SPRKKR

A spin polarized relativistic Korringa-Kohn-Rostoker (SPR-KKR) code
for Calculating Solid State Properties

Hubert Ebert

User's Guide

December 5, 2017

Universität München
Dept. Chemie – Phys. Chemie
Butenandtstr. 5-13
D-81377 München

H. Ebert
SPRKKR
A spin polarized relativistic Korringa-Kohn-Rostoker
(SPR-KKR) code for Calculating Solid State Properties

version 7.7

Prof. Dr. Hubert Ebert
Universität München
Dept. Chemie – Phys. Chemie
Butenandtstr. 5-13
D-81377 München
Germany

Contents

1 Introduction

1.1 The **SPRKKR**-package

This manual describes the use of the spin-polarized relativistic KKR band structure program package **SPRKKR**. Development of this package started during a post-doc stay of H. Ebert with B. L. Gyorffy in 1986/1987 and was continued later with contributions by M. Battocletti, D. Benea, M. Kosuth, J. Minar, A. Perlov, V. Popescu, and others. Several subroutines used by **SPRKKR** have been kindly supplied by H. Akai, S. Blügel, J. J. Rehr, W. M. Temmerman, and P. Strange. The package has seen a major extension by including the spectroscopy module of J. Braun, that was later on further extended. P. Blaha kindly made the LaTeX-files of the WIEN97 manual available, that have been used as a template for this manual.

The **SPRKKR**-package allows to calculate the electronic structure of arbitrary three-dimensional periodic systems, including in particular systems with chemical disorder. The treatment of two dimensional periodic systems (e.g. surfaces) can be done at the moment only by using an auxiliary system having three dimensional periodicity or by making use of the cluster approximation. The electronic structure calculation can be done in a non-relativistic, scalar-relativistic as well as fully relativistic mode. In the scalar-relativistic mode paramagnetic as well as spin-polarized systems can be treated, including non-collinear spin structures and arbitrary spin spirals. In the fully relativistic mode, paramagnetic as well as spin-polarized systems with an arbitrary spin configuration can be dealt with. On the basis of the electronic structure calculation many different properties can be investigated by means of the **SPRKKR**-package, with a strong emphasize on response functions and spectroscopic properties. This applies in particular for the fully relativistic mode.

At the moment the **SPRKKR**-package is restricted to the atomic sphere approximation (ASA) geometry for the charge density and potential. This restriction will be removed in the near future allowing in addition full potential type calculations.

The **SPRKKR**-package is based on the KKR-Green's function formalism that makes use of multiple scattering theory. This implies that the information on the electronic structure of a system is not expressed in terms of Bloch wave functions and eigenvalues but the corresponding Green's function, leading to an extreme flexibility of the method. An introduction to band structure theory, including the KKR method, can be found for example in a review article by S. Faulkner [1]. A more detailed presentation of the KKR method is given in the monographies of P. Weinberger [2], A. Gonis and W. Butler [3, 4], that also deal with the spin-polarized relativistic case. The KKR-formalism for this case will be sketched in short in the next section. More information and many applications can be found in two older review articles [5, 6] and a rather new one [7] as well as on the webpage http://www.kkr-gf.org/.

The next chapter describes the use of the **SPRKKR**-package for a number of standard applications. This is followed by a description of the use of the **plot** that allows to reformat and modify calculated data for display and analysis. Technical information and a guideline for installation of the **SPRKKR**-package can be found in chapter 9. Finally some references connected with applications of the **SPRKKR**-package are collected at the end. For a more complete list of references see [5] and [8].

1.2 Formal Background

A short description of the spin-polarized relativistic Korringa-Kohn-Rostoker Green's function method that supplies the formal background for the **SPRKKR**-package is given in the following. Throughout the conventions introduced in the books of M. E. Rose will be used [9, 10].

The spin-polarized relativistic Korringa-Kohn-Rostoker Green's function method is based on the following Dirac-Hamiltonian for a spin-polarized system

$$\left[\frac{\hbar}{i}c\vec{\alpha}\cdot\vec{\nabla}+\beta mc^2+V_{\rm eff}(\vec{r})+\beta\vec{\sigma}\cdot\vec{B}_{\rm eff}(\vec{r})\right]\Psi_i(\vec{r})=\epsilon_i\Psi_i(\vec{r}) \tag{1.1}$$

with

$$\vec{B}_{\rm eff}(\vec{r})=\vec{B}_{\rm ext}(\vec{r})+\frac{\partial E_{\rm xc}[n,\vec{m}]}{\partial\vec{m}(\vec{r})}\ . \tag{1.2}$$

The spin dependent potential is constructed within a relativistic version of spin density functional theory. With the potentials functions $V_{\rm eff}(\vec{r})$ and $\vec{B}_{\rm eff}(\vec{r})$ available, the first step is to solve the corresponding single site Dirac equation for every atom type. In principle, the effective magnetic field $\vec{B}_{\rm eff}(\vec{r})$ may point in any direction (non-collinear magnetism). However, calculations are simplified for $\vec{B}_{\rm eff}(\vec{r})=B_{\rm eff}(r)\hat{\vec{e}}_{\rm z}$. Therefore the single site Dirac equation is solved in a local frame of reference for which this condition is fulfilled. For this purpose the ansatz

$$\psi_\nu=\sum_\Lambda\psi_{\Lambda\nu} \tag{1.3}$$

is used with the partial waves $\psi_{\Lambda\nu}$ chosen to have the same form as the linearly independent solutions for a spherical symmetric potential:

$$\psi_\Lambda(\vec{r},E)=\left(\begin{array}{c} g_\kappa(r,E)\,\chi_\Lambda(\hat{\vec{r}})\\ if_\kappa(r,E)\,\chi_{-\Lambda}(\hat{\vec{r}})\end{array}\right)\ . \tag{1.4}$$

This leads to the following set of coupled radial differential equations:

$$
\begin{aligned}
P'_{\Lambda\nu} &= -\frac{\kappa}{r}P_{\Lambda\nu} + \left[\frac{E-V}{c^2}+1\right]Q_{\Lambda\nu} + \frac{B}{c^2}\sum_{\Lambda'}\langle\chi_{-\Lambda}|\sigma_{\mathrm{z}}|\chi_{-\Lambda'}\rangle Q_{\Lambda'\nu} && (1.5)\\
Q'_{\Lambda\nu} &= \frac{\kappa}{r}Q_{\Lambda\nu} - [E-V]\,P_{\Lambda\nu} + B\sum_{\Lambda'}\langle\chi_{\Lambda}|\sigma_{\mathrm{z}}|\chi_{\Lambda'}\rangle P_{\Lambda'\nu}\ . && (1.6)
\end{aligned}
$$

In practice the coupling is restricted to $\mu-\mu'=0$ and $l-l'=0$ with two functions $P_{\kappa\mu\nu}$ and $P_{-\kappa-1\,\mu\,\nu}$ coupled for $|\mu|<j$ and no coupling present otherwise.

Next, the single site t-matrix $t_{\Lambda\Lambda'}$ is calculated and transformed from the local to the global frame of reference (if necessary). With all t-matrices available, the multiple scattering problem is solved either by performing a BZ-integration

$$
\tau^{nn'}_{\Lambda\Lambda'}(E) = \frac{1}{\Omega_{\mathrm{BZ}}}\int_{\Omega_{\mathrm{BZ}}} d^3k[\underline{t}^{-1}(E)-\underline{G}(\vec{k},E)]^{-1}_{\Lambda\Lambda'}e^{i\vec{k}(\vec{R}_n-\vec{R}_{n'})} \qquad (1.7)
$$

or using the real space cluster approach. This means that the scattering path operator $\tau^{nn'}_{\Lambda\Lambda'}(E)$ is obtained by inverting the real space KKR-matrix:

$$
\underline{\underline{\tau}} = [\underline{\underline{m}} - \underline{\underline{G}}]^{-1}\ . \qquad (1.8)
$$

If there is substitutional disorder present in the system, this is accounted for by making use of the Coherent Potential Approximation (CPA) alloy theory. This implies that the CPA equations

$$
\begin{aligned}
x_{\mathrm{A}}\underline{\tau}^{nn,\mathrm{A}} + x_{\mathrm{B}}\underline{\tau}^{nn,\mathrm{B}} &= \underline{\tau}^{nn,\mathrm{CPA}} && (1.9)\\
\underline{\tau}^{nn,\alpha} &= \underline{\tau}^{nn,\mathrm{CPA}}\left[1+\left(\underline{t}^{-1}_{\alpha}-\underline{t}^{-1}_{\mathrm{CPA}}\right)\underline{\tau}^{nn,\mathrm{CPA}}\right]^{-1} && (1.10)
\end{aligned}
$$

have to be solved iteratively.

After the multiple scattering problem has been solved, the resulting scattering path operator has to be transformed back to the local frame of reference in case of a non-collinear spin configuration. Finally, the Green's function is given by:

$$
\begin{aligned}
G(\vec{r},\vec{r}\,',E) &= \sum_{\Lambda\Lambda'} Z^{n}_{\Lambda}(\vec{r},E)\tau^{nn'}_{\Lambda\Lambda'}(E)Z^{n'\times}_{\Lambda'}(\vec{r}\,',E)\\
&\quad -\sum_{\Lambda}\left[\, Z^{n}_{\Lambda}(\vec{r},E)J^{n\times}_{\Lambda}(\vec{r}\,',E)\Theta(r'-r)\right.\\
&\quad \left. +J^{n}_{\Lambda}(\vec{r},E)Z^{n\times}_{\Lambda}(\vec{r}\,',E)\Theta(r-r')\,\right]\delta_{nn'}\ . && (1.11)
\end{aligned}
$$

From this the charge density and spin magnetization are obtained by:

$$n(\vec{r}) = -\frac{1}{\pi}\Im\,\mathrm{Trace}\int^{E_F} dE\, G(\vec{r},\vec{r},E) \tag{1.12}$$

$$m(\vec{r}) = -\frac{1}{\pi}\Im\,\mathrm{Trace}\int^{E_F} dE\, \beta\sigma_{\mathrm{z}}\, G(\vec{r},\vec{r},E)\ . \tag{1.13}$$

Recalculating the potential functions V and B from these, the SCF-cycle is closed. Using an appropriate algorithm, V and B can be updated to achieve self-consistency.

From the Green's function most electronic properties can be derived straightforwardly. For example the spin and orbital magnetic moments are obtained from:

$$\mu_{\mathrm{spin}} = -\frac{\mu_{\mathrm{B}}}{\pi}\Im\,\mathrm{Trace}\int^{E_F} dE \int_V d^3r\, \beta\sigma_{\mathrm{z}}\, G(\vec{r},\vec{r},E) \tag{1.14}$$

$$\mu_{\mathrm{orb}} = -\frac{\mu_{\mathrm{B}}}{\pi}\Im\,\mathrm{Trace}\int^{E_F} dE \int_V d^3r\, \beta l_{\mathrm{z}}\, G(\vec{r},\vec{r},E)\ . \tag{1.15}$$

These expressions are based on the identity:

$$\Im\, G(E) = -\pi \sum_{\alpha} |\alpha\rangle\, \langle\alpha|\, \delta(E - E_\alpha)\ , \tag{1.16}$$

where $|\alpha\rangle$ are single particle electronic states with energies E_α. Many other electronic and spectroscopic properties can be derived from this identity as well.

The standard mode of the KKR package is the four component Dirac formalism sketched above. However, the program also allows to use more complex Hamiltonians or to manipulate the spin orbit coupling. Non- or scalar relativistic calculations are also possible to a certain extent. So far the atomic sphere approximation (ASA) is used as a geometrical constraint for the potential functions, charge density and spin magnetisation. A forthcoming release will supply a full potential mode.

1.3 Short Description

A short description of the various components of the **SPRKKR**-package and their interaction is given in the following.

1.3.1 The graphical user interface **xband**

The use of the SPR-KKR package is supported by the graphical user interface (GUI) **xband**. This tool allows to set up the various input files via the Create input file menu. An expert mode allows to set variables needed for testing, additional non standard output etc. **xband** also creates a dummy potential file to start a SCF calculation using **kkrscf**. Finally, **xband** allows to run the various programs (see 1.3.2) interactively or to submit batch jobs, to handle the various files (see 1.3.3) and to perform postprocessing and graphics using the programs **plot** (see 1.3.4) and **xmgrace**.

For further details see the **xband** manual.

1.3.2 Programs

The **SPRKKR** package consists of several programs used in general for different tasks.

- **kkrscf**
  This program is exclusively used for SCF-calculations.
- **kkrgen**
  This is the general purpose program that is used for most tasks, i.e., calculating the DOS, Bloch spectral functions, XAS spectra and so on.
- **embscf** & **embgen**
  These programs differ from **kkrscf** & **kkrgen** only in the way the τ-matrices are calculated. While the programs **kkrscf** & **kkrgen** use a BZ-integration the cluster approach is used here.
- **kkrchi**
  This program allows to calculate a number of linear response functions.
- **kkrspec**
  This program is specialized on various spectroscopic applications.

The programs **kkrscf**, **kkrgen**, **embscf**, **embgen**, **kkrchi**, and **kkrspec** are called, e.g., as:

```
kkrscf < Fe.inp
```

Standard output (and error) is recommended to be written to a file as shown below. In addition to the input file (here Fe.inp) at least a potential file has to be supplied. Its name is specified in the input file, e.g., via POTFIL = Fe.pot. Alternatively, the **SPRKKR** programs may be called via **xband**.

1.3.3 Files

The run of the **SPRKKR** programs **kkrscf**, **kkrgen**, **embscf**, **embgen**, **kkrchi** and **kkrspec** is controlled by a common input file. The information in this file is reduced to a minimum because most necessary data are supplied by the potential file.

The potential file contains all information on the geometrical and atomic configuration of a system as well as the atomic potentials to run a program. If a dummy potential file without potentials is created via **xband** it has the string SCF-START in the header. This file can exclusively be used by **kkrscf** to start a SCF-cycle. **kkrscf** writes the new potential after every iteration to the new potential file SYS.pot_new.

The subroutine ⟨POTFIT⟩ reads the potential file and is able to import potential data from a number of other program packages as for example the Stuttgart TB-LMTO program. To supply data from other sources ⟨POTFIT⟩ can easily be extended. The minimum information to be supplied is described in the header of ⟨POTFIT⟩.

Information on the calculation and results are usually written to standard output but may be redirected to a file by:

```
kkrgen < Fe.inp > Fe.out
```

or

```
kkrgen < Fe.inp | tee Fe.out
```

In addition data files may be needed or created by **SPRKKR** programs. The file names of all these files are derived from the variables `DATASET` and `ADSI` specified in the input file (see 2).

1.3.4 The post-processing program **plot**

The program **plot** reads in data files created by **kkrgen** and writes the data reformated to a file that can be passed directly to the graphics program **xmgrace**. Where appropriate it reads parameters from a second input file used for example to broaden XAS raw spectra.

plot may be activated either by calling:

```
plot < Fe.dos
```

or by using **xband**.

2 The **SPRKKR** input file

2.1 Input Format

The input file is organised in various SECTIONS. The start of a section has to be indicated by writing the section name, starting at the first column of a line. Every section may contain information distributed over several lines. Every non-blank character - apart from the COMMENT CHARACTER (#) - in column 1 is assumed to belong to a new section name and therefore ends the current section - if this is not forced by the end of the file (EOF).

Normally, for each SECTION a number of VARIABLES have to be specified by assigning a string or number to it by typing VARIABLE=VALUE. For example in section SCF the statement ′ VXC=VWN ′ means that the Vosko, Wilk, Nusair XC-parametrisation is selected. There have to be a blank before VARIABLE and a blank after VALUE that both are recognised by the program (the blank after VALUE can be omitted at the end of a line). VARIABLE may be typed in upper or lower case. and VALUE may be a STRING, a REAL number or an INTEGER number.

If VALUE is a STRING it may be typed in upper or lower case. The program will convert to upper or lower case where necessary. Only the strings for the DATASET and the filenames are not touched by the program. The programs interprete the characters , ; () {} [] as separators and delimiters and therefore they must not be part of any STRING. These characters as well as the blank are assumed to indicate the beginning and end of a STRING. The program takes care that your strings are not too long.

If VALUE is a REAL number all formats that are accepted by format free input of a FORTRAN - program are allowed: 12, 1.34, .99, 1E-5 and so on.

If VALUE is an INTEGER number type an integer and nothing else.

Some VARIABLEs may have a set of values i.e. are ARRAYs. Again the values may be a STRING, a REAL number or an INTEGER number. To assign a value to the elements of the ARRAY just type ARRAY={VALUE1, VALUE2, VALUE3, ..} use {} to indicate the start and end of the value set. If no delimiter is used the program assumes that only one element is present although more could be used. Notice that if less values are supplied than expected, the program fills the array with the last value specified. Use blank, ′,′ or ′;′ to separate the

VALUEs. The program takes care that you don't try to specify more elements than storage has been reserved and that your STRINGs are not too long.

Apart from VARIABLEs there are SWITCHes to be set. This is done just by typing them. If these SWITCHes do not occur in the input file, the DEFAULT value will be used. Some SWITCHes are stronger than others. Especially within SECTION TASK take care that just one switch is set.

The order of the VARIABLE=VALUE - and SWITCH - statements in a SECTION is arbitrary. However, if VAR=VALUE1 VAR=VALUE2 occurs, the first setting will be used.

All input text right to the COMMENT CHARACTER # is interpreted as comment. Lines starting with # are skipped.

Thus a typical input file may look like this:

```
#######################################################################
#  SPR-KKR input file    Fe.inp
#  created by xband on Wed Jul 11 13:46:53 CEST 2001
#######################################################################

CONTROL  DATASET = Fe
         POTFIL  = Fe.pot
         PRINT = 0

STRCONST ETA= 0.35  RMAX=2.9  GMAX=3.3

TAU      BZINT= POINTS  NKTAB= 250

ENERGY   NE=30 EMIN=-0.2

SCF      NITER=200 MIX=0.20 SCFVXC=VWN
               TOL=0.00001  ISTBRY=1

TASK     SCF
```

2.2 General Input Parameters

To run a **SPRKKR** program several variables have to be specified in the input file. For most of the variables/switches the programs have default values stored internally. These are given in the tables below. If the input file is set up via **xband**, it may suggest different values depending on the task. The input sections introduced in the following do not depend on a specific task. Those depending on the task are introduced later together with this task.

2.2.1 File names and output control

section CONTROL

VAR / SWITCH	default	description

`DATASET` = string	-	name of data set used to create file names
`ADSI` = string	-	additional data set identifier (optional). This string is added to `DATASET` when creating file names and can be used to distinguish program runs done with different parameter settings etc.
`POTFIL` = string	-	name of potential file. The absolute or relative path, as e.g. `POTFIL`=../Fe.pot may be used.
`PRINT` = integer	0	print level 0-5. This parameter controls only the output but has no effect on the program run.

2.2.2 Angular momentum expansion

The KKR-method is a minimum basis set method. This means that the angular-momentum expansion can be chosen according to the atomic properties of the atomic types. For transition metals it is therefore normally sufficient to have a maximum l-value of 2 (`NL` $= 3$). For systems with many atoms per unit cell it is in principle possible to set the l-expansion according to the atom types on the lattice sites. For US having the NaCl-structure one could choose `NL` $= 4$ for the U-site and `NL` $= 2$ for the S-site. At the moment this possibility, that would save storage and computer time, is not supported by all subroutines. For this reason a common l-expansion cutoff is used, that is fixed by the highest that occurs. For US this implies that `NL` $= 4$ is used for all sites. The program fixes the angular momentum authomatically to `NL` $= 3$ if transition metals are involved and to `NL` $= 4$ if rare earth or actinide atoms are in the system. A higher angular momentum cutoff can be obtained by setting the parameter `NL` within the section `SITES`. This is in particular important when dealing with MEXAFS at high energies.

section SITES

VAR / SWITCH	default	description
`NL`=integer	3 or 4	Common angular momentum cutoff parameter with `NL` $= l_{max} + 1$ preset according to the atomic types in the system

2.2.3 Calculating the scattering path operator

The scattering path operator τ can be calculated by a BZ-integration (see Eq. (1.7)) or the cluster approach; i.e., the real space KKR matrix is inverted (see Eq. (1.8)). The first approach is used in **kkrscf**, **kkrgen**, **kkrchi** and **kkrspec**, while the later one is used by **embscf** and **embgen**.

section TAU

VAR / SWITCH	default	description
`BZINT`=string	POINTS	mode of BZ-integration WEYL : Weyl method POINTS : special points method
`NKMIN`=integer	300	min. and max. number of $\vec{k}$-points
`NKMAX`=integer	500	for Weyl integration
`NKTAB`=integer	300	number of $\vec{k}$-points for special points method
`CLUSTER`	OFF	Cluster type calculation; i.e. calculate τ by inverting the real space KKR matrix. Specify cluster center using `IQCNTR` or `ITCNTR` and its size using `NSHLCLU` or `CLURAD`.
`NSHLCLU`=integer	-	number of atomic shells around the central atom of a cluster.
`CLURAD` =integer	-	radius of the cluster in multiples of `ALAT`.
`IQCNTR` =integer	-	the center of the cluster is set at the site position with number `IQCNTR` of the specified basis.
`ITCNTR` =integer	-	the center of the cluster is set at one of the site positions that is occupied by the atomic type `ITCNTR`.
`NLOUT` =integer	3	the calculated τ-matrix is printed up to l_{max} =`NLOUT`.
`MOL`	OFF	Cluster type calculation but for a molecular system. The system is specified as for `CLUSTER`.

The scattering path operator τ is usually calculated by a BZ-integration, with different methods available:

- The Weyl method (`BZINT=WEYL`) is a point sampling method using more or less random points. The number of $\vec{k}$-points used for the integration varies quadratically between 0.0 and `ImE` according to the imaginary part of the energy.
- The special point method (`BZINT=POINTS`) uses a regular $\vec{k}$-point grid with `NKTAB` points. It is the standard method and gives a good compromise concerning accuracy and efficiency. For `BZINT=POINTS` the parameter `NKTAB` will be adjusted to allow a regular mesh.

In all cases the $\vec{k}$-mesh specifications refer to the smallest possible irreducible part of the BZ; i.e. for cubic systems one has: $V_{IBZ} = V_{BZ}/48$.

Using the real space cluster approach by setting the switch `CLUSTER` is sometimes sufficient or more appropriate than performing a BZ-integration, e.g. when calculating EXAFS spectra for high energies. In both cases the same input files can be used. The major difference is that the BZ-integration treats an infinite solid while for the cluster mode a finite cluster is cut out of the infinite solid specified by the parameters `NSHLCLU`, `CLURAD`, `IQCNTR` and `ITCNTR` assumed to be big enough to represent the infinite solid. Using a BZ-integration the site-diagonal scattering operator τ^{qq} is calculated for all `NT` sites of the basis in the unit cell. For the cluster mode, only the scattering operator τ^{qq} for the center of the cluster is calculated, because the results will be the less bulk-representative the more one goes away from the center. Therefore, all information printed out for the other sites have to be ignored.

Setting the switch `MOL` a finite cluster of atoms; i.e. a molecule can be treated. In this case `NQ` is simply the number of atoms in the molecule. The atomic configuration is specified by the basis of an arbitrary lattice; i.e. the primitive lattice vectors are dummy input while `ALAT` specifies the dimensions of the system. While for `CLUSTER` the cluster atoms are embedded in a free electron sea, i.e. one deals with continuum states, it is assumed for `MOL` that one is dealing with bound electronic states. In contrast to `CLUSTER` the scattering operator τ^{qq} for all sites within the molecule is calculated properly.

2.2.4 $\vec{k}$-dependent KKR structure constants

section STRCONST

VAR / SWITCH	default	description
`ETA`=real	calculated	Ewald parameter
`RMAX`=real	calculated	convergency radius in real space
`GMAX`=real	calculated	convergency radius in reciprocal space

The calculation of the $\vec{k}$-dependent KKR structure constant matrix $\underline{G}(\vec{k}, E)$ is controlled by three convergence parameters. `ETA` determines the relative weight of the real and reciprocal space lattice sums, that are determined by the convergence radii `RMAX` and `GMAX`, respectively. These convergence parameters have to be optimised anew if the lattice structure, the lattice parameter or the energy or $\vec{k}$-range used is changed. This is done by the program if no values are applied via the input file. In some cases, in particular if one works at high energies, it might be necessery to set the convergence *by hand*. For this purpose one can start from the values set by **kkrgen** or **kkrscf** (see the output file).

2.2.5 CPA-calculations for disordered alloy systems

section CPA

VAR / SWITCH	default	description
`NITER`=integer	20	maximum number of CPA iterations
`TOL`=real	0.0001	threshold for stopping CPA-cycle

For a system with substitutional disorder, the CPA is used. The listed variables control the CPA cycle specified by Eqs. (1.9) and (1.10).

2.2.6 Calculation mode

If not specified otherwise the programs of the **SPRKKR**-package assume that a magnetic system should be treated in a fully relativistic way. By setting the parameter `SP-SREL` in the section `MODE` a scalar relativistic calculation can be done instead for a magnetic system. This is useful when starting the SCF-cycle (see below), because it is somewhat faster than the fully relativistic mode. Because in the `SP-SREL`-mode the same representation is used as for the fully relativistic one, all types of calculations described in the next chapters can be done in this mode.

section MODE

VAR / SWITCH	default	description
`SP-SREL`	OFF	work in the spin-polarized scalar-relativistic mode

If it is known that the system considered is non-magnetic, one can make use of this by setting the switch `NONMAG`.

section CONTROL

VAR / SWITCH	default	description
`NONMAG`	OFF	perform a non-spin-polarized calculation for a paramagnetic system

This leads to a higher symmetry for the system and accordingly in general to shorter runtime for the SCF-cycle because a smaller part of the Brillouin-zone has to be sampled.

2.2.7 Orientation of the magnetisation

If magnetic systems are considered by the **SPRKKR** programs it is in general assumed that the magnetic moments of all atoms point parallel or antiparallel to the crystallographic z-axis. **SPRKKR** allows to consider different configurations as well. Using the parameter `MDIR` in section `MODE` the direction of the magnetisation can be oriented in an arbitrary direction. This implies that for all lattice sites the same orientation is assumed.

section MODE

VAR / SWITCH	default	description
`MDIR` $= \{x, y, z\}$	$\{0, 0, 1\}$	Common magnetisation direction vector with x, y and z in Cartesian coordinates. The normalisation is arbitrary.

Internally, `MDIR` is converted to corresponding Euler angles (α, β, γ) (with γ redundant at the moment), that specify a local frame of reference. The Dirac equation for the single site problem is solved for this, because it reduces the spin dependent potential term in the Dirac equation to $\mu_B B_{eff}(r)\sigma_z$. Instead of using a common magnetisation direction for all lattice sites the direction may also be set individually for each lattice site (non-collinear spin structures). This is done using the parameter `MDIR`$\star$ (with $\star = 1, ...,$ `NQ`) in the section `MODE`.

section MODE

VAR / SWITCH	default	description
`MDIR`$\star$ $= \{x, y, z\}$	$\{0, 0, 1\}$	orientation vector for the spin magnetisation on the lattice site $\star$, with $\star$ standing for the site number `IQ` $= 1, ...,$`NQ`

2.2.8 Manipulating the spin-orbit coupling

When dealing with spin-orbit induced properties it is often interesting to demonstrate the connection of the investigated effect and the spin-orbit coupling. This can be done by manipulating the spin-orbit coupling. The **SPRKKR** package allows this in several ways [11, 12]. The most simple way to manipulate the spin-orbit coupling is to change the speed of light c. Because the most prominent relativistic corrections are proportional to $1/c^2$, one approaches the non-relativistic limit for c going to infinity. Accordingly, the **SPRKKR** input uses a scaling parameter `C`$\star$ $= (c_0/c)^2$ (with $\star = 1, ...,$ `NT`) that has to be put to small values for the non-relativistic limit (for numerical reasons one should have `C`$\star$ $\geq 10^{-4}$), to 1 for a relativistic and > 1 for a ultra-relativistic calculation.

section MODE

VAR / SWITCH	default	description
`C⋆` $=\{real\}$	1.0	scale the speed of light for atom type *, with ⋆ standing for `IT` $= 1, ..., $ `NT`

The scheme described above obviously manipulates all relativistic effects simultaneously. To scale exclusively the strength of the spin-orbit coupling, one may use the parameter `SOC⋆`

section MODE

VAR / SWITCH	default	description
`SOC⋆` $=\{real\}$	1.0	scale the strength of the spin-orbit coupling for atom type ⋆ standing for `IT` $= 1, ..., $ `NT`

Setting `SOC⋆` $= \{0.0\}$ suppresses the spin-orbit coupling completely and the calculation corresponds to a so-called scalar relativistic one. The parameter `SOC⋆` in addition allows to use only parts of the spin-orbit coupling. Setting `SOC⋆`$= -1$ only the spin-diagonal part $\xi\sigma_z l_z$ is used, while for `SOC⋆`$= -2$ only the spin-off-diagonal or spin-mixing part $\xi(\sigma_x l_x + \sigma_y l_y)$ is used in the calculations.

3 Electronic structure calculations

3.1 Creating self-consistent potentials

To perform SCF calculations using **kkrscf** an input and potential file has to be supplied. Setting up the input via **xband** a dummy potential file that supplies all information on the system without potential functions is created. (Alternatively, a potential file stemming from another band structure calculation can be used to start from.) **kkrscf** writes after every iteration the new potential functions to DATASET.pot_new, with DATASET.pot the name of the original potential file. Thus, to use the new potential file rename it to DATASET.pot or set POTFIL = DATASET.pot_new in the input files used later on.

The specific part of the input file DATASET.inp supplies the following parameters:

section TASK

VAR / SWITCH	default	description
`SCF`	ON	perform a SCF calculation

section SCF

VAR / SWITCH	default	description
`VXC`=string	VWN	parametrisation of the exchange-correlation potential VWN: Vosko, Wilk, Nusair JWM: Janak, Williams, Moruzzi VBH: von Barth, Hedin
`ALG`=string	BROYDEN2	SCF algorithm TCHEBY : Tchebychev BROYDEN2: Broyden's second method

`NITER`=integer	200	number of iterations
`MIX`=real	0.20	SCF-mixing parameter
`TOL`=real	0.00001	SCF-tolerance
`ISTBRY`=integer	1	start Broyden after `ISTBRY` iterations
`ITDEPT`=integer	40	iteration depth for Broyden algorithm
`QION`=$\{Q_1,...,Q_{\texttt{NT}}\}$	*calculated*	guess for the ionic charges Q_t for `NT` atomic types
`MSPIN`=$\{\mu_1,...,\mu_{\texttt{NT}}\}$	*table*	guess for the spin magnetic moment $\mu_{spin,t}$ for `NT` atomic types
`USEVMATT`	OFF	set up the starting potential using the original Mattheiss construction for the potential V instead of the charge density

section ENERGY

VAR / SWITCH	default	description
`NE`=integer	30	number of E-mesh points
`EMIN`=real	-0.15	real part of lowest E-value

If **kkrscf** is started using a dummy potential file created by **xband**, a reasonable guess for the potential is constructed first. By default this is done using a superposition of atomic charge densities (Mattheiss construction) and making use of the corresponding ionic charges. A guess for the ionic charges (`QION`) may also be supplied by the user to force the potential in a certain direction. Also the spin magnetic moment `MSPIN` may be supplied to force a certain spin configuration to start from – otherwise the elemental values for `MSPIN` will be assumed. As an alternative the Mattheiss construction may be applied for the potential directly by setting the switch `USEVMATT` . In this case it is only sensible to specify `MSPIN` in addition.

After setting up the starting potential, the Fermi energy is fixed by calculating the DOS along a straight path parallel to the real axis. The corresponding DOS is written to the file DATASET_SCFSTART.dos. Use **plot** to create from this file a corresponding **xmgrace** file. Check whether the DOS looks reasonable. If the DOS at the bottom of the energy window is not more or less 0, i.e. the bottom of the energy band should lie within the energy window, decrease `EMIN` and start again.

The SCF calculations are done using an energy path in the complex plane that is specified by the settings in the section `ENERGY` (see Fig. 3.1). The SCF convergence is checked after each

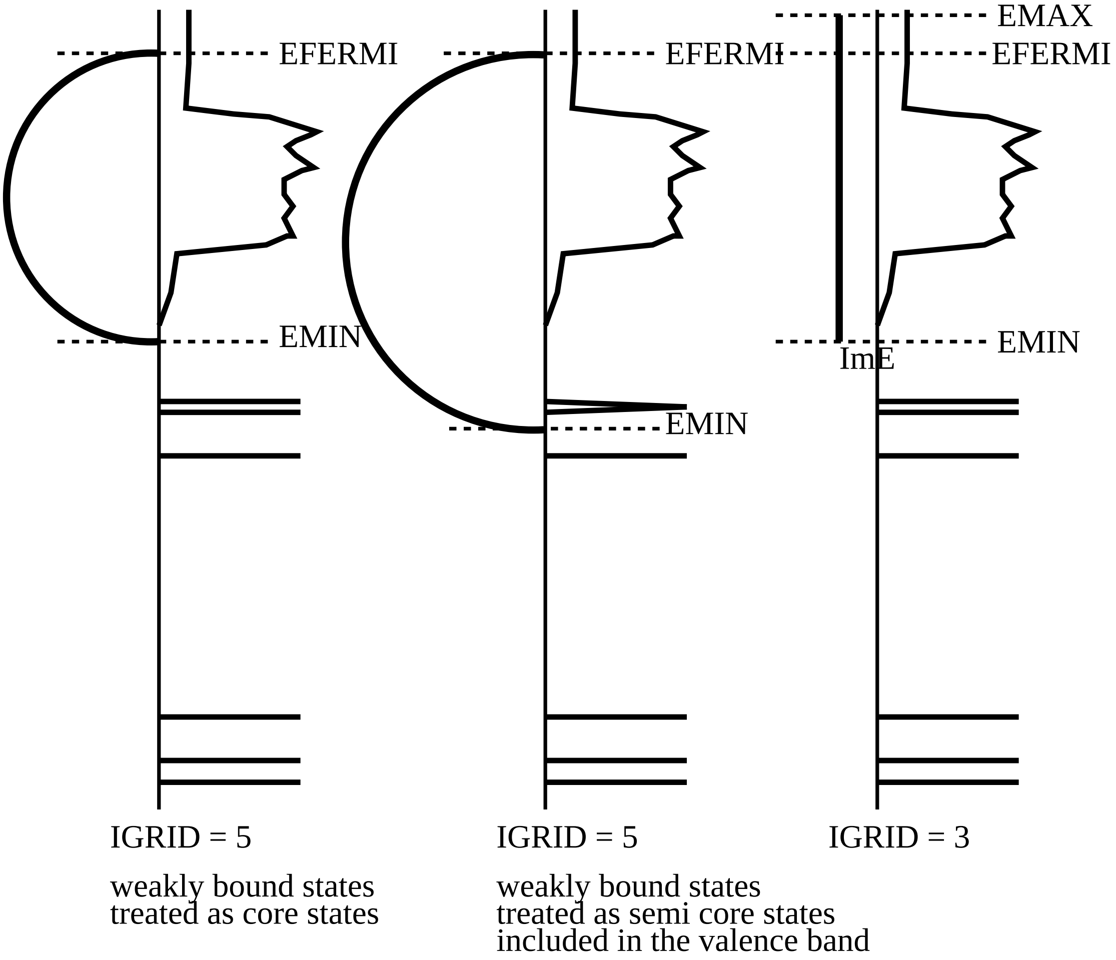


Figure 3.1: The various energy paths used for the SCF-calculations. The right part of each figure shows the electronic spectrum, while the left part shows a path in the complex energy plane; i.e. the distance from the vertical energy axis gives the imaginary part of the energy. The path shown on the right hand side runs parallel to the real axis and is used to fix the Fermi energy. The arc-like contour path in the complex energy plane is used for the energy integration.

iteration and the status is written to standard output. The first line after the $\star$-line gives the SCF iteration number, the largest rms errors (among the various atom types `IT`) for V and B (to be compared with `TOL`), the new Fermi energy E_F, its correction, the DOS at E_F and the total spin and orbital magnetic moments.

```
rms-error for type  1:  V =  2.1055E-03  ,  B =  2.0821E-02
iter.   6     average:  V =  2.1055E-03  ,  B =  2.0821E-02

*******************************************************************************
  6 ERR 0.211E-02 0.208E-01 EF  0.76199  0.00307 D  14.537 M   2.1740   0.0536
    ETOT      -2541.050268     SCF - cycle not converged    -    continue
*******************************************************************************
```

The convergence behavior may be checked by piping the output to a file DATASET.out and using the Unix command `grep " ERR " DATASET.out`. If convergence is achieved or the maximum number of iterations (`NITER`) is exhausted **kkrscf** stops. In the later case one can restart from the new potential file (see above). If there are problems with the convergence, one should first decrease the mixing parameter `MIX`. Using simple mixing at the beginning of the SCF cycle, may also help (increase `ISTBRY`).

Files used:

Filename	unit	I/O	description
DATASET.inp	5	I	input file described above
DATASET.pot	4	I	input potential read in by ⟨POTFIT⟩. This file is created via **xband** or stems from a previous run of **kkrscf**.
DATASET.pot_new	4	O	The new output potential is written after each SCF-iteration in ⟨MAIN⟩ of **kkrscf**.
DATASET_SCFSTART.dos	10	O	DOS for a path parallel to real axis for 0-th SCF-iteration if **kkrscf** starts from scratch (only for checking purposes). Opened in ⟨SCFINITPOT⟩ and written by ⟨CALCDOS⟩.
DATASET_ATOM.inf	8	O	Created if `IPRINT`$\geq$ 1: information on the atomic type calculations done to set up a guess for the charge density if **kkrscf** starts from scratch (only for checking purposes). Opened in ⟨SCFINITPOT⟩ and written by Desclaux core routines.

NOTE

The dummy potential file DATASET.pot can be used exclusively by **kkrscf**. After each SCF iteration the new potential is written to the output potential file DATASET.pot_new. This means in particular that **kkrscf** can be restarted if problems occurred after copying the file DATASET.pot_new to DATASET.pot. After the SCF convergence has been reached copy DATASET.pot_new to DATASET.pot. For all examples following section 3.1 it will be assumed that a SCF potential is available in the file DATASET.pot.

Example

Running **kkrscf** for the ordered compound FeCo in the CsCl-structure ($a \approx 5.365$) the input file created by **xband** should look like that:

```
################################################################################
#  SPR-KKR input file    FeCo_SCF.inp
#  created by xband on Sat Jan 15 18:46:13 CET 2005
################################################################################

CONTROL  DATASET = FeCo
         ADSI    = SCF
         POTFIL  = FeCo.pot
         PRINT = 0

TAU      BZINT= POINTS  NKTAB= 250

ENERGY   GRID={5}  NE={30}
         EMIN=-0.2   ImE=0.0 Ry

SCF      NITER=200 MIX=0.20 VXC=VWN
         TOL=0.00001  MIXOP=0.20  ISTBRY=1
```

As a first step **kkrscf** always creates a guess for the potential. Then the corresponding Fermi energy is determined from a DOS calculation for an energy path running parallel to the real energy axis until the required number of valence electrons is found, i.e. until the Fermi energy is reached. The DOS (with the energy given with respect to the muffin-tin zero) is written to the file FeCo_SCF_SCFSTART.dos and can be viewed using **plot** via **xband**. For the present example one gets the three files FeCo_SCF_SCFSTART.dos.agr, FeCo_SCF_SCFSTART.dos_Co.agr, and FeCo_SCF_SCFSTART.dos_Fe.agr, that give the total and partial DOS and that can be viewed by invoking **xmgrace**. As an example the component resolved DOS of FeCo and the corresponding partial DOS of Fe is shown in Fig. 3.2. Although the DOS is calculated for a non-self-consistent potential it should nevertheless look *reasonable*. In general the guess potential is quite good as can be seen from the DOS curves shown in Fig. 3.9, that are based on the converged potentials. However, if the DOS in SYSTEM_SCF_SCFSTART.dos looks *unreasonable* one should try to influence the set up of the guess for the potential – in particular to enforce a small charge transfer.

During the SCF-iterations one can monitor the progress by grepping the string "ERR" in the output file FeCo_SCF.out. This can be done via **xband** using the out-file menu giving the following table:

```
#####################################################
###  executing the unix command: grep "ERR" FeCo_SCF.out
#####################################################
###  Wait for the message     EXECUTION COMPLETED   ###
###  stop execution using red   STOP  button        ###
###  or  <Control-c>  in output field               ###
#####################################################
   1 ERR 0.646E+00 0.904E-01 EF  0.76638  0.01562 D  22.501 M   4.2321   0.1270
   2 ERR 0.195E+01 0.133E+00 EF  0.77586  0.00948 D  27.758 M   4.1812   0.1369
   3 ERR 0.105E+00 0.780E-01 EF  0.77433 -0.00152 D  39.760 M   4.1708   0.1288
   4 ERR 0.781E-01 0.672E-01 EF  0.77541  0.00108 D  33.596 M   4.2027   0.1302
   5 ERR 0.195E-01 0.330E-01 EF  0.77989  0.00448 D  27.410 M   4.2976   0.1307
```

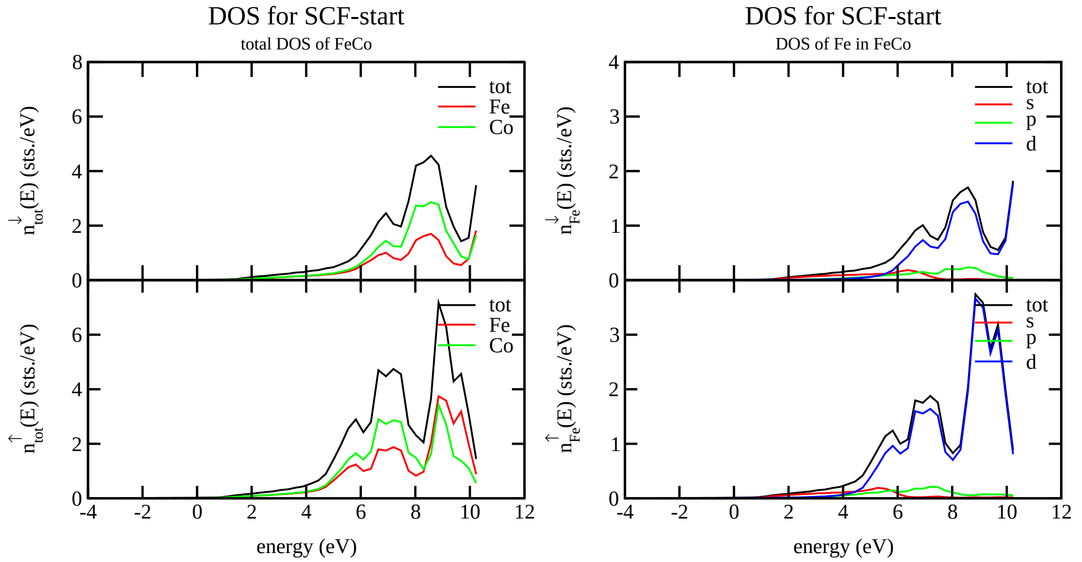


Figure 3.2: The component resolved DOS (left) and the spin and angular momentum resolved partial DOS of Fe (right) in FeCo for a starting potential created by guess.

```
     6 ERR 0.299E-02 0.180E-01 EF   0.78127  0.00137 D  30.354 M   4.3421   0.1309
     7 ERR 0.200E-02 0.105E-01 EF   0.78212  0.00085 D  29.733 M   4.3650   0.1308
     8 ERR 0.517E-02 0.187E-02 EF   0.78340  0.00128 D  28.407 M   4.3906   0.1306
     9 ERR 0.446E-02 0.211E-02 EF   0.78377  0.00037 D  28.911 M   4.4001   0.1306
    10 ERR 0.292E-02 0.144E-02 EF   0.78358 -0.00019 D  29.675 M   4.3956   0.1306
    11 ERR 0.326E-02 0.121E-02 EF   0.78370  0.00012 D  29.302 M   4.3977   0.1305
    12 ERR 0.296E-02 0.113E-02 EF   0.78374  0.00004 D  29.330 M   4.3984   0.1305
    13 ERR 0.240E-02 0.424E-03 EF   0.78407  0.00034 D  28.731 M   4.4037   0.1305
    14 ERR 0.128E-02 0.202E-03 EF   0.78416  0.00009 D  28.822 M   4.4047   0.1305
    15 ERR 0.579E-03 0.140E-03 EF   0.78414 -0.00002 D  28.935 M   4.4042   0.1305
    16 ERR 0.230E-03 0.801E-04 EF   0.78414  0.00000 D  28.911 M   4.4043   0.1305
    17 ERR 0.277E-04 0.374E-04 EF   0.78415  0.00001 D  28.908 M   4.4043   0.1305
    18 ERR 0.451E-04 0.361E-04 EF   0.78415  0.00000 D  28.912 M   4.4043   0.1305
    19 ERR 0.459E-04 0.341E-04 EF   0.78415  0.00000 D  28.912 M   4.4043   0.1305
    20 ERR 0.444E-04 0.321E-04 EF   0.78415  0.00000 D  28.912 M   4.4043   0.1305
    21 ERR 0.112E-04 0.416E-05 EF   0.78415  0.00000 D  28.912 M   4.4043   0.1305
    22 ERR 0.706E-05 0.155E-05 EF   0.78415  0.00000 D  28.914 M   4.4043   0.1305

##########################################################
###              EXECUTION COMPLETED                   ###
##########################################################
```

The various columns give the iteration number, the errors in the potential functions V and B (`ERR`), the Fermi energy E_F and its correction (`EF`), the total DOS $n(E_F)$ at the Fermi energy (`D`) and the total spin and orbital magnetic moments (`M`). If the variations are too wild, in particular if `ERR` does not decrease in the average, reduce the mixing parameter. The information extracted via grep is also written to the file FeCo_SCF.log to allow a direct graphical display using e.g. **xmgrace**. For that reason, the decadic logarithm of the errors in the potential functions V and B (`RMSAVV` and `RMSAVB` corresponding to `ERR`) is tabulated.

For the settings used in the example the SCF-cycle for FeCo is converged after 22 iterations. After convergence the following final result is written to the end of the output file FeCo_SCF.out:

```
*******************************************************************************

SPRKKR-run for: FeCo

results extrapolated to corrected FERMI - ENERGY:
CHARGE MISFIT       0.00000
E_F CORRECTION      0.00000
NEW FERMI ENERGY    0.78415   extrapolated


 31 E= 0.7842 0.0000         IT= 1  Fe
        DOS      NOS     P_spin   m_spin    P_orb    m_orb    B_val      B_core
 s    0.3690   0.6415    0.1235   0.0011  -0.00001  0.00000   44.12 s   -292.28
 p    0.5234   0.8120    0.3016  -0.0409   0.00501  0.00070    1.16 ns     1.24
 d   16.1427   6.5122  -13.0394   2.7266   0.41423  0.05753   21.98 cor -291.04
sum  17.0351   7.9657  -12.6143   2.6868   0.41923  0.05823   67.26 v+c -223.78
E_band         4.52188022 [Ry]
-------------------------------------------------------------------------------

 31 E= 0.7842 0.0000         IT= 2  Co
        DOS      NOS     P_spin   m_spin    P_orb    m_orb    B_val      B_core
 s    0.3585   0.6623    0.0079  -0.0060   0.00000  0.00000  -16.01 s   -194.18
 p    0.9308   0.7984   -0.0566  -0.0415   0.01734  0.00019    0.69 ns     0.90
 d   10.5895   7.5737   -8.1270   1.7651  -0.40534  0.07207   43.77 cor -193.28
sum  11.8787   9.0343   -8.1756   1.7176  -0.38801  0.07226   28.44 v+c -164.83
E_band         5.19735929 [Ry]
-------------------------------------------------------------------------------
TOT  28.9139  17.0000             4.4043           0.13049
E_band         9.71923951 [Ry]
===============================================================================

*******************************************************************************
                    setting up new potential in <SCFNEWPOT>
*******************************************************************************

    IT  JTOP   Z    QEL       DQT       DVCT     IQ    DQQ       VMAD
     1   721  26  25.96567  -0.03433   0.00000    1  -0.03433   0.02605
     2   721  27  27.03433   0.03433   0.00000    2   0.03433  -0.02605

    total energy       -5323.51256145

*******************************************************************************
         shift of muffin-tin zero  VMTZ  -0.789846
*******************************************************************************

    rms-error for type  1:  V =  2.6516D-06     B =  1.3912D-06
    rms-error for type  2:  V =  7.0629D-06     B =  1.5496D-06
    iter.  22     average:  V =  7.0629D-06     B =  1.5496D-06

    execution time for last iteration       202.740 secs


*******************************************************************************
 22 ERR 0.706E-05 0.155E-05 EF  0.78415  0.00000 D  28.914 M   4.4043   0.1305
    ETOT      -5323.51256145    SCF - cycle converged !!!!!!!!!
*******************************************************************************
```

3.2 Phase shift $\delta_\kappa(E)$

The calculation of the phase shift $\delta_\kappa(E)$ and the corresponding logarithmic derivative $D_\kappa(E)$ can be used for checking the potential, to get average values for the spin orbit and exchange splitting parameters and to get an overview for the range of bands (e.g. 3d–4d–5d). Although the phase shift concept can be applied also for the spin polarized relativistic case, leading to a corresponding matrix $\delta_{\Lambda\Lambda'}(E)$, this is not done here. Instead, the phase shift $\delta_\kappa(E)$ is first calculated ignoring the exchange splitting and then ignoring the spin orbit splitting.

To calculate the phase shift run **kkrgen** with an input and potential file supplied. The specific part of the input file DATASET.inp supplies the following parameters:

section TASK

VAR / SWITCH	default	description
`PSHIFT`	OFF	Calculate the phase shifts $\delta_\kappa(E)$ and logarithmic derivative $D_\kappa(E)$ for all atom types `IT` in the system.

section ENERGY

VAR / SWITCH	default	description
`NE`=integer	100	number of E-mesh points
`EMIN`=real	0.0001	lowest E-value
`EMAX`=real	1.0	highest E-value

If there is a resonance in the d- or f-chanel for energies below 1 Ry, the exchange and spin-orbit splitting parameters are written by **kkrgen** to standard output. These values represent energy independent average values. For the energy dependent spin-orbit splitting parameter see section 3.4. The energy dependent phase shift and logarithmic derivative are written to files, that can be used directly by **xmgrace**.

Files used:

Filename	unit	I/O	description

DATASET.inp	5	I	input file described below
DATASET.pot	4	I	input potential read in by ⟨POTFIT⟩.
DATASET_pshift_AT.agr	7	O	phase shift $\delta_\kappa(E)$ for the atom type AT written by ⟨PSHIFT⟩ and formatted for viewing directly with **xmgrace**
DATASET_logdrv_AT.agr	7	O	logarithmic derivative $D_\kappa(E)$ for the atom type AT written by ⟨PSHIFT⟩ and formatted for viewing directly with **xmgrace**

Example

To calculate the phase shifts $\delta_\kappa(E)$ and the corresponding logarithmic derivatives $D_\kappa(E)$ for the ordered compound FeCo in the CsCl-structure the input file created by **xband** should look like this:

```
###############################################################################
#  SPR-KKR input file    FeCo_PSHIFT.inp
#  created by xband on Mon Jan 17 21:46:44 CET 2005
###############################################################################

CONTROL  DATASET = FeCo
         ADSI    = PSHIFT
         POTFIL  = FeCo.pot
         PRINT = 0

ENERGY   GRID={3}  NE={100}
         EMIN=0.0001   EMAX=1.0   ImE=0.0 Ry

TASK     PSHIFT
```

The file FeCo.pot has to contain the (usually converged) potential created by **kkrscf**. Running **kkrgen** the phase shifts $\delta_\kappa(E)$ will be written to the files FeCo_PSHIFT_pshift_Fe.agr and FeCo_PSHIFT_pshift_Co.agr, while the logarithmic derivatives $D_\kappa(E)$ will be written to FeCo_PSHIFT_logdrv_Fe.agr and FeCo_PSHIFT_logdrv_Co.agr, respectively. These files can be viewed directly using **xmgrace**, i.e. there is no need for post-processing. As an example the phase shift and logarithmic derivatives of Fe in FeCo is shown in Fig. 3.3.

If the phase shift $\delta_\kappa(E)$ shows a resonance, as it is typical for d-electrons ($l = 2$) of transition metals, an average value for the exchange and spin-orbit splittings, ΔE_{xc} and ΔE_{so}, respectively, are determined and printed to the output file.

```
IT= 1  Fe

phase shift  written to the file FeCo_PSHIFT_pshift_Fe.agr

deduced from the resonance with E_res < 1 Ry for l = 2
SO-splitting    0.01092 Ry   0.14854 eV
```

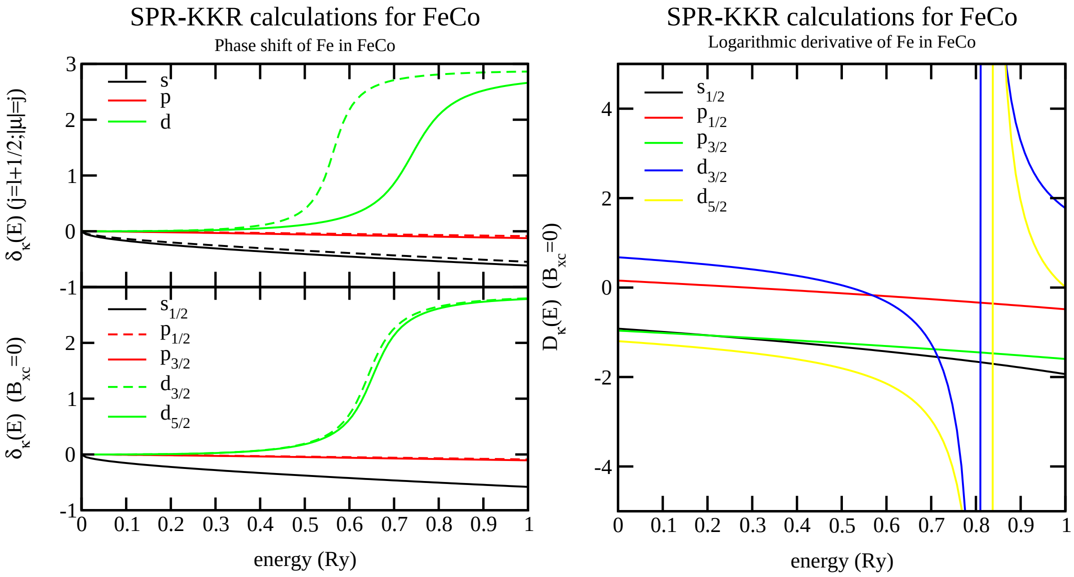


Figure 3.3: The phase shifts $\delta_\kappa(E)$ and the corresponding logarithmic derivatives $D_\kappa(E)$ of Fe in the ordered compound FeCo.

```
XC-splitting    0.18428 Ry   2.50724 eV

log. deriv.  written to the file FeCo_PSHIFT_logdrv_Fe.agr


IT= 2  Co

phase shift  written to the file FeCo_PSHIFT_pshift_Co.agr

deduced from the resonance with E_res < 1 Ry for l = 2
SO-splitting    0.01354 Ry   0.18416 eV
XC-splitting    0.12236 Ry   1.66474 eV

log. deriv.  written to the file FeCo_PSHIFT_logdrv_Co.agr
```

3.3 Plotting of wave functions

The valence band and core level wave functions used by the **SPRKKR** package may be ploted using **kkrgen**. The specific part of the input file DATASET.inp supplies the following parameters:

section TASK

VAR / SWITCH	default	description

`SOCPAR`	OFF	Calculate the spin-orbit-splitting parameters $\xi_l(E)$ as a function of the energy E for all atom types `IT` in the system.
`WFPLOT`	OFF	plot wave functions
`IT`	1	select atom type `IT`
`STATE`	BAND	BAND: valence band CORE: core level
`L`		s, p, d, ...-like wave function for STATE=BAND
`CL`		1s, 2s, 2p, ...-like wave function for STATE=CORE
`mj`	+1/2	+1/2, -1/2, +3/2, -3/2, ... m_j-character of wave function

section ENERGY

VAR / SWITCH	default	description
`NE`=integer	1	number of E-mesh points
`EMIN`=real	0.5	lowest E-value
`EMAX`=real	0.5	highest E-value

The energy dependent spin-orbit-splitting parameters $\xi_l(E)$ are written to files, that can be used directly by **xmgrace**.

Files used:

Filename	unit	I/O	description
DATASET.inp	5	I	input file described below
DATASET.pot	4	I	input potential read in by ⟨POTFIT⟩.
DATASET_*_AT.agr	7	O	data file for valence band or core wave functions, that can directly be viewed with **xmgrace**

Example

The input file to plot the d-like wave functions of Fe in FeCo should look like this:

```
################################################################################
#  SPR-KKR input file    FeCo_WFPLOT.inp
#  created by xband on Sat Jan 22 17:45:12 CET 2005
################################################################################

CONTROL  DATASET = FeCo
         ADSI    = WFPLOT
         POTFIL  = FeCo.pot
         PRINT = 0

ENERGY   GRID={3}  NE={1}
         EMIN=0.5   EMAX=0.5   ImE=0.0 Ry

TASK     WFPLOT     IT=1  (Fe)
         STATE=BAND  L=d    mj=+1/2    # E taken from  [ENERGY]
#        STATE=CORE  CL=2p  mj=+1/2
```

The run of **kkrgen** creates the wave function files to be viewed directly via **xmgrace**. A run with the input file shown above and a second one for the 2p-core levels leads to the results shown in Fig. 3.4.

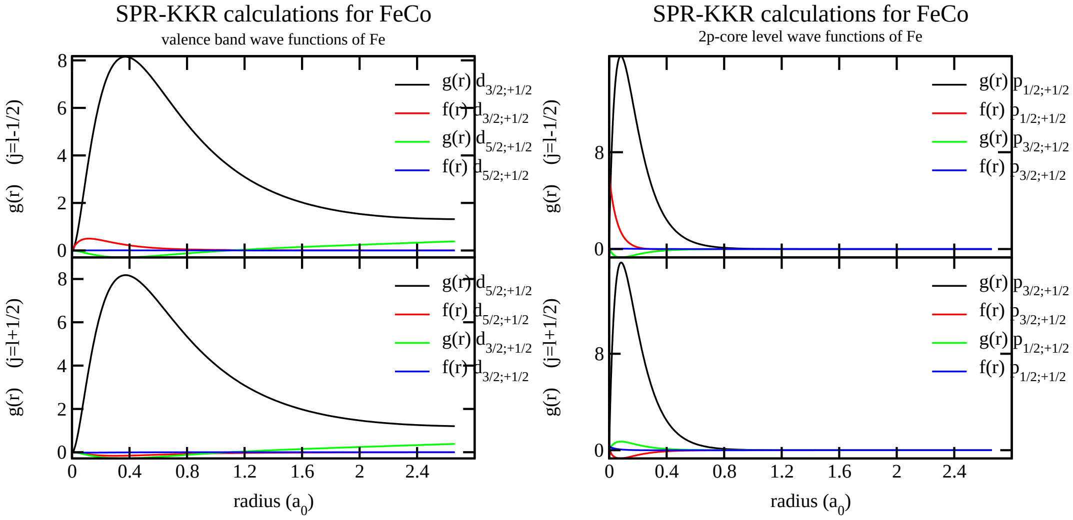


Figure 3.4: The valence band d-electron (left) and 2p-core state (right) wave function for Fe in FeCo. The core wave functions have been created in a second run of **kkrgen**.

3.4 Spin-orbit parameter

The spin-orbit-splitting parameter $\xi_l(E)$ depends on the energy E and can be calculated l-resolved using **kkrgen** with an input and potential file supplied. The specific part of the input file DATASET.inp supplyies the following parameters:

section TASK

VAR / SWITCH	default	description
`SOCPAR`	OFF	Calculate the spin-orbit-splitting parameters $\xi_l(E)$ as a function of the energy E for all atom types `IT` in the system.

section ENERGY

VAR / SWITCH	default	description
`NE`=integer	100	number of E-mesh points
`EMIN`=real	0.0001	lowest E-value
`EMAX`=real	1.0	highest E-value

The energy dependent spin-orbit-splitting parameters $\xi_l(E)$ are written to files, that can be used directly by **xmgrace**.

Files used:

Filename	unit	I/O	description
DATASET.inp	5	I	input file described below
DATASET.pot	4	I	input potential read in by ⟨POTFIT⟩.
DATASET_soc_AT.agr	7	O	spin-orbit-splitting parameters $\xi_l(E)$ for the atom type AT written by ⟨SOCPAR⟩ and formatted for viewing directly with **xmgrace**

Example

To calculate the spin-orbit-splitting parameter $\xi_l(E)$ for the ordered compound FeCo the input file created by **xband** should look like this:

```
###############################################################################
#   SPR-KKR input file    FeCo_SOCPAR.inp
#   created by xband on Mon Jan 17 22:05:32 CET 2005
###############################################################################

CONTROL  DATASET = FeCo
         ADSI    = SOCPAR
         POTFIL  = FeCo.pot
         PRINT = 0

ENERGY   GRID={3}  NE={100}
         EMIN=0.0001   EMAX=1.0   ImE=0.0 Ry

TASK     SOCPAR
```

The file FeCo.pot has to contain the (usually converged) potential created by **kkrscf**. Running **kkrgen** the energy-dependent spin-orbit-splitting parameter $\xi_l(E)$ will be written to the files FeCo_SOCPAR_soc_Fe.agr and FeCo_SOCPAR_soc_Co.agr. In addition an effective energy-dependent exchange splitting ΔE_{xc} is written to the files FeCo_SOCPAR_exc_Fe.agr and FeCo_SOCPAR_exc_Co.agr. All these files can be viewed directly using **xmgrace**, i.e. there is no need for post-processing. As an example the spin-orbit-splitting parameter of Fe in FeCo is shown in Fig. 3.5.

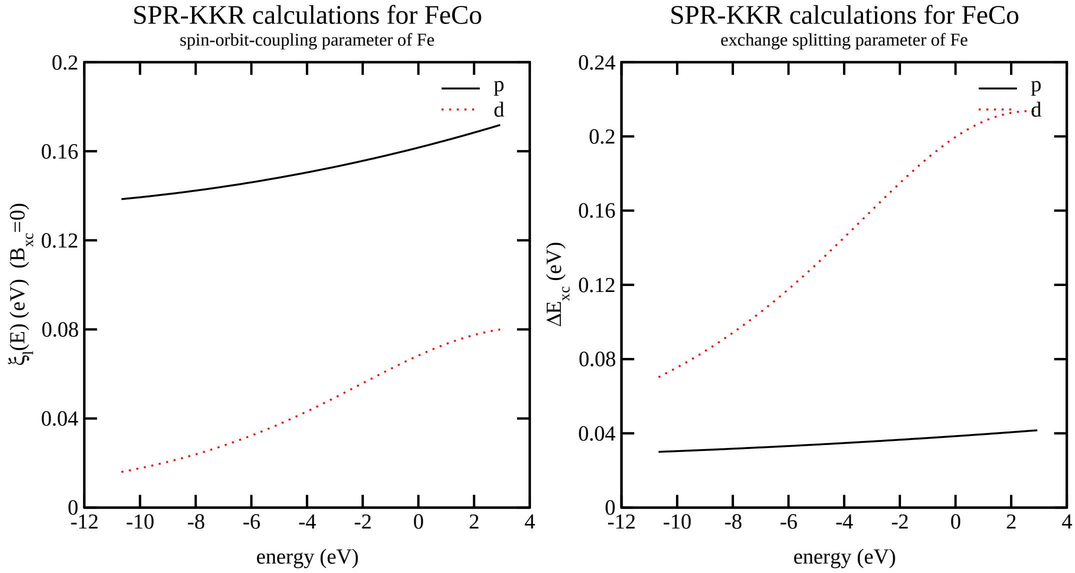


Figure 3.5: The energy dependent spin-orbit-splitting and exchange splitting parameters $\xi_l(E)$ and ΔE_{xc}, repectively, for p- and d-electrons of Fe in FeCo.

3.5 Dispersion relation $E(\vec{k})$

To calculate the dispersion relation $E(\vec{k})$ use **kkrgen** with an appropriate input and potential file. The specific part of the input file supplies the following parameters:

section TASK

VAR / SWITCH	default	description
`EKREL`	OFF	Calculate the dispersion relation $E(\vec{k})$ for an ordered system.
`EMIN`=real	-0.1	energy range for
`EMAX`=real	1.0	$E(\vec{k})$ relation
`NE` =integer	1000	number of E-points – fixes tolerance for $E(\vec{k})$
`NK`=integer	51	total number of $\vec{k}$-points
`KPATH`=integer	-	predefined path in $\vec{k}$-space see table 3.1
`NKDIR`=integer	1	directions in $\vec{k}$-space treated
`KA`$\star$={x,y,z}	{0,0,0}	first and last $\vec{k}$-vector for segment $\star$
`KE`$\star$={x,y,z}	{1,0,0}	in $\vec{k}$-space in multiples of $2\pi/a$ and rectangular coordinates with $\star = 1, ...,$`NKDIR`

The dispersion relation $E(\vec{k})$ is calculated for an energy range fixed by the variables `EMIN` and `EMAX` with the accuracy determined by the energy step `(EMAX-EMIN)/(NE-1)`. There are two ways to specify the corresponding path in $\vec{k}$-space:

- select a predefined path by using the variable `KPATH`. See the list above for the available settings.
- specify the number `NKDIR` of segments of a user defined path and give for all segments the first and last $\vec{k}$-vectors.

kkrgen determines the dispersion relation $E(\vec{k})$ from the number $N(E)$ of positive eigen values of the KKR-matrix. For are given $\vec{k}$ $N(E)$ is calculated for every energy value of the mesh specified by `NE`, `EMIN`, and `EMAX`. An energy eigen value is indicated by a change in $N(E)$ for increasing E. **kkrgen** uses the constant-E mode to calculate the dispersion relation. This may lead to rather long execution times.

kkrgen writes the result to a file DATASET.bnd. Use **plot** to convert the data to a **xmgrace** file.

Files used:

Bravais lattice	`KPATH`	path
orb	1	Γ-Σ-X-G-U-A-Z-Λ-Γ-Δ-Y-H-T-B-Z + X-D-S-C-Y + U-P-R-E-T + S-Q-T
	2	Γ-Σ-X-G-U-A-Z-Λ-Γ-Δ-Y-H-T-B-Z
	3	Γ-Σ-X-G-U-A-Z-Λ-Γ
	4	Γ-Δ-Y-H-T-B-Z
hex	1	Γ-Σ-M-T′-K-T-Γ-Δ-A-R-L-S′-H-S-A + M-U-L + K-P-H
	2	Γ-Σ-M-T′-K-T-Γ-Δ-A-R-L-S′-H-S-A
	3	Γ-Σ-M-T′-K-T-Γ-Δ-A
	4	Γ-Σ-M
	5	K-T-Γ
sc	1	Γ-Δ-X-Y-M-V-R-Λ-Γ-Σ-M
	2	Γ-Δ-X-Y-M-V-R-Λ-Γ
	3	Γ-Δ-X-Y-M-V-R
	4	Γ-Δ-X-Y-M
fcc	1	X-Δ-Γ-Λ-L-Q-W-N-K-Σ-Γ + L-M-U-S-X-Z-W-D-U
	2	X-Δ-Γ-Λ-L-Q-W-N-K-Σ-Γ
	3	X-Δ-Γ-Λ-L
	4	Γ-Δ-X
	5	Γ-Λ-L
bcc	1	Γ-D-H-G-N-Σ-Γ-Λ-P-F-H + N-D-P
	2	Γ-D-H-G-N-Σ-Γ-Λ-P-F-H
	3	Γ-D-H-G-N-Σ-Γ-Λ-P
	4	Γ-D-H-G-N-Σ-Γ
	5	Γ-D-H

Table 3.1: Parameter `KPATH` used to specify various paths in $\vec{k}$-space for the different Bravais lattices.

Filename	unit	I/O	description
DATASET.inp	5	I	input file described below
DATASET.pot	4	I	input potential read in by ⟨POTFIT⟩.
DATASET.bnd	10	O	dispersion relation $E(\vec{k})$ together with information on the path in $\vec{k}$-space. Use **plot** to obtain the corresponding **xmgrace** file.

Example

To calculate the density of states for the ordered compound FeCo the input file created by **xband** should look like this:

```
###############################################################################
#  SPR-KKR input file    FeCo_EKREL.inp
#  created by xband on Sat Jan 22 17:54:33 CET 2005
###############################################################################

CONTROL  DATASET = FeCo
         ADSI    = EKREL
         POTFIL  = FeCo.pot
         PRINT = 0

TASK     EKREL   EMIN=-0.1  EMAX=1.0 Ry  NE=600
         NK = 200    KPATH = 4
```

The file FeCo.pot has to contain the (usually converged) potential created by **kkrscf**. Running **kkrgen** the density of states data will be written to the file FeCo_DOS.dos. The content of the file can be viewed using **plot** via **xband**. For the present example one gets the three files FeCo_DOS.dos.agr, FeCo_DOS.dos_Co.agr, and FeCo_DOS.dos_Fe.agr, that give the total and partial DOS and that can be viewed by invoking **xmgrace**. As an example the partial DOS of Fe is shown in Fig. 3.9. Note that the energy range is given in the input file in units of Ry with respect to the muffin-tin zero. For the display of the DOS the energy range is converted to eV with respect to the Fermi energy, i.e. $E = 0$ corresponds to $E = E_F$.

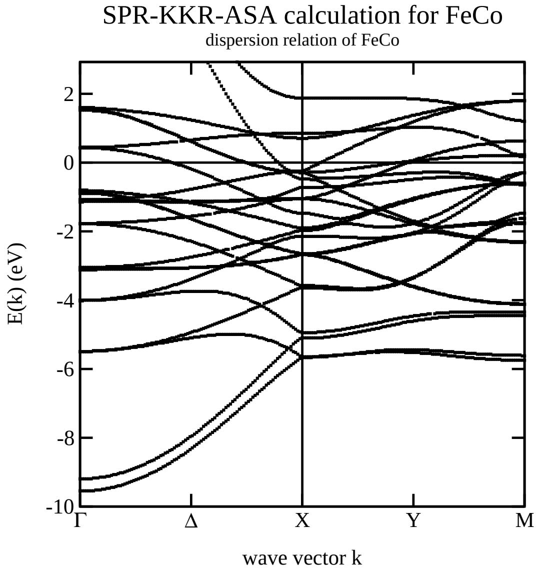


Figure 3.6: The dispersion relation $E(\vec{k})$ of FeCo for the wave vector $\vec{k}$ along the path $\Gamma - X - M$.

3.6 Bloch spectral function $A_B(E,\vec{k})$

The Bloch spectral function $A_B(E,\vec{k})$ can be seen as a $\vec{k}$-resolved DOS function [13]. For an ordered system it is a δ-like function, that carries the same information as the dispersion relation $E(\vec{k})$. Calculating $A_B(E,\vec{k})$ for an ordered system at complex energies is therefore on alternative way to represent $E(\vec{k})$, with a broadening according to the imaginary part of E. For a disordered systems $E(\vec{k})$ is not well defined, while $A_B(E,\vec{k})$ can still be used to represent the electronic band structure. The Bloch spectral function $A_B(E,\vec{k})$ is obtained by running **kkrgen** with appropriate potential and input files. The specific part of the input file DATASET.inp supplyies the following parameters:

section TASK

VAR / SWITCH	default	description
`BLOCHSF`	OFF	Calculate the Bloch spectral function $A_B(E,\vec{k})$.

section ENERGY

VAR / SWITCH	default	description
`NE`=integer	-	number of E-mesh points
`EMIN`=real	-	lowest E-value
`EMAX`=real	-	highest E-value
`ImE`=real	0.01	imaginary part of E
`NK`=integer	51	total number of $\vec{k}$-points
`KPATH`=integer	-	predefined path in $\vec{k}$-space. See section 3.5 for a list of available settings.
`NKDIR`=integer	1	directions in $\vec{k}$-space treated
`KA`$\star$={x,y,z}	{0,0,0}	first and last $\vec{k}$-vector for segment $\star$
`KE`$\star$={x,y,z}	{1,0,0}	in $\vec{k}$-space in multiples of $2\pi/a$ and rectangular coordinates with $\star = 1, ...,$`NKDIR`
`NK1`=integer	-	number of $\vec{k}$-vectors along $\vec{k}_1$.
`NK2`=integer	-	number of $\vec{k}$-vectors along $\vec{k}_2$.
`K1`={x,y,z}	{1,0,0}	first $\vec{k}$-vector to span a two-dimensional region in $\vec{k}$-space

`K2`={x,y,z}	{0,1,0}	second $\vec{k}$-vector to span a two-dimensional region in $\vec{k}$-space

.

The Bloch spectral function $A_B(E,\vec{k})$ may be calculated for:

- a certain range of the energy E along a path in $\vec{k}$-space. In this case the input parameters are chosen as for the calculation of the dispersion relation $E(\vec{k})$. The energy mesh is specified by the parameters `NE`, `EMIN`, and `EMAX`, while there are two ways to specify the corresponding path in $\vec{k}$-space:
  - select a predefined path by using the variable `KPATH`. See the list above for the available settings.
  - specify the number `NKDIR` of segments of a user defined path and give for all segments the first and last $\vec{k}$-vectors.
- a fixed energy E and a rectangular region in $\vec{k}$-space. In this case the region is specified via two spanning vectors `K1` and `K2` with `NK1` and `NK2` grid points along $\vec{k}_1$ and $\vec{k}_2$. Typically the energy is set to the Fermi energy E_F leading to a cut through the Fermi surface.

Calculation of the Bloch spectral function $A_B(E,\vec{k})$ for real energies is sensible only for systems with chemical disorder, that usually are treated using the CPA. In this case the CPA equations has to be solved first for the required energy mesh. The resulting CPA scattering path operator $\tau^{CPA}(E)$ and the inverse of the corresponding single site t-matrix $(t^{CPA}(E))^{-1}$ will be stored in a file DATASET.tau. Ordered systems can be treated by working at complex energies (set `ImE` to a finite value).

Files used:

Filename	unit	I/O	description
DATASET.inp	5	I	input file described below
DATASET.pot	4	I	input potential read in by ⟨POTFIT⟩.
DATASET.tau	9	I/O	contains the non-vanishing elements of the site-dependent scattering path operator matrix $\tau^q_{\Lambda\Lambda'}$ and inverse site t-matrix $(t^q(E))^{-1}_{\Lambda\Lambda'}$ for the specified energy grid. Written in ⟨PROJTAU⟩.
DATASET.bsf	10	O	Bloch spectral function $A_B(E,\vec{k})$ together with information on the energy and $\vec{k}$ range. Use **plot** to obtain a corresponding graphics file. Because **xmgrace** cannot handle 3D-graphics output is generated for the alternate graphics program **plotmtv** and **xmatrix**.

Example

To calculate the spin-resolved Bloch spectral function (BSF) for bcc-Fe the input file created by **xband** should look like this:

```
###############################################################################
#  SPR-KKR input file    Fe_BLOCHSF.inp
#  created by xband on Thu Jun  2 16:02:14 CEST 2005
###############################################################################

CONTROL  DATASET = Fe
         ADSI    = BLOCHSF
         POTFIL  = Fe.pot
         PRINT = 0

TAU      BZINT= POINTS  NKTAB= 250

ENERGY   GRID={3}  NE={260}
         EMIN=-0.2   EMAX=1.0   ImE=0.001 Ry

TASK     BSF
         NK = 260    KPATH = 5
```

The file Fe.pot has to contain the (usually converged) potential created by **kkrscf**. Running **kkrgen** the BSF will be written to the file Fe_BLOCHSF.bsf, which then can be visualised using **plot** from within **xband**.

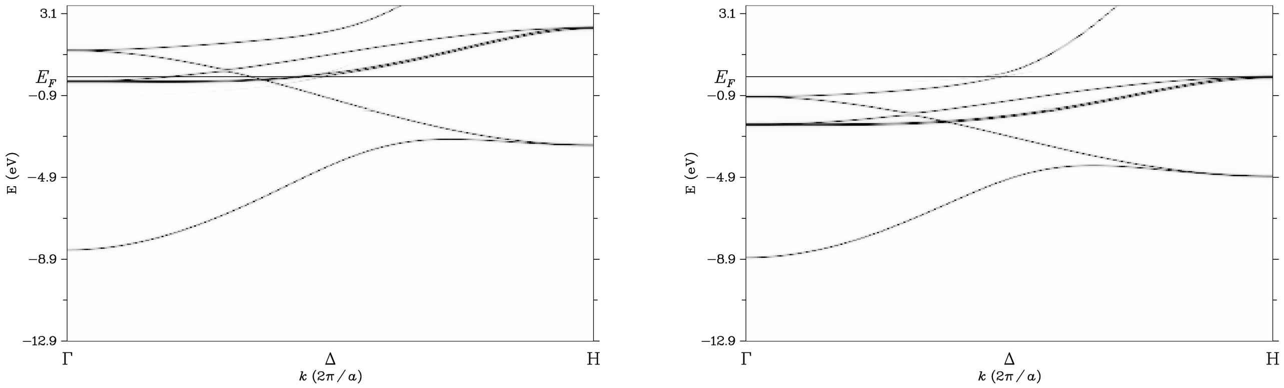


Figure 3.7: Spin-up (left) and spin-down (right) BSF for bcc-Fe.

Furthermore, one can also calculate different cuts through Fermi surfaces. For this, the **xband** created inputfile should look like this:

```
###############################################################################
#  SPR-KKR input file    Fe_FERMI.inp
#  created by xband on Fri Jun  3 13:45:34 CEST 2005
###############################################################################

CONTROL  DATASET = Fe
         ADSI    = BLOCHSF
         POTFIL  = Fe.pot
         PRINT = 0

TAU      BZINT= POINTS  NKTAB= 250

ENERGY   GRID={3}  NE={1}
         EMIN=0.7458405583   EMAX=0.7458405583   ImE=0.001 Ry

TASK     BSF
         NK1 = 260    K1 = {1.0, 0.0, 0.0 }
         NK2 = 260    K2 = {0.0, 1.0, 0.0 }
```

Note, that the parameters EMIN and EMAX must be set to the Fermi energy which is obtained from the converged potential file. created by **kkrscf**. Running **kkrgen** the data will be written to the file Fe_BLOCHSF.bsf, which then can be visualised using **plot** from within **xband**.

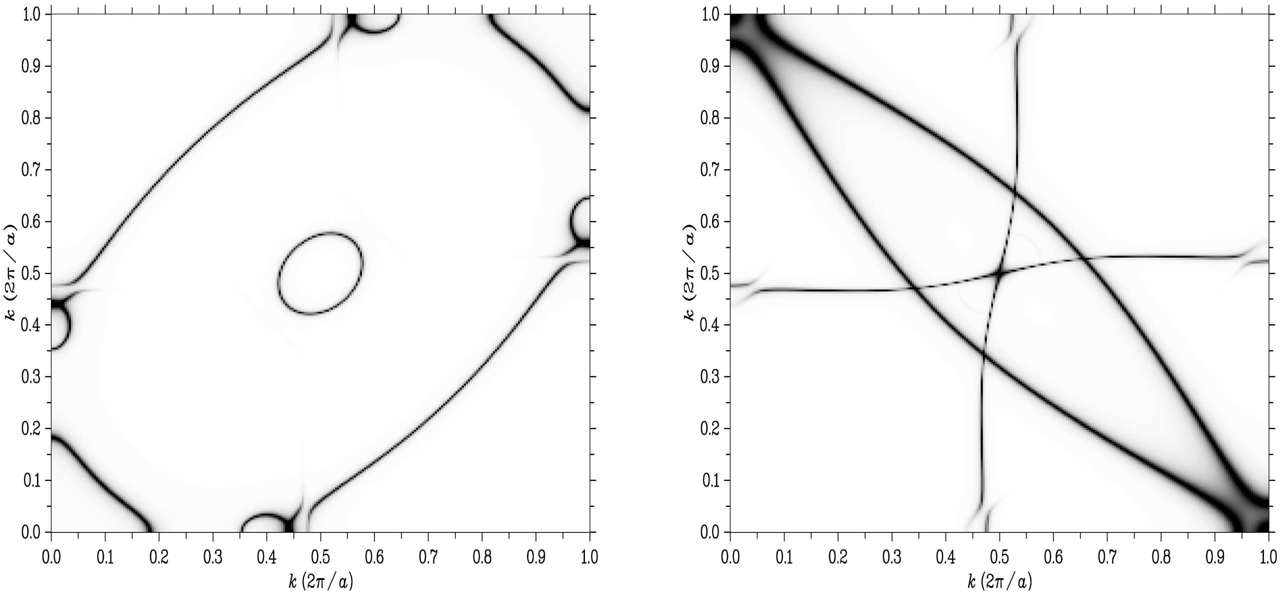


Figure 3.8: Cut by Γ-H×Γ-H-plane through the Fermi surface of bcc-Fe. Spin-up and spin-down BSF are shown on the left and right panels, respectively.

3.7 Density of States $n(E)$

To calculate the angular momentum, spin and component resolved density of states (DOS) $n(E)$ run **kkrgen** supplying an input and potential file. In addition the κ-resolved DOS and the spin and orbital polarizations, `P_spin`= $d\langle\sigma_z\rangle/dE$ and `P_orb`= $d\langle l_z\rangle/dE$, can be obtained.

The specific part of the input file DATASET.inp supplyies the following parameters:

section TASK

VAR / SWITCH	default	description
`DOS`	ON	Calculate the DOS

section ENERGY

VAR / SWITCH	default	description
`NE`=integer	100	number of E-mesh points
`EMIN`=real	-0.2	real part of lowest E-value
`EMAX`=real	1.2	real part of highest E-value
`ImE`=real	0.01	imaginary part of E

See Fig. 3.1 for a graphical representation of the various paths in the complex energy plane.

section CONTROL

VAR / SWITCH	default	description
`WRKAPDOS`	OFF	write l- and κ-resolved DOS
`WRPOLAR`	OFF	write spin- and orbital polarization
`WRTAU`	OFF	write τ-matrix τ_t projected for all components `IT` of the system to file DATASET.tau for later use
`WRTAUMQ`	OFF	write τ-matrix τ_q and inverse t-matrix m_q for all sites `IQ` of the system to file DATASET.tau for later use

kkrgen writes the following information to standard output:

```
 34 E= 0.4061 0.0100          IT= 1  Ti
               DOS  [1/Ry]  |  m_spin  [m_B]  |  m_orb   [m_B]  |   B_tot   [kG]
INT(DE) crystal    1.814            0.771             0.009           1056.0
TOTAL   crystal    3.641            0.099             0.064           3661.1
INT(DE) backscat.  -0.317          -0.734             0.014          -1862.1
TOTAL   backscat.  -0.689          -0.548             0.046           3567.5
--------------------------------------------------------------------------------
```

This block is printed for every energy and component `IT` and gives the DOS, spin-polarization `P_spin=` $d\langle\sigma_z\rangle/dE$, orbital polarization `P_orb=` $d\langle l_z\rangle/dE$, and valence band hyperfine field in their integrated (`INT(DE)`) and differential form. The full values (`crystal`) as well as their back scattering parts are given separately. These are obtained from Eq. (1.11) by ignoring the second irregular part and replacing $\tau^{nn'}_{\Lambda\Lambda'}(E)$ by $\tau^{nn'}_{\Lambda\Lambda'}(E)-t^{n}_{\Lambda\Lambda'}(E)$. For `IPRINT` ≥ 1 a decomposition according to the magnetic quantum number μ is given in addition.

kkrgen stores the calculated DOS in the file DATASET.dos. Use **plot** to convert the data and write them to **xmgrace**-compatible files *dos*.agr. The program **plot** creates for every component `IT` of a system a file with its l- and spin resolved DOS. For a multi-component system in addition a file with the total concentration weighted DOS curves is created.

Calculating the DOS or, equivalently, the scattering path operator τ using a BZ-integration, a finite imaginary part of the energy has to be used. The smaller `ImE` the denser the $\vec{k}$-mesh has to be, to get smooth DOS curves. A cluster calculation, on the other hand, can be done for a vanishing `ImE`. However, the cluster size has to be the larger the smaller `ImE` is to achieve convergency.

For checking the potential it is often helpful to calculate the single site DOS (obtained by replacing τ by t). This avoids a lengthy BZ-integration and can be achieved by setting `BZINT=WEYL`, `NKMIN=0`, and `NKMAX=0` in section `KMESH`.

Files used:

Filename	unit	I/O	description
DATASET.inp	5	I	input file described below
DATASET.pot	4	I	input potential read in by ⟨POTFIT⟩.
DATASET.dos	10	O	angular momentum, spin and component resolved DOS formatted to be passed to **plot**, that creates corresponding **xmgrace**-compatible files.
DATASET.kap	13	O	If `WRKAPDOS` has been set: l- and κ- and component-resolved DOS, i.e., s-, p-, d-, ..., $s_{1/2}$-, $p_{1/2}$-, $p_{3/2}$-, ... like DOS for the specified energy grid written in ⟨CALCDOS⟩. Use **plot** to convert the data and write them to **xmgrace**-compatible files *kapdos*.agr.

DATASET.pol	14	O	If `WRPOLAR` has been set: spin- and orbital polarization for the specified energy grid written in ⟨CALCDOS⟩. Use **plot** to convert the data and write them to **xmgrace**-compatible files *polar*.agr.
DATASET.tau	9	O	contains the non-vanishing elements of the scattering path operator matrix $\tau_{\Lambda\Lambda'}$ for the specified energy grid. Written in ⟨PROJTAU⟩. • If `WRTAU` has been set: component-projected τ-matrix τ_t for all components `IT` of the system. • If `WRTAUMQ` has been set: τ-matrix and inverse t-matrix, τ_q and m_q, for all sites `IQ` of the system.

Example

To calculate the density of states for the ordered compound FeCo in the CsCl-structure ($a \approx 5.365$) the input file created by **xband** should look like this:

```
###############################################################################
#  SPR-KKR input file    FeCo_DOS.inp
#  created by xband on Sun Jan 16 19:48:53 CET 2005
###############################################################################

CONTROL  DATASET = FeCo
         ADSI    = DOS
         POTFIL  = FeCo.pot
         PRINT = 0

TAU      BZINT= POINTS  NKTAB= 250

ENERGY   GRID={3}  NE={50}
         EMIN=-0.2   EMAX=1.0   ImE=0.01 Ry

TASK     DOS
```

The file FeCo.pot has to contain the (usually converged) potential created by **kkrscf**. Running **kkrgen** the density of states data will be written to the file FeCo_DOS.dos. The content of the file can be viewed using **plot** via **xband**. For the present example one gets the three files FeCo_DOS.dos.agr, FeCo_DOS.dos_Co.agr, and FeCo_DOS.dos_Fe.agr, that give the total and partial DOS and that can be viewed by invoking **xmgrace**. As an example the partial DOS of Fe is shown in Fig. 3.9. Note that the energy range is given in the input file in units of Ry with respect to the muffin-tin zero. For the display of the DOS the energy range is converted to eV with respect to the Fermi energy, i.e. $E = 0$ corresponds to $E = E_F$.

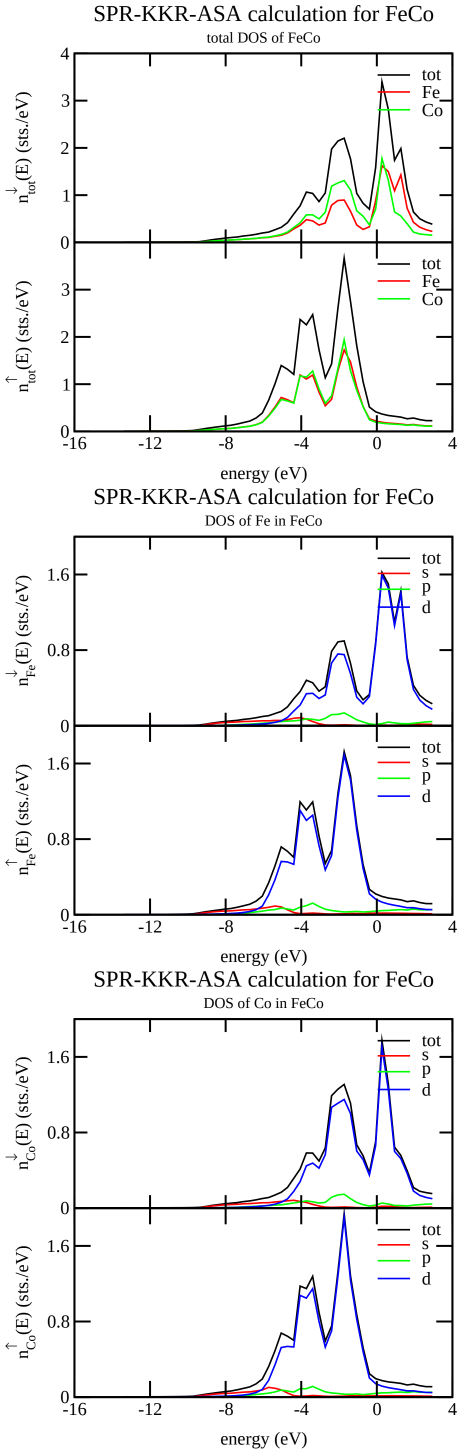


Figure 3.9: Top: total and partial DOS of FeCo. Below: The spin and angular momentum resolved partial DOS of Fe and Co in FeCo.

4 Calculating ground state properties

4.1 Magnetic Moments and Hyperfine Fields

The spin and orbital magnetic moments as well as the magnetic hyperfine fields [14] of all atomic types are calculated by **kkrscf** during every SCF cycle by performing a corresponding energy integral in the complex plane (see for example Eqs. (1.14) and (1.15)). Because the integrated DOS (NOS) for the last energy mesh point may differ from the required valence electron number an update of the Fermi energy is made and extrapolated results are written to standard output

```
 31 E= 0.7720 0.0000          IT= 1  Fe
        DOS      NOS     P_spin   m_spin    P_orb    m_orb    B_val      B_core
 s    0.3188   0.6384    0.0732  -0.0004    0.00000  0.00000   12.22 s   -151.99
 p    0.6935   0.8009   -0.0166  -0.0198    0.01013 -0.00006    0.11 ns     0.70
 d   21.4229   6.5607   -3.7996   1.9005   -0.06931  0.04993   20.98 cor -151.30
sum  22.4352   8.0000   -3.7429   1.8803   -0.05919  0.04987   33.31 v+c -117.99
-------------------------------------------------------------------------------
TOT  22.4352   8.0000             1.8803             0.04987
```

Columns 2, 4, 6 and 7 give in a l-resolved way the partial charge, spin moment, orbital moment, and valence hyperfine field of atomic type `IT`. The last column gives the s-, non-s- and total core hyperfine field as well as the total field (`v+c`).

The same results, but with a somewhat different format, can be obtained by running **kkrgen** for a given potential. The specific part of the input file may look like this

```
ENERGY   GRID={5} NE={30} EMIN=-0.2

TASK     DOS
```

supplying some of the following parameters

section TASK

VAR / SWITCH	default	description
`DOS`	ON	Calculate the DOS

section ENERGY

VAR / SWITCH	default	description
`GRID`=integer	-	5: circular energy path in complex plane
`NE`=integer	-	number of E-mesh points
`EMIN`=real	-0.2	real part of lowest E-value
`EMAX`=real	-	real part of highest E-value
`EF`=real	-	Fermi energy E_F
`SEARCHEF`	OFF	search E_F using `EF` as a first guess

If a circular path in the complex energy plane has been chosen (`GRID`=5), the upper end of the path (`EMAX`) is set to the Fermi energy `EF`. If the Fermi energy is not supplied via the potential file (for example if the potential is imported) **kkrgen** may search the proper Fermi energy. For this purpose the switch `SEARCHEF` has to be set and a reasonable guess for the Fermi energy `EF` has to be supplied.

4.2 Disordered Local Moment

To do DLM calculations that simulate temperature effects for the magnetisation, a CPA system should be set up with the help of **xband**.

For each given site, specify two types, both with the same element but with different spin. Their concentrations have to sum up to 1. Typically one might be interested in x(up) = x(down) = 0.5 (paramagnetic state).

To specify the spin, set the input variable `MSPIN` as mentioned in chapter 3.1. Use opposite signs for spin up and spin down for the two types of each site. **xband** can be used to set `MSPIN`: clicking on the `SCF` checkbox in the "create input" expert mode will show several options, `MSPIN` being one of them.

This method creates a metastable start potential. To prevent the system from flipping all spins to the same direction and relaxing to the global energy minimum (e.g. ferromagnetic state) during the SCF cycle, lower the SCF mixing parameter `MIX` to stabilize the local minimum of the metastable phase until you can converge your system in the desired phase.

section SCF

VAR / SWITCH	default	description

`MSPIN`=$\{\mu_1,\ldots,\mu_{\texttt{NT}}\}$	*table*	guess for the spin magnetic moment $\mu_{spin,t}$ for `NT` atomic types, use opposite signs for types occupying one site
`MIX`=real	0.20	SCF-mixing parameter, adapt to stabilize metastable phase

4.3 Magnetic Form Factor

Magnetic form factors, that may be used to discuss neutron or high energy X-ray scattering experiments, are calculated using **kkrgen** by making use of the so-called dipole approximation [15]. The specific part of a corresponding input file supplies some of the following parameters

section TASK

VAR / SWITCH	default	description
`FMAG`	OFF	Calculate the spin and orbital densities and corresponding magnetic form factors f

section ENERGY

VAR / SWITCH	default	description
`GRID`=integer	-	5: circular energy path in complex plane
`NE`=integer	-	number of E-mesh points
`EMIN`=real	-0.2	real part of lowest E-value

kkrgen writes the resulting spin, orbital and total magnetic densities and corresponding form factors formatted for **xmgrace**.

Files used:

Filename	unit	I/O	description

DATASET.inp	5	I	input file described below
DATASET.pot	4	I	input potential read in by ⟨POTFIT⟩.
DATASET_dens_AT.agr	80	O	spin, orbital and total magnetic densities for the atom type AT written by ⟨DENSITY⟩ and formatted for viewing directly with **xmgrace**
DATASET_fmag_AT.agr	80	O	spin, orbital and total magnetic form factor for the atom type AT written by ⟨DENSITY⟩ and formatted for viewing directly with **xmgrace**

Example

To calculate the magnetic form factors for the ordered compound Fe_3Pt the input file created by **xband** should look like this:

```
################################################################################
#  SPR-KKR input file    PtFe_FMAG.inp
#  created by xband on Mon May 30 11:53:52 CEST 2005
################################################################################

CONTROL  DATASET = PtFe3
         ADSI    = FMAG
         POTFIL  = PtFe3.pot
         PRINT = 0

TAU      BZINT= POINTS  NKTAB= 250

ENERGY   GRID={5}  NE={30}
         EMIN=-0.2 Ry

TASK     FMAG
```

The file FePt3.pot has to contain the (usually converged) potential created by **kkrscf**. Running **kkrgen** the magnetic form factors will be written to files PtFe3_FMAG_fmag_AT.agr, which then can be visualised using **xmgrace** or using **plot** from within **xband**. Furthermore the files PtFe3_FMAG_dens_AT.agr will be written which contain the charge-, spin- and orbital-density curves for the different atom types.

4.4 Exchange coupling parameters J_{ij}

Using the SPR-KKR package it is possible to calculate the exchange coupling parameters J_{ij} of the Heisenberg model:

$$H_{ex} = -\sum_{i \neq j} J_{ij} \mathbf{e}_i \mathbf{e}_j \ ,$$

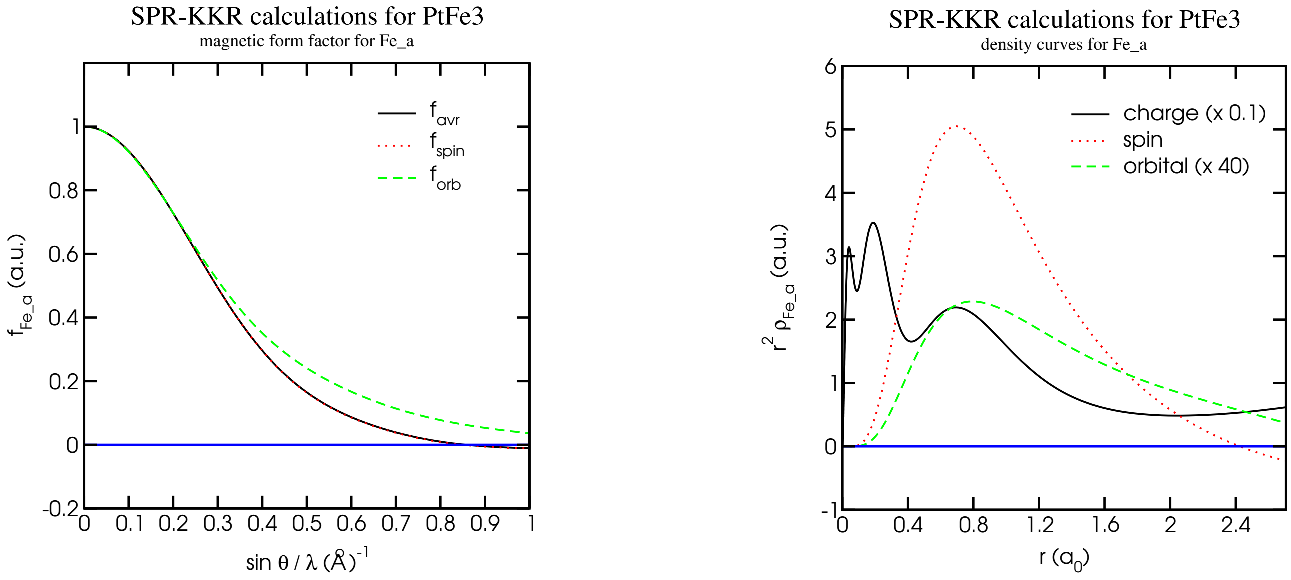


Figure 4.1: Magnetic form factors (left) and density curves (right) for Fe in the ordered compound Fe_3Pt.

where $\mathbf{e}_i$ and $\mathbf{e}_j$ are unit vectors having the directions of corresponding local magnetic moments on sites i and j. The exchange coupling parameter J_{ij} can be found from ab initio calculations based on the KKR Green's function method using the formulation of Lichtenstein et al. [J. Phys. F: Met. Phys., 14, L125 (1984)]:

$$J_{ij} \;=\; \frac{1}{4\pi} Im \int^{E_F} dE \;\text{Trace}\;(t^{-1}_{i\uparrow} - t^{-1}_{i\downarrow})\tau^{ij}_{\uparrow}(t^{-1}_{j\uparrow} - t^{-1}_{j\downarrow})\tau^{ji}_{\downarrow}$$

The site i in these calculations is assumed to be positioned in the centre of a cluster of radius $R_{clu} = max|R_i - R_j|$. The exchange coupling parameters are calculated with respect to the central site i of a cluster with the radius R_{clu}. This calculation can be done running **kkrgen** with an input and potential file supplied.

The specific part of the input file supplies the following parameters:

section TASK

VAR / SWITCH	default	description
`JXC`	OFF	Calculate the J_{ij}
`CLURAD`=integer	2.2	the radius of a sphere restricting the cluster of atoms for which the exchange coupling constants will be calculated

Files used:

Filename	unit	I/O	description
DATASET.inp	5	I	input file described below
DATASET.pot	4	I	input potential read in by ⟨POTFIT⟩.
DATASET_J_ij.dat	80	O	contains the values of the exchange coupling between the spin magnetic moment placed at the centre of a cluster with magnetic moment of each atom in the cluster restricted by the value `CLURAD` - the cluster radius.
DATASET_Jij_IT.agr	80	O	Dependence of exchange coupling parameters on the distance between spin magnetic moments. Can be viewed with the program **xmgrace**.

The output file contains in the end the same table of exchange coupling parameters as the DATASET.dat file as well as the values of $J_0 = \sum_j J_{0j}$ parameters for each nonequivalent atom in a unit cell:

```
*******************************************************************************
                                   <XCCPLJ0>
                           XC-coupling constants J_0
*******************************************************************************

    IT =   1  Ni     J0 =   0.003486 Ry    0.047428 eV

    IQ =   1        J0 =   0.003486 Ry    0.047428 eV
    occupation     IT =   1
    equiv. sites   IQ =   1

    Curie temperature within mean field approximation  T_C = 387.7 K


*******************************************************************************
                                  <XCCPLJIJ>
                           XC-coupling constants J_ij
*******************************************************************************


    IQ =  1 IT =  1                     JQ =  1 JT =  1
       ->Q = (  0.000,  0.000,  0.000)        ->Q = (  0.000,  0.000,  0.000)
    ITAUIJ ITAUJI   N1 N2 N3    DRX    DRY    DRZ     DR     J_ij [Ry]  J_ij [eV]
          1     12   -1  0  0   0.000 -0.500 -0.500   0.707    0.000167   0.002268
          2     11   -1  0  1   0.500  0.000 -0.500   0.707    0.000167   0.002268
          3     10   -1  1  0   0.500 -0.500  0.000   0.707    0.000164   0.002234
```

Example

To calculate the exchange coupling constants for Fe the input file created by **xband** should look like this:

```
################################################################################
#   SPR-KKR input file    Fe_JXC.inp
#   created by xband on Sat May 14 22:16:45 CEST 2005
################################################################################

CONTROL  DATASET = Fe
         ADSI    = JXC
         POTFIL  = Fe.pot
         PRINT = 0

TAU      BZINT= POINTS  NKTAB= 1000

ENERGY   GRID={5}  NE={30}
         EMIN=-0.2 Ry

TASK     JXC   CLURAD=4.7
```

The file Fe.pot has to contain a converged potential created by **kkrscf**. Running **kkrgen** the exchange coupling constants will be written to the file Fe_JXC_J_ij.dat which looks like:

```
 ********************************************************************************
                                   <XCCPLJIJ>
                            XC-coupling constants J_ij
 ********************************************************************************


          number of sites   NQ =
             1   1  1 1.000

     IQ =  1 IT =  1                     JQ =  1 JT =  1
        ->Q = (  0.000,  0.000,  0.000)        ->Q = (  0.000,  0.000,  0.000)
     ITAUIJ ITAUJI   N1 N2 N3    DRX    DRY    DRZ     DR     J_ij [Ry]  J_ij [eV]
           1      8   -1 -1 -1  -0.500 -0.500 -0.500   0.866    0.001368   0.018610
           2      7   -1  0  0   0.500 -0.500 -0.500   0.866    0.001368   0.018610
           3      6    0 -1  0  -0.500  0.500 -0.500   0.866    0.001368   0.018610
           4      5    0  0 -1  -0.500 -0.500  0.500   0.866    0.001368   0.018610
           5      4    0  0  1   0.500  0.500 -0.500   0.866    0.001368   0.018610
           6      3    0  1  0   0.500 -0.500  0.500   0.866    0.001368   0.018610
           7      2    1  0  0  -0.500  0.500  0.500   0.866    0.001368   0.018610
           8      1    1  1  1   0.500  0.500  0.500   0.866    0.001368   0.018610
           9     14   -1 -1  0   0.000  0.000 -1.000   1.000    0.000770   0.010473
          10     13   -1  0 -1   0.000 -1.000  0.000   1.000    0.000762   0.010371
          11     12    0 -1 -1  -1.000  0.000  0.000   1.000    0.000762   0.010371
          12     11    0  1  1   1.000  0.000  0.000   1.000    0.000762   0.010371
          13     10    1  0  1   0.000  1.000  0.000   1.000    0.000762   0.010371
          14      9    1  1  0   0.000  0.000  1.000   1.000    0.000770   0.010473
          15     26   -2 -1 -1   0.000 -1.000 -1.000   1.414   -0.000034  -0.000466
          16     25   -1 -2 -1  -1.000  0.000 -1.000   1.414   -0.000034  -0.000466
          17     24   -1 -1 -2  -1.000 -1.000  0.000   1.414   -0.000034  -0.000465
          18     23   -1  0  1   1.000  0.000 -1.000   1.414   -0.000034  -0.000466
          19     22   -1  1  0   1.000 -1.000  0.000   1.414   -0.000034  -0.000465
          20     21    0 -1  1   0.000  1.000 -1.000   1.414   -0.000034  -0.000466
          21     20    0  1 -1   0.000 -1.000  1.000   1.414   -0.000034  -0.000466
          22     19    1 -1  0  -1.000  1.000  0.000   1.414   -0.000034  -0.000465
          23     18    1  0 -1  -1.000  0.000  1.000   1.414   -0.000034  -0.000466
          24     17    1  1  2   1.000  1.000  0.000   1.414   -0.000034  -0.000465
          25     16    1  2  1   1.000  0.000  1.000   1.414   -0.000034  -0.000466
          26     15    2  1  1   0.000  1.000  1.000   1.414   -0.000034  -0.000466
```

If there are more than one atom in a unit cell, the file Fe_JXC_J_ij.dat will contain also the corresponding tables for different combinations `IQ, IT` and `JQ, JT`. The numbers `N1,`

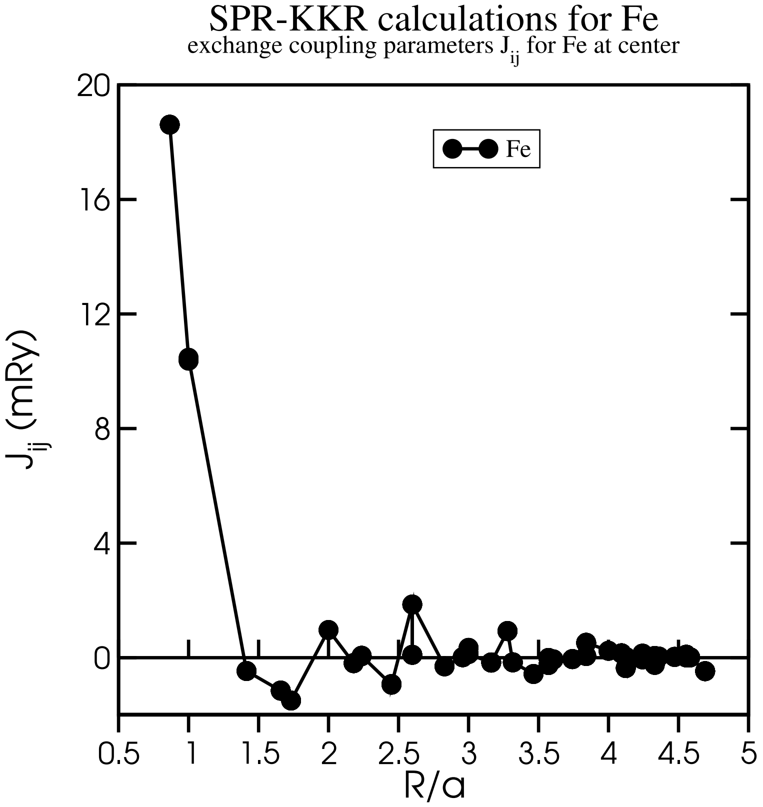


Figure 4.2: The exchange coupling parameter of Fe as a function of the distance R between the spin magnetic moments.

`N2, N3` in the table are the crystallographic coordinates of local magnetic moments in a cluster interacting with the central one, while `DRX, DRY, DRZ` are their Cartesian coordinates. The column denoted by `DR` contains the distances from central atom to each atom of a cluster. Both the Cartesian coordinates and distances are represented in units of lattice constants. The last two columns in the table denoted by `J_ij [Ry]` and `J_ij [eV]` contain the values of exchange coupling parameters represented in Ry and eV respectively.

Also the corresponding **xmgrace** - compatible file Fe_JXC_Jij_Fe.agr will be created, which can be viewed by invoking **xmgrace**. This file contains the plot of exchange coupling parameters vs. the distance between the spin magnetic moments which is shown in Fig. 4.2.

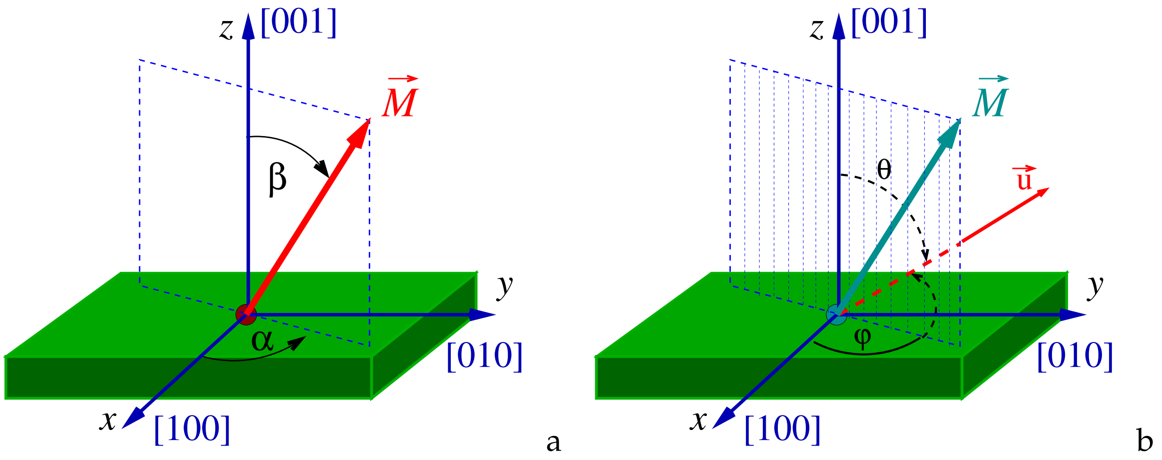


Figure 4.3: Geometry for torque calculations

4.5 Magnetic torque $T_{\vec{u}}$

The magneto-crystalline anisotropy can be investigated via magnetic torque calculations [16, 17]. This method has obvious advantages when compared to the total energy calculations since the MCA can be obtained from a single calculation and reliance is not placed on the accurate extraction of a small difference between two energies [16]. The magnetic torque $\vec{T}_i^{(\hat{n}_i)} = -\partial E(\{\hat{n}_k\})/\partial \hat{n}_i \times \hat{n}_i$ acting on the magnetic moment $\vec{m}_i$ of the atomic site i, with a unit vector $\hat{n}_i = \vec{m}_i/|\vec{m}_i|$ pointing along the direction of the magnetization $\vec{M}$ (Fig. 4.3a). The component of the magnetic torque with respect to the axis $\hat{u}$

$$T_{\hat{u}}(\theta, \phi) \;=\; -\partial E(\vec{M}(\theta, \phi))/\partial \theta \qquad (4.1)$$

is calculated in accordance to [16, 17]. Here, the $\hat{u}$ vector specified by the angles θ and ϕ (see Fig. 4.3b) lies within the surface plane and is perpendicular to the direction of the magnetic moment $\hat{n}$. For an uniaxial anisotropy a special geometry can be used which gives a simple relationship between the magnetic torque and the energy difference between the in-plane and out-of-plane magnetization directions. Setting $\theta = \pi/4$, the torque component $T_{\hat{u}}$ gives the ϕ dependent energy difference $T_{\hat{u}}(\theta = \pi/4, \phi) = E_{||}(\phi) - E_{\perp}$ [18, 16, 19].

The implementation within the SPR-KKR package of magnetic torque calculations is based on [16]. The result printed out in the output file corresponds to the derivative

$$\tilde{T}_{\hat{u}}(\theta, \phi) \;=\; \partial E(\vec{M}(\theta, \phi))/\partial \theta \; , \qquad (4.2)$$

which can be used to obtain the magnetic torque value

$$T_{\hat{u}}(\theta,\phi) \;=\; -\tilde{T}_{\hat{u}}(\theta,\phi)\ . \qquad (4.3)$$

The specific part of the input file supplies the following parameters:

section TASK

VAR / SWITCH	default	description
`TORQUE`	OFF	Calculate magnetic torque
`THETAQ`=real		
`PHIQ`=real		the angles characterizing orientation of the direction $\hat{u}$

section MODE

VAR / SWITCH	default	description
`MALF`=real		
`MBET`=real		
`MGAM`=real		the angles characterizing the orientation of the magnetic moment direction $\hat{n}$

section ENERGY

VAR / SWITCH	default	description
`GRID`=integer	-	8: circular energy path in complex plane
`NE`=integer	-	number of E-mesh points
`EMIN`=real	-0.2	real part of lowest E-value

Files used:

Filename	unit	I/O	description

DATASET.inp	5	I	input file described below
DATASET.pot	4	I	input potential

(Taking `MALF, MBET, MGAM` and `THETAQ, PHIQ` angles, pay attention that $\hat{u}$ direction is orthogonal to the plane of rotation of the magnetization direction (Fig. 4.3).)

Running **kkrgen** the results are printed out to the output file. The output file contains in the end the site-resolved magnetic torque values in Ry as well as total torque both in Ry and meV:

```
 ##############################################################################
           Calculation of magnetic torque
 ##############################################################################


          reading TAU-file for:   TAUQ(IQ)  and    mss(IQ)
          number of tabulated energies    16
          format type                      3



          Site resolved magnetic torque:

          IQ =  1          0.0000099517
          IQ =  2         -0.0000032936


          Total:

                  0.0000066581     Ryd
                  0.0905885007     meV
```

Example

To calculate the magnetic torque for Fe-Ni alloy the input file should look like this:

```
###############################################################################
#  SPR-KKR input file    Torque.inp
#  created by xband on Wed Sep 16 09:00:26 CEST 2009
###############################################################################

CONTROL  DATASET = FeNi
         ADSI    = Torque
         POTFIL  = FeNi.pot_new
         PRINT = 0

SITES    NL={4}

TAU      BZINT= POINTS  NKTAB= 1000

CPA      TOL=0.000001
```

```
ENERGY   GRID={8}  NE={16}
         EMIN=-0.2 Ry

MODE     MALF={0.0} MBET={45.0} MGAM={0.0}

TASK     TORQUE THETAQ={90.0} PHIQ={90.0}
```

The file FeNi.pot has to contain a converged potential created by **kkrscf** for magnetization direction along $\hat{z}$ axis (default directions), or, the same, along $[0, 0, 1]$ direction. One has to be sure that `KMROT = 0` in this file. One has to use the energy mesh `GRID=8`, where the number of energy points `NE=16` gives satisfying results. One can use more dense energy mesh, e.g., with 24 or 32 energy points. In the case of alloys, the CPA tolerance has to be taken 1E-7, both, within the SCF calculations to create converged potential as well as for torque calculations. The number of k points used for torque calculations should be large enough to get convergence with respect to this value.

At the end of the output file (see the example above), the total torque is printed out both in Ry and meV. Site-resolved magnetic torque is presented in Ry for Fe site (IQ = 1) and for Ni site (IQ = 2). A positive sign of these values means the torque direction toward $\hat{z}$ or $[0, 0, 1]$ direction.

As an example, the results are presented for Fe-Co alloy [20] for random bct $Fe_{0.4}Co_{0.6}$ alloy as a function of the tetragonal strain c/a. For all calculations the following input was used:

```
################################################################################
#  SPR-KKR input file    Torque.inp
#  created by xband on Wed Sep 16 09:00:26 CEST 2009
################################################################################

CONTROL  DATASET = FeCo
         ADSI    = Torque
         POTFIL  = FeCo.pot_new
         PRINT = 0

SITES    NL={4}

TAU      BZINT= POINTS  NKTAB= 1000

ENERGY   GRID={8}  NE={16}
         EMIN=-0.2 Ry

MODE     MALF={0.0} MBET={45.0} MGAM={0.0}

TASK     TORQUE THETAQ={90.0} PHIQ={90.0}
```

The results are compared with total energy calculations in Fig. 4.4. Positive values of the results in the output file mean the torque direction toward $\hat{z}$ direction.

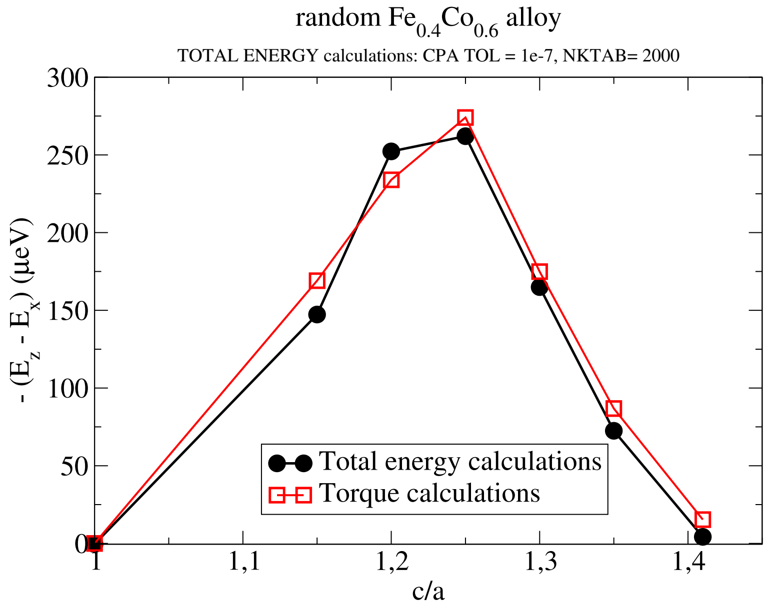


Figure 4.4: MCA for for random bct $Fe_{0.4}Co_{0.6}$ alloy as a function of the tetragonal strain c/a.

5 Linear response calculations

5.1 Magnetic susceptibility

The magnetic susceptibility as a linear response to a static uniform external magnetic field can be calculated for pure-element and alloy systems using **kkrchi**. Calculations have to be done in fully relativistic mode (default).

The specific part of the input includes the parameters

```
TASK     CHI
```

supplying the following parameters

section TASK

VAR / SWITCH	default	description
`CHI`	OFF	perform susceptibility calculation

Files used:

Filename	unit	I/O	description
DATASET.inp	5	I	input file described above
DATASET.pot	4	I	input potential read in by ⟨POTFIT⟩.

5.1.1 Example

To calculate the magnetic susceptibility χ for Pt fcc one has to use the input files created by **xband**. Thus, the input file DATASET.inp can look like:

```
CONTROL  DATASET     = Pt
         ADSI        = CHI
         POTFIL      = Pt.pot
         PRINT = 0
         CHIPRINT = 0

TAU      BZINT= POINTS  NKTAB= 1000

SITES    NL = {4}

ENERGY   GRID={5}  NE={32}
         EMIN=-0.2

TASK     CHI
```

The results are stored in the output file DATASET.out, the last table which looks like:

```
 *******************************************************************************
                results  INCLUDING  additional high-field terms
 *******************************************************************************


 ===============================================================================

          results for atom type  IT= 1  Pt


          polarisations at the Fermi level

                            DOS         P_spin    P_orb(dn)   P_orb(up)
                         22.651641    0.000000   -1.526865    1.526865
                     s    0.262696    0.000000   -0.000010    0.000010
                     p    0.563146    0.000000   -0.012859    0.012859
                     d   21.538038    0.000000   -1.542544    1.542544
                     f    0.287761    0.000000    0.028547   -0.028547


          magnetic susceptibility

          CHI tot    :     147.566783 x 10^(-6) cm^3/mol

          CHI S      :     128.267455

          CHI SS     :     119.574044      0:   54.023239     XC:   65.550805
                     s       0.348482            0.504399           -0.155917
                     p      -0.061200            1.154739           -1.215939
                     d     118.376231           51.725905           66.650326
                     f       0.910531            0.638196            0.272335

          ICHI(S)    :       0.548203

          CHI SO     :       8.693412      0:    7.868861     XC:    0.824550
```

```
CHI O      :      61.425144

CHI OO     :      43.668291      0:   39.481738     XC:    4.186553
           s       0.000024            0.000022            0.000003
           p       2.378283            2.235666            0.142617
           d      40.134021           36.151214            3.982807
           f       1.155964            1.094837            0.061127
        dn s       0.000012            0.000011            0.000001
           p       1.189141            1.117833            0.071308
           d      20.067010           18.075607            1.991403
           f       0.577982            0.547418            0.030563
        up s       0.000012            0.000011            0.000001
           p       1.189141            1.117833            0.071308
           d      20.067010           18.075607            1.991403
           f       0.577982            0.547418            0.030563

ICHI(O)    :       0.095872

CHI OS     :      17.756853      0:    7.868861     XC:    9.887992

CHI DIA    :     -42.125816   core:  -15.655950     cb:  -26.469867
           s                                              -3.106915
           p                                              -3.841841
           d                                             -18.582767
           f                                              -0.938330
```

It gives the following results:
- `CHI tot` is the total magnetic susceptibility `CHI S` + `CHI O` + `CHI DIA`
- `CHI S` is the total spin magnetic susceptibility including enhanced spin susceptibility `CHI SS` and SOC induced spin susceptibility `CHI SO`
- `CHI O` is the total Van Vleck orbital susceptibility including enhanced orbital susceptibility `CHI OO` and SOC induced orbital susceptibility `CHI OS`
- `CHI DIA` is the Langevin orbital (diamagnetic) susceptibility consisting of core and band electron contributions.

The panel `0:` gives un-enhanced susceptibility values and the panel `XC:` gives the enhancement due to exchange (for spin) or orbital polarization (for orbital) effects.

In the case of alloys, there are two tables at the end of DATASET.out file, giving the element resolved magnetic susceptibilities. The total magnetic susceptibility is printed out at the end of the file

```
  ==============================================================================

        total magnetic susceptibility     181.883636 x 10^(-6) cm^3/mol

 ==============================================================================
```

5.2 Gilbert damping parameter

5.2.1 Gilbert damping at $T = 0$ K

The Gilbert damping parameter α can be calculated using **kkrchi**. The calculations at $T = 0K$ can be performed only for alloy systems. The specific part of the input includes the parameters

```
SITES    NL={4}

ENERGY   GRID={3} NE={1}

TASK     Gilbert
```

supplying the following parameters

section TASK

VAR / SWITCH	default	description
`Gilbert`	OFF	Perform calculations of the Gilbert damping parameter

Important issues for Gilbert damping α calculations:
- One has to use the fully relativistic mode (i.e. default mode);
- One has to use a large amount of k-points in the BZ. To find the required number of k points (`NKTAB` in the input file), one has to investigate a convergence of the results (i.e., calculated α) with respect to this number;
- The `NL` value is recommended to be taken not less than `NL = 4` in the case of d-systems (that corresponds to the cutoff $l_{max} = 3$ for the angular momentum expansion of the Green's function), `NL = 5` in the case of f-systems (i.e. $l_{max} = 4$), etc.

Files used:

Filename	unit	I/O	description
DATASET.inp	5	I	Input file described above
DATASET.pot	4	I	Input potential read in by ⟨POTIO⟩.

The results are printed to the file DATASET.out. The **non-normalized** total Gilbert damping parameter per unit cell are printed out as follows (see below)

```
Gilbert damping parameter ALPHA

              total            term 0         term 1

 (-VC)        XXTOT            XX0             XX1
 (+VC)        YYTOT            YY0             YY1
```

where `XXTOT` represents the result obtained without vertex corrections `(-VC)` and `YYTOT` - the result with vertex corrections `(+VC)` taken into account. The latter value, `YYTOT`, scaled by the factor g_{eff}/μ_{tot} gives the value of the Gilbert damping parameter α, where $g_{eff} = 2(1 + \mu_{orb}/\mu_{spin})$ is given in terms of the spin and orbital moments, μ_{spin} and μ_{orb}, respectively, and the total magnetic moment $\mu_{tot} = \mu_{spin} + \mu_{orb}$.

The element resolved spin (SMT = μ_{spin}) and orbital (OMT = μ_{orb}) moments, g-factor (GEFF = $g_{eff} = 2(1+\mu_{orb}/\mu_{spin})$) as well as the scaling factors (G/MEFF = g_{eff}/μ), for the alloy $\mathrm{A}_x\mathrm{B}_y$ are given in a table like this:

```
magnetic moment information   with  GEFF = 2(SMT+OMT)/SMT

IQ   IO   IT  TXT     CONC     SMT       OMT         GEFF   G/MEFF
 1    1    1   A        x      mA_s      mA_o          gA     SA
 2    1    2   B        y      mB_s      mB_o          gB     SB
average (site)                 <m_s>     <m_s>      <g_eff>  <S>
average (sum)              Sum xi*mi_s Sum xi*mi_o  <g_eff>  <SS>
```

The element projected Gilbert damping components of the α tensor, namely α_{xx} and α_{yy} components are given in a table like this: table

```
site and component resolved ALPHA (including factor g_eff / m_eff)

                 alfa (+VC)           alfa (-VC)           alfa (0)
IQ  TXT          xx        yy         xx        yy         xx        yy
1    A           XXA+     YYA+        XXA-     YYA-       XXA0      YYA0
2    B           XXB+     YYB+        XXB-     YYB-       XXB0      YYB0
average (site)  <XX+>     <YY+>       <XX->     <YY->     <XX0>     <YY0>
average (sum)   <XXS+>    <YYS+>      <XXS->    <YYS->    <XXS0>    <YYS0>
```

The results to compare with the experiment are `<XXS+> and <YYS+>` given in the field `average (sum)`

Example

To calculate the Gilbert damping parameter α for $\mathrm{Fe}_{0.5}\mathrm{Co}_{0.5}$ one has to use the input files created by **xband**. The input file DATASET.inp might look like:

```
CONTROL  DATASET = FeCo
         ADSI    = Gilbert
         POTFIL  = FeCo_Gilbert.pot
         PRINT = 0

SITES    NL={3}
```

```
# NL=3 is used only for demonstration

CPA      TOL=0.0000001

TAU      BZINT= POINTS  NKTAB= 1000
# small NKTAB is used only for demonstration

ENERGY   GRID={3} NE={1}

TASK      Gilbert
```

The results are written to the output file DATASET.out with the last table looking like this:

```
          Gilbert damping parameter ALPHA

                      total            term 0         term 1
          (-VC)     -0.0080082371      0.1543060039  -0.1623142410
          (+VC)      0.0013826213      0.1543060039  -0.1529233825


          to be scaled by factor g_eff / m_eff


     magnetic moment information   with  GEFF = 2(SMT+OMT)/SMT

     IQ   IO   IT  TXT     CONC    SMT     OMT     GEFF   G/MEFF
      1    1    1  Fe     0.500   2.530   0.056   2.045   0.791
      1    2    2  Co     0.500   1.730   0.086   2.099   1.156
     average (site)               2.130   0.071   2.067   0.939
     average (sum)                2.130   0.071   2.067   0.939


     site and component resolved ALPHA (including factor g_eff / m_eff)

                      alfa (+VC)           alfa (-VC)           alfa (0)
     IQ  TXT          xx        yy         xx       yy         xx       yy
      1  Fe         0.000625  0.000625   -0.00713 -0.00713    0.12787  0.12787
      1  Co         0.002284  0.002284   -0.00809 -0.00809    0.16983  0.16983
     average (site)  0.001454  0.001454   -0.00761 -0.00761    0.14885  0.14885
     average (sum)   0.001298  0.001298   -0.00752 -0.00752    0.14491  0.14491
```

5.2.2 Gilbert damping at finite temperature

The Gilbert damping parameter α can be calculated also for $T \neq 0K$. These calculations are allowed both for ordered as well for disordered systems, i.e. for alloys, for pure elemental materials and for ordered compounds. Similarly to the case of $T = 0$ K, one has to use the fully relativistic mode for the α calculations, with a large amount of k-points in the BZ (specified by `NKTAB` in the input file). The `NL` value **has to be taken** not less than `NL = 4` in the case of d-systems (that corresponds to the cutoff $l_{max} = 3$ for the angular momentum expansion of the Green's function), `NL = 5` in the case of f-systems (i.e. $l_{max} = 4$), etc.

The finite temperature calculations can be done using two different modes specified by the parameter `SETFLUCT`.

MODE 1 (specified by `SETFLUCT = MLIN` with linear temperature grid, used only in the lattice vibrations mode) requires the following parameters

section TASK

VAR / SWITCH	default	description
`Gilbert`	OFF	Perform calculation of the Gilbert damping parameter
`SETFLUCT`	OFF	Linear temperature grid (used only in the lattice vibrations mode)
`TMPMIN`	OFF	Lower limit of the temperature region
`TMPMAX`	OFF	Upper limit of the temperature region
`NTMP`	OFF	Number of temperature points used for $\alpha(T)$ calculations
`NVIBRA`	OFF	Number of directions for atomic displacements representing thermal lattice vibrations

NOTE that this mode allows to take into account only the electron scattering due to lattice vibrations! In the current version, always `NVIBRA`$= 14$ when `NVIBRA`> 1 in the input file.

Example

The input file DATASET.inp in this cas might look like:

```
CONTROL  DATASET = Fe
         ADSI    = GILBERT
         POTFIL  = Fe.pot_new
         PRINT = 0

SITES    NL={3}
# NL=3 is used only for demonstration

CPA      TOL=0.00000001

TAU      BZINT= POINTS  NKTAB=1000
# small NKTAB is used only for demonstration

ENERGY   GRID={3} NE={1}

TASK    Gilbert  SETFLUCT=MLIN TMPMIN=100 TMPMAX=100 NTMP=1
        VIBRA NVIBRA=14
```

The results are written to the output file DATASET.out with the last table looking like this:

```
     Gilbert damping parameter ALPHA

                  total             term 0         term 1
     (-VC)      0.0053354973      0.1134883890  -0.1081528918
     (+VC)      0.0052968091      0.1134883890  -0.1081915800

     to be scaled by factor g_eff / m_eff


magnetic moment information   with  GEFF = 2(SMT+OMT)/SMT

IQ   IO   IT  TXT     CONC    SMT     OMT     GEFF   G/MEFF
 1    1    1  Fe     1.000   2.280   0.054   2.047   0.877
average (site)               2.280   0.054   2.047   0.877
average (sum)                2.280   0.054   2.047   0.877


site and component resolved ALPHA (including factor g_eff / m_eff)

                  alfa (+VC)          alfa (-VC)          alfa (0)
IQ  TXT           xx        yy          xx       yy         xx       yy
 1  Fe          0.004647  0.004647    0.00468  0.00468    0.09956  0.09956
average (site)  0.004647  0.004647    0.00468  0.00468    0.09956  0.09956
average (sum)   0.004647  0.004647    0.00468  0.00468    0.09956  0.09956
```

MODE 2 specified by `SETFLUCT = M_T` requires the following parameters

section TASK

VAR / SWITCH	default	description
`Gilbert`	OFF	Perform calculation of the Gilbert damping parameter
`SETFLUCT`	OFF	Temperature grid decribed in an additional data file
`FLUCTFIL`	OFF	Data file which contains the information about temperature dependent magnetizstion, taken from the experiment or Monte Carlo simulations
`NVIBRA`	OFF	Number of directions for atomic displacements representing thermal lattice vibrations
`NFTET`	OFF	Number of grid points specifying θ angle
`NFPHI`	OFF	Number of grid points specifying ϕ angle

NOTE that this mode allows to take into account the electron scattering due to lattice vibrations as well as due to spin fluctuations. The grid for the spin fluctuations is characterized by the numbers `NFTET` and `NFPHI`. If `NFTET = 1` and `NFPHI = 1` but `NVIBRA` is bigger than 1, only lattice vibrations are taken into account. On the other hand if `NVIBRA = 1` but `NFTET` is bigger than 1 and `NFPHI` is bigger than 1, only spin fluctuations are taken into account.

Example

To calculate the Gilbert damping parameter α for $\mathrm{Fe}_{0.999}\mathrm{Vc}_{0.001}$ the input file DATASET.inp might look like:

```
CONTROL  DATASET = FeVac
         ADSI    = Gilbert
         POTFIL  = FeVac_Gilbert.pot
         PRINT = 1

CPA      TOL=0.00001

TAU      BZINT= POINTS  NKTAB= 300

ENERGY   GRID={3} NE={1}

TASK      Gilbert SETFLUCT=M_T  FLUCTFIL=FeVac_Gilbert.dat
          NVIBRA=14  NFTET=30 NFPHI=4
```

The file FeVac_Gilbert.dat looks like:

```
2                               ! number of TYPES
*******************************************************************************
TYPES
    1     Fe      1
    2    Vac      1
*******************************************************************************
2                               ! number of temperature steps
1023.1                          ! TCURIE
0.        1.0000     1.0000     ! magnetization at T=0 - only for nomalization
250.6     0.9865     0.9865     ! first temperature step
814.3     0.8024     0.8024     ! each column corresponds to different TYPE
501.1     0.9463     0.9463
```

NOTE: the magnetization at $T = 0$ is used only for normalization. Tis is important when the magnetization is given in absolute units.

The results are stored in the output file DATASET.out with the last table looking like this:

```
          Gilbert damping parameter ALPHA

                       total             term 0         term 1
          (-VC)      0.0019641976      0.4685707032  -0.4666065056
          (+VC)      0.0091442656      0.4685707032  -0.4594264376

          to be scaled by factor g_eff / m_eff


     magnetic moment information   with  GEFF = 2(SMT+OMT)/SMT

     IQ   IO   IT  TXT     CONC    SMT     OMT     GEFF   G/MEFF
      1    1    1  Fe     0.999   2.270   0.052   2.046   0.881
      1    2    2  Vc     0.001  -0.002  -0.004   5.443-890.531
     average (site)               2.268   0.052   2.046   0.882
     average (sum)                2.268   0.052   2.046   0.882
```

```
     site and component resolved ALPHA (including factor g_eff / m_eff)

                      alfa (+VC)            alfa (-VC)            alfa (0)
     IQ  TXT           xx        yy          xx       yy          xx       yy
      1  Fe          0.008065  0.008065    0.00173  0.00173    0.41327  0.41327
      1  Vc         -0.001862 -0.001862   -0.00044 -0.00044   -0.00012 -0.00012
     average (site)  0.008055  0.008055    0.00173  0.00173    0.41285  0.41285
     average (sum)   0.008065  0.008065    0.00173  0.00173    0.41327  0.41327
```

5.3 Resistivity

The electrical conductivity can be calculated using **kkrchi** (more precisely, the symmetric part of the conductivity tensor is calculated via the Kubo-Greenwood formalism). The conductivity calculation is performed for alloy systems at $T = 0K$.

The specific part of the input includes the parameters

```
SITES    NL={4}

ENERGY   GRID={3} NE={1}

TASK     SIGMA
```

One has to use a large amount of k-points in the BZ for SIGMA calculations, i.e. one has to investigate a convergence with respect to the number `NKTAB` in the input file.

Files used:

Filename	unit	I/O	description
DATASET.inp	5	I	input file described above
DATASET.pot	4	I	input potential read in by ⟨POTIO⟩.

The results are printed to the file DATASET.out:

```
     SIGMA
     -----
                  without Vertex                              with Vertex
MUE NUE        a.u.       [1/muOhm*cm]              a.u.         [1/muOhm*cm]
 1   1   111.72177796     0.07044670         146.56535831       0.09241748
 1   2     0.00000000     0.00000000           0.00000000       0.00000000
 1   3     0.00000000     0.00000000           0.00000000       0.00000000
 2   1     0.00000000     0.00000000           0.00000000       0.00000000
 2   2   111.72177796     0.07044670         146.56535831       0.09241748
 2   3     0.00000000     0.00000000           0.00000000       0.00000000
 3   1     0.00000000     0.00000000           0.00000000       0.00000000
 3   2     0.00000000     0.00000000           0.00000000       0.00000000
 3   3   108.35966863     0.06832670         140.25367663       0.08843762
```

where `MUE,NUE` represent the components of the conductivity tensor $\sigma_{\mu\nu}$. The conductivity values are given in a.u. and in SI units $(\mu\Omega^{-1}cm^{-1})$. Results obtained without vertex corrections and with vertex corrections are printed.

5.3.1 Example

To calculate the conductivity for $Fe_{0.5}Pd_{0.5}$ one has to use the input files created by **xband**. The input file DATASET.inp can look like:

```
CONTROL  DATASET     = Fe0.5Pd0.5
         ADSI        = SIGMA
         POTFIL      = Fe0.5Pd0.5.pot_new
         PRINT = 0

SITES    NL = { 4 }

TAU      BZINT= POINTS  NKTAB= 100000

ENERGY   GRID={3} NE={1}

TASK     SIGMA
```

The results are stored in the output file DATASET.out:

```
Sigmatensor total:
==================

     SIGMA
     -----
                  without Vertex                         with Vertex
MUE NUE       a.u.       [1/muOhm*cm]             a.u.        [1/muOhm*cm]
 1   1   111.72177796     0.07044670        146.56535831       0.09241748
 1   2     0.00000000     0.00000000          0.00000000       0.00000000
 1   3     0.00000000     0.00000000          0.00000000       0.00000000
 2   1     0.00000000     0.00000000          0.00000000       0.00000000
 2   2   111.72177796     0.07044670        146.56535831       0.09241748
 2   3     0.00000000     0.00000000          0.00000000       0.00000000
 3   1     0.00000000     0.00000000          0.00000000       0.00000000
 3   2     0.00000000     0.00000000          0.00000000       0.00000000
 3   3   108.35966863     0.06832670        140.25367663       0.08843762
SMA from inv. cond. (with VC)                    [%]:     4.433682

     Resistivity
     -----------
                  without Vertex                         with Vertex
MUE NUE       a.u.       [1/muOhm*cm]             a.u.        [1/muOhm*cm]
 1   1     0.00895081    14.19512986          0.00682289      10.82046375
 1   2     0.00000000     0.00000000          0.00000000       0.00000000
 1   3     0.00000000     0.00000000          0.00000000       0.00000000
 2   1     0.00000000     0.00000000          0.00000000       0.00000000
 2   2     0.00895081    14.19512986          0.00682289      10.82046375
 2   3     0.00000000     0.00000000          0.00000000       0.00000000
 3   1     0.00000000     0.00000000          0.00000000       0.00000000
```

```
 3   2     0.00000000     0.00000000          0.00000000       0.00000000
 3   3     0.00922853    14.63556659          0.00712994      11.30740516
SMA from resistivity (with VC)                      [%]:     4.433682

Results without Vertex-corrections:
===================================

Conductivity in a.u.:                  110.60107
Conductivity in [1/(Ohm*m)]         6974003.15023
Inverse Conductivity [muOhm.cm]         14.33897
Resistivity in [muOhm.cm]               14.34194 (no VC)

Results including Vertex-corrections:
=====================================

Conductivity in a.u.:                  144.46146
Conductivity in [1/(Ohm*m)]         9109086.04892
Inverse Conductivity [muOhm.cm]         10.97805
Resistivity in [muOhm.cm]               10.98278 (with VC)
Inclusion of vertex-correction increases conductivity by  30.6149%
```

5.3.2 Resistivity at finite temperature

The resistivity $\rho_{\alpha\beta}$ can be also calculated for $T \neq 0K$. Similarly to the Gilbert damping calculations, $\rho_{\alpha\beta}(T)$ calculations are allowed for ordered as well for disordered materials. In this case, in the input file the required parameters have to be specified, as it is described above in the section 5.2.2.

6 Calculating spectroscopic properties

6.1 Calculation of spectroscopic properties included in kkrgen

6.1.1 General remarks

Any expression for a spectroscopic property calculated by the **SPRKKR** programs involve the site-diagonal energy dependent scattering path operator $\tau^{nn}_{\Lambda\Lambda'}(E)$. When starting e.g. **kkrgen** with the input setting `TASK XAS` **kkrgen** first checks whether a corresponding file DATASET.tau containing $\tau^{nn}_{\Lambda\Lambda'}(E)$ is available. If this is not the case, **kkrgen** will first calculate the scattering path operator. Depending on the `TASK` it will store the component-projected one $\tau^{nn\alpha}_{\Lambda\Lambda'}(E)$ (e.g. for `TASK XAS`) or the site-specific one $\tau^{nn}_{\Lambda\Lambda'}(E)$ together with the effective scattering t-matrices $m^{n}_{\Lambda\Lambda'}(E) = (t^{n}(E))^{-1}_{\Lambda\Lambda'}$ (e.g. for `TASK VBXPS`). After the file DATASET.tau has been created, it will be read energy by energy and the spectroscopic property will be calculated. The file DATASET.tau will be kept for later use. For example one may calculate in a first run of **kkrgen** the K-edge spectrum of Fe in FePt. In a later run, that uses the file DATASET.tau created by the first run, one may calculate, over the same energy for the final states, the $L_{2,3}$-spectra of Pt in FePt.

For spectroscopic properties, that depend on the polarization of absorbed or emitted light, three polarization states are generally considered: left and right circularly and z-linearly polarized light. These polarization states are referred to a local frame of reference of an atom for which the magnetization points along the local z-axis. This means that locally always a polar geometry is considered. This has to be kept in mind when dealing with systems with their magnetization not aligned along the crystallographic z-axis.

6.1.2 X-ray absorption spectroscopy

A calculation of X-ray absorption spectra [8] can be done using the programs **kkrgen** or **embgen**. Absorption spectra are calculated for three polarizations ($+$,$-$ and z). The corresponding part of DATASET.inp supplies the following parameters:

section ENERGY

VAR / SWITCH	default	description
`GRID`=integer	6	special energy mesh for XAS calculations, that is denser in the white line regime
`NE`=integer	-	number of E-mesh points
`EMAX`=real	-	real part of maximal E-value
`IME`=real	-	imaginary part of energy

section TASK

VAR / SWITCH	default	description
`XAS`	OFF	Calculates X-ray absorption spectra
`IT`=integer	1	atom type `IT`
`CL`=string	2P	initial core level shell
`MECHECK`	OFF	check matrix elements for testing purposes
`OUTPUT`=string	MBARN	MBARN : output of absorption coefficient μ as $\mu_{atom} = \mu\, V_{uc}$ in [Mbarn] SIGMA : output as absorptive part of optical conductivity $\sigma = \mu\, c/4\pi$ in $[10^{15} s^{-1}]$

When **kkrgen** is used for the XANES-region, a sufficiently high number of $\vec{k}$-points should be set and the structure constants need to be converged (`RMAX` and `GMAX` values set high enough).

If **embgen** is used instead of **kkrgen** to get the multiple scattering path operator τ one is able to calculate X-ray absorption spectra for both XANES and EXAFS regions. In this case one has first of all to set the calculation of τ in real-space mode within a finite cluster. It is then necessary to set the cluster size and to specify around which atomic site of the unit cell should be the cluster built:

```
TAU       CLUSTER   NSHLCLU=4  IQCNTR=1
```

or

```
TAU       CLUSTER   CLURAD=1.5  IQCNTR=1
```

In the first example the size of the cluster is specified by the number of coordination shells `NSHLCLU` around the site `IQCNTR`, while the second gives explicitly the radius `CLURAD` (in lattice constant units) of a spherical cluster around `IQCNTR`.

It is important to keep in mind that the scattering path operator τ as obtained from **embgen** is meaningful only for the central cluster site `IQCNTR` and the corresponding DATASET.tau file created by **embgen** contains only one τ at the end of the execution. Therefore, the atomic type `IT` specified in the `TASK XAS` line should really occupy the atomic site `IQCNTR`. Choosing another `IT` might require a new `IQCNTR` and consequently the DATASET.tau file has to be either removed or renamed – in any case created anew.

The two different X-ray absorption regimes – XANES and EXAFS – can be essentially distinguished by different cross-sections of the photo-electron; therefore, using **embgen** for a XANES calculation requires rather big clusters (up to 200 atoms) whereas smaller (around 50 atoms) are usually enough for an EXAFS spectrum, corresponding to a reduced cross-section. In the latter case, however, due to the high energy of the final state, a larger value for the angular momentum expansion is needed, in general $l_{\max} = 6$. Nevertheless, the selection rule $\Delta l = \pm 1$ allows one to reduce the size of the output τ matrix using the variable `NLOUT`, e.g., for $M_{4,5}$-edges one needs no final states higher than f-states, $l = 3$, implying `NLOUT=4`. The input file may look than like

```
SITES   NL=7 (lmax+1)
TAU       CLUSTER   NSHLCLU=4  IQCNTR=1  NLOUT=4
```

Regardless which program is used to calculate X-ray absorption spectra, **kkrgen** or **embgen**, the final result will be the spectroscopy file DATASET.rat which can be plotted using **plot**.

Files used:

Filename	unit	I/O	description
DATASET.inp	5	I	input file
DATASET.pot	4	I	input potential read in by ⟨POTFIT⟩.
DATASET.rat	7	O	XAS spectroscopy file created by ⟨XRAYSPEC⟩.

Example

Assuming we have already an SCF potential for bcc-Fe stored in the file Fe.pot, we want to calculate the K-edge absorption spectrum of Fe using **kkrgen**. The input file should look like

```
################################################################################
#  SPR-KKR input file    Fe_XAS.inp
#  created by xband on Mon May  2 15:00:44 CEST 2005
################################################################################

CONTROL  DATASET = Fe
         ADSI    = XAS
         POTFIL  = Fe.pot
         PRINT = 0

TAU      BZINT= POINTS  NKTAB= 250

ENERGY   GRID={6}  NE={180}
         EMAX=4.0   ImE=0.01 Ry

TASK     XAS      IT=1  (Fe)  CL=1s
```

At the end of the execution, the file Fe_XAS_Fe_K.rat will be created, which has to be processed by **plot**.

Alternatively, one can use **embgen** with the only difference in the `TAU` line

```
TAU      CLUSTER   IQCNTR=1   CLURAD=1.5
```

A comparison of the two calculations is presented in Fig. 6.1. It is obvious that the cluster size chosen here for the real-space calculation is still too small.

The corresponding EXAFS spectrum at the K-edge of Fe can be calculated using **embgen** with the input file accordingly modified for the energy range (see Fig. 6.2)

```
################################################################################
#  SPR-KKR input file    Fe_XAS.inp
#  created by xband on Mon May  2 15:00:44 CEST 2005
################################################################################

CONTROL  DATASET = Fe
         ADSI    = XAS
         POTFIL  = Fe.pot
         PRINT = 0

TAU      CLUSTER   IQCNTR=1   CLURAD=2.0

ENERGY   GRID={6}  NE={180}
         EMAX=30.0   ImE=0.01 Ry

TASK     XAS      IT=1  (Fe)  CL=1s
```

Note the value for `EMAX` and the fact that, since we are not interested in the near-edge region, `GRID=3` can also be used.

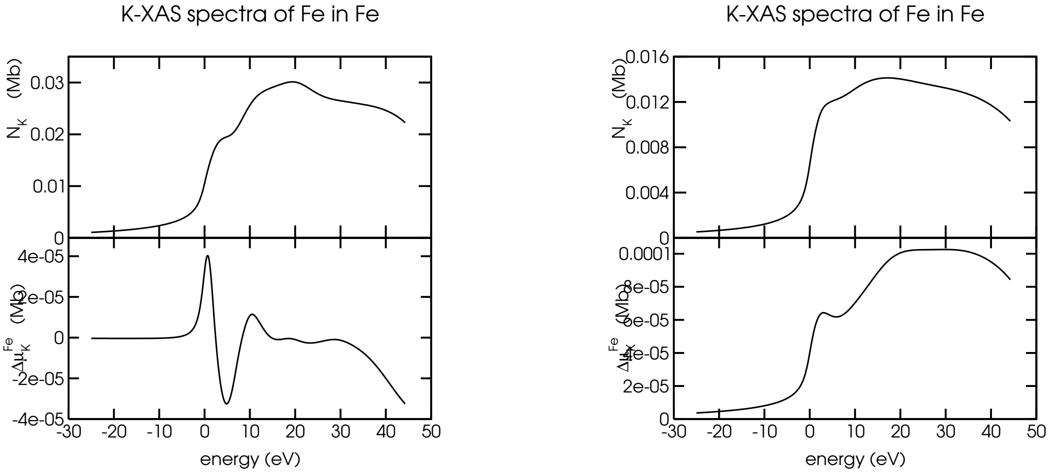


Figure 6.1: X-ray absorption spectrum at the K-edge of Fe in bcc-Fe. Left: results obtained using a Brillouin-zone integration for τ (program **kkrgen**). Right: results obtained by means of a real-space calculation for τ (program **embgen**) with a cluster size of $1.5\,a_{\mathrm{Fe}}$. Top panel of each figure shows the polarization averaged absorption coefficient $\bar{\mu}$, the dichroic signal $\mu^+ - \mu^-$ is shown at the bottom. **Note**: the broadening may lead to artefacts at the high energy end of the spectrum. The cluster calculation is not converged !

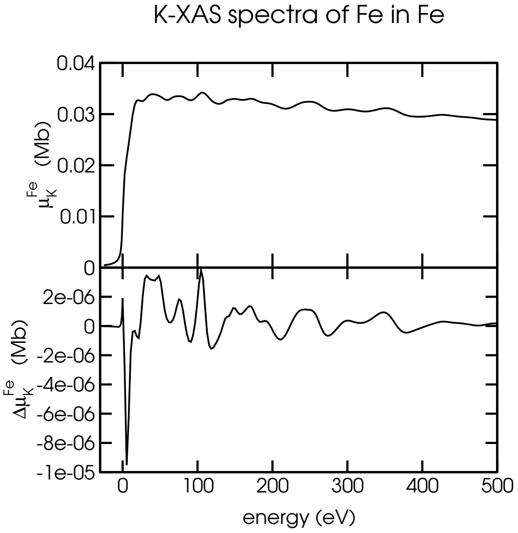


Figure 6.2: EXAFS and MEXAFS spectra at the K-edge of Fe in bcc-Fe. Top: polarization averaged absorption coefficient $\bar{\mu}$; bottom: the dichroic signal $\mu^+ - \mu^-$.

6.1.3 X-ray emission spectroscopy

For a calculation of X-ray electron emission spectra [21] the corresponding part of DATASET.inp may look like this:

```
ENERGY GRID={6} NE=150  EMAX=4.0 IME=0.01

TASK XES CL=2p IT=1
```

For further details see Sec. 6.1.2.

6.1.4 X-ray magneto-optics

X-ray resonant magneto-optical properties are calculated using the programs **kkrgen** or **embgen** respectively (see Section 6.1.1). The resulting full complex conductivity tensor $\sigma_{\alpha\beta}$ [22] is written into the file `DATASET.rat`. The corresponding part of DATASET.inp supplies the following parameters:

section TASK

VAR / SWITCH	default	description
`XMO`	OFF	Calculates X-ray resonant magneto-optics
`IT`=integer	1	index number of atom type
`CL`=string	2p	Initial core level shell
`NE3`=integer	40	
`EMAX3`=real	8.0	
`TSELECT`=integer	0	use exclusively the resonant (`TSELECT`=1) or non-resonant (`TSELECT`=2) contribution to $\boldsymbol{\sigma}$ or both (`TSELECT`=0)
`TAUCORE`=real	tabulated	core level life time parameter τ in [eV] if $l_{core} > 0$ specify 2 values for: $j_{core} = l_{core} - 1/2$ and $j_{core} = l_{core} + 1/2$
`MECHECK`	OFF	check matrix elements

section ENERGY

VAR / SWITCH	default	description

`GRID`=integer	7	special Gaussian energy mesh for XMO calculations
`NE`=integer	-	number of E-mesh points
`EMAX`=real	-	real part of maximal E-value
`IME`=real	-	imaginary part of energy

The created spectroscopy file DATASET.rat can be plotted using **plot**. Some X-ray magneto-optical spectra (i.e. Faraday rotation, Kerr rotation etc.), will be printed as a **xmgrace** file. Any other calculations based on conductivity tensor $\sigma_{\alpha\beta}$ could be implemented into the **plot** program. Technical problems and hints of calculation of the XMO are the same as in the section 6.1.2.

Files used:

Filename	unit	I/O	description
DATASET.inp	5	I	input file
DATASET.pot	4	I	input potential read in by ⟨POTFIT⟩.
DATASET.rat	7	O	file written by ⟨XRAYSPEC⟩ includes full conductivity tensor $\boldsymbol{\sigma}$

Example

For calculations of X-ray magneto-optical effects of Fe the input file created by xband should look like the following one.

```
################################################################################
#  SPR-KKR input file    Fe_XMO.inp
#  created by Xband on Fri Jan 25 16:40:31 CET 2002
################################################################################

CONTROL  DATASET = Fe
         ADSI    = XMO
         POTFIL  = Fe.pot_new
         PRINT = 0

STRCONST ETA= 0.35  RMAX=2.9  GMAX=3.3

TAU      BZINT= POINTS  NKTAB= 250

ENERGY   GRID={7}  NE={180}
         EMAX=4.0   ImE=0.01

TASK     XMO     IT=1  CL=2p
```

Running **kkrgen** the X-ray magneto-optical spectra will be written to the file DATASET.rat. The **plot** program via **xband** applied to DATASET.rat creates the following **xmgrace** files: xas_Fe_L23_eps_dia.agr, xas_Fe_L23_eps_off.agr, xas_Fe_L23_kerr.agr, xas_Fe_L23_n_dic.agr, xas_Fe_L23_sig_dia.agr, xas_Fe_L23_sig_dia.agr, xas_Fe_L23_refl.agr that can be viewed using **xmgrace**. As examples the diagonal part of conductivity tensor (file xas_Fe_L23_sig_dia.agr) and Kerr rotation (file xas_Fe_L23_kerr.agr) plotted by the **plot** program are presented below.

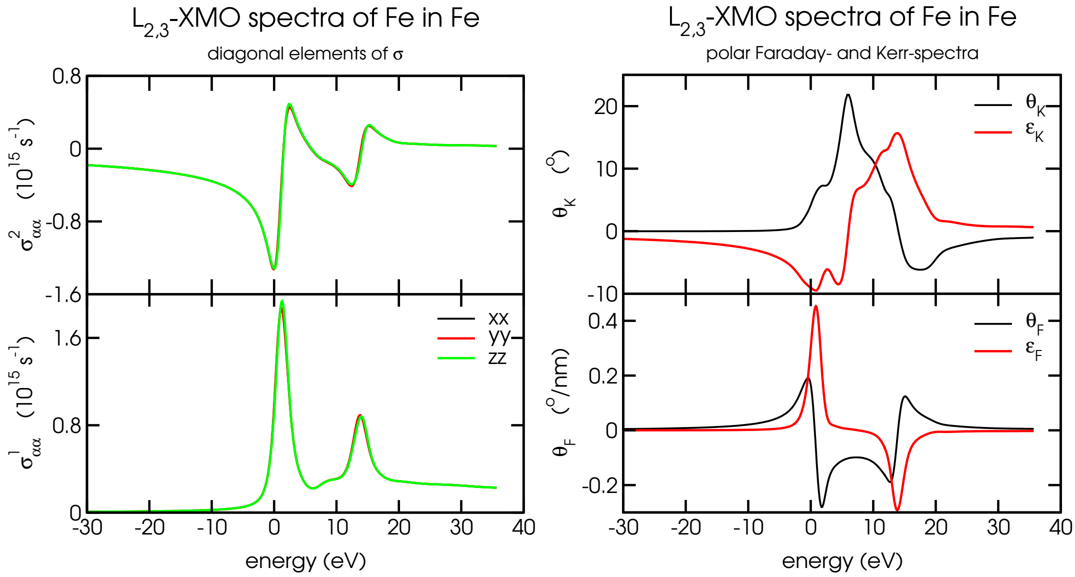


Figure 6.3: Left: the lower and upper panel shows the real and imaginary parts of the diagonal optical conductivity σ_{xx} at the L$_{2,3}$-edge of bcc-Fe. Right: the lower and upper panels show the resulting polar Faraday- and Kerr-rotation (θ) and ellipticity (ϵ) spectra. A Lorenzian broadening with 0.3 Ry for the core levels and a Gaussian broadening with 0.4 Ry were applied.

6.1.5 Non-relativistic Appearance Potential Spectroscopy

The calculation of Spin-Resolved Appearance Potential Spectra (SR-APS) can be performed using **kkrgen**, and requires two separate runs of the program. The procedure is based on the following expression for the SR-APS transition probability [23]:

$$P_\sigma(E_2) = \int_{E_a}^{E_b} dE \sum_{l,l',\sigma'} n_{l\sigma}(E) n_{l'\sigma'}(E') W_{l\sigma,l'\sigma'}(E,E') \quad , \tag{6.1}$$

with $E' = E_2 - E + E_c$. In this equation $n_{l\sigma}(E)$ is the angular momentum- and spin-resolved density of states (DOS) and $W_{l\sigma,l'\sigma'}(E,E')$ an effective cross-section. The range of energy for the folding of the DOS curves is $E_a = E_F$ and $E_b = E_2 + E_c - E_F$, E_c and E_2 being the energies of the initial core and LEED states, respectively.

In the first run of **kkrgen**, one has to create and store the DOS data, as it is described in Sec. 3.7, using the Fermi energy as the lowest energy value, while the highest energy and the number of energy points are chosen by the user. Your DATASET.inp might look like

```
ENERGY GRID={3} NE=50 EMIN=0.7169 EMAX=2.0 IME=0.01

TASK DOS
```

The second run of **kkrgen** will use the generated DATASET.dos file so you should not change any of the parameters in the `ENERGY` token, but only to specify the new task as

```
TASK APS CL=2p IT=1 GAMMA=0.1 OUTGNU CALCLS
```

The main output is the file DATASET.aps that has to be supplied to the **plot** program. The **plot** program allows you to account for apparative and initial state finite life-time broadening and generates **xmgrace** files.

Final state life-time broadening is accounted for in **kkrgen**, by setting a positive value for `GAMMA`. The DOS curves are broadened with a linearly energy-dependent Lorentzian $W_L = \gamma(E_2 - E_F)$ prior to the self-convolution. The broadened DOS is written out in a format suitable for the **plot** program.

section TASK

VAR / SWITCH	default	description
`APS`	OFF	calculates the Spin-Resolved Appearance Potential Spectra
`CL`=string	2P	initial core level
`IT`=integer	1	index of atom type for which the spectra should be calculated
`GAMMA`=real	-0.1	parameter for the linear energy-dependent broadening of DOS. If smaller equal zero, no broadening is done
`OUTGNU`	OFF	Generates auxiliary files containing the cross-sections $W_{l\sigma,l'\sigma'}(E,E')$ for a 20×20 energy grid
`CALCLS`	OFF	It calculates each angular momentum- and spin-projected APS and stores these spectra in separate files

Files used:

Filename	unit	I/O	description
DATASET.inp	5	I	input file
DATASET.pot	4	I	input potential read in by ⟨POTFIT⟩.
DATASET.dos	10	O/I	DOS file
DATASET.aps	7	O	calculated SR-APS spectra written by ⟨NRAPS⟩
DATASET.dos.brd	12	O	broadened-DOS file
DL$_1$L$_2$S$_1$S$_2$	51	O	If `OUTGNU` has been set, the direct term (D),
EL$_1$L$_2$S$_1$S$_2$	52	O	the exchange term (E),
DEL$_1$L$_2$S$_1$S$_2$	53	O	the cross-term (D^*E) and
WL$_1$L$_2$S$_1$S$_2$	54	O	the total cross-section (W) are calculated and stored in a format suitable for the **gnuplot** program. The notation is L$_{1(2)}$ = s, p, d,…and S$_{1(2)}$ = U/D for spin Up or Down; e.g. EsdUD.
DATASET.aps.L$_1$L$_2$	7	O	If `CALCLS` has been set, the angular momentum- and spin-projected APS are stored in these files, with L$_{1(2)}$ = s, p, d, …They have to be supplied to the **plot** program and treated in the same way as the standard `aps`-file

Example

To calculate the spin-resolved apperance potential spectra for Ni the input file created by **xband** should look like this:

```
################################################################################
#  SPR-KKR input file    Ni_APS.inp
#  created by Xband on Wed Feb 13 12:19:41 CET 2002
################################################################################

CONTROL  DATASET = Ni
         ADSI    = APS
         POTFIL  = Ni.pot
         PRINT = 0

STRCONST ETA=0.30   RMAX=3.30   GMAX=3.30

TAU      BZINT= POINTS  NKTAB= 250

ENERGY   GRID={3}  NE={50}
```

```
          EMIN=0.7169   EMAX=2.0   ImE=0.01

TASK      APS CL=2p IT=1 GAMMA=0.0
```

The file Ni.pot has to contain the (usually converged) potential created by **kkrscf**. Running **kkrgen** the APS spectra will be written to file Ni_APS.Ni.L23.aps, which one converts than by using **plot** from within **xband** to Ni_APS.Ni.L23.agr, which then can be visualised by using **xmgrace** (See Fig. 6.4).

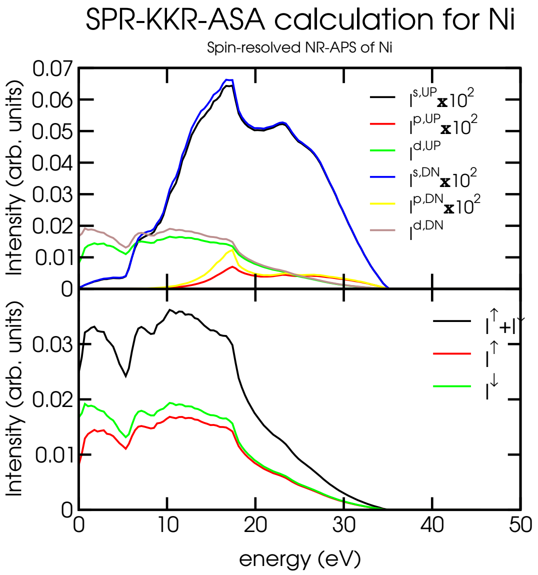


Figure 6.4: The lower and upper panel shows the spin-orbital- and spin-resolved non-relativistic APS spectra for Ni.

6.1.6 Auger Electron Spectroscopy

Non-relativistic Auger Electron Spectroscopy

Because APS is essentially an inverse Auger experiment, the calculation of non-relativistic Spin-Resolved Auger Electron Spectra (SR-AES) is done in a similar way as described above for the SR-APS [23, 24]. Note that in the case of SR-AES, the self-convolution is done in the occupied part of the valence band, that is, in creating the DOS file, the Fermi energy is now the highest energy value. The `DATASET.inp` can have for the first step the form

```
ENERGY GRID={3} NE=50 EMIN=0.2 EMAX=0.7447 IME=0.01

TASK DOS
```

and, for the second run of **kkrgen**, to obtain the SR-AES written out to DATASET.aes,

```
TASK NRAES CL=2p IT=1
```

Note that there is no DOS-broadening in the case of SR-AES. For further details see Sec. 6.1.5.

Example

For calculations of the non-relativistic SR-AES of Fe the input file created by **xband** should look like the following one.

```
###############################################################################
#  SPR-KKR input file    Fe_NRAES.inp
#  created by xband on Thu Apr 28 12:26:56 CEST 2005
###############################################################################

CONTROL  DATASET = Fe
         ADSI    = NRAES
         POTFIL  = Fe.pot_new
         PRINT = 0

TAU      BZINT= POINTS  NKTAB= 250

ENERGY   GRID={3}  NE={50}
         EMIN=0.2  EMAX=0.7458  ImE=0.01 Ry

# NOTE:  run the program first to create the DOS
#        using this input file as it is i.e. with    TASK  DOS
#        for the second run supress the line with    TASK  DOS (insert #)
#        and activate the line with TASK  NRAES (remove #)

TASK     DOS

#TASK     NRAES     IT=1  (Fe)  CL=2p
```

In the first step one has to calculate the DOS from the bottom of the valence band up to the fermi level. Running **kkrgen** the DOS will be written to the file DATASET.dos. In the second step one has to comment the line `TASK DOS` and uncomment the line `TASK NRAES`. Then running **kkrgen** the non-relativistic SR-AES will be written to the file DATASET.aes. The **plot** program via **xband** applied to DATASET.aes creates the following **xmgrace** file DATASET.aes.agr. As example the broadened SR-AES spectra of bcc-Fe is shown in Fig. 6.5.

To represent intrinsic lifetime effects Lorenzian broadening has been applied with an energy-dependent broadening parameter W=A+B*(EF-E)^2 where A=0.5 eV and B=0.3 eV. The finite apparative resolution has been represented by Gaussian broadening with the parameter 0.4 eV for the full width at half maximum.

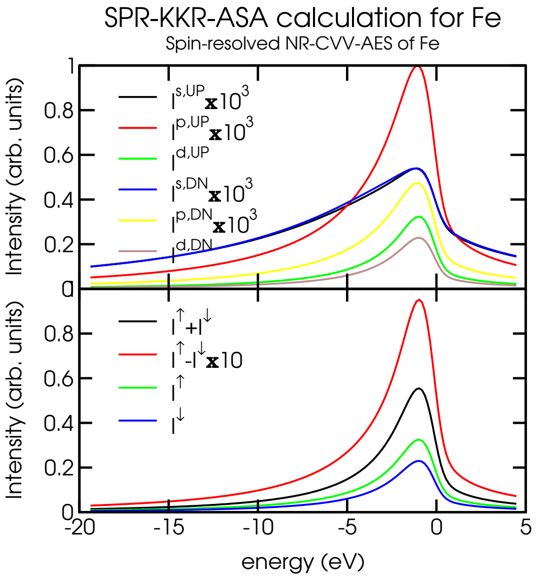


Figure 6.5: In the upper panel one can see the angular momentum contributions per spin for bcc-Fe. Spin-resolved spectra are shown in the down panel.

Relativistic Auger Electron Spectroscopy

Using **kkrgen** allows to calculate spin resolved core-valence-valence (CVV-AES) auger electron spectra [24]. The present implementation deals only with angle integrated spectra with the single-site scattering approximation used for the final time-reversed-LEED states. The specific part of the input file DATASET.inp may look like this

```
ENERGY GRID={3} NE=50 EMIN=-0.1 IME=0.01

TASK AES CL=2p IT=1
```

supplying the following parameters:

section TASK

VAR / SWITCH	default	description
`AES`	OFF	Calculates spin resolved fully relativistic AES spectra
`IT`=integer	1	atom type
`CL`=string	2P	initial core level shell

`PRINTME`	OFF	writing coulomb matrix elements in the ME* files with * = 1, ...,`NCST` and `NCST` the number of core states involved.
`NEME`	10	Number of energies for which the matrix elements will be calculated

Calculated spectra are written to the file DATASET.aes. Use **plot** program to obtain spectra in a **xmgrace** file.

Files used:

Filename	unit	I/O	description
DATASET.inp	5	I	input file
DATASET.pot	4	I	input potential read in by ⟨POTFIT⟩.
DATASET.aes	7	O	calculated AES spectra written by ⟨AES⟩
ME*	24	O/I	coulomb matrix elements (* = 1, ...,`NCST`)
SCRATCH	99	I/O	τ matrix for valence states

Example

For calculations of the fully relativistic CVV-AES of Fe the input file created by **xband** should look like the following one.

```
###############################################################################
#  SPR-KKR input file    Fe_AES.inp
#  created by xband on Wed Apr 27 15:33:29 CEST 2005
###############################################################################

CONTROL  DATASET = Fe
         ADSI    = AES
         POTFIL  = Fe.pot_new
         PRINT = 0

TAU      BZINT= POINTS  NKTAB= 1000

ENERGY   GRID={3}  NE={50}
         EMIN=-0.2   ImE=0.01 Ry

TASK     AES      IT=1  (Fe)  CL=2p    NEME=30
```

Then running **kkrgen** the fully relativistic CVV-AES will be written to the file DATASET.aes. The **plot** program via **xband** applied to DATASET.aes creates the following **xmgrace** file DATASET.aes.agr. As examples the broadened SR-AES spectra of bcc-Fe is shown in Fig. 6.6.

To represent intrinsic lifetime effects Lorenzian broadening has been applied with an energy-dependent broadening parameter W=A+B*(EF-E)^2 where A=0.1 eV and B=0.01 eV. The finite apparative resolution has been represented by Gaussian broadening with the parameter 0.4 eV for the full width at half maximum.

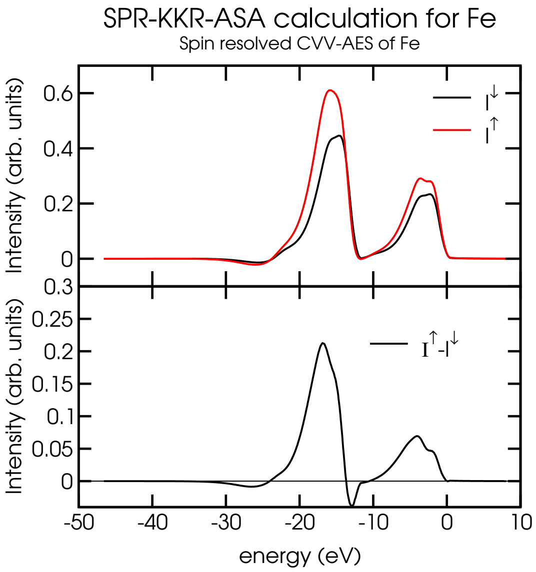


Figure 6.6: In the upper panel one can see the spin-resolved spectra for bcc-Fe and in the down panel the difference between the spin-up and spin-down intensities is plotted.

NOTE

One should use at least 50 energy points `(NE=50)` and `NEME` should be set at least 30 to get reasonable results.

6.1.7 X-ray Photoemission spectroscopy

Valence Band X-ray Photoemission Spectroscopy

kkrgen or **embgen** allows to calculate spin resolved valence band photoemission spectra in the angle integrated mode [25]. Usually the spectra are calculated for three different polarisations (right, left circularly polarised light and z-polarisation) within the single-site approximation for the final states. The input file should supply the following parameters:

section TASK

VAR / SWITCH	default	description
`VBPES`	OFF	Calculate angle integrated spin resolved VB-XPS spectra
`EPHOT`=real	1253.6	photon energy in eV 1: 1253.6 eV Mg-K$_{\alpha_{1,2}}$ 2: 1486.6 eV Al-K$_{\alpha_{1,2}}$
`USETAUNN`	OFF	single site approximation for final states suppressed

The resulting spectroscopy file DATASET.xps can be further processed using **plot**.

Files used:

Filename	unit	I/O	description
DATASET.inp	5	I	input file
DATASET.pot	4	I	input potential read in by ⟨POTFIT⟩.
DATASET.pes	7	O	calculated VBXPS spectra written by ⟨VBXPS⟩.
DATASET.tau.nnf.`EPHOT`	20	I/O	τ file for final states

Example

To calculate the angular-intergated VB-XPS for the fcc-Ni the input file created by **xband** should look like this:

```
###############################################################################
#  SPR-KKR input file    Ni_VBXPS.inp
#  created by xband on Mon May 23 16:14:14 CEST 2005
###############################################################################

CONTROL  DATASET = Ni
         ADSI    = VBXPS
         POTFIL  = Ni.pot
         PRINT = 0
```

```
TAU       BZINT= POINTS  NKTAB= 250

ENERGY    GRID={3}  NE={200}
          EMIN=-0.2   ImE=0.01 Ry

TASK      VBPES    EPHOT=600
```

The file Ni.pot has to contain the (usually converged) potential created by **kkrscf**. Running **kkrgen** the VB-XPS spectra will be written to files Ni_VBXPS.xps, which then can be visualized using **plot** from within **xband**. As a result of the **plot** program **xmgrace** files will be created:

- Ni_VBXPS_vb-xps.agr
- Ni_VBXPS_vb-xps_spin.agr

First file contains polarization averaged total and component resolved spectra. Second file contains spin- and polarization resolved VB-XPS spectra. The meaning of the legend is following:

- $POL(+) = I_+^\uparrow - I_+^\downarrow$
- $TOT(+) = I_+^\uparrow + I_+^\downarrow$
- etc. for all other polarizations

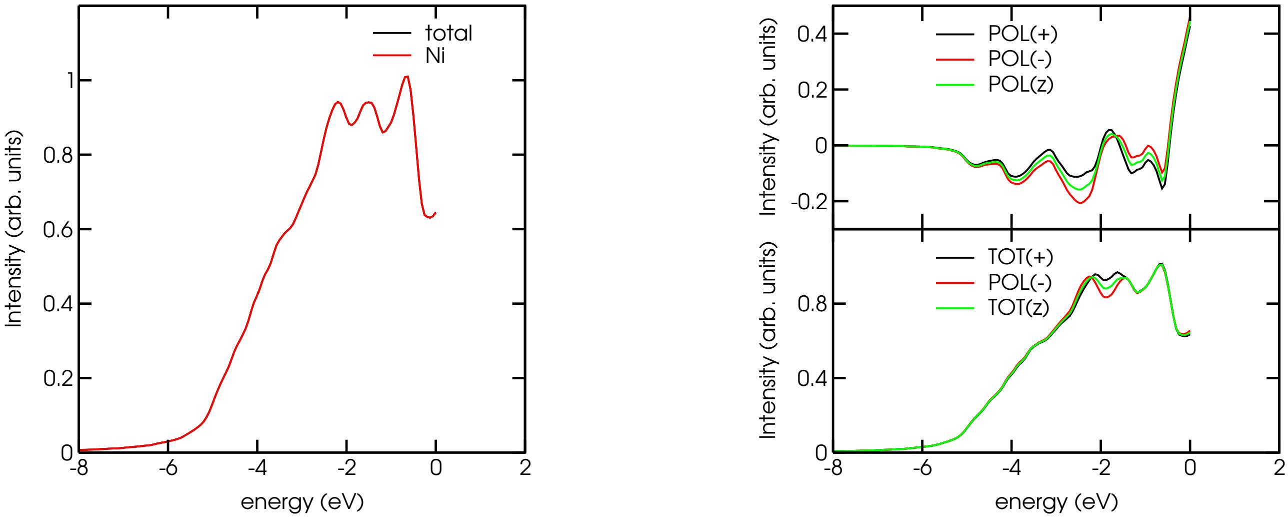


Figure 6.7: Left: spin and angle integrated VB-XPS spectrum of fcc-Ni for a photon energy of 600 eV. Right: Corresponding spin- and polarization-resolved spectrum.

Core Level X-ray Photoemission spectroscopy

kkrgen and **embgen** allow to calculate spin-resolved core level photoemission spectra [26]. The programs allow one to calculate CL-XPS in the angle integrated mode. The typical part of the input file DATASET.inp looks like this:

```
TASK CLXPS CL=2P EPHOT=1100 IME=0.03 USETSS
```

supplying the following parameters: **section** TASK

VAR / SWITCH	default	description
`CLXPS`	OFF	Calculate the core level X-ray photoemission spectra
`IT`=integer	1	atom type to calculate
`CL`=string	2P	initial core level shell
`EPHOT`=real	1253.6	photon energy in eV
`USETSS`	OFF	single site approximation for final states

The resulting spectra are written to the data-file `DATASET.xps`. Use **plot** for further processing and creating a corresponding **xmgrace** file.

Files used:

Filename	unit	I/O	description
DATASET.inp	5	I	input file
DATASET.pot	4	I	input potential read in by ⟨POTFIT⟩.
DATASET.pes	7	O	calculated CL-XPS spectra written by ⟨CLXPS⟩

Example

To calculate the angular-intergated CL-XPS for the fcc-Ni at the $L_{2,3}$-edge the input file created by **xband** should look like this:

```
###############################################################################
#  SPR-KKR input file    Ni_VBXPS.inp
#  created by xband on Mon May 23 16:14:14 CEST 2005
###############################################################################

CONTROL  DATASET = Ni
         ADSI    = VBXPS
         POTFIL  = Ni.pot
         PRINT = 0

TAU      BZINT= POINTS  NKTAB= 250

ENERGY   GRID={3}  NE={200}
         EMIN=-0.2   ImE=0.01 Ry

TASK     VBPES    EPHOT=600
```

The file Ni.pot has to contain the (usually converged) potential created by **kkrscf**. Running **kkrgen** the CL-XPS spectra will be written to files Ni_CLXPSNi.L23.xps, which then can be visualized using **plot** from within **xband**. As a result of the **plot** program **xmgrace** file Ni_CLXPSNi.L23.xps_Ni.igeo_1.agr will be created.

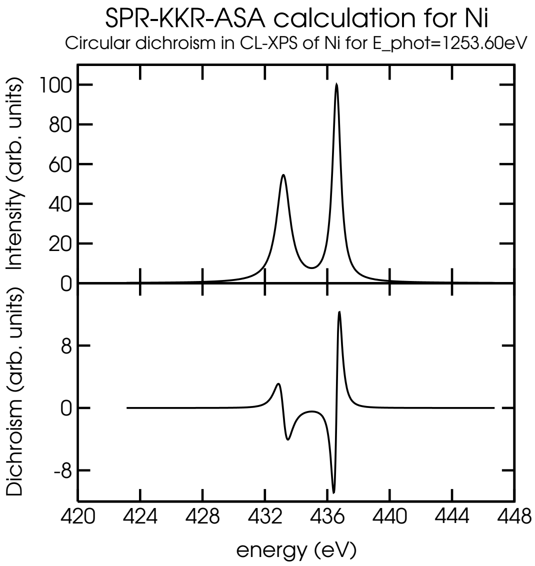


Figure 6.8: Top: angle integrated $L_{2,3}$ CL-XPS spectrum of fcc-Ni for a photon energy of 1253.6 eV. Bottom: the corresponding magnetic circular dichroism spectra.

6.1.8 Magnetic Compton profile

The Magnetic Compton Profile $J_{mag}(P_z)$, which is the one-dimensional projection of spin-resolved momentum density, is calculated using **kkrgen** with appropriate input and potential files. The specific part of the input file DATASET.inp supplies the following parameters

section TASK

VAR / SWITCH	default	description
`COMPTON`	OFF	calculate the Magnetic Compton Profile
`PNVEC`	{0,0,1}	scattering direction vector $\hat{p}_z$. This vector has to be parallel to the magnetization
`NPN`	50	number of grid-points along $\hat{p}_z$ direction
`NPP`	50	number of grid-points along $\hat{p}_x$ and $\hat{p}_y$ direction perpendicular to $\hat{p}_z$
`CP2D`	OFF	calculate the 2D projection of the $n^{\uparrow} - n^{\downarrow}$ perpendicular to `PNVEC`.

section ENERGY

VAR / SWITCH	default	description
`GRID`=integer	{5,3}	5: circular energy path in complex plane 3: linear energy path parallel to real axis
`NE`=integer	{30,50}	number of E-mesh points. Note: these values of E-mesh points ensure the convergency of the spectra.
`EMIN`=real	-	real part of lowest E-value
`splitss`	ON	set-up 2 paths for energy integration dealing separately with the backscattering and single site part of the Green's function. This avoids calculation of the matrix elements for the irregular wave function.

Example

For calculations of magnetic Compton profile of Fe the input file created by xband should look like the following one.

```
########################################################################
#  SPR-KKR input file    Fe_COMPTON.inp
#  created by xband on Wed Apr 27 17:44:57 CEST 2005
########################################################################

CONTROL  DATASET = Fe
         ADSI    = COMPTON
         POTFIL  = Fe.pot
         PRINT = 0

STRCONST  ETA=0.30 RMAX=3.30 GMAX=3.30

TAU      BZINT= POINTS  NKTAB= 250

ENERGY   GRID={5,3}  NE={30,50}
         EMIN=-0.2 Ry

TASK     COMPTON    PNVEC ={0.0,0.0,1.0}
         PNMAX=5.0   NPN=50    PPMAX=10.0   NPP=50
```

kkrgen writes $J_{up}(P_z)$, $J_{down}(P_z)$, $J_{sum}(P_z)$ and $J_{mag}(P_z)$ to files (see below). Use **plot** to account for experimental and intrinsic broadening and to reformat the spectra to be viewed via **xmgrace**.As an example, the magnetic Compton profile for Fe along [001] direction is shown in Fig. 6.9.

Files used:

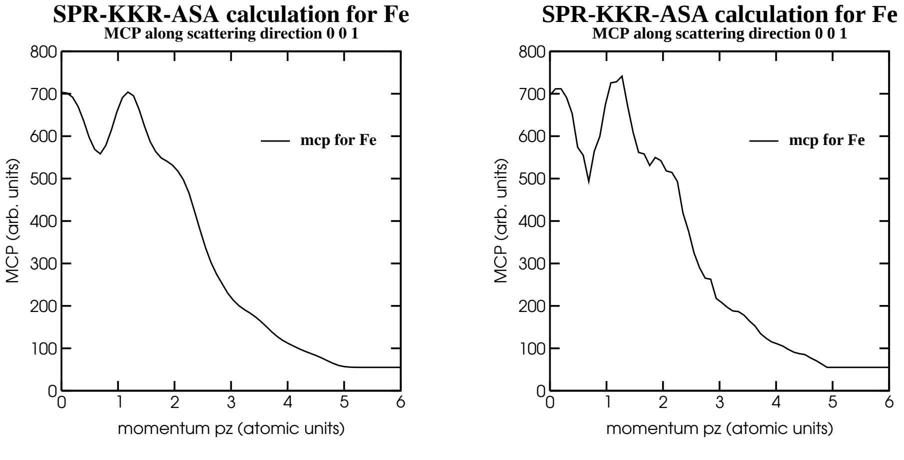


Figure 6.9: Magnetic Compton profile for Fe along [001] direction. The MCP spectra convoluted with a Gaussian of FWHM 0.3 a.u (left) is compared with the corresponding unbroadened MCP (right).

Filename	unit	I/O	description
DATASET.inp	5	I	input file described above
DATASET.pot	4	I	input potential read in by ⟨POTFIT⟩.
DATASET.mcp	50	O	Compton profiles J_{up}, J_{down}, J_{sum} and J_{mag} used by **plot**
DATASET_data.mcp	11	O	auxiliary file containing Compton profiles as datasets
DATASET_mcp2d.dat	21	O	The file containing 2D Compton profiles written if `CP2D` is active

NOTE: The calculations of the angle integrated VBXPS presented in this section are more relyable for the photon energy range from ≈ 200 eV to ≈ 1500 eV. For a more general implementation of the angle resolved and angle integrated photoemission see section 6.

6.1.9 Positron annihilation

The quantity measured in 2D-ACAR experiment is usually identified with the 2D projection of the electron-positron momentum density $n^{2\gamma}(p_x, p_y)$ in a direction perpendicular to the photon scattering direction. In order to calculate the $n^{2\gamma}(p_x, p_y)$ one has to follow the next steps:

Calculation of the positron energy

Run **kkrgen** in order to get the dispersion relation for the positron. Positrons are supposed to have the energy at the bottom of the band. The specific part of the input file DATASET.inp may look like this:

```
MODE    POSITRON
TASK    EKREL EMIN=-0.2 EMAX=1.0 NE=200 NK=300 KPATH=5
```

The description of the sections TASK and ENERGY are given in section 3.5

Example

For calculations of the positron energy for V the input file created by xband should look like the following one.

```
###############################################################################
#  SPR-KKR input file    V_DOS.inp
#  created by xband on Thu Sep 18 13:22:16 CEST 2003
###############################################################################

CONTROL  DATASET = V
         ADSI    = EKREL
         POTFIL  = V.pot
         PRINT = 0

TAU      BZINT= POINTS  NKTAB= 250

STRCONST ETA=0.30   RMAX=3.30   GMAX=3.30

MODE     POSITRON

TASK     EKREL   EMIN=-0.1  EMAX=1.0  NE=200
         NK = 300    KPATH = 5
```

Files used:

Filename	unit	I/O	description
DATASET.inp	5	I	input file described above
DATASET.pot	4	I	input potential read in by ⟨POTFIT⟩.
DATASET.bnd	10	O	dispersion relation $E(\vec{k})$ together with the information on the path in $\vec{k}$-space. Use **plot** to obtain the corresponding **xmgrace** file.

Preparation of the wave-function and scattering path operator τ for the positron

Run **kkrgen** with the following specific part of the input file DATASET.inp:

```
MODE    POSITRON
ENERGY GRID={3} NE={1} EMIN=xxx EMAX=xxx IME=0.01
TASK   DOS POSANIPREP
```

where xxx is the energy of the positron determined in the previous step.

Example

For calculation of the positron wave function and τ matrix for V the input file created by xband should look like the following one.

```
###############################################################################
#  SPR-KKR input file    V_DOS.inp
#  created by xband on Thu Sep 18 13:22:16 CEST 2003
###############################################################################

CONTROL  DATASET = V
         ADSI    = COMPTON
         POTFIL  = V.pot
         PRINT = 0

TAU      BZINT= POINTS  NKTAB= 250

STRCONST ETA=0.30   RMAX=3.30   GMAX=3.30

MODE     POSITRON

ENERGY   GRID={3}  NE={ 1  }
         EMIN= 0.4279  EMAX=0.4279   ImE=0.01

TASK      DOS POSANIPREP
```

Files used:

Filename	unit	I/O	description
DATASET.inp	5	I	input file described above
DATASET.pot	4	I	input potential read in by ⟨POTFIT⟩.
DATASET.pan	80	O	τ matrix and wave function of the positron are written in this file.

Calculation of the 2D-projection of the electron-positron momentum density

Run **kkrgen** with the following specific part of the input file DATASET.inp:

```
ENERGY GRID={5,3} NE={30,50} EMIN=-0.2
TASK   COMPTON POSANI NPN=50 NPP=50 MD2D SHFT PPMAX=5.0
```

section TASK

VAR / SWITCH	default	description

`COMPTON`	OFF	calculate the Magnetic Compton Profile
`POSANI`	OFF	calculate the electron-positron density of momentum
`MD2D`	1D	calculate the 2D projection of the electron-positron density of momentum
`SHFT`	OFF	$n^{2\gamma}$ (p_x, p_y) is represented with the point $(0, 0)$ in the middle of the p_x, p_y plane
`PNVEC`	{0,0,1}	photons scattering direction vector $\hat{p}_z$.
`NPN`	50	number of grid-points along $\hat{p}_z$ direction
`NPP`	50	number of grid-points along $\hat{p}_x$ and $\hat{p}_y$ direction perpendicular to $\hat{p}_z$ $\hat{p}_z$ direction
`PNMAX`	5.0	the length of the $\hat{p}_z$ vector (in atomic units).
`PPMAX`	10.0	the length of the $\hat{p}_x$ and $\hat{p}_y$ vectors (in atomic units).

section ENERGY

VAR / SWITCH	default	description
`GRID`=integer	{5,3}	5: circular energy path in complex plane 3: linear energy path parallel to real axis
`NE`=integer	{30,50}	number of E-mesh points. Note: these values of E-mesh points ensure the convergency of the spectra.
`EMIN`=real	-	real part of lowest E-value
`splitss`	ON	set-up 2 paths for energy integration dealing separately with the backscattering and single site part of the Green's function. This avoids calculation of the matrix elements for the irregular wave function.

Files used:

Filename	unit	I/O	description

DATASET.inp	5	I	input file described above
DATASET.pot	4	I	input potential read in by ⟨POTFIT⟩.
DATASET.pan	80	I	τ matrix and wave function of the positron.
DATASET_pa2d.dat	21	O	2D - $n^{2\gamma}$ (p_x, p_y) data file in a format suited for **gnuplot**.
DATASET_data.mpa	50	O	1D - $n^{2\gamma}$ (p_z) data file.

Example

For calculations of the 2D electron-positron momentum density perpendicular to [001] direction in V the input file created by xband should look like the following one.

```
###############################################################################
#  SPR-KKR input file    V_DOS.inp
#  created by xband on Thu Sep 18 13:22:16 CEST 2003
################################################################################

CONTROL  DATASET = V
         ADSI    = COMPTON
         POTFIL  = V.pot
         PRINT = 0

TAU      BZINT= POINTS  NKTAB= 250

STRCONST ETA=0.30   RMAX=3.30   GMAX=3.30

ENERGY   GRID={5,3}  NE={30,50}
         EMIN=-0.1

TASK     COMPTON   POSANI MD2D SHFT
         PNMAX=5.0   NPN=50   PPMAX=5.0   NPP=50
```

As an example, the 2D electron-positron momentum density perpendicular to [001] direction for V is shown in Fig. 6.10.

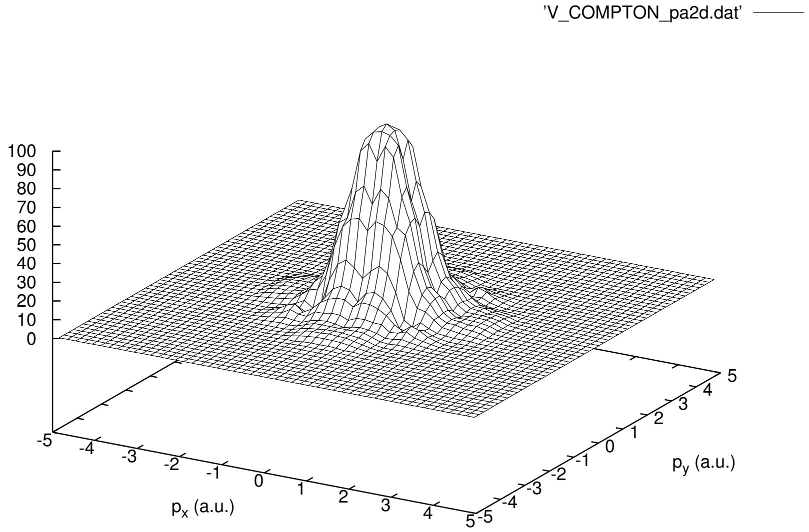


Figure 6.10: The `2D` electron-positron momentum density of bcc-V perpendicular to the [001] direction.

6.2 Calculation of spectroscopic properties included in kkrspec

6.2.1 General remarks

The **kkrspec** package is based on the one step model photoemission program as described in [27, 28, 29]. To run the **kkrspec** package in addition to the potential file `DATASET.pot` you need two files: `DATASET.inp` and `in_struc.inp`. `in_struc.inp` is a structure file which includes independent on the **kkrscf** mode the information on the semi infinite 2D structure used in **kkrspec** (see Sec. 6.2.2). Right now the **kkrspec** program includes calculations of the various properties. Those can be supplied by the corresponding part of `DATASET.inp`:

section TASK

VAR / SWITCH	default	description
`ARPES`		angle resolved photoemission spectroscopy

`AIPES`		angle integrated photoemission spectroscopy
`SPLEED`		spin polarized LEED (experimental feature)
`BAND`		band structure calculations (experimental feature)
`INPVER`	1	use 0 if you would like to use an old `input.inp` from old rslab
`STRVER`	1	0: code will look for `in_struc.inp`; 1: `in_struc.inp` will be generated automatically
		In the case of `STRVER=1` following parameters can be used in addition
`IQ_AT_SURF`	1	Specify atomic site IQ (from `DATASET.pot`) that should be at the topmost layer of the surface
`MILLER_HKL`	(0,0,1)	Miller indices for a given surface (created from bulk 3D potential`DATASET.pot`)
`CRYS_VECS`	.TRUE.	Miller indices with respect to crystalographic primitive vectors

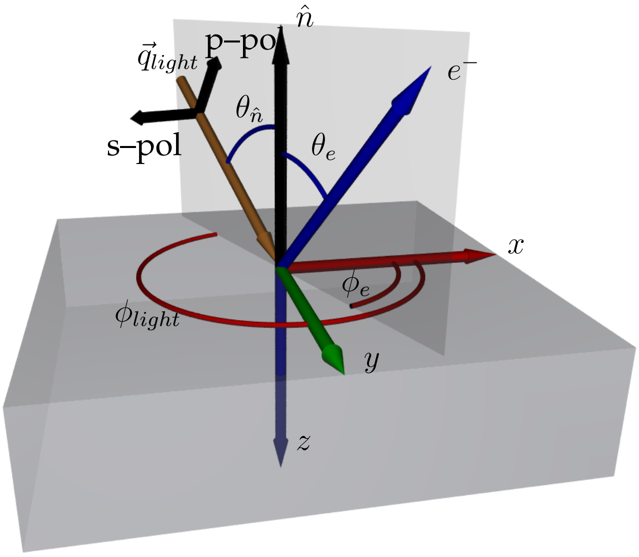
6.2.2 Structural information

kkrspec is based on the layered Korringa-Kohn-Rostokerformalism. It always runs for seminfinite 2D surfaces. Structural information about the 2D system in this mode will be stored in the `in_struc.inp` file. Depending on the option `STRVER` this file will be either automatically created (`STRVER=1`) or read in (`STRVER=0`). In the case of the automatic generation of `in_struc.inp` there are two possible cases:

- **kkrscf** calculation is done for the 3D bulk. The `in_struc.inp` for 2D surface is created from 3D bulk by specifying the corresponding Miller h,k,l indices and site IQ for the termination of the surface. Please note that this is only an approximation for the surface calculations as you will use for all layers a bulk like potential. However for the high photon energies (soft and hard x-ray PES) it is a quite good valid approximation on the other hand for the `UPS` regime typically you will get already reasonable spectra.
- In the case you performed **kkrscf** calculations for semi-infinite 2D System (`LIV`) **kkrspec** will create `in_struc.inp` automatically from the structural data of the 2D system. Please note that the surface (vacuum) potentials will be replaced by the model surface barrier.
- In the case of `STRVER=0` the **kkrspec** code will look for `in_struc.inp` file. Please note that all atomic positions are with respect to the 2D lattice vectors. The coordinate system is a lefthanded one with Z=0.0 at surface and increasing inside the stack of

layers. Each layer has to have an atom at the origin x=0.0 y=0.0 and relative shift between the layers are achieved by the `sep`.

Summary of structural information



- θ_e is defined with respect to the surface normal $\hat{n}$
- ϕ_e with respect to x
- θ_{light} is defined with respect to the z axis, e.g. if you define θ_{light} with respect to $\hat{n}$ you should use $-\theta_{\hat{n}}$ in the input file. Please note that normal incidence corresponds to $\theta_{light} = 0deg$. Often you can use $\theta_{\hat{n}}$ and ϕ_{light} instead, but you will have to take care about definition of the circularly polarised light. In doubt please check ″vacuum–wave–vector of the photon field″ and ″vector potential of the photon field″ as given in the spc–output file.
- Structural information is done with respect to the z,x,y coordinate system. In general you should check actual vectors $\vec{a}_1$ and $\vec{a}_2$ as well as reziprocal vectors. This information you can find in the spc–output file. (″reciprocal basis″)
- p–polarised light: polarisation of the light in the scattering plane (e.g. parallel and perpendicular component with respect to surface)
- s–polarised light: polarisation of the light parallel to the surface plane (e.g. only parallel component)

Please find bellow the description of an example `in_structur.inp` for the bcc Fe(001) surface.

```
# spa
   0.527000E+01
spa is the lattice constant a.
spa in a.u.
# 2d real space lattice vectors: !
# (ax, ay)
  0.100000E+01  0.000000E+00
# (bx, by)
  0.000000E+00  0.100000E+01
```

These are the 2-dimensional real space lattice vectors $\vec{a} = (a_x, a_y)$ and $\vec{b} = (b_x, b_y))$ describing the layer structure corresponding to the investigated surface. The following input will depend strongly on the type of lattice and the number of atoms in different layer. For a single atom structure like Fe(100) it looks like the following (with remarks on more complicated cases):

```
# lays
    1
```

`lays` gives the number of different layers in the structure. The following input is then repeated `lays` times.

```
# natl(1)
    1
```

`natl(lay)` is the number of atoms in the layer with number `lay=1,..,lays`. The following input is then repeated `natl(lay)` times.

```
# iat(1,1)
    1
# position: z, n, m, iq
  0.000000E+00  0.000000E+00  0.000000E+00
```

`iat(ia,lay)` is the type of atoms in locations `ia=1,2..natl(lay)`. The position (`pos(j,iat), j=1,3`) is the disposition of the atoms in locations `1,2,..,natl(lay)` along `(z,n,m)` in units of $spa, \vec{a}, \vec{b}$ with respect to the layers unit cell origin. So called `rumpled layer` are created if the `z` positions of atoms in the same layer are different. Note: There may be different number of atoms in different layers. The following input describes the sequence of layers from which the bulk crystal is build.
⇒ IQ: atomic site as defined in DATASET.pot

```
# lays  layb
    1    1
# surface barrier at sep(i,0): z, n, m
  0.290000E+00  0.000000E+00  0.000000E+00
```

`lays` is the number of layers in the overlayer plus bulk repeat unit sequence, and

`layb` is the first layer of the bulk repeat unit in sequence. The Zero layer (`sep(i,0), i=1,..,3, iseq(0)`) describes the vacuum located at (z, n, m) in units of $spa, \vec{a}, \vec{b}$.

Note: $z > 0$ is the distance of the surface from the topmost layer. At present, this has to be the same like in `in_bar.inp`.

The following input is repeated `lays` times.

```
# iseq(1)
    1
# sep(i,1): z, n, m
  0.500000E+00  0.500000E+00  0.500000E+00
# end structure
```

`iseq(lay)` is the layer type. (`sep(i,lay), i=1,..,3`) is the displacement of the origin of the unit cell in the next layer into the bulk from the unit cell origin of the current layer, along `(z, n, m)` in units of $spa, \vec{a}, \vec{b}$.

6.2.3 Angle resolved photoemission calculations

Calculations are done using the `TASK ARPES` by **kkrspec** package. Results of calculations are written in `DATASET.spc` file. Description of variables: colons are written in the header of spc file There is an additional output file `DATASET.out` with details of the **kkrspec** run. In order to run `ARPES` calculations there are general sections of the input file. **section** ENERGY

VAR / SWITCH	default	description
`EMIN_EV`		minimum of the energy window in eV with respect to the Fermi level
`EMAX_EV`		maximum of the energy window in eV with respect to the Fermi level
`NE`		number of energy points
`IMV_FIN_EV`	1.0	imaginary part of the potential in eV (final state), alternatively you can use `VIH` (in eV) or `IMV_FIN` (in Ry)

`IMV_INI_EV`	0.1	imaginary part of the potential in eV (initial state), alternatively you can use `VIL` (in eV) or `IMV_INI` (in Ry)
`EWORK_EV`	5	inner potential of the bulk crystal in eV

section SPEC_EL

VAR / SWITCH	default	description
`THETA`	45	scattering angle
`PHI`	90	scattering angle
`NT`	1	numbers of angular values for a rotation diagram C nt: polar, np: azimuth
`NP`	1	numbers of angular values for a rotation diagram C nt: polar, np: azimuth
`POL_E`	PZ	

section SPEC_EL Expert Mode

VAR / SWITCH	default	description
`TYP`	1	crystal coordinats in splout, xpsrun, or upsrun 0: i(e) diagram, 1: rotation diagram, → phi scan 2: scattering-angle diagram → theta scan 3: orthonormal projection — 3,4 only for angular resolved 4: stereographic projection — pe (ups, xps) note: nt=np → nx,ny
`ISTR`	0	beam number (h,k)
`POL0`	(0,0,0)	initial pol.
`POL0L`	(0,0,0)	initial pol. in the laboratory system
`Q1`	1+0i	amplitudes of the photoelectron used in spin polarized calculations
`Q2`	0+0i	amplitudes of the photoelectron used in spin polarized calculations

`Q3`	0+0i	amplitudes of the photoelectron used in spin polarized calculations
`Q4`	1+0i	amplitudes of the photoelectron used in spin polarized calculations

- set imaginary part of potential `VIL` and `VIH` for initial and final state to 0.0.
- layer1 set to number of layers in your surface
- number of reciprocal vectors ganz should be high enough. Always choose your vectors with respect to the parameters.
- `lanz1` and `lanz2` set to 1
  (Note: for alternative names of variables and sections see See. ARPES)

section SPEC_PH

VAR / SWITCH	default	description
`THETA`	45	
`PHI`	90	
`EPHOT`	21	photon energy in ev
`POL_P`		polarization of the light (P,S,C+,C-)

section SPEC_PH Expert Mode

VAR / SWITCH	default	description
`ALQ`	45	alignment of polarization vector or pol.ellipsis
`DELQ`	0	phase shift between real and imaginary part of e-vector, delq=90 for circular polarized light
`NPOL`	1	controls the polarization and dichroism 0: unpolarized and p-s dichroism for the calculation 1: p-pol or rcp or elliptical (depends on icirc, etc) 2: s-pol or lcp or elliptical (depends on icirc, etc) 3: dichroism (cdad, ldad)

`ICIRC`	1	controls polarisation / ellipticity of the photons 0: elliptically pol. light: alq, delq arbitrary 1: linear pol. light: alq arbitrary, delq=0 2: circular pol. light: alq=45, delq = 90
`IDREH`	0	idreh controls helicity of the photons 1, -1: sigma+, sigma- ⇒ right or left circular polarization (note: lcp, rcp are exchanged in some books) 0: linearly polarized (equals icirc=1)
`IFSP`	0	0 fixed photon azimuth angle 1: variable (equal to electron) azimuth angle
`THETA_FIX`		light and electrons are at fixed polarization angle

section SPEC_STR

VAR / SWITCH	default	description
`N_LAYDBL`		
`NLAT_G_VEC`		number of reciprocal lattice vectors
`N_LAYER`		number of bulk layer in a photoemission calculation

section SPEC

VAR / SWITCH	default	description
`SPOL`	2	

As mentioned above, the surface region is neighboured by the model surface barrier. Here are the corresponding parameters. Note: in the old vslab version those parameters have been included in file `in_bar.inp` with the following structure: This file defines the type and parameter for the surface potential. It contains the following four lines with data.

```
# ibar
  1
# epsx, barab
 0.1000000E-01 0.2900000E+00
# zpar(i),i=1,3
 0.0000000E+00-0.9000000E+00 0.9000000E+00
# bpar(i),i=1,3
 0.1340000E+01-0.2500000E+01 0.1018000E+01
```

The meaning of these parameters is:

ibar	controls type of surface barrier: = 0 step potential, = 1 rm - barrier [30], = 2 jj - barrier.
epsx	tolerance for barrier integration.
barab	distance of surface layer from the first bulk-layer.
zpar(1:3)	rm potential parameter.
bpar(1:3)	jj potential parameter.

Notes: The rm potential parameter may not be changed. The jj potential is actually not in use. The program will switch automatically to the rm potential.

section SPEC_STR

VAR / SWITCH	default	description
`SURF_BAR`		
`BARABU`	25	spin up: distance of surface layer from the first bulk-layer
`BARABD`	25	spin down: distance of surface layer from the first bulk-layer
`IBAR`	1	0: step, 1: rm-barrier, 2: rm + gauss
`EPSX`	0.005	tol for barrier integration
`ZPARU`	(0,-1,1)	spin up: rm potential parameter
`ZPARD`	(0,-1,1)	spin down: rm potential parameter
`BPAR`	(1.34,-2.0,1.018)	gaussian barrier parameter (distance, width, height)

6.2.4 Spin polarized LEED calculations

This TASK SPLEED is an experimental feature. Most of the parameters are the same as in Sec. angular resolved photoemission spectroscopy

6.2.5 Band structure calculations

In the case you performed 3D **kkrscf** calculations you can use **kkrspec** also to calculate band structure. In particular **kkrspec** allows to plot off normal and surface projected band structue. In order to perform these calculations you should set following parameters.

7 The post-processing program **plot**

Many of the files created by one of the **SPRKKR** programs may be further processed using the program **plot**, as for example the files DATASET.dos containing the component, spin and angular momentum resolved DOS. These files can be fed directly to **plot** by calling:

```
        plot < Fe.dos
```

A keyword in the first line of the data file tells **plot** how to interprete and process the data. As a results one gets a number of plot files (*.agr) that can be viewed, further processed and printed using **xmgrace**. **xmgrace** allows in particular to read in addition experimental data from other files that can be compared with the theoretical results.

Often it is necessary to supply **plot** with additional parameters to control the post-processing. This can be achieved by calling **plot** for example:

```
        plot < Fe.dos.pin
```

with a control file Fe.dos.pin that may look like this:

```
DATASET    Fe_DOS.dos

# remove leading # of sections and parameters to activate and insert values

#PLOT                             #  plot parameters
#     EMIN  =                     #  E plot range
#     EMAX  =                     #  E plot range
#     SCALE =                     #  scaling factor for functions

OUTPUT                            #  output control
     NOHEADER                     #  suppress header (xmgr-titles)
     TABLE                        #  tabulate curves in file for export
#     JDOS                        #  tabulate JDOS in file for export
```

For this control file the same rules as for the **SPRKKR** input files apply (see chapter 2), with the available sections listed below.

section DATASET

VAR / SWITCH	default	description
`FILENAME`	-	dataset file name

This information is obligatory if a control file is used. In this case **plot** reads first the control file and then opens the dataset file to read the data.

section PLOT

VAR / SWITCH	default	description
`EMIN` =real	*from data*	lower boundary of plotted energy range
`EMAX` =real	*from data*	upper boundary of plotted energy range
`SCALE`=real	-	scaling factor for functions

For most dataset file type **plot** ignores these variables in the control file and sets the energy range to be plotted according to the supplied data. Use **xmgrace** in this case to set the energy rangy.

section OUTPUT

VAR / SWITCH	default	description
`NOHEADER`	OFF	suppress the header i.e. the titles in the **xmgrace**-plots
`TABLE`	OFF	tabulate DOS or XAS curves in file for export
`JDOS`	OFF	tabulate JDOS in file for export if a DOS - file is processed

section BROADEN

VAR / SWITCH	default	description
`WG`=real	-	Gaussian broadening parameter (in eV) to account for apparative broadening when dealing with electronic spectra. Used for `TASK`: `XAS`, `XES`, `XMO`, `XRS`, `APS`, `NRAES`, `AES`, `CLXPS`, `VBXPS`.
`WL`={real,real}	-	Lorentzian broadening parameter (in eV) to account for a finite lifetime when dealing with core level spectra. In case of a spin-orbit split core level, supply 2 values for $j_{core} = l_{core} - 1/2$ and $j_{core} = l_{core} + 1/2$. Used for `TASK`: `XAS`, `XES`, `XMO`, `XRS`, `APS`, `NRAES`, `AES`, `CLXPS`.

`WLAES=` `{real,real}`	-	Lorentzian broadening parameters Γ_0 and γ (in eV) to account for a finite lifetime when dealing with Auger electron spectroscopy. The energy dependent broadening is done for $\Gamma(E) = \Gamma_0 + \gamma(E - E_F)^2$. Used for `TASK`: `NRAES`, `AES`.
`WLVBXPS=` `{real,real}`	-	Lorentzian broadening parameters Γ_0 and γ (in eV) to account for a finite lifetime when dealing with valence band photoemission spectra. The energy dependent broadening is done for $\Gamma(E) = \Gamma_0 + \gamma(E - E_F)^2$. Used for `TASK`: `VBXPS`.
`MLD`	OFF	Switch to deal with the magnetic linear dichroism instead the circular one. Used for `TASK`: `XAS`, `XES`, `CLXPS`.
`MTV`	ON	Switch to create the Bloch spectral function in a format suited for **plotmtv**. This is the default. Used for `TASK`: `BLOCHSF`.
`XMATRIX`	OFF	Switch to create the Bloch spectral function in a format suited for **xmatrix**. Used for `TASK`: `BLOCHSF`.

The program **plot** is part of the **xband** package. It is therefore most convenient to run **plot** via **xband**. **xband** allows to pass the data file directly to **plot** and to create the appropriate **xmgrace** files that way (left mouse click). Alternatively, one can invoke a menu that asks for the parameters given above (middle mouse klick) to create a control file first and then the **xmgrace** files. Because **xband** reads the keyword from the data file, the menu asks only for specific parameters.

8 Examples

The following example gives you a short introduction how to use the **SPRKKR**-package in connection with **xband**. It is assumed that you installed **xband** and that the programs **kkrscf** and **kkrgen** are compiled (see 9.1) and copied to your directory ~/bin.

Start **xband**.

Click CREATE SYSTEM.
Click Structure type.
Select BCC by mouse click.
Insert 5.4 in the entry field for a.
Click CONFIRM.
Click input OK – GO ON.
Click all done – RETURN.
Click the line

```
   1   1   a     0.000000     0.000000     0.000000
```

in the table at the bottom.
Click specify occupant.
Click Fe.
Click DONE – RETURN.
You have now created the system file Fe.sys that is listed on the info screen.

Click SPR-KKR to continue.
Click create input.
Click the scf button in the TASK field (left column).
Click write inpfile quit.

This step created the input file Fe_SCF.inp together with a potential file Fe.pot, that contains enough information to start the SCF calculations.

If you press $\boxed{\text{write+edit inpfile quit}}$ instead, the editor is invoked and you may check or modify the input file.

Click $\boxed{\text{kkrscf}}$ to invoke the menu and select run (interactive). This starts in a separate yellow xterm window the **kkrscf** program. While the program runs, you may check its progress by clicking $\boxed{\text{out-file menu}}$ and selecting grep "ERR" Fe_SCF.out. After **kkrscf** completed successfully you may destroy the yellow window – the output is kept in Fe_SCF.out.

The converged potential is now stored in Fe.pot_new. Click $\boxed{\text{files}}$ and select Fe.pot_new. Change the file name in the entry field (right bottom) to Fe.pot and click $\boxed{\text{move}}$ to move Fe.pot_new to Fe.pot. This step is done because the default name for the potential file is Fe.pot.

Click $\boxed{\text{create input}}$.
Click $\boxed{\text{write inpfile quit}}$.
This step creates the input file Fe_DOS.inp for a DOS calculation.

Click $\boxed{\text{kkrgen}}$ and select file run (interactive) option. As before **kkrgen** runs in a yellow xterm window and writes output simultaneously to Fe_DOS.out. In addition a data file Fe_DOS.dos will be created.

Click $\boxed{\text{plot}}$ to invoke the plot menu. Select Fe_DOS.dos in the datafile list by left mouse click. This invokes the program **plot** that creates a file Fe_DOS.dos_Fe.agr listed in the right file column. Select this file by left mouse click. This last step invokes the graphics program **xmgrace** that should now display the spin-resolved DOS of bcc-Fe.

Further examples for the use of the **SPRKKR**-package can be downloaded from http://olymp.cup.uni-muenchen.de/ak/ebert/SPRKKR/examples.html. The downloaded file has to be unpacked using `gunzip` and `tar`. It contains a postscript file describing a number of applications of the **SPRKKR**-package with the corresponding input and output files included. More information can be found in the enclosed `README`-file.

9 Technical details, installation and dimensioning

9.1 Technical details

The **SPRKKR**-package is a collection of programs written throughout in FORTRAN 77 using some FORTRAN 95 extensions, that make the use of a FORTRAN 95 compiler unavoidable. The program package has been used so far under various Unix platforms (IBM, Compac/DEC, PC (Linux), Fujitsu, Cray) without problems, but is mainly developed and tested on PC Linux with Intel FORTRAN compiler. Of course the required computer resources depend on the size of the investigated system. For many applications a PC under Linux with around 1-2 GHz, 512 MB core memory and 300 MB disk space should be sufficient.

The **SPRKKR**-package is split into the programs **kkrscf**, **kkrgen**, **kkrchi**, **kkrspec**, **embgen** and **embscf** with the corresponding main programs in the files kkrscf.f, kkrgen.f, kkrchi.f, kkrspec.f, embgen.f and embscf.f respectively, that in turn are generated automatically from the generic file kkrmain.f. The programs have many subroutines in common. Most of the subroutines are in separate files with the file name identical to the name of the subroutine. All source files are collected in one single directory together with a makefile and several auxiliary files.

9.2 Installation

9.2.1 Required libraries

The **SPRKKR**-package uses the mathematical libraries `BLAS` and `LAPACK`.

SPRKKR uses MPI libraries for multi processor calculations. For compiling and using **SPRKKR**, an MPI environment like `openmpi`, including its development packages, has to be installed.

If these libraries are not available on your computer, you usually can use your Linux's software management to install them.

9.2.2 Unpacking

To install the **SPRKKR**-package, create a program directory, copy the program package file to that directory and unpack it:

```
mkdir progdir
cp sprkkr.tar.gz progdir
cd progdir
tar -xvfz sprkkr.tar.gz
```

9.2.3 Preparing the make file

To configure the makefile, copy the supplied `make.inc_example` to `make.inc` and adapt it to your system.

Use the variable `LIB` to supply the necessary information where to find the mathematical libraries `BLAS` and `LAPACK` on your computer.

Even if not using MPI, you have to specify the correct path to mpif.h with the `INCLUDE` variable, e.g. `INCLUDE= -I/usr/lib/openmpi/include`

The variables `FC` and `LINK` contain the FORTRAN compiler. When using openmpi, use the compiler wrapper `mpif90.openmpi` .

Depending on your FORTRAN compiler, you have to adapt the compiler flags in the variable `FFLAGS`. Confer to your compiler's documentation to find settings for optimization and debugging that fit best to your needs and your CPU.

Specify via the variable `BIN` in which directory you want to keep the executable programs, for example by setting `BIN=.` or `BIN=~/bin` .

Specify a string to be added to each program name with the `VERSION` variable, e.g. a version name. Using an empty string, `VERSION=` ,yields the plain program name **kkrgen** etc.

9.2.4 Compiling

To create the executable programs **kkrscf**, **kkrgen**, **kkrchi**, **kkrspec** and the embedded cluster programs **embscf** and **embgen** use make by calling

```
make scf
make gen
make chi
make spec
make embscf
make embgen
```

or

```
make all
```

and to create the MPI versions call

```
make scfmpi
```

and so on, or

```
make allmpi .
```

The binaries will be named `kkrscf<VERSION>` etc. and `kkrscf<VERSION>MPI` .

The programs can now be used in the standard way, e.g. via

```
/home/.../progdir/kkrscf < Fe.inp > output
```

MPI calculations are started via

```
mpirun -np X /home/.../progdir/kkrscfMPI Fe.inp > output
```

where X is the number of CPUs you want to use. Note the missing symbol "<". Of course the full path `/home/.../progdir/` can be omitted if you add the program directory to your standard path PATH. Alternatively you may copy the executable to any directory that is in your standard path PATH, as e.g. ~/bin.

9.3 Dimensioning

The dimension of all arrays that depend on the specific application are fixed automatically using the FORTRAN allocate function. This means that the users is normally not bothered by dimensioning as long as the system he wants to deal with does not require more storage than his computer supplies. In that case the program will crash with a segmentation error (i.e. the user has to be less ambitious or to buy more core memory ...).

The only exception is the variable `NRMAX` that is fixed to 750 in the file mod_rmesh.f. Thus, if more radial mesh points should be used the setting in mod_rmesh.f should be changed and the programs recompiled.

Bibliography

[1] J. S. Faulkner, Prog. Mater. Sci. **27**, 1 (1982).

[2] P. Weinberger, *Electron Scattering Theory for Ordered and Disordered Matter* (Oxford University Press, Oxford, 1990).

[3] A. Gonis, *Green functions for ordered and disordered systems* (North-Holland, Amsterdam, 1992).

[4] A. Gonis and W. H. Butler, *Multiple scattering in solids*, *Graduate Texts in Contemporary Physics* (Springer, Berlin, 1999).

[5] H. Ebert, in *Electronic Structure and Physical Properties of Solids*, Vol. 535 of *Lecture Notes in Physics*, edited by H. Dreyssé (Springer, Berlin, 2000), p. 191.

[6] H. Ebert and E. K. U. Gross, in *Relativistic Effects in Heavy-Element Chemistry and Physics*, edited by B. Hess (Wiley, Chichester, 2003), p. 163.

[7] H. Ebert, D. Ködderitzsch, and J. Minár, Rep. Prog. Phys. **74**, 096501 (2011).

[8] H. Ebert, Rep. Prog. Phys. **59**, 1665 (1996).

[9] M. E. Rose, *Elementary Theory of Angular Momentum* (Wiley, New York, 1957).

[10] M. E. Rose, *Relativistic Electron Theory* (Wiley, New York, 1961).

[11] H. Ebert, H. Freyer, A. Vernes, and G.-Y. Guo, Phys. Rev. B **53**, 7721 (1996).

[12] H. Ebert, H. Freyer, and M. Deng, Phys. Rev. B **56**, 9454 (1997).

[13] H. Ebert, A. Vernes, and J. Banhart, Solid State Commun. **104**, 243 (1997).

[14] H. Ebert, P. Strange, and B. L. Gyorffy, J. Phys. F: Met. Phys. **18**, L135 (1988).

[15] J. Strempfer *et al.*, Eur. Phys. J. B **14**, 63 (2000).

[16] J. B. Staunton *et al.*, Phys. Rev. B **74**, 144411 (2006).

[17] S. Bornemann *et al.*, Eur. Phys. J. D **45**, 529 (2007).

[18] X. Wang, D. Wang, R. Wu, and A. J. Freeman, J. Magn. Magn. Materials **159**, 337 (1996).

[19] L. Udvardi, L. Szunyogh, K. Palotás, and P. Weinberger, Phys. Rev. B **68**, 104436 (2003).

[20] I. Turek, J. Kudrnovský, and K. Carva, Phys. Rev. B **86**, 174430 (2012).

[21] S. Ostanin, V. Popescu, H. Ebert, and H. Dreyssé, J. Phys.: Cond. Mat. **11**, 9095 (1999).

[22] J. Minár and H. Ebert, Solid State Commun. **118**, 383 (2001).

[23] H. Ebert and V. Popescu, Phys. Rev. B **56**, 12884 (1997).

[24] J. Minár, V. Popescu, and H. Ebert, Phys. Rev. B **62**, 10051 (2000).

[25] H. Ebert and J. Schwitalla, Phys. Rev. B **55**, 3100 (1997).

[26] H. Ebert and G. Y. Guo, J. Magn. Magn. Materials **148**, 174 (1995).

[27] J. Braun, Rep. Prog. Phys. **59**, 1267 (1996).

[28] J. Minár, J. Braun, S. Mankovsky, and H. Ebert, J. Electron. Spectrosc. Relat. Phenom. **184**, 91 (2011).

[29] J. Braun *et al.*, Phys. Rev. B **82**, 024411 (2010).

[30] G. Malmström and J. Rundgren, Comp. Phys. Commun. **19**, 263 (1980).