Phonon dispersion relation and density of states of a simple cubic lattice

Student project for the course Molecular and Solid State Physics

by Eva Meisterhofer

Contents

1 The linear spring model

A simple model for describing lattice vibrations in a crystal is to assume that the atoms are masses connected by linear springs. With this approximation we want to calculate the phonon dispersion relation and density of states for a simple cubic lattice.

1.1 Nearest neighbours

When considering only the six nearest neighbours, the equation of motion in the x-direction (Newton's law) is:

$$m\frac{d^2u^x_{lmn}}{dt^2} = C[(u^x_{l+1mn} - u^x_{lmn}) + (u^x_{l-1mn} - u^x_{lmn})]$$

The solutions of this differential equation are eigenfunctions of the translation operator:

$$u^x_{lmn} = A^x_{\vec{k}} \exp\left[i(l\vec{k}\vec{a}_1 + m\vec{k}\vec{a}_2 + n\vec{k}\vec{a}_3)\right] = A^x_{\vec{k}} \exp\left[i(lk_xa + mk_ya + nk_za)\right]$$

The expressions for the y- and z-directions are similar. Since the three equations of motion are independent, the three-dimensional problem decouples into three one-dimensional problems.
By substituting the eigenfunction solutions into Newton's law, the differential equations become algebraic equations which can be used to calculate the dispersion relation.

$$m\omega^2\vec{u}_{lmn} = \mathbf{M}\vec{u}_{lmn}$$

$$\mathbf{M} = \begin{pmatrix} 4C\sin^2\left(\frac{ak_x}{2}\right) & 0 & 0 \\ 0 & 4C\sin^2\left(\frac{ak_y}{2}\right) & 0 \\ 0 & 0 & 4C\sin^2\left(\frac{ak_z}{2}\right) \end{pmatrix}$$

The phonon dispersion relation can be obtained by calculating the eigenvalues λ of the Matrix $\mathbf{M}$:

$$\omega = \sqrt{\frac{\lambda}{m}}$$

Since the eigenvalues of a diagonal matrix are the diagonal elements, the dispersion relation for a simple cubic lattice considering only the nearest neighbours is:

$$\omega_1 = \sqrt{\frac{4C}{m}}\left|\sin\left(\frac{ak_x}{2}\right)\right|, \omega_2 = \sqrt{\frac{4C}{m}}\left|\sin\left(\frac{ak_y}{2}\right)\right|, \omega_3 = \sqrt{\frac{4C}{m}}\left|\sin\left(\frac{ak_z}{2}\right)\right|$$

1.2 Nearest and next nearest neighbours

When considering the 6 nearest and the 12 next nearest neighbours, the equation of motion in the x-direction is:

$$m\frac{d^2u^x_{lmn}}{dt^2} = C_1[(u^x_{l+1mn} - u^x_{lmn}) + (u^x_{l-1mn} - u^x_{lmn})]$$
$$+ \frac{C_2}{2}[(u^x_{l+1m+1n} - u^x_{lmn}) + (u^x_{l+1m-1n} - u^x_{lmn}) + (u^x_{l-1m+1n} - u^x_{lmn}) + (u^x_{l-1m-1n} - u^x_{lmn})$$
$$+ (u^x_{l+1mn+1} - u^x_{lmn}) + (u^x_{l+1mn-1} - u^x_{lmn}) + (u^x_{l-1mn+1} - u^x_{lmn}) + (u^x_{l-1mn-1} - u^x_{lmn})$$
$$+ (u^y_{l+1m+1n} - u^y_{lmn}) - (u^y_{l+1m-1n} - u^y_{lmn}) - (u^y_{l-1m+1n} - u^y_{lmn}) + (u^y_{l-1m-1n} - u^y_{lmn})$$
$$+ (u^z_{l+1mn+1} - u^z_{lmn}) - (u^z_{l+1mn-1} - u^z_{lmn}) - (u^z_{l-1mn+1} - u^z_{lmn}) + (u^z_{l-1mn-1} - u^z_{lmn})]$$

C_1 is the spring constant for the nearest and C_2 the spring constant for the next nearest neighbours. The expressions for the y- and z-directions are similar. By substituting the eigenfunction solutions into the equations of motion, the three coupled differential equations become algebraic equations:

$$m\omega^2\vec{u}_{lmn} = \mathbf{M}\vec{u}_{lmn}$$

The phonon dispersion relation can be obtained by calculating the eigenvalues λ of the Matrix $\mathbf{M}$:

$$\omega = \sqrt{\frac{\lambda}{m}}$$

$$
\mathbf{M} = \begin{pmatrix}
\begin{matrix} 2C_1[1-\cos(ak_x)] \\ +2C_2[2-cos(ak_x)cos(ak_y)-cos(ak_x)cos(ak_z)] \end{matrix} & 2C_2\sin(ak_x)\sin(ak_y) & 2C_2\sin(ak_x)\sin(ak_z) \\
2C_2\sin(ak_x)\sin(ak_y) & \begin{matrix} 2C_1[1-\cos(ak_y)] \\ +2C_2[2-cos(ak_x)cos(ak_y)-cos(ak_y)cos(ak_z)] \end{matrix} & 2C_2\sin(ak_y)\sin(ak_z) \\
2C_2\sin(ak_x)\sin(ak_z) & 2C_2\sin(ak_y)\sin(ak_z) & \begin{matrix} 2C_1[1-\cos(ak_z)] \\ +2C_2[2-cos(ak_x)cos(ak_z)-cos(ak_y)cos(ak_z)] \end{matrix}
\end{pmatrix}
$$

2 Numerical calculation in Matlab

2.1 Matlab Files

The following Matlab program calculates and plots the phonon dispersion relation and density of states for simple cubic considering the nearest and next nearest neighbours. The calculation is performed for a set of different quotients of the two spring constants $\frac{C_1}{C_2}$.

```
% Phonon dispersion relation and density of states for a simple cubic
% lattice using the linear spring model

%----parameters------------------------------------------------------------
% dimensions
d = 3;

% lattice constant in real space in meters
a = 1;

% quotient of the spring constants for nearest & next nearest neighbours
% c_quot = c1/c2
c_quot = [inf,6,3,2,1]; %(guess)

% number of points in k-space
n = 1000; % for plotting the dispersion relationship
n_dos = 10000000; % for calculating the DoS

% number of histogram bins for the DoS
w_bin = 150;

% brilloin zone symmetry points
g = [0;0;0];
x = [0;pi/a;0];
m = [pi/a;pi/a;0];
r = [pi/a;pi/a;pi/a];

% dispersion relation plot order (symmetry points)
po = [g,x,m,g,r];
po_label = {'G','X','M','G','R'};

%----phonon dispersion relation---------------------------------------------
% creating n linearly spaced k vectors between each pair of symmetry
% points:
n_po = size(po,2)-1;
k = nan(d,(n-1)*n_po+1); % kx;ky;kz
kk = zeros(1,size(k,2)); % length of path in k-space for plot axis
kk_s = zeros(1,n_po+1); % values of kk at the symmetry points

k(:,1) = po(:,1);
kk(1) = kk_s(1);
for ind1 = 1:n_po

    ind1_start = (ind1-1)*(n-1)+2;
    ind1_end = ind1*(n-1)+1;

    kk_s(ind1+1) = kk_s(ind1)+norm(po(:,ind1+1)-po(:,ind1));
    temp_kk = linspace(kk_s(ind1),kk_s(ind1+1),n);
    kk(1,ind1_start:ind1_end) = temp_kk(2:end);

    for ind2 = 1:d
        temp_k = linspace(po(ind2,ind1),po(ind2,ind1+1),n);
        k(ind2,ind1_start:ind1_end) = temp_k(2:end);
    end
end


% calculating the frequencies omega/(sqrt(C/m)) for each k from the dispersion relation:
% nearest neighbours
w1 = fun_disp1(k,a);
```

```
% nearest & next nearest neighbours
numel_c_quot = numel(c_quot);
w2 = nan(size(k,1),size(k,2),numel_c_quot);
for ind1 = 1:numel_c_quot
  w2(:,:,ind1) = fun_disp2(k,a,c_quot(ind1));
end



% plotting the dispersion relation:
% nearest neighbours
ymax1 = round(11*max(w1(:)))/10;
figure(1)
subplot(2,1+numel_c_quot,1)
plot(kk,w1,'b-')
for ind1 = 1:(n_po+1)
    line([kk_s(ind1) kk_s(ind1)],[0 ymax1], 'Color','r')
end
ylim([0 ymax1])
xlim([kk(1) kk(end)])
title('Phonon dispersion relation of sc, nearest neighbours')
ylabel('\omega_{norm} = \omega / sqrt(C/m)')
set(gca,'XTick',kk_s)
set(gca,'XTickLabel',po_label)

% nearest and next neighbours
for ind1 = 1:numel_c_quot
    ymax2 = round(11*max(max(w2)))/10;
    subplot(2,1+numel_c_quot,1+ind1)
    plot(kk,w2(:,:,ind1),'b-')
    for ind2 = 1:(n_po+1)
        line([kk_s(ind2) kk_s(ind2)],[0 ymax2(ind1)], 'Color','r')
    end
    ylim([0 ymax2(ind1)])
    xlim([kk(1) kk(end)])
    title({'Phonon dispersion relation of sc,'; 'nearest and next nearest neighbours';['C1/C2 =
        ', num2str(c_quot(ind1))]})
    ylabel('\omega_{norm} = \omega / sqrt(C_1/m)]')
    set(gca,'XTick',kk_s)
    set(gca,'XTickLabel',po_label)
end


%-----density of states---------------------------------------------------
% choosing random k vectors in the first brilloin zone
k_rand = 2*pi/a*(rand(d,n_dos)-0.5);

% calculating the frequencies from the dispersion relation:
% nearest neighbours
w1_rand = fun_disp1(k_rand,a);

% nearest & next nearest neighbours
w2_rand = nan(size(k_rand,1),size(k_rand,2),numel_c_quot);
for ind1 = 1:numel_c_quot
   w2_rand(:,:,ind1) = fun_disp2(k_rand,a,c_quot(ind1));
end



% histogram
% nearest neighbours
[dw1,wn1] = hist(w1_rand(:),w_bin);
dos1 = dw1/n_dos * w_bin/(max(wn1)-min(wn1)) * 1/(a^3); %normalisation

subplot(2,1+numel_c_quot,2+numel_c_quot)
plot(wn1,dos1)
xlim([0 wn1(end)])
title('Phonon density of states of sc, nearest neighbours')
xlabel('\omega_{norm} = \omega / sqrt(C/m)')
ylabel('D(\omega_{norm})')

%DoS text file
fid = fopen('sc_dos_onlynearest.txt','w');
fprintf(fid,'%s\n','Phonon density of states of sc, only nearest neighbours');
```

```
fprintf(fid,'%s\t%s\n', 'omega/sqrt(C/m)','D(omega/sqrt(C/m))');
fprintf(fid,'%f\t%f\n', [wn1; dos1]);
fclose(fid);


% nearest & next nearest neighbours
dw2 = nan(numel_c_quot,w_bin);
wn2 = nan(numel_c_quot,w_bin);
dos2 = nan(numel_c_quot,w_bin);
for ind1 = 1:numel_c_quot
w2_rand_ind1 = w2_rand(:,:,ind1);
[dw2(ind1,:),wn2(ind1,:)] = hist(w2_rand_ind1(:),w_bin);
dos2(ind1,:) = dw2(ind1,:)/n_dos * w_bin/(max(wn2(ind1,:))-min(wn2(ind1,:))) * 1/(a^3); % normalisation

subplot(2,1+numel_c_quot,2+numel_c_quot+ind1)
plot(wn2(ind1,:),dos2(ind1,:))
xlim([0 wn2(ind1,end)])
title({'Phonon density of states of sc,'; 'nearest and next nearest neighbours';['C2/C1 = ', num2str(c_quot(ind1))]})
xlabel('\omega_{norm} = \omega / sqrt(C_1/m)')
ylabel('D(\omega_{norm})')

%DoS text file
fid = fopen(['sc_dos_',num2str(c_quot(ind1)),'.txt'],'w');
fprintf(fid,'%s\n','Phonon density of states of sc, nearest and next nearest neighbours');
fprintf(fid,'%s%s\n','C2/C1 = ', num2str(c_quot(ind1)));
fprintf(fid, '%s\t%s\n', 'omega/sqrt(C_1/m)','D(omega/sqrt(C_1/m))');
fprintf(fid, '%f\t%f\n', [wn2(ind1,:); dos2(ind1,:)]);
fclose(fid);
end
```

```
function [w] = fun_disp1(k,a)
% Phonon dispersion relation for a simple cubic lattice
% in the linear spring model considering only the nearest neighbours.
% Input: matrix k with wavenumber column vectors [kx;ky;kz],
% lattice constant a
% Output: frequencies w = [w1;w2;w3]

w = 2*abs(sin(k*a/2));

end
```

```
function [w] = fun_disp2(k,a,c_quot)
% Phonon dispersion relation for a simple cubic lattice in the linear
% spring model considering the nearest and next nearest neighbours.
% Input: matrix k with wavenumber column vectors [kx;ky;kz],
% lattice constant a, spring constant quotient c_quot
% Output: frequencies w = [w1;w2;w3]

w = nan(size(k));
M = nan(3);

for ind1 = 1:size(k,2)
    % Berechnung der Matrix M
    M(1,1) = 2*(1-cos(a*k(1,ind1)))+2/c_quot*(2-cos(a*k(1,ind1))*cos(a*k(2,ind1))-cos(a*k(1,ind1))*cos(a*k(3,ind1)));
    M(1,2) = 2/c_quot*sin(a*k(1,ind1))*sin(a*k(2,ind1));
    M(1,3) = 2/c_quot*sin(a*k(1,ind1))*sin(a*k(3,ind1));
    M(2,1) = M(1,2);
    M(2,2) = 2*(1-cos(a*k(2,ind1)))+2/c_quot*(2-cos(a*k(1,ind1))*cos(a*k(2,ind1))-cos(a*k(2,ind1))*cos(a*k(3,ind1)));
    M(2,3) = 2/c_quot*sin(a*k(2,ind1))*sin(a*k(3,ind1));
    M(3,1) = M(1,3);
    M(3,2) = M(2,3);
    M(3,3) = 2*(1-cos(a*k(3,ind1)))+2/c_quot*(2-cos(a*k(1,ind1))*cos(a*k(3,ind1))-cos(a*k(2,ind1))*cos(a*k(3,ind1)));

    %Berechnung von omega/sqrt(C1/m) aus den Eigenwerten von M:
    w(:,ind1) = sqrt(eig(M));
end

end
```

2.2 Figures

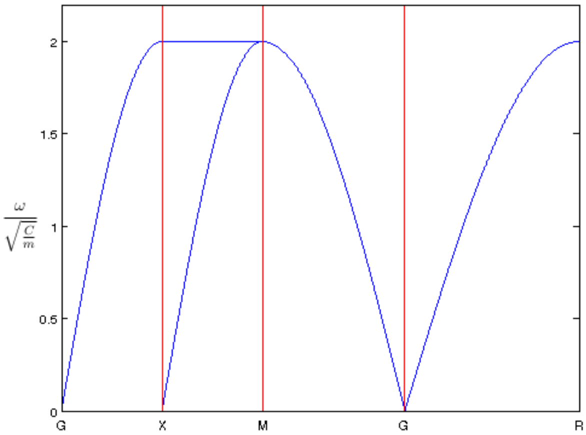

Figure 1: Phonon dispersion relation for simple cubic, only nearest neighbours

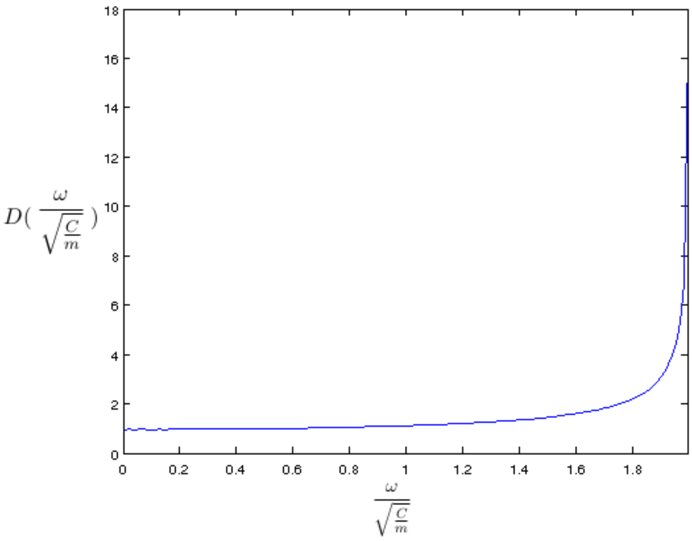

Figure 2: Phonon density of states for simple cubic, only nearest neighbours

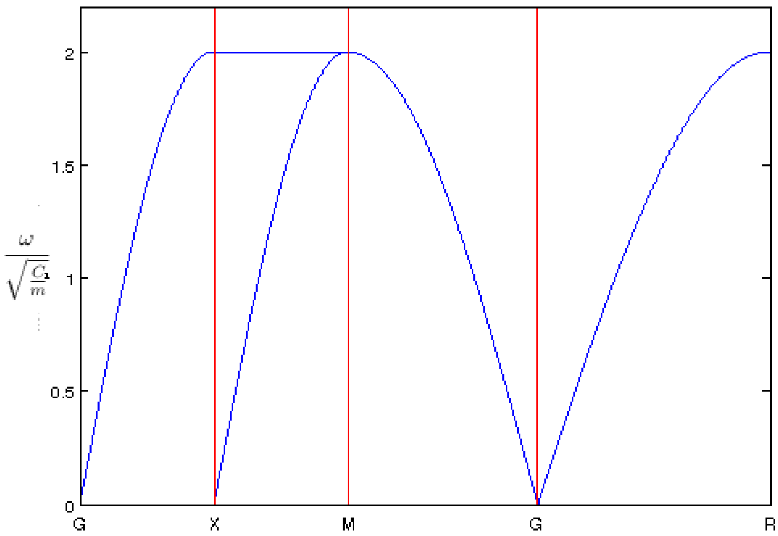

Figure 3: Phonon dispersion relation for simple cubic, nearest and next nearest neighbours, $\frac{C_1}{C_2} = \infty$

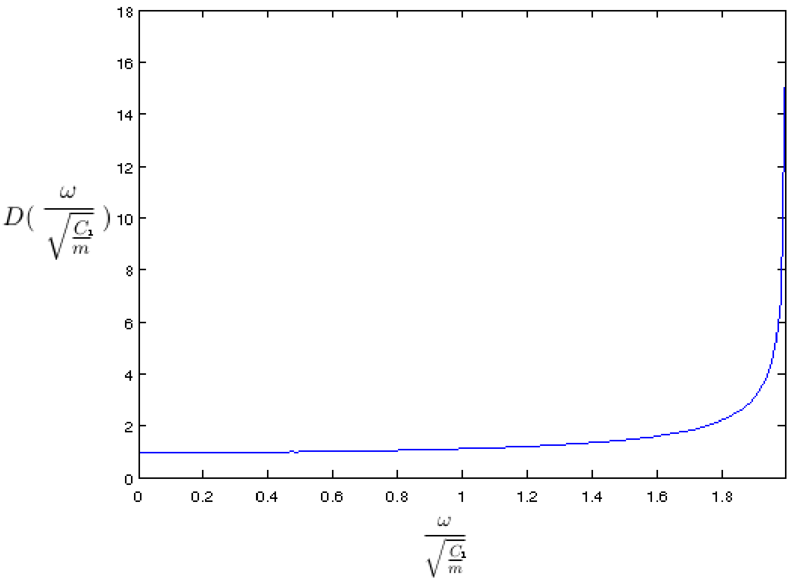

Figure 4: Phonon density of states for simple cubic, nearest and next nearest neighbours, $\frac{C_1}{C_2} = \infty$

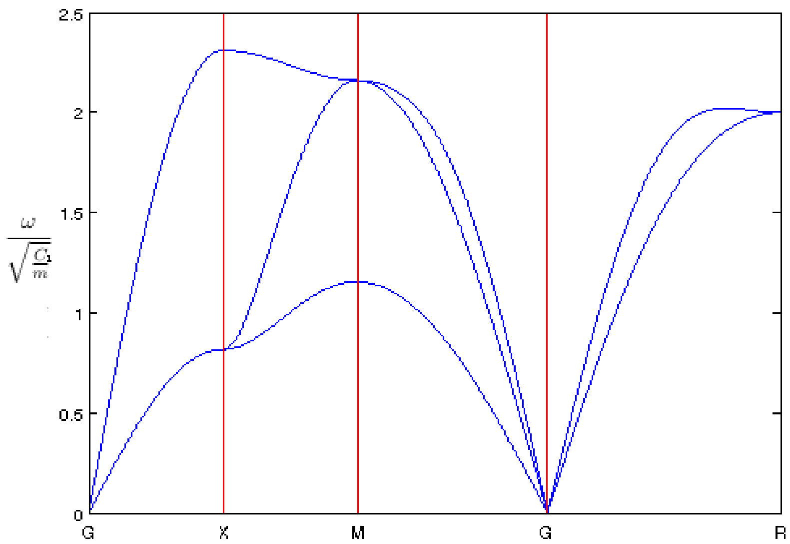

Figure 5: Phonon dispersion relation for simple cubic, nearest and next nearest neighbours, $\frac{C_1}{C_2} = 6$

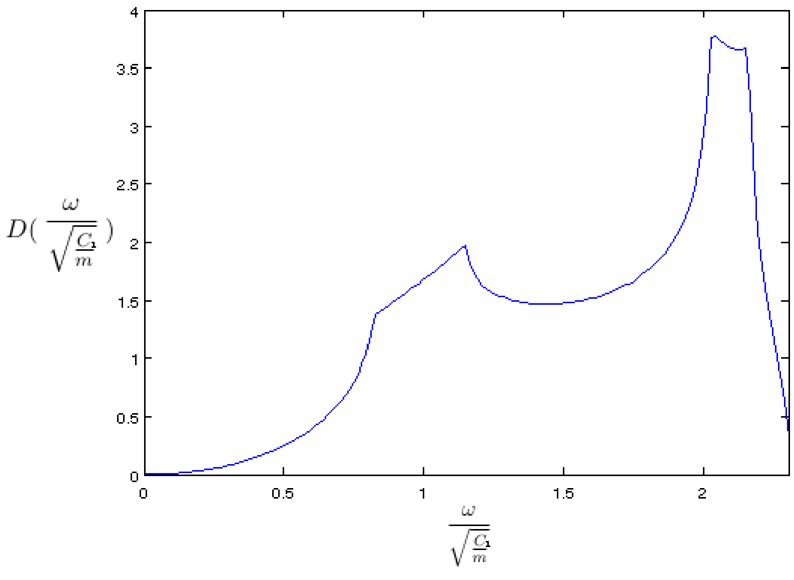

Figure 6: Phonon density of states for simple cubic, nearest and next nearest neighbours, $\frac{C_1}{C_2} = 6$

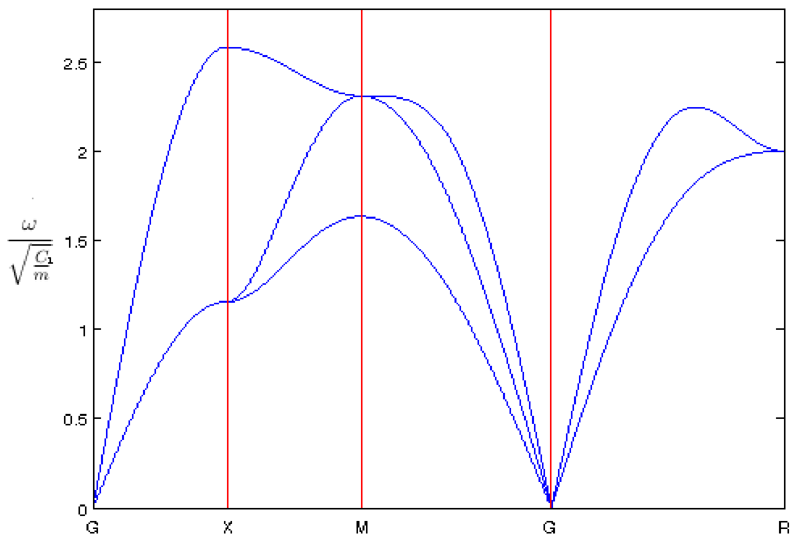

Figure 7: Phonon dispersion relation for simple cubic, nearest and next nearest neighbours, $\frac{C_1}{C_2} = 3$

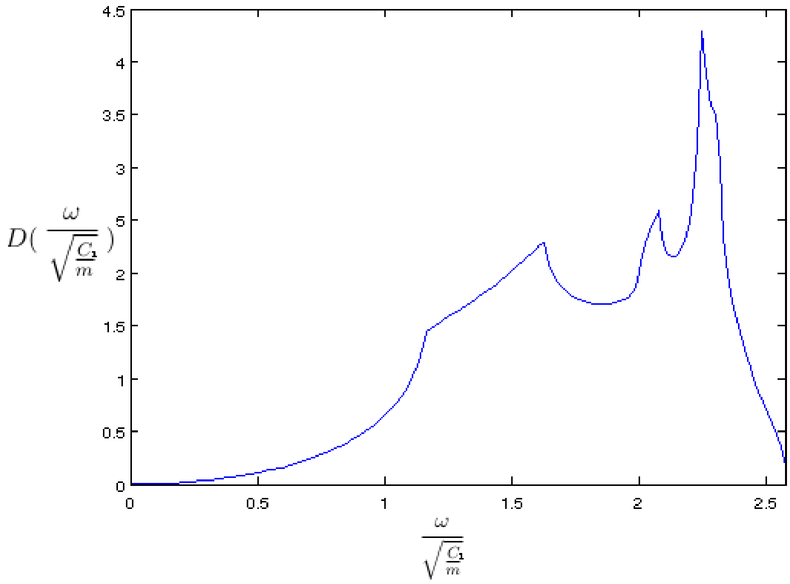

Figure 8: Phonon density of states for simple cubic, nearest and next nearest neighbours, $\frac{C_1}{C_2} = 3$

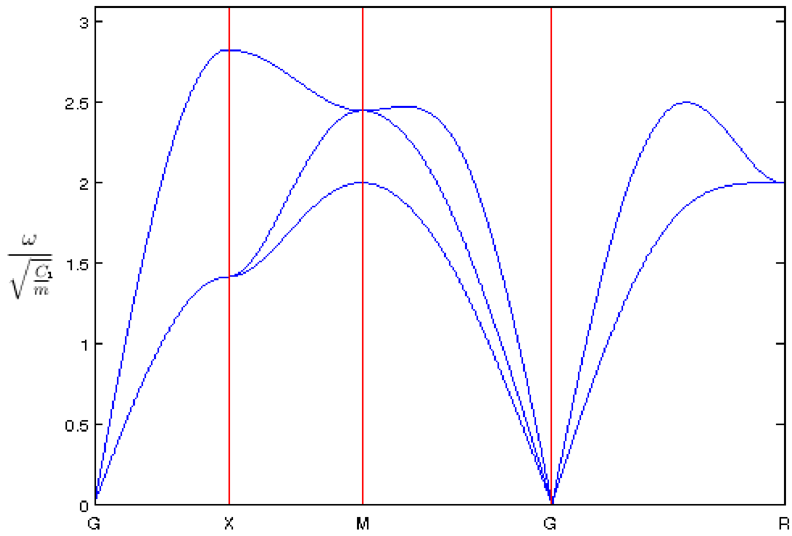

Figure 9: Phonon dispersion relation for simple cubic, nearest and next nearest neighbours, $\frac{C_1}{C_2} = 2$

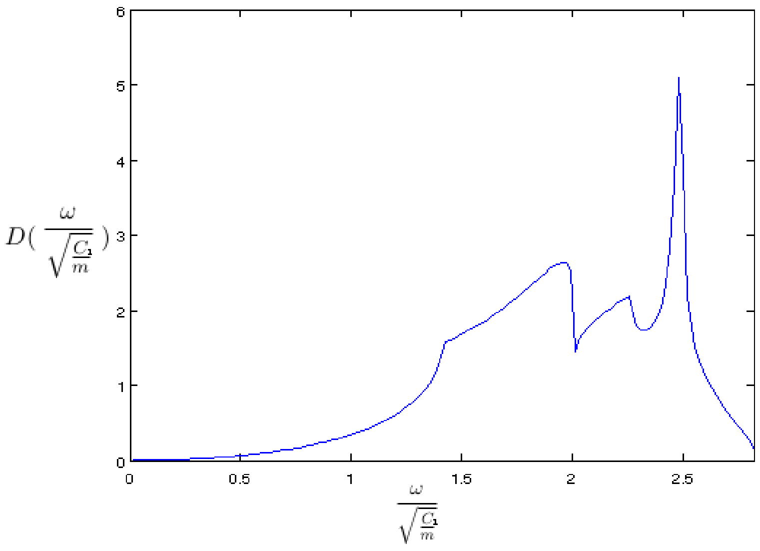

Figure 10: Phonon density of states for simple cubic, nearest and next nearest neighbours, $\frac{C_1}{C_2} = 2$

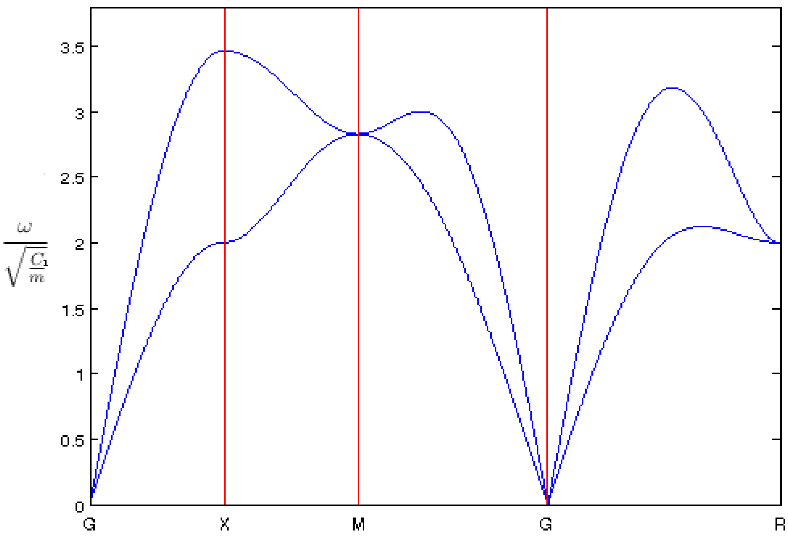

Figure 11: Phonon dispersion relation for simple cubic, nearest and next nearest neighbours, $\frac{C_1}{C_2} = 1$

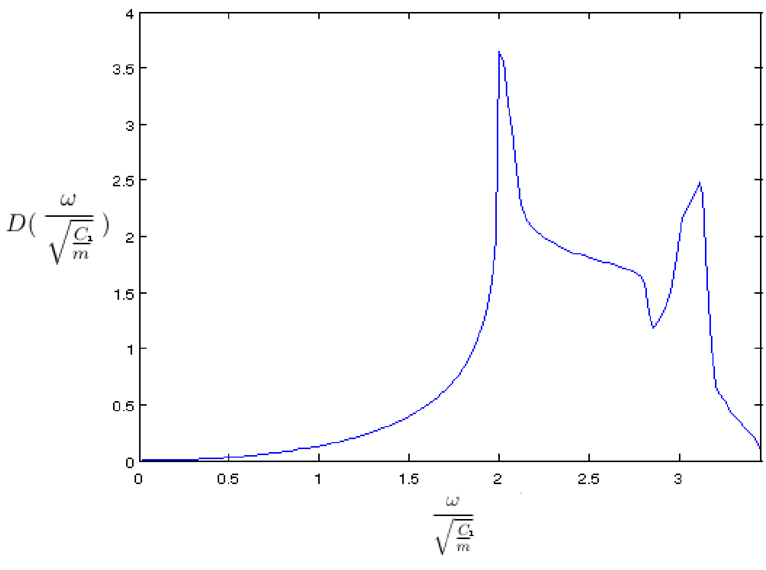

Figure 12: Phonon density of states for simple cubic, nearest and next nearest neighbours, $\frac{C_1}{C_2} = 1$