Intermediate Flux Densities

The Fourier analysis of local flux density is given by

$$h_z\left(\vec{r}\right) = \sum_{\vec{Q}} h_{\vec{Q}} e^{i\vec{Q}\cdot\vec{r}}$$

Satisfying the London equations, one gets that the flux density becomes

$$h_z\left(\vec{r}\right) = B\sum_{\vec{Q}} \frac{e^{i\vec{Q}\cdot\vec{r}}}{1+\lambda^2 Q^2}$$

where $\vec{Q}$ is the reciprocal lattice vector

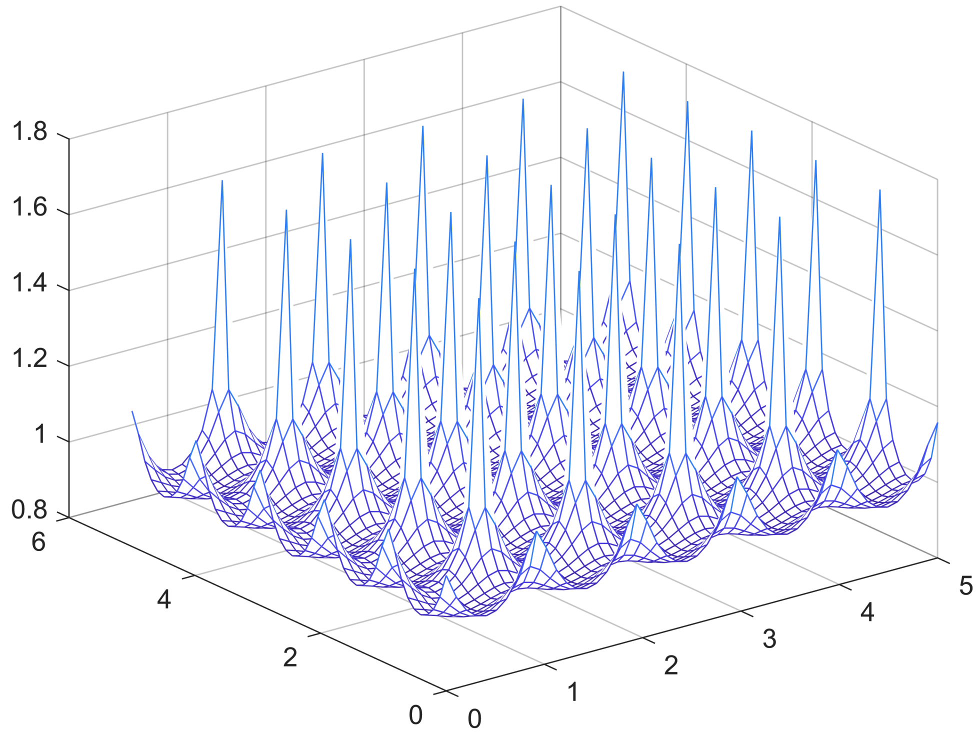
$$\vec{Q} = h\vec{b}_1 + k\vec{b}_2 \qquad h, k \in Z$$

In a square array the reciprocal lattice vector is

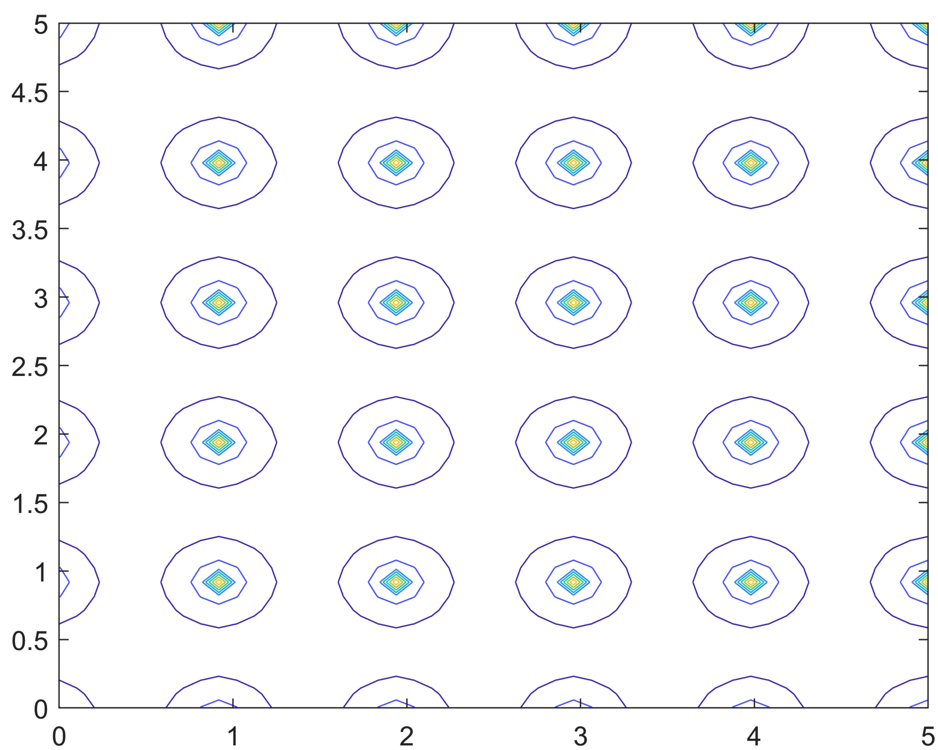
$$\vec{b}_i = \frac{2\pi}{a}\vec{e_i}$$

The flux density for some given $\vec{r}$ is then:

```
datasquare=zeros(50);
for m=1:50
    for n=1:50
        datasquare(m,n)=fluxdensitySquare(1,1,1,m*0.1,n*0.1);
    end
end
x = linspace(0,5,size(datasquare,2));
y = linspace(0,5,size(datasquare,1));
mesh(x,y,real(datasquare))
```



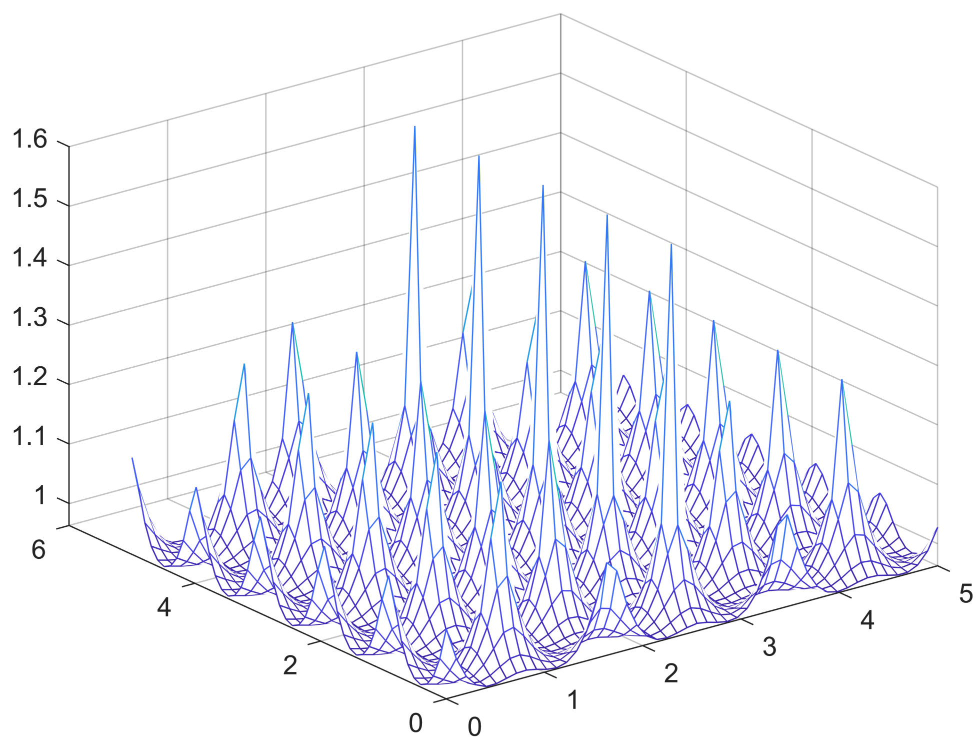

```
contour(x,y,real(datasquare))
```



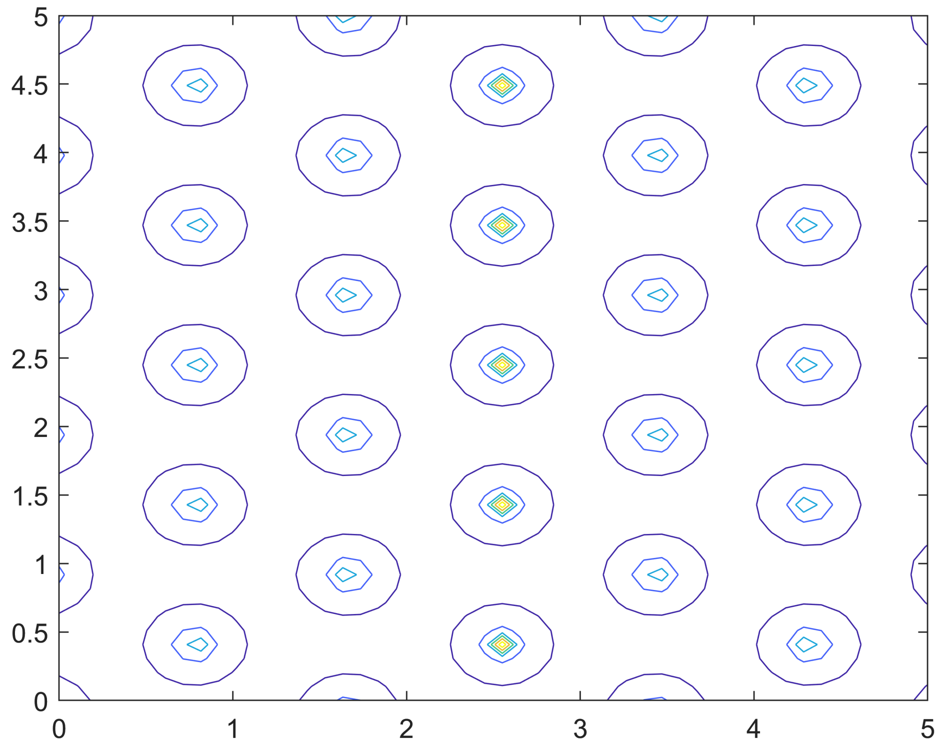
In a traingular one the lattice vector becomes

$$\overrightarrow{\mathrm{b}_1} = \frac{2\pi}{a}\left(\overrightarrow{e_1} - \frac{1}{\sqrt{3}}\overrightarrow{e_2}\right), \ \overrightarrow{\mathrm{b}_2} = \frac{2\pi}{a}\frac{2}{\sqrt{3}}\overrightarrow{e_2}$$

```
datatri=zeros(50);
for m=1:50
    for n=1:50
        datatri(m,n)=fluxdensityTri(1,1,1,m*0.1,n*0.1);
    end
end
x = linspace(0,5,size(datatri,2));
y = linspace(0,5,size(datatri,1));
mesh(x,y,real(datatri))
```




```
contour(x,y,real(datatri))
```



The free energy increase per unit volume is then

$$F - F_{s0} = \frac{B^2}{8\pi}\sum_{\vec{Q}}\frac{1}{1+\lambda^2 Q^2}$$

For the square array one gets

```
dFsq=0;
a=1;
b=1;
lpd=1;
for h=-50:50
    for k=-50:50
        index=[h k];
        Qbase=2*pi/a*[1 1];
        Q=Qbase.*index;
        dFsq=dFsq+1/(1+(lpd*norm(Q))^2);
    end
end
dFsq=b^2/(8*pi)*dFsq
```

```
dFsq = 0.0678
```

As for the triangular array one gets

```
dFtr=0;
a=1;
b=1;
b1=2*pi/a*[1 -1/sqrt(3)];
b2=2*pi/a*2/sqrt(3)*[0 1];
lpd=1;
for h=-50:50
    for k=-50:50
        Q=h*b1+k*b2;
        dFtr=dFtr+1/(1+(lpd*norm(Q))^2);
    end
end
dFtr=b^2/(8*pi)*dFtr
```

```
dFtr = 0.0638
```

```
function fluxDens=fluxdensitySquare(lpd,b,a,x,y)
    fluxDens=0;
    r= [x y];
    for h=-50:50
        for k=-50:50
            index=[h k];
            Qbase=2*pi/a*[1 1];
            Q=Qbase.*index;
            fluxDens=fluxDens+b*exp(1i*dot(Q,r))/(1+(lpd*norm(Q))^2);
        end
    end
end

function fluxDens=fluxdensityTri(lpd,b,a,x,y)
    fluxDens=0;
    r= [x y];
    b1=2*pi/a*[1 -1/sqrt(3)];
    b2=2*pi/a*2/sqrt(3)*[0 1];
    for h=-50:50
        for k=-50:50
            Q=h*b1+k*b2;
            fluxDens=fluxDens+b*exp(1i*dot(Q,r))/(1+(lpd*norm(Q))^2);
        end
    end
end
```