Contents

1 Construct a small characteristic function and display it

```
use v5.36;
use PDL;
my $N_sml=21;
say my $b_small=zeroes($N_sml,$N_sml)->rvals<.35*$N_sml; # Square array of cylinders
```

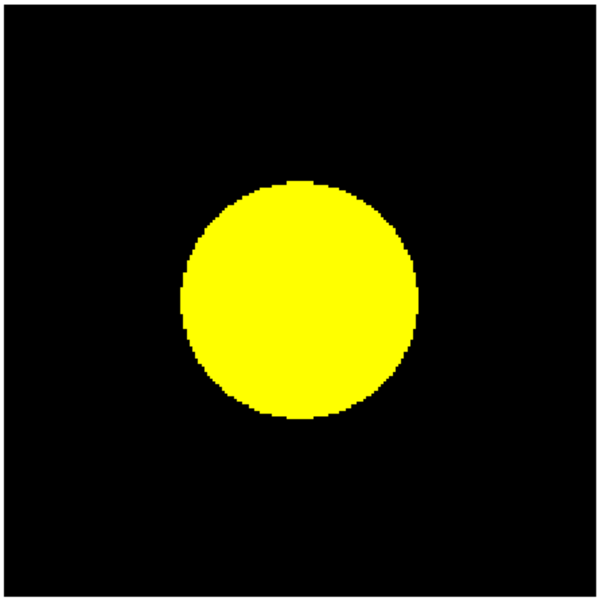
```
[
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0]
 [0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0]
 [0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0]
 [0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0]
 [0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0]
 [0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0]
 [0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0]
 [0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0]
 [0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0]
 [0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0]
 [0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0]
 [0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0]
 [0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
]
```

Ones wihin particle, zeroes in interstices.

2 Construct a larger, working characteristic function

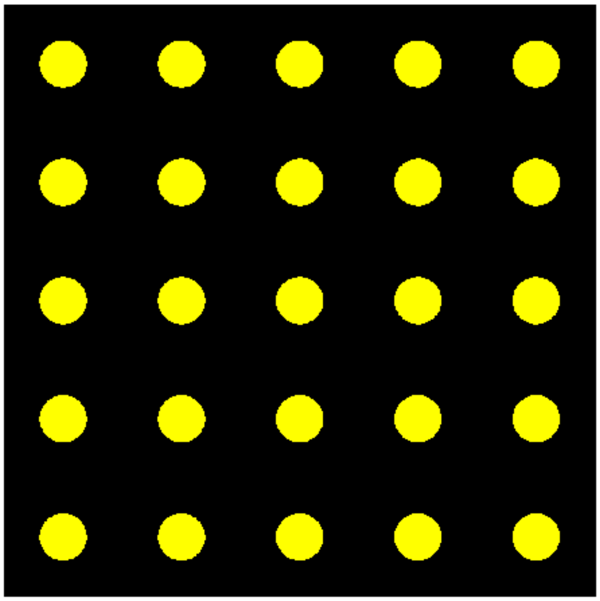
```
use PDL::Graphics::Gnuplot;
my $N=201;
my $R=0.2;
my $b=zeroes($N,$N)->rvals<$R*$N;
gplot({term=>'pngcairo', output=>'b.png', justify=>1, colorbox=>0, xtics=>0, ytics=>0}
      with=>'image', $b);
```

3 Display unit cell



4 Display lattice

```
use Photonic::Utils qw(tile);
gplot({term=>’pngcairo’, output=>’repeated_b.png’, justify=>1, colorbox=>0, xtics=>0, y
      with=>’image’, tile($b,5,5));
```



5 Use it to initialize a Geometry object

```
use Photonic::Geometry::FromB;
my $k=pdl(1,0); # wavevector along x
my $L=1;  # lattice parameter
my $g=Photonic::Geometry::FromB->new(B=>$b, Direction0=>$k, L=>pdl($L,$L));
```

6 Explore some geometry attributes.

```
say "Filling fraction: ", $g->f;
say "Reciprocal lattice info: ", $g->G->info;
use PDL::Constants qw(PI);
say "Scaled reciprocal lattice, some magnitudes: ",
    $g->G->abs2->sumover->sqrt->slice("0:3,0:3")*$L/(2*PI); # scaled to Brillouin cell
say "Normalized reciprocal lattice, sample of magnitudes:",
    $g->GNorm->abs2->sumover->sqrt->slice("0:3,0:3");
```

```
Filling fraction: 0.125566198856464
Reciprocal lattice info: PDL: Double D [2,201,201]
Scaled reciprocal lattice, some magnitudes:
```

```
[
 [               0                1                2                3]
 [               1        1.4142136         2.236068        3.1622777]
 [               2         2.236068        2.8284271        3.6055513]
 [               3        3.1622777        3.6055513        4.2426407]
]

Normalized reciprocal lattice, sample of magnitudes:
[
 [1 1 1 1]
 [1 1 1 1]
 [1 1 1 1]
 [1 1 1 1]
]
```

7 Use Geometry to initialice a calculator of Haydock coefficients

```
use Photonic::LE::NR2::Haydock;
my $Nh=$N;  # desired number of Haydock coefficients
my $h=Photonic::LE::NR2::Haydock->new(geometry=>$g, nh=>$Nh);
```

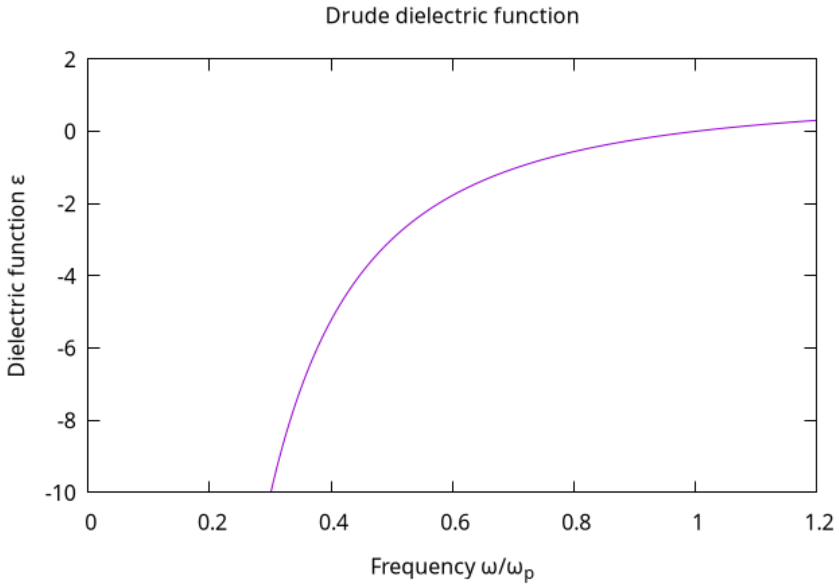
8 Use Haydock object to initialize a projected dielectric ϵ function calculator

```
use Photonic::LE::NR2::EpsL;
my $eps=Photonic::LE::NR2::EpsL->new(
    haydock=>$h, nh=>$Nh, epsA=>1+0*i(), epsB=>2+0*i());
say sprintf "Square array of glass rods, epsilon=%.2f", $eps->epsL->re;

Square array of glass rods, epsilon=1.09
```

9 Metallic (Drude) dielectric response

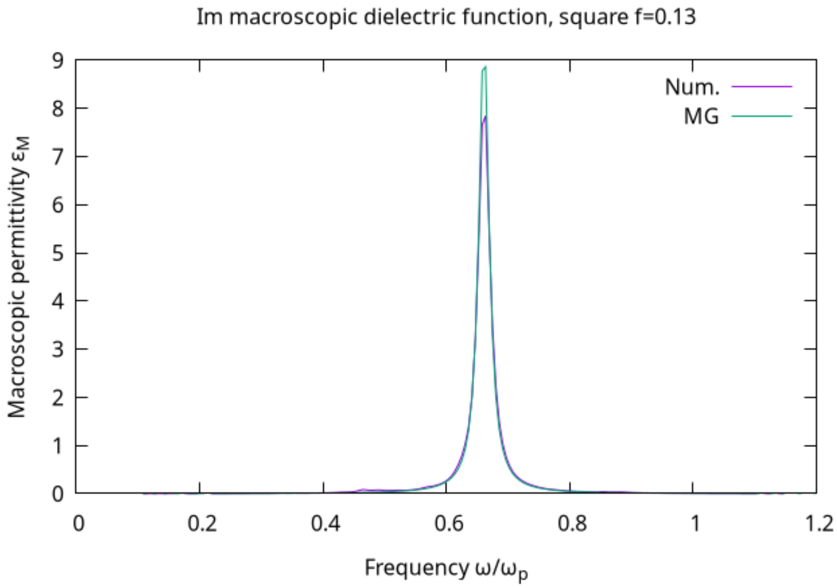
```
my $w=zeroes(200)->xlinvals(.1,1.2);
my $tau=50;
my $drude=1-1/($w*($w+i()/$tau));
gplot({term=>'pngcairo', output=>'drude.png',
       title=>"Drude dielectric function",
       xlabel=>"Frequency {/Symbol w/w}_p",
       ylabel=>"Dielectric function {/Symbol e}",
       yrange=>[-10,2]}, with=>'lines', $w, $drude->re);
```



10 Spectrum for square array of metallic cylindrical wires

```
my $eps_wire=pdl(
    map {Photonic::LE::NR2::EpsL->new(
      haydock=>$h, nh=>$Nh, epsA=>1+0*i(), epsB=>$_
      )->epsL
    }  $drude->dog
    );
my $eps_MG=($drude+1+$g->f*($drude-1))/($drude+1-$g->f*($drude-1));
gplot({term=>'pngcairo', output=>'square.png',
       title=>sprintf("%s f=%.2f",
  "Im macroscopic dielectric function, square",@{[$g->f]}),
       xlabel=>"Frequency {/Symbol w/w}_p",
       ylabel=>"Macroscopic permittivity {/Symbol e}_M"},
       {with=>’lines’, legend=>"Num."}, $w, $eps_wire->im,
       {with=>’lines’, legend=>"MG"}, $w, $eps_approx->im);
```

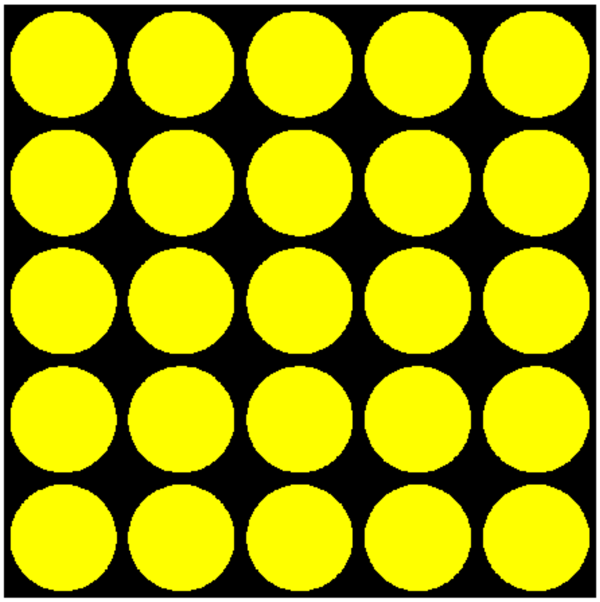
```
Global symbol "$eps_approx" requires explicit package name (did you forget to declare "
BEGIN not safe after errors--compilation aborted at reply input line 18.
```

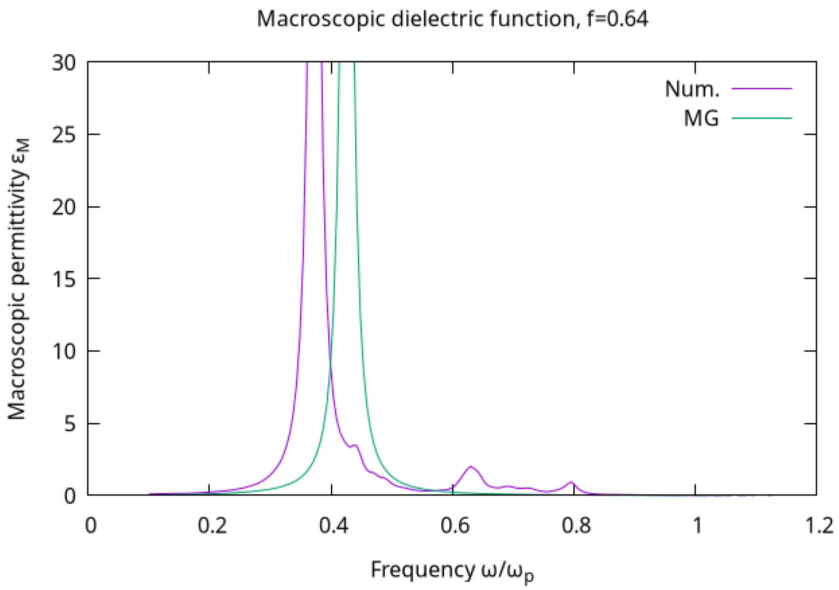


11 Increase filling fraction

```
my $R_fat=.45;
my $b_fat=zeroes($N,$N)->rvals<$R_fat*$N;
my $g_fat=Photonic::Geometry::FromB->new(
    B=>$b_fat, Direction0=>$k, L=>pdl($L,$L));
my $h_fat=Photonic::LE::NR2::Haydock->new(
    geometry=>$g_fat, nh=>$Nh);
my $eps_fat=pdl(
    map {Photonic::LE::NR2::EpsL->new(
      haydock=>$h_fat, nh=>$Nh, epsA=>1+0*i(), epsB=>$_)->epsL}
    $drude->dog
    );
my $eps_MG_fat=($drude+1+$g_fat->f*($drude-1))/($drude+1-$g_fat->f*($drude-1));
my $tiled_fat=tile($b_fat,5,5);
```

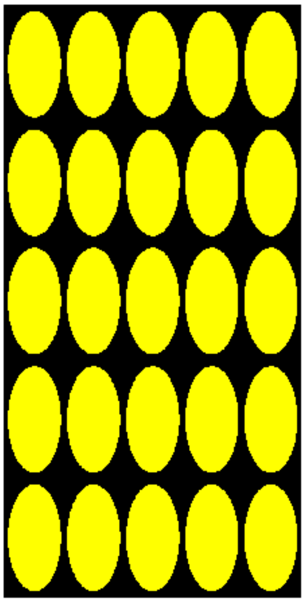
```
gplot({term=>'pngcairo', output=>'repeated_fat.png', justify=>1, colorbox=>0, xtics=>0
      with=>'image', $tiled_fat);
gplot({term=>'pngcairo', output=>'fat.png',
       title=>sprintf("%s, f=%.2f",
        "Macroscopic dielectric function",@{[$g_fat->f]}),
       xlabel=>"Frequency {/Symbol w/w}_p",
       ylabel=>"Macroscopic permittivity {/Symbol e}_M",
       yrange=>[0,30]},
       {with=>'lines', legend=>"Num."}, $w, $eps_fat->im,
       {with=>'lines', legend=>"MG"}, $w, $eps_MG_fat->im);
```

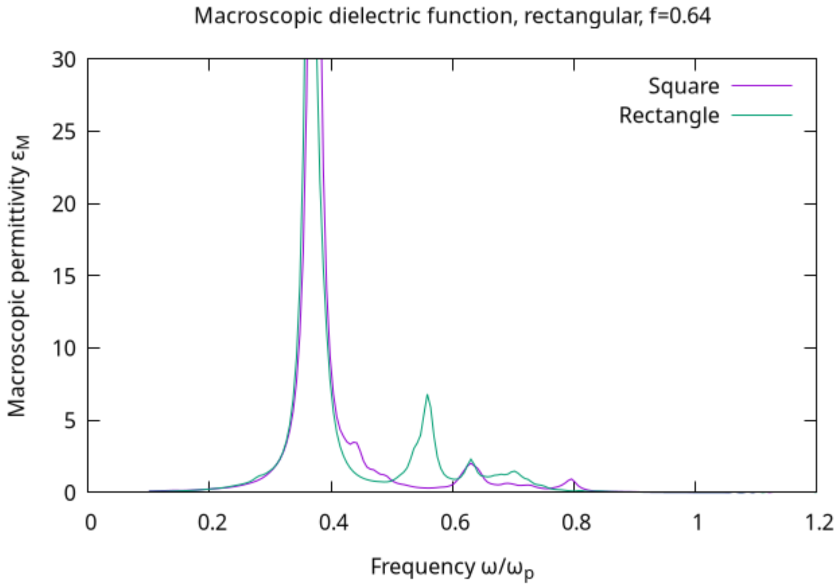




12 Change the aspect ratio

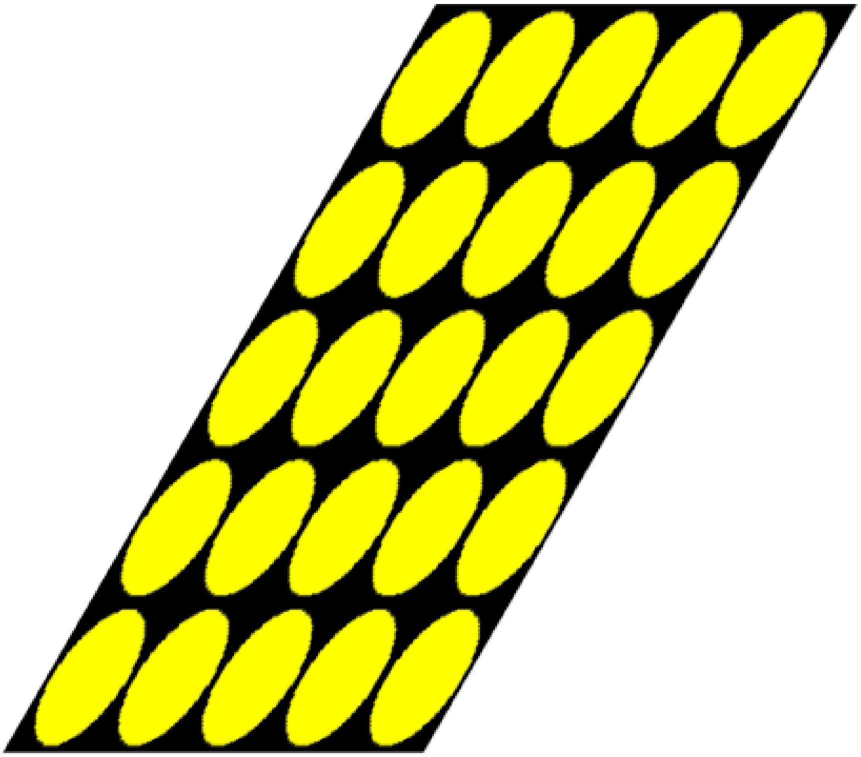
```
my $g_long=Photonic::Geometry::FromB->new(
    B=>$b_fat, Direction0=>$k, L=>pdl($L,2*$L));
my $h_long=Photonic::LE::NR2::Haydock->new(
    geometry=>$g_long, nh=>$Nh);
my $eps_long=pdl(
    map {Photonic::LE::NR2::EpsL->new(
      haydock=>$h_long, nh=>$Nh, epsA=>1+0*i(), epsB=>$_)->epsL}
    $drude->dog
    );
gplot({term=>’pngcairo’, output=>’repeated_rectangular.png’, justify=>1, colorbox=>0, x
      with=>’image’, $tiled_fat->xvals, 2*$tiled_fat->yvals, $tiled_fat);
gplot({term=>’pngcairo’, output=>’rectangular.png’,
       title=>sprintf("%s, f=%.2f",
        "Macroscopic dielectric function, rectangular",
        @{[$g_long->f]}),
       xlabel=>"Frequency {/Symbol w/w}_p",
       ylabel=>"Macroscopic permittivity {/Symbol e}_M",
       yrange=>[0,30]},
       {with=>'lines', legend=>"Square"}, $w, $eps_fat->im,
       {with=>'lines', legend=>"Rectangle"}, $w, $eps_long->im);
```

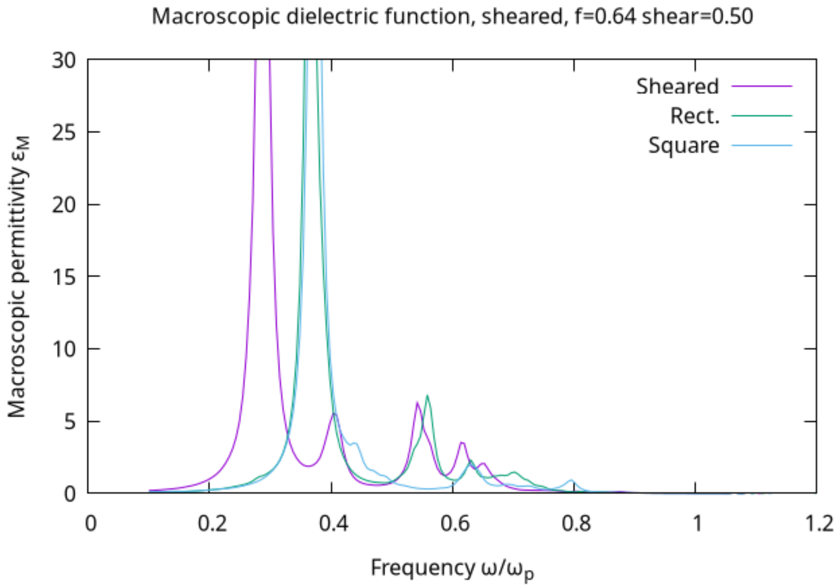



Macroscopic dielectric function, rectangular, f=0.64
Macroscopic permittivity ϵ_M
Frequency ω/ω_p
Square
Rectangle
0
5
10
15
20
25
30
0
0.2
0.4
0.6
0.8
1
1.2

13 Deform (shear) the structure

```
my $shear=0.5;
my $g_sheared=Photonic::Geometry::FromB->new(
    B=>$b_fat, Direction0=>$k, L=>pdl($L,2*$L),
    primitive=>pdl([[1,0],[$shear,1]]));
my $h_sheared=Photonic::LE::NR2::Haydock->new(
    geometry=>$g_sheared, nh=>$Nh);
my $eps_sheared=pdl(
    map {Photonic::LE::NR2::EpsL->new(
      haydock=>$h_sheared, nh=>$Nh, epsA=>1+0*i(), epsB=>$_)->epsL}
    $drude->dog
    );
gplot({term=>'pngcairo', output=>'sheared.png',
       title=>sprintf("%s, f=%.2f shear=%.2f",
        "Macroscopic dielectric function, sheared",
        @{[$g_long->f]}, $shear),
       xlabel=>"Frequency {/Symbol w/w}_p",
       ylabel=>"Macroscopic permittivity {/Symbol e}_M",
       yrange=>[0,30]},
       {with=>'lines', legend=>"Sheared"}, $w, $eps_sheared->im,
       {with=>'lines', legend=>"Rect."}, $w, $eps_long->im,
       {with=>'lines', legend=>"Square"}, $w, $eps_fat->im);
```




Macroscopic dielectric function, sheared, f=0.64 shear=0.50
Macroscopic permittivity ε_M
Frequency ω/ω_p
0
5
10
15
20
25
30
0
0.2
0.4
0.6
0.8
1
1.2
Sheared
Rect.
Square