

Machikaneyama2000 の使用に関するメモ
大阪大学大学院理学研究科物理学専攻
赤井久純

(1) はじめに

Machikaneyama2000 は KKR-CPA 法を用いて置換型合金・混晶の電子状態を密度汎関数法の局所密度近似の枠内で第一原理的に計算するための計算機コードである。CPA(coherent potential approximation)は Soven をはじめとする何人かの人々によって独立にほぼ同時期に考え出された、置換型不規則合金を扱うための近似法であり、統計力学における分子場近似によく似た考え方で不規則系の電子状態を要領よく高い精度で扱うことができる。CPA は最初タイトバインディング模型を念頭に開発されたが、その後、ス波と Soven によって独立にバンド計算の手法である KKR 法 (グリーン関数法) に適用可能であることが見いだされ、KKR-CPA 法として発展した。ただし、KKR-CPA 法は普通のバンド計算とは比較にならないほど大規模な計算が必要とされるために、実際に KKR-CPA 法が第一原理計算として使われるようになったのは 1980 年代になってからである。

KKR-CPA 法については講義ノート

"Korringa-Kohn-Rostoker Method - Brief summary of the theory of scattering -"
by H. Akai,

を参照いただきたい。教科書としては

"Green Functions for Ordered and Disordered Systems"
by A. Gonis
North-Holland, 1994
ISDN 044488009868

をあげておく。

論文としては、歴史的に初めて第一原理計算に応用された例として

H. Akai, J. Phys. Soc. Jpn 51, 468 (1982)

初期の計算手法に関する議論は

H. Akai, J. Phys. Conden. Matter 1, 211 (1989).

全エネルギーや不規則局所磁気モーメントを計算する手法に関しては

H. Akai and P. H. Dederichs, Phys. Rev. B 47, 8793 (1993).

これら以前の上記論文の参考文献中にあげられている。
また、最近の磁性半導体に対する応用例としては

H. Akai, Phys. Rev. Lett. 81, 3002 (1998)

をあげておく。

(2) KKR-CPA コード

KKR-CPA コードは PC, ワークステーション, 大型汎用計算機, スーパーコンピュータのいずれでも動作可能である. 特に最近の PC の発展の結果 PC でも余り不自由を感じないで計算が可能になっている. また, パラレル化やベクトル化も効率良く行うことができ, ベクトル化で 20 倍程度, パラレル化でもノード数に応じて比較的簡単に高速化をはかれる. 物性研のスーパーコンピュータ 10 ノードを使って 200 倍程度の高速化が得られている.

KKR-CPA コードは不規則合金・混晶を想定して作られているが, 規則合金や単体の系の計算も問題なく実行することができる. その場合 CPA は実行されず, 純粋な系のバンド計算が実行される. ただし, 多くのバンド計算で用いられているような対角化の方法にはよらず, グリーン関数を直接計算することによってバンド計算が実行される. したがって, 純粋な系のバンド計算を実行すると同時にその系を母体とした不純物系の計算も実行するというようなことも容易に出来る. 例えば鉄のバンド計算を実行すると同時に鉄中にニッケル不純物の計算を済ませることができる.

計算機に依存するサブルーチンは以下の 2 個である.

unlock

update

これらは, ソースファイル中のコメントに従って, 適当に計算機に適合するように変更 (多くの場合は単にコメントマークの付け直し) する必要がある. いずれも時刻を取得するルーチンであり, 変更はほとんど自明である.

パッケージにはソースコード, 例題としての **input file**, コンパイルするための **makefile** が含まれている. ソースはディレクトリー **source** の下にすべておかれている. 一つのサブルーティンにつき一つのファイルとなっている. ソースは **FORTRAN77** を用いて書かれており, コンパイルは標準的なフォートランコンパイラを用いて **makefile** に従ってコンパイルする. **main program** は **source/specx.f** である. オリジナルには **main program** は **source/spec#.f** であって, #のところには 1, 2, 3...等の数字が入り, この数字を扱うことの出来るユニットセル中の最大原子数に対応させていた. 例えば **source/spec7.f** はユニットセル中に 7 個の原子が含まれる系までを扱うことができる **main program** で一般に最大原子数を決めたとき, その系を扱うことの出来る **spec#.f** は **spec1.f** をもとにして作成していた. しかし現在使用のパッケージでは一律に **specx.f** としている. ユーザーは場合に応じて **specx.f** を手直しして使うことになる.

specx は **main program** であるが, 主要な役割は **dimension** の宣言をすることであり, それ以外には **input data** を読み込むプログラムを起動することと, 実質的な **main program** である **source/spmain.f** を起動することだけである.

(3) データー入力

input はキーボードなどの標準入力からの入力となるが、ファイルを作っておいてパイプ入力するのが普通であろう。例として、ディレクトリー in に in/ymnal, in/fe3pt, in/fe70ni30, in/fe, in/inmn6mn0as0as 等が納められている。in/ymnal にはコメント付きとなっており、一応それだけで理解できる形になっている。input ファイルを作る規則は次の通りである。

- 1) 自由形式であり、フォーマットはない。どこで行変えがなされてもよく、重要なのはデータの順序だけである。
- 2) データの区切りは空白またはカンマである。
- 3) 空白行は読み飛ばされる。
- 4) 先頭が"C", "c" あるいは"#"で始まる行はコメント行である読み飛ばされる。このことはしばしばデータ入力の失敗を引き起こすので注意する必要がある。たとえば炭素に C という名前をつけてデーターとして行の先頭から入力すると、コメントとみなされて読み飛ばされることになる。
- 5) データは省略することができる。ただし、省略したデータに対応して区切りのカンマを入れる必要がある。省略したデータに対してはデフォルト値が与えられる。
- 6) 1セットの入力データのあとに更に入力データのセットを続けておくことが出来る。それぞれのセットは独立した計算として実行される。

7) 以下で説明するキーワードについては大文字はすべて小文字に変換された上で解釈される.

以下に YMnAl 合金の場合について入力データセットの例をあげて説明する. c で始まる行はすべてコメント行で実際には読み飛ばされる.

```
c----- input data -----
c go/ngo/dos/dsp/spc   file name
c-----
  go                    data/ymnal

c fcc/bcc/hcp/sc/...  a  b/a  c/a  alpha  beta  gamma
c-----
      fcc              14.0  1.0  1.0, , ,

c edelt width nrl/sra mjw/vbh/vwn mag/nmag/-mag/rvrs/kick  init/1st/2nd
c-----
  0.001    1.    nrl    mjw    nmag    2nd

c quit/update  0/1/2/./t/l/m/h/u  iteration  mixing
c-----
      update      t      40      0.024

c number of type
c-----
      2

c type  components  rtin  field  l_max  Z  concentration
c-----
      Y      1      0      0.0    2    39    100
      MnAl   2      0      0.0    2    25    95
                                   13    5

c number of atoms
c-----
      6

c position                type
c-----
0.125a  0.125b  0.125c  Y
0.875a  0.875b  0.875c  Y
0.5a    0.5b    0.5c    MnAl
0.0a    0.5b    0.5c    MnAl
0.5a    0.0b    0.5c    MnAl
0.5a    0.5b    0.0c    MnAl

c crystal axis system is also possible, e.g. 0.5a, 0.5b, 0.5c.
```

1) 最初のデータは go, ngo, dos, dsp, spc のいずれかで、以下のデータに対する処理の種類を指示する。それぞれの意味は次の通りである。

go 通常の self-consistent な計算を実行する。

ngo 何もしない。すなわち、続く 1 セットに相当する一連の入力データは単に読み飛ばされる。

dos self-consistent な反復計算をしないで、最新のポテンシャルデータを用いて状態密度が計算される。したがってあらかじめ通常の self-consistent な計算を完了しておかなければ意味のある結果は得られない。

dsp 途中の結果が出力されず。計算結果がおかしいときなどにチェックのために使用。

spc 通常のバンド計算の dispersion relation に相当するものは、不規則合金では Bloch spectral function である。dispersion relation が与えられた k ベクトルに対するエネルギー固有値に対応したデルタ関数であるのに対して、不規則性による散乱によってエネルギー固有状態が寿命をもつために幅をもつようになる。そのような情報を持った Bloch spectral function を出力する。dos と同様に self-consistent な計算を行わずに最新のポテンシャルデータを用いて計算される。

2) このデータはポテンシャルデータを入出力するファイルを指定する。このファイルは実際に存在すればそのファイルが用いられるし、存在しなければ新たに作られる。

3) 1 4 個のブラベー格子の一つを指定する。(1)fcc (2)bcc (3)hcp (4)sc (5)bct (fct)
(6)simple tetragonal (7)face centered orthorhombic (8)body centered orthorhombic
(9)base centered orthorhombic (10)simple orthorhombic
(11)base centered monoclinic (12)simple monoclinic
(13)triclinic (14)rhombic (trigonal) (15)fct (bct)

に対してそれぞれ、fcc, bcc, hcp (hex), sc, bct, st, fco, bco, bso, so, bsm, sm,

trc, rhb (trg), fct のように指定する。(5)bct と(15)fct は同じものであるが、fcc と bcc のどちらにより近いかということで fct と bct を使い分けることもできる。hcp と hex, rhb と trg はまったく同じものである。

4) ~ 9) 格子定数を指定する。a, c/a, b/a, alpha, beta, gamma を指定するが、たとえば立方晶系ならば強制的に $c/a=b/a=1$, $\alpha=\beta=\gamma=90$ にとられる。実際に基本格子ベクトルと逆格子ベクトルがどのように作られるかはサブルーチン prmvec をみればわかる。また、c/a, b/a などは多くの場合省略することができ、その場合には省略の規則に従って該当する数だけのカンマが入力される。また、a を省略するかあるいは $a=0$ とすると適当な理論値がまた、a として 1 0 0 0 0 より大きい値を与えると実験値が用いられる。

1 0), 1 1) delta と width が入力される。delta の意味はフェルミエネルギーに付与される微小な虚数部分である。本プログラムパッケージで計算されているのは基本的にグリーン関数である。グリーン関数はその定義から複素エネルギー面上の正の実軸はブランチカットになっており、グリーン関数は定義されない。複素平面状の虚数部分の正の側から実軸に近づいた極限として実エネルギーでのグリーン関数が定義されるが、数値計算上は有限の虚数部分を与えて計算するが、そのときの虚数部分の大きさである。普通は 0.001 程度の値を与えて計算される。なお、フェルミエネルギー以外のエネルギーについては一般の複素エネルギーを用いて計算がなされており、当然大きな虚数部分をふくんでいる。また、width の意味はグリーン関数が計算されるエネルギー領域の広さである。たとえば、width=1 であればフェルミエネルギー以下 1Ry のところからフェルミエネルギーまでが計算される。width の大きさは価電子帯を完全にカバーするものでなければならない。価電子帯の幅がわからない場合には大きめの width を用いて荒い計算をしておき、状態密度を出力してその結果から価電子帯の幅を見積もり、適切な width を選ぶようにすれば良い。

1 2) `nrl` または `sra` というオプションを指定する. `nrl` は非相対論的計算を実行, `sra` はスカラー相対論的計算 (すなわちスピン軌道相互作用を落とした相対論的計算) を実行.

1 3) `mjw`, `vbh`, `vwn` の一つを指定. いずれも密度汎関数法の局所近似の parametrization の方法で, `mjw` は Moruzzi, Janak and Williams によるもの, `vbh` は von Barth and Hedin によるもの, `vwn` は Vosko, Wilk and Nusair によるものである. 実際にはこれ以外に GGA にも対応しており指定することができるが, その動作については保証していない. 詳しくはサブルーチン `potenv` のコメントおよび中身をみればわかるようになっている.

1 4) `mag`, `nmag`, `-mag`, `rvrs`, `kick` のいずれかを指定できる. `mag` は磁性をもつ解を得たい場合に指定. 強磁性や反強磁性やスピングラス, 非磁性母体中の磁性不純物などを扱うことができる. `nmag` は非磁性解を得たいときに使用. 非磁性の物質や高温で `nonmagnetic` になっている状態の計算やテスト計算に用いる. 計算時間は半分ですませることができる. `-mag` と `rvrs` は同じものであり, 計算の結果や途中で, 磁化が負になった場合 (それでも一向に差し支えないわけであるが) それを正にしたいときに指定すると磁化の符号を逆転することができる. 非磁性状態で計算されたポテンシャルデータを用いて磁性状態を計算しようとする場合, オプションを `nmag` から `mag` に変えれば良いわけであるが, それだけでは `self-consistent` な計算を続けても非磁性状態の結果が求まってしまう (非磁性状態はどんな場合でも解のひとつになり得るから). スピン空間でも対称性をいったん壊してやらなければ磁性状態の解へと移ってくれない. このようなときに `kick` というオプションを用いれば人為的に対称性を壊し, 磁性状態へと移行することができる.

1 5) `init`, `1st`, `2nd` のいずれかを指定する. ポテンシャルデータは通常 2 組保存される. 最初の 1 組 (レコード) は二番目のレコードを作り出すための初期値として用いられたポテンシャルデータが収められている. このように二組のポテンシャルデータを保有することによって, 最新のデータと一つ前のデータを参照することができる. これは, たとえば一連の計算によって求まっているポテンシャルデータを用いて, さらに `self-consistent` 計算を行い精度を上げようとしたが, 結果が思わしくないで, 一つ前の結果に戻りたいなどというときに用いることができる. `2nd` を指定すると最新のデータを, `1st` を指定するとそれより一つ前のデータを初期ポテンシャルデータと用いて計算することになる. また, `init` を指定すると保存されたポテンシャルデータを用いずに, 原子計算から初期ポテンシャルを作り出して, 計算をはじめ. 省略すると `2nd` が仮定されるが, ポテンシャルデータが存在しない, あるいはファイル自身が存在しない場合には強制的に `init` として計算される.

1 6) `quit`, `update` のいずれかを指定できる. `update` を指定すると計算によって得られたポテンシャルデータでファイルのレコードを更新する. 更新の仕方は, ファイル上の 2 番目のレコードで 1 番目のレコードを置き換え, 計算された最新のポテンシャルデータが 2 番目のデータとなる. `quit` を指定した場合にはファイルの書き換えは一切起こらない. 1) で `dos`, `spc` を指定した場合には, たとえここで `update` を指定していても強制的に `quit` に置き換えられる.

1 7) ブリルアン域内の積分に用いられる k ベクトルの数に関係したオプションを指定する. 整数もしくは `t`, `l`, `m`, `h`, `u` のいずれかのデータを与える. 整数の場合にはブリルアン域の $(0,0,0)$ から $(1,0,0)$ までの分割数を与える. `0` を用いると $(0,0,0)$ の 1 点のみが用いられる. また, `t` は `test` の意味で `0` を指定したのと同じである. `l`, `m`, `h`, `u` はそれぞれ, `low`, `medium`, `high`, `ultra high quality` を意味し, この順番に積分に用いられる k ベクトルの数は増加する. `l`, `m`, `h`, `u` のそれぞれがどのような整数値に相当するかはブラベー格子によるが, サブルーチン `nfqlty` に与えられている (かなりいいかげんな値).

1 8) self-consistent な反復計算の最大反復数を与える。たとえば40とすれば40回の反復計算の後計算は停止する。それ以前に self-consistent な収束条件を満たせばその時点で停止する。収束条件はメインプログラムの中で与えられており (data 文中の tol の値で $1d-4$ ~ $1d-6$ ぐらいが適当) 適当に変更することができる。一般には結果をみながら、収束しているようであるならば、収束条件を満たしていなくても十分であり、収束条件はあくまで単なる目安である。

1 9) 反復計算のためのいわゆる mixing parameter である。チェビシェフ加速による反復計算を行っているために通常の mixing parameter ではないが、大きいほどポテンシャル更新に程度が大きいと考えて良い。収束する限りにおいては大きい方が収束が早い。通常の反復計算と同じで、大きすぎると容易に発散する。最適な大きさは問題に依存するが、経験上は0.01~0.03程度が良いようである。困難な問題の場合(収束が難しい)0.004程度あるいはそれ以下が用いられる。

2 0) これ以降はユニットセルに含まれる原子の情報である。まず最初に与えるデータは含まれるサイトの種類の数である。たとえば YMn_0.95Al_0.05 の場合、Mn と Al は互いに固溶しておりこれは Mn が95%と Al が5%からなる一つのサイトとみなせる。また Y は単独で一つのサイトを作っている。従って、この場合にサイトの数は二個である。

2 1) ~2 7) これらのデータはサイトの種類の数だけ繰り返して与えられる。まずサイトに任意の名前を与える。ここでは1番目のサイトに対して Y という名前をつけている。次に、このサイトはただ一つの原子 Y からなるので、component の数1を与えている。rtin はマフィンティン球の半径であるが、特に指定する必要がなければ0を与えておけば touching radius になるようにプログラムが自動生成する。また field は外部磁場を与える場合で、磁場によるゼーマン分裂の大きさを Ry 単位で与える。l_max はそのサイトのポテンシャルによる散乱が $l=0$ から $l=l_{\max}$ まで考慮されることを意味する。たとえば $l_{\max}=2$ ならば d 散乱までが考慮され、それ以上の部分波に対する散乱はない(すなわち平面波と同じになる)として取り扱うことを意味する。Zはそのサイトを構成する原子の原子番号でその次の concentration とペアで指定する。Yサイトを構成するのはただ一つの原子 Y であるから Y の原子番号39と concentration 100を与えている。ただし concentration はサイト内で規格化されて使われるので、この場合にはどのような数字を与えても結果はおなじである(すなわち規格化されて最終的に数字1に変換される)。これで1番目のサイトに関するデータはおしまいである。次に二番目のサイトであるが、このサイトには MnAl という名前を与えた。component の数は Mn 原子と Al 原子があるので2である。rtin=0, field=0としている。また構成原子として Mn である $Z=25$ と Al である $Z=13$ を与えている。それぞれの concentration は95%と5%であるので、95と5という数字をあたえているが、これは0.95と0.05などと書いても結果は規格化されるので同じである。

2 8) サイトの種類に関するデータをすべて入力すると次はそれらのサイトの位置に関する情報を与える。まず、ユニットセル内に存在するサイトの数を入力する。同じ種類のサイトであっても位置が違えば異なった位置として勘定しなければならない。今の場合ユニットセルあたり6個の違った位置があるので数字6を与える。

2 9) ~ 3 2) 異なった位置のそれぞれについての情報を与える. 最初に位置を与える. 3通りの与え方があり, 今の場合に様に $0.125a\ 0.125b\ 0.125c$ のように与える方法, $0.5x\ 0.5y\ 0.5z$ のように x, y, z が数字のあとにつく与え方, およびとただ単に数字 (たとえば $0.5\ 0.5\ 0.5$ のように) を与える方法がある. 数字の後に a, b, c がついて いる場合にはそれは基本ベクトル a, b, c の数値倍と解釈される. 数字の後に x, y, z が ついている場合には x, y, z 方向の単位ベクトルの数値倍と解釈される. また単なる数値 の場合には1番目, 2番目, 3番目の数字に対して x, y, z が省略されているものと解釈 される. 実際に指定される位置はベクトル $0.125a+0.125b+0.125c$ あるいは $0.5x+0.5y+0.5z$ などとなる. また, これらは混在してよい. たとえば, $0.5\ 0.5y\ 0.125c$ などの指定が 可能である. また, a, b, c, x, y, z が単独で現れた場合にはアルファベットの前に 数字1があるものと解釈される. これらの位置の指定のあとにそのサイトの種類を指定する 文字が与えられる. たとえば, いまの場合6個の異なった位置があるが, 1番目と2番目 の位置はサイト Y であり, 位置3~6はサイト MnAl である. 一般には同じ種類のサイトが ユニットセルの中に複数回現れて良い. また, 2 1) ~ 2 7) で定義していないサイトが ここに現れてはならない.

一番最初のデータ 1) で `spc` 以外を指定した場合は以上で1セットのデータの入力が終わ りである. このようなデータのセットは任意個続けることができる. つまり, サイトの 位置の入力のあとから, 再び `go` などではじまるデータを置いて良い. 計算はデータのセット がなくなるまで連続して行なわれる.

`spc` を指定した場合には例外で, この後に Bloch spectral function を計算したい k 点 (k ベクトル) を任意個数指定する. たとえば

```
0 0 0
0.1 0 0
0.2 0 0
0.3 0 0
0.4 0 0
0.5 0 0
```

のように入力するとガンマ点から X 点までの k 点について 0.1 間隔で spectral function が 計算される. このような k ベクトルのデータがなくなるまで (つまり file の最後が検出さ れるか次の計算の始まりを示す `go` などのデータが検出されるまで) 計算される.

以上が YMnAl の場合であるが, `inmn6mn0as0as` の場合も input file の意味を探ることは 容易である. ここでは原子番号として $Z=0$ の指定があるが, これは原子空孔を意味する. ダイヤモンド構造や zincblende 構造の結晶の場合オープン構造であり, 計算の精度を あげるために原子の無い位置にも空孔を入れているが, 必ずしも必要なわけではない.

もっとも単純な例として純粋な鉄の場合を以下に示しておく。この場合には不規則合金出ないために、自動的に cpa の部分ははずされて計算するために計算時間は合金の場合に比べてかなり高速になる。コメントがはずされているためにコンパクトになっているが、本質的な部分は変わり無い。

```
go data/fe bcc 5.27,,,,,
0.001 1.0 nrl mjw mag 2nd
update 6 80 0.024
1
Fe 1 0 0 26 2 100
1
0 0 0 Fe
```

(3) 出力

出力の例がディレクトリ-out に収められている。上の例 in/fe を用いて計算された結果でそれぞれの機械でコンパイルした実行ファイルを用いて得られた結果と比較できるようになっている。結果は 80 回の反復計算の後 no-convergence で終了しているがこれは問題ない。反復の経過をみればわかるとおり、磁化や全エネルギーは 60 回程度の反復計算で実用上完全に収束している。エネルギー値、磁化等はユニットセルあたりでエネルギーはすべて Ry 単位、磁化はボーアマグネトン単位にしている。その他、各原子のげマフィンティン球内に含まれる角運動量別の電子数や球内の磁化、超微細磁場やアイソマーシフトに相当する原子核上での電子密度、コア電子のエネルギー準位などが出力される。

(4) メインプログラムの生成について

spec1.f から任意の spec#.f を生成する方法について説明する。specx.f のパラメータ文の変更を行う。変更するパラメータは

parameter

```
& (natmmx=1, ncmpmx=1, msizemx=25, mxlmx=4, nk1x=550, nk3x=21,
& msex=201, ngmx=15, nrpmx=250, ngpmx=250, npmx=200, msr=400)
```

の内で、natmmx, ncmpmx, msizemx であり、natmmx がユニットセル内の異なった位置の数に対応する。たとえば、先の YMnAl では 2 8) で指定した 6 が natmmx より小さくなくてはならない。また、ncmpmx は 2 1) ~ 2 7) に現れる原子の種類の数に対応する。たとえば、いまの場合は Y と Mn と Al があらわれるので、ncmpmx は 3 あるいはそれ以上であれば良い。注意しなければならないのはたとえ同じ原子であっても違う種類のサイトに扱われる場合はちがう原子として勘定しなければならない。また、同一の種類 of サイトと構成する場合でも異なった原子と考えなければならない場合がある。たとえばスピングラスで、磁気モーメントが上を向いた Mn 原子と下を向いた Mn 原子を区別して考えるときで、inmn6mn0as0as はその例である。sizemx はユニットセル内の異なった位置のそれぞれで考慮される部分波散乱の数をすべて足し合わせた和である。たとえばすべての位置で d 波散乱 (部分波の数は $3^2 = 9$ 個) まで考慮する場合には $9 \times 6 = 54$ より大きければ良い。

上記のパラメータの値は最大値を与えるもので、たとえば上記の例で与えられる **main program** を用いればユニットセルに6個の異なった位置と3個の異なった成分原子をもつ **YMnAl** を扱うことも1個の位置と1個の成分原子しかもたない **Fe** を扱うことも可能である。しかし、上記のパラメータが大きいほど必要とするメモリーは大きくなるので、大きなプログラムですべてをまかなうのはリソースの消費につながる場合がある。

これら以外に場合に応じて調整を要するパラメータは **nk1x**, **nk3x**, **mxl** である。**nk1x+nk3x** はブリルアン域内の積分に用いられる **k** 点の数の上限を与えている。上の例では **581** 点まで **k** 点の数を増やすことができる。一般にはユニットセル内の異なった位置の数が増えれば **k** 点の数を減らすことができる（また、そうしなければ計算時間が増えて計算が困難になる）。たとえば、ユニットセルに6個の異なった位置が含まれる場合には **10~40** 点程度で十分な結果が出せる。テスト計算ならば数点でもかまわない。**nk3x** は **Bloch spectral function** を計算する **k** 点の数の対応するが、普通 **1** 程度で十分である。**mxl** は計算に用いる最大の部分波散乱角運動量+1 である。**s, p, d** 状態まで計算する場合には **mxl=3**, **f** 状態まで計算する場合には **mxl=4** を用いる。希土類やアクチナイドを除いて定性的な議論には **mxl=3** で十分であろう。

data 文で変更の可能性があるものは、**mse0** と **tol** である。現在 **mse0=35** になっている。これはエネルギーに関する複素積分を実行する際の複素平面状のメッシュの数に対応するが、高精度の計算が必要な場合には適宜増やすことが可能である。もちろんそれに比例して計算時間も増加する。また、**tol** は **self-consistency** の基準を与えるパラメータで **1-d6** は電荷密度の **rms (root mean square) error** が **100** 万分の **1** という意味である。出力に現れる **err** はその常用対数である。

上のようなサイトの位置の個数に応じたプログラムの変更を行ったときは、**program** 名を

program spec6

のように付け替えを行って、それを **source/spec6.f** という名前をつけて別のファイルとして **save** しても良い（本来そのようにしていた）。また、そのとき同時に **makfile** の1行目の

program = specx

を変更して

program = spec6

のように変更して **make** する。その結果 **spec6** のような実行ファイルが生成される。