Installing and Running AKAI KKR

Gerhard H. Fecher
Johannes Gutenberg - Universität, Mainz
Institut für anorganische Chemie und analytische Chemie
55099 Mainz, Germany
e-mail: fecher@uni-mainz.de
homepage: www.ghfecher.de

April 22, 2007

Abstract

A short description *how to run* the AKAI KKR program specx is given.

1 Installing the AKAI KKR program specx

It is assumed that you registered with H. Akai and downloaded the package like described in his e-mail reply.
In the following, it is further assumed that the downloaded package is:
cmd3.tar.gz.

1.1 Installing specx

Please, be sure to read **and** understand the manuals of your operating system and fortran compiler.
Below it is described shortly how to install specx under LINUX or Windows.
You may ask *why to install the program under Windows* ? The answer is: The author likes the debugging features of the Compaq HP (formerly Digital) Visual Fortran graphical users interface. This allows to run and debug programs directly from the GUI. Together with the capabilities of the built in editor, debugging is very fast and most of the times easier then using a series of programs from the command prompt.
For example:
1) The LINUX linker, as used by all Linux Fortran compilers, does not detect if you exceed the maximum size of an array (see next sections), but the Windows linker does.
2) The Intel fortran debugger idb81, does not find the reason for
* segmentation faults **, or ** adress errors **,
in most cases it just crashes. Seemingly, the debug option of Intel ifort 8.1 does not work correctly at all. Newer versions are not better in that case.

1.1.1 Limitations

All Fortran compilers mentioned below allow a maximum of 2.1 Gbyte for array dimensions. That is, for an array $a(n)$ the limit is $n_{max} = (2^{31} - 1)$ byte independent of the type of a. This limits the maximum dimension of the working arrays used by the program (see section: Internal parameter).
Static linking may not work using Intel ifort 8.x compilers. This it at least suggested by Intel. However, the athor has the bad experience that programs may work correctly only if linked statically but not if linked dynamically. Actually, there are obviously some conflicts between Intel and Linux libraries.
The Portland group compiler needs always dynamic linking.

1.1.2 Installing under LINUX

Make sure you have access to the directories where you like to put the files and program, and that the necessary utilities to unpack files and a Fortran compiler (f77) is installed on your system.
After download of the compressed files, copy the archive file into a working directory (e.g.: /../mydirectory, where /../ assigns the full path to the directory, e.g.: /home/user/). Then unpack the source code and other files and compile the program using make, like shown in the following:
> cd /../mydirectory
> gunzip cmd3.tar.gz
> tar xvf cmd3.tar
> cd cpa2002v006d
> make
This will create the executable: *specx*.
Note: The makefile coming with the package uses f77 as Fortran compiler. Usually, the Gnu g77 Fortran compiler is installed as standard under LINUX and used as f77. It is recommended, however, to use a better optimizing compiler. In that case the makefile should be changed to account for the compiler and to revise the compiler options.
The default makefile from cmd3 will put specx into ../cpa2002v006d or similar, if you did not rename it.
After compilation and linking, you may put specx into a directory for executables, in case not already done, create one first like follows (e.g.: ../ may be: /usr/local/bin):
> mkdir ../bin
> cp specx ../bin
Set the environment variables correctly, in case you like to make specx accessible for other users.

1.1.3 Installing under Windows

Create a new directory for the files. Use an unpacker (e.g.: WinRar or WinZip) to unpack cmd3.tar.gz into that directory. Thereby, make sure that the internal directory structure is retained.

a) If you have Compaq Visual Fortran (or the older Version from Digital) installed then start the developer studio. Create a new, empty project as *Fortran Console Application* named specx in the sub-folder ..\cpa2002v006d. Add all files (*.f) from ..\cpa2002v006d\source into the workspace folder *specx files, source files*. Change the active configuration to *release* and set the appropriate options of the project settings. Set the path where to put the executable (otherwise the standard setting of VF will be ..\release). Finally, build specx.
b) Proceed the same way for the Intel Fortran compiler ifort together with the Microsoft Visual program development environment.
This should create the executable: *specx.exe*.
Put the executable into an appropriate directory and set the environment variables correctly, so that specx is accessible from different directories and for other users.

1.2 Operating systems and compilers

The Akai-KKR program specx was successfully compiled by the author using the following operating systems and Fortran compilers:

System	Version	Compiler
Linux	SUSE 8.0 - 9.1	Gnu g77
		Intel ifc 6.0 ... ifort 8.1
		Portland Group pgf77, pgf90; 5.1
	SUSE 9.3	Intel ifort 9.0
Windows	Win 98	Compaq VF 6.6C
	Win XP	Compaq VF 6.6C
		Intel ifort 8.1

Large arrays (> 2.1Gbyte) are not yet tested for Intel ifort 9.0, but should principally be possible on Intel processors with EMT64 extensions or on xx64 processors of AMD. Supposed one has enough RAM installed.
In case of troubles and for comparison, modified make-files and project-files for the different environments are available from the authors web page. Note that such files may strongly depend on the used system and therefore may not work on other computers.

2 Running the AKAI-KKR program specx

The directories for input, output, etc. are built during the installation of cmd3. (Their use is described in one of the files of the cmd3 package written in Japanese.)
However, one may like to use some own path for the files. In that case, the easiest way is to run specx at the command prompt from the directory where you have the input file (named e.g.: infile) as:
> specx < infile > outfile

If the directory containing the specx executable is not in the path environment then use:
> /../specx < infile > outfile
where /../ may be the full path to specs, e.g.: /usr/local/bin/, or wherever you did put it.
Note The path for DOS or Windows operating systems is written like: C:\usr\local\bin\, or similar.
Using the Linux operating system, specx may also be run using the batch command, allowing several calculations being started at once and then being executed successively. The command may look like (please check your version of Linux for the correct use of the batch command and its options using batch -h or similar):
> batch -f calc.job
where the file calc.job may contain at least the line:
/../specx < infile > outfile
or you may specify a path to the directory where to find input and to put the output, e.g.:
cd /home/xyz/akai-kkr/system
/usr/local/bin/specx < infile > outfile
You may also run specx from the XBand graphical user interface provided by H. Ebert. In that case you may have to rename specx and put it into an appropriate directory (see XBand manual) so that XBand can find it. However, the input file prepared by XBand needs some cosmetics in order to be compatible to the latest version of specx.

3 Input for the AKAI-KKR program specx

Atomic units are used: all energies are in Rydberg ($1Ry = 13.606eV$) and lengths are in multiples of Bohr's radius ($a_{0B} = 0.5292$Å).
The following paragraphs describe the input data that are read from infile:

3.1 General

calctyp	process to be executed (go, etc..).		ids
	go	scf calculation: make self consistency,	0
	log	like go but writes to .log file,	0
	dos	calculate density of states,	1
	dsp	display the result without any iteration process,	2
	mcd	magnetic X-ray dichroism (same as xmd),	3
	spc	Bloch spectral function,	4
	fsm	fixed spin moment procedure used.	5

file file name used for input and output of potential.

Note: For calctyp = mcd or xmd, the K-edge XMCD is calculated only when p valence states are available, and only with reltyp = nrlls or srals (see below).

3.2 Lattice type and parameter

brvtyp	type of the bravais lattice		ibrav
	fcc	face centered cubic,	1
	bcc	body centered cubic,	2
	hcp	hexagonal closed packed,	3
	sc	simple cubic,	4
	bct	centered tetragonal,	5
	st	simple tetragonal,	6
	fco	face centered orthorhombic,	7
	bco	body centered orthorhombic,	8
	bco	base centered orthorhombic,	9
	so	simple orthorhombic,	10
	bsm	base centered monoclinic,	11
	sm	simple monoclinic,	12
	trc	triclinic,	13
	rhb,	trg rhombohedral trigonal,	14
	fct	face centered tetragonal,	15
	hex	= hcp,	3
	aux	primitive vectors.	16, 17

Note: For practical reason, fct and bct are treated differently. brvtyp = aux means that the primitive vectors will be given as subsequent data. In this case, the input data for c/a, b/a, alpha, beta, and gamma should not appear but just be skipped.
The following input defines the crystal lattice and depends on the setting of brvtyp.

if brvtyp = aux :

r 3 primitive vectors.
x_1, y_1, z_1
x_2, y_2, z_2
x_3, y_3, z_3
these vectors should be input only when brvtyp = aux is specified.

a lattice parameter in atomic units.

if brvtyp $\neq$ aux :

a, c/a,b/a, alpha, beta, gamma: geometrical data.

a	lattice constant, in atomic units, along a axis,
c/a	c/a ratio,
b/a	b/a ratio,
alpha	angle α between b and c axis in degree,
beta	angle β between c and a axis in degree,
gamma	angle γ between a and b axis in degree.

3.3 Specify type of calculation

edelt small imaginary part attached to the Fermi energy.

ewidth width of the energy window covered by the energy contour.

reltyp type of relativistic treatment

nrl	non-relativistic treatment of core level,
sra	semi-relativistic treatment of core level,
nrlls	non-relativistic with spin-orbit interaction,
srals	semi-relativistic with spin-orbit interaction.

sdftyp type of the parametrization of the exchange energy.

		isdf
vbh	v. Barth and Hedin [1],	1
mjw	Moruzzi, Janak, and Williams [2],	2
vwn	Vosko, Wilk, and Nussair [3, 4],	3
lmmjw	non-local + vbh, Langreth and Mehl [5, 6],	4
lmvbh	not implemented,	5
pymjw	non-local + mjw, Perdew et al [7, 8, 9],	6
pyvbh	not implemeted,	7
pyvwn	non-local + vwn, Perdew et al [7, 8, 9],	8
gga91	GGA, Perdew et al [10],	9
ev	GGA only, Engel and Vosko [11].	10

magtyp magnetic state.

mag	magnetic (spin-polarized) calculation,
nmag	non-magnetic,
rvrs	magnetic calculation with reversed spin up/down data,
-mag	the switches rvrs and -mag are equivalent,
kick	is used to kick off the system to be magnetic (not in asa calculations).

Note 1: The switch *asa* may be added to the sdftyp, e.g.: mjwasa means that the atomic sphere approximation (ASA) is to be exploited together with the Moruzzi-Janak-Williams parametrization of the exchange potential.
Note 2: The switch *-kick* does not work together with *asa*.

record record used as input potential

init	used in scf calculations, creates new potential,
1st	may be used for dos, spc, etc.,
2nd	used in dos, spc, etc., or for continuation of scf.

outtyp whether to update the potential file or not (update e.g. for go, quit eg. for dos).

bzqlty quality of BZ mesh (t, l, m, h, u, or any integer number between 0 and 99).

maxitr maximum number of iterations in scf calculation.

pmix mixing parameter in scf calculation (see: [12]).

Note 1: record = 1st or 2nd may also be used to continue a scf calculation if it did not converge after maxitr iterations or if the scf calculation was stopped for other reasons. (Take care which parameters are allowed to be changed if you like to continue a calculation that stopped with a warning or error message.)
Note 2: bzqlty sets the k-mesh to nk irreducible points depending on the lattice type (see function nfqlty). The value of nk used for thr calculation is given in the output file. The qualities t..u are only defined for brvtyp: 1..6, 8, 14, 15 (see table below) in all other cases an integer value has to be set. This is important if brvtyp is set to aux (16) from XtalEdit ! The maximum possible value of nk will depend on the parameter $nk1x$ and $nkmx = nk1x + nk3x$ in specx. ($nk1x$: maximum number of k-points in scf or dos calculation, $nk3x$: maximum number of k-points in spc calculation, additionally needed.) bzqlty = 0 may be used for test cases only. The predefined values t..u for bzqlty are:

		fcc	bcc	hcp	sc	bct	st	bco	rhb	fct
t	test run	3	3	1	2	2	1	2	3	2
l	low quality	5	6	2	4	4	2	4	5	4
m	medium	9	11	4	7	8	5	8	10	8
h	high	12	15	5	9	10	7	10	12	10
u	ultra high	16	20	7	13	15	9	15	17	15

Note: The setting bzqlty = h should work in most of the given cases. However, one should always check the actually used value of nk in outfile.

3.4 Atomic parameter

ntyp number of inequivalent sites called 'type'.

type name specifying the type (may start by alphabet).
ncmp number of component atoms on this site.
rmt muffin-tin radius for this site,
if 0 then it is calculated by the program.
field external, local magnetic field on this site.
lmxtyp maximum l value considered for this site.
anclr nuclear charge of the component atom.
conc concentration of atom anclr
(relative probability of the occupation of the component atom
at the site, may be given in %).
...

Note: Above data for ancrl and conc are repeated until all the ncmp component atoms on this site are given. Then the set of data (type, ncmp, rmt, field, anclr, conc, anclr, conc) have to be be repeated for all ntyp types.

natm number of atoms in the unit cell.

atmicx atomic position in the unit cell (in units of a).
atmtyp type, name of the type given above.
...

Note 1: atmicx may also be written as (x*v1, y*v2, z*v3), where v1, v2, or v3 are a, b, or c. Here a, b, and c are the primitive vectors. e.g. 0.5 0.5 0.5 or 0.5x 0.5y 0.5z means (0.5,0.5,0.5), but 0.5a 0.5b 0.5c means 0.5a+0.5b+0.5c. 0.5 0.5 0.5c and 0.5b 0.5a 0.5c, etc. are allowed, too.
Note 2: Above data, atmicx and atmtyp, have to be repeated until all the atoms in the unit cell are completed.

3.5 Additional parameter

a) Fixed spin calculation:
If the calculational mode is set to fsm (ids = 5) then the value for the fixed spin has to be given at the end of the input file.
b) Bloch spectral function:
If the calculation mode is set to spc then the k-vectors for the calculation of the Bloch spectral function have to be given at the end of the input file. Examples of different sets of k-vectors for spc calculations are given below.

kvec k-vectors.
k_x, k_y, k_z
...

4 Internal parameter of the AKAI KKR program specx

Several parameter controlling the calculation are fixed in specx.f as they control the size of arrays. The predefined values or the given alternative sets are usually adequate for simple systems. However, one may have to change some of those the more complex problems become.
The parameters being fixed in specx.f lines 7, 31, 32, and 41*ff* are:

```
      parameter
...
c--- sample for natm=4, ncmpmx and msizmx are not unique.
     & (natmmx=4, ncmpmx=4, msizmx=36, mxlmx=4, nk1x=250, nk3x=21,
     & msex=201, ngmx=15, nrpmx=250, ngpmx=250, npmx=200, msr=400)
...
c--- the following part is used for all cases.
      parameter
     & (lengx=msex, ndmx=natmmx*(natmmx-1)+1, mxlfac=5-mxlmx,
     & lastmx=2900d0/3d0**mxlfac, ntypmx=ncmpmx, nkmx=nk1x+nk3x)
```

mxlmx is $l_{max}+1$ where l_{max} is the maximum angular momentum used in the calculation. lastmx should be larger than 2, 37, 243, 964, and 2854 for mxl=1, 2, 3, 4, and 5,respectively.
Normally, the maximum size needed for the work area (nwk) is determined by the maximum of $4*natmmx+ngpmx+ntypmx$, $ncmpx*msr$, $5*msr$, $2*mxlmx**2*ncmpmx*msex$, and 7620. The last number, 7620, depends on the number of k-points used in the BZ; 7620 corresponds to the case where $32*10^4$ k-points (about $68\times68\times68$) in the full BZ are used. This number, however, should not be reduced to less than 7620 even if less k-points in the BZ are used.
The most important parameter have the following meaning and they cause the restrictions:

parameter	restricts	meaning
natmmx	natm	maximum number of atoms in the unit cell.
ncmpmx	ncmp(ntype)	maximum number of types over all sites, $\geq\sum_{i=1}^{ntype} ncmp(i)$.
msizmx	lmxtyp(natm)	maximum number of orbital momenta over all atoms, $\geq\sum_{i=1}^{natm}(lmxtyp(itype(i))+1)^2$.
mxlmx	lmxtyp	maximum orbital momentum +1 ($l_{max}+1$), $\geq\max(lmxtyp(itype)+1)$.
nk1x	bzqlty	maximum number of k-points for BZ integration,
nk3x	kvec	maximum number of k-points for spc calculation, for scf and dos $nkmx=nk1x+nk3x$.
msex		number of energy points in dos, mcd, and spc calculation, see also next paragraph.
msr		number of points in r-space for potential, etc..

Note 1: The increase of these parameters will increase the size of various arrays, so be sure to have enough memory installed.
Note 2: Use $msr=421$ for compatibility with the rslab photoemission program (not available for public).

5 Energy mesh and range

a) The energy mesh in calculations of dos (go=dos), magnetic X-ray dichroism at K-edge (mcd, xmd), or Bloch spectral functions (spc) is determined like follows:

- the imaginary part is determined by: *edelt*,
- the range of energies is given by: *ewidth*,
- the number of energies is determined by the parameter $kmx=msex$ as set in specx.f,

- the step width is given by: $de = ewidth/(kmx - 1)$,
- the Fermi energy ef is taken from the scf potential file,
- the parameter ref (= 0.8 actually) is fixed in cemesr, it determines the part of the energy mesh being below the Fermi energy,
- the lower boundary is given by: $el = ef - ewidth * ref$ in practice, el is calculated in such a way that ef is a point of the energy mesh, this means that ref is only roughly the part of the energy mesh below ef.

b) The energy mesh in scf calculations (calctyp = go, etc) is determined by the parameter $kmx = mse0$ (= 65 or 35) actually being fixed in specx.f. For this purpose, cemesh generates a semi elliptic energy contour. The mesh points are located following Fermi's distribution function such that they are distributed densely near the real axis.
The fixed energy parameter mse0 is found in lines 111-114 of specx.f:

```
c      data ef0/7.0d-1/, dex/3d-3/, emxr/1.0d0/, mse0/65/
      data ef0/7.0d-1/, dex/3d-3/, emxr/1.0d0/, mse0/35/
     &    ,xlim/1.5d0/, ng/ngmx/, meshr/msr/, tol/1.0d-6/
     &    ,ids/0/, inv/0/
```

6 k-vectors for spc calculation

Here, some useful path for calculation of the Bloch spectral function are given. k is given in units of $2\pi/a$.

6.1 Cubic crystal systems

6.1.1 simple cubic (sc): $P : m\bar{3}m$

k-point	k-vectors.		
	n_1	n_2	n_3
Γ	0	0	0
X	0	0	1/2
R	1/2	1/2	1/2
M	1/2	1/2	0
T			

k-direction
$\Gamma \to X \quad = \Delta$
$\Gamma \to R \quad = \Lambda$
$\Gamma \to M \quad = \Sigma$
$R \to M \quad = T$

Nice path: $X \to \Gamma \to R \to M \to \Gamma$

6.1.2 face centered cubic (fcc): $F : m\bar{3}m$

k-point	k-vectors.		
	n_1	n_2	n_3
Γ	0	0	0
X	1/2	0	1/2
L	1/2	1/2	1/2
K	3/4	3/8	3/8
W	1/4	1/2	3/4
U		$(= K)$	

k-direction		surface
$\Gamma \to X$	$= \Delta$	001
$\Gamma \to L$	$= \Lambda$	111
$\Gamma \to K$	$= \Sigma$	110
$X \to U$	$= S$	
$X \to W$	$= Z$	
$L \to W$	$= Q$	

Nice path: $X \to \Gamma \to L \to W \to K \to \Gamma$

6.1.3 body centered cubic (bcc): $I : m\bar{3}m$

k-point	k-vectors.		
	n_1	n_2	n_3
Γ	0	0	0
H	1/2	1/2	-1/2
P	1/2	1/2	1/4
N	1/2	0	0

k-direction		surface
$\Gamma \to H$	$= \Delta$	001
$\Gamma \to P$	$= \Lambda$	
$\Gamma \to N$	$= \Sigma$	110
$H \to P$	$= F$	
$H \to N$	$= G$	
$P \to N$	$= D$	

Nice path: $H \xrightarrow{\Delta} \Gamma \xrightarrow{\Lambda} P \to N \xrightarrow{\Sigma} \Gamma$

6.2 Tetragonal crystal systems

6.2.1 simple tetragonal (st): $P : 4/mmm$

k-point	k-vectors.		
	n_1	n_2	n_3
Γ	0	0	0

M	1/2	1/2	0
Z	0	0	1/2
A	1/2	1/2	1/2
X	0	1/2	0
R	0	1/2	0

k-direction
$\Gamma \to X \quad = \Delta$
$\Gamma \to Z \quad = \Lambda$
$\Gamma \to M \quad = \Sigma$
$M \to X \quad = Y$
$M \to A \quad = V$
$Z \to R \quad = U$
$Z \to A \quad = S$
$R \to A \quad = T$
$X \to R \quad = W$

Nice path: $X \to \Gamma \to Z \to A \to M \to \Gamma$

6.2.2 centered tetragonal (bct): $I : 4/mmm$

k-point	k-vectors.		
	n_1	n_2	n_3
Γ	0	0	0
X	0	0	1/2
M	1/2	1/2	-1/2
P	1/4	1/4	1/4
N	0	1/2	0
Z	-1/2	1/2	1/2

k-direction
$\Gamma \to M \quad = \Delta$
$\Gamma \to X \quad = \Lambda$
$\Gamma \to Z \quad = \Sigma$
$X \to P \quad = W$
$P \to N \quad = Q$

Nice path: $M \to \Gamma \to X \to P$

7 Example Input

Before running the program, be sure that it was compiled with appropriate parameter settings in specx.f. For the example given below, they should be at least:

```
      parameter
     & (natmmx=4, ncmpmx=4, msizmx=57, mxlmx=4, nk1x=250, nk3x=21,
     &  msex=201, ngmx=15, nrpmx=250, ngpmx=250, npmx=200, msr=400)
```

The complete input (the comment lines starting with # may, indeed, be removed) for a scf-cpa calculation of an alloy with some random distribution of particular atoms may look like:

```
#---------------------- input data ---------------------------#
#- calctyp  file ------------------------------------------#
   go       co2crfeal_pot
#- brvtyp  a      b/a  c/a  alpha  beta  gamma ---------------#
   fcc     10.822 1.0  1.0  90.0   90.0  90.0
#- edelt  ewidth  reltyp  sdftyp  magtyp ---------------------#
   0.001  1.0     nrl     vwnasa  mag
#- record  outtyp  bzqlty  maxiter  pmix ---------------------#
   init    update  h       200      0.02
#- ntyp ------------------------------------------------------#
   3
#- type  ncmp  rmt  field  lmx  anclr  conc -------------------#
   Co    1     0    0      3    27     100.0
   CrFe  2     0    0      3    24      60.0
                             26      40.0
   Al    1     0    0      2    13     100.0
#- natm ------------------------------------------------------#
   4
#- atmicx -------------------- atmtyp -------------------------#
   0.25000  0.25000  0.25000   Co
   0.75000  0.75000  0.75000   Co
   0.00000  0.00000  0.00000   CrFe
   0.50000  0.50000  0.50000   Al
#---------------------- end input data ----------------------#
```

In short, this may also be written like:

```
   go  co2crfeal_pot
   fcc  10.822 1.0 1.0, , , ,
   0.001  1.0  nrl  vwnasa  mag
   init  update  h  200  0.02
   3
   Co    1  0  0  3  27  100
   CrFe  2  0  0  3  24  60
                     26  40
   Al    1  0  0  2  13  100
   4
   0.25  0.25  0.25   Co
   0.75  0.75  0.75   Co
   0.00  0.00  0.00   CrFe
   0.50  0.50  0.50   Al
```

The author experienced some problems with the number of blanks starting each line (comment lines should always start with # as first character), most probably while changing from DOS to Linux files. It seems that starting data lines with

3 blanks works in all cases. See also the comment of H. Akai in cnsole.f on the use of c or C to assign comment lines.
After the scf calculation is finished, one first may like to check the density of states using the following input:

```
#---------------------- input data --------------------------#
#- calctyp  file ------------------------------------------#
   dos      co2crfeal_pot
#- brvtyp  a      b/a  c/a  alpha  beta  gamma ---------------#
   fcc     10.822 1.0  1.0  90.0   90.0  90.0
#- edelt  ewidth  reltyp  sdftyp  magtyp ----------------------#
   0.001  1.0     nrl     vwnasa  mag
#- record  outtyp  bzqlty  maxiter  pmix ---------------------#
   2nd     quit    h       1        0.02
#- ntyp ------------------------------------------------------#
   3
#- type  ncmp  rmt  field  lmx  anclr  conc -------------------#
   Co    1     0    0      3    27     100.0
   CrFe  2     0    0      3    24      60.0
                             26      40.0
   Al    1     0    0      2    13     100.0
#- natm ------------------------------------------------------#
   4
#- atmicx -------------------- atmtyp -------------------------#
   0.25000  0.25000  0.25000   Co
   0.75000  0.75000  0.75000   Co
   0.00000  0.00000  0.00000   CrFe
   0.50000  0.50000  0.50000   Al
#---------------------- end input data ----------------------#
```

Note that changes were only made in lines 3 and 9 compared to the scf input. The changes in line 9 are not essentially necessary as those will be recognized automatically by specx itself.
In case that the DOS is too much smoothed or too noisy, change the parameter for edelt in line 7 of the input to smaller or larger values, respectively. Change the parameter ewidth in line 7 to obtain a larger or smaller range of energies for the DOS. However, if you like to have more energies then you have to change the parameter msex in specx.f and to recompile the program (see also remarks in the previous paragraphs).
Finally, we like to calculate the Bloch spectral function for one of the directions of high symmetry. In that case, one needs appropriate k-vectors as additional input.
For the Δ direction of the fcc lattice and 21 k-points, the input is:

```
#---------------------- input data --------------------------#
#- calctyp  file ------------------------------------------#
   spc      co2crfeal_pot
#- brvtyp  a      b/a  c/a  alpha  beta  gamma ---------------#
   fcc     10.822 1.0  1.0  90.0   90.0  90.0
```

```
#- edelt  ewidth  reltyp  sdftyp  magtyp ----------------------#
   0.001  1.0     nrl     vwnasa  mag
#- record  outtyp  bzqlty  maxiter  pmix ----------------------#
   2nd     quit    h       1        0.02
#- ntyp -----------------------------------------------------#
   3
#- type  ncmp  rmt  field  lmx  anclr  conc -------------------#
   Co    1     0    0      3    27     100.0
   CrFe  2     0    0      3    24      60.0
                             26      40.0
   Al    1     0    0      2    13     100.0
#- natm ----------------------------------------------------#
   4
#- atmicx -------------------- atmtyp -------------------------#
   0.25000  0.25000  0.25000   Co
   0.75000  0.75000  0.75000   Co
   0.00000  0.00000  0.00000   CrFe
   0.50000  0.50000  0.50000   Al
#- k-vectors for spc  ---------------------------------------#
#- fcc Gamma -- Delta -- X  ----------------------------------#
   0.00000000  0.00000000  0.00000000
   0.00000000  0.02500000  0.02500000
   0.00000000  0.05000000  0.05000000
   0.00000000  0.07500000  0.07500000
   0.00000000  0.10000000  0.10000000
   0.00000000  0.12500000  0.12500000
   0.00000000  0.15000000  0.15000000
   0.00000000  0.17500000  0.17500000
   0.00000000  0.20000000  0.20000000
   0.00000000  0.22500000  0.22500000
   0.00000000  0.25000000  0.25000000
   0.00000000  0.27500000  0.27500000
   0.00000000  0.30000000  0.30000000
   0.00000000  0.32500000  0.32500000
   0.00000000  0.35000000  0.35000000
   0.00000000  0.37500000  0.37500000
   0.00000000  0.40000000  0.40000000
   0.00000000  0.42500000  0.42500000
   0.00000000  0.45000000  0.45000000
   0.00000000  0.47500000  0.47500000
   0.00000000  0.50000000  0.50000000
#---------------------- end input data ----------------------#
```

References

[1] U. v. Barth and L. Hedin; J. Phys. C: Solid State Phys. **5**, 1629 (1972).

[2] V. L. Moruzzi, J. F. Janak, and A. R. Williams; *Calculated Properties of Metals*, Pergamon Press, New York (1978).

[3] S. H. Vosko, L. Wilk, and M. Nussair; Can. J. Phys. **58**, 1200 (1980).

[4] S. H. Vosko and L. Wilk; Phys. Rev. B **22**, 3812 (1980).

[5] D. C. Langreth, and M. J. Mehl; Phys. Rev. B **28**, 1809 (1983).

[6] F. W. Kutzler and G. S. Painter; Phys. Rev. B **37**, 2850 (1988).

[7] J. P. Perdev; Phys. Rev. B **33**, 8822 (1986).

[8] J. P. Perdev and W. Yue; Phys. Rev. B **33**, 8800 (1986).

[9] P. Mlynarski and D. R. Salahub; Phys. Rev. B **43**, 1399 (1991).

[10] J. P. Perdew, J. A. Chevary, S. H. Vosko, K. A. Jackson, M. R. Pederson, D. J. Singh, and C. Fiolhais; Phys. Rev. B **46**, 6671 (1992).

[11] E. Engel and S. H. Vosko; Phys. Rev. B **47**, 13164 (1993).

[12] H. Akai and P. P. Dederichs; J. Phys. C **18**, 2455 (1985).