Spex Manual

Version 05.13

Nov 09, 2022

Contents

Spex is a computer code based on many-body perturbation theory. It uses the all-electron full-potential linearized augmented plane-wave method (FLAPW), which provides an accurate basis set for all kinds of materials including transition metals, oxides, and even f-electron systems.

Currently Spex can calculate quasiparticle properties (one-shot and self-consistent) using the *GW* approximation, EELS and optical spectra as well as total energies in the RPA approximation, and spin-wave and optical (experimental) spectra from the Bethe-Salpeter equation, Hubbard *U* parameters, Wannier interpolation, and more. (TDDFT is implemented but unmaintained at the moment.)

It needs input from a converged DFT calculation, which can be generated by Fleur but also by any other FLAPW-DFT code.

For further questions about the Spex code, write to spex-users@fz-juelich.de

If you use Spex for your research, please cite the following work:

> Christoph Friedrich, Stefan Blügel, Arno Schindlmayr, "Efficient implementation of the *GW* approximation within the all-electron FLAPW method", *Phys. Rev. B 81, 125102 (2010).*

CHAPTER 1

Installation

To download the Spex code, please register at https://forms.gle/Zu1QXFU94FMfmuzx9
For questions about the Spex code, write to spex-users@fz-juelich.de

1.1 Prerequisite software

For compiling the source you need:

- a modern FORTRAN compiler,
- the BLAS, LAPACK, FFTW, and HDF5 libraries as well as the Wannier90 library if you need maximally localized Wannier functions,
- an implementation of the MPI-3 standard if you intend to compile the parallel version.

1.2 Compilation

Once you have obtained the code ("spexVERSION.tgz") unpack it and change to the "spexVERSION" directory, where "VERSION" is the version number, e.g. "05.00". There, you find a configure shell script that helps to generate a Makefile for your system. Consult the file "INSTALL" for a detailed description of the compilation process. In short, if all necessary libraries are in standard directories, `./configure` and `make` should suffice to create working executables. Special options can be given to the configure script:

- `--enable-mpi`: Build parallel version (requires an MPI-3 compiler).
- `--enable-load`: Load wavefunctions from harddisc when needed, otherwise stored in memory (experimental).
- `--with-wan`: Build with Wannier90 support (keywords MAXIMIZE and FROZEN require the Wannier90 library, whose location can be specified with `--with-wan=DIR`).
- `--with-dft=PROG`: Build with an interface to the DFT program "PROG" instead of the default, which currently is Fleur MaX Release 5.1 version 34; supported interfaces are "fleur": Fleur v0.26b (2019.03), "fleurR3":

Fleur MaX Release 3 v0.27, "fleurR5": Fleur MaX Release 5 version 33, and "fleurR5.1": Fleur MaX Release 5.1 version 34.

- `--prefix=PREF`: Executables will be installed in directory "PREF" (/usr/local is default).

Enter `./configure -h` for a full list of configuration options.

Note: If the configuration fails, the file "config.log" should be consulted for more details about the failure. In most cases, the compiler does not find a required library or an include file, in which case the corresponding directories can be specified with the options `--with-libdir` and `--with-includedir`, respectively. Also see Section 9.1 for common problems in the configuration step.

The compilation creates the executables

"spex.inv" for systems with inversion symmetry and without spin-orbit coupling,

"spex.noinv" for systems without inversion symmetry or with spin-orbit coupling,

"spex.extr" data extraction utility,

The reason for having two executables, one for systems with inversion symmetry and the other for systems without, is that many quantities (such as the Coulomb matrix, the screened interaction on the imaginary axis, the exchange self-energy, etc.) can be declared in the code as real-valued instead of complex-valued arrays in the case of inversion symmetry, which speeds up the computation considerably. (The floating-point multiplication of complex numbers are about eight times slower than the multiplication of real numbers.) For the sake of code clarity, it is then advantageous to have two executables.

The user can call "spex.inv" or "spex.noinv" directly or use the Spex launcher script **"spex"**, which executes the proper executable according to the DFT data in the current directory.

The executables and shell scripts may be installed with `make install`. Installation is optional. See the file "INSTALL" for details. If the directory "PREF/bin" (see –prefix option) is in the $PATH environment variable, Spex can be called from the command line simply by "spex".

Note: If the executables are to be installed in a directory outside your $HOME directory tree, you might have to run the sudo command `sudo make install` requiring you to be in the "sudoers" list.

1.3 Upgrading

The Spex directory "spexVERSION" contains a script to upgrade Spex to the latest version. To upgrade, simply type:

```
$ ./upgrade
```

To just check if there is a newer version available, type:

```
$ ./upgrade --check
```

Upgrading requires the parent directory "spexVERSION/.." to be writable. The upgrade script creates a directory "spexNEW_VERSION" there, unpacks the new archive, configures, and compiles the source. For the configuration, the upgrade script uses the last configuration command issued for the old version (in "spexVERSION"). Note that for the last step (compilation), all modules required for compilation (if applicable) have to be loaded.

In case of failure, consult the file "upgrade.log" for more information. An error will cause the upgrade script to stop immediately. It does not undo all steps up to the point of failure. If you want to try upgrading again, you might

have to remove the directory "spexNEW_VERSION" by hand (`rm -r ../spexNEW_VERSION` from the directory "spexVERSION").

CHAPTER 2

Getting Started

This section is intended to give a first impression on how a typical calculation is carried out. Details are given in later sections.

Many-body perturbation theory is defined with respect to a formally unperturbed system, i.e., a system without electron-electron interaction. Any mean-field theory, e.g., Hartree-Fock or Kohn-Sham (KS) DFT, provides such a system. (For simplicity of presentation, we will assume throughout this manual that KS-DFT is used for the reference system.) Thus, before running Spex, we must find a self-consistent mean-field solution. Normally this first step is fast and we can use a dense k-point set. For the calculation with Spex, we have to generate a new (and usually coarser) k-point set and calculate the corresponding wavefunctions and energies with the mean-field program in a second run. Thus, a typical procedure is

1. self-consistent DFT calculation (e.g., with Fleur),
2. generate new k-point set with Spex,
3. second run of the DFT program to generate the eigenstates at the new k points,
4. finally run Spex.

Of course, for subsequent Spex runs, steps 1-3 need not be repeated, unless the reference mean-field system is to be recalculated with different parameters (e.g., different k-point set or different crystal structure). For the first and third step consult the manual of your DFT program. (The Fleur manual is available online. As explained in more detail in Section 8, you must set `gw=1` and `gw=2` for the first and third step, respectively.) For the second and fourth step you need to create an input file "spex.inp" for Spex. A very simple input file for a *GW* calculation of Si is this:

```
# Quasiparticle energies of bands at the high-symmetry points Gamma, X, and L.

BZ 4 4 4
WRTKPT

KPT X=[0,0,1] L=1/2*[1,1,1]
JOB GW 1:(1,2,5) X:(1,3,5) L:(1-3,5)
```

Note: Refer to the tutorials and other chapters for a full list of available keywords

For the second step, only the lines beginning with `BZ` and `WRTKPT` are necessary. The line `BZ 4 4 4` defines a $4 \times 4 \times 4$ k-point set, and the keyword `WRTKPT` tells Spex to write the (irreducible) k-points to a file ("kpts" for Fleur) and stop.

For the fourth step, you must remove (or comment out) the keyword `WRTKPT`. Now all other parameters are needed. The results are written to standard output and should be piped to an output file (Section 3.3).

Note: As an alternative to `WRTKPT`, the command line option `-w` can be used (`spex -w`). It sets the keyword `WRTKPT` automatically (Section 3.1).

Note: Unless you intend to run self-consistent *GW* calculations, the steps 2-4 can be combined into a single Spex run using the keyword `ITERATE`.

CHAPTER 3

General information

Although the Spex executables ("spex.inv", "spex.noinv") can be run directly, we assume in the following that the Spex launcher script ("spex") is used (see Section 1).

Note: Paragraphs discussing advanced options are preceded with (*), and the ones about obsolete, unmaintained, or experimental options are marked with (**). You can safely skip the paragraphs marked with (*) and (**) at first reading.

3.1 Command-line options

The Spex code recognizes several command-line options (`spex [OPTIONS]`):

- `--version`: output version information and exit
- `--help`: display help and exit
- `--inp=file`: use file as input file instead of the default "spex.inp"
- `-w`: set `WRTKPT` automatically
- `-o=opt`: use special integer option parameter `opt` (for debugging and testing).

3.2 Input file

By default, Spex reads input parameters from the file "spex.inp". The syntax is as follows.

- The file "spex.inp" does not use a fixed format, and the keywords may be given in any order.
- Each keyword is given on one line together with its parameters following it.
- Keywords must not be specified more than once.
- Some keywords are grouped in sections. A section starts with `SECTION sectionname` and ends with `END`.

- All keywords (and section names) are in capital letters.
- Everything after a # sign is interpreted as a comment.
- Comments starting with ## and ### are written to standard output and standard error, respectively, see below.
- Empty lines are allowed.
- Line continuation is possible using a single backslash \ at the end of the previous line.
- Most keyword parameters have default values that are used if the keyword is not specified.
- By default, parameters are given in atomic units. In the case of energies, it is possible to use eV instead, e.g., `1.0eV`.
- The input file can be truncated with the keyword `EXIT`. Everything after `EXIT` is ignored.
- Currently, there are six section keywords: `MBASIS`, `WFPROD`, `COULOMB`, `SUSCEP`, `SENERGY`, and `WANNIER`.

An example for a section in the input file is

```
SECTION SENERGY
  CONTINUE
  SPECTRAL {-10eV:1eV,0.01eV}
END
```

(The indentation is not mandatory and only included here for clarity.)

Spex writes the input file in formatted form to standard output without comments except for the special ## and ### comments, which are written, respectively, to standard output (as a comment) and to standard error (as a warning). The former can be used to give details about the calculation (example `## silicon bulk (QSGW)`) and the latter to give a special warning (example `### test run with extreme parameters`) that, for greater visibility, should appear in the standard error stream.

Note: The keywords are detailed in the coming chapters. If a keyword belongs to a section, the section's name is specified as well, for example, `CONTINUE (SENERGY)`.

The shell script "spex.setkey" allows the Spex input file to be modified from the command line. This is useful, in particular, for a multiple calculations run from a batch script. Type "spex.setkey –help" for more details.

3.3 Output files

The main output file of a Spex calculation is written directly to standard output (stdout). Errors, warnings, and (additional) info(rmation) are written to standard error (stderr). Errors stop the program run. Errors, warnings, and infos are of the form:

```
SPEX-ERROR (source.f:0123) Error message.
SPEX-BUG (source.f:0123) Error message.
SPEX-WARNING (source.f:0123) Warning message.
SPEX-INFO (source.f:0123) Info message.
```

respectively, where "source.f" is the name of the source file and 0123 is the respective line where the error (etc.) has occurred. An error message starting with SPEX-BUG is likely caused by a bug in the code (please inform the developers). A warning message indicates a possible problem in the calculation, but the calculation continues anyway. An info message informs about a less important issue, for example, about the usage of obsolete syntax in the input file.

Often, when an MPI run stops with an error, the output gets cluttered with many error messages from the MPI library and the operating system. In this case, you should find the relevant Spex error message before these lines.

It is recommendable to pipe the output to a file (in Bash)

Pipes stdout to "spex.out" and stderr to the screen:

```
$ spex > spex.out
```

Pipes stdout and stderr to "spex.out":

```
$ spex &> spex.out
```

Pipes stdout to "spex.out" and stderr to "spex.err":

```
$ spex > spex.out 2> spex.err
```

For the parallelized version, Spex has to be run through an MPI launcher, for example, `mpiexec -np 8 spex`. See Section 7 for details.

Depending on the job type and keywords, there are additional output files containing, for example, the resulting spectra. These output files can be in ASCII and binary format. Further details are given in Section 5.

For reference, we give a list here:

"spex.cor" Bare and screened Coulomb matrix. (Depending on the type of calculation, the susceptibility or dielec matrix might be written to "spex.cor" instead of the screened Coulomb matrix.)

"spex.cot" T matrix

"spex.mb" MT product basis

"spex.core" Core contribution to susceptibility (it is k-independent, therefore precalculated)

"spex.dos" Density of states

"spex.sigx" Exchange (HF or COHSEX) self-energy

"spex.sigc" Correlation (GW) self-energy

"spex.sigt" Correlation (GT) self-energy

"spex.ccou" *Contracted* screened Coulomb interaction (e.g., for COHSEX)

"spex.wcou" Wannier-projected Coulomb interaction parameters (Hubbard U)

"spex.uwan" Wannier transformation matrix U

"spex.kolap" Overlap matrix (across k points) for Wannier construction with Wannier90

"suscep", "dielec", etc. Files containing spectra

"<KS>" KS eigensolutions (the name depends on the DFT program used, e.g., "KS.hdf".)

CHAPTER 4

Common Keywords

This section discusses several settings that are common for all calculations. Some of them are explained in more detail in later chapters.

4.1 General Settings

4.1.1 JOB

Most Spex runs need a job definition, which defines what Spex should do, e.g., `JOB GW 1:(1-5)` for a *GW* calculation of the specified set of bands. The available jobs are

- `GW` - *GW* method
- `HF` - Hartree-Fock method
- `PBE0` - PBE0 hybrid functional
- `SX` - Screened exchange
- `COSX` - Coulomb-hole screened-exchange (COHSEX)
- `GT` - *GT* method
- `GWT` - *GW* and *GT* combined
- `HFENE` - Total Hartree-Fock exchange energy
- `RPAENE` - RPA correlation energy
- `SUSCEP` - Susceptibility (Polarization function)
- `SUSCEPR` - Renormalized susceptibility (RPA or Bethe-Salpeter)
- `DIELEC` - Dielectric function
- `SCREEN` - Screened interaction
- `SCREENW` - Screened interaction projected onto Wannier basis

- `KS` - Single-particle energies (e.g., KS energies), included for completeness

Details of these jobs are explained in subsequent sections. The job definition must not be omitted but may be empty: `JOB`, in which case Spex will just read the wavefunctions and energies, perform some checks and some elemental calculations (e.g., Wannier interpolation), and stop.

The job `KS` is included only for convenience. When given, Spex simply writes the single-particle energies (e.g., KS energies), read from the input data, to the output in the *standard* format, similar to the output of *GW* or HF values.

In principle, Spex supports multiple jobs such as `JOB GW 1:(1-5) DIELEC 1:{0:1,0.01}`. This feature is, however, seldom used and is not guaranteed to work correctly in all versions.

Table 1: Examples

`JOB COSX 1:(1-5)`	Perform COHSEX calculation for states 1,2,...,5 at the first k point.
`JOB COSX 1:(1-5) SCREEN 2:{0:1,0.01}`	Subsequently, calculate the screened interaction on a frequency mesh.
`JOB GW 1:(1,2,5-8) 7:(1-3,6-8)`	Perform *GW* calculation for the specified states at the first and seventh k point.
`JOB`	Just perform some checks and stop. (Same as omitting `JOB`.)

4.1.2 BZ

There are only two keywords that must not be omitted from the input file. The first is `JOB` (Section 4.1.1), the second `BZ`, which defines the k-point set for the sampling of the Brillouin zone. The keyword requires four integer arguments, e.g., `BZ 6 6 4`, defining, in this case, a $6 \times 6 \times 4$ k-point set. With the command-line option `-w` (`spex -w`) (or, alternatively, if the keyword `WRTKPT` is set), Spex uses the `BZ` definition to construct the irreducible wedge of the k-point set and writes the respective list of k points to a file, which can then be read by the DFT code.

Table 2: Example

`BZ 8 4 2.`	Define an $8 \times 4 \times 2$ k-point set.

4.1.3 WRTKPT

(*) When set, Spex writes a k-point file (to be read by the DFT code) and stops (also see Section 4.1.2). Users are advised to use the equivalent and more convenient command-line option `-w`, instead (Section 3.1).

4.1.4 MEM

Calculations based on many-body perturbation theory are costly, not only in terms of computation time but also in terms of memory. Since the problem sizes (e.g., number of atoms) are thus effectively restricted by the memory hardware of the computer, the latter constitutes in some ways a more serious limit than the computation time the user can afford to invest. Therefore, an upper limit for the memory that Spex is allowed to use can be defined with the keyword `MEM` followed by the upper limit, either in MB or in percent of the total memory, as read from the entry "MemTotal" of the system file "/proc/meminfo". If `MEM` is not specified, Spex sets the limit to 50% of the total memory. If reading from "/proc/meminfo" fails, `MEM` defaults to 10 GB. In parallel Spex, the value would correspond to the upper limit per (physical) node. If possible, Spex will divide the work load into packets in such a way that the memory usage stays below the limit. (Spex might use a bit more memory than specified. Also keep in mind that the operating system and other tasks running on the computer need memory.)

Table 3: Example

`MEM 2000`	Restrict memory usage to 2 GB.
`MEM 25%`	Restrict memory usage to 25% of the total memory.

4.1.5 STOREBZ

(*) Spex reads (or calculates) wavefunctions and energies as a first step. In general, they are needed at all k vectors in the set (e.g., for all 64 points of a $4 \times 4 \times 4$ set) and would thus occupy a lot of memory. By default, only the wavefunctions and energies in the irreducible wedge of the Brillouin zone (IBZ) are stored in memory (e.g., eight instead of the 64 points in the case of a diamond lattice). Spex generates the wavefunctions on k points outside the IBZ on the fly (by applying symmetry operations). Of course, this implicates a small computational overhead and might slow down the calculation. The keyword `STOREBZ` switches this feature off and stores the wavefunctions and energies at all k points of the Brillouin zone, which can also be helpful in code development because all wavefunctions are available in memory at all times and do not have to be generated. (Until version 05.05, `STOREBZ` was the default, and the new default was chosen with `STOREIBZ`.)

4.1.6 –enable-load

(*) Another possibility to reduce the memory consumption is to reconfigure (and recompile) the code with the option `--enable-load` (or recompile with the flag `-DLOAD` in the Makefile). With this option, Spex stores the wavefunctions in memory only temporarily and rereads them from harddisc whenever they are needed. Obviously, this requires a fast harddisc and network connection. The HDF5 library is used for the data transfer. There is no possibility to set the option in the input file because it strongly affects the way the wavefunction data is handled in the code.

4.1.7 RESTART

This is an important keyword. It enables (a) continuing a calculation that has, for some reason, stopped or crashed and (b) reusing data of a previous calculation. If specified, Spex will store intermediate results in a file or read them if the particular data is already present in the file. Even if `RESTART` was not specified in the first run, Spex might still be able to reuse data from a previous run with `RESTART 2`. (This option usually gives less freedom for a change of parameters.) It depends on the particular job definition whether `RESTART` and/or `RESTART 2` are available and how they work. See below for details.

Table 4: Examples

`RESTART`	Read/write restart file.
`RESTART 1`	Same as `RESTART`
`RESTART 2`	Try to reuse data from standard output files.

(*) We will come back to this keyword in the tutorials. Here, we give a more detailed explanation as reference.

Spex writes several files to harddisc during the program run (see Section 3.3 for details): “spex.cor”, “spex.cot”, “spex.ccou”, “spex.mb”, “spex.core”, “spex.sigx”, “spex.sigc”, “spex.sigt”, “spex.wcou”, “spex.uwan”, “spex.kolap”, as well as several files containing spectra (“suscep”, “dielec”, etc.) and a file “<KS>” containing the KS wavefunctions. (The name of the latter depends on the DFT program used.)

In addition to the arguments 1 and 2, described above, the keyword `RESTART` can be given with a general five-digit integer number, e.g., `22233`, which explicitly specifies which files should be read and/or written. (The first digit is optional. It is interpreted as 0 if omitted.) The digits can be 0 (don’t read and don’t write), 1 (read but don’t write), 2 (write but don’t read), and 3 (read and write). The following table shows to which file(s) each digit refers to.

Table 5: RESTART codes

	"spex.uwan" "spex.kolap"	"spex.sigc" "spex.sigt" "spex.wcou" spectra files[1]	"spex.sigx"	"spex.cor" "spex.cot" "spex.ccou" "spex.mb" "spex.core"[1]	"<KS>"
Don't read, don't write	0	0	0	0	0
Read but don't write	1	1	1	1	1
Write but don't read	2	2	2	2	2
Read and write	3	3	3	3	3

[1]The spectra files and "spex.core" are always written.

The arguments 1 and 2 correspond to `22233` and `33333`, respectively. The default (i.e., without `RESTART`) is `02202`.

When using `MPIKPT` in a parallel calculation, the restart data is written to multiple files: "spex.sigx", "spex.sigx.2", "spex.sigx.3", etc. (accordingly for "spex.sigc") as well as "spex.cor.1", "spex.cor.2", etc. In the latter case, there is also a directory "spex.cor.map" containing, for each k point, links to the respective *cor* file. Whenever restart files need to be copied, make sure that all files (including the directory) are copied. Apart from that, Spex will always manage to read the correct data, independently of whether the calculation is in parallel or serial, with or without `MPIKPT`.

Warning: The `RESTART` option is a powerful tool. It is not only of practical importance due to the possibility of restarting calculations or reusing data of previous calculations. It also gives the user a lot of freedom to tailor calculations, for example, using different k-point sets (or different numbers of bands, etc.) for G and W. However, this freedom comes with the risk that calculations are performed with inconsistent data. Spex has only limited capabilities to catch such inconsistencies. The user should be aware of that.

4.2 Input data

Spex needs data from a previous self-consistent solution of a mean-field system (e.g., KS-DFT). Several keywords can be used to analyze, check, and modify the input data.

4.2.1 NBAND

The Green function involves a summation over electronic bands, as do the polarization function [Eq. (5.5)] and the self-energy. The convergence with respect to the number of bands is a very important aspect in calculations based on many-body perturbation theory. The keyword `NBAND n` sets the number of bands to n. Obviously, n must not be larger than the number of bands provided by the input data. It should be noted that the actual number of bands can differ from n and also from k point to k point because Spex makes sure that degenerate subspaces are not cut in order to avoid symmetry breaking. If n is a real number (i.e., if it contains a decimal point: `3.0`), it is interpreted as the maximal band energy. Then, all bands up to this energy are included. This is helpful if one wants to use a single parameter for calculations with differently sized unit cells. For example, when the unit cell doubles, one would have to include twice as many bands to reach the same accuracy, while the maximal band energy (practically) remains unchanged. If `NBAND` is unset, all available bands are used.

Table 6: Examples

`NBAND 100`	Use 100 bands per k point.
`NBAND 3.0`	Use all bands up to 3 Hartree.
`NBAND 50.0eV`	Use all bands up to 50 eV.

4.2.2 CORESOC

By default, Spex averages over the SOC-splitted states. For example, the two states $2p^{1/2}$ and the four states $2p^{3/2}$ are averaged to give the six quasi-non-relativistic states of the 2p shell (or the three spin-up 2p states and the three spin-down 2p states in the case of spin polarization). Then, all core states of a given shell have the same energy and the same radial form of the wavefunction. The averaging procedure can be avoided by specifying `CORESOC`. The SOC-splitted states and the different energies and wavefunctions are then retained and taken into account in the calculations of the self-energy, polarization function, et cetera. In the case of spin polarization, the Weiss field gives rise to further lifting of degeneracies. Spex calculates the respective new splittings, energies, and wavefunctions. For reference, the core-state energies are written to the output file. Usually, the effect of `CORESOC` is numerically small, but it still might play an important role for *GW* calculations in the case of very small band gaps.

4.2.3 PROJECT (ANALYZE)

This keyword lets Spex analyze the orbital character of the wavefunctions. The LAPW basis set (see Section 6.1) facilitates such an analysis by virtue of the atomic (muffin-tin) spheres, in which the wavefunctions are expanded in terms of spherical harmonics: $\phi_{\mathbf{k}n}(\mathbf{r}) = \sum_{lm} \varphi^{\mathbf{k}n}_{lm}(r) Y_{lm}(\hat{\mathbf{r}})$, where the $\varphi^{\mathbf{k}n}_{lm}(r)$ are radial functions (defined on a radial grid). For example, the contribution of the *s* orbital to $\phi_{\mathbf{k}n}(\mathbf{r})$ would be given by $\langle \varphi^{\mathbf{k}n}_{00} | \varphi^{\mathbf{k}n}_{00} \rangle$ (and is between 0 and 1). By default, the projection is onto *s*, *p*, *d*, *f*, and *g* orbitals (summed over the magnetic quantum number *m* and over all equivalent atoms). Alternatively, the orbitals can be given as arguments after `PROJECT` following the same syntax as the Wannier orbital choice (see Section 6.3.1). With this syntax, the default would be defined by `[s,p,d,f,g]` for each atom type. See below for more examples.

The results are given in the form of a table in the file "spex.binfo" containing the orbital decomposition for each eigenstate at one or more k points (at the Γ point, the additional "+" point if defined, see Section 5.1.2, and at all k points that may be specified in the `JOB` definition, which allows giving the k point in a dummy job such as `JOB KS X:(1)`). If `DOS` (Section 4.2.4) is given, a corresponding projected density-of-states (DOS) plot is written to the file "spex.dos". The orbital analysis is also carried out for quasiparticle wavefunctions if they are different from the non-interacting reference wavefunctions (i.e., whenever the `FULL` keyword is used, see Section 5.1.19). In addition, it is done for DFT band-structure calculations (`ITERATE BANDS`, see Section 4.2.9) and Wannier band interpolations (DFT, *GW*, ...) (`INTERPOL`, see Section 5.1.17 and Section 6.3.7), enabling the plot of "fat bands".

Table 7: Examples

`PROJECT`	Calculate orbital decomposition (and DOS plots, see Section 4.2.4) for *s*, *p*, *d*, *f*, and *g* orbitals, corresponding to the default argument `[s,p,d,f,g]` for each atom type.
`PROJECT [p,eg]`	Do the projection, instead, on *p* and e_g orbitals of the first atom type. Two values are given, the contribution of *p* states summed over p_x, p_y, and p_z as well as over all equivalent atoms, and the correspondingly summed contribution of the e_g states.
`PROJECT (px,py,pz,dx2y2,dz2)`	The projection is on each orbital separately: p_x, p_y, p_z, $d_{x^2-y^2}$, and d_{z^2} and only for the first atom.
`PROJECT () (px/90/0/0)`	The first atom is omitted in the analysis (empty parentheses), and the p_x orbital at the second atom is rotated using Euler angles, see Section 6.3.1. In this particular case, the resulting orbital is identical to p_y.

4.2.4 DOS (ANALYZE)

This lets Spex write a density-of-states (DOS) plot to the file "spex.dos". See Section 4.2.3 for a definition of the corresponding orbitals for a projected DOS plot.

4.2.5 FIXPHASE

The phases of eigenstates are not unique, since multiplying an eigenstate with a complex phase factor (c with $|c| = 1$, e.g., $c = -1$) gives a valid eigenstate with the same eigenvalue. Likewise, any unitary transformation of a set of degenerate eigenstates gives an equally valid set of degenerate eigenstates with the same (common) energy eigenvalue. The wave-function coefficients are thus not uniquely defined. This can be problematic for quantities that are defined with respect to the wave functions, e.g., the Wannier *U* matrix or the *GW* self-energy matrix (calculated with `GW FULL`) because a recalculation of the eigenstates is likely to change the phases (and rotations in degenerate eigenspaces). Spex catches such inconsistencies and will stop with an error message ("inconsistent checksum"). On the other hand, the keyword `FIXPHASE` can be used to make the wave-function coefficients unique. Obviously, to take effect, `FIXPHASE` has to be set in all calculations that write or read quantities defined with respect to the eigenstates, e.g., the Wannier *U* matrix ("spex.uwan") or the self-energy matrix ("spex.sigx" or "spex.sigc"). `FIXPHASE` fixes both the phases of the eigenstates and the rotation within degenerate eigenstates. It is not set by default.

4.2.6 ENERGY

(*) In some cases, it may be necessary to replace the energy eigenvalues, provided by the mean-field (DFT) calculation, by energies (e.g., *GW* quasiparticle energies) obtained in a previous Spex run, for example, to determine the *GW* Fermi energy or to perform energy-only self-consistent calculations. This can be achieved with the keyword `ENERGY "file"`, where "file" contains the new energies in eV. The format of "file" corresponds to the output of the `spex.extr` utility: `spex.extr g spex.out > file`. It must be made sure that "file" contains energy values for the whole irreducible Brillouin zone. Band energies not contained in "file" will be adjusted so that the energies are in ascending order (provided that there is at least one energy value for the particular k point). The corresponding wave functions will be ordered accordingly.

This keyword can also be used to define a scissors operator, see examples.

Table 8: Example

`ENERGY "energy.inp"`	Replace the mean-field energy eigenvalues by the energies provided in the file "energy.inp".
`ENERGY 0.8eV`	Apply a scissors operator with 0.8 eV.

4.2.7 DELTAEX

(*) This keyword modifies the exchange splitting of a collinear magnetic system, i.e., it shifts spin-up and spin-down energies relative to to each other so as to increase or decrease the exchange splitting. With `DELTAEX x`, the spin-up (spin-down) energies are lowered (elevated) by x/2. The parameter x can be used to enforce the Goldstone condition in spin-wave calculations [Phys. Rev. B 94, 064433 (2016)]

Table 9: Example

`DELTAEX 0.2eV`	Increase the exchange splitting by 0.2 eV (spin-up/down energies are decreased/increased by 0.1 eV)

4.2.8 PLUSSOC

(*) With this keyword, Spex adds spin-orbit coupling (SOC) to the mean-field solution using a one-shot second-variation approach and stops, so it is non-iterative but updates the wavefunctions and energies. The new energies are written to standard output and, together with the new wavefunctions, to new input files on the harddisc, thus overwriting the old input data. In the case of Fleur v0.26b (2019.03), the modified files are "gwa", "KS.hdf", and "spex.sym". The latter is written only if the system is spin polarized (non-collinear magnetism). It has the same form as "sym.out". Spex reads the file "spex.sym" if it exists and "sym.out" otherwise. After the `PLUSSOC` calculation, a subsequent Spex run will read the new input data, now including SOC. (The keyword `PLUSSOC` must be removed for this new run, otherwise Spex stops with an error, since it cannot add SOC to a solution that contains it already.)

A possible application of this feature is to add SOC corrections to *GW* quasiparticle energies already calculated. To this end, store the *GW* energies to a file using the "spex.extr" utility (example: `spex.extr g spex.out > energy.inp`). Then, tell Spex to replace the single-particle energies (that it reads from the input data) by the ones from the file ("energy.inp") with the keyword `ENERGY` (`ENERGY energy.inp`, see Section 4.2.6) and add `PLUSSOC`. In a subsequent run, Spex will apply the SOC operator to the *GW* quasiparticle energies using a second-variation technique, write the corrected energies to the output, and write the new input data files. The latter can be used in a subsequent Spex run, for example, to calculate a Wannier-interpolated band structure, containing *GW* quasiparticle and SOC corrections. From a theoretical point of view this approach is a bit inconsistent [Phys. Rev. B 88, 165136 (2013)], but it is considerably faster than running *GW* with SOC from the start. Besides, the latter approach is currently not applicable to systems with noncollinear magnetism. So, using the present procedure with `PLUSSOC` is the only viable approach in this case.

4.2.9 ITERATE

(*) If `ITERATE` is specified, Spex only reads the LAPW basis set from the input data, provided by the mean-field (DFT) code, but solves the Kohn-Sham (KS) equations (as a generalized eigenvalue problem) at the k-points itself. This calculation effectively replaces the second run of the DFT code. (In this sense, the name of the keyword is a bit misleading, as the calculation is non-iterative.) `ITERATE` can also run band structure calculations. The k-point path is taken from `KPTPATH` or from a file, see below. The band structure is written to the file "bands". Using `PROJECT` it is also possible to calculate band structures with orbital projections ("fat bands"). Spex offers the possibility of modifying the LAPW basis set (when `ITERATE` is used) in the section `LAPW` of the input file (Section 6.1) The keyword `ITERATE` is not available for executables compiled with `-DLOAD` (configured with `--enable-load`).

Table 10: Examples

`ITERATE`	Solve KS equations. (Includes SOC if it is included in the underlying DFT calculation.)
`ITERATE NR`	Solve non-relativistic KS equations. (Excludes SOC.)
`ITERATE SR`	Solve scalar-relativistic KS equations. (Excludes SOC.)
`ITERATE FR`	Solve relativistic KS equations. (Includes SOC.)
`ITERATE MIN=-1`	Solve KS equations and neglect eigenvalues below -1 htr.
`ITERATE SR MIN=-5eV`	Solve scalar-relatvisitic KS equations and neglect eigenvalues below -5 eV.
`ITERATE FR STOP`	Solve relativistic KS equations (including SOC), then stop.
`ITERATE BANDS`	Run band structure calculation along the k-point path defined by `KPTPATH`, then stop.
`ITERATE FR BANDS "qpts"`	Include SOC in the band structure calculation and take the k-point path from the file "qpts".

Note: `ITERATE NR` (or `SR` or `FR`) overrides the SOC flag from the DFT calculation. For example, `ITERATE FR` includes SOC irrespectively of whether the underlying DFT calculation (which has generated the effective KS potential) has been run with or without SOC.

4.2.10 NOSYM

(*) Spex uses spatial (and time-reversal) symmetry to speed up calculations. However, in case of large systems and/or slightly broken symmetry (small structural distortions), the automatic detection of symmetry and its usage can be problematic. The keyword `NOSYM` allows switching off (riskier procedures involving) symmetry.

4.2.11 CHKMISM and CHKOLAP

(*) The set of wavefunctions read from harddisc can be checked in terms of their orthonormality and continuity (mismatch) at the muffin-tin (MT) boundary with the keywords `CHKOLAP` and `CHKMISM`, respectively. The former test gives a value for the deviation from exact orthonormality. The smaller the value, the better the condition of orthonormality is fulfilled. If the deviation is too large, the program stops. The second keyword yields, for each k point, the average and maximal MT mismatch of the wavefunctions. If any of these tests yield too large errors, there could be a problem with the read-in process or with the data written by the DFT code. The developers should be informed in this case.

4.2.12 MTACCUR

(*) The LAPW method relies on a partitioning of space into MT spheres and the interstitial region. The basis functions are defined differently in the two regions, interstitial plane waves in the latter and numerical functions in the spheres with radial parts $u(r), \dot{u}(r) = \partial u(r)/\partial\epsilon$, $u^{\mathrm{LO}}(r)$ and spherical harmonics $Y_{lm}(\hat{\mathbf{r}})$. The plane waves and the angular part of the MT functions can be converged straightforwardly with the reciprocal cutoff radius $g_{\max}$ and the maximal l quantum number $l_{\max}$, respectively, whereas the radial part of the MT functions is not converged as easily. The standard LAPW basis is restricted to the functions u and $\dot{u}$. Local orbitals u^{LO} can be used to extend the basis set, to enable the description of semicore and high-lying conduction states. The accuracy of the radial MT basis can be analyzed with the keyword `MTACCUR e1 e2` which gives the MT representation error [Phys. Rev. B 83, 081101] in the energy range between e1 and e2. (If unspecified, e1 and e2 are chosen automatically.) The results are written to the output files "spex.mt.t" where t is the atom type index, or "spex.mt.s.t" with the spin index s(=1 or 2) for spin-polarized calculations. The files contain sets of data for all l quantum numbers, which can be plotted separately with gnuplot (e.g., `plot "spex.mt.1" i 3` for $l = 3$)

Table 11: Examples

`MTACCUR -1 2`	Calculate MT representation error between -1 and 2 Hartree.
`MTACCUR`	Automatic energy range.

Note: Paragraphs discussing advanced options are preceded with (*), and the ones about obsolete, unmaintained, or experimental options are marked with (**). You can safely skip the paragraphs marked with (*) and (**) at first reading.

CHAPTER 5

Tutorials

In this section, we provide detailed tutorials for the main features of the Spex code.

5.1 *GW* calculations

The Spex code was started as a pure *GW* code. The *GW* approach remains the main computational method implemented in the code.

5.1.1 One-shot *GW* approach

A simple input file for a one-shot *GW* calculation for bulk silicon was already presented in Section 2. The following, equivalent input file is even more minimalistic.

```
BZ 4 4 4
JOB GW 1:(1,2,5) 7:(1,3,5) 3:(1-3,5)
```

It only contains the two essential keywords that any Spex input file must contain: `BZ` and `JOB`, specifying, respectively, the k mesh for the Brillouin-zone sampling and the requested type of calculation. Here, we calculate quasiparticle corrections for the bands (1,2,5) at the first, (1,3,5) at the seventh, and (1,2,3,5) at the third k point. The band indices are defined according to the ordering in the mean-field input data. In the example, we have left out some band indices because of energy degeneracy. For example, bands 3 and 4 (of the first k point) are degenerate with band 2.

In the case of a spin-polarized system, Spex calculates the quasiparticle energies for both spins by default. Alternatively, one can choose the spin index by using u (spin up) or d (spin down), e.g., `JOB GW 1:u(1,2,5) 1:d(1,2,5)` is identical to `JOB GW 1:(1,2,5)`.

The same definition of bands is used for the job types `HF`, `PBE0`, `SX`, `COSX`, `GT`, `GWT`, and `KS`. For example, `JOB COSX 1:(1,2,5)` would run a COHSEX calculation. Many of the keywords described in this section work identically for the other job types, others can only be used for *GW* calculations. If a keyword is inapplicable to the defined job, it is simply ignored by Spex (e.g., `SPECTRAL` for a PBE0 calculation).

5.1.2 KPT

The k points follow an internal ordering: first the irreducible (parent) k points, then the remaining equivalent k points. A list of the irreducible k-vectors is written to the output file. In the present example, there are eight irreducible k points and 64 k points in total. All 64 k-point indices can be used in the job definition. We have chosen the ones from the irreducible set. They correspond to the Γ (index 1), X (index 7), and L point (index 3). Clearly, for a different k-point set, e.g., $8 \times 8 \times 8$, the k-point indices would change (except for the index 1, which always corresponds to the Γ point). Therefore, there is the possibility of defining k-point labels by

`KPT X=1/2*(0,1,1) L=1/2*(0,0,1)` or `KPT X=[1,0,0] L=1/2*[1,1,-1]`

The labels can be ten characters long and may include upper- and lower-case characters, numbers, and also special characters except blanks. Of course, the labels must not be integer numbers. Fractions can also be used inside the brackets, e.g., `X=(0,1/2,1/2)` would be identical to the first k-point label. The two `KPT` definitions shown above are equivalent. The first gives the k vectors in internal coordinates, i.e., in the basis of the reciprocal basis vector, the second in cartesian coordinates in units of $2\pi/a$ with the lattice constant a. In the case of the fcc lattice of silicon, the lattice basis vectors are $a_1 = a(0,1,1)/2$, $a_2 = a(1,0,1)/2$, and $a_3 = a(1,1,0)/2$. The reciprocal basis vectors are thus $b_1 = 2\pi/a(-1,1,1)$ et cetera. For the X point, e.g., one then gets $(a_2 + a_3)/2 = 2\pi/a(1,0,0)$. In order for the cartesian coordinates (square brackets) to be interpreted correctly, Spex obviously needs to know the lattice constant a. (In Fleur, the lattice constant is taken to be the global scaling factor for the lattice vectors.) The so-defined k points must be elements of the k mesh defined by `BZ`. We will later discuss how points outside the k mesh can be considered using the special label `+`. With the k labels, the above job definition can be written as `JOB GW 1:(1,2,5) X:(1,3,5) L:(1-3,5)` as in the input file described in Section 2. There are two more special k-point labels: `IBZ` and `BZ` (e.g., `JOB GW IBZ:(1,2,5)`) stand for all k points in the irreducible and the full k set, respectively. (The label `BZ` is included for completeness but is not needed in practice.) The `IBZ` label is helpful when a Wannier interpolation of *GW* quasiparticle energies or a self-consistent *GW* calculation is to be performed, which require the self-energy to be calculated for the whole irreducible Brillouin zone.

The quasiparticle energies are written to standard output in tabular form:

```
Bd      vxc     sigmax     sigmac          Z         KS         HF          GW lin/dir
1 -12.15897 -19.20415   6.60649   0.65146  -6.19011 -13.23529  -6.36401   0.73681
                        1.05993  -0.10556                      -6.36806   0.72704
2 -13.30408 -14.45814   0.69809   0.76876   5.77669   4.62263   5.42615   0.00070
                        0.00039  -0.00088                       5.42695   0.00081
5 -11.49631  -7.27160  -3.86726   0.76385   8.34013  12.56485   8.61313  -0.00731
                       -0.00708  -0.00533                       8.61239  -0.00749
```

The columns are

- `Bd` band index,
- `vxc` expectation value of the exchange-correlation potential v^{xc}, e.g., -12.15897 eV,
- `sigmax` expectation value of the exchange self-energy Σ^x, e.g., -19.20415 eV,
- `sigmac` expectation value of the correlation self-energy Σ^c, e.g., (6.60649+1.05993*i*) eV (note that the self-energy is complex),
- `Z` renormalization factor Z, e.g., (0.65146-0.10556*i*) eV,
- `KS` mean-field energy eigenvalue (e.g., from the previous KS-DFT calculation), e.g., -6.19011 eV,
- `HF` Hartree-Fock value (one-shot), e.g., -13.23529 eV,
- `GW` *GW* quasiparticle values, e.g., (-6.36401+0.73681*i*) eV (linearized solution) and (-6.36806+0.72704*i*) eV (direct solution).

The two *GW* values for the quasiparticle energy follow two common methods to approximately solve the quasiparticle

equation

$$\left\{-\frac{\nabla}{2}+v^{\mathrm{ext}}(\mathbf{r})+v^{\mathrm{H}}(\mathbf{r})\right\}\psi_{\mathbf{k}n}(\mathbf{r})+\int\Sigma^{\mathrm{xc}}(\mathbf{r},\mathbf{r}';E_{\mathbf{k}n})\psi_{\mathbf{k}n}(\mathbf{r}')d^3r'=E_{\mathbf{k}n}\psi_{\mathbf{k}n}(\mathbf{r}) \tag{5.1}$$

where v^{ext}, v^{H}, Σ^{xc}, $\psi_{\mathbf{k}n}$, $E_{\mathbf{k}n}$ are the external and Hartree potential, the *GW* self-energy, and the quasiparticle wavefunction and energy, respectively. Taking the difference $\Sigma^{\mathrm{xc}}-v^{\mathrm{xc}}$ as a small perturbation, we can write the quasiparticle energy as a correction on the mean-field eigenvalue

$$E_{\mathbf{k}n}=\epsilon_{\mathbf{k}n}+\langle\phi_{\mathbf{k}n}|\Sigma^{\mathrm{xc}}(E_{\mathbf{k}n})-v^{\mathrm{xc}}|\phi_{\mathbf{k}n}\rangle\approx\epsilon_{\mathbf{k}n}+Z_{\mathbf{k}n}\langle\phi_{\mathbf{k}n}|\Sigma^{\mathrm{xc}}(\epsilon_{\mathbf{k}n})-v^{\mathrm{xc}}|\phi_{\mathbf{k}n}\rangle \tag{5.2}$$

with the single-particle wavefunction $\phi_{\mathbf{k}n}$ and the frequency-independent potential v^{xc}, which in the case of a KS solution would correspond to the local exchange-correlation potential; the nonlocal Hartree-Fock exchange potential and the *hermitianized* self-energy of QSGW (see below) are other examples. $Z_{\mathbf{k}n}=[1-\partial\Sigma^{\mathrm{xc}}/\partial\omega(\epsilon_{\mathbf{k}n})]^{-1}$ is called the renormalization factor. The two expressions on the right-hand side correspond to the "direct" (iterative) and "linearized" (non-iterative) solutions given in the output. The direct solution takes into account the non-linearity of the quasiparticle equation and is thus considered the more accurate result. However, there is usually only little difference between the two values. Up to this point, the job syntax for Hartree Fock (`JOB HF`), PBE0 (`JOB PBE0`), screened exchange (`JOB SX`), COHSEX (`JOB COSX`), and *GT* (`JOB GT` and `JOB GWT`) calculations is identical to the one of *GW* calculations, e.g., `JOB HF FULL X:(1-10)`. (The keyword `FULL` is explained in Section 5.1.19.) Except the latter (*GT*), all of these methods are mean-field approaches, so one only gets one single-particle energy (instead of a ''linearized" and a ''direct" solution) for each band.

5.1.3 SPECTRAL (SENERGY)

(*) It should be pointed out that the quasiparticle energies given in the output rely on the quasiparticle approximation. The more fundamental equation, as it were, is the Dyson equation

$$G(\omega)=G_0(\omega)+G_0(\omega)[\Sigma^{\mathrm{xc}}(\omega)-v^{\mathrm{xc}}]G(\omega)$$

which links the interacting Green function G to the non-interacting KS one G_0 and which, in principle, requires the self-energy to be known on the complete ω axis. The spectral function measured in photoelectron spectroscopy is directly related to the Green function by

$$A(\mathbf{k},\omega)=\pi^{-1}\,\mathrm{sgn}(\omega-\epsilon_{\mathrm{F}})\;\mathrm{tr}\left\{\mathrm{Im}[\omega I-H^{\mathrm{KS}}(\mathbf{k})-\Sigma^{\mathrm{xc}}(\mathbf{k},\omega)]^{-1}\right\}\,,$$

where the trace (tr) is over the eigenstates and ϵ_{F} is the Fermi energy. The spectral function can be evaluated using the keyword `SPECTRAL` in the section `SENERGY`. Optionally, one can define a frequency mesh by the keyword `SPECTRAL` in the section `SENERGY`, in the examples below from -10 eV to 1 eV in steps of 0.01 eV. (The Fermi energy is at 0 eV.) If there is no explicit mesh, Spex chooses one automatically. The spectral function is then written to the file "spectral", one block of data for each k point given in the job definition. There is another optional parameter given at the end of the line (third example below), which can be used to bound the imaginary part of the self-energy and, thus, the quasiparticle peak widths from below by this value (unset if not given). This can be helpful in the case of very sharp quasiparticle peaks that are otherwise hard to catch with the frequency mesh. The `SPECTRAL` keyword can be used in conjunction with band-structure calculations, see Section 5.1.14 for details.

Table 1: Examples

`SPECTRAL`	Write spectral function Im$G(\omega)$ to the file "spectral"; frequency mesh automatically defined.
`SPECTRAL {-10eV:1eV,0.01eV}`	Write spectral function on the specified frequency mesh (relative to the Fermi energy).
`SPECTRAL {-10eV:1eV,0.01eV} 0.002eV`	Bound imaginary part from below by 0.002 eV, preventing peak widths to become too small to be plotted.

5.1.4 *GW* basics

Alhough *GW* calculations can be readily performed with the default settings, the user should be familiar to some degree with the details of the computation and how he/she can influence each step of the program run. Also note that the default settings might work well for one physical system but be unsuitable for another. To lay the basis for the subsequent sections, we briefly recapitulate the basics of the *GW* method.

The *GW* self-energy

$$\Sigma^{\mathrm{xc}}(\mathbf{r},\mathbf{r}';\omega) = \frac{i}{2\pi}\int_{-\infty}^{\infty} G(\mathbf{r},\mathbf{r}';\omega+\omega')\,W(\mathbf{r},\mathbf{r}';\omega')e^{i\eta\omega'}\,d\omega'\,. \tag{5.3}$$

can be understood as a scattering potential that contains the exact exchange potential and correlation effects through the inclusion of W, the dynamically screened interaction, which incorporates the screening of the many-electron system into an effective dynamical potential, obtained from the dielectric function ε through

$$W(\mathbf{r},\mathbf{r}';\omega) = \int \varepsilon^{-1}(\mathbf{r},\mathbf{r}'';\omega)v(\mathbf{r}'',\mathbf{r}')\,d^3r''\,.$$

This integral equation turns into a matrix equation

$$W_{\mu\nu}(\mathbf{k},\omega) = \sum_{\eta}\varepsilon^{-1}_{\mu\eta}(\mathbf{k},\omega)v_{\eta\nu}(\mathbf{k})$$

if the quantities W, ε, and v are represented in the mixed product basis (Section 6.2), which thus has to be converged properly. The dielectric function, in turn, describes the change of the internal potential through screening processes and is related to the polarization matrix by $\varepsilon(\mathbf{k},\omega) = 1 - P(\mathbf{k},\omega)v(\mathbf{k})$ in matrix notation. The polarization function is one of the main quantities in the *GW* scheme. Its evaluation is described in Section 5.2.

The self-energy can be written as the sum of two terms, the first of which is the exact non-local exchange potential of Hartree-Fock theory, the remainder can be interpreted as a correlation self-energy and has the mathematical form of Eq. (5.3) with $W(\omega)$ replaced by $W^{\mathrm{c}}(\omega) = W(\omega) - v$. The frequency integration is carried out analytically for the exchange part (by summing over the residues). The correlation part is more complex to evaluate because of the frequency dependence of the interaction. (Fast electrons experience a different potential than slow electrons.)

There are several ways to represent the self-energy as a function of frequency. The default method is *analytic continuation*, in which the screened interaction and the self-energy are evaluated on a mesh of purely imaginary frequencies. The self-energy is then analytically continued to the complete complex frequency plane ($\Sigma^{\mathrm{xc}}(i\omega) \to \Sigma^{\mathrm{xc}}(z)$, $\omega \in \mathcal{R}$, $z \in \mathcal{C}$). This has several advantages over the usage of the real frequency axis. First, $W(\omega)$ is a hermitian (or real-symmetric) matrix if ω is purely imaginary. Second, W and Σ^{xc} show a lot of structure along the real axis, whereas they are much smoother on the imaginary axis, thereby making it easier to sample and interpolate these functions. Third, after the analytic continuation the self-energy is known, in principle, as an analytic function on the complete complex plane. And fourth, the method requires only few parameters and is, therefore, easy to handle. The main disadvantage lies in the badly controlled extrapolation of the Padé approximants, which can sometimes produce "outlier values", with a potential adverse effect on the accuracy and reliability of the method. Therefore, there is a more sophisticated but also more complex method called *contour integration*, in which the frequency convolution is performed explicitly, yielding the self-energy directly for selected frequencies on the real axis. This method requires the screened interaction W to be calculated in a small range on the real frequency axis, another contribution comes from an integration along the imaginary frequency axis. As a third alternative, there is a *hybrid* method that is nearly as accurate as the "explicit" contour integration method, but it is faster and avoids the calculation of W for real frequencies altogether.

5.1.5 MESH (SENERGY)

All methods to calculate the self-energy employ a mesh of purely imaginary frequencies, which extends from 0 to some maximal $i\omega_{\max}$. The number of mesh points N and the maximal frequency must be provided as parameters, for example `MESH 10 10.0`, which is the default. The mesh is defined by $i\omega_n = i\omega_{\max} f_n/f_N$ with

$f_n = \{(N-1)/[0.9(n-1)] - 1\}^{-1}$, $n = 1, 2, ...$ It is dense for small ω, where the quantities have a lot of structure, and coarse for large ω. Sometimes it is helpful to make the mesh even finer for small ω. This is possible by specifying, for example, `10+3`, which would yield three, two, and one extra equidistant frequencies in the ranges $[\omega_1, \omega_2]$, $[\omega_2, \omega_3]$, and $[\omega_3, \omega_4]$, respectively. If the second argument is defined negative ($-\omega_{\max}$), then $f_n = \{N/(n-1) - 1\}^{-1}$. The latter definition is rarely used. One can also employ a user-defined mesh provided in a file (one frequency per line, comments `# ...` are allowed).

Table 2: Examples

`MESH 12 15.0`	Use a mesh containing twelve frequencies for the range $[0, i15]$ htr.
`MESH 12+2 15.0`	Add points at low frequencies, two (one) between ω_1 and ω_2 (ω_2 and ω_3).
`MESH 12 -15.0`	Use alternative mesh definition.
`MESH "mesh.dat"`	Read frequency mesh from file "mesh.dat".

5.1.6 CONTINUE (SENERGY)

The keyword `CONTINUE` (section `SENERGY`) chooses the analytic continuation method. It can have optional parameters. If given without parameters (or if not specified at all), Padé approximants are used. An integer number, such as `CONTINUE 2`, lets Spex perform a fit to an *n*-pole function (here, *n*=2) with the Newton method. The latter approach is somewhat obsolete by now and recommended only for test calculations. The argument can have additional flags `+`, `c`, and `*`. Any combination is possible. They are explained in the examples.

Table 3: Examples

`CONTINUE`	Use Padé approximants (Default).
`CONTINUE 2`	Fit to the two-pole function $a_1/(\omega - b_1) + a_2/(\omega - b_2)$.
`CONTINUE 2+`	Include a constant in the fit function $a_1/(\omega - b_1) + a_2/(\omega - b_2) + c$.
`CONTINUE 2c`	Take constraints (continuity of value and gradient at $\omega = 0$) into account when fitting.
`CONTINUE 2*`	Allow parameters b_i with positive imaginary parts (should be negative) to contribute. (Default with Padé method.)

5.1.7 CONTOUR (SENERGY)

The second method is *contour integration*, in which the frequency integration is performed explicitly, however not along the real frequency axis but on a deformed integration contour that avoids the real frequency axis as best as possible. This integration contour starts from $-\infty$, describes an infinite quarter circle to $-i\infty$, then runs along the imaginary frequency axis to $i\infty$, and finishes, again with an infinite quarter circle, to ∞. The two quarter circles do not contribute to the integral (because the integrand behaves as $\propto \omega^{-2}$). Furthermore, depending on the frequency argument ω of the self-energy, one has to add a few residues coming from the poles of the Green function in the interval $[0, \epsilon_{\rm F} - \omega]$ if $\omega < \epsilon_{\rm F}$ and $[\epsilon_{\rm F} - \omega, 0]$ otherwise, which requires the correlation screened interaction $W^{\rm c}(\omega)$ to be evaluated on this interval of the real axis. As a consequence, the calculations are more demanding in terms of computational complexity and cost (time and memory). Also, contour integration requires additional input parameters and is therefore somewhat more difficult to use. However, the results are more accurate. In particular, they are not affected by the "ill-definedness" of the analytic continuation.

The corresponding keyword is called `CONTOUR` and belongs to the section `SENERGY`. Obviously, `CONTOUR` and `CONTINUE` must not be given at the same time. The keyword `CONTOUR` expects two arguments. The first defines the frequencies ω, for which the self-energy $\Sigma^{\rm xc}(\omega)$ is to be evaluated. At least, two frequencies are needed to approximate the self-energy as a linear function in ω and, thus, to calculate the ''linearized" solution of the quasiparticle equation [Eq. (5.2)]). For this, a single value suffices (example `0.01`), with which the self-energy for a state $\mathbf{k}n$ is evaluated at two frequencies ($\epsilon_{\mathbf{k}n} - 0.01$ htr and $\epsilon_{\mathbf{k}n} + 0.01$ htr). The more accurate ''direct" solution is only available if we specify a range of frequencies for the self-energy instead of a single number, making an interpolation of the self-energy beyond a linear function possible. This is possible by an argument such as `+-{-0.1:0.15,0.01}`. Here, the range

of values is relative to $\epsilon_{\mathbf{k}n} > \epsilon_{\mathrm{F}}$. The range is reversed (to -0.15:0.1 in the example) for occupied states ($\epsilon_{\mathbf{k}n} < \epsilon_{\mathrm{F}}$) to reflect the fact that occupied and unoccupied states tend to shift in opposite directions by the renormalization. (Thus, it makes sense to choose a frequency range shifted to larger energies instead of a symmetrical one.) One can also specify an absolute frequency mesh, example `{-0.3:0.3,0.01}`, where the energies are given relative to the Fermi energy. Defining such an absolute frequency mesh is mandatory for calculations combining `CONTOUR` with `FULL` (Section 5.1.19).

It is sometimes a bit inconvenient to determine suitable values for the upper and lower bound of `{...}`. Therefore, Spex allows the usage of wildcards for one of the boundaries or both (see examples). The lower (upper) bound is then set at 0.07 htr below (above) the lowest (highest) mean-field band energy defined in the `JOB GW` line. The "margin" of 0.07 htr is added to avoid that the renormalization pushes the quasiparticle energy beyond the frequency grid. It is also possible to define the margin explicity. For example, `{*-0.07:*+0.07,0.01}` would be identical to `{*:*,0.01}`.

The second argument to `CONTOUR` gives the increment of the (equidistant) real-frequency mesh for $W(\omega')$. The lower and upper boundaries of this mesh are determined automatically from the first argument. As a third method, Spex also allows the second argument to be omitted altogether. Then, it uses a *hybrid* method where the screened interaction is analytically continued from the imaginary axis (where it has to be known for the evaluation of the integral along this axis, see above) to the real axis, thereby obviating the need of calculating and storing *W* on a real-frequency mesh. The disadvantage is that the badly controlled Padé extrapolation introduces an element of randomness (also see keyword `SMOOTH` below). Our experience so far is that the *hybrid* method is in-between the two other methods in terms of both computational cost and numerical accuracy.

Table 4: Examples

`CONTOUR 0.01 0.005`	Use contour integration to obtain the two values $\Sigma^{\mathrm{xc}}(\epsilon \pm 0.01\,\mathrm{htr})$ with the KS energy ϵ, giving Σ^{xc} as a linear function.
`CONTOUR +-{-0.1:0.15,0.01} 0.005`	Calculate $\Sigma^{\mathrm{xc}}(\omega)$ on a frequency mesh relative to the KS energy ϵ.
`CONTOUR {*:*,0.01} 0.005`	Use an absolute frequency mesh (relative to the Fermi energy) instead. Wildcards are used for the upper and lower bounds.
`CONTOUR {*:*,0.01}`	Use *hybrid* method.
`CONTOUR {*-0.1:*+0.1,0.01}`	Use *hybrid* method with a different margin of 0.1 htr (see text).

Note: Until version 05.00pre31, the syntax was `{...}` for `+-{...}` and `[{...}]` for `{...}`.

5.1.8 FREQINT (SENERGY)

(*) Independently of whether $\Sigma^{\mathrm{xc}}(i\omega_n)$ (`CONTINUE`) or $\Sigma^{\mathrm{xc}}(\omega_n)$ (`CONTOUR`) is evaluated, an important step in the calculation of the self-energy is to perform the frequency convolution $\int_{-\infty}^{\infty} G(z+i\omega')W(i\omega')d\omega'$ with $z = i\omega_n$ or $z = \omega_n$. For this frequency integration, we interpolate *W* and then perform the convolution with the Green function analytically. The keyword `FREQINT` determines how the interpolation should be done. It can take the two arguments `SPLINE` (default) and `PADE` for spline [after the transformation $\omega' \to \omega'/(1+\omega')$] and Padé interpolation, respectively. In the case of *GT* calculations, there is a similar frequency integration with the *T* matrix replacing *W*. An experimental option is `NONLIN`, which invokes a new tetrahedron k integration method that uses weight functions instead of weight factors. This enables smooth integrations over strongly varying (highly nonlinear) functions, which is particularly useful for the magnetic *T* matrix with its very sharp spin-wave peaks. Therefore, the default for *GT* calculations is `NONLIN`. `FREQINT NONLIN`, in particular, affects the calculation of the residues part in a `CONTOUR` calculation but also the integration along the imaginary frequency axis (both for `CONTINUE` and `CONTOUR`), where it uses Padé interpolation. We note that `FREQINT NONLIN` and, to a lesser extent, also `FREQINT PADE` tend to break energy degeneracies slightly.

Table 5: Examples

`FREQINT SPLINE`	Use spline interpolation for *W* (or *T*) in the frequency convolution $G \cdot W$ ($G \cdot T$) (default for *GW*).
`FREQINT PADE`	Use Padé interpolation.
`FREQINT NONLIN`	Use new tetrahedron k integration method (default for *GT*).

5.1.9 SMOOTH (SENERGY)

(*) We have already mentioned that the Padé approximation can lead to spurious features in the extrapolated or interpolated quantity, i.e., the self-energy or the screened interaction (also see `FREQINT`), if one of the Padé poles happen to lie close to the real (or imaginary) frequency axis. To avoid such unphysical results, we can make use of an averaging procedure where, for each set of values, $\{\Sigma^{\mathrm{xc}}(i\omega_n)\}$ or $\{W(i\omega_n)\}$, we evaluate a large number of Padé approximants with randomly shifted frequency points according to $\omega_n \to \omega_n(1 + 10^{-7} r_n)$ with random numbers $r_n \in [-0.5, 0.5]$ and average over them. This simple but effective way to smoothen the functions is activated with the keyword `SMOOTH` in section `SENERGY`. It can have up to two arguments, giving the number of Padé approximants over which to average, the first for the self-energy, the second for the screened interaction or the *T* matrix in the case of *GT* calculations (`FREQINT PADE`). Smoothening is usually unnecessary for *GW* calculations but is helpful in the case of the *GT* approach. By default, no smoothening is applied. Note that this feature leads to a small degree of randomness in the results.

Table 6: Examples

`SMOOTH 500`	Average over 500 Padé approximants to smoothen the self-energy.
`SMOOTH 500 250`	In addition, average over 250 approximants to smoothen *W* (or the *T* matrix). (Requires `FREQINT PADE`.)

5.1.10 ZERO (SENERGY)

(*) Both the exchange (HF) and the correlation self-energy contain terms of the form $\langle ... \rangle \langle ... \rangle V(\mathbf{k})$ with $V = v$ and $V = W$, respectively. The quantities $\langle ... \rangle$ are the "projections" discussed in Section 6.2. One of the involved projections acquires the form $\langle e^{i\mathbf{k}r} \phi_{\mathbf{q}n} | \phi_{\mathbf{k}+\mathbf{q}n'} \rangle$, which behaves as $\propto k$ in the long-wavelength limit $k \to 0$ and $n \neq n'$. The prefactor is obtained from $k \cdot p$ perturbation theory. Multiplying two of these projections with $V \propto k^{-2}$ (or one with $V \propto k^{-1}$, which is valid for the "wings" of *W*) gives rise to zero-order terms. These zero-order terms (which only appear at the Γ point) can improve the k-point convergence. However, in the case of systems with small band gaps, the zero-order terms can become unnaturally large (because the integrand is strongly varying in the neighborhood of k=0 in such systems), impeding a favourable convergence with respect to the k-point sampling. The keyword `ZERO` (section `SENERGY`) includes the zero-order terms. They are neglected by default.

Note: Keyword `ZERO` replaces `NOZERO` of older versions (until 05.00pre31), which switched off zero-order corrections. So, starting from version 05.00pre32, the default has changed to excluding zero-order corrections.

5.1.11 ALIGNVXC (SENERGY)

(*) Equation (5.1) depends explicitly on the mean-field starting point through the eigensolutions $\{\phi_{\mathbf{k}n}\}$. This dependence is more obvious from Eq. (5.2), which contains the exchange-correlation potential explicitly. Thus, a constant shift $v^{\mathrm{xc}} \to v^{\mathrm{xc}} + \Delta$ will not only shift the quasiparticle energies as a whole (as it would the KS energies) but also individually so that energy differences (e.g., the quasiparticle band gap) can depend on this constant shift. (Again, this is different from the KS gap, which would not be affected.) The underlying reason is that quasiparticle energies represent total-energy differences (between the N and the $N \pm 1$ particle system) and are thus defined as absolute

energies, which depend individually on the "energy zero" set by v^{xc}. One can utilize this dependence to simulate the self-consistency condition that the input and output ionization potentials (valence-band maximum) be identical; in other words, the ionization potential should not change by the quasiparticle renormalization. The keyword `ALIGNVXC` in section `SENERGY` enforces this condition. To this end, the valence-band maximum should be included in the job definition. (Spex uses the highest occupied one among the states defined after `JOB` for the correction.) One can also specify the shift explicitly as a real-valued argument after `ALIGNVXC`. This is useful if a different criterion is to be used, for example, requiring the Fermi energy (metallic case) or core state energies (core spectroscopy) to remain unchanged. In this case, the shift has to be determined manually by repeatedly trying different shift values until the constancy of the desired quantity is achieved (because the shift affects the results, so there is a feedback effect). In this trial-and-error approach, it is very helpful to use `RESTART 2` (Section 5.1.13), in which case the self-energy is read from harddisc and the calculation takes very little time.

Table 7: Examples

`ALIGNVXC`	Align exchange-correlation potential in such a way that the ionization potential remains unchanged by the quasiparticle correction.
`ALIGNVXC 0.2eV`	Apply a constant positive shift of 0.2 eV to the exchange-correlation potential.

5.1.12 VXC (SENERGY)

(*) The matrix elements of the exchange-correlation potential are needed in the solution of Eq. (5.2). By default, these matrix elements are calculated by Spex using the potentials from the input data. Alternatively, they can be taken from data provided by the DFT code (e.g., in the file "vxcfull" written by Fleur). For that, the keyword `VXC` in section `SENERGY` can take the arguments `CALC` (default) and `READ`. (In some cases, `VXC READ` is not available, e.g., for PBE0 calculations.)

Table 8: Examples

`VXC READ`	Read v^{xc} expectation values from harddisc.
`VXC CALC`	Calculate v^{xc} expectation values using the xc potential defined in stars and lattice harmonics.

5.1.13 RESTART

Spex can reuse data from a previous *GW* run that has finished successfully, crashed, or has been stopped by the user. (See Section 4.1.7 for more details.) A *GW* calculation consists mainly of a loop over the irreducible k points. For each k point, Spex (a) calculates the matrix $W(\mathbf{k})$ and (b) updates the self-energy matrix (or expectation values) Σ^{xc} with the contribution of the current k point (and its symmetry-equivalent k points). After completion of step (b), the current self-energy is always written to the (binary) files "spex.sigx" and "spex.sigc". If the `RESTART` option is specified (independently of its argument), Spex also writes the matrix $W(\mathbf{k})$ (in addition to some other data) to the (HDF5) file "spex.cor" unless it is already present in that file. If it is present, the corresponding data is read instead of being calculated. In this way, the keyword `RESTART` enables reusage of the calculated $W(\mathbf{k})$ from a previous run. The matrix $W(\mathbf{k})$ does not depend on the k points and band indices defined after `JOB`. So, these parameters can be changed before a run with `RESTART`, in which the *W* data is then reused. For example, band structures can be calculated efficiently in this way (Section 5.1.14). Especially for long calculations, it is recommended to use the `RESTART` option. Spex can also restart a calculation using self-energy data contained in "spex.sigx" and "spex.sigc". To this end, the argument 2 is added: `RESTART 2`. Spex then starts with the k point, at which the calculation was interrupted before. In contrast to "spex.cor", the files "spex.sigx" and "spex.sigc" do depend on the job definition, which must therefore **not** be changed before a run with `RESTART 2`. However, there are few parameters (see below) that may be changed before a rerun with `RESTART 2`. These concern details about how the quasiparticle equation is solved after completion of the self-energy calculation. The following logical table gives an overview.

	spex.cor	spex.sigx/c
–	–	write
RESTART	read-write	write
RESTART 2	read-write	read-write

The different rules for "spex.cor" and "spex.sigx/c" are motivated by the facts that (a) the file "spex.cor" is much bigger than "spex.sigx/c" (so, writing of "spex.cor" to harddisc should not be the default), (b) the files "spex.sigx/c" include the updated self-energy (requiring more computation than for W, thus representing "more valuable" data).

There are other (binary) files that are written during a program run and that may be reused later. The ones relevant for *GW* (same rules as for "spex.cor") are

- spex.mb: contains MT part of product basis,
- spex.ccou: contains "contracted" screened interaction (i.e., summed over $\mathbf{k}$); this contraction is needed for the self-energy core contribution (CORES), for COSX, and for IBC,
- spex.core: contains core contribution to W (keyword CORES, Section 5.1.18).

The RESTART option can be exploited to customize the calculations. For example, it is possible to perform a self-consistent QSGW calculation while keeping W fixed to the LDA level throughout the iterations. (The screened interaction is then always read from "spex.cor" and the MT product basis from "spex.mb".) As another example, when CORES is used, the MT product basis additionally contains products of core states with LAPW basis functions. To avoid that, one can run a calculation without CORES first (which can be stopped just after the product basis is written to "spex.mb"), followed by a calculation with RESTART. The product basis is then read from "spex.mb" (excluding the core-basis products) instead of being constructed anew.

As already mentioned above, changing input parameters in "spex.inp" and running a calculation with RESTART might lead to a conflict between the changed parameters and the data read from the restart files. Spex tries to detect such conflicts. In case of conflict, Spex either stops the calculation ("spex.cor") or recalculates the data and overwrites the files ("spex.sigx/c"). It is, however, not guaranteed that Spex can detect all possible conflicts. So, care has to be taken if parameters are changed in calculations with RESTART.

The following is a (incomplete) list of parameters that may be changed for a calculation with RESTART 2, in which the files "spex.sigx/c" are read: ALIGNVXC, VXC, SPECTRAL, SMOOTH, CONTINUE, WRITE, all of section WANNIER (e.g., for Wannier interpolation). In the case of RESTART 1 (or just RESTART) (i.e, when "spex.cor" is to be read), one may change most parameters except: MESH, 2nd argument of CONTOUR, JOB types (most cases). Many changes of parameters will be overridden with values from the restart files (e.g., parameters in the section MBASIS by data from the file "spex.mb"). Other changed parameters might affect only the self-energy but not the screened interaction W, for example, NBAND. (The latter allows the band convergence of W and Σ^{xc} to be checked separately.) Again, this can be used efficiently to customize a calculation.

Table 9: Examples

RESTART	Spex tries to read the screened interaction W (and other data) from the file "spex.cor". If the required data does not exist, the calculation proceeds normally and appends W to the file (reusable in a later restart).
RESTART 1	Same as RESTART.
RESTART 2	Implies RESTART. Spex reads the self-energy from the files "spex.sigx" and "spex.sigc" if they exist. If not, the calculation proceeds normally.

5.1.14 *GW* band structures

In this section, three ways of calculating *GW* band structures are discussed.

- Using the KPTPATH label (single Spex run, Section 5.1.15),

- With the special + label in KPT for additional q points (multiple Spex runs, Section 5.1.16),
- Wannier interpolation (single – but long – Spex run, Section 5.1.17).

The calculation of a *GW* band structure is not as straightforward as in the case of KS-DFT, where the local nature of the effective potential (e.g., LDA or GGA) allows the KS Hamiltonian to be diagonalized independently for each q point along a q-point path. The *GW* self-energy, on the other hand, is non-local. As a consequence, its evaluation involves a convolution in reciprocal space $\Sigma^{\mathrm{xc}}(\mathbf{q}) = i \sum_{\mathbf{q}} G(\mathbf{q}+\mathbf{k})W(-\mathbf{k})$ (in simplified notation), requiring summations over the full Brillouin zone for any arbitrarily chosen q point. So, in addition to $\mathbf{q}$, one would need to include all shifted points $\mathbf{q}+\mathbf{k}$, as well, where $\mathbf{k}$ stands for the k points defined in BZ.

If we restrict ourselves to the set of k points defined in BZ, the band structure along a certain high-symmetry line (e.g., $\Gamma - X$) can consist only of the few k points that happen to lie on this line. Sometimes this is already sufficient. For example, if the *GW* renormalization affects the band dispersion only little, then a few points along the high-symmetry line are enough to shift/modify the KS bands accordingly and produce, in this way, a quasiparticle band structure. Such a calculation can be performed in a single Spex run.

5.1.15 KPTPATH

As an example, we want to plot a band structure for the path from L over Γ to X for our Si example. In principle, we would have to identify all k-point indices along this path and set the job definition accordingly, but Spex can do this for you. The k-point path can be defined with a line `KPTPATH (L,1,X)`. (A user-defined set of k points can also be provided through a file, see examples.) In the job definition, one can then use the special label PATH to address the corresponding list of k points, e.g., `JOB GW PATH:(1-10)`. Spex will automatically choose the k points that lie on the path defined by KPTPATH and calculate the *GW* quasiparticle energies. As usual, the results are written to the output file. The data can be extracted from the output file with the spex.extr utility: `spex.extr g -b spex.out > bandstr` (assuming that the output has been piped into the file "spex.out"). The file "bandstr" then contains a plottable list of xy values (momentum/energy) for each quasiparticle band. However, in most cases the band structure will not be smooth enough because of the small number of k points. One solution is, as already mentioned, to modify a KS band structure. Another possibility is to interpolate the energies with splines; the spex.extr utility has this capability. Finally, one could simply use much denser k-point sets, but this would entail very expensive calculations.

The keyword KPTPATH can also be used for the other two methods to calculate a band structure (see Section 5.1.16 and Section 5.1.17). In these methods, one is not restricted to the k points of the k set and can define k-point paths with arbitrary step sizes. To this end, an integer argument can be specified at the end of the k-path definition. For example, `KPTPATH (L,1,X) 50` would give a k path with a step size of (roughly) $K/50$, where K is the "average reciprocal lattice constant", defined as $K = 2\pi/\sqrt[3]{\Omega}$ with Ω the volume of the unit cell. The actual step size between the defined k points (L, Γ, and X in the example) is adjusted in such a way that these k points lie on the path (hence the use of the word "roughly" above). Larger integer arguments give denser k meshes. The default is 20 for the method explained in Section 5.1.16 and 100 for the one in Section 5.1.17.

Table 10: Examples

`KPTPATH (L,1,X)`	Define k-point path from L over the Γ point (k index is always 1) to X. The labels L and X must be defined with KPT.
`JOB GW PATH:(1-10)`	Run *GW* calculation for the first ten bands at all k points defined by KPTPATH.
`KPTPATH (1,N) 100`	Define k-point path from Γ to N and use step size of $2\pi/(100\sqrt[3]{\Omega})$.
`KPTPATH "my_kpath" 100`	Read k-point path from file "my_kpath".

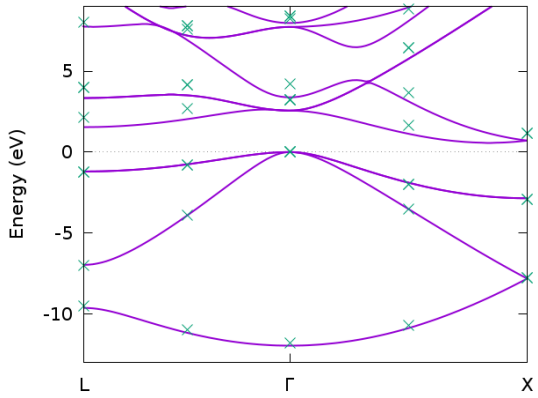

Fig. 1: Band structure for bulk silicon from DFT (solid line) and *GW* (symbols) calculated with `JOB GW PATH:(1-12)`.

5.1.16 KPT +=...

This section explains an extension to the keyword `KPT` (Section 5.1.2), which will later enable the calculation of smooth band structures. The extension enables adding arbitrary q points, one at a time, allowing one to investigate the quasiparticle spectrum at momenta that are not element of the original k-point set (defined by `BZ`). For example, in Si the conduction-band minimum is not at a high-symmetry k point but at around 3/4 along the line $\Gamma - \mathrm{X}$. This particular q point can be defined using the special k label "+" by `+=[0,0,0.75]` (alternatively, `+=3/8*(1,1,0)` in internal coordinates) after the keyword `KPT`. As explained above, each additional point $\mathbf{q}$ must in principle be accompanied with all shifted points $\mathbf{q}+\mathbf{k}$. So, before running Spex, we must let the DFT code generate the wavefunctions and energies at the shifted k-point set $\{\mathbf{q}+\mathbf{k}\}$. This is done in the usual way by creating the k-point file (Section 4.1.2) and running the DFT code. (In the case of Fleur, the shifted k points are appended to the list in the file "kpts".) The additional q point can be addressed in the job definition with the special "+" label, e.g., `JOB GW 1:(1-5) +:(1-5)`, which will yield the bands 1-5 for the Γ point and the additional q point. The energy difference between the fourth state at Γ and the fifth state at $\mathbf{q}$ yields the fundamental *GW* gap.

Table 11: Examples

`KPT +=1/7*[1,0,0]`	Define special k point at 1/7 along the line $\Gamma - \mathrm{X}$ (here, the X point in x direction) of an fcc structure.
`KPT +=1/14*(0,1,1)`	The same in internal coordinates.
`JOB GW +:(1-10)`	Run *GW* calculation for the states 1-10 at the added q point.

The possibility of adding arbitrary q points enables the calculation of smooth band structures. To this end, a *GW* run (together with the generation of the eigenstates with the DFT code) has to be performed for each q point in a list of q points that make up a path in the Brillouin zone, e.g., from Γ to X. This list is defined with the keyword `KPTPATH` as explained above. Fortunately, the screened interaction has to be calculated only once and can be reused for the other points. For this, the keyword `RESTART` should be given in the input file. Of course, we also need a job definition, such as `JOB GW +:(1-10)`. Here, only the additional q point should be specified. It would be difficult to extract the data for the band structure plot otherwise.) Now, running Spex with `KPTPATH` defined in the input file and `WRTKPT` (or with the command line option `-w`) creates two files, containing the usual list of k points (Fleur: "kpts") and the list of q points (Fleur: "qpts"). For each of the q points, we would have to run, in this order, Spex, Fleur, and Spex again. To simplify this task, there is a shell script called "spex.band". This shell script performs all the necessary steps automatically. It uses the same environment variables as "spex.selfc" (Section 5.1.19) and produces, for each q point, one output file named "spex_NNN.out", where NNN is a three-digit counting index, or more digits if the index exceeds 999.

From these files the band-structure data is extracted with `spex.extr g -b spex_???.out > bandstr`. The

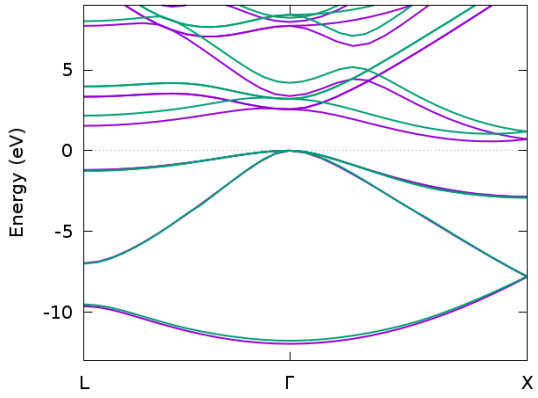

Fig. 2: Band structure for bulk silicon from DFT (magenta) and *GW* (green) calculated with multiple runs of Spex using the "spex.band" shell script.

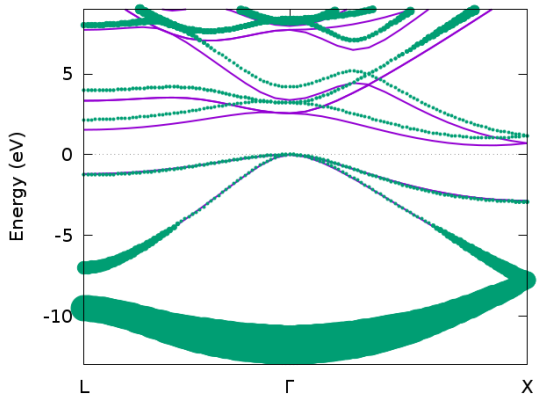

Fig. 3: Band structure for bulk silicon from DFT (magenta) and *GW* (green) with lifetime broadening.

data is written to the file `bandstr`. The band structure can then be plotted with "xmgrace" or "gnuplot". If you extract the band-structure data, instead, with `spex.extr g -b -c spex_???.out > bandstr`, you get an additional column, which contains the imaginary parts of the quasiparticle energies. These imaginary parts are proportional to the line width and inversely proportional to the excitation lifetime. The line widths can be plotted together with the band structure in "gnuplot" with `plot "bandstr" using 1:2:(10*abs(3)) with points ps var`. You will see that the bands get wider the further they are away from the Fermi energy.

If "spex.inp" contains a `SPECTRAL` definition (Section 5.1.3), the script "spex.band" also renames the file "spectral", written in each Spex run, to "spectralNNN" with the same counting index NNN as above. The spectral data can be compiled with the "spex.extr" utility (e.g., `spex.extr s -b spectral??? > spec`) into a single data file containing a list of xyz value triples: the Bloch momenta as the first column, the frequency as the second, and the value of the spectral function as the third. With "gnuplot", for example, a momentum-frequency plot of the spectral function with color coding is then prepared with the command `plot "spec" using 1:2:3 palette with lines`. (The respective color range is set with `set cbrange [...]`, see the Gnuplot manual.)

5.1.17 INTERPOL

As a third alternative, we can use Wannier interpolation to interpolate between the few k points of the original k mesh. The construction of Wannier orbitals is explained in Section 6.3. Here, we use a minimalistic definition for demonstration purposes. We have to modify and add some lines to `spex.inp`:

```
JOB GW IBZ:(1-4)
SECTION WANNIER
  ORBITALS 1 4 (sp3)
  MAXIMIZE
  INTERPOL
END
```

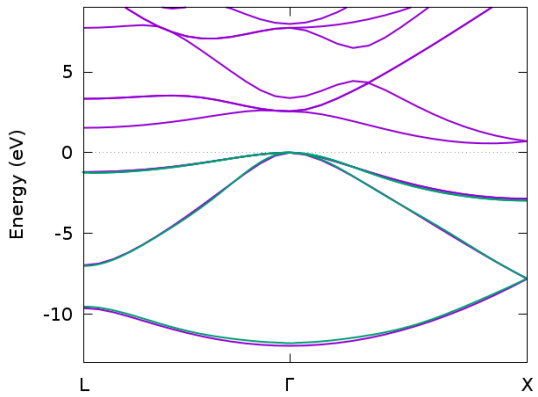

Fig. 4: Band structure for bulk silicon from DFT (magenta) and *GW* (green) calculated with Wannier interpolation (only filled bands).

Please also remove the entry `+=[0,0,0.75]` from the `KPT` line. The first line lets SPEX calculate quasiparticle energies in the whole irreducible Brillouin zone (IBZ). (The Wannier interpolation can be understood as a back-and-forth Fourier transformation with a real-space truncation of matrix elements in-between. The Fourier transformation from momentum to real space involves the band energies in the whole BZ.) The section `WANNIER` defines a set of maximally localized Wannier functions (MLWFs) of bonding sp^3 hybrid orbitals. These four hybrid orbitals are generated from the four lowest valence bands (first two arguments after `ORBITALS`). Running Spex will construct the MLWFs and use them to interpolate the band structure. The band structure data is written to the files "bands0" for the KS energies and "bands1" for the *GW* quasiparticle energies and can be plotted with "xmgrace" or "gnuplot". The file "bands1" has one data column more than "bands0". This additional column contains the interpolated imaginary parts

of the quasiparticle energies. You can plot the band structure with the line thickness scaled by the imaginary part in the same way as described before.

The band structure is calculated along the k-point path defined by `KPTPATH` (Section 5.1.15). Alternatively, one can specify a file after `INTERPOL`, e.g., `INTERPOL "qpts"`. The k-point path is then taken from that file. (The file structure is the same as that of the file "qpts" written by `spex -w` when `KPTPATH` is defined. 1st line: number of k points, common denominator; 2nd, 3rd, ... lines: k points defined as three-tuples.)

Wannier interpolation can also be used in conjunction with the keyword `SPECTRAL` (Section 5.1.3). The interpolated spectral function is written to the file "spectralw" or, if the system is spin-dependent, to the files "spectralw1" and "spectralw2" for the spin up and spin down channels, respectively. These files can be plotted in the same way as described in Section 5.1.16.

The plot of "fat bands" (orbital projections) is possible in combination with the keyword `PROJECT` (Section 4.2.3). By default, the projection onto Wannier orbitals ("Wannier projections") are given for each state in the output file (e.g., "bands0").

Further details about Wannier functions are found in Section 6.3 and some practical advice on Wannier interpolation in Section 6.3.7.

5.1.18 CORES

By default, the *n* summation in the Green function

$$G_0(\mathbf{r},\mathbf{r}';\omega) = \sum_{\mathbf{k}}\sum_{n} \frac{\phi_{\mathbf{k}n}(\mathbf{r})\phi^*_{\mathbf{k}n}(\mathbf{r}')}{\omega - \epsilon_{\mathbf{k}n} + i\eta\,\mathrm{sgn}(\epsilon_{\mathbf{k}n} - \epsilon_\mathrm{F})} \tag{5.4}$$

for the self-energy [Eq. (5.3)] includes only the valence and conduction states represented in the LAPW basis but excludes the core states. In theory, of course, the core states should be included. Their contribution is, however, numerically small. Spex allows selective core states to be included in the evaluation of the self-energy with the keyword `CORES`. This keyword also affects the calculation of the polarization function, see Section 5.2.4. As arguments, Spex expects a comma separated list of core states between round brackets, e.g., `(2s,2p,3d)`, for each atom type. *Empty* definitions `()` are allowed. If `CORESOC` is specified, one can also address the SOC-splitted states individually, e.g., `(2p1/2)`.

Table 12: Examples

`CORES () (1s)`	Include the 1s state of the second atom type.
`CORES (2s,2p1/2)`	Include the 2s and the $2p^{1/2}$ states but exclude the $2p^{3/2}$ states.

5.1.19 Beyond perturbative *GW* and QSGW

One can also go beyond the perturbative solution of Eq. (5.1) and solve this equation by iterative diagonalization in the basis of the single-particle states. This requires the full self-energy matrix including off-diagonal elements to be calculated. The respective line in the input file would read `JOB GW FULL 1:(1-10) X:(1-10) L:(1-10)` giving the self-energy as 10×10 matrices at the Γ, X, and L points. (In principle, it is also possible to have a band definition like `(1-5,9,10)`, which would yield a 7×7 matrix. Degenerate states, not included in the definition, are automatically included.) In `FULL` calculations, the Spex code uses irreducible representations (irreps) to speed up the calculation. As a result, the bands are grouped in terms of these irreps (called "blocks") in the output. For example, in the example below (for the X point) the block is composed of the bands 3, 4, 9, and 10. If a block contains only a single band or a single subgroup of degenerate states, the full solution is omitted because it would be identical to the perturbative (diagonal) solution. The full solutions are tabulated in the output:

```
Bd        KS      olap        HF      olap        GW
 3   2.92306   0.99971   0.12683   0.99985   2.50496  -0.06006
 4   2.92306   0.99971   0.12683   0.99985   2.50496  -0.06006
 9  16.77136   0.99732  23.25660   0.99962  16.97051  -0.30294
10  16.77136   0.99732  23.25660   0.99962  16.97051  -0.30294
```

The values with the label `HF` are the Hartree-Fock values (from diagonalization). They are **not** necessarily ordered according to energy but related to the KS states (with band indices labeled "Bd") with a condition of maximal overlap (and at the same time making sure that a one-to-one correspondence can be established). The corresponding overlap is given on the left of the HF column. The full *GW* energies (column labeled `GW`; the last column lists the imaginary parts) are related to the KS states as well. This is due to the fact that the quasiparticle equation is non-linear in energy and must, therefore, be solved iteratively for each quasiparticle state. The iterations are started with the corresponding KS energy $\epsilon_{\mathbf{k}n}$, giving $H^{\mathrm{KS}} + \Sigma^{\mathrm{xc}}(\epsilon_{\mathbf{k}n}) - v^{\mathrm{xc}}$ as the (complex) quasiparticle Hamiltonian. The diagonalization then yields a spectrum of states, of which the one with the largest overlap is picked for the next iteration. In the first few iterations, the off-diagonal elements are switched on adiabatically to alleviate a smooth convergence. The overlap of the final quasiparticle wavefunction with the original KS state is given, too. Although, this approach usually yields well-defined quasiparticle solutions, it cannot be ruled out that the iterative procedure, when started at two different energies, might result in the same quasiparticle energy. This can happen especially for states far away from the Fermi energy, which do not have a well-defined quasiparticle character anymore.

If the job definition contains `FULL` and `IBZ`, the full *GW* self-energy matrix is evaluated for the whole IBZ, which enables self-consistent calculations in the framework of the quasiparticle self-consistent *GW* (QSGW) approach. In this approach, one creates a mean-field system from the *GW* self-energy whose single-particle energies are as close as possible to the quasiparticle energies. This mean-field system is subsequently solved to self-consistency in a DFT code. The resulting solution can then serve as a starting point for a new one-shot *GW* calculation, which constitutes the second iteration and so on until the quasiparticle energies are converged. The construction of the mean-field system is, to some extent, arbitrary. We use the following definition, which is slightly modified from the original work [Phys. Rev. Lett. 93, 126406]:

$$A_{\mathbf{k}nn} = Z^{-1}_{\mathbf{k}n} \langle \phi_{\mathbf{k}n} | \Sigma^{\mathrm{xc}}(\epsilon_{\mathbf{k}n}) | \phi_{\mathbf{k}n} \rangle$$

for diagonal elements and

$$A_{\mathbf{k}nn'} = \langle \phi_{\mathbf{k}n} | \Sigma^{\mathrm{xc}}(\epsilon_{\mathbf{k}n}) + \Sigma^{\mathrm{xc}}(\epsilon_{\mathbf{k}n'}) | \phi_{\mathbf{k}n'} \rangle$$

for off-diagonal elements. The *hermitianized* QSGW operator is then obtained from $\Sigma^{\mathrm{xc,H}} = (A + A^\dagger)/2$. The difference to the original definition is the inclusion of the renormalization factor to better reproduce the *GW* quasiparticle energies. The *hermitianized* matrix, or rather the difference $\Sigma^{\mathrm{xc,H}} - v^{\mathrm{xc}}$, is written to the file "spex.qsgw", which is later read by the DFT code. The exact procedure of a QSGW calculation depends on the DFT code used.

As an example, we explain the necessary steps for a calculation with Fleur v0.26b (2019.03):

- `rm fleur.qsgw` - remove the file "fleur.qsgw" containing the *hermitianized* matrix used internally by Fleur from previous QSGW iterations.
- `rm broyd*` - remove Broyden information about previous iterations because this information is inconsistent with the new Hamiltonian (the SCF calculation does not converge otherwise).
- Set `gw=3` in the Fleur input file.
- **Run Fleur.** Fleur translates the file "spex.qsgw" (in basis of eigenstates) to a file "fleur.qsgw" (in LAPW basis). Then, a SCF run is performed where, instead of the KS Hamiltonian H^{KS}, the Hamiltonian $H^{\mathrm{KS}} + (\Sigma^{\mathrm{xc,H}} - v^{\mathrm{xc}})$ is employed. After the run finishes, the `gw=2` step has to be performed.
- Set `gw=2` in the Fleur input file.
- Run Fleur.

Apart from the usual output data for a *GW* calculation, Fleur writes, in addition, the file "qsgw", which contains the matrix elements of $\Sigma^{\mathrm{xc,H}} - v^{\mathrm{xc}}$ in the basis of the new eigenstates. The file "qsgw" is later read by Spex to subtract the double counting. Of course, doing these steps manually is tedious and error-prone. Therefore, we provide the shell script `spex.selfc`, which performs the steps needed for a self-consistent calculation automatically. For example, `spex.selfc 5` will run five iterations. The shell script assumes that Spex and Fleur are run with `spex` and `fleur.x`, respectively. Environment variables can be used to change this:

- `SPEX_EXEC` - Spex executable,
- `DFT_EXEC` - Fleur executable,
- `SPEX_THRU` - Command to prepend Spex executable (e.g., queue command or MPI launcher such as `mpiexec -np 20`),
- `DFT_THRU` - Command to prepend Fleur executable.
- `THRU` - Default command to prepend Spex and Fleur executables.

The job types `HF`, `PBE0`, `SX`, and `COSX` also allow `FULL` in the definition, i.e., a full matrix is evaluated, enabling a self-consistent calculation in the same way as QSGW (see above). The file names used are the same: "spex.qsgw", "fleur.qsgw", "qsgw", even though it might actually be a HF calculation. The job `GT` (also `GWT`) does not allow for self-consistency at the moment.

5.2 Polarization function

The polarization function gives the linear change in the electronic density of a non-interacting system with respect to changes in the effective potential. It is, thus, a fundamental quantity in the calculation of screening properties of a many-electron systems. For example, the dielectric function, instrumental in the calculation of spectroscopic quantities (e.g. EELS) and the screened interaction needed in *GW*, is related to the polarization matrix through $\varepsilon(\mathbf{k},\omega) = 1 - P(\mathbf{k},\omega)v(\mathbf{k})$, here given in matrix notation. The corresponding explicit formula for matrix elements of *P* in the mixed product basis is

$$\begin{aligned} P_{\mu\nu}(\mathbf{k},\omega) &= 2\sum_{\mathbf{q}}^{\mathrm{BZ}}\sum_{n}^{\mathrm{occ}}\sum_{n'}^{\mathrm{unocc}} \langle M_{\mathbf{k}\mu}\phi_{\mathbf{q}n}|\phi_{\mathbf{k}+\mathbf{q}n'}\rangle\langle\phi_{\mathbf{k}+\mathbf{q}n'}|\phi_{\mathbf{q}n}M_{\mathbf{k}\nu}\rangle \\ &\quad\cdot\left(\frac{1}{\omega+\epsilon_{\mathbf{q}n}-\epsilon_{\mathbf{q}+\mathbf{k}n'}+i\eta} - \frac{1}{\omega-\epsilon_{\mathbf{q}n}+\epsilon_{\mathbf{q}+\mathbf{k}n'}-i\eta}\right) \\ &= \int_{-\infty}^{\infty}\frac{S_{\mu\nu}(\mathbf{k},\omega')}{\omega-\omega'+i\eta\mathrm{sgn}(\omega')}d\omega' \,. \end{aligned} \tag{5.5}$$

A system without spin polarization has been assumed, hence the factor 2. In case of spin polarization, the formula has a spin summation instead of the factor 2 and corresponding spin quantum numbers. In the case of spin-orbit coupling (and no spin polarization), two spin summation labels are added, one in front of each "projection" $\langle\ldots\rangle$.

We have implicitly defined the spectral function *S* in Eq. (5.5), an explicit expression for which is basically given by the formula in the middle with the $1/(\omega\ldots)$ replaced by $\delta(\omega\ldots)$. (Technically, the $M_{\mathbf{k}\mu}(\mathbf{r})$ form the eigenbasis of the Coulomb matrix, so they are linear combinations of the mixed product basis functions.)

The expression for Eq. (5.5) involves an infinite empty-band summation over n'. In practice, this summation is truncated by the number of bands (keyword `NBAND`), which thus becomes an important convergence parameter. The self-energy contains an infinite band summation as well.

In principle, the k summation is infinite, too: In an infinite crystal, there are infinitely many k points, and the k summation formally turns into a k integration. In practice, of course, there can only be a discrete sampling of the Brillouin zone. The tetrahedron method enables a geometrical interpolation between the explicit k points and, thus, an approximate k integration. Therefore, the tetrahedron method is the default and recommended method. However, there are two other implemented methods: Simple summation over the k-point mesh (see `HILBERT NONE`) and the Gaussian method (see `GAUSS`).

5.2.1 HILBERT (SUSCEP)

The most important keyword in the calculation of the polarization function is `HILBERT`, which defines a (Hilbert) mesh for the frequency integration in Eq. (5.5). (The old keyword `FSPEC` still works but is deprecated.) The Hilbert mesh, used in conjunction with the tetrahedron and the Gaussian method, extends from the lowest (roughly the band gap) to the highest virtual electron transition energy. (Note that the spectral function is symmetric $S(\omega) = S(-\omega)$, which makes a mapping to $\int_0^\infty ...$ possible.) The lower and upper bounds are, thus, predetermined by the input data. As the spectral function usually shows more variation at low energies (also, these energies are more important from a perturbation theory point of view), we employ an exponential mesh, which is dense at low and coarse at high energies. Two parameters are necessary to define the Hilbert mesh $\{w_i\}$, for example, the first step size $\omega_2 - \omega_1$ and the stretching factor $a = (\omega_{i+1} - \omega_i)/(\omega_i - \omega_{i-1})$. These two parameters are accepted as real-valued arguments (example: `HILBERT 0.01 1.04`). However, with this definition it is not straightforward to increase the density of the mesh without changing the exponential form of the mesh. Therefore, it is recommended to use a different definition, namely using an integer as first argument (without a period!) and a real number as second (example: `HILBERT 30 40.0`; the second argument is always interpreted as real-valued, so `HILBERT 30 40` is equivalent). The first argument gives the number of mesh points between 0 and 5 htr and the second argument is the "accumulated" stretching factor at 5 htr, i.e., by what factor the step size at 5 htr has increased from $\omega_2 - \omega_1$. (The reason for defining an arbitrary but fixed energy of 5 htr is that modifying the energy range, e.g., by changing `NBAND`, then leaves the form of the Hilbert mesh unchanged.) In this way, increasing the first argument makes the mesh denser but does not affect the exponential form of the mesh, which, in turn, is governed by the second argument. So, there are two possible definitions (distinguished by whether the first argument has a period or not). For convenience, Spex always writes the equivalent parameters of the other definition to the output. By default, Spex uses a Hilbert mesh defined by `HILBERT 50 30.0` for band-gap materials and `HILBERT 150 50.0` for metals, unless a spectral frequency mesh is defined (e.g., `DIELEC {0:1,0.01}`), in which case Spex adjusts the Hilbert mesh to this frequency mesh. An idea of how well a given Hilbert mesh resolves the spectral function can be got by specifying `WRITE INFO` in the input file. Then, a file "spex.sf.nnn" (where nnn is a three-digit counting index) is written for each k point, and it contains the spectral function for the head element of *P* on the Hilbert mesh as plottable data. One can also define `HILBERT NONE` as a special option, which implements the k summation as simple sums over the k points without interpolation or broadening. `HILBERT NONE` is only available if the polarization function is to be calculated for purely imaginary frequencies (including zero).

Table 13: Examples

`HILBERT NONE`	Simple summation over k points.
`HILBERT 50 30.0`	Use Hilbert mesh with fifty frequencies (between 0 and 5 htr) and an accumulated stretching factor of 30 at 5 htr.
`HILBERT 0.01 1.05`	Use Hilbert mesh with a first step size of $\omega_2 - \omega_1 = 0.01$ htr and a stretching factor of $a = 1.05$. (First argument is real-valued.)

5.2.2 MULTDIFF (SUSCEP)

(*) In the limit $k \to 0$, the projections in the numerator of Eq. (5.5) approach linearly to zero. However, when calculating the dielectric function, one has to multiply with $\sqrt{4\pi}/k$ (square root of Coulomb matrix) in this limit. So, the first order of $\langle e^{i\mathbf{kr}}\phi_{\mathbf{q}n}|\phi_{\mathbf{k}+\mathbf{q}n'}\rangle$ (corresponding to $\mu = 1$) in k becomes relevant. Using $k \cdot p$ perturbation theory, one can show that the linear term is proportional to $(\epsilon_{\mathbf{q}n'} - \epsilon_{\mathbf{q}n})^{-1}$. This can lead to numerical problems if the two energies are very close to each other. Therefore, when treating the Γ point ($k = 0$), Spex multiplies the linear term with this energy difference, resulting in smooth and non-divergent values, and takes the energy difference into account by replacing $S(\omega) \to S(\omega)/\omega$ in the frequency integration of Eq. (5.5), thereby avoiding the numerical difficulties. By default, Spex does that only at $k = 0$. The behavior can be changed with the keyword `MULTDIFF` in the section `SUSCEP`. As an alternative, the energy differences can also be incorporated into the integration weights, which is arguably even more stable numerically, see option `INT` below.

Table 14: Examples

`MULTDIFF OFF`	Never separate (divergent) energy differences.
`MULTDIFF ON`	Always separate energy differences (i.e., for all k points).
`MULTDIFF INT`	Use default behavior (separate for $k = 0$, do not for $k \neq 0$) but stick energy differences into integration weights.
`MULTDIFF INTON`	Always separate energy differences by sticking them into integration weights.

5.2.3 PLASMA (SUSCEP)

(*) In the case of metals, the polarization function contains an additional term, the so-called Drude term, which gives rise to metallic screening (diverging static dielectric constant, short-range static *W*). The Drude term stems from virtual intraband transitions across the Fermi surface. It can be formulated with the square of the plasma frequency, which in turn is evaluated by an integration over the Fermi surface. The Drude term can be treated analytically and, as long as the Fermi surface is sufficiently big, it normally does not pose a numerical problem. However, a very small Fermi surface eventually leads to a very sharp "Drude peak" in the *GW* self-energy, impeding a straight-forward numerical solution of the non-linear quasiparticle equation. One could also say that, while the Drude term is actually treated correctly, it gets too much weight because of the coarseness of the k-point mesh. It is, therefore, sometimes helpful to modify the plasma frequency. This is possible with the keyword `PLASMA`. Its argument replaces the plasma frequency calculated by the code. Setting `PLASMA 0` switches the Drude term off altogether. Of course, `PLASMA 0` thus also disables metallic screening, which might be unphysical. Therefore, there is a special option, `PLASMA METAL`, which scales the head element of $W(\mathbf{k}, \omega)$ in the limit $k \to 0$ to enforce metallic screening. This latter option can be helpful, for example, in the case of semimetallic systems with a tiny Fermi surface.

Table 15: Examples

`PLASMA 1.5eV`	Set plasma frequency manually to 1.5 eV.
`PLASMA 0`	Set plasma frequency to zero. Disables Drude term.
`PLASMA METAL`	Disable Drude term but enforce metallic screening by scaling the head element of *W*.

5.2.4 CORES

By default, the *n* summation of Eq. (5.5) extends only over the valence states (represented in the LAPW basis) and excludes the core states. The core states can be included in this summation in the same way as for the *GW* self-energy (see Section 5.1.18). If `CORESOC` is specified, the SOC-splitted core states are included, otherwise the SOC-averaged core states (see Section 4.2.2). In the former case, the Weiss field of a spin-polarized system can further lift the degeneracies. This is taken into account as well. See Section 5.1.18 for details and examples.

5.2.5 TETRAF (SUSCEP)

(*) In rare cases, especially for very large k-point sets, the determination of the tetrahedron weights (Timing `wghts`) can become computationally expensive. This is because Spex calculates tetrahedron weights for all k points by default, which makes it possible to average over equivalent k points, thereby avoiding a (slight) symmetry breaking connected with the spatial form of the tetrahedra. (The tetrahedra are not unique!) One can disable this averaging by setting `TETRAF` in the section `SUSCEP`. If set, tetrahedron weights are only calculated for the irreducible part of the BZ. This accelerates the calculation of the weights but introduces slight deviations due to symmetry breaking.

5.2.6 WGHTTHR (WFPROD)

(*) In both tetrahedron and Gaussian methods, integration weights are calculated to perform the k summation. To be more precise, an integration weight is calculated for each virtual transition, i.e., for each combination of band index

pair (occ-nocc) and k point. One can reduce the number of terms by introducing a threshold value below which the respective weight is neglected. The corresponding keyword is `WGHTTHR` in the section `SUSCEP`. The default value is 10^{-10}. It is usually not necessary to change this value in practice.

5.2.7 GAUSS

(*) The Gaussian method is included mostly for testing. It does not only affect the polarization function but also all other k summations (such as for the Hartree-Fock exchange potential and the determination of the Fermi energy). Therefore, `GAUSS` is a global keyword. It effectively replaces each single-particle eigenvalue by a Gaussian function with a certain width so that the density of states becomes a smooth function. In the calculation of the polarization function, one additionally needs a finite width for the Fermi edge. The keyword `GAUSS` needs the two width parameters as arguments given as real positive values. The advantage of the Gaussian method is its relative mathematical simplicity compared to the tetrahedron method and, in particular, that the Gaussian method cannot give rise to symmetry breaking.

Table 16: Example

`GAUSS 0.001 0.01`	Use Gaussian k-integration method with widths 0.001 and 0.01 htr. (Mostly used for testing.)

5.2.8 DISORDER (SUSCEP)

(*) Mathematically, the parameter η in (5.5) is a positive infinitesimal ensuring the correct time order of the response quantity. Spex effectively treats it as an infinitesimally small positive parameter. Sometimes, however, it can be useful to use a finite value for η, e.g., to simulate disorder in a material. This is possible with the keyword `DISORDER` in the section `SUSCEP`. Note that this keyword has rarely been used so far.

Table 17: Example

`DISORDER 1000`	Use a finite value $\eta = 1/(2 \cdot 1000)$ htr.

5.3 Spectra

The dielectric function $\epsilon_{\mathbf{G}\mathbf{G}'}(\mathbf{k}, \omega)$ is related to several spectroscopic experimental techniques where the system is perturbed (excited) by some incoming beam of particles but does not lose or gain particles (in contrast to photoemission, for example, where electrons are lost from the sample).

In optical spectroscopy measurements, the absorption cross section is proportional to the imaginary part of the long-wavelength macroscopic dielectric function $\varepsilon_M(\omega)$, which can be obtained from the microscopic dielectric function by

$$\varepsilon_M(\omega) = \lim_{\mathbf{k}\to 0} \frac{1}{\varepsilon^{-1}_{\mathbf{00}}(\mathbf{k}, \omega)} .$$

Here, ε^{-1} refers to the matrix inverse.

Another example is electron-energy loss spectroscopy (EELS), where the scattering cross section can be shown to be proportional to the head element ($\mathbf{G} = \mathbf{G}' = 0$) of the inverse dielectric function

$$\varepsilon^{-1}_{\mathbf{00}}(\mathbf{k}, \omega) .$$

Spex can be used to calculate the dielectric function. The respective job is called `DIELEC`. Arguments are expected to define the k point index or label and a frequency range. For example, `JOB DIELEC X:{0:1,0.001}` would yield

the dielectric function evaluated at the X point between 0 and 1 htr in steps of 0.001 htr. Two files are written, "dielec" and "dielecR", containing the head elements of the (bare) dielectric function $\varepsilon_{\mathbf{00}}(\mathbf{k},\omega)$ and of the renormalized dielectric function $1/\varepsilon^{-1}_{\mathbf{00}}(\mathbf{k},\omega)$. The imaginary part of the latter can directly be identified with an optical absorption spectrum (excluding excitonic effects). For the EELS spectrum, one would have to plot the imaginary part of the respective reciprocal value (for example, in "gnuplot": `plot "dielecR" using 1:(imag(1/($2+{0,1}*$3)))`).

Alternatively, the frequency mesh can be provided through a file, e.g., `DIELEC X:"my_frequency_file"`. The file should contain a list of frequencies, one frequency per line. Complex frequencies can be defined by, e.g., `(1,0.2)` ($= 1 + 0.2i$). Comments (`# ...`) and empty lines are allowed.

In the same way, other quantities can be calculated: the polarization function with `SUSCEP` (written to the file "suscep"), the renormalized polarization function ($R = P + PvR$) with `SUSCEPR` (written to the file "suscepR"; "suscep" is also written), the screened interaction with `SCREEN` (written to the file "screen"). In the first case (`SUSCEP`), one can also specify spin indices. For example, `JOB SUSCEP X:ud{0:1,0.001}` restricts the polarization function to virtual up→down transitions. Other valid spin labels are `uu`, `dd`, `du`, and `+` (spin summed, default).

Since a dielectric matrix (or susceptibility, screened interaction etc.) is calculated for each frequency (e.g., thousand frequencies in the case of `{0:1,0.001}`), the memory demand can be very large for big systems. In such a case, it makes sense to divide the job into several smaller jobs. This can be done by hand (example: `JOB DIELEC X:{0:0.499,0.001} X:{0.5:1,0.001}`) but also by a special adjunct to the definition of the frequency range. For example, `JOB DIELEC X:{0:1,0.001}/2` would give the same subdivision as the previous example.

By default, only the head element is written to the output files. More elements can be written using, e.g., `JOB SUSCEP 1:{0.5eV:2eV,0.001eV},OUT=4`, which would yield 4×4 matrices instead of only the head element. One can also write the full matrix to a binary file with `...,OUT=BIN`.

The special + k-point label can be used as well (e.g., `JOB SUSCEP +:{0..50eV,0.01eV}`). This enables using the script "spex.band" to prepare a whole list of spectra for a series of Bloch vectors on a high-symmetry line. As explained in Section 5.1.16 for the file "spectral", the script appends a three-digit counting index to the name of the output file ("suscep" in the present example, which is thus renamed to "suscep001", "suscep002", ...). The data can then be compiled into a single file using the "spex.extr" utility. (See Section 5.1.16 for details).

5.4 Total xc energies

Spex can calculate total exchange and correlation energies of the many-electron system. The total exchange energy is defined by the Hartree-Fock expression

$$E^{\mathrm{x}} = -\frac{1}{2}\sum_{\sigma}\sum_{\mathbf{k}n}\sum_{\mathbf{k}'n'}\iint \frac{\phi^{\sigma *}_{\mathbf{k}n}(\mathbf{r})\phi^{\sigma}_{\mathbf{k}'n'}(\mathbf{r})\phi^{\sigma *}_{\mathbf{k}'n'}(\mathbf{r}')\phi^{\sigma}_{\mathbf{k}n}(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d^3r\, d^3r' \,. \tag{5.6}$$

The corresponding job is called `HFENE`. Internally, it works very much like a HF calculation. So, all parameters pertaining to `HF` are available to `HFENE`, as well. However, the final result is a single number, E^{x}.

The total correlation energy can be calculated using the adiabatic-connection fluctuation-dissipation theorem (ACFDT) with the random-phase approximation (RPA) for the (partially renormalized) response function χ_λ ($\lambda = [0,1]$). This response function is identical to the polarization function P (Section 5.2) for $\lambda = 0$ and to the "renormalized polarization function" R mentioned in Section 5.3 for $\lambda = 1$.

The ACFDT method gives the following form of the correlation energy functional

$$E_{\mathrm{c}}[n] = -\int_0^1 d\lambda \iint d^3r d^3r' \frac{1}{|\mathbf{r}-\mathbf{r}'|} \times \left[\int_0^\infty \frac{du}{2\pi}\chi_\lambda(\mathbf{r},\mathbf{r}',iu) - \chi_0(\mathbf{r},\mathbf{r}',iu)\right] \tag{5.7}$$

where χ_0 corresponds to the (bare) polarization function [Eq. (5.5)] and χ_λ is a partially renormalized response

function (i.e., with a scaled electron-electron interaction) fulfilling

$$\chi_\lambda(\mathbf{r},\mathbf{r}',\omega) = \chi_0(\mathbf{r},\mathbf{r}',\omega) + \iint d^3r'' d^3r''' \chi_0(\mathbf{r},\mathbf{r}'',\omega) \times \left[\frac{\lambda}{|\mathbf{r}''-\mathbf{r}'''|} + f_{\mathrm{xc},\lambda}(\mathbf{r}'',\mathbf{r}''',\omega)\right] \chi_\lambda(\mathbf{r}''',\mathbf{r}',\omega)\,,$$

here written in the general form including the (scaled) exchange-correlation kernel $f_{\mathrm{xc},\lambda}$. For RPA, $f_{\mathrm{xc},\lambda} = 0$.

The evaluation of Eq. (5.7) requires the polarization function to be calculated for all (irreducible) k points on an imaginary frequency mesh. The procedure is thus similar to a one-shot *GW* calculation and needs a similar computation time. Many of the parameters discussed in connection with the *GW* method (e.g., details related to the calculation of the polarization function, Section 5.2) can be used for ACFDT-RPA in the same way. In particular, the same rules using the `RESTART` option (Section 5.1.13) apply. It is, for example, possible to reuse the file "spex.cor" obtained from a previous *GW* calculation for an ACFDT-RPA calculation. A run with `RPAENE` comprises the calculation of the total exchange energy using Eq. (5.6).

5.5 Interaction parameters (Hubbard *U*)

In a many-electron system, the electrons are correlated with each other through the Coulomb interaction. The motion of one electron depends on the dynamics of all others. The Coulomb interaction is thus responsible for the fact that we have to deal with a highly complex many-body problem. DFT takes electronic correlation into account, but most of the available functionals are suitable only for weak to moderate correlation. One way to go beyond standard DFT is by many-body perturbation theory, for example, using the *GW* approximation. However, this approach has its limits for very strong electronic correlations as in Mott insulators, for example.

As an alternative, one can resort to methods with a special treatment of local correlations, for example, the so-called LDA+*U* scheme, in which the local-density approximation (LDA) of DFT is augmented by an on-site Coulomb repulsion term and an exchange term with the Hubbard *U* and Hund exchange *J* parameters, respectively.

A more elaborate computational scheme, which combines many-body model Hamiltonian methods with DFT, is the so-called LDA+DMFT method. In this scheme, the interacting many-body system is mapped onto the subspace of localized states, for example, formed by d orbitals. The subspace contains much less electrons and can thus be treated with high-level quantum many-body approaches such as quantum Monte Carlo, numerical renormalization group, diagrammatic techniques, exact diagonalization, et cetera.

The other electrons, not included in the subspace, cannot simply be ignored. For example, their screening processes lead to a renormalization of the effective electron-electron interaction that acts in the subspace, again giving rise to the parameters *U* and *J*. Thus, the Coulomb interaction parameters play a crucial role in the study of the correlation effects in solids.

There are two ways to obtain the Hubbard *U* parameter from first principles, constrained local-density approximation (cLDA) and constrained random-phase approximation (cRPA). The latter is implemented in the Spex code and relies on the fact that, since the electrons outside the localized subspace can be assumed delocalized, the random-phase approximation should be appropriate for describing their screening contribution. The cRPA offers an efficient way to calculate the effective Coulomb interaction parameters in solids. Moreover, cRPA allows individual Coulomb matrix elements to be determined, e.g., on-site, off-site, intra-orbital, inter-orbital, and exchange, as well as their frequency dependence.

This is a minimal example input file for a Hubbard *U* calculation (here, employing a $4 \times 4 \times 4$ k-point set):

```
BZ 4 4 4
JOB SCREENW {0}

SECTION WANNIER
```

(continues on next page)

(continued from previous page)

```
  ORBITALS 1 8 (d)
END

SECTION SUSCEP
  HUBBARD
END
```

The section WANNIER contains the parameters for the construction of the Wannier functions, see Section 6.3. (Note that the present values an example!) The Wannier set spans the correlated subspace. Without HUBBARD, Spex would calculate parameters for the fully screened interaction instead of the partially screened Hubbard U interaction. See below for further details.

5.5.1 JOB SCREENW ...

In general, the bare and (fully or partially) screened interaction is calculated in reciprocal space and represented in the mixed product basis (Section 6.2). The job definition SCREENW tells Spex to project this interaction potential onto the set of local Wannier functions defined in the section WANNIER (Section 6.3). The resulting interaction matrix elements

$$V_{n_1n_2,n_3n_4}(\omega) = \langle n_1n_2|V(\omega)|n_3n_4\rangle = \iint w^*_{n_1}(\mathbf{r})w_{n_3}(\mathbf{r})V(\mathbf{r},\mathbf{r}';\omega)w^*_{n_2}(\mathbf{r}')w_{n_4}(\mathbf{r}')d^3rd^3r' \tag{5.8}$$

are written to the file "spex.cou" for the bare interaction ($V = v$) and the screened interaction ($V = U$ if HUBBARD is specified, otherwise $V = W$, see below). In addition, if $\omega = 0$ is included in the frequency mesh, effective interaction parameters (Hubbard-Hund and/or Kanamori parameterization) are written to the output file for both the bare (v) and the screened interaction (U or W). (Note again that, without the keyword HUBBARD, the effective parameters are obtained from the fully screened interaction W and would thus be unsuitable for a Hubbard model Hamiltonian.) The Hubbard-Hund parameters are given for each complete Wannier l shell (e.g., specifying (d) in ORBITALS defines the complete d shell, see Section 6.3), here written for the partially screened interaction U,

$$U_l = \frac{1}{(2l+1)^2}\sum_{m=-l}^{l}\sum_{m'=-l}^{l}U_{mm',mm'}(0)$$

$$J_l = U_l - \frac{1}{2l(2l+1)}\sum_{m=-l}^{l}\sum_{m'=-l}^{l}[U_{mm',mm'}(0) - U_{mm',m'm}(0)]$$

The Kanamori parameters are given for groups of t_{2g} ($N = 3$) or e_g orbitals ($N = 2$) as

$$U_l = \frac{1}{N}\sum_{m=-l}^{l}U_{mm,mm}(0)$$

$$U'_l = \frac{1}{N(N-1)}\sum_{m=-l}^{l}\sum_{m'=-l,m'\neq m}^{l}U_{mm',mm'}(0)$$

$$J_l = \frac{1}{N(N-1)}\sum_{m=-l}^{l}\sum_{m'=-l,m'\neq m}^{l}U_{mm',m'm}(0)$$

In the case of a spin-polarized system, the spin-up and spin-down Wannier functions differ slightly, and the respective spin-indices have to be specified in the job definition using one of the labels uu, dd, du, or + (spin summed). There is no default.

Table 18: Examples

`JOB SCREENW {0:40eV,0.1eV}`	Calculate $W(\mathbf{r},\mathbf{r}';\omega)$ [$U(\mathbf{r},\mathbf{r}';\omega)$ if `HUBBARD` is specified] and project it onto the Wannier functions; {...} defines the frequency mesh.
`JOB SCREENW {0}`	The same only for the static case $\omega = 0$. (Here, `{0}` is short for `{0:0,1}`.)
`JOB SCREENW uu{0}`	The same for the spin-polarized case: the up-up combination of Wannier functions is selected.

5.5.2 HUBBARD (SUSCEP)

With the keyword `HUBBARD` (section `SUSCEP`), the screening that takes place in the correlated subspace is eliminated from the screened interaction, thus yielding the partially screened interaction and, with `JOB SCREENW ...`, the effective interaction parameters for a Hubbard model Hamiltonian. The correlated subspace is spanned by the Wannier functions. So, the Wannier definition affects the calculation in two ways: (1) It defines the local Wannier basis on which the (bare and) partially screened interaction is projected, and (2) it spans the subspace whose screening is eliminated. Sometimes it makes sense to separate these two effects, for example, if the subspace is spanned by Wannier functions located at several (possibly symmetry-equivalent) atoms, but the effective parameters are only needed for one of the atoms (on-site interactions). Such a calculation is performed in two steps with a file name as argument (example: `HUBBARD hub`). In the first step (i.e, the file does not exist yet), one specifies the full set of Wannier functions. When Spex is run, it writes information about the screening contribution that takes place in the correlated subspace (and that is to be eliminated) to the specified file ("hub") and stops. Before the second run, one can modify the Wannier definition (e.g., with the keyword `SUBSET`). In the present example, one would reduce the Wannier set (to the orbitals located at one and the same atom). The second Spex run then calculates the partially screened interaction and projects it onto the reduced set of Wannier functions.

Table 19: Examples

`HUBBARD`	Calculate Hubbard *U* interaction matrix including effective parameters.
`HUBBARD hub.dat`	Write information about subspace screening to "hub.dat", if the file does not exist, and calculate the Hubbard *U* interaction matrix using data from "hub.dat" otherwise.

5.5.3 RESTART

Since `SCREENW` calculations can be quite costly, the keyword `RESTART` enables a restart of a calculation that has stopped before finishing. The usage is similar to *GW* calculations (Section 5.1.13). The file "spex.cor" contains the (fully or partially) screened interaction for each (completed) k point, and the file "spex.wcou" holds the last update of the projected interaction matrix. The logical table is similar to Section 5.1.13. (See Section 4.1.7 for more details.)

	`spex.cor`	`spex.wcou`
–	–	write
`RESTART`	read-write	write
`RESTART 2`	read-write	read-write

5.5.4 DISENTGL (WANNIER)

(*) There are different ways to eliminate the screening channels in the case of an entangled band structure, i.e., if the correlated subspace is not made up of isolated bands. In this case, it is not possible to unambiguously define the screening that takes place only in the subspace. The default approach of Spex, described in [Phys. Rev. B 83, 121101 (2011)], scales each virtual transition $\phi \to \phi'$ by the probability that the transition takes place in the subspace. (To

be more precise, it is the probability that the electron-hole pair occupying the states ϕ and ϕ' happens to be in the subspace.) There is another method, called *disentanglement* method [Phys. Rev. 80, 155134 (2009)], in which the reference mean-field system is modified in such a way that each state ϕ is either fully contained in the subspace or completely outside (corresponding to probabilities 1 or 0). This method can be used by specifying `DISENTGL` in the section `WANNIER`.

Warning: Using this keyword actually modifies the mean-field system! (The keyword has not been tested thoroughly yet.)

5.5.5 RSITE (WANNIER)

(*) The interaction parameters are calculated between Wannier orbitals located at the atoms of the first unit cell. Let us denote two site position vectors as $\mathbf{r}_1$ and $\mathbf{r}_2$. We have on-site parameters for $\mathbf{r}_1 = \mathbf{r}_2$ and off-site parameters for $\mathbf{r}_1 \neq \mathbf{r}_2$ (which is, of course, possible only for a multi-atom basis). However, one of the atoms, say the first, can also be located in a different unit cell shifted by a lattice vector $\mathbf{R}$, i.e., at the position $\mathbf{r}_1 + \mathbf{R}$, which gives rise to another class of off-site interaction parameters (which exists also for a single-atom basis). The keyword `RSITE` can be used to define the lattice vectors $\mathbf{R}$. The following examples show the syntax.

Table 20: Examples

`RSITE (0,0,1) (0,1,1) (1,1,1)`	Defines three lattice vectors. If (0,0,0) is not included explicitly, as in the present example, it is prepended to the list automatically.
`RSITE (0,0,0)-(0,5,10)`	Defines the vectors (0,0,0), (0,1,2), (0,2,4), ..., (0,5,10).
`RSITE (0,0:2,0:2)`	Defines the vectors (0,0,0), (0,0,1), (0,0,2), (0,1,0), ..., (0,2,2).

5.5.6 WBLOCH (WANNIER)

(*) The evaluation of the interaction parameters involves projections of the form $\langle M_{\mathbf{k}\mu} w_{\mathbf{q}n} | w_{\mathbf{k}+\mathbf{q}n'} \rangle$, where $M_{\mathbf{k}\mu}(\mathbf{r})$ are the functions of the mixed product basis (see Section 6.2), in which the screened (bare) interaction potential is expanded, and $w_{\mathbf{k}n}(\mathbf{r})$ are the Wannier Bloch functions (corresponding to Eq. (6.1) of Section 6 without the $\mathbf{k}$ summation). Until version 05.03., the coefficients for the latter (with respect to the LAPW basis) were constructed explicitly. Since version 05.04., these coefficients are not used anymore, and we instead compute $\langle M_{\mathbf{k}\mu} \phi_{\mathbf{q}m} | \phi_{\mathbf{k}+\mathbf{q}m'} \rangle$ with the single-particle eigenfunctions $\phi_{\mathbf{k}m}(\mathbf{r})$, allowing the symmetry properties of the eigenfunctions to be exploited to speed up the calculation. To this end, irreducible representations are employed. The Wannier expansion coefficients $U_{\mathbf{k}m,n}$ are multiplied later. The Wannier set, on the other hand, is not guaranteed to reflect the full symmetry of the system. The new algorithm is substantially faster than the old one, in particular, for large k-point sets, if the system is highly symmetric. Furthermore, the new algorithm has less restrictions than the old one. For example, it can be used in conjunction with `MAXIMIZE` and `SUBSET` (and without `STOREBZ`) in contrast to the old algorithm. However, if, on the other hand, the system has only low symmetry and/or the number of Wannier functions (index n) is much smaller than the number of eigenstates (index m) from which the Wannier set is constructed, then the old algorithm might be more efficient. In addition, slight distortions of the system can make the usage of irreducible representations critical, in which case the old algorithm should be used. (Also see keyword `NOSYM`, Section 4.2.10.) The old algorithm can still be used with the keyword `WBLOCH`.

Note: Paragraphs discussing advanced options are preceded with (*), and the ones about obsolete, unmaintained, or experimental options are marked with (**). You can safely skip the paragraphs marked with (*) and (**) at first reading.

CHAPTER 6

Basis sets

6.1 LAPW basis

The reader is supposed to be familiar with the LAPW basis set. This section is intended as a short recapitulation and to define the notation used in the manual.

The LAPW method relies on a partitioning of space into non-overlapping MT spheres centered at the atomic nuclei and the interstitial region. The basis functions are defined differently in the two regions, plane waves in the interstitial and numerical functions in the spheres, consisting of a radial part $u^a_{lp}(r)$ (a is the atomic index) and the spherical harmonics $Y_{lm}(\hat{\mathbf{r}})$. p is an index to distinguish different types of radial functions: $u^a_{l0} = u^a_l$, $u^a_{l1} = \dot{u}^a_l = \partial u^a_l / \partial \epsilon$, $u^a_{lp} = u^{a,\mathrm{LO}}_l$, $p \geq 2$, where we have used the usual notation of LAPW. Augmented plane waves are formed by matching linear combinations of the u and $\dot{u}$ to the interstitial plane waves, forming the standard LAPW basis set. Local orbitals u^{LO} can be used to extend the basis set, to enable the description of semicore and high-lying conduction states. The plane waves and the angular part of the MT functions can be converged straightforwardly with the reciprocal cutoff radius $g_{\max}$ and the maximal l quantum number $l_{\max}$, respectively, whereas the radial part of the MT functions is more difficult to improve systematically, but it is possible, see [Comput. Phys. Commun. 184, 2670 (2013)].

Running the one-shot mean-field calculation with Spex (using `ITERATE`, see Section 4.2.9) offers the possibility of modifying the LAPW basis in "spex.inp". This can be useful for *GW* calculations, which require an accurate basis for a much larger energy range (including many unoccupied states) than in DFT. The respective keywords are defined in the section `LAPW` and are described in the following.

6.1.1 GCUT (LAPW)

The set of $\mathbf{G}$ vectors depends on the k point and is defined with the cutoff condition $|\mathbf{k} + \mathbf{G}| < G_{\mathrm{cut}}$. The parameter G_{cut} can be defined with the keyword `GCUT`, e.g., `GCUT 4.0` (in Bohr^{-1}).

6.1.2 LCUT (LAPW)

In the muffin-tin spheres, the basis functions are expanded in radial functions $u_{lp}(r)$ and spherical harmonics $Y_{lm}(\hat{\mathbf{r}})$. The keyword `LCUT` lets you define the cutoff for the *l* quantum number separately for each atom type.

Table 1: Example

`LCUT 10 8`	An *l* cutoff of 10 and 8 is used for the first and second atom type, respectively.

6.1.3 EPAR (LAPW)

The radial functions $u_{lp}(r)$ are determined from scalar-relativistic Dirac equations inside the MT spheres with energy parameters E_{lp}. (As one cannot impose a boundary condition at the MT boundary, the energy does not result from the solution as an eigenvalue but must be provided as a parameter.) The energy parameters for the functions $u_{l0}(r)$ [usually denoted by $u_l(r)$] and $u_{l1}(r)$ [$\dot{u}_l(r)$] (having the same parameter in a given *l* channel) are defined with the keyword `EPAR`. There can be at most as many arguments as atom types. (For atom types without an explicit definition, the energy parameters remain unchanged.) Each argument is bounded by parentheses `(...)`. An empty definition `()` is allowed and means that the energy parameters of this atom type remain unchanged. Otherwise, one writes a comma-separated list of parameters starting at $l = 0$ (then, $l = 1$, etc.). Parameters not given will just be set identical to the last one given. A wildcard ($\star$) selects the energy parameter from the input data. Energy parameters can be given explicitly as real numbers (in Hartree or eV with the unit `eV`) or as main quantum numbers of the atomic shell. The usage is shown in the examples.

Table 2: Examples

`EPAR (-0.01,0.04,0.06)`	The energy parameters for *s*, *p*, *d*, and *f* (etc.) functions are defined to be -0.01, 0.04, 0.06, 0.06 (etc.). (All values in Hartree.)
`EPAR () (-0.01/-0.005,*,3,*)`	The energy parameters for the first atom type are unchanged. The energy parameters for the *s* functions of the second atom type are -0.01 htr for spin up and -0.005 htr for spin down. The parameters for the *p* functions are unchanged from the input. The parameters for the *d* functions are taken from an atomic solver (3*d* orbital) for both spins. The parameters for higher *l* are unchanged.
`EPAR () (-0.01/-0.005,*,3d,*)`	Same as before. The possibility of defining `3d` explicitly is included for better readability of the input file.

6.1.4 LO (LAPW)

Local orbitals (LOs) are used to improve the basis set in the atomic MT spheres. There are two use cases: First, to describe low-lying semicore states (which are too close to the valence bands to be described as core states confined in the MT spheres). Second, to improve the description of high-lying unoccupied states, which are relevant for *GW* calculations and, generally, for response quantities. LOs are defined by a similar syntax as `EPAR`. However, there is no fixed order in the *l* quantum number. Therefore, one always has to define the *l* quantum number together with the energy parameter or the main quantum number. Since there is no fixed order, the wildcard ($\star$) used in `EPAR` to "skip" *l* quantum numbers is unnecessary and not available. However, there is another use of the wildcard ($\star$). If given as the first argument (without parentheses), the LOs from the input data are kept and the new LOs are added to this set. Without $\star$, only the LOs defined after `LO` will be used (and the ones from the input data are removed). There is a special LO definition that lets Spex automatically add local orbitals to improve the MT basis representation for unoccupied states. In particular, Spex adds local orbitals in such a way that the MT basis is able to accurately represent states over the energy range $v_0 < \epsilon < v_0 + G_{\mathrm{cut}}^2/2$ with the average effective potential v_0 and the momentum cutoff G_{cut}. [In this energy range, the wavefunctions are well described in the interstitial region (by the plane-wave basis defined with G_{cut}.)] Here, "accurate" means that states in this energy range can be represented in the MT spheres within an accuracy of at least 99% (also see keyword `MTACCUR`, Section 4.2.12). An example for this special LO definition is `spd:+`, in which case LOs would be automatically added in the *s*, *p*, and *d* channel. Note that adding many local orbitals can make the basis set so flexible that core states might appear as eigenstates (perhaps badly

described as high-lying "ghost" bands). In this case (if the spurious band is below and well separated from the valence region), `ITERATE` allows the specification of an eigenenergy cutoff below which all eigenstates are cut away. The usage of the keyword `LO` is shown in the examples.

Table 3: Example

`LO (s:-30.4eV,p:-29.2eV)`	Define *s* and *p* local orbitals with explicit energy parameters -30.4 and -29.2 eV. (The default unit, i.e., without `eV`, is Hartree units.)
`LO (l=0:-30.4eV,l=1:-29.2eV)`	Same as before defined with an alternative syntax.
`LO (l=0:-30.4eV/-29.5eV,l=1:-29.2eV/-28.3eV)`	Same as before for a spin-polarized system.
`LO () (3s,3p)`	Define *s* and *p* local orbitals for the second atom type with parameters obtained from an atomic solver (for the description of semicore states).
`LO (3d,spd:+)`	Define 3*d* local orbital and let Spex automatically add local orbitals to improve the description of the unoccupied spectrum in the *s*, *p*, and *d* channel.
`LO * (spd:+)`	Add higher-energy local orbitals as before but also keep local orbitals from the input file (which might include semicore LOs.)

6.1.5 MTACCUR (ANALYZE)

(*) The accuracy of the radial MT basis can be analyzed with the keyword `MTACCUR` followed by two numbers e_1 and e_2. Spex then calculates the MT representation error [Phys. Rev. B 83, 081101] in the energy range between e_1 and e_2. (If unspecified, upper and lower bounds are chosen automatically.) The results are written to the output files "spex.mt.*t*" where *t* is the atom type index, or "spex.mt.*s*.*t*" with the spin index *s*(=1 or 2) for spin-polarized calculations. The files contain sets of data for all l quantum numbers, which can be plotted separately with "gnuplot" (e.g., `plot "spex.mt.1" i 3` for $l = 3$).

Table 4: Examples

`MTACCUR -1 2`	Calculate MT representation error between -1 and 2 Hartree.
`MTACCUR`	Automatic energy range.

6.2 Mixed product basis

The methods implemented in Spex distinguish themselves from conventional DFT with local or semilocal functionals by the fact that individual two-particle scattering processes are described explicitly. In such processes, the state of an incoming particle and the state that it scatters into form products of wavefunctions. Many of the physical quantities, such as the Coulomb interaction, the polarization function, and the dielectric function, become matrices when represented in a suitable product basis. In the context of the FLAPW method, the mixed product basis is an optimal choice. It consists of two separate sets of functions that are defined only in one of the spatial regions, while being zero in the other, the first is defined by products of the LAPW MT functions and the second by interstitial plane waves. (The products of plane waves are plane waves again.) The corresponding parameters are defined in the section `MBASIS` of the input file.

6.2.1 GCUT (MBASIS)

If N is the number of LAPW basis functions, one would naively expect the number of product functions to be roughly N^2. In the case of the interstitial plane waves, this is not so, since, with a cutoff $g_{\max}$, the maximum momentum of the product would be $2g_{\max}$, leading to $8N$ as the number of product plane waves. Fortunately, it turns out that the basis size can be much smaller in practice. Therefore, we introduce a reciprocal cutoff radius $G_{\max}$ for the interstitial plane waves and find that, instead of $G_{\max} = 2g_{\max}$, good convergence is achieved already with $G_{\max} = 0.75g_{\max}$, the default value. The parameter $G_{\max}$ can be set to a different value with the keyword `GCUT` in the section `MBASIS` of the input file.

Table 5: Example

`GCUT 2.9`	Use a reciprocal cutoff radius of 2.9 Bohr^{-1} for the product plane waves.

6.2.2 LCUT (MBASIS)

The MT functions are constructed from the products $u^a_{lp}(r)Y_{lm}(\hat{\mathbf{r}})u^a_{l'p'}(r)Y_{l'm'}(\hat{\mathbf{r}})$. (Obviously, the atom index a must be identical.) The product of spherical harmonics expands into a linear combination of $Y_{LM}(\hat{\mathbf{r}})$ with $L = |l - l'|, ..., l + l'$ and $|M| \leq L$. In principle, this leads to a cutoff of $2l_{\max}$ for the products, but, as with the plane waves, one can afford to use a much smaller cutoff parameter in practice. We use $L_{\max} = l_{\max}/2$ as the default. The corresponding keyword in the input file is called `LCUT`.

Table 6: Example

`LCUT 5 4`	Use l cutoffs of 5 and 4 for two atom types in the system.

6.2.3 SELECT (MBASIS)

In the LAPW basis, the matching conditions at the MT boundaries require large l cutoff values, typically around $l = 10$, giving a correspondingly large number of radial functions. Not all of these functions must be considered in the product basis set. The keyword `SELECT` enables a specific choice of the radial functions. When specified, an entry for each atom type is required. The best way to describe the syntax of `SELECT` is with examples:

Table 7: Examples

`SELECT 2;3`	Use products of $u_{lp}u_{l'p'}$ with $l \leq 2$ and $l' \leq 3$ for $p = 0$ (so-called u) and no function of $p = 1$ (so-called $\dot{u}$).
`SELECT 2,1;3,2`	Same as before but also include the $\dot{u}$ functions with $l \leq 1$ and $l' \leq 2$.
`SELECT 2,,1100;3,,1111`	Same as first line but also include the local orbitals $p \geq 2$, which are selected (deselected) by "1" ("0"): here, the first two and all four LOs, respectively. The default behavior is to include semicore LOs but to exclude the ones at higher energies.
`SELECT 2,1,1100;3,2,1111`	Same as second line with the LOs.

The LOs are selected by series of 1s and 0s. If there are many LOs, a definition like `0000111` can be abbreviated by `4031` (four times `0`, three times `1`).

One of the most important quantities to be calculated in *GW* calculations (and related methods) are the "projections" $\langle M_{\mathbf{k}\mu}\phi_{\mathbf{q}n}|\phi_{\mathbf{k}+\mathbf{q}n'}\rangle$, which describe the coupling of a propagating particle to an interaction line. Often, the two states with n and n' are occupied and unoccupied, respectively. In this sense, the first set of `SELECT` parameters before the semicolon refers to the occupied states, the set after the semicolon refers to the unoccupied states. For $l = 2$ with $2l$ or $2l + 1$ being the l cutoff specified by `LCUT`, the default would be `2;3`.

6.2.4 TOL (MBASIS)

(*) The set of MT products selected by `SELECT` can still be highly linearly dependent. Therefore, in a subsequent optimization step one diagonalizes the MT overlap matrix and retains only those eigenfunctions whose eigenvalues exceed a predefined tolerance value. This tolerance is 0.0001 by default and can be changed with the keyword `TOL` in the input file.

Table 8: Example

`TOL 0.00001`	Remove linear dependencies that fall below a tolerance of 0.00001 (see text).

6.2.5 OPTIMIZE (MBASIS)

Despite the favorable convergence behavior of the mixed product basis, it can still be quite large. In the calculation of the screened interaction, each matrix element, when represented in the basis of Coulomb eigenfunctions, is multiplied by $\sqrt{v_\mu v_\nu}$ with the Coulomb eigenvalues $\{v_\mu\}$. This gives an opportunity for reducing the basis-set size further by introducing a Coulomb eigenvalue threshold $v_{\min}$. The reduced basis set is then used for the polarization function, the dielectric function, and the screened interaction. The parameter $v_{\min}$ can be specified after the keyword `OPTIMIZE MB` in three ways: first, as a "pseudo" reciprocal cutoff radius $\sqrt{4\pi/v_{\min}}$ (which derives from the plane-wave Coulomb eigenvalues $v_{\mathbf{G}} = 4\pi/G^2$), second, directly as the parameter $v_{\min}$ by using a negative real number, and, finally, as the number of basis functions that should be retained when given as an integer. The so-defined basis functions are mathematically close to plane waves. For testing purposes, one can also enforce the usage of plane waves (or rather projections onto plane waves) with the keyword `OPTIMIZE PW`, in which case the Coulomb matrix is known analytically. No optimization of the basis is applied, if `OPTIMIZE` is omitted.

Table 9: Examples

`OPTIMIZE MB 4.0`	Optimize the mixed product basis by removing eigenfunctions with eigenvalues below $4\pi/4.0^2$.
`OPTIMIZE MB -0.05`	Optimize the mixed product basis by removing eigenfunctions with eigenvalues below 0.05.
`OPTIMIZE MB 80`	Retain only the eigenfunctions with the 80 largest eigenvalues.
`OPTIMIZE PW 4.5`	Use projected plane waves with the cutoff 4.5 Bohr^{-1} (for testing only, can be quite slow).

In summary, there are a number of parameters that influence the accuracy of the basis set. Whenever a new physical system is investigated, it is recommendable to converge the basis set for that system. The parameters to consider in this respect are `GCUT`, `LCUT`, `SELECT`, and `OPTIMIZE`.

Finally, we show a section `MBASIS` for a system with three atom types as an example

```
SECTION MBASIS
  GCUT 3.0
  LCUT 5 4 4
  SELECT 3,2;4;2 2;3 2;3
  TOL 0.0001
  OPTIMIZE MB 4.5
END
```

6.2.6 NOAPW (MBASIS)

(*) By default, Spex constructs augmented plane waves from the mixed product basis in a similar way as in the LAPW approach. In this way, the unphysical "step" at the MT sphere boundaries is avoided. It also leads to a further reduction of the basis set without compromising accuracy. This APW construction can be avoided with the keyword `NOAPW`.

6.2.7 ADDBAS (MBASIS)

(*) This keyword augments the mixed product basis by the radial u_{lp} functions, the l and p parameters are according to the first part of the SELECT definition. This improves the basis convergence for COHSEX calculations.

6.2.8 CHKPROD (MBASIS)

(*) Checks the accuracy of the MT product basis.

6.2.9 WFADJUST (MBASIS)

(**) Removes wavefunction coefficients belonging to radial functions that are not used to construct the MT product basis. (Currently the coefficients are actually not removed from memory.)

6.2.10 FFT (WFPROD)

(*) When the interaction potential is represented in the mixed product basis, the coupling to the single-particle states involve projections of the form $\langle M_{\mathbf{k}\mu}\phi_{\mathbf{q}n}|\phi_{\mathbf{k}+\mathbf{q}n'}\rangle$. The calculation of these projections can be quite expensive. Therefore, there are a number of keywords that can be used for acceleration. Most of them are, by now, somewhat obsolete. An important keyword, though, is FFT in the section WFPROD of the input file. When used (see examples), the interstitial terms are evaluated using Fast Fourier Transformations (FFTs), i.e., by transforming into real space (where the convolutions turn into products), instead of by explicit convolutions in reciprocal space. For small systems the latter is faster, but for large systems the FFTs give a better scaling with system size (see note). The optional argument gives the reciprocal cutoff radius used for the FFT. A run with FFTs can be made to yield results identical to the explicit summation. This requires an FFT reciprocal cutoff radius of at least $2g_{\max} + G_{\max}$, which can be achieved by setting FFT EXACT, but such a calculation is quite costly. It is, therefore, advisable to use smaller cutoff radii, thereby sacrificing a bit of accuracy but speeding up the computations a lot. If given without an argument, Spex will use 2/3 of the above *exact* cutoff, which is a compromise between precision and computational cost. One can also specify an explicit cutoff by a real-valued argument explicitly. Typical values are between 6 and 8 Bohr^{-1}.

Note: (Since version 05.06:) FFTs are included by default for large systems (unit-cell volume larger than 1200 Bohr^3) with the reciprocal cutoff radius $\frac{2}{3}(2g_{\max} + G_{\max})$. Otherwise, explicit convolutions are used.

Table 10: Examples

FFT 6	Use FFTs with the cutoff 6 Bohr^{-1}.
FFT	Use the default cutoff $\frac{2}{3}(2g_{\max} + G_{\max})$. This is the cutoff automatically used for large systems.
FFT EXACT	Use the exact cutoff $2g_{\max} + G_{\max}$.
FFT OFF	Do not use FFTs.

6.2.11 APPROXPW (WFPROD)

(**) Since version 02.03 the interstitial part of the projections is calculated exactly, leading to favorable convergence with respect to GCUT. However, the exact evaluation takes considerably more time. If APPROXPW is set, the old and approximate way of calculating the products is used instead. (Only included for testing.)

6.2.12 LCUT (WFPROD)

(**) The MT part of the projections can be accelerated as well using the the keyword `LCUT` in the section `WFPROD`. This keyword should not be confused with the keyword of the same name in the section `MBASIS`, Section 6.2.2. It can be used to restrict the l cutoff of the wavefunctions (only for the projections), one l cutoff for each atom type.

6.2.13 MINCPW (WFPROD)

(**) The keyword `MINCPW` modifies the plane-wave wave-function coefficients such that their absolute values become as small as possible. This reduces the error in the evaluation of the wave-function products arising from the finite G cutoff and leads to better convergence with respect to `GCUT` and smoother band structures, when `APPROXPW` is used. The coefficients can be modified, because the set of IPWs is overcomplete. If set, the eigenvectors of the overlap matrix with the `n` smallest eigenvalues are used for this modification. Note that it only makes sense to use this together with `APPROXPW`.

6.3 Wannier orbitals

For many features of the Spex code, Wannier functions are required: Wannier interpolation, spin-wave calculations, *GT* self-energy, Hubbard *U* parameter. Wannier functions are constructed by a linear combination of single-particle states. They can be viewed as Fourier transforms of the Bloch states. While Bloch states have a definite k vector and are delocalized over real space, the Wannier functions are localized at specific atomic sites and delocalized in k space. The mathematical definition is

$$w_{\mathbf{R}n}(\mathbf{r}) = \frac{1}{N}\sum_{\mathbf{k}} e^{-i\mathbf{k}\mathbf{R}} \sum_{m} U_{\mathbf{k}m,n}\phi_{\mathbf{k}m}(\mathbf{r}) \tag{6.1}$$

with the lattice vector $\mathbf{R}$, the orbital index n, the number of k vectors N, and the transformation matrix U. The Wannier orbital is localized in the unit cell at $\mathbf{R}$ and, in most cases, at a specific atom in that unit cell. The Wannier transformation matrix U is unitary if there are as many Wannier orbitals as there are Bloch bands included in the construction. There can be more Bloch bands but not less. There are several ways to determine U. The simplest is a projection method, in which the Bloch eigenstates are projected onto a localized atomic (muffin-tin) function $u_n(\mathbf{r})$, which can be of any orbital character (including hybrid orbitals). This gives an auxiliary matrix

$$U'_{\mathbf{k}m,n} = \langle u_n | \phi_{\mathbf{k}m} \rangle \,, \tag{6.2}$$

which is then (Loewdin) orthonormalized to yield the transformation matrix U. We call the Wannier functions calculated with this U "first-guess" Wannier functions, as they can be used as the input (first-guess) functions for a disentanglement and/or maximal localization procedure with the Wannier90 library, which then yields a new, refined U.

Wannier functions are defined in the section `WANNIER`:

```
SECTION WANNIER
  ORBITALS 1 4 [sp3]
  MAXIMIZE
END
```

The example shows the definition of four (bonding) sp^3 hybrid orbitals for silicon.

6.3.1 ORBITALS (WANNIER)

This line is the main definition of Wannier orbitals giving the orbital characters for the *first-guess* orbitals and the energy window. The energy window is defined by the first two arguments, the indices of the lower and upper band

[summation index m in Eq. (6.1)]. Note that if, in the above example, the fourth state is degenerate with the fifth at some k point, the fifth state will automatically be included as well. (This is the default behavior. It can be changed to excluding the fourth band or to ignoring degeneracies altogether by setting preprocessor macros, see the source file "wannier.f".) A wildcard (`*`) can be set in place of the upper band index, which lets Spex determine a (minimal) band index automatically. (Note that this will not necessarily yield an optimal upper index.) After the energy window follows the definition of the orbital characters of the projection functions used to calculate the *first-guess* Wannier functions. Although strictly not guaranteed, the orbital character is usually preserved not only after projection in the first-guess functions but even after the maximal localization procedure (see Section 6.3.2). The orbital character is defined for each atom type (round brackets) or each atom (square brackets). Empty brackets, such as `()` or `[]`, are allowed. For example, `(p) () (d)` gives three p orbitals in the first atom, five d orbitals in the third, and none in the second. The number of bands in the energy window must not be smaller than the number of Wannier orbitals (but can be larger).

Table 11: Examples

`ORBITALS 1 18 [s,p,d]`	Define s, p, and d Wannier orbitals in the first (and perhaps only) atom.
`ORBITALS 6 11 (t2g)`	Define t_{2g} Wannier orbitals in the first atom type (here two atoms) from the sixth through the 11th band.
`ORBITALS 6 * (t2g)`	Same as before, but the upper band index will be determined automatically.
`ORBITALS 1 9 [] [d]`	Define d Wannier orbitals in the second atom.

The above example for the `WANNIER` section is for bulk silicon, which has two atoms in the unit cell. Four sp^3 hybrid orbitals are defined for the first and none for the second. (Trailing empty brackets as in `(sp3) ()` can be omitted.) In this case, the four bands of the energy window (`1 4`) are all formed by bonding sp^3 orbitals. Therefore, by projection, the resulting Wannier orbitals are evenly distributed over the two atoms. This is an effect of the system's symmetry and not explicitly caused by the particular Wannier definition. If one wants to describe the antibonding orbitals as well, one would have to define, for example, `(sp3) (sp3)` or just `[sp3]`, and increase the energy window. Since the empty states of silicon are entangled, simply doubling the energy window (`1 8`) does not suffice. We will discuss the case of entangled bands below.

The orbital characters are defined in the same way as in Wannier90. The following orbital labels are available: `s`, `p`, `d`, `f`, `g`, `px`, `py`, `pz`, `dxy`, `dyz`, `dz2`, `dxz`, `dx2y2`, `t2g`, `eg`, `fz3`, `fxz2`, `fyz2`, `fzxy`, `fxyz`, `fxxy`, `fyxy`, `sp`, `sp2`, `sp3`, `sp3d`, and `sp3d2`. Consult the Wannier90 manual for details. In addition, there are capitalized labels (`S`, `P`, `D`, `F`, `G`), in which case complex spherical harmonics are used instead of real spherical harmonics. (Obviously, `S` is equivalent to `s` and included only for completeness. In practice, real and complex spherical harmonics behave identically in the present context.) Wannier functions (except for s orbitals) have an angular dependence. Sometimes it is necessary to specify a particular spatial orientation of Wannier orbitals. For this purpose, one can give Euler angles (in degrees), for example, `(s,px/30/60/-90,d)`, through which the Wannier orbital (of p_x symmetry in this case) is rotated; 1st angle: rotation about z axis, 2nd angle: rotation about new (rotated) x axis, 3rd angle: rotation about new (rotated) z axis.

The so-generated Wannier function are *first-guess* Wannier functions. As explained above, the expansion coefficients $U_{\mathbf{k}n,m}$ are obtained from projecting the Bloch functions $\phi_{\mathbf{k}n}$ onto localized functions with the chosen orbital character and a subsequent orthonormalization procedure to make the Wannier set orthonormal. For many purposes, the resulting set of Wannier functions is already usable. However, it is not yet maximally localized. The following keyword enforces maximal localization with Wannier90 (using the Wannier90 library), yielding the set of maximally localized Wannier functions (MLWFs).

6.3.2 MAXIMIZE (WANNIER)

The maximal localization procedure of Wannier90 is applied starting from the set of first-guess Wannier functions. The input file for Wannier90 ("wannier.win") is generated automatically by Spex if it is not present yet. The user can still provide his own "wannier.win" and, in this way, fine-tune the localization procedure. However, some of the parameters (`num_wan` and `num_bands`) might still be changed by Spex if they are inconsistent with the Spex

input (in ORBITALS). The output of Wannier90 is written to the file "wannier.wout". This file should be checked for convergence of the maximal localization. For further details, consult the Wannier90 manual.

Note: MAXIMIZE requires Spex to be linked to the Wannier90 library. To this end, the Spex configure script for compilation has to be run with the option `--with-wan`. See Section 1.

6.3.3 FROZEN (WANNIER)

In the case of entangled bands, it is often necessary to add an extra step in the construction of MLWFs, the disentanglement, which is also an iterative optimization procedure. To this end, one defines a *frozen energy window*, which lies inside the *outer* energy window defined by ORBITALS. The frozen energy window is defined by two energies, the lower and the upper bound. By default, the lower bound is assumed to coincide with the minimum energy of the lowest band of the outer window, i.e., the first argument of ORBITALS. If FROZEN is given without an argument, the upper bound is *guessed* by Spex. This guess can be overridden by giving the upper energy as an argument to FROZEN. Alternatively, one can specify the Wannier90 parameters dis_froz_min and dis_froz_max in the file "wannier.win". For illustration, the lower and upper bounds of the frozen energy window are shown in the figure below for the example of iron (spin-up). Giving FROZEN without MAXIMIZE is possible. In this case, the disentanglement procedure is performed but not the maximal localization procedure. You should check the convergence of the disentanglement procedure in the file "wannier.wout". For further details, consult the Wannier90 manual.

Table 12: Examples

FROZEN	Run disentanglement procedure. Frozen window bounds are guessed by Spex.
FROZEN 5eV	Run disentanglement procedure and use 5 eV as upper bound.
FROZEN 8. 5eV 10eV	Run disentanglement procedure with different upper bounds for spin-up and spin-down.

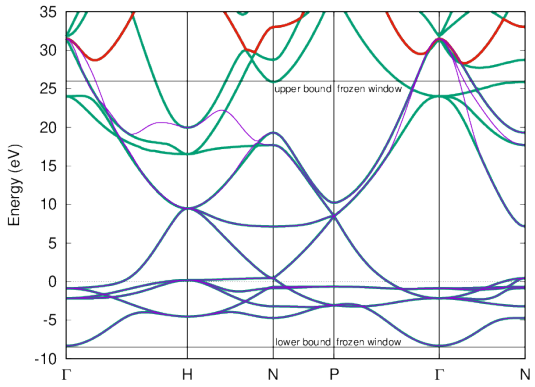

Fig. 1: Comparison of the explicit (green lines) and Wannier-interpolated (magenta lines) DFT band structure for iron (spin-up). The Wannier functions are constructed with ORBITALS 1 * (s,p,d), MAXIMIZE, and FROZEN. The lower bound of the outer energy window is set at the lowest valence band (band index 1), and the upper bound is set automatically by Spex (wildcard "*"). In this case, it is set to the 12th band, here marked in red. The lower and upper bounds of the inner frozen energy window for disentanglement are guessed by Spex and shown in the figure as well.

Note: FROZEN requires Spex to be linked to the Wannier90 library. To this end, the Spex configure script for compilation has to be run with the option `--with-wan`. See Section 1.

6.3.4 SUBSET (WANNIER)

With this keyword, a subset of Wannier functions can be selected from the full set. This can be helpful to reduce computational cost in large systems if one is only interested in on-site parameters: The full set (usually comprising Wannier functions at several atoms) can then be reduced to Wannier functions at a single atom (compare Section 5.5.2). Another use case would be to select a particular shell, e.g., the d subspace, from a larger Wannier set, e.g., `(s,p,d)`. At first, it might be surprising that a special keyword is introduced for that, since a more intuitive way would be to simply change the `ORBITALS` definition, e.g., from `(s,p,d)` to `(d)`. However, the d orbitals created with a definition `(d)` are different from the d orbitals created with `(s,p,d)`. This is because the Wannier construction (orthonormalization, disentanglement, maximal localization) is an optimization procedure that always involves the whole Wannier set. The Wannier orbitals are not created independently of each other. So, `SUBSET` allows you to pick a Wannier subset from a larger set, preserving the orbitals' shape. The keyword `SUBSET` expects a list of Wannier indices (see examples), following the order of the generated Wannier functions. (In most cases, this order is according to the `ORBITALS` definition, see the "orbital decomposition" in the output.)

Table 13: Examples

`SUBSET (5-9)`	Select full d shell from the set (s,p,d).
`SUBSET (1,5-7)`	Select s and t_{2g} functions from the set (s,p,t2g).

6.3.5 PLOT

Spex can plot the Wannier orbitals on a 3D grid and write it to XSF files for XCrySDen or Vesta ("wannier1.xsf", "wannier2.xsf", etc. or "wannier1s1,xsf", "wannier1s2.xsf", "wannier2s1,xsf", etc. in case of spin polarization). The plot region and resolution can be set with several optional arguments after `PLOT`: The first argument defines the units of the coordinate system: `UC` (unit cell), `SC` (supercell), or `CART` (cartesian coordinates in Bohr). (The default is `UC`.) The second argument defines the plot region. For example, `{0:1,0:1,0:1}` (this is the default) would set it to the unit cell (with `UC`), the supercell (with `SC`), or a cube with a side length of 1 Bohr (with `CART`), also see figure and examples. All coordinates are with respect to the origin of the coordinate system at (0,0,0). The resolution is set by the last three integers, giving the number of grid points along the three dimensions. The default is "20 20 20". (Note that 3D plots can be computationally expensive.)

Table 14: Examples

`PLOT`	Plot Wannier orbitals in the default plot region (unit cell) with $20 \times 20 \times 20$ points. (Same as `PLOT UC`.)
`PLOT 30 30 30`	Increase resolution. (Same as `PLOT UC 30 30 30`.)
`PLOT UC {-1:2,-1:2,0:0.5}`	Additionally, include the eight unit cells adjacent to the central one in the first two dimensions (e.g., xy plane) but cut the plot region at half the unit cell along the third dimension (e.g., z direction).
`PLOT SC {-0.5:0.5,-0.5:0.5,-0.5:0.5}`	Define the plot region to be the supercell but place the origin of the coordinate system into the center of the plot.
`PLOT CART {-1.2:2.5,-2.4:2.5,0:5} 100 100 100`	Define the plot region to be a cuboid with specified dimensions (in Bohr) and use high resolution.

6.3.6 RESTART

If `RESTART` is set, Spex writes the Wannier functions (the U matrix) to the file "spex.uwan" if it does not exist. Otherwise, the Wannier functions are read from "spex.uwan". In the same way the file "spex.kolap" is written or read containing the overlap of wavefunctions at neighboring k points. These overlaps are needed for the maximalization

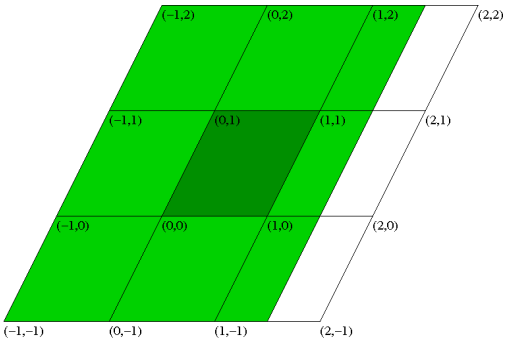

Fig. 2: Wannier plot region in two dimensions. The parallelograms are different unit cells (or supercells). The dark green area corresponds to the definition (0:1,0:1). The complete (dark and light) green area corresponds to (-1:1.5,-1:2).

procedure. (The file "spex.kolap" can be reused if the orbitals in `ORBITALS` are changed but not the outer energy window, i.e., the first two arguments.)

6.3.7 INTERPOL (WANNIER)

The keyword `INTERPOL` has already been discussed in Section 5.1.17. We explain it here for completeness and provide further advice on its usage.

When given, Spex performs a Wannier interpolation for the reference mean-field system (output to the file "bands0") and of the calculated *GW* (HF, COHSEX, et cetera) quasiparticle energies (output to the file "bands1"). It also gives, for each state, Wannier-function projections.

The quality of the Wannier interpolation depends crucially on whether the bands to be interpolated are representable in the set of Wannier orbitals defined by `ORBITALS`. Finding the right parameters requires experience and often some experimentation. A good set of parameters (as basis for further fine tuning) is usually found in the following way:

The bands to be interpolated should be predominantly described by some specific set of atomic orbitals, e.g., sp^3 hybrids in diamond structure, t_{2g} (e_g) bands in octahedral coordination, or d bands crossed by itinerant s bands (perhaps including p bands) in transition metals. These orbital characters are provided in round or square brackets for individual atoms or atom types, respectively, see Section 6.3.1.

The lower band index is usually easy to determine from the band structure (as it corresponds to the lowest band to be interpolated). In the case of an isolated group of bands (i.e., with band gaps below and above), the upper band index should obviously correspond to the topmost band, and a good interpolation is usually obtained with the `ORBITALS` definition and `MAXIMIZE`. Even just the first-guess orbitals (obtained without `MAXIMIZE`) sometimes gives a good interpolation in this case.

However, in most cases, the bands to be interpolated do not form an isolated set of bands, and the band structure is *entangled*. The upper (only in few cases also the lower) band index is then not uniquely defined. It is then recommended to simply set the upper index as a wildcard ($\star$) in the input file, which tells Spex to try to determine the index automatically. Often, the automatic choice is already good enough. Usually, the Wannier interpolation of entangled bands requires "maximally localized Wannier functions" that have been determined with a disentanglement procedure. So, we need in addition the keywords `MAXIMIZE` and `FROZEN`. The parameter for the latter (upper bound for frozen window), when omitted, will be determined automatically.

The keyword `INTERPOL` can be combined with `PROJECT` (Section 4.2.3) for the plot of "fat bands".

As an example, we give the section `WANNIER` for an interpolation of the valence band structure of iron. The result is shown in the figure above.

```
SECTION WANNIER
  ORBITALS 1 * (s,p,d)
  MAXIMIZE
  FROZEN
  INTERPOL
END
```

Table 15: Examples

`INTERPOL`	Perform Wannier interpolation of the input data (output "bands0") and any calculated energies, e.g., *GW* ("bands1"). The corresponding k-point path is defined by `KPTPATH`.
`INTERPOL "qpts"`	Same as before but the k-point path is taken from the file "qpts".
`INTERPOL ENERGY "energy.inp"`	Same as `INTERPOL` but the energies for the interpolation written to "bands0" are taken from the file "energy.inp". The file format is identical to the one used for the global keyword `ENERGY` (Section 4.2.6). In most cases, the resulting interpolation is identical to using `INTERPOL` together with the global `ENERGY "energy.inp"`, but the latter also reorders the eigenstates according to the energies, whereas the one-liner `INTERPOL ENERGY "energy.inp"` does not. (Another obvious difference is that `ENERGY` changes the energies globally for the whole Spex run, while `INTERPOL ENERGY` changes the energies only for the Wannier interpolation.) This reordering can affect the Wannier basis (if it is constructed in this run) or would lead to an inconsistency with a Wannier *U* matrix read from "spex.uwan". The latter case is caught by Spex, which would then stop with an error ("inconsistent checksum"). All these problems are circumvented by the one-liner.
`INTERPOL "qpts" ENERGY "energy.inp"`	Combine the last two cases.

6.3.8 BACKFOLD (WANNIER)

(*) The atoms of the basis of a Bravais lattice are not unique in the sense that one can always translate an atom by an arbitrary lattice vector from **r** to **r+R** and still describe the same periodic crystal with it. This means that any result (DFT, *GW*, ...) should be (and is) invariant with respect to such translations. However, there are exceptions for methods that rely on a real-space representation. For example, the Wannier interpolation (or its quality) *does* depend on the relative positions of the atoms. (Other examples are the BSE Wannier approach, the *GT* method, and off-site Hubbard *U* parameters.) This is because it is based on a real-space cutoff of Hamiltonian matrix elements $\langle w^a_{\mathbf{R}n}|H|w^{a'}_{\mathbf{R}'n'}\rangle$, where the two Wannier functions may be located at different atoms a and a'. (For clarity, we have included specific atom indices.) The cutoff criterion is with respect to $|\mathbf{R}-\mathbf{R}'|$. But the actual distance of the Wannier functions is $|\mathbf{r}_a+\mathbf{R}-\mathbf{r}_{a'}-\mathbf{R}'|$ with the atom positions $\mathbf{r}_a$ and $\mathbf{r}_{a'}$ in the unit cell. So, the relative location of the atoms (Wannier centers) do matter, and the interpolation works best when the atoms are as close to each other as possible. Therefore, Spex automatically exploits the translational freedom described above to shift the atoms so as to fulfill this condition as well as possible. However, Spex might not find the optimal atomic configuration, or the user might want to place the Wannier centers at certain atoms of his/her choice or avoid the atomic backfolding altogether (see examples). For this purpose, there is the keyword `BACKFOLD`, which expects as arguments one or more translation lattice vectors $\mathbf{R}_a$, maximally as many as the number of atoms. The Wannier centers will be placed at $\mathbf{r}_a + \mathbf{R}_a$. If there are less vectors than atoms in the unit cell, the remaining vectors are set to the null vector. (Note for experts: The specific Wannier centers are taken into account in the construction of the Wigner-Seitz supercell for Wannier interpolation in a procedure that is known as "use_ws_distance" in Wannier90. This procedure is always activated in Spex but can be switched off in "wannier.f" by commenting out the macro "WS_wancent".)

Table 16: Examples

`BACKFOLD (-1,-1,-1) (1,1,1)`	Shift first atom by $\mathbf{R}_1 = (-1,-1,-1)$ and the second atom by $\mathbf{R}_2 = (1,1,1)$. The other atoms (if any) are unshifted.
`BACKFOLD (0,0,0)`	Do not shift any atoms.

6.3.9 IRREP (WANNIER)

(*) The set of Wannier functions should reflect the symmetry of the system. This can be used to define matrix representations that determine how the Wannier functions transform when a given symmetry operation is applied. These matrix representations can be used to speed up the evaluation of certain Wannier-expanded quantities, for example, in spin-wave and *GT* calculations. (The name `IRREP` is a bit misleading because the matrices are not irreducible in general.) Unfortunately, Wannier90 tends to break the symmetry of the Wannier set, so `IRREP` can, in most cases, not be applied to MLWFs. However, unless the `ORBITALS` definition itself does not break the symmetry, the first-guess Wannier functions reflect the full symmetry, and `IRREP` can be used.

6.3.10 UREAD (WANNIER)

(**) This keyword makes Spex read Wannier functions generated by Fleur. (Works only with old version of Fleur. Currently unmaintained.)

Note: Paragraphs discussing advanced options are preceded with (*), and the ones about obsolete, unmaintained, or experimental options are marked with (**). You can safely skip the paragraphs marked with (*) and (**) at first reading.

CHAPTER 7

Parallelized version

7.1 General remarks

The parallelized version of Spex uses the MPI standard 3.1. It can be run on several CPUs on the same node or on several nodes with the command `mpirun` or `mpiexec` or whatever MPI launcher your computer system uses. In principle, there are no restrictions with respect to the number of processes. A better performance is expected, though, if the number of processes is not a prime number but has a long prime factorization because this gives the code more freedom to distribute the work among the processes.

The default parallelization strategy of Spex is conservative in the sense that memory demand and load imbalances are minimized. Often, the parallelized run can be sped up substantially by using `MPIKPT` or `MPIBLK`, see below.

7.2 Special MPI keywords

7.2.1 MPIKPT

In many calculation types (*GW*, Hubbard *U* calculations, COHSEX, ...) there is an outer loop over the k-point set. By default, Spex does not parallelize over this loop because different k points need different computation times depending on their symmetry, making the work distribution non-trivial. However, if there are many k points, it is recommendable to additionally parallelize over this loop. This can be enabled with the keyword `MPIKPT`. The k-loop parallelization is over nodes, not over processes. The computation for each individual k point runs in parallel over the processes on the respective node, in the same way as all processes would without `MPIKPT`. If you only have a single node (or very few nodes) available, you can still use `MPIKPT` in conjunction with `MPISPLIT`, which allows processes to be grouped into virtual nodes.

7.2.2 MPIBLK (SENERGY)

Another special parallelization layer is the parallelization over blocks of the self-energy matrix (or over the diagonal elements). This may speed up the calculation if there are many blocks but may also result in work imbalance. (Different blocks need different computation times.) Parallelization over blocks is enabled with the keyword `MPIBLK`. (An

optional argument, e.g., `MPIBLK 5`, can be used to fine-tune the work distribution. It gives the "relative computational overhead" of each block that does not scale with the number of bands. The default value is 10.) `MPIBLK` is enabled automatically except for `GW FULL` calculations, because the different sizes of self-energy blocks in `GW FULL` may lead to work imbalances. We note that `MPIBLK` increases the memory demand.

Table 1: Examples

`MPIBLK`	Enable parallelization over self-energy blocks (or diagonal elements).
`MPIBLK 50`	Enable parallelization with assumed large "computational overhead".
`MPIBLK 0`	Disable parallellization over blocks.

7.2.3 MPISPLIT

(*) The shared-memory functionality of MPI 3.1 is used for several big arrays, which allows the same memory region to be accessed by several MPI processes. By default, all processes running on one node share the memory. It can be reasonable to change this behavior to, e.g., having processes on the same socket or on the same NUMA domain to share memory. This is possible with `MPISPLIT NODE` (default), `MPISPLIT SOCKET` (only works with `OpenMPI`), or `MPISPLIT SHRD=16`, where, in this example, groups of 16 processes will share memory: the ranks 0-15, 16-31, etc. Using this option increases the memory consumption but might be advantageous in terms of memory bandwidth and computation time.

7.2.4 MPISYM (SENERGY)

(*) Using Padé approximants in the evaluation of the *GW* self-energy (`CONTINUE` or `CONTOUR` with Padé approximant for *W* or `FREQINT PADE`) might lead to a slight symmetry breaking in the quasiparticle energies, leading to unphysical lifting of degeneracies. (This is caused by the fact that Thiele's continued-fraction Padé formula is numerically unstable, especially for a large number of imaginary frequencies.) Since these errors are usually very small, this is not a big problem. Furthermore, when the full self-energy matrix is calculated (e.g., `GW FULL`), Spex performs a symmetrization of the self-energy matrix, which enforces the correct degeneracies again. However, for testing purposes, it is possible to enforce the correct symmetries already in the evaluation of the self-energy by using the keyword `MPISYM`. This requires additional communication among the processes, potentially slowing down the calculation due to the necessary blocking synchronization.

7.2.5 RESTART

In the parallelized version, the `RESTART` option works in exactly the same way as for the serial version (see Section 4.1.7, Section 5.1.13, and Section 5.5.3). However, the restart data might be written to separate files when `MPIKPT` is used. (The underlying reason for this is that binary or HDF5 files can be written in parallel, i.e., by all processes at the same time, only if the dataset sizes are known in advance. This is not the case for the restart data.) Instead of a single file "spex.cor", Spex writes the files "spex.cor.1", "spex.cor.2", et cetera, and a directory "spex.cor.map", which contains, for each k point, *links* to the respective *cor file* that contains the data. Furthermore, in addition to "spex.sigc" ("spex.sigx", "spex.wcou", "spex.ccou", "spex.core"), the files "spex.sigc.2", "spex.sigc.3", et cetera, might be written (and analogously for the other file names). These multiple files should be taken into account, when restart files are transferred. Switching between different numbers of processes, different numbers of nodes, or between serial and parallel runs should not lead to problems. Spex should be able to always read the correct data.

Note: Paragraphs discussing advanced options are preceded with (*), and the ones about obsolete, unmaintained, or experimental options are marked with (**). You can safely skip the paragraphs marked with (*) and (**) at first reading.

CHAPTER 8

Spex and Fleur

Spex needs input data about a reference mean-field system, for example, the KS system of DFT. A self-consistent mean-field calculation must thus precede any Spex calculation. The following three steps prepare the input data for Spex.

Step 1 Self-consistent mean-field calculation.

Step 2 Generate special k-point set with Spex.

Step 3 Diagonalize mean-field Hamiltonian with the potential from step 1, which produces the eigensolutions at the new k points.

8.1 Fleur

The Fleur code is an implementation of KS-DFT based on the FLAPW method. Consult the Fleur manual for details.

We distinguish between an old (v0.26b) and a new generation (MaX releases) of Fleur versions. In the long run, the old versions will become obsolete. However, as of now, the new Fleur versions have not been completely integrated. Therefore, both are documented here.

8.1.1 Fleur v0.26b (2019.03)

In the Fleur input file there is a flag for writing out the necessary data for a Spex calculation. It is called `gw` and is appended to line 23 of the Fleur "inp" file:

```
23|vchk=F,cdinf=F,pot8=F,gw=n,numbands=N
```

n can take four values:

- `gw=0`: No output files are generated (default; the same as omitting `gw` and `numbands` altogether).
- `gw=1`: Some basic parameters are written to several files. Otherwise Fleur runs as usual.
- `gw=2`: All output files necessary for Spex are created and Fleur stops after one iteration (without updating the potential). In this case you should also set `pot8=T` and the maximal number of bands `numbands=...`

- `gw=3`: Self-consistent cycle for QSGW. Same as `gw=1` but adds $\Sigma^{\mathrm{xc,QSGW}}(\mathbf{r},\mathbf{r}') - v^{\mathrm{xc}}(\mathbf{r})$ to the Hamiltonian in each iteration.

The output files are

- "gwa": Basic parameters including the atomic numbers and positions in the unit cell, lattice parameters and basis vectors, FLAPW l cutoff, and local-orbital parameters,
- "LATTC": FLAPW G cutoff,
- "radfun": radial basis functions,
- "ecore": core-electron functions,
- "eig": k points, wave-function coeffients (interstitial), energies,
- "abcoeff": wave-function coefficients (muffin tins),
- "KS.hdf": alternative to "eig" and "abcoeff" (now the default),
- "vxc": expectation values of the xc potential (diagonal elements, obsolete),
- "vxcfull": matrix of the xc potential,
- "qsgw": matrix of the QSGW self-energy (only for QSGW calculations).

Furthermore, Spex needs the Fleur file "sym.out".

8.1.2 Fleur MaX Release 3 v0.27

Spex has to be configured with `--with-dft=fleurR3` to compile the interface to Fleur MaX Release 3 (and 4) (see Section 1).

In the Fleur "inp.xml" file, there is a flag for writing out the necessary data for a Spex calculation. It can be found at the following XML-path `/calculationSetup/expertModes/@gw`, the `numbands` flag can be set at `/calculationSetup/cutoffs/@numbands`.

When using the input generator for fleur it is adviseable to set the option `-gw`. This writes out additional options in the "inp.xml", which already include the settings for an alternative k-point set ("kpts_gw").

The following modes are available:

- `gw=0`: No output files are generated (default).
- `gw=1`: The files "basis.hdf", "pot.hdf", and "ecore" are written. Needed for Spex to construct the "kpts_gw" file. The broyden history is reset.
- `gw=2`: All output files necessary for Spex are created and Fleur stops after generating the KS eigensolutions (without updating the potential).

The output files are:

- "basis.hdf": including all information about the basis (atoms, cell, kpts, muffin tin and planewave basis information),
- "ecore": core-electron functions,
- "eig_gw.hdf": energies, wave-function coeffients (interstitial), wave-function coefficients (atomic spheres),
- "pot.hdf": potentials (pottot, potcoul, potx) and relevant information on the stepfunction and structure.

Furthermore, Spex needs the Fleur file "sym.out", which contains the symmetry operations of the system.

Warning: Self-consistent *GW* calculations (also HF, PBE0, etc.) are not yet possible with the new Fleur code. Please use the old Fleur code for this feature.

A short tutorial can be found at: https://www.flapw.de/site/CECAM-2019-tut/Day_4_Hands-on_Spex/

8.1.3 Fleur MaX Release 5 version 33

Spex has to be configured with `--with-dft=fleurR5` to compile the interface to the Fleur MaX Release 5 (see Section 1).

The following changes from Fleur MaX Release 3 (and 4) apply:

- The Fleur "inp.xml" file now contains a `listName` entry for the requested k-point set defined in "kpts.xml". Spex writes its k-point set into "kpts.xml" with the name (`listName`) "spex". This supercedes the file "kpts_gw" of Release 3 (and 4). For the "gw=2" run, the entry `listName="spex"` has to be set in "kpts.xml". (Spex also writes its q-point path into "kpts.xml" with the name "spex_band".)
- "pot.hdf" now also contains the symmetry information. Spex does not need an external symmetry file ("sym.out") anymore.
- The lattice scaling factor (interpreted by Spex as the lattice parameter) is now undefined and generally set to 1.

The following modes are available:

- `gw=0`: No output files are generated (default).
- `gw=1`: The files "basis.hdf", "pot.hdf", and "ecore" are written. Needed for Spex to complement the "kpts.xml" file. The broyden history is reset.
- `gw=2`: All output files necessary for Spex are created and Fleur stops after generating the KS eigensolutions (without updating the potential).

8.1.4 Fleur MaX Release 5.1 version 34

Spex has to be configured with `--with-dft=fleur5.1` to compile the interface to the Fleur MaX Release 5.1 (see Section 1).

The following changes from Fleur MaX Release 5 apply:

- The flag `gw=...` in "inp.xml" has been renamed to `spex=....`
- The Spex path name in "kpts.xml" has been changed from "spex_band" to "spex-path".

Warning: Self-consistent *GW* calculations (also HF, PBE0, etc.) are not yet possible with the Fleur MaX versions. Please use the old Fleur code for this feature. Input data for $G^{\mathrm{SOC}}W^{\mathrm{SOC}}$ calculations cannot be provided by Fleur, but `ITERATE` can be used for that purpose.

8.2 Remarks about MT basis

8.2.1 Conduction states

As in a *GW* calculation the conduction states enter both the Green function and the screened interaction, they should be accurately described already in the KS system which is our starting point for the perturbation treatment. The LAPW

basis, however, only guarantees well described occupied states. In order to obtain well-converged *GW* results, one must make the basis set more flexible, especially in the MT regions, by introducing local orbitals either at high energy parameters or as higher-order energy derivatives [Phys. Rev. B 74, 045104 (2006), Comput. Phys. Commun., 184, 2670 (2013)].

8.2.2 Semicore states

High-lying semicore states can appear as badly described "ghost bands" in the valence band region leading to a wrong DFT ground state. A possible solution is to use local orbitals and extend the energy window, such that the semicore states are treated as valence states. On the other hand if the ghost bands are above the Fermi energy, they usually pose no problem in a DFT calculation. In a *GW* calculation, however, the conduction states are also relevant, and one must make sure that there are no ghost bands at all in the energy spectrum. Therefore, it might be necessary to treat more (and deeper) semicore states as valence states than are needed in the DFT calculation. As an indication for such a case the overlap of core states with basis and wavefunctions calculated in the routine "checkinput" should be checked. As an example, here is the output for Strontium Titanate with a Titanium 3s ghost band:

```
Overlap <core|basis>
Atom type  1
         u(s)    udot(s)
1s -0.001431  0.000644
2s -0.000830  0.000414
3s  0.287789 -0.878671
         u(p)    udot(p)   ulo(p) ...
2p -0.000665  0.000112 -0.000494
...
Maximum overlap <core|val> at (band/kpoint)
Atom type  1
1s  0.000611 (031/ 001)
2s  0.000343 (031/ 001)
3s  0.664265 (047/ 001)
2p  0.000340 (003/ 036)  0.000481 (003/ 042)  0.000340 (003/ 036)
...
```

CHAPTER 9

Spex FAQ

9.1 Configuration

Configuration stops with "MPI-3 compiler requested, but couldn't use MPI-3."

Your MPI compiler does not support the MPI-3 standard, which is required for the parallelized version. Make sure that the correct compiler is used. You can specify the compiler with the environment variables `FC` or `MPIFC`, e.g., directly on the command line `./configure --enable-mpi MPIFC=mpif90`.

Configuration stops with "Could not compile with HDF5."

This problem occurs quite often. It can be caused by one of the following reasons: (1) you do not have the HDF5 library, (2) you have HDF5, but it is not installed in a standard directory (then, use the option `--with-hdf5` to specify the HDF5 root directory, e.g., `./configure --with-hdf5=<path-to-hdf5>`, which should contain "<path-to-hdf5>/lib" and "<path-to-hdf5>/include"; alternatively, you can specify the location of the include file "hdf5.mod" with the option `--with-includedir=<path-to-hdf5-include>`), or (3) you have the HDF5 library, but it has been compiled with a different compiler than the one you are using for compiling Spex. In the latter case (3), please note that you might be using a different compiler than you think. (For example, even after loading a particular compiler module, some systems still use a different Fortran compiler, often GFortran, by default if you do not specify otherwise.) Then, you should use the environment variables `FC` or `MPIFC` as described before.

Configuration stops with "Could not link HDF5 library."

The linker does not find the HDF5 library. This can happen if the HDF5 library is installed in a non-standard directory (as is the case with many computer systems). In this case, you can use the option `--with-hdf5=<path-to-hdf5>` (see above) or you specify the location of the HDF5 library with the option `--with-libdir=<path-to-hdf5-lib>`. Consult your system administrator to inquire about the correct path.

Configuration stops with "Could not link parallel HDF5 library."

Your HDF5 library has been compiled without parallelization support. This is actually not an error anymore. Spex can be compiled with the serial HDF5 version.

You can get further information from the file "config.log".

9.2 Compilation

My compiler fails to compile the source.

In case of compiler errors, the first thing to try is cleaning up with `make clean` before running `make`. If this does not solve the problem, please inform us. We have successfully compiled and run the code with the Intel Fortran compiler (since version 12.1.3), GFortran (since version 4.9), and NAG Fortran (since version 6.2 6214). As of May 2018, PGI compilers have a bug (TPR 23745) that prevents compilation of the Spex code. For the parallelized version, we have used Intel MPI, MPICH, and Open MPI.

My compiler fails to link the object files.

If the configuration step has run successfully (with the same system configuration), linking the object files should work. In any case, before reporting a problem, please try cleaning up before compilation: `make clean ; make`.

I want to interface Spex to my own DFT program.

The data transfer between the DFT program and Spex is controlled by the routines in "readwrite_PROG.f" (where PROG stands for the name of the DFT program). You will have to create this file (if it does not exist yet). A template file "readwrite_tmpl.f" with explanations is provided with the source. The new file "readwrite_PROG.f" is included in the build process by reconfiguring (`./configure`) with the option `--with-dft=PROG`.

9.3 Usage

Spex needs too much memory.

Try reducing the maximally allowed memory storage with the keyword `MEM`. Have a look at the parameters of your calculation. Too high parameters can easily exhaust the computer memory. For example, the calculation of spectra usually requires the storage of large (polarization) matrices for each frequency of the mesh. Dividing the large frequency mesh into several smaller ones (see Sec. Section 5.3) might help in this case. Furthermore, note that the keyword `STOREBZ` requires a lot of memory.

Degenerate quasiparticle states have slightly different energies or I get slightly different results at equivalent k points

There are several possible reasons for this:

- In your DFT output groups of degenerate states are cut at the top. Then you should specify `NBAND` explicitly.
- In the calculation of the susceptibility groups of degenerate states are split into different packets. This can only be avoided with a larger `MEM` if possible.
- The tetrahedron method breaks the symmetry. You can use `GAUSS` to test this.

My GW calculations converge badly wrt the k-point set.

This might be caused by using the keyword `ZERO` for systems with small band gaps (e.g. GaAs). In this case, try without `ZERO`, which should guarantee a smooth (but slower) convergence.

My calculations do not converge wrt the number of bands.

The parameters are not completely independent. If you increase the number of bands, you might need a higher G cutoff (`GCUT` in section `MBASIS`). You might also have to increase the number of frequencies for sampling the spectral function. Instead, you can also specify the value of the first nonzero frequency point as the first argument to `FSPEC`. Then the number of frequencies automatically increases as you increase the number of bands.

The real-space (reciprocal-space) summation in the routine structureconstant takes very long.

Increase (decrease) `SCALE` in section `COULOMB`.

There are jumps in the band structure.

Band structures should be calculated without the keyword `ZERO`. In the case of a metal a jagged band structure (especially in Hartree-Fock) might also be caused by the tetrahedron method. Then you must improve the k-point sampling. Alternatively, you may try the Gauss integration method (`GAUSS`).

9.4 Error Messages

This is not a complete list of errors. In many error messages a possible solution is already indicated. These are not listed. The program might also stop at places where this is not supposed to happen. Then, the error message starts with `SPEX-BUG` (see Section 3.3). In this case you should contact the developers. Error messages have the form `SPEX-ERROR (source.f:0123) Error message`, where "source.f" is the name of the source file and 0123 is the respective line where the error occurred. We shorten this to `(source.f) Error message` in this list.

(read_write.f) k-point sets inconsistent (check . . .)

This error usually occurs if the k-point set defined in "spex.inp" (keyword `BZ`) conflicts with the one the DFT program has used for writing the input data. Make sure that the k-point sets are identical by generating the k-point set with Spex and running the one-shot DFT calculation in the proper order (steps 2 and 3 in Section 2). (Only for older versions:) The error can also be caused by the internal structure of the direct-access binary file containing the wavefunction data. Make sure that Spex and the DFT code are compiled in such a way that these files are treated identically.

(global.f) argument not an element of k-point set.

This error can be caused by inconsistent k-point sets, too. (See previous error message!) The error might also occur if the k points defined by `KPT` are not elements of the set (labels other than `+`), for example, if you use a definition with square brackets (such as `KPT A=[1,1,0]`), but the lattice constant is not properly defined (such definitions are interpreted with respect to the lattice constant). It is recommended to use internal coordinates (such as `KPT A=(0,0,1)`). See Section 5.1.2.

(coulombmatrix.f) Negative eigenvalue.

Your Coulomb matrix seems to be underconverged. Try to increase the parameter `LEXP`. If this does not solve the problem, the reason might be that the LAPACK routine fails to solve the general eigenvalue problem because of a nearly singular overlap matrix. Then you might try the option `CUTZERO` which removes linear dependencies from the overlap matrix and reattempts the diagonalization. The accuracy of the Coulomb matrix can be tested with the (undocumented) keywords `CHKCOUL`, `TSTCOUL`, and `STEPRAD` (all in section `COULOMB`).

(getinput.f) Defective symmetry transformation matrix.

A calculated symmetry transformation matrix turned out to be non-unitary. This is usually caused by insufficient precision of the lattice basis vectors provided by the DFT program. The obvious solution is to make sure that the lattice basis vectors are defined with sufficient precision in the DFT calculation. In particular, they have to conform to the crystal symmetries with a precision of at least twelve significant digits.

(getinput.f) Symmetry operation . . . rotates spin quantization axis.

Similarly to the previous one, this error is caused by insufficient precision: Spex expects the spin-quantization axis to be invariant with respect to the crystal symmetries in the case of non-collinear magnetism. Obvious solution: Make sure that the spin-quantization axis is defined with sufficient precision in the DFT run.

(irreps.f) Broken unitarity ...
(irreps.f) Deviation in characters; too large irrep threshold increase: ...

Both errors are caused by eigenstates (read from the DFT output) that are symmetry broken, i.e., they do not reflect sufficiently the crystal symmetries. This could be a problem of the DFT code. At the moment, there is no general solution to this problem. The second error can be circumvented by setting the parameter `irrep_threshold` in "irreps.f" to 0, which might lead to slower calculations. As a last resort, you can set the keyword `NOSYM` (Section 4.2.10), which switches off usage of symmetry.

(quasiparticle.f) Solution of quasiparticle equation did not converge after 5000 iterations.

If you use analytic continuation (`CONTINUE`), you probably have poles very close to the real frequency axis (normally with a small weight). Usually a slight change of parameters solves the problem. You can also try to use the keyword `SMOOTH` (Section 5.1.9). In the case of contour integration (`CONTOUR`), strong spectral structure in the self-energy can cause this problem. For example in metallic systems, a small Drude frequency can give rise to a delta-like peak in the self-energy. A possible solution in this case is to use `PLASMA METAL` (Section 5.2.3).

(Hwrapper.f) Fatal HDF5 error.

An error in the HDF5 library has occurred. There will be more informative error messages issued by the library itself. A possible reason could be that the HDF5 file misses a particular data set or attribute. In this case, make sure that the DFT program and Spex are called in the proper order (Section 2 and Section 8). If this does not solve the problem, please report the error. (Note that the same error can also occur in the source files "read_write.f", "correlation.f", and "correlation_wannier.f".)

The following error messages are displayed when the program run was terminated abnormally by the operating system.

error while loading shared libraries: lib...: cannot open shared object file: No such file or directory

The library "lib..." is not found in standard directories. In this case, the location of the library has to be defined in the environment variable `LD_LIBRARY_PATH` of the shell. (Consult your Unix or Linux manual.)

Segmentation fault

A forbidden area of memory was accessed during the program run. First, you should make sure that the operating system allows unlimited size of stack memory. You can set the stack size from the shell (BASH: `ulimit -s unlimited`, CSH: `limit stacksize unlimited`). As an alternative, some compilers provide options to prevent the usage of stack memory (e.g., `-heap-arrays` in the Intel Fortran Compiler). If this does not solve the problem, it is possible that the error is caused by a real bug in the Spex code. Please report such an error. You can help the developers by running the same calculation again with a Spex executable that has been compiled with `make ADD=-DCHECK` (slower execution). Please send the resulting output. Unfortunately, compilers and libraries can have bugs themselves. It is not unlikely that the "segfault" is caused by such a bug. Therefore, it is worthwhile to recompile with different compilers and/or libraries and run the program again.

CHAPTER 10

Keyword Reference

10.1 Global keywords

- ALIGNBD not documented
- BANDOMIT not documented
- BLOECHL not documented
- BZ Section 4.1.2
- CHKMISM Section 4.2.11
- CHKOLAP Section 4.2.11
- CORES Section 5.2.4
- CORESOC Section 4.2.2
- CUTZERO not documented
- DELTAEX Section 4.2.7
- ENERGY Section 4.2.6
- FIXPHASE Section 4.2.5
- GAUSS Section 5.2.7
- IBC not documented
- ITERATE Section 4.2.9
- JOB Section 4.1.1
- KPT Section 5.1.2
- KPTPATH Section 5.1.15
- MEM Section 4.1.4
- MPIKPT Section 7.2.1

- MPISPLIT Section 7.2.3
- NBAND Section 4.2.1
- NOSYM Section 4.2.10
- PLUSSOC Section 4.2.8
- RESTART Section 4.1.7
- STOREBZ Section 4.1.5
- TIMING not documented
- TRSOFF not documented
- WRITE not documented
- WRTKPT Section 4.1.3

10.2 Sections

10.2.1 ANALYZE

- DIPOLE not documented
- KINETIC not documented
- MTACCUR Section 4.2.12
- DOS Section 4.2.4
- PROJECT Section 4.2.3

10.2.2 COULOMB

- CHKCOUL not documented
- LEXP not documented
- MULTIPOLE not documented
- NOSTORE not documented
- STEPRAD not documented
- TSTCOUL not documented

10.2.3 LAPW

- EPAR Section 6.1.3
- GCUT Section 6.1.1
- LCUT Section 6.1.2
- LO Section 6.1.4

10.2.4 MBASIS

- ADDBAS Section 6.2.7
- CHKPROD Section 6.2.8
- GCUT Section 6.2.1
- LCUT Section 6.2.2
- NOAPW Section 6.2.6
- OPTIMIZE Section 6.2.5
- SELECT Section 6.2.3
- TOL Section 6.2.4
- WFADJUST Section 6.2.9

10.2.5 SENERGY

- ALIGNVXC Section 5.1.11
- CONTINUE Section 5.1.6
- CONTOUR Section 5.1.7
- DIVLOG not documented
- FREQINT Section 5.1.8
- MESH Section 5.1.5
- MPIBLK Section 7.2.2
- MPISYM Section 7.2.4
- ORDER not documented
- SMOOTH Section 5.1.9
- SPECTRAL Section 5.1.3
- VXC Section 5.1.12
- ZERO Section 5.1.10

10.2.6 SUSCEP

- DISORDER Section 5.2.8
- FPADE not documented
- FSPEC obsolete, replaced by `HILBERT`
- HILBERT Section 5.2.1
- HUBBARD Section 5.5.2
- MULTDIFF Section 5.2.2
- PLASMA Section 5.2.3
- TETRAF Section 5.2.5

- WGHTTHR Section 5.2.6

10.2.7 WANNIER

- BACKFOLD Section 6.3.8
- CUTGOLD not documented
- DISENTGL Section 5.5.4
- FROZEN Section 6.3.3
- INTERPOL Section 6.3.7
- IRREP Section 6.3.9
- MAXIMIZE Section 6.3.2
- ORBITALS Section 6.3.1
- PLOT Section 6.3.5
- RSITE Section 5.5.5
- SUBSET Section 6.3.4
- UREAD Section 6.3.10
- WBLOCH Section 5.5.6
- WSCALE not documented

10.2.8 WFPROD

- APPROXPW Section 6.2.11
- FFT Section 6.2.10
- LCUT Section 6.2.12
- MINCPW Section 6.2.13
- MPIMT not documented
- MPIPW not documented

CHAPTER 11

Selected publications

C. Friedrich, S. Blügel, and A. Schindlmayr, "Efficient implementation of the *GW* approximation within the all-electron FLAPW method", Phys. Rev. B 81, 125102 (2010).

C. Friedrich, A. Schindlmayr, and S. Blügel, "Efficient calculation of the Coulomb matrix and its expansion around k=0 within the FLAPW method", Comput. Phys. Commun. 180, 347 (2009).

C. Friedrich, M. C. Müller, and S. Blügel, "Band convergence and linearization error correction of all-electron *GW* calculations: The extreme case of zinc oxide", Phys. Rev. B 83, 081101 (2011); 84, 039906(E) (2011).

C. Friedrich, M. Betzinger, M. Schlipf, S. Blügel, and A. Schindlmayr, "Hybrid functionals and *GW* approximation in the FLAPW method", J. Phys.: Condens. Matter 24, 293201 (2012).

E. Sasioglu, C. Friedrich, and S. Blügel, "Effective Coulomb interaction in transition metals from constrained random-phase approximation", Phys. Rev. B 83, 121101(R) (2011).

T. O. Wehling, E. Sasioglu, C. Friedrich, A. I. Lichtenstein, M. I. Katsnelson, and S. Blügel, "Strength of effective Coulomb interactions in graphene and graphite", Phys. Rev. Lett. 106, 236805 (2011).

E. Sasioglu, C. Friedrich, and S. Blügel, "Strength of the effective Coulomb interaction at metal and insulator surface", Phys. Rev. Lett. 109, 146401 (2012).

M. Betzinger, C. Friedrich, and S. Blügel, "Hybrid functionals within the all-electron FLAPW method: Implementation and applications of PBE0", Phys. Rev. B 81, 195117 (2010).

M. Betzinger, C. Friedrich, S. Blügel, and A. Görling, "Local exact exchange potentials within the all-electron FLAPW method and a comparison with pseudopotential results", Phys. Rev. B 83, 045105 (2011).

M. Schlipf, M. Betzinger, C. Friedrich, M. Lezaic, and S. Blügel, "Implementation of a screened hybrid functional within the FLAPW method and its application to GdN", Phys. Rev. B 84, 125142 (2011).

I. Aguilera, C. Friedrich, and S. Blügel, "Electronic phase transitions of bismuth under strain from relativistic self-consistent *GW* calculations", Phys. Rev. B 91, 125129 (2015).

E. Sasioglu, A. Schindlmayr, C. Friedrich, F. Freimuth, and S. Blügel, "Wannier-function approach to spin excitations in solids", Phys. Rev. B 81, 054434 (2010).

C. Friedrich, E. Sasioglu, M. Müller, A. Schindlmayr, and S. Blügel, "Spin excitations in solids from many-body perturbation theory", Top. Curr. Chem. 347, 259-301 (2014).

C. Friedrich, M. C. T. D. Müller, and S. Blügel, "Many-body spin excitations in ferromagnets from first principles" in Handbook of Materials Modeling. Volume 1 Methods: Theory and Modeling, edited by S. Yip and W. Andreoni (Springer Berlin Heidelberg, 2018).

M. C. T. D. Müller, C. Friedrich, and S. Blügel, "Acoustic magnons in the long-wavelength limit: Investigating the Goldstone violation in many-body perturbation theory", Phys. Rev. B 94, 064433 (2016).

M. C. T. D. Müller, C. Friedrich, and S. Blügel, "Electron-magnon scattering in elementary ferromagnets from first principles: lifetime broadening and band anomalies", Physical Review B 100, 045130 (2019)

D. Nabok, S. Blügel, and C. Friedrich, "Electron-plasmon and electron-magnon scattering in ferromagnets from first principles by combining *GW* and *GT* self-energies", npj Comput. Mater. 7, 178 (2021)

C. Friedrich,"Tetrahedron integration method for strongly varying functions: Application to the *GT* self-energy", Phys. Rev. B 100, 075142 (2019)

R. Sakuma, C. Friedrich, T. Miyake, S. Blügel, and F. Aryasetiawan, "*GW* calculations with spin-orbit coupling: application to Hg chalcogenides", Phys. Rev. B 84, 085144 (2011).

I. Aguilera, C. Friedrich, G. Bihlmayer, and S. Blügel, "*GW* study of topological insulators Bi2Se3, Bi2Te3, and Sb2Te3: beyond the perturbative one-shot approach", Phys. Rev. B 88, 045206 (2013).

I. Aguilera, C. Friedrich, and S. Blügel, "Spin-orbit coupling in quasiparticle studies of topological insulators", Phys. Rev. B 88, 165136 (2013).

I. Nechaev, I. Aguilera, C. Friedrich, E. V. Chulkov, and S. Blügel, "Many-Body Effects in the Electronic Structure of Topological Insulators" in Topological Insulators: Fundamentals and Perspectives, ed. Frank Ortmann, Stephan Roche, Sergio O. Valenzuela ISBN: 978-3-527-33702-6 (Wiley 2015).

M. Betzinger, C. Friedrich, S. Blügel, and A. Görling, "Precise response functions in all-electron methods: Application to the optimized-effective-potential approach", Phys. Rev. B 85, 245124 (2012).

M. Betzinger, C. Friedrich, and S. Blügel, "Precise response functions in all-electron methods: Generalization to nonspherical perturbations and application to NiO", Phys. Rev. B 88, 075130 (2013).

M. Betzinger, C. Friedrich, A. Görling, and Stefan Blügel, "Precise all-electron dynamical response functions: Application to COHSEX and the RPA correlation energy", Phys. Rev. B 92, 245101 (2015).