The EXCITING Code Users' Manual
Version 0.9.74

J. K. Dewhurst, S. Sharma, C. Ambrosch-Draxl
Institut für Physik, Karl-Franzens-Universität,A-8010 Graz, Austria

Ch. Brouder
Institut de minéralogie et de physique des milieux condensés,
Université Pierre et Marie Curie, 4 place Jussieu,
75252 Paris cedex, France
Bessy GmbH, Albert-Einstein-Str. 15, 12489 Berlin, Germany

Contents

Chapter 1

Introduction

Welcome to the EXCITING code users' manual! The EXCITING code is a state-of-the-art full-potential linearised augmented-planewave (FP-LAPW) code for determining the properties of crystalline solids. It was developed mainly at the Karl-Franzens-Universität Graz as part of the EXCITING EU Research and Training Network project [1]. The guiding philosophy during the implementation of the code was to keep it as simple as possible for both users and developers without compromising on its capabilities. All the routines are released under either the GNU General Public License (GPL) or the GNU Lesser General Public License (LGPL) in the hope that they may inspire other scientists to implement new and, well, exciting developments in the field of density functional theory.

1.1 Acknowledgements

Lots of people contributed to the EXCITING code with ideas, checking and testing, writing code or documentation and general encouragement. They include Lars Nordström, Clas Persson, Christian Brouder, Rickard Armiento, Andrew Chizmeshya, Per Anderson, Igor Nekrasov, Fredrik Bultmark, Sushil Auluck and Frank Wagner. Special mention of David Singh's useful book *Planewaves, Pseudopotentials and the LAPW Method* [2] must also be made. Finally we would like to acknowledge the generous support of Karl-Franzens-Universität Graz, as well as the EU Marie-Curie Research Training Networks initiative.

Kay Dewhurst, Sangeeta Sharma and Claudia Ambrosch-Draxl

Graz, 2005.

Chapter 2

Compiling and running EXCITING

2.1 Compiling the code

Unpack the code from the archive file. Run the command

```
setup
```

in the `exciting` directory and select the appropriate system and compiler. We highly recommend that you edit the file `make.inc` and tune the compiler options for your particular system. You can also make use of machine-optimised `BLAS/LAPACK` libraries if they are available, but make sure they are version 3.0. Setting the OpenMP options of your compiler will enable EXCITING to run in parallel mode on multiprocessor systems. Following this, run

```
make all
```

This will hopefully compile the entire code and all the libraries into one executable, `exciting`, located in the `src` directory. It will also compile a few useful auxilliary programs, namely `spacegroup` for producing crystal geometries from spacegroup data, `species` for generating species files, and `eos` for fitting equations of state to energy-volume data. If you want to compile everything all over again, then run `make clean` from the `exciting` directory, followed by `make all`.

2.2 Running the code

As a rule, all input files for the code are in lower case and end with the extension `.in`. All output files are uppercase and have the extension `.OUT`.

For most cases, the user will only need to modify the file `exciting.in`. In this file input parameters are arranged in blocks. Each block consists of a block name on one line and the block variables on subsequent lines. Almost all blocks are optional: the code uses reasonable default values in cases where they are absent. Blocks can appear in any order, if a block is repeated then the second instance is used. Comment lines can be included in the input file and begin with the `!` character.

The only other input files are those describing the atomic species which go into the crystal. These files are found in the `species` directory and are named with the element symbol and the extension `.in`, for example `Sb.in`. They contain parameters like the atomic charge, mass, muffin-tin radius, occupied atomic states and the type of linearisation required. Users should not have to modify these files in the majority of cases.

The best way to learn to use EXCITING is to run the examples included with the package. These can be found in the `examples` directory and use many of the code's capabilities. The following section which describes all the input parameters will be of invaluable assistance.

2.3 Input blocks

This section lists all the input blocks available. It is arranged with the name of the block followed by a table which lists each parameter name, what the parameter does, its type and default value. A horizontal line in the table indicates a new line in `exciting.in`. Below the table is a brief overview of the block's function.

2.3.1 `atoms`

`nspecies`	number of species	integer	0
`spfname(i)`	species filename for species `i`	string	-
`natoms(i)`	number of atoms for species `i`	integer	-

`atposl(j,i)`
`bfcmt(j,i)`
	atomic position in lattice coordinates[1] for atom `j`
	muffin-tin external magnetic field in Cartesian coordinates for atom `j`
		real(3)
		real(3)
			-
			-

Defines the atomic species as well as their positions in the unit cell and the external magnetic field applied throughout the muffin-tin. These fields are used to break spin symmetry and should be considered infinitesimal as they

[1] When `molecule` is `.true.`, the atomic positions are in Cartesian coordinates

do not contribute directly to the total energy. Collinear calculations are more efficient if the field is applied in the z-direction. One could, for example, set up an anti-ferromagnetic crystal by pointing the field on one atom in the positive z-direction and in the opposite direction on another atom. See also `sppath`, `bfieldc` and `molecule`.

2.3.2 `autokpt`

`autokpt`	`.true.` if the **k**-point set is to be determined automatically	logical	`.false.`

See `rlambda`.

2.3.3 `autormt`

`autormt`	`.true.` if muffin-tin radii should be determined automatically	logical	`.false.`

This variable should be set to true if the muffin-tin radii are to be determined automatically. See `rmtapm` for details.

2.3.4 `avec`

`avec(1)`	first lattice vector	real(3)	$(1.0, 0.0, 0.0)$
`avec(2)`	second lattice vector	real(3)	$(0.0, 1.0, 0.0)$
`avec(3)`	third lattice vector	real(3)	$(0.0, 0.0, 1.0)$

Lattice vectors of the crystal in atomic units (Bohr).

2.3.5 `beta0`

`beta0`	initial mixing parameter and increment	real	0.1

This sets the initial parameter used for mixing the old and new potentials during the self-consistent cycle. For some materials, such as magnetic metals, this should be made smaller to avoid instability. The code automatically adjusts the mixing parameter to the optimial size. Making `beta0` too large can result in instability and poor convergence. See `betamax` as well as the routine `mixer` in the developers' guide.

2.3.6 `betamax`

`betamax`	maximum mixing parameter	real	1.0

The mixing parameter is adjusted in increments of `beta0` to optimise that rate of convergece. `betamax` sets the upper limit to this parameter. Making this too large can result in poor convergence due to charge sloshing.

2.3.7 bfieldc

`bfieldc`	global external magnetic field in Cartesian coordinates	real(3)	$(0.0, 0.0, 0.0)$

This is used to break spin symmetry for spin-polarised calculations. It should considered to be infinitesimal as it has no direct contribution to the total energy. This field is applied throughout the entire unit cell. To apply a magnetic field in particular muffin-tins use the `bfcmt` vectors in the `atoms` block. Collinear calculations are more efficient if the field is applied in the z-direction.

2.3.8 bfinite

`bfinite`	`.true.` if the external magnetic fields are to be considered finite	logical	`.false.`

If `bfinite` is `.false.` then the external magnetic fields `bfieldc` and `bfcmt` are taken to be infinitesimal (i.e. their only purpose is to break the spin symmetry). In this case the external field contribution to the total energy is set to zero. When `bfinite` is `.true.` everything is the same except that the integral over the unit cell

$$E_{\mathbf{B}\mathrm{ext}} = \int \mathbf{m}(\mathbf{r}) \cdot \mathbf{B}_{\mathrm{ext}}(\mathbf{r}) d\mathbf{r}$$

is added to the total energy.

2.3.9 chgexs

`chgexs`	excess electronic charge	real	0.0

This controls the amount of charge in the unit cell beyond that required to maintain neutrality. It can be set positive or negative depending on whether electron or hole doping is required.

2.3.10 deband

`deband`	initial band energy step size	real	0.0025

The initial step length used when searching for the band energy, which is used as the APW linearisation energy. This is done by first searching upwards in energy until the radial wavefunction at the muffin-tin radius is zero. This is the energy at the top of the band, denoted $E_{\rm t}$. A downward search is now performed from $E_{\rm t}$ until the slope of the radial wavefunction at the muffin-tin radius is zero. This energy, $E_{\rm b}$, is at the bottom of the band. The band energy is taken as $(E_{\rm t}+E_{\rm b})/2$. If either $E_{\rm t}$ or $E_{\rm b}$ cannot be found then the band energy is set to the default value.

2.3.11 `deltaem`

`deltaem`	the size of the **k**-vector displacement used when calculating numerical derivatives for the effective mass tensor	real	0.01

See `ndspem` and `vklem`.

2.3.12 `deltaph`

`deltaph`	the size of the atomic displacement used for calculating dynamical matrices	real	0.03

Phonon calculations are performed by constructing a supercell corresponding to a particular **q**-vector and making a small periodic displacement of the atoms. The magnitude of this displacement is given by `deltaph`. This should not be made too large, as anharmonic terms could then become significant, neither should it be too small as this can introduce numerical error.

2.3.13 `dos`

`nwdos`	number of frequency/energy points in the DOS or optics plot	integer	500
`ngrdos`	effective **k**-point mesh size to be used for Brillouin zone integration	integer	100
`nsmdos`	level of smoothing applied to DOS/optics output	integer	0
`wintdos`	frequency/energy window for the DOS or optics plot	real(2)	$(-0.5,0.5)$

DOS and optics plots require integrals of the kind

$$g(\omega_i)=\frac{\Omega}{(2\pi)^3}\int_{\rm BZ} f({\bf k})\delta(\omega_i-e({\bf k}))d{\bf k}.$$

These are calculated by first interpolating the functions $e(\mathbf{k})$ and $f(\mathbf{k})$ with the trilinear method on a much finer mesh whose size is determined by `ngrdos`. Then the ω-dependent histogram of the integrand is accumulated over the fine mesh. If the output function is noisy then either `ngrdos` should be increased or `nwdos` decreased. Alternatively, the output function can be artificially smoothed up to a level given by `nsmdos`. This is the number of successive 3-point averages to be applied to the function.

2.3.14 `dtauoep`

`dtauoep`	step length increment for the exact exchange iterative solver	real	0.5

See `maxitoep` and `tau0oep`

2.3.15 `eminvnl`

`eminvnl`	a core state with energy below this value is not included in the calculation of the non-local matrix elements or the OEP residues	real	-1×10^6

2.3.16 `epschg`

`epschg`	maximum allowed error in the calculated total charge beyond which a warning message will be issued	real	1×10^{-3}

2.3.17 `epsforce`

`epsforce`	convergence tolerance for the forces during a structural optimisation run	real	1×10^{-3}

If the mean absolute value of the atomic forces is less than `epsforce` then the structural optimisation run is ended. See `tasks`.

2.3.18 `epshf`

`epshf`	convergence tolerance for the average absolute value of eigenvalues in a Hartree-Fock calculation	real	1×10^{-4}

See `hartfock`.

2.3.19 epslat

epslat	vectors with lengths less than this are considered zero	real	10^{-6}

Sets the tolerance for determining if a vector or its components are zero. This is to account for any numerical error in real or reciprocal space vectors.

2.3.20 epsocc

epsocc	smallest occupancy for which a state will contribute to the density	real	1×10^{-8}

2.3.21 epspot

epspot	convergence criterion for the effective potential and field	real	1×10^{-6}

If the RMS change in the effective potential and magnetic field is smaller than epspot, then the self-consistent loop is considered converged and exited. For structural optimisation runs this results in the forces being calculated, the atomic positions updated and the loop restarted. See also maxscl.

2.3.22 evalmin

evalmin	valence eigenvalue minimum	real	-4.5

Any valence states with eigenvalues below evalmin are not occupied and a warning message is issued.

2.3.23 fixspin

fixspin	.true. if the spin moment should be fixed	logical	.false.

Set to .true. for fixed spin moment calculations. See also momfix, taufsm and spinpol.

2.3.24 fracinr

fracinr	fraction of the muffin-tin radius up to which lmaxinr is used as the angular momentum cut-off	real	0.25

See lmaxinr.

2.3.25 gmaxvr

`gmaxvr`	maximum length of $\|\mathbf{G}\|$ for expanding the interstitial density and potential	real	12.0

See also `rgkmax`.

2.3.26 hartfock

`hartfock`	`.true.` if a Hartree-Fock calculation is required	logical	`.false.`

Self-consistent Hartree-Fock is available as an experimental feature. See also `epshf`.

2.3.27 kstlist

`kstlist(i)`	ith **k**-point and state pair	integer(2)	$(1,1)$

This is a user-defined list of **k**-point and state index pairs which are those used for plotting wavefunctions and writing **L**, **S** and **J** expectation values. Only the first pair is used by the aforementioned tasks. The list should be terminated by a blank line.

2.3.28 lmaxapw

`lmaxapw`	angular momentum cut-off for the APW functions	integer	8

2.3.29 lmaxinr

`lmaxinr`	angular momentum cut-off for themuffin-tin density and potential on the inner part of the muffin-tin	integer	2

Close to the nucleus, the density and potential is almost spherical and therefore the spherical harmonic expansion can be truncated a low angular momentum. See also `fracinr`.

2.3.30 lmaxmat

`lmaxmat`	angular momentum cut-off for the outer-most loop in the hamiltonian and overlap matrix setup	integer	5

2.3.31 `lmaxvr`

`lmaxvr`	angular momentum cut-off for the muffin-tin density and potential	integer	7

2.3.32 `lradstp`

`lradstp`	radial step length for determining coarse radial mesh	integer	4

Some muffin-tin functions (such as the density) are calculated on a coarse radial mesh and then interpolated onto a fine mesh. This is done for the sake of efficiency. `lradstp` defines the step size in going from the fine to the coarse radial mesh. If it is too large, loss of precision may occur.

2.3.33 `maxitoep`

`maxitoep`	maximum number of iterations when solving the exact exchange integral equations	integer	30

See `tau0oep` and `dtauoep`

2.3.34 `maxscl`

`maxscl`	maximum number of self-consistent loops allowed	integer	200

This determines after how many loops the self-consistent cycle will terminate if the convergence criterion is not met. If `maxscl` is 1 then the density and potential file, `STATE.OUT`, will **not** be written to disk at the end of the loop. See `epspot`.

2.3.35 `molecule`

`molecule`	`.true.` if the system is an isolated molecule	logical	`.false.`

If `molecule` is `.true.`, then the atomic positions, $\mathbf{r}$, given in the `atoms` block are assumed to be in Cartesian coordinates. The lattice vectors are also set up automatically with the ith vector given by

$$\mathbf{A}^i = a_i \hat{\mathbf{e}}^i,$$

where

$$a_i = \max_{\alpha,\beta} \left| \mathbf{r}_i^\alpha - \mathbf{r}_i^\beta \right| + d_{\mathrm{vac}}$$

with α and β labeling atoms, and $d_{\rm vac}$ determines the size of the vacuum around the molecule. The last variable is set by the input parameter vacuum.

2.3.36 momfix

momfix	the desired moment in fixed spin moment calculations	real(3)	$(0.0, 0.0, 0.0)$

See fixspin, taufsm and spinpol.

2.3.37 ndspem

ndspem	the number of **k**-vector displacements in each direction around vklem when computing the numerical derivatives for the effective mass tensor	integer	1

See deltaem and vklem.

2.3.38 nempty

nempty	the number of empty states	integer	5

Defines the number of eigenstates beyond that required for charge neutrality. When running metals it is not known *a priori* how many states will be below the Fermi energy for each **k**-point. Setting nempty greater than zero allows the additional states to act as a buffer in such cases. Furthermore, magnetic calculations use the first-variational eigenstates as a basis for setting up the second-variational Hamiltonian, and thus nempty will determine the size of this basis set. Convergence with respect to this quantity should be checked.

2.3.39 ngridk

ngridk	the **k**-point mesh sizes	integer(3)	$(1, 1, 1)$

The **k**-vectors are generated using

$$\mathbf{k} = (\frac{i_1}{n_1}, \frac{i_2}{n_2}, \frac{i_3}{n_3}) + \mathbf{v}_{\rm off},$$

where i_j runs from 0 to $n_j - 1$ and $0 \le \mathbf{v}_{{\rm off};j} < 1$ for $j = 1, 2, 3$. See also reducek and vkloff.

2.3.40 ngridq

ngridq	the phonon **q**-point mesh sizes	integer(3)	$(1, 1, 1)$

Same as ngridk, except that this mesh is for the phonon **q**-points. See also reduceq.

2.3.41 nosource

nosource	when set to .true., source fields are projected out of the exchange-correlation magnetic field	logical	.false.

Experimental feature.

2.3.42 nosym

nosym	when set to .true. no symmetries, apart from the identity, are used anywhere in the code	logical	.false.

2.3.43 notes

notes(i)	the ith line of the notes	string	-

This block allows users to add their own notes to the file INFO.OUT. The block should be terminated with a blank line, and no line should exceed 80 characters.

2.3.44 nprad

nprad	radial polynomial order	integer	4

This sets the polynomial order for the predictor-corrector method when solving the radial Dirac and Schrödinger equations, as well as for performing radial interpolation in the plotting routines.

2.3.45 nstfsp

nstfsp	number of states to be included in the Fermi surface plot file	integer	6

2.3.46 `nwrite`

`nwrite`	number of iterations after which `STATE.OUT` is to be written	integer	0

Normally, the density and potentials are written to the file `STATE.OUT` only after completion of the self-consistent loop. By setting `nwrite` to a positive integer the file will be written during the loop every `nwrite` iterations.

2.3.47 `optcomp`

`optcomp`	the components of the first- or second-order optical tensor to be calculated	integer(3)	$(1, 1, 1)$

This selects which components of the optical tensor you would like to plot. Only the first two are used for the first-order tensor.

2.3.48 `phwrite`

`nphwrt`	number of **q**-points for which phonon modes are to be found	integer	1
`vqlwrt(i)`	the ith **q**-point in lattice coordinates	real(3)	$(0.0, 0.0, 0.0)$

This is used in conjunction with `task` $= 230$. The code will write the phonon frequencies and eigenvectors to the file `PHONON.OUT` for all the **q**-points in the list. The **q**-points can be anywhere in the Brillouin zone and do not have to lie on the mesh defined by `ngridq`. Obviously, you should have computed all the dynamical matrices first using `task` $= 200$.

2.3.49 `plot1d`

`nvp1d` `npp1d`	number of vertices number of plotting points	integer integer	2 200
`vvlp1d(i)`	lattice coordinates for vertex `i`	real(3)	$(0.0, 0.0, 0.0) \rightarrow (1.0, 1.0, 1.0)$

Defines the path in either real or reciprocal space along which the 1D plot is to be produced. The user should provide `nvp1d` vertices in lattice coordinates.

2.3.50 `plot2d`

`vclp2d(1)`	first corner (origin)	real(3)	$(0.0, 0.0, 0.0)$
`vclp2d(2)`	second corner	real(3)	$(1.0, 0.0, 0.0)$
`vclp2d(3)`	third corner	real(3)	$(0.0, 1.0, 0.0)$
`np2d`	number of plotting points in both directions	integer(2)	$(40, 40)$

Defines corners of the parallelogram and the mesh size used for producing 2D plots.

2.3.51 `plot3d`

`nup3d`	number of unit cells to plot	integer(3)	$(1, 1, 1)$
`np3d`	number of plotting points each direction	integer(3)	$(20, 20, 20)$

Defines the number of unit cells in each direction to be plotted in 3D as well as the size of the plotting mesh. The `nup3d` parameter is also used to define the number of reciprocal lattice unit cells to be plotted for Fermi surface plots.

2.3.52 `primcell`

`primcell`	`.true.` if the primitive unit cell should be found	logical	`.false.`

Allows the primitive unit cell to be determined automatically from the conventional cell. This is done by searching for lattice vectors among all those which connect atomic sites, and using the three shortest which produce a unit cell with non-zero volume.

2.3.53 `reducek`

`reducek`	set to `.true.` if the **k**-point set is to be reduced with the crystal symmetries	logical	`.true.`

See also `ngridk` and `vkloff`.

2.3.54 `reduceq`

`reduceq`	set to `.true.` if the **q**-point set is to be reduced with the crystal symmetries	logical	`.true.`

See also `ngridq`.

2.3.55 rgkmax

rgkmax	$R^{\rm MT}_{\rm min} \times \max(\|\mathbf{G}+\mathbf{k}\|)$	real	7.0

This sets the maximum length for the $\mathbf{G}+\mathbf{k}$ vectors, defined as rgkmax divided by the smallest muffin-tin radius.

2.3.56 rlambda

rlambda	maximum de Broglie wavelength of **k**-vectors	real	20.0

Used for the automatic determination of the **k**-point mesh. If autokpt is set to .true. then the mesh sizes will be determined by $n_i = \lambda/|\mathbf{A}_i| + 1$.

2.3.57 rmtapm

rmtapm	parameters governing the automatic generation of the muffin-tin radii	real(2)	$(0.25, 0.95)$

When autormt is set to true, the muffin-tin radii are found automatically from the formula

$$R_i \propto 1 + \zeta |Z_i|^{1/3},$$

where Z_i is the atomic number of the ith species, ζ is stored in rmtapm(1) and the value which governs the distance between the muffin-tins is stored in rmtapm(2). When rmtapm(2) $= 1$, the closest muffin-tins will touch.

2.3.58 scale

scale	lattice vector scaling factor	real	1.0

Scaling factor for all three lattice vectors. Applied in conjunction with scale1, scale2 and scale3.

2.3.59 scale1/2/3

scale1/2/3	separate scaling factors for each lattice vector	real	1.0

2.3.60 scissor

scissor	the scissors correction	real	0.0

This is the scissors shift applied to states above the Fermi energy. Affects DOS, optics and band structure plots.

2.3.61 scrpath

`scrpath`	scratch space path	string	`./`

This is the path to scratch space where the eigenvector file `EIGVEC.OUT` will be written. If the local directory is accessed via a network then `scrpath` can be set to a directory on the local disk, for example `/tmp/`. Note that the forward slash `/` at the end of the string must be included.

2.3.62 spinorb

`spinorb`	set to `.true.` if a spin-orbit coupling is required	logical	`.false.`

If `spinorb` is `.true.`, then a $\boldsymbol{\sigma} \cdot \mathbf{L}$ term is added to the second-variational Hamiltonian. See `spinpol`.

2.3.63 spinpol

`spinpol`	set to `.true.` if a spin-polarised calculation is required	logical	`.false.`

If `spinpol` is `.true.`, then the spin-polarised Hamiltonian is solved as a second-variational step using two-component spinors in the effective magnetic field. The first variational scalar wavefunctions are used as a basis for setting this Hamiltonian.

2.3.64 spinsprl

`spinsprl`	set to `.true.` if a spin-spiral calculation is required	logical	`.false.`

Experimental feature for the calculation of spin-spiral states. See `vqlss` for details.

2.3.65 sppath

`sppath`	path where the species files can be found	string	`./`

Note that the forward slash `/` at the end of the string must be included.

2.3.66 stype

`stype`	integer defining the type of smearing to be used	integer	0

A smooth approximation to the Dirac delta function is needed to compute the occupancies of the Kohn-Sham states. The variable `swidth` determines the width of the approximate delta function. Currently implemented are

0	Gaussian
1	Methfessel-Paxton order 1, Phys. Rev. B **40**, 3616 (1989)
2	Methfessel-Paxton order 2
3	Fermi-Dirac

2.3.67 `swidth`

`swidth`	width of the smooth approximation to the Dirac delta function	real	0.01

See `stype` for details.

2.3.68 `tasks`

`task(i)`	the ith task	integer	-1

A list of tasks for the code to perform sequentially. The list should be terminated with a blank line. Each task has an associated integer as follows:

-1	Write out the version number of the code.
0	Ground state run starting from the atomic densities.
1	Resumption of ground state run using density in `STATE.OUT`.
2	Structural optimisation run starting from the atomic densities, with atomic positions written to `GEOMETRY.OUT`.
3	Resumption of structural optimisation run using density in `STATE.OUT` but with positions from `exciting.in`.
10	Total, partial and interstitial density of states (DOS).
15	Output $\mathbf{L}$, $\mathbf{S}$ and $\mathbf{J}$ expectation values.
20	Band structure plot.
21	Band structure plot which includes angular momentum characters for every atom.
25	Compute the effective mass tensor at the $\mathbf{k}$-point given by `vklem`.
31, 32, 33	1/2/3D charge density plot.
41, 42, 43	1/2/3D exchange-correlation and Coulomb potential plots.
51, 52, 53	1/2/3D electron localisation function (ELF) plot.
61, 62, 63	1/2/3D wavefunction plot: $\|\Phi_{i\mathbf{k}}\|^2$.
72, 73	2/3D plot of magnetisation vector field, $\mathbf{m}$.
82, 83	2/3D plot of exchange-correlation magnetic vector field, $\mathbf{B}_{\rm xc}$.
91, 92, 93	1/2/3D plot of $\nabla\cdot\mathbf{B}_{\rm xc}$.
100	3D Fermi surface plot using the scalar product $p(\mathbf{k})=\Pi_i(\epsilon_{i\mathbf{k}}-\epsilon_{\rm F})$.
101	3D Fermi surface plot using separate bands (minus the Fermi energy).
110	Calculation of Mössbauer contact charge densities and magnetic fields at the nuclear sites.
115	Calculation of the electric field gradient (EFG) at the nuclear sites.
120	Output of the momentum matrix elements $\langle\Phi_{i\mathbf{k}}\|-i\nabla\|\Phi_{j\mathbf{k}}\rangle$.
121	Linear optical response tensor.
162	Scanning-tunneling microscopy (STM) image.
142, 143	2/3D plot of the electric field $E\equiv\nabla V_{\rm C}$.
200	Calculation of dynamical matrices on a $\mathbf{q}$-point set defined by `ngridq`.
210	Phonon density of states.
220	Phonon dispersion plot.
230	Phonon frequencies and eigenvectors for an arbitrary $\mathbf{q}$-point.

2.3.69 tau0atm

`tau0atm`	the step size to be used for structural optimisation	real	0.2

The position of atom α is updated on step m of a structural optimisation run using

$$\mathbf{r}_{\alpha}^{m+1} = \mathbf{r}_{\alpha}^{m} + \tau_{\alpha}^{m}\left(\mathbf{F}_{\alpha}^{m} + \mathbf{F}_{\alpha}^{m-1}\right),$$

where τ_α is set to `tau0atm` for $m = 0$, and incremented by the same amount if the atom is moving in the same direction between steps. If the direction changes then τ_α is reset to `tau0atm`.

2.3.70 tau0oep

`tau0oep`	initial step length for the exact exchange iterative solver	real	0.5

See `maxitoep` and `dtauoep`

2.3.71 taufsm

`taufsm`	the step size to be used when finding the effective magnetic field in fixed spin moment calculations	real	0.01

An effective magnetic field, $\mathbf{B}_{\rm FSM}$, is required for fixing the spin moment to a given value, $\boldsymbol{\mu}_{\rm FSM}$. This is found by adding a vector to the field which is proportional to the difference between the moment calculated in the ith self-consistent loop and the required moment:

$$\mathbf{B}_{\rm FSM}^{i+1} = \mathbf{B}_{\rm FSM}^{i} + \lambda\left(\boldsymbol{\mu}^{i} - \boldsymbol{\mu}_{\rm FSM}\right),$$

where λ is proportional to `taufsm`. See also `fixspin`, `momfix` and `spinpol`.

2.3.72 tfibs

`tfibs`	set to `.true.` if the IBS correction to the force should be calculated	logical	`.true.`

Because calculation of the incomplete basis set (IBS) correction to the force is fairly time-consuming, it can be switched off by setting `tfibs` to `.false.` This correction can then be included only when necessary, i.e. when the atoms are close to equilibrium in a structural relaxation run.

2.3.73 `tforce`

`tforce`	set to `.true.` if the force should be calculated at the end of the self-consistent cycle	logical	`.false.`

This variable is automatically set to `.true.` when performing structural optimisation.

2.3.74 `tsymctr`

`tsymctr`	set to `.true.` if the crystal should be shifted to the optimal symmetry center	logical	`.true.`

By default, the code searches atomic positions as well as mid-points between atoms for the center of highest symmetry, and then shifts the atomic basis vectors so that this point becomes the origin.

2.3.75 `usegdft`

`usegdft`	set to `.true.` if the generalised DFT correction of L. Fritsche and Y. M. Gu, Phys. Rev. **B** 48, 4250 (1993), is to be used	logical	`.false.`

Experimental feature – may not be working correctly.

2.3.76 `vacuum`

`vacuum`	the size of the vacuum region around a molecule	real	8.05

See `molecule`.

2.3.77 `vklem`

`vklem`	the **k**-point in lattice coordinates at which to compute the effective mass tensors	real(3)	$(0.0, 0.0, 0.0)$

See `deltaem` and `ndspem`.

2.3.78 `vqlss`

`vqlss`	the **q**-vector of the spin-spiral state in lattice coordinates	real(3)	$(0.0, 0.0, 0.0)$

Spin-spirals arise from spinor states assumed to be of the form

$$\Psi^{\mathbf{q}}_{\mathbf{k}}(\mathbf{r}) = \begin{pmatrix} U^{\mathbf{q}\uparrow}_{\mathbf{k}}(\mathbf{r})e^{i(\mathbf{k}+\mathbf{q}/2)\cdot\mathbf{r}} \\ U^{\mathbf{q}\downarrow}_{\mathbf{k}}(\mathbf{r})e^{i(\mathbf{k}-\mathbf{q}/2)\cdot\mathbf{r}} \end{pmatrix}.$$

These are determined using a second-variational approach, and give rise to a magnetisation density of the form

$$\mathbf{m}^{\mathbf{q}}(\mathbf{r}) = (m_x(\mathbf{r})\cos(\mathbf{q}\cdot\mathbf{r}), m_y(\mathbf{r})\sin(\mathbf{q}\cdot\mathbf{r}), m_z(\mathbf{r})),$$

where m_x, m_y and m_z are lattice periodic. See also `spinprl`.

2.3.79 `vkloff`

`vkloff`	the $\mathbf{k}$-point offset vector in lattice coordinates	real(3)	$(0.0, 0.0, 0.0)$

See `ngridk`.

2.3.80 `xctype`

`xctype`	integer defining the type of exchange-correlation functional to be used	integer	3

Currently implemented are:

1	No exchange-correlation funtional ($E_{\rm xc}\equiv 0$)
2	LDA, Perdew-Zunger/Ceperley-Alder, Phys. Rev. B **23**, 5048 (1981)
3	LSDA, Perdew-Wang/Ceperley-Alder, Phys. Rev. B **45**, 13244 (1992)
4	LDA, X-alpha approximation, J. C. Slater, Phys. Rev. 81, 385 (1951)
20	GGA, Perdew-Burke-Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996)
21	GGA, Revised PBE, Zhang-Yang, Phys. Rev. Lett. **80**, 890 (1998)
26	GGA, Wu-Cohen exchange (WC06) with PBE correlation, Phys. Rev. B 73, 235116 (2006)
30	GGA, Armiento-Mattsson (AM05) spin-unpolarised functional, Phys. Rev. B 72, 085108 (2005)

Bibliography

[1] EXCITING code developed under the Research and Training Network EXCITING funded by the EU, contract No. HPRN-CT-2002-00317, 2004.

[2] D. J. Singh. *Planewaves, Pseudopotentials and the LAPW Method.* Kluwer Academic Publishers, Boston, 1994.