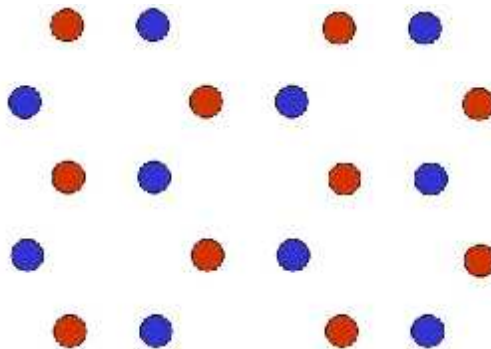
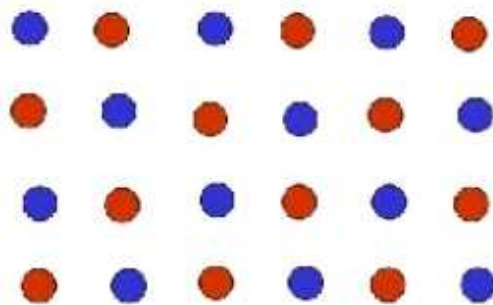


Berechnung der Madelung Konstante und des kritischen Radius von zweidimensionalen Kristallstrukturen

Manuel Pfeifenberger, 0831365



Studentenprojekt Festkörperphysik WS 2010/11

Projektbeschreibung

1 Madelung Konstante

Der erste Teil des Projekts bestand darin, die Madelung Konstante M der beiden auf dem Deckblatt schematisch dargestellten zweidimensionalen Kristallstrukturen (links kubisch; rechts hexagonal) zu bestimmen.

$$M = \sum_i \frac{(\pm)z_j}{r_{ij}/r_0}$$

z_j ... Anzahl der Atome in der Nachbarschale j . (positives Vorzeichen wenn Atom entgegengesetzte Ladung wie Bezugatom besitzt, negatives Vorzeichen bei gleicher Ladung)

r_{ij} ... Abstand der Atome in der Nachbarschale j zum Bezugatom i

r_0 ... Abstand zu den nächsten Nachbarn

Für die Berechnung werden 2 dimensionale Arrays mit negativen und positiven Einsen, welche die beiden Atomsorten darstellen, entsprechend den Strukturen besetzt. Anschließend werden die Abstände zum Ursprung berechnet, sortiert und gleich große Abstände zu Schalen addiert.

Relativ kleine Abstände entsprechen bereits einer großen Anzahl von Schalen (Atome mit demselben Abstand vom Ursprungsatom). Die Distanz von 1000 Abständen zu den nächsten Nachbarn entspricht bei der kubischen Struktur der 216341. Schale und bei der hexagonalen Struktur der 180874. Schale.

Die Mittelwerte der Madelung Konstanten für diese Distanz betragen:

- ***Kubische Struktur: 1.6158***
- ***Hexagonale Struktur: 1.5418***

Die Rechenzeit des Computers wird dabei schon sehr lang (etwa 2,5-3 min). Im Vergleich dazu, benötigt das Programm für eine Distanz von 100 Nachbarsabständen nur den Bruchteil einer Sekunde.

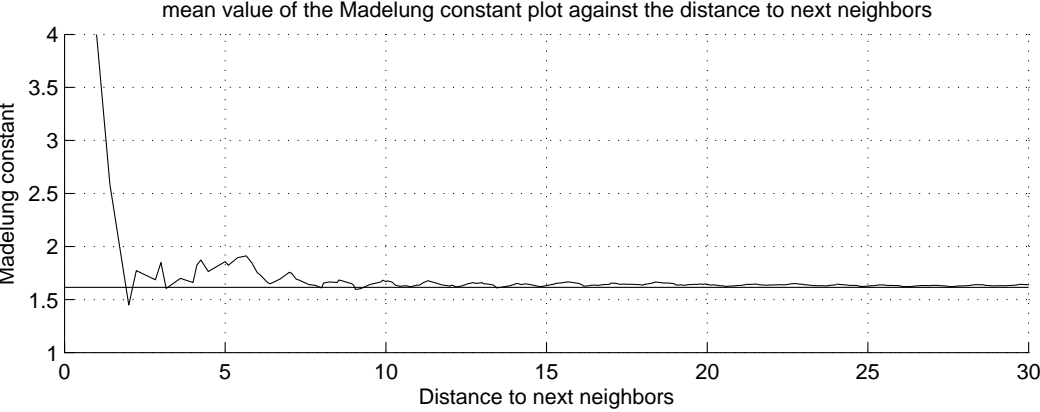
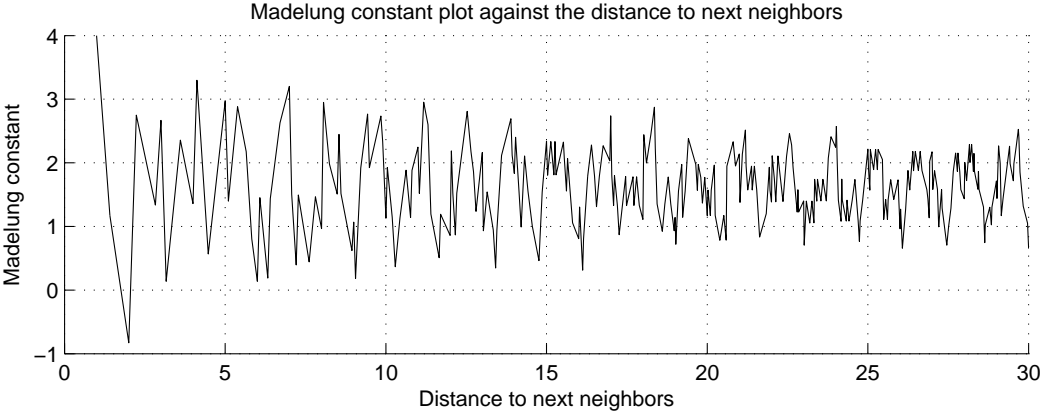
Die folgenden Seiten beinhalten:

- Auszüge aus den Tabellen der Matlab Konsole
- Diagramme der Madelungkonstante bzw. der laufenden Mittelwerte derselbigen
- Programmcode (Madelung2dim.m; Madelung2dimhex.m)

(jeweils berechnet bis zu 30 Nachbarsabständen bei beiden Strukturen)

1.1 Tabelle (aus Matlab Konsole) und Plots der kubischen Struktur

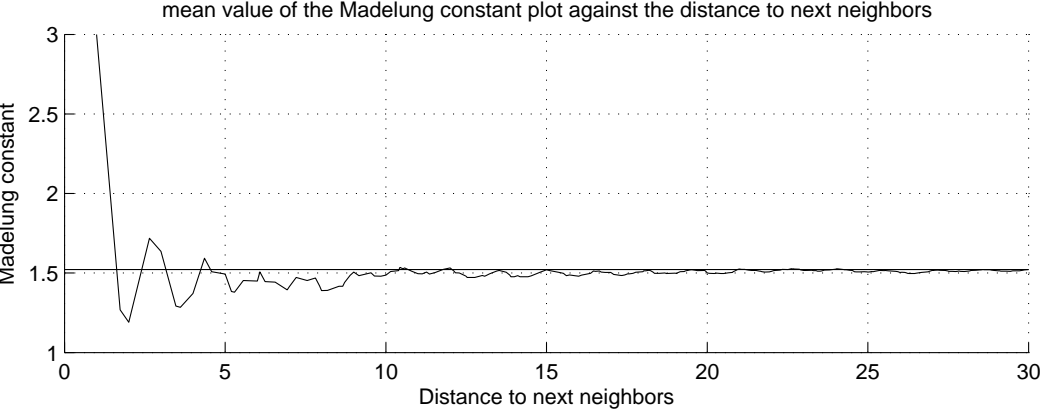
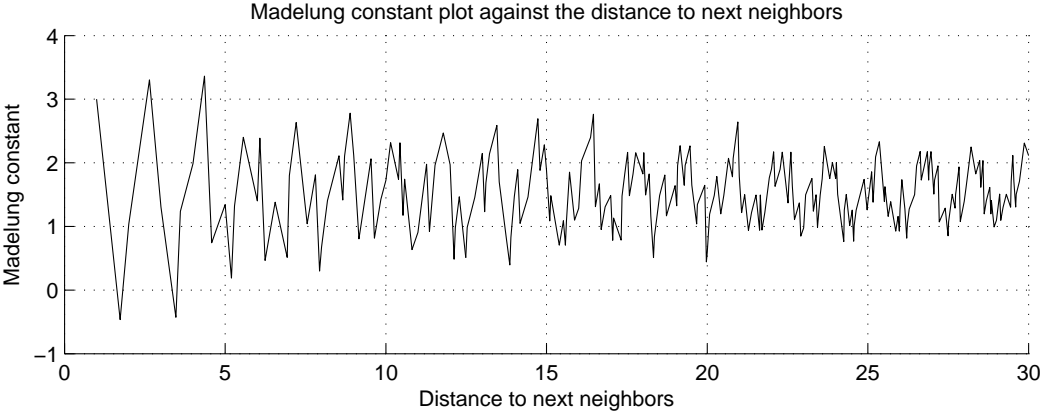
Shell	Distance	Number of Neighbors	Charge	Madelung Constant
1	1.00000	4	-1	4.000
2	1.41421	4	1	1.172
3	2.00000	4	1	-0.828
4	2.23607	8	-1	2.749
5	2.82843	4	1	1.335
6	3.00000	4	-1	2.668
7	3.16228	8	1	0.139
8	3.60555	8	-1	2.357
9	4.00000	4	1	1.357
10	4.12311	8	-1	3.298
11	4.24264	4	1	2.355
12	4.47214	8	1	0.566
13	5.00000	12	-1	2.966
14	5.09902	8	1	1.397
15	5.38516	8	-1	2.883
16	5.65685	4	1	2.176
17	5.83095	8	1	0.804
18	6.00000	4	1	0.137



1.2 Tabelle (aus Matlab Konsole) und Plots der hexagonalen Struktur

Shell Distance Number of Neighbors Charge Madelung Constant

Shell	Distance	Number of Neighbors	Charge	Madelung Constant
1	1.00000	3	-1	3.000
2	1.73210	6	1	-0.464
3	2.00000	3	-1	1.036
4	2.64580	6	-1	3.304
5	3.00000	6	1	1.304
6	3.46410	6	1	-0.428
7	3.60560	6	-1	1.236
8	4.00000	3	-1	1.986
9	4.35890	6	-1	3.362
10	4.58260	12	1	0.744
11	5.00000	3	-1	1.344
12	5.19620	6	1	0.189
13	5.29150	6	-1	1.323
14	5.56780	6	-1	2.400
15	6.00000	6	1	1.400
16	6.08280	6	-1	2.387
17	6.24500	12	1	0.465
18	6.55740	6	-1	1.380



1.3 Programmcode zur Berechnung der Madelung Konstante der kubischen Struktur

Matlab file: Madelung2dim.m

```
%Calculating the madelung constant of a 2dim cubic structure

clear all;

%max distance for the calculation (in units of lattice constant)
na=30;

%first sort of atoms (positive charge)
gc=1*zeros(2*na+1,2*na+1);
gc(2:2:end,2:2:end)=1;
gc(1:2:end,1:2:end)=1;
%second sort (negative charge)
gc2=-1*ones(2*na+1,2*na+1);
gc2(2:2:end,2:2:end)=0;
gc2(1:2:end,1:2:end)=0;
g=gc+gc2; % 2dim cubic structure

tic;
dmat=zeros((2*na+1)*(2*na+1),4);

for b=1:2*na+1
    for a=1:2*na+1
        if g(a,b) ~=0
            dmatn=(b-1)*(2*na+1)+a;
            b2=b-na-1;
            a2=a-na-1;
            dmat(dmatn,1)=sqrt(a2^2+b2^2); %distance
            dmat(dmatn,2)=a2; %coordinates
            dmat(dmatn,3)=b2;
            dmat(dmatn,4)=g(a,b); %charge
        end
    end
end

charge0=find(dmat(:,4)~=0); %order the distances
dmat2=dmat(charge0,:);
dmat3=sortrows(dmat2,1);
dmat4=dmat3(2:end,:);
maxr=find(dmat4(:,1)≤na); %confine the crystal structure
dmat5=dmat4(maxr,:);

%calculate number of atoms with the same distance (shell)
dmata=zeros(size(dmat5,1),4);
dmata(1,:)= [0,dmat5(1,1),1,dmat5(1,4)];
distanceref=dmat5(1,1);
ish=1; %shell number
```

```

for la=2:size(dmat5,1)
    if dmat5(la,1)≠distanceref
        dmata(ish,1)=ish;
        ish=ish+1;
        dmata(ish,2)=dmat5(la,1);
        dmata(ish,3)=1;
        dmata(ish,4)=dmat5(la,4);
        distanceref=dmat5(la,1);
    else
        dmata(ish,3)=dmata(ish,3)+1; %Counting the atoms/shell
    end
end

dmata(ish,1)=ish;
distance0=find(dmata(:,2)≠0);
dmata2=dmata(distance0,:);

%madelung constant
distance1=dmata2(:,2);
numberofa=dmata2(:,3);
madelung_const=zeros(1,size(dmat2,1));
madelung_const=numberofa./distance1;

for n= 2:size(dmata2,1); %reference charge +1
    if dmata2(n,4)<0
        madelung_const(n)=madelung_const(n-1)+madelung_const(n);
    else
        madelung_const(n)=madelung_const(n-1)-madelung_const(n);
    end
end

%mean value of the madelung constant
mv=zeros(1,size(dmata2,1));
for m=1:size(dmata2,1)
    mv(1,m)=sum(madelung_const(1:m))/m;
end
time=toc;

%table
fprintf('Shell\t Distance\tNumber of Neighbors\tCharge\t Madelung Const.\n');
disp('_____');
for o=1:size(dmata2,1)
    fprintf(' %4d \t %4.5f \t %7d \t %11d \t %12.3f \n',dmata2(o,1),...
            dmata2(o,2),dmata2(o,3),dmata2(o,4),madelung_const(o));
end
fprintf('\n');
disp(['Computing Time: ',num2str(time),' sec.']);
disp(['madelung constant:',num2str(mv(1,size(dmata2,1)))]);

%Plot
figure(1)

```

```
subplot(2,1,1)
grid on;
hold on;
plot(distancel,madelung_const);
title('Madelung constant plot against the distance to next neighbors');
xlabel('Distance to next neighbors');
ylabel('Madelung constant');
hold off;
subplot(2,1,2)
grid on;
hold on;
plot(distancel,mv,[0,distancel(end)],[1.6155,1.6155]);
title('mean value plot against the distance to next neighbors');
xlabel('Distance to next neighbors');
ylabel('Madelung constant');
hold off;

%print -deps madcub1
```

1.4 Programmcode zur Berechnung der Madelung Konstante der hexagonalen Struktur

Das Programm ist bis auf den Aufbau der Gitterstruktur am Anfang mit dem Programmcode unter 1.3 ident. Deshalb wird nur dieser Teil angeführt.

Matlab file: Madelung2dimhex.m

```
%calculating Madelung constant of a 2dim hexagonal structure

clear all;

%basic structure
gs1=[1,0,0,0,-1,0];
gs2=[0,-1,0,1,0,0];
gs=[gs1;gs2];

%lattice structure
distc=30; % max distance for the calculation
        %(in units of the distance to the nearest neighbors)
xh=ceil(sqrt(3)*2*distc);
yh=ceil(sqrt(3)*2*distc);
xhh=ceil(xh/6);
yhh=ceil(yh/2);
hexlat= repmat(gs,yhh,xhh);
hexlat=[hexlat,hexlat];
hexlat=[hexlat;hexlat];
hexlat1=hexlat(:,1);
hexlat=[hexlat,hexlat1];
hexlat2=hexlat(1,:);
hexlat=[hexlat;hexlat2]; %2dim hexagonal structure

:
:
:
```


2 Kritischer Radius

Im zweiten Teil wird das kritische Radiusverhältnis, ab welchem die Kristallstrukturen nicht mehr stabil sind, bestimmt.

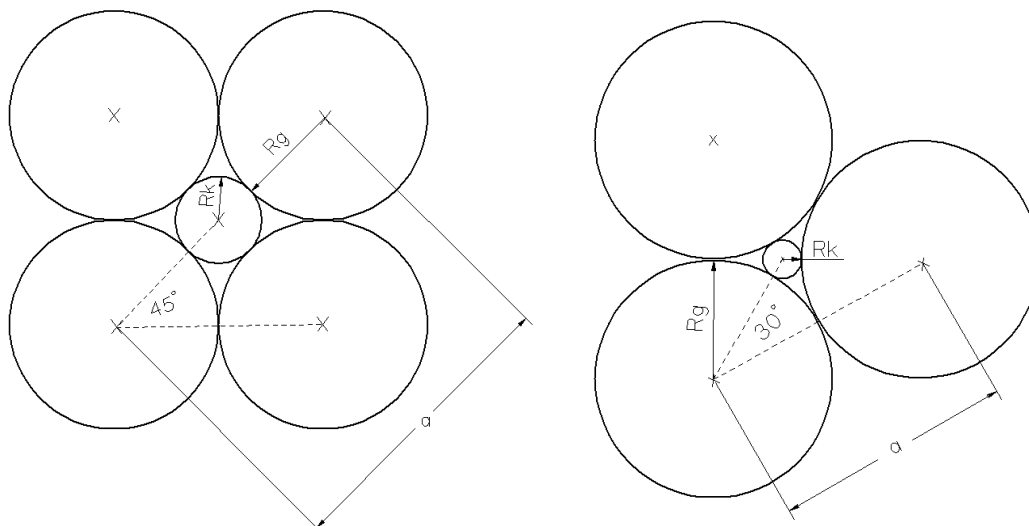


Abbildung 1: Berechnung des kritischen Radiusverhältnisses (a=Abstand zum nächsten Nachbarn)

Für die kubische Struktur (siehe Abbildung 1 links) gilt:

$$\begin{aligned} (R_G + R_K) \underbrace{\cos(45)}_{1/\sqrt{2}} &= R_G \\ \Rightarrow R_K &= (\sqrt{2} - 1) R_G \\ \Rightarrow R_G/R_K &= 1/(\sqrt{2} - 1) = 2.414 \end{aligned}$$

Für die hexagonale Struktur (siehe Abbildung 1 rechts) gilt:

$$\begin{aligned} (R_G + R_K) \underbrace{\cos(30)}_{\sqrt{3}/2} &= R_G \\ \Rightarrow R_K &= (2/\sqrt{3} - 1) R_G \\ \Rightarrow R_G/R_K &= \sqrt{3}/(2 - \sqrt{3}) = 6.464 \end{aligned}$$

Wenn der kritische Radius erreicht wird, ist die Struktur nicht mehr stabil, da das mittlere, kleinere Atom nicht mehr eindeutig zwischen den größeren Atomen positioniert ist. Daraus folgt, dass bei der hexagonalen Struktur der Größenunterschied zwischen den beiden Atomsorten fast dreimal so groß sein kann, als bei der kubischen Struktur.

3 Quellenangaben

- Iterative Bestimmung der Madelung Konstante für CsCl (Project Work Solid State Physics WS2009/10; Yao Shan, Robert Krisper)
- Abbildungen der beiden Strukturen (Deckblatt):
<http://lamp.tu-graz.ac.at/hadley/ss1/lectures10/nov10/slide17.html>