

# Triangular lattice

Luis Eduardo Puente and W. Luis Mochán

October 1, 2021

## 1 Introduction

We want to check Photonic using non-orthogonal coordinates. A good example is the hexagonal lattice of cylinders. For low density it must agree with the 2D Maxwell-Garnett formula

$$\epsilon_M = \frac{\epsilon_B(1+f) + \epsilon_A(1-f)}{\epsilon_B(1-f) + \epsilon_A(1+f)} \epsilon_A, \quad (1)$$

where  $\epsilon_M$  is the macroscopic response,  $\epsilon_A$  is the response of the host,  $\epsilon_B$  is the response of the cylinders and  $f$  is the filling fraction. On the other hand, for high density, the system may be viewed as an hexagonal lattice, with primitive vectors

$$\mathbf{a} = (1, 0, 0), \quad \mathbf{b} = (1/2, \sqrt{3}/2), \quad (2)$$

or as a rectangular lattice with unit vectors

$$\mathbf{a} = (1, 0), \quad \mathbf{b} = (0, \sqrt{3}), \quad (3)$$

but with a basis made of one cylinder split into the four vertices and another cylinder at the center. Thus, comparisons may be made to test `Photonic` with non cartesian coordinates.

## 2 Cartesian coordinates, rectangular pixels

To build the code we begin with the required pragmas and packages.

```
# Calculate and plot the macroscopic response
use warnings;
use strict;
use v5.12;
```

```

use Getopt::Long;
use List::Util qw(reduce);
use IO::Prompter;
use PDL;
use PDL::NiceSlice;
use PDL::Graphics::Gnuplot;
use Photonic 0.021;
use Photonic::Utils qw(tile);
use Photonic::Geometry::FromB;
use Photonic::LE::NR2::Haydock;
use Photonic::LE::NR2::EpsTensor;

```

Figure 1: packages

We initialize the required parameters.

```

my $N; # 2N+1 is number of pixels along x
my $epsa; # dielectric function of host
my $tau; # drude lifetime, normalized to w_p=1
my ($wmin, $wmax, $Nw); # Min, max and number of frequencies.
my $radius; # radius of cylinders
my $nh; # number of Haydock coefficients to use

my $options=q(
    'N=i'=>\$N, # 2N+1=unit cell pixels
    'epsa=f'=>\$epsa, # dielectric function of host
    'tau=f'=>\$tau, # Drude lifetime
    'wmin=f'=>\$wmin, # minimum frequency
    'wmax=f'=>\$wmax, # maximum frequency
    'Nw=i'=>\$Nw, # Number of frequencies
    'radius=f'=>\$radius, # radius of cylinders
    'nh=i'=>\$nh, # number of Haydock coefficients to use
);
my %options=(eval $options);
die "Bad option definition; $@" if $@;
GetOptions(%options) or usage($options, "Bad options");
usage($options, "Undefined parameters")
    unless List::Util::all
        {defined $_}
            ($N, $epsa, $tau, $wmin, $wmax, $Nw, $radius, $nh);
$epsa=r2C($epsa); #make complex

```

```
my $w=zeros($Nw)->xlinvals($wmin, $wmax);
```

Figure 2: initialization

We used a simple Drude dielectric function for the metallic cylinders,

$$\epsilon_D = 1 - \frac{\omega_p^2}{\omega(\omega + i/\tau)}, \quad (4)$$

where  $\omega_p$  is the plasma frequency and  $\tau$  the lifetime. We use  $\omega_p$  as our unit of frequency, so that the code is

```
my $epsb=1-1/($w*($w+i()/tau)); # Drude
```

Figure 3: drude

We need a routine to print usage messages.

```
sub usage {
  my ($options, $message)=@_;
  say $message if defined $message;
  say $options;
  exit 1;
}
```

Figure 4: usage

Now we build a geometry using cartesian coordinates.

```
my $lattice_name="Triangular lattice, rectangular pixels";
my $N2=2*$N+1; # pixels of cell;
my $B=zeros($N2+1, $N2+1);
my $r=$B->ndcoords/$N2*pdl(1,sqrt(3)); # coordinates into unit cell
$B= # distance to center or corners smaller than radius
  reduce { $a|$b }
  map {($r-$r(:, $->[0], $->[1]))->abs2->sumover<$radius**2>
    ([0,0], [$N2,0], [$N2, $N2], [0,$N2],[$N,$N])};
$B=$B(0:-2, 0:-2); # remove redundant edges
# Initialize a geometry
my $g=Photonic::Geometry::FromB->new(B=>$B, L=>pdl(1,sqrt(3)));
```

Figure 5: cartesian

We make a routine for the calculation of the dielectric tensor using threading for iterating over frequencies.

```

sub epst {
  my ($geometry, $nh, $epsa, $epsb)=@_;
  my $epst=null;
  my $haydock;
  _epst($epsa, $epsb, $epst, $geometry, $haydock, $nh);
  $epst;
}

BEGIN {
  thread_define '_epst(epsa();epsb());[o]epst(n=2,n=2)), NOtherPars=>3'
=> over {
  my ($epsa, $epsb, $epst, $geometry, $haydock, $nh)=@_;
  if(defined $haydock){
    $epst.=Photonic::LE::NR2::EpsTensor->new(
      geometry=>$geometry, epsA=>$epsa, epsB=>$epsb,
      haydock=>$haydock, nh=>$nh, reorthogonalize=>1
    )->epsTensor;
  }else{
    my $epsObj=Photonic::LE::NR2::EpsTensor->new(
      geometry=>$geometry, epsA=>$epsa, epsB=>$epsb, nh=>$nh,
      reorthogonalize=>1);
    my $rem=$epsObj->epsTensor;
    $epst.=$epsObj->epsTensor;
    $haydock=$epsObj->haydock;
    $_[4]=$haydock; # Modify argument for next threaded calls!
  }
};
}

```

Figure 6: epsilon

We use 6 to obtain the dielectric function.

```
my $epsM=epst($g, $nh, $epsa, $epsb);
```

Figure 7: compute

For comparison, I calculate the Maxwell-Garnett response.

```

my $f=$g->f; # filling fraction
my $epsMG=$epsa*($epsa*(1-$f)+$epsb*(1+$f))
/($epsa*(1+$f)+$epsb*(1-$f));

```

Figure 8: MG

And we plot the results,

```
# Plot the characteristic function
my $gp=PDL::Graphics::Gnuplot->new;
my $B_tiled=tile($B, 3,3);
my $r_tiled=$B_tiled->ndcoords*pd1(1,sqrt(3))/N2;
$gp->plot({title=>"$lattice_name, r=$radius, N=$N",
          colorbox=>0, justify=>1},
          with=>'image', $r_tiled((0)), $r_tiled((1)), $B_tiled);
prompt -void, -single, "Ready?";
```

Figure 9: plot-characteristic

the Drude response,

```
# Plot the drude response
$gp->plot({title=>"Drude response: {/Symbol w}_p=1, {/Symbol t}=$tau",
          xlabel=>"Frequency {/Symbol w}",
          ylabel=>"Dielectric function {/Symbol e}_b"},
          with=>'lines', $w, $sepsb->re, with=>'lines', $w, $sepsb->im);
prompt -void, -single, "Ready?";
```

Figure 10: plot-drude

and the macroscopic response.

```
# Plot the macroscopic response
$gp->plot({title=>"Macroscopic response, $lattice_name, r=$radius, N=$N, nh=$nh",
          xlabel=>"Frequency {/Symbol w}",
          ylabel=>"Macroscopic response {/Symbol e}_M"},
          {legend=>'xx re', with=>'lines'},
          $w, $sepsM((0),(0))->re,
          {legend=>'yy re', with=>'lines'},
          with=>'lines', $w, $sepsM((1),(1))->re,
          {legend=>'xy re', with=>'lines'},
          with=>'lines', $w, $sepsM((0),(1))->re,
          {legend=>'MG re', with=>'lines'},
          with=>'lines', $w, $sepsMG->re,
          {legend=>'xx im', with=>'lines'},
          with=>'lines', $w, $sepsM((0),(0))->im,
          {legend=>'yy im', with=>'lines'},
```

```

        with=>'lines', $w, $epsM((1),(1))->im,
        {legend=>'xy im', with=>'lines'},
        with=>'lines', $w, $epsM((0),(1))->im,
        {legend=>'MG im', with=>'lines'},
        with=>'lines', $w, $epsMG->im);
prompt -void, -single, "Ready?";

```

Figure 11: plot-macroscopic

Finally, we join the pieces

```

<<packages>>
<<initialization>>
<<drude>>
<<cartesian>>
<<compute>>
<<epsilon>>
<<MG>>
<<plot-characteristic>>
<<plot-drude>>
<<plot-macroscopic>>
<<usage>>

```

Figure 12: triangular.pl

## 2.1 Results

In Fig. 13 we show our system, consisting of a triangular lattice. We took the radii of the cylinders as  $r = 0.3a$ , with  $a$  the lattice parameter. We took as a unit cell a centered rectangle with sides  $a$  and  $\sqrt{3}a$ .

In Fig. 14 we show the dielectric function of the cylinders, modeled through the Drude response. We fixed  $\omega_p = 1$  and took  $\tau = 1$ . In Fig. 15 we show the macroscopic response calculated with **Photonic** for the triangular lattice of Drude cylinders within vacuum shown in Fig. 13, each with the Drude response shown in 14. It may be appreciated that the system is isotropic, with the same response along the  $x$  and  $y$  directions, except for a small structure in  $\epsilon_{yy}$  at  $\omega = 0.56\omega_p$  and smaller height at the main resonance, which is slightly blue-shifted with respect to the MG and  $xx$  responses. The differences may be due to the use of rectangular pixels with an aspect ratio  $1 : \sqrt{3}$ . Similarly, the off-diagonal terms are zero except for small errors at resonance. For comparison purposes, we also show the

Triangular lattice, rectangular pixels,  $r=.3$ ,  $N=100$

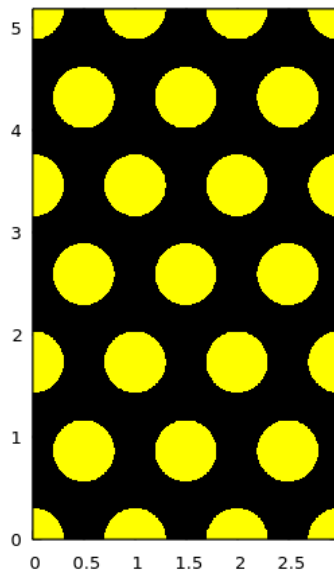


Figure 13: The system consists of a triangular lattice of cylinders with a centered rectangular unit cell. The radius of the cylinders is  $r = 0.3a$ .

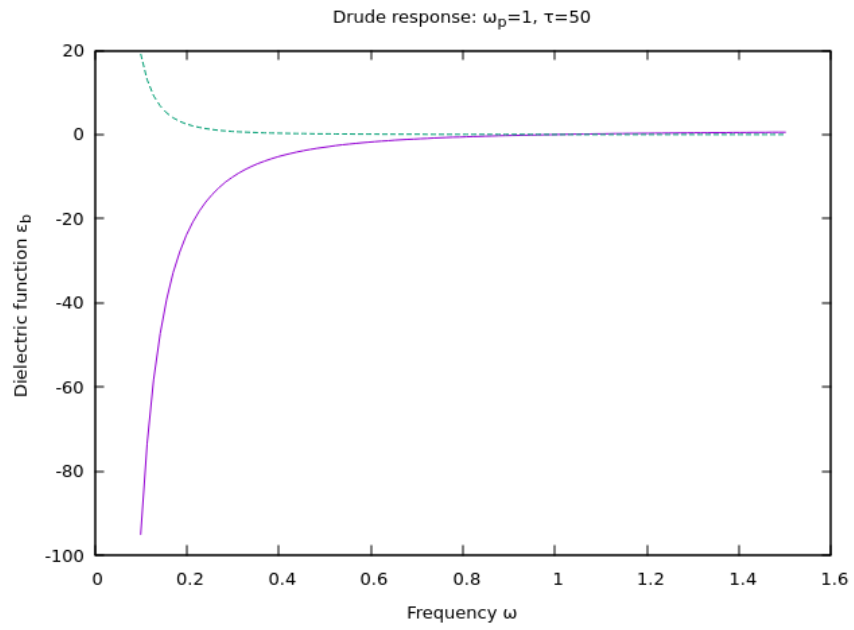


Figure 14: The cylinders are described by a Drude dielectric function. We normalize the plasma frequency  $\omega_p = 1$  and used a lifetime  $\tau = 50$ .



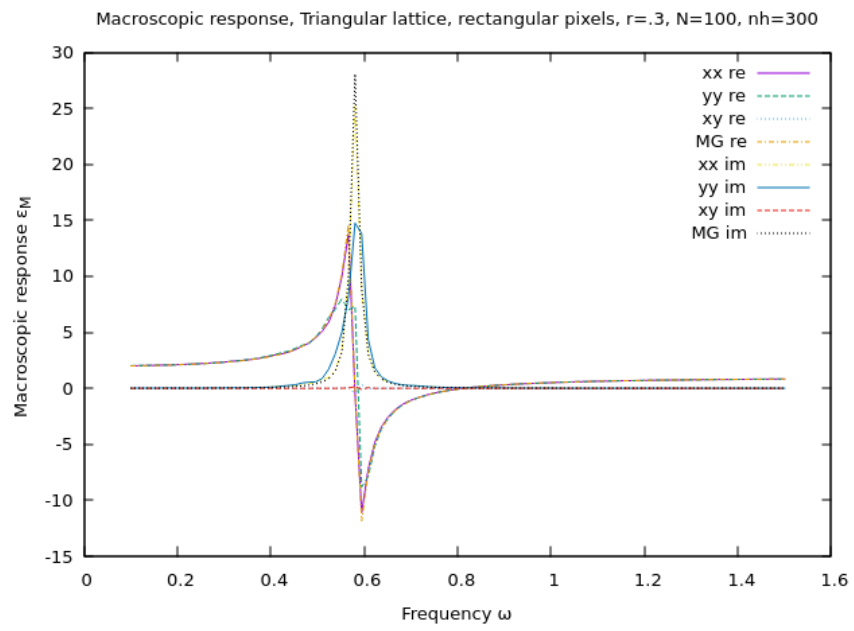


Figure 15: Macroscopic response for the system shown in Fig. 13. We show the real and imaginary parts of  $\epsilon_{xx}$ ,  $\epsilon_{xy} = \epsilon_{yx}$ , and  $\epsilon_{yy}$ , as well as the result of Maxwell-Garnett theory.

Maxwell-Garnett response. It is qualitatively similar but with a smoother line and a larger height. The resonance is redshifted with respect to the dipolar surface plasma frequency  $\omega_{sp} = \omega_p/\sqrt{2}$ .

To obtain the results above we ran the program as follows:

```
./triangular.pl -N 100 -epsa 1 -tau 50 -radius .3\  
                -wmin .1 -wmax 1.5 -Nw 100 -nh 300
```

Figure 16: run.3

In order to study higher filling fractions, we run the program with a larger radius  $r = 0.45a$ . In Fig. 17 we show the system. Notice that neighboring cylinders are very close to each other. In Fig. 18 we show

Triangular lattice, rectangular pixels,  $r=.45$ ,  $N=100$

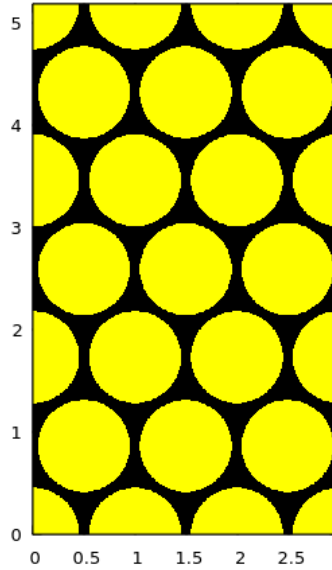


Figure 17: Triangular lattice, as in Fig. 13 but for a larger radius  $r = 0.45a$ .

the corresponding macroscopic response. The red shift is much larger, but nevertheless, the difference between the Maxwell-Garnett and the **Photonic** result is very small. Furthermore, the contributions from larger multipoles are not very large, as compared to the case in, for example, a square lattice.

To obtain these we ran the program as follows:

```
./triangular.pl -N 100 -epsa 1 -tau 50 -radius .45\  
                -wmin .1 -wmax 1.5 -Nw 100 -nh 300
```

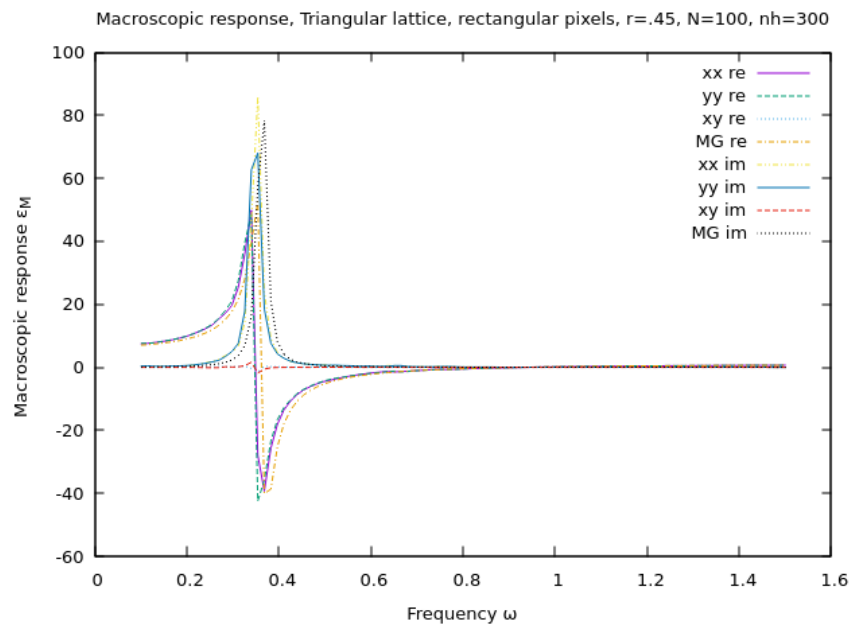


Figure 18: Macroscopic response as in Fig. 15 but for a larger radius  $r = 0.45a$ , as in Fig. 17.

### 3 Cartesian coordinates, square pixels

To minimize errors due to the use of rectangular pixels, we change the code for the characteristic function, increasing the number of pixels vertically.

```
my $lattice_name="Triangular lattice, square pixels";
my ($N2x, $N2y)= map {2*$_+1}
                  my ($Nx, $Ny)=map {sprintf "%d", $_}
                  ($N, sqrt(3)*$N); #convert to integer
my $B=zeros($N2x+1, $N2y+1);
my $r=$B->ndcoords/pdl($N2x,$N2y)*pdl(1,sqrt(3)); # coordinates
$B= # distance to center or corners smaller than radius
    reduce { $a|$b }
    map {($r-$r(:,$_->[0], $_->[1]))->abs2->sumover<$radius**2>
        ([0,0], [$N2x,0], [$N2x, $N2y], [0,$N2y],[$Nx,$Ny]);
    $B=$B(0:-2, 0:-2); # remove redundant edges
# Initialize a geometry
my $g=Photonic::Geometry::FromB->new(B=>$B, L=>pdl(1,sqrt(3)));
```

Figure 19: cartesian-1

We have to modify the code to show the characteristic function.

```
# Plot the characteristic function
my $gp=PDL::Graphics::Gnuplot->new;
my $B_tiled=tile($B, 3,3);
my $r_tiled=$B_tiled->ndcoords*pdl(1,sqrt(3))/pdl($N2x,$N2y);
$gp->plot({title=>"$lattice_name, r=$radius, N=$N",
          colorbox=>0, justify=>1},
         with=>'image', $r_tiled((0)), $r_tiled((1)), $B_tiled);
prompt -void, -single, "Ready?";
```

Figure 20: plot-characteristic-1

The complete code then becomes

```
<<packages>>
<<initialization>>
<<drude>>
<<cartesian-1>>
<<compute>>
<<epsilon>>
```

```
<<MG>>  
<<plot-characteristic-1>>  
<<plot-macroscopic>>  
<<usage>>
```

Figure 21: triangular1.pl

### 3.1 Results

In Fig. 22 we show our system for  $r = 0.3a$ . The figure looks similar to 13, so our characteristic function seems correct. In Fig. 23 we show the

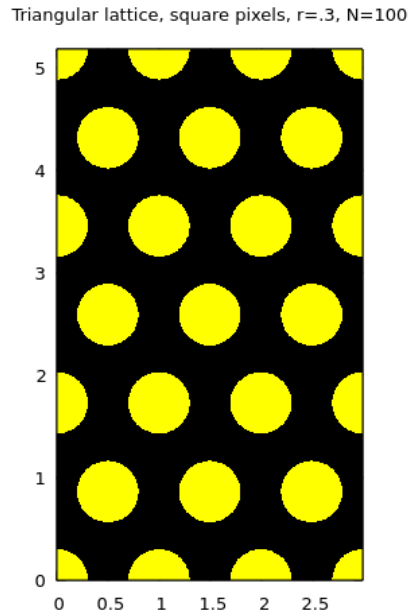


Figure 22: Triangular lattice of cylinders with  $r = 0.3a$ , as in Fig. 13 but with almost square pixels.

corresponding macroscopic response. The system looks more isotropic and the response is smoother than with rectangular pixels.

To obtain the results above we ran the program as follows:

```
./triangular1.pl -N 100 -epsa 1 -tau 50 -radius .3\  
-wmin .1 -wmax 1.5 -Nw 100 -nh 300
```

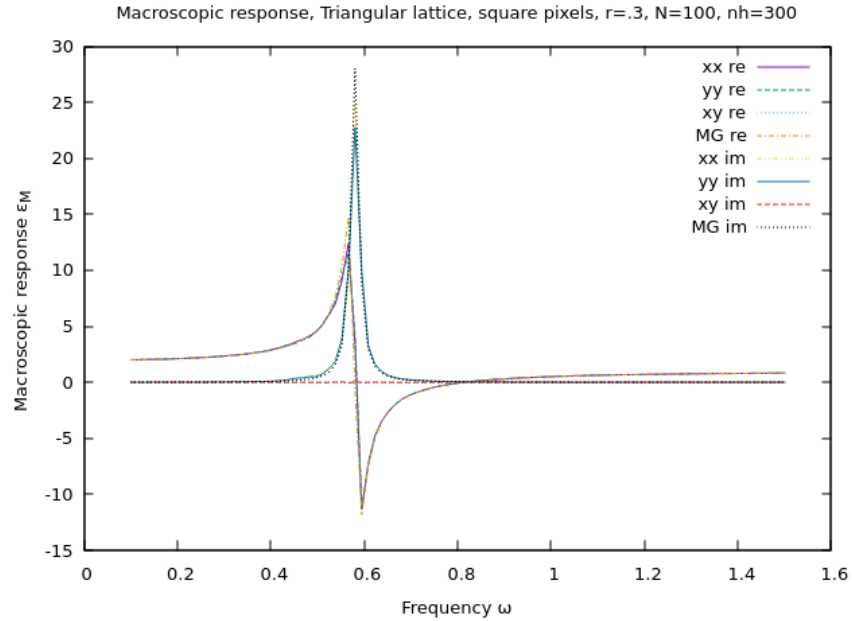


Figure 23: Macroscopic response, as in Fig. 15 but with almost square pixels.

Figure 24: name.3-1

In obtain results also for a larger fraction corresponding to a radius  $r = 0.45a$ . In Fig. 25 we show the system, which seems identical to Fig. 17. In Fig. 26 we show the corresponding macroscopic response. The results agree with those of Fig. 18 and are slightly better.

To obtain these we ran the program as follows:

```
./triangular1.pl -N 100 -epsa 1 -tau 50 -radius .45\  
-wmin .1 -wmax 1.5 -Nw 100 -nh 300
```

Figure 27: run.45-1

## 4 Primitive cell

Now, I want to compare the results above to those resulting from a primitive cell with primitive vectors  $\mathbf{a} = (1, 0)$  and  $\mathbf{b} = (1, \sqrt{3})/2$ . Notice that to obtain the actual Cartesian coordinates I have to perform linear combinations of the primitive vectors and the vector indices.

Triangular lattice, rectangular pixels,  $r=.45$ ,  $N=100$

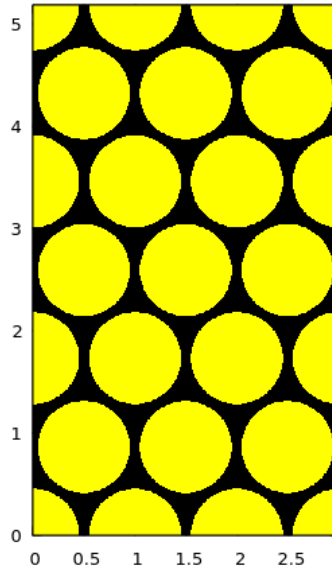


Figure 25: Triangular lattice, as in 17 but using nearly square pixels.

```

my $lattice_name="Triangular lattice, primitive cell";
my $N2=2*$N+1; # pixels of cell;
my $units=pd1[[1,0], [1/2, sqrt(3)/2]];
my $L=pd1(1,1);
my $B=zeros($N2+1, $N2+1);
# $B->ndcoords has indices a,n,m, with a=0,1 numbering primitive
# vectors and n,m numbering pixels
# $units has coordinates i,a with i Cartesian indices and a numbering
# primitive vectors. Thus, to sum over a we have to transpose units
# and add a dummy index (i) to $B->ndcoords
my $r=($B->ndcoords->dummy(1)/$N2*$units->mv(1,0))->sumover; # coordinates
$B= reduce { $a|$b } # distance to corners smaller than radius
  map {($r-$r(:, $_->[0], $_->[1]))->abs2->sumover<$radius**2}
  ([0,0], [$N2,0], [$N2, $N2], [0,$N2]);
$B=$B(0:-2, 0:-2); # remove redundant edges
# Initialize a geometry
my $g=Photonic::Geometry::FromB->new(B=>$B, L=>$L, units=>$units);

```

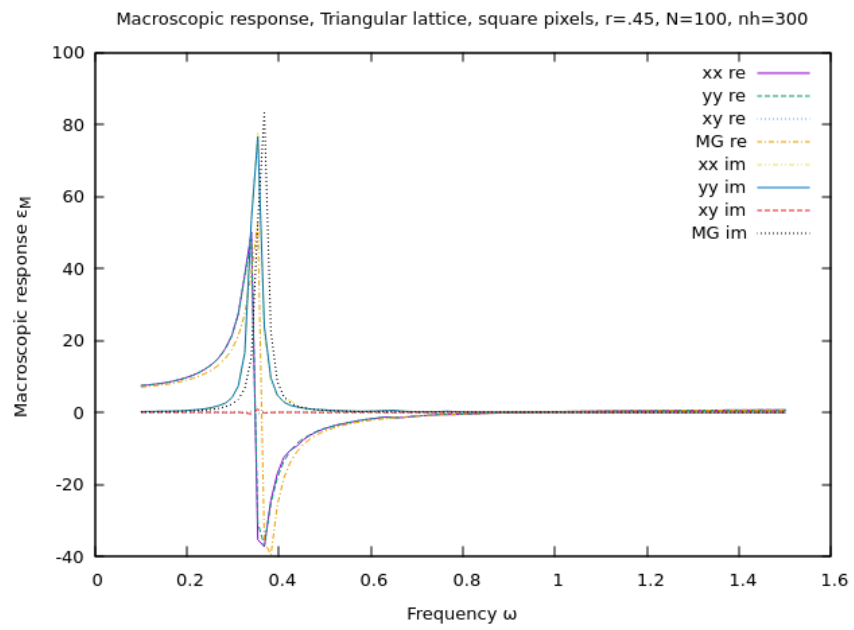


Figure 26: Macroscopic response, as in Fig. 18 but calculated with nearly square pixels.



Figure 28: primitive

I plot the characteristic function.

```
# Plot the characteristic function
my $gp=PDL::Graphics::Gnuplot->new;
my $B_tiled=tile($B, 3,3);
my $r_tiled=($B_tiled->ndcoords->dummy(1)/$N2*$units->mv(1,0))->sumover;
$gp->plot({title=>"$lattice_name, r=$radius, N=$N",
          colorbox=>0, justify=>1},
          with=>'image', $r_tiled((0)), $r_tiled((1)), $B_tiled);
prompt -void, -single, "Ready?";
```

Figure 29: plot-characteristic-2

The code is then

```
<<packages>>
<<initialization>>
<<drude>>
<<primitive>>
<<compute>>
<<epsilon>>
<<MG>>
<<plot-characteristic-2>>
<<plot-macroscopic>>
<<usage>>
```

Figure 30: triangular2.pl

## 4.1 Results

In Fig. 31 we show our system for  $r = 0.3a$ . The figure looks similar to 13, so our characteristic function seems correct, though it naturally is an oblique parallelogram and not a rectangle. In Fig. 32 we show the corresponding macroscopic response. To obtain the results above we ran the program as follows:

```
./triangular2.pl -N 100 -epsa 1 -tau 50 -radius .3\
-wmin .1 -wmax 1.5 -Nw 100 -nh 300
```

Figure 33: run.3-2

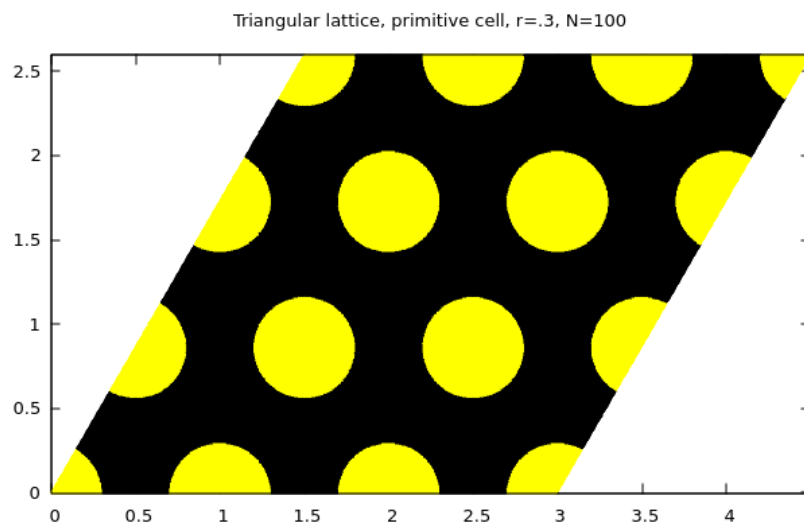


Figure 31: Triangular lattice of cylinders with  $r = 0.3a$ , as in Fig. 13 but with a primitive unit cell with non-orthogonal primitive vectors.

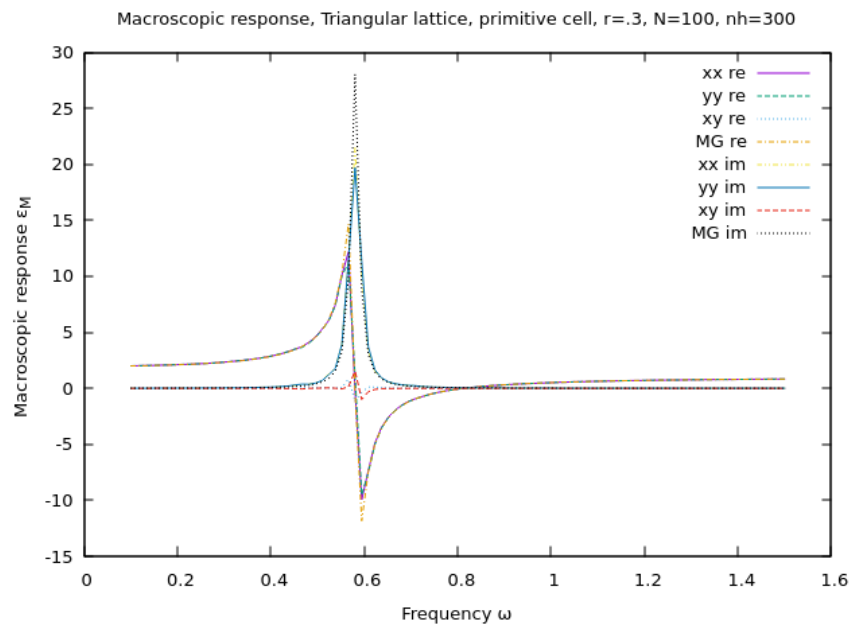


Figure 32: Macroscopic response, as in Fig. 15 but with a primitive non-orthogonal cell as in Fig. 31.

We obtain results also for a larger fraction corresponding to a radius  $r = 0.45a$ . In Fig. 34 we show the system, which corresponds to Fig. 17. In Fig. 35 we show the corresponding macroscopic response. The results

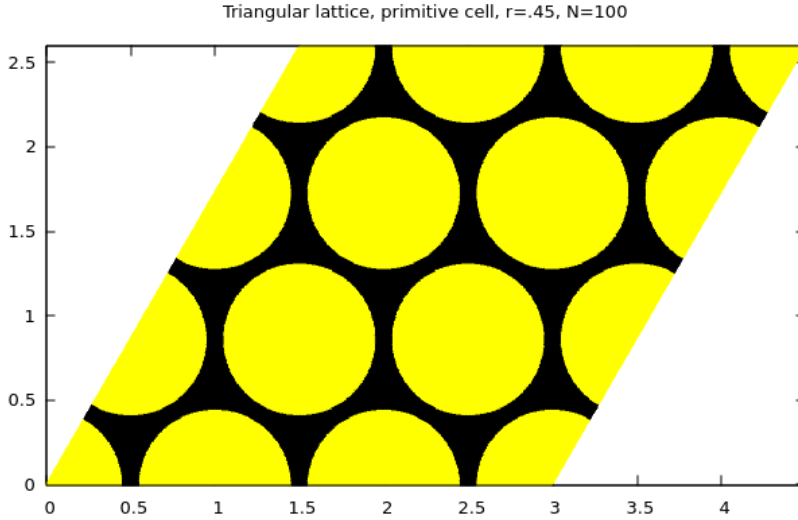


Figure 34: Triangular lattice, as in 17 but using a primitive nonorthogonal cell.

agree with those of Fig. 18 and are slightly better.

To obtain these we ran the program as follows:

```
./triangular2.pl -N 100 -epsa 1 -tau 50 -radius .45\  
-wmin .1 -wmax 1.5 -Nw 100 -nh 300
```

Figure 36: run.45-2

## 5 Honeycomb

Now I can use the previous results to calculate a Honeycomb hexagonal lattice, which is simply a triangular lattice with a basis. As before, the primitive vectors are  $\mathbf{a} = (1,0)$  and  $\mathbf{b} = (1, \sqrt{3})/2$ , and the particles are situated at  $\mathbf{r}_0 = (0,0)$  and  $\mathbf{r}_1 = (1, \tan \pi/6)$ . Thus, the code to produce the primitive cell is the following.

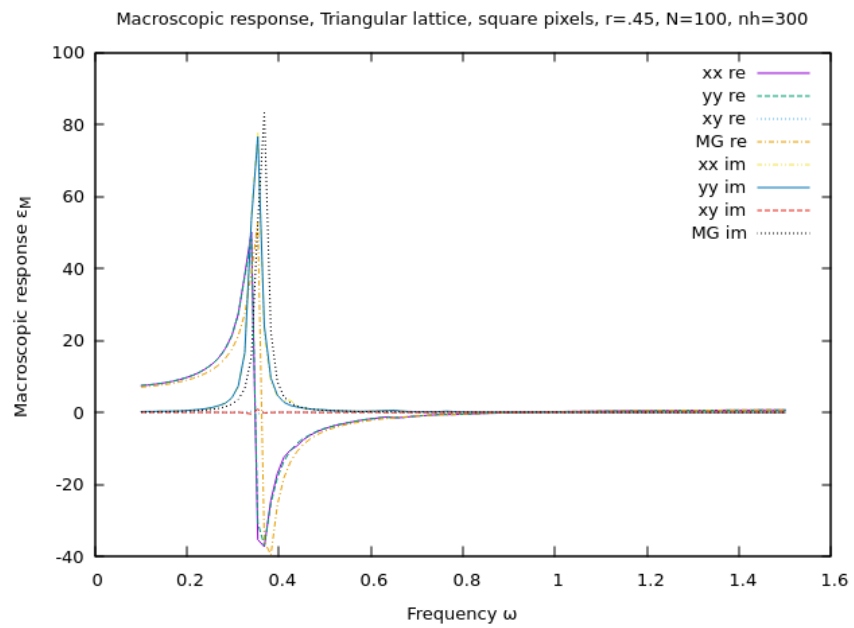


Figure 35: Macroscopic response, as in Fig. 18 but calculated with a non-orthogonal primitive unit cell as in Fig. 34.

```

my $lattice_name="Honeycomb lattice";
my $pi=atan2(1,1)*4;
my $r1=pd1(1,tan($pi/6));
my $N2=2*$N+1; # pixels of cell;
my $units=pd1[[1,0], [1/2, sqrt(3)/2]];
my $L=pd1(1,1);
my $B=zeros($N2+1, $N2+1);
# $B->ndcoords has indices a,n,m, with a=0,1 numbering primitive
# vectors and n,m numbering pixels
# $units has coordinates i,a with i Cartesian indices and a numbering
# primitive vectors. Thus, to sum over a we have to transpose units
# and add a dummy index (i) to $B->ndcoords
my $r=($B->ndcoords->dummy(1)/$N2*$units->mv(1,0))->sumover; # coordinates
$B= reduce { $a|$b } # distance to corners smaller than radius
      map {($r-$r(:, $_->[0], $_->[1]))->abs2->sumover<$radius**2}
      ([0,0], [$N2,0], [$N2, $N2], [0,$N2]);
$B |= ($r-$r1)->abs2->sumover<$radius**2; # center particle
$B=$B(0:-2, 0:-2); # remove redundant edges
# Initialize a geometry
my $g=Photonic::Geometry::FromB->new(B=>$B, L=>$L, units=>$units);

```

Figure 37: honeycomb

I plot the characteristic function.

```

# Plot the characteristic function
my $gp=PDL::Graphics::Gnuplot->new;
my $B_tiled=tile($B, 3,3);
my $r_tiled=($B_tiled->ndcoords->dummy(1)/$N2*$units->mv(1,0))->sumover;
$gp->plot({title=>"$lattice_name, r=$radius, N=$N",
          colorbox=>0, justify=>1},
          with=>'image', $r_tiled((0)), $r_tiled((1)), $B_tiled);
prompt -void, -single, "Ready?";

```

Figure 38: plot-characteristic-h

The code is then

```

<<packages>>
<<initialization>>
<<drude>>
<<honeycomb>>

```

```

<<compute>>
<<epsilon>>
<<MG>>
<<plot-characteristic-h>>
<<plot-macroscopic>>
<<usage>>

```

Figure 39: honeycomb.pl

## 5.1 Results

In Fig. 40 we show our system for  $r = 0.15a$ . In Fig. 41 we show the

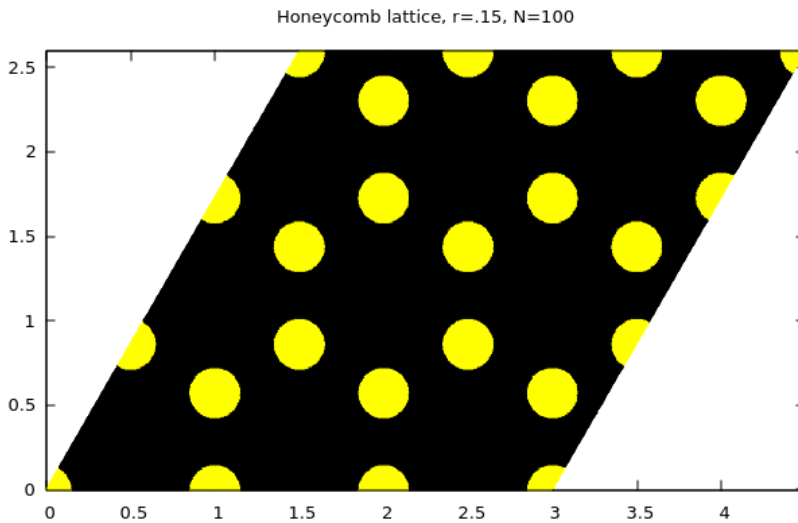


Figure 40: Honeycomb lattice of cylinders with  $r = 0.1a$ .

corresponding macroscopic response. The result is quite isotropic, except for a small region close to the resonance, where there is a small  $xy$  coupling. This could result from the pixelated nature of the lattice. Curiously, its imaginary part may take negative values. This may seem unphysical, but as long as the anti-Hermitian part of the response is positive-definite tensor, it is not. There is also a small peak close to the surface plasmon frequency

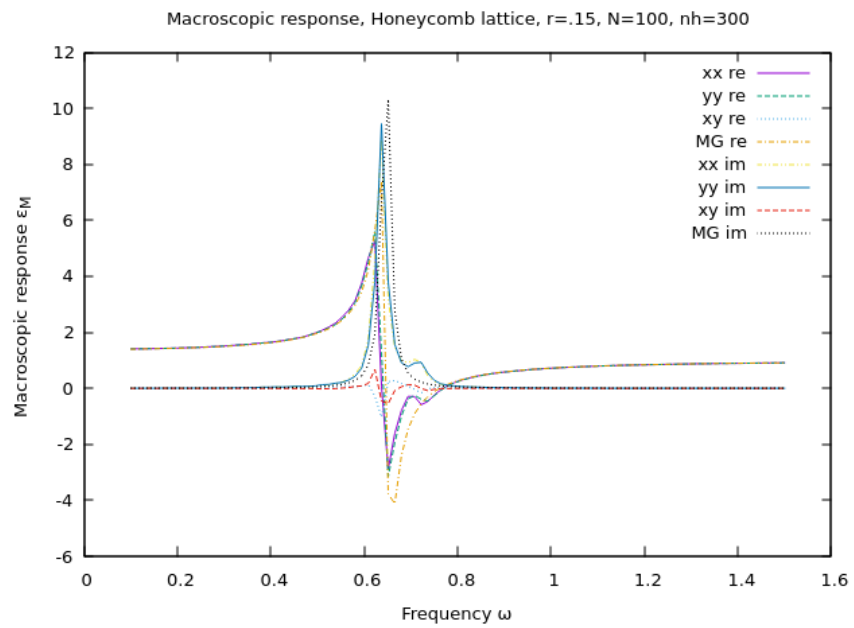


Figure 41: Macroscopic response for the honeycomb lattice with  $r = 0.15a$  shown in Fig. 40.



$\omega_p/\sqrt{2}$  of the isolated cylinder. We recall that all multipolar resonances of the isolated cylinder are degenerate in the long wavelength limit, so this peak may be related to the excitation of multipoles higher than the dipole. These were not noticeable in the simple triangular lattice, as the environment of the individual cylinder resembles more a circularly symmetric one. We might also have to consider that in this lattice there is an additional degeneracy as it has a basis with two sites that are equivalent except for a  $\pi$  rotation. .

To obtain the results above we ran the program as follows:

```
./honeycomb.pl -N 100 -epsa 1 -tau 50 -radius .15\  
-wmin .1 -wmax 1.5 -Nw 100 -nh 300
```

Figure 42: run.15-h

We obtain results also for a larger fraction corresponding to a radius  $r = 0.25a$ . In Fig. 43 we show the system, In Fig. 44 we show the corresponding

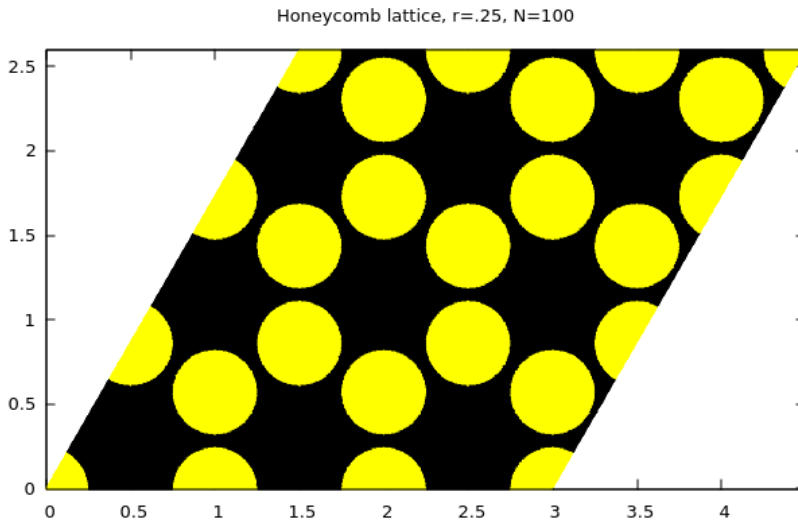


Figure 43: Triangular lattice, as in 17 but using a primitive nonorthogonal cell.

macroscopic response. In this more dense case we notice that isotropy is still maintained, except for an  $xy$  component close to the main resonance. There are additional structures, some red and some blue shifted from the isolated

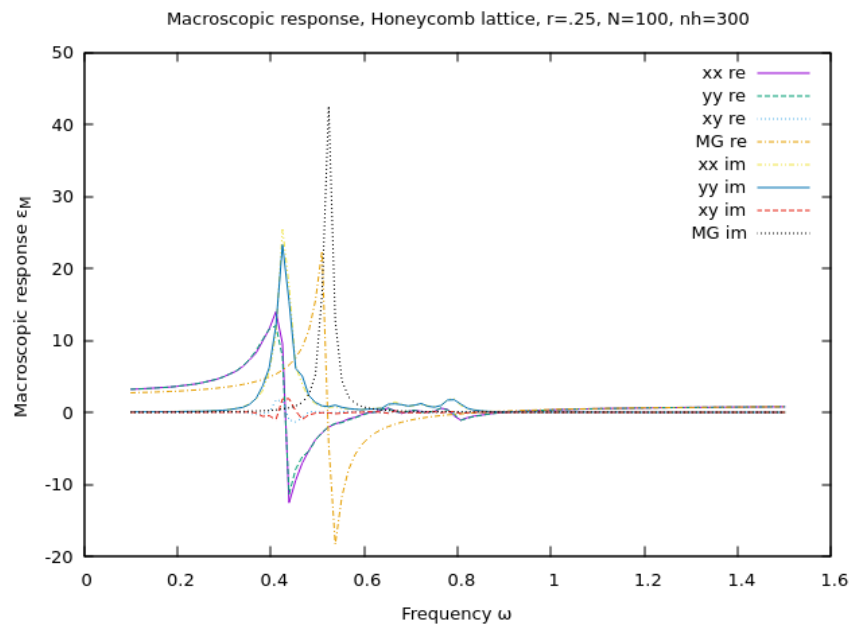


Figure 44: Macroscopic response of a honeycomb lattice, as in Fig. 41, but for a larger radius  $r = 0.25a$ , as in Fig. 43.

surface plasma frequency. They may be due to multipolar excitations. They are shifted in both directions as there are two equivalent sublattices and thus a mode repulsion. I guess symmetric modes would be redshifted and antisymmetric modes blueshifted. Not both would couple to the symmetric dipolar mode, which is the optically active one. We notice there is a large shift with respect to MG. As this lattice is far from close packed, the particles get very close together for relatively small filling fraction. Thus, the large shifts.

To obtain these we ran the program as follows:

```
./honeycomb.pl -N 100 -epsa 1 -tau 50 -radius .25\  
              -wmin .1 -wmax 1.5 -Nw 100 -nh 300
```

Figure 45: run.25-h

## 6 Run all

To run all the codes above I create a small script.

```
<<run.3>>  
<<run.45>>  
<<run.3-1>>  
<<run.45-1>>  
<<run.3-2>>  
<<run.45-2>>  
<<run.15-h>>  
<<run.25-h>>
```

Figure 46: run-all