How to run photonic crystal slab waveguide calculations with mpb

Daryl Beggs & Sebastian Schulz
beggs@amolf.nl, sas35@st-andrews.ac.uk

Background

If you have not used MPB before, please read the MPB manual:
`http://web.mit.edu/meep_v1.0/mpb/doc/index.html` or
`http://ab-initio.mit.edu/wiki/index.php/MPB_manual`

We have prepared a script to make bandstructure and propagation loss calculations on photonic crystal waveguides simple. The script is called `W1_3D_v1.ctl`. You should get a copy of this from `www.st-andrews.ac.uk/microphotonics`. The scripts are made available under the the free GPL license.

MPB works by defining a unit cell and using periodic boundary conditions it extends this unit cell infinitely in all directions. It then uses a plane-wave method for calculating the eigenfrequencies of the defined structure for any values of the Bloch wavevector **K** that you ask for.

In the case of a photonic crystal slab waveguide, it is convenient to define a rectangular unit cell as shown in figure 1a. The unit cell is indicated by the grey boxes, and both in-plane and out-of-plane directions are shown. As the only true direction of periodicity that we wish to consider is along the waveguide (the *x*-direction), we must make the unit cell large enough in the *y*-direction such that the fields confined to the waveguide don't overlap with the artificially created waveguides in neighbouring supercells. Also, the space above and below the slab (in the *z*-direction) must be large enough such that the field confined to the slab don't overlap with the neighbouring slabs above or below.

Also, as we are only interested in the direction along the waveguide (the truly periodic direction, here along the *x*-axis), the we can confine our bandstructure calculations to this direction only – i.e. we take the wavevectors **K**=(K_x, 0, 0). By default the script considers values of K_x to be 0.3, 0.31, 0.32, …, 0.49, 0.5. Note that these are dimensionless values in units of $a/2\pi$, such that K_x=0.5 is at the bandedge (i.e. the edge of the Brilliuon Zone). This can be altered using the parameters `Ks`, `Ke` and `Kinterp`. `Ks` defines the start point, `Ke` defines the end point, and `Kinterp` how many values are interpolated between the start and end points. See later for how to pass command line arguments.

To run the script just type one of the commands:

```
mpb W1_3D_v1.ctl > yourfile.out
mpb-split 4 W1_3D_v1.ctl > yourfile.out
```

The first runs a single process of MPB, calculating at each K-value in turn. With normal parameters, each K-value should take around 45 min to run, and use around 1GB of RAM. MPB supports multicore processors, so the second command takes advantage of this by splitting the K-values to be calculated across the different processors and aggregating the results at the end. Please not that our script does not currently work with MPI.

These will run all the default parameters (a silicon W1 membrane photonic crystal waveguide) and place the output in `yourfile.out`.

To reformat the output and extract your results, use the commands below:
To extract the bandstructure:

```
awk -f bsfiles.awk yourfile.out > yourfile_bs.dat
```

To extract the group index (assuming you have calculated the group velocities):

```
awk -f ngfiles.awk yourfile.out > yourfile_ng.dat
```

The files `bsfile.awk`, `ngfiles.awk` and `extract_coefficients.sh` can be found in the same location as `W1_3D_v1.ctl`. The file `bsfiles.awk` searches for line with the phrase "zevenfreqs" and prints out the **K**-vectors (actually the value of K_x) and the frequencies (in dimensionless units of a/λ) of all the calculated bands. The file `ngfiles.awk` searches for the lines with "zevenvelocities" and prints out the **K**-vector, frequency and *x*-component of the group velocity of the band of interest (in usual cases b=23 in the file).

If you want to change the default parameters, you can do so at the command line. The format of the commands is (for example):

```
mpb r0=0.35 W1_3D_v1.ctl > yourfile.out
```

which alters the size of the circles from 0.3 (the default) to 0.35. As many of these arguments as you like can be passed, so

```
mpb r0=0.35 r1=0.29 r2=0.34 W1_3D_v1.ctl > yourfile.out
```

alters the sizes of the circles in the first row to 0.29, those in the second to 0.34 and the rest to 0.35. Note when using mpb-split that the number of processors should be the first parameter passed, e.g.

```
mpb-split 8 r0=0.35 r1=0.29 r2=0.34 W1_3D_v1.ctl > yourfile.out
```

The size and positions of the first three rows of holes can be altered by using the parameters `r1`, `r2`, `r3`, `s1`, `s2`, `s3`, `p1`, `p2`, and `p3`. The r-parameters adjust the radius of the holes (see fig. 1b). The s-parameters shift the hole positions away from the waveguide (see fig. 1c). The p-parameters shift the hole positions along the waveguide (see fig. 1d). These parameters are entered as fractions of the lattice constant (either `a1` or `a2` – see below). Also, the width of the waveguide can be altered by using the `w` parameter, which sets the width of the central waveguide to be `w*a*sqrt(3)`; see figure 1e. Thus, setting `w=1` gives a W1 waveguide, and other values give a so-called Ww waveguide. Note, however, that in this definition a W2 waveguide is NOT two rows of holes removed.

Usually all parameters can be entered as dimensionless ones, by parameterising them in some or other units of *a*, the lattice constant. However, in real-life situations, the thickness of the slab that the waveguides are fabricated in is fixed – i.e. it does not scale with the lattice constant, which is determined by your in-plane design. Therefore the lattice constant and slab thickness are entered as real values in nm using the parameters `a1nm` and `hslabnm`.

Internally they are converted into dimensionless values `a1` and `hslab` by dividing them by the value you enter for `a1nm`. Also the lattice can be "stretched" (or "squashed", see fig. 1f) along the direction of the waveguide, by setting the parameter `a2nm` as a different value to `a1nm`.

Tutorial

For this tutorial we calculate the data for figure 4 b from reference [1].
To run the mpb-calculation run one of the following commands:

```
mpb a1nm=410.0 calculation-type=4 r0=0.270 s1=-0.127
W1_3D_v1.ctl > example.out

mpb-split X a1nm=410.0 calculation-type=4 r0=0.270 s1=-0.127
W1_3D_v1.ctl > example.out
```

where X is the number of processors that you want to use.

This will calculate the dispersion curve, group velocity curve and scattering coefficients for the structure. The waveguide is based on a W1 waveguide in a silicon-air membraned structure, with 410nm lattice period, ~220nm diameter holes and the first row of holes is shifted 52nm away from the defect.
As this is a fairly time consuming calculations (15-20 processor hours) we recommend running this over night or over a weekend.
Once the calculation is finished, please move on to the data analysis tutorial.

Data Analysis

Please read the MPB data analysis tutorial first at:
http://ab-initio.mit.edu/wiki/index.php/MPB_Data_Analysis_Tutorial

The calculation described in the previous chapter will have produced two files. One called `epsilon.h5` and example.out. In this chapter we will only consider `example.out`; for help on how to use `epsilon.h5` refer to the h5utils manual at:
http://ab-initio.mit.edu/wiki/index.php/H5utils

`example.out` contains the full results from our calculation, that means the bandstructure, group velocity and scattering coefficient data, alongside information about the structural parameters (slab thickness, period, hole radius etc.). We have provided several scripts that make it easy to reformat the output. First run the following two commands:

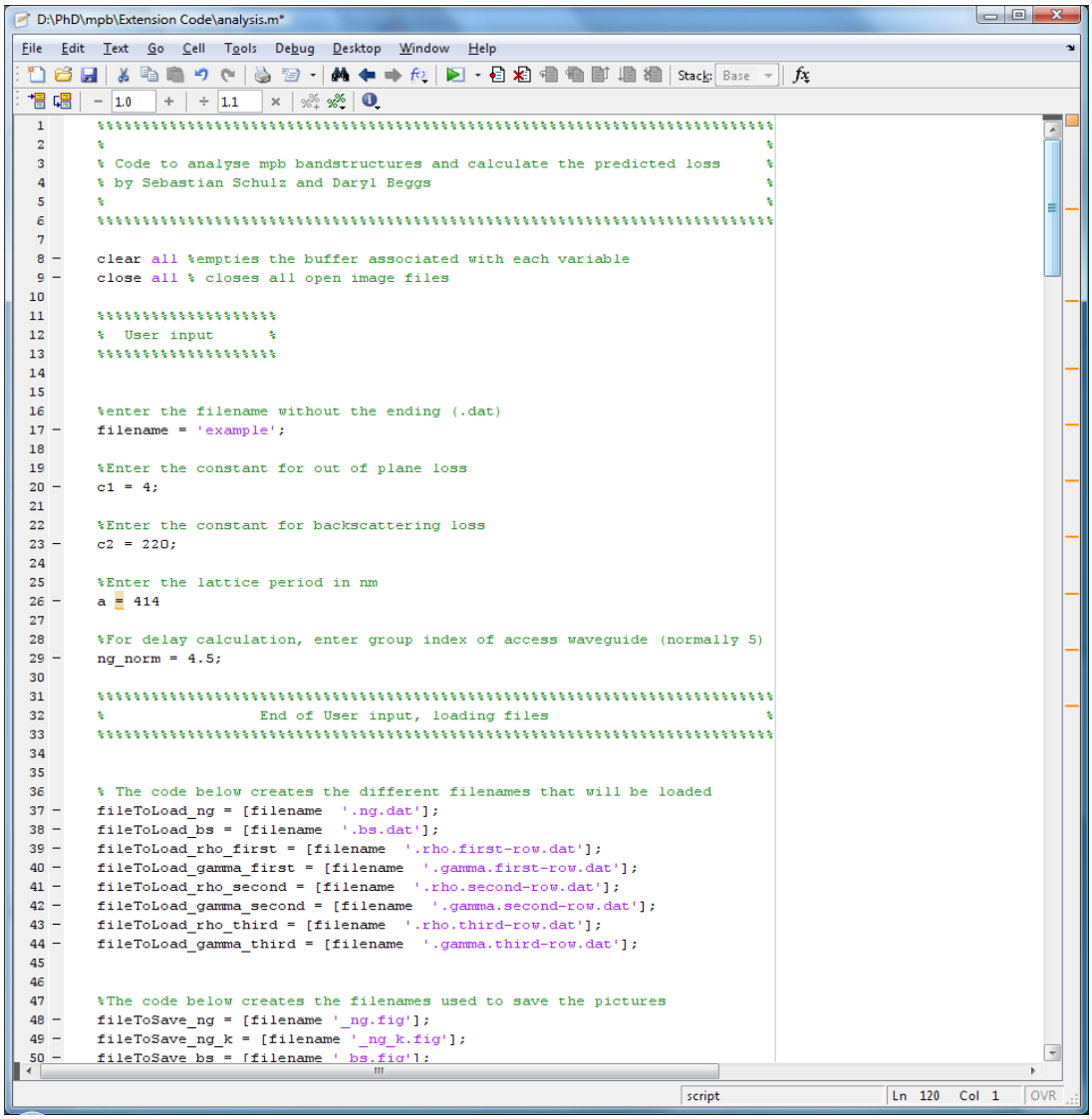
```
awk -f bsfiles.awk example.out > example.bs.dat
awk -f ngfiles.awk example.out > example.ng.dat
```

The first command searches for lines with the phrase "zevenfreqs" (indicating bandstructure data) and prints out the **K**-vectors (actually the value of K_x) and the frequencies (in dimensionless units of a/λ) of all the calculated bands.

The second command searches for the lines with "zevenvelocities" (indicating group velocity data) and prints out the **K**-vector, frequency and x-component of the group index ($n_g = 1/v_g$) of the band of interest (in this case b = 23 in the file).

To extract the scattering coefficients open the file `extract_data.sh`:
This shell script searches for lines with "integral-holes-list" and "gamma" (indicating out of plane scattering) or "rho" (indicating backscattering) and prints out the scattering coefficients for the first three rows of holes (subsequent rows have a negligible contribution to the propagation loss). If you want to run this script for an output file other than `example.out`, open the script in an editor and replace "`example.out`" with the corresponding filename. Then run:
`extract_data.sh`

We are now ready to predict the propagation loss according to equation 2 from reference [1]. For this we have provided an easy to use matlab script, which uses the `*.dat` files from above as input.
Open and run the script: `loss_analysis.m` using matlab. You should see the same as below.



The parameters c_1 and c_2 are explained in more detail in reference [1]. The default values are twice those from reference [1], since we made some small changes to the code since then. As these parameters are dependent on the fabrication quality of your device, the values that provide the best fit to your own experimental data might deviate from the default.
The script works as follows:

1. It reads all the input files generated in the previous steps and creates corresponding variables
2. Next it converts all the frequencies to wavelength in nm using the `period` parameter (therefore if you choose a different period for your calculation you need to change the value here).
3. The different contributions to gamma and rho are then added together
4. Equation 2 from [1] is used to calculate the propagation loss alpha (in dB/cm) for each k-point
5. The bandstructure, group index and propagation loss are plotted and saved to corresponding matlab figure files.

The plot of propagation loss against group index is the theoretical line from fig. 4B in [1]. You will see that there are two plots of propagation loss against group index, with the loss expressed in dB/cm and dB/ns. As explained in [2] it can be very useful to express propagation loss in dB/ns. This value is calculated inside the script using the equation:

```
loss_time = 30*loss./(abs(ng)-ng_norm)
```

where `loss_time` is in dB/ns, `loss` is in dB/cm, `abs(ng)` is the modulus of the group index and `ng_norm` is defined at the beginning of `loss_anaysis.m`. The factor of 30 is the speed of light (3e8m/s = 30cm/ns). In this equation the time that gives the /ns for dB/ns is the extra delay caused by moving through the structure at `ng_norm` instead of ng. The default value of `ng_norm` is 4.5 as most silicon wires and photonic crystals have a group index of 4-5 in the fast light regime and therefore this calculates the loss per ns delay.
However if you want to know the loss relative to light propagating in an alternative medium enter the group index of the alternative medium as `ng_norm`.

For help with the parameters available for the mpb calculation please refer to the table and figure below.

Further Resources

mpb-discuss mailing list (http://ab-initio.mit.edu/cgi-bin/mailman/listinfo/mpb-discuss)
MIT MPB website (http://ab-initio.mit.edu/wiki/index.php/Main_Page)
"Molding the flow of ligh" John D. Joannopoulos, Steven G. Johnson, Joshua N. Winn, and Robert D. Meade. Princeton University Press 2008, http://ab-initio.mit.edu/book/

Reference

parameter			default	notes
`a1nm`	the lattice constant		400	in units of nm

a2nm	lattice constant along the waveguide		a1nm	in units of nm
calculation-type	chooses what to calculate 0: basic calculation of bandstr 1: bandstr + group velocities 2: bandstr + output fields profiles 3: bandstr + group velocities + output fields profiles 4: bandstr + group velocities + scattering coefficients 5: bandstr + group velocities + scattering coefficients + output field profiles		1	
hslabnm	height of the slab		220	in units of nm
index-slab	refractive index of the slab material		3.48	defaults to Si
index-holes	refractive index of the holes and cladding		1	defaults to air
Kinterp	the number of K-points *between* Ks and Ke to calculate		19	integer
Ks	the first K-point to calculate		0.3	in units of a/2π
Ke	the final K-point to calculate		0.5	in units of a/2π
numbands	the number of bands to calculate		30	integer
p1	shift of first row of holes along the waveguide		0	in units of a2
p2	shift of second row of holes along the waveguide		0	in units of a2
p3	shift of third row of holes along the waveguide		0	in units of a2
r0	background radius of holes		0.3	in units of a1
r1	radius of first row of holes		r0	in units of a1
r2	radius of second row of holes		r0	in units of a1
r3	radius of third row of holes		r0	in units of a1
res	resolution of grid points in the plane of the slab		16	per a1
resslab	resolution of the gridpoints out-of-plane of the slab		30	per a1
s1	shift of first row of holes inwards		0	in units of a1
s2	shift of second row of holes inwards		0	in units of a1
s3	shift of third row of holes inwards		0	in units of a1
w	width of the waveguide (i.e. Ww) – NOTE: W2 is not two holes missing		1	in units of a*sqrt(3)
W1band	the band index number of interest		23	integer

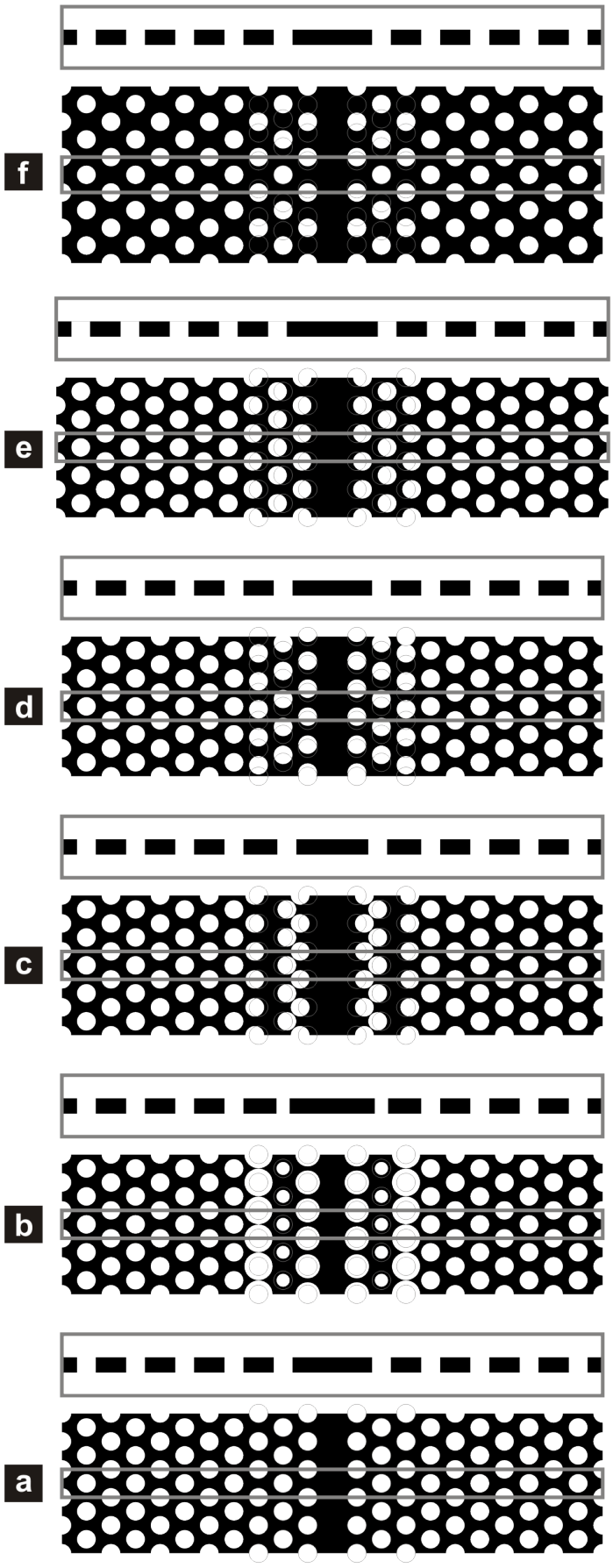


Figure 1

References

[1] L.O'Faolain et al. Optics Express **18,** 27627-27638 (2010)
[2] S.A.Schulz et al. Journal of Optics **12,** 104004 (2010)