# The Augmented Spherical Wave Method

An Extended User Guide

Version 1.9

Volker Eyert

April 14, 2005

Copyright © 1999-2003 Volker Eyert
All rights reserved

# Contents


**1 Introduction** **1**
- 1.1 Overview . . . 1
- 1.2 Physical background . . . 2
- 1.3 Programming . . . 5
- 1.4 Installation . . . 6
- 1.5 Linking with other software . . . 7
- 1.6 Legal matters . . . 7
- 1.7 Known bugs . . . 9
- 1.8 Acknowledgement . . . 9

**2 Execution of the ASW programs: Examples** **11**
- 2.1 A simple case: Cu . . . 11
  - 2.1.1 CTRL file . . . 12
  - 2.1.2 Execution of the main programs . . . 13
  - 2.1.3 Execution of the plot programs . . . 15
  - 2.1.4 Advanced features . . . 21
  - 2.1.5 Crystal orbital overlap population . . . 23
  - 2.1.6 Fermi surface . . . 27
- 2.2 A more complicated structure: $FeS_2$ . . . 28
  - 2.2.1 CTRL file and sphere packing . . . 28
  - 2.2.2 Execution of the main programs . . . 37
  - 2.2.3 Execution of the plot programs . . . 38
- 2.3 A magnetic system: $CrO_2$ . . . 40
  - 2.3.1 CTRL file and sphere packing . . . 42
  - 2.3.2 Spin-degenerate calculations . . . 44
  - 2.3.3 Spin-polarized ferromagnetic calculations . . . 56
  - 2.3.4 Spin-polarized antiferromagnetic calculations . . . 59

**3 Organization of the ASW program package** **63**
- 3.1 Main programs and shellscripts . . . 63
  - 3.1.1 mnmpr.run, mnmpr.x . . . 63
  - 3.1.2 mnhlp.x . . . 64
  - 3.1.3 mnsym.run, mnsym.x . . . 67
  - 3.1.4 mnfre.run, mnfre.x . . . 67
  - 3.1.5 mnovl.x . . . 67
  - 3.1.6 mnpac.run, mnpac.x . . . 68

3.1.7 mnstr.x . . . 68
3.1.8 mnscf.run, mnscf.x . . . 68
3.1.9 mndos.x . . . 69
3.1.10 mnbnd.run, mnbnd.x . . . 69
3.1.11 mnall.x . . . 69
3.1.12 mnscl.run, mnscl.x . . . 70
3.1.13 upctr . . . 70
3.1.14 susan . . . 70
3.2 Plot programs and shellscripts . . . 71
3.2.1 plstr.run, plstr.x . . . 72
3.2.2 plfre.run, plfre.x . . . 72
3.2.3 plbnd.run, plbnd.x . . . 72
3.2.4 pldos.run, pldos.x . . . 73
3.2.5 plcop.run, plcop.x . . . 73
3.2.6 plbnd.lx, pldos.lx, plcop.lx . . . 73
3.2.7 plbnd.tex, pldos.tex, plcop.tex . . . 73
3.3 Installation shellscripts . . . 73
3.3.1 Makefile . . . 73
3.3.2 mkall.x . . . 74
3.3.3 upshl . . . 74
3.4 Data files . . . 74
3.4.1 README . . . 74
3.4.2 INSTALL . . . 74
3.4.3 COPYRIGHT . . . 74
3.4.4 LICENCE . . . 74
3.4.5 CTRL . . . 74
3.4.6 CBAK . . . 75
3.4.7 CNEW . . . 75
3.4.8 HELP . . . 75
3.4.9 The atomic files . . . 82
3.4.10 FREE . . . 83
3.4.11 STRU . . . 83
3.4.12 MIX . . . 84
3.4.13 BNDE . . . 84
3.4.14 BNDV . . . 84
3.4.15 DOS . . . 84
3.4.16 COOP . . . 84
3.4.17 out*** . . . 84
3.4.18 bnd.tex, dos.tex, coop.tex . . . 84
3.4.19 free.ps, bnd.ps, dos.ps, coop.ps . . . 84

**4 The main input file: CTRL 87**
4.1 Category HEADER . . . 88
4.2 Category VERSION . . . 88
4.2.1 Token ASW- . . . 88
4.3 Category IO . . . 88

4.3.1 Token HELP= . . . 89
4.3.2 Token SHOW= . . . 89
4.3.3 Token VERBOS= . . . 89
4.3.4 Token IACTIV= . . . 89
4.3.5 Token CLEAN= . . . 90
4.3.6 Token WRITE= . . . 90
4.3.7 Token EXTENS= . . . 90
4.4 Category OPTIONS . . . 90
4.4.1 Token REL= . . . 90
4.4.2 Token LSCPL= . . . 90
4.4.3 Token NSPIN= . . . 90
4.4.4 Token AFSYM= . . . 91
4.4.5 Token BEXT= . . . 91
4.4.6 Token XCPAR= . . . 91
4.4.7 Token GGA= . . . 92
4.4.8 Token OVLCHK= . . . 92
4.4.9 Token CCOR= . . . 92
4.4.10 Token FULPOT= . . . 92
4.4.11 Token CORDRD= . . . 92
4.5 Category STRUC (mandatory) . . . 93
4.5.1 Token UNITS= . . . 93
4.5.2 Token ALAT= (mandatory) . . . 93
4.5.3 Token PLAT= (mandatory) . . . 93
4.5.4 Token SLAT= . . . 93
4.5.5 Token BBYA= . . . 94
4.5.6 Token BLAT= . . . 94
4.5.7 Token CBYA= . . . 94
4.5.8 Token CLAT= . . . 94
4.5.9 Token GAMMA= . . . 94
4.6 Category CLASS (mandatory) . . . 94
4.6.1 Token NCLASS= . . . 95
4.6.2 Token ATOM= (mandatory) . . . 95
4.6.3 Token Z= (mandatory) . . . 95
4.6.4 Token R= . . . 95
4.6.5 Token R/RA= . . . 95
4.6.6 Token LMXL= . . . 95
4.6.7 Token LMXI= . . . 96
4.6.8 Token CONF= . . . 96
4.6.9 Token COORB= . . . 96
4.6.10 Token QVAL= . . . 96
4.6.11 Token MVAL= . . . 96
4.7 Category SITE (mandatory) . . . 96
4.7.1 Token NBAS= . . . 97
4.7.2 Token CARTP= . . . 97
4.7.3 Token CHOUT= . . . 97
4.7.4 Token ATOM= (mandatory) . . . 97

4.7.5 Token POS= (mandatory) . . . 97
4.7.6 Token SPIN= . . . 97
4.8 Category SYMGRP . . . 97
4.8.1 Token GENPOS= . . . 98
4.8.2 Token SYMOPS= . . . 98
4.8.3 Token CARTR= . . . 99
4.8.4 Token CARTT= . . . 99
4.9 Category PACK . . . 99
4.9.1 Token FILLNG= . . . 99
4.9.2 Token OBYDMX= . . . 99
4.9.3 Token OBYRMX= . . . 99
4.9.4 Token ESBONS= . . . 100
4.9.5 Token NCEMAX= . . . 100
4.9.6 Token RADMIN= . . . 100
4.9.7 Token RADMAX= . . . 100
4.9.8 Token POTWIN= . . . 100
4.9.9 Token RADACC= . . . 101
4.9.10 Token POSACC= . . . 101
4.10 Category ENVEL . . . 101
4.10.1 Token NKAP= . . . 101
4.10.2 Token EKAP= . . . 101
4.10.3 Token EWPAR= . . . 101
4.10.4 Token EWTOL= . . . 102
4.11 Category BZSMP . . . 102
4.11.1 Token NKABC= . . . 102
4.11.2 Token BZINT= . . . 102
4.11.3 Token EMIN= . . . 103
4.11.4 Token EMAX= . . . 103
4.11.5 Token NDOS= . . . 103
4.11.6 Token NORD= . . . 103
4.11.7 Token WIDTH= . . . 103
4.11.8 Token EFTOL= . . . 103
4.11.9 Token SAVDOS= . . . 104
4.11.10 Token SAVCOOP= . . . 104
4.11.11 Token CTYPE= . . . 104
4.11.12 Token MSPLIT= . . . 104
4.11.13 Token TEMPFD= . . . 104
4.12 Category CONTROL . . . 105
4.12.1 Token START= . . . 105
4.12.2 Token QUIT= . . . 106
4.12.3 Token FREE= . . . 106
4.12.4 Token NITBND= . . . 106
4.12.5 Token CNVG= . . . 107
4.12.6 Token CNVGET= . . . 107
4.12.7 Token NITATM= . . . 107
4.12.8 Token CNVGQA= . . . 107

4.13 Category MIXING . . . 108
4.13.1 Token NMIXB= . . . 108
4.13.2 Token BETAB= . . . 108
4.13.3 Token INCBB= . . . 108
4.13.4 Token NMIXA= . . . 108
4.13.5 Token BETAA= . . . 108
4.14 Category SUPCELL . . . 109
4.14.1 Token ALAT= . . . 109
4.14.2 Token PLAT= . . . 109
4.14.3 Token SLAT= . . . 109
4.14.4 Token BBYA= . . . 109
4.14.5 Token BLAT= . . . 110
4.14.6 Token CBYA= . . . 110
4.14.7 Token CLAT= . . . 110
4.14.8 Token GAMMA= . . . 110
4.14.9 Token EQUIV= . . . 110
4.14.10 Token CARTS= . . . 110
4.14.11 Token PSHIFT= . . . 110
4.14.12 Token CARTQ= . . . 110
4.14.13 Token QSWAVE= . . . 111
4.15 Category SYMLIN . . . 111
4.15.1 Token NPAN= . . . 111
4.15.2 Token NPTS= . . . 111
4.15.3 Token ORBWGT= . . . 111
4.15.4 Token CARTE= . . . 111
4.15.5 Token LABEL= . . . 112
4.15.6 Token ENDPT= . . . 112
4.16 Category PLOT . . . 112
4.16.1 Token CARTV= . . . 112
4.16.2 Token ORIGIN= . . . 112
4.16.3 Token RPLOT1= . . . 112
4.16.4 Token RPLOT2= . . . 112
4.16.5 Token RPLOT3= . . . 112
4.16.6 Token NPDIV1= . . . 112
4.16.7 Token NPDIV2= . . . 113
4.16.8 Token NPDIV3= . . . 113

**5 The ASW database** **115**

**A Brillouin zones** **121**

**Bibliography** **125**

# Chapter 1

# Introduction

## 1.1 Overview

Since its invention in the late seventies the Augmented Spherical Wave (ASW) method has become one of the most widespread methods used for density functional based electronic structure calculations. Its minimal basis set allows for a natural interpretation of materials properties and makes it one of the fastest all-electron methods. The present User Guide addresses to practioners who want to apply the ASW method to actual problems but are not (yet) interested in a detailed understanding of the underlying formalism.

The original version of the ASW method was developed in the seventies at the IBM research lab in Yorktown Heights by A. R. Williams, J. Kübler, and C. D. Gelatt [10]. According to the authors, their worked was based on the concept of renormalized atom calculations as proposed by Watson, Ehrenreich, Hodges, and Gelatt [13, 12, 11] and inspired by the ideas of the linear methods as presented by O. K. Andersen [14].

In the eighties, the ASW program saw several revisions and extensions mainly done by the Darmstadt group of J. Kübler. These included the development of a version for treating non-collinear spin arrangements [8, 9] as well as the implementation of a first full potential ASW method by myself [1]. In addition, J. Sticht and I put a lot effort in tuning the program to optimal performance especially for vector machines.

Commercializing the code set in by the end of the eighties, when the California based software company BIOSYM (later Materials Science Incorporation, MSI, now Accelrys) started distribution of the standard Darmstadt code.

Nevertheless, the old Darmstadt version still suffered from many drawbacks. Programming was completely done in a rather old fashioned Fortran77 style using e.g. variables and arrays of mixed accuracy. File handling was quite complicated and reading from the input files was done in fixed format. In the startup phase of a calculation several different programs and input files had to be used. All this made the program rather user-unfriendly and error-prone.

In order to overcome these difficulties of the old Darmstadt version, I started a completely new implementation of the code. Programming was done in a clean and systematic style. The source was held fully self-contained by including standard

BLAS or LAPACK routines for the linear algebra problems. Graphics is based to large parts on Gnuplot, which is likewise public domain software. The file handling and the organization of the program was much improved taking away many standard steps from the users responsibility. The interface between the user and the program was completely reshaped and now allows for a very flexible input. Furthermore, a lot more properties can be calculated, which fact facilitates analysis of the electronic properties a lot and make the program exceptional among many other methods. Last not least, the program package has turned out to be very stable and efficient. The latest version currently available is ASW-1.9, to which this handbook applies. This version was originally designed for Unix/Linux machines and most of the commands mentioned below refer to this environment. More information about the ASW method and the program package can be obtained from the ASW homepage,

**http://www.physik.uni-augsburg.de/∼eyert/aswhome.shtml**

where most of the properties accessible by the programs are discussed and a full list of publications and theses related to the ASW method is given.

Since 1998 a WINDOWS version of the present ASW version including a very clever graphics interface and access to the worlds largest crystallographic database is distributed under the name "Electra" (for "Electronic Structure and Analysis") by Materials Design Inc., Angel Fire, NM and Materials Design s.a.r.l., Le Mans. See

**http://www.materialsdesign.com**

for more information.

Any questions, criticisms, and suggestions concerning the ASW program package or this manual are welcome. Please feel free to contact me via email to

**eyert@physik.uni-augsburg.de**

This user guide falls into three parts. Following this introductory chapter the capabilities of the program package are demonstrated by three examples in Chap. 2. Chap. 3 comprises an overview over all programs, shellscript, and data files coming with the distribution or being created during execution. A detailed discussion of the main input file of the package, the CTRL file is finally given in Chap. 4. While the more experienced practioner will use this chapter as a reference, beginners are strongly recommended to work through the examples of Chap. 2 first.

## 1.2 Physical background

As many other *ab initio* approaches the ASW method is based on the Born-Oppenheimer approximation, which allows to consider the electronic structure independent of the lattice dynamics. Furthermore, it makes use of crystalline periodicity and the Bloch theorem built thereon. Finally, the ASW method relies on density functional theory (DFT) as founded by Hohenberg, Kohn, and Sham.

As it stands the ASW program package employs the local density approximation (LDA) coming with DFT. By now, all common parametrizations have been implemented. Alternatively, the generalized gradient approximation (GGA) in all known parametrizations can be used.

As all other augmentation schemes the method is built upon a special form of the muffin-tin approximation (MTA), namely the atomic sphere approximation (ASA) as invented by Andersen. It is characterized by the requirement that the atomic spheres must fill space completely. Inside the spheres the potential as well as the charge densities are assumed to be spherical symmetric. While this socalled shape approximation of both the MTA and the ASA might seem as crude approximations to the full crystal potential the ASA actually is not. This can be readily understood from the fact that the ASA condition creates overlap regions in the bonding region between two atoms. In these regions the sum of the two ASA potentials experiences an effective downshift as compared to the potentials of the single spheres and thus mimics the full potential quite well. Indeed, electronic properties calculated using ASA based methods can be hardly distinguished from those growing out of the much more elaborate full potential methods.

However, due to the spherical symmetry and the overlap of the potential wells calculated total energies still lack the accuracy needed e.g. for calculating elastic constants of phonon frequencies. For this reason I have implemented a new full potential version, which will be soon available.

As an all-electron scheme the ASW method fully includes both the core and the valence electrons in the self-consistent field calculations. For this reason, it allows full coverage of the periodic table including transition metal atoms as well as the lanthanides and actinides. The codes is applicable to metal, semiconductors and insulators without any restrictions.

Much of the efficiency of the program package stems from the use of a minimal basis set built from atomiclike ($s$, $p$, $d$) orbitals. As a consequence, each atom contributes with nine or 16 basis functions to the secular matrix, which is much less than the $\approx 100$ functions necessary in plane wave based methods. Furthermore, the atomiclike orbitals allow for an easy setup of e.g. partial densities of states and a much more natural interpretation of the results.

Additional speedup of the program results from the fact that the ASW method like the LMTO or LAPW is a socalled linearized method. The latter approach as invented by Andersen is based on the observation that the energy dependence of the wave function is rather weak and, hence, may be well approximated by only few terms of a Taylor expansion. Within the ASA only the first two terms are actually needed (the famous $\phi$ and $\dot{\phi}$ of the LMTO method). As a result, solving for the zeros of the secular matrix reduces to an eigenvalues problem, which is tractable by standard routines, and can thus be done much faster as the complicated root tracing done before.

Finally, as it stands the ASW package allows for non-relativistic as well as scalar-relativistic calculations and it can handle both spin-restricted and spin-polarized calculations, the latter including a special branch for antiferromagnetic order. Much work has been put into the code for accelerating the iterations towards self-consistency.

Last not least, due to the implementeation of the sphere geometry optimization

(SGO) algorithm the ASW programs allow for considering closed-packed systems as well as rather open crystal structures. The SGO algorithm performs an automatic sphere packing including the generation of empty sphere positions and determination of optimal atomic sphere radii for use with the ASA.

Over the years more and more physical and chemical properties have become available by the ASW program package. Among them are the following.

- Electronic Properties
  - electronic dispersions $E(\mathbf{k})$ ("band structure")
  - electronic wave functions $\psi(\mathbf{k}) = \sum_i c_i H_i(\mathbf{k})$ $\rightarrow$ projected band structure
  - total/partial (site/state projected) densities of states (DOS)
  - Fermi surface
  - charge densities (near future)
  - electric field gradients (near future)
  - charge densities at the nuclei $\rightarrow$ isomer shifts
- Cohesive and Elastic Properties
  - cohesive energy
  - bulk modulus
  - elastic constants (near future)
  - phonon frequencies (near future)
  - forces $\rightarrow$ structure optimization (near future)
- Chemical Bonding
  - total/partial crystal orbital overlap population (COOP), total/partial crystal orbital Hamiltonian population (COHP), total/partial covalence energy ($E_{cov}$)
- Magnetic Properties
  - total and site/state projected magnetic moments
  - magnetic ordering (ferro-, ferri-, antiferromagnetic)
  - magnetic energy gains
  - spin densities (near future)
  - spin densities at the nuclei $\rightarrow$ hyperfine fields

Suggestions for additional properties, which should be implemented are always welcome. Just sent an email to the address given above.

## 1.3 Programming

At present, the ASW program package consists of ≈ 100000 lines of source code, which are organized in a dozen main programs and about 400 subroutines. While the first versions were written in Fortran77, the latest version has been fully adapted to the new Fortran95 standard. In particular, dynamic memory now makes the package much faster and more flexible than before. Moreover, switching to Fortran95 has enhanced portability of the code considerably.

In setting up the program I have much benefitted from *ftnchek*, which is a public domain tool for syntax checking of Fortran programs. Among other things it provides means to check data exchange between the main program and subroutines, which usually is note done by a compiler. I higly recommend *ftnchek* to any Fortran programmer.

As it stands the ASW program package is fully self-contained without the need to link any library. Standard routines for e.g. matrix inversion and diagonalization are taken from the LAPACK library. For plotting the calculated data the programs provide interfaces to GNUPLOT, RASMOL, and XMakeMol, which are public domain software. As a consequence, commercial libraries are not needed at all.

Finally, the program is fully portable to and has been tested on a large variety of platform including

- IBM RS6000 (AIX; xlf, xlf90)
- HP 9000 (HP-UX; f90)
- CRAY Y-MP, YEL, J90, T90 (UNICOS; f90)
- SNI S400 (UXP/M; frt)
- SNI VPP500 (UXP/M; frtpx)
- Sun SPARC (Solaris; f90)
- Convex (ConvexOS; fc -f90)
- Compaq Alpha (True64 UNIX; f90)
- PC (AMD/Intel) (Linux; VAST f95, Lahey/Fujitsu Fortran f95, PGI f90, AbsoftPro Fortran, NAG f95)
- PC (AMD/Intel) (Windows95/98/NT; Compaq Visual Fortran, Lahey Fortran 95)
- all platforms (GNU g95)

For this reason, installing the package poses no problems and can be done within a few minutes.

## 1.4 Installation

Since the ASW package has been written with portability in mind, installation on various platforms is straightforward. In particular, only few steps are needed.

1. Edit the Makefile:
   In the first part of the Makefile several items must be specified:

   | | |
   |---|---|
   | COMPILER | Machine and/or compiler |
   | ARCH | Machine architecture (for IBM only) |
   | LOCALLIB | Use of local BLAS library (recommended) |
   | APPLICATION | Switch between testing and production runs |

   The ASW program package uses several routines from the LAPACK and BLAS libraries. The source codes of these routines are included in the distribution and will be compiled and linked into the executables in case the entry LOCALLIB is left empty. This option makes the ASW package completely self-contained. No additional libraries are necessary to run the programs. Yet, on most machines vendor specific installations of these libraries exist, which are able to make use of machine specifications and thus in many cases are very fast. Hence, whenever a precompiled LAPACK or BLAS routine exists, it's use is highly recommended. In this case you should specify LOCALLIB = USE. Recent developments in the field of mathematical libraries have led to the Automatically Tuned Linear Algebra Subroutine (ATLAS) library, which is available also for PC's running under Linux, could be easily installed, and outperforms even some vendor supplied libraries.

   For the last item, please use APPLICATION = PROD. Finally, the installation directory, where the executables and the LaTeXenvelope files will be copied, must be given.

2. Edit the files stuni.f, jbout.f and timex.f:
   Select the settings valid for your machine by (un)commenting the corresponding lines in the source code. Yet, due to the increased portability of Fortran95, this step can be skipped on most machines.

3. Edit the file mnbeg.f:
   This file is common to all main programs and contains parameters, which control the size of many arrays. Please check the maximum number of interstitial energies (kappa-values; keep 1 if you are in doubt) and the maximum number of atoms. Again, this step can be skipped in the first installation.

4. Type make all:
   This step, which alternatively can be performed as background procedure by invoking the shell-script mkall.x, generates all executables. Due to the many new features incorporated in the Makefile its execution might require GNU make, which is free software and available from many servers, e.g. www.gnu.org.

5. Type make install:
   This copies all executables, shell-scripts and LaTeXenvelope files to the directory specified in step 1 above.

## 1.5 Linking with other software

There are some public domain software packages used by the ASW software, namely BLAS, ATLAS, LAPACK, GNUPLOT, RasMol, XMakeMol, and ftnchek, which are all public domain.

However, all BLAS, ATLAS, and LAPACK routines needed by the ASW software are already included in the ASW package. So, you don't need to transfer extra files.

## 1.6 Legal matters

Since the ASW program package is commercially distributed legal matters have to be considered. Most of this is contained in the file COPYRIGHT coming with the distribution, which is printed here for completeness.

```
**************************************************************************
***                      ASW software package                          ***
**************************************************************************
***                           Version 1.9                              ***
**************************************************************************
***                         Copyright notice                           ***
**************************************************************************
**************************************************************************

    The ASW software package was written by Volker Eyert.

    In setting up the package the author has benefitted from
    experience gained during stays at the following institutes:

      Institut fuer Festkoerperphysik, TH Darmstadt,
      Hochschulstr. 6, D-64289 Darmstadt
      Institut fuer Physikalische Chemie, TH Darmstadt,
      Petersenstr. 20, D-64287 Darmstadt
      Max-Planck-Institut fuer Festkoerperforschung,
      Heisenbergstr. 1, D-70569 Stuttgart
      Hahn-Meitner-Institut,
      Glienicker Str. 100, D-14109 Berlin
      Institut fuer Physik, Universitaet Augsburg,
      Memminger Str. 6 and Universitaetsstr. 1, D-86135 Augsburg

**************************************************************************

    Version ASW-1.9 11.01.2002 Volker Eyert
    Copyright (C) 1992-2002 Volker Eyert
**************************************************************************


General:
```

```
Here and in the following the term "the author" refers to Volker Eyert
at the last of the above adresses.

You can contact the author via normal mail to the last of the above
adresses or via email to:
  eyert@physik.uni-augsburg.de

Here and in the following the term "ASW software package" refers to the
total of all files contained in the distribution (in source and object
form).

Here and in the following the term "core of the ASW software package"
refers to the total of all files contained in the distribution (in
source and object form) except for those files which explicitly contain
a copyright notice by other authors and except for the LAPACK and BLAS
routines coming with the distribution.


  ***********************************************************************


 Copyright notice:

The core of the ASW software package is copyrighted software.

It is not allowed to change any of the copyright notices in the core
of the ASW software package.

It is not allowed to redistribute the core of the ASW software package
or any part of it without prior written permission of the author.

It is not allowed to incorporate any part of the core of the ASW software
package into any other software without prior written permission of the
author.

It is illegal to commercially distribute the core of the ASW software
package as a whole or any part of it or to incorporate any part of it
into a commercial product without prior written permission of the
author.


  ***********************************************************************


 Disclaimer:

The ASW software package is distributed in the hope that it will be
useful, but WITHOUT ANY WARRANTY; without even the implied warranty
of MERCHANTIBILITY or FITNESS FOR A PARTICULAR PURPOSE.


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

## 1.7 Known bugs

So far the following bugs are known, which have not yet been removed from the code:

1. The packing program mnpac.run was originally designed for the spin-degenerate case only. As a consequence, it produces wrong results if NSPIN=2 is specified in the CTRL file. Please make sure that NSPIN=1 in the CTRL file or that this token is missing at all.

## 1.8 Acknowledgement

In setting up the ASW package I have benefitted from discussions with quite many people. Not trying to be complete I mention especially Ole Andersen, Jürgen Kübler, Alexander Mavromaras, Michael Methfessel, Peter Schmidt, Michael Stephan, Jürgen Sticht, and Erich Wimmer.

# Chapter 2

# Execution of the ASW programs: Examples

In the present chapter we will discuss how the programs coming with the ASW distribution can be used. In doing so we will apply the programs to two different examples, namely Cu, $FeS_2$, and $CrO_2$.

## 2.1 A simple case: Cu

Elemental copper has been used as a test case for electronic structure programs since long. This is due to several reasons. As we will learn below the electronic structure of this metal is strongly influenced by three groups of electrons. While the core electrons are tightly bound to the nucleus and, hence, have a very high charge density at the center of the atom but an almost negligible density in the outer region, the wave functions of the $4s$ and $4p$ electrons reach far out and spread even in the region of neighbouring atoms. Finally, the $3d$ electrons are somewhat intermediate in that they likewise contribute to the overlap with neighbouring atoms, hence, to the metallic bonding but still are rather well localized near the nuclei. Since the $3d$ states of copper are almost filled they contribute only weakly to the Fermi surface, which is therefore formed mostly by the $4s$ and $4p$ states. As a consequence, although the $3d$ states are found well below $E_F$ any changes or errors in the $4sp$–$3d$ overlap strongly affect the Fermi surface. For this reason copper is regarded as a sensitive test case. Calculated electronic structures for this material were compared to the results of photoemission experiments already in the early 1960's by Burdick [17].

The following calculations proceed in several steps. First we have to set up the CTRL file, which is the only input file for the calculations. All other files needed during the execution of the programs are created automatically and they are deleted in case the program does not need them any more. Once the self-consistent calculation has converged, additional main programs may be invoked to evaluate the band structure or the partial DOS. Finally, the plot programs can be used to visualize the results.

### 2.1.1 CTRL file

Let us start out by listing a minimal CTRL file for Cu.

```
HEADER  Cu fcc
        data by Landolt-Bernstein
VERSION ASW-2.0
STRUC   ALAT=6.83079 SLAT=FCC
CLASS   ATOM=CU Z=29
SITE    ATOM=CU POS= 0.0  0.0  0.0
```

Each CTRL file is grouped into categories, which begin with a keyword at the beginning of a line. In the listing, five categories are identified, namely HEADER, VERSION, STRUC, CLASS, and SITE. Each category comprises additional information, which, except for the first two categories, is passed to the program by socalled tokens. They have the form TOKEN=, followed by one or more integers, real numbers, strings or switches (T or F). To be specific, in the present example the category STRUC contains all information about the Bravais lattice in form of the lattice constant, ALAT=6.83079 and the Bravais lattice type, SLAT=FCC (face-centered cubic). While this fixes the unit cell of the crystal, information about the geometry inside the unit cell is contained in the category SITE, which gives the positions of all atoms. In the present case, only one atom exists, which is located at the origin. This atom is labelled by ATOM=CU. Finally, the type of the atom located at the origin has to be specified. This is done in category CLASS, where the atom at the origin via its label is assigned the atomic number 29. Note that the label CU could be replaced by anything else. However, the program assumes that labels will not exceed six characters.

Finally, the category HEADER holds space for up to 20 lines of text for user comments. However, note that the first line (the one containing the keyword HEADER) is exceptional as its entry by default is used as a title line for all plots. Last not least, category VERSION specfies the program version to be used.

While the above CTRL file already allows for a complete calculation the standard CTRL file looks like the following.

```
HEADER  Cu fcc
        data by Landolt-Bernstein
VERSION ASW-2.0
IO      HELP=F SHOW=T VERBOS=30 CLEAN=T
OPTIONS REL=T OVLCHK=T
STRUC   ALAT=6.83079 SLAT=FCC
CLASS   ATOM=CU Z=29 R/RA=2.66945 LMXL=2 CONF=4 4 3 4
                QVAL= 1.0  0.0  10.0  0.0
SITE    ATOM=CU POS= 0.0  0.0  0.0
SYMGRP
ENVEL   EKAP=-0.015
BZSMP   NKABC=6 0 0 BZINT=SMS EMIN=-1.0 EMAX=1.5 NDOS=1000
        NORD=5 WIDTH=0.02 EFTOL=1.0D-04 SAVDOS=F
CONTROL START= QUIT= FREE=F NITBND=99 CNVG=1.0D-08 CNVGET=1.0D-08
        NITATM=50 CNVGQA=1.0D-10
MIXING  NMIXB=5 BETAB=0.5 INCBB=T NMIXA=5 BETAA=0.5
SYMLIN  NPAN=5 NPTS=400 ORBWGT=F CARTE=F
```

```
        LABEL=W ENDPT= 0.500  0.250  0.750
        LABEL=L ENDPT= 0.500  0.500  0.500
        LABEL=g ENDPT= 0.000  0.000  0.000
        LABEL=X ENDPT= 0.500  0.000  0.500
        LABEL=W ENDPT= 0.500  0.250  0.750
        LABEL=K ENDPT= 0.375  0.375  0.750
PLOT    CARTV=T
        ORIGIN= 0.0  0.0  0.0
        RPLOT1= 1.0  0.0  0.0
        RPLOT2= 0.0  1.0  0.0
        RPLOT3= 0.0  0.0  1.0
        NPDIV1=50 NPDIV2=50
```

It contains a lot more categories and tokens, all of which are discussed in detail in Chap. 4. Here we mention, in particular, the switch to scalar-relativistic calculations, REL=T, the types of orbitals used in the calculations as specified by the maximum angular momentum, LMXL=2, the principal quantum numbers of all orbitals CONF=4 4 3 4, and the respective orbital occupations QVAL=.... In case you are in doubt about the input for these three tokens just leave them out. The program will add them for you. Moreover, we have specified the number of **k**-points to be used in the Brillouin zone integration by the token NKABC=6 0 0 (see Chap. 4). Finally, categories SYMLIN and PLOT, respectively, hold information about the lines within the first Brillouin zone, along which the band structure is plotted, and the region in real space used for plotting the crystal structure as well as the charge density and potential.

### 2.1.2 Execution of the main programs

Having generated the CTRL file the ASW program for self-consistent field calculations can be started by typing mnscf.run at the operating systems prompt. However, for Unix/Linux systems the distribution offers several useful shellscripts. Here we just type mnscf.x at the systems prompt, which starts a background job, with the output sent to file *outscf*. The program will use about 13 iterations to full convergence. On a standard PC the total execution time will be about 2-3 seconds. After completion the output file *outscf* contains all the information collected during the iterations. In addition, a file named CU (the naming is due to the token ATOM= in categories CLASS and SITE) has been created, which contains all information specific to the respective atom. These data are needed by the program. For this reason, you should not delete an atomic file or change it by hand. Otherwise the program might have to start from scratch again. There has been a quite complicated file checker implemented, which detects the status of all atomic files and after that decides, which calculation can be performed.

Once self-consistency has been achieved you are able to calculate the band structure by typing mnbnd.run or, better, mnbnd.x. This program uses the information contained in the files CTRL and CU and generates a new file called BNDE, which comprises the eigenvalues for each **k**-point, hence, the band structure. Output is stored in file *outbnd*, Calculation of the band structure will take about half a second.

In addition, after having changed SAVDOS=F to SAVDOS=T as well as QUIT=

to QUIT=BND and running mnscf.run again, you will obtain a file DOS, which holds all information about the partial densities of states for plotting. The same can be achieved by invoking the shellscript mndos.x, which does the neccessary changes in the CTRL file (and resets to the original state at the end) and writes output to file *outdos*.

The previous calculations can be made automatic by typing mnall.x. This shellscript performs a self-consistent calculation for NKABC=6 0 0 (this must be set in the CTRL file), then switches to NKABC=8 0 0, and then in several steps to NKABC=30 0 0. For each of these **k**-point grids a self-consistent calculation is performed. Note that this saves a lot of execution time as compared to starting from scratch with a **k**-point density of 30. This is due to the "restart" facility of the ASW program mnscf.run. Whenever the file checker in the program detects atomic files, which conform with the general settings of the CTRL file as e.g. the lattice type or the number and types of orbitals the program is able to use this information as a starting point for further calculations. In particular, once the calculations for NKABC=6 0 0 have converged the subsequent calculation for NKABC=8 0 0 starts out from the converged files rather than starting from scratch. As a result, only few iterations will be needed for full convergence at the higher **k**-space grid. Such a stepwise increase of the Brillouin zone density serves the additional purpose that on comparing the results we are able to check convergence of the result with respect to the fineness of the **k**-point grid. The results of all these calculations are stored in the files *outlst6* to *outlst30*. Finally, the band structure and the (partial) densities of states are calculated.

In order to have quick check of the progress of a self-consistent calculation just type e.g. susan outlst30, which will write the following lines to screen

```
ASW-1.9, program MNSCF started on majestix at Fri, 08 Mar 2002, 14:14:02.
Calculation converged after   5 iteration(s).
                Start of Iteration   5
  2480 irreducible k-points generated from   27000( 30, 30, 30).
           Fermi energy - MTZ =     0.641842 Ryd.
           DOS at Fermi energy:     3.084967 1/Ryd.
Mean-square residual:           0.184027D-14
    Madelung energy:       0.000000
      Zeeman energy:       0.000000
          total 3pV:      -0.170258
      virial energy:   -3304.885873
 variational energy:   -3304.885885
 qdiff =       0.00000000  <        0.00000001
 ediff =       0.00000000  <        0.00000001
ASW-1.9, program MNSCF ended on majestix at Fri, 08 Mar 2002, 14:14:14.
```

They include information about the **k**-space grid, the position of the Fermi energy, the DOS at $E_F$, the (variational) total energy, and the self-consistency level already reached as coded by the entries qdiff and ediff.

### 2.1.3 Execution of the plot programs

By now, you are in a position to invoke the plotting routines. Just type plbnd.run and enter the following dialog.

```
 ASW-1.9, program PLBND started on majestix at Fri, 08 Mar 2002, 18:02:21.

     Copyright (C) 1992-2002 Volker Eyert
     Please see file COPYRIGHT for details

 All input will be echoed to file PLIB.

 Enter terminal type:
 1 = X-Windows (default)
 2 = PC-Screen (vt220-emulation)
 3 = suppress terminal output
/
 Enter output device:
 1 = Postscript (default)
 2 = Color postscript
 3 = LaTeX
 4 = LaTeX (VE's way)
 5 = HP LaserJet III (PCL5)
 6 = HP LaserJet II
 7 = GIF
 8 = leave the decision for later
 9 = suppress output to file
/
 Enter title:
/
 Energies in Rydberg (f) or eV (t, default)?
/
 Energies relative to MTZ (0) or EFermi (F, default)?
/
 Portrait (P, default), landscape (L) or encapsulated postscript plot (E)?
/
 Energies connected by lines (t) (default f)?
/

 Please wait a moment: I'm reading the bands.
 Timing for   39 points out of  398:     0.00000 sec.

 Ebot =    -9.643162 eV , Etop =    28.337884 eV  relative to EF
 Emin =   -10.000000 eV , Emax =     6.000000 eV  relative to EF
/
 Enter new Emin, Emax to change these defaults:
 task  1, total   :  cpu time:     0.01000 sec

 ASW-1.9, program PLBND ended on majestix at Fri, 08 Mar 2002, 18:02:21.
```

Here / entered on the programs prompt will keep the default values proposed by the program. All the questions are self-explaining. After this dialog, new files have been created, as there are BND.GNU, BND1, and BOX. By typing gnuplot BND.GNU you will invoke gnuplot and obtain the plot on screen. In addition, a postscript file *bnd.ps* is created, also containing the plot. For Cu the result is shown

in Fig. 2.1. As we will discuss in more detail below, the band structure comprises

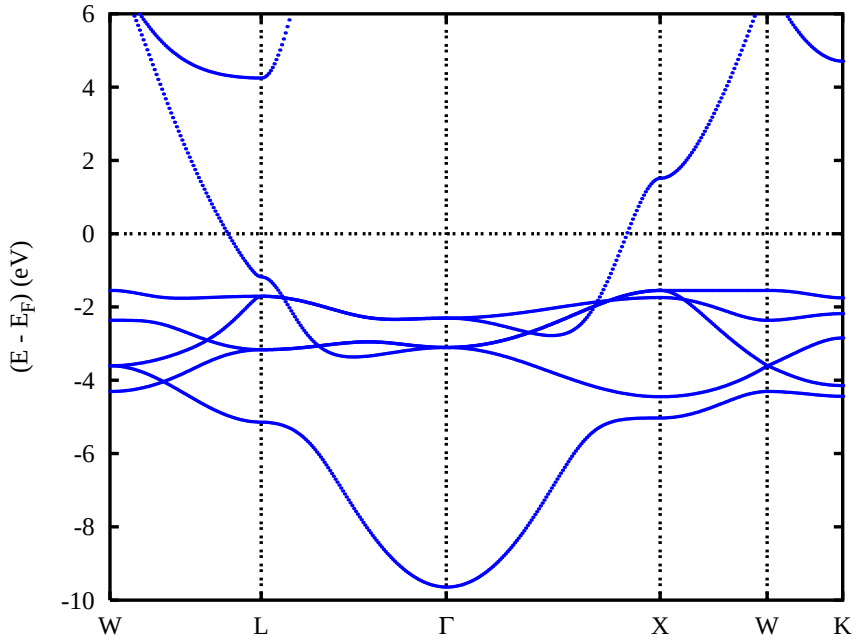



**Figure 2.1:** Electronic structure of Cu along selected symmetry lines of the first Brillouin zone of the face-centered cubic lattice, see A.1.

a parabolic band starting at the Γ-point at about -9.64 eV and ending near the W-point at 6 eV. This *s*-like band hybridizes with the five *d*-bands, which are visible in the energy interval from -5.5 to -1.5 eV. Note the degeneracies of the *d* bands especially along the lines Γ-L and Γ-X.

Still the procedure can be simplified by calling the shellscript plbnd.x rather than typing plbnd.run. This will run the plot program and invoke Gnuplot automatically. Moreover, the intermediate files BND.GNU, BND1 and BOX will be deleted by the shellscript.

However, user-friendlyness goes even beyond. During execution the plot routine plbnd.run has echoed all your input to a new file called PLIB. By inspecting this file you will find a fully commented echo of all your input. Rename this file to e.g. PLIBf, type plbnd.x PLIBf, and you obtain the same result as before without having to type anything else. This machinery could now be used to specify e.g. a different energy interval. Just edit file PLIBf and type plbnd.x PLIBf again. Yet, since file PLIB is just a simple echo, you should not change the order of the lines.

Next we are going to plot the partial densities of states. Now typing pldos.x you enter the following dialog.

```
ASW-1.9, program PLDOS started on majestix at Fri, 08 Mar 2002, 18:41:10.

    Copyright (C) 1992-2002 Volker Eyert
    Please see file COPYRIGHT for details

All input will be echoed to file PLID.

Enter terminal type:
```

```
 1 = X-Windows (default)
 2 = PC-Screen (vt220-emulation)
 3 = suppress terminal output
/
 Enter output device:
 1 = Postscript (default)
 2 = Color postscript
 3 = LaTeX
 4 = LaTeX (VE's way)
 5 = HP LaserJet III (PCL5)
 6 = HP LaserJet II
 7 = GIF
 8 = leave the decision for later
 9 = suppress output to file
2
 Enter title:
/
 Energies in Rydberg (f) or eV (t, default)? DOS scaled accordingly.
/
 Energies relative to MTZ (0) or EFermi (F, default)?
/
 Portrait (P, default), landscape (L) or encapsulated postscript plot (E)?
E
 Energy axis to the top (f) or right (t, default)?
/
 Plot DOS (f, default) or integrated DOS (t)?
/

 Please wait a moment: I'm reading the partial DOS.

 Start setting up curve 1:
 Plot partial (f) or total DOS (t, default)?
f
 For partial DOS: Rotate reference frame for orbitals?
 Enter rotation symbol (Default: E):
/

 Enter orbital(s) to be included:

 Class CU    : atom   1 at     0.000000    0.000000    0.000000
 Select from the following orbitals:
 0: all         -1: all p       -2: all d       -3: all f
 1: s            2: y            3: z            4: x            5: xy
 6: yz           7: 3z^2-r^2     8: xz           9: x^2-y^2     10: 3x^2y-y^3
11: xyz         12: 5yz^2-yr^2  13: 5z^3-3zr^2  14: 5xz^2-xr^2  15: x^2z-y^2z
16: x^3-3xy^2
1/

 The following orbitals have been selected:
 Class CU    : atom    1, orbital s

 Enter scaling factor (default 1.0):
/

 Broadening of this curve (t) or not (f, default)?
```

```
/

 Select curve style (default: 1) (A * marks styles already selected):
solid / green        (1)   , dashed1 / blue       (2)   ,
dashed2 / red        (3)   , dotted / magenta     (4)   ,
dashdotted1 / cyan   (5)   , dashdotted2 / yellow (6)   ,
chaindashed1 / black (7)   , chaindashed2 / coral (8)   ,
chaindashed3 / gray  (9)   ,
/

 Enter curve label ("/" to suppress)
Cu 4s

 Please wait a moment: I'm working on this curve.
 Set up more curves (t)? Default is f.
t

 Start setting up curve 2:
 Plot partial (f) or total DOS (t, default)?
f
 For partial DOS: Rotate reference frame for orbitals?
 Enter rotation symbol (Default: E):
/

 Enter orbital(s) to be included:

 Class CU    : atom   1 at     0.000000    0.000000    0.000000
 Select from the following orbitals:
 0: all         -1: all p       -2: all d       -3: all f
 1: s            2: y            3: z            4: x            5: xy
 6: yz           7: 3z^2-r^2     8: xz           9: x^2-y^2     10: 3x^2y-y^3
11: xyz         12: 5yz^2-yr^2  13: 5z^3-3zr^2  14: 5xz^2-xr^2  15: x^2z-y^2z
16: x^3-3xy^2
-1/

 The following orbitals have been selected:
 Class CU    : atom    1, orbital y
 Class CU    : atom    1, orbital z
 Class CU    : atom    1, orbital x

 Enter scaling factor (default 1.0):
/

 Broadening of this curve (t) or not (f, default)?
/

 Select curve style (default: 2) (A * marks styles already selected):
solid / green        (1) * , dashed1 / blue       (2)   ,
dashed2 / red        (3)   , dotted / magenta     (4)   ,
dashdotted1 / cyan   (5)   , dashdotted2 / yellow (6)   ,
chaindashed1 / black (7)   , chaindashed2 / coral (8)   ,
chaindashed3 / gray  (9)   ,
/

 Enter curve label ("/" to suppress)
```

```
Cu 4p

 Please wait a moment: I'm working on this curve.
 Set up more curves (t)? Default is f.
t

 Start setting up curve 3:
 Plot partial (f) or total DOS (t, default)?
/
 For partial DOS: Rotate reference frame for orbitals?
 Enter rotation symbol (Default: E):
/

 Enter orbital(s) to be included:

 Class CU    : atom   1 at     0.000000    0.000000    0.000000
 Select from the following orbitals:
 0: all         -1: all p       -2: all d       -3: all f
 1: s            2: y            3: z            4: x            5: xy
 6: yz           7: 3z^2-r^2     8: xz           9: x^2-y^2     10: 3x^2y-y^3
11: xyz         12: 5yz^2-yr^2  13: 5z^3-3zr^2  14: 5xz^2-xr^2  15: x^2z-y^2z
16: x^3-3xy^2
5
6
8/

 The following orbitals have been selected:
 Class CU    : atom    1, orbital xy
 Class CU    : atom    1, orbital yz
 Class CU    : atom    1, orbital xz

 Enter scaling factor (default 1.0):
/

 Broadening of this curve (t) or not (f, default)?
/

 Select curve style (default: 3) (A * marks styles already selected):
solid / green        (1) * , dashed1 / blue       (2) * ,
dashed2 / red        (3)   , dotted / magenta     (4)   ,
dashdotted1 / cyan   (5)   , dashdotted2 / yellow (6)   ,
chaindashed1 / black (7)   , chaindashed2 / coral (8)   ,
chaindashed3 / gray  (9)   ,
/

 Enter curve label ("/" to suppress)
Cu 3d t_{2g}

 Please wait a moment: I'm working on this curve.
 Set up more curves (t)? Default is f.
t

 Start setting up curve 4:
 Plot partial (f) or total DOS (t, default)?
/
```

```
 For partial DOS: Rotate reference frame for orbitals?
 Enter rotation symbol (Default: E):
/

 Enter orbital(s) to be included:

 Class CU    : atom   1 at     0.000000    0.000000    0.000000
 Select from the following orbitals:
 0: all         -1: all p       -2: all d       -3: all f
 1: s            2: y            3: z            4: x            5: xy
 6: yz           7: 3z^2-r^2     8: xz           9: x^2-y^2     10: 3x^2y-y^3
11: xyz         12: 5yz^2-yr^2  13: 5z^3-3zr^2  14: 5xz^2-xr^2  15: x^2z-y^2z
16: x^3-3xy^2
7
9/

 The following orbitals have been selected:
 Class CU    : atom    1, orbital 3z^2-r^2
 Class CU    : atom    1, orbital x^2-y^2

 Enter scaling factor (default 1.0):
/

 Broadening of this curve (t) or not (f, default)?
/

 Select curve style (default: 4) (A * marks styles already selected):
solid / green        (1) * , dashed1 / blue       (2) * ,
dashed2 / red        (3) * , dotted / magenta     (4)   ,
dashdotted1 / cyan   (5)   , dashdotted2 / yellow (6)   ,
chaindashed1 / black (7)   , chaindashed2 / coral (8)   ,
chaindashed3 / gray  (9)   ,
/

 Enter curve label ("/" to suppress)
Cu 3d e_g

 Please wait a moment: I'm working on this curve.
 Set up more curves (t)? Default is f.
/

 Center of gravity of DOS curve  1 (Spin 1) - EF :    0.8963 eV
 Center of gravity of DOS curve  2 (Spin 1) - EF :    2.3616 eV
 Center of gravity of DOS curve  3 (Spin 1) - EF :   -2.3706 eV
 Center of gravity of DOS curve  4 (Spin 1) - EF :   -2.6160 eV

 Indicate center of gravities (f, default)?
/

 Ebot =   -10.065942 eV , Etop =    11.669161 eV  relative to EF
 Emin =   -11.000000 eV , Emax =     6.000000 eV  relative to EF
 Enter new Emin, Emax to change these defaults:
-10
6
```

```
 DOStop =     3.772949 1/eV
 DOSmax =     4.000000 1/eV
 Enter new DOSmax to change this default:
/
 Enter energetic position of curve labels ("/" to use default):
/
 task  1, total   :  cpu time:     0.01000 sec

 ASW-1.9, program PLDOS ended on majestix at Fri, 08 Mar 2002, 18:41:10.
```

This dialog is much more complex than the one before, since we generated four curves and deviated from the default settings. The resulting plot is shown in Fig. 2.2, where we identify the partial DOS corresponding to the Cu $4s$, $4p$, and $3d$

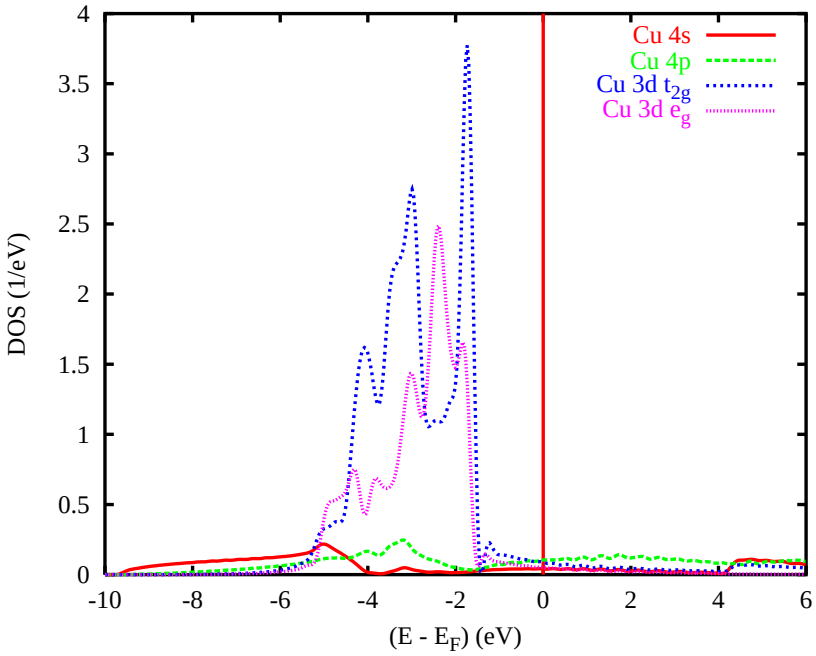



**Figure 2.2:**Partial densities of states of Cu.

orbitals. The latter have been split into their $t_{2g}$ and $e_g$ components as resulting from the cubic crystal field splitting. In particular, we observe rather small $4s$ and $4p$ contributions in the whole energy range. They grow out of the aforementioned parabolic-like band, which leads, in particular, to the square-root behaviour of the $4s$ partial DOS at low energies. At higher energies, the $3d$ states set in, which are limited to the energy interval between -5.5 and -1.5 eV but give large contributions.

As for the band structure an echo file has been created, which is called PLID. Rename this file to e.g. PLIDf and type pldos.x PLIDf in order to reproduce the plot. Of course, this plot routine could be also called without using the shell script, in which case we would have to type pldos.run < PLIDf at the systems prompt. Note the Unix/Linux < here, which is already included in the shellscript.

### 2.1.4 Advanced features

So far, we were able to decompose the total DOS into partial densities of states and were thus able to identify orbital contributions in certain energy intervals. However,

a corresponding analysis at a **k**-point level would be welcome. Such a tools exists in the ASW program package. In order to use it we enter the CTRL file and change in category SYMLIN the token NPTS=400 to NPTS=200 and ORBWGT=F to ORBWGT=T. It is suggested to first copy the files CTRL and CU to a separate directory and do the changes of the CTRL file there. After having changed the CTRL file as described type mnbnd.x. As before this will produce a file BNDE. In addition, a file named BNDV is generated, which contains all the eigenvectors.

Next you invoke again the plotting shellscript plbnd.x and enter the following dialog.

```
 ASW-1.9, program PLBND started on majestix at Fri, 08 Mar 2002, 21:26:15.

     Copyright (C) 1992-2002 Volker Eyert
     Please see file COPYRIGHT for details

 All input will be echoed to file PLIB.

 Enter terminal type:
 1 = X-Windows (default)
 2 = PC-Screen (vt220-emulation)
 3 = suppress terminal output
/
 Enter output device:
 1 = Postscript (default)
 2 = Color postscript
 3 = LaTeX
 4 = LaTeX (VE's way)
 5 = HP LaserJet III (PCL5)
 6 = HP LaserJet II
 7 = GIF
 8 = leave the decision for later
 9 = suppress output to file
4
 Enter title:
/
 Energies in Rydberg (f) or eV (t, default)?
/
 Energies relative to MTZ (0) or EFermi (F, default)?
/
 Portrait (P, default) or landscape plot (L)?
/
 Energies connected by lines (t) (default f)?
/
 Plot orbital character (default f)?
t
 For plotting orbital character: Rotate reference frame for orbitals?
 Enter rotation symbol (E for unity):
/

 Enter orbital(s) to be included:

 Class CU    : atom   1 at     0.000000    0.000000    0.000000
 Select from the following orbitals:
 0: all         -1: all p       -2: all d
```

```
 1: s            2: y            3: z            4: x            5: xy
 6: yz           7: 3z^2-r^2     8: xz           9: x^2-y^2
1

 The following orbitals have been selected:
 Class CU    : atom    1, orbital s

 Please wait a moment: I'm reading the bands.
 Timing for   19 points out of  199:     0.00000 sec.

 Ebot =    -9.643162 eV , Etop =    28.337884 eV  relative to EF
 Emin =   -10.000000 eV , Emax =     6.000000 eV  relative to EF
 Enter new Emin, Emax to change these defaults:
/
 Rescale orbital weights (default:   0.4000eV )?
/

 Please wait a moment: I'm working on the weights.
 Timing for   19 points out of  199:     0.00000 sec.

 Enter the horizontal and vertical extension in mm.
 Default: 100.0 x  90.0
/
 task  1, total   :  cpu time:     0.04000 sec

 ASW-1.9, program PLBND ended on majestix at Fri, 08 Mar 2002, 21:26:15.
```

It differs from the above dialog in asking for

```
Plot orbital character (default f)?
```

and the particular orbital(s) to be included. This section is similar to the plotting of the partial DOS above. The previous dialog will produce a file *bnd.tex*, which can be converted into *bnd.ps* by typing plbnd.lx at the system prompt. The result is shown in Fig. 2.3. In a similar way, orbital weighted band structures for the $4p$, $3dt_{2g}$, and $3de_g$ states can be obtained. They are shown in Figs. 2.4, 2.5, and 2.6. In all these figures we recognize the same band structure as already given in Fig. 2.1. Yet, in Figs. 2.3 to 2.6 to each band at each **k**-point has been appended a bar. The length of each bar is a measure of the contribution of a particular orbital to the respective wavefunction. To be specific, we identify large $4s$ and $4p$ contributions at the bottom as well as at the high energy branches of the parabola starting at -9.64 eV. In contrast the $3d$ orbitals dominate in the energy interval from -5.5 to -1.5 eV. These observations thus confirm our arguments put forward in the above discussion of the bare band structure and the partial densities of states. Actually, these orbital weighted band structures are a necessary prerequisite for a complete understanding of the electronic structure. Yet, only few band structure packages offer this analytical tool.

### 2.1.5 Crystal orbital overlap population

Another feature, which is not standard in modern band structure codes, is the implementation of the crystal orbital overlap populations (COOP). The COOP as

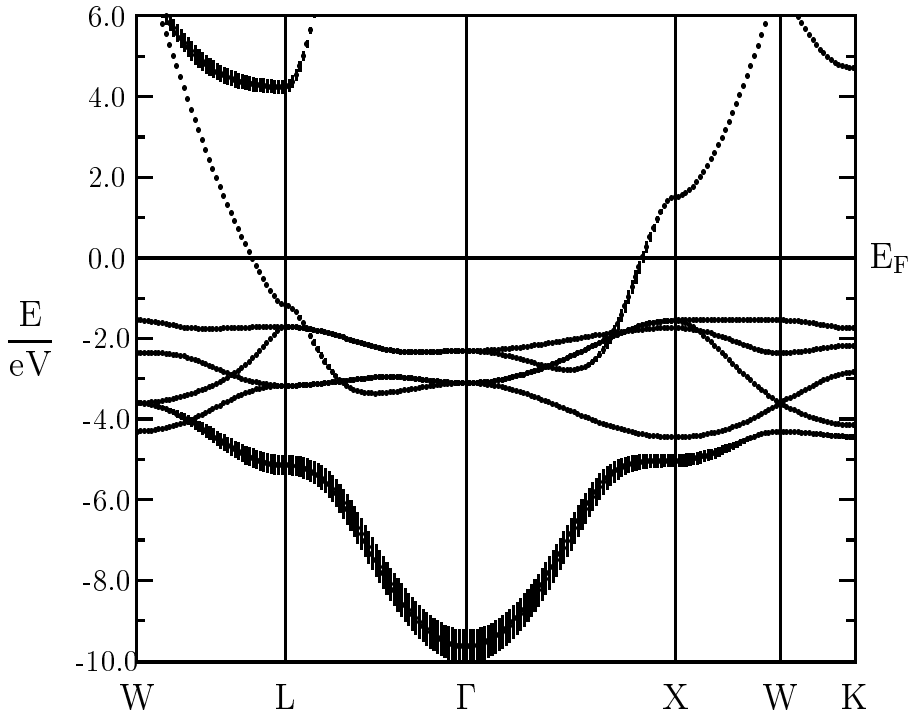



**Figure 2.3:** Weighted electronic structure of Cu. The width of the bars given for each band indicates the contribution due to the 4*s* orbital.

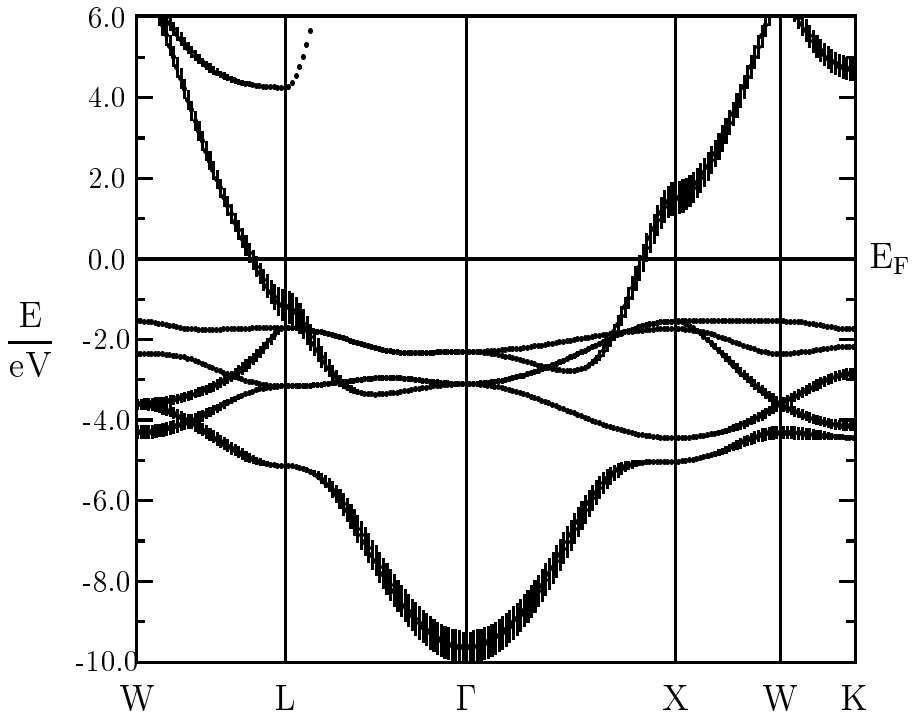



**Figure 2.4:** Weighted electronic structure of Cu. The width of the bars given for each band indicates the contribution due to the 4*p* orbitals.

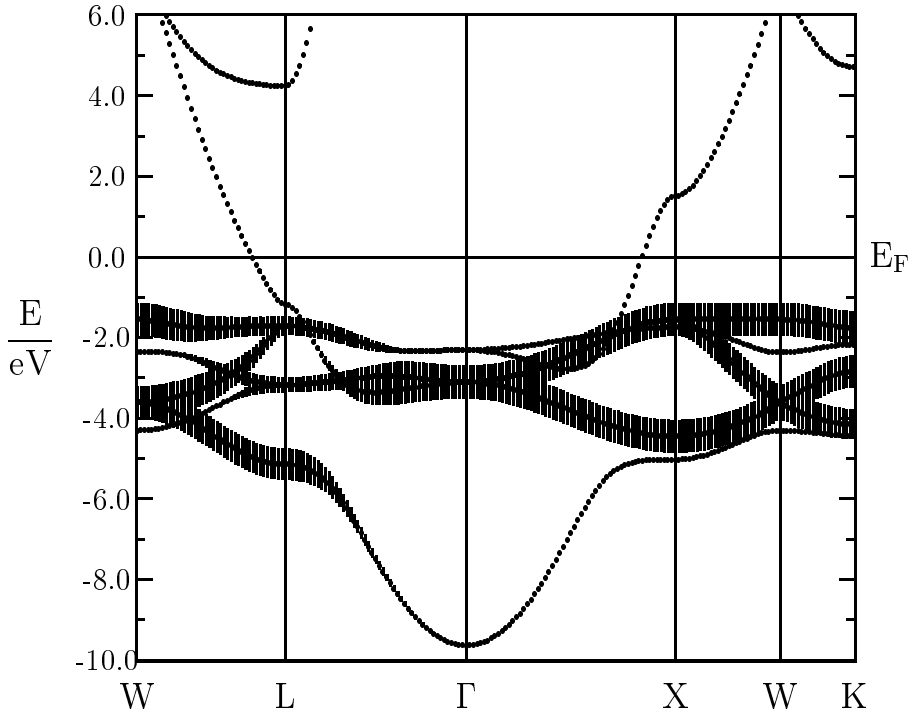



**Figure 2.5:** Weighted electronic structure of Cu. The width of the bars given for each band indicates the contribution due to the $3dt_{2g}$ orbitals.

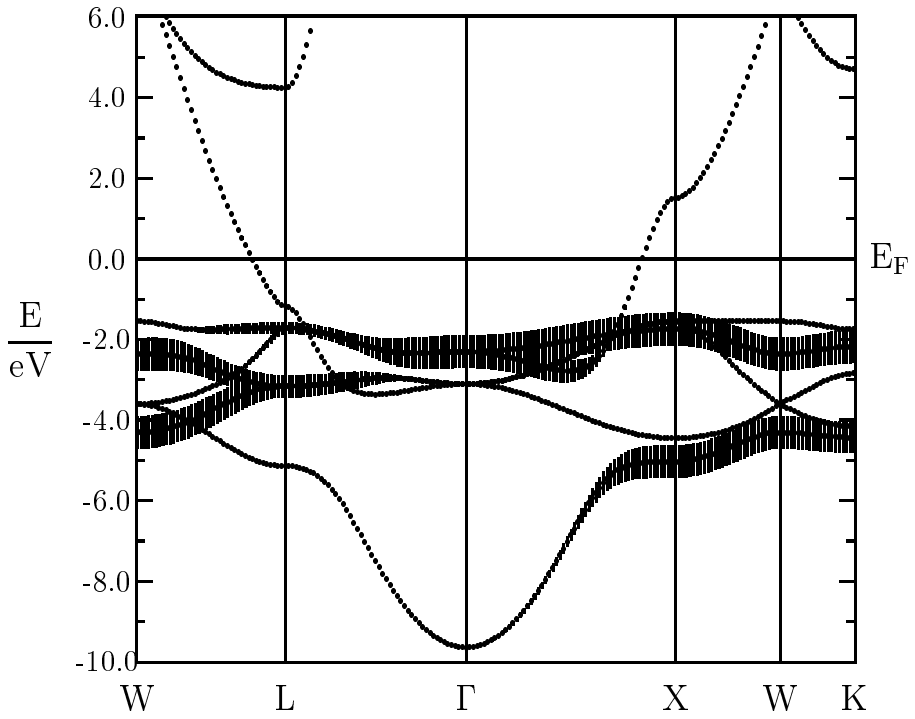



**Figure 2.6:** Weighted electronic structure of Cu. The width of the bars given for each band indicates the contribution due to the $3de_g$ orbital.

originally proposed by Hoffmann allows for an analysis of the chemical bonding [18]. Evaluation of the COOP has been recently implemented in the ASW method [19] (see also Refs. [3, 20]) and was successfully applied to the interpretation of bonding properties of various compounds [3]. Very recently, Fähnle and coworkers developed an extension of the COOP, the socalled covalence energy [21], which we have also implemented in the ASW package. The result for Cu is shown in Fig. 2.7. There

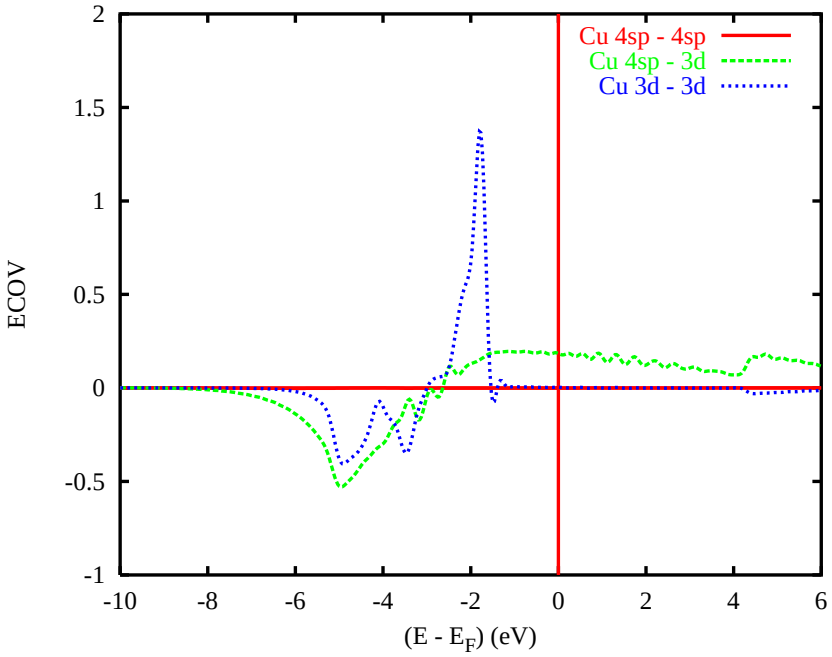



**Figure 2.7:**Partial covalence energies ($E_{cov}$) of Cu.

we observe three curves, which are energy resolved measures of the chemical bonding between the respective orbitals. Negative and positive contributions indicate bonding and antibonding states, respectively. While the 4*sp*–4*sp* bonding is almost negligible, the bonding between 4*sp* and 3*d* states as well as the 3*d*–3*d* bonding is clearly visible. Both are bonding below -2.2 eV and antibonding above. Since the 3*d*–3*d* antibonding behaviour dominates we can conclude that the metallic bonding, hence, crystal stability is carried to a large part by the 4*sp*–3*d* bonding.

Calculation of the crystal orbital overlap population or else, the covalence energy, is done in the same step as the calculation of the partial DOS. In order to obtain these curves, you have to specify the respective orbitals in the CTRL file. Just add in category CLASS the token COORB=, followed by the angular momenta of the orbitals, you want to be considered. In the present case we have added COORB=0 1 2. Furthermore, we inserted the entry SAVCOOP=F in category BZSMP. Calculation is then invoked by typing mndos.x at the systems prompt. Within this shellscript the setting SAVCOOP=F is automatically changed. Alternatively, you could specify the tokens COORB= at the very beginning before the self-consistent calculations start and then invoke the shellscript mnall.x, which likewise includes calculation of the COOP. It generates a new file named COOP. In order to plot the results just type plcop.x and answer the prompts of the program in close accordance with those required by the plotting routine for the DOS. However, note that for each

curve two orbitals are required.

### 2.1.6 Fermi surface

In closing this section we take up the discussion at the beginning and display in Fig. 2.8 the Fermi surface of Cu as measured by photoelectron spectroscopy and

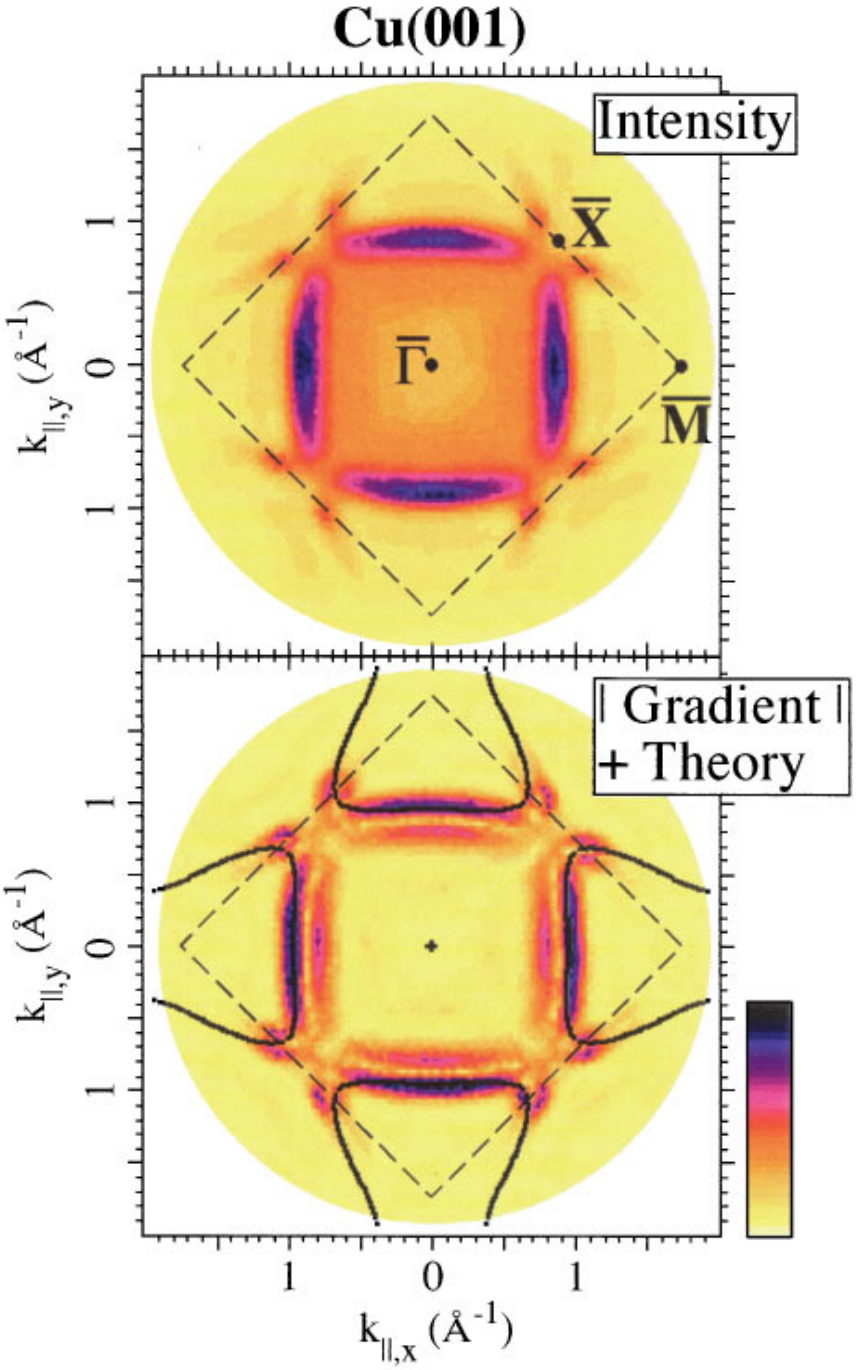



**Figure 2.8:**Measured and calculated Fermi surfaces of Cu [22].

calculated by the ASW method [22]. The agreement between both data is clearly

visible. Unfortunetely, the calculation of Fermi surfaces is not yet fully implemented in the code and thus not generally available.

## 2.2 A more complicated structure: $FeS_2$

While the calculations for Cu didn't even allow for a coffee break we next turn to a more complicated case with an increased demand of computer resources. To be specific, we turn to the case of iron pyrite, $FeS_2$, which is the prototype member of a whole class of transition metal disulfides. Background information about this compound is given in Ref. [23]. Due to the increased complexity of the crystal structure we will use additional programs coming with the ASW pacakge.

### 2.2.1 CTRL file and sphere packing

The pyrite structure is based on a simple cubic lattice with iron atoms located at the corner of the cell and sulfur pairs aligned along the spave diagonal. This information is contained in the following CTRL file.

```
HEADER  FeS2 sc
        data by E. D. Stevens, M. L. DeLucia, and P. Coppens,
        Inorg. Chem. 19, 813 (1980).
VERSION ASW-2.0
IO      HELP=F SHOW=T VERBOS=30 CLEAN=T
OPTIONS REL=T OVLCHK=T
STRUC   ALAT=10.23476 SLAT=SC
CLASS   ATOM=FE Z=26
        ATOM=S  Z=16
SITE    CARTP=T
        ATOM=FE POS= 0.000000  0.000000  0.000000
        ATOM=FE POS= 0.000000 -0.500000 -0.500000
        ATOM=FE POS=-0.500000  0.000000 -0.500000
        ATOM=FE POS=-0.500000 -0.500000  0.000000
        ATOM=S  POS= 0.384840  0.384840  0.384840
        ATOM=S  POS= 0.115160 -0.384840 -0.115160
        ATOM=S  POS=-0.384840 -0.384840 -0.384840
        ATOM=S  POS=-0.115160  0.384840  0.115160
        ATOM=S  POS=-0.115160  0.115160 -0.384840
        ATOM=S  POS=-0.384840 -0.115160  0.115160
        ATOM=S  POS= 0.115160 -0.115160  0.384840
        ATOM=S  POS= 0.384840  0.115160 -0.115160
SYMGRP  GENPOS=F SYMOPS=R2X:T( 0.5, 0.5, 0.0) R3D I
ENVEL   EKAP=-0.015
BZSMP   NKABC=6 0 0 BZINT=SMS EMIN=-1.0 EMAX=1.5 NDOS=1000
        NORD=3 WIDTH=0.01 EFTOL=1.0D-04 SAVDOS=F SAVCOOP=F
CONTROL START= QUIT= FREE=F NITBND=99 CNVG=1.0D-08 CNVGET=1.0D-08
        NITATM=50 CNVGQA=1.0D-10
MIXING  NMIXB=5 BETAB=0.5 INCBB=T NMIXA=5 BETAA=0.5
SYMLIN  NPAN=6 NPTS=400 ORBWGT=F CARTE=F
        LABEL=R ENDPT= 0.5  0.5  0.5
        LABEL=g ENDPT= 0.0  0.0  0.0
        LABEL=X ENDPT= 0.5  0.0  0.0
```

```
        LABEL=M ENDPT= 0.5  0.5  0.0
        LABEL=R ENDPT= 0.5  0.5  0.5
        LABEL=X ENDPT= 0.0  0.5  0.0
        LABEL=M ENDPT= 0.5  0.5  0.0
PLOT    CARTV=T
        ORIGIN= 0.5  0.0  0.0
        RPLOT1= 1.0  0.0  0.0
        RPLOT2= 0.0  1.0  0.0
        RPLOT3= 0.0  0.0  1.0
        NPDIV1=50 NPDIV2=50
```

Due to the symmetry of the crystal the simple cubic cell comprises four iron and eight sulfur atoms in total. The crystal structure is shown in Fig. 2.9. In this plot

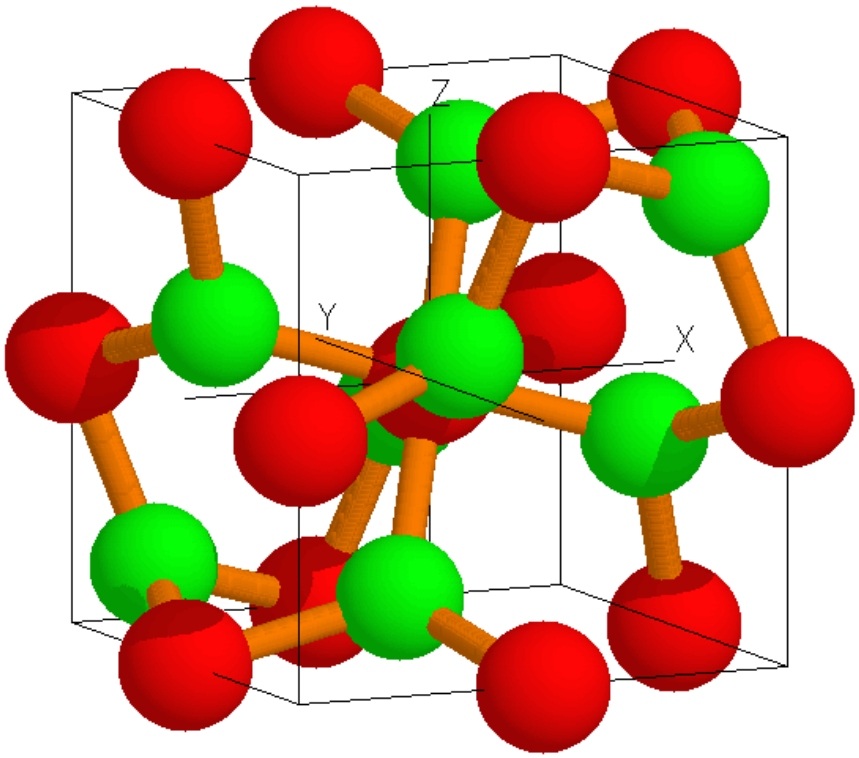



**Figure 2.9:** Crystal structure of $FeS_2$. Iron and sulfur atoms are printed in red and green, respectively.

the origin has been shifted by half the cell edge (see token ORIGIN=0.5 0.0 0.0 in category PLOT). As a consequence, one Fe atom is observed in the center of the plot while atoms of the sulfur pair appear in the upper right and lower left corner.

The pyrite crystal structure is best described in terms of the NaCl structure with the sublattices occupied by iron atoms and the centers of gravity of sulfur atom pairs, respectively. These sulfur dumb-bells are oriented along the $\langle 111 \rangle$ axes. Being 2.161 Å their bond length is still shorter than the Fe-S distance of 2.265 Å.

Whereas the sulfur atoms are tetrahedrally coordinated by one sulfur and three iron atoms the six nearest neighbour sulfur atoms at each iron site form slightly distorted octahedra. The distorted $FeS_6$ octahedra are interlinked by common corners and, due to the formation of the $\langle 111 \rangle$ sulfur pairs, have rotated away from the cartesian axes by about 23°. For a two dimensional crystal the situation is sketched in Fig. 2.10. Obviously, the formation of the $\langle 111 \rangle$ sulfur pairs does not destroy the square

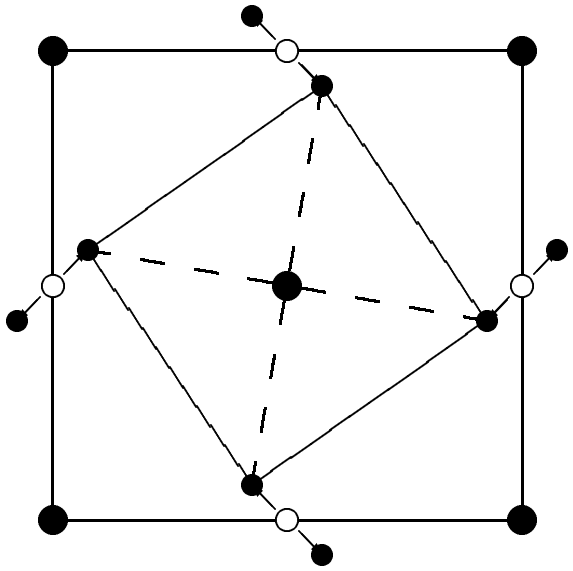


**Figure 2.10:** Two-dimensional analogue of the pyrite structure. Big and small filled circles designate iron and sulfur atoms, respectively. Small open circles mark the ideal positions of the rocksalt structure.

planar coordination of the iron atoms. Instead, the squares built by the sulfur atoms just shrink and rotate. Since the orientation of the dumb-bells conforms with the cubic point group the underlying Bravais lattice is no longer face-centered but simple cubic and the unit cell comprises four formula units.

In the CTRL file listed above we have already included information about the crystal symmetry in category SYMGRP. It contains two tokens, namely, SYMOPS= for a minimal set of symmetry operation, the socalled generators of the space group. By building all products of these generators the whole space group can be obtained. This information can be used in two different ways. First, for GENPOS=F the program performs a symmetry check using the information about the lattice and the atomic sites. As a result, all allowed symmetry operations are displayed. In addition, they are compared to the generators given in the CTRL file. This cross check could be likewise used to force the program not to use the full crystal symmetry but only a subgroup of the space group. E.g. if one specifies SYMOPS=E then only the identity operation will be used in the **k**-point loop, hence, the integration extends over the whole Brillouin zone rather than its irreducible wedge. Second, if GENPOS=T is specified, the program proceeds in a different way. In this case the symmetry check of the atomic sites is omitted. Instead, the program generates all atomic sites by applying the symmetry operations, which arise from the generators, to all atoms in the CTRL file. For this reason, we would not have to specifiy all atomic sites but only one iron and one sulfur atom. This is illustrated by the following listing.

```
HEADER  FeS2 sc
        data by E. D. Stevens, M. L. DeLucia, and P. Coppens,
        Inorg. Chem. 19, 813 (1980).
VERSION ASW-2.0
IO      HELP=F SHOW=T VERBOS=30 CLEAN=T
OPTIONS REL=T OVLCHK=T
STRUC   ALAT=10.23476 SLAT=SC
CLASS   ATOM=FE Z=26
        ATOM=S  Z=16
SITE    CARTP=T
        ATOM=FE POS= 0.000000  0.000000  0.000000
        ATOM=S  POS= 0.384840  0.384840  0.384840
SYMGRP  GENPOS=T SYMOPS=R2X:T( 0.5, 0.5, 0.0) R3D I
ENVEL   EKAP=-0.015
BZSMP   NKABC=6 0 0 BZINT=SMS EMIN=-1.0 EMAX=1.5 NDOS=1000
        NORD=3 WIDTH=0.01 EFTOL=1.0D-04 SAVDOS=F SAVCOOP=F
CONTROL START= QUIT= FREE=F NITBND=99 CNVG=1.0D-08 CNVGET=1.0D-08
        NITATM=50 CNVGQA=1.0D-10
MIXING  NMIXB=5 BETAB=0.5 INCBB=T NMIXA=5 BETAA=0.5
SYMLIN  NPAN=6 NPTS=400 ORBWGT=F CARTE=F
        LABEL=R ENDPT= 0.5  0.5  0.5
        LABEL=g ENDPT= 0.0  0.0  0.0
        LABEL=X ENDPT= 0.5  0.0  0.0
        LABEL=M ENDPT= 0.5  0.5  0.0
        LABEL=R ENDPT= 0.5  0.5  0.5
        LABEL=X ENDPT= 0.0  0.5  0.0
        LABEL=M ENDPT= 0.5  0.5  0.0
PLOT    CARTV=T
        ORIGIN= 0.5  0.0  0.0
        RPLOT1= 1.0  0.0  0.0
        RPLOT2= 0.0  1.0  0.0
        RPLOT3= 0.0  0.0  1.0
        NPDIV1=50 NPDIV2=50
```

This representation has an additional advantage. In the work on $FeS_2$ we had to check the influence of atomic displacements on the electronic properties [23]. In order to do so, the sulfur $x$ value was changed from $x = 0.38484$ to $x = 0.38084$ and 0.38884. With only one S atom given in the CTRL file and using the symmetry of the space group this is an easy task.

As is obvious from Fig. 2.9 the crystal structure comprises rather large voids between the atoms. However, since the standard ASW method is based on the atomic sphere approximation (ASA), where all space is completely filled by (necessarily overlapping) atomic spheres this would lead to too large overlap regions in the pyrite structure. We recall that the ASA as introduced by Andersen was meant as an approximation to the full crystal potential, which could be well modeled by slightly overlapping potential wells. The validity of this approach has been checked in a lot of cases. However, for open crystal structures the approximation becomes to crude and a trick has to be applied. It consists of inserting socalled empty spheres into the crystal structures, within which the electronic charge density and potential can vary and, hence, help modeling the total potential in a much improved way. Again, this has been checked with full potential calculations, which include the full crystal potential rather than employing the shape approximation coming with the

ASA. As a matter of fact, surprisingly good agreement has found.

For the present case, two additional programs come into play. First, type mnfre.run or mnfre.x. This program will calculate the charge density and potential arising from the superposition of these quantities as calculated for free atoms located at the respective crystallographic sites. As has been argued by Mattheiss the resulting potential resembles the final full crystal potential very much and, hence, can be well used to locate the position of possible empty spheres. The program generates a file FREE, which holds the charge density and potential for plotting along the line or within a plane defined by tokens ORIGIN=, RPLOT1=, and RPLOT2=, respectively. An example is shown in Fig. 2.11, where we identify the potential well

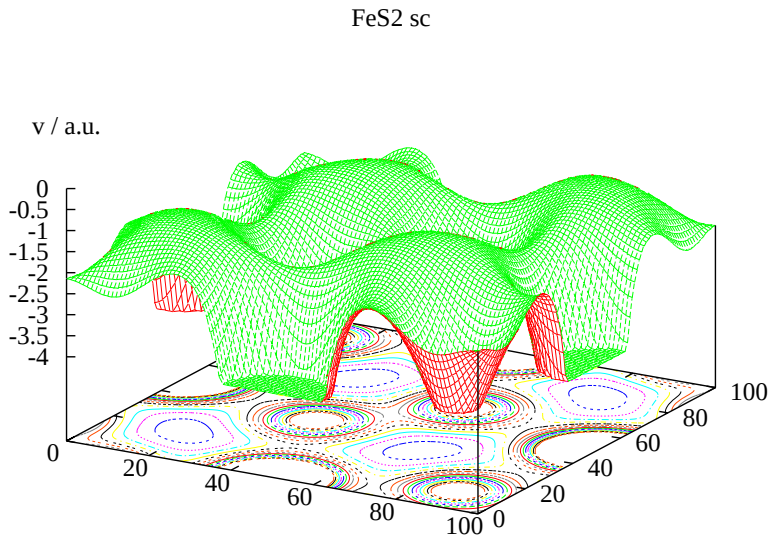



**Figure 2.11:**Overlapping free atom potential of fcc $FeS_2$.

of the sulfur pair in the middle of the plot. While program mnfre.run merely serves the purpose of plotting the aformentioned quantities, the actual search for possible empty sphere positions is done by a different program. Type mnpac.run or mnpac.x to invoke the socalled sphere geometry optimization (SGO) algorithm. This program proceeds in two different steps. First it scans the overlapping free atom potential along lines connecting the atoms and finds optimal muffin-tin radii, i.e. radii for non-overlapping spheres. These are taken as a guidline for the atomic sphere radii. Second, it starts a search through the unit cell and identifies that position, where the largest possible empty sphere could be placed. Having found such a empty sphere candidate, the radii of all spheres, including the "physical" spheres and all empty spheres found so far, are blown up until an acceptable overlap is reached. With the resulting atomic sphere radii the overall space filling is checked, and if this is not yet complete, the process is repeated by searching for addtional empty spheres until the ASA condition is obeyed.

The program mnpac.run generates two new files. File STRU contains a list of all atoms located within the space defined by tokens ORIGIN=, RPLOT1=,

RPLOT2=, and RPLOT3=. These can be used for plotting the crystal structure. To be specific, type plstr.x and enter the following dialog.

```
 ASW-2.0, program PLSTR started on majestix at Mon, 11 Mar 2002, 12:38:05.

     Copyright (C) 1992-2002 Volker Eyert
     Please see file COPYRIGHT for details

 All input will be echoed to file PLIS.

 Enter terminal type:
 1 = X-Windows (default)
 2 = PC-Screen (vt220-emulation)
 3 = suppress terminal output
/
 Enter output device (default: 1):
 1 = Color postscript
 2 = Postscript
 3 = GIF
/
 Enter title:
/
 Use unit cell (f, default) or Wigner-Seitz cell (t)?
/

 The following colours have been assigned to the classes:
  FE    : red         ,  S     : green       ,

 Select classes to be assigned a new colour
 (enter blank or "/" when you're done):
/

 By default the bonds will be coloured orange      .
 Enter blank or "/" to accept or select from the following colours:
  red          [r ],  green        [g ],  blue         [b ],  yellow       [y ]
  magenta      [m ],  cyan         [c ],  orange       [o ],  blueviolet   [bv]
  cyanogreen   [cg],  yellowgreen  [yg],  purple       [p ],  cyanoblue    [cb]
  violet       [v ],  pale red     [pr],  pale blue    [pb],  pale green   [pg]
  pale yellow  [py],  pale violet  [pv],  pale cyan    [pc],  black        [bk]
  gray1 (dark) [g1],  gray2        [g2],  gray3        [g3],  gray4        [g4]
  gray5        [g5],  gray6 (lght) [g6],  white        [wh]
/

 Muffin-tin spheres (M, default), ASA spheres (A), linear radii scaling (R)
 or linear volume scaling (V):
R

 Enter scale factor [  0.100...  1.342] (default:   1.000):
0.6

 The plot space contains    1 unit cell(s).

 Enter strings for rotations. Default strings are:
 rotate x  90
 rotate y  -5
```

```
 rotate x  10
/

 Overwrite or enter "/" when you're done.
 task  1, total       :  cpu time:     0.00000 sec

 ASW-2.0, program PLSTR ended on majestix at Mon, 11 Mar 2002, 12:38:05.
```

At the end the shellscript will automatically ask you for calling either RasMol or XMakemol and display a picture of the crystal structure similar to the one shown in Fig. 2.9. All your input to the plot routine plstr.run will be echoed to file PLIS.

In case you have already run through the SGO algorithm and just want to create a new plot of the crystal structure type mnstr.x. This shell script will likewise call program mnpac.run but, by changing token QUIT= to QUIT=STR, will stop execution after file STRU has been written.

In addition to generating this file, program mnpac.run will write a new version of the CTRL file to file CNEW, which contains all the empty sphere positions as well as atomic sphere radii of all atoms. Recall that the programs never write to file CTRL! Finally, mnpac.run proposes the number of partial waves to be included by the tokens LMXL= and CONF=. These two tokens have been set quite conservatively and might need an adjustment the subsequent self-consistent field calculation. We will come back to this point below. Of course, the corresponding charges as given by token QVAL= are all zero. The file CNEW looks like the following.

```
HEADER  FeS2 sc
        data by E. D. Stevens, M. L. DeLucia, and P. Coppens,
        Inorg. Chem. 19, 813 (1980).
VERSION ASW-2.0
IO      HELP=F SHOW=T VERBOS=30 CLEAN=T
OPTIONS REL=T OVLCHK=T
STRUC   ALAT=10.23476 SLAT=SC
CLASS   ATOM=FE Z=26 R/RA=2.33212 LMXL=2 CONF=4 4 3 4 COORB=2
                QVAL= 2.0  0.0  6.0  0.0
        ATOM=S  Z=16 R/RA=2.25163 LMXL=2 CONF=3 3 3 4 COORB=1
                QVAL= 2.0  4.0  0.0  0.0
        ATOM=E1 Z= 0 R/RA=1.38299 LMXL=0 CONF=1 2 3
                QVAL= 0.0  0.0  0.0
        ATOM=E2 Z= 0 R/RA=1.04353 LMXL=0 CONF=1 2 3
                QVAL= 0.0  0.0  0.0
        ATOM=E3 Z= 0 R/RA=0.77989 LMXL=0 CONF=1 2
                QVAL= 0.0  0.0
        ATOM=E4 Z= 0 R/RA=0.78798 LMXL=0 CONF=1 2
                QVAL= 0.0  0.0
SITE    CARTP=T
        ATOM=FE POS= 0.000000  0.000000  0.000000
        ATOM=FE POS= 0.000000 -0.500000 -0.500000
        ATOM=FE POS=-0.500000  0.000000 -0.500000
        ATOM=FE POS=-0.500000 -0.500000  0.000000
        ATOM=S  POS= 0.384840  0.384840  0.384840
        ATOM=S  POS= 0.115160 -0.384840 -0.115160
        ATOM=S  POS=-0.384840 -0.384840 -0.384840
```

```
ATOM=S  POS=-0.115160  0.384840  0.115160
ATOM=S  POS=-0.115160  0.115160 -0.384840
ATOM=S  POS=-0.384840 -0.115160  0.115160
ATOM=S  POS= 0.115160 -0.115160  0.384840
ATOM=S  POS= 0.384840  0.115160 -0.115160
ATOM=E1 POS= 0.210048  0.294775  0.110792
ATOM=E1 POS=-0.210048 -0.294775 -0.110792
ATOM=E1 POS=-0.289952  0.205225 -0.110792
ATOM=E1 POS= 0.289952 -0.205225  0.110792
ATOM=E1 POS=-0.210048 -0.205225  0.389208
ATOM=E1 POS= 0.210048  0.205225 -0.389208
ATOM=E1 POS=-0.110792 -0.289952  0.205225
ATOM=E1 POS= 0.110792  0.289952 -0.205225
ATOM=E1 POS=-0.205225  0.389208 -0.210048
ATOM=E1 POS= 0.205225 -0.389208  0.210048
ATOM=E1 POS= 0.205225 -0.110792 -0.289952
ATOM=E1 POS=-0.205225  0.110792  0.289952
ATOM=E1 POS=-0.389208  0.289952 -0.294775
ATOM=E1 POS= 0.389208 -0.289952  0.294775
ATOM=E1 POS= 0.289952 -0.294775 -0.389208
ATOM=E1 POS=-0.289952  0.294775  0.389208
ATOM=E1 POS=-0.294775 -0.389208  0.289952
ATOM=E1 POS= 0.294775  0.389208 -0.289952
ATOM=E1 POS= 0.389208 -0.210048 -0.205225
ATOM=E1 POS=-0.389208  0.210048  0.205225
ATOM=E1 POS= 0.110792  0.210048  0.294775
ATOM=E1 POS=-0.110792 -0.210048 -0.294775
ATOM=E1 POS= 0.294775  0.110792  0.210048
ATOM=E1 POS=-0.294775 -0.110792 -0.210048
ATOM=E2 POS= 0.403309  0.242766 -0.369247
ATOM=E2 POS=-0.403309 -0.242766  0.369247
ATOM=E2 POS=-0.096691  0.257234  0.369247
ATOM=E2 POS= 0.096691 -0.257234 -0.369247
ATOM=E2 POS=-0.403309 -0.257234 -0.130753
ATOM=E2 POS= 0.403309  0.257234  0.130753
ATOM=E2 POS= 0.369247 -0.096691  0.257234
ATOM=E2 POS=-0.369247  0.096691 -0.257234
ATOM=E2 POS=-0.257234 -0.130753 -0.403309
ATOM=E2 POS= 0.257234  0.130753  0.403309
ATOM=E2 POS= 0.257234  0.369247 -0.096691
ATOM=E2 POS=-0.257234 -0.369247  0.096691
ATOM=E2 POS= 0.130753  0.096691 -0.242766
ATOM=E2 POS=-0.130753 -0.096691  0.242766
ATOM=E2 POS= 0.096691 -0.242766  0.130753
ATOM=E2 POS=-0.096691  0.242766 -0.130753
ATOM=E2 POS=-0.242766  0.130753  0.096691
ATOM=E2 POS= 0.242766 -0.130753 -0.096691
ATOM=E2 POS=-0.130753 -0.403309 -0.257234
ATOM=E2 POS= 0.130753  0.403309  0.257234
ATOM=E2 POS=-0.369247  0.403309  0.242766
ATOM=E2 POS= 0.369247 -0.403309 -0.242766
ATOM=E2 POS= 0.242766 -0.369247  0.403309
ATOM=E2 POS=-0.242766  0.369247 -0.403309
ATOM=E3 POS=-0.217983  0.461787  0.344830
ATOM=E3 POS= 0.217983 -0.461787 -0.344830
```

```
        ATOM=E3 POS= 0.282017  0.038213 -0.344830
        ATOM=E3 POS=-0.282017 -0.038213  0.344830
        ATOM=E3 POS= 0.217983 -0.038213  0.155170
        ATOM=E3 POS=-0.217983  0.038213 -0.155170
        ATOM=E3 POS=-0.344830  0.282017  0.038213
        ATOM=E3 POS= 0.344830 -0.282017 -0.038213
        ATOM=E3 POS=-0.038213  0.155170  0.217983
        ATOM=E3 POS= 0.038213 -0.155170 -0.217983
        ATOM=E3 POS= 0.038213 -0.344830  0.282017
        ATOM=E3 POS=-0.038213  0.344830 -0.282017
        ATOM=E3 POS=-0.155170 -0.282017 -0.461787
        ATOM=E3 POS= 0.155170  0.282017  0.461787
        ATOM=E3 POS=-0.282017 -0.461787 -0.155170
        ATOM=E3 POS= 0.282017  0.461787  0.155170
        ATOM=E3 POS=-0.461787 -0.155170 -0.282017
        ATOM=E3 POS= 0.461787  0.155170  0.282017
        ATOM=E3 POS= 0.155170  0.217983 -0.038213
        ATOM=E3 POS=-0.155170 -0.217983  0.038213
        ATOM=E3 POS= 0.344830 -0.217983  0.461787
        ATOM=E3 POS=-0.344830  0.217983 -0.461787
        ATOM=E3 POS= 0.461787  0.344830 -0.217983
        ATOM=E3 POS=-0.461787 -0.344830  0.217983
        ATOM=E4 POS= 0.125076  0.366317 -0.027164
        ATOM=E4 POS=-0.125076 -0.366317  0.027164
        ATOM=E4 POS=-0.374924  0.133683  0.027164
        ATOM=E4 POS= 0.374924 -0.133683 -0.027164
        ATOM=E4 POS=-0.125076 -0.133683 -0.472836
        ATOM=E4 POS= 0.125076  0.133683  0.472836
        ATOM=E4 POS= 0.027164 -0.374924  0.133683
        ATOM=E4 POS=-0.027164  0.374924 -0.133683
        ATOM=E4 POS=-0.133683 -0.472836 -0.125076
        ATOM=E4 POS= 0.133683  0.472836  0.125076
        ATOM=E4 POS= 0.133683  0.027164 -0.374924
        ATOM=E4 POS=-0.133683 -0.027164  0.374924
        ATOM=E4 POS= 0.472836  0.374924 -0.366317
        ATOM=E4 POS=-0.472836 -0.374924  0.366317
        ATOM=E4 POS= 0.374924 -0.366317  0.472836
        ATOM=E4 POS=-0.374924  0.366317 -0.472836
        ATOM=E4 POS=-0.366317  0.472836  0.374924
        ATOM=E4 POS= 0.366317 -0.472836 -0.374924
        ATOM=E4 POS=-0.472836 -0.125076 -0.133683
        ATOM=E4 POS= 0.472836  0.125076  0.133683
        ATOM=E4 POS=-0.027164  0.125076  0.366317
        ATOM=E4 POS= 0.027164 -0.125076 -0.366317
        ATOM=E4 POS= 0.366317 -0.027164  0.125076
        ATOM=E4 POS=-0.366317  0.027164 -0.125076
SYMGRP  GENPOS=F SYMOPS=R2X:T( 0.5, 0.5, 0.0) R3D I

ENVEL   EKAP=-0.015
BZSMP   NKABC=6 0 0 BZINT=SMS EMIN=-1.0 EMAX=1.5 NDOS=1000
        NORD=3 WIDTH=0.01 EFTOL=1.0D-04 SAVDOS=F SAVCOOP=F
CONTROL START= QUIT= FREE=F NITBND=99 CNVG=1.0D-08 CNVGET=1.0D-08
        NITATM=50 CNVGQA=1.0D-10
MIXING  NMIXB=5 BETAB=0.5 INCBB=T NMIXA=5 BETAA=0.5
SYMLIN  NPAN=6 NPTS=400 ORBWGT=F CARTE=F
```

```
        LABEL=R ENDPT= 0.5  0.5  0.5
        LABEL=g ENDPT= 0.0  0.0  0.0
        LABEL=X ENDPT= 0.5  0.0  0.0
        LABEL=M ENDPT= 0.5  0.5  0.0
        LABEL=R ENDPT= 0.5  0.5  0.5
        LABEL=X ENDPT= 0.0  0.5  0.0
        LABEL=M ENDPT= 0.5  0.5  0.0
PLOT    CARTV=T
        ORIGIN= 0.5  0.0  0.0
        RPLOT1= 1.0  0.0  0.0
        RPLOT2= 0.0  1.0  0.0
        RPLOT3= 0.0  0.0  1.0
        NPDIV1=50 NPDIV2=50
```

For $FeS_2$, four types of empty spheres have been added with radii ranging from 0.78 to 1.38 Bohr radii. In total, 96 empty spheres have been inserted into the pyrite structure. Moreover, the program has specified optimal radii for iron and sulfur atoms.

All actions taken by the packing program mnpac.run are recorded in the output file *outpac*, which among other things contains the results of a final overlap check. There the overlap limits are repeated (as a default a linear overlap of 15% is used) and all lines with a linear overlap larger than 10% complemented by one or more exclamation marks. Note that an extra overlap bonus is given to the empty spheres due to the flatness of the potential within these spheres.

### 2.2.2 Execution of the main programs

Having run the packing program mnpac.run we are able to run the self-consistent field calculation just as in the case for Cu above. To do so, we copy file CNEW to CTRL and just type mnscf.x or else mnall.x depending on the amount of calculations to be performed. Already after a few iteration susan outlst6 will tell us that the program found a finite indirect band gap, which qualifies iron pyrite as an semiconductor. Now it is time to browse the input. Going to the end and searching backward for the string "Fermi" will put you on the top of a list containing for each class and partial wave (angular momentum) the amount of charge and the contribution to the density of states at $E_F$. Of course, for an insulator or semiconductor the second column contains only zeros. Special attention deserves the first column. If compared to the atomic configurations these partial occupations reflect the charge transfer between the orbitals. If summed over all orbitals of an atom they give rise to the deviation from the neutrality of an atom, which is given in an extra list above. For most atoms the occupation of the orbital with highest angular momentum is quite small, below 0.1 electron. This guarantess convergence of angular momentum expansions of the wave functions. In case the charge contained in the highest $l$ state is above 0.1 a warning is issued. Values up to 0.13 are still acceptable, otherwise value given by LMXL= must be increased. In contrast, if the numbers for the two highest $l$ orbitals and their sum is below 0.1 electrons, you might decrease the value set by LMXL= by one. This has been already done in the above listed CTRL file. Note that each orbital included (recall that there are $2l + 1$ states for each $l$) increases

the size of the secular matrix and thus leads to an increase of execution time and memory. CPU time scales roughly with the cube of the number of orbitals.

In case you realize during the first iterations that the number of orbitals for a particular atom should be changed you have to stop execution and delete all atomic files (those given by the entries ATOM= in category CLASS). In addition, you must delete the file MIX, which holds information about the progress of the iterations and is deleted automatically once self-consistency has been reached. After that adjust the angular momenta given by the tokens LMXL=. In case you have increased the LMXL= values corresponding entries have to be added to tokens CONF= and QVAL=. However, these latter tokens could likewise be completely erased since the program creates them automatically. Finally, the calculation can be restarted by typing mnscf.x or mnall.x at the systems prompt. For the latter the complete cycle of calculations will take several hours, enough for a coffee break.

### 2.2.3 Execution of the plot programs

After the self-consistent field calculations have converged and both the band structure and the (partial) DOS been calculated, we turn to the plotting. Typing plbnd.x and following the dialog we obtain the band structure displayed in Fig. 2.12. Five

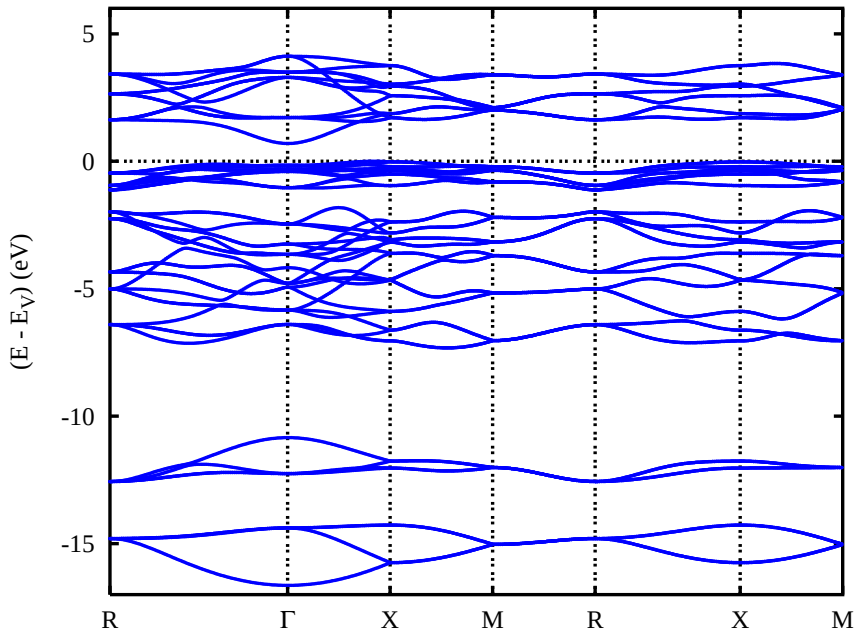



**Figure 2.12:** Electronic structure of $FeS_2$ along selected symmetry lines of the first Brillouin zone of the simple cubic lattice, see A.1.

groups of bands are observed. While two groups are found below -10 eV, a wide group shows up between approx. -7.5 and -2 eV. Just below and above the Fermi energy a rather narrow and a wide group follow. These latter two groups are separated by an optical band gap of $\approx 0.7$ eV. Note that all energies are referred to the valence band maximum.

The predominant orbital character of these five groups is read off from the partial densities of states as shown in Fig. 2.13. Obviously, the lowest two groups are made

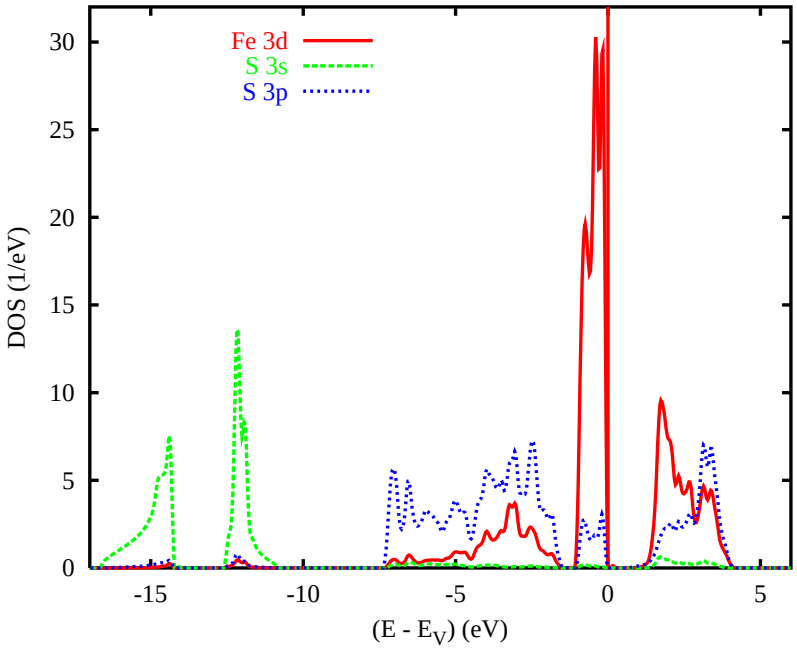



**Figure 2.13:**Partial densities of states of $FeS_2$.

exclusively from the semicore-like S 3*s* states, whereas the higher lying states trace back to a mixture of Fe 3*d* and S 3*p* bands. The latter orbitals dominate especially the wider groups between approx. -7.5 and -2 eV as well as between 2.7 and 4.5 eV. In contrast the sharp peak just below $E_V$ is due almost exclusively to the Fe 3*d* states. We will learn about a means for a more detailed analysis of these states below. For the time being we turn again to the orbital weighted band structures. The results shown in Fig. 2.14 clearly confirm the observation made in the partial

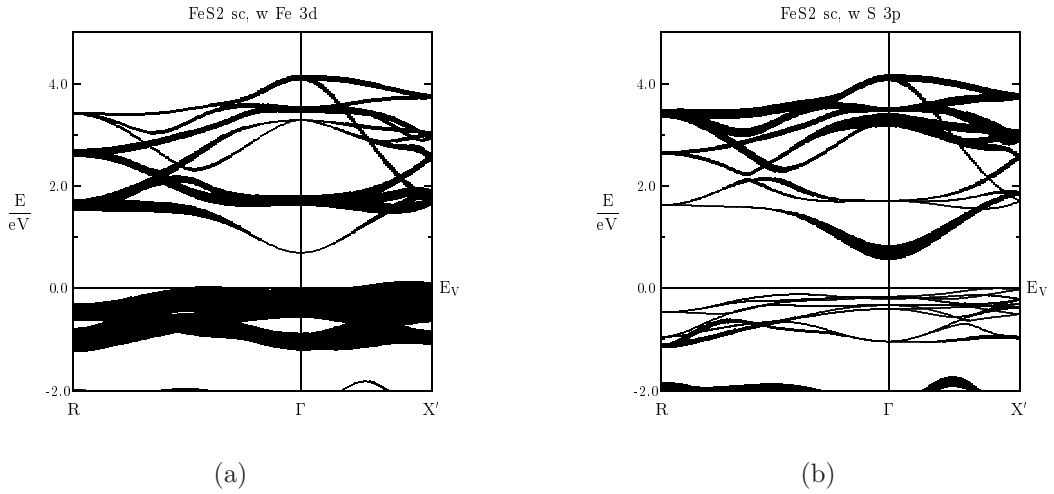



**Figure 2.14:** Weighted electronic bands of $FeS_2$. The width of the bars given for each band indicates the contribution due to the (a) Fe 3*d* and (b) S 3*p* orbitals, respectively.

DOS that the upper valence band as well as the lower half of the conduction band

group trace back to the Fe $3d$ states, while the upper half of the conduction band is more S $3p$ like. Yet, the lowest edge of the conduction band near the Γ-point is almost exclusively due to the S $3p$ orbitals and explains the sensitivity of the optical band gap to the bond length within the sulfur pair as found in Ref. [23].

## 2.3 A magnetic system: $CrO_2$

Apart from the elemental magnets Fe, Co, Ni, and Gd, chromium dioxide, $CrO_2$, probably belongs to the most wellknown ferromagnetic metals. Of course, this traces back to its applicability as a material for magnetic recording tapes, which is a consequence of the high Curie point of $T_C = 391K$ well above room temperature.

Additional interest in this compound stems from the fact that in general the transition metal dioxides comprise the largest class of oxides with a similar structure and showing a large variety of physical properties. As is obvious from Tab. 2.1 this diversity shows up mainly in the $3d$ series, which comprises the large-gap

**Table 2.1:** Properties of transition metal dioxides with rutile-related structure.

| | $d^0$ | $d^1$ | $d^2$ | $d^3$ | $d^4$ | $d^5$ | $d^6$ |
|---|---|---|---|---|---|---|---|
| $3d$ | $TiO_2$ (S) | $VO_2^*$ (M–S) | $CrO_2$ (F–M) | $MnO_2$ (AF–S) | | | |
| $4d$ | | $NbO_2^*$ (M–S) | $MoO_2^*$ (M) | $TcO_2^*$ (M) | $RuO_2$ (M) | $RhO_2$ (M) | |
| **5d** | | $TaO_2$ (?) | $WO_2^*$ (M) | $ReO_2^*$ (M) | $OsO_2$ (M) | $IrO_2$ (M) | $PtO_2^*$ (M) |

* deviations from rutile, M = metal, S = semiconductor
F/AF = ferro-/antiferromagnet

semiconductor $TiO_2$, the metal-insulator system $VO_2$, the half-metallic ferromagnet $CrO_2$, and the antiferromagnetic semiconductor $MnO_2$ [3]. In contrast, except for $NbO_2$, which, like $VO_2$ undergoes a metal-insulator transition accompanied by a structural transition, the $4d$ and $5d$ compounds are neither semiconducting nor magnetic. Nevertheless, there exist several members in each group, which display small but characteristic deviations from the rutile structure [4].

The electronic structure of $CrO_2$ has been under discussion for a long time. The first calculation performed by Schwarz using the ASW method revealed $CrO_2$ as a half-metallic ferromagnet, where spin-majority electrons carry metallic conductivity while the spin-minority bands show a gap at the Fermi energy [24]. Subsequent

calculations by Matar and coworkers confirmed these findings and revealed the instability of the compound towards antiferromagnetism on slight changes of the crystal structure as might be caused by the fabrication of thin films [25, 26]. Since then several workers have dealt with the material in order to explain the ferromagnetic order.

The rutile structure of $CrO_2$ is based on a simple tetragonal lattice with space group $P4_2/mnm$ ($D_{4h}^{14}$, No. 136) and lattice constants $a = 4.4219$Å, $c = 2.9166$Å [27]. The metal atoms are located at the Wyckoff positions (2a): $(0,0,0)$, $(\frac{1}{2},\frac{1}{2},\frac{1}{2})$ and the oxygen atoms occupy the positions (4f): $\pm(u,u,0), \pm(\frac{1}{2}+u,\frac{1}{2}-u,\frac{1}{2})$ with $u = 0.3024$. The rutile structure is displayed in Fig. 2.15. The structure can be

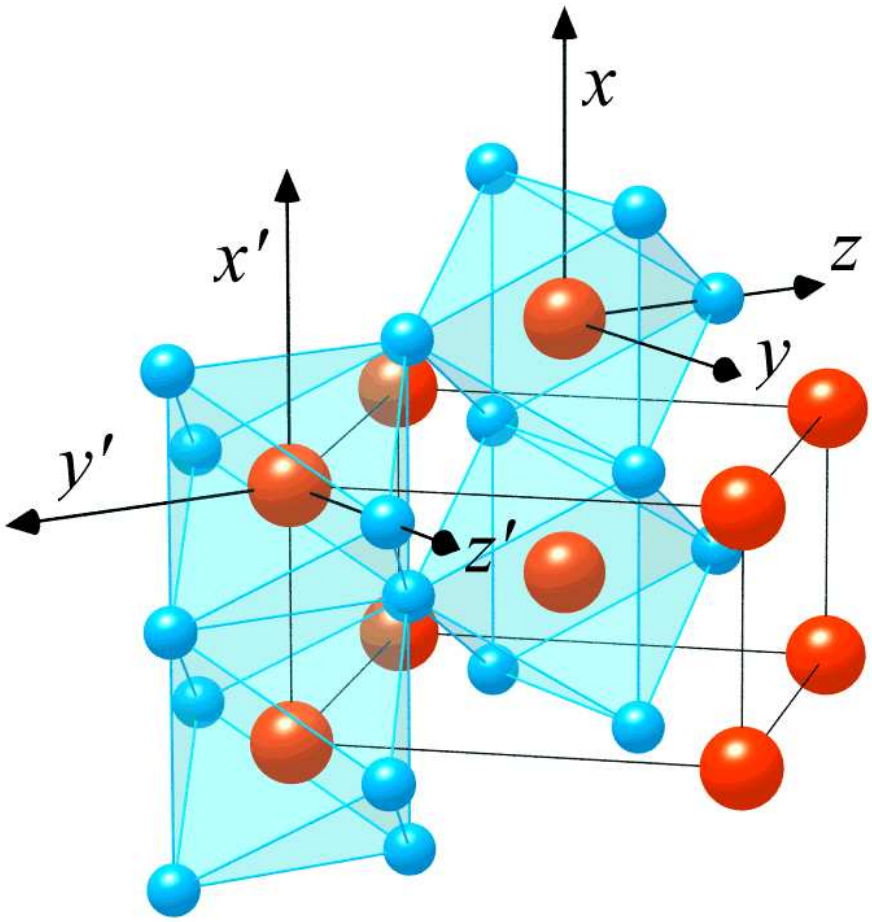



**Figure 2.15:** The rutile structure. Large and small spheres denote metal and ligand atoms, respectively.

alternatively visualized in terms of a body-centered tetragonal lattice formed by the metal atoms, where each metal atom is surrounded by an oxygen octahedron. Octahedra centered at the corners and the center of the cell are rotated by 90° about the tetragonal $c$ axis relative to each other. As a consequence, the lattice translational symmetry reduces to simple tetragonal and a unit cell with two formula units results. Octahedra, which are neighboured along the rutile $c$ axis, share edges whereas the resulting octahedral chains are interlinked via corners. Each octahedron

has orthorhombic symmetry although the deviations from tetragonal and even cubic geometry for most compounds are relatively small and still allow for a discussion in terms of the latter. There exist two different metal-oxygen distances, namely, the apical distance, which is between metal and oxygen atoms having the same $z$ value, and the equatorial distance between the metal atom and the four neighbouring ligand atoms with $z = z_{\mathrm{metal}} \pm 1/2$ [3].

As has been already indicated in Fig. 2.15 it is useful to discuss the electronic structure of rutile-type compounds in terms of local coordinate systems centered at each metal site. Note that due to the different orientation of octahedra centered at the corner and the center of the rutile cell, the local $z$ axes point alternately along the [110] and [1$\bar{1}$0] direction. In contrast to the usual adjustment of the $x$ and $y$ axes parallel to the metal-ligand bonds we have rotated these axes by 45° about the local $z$ axes such that they are parallel and perpendicular, respectively, to the rutile $c$ axis.

With the previous choice of local coordinate systems the $e_g$ states resulting from the cubic part of the crystal field splitting of the metal $d$ orbitals comprise the $d_{3z^2-r^2}$ and $d_{xy}$ orbitals whereas the $t_{2g}$ states are made of the $d_{x^2-y^2}$, $d_{xz}$, and $d_{yz}$ orbitals. While the $d_{x^2-y^2}$ orbitals point along the rutile $c$ and the local $y$ axes, i.e. towards the edges of the basal plane of the octahedron the $d_{xz}$ and $d_{yz}$ orbitals are directed towards the faces.

### 2.3.1 CTRL file and sphere packing

All the previous informations have been condensed into the following CTRL file.

```
HEADER  CrO2 rutile
        data by A. A. Bolzan, C. Fong, B. J. Kennedy, C. J. Howard,
        Acta Cryst. B53, 373 (1997).
VERSION ASW-1.9
IO      HELP=F SHOW=T VERBOS=30 CLEAN=T
OPTIONS REL=T OVLCHK=T
STRUC   ALAT=8.35618 SLAT=ST CBYA=0.65958
CLASS   ATOM=CR Z=24
        ATOM=O  Z= 8
SITE    CARTP=F
        ATOM=CR POS= 0.000000  0.000000  0.000000
        ATOM=CR POS= 0.500000  0.500000  0.500000
        ATOM=O  POS= 0.302400  0.302400  0.000000
        ATOM=O  POS=-0.302400 -0.302400  0.000000
        ATOM=O  POS=-0.197600  0.197600  0.500000
        ATOM=O  POS= 0.197600 -0.197600  0.500000
SYMGRP
ENVEL   EKAP=-0.015
BZSMP   NKABC=6 0 0 BZINT=SMS EMIN=-1.0 EMAX=1.5 NDOS=1000
        NORD=3 WIDTH=0.01 EFTOL=1.0D-04 SAVDOS=F SAVCOOP=F
CONTROL START= QUIT= FREE=F NITBND=99 CNVG=1.0D-08 CNVGET=1.0D-08
        NITATM=50 CNVGQA=1.0D-10
MIXING  NMIXB=5 BETAB=0.5 INCBB=T NMIXA=5 BETAA=0.5
SYMLIN  NPAN=10 NPTS=400 ORBWGT=F CARTE=F
        LABEL=g ENDPT= 0.0  0.0  0.0
        LABEL=X ENDPT= 0.0  0.5  0.0
```

```
        LABEL=R ENDPT= 0.0  0.5  0.5
        LABEL=Z ENDPT= 0.0  0.0  0.5
        LABEL=g ENDPT= 0.0  0.0  0.0
        LABEL=R ENDPT= 0.0  0.5  0.5
        LABEL=A ENDPT= 0.5  0.5  0.5
        LABEL=g ENDPT= 0.0  0.0  0.0
        LABEL=M ENDPT= 0.5  0.5  0.0
        LABEL=A ENDPT= 0.5  0.5  0.5
        LABEL=Z ENDPT= 0.0  0.0  0.5
PLOT    CARTV=F
        ORIGIN= 0.0  0.0  0.0
        RPLOT1= 1.0  0.0  0.0
        RPLOT2= 0.0  1.0  0.0
        RPLOT3= 0.0  0.0  1.0
        NPDIV1=50 NPDIV2=50
```

A few new tokens are of special interest here. Scoping with the tetragonal crystal structure we have inserted the token CBYA=, which contains the ratio $c/a$ of the tetragonal lattice constants. Alternatively, we could have inserted the lattice constant $c$ directly by the token CLAT=. In particular for non-cubic crystal structures it is useful to specify the atomic positions in terms of the primitive translations, which could be non-orthogonal or of unequal length, rather than the Cartesian coordinate system scaled by the lattice constant $a$.. The corresponding switch is covered by the token CARTP=F. As a consequence, the last component of the second chromium site as well as half of the oxygen sites is given by 0.5 rather than 0.32979 ($= \frac{c}{2a}$).

As for iron pyrite the openness of the rutile structure requires to insert empty spheres. Again this is done by executing mnpac.x. It produces file CNEW, which deviates from the above CTRL file by the categories CLASS and SITE updated as follows.

```
CLASS   ATOM=CR Z=24 R/RA=2.14485 LMXL=2 CONF=4 4 3 4
                QVAL= 1.0  0.0  5.0  0.0
        ATOM=O  Z= 8 R/RA=1.83785 LMXL=1 CONF=2 2 3
                QVAL= 2.0  4.0  0.0
        ATOM=E1 Z= 0 R/RA=1.65282 LMXL=1 CONF=1 2 3
                QVAL= 0.0  0.0  0.0
        ATOM=E2 Z= 0 R/RA=1.76299 LMXL=1 CONF=1 2 3
                QVAL= 0.0  0.0  0.0
        ATOM=E3 Z= 0 R/RA=0.77096 LMXL=0 CONF=1 2
                QVAL= 0.0  0.0
SITE    CARTP=F
        ATOM=CR POS= 0.000000  0.000000  0.000000
        ATOM=CR POS= 0.500000  0.500000  0.500000
        ATOM=O  POS= 0.302400  0.302400  0.000000
        ATOM=O  POS=-0.302400 -0.302400  0.000000
        ATOM=O  POS=-0.197600  0.197600  0.500000
        ATOM=O  POS= 0.197600 -0.197600  0.500000
        ATOM=E1 POS= 0.000000  0.500000 -0.250000
        ATOM=E1 POS= 0.000000  0.500000  0.250000
        ATOM=E1 POS=-0.500000  0.000000 -0.250000
        ATOM=E1 POS=-0.500000  0.000000  0.250000
        ATOM=E2 POS=-0.176474 -0.176474 -0.500000
        ATOM=E2 POS= 0.176474  0.176474 -0.500000
```

```
ATOM=E2 POS= 0.323526 -0.323526  0.000000
ATOM=E2 POS=-0.323526  0.323526  0.000000
ATOM=E3 POS= 0.218838  0.411732 -0.355939
ATOM=E3 POS=-0.218838 -0.411732  0.355939
ATOM=E3 POS=-0.281162  0.088268 -0.144061
ATOM=E3 POS= 0.281162 -0.088268  0.144061
ATOM=E3 POS=-0.411732 -0.218838  0.355939
ATOM=E3 POS= 0.411732  0.218838 -0.355939
ATOM=E3 POS= 0.281162 -0.088268 -0.144061
ATOM=E3 POS=-0.281162  0.088268  0.144061
ATOM=E3 POS= 0.411732  0.218838  0.355939
ATOM=E3 POS=-0.411732 -0.218838 -0.355939
ATOM=E3 POS=-0.218838 -0.411732 -0.355939
ATOM=E3 POS= 0.218838  0.411732  0.355939
ATOM=E3 POS= 0.088268 -0.281162  0.144061
ATOM=E3 POS=-0.088268  0.281162 -0.144061
ATOM=E3 POS=-0.088268  0.281162  0.144061
ATOM=E3 POS= 0.088268 -0.281162 -0.144061
```

To be specific, 24 empty spheres of three different types have been added with radii ranging from 0.78 to 1.76 $a_B$.

### 2.3.2 Spin-degenerate calculations

With the complete CTRL file at hand we are in a position to perform the self-consistent calculations for non-magnetic $CrO_2$ by typing mnall.x at the systems prompt. Although it is possible to start with the spin-polarized calculations right-away, it is recommended to do the calculations for the spin-degenerate case in advance. First, these calculations need half the execution time since the secular matrix has to be constructed and solved for only one spin direction. Owing to the restart facility of the ASW program mnscf.run the subsequent spin-polarized calculations can start away with the converged files of the spin-degenerate calculations rather then starting from scratch. Second, the result for the spin-degenerate case establish a reference, to which the results of the spin-polarized calculation can be compared. In particular, by checking the total energies we are able find out the relative stability of the ferromagnetic state.

After convergence has been achieved susan outlst30 summarizes the results as follows.

```
ASW-1.9, program MNSCF started on obelix at Sat, 09 Mar 2002, 16:10:19.
Calculation converged after   6 iteration(s).
               Start of Iteration   6
  1800 irreducible k-points generated from   27000( 30, 30, 30).
          Fermi energy - MTZ =     0.752253 Ryd.
          DOS at Fermi energy:   103.450981 1/Ryd.
Mean-square residual:           0.546051D-12
    Madelung energy:      -3.742272
      Zeeman energy:       0.000000
          total 3pV:      -0.706054
      virial energy:   -4794.027844
 variational energy:   -4794.027883
 qdiff =       0.00000000  <        0.00000001
```

```
  ediff =       0.00000001  <        0.00000001
 ASW-1.9, program MNSCF ended on obelix at Sat, 09 Mar 2002, 17:02:39.
```

In particular, non-magnetic $CrO_2$ turns out to be a metal with a rather high density of states at $E_F$. Within the framework of the Stoner theory of magnetism this is interpreted as a precursor of the ferromagnetic order of this material. We display in Fig. 2.16 the electronic states along selected high symmetry lines within the

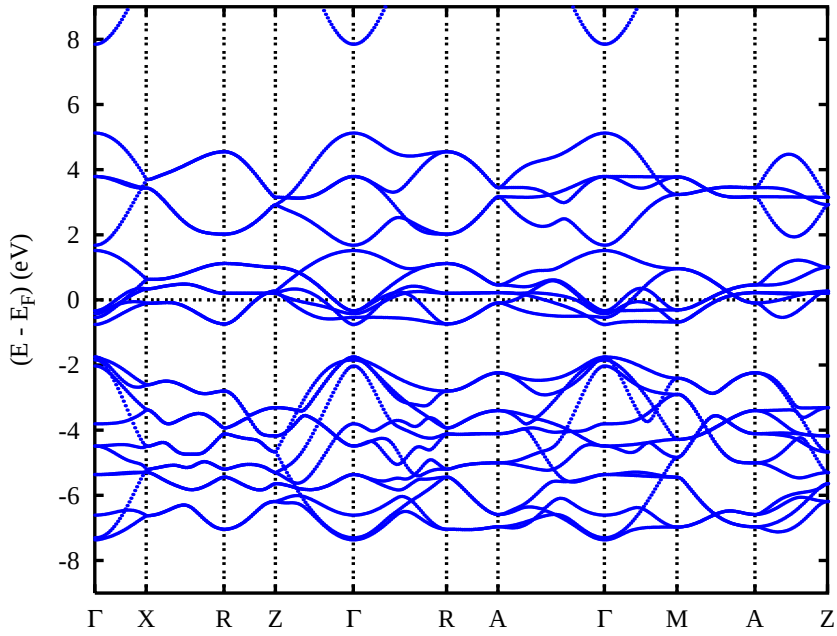



**Figure 2.16:** Electronic bands of $CrO_2$ along selected symmetry lines within the first Brillouin zone of the simple tetragonal lattice, Fig. A.5(a).

first Brillouin zone of the simple tetragonal lattice, Fig. A.5(a). The corresponding density of states (DOS) is given in Fig. 2.17. In Figs. 2.16 and 2.17 four groups of bands are identified. In the energy range from -7.6 to -1.8 eV we observe 12 bands, which trace back mainly to O $2p$ states but have a non-negligible contribution due to the Cr $3d$ states. Bands are most easily counted along the direction X-R where they are twofold degenerate. The next two groups, which extend from -0.8 to 1.7 eV and from 1.7 to 5.2 eV, contain six and four bands, respectively and originate mainly from Cr $3d$ states. Yet, $p-d$ hybridization causes additional O $2p$ contributions in this energy range. Finally, we observe Cr $4s$ states starting at 8.0 eV.

Crystal field splitting expected from the fact that the metal atoms are located at the centers of slightly distorted $CrO_6$ octahedra leads to almost perfect energetical separation of the $t_{2g}$ and $e_g$ manifolds of the Cr $3d$ states. Yet, as already mentioned above the standard axes of the local octahedra differ from the Cartesian coodinate system used to specify the primitive translations. For this reason, a transformation of the coordinate system has to be performed before the projected densities of states can be plotted. This is achieved by specifying the correct rotation in the following dialog.

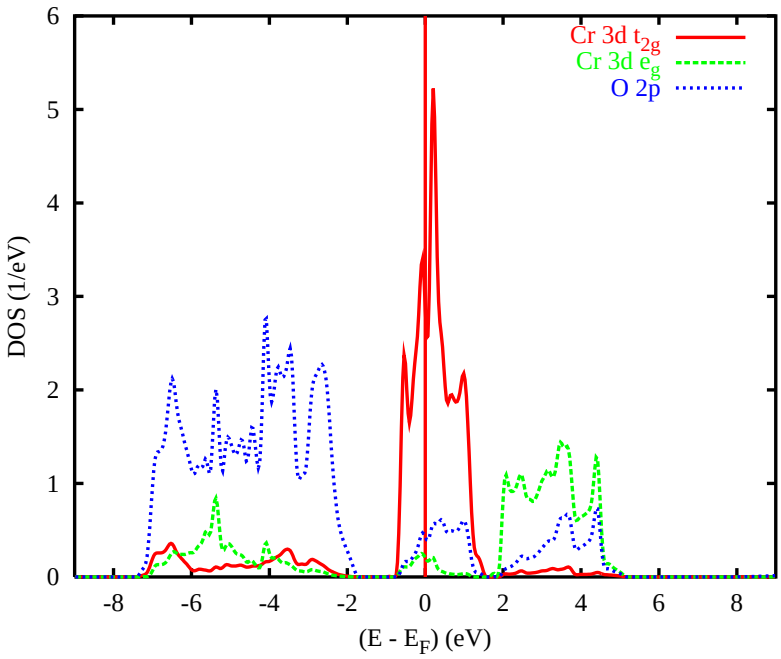



**Figure 2.17:**Partial densities of states (DOS) of $CrO_2$ per formula unit.

```
 ASW-1.9, program PLDOS started on majestix at Thu, 14 Mar 2002, 15:00:25.

     Copyright (C) 1992-2002 Volker Eyert
     Please see file COPYRIGHT for details

 All input will be echoed to file PLID.

 Enter terminal type:
 1 = X-Windows (default)
 2 = PC-Screen (vt220-emulation)
 3 = suppress terminal output
/
 Enter output device:
 1 = Postscript (default)
 2 = Color postscript
 3 = LaTeX
 4 = LaTeX (VE's way)
 5 = HP LaserJet III (PCL5)
 6 = HP LaserJet II
 7 = GIF
 8 = leave the decision for later
 9 = suppress output to file
2
 Enter title:

 Energies in Rydberg (f) or eV (t, default)? DOS scaled accordingly.
/
 Energies relative to MTZ (0) or EFermi (F, default)?
/
 Portrait (P, default), landscape (L) or encapsulated postscript plot (E)?
E
 Energy axis to the top (f) or right (t, default)?
```

```
/
 Plot DOS (f, default) or integrated DOS (t)?
/

 Please wait a moment: I'm reading the partial DOS.

 Start setting up curve 1:
 Plot partial (f) or total DOS (t, default)?
F
 For partial DOS: Rotate reference frame for orbitals?
 Enter rotation symbol (Default: E):
R8(-1,-1,0)*R4(1,-1,0)

 Enter orbital(s) to be included:
 Note: notation of the orbitals refers to the rotated reference frame!
 Select from the classes (blank or "/" to complete the list):
  CR      O       E1      E2      E3
CR

 Class CR    : atom   1 at     0.000000    0.000000    0.000000
 Select from the following orbitals:
 0: all         -1: all p       -2: all d       -3: all f
 1: s            2: y            3: z            4: x            5: xy
 6: yz           7: 3z^2-r^2     8: xz           9: x^2-y^2     10: 3x^2y-y^3
11: xyz         12: 5yz^2-yr^2  13: 5z^3-3zr^2  14: 5xz^2-xr^2  15: x^2z-y^2z
16: x^3-3xy^2
6
8
9/

 Class CR    : atom   2 at     0.500000    0.500000    0.329790
 Select from the following orbitals:
 0: all         -1: all p       -2: all d       -3: all f
 1: s            2: y            3: z            4: x            5: xy
 6: yz           7: 3z^2-r^2     8: xz           9: x^2-y^2     10: 3x^2y-y^3
11: xyz         12: 5yz^2-yr^2  13: 5z^3-3zr^2  14: 5xz^2-xr^2  15: x^2z-y^2z
16: x^3-3xy^2
/
 Select from the classes (blank or "/" to complete the list):
  CR      O       E1      E2      E3
/

 The following orbitals have been selected:
 Class CR    : atom    1, orbital yz
 Class CR    : atom    1, orbital xz
 Class CR    : atom    1, orbital x^2-y^2

 Enter scaling factor (default 1.0):
/

 Broadening of this curve (t) or not (f, default)?
/

 Select curve style (default: 1) (A * marks styles already selected):
solid / green        (1)   , dashed1 / blue       (2)   ,
```

```
dashed2 / red        (3)   , dotted / magenta     (4)   ,
dashdotted1 / cyan   (5)   , dashdotted2 / yellow (6)   ,
chaindashed1 / black (7)   , chaindashed2 / coral (8)   ,
chaindashed3 / gray  (9)   ,
/

 Enter curve label ("/" to suppress)
Cr 3d t_{2g}

 Please wait a moment: I'm working on this curve.
 Set up more curves (t)? Default is f.
T

 Start setting up curve 2:
 Plot partial (f) or total DOS (t, default)?
F
 For partial DOS: Rotate reference frame for orbitals?
 Enter rotation symbol (Default: R8(-1,-1,0)*R4(1,-1,0)):
/

 Enter orbital(s) to be included:
 Note: notation of the orbitals refers to the rotated reference frame!
 Select from the classes (blank or "/" to complete the list):
  CR      O       E1      E2      E3
CR

 Class CR    : atom   1 at     0.000000    0.000000    0.000000
 Select from the following orbitals:
 0: all         -1: all p       -2: all d       -3: all f
 1: s            2: y            3: z            4: x            5: xy
 6: yz           7: 3z^2-r^2     8: xz           9: x^2-y^2     10: 3x^2y-y^3
11: xyz         12: 5yz^2-yr^2  13: 5z^3-3zr^2  14: 5xz^2-xr^2  15: x^2z-y^2z
16: x^3-3xy^2
5
7/

 Class CR    : atom   2 at     0.500000    0.500000    0.329790
 Select from the following orbitals:
 0: all         -1: all p       -2: all d       -3: all f
 1: s            2: y            3: z            4: x            5: xy
 6: yz           7: 3z^2-r^2     8: xz           9: x^2-y^2     10: 3x^2y-y^3
11: xyz         12: 5yz^2-yr^2  13: 5z^3-3zr^2  14: 5xz^2-xr^2  15: x^2z-y^2z
16: x^3-3xy^2
/
 Select from the classes (blank or "/" to complete the list):
  CR      O       E1      E2      E3

 The following orbitals have been selected:
 Class CR    : atom    1, orbital xy
 Class CR    : atom    1, orbital 3z^2-r^2

 Enter scaling factor (default 1.0):
/

 Broadening of this curve (t) or not (f, default)?
```

```
/

 Select curve style (default: 2) (A * marks styles already selected):
solid / green        (1) * , dashed1 / blue       (2)   ,
dashed2 / red        (3)   , dotted / magenta     (4)   ,
dashdotted1 / cyan   (5)   , dashdotted2 / yellow (6)   ,
chaindashed1 / black (7)   , chaindashed2 / coral (8)   ,
chaindashed3 / gray  (9)   ,
/

 Enter curve label ("/" to suppress)
Cr 3d e_g

 Please wait a moment: I'm working on this curve.
 Set up more curves (t)? Default is f.
T

 Start setting up curve 3:
 Plot partial (f) or total DOS (t, default)?
F
 For partial DOS: Rotate reference frame for orbitals?
 Enter rotation symbol (Default: R8(-1,-1,0)*R4(1,-1,0)):
/

 Enter orbital(s) to be included:
 Note: notation of the orbitals refers to the rotated reference frame!
 Select from the classes (blank or "/" to complete the list):
  CR      O       E1      E2      E3
O

 Class O     : atom   3 at     0.302400    0.302400    0.000000
 Select from the following orbitals:
 0: all         -1: all p       -2: all d
 1: s            2: y            3: z            4: x            5: xy
 6: yz           7: 3z^2-r^2     8: xz           9: x^2-y^2
-1

 Class O     : atom   4 at    -0.302400   -0.302400    0.000000
 Select from the following orbitals:
 0: all         -1: all p       -2: all d
 1: s            2: y            3: z            4: x            5: xy
 6: yz           7: 3z^2-r^2     8: xz           9: x^2-y^2
-1

 Class O     : atom   5 at    -0.197600    0.197600    0.329790
 Select from the following orbitals:
 0: all         -1: all p       -2: all d
 1: s            2: y            3: z            4: x            5: xy
 6: yz           7: 3z^2-r^2     8: xz           9: x^2-y^2
/

 Class O     : atom   6 at     0.197600   -0.197600    0.329790
 Select from the following orbitals:
 0: all         -1: all p       -2: all d
 1: s            2: y            3: z            4: x            5: xy
```

```
 6: yz           7: 3z^2-r^2     8: xz           9: x^2-y^2
/
 Select from the classes (blank or "/" to complete the list):
  CR      O       E1      E2      E3
/

 The following orbitals have been selected:
 Class O     : atom    3, orbital y
 Class O     : atom    3, orbital z
 Class O     : atom    3, orbital x
 Class O     : atom    4, orbital y
 Class O     : atom    4, orbital z
 Class O     : atom    4, orbital x

 For this choice of orbitals rotation of the reference frame is ignored.

 Enter scaling factor (default 1.0):
/

 Broadening of this curve (t) or not (f, default)?
/

 Select curve style (default: 3) (A * marks styles already selected):
solid / green        (1) * , dashed1 / blue       (2) * ,
dashed2 / red        (3)   , dotted / magenta     (4)   ,
dashdotted1 / cyan   (5)   , dashdotted2 / yellow (6)   ,
chaindashed1 / black (7)   , chaindashed2 / coral (8)   ,
chaindashed3 / gray  (9)   ,
/

 Enter curve label ("/" to suppress)
O 2p

 Please wait a moment: I'm working on this curve.
 Set up more curves (t)? Default is f.
/

 Center of gravity of DOS curve  1 (Spin 1) - EF :   -0.3869 eV
 Center of gravity of DOS curve  2 (Spin 1) - EF :    0.4067 eV
 Center of gravity of DOS curve  3 (Spin 1) - EF :   -3.2636 eV

 Indicate center of gravities (f, default)?
/

 Ebot =   -19.827771 eV , Etop =    10.172793 eV  relative to EF
 Emin =   -20.000000 eV , Emax =     6.000000 eV  relative to EF
 Enter new Emin, Emax to change these defaults:
-9
9

 DOStop =     5.222992 1/eV
 DOSmax =     6.000000 1/eV
 Enter new DOSmax to change this default:
/
 Enter energetic position of curve labels ("/" to use default):
```

```
/
 task  1, total   :  cpu time:     0.02000 sec

 ASW-1.9, program PLDOS ended on majestix at Thu, 14 Mar 2002, 15:00:25.
```

Note that after the first input of the rotation matrix

```
R8(-1,-1,0)*R4(1,-1,0)
```

it is kept as default for the following curves. However, the program detects the selection of all $p$ orbitals for the oxygen partial DOS and notifies the user that the rotation can be ignored in this case.

The single components of the partial DOS making the $t_{2g}$ and $e_g$ manifolds are displayed in Fig. 2.18, where again we have included only the single chromium atom

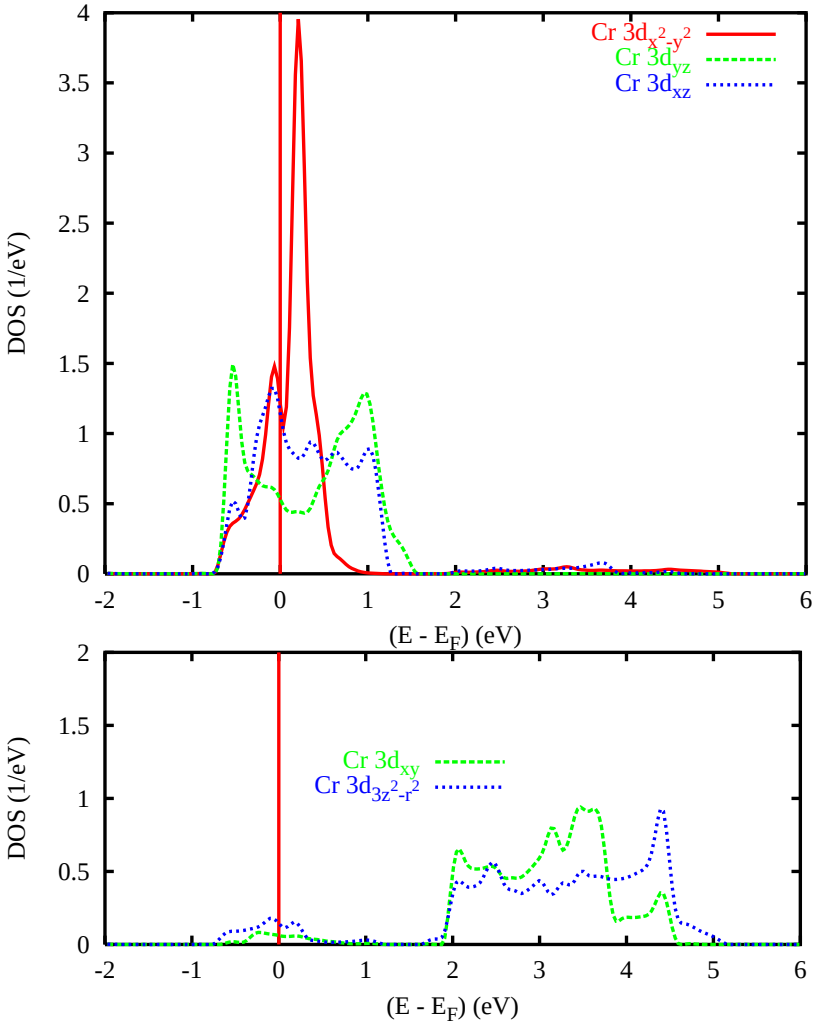



**Figure 2.18:** Partial Cr $3d$ $t_{2g}$ and $e_g$ densities of states (DOS) of $CrO_2$. Selection of orbitals is relative to the local rotated reference frame.

at $(0, 0, 0)$ and used the local rotated reference frame. As in Fig. 2.17 the almost perfect energetical separation of the $3d$ $t_{2g}$ and $e_g$ groups of bands is clearly revealed. The former states appear almost exclusively in the energy range from -0.8 to 1.7

eV whereas the $e_g$ states dominate the bands between 1.7 and 5.2 eV. The small but finite $t_{2g}$ – $e_g$ configuration mixing is a measure of octahedral distortions, i.e. the deviation from local cubic symmetry. Contributions of the Cr $3d$ states to the oxygen bands are slightly larger for the $e_g$ states which, forming $\sigma$ bonds, experience a larger overlap with the O $2p$ states. For the same reason, the bonding-antibonding splitting is larger for the $e_g$ states as compared to the $t_{2g}$ states, which give rise to $\pi$ bonds.

Chemical bonding is addressed via the COOP or the covalence energy. $E_{cov}$-curves growing out of the plot routine plcop.run are displayed in Fig. 2.19. Obviously,

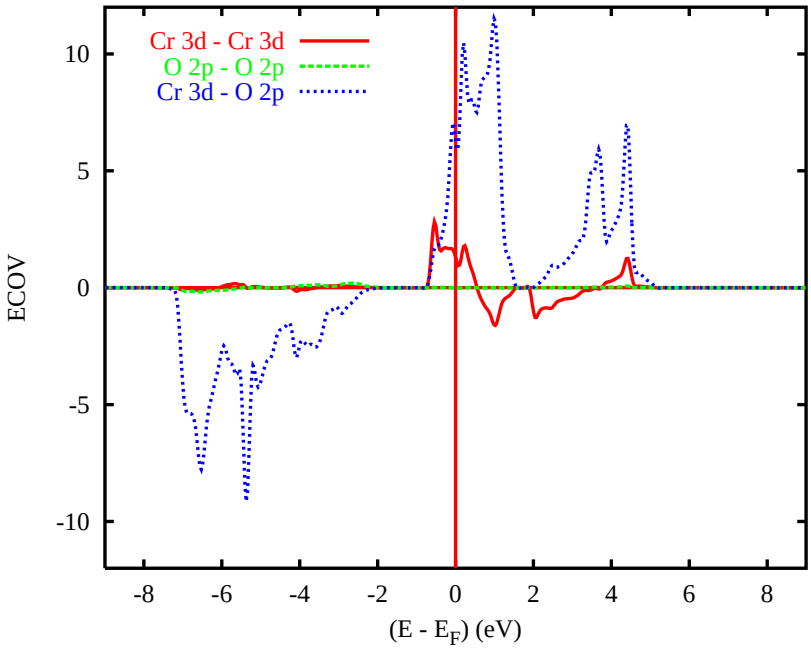



**Figure 2.19:**Partial covalence energies ($E_{cov}$) of $CrO_2$.

bonding within the O $2p$ states hardly contributes. In contrast, metal-ligand overlap is bonding and antibonding in the oxygen and chromium dominated electronic bands, this causing most of the stability of the compound. Finally, metal-metal bonding appears in the $t_{2g}$ and $e_g$ group of bands, being antibonding and bonding in the respective lower and upper part.

As has turned out in previous work on the neighbouring transition metal dioxides inspection of the weighted band structures is a necessary prerequisite for a deeper understanding of these materials. This holds especially for the chromium $3d$ $t_{2g}$ levels, which we address separately. For the $d_{x^2-y^2}$ orbital the result is shown in Fig. 2.20. It resulted from calling the plot program plbnd.x and entering the following dialog.

```
ASW-1.9, program PLBND started on majestix at Thu, 14 Mar 2002, 15:54:09.

    Copyright (C) 1992-2002 Volker Eyert
    Please see file COPYRIGHT for details

All input will be echoed to file PLIB.
```

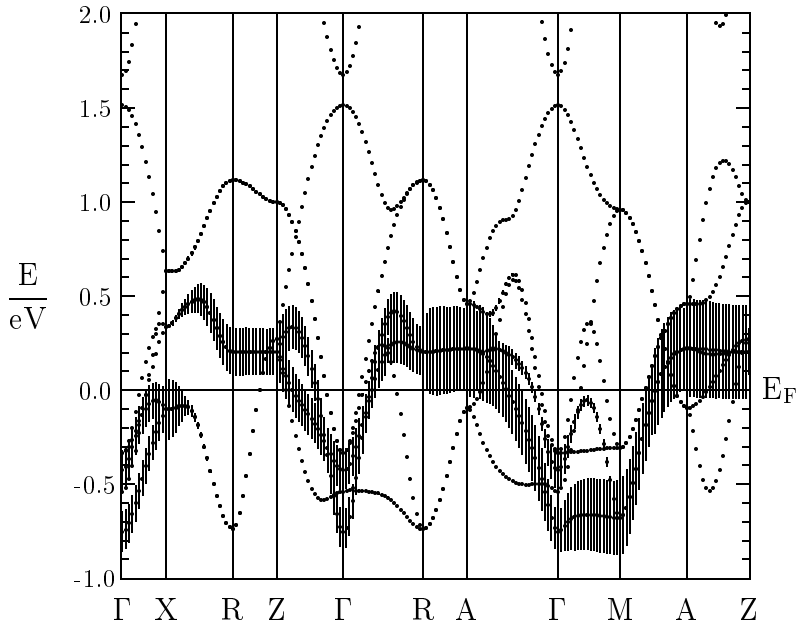



**Figure 2.20:** Weighted electronic bands of $CrO_2$. The width of the bars given for each band indicates the contribution due to the $3d_{x^2-y^2}$ orbital of the Cr atom at (0,0,0) relative to the local rotated reference frame.

```
 Enter terminal type:
 1 = X-Windows (default)
 2 = PC-Screen (vt220-emulation)
 3 = suppress terminal output
/
 Enter output device:
 1 = Postscript (default)
 2 = Color postscript
 3 = LaTeX
 4 = LaTeX (VE's way)
 5 = HP LaserJet III (PCL5)
 6 = HP LaserJet II
 7 = GIF
 8 = leave the decision for later
 9 = suppress output to file
4
 Enter title:

 Energies in Rydberg (f) or eV (t, default)?
/
 Energies relative to MTZ (0) or EFermi (F, default)?
/
 Portrait (P, default) or landscape plot (L)?
/
 Energies connected by lines (t) (default f)?
/
 Plot orbital character (default f)?
T
 For plotting orbital character: Rotate reference frame for orbitals?
```

```
 Enter rotation symbol (E for unity):
R8(-1,-1,0)*R4(1,-1,0)

 Enter orbital(s) to be included:
 Note: notation of the orbitals refers to the rotated reference frame!
 Select from the classes (blank or "/" to complete the list):
  CR      O       E1      E2      E3
CR

 Class CR    : atom   1 at     0.000000    0.000000    0.000000
 Select from the following orbitals:
 0: all         -1: all p       -2: all d
 1: s            2: y            3: z            4: x            5: xy
 6: yz           7: 3z^2-r^2     8: xz           9: x^2-y^2
9

 Class CR    : atom   2 at     0.500000    0.500000    0.329790
 Select from the following orbitals:
 0: all         -1: all p       -2: all d
 1: s            2: y            3: z            4: x            5: xy
 6: yz           7: 3z^2-r^2     8: xz           9: x^2-y^2
/
 Select from the classes (blank or "/" to complete the list):
  CR      O       E1      E2      E3
/

 The following orbitals have been selected:
 Class CR    : atom    1, orbital x^2-y^2

 Please wait a moment: I'm reading the bands.
 Timing for   19 points out of  194:     0.00000 sec.

 Ebot =   -19.922898 eV , Etop =    92.602619 eV  relative to EF
 Emin =   -20.000000 eV , Emax =     6.000000 eV  relative to EF
 Enter new Emin, Emax to change these defaults:
-1
2
 Rescale orbital weights (default:   0.0750eV )?
0.3

 Please wait a moment: I'm working on the weights.
 Timing for   19 points out of  194:     0.01000 sec.

 Enter the horizontal and vertical extension in mm.
 Default: 100.0 x  90.0
/
 task  1, total   :  cpu time:     0.16000 sec

 ASW-1.9, program PLBND ended on majestix at Thu, 14 Mar 2002, 15:54:09.
```

As in Sec. 2.1.4 the orbital to be included has to be specified. Here, the input "9" selects the $d_{x^2-y^2}$ orbital. However, before the rotation matrix has to be given in the same form as for program pldos.run above. Since LaTeXoutput was requested the plot routine plbnd.run writes to the file *bnd.tex*, from which the postscript file *bnd.ps* is created by invoking the shellscript plbnd.lx. The corresponding results for

the $d_{xz}$ and $d_{yz}$ orbitals are shown in Figs. 2.21 and 2.22. The following features

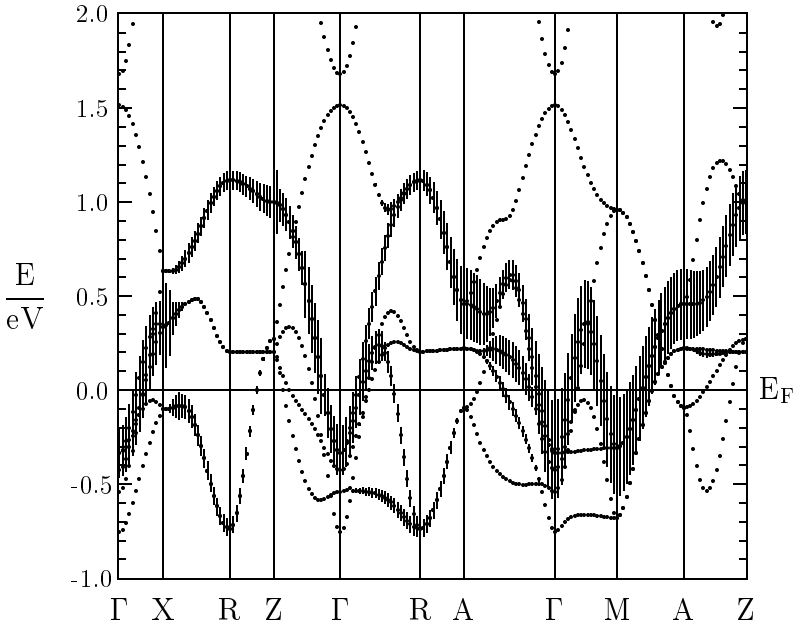



**Figure 2.21:** Weighted electronic bands of $CrO_2$. The width of the bars given for each band indicates the contribution due to the $3d_{xz}$ orbital of the Cr atom at (0,0,0) relative to the local rotated reference frame.

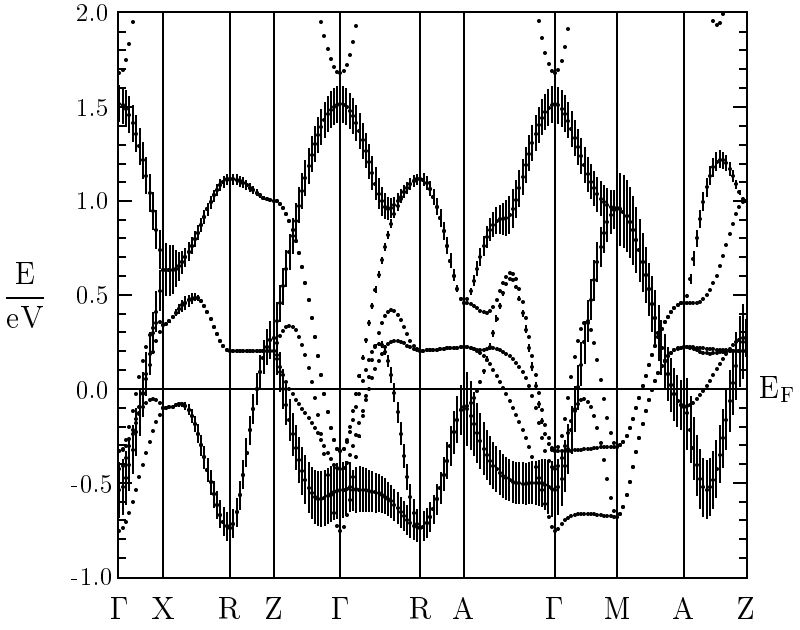



**Figure 2.22:** Weighted electronic bands of $CrO_2$. The width of the bars given for each band indicates the contribution due to the $3d_{yz}$ orbital of the Cr atom at (0,0,0) relative to the local rotated reference frame.

are worth mentioning. On relating Fig. 2.20 to the Brillouin zone of the simple tetragonal lattice, Fig. A.5(a) we observe negligible dispersions along all lines within

the planes spanned by the either the points Γ, X, and M, or else the points Z, R, and A. In contrast, dispersion along all lines connecting these plains is about 0.8 eV. Thus one finds a predominantly one-dimensional behaviour. This is different for the bands highlighted in Figs. 2.21 and 2.22, which display dispersions of a similar size along all lines.

### 2.3.3 Spin-polarized ferromagnetic calculations

Before adapting the CTRL file for the spin-polarized calculations you should copy it to another directory in order to preserve the results of the non-magnetic calculations. In order to perform spin-polarized calculations two new tokens have to be added to the new CTRL file. In category OPTIONS the entry NSPIN=2 must be inserted, which tells the program to perform a spin-polarized calculation. Furthermore, each orbital has to be given a starting value for the polarization. Otherwise the calculation would again converge to the spin-degenerate solution. Breaking the spin-symmetry is achieved by adding the token MVAL= for each atomic class in category CLASS followed by the polarization for each orbital. For ferromagnetic $CrO_2$ the new categroy CLASS looks like the following.

```
CLASS   ATOM=CR Z=24 R/RA=2.14485 LMXL=2 CONF=4 4 3 4
                QVAL= 1.0  0.0  5.0  0.0   MVAL= 0.0  0.0  2.0  0.0
        ATOM=O  Z= 8 R/RA=1.83785 LMXL=1 CONF=2 2 3
                QVAL= 2.0  4.0  0.0        MVAL= 0.0  0.0  0.0
        ATOM=E1 Z= 0 R/RA=1.65282 LMXL=1 CONF=1 2 3
                QVAL= 0.0  0.0  0.0        MVAL= 0.0  0.0  0.0
        ATOM=E2 Z= 0 R/RA=1.76299 LMXL=1 CONF=1 2 3
                QVAL= 0.0  0.0  0.0        MVAL= 0.0  0.0  0.0
        ATOM=E3 Z= 0 R/RA=0.77096 LMXL=0 CONF=1 2
                QVAL= 0.0  0.0             MVAL= 0.0  0.0
```

Once these changes have been made you could type mnall.x as in the spin-degenerate case, Sec. 2.3.2. However, since you have only copied the CTRL file to the new directory the program mnscf.run will start the spin-polarized calculations from scratch. An alternative and very elegant way consists of also copying the atomic files of the converged non-magnetic calculations. Now invoking mnall.x lets the program mnscf.run (which is the first program called from the shellscript mnall.x) detect the preconverged files. As already outlined in Sec. 2.1 this will invoke the restart facility of the program. During this startup phase the information from the converged spin-degenerate calculations is combined with the starting values for the magnetic moments per orbital as given in the CTRL file in order to find an optimal starting point for the spin-polarized calculation. As a consequence, very fast iteration to full self-consistency is achieved.

After convergence susan outlst30 summarizes the results as follows.

```
 ASW-1.9, program MNSCF started on majestix at Sat, 09 Mar 2002, 16:19:41.
 Calculation converged after   7 iteration(s).
                Start of Iteration   7
   1800 irreducible k-points generated from   27000( 30, 30, 30).
           Fermi energy - MTZ =     0.737422 Ryd.
```

```
            Indirect band gap =     0.140526 Ryd =     1.911955 eV
           DOS at Fermi energy:    38.499331 1/Ryd.
 Magnetic moment of unit cell =     4.000000
 Mean-square residual:           0.370346D-13
     Madelung energy:      -3.760944
       Zeeman energy:       0.000000
           total 3pV:      -0.204459
       virial energy:   -4794.076204
  variational energy:   -4794.076245
  qdiff =       0.00000000  <        0.00000001
  ediff =       0.00000000  <        0.00000001
 ASW-1.9, program MNSCF ended on majestix at Sat, 09 Mar 2002, 17:35:06.
```

According to this output a magnetic solution with a magnetic moment of $4.0\mu_B$ per unit cell has indeed been found. Its stability becomes obvious from a comparison of the total energy to that of the non-magnetic solution, the difference being 24 mRyd per formula unit. Worth mentioning is the considerable decrease of the DOS at $E_F$ as compared to the non-magnetic case. In addition, an indirect band gap of $\approx 1.9$ eV is obtained. Taken together these findings confirm the half-metallic of ferromagnetic behaviour of $CrO_2$ as reported already by Schwarz. This property is clearly visible in the band structure, Fig. 2.23, as well as in the corresponding

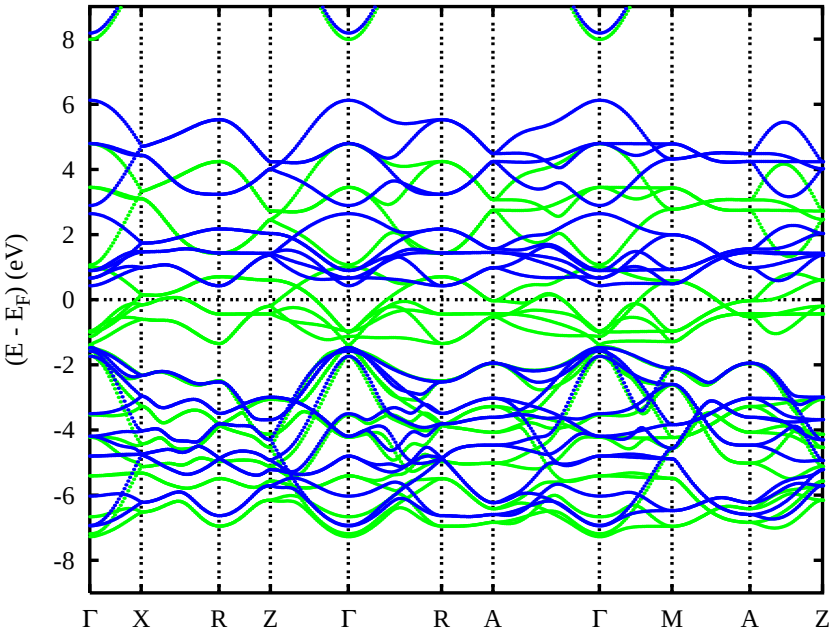



**Figure 2.23:** Electronic bands of ferromagnetic $CrO_2$ along selected symmetry lines within the first Brillouin zone of the simple tetragonal lattice, Fig. A.5(a).

density of states (DOS) as given in Fig. 2.24. While the O $2p$-dominated spin-up and spin-down states still occupy similar energy intervals the Cr $3d$ bands experience spin-splitting of about 2 eV. As a consequence, spin-minority Cr $3d$ states are shifted above the Fermi energy and left unoccupied. In contrast, metallic conductivity is carried alone by the spin-majority states. As for the spin-degenerate case the partial

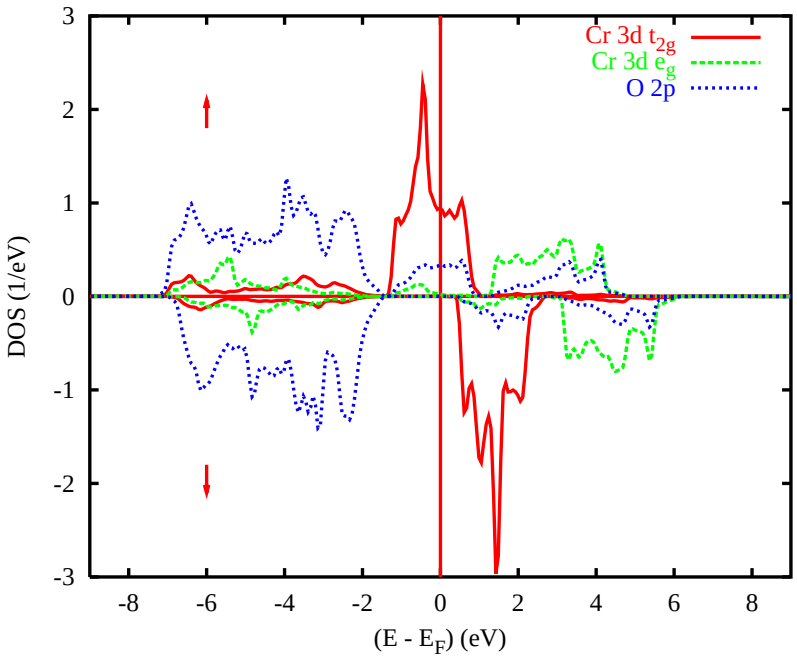



**Figure 2.24:** Partial densities of states (DOS) of ferromagnetic $CrO_2$ per formula unit.

DOS can be further decomposed in the single $d$ components. The results for the $3d$ $t_{2g}$ states are shown in Fig. 2.25. Worth mentioning is the up- and downshift of all

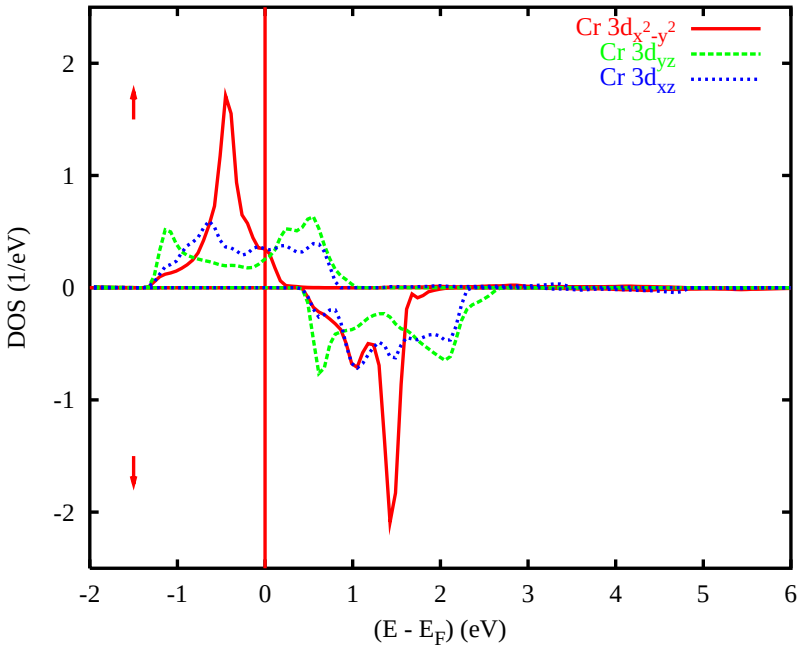



**Figure 2.25:** Partial Cr $3d$ $t_{2g}$ DOS of ferromagnetic $CrO_2$. Selection of orbitals is relative to the local rotated reference frame.

three spin-up and spin-down partial DOS as compared to the spin-degenerate case, Fig. 2.18, which is responsible for the decrease of the DOS at $E_F$. In agreement with the Stoner criterion it causes an energy gain especially for the spin-up $d_{x^2-y^2}$-like

electrons. In contrast, less energy has to be paid by the spin-down electrons since the sharp peak of the $d_{x^2-y^2}$ partial DOS was above $E_F$ already in the non-magnetic case.

### 2.3.4 Spin-polarized antiferromagnetic calculations

As already mentioned above $CrO_2$ shows an instability towards antiferromagnetic order if the lattice constants $a$ and $c$ or the the oxygen parameter $u$ are changed [26]. In the present section we demonstrate how the calculations for an antiferromagnet could be efficiently performed. Again, the CTRL file and the atomic files of the converged spin-polarized calculation for ferromagnetic $CrO_2$ should be first copied to a different directory. After that two changes have to be made. First, in order to take advantage of the spin-sublattice symmetry coming with antiferromagnetic order, we insert the token AFSYM=T into the category OPTIONS. Second, in order that the program can detect the particular antiferromagnetic order, each atomic site has to desginated as spin-up or spin-down. This is achieved by the tokens SPIN= to be appended to each atomic position in category SITE. The updated category will look like the following.

```
SITE    CARTP=F
        ATOM=CR POS= 0.000000  0.000000  0.000000 SPIN=UP
        ATOM=CR POS= 0.500000  0.500000  0.500000 SPIN=DN
        ATOM=O  POS= 0.302400  0.302400  0.000000 SPIN=UP
        ATOM=O  POS=-0.302400 -0.302400  0.000000 SPIN=UP
        ATOM=O  POS=-0.197600  0.197600  0.500000 SPIN=DN
        ATOM=O  POS= 0.197600 -0.197600  0.500000 SPIN=DN
        ATOM=E1 POS= 0.000000  0.500000 -0.250000 SPIN=DN
        ATOM=E1 POS= 0.000000  0.500000  0.250000 SPIN=UP
        ATOM=E1 POS=-0.500000  0.000000 -0.250000 SPIN=DN
        ATOM=E1 POS=-0.500000  0.000000  0.250000 SPIN=UP
        ATOM=E2 POS=-0.176474 -0.176474 -0.500000 SPIN=DN
        ATOM=E2 POS= 0.176474  0.176474 -0.500000 SPIN=DN
        ATOM=E2 POS= 0.323526 -0.323526  0.000000 SPIN=UP
        ATOM=E2 POS=-0.323526  0.323526  0.000000 SPIN=UP
        ATOM=E3 POS= 0.218838  0.411732 -0.355939 SPIN=DN
        ATOM=E3 POS=-0.218838 -0.411732  0.355939 SPIN=DN
        ATOM=E3 POS=-0.281162  0.088268 -0.144061 SPIN=UP
        ATOM=E3 POS= 0.281162 -0.088268  0.144061 SPIN=UP
        ATOM=E3 POS=-0.411732 -0.218838  0.355939 SPIN=DN
        ATOM=E3 POS= 0.411732  0.218838 -0.355939 SPIN=DN
        ATOM=E3 POS= 0.281162 -0.088268 -0.144061 SPIN=UP
        ATOM=E3 POS=-0.281162  0.088268  0.144061 SPIN=UP
        ATOM=E3 POS= 0.411732  0.218838  0.355939 SPIN=DN
        ATOM=E3 POS=-0.411732 -0.218838 -0.355939 SPIN=DN
        ATOM=E3 POS=-0.218838 -0.411732 -0.355939 SPIN=DN
        ATOM=E3 POS= 0.218838  0.411732  0.355939 SPIN=DN
        ATOM=E3 POS= 0.088268 -0.281162  0.144061 SPIN=UP
        ATOM=E3 POS=-0.088268  0.281162 -0.144061 SPIN=UP
        ATOM=E3 POS=-0.088268  0.281162  0.144061 SPIN=UP
        ATOM=E3 POS= 0.088268 -0.281162 -0.144061 SPIN=UP
```

Note that for each class there must be an equal number of SPIN=UP and SPIN=DN entries. Again the self-consistent calculation is started by typing mnall.x

at the systems prompt. In case you have forgotten to copy the atomic files of the ferromagnetic calculation the program will start from scratch and, hence, need more iterations. Of course, for antiferromagnetic materials it is *not* not necessary to perform a ferromagnetic calculation first. You could also use the atomic files of a spin-degenerate calculation as a starting point for an antiferromagnetic calculation.

After convergence susan outlst30 summarizes the results as follows.

```
ASW-1.9, program MNSCF started on majestix at Wed, 13 Mar 2002, 08:30:07.
Calculation converged after   7 iteration(s).
               Start of Iteration   7
  3600 irreducible k-points generated from   27000( 30, 30, 30).
          Fermi energy - MTZ =     0.730387 Ryd.
          DOS at Fermi energy:    92.466301 1/Ryd.
Magnetic moment of UP atoms  =     1.711041
Magnetic moment of unit cell =     0.000000
Mean-square residual:           0.270493D-12
    Madelung energy:      -3.744479
      Zeeman energy:       0.000000
          total 3pV:      -0.315314
      virial energy:   -4794.054609
 variational energy:   -4794.054649
 qdiff =       0.00000000  <        0.00000001
 ediff =       0.00000000  <        0.00000001
ASW-1.9, program MNSCF ended on majestix at Wed, 13 Mar 2002, 09:43:39.
```

According to this output a magnetic solution with a magnetic moment of $\approx 1.7\mu_B$ per sublattice has indeed been found. However, while being stable as compared to the spin-degenerate case, its total energy is by 11 mRyd per formula higher as that of the ferromagnetic solution.

The electronic structure is displayed in Fig. 2.26 and the corresponding partial DOS given in Fig. 2.27. Note the degeneracy of the spin-up and spin-down bands along most of the symmetry lines, which is typical for an antiferromagnet. The splittings along the lines A-Γ, Γ-M, and A-Z are indicative of the reduced symmetry of the antiferromagnetic structure, which calls for a larger irreducible wedge of the first Brillouin zone.

Finally, the partial Cr $3d$ $t_{2g}$ DOS are shown in Fig. 2.28, where again the rotated reference frame was used. In contrast to the ferromagnetic case, energetical down-shift of the spin-up states is much less. As a consequence the $d_{x^2-y^2}$ peak appears near the Fermi energy and contributes to the rather large DOS at $E_F$. Furthermore, it is responsible for the increase in total energy as compared to ferromagnetic $CrO_2$.

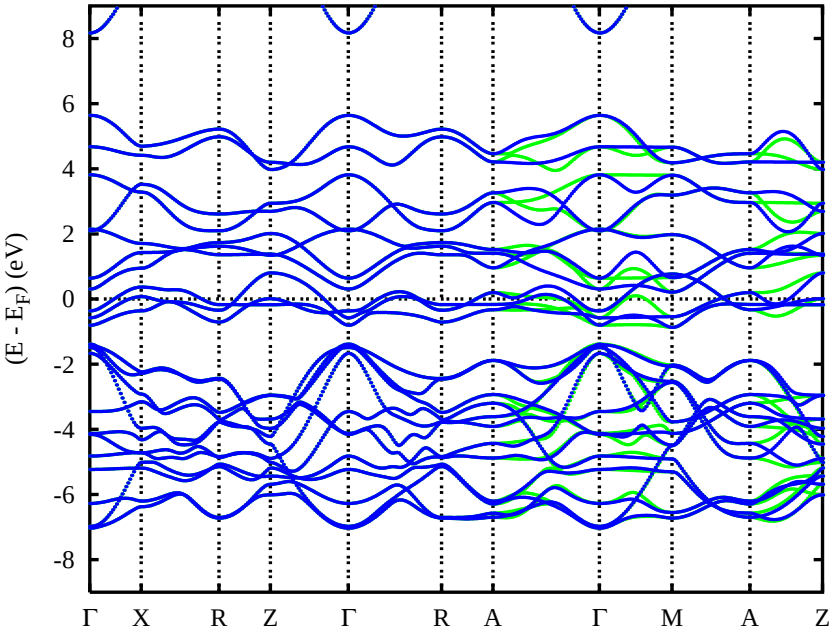



**Figure 2.26:** Electronic bands of antiferromagnetic $CrO_2$ along selected symmetry lines within the first Brillouin zone of the simple tetragonal lattice, Fig. A.5(a).

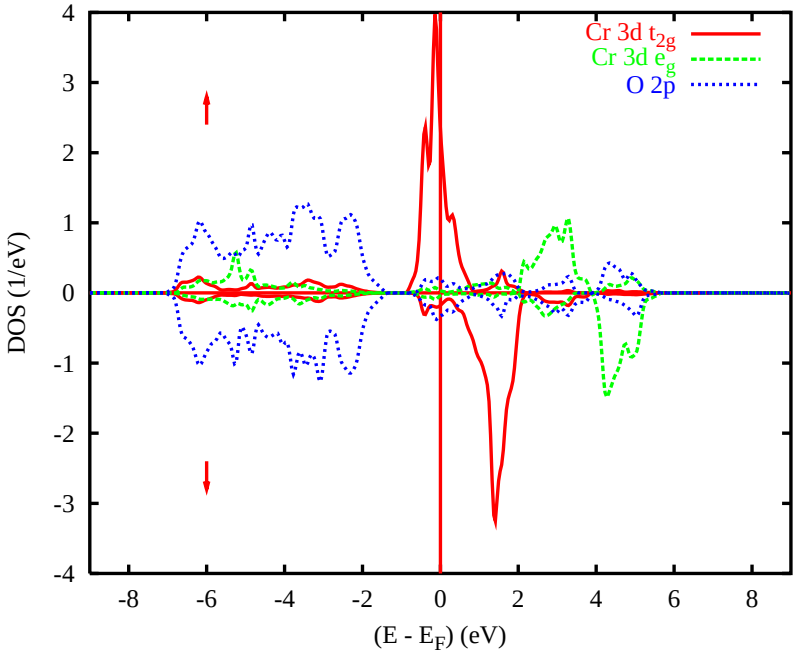



**Figure 2.27:**Partial DOS of antiferromagnetic $CrO_2$ per formula unit.

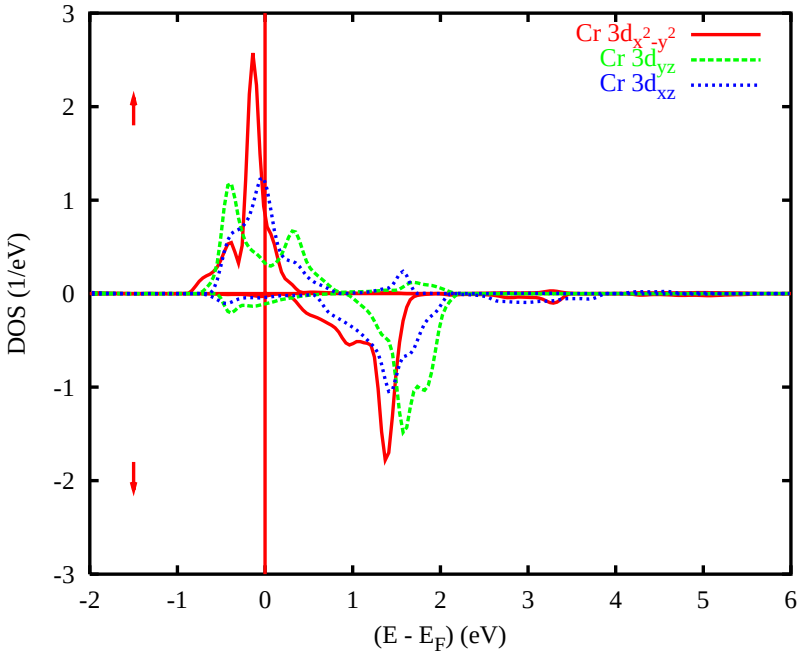



**Figure 2.28:** Partial Cr $3d$ $t_{2g}$ DOS of antiferromagnetic $CrO_2$. Selection of orbitals is relative to the local rotated reference frame.

# Chapter 3

# Organization of the ASW program package

In the present chapter an overview over all files coming with the distribution is given. This includes mainly four groups. While the calculations are performed by one of the main programs the plot programs serve the purpose of visualizing the results. Work of both steps is made easier by a bunch of shellscripts. Finally, a list of all files, which are either included in the distribution or else created by one of the programs, is given.

## 3.1 Main programs and shellscripts

In the present section all main programs of the ASW distribution are discussed. In contrast to the plot programs presented in the following section, all main programs (with the exception of mnmpr.run) start reading the CTRL file.

As has been already outlined in Sec. 2.1, the ASW program package includes both the executables (with ending ".run") and corresponding shellscripts (with ending ".x"). The latter call the executables and specify the respective output files. In addition, the shellscripts provide commands for the most common batch queuing systems as IBM's LoadLeveler (LoadL), Network Queuing System (NQS), and the public domain packages Portable Batch System (PBS), Distributed Queuing System (DQS) as well as Sun's Grid Engine (SGE). Finally, some of the shellscripts include appropriate modifications of the CTRL file. A fully commented example shellscript is contained in the file mnhlp.x, which should be consulted for more information. In the following discussion we will use the names program, executable and shellscript synonymously.

### 3.1.1 mnmpr.run, mnmpr.x

While all main programs calculate the parameters of the actual machine at the very beginning, this program prints them out and stops. The output usually looks like file MACH coming with the distribution.

```
ASW-1.9, program MNMPR started on majestix at Fri, 11 Jan 2002, 08:31:10.
```

```
     Copyright (C) 1992-2002 Volker Eyert
     Please see file COPYRIGHT for details

 Machine-specific parameters affecting floating-point arithmethic:

 ibase:          2                (base of the machine)
 it:            53                (number of (base) digits in the mantissa)
 xeps:      2.22044604925031D-16  (= min, such that 1.0 + xeps > 1.0)
 xepsn:     1.11022302462516D-16  (= min, such that 1.0 - xepsn < 1.0)
 minexp:     -1021                (min exponent before (gradual) underflow)
 xzer:      2.22507385850720-308  (underflow threshold = base**(minexp-1))
 xmin:      2.22507385850720-308  (safe min, 1/xmin does not overflow)
 maxexp:      1024                (largest exponent before overflow)
 xinf:      1.79769313486232+308  (overflow threshold = base**maxexp*(1-xepsn))
 irnd:           1                (= 0/1 for chopping/rounding in addition)

 task  1, total   :  cpu time:     0.00000 sec

 ASW-1.9, program MNMPR ended on majestix at Fri, 11 Jan 2002, 08:31:10.
```

This toy program mainly serves the purpose of checking the compiler and machine capabilities.

### 3.1.2 mnhlp.x

In order to summarize the way most of the shellscripts work I have created this file, which is a fully commented version of the shellscript mnmpr.x.

```
#!/bin/sh
###############################################################################
#
#     Command file mnhlp.x for queuing a batch job executing an
#     ASW program. This file serves as a commented example file
#     for all the other shell-scripts.
#     The script is prepared for use with the LoadLeveler (LoadL),
#     the Network Queuing System (NQS), the Portable Batch System
#     (PBS), the Distributed Queuing System (DQS), and Sun's Grid
#     Engine (SGE).
#     Please adapt to your own environment and needs.
#
#     Version ASW-1.9 29.09.2003 Volker Eyert
#     Copyright (C) 1992-2003 Volker Eyert
#     Please see file COPYRIGHT for details.
###############################################################################
#
#
#     LoadLeveler specifications:
#
# Specify job queue:
# @ class        = veryshort
#
# Specify system and hardware requirements:
# @ requirements = (OpSys == {"AIX42" "||" "AIX43"}) && (Feature == "pwr3")
```

```
#
# Specify requirements for parallel execution:
#   job_type     = parallel
#   requirements = (Adapter == "hps_user")
#   min_processors =  1
#   max_processors = 14
# @ resources = ConsumableCpus(1) ConsumableMemory(80)
#
# Specify executable (done explicitly here; see below):
#   executable   =
#
# Specify stdin, stdout and stderr:
# @ input        = /dev/null
# @ output       = job.$(Cluster).$(Process).out
# @ error        = job.$(Cluster).$(Process).err
#
# Mark job not-restartable:
# @ restart      = no
#
# Suppress any notification:
# @ notification = never
#
# Finally queue the job:
# @ queue
#
###############################################################################
#
#
#     Network Queuing System (NQS) specifications:
#
# Specify job queue:
#QSUB -q high_prio
#
# Specify CPU-time and memory:
#QSUB -lT 600
#QSUB -lM 256MB
#
# Specify start from the current working directory:
if [ $QSUB_WORKDIR ]; then cd $QSUB_WORKDIR; fi
#
# Specify stdout and stderr:
#QSUB -o job.out
#QSUB -e job.err
#
# Specify shell:
#QSUB -s /bin/bash
#
# Mark the job not-restartable:
#QSUB -nr
#
# Require mailing at job starting and ending time:
# QSUB -mb
# QSUB -me
#
###############################################################################
```

```
#
#
#     Portable Batch System (PBS) specifications:
#
# Specify job queue:
#PBS -q dft
#
# Specify system and hardware requirements:
# PBS -l resc = "(arch==’solaris’)||(arch=’irix’)"
#
# Specify requirements for parallel execution:
#PBS -l nodes=1
#
# Specify CPU-time and memory:
# PBS -l cput=168:00:00
#PBS -l mem=256MB
#
# Specify start from the current working directory:
if [ $PBS_O_WORKDIR ]; then cd $PBS_O_WORKDIR; fi
#
# Specify stdout and stderr:
#PBS -o job.out
#PBS -e job.err
#
# Specify the shell:
#PBS -S /bin/bash
#
# Export all environmental variables:
#PBS -V
#
# Mark the job not-restartable:
#PBS -rn
#
# Suppress any notification:
#PBS -m n
#
###############################################################################
#
#
#     Distributed Queuing System (DQS) and Sun’s Grid Engine (SGE)
#     specifications:
#
# Specify job queue:
#$ -q dawai_express
#
# Specify CPU-time and memory:
#$ -l qty.eq.1,runtime.ge.600,memory.ge.256
#$ -l memory.ge.256
#
# Specify start from the current working directory:
#$ -cwd
#
# Specify stdout and stderr:
#$ -o job.out
#$ -e job.err
```

```
#
# Specify the shell:
#$ -S /bin/bash
#
# Export all environmental variables:
#$ -V
#
# Mark the job not-restartable:
#$ -r n
#
# Suppress any notification:
#$ -m n
#$
#
###############################################################################
#
BINDIR=$HOME/asw/aswhp
#
$BINDIR/mnmpr.run > outmpr
#
###############################################################################
```

You may have a look at this prototypical shellscript before adapting any of the other scripts.

### 3.1.3 mnsym.run, mnsym.x

This program reads in the CTRL file and performs a symmetry check of the crystal structure. Alternatively, depending in token GENPOS=, it builds the full unit cell from a minimal set of atomic positions and the symmetry generators given in the CTRL file. In both cases the result is printed to file outsym and a file CBAK is created, which contains an updated version of the CTRL file.

### 3.1.4 mnfre.run, mnfre.x

This program performs a free atom calculation for each atomic class in the unit cell and generates a first guess to the full charge density and potential from the superposition of the respective free atom quantities. This usually goes under the name Mattheiss construction. The calculation is done for points along a line or lying within a plane as defined by tokens ORIGIN=, RPLOT1=, and RPLOT2= depending on the number specified by tokens NPDIV1= and NPDIV2=. Set NPDIV2=0 to enforce a line plot. The result is written to file FREE and can be plotted using plfre.run or plfre.x.

### 3.1.5 mnovl.x

An overlap check of all atomic spheres can be performed using this shellscript, which calls mnfre.run but stops after the overlap check. The result is printed to file *outovl*, which lists the overlap limits, the positions and radii of all atomic spheres. In addition, a full list of pairs of atoms, their distance, and their overlap is given together

with exclamation marks pointing to large overlaps. The number of entries in the list can be controlled by token VERBOS=. With the default setting VERBOS=30 only pairs with a nonvanishing overlap are included.

### 3.1.6 mnpac.run, mnpac.x

In order to invoke the sphere packing using the sphere geometry optimization algorithm you have to call this program. In a first part free atoms are superposed in order to have the potential along lines connecting all atoms. From this information the muffin-tin radii of the physical atoms can be evaluated. In a second step the muffin-tin radii are blown up until the overlap limits specified by tokens OBYDMX= and OBYRMX= are reached. If space filling as fixed by token FILLING= can not yet be achieved, a search for the largest possible empty sphere sets in. Once this has been found again all spheres are blown up until the maximum space filling compatible with the overlap limits is reached. This process is repeated until full space filling has been achieved. However, see the description of all tokens in category PACK for more detailed information.

WARNING: As it stands this program was designed for the spin-degenerate case only. So make sure that NSPIN=1 (or this token is missing at all) in the CTRL file.

### 3.1.7 mnstr.x

This special variant of shellscript mnpac.x likewise calls the packing program but stops after the determination of the muffin-tin radii. The result as contained in file STRU can be used for plotting the crystal structure with the plot program plstr.run (or plstr.x).

### 3.1.8 mnscf.run, mnscf.x

This is the most comprehensive of all programs. After reading the CTRL file it performs the self-consistent calculations until full convergence or the maximum iteration count as specified by token NITBND= has been reached. In this course, the program calculates partial charges and magnetic moments, densities of states and the total energy and pressure. The program falls into two parts. In a band part, calculated intraatomic matrix elements are used to set up the secular matrix. From solving the corresponding eigenvalue problem eigenvalues and eigenvectors are obtained and the respective contributions added to the partial densities of states. This process is repeated for all **k**-points in the irreducible wedge of the first Brillouin zone. Finally, from the partial densities of states the partial charges and magnetic moments as well as the first four moments of the partial DOS are constructed. The latter could be translated into energies, hence, boundary conditions for the intraatomic wave functions. The band part usually takes more than 95% of the execution time of the program. Note that in during this time no files are read or written. Hence, if you want to stop the program for whatever reason without loosing the atomic files you should do it during the band part.

In the atomic part the partial charges and boundary conditions (logarithmic derivatives) provided by the band part are used to calculate the intraatomic wave functions and from this the intraatomic charge density. Within the notions of density functional theory and the local density approximation the latter gives rise to the effective potential and total energy. Finally, the potential enters a radial Schrödinger equation leading to new intraatomic wave functions and matrix elements for use in the band part. However, the effective potential could be likewise used to construct via the wave functions a new charge density. This gives rise to an iteration towards full intraatomic convergence within the limits set by the partial charges and logarithmic derivatives as grown out of the previous band part. Only after this self-consistency cycle has converged, new wave functions and matrix elements for the subsequent band part are supplied. Note that during the intraatomic calculations the atomic files are accessed several times. Stopping the program in this step thus might cause a loss of files.

### 3.1.9 mndos.x

During normal execution of program mnscf.run only $l$-resolved partial densities are usually calculated. This saves both execution time and memory. Once the iterations have converged, the shellscript mndos.x will call the program mnscf.run once more. If present, before tokens SAVDOS=F and SAVCOOP=F will be changed to SAVDOS=T and SAVCOOP=T, respectively, BZINT=SMS is set to BZINT=HPS and NKABC=6 is changed to NKABC=30. After the band part has been finished and the partial DOS and COOPs are written to files DOS and COOP the porgram stops and the above changes of the CTRL file are reset to their original values.

### 3.1.10 mnbnd.run, mnbnd.x

Another quantity of interest after full convergence has been achieved is the band structure. It is calculated by employing the shellscript mnbnd.x, which scans the **k**-space path defined by tokens ENDPT= and calculated eigenvalues, the $E(\mathbf{k})$. The result is written to file BNDE. In addition, if ORBWGT=T has been specified, eigenvectors, i.e. wave functions are calculated for latter plotting of orbital weighted band structures as shown in Figs. 2.3 to 2.6 and 2.14. The result is written to file BNDV. Note that this file might be very large up to some hundreds of MByte. In case you want to calculate orbital weighted band structures check the output file *outbnd* and search for the line beginning with

```
File BNDV will need approx.
```

which gives a good estimate of the disk space needed.

### 3.1.11 mnall.x

Most of the aforementioned calculations can be made automatic by employing the shellscript mnall.x, which calls the program mnscf.run successively for an increasing number of **k**-points ranging from NKABC=6 to NKABC=30. After this it includes

the calculations already covered by the shellscripts mnbnd.x and mndos.x. For compounds with larger unit cells or for semiconductors and insulators it is recommended to copy the shellscript mnall.x to the actual directory and delete the lines

```
$BINDIR/upctr "NKABC=16" "NKABC=20" &&
$BINDIR/mnscf.run > outlst20 &&
$BINDIR/upctr "NKABC=20" "NKABC=30" &&
$BINDIR/mnscf.run > outlst30 &&
```

In this case the self-consistent calculations will stop after the iterations for NKABC=16 have converged.

### 3.1.12 mnscl.run, mnscl.x

Whenever it is necessary to perform a supercell calculation this utility program does the supercell setup. In the CTRL file an additional category SUPCELL must be given, which otherwise looks identical to category STRUC but holds the information about the lattice of the supercell. In addition, token EQUIV=F can be used to copy existing atomic files to equivalent files of atoms located in different regions of the new supercell. Token PSHIFT= will introduce an overall shift of all atoms and, finally, token QSWAVE= holds a spin wave vector of a collinear antiferromagnetic structure imposed on the supercell.

### 3.1.13 upctr

This shellscript allows to update the CTRL file. Just type e.g. upshl ALAT=6.83079 ALAT=6.89979 to shrink the lattice constant of Cu by 1% or upctr QUIT= QUIT=BND to enforce a program stop after the band part. Actually, the script upctr is contained in many other of the above listed shellscripts.

### 3.1.14 susan

This utility shellscript has been created to extract a **SU**mmary of **S**elected results from the **A**SW output and to **N**otify about the progress of the iterations. The shellscript is listed here.

```
#!/bin/sh
#############################################################################
#
#     Shell-script susan SUmmarizes Selected results of the ASW
#     self-consistency cycle and Notifies about the progess of
#     the iterations.
#
#     Version ASW-1.9 11.01.2002 Volker Eyert
#     Copyright (C) 1994-2002 Volker Eyert
#     Please see file COPYRIGHT for details.
#############################################################################
#
#
```

```
if [ -s $1 ]
then
   echo " "
   grep "started on"                        $1
   grep "converged after"                   $1 | tail -1
   grep "Start of Iteration"                $1 | tail -1
   grep "irreducible k-points"              $1 | tail -1
   grep "Fermi energy - MTZ"                $1 | tail -1
   grep "band gap"                          $1 | tail -1
   grep "DOS at Fermi energy:"              $1 | grep "1/Ryd." | tail -1
   grep "Magnetic moment of UP atoms"       $1 | tail -1
   grep "Magnetic moment of unit cell"      $1 | tail -1
   grep "Unit cell deviates"                $1 | tail -1
   grep "square"                            $1 | tail -1
   grep "Madelung energy:"                  $1 | tail -1
   grep "Zeeman energy:"                    $1 | tail -1
   grep "total 3pV:"                        $1 | tail -1
   grep "virial energy:"                    $1 | tail -1
   grep "variational energy:"               $1 | tail -1
   grep "qdiff ="                           $1 | tail -1
   grep "ediff ="                           $1 | tail -1
   grep "ended on"                          $1
   echo " "
else
   echo usage:  susan filename
fi
#
#########################################################################
```

Typing e.g. susan outlst30 will generate the following about a 20 lines on screen (see Sec. 2.1.2) including information about the **k**-space grid, the position of the Fermi energy, the DOS at $E_F$, the total magnetic moment, the (variational) total energy, and the self-consistency level already reached as coded by the entries qdiff and ediff.

## 3.2 Plot programs and shellscripts

The plot programs coming with the distribution are distinguished from the main programs in that they do are completely independent of the CTRL file but start from files created by one of the main programs.

As for the main program there exist both executables (with ending ".run") and corresponding shellscripts (with ending ".x"), which latter call the executables and include an automatic call of the respective plot program as Gnuplot, RasMol, or XMakemol. In addition, intermediate files are deleted at the end. Whenever these files are needed e.g. for exchange with other plot porgams one should employ the executable rather than the shellscript. Again, we will use the names program, executable and shellscript synonymously in the following discussion.

As an extra service to the user all specifications of the plot as entered during the dialog are echoed to a file PLIx, where x=S, F, B, D, or C depending on the plot porgram. After the name of this file has been changed it can be edited and used for new plots by typing e.g. mnbnd.run < PLIBnew or mnbnd.x PLIBnew.

Another feature common to all plot programs concerns the title of the plot. On invoking any of the plot programs one is asked for specifying the title. Entering a "/" here will make the program use the title given by the first line of the CTRL file. In addition, if you type & arbitrary text the title will be constructed from the first line of the CTRL file complemented by the "arbitrary text".

### 3.2.1 plstr.run, plstr.x

This program starts from file STRU, which holds atomic positions and radii of all atoms within the plot region specified by tokens ORIGIN= and RPLOTn= and prepares for later use of RasMol or XMakemol. During the dialog colours and sizes of the atomic spheres as well as the plot perspective can be specified.

### 3.2.2 plfre.run, plfre.x

This program starts from file FREE, which holds the overlapping free atom charge densities and potential within the plot region specified by tokens ORIGIN= and RPLOTn=. It prepares for later use of Gnuplot. During the dialog you can choose between a contour or 3D plot. In addition, plot of the core charge density, valence charge density or both or else the potential can be specified.

### 3.2.3 plbnd.run, plbnd.x

This program allows for plotting of the (weighted) band structure. It start from file BNDE containing the $E(\mathbf{k})$ for all bands and, if token ORBWGT=T has been specified, from file BNDV, which holds the corresponding eigenvectors. These data are prepared for later plotting with either Gnuplot or LaTeX. In the latter case the shellscript plbnd.lx cwhas to be used in addition. During the dialog invoked by calling plbnd.x among other things the energy scale and reference as well as the energy window must be specified. An example has been given in Sec. 2.1.3.

If output should be prepared for Gnuplot the program usually creates a file *bnd.ps*, which holds a postscript copy of the plot. A particular nice feature arises from specifying writing to a LaTeXfile. This will produce a file *bnd.tex*. Next you have to invoke the shellscript plbnd.lx (see below), which will run LaTeXon a file named *plbnd.tex*, into which *bnd.tex* is included via LaTeX's input command. As a result, in case the band structure is not too complicated, hence, file *bnd.tex* is not too large, a final file *bnd.ps* is created. This LaTeXoption is particularly suited for plotting weighted band structures.

In the latter situation another important feature comes in. First, you are asked to specify the orbitals to be included. Actually, the program plots for each **k**-point and band the projection of a particular orbital onto the full wave function. Furthermore, the program allows to specify the coordinate system, within which the orbitals are selected. This is particularly useful, when the crystal field split $d$ orbitals of a transition metal atom are to be considered as in the prvious examples of $FeS_2$ or $CrO_2$. for this reason the plot routine asks for a rotation matrix, which can be given in the same form as the symmetry matrices in token SYMOPS of the

CTRL file. See Chap. 4 for more details and examples. Alternatively, the string characterizing the rotation can be written to a file named ROTS. In that case just enter "f" in the dialog.

### 3.2.4 pldos.run, pldos.x

The plot program pldos.x enables plotting of the total and partial densities of states (DOS). Both the DOS itself and the integrated DOS can be displayed. Input is read from file DOS. Intermediate files are prepared for both Gnuplot and LaTeX, however, for the DOS Gnuplot produces a nicer output, usually contained in the postscript file *dos.ps*. During the dialog the user is asked for the energy window, scale, and reference. In addition, for each curve the orbitals to be included and a scaling factor must be specified. Finally, the curve can be broadened by folding with a Gaussian or a Lorentzian. The number of curves is limited to nine by the porgram.

Finally, as for plotting the band structure a rotation matrix can be specified for each curve in order to make the program refer the orbitals to a rotated coordinate system.

### 3.2.5 plcop.run, plcop.x

This program must be used for plotting the total and partial crystal orbital overlap population (COOP). Again, also the integrated quantities can be plotted. Input is read from file COOP. If Gnuplot is specified for output usually a file *coop.ps*. The dialog proceeds in much the same way as for pldos.x. However, note that for each curve *two* orbitals have to given. Again, Gaussian or Lorentzian smearing can be used.

### 3.2.6 plbnd.lx, pldos.lx, plcop.lx

These shellscript include files bnd.tex, dos.tex, and coop.tex into LaTeXenvelope files plbnd.tex, pldos.tex, and plcop.tex, respectively, and run LaTeXas well as dvips. As a result, postscript files bnd.ps, dos.ps, and coop.ps, respectively, are obtained.

### 3.2.7 plbnd.tex, pldos.tex, plcop.tex

These are LaTeXenvelope files used to include bnd.tex, dos.tex, and coop.tex, respectively, and to produce postscript files of the band structure, DOS or COOP.

## 3.3 Installation shellscripts

### 3.3.1 Makefile

This is the makefile of the ASW program package for compiling and linking the source and object files in a Unix/Linux environment. Typing make all will automatically create all executables and typing make install will install them in a prespecified directory. A full description of this process has already been given in Sec. 1.4.

### 3.3.2 mkall.x

The compiling and linking step can be performed as a background job by invoking the shellscript mkall.x.

### 3.3.3 upshl

This shellscript updates the specification of the directory, where the executables are installed, in all shellscripts listed in Secs. 3.1 and 3.2. It is automatically called during the make install process.

## 3.4 Data files

Input data as well as data created by the ASW programs are spread over a large number of files. Yet, exchange of the user with the programs is limited essentially to a single file, i.e. the CTRL file. All other files serve the purpose of holding specific information about the distribution or are meant for information exchange between different programs. Neither of these latter files has to be changed by the user. Anyway, in the following list a short description is given for all files.

### 3.4.1 README

The README comprises a shortcut version of this user guide. Actually, the present manuscript has its roots in the README file. It contains very brief information about the distribution, installation and execution of the programs as well as some remarks about external software to be used with the ASW program package.

### 3.4.2 INSTALL

A concise description of the installation process is provided by this file.

### 3.4.3 COPYRIGHT

The COPYRIGHT file covers all the copyright information concerning the ASW program package. It is printed in Sec. 1.6 and should be read by any user at the very beginning.

### 3.4.4 LICENCE

The LICENCE file contains a single line indicating the expiration date for the present licence.

### 3.4.5 CTRL

The CTRL file is the one and only input file to the main programs. It

- is fully free format and thus allows for a flexible setup,

- is subdivided in sections (categories) each containing
- a set of catchwords (tokens),
- has many options,
- many useful defaults.

The CTRL file contains information about

- the crystal structure,
- the constituent atoms,
- the symmetry lines,
- settings for input/output,
- parameters for sphere packing,
- Brillouin zone sampling,
- iteration acceleration,
- and convergence criteria.

All files following in the present list are automatically created and deleted by the programs. A complete description of the CTRL file and all its categories and tokens is given in Chap. 4.

### 3.4.6 CBAK

All the main programs except for mnmpr.run write an updated copy of the CTRL file to file CBAK unless a different name is specified by token WRITE= in category IO.

### 3.4.7 CNEW

The packing program mnpac.run writes a new CTRL file named CNEW (remember: file CTRL is never written to), which includes possibly created empty sphere positions as well as the radii of all spheres. Before proceeding with the self-consistent calculations you must rename the file CNEW to CTRL.

### 3.4.8 HELP

A first overview over the capabilities of the ASW program package can be best obtained by creating the HELP file. In case this is missing in the distribution it can be most easily obtained by writing the line

IO HELP=T

to the CTRL and running any of the ASW main programs (those starting with "mn"). This will generate the HELP file listed here.

```
HELP-file for ASW-1.9


category HEADER
  title

category VERSION
  token ASW- of cast double
        Version/release of program which has written CTRL

category IO
  token HELP= of cast logical
        Switch to print HELP file
  token SHOW= of cast logical
        Switch to echo CTRL file to output
  token VERBOS= of cast integer
        Verbosity level for printing of output
  token IACTIV= of cast logical
        Switch to start interactive mode
  token CLEAN= of cast logical
        Switch to shrink atomic files after convergence
  token WRITE= of cast character
        Name of file to which a copy of CTRL file is written
  token EXTENS= of cast character
        Default extension for all files besides CTRL file

category OPTIONS
  token REL= of cast logical
        Switch to scalar-relativistic mode
  token LSCPL= of cast logical
        Switch to turn on LS-coupling
  token NSPIN= of cast integer
        Number of spin-channels
  token AFSYM= of cast logical
        Switch to use antiferromagnetic symmetry if present
  token BEXT= of cast double
        External magnetic field (in z-direction)
  token XCPAR= of cast character
        String for XC-parametrization
  token GGA= of cast character
        String for GGA-parametrization
  token OVLCHK= of cast logical
        Switch to perform overlap check
  token CCOR= of cast logical
        Switch to use combined correction to the ASA
  token FULPOT= of cast logical
        Switch to full potential calculation
  token CORDRD= of cast logical
        Switch to frozen-core calculations

category STRUC (mandatory)
  token UNITS= of cast character
```

```
        Units to be used in STRUC and SUPCELL
  token ALAT= of cast double (mandatory)
        Lattice constant A in UNITS
  token PLAT= of cast double and length 9 (mandatory)    --- OR:
        Primitive translations in units of A
  [
  token SLAT= of cast character
        String for Bravais lattice
  token BBYA= of cast double    --- OR:
        Ratio of the lattice constants B/A
  token BLAT= of cast double
        Lattice constant B in UNITS
  token CBYA= of cast double    --- OR:
        Ratio of the lattice constants C/A
  token CLAT= of cast double
        Lattice constant C in UNITS
  token GAMMA= of cast double
        Angle used for the monoclinic lattices
  ]

category CLASS (mandatory)
  token NCLASS= of cast integer
        Number of different atoms (classes)
  token ATOM= of cast character (mandatory)
        Class labels
  token Z= of cast integer (mandatory)
        Atomic numbers
  token R= of cast double    --- OR:
        Atomic sphere radii in atomic units
  token R/RA= of cast double
        Atomic sphere radii in arbitrary units
  token LMXL= of cast integer
        Maximum angular momentum including the lower waves
  token LMXI= of cast integer
        Maximum angular momentum including intermediate waves
  token CONF= of cast integer and length 4
        Principal quantum numbers for all orbitals
  token COORB= of cast integer
        Orbitals to be included in the COOP
  token QVAL= of cast double and length 4
        Valence charges for starting from scratch
  token MVAL= of cast double and length 4
        Valence moments for starting spinpolarized calculations

category SITE (mandatory)
  token NBAS= of cast integer
        Number of atoms
  token CARTP= of cast logical
        Switch to treat atomic positions as cartesian
  token CHOUT= of cast logical
        Switch to change interpretation of atomic positions
  token ATOM= of cast character (mandatory)
        Class labels
  token POS= of cast double and length 3 (mandatory)
        Positions of basis atoms
```

```
  token SPIN= of cast character
        String characterizing the spin direction

category SYMGRP
  token GENPOS= of cast logical
        Switch to complete atomic basis by use of symmetry
  token SYMOPS= of cast character and length 48
        Strings for space group generators
  token CARTR= of cast logical
        Switch to treat rotation axes as cartesian
  token CARTT= of cast logical
        Switch to treat fractional translations as cartesian

category PACK
  token FILLNG= of cast double
        Filling factor for atomic sphere volumes
  token OBYDMX= of cast double
        Maximum allowed overlap/distance
  token OBYRMX= of cast double
        Maximum allowed overlap/radius
  token ESBONS= of cast double
        Empty sphere bonus to be added to OBYDMX and OBYRMX
  token NCEMAX= of cast integer
        Maximum number of allowed empty sphere classes
  token RADMIN= of cast double
        Minimum muffin-tin radius for empty spheres
  token RADMAX= of cast double
        Maximum atomic sphere radius for empty spheres
  token POTWIN= of cast double
        Potential window used for setup of radii window
  token RADACC= of cast double
        Accuracy required for sphere radii
  token POSACC= of cast double
        Accuracy required for empty sphere positions

category ENVEL
  token NKAP= of cast integer
        Number of envelope function energies
  token EKAP= of cast double
        Array of envelope function energies
  token EWPAR= of cast double
        Ewald parameter without volume scaling
  token EWTOL= of cast double
        Precision sought for Ewald sums

category BZSMP
  token NKABC= of cast integer and length 3
        Number(s) of k-mesh points
  token BZINT= of cast character
        String for Brillouin-zone integration scheme
  token EMIN= of cast double
        Minimum energy for DOS calculation
  token EMAX= of cast double
        Maximum energy for DOS calculation
  token NDOS= of cast integer
```

```
        Number of division of the interval EMIN-EMAX
 ... if BZINT=HPS the following two tokens are needed:
  token NORD= of cast integer
        Order of approximant in high-precision BZ sampling
  token WIDTH= of cast double
        Broadening in high-precision BZ sampling
 ... if BZINT=LTM the following token is needed:
  token EFTOL= of cast double
        Precision sought for Fermi energy
  token SAVDOS= of cast logical
        Switch to save calculated partial DOSs
  token SAVCOOP= of cast logical
        Switch to save calculated COOPs
  token CTYPE= of cast character
        String for type of COOP calculation
  token MSPLIT= of cast logical
        Switch to calculate m-splitted COOPs
  token TEMPFD= of cast double
        Temperature for Fermi-Dirac folding

category CONTROL
  token START= of cast character
        String controlling program starting point
  token QUIT= of cast character
        String controlling program end point
  token FREE= of cast logical
        Switch to free atom calculations / free electron bands
  token NITBND= of cast integer
        Maximum number of band iterations
  token CNVG= of cast double
        Convergence tolerance for zero moments
  token CNVGET= of cast double
        Convergence tolerance for total energy
  token NITATM= of cast integer
        Maximum number of intraatomic iterations
  token CNVGQA= of cast double
        Convergence tolerance for atomic charges

category MIXING
  token NMIXB= of cast integer
        Number of previous iterations used in band mixing
  token BETAB= of cast double
        Mixing parameter for band mixing
  token INCBB= of cast logical
        Switch to increase band mixing parameter
  token NMIXA= of cast integer
        Number of previous iterations used in atom mixing
  token BETAA= of cast double
        Mixing parameter for atom mixing

category SUPCELL
  token ALAT= of cast double
        Supercell lattice constant A in UNITS
  token PLAT= of cast double and length 9    --- OR:
        Supercell primitive translations in units of A
```

```
  [
  token SLAT= of cast character
        String for supercell Bravais lattice
  token BBYA= of cast double    --- OR:
        Ratio of the supercell lattice constants B/A
  token BLAT= of cast double
        Supercell lattice constant B in UNITS
  token CBYA= of cast double    --- OR:
        Ratio of the supercell lattice constants C/A
  token CLAT= of cast double
        Supercell lattice constant C in UNITS
  token GAMMA= of cast double
        Angle used for the monoclinic supercell lattices
  ]
  token EQUIV= of cast logical
        Switch to keep subcells equivalent
  token CARTS= of cast logical
        Switch to treat position shift vector as cartesian
  token PSHIFT= of cast double and length 3
        Shift vector for the supercell atomic positions
  token CARTQ= of cast logical
        Switch to treat spin wave vector as cartesian
  token QSWAVE= of cast double and length 3
        Spin wave vector of (collinear) antiferromagnets

category SYMLIN
  token NPAN= of cast integer
        Number of symmetry lines (panels)
  token NPTS= of cast integer
        Maximum total number of points
  token ORBWGT= of cast logical
        Switch to calculate orbital weights
  token SPATH= of cast logical
        Switch to use spherical pathways
  token EPHOT= of cast double
        Photon energy fixing the radius of the sphere
  token CARTE= of cast logical
        Switch to treat endpoints as cartesian
  token LABEL= of cast character
        Labels of endpoints
  token ENDPT= of cast double and length 3
        Endpoints of symmetry lines
  token RDKPT= of cast logical
        Switch to read points from file

category PLOT
  token CARTV= of cast logical
        Switch to treat plot vectors as cartesian
  token ORIGIN= of cast double and length 3
        Origin of plot space
  token RPLOT1= of cast double and length 3
        Plot vector specifying plot space
  token RPLOT2= of cast double and length 3
        Plot vector specifying plot space
  token RPLOT3= of cast double and length 3
```

```
         Plot vector specifying plot space
   token NPDIV1= of cast integer
         Number of divisions of plot space
   token NPDIV2= of cast integer
         Number of divisions of plot space
   token NPDIV3= of cast integer
         Number of divisions of plot space
```

In addition, an output file is generated, which includes a full CTRL file, where all categories and tokens are given with their default values. This output looks like

```
 ASW-1.9, program MNSCF started on majestix at Wed, 13 Mar 2002, 08:38:28.

     Copyright (C) 1992-2002 Volker Eyert
     Please see file COPYRIGHT for details


 Write information to file HELP for HELP=T:

 SHOW=T lists default values for optional input.

 Echo CTRL file for SHOW=T:
HEADER
VERSION ASW-1.9
IO      HELP=T SHOW=T VERBOS=30 IACTIV=F CLEAN=T WRITE=CBAK EXTENS=
OPTIONS REL=T LSCPL=F NSPIN=1 AFSYM=F BEXT=0.0 XCPAR=VWN-VWN GGA= OVLCHK=T
        CCOR=T FULPOT=F CORDRD=F
STRUC   UNITS=BOHR ALAT=0.0 PLAT= 0.0  0.0  0.0
                                  0.0  0.0  0.0
                                  0.0  0.0  0.0
        SLAT= BBYA=1.0 BLAT=0.0 CBYA=1.0 CLAT=0.0 GAMMA=90.0
CLASS   NCLASS=1
        ATOM= Z= 0 R=0.0 R/RA=1.0 LMXL=2 LMXI=3 CONF=1 2 3 4
              QVAL= 0.0  0.0  0.0  0.0   MVAL= 0.0  0.0  0.0  0.0
SITE    NBAS=1 CARTP=T CHOUT=F
        ATOM= POS= 0.0  0.0  0.0 SPIN=UP
SYMGRP  GENPOS=F SYMOPS=                                              CARTR=T
        CARTT=T
PACK    FILLNG=1.0 OBYDMX=0.15 OBYRMX=0.4 ESBONS=0.05 RADMIN=0.5 RADMAX=5.0
        POTWIN=0.0 RADACC=1.0D-02 POSACC=1.0D-03
ENVEL   NKAP=1 EKAP=-0.015 EWPAR=2.44949 EWTOL=1.0D-10
BZSMP   NKABC=8 0 0 BZINT=SMS EMIN=-2.0 EMAX=1.0 NDOS=1000
        NORD=3 WIDTH=0.01 EFTOL=1.0D-05 SAVDOS=F SAVCOOP=F MSPLIT=F
        TEMPFD=300.0
CONTROL START= QUIT= FREE=F NITBND=20 CNVG=1.0D-08 CNVGET=1.0D-08
        NITATM=50 CNVGQA=1.0D-10
MIXING  NMIXB=5 BETAB=0.5 INCBB=F NMIXA=5 BETAA=0.5
SUPCELL ALAT=0.0 PLAT= 0.0  0.0  0.0
                       0.0  0.0  0.0
                       0.0  0.0  0.0
        SLAT= BBYA=1.0 BLAT=0.0 CBYA=1.0 CLAT=0.0 GAMMA=90.0 EQUIV=T CARTS=T
        PSHIFT= 0.0  0.0  0.0 CARTQ=T QSWAVE= 0.0  0.0  0.0
SYMLIN  NPAN=0 NPTS=400 ORBWGT=F SPATH=F EPHOT=0.0 CARTE=T
```

```
        LABEL= ENDPT= 0.0  0.0  0.0
PLOT    CARTV=T
        ORIGIN= 0.0  0.0  0.0
        RPLOT1= 0.0  0.0  0.0
        RPLOT2= 0.0  0.0  0.0
        RPLOT3= 0.0  0.0  0.0
        NPDIV1=0 NPDIV2=0 NPDIV3=0
 task  3, I/O     :  cpu time:     0.00000 sec

 Execution stopping for HELP=T.

 ASW-1.9, program MNSCF ended on majestix at Wed, 13 Mar 2002, 08:38:28.
```

### 3.4.9 The atomic files

Like most other files the atomic files are created automatically by the main program mnscf.run according to the entries ATOM= in categories CLASS and SITE of the CTRL file.

Each atomic file contains all information about the atom at hand. Similar to the CTRL file it falls in seven categories named GEN, MOMS, LDER, ESJH, VMTA, RHOV, and CORE. They comprise the following data.

GEN — Atomic number, angular momenta, valence state configuration, indicators for spin-polarized and relativistic calculations, information about the intraatomic radial mesh, core state orbitals, net atomic charge, Madelung potential, muffin-tin zero, and total energy contributions.

MOMS — Charges and energies as arising from the momentum analysis of the partial densities of states.

LDER — Logarithmic derivatives and P-functions for all partial waves.

ESJH — Hankel and Bessel energies and integrals for use in the intraatomic integrals entering the secular matrix.

VMTA — Spherical symmetric intraatomic single-particle potential.

RHOV — Spherical symmetric intraatomic valence electron charge density.

CORE — Core state energies and spherical symmetric intraatomic core electron charge density.

An example atomic file looks like.

```
GEN:  CU                                                written by ASW-1.9
 Z= 29  LMXL=2  LMXI=3  CONF= 4 4 3 4  NSPIN=1  REL=T
 RMAX=  2.669448  NR= 541  AR=  0.20000000D-01  BR=  0.544565163250D-04
 NCORE= 5  LCORE= 0 0 0 1 1  QCORE= 18
 NKAP=1  EKAP=   -0.015000
  QATM=  0.899994592740D-10
  VMAD=  0.000000000000D+00  VMTZ= -0.723947535848D+00
 VRMAX=  0.210033940992D+02
```

```
  EVBM=  0.113021354253D+01  ECBM=  0.470014259154D+00
 EFERM=  0.641841988344D+00
  EHRTR=   1438.20951104  EXCSM=   -130.04186989  VXCSM=   -172.17863463
  SUMTC=   3114.31998034  SUMEC=  -1905.37188135
  SUMTV=    186.76405917  SUMEV=     -3.44125736  3pV  =     -0.17025757
  3pV-T=  -3304.88587275  ETVAR=  -3304.88588502
MOMS:  CU
 N L          QMOM1             QMOM2             EMOM1             EMOM2
 4 0  0.4583953828D+00  0.2430151755D+00 -0.6119260083D+00 -0.2741818702D+00
 4 1  0.3397801707D+00  0.3913002460D+00 -0.5024522894D+00 -0.2199454608D+00
 3 2  0.3960311127D+01  0.5555928693D+01 -0.3865169875D+00 -0.2324652672D+00
 4 3  0.2285774676D-01  0.2841145833D-01 -0.3734102793D+00 -0.2291031680D+00
LDER:  CU
 N L          DMOM1             DMOM2             PMOM1             PMOM2
 4 0 -0.4157618407D+00 -0.1397855137D+01 -0.9818797857D+01 -0.1037459945D+02
 4 1  0.5204308714D+00  0.1637136965D-01 -0.5803326913D+01 -0.6266815400D+01
 3 2 -0.3854797793D+00 -0.1186359961D+02 -0.3679266193D+00 -0.1486703663D+01
 4 3  0.2608383567D+01  0.2479774246D+01  0.1204699823D+01  0.1187480604D+01
ESJH:  CU
 N L           EHAN              EBES              SHAN              SBES
 4 0 -0.2951559709D+00 -0.8072035319D+00  0.6513030651D+00  0.5940976548D+01
 4 1  0.5933872823D+00 -0.8269223615D+00  0.1711121267D+00  0.3107907593D+01
 3 2 -0.2868397473D+00 -0.1101618208D+01  0.3658993068D+01  0.8575664242D+00
 4 3  0.3436334774D+01 -0.8512466627D+00  0.1015912559D+01  0.7195771683D-01
```

A rather complex file check routine is invoked at the beginning of most of the main programs. During this check data from each atomic file is read in and compared to the information gained from previous reading of the CTRL file. Whenever inconsistencies arise the program will issue a message and respond adequately. For instance, if the lattice constant is changed in the CTRL file, the porgram will identify the inconsistency of the sphere radii. In the following, the radii will be automatically adjusted and the program enforces an intraatomic restart calculation in order to update the intraatomic integrals for the secular matrix. In contrast, if the program detects converged files but the CTRL files would require a start from scratch by the setting START=REN the program stops in order not to overwrite the atomic files. Finally, if a spin-polarized calculation finds converged spin-degenerate files, starting values for the intraatomic magnetic moments are imposed and the program enters a restart calculation. Please see Sec. 2.3 and Chap. 4 for more detailed information.

### 3.4.10 FREE

The information about the overlapping free atom charge densities and potentials as created by program mnfre.run go into this file. It is a formatted file, hence, it is written in ASCII.

### 3.4.11 STRU

This formatted file is written by programs mnfre.run and mnpac.run and contains the positions and muffin-tin radii of all atoms with the plot region specified in the CTRL file.

### 3.4.12 MIX

This file holds all information about the previous and the present band iterations to be used during the self-consistency cycle. It is automatically created by program mnscf.run and, after convergence has been achieved, it is deleted. However, in case you stop the program and restart after a change of orbitals as described in Sec. 2.2.2, this file must be deleted. Note that MIX is an unformatted file.

### 3.4.13 BNDE

The formatted file BNDE contains the band structure $E(\mathbf{k})$ as calculated as calculated by program mnbnd.run.

### 3.4.14 BNDV

If required according to token ORBWGT=T the eigenvectors are written to the unformatted file BNDV by program mnbnd.run.

### 3.4.15 DOS

The unformatted file DOS is written by program mnscf.run if token SAVDOS=T is set. It contains the partial DOS for later plotting.

### 3.4.16 COOP

For SAVCOOP=T and the respective settings COORB= program mnscf.run creates an unformatted file COOP, which holds all information about the partial crystal orbital overlap populations (COOP), the crystal orbital Hamiltonian population (COHP) or the covalence energy ($E_{cov}$).

### 3.4.17 out***

Naming of the output files is completely free. However, if you are using the shell scripts mn***.x output is written to file *out****, where the last three letters indicate the type of the program. In contrast, shellscript mnall.x writes output files *outlstnn*, where *nn* ranges from 6 to 30 according to the **k**-space grid.

### 3.4.18 bnd.tex, dos.tex, coop.tex

These LaTeXfiles are created by the plot programs plbnd.run, pldos.run, and plcop.run, respectively, if LaTeXoutput has been specified in the dialog.

### 3.4.19 free.ps, bnd.ps, dos.ps, coop.ps

postscript files created by the plot programs

These postscript files are created by the plot programs plfre.run, plbnd.run, pldos.run, and plcop.run, respectively, if Gnuplot postscript output has been specified in the dialog.

# Chapter 4

# The main input file: CTRL

In this chapter we describe the structure and content of the CTRL file, which is the main input file for the ASW package and, hence, the only interface between the user and the main programs. Besides information on the crystal structure and the atoms making the crystal it contains also more technical parameters controlling the execution. The CTRL file is fully free format, i.e. users do not have to obey any format specification. Yet, the file must not contain any TABs.

It is important to notice that the CTRL file is a exclusively used as an *input* file to the programs. None of the programs writes to this file at all. Only some shell scripts, like "upctr", are allowed to manipulate the CTRL file (see below). As a consequence, the CTRL file will never be destroyed even if you interrupt the program during execution or if your computer crashes. Whenever the program creates new data as e.g. empty sphere positions, atomic radii, or a supercell, it generates a new CTRL file called CNEW. In addition, most of the programs write an updated copy of CTRL to the file specified by the token WRITE=, which, by default, is CBAK.

The CTRL file is grouped into categories each starting in the first column of the file. Each category contains tokens specifying a portion of the input: e.g. in order to specify the lattice constant you just write ALAT=6.76 somewhere in the category STRUC. A complete list of all categories and tokens is supplied in the file HELP, which is printed in Sec. 3.4. It is included as a separate file in the ASW package, but could be generated running any of the main programs with HELP=T in category IO of the CTRL file.

At present, the ASW program package interprets 118 tokens grouped in 16 different categories. However, fortunetely only entries in three different categories must be specified by the user. For all other tokens exist meaningful and well tested default values. Information required from the user is related to the crystal's Bravais lattice, the type of atoms included (their atomic numbers), and the positions of these species within the unit cell of the crystal. These three types of information go into the categories STRUC, CLASS, and SITE, respectively, which therefore are marked mandatory in the list below.

Once you start the program, it automatically creates the atomic files which have just the names specified by token ATOM= in category CLASS. There is only one file for each atom. The program also creates (one) file for the mixing (file MIX) and deletes it after self-consistency has been reached. In addition, files DOS and

COOP (both unformatted) are created and written to in case you set SAVDOS=T and SAVCOOP=T, respectively in category BZSMP.

A short version of the following explicit description is given in the HELP file, which comes with the distribution but can be easily created by specifying HELP=T in category OPTIONS and running any of the ASW main programs. At the present stage you may just construct a CTRL file like

```
VERSION ASW-1.8
IO      HELP=T
```

and run an ASW program.

For most of the tokens listed below there exist default values, which have been thoroughly tested and tuned to optimal performance. For this reason, only very few information is actually needed to successfully run the ASW program. This includes atomic numbers and crystal structure data as lattice constants and atomic positions. The respective categories and tokens are indicated as mandatory below.

## 4.1 Category HEADER

This category is meant to hold a short description of the calculation. At present, this category is limited to 20 lines. Formally, the first line is treated differently from the following lines as it is interpreted as a title, which, by default, is appended to most plots. For this reason, I recommend to specify only a short title in the first line and to put longer comments into the following lines. Usually, I include references to important previous work on this compound in these lines.

## 4.2 Category VERSION

This category contains information about the version of the ASW package, which you are currently working with.

### 4.2.1 Token ASW- (cast real)

This token specifies the version and release number of the ASW program you are intending to work with. It consists of two one-digit numbers separated by a dot. For the present version the default is ASW-1.9. This token is useful if you want to use a CTRL file written by an elder version/release, where some tokens had been in other categories as they are in version 1.9. By specifying the correct version you can take advantage of the upward compatibility of the package.

## 4.3 Category IO

This category contains specifications about which and how much information the programs should write to output as well as to the atomic files.

### 4.3.1 Token HELP= (cast logical)

This logical switch allows to print out the HELP file as listed in Sec. 3.4, which comprises a short hand version of this chapter. Default is HELP=F. To create the HELP file, set HELP=T and run any of the main programs (those starting with "mn"). To be specific, create a CTRL file containing the lines

```
VERSION ASW-1.8
IO      HELP=T
```

and write "mnscf.x" at the operating system's prompt. In addition to the HELP file the program produces an output file (called outscf), which mainly contains a full CTRL file comprising all categories and tokens with their default values.

### 4.3.2 Token SHOW= (cast logical)

If set to true, this token makes the program echo the CTRL file to the output. This mode is highly recommended since it allows you to check the input for the calculation lateron. In addition, as an extra bonus, whenever you have lost the CTRL file you may easily recover it just by cutting the respective portion out of the output. For these reasons, the default is SHOW=T.

### 4.3.3 Token VERBOS= (cast integer)

Here you can specify a verbosity level for printing the output. The following verbosity levels are at present in use:

| | | | |
|---|---|---|---|
| verbosity | $\geq 0$ | : | nearly nothing is printed, |
| | $\geq 10$ | : | very terse, |
| | $\geq 20$ | : | terse, |
| | $\geq 30$ | : | normal, |
| | $\geq 40$ | : | verbose, |
| | $\geq 50$ | : | very verbose, |
| | $\geq 60$ | : | higher verbosity, |
| | $\geq 80$ | : | highest verbosity, |
| | $\geq 100$ | : | low-level debugging, |
| | $\geq 110$ | : | intermediate-level debugging, |
| | $\geq 120$ | : | high-level debugging. |

The default is VERBOS=30. In normal operation, you should not go beyond VERBOS=50. In particular, for VERBOS=80, the Hamiltonian matrix and related matrices are written to the output file, which, hence, will become very large.

### 4.3.4 Token IACTIV= (cast logical)

In future versions, this token will allow to switch to an interactive mode.

### 4.3.5 Token CLEAN= (cast logical)

Like the CTRL file the atomic files are organized in a collection of categories named GEN, MOMS, LDER, ESJH, VMTA, RHOV, and CORE. The first four of these comprise, respectively, general information about the atom at hand, the moments of the partial densities of states, the partial logarithmic derivatives, and the partial Hankel and Bessel energies and integrals. Finally, the last three, and by far most extensive, groups contain the effective single-particle potential as well as the self-consistent valence and core charge densities.

Since, within the standard ASW method, the latter three entities can be easily generated from the converged moments or the logarithmic derivatives, they can be removed in the atomic files, once a calculation has been completed. The token CLEAN=, which by default is CLEAN=T, makes the program delete the charge density and potential information, i.e. the categories VMTA, RHOV, and CORE, from the atomic file and thus allows for much reduced storage needs. Yet, see token CORDRD= below for exceptions.

### 4.3.6 Token WRITE= (cast character)

Here you can specify the name of an updated backup version of the CTRL file, which will be written just before completion. Default is WRITE=CBAK.

### 4.3.7 Token EXTENS= (cast character)

Using this token you may specify a standard extension appended to all file names created by the program as e.g. ".dat" or ".DAT". This is meant to conform with WINDOWS or VAX file name conventions. The default is empty, hence, all file names come without an extension.

## 4.4 Category OPTIONS

The following tokens can be used to specify details of the calculations.

### 4.4.1 Token REL= (cast logical)

This token allows to switch between non-relativistic and scalar-relativistic mode. The default is REL=T (scalar-relativistic).

### 4.4.2 Token LSCPL= (cast logical)

In future versions this token will enable for including the LS-coupling, which, however, is not yet implemented.

### 4.4.3 Token NSPIN= (cast integer)

Here you can specify the number of spin-channels. Default is NSPIN=1.

### 4.4.4 Token AFSYM= (cast logical)

In spin-polarized calculations, this token allows to use „spin-sublattice“ symmetry, which preserves the electronic states on going from one sublattice to the other and at the same time changing spins. This symmetry allows to perform the band calculation for one spin only and to extract the other spins results from the first ones. When AFSYM=T is specified, in addition, for each atom the „direction“ of spin has to be specified with the help of token SPIN= (see below). A more detailed description of how to perform spin-polarized calculations will be given in the respective section lateron.

### 4.4.5 Token BEXT= (cast double)

This token comprises an external magnetic field (in z-direction). The default value is BEXT=0.0. Note that use of finite magnetic fields has not yet been fully tested.

### 4.4.6 Token XCPAR= (cast character)

Since version ASW-1.7, token XCPAR= enables specification of different LDA schemes. Actually, almost all known parametrizations are included now. The desired LDA parametrization is specified in the form XCPAR=P-S, where

P denotes the parametrization by

KSG: Kohn, Sham, and Gaspar (exchange only),

GLW: Gunnarsson, Lundqvist, and Wilkins,

JMW: Janak, Moruzzi, and Williams,

VBH: von Barth and Hedin,

MJW: Moruzzi, Janak, and Williams,

PZ: Perdew and Zunger,

VWN: Vosko, Wilk, and Nusair,

PW: Perdew and Wang,

and

S denotes the spin interpolation by

GLW: Gunnarsson, Lundqvist, and Wilkins,

VBH: von Barth and Hedin,

VWN: Vosko, Wilk, and Nusair,

PW: Perdew and Wang.

Note that while the default was XCPAR=MJW-VBH in all versions up to ASW-1.6, it is now set to the more accurate XCPAR=VWN-VWN.

### 4.4.7 Token GGA= (cast character)

In version ASW-1.8, a new token, GGA= has been included, which allows specification of different schemes of the generalized gradient approximation (GGA). Possible GGA parametrizations are GGA=G where

G denotes the parametrization by

PW: Perdew and Wang 1991 (GGA-II),

EV: Engel and Vosko 1993,

PBE: Perdew, Burke, and Ernzerhof 1996.

The default is GGA= , which means to use none of these schemes.

### 4.4.8 Token OVLCHK= (cast logical)

In elder versions of the standard ASW method (and also the LMTO method) slight ambiguities came in by bad settings of the atomic sphere radii. Most of such inaccuracies come from too large overlap of the spheres. In particular for the ASW method, the situation has much improved by the invention of the sphere geometry optimization (SGO) algorthim, which allows to automatically evaluate the radii. However, it is still recommended to check the radii for too large overlap. This overlap check is invoked by the token OVLCHK=, which, by default, is OVLCHK=T.

### 4.4.9 Token CCOR= (cast logical)

This switch enforces use of the socalled combined correction to the ASA. Actually, the term combined correction is more related to the LMTO method since most early calculations were done with the pure ASA. For this reason, calculations employing the combined corrections, earned an extra bonus. In contrast, these corrections to the ASA were included in the ASW method in a very elegant manner from the very beginning and, hence, calculations without the combined corrections actually were never done. Hence, the default is CCOR=T.

### 4.4.10 Token FULPOT= (cast logical)

In future versions, this token will allow for full potential ASW calculations.

### 4.4.11 Token CORDRD= (cast logical)

For some instances as e.g. calculation of rather small energy differences, it might be necessary to freeze the core state during the self-consistency cycle. This is allowed for by setting CORDRD=T. Yet, the default is CORDRD=F. Note that in case of frozen core calculations the last of the aforementioned categories contained in the atomic files, i.e. the category CORE must not be deleted from the file after successful convergence. As a consequence, for CORDRD=T specifying CLEAN=T will remove only the categories VMTA and RHOV from the atomic files, but will preserve the category CORE.

## 4.5 Category STRUC (mandatory)

This category holds all the information about the Bravais lattice. Of course, its presence is necessary for any calculation.

### 4.5.1 Token UNITS= (cast character)

Here you can choose the units to be used for the following tokens, which may be atomic units (UNITS=BOHR, default) or Angstroems (UNITS=ANGS).

### 4.5.2 Token ALAT= (mandatory; cast double)

This token contains the lattice constant in the units specified by token UNITS=. The default is ALAT=0.0. As a consequence, if the lattice constant is not given a finite value the program aborts.

Actually, the tokens ALAT= and PLAT= below are complementary in that each of them can be scaled by a factor if the number(s) coming with the respective other token are scaled by the inverse. An equivalent statement holds for the tokens BBYA= and CBYA=, also given below. As a consequence, you may regard the input given by ALAT= as a length scale for the whole crystal used for the primitive translations.

### 4.5.3 Token PLAT= (mandatory; cast double; length 9)

Information about the Bravais lattice is covered by this token, which contains the three primitive translations $\mathbf{a}_i,\ i = 1, 2, 3$ in units of the lattice constant as specified by token ALAT=. All nine components must be given in the order $\mathbf{a}_{1x}$, $\mathbf{a}_{1y}$, $\mathbf{a}_{1z}$, $\mathbf{a}_{2x}$, $\mathbf{a}_{2y}$, $\mathbf{a}_{2z}$, $\mathbf{a}_{3x}$, $\mathbf{a}_{3y}$, $\mathbf{a}_{3z}$.

### 4.5.4 Token SLAT= (cast character)

As an alternative to explicitly specifying the primitive translations by PLAT= you may use the following six tokens, which rely on default lattice vectors coded in the ASW package. First of all, token SLAT= indicates a string for the Bravais lattice. Possible values are

| | |
|---|---|
| SC: | simple cubic |
| BCC: | body-centered cubic |
| FCC: | face-centered cubic |
| ST: | simple tetragonal |
| BCT: | body-centered tetragonal |
| SO: | simple orthorhombic |
| SCO: | base-centered orthorhombic |
| BCO: | body-centered orthorhombic |
| FCO: | face-centered orthorhombic |
| SM: | simple monoclinic |
| SCM: | base-centered monoclinic |

TRI: trigonal
HEX: hexagonal

Note that the primitive translations coming with these specifications correspond to the definitions of Bradley and Cracknell, which in some instances deviate from the definitions commonly used. For this reason, you are strongly urged to check the primitive translations, which are printed to output at the beginning of the programs execution.

### 4.5.5 Token BBYA= (cast double)

For the other lattice constants coming with non-cubic and non-trigonal systems there exist two different ways of specification. The token BBYA= fixes the ratio of the lattice constants ALAT and BLAT. As for ALAT= above the default is BBYA=0.0.

### 4.5.6 Token BLAT= (cast double)

Alternatively, the additional lattice constant can be specified by the token BLAT=, which gives the lattice constant B explicitly in the units specified by token UNITS=. Again the default is BLAT=0.0.

### 4.5.7 Token CBYA= (cast double)

This token comprises the ratio of the lattice constants C/A. The default is CBYA=0.0.

### 4.5.8 Token CLAT= (cast double)

Alternatively, the token CLAT= allows to specify the lattice constant C explicitly in the units specified by token UNITS=. Default is CLAT=0.0.

### 4.5.9 Token GAMMA= (cast double)

For monoclinic lattices the angle can be specified by the token GAMMA=. The default value is GAMMA=0.0.

## 4.6 Category CLASS (mandatory)

Once the Bravais lattice has been specified the information about the crystal structure is completed by giving the positions of the atoms within the unit cell as well as the species of atoms, which hold these sites. While the present category comprises all information about the atomic species, the atomic sites will be given in the following section of the CTRL file.

### 4.6.1 Token NCLASS= (cast integer)

The number of different atoms, which we call classes of atoms, can be specified in two different ways. The most comfortable and standard procdure consists of letting the programs count the number of ATOM= entries just below.

However, in the course of calculations it might be useful to exclude some of the classes listed below from consideration by the program. This can be most easily accomplished by shifting the corresponding ATOM= entries to the end of this category and reducing the number of classes, which are taken into account by the programs by setting NCLASS= to the reduced number of classes.

To summarize, if a number of classes is specified by a NCLASS=n entry the program tries to reads as many classes. Any subsequent ATOM= entries are ignored. On the other hand, if the are less ATOM= entries than specified by NCLASS=n, the program will warn for the missing classes and stop. The default is NCLASS=1.

### 4.6.2 Token ATOM= (mandatory; cast character)

Here you must label each of the classes. In the course of the calculations these labels are used to name the atomic files. My personal preference is to label classes by their names given in the periodic table extended by a roman number, whenever different classes with the same atomic number exist.

### 4.6.3 Token Z= (mandatory; cast integer)

This entry holds the atomic number. Note that the minimum input for each atom consists of giving a name by ATOM= and specifying the atomic number by Z=. All other input given in this category can be derived from these two tokens.

### 4.6.4 Token R= (cast double)

The atomic sphere radii can be specified in two different ways. In case you want to give absolute values, use this token. The default is R=0.0.

### 4.6.5 Token R/RA= (cast double)

As an alternative to the previous token relative atomic sphere radii may be chosen. In this case the radii will be scaled to the ASA-condition unless you specify a value for FILLNG= (see below) different from one. The default value is R/RA=1.0.

### 4.6.6 Token LMXL= (cast integer)

This token specifies for each atom the maximum angular momentum for the lower partial waves, which are taken into account in the secular matrix. The default values are derived from the previous input of the atomic number Z.

### 4.6.7 Token LMXI= (cast integer)

In addition, for each atom a maximum angular momentum for the intermediate waves is needed. These waves are used for expanding the Hankel envelope functions at neighbouring sites in Bessel envelope functions. As a default, the input for the lower wave maximum angular momentum plus one is used. Especially for small empty spheres, which carry a very small amount of charge already in their lower partial wave, it is useful to set the maximum $l$ value for the intermediate waves to that for the lower waves. To the contrary, only rare situations exist, where $l_{int}$ should be $l_{low} + 2$.

### 4.6.8 Token CONF= (cast integer; length 4)

While the previous two tokens specify the maximum number of partial waves per atoms, the principal quantum numbers for all these orbitals must be given here. Again, the default values are taken from a small database in accordance with the above input for the atomic number.

### 4.6.9 Token COORB= (cast integer)

With this token you can select the orbitals, which you want ot be included in the calculation as well as presentation of the crystal orbital overlap population (COOP), the crystal orbital Hamiltonian population (COHP) or the covalence energy ($E_{cov}$). The input consists just of the angular momenta of the desired orbital, i.e. for including $p$ and $d$ states you write COORB=1 2. As a default, no orbitals will be included.

### 4.6.10 Token QVAL= (cast double; length 4)

This token holds the valence charges for each atom, which will be needed if a calculation starts from scratch. As for the previous tokens, this input can be uniquely derived from the atomic number.

### 4.6.11 Token MVAL= (cast double; length 4)

Here you can give starting values for the magnetic moments of the valence states to be used for starting a spinpolarized calculation. These moments arise as the difference of the spin up and spin down partial charges. Reasonable default values for each atom are proposed by the program.

## 4.7 Category SITE (mandatory)

Finally, the structural information os completed by specifying the atomic sites in this category.

### 4.7.1 Token NBAS= (cast integer)

This token allows to specify the total number of atomic sites irrespective of the atomic species (class). The default value is 1. However, note that this token needs not to be given since the number of sites is counted by the program as the number of POS= entries (see below). Otherwise for token NBAS= the same holds as was already said for token NCLASS= above.

### 4.7.2 Token CARTP= (cast logical)

This switch specifies whether the following entries for the atomic positions are interpreted as Cartesian coordinates (CARTP=T) or as relative components in terms of the primitive translations (CARTP=F). The default value is CARTP=T.

### 4.7.3 Token CHOUT= (cast logical)

Sometimes it is useful to change the representation of the atomic positions from Cartesian coordinates to relative components or vice versa. By setting CHOUT=T you may enforce that the respective other represnetation is written to the backup copy of the CTRL file (see token WRITE= above). The default is CHOUT=F.

### 4.7.4 Token ATOM= (mandatory; cast character)

Before entering the atomic positions you must for each site give the class label in order to attach entries in categories CLASS and SITE to each other. Hence, the present token must match one of the ATOM= entries given in category CLASS.

### 4.7.5 Token POS= (mandatory; cast double; length 3)

Here you must specify the atomic position in the representation selected by the above token CARTP=.

### 4.7.6 Token SPIN= (cast character)

In case of antiferromagnetic order the site information may be complemented with information about the sublattice (spin up or down), the respective atom belongs to. This can be done by setting SPIN=UP or SPIN=DN for each atom. Obviously, the number of UP and DN entries must be equal for each class. Note that these entries are used only if AFSYM=T is given in category OPTIONS.

## 4.8 Category SYMGRP

Information about the analysis and use of space group symmetry underlying the crystal structure is supplied by this category. There exist essentially two modes. Either the symmetry of the crystal may be analyzed from the above information about the atomic classes and sites. This is the default. However, for large systems

it may be useful to enter only a minimal set of atomic sites and to set up the remaining sites from the given symmetry operations. Finally, in case that both the complete set of atomic sites and the generators for all symmetry operations are given both are checked for consistency.

### 4.8.1 Token GENPOS= (cast logical)

This switch indicates that the basis set of atomic positions is completed by using the minimal set of positions given in category SITE and the following generators of the full group of symmetry operations. The default is GENPOS=F.

### 4.8.2 Token SYMOPS= (cast character; length 48)

Here you may specify generators for the full group of the symmetry operations. These generators are most easily coded by using the following short hand notation: 'E' and 'I' denote unity operation and inversion, while 'R', 'S' and 'M' are short for proper and improper rotation, respectively and reflection. Finally ':T' stands for a fractional translation following any of the just mentioned operations. 'R' must be followed by an integer denoting the angle of rotation as the number of division of the full circle. Moreover, 'Ri', 'M' and ':T' must be followed by a vector '(a,b,c)' yielding the rotation axis, the vector perpendicular to the mirror plane and the translation vector, respectively. The components a, b, and c of the vectors must be real numbers, exceptions being the abbreviations 'C' and 'S' for 0.5 and 0.5*sqrt(3), respectively, and '0', '1', '1/2', '1/3', '1/4', and '1/6'. In addition, the following short hand notations exist. While 'D' is short for the diagonal, i.e. the vector (1,1,1), 'X', 'Y', and 'Z' abbreviate the vectors (1,0,0), (0,1,0), and (0,0,1). The following examples will help understanding this notation.

| | |
|---|---|
| R6(1,0,0) | a positive rotation by 60° about the x-axis, |
| R6(-1,0,0) | a positive rotation by 60° about the -x-axis, i.e. a negative rotation by 60° about the x-axis, |
| R6(C,S,0) | a positive rotation by 60° about the axis (0.5,0.8660,0), |
| R3D | a positive rotation by 120° about the (1,1,1)-axis, |
| R4(1,1,0):T(0,0,0.5) | a positive rotation by 90° about the (1,1,0)-axis followed by a translation by the vector (0,0,0.5), |
| MX | a reflection about the plane spanned by the y- and z-axis, which transforms x to -x, |
| I | the inversion. |

The operations R4X, MX, and R3D are sufficient to generate all 48 elements of the full cubic group. Finally, all the aforementioned operations may be combined by *, where A*B is converted to mat(A)*mat(B), i.e. the operations are applied from right to left. Note the originally 4x4-character of all operations when combining them.

### 4.8.3 Token CARTR= (cast logical)

This switch specifies whether the entries for the rotation axes are interpreted as Cartesian coordinates (CARTR=T) or as relative components in terms of the primitive translations (CARTR=F). The default value is CARTR=T.

### 4.8.4 Token CARTT= (cast logical)

This switch specifies whether the entries for the fractional translations are interpreted as Cartesian coordinates (CARTT=T) or as relative components in terms of the primitive translations (CARTT=F). The default value is CARTT=T.

## 4.9 Category PACK

This category summarizes all the options used by the sphere geometry optimization (SGO) algorithm. In the atomic sphere approximation (ASA) space is divided in spheres centered at the atomic sites. These spheres are required to fill space completely in order that the region between the spheres, the socalled interstitial region, vanishes. However, there exist crystal structures, where the aforementioned condition would lead to rather large overlaps of the atomic spheres. In that case, usually socalled empty spheres are included, which model the crystal potential in such large voids. The SGO algorithm automatically finds optimal empty spheres positions and the best radii of all, physical and empty spheres.

### 4.9.1 Token FILLNG= (cast double)

This token specifies the fraction of space, which eventually must be filled by the atomic spheres. Default is FILLNG=1.0. The special setting FILLNG=0.0 enforces creation of touching spheres.

### 4.9.2 Token OBYDMX= (cast double)

Maximum allowed overlap for any pair of spheres scaled to the distance between the sphere centers. A good value is OBYDMX=0.15, which is the default. For some systems, it might be necessary to go beyond this value in order to achieve complete space filling. However, you should not go beyond OBYDMX=0.2.

### 4.9.3 Token OBYRMX= (cast double)

Maximum allowed overlap for any pair of spheres scaled to the radii of the respective spheres. Default value is OBYRMX=0.4 but OBYRMX=0.5 is still acceptable. This token is important for systems containing large differences in sphere radii, where, hence, small spheres might be swallowed by large spheres if e.g. touching spheres are blown up to space filling.

### 4.9.4 Token ESBONS= (cast double)

Since the empty spheres are included in order to mimic the full crystal potential in large voids between the physcial spheres the potential within these spheres is rather flat. For that reason, empty spheres can have a larger overlap with the neighbouring spheres. ESBONS specifies this extra bonus to be to OBYDMX and OBYRMX. Default is ESBONS=0.05 and the actual value should not exceed ESBONS=0.1

### 4.9.5 Token NCEMAX= (cast integer)

This token fixes the maximum number of empty sphere classes sought for by the program. If space filling can not be achieved with less empty spheres the program issues a warning saying that the result of the sphere packing might depend on this parameter. The default value is 256.

### 4.9.6 Token RADMIN= (cast double)

This token gives the minimum muffin-tin radius for an empty sphere. Once the SGO algorithm subsequently identifies two possible empty spheres candidates with muffin-tin radii smaller than RADMIN the packing routine stops. Default value is RADMIN=0.5, which hardly has to be changed.

### 4.9.7 Token RADMAX= (cast double)

The token RADMAX= fixes the maximum atomic sphere radius for an empty sphere. This setting has two opposing consequences. A large radius for an empty sphere might be prohibitive since higher and higher partial waves (angular momenta) would be needed to properly describe the electronic wave function within such a large sphere. To the contrary, if the maximum radius is set to a rather small value, it might be difficult to fill space by spheres if large voids exist in the crystal structure. The default value is RADMAX=5.0, which is quite high in order not to limit the sphere size. For most crystal structures you will not need to change this value. Yet, in case you obtain large empty spheres, you might try RADMAX=3.0.

### 4.9.8 Token POTWIN= (cast double)

As it stands the SGO algorithm aims at an optimal modeling of the full crystal potential by slightly overlapping spherical potential wells. As a starting point for the full potential the overlapping free atom potential is used in a way first proposed by Mattheiss. This potential is a continous function in all space. If the potential is scanned along paths between the atomic sites the muffin-tin radii are related to the maxima of the potential along these paths. If overlapping spheres are sought for their radii correspond to positions on these paths, where the potential is by a certain amount lower than the maximum. As a consequence, the deviation from the mximum potential along a path can be used to define the overlap region of two spheres. This is done by token POTWIN=. The default value is POTWIN=0.0. Note that this token is kept mainly for consistency reasons.

### 4.9.9 Token RADACC= (cast double)

Here you can specify the accuracy required for the sphere radii. The default value of RADACC=0.02 hardly needs a change.

### 4.9.10 Token POSACC= (cast double)

This token gives the accuracy required for the empty sphere positions. As before, the default value of POSACC=0.003 usually is sufficient.

## 4.10 Category ENVEL

The ASW method got its name from the way basis functions for a subsequent variational procedure are constructed. To be specific, we start out from analytically known solutions of Schrödinger's equation with a constant potential, which are either plane waves or spherical waves. In the ASW method, only the latter used in the form of spherical Hankel functions. Within the atomic spheres, these spherical waves are replaced, or "augmented", by the solutions of Schrödinger's equation with the actual potential.

The present category specifies the analytical solutions outside the atomic spheres, which are the socalled envelope functions. The region outside the spheres is called the interstitial region. The tokens in this category usually need not be changed from their default value. Hence, in most cases you can ignore this category at all.

### 4.10.1 Token NKAP= (cast integer)

Here the number of envelope functions, i.e. the number of analytical solutions outside the atomic spheres per atom and angular momentum are fixed. The default is NKAP=1.

### 4.10.2 Token EKAP= (cast double)

EKAP= is used to specify the energies of the solutions of Schrödinger's equation in the interstitial region. The default value for the (first) solution is EKAP=-0.015, which is the classical ASW value.

### 4.10.3 Token EWPAR= (cast double)

The Ewald parameter is used to calculate the expansion of spherical Hankel functions centered at a specific atomic site in terms of spherical Bessel functions centered at a different site. Following a seminal method first proposed by the Ewald the resulting lattice sums are cut into two, the cutoff being the Ewald parameter. The default value is EWPAR=2.44949.

### 4.10.4 Token EWTOL= (cast double)

Using the token EWTOL you can specify the precision sought for the convergence of the aforementioned lattice sums. The default is EWTOL=1.0D-10.

## 4.11 Category BZSMP

The category BZSMP holds all information related to the Brillouin zone sampling.

### 4.11.1 Token NKABC= (cast integer; length 3)

This token specifies the numbers of mesh points to be taken along each of the reciprocal space primitive translations. As such it determines the **k**-space grid used for the Brillouin zone integrations. For examples, with NKABC=6 6 6, the Brillouin zone is divided into 216 microcells. A special notation has been invented for this token, which makes life much easier. Whenever one of the three numbers is set to 0 in the CTRL file, it is interpreted by the programs as being identical to the preceding number. To be specific, the following interpretations can be used

NKABC=6 0 0 ≡ NKABC=6 6 6
NKABC=6 8 0 ≡ NKABC=6 8 8
NKABC=6 0 8 ≡ NKABC=6 6 8

This feature enables fast switching to a finer grid after a calculation with a coarse grid has converged. The default setting for this token is NKABC=8 0 0. However, in most cases it is best to start from NKABC=6 0 0. After convergence, one should switch to NKABC=8 0 0, then 12, 16, 20, and 30. This way the dependence of the results on the fineness of the **k**-grid could be checked.

For metals, going up to NKABC=30 0 0 suffices in almost all cases, except for the nearly free electron metals like Al and Na. In contrast, for semiconductors and insulators much lower values as e.g. NKABC=16 0 0 are enough. The same holds for rather large unit cells comprising e.g. 30 or more atoms. Since these unit cells have smaller Brillouin zones going up to NKABC=16 0 0 will also suffice. Of course, these numbers are only rough estimates the particular choice being more or less a matter of the desired accuracy.

### 4.11.2 Token BZINT= (cast character)

The token BZINT= allows to choose the scheme used for the Brillouin zone integration. At present, it can be either BZINT=SMS (simple sampling), which is the default, or BZINT=HPS (high-precision sampling). The former scheme uses a simple histogram technique. For this reason it is highly efficient, however, due to its statistical nature the calculated densities of states look rather noisy. Still, this scheme is a very good choice as long as energy integrated quantities are aimed at as is the case during the iterations to self-consistency. After full self-consistency is achieved it is recommended to use the high-precision sampling as proposed by

Methfessel and Paxton [15], which produces very nice results. Future versions of the program will also include the linear tetrahedron method choosen by BZINT=LTM.

### 4.11.3 Token EMIN= (cast double)

The tokens EMIN= and EMAX= fix the energy interval used for the calculation of the (partial) densities of states. Note that this interval is used not only for visualization purposes but also during the iterations towards self-consistency, where the ntegrated DOS is needed for calculating the electronic charge density. For this reason EMIN and EMAX must be below the lowest valence band and higher than the Fermi energy, respectively. The default value is EMIN=-2.0 (Ryd).

### 4.11.4 Token EMAX= (cast double)

Token EMAX= fixes the upper bound of the interval used for the densities of states calculation. It must be above the Fermi energy. The default value is EMAX=1.0 (Ryd).

### 4.11.5 Token NDOS= (cast integer)

Token NDOS= gives the number of divisions of the [EMIN:EMAX]. Choosing a small value will result in inaccuracies of the calculated densities of states. However, a too high value will produces some statistical noise in the simple sampling scheme. A good choice is NDOS=nnnn, where nnnn is 1000 times the width of the energy interval [EMIN:EMAX] in eV. The default is NDOS=3000.

### 4.11.6 Token NORD= (cast integer)

In case that BZINT=HPS has been specified the following two tokens are needed:

Token NORD= fixes the order of the approximant in the high-precision Brillouin zome scheme of Methfessel and Paxton. While low values might cause inaccuracies, high values might lead to too sharp structures. A good choice is NORD=3, which is the default.

### 4.11.7 Token WIDTH= (cast double)

This token specifies the internal broadening used for the high-precision Brillouin zome scheme. The default value is WIDTH=0.01 (Ryd).

### 4.11.8 Token EFTOL= (cast double)

In case that BZINT=LTM has been specified the following token is needed in future versions:

Token EFTOL= gives the desired accuracy for the calculation of the Fermi energy.

### 4.11.9 Token SAVDOS= (cast logical)

Since the calculation of the partial ($m$-resolved) densities of states (DOS) requires additional CPU time, token SAVDOS= has been implemented to switch of calculation of all partial DOS during iteration towards self-consistency. For this reason the default is SAVDOS=F. Once full self-consistency is achieved you can choose SAVDOS=T and calculate the partial densities of states for plotting in one additional calculation. It is advantageous to switch from BZINT=SMS to BZINT=HPS in this case, too.

### 4.11.10 Token SAVCOOP= (cast logical)

The meaning of this token is equivalent to that of the previous token. During the iterations SAVCOOP=F should be used in order to suppress calculation of the crystal orbital overlap populations (COOP), the crystal orbital Hamiltonian population (COHP) or the covalence energy ($E_{cov}$) at all. The input consists just of the angular momenta of Only in the additional run after self-consistency has been reached, SAVCOOP=T might be used. Note that the latter choice is only effective if the orbitals to be included have been specified by tokens COORB= in category CLASS.

### 4.11.11 Token CTYPE= (cast character)

The token CTYPE= allows to choose the scheme used to describe chemical bonding. At present, it can be either CTYPE=COOP for the crystal orbital overlap population introduced by Hoffmann, CTYPE=COHP for the crystal orbital Hamilton population as proposed by Dronskowski and Blöchl, and CTYPE=ECOV for the covalence energy put forward by Fähnle and coworkers.

### 4.11.12 Token MSPLIT= (cast logical)

This token allows to calculated $m$-resolved COOP's. However, this feature has not been fully tested and, in addition, requires additional CPU time and memory. For these reasons, the default is MSPLIT=F.

### 4.11.13 Token TEMPFD= (cast double)

With this token you can specify the temperature (in Kelvin) entering the Fermi-Dirac distribution function, which will be used for folding with the calculated density of states. This allows to evaluate the position of the chemical potential from the condition that at the given temperature the numbers of excited holes and electrons must be identical. In a second step these numbers are resolved in contributions from atoms and angular momenta, i.e. the partial numbers of excited holes and electrons are calculated. This helps to quickly identify the relevant optical transitions. The default is TEMPFD=300 (K).

# 4.12 Category CONTROL

## 4.12.1 Token START= (cast character)

The tokens START= and QUIT= can be used to control the sequence of all calculations performed during execution esepcially of program mnscf.run. In general, ASW calculations towards self-consistency are subdivided into two parts. In one part, starting from integrals of the partial waves, the program sets up as well as diagonalizes the secular matrix for every **k**-point and supplies the first four moments of the partial densities of states. This part is called the band calculation. In the other part, the moments of the partial densities of states are used to calculate intraatomic wave functions, charge densities, and, using the notions of density functional theory, the intraatomic potentials. The latter in turn allow evaluation of the intraatomic integrals for use in the secular matrix. This part is called the atomic part. At start up of the program, guessed moments of the partial DOS are used to boot execution with the atomic calculations. Finally, since the moments are stored in the atomic files, they may be used to *restart* a calculations, again beginning with the intraatomic part. To conclude, four different starting points of the programs execution can be distinguished, which may be selected by token START= but, of course, also depend on the information gained in previous executions.

START=REN Execution starts from scratch. The program performs a renormalized atom calculation using the charges as specified by QVAL= and magnetic moments as given by MVAL=, respectively. In contrast, if FREE=T has been given, a free atom calculation is performed using a sphere radius of 20 Bohr radii.

START=RST In this case the program does a restart calculation, i.e. a renormalized atom calculation with the partial charges and logarithmic derivatives as specified in the atomic files.

START=ATM The program invokes one (or more) full band iterations starting with the intraatomic calculations. The number of band iterations to be done is specified by token NITBND=.

START=BND This is the default. The program invokes one or more full band iterations starting with the Brillouin zone sampling.

In case START= has not been specified the default START=BND will be used. Furthermore the program will look up the atomic files and check them if present. Depending on the outcome of this check the program will then decide automatically where to start. In addition, if START= has been specified this setting is compared to the content of the atomic files and the program might opt for a different starting point. If, for instance, the atomic sphere radii have been changed in the CTRL-file and, hence, are no longer in agreement with those given in the atomic files the program will automatically invoke a restart calculation. The same happens when a change from a non-spinpolarized to a spinpolarized calculation has been detected. Another case to be mentioned is the situation where START=REN is specified and

converged atomic files exist. In this case the program stops in order to prevent the converged atomic files from being overwritten.

### 4.12.2 Token QUIT= (cast character)

Token START= is complemented by token QUIT=, which allows to stop execution prior to the end of the calculation normally performed by the program. To be specific, at present the following entries could be used.

QUIT=SYM Any of the main programs stops after having performed the symmetry analysis.

QUIT=OVL Any of the main programs stops after having performed the symmetry analysis and the overlap check for the atomic spheres.

QUIT=REN Program mnscf.run stops after having finished a free atom calculation.

QUIT=ATM Program mnscf.run stops after having finished the intraatomic atom calculations for the first time.

QUIT=BND Program mnscf.run stops after having finished the band calculations for the first time.

QUIT=DOS Program mnscf.run stops after having finished the band calculations for the first time but before calculating new moments of the partial densities of states. This setting is particularly useful for calculating the partial densities of states in a single run after self-consistency has been achieved.

### 4.12.3 Token FREE= (cast logical)

This token has two different meanings. The default is FREE=F. First, if program mnscf.run is called with FREE=T is performs a free atom calculation for all atoms in the unit cell. Second, if used with program mnbnd.run, FREE=T enforces calculation of the free electron bands.

### 4.12.4 Token NITBND= (cast integer)

This token fixes the maximum number of iterations to be done by the program. The default value is NITBND=20. This token is useful if only a limited amount of CPU time is available. A good estimate for the total execution time can be evaluated as follows. For most applications, the band part of the calculations takes about 95% of the total execution time. Once the band calculation, i.e. the loop over all **k**-points has started, the program checks the CPU time per **k**-point and, on adding the CPU time needed for the previous atomic calculations, prints out the CPU time estimate for one iteration.

### 4.12.5 Token CNVG= (cast double)

With this token the desired accuracy of the calculation is fixed. Once this value has achieved, the self-consistency cycle stops. In particular, CNVG= is the accuracy for the root mean square difference of the intraatomic charge density resulting from the previous and the present iteration. The default value is CNVG=1.0D-8, which is *very* accurate. You might well switch to CNVG=1.0D-6 without loosing to much. However, note that going from $10^{-6}$ to $10^{-8}$ usually requires only very few iterations and, hence, can be achieved at a very low cost.

### 4.12.6 Token CNVGET= (cast double)

This token specifies the desired accuracy for the total energy. Whenever the difference between the total energies of the previous and the present iteration is below this value, and the condition set by CNVG= is fulfilled, the self-consistency cycly is stopped. The default is CNVGET=1.0D-8 (Ryd), which again is *very* accurate.

### 4.12.7 Token NITATM= (cast integer)

Actually, there exist two different self-consistency cycles within the ASW main program mnscf.run. The *outer* cycle, also called the band iteration, is the usual one, which aims at making the whole electronic charge density, the effective potential, the Hamiltonian matrix etc. self-consistent. In addition, the intratomic calculations, which start from moments of the densities of states and make available the intraatomic matrix elements entering the Hamiltonian matrix, run through an intraatomic self-consistency cycle. During this intraatomic iteration, the intraatomic charge density and potential are made self-consistent while preserving the moments of the partial DOS. Note that the following band iteration might result in different moments, hence, intraatomic self-consistency does not mean full self-consistency of the whole crystal.

In contrast to token NITBND= above token NITATM= specifies the number of intraatomic iterations. The default value is NITATM=50, which hardly needs a change. Note that the contribution of the intraatomic calculations to the total execution time is negligible.

### 4.12.8 Token CNVGQA= (cast double)

This token fixes the desired accuracy of the intraatomic calculations. Once this value has achieved, the intraatomic self-consistency cycle stops. In particular, CNVGQA= is the accuracy for the root mean square difference of the intraatomic charge density resulting from the previous and the present intraatomic iteration. The default value is CNVG=1.0D-10, which is *very* accurate. However, due to the very low execution time of the intraatomic calculations and the speedup of the expensive band iterations by accurate atoms, this value is well justified and needs no changes.

## 4.13 Category MIXING

This category can be used to influence the iteration towards self-consistency. Actually, there exist two different self-consistency loops. One, which is called the band iteration, affects the calculation of partial charges and logarithmic derivatives for each orbtial once the Brillouin zone sampling has been finished. The other, intraatomic self-consistency loop is performed for each atom as part of the intraatomic calculations. All this is explained in more detail in my recent overview over the ASW method [5]. Both types of iterations are accelerated by using the extended Anderson mixing, which superseeds both the elder versions of the Anderson mixing and the Broyden update [2]. The extended Anderson mixing is controlled by two parameters, namely the number of previous iterations to be mixed in and the mixing parameter $\beta$.

The values proposed for all tokens in this categories have been thoroughly tested in a huge number of applications. They hardly need a change. However, whenever convergence is rather slow, you might reduces the $\beta$ value used for the band iterations to BETAB=0.3 and switch of the automatic increase of $\beta$ by setting INCBB=F.

### 4.13.1 Token NMIXB= (cast integer)

This token holds the number of previous band iterations to be used for the band mixing. The default is NMIXB=5.

### 4.13.2 Token BETAB= (cast double)

Here you can specify the mixing parameter to be used for the band mixing. The default is BETAB=0.5.

### 4.13.3 Token INCBB= (cast logical)

With INCBB=T this switch allows to increase the $\beta$ value used for the band iterations, hence, to speedup these iteration, once the self-consistency loop has stabilized.

### 4.13.4 Token NMIXA= (cast integer)

This token holds the number of previous intraatomic iterations to be used for the atomic mixing. The default is NMIXA=5.

### 4.13.5 Token BETAA= (cast double)

Here you can specify the mixing parameter to be used for the intraatomic mixing. The default is BETAA=0.5.

# 4.14 Category SUPCELL

## 4.14.1 Token ALAT= (cast double)

This token contains the supercell lattice constant in the units specified by token UNITS=. Default is ALAT=0.0.

## 4.14.2 Token PLAT= (cast double; length 9)

Information about the Bravais lattice is covered by this token, which contains the three supercell primitive translations $\mathbf{a}_i,\ i = 1, 2, 3$ in units of the lattice constant. All nine components must be given in the order $\mathbf{a}_{1x}$, $\mathbf{a}_{1y}$, $\mathbf{a}_{1z}$, $\mathbf{a}_{2x}$, $\mathbf{a}_{2y}$, $\mathbf{a}_{2z}$, $\mathbf{a}_{3x}$, $\mathbf{a}_{3y}$, $\mathbf{a}_{3z}$.

## 4.14.3 Token SLAT= (cast character)

As an alternative to explicitly specifying the supercell primitive translations by PLAT= you may use the following six tokens, which rely on default lattice vectors coded in the ASW package. First of all, token SLAT= indicates a string for the Bravais lattice. Possible values are

| | |
|---|---|
| SC: | simple cubic |
| BCC: | body-centered cubic |
| FCC: | face-centered cubic |
| ST: | simple tetragonal |
| BCT: | body-centered tetragonal |
| SO: | simple orthorhombic |
| SCO: | base-centered orthorhombic |
| BCO: | body-centered orthorhombic |
| FCO: | face-centered orthorhombic |
| SM: | simple monoclinic |
| SCM: | base-centered monoclinic |
| TRI: | trigonal |
| HEX: | hexagonal |

Note, however, that the primitive translations coming with these specifications correspond to the definitions of Bradley and Cracknell, which in some instances deviate from the definitions commonly used. For this reason, you are strongly urged to check the primitive translations, which are printed to output at the beginning of the programs execution.

## 4.14.4 Token BBYA= (cast double)

This token comprises the ratio of the supercell lattice constants B/A. Default is BBYA=0.0.

### 4.14.5 Token BLAT= (cast double)

Alternatively, this token allows to specify the supercell lattice constant B explicitly in the units specified by token UNITS=. Default is BLAT=0.0.

### 4.14.6 Token CBYA= (cast double)

This token comprises the ratio of the supercell lattice constants C/A. Default is CBYA=0.0.

### 4.14.7 Token CLAT= (cast double)

Alternatively, this token allows to specify the supercell lattice constant C explicitly in the units specified by token UNITS=. Default is CLAT=0.0.

### 4.14.8 Token GAMMA= (cast double)

Here you insert the angle to be used for the monoclinic lattices. Default value is GAMMA=0.0.

### 4.14.9 Token EQUIV= (cast logical)

This switch enforces that after construction of the supercell atoms located at the equivalent positions in different subcells still belong to the same class even if they are not related by a symmetry operation of the supercell. The default is EQUIV=T.

### 4.14.10 Token CARTS= (cast logical)

This switch specifies whether the following entries for the shift vector are interpreted as Cartesian coordinates (CARTS=T) or as relative components in terms of the primitive translations (CARTS=F). The default value is CARTS=T.

### 4.14.11 Token PSHIFT= (cast double; length 3)

This token holds a shift vector, which is added to all atomic positions after a supercell has been constructed. Note, however, that in order to avoid confusion this token is interpreted only by routine mnscl. The default is PSHIFT= 0.0 0.0 0.0.

### 4.14.12 Token CARTQ= (cast logical)

This switch specifies whether the following entries for the spin wave vector are interpreted as Cartesian coordinates (CARTQ=T) or as relative components in terms of the primitive translations (CARTQ=F). The default value is CARTQ=T.

### 4.14.13 Token QSWAVE= (cast double; length 3)

With this token you may specify a spin wave vector for a (collinear) antiferromagnetic structure to be imprinted on the supercell. The simplest case is that of doubling the unit cell and treating the two subcells as the spin up and spin down sublattice. The spin wave vector must be given in units of $2\pi$ divided by the supercell lattice constant $a$; its default is QSWAVE= 0.0 0.0 0.0.

## 4.15 Category SYMLIN

This category comprises all the information needed for plotting the band structure.

### 4.15.1 Token NPAN= (cast integer)

This token holds the number of panels, which is the number of different symmetry lines in the Brillouin zone, along which the band structure is represented. The default is NPAN=0. However, note that for this token the same as for tokens NCLASS= and NBAS= holds. The number of symmetry lines will be calculated from the following entries, but token NPAN= can be used to hide additional entries from the program.

### 4.15.2 Token NPTS= (cast integer)

Here you can give the maximum number of points to be used for the band structure plot. The actual number will be calculated by the program and usually will be slightly less than the input given by this token. The default is NPTS=400. However, in case you choose to calculate orbital projected band structures (see below), you should use NPTS=200.

### 4.15.3 Token ORBWGT= (cast logical)

This switch allows to plot orbital projected band structures. If ORBWGT=T is selected the program will print out the eigenvectors for each $\mathbf{k}$ point in addition to the eigenvalues. This allows to evaluate the contribution from each orbital of the basis set to the wave function at each $\mathbf{k}$ point and for each band, which may be represented as the width of a bar attached to each $E(\mathbf{k})$ in the plot. The default is ORBWGT=F.

### 4.15.4 Token CARTE= (cast logical)

This switch specifies whether the following entries for the end points of the symmetry lines are interpreted as Cartesian coordinates (CARTE=T) or as relative components in terms of the primitive translations (CARTE=F). The default value is CARTE=T.

### 4.15.5 Token LABEL= (cast character)

There exists standard notation for labelling the high symmetry points of the first Brillouin zone. These labels are usually appended to the band structure plots.

### 4.15.6 Token ENDPT= (cast double; length 3)

This token holds the end points of the symmetry lines, along which the band structure should be plotted.

## 4.16 Category PLOT

The last category contains information to be used for 3D plots of e.g. the crystal structure, the charge density or the potential. This comprises a set of vectors confining the plot space (a line, a plane or a volume) as well as the divisions along these vectors.

### 4.16.1 Token CARTV= (cast logical)

This switch specifies whether the following entries for the plot vectors are interpreted as Cartesian coordinates (CARTV=T) or as relative components in terms of the primitive translations (CARTV=F). The default value is CARTV=T.

### 4.16.2 Token ORIGIN= (cast double; length 3)

This token is useful, if the origin of the plotting volume deviates from the origin of the unit cell as specified by token PLAT= in category STRUC. Default is ORIGIN= 0.0 0.0 0.0.

### 4.16.3 Token RPLOT1= (cast double; length 3)

First plot vector specifying the plot space. Default is RPLOT1= 0.0 0.0 0.0.

### 4.16.4 Token RPLOT2= (cast double; length 3)

Second plot vector specifying the plot space. Default is RPLOT2= 0.0 0.0 0.0.

### 4.16.5 Token RPLOT3= (cast double; length 3)

Third plot vector specifying the plot space. Default is RPLOT3= 0.0 0.0 0.0.

### 4.16.6 Token NPDIV1= (cast integer)

Number of divisions along the first plot vector. Default is NPDIV1=50.

### 4.16.7 Token NPDIV2= (cast integer)

Number of divisions along the first plot vector. Default is NPDIV2=50.

### 4.16.8 Token NPDIV3= (cast integer)

Number of divisions along the first plot vector. Default is NPDIV3=50.

# Chapter 5

# The ASW database

Over the years I have many people using the ASW method have contributed to what is now called the ASW database. It consists of a large number of CTRL files for various elemental systems and compounds. The complete list is given below.

```
List of CTRL files contained in the ASW database:
(status as of 11.03.2002:  556 entries)
```

| | | |
|---|---|---|
| CTRL_Ag | CTRL_K2CuF4scoaf | CTRL_SmSe |
| CTRL_AgBr | CTRL_K2CuF4scod05 | CTRL_SmTe |
| CTRL_AgEr | CTRL_K2CuF4scod05af | CTRL_Sn |
| CTRL_AgI | CTRL_K2CuF4scod10 | CTRL_Sn3Sb2S6 |
| CTRL_Al | CTRL_K2CuF4scod10af | CTRL_SnO2 |
| CTRL_Al2O3+Nb8 | CTRL_K2CuF4scod15 | CTRL_SnS |
| CTRL_AlAs | CTRL_K2CuF4scod15af | CTRL_SnSbO2 |
| CTRL_AlMo4S4fcc | CTRL_K2CuF4scod17 | CTRL_Sr |
| CTRL_AlMo4S4tri | CTRL_K2CuF4scod17af | CTRL_Sr2RuO4 |
| CTRL_AlMo4S4trife | CTRL_K2CuF4scod20 | CTRL_Sr2TiO4 |
| CTRL_AlMo4S4triid | CTRL_K2CuF4scod20af | CTRL_Sr2VO4 |
| CTRL_AlP | CTRL_K2CuF4scod25 | CTRL_Sr6NGa5 |
| CTRL_AlSb | CTRL_K2CuF4scod25af | CTRL_SrB6 |
| CTRL_Ar | CTRL_K2CuF4scod30 | CTRL_SrCd2Sb2 |
| CTRL_Au | CTRL_K2CuF4scod30af | CTRL_SrCu3Ru4O12 |
| CTRL_AuZn3 | CTRL_K2NiF4bct | CTRL_SrFeO3 |
| CTRL_B2O3 | CTRL_K2NiF4sco | CTRL_SrFeO3afd |
| CTRL_BN | CTRL_K2NiF4scoaf | CTRL_SrFeO3afz |
| CTRL_Ba | CTRL_K2NiF4scod05 | CTRL_SrFeO3fe |
| CTRL_BaAl10MgO17 | CTRL_K2NiF4scod05af | CTRL_SrTiO3 |
| CTRL_BaB2O4 | CTRL_K2NiF4scod10 | CTRL_TTF-TCNQ |
| CTRL_BaGa2O4 | CTRL_K2NiF4scod10af | CTRL_Ta |
| CTRL_BaNiO3 | CTRL_K2NiF4scod15 | CTRL_Ta2ISe8 |
| CTRL_BaO | CTRL_K2NiF4scod15af | CTRL_Ta2Se8 |
| CTRL_BaOn | CTRL_K2NiF4scod20 | CTRL_TaCrAl |
| CTRL_BaTiO3 | CTRL_K2NiF4scod20af | CTRL_TaFe2 |
| CTRL_BaVS3hex | CTRL_KAl3 | CTRL_TaFeAl |
| CTRL_Be | CTRL_KB6 | CTRL_TaNiAl |
| CTRL_BeO | CTRL_Kr | CTRL_TaS2-1T |

CTRL_Bi2CuO4
CTRL_Bi2Sr2CaCuO8
CTRL_C
CTRL_C3N4beta
CTRL_C60
CTRL_Ca
CTRL_Ca2NiSn
CTRL_Ca2RuO4lt
CTRL_Ca2RuO4mt
CTRL_Ca2RuO4so
CTRL_Ca4Mn4O10
CTRL_Ca7LaB48
CTRL_CaAl2
CTRL_CaB6
CTRL_CaCu3Mn4O12
CTRL_CaCu3Ru4O12
CTRL_CaGa2O4sm
CTRL_CaGa2O4smo
CTRL_CaGa2O4so
CTRL_CaGa2O4soo
CTRL_CaO
CTRL_CaOn
CTRL_CaTiO3
CTRL_Cd
CTRL_CdS
CTRL_CdTe
CTRL_Cdia
CTRL_Ce
CTRL_Ce2NiSn
CTRL_CeAg
CTRL_CeAl2
CTRL_CeAl3
CTRL_CeCu2Si2
CTRL_CeI2
CTRL_CeIrIn5
CTRL_CeNi2Ge2
CTRL_CePd3
CTRL_CeRu2Si2
CTRL_Co4N
CTRL_CoO
CTRL_CoOfcc
CTRL_CoOtrg
CTRL_CoOtrgaf
CTRL_CoS2
CTRL_CoSb3
CTRL_Cofcc
CTRL_Cohcp
CTRL_Cr
CTRL_CrO2
CTRL_La
CTRL_La2BaCuO5
CTRL_La2BaCuO5afc
CTRL_La2BaCuO5afn
CTRL_La2BaCuO5afp
CTRL_La2BaCuO5fe
CTRL_La2BaCuO5fenew
CTRL_La2BaCuO5new
CTRL_La2O3
CTRL_La5Ti5O17
CTRL_LaAl2
CTRL_LaB6
CTRL_LaCu3Ru4O12
CTRL_LaF3
CTRL_LaI2
CTRL_LaMgBO
CTRL_LaMnO3sc
CTRL_LaMnO3so
CTRL_LaMnO3soaf
CTRL_LaNi5
CTRL_LaNi5H7
CTRL_LaTiO3
CTRL_LaTiO3sc
CTRL_Li
CTRL_Li12Si7
CTRL_Li2FeS2
CTRL_Li2RuS2
CTRL_Li2WO4
CTRL_Li3FeN2
CTRL_LiF
CTRL_LiH
CTRL_LiMn2O4
CTRL_LiTi2O4
CTRL_LiTi2O4id
CTRL_LiV2O4fcc
CTRL_LiV2O4fccid
CTRL_LiV2O4tri
CTRL_LiV2O4triid
CTRL_LiYF4
CTRL_Lu
CTRL_Lu2O3
CTRL_LuAl3
CTRL_LuCu4Au
CTRL_LuCu4Pd
CTRL_Mg
CTRL_Mg2Ge
CTRL_Mg2Si
CTRL_MgB2
CTRL_MgGa2O4
CTRL_TaS2-2H
CTRL_TaS2E-1T
CTRL_TaS2E-2H
CTRL_TaS2Na-1T
CTRL_TaS2Na-2H
CTRL_TaSe2-1T
CTRL_TaSe2-2H
CTRL_TaTe2
CTRL_Tb
CTRL_Tc
CTRL_Th
CTRL_Ti
CTRL_Ti2O3dM
CTRL_Ti2O3dY
CTRL_Ti2O3h108
CTRL_Ti2O3h14
CTRL_Ti2O3h202
CTRL_Ti2O3h296
CTRL_Ti2O3ht
CTRL_Ti2O3rt
CTRL_Ti5O9LT
CTRL_Ti5O9RT
CTRL_TiB
CTRL_TiCfcc
CTRL_TiCr2
CTRL_TiFeAl
CTRL_TiO2
CTRL_TiS2
CTRL_TiTe2
CTRL_Tl
CTRL_TlBr
CTRL_TlCl
CTRL_TlI
CTRL_Tm
CTRL_TmAl3
CTRL_U
CTRL_U3Co4Ge7
CTRL_UAl3
CTRL_UB2
CTRL_UB4
CTRL_UB6
CTRL_UCo2Ge2
CTRL_UCo2Ge2feh
CTRL_UCo2Ge2fel
CTRL_UCoSn
CTRL_UCr2Ge2
CTRL_UCu2Ge2
CTRL_UFe2Ge2
CTRL_UGe3

CTRL_CrO2af
CTRL_CrO2fe
CTRL_CrOCl
CTRL_Crfcc
CTRL_Crhcp
CTRL_Cs
CTRL_CsBr
CTRL_CsCl
CTRL_CsI
CTRL_CsMn[Cr(CN)6]
CTRL_CsNi[Cr(CN)6]
CTRL_Cu
CTRL_Cu12
CTRL_Cu16
CTRL_Cu2O
CTRL_Cu32
CTRL_Cu4
CTRL_Cu48
CTRL_Cu8
CTRL_CuAlS2
CTRL_CuGaS2
CTRL_CuGaSe2
CTRL_CuGaSe2hetZnSe
CTRL_CuInS2
CTRL_CuInS2id
CTRL_CuInSe2
CTRL_CuS2
CTRL_Cu[NH3]4SO4H2O
CTRL_Dy
CTRL_DyNi5
CTRL_Er
CTRL_Es
CTRL_EuB6
CTRL_EuO
CTRL_EuS
CTRL_Fe
CTRL_Fe16
CTRL_Fe2
CTRL_Fe32
CTRL_Fe3O4fcc
CTRL_Fe3O4so
CTRL_Fe3O4soid
CTRL_Fe4
CTRL_Fe48
CTRL_Fe4N
CTRL_Fe4Nfe
CTRL_Fe8
CTRL_FeB2
CTRL_FeNCrN
CTRL_MgO
CTRL_MgOn
CTRL_MgV2O4
CTRL_MgV2O5
CTRL_Mn
CTRL_Mn2[H2O]5Mo[CN]7
CTRL_Mn3Sn
CTRL_Mn3Snaf
CTRL_Mn3Snfe
CTRL_Mn4N
CTRL_MnAs
CTRL_MnBi
CTRL_MnO
CTRL_MnO2
CTRL_MnO2af
CTRL_MnOfcc
CTRL_MnOtrg
CTRL_MnOtrgaf
CTRL_MnWO4
CTRL_MnWO4af1
CTRL_MnWO4fe
CTRL_Mo
CTRL_Mo3Si
CTRL_MoNfcc
CTRL_MoNhex
CTRL_MoO2mon
CTRL_MoO2mondo
CTRL_MoO2moni
CTRL_MoO2monns
CTRL_MoO2monod
CTRL_MoO2monoz
CTRL_MoO2rut
CTRL_MoO3
CTRL_MoSi2
CTRL_NOC6
CTRL_Na
CTRL_Na8Si46
CTRL_NaCl
CTRL_NaCu3Ru4O12
CTRL_NaV2O5
CTRL_NaV2O5new
CTRL_NaV6O15
CTRL_NaW2
CTRL_Nb
CTRL_NbCfcc
CTRL_NbCr2
CTRL_NbCrAl
CTRL_NbFeAl
CTRL_NbNiAl
CTRL_UMn2Ge2
CTRL_UNi2Ge2
CTRL_UO2
CTRL_UPt3
CTRL_URhAl
CTRL_V
CTRL_V2O3LTcm
CTRL_V2O3LTmon
CTRL_V2O3LTmonaf
CTRL_V2O3RTcm
CTRL_V2O3RThex
CTRL_V2O3RTmon
CTRL_V2O3RTmonaf
CTRL_V2O3RTtri
CTRL_V2O5
CTRL_V2O5id
CTRL_V2O5idd
CTRL_V2O5idz
CTRL_V2O5sco
CTRL_V3O5HT
CTRL_V3O5LT
CTRL_V4O7LT
CTRL_V4O7RT
CTRL_V4O7RTjp
CTRL_V4O7id
CTRL_V5O9LT
CTRL_V5O9RT
CTRL_V6O11LT
CTRL_V6O11RT
CTRL_VH
CTRL_VO
CTRL_VO2mon1
CTRL_VO2mon1i
CTRL_VO2mon1w
CTRL_VO2mon1x10
CTRL_VO2mon1x20
CTRL_VO2mon1x60
CTRL_VO2mon2
CTRL_VO2mon2af
CTRL_VO2mon2afx20
CTRL_VO2mon2afx40
CTRL_VO2mon2afx60
CTRL_VO2mon2x20
CTRL_VO2mon2x40
CTRL_VO2mon2x60
CTRL_VO2rut
CTRL_VO2rutw
CTRL_V[Cr(CN)6]
CTRL_W

CTRL_FeOCl
CTRL_FeOfcc
CTRL_FeOtrg
CTRL_FeOtrgaf
CTRL_FeS2
CTRL_FeSi
CTRL_FeSi2
CTRL_FeTi
CTRL_FeWO4
CTRL_FeWO4af1
CTRL_FeWO4fe
CTRL_Ga
CTRL_Ga2O3
CTRL_Ga2O3tri
CTRL_GaAs
CTRL_GaP
CTRL_GaSb
CTRL_Gafcc
CTRL_Gd
CTRL_Gd2O2S
CTRL_GdAg
CTRL_GdAgaf001
CTRL_GdAgaf110
CTRL_GdAgaf111
CTRL_GdAl2
CTRL_GdS
CTRL_GdSscaf
CTRL_GdStrg
CTRL_GdStrgaf
CTRL_GdTiO3
CTRL_GdZn
CTRL_GdZnaf001
CTRL_GdZnaf110
CTRL_GdZnaf111
CTRL_Ge
CTRL_GeO2
CTRL_H
CTRL_He
CTRL_Hf
CTRL_HfAl3
CTRL_HfSe2
CTRL_Hg
CTRL_HgMg3
CTRL_Ho
CTRL_In
CTRL_In2O3
CTRL_In2O3b1
CTRL_In2O3id
CTRL_In2O3s1
CTRL_NbO2bct
CTRL_NbO2bctb
CTRL_NbO2bctbi
CTRL_NbO2rut
CTRL_NbSe2-2H
CTRL_Nd
CTRL_Nd2BaCuO5
CTRL_NdCu3Ru4O12
CTRL_Ne
CTRL_Ni
CTRL_Ni3Al
CTRL_Ni3C
CTRL_NiF2
CTRL_NiO
CTRL_NiOfcc
CTRL_NiOtrg
CTRL_NiOtrgaf
CTRL_NiS
CTRL_NiS2
CTRL_NiS2af
CTRL_NiSe2
CTRL_NiTi
CTRL_Nibcc
CTRL_NpB2
CTRL_Os
CTRL_P2O5
CTRL_Pb
CTRL_PbBiCuS3
CTRL_PbO2
CTRL_Pd
CTRL_Pr
CTRL_Pt
CTRL_Pu
CTRL_Rb
CTRL_Rb7Na16Sb7
CTRL_Re
CTRL_Rh
CTRL_Ru
CTRL_Ru2Si3
CTRL_Ru4Se4[CO]12
CTRL_Ru4Se4[CO]12dis
CTRL_Ru4Se4[CO]12id
CTRL_Ru4Se4[CO]12idtri
CTRL_Ru4Se4[CO]12tri
CTRL_Ru6C[CO]17
CTRL_RuC
CTRL_RuO2
CTRL_RuOC10H16
CTRL_RuS2
CTRL_WAl3
CTRL_WS2-2H
CTRL_WS2-2Hdl
CTRL_WS2-2Hsl
CTRL_WSe2-2H
CTRL_WTe2-1T
CTRL_WTe2-2H
CTRL_WTe2-Td
CTRL_Xe
CTRL_Y
CTRL_Y2O2S
CTRL_Y2O3
CTRL_Y3Co4Ge7
CTRL_YAl2
CTRL_YBO3
CTRL_YBO3hex
CTRL_YBa2Cu3O7
CTRL_YCuO2
CTRL_YIr2Si2bct
CTRL_YIr2Si2st
CTRL_Yb
CTRL_YbAl3
CTRL_YbCo2Ge2
CTRL_YbCu4Au
CTRL_YbCu4Pd
CTRL_Zn
CTRL_ZnGa2O4
CTRL_ZnO
CTRL_ZnS
CTRL_ZnS2
CTRL_ZnSbct
CTRL_ZnSdfcc
CTRL_ZnSdfccs1
CTRL_ZnSdfccs2
CTRL_ZnSdsc
CTRL_ZnSdscs1
CTRL_ZnSe
CTRL_ZnSsc
CTRL_ZnSscs1
CTRL_Zr
CTRL_ZrAl2
CTRL_ZrCo2
CTRL_ZrCr2cub
CTRL_ZrCr2hex
CTRL_ZrFe2
CTRL_ZrMn2
CTRL_ZrMo2
CTRL_ZrV2
CTRL_ZrW2

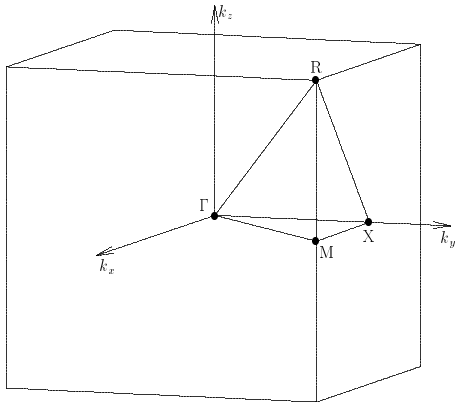
```
CTRL_In2O3s2        CTRL_RuSe2          CTRL_ZrZn2
CTRL_InAs           CTRL_Sc             CTRL_[BEDT-TTF]2I3
CTRL_InP            CTRL_ScAl2          CTRL_[V0.962Cr0.038]2O3
CTRL_InSb           CTRL_Si             CTRL_[V0.990Cr0.010]2O3a
CTRL_Infcc          CTRL_Si46           CTRL_[V0.990Cr0.010]2O3b
CTRL_Ir             CTRL_SiAl3          CTRL_[VO]2P2O7
CTRL_IrO2           CTRL_SiO2           CTRL_[VO]2P2O7hp
CTRL_K              CTRL_SmAl3          CTRL_[VO]2P2O7id
CTRL_K2CuF4bct      CTRL_SmB6
CTRL_K2CuF4sco      CTRL_SmS
```

Whenever you want to perform a calculation for one of the systems already in the database just copy the CTRL file. However, please read the README file contained in the database before. In turn, I do welcome any new CTRL file.

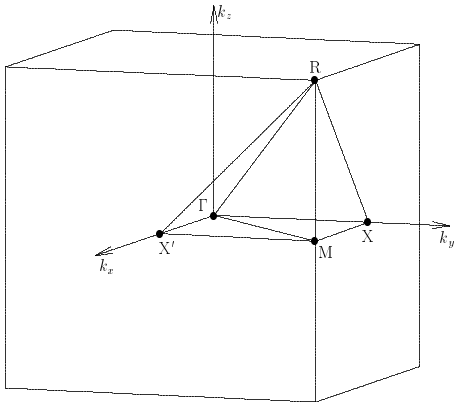
# Appendix A

# Brillouin zones

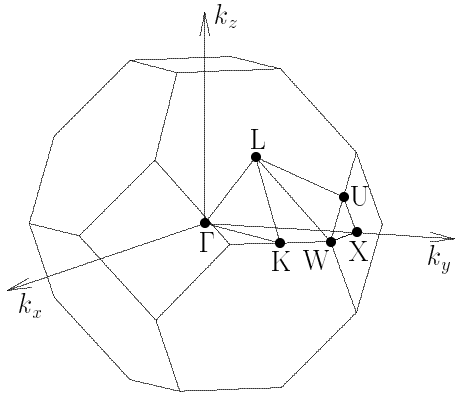
This Appendix displays the Brillouin zones of some lattices.



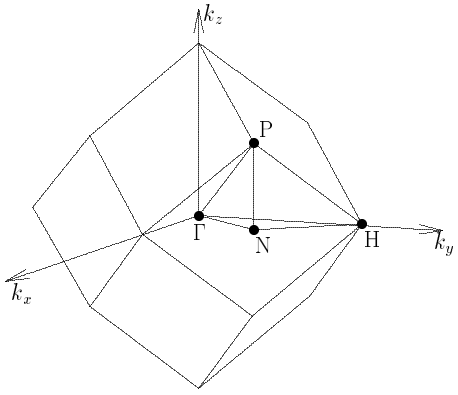

(a) simple cubic




(b) $T_h^6$ structure

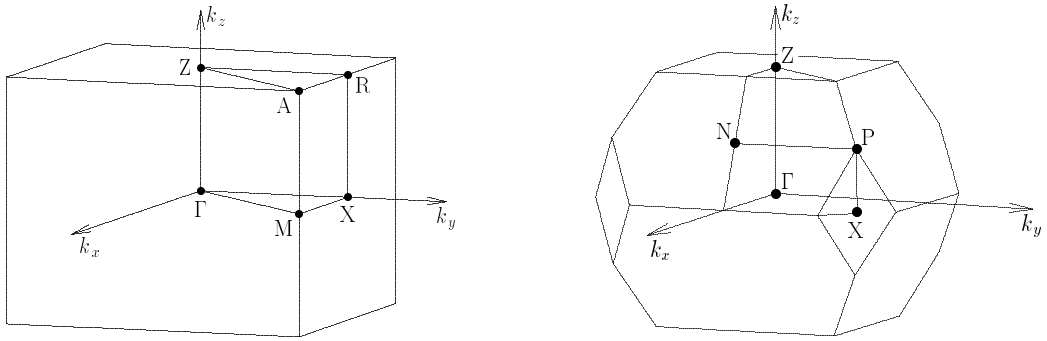



(c) face-centered cubic



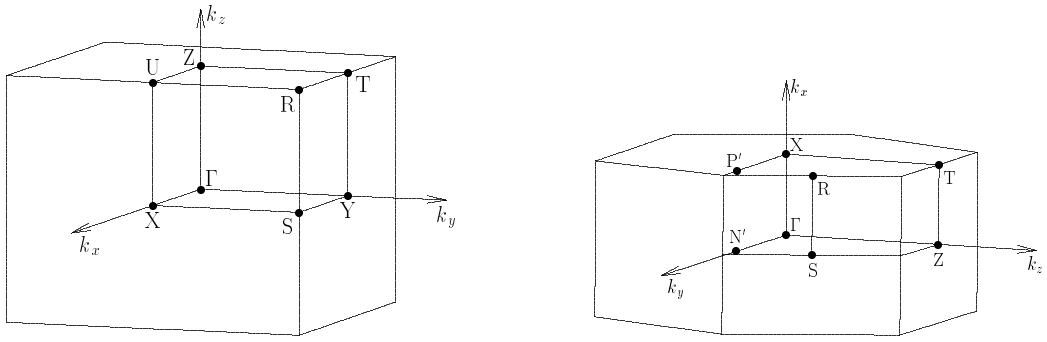

(d) body-centered cubic

**Figure A.1:**Brillouin zones of cubic lattices.




(a) simple tetragonal (b) centered tetragonal

**Figure A.2:**Brillouin zones of tetragonal lattices.




(a) simple orthorhombic (b) base-centered orthorhombic

**Figure A.3:**Brillouin zones of orthorhombic lattices.

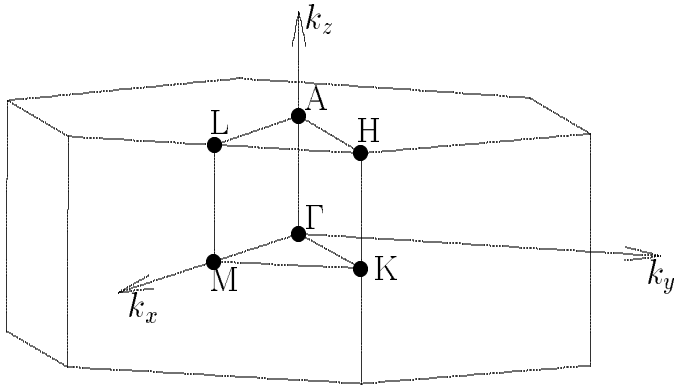
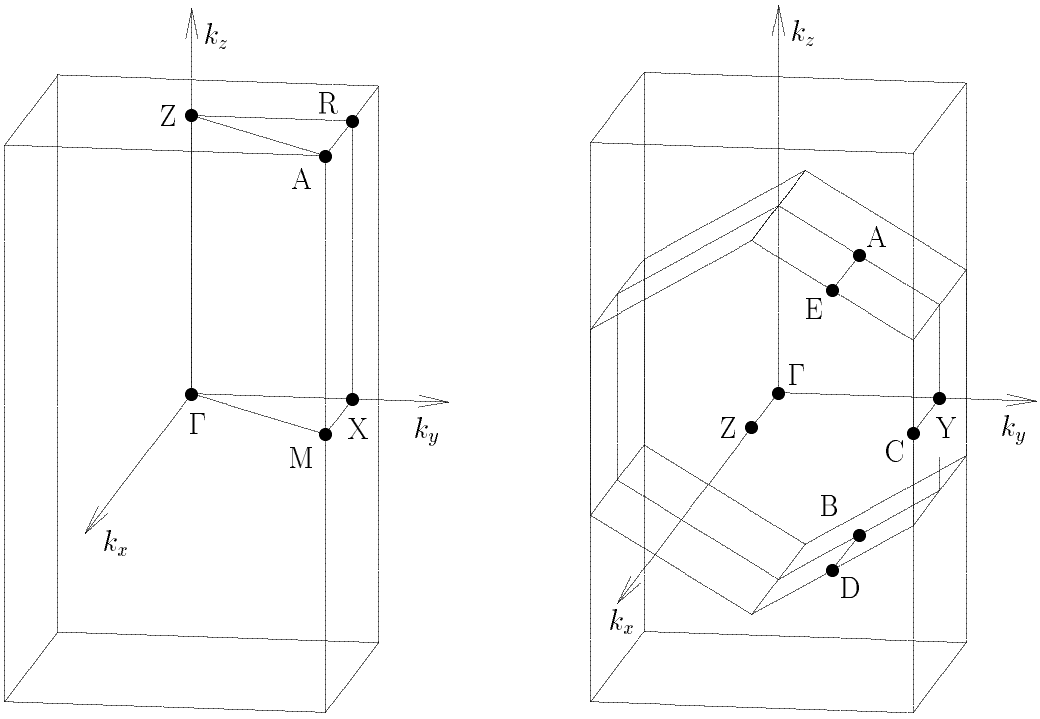



**Figure A.4:**Brillouin zones of hexagonal lattice.




(a) simple tetragonal

(b) simple monoclinic

**Figure A.5:**Brillouin zones of rutile and related structures.

# Bibliography


[1] V. Eyert, *Entwicklung und Implementation eines Full-Potential-ASW-Verfahrens*, Ph.D. Thesis, Technische Hochschule Darmstadt, 1991.

[2] V. Eyert, J. Comput. Phys. **124**, 271 (1996).

[3] V. Eyert, *Electronic structure calculations for crystalline materials*, in: *Density Functional Methods: Applications in Chemistry and Materials Science*, edited by M. Springborg (Wiley, Chichester, 1997), pp. 233-304.

[4] V. Eyert, *Octahedral Deformations and Metal-Insulator Transition in Transition Metal Chalcogenides*, Habilitation Thesis, Universität Augsburg, 1998.

[5] V. Eyert, *Basic notions and applications of the augmented spherical wave method*, Int. J. Quantum Chem. **77**, 1007-1031 (2000).

[6] V. Eyert and K.-H. Höck, Phys. Rev. B **57**, 12727 (1998).

[7] J. Kübler and V. Eyert, *Electronic structure calculations*, in: *Electronic and Magnetic Properties of Metals and Ceramics*, edited by K. H. J. Buschow (VCH Verlagsgesellschaft, Weinheim, 1992), pp. 1-145;
Volume 3A of *Materials Science and Technology*, edited by R. W. Cahn, P. Haasen, and E. J. Kramer (VCH Verlagsgesellschaft, Weinheim, 1991-1996).

[8] J. Kübler, K.-H. Höck, J. Sticht, and A. R. Williams, J. Phys. F **18**, 469 (1988).

[9] J. Kübler, K.-H. Höck, J. Sticht, and A. R. Williams, J. Appl. Phys. **63**, 3482 (1988).

[10] A. R. Williams, J. Kübler, and C. D. Gelatt, Jr., Phys. Rev. B **19**, 6094 (1979).

[11] C. D. Gelatt, Jr., H. Ehrenreich, and R. E. Watson, Phys. Rev. B **15**, 1613 (1977).

[12] L. Hodges, R. E. Watson, H. Ehrenreich, Phys. Rev. B **5**, 3953 (1972).

[13] R. E. Watson, H. Ehrenreich, and L. Hodges, Phys. Rev. Lett. **15**, 829 (1970).

[14] O. K. Andersen, Phys. Rev. B **12**, 3060 (1975).

[15] M. S. Methfessel and A. T. Paxton, Phys. Rev. B **40**, 3616 (1989).

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

[17] G. A. Burdick, Phys. Rev. **129**, 138 (1963)

[18] R. Hoffmann, *Solids and Surfaces: A Chemist's View of Bonding in Extended Structures*, VCH, New York, 1988.

[19] V. Eyert and S. F. Matar, (unpublished results, 1995).

[20] V. Eyert, B. Siberchicot, and M. Verdaguer, Phys. Rev. B **56**, 8959 (1997).

[21] N. Börnsen, B. Meyer, O. Grotheer, and M. Fähnle, J. Phys.: Cond. Matt. **11**, L287 (1999).

[22] Th. Straub, R. Claessen, P. Steiner, S. Hüfner, V. Eyert, K. Friemelt, and E. Bucher, Phys. Rev. B **55**, 13473-13478 (1997).

[23] V. Eyert, K.-H. Höck, S. Fiechter and H. Tributsch, Phys. Rev. B **57**, 6350 (1998).

[24] K. Schwarz, J. Phys. **F16**, L211 (1986).

[25] S. Matar, G. Demazeau, J. Sticht, V. Eyert, and J. Kübler, J. Phys. I France **2**, 315 (1992); J. Phys. I France **4**, 1259 (1994).

[26] S. Matar, V. Eyert, J. Sticht, J., Kübler, and G. Demazeau, J. Phys. I France **4**, 1199 (1994).

[27] A. A. Bolzan, C. Fong, B. J. Kennedy, and C. J. Howard, Acta Cryst. B **53**, 373 (1997).