

# 立方対称の結晶場レベルの計算プログラム cubicmn.f

柳瀬 章

August 20, 2005

ここでの記述はこのページにある angular.pdf 中の記事に依存している。式の番号とか、表の番号の引用を適宜行うので、あらかじめ用意されたい。

この cubicmn.f は、f 電子の数、と全角運動量 J を指定して立方対称の結晶場の中での固有状態、固有エネルギーを計算するプログラムで、「結晶場中の希土類イオン」のページで紹介している手法の例として示す。angular.pdf の 4.1 節で説明している 4.1 式に現れた縮約行列やクレブシュ=ゴードン係数を計算し、行列要素を求めて、それを対角化して固有値、固有状態を計算するプログラムの一例である。

$O_h, O, T_d$  の対称性を持つ場合には、式 (3.5) に示すように、結晶場は二つのパラメーター  $B_c^4, B_c^6$  で記述できる。このプログラムでは  $B_c^4$  をエネルギーの単位にすることで、結晶場のエネルギーレベルを  $B_c^6$  の関数で表すことにしている。結果の例を Figure 1 と、Figure 2 に示している。

Prog 1 は f 電子数が奇数で J が半整数になる場合のメインプログラムである。CALL 文の 2 番目にある TSTRJM は TSPACE のセットに含まれるように設計されており、O 群だけに対応しているので、最初に TSPACE を IL=1 で CALL している。次の READ 文の NF は f 電子数、JJ は半整数の全角運動量の値 J の 2 倍の整数値である。このプログラムは f 電子数が奇数で、J が半整数の場合を扱っている。f 電子数が偶数の場合は Prog 7 に示している。次の TSTRJM, TSJMDS の説明はこの「結晶場中の希土類イオン」のページにある。

## Prog 1 f 電子数が奇数のときのメインプログラム

```
C-----cubicmn.f-----
      IMPLICIT REAL*8(A-H,O-Z)
      COMPLEX*16 U(100)
      INTEGER KP(100),INS(4,24)
      REAL*8 WB(3),CUB(2)
      INTEGER IREG(10,3),NREG(3)
      CALL TSPACE(1)
      READ(5,*) NF,JJ
      CALL TSTRJM(JJ,U,KP,INS)
      CALL TSJMDS(JJ,U,KP,INS)
      JJA=JJ
      JJB=JJ
      CALL REDMAT(NF,JJA,JJB,WB)
      WRITE(6,*) ' REDUCED MATRIX '
      WRITE(6,610) (WB(I),I=1,3)
610  FORMAT(8F10.5)
      CALL GTNIRG(NREG,IREG,INS,JJ)
      WRITE(6,*) (NREG(IR),IR=1,3)
      WRITE(6,*) ((IREG(N,IR),N=1,NREG(IR)),IR=1,3)
      CUB(1)=1.0D0
      DO 30 ICR=1,21
      CUB(2)=0.1D0*(ICR-11)
      CALL GTEIGEN(NF,JJ,U,KP,INS,NREG,IREG,WB,CUB)
30  CONTINUE
      STOP
      END
```

次に CALL されている REDMAT は、式 (4.1) の縮約行列を計算して、 $k=2,4,6$  の値を WB(3) に入れて返すサブルーチンである。JJA,JJB はそれぞれ (4.1) の  $J, J'$  の 2 倍である。入力引数に L,S が無いが、あたえられた NF にしたがって、フントの規則の状態の L,S を Prog 2 に示すように SUBROUTINE REDMAT の最初

の部分で求めている。プログラム中の変数 JA が L の 2 倍、JS が S の二倍である。また IPRIM(8) のデータは最初の 8 個の素数、IPCML(8,3,3) は angular.pdf の Table 4.2 の中である。

以下このサブルーチンは angular.pdf の 4.1 節の記述をなぞっている、詳しい説明は省略する。(4.5) 式がフントの規則の状態を仮定しているので、このサブルーチンはこの適用の制限を持っている。なお、このサブルーチンは f 電子数が奇数でも、偶数でも共通に使えるように作られている。このため f 電子数が偶数で J が整数の場合でも、JA1,JA2 の引数は 2 倍の整数値を Prog 7 のように使う。

## Prog 2 縮約行列を計算するサブルーチン REDMAT

```

C-----redmat.f-----
      SUBROUTINE REDMAT(NF,JA1,JA2,WB)
      INTEGER IPRIM(8)/2,3,5,7,11,13,17,19/
      INTEGER IPCMPL(8,3,3)/
& 2,-1,-1,1, 0,0,0,0, 1, 0, 0,1,-1,0,0,0, 2,-1,2,1,-1,-1,0,0,
& 1,-3,-1,0, 1,1,0,0, 3,-2, 0,0,-1,1,0,0, 2,-3,3,0,-1,-1,1,0,
& 1,-2,-1,1,-1,1,0,0, 2,-2, 0,1,-3,1,1,0, 2,-2,4,0,-3,-1,1,1/
      INTEGER ISIGN(3,6)/-1,1,-1, -1,-1,1, -1,-1,-1,
&
&          1,1,1, 1,1,-1, 1,-1,1/
      REAL*8 CA,WA,WB(3),WBB
      IF(NF.LT.1.OR.NF.EQ.7.OR.NF.GT.13) THEN
        WRITE(6,*) ' Wrong number of NF=',NF
        RETURN
      END IF
      IF(NF.LE.6) N=NF
      IF(NF.GT.7) N=14-NF
      JA=0
      JS=0
      DO 3 I=1,N
        JA=JA+6-2*(I-1)
        JS=JS+1
3      CONTINUE
      IF(N.LE.3) NN=N
      IF(N.GE.4) NN=7-N
      DO 1 JBB=4,12,4
        JB=JBB
        KK=JBB/4
        WA=1.0D0
        DO 11 II=1,8
          IF(IPCML(II,KK,NN).LT.0) THEN
            WA=WA/(IPRIM(II)**IABS(IPCML(II,KK,NN)))
          ELSE IF(IPCML(II,KK,NN).GT.0) THEN
            WA=WA*(IPRIM(II)**IPCML(II,KK,NN))
          END IF
11      CONTINUE
        WA=SQRT(WA)*ISIGN(KK,N)
        WRITE(6,601) JA/2,JB/2,WA
601     FORMAT('/' L=',I3,' K=',I3,F14.10)
        JJA1=JA1
        JJA2=JA2
        IW=JJA1+JA+JS
        IF(NF.LE.6) THEN
          C      WRITE(6,*) N,JA,JS,J,IW
          IF(MOD(IW,2).NE.0) THEN
            WRITE(6,*) N,JA,JS,J,IW
            STOP 'STOP AT IW'
          END IF
          IW=IW/2
          CALL T6JSYM(CA,JA,JB,JA,JJA1,JS,JJA2,IND,0,J1,J2,J3)
          WBB=WA*CA*SQRT(JJA1+1.0D0)
          IF(MOD(IW,2).EQ.1) WBB=-WBB
          C      WRITE(6,600) WBB
          C 600     FORMAT(F14.10)
        ELSE IF(NF.GT.7) THEN
          IW=JJA1+JA+JS

```

```

C          WRITE(6,*) N,JA,JS,J,IW
          IF(MOD(IW,2).NE.0) THEN
              WRITE(6,*) N,JA,JS,J,IW
              STOP 'STOP AT IW'
          END IF
          IW=IW/2
          CALL T6JSYM(CA,JA,JB,JA,JJA1,JS,JJA2,IND,0,J1,J2,J3)
          WBB=-WA*CA*SQRT(JJA1+1.0D0)
          IF(MOD(IW,2).EQ.1) WBB=-WBB
C          WRITE(6,600) WBB
          END IF
          WB(JBB/4)=WBB
1    CONTINUE
      WRITE(6,610) NF,JJA1,JJA2,(WB(I),I=1,3)
610  FORMAT(' REDUCED MATRIX FOR NF=',I3,
&         ' JA1=',I3,'/2', ' JA2=',I3,'/2',/3F14.10)
      RETURN
      END

```

Prog 1 で次に CALL している GTNIRG は Prog 3 に示すように、TSTRJM の引数 INS の内容を、変換して各既約表現に属する状態の数 NREG(3) と、その代表が 2J+1 個の基底の何処にあるかを IREG(10,3) で返してくれる。

### Prog 3 サブルーチン GTNIRG

```

C SUBROUTINE GTNIRG =====3=====4=====5=====6=====7
C
C    2005/05/06 YANASE
C
C    OUTPUTS OF INS(4,24) FROM TSTRJM ARE RECONSTRUCTED TO
C    NREG(3) AND IREG(3,10)
C
C-----1-----2-----3-----4-----5-----6-----7
C
      SUBROUTINE GTNIRG(NREG,IREG,INS,JJ)
      INTEGER INS(4,24)
      INTEGER IREG(10,3),NREG(3)
      DO 10 IIR=1,3
10     NREG(IIR)=0
      DO 11 I=1,JJ+1
          IF(INS(3,I).NE.1) GO TO 11
          IIR=INS(4,I)
          NREG(IIR)=NREG(IIR)+1
          IREG(NREG(IIR),IIR)=I
11     CONTINUE
      RETURN
      END

```

Prog 1 の最後に CALL している SUBROUTINE GTEIGEN を Prog 4 に示す。このサブルーチンがここで紹介しているプログラムの主要部分である。(3.5) 式の定義にしたがって、あたえられた  $B_c^4, B_c^6$  の値、CUB(1),CUB(2) に相当する行列要素を各既約表現ごとに求め、それを対角化して固有値固有状態を計算する。このサブルーチンの引数は最後の CUB(2) 以外は今までのルーチンの引数である。またこのサブルーチンの引数は全て入力で、計算結果は標準出力 (機番 6 の WRITE 文) へと機番 1 2 への出力である。

DO 12 のループで  $E_{1/2}, E_{5/2}, G_{3/2}$  の 3 種類の既約表現毎に、ハミルトニアン行列 H(10,10) を計算し、それを対角化して固有エネルギー、固有ベクトルを求める。既約表現のベースベクトルが、 $|J, M\rangle$  に係数  $U(I1), U(J1)$  を掛けた形の和で与えられているが、DO 14, DO 145 のように、両側の和を取らなければならない。TSTRJM や、次の Prog 7 で使われる TSTRLM では、射影演算子だけでベースが作れないので直交化の手続きが使われているからである。このため TSPACE の本の p152 の (8.6) の変形を使用することができない。

行列要素の計算は (4.1) 式にほとんど忠実にしたがっているが、分母にある  $\sqrt{2J+1}$  は除いている。(4.2) の分子にある  $\sqrt{2J+1}$  と打ち消すからである。もちろん先に説明した REDMAT でも、分子にあるこの係数を除いている。

既約表現に属する状態が複数あれば対角化のサブルーチン JACOBI が CALL される。今の応用では行列の次元は (3.14), (3.15) に示すようにたかだか3であるので、論理が簡単なヤークービ法を採用している。このサブルーチンは Prog 5 に示している。このプログラムは、N=1 で CALL すると誤動作の可能性があるので Prog 4 では NN=1 を特別に扱っている。プログラムの中の変数 SUME は結晶場のエネルギーレベルの中心が定義によって0になることを使って、計算のチェックを行うために計算している。最後に出力している SUME の値は0にならねばならない。

#### Prog 4 固有エネルギー、固有ベクトルを計算するサブルーチン GTEIGEN

```

SUBROUTINE GTEIGEN(NF, JJ, U, KP, INS, NREG, IREG, WB, CUB)
IMPLICIT REAL*8(A-H, O-Z)
COMPLEX*16 U(100)
INTEGER KP(100), INS(4, 24)
REAL*8 H(10, 10), WA, CA, CB
REAL*8 VR(10, 10), E(10), CUB(2), WB(3)
INTEGER IREG(10, 3), NREG(3)
WRITE(6, 600) NF, JJ
600 FORMAT(/' NF=', I3, ' J=', I5, '/2')
WRITE(12, *) NF, JJ
WRITE(6, *) ' BC4 and BC6'
WRITE(6, 610) CUB(1), CUB(2)
WRITE(12, 610) CUB(1), CUB(2)
SUME=0.0DO
DO 12 IIR=1, 3
WRITE(12, *) IIR, NREG(IIR)
IF(NREG(IIR).EQ.0) GO TO 12
WRITE(6, *) ' IIR=', IIR, ' NUMBER OF STATES', NREG(IIR)
DO 13 J=1, NREG(IIR)
DO 13 I=1, J
WA=0.0DO
DO 14 J1=INS(1, IREG(J, IIR)), INS(2, IREG(J, IIR))
MJ=JJ-2*(KP(J1)-1)
DO 145 I1=INS(1, IREG(I, IIR)), INS(2, IREG(I, IIR))
MI=JJ-2*(KP(I1)-1)
KK=MI-MJ
IF(KK.EQ.0.OR.KK.EQ.8.OR.KK.EQ.-8) THEN
CALL TCGCOF(CA, JJ, 8, JJ, MJ, KK, IND, 0, JJ1, JJ2, JJ3)
CALL TCGCOF(CB, JJ, 12, JJ, MJ, KK, IND, 0, JJ1, JJ2, JJ3)
IF(KK.EQ.0) WA=WA+(CA*WB(2)*CUB(1)+CB*WB(3)*CUB(2))
& *DCONJG(U(I1))*U(J1)
& IF(ABS(KK).EQ.8) WA=WA+(SQRT(5.0DO/14.DO)*CA*WB(2)*CUB(1)
& -SQRT(7.0DO/2.0DO)*CB*WB(3)*CUB(2))
& *DCONJG(U(I1))*U(J1)
END IF
145 CONTINUE
14 CONTINUE
H(J, I)=WA
IF(J.NE.I) H(I, J)=WA
13 CONTINUE
WRITE(6, *) ' HAMILTONIAN'
DO 15 J=1, NREG(IIR)
WRITE(6, 610) (H(I, J), I=1, NREG(IIR))
15 CONTINUE
NN=NREG(IIR)
IF(NN.GT.1) THEN
CALL JACOBI(H, VR, NN)
WRITE(6, *) ' EIGEN VALUE'
WRITE(6, 610) (H(I, I), I=1, NN)
WRITE(12, 610) (H(I, I), I=1, NN)
610 FORMAT(8F10.5)
WRITE(6, *) ' EIGEN VECTOR'
DO 31 I=1, NN
E(I)=H(I, I)
IF(IIR.LE.2) SUME=SUME+E(I)
IF(IIR.EQ.3) SUME=SUME+E(I)*2.0DO

```

```

        WRITE(6,610) (VR(I,J),J=1,NN)
        WRITE(12,610) (VR(I,J),J=1,NN)
31    CONTINUE
    ELSE
        E(1)=H(1,1)
        WRITE(6,*) ' EIGEN VALUE'
        WRITE(6,610) E(1)
        WRITE(12,610) E(1)
        VR(1,1)=1.0D0
        WRITE(6,610) VR(1,1)
        WRITE(12,610) VR(1,1)
        IF(IIR.LE.2) SUME=SUME+E(1)
        IF(IIR.EQ.3) SUME=SUME+E(1)*2.0D0
    END IF
12    CONTINUE
    WRITE(6,*) ' ENERGY SUM'
    WRITE(6,610) SUME
    RETURN
    END

```

### Prog 5 ヤコービ法による、実対称行列の対角化のサブルーチン *JACOBI*

```

SUBROUTINE JACOBI(A,U,N)
REAL*8 A(10,10),U(10,10),DMA,EPS
REAL*8 W,W1,W2,W3,ALF,BET,GAM,S,C
EPS=1.0D-7
DO 11 J=1,N
DO 11 I=1,N
U(I,J)=0.0D0
IF(I.EQ.J) U(I,J)=1.0D0
11 CONTINUE
K=0
12 K=K+1
DMA=0.0D0
DO 13 I=1,N-1
DO 13 J=I+1,N
IF(ABS(A(I,J)).LE.DMA) GO TO 13
DMA=ABS(A(I,J))
IP=I
IQ=J
13 CONTINUE
IF(DMA.LE.EPS) GO TO 17
W1=A(IP,IP)
W2=A(IP,IQ)
W3=A(IQ,IQ)
ALF=-W2
BET=0.5D0*(W1-W3)
GAM=ABS(BET)/SQRT(ALF**2+BET**2)
S=SQRT(0.5D0*(1.0D0-GAM))
IF(ALF*BET.LT.0.0D0) S=-S
C=SQRT(1.0D0-S**2)
DO 14 J=1,N
W=A(IP,J)*C-A(IQ,J)*S
A(IQ,J)=A(IP,J)*S+A(IQ,J)*C
14 A(IP,J)=W
DO 15 J=1,N
A(J,IP)=A(IP,J)
15 A(J,IQ)=A(IQ,J)
W=2.0D0*W2*S*C
A(IP,IP)=W1*C**2+W3*S**2-W
A(IQ,IQ)=W1*S**2+W3*C**2+W
A(IP,IQ)=0.0D0
A(IQ,IP)=0.0D0
DO 16 I=1,N
W=U(I,IP)*C-U(I,IQ)*S
U(I,IQ)=U(I,IP)*S+U(I,IQ)*C

```

```

16 U(I,IP)=W
   GO TO 12
17 CONTINUE
   RETURN
   END

```

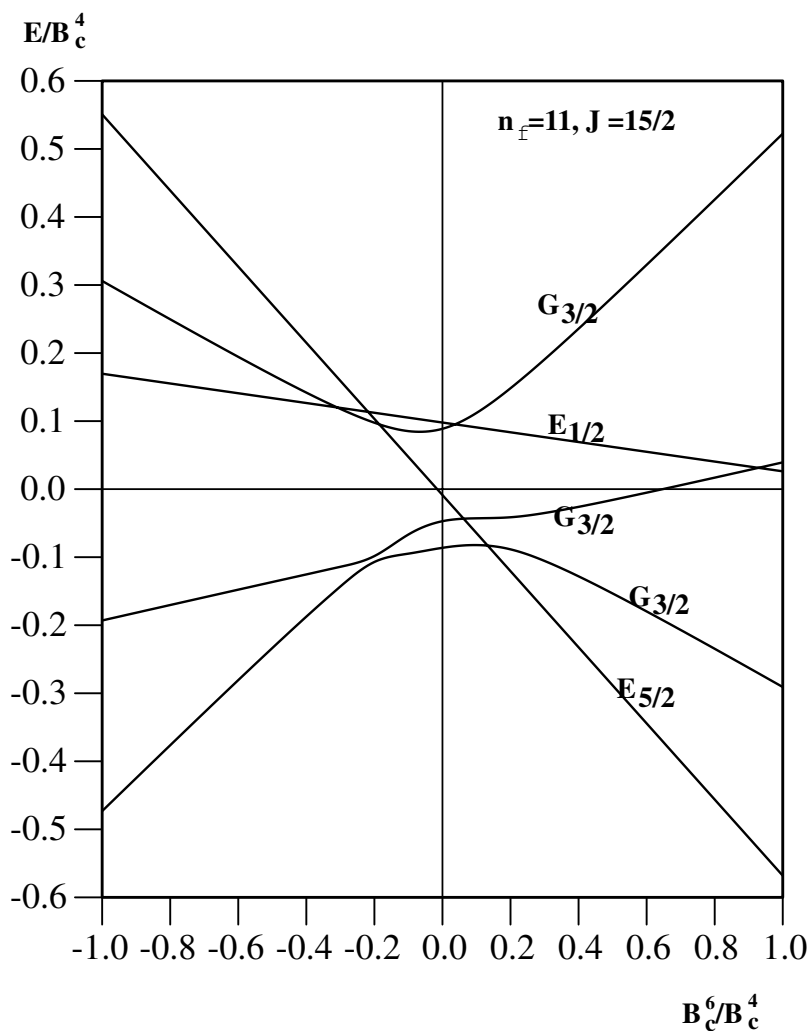


Figure 1: f 電子数が 11 で  $J=15/2$  の 16 重に縮退したレベルが立方対称の結晶場で分裂する様子を示している。 $B_c^4$  をエネルギーの単位にすることで、 $B_c^6$  の関数で示している。angular.pdf の式 (3.5) 参照

Figure 1 に計算結果の一例をグラフで示している。この図は Prog 6 で描いている。このプログラムは AYPLOT を用いている。

DO 21 で 21 組の計算結果をまとめて読み込む。Prog 5 のヤコービ法のプログラムは固有値の大ききでソートしてくれていないので、DO 23 のところで大ききの順にソートしている。

プログラムでは外枠と目盛り、およびグラフを描いている。その他の記号などは、Illustrator で入れている。計算値の内挿には AYBAND が用いている S3N を端条件を 3 次微分 0 にして使用している。この部分は AYBAND についているものと同じだが、Prog 6 の最後に付けてある。

**Prog 6** Prog 1 で作成した出力ファイル *fort.12* を読んで、結果をグラフにするプログラム

```

IMPLICIT REAL*8(A-H,O-Z)
REAL*8 E(9),A(21,9,3),C6(21)
REAL*8 X(21),Y(21),SM(21),XX(201),YY(201)
INTEGER NN(3)
REAL*4 XP,YP,XM,YM,FV
CHARACTER*1 ICAR(3)/'1','2','3'/
AMA=0.0DO
AMI=0.0DO
DO 21 J=1,21
READ(12,*) NF, JJ
READ(12,*) C4, C6(J)
DO 22 IR=1,3
  READ(12,*) IIR, NN(IIR)
  IF(NN(IIR).EQ.0) GO TO 22
  READ(12,*) (E(I), I=1, NN(IIR))
  DO 221 I=1, NN(IIR)
    READ(12,*)
221  CONTINUE
    IF(IR.NE.IIR) GO TO 33
    IF(NN(IIR).GE.2) THEN
      A(J, 1, IR)=E(1)
      IK=1
      DO 23 I=2, NN(IIR)
        DO 48 JK=1, IK
          JJK=IK-JK+1
          IF(A(J, JJK, IR).LT.E(I)) GO TO 49
          A(J, JJK+1, IR)=A(J, JJK, IR)
48      CONTINUE
          JJK=0
49      A(J, JJK+1, IR)=E(I)
          IK=IK+1
23      CONTINUE
        ELSE
          A(J, 1, IR)=E(1)
        END IF
        WRITE(6, 601) J, IIR, NN(IIR), (A(J, II, IR), II=1, NN(IIR))
601  FORMAT(3I5, 9F10.5)
        DO 222 II=1, NN(IIR)
          IF(AMA.LT.A(J, II, IR)) AMA=A(J, II, IR)
          IF(AMI.GT.A(J, II, IR)) AMI=A(J, II, IR)
222  CONTINUE
22  CONTINUE
21  CONTINUE
DO 40 II=1, 201
  XX(II)=-1.0DO+(II-1)*0.01DO
40  CONTINUE
  XM=100.0
  YM=120.0
  CALL AYPSTR(81)
  CALL AYORIG(50.0, 40.0)
  CALL LINEWD(1.2)
  CALL MOVETO(0.0, 0.0)
  CALL LINETO(XM, 0.0)
  CALL LINETO(XM, YM)
  CALL LINETO(0.0, YM)
  CALL LINETO(0.0, 0.0)
  CALL LINEWD(0.75)
  CALL MOVETO(0.5*XM, 0.0)
  CALL LINETO(0.5*XM, YM)
  CALL MOVETO(0.0, 0.5*YM)
  CALL LINETO(XM, 0.5*YM)
  CALL LINEWD(0.8)
DO 51 I=-6, 6
  XP=-4.0
  YP=(I+6)*(YM/12.0)

```

```

FV=I*0.1
CALL MOVETO(XP,YP)
CALL LINETO(0.0,YP)
IF(I.LT.0) THEN
  CALL FWRITE(15,FV,4,1,XP-9.5,YP-1.6,0.0)
ELSE
  CALL FWRITE(15,FV,3,1,XP-8.0,YP-1.6,0.0)
END IF
51 CONTINUE
DO 52 I=-5,5
  XP=(I+5)*(XM/10.0)
  YP=-4.0
  CALL MOVETO(XP,YP)
  CALL LINETO(XP,0.0)
  FV=I*0.2
  IF(I.LT.0) THEN
    CALL FWRITE(15,FV,4,1,XP-4.5,YP-5.0,0.0)
  ELSE
    CALL FWRITE(15,FV,3,1,XP-3.0,YP-5.0,0.0)
  END IF
52 CONTINUE
CALL LINEWD(1.0)
DO 41 IR=1,3
  IF(NN(IR).EQ.0) GO TO 41
DO 42 K=1,NN(IR)
  DO 43 I=1,21
    X(I)=C6(I)
    Y(I)=A(I,K,IR)
    XP=(X(I)+1.0)*(XM/2.0)
    YP=(Y(I)+0.6)*(YM/1.2)
    CALL NRMARK(15,ICAR(IR),XP,YP)
43 CONTINUE
  NNN=21
  MMM=201
  C1=4.0*((Y(2)-Y(1))/(X(2)-X(1)))
  AMU1=2.0DO
  CN=4.0*((Y(NNN)-Y(NNN-1))/(X(NNN)-X(NNN-1)))
  ALMN=2.0DO
  CALL S3N(X,Y,SM,XX,YY,NNN,MMM,C1,CN,AMU1,ALMN)
  XP=(XX(1)+1.0)*(XM/2.0)
  YP=(YY(1)+0.6)*(YM/1.2)
  CALL MOVETO(XP,YP)
  DO 44 I=2,201
    XP=(XX(I)+1.0)*(XM/2.0)
    YP=(YY(I)+0.6)*(YM/1.2)
    CALL LINETO(XP,YP)
44 CONTINUE
42 CONTINUE
41 CONTINUE
CALL AYPEND
33 CONTINUE
STOP
END
SUBROUTINE S3N(X,Y,SM,XX,YY,N,NN,C1,CN,AMU1,ALMN)
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION X(99),Y(99),SM(99),XX(200),YY(200)
DIMENSION H(99),ALM(99),AMU(99),C(99),P(99),Q(99),U(99)
N1=N-1
DO 110 I=2,N
  H(I)=X(I)-X(I-1)
110 CONTINUE
DO 120 I=2,N1
  ALM(I)=H(I+1)/(H(I)+H(I+1))
  AMU(I)=1.0-ALM(I)
120 CONTINUE

```



```

DO 130 I=2,N1
C(I)=3.0*(ALM(I)*(Y(I)-Y(I-1))/H(I)+AMU(I)*(Y(I+1)-Y(I))/H(I+1))
130 CONTINUE
C(1)=C1
C(N)=CN
AMU(1)=AMU1
ALM(N)=ALMN
P(1)=2.0
Q(1)=-AMU(1)/P(1)
U(1)=C(1)/P(1)
DO 140 K=2,N
P(K)=ALM(K)*Q(K-1)+2.0
Q(K)=-AMU(K)/P(K)
U(K)=(C(K)-ALM(K)*U(K-1))/P(K)
140 CONTINUE
SM(N)=U(N)
DO 150 K=1,N1
K1=N1-K+1
SM(K1)=Q(K1)*SM(K1+1)+U(K1)
150 CONTINUE
DO 160 I=1,NN
XXI=XX(I)
DO 170 K=2,N
IF(XXI.GT.X(K)) GO TO 170
J1=K
GO TO 180
170 CONTINUE
180 J=J1-1
SMJ=SM(J)
SMJ1=SM(J1)
YJ=Y(J)
YJ1=Y(J1)
HJ1=H(J1)
XJ1=X(J1)-XXI
XJ=XXI-X(J)
HJ2=HJ1*HJ1
HJ3=HJ2*HJ1
YY(I)=SMJ*XJ1*XJ1*XJ/HJ2-SMJ1*XJ*XJ*XJ1/HJ2+YJ*XJ1*XJ1*(2.0*XJ+
& HJ1)/HJ3+YJ1*XJ*XJ*(2.0*XJ1+HJ1)/HJ3
160 CONTINUE
RETURN
END

```

f 電子数が偶数で J が整数でも、ほとんど同じプログラムで計算できるが、いくらかの変更が必要である。メインプログラムは Prog 7 である。入力する J の値 JJ はもともと整数だからそのまま与える。ここでは既約表現のベースを作るのに TSPACE に標準パッケージに含まれている TSTRLM を使う。このサブルーチンは一電子状態を対象にして作られているが、ここで扱っているような多電子の場合でも使うことができる。ただし、使用するときには Prog 8 で説明するようには少し注意が必要である。次の REDMAT は JJA,JJB が整数、半整数共通に作られているので J の値 JJ を二倍して引数にする。

#### Prog 7 f 電子数が偶数のときのメインプログラム

```

IMPLICIT REAL*8(A-H,0-Z)
COMPLEX*16 U(100)
INTEGER KP(100),INS(4,24)
REAL*8 WB(3),CUB(2)
INTEGER IREG(10,5),NREG(5)
CALL TSPACE(1)
READ(5,*) NF, JJ
CALL TSTRLM(JJ,U,KP,INS)
CALL TSLMDS(JJ,U,KP,INS)
JJA=JJ*2
JJB=JJ*2

```

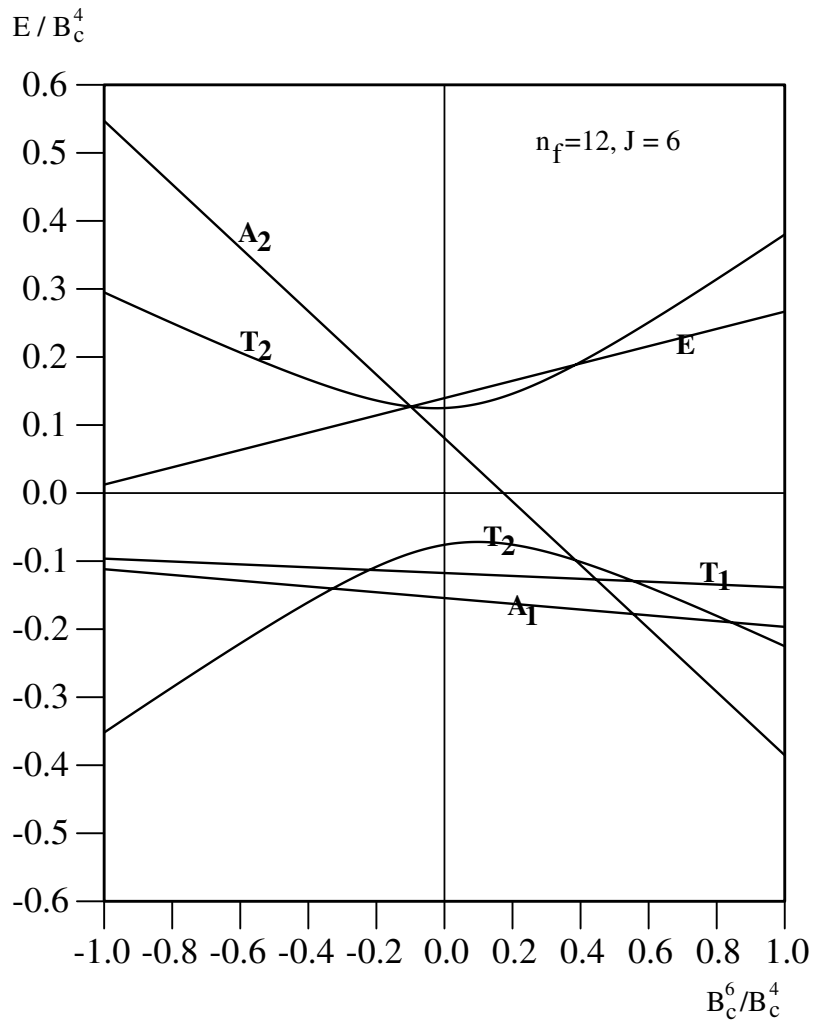


Figure 2: f 電子数が 12 で  $J=6$  の 1 3 重に縮退したレベルが立方対称の結晶場で分裂する様子を示している。 $B_c^4$  をエネルギーの単位にすることで、 $B_c^6$  の関数で示している。angular.pdf の式 (3.5) 参照

```

CALL REDMAT(NF, JJA, JJB, WB)
WRITE(6,*) ' REDUCED MATRIX'
WRITE(6,610) (WB(I), I=1,3)
610  FORMAT(8F10.5)
CALL GTNIRE(NREG, IREG, INS, JJ)
WRITE(6,*) (NREG(IR), IR=1,5)
WRITE(6,*) ((IREG(N, IR), N=1, NREG(IR)), IR=1,5)
CUB(1)=1.0D0
DO 30 ICR=1,21
CUB(2)=0.1D0*(ICR-11)
CALL GTEIGEE(NF, JJA, U, KP, INS, NREG, IREG, WB, CUB)
30  CONTINUE
STOP
END

```

サブルーチン TSTRM は  $IL > 0$  で CALL すると、 $O_h$  群の既約表現のベースを返すように作られている。したがって  $INS(4,n)$  で返される値は 1-10 である。ここでは反転に対する対称性はイオンが置かれた位置の対称性には関係なく、f 電子数が偶数だから偶と決まっており、O 群の既約表現のベースが必要である。 $O_h$  群から O

群への変換は簡単で Prog 8 でしているように  $IIR=(INS(4,I)+1)/2$  で、既約表現番号 1-5 がえられる。

**Prog 8** f 電子数が偶数のときの各既約表現に属する状態数と、その場所を探すサブルーチン *GTNIRE*

```

C SUBROUTINE GTNIRE =====3=====4=====5=====6=====7
C
C      2005/05/06 YANASE
C
C      OUTPUTS OF INS(4,24) FROM TSTRLM ARE RECONSTRUCTED TO
C      NREG(5) AND IERG(10,5)
C
C-----1-----2-----3-----4-----5-----6-----7
C
      SUBROUTINE GTNIRE(NREG,IREG,INS,JJ)
      INTEGER INS(4,24)
      INTEGER IREG(10,5),NREG(5)
      DO 10 IIR=1,5
10      NREG(IIR)=0
      DO 11 I=1,2*JJ+1
          IF(INS(3,I).NE.1) GO TO 11
          IIR=(INS(4,I)+1)/2
          NREG(IIR)=NREG(IIR)+1
          IREG(NREG(IIR),IIR)=I
11      CONTINUE
      RETURN
      END

```

**Prog 9** f 電子数が偶数のときに、固有値固有状態を求めるプログラム

```

      SUBROUTINE GTEIGEE(NF,JJ,U,KP,INS,NREG,IREG,WB,CUB)
      IMPLICIT REAL*8(A-H,O-Z)
      COMPLEX*16 U(100)
      INTEGER KP(100),INS(4,24)
      REAL*8 H(10,10),WA,CA,CB
      REAL*8 VR(10,10),E(10),CUB(2),WB(3)
      INTEGER IREG(10,5),NREG(5)
      WRITE(6,600) NF,JJ/2
600  FORMAT('// NF=',I3,' J=',I5?)
      WRITE(12,*) NF,JJ/2
      WRITE(6,*) ' BC4 and BC6'
      WRITE(6,610) CUB(1),CUB(2)
      WRITE(12,610) CUB(1),CUB(2)
      SUME=0.ODO
      DO 12 IIR=1,5
          WRITE(12,*) IIR,NREG(IIR)
          IF(NREG(IIR).EQ.0) GO TO 12
          WRITE(6,*) ' IIR=',IIR,' NUMBER OF STATES',NREG(IIR)
          DO 13 J=1,NREG(IIR)
              DO 13 I=1,J
                  WA=0.ODO
                  WRITE(6,*) INS(1,IREG(J,IIR)),INS(2,IREG(J,IIR))
                  WRITE(6,*) INS(1,IREG(I,IIR)),INS(2,IREG(I,IIR))
                  DO 14 J1=INS(1,IREG(J,IIR)),INS(2,IREG(J,IIR))
                      MJ=JJ-2*(KP(J1)-1)
                  DO 145 I1=INS(1,IREG(I,IIR)),INS(2,IREG(I,IIR))
                      MI=JJ-2*(KP(I1)-1)
                      KK=MI-MJ
                  WRITE(6,*) ' J1,I1',J1,I1,'MJ,MI',MJ,MI
                  WRITE(6,*) ' U=',U(J1),U(I1)
                  WRITE(6,*) CUB(1),CUB(2)
                  WRITE(6,*) WB(2),WB(3)
                  IF(KK.EQ.0.OR.KK.EQ.8.OR.KK.EQ.-8) THEN
                      CALL TCGCOF(CA,JJ,8,JJ,MJ,KK,IND,0,JJ1,JJ2,JJ3)
                      CALL TCGCOF(CB,JJ,12,JJ,MJ,KK,IND,0,JJ1,JJ2,JJ3)
                      IF(KK.EQ.0) WA=WA+(CA*WB(2)*CUB(1)+CB*WB(3)*CUB(2))

```

```

&          *DCONJG(U(I1))*U(J1)
      IF(ABS(KK).EQ.8) WA=WA+(SQRT(5.0DO/14.DO)*CA*WB(2)*CUB(1)
&          -SQRT(7.0DO/2.0DO)*CB*WB(3)*CUB(2))
&          *DCONJG(U(I1))*U(J1)
      END IF
C      WRITE(6,*) J,I,J1,I1,WA
145     CONTINUE
14     CONTINUE
      H(J,I)=WA
      IF(J.NE.I) H(I,J)=WA
13     CONTINUE
      WRITE(6,*) ' HAMILTONIAN'
      DO 15 J=1,NREG(IIR)
      WRITE(6,610) (H(I,J),I=1,NREG(IIR))
15     CONTINUE
      NN=NREG(IIR)
      IF(NN.GT.1) THEN
        CALL JACOBI(H,VR,NN)
        WRITE(6,*) ' EIGEN VALUE'
        WRITE(6,610) (H(I,I),I=1,NN)
        WRITE(12,610) (H(I,I),I=1,NN)
610     FORMAT(8F10.5)
        WRITE(6,*) ' EIGEN VECTOR'
        DO 31 I=1,NN
          E(I)=H(I,I)
          IF(IIR.LE.2) SUME=SUME+E(I)
          IF(IIR.EQ.3) SUME=SUME+E(I)*2.0DO
          IF(IIR.GE.4) SUME=SUME+E(I)*3.0DO
          WRITE(6,610) (VR(I,J),J=1,NN)
          WRITE(12,610) (VR(I,J),J=1,NN)
31     CONTINUE
      ELSE
        E(1)=H(1,1)
        WRITE(6,*) ' EIGEN VALUE'
        WRITE(6,610) E(1)
        WRITE(12,610) E(1)
        VR(1,1)=1.0DO
        WRITE(6,610) VR(1,1)
        WRITE(12,610) VR(1,1)
        IF(IIR.LE.2) SUME=SUME+E(1)
        IF(IIR.EQ.3) SUME=SUME+E(1)*2.0DO
        IF(IIR.GE.4) SUME=SUME+E(1)*3.0DO
      END IF
12     CONTINUE
      WRITE(6,*) ' ENERGY SUM'
      WRITE(6,610) SUME
      RETURN
      END

```

Prog 10 f 電子数が偶数のときの、結果をグラフにするプログラム

```

IMPLICIT REAL*8(A-H,O-Z)
REAL*8 E(9),A(21,9,5),C6(21)
REAL*8 X(21),Y(21),SM(21),XX(201),YY(201)
INTEGER NN(5)
REAL*4 XP,YP,XM,YM,FV
CHARACTER*1 ICAR(5)/'1','2','3','4','5'/
AMA=0.0DO
AMI=0.0DO
DO 21 J=1,21
  READ(12,*) NF, JJ
  READ(12,*) C4, C6(J)
DO 22 IR=1,5
  READ(12,*) IIR, NN(IIR)
  IF(NN(IIR).EQ.0) GO TO 22

```

```

        READ(12,*) (E(I),I=1,NN(IIR))
        DO 221 I=1,NN(IIR)
            READ(12,*)
221    CONTINUE
        IF(IR.NE.IIR) GO TO 33
        IF(NN(IIR).GE.2) THEN
            A(J,1,IR)=E(1)
            IK=1
            DO 23 I=2,NN(IIR)
                DO 48 JK=1,IK
                    JJK=IK-JK+1
                    IF(A(J,JJK,IR).LT.E(I)) GO TO 49
                    A(J,JJK+1,IR)=A(J,JJK,IR)
48            CONTINUE
                JJK=0
49            A(J,JJK+1,IR)=E(I)
                IK=IK+1
23        CONTINUE
        ELSE
            A(J,1,IR)=E(1)
        END IF
        WRITE(6,601) J,IIR,NN(IIR),(A(J,II,IR),II=1,NN(IIR))
601    FORMAT(3I5,9F10.5)
        DO 222 II=1,NN(IIR)
            IF(AMA.LT.A(J,II,IR)) AMA=A(J,II,IR)
            IF(AMI.GT.A(J,II,IR)) AMI=A(J,II,IR)
222    CONTINUE
22    CONTINUE
21    CONTINUE
        DO 40 II=1,201
            XX(II)=-1.0D0+(II-1)*0.01D0
40    CONTINUE
        XM=100.0
        YM=120.0
        CALL AYPSTR(81)
        CALL AYORIG(50.0,40.0)
        CALL LINEWD(1.2)
        CALL MOVETO(0.0,0.0)
        CALL LINETO(XM,0.0)
        CALL LINETO(XM,YM)
        CALL LINETO(0.0,YM)
        CALL LINETO(0.0,0.0)
        CALL LINEWD(0.75)
        CALL MOVETO(0.5*XM,0.0)
        CALL LINETO(0.5*XM,YM)
        CALL MOVETO(0.0,0.5*YM)
        CALL LINETO(XM,0.5*YM)
        CALL LINEWD(0.8)
        DO 51 I=-6,6
            XP=-4.0
            YP=(I+6)*(YM/12.0)
            FV=I*0.1
            CALL MOVETO(XP,YP)
            CALL LINETO(0.0,YP)
            IF(I.LT.0) THEN
                CALL FWRITE(15,FV,4,1,XP-9.5,YP-1.6,0.0)
            ELSE
                CALL FWRITE(15,FV,3,1,XP-8.0,YP-1.6,0.0)
            END IF
51    CONTINUE
        DO 52 I=-5,5
            XP=(I+5)*(XM/10.0)
            YP=-4.0
            CALL MOVETO(XP,YP)
            CALL LINETO(XP,0.0)

```

```

FV=I*0.2
IF(I.LT.0) THEN
  CALL FWRITE(15,FV,4,1,XP-4.5,YP-5.0,0.0)
ELSE
  CALL FWRITE(15,FV,3,1,XP-3.0,YP-5.0,0.0)
END IF
52 CONTINUE
CALL LINEWD(1.0)
DO 41 IR=1,5
  IF(NN(IR).EQ.0) GO TO 41
DO 42 K=1,NN(IR)
  DO 43 I=1,21
    X(I)=CG(I)
    Y(I)=A(I,K,IR)
    XP=(X(I)+1.0)*(XM/2.0)
    YP=(Y(I)+0.2)*(YM/0.4)
    CALL NRMARK(15,ICAR(IR),XP,YP)
43 CONTINUE
    NNN=21
    MMM=201
    C1=4.0*((Y(2)-Y(1))/(X(2)-X(1)))
    AMU1=2.0DO
    CN=4.0*((Y(NNN)-Y(NNN-1))/(X(NNN)-X(NNN-1)))
    ALMN=2.0DO
    CALL S3N(X,Y,SM,XX,YY,NNN,MMM,C1,CN,AMU1,ALMN)
    XP=(XX(1)+1.0)*(XM/2.0)
    YP=(YY(1)+0.6)*(YM/1.2)
    CALL MOVETO(XP,YP)
    DO 44 I=2,201
      XP=(XX(I)+1.0)*(XM/2.0)
      YP=(YY(I)+0.2)*(YM/0.4)
      CALL LINETO(XP,YP)
44 CONTINUE
42 CONTINUE
41 CONTINUE
  CALL AYPEND
33 CONTINUE
  STOP
  END

```

最後の二つのサブルーチンの説明は、J が半整数のときの説明と重複するので省略した。Prog 10 で使用している S3N は、Prog 6 についているものを使用すればよい。