BICON-CEDiT

Extended-Hückel Band Structure
and
Crystal Electronic Dipole induced Transitions Calculations

Manual

Martin Brändle, Ruedi Rytz and Gion Calzaferri
Department of Chemistry and Biochemistry
University of Bern
Freiestr. 3, CH-3012 Bern, Switzerland

Bern, August 1997

Foreword

BICON-CEDiT is a set of FORTRAN programs that allow extended-Hückel tight-binding calculations on 1D-3D solid-state compounds, visualisation of results like band structures, density of states (DOS) and crystal orbital overlap curves (COOP), and the calculation of oscillator strengths of crystal electronic dipole induced transitions (CEDiT).

The package consists of the extended-Hückel band structure and MO program **EHMACC**, the program **EHPC** for energy and density of states calculations, the programs **BANDPLOT** and **DOSPLOT** for graphical output, the program **CEDiT** for calculation of transition oscillator strengths in the solid, and some miscellaneous tools.

EHMACC was originally written by M.-H. Whangbo in the laboratory of R. Hoffmann, Dept. of Chemistry, Cornell University, Ithaca, N.Y., Dec. 1976. Later modifications and additions by T. Hughbanks, M. Kertesz, S. Wijeyesekera, C. Wilker and C. Zheng in the laboratory of R. Hoffmann, Dept. of Chemistry, Cornell University, Ithaca, N.Y. Modified by M. Evain in the laboratory of M.-H. Whangbo, Dept of Chemistry, North Carolina State University, Raleigh N.C., Dec. 1987.
New features include a two-body repulsive energy term, a distance-dependent weighted Wolfsberg-Helmholz formula, symmetry analysis of the crystal orbitals, use of fractional coordinates, and automated geometry variation, added by Martin Brändle in the laboratory of Gion Calzaferri, Department of Chemistry and Biochemistry, University of Berne, Freiestr. 3, CH-3012 Berne, Switzerland.

EHPC was originally written by T. Hughbanks, M. Kertesz, S.Wijeyesekera, C. Wilker, and C. Zheng in the laboratory of R. Hoffmann, Dept. of Chemistry, Cornell University, Ithaca, N.Y. Written as a separate package by M.-H. Whangbo, Dept. of Chemistry, North Carolina State University, Raleigh N.C., 1981. Modified by M. Evain in the laboratory of M.-H. Whangbo, Dept. of Chemistry, North Carolina State University, Raleigh N.C., 1987.
New features include Gaussian smoothed integrals of AO and FMO contributions to the DOS and of COOP curves, and selection of geometry variations, added by Martin Brändle in the laboratory of Gion Calzaferri, Department of Chemistry and Biochemistry, University of Berne, Freiestr. 3, CH-3012 Berne, Switzerland.

BANDPLOT and **DOSPLOT** were written by Martin Brändle in the laboratory of G. Calza-ferri, Department of Chemistry and Biochemistry, University of Berne, Freiestr. 3, CH-3012 Berne, Switzerland. The band interpolation subroutines for BANDPLOT were adapted from the BFIT program written by S. Wijeysekera in the laboratory of R. Hoffmann, Dept. of Chemistry, Cornell University, Ithaca, N.Y.

CEDiT was written by Ruedi Rytz in the laboratory of Gion Calzaferri, Department of Chemistry and Biochemistry, University of Berne, Freiestr. 3, CH-3012 Berne, Switzerland.

This manual is based partially on the manuals for the QCPE #571[38] package and for the programs[37] used in the group of Roald Hoffmann.

Table of Contents

1 INTRODUCTION

BICON-CEDiT is a set of FORTRAN programs that allow extended Hückel tight-binding calculations on 1D-3D solid-state compounds, visualisation of results like band structures, density of states (DOS) and crystal orbital overlap curves (COOP), and the calculation of oscillator strengths of crystal electronic dipole induced transitions (CEDiT).

The **EHMACC** part of the programs solves the extended-Hückel secular equation for solid state compounds and molecules. *Slater*-type s, p, d, and f orbitals, different Wolfsberg-Helmholz approximations, a two-body repulsive energy term, an automated geometry variation, a fragment molecular orbital FMO option, and a symmetry analysis of the crystal orbitals are included. Input is easily done by creating an input file containing legible commands, keywords and parameters with the users favourite editor.

The **EHPC** program calculates density of states DOS and DOS-weighted properties like the total energy, the *Fermi* level, and the COOP of an infinite solid. The input concept is the same as with EHMACC.

BANDPLOT allows the band interpolation and the graphic representation of bands. The user is guided through a simple text menu system which enables very fast input and control of the various graphic parameters.

DOSPLOT is used for the graphic representation of density of states weighted properties. The user is guided through a simple text menu system.

CEDiT calculates oscillator strengths of dipole induced transitions for crystals.

Some miscellaneous tools have been added, like **k**-point sets**,** **k**-point set programs and a coordinate transformation procedure **TRANSC**.

All programs are written in FORTRAN and run on Windows NT i386 PCs. No machine specific specialities have been included. The programs should be easily portable to other operating systems. Versions have been tested on AIX, Linux, VMS, MS-DOS, and MS-Windows. We provide, however, no support for porting the programs.

1.1 Conventions

Tables are written in HELVETICA typeface.

Names are written *italic*.

Filenames, computer input and output are written in `Courier` typeface.

Bold typeface in formulas is used for matrices and vectors.

2 QUICK START

2.1 Installation and requirements

The whole package needs approximately 10 MBytes of disk space.

Maximum memory requirement with the present program dimensions (see Chapter 2.4) is 29 MByte.

The package can be obtained as zipped files (`examples`, `exe`, `k_sets`, `sources`) from `http://iacrs1.unibe.ch/`.

Copy the files to the drive and directory where you want to install the package. Type

```
pkunzip -d examples.zip
pkunzip -d exe.zip
pkunzip -d k_sets.zip
pkunzip -d sources.zip
```

You obtain the directories `examples`, `exe`, `k_sets`, `sources`. For the contents see Chapter 2.6. The `exe`-directory is the working directory with the executables.

If you need to have pkunzip, see

`http://iacrs1.unibe.ch/members/ie_manual.html#package`

2.2 General usage

The order of use of the programs depends on the job one wants to perform, see Figure 1.

The calculation of band structure diagrams requires the knowledge of the symmetry properties of the structure to be investigated. First EHMACC has to be called to solve the Schrödinger equation along several **k**-points along the high symmetry lines of the Brillouin zone, then BANDPLOT has to be used to interpolate and plot the bands between the calculated **k**-points.

Properties like total energy, DOS, or COOP require their calculation at many **k**-points distributed over the irreducible Brillouin zone. EHPC selects and averages over them, while DOSPLOT is used for displaying the calculated properties.

Oscillator strengths also require calculations at many **k**-points distributed over the irreducible Brillouin zone. EHMACC is used to generate the necessary crystal orbital information, then CEDiT is called to calculate the oscillator strengths.

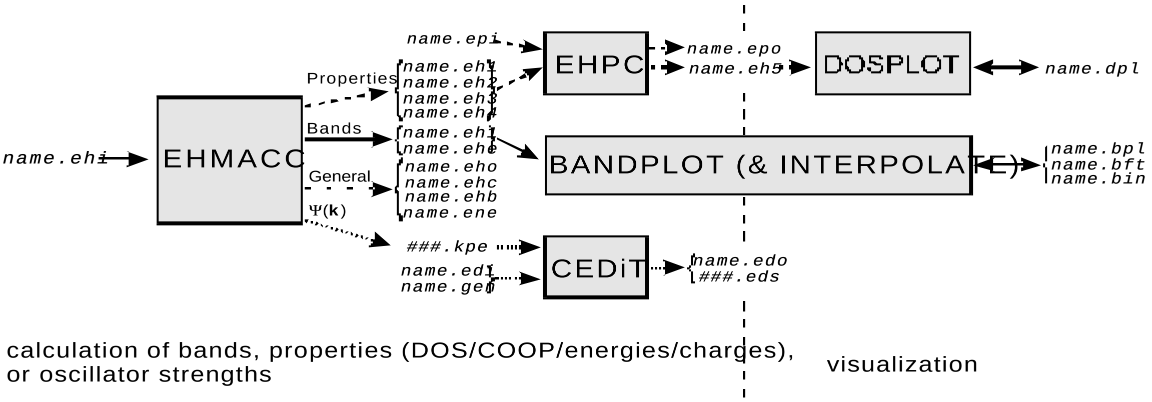


Figure 1: Flowchart of program usage. The italic names denote the files being read and written.

2.3 Example band structure calculation - Alternating all-trans-polyacetylene

Band structure of all-trans-polyacetylene in its alternating geometry (r(C_1-C_2)=1.36Å, r(C_2-C_a)=1.50Å). See also [46],[16].

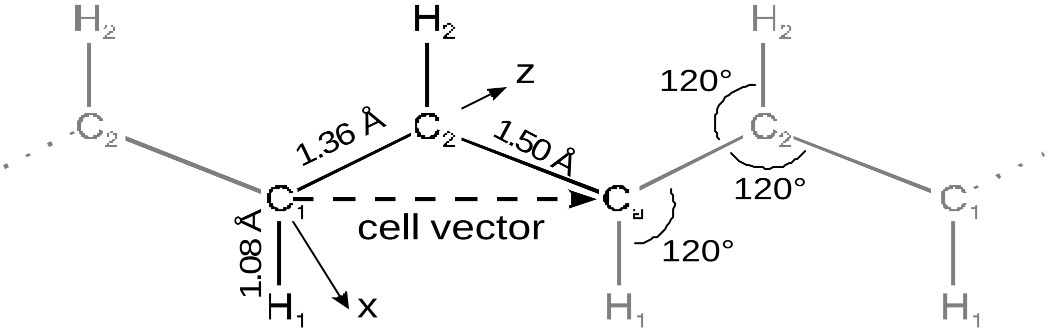


Figure 2: Alternating geometry of all-*trans*-polyacetylene, definition by internal coordinates.

1. Open a DOS command prompt window.
2. Copy the files `polyb136.*` in the `examples` directory to the working directory `exe`. Change to the `exe` directory.
3. You may examine the input file `polyb136.ehi` for the band structure calculation. For the meanings of the keywords refer to Chapter 4.1.
4. Type `ehmacc.bat polyb136`
5. After the calculation has finished, you obtain `polyb136.eh1`, `polyb136.ehc`, `polyb136.ehe`, `polyb136.eho`. Examine the output file `polyb136.eho`. Compare it with the file `polyb136.out` to check the correctness of the results.

An interpolated band structure is provided by the files `polyb136.bft` and `polyb136.bin`.

1. In the DOS command prompt window, type `bandplot.bat polyb136`

2. A new window `bandplot` is created. Maximize the window. Choose menu `Window`, entry `Tile`. The screen should look like this:

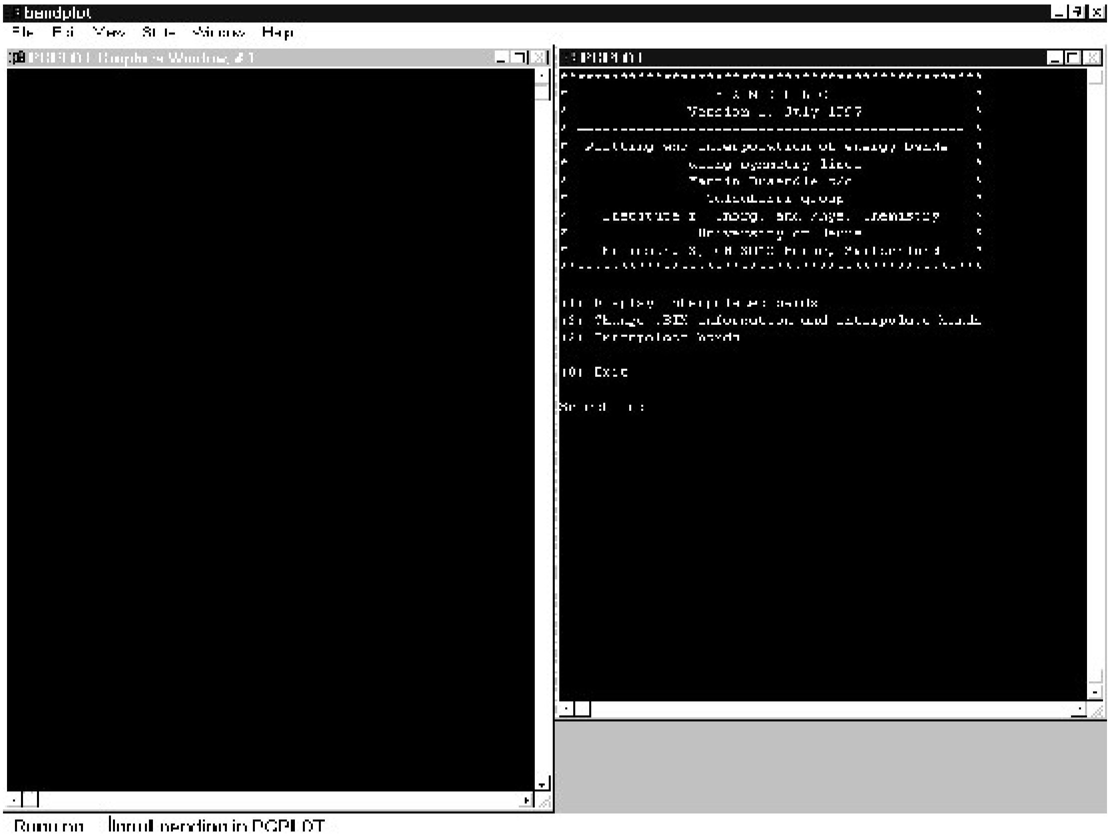


Figure 3: Startup screen of BANDPLOT (Windows NT version).

3. At left you see a graphics window, at right a terminal window. Select option (1) in the terminal window by entering a 1.
4. Use the default for the number of symmetry lines.
5. Numbers for energies and symmetry line widths require a decimal point. Provide an energy range of `-35.` to `65.` eV. The band structure will be drawn, and you enter a graph modification menu.

6. Select option (12). You should obtain a plot like this:

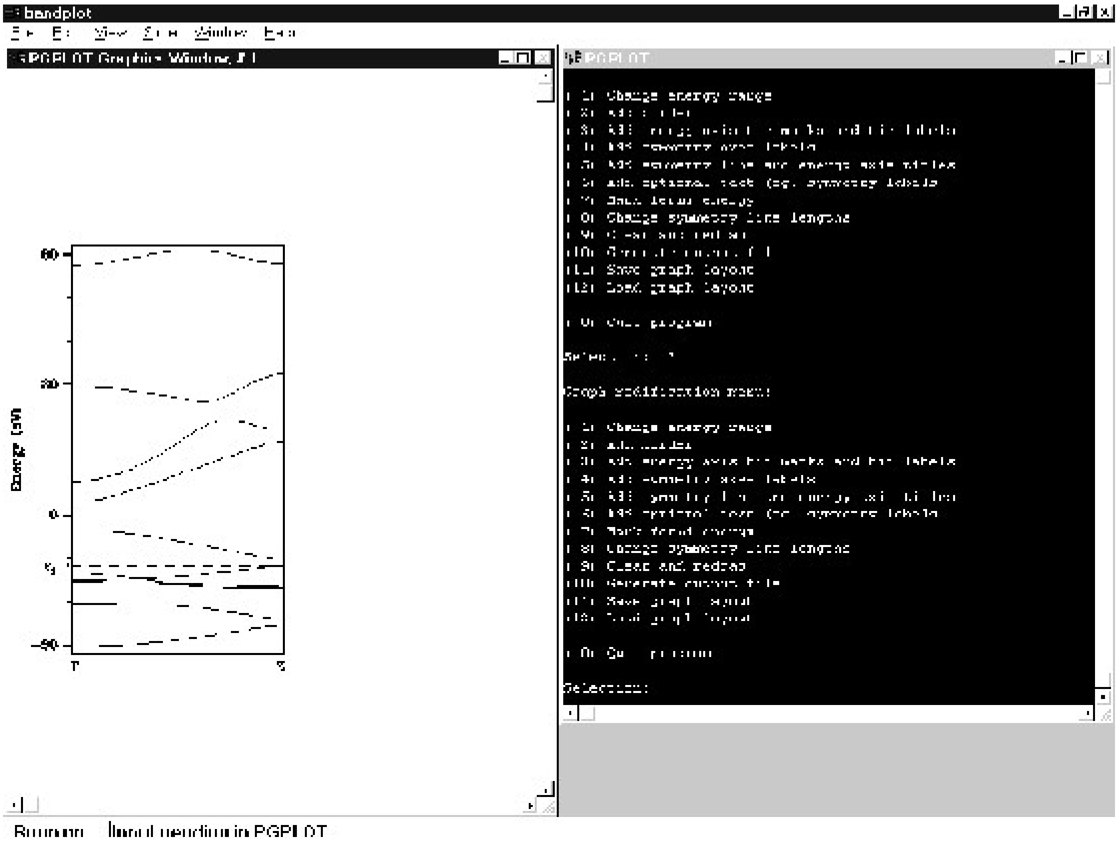


Figure 4: Band structure of alternating all-trans-polyacetylene generated by BANDPLOT.

For further explanation of the program see Chapter 4.4.

2.4 Example DOS calculation - Alternating all-trans-polyacetylene

The DOS of all-trans-polyacetylene, its AO projections and the COOP for the C-C and the C-H bonds are calculated over a mesh of 101 equally dispersed **k**-points of the linear *Brillouin* zone.

I. Open a DOS command prompt window.
1. Copy the files `polyd136.*` in the `examples` directory to the working directory exe. Change to the `exe` directory.
2. You may examine the input file `polyd136.ehi` for the band structure calculation. For the meanings of the keywords refer to Chapter 4.1.
3. Type `ehmacc.bat polyd136`
4. After the calculation has finished, you obtain `polyd136.eh1`, `polyd136.eh2`, `polyd136.eh3`, `polyd136.eho`. Examine the output file `polyd136.eho`.
5. For the integration and smoothing of the DOS and the COOP, EHPC has to be called. First inspect the input file `polyd136.epi`. For the meaning of keywords see Chapter 4.2.
6. Type `ehpc.bat polyd136`

7. After the calculation has finished, you obtain the new files `polyd136.eh5` and `polyd136.epo`. Examine the output file `polyd136.epo`. Compare it with the file `polyd136.out` to check the correctness of the results.
8. To visualize the DOS, the DOS projections and the COOP, type `dosplot.bat polyd136`
9. A new window `dosplot` is created, similar to BANDPLOT. Maximize the window. Choose menu `Window`, entry `Tile`. Your startup screen should look similar to Figure 2.
10. Select option (11). You should obtain a plot like this:

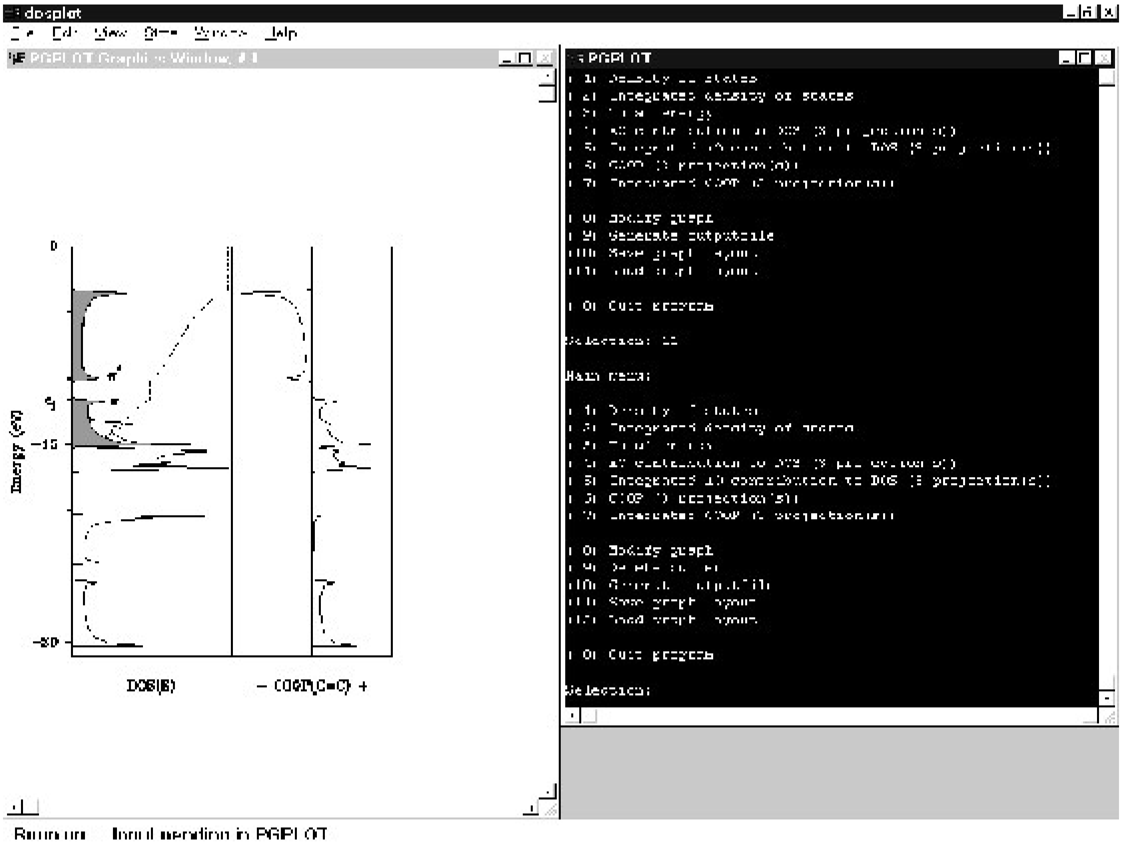


Figure 5: DOS and COOP of all-trans-polyacetylene generated by DOSPLOT.

For further explanation of the program see Chapter 4.3.

2.5 Example CEDiT Calculation - Alternating all-trans-polyacetylene

The $\pi^* \leftarrow \pi$ transitions in the band-gap region of a Peierls distorted all-trans-polyacetylene chain are calculated at 31 equally spaced **k**-points along the linear *Brillouin zone*.

1. Open a DOS command prompt window.
2. Copy the files `polye136.*` in the `examples` directory to the working directory `exe`.
3. The file `polye136.ehi` is adapted from `polyd136.ehi`. For the meanings of the keywords we refer to Chapter 4.1.

4. Type `ehmacc.bat polye136`
5. After successful termination of the calculation you will find the following new files in your working directory: polye136`.eh1`, `polye136.ehc`, `polye136.kp`, and 31 files that are continuousely numbered `001.kpe - 031.kpe`. They contain the relevant information for the CEDiT calculation.
6. `polye136.edi` contains the controlling information for the `CEDiT` program (cf. Chapter 4.5). Type `CEDiT <polye136.edi >polye136.edo`
7. A successful calculation will result in the output of the files `polye136.edo` and `001.eds - 031.eds`. You may want to have a look at `polye136.edo`, where the calculated oscillator strengths per **k**-point are given. Compare it with the file `polye136.out` to check the correctness of the results.

2.6 Default dimensions of the executables on the distribution

Table 1: Default dimensions of the executables on the distribution.

Program	Quantity	Dimension
EHMACC	# of atoms	100
	# of AOs	400
	# of bonds	1000
	# of **k**-points	300
	# of integration steps	6000
EHPC	# of atoms	100
	# of AOs	400
	# of bonds	1000
	# of **k**-points	300
	# of energy mesh points	1000
BANDPLOT	# of bands	1000
DOSPLOT	# of curve buffers	2x8
	# of energy mesh points	1000
CEDIT	# of atoms	100
	# of AOs	400
	# of transitions	5000

2.7 Contents of distribution

`exe:`

```
CEDiT.exe    bandplot.bat bandplot.exe dosplot.bat  dosplot.exe
ehmacc.bat   ehmacc.exe   ehpc.bat     ehpc.exe     grfont.dat
transc.exe
```

`examples:`

```
c_gdos.ehi   c_gdos.epi   c_gitter.bft c_gitter.bin c_gitter.bpl
c_gitter.ehi co_100ba.ehi co_100ds.ehi co_100ds.epi co_ni_18.ehi
co_ni_18.epi co_ni_30.ehi co_ni_30.epi co_ni_45.ehi co_ni_45.epi
diamond.ehi  mos2band.ehi mos2osc.edi  mos2osc.ehi  mos2osc.gen
mos2prop.ehi mos2prop.epi ni_100ba.ehi ni_100ds.ehi ni_100ds.epi
polyb136.bft polyb136.bin polyb136.bpl polyb136.cmp polyb136.ehi
polyb143.ehi polyd136.cmp polyd136.dpl polyd136.ehi polye136.edi
```

```
polye136.ehi polye136.gen polyd136.epi polyd143.ehi polyd143.epi
polyvary.ehi sodal_ba.ehi sodal_ds.ehi sodal_ds.epi
```

k_sets:

```
bcc_008.dat  crect2.128   hex2b066.dat prect1.009   sqr1_015.dat
bcc_010.dat  crect2.162   hex2b078.dat prect1.016   sqr1_021.dat
bcc_016.dat  crect2.200   hex2b091.dat prect1.025   sqr1_028.dat
bcc_020.dat  fcc_010.dat  hex2b105.dat prect1.036   sqr1_036.dat
bcc_030.dat  fcc_014.dat  hex2b120.dat prect1.049   sqr1_045.dat
bcc_035.dat  fcc_020.dat  hex2b136.dat prect1.064   sqr1_055.dat
bcc_048.dat  fcc_026.dat  hex2b153.dat prect1.081   sqr1_066.dat
bcc_056.dat  fcc_035.dat  hex2b171.dat prect1.100   sqr1_078.dat
bcc_074.dat  fcc_044.dat  hex2b190.dat prect1.121   sqr1_091.dat
bcc_084.dat  fcc_056.dat  lin_006.dat  prect1.144   sqr1_105.dat
bcc_106.dat  fcc_068.dat  lin_011.dat  prect1.169   sqr1_120.dat
bcc_120.dat  fcc_084.dat  lin_026.dat  prect1.196   sqr1_136.dat
bcc_149.dat  fcc_100.dat  lin_051.dat  prect2.016   sqr1_153.dat
bcc_165.dat  fcc_120.dat  lin_101.dat  prect2.025   sqr1_171.dat
bcc_199.dat  fcc_140.dat  obliq1.010   prect2.036   sqr1_190.dat
bcc_220.dat  fcc_165.dat  obliq1.020   prect2.049   sqr2_015.dat
bcc_262.dat  fcc_190.dat  obliq1.034   prect2.064   sqr2_021.dat
crect1.005   fcc_220.dat  obliq1.052   prect2.081   sqr2_028.dat
crect1.013   fcc_250.dat  obliq1.074   prect2.100   sqr2_036.dat
crect1.025   hex1a007.dat obliq1.100   prect2.121   sqr2_045.dat
crect1.041   hex1a019.dat obliq1.130   prect2.144   sqr2_055.dat
crect1.061   hex1a037.dat obliq1.164   prect2.169   sqr2_066.dat
crect1.085   hex1a062.dat obliq1.202   prect2.196   sqr2_078.dat
crect1.113   hex1a091.dat obliq2.018   sc_010.dat   sqr2_091.dat
crect1.145   hex1a127.dat obliq2.032   sc_020.dat   sqr2_105.dat
crect1.181   hex1a169.dat obliq2.050   sc_035.dat   sqr2_120.dat
crect2.008   hex2b015.dat obliq2.072   sc_056.dat   sqr2_136.dat
crect2.018   hex2b021.dat obliq2.098   sc_084.dat   sqr2_153.dat
crect2.032   hex2b028.dat obliq2.128   sc_120.dat   sqr2_171.dat
crect2.050   hex2b036.dat obliq2.162   sc_165.dat   sqr2_190.dat
crect2.072   hex2b045.dat obliq2.200   sc_220.dat
crect2.098   hex2b055.dat prect1.004   sc_286.dat
```

sources:

```
bandplot\    cedit\       dosapp\     dosplot\      ehmacc\
ehpc\        kmeshprg\    transc\
```

sources\bandplot:

```
READ.ME      a.f         bandplot.f    bfit.f       bpaxes.f
bpband.f     bpbrdr.f    bpbsrt.f      bpcopy.f     bpener.f
bpfrmi.f     bpgrph.f    bpopen.f      bpraw.f      bprdrw.f
bpselc.f     bpsrt2.f    bpsyml.f      bptica.f     bptics.f
bptrap.f     da.f        dinv.f        dos.f        eread.f
inv.f        sizes       tsort.f       bandplot.mak bandplot.mdp
```

sources\cedit:

```
abfns.f   addk.f     bdiag.f   cryst.f   ddovl.f     delta.f
dfovl.f   diag.f     dodele.f  dopele.f  dosele.f    edit.f
eulerd.f  eulerf.f   eulerp.f  fpovl.f   getsel.f    lcase.f
lovlap.f  matrix.f   mout.f    obdiag.f  out.f       pdoce.f
pdovl.f   podele.f   popele.f  posele.f  ppovl.f     reada.f
readl.f   readwa.f   sdovl.f   sfovl.f   sodele.f    sopele.f
sosele.f  specout.f  spoce.f   spovl.f   truncls.f   writel.f
wrtkey.f  dconst.inc dfns.inc  dnorm.inc edconst.inc pconst.inc
```

```
pfns.inc  pnorm.inc  snorm.inc sizes     cedit.mak   cedit.mdp
```

sources\dosapp:

```
dosapp.f
```

sources\dosplot:

```
READ.ME      dosplot.f   dpapnd.f      dpaxes.f     dpbrdr.f
dpfrmi.f     dpgrph.f    dplist.f      dpolay.f     dpplot.f
dprdrw.f     dpselc.f    dptext.f      dptica.f     dptics.f
dptitl.f     dptrap.f    sizes         dosplot.mak  dosplot.mdp
```

sources\ehmacc:

```
abfns.f      addk.f      aoout.f       arrayd.f     arrays.f
bfmo.f       bhmat.f     blkdat.f      bpop.f       bsmat.f
buildz.f     cboris.f    cchol.f       clexit.f     cryst.f
ctql2.f      ctred2.f    distc.f       ehfmo.f      ehmacc.f
etot.f       ftocar.f    gaoocc.f      getfmo.f     getkd.f
getkp.f      getlat.f    getnam.f      getnum.f     getorb.f
getper.f     getpos.f    getsel.f      getsym.f     getvar.f
grad.f       hmat.f      huckel.f      inmodi.f     lovlap.f
matmul.f     matout.f    matpac.f      mopop.f      mopop1.f
mov.f        movlap.f    mu.f          oedit.f      omatat.f
omatbo.f     omatbp.f    omatt.f       omatt2.f     omatt3.f
omf1.f       omfrag.f    orbrd.f       outmat.f     param.f
part1.f      part2.f     pobond.f      popula.f     prbond.f
rbond.f      reada.f     readl.f       repul.f      repul1.f
repul2.f     search.f    shout.f       sizes        smat.f
srest.f      suppr.f     symman.f      tempo.f      unique.f
varygm.f     vec.f       vprod.f       writel.f     wrtkey.f
ehmacc.mak   ehmacc.mdp
```

sources\ehpc:

```
READ.ME      clexit.f    ehpc.f        fdst.f       getnam.f
getorb.f     gpopt.f     gproj.f       gprojb.f     gprojf.f
integr.f     output.f    parray.f      ppart.f      projt.f
projtb.f     reada.f     readl.f       search.f     sizes
smootb.f     smooth.f    smooto.f      writel.f     ehpc.mak
ehpc.mdp
```

sources\kmeshprg:

```
bcc.f        crect1.f    crect2.f      fcc.f        hexag1.f
hexag2.f     linear.f    obliq1.f      obliq2.f     prect1.f
prect2.f     sc.f        squar1.f      squar2.f
```

sources\transc:

```
READ.ME      matmul.f    readl.f       reada.f      transc.f
transc.mak   transc.mdp
```

3 THEORY AND PRACTICAL ADVICES

3.1 Generalized tight binding method

In performing LCAO calculations for a crystalline solid (i.e. tight-binding calculations), one begins with a symmetry-adapted basis set of orbitals $\{\phi_\mu(\boldsymbol{k})\}$ that are Bloch sums of AOs χ_μ of the unit cells:

$$\phi_\mu(\boldsymbol{k}) = N^{-1/2} \sum_{\boldsymbol{R}} e^{i\boldsymbol{k}\bullet\boldsymbol{R}} \chi_\mu(\boldsymbol{r}-\boldsymbol{R}) \tag{1}$$

The sum over $\boldsymbol{R}$ is a summation over the N unit cells of the crystal; the function $\chi_\mu(\boldsymbol{r}-\boldsymbol{R})$ is the μ^{th} AO of a unit cell at the lattice site specified by $\boldsymbol{R}$. The factor $e^{i\boldsymbol{k}\bullet\boldsymbol{R}}$ specifies the phase change in the orbitals $\{\phi_\mu(\boldsymbol{k})\}$ upon moving from a given reference unit cell (at the origin) to a unit cell at the site specified by $\boldsymbol{R}$. The orbitals $\{\phi_\mu(\boldsymbol{k})\}$ are symmetry-adapted in the sense that the translational symmetry has been fully exploited in employing the basis. Thus, the full crystal orbital problem can be divided into separate problems for each wave vector $\boldsymbol{k}$, each problem being of the dimension of the number of atomic orbitals per unit cell. Because of the periodicity of $e^{i\boldsymbol{k}\bullet\boldsymbol{R}}$ (with respect to $\boldsymbol{k}$ in this context), one may restrict oneself to the first *Brillouin* zone (BZ) in $\boldsymbol{k}$ space.

Consider now the one-electron *Schrödinger* equation

$$\hat{H}\Psi_n(\boldsymbol{k}) = E_n(\boldsymbol{k})\Psi_n(\boldsymbol{k}) \tag{2}$$

where the crystal orbitals $\{\Psi_n(\boldsymbol{k})\}$ (CO) with band index n are expanded in terms of the *Bloch* basis functions

$$\Psi_n(\boldsymbol{k}) = \sum_\mu c_{\mu n}(\boldsymbol{k})\phi_\mu(\boldsymbol{k}) \tag{3}$$

Substitution of (3) into the *Schrödinger* equation and left multiplication of both sides by $\phi_\nu^*(\boldsymbol{k})$ and integration over $d\boldsymbol{r}$ gives

$$\sum_\mu c_{\mu n}(\boldsymbol{k}) \int \phi_\nu^*(\boldsymbol{k})\hat{H}\phi_\mu(\boldsymbol{k})d\boldsymbol{r} = E_n(\boldsymbol{k}) \sum_\mu c_{\mu n}(\boldsymbol{k}) \int \phi_\nu^*(\boldsymbol{k})\phi_\mu(\boldsymbol{k})d\boldsymbol{r} \tag{4}$$

With the definitions of

$$H_{\nu\mu}(\boldsymbol{k}) = \left\langle \phi_\nu(\boldsymbol{k}) \middle| \hat{H} \middle| \phi_\mu(\boldsymbol{k}) \right\rangle \tag{5}$$

$$S_{\nu\mu}(\boldsymbol{k}) = \left\langle \phi_\nu(\boldsymbol{k}) \middle| \phi_\mu(\boldsymbol{k}) \right\rangle \tag{6}$$

and after application of the variational principle, equation (4) simplifies to

$$\sum_{\mu} c_{\mu n}(\boldsymbol{k}) H_{\nu\mu}(\boldsymbol{k}) = E_n(\boldsymbol{k}) \sum_{\mu} c_{\mu n}(\boldsymbol{k}) S_{\nu\mu}(\boldsymbol{k}) \tag{7}$$

or the equivalent eigenvalue equation

$$\boldsymbol{H}(\boldsymbol{k})\boldsymbol{c}_n(\boldsymbol{k}) = E_n(\boldsymbol{k})\boldsymbol{S}(\boldsymbol{k})\boldsymbol{c}_n(\boldsymbol{k}) \tag{8}$$

The matrix elements (5) and (6) can be simplified, shown here for $S_{\nu\mu}(\boldsymbol{k})$:

$$S_{\nu\mu}(\boldsymbol{k}) = \int \sum_{\boldsymbol{R}} \sum_{\boldsymbol{R}'} \mathrm{e}^{i\boldsymbol{k}\bullet(\boldsymbol{R}-\boldsymbol{R}')} \chi_\nu^*(\boldsymbol{r}-\boldsymbol{R}')\chi_\mu(\boldsymbol{r}-\boldsymbol{R})d\boldsymbol{r} \tag{9}$$

Setting $\boldsymbol{R}'' = \boldsymbol{R} - \boldsymbol{R}'$ and changing variables: $\boldsymbol{R}'' \Rightarrow \boldsymbol{R}, \quad \boldsymbol{r} \Rightarrow \boldsymbol{r} + \boldsymbol{R}'$ yields

$$S_{\nu\mu}(\boldsymbol{k}) = \sum_{\boldsymbol{R}} \mathrm{e}^{i\boldsymbol{k}\bullet\boldsymbol{R}} \int \chi_\nu^*(\boldsymbol{r})\chi_\mu(\boldsymbol{r}-\boldsymbol{R})d\boldsymbol{r} \tag{10}$$

and in analogy

$$H_{\nu\mu}(\boldsymbol{k}) = \sum_{\boldsymbol{R}} \mathrm{e}^{i\boldsymbol{k}\bullet\boldsymbol{R}} \int \chi_\nu^*(\boldsymbol{k})\hat{H}\chi_\mu(\boldsymbol{k})d\boldsymbol{r} \tag{11}$$

Thus the S and H matrices expressed as a function of the wavevector $\boldsymbol{k}$ can be written in terms of these same matrices as a function of the real vector $\boldsymbol{R}$:

$$\boldsymbol{S}(\boldsymbol{k}) = \sum_{\boldsymbol{R}} \mathrm{e}^{i\boldsymbol{k}\bullet\boldsymbol{R}} \boldsymbol{S}(\boldsymbol{R}) \tag{12}$$

$$\boldsymbol{H}(\boldsymbol{k}) = \sum_{\boldsymbol{R}} \mathrm{e}^{i\boldsymbol{k}\bullet\boldsymbol{R}} \boldsymbol{H}(\boldsymbol{R}) \tag{13}$$

where the elements of the matrices in terms of $\boldsymbol{R}$ are given as the integrals in (10) and (11).

3.2 The extended-Hückel tight-binding method[9],[10]

In the *extended-Hückel tight-binding method,*[9],[10] $\hat{H}$ is given as one-electron operator for the valence electrons. Electron-electron repulsion is neglected. The diagonal elements of $\boldsymbol{H}(\boldsymbol{R})$ over two AOs $\chi_s(\boldsymbol{r})$ in the same unit cell are set equal to the valence orbital ionization potentials (VOIP). All other elements of $\boldsymbol{H}(\boldsymbol{R})$ are approximated by the *Wolfsberg-Helmholz formula*:[11]

$$H_{st}(\boldsymbol{R}) = \tfrac{1}{2}K(H_{ss} + H_{tt})S_{st}(\boldsymbol{R}) \tag{14}$$

K can be a constant[11], or weighted[12]

$$K = \kappa + \Delta^2 + \Delta^4(1-\kappa), \quad \Delta = \frac{H_{ss}-H_{tt}}{H_{ss}+H_{tt}} \tag{15}$$

or be given by the distance-dependent formula[17],[21]

$$K = 1 + (\kappa + \Delta^2 - \Delta^4 \kappa)\mathrm{e}^{-\delta(R-d_0)/q} = 1 + K'\mathrm{e}^{-\delta(R-d_0)/q},\ q = 1 + \{[(R-d_0) - |R-d_0|]\delta\}^2 \tag{16}$$

d_0 is the sum of the orbital radii[17]. κ and δ are empirical parameters ($0.4 \le \kappa \le 1.5, 0 < \delta < 1\text{Å}^{-1}$). The default values $\kappa = 1.0$ and $\delta = 0.35$ Å^{-1} are provided by the EHMACC program.
In the case of the simple or the weighted formula, $\boldsymbol{H}(\boldsymbol{k})$ is expressed as[18]

$$H_{ss}(\boldsymbol{k}) = H_{ss}^{(0,0)}(1 - K + KS_{ss}(\boldsymbol{k})) \tag{17}$$
$$H_{st}(\boldsymbol{k}) = \tfrac{1}{2}K(H_{ss}^{(0,0)} + H_{tt}^{(0,0)})S_{st}(\boldsymbol{k}) \tag{18}$$

In the case of the distance-dependent Wolfsberg-Helmholz formula one gets[16]

$$H_{ss}(\boldsymbol{k}) = H_{ss}^{(0,0)}S_{ss}(\boldsymbol{k}) + 2H_{ss}^{(0,0)}K' \sum_{m=1}^{M} \cos(\boldsymbol{k} \bullet \boldsymbol{R}_m)S_{ss}^{(0,m)}\mathrm{e}^{-\delta(R_{ss}^{(0,m)}-d_0)/q_{ss}^{(0,m)}} \tag{19}$$
$$H_{st}(\boldsymbol{k}) = \tfrac{1}{2}(H_{ss}^{(0,0)} + H_{tt}^{(0,0)})(S_{st}(\boldsymbol{k}) + K'S'_{st}(\boldsymbol{k})) \tag{20}$$
$$S_{st}(\boldsymbol{k}) = S_{st}^{(0,0)} + \sum_{m=1}^{M} \cos(\boldsymbol{k} \bullet \boldsymbol{R}_m)(S_{st}^{(0,m)} + S_{ts}^{(0,m)}) + i\sum_{m=1}^{M} \sin(\boldsymbol{k} \bullet \boldsymbol{R}_m)(S_{st}^{(0,m)} - S_{ts}^{(0,m)}) \tag{21}$$
$$\begin{aligned} S'_{st}(\boldsymbol{k}) = {} & S_{st}^{(0,0)}\mathrm{e}^{-\delta(R_{st}^{(0,0)}-d_0)/q_{st}^{(0,0)}} + \\ & + \sum_{m=1}^{M} \cos(\boldsymbol{k} \bullet \boldsymbol{R}_m)\left(S_{st}^{(0,m)}e^{-\delta(R_{st}^{(0,m)}-d_0)/q_{st}^{(0,m)}} + S_{ts}^{(0,m)}e^{-\delta(R_{ts}^{(0.m)}-d_0)/q_{ts}^{(0,m)}}\right) + \\ & + i\sum_{m=1}^{M} \sin(\boldsymbol{k} \bullet \boldsymbol{R}_m)\left(S_{st}^{(0,m)}e^{-\delta(R_{st}^{(0,m)}-d_0)/q_{st}^{(0,m)}} - S_{ts}^{(0,m)}e^{-\delta(R_{ts}^{(0.m)}-d_0)/q_{ts}^{(0,m)}}\right) \end{aligned} \tag{22}$$

m is an index formed by combination of three indices (m_1, m_2, m_3) running over the three translation directions of the three-dimensional periodic crystal. In analogy m can also be written for one- and two-dimensional crystals. The superscripts $(0, m)$ denote the interaction between the origin unit cell and the m-th unit cell. One performs the summation over M unit cells in each direction. The truncation criterion M is estimated from the dependence of the overlap integrals between two AOs on the distance $R_{st}^{(0,m)}$. M can be specified by the number of neighbour unit cells within the keyword CELL (cf. Chapter 4.1.3).
The extended-Hückel total energy E_{EHMO} per unit cell for closed-shell systems is obtained by summing over all occupied one-electron levels $E_n(\boldsymbol{k})$ and by integration over the whole *Brillouin* zone:

$$E_{EHMO} = \frac{1}{V_{BZ}} \int_{BZ} 2\sum_{n}^{occ} E_n(\boldsymbol{k})d\boldsymbol{k} \tag{23}$$

In open-shell systems first the *Fermi* level ε_f as a function of the electron occupation has to be evaluated by integrating the density of states $g(E)$, which gives the total number of electrons:

$$n(\varepsilon_f) = \int_{E_b}^{\varepsilon_f} g(E)dE \tag{24}$$

$$E_{EHMO} = \int_{E_b}^{\varepsilon_f} g(E) E dE \tag{25}$$

E_b is the lowest energy level given by the calculation. The stabilization energy ΔE_{EHMO} per unit cell corresponds to the electronic energy change of the whole system with respect to energy of the free atoms:

$$\Delta E_{EHMO} = E_{EHMO} - \sum_s b_s^0 H_{ss} \tag{26}$$

The b_s^0 are the occupation numbers of the AO levels of the free atoms. The total energy E_{tot} is obtained by adding an electrostatic repulsion term E_{Rep} to ΔE_{EHMO}:

$$E_{tot} = \Delta E_{EHMO} + E_{Rep} \tag{27}$$

$$E_{Rep} = \frac{1}{2} \sum_m^M \sum_\mu \sum_\nu E_{Rep_{\mu\nu}}^{(0,m)} \tag{28}$$

The summation runs over all unit cells and all over all interactions $E_{Rep_{\mu\nu}}$ between two centers μ and ν.[16] The truncation criterion M can be taken the same as for the overlap integrals, because the electrostatic repulsion interaction terms are rapidly declining with distance. The electrostatic interaction between two centers is given by[17]

$$E_{Rep_{\mu\nu}} = \frac{Z_\mu Z_\nu}{R_{\mu\nu}} - \frac{1}{2}\left(Z_\mu \int_0^\infty \frac{\rho_\nu}{|R_{\mu\nu} - r|} dr + Z_\nu \int_0^\infty \frac{\rho_\mu}{|R_{\mu\nu} - r|} dr \right) \tag{29}$$

$$\int_0^\infty \frac{\rho_\mu}{|R_{\mu\nu} - r|} dr = \frac{1}{R_{\mu\nu}} \sum_{nl} b_{\mu,nl}^0 \left[1 - \frac{e^{-2\zeta_{\mu,nl} R_{\mu\nu}}}{2n} \sum_{p=1}^{2n} (2 R_{\mu\nu} \zeta_{\mu,nl})^{2n-p} \frac{p}{(2n-p)!} \right] \tag{30}$$

The total energy E_{tot} is calculated when using the keyword VARIATION and the card VARY.

3.3 How to choose a set of k-points

One must first answer the question: „What information is needed - average properties or band structure?" Different types of **k**-point sets are needed for each, thus it is most often necessary to run a calculation twice - once for average properties like the total energy, the density of states (DOS) or crystal orbital overlap population (COOP), and once for the band structure.

3.3.1 k-point sets for average properties

If average properties are needed, then a mesh of **k**-points must be created. This mesh should sample „typical" points of the *Brillouin* zone (BZ). Thus high symmetry points and lines1) should be avoided, if possible, since these are not representative of the vast majority of points in the BZ. The mesh need not span the full BZ, since regions of the BZ can be related 1) by symmetry elements of

---

1) For the definition of these terms see for example [4]

the space group and 2) by inversion symmetry, i.e. $f(\mathbf{k}) = f(-\mathbf{k})$. The smallest region from which the full BZ can be produced by one of these two procedures is called the „irreducible wedge", or irreducible *Brillouin* zone IBZ. Note, that the mesh should have the full symmetry of the space group.

Example: 2D tetragonal system: A square net of carbon atoms with lattice vector of length a.

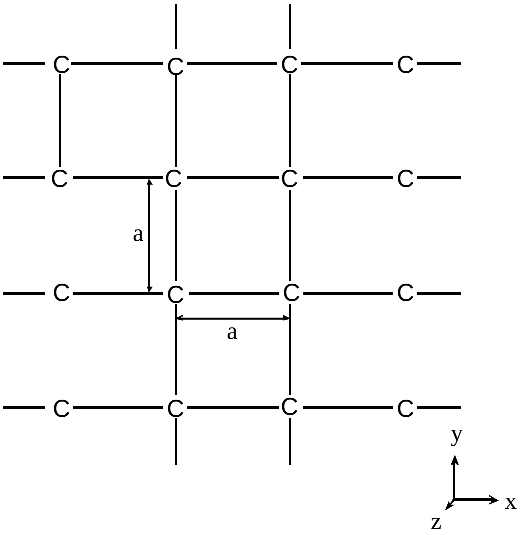


Figure 6: Square carbon lattice. The unit cell contains one atom.

Due to the mirror planes perpendicular to the x and to the y axis and diagonal in between, the full *Brillouin* zone reduces to the irreducible wedge with 1/8 of the original area:

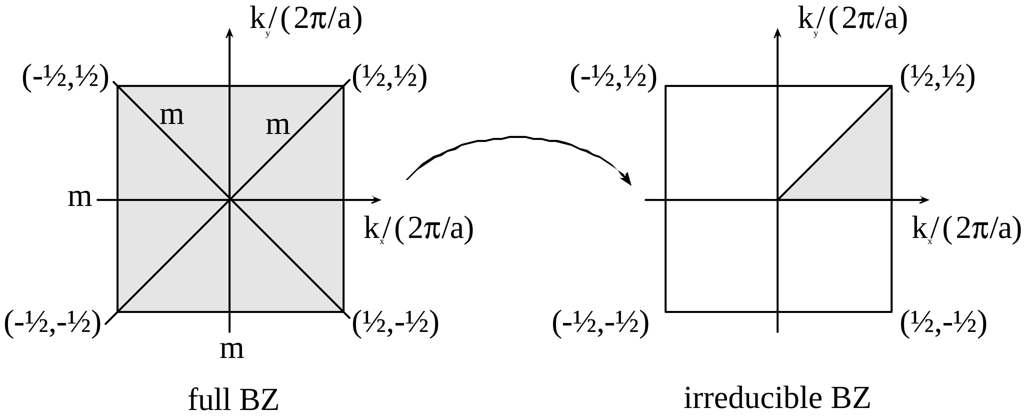


Figure 7: Full and irreducible square BZ.

Example: **k**-point set with full symmetry of space group and offset from zone edges to avoid most high symmetry lines (cf. example file `c_gdos.ehi`).

k_x	k_y	k_z	weight
0.05	0.05	0	1
0.15	0.05	0	2
0.15	0.15	0	1
0.25	0.05	0	2
0.25	0.15	0	2
0.25	0.25	0	1
0.35	0.05	0	2
0.35	0.15	0	2
0.35	0.25	0	2
0.35	0.35	0	1
0.45	0.05	0	2
0.45	0.15	0	2
0.45	0.25	0	2
0.45	0.35	0	2
0.45	0.45	0	1

Table 2: 15 **k**-points of square IBZ.

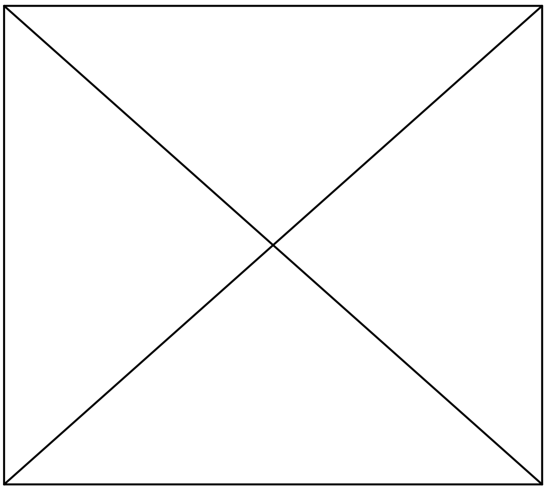


Figure 8: Properties **k**-point set for square BZ

weight counts the number of times an equivalent **k**-point appears in the full BZ. In this particular case, the points along $k_x = k_y$ appear half as often as the general points. The point (0,0,0) would appear one-eight as often as the general points.

Efficient **k**-point sets for the two-dimensional Brillouin zones, based on a geometrical method, have been described by [23]. For three-dimensional BZ a method based on plane waves has been proposed [28] and generalized [24].

Several programs to generate sets in the required size are available, see chapter 4.6.1.

3.3.2 k-point sets for band structures

By design, these points must lie along high symmetry lines. Typically 5 or 6 points along each line, including the end points, are sufficient for generation of a smooth band.

Example: **k**-point set for band structure of above example of square carbon lattice (cf. example file `c_gitter.ehi`)

k_x	k_y	k_z
0.000	0.000	0.000
0.125	0.000	0.000
0.250	0.000	0.000
0.375	0.000	0.000
0.500	0.000	0.000
0.500	0.125	0.000
0.500	0.250	0.000
0.500	0.375	0.000
0.500	0.500	0.000
0.375	0.375	0.000
0.250	0.250	0.000
0.125	0.125	0.000

Table 3: Symmetry line k-points of

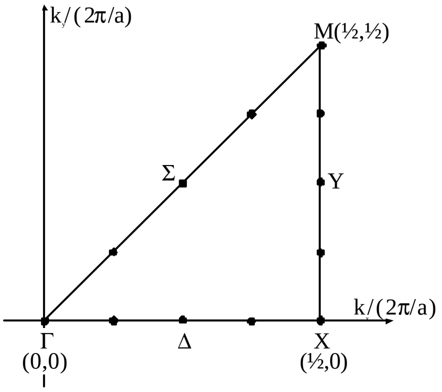


Figure 9: Band structure **k**-point set of square BZ

square BZ

The high symmetry points and lines of the 2D *Brillouin* zones are described in [23], for the 3D BZ in [4], [2] or [5].

3.3.3 Symmetry analysis

The correct crossing of bands requires the knowledge of the symmetry properties of their respective crystal orbitals. In order to determine the irreducible representation of each crystal orbital at each point along a specific symmetry line, it is first necessary to determine the point group of the line, or equivalently, to find which of the symmetry elements of the full space group are retained along that symmetry line.

The star of **k** is defined as the result of the operation of a set of symmetry elements on a **k**-vector. If the star of **k** remains **k** for all **k**-vectors defining a symmetry line for a given set of symmetry operations, then those symmetry elements are retained along that line. Thus a symmetry element which takes **k** into -**k**, or into some other vector, is not retained. The group for which $\boldsymbol{Symmop}\,\boldsymbol{k} = \boldsymbol{k}$, **Symmop** being an element of the point group of the *Bravais* lattice, is called little co-group [4].

Example: 2D tetragonal system: Square carbon net, see Figure 6. The symmetry elements retained along the Δ symmetry line (**k** ranging from (0,0,0) to (½,0,0)) are:

C_2 colinear with x axis
σ_v in xy plane
σ_h in xz plane

These operations do not change the direction of any vector (k_x,0,0), $k_x \in [0\ldots0.5]$. The point group of the symmetry line is C_{2v} having four irreducible representations.

The symmetry analysis part of the band structure program EHMACC is based on the transformation of crystal orbital coefficients under a symmetry operation of the point group of a high symmetry line of the irreducible *Brillouin* zone [19]. It is implemented only for the crystal orbital part of EHMACC. The program is not able to determine the space group of the solids and therefore the point groups of the underlying symmetry lines. Instead the user has to provide these symmetry operations which are valid for each symmetry line. This has the advantage that the solid-state structure can be oriented freely for users convenience. It is only necessary to define the minimal set of symmetry operations so that the irreducible representations of the crystal orbitals are completely determined.

Example: For the above square carbon lattice, the following symmetry operations are sufficient:

Δ line: σ_h in xy plane, σ_v in xz plane

Y line: σ_h in xy plane
Σ line: σ_h in xy plane, $\sigma_v \perp$ xy plane and parallel to Σ line

The symmetry analysis part applies each of these symmetry operations onto the CO coefficients and generates a new set of coefficients. These are compared with the generating set, resulting in a **+1** for a symmetric (congruent) CO with respect to a symmetry operation, a **-1** for antisymmetric CO with respect to a symmetry operation, and a **0** for a degenerate CO (or a wrongly defined symmetry operation).

Restrictions: Because the crystal wave functions are not symmetry adapted with respect to point group symmetry, degenerate wave functions can not be transformed in the correct manner. Only rotation axes and mirror planes are available. Since glide planes and screw axes are not implemented, no additional conditions for further distinction of bands are supplied.

3.4 Interpolation of energy bands

In principle a band structure along a symmetry line should be calculated at an infinite number of $\boldsymbol{k}$ -points along this line. Since the eigenvalues $E_n(\boldsymbol{k})$ and their gradients $\nabla_{\boldsymbol{k}} E_n(\boldsymbol{k})$ with respect to $\boldsymbol{k}$ are continuous functions, it suffices to calculate the energy levels at a few discrete $\boldsymbol{k}$-points and then to interpolate them by an accurate analytical function. Such an interpolation has been proposed by Kertész and Hughbanks [15]. It is based on symmetry-adapted Fourier functions (or plane waves) as basis set:

$$A_m(\boldsymbol{k}) = \sum_{\alpha \in G_0} \exp(i\boldsymbol{k} \bullet \alpha \boldsymbol{R}_m) \tag{31}$$

$\boldsymbol{R}_m$ is a lattice vector of the direct Bravais lattice. The summation runs over all symmetry elements α of the point group G_0 of the crystal. Therefore the symmetry properties of the $A_m(\boldsymbol{k})$-functions are the same than those of the energy bands. Since inversion symmetry is supposed, these functions are real. We require, that the expansion

$$E(\boldsymbol{k}_i) = \langle E \rangle + \sum_m{}' b_m A_m(\boldsymbol{k}_i) \tag{32}$$

runs through all calculated eigenvalues. The primed summation leaves out $\boldsymbol{R}_m = 0$. $\langle E \rangle$ is the mean energy of a band n under consideration. It is further required that the derivatives

$$\left.\frac{\partial E}{\partial k_s}\right|_{\boldsymbol{k}_i} = -\sum_m \sum_{\alpha \in G_0} (\alpha \boldsymbol{R}_m)_s \sin(\boldsymbol{k}_i \bullet \alpha \boldsymbol{R}_m), \quad s = 1, \ldots, d \tag{33}$$

are equal to the analytical derivatives (36). d is the dimension of the periodic lattice. The „roughness”

$$\rho = \sum_{m} |b_m|^2 \exp(\sigma |\boldsymbol{R}|_m) \tag{34}$$

is minimized by damping the expansion coefficients b_m to small lattice vectors with an exponential term in order to avoid oscillating Fourier components, see [15].
The analytic gradients of the energies with respect to $\boldsymbol{k}$ are obtained by deriving the secular equation and by using the normalization condition

$$\boldsymbol{c}_n^T(\boldsymbol{k})\boldsymbol{S}(\boldsymbol{k})\boldsymbol{c}_n(\boldsymbol{k}) = 1 \tag{35}$$

With the standard Wolfsberg-Helmholz approximation one obtains [20]

$$\tfrac{\partial E_n(\boldsymbol{k})}{\partial \boldsymbol{k}} = \boldsymbol{c}_n^T(\boldsymbol{k})\Big(\tfrac{\partial \boldsymbol{H}(\boldsymbol{k})}{\partial \boldsymbol{k}} - E_n(\boldsymbol{k})\tfrac{\partial \boldsymbol{S}(\boldsymbol{k})}{\partial \boldsymbol{k}}\Big) \tag{36}$$

$$\tfrac{\partial}{\partial \boldsymbol{k}} H_{st}(\boldsymbol{k}) = \tfrac{1}{2} K(H_{ss}^{(0,0)} + H_{tt}^{(0,0)}) \tfrac{\partial}{\partial \boldsymbol{k}} S_{st}(\boldsymbol{k}) \tag{37}$$

$$\tfrac{\partial}{\partial \boldsymbol{k}} S_{st}(\boldsymbol{k}) = -2\pi \sum_{m}^{M} m \sin(\boldsymbol{k} \bullet \boldsymbol{R}_m)(S_{st}^{(0,m)} + S_{ts}^{(0,m)}) + 2\pi i \sum_{m}^{M} m \cos(\boldsymbol{k} \bullet \boldsymbol{R}_m)(S_{st}^{(0,m)} - S_{st}^{(0,m)}) \tag{38}$$

m runs over the unit cells of the crystal. Application of the distance-dependent Wolfsberg-Helmholz approximation yields [16]

$$\tfrac{\partial}{\partial \boldsymbol{k}} H_{st}(\boldsymbol{k}) = \tfrac{1}{2}(H_{ss}^{(0,0)} + H_{tt}^{(0,0)})\Big[\tfrac{\partial}{\partial \boldsymbol{k}} S_{st}(\boldsymbol{k}) + K' \tfrac{\partial}{\partial \boldsymbol{k}} S'_{st}(\boldsymbol{k})\Big] \tag{39}$$

$$\begin{aligned} \tfrac{\partial}{\partial \boldsymbol{k}} S'_{st}(\boldsymbol{k}) = & -2\pi \sum_{m}^{M} m \sin(\boldsymbol{k} \bullet \boldsymbol{R}_m)\Big(S_{st}^{(0,m)} \mathrm{e}^{-\delta(R_{st}^{(0,m)} - d_0)/q_{st}^{(0,m)}} + S_{ts}^{(0,m)} \mathrm{e}^{-\delta(R_{ts}^{(0,m)} - d_0)/q_{ts}^{(0,m)}}\Big) + \\ & + 2\pi i \sum_{m}^{M} m \cos(\boldsymbol{k} \bullet \boldsymbol{R}_m)\Big(S_{st}^{(0,m)} \mathrm{e}^{-\delta(R_{st}^{(0,m)} - d_0)/q_{st}^{(0,m)}} - S_{ts}^{(0,m)} \mathrm{e}^{-\delta(R_{ts}^{(0,m)} - d_0)/q_{ts}^{(0,m)}}\Big) \end{aligned} \tag{40}$$

3.5 Evaluation of density of states (DOS) and DOS-weighted properties[1],[25]

Often one wants to calculate quantities which contain one-electron properties $Q_n(\boldsymbol{k})$ as weigh- ted sums over the energy levels

$$Q = 2 \sum_{n,\boldsymbol{k}} Q_n(\boldsymbol{k}) \tag{41}$$

The sum runs over all $\boldsymbol{k}$ in the unit cell for every band n. In the case of an infinite crystal, the possible $\boldsymbol{k}$-values come close together, and the sum may be replaced by an integral over one single unit cell:

$$q = \lim_{V \to \infty} \frac{Q}{V} = 2 \sum_n \int \tfrac{1}{(2\pi)^3} Q_n(\boldsymbol{k}) d\boldsymbol{k} \tag{42}$$

If $Q_n(\boldsymbol{k})$ only depends on n and $\boldsymbol{k}$ over the energy $E_n(\boldsymbol{k})$ one can define a density of states per unit volume $g(E)$ (or simply density of states DOS), so that

$$q = \int g(E) Q(E) dE \tag{43}$$

$$g(E) = \sum_n g_n(E) \tag{44}$$

The density of states $DOS(E)$ counts all energy levels $E_n(\boldsymbol{k})$ in an energy range $\Delta E = E \ldots E + dE$:

$$DOS(E)\Delta E = \Delta N, \quad DOS(E) = \left(\frac{\Delta E}{\Delta N}\right)^{-1} \tag{45}$$

Be $\bar{q}$ now a property we want to calculate in analogy to eq. (43) as integral over one-electron properties $Q(\boldsymbol{k})$. If $Q(\boldsymbol{k})$ contains the point group symmetry of the *Bravais* lattice, we have to determine the integral only over the irreducible *Brillouin* zone (IBZ):

$$\bar{q} = \frac{1}{V_{IBZ}} \int_{IBZ} Q(\boldsymbol{k}) d\boldsymbol{k} \tag{46}$$

Normally the property $Q(\boldsymbol{k})$ does not possess a simple analytical form, so that its calculation gets complicated and time-consuming. Moreover an accurate determination of eq. (46) would require the knowledge of $Q(\boldsymbol{k})$ at an infinite number of $\boldsymbol{k}$-points in the IBZ. One tries to estimate the integral by dividing the integrand $Q(\boldsymbol{k})$ into an interpolating function $\varphi(\boldsymbol{k})$ and a reminder $O(\boldsymbol{k})$ which is neglected furthermore but can be used for estimation of the accuracy of the integral over $\varphi(\boldsymbol{k})$. The interpolating function is written as a linear combination of a basis set $\{\varphi_i\}$:

$$\varphi(\boldsymbol{k}) = \sum_{i=1}^{N} a_i \varphi_i(\boldsymbol{k}) \tag{47}$$

The coefficients a_i may be determined from the set of equations

$$Q(\boldsymbol{k}_j) = \sum_{i=1}^{N} a_i \varphi_i(\boldsymbol{k}_j), \quad j = 1, 2, \ldots, N \tag{48}$$

If $\left|\varphi_i(\boldsymbol{k}_j)\right| \neq 0$, then one gets coefficients linearly dependent on $Q(\boldsymbol{k}_j)$, and one replaces

$$\frac{1}{V_{IBZ}} \int_{IBZ} Q(\boldsymbol{k}) d\boldsymbol{k} = \sum_{i=1}^{N} w_i Q(\boldsymbol{k}_i) \tag{49}$$

The w_i are weighting factors normally scaling to the inverse of the order of the point group $G_0(\boldsymbol{k})$. The diverse *Ansätze* for the expansion functions $\{\varphi_i\}$ are described in [25].
In practice, a few dozens of $\boldsymbol{k}$-points suffice for an accurate determination of DOS, COOPs, total energies or of the *Fermi* level. For the graphical representation of the former three pro- perties one replaces the delta-function in

$$g_n(E) = \int \frac{1}{4\pi^3}\delta(E - E_n(\boldsymbol{k}))d\boldsymbol{k} \tag{50}$$

by Gaussian functions $\exp(-(E-E_i)^2/\sigma^2)$ being centered on equidistant energy mesh points E_i. The area below the gaussian curve is attributed to the amplitude at the drawing point E_i. It corresponds to the number of calculated energy levels being counted in the range $E_i - x\sigma \ldots E_i + x\sigma$ and weighted by the respective property f. σ is the half-width of the gaussian and can be estimated from the dispersion of the energy bands. x is determined by a criterion, below which the value of the gaussian is set to zero (see chapter 4.2.3, keywords **DELTA** and **LIMIT**), see Table 4.

Table 4: Relation between truncation criterion LIMIT and number of halfwidths x.

LIMIT	$x = \sqrt{-\ln(LIMIT)}$
10^{-5}	3.39
10^{-4}	3.03
10^{-3}	2.63
10^{-2}	2.15
10^{-1}	1.52
1	0.00

3.6 Converting from the AO to the FMO (fragment MO) basis set[13]

The crystal orbitals can also be expressed in terms of a *Bloch* basis set derived from fragment molecular orbitals (FMOs) of the unit cell, rather than atomic orbitals. The fragment *Bloch* basis set $\{\Phi_a\}$ is expressed in terms of the unit cell FMOs $\{\mathrm{X}_a\}$:

$$\Phi_a = \frac{1}{\sqrt{N}}\sum_{\boldsymbol{R}} \mathrm{e}^{i\boldsymbol{k}\bullet\boldsymbol{R}}\mathrm{X}_a(\boldsymbol{r}-\boldsymbol{R}) \tag{51}$$

Any member of the FMO basis set $\{\Phi_a(\boldsymbol{k})\}$ can be written as linear combination of *Bloch* basis sets over AOs $\{\phi_\mu(\boldsymbol{k})\}$:

$$\Phi_a(\boldsymbol{k}) = \sum_{\mu} b_{\mu a}\phi_\mu(\boldsymbol{k}) \tag{52}$$

The crystal orbitals $\{\Psi_n(\boldsymbol{k})\}$ can then be expressed as

$$\Psi_n(\boldsymbol{k}) = \sum_a d_{an}(\boldsymbol{k})\Phi_a(\boldsymbol{k}) = \sum_a d_{an}(\boldsymbol{k}) \sum_\mu b_{\mu a}\phi_\mu(\boldsymbol{k}) \tag{53}$$

It is the matrix $\boldsymbol{D}$ which we wish to solve for. From the definition of the crystal orbitals $\{\Psi_n(\boldsymbol{k})\}$ in terms of AO *Bloch* functions follows

$$\sum_\mu c_{\mu n}(\boldsymbol{k})\phi_\mu(\boldsymbol{k}) = \sum_a d_{an}(\boldsymbol{k}) \sum_\mu b_{\mu a}\phi_\mu(\boldsymbol{k}) \tag{54}$$

$$\therefore \quad c_{\mu n}(\boldsymbol{k}) = \sum_a d_{an}(\boldsymbol{k}) b_{\mu a} \tag{55}$$

or in matrix formulation:

$$\boldsymbol{C} = \boldsymbol{B}\boldsymbol{D} \tag{56}$$

Solve for $\boldsymbol{D}$:

$$\boldsymbol{D} = \boldsymbol{B}^{-1}\boldsymbol{C} \qquad \text{or} \qquad d_{an}(\boldsymbol{k}) = \sum_\mu (b^{-1})_{a\mu} c_{\mu n}(\boldsymbol{k}) \tag{57}$$

$\boldsymbol{B}^{-1}$ can be determined from the normalisation condition for the FMOs α and β in the same fragment

$$\langle \Phi_a | \Phi_\beta \rangle = \delta_{a\beta} \tag{58}$$

In matrix notation, this can be written as

$$\boldsymbol{B}^T \boldsymbol{S}_{block} \boldsymbol{B} = \mathbf{1} \ (\mathbf{1}\text{=identity matrix}) \tag{59}$$

$$\boldsymbol{B}^{-1} = \boldsymbol{B}^T \boldsymbol{S}_{block} \tag{60}$$

where $\boldsymbol{S}_{block}$ is the $\boldsymbol{S}$ matrix for the fragments in the unit cell taken to be in block form such that the matrix elements between AOs on different fragments are zero. Use this expression to substitute for $\boldsymbol{B}^{-1}$:

$$\boldsymbol{D} = \boldsymbol{B}^T \boldsymbol{S}_{block} \boldsymbol{C} \tag{61}$$

3.7 Mulliken population analysis

For the AO basis $\{\phi_\mu(\boldsymbol{k})\}$, using bra-ket notation, the crystal orbitals $\{\Psi_n(\boldsymbol{k})\}$ are written

$$|\Psi_n(\boldsymbol{k})\rangle = \sum_\mu c_{\mu n}(\boldsymbol{k}) |\phi_\mu(\boldsymbol{k})\rangle \tag{62}$$

The „charge” in each AO $\phi_\mu(\boldsymbol{k})$ for the n$^{\text{th}}$ crystal orbital $\Psi_n(\boldsymbol{k})$ is

$$\begin{aligned} Q^{AO}_{\mu n}(\boldsymbol{k}) &= 2c^*_{\mu n}(\boldsymbol{k})\langle\phi_\mu(\boldsymbol{k})|\Psi_n(\boldsymbol{k})\rangle = 2c^*_{\mu n}(\boldsymbol{k})\sum_\nu c_{\nu n}(\boldsymbol{k})\langle\phi_\mu(\boldsymbol{k})|\phi_\nu(\boldsymbol{k})\rangle \\ &= 2c^*_{\mu n}(\boldsymbol{k})[\boldsymbol{S}(\boldsymbol{k})\bullet\boldsymbol{C}(\boldsymbol{k})]_{\mu n} \end{aligned} \tag{63}$$

For the FMO basis, we want the „charge” in each FMO $\Phi_\alpha(\boldsymbol{k})$ for the n^{th} crystal orbital $\Psi_n(\boldsymbol{k})$

$$Q^{FMO}_{\alpha n}(\boldsymbol{k}) = 2d^*_{\alpha n}(\boldsymbol{k})\sum_\beta d_{\beta n}\langle\Phi_\alpha(\boldsymbol{k})|\Phi_\beta(\boldsymbol{k})\rangle \tag{64}$$

Defining

$$V_{\alpha\beta}(\boldsymbol{k}) \equiv \langle\Phi_\alpha(\boldsymbol{k})|\Phi_\beta(\boldsymbol{k})\rangle = \sum_{\mu\nu} b^*_{\alpha\mu}S_{\mu\nu}(\boldsymbol{k})b_{\nu\beta} \quad \text{or} \quad \boldsymbol{V} = \boldsymbol{B}^T\boldsymbol{S}\boldsymbol{B} \tag{65}$$

then:

$$Q^{FMO}_{\alpha n}(\boldsymbol{k}) = 2d^*_{\alpha n}(\boldsymbol{k})\sum_\beta d_{\beta n}(\boldsymbol{k})V_{\alpha\beta}(\boldsymbol{k}) = 2d^*_{\alpha n}(\boldsymbol{k})[\boldsymbol{V}\boldsymbol{D}]_{\alpha n} \tag{66}$$

This can be written in terms of $\boldsymbol{S}$ using

$$\boldsymbol{V}\boldsymbol{D} = \boldsymbol{B}^T\boldsymbol{S}\boldsymbol{B}\boldsymbol{B}^{-1}\boldsymbol{C} = \boldsymbol{B}^T\boldsymbol{S}\boldsymbol{C} \tag{67}$$

$$Q^{MO}_{\alpha n}(\boldsymbol{k}) = 2d^*_{\alpha n}(\boldsymbol{k})[\boldsymbol{B}^T\boldsymbol{S}\boldsymbol{C}]_{\alpha n} \tag{68}$$

Summarising for both AO and FMO cases, if we let $\boldsymbol{T}(\boldsymbol{k}) = \mathbf{S}(\boldsymbol{k})\boldsymbol{C}(\boldsymbol{k})$:

$$Q^{AO}_{\mu n}(\boldsymbol{k}) = 2c^*_{\mu n}(\boldsymbol{k})\boldsymbol{T}_{\mu n}(\boldsymbol{k}) \tag{69}$$

$$Q^{FMO}_{\alpha n}(\boldsymbol{k}) = 2d^*_{\alpha n}(\boldsymbol{k})[\boldsymbol{B}^T\boldsymbol{T}(\boldsymbol{k})]_{\alpha n} \tag{70}$$

3.8 Oscillator strengths

For the calculation of oscillator strengths in the solid we refer to Ref. [22].

3.9 Parameter sets

For a discussion of Slater exponents and coefficients we refer to [53], [58-72].

For a discussion of Coulomb integrals H_{ii} and valence orbital ionization energies VOIE we refer to [45],[53-58].

Below we give a Table of references in which extended-Hückel parameters for the corresponding elements are reported. The references are not complete, they can, however, be used as a guide. We recommend in every case to check the parameters.

Table 5: References for Slater exponents of the elements.

Element	Reference	Element	Reference	Element	Reference
Ac	[77]	He	[90]	Pd	[99]
Ag	[78]	Hf	[77]	Pt	[85]
Al	[79],[76]	Hg	[91],[69]	Rb	[105]
As	[79]	I	[92]	Re	[100]
Au	[81]	In	[88]	Rh	[85],[101]
B	[76]	Ir	[93]	Ru	[89],[102],[116],[117]
Be	[80]	K	[83]	S	[75]
Bi	[82]	La	[94]	Sb	[103]
Br	[83]	Li (Li_2)	[76]	Sc	-
C	[10],[75]	Li(metal)	[95]	Se	[104],[106]
Ca	[84]	Mg	[96]	Si	[74],[78],[107],[52]
Cd	[69]	Mn	[54],[85]	Sm	[77]
Ce	[77]	Mn(metal)	[14]	Sn	[105]
Cl	[85]	Mo	[85]	Sr	[105]
Co	[86],[73]	N	[75]	Ta	[108]
Co(metal)	[14]	Na (Na_2)	[76]	Tc	[109]
Cr	[85]	Na(metal)	[95]	Te	[110],[69]
Cr(metal)	[85]	Ne	[90]	Th	[111]
Cs	[83]	Nb	[85]	Tl(metal)	[14]
Cu	[87]	Ni	[97]	U	[111]
F (F_2)	[76]	Ni(metal)	[14]	V	[112]
Fe	[85],[86]	O	[75]	V(metal)	[14]
Fe(metal)	[14]	Os	[116]	W	[100]
Ga	[88]	P	[85]	Yb	[113]
Gd	[77]	Pa	[77]	Zn	[115]
Ge	[89]	Pb	[95],[69]	Zr	[114]
H	[10]				

3.10 Charge iteration

This part has not yet been implemented.

4 PROGRAMS

4.1 EHMACC

4.1.1 Description

EHMACC is a program for extended-Hückel molecular and crystal calculations. It allows MO calculations on molecules and band orbital calculations on solids within the framework of the extended-Hückel method.

Originally written by M.-H. Whangbo in the laboratory of R. Hoffmann, Dept. of Chemistry, Cornell University, Ithaca, N.Y., Dec. 1976.

Later modifications and additions by T. Hughbanks, M. Kertesz, S. Wijeyesekera, C. Wilker and C. Zheng in the laboratory of R. Hoffmann, Dept. of Chemistry, Cornell University, Ithaca, N.Y.

Modified by M. Evain in the laboratory of M.-H. Whangbo, Dept of Chemistry, North Carolina State University, Raleigh N.C., Dec. 1987.

New features include a distance-dependent weighted Wolfsberg-Helmholz approximation, an electrostatic core-core repulsion term, an automated geometry variation, and a symmetry analysis of the crystal orbitals, added by M. Brändle in the laboratory of G. Calzaferri, Department of Chemistry and Biochemistry, University of Berne, Freiestr. 3, CH-3012 Berne, Switzerland.

4.1.2 Invoking the program, input and output files

All input and output files will be generated in ASCII format. Scratch files are written in binary format. Filenames are free, but fixed extensions have to be used.

Input:	`filename.ehi`	- Holds commands, structure information and parameters for extended-Hückel MO or band calculation.
Output:	`filename.eho`	- Collects all results of an EH band/MO calculation.
	`filename.ehc`	- Holds cartesian atom positions generated with ZMAT keyword. This file is formatted for input to the high-quality molecule and crystal structure display program SCHAKAL[35].
	`filename.ehb`	- Holds bonds generated with keyword BOND.
	`filename.ehe`	- Holds energies and energy gradients for interpolation of bands.
	`filename.eh1`	- Holds information about number of atoms, energies etc.
	`filename.eh2`	- Holds information about AO population.
	`filename.eh3`	- Holds information about COOP.
	`filename.eh4`	- Holds information about FMO population.
	`filename.ene`	- Holds total electronic energy E_{EHMO}, stabilisation energy DE_{EHMO}, repulsion energy E_{Rep}, total energy $E_{tot} = DE_{EHMO}+E_{Rep}$, and *Fermi* level as function of varied coordinates.
	`###.kpe`	- Holds wavefunctions, energies, and geometry at each **k**-point calculated. Only produced with keyword EDITOUT.

Scratch:	`ehmacc.s07`	- Holds intermediate results of FMO calculation.
	`ehmacc.s09`	- Holds intermediate Mulliken charges.
	`ehmacc.s11`	- Holds intermediate results of FMO calculation.
	`ehmacc.s13`	- Holds intermediate derivatives of Bloch summed overlap integral elements versus **k** for band gradient calculation.
	`ehmacc.s20`	- Holds **k**-point weights and eigenvalues for total energy calculation.
	`ehmacc.s21`	- Holds overlap matrix and pointers from atom to orbital numbers.
	`ehmacc.s22`	- Holds Hückel matrix and pointers from atom to orbital numbers.
	`ehmacc.s23`	- Holds Hückel matrix elements for selection with AOKEEP or AODELETE option, **S**(**k**) and **C**(**k**), and for intermediate results in band gradient calculation.

Windows NT version:

1. Open a DOS command prompt window.
2. Type `ehmacc.bat inputfilename`

   No extension is needed: An input file with the extension `.ehi` is assumed.

   If you intend to start a calculation with the **RESTART** option (see keywords below), type `ehmacc.bat inputfilename restart`

4.1.3 Format of the input file (`filename.ehi`, UNIT 5)

All input to the program is unformatted. Spaces may be set anywhere except, of course, within an independent item (card name, a number or a chemical symbol). Spaces must be set to separate independent items. A physical line of input may not exceed a total of 80 characters. The program handles capital letters as well as lowercase letters and does not distinguish between both of them.

The input file `filename.ehi` is divided into sections („cards") .The sequence of the cards described below should be maintained. No input program is provided, instead the user can use his favourite editor for file creation. Blank lines may be inserted anywhere for user's convenience. Any line containing an exclamation mark **!** anywhere is considered a comment.

Card #**1 TITLE**

Number : 1
Requirement : None
Object : Title for user's convenience.
Sequence : None.
Example : `TITLE Polyacetylene`

Card #**2 KEYWRD** keywords

Number : Unlimited.
Requirement : None.
Object : Controls the input sequence, the calculation path and the output options.
Sequence : None. The keywords must be separated by either commas or blanks.
Example : `KEYWRD BAND EOUT DISDEPWH`
`KEYWRD COOUT`
`KEYWRD RHO=20`

List of keywords:

AOOUT : Select all atomic orbital (AO) coefficients for each molecular orbital (MO) or crystal orbital (CO) for printing purposes. Printing is released by card **KPOUT**.

AOOUT=m TO n : Choose AO m through AO n for printing purposes. Printing is released by card **KPOUT**.

AOPOP : Perform an AO population analysis. The results are written to `filename.eh2`.

AOPOP=m : Perform an AO population analysis from CO #1 to CO #m (use only with band).The results are written to `filename.eh2`.

AUFBAU : Only valid together with keyword **EAOORB** or **REPUL**. Generates atomic orbital occupation numbers b_i^0 according to the Aufbau principle.

BAND : Perform a band structure calculation.

BONDS : Create the outputfile `filename.ehb` with the bonds generated by keyword DIST for plotting purposes.

CDELTA=δ : δ of the distance-dependent weighted Wolfsberg-Helmholz formula. Use only with keyword DISDEPWH.

CHARGE=n : Overall charge per molecule/unit cell (e.g. NH_4^+ ⇨ CHARGE=1).

CHECKFILE : Check the input file (`filename.ehi`) and then stop. You can also use this option together with the keyword **SYMMETRY** and the symmetry elements defined in card **SYMMOP** to perform a symmetry check of your structure without performing a band calculation.

CHECKGEO : Check the input file down to card **DIST** and then stop. A file `filename.ehc` will be generated which holds the cartesian coordinates of your structure.

CKAPPA=κ : If keyword **DISDEPWH** is chosen, κ is the constant of the distance-dependent weighted Wolfsberg-Helmholz formula, else it serves as constant for the standard or weighted Wolfsberg-Helmholz formula (keywords **UWHIJ, WHIJ**) with $K=1+\kappa$.

COOUT : Select all COs for printing purposes. Printing is released by card **KPOUT**.

COOUT=m TO n : Select CO #m to CO #n for printing purposes. Printing is released by card **KPOUT**.

DISDEPWH : Calculate H_{ij} with distance-dependent weighted Wolfsberg-Helmholz formula . If not specified (keywords CKAPPA, CDELTA), $\kappa=1.0$ and $\delta=0.35$ Å^{-1} will be chosen by default.

DIST : Calculate the interatomic distances lower than 3.5 Å.

DISTN=n : Limit the number of calculated distances to the n smallest ones.

DISTM=x : Limit the largest distance calculated to x Å.

DOUBLET : Determines the occupation of levels: Doublet state required.

EAOORB : Calculate $\Sigma b_i^0 H_{ii}$, H_{ii} denoting the valence orbital ionization energies VOIE of the atoms, b_i^0 the corresponding occupation number of the AOs. Can be used for calculating the orbital stabilization energy ΔE_{EHMO} of neutral systems. Will also be calculated by default, see keyword **REPUL**. If keyword **AUFBAU** is defined, atomic occupation numbers are generated according to the Aufbau principle.

EDITOUT : Used for CEDiT. Generates one EDiT file per **k**-point.

EH : Perform an extended-Hückel MO calculation.

EOUT : Select all eigenvalues (energies) for printing purposes. Printing is released by card **KPOUT**.

EOUT=m TO n : Select eigenvalues from MO (CO) #m through MO (CO) #n for printing purposes. Printing is released by card **KPOUT**.

FERMI=x : Fermi level. This allows a calculation of the total energy for individual **k**-points.

FMO : Perform a fragment analysis.

FMOPOP : Perform a fragment MO (FMO) population analysis (use only with keyword **BAND**). The results are written to `filename.eh4`.

FMOPOP=m : Perform a fragment FMO population analysis from CO #1 to CO #m (use only with keyword **BAND**). The results are written to `filename.eh4`.

FORCE : Used, if non-standard *Slater* orbital definitions are supplied (see card #9, **STO**)

FRACT : Geometry input with fractional coordinates. See card #3, **LATTICE**.

GRAD : Calculate and print energy gradients (use only with keywords **BAND** and **EOUT**). The results are written to `filename.ehe`.

HKOUT : Select all *Bloch* sum matrix elements **H**(**k**) over the coulomb and resonance integrals for printing purposes. Printing is released by card **KPOUT**.

HKOUT=m TO n : Select *Bloch* sum matrix elements **H**(**k**) over the coulomb and resonance integrals from $\mathbf{H}(\mathbf{k})_{mm}$ to $\mathbf{H}(\mathbf{k})_{nn}$ for printing purposes. Printing is released by card **KPOUT**.

HOUT : Print the Huckel matrix H_{ij}.

HOUT=m TO n : Print H_{ij} from i=j=m to i=j=n.

KAPDEL1 : Together with keyword **DISDEPWH**: Select atom-interaction specific (κ,δ).

KAPDEL2 : Together with keyword **DISDEPWH**: Select atom-specific (κ,δ). The interaction between two atoms is given by $\kappa=(\kappa_1+\kappa_2)/2$, $\delta=(\delta_1+\delta_2)/2$. See card **KAPDEL2**.

MOOUT : Select all MOs for printing purposes.

MOOUT=m TO n : Select MO #m through MO #n for printing purposes.

NELEC=n : Number of electrons. If introduced, will be compared to the calculated value.

ORBRADOUT : Prints orbital radii.

OVLPOP : Perform an overlap population analysis for all COs (use only with keyword **BAND**) over all bonds defined by the keywords **DIST**, **DISTM**, **DISTN**, or by card #6, **DIST**. The results are written to `filename.eh3`.

OVLPOP=m TO n : Perform an overlap population analysis from CO #m to CO #n (use only with keyword **BAND**) over all bonds defined by the keywords **DIST**, **DISTM**, **DISTN**, or by card #6, **DIST**. The results are written to `filename.eh3`.

POLAR : Print the polar component of the complex band orbital coefficients C_{ij} (use only with keywords **BAND** and **COOUT**).

QUARTET : Determines the occupation of levels: Quartet state required.

QUINTET : Determines the occupation of levels: Quintet state required.

REPUL : Calculate core-core repulsion energy per molecule/unit cell. If keyword **AUFBAU** is defined the atomic occupation numbers are generated according to the Aufbau principle.

RESTART=n : Restart the calculation after wanted or unwanted interruptions. If multiple geometry variations were calculated, restarts at beginning of a **k**-point set of geometry variation # n. If one geometry variation was calculated, restarts at **k-**point # n. The S and H matrices are recalculated (use only with keyword **BAND**).

RHO=ρ : Distance in Å beyond all overlap integrals are set equal to zero. A default value of 10.0 Å is provided.

SEXTET : Determines the occupation of levels: Sextet state required.

SINGLET : Determines the occupation of levels: Singlet state required.

SKOUT : Select all *Bloch* sum matrix elements **S**(**k**) over the overlap integrals for printing purposes. Printing is released by card **KPOUT**.

SKOUT=m TO n : Select *Bloch* sum matrix elements **S**(**k**) over the overlap integrals from $\mathbf{S}(\mathbf{k})_{mm}$ to $\mathbf{S}(\mathbf{k})_{nn}$ for printing purposes. Printing is released by card **KPOUT.**

SOUT : Print overlap integrals S_{ij}.

SOUT=m TO n : Print S_{ij} from i=j=m to i=j=n.

SYMMETRY : Invokes symmetry analysis. Only needed for band calculations. Strongly recommended for band plotting purposes.

TRIPLET : Determines the occupation of levels: Triplet state required.

UWHIJ : Calculate H_{ij} with normal Wolfsberg-Helmholz formula. If keyword **CKAPPA** is not specified, default K=1.75.

VARIATION : Automated geometry variation. See card #5, **VARY**. Calculates E_{EHMO}, ΔE_{EHMO} and *Fermi* level ε_f for **NELEC** electrons, electrostatic core-core repulsion term E_{Rep}, and total energy E_{tot} for each variation step and per unit cell.

WHIJ : Calculate H_{ij} with weighted Wolfsberg-Helmholz formula (default). If keyword **CKAPPA** is not specified, default K=1.75.

WINDOW=x TO y : Energy window (x to y eV) to limit **AOPOP**, **FMOPOP** and **OVLPOP** calculations (use only with keyword **BAND**. The lower limit x is used only for **OVLPOP** calculations.

XYZ : Geometry input with cartesian coordinates.

ZMAT : Geometry input with Z matrix (GAUSSIAN-style).

Card #3 LATTICE

Number : 1
Requirement : Only together with keyword **FRACT**.
Object : Read lattice constants.
Sequence : a b c alpha beta gamma
Example :

```
LATTICE 5.4
LATTICE 5.3 4.8
LATTICE 7.6 4.32 3.48
LATTICE 8.84 4.71 3.9 120.5 70. 70.
```

Card #4 POS

Number : Limited to NATMMX (see includefile sizes).
Requirement : 1 card minimum.
Object : Read the geometry.
Sequence : name n1 r n2 theta n3 phi (for Z matrix, keyword **ZMAT**).
name x y z (for cartesian and fractional coordinates, keywords **XYZ** or **FRACT**).
Example :

```
POS ?-a
POS ?-b 1 1.0
POS ?-c 2 1.0 1 90.0
POS SI-0101 3 2.69795 2 54.73561 1   45.0
POS O-0101 3 2.64428 2 90.0     1   45.0
POS H-0101 4 1.45000 3 180.
POS Si-0101 0.25 0.25 0.75
```

ZMAT option:

The first atom is at (0,0,0). The second atom defines the z axis. The third atom defines the xz plane. **Caution!: This coordinate input style does not conform to the INPUTC/ICONC (QCMP #116)[41] one, but obeys the conventions of the Gaussian programs.**

name : Complete name of the atom to be added.
r : Distance new atom - atom1 in Å.
theta : Angle new atom - atom1 - atom2 in degree.
phi : Dihedral angle new atom-atom1 - atom2 - atom3 in degree.
n1 : Sequential number of atom1.
n2 : Sequential number of atom2.
n3 : Sequential number of atom3.

XYZ option:

name : Complete name of the atom to be added.
x : x coordinate in Å.
y : y coordinate in Å.
z : z coordinate in Å.

FRACT option:

name : Complete name of the atom to be added.
x : Fractional x coordinate.
y : Fractional y coordinate.
z : Fractional z coordinate.

Atom name format conventions: Sy-mmnn

Sy : Element symbol (e.g. Cu, H, Pb ..). **Dummy atoms** may be used for easy geometry specification (only with keyword **ZMAT**). They should be marked by the symbol '**?**'.
mm : Atom type number that allows differentiation between atoms non related by a symmetry operation (e.g. Cu-01, Cu-02, Cu-03 ...).
nn : Symmetry operation number that allows differentiation between equivalent atoms (e.g. Cu-0101, Cu-0102 ...).

Card #5 **CELL**

Number : Limited to 3. Determines dimensionality of the periodicity of the crystal structure.
Requirement : Only with keyword **BAND**.
Object : Read the cell vectors.
Sequence : name1 TO name2 WITH n NEIGHBORS (**XYZ** or **ZMAT** option).
WITH n NEIGHBORS (**FRACT** option, cell vectors given by card #3, **LATTICE**).
Example : `CELL Si-0101 TO ?-c WITH 4 NEIGHBORS`
`CELL WITH 2 NEIGHBORS`

name1 : Complete name (Sy-mmnn) of the vector-origin atom (e.g. Cu-0101).
name2 : Complete name of the vector-end atom (e.g. C-a)
n : Number of neighbor cells in the lattice sums for **H** and **S**.

Card #6 **VARY**

Number : Unlimited.
Requirement : Only with keyword **VARIATION**.
Object : Determine bonds or angles to be varied.
Sequence : linenumber component increment steps COUPLED
Example :

```
VARY 3 LENGTH 0.02 11
VARY 4 ANGLE 1.0 11 COUPLED
VARY 1 X     -0.5 15
```

linenumber	: Number of **POS**-line of which a component shall be varied (see card #4, **POS**). If lattice constants shall be varied in case of fractional coordinates, supply a 0 (zero).
component	: Coordinate component to be varied. In case of Z-matrix (keyword **ZMAT**): LENGTH, ANGLE, DIHEDRAL In case of cartesian coordinates (keyword **XYZ**): X, Y, Z In case of fractional coordinates (keyword **FRACT**): X, Y, Z, A, B, C, ALPHA, BETA, GAMMA
increment	: Variation increment in [Å] or [°].
steps	: Number of variation steps. Start and end point must be counted, e.g., if a coordinate component shall be varied from 1.1 to 2.1 Å in 0.1 Å steps, supply 11 steps („fence stake principle”).
COUPLED	: Optional. If supplied, motion is coupled to preceding line.

Card #7 **DIST**

Number	: Unlimited.
Requirement	: Only with keyword **OVLPOP** when keywords **DIST**, **DISTN** and **DISTM** are absent.
Object	: Calculate a set of bonds.
Sequence	: name1 TO name2 BETWEEN x AND y
Example	: `DIST Si TO XX BETWEEN 1.5 AND 1.9`

name1	: Generic name of the first atom.
name2	: Generic name of the second atom.
x	: Minimum distance of the search.
y	: Maximum distance of the search.

Format conventions for generic names of atoms:

XX	: All atoms (e.g. Cu-0204, Cu-0103, H-0101 ...).
Sy	: All atoms of the kind Sy (Cu, e.g. Cu-0103, Cu-0201 ...).
Sy-mm	: All atoms of the kind Sy-mm (Cu-02, e.g. Cu-0201, Cu-0202 ...).
Sy-xxnn	: All atoms of the kind Sy-( )nn (Cu-xx02, e.g. Cu-0102, Cu-0202, Cu-0302 ...).
Sy-mmnn	: Atom Sy-mmnn (e.g. Si-0103).

Card #8 **DZETA**

Number	: Unlimited.
Requirement	: None.
Object	: Impose double zeta functions for specified *Slater* type orbital (STO). The default option is single zeta if available from the internal table.
Sequence	: orb OF name
Example	: `DZETA t. OF S`

orb	: Generic name of orbitals. DZETA options apply only for the first five possibilities t., s., : p., d., f. described below.
name	: Generic name of atoms (s. card #7, **DIST**).

Format conventions for generic names of orbitals for a given atom:

t.	: All orbitals.
s.	: All s orbitals.
p.	: All p orbitals.
d.	: All d orbitals.
f.	: All f orbitals.

Names of single orbitals:

s

px
py
pz

dx2-y2
dz2
dxy
dxz
dyz

fz3
fxz2
fyz2
fxyz
fz(x2-y2)
fx(x2-3y2)
fy(3x2-y2)

The orbital sequence is in the order described above.

Card #9 **STO**

Number : Limited to NSYMMX sets (see includefile `sizes`), each set with a maximum of cards.
Requirement : None.
Object : Allows user's Slater type orbitals.
Sequence : main card: symbol orbitaltypes
following cards: energy exp1 coeff1 exp2 coeff2 (for each orbital type 1 card)
Example :

```
STO O 2s. 2p.
 -28.4 2.575 1.0 0.0 0.0
 -12.4 2.275 1.0 0.0 0.0
```

Symbol : Element symbol
Orbitaltypes: : Possibilities: ns.
ns. np.
ns. np. (n-1)d.
ns. np. (n-1)d. (n-2)f., where n is the principal quantum number.

Energy : Valence shell ionization energy H_{ii} [eV].
Exp1 : First Slater orbital exponent.
Coeff1 : First double zeta coefficient (1.0 for single zeta type).
Exp2 : Second Slater orbital exponent (0.0 for single zeta type).
Coeff2 : Second double zeta coefficient (0.0 for single zeta type).

Card #10 **ELCONF**

Number : Limited to NATMMX sets (see includefile `sizes`).
Requirement : None.
Object : Reads the valence electron configuration of the elements. May be required by keywords EAOORB or REPUL, if non-standard element names are supplied. The electron configurations will not be checked.
Sequence : name s p d f
Example :

```
ELCONF Si-0101 2 2 0 0
ELCONF O 2 4 0 0
```

name : Generic name of atom (explanation see card #7, **DIST**).
s : Number of s electrons.
p : Number of p electrons.
d : Number of d electrons.
f : Number of f electrons.

Card #11 **CORECHG**

Number : Limited to NATMMX sets (see includefile `sizes`).
Requirement : None.
Object : Reads the core charge of elements. May be required by keywords **EAOORB** or **REPUL**, if non-standard element names are supplied.
Sequence : name core-charge
Example : `CORECHG Si 4`
`CORECHG C-01 4`

name : Generic name of atom (explanation see card #7, **DIST**).
core-charge : Charge of core. Will not be checked.

Card #12 **KAPDEL1**

Number : Unlimited sets.
Requirement : Only with keyword KAPDEL1.
Object : Sets atom-interaction specific (κ,δ) parameters.
Sequence : KAPDEL1 κ δ BETWEEN name1 AND name2
Example : `KAPDEL1 0.9 0.34 BETWEEN Si-0101 AND Cl-02`

name1 : Generic name of atom 1 (see card #7, **DIST**).
name2 : Generic name of atom 2 (see card #7, **DIST**).

Card #13 **KAPDEL2**

Number : Unlimited sets.
Requirement : Only with keyword KAPDEL2.
Object : Sets atom-specific (κ,δ) parameters.
Sequence : KAPDEL2 κ δ AT name
Example : `KAPDEL2 0.85 0.37 AT Si`

name Generic name of atom (see card #7, **DIST**).

Card #14 **AODELETE**

Number : Unlimited.
Requirement : None, incompatible with card **AOKEEP**.
Object : Delete unwanted orbitals in order to obtain an orbital subset.
Sequence : orb OF name
Example : `AODELETE py OF C`

orb : Generic name of orbitals (see card #7, **DZETA**).
name : Generic name of atoms (see card #6, **DIST**).

Card #15 **AOKEEP**

Number : Unlimited.
Requirement : None, incompatible with card **AODELETE**.
Object : Select an orbital subset.
Sequence : orb OF name
Example : `AOKEEP s. OF C`
`AOKEEP px OF C`
`AOKEEP pz OF C`

orb : Generic name of orbitals (see card #8, **DZETA**).
name : Generic name of atoms (see card #7, **DIST**).

Card #16 **AOOUT**

Number : Unlimited.
Requirement : None.
Object : Select eigenvector output in terms of orbitals. Printing is released by card **KPOUT**.
Sequence : orb OF name
Example : `AOOUT s. OF C-0101`

orb : Generic name of orbitals (see card #8, **DZETA**).
name : Generic name of atoms (see card #7, **DIST**).

Card #17 **SOUT**

Number : Unlimited.
Requirement : None.
Object : Select S_{ij} output in terms of orbitals.
Sequence : orb OF name
Example : `SOUT py OF C-0101`

orb : Generic name of orbitals (see card #8, **DZETA**).
name : Generic name of atoms (see card #7, **DIST**).

Card #18 **HOUT**

Number : Unlimited.
Requirement : None.
Object : Select H_{ij} output in terms of orbitals.
Sequence : orb OF name
Example : `HOUT dxy OF Fe`

orb : Generic name of orbitals (see card #8, **DZETA**).
name : Generic name of atoms (see card #7, **DIST**).

Card #19 **SKOUT**

Number : Unlimited.
Requirement : None.
Object : Select $S_{ij}(\mathbf{k})$ output in terms of orbitals. Printing is released by card **KPOUT**.
Sequence : orb OF name
Example : `SKOUT py OF C-0101`

orb : Generic name of orbitals (see card #8, **DZETA**).
name : Generic name of atoms (see card #7, **DIST**).

Card #20 **HKOUT**

Number : Unlimited.
Requirement : None.
Object : Select $H_{ij}(\mathbf{k})$ output in terms of orbitals. Printing is released by card **KPOUT**.
Sequence : orb OF name
Example : `HKOUT dxy OF Fe`

orb : Generic name of orbitals (see card #8, **DZETA**).
name : Generic name of atoms (see card #7, **DIST**).

The following three cards that define fragments must be together and for each fragment the order **FMOIN**, **FMOOPT**, **FMOOUT** is imperative. A maximum of NFMOMX fragments is possible (see includefile `sizes`). The card names require the fragment number m. All atoms must be assigned to a fragment.

Card **#21 FMOINm**

Number : Unlimited.
Requirement : Only for a fragment analysis.
Object : Define fragment m in terms of orbitals.
Sequence : orb OF name
Example : FMOIN1 s. OF C
FMOIN1 p. OF C
FMOIN1 s. OF H-01

orb : Generic name of orbitals (see card #8, **DZETA**).
name : Generic name of atoms (see card #7, **DIST**).

Card #22 **FMOOPTm**

Number : Unlimited.
Requirement : None.
Object : Select output options for fragment m.
Sequence : option1 option2 …
Example : `FMOOPT1 EOUT FMOUT MOFMO`

Options:

EOUT : Print FMO energies.
EOUT=m TO n : Print FMO energies from FMO m through FMO n.
FMOOUT : Print AO coefficients for all FMOs.
FMOOUT=m TO n : Print AO coefficients from FMO m to FMO n.
AOOUT : Print all AO coefficients for selected FMO.
AOOUT=m TO n : Print AO coefficients from AO m to AO n for selected FMO.
FMOOVLP : Print overlap matrix between FMOs.
FMOINT : Print interaction energy matrix between FMOs.
MOFMO : Print MOs or COs (expressed int terms of FMOs) in the range specified by MOOUT or COOUT.

Card #23 **FMOOUTm**

Number : Unlimited.
Requirement : Only for a fragment analysis.
Object : Select the eigenvector output of fragment m in terms of orbitals.
Sequence : orb OF name
Example :

```
FMOOUT1 s. OF C
FMOOUT1 p. OF C
FMOOUT1 s. OF H-01
```

orb : Generic name of orbitals (see card #8, **DZETA**).
name : Generic name of atoms (see card #7, **DIST**).

Card #24 **SYMMOPn**

Number : Maximum 10 symmetry operations.
Requirement : None. Required together with keyword **SYMMETRY**.
Object : Defines symmetry operations.
Sequence : SYMMOPn operation definition
Example :

```
SYMMOP1 M AT ATOM MO-0101 NORMAL ATOM MO-0102
SYMMOP2 C2 AT 0 0 0 DIR 1 1 1
```

n : Number of the symmetry operation
operation : There a three different symmetry operations available:

- E or 1 Unit operation. Is sometimes needed for automatic labelling of a symmetry line later for the BANDPLOT program.
- M Mirror plane. Definition by:
  - 3 points (coordinates or atoms):
    AT ATOM name1 AT ATOM name2 AT ATOM name3 or
    AT x1 y1 z1 A**T** x2 y2 z2 AT x3 y3 z3 or
    mixtures between atoms and coordinates
  - 1 point and 1 normal vector end point (coordinates or atoms):
    AT ATOM name1 NORMAL ATOM name2
    AT x1 y1 z1 NORMAL x2 y2 z2 or
    mixtures between atom and coordinates or
- Cn n rotation axis. n can be 2,3,4,6. Definition by:
  - 1 startpoint and 1 endpoint (coordinates or atoms):
    AT ATOM name1 DIR ATOM name2 or
    AT x1 y1 z1 DIR x2 y2 z2 or
    mixtures between atoms and coordinates

Card #25 **KPIN**

Number : Unlimited.
Requirement : Only with keyword **BAND**.
Object : Define the **k**-points and the symmetry operations for which the calculation has to be done.
Sequence : k1 k2 k3 weight LINEn1 OPm1 OPm2 … LINEn2 OPm2 …
Example : `KPIN 0.1000 0.0000 0.0000 0 LINE1 OP2`
`KPIN 0 0 0`

k1 : Component of **k** along the first direction.
k2 : Component of **k** along the second direction.
k3 : Component of **k** along the third direction.
weight : Relative weight of the given **k**-point, which is proportional to the volume in 3D- (area in 2D- or length in 1D-) reciprocal space. If weight=0.0, it will be determined by the number of **k**-points.
LINEn : Optional. n designates the number of the symmetry line this **k**-point belongs to. A maximum of three different symmetry lines a **k**-point belongs to can be given.
OPm : Optional. m designates the symmetry element defined above (see card #24, **SYMMOP**) to be applied onto the crystal wavefunctions.

Card #26 **KPOUT**

Number : Unlimited.
Requirement : None
Object : Select output options for the previously given **k**-point and releases printing.
Sequence : option1 option2 …
Example : `KPOUT COOUT AOPOP`

Options:

COOUT	: Print AO coefficients for selected CO.
AOPOP	: Print charge matrix (gross population of AOs).
RAOPOP	: Print reduced charges (gross populations of atoms).
FMOPOP	: Print FMO charge matrix.
RFMOPOP	: Print reduced FMO charges.
COFMO	: Print FMO band orbitals.
OVLPOP	: Print overlap population.
HKOUT	: Print **H**(**k**).
SKOUT	: Print **S**(**k**).

Card #27 **END**

Number	: 1
Requirement	: None.
Object	: Interrupt the flow of input cards.
Sequence	: None.
Example	: `END`

4.1.4 Advices for EHMACC input file

The minimum information required for a band structure calculation is:

Required information	Required card
Band calculation	KEYWRD BAND
Type of coordinates	KEYWRD XYZ, FRACT or ZMAT
E(**k**) vs. **k** gradients	KEYWRD GRAD
Symmetry analysis	KEYWRD SYMMETRY
In case of fractional coordinates, lattice constants	LATTICE lattice constants
Coordinates of atoms, including atoms defining cell vectors	POS name coordinates
Cell information	CELL
Atom parameters	STO
Symmetry operations for symmetry analysis	SYMMOPm
k-points attributed to symmetry line(s) and symmetry operations	KPIN k1 k2 k3 LINEn OPm

The minimum information required for a properties calculation is:

Required information	Required card
Band calculation	KEYWRD BAND
Type of coordinates	KEYWRD XYZ, FRACT or ZMAT
AO-, FMO- or overlap population	KEYWRD AOPOP or FMOPOP or OVLPOP
If overlap population, determine bonds	KEYWRD DIST
In case of fractional coordinates, lattice constants	LATTICE lattice constants
Coordinates of atoms, including atoms defining cell vectors	POS name coordinates
Cell information	CELL
Atom parameters	STO
properties **k**-point set with weights	KPIN k1 k2 k3 weight

The minimum information required for an oscillator strength calculation is:

Required information	Required card
Band calculation	KEYWRD BAND
Generation of Ψ(**k**) files	KEYWRD EDITOUT
Type of coordinates	KEYWRD XYZ, FRACT or ZMAT
In case of fractional coordinates, lattice constants	LATTICE lattice constants
Coordinates of atoms, including atoms defining cell vectors	POS name coordinates
Cell information	CELL
Atom parameters	STO
properties **k**-point set with weights	KPIN k1 k2 k3 weight

4.1.5 Format of the output files

4.1.5.1 `filename.eh1` (UNIT `1`)

Line	Variables	Format
1	NKP,NDIM,NATOM,NELEM,NDEL,NVARY NKP: Number of **k**-points NDIM: Number of COs NATOM: Number of atoms NELEM: Number of element symbols NDEL: Number of deleted AOs NVARY: Number of variations	6I6
2...2+NDIM-1	INDT(I,J), (I=1,NDIM,J=1,2); IND(I), (I=1,NDIM) INDT(I,1): Full atom name INDT(I,2): Orbital name IND: =0: Orbital will be used; =1: Orbital deleted	2A11,I6
2+NDIM...2+NDIM+NELEM-1	SYMBOL(I),NS(I),NP(I),ND(I),NF(I); (I=1,NELEM) SYMBOL: Element names NS: Main quantum numbers for s,p,d,f orbitals NP: ND: NF:	A3,4I2
2+NDIM+NELEM… 2+NDIM+NELEM+NATOM-1	AC(I),NORBEF(I),NORBE(I); (I=1,NATOM) AC: Full atom name NORBEF: Full number of orbitals of an atom NORBE: Number of orbitals of an atom after deletion	A8,2I6
2+NDIM+NELEM+NATOM… 2+NDIM+NELEM+NATOM+ +NKP*(NDIM+1)-1	NKP times: XK(1),XK(2),XK(3),WEIGHT DD(I); (I=1,NDIM) XK: **k**-point components WEIGHT: weight of each **k**-point DD: Energy levels	 4F12.6 F12.6
2+(NKP+1)*NDIM+ NELEM+NATOM+NKP… 2+(NKP+1)*NDIM+ NELEM+NATOM+NKP+1	IORD IORD: Flag for program EHPC, =0 if energies not sorted else NKP*NDIM pointers to sorted energy levels appended (Format I6).	I2

The last two rows of the table are repeated NVARY times.

4.1.5.2 `filename.eh2` (UNIT 2)

Holds AO-coefficients for later calculation of AO projections to the DOS by the program EHPC.

Line	Variables	Format
1.00	IIPOPE IIPOPE: Number of AO projections	I6
2...NDIM*IIPOPE+1	P(I,J); (I=1,NDIM, J=1,IIPOPE) P: AO charge matrix at **k**-point for each crystal orbital I	F8.5

The first and second row of the table are repeated for each **k**-point. For multiple geometry variations all is repeated NVARY times.

4.1.5.3 `filename.eh3` (UNIT 3)

Holds reduced overlap populations defined by cards DIST for later calculation of crystal orbital overlap populations.

Line	Variables	Format
1.00	NEQU NEQU: Number of projections defined by **DIST** card	I4
2	JOVLB, JOVLE JOVLB: Lowest crystal orbital in energy range JOVLE: Highest crystal orbital in energy range	2I4
3...NEQU+2	IDAT(LOQ,1),IDAT(LOQ,2),DDX(LOQ),NDIVIS; LOQ=1,NEQU IDAT(LOQ,1): Name of first atom defining bond. IDAT(LOQ,2): Name of second atom defining bond. DDX(LOQ): Length of bond. NDIVIS: Number of bonds.	2A7,F10.5,I4
NEQU+3 ...NEQU+3+ (IOVLE-IOVLB+2)*NEQU	DD(I,J); I=1,IOVLE-IOVLB+2,J=1,NEQU DD(I,J): Reduced overlap population for crystal orbital I for each projection J	F8.5

For the first **k**-point rows 2-4 of the table are written, for the next **k**-points only the second and the fourth row are written. For multiple geometry variations all is repeated NVARY times.

4.1.5.4 `filename.eh4` (UNIT 4)

Holds FMO coefficients for later calculation of FMO projections of DOS.

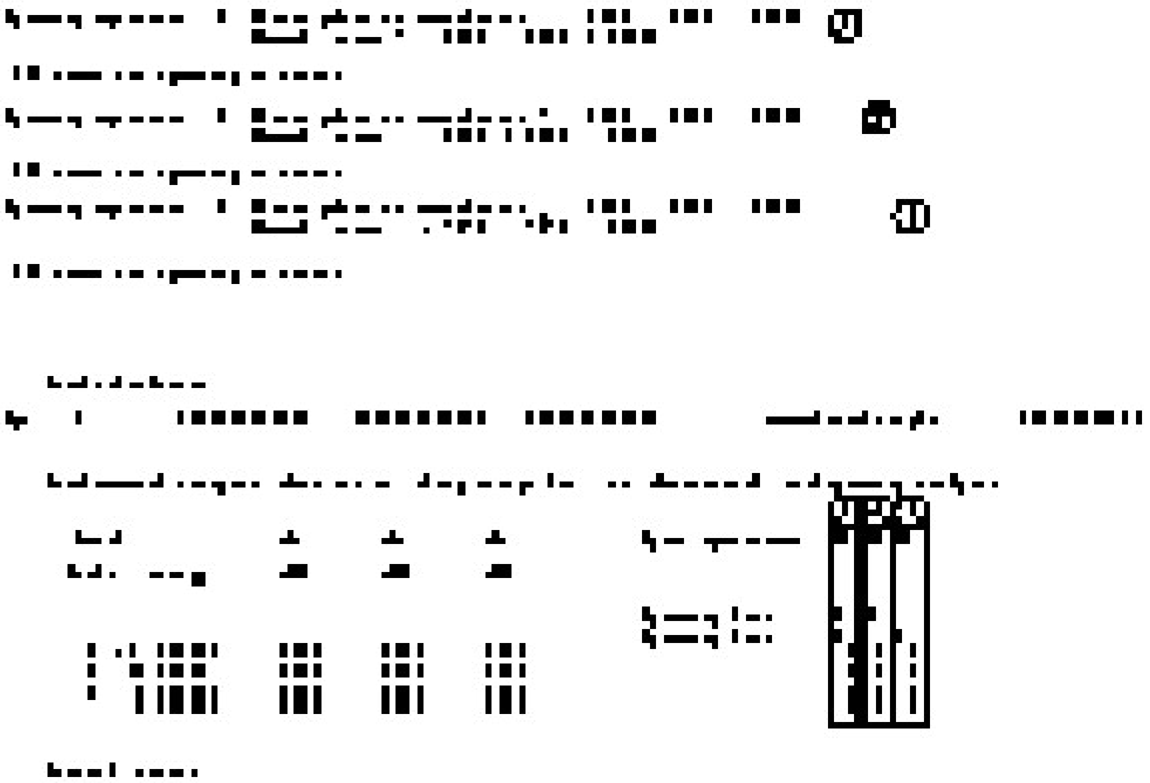
Line	Variables	Format
1	JJPOPE JJPOPE: Number of FMO populations	I6
2...NDIM*JJPOPE+1	F(I,J); (I=1,NDIM, J=1,JJPOPE) F: FMO charge matrix at k-point for each crystal orbital I	F8.5

The first and second row of the table are repeated for each **k**-point. For multiple geometry variations all is repeated NVARY times.

4.1.5.5 `filename.eho` (UNIT 6)

Written free format output.

The output generated by the symmetry analysis of the crystal orbitals (keyword **SYMMETRY**) is organized in colums, see Listing 1. In this example (see `c_gitter.ehi`), the symmetry operation #1, a mirror plane (M) lying in the xy-plane, belongs to both symmetry lines #1 and #3. Symmetry operation #2 only belongs to symmetry line #1. Symmetry operation #3 only belongs to symmetry line #3:



Listing 1: Part of `c_gitter.eho` output.

Besides the energy levels and the gradients the results of the symmetry analysis are written for each crystal orbital (CO). The CO associated with band #1 does not change the sign (**+1**) upon application of symmetry operation #1, whereas CO #2 does (**-1**). **0** means that the AO-coefficients have changed after application of a symmetry operation due to degeneracy.

4.1.5.6 `filename.ehc` (UNIT `8`)

Holds Cartesian coordinates of atoms. Dummy atom coordinates are removed. File is written in SCHAKAL[33] input file format. Each geometry variation `varnumber` is ordered in groups of atoms and can be displayed separately with the `x x #'varnumber` command in SCHAKAL. In the case of fractional coordinates only the first varied geometry is written.

Minimal requirements:

Line	Contents
1	CELL a b c a b g
2..1+NATOM	ATOM name x y z . .
2+NATOM	END

If you do not intend to convert the cartesian coordinates into fractional coordinates by the aid of the program TRANSC (see chapter 4.6.3), remove the BOX statement in the file.

4.1.5.7 `filename.ehb` (UNIT `12`)

Holds bonds defined by the DIST cards. Generated by keyword **BONDS**.

4.1.5.8 `filename.ehe` (UNIT `14`)

Holds energies and energy gradients $\partial E_i/\partial \boldsymbol{k}_j$ (Eqs. (36)-(40)), symmetry label number and results of symmetry analysis for each calculated **k**-point. Generated by keywords **GRAD** and **SYMMETRY**.

Line	Variables	Format
1	XK(I); (I=1,3) XK: Components of **k** vector as multiple of 2π/(unit cell length)	3F10.7
2…4	ILINE(I), NLINOP(I), IPOINT(I,J); (I=1,3, J=1,MXSYME) For three different symmetry lines a **k**-point can be assigned: ILINE: Label of symmetry line NLINOP: Number of symmetry operations for symmetry line ILINE IPOINT: Labels of symmetry operations defined for symmetry line (see keyword **SYMMOP**)	12I3

5...4+LE	D(I),DE1(I),DE2(I),DE3(I),I,ISYMRE(I,J); (I=1,LE, J=1,MXSYME) D: Orbital energy DE1,DE2,DE3: Energy derivative versus component of **k** vector I: Number of orbital, I counts up to number LE of levels to be written ISYMRE: Results of symmetry operation on orbital (1=congruent, -1=antisymmetric, 0=degenerate or not defined by IPOINT)	4F10.5,I4,10I3

Rows 1-3 of the table are repeated for each **k**-point.

4.1.5.9 `filename.ene` (UNIT `15`)

Holds varied coordinates, total electronic energy E_{EHMO}, electronic stabilization energy ΔE_{EHMO}, electrostatic core-core repulsion energy E_{Rep}, total energy $E_{tot} = \Delta E_{EHMO} + E_{Rep}$ per unit cell, and *Fermi* level ε_f. The energies are in [eV]. Generated by keyword **VARIATION**.

4.1.5.10 `###.kpe` (`###` = `001` - number of last k-point)

Holds wavefunctions, energies, and geometry at each **k**-point calculated. Generated by keyword **EDITOUT**.

The `*.kpe` files are continuousely numbered `001.kpe`, `002.kpe`, ... and so forth. A sample `kpe` file as taken from the calculation on all-trans-polyacetylene is given below:

Line	Content
1	KPE
2	0.00000000 0.00000000 0.00000000 0.00561798
3	10
4	1 -.3033697E+02
5	2 -.2041347E+02
6	3 -.1539953E+02
7	4 -.1504642E+02
8	5 -.1329797E+02
9	6 -.3500997E+01
10	7 0.3083166E+01
11	8 0.7782812E+01
12	9 0.2956654E+02
13	10 0.5743839E+02
14	4
15	1 0.0000000E+00 0.0000000E+00 0.0000000E+00
16	2 0.0000000E+00 0.0000000E+00 0.1360000E+01
17	3 0.9353074E+00 0.0000000E+00 -.5400000E+00
18	4 -.9353074E+00 -.9546518E-11 0.1900000E+01
19	EHTB-Rechnung
20	4 10
21	1 1
22	1 4
23	2 2
24	5 8
25	3 3
26	9 9
27	4 4
28	10 10

```
29   -30.3370  1
30   0.4863 -0.0256  0.0000  0.0257  0.4863  0.0256  0.0000 -0.0257  0.0626
      0.0626
31
32   0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
      0.0000
33   -20.4135  2
34   0.2792  0.2117  0.0000 -0.1476 -0.2792  0.2117  0.0000 -0.1476  0.3426
     -0.3426
35
36   0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
      0.0000
37   -15.3995  3
38   0.0868 -0.3385  0.0000  0.2996  0.0868  0.3385  0.0000 -0.2996 -0.4275
     -0.4275
39
40   0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
      0.0000
```

The file does not end at line 40. It is cut there as the remainder part does not contain new relevant information.

The general format is:

Line	Variables	FORMAT
1	A marker called `KPE` to distinguish `*.kpe` from the closely related MOBY files that are used in order to work out oscillator strengths in molecules with the ICON-EDiT program package.[44] MOBY is a molecular modelling package written by *Udo Höweler*.[43] Its input files with the extension `*.mo` were found convenient for ICON-EDiT as well as for the BICON-CEDiT program packages. However, in the latter case a slightly modified form is used to take the needs of band structures into account (e.g., complex wave functions).	A
2	kx ky kz weight Components of **k** vector and respective weight.	4(1X,F11.8)
3	IEE Number of energy bands	I3
4...4+NDIM-1	I,ENERGY(I); I=IEB,IEE Band energies in ascending order from band IEB to band IEE.	I3,1X,E13.7E2
4+NDIM	INATOM Number of atoms in the unit cell.	I3
4+NDIM+ INATOM	I,X(I),Y(I),Z(I); I=1,INATOM Positions of the atoms in the unit cell in Å. The type of the respective atom is determined by means of the corresponding `*.gen` file (see Chapter 4.5.5).	I3, 3(1X, E13.7E2)
4+NDIM+ INATOM+1	not used (a dummy string is written)	A
4+NDIM+ INATOM+2	INATOM,CSORB(INATOM+1)-1 Number of atoms and atomic orbitals per unit cell.	2I3

4+NDIM+ 3*INATOM+2	For each atom I two lines: I,I CSORB(I),CSORB(I+1)-1 CSORB: Number of atomic orbital belonging to atom I	 2I3 2I3
4+NDIM+ 3*INATOM+3	ENERGY(I),I Energy of crystal orbital I	2X, F8.4, I3
4+NDIM+ 3*INATOM+3+IEE/10	CORE(J,I), J=1,IEE COIM(J,I), J=1,IEE Coefficients of the atomic orbitals constituting the crystal orbital. CORE: Real part, COIM: Imaginary part. An empty line is inserted if MOD(IEE,10)=0.	10(1X,F7.4)

The last two table lines are repeated for every crystal orbital.

4.1.6 Required disk space

The disk space in bytes required for the scratch files depends on the dimensionality of the periodic structure and the number of nearest neighbour cells in each dimension (card #4, **CELL**) which are taken for the *Bloch* sums.

n=number of AOs, N=number of neighbour cells in *Bloch* sums, m=number of atoms

d=1, if normal Wolfsberg-Helmholz-formula is used,

d=2, if distance-dependent Wolfsberg-Helmholz-formula is used

1D:	Energy gradients	$8*dNn^2$
	Overlap integrals	$8*(N+1)n^2+4m$
2D:	Energy gradients	$8*2d[2N^2+2N]n^2$
	Overlap integrals	$8*[2N^2+2N+1]n^2+4m$
3D:	Energy gradients	$8*3d[4N^3+6N^2+3N]n^2$
	Overlap integrals	$8*[4N^3+6N^2+3N+1]n^2+4m$

4.1.7 Compiling and linking the program

Some small changes due to different file access in different operating systems have to be made to the sources, see `READ.ME` file.

All relevant parameters for the maximum number of atomic orbitals etc. are described in the include-file `sizes` and can be changed there.

Makefiles are provided for Microsoft Fortran Powerstation 4.0 for Windows NT. They, however, contain the path names of the original distribution. To obtain the correct path names of your installation, remove the files in the FileView window within Microsoft Developer Studio, and reinsert them again by using the menu entry `Insert/Files into Project...`.

4.2 EHPC

4.2.1 Description

On the basis of the files produced by EHMACC (i.e. `filename.eh1`, `filename.eh2`, `filename.eh3`, `filename.eh4`), this program calculates the *Fermi* level, the density of states DOS, the integrated density of states, the projected density of states, the crystal orbital overlap population COOP and the total electronic energy of a crystalline solid as a function of band filling or energy.

The results (`filename.eh5`) are used by the program DOSPLOT for plotting.

Originally written by T. Hughbanks, M. Kertesz, S.Wijeysekera, C. Wilker, and C. Zheng in the laboratory of R. Hoffmann, Dept. of Chemistry, Cornell University, Ithaca, N.Y.

Written as a separate package by M.-H. Whangbo, Dept. of Chemistry, North Carolina State University, Raleigh N.C., 1981.

Modified by M. Evain in the laboratory of M.-H. Whangbo, Dept. of Chemistry, North Carolina State University, Raleigh N.C., 1987.

Gaussian smoothed integrals of AO and FMO contributions to the DOS and of COOP curves, and selection of geometry variations added by Martin Brändle in the laboratory of Gion Calzaferri, Department of Chemistry and Biochemistry, University of Berne, Freiestr. 3, CH-3012 Berne, Switzerland.

4.2.2 Invoking the program, input and output files

All input and output files will be generated in ASCII format. Filenames are free, but fixed extensions have to be used.

Input:	`filename.epi`	• Holds commands and parameters for properties calculation.
	`filename.eh1`	• Holds information about number of atoms, energies etc.
	`filename.eh2`	• Holds information about AO population.
	`filename.eh3`	• Holds information about COOP.
	`filename.eh4`	• Holds information about FMO population.
Output:	`filename.epo`	• Collects all results of a properties calculation.
	`filename.eh5`	• Holds results in format easily readable for DOSPLOT.
Scratch:	`ehpc.s08`	• Holds temporary results for geometry variation.

Windows NT version:

1. Open a DOS command prompt window.
2. Type `ehpc.bat filename`

   No extension is needed: An input file with the extension `.epi` is assumed.

4.2.3 Format of the input file (`filename.epi`, UNIT 5)

All input to the program is unformatted. Spaces may be set anywhere except, of course, within an independent item (card name, a number or a chemical symbol). Spaces must be set to separate independent items. A physical line of input may not exceed a total of 80 characters. The program understands capital letters as well as lowercase letters and does not distinguish between both of them.

The input file `filename.epi` is divided into sections („cards") .The sequence of the cards described below should be maintained. No input program is provided, instead the user can use his favourite editor for file creation. Blank lines may be inserted anywhere for user's convenience. Any line containing an exclamation mark **!** anywhere is considered a comment.

Card #1 **TITLE**

Number	:	1
Requirement	:	None.
Object	:	Title for user's convenience.
Sequence	:	None.
Example	:	`TITLE Polyacetylene`

Card #2 **KEYWRD**

Number	:	Unlimited.
Requirement	:	None.
Object	:	Control the input sequence, the calculation path and the output options.
Sequence	:	keyword1 keyword2 keyword3

Example	:	`KEYWRD IDOS SAOPOP NMESH=800`
		`KEYWRD SDOS`
		`KEYWRD EMAX=10.0`

List of keywords:

AVERAGE	:	Calculates an average overlap population per bond from a set of equivalent bonds indicated by card **OVLPROJm** (default: calculates an accumulated overlap population for a set of equivalent bonds).
CHECKFILE	:	Checks the input and stops then.
DELTA=d	:	Gaussian half-width as described in chapter 3.5 (default DELTA=0.08).
EMAX=x	:	Upper limit x of the energy window for gaussian smoothing.
EMIN=y	:	Lower limit y of the energy window for gaussian smoothing.
IAOPOP	:	Integrated AO population, in the range of electron numbers defined by the keyword **INTEGR**, for the projections selected in the cards **AOPROJm**. Requires keyword **AOPOP** in EHMACC input file.
IDOS	:	Integrated density of states in the range of electron numbers defined by the keyword **INTEGR**.
IFMOPOP	:	Integrated FMO population, in the range of electron numbers defined by the keyword **INTEGR**, for the projections selected in the cards **FMOPROJm**. Requires keyword **FMOPOP** in EHMACC input file.
INTEGR=m TO n	:	The range, m to n, of electron numbers for which to carry out integration without gaussian smoothing.

IOVLPOP	:	Integrated overlap population, in the range of electron numbers defined by the keyword **INTEGR**, for the projections selected in the cards **OVLPROJm**. Requires keyword **OVLPOP** in EHMACC input file.
LIMIT=x	:	The limit x for the gaussian smoothing below which the value of the gaussian is considered to be zero (default LIMIT=1.0E-4). See chapter 3.5.
NMESH=n	:	Number of mesh points between **EMAX** and **EMIN** for gaussian smoothing (maximum: MXMESH, see includefile `sizes`).
SAOPOP	:	Gaussian smoothed AO population and integrated AO population, in the range of the energy window between **EMAX** and **EMIN**, for the projections selected in the cards **AOPROJm.** Used for graphical representation with DOSPLOT. Requires keyword **AOPOP** in EHMACC input file.
SDOS	:	Gaussian smoothed density of states in the range of the energy window between **EMAX** and **EMIN**. Used for graphical representation with DOSPLOT.
SFMOPOP	:	Gaussian smoothed FMO population and integrated FMO population, in the range of the energy window between **EMAX** and **EMIN**, for the projections selected in the cards **FMOPROJm**. Used for graphical representation with DOSPLOT. Requires keyword **FMOPOP** in EHMACC input file.
SOVLPOP	:	Gaussian smoothed overlap population (COOP) and integrated overlap population, in the range of the energy window between **EMAX** and **EMIN**, for the projections selected in the cards **OVLPROJm**. Used for graphical representation with DOSPLOT. Requires keyword **OVLPOP** in EHMACC input file.
STEP=x	:	Fractional number x of electrons to be added in each integration step (see keyword **INTEGR**, minimum is 0.1 electrons).
VARIATION=m	:	Selects geometry variation number m. Both integrated and smoothed properties will be calculated and a full output generated. Requires keyword **VARIATION** in EHMACC input file.
VARIATION=m TO n	:	Selects geometry variation from m through n. Only integrated properties (keywords **IDOS**,**IAOPOP**,**IFMOPOP** and **IOVLPOP**) will be calculated and a brief output is written. Requires keyword **VARIATION** in EHMACC input file.

Projection cards:

The maximum number m of each projection is limited to MXCARD (default=5, see includefile `sizes`).

The projections have to be entered in sequence (e.g. **AOPROJ1, AOPROJ2, AOPROJ3** ...)

Many consecutive cards may be used to define each projection m.

Card #3 **AOPROJm**

Number	:	Unlimited.
Requirement	:	Only for keywords **IAOPOP** or **SAOPOP**.
Object	:	Define the AOs included in the projection m (see chapter 4.1.3 , cards #4, **POS** and #8, **DZETA**).
Sequence	:	orb 1 OF name1, orb2 of name2
Example	:	`AOPROJ1 px OF S-0101` `AOPROJ1 py OF S-0101`

Card #4 **FMOPROJm**

Number : Unlimited.
Requirement : Only for keywords **IFMOPOP** or **SFMOPOP**.
Object : Define the FMOs included in the projection m.
Sequence : i1-i2 k1 …
Example : `FMOPROJ1 1-5 7`
`FMOPROJ2 3`
`FMOPROJ2 5-8`

i1-i2 : All FMOs i1 through i2
k1 : Individual FMO k1

Card #5 **OVLPROJm**

Number : Unlimited.
Requirement : Only for **IOVLPOP** or **SOVLPOP**.
Object : Define the bonds included in the projection m (see chapter 4.1.3, card #7, **DIST**, and EHMACC output `filename.eho`).
Sequence : name1 TO name2 BETWEEN x AND y
Example : `OVLPROJ1 C-0101 TO C-0102 BETWEEN 1.30 AND 1.36`

Card #6 **END**

Number : 1
Requirement : None.
Object : Interrupt the flow of input cards.
Sequence : None.
Example : `END`

4.2.4 Format of the output files

4.2.4.1 `filename.epo` (UNIT 6)

Holds commented results. Free format ASCII output is produced.

4.2.4.2 `filename.eh5` (UNIT 7)

Line	Variables	Format
1	J1,NPROJ,NPROJF,NPROJB,NMESH J1: 3=keyword **SDOS** selected, 0=keyword **SDOS** not selected NPROJ: Number of AO projections NPROJF: Number of FMO projections NPROJB: Number of COOP projections NMESH: Number of energy mesh points	(5I4)
2	NCARD(I); I=1,5 NCARD: Number of description strings for AO projections	(5I4)
3	NCARDF(I); I=1,5 NCARDF: Number of description strings for FMO projections	(5I4)

4	NCARDB(I); I=1,5 NCARDB: Number of description strings for COOP projections	(5I4)
5...104	TITL(I,J); (J=1,20, I=1,5) TITL: Description strings for AO projections	(A80)
105...204	TITLF(I,J); (J=1,20, I=1,5) TITLF: Description strings for FMO projections	(A80)
205...304	TITLB(I,J); (J=1,20, I=1,5) TITLB: Description strings for COOP projections	(A80)
305 ...305+NMESH-1	EDX(I); I=1,NMESH EDX: Energy mesh with NMESH energy points	(F12.5)
305+NMESH ...305+2*NMESH-1	XND(I); I=1,NMESH XND: Smoothed density of states	(F12.5)
305+2*NMESH ...305+3*NMESH-1	XNDI(I); I=1,NMESH XNDI: Smoothed integrated density of states	(F12.5)
305+3*NMESH ...305+4*NMESH-1	TEDI(I); I=1,NMESH TEDI: Smoothed total energy	(F12.5)
305+4*NMESH ...305+(4+2*NCARD)*NMESH-1	XND(I); I=1,NMESH XNDI(I); I=1,NMESH XND: Smoothed AO population XNDI: Smoothed integrated AO population only with keyword **SAOPOP**, for each projection card	(F12.5)
305+(4+2*NCARD)*NMESH ...305+(4+2*(NCARD+NCARDF))*NMESH-1	XND(I); I=1,NMESH XNDI(I); I=1,NMESH XND: Smoothed FMO population XNDI: Smoothed integrated FMO population only with keyword **SFMOPOP**, for each projection card	(F12.5)
305+(4+2*NCARD+NCARDF)*NMESH ...305+(4+2*(NCARD+NCARDF+NCARDB))*NMESH-1	XND(I); I=1,NMESH XNDI(I); I=1,NMESH XND: Smoothed COOP XNDI: Smoothed integrated COOP only with keyword **SOVLPOP**, for each projection card	(F12.5)

Table rows 9-11 are only written if keyword **SDOS** was selected.

4.2.5 Compiling and linking the program

Some small changes due to different file access in different operating systems have to be made to the sources, see `READ.ME` file.

All relevant parameters for the maximum number of atomic orbitals etc. are described in the include-file `sizes` and can be changed there.

Makefiles are provided for Microsoft Fortran Powerstation 4.0 for Windows NT. They, however, contain the path names of the original distribution. To obtain the correct path names of your installation, remove the files in the FileView window within Microsoft Developer Studio, and reinsert them again by using the menu entry `Insert/Files into Project...`.

4.3 DOSPLOT

4.3.1 Description

DOSPLOT is used for plotting density of states (DOS), integrated DOS, AO projections of the DOS, the total energy, and crystal orbital overlap populations (COOP).
DOSPLOT uses the PGPLOT graphics subroutine library by T. J. Pearson, California Institute of Technology, `http://astro.caltech.edu/~tjp/pgplot/` . This library supports many operating systems and a wealth of graphic drivers.[40]

Written by Martin Brändle in the laboratory of G. Calzaferri, Department of Chemistry and Biochemistry, University of Berne, Freiestr. 3, CH-3012 Berne, Switzerland.

4.3.2 Input and output files

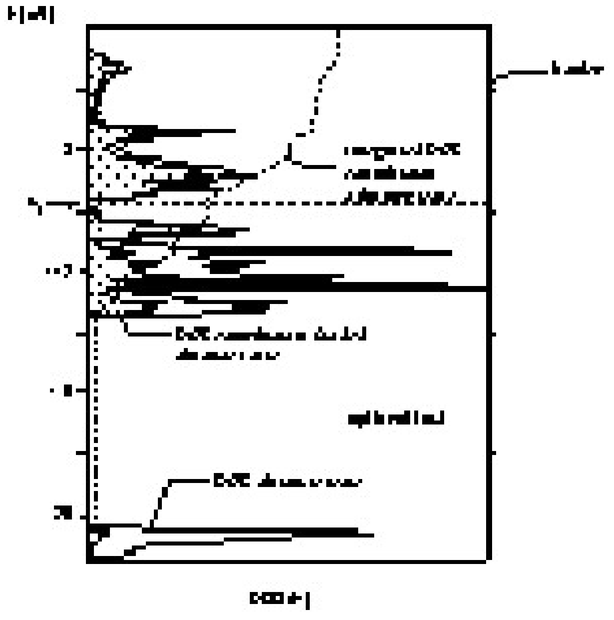
Filenames are free, but fixed extensions have to be used.

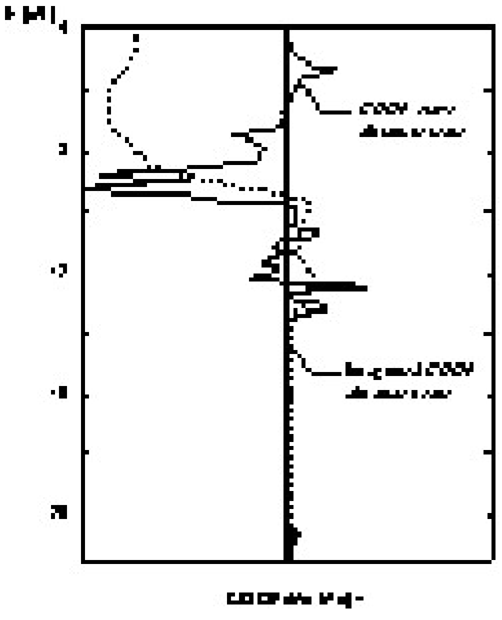
Input:	`filename.eh5`	- Required. Holds results in format easily readable for DOSPLOT.
	`filename.dpl`	- If existing, holds graph layout data.
Output:	`dosplot.eps`	- Postscript or encapsulated postscript output, color and gray scale.
	`dosplot.plt`	- HPGL output.
	`filename.dpl`	- Graph layout data.

4.3.3 Program philosophy

Command input and graphical output are separated in two different windows: Input is done in the terminal window where the program has been called, the graphs are output in a separate window.

There are three different types of plots: DOS, COOP and combined DOS/COOP, see below:





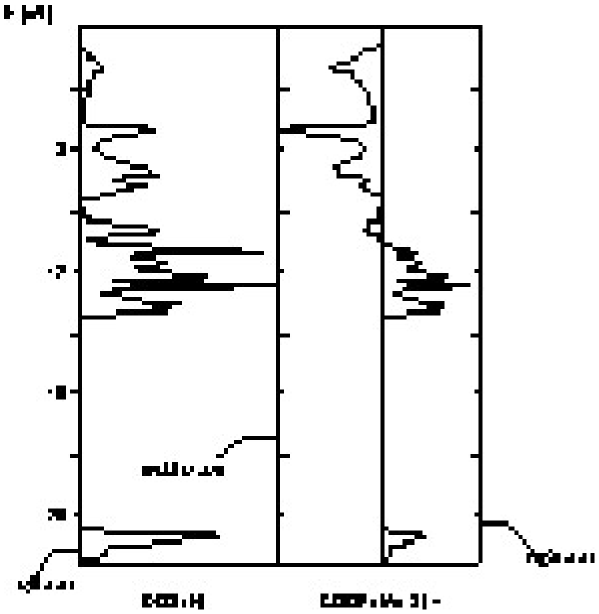


Figure 10: Different types of DOS/COOP plots.

Each property calculated by EHPC (keywords **SDOS**, **SAOPOP**, **SFMOPOP**, and **SOVLPOP**) can be read into curve buffers, which are generated sequentially. Curve buffers exist for DOS and COOP plots. The property of a curve (fill shade and line type) is set by curve buffer number. The number of curve buffers can be set in the includefile `sizes`.

4.3.4 Usage

Windows NT version:

1. Open a DOS command prompt window.
2. Type `dosplot.bat inputfile`

   No extension is needed: An input file with the extension `.eh5` is assumed.
3. A new window `dosplot` is created. Maximize the window. Choose menu `Window`, entry `Tile`. Your startup screen should look like this:

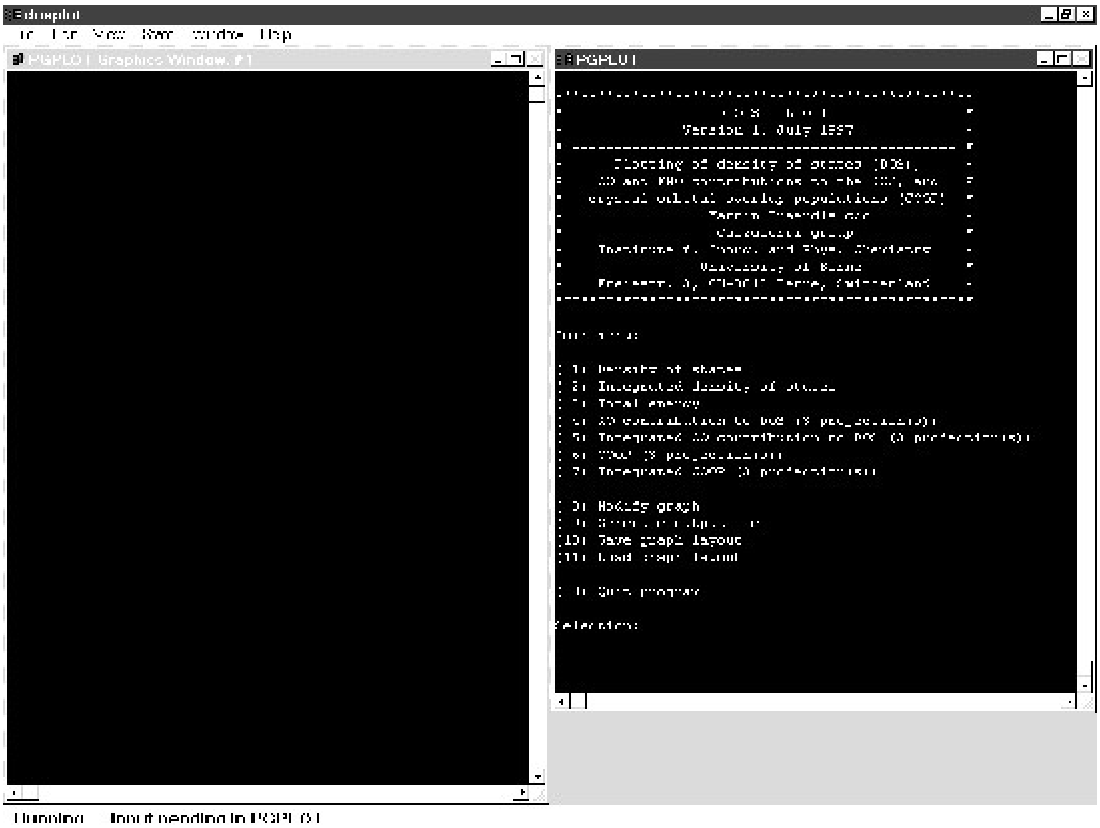


Figure 11: Startup screen of DOSPLOT (Windows NT version).

All functions of DOSPLOT are available through menus. You enter the main menu for display of calculated properties and for the printing of your final graphical representation:

```
Main menu:

( 1) Density of states
( 2) Integrated density of states
( 3) Total energy
( 4) AO contribution to DOS (5 projection(s))
( 5) Integrated AO contribution to DOS (5 projection(s))
( 6) FMO contribution to DOS (1 projection(s))
( 7) FMO contribution to DOS (1 projection(s))
( 8) COOP (2 projection(s))
( 9) Integrated COOP (2 projection(s))

(10) Modify graph
(11) Delete curve
(12) Generate outputfile
(13) Save graph layout
(14) Load graph layout

( 0) Quit program

Selection:
```

The number of menu entries depends on the type of properties calculated with EHPC. After selection of a calculated property into a curve buffer (in this case menu entry `1-9`), you may be asked some self-explaining questions. Integrated DOS contributions are plotted in a relative scale with respect to a maximum you will be asked for. Normally this maximum corresponds to the number of electrons of the corresponding AO or FMO contribution when all bands are filled.
By choosing `Modify graph` you reach the

```
Graph modification menu:

( 1) Change energy range
( 2) Change x range
( 3) Add border
( 4) Add axes tic marks and tic labels
( 5) Add axes titles
( 6) Add optional text
( 7) Close or open a curve (toggle)
( 8) Change linetype of a curve
( 9) Fill area below a DOS curve
(10) Mark fermi energy
(11) Change buffer ranges
(12) Change DOS buffer drawing offsets
(13) List curve buffer contents

( 0) Return to main menu
```

Always enter floating point numbers (energies, buffer and x-range) with a decimal point.

4.3.4.1 Meaning of x range and buffer range

x range determines the width of curves in the DOS or COOP window between the borders. It is measured in relative units: default x range = 1.0 (100%). Changing of x range affects all curves displayed. Setting x range = 2.0 reduces the amplitudes of the curves to half of their size, x range = 0.5 doubles their amplitudes. If you do not want that the peaks of the curves coincide with the right border, set x range = 1.01.
The curves in absolute units are plotted with respect to the maximum value of all curve buffers for a corresponding type, e.g. DOS or COOP. Buffer range affects only these curves. By default it is set to the absolute minimum and absolute maximum of all buffers, but can be changed with option (11).

4.3.4.2 Entering text

Axis titles have fixed positions.
Optional text is entered and positioned with the aid of the mouse. By entering a blank line you return to the graph modification menu.

Windows NT version: Enter text, position and text angle, then move mouse cursor to graphics window. Click left mouse button. The cross follows your mouse pointer. Position the cross. Press any keyboard key.

Characters are displayed as Hershey characters (vector graphics). For many applications their quality is sufficient. If you need better quality, produce an output file and change the characters in a drawing program.

Special characters, superscript and subscript modes and some graphic signs can be set by inline-commands, which have to be preceded by a backslash `\` and followed by a space. The syntax of the commands comes close to the one used in Lotus Manuscript© or T$_E$X equation mode. It's also possible to use the commands and symobls of the PGPLOT graphics subroutine library, see `http://astro.caltech.edu/~tjp/pgplot/`.

Table 6: Greek letters

\ALPHA Α \BETA Β \CHI Χ \DELTA Δ \EPSILON Ε \PHI Φ \GAMMA Γ \ETA Η \IOTA Ι \KAPPA Κ \LAMBDA Λ \MU Μ	\NU Ν \OMIKRON Ο \PI Π \THETA Θ \SIGMA Σ \TAU Τ \UPSILON Υ \OMEGA Ω \XI Ξ \PSI Ψ \ZETA Ζ	\alpha α \beta β \chi χ \delta δ \epsilon ε \phi φ \gamma γ \eta η \iota ι \kappa κ \lambda λ \mu μ	\nu ν \omikron ο \pi π \theta θ \rho ρ \sigma σ \tau τ \upsilon υ \omega ω \xi ξ \psi ψ \zeta ζ

Table 7: Commands and special characters:

Command	Usage	Example	Printed result
\piovera	Draws symbol for right edge of one-dimensional *Brillouin* zone	`\piovera`	$\frac{\pi}{a}$
\space	Draws a space	`a \super \space 2 \endsuper`	a^2
\\	Prints backslash	`\\`	\
\infinity		`\infinity`	∞
\angstrom		`\angstrom`	Å
\+-		`a \+- \space b`	

\super	Superscripts text	`a \super 2 \super 2 \endsuper \endsuper`	$a^{2^{2}}$
\endsuper	End of superscript	`a \super 2 \super 3 \endsuper *4 \endsuper`	$a^{2^{3}*4}$
\sub	Subscripts text	`a \sub 2 sub 3 \endsub \endsub`	$a_{2_{3}}$
\endsub	End of subscript	`C \sub x \sub 3 \endsub \endsub H \sub x \sub 3 \endsub -2 \endsub`	$C_{x_{3}}H_{x_{3}-2}$
\back	Sets cursor back one character	a \back ‘	$\acute{a}$
\arrowdown		`a \arrowdown b`	$a \downarrow b$
\arrowup		`a \arrowup b`	$a \uparrow b$
\arrowlft		`a \arrowlft b`	$a \leftarrow b$
\arrowrgt		`a \arrowrgt b`	$a \rightarrow b$

4.3.4.3 Line types

The line types are selected by numbers:
1 = line, 2 = dashed, 3 = dash-dotted, 4 = dotted, 5 = dash-dot-dot-dotted

4.3.4.4 Fill textures

There are 16 different fill shades selected by numbers. Fill shades are only active when the curve in a buffer has been closed. On screen, fill shades are displayed as colors. They are converted to gray scale depending on output file selection.

Table 8: Fill shades

0	no shading
1	black
2	red
3	green
4	blue
5	cyan
6	magenta
7	yellow
8	orange
9	light green
10	pale green
11	light blue
12	blue
13	purple
14	dark grey
15	light grey

4.3.5 Printing

Windows NT version:

Two possibilities exist for printing:

High quality vector output: Select `Generate output file` in the main menu.
For quick printing or saving of a bitmap: Menu at the top left, „`File/Print ...`” or „`File/Save ...`”. Your graphics card must be set to 256 colors.

4.3.6 Format of graph layout file `filename.dpl` (UNIT 7)

Line	Variables	Format
1	IBUFDS IBUFDS: Number of used DOS curve buffers.	I2
2..1+IBUFDS	IBFNDX(I),ILTYPE(I),ISHADE(I),FARBTR(I),FCLOSE(I), DOFFST(I),BUFMIN(I),BUFMAX(I),BUFTIT(I), I=1,IBUFDS IBFNDX: Curve retrieving index from `filename.eh5`. ILTYPE: Line type of curve. ISHADE: Fill shade index. FARBTR: If .TRUE., curve in arbitrary units. FCLOSE: If .TRUE., curve is closed and can be shaded. DOFFST: DOS curve offset. BUFMIN: Minimal value of curve buffer. BUFMAX: Maximal value of curve buffer. BUFTIT: Descriptor string of buffer.	3I2,2L1,3F12.5,A72
2+IBUFDS	IBUFCP IBUFCP: Number of used COOP curve buffers.	I2
3+IBUFDS … 2+IBUFDS+IBUFCP	IBFNDX(I),ILTYPE(I),ISHADE(I),FARBTR(I),FCLOSE(I),BUFMIN(I),BUFMAX(I),BUFTIT(I), I=MAXBUF+1,MAXBUF+IBUFCP MAXBUF: Maximum number of curve buffers (see file `sizes`) Same meaning of variables, but for COOP curve buffers.	3I2,2L1,2F12.5,A72
3+IBUFDS+IBUFCP	FSELEC(I), I=1,MAXSEL FSELEC: Graph options, will be perfomed if true (T) FSELEC(3): Border FSELEC(4): Axes tic marks FSELEC(5): Axes titles FSELEC(6): Free text FSELEC(10): Fermi level	20L1

4+IBUFDS+IBUFCP	VPXDOO,VPXDOW,VPXCOO,VPXCOW,VPXOFF, VPXAXW,VPXFUL,VPYOFF,VPYAXH,VPYFUL Viewport coordinates: VPXDOO: Lower left corner of DOS graph, x coord. VPXDOW: Width of DOS graph. VPXCOO: Lower left corner of COOP graph, x coord. VPXCOW: Width of COOP graph. VPXOFF: Default lower left corner, x coordinate. VPXAXW: Width of x axis. VPXFUL: Full viewport width. VPYOFF: Default lower left corner, y coordinate. VPYAXH: Height of y axis. VPYFUL: Full viewport height.	10F10.5
5+IBUFDS+IBUFCP	EMIN,EMAX EMIN: Lower boundary of energy range. EMAX: Upper boundary of energy range.	2F10.5
6+IBUFDS+IBUFCP	ETCMIN,ETCMAX,ETCDIS,ETCDS2 ETCMIN: Lower boundary of energy axis tics. ETCMAX: Upper boundary of energy axis tics. ETCDIS: Distance between unlabelled energy axis tics. ETCDS2: Distance between labelled energy axis tics.	4F10.5
7+IBUFDS+IBUFCP	DTCMIN,DTCMAX,DTCDIS,DTCDS2 DTCMIN: Lower boundary of DOS axis tics. DTCMAX: Upper boundary of DOS axis tics. DTCDIS: Distance between unlabelled DOS axis tics. DTCDS2: Distance between labelled DOS axis tics.	4F10.5
8+IBUFDS+IBUFCP	CTCMIN,CTCMAX,CTCDIS,CTCDS2 CTCMIN: Lower boundary of COOP axis tics. CTCMAX: Upper boundary of COOP axis tics. CTCDIS: Distance between unlabelled COOP axis tics. CTCDS2: Distance between labelled COOP axis tics.	4F10.5
9+IBUFDS+IBUFCP	XDOSMN,XDOSMX,XCOPMX,XRANGE, DOSMN,DOSMX,COOPMX XDOSMN: Minimum of visible DOS axis range. XDOSMX: Maximum of visible DOS axis range. XCOPMX: Maximum of visible COOP axis range. XRANGE: x range. DOSMN: Minimum of all DOS curve buffers. DOSMX: Maximum of all DOS curve buffers. COOPMX: Maximum of all COOP curve buffers.	7F10.5
10+IBUFDS+IBUFCP	EFERMI,FFLINE EFERMI: *Fermi* level. FFLINE: Draw line at *Fermi* level if true (T).	F10.5,L1

11+IBUFDS +IBUFCP	ITSDLF,ITSDMD,ITSDRT,ITSDUP,ITSDLO, ITSCUP,ITSCLO Axis sides of tic marks (0: none, 1: left or lower, 2: right or upper, 3:both) ITSDLF: Left energy axis. ITSDMD: Middle energy axis. ITSDRT: Right energy axis. ITSDUP: Upper DOS axis. ITSDLO: Lower DOS axis. ITSCUP: Upper COOP axis. ITSCLO: Lower COOP axis.	7I1
12+IBUFDS+ IBUFCP	CDOSTI CDOSTI: DOS axis title.	A80
13+IBUFDS+ IBUFCP	CCOPTI CCOPTI: COOP axis title.	A80
13+IBUFDS+ IBUFCP	CENETI CENETI: Energy axis title.	A80
13+IBUFDS+ IBUFCP	ITXT ITXT: Number of free text strings.	I4
14+IBUFDS+ IBUFCP … 13+IBUFDS+ IBUFCP+ITXT	TEXTXY(I,1),TEXTXY(I,2),TXTANG(I),ITXPOS(I), CTEXT(I),I=1,ITXT TEXTXY: Coordinates of free text strings. TXTANG: Text angle. ITXPOS: Text position. CTEXT: Free text strings.	3F10.5,I2,A80

4.3.7 Compiling and linking

Makefiles are provided for Microsoft Fortran Powerstation 4.0 for Windows NT. They, however, contain the path names of the original distribution. To obtain the correct path names of your installation, remove the files in the FileView window within Microsoft Developer Studio, and reinsert them again by using the menu entry `Insert/Files into Project...`.

The sources must be linked with the PGPLOT graphics subroutine library. The PGPLOT library is available at `http://astro.caltech.edu/~tjp/pgplot/`.

Porting to different operating systems or different graphic libraries (e.g. NAG[41] and GKS[42]) is possible. We do, however, not give any support for this.

Some small changes due to different file access in different operating systems have to be made to the sources, see `READ.ME` file.

The file `sizes` contains and describes all relevant parameters that change the maximum number of atomic orbitals, the number of curve buffers etc. Also the default values for the size of the plots can be changed. It is, however, recommended that these default parameters are not changed, since the ratio of the axes, $width/height = 1/\sqrt{2}$, conforms to european DIN paper size settings (A4). The plot dimensions should be compatible to the graphs generated by BANDPLOT, see includefile `sizes`.

4.4 BANDPLOT

4.4.1 Description

BANDPLOT is used for plotting and interpolating bands between **k**-points along symmetry lines calculated with EHMACC.

BANDPLOT uses the PGPLOT graphics subroutine library by T. J. Pearson, California Institute of Technology, `http://astro.caltech.edu/~tjp/pgplot/` . This library supports many operating systems and a wealth of graphic drivers.[40]

Graphic part and ordering of energy levels according to symmetry analysis written by Martin Brändle in the laboratory of G. Calzaferri, Department of Chemistry and Biochemistry, University of Berne, Freiestr. 3, CH-3012 Berne, Switzerland.

The band interpolation subroutines were incorporated and adapted by Martin Brändle from the BFIT program written by S. Wijeysekera in the laboratory of R. Hoffmann, Dept. of Chemistry, Cornell University, Ithaca, N.Y.

4.4.2 Input and output files

Filenames are free, but fixed extensions have to be used:

Input:	`filename.eh1`	- Required. Holds information about number of atoms, energies etc. Generated by EHMACC.
	`filename.ehe`	- Required. Holds results in format easily readable for BANDPLOT. Generated by EHMACC.
	`filename.bin`	- If existing, holds sorting information for bands.
	`filename.bft`	- If existing, holds fitted energy bands.
	`filename.bpl`	- If existing, holds graph layout data.
Output:	`filename.bin`	- Holds sorting information for bands.
	`filename.bft`	- Holds fitted energy bands.
	`filename.bpl`	- Holds graph layout data.
	`bandplot.eps`	- Postscript or encapsulated postscript output.
	`bandplot.plt`	- HPGL output.

4.4.3 Startup

Windows NT version:

1. Open a DOS command prompt window.
2. Type `bandplot.bat inputfile`

   No extension is needed: Input files with the extensions `.eh1` and `.ehe` are assumed.
3. A new window „bandplot” is created. Maximize the window. Choose menu `Window`, entry `Tile`. The screen should look like this

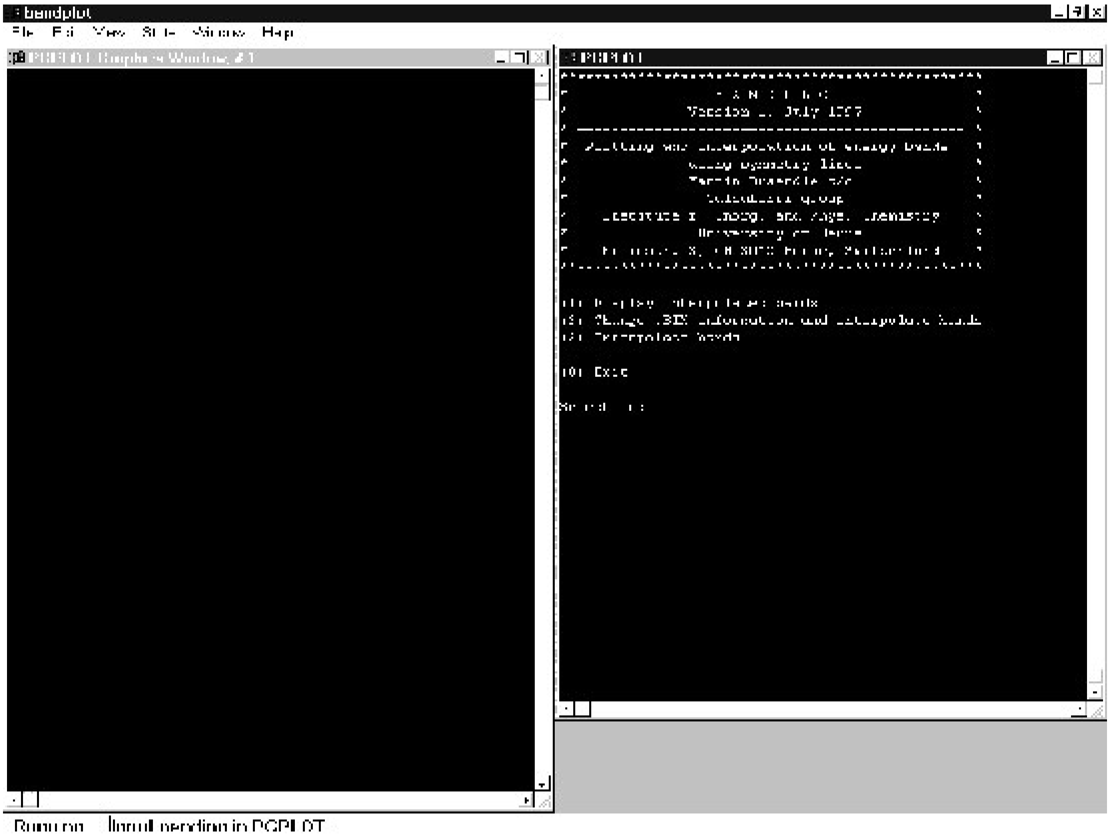


Figure 12: Startup screen of BANDPLOT (Windows NT version).

4. At left you see a graphics window, at right a terminal window.

4.4.4 Program concepts and usage

Command input and graphical output are separated in two different windows: Input is done in the terminal window where the program has been called, the graphs are output in a separate window.

For final graphical representation and print-out of a band structure BANDPLOT needs interpolated band data and band sorting information stored in `filename.bft` and in `filename.bin`, respectively. You have to generate this information first. The procedure is sketched in the flowchart below:

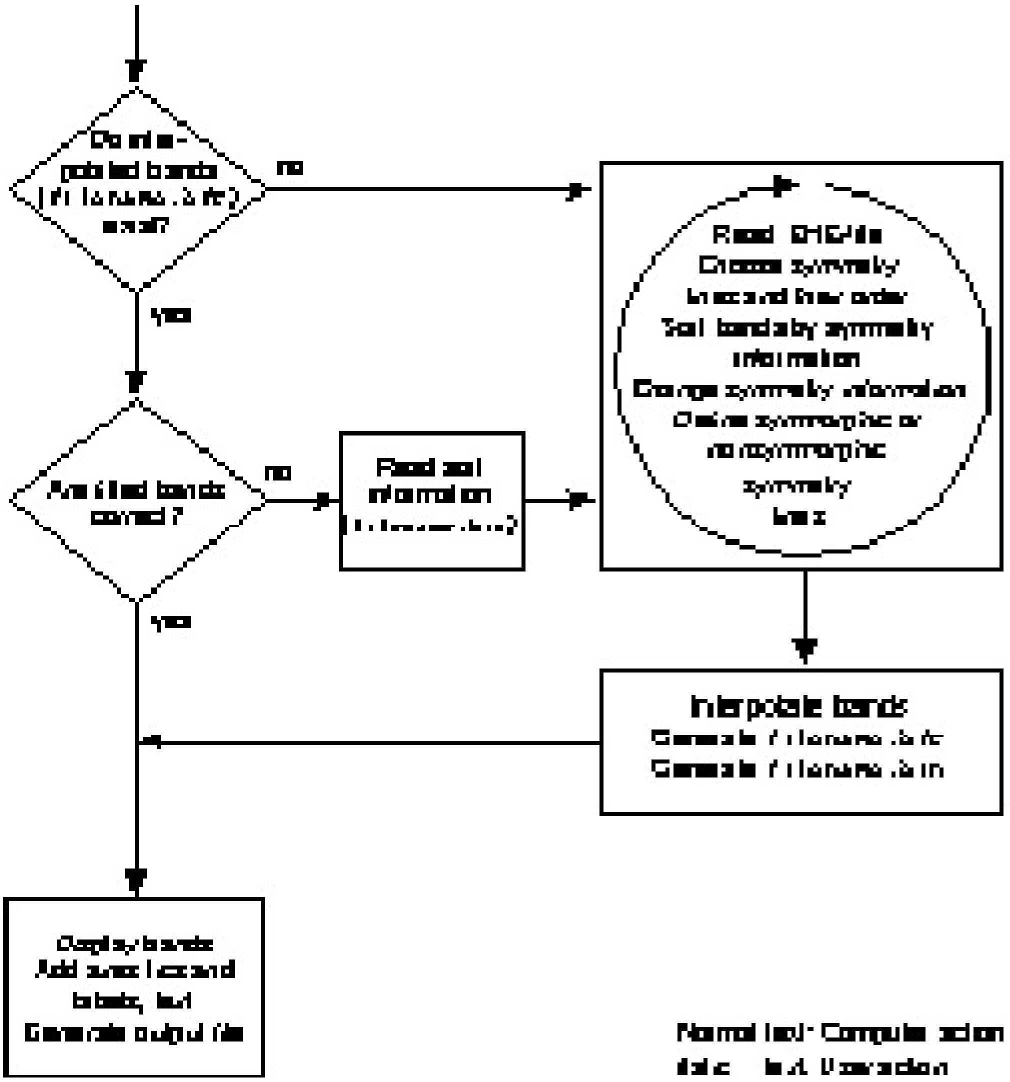


Figure 13: Flowchart of BANDPLOT usage.

After answering how many and which bands you want to interpolate, you enter a loop starting with

```
Symmetry line labels:   1 3 2
Choose label of line (0=exit and interpolate
previously chosen lines):
```

where you have to select the symmetry lines you want to plot and that you have labeled with numbers in your EHMACC input file (see card #24, **KPIN**). Select the symmetry lines in the order you want them plotted.

The program searches all **k**-points belonging to a chosen symmetry line and asks, if they are in ascending or descending order. Then it sorts the bands according to the symmetry information given in `filename.ehe`. You enter a small menu that allows to change the sorting order of the bands manually:

```
(1) Change energy range
(2) Change crossing of bands
(3) Copy one symmetry set to another
(4) Get back original noncrossed state
(5) Exit

Selection:
```

Enter energies with a decimal point. Options (2) and (3) deserve further attention:

With option (2) you can change the crossing of two neighbor bands. For example you know that band 2 and 3 have to be crossed between the third and the fourth **k**-point in Figure 15. The program draws nodes at the location of each calculated energy level.

1. Select[1] first on node 3 of band 2 (Figure left).
2. Select then node 4 of band 3 (Figure right).

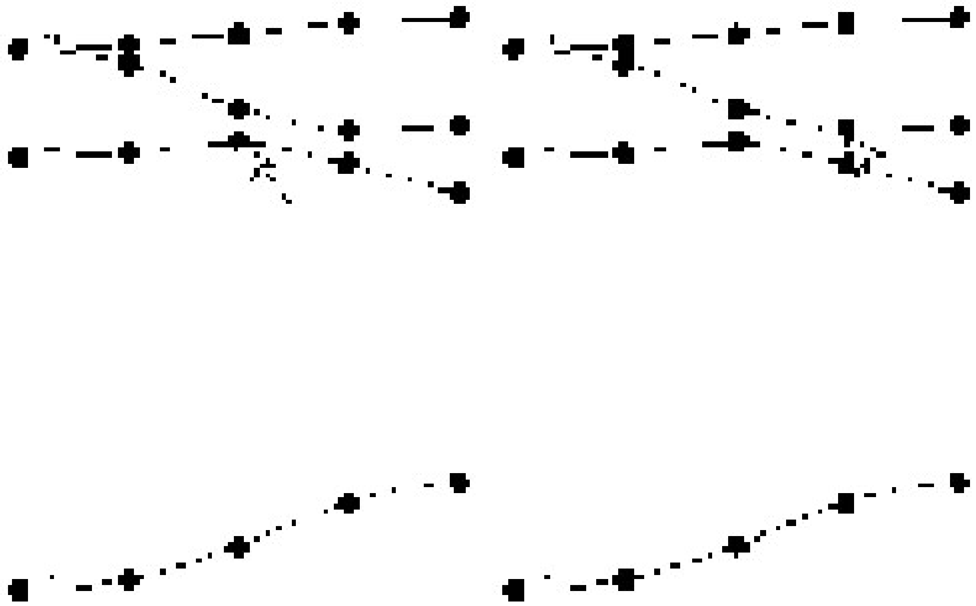


Figure 14: Crossing of two bands. Select nodes as indicated from left two right.

If you want to uncross two bands, you have to follow the nodes of either one band:

1. Select[1] first on node 3 of band 2 (Figure left).
2. Select then node 4 of band 3 (Figure right).

---

[1] Windows NT/PGPLOT version: Move mouse cursor to graphics window. Click left mouse button. The cross follows your mouse pointer. Position the cross at the node. Press any keyboard key.

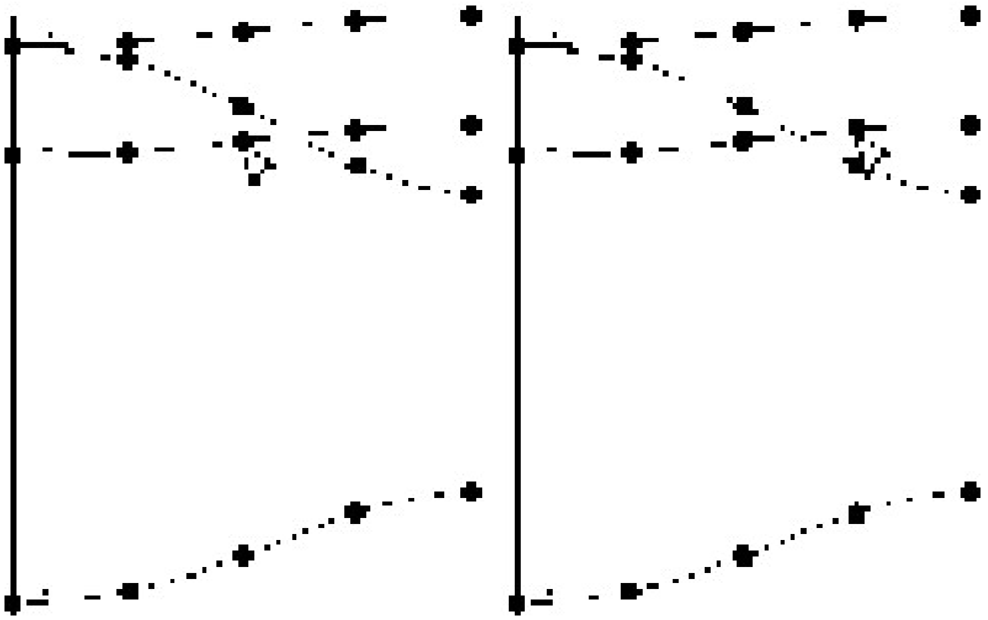


Figure 15: Uncrossing of two bands. Select nodes as indicated from left to right.

Option (3) is useful if the program is not able to cross some bands correctly due to degenerate levels at the beginning or the end of a symmetry line. For example, you have chosen the Σ symmetry line in the square carbon lattice example (`c_gitter.ehi`). At the M symmetry point (**k**=(0.5,0.5)), band 2 and band 3 are degenerate, and the symmetry analysis results in two zeros for the mirror plane operation perpendicular to the carbon lattice plane and along the Σ symmetry line, see Listing 3 of `c_gitter.ehe`. At the next **k**-point along the Σ line, this degeneracy is removed.

The band sorting algorithm of BANDPLOT takes the symmetry information (bold numbers in Listing 3) of the first energy level at the first **k**-point of the Σ line and tries to track the path of this symmetry information of band #1 along the **k**-points of the Σ line. In this example this leads to the result that band #1 will connect all energy levels #1. Then the program continues with the symmetry information of energy level #2 (**`1`** ***`0`***), trying to track the path of band #2. Since the degeneracy is removed at **k**-point #2, the algorithm can’t find a path for this band, and will therefore retry the path tracking with the symmetry information at **k**-point #2. A simple means to correct this wrong behaviour is to replace the two bold-italic ***`0`*** at **k**-point #1 by a **`-1`** and a **`+1`**, which are the symmetry labels evolving when the symmetry is lowered. One can do this by either editing `c_gitter.ehe` or by using menu option (3), which allows to copy a symmetry label set from e.g. **k**-point #2 to **k**-point #1 for a certain band range. The bands are then sorted correctly.

```
 .5000000 .5000000 .0000000                                    (k-point 1 of Σ line)
```

```
  2 1 1 0 0 0 0 0 0 0 0 0                                     (symmetry line info: line
  3 2 1 3 0 0 0 0 0 0 0 0                                      label, number of symmetry
  0 0 0 0 0 0 0 0 0 0 0 0                                       operations, op1 op2 … ...)
 -17.59524    .00000    .00000    .00000   1  1 0 1 0 0 0 0 0 0 0    (energies, gradients,
weights,
 -12.49274    .00000    .00000    .00000   2  1 0 0 0 0 0 0 0 0 0    band number,
 -12.49274    .00000    .00000    .00000   3  1 0 0 0 0 0 0 0 0 0    symmetry analysis info)
 -10.49690    .00000    .00000    .00000   4 -1 0 1 0 0 0 0 0 0 0
  .3750000 .3750000 .0000000                                          (k-point 2 of Σ line)
  3 2 1 3 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
 -19.16462   9.88839   9.88839    .00000   1  1 0  1 0 0 0 0 0 0 0
 -12.14305  -2.77437 -2.77437     .00000   2  1 0 -1 0 0 0 0 0 0 0
 -11.28960  -7.03613 -7.03613     .00000   3  1 0  1 0 0 0 0 0 0 0
 -10.76739   2.03505   2.03505    .00000   4 -1 0  1 0 0 0 0 0 0 0
  .2500000 .2500000 .0000000                                          (k-point 3 of Σ line)
  3 2 1 3 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
 -21.66559   9.24911   9.24911    .00000   1  1 0  1 0 0 0 0 0 0 0
 -11.39906   2.72538   2.72538    .00000   2 -1 0  1 0 0 0 0 0 0 0
 -11.15133  -4.94207 -4.94207     .00000   3  1 0 -1 0 0 0 0 0 0 0
  -9.78675  -4.29229 -4.29229     .00000   4  1 0  1 0 0 0 0 0 0 0
  .1250000 .1250000 .0000000                                          (k-point 4 of Σ line)
  3 2 1 3 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
  0 0 0 0 0 0 0 0 0 0 0 0
 -23.52083   5.25372   5.25372    .00000   1  1 0  1 0 0 0 0 0 0 0
 -11.99081   1.78681   1.78681    .00000   2 -1 0  1 0 0 0 0 0 0 0
  -9.88831  -4.50006  -4.50006    .00000   3  1 0 -1 0 0 0 0 0 0 0
  -9.23608   -.41434   -.41434    .00000   4  1 0  1 0 0 0 0 0 0 0
```

Listing 2: Part of `c_gitter.ehe` output file.

After selection of option (3) you might enter:

```
Enter k-point number to copy symmetry info from (min=1, max= 5): 2
Enter k-point number to copy symmetry info to (min=1, max= 5): 1
Lower band (min= 1, max= 4): 2
Higher band (min= 1, max= 4): 4
```

If you agree with the band structure along the chosen symmetry line, choose menu option (5). After answering some questions you return to

```
Symmetry line labels:   1 3 2
Choose label of line (0=exit and interpolate
previously chosen lines):
```

where you can select the next symmetry line or exit for the interpolation of the bands along the selected symmetry lines.

After interpolation you enter the main menu:

```
(1) Display interpolated bands
(2) Change .BIN information and interpolate bands
```

```
(3) Interpolate bands

(0) Exit

Selection:
```

You will always start up here, if interpolated band data and band sorting information exist in `filename.bft` and `filename.bin`, respectively. Option (2) is for the correction of the band sorting info. Choose option (3), if you might have changed the band sorting information in `filename.bin` by hand. For the final display and print-out select menu option (1). After answering some questions you enter the

4.4.5 Graph modification menu

```
( 1) Change energy range
( 2) Add border
( 3) Add energy axis tic marks and tic labels
( 4) Add symmetry axes labels
( 5) Add axes titles
( 6) Add optional text (e.g. symmetry labels)
( 7) Mark fermi energy
( 8) Change symmetry line lengths
( 9) Clear and redraw
(10) Generate output file
(11) Save graph layout
(12) Load graph layout

( 0) Quit program

Selection:
```

Menu option (4) allows to set the names for the high symmetry points of the *Brillouin* zone. Menu option (5) sets the symmetry line labels and the energy axis title. All texts (options (4-6)) can be input as described in chapter 4.3.4.2.

4.4.5.1 Printing

Windows NT version:

Two possibilities exist for printing:

High quality vector output: Select `Generate output file` in the main menu.

For quick printing or saving of a bitmap: Menu at the top left, „`File/Print ...`” or „`File/Save ...`”. Your graphics card must be set to 256 colors.

4.4.6 Format of graph layout file `filename.bpl` (`UNIT 15`)

Line	Variables	Format
1	NSYM NSYM: Number of symmetry lines	I2
2	FSELEC(I),I=1,MAXSEL FSELEC: Graph options, if true (T), will be perfomed FSELEC(2): Border FSELEC(3): Axes tic marks FSELEC(4): Axes labels FSELEC(5): Symmetry line and energy axis titles FSELEC(6): Free text FSELEC(7): Fermi level	20L1
3	VPXOFF,VPYOFF,VPXAXW,VPYAXH,VPXFUL,VPYFUL VPXOFF: Left-lower edge of x-axis in viewport coordinates VPYOFF: Left-lower edge of y-axis in viewport coordinates VPXAXW: Length of x-axis in viewport coordinates VPYAXH: Height of y-axis in viewport coordinates VPXFUL: Full x viewport VPYFUL: Full y viewport	6F10.5
4 … 4+NSYM	VPXSCT(I),I=1,NSYM+1 VPXSCT: Length of symmetry lines in relative coordinates	F10.5
5+NSYM	EMIN,EMAX,EFERMI EMIN: Lower boundary of energy range [eV] EMAX: Higher boundary of energy range [eV] EFERMI: Fermi level [eV]	3F10.5
6+NSYM	ETCMIN,ETCMAX,TICDIS,TICDS2 ETCMIN: Lower boundary of tic range [eV] ETCMAX: Upper boundary of tic range [eV] TICDIS: Distance between two tics [eV] TICDS2: Distance between two labelled tics [eV]	4F10.5
7+NSYM	ITICSD(I),I=1,NSYM+1 ITICSD: Axis side of tic marks (0: none, 1: left, 2: right, 3:both)	20I1
8+NSYM … 8+2*NSYM	CLABEL(I), I=1,NSYM+1 CLABEL: Axis labels.	A80
8+2*NSYM … 7+3*NSYM	CXAXLB(I),I=1,NSYM CXAXLB: Symmetry line titles	A80
8+3*NSYM	CYAXLB CYAXLB: Energy axis title	A80
9+3*NSYM	ITXT ITXT: Number of free texts	I4

10+3*NSYM … 9+3*NSYM+ITXT	TEXTXY(I,1),TEXTXY(I,2),TXTANG(I),ITXPOS(I), CTEXT(I),I=1,ITXT TEXTXY: Coordinates of text TXTANG: Text angle ITXPOS: Text position CTEXT: Free text	3F10.5,I2,A80
10+3*NSYM+ITXT	FBORDR,FFLINE FBORDR: Border FFLINE: Line marking *Fermi* level	2L1

4.4.7 Format of `filename.bin`

Line	Variables	Format
1	NBTOT,NNOFIT,NBFIT NBTOT: Total number of bands NNOFIT: Number of bands not to be interpolated (band 1 to NNOFIT) NBFIT: Number of bands to be interpolated (band NNOFIT+1 to NNOFIT+NBFIT).	3I4
2	ILABEL,NK,ISYMMS,ISYMME,IUSEEN,IUSESY ILABEL: Label of symmetry line NK: Number of **k**-points along symmetry line ILABEL. ISYMMS: =0: Start **k**-point of symmetry line symmorphic, else nonsymmorphic. ISYMME: =0: End **k**-point of symmetry line symmorphic, else nonsymmorphic. IUSEEN: =0: Neither use start and end point derivatives. =1: Use start point derivatives. =2: Use end point derivatives. =3: Use both start and end point derivatives. IUSESY: =0: Use symmorphic point derivatives.	6I3
3	KPOINT KPOINT: Index to **k**-point and its band energies and symmetries to be read in from `filename.EHE`.	I4
4 … 3+NBTOT/20	IORD(I,K),I=1,NBTOT IORD: Pointer to energy level associated with band I of **k**-point K.	20I4

Table rows 3 and 4 are repeated NK times per symmetry line.

Table rows 2-4 are repeated as many symmetry lines have been chosen.

4.4.8 Format of `filename.bft`

Line	Variables	Format
1	NSYM NSYM: Number of symmetry lines	I2
2	NPTS,NBAND NPTS: Number of interpolated mesh points per band NBAND: Number of bands to be plotted	2I3
3 … 2+NBAND*NPTS	((BNDENE(J,K),J=1,NBAND),K=1,NPTS) BNDENE: Interpolated mesh points of bands.	F10.5

Table rows 2 and 3 are repeated for each symmetry line.

4.4.9 Compiling and linking

Makefiles are provided for Microsoft Fortran Powerstation 4.0 for Windows NT. They, however, contain the path names of the original distribution. To obtain the correct path names of your installation, remove the files in the FileView window within Microsoft Developer Studio, and reinsert them again by using the menu entry `Insert/Files into Project...`.

The sources must be linked with the PGPLOT graphics subroutine library. The PGPLOT library is available at `http://astro.caltech.edu/~tjp/pgplot/`.

Porting to different operating systems or different graphic libraries (e.g. NAG[38] and GKS[39]) is possible. We do, however, not give any support for this.

Some small changes due to different file access in different operating systems have to be made to the sources, see `READ.ME` file.

The file `sizes` contains and describes all relevant parameters that change the maximum number of atomic orbitals, the number of curve buffers etc. Also the default values for the size of the plots can be changed. It is, however, recommended that these default parameters are not changed, since the ratio of the axes, $width/height = 1/\sqrt{2}$, conforms to european DIN paper size settings (A4). The plot dimensions should be compatible to the graphs generated by DOSPLOT, see includefile `sizes`.

4.5 CEDiT

4.5.1 Description

CEDiT (Crystal Electronic Dipole-induced Transitions) is a program that allows the calculation of oscillator strengths in solids based on extended-Hückel tight-binding crystal orbitals. The program was written by R. Rytz in the laboratory of G. Calzaferri, Department of Chemistry and Biochemistry, University of Berne, Freiestrasse 3, CH-3012 Bern.

4.5.2 Invoking the program, input and output files

All input and output files are in ASCII format. No scratch files will be written. Filenames are free, but fixed extensions have to be used.

Input:	`filename.edi`	- Holds commands for the CEDiT program.
	`filename.gen`	- Holds principle quantum numbers and *Slater* exponents.
	`###.kpe`	- Hold wavefunctions, energies, and geometry at each **k**-point calculated.
Output:	`filename.edo`	- Collects all results of a CEDiT calculation.
	`###.eds`	- Holds transition energies and oscillator strengths per **k**-point.

First an EHMACC calculation must be carried out to generate the respective crystal orbitals and energies. The keyword **EDITOUT** must be used. The relevant information for the CEDiT calculation will be stored in the so-called `*.kpe` files. Electronic transitions may occur at any **k** point in the irreducible *Brillouin* zone, and are thus a property of the solid. Thus, make sure to include enough points with the respective weighting factor in `filename.ehi` analogous to a property calculation (cf. Chapters 3.3.1 and 3.5). Then CEDiT has to be invoked for the calculation of the oscillator strengths.

Windows NT version:

1. Open a DOS command prompt window.
2. Type `ehmacc.bat inputfilename`

   No extension is needed: An input file with the extension `.ehi` is assumed.

   If you intend to start a calculation with the **RESTART** option (see Chapter 4.1.3), type `ehmacc.bat inputfilename restart`
3. After successful termination of the calculation you will find the following new files in your working directory: `inputfilename.eh1`, `inputfilename.ehc`, `inputfilename.eho`, `inputfilename.kp`, and a series of files that are continuously numbered `001.kpe` - `###.kpe`, `###` being the number of the last **k**-point. The latter contain the relevant information for the CEDiT calculation.
4. `inputfilename.edi` is needed to control the CEDiT program. `inputfilename.gen` supplies the principle quantum numbers and the *Slater* exponents.
5. Type `CEDiT <inputfilename.edi >inputfilename.edo`

6. A successful calculation will result in the output of the files `inputfilename.edo` and `001.eds` - `###.eds`.

4.5.3 Format of the input file `filename.edi`

The file consists of three sections: A title, a keyword section, and a section containing a list of stanzas. A line comprising an exclamation mark „!” is considered a comment.

Card **#1 TITLE**

Number	:	1
Requirement	:	None
Object	:	Title for user’s convenience.
Sequence	:	None.
Example	:	`TITLE Polyacetylene`

Card **#2 KEYWRD** keywords

Number	:	Unlimited.
Requirement	:	None.
Object	:	Controls the input sequence, the calculation path and the output options.
Sequence	:	None. The keywords must be separated by either commas or blanks.

Example	:	`KEYWRD GENFILE=polye136.gen`
		`KEYWRD CRYST`
		`KEYWRD POL=XYZ, MATRICE=ALL`

List of keywords:

GENFILE=	:	Name of the `*.gen` file without extension.
MOFILE=	:	Name of the `*.mo` file. In the case as discussed here where we calculate electronic transition intensities in solids, the `*.mo` files are called `*.kpe` (one file per **k** point). The range of the `*.kpe` files to be processed is determined by the range of **k** points (see stanza [KPOINTS]). Thus, an asterisk has to be declared.
POL=	:	X, Y, and Z give the respective polarisations of the transitions. Any combination and order are acceptable.
MATRICE=	:	Here, you may choose between the two switches ALL and BDIAG. BDIAG causes the program to consider only the block-diagonal parts of the transition matrix. Do not use this option unless you are absolutely sure what you are doing. ALL is default
TRANSOUT	:	Prints the transition energies and oscillator strengths.
CRYST	:	Tells CEDiT to start a crystal calculation. This keyword is indispensable for the computation of electronic transitions in solids.
SPECOUT	:	Releases printing of `*.eds` files (one per **k** point). Such files are suited for the simulation of a spectrum by fitting Gaussian lobes to the worked out line sepctrum. An example of a `*.eds` file is following up this list.
MATOUT	:	Causes the output of the transition-dipole-length matrix. The values are in atomic units of length (a.u.).
CORRECT	:	Corrects for possible degeneracies of the initial crystal orbital (see Eq. (12) of Ref. [xxx]).
TOLERANCE=	:	Used in connection with CORRECT. If ΔE is smaller than this value, two neighboring wave functions are considered degenerate. Default is 1.0×10^{-5}.

Following the keywords we have a section of three stanzas that are **[WAVERANGE], [CELL]**, and **[KPOINTS]**. They can be arranged any way you like.

Card #**3 [WAVERANGE]**

Number : 1

Requirement : 1 card required.

Object : Defines start and end wave function for the transition. Comprises two entries START and END both of which are succeeded by the respective number(s) of the wave function(s). It is possible to specify a range.

Sequence : [WAVERANGE]
START i1 i2 (i1=number of first wave function, i2=number of last wave function)
END j1 j2 (j1=number of first wave function, j2=number of last wave function)

Example :

```
[WAVERANGE]
START 4 6
END 7 9
```

Card #**4 [CELL]**

Number : 1

Requirement : 1 card required.

Object : States the unit cell basis vectors and the number of neighbor cells. The vector(s) have to be given in Cartesian coordinates x y z. The number n of neighbour unit cells should be of the order of half the number used for the actual band structure calculation. That is, if the band structure is worked out with four neighbor cells (cf. `*.ehi`), the declaration of two cells for CEDiT should do. Hint: The unit cell basis vectors are output in the respective `*.eho` file.

Sequence : [CELL]
1st vector x y z n neighbors
2nd vector x y z n neighbors
3rd vector x y z n neighbors

Example :

```
[CELL]
1st vector 1.299038  .000000  2.110000 2 neighbors
```

Card #**5 [KPOINTS]**

Number : 1

Requirement : 1 card required.

Object : Defines the range of **k** points using the entries START and END. Make sure that the corresponding files `*.kpe` are located in the same directory as your input files. If you should have more than one computer available, it is always possible to split the computation amongst different machines by creating more than one `*.edi` file and distributing the respective *.kpe files. For example, if you have decided to consider 90 **k** points in the irreducible part of the Brillouin zone you may wish to generate the files filename_1-45.edi and filename_46-90.edi with the [KPOINTS] entries (START 1 END 45) and (START 46 END 90), respectively.

Sequence : [KPOINTS]
START I
END j

Example :

```
[KPOINTS]
START 1
END 31
```

4.5.4 Format of the input files `*.kpe`

These files are produced by EHMACC. See Chapter 4.1.5.10.

4.5.5 Format of the input file `filename.gen`

In addition to the information comprised in the `*.kpe` files, CEDiT needs the *Slater* exponents and principal quantum numbers of the atomic orbitals. These must be specified in a file with the extension `*.gen`. The reason for keeping this information in a separate file is due to the history of the oscillator strength calculation program that was first written to work out optical spectra of molecules. The respective program (EDiT) is part of the ICON-EDiT program package, available at `http://iacrs1.unibe.ch`. The crystal part was added two years later and the program given the name CEDiT. For compatibility reasons we kept to the convention of reading the *Slater* exponents and the principal quantum numbers from `filename.gen` that is the most frequently used input file of the ICON-EDiT package.[44] It is in fixed FORTRAN format. If you do not like editing a `*.gen` file in a text editor of your choice putting up with the hassle of the somewhat strict text format, you can still download the ICON-EDiT package and create a `*.gen` file with the self-explanatory input program INPUTC (for instructions, see the hint below). Most of the information contained in a regular `*.gen` file is not used by CEDiT but taken from the respective `*.kpe` files. Thus, it is not difficult to generate a `"*.gen"` file comprising enough information for CEDiT. An example, namely `polye136.gen` is given below:

Line 1	` * C * H`
Line 2 and following	`C   4   21.7100-21.40   21.6250-11.40` `H   1   11.3000-13.60`

For the exact spacing see the original file included in the `examples` directory.

The general format is:

Line	FORMAT	Explanation
1	40A2	Element symbols. An asterisk initiates reading of the next unknown atom definition card (line 2 amd following). As at the beginning there are no known atoms, line 1 always has to start with `"*"`. The sequence has to agree with the atom positions in the respective `*.kpe` files.

2 and following	A2,I3,3(I3,2F6.3),F6.4,F6.3,F6.4	Atom definition card (as defined in INPUTC). Sequence SYMB VELEC NS EXPS COULS NP EXPP COULP ND EXPD1 COULD C1 EXPD2 C2 SYMB: Symbol of atom. VELEC: Number of valence electrons. NS: Principal quantum number of s electron. EXPS: *Slater* exponent of s orbital. COULS: VOIE of s electron. NP: Principal quantum number of p electron. EXPP: *Slater* exponent of p orbital. COULP: VOIE of p electron. ND: Principal quantum number of d electron. EXPD1: First *Slater* exponent of d orbital. COULD: VOIE of d electron. C1: Coefficient of EXPD1. EXPD2: Second *Slater* exponent of d electron. C2: Coefficient of EXPD2.

CEDiT does not know of any specific atom names, but only cares for the *Slater* exponents and the principal quantum numbers. Thus you may invent your own names. Changing `polye136.gen` to:

Line 1	`* G * H`
Line 2 and following	`G   4   21.7100-21.40   21.6250-11.40` `H   1   11.3000-13.60`

results in the same CEDiT calculation.

<u>Hint:</u> For unit cells comprising many atoms we suggest the following procedure in order to get a valid `*.gen` file:

1. Download ICON-EDiT from `http://iacrs1.unibe.ch/`.
2. Install INPUTC.
3. Start INPUTC.
4. Choose `Option #1 Cartesian Coordinates`.
5. Give a `filename` (without extension).
6. Type in the atomic symbols of the atoms comprised in your unit cell in the same order as in the respective `*.ehi` file.
7. INPUTC will then ask for the Cartesian coordinates. Just type <Return> until the program quits.
8. Delete all entries in `filename.kar` up to the first line that starts with an asterisk.
9. Rename `filename.kar` to `filename.gen`.

Make sure that the *Slater* exponents in `filename.gen` and `filename.ehi` agree!

4.5.6 Output files

Two different types of files may be output upon a CEDiT calculation. These are `filename.edo` and if you have specified the keyword SPECOUT a number of `*.eds` files that correspond to the number of considered **k** points. The `*.eds` files contain the essence of the calculation in a form that is convenient for further manipulation in order to generate absorbtion spectra. This can be done, e.g, by fitting Gaussian lobes at the observed line spectra.

Note: Even though kept in different `*.eds` files all transitions at all **k** points contribute to the respective spectrum. Thus, prior to use this information in oder to simulate an absorption spectrum, you may wish to merge the contents of the `*.eds` files in `all.eds`. Windows NT version: Type

```
for %i in (*.eds) do type %i >>all.eds
```

Just below, we give the `001.eds` file:

```
75293.4   0.00000000   0.00000000   0.00968286   0.00968286     5     6
```

The general format is as follows:

WAVNUM,OSCX,OSCY,OSCZ,TOTOSC,INMO,FIMO		F8.1,2X,4(F11.8, 2X),2(I4,2X)
WAVNUM:	Transition energy in wave numbers.	
OSCX:	Corrected oscillator strength in *x* polarisation	
OSCY:	Corrected oscillator strength in *y* polarisation.	
OSCZ:	Corrected oscillator strength in *z* polarisation.	
TOTOSC:	Isotropic corrected oscillator strength.	
INMO:	Number of the initial wave function.	
FIMO:	Number of the end wave function.	

4.5.7 Compiling and linking the program

Some small changes due to different file access in different operating systems have to be made to the sources, see `READ.ME` file.

All relevant parameters for the maximum number of atomic orbitals etc. are described in the include-file `sizes` and can be changed there.

Makefiles are provided for Microsoft Fortran Powerstation 4.0 for Windows NT. They, however, contain the path names of the original distribution. To obtain the correct path names of your installation, remove the files in the FileView window within Microsoft Developer Studio, and reinsert them again by using the menu entry `Insert/Files into Project...`.

4.6 TOOLS

Written by Martin Brändle in the laboratory of G. Calzaferri, Department of Chemistry and Biochemistry, University of Berne, Freiestr. 3, CH-3012 Berne.

4.6.1 k-point set programs and k-point sets

Sources have been added to the package for programs which generate an average properties **k**-point set for some specific *Brillouin* zones:

k-point set		Program	Reference
1D:	Linear mesh	linear.for	None
2D:	Square, type 1	squar1.for	[21]
	Square, type 2	squar2.for	[21]
	Primitive rectangular, type 1	prect1.for	[21]
	Primitive rectangular, type 2	prect2.for	[21]
	Centered rectangular, type 1	crect1.for	[21]
	Centered rectangular, type 2	crect2.for	[21]
	Hexagonal, type 1a	hexag1.for	[21]
	Hexagonal, type 2b	hexag2.for	[21]
3D:	Simple cubic	sc.for	According to [22]
	Body-centered cubic	bcc.for	[22]
	Face-centered cubic	fcc.for	[22]

Simply compile them on your preferred operating system. In every program, an optional string may be entered that is repeated after every **k**-point described by the card **KPIN** (see chapter 4.1.3). Such a string may consist of a **KPOUT** statement, but may also be empty.

4.6.2 DOSAPP

DOSAPP merges two `filename.eh5` files into a third one. This can be used for generating DOS and COOP curves with gaussian functions fitted with different width within the same plot, especially when it is difficult to find a good δ-value for different energy regions having small and very large band dispersions. It is required that the highest energy mesh point of the first input file equals the lowest one in the second input file.

Windows NT version:

1. Open a DOS command prompt window
2. Type `dosapp.bat filename1.eh5 filename2.eh5 filename3.eh5`

4.6.3 TRANSC

TRANSC converts cartesian coordinates to fractional coordinates and vice versa. A SCHAKAL-compatible file (`filename.ehc` generated by EHMACC) is needed as input. Conversion from fractional to Cartesian coordinates is done without user interaction. Conversion from Cartesian to fractional coordinates invokes a user dialogue. The conversion algorithm[39] presumes that the a axis of the fractional coordinate system coincides with the x axis of the cartesian coordinate system, and that the b axis lies in the x-y-plane. The relative position of the coordinate systems can be specified (origin and rotation).

Window NT version:

1. Open a DOS command prompt window
2. Type `transc`

The program will then ask for the names of the input (UNIT `1`) and the output file (UNIT `2`).

5 EXAMPLES

The examples are located in the directory `examples`. Copy the input files to your working directory.

5.1 1D system - all-*trans*-polyacetylene

Comparison of the band structures and of the DOS of all-trans-polyacetylene in its alternating geometry (r(C_1-C_2)=1.36Å, r(C_2-C_a)=1.50Å) and non-alternating geometry (r(C_1-C_2)= r(C_2-C_a)=1.43Å). See also [46],[16].

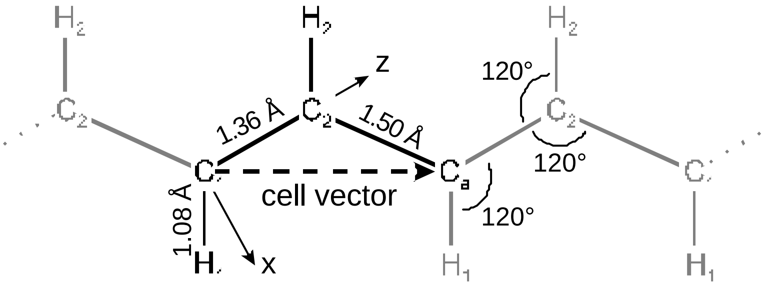


Figure 16: Alternating geometry of all-*trans*-polyacetylene, definition by internal coordinates

Band structure calculations

Input files: `polyb136.ehi`, `polyb143.ehi`

In the non-alternating case, the point group of the symmetry line contains an additional screw axis $\bar{2}$ (or a glide plane) going through the centres of the carbon-carbon bonds. Therefore the line group of non-alternating all-*trans*-polyacetylene is nonsymmorphic. This results in an additional crossing of band 3 and band 5 around the middle of the symmetry line, which is removed upon distortion to the alternating form. Since the symmetry analysis part of EHMACC can not handle screw axes and glide planes, you have to cross the two bands by hand in the BANDPLOT program. At the end of the symmetry line, the energy gradients with respect to k are non-zero.

Properties calculations

Input files: `polyd136.ehi`, `polyd143.ehi`, `polyd136.epi`, `polyd143.epi`

Oscillator strength calculation

Input files: `polye136.ehi`, `polye136.edi`, `polye136.gen`

5.2 1D system - all-*trans*-polyacetylene with geometry variation

Variation of length of alternating carbon-carbon bonds in all-*trans*-polyacetylene. See also [16].

Input file: `polyvary.ehi`

For the determination of the minima use the generated `polyvary.ene`.

5.3 2D system - Square carbon array

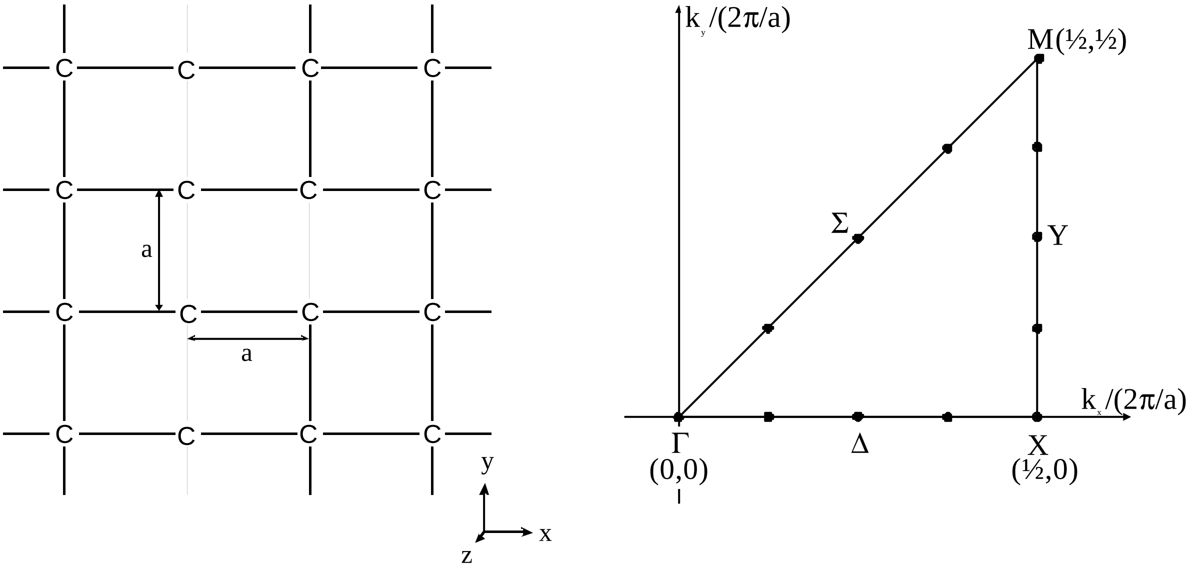


Figure 17: Left: Square carbon lattice. The unit cell contains one atom. Right: Symmetry lines along which the band structure is calculated.

Band structure calculation

See also chapter 3.3.
Input file: `c_gitter.ehi`

Properties calculation

Input files: `c_gdos.ehi`, `c_gdos.epi`

5.4 2D system - Molybdenum sulfide[47] slab ${}^{2}_{\infty}[\mathrm{MoS}_2]$

Many transition-metal dichalcogenides display a characteristic layered structure. Two-dimensional slabs are formed by two layers of close-packed chalcogenide atoms sandwiching one metal layer between them in MX_2 (X=S,Se). Then these layers are stacked, with just van der Waals contacts between them. The two chalcogenide layers forming a slab can be stacked directly above each other, making trigonal prismatic holes for the metals. This is shown in Figure 18 for MoS_2. Alternatively, the layers may stagger forming octahedral holes. The general features of the energy bands of these two structures have been discussed.[48]

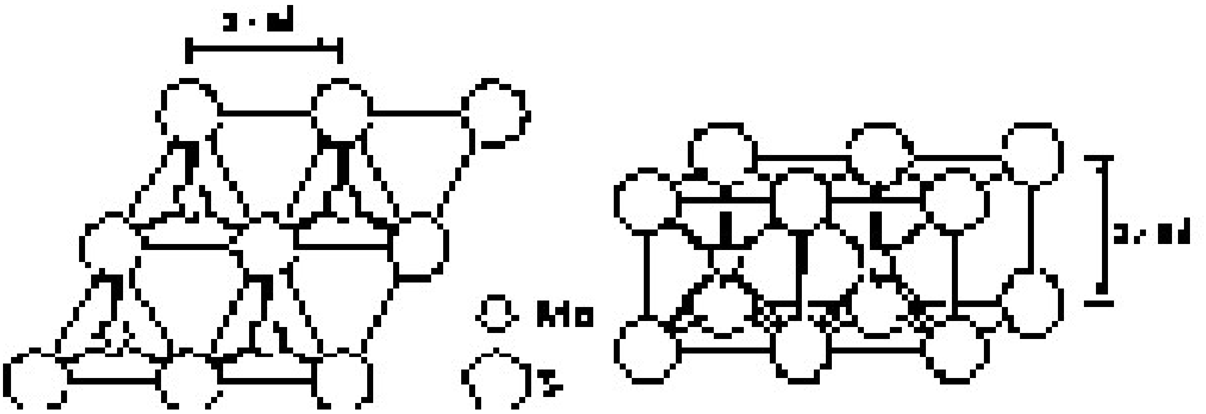


Figure 18: MoS_2 layer structure with trigonal prismatic coordination of Mo. Distances according to [49]. Layer group p3m1/m.

The direct and reciprocal lattice vectors of this structure with layer group p3m1/m and the symmetry operations used in the symmetry analysis part valid for the symmetry lines Σ, T' and T of the hexagonal *Brillouin* zone are shown in Figure 19. The properties are calculated over 91 k-points in the irreducible wedge of the hexagonal BZ.

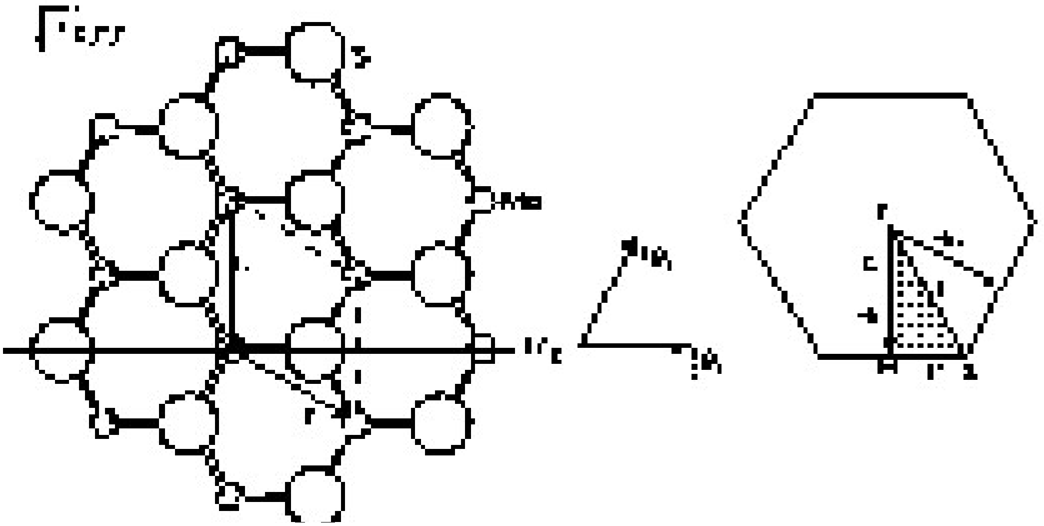


Figure 19: Left: Direct and reciprocal lattice vectors in ${}^{2}_{\infty}[\mathrm{MoS}_2]$, layer group p3m1/m, and symmetry elements valid for symmetry lines Σ, T' and T. Right: Hexagonal *Brillouin* zone and irreducible wedge.

Band structure calculation

Input file: `mos2band.ehi`

Properties calculation

Input files: `mos2prop.ehi`, `mos2prop.epi`

Oscillator strength calculation[22] (time consuming)
Input files: `mos2osc.ehi`, `mos2osc.edi`, `mos2osc.gen`

5.5 Molecules on surfaces: Carbon monoxide on Ni(100)[45], [3]

Adsorption of a molecule on a surface is best modelled by a slab model. The bulk structure is extended infinitely into two dimensions and periodic boundary conditions are applied, and the adsorption is described in the third dimension. A monolayer of carbon monoxide, a four layer slab of Ni(100), and CO adsorbed on the Ni(100) will be studied in this example. Refer to [45] and [3], p. 23ff, and p.71ff.

The structure of Ni(100) with adsorbed carbon monoxide is shown in Figure 20. The primitive unit cell of Ni(100) is square, with a side length of 2.49 Å being the distance of the Ni atoms in the metal. The coverage with CO molecules is 0.5, e.g. one CO molecule per two unit cells of the Ni(100) slab. Therefore a supercell description is applied to Ni(100) slab and to the surface with adsorbed CO, extending the cell dimensions to $\sqrt{2}$ times the unit cell length of Ni(100) and therefore doubling the unit cell area.

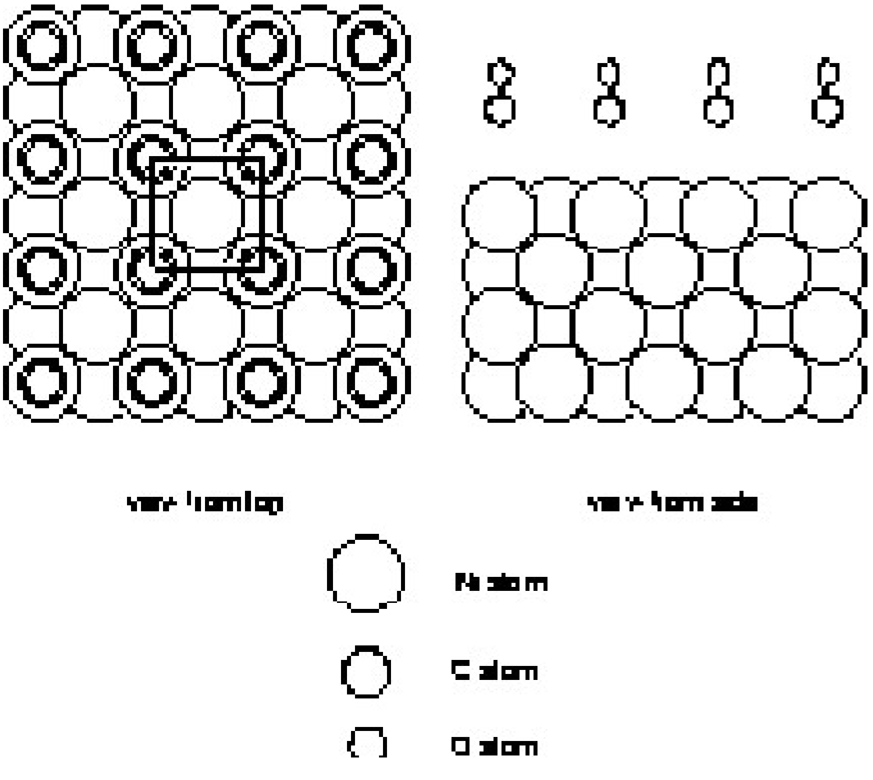


Figure 20: Structure of Ni(100) surface with 0.5 coverage of carbon monoxide.

A series of input files is provided.

Band structure calculations

CO(100) monolayer input file: `co_100ba.ehi`. Investigate here the interaction of CO molecules as a function of distance or coverage, respectively.

Ni(100), four layers, input file: `ni_100ba.ehi`

CO on Ni(100), four layers, calculation at Γ point: `co_nigam.ehi`. This file is provided for the analysis of the FMO contributions.

Properties calculations

CO(100) monolayer input file: `co_100ds.ehi`, `co_100ds.epi`

Ni(100), four layers, input file: `ni_100ds.ehi`, `ni_100ds.epi`

CO on four layers of Ni(100), coverage 0.5,

distance Ni-C=1.8 Å, input file: `co_ni_18.ehi`, `co_ni_18.epi`

distance Ni-C=3.0 Å, input file: `co_ni_30.ehi`, `co_ni_30.epi`

distance Ni-C=4.5 Å, input file: `co_ni_45.ehi`, `co_ni_45.epi`

5.6 3D system - Silica sodalite $^3_\infty[Si_{12}O_{24}]$

The structure of silica sodalite, a clathrasil compound, is shown in Figure 21. It consists of bcc-packed cuboctahedral sodalite cages, which are also a building unit of zeolite A.[52] The space group is $Im\bar{3}m$, with unit cell length a=8.83Å. The Si-O bond length is 1.587 Å, the Si-O-Si angle 159.7°.[50]

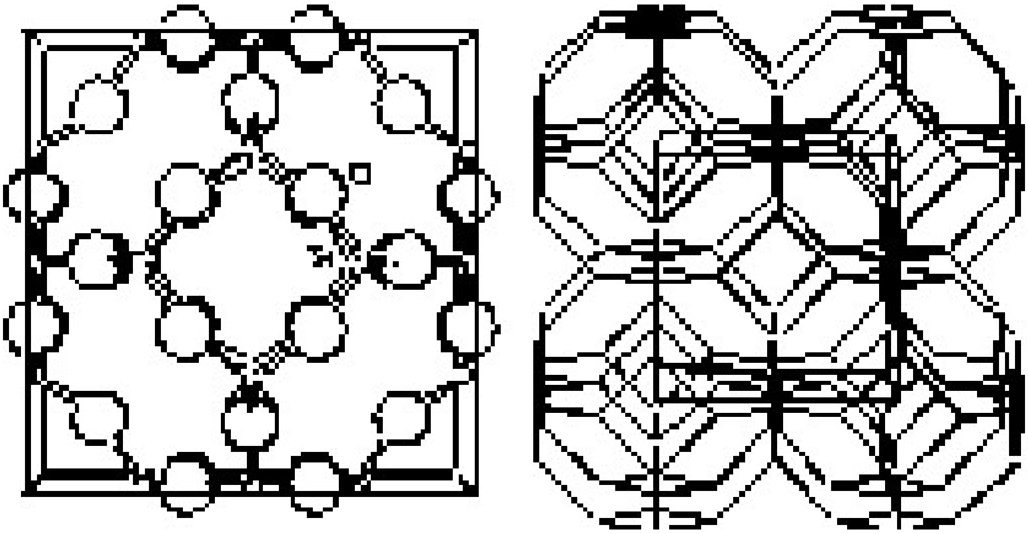


Figure 21: Single sodalite cage in unit cell and body-centred cubic packed sodalite lattice, space group $Im\bar{3}m$.

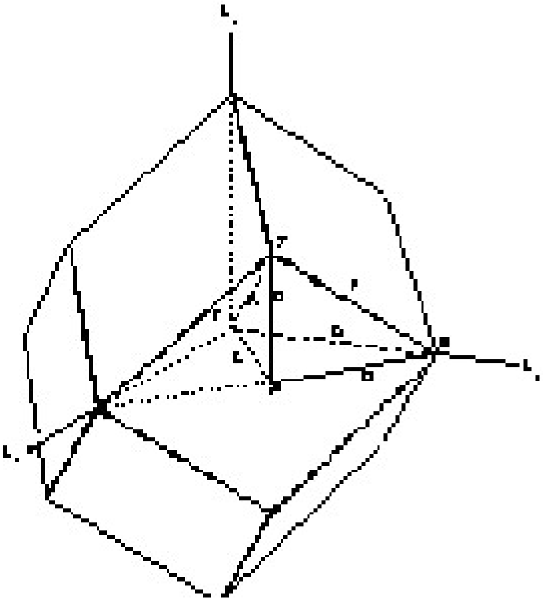


Figure 22: *Brillouin* zone of the bcc lattice. The irreducible wedge is bound by the labelled symmetry lines.

Band structure calculation[19]

The band structure calculation input file uses the conventional bcc unit cell with the highest point group symmetry O_h. Therefore a backfolding of the bands results.

Input file: `sodal_ba.ehi`

Properties calculation[19]

The properties calculation uses the primitive bcc unit cell. The formula or the number of atoms per unit cell reduces to $^{3}_{\infty}[Si_6O_{12}]$.

Input file: `sodal_ds.ehi`, `sodal_ds.epi`

You may investigate the wavefunctions leading to the sharp peak in the Si-3*s*/O-2*s* region, caused by so-called superdegenerate levels.[51] Note also the strong Si-O overlap although the levels are degenerate.

5.7 3D system - Diamond

The band structure of diamond is calculated along the L→ Γ and Γ→X high symmetry lines. The carbon 2s 2p basis set is augmented with 3s functions.[21]

Input file: `diamond.ehi`

6 ACKNOWLEDGEMENTS

This work was supported by grant BEW Nr. 10441/58375 financed by the Bundesamt für Energiewirtschaft and by grant NF 2000-046617.96/1 financed by the Schweizerische Nationalfonds. We would like to thank Dr. Peter Bornhauser and Dominik Brühwiler for their careful reading of the manual and their testing of the various programs and examples.

7 REFERENCES

7.1 General principles and theory

[1] N.W. Ashcroft and N.D. Mermin, *Solid State Physics*, Holt-Saunders, Philadelphia (1976).

[2] M. Lax, *Symmetry Principles in Solid State and Molecular Physics*, Wiley, New York (1974).

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

[4] C.J. Bradley and A.P. Cracknell, *The Mathematical Theory of Symmetry in Solids - Representation Theory for Point Groups and Space Groups*, Clarendon Press, Oxford (1972).

[5] J.F. Cornwell, *Group Theory in Physics*, Vol. I, Academic Press, London (1984).

[6] J.F. Cornwell, *Group Theory and Electronic Energy Bands in Solids*, North-Holland Publishing Company, Amsterdam (1969).

[7] F.A. Cotton, *Chemical Applications of Group Theory*, 3rd edition, John Wiley & Sons, New York (1990).

[8] N.F.M. Henry and K. Lonsdale, *International Tables for X-Ray Crystallography, Vol. I, Symmetry Groups*, Kynoch, Birmingham (1965).

[9] M.-H. Whangbo and R. Hoffmann, J. Am. Chem. Soc. **100** (1978), 6093.

[10] R. Hoffmann, J. Chem. Phys. **39** (1963), 1397.

[11] M. Wolfsberg and L. Helmholz, J. Chem. Phys. **20** (1952), 837.

[12] J.H. Ammeter, H.-B. Bürgi, J.C. Thibeault, and R. Hoffmann, J. Am. Chem. Soc. **100** (1978), 3686.

[13] T. Hughbanks and R. Hoffmann, J. Am. Chem. Soc. **105** (1983), 1150.

[14] J.-Y. Saillard and R. Hoffmann, J. Am. Chem. Soc. **106** (1984), 2006.

[15] M. Kertesz and T. Hughbanks, Phys. Rev. B**24** (1981), 6870.

[16] M. Brändle and G. Calzaferri, Helv. Chim. Acta **67** (1993), 2350.

[17] G. Calzaferri, L. Forss, and I. Kamber, J. Phys. Chem. **93** (1989), 5366.

[18] A. Imamura, J. Chem. Phys. **52** (1970), 3168.

[19] M. Brändle, PhD thesis, University of Berne (1994).

[20] a) G. Del Re, J. Ladik, and G. Biczó, Phys. Rev. **155** (1967), 997.
b) P.-O. Löwdin, J. Molec. Spectroscopy **13** (1964), 326.
c) J. Delhalle, Ann. Soc. Sci. Bruxelles, Ser. III **86** (1972), 227.

[21] G. Calzaferri and R. Rytz, J. Phys. Chem. **100** (1996), 11122.

[22] a) R. Rytz and G. Calzaferri, J. Phys. Chem. B **101** (1997), 5664.
b) R. Rytz, PhD thesis, University of Bern (1997).

7.2 Recommended k-point sets

Recommended:

[23] R. Ramírez and M.C. Böhm, Int. J. Quant. Chem. **30** (1986), 391.

[24] A.H. MacDonald, Phys. Rev. B**18** (1978), 5897.

[25] R. A. Evarestov and P. Smirnov, phys. stat. sol. (b) **119** (1983), 9 (review article).

Further:

[26] D.J. Chadi and M.L. Cohen, Phys. Rev. B**8** (1973), 5747.

[27] D.J. Chadi, Phys. Rev. B**16** (1977), 1746.

[28] H.J. Monkhorst and J.D. Pack, Phys. Rev. B**13** (1976), 5188.

[29] J.D. Pack and H.J. Monkhorst, Phys. Rev. B**16** (1977), 1748.

[30] V.K. Bashenov, M. Bardashova, and A.M. Mutal, phys. stat. sol. (b) **80** (1977), K89.

[31] P.J. Lin-Chung, phys. stat. sol. (b) **85** (1978), 743.

[32] R.A. Evarestov, M.I. Petrashen, and E.M. Ledovskaya, phys. stat. sol. (b) **68** (1975), 453.

[33] A.V. Bandura and R.A. Evarestov, phys. stat. sol. (b) **64** (1974), 635.

[34] M.L. Zolotarev and A.S. Poplavnoi, Sov. Phys. J. **26** (1983), 78.

7.3 Programs

[35] a) E. Keller, *SCHAKAL 88, A FORTRAN Program for the Graphic Representation of Molecular and Crystallographic Models*, Kristallographisches Institut der Albert-Ludwigs-Universität Freiburg / FRG (1988).
b) E. Keller, J. Appl. Cryst. **22** (1989), 19.

[36] E. Zakrajsek and J. Zupan, Ann. Soc. Sci. Bruxelles, Ser. II **89** (1975), 337.

[37] R. Hoffmann group, *Manuals for extended-Hückel calculations, HP Version*, Cornell University, Ithaca, N.Y. (1989).

[38] M.-H. Whangbo, M. Evain, T. Hughbanks, M. Kertesz, S. Wijeyesekera, C. Wilker, C. Zheng and R. Hoffmann, QCPE #571, *Extended-Hückel Molecular, Crystal and Properties Package*, Quantum Chemical Program Exchange (1989).

[39] J.D. Dunitz, *X-Ray Analysis and the Structure of Organic Molecules*, Cornell Univ., (1979).

[40] T.J. Pearson, *PGPLOT graphics subroutine library*, California Institute of Technology (1997), `http://astro.caltech.edu/~tjp/pgplot/`

[41] *NAG Graphics Library*, The Numerical Algorithms Group, NAG Ltd., Oxford.

[42] GKS, *Graphical Kernel System*, ISO 7942.

[43] U. Höweler, *MOBY 1.5, Molecular Modelling on the PC*, Springer Verlag, `http://www.springer-ny.com/catalog/emedia/moby/moby.html`

[44] a) G. Calzaferri, R. Rytz, and M. Brändle, *ICON-EDiT, Extended Hückel Molecular Orbital and Transition Dipole Moment Calculations*, (1997), available via `http://iacrs1.unibe.ch/`
b) G. Calzaferri and M. Brändle, *ICONC & INPUTC, QCPE-program QCMP #116*, QCPE Bulletin **12** (1992).

7.4 Examples

[45] S. Sung and R. Hoffmann, J. Am. Chem. Soc. **107** (1985), 578.

[46] R. Hoffmann, C. Janiak, and C. Kollmar, Macromolecules **24** (1991), 3725.

[47] M. Brändle, G. Calzaferri, and M. Lanz, Chem. Phys. **201** (1995), 141.

[48] M. Kertesz and R. Hoffmann, J. Am. Chem. Soc. **106** (1984), 3453.

[49] R.G. Dickinson and L. Pauling, J. Am. Chem. Soc. **45** (1923), 1466.

[50] J.W. Richardson, Jr., J.J. Pluth, J.V. Smith, W.J. Dytrych, and D.M. Bibby, J. Phys. Chem. **92** (1988), 243.

[51] T. Hughbanks, J. Am. Chem. Soc. **107** (1985), 6851.

[52] M Brändle and G. Calzaferri, Res. Chem. Intermed. **20** (1994), 783.

7.5 Parameter sets

[53] S.P. McGlynn, L.G. Vanquickenborne, M. Kinoshita, and D.G. Carroll, *Introduction to Applied Quantum Chemistry*, Holt, Rinehart and Winston, Inc., New York, (1972).

[54] C.J. Ballhausen and H.B. Gray, *Molecular Orbital Theory*, W.A. Benjamin Inc., New York, (1964).

[55] H. Basch, A. Viste, and H.B. Gray, Theor. Chim. Acta **3** (1965),458.

[56] H. Basch and H.B. Gray, Theor. Chim. Acta **4** (1967), 367.

[57] V.I. Baranovskii and A.B. Nikol'skii, Teoreticheskaya I Experimental'naya Khimiya **3** (1967), 527.

[58] S. Alvarez, *Tables of Parameters for extended-Hückel Calculations*, Universitat de Barcelona (1989).

[59] E. Clementi and C. Roetti, At. Data Nucl. Tables **14** (1974), 177.

[60] E. Clementi, C.L. Raimondi, and W.P. Reinhardt, J. Phys. Chem. **47** (1967), 1300.

[61] J.W. Richardson, W.C. Nieuwpoort, R.R. Powell, and W.F. Edgell, J. Chem. Phys. **36** (1962), 1057.

[62] G. Burns, J. Phys. Chem. **41** (1964), 1521.

[63] M.A. Whitehead, in *Semi-Empirical All-Valence-Electron SCF-MO Theory in* Sigma *MO Theory*, O. Sinanoglu and K. Wilberg, eds., Yale Press, (1970), 50.

[64] D.A. Brown and A. Owens, Inorg. Chim. Acta **5** (1971), 675.

[65] N. Bessis and G. Bessis, J. Chem. Phys. **74** (1981), 3628.

[66] J.C. Slater, Phys. Rev. **36** (1930), 51.

[67] S. Larsson and P. Pyykkö, Chem. Phys. **101** (1986), 355.

[68] R.G. Edgell, M. Hotokka, L. Laaksonen, P. Pyykkö, and J.G. Snijders, Chem. Phys. **72** (1982), 237.

[69] A. Viste, M. Hotokka, L. Laaksonen, and P. Pyykkö, Chem. Phys. **72** (1982), 225.

[70] P. Pyykkö, L. Laaksonen, J. Phys. Chem. **88** (1984), 4892.

[71] N.J. Fitzpatrick and G.H. Murphy, Inorg. Chim. Acta **87** (1984), 41.

[72] N.J. Fitzpatrick and G.H. Murphy, Inorg. Chim. Acta **111** (1986), 139.

[73] G. Calzaferri, Chimia **40** (1986), 74, 435.

[74] G. Calzaferri and R. Hoffmann, J. Chem. Soc. Dalton Trans. (1991), 917.

[75] M. Brändle and G. Calzaferri, Helv. Chim. Acta **76** (1993), 924.

[76] A.B. Anderson and R. Hoffmann, J. Chem. Phys. **60** (1974), 4271.

[77] J.V. Ortiz and R. Hoffmann, Inorg. Chem. **24** (1985), 2095.

[78] a) G. Calzaferri and L. Forss, Helv. Chim. Acta **69** (1986), 873.
b) G. Calzaferri and L. Forss, Helv. Chim. Acta **70** (1987), 465.

[79] D.J. Underwood, M. Novak, and R. Hoffmann, J. Am. Chem. Soc. **106** (1984), 2837.

[80] E. Canadell and O. Eisenstein, Inorg. Chem. **22** (1983), 3856.

[81] S. Komiya, T.A. Albright, R. Hoffmann, and J.K. Kochi, J. Am. Chem. Soc. **99** (1977), 8440.

[82] L.L. Lohr and P. Pyykkö, Chem. Phys. Lett. **62** (1979), 333.

[83] S. Alvarez, F. Mota, and J. Novoa, J. Am. Chem. Soc. **109** (1987), 6586.

[84] C. Zheng and R. Hoffmann, J. Am. Chem. Soc. **108** (1986), 3078.

[85] R.H. Summerville and R. Hoffmann, J. Am. Chem. Soc. **98** (1976), 7240.

[86] G. Calzaferri and F. Felix, Helv. Chim. Acta **60** (1977), 730.

[87] P.J. Hay, J.C. Thibeault, and R. Hoffmann, J. Am Chem. Soc. **97** (1975), 4884.

[88] E. Canadell, O. Eisenstein, and J. Rubio, Organometallics **3** (1984), 759.

[89] D.L. Thorn and R. Hoffmann, Inorg. Chem. **17** (1978), 126.

[90] W.A. Harrison, in *Electronic Structure and the Properties of Solids*, W. Freeman and Co., San Francisco (1980), 551.

[91] D.J. Underwood, R. Hoffmann, K. Tarsumi, A. Nakamura, and Y. Yamamoto, J. Am. Chem. Soc. **107** (1985), 5968.

[92] E. Canadell and O. Eisenstein, Inorg. Chem. **22** (1983), 2398.

[93] D.L. Dubois and R. Hoffmann, Nouv. J. Chim. **1** (1977), 479.

[94] C. Zheng and R. Hoffmann, Inorg. Chem. **28** (1989), 1074.

[95] E.L. Garfunkel and C. Minot, J. Solid State Chem. **65** (1986), 72.

[96] C. Zheng and R. Hoffmann, Z. Naturforsch. **41b** (1986), 292.

[97] J. W. Lauher, M. Elian, R.H. Summerville, and R. Hoffmann, J. Am. Chem. Soc. **98** (1976), 3219.

[98] E. Canadell, O. Eisenstein, and T. Hughbanks, Inorg. Chem. **23** (1984), 2435.

[99] K. Tatsumi, R. Hoffmann, A. Yamamoto, and J.K. Stille, Bull. Chem. Soc. Jpn. **54** (1981), 1857.

[100] A. Dedieu, T.A. Albright, and R. Hoffmann, J. Am. Chem. Soc. **101** (1979), 3141.

[101] R. Hoffmann, C. Minot, and H.B. Gray, J. Am. Chem. Soc. **106** (1984), 2001.

[102] K. Tatsumi and R. Hoffmann, J. Am. Chem. Soc. **103** (1981), 3340.

[103] T. Hughbanks, R. Hoffmann, M.-H. Whangbo, K.R. Stewart, O. Eisenstein, and E.Canadell, J. Am. Chem. Soc. **104** (1982), 3876.

[104] R. Hoffmann, S. Shaik, J.C. Scott, M.-H. Whangbo, and M.J. Foshee, J. Solid State Chem. **34** (1980), 263.

[105] J. Hinze and H.H. Jaffé, J. Chem. Phys. **67** (1963), 1501.

[106] M.-H. Whangbo, W.M. Walsh, R.C. Haddon, and F. Wudl, Solid State Comm. **43** (1982), 673.

[107] T. Anh, M. Elian, and R. Hoffmann, J. Am. Chem. Soc. **100** (1978), 110.

[108] D.M. Hoffmann, R. Hoffmann, and R.R. Fiesel, J. Am. Chem. Soc. **104** (1982), 3858.

[109] R.A. Wheeler and R. Hoffmann, Angew. Chem., Int. Ed. Engl. **25** (1986), 822.

[110] M.-H. Whangbo and P. Gressier, Inorg. Chem. **23** (1984), 1228.

[111] K. Tatsumi and A. Nakamura, J. Am. Chem. Soc. **109** (1987), 3195.

[112] P. Kubácek, R. Hoffmann, and Z. Havlas, Organometallics **1** (1982), 180.

[113] W. Tremel, R. Hoffmann, J. Am. Chem. Soc. **109** (1987), 124.

[114] K. Tatsumi, A. Nakamura, P. Hofmann, P. Stauffert, and R. Hoffmann, J. Am. Chem. Soc. **107** (1985), 4440.

[115] J. Silvestre and T.A. Albright, Isr. J. Chem. **23** (1983), 139.

[116] E. Amouyal, M. Mouallem-Bahout, and G. Calzaferri, J. Phys. Chem. **95** (1991), 7641.

[117] G. Calzaferri and R. Rytz, J. Phys. Chem. **99** (1995), 12141.

8 APPENDIX

8.1 Subroutine connectivity diagrams of programs

If a routine occurs multiple times, its subroutines are only listed at the first occurrence.
># : Fortran-UNIT written by this subroutine
<# : Fortran-UNIT read by this subroutine
<>#: Fortran-UNIT read/written by this subroutine

8.1.1 EHMACC

```
1 EHMACC- - -
   |
2 PART1---------------------------------------------------- - - -
   |                    |             |            |      |
3 GETSEL--------+      GETLAT-+      GETPOS-+     FTOCAR BUILDZ-+
   |      |      |      |      |      |      |     |      |      |
4 READL WRITEL WRTKEY READL WRITEL WRITEL READL MATMUL VEC    VPROD
                                                                 |
5                                                                WRITEL

2 -PART1- - -
   |
3 INMODI-------------------------------------------------+
   |>1    |      |      |      |     |      |     |      |
4 ADDK   WRITEL GETNAM GETNUM READL READA DISTC PRBOND RBOND
                        |      >12   |
5                       SUPPR        WRITEL

2 -PART1------------------------------------------------------------------------------- - - -
   |             |                                        |                           |
3 GETVAR-+      PARAM------------------------------+     AOOUT-----------------+     SHOUT-----------------+
   |      |      |      |      |             |      |     |      |             |      |      |             |
4 WRITEL READL WRITEL READL SEARCH-+      GAOOCC GETKD ADDK   SEARCH-+      READL ADDK   SEARCH-+      READL
                               |      |                          |      |                    |      |
5                              GETORB GETNAM                     GETORB GETNAM               GETORB GETNAM

2 -PART1---------------------------------------------------------------------------------------------------+
   |                                  |                            |                  |                    |
3 GETFMO-----------------------+      GETSYM----------------+     GETKP---------+    VARYGM---------+     CLEXIT
   |      |       |            |      |     |     |       |        |      |    |      |      |      |
4 ADDK   WRITEL SEARCH-+      READL READL READA WRITEL GETPER WRITEL READL ADDK FTOCAR BUILDZ INMODI
                  |      |                                  |
5                 GETORB GETNAM                             MATMUL

1 -EHMACC
   |
2 PART2----------------------------------------------------------+
   |                                                |             |
3 ARRAYS----------------------+                    ARRAYD- - -   VARYGM
   |      |                    |                    |
4 WRITEL SMAT---------+      HMAT----------+      CRYST- - -
          |>21   |      |      |<>21 |       |      |>1
          |      |      |      |<>22 |       |      |<>23
          |      |      |      |<>23 |       |      |>20
          |      |      |      |     |       |      |
5         MOVLAP OMATT2 UNIQUE OMATT2 TEMPO UNIQUE EHFMO-------------------------------------------------+
          |             >19            |            |<>11                 |      |      |      |      |      |
6         MOV---+                     OMATT2       CBORIS--------+      ADDK   WRITEL OMATT3 MATPAC OUTMAT OMFRAG
          |     |                                   |      |      |
7         ABFNS LOVLAP                              CCHOL CTRED2 CTQL2

3 -ARRAYD- - -
   |
4 -CRYST------------------------------------------------------------------------------------------------------------+
   |      |      |                    |      |      |      |      |                    |                    |       |
5 BSMAT BHMAT CBORIS---------+     SYMMAN GRAD   OMATBO OMATBP BFMO----------+      BPOP-----------+     REPUL ETOT
   >13    <22    |       |      |            |<23                 |<>7   |      |      |<>9   |      |      |      <20
   <21           |       |      |            |<13                 |<>11  |      |      |>2    |      |      |     >15
                 |       |      |            |>14                 |<23   |      |      |>3    |      |      |
                 |       |      |            |                    |      |      |      |<19   |      |      |
                 |       |      |            |                    |      |      |      |<21   |      |      |
                 |       |      |            |                    |      |      |      |<23   |      |      |
                 |       |      |            |                    |      |      |      |      |      |      |
6                CCHOL CTRED2 CTQL2        MU               OMF1   OMFRAG OUTMAT OMATT MOPOP1 POBOND REPUL1
                                             <22                                                            |
                                             <23                                                            |
                                                                                                            |
7                                                                                                           REPUL2
```

```
3 -ARRAYD
   |
```

```
4 HUCKEL--------------------------------------------+
   ¦<21                 ¦      ¦      ¦      ¦      ¦
   ¦<22                 ¦      ¦      ¦      ¦      ¦
   ¦>11                 ¦      ¦      ¦      ¦      ¦
   ¦                    ¦      ¦      ¦      ¦      ¦
5 CBORIS---------+     WRITEL REPUL OMATT3 POPULA EHFMO--------------------------------------------------+
   ¦      ¦      ¦             ¦             ¦<21   ¦<>7                  ¦      ¦      ¦      ¦      ¦      ¦
   ¦      ¦      ¦             ¦             ¦      ¦<>11                 ¦      ¦      ¦      ¦      ¦      ¦
   ¦      ¦      ¦             ¦             ¦      ¦<>21                 ¦      ¦      ¦      ¦      ¦      ¦
   ¦      ¦      ¦             ¦             ¦      ¦<>22                 ¦      ¦      ¦      ¦      ¦      ¦
6 CCHOL CTRED2 CTQL2         REPUL1        MOPOP CBORIS---------+      ADDK   WRITEL OMATT3 MATPAC OUTMAT OMFRAG
                               ¦                    ¦       ¦      ¦

7                              REPUL2               CCHOL   CTRED2 CTQL2
```

8.1.2 EHPC

```
1 EHPC-----------+
   ¦             ¦
2 GPOPT—+      PARRAY-+
   ¦      ¦      ¦<1    ¦
   ¦      ¦      ¦      ¦
3 READL READA WRITEL PPART-----------------------------------------------------------------------------+- - -
                        ¦>1                               ¦                    ¦                          ¦
                        ¦>7                               ¦                    ¦                          ¦
                        ¦                                 ¦                    ¦                          ¦
4                       GPROJ----------------------+      GPROJF--------+      GPROJB--------------+     INTEGR
                        ¦      ¦      ¦            ¦      ¦      ¦      ¦      ¦<3    ¦      ¦     ¦
                        ¦      ¦      ¦            ¦      ¦      ¦      ¦      ¦      ¦      ¦     ¦
5                       READA WRITEL SEARCH-+     READL READA WRITEL READL WRITEL GETNAM READL READA
                                      ¦     ¦
6                                    GETORB GETNAM


3 -PPART-------------------------------------------+
   ¦      ¦      ¦                    ¦            ¦
4 PROJT PROJTB SMOOTH--------+      SMOOTO-+     SMOOTB-+
   ¦<2    ¦<3    ¦>7    ¦      ¦      ¦<1   ¦      ¦<1    ¦
   ¦<4    ¦      ¦      ¦      ¦      ¦<2   ¦      ¦<3    ¦
   ¦<>8   ¦      ¦      ¦      ¦      ¦<4   ¦      ¦>7    ¦
   ¦      ¦      ¦      ¦      ¦      ¦>7   ¦      ¦      ¦
   ¦      ¦      ¦      ¦      ¦      ¦     ¦      ¦      ¦

5 WRITEL WRITEL FDST   WRITEL OUTPUT WRITEL OUTPUT WRITEL OUTPUT
```

8.1.3 DOSPLOT

```
1 DOSPLT------------------------------------+
   ¦<1    ¦      ¦      ¦      ¦            ¦
   ¦<>7   ¦      ¦      ¦      ¦            ¦
   ¦      ¦      ¦      ¦      ¦            ¦
2 NAGGKS DPAPND DPPLOT DPOLAY DPGRPH- - - DPRDRW-------------------------------+
                                            ¦      ¦      ¦      ¦      ¦      ¦
3                                           DPPLOT DPBRDR DPAXES DPTITL DPTEXT DPFRMI
                                             <1            ¦      ¦      ¦      ¦
4                                                         DPTICA DPTRAN DPTRAN DPTRAN
                                                                  ¦      ¦      ¦
5                                                                DPTXEX DPTXEX DPTXEX


2 -DPGRPH----------------------------------------------------------------+
   ¦      ¦      ¦      ¦      ¦      ¦      ¦      ¦      ¦      ¦      ¦
3 DPPLOT DPBRDR DPAXES DPTITL DPFRMI DPTICS DPTICA DPTRAN DPSELC DPLIST DPRDRW
                                                    ¦

4                                                   DPTXEX
```

8.1.4 BANDPLOT

```
1 BANDPL--------------------------------------------------------------------------------------------+
   ¦<1    ¦      ¦      ¦      ¦      ¦      ¦      ¦      ¦                                 ¦      ¦
   ¦<2    ¦      ¦      ¦      ¦      ¦      ¦      ¦      ¦                                 ¦      ¦
   ¦<3    ¦      ¦      ¦      ¦      ¦      ¦      ¦      ¦                                 ¦      ¦
   ¦>4    ¦      ¦      ¦      ¦      ¦      ¦      ¦      ¦                                 ¦      ¦
   ¦<14   ¦      ¦      ¦      ¦      ¦      ¦      ¦      ¦                                 ¦      ¦
   ¦      ¦      ¦      ¦      ¦      ¦      ¦      ¦      ¦                                 ¦      ¦
2 NAGGKS BPAXES BPBSRT BPSRT2 BPRAW BPENER BPSYML BPSELC BFIT----------+                   BPBAND BPGRPH - -
                                                           ¦<9    ¦      ¦                    <2    ¦
                                                           ¦>11   ¦      ¦                          ¦
                                                           ¦<12   ¦      ¦                          ¦
                                                           ¦      ¦      ¦                          ¦
3                                                          TSORT EREAD DOS-----------+            BPENER
                                                            <9     <10   ¦>11   ¦      ¦
                                                            >12          ¦      ¦      ¦
4                                                                        A      DA     INV
                                                                                ¦
5                                                                               DINV


2 -BPGRPH--------------------------------------------------+
   ¦<>15 ¦      ¦      ¦      ¦      ¦      ¦      ¦      ¦
   ¦      ¦      ¦      ¦      ¦      ¦      ¦      ¦      ¦
3 BPAXES BPBAND BPBRDR BPTICS BPTICA BPTRAN BPFRMI BPSYML BPRDRW-----------------------------+
                                      ¦                    ¦      ¦      ¦      ¦      ¦      ¦
```

```
4                                     BPTXEX               BPAXES BPBAND BPBRDR BPTICA BPTRAN BPFRMI
```

8.1.5 DOSPRT

```
1 DOSPRT
  ¦<1
  ¦<2
  ¦
2 DPRDRW------------------------------------+
  ¦      ¦      ¦             ¦      ¦      ¦
3 DPPLOT DPBRDR DPAXES-+      DPTITL DPTEXT DPFRMI-+
  ¦<1    ¦      ¦      ¦      ¦      ¦      ¦      ¦
  ¦      ¦      ¦      ¦      ¦      ¦      ¦      ¦
4 DPLWID DPLWID DPTICA DPLWID DPRTRA DPRTRA DPRTRA DPLWID
                              ¦      ¦      ¦

5                             DPRTXX DPRTXX DPRTXX
```

8.1.6 BANDPRT

```
1 BANDPRT
  ¦<2
  ¦
2 BPRDRW-----------------------------+
  ¦      ¦      ¦      ¦      ¦      ¦
3 BPAXES BPBAND BPBRDR BPTICA BPRTRA BPFRMI-+
  ¦       <1    ¦             ¦      ¦      ¦
4 BPLWID        BPLWID        BPRTXX BPRTRA BPLWID
                                     ¦

5                                    BPRTXX
```

8.1.7 TRANSC

```
1 TRANSC--------+
  ¦>2    ¦      ¦
  ¦      ¦      ¦
2 READL READA MATMUL

   <1     <1
```

8.2 Old version - Chapter 2.1 Installation and system requirements

8.2.1 AIX

The UNIX network queuing system NQS is recommend in order to perform batch jobs. NQS is public domain and can be obtained by anonymous ftp on various server sites, e.g. on `iacrs1.unibe.ch`. For correct operation of the graphic programs, you have to install the PHIGS/GKS-CO[39],[40] run time libraries.

The package is delivered as `tar`ed file on tape. If you want to provide the programs for every user on your system, login as `root`. Change to the directory where you want to install the package.

Type

```
tar -xvf tapedevice
```

If the programs are provided globally, copy all files in `qcpe/exe/aix/.` to `/usr/local/bin/.`

Adapt the NQS queue-types in the `ehmacc` and `ehpc` shell scripts to your conventions.

If you use the programs for your own, create a `~/bin/` directory and copy the files in `qcpe/exe/aix/.` to there. Insert `~/bin` into your PATH (see `.profile (ksh)` or `.login (csh)`).

If you don't have NQS at hand, you have to edit the `ehmacc` and `ehpc` shell scripts and to remove the `qsub -q $qtype` statements.

8.2.2 MS-DOS/MS-Windows

At least 8MBytes of memory is required for correct operation. Executables for MS-DOS and MS-Windows are supplied. The MS-DOS programs come along with a 32bit Pharlap memory extender provided by Microsoft FORTRAN Powerstation. If you recompile them with Microsoft FORTRAN V5.1, you have to purchase the Pharlap extender for correct operation.
EMATEK GmbH in Cologne, Germany has been so kind to provide a MS-Windows interface for GKS for correct operation of the programs BANDPLOT and DOSPLOT.
The package is delivered as zipped file on floppy diskettes. Change to the drive and directory where you want to install the package.
Type

```
a:pkunzip -d a:qcpe_new.zip your-drive-letter:\
```

Copy all files in the directory `qcpe_new\exe\msdos` to the directory `qcpe_new`.
If your printer is not attached to port `LPT1:` but instead prints on file, change entry #2 in `\qcpe_new\KERNEL.SYS` from `LPT1:` to `FILE:`.

8.2.3 VMS

In order to run the graphic programs BANDPLOT and DOSPLOT properly, you have to install the GKS run time library. The package is delivered as `BACKUP`-saveset on TK50 tape. If you want to provide the programs for every user on your system, login as `SYSTEM`. Change to the drive and directory where you want to install the package.
Type

```
BACKUP tapedevice:qcpe_new.bck/sav/label=band *.*
```

Executables for VAXes and DEC-Alpha machines running OPEN-VMS are supplied, see `[.qcpe_new.exe.vms]READ.ME` file.
If the programs are provided globally, insert `[.qcpe_new.exe.vms]login.ins` in your `sys$manager:sylogin.com`. Edit the paths in the inserted text. Notify your users to insert into their `login.com` a symbol name `BANDWORK`, which points to their directory where their input- and output files shall be stored.
If you use the programs for your own, insert `[.qcpe_new.exe.vms]login.ins` in your `login.com`. Edit the paths in the inserted text. Define a symbol `BANDWORK`, which points to your directory where the input- and output files shall be stored.
Logout and login again to actualize your changes.

8.3 Old version - Chapter 2.2 Example band structure calculation - Square carbon lattice

2D tetragonal system: A square net of carbon atoms with lattice vector of length a. Refer to [3], p. 16ff .

8.3.1 AIX

Copy `c_gitter.*` in the `qcpe_new/examples` directory to your local working directory. You may examine the input file `c_gitter.ehi` for the band structure calculation. For the meanings of the keywords refer to chapter 4.1.

Type

```
ehmacc c_gitter
```

After the calculation has finished, you obtain `c_gitter.eh1`, `c_gitter.ehc`, `c_gitter.ehe`, `c_gitter.eho`. Examine the output file `c_gitter.eho`.

An interpolated band structure has been provided by the files `c_gitter.bft` and `c_gitter.bin`. To visualize the band structure type

```
bandplot c_gitter
```

Select then option (1). Numbers for energies and symmetry line widths require a decimal point. Provide an energy range of `-25.0` eV to `-5.0` eV. The band structure will be drawn, and you enter a graph modification menu. Select option (12). For further explanation of the program refer to chapter 4.4.

8.3.2 MS-DOS

Copy `c_gitter.*` in the `qcpe_new\examples` directory to the working directory `qcpe_new`. Change into this directory.
You may examine the input file `c_gitter.ehi` for the band structure calculation. For the meanings of the keywords refer to chapter 4.1.
Type

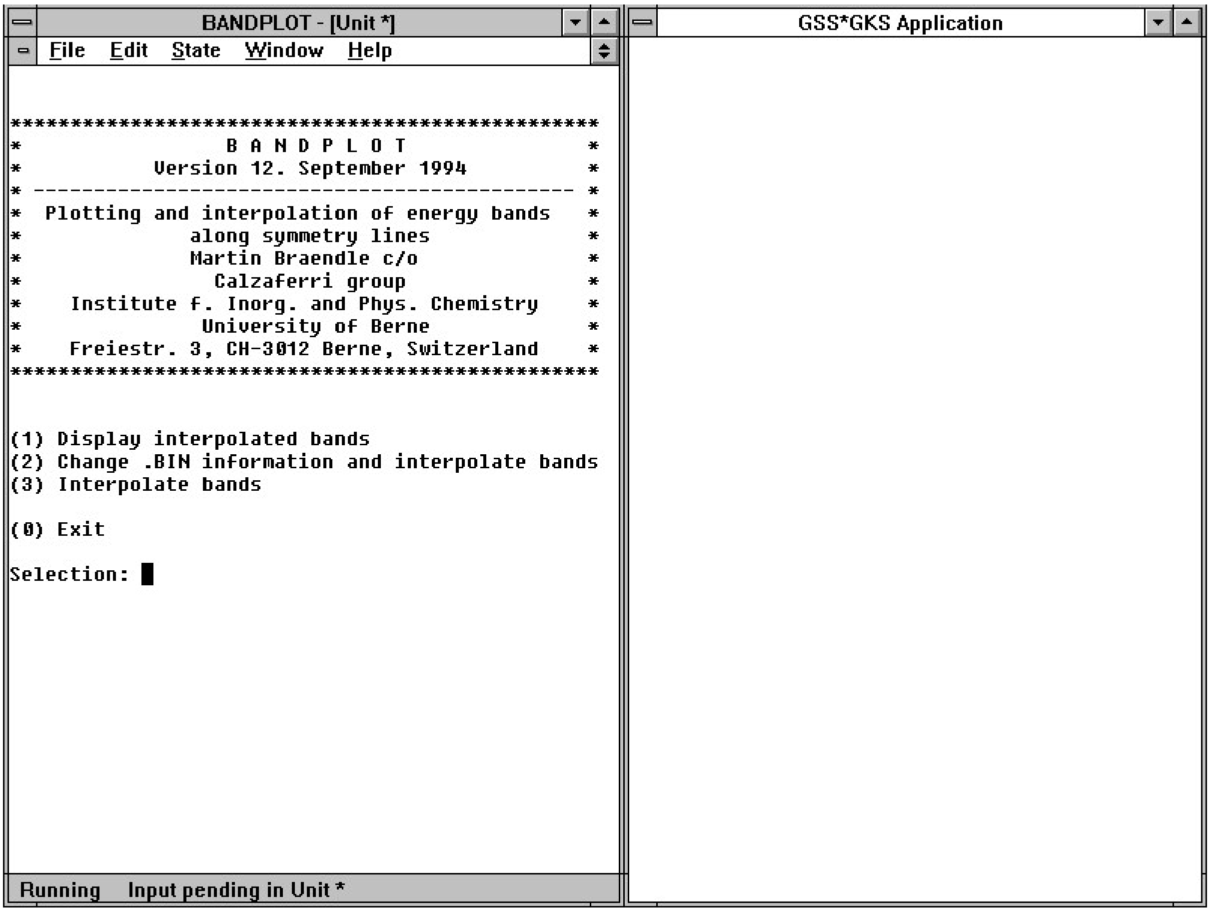
```
ehmacc.bat c_gitter
```

After the calculation has finished, you obtain `c_gitter.eh1`, `c_gitter.ehc`, `c_gitter.ehe`, `c_gitter.eho`. Examine the output file `c_gitter.eho`.

An interpolated band structure has been provided by the files `c_gitter.bft` and `c_gitter.bin`. To visualize this band structure start MS-Windows. Start FileManager. Change into the working directory `qcpe_new`. Select Run… in the File menu. Enter:

```
bandplts.bat c_gitter
```

MS-Windows executes `bandplts.bat` by changing into MS-DOS. It copies the required files to their proper filenames needed by BANDPLOT. After that, double click on `bandplot.exe`. A full screen graphical window called `GSS*GKS Application` will be opened. Scale it back to smaller size. Position the terminal window and the graphical window like this:



```
*************************************************
*                  B A N D P L O T              *
*            Version 12. September 1994         *
* --------------------------------------------- *
*  Plotting and interpolation of energy bands   *
*              along symmetry lines             *
*              Martin Braendle c/o              *
*                Calzaferri group               *
*    Institute f. Inorg. and Phys. Chemistry    *
*               University of Berne             *
*    Freiestr. 3, CH-3012 Berne, Switzerland    *
*************************************************


(1) Display interpolated bands
(2) Change .BIN information and interpolate bands
(3) Interpolate bands

(0) Exit

Selection:
```

Figure 1: Startup screen of BANDPLOT (MS-Windows version).

Select then option (1) in the terminal window. Numbers for energies and symmetry line widths require a decimal point. Provide an energy range of `-25.0` eV to `-5.0` eV. The band structure will be drawn, and you enter a graph modification menu. Select option (12). For further explanation of the program refer to chapter 4.4.

8.3.3 VMS

Copy `c_gitter.*` in the `[.qcpe_new.examples]` directory to your local working directory defined by the symbol name BANDWORK.

You may examine the input file `c_gitter.ehi` for the band structure calculation. For the meanings of the keywords refer to chapter 4.1.

Type

```
ehmacc c_gitter
```

After the calculation has finished, you obtain `c_gitter.eh1`, `c_gitter.ehc`, `c_gitter.ehe`, `c_gitter.eho`. Examine the output file `c_gitter.eho`.

An interpolated band structure has been provided by the files `c_gitter.bft` and `c_gitter.bin`. To visualize the band structure type

```
bandplot c_gitter
```

An icon named „GKS” appears. Deiconify to get the graph window. Select then option (1) in the terminal window. Numbers for energies and symmetry line widths require a decimal point. Provide an energy range of `-25.0` eV to `-5.0` eV. The band structure will be drawn, and you enter a graph modification menu. Select option (12). For further explanation of the program refer to chapter 4.4.

8.4 Old version- Chapter 2.3: Example DOS calculation - Square carbon lattice

The DOS of the square carbon lattice, its AO projections and the COOP for the C-C bonds are calculated over a mesh of 15 equally dispersed **k**-points.

Figure 2: Properties **k**-point set for square BZ

8.4.1 AIX

Copy `c_gdos.*` in the `qcpe_new/examples` directory to your local working directory. You may examine the input file `c_gdos.ehi` for the band structure calculation. For the meanings of the keywords refer to chapter 4.1.
Type

```
ehmacc c_gdos
```

After the calculation has finished, you obtain `c_gdos.eh1`, `c_gdos.eh2`, `c_gdos.eh3`, `c_gdos.ehc`, and `c_gdos.eho`. Examine the output file `c_gdos.eho`.
For the integration and smoothing of the DOS and COOP, EHPC has to be called. Inspect first the input file `c_gdos.epi`. For the meanings of the keywords refer to chapter 4.2.
Type

```
ehpc c_gdos
```

After the calculation has finished, you obtain the new files `c_gdos.eh5` and `c_gdos.epo`. Examine the output file `c_gdos.epo`.

To visualize the DOS, the DOS projections and the COOP, type

```
dosplot c_gdos
```

Try out the several menu entries. The program DOSPLOT is described in chapter 4.3.

8.4.2 MS-DOS

Copy `c_gdos.*` in the `qcpe_new\examples` directory to your working directory `qcpe_new`.
You may examine the input file `c_gdos.ehi` for the band structure calculation. For the meanings of the keywords refer to chapter 4.1.
Type

```
ehmacc.bat c_gdos
```

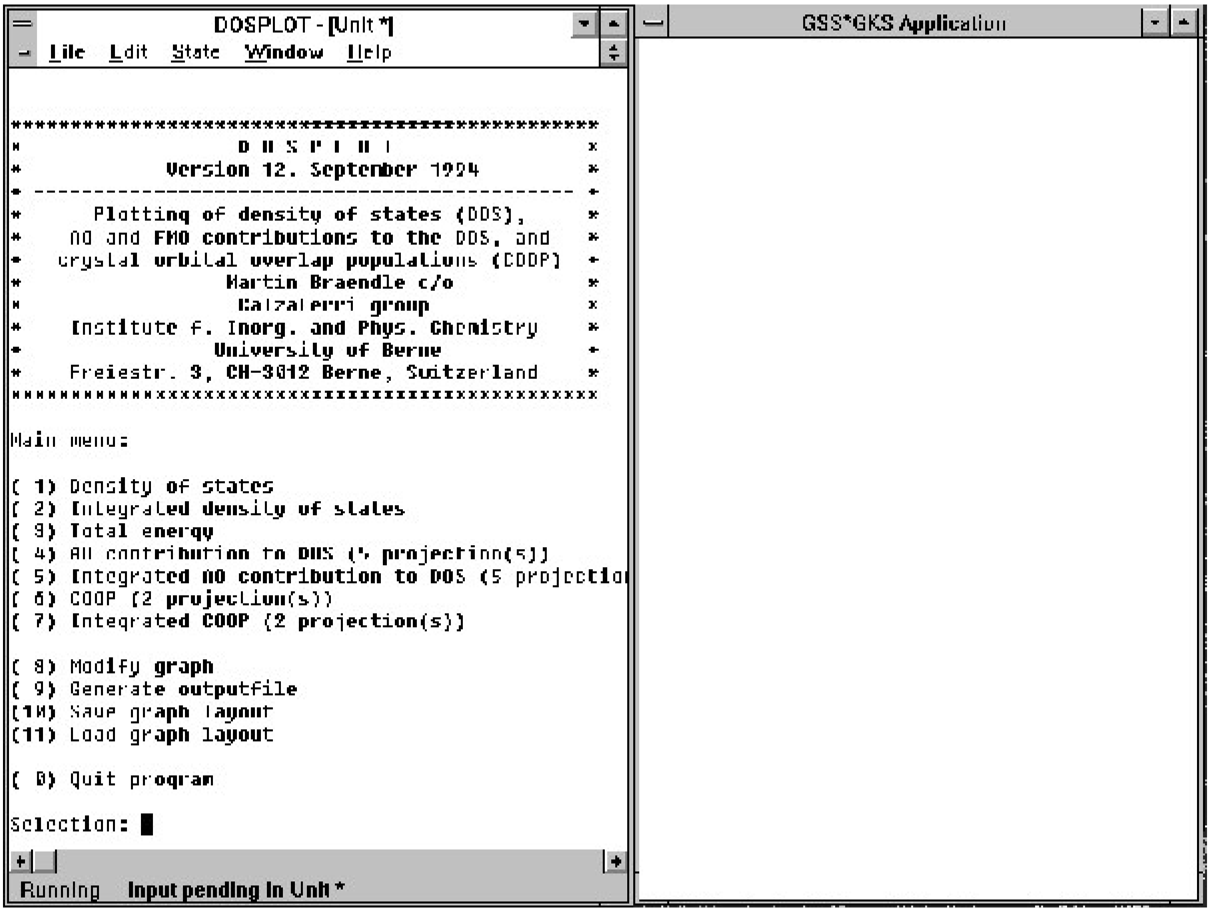
After the calculation has finished, you obtain `c_gdos.eh1`, `c_gdos.eh2`, `c_gdos.eh3`, `c_gdos.ehc`, and `c_gdos.eho`. Examine the output file `c_gdos.eho`.
For the integration and smoothing of the DOS and COOP, EHPC has to be called. Inspect first the input file `c_gdos.epi`. For the meanings of the keywords refer to chapter 4.2.
Type

```
ehpc.bat c_gdos
```

After the calculation has finished, you obtain the new files `c_gdos.eh5` and `c_gdos.epo`. Examine the output file `c_gdos.epo`.
To visualize the DOS start MS-Windows. Start FileManager. Change into the working directory `qcpe_new`. Select Run… in the File menu. Enter:

```
dosplots.bat c_gdos
```

MS-Windows executes `dosplots.bat` by changing into MS-DOS. It copies the required files to the proper file names needed by DOSPLOT. After that, double click on `dosplot.exe`. A full screen graphical window called `GSS*GKS Application` will be opened. Scale it back to smaller size. Position the terminal window and the graphical window like this:



```
**************************************************
*                  D O S P L O T                 *
*          Version 12. September 1994            *
* ---------------------------------------------- *
*     Plotting of density of states (DOS),       *
*    AO and FMO contributions to the DOS, and    *
*   crystal orbital overlap populations (COOP)   *
*               Martin Braendle c/o              *
*                Calzaferri group                *
*    Institute f. Inorg. and Phys. Chemistry     *
*              University of Berne               *
*    Freiestr. 3, CH-3012 Berne, Switzerland     *
**************************************************

Main menu:

( 1) Density of states
( 2) Integrated density of states
( 3) Total energy
( 4) AO contribution to DOS (5 projection(s))
( 5) Integrated AO contribution to DOS (5 projectio
( 6) COOP (2 projection(s))
( 7) Integrated COOP (2 projection(s))

( 8) Modify graph
( 9) Generate outputfile
(10) Save graph layout
(11) Load graph layout

( 0) Quit program

Selection:
```

Figure 3: Startup screen of DOSPLOT (MS-Windows version).

Try out the several menu entries. A description of the program is given in chapter 4.3.

8.4.3 VMS

Copy `c_gdos.*` in the `[.qcpe_new.examples]` directory to your local working directory defined by the symbol-name BANDWORK.
You may examine the input file `c_gdos.ehi` for the band structure calculation. For the meanings of the keywords refer to chapter 4.1.
Type

```
ehmacc c_gdos
```

After the calculation has finished, you obtain `c_gdos.eh1`, `c_gdos.eh2`, `c_gdos.eh3`, `c_gdos.ehc`, and `c_gdos.eho`. Examine the output file `c_gdos.eho`.
For the integration and smoothing of the DOS and COOP, EHPC has to be called. Inspect first the input file `c_gdos.epi`. or the meanings of the keywords refer to chapter 4.2.
Type

```
ehpc c_gdos
```

After the calculation has finished, you obtain the new files `c_gdos.eh5` and `c_gdos.epo`. Examine the output file `c_gdos.epo`.
To visualize the DOS, the DOS projections and the COOP, type

```
dosplot c_gdos
```

An icon named „GKS” appears. Deiconify it in order to get the grapical window. Try out the several menu entries. A description of the program is given in chapter 4.3.

8.4.4 Old version - Chapter 4.1: EHMACC

VMS version:

Call the program by typing

```
EHMACC inputfilename
```

No extension is needed: An input file with the extension `.ehi` is assumed. It must reside in the directory defined by the symbol name `BANDWORK`. A batch job will be submitted to the operating system (Batchqueue `SYS$BATCH`). The current status of the batch job can be monitored by typing the logfile `filename.log_ehmacc`. It also contains system error messages in case of program interruption.

AIX version:

Call the program by typing

```
ehmacc inputfilename (queue-type)
```

No extension is needed: An input file with the extension `.ehi` is assumed. A subdirectory `filename/` is produced and the input file moved there in order to separate the temporarily generated scratch files from other running batch jobs. Then an NQS-batchjob is started. The resulting output files will be moved back to the calling directory.

MS-DOS version:

Call the EHMACC program by typing

```
ehmacc.bat inputfilename
```

No extension is needed: An input file with the extension `.ehi` is assumed.

8.4.5 Old version - Chapter 4.2: EHPC

VMS version:

Call the program by typing

```
EHPC filename
```

No extension is needed: An inputfile with the extension `.epi` is assumed. A batch job will then be submitted to the operating system (Batchqueue `SYS$BATCH`). The current status of the batch job can be monitored by typing the logfile `filename.log_ehpc`. It also contains system error messages in case of program interruption.

AIX version:

Call the EHPC program by typing

```
ehpc filename (queue-type)
```

No extension is needed: An input file with the extension `.epi` is assumed. A subdirectory with name of the filename is produced and the input file moved there. Then an NQS-batchjob is started. The resulting output files will be moved back to the calling directory.

MS-DOS version:

Call the EHPC program by typing

```
ehpc filename
```

No extension is needed: An input file with the extension `.epi` is assumed.

8.4.6 Old version - Chapter 4.3.4 DOSPLOT Usage

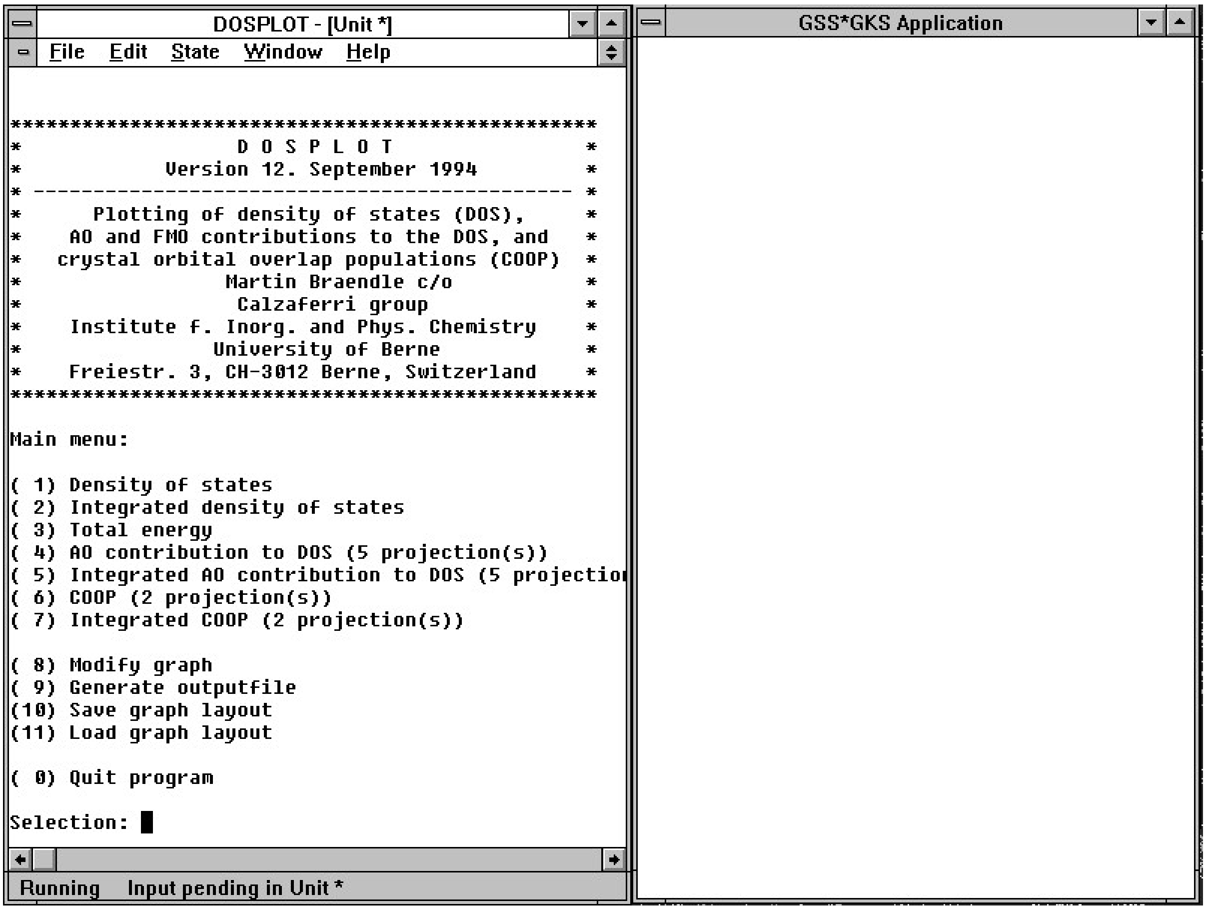
VMS and AIX version:

1. Call the program by typing `dosplot filename [display]`
   No extension is to be supplied, a default extension `.eh5` is assumed. `display` is optional and sets the IP address where the X-window has to be displayed.
2. In the VMS version, an icon called `GKS` appears. Deiconify it and position and size the graph window as you want.

MS-Windows version:

Since Microsoft Fortran requires fixed file names in the sources, two shell scripts `DOSPLOTS.BAT` and `DOSPLOTE.BAT` have been provided, which copy the files to their correct names before and after `DOSPLOT.EXE` is used. You have to perform the following procedure:

1. Call FileManager, change into the directory where `DOSPLOTS.BAT`, `DOSPLOTE.BAT`, `DOSPLOT.EXE`, and your `filename.EH5` (and possibly `filename.DPL`) reside.
2. Select menu `File`, menu-entry `Run`...
3. Type in: `DOSPLOTS.BAT filename`
4. Windows executes `DOSPLOTS.BAT` by changing into MS-DOS, copies the required files to `FORT.#`, then returns back.
5. If `GKSERROR.DAT` exists, delete it.
6. Call `DOSPLOT.EXE` by double-clicking with the mouse on it.
7. A full screen graphical window called `GSS*GKS Application` will be opened. Scale it back to smaller size. Position the terminal window and the graphical window like this:



```
*************************************************
*                  D O S P L O T                *
*            Version 12. September 1994         *
* --------------------------------------------- *
*       Plotting of density of states (DOS),    *
*     AO and FMO contributions to the DOS, and  *
*    crystal orbital overlap populations (COOP) *
*                  Martin Braendle c/o          *
*                   Calzaferri group            *
*     Institute f. Inorg. and Phys. Chemistry   *
*                 University of Berne           *
*     Freiestr. 3, CH-3012 Berne, Switzerland   *
*************************************************

Main menu:

( 1) Density of states
( 2) Integrated density of states
( 3) Total energy
( 4) AO contribution to DOS (5 projection(s))
( 5) Integrated AO contribution to DOS (5 projectio
( 6) COOP (2 projection(s))
( 7) Integrated COOP (2 projection(s))

( 8) Modify graph
( 9) Generate outputfile
(10) Save graph layout
(11) Load graph layout

( 0) Quit program

Selection:
```

Figure 4: Startup screen of DOSPLOT (MS-Windows version).

8. After work, select menu `File`, menu-entry `Run`...
9. Type in: `DOSPLOTE.BAT filename`, then select OK.
10. Windows executes `DOSPLOTE.BAT` by changing into MS-DOS, copies the files `FORT.#` to the required file names, then returns back to File Manager.

8.4.7 Old version - Chapter 4.3.5 Printing

VMS version:

Select `Generate output file` in the main menu.

The fill shade scale factor determines the granularity of the fill shades. If the original size of the output will be kept, set scale factor between 0.3 and 0.5. If the output shall be zoomed down for later publication, set scale factor = 1.0.

MS-Windows version:

Select `Generate output file` in the main menu and then `(1) PostScript output`.

You can select any output device by configuring a printer in the Control Panel of MS-Windows. For correct operation, you have to enter the printer port name (example: `FILE:` or `LPT1:`) in the `KERNEL.SYS` file where your `DOSPLOT.EXE` resides:

```
0:WISS
1:DISPLAY;MOUSE
```

```
2:LPT1:                    change here
3:GMO
```

Listing 3: `KERNEL.SYS` file.

AIX version:
Due to lack of a PostScript driver in the present PHIGS version it is not possible to print directly. Save your graph layout, then exit the program and type

```
dosprt filename
```

A PostScript file `filename.ps` is written. If you want to import it as encapsulated PostScript file into a word processor, remove the end-of-file character (Control-D) at the end of the file, delete all text between the curly brackets `{}` in the `/xyinitsys` definition near the beginning of the file, and adjust the bounding box in the `%%BoundingBox:` statement.

8.4.8 Old version - Chapter 4.3.7 Compiling DOSPLOT

The sources contain subdirectories AIX, MSDOS, VMS with different Makefiles.
In order to compile and link DOSPLOT, the GKS (or PHIGS) and the NAG Graphics Mark 3 libraries must be installed on your system. You have also to install the GKS interface of the NAG Graphics library.
Some small changes due to different file access in different operating systems have to be made to the sources, see `READ.ME` file. For the MS-Windows version Microsoft Fortran V5.1 is needed. This may change in the near future.
All relevant parameters changing the maximum number of atomic orbitals, the number of curve buffers etc. are contained and described in the includefile `dppar.for`. Also the default values for the size of the plots can be changed. It is recommended that these default parameters are not changed, since the ratio of the axes, $width/height = 1/\sqrt{2}$, conforms to standard european DIN paper size settings (A4). The plot dimensions should be compatible to the graphs generated by BANDPLOT, see includefile `bppar.for`. After changing `dppar.for` you have to recompile each subroutine.

In order to compile and link DOSPRT, only the NAG Graphics Library Mark 3 is needed. You have to install the Adobe PostScript interface (APS) of this library. For the MS-Windows version Microsoft Fortran V5.1 is needed. All relevant parameters changing the maximum number of atomic orbitals, the number of curve buffers etc. are contained and described in the includefile `dprpar.for`. After changing this file you have to recompile each subroutine.

8.4.9 Old version - Chapter 4.4.3 BANDPLOT startup

VMS and UNIX version:

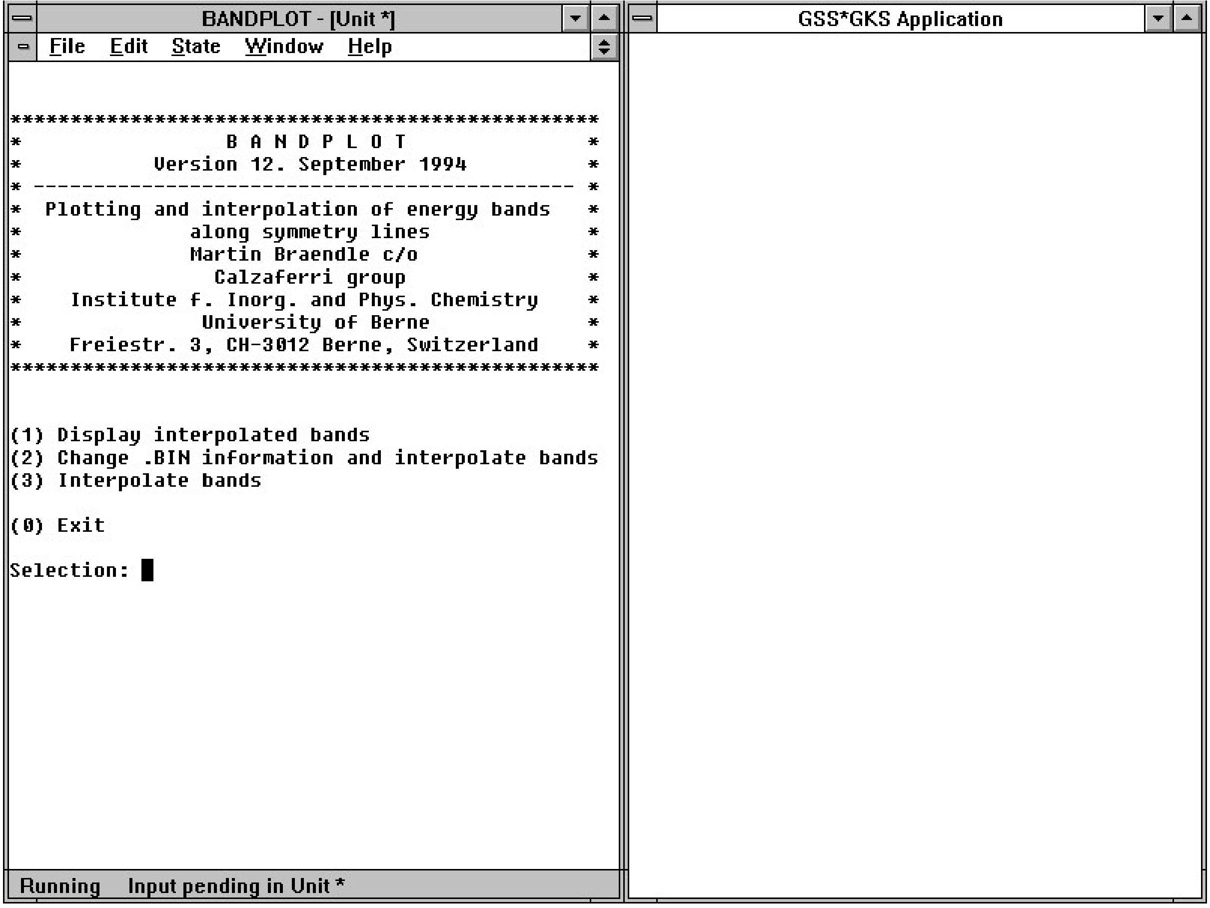
1. The program is called by typing `bandplot filename [display]`
   No extension is to be supplied, default extensions `.ehe` and `.eh1` are assumed .
   `display` is optional and sets the IP address where the X-window has to be displayed.

2. In the VMS version, an icon called `GKS` appears. Deiconify it and position and size the graph window as you want.

MS-Windows version:

Since Microsoft Fortran requires fixed file names in the sources, two shell scripts `BANDPLTS.BAT` and `BANDPLTE.BAT` have been provided, which copy the files to their correct names before and after `BANDPLOT.EXE` is used. You have to perform the following procedure:

1. Call FileManager, change into the directory where `BANDPLTS.BAT`, `BANDPLTE.BAT`, `BANDPLOT.EXE`, and your files `filename.ehe` and `filename.eh1` reside.
2. Select menu `File`, menu-entry `Run`...
3. Type in: `BANDPLTS.BAT filename`, then select OK.
4. Windows executes `BANDPLTS.BAT` by changing into MS-DOS, copies the required files to `FORT.#`, then returns back to FileManager.
5. If `GKSERROR.DAT` exists, delete it.
6. Call `BANDPLOT.EXE` by double-clicking with the mouse on it.
7. A full screen graphical window called `GSS*GKS Application` will be opened. Scale it back to smaller size. Position the terminal window and the graphical window like this:



```
**************************************************
*                 B A N D P L O T                *
*           Version 12. September 1994           *
* ---------------------------------------------- *
*  Plotting and interpolation of energy bands    *
*              along symmetry lines              *
*              Martin Braendle c/o               *
*                Calzaferri group                *
*    Institute f. Inorg. and Phys. Chemistry     *
*               University of Berne              *
*    Freiestr. 3, CH-3012 Berne, Switzerland     *
**************************************************


(1) Display interpolated bands
(2) Change .BIN information and interpolate bands
(3) Interpolate bands

(0) Exit

Selection:
```

Figure 5 Startup screen of BANDPLOT (MS-Windows version).

8. After work, select menu `File`, menu-entry `Run`...
9. Type in: `BANDPLTE.BAT filename`, then select OK.

10. Windows executes `BANDPLTE.BAT` by changing into MS-DOS, copies the files `FORT.#` to the required file names, then returns back to File Manager.

8.4.10 Old version - Chapter 4.4.5.1 BANDPLOT Printing

VMS version:
Select `Generate output file` in the graph modification menu. If there are more than three symmetry lines per graph, the paper orientation changes from portrait to landscape.

MS-Windows version:
Select `Generate output file` in the graph modification menu and then `(1) PostScript output`.
You can select any output device by configuring a printer in the Control Panel of MS-Windows. You have to choose the paper orientation by your own. For correct operation, enter the printer port name (example: `FILE:` or `LPT1:`) in the `KERNEL.SYS` file where your `BANDPLOT.EXE` resides:

```
0:WISS
1:DISPLAY;MOUSE
2:LPT1:                    change here
3:GMO
```

Listing 4: `KERNEL.SYS` file.

AIX version:
Due to lack of a PostScript driver in the present PHIGS version it is not possible to print directly. Save your graph layout, then exit the program and type

```
bandprt filename
```

A PostScript file `filename.ps` is written. If you want to import it as encapsulated PostScript file into a word processor, remove the end-of-file character (Control-D) at the end of the file, delete all text between the curly brackets `{}` in the `/xyinitsys` definition near the beginning of the file, and adjust the bounding box in the `%%BoundingBox:` statement.