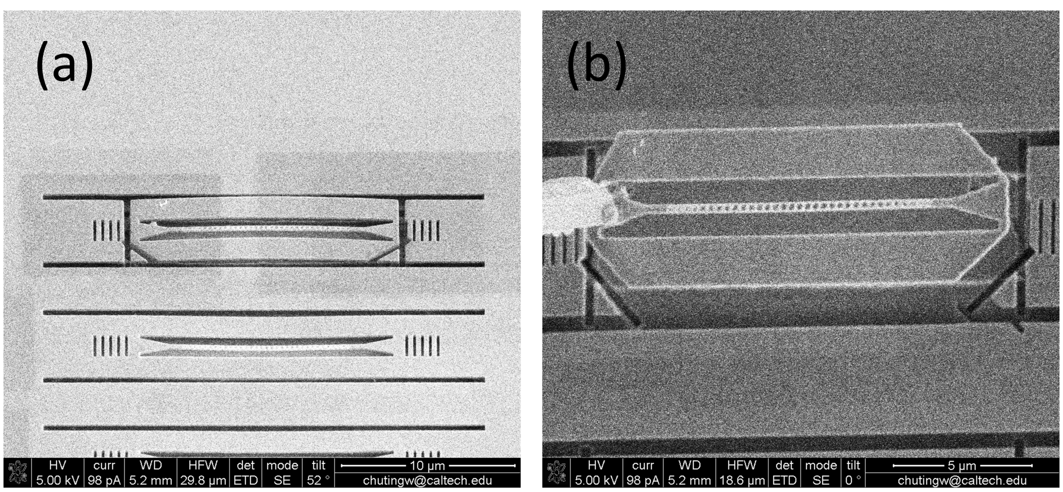

Figure A.3: Transferring a part of devices using a nanomanipulator (a) Cut through between the 2 patterned lines before grating couplers because undercut wasn't enough to detach grating couplers from the substrate (b) The probe at the left side is welded to platinum, deposited around the grating tapered part, and the device is lifted up.

Appendix B

RELATED CODES

Si oxidation time estimation based on desired thickness

```
% Calculates oxidation rate from Equations taken from a chapter written by
% B. E. Deal in Semiconductor materials and process technology handbook
%: for very large scale integration (VLSI) and ultra large scale integration (ULSI)
%/ edited by Gary E. McGuire. (pp. 48-57)

%Input%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
mode=1;%Oxidation condition 1:wet,2:dry
siori=1;%Si orientation 1:(100),2:(111)
T=1000;%Temperature in Celcius
xi=2.5;%Initial oxide thickness (nm)
xsi=250;%Final thickness of Si consumed (nm), e.g. if you have 500nm Si and
        %want to make it 360nm, xsi=140nm

fit=1; %Fit the actual data below to scale the theoretial curve. 1:yes,2:no
%Acutual data of time(hr) and xsi(um)

% %Newer wet oxidation data (after 3/24 when furnace 1 is replaced with new tube)
p1=[4/3,0.24176];%(hr,(oxidized si thickness)um)
p2=[32/60,0.13901];
p3=[1.5,0.26377];

%Newer dry oxidation data
% p1=[0.3,0.01146];
% p2=[0.1667,0.00676];

data=cat(1,p1,p2,p3);


%Basic calculation%%%%%%%%%%%%%%%%%%%%%%%%
oxd=2.20*10^22;%molecular density of SiO2 (/cm^3)
sid=4.99*10^22; %Atomic density of Si(/cm^3)
%Oxide thickness:consumed Si thickness=

xo=xsi*sid/oxd;%Final oxide thickness (nm)
fprintf('Final oxide thickness %f\n\n',xo)

xi=xi/1000;%change units to um
xo=xo/1000;
%%%%%%%%%%%%%%%%%%%%%%%%%

%Parameters for oxidation thickness equation%%%%%%%%%
C1d=7.72*10^2;%(um^2/hr)
C2d=6.23*10^6;%(um/hr)
E1d=1.23; %(eV)
E2d=2.0;

C1w=3.86*10^2;
C2w=1.63*10^8;
E1w=0.78; %(eV)
E2w=2.05;
k=8.617*10^(-5);%eV/K
%%%%%%%%%%%%%%%%%%%%%%%%%%%

if siori==1
    C2d=C2d/1.7;
    C2w=C2w/1.7;
```

Si oxidation time estimation based on desired thickness

```
end

switch mode
    case 1
B=C1w*exp(-E1w/k/(T+273));
B_A=C2w*exp(-E2w/k/(T+273));

    case 2
B=C1d*exp(-E1d/k/(T+273));
B_A=C2d*exp(-E2d/k/(T+273));
end
A=B/B_A;

%Oxidation thickness
tau=xi^2/B+xi/B_A;
t=xo/B_A+xo^2/B-tau;% calculate the time needed to have final ox thickness
t_list=linspace(0,2*t,300);
xo_list=1/2*(-A+sqrt(A^2+4*B*t_list+4*B*tau));
hr=floor(t);
min=floor((t-hr)*60);
sec=floor(((t-hr)*60-min)*60);
fprintf('Theory oxidation time %d:%d:%d\n\n',hr,min,sec)

%Fitting actual data%%%%%%%%%%%%%%
if fit==1
xop=data(:,2)*sid/oxd;%oxide thickness for data
func=@(a,t_data)(1/2*(-a(1)+sqrt(a(1)^2+4*a(2)*t_data+4*a(2)*tau)));
iguess=[A,B];
[beta,R]=nlinfit(data(:,1),xop,func,iguess,statset('MaxIter', 1e6));
xo_scaled=1/2*(-beta(1)+sqrt(beta(1)^2+4*beta(2)*t_list+4*beta(2)*tau));
t_scaled=xo*beta(1)/beta(2)+xo^2/beta(2)-tau;

shr=floor(t_scaled);
smin=floor((t_scaled-shr)*60);
ssec=floor(((t_scaled-shr)*60-smin)*60);
fprintf('Scaled theory oxidation time %d:%d:%d\n\n',shr,smin,ssec)
%%%%%%%%%%%

plot(t_list,xo_list,t_list,xo_scaled,'r',data(:,1),xop,'ro')
legend('Theory','Scaled theory','data')
title('Oxidation calculation')
ylabel('Oxide thickness (um)')
xlabel('time (hr)')
else
plot(t_list,xo_list)
ylabel('Oxide thickness (um)')
xlabel('time (hr)')
end
```

B.1 MEEP codes

2D cylindrical ring resonator simulation


```
(define-param xo 0)
(define-param yo 0)
(define-param h 0.36)  ;height of the ring
(define-param radi 3.75)  ;external radius of the ring
(define-param w 0.30)  ;width of the ring
(define-param res 40)  ;resolution
(define-param fcen 0.9341)     ; pulse center frequency 1070nm
(define-param df 0.005)       ; pulse width (in frequency)
(define-param tim 1000)  ;running time
(define-param dpml 0.5)  ; thickness of PML (one side)
(define-param pad 0.5)  ; thickness of pad b/w PML and edge of the ring (one side)

(define n_Si 3.550) ; refractive index of c-Si at 1070nm
(define n_SiC 2.637) ; refractive index of SiC

(define-param sx (+(* radi 2) (* dpml 2) (* pad 2)))
(define-param sy (+(* radi 2) (* dpml 2) (* pad 2)))

(set! geometry-lattice (make lattice (size sx sy no-size)))

(set! default-material (make medium (index 1))) ; air

(set! geometry (list

            (make cylinder (center xo yo) (radius radi) (height infinity)
                (material (make dielectric (index n_Si)))) ; Si ring

            (make cylinder (center xo yo (/ h 2)) (radius (- radi w)) (height infinity)
                (material (make dielectric (index 1)))) ; center hole


))

(set! pml-layers (list (make pml (thickness dpml))))

(set! sources (list
           (make source
                 (src (make gaussian-src (frequency fcen) (fwidth df)))
                 (component Ez)
                 (center (+ xo (- radi (/ w 2))) yo) (size h h))
            ))

(set-param! resolution res) ;Resolution

(run-sources+ tim
        (at-beginning output-epsilon)
        (after-sources (harminv Ex (vector3 (+ xo (- radi (/ w 2))) yo ) fcen df))
        (after-sources (harminv Ez (vector3 (+ xo (- radi (/ w 2))) yo ) fcen df))

              )

(run-until (/ 1 fcen) (at-every (/ 1 fcen 8) output-efield))
```

3D ring resonator with waveguides simulation


```
(define-param xo 0)
(define-param yo 0)
(define-param h 0.36)  ;height of the ring
(define-param radi 3.75)  ;external radius of the disk
(define-param w 0.3)  ;width of the ring
(define-param sp 0.3)  ;ring waveguide spacing
(define-param wgw 0.3)  ;width of the waveguide

(define-param res 40)  ;resolution
(define-param fcen 0.9285)     ; pulse center frequency 1070nm
(define-param df 0.01)        ; pulse width (in frequency)
(define-param tim 1000)  ;running time
(define-param dpml 0.5)  ; thickness of PML
(define-param pad 0.5)  ; thickness of pad b/w PML and edge of the ring (one side)

(define n_Si 3.550) ; refractive index of Si at 1070nm
(define n_SiC 2.637) ; refractive index of SiC


(define-param sx (+(* radi 2) (* dpml 2) (* pad 2)))
(define-param sy (+(* radi 2) (* dpml 2) (* pad 2) (* sp 2) (* wgw 2)))
(define-param sz (+ h (* dpml 2) (* pad 2)))

(set! geometry-lattice (make lattice (size sx sy sz)))

(set! default-material (make medium (index 1))) ; air

;(define-param guide? true);

(set! geometry (list

            (make block (center xo yo (/ sz -4)) (size sx sy (/ sz 2))
                (material (make dielectric (index n_SiC)))) ; SiC substrate

            (make cylinder (center xo yo (/ h 2)) (radius radi) (height h) (axis 0 0 1)
                (material (make dielectric (index n_Si)))) ; Si ring

            (make cylinder (center xo yo (/ h 2)) (radius (- radi w)) (height h) (axis 0 0 1)
                (material (make dielectric (index 1)))) ; center hole

            (make block (center xo (+ yo (* radi -1) (* sp -1) (* wgw -0.5)) (/ h 2)) (size sx
            wgw h)
                (material (make dielectric (index n_Si)))) ; waveguide

            (make block (center xo (+ yo radi sp (* wgw 0.5)) (/ h 2)) (size sx wgw h)
                (material (make dielectric (index n_Si))))
))

(set! pml-layers (list (make pml (thickness dpml))))

(set! sources (list
           (make source
                 (src (make gaussian-src (frequency fcen) (fwidth df)))
                 (component Ez)
                 (center (+ xo radi (/ w -2)) yo (/ h 4)) (size h h h))
            ))

(set-param! resolution res)

(run-sources+ tim
        (at-beginning output-epsilon)
        (after-sources (harminv Ez (vector3 (+ xo radi (/ w -2)) yo (/ h 4)) fcen df))
              )
```

3D ring resonator with waveguides simulation


```
(run-until (/ 1 fcen) (at-every (/ 1 fcen 4) output-efield))
```

2D grating coupler flux simulation


```
; 2D waveguide with gratings on the right, source in the waveguide polarized in the TM
direction (Ey)

(define-param per 1.00)
(define-param duty 0.2)
(define-param h 0.54)
(define-param dpml 0.5)

(define sx 15)
(define sy 10)

(define-param ref false); if true, it's just a waveguide

(define n_Si 3.550) ; refractive index of Si at 1070nm
(define n_SiC 2.637) ; refractive index of SiC

(define-param fcen 0.9341)     ; pulse center frequency 1070nm
(define-param df 0.01)       ; pulse width (in frequency)

(define-param nfreq 10) ; number of frequencies at which to compute flux
(set-param! resolution 40) ; simulation resolution

(set! geometry-lattice (make lattice (size sx sy no-size)))

(set! geometry
      (append
        (list
            (make block (center 0 (+ (/ sy -4) (/ h 2))) (size sx h) ;Grating
                  (material (make dielectric (index n_Si))))
            (make block (center 0 (* sy (/ -3 8))) (size sx (/ sy 4)) ;Substrate
                  (material (make dielectric (index n_SiC)))))
        (geometric-object-duplicates (vector3 per 0 0) 0 36 ;Grating trenches
                              (make block
                              (center (/ sx -4) (+ (/ sy -4) (/ h 2)))
                                (size (* (- 1 duty) per) h)
                                (material (make dielectric (index 1)))))))


(set! pml-layers (list (make pml (thickness dpml))))


(set! sources (list (make source
              (src (make gaussian-src (frequency fcen) (fwidth
              df)))
              (component Ey) (center (+ (/ sx -2) dpml 0.1) (+ (/ sy -4) (/ h 2))) (size 0 h))))

(define trans ; transmitted flux to y direction
      (add-flux fcen df nfreq
                    (make flux-region
                     (center (- (/ sx 8) (/ dpml 2)) (- (/ sy 2) dpml 0.5)) (size (- (* sx
                     0.75) dpml) 0))))
(define trans2 ;transmitted flux to x
direction
        (add-flux fcen df nfreq
                   (make flux-region
                   (center (- (/ sx 2) dpml 0.5) (/ sy 8)) (size 0 (- (* sy 0.75) 2)))))


(run-sources+
 (stop-when-fields-decayed 50 Ey
                              (vector3 (+ (/ sx -2) dpml 0.1) (+ (/ sy -4) (/ h 2)))
                          1e-3)
                           )
```

2D grating coupler flux simulation


```
;(run-until 50 (at-beginning output-epsilon) (at-every 10 output-efield) )
(display-fluxes trans trans2)
```

2D grating coupler flux normalization (simple waveguide)

C:\Users\Faraon Lab\Documents\Andrei_lab\MEEP\20170418_c-Si_ring_on_SiC\2D grating for TM\2D_Si_wg_on_SiC_TM.ctl   Wednesday, September 11, 2019 3:36 PM

```
; 2D waveguide with gratings on the right, source in the waveguide polarized in the TM direction

(define-param h 0.54)
(define-param dpml 0.5)

(define sx 15)
(define sy 10)

(define n_Si 3.550) ; refractive index of Si at 1070nm
(define n_SiC 2.637) ; refractive index of SiC

(define-param fcen 0.9341)     ; pulse center frequency 1070nm
(define-param df 0.01)       ; pulse width (in frequency)

(define-param nfreq 10) ; number of frequencies at which to compute flux
(set-param! resolution 40) ; simulation resolution

(set! geometry-lattice (make lattice (size sx sy no-size)))
(set! default-material (make medium (index 1))) ; air

(set! geometry
        (list
            (make block (center 0 (+ (/ sy -4) (/ h 2))) (size sx h)
                  (material (make dielectric (index n_Si))))
            (make block (center 0 (* sy (/ -3 8))) (size sx (/ sy 4))
                  (material (make dielectric (index n_SiC))))))


(set! pml-layers (list (make pml (thickness dpml))))


(set! sources (list (make source
              (src (make gaussian-src (frequency fcen) (fwidth
              df)))
              (component Ey) (center (+ (/ sx -2) dpml 0.1) (+ (/ sy -4) (/ h 2))) (size 0 h))))

(define fluxi ;initial flux (corresponding right at the starting point of grating in other
programs)
        (add-flux fcen df nfreq
                   (make flux-region
                   (center (/ sx -4)  (+ (/ sy -4) (/ h 2))) (size 0 (* h 2)))))
(define fluxf ;flux go through the end
        (add-flux fcen df nfreq
                   (make flux-region
                   (center (- (/ sx 2) dpml 0.1) (+ (/ sy -4) (/ h 2))) (size 0 (* h 2)))))

(run-until 150
       (at-beginning output-epsilon)
       (at-end output-efield))

(run-sources+
 (stop-when-fields-decayed 50 Ey
                             (vector3 (- (/ sx 2) dpml 0.1) (+ (/ sy -4) (/ h 2))) 1e-4))
;(run-until (/ 1 fcen) (at-every (/ 1 fcen 8) output-efield))

(display-fluxes fluxi fluxf)
```

3D grating coupler flux simulation

C:\Users\Faraon Lab\Documents\Andrei_lab\MEEP\20170418_c-Si_ring_on_SiC\3D grating for TM\Si_curved_grating_on_SiC_flux_TM.ctl Wednesday, September 11, 2019 3:31 PM

```
; TM mode (Ez)

(define-param wgh 0.24)  ;height of the waveguide + grating
(define-param wgl 0.50)  ;length of the waveguide (doesn't include pad or dpml)
(define-param wgw 0.40)  ;width of the waveguide
(define-param gangle (* (/ 105 180) pi))  ;Full angle of the grating (deg)
(define ga (* (/ gangle 180) pi));Full angle of the grating (rad)
(define-param gper 0.470)  ;grating period
(define-param gduty 0.82)  ;duty cycle of the grating
(define ggap (* gper (- 1 gduty)))  ;width of the grating gap
(define-param gn 6) ;Grating number
(define-param i 0)

(define-param res 40)  ;resolution
(define-param fcen 0.9341)     ; pulse center frequency 1070nm
(define-param df 0.01)       ; pulse width (in frequency)
(define-param tim 100)  ;running time
(define-param dpml 0.5)  ; thickness of PML (one side)
(define-param pad 0.5)  ; thickness of pad b/w PML and edge of the ring (one side)

(define-param n_Si 3.550) ; refractive index of c-Si at 1070nm
(define-param n_SiC 2.637) ; refractive index of SiC
(define-param nfreq 10) ; number of frequencies at which to compute flux

(define sx (- (+ wgl (* gper 6) (* pad 2) (* dpml 2)) ggap))
(define sy (+ (* pad 2) (* dpml 2) (* (sin (/ ga 2)) gper 12)))
(define sz 7.5)

(set! geometry-lattice (make lattice (size sx sy sz)))

(set! default-material (make medium (index 1))) ; air


(set! geometry (list


            (make cylinder (center (- (+ (/ sx -2) dpml pad wgl) ggap) 0 (+ (/ sz -4) (/ wgh
            2))) (radius (* gper 6)) (height wgh) (axis 0 0 1)
                (material (make dielectric (index n_Si)))) ; c-Si cylinder (most outer grating
                one)

            (make cylinder (center (- (+ (/ sx -2) dpml pad wgl) ggap) 0 (+ (/ sz -4) (/ wgh
            2))) (radius (- (* gper 6) (- gper ggap))) (height wgh) (axis 0 0 1)
                 (material (make dielectric (index 1)))) ; air cylinder (make a grating
                 shape)

            (make cylinder (center (- (+ (/ sx -2) dpml pad wgl) ggap) 0 (+ (/ sz -4) (/ wgh
            2))) (radius (* gper 5)) (height wgh) (axis 0 0 1)
                (material (make dielectric (index n_Si)))) ; c-Si cylinder (most outer grating
                one)
            (make cylinder (center (- (+ (/ sx -2) dpml pad wgl) ggap) 0 (+ (/ sz -4) (/ wgh
            2))) (radius (- (* gper 5) (- gper ggap))) (height wgh) (axis 0 0 1)
                 (material (make dielectric (index 1)))) ; air cylinder (make a grating shape)

            (make cylinder (center (- (+ (/ sx -2) dpml pad wgl) ggap) 0 (+ (/ sz -4) (/ wgh
            2))) (radius (* gper 4)) (height wgh) (axis 0 0 1)
                (material (make dielectric (index n_Si)))) ; c-Si cylinder (most outer grating
                one)
            (make cylinder (center (- (+ (/ sx -2) dpml pad wgl) ggap) 0 (+ (/ sz -4) (/ wgh
            2))) (radius (- (* gper 4) (- gper ggap))) (height wgh) (axis 0 0 1)
                 (material (make dielectric (index 1)))) ; air cylinder (make a grating shape)

            (make cylinder (center (- (+ (/ sx -2) dpml pad wgl) ggap) 0 (+ (/ sz -4) (/ wgh
            2))) (radius (* gper 3)) (height wgh) (axis 0 0 1)
```

3D grating coupler flux simulation


```
                (material (make dielectric (index n_Si)))) ; c-Si cylinder (most outer grating
                one)
            (make cylinder (center (- (+ (/ sx -2) dpml pad wgl) ggap) 0 (+ (/ sz -4) (/ wgh
            2))) (radius (- (* gper 3) (- gper ggap))) (height wgh) (axis 0 0 1)
                 (material (make dielectric (index 1)))) ; air cylinder (make a grating shape)

            (make cylinder (center (- (+ (/ sx -2) dpml pad wgl) ggap) 0 (+ (/ sz -4) (/ wgh
            2))) (radius (* gper 2)) (height wgh) (axis 0 0 1)
                (material (make dielectric (index n_Si)))) ; c-Si cylinder (most outer grating
                one)
            (make cylinder (center (- (+ (/ sx -2) dpml pad wgl) ggap) 0 (+ (/ sz -4) (/ wgh
            2))) (radius (- (* gper 2) (- gper ggap))) (height wgh) (axis 0 0 1)
                 (material (make dielectric (index 1)))) ; air cylinder (make a grating shape)

            (make cylinder (center (- (+ (/ sx -2) dpml pad wgl) ggap) 0 (+ (/ sz -4) (/ wgh
            2))) (radius (* gper 1)) (height wgh) (axis 0 0 1)
                (material (make dielectric (index n_Si)))) ; c-Si cylinder (most outer grating
                one)
            (make cylinder (center (- (+ (/ sx -2) dpml pad wgl) ggap) 0 (+ (/ sz -4) (/ wgh
            2))) (radius (- (* gper 1) (- gper ggap))) (height wgh) (axis 0 0 1)
                 (material (make dielectric (index 1)))) ; air cylinder (make a grating
                 shape)


            (make block (center (- (+ (/ sx -2) dpml pad wgl) ggap (* 3 gper (sin (/ ga 2))))
            (* gper 3 (cos (/ ga 2))) (+ (/ sz -4) (/ wgh 2)))
            (size (* 12 gper) (* 6 gper)  wgh) (e1 (cos (/ ga 2)) (sin (/ ga 2)) 0)  (e2 (sin
            (/ ga 2)) (- 0 (cos (/ ga 2))) 0) (e3 0 0 1)
              (material (make dielectric (index 1)))) ; Air block at +y direction (to make the
              grating angle)

            (make block (center (- (+ (/ sx -2) dpml pad wgl) ggap (* 3 gper (sin (/ ga 2))))
            (* gper -3 (cos (/ ga 2))) (+ (/ sz -4) (/ wgh 2)))
            (size (* 12 gper) (* 6 gper)  wgh) (e1 (cos (/ ga 2)) (- 0 (sin (/ ga 2))) 0) (e2
            (sin (/ ga 2)) (cos (/ ga 2)) 0)  (e3 0 0 1)
              (material (make dielectric (index 1)))) ; Air block at -y direction (to make the
              grating angle)


            (make block (center (/ (- (+ dpml pad wgl) sx) 2) 0 (+ (/ sz -4) (/ wgh 2))) (size
            (+ dpml pad wgl 0.2) wgw wgh)
              (material (make dielectric (index n_Si)))) ; c-Si waveguide

            ; (make block (center (- (/ (+ pad dpml) 2) 0.2) 0 (- (/ sz 2) dpml 0.5)) (size (-
            sx (* dpml 2) pad wgl) (- sy (* dpml 2)) 0.2)
              ; (material (make dielectric (index n_Si)))) ; test flux block

            ; (make block (center (- (/ sx 2) dpml 0.2) 0 (- (+ (/ sy 8) (* wgh 1.5) (* dpml
            -0.5)) 0.2)) (size 0.2 (- sy (* dpml 2)) (- (* sz 0.75) dpml (* wgh 3)) )
              ; (material (make dielectric (index n_Si)))) ; test flux block

           (make block (center 0 0 (* sz (/ -3 8))) (size sx sy (/ sz 4))
              (material (make dielectric (index n_SiC)))) ; SiC substrate

))

(set! symmetries (list

(make mirror-sym (direction Y) (phase -1))    ;Use odd mirror symmetry for xz plane at y=0

))

(set! pml-layers (list (make pml (thickness dpml))))
```

3D grating coupler flux simulation

C:\Users\Faraon Lab\Documents\Andrei_lab\MEEP\20170418_c-Si_ring_on_SiC\3D grating for TM\Si_curved_grating_on_SiC_flux_TM.ctl   Wednesday, September 11, 2019 3:31 PM

```
(set! sources (list
           (make source
                 (src (make gaussian-src (frequency fcen) (fwidth df)))
                 (component Ez)
                 (center (+ (/ sx -2) dpml 0.1) 0 (+ (/ sz -4) (/ wgh 2))) (size 0 wgw wgh))
            ))

(set-param! resolution res)

;Setting flux region to compute flux through specified area
(define trans ; transmitted flux to y direction
      (add-flux fcen df nfreq
                    (make flux-region
                     (center (- (/ (+ pad dpml) 2) 0.2) 0 (- (/ sz 2) dpml 0.5)) (size (- sx (*
                     dpml 2) pad wgl) (- sy (* dpml 2)) 0))))

(define trans2 ;transmitted flux to x
direction
        (add-flux fcen df nfreq
                   (make flux-region
                   (center (- (/ sx 2) dpml 0.2) 0 (- (+ (/ sy 8) (* wgh 1.5) (* dpml -0.5))
                   0.2)) (size 0 (- sy (* dpml 2)) (- (* sz 0.75) dpml (* wgh 3)) ))))


(run-until 300 (at-beginning output-epsilon) (at-every 30 output-efield) )
(display-fluxes trans trans2)
```

3D grating coupler flux normalization (simple waveguide)


```
; TM mode (Ez)

(define-param wgh 0.24)  ;height of the waveguide + grating
(define-param wgl 0.50)  ;length of the waveguide (doesn't include pad or dpml)
(define-param wgw 0.40)  ;width of the waveguide
(define-param gangle 0)  ;Full angle of the grating (deg)
(define ga (* (/ gangle 180) pi));Full angle of the grating (rad)
(define-param gper 0.470)  ;grating period
(define-param gduty 0.82)  ;duty cycle of the grating
(define ggap (* gper (- 1 gduty)))  ;width of the grating gap
(define-param gn 6) ;Grating number
(define-param i 0)

(define-param res 40)  ;resolution
(define-param fcen 0.9341)     ; pulse center frequency 1070nm
(define-param df 0.01)       ; pulse width (in frequency)
(define-param tim 100)  ;running time
(define-param dpml 0.5)  ; thickness of PML (one side)
(define-param pad 0.5)  ; thickness of pad b/w PML and edge of the ring (one side)

(define-param n_Si 3.550) ; refractive index of c-Si at 1070nm
(define-param n_SiC 2.637) ; refractive index of SiC
(define-param nfreq 10) ; number of frequencies at which to compute flux

(define sx (- (+ wgl (* gper 6) (* pad 2) (* dpml 2)) ggap))
(define sy (+ (* pad 2) (* dpml 2) (* (sin (/ ga 2)) gper 12)))
(define sz 7.5)

(set! geometry-lattice (make lattice (size sx sy sz)))

(set! default-material (make medium (index 1))) ; air


(set! geometry (list



            (make block (center (/ (- (+ dpml pad wgl) sx) 2) 0 (+ (/ sz -4) (/ wgh 2))) (size
            (* sz 3) wgw wgh)
              (material (make dielectric (index n_Si)))) ; c-Si waveguide



            (make block (center 0 0 (* sz (/ -3 8))) (size sx sy (/ sz 4))
              (material (make dielectric (index n_SiC)))) ; SiC substrate

))

(set! symmetries (list

(make mirror-sym (direction Y) (phase -1))    ;Use odd mirror symmetry for xz plane at y=0

))

(set! pml-layers (list (make pml (thickness dpml))))

(set! sources (list
           (make source
                 (src (make gaussian-src (frequency fcen) (fwidth df)))
                 (component Ez)
                 (center (+ (/ sx -2) dpml 0.1) 0 (+ (/ sz -4) (/ wgh 2))) (size 0 wgw wgh))
            ))

(set-param! resolution res)
```

3D grating coupler flux normalization (simple waveguide)


```
;Setting flux region to compute flux through specified area
(define trans ; transmitted flux right before going into grating
part
      (add-flux fcen df nfreq
                    (make flux-region
                     (center (+ (/ sx -2) dpml pad wgl) 0 (+ (/ sz -4) (/ wgh 2))) (size 0 (*
                     wgw 1.5 ) (* wgh 1.5 )) )))


(define trans2 ;transmitted flux to x
direction
      (add-flux fcen df nfreq
                    (make flux-region
                     (center (- (/ sx 2) dpml 0.1) 0 (+ (/ sz -4) (/ wgh 2))) (size 0 (* wgw
                     1.5 ) (* wgh 1.5 )) )))



(run-until 300 (at-beginning output-epsilon) (at-every 30 output-efield) )
(display-fluxes trans trans2)
```