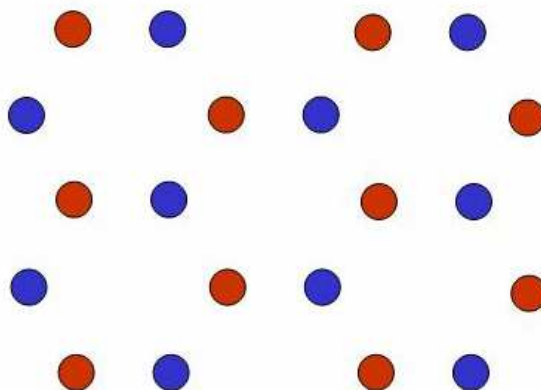
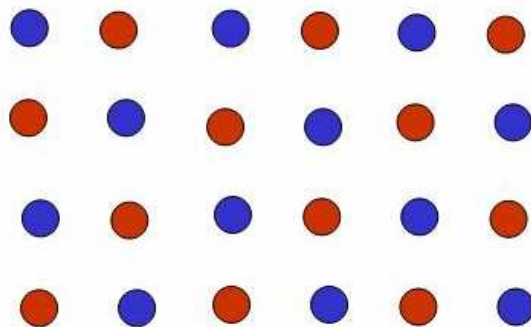


Projektarbeit Molekül- und Festkörperphysik
WS 2010/2011

Iterative Bestimmung der Madelung-Konstante für zwei zweidimensionale Kristallstrukturen

Robert TRIEBL, 0830677



Einleitung

Ziel dieses Projekts war es, die Madelung-Konstante zweier zweidimensionaler Kristallstrukturen zu bestimmen. Diese ist durch die Gleichung

$$M_i = \sum_j \frac{z_j}{r_{ij}/r_0}$$

gegeben.

z_j ... Anzahl der Atome in einer Schale

r_{ij} ... Abstand des Atoms j vom Bezugatom i .

In allen Programmen wurde $r_i = (0,0,0)$ gewählt.

r_0 ... Abstand zum nächsten Nachbarn

Da meine Programme aus einer Vielzahl von Schleifen und keinen Matrizen bestehen, wurden sie in C geschrieben. Sie sind am Ende dieses Dokuments zu finden. Die graphische Auswertung hingegen erfolgte mit Matlab.

Zur Unterscheidung der beiden Gitter wird das erste zweidimensionale NaCl Struktur, die zweite hexagonale Struktur genannt, obgleich es sich wegen der unterschiedlichen Atomsorten um kein wirklich hexagonales Gitter handelt.

Zweidimensionale NaCl-Struktur

Berechnung der einzelnen Kugelschalen:

Der Programmcode befindet sich im Anhang unter `madelungconstantNaCl1.c`.

Zuerst wurde die Madelung-Konstante in Abhängigkeit vom Abstand berechnet, wobei jeder im Kristall vorkommende Abstand berücksichtigt wurde. Dazu wurde in einer Schleife der k -te Abstand zum Ursprung gesucht, indem in zwei weiteren Schleifen der Gitterplatz (i,j) betrachtet wurde. Der kleinste so gefundene Abstand, der größer als der Abstand der vorigen Schale ist, wurde als Eintrag in den Vektor $R_1[k]$ abgespeichert. Für alle Atome mit diesem Abstand wurde an der k -ten Stelle des Vektors M der Summand $1/r(i,j)$ mit entsprechendem Vorzeichen addiert. In meinem endgültigen Programm wurde nur über einen Teil der Atome (entsprechend der Symmetrie über ein Achtel) summiert, daher wurden die Summanden mit dem Faktor „symmetryfactor“ modifiziert. $M[k]$ gibt somit den Beitrag der k -ten Schale mit dem Abstand $R_1[k]$ zur Madelung-Konstante an.

Schließlich wurden noch die Elemente des zu langen Vektors R_1 in einen neuen Vektor R kopiert und die kumulierte Summe von M im Vektor „madelungconstant“ gespeichert.

Diese so erhaltene Kurve (Abbildung 1) oszilliert sehr stark, die Madelung-Konstante kann nur grob abgeschätzt werden. Um zu untersuchen, ob Konvergenz zu erwarten ist, habe ich die Madelung-Konstante bis $R_{max}=400$ geplottet, was aufgrund der vielen unterschiedlichen Abständen bei großem R schon an die Grenzen der Rechenkapazität geht (Abbildung 2).

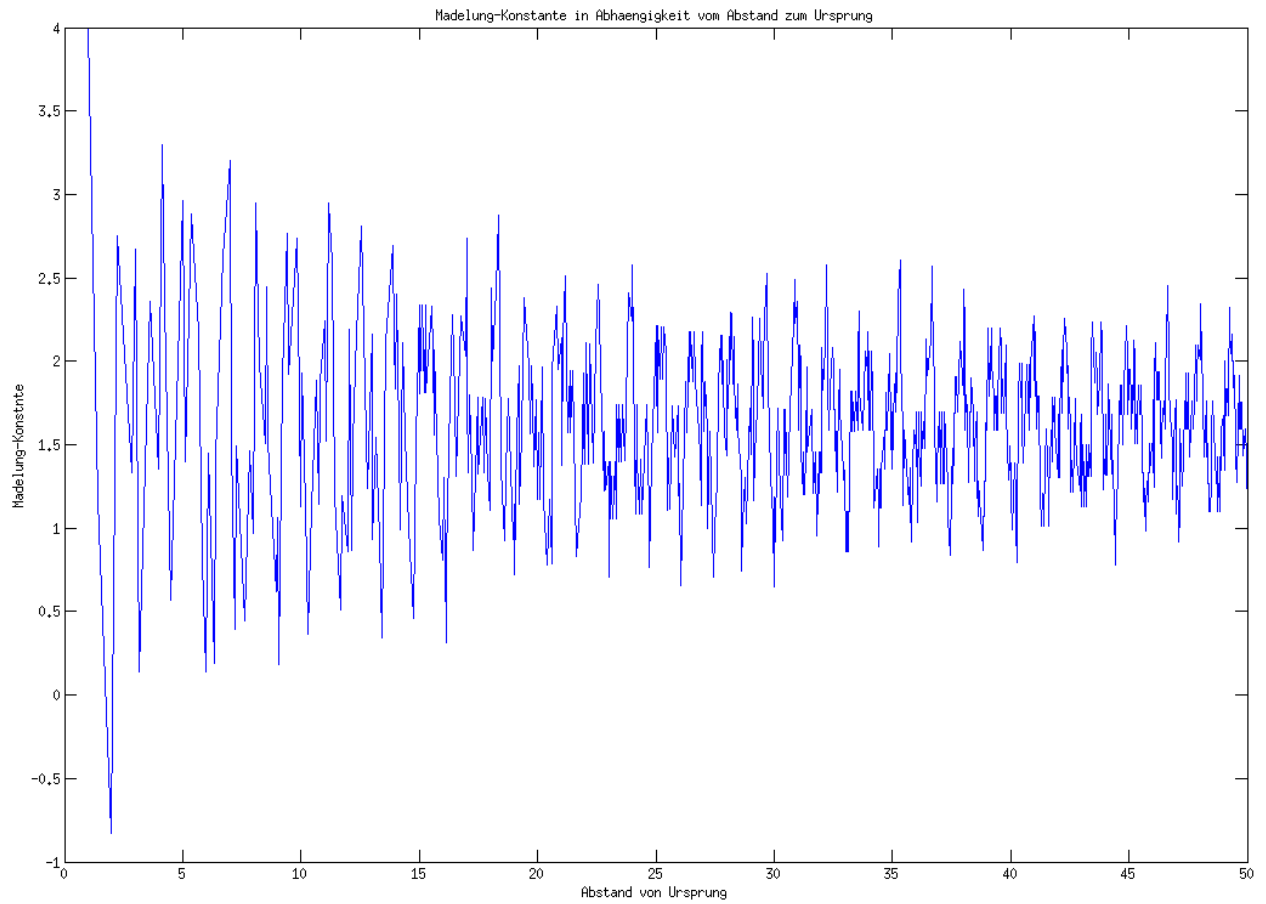


Abbildung 1: Madelung-Konstante in Abhängigkeit vom Abstand bis $R=50$.

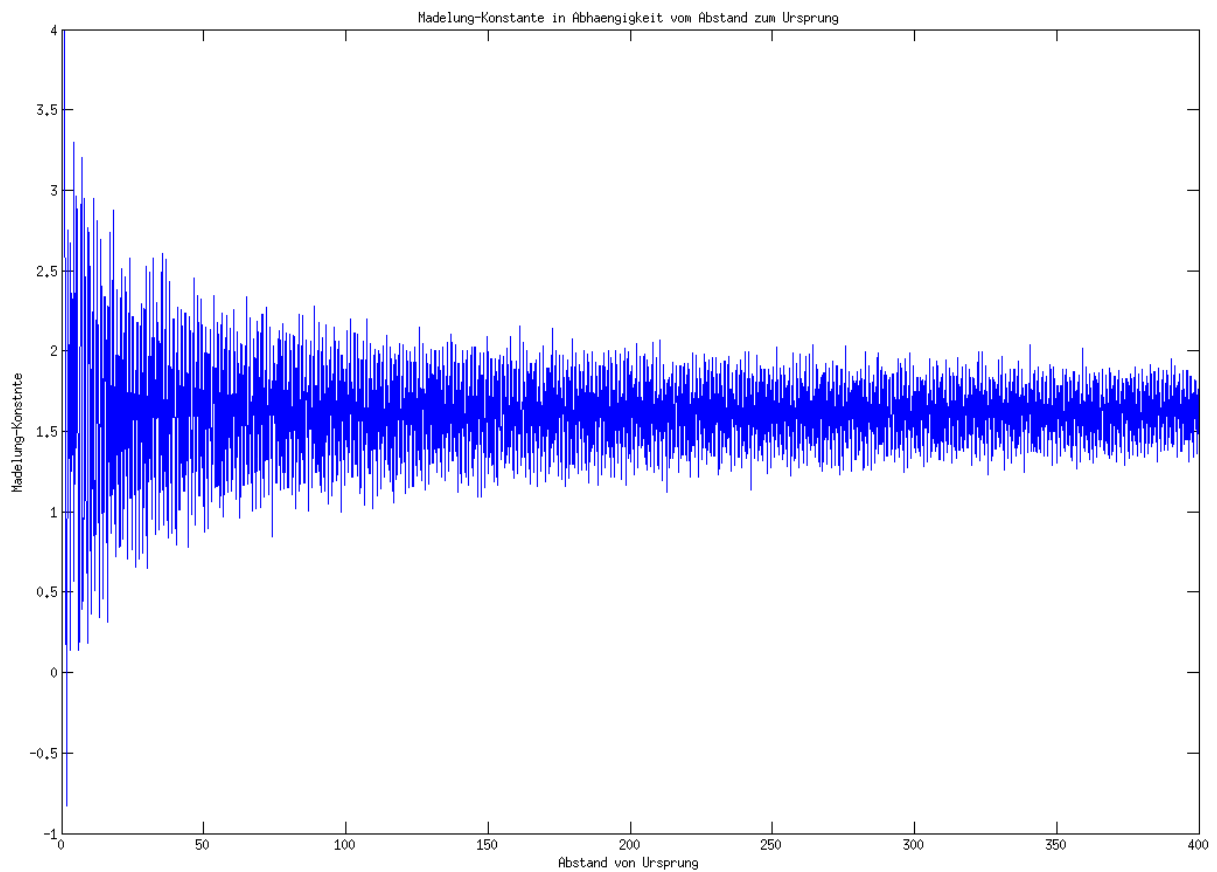


Abbildung 2: Madelung-Konstante in Abhängigkeit vom Abstand bis $R=400$.

Da auch größerer Rechenaufwand offensichtlich keine großen Fortschritte bringt, wurde der kumulierte Mittelwert berechnet. Ich habe mich dabei aber auf die letzten 50% der Punkte beschränkt, da hier die Ausreißer geringer ausfallen als zu Beginn. Als rote Linie wurde der exakte Wert (1.615543) eingezeichnet. Der Mittelwert scheint also ein guter Schätzwert für die Madelung-Konstante zu sein (Abbildung 3).

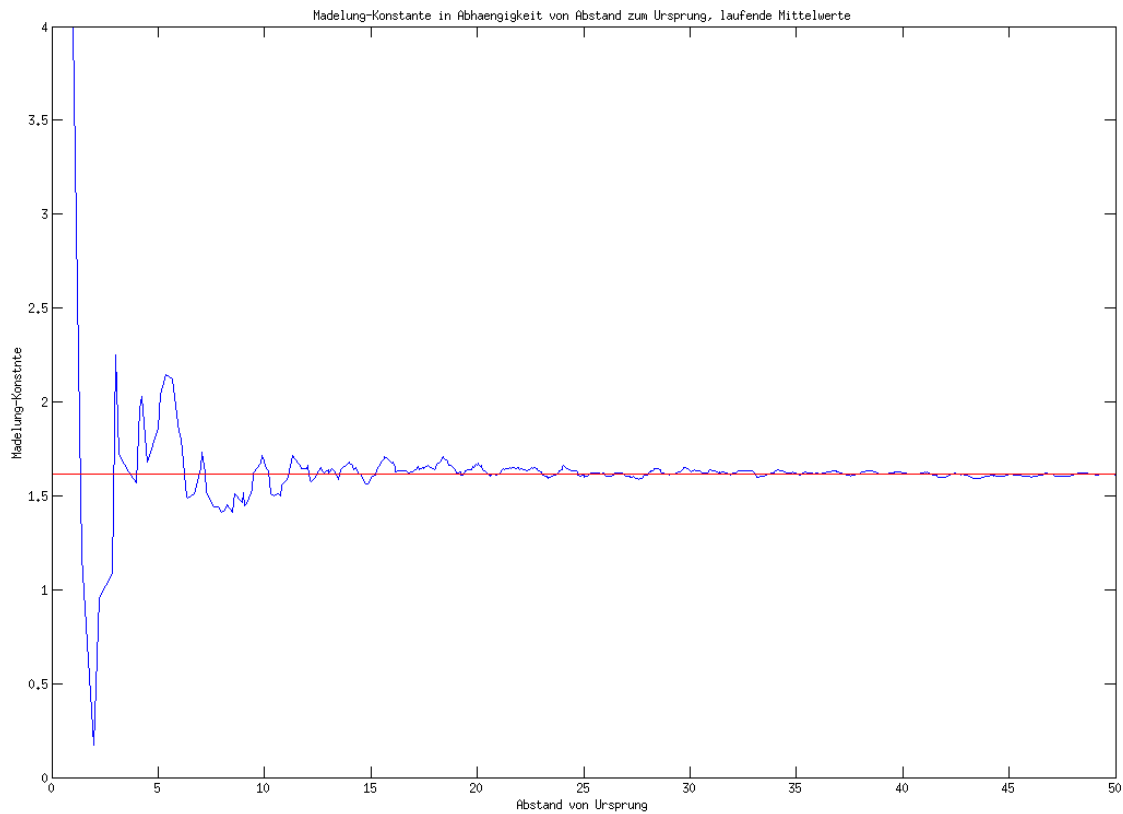


Abbildung 3: Laufende Mittelwerte der Madelung-Konstante in Abhängigkeit vom Abstand bis $R=50$.

Abschließend habe ich noch die Leistungsfähigkeit dieses Programms untersucht und den Mittelwert bis $R=400$ graphisch dargestellt, jedoch mit anderer Skalierung der y-Achse.

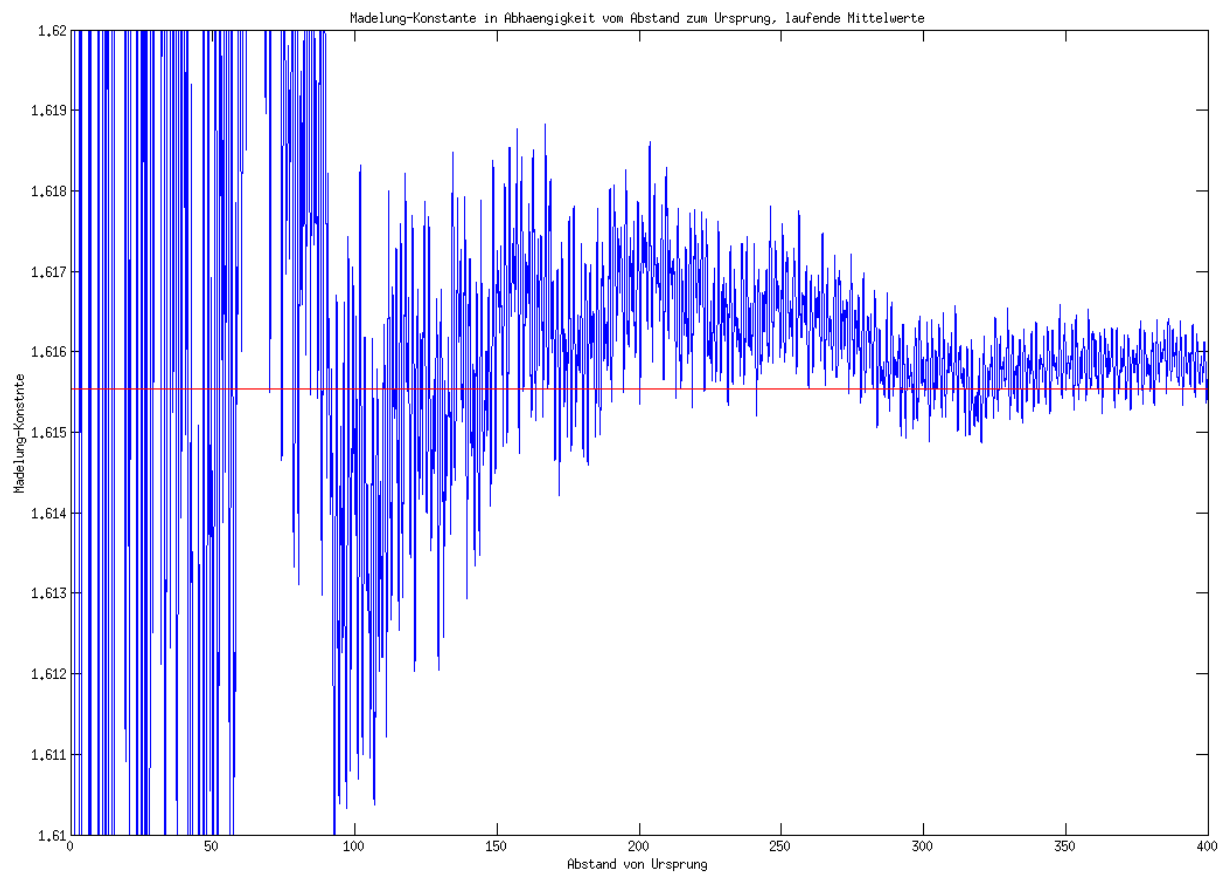


Abbildung 4: Laufende Mittelwerte der Madelung-Konstante in Abhängigkeit vom Abstand bis $R=400$.

Die Madelung-Konstante kann also bis auf eine Ungenauigkeit von 0.001 zu 1.616 ± 0.001 bestimmt werden. Dies stimmt mit dem tatsächlichen Wert von 1.61554 gut überein.

Berechnung Madelung-Konstante für ganzzahlige Abstände:

Programmcode: madlungconstantNaCl2.c

Weiters kann man dazu übergehen, nicht für jeden vorkommenden Abstand die Madelung-Konstante auszurechnen, sondern nur bei ganzzahligen R . Für jedes R werden dann alle Abstände von Atomen mit Abständen vom Ursprung kleiner oder gleich R bestimmt und die Madelung-Konstante ausgerechnet. Da man nicht alle Madelung-Konstante ausrechnet, kann man die Madelung-Konstante bei großen R abschätzen und eine mögliche Konvergenz überprüfen. Für große R ist diese Methode aber auch sehr rechenintensiv, da man sich ja zu jedem R in der Ordnung von R^2 Punkte ausrechnen muss. Nach sehr langer Rechnung erhält man:

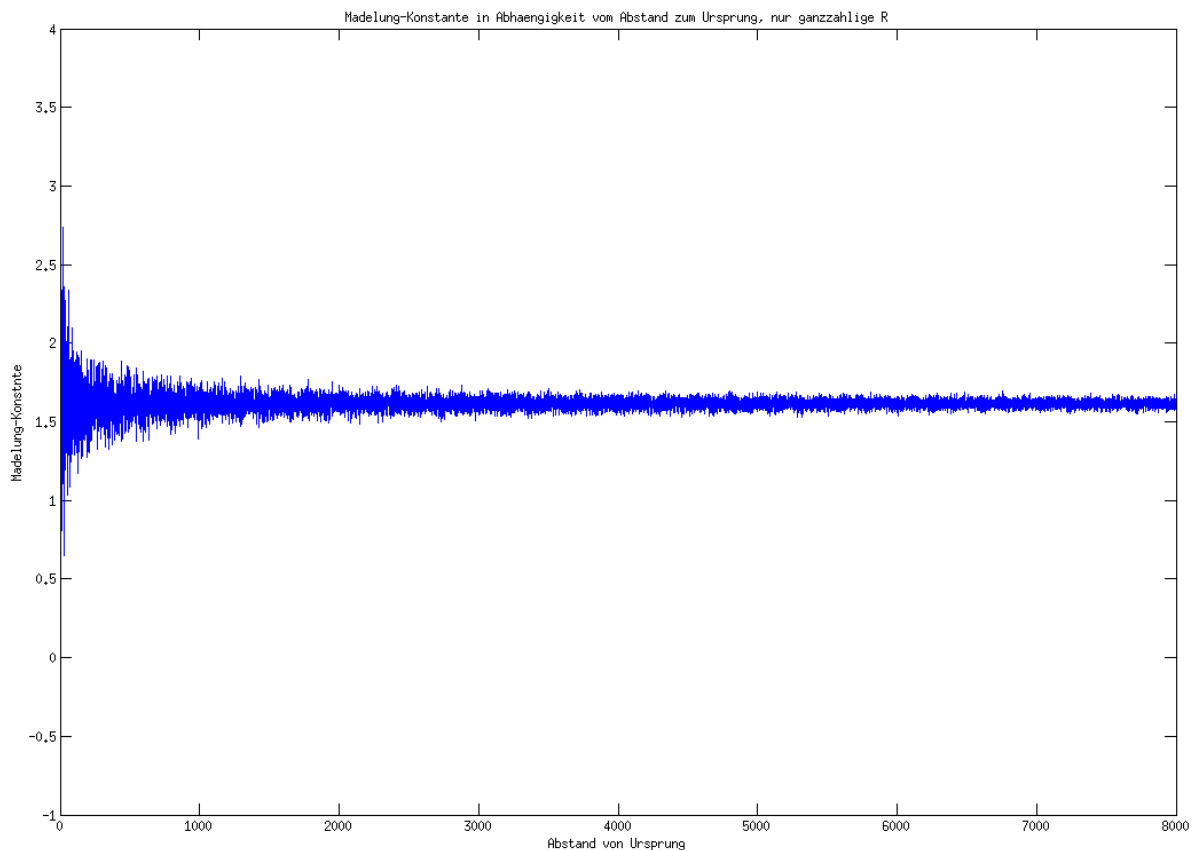


Abbildung 5: Madelung-Konstante für ganzzahlige R bis $R_{\max}=8000$;

Man erkennt, dass man die Madelung-Konstante genauer bestimmen kann, aber die Schwankungen immer noch sehr groß sind. Auch hier kommt man durch Mittelung zu einem genaueren Wert, jedoch muss man bedenken, dass der Mittelwert einen systematischen Fehler enthält, da man den Wert der Madelung-Konstante nur unter sehr speziellen Bedingungen (R eine ganze Zahl) berücksichtigt (siehe Abbildung 6).

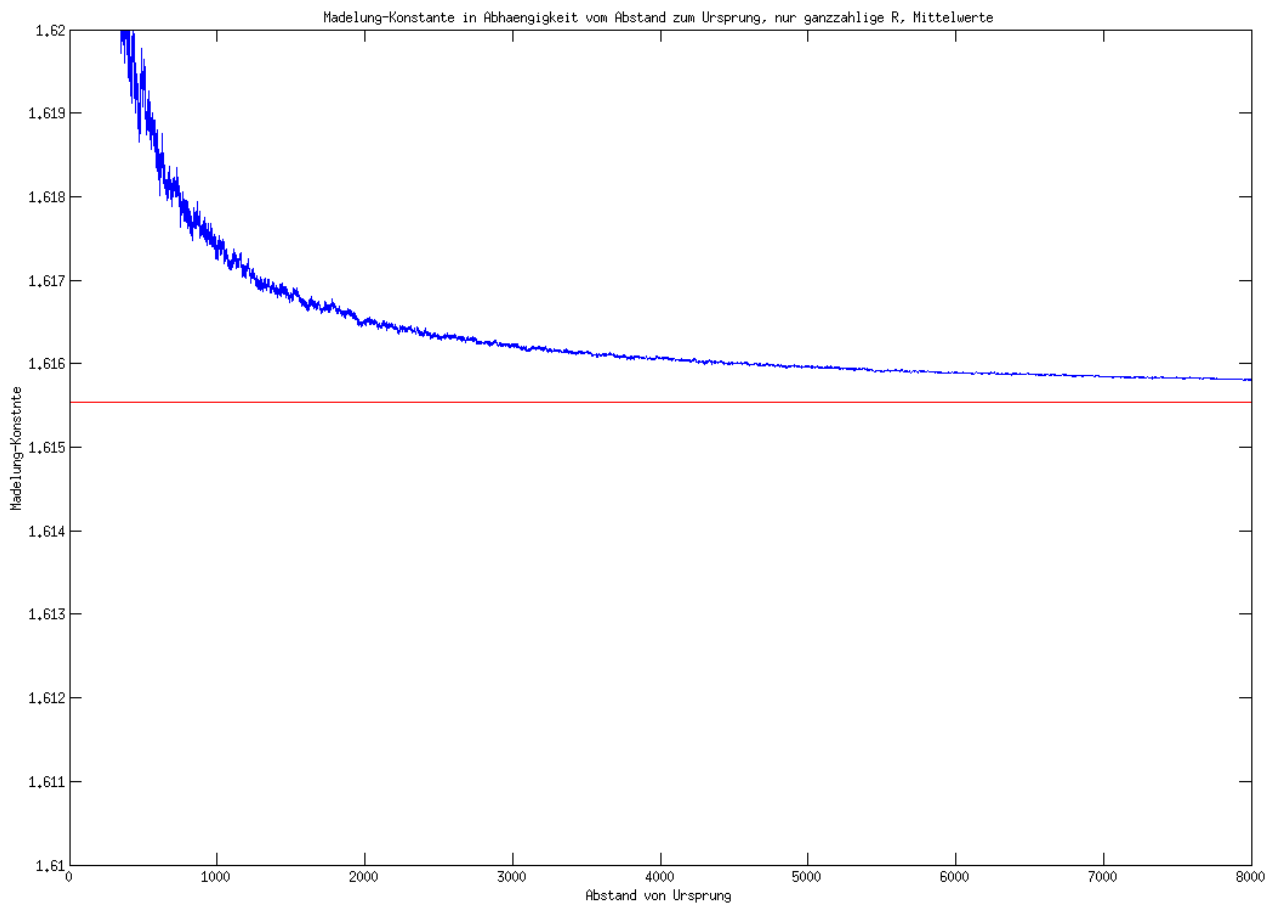


Abbildung 6: Laufende Mittelwerte der Madelung-Konstante für ganzzahlige R

Man erhält, wie schon zuvor, Abweichungen in der Größenordnung von 0.001, jedoch wird diese Fehlerabschätzung aufgrund eines nicht quantifizierbaren systematischen Fehlers schwieriger.

Berechnung über ladungsneutrale Quadrate

Programmcode: `madelungconstantNaCl3.c`

Wesentlich genauer und auch mit viel weniger Rechenaufwand kann man die Madelung-Konstante erhalten, wenn man nicht in Kreisschalen den Abstand vergrößert, sondern in Quadraten mit ganzzahliger Seitenlänge. Die Ladung der Atome an den Rändern wird so gewählt, dass Ladungsneutralität gewährleistet ist. Somit zählen Atome in Ecken nur zu einem Viertel, Atome an der Seite zur Hälfte.

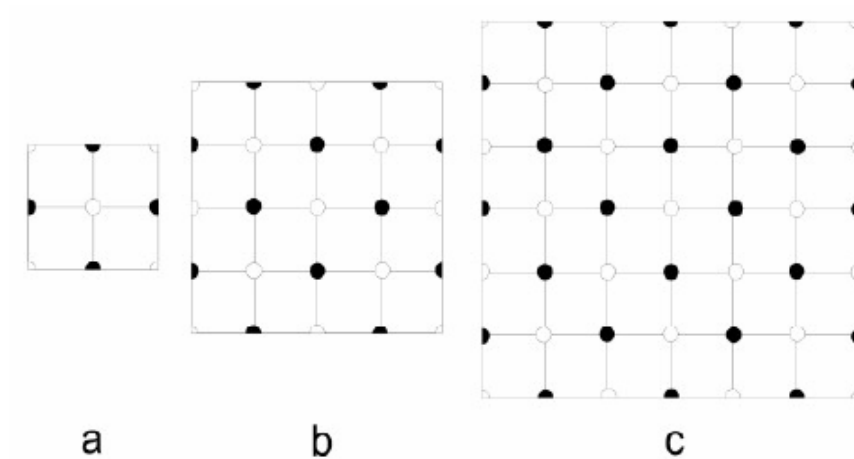


Abbildung 7: Demonstration der ladungsneutralen Einheiten.

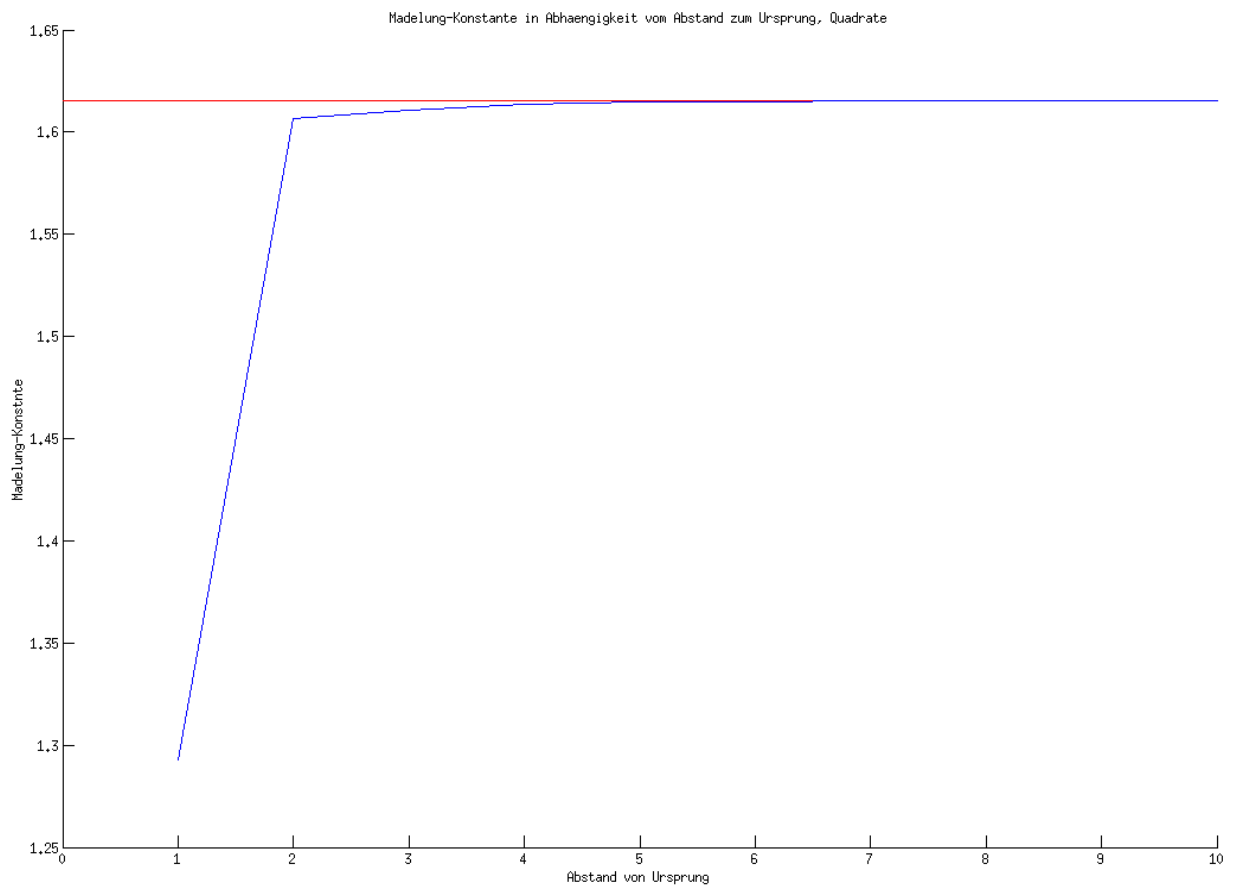


Abbildung 8: Madelung-Konstante für Quadrate mit Seitenlängen bis $2 \cdot 10 = 20$.

Man sieht, dass diese Folge viel schneller konvergiert, bereits bei $R=2$ hat man nur mehr Abweichungen von ca. 0.02.

Diese Methode ist sehr schnell, es reicht schon ein maximaler Abstand von $R=100$ um die Madelung-Konstante auf $10E-7$ genau innerhalb eines Bruchteils einer Sekunde auszurechnen.

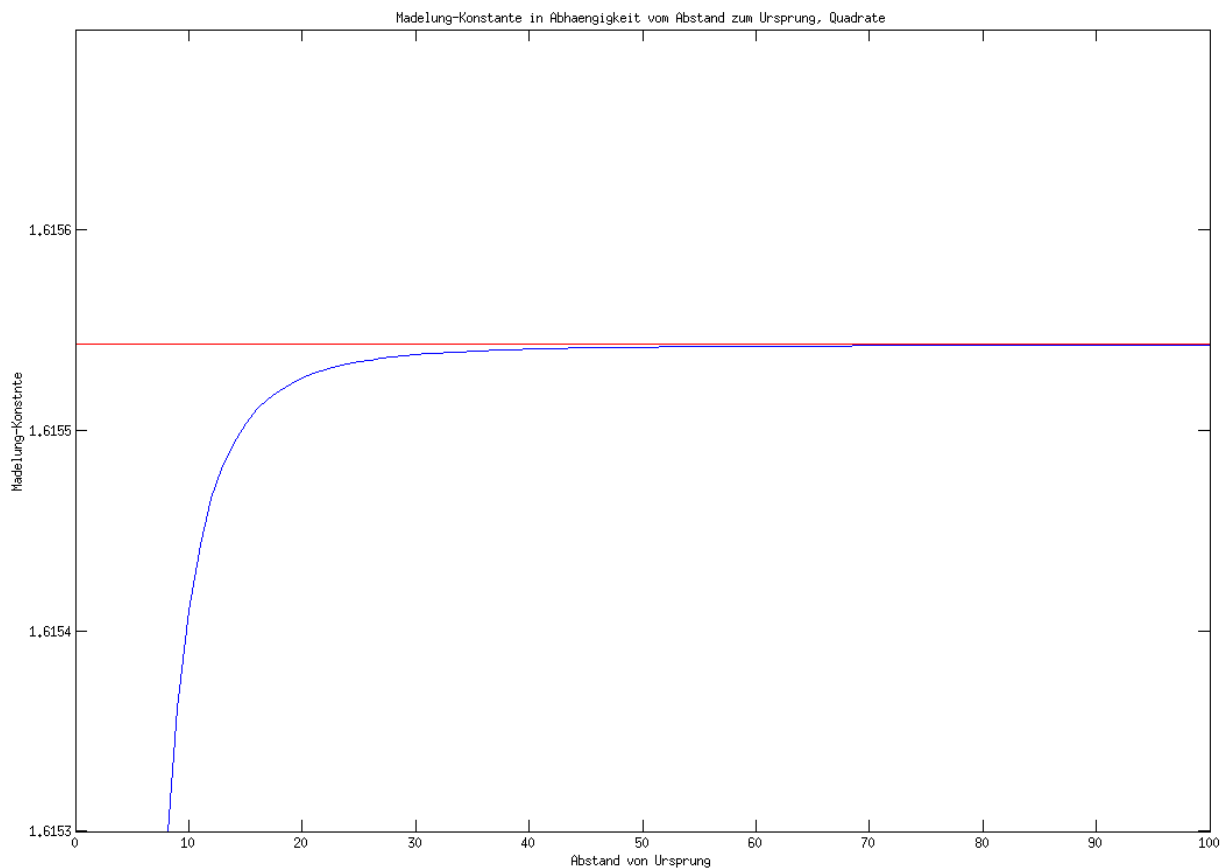


Abbildung 9: Madelung-Konstante für Quadrate mit Seitenlängen bis $2 \cdot 100 = 200$.

Wenn man nicht alle Werte berechnet, um die Konvergenz zu überprüfen, sondern nur gezielt Werte bei gewissen Abständen ausrechnet, kann man die Genauigkeit weiter erhöhen (erste abweichende Ziffer zu genaueren Ergebnissen wurde hervorgehoben):

R=100 M=1.615542**4**9413306650
R=1000 M=1.615542626**5**8020531
R=10000 M=1.615542626711**5**9199
R=30000 M=1.615542626711**2**9933

Man erkennt, dass man auf diese Weise die Madelung-Konstante fast auf beliebige Genauigkeit berechnen kann, wobei man die Genauigkeit der Ergebnisse nur anhand der Größenordnung der Differenzen bei unterschiedlichen Abständen abschätzen kann. Ein mathematisch exaktes Fehlerintervall ist nicht möglich, da man keine oberen Schranken oder Reihenrestglieder etc. hat. Bei den Genauigkeiten, die man bei $R=10000$ erreicht, stößt man aber auch schon an die Grenzen der Numerischen Genauigkeit.

Zweidimensionale hexagonale Struktur

Berechnung der einzelnen Kugelschalen:

Der Programmcode befindet sich im Anhang unter `hexagonalmadelungconstant1.c`.

Hier wurde analog wie bei der zweidimensionalen NaCl Struktur vorgegangen. Zuerst wurde wieder die Madelung-Konstante durch Berechnung der einzelnen Kugelschalen abgeschätzt. Der größte Unterschied in diesem Programm zum vorherigen liegt darin, dass die Koordinaten nicht mehr exakt den Schleifenindices entsprechen, sondern berechnet werden müssen (siehe Programmcode). Außerdem wurden aufgrund der dreizähligen Symmetrie die Schleifengrenzen und Symmetriekoeffizienten anders gewählt. Ergebnisse: Siehe Abbildungen 10 und 11.

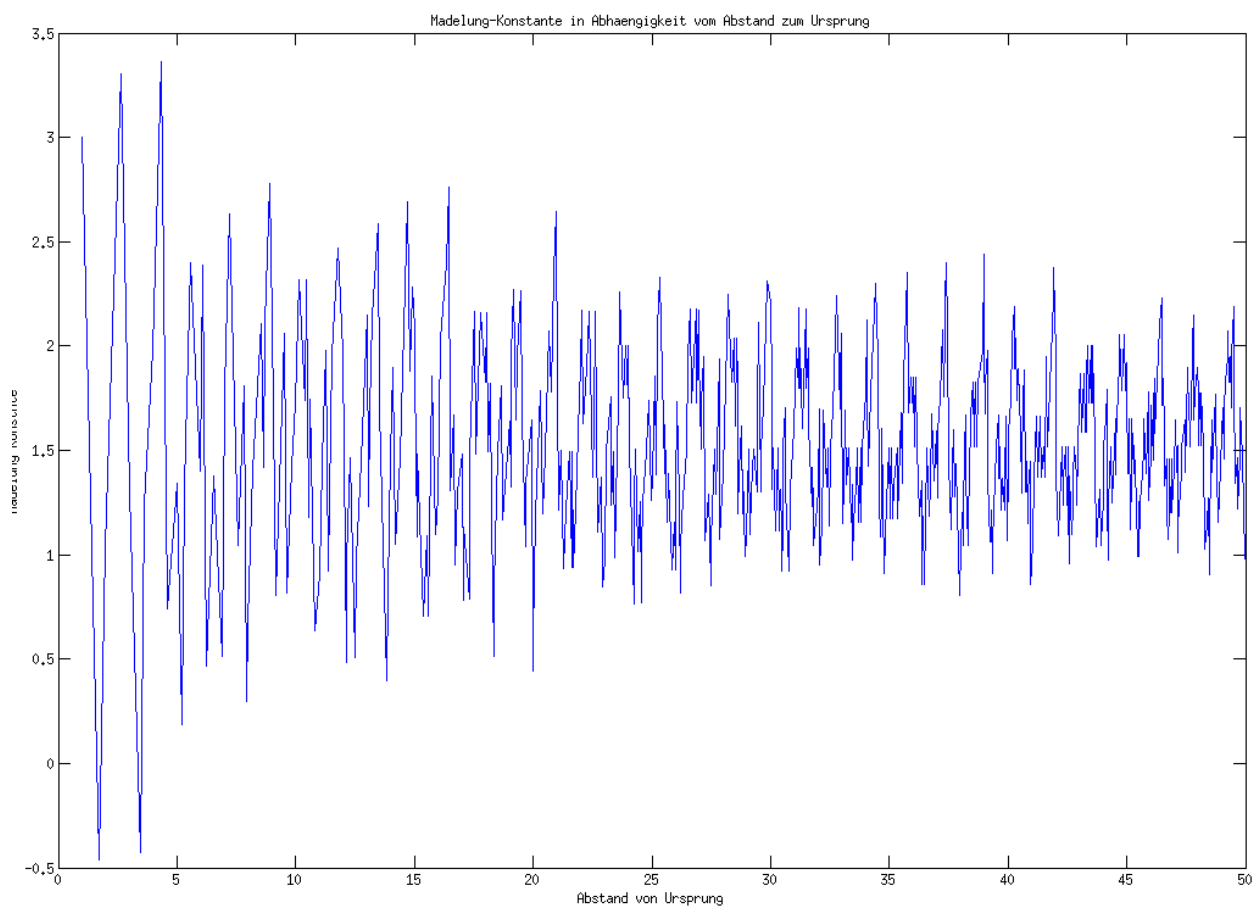


Abbildung 10: Madelung-Konstante in Abhängigkeit von Abstand bis $R=50$.

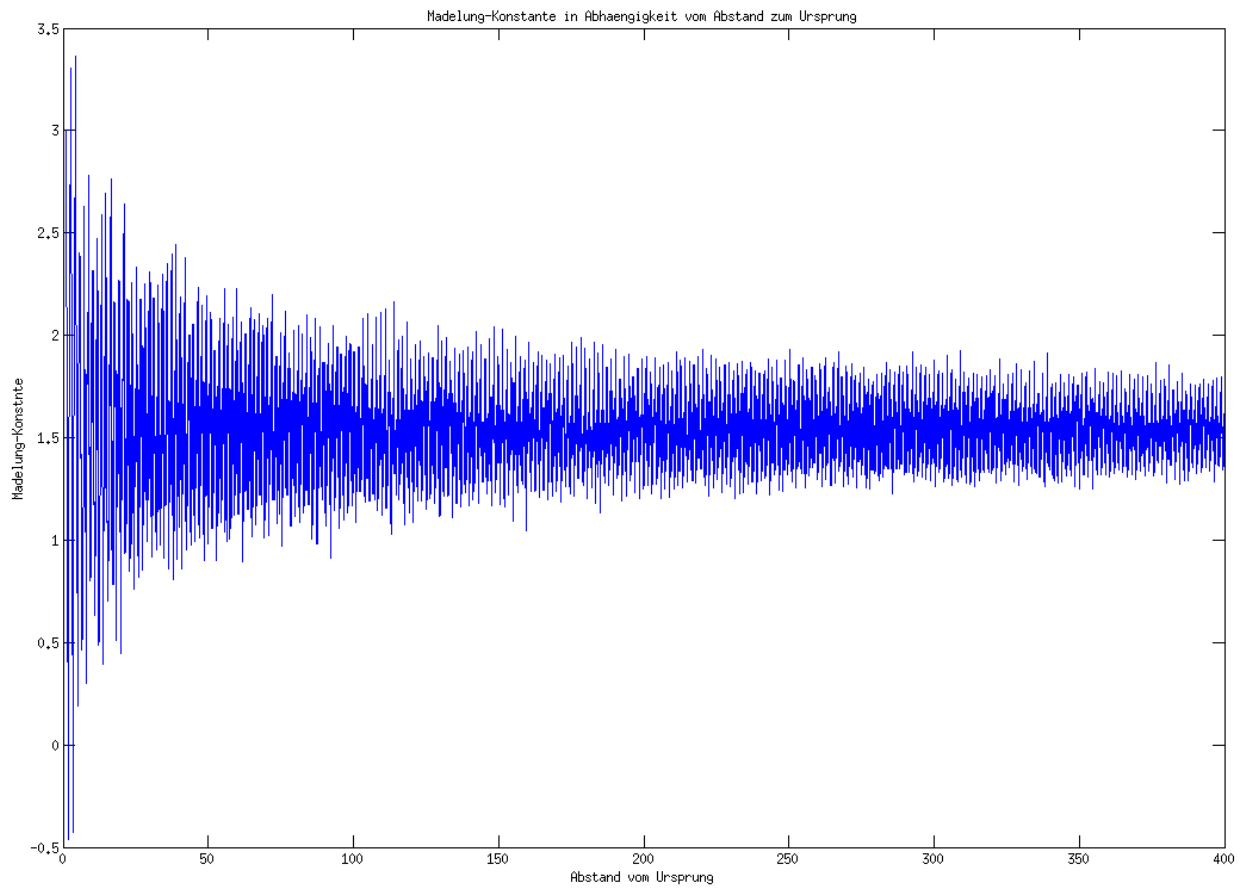


Abbildung 11: Madelung-Konstante in Abhängigkeit von Abstand bis $R=400$.

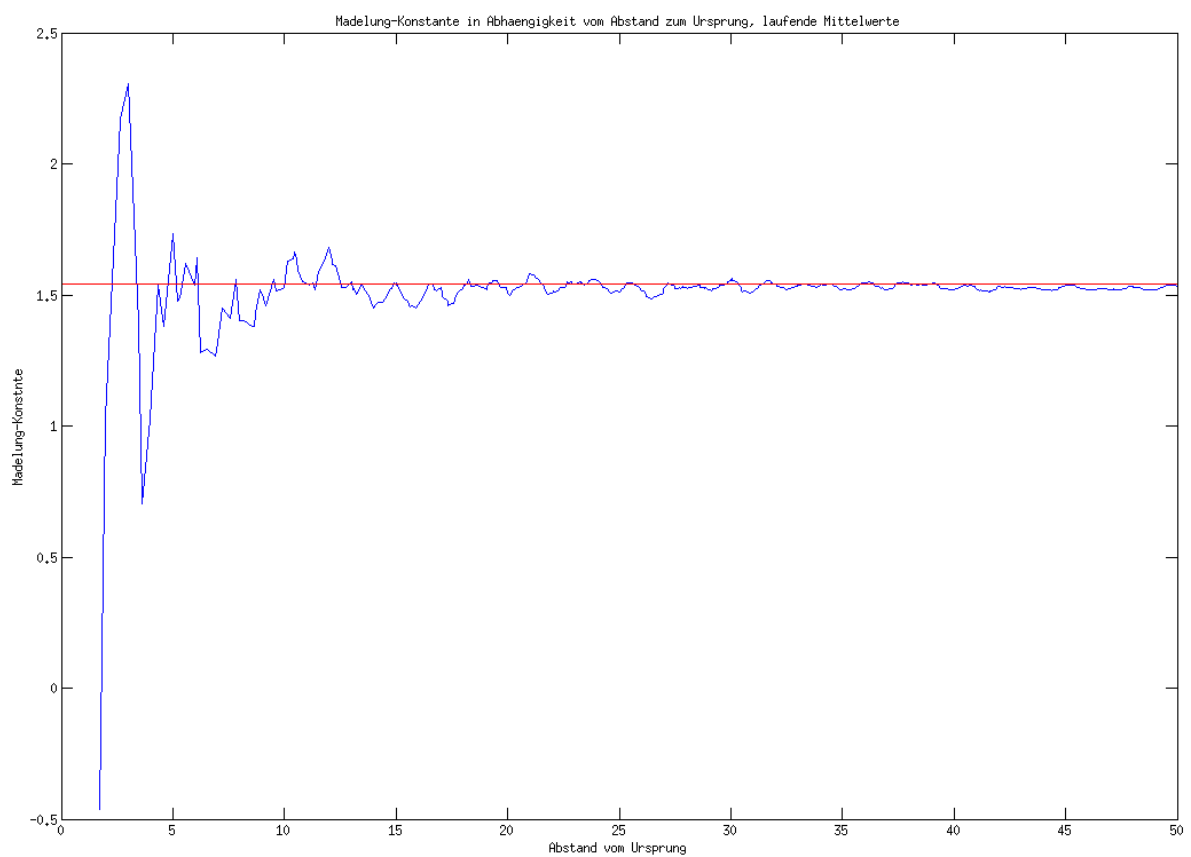


Abbildung 12: Laufende Mittelwerte das Madelung-Konstante bis $R=50$

Die Graphiken sehen genau so verrauscht aus wie die des NaCl-Gitters. Auch hier kann man Genauigkeiten von weniger als 0.1 nur durch Mittelwertbildung erhalten. Man erkennt auch hier, dass die Mittelwertbildung zu wesentlich schnellerer Konvergenz führt. (Siehe Abbildung 12) Um das Optimum aus diesem Programm herauszuholen, betrachtete ich wieder die Mittelwerte bis zu einem Abstand von $R_{\max}=400$. (Siehe Abbildung 13)

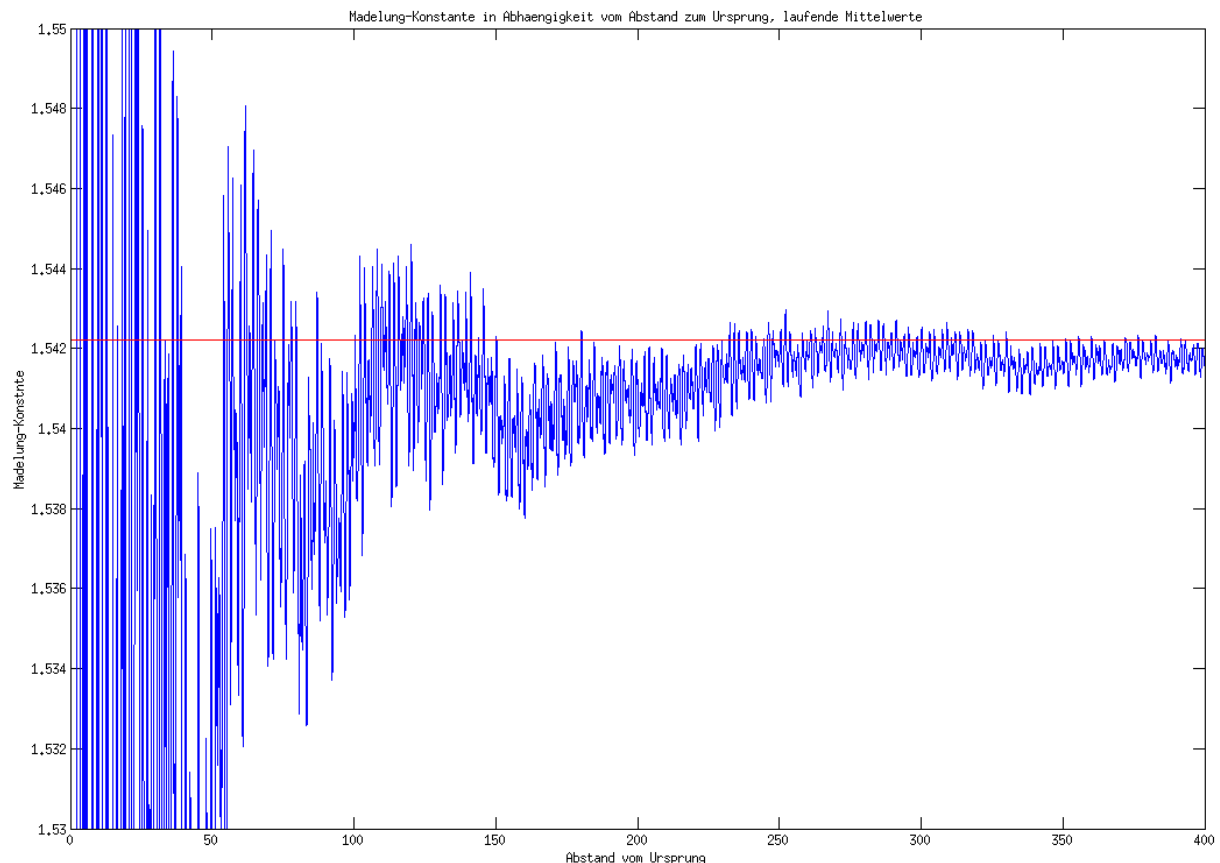


Abbildung 13: Laufende Mittelwerte der Madelung-Konstante bis $R=400$

Daraus kann die Madelung-Konstante, wieder mit einer Ungenauigkeit von 0.001, zu 1.542 ± 0.001 geschätzt werden. Sie ist somit kleiner als die des zweidimensionalen NaCl-Gitters.

Berechnung über ladungsneutrale Sechsecke

Programmcode: `hexagonalmadelungconstant2.c`

Da die Berechnung der Madelung-Konstante nur für ganzzahlige Abstände zu keinem bedeutsamen Erkenntnisgewinn führte, wurde bei dieser Kristallstruktur darauf verzichtet.

Stattdessen ging ich gleich dazu über, statt eines Kreises ein ladungsneutrales Vieleck zu vergrößern. Hier verwendete ich wegen der Symmetrie ein Sechseck, dessen Atome in Inneren den Symmetriefaktor 1 erhalten, die Atome an den Seiten $1/2$ und die an den Ecken $1/3$. Eine negative Ladung sitzt im Ursprung, die Eckpunkte des Sechseckes wurden bei sechs weiteren negativen Ladungen gewählt (siehe Abb. 14).

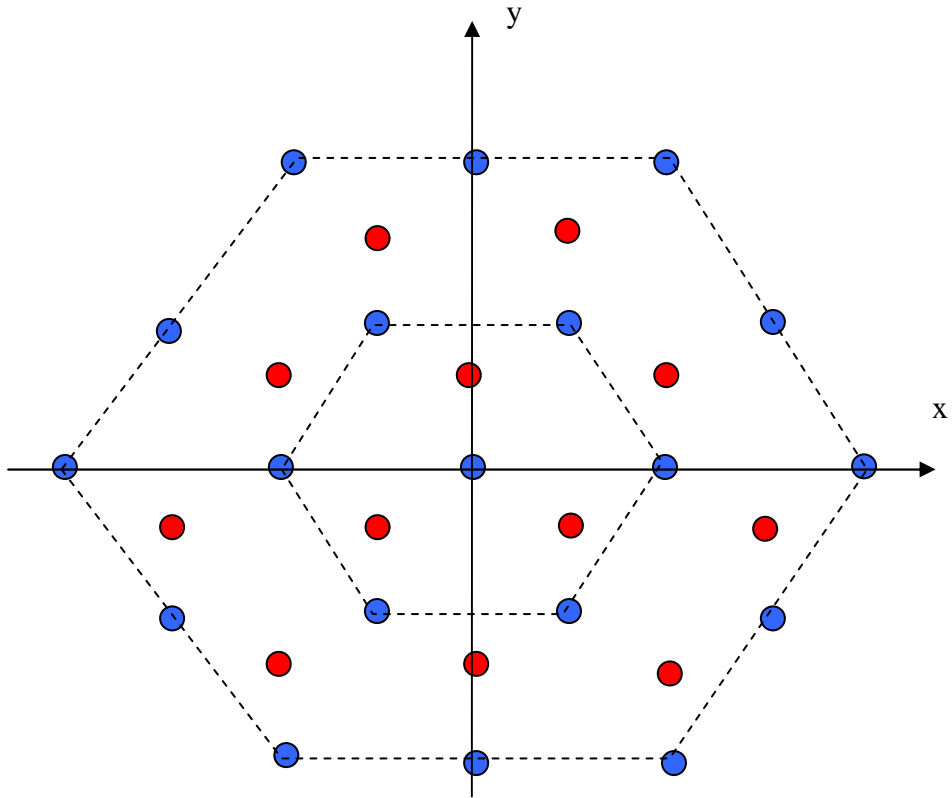


Abbildung 14: Bei der Berechnung verwendete Sechsecke

Dadurch ergibt sich:

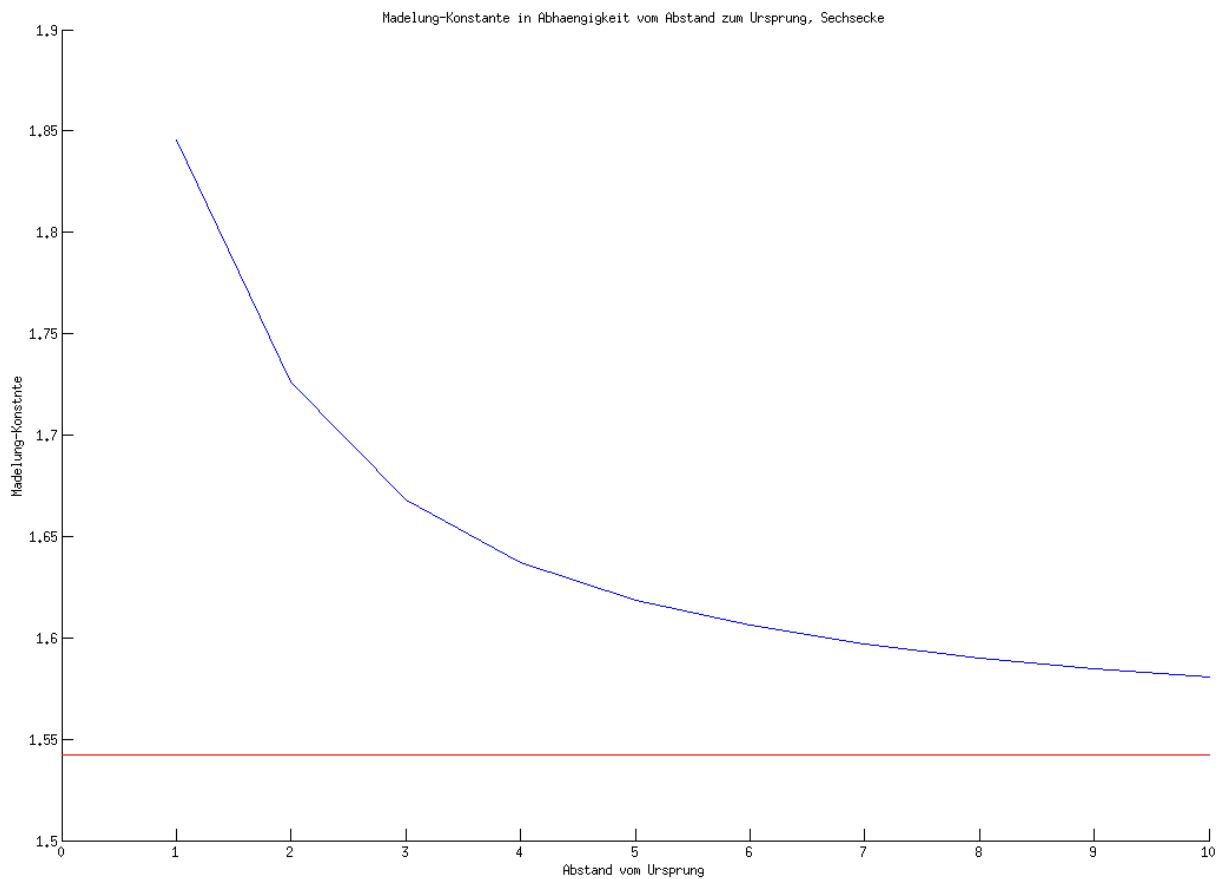
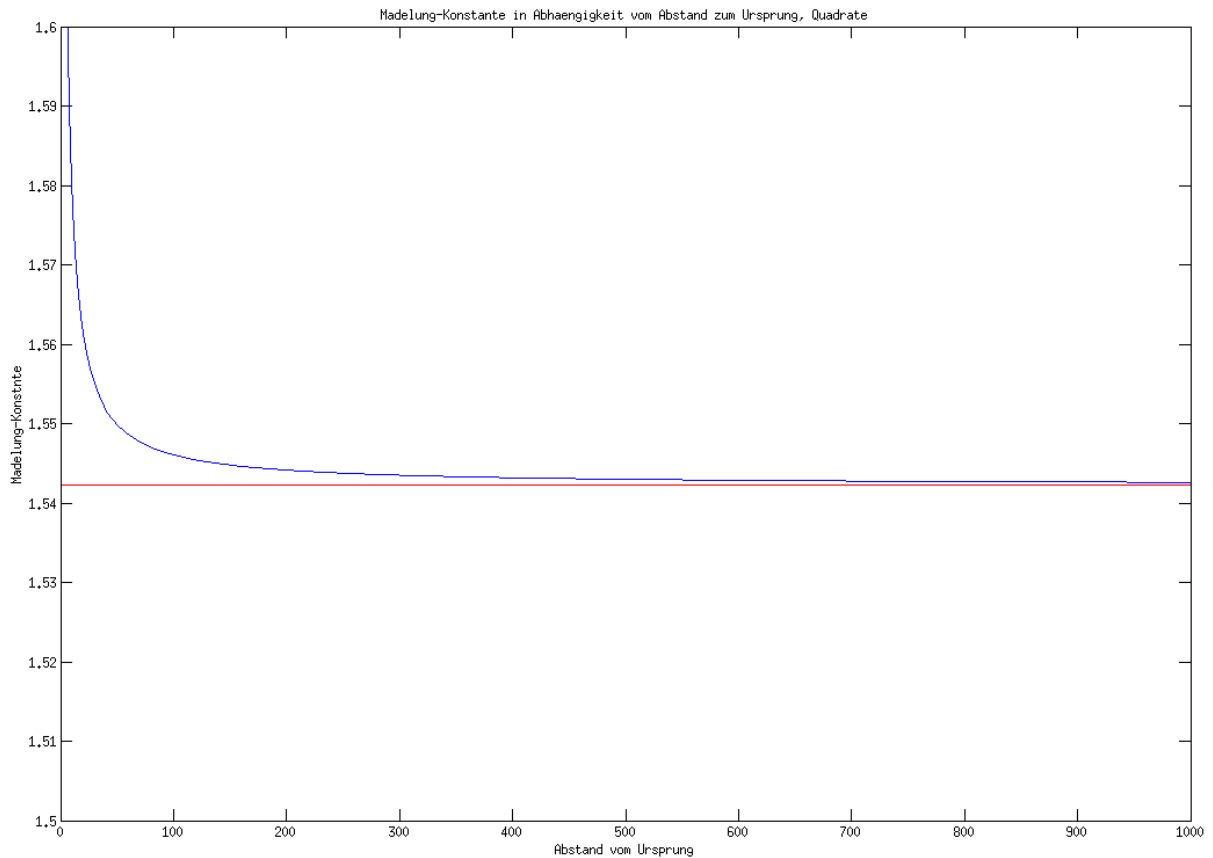


Abbildung 15: Madelung-Konstante für Quadrate mit Seitenlängen bis $2 \cdot 10 = 20$.

Die Konvergenz ist bei Weitem nicht so schnell wie bei den Quadraten der ersten Teilaufgabe. Zu einer einigermaßen genauen Darstellung muss man daher zu wesentlich größeren Abständen, hier 1000, übergehen.



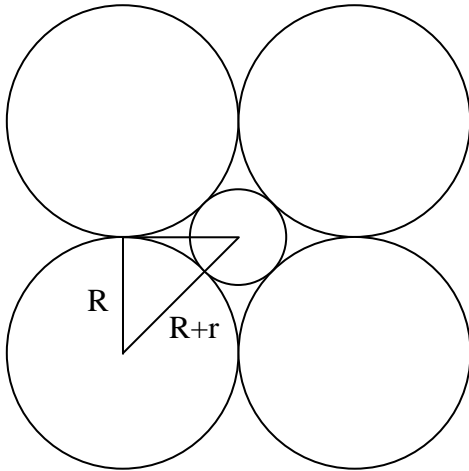
Um eine endgültige genaue Abschätzung zu erhalten, wurden wieder einige Werte bei großen Abständen berechnet:

R=1000	M=1.54260462181833291
R=5000	M=1.54229670174237521
R=10000	M=1.54225821172441568
R=20000	M=1.54223896671556382
R=30000	M=1.54223255171188600
R=40000	M=1.54222934421085345
R=50000	M=1.54222741970947697

Da es sich hier um obere Grenzen handelt (Reihe ist monoton fallend), schätze ich die Madelung-Konstante zu 1.54222 ± 0.00001 . Da ich in der Literatur keine Vergleichswerte fand, nahm ich diesen Wert auch in allen Graphiken als exakten Wert an, den ich als rote Linie eingezeichnet habe. Die Ungenauigkeit von $10E-5$ liegt in allen Graphen innerhalb der Strichdicke.

Kritische Radien

2D-NaCl-Struktur

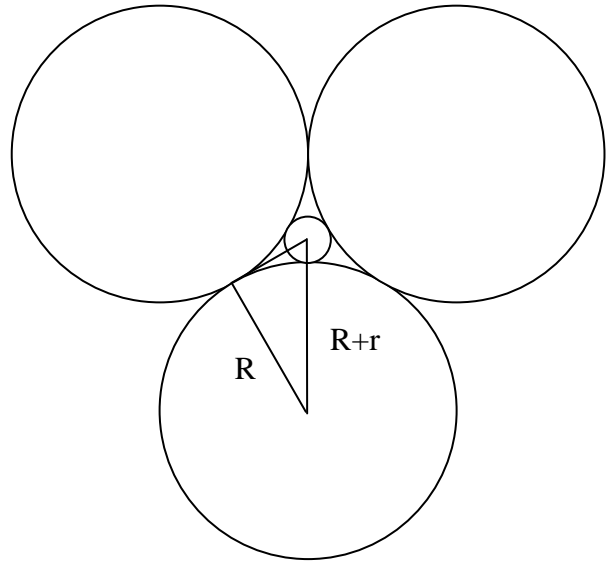


$$\sin(45^\circ) = \frac{R}{R+r} = \frac{\sqrt{2}}{2}$$

$$R+r = R \cdot \sqrt{2}$$

$$\frac{r}{R} = \sqrt{2} - 1 \approx 0.4142$$

hexagonale Struktur



$$\sin(60^\circ) = \frac{R}{R+r} = \frac{\sqrt{3}}{2}$$

$$R+r = R \cdot \frac{2}{\sqrt{3}}$$

$$\frac{r}{R} = \frac{2}{\sqrt{3}} - 1 \approx 0.1547$$

Die zweidimensionale NaCl Struktur ist somit zwar aufgrund der höheren Madelung-Konstante energetisch günstiger, aber hypothetische 2D-Atome könnten bei sehr unterschiedlichen Atomradien ($r/R < 0.4142$) nur in der zweiten Struktur kristallisieren.

Programme

madelungconstantNaCl1.c

```
#include <stdio.h>
#include <math.h>

int main(){

int rmax,delete1,delete2,i,j,k,x,y,sign,symmetryfactor;
double rnew,r,anew;

FILE *fout;
fout=fopen("madelungsconstant.output","w");

FILE *fout2;
fout2=fopen("madaverage.output","w");

printf("rmax eingeben:\n");
scanf("%d",&rmax);

double rxy[rmax+1][rmax+2];
double R_1[rmax*rmax];
double M[rmax*rmax];

R_1[1]=1.;
M[1]=0.;

//suche zuerst die Beiträge der einzelnen Schalen

while (k<=rmax*rmax){
    rnew=10e8;
    for (j=1;j<=rmax;j++){
        for (i=1;i<=j+1;i++){
            x=j;
            y=i-1;
            r=sqrt(x*x+y*y);
            if ((R_1[k-1]<r)&&(r<=rnew)){
                if (r<rnew){
                    rnew=r;
                    M[k]=0;
                    R_1[k]=rnew;
                }

                if (((x+y)%2)==1){
                    sign=1;
                }
                else{
                    sign=-1;
                }

                if (x==0||y==0||x==y)
                    {symmetryfactor=4;}
                else
                    {symmetryfactor=8;}

                anew=sign*symmetryfactor/r;
            }
        }
    }
}
```



```

        M[k]=M[k]+anew;

    }

}

}
if(R_1[k]>=rmax){
    break;
}

printf("%d\n",k);

k=k+1;
}

double R[k+1];
double madelungsconstant[k+1];
double madaverage[k+1];
int count;

//nun werden die Beiträge der einzelnen Schalen zu Madelung-Konstante
aufsummiert:

printf("madelungsconstant\n\n");

for (j=1;j<=k;j++){
    R[j]=R_1[j];
    madelungsconstant[j]=0;
    for (i=1;i<=j;i++){
        madelungsconstant[j]+=M[i];
    }
    printf("%d\n",j);
    fprintf(fout,"%lf    %lf\n",R[j],madelungsconstant[j]);
}

//Mittelwertbildung:

printf("madelungsconstant average\n\n");

for (j=1;j<=k;j++){
    madaverage[j]=0;
    count=0;
    for (i=(j*5)/10+1;i<=j;i++){
        count+=1;
        madaverage[j]+=madelungsconstant[i];
    }
    madaverage[j]=madaverage[j]/count;
    printf("%d\n",j);
    fprintf(fout2,"%lf    %lf\n",R[j],madaverage[j]);
}
printf("Fahne8 -> End\n");
printf ("mandelungsconstant=%lf    %d\n",madaverage[j-1],j);

return 0;
}

```

madleungconstantNaCl2.c

```
#include <stdio.h>
#include <math.h>

int main(){
    int Rmax,Rmin,rmax,x,y,sign,symmetryfactor,deltarmax;
    double r,madelungsconstant,sum,madaverage;

    FILE *fout;
    fout=fopen("madelungsconstant3.output","w");

    printf("Rmax eingeben:\n");
    scanf("%d",&Rmax);
    printf("Rmin eingeben:\n");
    scanf("%d",&Rmin);
    deltarmax=1;

    /*int R[Rmax+1];
    double Mandelungsconstant[Rmax+1];*/

    sum=0;
    madaverage=0;

    for(rmax=Rmin;rmax<=Rmax;rmax+=deltarmax){

        madelungsconstant=0;
        for(x=1;x<=rmax;x++){
            for(y=0;(sqrt(x*x+y*y)<=rmax)&&(y<=x);y++){
                r=sqrt(x*x+y*y);
                if((x+y)%2==1){
                    sign=1;
                }
                else{
                    sign=-1;
                }
                if(x==0||y==0||x==y)
                    {symmetryfactor=4;}
                else
                    {symmetryfactor=8;}
                madelungsconstant+=symmetryfactor*sign/r;
            }
        }
        sum+=madelungsconstant;
        madaverage=sum/(rmax-Rmin+1);
        printf("rmax= %d madelungsconstant=%lf\n",rmax,madelungsconstant);
        fprintf(fout,"%d %lf %lf\n",rmax,madelungsconstant,madaverage);
    }

    return 0;

}
```

madelungconstantNaCl3.c

```
#include <stdio.h>
#include <math.h>

int main(){
    int Rmax,R,x,y,Rmin;
    int symmetryfactor,sign;
    double madelungsconstant,r;

    FILE *fout;
    fout=fopen("madelungsconstant4.output","w");

    printf("Rmax eingeben:\n");
    scanf("%d",&Rmax);
    printf("Rmin eingeben:\n");
    scanf("%d",&Rmin);

    for(R=Rmin;R<=Rmax;R++){
        madelungsconstant=0;
        for(x=1;x<=R;x++){
            for(y=0;y<=x;y++){
                r=sqrt(x*x+y*y);

                if (((x+y)%2)==1){
                    sign=1;
                }
                else{
                    sign=-1;
                }

                if (x==0 || y==0 || x==y)
                    {symmetryfactor=4;}
                else
                    {symmetryfactor=8;}
                if(x==R)
                    {symmetryfactor=symmetryfactor/2;}
                if(y==R)
                    {symmetryfactor=symmetryfactor/2;}
                madelungsconstant+=sign*symmetryfactor*1./r;
            }
        }
        printf("Schleifenende: %d %20.17f\n",R,madelungsconstant);
        fprintf(fout,"%d %20.17f\n",R,madelungsconstant);
    }
    printf("Rbest=%d madelungsconstant=%20.17f\n",R,madelungsconstant);
    printf("fertig!");
    return 0;
}
```

hexagonalmadelungconstan1.c

```
#include <stdio.h>
#include <math.h>

int main(){

    int rmax,d,i,j,l,imax,k,sign,symmetryfactor,count;
```

```

double x,y,r,rnew,a, Mnew;
double gen=10e-20;

FILE *fout;
fout=fopen("hexagonal.output","w");

FILE *fout2;
fout2=fopen("hexaverage.output","w");

printf("rmax eingeben:\n");
scanf("%d",&rmax);

double R_1[rmax*rmax];
double M[rmax*rmax];
R_1[0]=1;

//suche zuerst die Beiträge der einzelnen Schalen

for(k=0;;){

    rnew=10e8;
    for (j=1;j<=rmax*4./3.+4;j++){

        //je nach Reihennummer sind die y-Werte zu berechnen:
        switch(j%4){
            case 0: y=j/4*3;
                    imax=j/4*3;
                    break;
            case 1: y=j/4*3+1;
                    imax=j/4*3+1;
                    break;
            case 2: y=j/4*3+1.5;
                    imax=j/4*3+1;
                    break;
            case 3: y=j/4*3+2.5;
                    imax=j/4*3+2;
                    break;
        }
        for(i=0;i<=imax;i++){

            // Grenzen wegen Symmetrie!!!

            switch(j%4){
                case 0: x=i*sqrt(3);
                        break;
                case 1: x=i*sqrt(3);
                        break;
                case 2: x=(i+0.5)*sqrt(3);
                        break;
                case 3: x=(i+0.5)*sqrt(3);
                        break;
            }
            r=sqrt(x*x+y*y);
            if(r>rmax+gen){
                break;
            }

            if ((R_1[k-1]<r)&&(r<=rnew+gen)){
                if (rnew-r>gen){
                    rnew=r;
                    M[k]=0;
                    R_1[k]=rnew;
                }
            }
        }
    }
}

```

```

        if(j%2){
            sign=1;
        }
        else{
            sign=-1;
        }
        if (x<gen||i==imax)
            {symmetryfactor=3;}
        else
            {symmetryfactor=6;}

        Mnew=sign*symmetryfactor/r;
        M[k]=M[k]+Mnew;
    }
}
}
if(R_1[k]>=rmax){
    break;
}
k=k+1;
}

double R[k+1];
double madelungsconstant[k+1];
double madaverage[k+1];

// nun werden die Beiträge der einzelnen Schalen zur Madlung-Konstante
aufsummiert

printf("madelungsconstant\n\n");

for (j=0;j<=k;j++){
    R[j]=R_1[j];
    madelungsconstant[j]=0;
    for (l=0;l<=j;l++){
        madelungsconstant[j]+=M[l];
    }
    fprintf(fout, "%lf    %lf\n",R[j],madelungsconstant[j]);
}

printf("madelungsconstant average\n\n");

//Mittelwertbildung:

for (j=0;j<=k;j++){
    madaverage[j]=0;
    count=0;
    for (l=(j*5)/10+1;l<=j;l++){
        count+=1;
        madaverage[j]+=madelungsconstant[l];
    }
    madaverage[j]=madaverage[j]/count;
    fprintf(fout2, "%lf    %lf\n",R[j],madaverage[j]);
}
printf("Fahne8 -> End\n");
printf ("mandelungsconstant=%lf    %d\n",madaverage[j-1],j);

return 0;
}

```

hexagonalmadelungconstant2.c

```
#include <stdio.h>
#include <math.h>

int main(){
    int Nmax,Nmin,N,i,j,Rbest;
    int sign;
    double
madelungsconstant ,rn ,rp ,gen ,xp ,yp ,xn ,yn ,symmetryfactorn ,symmetryfactorp;

    FILE *fout;
    fout=fopen("hexagonal2.output","w");
    printf("Nmin eingeben:\n");
    scanf("%d",&Nmin);
    printf("Nmax eingeben:\n");
    scanf("%d",&Nmax);

    for(N=Nmin;N<=Nmax;N++){
        madelungsconstant=0;
        for(i=0;i<=N;i++){
            for(j=0;j<=N;j++){

                xn=i*sqrt(3)-sqrt(3)/2.*j;
                yn=1.5*j;
                rn=sqrt(xn*xn+yn*yn);

                xp=xn-sqrt(3)/2.;
                yp=yn+0.5;
                rp=sqrt(xp*xp+yp*yp);

                symmetryfactorn=3;
                if (i==0||j==0||i==N||j==N)
                    {symmetryfactorn=3./2.;
                    if(i==0&&j==0)
                        {symmetryfactorn=0.;}
                    }
                if((i==N && j==0) || (j==N && i==0))
                    {symmetryfactorn=1./2.;}
                if (i==N && j==N)
                    {symmetryfactorn=1.;}

                symmetryfactorp=3.;
                if(j==N||i==0)
                    {symmetryfactorp=0.;}

                if(i!=0||j!=0){
                    madelungsconstant+=symmetryfactorp*1./rp-
symmetryfactorn*1./rn;
                }
            }
        }

        printf("Schleifenende: %d %20.17f\n",N,madelungsconstant);
        fprintf(fout,"%d %lf\n",N,madelungsconstant);
    }

    printf("fertig!");
    return 0;
}
```