TPERSP

柳瀬 章

平成 14 年 9 月 24 日

1 はじめに

空間群のプログラム TSPACE には結晶構造の図、フェルミ面の図、ブリルアンゾーンの図、立方・六方調和関数の図の立体透視図が数多く掲載されている。これらの図は TPERSP を用いて描かれている。最近開発した AYPLOT の上で動作するようにこの TPERSP を書き直して、公開できるようになったので、その使用法をここに紹介する。フェルミ面や結晶構造を描くことを目的に開発されているため、ここでの説明も空間群についての引用が多くなる。説明で TSPACE の本と記述が重複する場合は随時それを TSP pxxx のように引用する。

バンド計算は 3 次元結晶の電子状態を計算し、逆格子空間の $\vec{k}$ の関数としてエネルギーを求めたり、電荷密度を求める。つまり結果は格子空間か、逆格子空間の関数である。これらの結果を表現する方法としては、3 次元空間の等値面を示すことが有力な手段である。TPERSP は逆格子空間の等エネルギー面の作図用として開発されたが、もちろん実空間の電荷分布等を示すことにも使用できる。

3 次元空間の物体を 2 次元の図にする方法に「ワイヤーフレームモデル」、「サーフェースモデル」、「ソリッドモデル」がある。情報量はこの順に多くなり、処理に要する計算量も増加する。一般にはこの種の CG の目的は、現実に存在する人物とか自動車などを如何に本当らしく見せるかにあるようである。しかしここでの目的は逆格子空間の等エネルギー面と関連する必要な情報を如何に忠実に、表現するかにある。もともと理論の上で存在するフェルミ面を本当らしく見せることにはなんの意味もない。つまり細部までの情報を、少ない計算量で表現できるワイヤーフレームモデルが最も適合していることになる。

図 1 は TPERSP で画いた Cu のフェルミ面の図である。このシステムでできること、作成される図の性格等を示すために話のはじめに置いてある。TPERSP の図は、この図のようにさまざまな平面の上での等高線の集まりである。図 1 は $(2\pi/a)(110)$ の X 点を中心にして、

$$(2\pi/a)(000), (2\pi/a)(111), (2\pi/a)(11-1), (2\pi/a)(200), (2\pi/a)(020), (2\pi/a)(220)$$

の 6 個の Γ 点を中心にしたフェルミ面を画いているが、ここで、手前の $(2\pi/a)(000)$ を中心にした部分の上半分を切り取っている。このように自由に一部を切り取れる機能を持っているのが、TPERSP の一つの特徴である。TSPp114 の図 7.1 ではさらに複雑な切り取りがなされている。またこの図で、手前のものに隠されている線はほとんど消してあるが、一部の隠れた線は破線で表している。一つの図の中で消したり、破線にしたりを自由に選べるのも TPERSP のもう一つの特徴である。このような機能の使用法は 6 節で述べる。

3 次元の関数を表現するもう一つの有力な手段は、3 次元空間に適当に設定した平面上での等高線を画く方法である。図 2 は、図 1 に見えている X 点を中心にした軌道を含む面で切った断面図である。この図では等エネルギー線をフェルミエネルギーの上下のエネルギーで、それぞれ１０本描いている。TPERSP を用いたこのような図の作図法は 8 節で述べる。

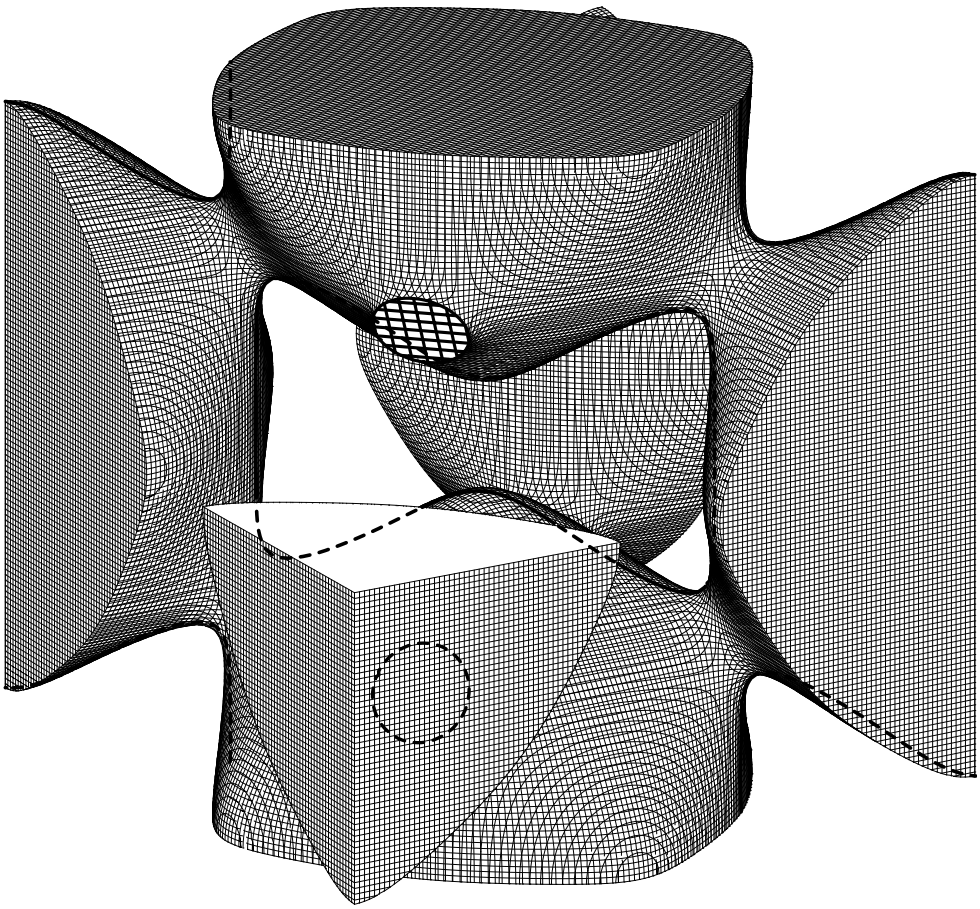


図 1: **Cu のフェルミ面**

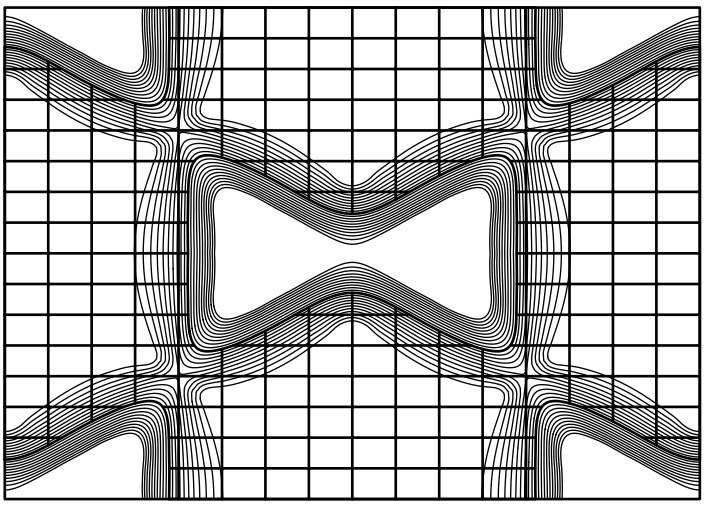


図 2: **Cu のフェルミ面の断面図**

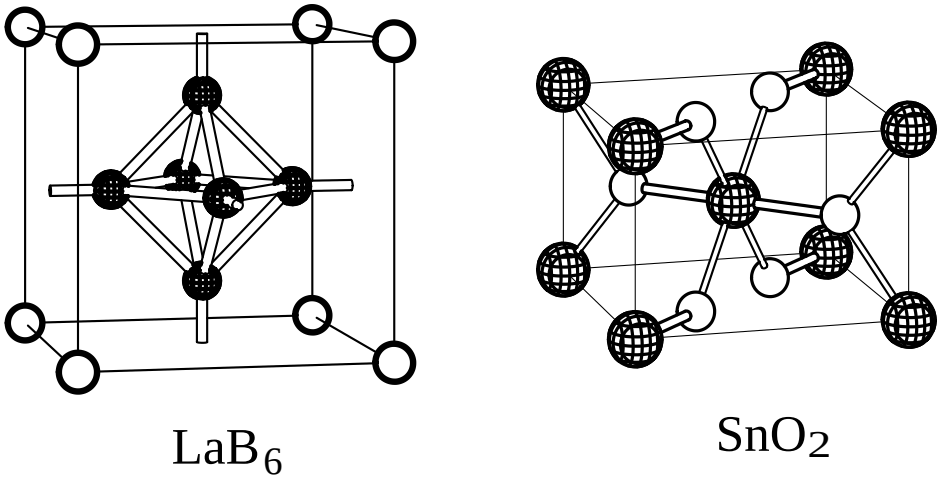


図 3: **結晶構造の例**

さらに、TPERSP には図 3 のように球と円柱の組み合わせで結晶構造のモデルを作図する機能も用意されている。このような図の作図法は 7 節で述べる。

今回 TPERSP を AYPLOT の上にのせるに際して、従来のものに重大な変更を行った。それは TPERSP の扱う実数データをすべて倍精度に変えたことである。以下に説明するように TPERSP は TSPACE と連携することが多い。TSPACE が倍精度になっている以上 TPERSP も倍精度にしておくほうが便利であるのが一つの理由である。さらに TPERSP はけっこう微妙な処理を行っているので、精度の高いほうが都合がよいことはもちろんである。 TPERSP と TSPACE はどちらも開発の当初は単精度であった。これは当時の計算機の記憶容量が非常に小さかったことと、最初にターゲットにされた機械が３６ビットという特殊なものであったからである。２０年間も動いてきたソフトを倍精度に変えることは新しいバグを発生させる可能性が高い。いろいろテストはしてあるが利用に際してはこの点に注意が必要である。

2　予備設定のプログラム

TPERSP が用いる透視図投影では、対象と視点の間に図 4 のように投影面をおき、対象上の点と視点を結ぶ直線がこの投影面と交わる点を投影点として作図する。

TPERSP では対象となる物はすべて直方体の中に存在するとしている。これは隠線処理（手前にあるものに隠されて見えない線を描かないようにすること）を行うためである。　この隠線処理という言葉には、大辞林 には「陰線処理」という漢字があててある。しかし広辞苑には「陰線」にはまったく異なる意味が書いてある。一方たとえば日経パソコン発行の新語辞典には「隠線処理」と書いてある。漢字の意味としては「陰」は光りの当らない陰という意味で「隠」は隠すとか、隠れるという意味であるから、明らかに隠線処理と書いた方がよさそうである。また「隠」の音は「イン」と「オン」と二つあるので、「いんせんしょり」と読むか、「おんせんしょり」と読むかという選択も残される。「いんせんしょり」と読むと「陰線処理」と漢字をあてられるので、「おんせんしょり」と読むべきであるという説も成り立つが、ここでは隠線処理と書いて、「いんせんしょり」と読むことにする。

対象物を格納している直方体と、投影面、視点の位置関係を確定し、併せて必要な初期設定を行うために、最初に

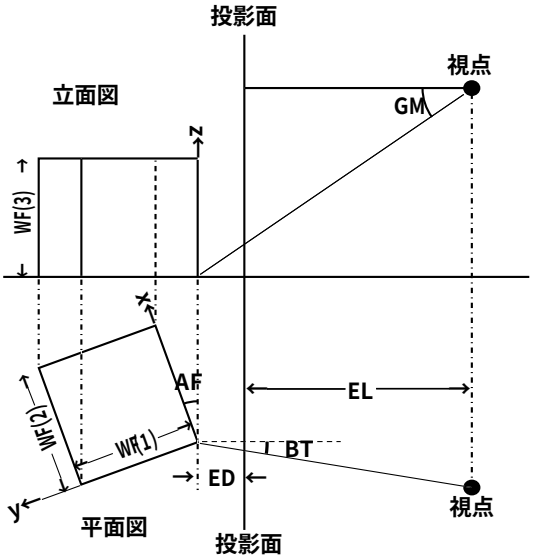


図 4: **透視図投影の説明図**

```
CALL TPERSP(XFU,WF,AF,BT,GM,ED,EL)
```

とする。最初の引数 XFU(3,2) で直方体に含まれる関数座標の x, y, z 三方向での価の範囲を

$$XFU(1,1) \leq x \leq XFU(1,2),\ \ XFU(2,1) \leq y \leq XFU(2,2),\ \ XFU(3,1) \leq z \leq XFU(3,2) \qquad (1)$$

と指定する。WF(3) はこの直方体の図としての大きさで mm 単位であたえる。ブリルアンゾーンやフェルミ面を画く場合には XFU は TSP p96 の下のほうに書いてある a^* を単位にする直交座標系を用いるのが便利である。WF の三方向の倍率は独立にとれるが、実際は等しい倍率になるようにしておくほうが、図が不自然にならない。AF 以下の引数の意味は、図 4 に示されている。AF,BT,GM はそれぞれ α, β, γ のことである。

TPERSP が用いるこの投影法は、鳥瞰図と呼ばれるものとは異なっている。鳥瞰図では対象物と視点を結ぶ線にだいたい垂直になるように投影面が置かれる。TPERSP では図に示すように、いつでも垂直に立ててある。これで対象物の垂直な線は、図の上でもいつでも互いに平行になる。結晶構造とか、ブリルアンゾーンの図を鳥瞰図の投影法で描くと図の下の部分がすぼまって、不安定な印象をあたえてしまうのであまり適当ではない。一方 TPERSP が用いるこの投影法にも欠点がある。一番端的にこの欠点が現れるのは、結晶構造の図で描く球の輪郭が円ではなく、楕円になることである。図 3 ではほとんど円に見えているが、実際は楕円である。このことは視点の位置が高すぎると顕著になるので、GM の値を特殊な場合を除けば、-15 度程度にしておくのが望ましい。また BT の値はいつも 0 にしておくのが望ましい。この引数は $y = f(x, p)$ のように、パラメターを含む関数で、いろいろなパラメーターの値の関数値を、一つの図で示すときに少しずつずらしながら描くためのものであった。

3　ブリルアンゾーンの図、TPHILD

プログラム 3.1 は、面心立方格子のブリルアンゾーンの、図 5 を画くプログラムである。まず TSP p97 の プログラム例 6.1 のように空間群と格子定数を決めている。ここでは空間群は TSP p69 で説明して

いる TSPNGE を使っている。空間群と格子定数が決定されれば、ブリルアンゾーンは確定する。そこで TSBZEG を CALL してブリルアンゾーンのエッジの両端の座標 CO を求める。TSBZEG の使用法と引数の説明は TSPp97 と p237 にある。

プログラム 3.1

```
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION NRECPT(3,30),RECTAX(4,30),CO(3,2,100)
      REAL*8 XFU(3,2),WF(3)
      DATA A,B,C,CA,CB,CC/1.0,1.0,1.0,0.0,0.0,0.0/
      DATA XFU/-1.0,-1.0,-1.0,1.0,1.0,1.0/
      DATA WF/160.0,160.0,160.0/
      DATA AF,BT,GM,ED,EL/ 10.0,0.0,-10.0,150.0,1500.0/
      CALL TSPNGE(210,1,3)
      CALL TSLATC(A,B,C,CA,CB,CC)
      CALL TSBZEG(NRECPT,RECTAX,NRP,CO,NLIN)
      CALL AYPSTR(99)
      CALL AYORIG(30.0,30.0)
      CALL TPERSP(XFU,WF,AF,BT,GM,ED,EL)
      CALL TPTRAC(1)
      CALL TPHILP(-1.0D0)
      CALL TPHILD(0.0,1,6,5,0)
      DO 42 I=1,NLIN
      X1=CO(1,1,I)
      Y1=CO(2,1,I)
      Z1=CO(3,1,I)
      X2=CO(1,2,I)
      Y2=CO(2,2,I)
      Z2=CO(3,2,I)
      CALL TPLINE(X1,Y1,Z1,X2,Y2,Z2,30)
   42 CONTINUE
      CALL AYPEND
      STOP
      END
      FUNCTION FUNCD(XA)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION XA(3)
      FUNCD=1.0
         KX=1000000.0*XA(1)
         KY=1000000.0*XA(2)
         KZ=1000000.0*XA(3)
         CALL TSKFBZ(KX,KY,KZ,1000000,IND)
         IF(IND.NE.0) FUNCD=-1.0
      RETURN
      END
      FUNCTION LFUNC(XA)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/BZPL/KKG(3,20),NNG,IEOH,INDC,IBR,ICUT
      DIMENSION XA(3)
      LFUNC=-1
      RETURN
      END
      FUNCTION FUNC(XX)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION XX(3)
      FUNC=0.0
      RETURN
      END
```

プログラム 3.1 には、三個の関数サブルーチン FUNCD,LFUNC,FUNC がついている。これらは隠線処理と切り取りを行うためのもので、必ずこの名前で使用する。LFUNC は切り取りを指示する関数サブルーチンで $LFUNC(XA(3)) = 1$ となる座標 $XA(3)$ の点は切り取られていることになる。プログラム 3.1 ではどこでも -1 としているので、どこも切り取られてはいない。切り取る必要がなければ、この関数サブルーチンをつけておくことになる。図 8 や図 1 のプログラムでは少し込み入ったものになるが、6 節で紹介

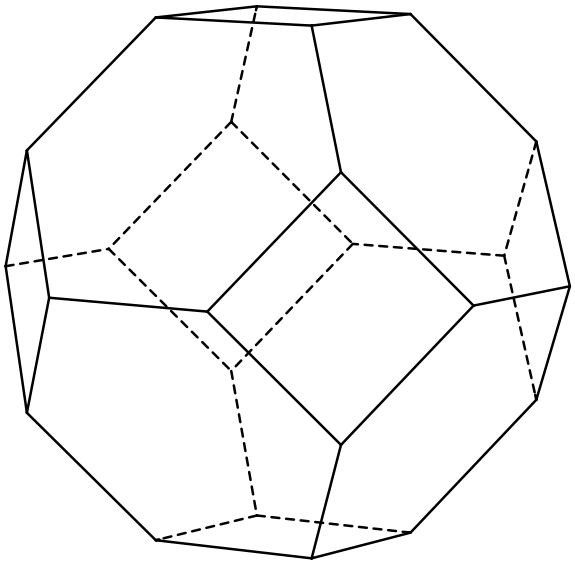


図 5: 面心立方格子のブリルアンゾーン

する。

関数サブルーチン FUNC はプログラム 3.1 のように、

```
CALL TPHILP(VA)
```

で与えられる、関数値 VA を使って、$FUNC(XA(3)) \leq VA$ となる座標 $XA(3)$ の点にはものがあるとする。ここでは FUNC はどこでも 0 で、$VA = -1.0$ としてあるから、どこにもものがないことになる。図 1 や図 8 のプログラムのような場合には、等値面を画きたい関数と同じものが FUNC に使用される。フェルミ面の場合では VA はフェルミエネルギー E_F にする。ホール面を画きたいときは、不等号を逆にする必要があるが、

```
CALL TPSCHG(-1)
```

とすればよい。

関数サブルーチン FUNCD は

```
CALL TPHILD(VA,IFLAG,IS1,IS2,IS2)
```

で与えられる関数値 VA を使って、$FUNCD(XA(3)) \leq VA$ となる座標 $XA(3)$ の点にはそれに隠されるた線は、破線で表すようにするものがあるとする。プログラム 3.1 では、FUNCD のほうは機能するようにしてあって、図 5 でブリルアンゾーンのむこう側の線を破線にしている。この関数サブルーチンでは TSPp113 で説明している TSKFBZ を使って

```
CALL TSKFBZ(KX,KY,KZ,IC,IND)
```

としている。a^* 単位であたえられた、座標の実数値を百万倍して切り捨てで整数値にし、IC を百万にして CALL している。このサブルーチンはあたえられた $\vec{k}$ 点がブリルアンゾーンの外であれば $IND = 0$ となるから、これでブリルアンゾーンの内部 (TSPp96 にブリルアンゾーンの内部とブリルアンゾーン内の使い分けの説明がある。) の点に対して関数値が -1 になり、$VA = 0$ としてあるから、ブリルアンゾーンのな

かにものがつまっていることになる。このプログラムでは簡単のために立方格子をあつかっているが、対称性が異なる格子を扱う場合は

```
XK=XA(1)
YK=XA(2)
ZK=XA(3)
CALL GETKVC(XK,YK,ZK,AK,BK,CK)
```

で直交座標から a^*, b^*, c^* をベースにする座標に変換してから、この手続きをすればよい。GETKVC は TSPp234 に説明がある。

TPHILD の引数 IS1,IS2,IS3 はサブルーチン SETDAS の引数にそのまま対応していて、この判定で見えないとした線はこれで指定される破線（一点鎖線）になる。２番目の引数 IFLAG はこの価が 0 ならば FUNCD による判定を省略して無条件に ”見えない ”とする。このようにしておいても、IS1,IS2,IS3 を 0,0,0 にしておけばすべての線が実線になる。画く線を無条件に破線や一点鎖線にしたいときには、この機能を使うことができる。IFLAG=0 の場合は、関数サブルーチン FUNCD は CALL されない。したがって必要無いことになるが、ほとんどの FORTRAN の処理系ではリンカーが未定義のルーチンがあるとしてエラーになる。なんでもよいからこの名前の関数サブルーチンをつけておかねばならない。

TPERSP の中での初期設定は

```
CALL TPHILD(0.0,0,0,0,0)
```

としてある。これで無条件で画かれる線は実線になる。図 1 や図 8 を画くプログラムでは、少し複雑な FUNCD が使われている。

ブリルアンゾーンのエッジを画くためには

```
CALL TPLINE(X1,Y1,Z1,X2,Y2,Z2,NC)
```

のように、三次元空間の座標 (X_1, Y_1, Z_1) から (X_2, Y_2, Z_2) へ直線を引くサブルーチンを CALL する。ここで最後の引数 NC は途中に設けるチェックポイントの数を与えている。TPERSP では隠線処理を行っている。もしここで設定する二つの座標の両方が見えていると判定されて、しかも途中にチェックポイントがないと、直線全体が見えていると判定される。ここで例にしている、一つのブリルアンゾーンの場合には凸多面体であるので、これでもかまわないが一般には途中で見えない部分を含む場合がある。必要なチェックポイントの数は画く図によってさまざまであるが、ほとんど場合プログラム 3.1 で使っている 30 で十分である。

4 隠線処理のアルゴリズム

TPERSP が使用している隠線処理のアルゴリズムを図 6 に示している。はじめに対象点が設定した直方体の外側にないか、切り取られていないかの判定がされる。図に示すように変数 IHL がこの部分の動きを制御している。変数 IHL は TPERSP の COMMON 領域 /TPERS1/ に含まれている。初期設定では 0 にしてある。この場合には対象点が直方体の外側や切り取られた点でないことを確認したのち、その点と視点を結ぶ線上にものがあるかどうかを検査する。

1. IHL=0 普通に隠線処理を行う。
   - 対象点が直方体の中にあり、

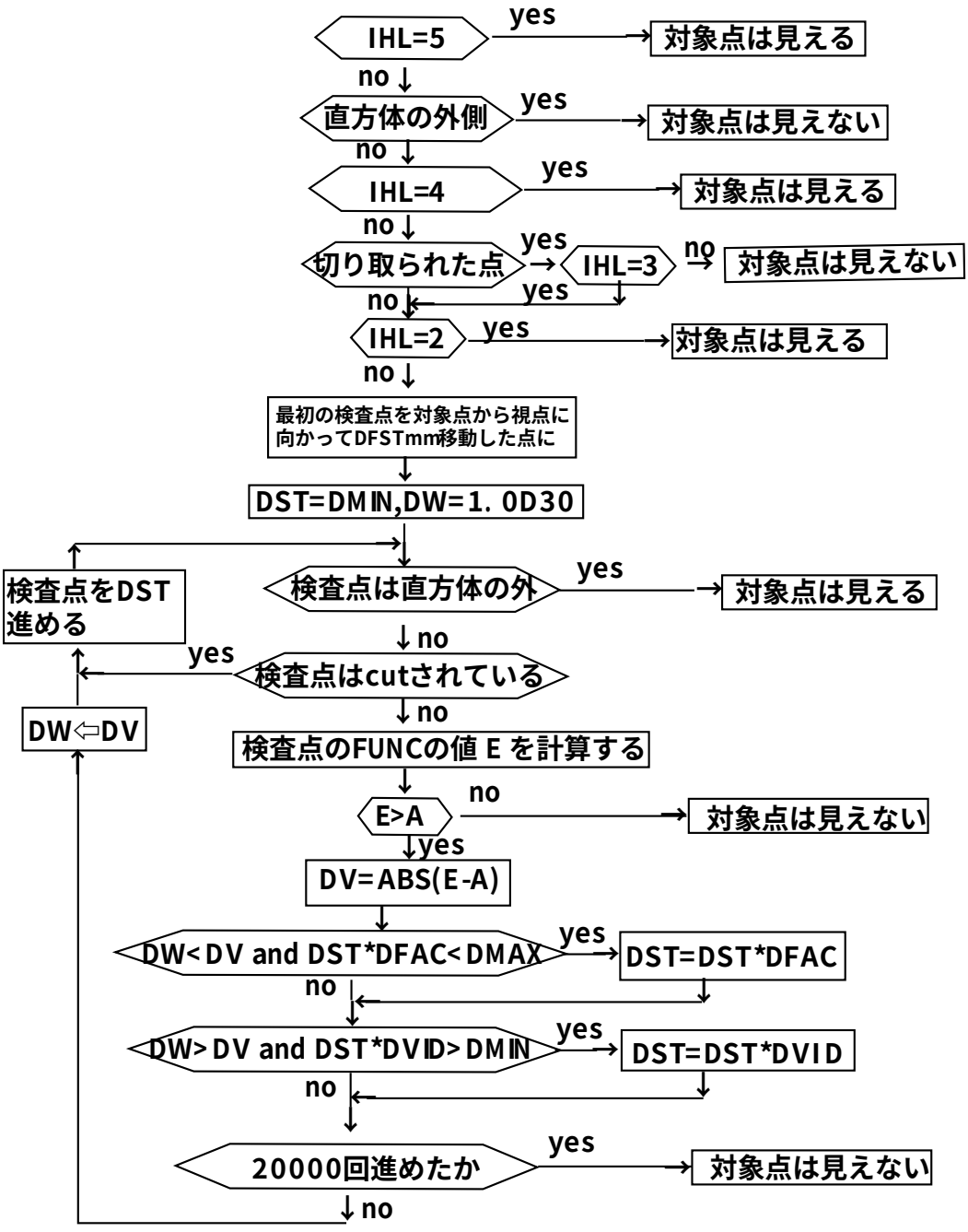


図 6: 隠線処理のアルゴリズム

- 切り取られていなくて、
- 対象点と視点の間にものがなければ、

その対象点は見えるとし、上の条件を一つでも満たさない点は見えないとする。

2. IHL=5 無条件に対象点は見えるとする。
3. IHL=4 直方体の中にあれば無条件に対象点は見えるとする。
4. IHL=3 切り取られていても無条件には見えないとしない以外は IHL=0 と同じである。
5. IHL=2

   - 対象点が直方体の中にあり、
   - 切り取られていなければ、

   その対象点は見えるとする。つまりこの場合は対象点と視点の間にものがあっても見えるとする。

上で IHL の値が 1 の場合が抜けているが、これは 7 節の結晶構造の作図の場合の特殊な隠線処理用である。またこの値を負にするのは 8 節で述べる断面図の作図用である。対象点の切り取り関係の処理が終わると、対象点と視点の間にものがないかの判定になる。2 点を結ぶ方向を決めて、その方向に DFST(mm) だけ移動した点を最初の検査点にする。以後 DST だけ検査点を移動しながら、しらべて直方体の外にでるまで続ける。DST の値は図に示すように状況に応じて変化させている。浅瀬を行く船のように操作をする。もちろん検査点が荒いと、手前にあるものを見落として見えない線が図にあらわれる。あまり細かいと検査に手間取って計算量が増加する。初期設定でのパラメーターの値は

$$DFST = 0.2,\ DMIN = 0.1,\ DMAX = 10.0,\ DFAC = 2.0,\ DVID = 0.25 \tag{2}$$

としてある。これらのパラメーターを変更するのには

```
CALL TPHLDG(DFST,DMIN,DMAX,DFAC,DVID)
CALL TPHLDS(DFST,0.01D0,DMAX,1.2D0,DVID)
```

のようにする。この形式で変えたいパラメーターだけを変えることができる。これで

```
DMIN=0.01,  DFAC=1.2
```

になってより細かく検査をするようになる。TPHLDG は現在の設定値を取り出し、TPHLDS で希望するパラメーターを与える。図 1 の作図は、初期設定のままのパラメーターで 行ったため、手前に見えているネックの切り口にもれが見られたが AdobeIllustrator に取り込んでから消してある。このソフトが使える環境にあれば、隠線処理のパラメーターに気を使うよりは、あとで消す方が楽である。

5 立方調和関数

関数サブルーチン FUNC を正規に使って、実際に隠線処理を行って作図する例として、まず TSPp269 の付録 2.B にのせている立方・六方調和関数の図を描くプログラムを紹介する。この付録にのせている図を描いたプログラムは 506 行からなる少し複雑なプログラムなので、図 7 の f 関数の一つに特化したかた

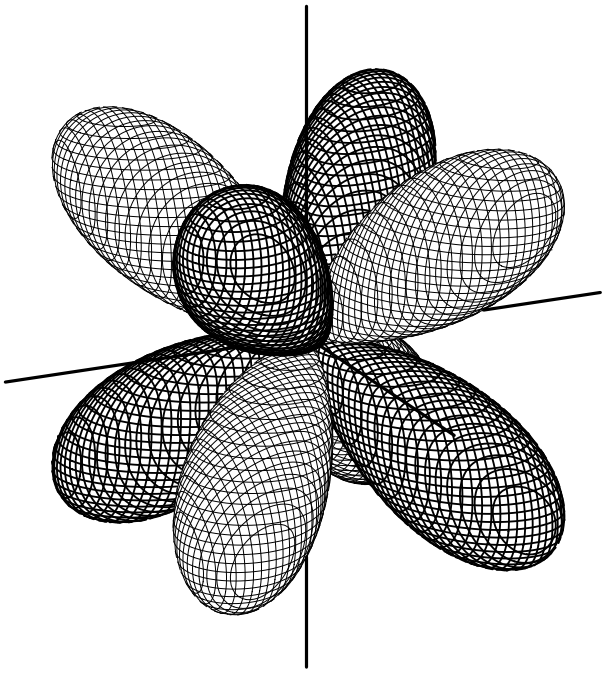


図 7: **立方調和関数 fxyz**

ちで紹介する。これらの調和関数では正負の符号の違いを線の濃淡で表現している。これを実現する方法の説明がここでの重点になる。調和関数の図示には、その角度依存性を $f(\theta, \phi)$ として

$$|f(\theta, \phi)| = r$$

であたえられる面が描かれる。これは図 7 の場合に特化すれば

$$|xyz| = r^4$$

となる。符号の違いを濃淡で示すためには、

$$xyz - r^4 = 0, \quad -xyz - r^4 = 0$$

の二つの等値面を線の太さを互いに変えて描けばよい。プログラム 5.1 は、TPERSP を CALL したのち TPTRAC を CALL している。このサブルーチンは作図の進行状況を標準出力に報告するように指定するもので、引数の値が 0 でエラー以外は報告なし、1 でほどほどの量の報告、2 でペンの動きの逐次報告になる。

プログラム 5.1 のサブルーチン TPFXYZ は三次元の等値面を描くサブルーチンである。このプログラムでは調和関数の正の領域を表す FUNCB と負の領域を表す FUNCA を使っている。つまりこのサブルーチンの最初の引数は倍制度の関数サブルーチン名で、これであたえられる関数値が二番目の引数の値になる等値面が描かれる。図 7 のように、等値面の図は実際には

$$x = constant, \quad y = constant, \quad z = constant$$

の平面の上での等高線の集まりである。引数 NN(3) は TPERSP の引数 XFU で与えられた範囲をｘｙｚ方向のそれぞれで分割する分点の数を与える。この例のように奇数にとると、範囲の中点が分点に含まれ

る。$x = constant$ の面の上の等高線は、その面の上で $NN(2) \cdot NN(3)$ のメッシュの上の関数値を用いて描かれる。このとき x の値は $NN(1)$ 個の分点を $NK(1)$ 置きにとっている。TPFXYZ の最後の引数 IGR は等値面が XFU で与えられた範囲の境界を切る場合に、その境界の平面上にメッシュを描いて蓋をするかどうかを指定する。このプログラムでは切らないことが分かっているので描かないほうの指定 0 にしている。図 1 では、この引数を 1 にして境界面にメッシュを描いて蓋をしている。

プログラム 5.1

```
      IMPLICIT REAL*8 (A-H,O-Z)
      EXTERNAL FUNCA,FUNCB
      REAL*8 XFU(3,2),WF(3)
      INTEGER NN(3),NK(3)
      D12=0.2
      XFU(1,1)=-D12
      XFU(2,1)=-D12
      XFU(3,1)=-D12
      XFU(1,2)=D12
      XFU(2,2)=D12
      XFU(3,2)=D12
      SIZE=100.0
      WF(1)=SIZE
      WF(2)=SIZE
      WF(3)=SIZE
C
      IFL=70
      AF=25.0
      BT=0.0
      GM=-20.0
      ED=150.0
      EL=1500.0
      CALL AYPSTR(IFL)
      CALL AYORIG(20.0,20.0)
      CALL TPERSP(XFU,WF,AF,BT,GM,ED,EL)
      CALL TPTRAC(1)
      CALL TPSCHG(-1)
      EF=0.0
      CALL TPHILP(EF)
      CALL TPHILD(0.0,0,0,0,0,0)
      DO 11 I=1,3
      NN(I)=161
      NK(I)=2
   11 CONTINUE
      IGR=0
      CALL LINEWD(0.7)
      CALL TPFXYZ(FUNCB,EF,NN,NK,IGR)
      CALL LINEWD(0.1)
      CALL TPFXYZ(FUNCA,EF,NN,NK,IGR)
      CALL LINEWD(1.2)
      X1=-D12
      Y1=0.0
      Z1=0.0
      X2=D12
      Y2=0.0
      Z2=0.0
      CALL TPLINE(X1,Y1,Z1,X2,Y2,Z2,30)
      X1=0.0
      X2=0.0
      Y1=-D12
      Y2=D12
      CALL TPLINE(X1,Y1,Z1,X2,Y2,Z2,30)
      Y1=0.0
      Y2=0.0
      Z1=-D12
      Z2=D12
      CALL TPLINE(X1,Y1,Z1,X2,Y2,Z2,30)
      CALL AYPEND
```

```
STOP
END
```

この例では描く対象が調和関数の正の部分と負の部分の二つあるので隠線処理関数 FUNC はプログラム 5.2 のように FUNCA,FUNCB の値のどちらか大きい方をとることにしている。またここでは関数値が正の領域にものがあるとしなければならないので、プログラム 5.1 で

```
CALL TPSCHG(-1)
EF=0.0
CALL TPHILP(EF)
```

としている。

TPERSP では計算時間を計るサブルーチン PTIME を使っている。プログラム 5.2 の最後の部分ではそれを IBM-AIX のフォートランのサブルーチン ETIME を使って作っている。ちなみに PTIME は TPERSP が開発された頃 NEC の機械のサブルーチンで CLOCKM というのは FLAPW が開発された当時の富士通の機械のサブルーチンである。このところは使用するシステムに合わせて変更しなければならない。TPERSP の中では PTIME は計算時間を計っているだけで、作図のアルゴリズムには関係ないので、正確な時間を返さないで、たとえばいつでも 1 を返すようなものでもかまわない。最近のシステムでは作図に要する時間は問題にならなくなっている。

プログラム 5.2

```
FUNCTION FUNCD(XA)
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION XA(3)
 FUNCD=0.0
RETURN
END
INTEGER FUNCTION LFUNC(X)
IMPLICIT REAL*8(A-H,O-Z)
REAL*4 X(3)
LFUNC=0
RETURN
END
FUNCTION FUNC(XX)
REAL*4 XX(3)
WA=FUNCA(XX)
WB=FUNCB(XX)
IF(WA.GE.WB) THEN
  FUNC=WA
ELSE
  FUNC=WB
END IF
RETURN
END
FUNCTION FUNCB(XX)
IMPLICIT REAL*8(A-H,O-Z)
REAL*8 KB
DIMENSION XX(3)
RR=XX(1)**2+XX(2)**2+XX(3)**2
R4=RR*RR
FUNCB=XX(1)*XX(2)*XX(3)-R4
RETURN
END
FUNCTION FUNCA(XX)
IMPLICIT REAL*8(A-H,O-Z)
REAL*8 KB
DIMENSION XX(3)
RR=XX(1)**2+XX(2)**2+XX(3)**2
R4=RR*RR
FUNCA=-XX(1)*XX(2)*XX(3)-R4
```

```
      RETURN
      END
      SUBROUTINE CLOCKM(ITIME)
C
C       FOR UNIX
C
      DIMENSION TERY(2)
C
C     etime for AIX FORTRAN
      CALL ETIME_(TERY)
C     etime for SUN FORTARAN
C     CALL ETIME(TERY)
      ITIME=TERY(1)*1000
      RETURN
      END
      SUBROUTINE PTIME(T)
      REAL*8 T
      CALL CLOCKM(IT)
      T=IT/3600000.0
      RETURN
      END
```

6 フェルミ面のプログラム

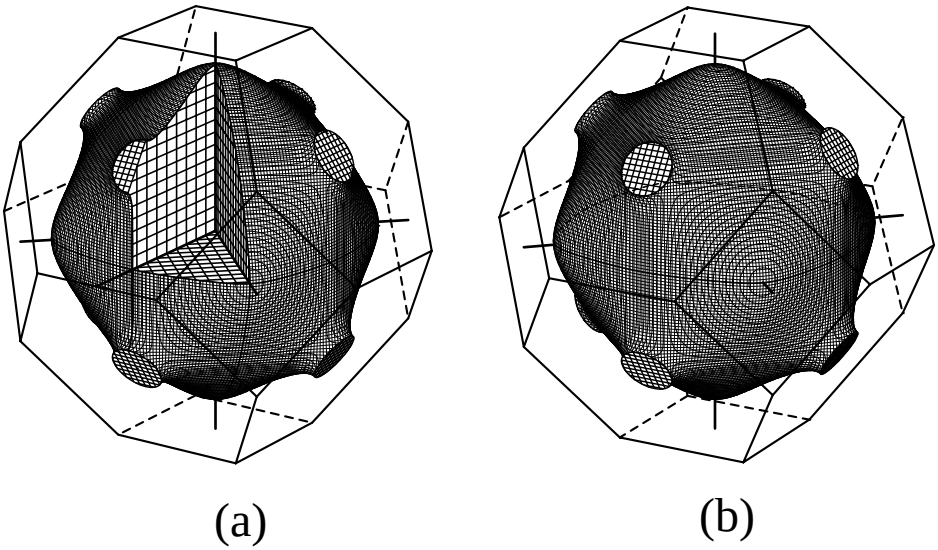


図 8: **Cu のフェルミ面１**

フェルミ面を描くためには逆格子空間のどの $\vec{k}$ に対してもバンドのエネルギー値をあたえる関数サブルーチンが必要である。その都度バンド計算を実行するのはもちろん実際的ではない。実際には、逆格子空間の比較的あらいメッシュ点上で計算されたものから細かいメッシュに内挿したものを作り、それを使った逆格子空間のどの $\vec{k}$ に対してもバンドのエネルギー値をあたえる関数サブルーチン FUNC が作られる。この手続きを実現する手段には空間群の既約表現に関する情報が使われている。この部分は説明が長くなるので、別稿にすることにし、ここでは切り取り関数や隠線処理関数の使い方を主に説明する。この .pdf ファイルがある同じページに、ここで示している銅のフェルミ面を描いたプログラムのソースがあるのでそれを参考にされたい。またこの節の説明にでてくる fort.31 を作成したソースとバンド計算の結果も同様に置

いてある。

図 8(a) を画くプログラムの最初の部分をプログラム 6.1 に示す。最初の COMMON 文は FUNC にパラメーターを渡すためのものである。この部分では空間群と格子定数の情報を fort.1 から読み込んで、TSPp73 のプログラム例 5.3 の方式を用いて空間群と格子定数を設定したのち、TSBZEG を CALL してブリルアンゾーンを画くデータを取り込んでいる。この fort.1 はバンド計算のプログラムが使用しているものを流用している。

プログラム 6.1

```
      IMPLICIT REAL*8 (A-H,O-Z)
      EXTERNAL FUNC
      COMMON/ENRSPL/EM(366145),PN(3),P(3),N(3),JS
      DIMENSION NRECPT(3,30),RECTAX(4,30),CO(3,2,100)
      DIMENSION JB(2,3)
      REAL*8 XFU(3,2),WF(3)
  REAL*8 XX(3),XC(3),DR(3)
      INTEGER M(3),NN(3),NK(3)
      DATA XFU/-1.0,-1.0,-1.0,1.0,1.0,1.0/
      DATA WF/120.0,120.0,120.0/
      DATA AF,BT,GM,ED,EL/10.0,0.0,-20.0,150.0,1500.0/
      READ(1,*)
      READ(1,*) IL,NGEN,INV
      CALL TSPACE(IL)
      DO 1 I=1,NGEN
      READ(1,*) JA,((JB(J,K),J=1,2),K=1,3)
      CALL TSGENR(JA,JB)
    1 CONTINUE
      CALL TSPGRP(INV)
      CALL TSPGDS
      READ(1,*) A,B,C
      READ(1,*) CA,CB,CC
      CALL TSLATC(A,B,C,CA,CB,CCO)
      CALL TSBZEG(NRECPT,RECTAX,NRP,CO,NLIN)
```

プログラム 6.2 はバンドエネルギー $E(\vec{k})$ を任意の $\vec{k}$ に対して計算する関数サブルーチン FUNC が使用するデータを fort.31 から読み込んで COMMON/ENRSPL/ にしまっている。Cu の場合５枚の d バンドと１枚の s バンドが重なっているので、都合６枚のバンドを順次内挿して fort.31 に書き出してある。

```
      READ(31) IXMAX,NB,NNEE
```

で各バンドのデータについた見出し部分を読んでいる。IXMAX は Γ 点から X 点までの刻み点の数で両端もカウントしている。 NB はバンド番号で、NNEE は三次元のメッシュ点の数である。図 8(a) の作図に使用されたデータは IXMAX=129,NNEE=366145 である。

プログラム 6.2

```
      NB=6
      DO 10 J=1,NB
      READ(31) IXMAX,NB,NNEE
      READ(31) (EM(K),K=1,NNEE)
      WRITE(6,*) IXMAX,NB,NNEE
   10 CONTINUE
      PN(1)=1.0
      PN(2)=1.0
      PN(3)=1.0
      N(1)=IXMAX-1
      N(2)=N(1)
      N(3)=N(1)
      JS=3
      DO 51 I=1,3
      P(I)=PN(I)/FLOAT(N(I))
   51 CONTINUE
```

プログラム 6.3 は AYPSTR と TPERSP を起動したのち TPHILD で FUNCD で隠される線を破線にするように設定している。さらに CALL TPCIHL(3) で切り取られた部分を残すように指定している。これは図 8(a) でフェルミ面は切り取られているが、ブリルアンゾーンのエッジのその部分は残してあるのに対応している。これだけの手当をしたのち TSBZEG があたえた座標にしたがってブリルアンゾーンのエッジを描き、Δ 軸を 3 本と Σ 軸を 1 本描いている。

プログラム 6.3

```
      CALL AYPSTR(81)
      CALL AYORIG(30.0,30.0)
      CALL TPERSP(XFU,WF,AF,BT,GM,ED,EL)
      CALL TPTRAC(1)
      EF=0.41714
      CALL TPHILP(EF)
      CALL TPHILD(0.0,1,6,5,0)
      CALL LINEWD(1.6)
      CALL TPCIHL(3)
      DO 42 I=1,NLIN
      X1=CO(1,1,I)
      Y1=(CO(2,1,I)
      Z1=(CO(3,1,I)
      X2=CO(1,2,I)
      Y2=(CO(2,2,I)
      Z2=(CO(3,2,I)
      CALL TPLINE(X1,Y1,Z1,X2,Y2,Z2,30)
   42 CONTINUE
C
      CALL LINEWD(2.0)
      CALL TPLINE(-1.0D0,0.0D0,0.0D0,1.0D0,0.0D0,0.0D0,30)
      CALL TPLINE(0.0D0,-1.0D0,0.0D0,0.0D0,1.0D0,0.0D0,30)
      CALL TPLINE(0.0D0,0.0D0,-1.0D0,0.0D0,0.0D0,1.0D0,30)
      CALL TPLINE(0.0,0.0,0.0,-0.75D0,-0.75D0,0.0,30)
```

プログラム 6.4 はフェルミ面を描くためのメインな部分である。CALL TPCIHL(0) で切り取りの処理を正常にもどし、CALL TPHILD(0.0,0,0,0,0) で無条件に実線で描くようにしている。TPFXYZ については前節で説明してある。この例のように多くの線で表すときは線の太さはできるだけ細くするほうがよい。サブルーチン

```
TPSECT(FUNCT,AA,DR,XC,IA,IG,NX,NY,KX,KY)
```

は関数サブルーチン FUNCT で与えられる関数値が AA になる面を、中心座標 XC(3) で、方向余弦 DR(3) で決まる平面で切った切り口のを描くサブルーチンである。この例に示すように DR(3) の値は必ずしも規格化しておかなくてもよい。また XC(3) は文字通りの中心座標でなくてもよい。その平面の上ならばどこの点を指定しても、結果は変わらない。同じ面で AA の値を変えた等高線を続けて描く場合には IA=1 にする。IG=1 にすると図にあるように切り口に格子模様を描く。これを省略するときは IG=0 にする。刻み数 NX,NY であたえる格子点上で計算された関数値をもとにして、等高線が描かれる。IG=1 にしたときに描かれる格子模様は KX,KY おきの格子点を通って描かれる。したがってこの値はそれぞれ NX-1,NY-1 の約数になっているのが望ましい。

プログラム 6.4

```
      CALL LINEWD(0.1)
      CALL TPCIHL(0)
      CALL TPHILD(0.0,0,0,0,0)
      DO 11 I=1,3
      NN(I)=129
      NK(I)=1
   11 CONTINUE
      CALL TPFXYZ(FUNC,EF,NN,NK,1)
      CALL LINEWD(1.2)
```

```
      XC(1)=0.0
      XC(2)=0.0
      XC(3)=0.0
      DR(1)=0.0
      DR(2)=0.0
      DR(3)=1.0D0
      CALL TPSECT(FUNC,EF,DR,XC,0,1,129,129,4,4)
      DR(1)=1.0D0
      DR(2)=-1.0D0
      DR(3)=0.0
      CALL TPSECT(FUNC,EF,DR,XC,0,1,129,129,4,4)
      DR(1)=1.0D0
      DR(2)=0.0
      DR(3)=0.0
      CALL TPSECT(FUNC,EF,DR,XC,0,1,129,129,4,4)
      CALL LINEWD(1.1)
      DO 12 IX=1,2
      DO 12 IY=1,2
      DO 12 IZ=1,2
         DR(1)=1.0D0*(3-2*IX)
         XC(1)=0.5D0*DR(1)
         DR(2)=1.0D0*(3-2*IY)
         XC(2)=0.5D0*DR(2)
         DR(3)=1.0D0*(3-2*IZ)
         XC(3)=0.5D0*DR(3)
         CALL TPSECT(FUNC,EF,DR,XC,0,1,257,257,4,4)
 12   CONTINUE
      CALL AYPEND
      STOP
      END
```

プログラム 6.5 の FUNCD は、プログラム 3.1 のそれと同じである。切り取り関数 LFUNC は同様な論理でブリルアンゾーンの外側を切り取り、さらに図 8(a) の 3 平面で切り取りを行っている。前に述べたように FUNC の説明は別稿に譲ることにするが、等高線を描くためのデータだけであれば、メッシュ点上の値だけで十分であるが 4 節で説明した隠線処理のアルゴリズムの実行には任意の $\vec{k}$ に対するエネルギー値が必要になることを注意しておく。

プログラム 6.5

```
      FUNCTION FUNCD(XA)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION XA(3)
      FUNCD=1.0
      DO 1 I=1,NNG
         KX=1000000.0*XA(1)-1000000*KKG(1,I)
         KY=1000000.0*XA(2)-1000000*KKG(2,I)
         KZ=1000000.0*XA(3)-1000000*KKG(3,I)
         CALL TSKFBZ(KX,KY,KZ,1000000,IND)
         IF(IND.NE.0) FUNCD=-1.0
    1 CONTINUE
      RETURN
      END
      FUNCTION LFUNC(XA)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION XA(3)
      LFUNC=-1
         KX=1000000.0*XA(1)
         KY=1000000.0*XA(2)
         KZ=1000000.0*XA(3)
         CALL TSKFBZ(KX,KY,KZ,1000000,IND)
         IF(IND.EQ.0) LFUNC=1
      IF(XA(1).LT.0.0.AND.XA(2).LT.XA(1)-1.0D-4
     &      .AND.XA(3).GT.0.0) LFUNC=1
      RETURN
      END
C
```

プログラム 6.6 は、図 8(b) のための FUNCD,LFUNC である。ここでは LFUNC はブリルアンゾーンの外側を切り取るだけにしている。FUNCD は COMMON 変数 INDD を使って、プログラム 6.5 ののものと、FUNC とを使い分けるようにしている。

プログラム 6.6

```
      FUNCTION FUNCD(XA)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/BZPL/INDD
      DIMENSION XA(3)
      FUNCD=1.0
      IF(INDD.EQ.0) THEN
         KX=1000000.0*XA(1)
         KY=1000000.0*XA(2)
         KZ=1000000.0*XA(3)
         CALL TSKFBZ(KX,KY,KZ,1000000,IND)
         IF(IND.NE.0) FUNCD=-1.0
      ELSE IF(INDD.EQ.1) THEN
         WW=FUNC(XA)
         FUNCD=WW
      END IF
      RETURN
      END
      FUNCTION LFUNC(XA)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION XA(3)
      LFUNC=-1
         KX=1000000.0*XA(1)
         KY=1000000.0*XA(2)
         KZ=1000000.0*XA(3)
         CALL TSKFBZ(KX,KY,KZ,1000000,IND)
         IF(IND.EQ.0) LFUNC=1
      RETURN
      END
```

プログラム 6.7 のように、メインプログラムで TPHILD を CALL して、FUNCD による制御の定数をかえ、さらに INDD=1 として FUNCD の関数値を FUNC のそれになるように切り替える。CALL TPCIHL(2) とすると、4 節でのべたように、FUNC で隠される線を消す機能が押さえられるので、続いて CALL される TPSECT による軌道がフェルミ面の向こう側を破線にして描かれることになる。

プログラム 6.7

```
      CALL LINEWD(1.2)
      CALL TPHILD(EF,1,4,4,0)
      INDD=1
      CALL TPCIHL(2)
      XC(1)=0.0
      XC(2)=0.0
      XC(3)=0.0
      DR(1)=0.0
      DR(2)=0.0
      DR(3)=1.0D0
      CALL TPSECT(FUNC,EF,DR,XC,0,0,129,129,4,4)
      CALL AYPEND
```

プログラム 6.8 は、図 1 のための FUNCD,LFUNC である、ここではブリルアンゾーンを描かないので、FUNCD は FUNC そのままにしてある。LFUNC は COMMON 変数 ICUT を用いて切り取りの方法を 3 通りに変えている。ICUT=0 では 0 0 0 を中心にしたブリルアンゾーンの上半分を切り取っている。ICUT=1,ICUT=2 は L 点のネックを描くための切り取り関数で、このように手のこんだことをするのは、図 12 に示すように、この断面の切り口に隣のゾーンの大きな軌道があり、この図ではこの部分は描かない方がすっきりするためである。

プログラム 6.8

```
      FUNCTION FUNCD(XA)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/DICUT/ICUT
      DIMENSION XA(3)
         WW=FUNC(XA)
         FUNCD=WW
      RETURN
      END
      FUNCTION LFUNC(XA)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/DICUT/ICUT
      DIMENSION XA(3)
      LFUNC=-1
      IF(ICUT.EQ.0) THEN
         KX=1000000.0*XA(1)
         KY=1000000.0*XA(2)
         KZ=1000000.0*XA(3)
         CALL TSKFBZ(KX,KY,KZ,1000000,IND)
         IF(IND.NE.0.AND.XA(3).GT.0.0) LFUNC=1
      ELSE IF(ICUT.EQ.1) THEN
         KX=1000005.0*XA(1)
         KY=1000005.0*XA(2)
         KZ=1000005.0*XA(3)
         CALL TSKFBZ(KX,KY,KZ,1000000,IND)
         IF(IND.NE.0.AND.XA(3).GT.0.0) LFUNC=1
         IF(ABS(XA(2)-XA(1)).GT.1.0D0) LFUNC=1
         IF(IND.EQ.0.AND.XA(3).LT.0.0) LFUNC=1
      ELSE IF(ICUT.EQ.2) THEN
         IF(ABS(XA(2)-XA(1)).GT.1.0D0) LFUNC=1
         IF(XA(3).GT.0.0) LFUNC=1
      END IF
      RETURN
      END
```

プログラム 6.9 は、図 1 のためのメインプログラムの一部である。ここでは ICUT の値を変えたり、CALL TPCIHL(2) で FUNC による隠線処理を無効にしたりしている。

プログラム 6.9

```
     ICUT=0
      CALL TPFXYZ(FUNC,EF,NN,NK,1)
      CALL LINEWD(1.3)
         ICUT=1
         DR(1)=1.0D0
         XC(1)=0.5D0
         DR(2)=1.0D0
         XC(2)=0.5D0
         DR(3)=1.0D0
         XC(3)=0.5D0
      CALL TPSECT(FUNC,EF,DR,XC,0,1,257,257,4,4)
      CALL TPHILD(EF,1,4,4,0)
      CALL TPCIHL(2)
      CALL LINEWD(1.4)
        ICUT=2
         DR(1)=1.0D0
         XC(1)=0.5D0
         DR(2)=1.0D0
         XC(2)=0.5D0
         DR(3)=-1.0D0
         XC(3)=-0.5D0
      CALL TPSECT(FUNC,EF,DR,XC,0,0,257,257,4,4)
      CALL LINEWD(1.5)
        ICUT=0
        XC(1)=1.0
```

```
      XC(2)=1.0
      XC(3)=0.0
      DR(1)=1.0
      DR(2)=1.0
      DR(3)=0.0D0
    CALL TPSECT(FUNC,EF,DR,XC,0,0,129,129,4,4)
    CALL AYPEND
```

7 結晶構造の作図

結晶構造の作図も TPERSP の重要な機能である。原子の位置を球で表し、その間を円柱で結ぶ図 3 に示すような結晶構造を、TPERSP を使って表すことができる。プログラム 7.1 は、このために一般的に使える tscsdtmn.f の最初の部分である。この部分は、ほとんど宣言文だけであるが、最後の

CALL TSPPRP(NAT,NKA,KKAT)

で、バンド計算に使用するファイル fort.1 からデータを読んで空間群と結晶構造と格子定数を TSPACE の COMMON 変数にあたえている。 このサブルーチンの内容については TSP p73 の § 5.5 に、具体例と共に詳しい記述があるので参照されたい。NAT は単位胞に含まれる原子数、NKA は原子の種類の数、KKAT(100) 原子の番号と種類の番号の対応表が入っている。

プログラム 7.1

```
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*4 WIDTH,XSS,ZSS
      INTEGER KKAT(100)
     &      ,JBO(2,10),NKO(30),IPT(10)
      REAL*8 RB(30),RR(10)
      REAL*8 WFF(3),XFU(3,2)
      REAL*8 XXFU(3,2),XC(3,600),X1(3,600),X2(3,600)
     &      ,XOUT(3,30),BOL(30),VA(3,50)
      REAL*8 XCC(3),R
      INTEGER IK(300),JBB(2,600),IBB(600)
      CHARACTER*1 CMARK(61)
      DATA CMARK/'1','2','3','4','5','6','7','8','9','A',
     &           'B','C','D','E','F','G','H','I','J','K','L',
     &           'M','N','O','P','Q','R','S','T','U','V',
     &           'W','X','Y','Z','a','b','c','d','e','f',
     &           'g','h','i','j','k','l','m','n','o','p',
     &           'q','r','s','t','u','v','w','x','y','z'/
      CALL TSPPRP(NAT,NKA,KKAT)
```

プログラム 7.2

```
      READ(5,*) NBO
      write(6,*) NBO
      IF(NBO.GT.0) THEN
      DO 4 I=1,NBO
      READ(5,*) (JBO(J,I),J=1,2),BOL(I),RB(I)
    4 CONTINUE
      CALL TSBOGN(JBO,BOL,NBO,0)
      JF=6
      CALL TSBOVR(JF)
      END IF
      READ(5,*) (RR(I),I=1,NKA)
      READ(5,*) (IPT(I),I=1,NKA)
      READ(5,*) MA,MB,MC,KABC
      READ(5,*) NOUT
      IF(NOUT.NE.0) THEN
        DO 5 I=1,NOUT
    5   READ(5,*) (XOUT(J,I),J=1,3),NKO(I)
      END IF
```

```
      CALL TSCSDT(MA,MB,MC,KABC,XOUT,NKO,NOUT
     &    ,XXFU,XC,IK,NNS,JBB,IBB,NBB,X1,X2,NLINE)
      write(6,*) ' NNS=',NNS,' NBB=',NBB
      write(6,600) (IK(I),I=1,NNS)
      write(6,600) (IBB(I),I=1,NBB)
      write(6,600) ((JBB(J,I),J=1,2),I=1,NBB)
 600  FORMAT(20I4)
      READ(5,*) SIZE,AF,BT,GM,ED,EL
      READ(5,*) ILF,IPOINT
      RMAX=0.0
      DO 10 I=1,NKA
      IF(RMAX.LT.RR(I)) RMAX=RR(I)
   10 CONTINUE
      DO 31 I=1,3
      XFU(I,1)=XXFU(I,1)*SIZE-RMAX
      XFU(I,2)=XXFU(I,2)*SIZE+RMAX
   31 CONTINUE
      WFF(1)=XFU(1,2)-XFU(1,1)
      WFF(2)=XFU(2,2)-XFU(2,1)
      WFF(3)=XFU(3,2)-XFU(3,1)
      CALL AYPSTR(ILF)
      CALL AYORIG(10.0,10.0)
      CALL TPERSP(XFU,WFF,AF,BT,GM,ED,EL)
      CALL TPTRAC(1)
```

プログラム 7.2 は tscsdtmn.f の二番目の部分である。ここではまず、原子を結ぶ円柱を決めている。TSBOGN、TSBOVR については TSP p184 の§9.2 に、説明がある。変数 BOL(I) は TSP p185 にも説明があるように、小数点以下５桁の数であたえなければならない。続いて原子の種類ごとに半径とパターンを読み込んでいる。使用できるパターンの見本は図 9 に示されている。

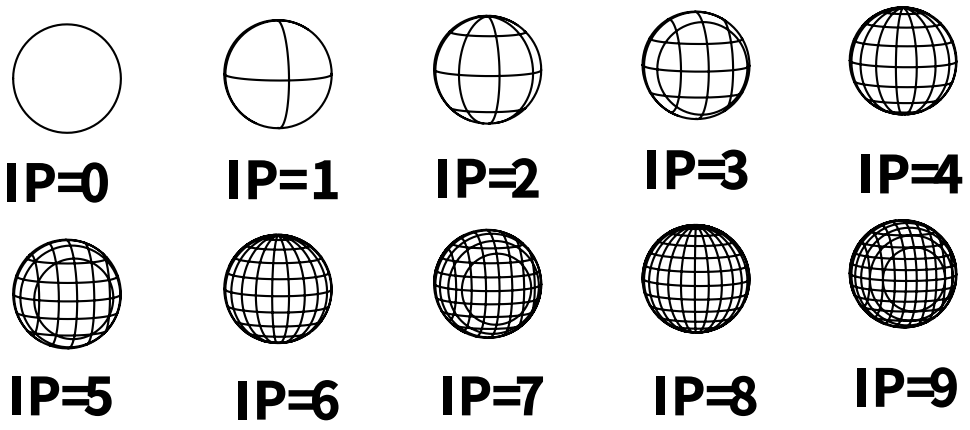
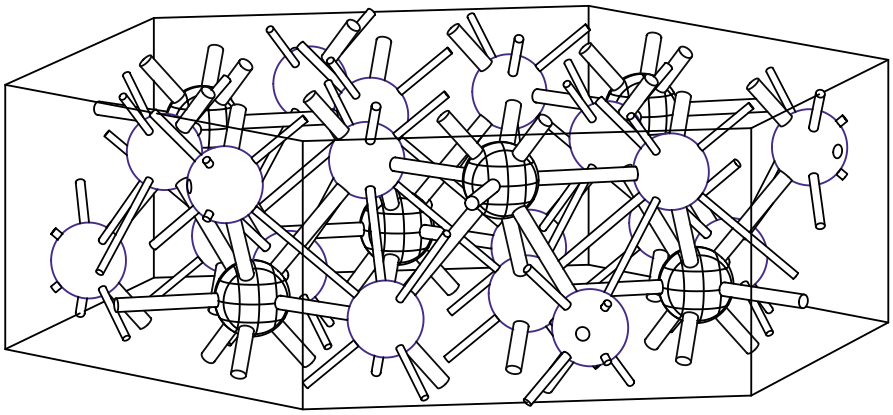


図 9: **球につけるパターン見本**

TSCSDT の引数の説明は TSP p230 に詳しく説明されている。NA=0 という六方晶の特殊な引数の使い方は図 10 や、TSP p75 の図のように六方晶らしく見せるための手段である。普通に単位胞を描くと、単斜晶か直方晶のように見えてしまう。

XXFU,XC,X1,X2 の単位は立方対称でなくても共通にとられているので、図の大きさを決める変数 SIZE を共通に掛けている。最大の球の半径の分だけ直方体を広げて TPERSP を CALL するときの引数にしているが、これは表す単位胞の面、辺、角にくる球を収容するためである。このプログラムでは XFU がすでに mm 単位にしてあるので、WFF は単純に XFU の差にしてある。

プログラム 7.3



$LaCl_3$

図 10: **六方晶の少し複雑な例**

```
   CALL TPCLEA
   DO 51 I=1,NNS
   JP=0
   IF(IK(I).NE.0) JP=IPT(KKAT(IK(I)))
   IF(IK(I).NE.0) R=RR(KKAT(IK(I)))
   IF(IK(I).EQ.0) R=0.001
   XCC(1)=XC(1,I)*SIZE
   XCC(2)=XC(2,I)*SIZE
   XCC(3)=XC(3,I)*SIZE
   CALL TPSETS(XCC,R,JP)
   IF(IK(I).NE.0.AND.IPOINT.NE.0) THEN
     XA=XCC(1)
     YA=XCC(2)
     ZA=XCC(3)
     CALL TPPOSI(XA,YA,ZA,XS,ZS)
     XSS=XS
     ZSS=ZS
     CALL NRMARK(IPOINT,CMARK(IK(I)),XSS,ZSS)
   END IF
51 CONTINUE
   DO 52 I=1,NBB
   R=RB(IBB(I))
   J1=JBB(1,I)
   J2=JBB(2,I)
   CALL TPSETB(J1,J2,R)
52 CONTINUE
   DO 53 I=1,NNS
   IF(IK(I).NE.0) THEN
     WIDTH=1.0D0+KKAT(IK(I))*0.01D0
     WRITE(6,601) I,KKAT(IK(I)),WIDTH
     CALL LINEWD(WIDTH)
     JJ=I
     CALL TPDRWS(JJ)
   END IF
53 CONTINUE
   DO 54 I=1,NBB
   JJ=I
   WIDTH=1.0D0-IBB(I)*0.01D0
   WRITE(6,601) I,IBB(I),WIDTH
```

```
 601  FORMAT(2I4,F6.2)
      CALL LINEWD(WIDTH)
      CALL TPDRWB(JJ)
   54 CONTINUE
      CALL LINEWD(1.0)
      DO 55 I=1,NLINE
      X11=X1(1,I)*SIZE
      Y1=X1(2,I)*SIZE
      Z1=X1(3,I)*SIZE
      X22=X2(1,I)*SIZE
      Y2=X2(2,I)*SIZE
      Z2=X2(3,I)*SIZE
      CALL TPLINE(X11,Y1,Z1,X22,Y2,Z2,191)
   55 CONTINUE
      CALL AYPEND
      STOP
      END
```

プログラム 7.3 は tscsdtmn.f の三番目の部分である。ここに TPERSP が用意している結晶構造用のサブルーチンがすべて現れている。最初の TPCLEA は球と円柱の登録数を 0 にセットするほか、隠線処理を、物として球と円柱しか存在しないときに有効で効率的な方法に切り換えている。つまり、4 節で説明したような、対象点と視点の間をステップを踏んで逐次ものがあるかどうかを判定するのではなく、幾何学の公式を使って対象点と視点を結ぶ直線が球や円柱と交わるかどうかを判定するように切り替える。続いて TSCSDT が返してきた球と円柱を置いていく。球を置くサブルーチンが TPSETS で、中心の座標 XCC(3) 半径 R パターン JP をあたえて一つづつ置いていく。原子の種類ごとに半径とパターンを変えることにしているので、IK(I) に格納されている原子の番号を、KKAT(IK(I)) として原子の種類の番号に付け替えている。また、このとき円柱が単位胞の境界面で終わる位置を示している JP=0 の球も置かなければならない。これは円柱を置く TPSETB が両端の球の番号 J1,J2 で円柱の位置を決めているからである。TPSETB の最後の引数 R は円柱の半径である。境界面に置く球の半径を 0.001 としているが、この値を大きくすると隠線処理でその球に隠される部分の線が消えてしまうことがある。この 0.001 という値は事実上の 0 になっている。

球の中心の三次元座標が順次あたえられているここで、その点に対応する投影面上の二次元座標を求めるために CALL TPPOSI(XA,YA,ZA,XS,ZS) としている。ここで求まった XS,ZS の位置に AYPLOT のサブルーチン NRMARK を直接 CALL して図 11 に示すように原子の番号をそれぞれの球につけることができる。このとき AYPLOT は単精度のデータを使用していることに注意しなければならない。

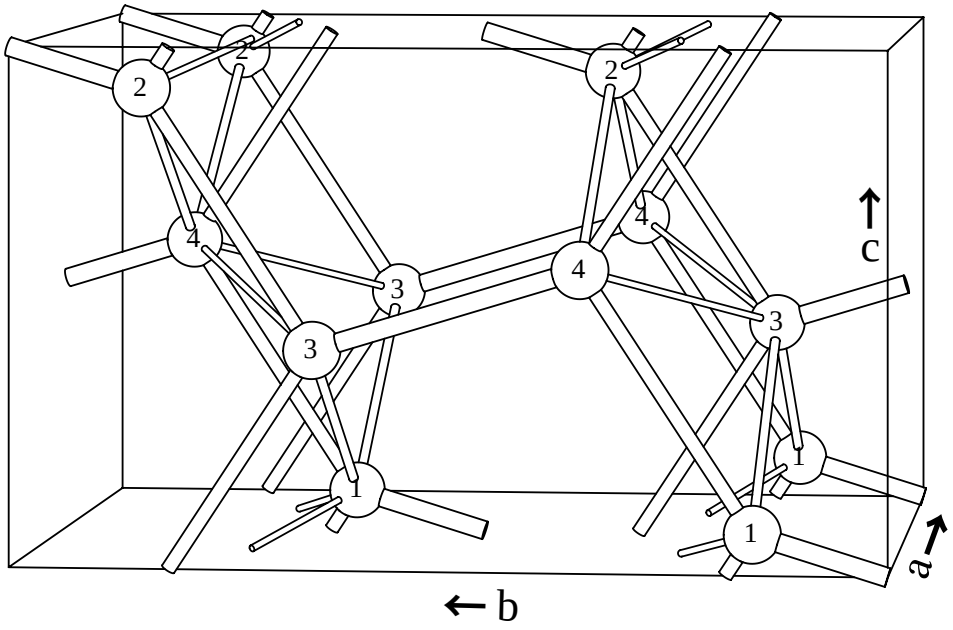
描くすべての球と円柱を置いた後、TPDRWS TPDRWB を CALL して球と円柱をそれぞれ描いていく。このとき線の幅を 0.01 ポイントづつ原子と円柱の種類で変えているが、これは実際の線の幅を変えることが目的ではなく、Illustrator での操作で同じ線幅の線を選択する機能を生かすための、準備である。Illustrator は 0.01 ポイントの線幅の違いを、違いとして認識するようである。これで球の色を原子の種類別にすることができる。

最後に結晶軸を TPLINE を用いて描いている。結晶構造ではなく、分子を描くのであれば、この部分をとばしてしまえばよい。

プログラム 7.4 は tscsdtmn.f の四番目の部分である。PTIME は使用するシステムに合わせて直す必要がある。FUNC,FUNCD,LFUNC は CALL されることはないので、どんなものでも、関数名だけあっていればよい。

プログラム 7.4

```
      SUBROUTINE CLOCKM(ITIME)
C
C       FOR UNIX
C
      DIMENSION TERY(2)
```



C- CENTERED LATTICE

Cmca D_{2h}^{18}

GROUP	ELEMENTS			ATOMIC POSITION			
1	1	E	X Y Z	0/1 0/1 0/1	0/1 1/2 1/2	1/2 0/1 1/2	1/2 1/2 0/1
2	2	C2X	X -Y -Z	0/1 0/1 0/1	0/1 1/2 1/2	1/2 0/1 1/2	1/2 1/2 0/1
3	3	C2Y	-X Y -Z	0/1 0/1 0/1	1/2 0/1 1/2	1/2 1/2 0/1	0/1 1/2 1/2
4	4	C2Z	-X -Y Z	0/1 0/1 0/1	1/2 0/1 1/2	1/2 1/2 0/1	0/1 1/2 1/2
5	25	IE	-X -Y -Z	0/1 0/1 0/1	0/1 1/2 1/2	1/2 0/1 1/2	1/2 1/2 0/1
6	26	IC2X	-X Y Z	0/1 0/1 0/1	0/1 1/2 1/2	1/2 0/1 1/2	1/2 1/2 0/1
7	27	IC2Y	X -Y Z	0/1 0/1 0/1	1/2 0/1 1/2	1/2 1/2 0/1	0/1 1/2 1/2
8	28	IC2Z	X Y -Z	0/1 0/1 0/1	1/2 0/1 1/2	1/2 1/2 0/1	0/1 1/2 1/2

LATTICE CONSTANTS ARE SET AS

A= 8.52399 B= 14.44658 C= 8.53533

CA= .00000 CB= .00000 CC= .00000

8f $0,y,z \quad 0,\bar{y},\bar{z} \quad 0,y+1/2,\bar{z}+1/2 \quad 0,\bar{y}+1/2,z+1/2$

$y = 0.1525 \qquad z = 0.0785$

図 11: **Ga の結晶構造**

```
C
C     etime for AIX FORTRAN
      CALL ETIME_(TERY)
C     etime for SUN FORTARAN
C     CALL ETIME(TERY)
      ITIME=TERY(1)*1000
      RETURN
      END
      SUBROUTINE PTIME(T)
      REAL*8 T
      CALL CLOCKM(IT)
      T=IT/3600000.0
      RETURN
      END
      FUNCTION FUNC(XX)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION XX(3)
      FUNC=0.0
      RETURN
      END
      FUNCTION FUNCD(XA)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION XA(3)
      FUNCD=1.0
      RETURN
      END
      FUNCTION LFUNC(XA)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/BZPL/KKG(3,20),NNG,IEOH,INDC,IBR,ICUT
      DIMENSION XA(3)
      LFUNC=-1
      RETURN
      END
```

結晶構造の作図プログラム tscsdtmn.f には、さらにサブルーチン TSPPRP がついているが、この部分の説明は TSP p60 の５章の記述と重複するので省略する。プログラムはこの PDF が置いてあるページに他のプログラムと共に置いてある。またこの節で例にした LaCl3,Ga のための fort.1 も同様に置いてある。

入力データ 7.1

```
   4
 1 1   0.54030  2.0
 1 1   0.59991  1.5
 1 1   0.60660  1.0
 1 1   0.61963  0.7
 6.0
 0
 1 1 1 1
 0
 110.0 86.0 0.0 -10.0 100.0 1000.0    size of a(mm),AF,BT,GM,ED,EL
91,15                                 IFL,IPOINT for number on atoms
```

入力データ 7.1 は図 11 の Ga のためのものである。互いに反転対称の位置になる、最近接原子は近くにいるが、第二から第四近接まではそれぞれ二個ずつが、ほとんど同じ距離にある。太さを順次細くして示すようになっている。図 11 には空間群、格子定数、原子位置の情報も併せて示している。この構造は底心直方格子で原子位置が a 面上にあるので、単位胞に４個の原子がそれぞれ３個ずつ見えている。またこの構造は b 軸が長いので AF=86(度) と、大きく回してその長さが見えるように設定したいる。原子がすべて c 面から離れているため、原子を結ぶ円柱の多くが、c 面を横切っている。TPERSP の投影法が上下の方向を、投影された図の上下に忠実にしている特徴がここで生きてくる。つまり下の c 面で終わっている円柱がその真上にある円柱の端につながっているのが見てとれる。右下の角で終わっている円柱の続きは、もちろん左上の角である。入力データ 7.2 は図 10 のためのものである。ここでは IPOINT=0 として球につける番号を省略している。

入力データ 7.2

```
    2
    1 2  0.39455 1.5
    1 2  0.39494 1.5
    13.0  10.0
      3    0
     0 0 2 1
     0
150.0 10.0 0.0 -10.0 100.0 3000.0
92 0
```

TSPACE の本の付録のディスクに入っている TSPACE.F に含まれる SUBROUTINE TSCSDT は TSP p230 の引数の説明と少し違っている。IK(300) で返される値が説明では、「TSCRST の VA の上の番号」となっているが、この SUBROUTINE TSCSDT では原子の種類番号となっている。同じディスクに入ってる tscsdtmn.f を使うのにはこれでよいが、この節で説明した新しい tscsdtmn.f では説明のとおり、「TSCRST の VA の上の番号」でなければならない。このようにした tscsdt.f がこの同じページに置いてあるので入れ替えていただきたい。

8 TPCSEC

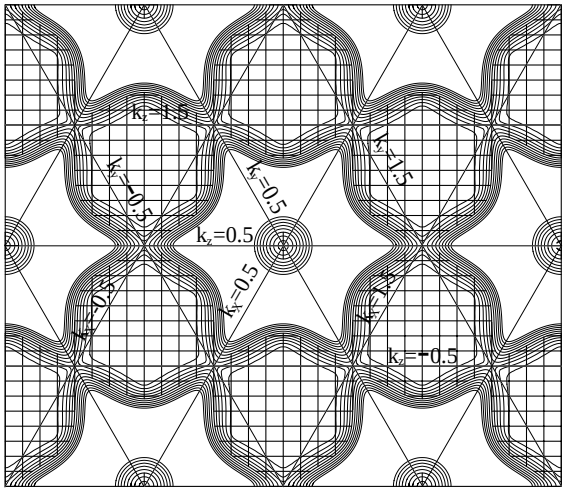


図 12: **Cu のフェルミ面の断面図　L 点を中心の 111 面**

プログラム 8.1 は図 2 のような断面図の作図プログラムである。図の範囲をサブルーチン TPERSP ではなく、TPCSEC であたえる。ここでは断面図をそのまま出力するので投影に必要なパラメーターは取り除いている。断面図はやはり TPSECT を用いて描く。この場合 XC(3),DR(3) で決められる平面が XFU(3,2) で決められる直方体を切った切り口は必ずしも矩形にならない。TPSECT は 3 角形になったり、6 角形になったりする切り口を内包する矩形を作って作図を実行する。この使用法では隠線処理の必要はないので 4 の説明にある IHL をこの使用法専用の IHL=1 にする。しかしこの場合も FUNC,FUNCD,LFUNC はつけて置かなければならない。

入力データ 8.1 が図 12 のものである。この図は Cu のフェルミ面を L 点を中心にして 111 に垂直に切った面の断面図を少し広い範囲で示している。この図に描いてある三角格子と座標の値は AdobeIllustrator

に取り込んでから描き入れてある。また入力データ 8.2 は図 2 のものである。

プログラム 8.1

```
      IMPLICIT REAL*8 (A-H,O-Z)
      EXTERNAL FUNC
      DIMENSION NRECPT(3,30),RECTAX(4,30)
      REAL*8 XFU(3,2),WF(3),EF,EE
      REAL*8 EM,PN,p
      REAL*8 XX(3),XC(3),DR(3)
      common/enrspl/EM(366145),PN(3),p(3),n(3),js
      INTEGER M(3),NN(3),NK(3)
      READ(21,*) NB,IEOH
      DO 10 J=1,NB
      READ(31) IXMAX,NB,NNEE
      READ(31) (EM(K),K=1,NNEE)
      WRITE(6,*) IXMAX,NB,NNEE
   10 CONTINUE
      PN(1)=1.0
      PN(2)=1.0
      PN(3)=1.0
      n(1)=IXMAX-1
      n(2)=n(1)
      n(3)=n(1)
      js=3
      DO 51 I=1,3
      P(I)=Pn(I)/FLOAT(N(I))
   51 CONTINUE
      WRITE(6,*) WWWW
      READ(21,*) (XFU(I,1),I=1,3)
      READ(21,*) (XFU(I,2),I=1,3)
      READ(21,*)  SIZE
      WF(1)=SIZE
      WF(2)=WF(1)
      WF(3)=WF(1)
C
      READ(21,*) (XC(I),I=1,3)
      READ(21,*) (DR(I),I=1,3)
      READ(21,*) EF,DDE,NDE
        write(6,*) EF,DDE,NDE
      READ(21,*) N241,NK2
        write(6,*) XC,DR,EF
        write(6,*) N241,NK2
      READ(21,*) IFL
      CALL AYPSTR(IFL)
      CALL AYORIG(30.0,30.0)
      CALL TPCSEC(XFU,WF)
      CALL TPTRAC(1)
C      IEOH=1
      CALL TPSCHG(IEOH)
      CALL LINEWD(2.0)
        NNX=N241
        NNY=N241
        KKX=NK2
        KKY=NK2
         CALL TPSECT(FUNC,EF,DR,XC,0,1,NNX,NNY,KKX,KKY)
         CALL LINEWD(1.0)
         DO 31 IE=1,NDE
            EE=EF+IE*DDE
         CALL TPSECT(FUNC,EE,DR,XC,1,0,NNX,NNY,KKX,KKY)
            EE=EF-IE*DDE
         CALL TPSECT(FUNC,EE,DR,XC,1,0,NNX,NNY,KKX,KKY)
 31      CONTINUE
      CALL AYPEND
      STOP
      END
```

入力データ 8.1

```
 6 1
```

```
-1.0 -1.0 -1.0
2.0 2.0 2.0
130.0
0.5 0.5 0.5
1.0 1.0 1.0
0.41714 0.02 5
257 8
84
```

入力データ 8.2

```
6 1
0.0 0.0 -1.0
2.0 2.0 1.0
130.0
1.0 1.0 0.0
1.0 1.0 0.0
0.41714 0.01 10
257 16
88
```

9 おわりに

ここで紹介した TPERSP のソース TPERSP.f をはじめ、説明に使われた図を描くためのソース、およびデータは、この .pdf ファイルがある同じページに置いてあるので試用されたい。もちろんコンパイルするときには、tpersp.f だけではなく、TSPACE と AYPLOT もリンクできるようにしなければならない。