COMSOL MULTIPHYSICS®

Bandgap Analysis of a Photonic Crystal

SOLVED WITH COMSOL MULTIPHYSICS 3.5a

COMSOL

Bandgap Analysis of a Photonic Crystal

This model performs a bandgap analysis of a photonic crystal similar to the one used in the model "Photonic Crystal" on page 228.

Introduction

The model investigates the wave propagation in a photonic crystal that consists of GaAs pillars placed equidistant from each other. The distance between the pillars determines a relationship between the wave number and the frequency of the light that prevents light of certain wavelengths to propagate inside the crystal structure. This frequency range is called the photonic bandgap (Ref. 2). There are several bandgaps for a certain structure, and this model extracts the bandgaps for the lowest bands of the crystal.

Model Definition

This model is similar to the Photonic Crystal waveguide model. The difference is that in this model the crystal itself is analyzed instead of a waveguide. Because it has a repeated pattern it is possible to use periodic boundary conditions. As a result, only one pillar is needed for this simulation. The model contains a small asymmetry, which is not present in the photonic crystal waveguide model, to remove difficulties of eigenfunctions with identical eigenvalues.

There are two main complications with this bandgap analysis. Firstly, the refractive index of GaAs is frequency dependent. Secondly, the wave vector must be ramped for the band diagram. Although you can solve each of these complications with the eigenvalue solver separately, the two combined make it difficult without a script. However, it is possible to solve a nonlinear problem with the stationary solver, using the eigenvalue as an unknown. The equation for the eigenvalue is a normalization of the electric field, so the average field is unity over the domain. The nonlinear solver finds the correct eigenvalue with an updated refractive index to the found eigenvalue. Furthermore, the parametric solver can sweep the wave vector, $\boldsymbol{k}$. The eigenvalue is equal to the squared wave vector in free space,

$$k_0^2 = \Lambda .$$

The eigenvalue is denoted Λ to avoid confusion with the free-space wavelength, which is denoted with λ_0. The relation between Λ and λ_0 is

$$\lambda_0 = \frac{2\pi}{\sqrt{\Lambda}} .$$

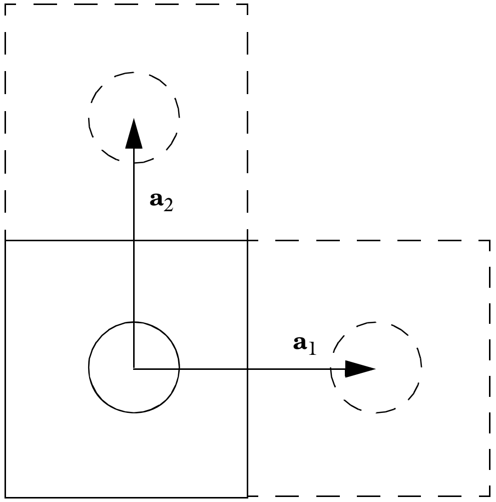
The wave vector for the propagating wave, k, enters the simulation as Floquet periodicity boundary conditions (Ref. 1),

$$E_z(2) = E_z(1)e^{-i\beta} ,$$

where β is a phase factor determined by the wave vector and the distance, d, between the periodic boundaries:

$$\beta = kd .$$

The range for the swept k is determined by the reciprocal lattice vectors of the photonic crystal, and these are determined from the primitive lattice vectors. For a 2D crystal there are two lattice vectors, $\mathbf{a}_1$ and $\mathbf{a}_2$, defined in the following figure.



The reciprocal lattice vectors are calculated from $\mathbf{a}_1$ and $\mathbf{a}_2$ using the relations

$$\begin{aligned}\mathbf{b}_1 &= 2\pi\frac{\mathbf{a}_2\times\mathbf{a}_3}{\mathbf{a}_1\cdot(\mathbf{a}_2\times\mathbf{a}_3)}\\ \mathbf{b}_2 &= 2\pi\frac{\mathbf{a}_3\times\mathbf{a}_1}{\mathbf{a}_1\cdot(\mathbf{a}_2\times\mathbf{a}_3)}\end{aligned},$$

where $\mathbf{a}_3$ is assumed to be the unit vector $\mathbf{e}_z$. When $\mathbf{a}_1$ and $\mathbf{a}_2$ are perpendicular to each other and to $\mathbf{a}_3$, $\mathbf{b}_1$ and $\mathbf{b}_2$ become

$$\begin{aligned}\mathbf{b}_1 &= \frac{2\pi}{|\mathbf{a}_1|}\frac{\mathbf{a}_1}{|\mathbf{a}_1|}\\ \mathbf{b}_2 &= \frac{2\pi}{|\mathbf{a}_2|}\frac{\mathbf{a}_2}{|\mathbf{a}_2|}\end{aligned}.$$

The solution process is rather complicated, and you have to find proper initial conditions for the nonlinear parameter ramp. This is crucial because the system has several solutions, one for each eigenvalue. In COMSOL Multiphysics you can first use the eigenvalue solver to locate an approximate solution at any k-vector. The solution is not exact due to the frequency-dependent refractive index of GaAs. It is possible to repeat the eigenvalue calculation to get closer to the final solution. You can often switch directly to the time-harmonic analysis type after the first iteration of the eigenvalue problem, using that solution as initial guess for the nonlinear solver. With the nonlinear solver you can now perform a nonlinear ramp from $k = 0$ to $k = 0.5$, which represents half the reciprocal vector given by a linear combination of $\mathbf{b}_1$ and $\mathbf{b}_2$ in some predefined direction, for example $(1, 1)$.

The solution steps described above can be automated with a script in MATLAB. The automation is recommended if you are interested in doing several sweeps for different bands and for different k directions.

Results and Discussion

Figure 1 contains the field profile for $k = 0.5$ in the $(1, 1)$ direction. The frequency for this band and k-vector is 395 THz, and the $k = 0$ eigenvalue for this band corresponds to a frequency of 423 THz.

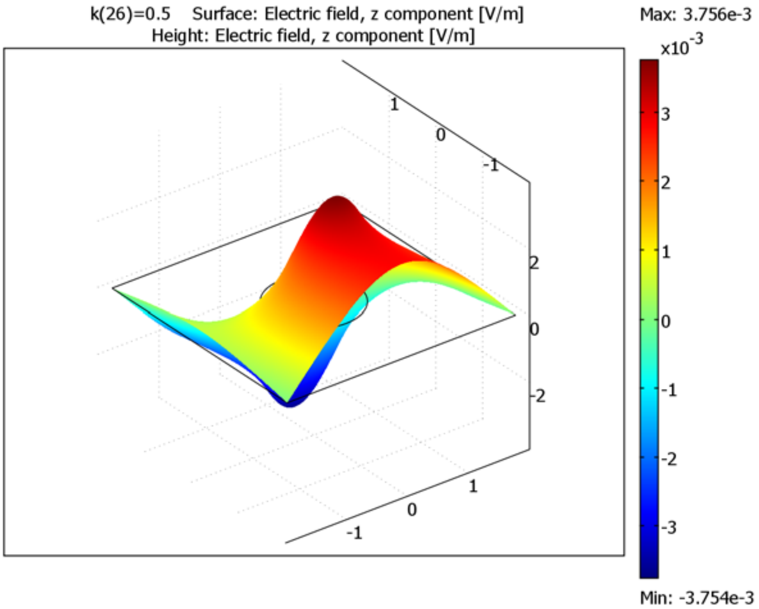


Figure 1: Field profile, Ez, at k = 0.5 for the first band in the (1, 1) direction.

The five lowest bands for the (1, 1) direction appear in Figure 2 below. The three lowest bands lie close and do not really show any true bandgap. A bandgap appears between the third and fourth band.

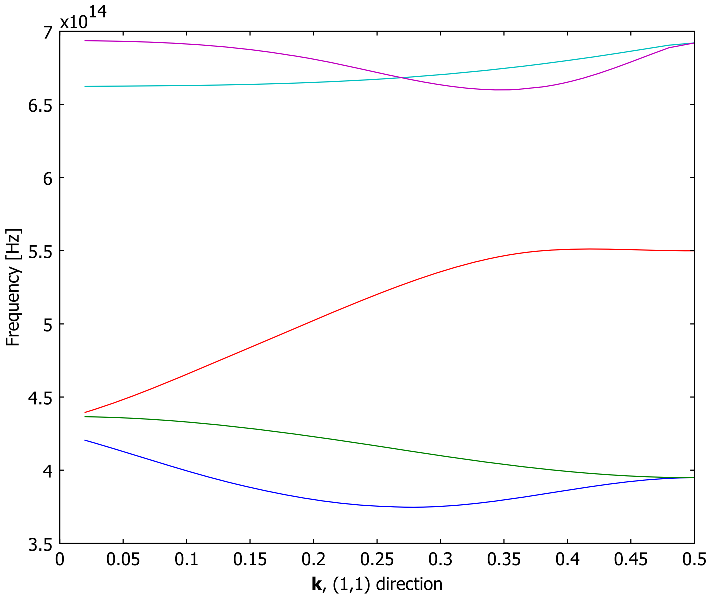


Figure 2: Band diagram of the bands in the (1, 1) direction.

Using a script, it is also possible to perform a sweep in different directions and for several bands, together with the sweep of the k-vector magnitude. Such an analysis results in several band surfaces in k-space. Figure 3 on page 6 shows the five lowest band surfaces obtained using the script created under the section “Modeling Larger

Sweeps With MATLAB" on page 13. There is a large bandgap between the third and fourth band surface, and no waves in this frequency range can propagate in the crystal.

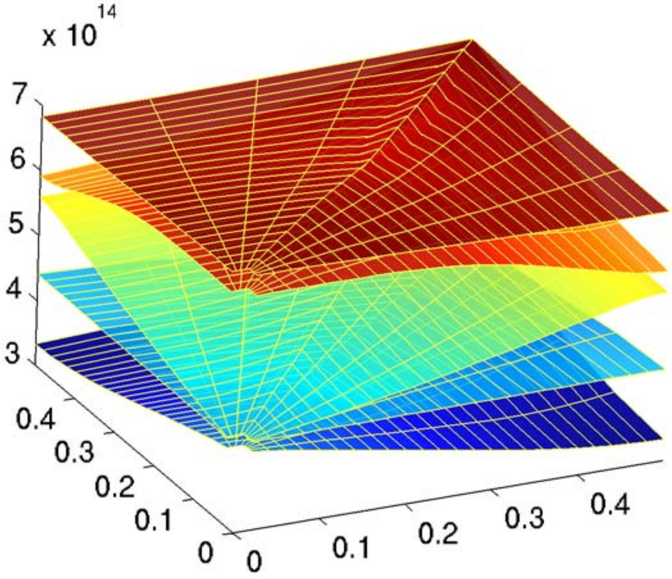


Figure 3: Band surfaces for the five lowest bands for one quadrant in k-space.

References

1. C. Kittel, *Introduction to Solid State Physics*, 7th ed., John Wiley & Sons, New York, 1996.

2. J. D. Joannopoulus, R. D. Meade, J. N. Winn, *Photonic Crystals (Modeling the Flow of Light)*, Princeton university press, 1995.

Model Library path: `RF_Module/Optics_and_Photonics/bandgap_photonic_crystal`

Modeling Using the Graphical User Interface

MODEL NAVIGATOR

1 Select **2D** from the **Space dimension** list.

2 In the list of application modes, select **RF Module>In-Plane Waves>TE Waves>Eigenfrequency analysis**.

3 Click **OK**.

OPTIONS AND SETTINGS

1 From the **Options** menu, choose **Constants**.

2 In the **Constants** dialog box, define the following constants with names, expressions, and (optional) descriptions:

NAME	VALUE	DESCRIPTION
k	0	Fraction of wave vector magnitude
k1	1	1st component of wave direction vector
k2	1	2nd component of wave direction vector
a1x	375[nm]	x-component of 1st lattice vector
a1y	0[nm]	y-component of 1st lattice vector
a2x	0[nm]	x-component of 2nd lattice vector
a2y	375[nm]	y-component of 2nd lattice vector

Scalar Expressions

1 From the **Options** menu, choose **Expressions>Scalar Expressions**.

2 In the **Scalar Expressions** dialog box, define the following variables with names, expressions, and (optional) descriptions:

NAME	EXPRESSION	DESCRIPTION
kx	k*(k1*b1x+k2*b2x)	1st k-component for periodic condition
ky	k*(k1*b1y+k2*b2y)	2nd k-component for periodic condition
b1x	2*pi*a2y/(a1x*a2y-a1y*a2x)	x-component of 1st reciprocal lattice vector
b1y	-2*pi*a2x/(a1x*a2y-a1y*a2x)	y-component of 1st reciprocal lattice vector
b2x	-2*pi*a1y/(a1x*a2y-a1y*a2x)	x-component of 2nd reciprocal lattice vector

NAME	EXPRESSION	DESCRIPTION
b2y	2*pi*a1x/(a1x*a2y-a1y*a2x)	y-component of 2nd reciprocal lattice vector
n_Air	1	Refractive index of air
n_GaAs	3.3285e5[1/m]*lambda0_rfwe+ 3.5031	Dispersive refractive index of GaAs

GEOMETRY MODELING

1 From the **Draw** menu, choose **Specify Objects>Rectangle**.

2 In the dialog box that appears, define the rectangle properties according to the table below.

WIDTH	HEIGHT	BASE	(X, Y)
3.75e-7	3.75e-7	Center	(0, 0)

3 From the **Draw** menu, choose **Specify Objects>Circle**.

4 In the dialog box that appears, define the circle properties according to the table below.

RADIUS	BASE	(X, Y)
7e-8	Center	(0, 0)

5 Finally, click the **Zoom Extents** button on the Main toolbar to simplify the boundary and subdomain selection.

PHYSICS SETTINGS

Boundary Conditions

1 Open the **Boundary Settings** dialog box from the **Physics** menu.

2 Select all boundaries and select the **Periodic condition** from the **Boundary condition** list.

3 From the **Type of periodicity** list, choose **Floquet periodicity**.

4 Enter kx and ky in the edit fields for the **k**-vector.

5 The selected boundaries represents two groups of periodic conditions, one between Boundaries 1 and 4, and the other between Boundaries 2 and 3. You must specify a different periodic pair index to separate them. Select Boundaries 2 and 3 and enter 2 in the **Periodic pair index** edit field.

6 Click **OK**.

Subdomain Settings

1 Open the **Subdomain Settings** dialog box from the **Physics** menu and specify the subdomain settings according to the following table. Prior to specifying the refractive index, n, it is necessary to click the **Specify material properties in terms of refractive index** option button.

PROPERTY	SUBDOMAIN 1	SUBDOMAIN 2
n	`n_Air`	`n_GaAs`

2 Click the **Init** tab and enter `Ez` in the **Ez(t_0)** edit field for both subdomains.

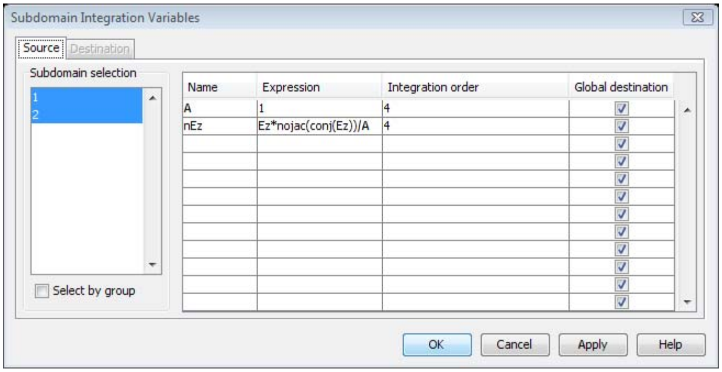
3 Click **OK**.

Integration Coupling Variables

When the nonlinear solver calculates the eigenvalue it is necessary to provide a second equation to the system. This model uses a normalization taking the average value of the squared magnitude of the electric field over the entire domain. The average value is calculated using an integration coupling variable, and an ODE forces that variable equal to 1.

1 From the **Options** menu choose **Integration Coupling Variables>Subdomain Variables**.

2 In the dialog box that appears, define two variables according to the table below. Make sure that the **Global destination** check box is selected for both variables.



Subdomain Integration Variables

Source | Destination

Subdomain selection

1
2

□ Select by group

Name	Expression	Integration order	Global destination
A	1	4	☑
nEz	Ez*nojac(conj(Ez))/A	4	☑
			☑
			☑
			☑
			☑
			☑
			☑
			☑
			☑
			☑
			☑

NAME	SUBDOMAIN	EXPRESSION
A	1,2	1
nEz	1,2	`Ez*conj(Ez)/A`

3 Click **OK**.

MESH GENERATION

1 Open the **Free Mesh Parameters** dialog box from the **Mesh** menu.

2 Select the **Finer** mesh size from the **Predefined mesh sizes** list.

3 Click **Remesh** and then click **OK**.

COMPUTING THE EIGENVALUE SOLUTION

As a first step, calculate a good initial guess for the `freq` ODE variable and the electric field, using the eigenvalue analysis of the photonic crystal. Because the refractive index depends on the wavelength, it is necessary to provide a linearization point for the eigenvalue.

1 Open the **Solver Parameters** dialog box from the **Solve** menu. Make sure that **Eigenfrequency** is chosen in the **Analysis** list.

2 Type `1` in the **Desired number of eigenfrequencies** edit field, and type `4e14` in the **Search for eigenfrequencies around** edit field.

3 Click the **Eigenfrequency** tab, and type `-i*2*pi*4e14` in the **Eigenvalue linearization point** edit field. This is the eigenvalue that corresponds to the eigenfrequency $4\cdot10^{14}$ Hz.

4 Click the **Advanced** tab. Select the **Use Hermitian transpose of constraint matrix and in symmetry detection** check box. Click **OK**.

5 Click the **Solve** button on the Main toolbar. The calculated solution has not reached the correct eigenvalue, because the frequency $4\cdot10^{14}$ Hz was used for the refractive index calculation of GaAs and the returned eigenvalue is close to $4.2\cdot10^{14}$ Hz. It is possible to update the linearization point and solve again, but for the parametric sweep defined later it is more efficient to solve a nonlinear problem. The current eigenvalue solution is good enough as an initial guess for that nonlinear parametric sweep.

COMPUTING THE NONLINEAR SOLUTION

Now you use the eigenvalue solution as initial guess for the nonlinear problem. The use of the nonlinear solver enables sweeping of variables with the parametric solver. Here you ramp the magnitude of the k-vector from 0 to 0.5, which is in units of the reciprocal lattice vectors, $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{b}_3$.

1 Open the **Solver Parameters** dialog box from the **Solve** menu. Choose **Harmonic propagation** from the **Analysis** list.

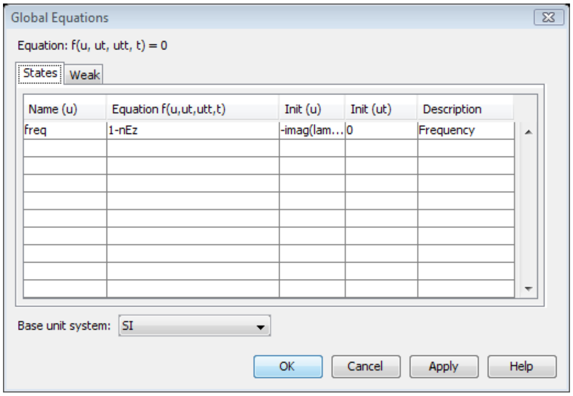
2 Select the **Parametric** solver from the **Solver** list. Type `k` in the **Parameter name** edit field and `range(0,0.02,0.5)` in the **Parameter values** edit field.

3 Click the **Parametric** tab. Select the **Manual tuning of parameter step size** check box, and type 1 in the **Minimum step size** and **Maximum step size** edit fields. Click **OK**.

Global Equations

1 From the **Physics** menu, choose **Global Equations**.

2 In the **Global Equations** dialog box, define the following variable and equation:



Equation: f(u, ut, utt, t) = 0

Name (u)	Equation f(u,ut,utt,t)	Init (u)	Init (ut)	Description
freq	1-nEz	-imag(lam...	0	Frequency

Base unit system: SI

NAME (U)	EQUATION	INIT (U)	DESCRIPTION
freq	1-nEz	-imag(lambda)/(2*pi)	Frequency

3 Click **OK**.

Scalar Variables

1 From the **Physics** menu, choose **Scalar Variables**.

2 In the dialog box that appears, enter the value freq for the nu_rfwe variable. Click **OK**.

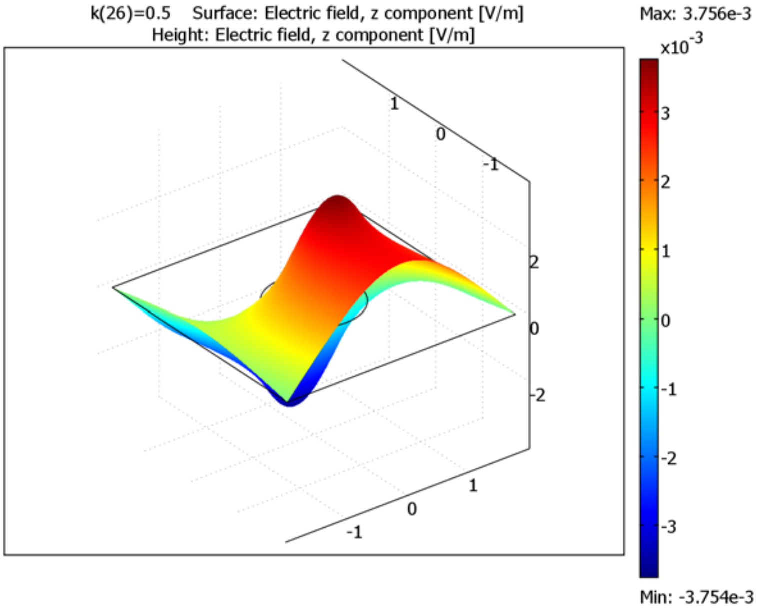
3 Click the **Solve** button on the Main toolbar. COMSOL Multiphysics uses the latest eigenvalue to calculate the initial guess for the ODE variable freq.

During the sweep, the solver uses the latest solution for both freq and Ez, tracing the band starting at $k = 0$ when k increases. You need to choose the step size of k carefully, because a too large step may cause the solution to “jump” to a different band if the bands lie close and have similar field profiles. This is hard to avoid in some situations, because the bands can meet in some points.

POSTPROCESSING AND VISUALIZATION

By default you see the field profile for the current solution in the main window. This profile can be used to check that the k-vector sweep stays on the same band. The field profile changes only slightly when the magnitude of the *k*-vector increases, and a sudden change in field profile between two *k*-vector steps can indicate that the solution has jumped to a different band.

Click the **3D Surface Plot** toolbar button to get a height scale on the default surface plot.

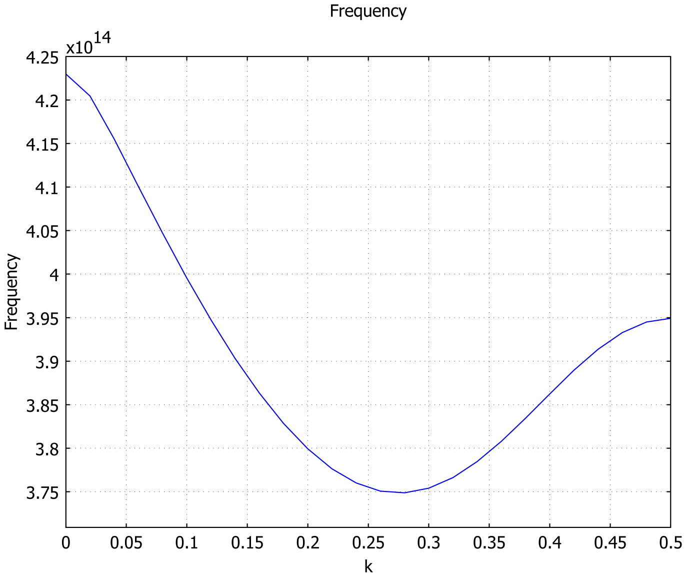


The most interesting plot is the band diagram of the photonic crystal. Follow the steps below to generate the band diagram for one band.

1 Choose **Global Variables Plot** from the **Postprocessing** menu.

2 In the **Global Variables Plot** dialog box, select the variable **Frequency** from the **Predefined quantities** list and press the **>** button. This is the ODE variable `freq`.

3 Click **OK** to see the following plot.



If you plan to do another sweep for a different band, just repeat the steps starting from "Computing the Eigenvalue Solution" on page 10. You have to change the eigenvalue shift and linearization point to match the new band for $k = 0$. If you want to plot the new band in the same figure window, make sure that the **Keep current plot** check box is selected in the **Global Variables Plot** dialog box.

Modeling Larger Sweeps With MATLAB

Performing more than three sweeps is best done using a script or a function. The following script takes the FEM structure, k magnitudes, directions, and eigenfrequency shifts, and performs several sweeps over all the possible variations. If the direction spans several directions in k-space, you can plot the results as a surface for each band.

Note: The instructions below assume that you are running COMSOL Multiphysics with MATLAB.

1 In COMSOL Multiphysics, press Ctrl+F to export the current model to the MATLAB workspace. The model is now available at the command prompt through the structure named `fem`.

2 Type `band_diagram_sweep` to run the following script (available in your path if you have MATLAB), which controls a sweep over five bands and one quadrant in *k*-space.

Note: The script takes a long time to run because it calculates a large number of data points.

```
% k-vector factor to sweep
k = 0.02:0.02:0.5;
% Directions to create band surfaces
dir0 = [ones(1,5) 0.75:-0.25:0 ; 0:0.25:1 ones(1,4)];
% Number of bands
nbands = 5;
% Frequency linearization point
freq0 = 4e14;
% Update magnitude
fem.const{2*strmatch('k',fem.const(1:2:end),'exact')} = k(1);
% Clear frequency variable
nu = [];
% Loop over all directions
for dir = dir0
    % Update direction
    fem.const{2*strmatch('k1',fem.const(1:2:end), ...
                         'exact')} = dir(1);
    fem.const{2*strmatch('k2',fem.const(1:2:end), ...
                         'exact')} = dir(2);
    % Switch to eigenvalue analysis
    fem.appl{1}.prop.analysis = 'eigen';
    fem = multiphysics(fem);
    fem.xmesh = meshextend(fem);
    % Solve eigenvalue problem
    init = asseminit(fem,'init','0');
    fem.sol = femeig(fem, ...
                  'init',init, ...
                  'conjugate','on', ...
                  'solcomp',{'Ez'}, ...
                  'neigs',nbands, ...
                  'shift',-i*2*pi*freq0, ...
                  'eigref',sprintf('-i*2*pi*%g',freq0));
    % Store solution guess for all bands
    sole = fem.sol;
    % Switch to harmonic analysis
    fem.appl{1}.prop.analysis = 'harmonic';
    fem = multiphysics(fem);
    fem.xmesh = meshextend(fem);
    % Loop over all bands
    for ind = 1:length(sole.lambda)
        % Print some progress information
        disp(sprintf(['Direction = (%g,%g), ' ...
                     'Band number %i'],dir,ind));
        init = asseminit(fem,'u',sole,'solnum',ind);
        % Sweep in direction using eigenvalue solution guess
        fem.sol = femnlin(fem, ...
                         'init',init, ...
                         'conjugate','on', ...
                         'pname','k', ...
                         'plist',k, ...
                         'porder',0, ...
                         'pminstep',0.01, ...
                         'pmaxstep',0.01);
        nu = cat(1,nu,postint(fem,'nu_rfwe','edim',0,'dl',1,...
                               'solnum','all'));
    end
end
% Reshape the nu data to have bands in the 3rd dimension
nu = reshape(nu,[nbands size(dir0,2) length(k)]);
nu = permute(nu,[2 3 1]);
% Save data for later use
save band_diagram_data k dir0 nu
% Create a grid for plotting
[K,DIRx] = meshgrid(k,dir0(1,:));
[K,DIRy] = meshgrid(k,dir0(2,:));
X = DIRx.*K;
Y = DIRy.*K;
% Plot one surface for each band
figure(2);
hold on;
for ind = 1:size(nu,3)
    h = mesh(X,Y,real(nu(:,:,ind)));
    % Set transparency and colors
    set(h,'EdgeColor',[1 1 0.3],'FaceColor','interp', ...
          'FaceAlpha',0.6);
end
% Band diagram in (1,1) direction
figure(3);
plot(k,permute(real(nu(5,:,:)),[3 2 1]));
xlabel('k, (1,1) direction');
ylabel('Frequency [Hz]');
```

You can find more information on the script commands in the *COMSOL Multiphysics Reference Guide* and the MATLAB documentation.