

NUMERISCHE METHODEN IN DER PHYSIK

Erste Übung WS 2012/2013 C

Numerische Auswertung wichtiger Integrale aus der Thermodynamik kristalliner Festkörper mittels Gauss-Integration und kubischen Splines

Darstellung kubischer Splines: Skriptum Kap. 3.2.1 - 3.2.4
Darstellung der Gauss-Integration: Skriptum Kap. 6.7

Von mir zur Verfügung gestellte Programme:

SPLINE_TEST.C

SPLINE.C

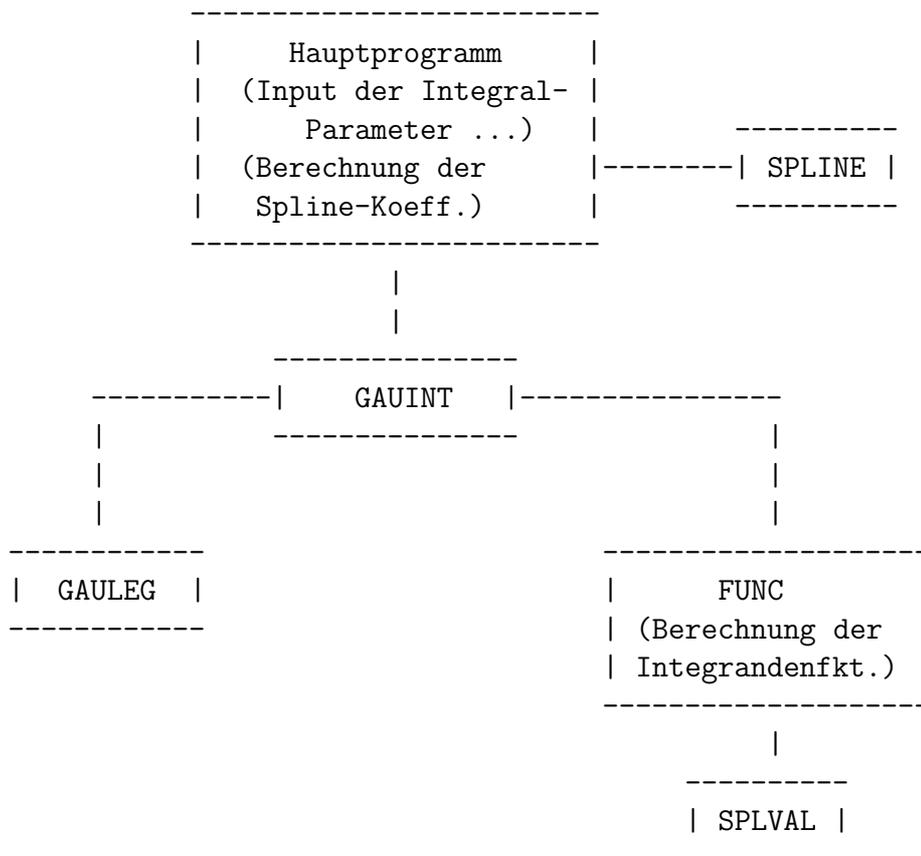
SPLVAL.C

GAUINT_TEST.C

GAUINT.C

GAULEG.C

Struktur des Gesamtprogrammes:



Die Programme "Hauptprogramm" und "FCT" sind Ihr Problem, die Programme GAUINT und GAULEG sowie SPLINE und SPLVAL stelle ich zur Verfügung.

- Diese erste Übung dient neben der Lösung eines physikalisch interessanten Problems auch als *Einstiegsproblem* für diejenigen unter Ihnen, die noch keine Erfahrung mit einer höheren Programmiersprache wie C haben. Aus diesem Grund finden Sie auf dem File GAUINT_TEST.C das vollständige Listing eines Testprogramms, das ein konkretes Integrationsproblem mittels der *Gauss-Legendre-Quadratur* numerisch löst.
- Arbeiten Sie dieses Listing sorgfältig ("Anweisung für Anweisung") durch: zusammen mit einem Lehrbuch über C sollten Sie rasch verstehen, wie man ein solches Programm aufbaut.
- Obwohl ich im Rahmen dieser Übungen keinen C-Kurs anbieten kann, stehe ich natürlich (insbesondere am Anfang der Übungen) für alle Programmierfragen zur Verfügung.

Programmstruktur gauint_test in C

```

/*****
PROGRAM gauint_test

Testprogramm fuer eine numerische Gauss-Legendre-Integration.

H. Sormann 31-8-2000
Last Update 28-8-2005
*/

#include <stdio.h>           // Bibliothek fuer Input/Output etc.
#include <math.h>           // Math-Bibliothek

#include "nrutil.c"         // Software for array manipulation
                           // (aus den Numerical Recipes C (1994))

int fcalls;                // globale Variable

/*****/

double func(double x)
//      Funktionsroutine zur Berechnung der Integrandenfunktion
//      fuer das Testintegral.
{
    double wert;

    fcalls++;

```

```

wert=log(1.0+x)/x/(1.0+x);

return(wert);
}

/*****

#include "gauleg.c"      // Einbeziehung des Programms GAULEG.C
                        // (Berechnung der Gauss-Legendre-Parameter)
                        // (aus den Numerical Recipes C (1994))

/*****

void gauint(double c, double d,int irel, double gen, int manf, int minkr,
            int mmax, double *s, double *error, int *fehler,int itest)
/*
  Programm zur naeherungsweise Berechnung eines bestimmten Integrals
  mittels der Quadraturformel Gauss-Legendre. Die erforderlichen Parameter
  werden von der Routine GAULEG berechnet.

  Die Parameter von GAUINT werden im Abschnitt 6.7.3 des Skriptums genau
  beschrieben, bis auf den letzten (zusaetzlichen) Parameter itest ,
  der die folgende Bedeutung hat:

      itest = 1   GAUINT gibt Zwischenwerte der Integration aus
                  (fuer die Testphase)
      itest ungleich 1   kein Testoutput

  reelle Daten in doppelter Genauigkeit (DOUBLE)!

  Last update:  4-9-2000
*/
{
  int m,j;
  double olds, *w, *z;

  *fehler=1;

  if(manf<1 || minkr<1) {printf("MANF oder MINKR kleiner 1\n"); return;}
  if(manf>mmax) {printf("Anfangs-Ordnung groesser MMAX\n"); return;}

  m=manf;
  olds=1.e30;

```

```

// Fuer Testzwecke:
if(itest==1) {
    if(irel==1) printf("    M    ERGEB          ERR-NUM          relGEN\n");
    else      printf("    M    ERGEB          ERR-NUM          absGEN\n");
}
// *****

do {
    z=dvector(1,m);
    w=dvector(1,m);

    gauleg(c,d,z,w,m);

    *s=0.e0;
    for(j=1;j<=m;j++) *s+=w[j]*func(z[j]);

    free_dvector(z,1,m);
    free_dvector(w,1,m);

    *error=fabs(*s-olds);
    if(irel == 1) {
        *error/=fabs(olds);
    }

// Fuer Testzwecke:
    if(itest==1)
        if(m > manf) printf("%4i    %13.6e    %13.6e    %13.6e\n",
                               m-minkr,olds,*error,gen);
// *****

    if(*error < gen) {*fehler=0; *s=olds;}

    olds=*s;
    m+=minkr;

} while (m<=mmax && *fehler==1);

if(*fehler==1) {
    printf("FEHLER: Rechengenauigkeit bei Ordnung %d nicht erreicht!!\n",
           mmax);
}
}

/*****/

```

```

int main()          // Hauptprogramm
{
    int manf,minkr,mmax,irel,fehler,itest;
    double c,d,exakt,gen,acc,s;

// Integral-Testproblem: Integral-Grenzen und exakter Wert
c=0.e0;
d=1.e0;
exakt=M_PI*M_PI/12.e0 - log(2.e0)*log(2.e0)/2.e0; // M_PI von <math.h>

printf("Input: absolute (0) oder relative (1) Genauigkeit ?\n");
scanf("%d", &irel);
printf("Input: Genauigkeit = ?\n");
scanf("%lf", &gen);

printf("Input: Anfangsordnung der Quadratur = ?\n");
scanf("%d", &manf);
printf("Input: Inkrement der Quadratur-Ordnung = ?\n");
scanf("%d", &minkr);
printf("Input: Testoutput von GAUINT JA/NEIN (1/0)?\n");
scanf("%d", &itest);
mmax=500;          // maximale Quadratur-Ordnung

printf("\n\n");
printf("NUM. METH. IN DER PHYSIK GAUSS-LEGENDRE\n");
printf("INTEGRAL-TESTPROBLEM\n");

fcalls=0;
gauint(c,d,irel,gen,manf,minkr,mmax,&s,&acc,&fehler,itest);

if(fehler==1) printf("Fehler in GAUINT !!!\n");
else {
    printf("\nEXAKT =          %14.10f\n", exakt);
    printf("Ergebnis =      %14.10f\n\n", s);
    if(irel != 1) {
        printf("geforderte Gen. (abs) = %le\n", gen);
        printf("berechnete Gen. (abs) = %le\n", acc);
        printf("wahre Gen.          (abs) = %le\n", fabs(s-exakt));
    }
    else {
        printf("geforderte Gen. (rel) = %le\n", gen);
        printf("berechnete Gen. (rel) = %le\n", acc);
        printf("wahre Gen.          (rel) = %le\n", fabs(s-exakt)/fabs(exakt));
    }
    printf("Dazu waren %d Funktionsaufrufe noetig.\n", fcalls);
}
return (0);
}

```

1. Aufgabe

Austesten des GAUTEST-Programms durch Anwendung auf das bestimmte Integral

$$I = \int_0^1 dx \frac{\ln(1+x)}{x(1+x)} = \frac{\pi^2}{12} - \frac{1}{2}(\ln 2)^2 = 0.5822405265\dots$$

Gefordert ist die Unterschreitung des absoluten Fehlers 10^{-8} , d. h. es gilt:

$$\text{GEN} = 10^{-8}.$$

Für MANF nehmen Sie den Wert 4 und für MINKR den Wert 2.

Testergebnis:

NUM.	METH.	IN DER PHYSIK	GAUSS-LEGENDRE	
INTEGRAL-TESTPROBLEM				
M	ERGEB	ERR-NUM	absGEN	
4	5.822386e-01	1.920632e-06	1.000000e-08	
6	5.822405e-01	1.924312e-09	1.000000e-08	

EXAKT = 0.5822405265

Ergebnis = 0.5822405245

geforderte Gen. (abs) = 1.000000e-08

berechnete Gen. (abs) = 1.924312e-09

wahre Gen. (abs) = 1.926151e-09

Dazu waren 18 Funktionsaufrufe noetig.

Programmstruktur spline_test in C

```

/*****
PROGRAM spline_test

Testprogramm fuer eine numerische Interpolation mittels
kubischer Splines

H. Sormann 31-8-2000
Last Update 22-9-2010
*/

#include <stdio.h>           // Bibliothek fuer Input/Output etc.
#include <math.h>           // Math-Bibliothek

#include "nrutil.c"         // Software for array manipulation
                           // (aus den Numerical Recipes C (1994))

#include "spline.c"
#include "splval.c"

/*****/

int main()    // Hauptprogramm
{
    int    i,fehler;
    double x[6]={0.0, 1.0, 1.6, 1.9, 2.3, 2.7};
    double y[6]={0.0, 0.2, -0.1, -0.6, 0.0, 0.5};
    double *a,*b,*c,*d,xarg,yspline,yd,yydd;

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

    // Ausgabe der 5 Stuetzpunkte:
    printf("\nPROGRAMM SPLINE-TEST (WS 2010/2011)\n\n    X          Y\n");
    for(i=1;i<=5;i++) printf("   %6.3f   %6.3f\n",x[i],y[i]);

    // Berechnung der Spline-Koeffizienten:
    a=dvector(1,5);
    b=dvector(1,5);
    c=dvector(1,5);
    d=dvector(1,5);
    spline(5,x,y,a,b,c,d);

    // Ausgabe der Spline-Koeffizienten:
    printf("\nSPLINE-KOEFFIZIENTEN\n    A          B          C          D\n");
    for(i=1;i<=4;i++) printf("   %8.4f   %8.4f   %8.4f   %8.4f\n",
                               a[i],b[i],c[i],d[i]);
}

```

```
// Berechnung der Interpolationskurve:
for(xarg=1.0;xarg<=2.7;xarg+=0.01) {
    fehler=splval(5,x,a,b,c,d,xarg,&yspline,&yyd,&yydd);
    if(fehler==1) printf("Spline-Fehler bei xarg = %12.5e\n",xarg);
    fprintf(fout," %12.5e    %12.5e\n",xarg,yspline);
}

free_dvector(a,1,5);
free_dvector(b,1,5);
free_dvector(c,1,5);
free_dvector(d,1,5);

return(0);
}
```

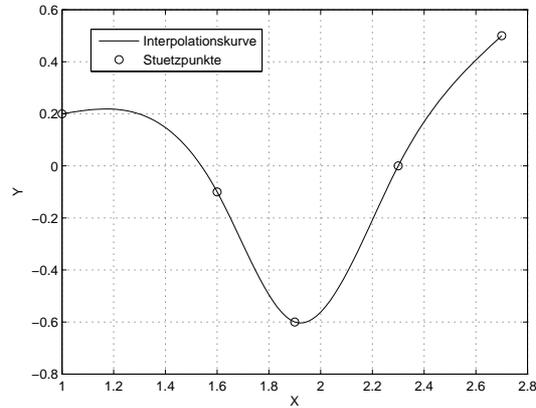


Figure 1: Demonstration einer Spline-Interpolation (Aufgabe 2).

2. Aufgabe

Austesten der Programme `SPLINE` und `SPLVAL` durch Anwendung auf das einfache Beispiel VL-Skriptum, Kap.3.2.4:

Beispiel 1: Demonstration einer Interpolation der fünf Stützpunkte

x	y
1.0	0.2
1.6	-0.1
1.9	-0.6
2.3	0.0
2.7	0.5

mittels natürlicher kubischer Splines.

`SPLINE` liefert die folgenden Koeffizienten:

	A	B	C	D
1	0.2000	0.1628	0.0000	-1.8410
2	-0.1000	-1.8256	-3.3139	12.8117
3	-0.6000	-0.3547	8.2167	-8.9497
4	0.0000	1.9228	-2.5229	2.1024

Die entsprechende Interpolationsfunktion $I(x)$ ist in der Abbildung 1 dargestellt.

3. Aufgabe

Numerische Berechnung der Wärmekapazität der Elektronen in Metallen

1. Problemstellung:

Obwohl die theoretischen Grundlagen dieses physikalischen Problems in der statistischen Physik, der Quantentheorie und der Festkörperphysik "beheimatet" sind, also in Fachgebieten, die Sie zum grössten Teil noch vor sich haben, wird es Ihnen keine Schwierigkeiten bereiten, die Basisphysik dieser Übungsaufgabe zu verstehen:

- **erste Aussage:** um die Temperatur eines Körpers zu erhöhen, muss dem Körper Energie zugeführt werden, d.h.: die innere Energie eines Körpers ist eine Funktion seiner (absoluten) Temperatur (in Kelvin):

$$E = E(T). \quad (1)$$

- **zweite Aussage:** diese innere Energie ist die Summe der Energien aller Teilchen, aus welchen der Körper besteht. Da sich ein metallischer Festkörper aus seinem (regelmässigen) Ionengitter und den in diesem Gitter relativ frei beweglichen Elektronen zusammensetzt, wird dessen Thermodynamik durch die *innere Energie des Ionengitters* und durch die *innere Energie der Metallelektronen* bestimmt:

$$E(T) = E_{ion}(T) + E_e(T) \quad (2)$$

In dieser Übung geht es um den elektronischen Anteil der Gesamtenergie. Wegen des regelmässigen (periodischen) Aufbaus des Kristalls genügt es, die Verhältnisse in einer *Einheitszelle* Ω_0 zu untersuchen. Daraus folgt die *Energiedichte*

$$u_e(T) = \frac{E_e(T)}{\Omega_0} = \frac{1}{\Omega_0} \int_{-\infty}^{+\infty} d\epsilon \epsilon N(\epsilon) f(\epsilon, T). \quad (3)$$

Diese Formel für die innere Energiedichte eines Elektronengases in einem kristallinen Festkörper ist nicht schwer zu verstehen:

ϵ ist die Energie irgendeines Elektronenzustandes, die Funktion $N(\epsilon)$ ist die Dichte der quantenmechanisch zur Verfügung stehenden Elektronenzustände, und $f(\epsilon, T)$ gibt an, mit welcher statistischen Wahrscheinlichkeit ein Quantenzustand der Energie ϵ bei einer absoluten Temperatur T von einem Elektron besetzt wird: da es sich bei den Elektronen um *fermionische* Elementarteilchen handelt, gilt die Ihnen sicher bereits bekannte *Fermi-Dirac*-Statistik mit der Verteilungsformel

$$f(\epsilon, T) = \frac{1}{e^{[\epsilon - \mu(T)]/k_B T} + 1} \quad (4)$$

wobei die Funktion $\mu(T)$ das *chemische Potential* des Elektronengases darstellt.

Summiert (bzw. integriert) man im Sinne von Gl. (3) über den gesamten Energieraum, erhält man die Gesamtenergie E_e der Metallelektronen im Volumen Ω_0 (bzw. deren Energiedichte) bei der Temperatur T

$$u_e(T) = \frac{1}{\Omega_0} \int_{-\infty}^{+\infty} d\epsilon \epsilon N(\epsilon) \frac{1}{e^{[\epsilon - \mu(T)]/k_B T} + 1}. \quad (5)$$

Wichtiger als diese Gesamtenergie ist für die Thermodynamik des Festkörpers die Angabe darüber, wie sich die innere Energie unter Variation von T verändert. Die entsprechende Grösse

$$c_v(T) = \frac{\partial}{\partial T} u_e(T) \quad (6)$$

nennt man die *Wärmekapazität (heat capacity)* pro Volumeneinheit der Elektronen im Festkörper mit der Dimension $J/(m^3 K)$.

Eine wichtige Grösse, die zur Auswertung von Gl. (5) erforderlich ist, ist die schon erwähnte *Zustandsdichte (density-of-states DOS)* des Elektronengases. Diese materialabhängige Funktion, welche die Dimension einer inversen Energie (meist $1/eV$) hat, kann mit Hilfe moderner *Bandstrukturprogramme* sehr genau berechnet werden. Alle DOS-Funktionen, die Sie für diese Übung benötigen, stehen zum Downloaden zur Verfügung.

Bevor über die numerische Berechnung von $c_v(T)$ gesprochen wird, folgt noch eine kurze Wiederholung der Fermi-Dirac-Formel:

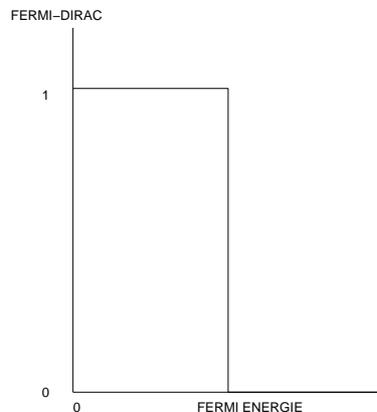
Um diese Formel zu verstehen, geht man am besten vom absoluten Temperatur-Nullpunkt

$$f(\epsilon, 0) = \lim_{T \rightarrow 0} \frac{1}{e^{[\epsilon - \mu(T)]/k_B T} + 1} = \Theta(\mu(0) - \epsilon) \quad (7)$$

aus, wobei Θ die *Stufenfunktion (Heaviside function)* mit der Eigenschaft

$$\Theta(x) = \begin{cases} 0 & \text{für } x < 0 \\ 1 & \text{für } x > 0 \end{cases}$$

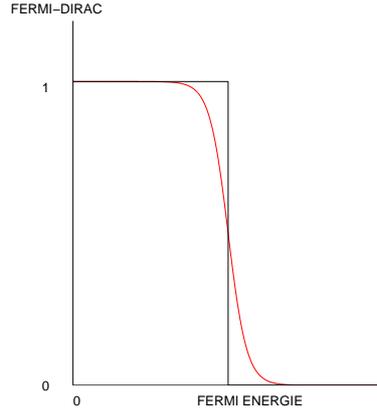
ist. Somit ergibt sich Gl. (7) in der Form



Am absoluten Nullpunkt sind alle quantenmechanisch möglichen Energiezustände bis zur Energie $\mu(0)$ sicher besetzt (Besetzungswahrscheinlichkeit = 1), und ab dieser Energie sind alle Zustände sicher unbesetzt (Besetzungswahrscheinlichkeit = 0). Die Energie, bei welcher der Sprung von 1 auf 0 (Fermi-Kante) erfolgt, heisst die *Fermienergie*:

$$\mu(0) = \epsilon_F . \quad (8)$$

Erhöht man nun die Temperatur, kommt es durch thermische Anregung der Elektronen zu einer "Abschmierung" der Fermi-Kante:



Eine noch nicht diskutierte Frage ist die Abhängigkeit des chemischen Potentials von der Temperatur. Im Mittelpunkt dieser Diskussion steht eine Gleichung, welche der Gl. (5) sehr ähnlich ist, bis auf einen bedeutsamen Unterschied:

$$Z_0 = \int_{-\infty}^{+\infty} d\epsilon N(\epsilon) \frac{1}{e^{[\epsilon - \mu(T)]/k_B T} + 1} . \quad (9)$$

Was bedeutet das Integral auf der rechten Seite dieser Gleichung? Es handelt sich offenbar um die Integration über alle - bei einer bestimmten Temperatur - besetzten Energiewerte, und der Wert dieses Integrals muss natürlich die (bekannte) *Zahl der Valenzelektronen in der Einheitszelle Ω_0 sein!*

D.h., die Gl. (9) kann als *implizite Bestimmungsgleichung* für die Funktion $\mu(T)$ angesehen werden. Für jede gewählte Temperatur T muss der Wert von μ genauso lauten, dass das Integral in Gl. (9) exakt die Zahl Z_0 ergibt.

Wurde das chemische Potential des Elektronengases $\mu(T)$ auf diese Weise bestimmt, sind im Prinzip alle Grössen bekannt, die man zur numerischen Auswertung der Gleichungen (5) und (6) benötigt:

$$c_v(T) = \frac{1}{\Omega_0} \int_{-\infty}^{+\infty} d\epsilon \epsilon N(\epsilon) \frac{\partial}{\partial T} \left\{ \frac{1}{e^{[\epsilon - \mu(T)]/k_B T} + 1} \right\} . \quad (10)$$

Für die konkreten Rechnungen wird aus praktischen Gründen die Energieskala so gewählt, dass die Fermienergie den Wert Null hat. Diese Skalenschiebung hat keinen Einfluss auf das Ergebnis $c_v(T)$, und die aus Glg.

(9) resultierenden Werte $\mu(T)$ bedeuten dann einfach die Abweichungen von der Fermienergie.

Eine analytische Durchführung der Ableitung nach T in Gl. (10) führt zum Resultat

$$c_v(T) = -\frac{1}{\Omega_0} \int_{-\infty}^{+\infty} d\epsilon \epsilon N(\epsilon) \frac{\exp[x(T)]}{\{\exp[x(T)] + 1\}^2} \frac{dx(T)}{dT} \quad (11)$$

mit

$$x \equiv \frac{\epsilon - \mu(T)}{k_B T}, \quad (12)$$

$$\frac{dx(T)}{dT} = [-\mu'(T)T - \epsilon + \mu(T)] / [k_B T^2] \quad (13)$$

und

$$\mu'(T) = \frac{d\mu(T)}{dT}. \quad (14)$$

Das Folgende ohne Beweis:¹ Für nicht zu grosse Temperaturen und für Metalle mit ausgeprägt kräftefreien Valenzelektronen (Alkalimetalle, Aluminium, Magnesium, Kupfer?,...) hat A. Sommerfeld schon in den 20er-Jahren des vorigen Jahrhunderts einen sehr einfachen Ausdruck für $c_v(T)$ angegeben, nämlich den linearen Zusammenhang

$$c_v(T) = \frac{\pi^2}{3\Omega_0} N(0) k_B^2 T. \quad (15)$$

Beachten Sie, dass wegen der oben erfolgten Verschiebung der Energie-Skala $\epsilon = 0$ die Fermi-Energie ϵ_F bedeutet: $N(0)$ bedeutet also *die Zustandsdichte der Elektronen an der "Fermi-Kante"*.

2. Die numerische Auswertung von $c_v(T)$

soll nun für die Metalle Natrium, Kupfer und Platin durchgeführt werden. Dabei kann für den interessierenden Temperaturbereich eine wesentliche Vereinfachung der obigen Formeln erfolgen: Testrechnungen haben gezeigt, *dass die Temperaturabhängigkeit des chemischen Potentials $\mu(T)$ vernachlässigbar klein ist!*

Das bedeutet, dass in den Gleichungen (11)-(14) die folgenden Näherungen gelten:

$$\mu(T) \approx \mu(0) = 0 \quad \text{und} \quad \mu'(T) \approx 0. \quad (16)$$

Unter Berücksichtigung dieser Approximationen erhält man nach kurzer Rechnung

$$c_v(T) = \frac{1}{\Omega_0 k_B T^2} \int_{-\infty}^{+\infty} d\epsilon \epsilon^2 N(\epsilon) \frac{\exp(\epsilon/k_B T)}{[\exp(\epsilon/k_B T) + 1]^2}. \quad (17)$$

¹Eine genaue Darstellung dieser Sommerfeld-Näherung finden Sie z.B. in meinem Skriptum "Theoretische Festkörperphysik" in den Abschnitten 2.6ff.

Ihre Aufgabe besteht nun darin, die Gleichungen (15) und (17) per Computer für Na, Cu und Pt auszuwerten und zu diskutieren.

3. Das leidige Einheitenproblem:

Es ist zwar keine grosse Sache in der Physik, aber immer wieder ein Nervenproblem: "Wie gehe ich mit den Konstanten und Einheiten um, bevor ich die Formeln programmiere?"

Ich möchte Ihnen für diese Übung, bei der es für Sie ja hauptsächlich um die numerischen Probleme bzw. um die physikalische Bedeutung der Resultate geht, diese Arbeit abnehmen:

Wenn Sie c_v als *molare* Wärmekapazität in Einheiten [J/mol/K] nehmen, das Volumen Ω_0 der Einheitszelle in [Bohr³], die Energie ϵ in [eV], die Zustandsdichte N in [1/eV], und die Temperatur in [K], lautet die Formel (17)

$$c_v(T) = 1.2545786 \cdot 10^{13} \frac{m}{\rho} \frac{1}{\Omega_0 T^2} \int_{-\infty}^{+\infty} d\epsilon \epsilon^2 N(\epsilon) \frac{\exp(x)}{[\exp(x) + 1]^2} \quad (18)$$

mit

$$x = 1.1604155 \cdot 10^4 \frac{\epsilon}{T},$$

und die Sommerfeld-Näherung (15) hat die Form

$$c_v(T) = 26.41409 \frac{m}{\rho} \frac{N(0)}{\Omega_0} T. \quad (19)$$

In den Gleichungen (18) und (19) bedeutet m die Atommasse des Materials (in Gramm) und ρ ist die Dichte in Einheiten [kg/m³].

4. Die Parameter der Rechnung:

	Ω_0 [Bohr ³]	Zustandsdichte $N(\epsilon)$ N in [1/eV], ϵ in [eV]	m	ρ [kg/m ³]	Schmelz- temperatur [K]
Na	256.2881	dosNEU_Na	22.99	971	371
Cu	79.6810	dosNEU_Cu	63.55	8960	1357
Pt	101.9794	dosNEU_Pt	195.00	21450	2045

Alle Rechnungen sind im Bereich $100 \leq T \leq 4000$ K durchzuführen, mit einem Inkrement von $\Delta T=100$ K.

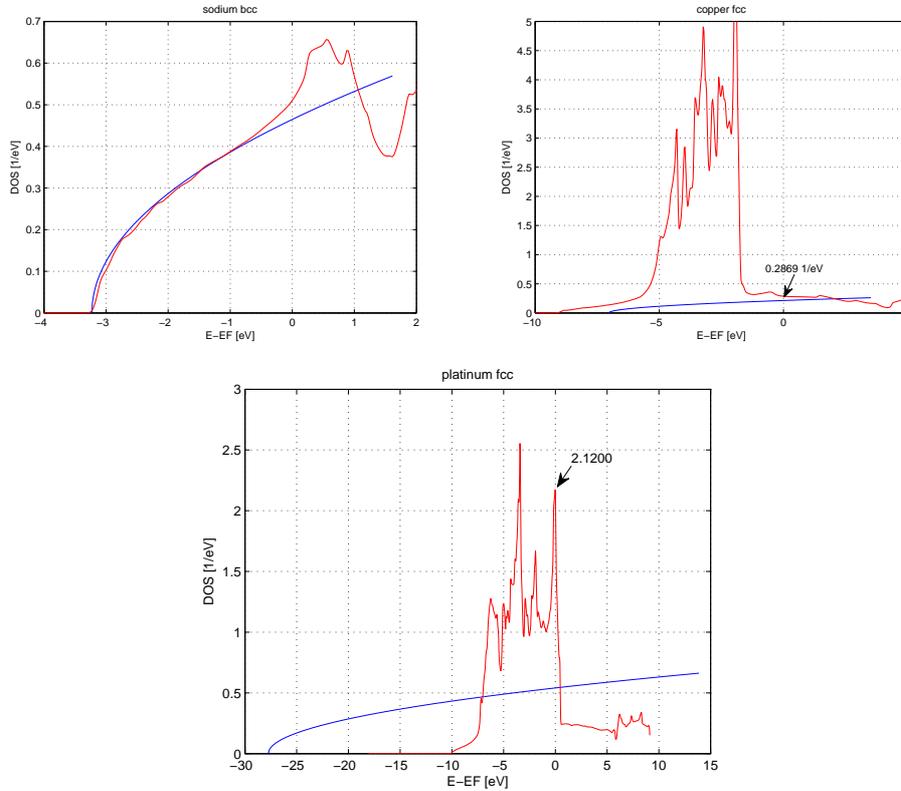
5. Die Zustandsdichten $N(\epsilon)$

für Na, Cu und Pt habe ich mit dem Bandstrukturprogramm WIEN2k² berechnet und unter den oben genannten Filenamen zum Downloaden bereitgestellt. Jeder File gibt in der ersten Zeile die Zahl der Datenpunkte,

²P. Blaha, K. Schwarz, G. Madsen, D. Kvasnicka, J. Luitz, WIEN2k: *An Augmented Plane Wave + Local Orbitals Program*, 2001.

und ab der zweiten Zeile finden Sie in der ersten Spalte die Energie ϵ in [eV] und in der zweiten Spalte die Zustandsdichte N in [1/eV]. $E = 0$ bezeichnet die Fermi-Energie.

In den folgenden Bildern sind die Zustandsdichten der drei Metalle dargestellt:



Diese Diagramme zeigen die DOS der Valenzelektronen in bcc Na (oben links), fcc Cu (oben rechts) und fcc Pt (unten), wobei die roten Kurven die DOS-Ergebnisse aus dem Bandstrukturprogramm WIEN2k darstellen. Die Energieskalen sind so gewählt, dass 0 die Fermienergie bedeutet.

Die blauen Kurven zeigen nun die entsprechenden "Sommerfeld"-Zustandsdichten; dies sind jene DOS-Kurven, welche den einfachen linearen Zusammenhang zwischen der Wärmekapazität c_v und der Temperatur T ergeben, die in Glg. (19) beschrieben ist.

Daraus kann man das Folgende ableiten:

- Beim Natrium ist die Übereinstimmung zwischen der "roten" und der "blauen" DOS-Kurve relativ gut, nur im Bereich der Fermienergie und darüber gibt es Abweichungen. Es ist also zu erwarten, dass die Näherungslösung (19) gut zum exakten Ergebnis (18) passen wird.
- Beim Kupfer scheint auf dem ersten Blick die "blaue" nichts mit der "roten" Kurve gemein zu haben. Dies ist jedoch nicht so: man kann

nämlich zeigen, dass bei der Formel (18) der grösste Beitrag zum Integral aus dem Bereich um die Fermienergie kommt, und dort sind die beiden Kurven relativ nahe beieinander. Es ist also auch beim Cu zu hoffen, dass die "Sommerfeld-Näherung" nicht weit vom exakten Ergebnis entfernt sein wird.

- Beim Platin ist es aber klar: hier hat die "blaue" keinerlei Ähnlichkeit mit der "roten" Kurve, insbesondere auch nicht im kritischen Bereich um die Fermienergie. Man muss bei diesem Material davon ausgehen, dass die "Sommerfeld-Approximation" (19) mit der aus Gl. (18) resultierenden exakten Lösung für die molare Wärmekapazität $c_v(T)$ nur sehr schlecht übereinstimmen wird.

6. Die numerische Auswertung der Glg. (18)

ist natürlich Ihre Sache. Ich möchte daher nur auf ein wichtiges Problem eingehen, nämlich auf die Frage: *Wie geht man bei der Integration in Gl. (18) mit der Funktion $N(\epsilon)$ um?*

Das Problem ist, dass Sie im Computer diese Funktion, die nur als Tabelle vorliegt, für jedes beliebige ϵ parat haben müssen. Die übliche Art, mit einer solchen Situation umzugehen, ist die folgende:

- Man berechnet die Spline-Koeffizienten der Tabellenwerte $N(\epsilon)$ mittels des C-Programmes *spline.c*,
- Im Funktionsunterprogramm FUNC, wo die Integrandenfunktion ausgewertet wird, verwendet man das C-Programm *splval.c*, um für jedes gewünschte ϵ den Wert $N(\epsilon)$ zu erhalten (vgl. das Programmschema am Anfang dieser Übungsbeschreibung).

7. Die geforderten Ergebnisse:

- Berechnen Sie die exakten molaren Wärmekapazitätskurven $c_v(T)$ für die Elektronen in Natrium, Kupfer und Platin durch numerische Auswertung der Formel (18), und vergleichen Sie diese Ergebnisse mit den entsprechenden linearen Näherungen gemäss Gl. (19). Erstellen Sie für jedes Material ein eigenes Diagramm für den Temperaturbereich $100 \leq T \leq 4000$ K.
- Es ist natürlich klar, dass Temperaturen bis zu 4000 K hypothetisch sind, da bei diesen Temperaturen die gegebenen Metalle längst keine Festkörper mehr sind. Markieren Sie daher in Ihren Diagrammen die jeweiligen, in der Parameter-Tabelle angegebenen Schmelztemperaturen.
- Überprüfen Sie insbesondere, ob Ihre exakten c_v -Kurven im Grenzfall kleiner Temperaturen ($T \leq 500$ K) die jeweilige "Sommerfeld-Gerade" annähern.