# ***LEV00*** & ***TETR***:
# user-friendly packages for DFT codes
# ***VASP/SIESTA***

L. N. Kantorovich

October 16, 2007

This is a documentation for a package of utilities which have been written (initially at Keele and then continued in London at UCL and KCL) as pre- and post-processing tools for DFT codes working with periodic boundary conditions. At the beginning, DFT codes ***CASTEP*** and ***CETEP*** were supported. At a later stage a complete support for the ***VASP*** code has also been added. Then, the whole structure of the two codes has been reshuffled in order to allow for a more flexible support for other codes and input/output formats. Recently, an almost complete support for the ***SIESTA*** code has also been added. Starting from version 3.0 of ***lev00***, the support for the codes ***CASTEP*** and ***CETEP*** has been withdrawn. These codes are no longer supported.

The applications (i.e. offered functionality) include:

- various modifications of the system geometry (e.g. changing lattice vectors, breeding the system (supercells), merging geometries from several files of different formats (e.g. adding a molecule from ***SIESTA*** input file `H2O.fdf` to the surface ***VASP*** file `CONTCAR`), constructing a surface cell from a bulk one, etc.);
- construction of the system geometry files from the space group symbol;
- construction of molecular clusters with specific boundary conditions out of a bulk unit cell;
- a powerful engine for phonons calculations (molecules and crystals);
- calculations of the total and projected electronic Density of States (DOS),
- plotting of charge and spin total and partial densities, calculating atomic charges, saving the `cube` format file for ***gOpenMol*** contour plotting, etc.
- an intelligent search for surface states, exploration of the (partial) electronic (spin) density
- and a lot more!

# Contents

# Chapter 1

# Introduction

## 1.1 What is this all about?

Most of the modern condensed matter computer codes implement *periodic boundary conditions* (PBC). These are used not only in calculating the crystal bulk, but also for calculating single molecules, surfaces, molecules adsorbed on surfaces, and so on. Care is needed when working in PBC, especially if complex structures are to be modelled, such e.g. as steps, kinks, dislocations. On the other hand, a calculation of the electronic DOS or previewing electronic densities require specific tools that understand PBC and the way the DFT codes output their data.

Many workers in the field have their own little routines that do the job. I also started from those. However, I quickly realised that I cannot remember what was in each of the routines let along their names, input file formats, etc. That is why it became apparent that there is a need for a single user-friendly code that would do everything you might want in a way that would not require input files and remembering how to run it, would understand the input and output files of the DFT codes and would have a lot of useful features.

Initially, I combined all my little codes into a package that included **k**-point and unit cell generator and a DOS calculator. At the time, only the ***CASTEP*** code was supported. As the new features being developed and new problems arose, a lot of useful functrionality have been added. Most importantly, the interface in the package codes has been completely rewritten to allow for different formats. This way support for ***VASP*** and ***SIESTA*** has been allowed and was eventually added. I can now say that the real purpose of the package is to accompany any grid based plane wave or local basis set DFT code that utilise periodic boundary conditions, the codes like thos ementioned above.

The two most important codes in the package are ***tetr*** and ***lev00***. The first one is mainly to be used for *pre-processing* of the input data, e.g. complex modifications of your geometry input files, including the **k**-points, i.e. *before* a DFT run. The second code is used for the *post-processing*, i.e. calculating electronic DOS and charge density maps *after* the DFT run. As one particular application has been recently added, namely the phonon calculation engine, the ***tetr*** has gone long way beyond its initial simple pre-processing purpose and now does both things: it creates the necessary input files for ***VASP*** and ***SIESTA***, and then, after the calculation is performed with those codes, the ***tetr*** is run again to calculate the frequencies, eigenvectors and phonon DOS.

The most sophisticated support is provided for the old **CASTEP/CETEP** code. These are now redundant and their support was stopped a while ago. Presently, ***VASP*** and ***SIESTA*** are supported only.

## 1.2 Menus

The main codes ***tetr*** and ***lev00***, as well as some others, have a special tex-based interface (Text User Interface, TUI) that reminds menus. All menus work in a very simple and straightforward fashion: every menu item has either a number or a symbol. At each point in time you see on the screen the whole menu.

To execute a particular option you press the number (symbol) corresponding to the option required paying special attention to small/capital characters if present, and then press Enter. Normally this would cause some kind of calculation and the original menu would reappear (it may be different now!), so that something else can be done. In some other cases choosing an option invokes a different menu (sub-menu) or may change some settings - you should notice it when the original menu reappears!

Many menus contain several parts. Usually, the upper part of the menus contain the executable bits, the middle - some information options, while at the bottom of the menu one can find various options that control the other options of the menu (the settings). An example of a typical menu is given below (it is, in fact, the main menu of ***tetr***):

```
/--------------------------------------------------\\
|.................... MAIN MENU:.....................|
\\--------------------------------------------------/
>>>>>>>>>>>>> Create unit cell from scratch <<<<<<<<
[current geometry will be lost]
B. Generate supercell geometry from scratch
Pc.  Generate primitive cell from space group symbol
>>>>>>>>>>>>>>>>> Cluster construction <<<<<<<<<<<<
Cl.  Build up a cluster from the lattice
>>>>>>>>>>>>> Geometry file already exists <<<<<<<<<<
M. Complex cell modifications
KP. k-points generation
>>>>>>>>>>>>>>>>> Coulomb potential <<<<<<<<<<<<<<<
Ew.  Coulomb potential inside the unit cell
>>>>>>>>>>>>>>> Calculation of phonons <<<<<<<<<<<<<
Vm.  Vibrations of molecules => preparation
Vc.  Vibrations of crystals (point group) => preparation
Vs.  Vibrations of crystals (space group) => preparation
V2.  Vibrations => calculation [normal modes]
V3.  Vibrations => application [DOS, masses]
>>>>>>>>>>>> Representation/visualisation <<<<<<<<<<<
Cm.  Compare geometry with the one in another file
D. Distances between atoms
Ch.  Check specified point in the cell
Sb.  Save in a box running across adjacent cells
>>>>>>>>>>>>>> Potential energy surface <<<<<<<<<<<<<
```

```
PE. PES: move atoms keeping the symmetry
---------- g e n e r a l   s e t t i n g s -----------
XY. Format of <geom.xyz> is set to Xmol
R. Read geometry from another file
S. Save
Q. Quit.
Specify the character and press ENTER ------->
```

We shall be using a bold font for the menu options in what follows.

## 1.3 Miscellaneous

At the moment the package can be used on many computers where graphical package ***gnuplot*** is available. Some tricks employed in the code (like the internal write/read statements, for example, or reading from the command line) which make the code user-friendly, may be platform-dependent, but these should be ok in Linux. Make files are available for Linux.

I recommend to put all the source in the directory ~/TOOLS in your home directory and all necessary scripts (see directory SCRIPTS) - in your ~/bin directory. If you have done this, the scripts may work straight away.

# Chapter 2

# Versions history

## 2.1 Utility *tetr*

### Version 3.97.5 11.05.2007

- **NEW**: IR intensity is implemented within the **V3** menu
- bug fixes.

### Version 3.97.3 12.04.2007

- **NEW**: `get_forces` routine reads the final (last) forces in the ***VASP/SIESTA*** output file; it also checks if the structural relaxation successfully finished and the tip force is only printed out if this is so; atomic forces are read in by `get_forces` in the `run_AFM` script and the tip force is calculated.
- **NEW**: Tip height is specified within the **AF** option (the new option **Z**); the value corresponding to the difference in height between the highest surface and the lowest tip atoms is suggested. The value is used in the AFM script as the actual tip height when collecting forces and saving in files
- some bug fixes

### Version 3.97.2, 4.03.2007

- **NEW**: reading/writing of the constraints box from/to ***SIESTA*** `fdf` file; fixed atoms are checked immmediately in the **M** menu and shown
- **NEW**: one can choose the atoms to be fixed in the **M** menu
- **NEW**: new options **FT**, **FS** and **S** are added to the **AF** menu (AFM script) to choose fixed atoms in the tip (**FT**), surface (**FS**) and to save the input file (**S**).

**Version 3.97, 21.02.2007**

- **NEW**: comma added as a delimiter in tag lists (option **T** and some others); this was done to faciliate using ***tetr*** in scripts (Section 3.7).

- **NEW**: AFM option in the main menu (option **AF**) which creates a shell-script `run_AFM` for running tip on the 3D grid with respect to the surface; the script creates directories, files, etc. and performs actual calculations. Works with ***VASP*** and ***SIESTA***.

**Version 3.96, 12.02.2007**

- **NEW**: option **Ss** added in the main menu to combine ***SIESTA*** `fdf` and `XV` files into one `INPUT_FILE-relaxed.fdf` file; useful for creating a single `fdf` file after relaxation containing all the same options and the new geometry (e.g. to move on with vibrations).

**Version 3.95, 07.01.2007**

- A bug in 'reading the rest in siesta file' corrected.

- **NEW**: **k**-points and weights within the DOS option of **KP** are generated for ***SIESTA***; they are written into a special block `BandPoints`.

**Version 3.94a-3.94h16, 13.11.2005 - 18.12.2006**

- **NEW**: option **Pc** added to the main menu (cell construction from the space group symbol). Also added a check of the existence of two new files (`space.gr.int`, `gropup.list.int`) required for the **Pc** option to work (space group information); corresponding changes in ***do_tar_improved***, ***do_tar***. 3D space group generators were checked against the International Tables; the fractional tranlsations are given with respect to the conventional cell; definitions of direct lattice vectors have been also changed appropriately. Several options added to the **Pc** menu to get information about the space group (options **Sy**, **Re**, **Ca**).

- 2D space groups added and checked against the Tables; **2D** option added to the **Pc** menu. Option **Ad** added to the **Pc** menu.

- Various bugs corrected, including an embarrassing bug concerning generation of 2D reciprocal lattice vectors.

- Option **Pc** largerly rewritten. New options in **Pc** added (**S**, **T**). Tags added in **Pc**.

- Bugs fixed related to tagging in **M**.

- Options **Rm** and **Rn** added to **Pc**.
- Added options **Mv** and **Rt** into **Pc**.
- **R** in **M**-menu was changed into **Rt**.
- Change default to <Fractional> for the **An** option in the **B** menu; title of menu **7** changed.
- Menu **Vm**, **Vs**, **Vc**: after an irrep is chosen, it is possible to deactivate it by choosing the same irrep the 2nd time; useful, if decided to run **Au** for all irreps.
- Portability problem fixed (e.g. a redundant option **FE** in **V2** did not work).
- Print statement fixed for the lattice vectors in the lattice **B** menu.
- **NEW**: option **Sf** added to **Pc** and **M** menus: it generates a fortran code for ***SIESTA*** `constr.f` routine to keep symmety in the system during geometry optimisation (symmetrise forces).
- Used `character*(*)` construction for all string tools instead of explicit specification.
- Species 'XX' is introduced when the Mendelev's number is beyond 82; a warning is given; ***tetr*** does not change species to ' ' if not recognised, but instead keeps it as entered; when saving ***SIESTA*** input file, a warning is given if species is not recognised.
- Tools have been written to check symmetry information (groups, irreps) (in directory `Symmetry`).
- An error in irreps matrices and characters of $C_{6h}$ was corrected.
- **NEW**: stress symmetry option **St** added to **Pc** menu to check symmetry on the stress tensor; after that, this option was removed (the file with this option is still in `./Stress`).
- **NEW**: conditions on the stress tensor added to option **Sf** of **Pc**
- Group $S_6$ added to `symgrp.rep` file, not yet to other places.
- **NEW**: division of the system into "parts" is implemented in **M**, having mainly in mind the BSSE calculation with ***SIESTA***.
- **NEW**: option **B2** added in **M**: writes input ***SIESTA*** files and the script ***BSSE_script.run*** which contains completely automatic calculation of the BSSE correction. At the same time, the density (`RHO` file) can also be saved; this is asked when the script is being created.

- **NEW**: a separate option **B3** in **M** added for the difference density script; generates directory structure, input files and the script (both ***SIESTA*** and **VASP**).

### Version 3.93-3.93a, 13.7.2005 - 18.7.2005

- Bugs fixed.
- A new option **Mv->UU** added; also, when leaving the option **Mv**, the old geometry (at the entry) will be restored.
- The title of item **4** in the point group menu clarified.
- Routine `specify_angle()` completely rewritten.
- Option **R** of the **M**-menu was renamed into **Rt**.
- Option **5** from the **M**-menu removed.
- The routine calculating the Madelung potential has been substantially modified to accomodate the same logic as other ***tetr*** routines; new options **Co**, **Sy** added.
- A completely new manual released - fully up-to-date!

### Version 3.92d-h, 13.6.2005 - 21.6.2005

- Option 'via energy' in vibrations is disabled - there are some bugs left after many modifications in 'via forces' option, I do not feel those would actually be necessary to fix;
- Reading an integer out of a character seems to give sometimes zero rather than an error; put a fix for this, but not sure if it is ok in general;
- compatibility and other bugs fixed;
- `ITI` is initialised to 0 at the beginning; if a geometry exists before entering the **B** option of the main menu (creating from scratch), then the existing geometry is kept in a buffer and can be recovered at the end of the construction process when going back (up) to the main menu; as a result, two options to leave appear: **P** (proceed from the current one) and **Q** (quit, restore the previous one).
- Edited help after **Vs->Au** option.
- Reading the irrep numbers in the menus of **Vm**, **Vc**, **Vs** is done without using fancy routines - could not make them work properly anyway since they are still platform dependent; to be worked out someday!
- In calculating phonon DOS, imaginary frequencies are ignored.

- Ge mass is added to the data base, all entries commented directly.
- In option **V3** -> **Am** it is now not possible to change the mass of a fixed atom as it does not make sense anyway.
- In option **V3** -> **Ei**, the natural logarithm of a product of all real frequencies (in cm$^{-1}$) is printed out for the transition rate calculations.
- In **V3**: it is checked if the order of atoms is different from the input file, and if it is, a warning is printed at the top of the menu: it is important if atomic masses are changed (option **Am**) or projected DOS is calculated (option **pD**, then **D2**).
- In **V3**: a new option **W** is added that dumps atomic positions into a `geom.xyz` file; this is useful if atomic numbering is different from the input file.

### Version 3.92c, 13 June 2005

- **NEW**: routine `hat()` modified to give a number of useful things including the position of the home directory of ***tetr***; from now on the enviromental variable `$HOME_TETR` must be set to the correct position of the ***tetr*** directory.
- ***tetr*** is called from inside via `$HOME_TETR/tetr`.
- The script ***collect_forces*** is called for all relevant irreps from inside; no need then to run it manualy anymore; so far, this should only work for ***VASP***; the script was also modified by attaching `$HOME_TETR` to ***get_forces***; thus, no need to do soft links anymore! Great relief!
- Tables in options **Sa** of **V3** are printed by components for convenience - was not actually done in the previous version.
- ***SIESTA*** was incorporated into the vibrational options: firstly, ***get_forces*** routine modified to determine automatically which code (so far, only ***VASP*** or ***SIESTA***) is being used for forces; secondly, ***run_DFT_script*** script was modified accordingly to create an `OUTCAR` file from ***SIESTA*** standard output (the only place that contains forces).
- **NEW**: from now on, each version is to be assembled using ***tmake*** script rather than directly using ***make***; this gives the version number and the build date. Users must still use ***make*** without changing any files supplied.
- Script ***collect_forces*** was enforced with a number of checks (existence of files, directories, `OUTCAR` files correctness).
- When reading ***SIESTA*** input file, the screen freezing (”hit ENTER when ready...”) was removed to facilitate running ***tetr*** from inside for the **Au** options of **Vm**/**Vc**/**Vs**.

- A bug in writing input file name for siesta in ***run_DFT_script*** fixed; there was also a bug in writing the input ***SIESTA*** file into `input_for.tetr` file that was also corrected. 14.4.2004, version 3.40
- a bug in related to tagged atoms in **Co** corrected
- a new option **Sb** in the main menu to save atoms in a box running across the adjacent cells of the system in the `xyz` format.

**Version 3.91, 11.6.2005**

- A little bug in option **Fx** of vibration menu fixed.
- **NEW** development initiated: option **Vs** of the main menu added that would do vibrations using internal translations of the supercell. Main advantage: every **k**-point (corresponding to the Gamma point of the reduced BZ) gives its own irrep, so that all vibrations will be characterised by **k**-points automatically.
- A default displacement in option **Da** of **Vm/Vc/Vs** menus changed from 0.02 to 0.01
- A small bug that may affect precision in `solve_vibr()` fixed.
- Increased dafault values in `sym_par.inc`.
- Diaginalisation of the dynamical matrix for the chosen irrep in option **V2** is performed automatically upon choosing (activating) the irrep, so that option Ei in **V2** of the main menu was removed.
- A bug in option **Su** of **M** related to construction of supercell lattice vectors, fixed.
- Tables in options **Sv**, **Sa** of **V2**, **V3** are printed by components for convenience.
- When ***tetr*** in **Vm**, **Vc**, **Vs** is launched from inside, all output is sent to the file [out] (before, some of it was still sent to the screen); this avoids some annoying error messages showing up on the screen.
- When reading irrep number in menu **V2**, a restriction on the max irrep number (= to 12) removed; it could be any number now; this is necessary for option **Vs** that allows for very large numbers of irreps possible.
- **NEW**: routine `hat()` changed to faciliate an automatic detection of the version number and the build date; it will also show the todays date every time ***tetr*** is used.

This version was checked on a number of examples in ***MM***. Seems to work! Options **Vs** and **Vc** give the same results.Version 3.51, 21.10.04

- Bug fixes in options **V1** and **V2**;
- Option **Sx** added to **V1**, **V2** menus to 'show'/'not show' fixed atoms when displaying generalised coordinates or normal modes.
- A more geneal format for the files `F_#.mat` added (algorithm Forces): forces should be the last three columns in each block. This is the case in most DFT codes in which forces are shown. With this option in place, one can simply cut and paste the lines with forces from e.g. the ***VASP*** or ***SIESTA*** output files into the appropriate `F_#.mat` file.
- Atomic masses are entered automatically (at least for the first half of the Table) from mendeleev.inc; consequently, some changes in **V1** were required and in some other routines where atomic masses were asked.
- Small changes to the option Am of the **V1** menu.
- Bugs in the `Makefile` (missing dependencies) corrected.
- Bugs in `restore_geometry` routine fixed.

**Version 3.81, 5.6.2005**

- Atomic displacements in phonon eigenvectors appear in the order of atomic coordinates x,y,z for convenience (in tables of **Ea** and in the files `N_MODES_#.res`).
- A bug in option **Cm** of the main menu corrected; also, some cosmetic changes in the table there.
- A new option **Rf** added to the menu **M** that allows correcting atomic positions relative to a reference file of any format. This is ussually needed for ***VASP*** users as ***VASP*** likes changing atomic fractional coordinates during relaxation to fit between 0 and 1. This is NOT very convenient e.g. for slab calculations. This same option exists in ***do_param***, but I thought it would be useful to have it in ***tetr*** as well. The extra flexibility (does not mean of course that it is badly needed!) is that you can compare e.g. ***VASP*** file with that of ***SIESTA***.
- Options **7** and **77** of the **M** menu renamed as **Ex** and **Su**, respectively.
- A small (and not very important) bug with translating relaxation flags when breeding the lattice using option **Su** in **M** menu fixed.
- Yet another bug in option **Su** of the **M** menu corrected.
- A bug in routine `ask()` fixed - do not know where it has come from!

### Version 3.8, 28.5.2005

- some unexplained bug in `do_input_scripts()` corrected (with **Da**) - I changed **Ds** -> **Da** in the main menu in **Vm/Vc** after releasing the first stable version without checking; it should have gone all the way down to `do_input_scripts()`.

### Versions 3.76-3.77, 15.2.2005

- **NEW**: **V3** menu started to take shape; mass and DOS added, projected DOS added.
- A bug in closing the `vibr_info.dat` file fixed; some other bugs fixed related to atom numbering in **V3**.

This is a stable working version that supports V3 in full including projected DOS and `movie.xyz` option. What is now required is a manual, will do shortly.

### Version 3.75, 30.1.2005

- Some small changes in the tools routines.
- Option **Vm** replaces **V1** in the case of molecules in the main program.
- **NEW**: option Vc added to do vibrations for crystals.
- ICMPL array has been put back. It is needed if a group has 2 complex conjugate irreps that are artificially made into a 2D real reducible one. It is needed for a number of point groups. [actually, it is not used at all! - something to look up later!]
- Option **Am** in `mol_vibr()` is made into a menu to benefit from existing masses.
- Cleaning up of the symmetry routines.
- Makefile updated.

It is a stable version for molecules: option **Vm** should definitely work. Option **Vc** has not yet been carefully tested, so that it may not work.

### Version 3.74, 27.1.2005

- A bug in `get_all_from_siesta()` routine corrected.

### Version 3.73, 15.1.2005

- **NEW**: option **V3** in the main menu added to calculate the vibrational frequencies with the old foce-constant matrix but different masses (e.g. isotope shift effect) - just a test part, restoring the whole fcm matrix from its symmetrical parts.
- Better explanations when generating **k**-points for the BAND option of **KP** menu.
- BAND option in the **KP** menu: devision by zero avoided.

### Version 3.72, 10.1.2005

- In the case of 4 displacements, both amplitudes can be chosen in **V1**.
- Option **Au** in **V1**: asks about the type and name of the input file using the standard 'save' menu; thus, `input_for.tetr` file does not require editing and is executed from inside ***tetr***.

**Version 3.71, 5.1.2005** This is a correctly working version: **V1**, **V2** were checked on C defect in a Si-cluster.

- bug fixed in reading forces;
- a numerical prefactor to the frequencies in $\mathrm{cm}^{-1}$ corrected.

### Version 3.7, 3.1.2005

- Some bugs fixed.
- Printing of the tables of eigenvlues/vectors in options **Sv** and **Sa** of the **V2** menu is made by parts in the cases of large numbers of eigenvectors;
- in addition, the results are dumped into files `EIGEN_irrep.res` (option **Sv**) and `N_MODES_irrep.res` (option **Sa**) for further reference.

### Version 3.61, 1.1.2005

- Default for the relaxation amplitude was changed to 0.02 for the option **V1** (works better for ***VASP***, it seems).
- A bug in `get_forces.f` fixed.
- **NEW**: an automatic option **Au** in **V1** is modified to have it done for all irreps at once; also, **Gw** and **Sw** options are also inculded inside, so that it became really automatic.

- **NEW**: a new code, ***MM***, based on the ***sci-fi*** code, is written in order to facilitate testing the **V1**, **V2** options.
- Some cosmetic changes in **V1**, **V2** menus; in particular, dimension of irreps is shown in menus + little help is given on the screen after **Au** option.

**Version 3.6, 20.12.04**

- **Cl** option: added menu option **W** to write `geom.xyz` file anyway
- Two new examples created (Si cluster and O defect in a Si cluster) to debug the **V1** and **V2** options.

**Version 3.54, 13.12.04**

- Minor changes in routines supporting **V1** and **V2** options.
- **NEW**: oprion **Au** in **V1** added:
  - to create an input file `input_for.tetr` for automatic generation of all geometry files for ***VASP/SIESTA*** by ***tetr*** for the given irrep: to be run as: ***tetr*** `< input_for.tetr`. Asks about the code (v/s) and method (Energy/Force). The file should be edited for the ***VASP/SIESTA*** input file (relaxed geometry);
  - 1, 2 and 4 displacements are allowed (option **Nd**);
  - a script ***run_DFT_script*** is created to run ***VASP/SIESTA*** automatically on all these input files. Should work with ***VASP***; some additional editing is necessary for ***SIESTA***;
  - creates a directory superstructure for the vibrations calculations.
- Script ***collect_forces*** modified appropriately: several amplitudes are allowed. The file `F_#.mat` will contain all the data, its format is slightly different.
- Correspondingly, the new format `F_#.mat` file is read in. Depending on the number of amplitudes used (1,2 or 4), different expressions are used for the calculation of the Hessian matrix.

**Version 3.53, 15.11.04**

- Cosmetic changes in the LATTICE menu.
- Cosmetic changes in routine `mol_vibr`.
- Position of the molecule in the centre of mass is calculated with real (instead of unit) masses for option **V1**.

- A number of bugs fixed.

- Options **V1** and **V2**: when writing `movie.xyz` file for a movie of the given vibrational mode, the whole 360 degrees period is written which is more convenient for previewing.

- Option **V2**: file tmp.tmp is to be deleted after being used by gnuplot.

- Comments starting from ## can be put arbitrarily in `F_#.mat` and `E_#.mat` files (e.g. the value of amplitude used for distorting geometry).

- **NEW**: to facilitate the **V2** option with the algorithm Force, a script ***collect_forces*** and a separate routine (used by the script) ***get_force*** were added. These are to be used to create automatically `F_#.mat` files out of the corresponding `OUTCAR` files written by ***VASP*** . (Some editing of the file `get_force.f` may be required if another DFT code is to be used.) Note that ***get_force*** is compiled at the same time as ***tetr***.

- Option **V2**: when the forces are read in from `F_#.mat` file, available elements before symmetrisation are shown even if not all `OUTCAR` files are available, even though the complete force constant matrix (FCM) cannot be constructed. This is useful for finding an appropriate distortion displacement in **V1**, as it allows to check if the available part the FCM is symmetircal or not.

**Version 3.52, 8.11.04**

- Bugs in the routine [`specify_angle`] fixed.

- Fixed orbits in **V1** menu (option Sa) are shown by either Y or N (instead of T or F).

- Known problem: **V1** gives atomic coordinates in a different order without changing their relaxation flags. This should not be considered a big problem since relaxation is not required anyway when calculating phonons.

**Version 3.51, 21.10.04**

- Bug fixes in options **V1** and **V2**;

- Option **Sx** added to **V1**, **V2** menus to 'show'/'not show' fixed atoms when displaying generalised coordinates or normal modes.

- A more geneal format for the files `F_#.mat` added (algorithm Forces): forces should be the last three columns in each block. This is the case in most DFT codes in which forces are shown. With this option in place, one can simply cut and paste the lines with forces from e.g. the ***VASP*** or ***SIESTA*** output files into the appropriate `F_#.mat` file.

- Atomic masses are entered automatically (at least for the first half of the Table) from mendeleev.inc; consequently, some changes in **V1** were required and in some other routines where atomic masses were asked.
- Small changes to the option Am of the **V1** menu.
- Bugs in the `Makefile` (missing dependencies) corrected.
- Bugs in `restore_geometry` routine fixed.

**20.10.2004, version 3.5**

- Upper boundary (`NOKD`) for number of species introduced.
- Options **Sb** and **S** of the main menu changed their position.
- **NEW** development: options **V1** and **V2** of the main menu (vibrations) have been completely rewritten. New features:
  - allows fixing atoms in orbits
  - interface for previewing of transformations and normal coordinates
  - movie `xyz` files for viewing vibrations and generalised coordinates
  - force constant matrix in terms of generalised coordinates (linear combinations of displacements) is calculated by either of the two algorithms:
    - algorithm Energy: by interpolating several total energies with a porabola (more expensive to use)
    - algorithm Force: by using atomic forces (less expensive).

**12.9.2004, version 3.5**

- Bugs related to option **Rt** (menus **M** and **Cl**) fixed by forcing the order of rotations around Cartesian axes to be specified;
- option **Hs** added to all menus to allow for disabling/enabling axes indication (via artificial H atoms) in the `xyz` files.

**20.01.2004, version 3.39** Option **mH** of **Cl** menu: negative displacements are allowed as well (was a bug?).

**20.01.2004, version 3.38**

- bugs in **Cl** related to choosing nn fixed;
- while choosing nn, a shprerical belt of a certain width is allowed, option **Nd** (**Cl** menu);
- **Cc** option of **Cl** menu: a distance from the 1st atom in the cluster is added to the printed table.

**4.11.2003, version 3.37** Bugs fixes.

**3.11.2003, version 3.36**

- for convenience, species symbols added to the ***VASP*** `CONTCAR` file on the title line
- added more support for rotation (option **R**) in the **M** menu: it is now possible to choose 3 points (e.g. atoms within the tagged list) and then the rotation is calculated so that these atoms are in either of the XY, XZ or YZ planes (sub-options **Pz**, **Py** and **Px**, respectively). Correspondingly, if this option is used, the central point for the rotation is chosen in the centre of the triangle formed by the three points. This is especially convenient if, say, a molecule above a surface is to be rotated.

**19.6.2003, version 3.35**

- A more general tagging "by atoms" and "by lists of atoms" implemented; some bugs fixed;
- for convenience, tagging is shown when invoking **Cc** (cluster) or **Co** (change) options.

**3.06.2003, version 3.34**

- Improved reading of ***SIESTA*** input files;
- a bug related to tagging of atoms which were added from another file (option **Ad** in **M** menu), fixed;
- atoms after move (option **Mv**) and rotate (option **R**) in menu **M** remain tagged if were tagged before moving.

**30.5.2003, version 3.33** Option **44** added in the **M** menu for clustering atoms around a given point

**Version 3.32, 2 May 2003** Option **Cm** added in the main menu to compare geometries of two files which may have different translations (as ***VASP*** is notorious doing wrong things with geometry!).

**9 April 2003**

- general cell -> supercell transformation;
- Support for *.mol format added.

## 2.2 Utility *lev00*

**16.10.2007, Version 3.0** This is MAJOR DEVELOPMENT: it is now `f90` code with memory allocation, so that there is no need to run ***do_param*** and ***get_param_siesta*** to recompile the code for the specific system. It is now only required a single compilation during installation. The codes ***do_param*** and ***get_param_siesta*** are now completely redundant together with the `lev00.comp` script!

**NEW** features:

- combines ***do_param*** and ***get_param_siesta*** routines with ***lev00*** into a single package
- all array boundaries are worked out automatically from the ***VASP*** or ***SIESTA*** output and/or input files, where appropriate, and all arrays which depend on the system size paramaters are allocated and deallocated
- other `f90` features are used including modules instead of common blocks
- ***CASTEP*** and ***CETEP*** capabilities have been removed as redundant
- some cleaning up of the routines and little changes of the menu options around the code, e.g. all (except that in the `bastr.f90` routine) 'stop' commands have been removed so that the run could be safely continued even if something goes wrong.

**2.04.2007, Version 2.46.3**

- bug fixes and some minor modifications to routines.

**10.02.2007, Version 2.46.1**

- Bug fixes.
- New design for the hat (when ***lev00*** starts), similar to that of ***tetr***; `tmake` script creates `build.hat` file which is attached to `hat.f` .

**7.01.2007, Version 2.46**

- **NEW**: ***get_pram_siesta*** has been modified to read in `NBANDS` and `NKPTS` from ***SIESTA*** DOS run (if the new option `BandPoints` in on), as well as band energies into `band.out` file; this paves it way for complete DOS support in ***lev00*** for ***SIESTA***.
- **NEW**: corresponding changes have been done in ***lev00*** routines to support total DOS calculation with ***SIESTA***: presently only menu options **0** and **3** of DOS (smearing tests and total DOS) are supported; support for the projected DOS should be added later.

- `character*(*)` construction introduced in all routines to handle variable string lengths.
- **NEW**: it is possible to set the upper limit to the energy ($x$ axis) when previewing the DOS graphs with ***gnuplot***.
- An old bug in working out physical bands, when the smearing is on, fixed.

### 24.7.2005, Version 2.45

- The main menu corrected and some bugs fixes.
- Option **7** of the density menu is linked to ***CASTEP*** only.
- Removed 'island' message (given if option **4** of the main menu) for ***VASP*** and ***SIESTA***.
- Option **4** of the main menu is directly linked to ***CASTEP/CETEP*** only; in the case of ***VASP***, the file name is asked (`CHGCAR`, `PARCHG` or `LOCPOT`) before reading in.
- Manipulation option rewritten from scratch; includes ***SIESTA*** as well this time.
- Option **4** -> **4** removed (writing partial density).
- Option **8** of the main menu (shift density) was moved to the density menu as option **12**, which allowed to generalise it for any code.

### 2.2.2005, Version 2.44

- Makefile `make00.lev` updated.
- **XY** option in ***gOpenMol*** menu of the density menu: the default value changed to keep atoms inside the box.
- A bug in `for_gOpenMol()` routine (wrong calculation of `center()`) corrected. Also, a very subtle bug (thanks to Chris!) in `for_gOpenMol()` routine, that resulted in a shift of the countour surfaces with respect to the geometry, corrected.

### 19.1.2005, Version 2.43

- Can read unformatted ***SIESTA*** `RHO` (electron/spin density) files as well; detects automatically if the file is formatted or unformatted. [Note: may not work if the unformatted `RHO` file was produced on another platform.]
- Debugging printout removed from `vasp_dens()` routine.

### 10.1.2005, Version 2.42

- Another format for reading k-points from the `OUTCAR` file implemented as the one existed before did not seem to appear in the `OUTCAR` file in some cases.

### 24.11.2004, Version 2.41

- `make00.lev` changed.

- `PROCAR` file with projected DOS is now working with option `LORBIT=11` of ***VASP***. Still, only total $p$ and $d$ contributions are worked out by ***lev00***, although all components of those are available in `PROCAR`. Can easily be taken into account as well, but later.

### 8.10.2004, Version 2.4

- The multiplication factor for the density used in line/plane options (`multcon`) has been changed from 100 to 1, since the density is given in actual units ($\AA^{-3}$).

- An option **Ap** is added to the plane menu to show (X,Y) positions of all atoms with respect to the local coordinate system fixed with the plane including the out-of-plane distance.

- The title line in the `CONTCAR` file is also printed out on the screen for convenience to have the order of species.

### 30.3.2004 - 15.04.2004, Version 2.3

- A bug related to reading density of ***SIESTA*** (thanks to Ross) fixed.

- Menus throughout the code modified to allow character inputs instead of only numbers.

- Option **An** to choose the way coordinates are given, added. **Co** is added in all menus options related to density to view atomic positions in fractional/Cartesian format. **Q** is used to quit. This all is in accord with the general implementation of menus in ***tetr***. Still, other menus will have to be modified accordingly - later!

- Atomic species are now asked every time ***lev00*** is started and the coordinates are read in.

- **NEW** development: option **10** added in the density menu allowing to write input files for ***gOpenMol*** (see `http://www.csc.fi/gopenmol`) to plot ”orbitals” out of the density.

### 1.02.2004, Version 2.2

- A bug in the main routine (a call of `prep_plot()` in option MAP was missing) fixed.
- **NEW**: added option **44** in the plane menu: after the plane is chosen (i.e. the normal, central point and the dimensions), one can choose another plane parallel to the existing one. Note that if the plane was chosen using 3 points method, you will have to choose the same plane again by specifying the normal only (it is diplayed in the menu), so that the 3-point method becomes off.

### 22.01.2004, Version 2.1

- ***do_param*** writes `ISPIN` variable to the `param.inc` file.
- spin 1,2 for spin-polarised calculations is recognised by ***lev00*** when doing various DOS automatically. In the case of a spin-polarised calculation any DOS is without the spin prefactor of 2.
- structure of the main routine has been simplified considerably and is now more transparent; some ***CASTEP/CETEP*** options are not visible if ***VASP*** or ***SIESTA*** are on, and vice versa.
- when searching for a string consisting of 2 or 3 words, the string is broken down into words; this means that number of spaces between the words does not matter anymore. This is useful if e.g. the number of spaces was changed in the output file of the PW code which ***lev00*** needs (as e.g. happened between versions 4.4. and 4.6 of ***VASP***).

### 19.09.2003, Version 2.0

- Some bugs corrected.
- Added support for ***SIESTA*** (with some help from Adam Foster): ***lev00*** can now fully support charge and spin density options; due to a problem in setting users k-points in ***SIESTA***, there is still no support for DOS and PDOS. The necessary routine to create `param.inc` is called ***get_param_siesta***, it is in `TOOLS/SIESTA_Tools` directory.

# Chapter 3

# Utility *tetr*

## 3.1 What it can do

***tetr*** is an interactive menu-driven program which enables you:

- can read in existing input files from ***VASP***, ***SIESTA***, old ***CASTEP***, `xyz` format
- can save the current geometry in the same formats
- creates a geometry file from scratch, including from the space group symbol (both in 2D and 3D), or modifies an existing geometry in various ways
- builds a cluster from the bulk cell atomic positions; can terminate some or all of the cluster boundary atoms with H atoms;
- controls calculation of molecular or lattice vibrations taking into account irreducible representations of the system (either the point group or the group of internal translations within the supercell); in the case of a molecule, translations and rotations of the molecule as a whole are automatically eliminated; can do phonon DOS, animations of normal modes, recalculate the phonons for different isotops (keeping the same force constant matrix);
- creates **k**-points for the ground state (Manchorst-Pack), DOS (the tetrahedral method with complete account of the point symmetry of the cell) or band-structure calculations (along a chosen line in the Brillouin zone);
- calculates the Coulomb (Ewald) potential at an arbitrary point in the cell in the point-ion approximation,
- some additional useful options designed to analyze the system geometry: compare with other geometry files, e.g. before and after relaxation; a calculator of interatomic distances; check a point in the cell for equivalence with existing atoms; save atomic positions within a box across adjacent unit cells

- Potential energy surface (PES): move atoms in the cell keeping the point symmetry: this is useful for calculating the PES for the ground state symmetry.
- Creates a shell-script for moving one part of the system with respect to the other on a 3D grid (e.g. for AFM/STM calculations).

## 3.2 Installation and general information

The distribution file archive (e.g. `tetr_3.93a.tgz`) should be untarred and unzipped, e.g.

```
tar   -xzf   tetr_3.93a.tgz
```

This command will create a directory (e.g. `tetr_3.93a`) with the whole code distribution.

The code ***tetr*** must be compiled only once by performing the command

```
make
```

in your `~/TOOLS/TETR` directory.

An environmental variable `$HOME_TETR` must be set to the correct location of the ***tetr*** directory, e.g.

```
setenv HOME_TETR ~/TOOLS/TETR
```

for the csh or tcsh shells, or

```
export HOME_TETR=~/TOOLS/TETR
```

in the case of the bash shell. To solve this problem once and forever, put the right statement in the `.cshrc` or `.bashrc` files, respectively. The statement will be executed automatically (and the variable set) every time an `xterm` is started or at the login. To force the setting from an existing terminal window, run the `source` command (e.g. `source ~/.cshrc`).

## 3.3 Menus in *tetr*

Most of the menus have common options:

- **Co** - (info) show a table with Cartesian and fractional coordinates of all atoms in the unit cell
- **S** - save the current geometry into an input file (the code format to be chosen next)

- **Sy** - (info) show (usually, the point) symmetry of the current geometry; in most cases the lattice translations are also included, i.e. two points are thought to be equivalent if they are separated by a lattice translation
- **U** - undo the last step
- **An** - (setting) choose the format in which coordinates are to be specified for input in the menus: Angstrom, fractional or atom number (in order of atoms in the system)

A number of options of the main menu allows a user to change the current geometry in various ways. Some options have similar meanings in different menus, although their actual work depends on where they are called from, for example:

- **Mv** - move one or more atoms in the cell
- **Rt** - rotate one or more atoms

In many menus when a change in geometry is done, the code allows the user to dump the geometry manually or automatically into a file `geom.xyz` that is written in the `xyz` format. If you run any `xyz`-file previewer (like ***Xmol***), then you can preview the system during the modification process, and this is extremely useful! A number of options (mainly settings) control this:

- **XY** - (setting) the format of the `geom.xyz` file (either for ***Xmol***, ***Ymol*** or disable); in the latter case the file will not be written after each change of the geometry
- **Hs** - (setting) either add or not H atoms to the `geom.xyz` file to preview x,y,z axes (necessary for ***Xmol*** as it does not have the axes facility)
- **Bb** - (setting) breeding box is specified: this option allows you to preview not just the cell, but $N_1 \times N_2 \times N_3$ extention of it; note, that the actual cell is not changed, this option only affects the atoms written to the `geom.xyz` file
- **W** - force writing `geom.xyz` file for the current geometry

Normally, when choosing a set of atoms, two integers are specified, so that all atoms with numbers between (and including) them are in the list. For instance, choosing 5 and 10 would include atoms 5,6,7,8,9 and 10 into the list. In some cases a non-contagious list is required, e.g. 5, 7-10, 15 and 22. In this case you can "tag" atoms (option **T**) by naming all of them and using a dash if all numbers in between are to be included, e.g. 5 7-10 15 22 for the example above. Use either spaces or commas (or both) to separate the individual lists, e.g. 5, 7-10 15, 22 and 5 7-10,15,22 have the same effect as the sequence above. Note that once atoms are tagged, pressing **T** again untages them.

## 3.4 Getting started

When starting the code (e.g. type `./tetr`), it first checks if the variable $HOME_TETR is set correctly:

```
.....................................................
......  TETR is an interface to VASP/SIESTA/etc .......
.....................................................
>>>>>>>> 1st release:  25.07.1994
>>>>>>>> Version :  3.92h
>>>>>>>> Build :  Tue Jun 21 19:39:20 BST 2005
>>>>>>>> today :  Wed Jul 13 00:52:31 BST 2005
.....................................................
......  inquiries to L. Kantorovich .........
.....................................................
OK! File /home/lev/TOOLS/TETR/tetr exists!
....  TETR home directory:  /home/lev/TOOLS/TETR
OK! File /home/lev/TOOLS/TETR/symgrp.rep exists!
OK! File /home/lev/TOOLS/TETR/collect_forces exists!
OK! File /home/lev/TOOLS/TETR/get_forces exists!
.................................................
```

It must find the executables ***tetr*** and ***get_forces***, a script `collect_forces` and the file `symgrp.rep`. At the same time, the current date and version are shown.

Next, the following first menu is displayed that prompts a user to choose the input format for the geometry, the so-called *Input-menu*:

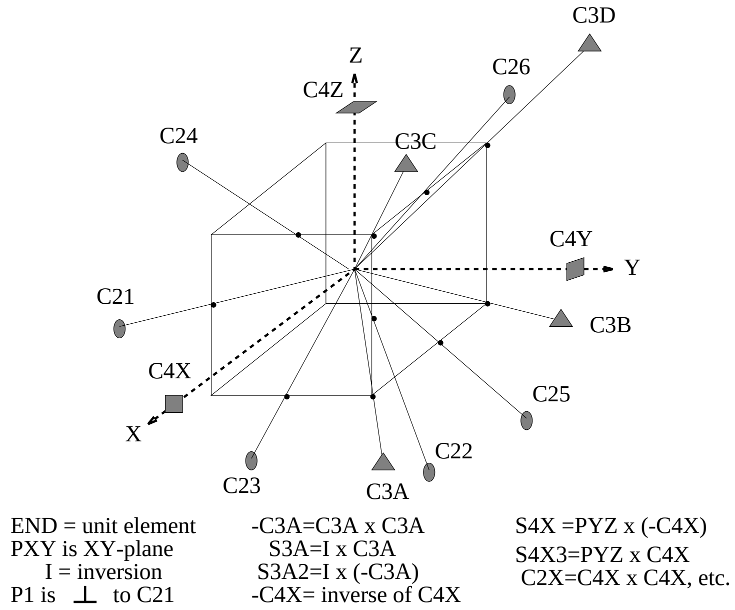
```
Choose the input format:
1.  open input as CASTEP [ *.geom]
2.  open input as CETEP / old CASTEP
3.  open input as VASP
4.  open input as SIESTA [ *.fdf ]
5.  open output as SIESTA [ *.XV ]
6.  open as XYZ-type file [ *.xyz ]
7.  NEW or go directly to the main menu
8.  QUIT
------------>
```

Options **1-6** imply existing files of the specified formats; you will then be prompted to choose the file name, and the input file will be read in. If successful, you will be prompted to the main menu, described in Section 3.6.1. If not (e.g. due to an error in the input file or becuase the latter does not exist), you will be prompted again to the menu above.

The option **7** corresponds to starting from scratch rather then reading an existing input file of one of the DFT codes. In this case a new input file is to be constructed and then saved in one of known formats.

## 3.5 Symmetry

The point group symmetry of the system is determined automatically and can be checked (option **Sy** in various menus). Normally, translations are also taken into account when checking equivalence of atoms, i.e. two point in space are thought to be equivalent if they are separated by a lattice translation $\mathbf{L} = n_1\mathbf{a}_1 + n_2\mathbf{a}_2 + n_3\mathbf{a}_3$ with $n_i$ being integers. There are some exceptions though corresponding to *molecular*



END = unit element
PXY is XY-plane
I = inversion
P1 is ⊥ to C21

-C3A=C3A x C3A
S3A=I x C3A
S3A2=I x (-C3A)
-C4X= inverse of C4X

S4X =PYZ x (-C4X)
S4X3=PYZ x C4X
C2X=C4X x C4X, etc.

Figure 3.1: Symmetry operations of the $O_h$ group.

properties, specifically options **Cl** (Section 3.6.4) and **Vm** (Section 3.6.8.5) of the main menu, when translations are ignored. The symmetry is used in many parts of the code.

At the moment, all symmetry operations are fixed to the axes of the Cartesian coordinate system. The complete list of the symmetry operations is shown in Figs. 3.1 and 3.2.

### 3.5.1 A note about symmetry

The fact that the point group operations are rigidly related to the Cartesian axes, the symmetry determined by ***tetr*** may not be as rich as it should be. On the other hand, in many cases it is useful to have the maximum symmetry possible (e.g. when generating **k**-points for the electronic DOS calculation). Therefore, it is advised to maximise the system symmetry by shifting and/or rotating the whole cell. This can be done in the M-menu, Section 3.6.5, where each time you change your system, the symmetry group is recalculated so that you can check the symmetry at every stage (option **Sy**).

In order to do a band-structure calculation, you should first relax your system and obtain the electronic density. Note that the ground state density can be left unchanged if you just *rotate* your coordinate system (we assume that it is written with respect to the fractional coordinates of the grid points in your cell; this supposed to be the case in both ***VASP*** and ***SIESTA***). However, if you have to *shift* your system by some vector, then the density matrix should be recalculated and a new file must be produced. This option is provided by the utility ***lev00*** (see Section 4.6.3.13). Therefore, it is always wise to think about it *before* running the ground state calculation: choose the most symmetrical orientation of the system with respect to the Cartesian axes, use ***tetr*** to

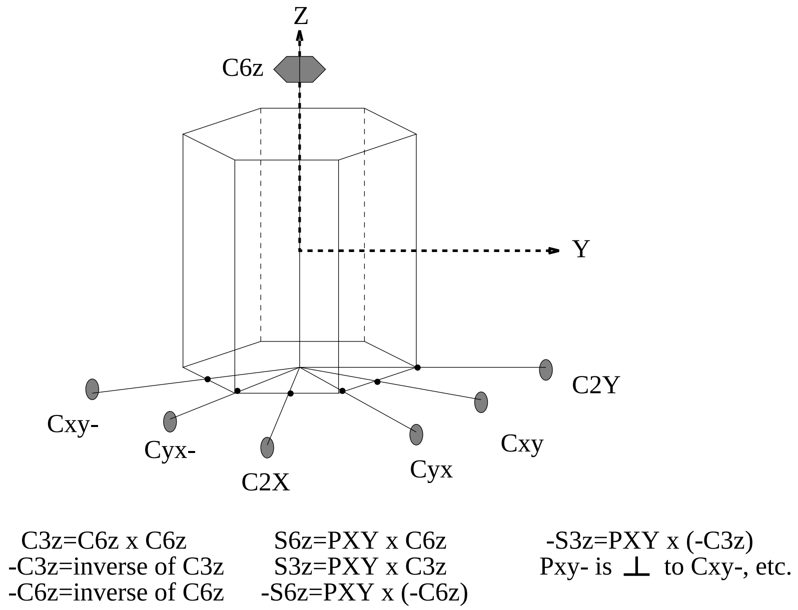


Figure 3.2: Additional symmetry operations due to the $D_{6h}$ group.

show the point group every time!

## 3.6 Detailed description of options in *tetr*

### 3.6.1 Main menu

The main menu looks like this:

```
/---------------------------------------------------\\
|....................  MAIN MENU:.....................|
\\---------------------------------------------------/
>>>>>>>>>>>>> Create unit cell from scratch <<<<<<<<
[current geometry will be lost]
B. Generate supercell geometry from scratch
Pc.  Generate primitive cell from space group symbol
>>>>>>>>>>>>>>>>> Cluster construction <<<<<<<<<<<<
Cl.  Build up a cluster from the lattice
>>>>>>>>>>>>> Geometry file already exists <<<<<<<<<<
M. Complex cell modifications
KP. k-points generation
>>>>>>>>>>>>>>>>> Coulomb potential <<<<<<<<<<<<<<<
Ew.  Coulomb potential inside the unit cell
>>>>>>>>>>>>>>> Calculation of phonons <<<<<<<<<<<<<
Vm.  Vibrations of molecules => preparation
Vc.  Vibrations of crystals (point group) => preparation
Vs.  Vibrations of crystals (space group) => preparation
V2.  Vibrations => calculation [normal modes]
V3.  Vibrations => application [DOS, masses]
>>>>>>>>>>>>>>> Moving part of the system <<<<<<<<<<<<
AF. Generation of the script for AFM modelling
>>>>>>>>>>>> Representation/visualisation <<<<<<<<<<<
Cm.  Compare geometry with the one in another file
D. Distances between atoms
Ch.  Check specified point in the cell
```

```
    Sb.  Save in a box running across adjacent cells
    >>>>>>>>>>>>>> Potential energy surface <<<<<<<<<<<<<
    PE. PES: move atoms keeping the symmetry
    ---------- g e n e r a l s e t t i n g s -----------
    XY. Format of <geom.xyz> is set to Xmol
    R. Read geometry from another file
    S. Save
    Ss.  Save SIESTA settings with geometry from another XV file
    Q. Quit.
    Specify the character and press ENTER ------->
```

The sections below explain all these options in some detail.

### 3.6.2 B - Generate supercell geometry from scratch

One can generate lattice vectors of the cell and atomic positions inside the cell from scratch. The initial *B-option* menu is this:

```
    ========= CHOOSE AN OPTION:
    1.  Number of species:  <===== undefined
    3.  Basic translations of the PRIMITIVE CELL <===== undefined
    -------------- Extention matrix {cell -> supercell} -----------
    5.  Specify directly
              | 1 0 0 |
    IJ(i,j)= | 0 1 0 |, ext= 1
              | 0 0 1 |
    -------------- g e n e r a l s e t t i n g s ----------------
    Or.  Move atoms closer to origin:  NO
    XY. Produce <geom.xyz> file after every change for Xmol
    Hs.  H atoms to be added to <geom.xyz> file:  NO
    An.  [For input] Coordinates are specified in:  <Angstroms>
    Bb.  Set the size of the breeding box for visualisation
    >>>>> Current setting for the breeding box:  <<<<<
    [ 0...  0] x [ 0...  0] x [ 0...  0]
    >>>>> extension = 1, # of atoms= 0
    Q. Quit:  return to the previous setting (if exists)


---------> Choose an appropriate option:
```

You must first specify all the data in "undefined" fields. Start from the option 1 and specify the primitive unit cell information. New options will appear that require additional information. The primitive lattice vectors $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ are specified either manually or from the list, both to be found in the following menu:

```
    --------------------------------------------------
    --------- Choose one Bravais lattices:  -----------
    --------------------------------------------------
    0.Manually
    1.Triclinic system
    2.Monoclinic system.  Simple lattice
    3.Monoclinic system.  Side-centered lattice
    4.Orthorhombic system.  Simple lattice
    5.Orthorhombic system.  Fcc-1 lattice
    6.Orthorhombic system.  Fcc-2 lattice
    7.Orthorhombic system.  Bcc lattice
    8.Tetragonal system.  Simple lattice
    9.Tetragonal system.  Bcc-1 lattice
    10.Tetragonal system.  Bcc-2 lattice (diff.setting)
    11.Cubic system.  Simple lattice
    12.Cubic system.  Bcc lattice
    13.Cubic system.  Fcc lattice
    14.Rhombohedral system.
    15.Hexagonal system


------> Choose the appropriate number ------->
```

The cell you want to construct may consist of several primitive cells with lattice vectors

$$\mathbf{A}_i = \sum_{j=1}^{3} T_{ij}\mathbf{a}_j \tag{3.1}$$

where $\mathbf{T}$ is the $3 \times 3$ transformation (extention) matrix consisting only of integer numbers. By default, the extention matrix is the unit matrix, i.e. $\mathbf{A}_i = \mathbf{a}_i$. However, using options **4** or **5** from the B-option menu, the matrix $\mathbf{T}$ can be specified either directly (option **4**) or by specifying the vectors $\mathbf{A}_i$(option **5**). In the latter case the matrix $\mathbf{T}$ is calculated and it is checked if it consists of integers only.

Finally, the unit cell is generated in option **7**. If you leave the B-option menu by pressing **Q** (and Enter), the main menu of ***tetr*** will appear, so that you can modify your setting, add **k**-points and then save them in the format of your choice.

### 3.6.3 Pc - Generate primitive cell from space group symbol

This option allows construction of a geometry file using the information of the International Tables for Crystollography, i.e. from the space group symbol. This can be done either in 2D or 3D. This option is useful if correct space group symmetry is required for the system to be constructed since ***tetr*** will automatically create all atoms in the cell from the minimum possible number of them; in other words, complete stars of atoms are created according to the specified space group.

The initial *Pc-menu* looks like this:

```
========= CHOOSE AN OPTION:
Sp.  Select the space group from the list
At.  Manually add an atom to the cell
Ad.  Add atoms from another file
--------- operations with multiple lists of atoms ------------
-------------- g e n e r a l s e t t i n g s ----------------
UU. ****** Start AGAIN from scratch ******
U. ********** UNDO the last step *************
Dm.  3D space groups (for bulk)
XY. Produce <geom.xyz> file after every change for Xmol
Hs.  H atoms to be added to <geom.xyz> file:  NO
Q. Quit:  return to the previous setting (if exists)
----------> Choose an appropriate option:
```

To start, choose first the dimension using option **Dm**. Then, choose the space group using option **Sp**. In the 2D case the corresponding menu is:

```
[ 1] p1 [ 2] p2 [ 3] pm [ 4] pg [ 5] cm [ 6] p2mm [ 7] p2mg [ 8] p2gg
[ 9] c2mm [ 10] p4 [ 11] p4mm [ 12] p4gm [ 13] p3 [ 14] p3m1 [ 15] p31m [ 16] p6
[ 17] p6mm
Choose the space group number from the Tables:
```

while in the 3D case the menu looks like this:

```
[ 1] C1_1 [ 2] Ci_1 [ 3] C2_1 [ 4] C2_2 [ 5] C2_3 [ 6] Cs_1 [ 7] Cs_2 [ 8] Cs_3
[ 9] Cs_4 [ 10] C2h_1 [ 11] C2h_2 [ 12] C2h_3 [ 13] C2h_4 [ 14] C2h_5 [ 15] C2h_6 [ 16] D2_1
[ 17] D2_2 [ 18] D2_3 [ 19] D2_4 [ 20] D2_5 [ 21] D2_6 [ 22] D2_7 [ 23] D2_8 [ 24] D2_9
[ 25] C2v_1 [ 26] C2v_2 [ 27] C2v_3 [ 28] C2v_4 [ 29] C2v_5 [ 30] C2v_6 [ 31] C2v_7 [ 32] C2v_8
[ 33] C2v_9 [ 34] C2v_10[ 35] C2v_11[ 36] C2v_12[ 37] C2v_13[ 38] C2v_14[ 39] C2v_15[ 40] C2v_16
[ 41] C2v_17[ 42] C2v_18[ 43] C2v_19[ 44] C2v_20[ 45] C2v_21[ 46] C2v_22[ 47] D2h_1 [ 48] D2h_2
[ 49] D2h_3 [ 50] D2h_4 [ 51] D2h_5 [ 52] D2h_6 [ 53] D2h_7 [ 54] D2h_8 [ 55] D2h_9 [ 56] D2h_10
[ 57] D2h_11[ 58] D2h_12[ 59] D2h_13[ 60] D2h_14[ 61] D2h_15[ 62] D2h_16[ 63] D2h_17[ 64] D2h_18
[ 65] D2h_19[ 66] D2h_20[ 67] D2h_21[ 68] D2h_22[ 69] D2h_23[ 70] D2h_24[ 71] D2h_25[ 72] D2h_26
[ 73] D2h_27[ 74] D2h_28[ 75] C4_1 [ 76] C4_2 [ 77] C4_3 [ 78] C4_4 [ 79] C4_5 [ 80] C4_6
[ 81] S4_1 [ 82] S4_2 [ 83] C4h_1 [ 84] C4h_2 [ 85] C4h_3 [ 86] C4h_4 [ 87] C4h_5 [ 88] C4h_6
[ 89] D4_1 [ 90] D4_2 [ 91] D4_3 [ 92] D4_4 [ 93] D4_5 [ 94] D4_6 [ 95] D4_7 [ 96] D4_8
[ 97] D4_9 [ 98] D4_10 [ 99] C4v_1 [100] C4v_2 [101] C4v_3 [102] C4v_4 [103] C4v_5 [104] C4v_6
[105] C4v_7 [106] C4v_8 [107] C4v_9 [108] C4v_10[109] C4v_11[110] C4v_12[111] D2d_1 [112] D2d_2
[113] D2d_3 [114] D2d_4 [115] D2d_5 [116] D2d_6 [117] D2d_7 [118] D2d_8 [119] D2d_9 [120] D2d_10
[121] D2d_11[122] D2d_12[123] D4h_1 [124] D4h_2 [125] D4h_3 [126] D4h_4 [127] D4h_5 [128] D4h_6
[129] D4h_7 [130] D4h_8 [131] D4h_9 [132] D4h_10[133] D4h_11[134] D4h_12[135] D4h_13[136] D4h_14
[137] D4h_15[138] D4h_16[139] D4h_17[140] D4h_18[141] D4h_19[142] D4h_20[143] C3_1 [144] C3_2
[145] C3_3 [146] C3_4 [147] C3i_1 [148] C3i_2 [149] D3_1 [150] D3_2 [151] D3_3 [152] D3_4
[153] D3_5 [154] D3_6 [155] D3_7 [156] C3v_1 [157] C3v_2 [158] C3v_3 [159] C3v_4 [160] C3v_5
[161] C3v_6 [162] D3d_1 [163] D3d_2 [164] D3d_3 [165] D3d_4 [166] D3d_5 [167] D3d_6 [168] C6_1
[169] C6_2 [170] C6_3 [171] C6_4 [172] C6_5 [173] C6_6 [174] C3h_1 [175] C6h_1 [176] C6h_2
[177] D6_1 [178] D6_2 [179] D6_3 [180] D6_4 [181] D6_5 [182] D6_6 [183] C6v_1 [184] C6v_2
[185] C6v_3 [186] C6v_4 [187] D3h_1 [188] D3h_2 [189] D3h_3 [190] D3h_4 [191] D6h_1 [192] D6h_2
[193] D6h_3 [194] D6h_4 [195] T_1 [196] T_2 [197] T_3 [198] T_4 [199] T_5 [200] Th_1
[201] Th_2 [202] Th_3 [203] Th_4 [204] Th_5 [205] Th_6 [206] Th_7 [207] O_1 [208] O_2
[209] O_3 [210] O_4 [211] O_5 [212] O_6 [213] O_7 [214] O_8 [215] Td_1 [216] Td_2
[217] Td_3 [218] Td_4 [219] Td_5 [220] Td_6 [221] Oh_1 [222] Oh_2 [223] Oh_3 [224] Oh_4
[225] Oh_5 [226] Oh_6 [227] Oh_7 [228] Oh_8 [229] Oh_9 [230] Oh_10
Choose the space group number from the Tables:
```

The number to be used to choose the group in either case is the correct group number in the Tables, while the symbol corresponds to the Schöonflis notations. Once the space group is known, a new option **Lt** appears that allows choosing the spacial parameters of the Bravais lattice, e.g. lengths $a$, $b$, $c$ of the conventional unit cell and the angles $\alpha$, $\beta$ and $\gamma$ between its sides. Note that only the minimum number of parameters sufficient to specify the lattice will be asked, e.g.

```
......O.K.! D4h_11 space group is found!!!
.... Crystal point-group is D4h
.... Singonia is: Tetragonal
.... Bravais lattice is: simple lattice
  Specify cell via {a,c} (A)
```

After that, the space group is known completely so that it is possible to have a look at it (option **Sy**) in detail, e.g. the following information is shown for the group $D_{4h}^{11}$:

```
>>>>>>>>>>>>>> Space group <<<<<<<<<<<<<<<<<<<<
_i__KG__NAZ_____{Fractional translation}______Int.Tables___
1 1 UNIT 0.00000 0.00000 0.00000 (x,y,z)
2 7 C2X 0.00000 0.00000 0.50000 (x,-y,-z+1/2)
3 9 C2Y 0.00000 0.00000 0.50000 (-x,y,-z+1/2)
4 5 C2Z 0.00000 0.00000 0.00000 (-x,-y,z)
5 2 C4Z 0.50000 0.50000 0.50000 (-y+1/2,x+1/2,z+1/2)
6 6 -C4Z 0.50000 0.50000 -0.50000 (y+1/2,-x+1/2,z+1/2)
7 19 C22 0.50000 0.50000 0.00000 (y+1/2,x+1/2,-z)
8 20 C21 0.50000 -0.50000 0.00000 (-y+1/2,-x+1/2,-z)
9 48 I 0.50000 0.50000 0.50000 (-x+1/2,-y+1/2,-z+1/2)
10 26 PYZ 0.50000 -0.50000 0.00000 (-x+1/2,y+1/2,z)
11 27 PXZ 0.50000 0.50000 0.00000 (x+1/2,-y+1/2,z)
12 25 PXY 0.50000 0.50000 0.50000 (x+1/2,y+1/2,-z+1/2)
13 28 S4Z 0.00000 0.00000 0.00000 (y,-x,-z)
14 31 S4Z3 0.00000 0.00000 0.00000 (-y,x,-z)
15 43 P2 0.00000 0.00000 0.50000 (-y,-x,z+1/2)
16 42 P1 0.00000 0.00000 -0.50000 (y,x,z+1/2)
```

```
Hit ENTER when done ...
>>>>>>>> Reciprocal lattice vectors (up tp 2*pi) <<<<<<<<<<
vec 1 => 1.00000 0.00000 0.00000
vec 2 => 0.00000 1.00000 0.00000
vec 3 => 0.00000 0.00000 0.33333
Hit ENTER when done ...
```

The first column of the "Space group" table shows the space group operation number, the 2nd - the internal operation number (not important for the user), the 3rd - rotation/mirror operation name (see Figs. 3.1 and 3.2 and Section 3.5), then follows three columns with the fractional translation (can be shown either in fractional or Cartesian coordinates via option **Ca**; in the former case the fractional coordinates can be shown either with respect to the primitive or conventional cell via option **Re**). Finally, the last column shows the Tables notation for the group operation using a general position point ($xyz$).

Once the lattice has been specified, atoms in the unit cell can also be added using **At** and/or **Ad**. The former one allows atoms to be added one by one, the latter - from another file.

Then, by applying **Gn** you can generate the complete unit cell. In this case all group operations (shown in **Sy**) are applied to each of the atoms known at this stage, and new atoms are generated if found. If some of the atoms were there already, they will be removed, so that each position will appear only once. Thus, this way complete *stars* of atoms in the cell can be obtained. Note that you can use the option **Gn** at any stage. If new atoms are added (via **At** and/or **Ad**), then complete stars for these atoms can only be created by applying **Gn** again; otherwise, some starts may be incomplete. Thus, always apply **Gn** at the end of generating the cell. After the complete stars have been obtained, the **Sy** option will show which atoms of the cell are obtained from every atom by applying symmetry operations, e.g.

```
>>>>>>>>>>>> Space group <<<<<<<<<<<<<<<<<
_i__KG__NAZ_____{Fractional translation}______Int.Tables___
1 1 UNIT 0.00000 0.00000 0.00000 (x,y,z)
2 7 C2X 0.00000 0.00000 1.50000 (x,-y,-z+1/2)
3 9 C2Y 0.00000 0.00000 1.50000 (-x,y,-z+1/2)
4 5 C2Z 0.00000 0.00000 0.00000 (-x,-y,z)
5 2 C4Z 0.50000 0.50000 1.50000 (-y+1/2,x+1/2,z+1/2)
6 6 -C4Z 0.50000 0.50000 -1.50000 (y+1/2,-x+1/2,z+1/2)
7 19 C22 0.50000 0.50000 0.00000 (y+1/2,x+1/2,-z)
8 20 C21 0.50000 -0.50000 0.00000 (-y+1/2,-x+1/2,-z)
9 48 I 0.50000 0.50000 1.50000 (-x+1/2,-y+1/2,-z+1/2)
10 26 PYZ 0.50000 -0.50000 0.00000 (-x+1/2,y+1/2,z)
11 27 PXZ 0.50000 0.50000 0.00000 (x+1/2,-y+1/2,z)
12 25 PXY 0.50000 0.50000 1.50000 (x+1/2,y+1/2,-z+1/2)
13 28 S4Z 0.00000 0.00000 0.00000 (y,-x,-z)
14 31 S4Z3 0.00000 0.00000 0.00000 (-y,x,-z)
15 43 P2 0.00000 0.00000 1.50000 (-y,-x,z+1/2)
16 42 P1 0.00000 0.00000 -1.50000 (y,x,z+1/2)
Hit ENTER when done ...
>>>>>>>> Reciprocal lattice vectors (up tp 2*pi) <<<<<<<<<<
vec 1 => 1.00000 0.00000 0.00000
vec 2 => 0.00000 1.00000 0.00000
vec 3 => 0.00000 0.00000 0.33333
Hit ENTER when done ...
>>>>>>>>>>>> Group operations on cell atoms <<<<<<<<<<<<<<<<<
—- Si atom 1 is transformed into atoms:
1 2 2 1 3 3 4 4 3 4 4 3 1 1 2 2
```

```
—- Si atom 2 is transformed into atoms:
2 1 1 2 4 4 3 3 4 3 3 4 2 2 1 1
—- Si atom 3 is transformed into atoms:
3 4 4 3 1 1 2 2 1 2 2 1 3 3 4 4
—- Si atom 4 is transformed into atoms:
4 3 3 4 2 2 1 1 2 1 1 2 4 4 3 3
Hit ENTER when done ...
```

Option **U1** appears after **Gn** has been applied, it keeps only first atoms of each star. This is useful if Wyckoff positions to be studied or during some intermediate stages of cell generation.

The complete menu at this stage may look like this:

```
========= CHOOSE AN OPTION:
Current number of atoms in the cell: 1
Current number of species: 1
   Si[ 1]
Sp. Space group has been selected as D4h_11
Space group system : Tt
Bravais lattice type: SL0
Lt. Basic translations of the PRIMITIVE CELL:
   vec 1 => 1.00000 0.00000 0.00000
   vec 2 => 0.00000 1.00000 0.00000
   vec 3 => 0.00000 0.00000 3.00000
At. Manually add an atom to the cell
Ad. Add atoms from another file
Gn. Generate the PRIMITIVE CELL: <===== Done!
——— operations with multiple lists of atoms ————
T. Tag atoms (specify multiple lists): currently OFF
————— g e n e r a l   s e t t i n g s —————-
UU. ****** Start AGAIN from scratch ******
U. ********** UNDO the last step *************
U1. Keep only the 1st atoms from each star
Dm. 3D space groups (for bulk)
XY. Produce <geom.xyz> file after every change for Xmol
Hs. H atoms to be added to <geom.xyz> file: NO
An. [For input] Coordinates are specified in: <Angstroms>
Bb. Set the size of the breeding box for visualisation
     > > > > > Current setting for the breeding box: < < < < <
     [ 0... 0] x [ 0... 0] x [ 0... 0]
     > > > > > extension = 1, # of atoms= 1
Sy. Show the symmetry
Ca. Fractional translations in Sy are in Cartesian
Re. Fractional translations in Sy reduced wrt primitive cell
Co. Show current atomic positions in fractional/Cartesian
W. Write <geom.xyz> file for preview using current breeding
P. Proceed: keep the curent setting
S. Save
Sf. Save the SIESTA code to symmetrise forces & stress tensor
Q. Quit: return to the previous setting (if exists)
———-> Choose an appropriate option:
```

You can now save the geometry as an input file (option **S**). In addition, from version 3.94h14 it is possible (option **Sf**) to save the ***f77*** code to be put into the ***SIESTA*** `constr.f` routine, to maintain the space group symmetry during the geometry relaxation, including both atomic positions and the stress tensor. The code is contained in `for_SIESTA_constr.file` file. Take the file and insert it into the `constr.f` file in the ***SIESTA*** `Src` directory and type ***make*** to recompile ***SIESTA***.

Additional options appear if atoms are tagged (option **T**):

```
——— operations with multiple lists of atoms ————
T. Untag atoms (multiple lists): currently ON
```

```
    >> Current set of tagged atoms: 1
Rm. Remove tagged atoms with their stars
Rn. Rename tagged atoms: change species
Mv. Move tagged atoms to another general position
Rt. Rotate tagged atoms about the X,Y,Z axes
```

You can (similarly to the corresponding options of the **M** menu of Section 3.6.5):

- remove tagged atoms from the cell (option **Rm**),
- rename them, i.e. species change (option **Rn**),
- move/diplace tagged atoms (option **Mv**) or
- rotate them (option **Rt**).

Note that complete stars of atoms will be tagged (if exist, i.e. if **Gn** has been applied before), moved, rotated, removed or renamed.

Finally, you can save the obtained geometry using **S** or proceed to the main menu (option **P**) to apply other available operations on the constructed cell (e.g. in the **M** menu of Section 3.6.5). Option **Q** will take you out of the **Pc** menu, the geometry generated will be forgotten.

### 3.6.4 Cl - Build up a cluster from the lattice

This option becomes available in the main menu if the geometry is known, i.e. it has either been read in or constructed from scratch. The idea is this: starting from one atom, build up all or some its nearest neighbours, then nearest neighbours to latter atoms and so on. Finally, if desired, some or all boundary atoms of the cluster are terminated by hydrogen atoms positioned at a certain distance from them and along the directions of the actual bonds of the bulk system.

The *Cl-menu* initially looks like this:

```
>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
...  CURRENT lattice vectors for the supercell ARE...
AJ(1):  10.88160 0.00000 0.00000
AJ(2):  0.00000 10.88160 0.00000
AJ(3):  0.00000 0.00000 10.88160
>>>>>>>>>>>>>>>>>>>>>> Current species are:  <<<<<<<<<<<<<<<<<<<<
Si O
========= CHOOSE AN OPTION:
/===================================================\\
|| HELP: when using options N and H, you may not ||
|| be bothered about internal cluster atoms as ||
|| those will be ignored anyway ||
\\===================================================/
R. Read existing cluster geometry
1.  START HERE: choose the first cluster atom
-------------- g e n e r a l s e t t i n g s ----------------
XY. Produce <geom.xyz> file after every change for Xmol
Hs.  H atoms to be added to <geom.xyz> file:  NO
Co.  Show lattice atomic positions
Hd.  Distance between an atom and H is:  1.00000000000000
Nd.  Spherical belt width when finding nn is:  0.100000001490116
W. Write <geom.xyz> file for the current cluster
```

```
Q. Proceed/Quit
----------> Choose an appropriate option:
```

At the top, lattice vectors of the unit cell are shown together with existing species. Start from option **1** that will prompt you to choose the very first atom of the cluster. After this is done, the menu grows into:

```
R. Read existing cluster geometry
1.  Cluster has been started from atom = 1
   Species = Si
   Position = ( 0.05194, 0.05194, 0.03418)
N. Build all first neighbours around atom(s)
H. Terminate a range of atoms with hydrogens
mH. Move a range of H atoms along bonds
5.  Rename a range of atoms:  change species
--------- operations with multiple lists of atoms ------------
T. Tag atoms (specify multiple lists):  currently OFF
--------------------- shift ----------------------------------
O. General shift of the cluster (new origin)
Cm.  Shift the cluster to the centre of mass
------------------------ final options ----------------------
Rt.  Rotate the WHOLE cluster about the X,Y,Z axes
Mv.  Move atom(s)
------------- g e n e r a l s e t t i n g s ----------------
XY. Produce <geom.xyz> file after every change for Xmol
Hs.  H atoms to be added to <geom.xyz> file:  NO
Co.  Show lattice atomic positions
Hd.  Distance between an atom and H is:  1.00000000000000
Nd.  Spherical belt width when finding nn is:  0.100000001490116
W. Write <geom.xyz> file for the current cluster
Cc.  Show current atomic positions in Cartesian
S. Save geometry
Q. Proceed/Quit
----------> Choose an appropriate option:
```

Now it is possible to choose the first shell of nearest neighbours of the first atom by pressing **N**. All nearest neighbours of the first atom will be added. This is decided within a certain thickness specified in settings by **Nd**. Two new options appear in the Cl-menu:

```
/3.  Remove a range of atoms from the cluster
\4.  Keep a range of atoms in the cluster:  remove the rest
```

that allow you to remove unwanted atoms from the cluster. Next time **N** is chosen, you will be prompted to give numbers of atoms around which the nearest neighbours are to be constructed. This process can be continued almost indefinitely and leads to a cluster of atoms that is a fragment of the bulk crystal.

At the end of the construction, by pressing option **H**, it is possible to terminate some or all atoms with hydrogens. Their distance from the boundary cluster atoms is determined by **Hd** in the settings. If the distance appeared wrong, you can either undo this step (option **U**), change the setting (option **Nd**) and then repeat the construction. Alternatively, the atom-H distances can be corrected using option **mH**.

Note that, when specifying a range of atoms (options **N** and **H**), it is possible to include internal atoms of the cluster as well in the lists; this is very convenient and does not have any effect at all as only external atoms will be considered anyway.

At each step, the current arrangement of the cluster atoms can be seen in **Cc**. If the option **XY** is on (either at ***Xmol*** or ***Ymol***), then you can preview the cluster by invoking ***Xmol*** or similar software and reading in the file `geom.xyz`. This file is dumped automatically in the current directory every time a change is made.

Additionally to the above mentioned options, it is also possible to change species (or "rename") a set of atoms (option **5**). The option **T** allows a different (and, in many cases, more convenient) way of choosing atoms "by lists" (Section 1.2). At any point the cluster can be shifted as a whole (options **O** and **Cm**).

At the end of the construction some or all atoms can be moved to new positions, option **Mv**, seen Section 3.6.13.2. Finally, the *whole* system can be rotated (option **Rt**, see Section 3.6.13.3). Note that either of the options **Mv** and **Rt** has been used, it will not longer be possible to add more atoms to the system and/or terminate boundary atoms with hydrogens, i.e. options **N** and **H** will be disabled (unvisible).

Finally, the cluster geometry can be saved in either an `xyz` or a `mol` file formats, the latter is an internal format of the **Cl** option. Files in the `mol` format can be read in by the option **R** of the Cl-menu (e.g. from the previous unfinished project).

### 3.6.5 M - Complex cell modifications

An extremely powerful option! It allows you to modify your existing geometry in a number of ways and build up your final periodic cell. You will be given an extended menu with self-explanatory options. You should be able to change your lattice vectors; extend your cell; rotate, shift the system; add, remove, shift atoms; change their species; construct a slab for the surface calculation using e.g. Millers indices, etc. What is more, after every step a `geom.xyz` (xyz-format) file with the system geometry is written in the current directory so that you can preview your cell on the fly as you build it using a molecular viewer (e.g. ***Xmol***)! In fact, you can even preview it as an extended cell (the option **Bb**). Every step can also be undone.

The M-menu looks like this:

```
>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<
...  CURRENT lattice vectors for the supercell ARE...
AJ(1):  10.88160 0.00000 0.00000
AJ(2):  0.00000 10.88160 0.00000
AJ(3):  0.00000 0.00000 10.88160
>>>>>>>>>>>>>>>> Current species are:  <<<<<<<<<<<<<<
Si O
========= CHOOSE AN OPTION:
/==================================================\\
|| HELP: a complex transformation is obtained by ||
|| applying elementary operations one after another||
\\==================================================/
1.  General shift of the system (new origin)
2.  Shift the system to the center of mass
3.  Rotate the system (the same origin)
4.  Move an atom to an equivalent position
Rf.  Choose equivalent positions wrt reference input file
44.  Cluster atoms around particular point
6.  Choose unit cell with respect to Miller indices:
   1st 2 lattice vectors will be in the (hkl) plane
Ex.  Diagonal "breeding":  construct a supercell
Su.  General "breeding":  construct a supercell
8.  Create/modify SLAB: change the 3rd(z) lattice vector
9.  Arbitrarily change lattice vectors (e.g.  for molecules)
10.  Manually add an atom to the cell
```

```
Ad.  Add atoms from another file
B1.  DIVISIIONS: divide up the system into molecules
    DIVISIIONS: current number of divisions = 0
/11.  Remove a range of atoms from the cell
\12.  Keep a range of atoms in the cell:  remove the rest
13.  Rename a range of atoms:  change species
Mv.  Move a range of atoms to another general position
Rt.  Rotate a range of atoms about the X,Y,Z axes
--------- operations with multiple lists of atoms ------------
T. Tag atoms (specify multiple lists):  currently OFF
------------- g e n e r a l s e t t i n g s ---------------
UU. ****** RESTORE the original serting ******
U. ********** UNDO the last step *************
XY. Produce <geom.xyz> file after every change for Xmol
Hs.  H atoms to be added to <geom.xyz> file:  NO
An.  [For input] Coordinates are specified in:  <AtomNumber>
Co.  Show current atomic positions in fractional/Cartesian
Sy.  Show the point symmetry
Bb.  Set the size of the breeding box for visualisation
>>>>> Current setting for the breeding box:  <<<<<
[ 0...  0] x [ 0...  0] x [ 0...  0]
>>>>> extension = 1, # of atoms= 65
W. Write <geom.xyz> file to preview using current breeding
S. Save.
Sf.  Save SIESTA code to symmetrise forces
Q. Proceed/Quit
----------> Choose an appropriate option:
```

At the top, the current lattice vectors and species are shown. What you can do is explained below:

- **1** - all atoms in the cell will be displaced to a new position, the displacement vector will be asked in units corresponding to the setting **An**
- **2** - similar, but the displacement vector will be calculated from the atomic masses
- **3** - a rotation of the coordinate system is specified by giving new directions for a pair of two existing primitive lattice vectors using the *existing coordinate system*; of course, the angle between them should be preserved! This may be an inconvenience, but in some cases the choice is obvious
- **4** - move an atom to an equivalent position by adding lattice vectors to its current coordinate; this may be e.g. useful when building up a slab
- **Rf** - some codes such as ***VASP*** change atomic positions in the cell in such a way that their fractional coordinates become between 0 and 1; this may be not really wanted since after the relaxation (with ***VASP***) it may be very difficult to recognise the cell; a cure is here: in this option upi will be prompted again to the Input-menu (Section 3.4) to read in another input file (e.g. before the relaxation) the atomic coordinates in the two geometries will be compared and proper lattice vectors applied to the relaxed atoms to "return" them to the original unit cell

- **44** - atoms in the system around a given point (atom) are shown that cluster around it; more specifically, the displayed atoms have fractional coordinates between -0.5 and 0.5 if the centre of the coordinate system is positioned at the chosen atom

- **6** - after specifying three Miller indices, a new equivalent set of primitive translations for the cell is given in which the first two vectors $\mathbf{A}_1$and $\mathbf{A}_2$ lie within the plane characterised by the Miller indices; use the option **3** above to rotate the whole cell so that your plane becomes perpendicular to the $z$ axes (if desired)

- **Ex** - extend you cell by elongating the three current primitive translations, i.e. the transformation matrix of Eq. (3.6.4) is diagonal: $T_{ij} = \delta_{ij}n_i$; the three integers $n_i$are to be specified

- **Su** - the current cell is extended using a general transformation matrix **T**

- **8** - the third lattice vector can be arbitrarily changed here; this is necessary in creating a slab out of a bulk cell assuming that the vacuum gap is opened via the third lattice vector; note that atomic positions in the cell will not change

- **9** - all three lattice vectors can be changed; this may be useful in calculating a molecule or a cluster, if the distance between images must be changed; of course, atomic positions inside the cell do not change

- **10** - an additional atom is added to the cell; its position must be unique, otherwise, the atom is not added

- **Ad** - similar, but in this case all atoms from another file (that may be written even in a different format) can be added to the current system; the Input-menu (Section 3.4) is called here

- **B1** - the whole system of atoms can be split into subsets, “molecules”; you specify sets of atoms (using the same format as for the tagging option, **T**), and divide up the whole system into molecules 1,2,3, etc., until you divide it all. When the new set overlaps with those chosen before, “overlapping” atoms do not change their molecule number. This option is used (see later **B2** and **B3**) to facilitate BSSE and difference density calculations.

- **11**, **12** - these two options remove a set of atoms from the cell

- **13** - this renames a set of atoms, i.e. it changes their species into another species; this may be useful, e.g. in changing the lowermost atoms of the slab into hydrogens to terminate properly the slab and thus simulate the bulk at the lower surface

- **Mv** - move a set of atoms to another position (Section 3.6.13.2)

- **Rt** - rotate a set of atoms (Section 3.6.13.3)

- **Sf** - similarly to the same option of the **Pc** menu (Section 3.6.3), it saves `for_SIESTA_constr.file` file with the ***f77*** code that will keep point group symmetry of the system during geometry relaxation with ***SIESTA*** (see Section 3.6.3 for more details).

Normally, a set of atoms is chosen by two atomic numbers, in which case all atoms with the numbers in between are also chosen. If it is required to have a more complex non-contagious set of atoms in the list, use the option **T**, that allows you to do this (see Section 1.2). Note that all atoms added to the system from another file (option **Ad**) are tagged automatically, so that you can move and/or rotate them.

Other useful options (in seetings) include:

- **U** - undo the last transformation

- **UU** - restore the original geometry (the one with which you entered the M-menu initially)

Once the necessary geometry is created (the lattice vectors, number of atoms in species and their positions in space), it can be saved using option **S** (Section 3.6.13.1).

New options **B1** (described above), **B2** (only available for ***SIESTA***) and **B3** were added in version 3.94h16. The latter two options appear when the whole system is completely divided into molecules:

```
B1.  DIVISIONS: redo divisions from scratch
    DIVISIONS: their current number = 2 <- complete
B2.  BSSE + density difference script
B3.  Generate script for density difference
```

By applying option **B2**, the BSSE correction calculation is greatly facilitated. The following happens:

- the user is asked if the electron density files should also be saved for each molecule; this is useful if density difference calculations for each molecule are planned at a later stage;

- then, directory `BSSE` is created, which contains subdirectories `1, 1-ghost, 2, 2-ghost,` etc. for every molecule; the directories contain the corresponding input files for the energy calculation: if directories `1,2,` etc. contain input files just for the molecules, directories `1-ghost, 2-ghost,` etc. contain input files of the same molecules with the ghost orbitals of all other molecules included explicitly;

- the script ***BSSE_script.run*** is saved which, if executed (the correct path to ***SIESTA*** should be edited if required), will go through all the input files in the `BSSE` directory, run ***SIESTA*** for each of them, and, finally, will work out the BSSE correction; results of the calculations will be summarised in the file `BSSE_info`.

After the calculation is finished, the subdirectories in BSSE will have all the output files, including the electron densities (RHO files) in directories 1,2, etc., if this option was chosen.

Option **B3** is restricted to only electron densities of the molecules and is thus useful for ***VASP*** users as well. It works in a manner similar to that of **B2**:

- directory Dens_Diff is created with subdirectories 1,2, etc. for each molecule; each subdirectory contains the corresponding ***VASP*** or ***SIESTA*** input file for the density calculation;

- the script ***Dens_Diff_script.run*** is saved which will perform the required calculation;

- the output files and the electron density files will be stored in the respective directories of the molecules 1,2, etc.

## 3.6.6 KP - k-points generation

### 3.6.6.1 KP-menu

The corresponding menu for the **k**-points (KP-menu) looks like this:

```
/------------------------------------------------\\
|................  K-point menu .................|
\\------------------------------------------------/
0.  BULK calculation
NEW reciprocal lattice vectors (up to 2*PI):
BJ0(1):  0.09190 0.00000 0.00000
BJ0(2):  0.00000 0.09190 0.00000
BJ0(3):  0.00000 0.00000 0.09190
1.  DOS
2.  BAND: eigenvalues along a k-direction
3.  Self-consistent (ordinary) calculation
Q. Return to the main menu
Specify the character and press ENTER ------->
```

The option **0** allows one to choose between bulk and slab calculation. It is assumed here that the vacuum gap between slabs runs along the third lattice vector $\mathbf{a}_3$. Although this vector may not be very large in actual calculation, in most cases related to the reciprocal space, it is convenient to assume that it is of an infinite length. This in turn implies that the Brillouin zone (BZ) in the case of a slab calculations is effectively 2D (planar, compressed in the third dimension). Thus, any **k**-point to be constructed in this menu will have its component along the third reciprocal vector $\mathbf{b}_3$ (the reciprocal and direct primitive translation vectors are related via $\mathbf{a}_i\mathbf{b}_j = 2\pi\delta_{i,j}$) removed, i.e. only $\mathbf{b}_1$ and $\mathbf{b}_2$ will be used.

Other options of the menu allows you to choose the **k**-points depending on the type of the DFT calculation to be performed:

- **1** - the most sophisticated option, it allows one to use the point symmetry of the system and split the BZ into many tetrahedra for the calculation of the electronic DOS. This is specifically described in the coming Sections below

- **2** - **k**-points along a direction in the BZ can be chosen; this is useful for a band-structure calculation, when dispersion curves, $\varepsilon_n(\mathbf{k})$, for each electronic band $n$ are to be calculated

- **3** - **k**-points for a self-consistent calculation are chosen using the Monchorst-Pack method (either two or three integers to be given in the 2D and 3D cases, respectively); only inversion is assumed in this case (i.e. the points **k** and $-\mathbf{k}$ are assumed to be equivalent)

Once **k**-points have been selected, a new option **S** appears (save, Section 3.6.13.1) that allows you to save the geometry and the **k**-points. This is ok for ***SIESTA*** since everything is contained in a single input file. However, it is a little inconvenient in the case of ***VASP*** since the **k**-points are always assumed to be in a different `KPOINTS` file (so that geometry may be saved more than once).

#### 3.6.6.2 DOS

Firstly, the point group of your system is generated as explained in Section 3.6.13.4. When the point group is built, you will be asked about the number of "*divisions*". The larger this number is, the larger is the number of tetrahedra generated and thus the larger is the number of nonequivalent **k**-points. All this vital information is shown on the screen as you play with the generator trying different number of divisions. It is worth mentioning that sometimes with the larger number of divisions (and the larger number of tetrahedra) you can get *the same* or only slightly larger number of **k**-points. Of course, the larger number of tetrahedra, the better.

Once the **k**-points are generated, a file `brill.dat` is written which contains an important information for the DOS calculation to be done by the ***lev00*** routine, see Section 4.6.2.

### 3.6.7 Ew - Coulomb potential inside the unit cell

Coulomb potential at any desired point in the cell can be calculated here employing point charges on atoms specified manually. The Ew-menu:

```
.............MENU for Madelung potential ..........
.........  Change these parameters if necessary:  .....
1.  Point of interest (Angstroms):  ( 0.05194, 0.05194, 0.03418)
Point of interest (fractionl):  ( 0.00477, 0.00477, 0.00314)
3.  Charges on species:  KNOWN
4.  Charges on ions:  KNOWN
Ok.  Adopt present setting and calculate the potential <- Done!
>>>> Madelung potential:  -.195715E-02
-------------- g e n e r a l s e t t i n g s ----------------
2.  Charges on all species are the same:  YES
An.  [For input] Coordinates are specified in:  <AtomNumber>
Co.  Show current atomic positions in fractional/Cartesian
Sy.  Show the point symmetry
Ew.  Ewald constant:  0.162885E+00
5.  Precision of the summations:  0.100000E-09
6.  Multiplication factor:  0.100000E+01
Vb.  Verbose:  NO
```

```
    Q. Quit:  skip the step and do NOT proceed.
    ------> Choose the item and press ENTER:
```

First, specify charges on atoms (options **3** and **4**). It is either possible to do it separately for every atom in the cell or only for different species (setting **2**). Then, choose the point where the potential is to be calculated (option **1**). The Ewald constant is chosen using a special algorithm; however, it can be changed. Finally, the potential is calculated using **Ok**. A file `madel.dat` is produced with all the results of the calculation.

## 3.6.8 Calculation of phonons (atomic vibrations)

### 3.6.8.1 Theory: method of displacements and forces

The group theory is heavily used in all options related to vibrations which are described in this Section. Let $G$ be a symmetry group of the system of interest. Three cases are pressently distinguished here:

- **Vm** - a molecule, in which case translations are completely ignored and only point group operations $g \in G$ are used, Section 3.6.8.5
- **Vc** - a crystal, in which translations are taken into account only when checking equivalence of atoms; however, still the point group $G$ is used, Section 3.6.8.6
- **Vs** - a crystal, in which translations are taken into account when checking equivalence of atoms and in builing up the group $G$, Section 3.6.8.7; in this special case so-called internal translations that make internal atoms of a supercell equivalent are used in $G$; $G$ is an *Abel* group; if the supercell was constructed by extending (“breeding”, see Section 3.6.5) the primitive cell $N$ times, the group of internal translations $G$ will contain $N$ elements including the zero translation (as unity); note that point group operations are not considered here (if they were, it would have been nearly a real space group then!); however, using this option it is possible to assign the phonons to the **k** points in the original BZ of the crystal (corresponding to the small primitive unit cell).

A consideration below is not based on any particular choice of the group $G$. The main idea of the method used is based on the expansion of the total system energy (per cell) in atomic displacements:

$$E = E_0 + \sum_i f_i u_i + \frac{1}{2}\sum_{i,j}\Phi_{ij}u_iu_j + \frac{1}{3!}\sum_{i,jk}\Phi_{ijk}u_iu_ju_k + \frac{1}{4!}\sum_{i,jkl}\Phi_{ijkl}u_iu_ju_ku_l + \ldots \quad (3.2)$$

where $E_0$is the total energy at zero displacements $u_i$. Here the index $i$ refers to all degrees of freedom of the atoms in the cell (their Cartesian coordinates). The matrices $\Phi_{ij}$, $\Phi_{ijk}$, etc. characterise the system response to the atomic displacements. Note that in spite of the fact that summation over $i$, $j$, etc. is performed within the cell, the method that follows gives exact phonons (see [1, 2, 3]) in the **k** points corresponding to the given extention unit cell → supercell.

The matrix $\Phi = \|\Phi_{ij}\|$ is the force constant matrix (FCM) sought for. If we know it, the dynamical matrix can be calculated as $\mathbf{D} = \left\| M_i^{-1/2} \Phi_{ij} M_j^{-1/2} \right\|$, where $M_i$ is the mass associated with $i$. To find $\Phi$, we use the method of finite displacements: atoms in the unit cell are displaced by small amounts $u_i$, the forces on all atoms are calculated, and these are used to obtain elements of $\Phi$.

Three methods are presently implemented that require various amount of calculation. In the first method, only one displacement per degree of freedom (e.g. atomic displacement alonf some Cartesian axis) is used. Then, ignoring all anharmonic terms in Eq. (3.2) and assuming that all atoms are in mechanical equilibrium ($f_i = 0$ for $\forall i$), one obtains for the forces on atoms for a single displacement $u_j$:

$$F_i = -\frac{\partial E}{\partial u_i} = -\Phi_{ij} u_j \text{ and } \text{ thus } \Phi_{ij} = -\frac{F_i}{u_j} \tag{3.3}$$

Therefore, in this method, in order to obtain the $ij$ element of the FCM, one can displace the degree of freedom $i$ and calculate the force on the degree $j$; alternatively, since the FCM is symmetric, one can displace $j$ and calculate the force on $i$. The first method requires very small displacements since anharmonic terms are not included.

In the second method one uses two displacements, $u_j$ and $-u_j$ for each degree of freedom. In this case the system may not be at equilibrium (i.e. $f_i$ may be non-zero), and one can keep all odd terms in the energy expression (3.2). Denoting by $F_i^+$ and $F_i^-$ the forces on $i$ due to both displacements of $j$, one obtains:

$$\Phi_{ij} = -\frac{F_i^+ - F_i^-}{2u_j} \tag{3.4}$$

Finally, it is also possible to take into account the term of the 4th order with respect to the atomic displacements in the energy expression (3.2). This would require to use four displacements $\pm u_j$ and $\pm v_j$. In this case the elements $\Phi_{ij}$can also be obtained, we do not give the final expression.

The method described above is referred to in the code as the "*Force method*". There is also another method implemented, called the "*Energy method*" that requires many more DFT runs. Historically, this method was implemented first in ***tetr*** and still remains in the code; we shall not consider it here (see `README` file for details on this method).

#### 3.6.8.2 Theory: practical side

The precision of the calculation can be checked in the following way. For simplicity, we consider the first method. We start from $u_1$ and in a single energy calculation (without relaxation, of course!) all forces $F_i$ are determined. Thus, in a single calculation, the whole row, $\Phi_{1i}$ of the FCM is obtained. Then, we take $u_2$ and repeat the calculation. It gives the complete second row, $\Phi_{2i}$. This way, the whole matrix is obtained. However, the FCM should be symmetric. Due to the higher anharmonic terms that were dropped in deriving the working expression for the FCM, Eq. (3.3), this condition, however, will not be satisfied. Thus, by checking the relative difference between elements $\Phi_{ij}$ and $\Phi_{ji}$, one can check its precision.

There are basically two ways in improving precision. Firstly, one can reduce the trial displacements $u_i$. This is tricky as the forces on atoms are calculated numerically in any DFT code, so that forces may become unrealiable if $u_i$ are too small. Note that it is vital to have reliable forces in the DFT code you use. Normally, it is necessary to use as high precision in the code as possible, especially, if small displacements $u_i$ are to be used. The other option is to use more displacements, at least two (works up to the 4th order in the energy expression). Still, make sure that the forces are at least two orders of magnitude more accurate than in ordinary ground state relaxation calculations.

#### 3.6.8.3 Theory: use of symmetry

The usage of symmetry may substantially reduce the number of DFT runs required to obtain all elements of the FCM. The idea is the following:

For the given group $G$, all cell atoms can be split into "orbits" or "stars" of atoms, $\xi$. Atoms within an orbit can be obtained from each other by applying an appropriate symmetry operation $g \in G$; atoms from different orbits cannot be related to each other in this way. The idea is to find appropriate linear combinations of the atomic displacements so that the number of degrees of freedom be reduced.

This is done, first, by finding symmetry-adapted displacements for each orbit separately:

$$\eta_{\Gamma\xi f\alpha} = \sum_i U^{(\Gamma)}_{\xi f\alpha,i} u_{\xi i} \tag{3.5}$$

where $u_{\xi i}$ are atomic Cartesian displacements for all atoms belonging to the orbit $\xi$, $\Gamma$ is an irreducible representation, $f$ - its repetition and $\alpha$- dimension index (for 2D and 3D representations). Combining all symmetry-adapted displacements (for all orbits and all represnetations), we get the same number of degrees of freedom as originally, when Cartesian displacements $u_i$ were used. Therefore, one can write

$$\eta_\nu = \sum_i U_{\nu i} u_i \tag{3.6}$$

where the transformation matrix $U = \|U_{\nu i}\|$ is a square matrix. This matrix can be constructed using the projection operator method well known in the group theory (see, e.g. Ref. [L. Kantorovich and Livshits- Phys. Stat. Sol. (b) **174** (1992) 79]). It can also be constructed as a unitary and real matrix. The latter property is obtained by combining 1D complex conjugate irreducible representation into a single real 2D representation, see details in the same paper. Thus, one can relate atomic displacements back to the symmetry-adapted ones via

$$u_i = \sum_\nu U_{\nu i} \eta_\nu \tag{3.7}$$

or $\mathbf{u} = \mathbf{U}^T\eta$ in the matrix form. Inserting these expansions into the energy expression (3.2), one obtains an important result:

$$E - E_0 = \sum_{\Gamma\xi f\alpha} F_{\Gamma\xi f\alpha}\eta_{\Gamma\xi f\alpha} + \frac{1}{2}\sum_{\Gamma\alpha}\sum_{\xi f\xi' f'} \Phi^{(\Gamma)}_{\xi f,\xi' f'}\eta_{\Gamma\xi f\alpha}\eta_{\Gamma\xi' f'\alpha} + \dots \tag{3.8}$$

Importantly, the sum

$$\sum_{i\in\xi}\sum_{j\in\xi'} U^{(\Gamma)}_{\xi f\alpha,i}\Phi_{ij}U^{(\Gamma')}_{\xi' f'\alpha',j} = \delta_{\Gamma\Gamma'}\delta_{\alpha\alpha'}\Phi^{(\Gamma)}_{\xi f,\xi' f'} \tag{3.9}$$

is diaginal in $\Gamma$ and does not depend on the dimension index $\alpha$. It is here where the simplification comes.

Therefore, the method works in the following way (again, for simplicity, the method of a single displacement is considered):

- firstly, a representation $\Gamma$ is chosen for the given orbit $\xi$
- then, atomic displacements are obtained for all atoms in the orbit using Eq. (3.7) (atoms of other orbits will not be displaced in this case)
- a DFT code is run for the new geometry and the Cartesian forces on *all* atoms of the cell are obtained
- the Cartesian forces $F_i$ behave similarly to atomic displacements; therefore, from all $F_i$ one can obtain their symmetry-adapted counterparts using
$$F_{\Gamma\xi f\alpha} = \sum_i U^{(\Gamma)}_{\xi f\alpha,i}F_i$$
- using these forces, the symmetry adapted elements of the FCM, $\Phi^{(\Gamma)}_{\xi f,\xi' f'}$, can be obtained as explained in Section 3.6.8.1; note that the matrix does not depend on index $\alpha$; therefore, it is sufficient to use $\alpha = 1$, which also reduces the number of DFT runs
- diagonalising the matrix $\Phi^{(\Gamma)} = \left\|\Phi^{(\Gamma)}_{\xi f,\xi' f'}\right\|$, one obtains squares of all phonon frequencies, $\omega^2_{\Gamma\lambda}$, for the given representation $\Gamma$, the index $\lambda$ numbering all solutions
- the eigenvectors, $\eta_{\Gamma\xi f\alpha}$, can be turned into actual displacements via Eq. (3.7); thus, normal coordinates for each mode $(\Gamma\lambda)$ can be obtained
- the procedure is repeated for another orbit $\xi$ and/or another $\Gamma$

Note that the FCM in Cartesian atomic displacements, $\Phi_{ij}$, can be restored from the matrices $\Phi^{(\Gamma)}$ using the inverse to the Eq. (3.9) transformation:

$$\Phi_{ij} = \sum_{\Gamma\alpha}\sum_{f\in\xi}\sum_{f'\in\xi'} U^{(\Gamma)}_{\xi f\alpha,i}\Phi^{(\Gamma)}_{\xi f,\xi' f'}U^{(\Gamma)}_{\xi' f'\alpha,j} \tag{3.10}$$

This transformation will prove useful if other atomic masses should be tried (isotops) and it will be used in option **V3** of the main menu, Section 3.6.8.9.

#### 3.6.8.4 Theory: Elimination of translations and rotations in the case of molecules

The method outlined in the previous Sections can be applied to molecules without hesitation. However, because molecules are free to displace and rotate in space as a whole, the first six (in the case of a linear molecule, five) frequencies will come out as zero. This can be dealt with exactly by eliminating translational and roatational degrees of freedom during the procedure of building the symmetry-adapted displacements. This is done by orthogonalising the displacements to six (five in the case of a linear molecule) special vectors: (i) three (two) vectors of the centre of mass representing the translation, and (ii) three vectors of the angular momentu, representing rotations. The orthogonalisation is performed using the standard Shmidt method. There will be six (five) linearly dependent vectors after this, which are discarded. Since the chosen six (five) vectors of translations and rotations are invariants, displacements remain symmetry-adapted after this procedure.

#### 3.6.8.5 Vm - Vibrations of molecules => preparation

To maximise the point group of your molecule, choose the centre of the coordinate system in the centre of mass (M-menu, option **2**), and then rotate the molecule appropriately (option **3** of the M-menu).

We shall use a water molecule $H_2O$ as an example throughout this Section (all the files for you to try are in `TETR/H2O` directory). The Vm-menu is like this:

```
______ Your crystal class group is Oh with elements:
UNIT C4Z C4Y PXY C2Z C3D S4Z3 -C4Z C23 I C3A S4Z
C3C C2Y P4 C24 C26 S3B C3B S4Y C25 S3C C22 -C4Y
PYZ C2X -C3C P1 C21 PXZ -C3B P2 -C3A P3 -C4X S3D2
-C3D S4Y3 C4X S3A2 S3A S3D S4X S4X3 S3B2 S3C2 P5 P6
______ Your largest symmorphic group is C2v with elements:
UNIT C2Z PYZ PXZ
.......  The group family is Oh ...
Molecule is NOT linear!
Shmidt:  initial number of vectors N= 6
Shmidt:  final number of vectors N= 6
>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<
MOLECULE: lattice translations IGNORED
========= CHOOSE an OPTION:
Fx.  Specify which orbits to fix
    Currently transl/rot constrains ARE applied!
Ir.  Collect info about symmetry adapted vectors <- undefined
-------------- G e n e r a l I n f o r m a t i o n ------------
Co.  Show general geometry information about orbits
Or.  Show detailed information about given orbit
Sy.  Show the point symmetry group info
Cn.  Show translational-rotational constraints:
    Current number of conditions = 6
---------------- G e n e r a l S e t t i n g s ----------------
Am.  Atomic masses for every species:
15.99940 [ O] 1.00797 [ H]
XY. Format of <movie.xyz> file is for Xmol
Da.  Distortion amplitude:  0.01000
Nd.  Number of displacements (for [Au] only):  2
Hs.  H atoms to be added to <movie.xyz> file:  NO
Q. Proceed/Quit
----------> Choose an appropriate option:
```

At the top, the point symmetry is shown. It is also checked if the molecule is linear. Note that it is assumed that in this case the molecule should be oriented along the $z$ axis. Also, it is explicitly stated that translations are ignored as explained above.

There are many options in this menu. However, most of them are advanced options and you do not need to know them. These are used by the code when it calls itself from inside. Therefore, the user-related options are described first and only then we shall briefly mention other options.

How the system is split into orbits can be seen in **Co**, the symmetry in **Sy**, atomic positions in each orbit in **Or**. Note that the order of atoms may be different from their original order since orbits, not atomic numbers, are now used for ordering. In the table of the **Co** option, old atomic numbers are given for each orbit for convenience.

By default, the rotational-translational constraints (Section 3.6.8.4) are applied, their number is shown under **Cn** (this option will show explicitly the actual vectors of translational and rotational degrees of freedom that are used in orthogonalising the symmetry-adapted displacements). In this case all atoms of the molecule are used in calculating vibrations.

However, it is possible to *fix* some of the orbits[1] so that they would not participate in vibrations. Use **Fx** option for this, it implements lists (as in option **T**, Section 3.3) for entering orbits numbers. If there are any orbits fixed, the rotational-translational constraints, of course, are *not* applied.

Check if the atomic masses are correct (option **Am**) and choose the number of displacements you will be using (option **Nd**) for each degree of freedom (between 1, 2 or 4) as explained in Section 3.6.8.1.

The next step is to use **Ir**. When this is done, all symmetry related information will be obtained, and the symmetry adapted coordinates calculated. The number of symmetry-adapted coordinates $\eta_{\Gamma\xi f 1}$ (note that, as was mentioned eralier, it is sufficient to consider only the first row for 2D and 3D representations, so that $\alpha = 1$) for the given irreducible representation $\Gamma$ of the group $G$ is determined by the total number of repetitions (subscript $f$) summed over all orbits $\xi$. Some $\Gamma$ may not enter at all in which case there will be no symmetry-adapted coordinates. After **Ir** is applied, the menu changes to this:

```
WARNING: general info written to [vibr_info.dat]
>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<
MOLECULE: lattice translations IGNORED
========= CHOOSE an OPTION:
Fx.  Specify which orbits to fix
Currently transl/rot constrains ARE applied!
Ir.  Collect info about symmetry adapted vectors <- Done
------ A c t i o n s   w i t h   A c t i v e I r r e p ----------
1.  irrep (A1 ) is 1D; # of vectors = 2
4.  irrep (B2 ) is 1D; # of vectors = 1
------------------- A u t o m a t i c   o p t i o n ------------
Au.  FULLY AUTOMATIC: input_for.tetr + directories + scripts
--------------- G e n e r a l   I n f o r m a t i o n ------------
Co.  Show general geometry information about orbits
Or.  Show detailed information about given orbit
Sy.  Show the point symmetry group info
```

[1] You cannot fix individual atoms as this will break the symmetry and thus the splitting into orbits.

```
Cn.  Show translational-rotational constraints:
    Current number of conditions = 6
---------------- G e n e r a l   S e t t i n g s ----------------
Am.  Atomic masses for every species:
    15.99940 [ O] 1.00797 [ H]
XY. Format of <movie.xyz> file is for Xmol
Da.  Distortion amplitude:  0.01000
Nd.  Number of displacements (for [Au] only):  2
Hs.  H atoms to be added to <movie.xyz> file:  NO
Q. Proceed/Quit
----------> Choose an appropriate option:
```

You can see that in the case of a water molecule (group $C_{2v}$), which has 3 atoms, there should be 9 degrees of freedom. However, since rotational-translational constrains are applied and the molecule is not linear, there will be 6 constrains, so that the number of degrees of freedom is reduced to 9-6=3. These are distributed over two 1D representations: $\Gamma = 1$ (called A1) has two symmetry-adapted coordinates, while $\Gamma = 4$ (B2) has only one. Other representations $\Gamma = 2, 3$) have none (there are 4 representations in this group, see option **Sy**).

Next, choose the distortion amplitude **Da**. This will set the magnitude of the atomic displacements (in Å) to be used when calculating forces. The default value of 0.01 Å is appropriate in most cases. See the discussion in Section 3.6.8.2.

Finally, apply the automatic option **Au**. This will prompt you to the (input) R-menu (Section 3.6.13.1) in which you must indicate the DFT code you will be running and choose the input file, this should be the same you used when starting the option **Vm**. After you are done, the following help message appear:

```
.......................................................
=================> Now, do the following:  <==============
1.  check script run_DFT_script and edit if necessary
2.  run the script to calculate DFT forces
3.  run option V2 and/or V3 of tetr to get eigenvectors/values
......................................................
press ENTER when ready ...
```

that tells you what to do next.

In order to understand this message, you should know what has actually happened. When you run the option **Au**, a set of directories `irrep_1` and `irrep_4` is created in the current directory. The new directories contain all geometry input files (named `1`,`2`,`3`, etc.) with the required distrotion geometries. The files correspond to the chosen DFT code and are distributed between a number of subdirectories related to the number of displacements per degree of freedom and the chosen distortion amplitude. In addition to this, ***tetr*** creates a script `run_DFT_script` to run. For example, if you intend to run ***VASP***, the script for the water molecule created by ***tetr*** will be this:

```
#!/bin/csh
################################################
# is used to run DFT to get 2nd derivative matrix
################################################
#......  irrep = 1
set dir=irrep_1/d=0.01000
foreach file ( 1 2 )
/bin/cp $dir/$file POSCAR
vasp
/bin/mv OUTCAR $dir/OUTCAR_$file
```

```
end
set dir=irrep_1/d=-0.01000
foreach file ( 1 2 )
/bin/cp $dir/$file POSCAR
vasp
/bin/mv OUTCAR $dir/OUTCAR_$file
end
#......  irrep = 4
set dir=irrep_4/d=0.01000
foreach file ( 1 )
/bin/cp $dir/$file POSCAR
vasp
/bin/mv OUTCAR $dir/OUTCAR_$file
end
set dir=irrep_4/d=-0.01000
foreach file ( 1 )
/bin/cp $dir/$file POSCAR
vasp
/bin/mv OUTCAR $dir/OUTCAR_$file
end
```

It is assumed that the `vasp` executable is available in the current directory. When running the script, it loops over all representations and all the geometry files inside their directories. Each time, a geometry file is copied to the current directory and the ***VASP*** code is run. After each run, the output file `OUTCAR` that contains forces will be copied back to the same directory from where the input file has come, and its number will be appended to the file name. At the end of the run of the script, each directory with the input files corresponding to distorted geometries, will also contain all the files with atomic forces. Note that the script should be modified if it is to be run on a parallel machine.

The first part of the calculation (and the most difficult one!) is over. To proceed to the calculation of phonon frequencies, normal coordinates and phonon DOS, run options **V2** (Section 3.6.8.8) and/or **V3** (Section 3.6.8.9).

Thus, the content of the help message above should now be clear.

### A short note on advanced options

Instead of using the automatic option **Au**, it is actually possible to do everything manually, going through each of the representations and every vector (symmetry adapted coordinate) one by one. Do do this, instead of choosing **Au**, use the shown number of the representations, **1** or **4** in our example. Suppose, we choose **1**, then the menu will change into this:

```
>>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
MOLECULE: lattice translations IGNORED
========= CHOOSE an OPTION:
Fx.  Specify which orbits to fix
    Currently transl/rot constrains ARE applied!
Ir.  Collect info about symmetry adapted vectors <- Done
------ A c t i o n s w i t h A c t i v e I r r e p ----------
1.  irrep (A1 ) is 1D; # of vectors = 2 <=== now active
4.  irrep (B2 ) is 1D; # of vectors = 1
Vn.  Pair of vectors to be activated:  <= undefined
Go.  Show symmetry adapted coordinates (before constr.)
Sa.  Show final symmetry adapted displacements
Sw.  Write final symmetry adapted displacements to W_1.mat
Gw.  Write G-matrix for eigenvalue problem to G_1.mat
```

```
Vs.  Write <movie.xyz> movie file for each displacement
------------------- A u t o m a t i c o p t i o n ------------
Au.  Generate:  input_for.tetr + directories + scripts
-------------- G e n e r a l I n f o r m a t i o n ------------
Co.  Show general geometry information about orbits
Or.  Show detailed information about given orbit
Sy.  Show the point symmetry group info
Cn.  Show translational-rotational constraints:
    Current number of conditions = 6
---------------- G e n e r a l S e t t i n g s ----------------
Am.  Atomic masses for every species:
15.99940 [ O] 1.00797 [ H]
XY. Format of <movie.xyz> file is for Xmol
Da.  Distortion amplitude:  0.01000
Nd.  Number of displacements (for [Au] only):  2
Hs.  H atoms to be added to <movie.xyz> file:  NO
Q. Proceed/Quit
----------> Choose an appropriate option:
```

Some new options appear that correspond to the "*active*" representation which is 1 in the case above. These are:

- **Vn** - give a pair of vectors; since we have only 2 vectors in this representation, you can either choose 1 1, 1 2 or 2 2; the choice would correspond to the vectors numbers to be used to distort the geometry; as you see, you can use either one vector (1 1 or 2 2) or two (1 2)
- **Go** - show symmetry adapted coordinates before rotational-tarnslational constraints (if any) are applied
- **Sa** - show final symmetry adapted displacements
- **Sw** - write final symmetry adapted displacements to `W_1.mat` file
- **Gw** - write a so-called G-matrix (related to the necessary transformation of the atomic masses and required for the eigenvalue problem, to a `G_1.mat` file
- **Vs** - write a movie file <`movie.xyz`> for one of the vectors; the format of the movie file is controlled by options **XY** and **Hs**; the movie file will contain a sequence of geometry files in the required `xyz` format in which the given vector is changed sinusoidally; use Xmol or similar software to preview the given symmetry-adapte coordinate; note, that this is not yet the normal mode!
- **Au** - if this option is used when one of the representation is active, then all geometry files (with the corresponding directory substructure) will be created automatically; this, however, will only be done for this particular representation.

#### 3.6.8.6 Vc - Vibrations of crystals (point group) => preparation

This options works very similar to the previous **Vm** option. The only difference is that lattice translations are used when checking equivalence of atoms. In fact, from the user's point of view, there are no noticable differences, repeat all the steps of the previous case, Section 3.6.8.5.

#### 3.6.8.7 Vs - Vibrations of crystals (space group) => preparation

This options works similarly to the **Vm** option. The menu is this:

```
>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<
<<SUPERCELL>> ----- CURRENT lattice vectors:  ----
#           SUPERCELL              PRIMITIVE CELL       EXTENTION
(1) 10.88160 0.00000 0.00000 | 10.88160 0.00000 0.00000 |1 0 0
(2) 0.00000 10.88160 0.00000 | 0.00000 10.88160 0.00000 |0 1 0
(3) 0.00000 0.00000 10.88160 | 0.00000 0.00000 10.88160 |0 0 1
========= CHOOSE an OPTION:
Su.  General "breeding":  construct a supercell <= undefined
-------------- G e n e r a l I n f o r m a t i o n ------------
---------------- G e n e r a l S e t t i n g s ----------------
UU. ****** RESTORE the original lattice ******
Am.  Atomic masses for every species:
    28.08600 [Si] 15.99940 [ O]
XY. Format of <movie.xyz> file is for Xmol
Da.  Distortion amplitude:  0.01000
Nd.  Number of displacements (for [Au] only):  2
Hs.  H atoms to be added to <movie.xyz> file:  NO
Q. Proceed/Quit
----------> Choose an appropriate option:
```

You should first choose the cell extention using the option **Su**, identical to the one in the M-menu, Section 3.6.5. When you are done, the menu changes:

```
>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<
<<SUPERCELL>> ----- CURRENT lattice vectors:  ----
#         SUPERCELL                PRIMITIVE CELL       EXTENTION
(1) 10.88160 0.00000 0.00000 | 10.88160 0.00000 0.00000 |1 0 0
(2) 0.00000 10.88160 0.00000 | 0.00000 10.88160 0.00000 |0 1 0
(3) 0.00000 0.00000 10.88160 | 0.00000 0.00000 10.88160 |0 0 1
========= CHOOSE an OPTION:
Su.  General "breeding":  construct a supercell <= Done!
Fx.  Specify which orbits to fix
Ir.  Collect info about symmetry adapted vectors <- undefined
-------------- G e n e r a l I n f o r m a t i o n ------------
Co.  Show general geometry information about orbits
Or.  Show detailed information about given orbit
Kp.  Show all k-points
Sy.  Show the symmetry group info
---------------- G e n e r a l S e t t i n g s ----------------
UU. ****** RESTORE the original lattice ******
Am.  Atomic masses for every species:
28.08600 [Si] 15.99940 [ O]
XY. Format of <movie.xyz> file is for Xmol
Da.  Distortion amplitude:  0.01000
Nd.  Number of displacements (for [Au] only):  2
Hs.  H atoms to be added to <movie.xyz> file:  NO
Q. Proceed/Quit
----------> Choose an appropriate option:
```

It is now possible to recognise the familiar structure of the Vm- and the Vc-menus. The option **Sy** is, however, different now: it shows **k** vectors corresponding to each of the irreducible representations of the group $G$ of internal translations, see Section 3.6.8.1. The option **Kp** gives similar information.

To accomplish the first stage of the phonons calculation, follow the same steps as in the previous two cases:

- first, fix orbits (if necessary) with **Fx**
- use **Ir**

- choose the number of displacements (1,2 or 4) in **Nd** and the distortion amplitude in **Da**, or leave the default values
- run the automatic option **Au**, which will prompt you to select the DFT code and specify the name of the input file corresponding to the undistorted geometry
- edit (if necessary) and then run the `run_DFT_script` to calculate the forces for each input file
- run options **V2** or **V3** of the main menu to calculate phonon frequencies, normal coordinates and the phonon DOS.

#### 3.6.8.8 V2 - Vibrations => calculation [normal modes]

Here it is possible to calculate the phonon frequencies and the normal modes for every representation of the group. The V2-menu initially looks as follows:

```
>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<
MOLECULE: lattice translations IGNORED
========= CHOOSE an OPTION:
FE. Current working algorithm:  via Forces
--------------- E x i s t i n g --- I r r e p s ----------------
1.  irrep (A1 ) is 1D; dim = 2; G-files[ T], F_files[ T] W-files[ T]
------ A c t i o n s w i t h A c t i v e I r r e p ----------
---------------- G e n e r a l S e t t i n g s ----------------
XY. Format of <movie.xyz> file is for Xmol
Hs.  H atoms to be added to <movie.xyz> file:  NO
Da.  Distortion amplitude for movies:  0.05000
Q. Proceed/Quit
----------> Choose an appropriate option:
```

The fist line reminds the user which option (**Vm**, **Vc** or **Vs** of the main menu) was used to generate the distirted geometries. In the example above, this was **Vm**.

If any of the existing representations are displayed under "*Existing Irreps*", then all the necessary files exist and were read in successfully (including the forces on atoms calculated by running the `run_DFT_script` for distorted geometries). Otherwise, they will not be shown. At least one representation should be available to proceed.

Then, choose the representation number, **1** in the example above (A1). The dynamical matrix for this representation will be diagonalised and the corresponding eigenvalues and eigenvectors calculated. The precision of the calculation is checked and, if the relative difference between nondiagonal elements of the FCM is large, an warning will be given[2]. The menu changes into:

```
>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<
MOLECULE: lattice translations IGNORED
========= CHOOSE an OPTION:
FE. Current working algorithm:  via Forces
--------------- E x i s t i n g --- I r r e p s ----------------
1.  irrep (A1) is 1D; dim = 2; G-files[T], F_files[T] W-files[T] <=== now active
------ A c t i o n s w i t h A c t i v e I r r e p ----------
Sv.  Show eigenvalues/vectors
Sa.  Show frequencies and normal modes
```

[2] You can check the file `F_1.out` file for more details (in general, `F_#.out` file, where # is the representation number). The final FCM is obtained by averaging both $(i, j)$ and $(j, i)$ non-diagonal elements.

```
Vs.  Write xyz movie file for each mode
---------------- G e n e r a l S e t t i n g s ----------------
XY. Format of <movie.xyz> file is for Xmol
Hs.  H atoms to be added to <movie.xyz> file:  NO
Da.  Distortion amplitude for movies:  0.05000
Q. Proceed/Quit
----------> Choose an appropriate option:
```

Three new options appeared:

- **Sv** - shows the calculated eigenvalues (the squares of frequencies in inconvenient units) and eigenvectors (with respect to the symmetry adapted displacements); the displayed information is not very useful and may only be needed for an advance user; a file `EIGEN_1.res` will be written with the identical information;
- **Sa** - this gives all phonon frequencies (in cm$^{-1}$) and normal coordinates; the latter are given in terms of atomic diaplcements; a file `N_MODES_1.res` will be written with identical information;
- **Vs** - each normal coordinate can be run sinusoidally, written into a movie file `movie.xyz` (controlled by options **Hs**, **Da** and **XY**) and then previewed with the ***Xmol*** or similar molecular viewer.

#### 3.6.8.9 V3 - Vibrations => application [DOS, masses]

This option works similarly to the previous one, **V2**, although the initial part is very different. At the beginning, all the necessary files (including forces) are read in. The data must be available for *all* representation, otherwise, you cannot proceed. Once everything is read in, the following V3-menu appears:

```
>>>>>>>>>>>>>>>>>>>>>>>>>><<<<<<<<<<<<<<<<<<<<<<<<<<<
MOLECULE: lattice translations IGNORED
========= CHOOSE an OPTION:
Am.  Change atomic masses for a range of atoms
------ A c t i o n s w i t h v i b r a t i o n s ----------
Ei.  Solve the eigenvalue problem
------------------------ p h o n o n D O S -----------------
pD. Choose atoms for projected phonon DOS
-------------- G e n e r a l I n f o r m a t i o n -----------
Co.  Show information about atoms
Sy.  Show the point symmetry group info
---------------- G e n e r a l S e t t i n g s ---------------
XY. Format of <movie.xyz> file is for Xmol
Hs.  H atoms to be added to <movie.xyz> file:  NO
Da.  Distortion amplitude for movies:  0.05000
Nd.  Number of dispersions:  1
Sg.  Dispersion for DOS (cm^-1):  10.00000
St.  Step for DOS (cm^-1):  0.10000
Up.  Upper limit for DOS: not set
W. Write <geom.xyz> file to preview current geometry
Q. Proceed/Quit
----------> Choose an appropriate option:
```

At this point the FCM in the Cartesian representation, $\Phi = \|\Phi_{ij}\|$ is available via Eq. (3.10). Change masses[3] in **Am** (symmetry is not used, therefore, division of atoms

[3] If there were fixed orbits in constructing distorted geometries in one of the options **Vm**, **Vc** and **Vs**, it is not possible to change the mass of any of the atoms in the fixed orbnits.

into orbits is irrelevant here), and then solve the eigenproblem in **Ei**. After you are done, the natural logarithm of the product of all real nonzero frequencies will be shown just under **Ei**. This is useful for the calculation of the transition rates prefactors. Also, three new options appear just under "*Actions with vibrations*":

```
Am.  Change atomic masses for a range of atoms
------ A c t i o n s w i t h v i b r a t i o n s ----------
Ei.  Solve the eigenvalue problem <= Done!
    Nat.  logarithm of product of frequencies = 170.087612
Sv.  Show eigenvalues/vectors
Sa.  Show frequencies and normal modes
Vs.  Write xyz movie file for a mode
```

These are similar to those of the V2-menu:

- **Sv** - show eigenvalues/eigenvectors
- **Sa** - show frequencies and the normal coordinates
- **Vs** - make the movie file `movie.xyz` for the chosen normal coordinate

Under "phonon DOS", a new option **D1** appeared that allows one to calculate and preview (***gnuplot*** is called from inside ***tetr***) the total phonon DOS. When calculating DOS, each eigenfrequency is smeared out by a single Gaussian function with dispersion shown under **Sg**. It is possible to select the appropriate dispersion by displaying several DOS curves with different dispersions: use option **Nd** for this, and then **D1**.

If a set of atoms is chosen via **pD**, the phonon DOS projected on these atoms can be previewed using **D2**. In this case only a single dispersion (the first one if there are many) will be used.

Some other miscellaneous options controlling DOS:

- **St** - frequency step when calculating the DOS
- **Up** - upper limit for the DOS (the vertical axis on the DOS graphs)

You can also preview the current geometry, use **W** for this: a `geom.xyz` file will be written that can be previewed using ***Xmol*** or a similar viewer.

### 3.6.9 AF - Generation of the script for AFM modelling

Using this routine, one can create a shell-script named `run_AFM` which can move a part of the system with respect to the other on a 3D grid. For instance, this could be an AFM tip moving above the surface from the higherst point down at each lateral position. The main menu at the entry:

```
T. Atoms to be moved (e.g.  of the tip) <- Not given!
Vd.  Vertical displacements:  <- Not given!
Lt.  Elementary displacements along the lateral grid:
[1] ( 0.10000, 0.00000)
[2] ( 0.00000, 0.10000)
P1.  Points along 1st lateral directions:  1
P2.  Points along 2nd lateral directions:  1
Cd.  Code to be used for the calculation:  SIESTA
```

```
Z. Tip height for this geometry:  0.0
-------------- g e n e r a l s e t t i n g s ----------------
XY. Produce <geom.xyz> file after every change for Xmol
Hs.  H atoms to be added to <geom.xyz> file:  NO
Co.  Show current atomic positions in fractional/Cartesian
Bb.  Set the size of the breeding box for visualisation
>>>>> Current setting for the breeding box:  <<<<<
[ 0...  0] x [ 0...  0] x [ 0...  0]
>>>>> extension = 1, # of atoms= 10
W. Write <geom.xyz> file to preview using current breeding
Q. Proceed/Quit
```

Using **T**, one chooses atoms belonging to the moving part (the tip). Option **Vd** allows one to choose vertical displacements for each point (these can be unevenly specified), to preview the chosen displacements, use **VD**. Then, **Lt** is lateral displacements vectors and **P1** and **P2**-number of lateral grid points along the two directions. The script can be produced either for ***VASP*** or ***SIESTA*** (default: the read in input at the entry to the routine), use option **Cd** for that. Other options have their usual meaning. The actual script is generated by **G** which is available after all necessary setting are in place.

It is also possible to modify the input files by setting frozen atoms of the tip (with **FT**), frozen atoms in the surface (with **FS**) and save the input file with constraints (with **S**).

### 3.6.10 Representation/visualisation

#### 3.6.10.1 Cm - Compare geometry with the one in another file

Here you can read in another file (Section 3.6.13.1) to compare with the current geometry. The two files must be consistent with each other; ideally, they represent the same system, but atomic positions are different.

A table with atomic positions and the difference in atomic coordinates in the two files is shown on the screen. Note that the periodic symmetry is taken into account when comparing two points in space. Atoms with a difference in atomic coordinates larger than 0.1 Å will be indicated by the asterisk *.

#### 3.6.10.2 D - Distances between atoms

This is a calculator of interatomic distances. There are two possibilities that are selected by option **0**. The first one is called *pair_limiting*. The menu:

```
..............MENU for exploring distances ...........
.........  Change these parameters if necessary:  ......
0.  Method for exploring distances:  pair_limiting
1.  Pair of atoms to be studied:  1 and 1
2.  Number of spheres of images to be 3
3.  Calculate distances
4.  Quit:  skip the step and do NOT proceed.
```

In this case for the given pair of atoms (optin **1**) all the images in the increasing order will be given by option **3** (up to the desired number of spheres neighbours, given in option **2**).

The other option is *all_shortest*. Its menu is simply

```
.............MENU for exploring distances ...........
.........  Change these parameters if necessary:  ......
0.  Method for exploring distances:  all_shortest
3.  Calculate distances
4.  Quit:  skip the step and do NOT proceed.
------> Choose the item and press ENTER:
```

It gives distances between all species (including images in adjacent cells); only the shortest distance in each case is chosen and then, after ordering, are shown on the screen (option **3**).

#### 3.6.10.3 Ch - Check specified point in the cell

The menu of this option is extremely simple:

```
Choose an option:
1.  Check position in the cell
Co.  Show current atomic positions in fractional/Cartesian
----------------------- s e t t i n g --------------------------
An.  [For input] Position is specified in:  <Fractional>
Pr.  Precision while checking position on equivalence:  1.0D-004
Q. Leave
----------> Choose the appropriate option:
```

When choosing position in the cell (in the representation controlled by **An**) in option **1**, it will be checked agains existing atomic positions so that you would know if it is unique within some tolerance from **Pr**.

#### 3.6.10.4 Sb - Save in a box running across adjacent cells

This option allows you to save atomic positions that fit into a box of your choice The box may run across several adjacent cells, in which case equivalent atoms will also be chosen. The menu:

```
...........MENU for XYZ-in-a-box writer ............
.........  Change these parameters if necessary:.......
Cp.  The box center is at:  ( 0.00000, 0.00000, 0.00000)
Directions of the box sides are fixed along:
        1 < 1.00000, 0.00000, 0.00000 >
        2 < 0.00000, 1.00000, 0.00000 >
        3 < 0.00000, 0.00000, 1.00000 >
L. Lengths of the box sides (in A) are:  1.00000 1.00000 1.00000
Ca.  Calculate atomic positions
W. Write atoms into <geom_box.xyz> file
------- G e n e r a l s e t t i n g s ---------
An.  Coordinates are specified in:  <Angstroms>
Co.  Show current atomic positions in fractional/Cartesian
XY. Use the following format for <geom_box.xyz> file:  Xmol
Hs.  H atoms to be added to <geom_box.xyz> file:  NO
cT. Centre of a triangle of points/atoms
mD. Middle distance between two points/atoms
------ L e a v e t h e m e n u -------------
Q. Return to the previous menu
------> Choose the item and press ENTER:
```

You can:

- **Cp** - choose the box centre
- **L** - lengths of the box sides

- **Ca** - when **L** and **Cp** are chosen, this options will calculate all atomic positions that will fit into the box (inclusing images of the cell atoms)
- **At** - appears when **Ca** has been done; you can preview all atoms that were found fit into the box
- **W** - will write atoms into a `geom_box.xyz` file

The menu has a little calculator built in to help in choosing the box centre between atoms:

- **cT** - will calculate the centre of a triangle of points/atoms
- **mD** - will calculate the middle distance between two points/atoms

### 3.6.11 PE - PES: move atoms keeping the symmetry

An atom can be displaced from its positions, and all other atoms of the *orbit*[4] this atom belongs to (see Section 3.6.8.3) will be moved *in concert* to keep the symmetry. Atoms of the orbit are detected automatically. This is useful if one wants to calculate the potential energy surface (PES) with respect to full-symmetry displacements of atoms in the system. See also Sections 3.5.1 about symmetry.

First of all, you will arrive at the point group generator menu (Section 3.6.13.4). Use it to accept the group automatically detected or build a new one. Then the following menu will appear:

```
______ So, your system point group is C2h
       with elements:
           UNIT I P1 C21
Choose an option:
>>>>>> atoms are numbered across species <<<<<<<
1.  Specify atom to be moved and show all equivalent
2.  Choose the method of displacing atoms
M. Move it!
-------------- g e n e r a l s e t t i n g s ----------------
UU. ****** RESTORE the original serting ******
U. ********** UNDO *********** the last step
XY. Produce <geom.xyz> file after every change for Xmol
Hs.  H atoms to be added to <geom.xyz> file:  NO
An.  [For input] Coordinates are specified in:  <AtomNumber>
Sy.  Check the symmetry after distortion
Co.  Show current atomic positions in fractional/Cartesian
Bb.  Set the size of the breeding box for visualisation
>>>>> Current setting for the breeding box:  <<<<<
     [ 0...  0] x [ 0...  0] x [ 0...  0]
>>>>> extension = 1, # of atoms= 65
W. Write <geom.xyz> file to preview using current breeding
S. Save.
Q. Proceed/Quit
----------> Choose the appropriate option:
```

Apart from the self-explanatory options in the settings, the first tree options do the actual job:

[4] Atoms belonging to the same orbit can be obtained from each other by applying symmetry operations (here: point group operations).

- **1** - specify an atom that represents the orbit you wnat to displace
- **2** - choose how to displace and by what amount
- **M** - displace atoms of tbhis orbit

The new geometry can be written into a file with the desired format in a usual way (Section 3.6.13.1) via option **S**. After this is done, you can run the DFT code of your choice to calculate the energy.

### 3.6.12 Ss - Save SIESTA settings with geometry from another XV file

This option allows one to merge ***SIESTA*** `fdf` and `XV` files into a single `fdf` file named `INPUT_FILE-relaxed.fdf`. It can be useful e.g. for saving after the calculation the final `fdf` file with the relaxed geometry.

### 3.6.13 Miscellaneous menus

#### 3.6.13.1 S/R - Save/Read menu

This menu is used to choose the correct format for writing (saving) or reading geometry file, options **S** and **R**, respectively. For instance, S-menu is this:

```
...  writing geometry and k-points ...
................................................
SAVE/WRITE geometry/k-points in one of the following formats:
0.  MM input
1.  CASTEP input
2.  CETEP / old CASTEP input
3.  VASP input <=== active
4.  SIESTA input [*.fdf]
6.  XYZ-type file [ *.xyz ]
7.  Default:  file CONTCAR [VASP]
   Existing file will move to CONTCAR.bak
8.  Quit:  do NOT save
------------>
```

R-menu is similar, only the heading changes. In either case you should choose the format of the file and the file name to save or read.

#### 3.6.13.2 Mv - Move atoms

To move an atom to new position, the following menu appears:

```
---------- Specify the method to displace:  --------------
1.  DIRECTION: 2 points (e.g.  2 atoms):  primary and target
2.  DIRECTION: or a vector
3.  FINAL POSITION: in Cartesian coordinates
4.  DISPLACEMENT: in Cartesian coordinates
------------------ setting ------------------------------
5.  [For input] Coordinates are specified in:  <AtomNumber>
---------------------------------------------------------
6.  QUIT: return to the menu and do nothing!
----------> Choose an appropriate option:
```

that allows several choices for moving the atom(s). You will be returned up to the menu tree automatically after making your input. To skip this step and do nothing, choose **6**.

#### 3.6.13.3 Rt - Rotate atoms

A rotation of atoms requires specifying rotational angles. These are given in the following Rt-menu:

```
---- Specify ANGLES of rotations in the correct ORDER (!!!)  ----
---[ CLOCKWISE rotations if looking down from the axes top ]----
------ METHOD 1:  give 3 angles DIRECTLY and the order ---------
X. Angle about X axis => 0.00000 rad
Y. Angle about Y axis => 0.95568 rad
Z. Angle about Z axis => -0.76687 rad
O. Order of rotations :  ZY
------ METHOD 2:  by giving plane (3 pnts) to be _|_ to an axis -
Px.  If the plane after rotation be _|_ to X axis
Py.  If the plane after rotation be _|_ to Y axis
Pz.  If the plane after rotation be _|_ to Z axis
-------------- g e n e r a l s e t t i n g s ------------------
C. Central point is:  ( 0.00000, 0.00000, 0.00000)
An.  [For input] Coordinates are specified in:  <AtomNumber>
Un.  [For input] Angles are given in:  degrees
----------------------------------------------------------------
Q. Done.  Return back to the main menu!
Ab.  ABORT: return to the main menu, DO NOTHING!
----------> Choose an appropriate option:
```

It is either possible to specify angles (either in degrees or radians, option **Un**) around $x, y, z$ axes and the order in which rotations are performed (option **O**) directly (Method 1), or, alternatively, you may want to have some plane of atoms in the final rotated state to be perpendicular to one of the axes (Method 2). In the latter case the rotational angles and their order will be calculated. The central point about which the rotations will be performed, can also be chosen (option **C**). Finally, to confirm the selection, you should do **Q**. To skip this step, choose **Ab**.

#### 3.6.13.4 Point group generator

The point group generator allows you to build a point group of your Bravais lattice. [5]

How does the generator work? The generator first checks the point symmetry of the Bravais lattice ("Your crystal class group is ...") and then each operation of this group is applied to every atom in the cell. In this way, an actual point group is built ("Your largest symmorphic group is ...") and shown (the name of the group and all the elements as shown in Figs. 3.1 and 3.2). A menu which opens

[5] A note for advanced users: the group generator disregards possible fractional translations of the system and therefore generates only the largest possible group corresponding to a *symmorphic* lattice. The actual point group of the BZ might be richer if your lattice corresponds to some nonsymmorphic space group. At the moment, this is not checked automatically but it is possible to add this functionality (i.e.fractional translations). This is, however, beyond of what I can do at the moment as it also requires the full space group analysis as well - something to think about in the future!

up here allows you to either adopt this group ("Adopt the largest symmorphic group"), continue generation of the group by adding new elements from the crystal class group ("Proceed generation from") or start generation from scratch. In the last two cases you can add/remove elements, the generator multiplies all the elements with each other, adds new elements (if necessary) and then shows you the current group generated so far.

If the symmetry and/or special constraints imposed on atomic coordinates have not been used at the stage of atomic relaxation while running a DFT code, the symmetry generator may fail to determine the correct symmetry group due to numerical errors in the atomic positions. Therefore, to make your symmetry group as rich as possible, you may need to generate the symmetry manually. One method is to use the geometry of the *unrelaxed* system (if it should have the same symmetry). Other (brr!!!) would be to edit your geometry file to make sure that there is no any numerical noise in the atomic coordinates (in some cases this is a real challenge!). Alternatively, use the point group generator: it allows you to play with the symmetry operations by adding/removing them, each time the whole group is generated and shown. It is a useful exercise in the group theory as well, especially in understanding group *generators*.

## 3.7 Scripting capabilities of *tetr*

Since ***tetr*** uses the Text User Interface (TUI), it can easily be used in scripts as a general UNIX routine. This can be done by using a file with the ***tetr*** commands in the right order.

To illustrate the idea, let us consider a simple problem in which it is necessary to calculate a potential energy surface of a water molecule in which a single atom #2 (one of its hydrogens) is moved by small steps of $\sqrt{2}/10$ Å along the $x = y$ direction in the cell. The ***VASP*** code is used. The initial geometry is contained in the file `POSCAR.cm`, given below:

```
made by tetr; --> species:  O H
1.00000000000000
5.00000000000000 0.00000000000000D+000 0.00000000000000D+000
0.00000000000000D+000 5.00000000000000 0.00000000000000D+000
0.00000000000000D+000 0.00000000000000D+000 5.00000000000000
1 2
Selective dynamics
Direct
0.000000000000 0.000000000000 -0.013266064393 T T T
0.152276020145 0.000000000000 0.105285414074 T T T
-0.152276020145 0.000000000000 0.105285414074 T T T
```

All other ***VASP*** files are the same for this calculation. Then, the following ***tetr*** commands will do a single step:

```
3
POSCAR.cm
O
H
M
Mv
2 2
4
```

```
0.1 0.1 0.0
S
3
POSCAR.1
Q
Q
```

Collect all these commands in a file `inp_for_tetr`. Now, if ***tetr*** is executed with this input file read in from the standart input, e.g.

```
tetr < inp_for_script > out
```

, then the following will happen: first of all, the ***VASP*** input file `POSCAR.cm` will be read in, then a chain **M**→**Mv** commands is executed, and the atom #2 is moved by a vector (0.1,0.1,0.0). Finally, the new geometry is saved in a file `POSCAR.1`, after which ***tetr*** is stopped. The following `sch` script will then run the required calculation for 10 steps:

```
#!/bin/csh
cp POSCAR.cm POSCAR.ref
foreach step ( 1 2 3 4 5 6 7 8 9 10 )
   tetr < inp_for_script > out
   mv POSCAR.1 POSCAR
   vasp
   mv OUTCAR OUTCAR.$step
   mv CONTCAR POSCAR.cm
end
mv POSCAR.ref POSCAR.cm
```

At the end of the calculation, the current directory will contain ten `OUTCAR.1,...,OUTCAR.10` files for each point of the calculation.

# Chapter 4

# Utility *lev00*

## 4.1 What it can do

***lev00*** is an interactive menu-driven (and quite intelligent!) program with quite a few applications. It implements the Tex User Interface (TUI) similar to that of ***tetr*** (Sections 1.2 and 3.3). It has a rich functionality, e.g.:

- DOS/LDOS applications (presently works for ***VASP***, ***SIESTA*** [1]):
  - plot of the total and projected DOS for various choices of groups of states (or for the whole set of those) using the method of tetrahedra; one can preview several curves at the time for different choices of groups of states; this last option has proven to be useful for the identification of peaks in the LDOS;
  - the DOS/LDOS can be smeared out by a Gaussian; this can be done either analytically or numerically;
  - there is also a possibility to play around with various dispersions for the Gaussian smearing and compare DOS calculated with them at the same time; this helps to select the correct dispersion for smearing.

- Charge/spin/partial densities (presently works fully for ***VASP***; however, most of the density related options work for ***SIESTA*** as well):
  - plot of charge/spin densities or partial charge densities in the cell along a line (1-dimensional) or in a plane (2-dimensional)
  - explore the charge density: it seeks peaks and gives their positions in the space of the cell, heights, amount of charge involved, etc.; this has been proved to be especially useful for studying partial charge densities or wave

[1] An application to ***SIESTA*** has been recently added as described in this manual below; however, a patch to ***SIESTA*** 2.0 is required which enables inputing **k**-points directly into the code (ask Lev Kantorovich or Emilio Artacho).

functions for particular states in the direct space which were thought to be responsible for particularly interesting features in the LDOS;

- integration of the (partial) charge densities over a set of spheres of given radii; this allows you to gain an idea about the chemical nature of bonding in the system, charges on atoms, etc.; note that spheres may overlap so that you can construct a quite sophisticated region in space to study the density!
- one can cut out parts of the density, simulate it by point charges; this may be useful if a complimentary calculation is to be done (e.g. a Coulomb correction, polarisation calculation for a defect outside the cell, etc.)
- dipole and quadrupole moments of the cell (useful, frankly, only for molecules, or any densities that go to zero at the border of the cell)
- a density file in a `Cube` format that allows one to plot nice surfaces of constant density using e.g. ***gOpenMol***
- construction of a general linear combination of two charge densities; for example, one can subtract one density from another;
- recalculation of the charge density with respect to a new position of the coordinate system (*shift*); this is necessary if the ground state calculation charge density has been obtained for a lower symmetry position of the coordinate system[2], while e.g. for the DOS one has to do the band-structure calculation with as high symmetry as possible to reduce the number of **k**-points. Instead of repeating the ground state calculation for the new choice of the coordinate system, it is possible here to recalculate and then save the electronic density for the new position *prior* to the band-structure run of the DFT code. This would save one ground state calculation.

- the total Coulomb potential (including the electronic part[3]) across the cell: plotting

In all cases the information obtained can be previewed using ***Gnuplot*** directly from ***lev00*** in a form of 1- or 2-dimensional plots. It can be checked, corrected (e.g., the position of the plane for a 2-dimensional plot), previewed again and then finally printed out to a data and/or Postscript file.

## 4.2 Installation and general information

The distribution file archive (e.g. `lev00_3.0.tgz`) should be untarred and unzipped, e.g.

[2] Recall (Section 3.5.1) that the symmetry is fixed by the position of the Cartesian coordinate system.

[3] Only implemented for the old ***CASTEP***.

```
tar   -xzf   lev00_3.0.tgz
```

in a directory of your choice (e.g. ~/TOOLS). Then go to this directory and type ***make*** which will compile ***lev00***.

In practice, what you should do depends on the code you intend to use, and this will be described in the proper Sections 5, 6 and **??** for the ***VASP*** and ***SIESTA*** codes, respectively.

## 4.3 Main definitions

Here we provide some basic definitions for a number of quantities which are calculated by the code:

- A charge in a sphere $P_R$ of radius $R$ is defined as

  $$Q = \int_{\mathbf{r}\in P_R} \rho(\mathbf{r})\mathrm{d}\mathbf{r}$$

  where the sphere is centred at point $\mathbf{P}$.

- A charge in a volume $V$ made by several overlapping spheres,

  $$V = \bigcup_R P_R \, ,$$

  is defined similarly, with $P_R$ replaced by $V$.

- Partial electronic density for the state ("band") $n$ is defined as:

  $$\rho_n(\mathbf{r}) = \frac{1}{N}\sum_{\mathbf{k}\in BZ} |\psi_{n\mathbf{k}}(\mathbf{r})|^2 = \sum_{\mathbf{k}\in SP} \omega_{\mathbf{k}} |\psi_{n\mathbf{k}}(\mathbf{r})|^2 \tag{4.1}$$

  where SP means a set of (special) $\mathbf{k}$-points (i.e. the actual $\mathbf{k}$-point sampling), $\psi_{n\mathbf{k}}(\mathbf{r})$ is the wave function and $\omega_{\mathbf{k}}$ is the weighting factor.

- Partial electronic density for an *island* of bands $n \in [n_1, n_2]$ is

  $$\rho_{[n_1,n_2]}(\mathbf{r}) = \sum_{n=n_1}^{n_2} \rho_n(\mathbf{r}) \tag{4.2}$$

- Partial $z-$densities, integrated over the surface plane ($x$ and $y$), useful for slab (surface) calculations:

  $$\rho_{[n_1,n_2]}(z) = \int_{\mathbf{r}\in(layer)} \rho(\mathbf{r})\mathrm{d}x\mathrm{d}y \tag{4.3}$$

- Total electronic Density of States (DOS):

$$N(\epsilon) = \frac{1}{N} \sum_{n} \sum_{\mathbf{k}\in BZ} \delta(\epsilon - \epsilon_{n\mathbf{k}}) \tag{4.4}$$

  where the sum over $n$ is run over all states (both occupied and unoccupied) and $\epsilon_{n\mathbf{k}}$ are Kohn-Sham eigenvalues.

- Local DOS (projected on a sphere):

$$N_{\mathbf{P}_R}(\epsilon) = \frac{1}{N} \sum_{n} \sum_{\mathbf{k}\in BZ} A_{n\mathbf{k}} \delta(\epsilon - \epsilon_{n\mathbf{k}}) \tag{4.5}$$

  where the weighting factors are defined as:

$$A_{n\mathbf{k}} = \int_{\mathbf{r}\in P_R} |\Psi_{n\mathbf{k}}(\mathbf{r})|^2 \mathrm{d}\mathbf{r} \tag{4.6}$$

  with the integration performed over the sphere $P_R$ of radius $R$ centered at point $\mathbf{P}$.

- Angular-momenta $(s, p, d)$ projected DOS is defined as above, but with the following factors:

$$A_{n\mathbf{k}} = \int_{\mathbf{r}\in P_R} \sum_{m=-l}^{l} \left| \langle \psi_{n\mathbf{k}}(\mathbf{r}) \right| R_{nl}(r) S_{lm}(\widehat{\mathbf{r}}) \rangle \left|^2 \mathrm{d}\mathbf{r} \right. \tag{4.7}$$

  where $S_{lm}(\widehat{\mathbf{r}})$ is a real spherical function for the momenta $l = 0, 1, 2$.

- A DOS projected on a layer (for the slab calculations). The factors in this case are:

$$A_{n\mathbf{k}} = \int_{\mathbf{r}\in (layer)} |\Psi_{n\mathbf{k}}(\mathbf{r})|^2 \mathrm{d}\mathbf{r} \tag{4.8}$$

  where the layer is thought to be perpendicular to the $z$-axis.

## 4.4 A note on calculating electronic DOS

The calculation of the total/local/projected DOS is performed using the method of tetrahedra. In this method the whole Brillouin zone (BZ) is split into tetrahedra, the energies $\varepsilon_{n\mathbf{k}}$ at the vertices of which are to be calculated by the DFT code. These energies are collected in a file `band.out` (Sections 5 - **??**) and then (together with the information contained in the file `brill.dat` written by ***tetr***) are used by ***lev00*** to calculate the DOS.

Calculation of the volume integrals in the reciprocal space in Eqs. (4.4), (4.5) is done analytically in the method of tetrahedra. It is also possible to do a *smearing* (convolution) of your original (raw) DOS curves for every band, $N_n(\epsilon)$, by calculating instead the integral

$$\bar{N}_n(\epsilon) = \int_{E_{bot}-\gamma\sigma}^{E_{top}+\gamma\sigma} g(\epsilon - \varepsilon) N_n(\varepsilon) d\varepsilon, \tag{4.9}$$

Here $g(\epsilon - \varepsilon)$ is a Gaussian function with some dispersion $\sigma$; $E_{bot}$ and $E_{top}$ define the "raw" boundaries for each physical band and the factor $\gamma$ allows you to broaden them. The energy integral in Eq. (4.9) can be calculated in two ways leading to *slow* and *fast* methods:

- *slow* - since in the method of tetrahedra the raw DOS is always a set of continuous functions of energy in finite intervals (within each tetrahedron), the integration can also be worked out analytically and the result be expressed via exponentials and error functions; this method is slow, but yet exact on many occasions.

- *fast* - alternatively, the integral in Eq. (4.9) can aslo be calculated numerically.

## 4.5 A note on calculating volume integrals

In many cases (e.g. when calculating a projected DOS (LDOS), atomic charges or when analysing the density as to be described in later sections) real space integrals are calculated numerically. Two methods are implemented for those which are referred to as "*conserving*" and "*non-conserving*" algorithms:

- *the conserving method*: provides the correct charge inside a sphere or a layer, each grid point in the cell is scanned once and only those grid points contribute which are positioned inside the volume; when checking whether a grid point is inside the volume, lattice translations are also applied. As the region of integration increases (e.g. the radius of a sphere), the integration volume approaches that of the cell. Still, the integration volume will never exceed that of the cell. Thus, this method "conserves" the volume of the cell.

- *the non-conserving method*: a finer grid is constructed in the region of interest (a sphere or a layer) and the integrals are calculated by summing up contributions on this finer grid (an interpolation is used). This method does not give the volume conservation as the volume increases, howewer, it is more efficient provided the integration volumes are sufficiently small.

If the size of the cell is large enough with respect to the region of integration, than both methods should give close results.

The conserving method is extremely demanding and scales linearly with the size of the system. The non-conserving method scales linearly with the size of its own grid. However, since this grid is limited to the size of the integration region, the time of the calculation does not depend on the size of the system at all, so that this method is extremely efficient.

In the conserving method the original grid is used with the directions along the cell basic vectors $\mathbf{a}_i$. That is why this method is inappropriate for calculations of the angular momenta projected DOS for non-simple-cubic cells or for the calculation of the dipole/quadrupole momenta of a molecule. On the contrary, the non-conserving method has its own grid which is always cubic (along the Cartesian axes) so that it has the atomic symmetry and is ideal for these calculations.

## 4.6 Detailed description of options of *lev00*

### 4.6.1 Main menu

When the compiled code is started, the first menu is shown that asks you to choose the DFT code you are working with:

```
.................................................
Choose the input format:
V. open input as VASP
S. open input as SIESTA
Q. QUIT
------------>
```

Enter **3** for ***VASP*** and **4** for ***SIESTA***. If the former case, you will also be asked about the chemical species in your `CONTCAR`-like file. Then, the main menu appears:

```
.................................................
Choose the problem to solve:
============== DOS options =======================
0.  Choose dispersion for the DOS smearing
1.  Projected DOS calculation
3.  Total DOS: you need only -> band.out, brill.dat
============== DENSITY options ===================
D. total/partial electron density, electrostatic potential
SD. spin density
M. manipulation with charge densities
============== ELECTR. potential options =========
CP. Coulomb potential (point,line,plane)
Q. Quit
------------>
```

Not all the options are shown at the same time. Also, depending on the code, The menu is split into three sections:

- electronic DOS:
  - **0** - here it is possible to plot the total DOS for a number of dispersions used for smearing; this is useful if it is not clear beforehand what value to us; as other DOS related options (**11** and **3**) allow you to use only one dispersion, go here first to find it;
  - **3** - here you can preview and plot the total DOS; it is possible to smear the DOS as well as to plot several curves with various ranges of bands (states) $[n_1, n_2]$;
  - **1** - this is similar to the previous one, but also allows plotting the $s, p, d-$projected (local) DOS alongside the total DOS;

- options related to the electron (spin) total (partial) densities; in fact, any object written by a DFT code in the format of the density (like the electrostatic potential produced by ***VASP***, see Section 5) can also be processed here
  - **D** - a more powerful option, works for any files in the proper format corresponding to the activated DFT code; could be a partial or total density or an electrostatic potential, etc.
  - **SD** - the same as **D**, but works particularly for the spin density (appears only for spin-polarised calculations)
  - **M** - a linear combination of two densities can be calculated, works for ***VASP*** and ***SIESTA***
- **CP** - electrostatic potential of the lattice at a point, along a line or within a plane; in all cases works with point charges.

## 4.6.2 Calculation of the DOS

### 4.6.2.1 Physical bands

Generally, while running any of the DOS options, you will be given a table of "physical bands" in your system extracted from the eigenvalues, $\varepsilon_{n\mathbf{k}}$. The physical bands boundaries give the starting and the ending position of every energy region that contains coninuous energy levels. The individual bands (all energies $\varepsilon_{n\mathbf{k}}$ for the *given* state $n$ and *all* **k**-points) may overlap, in which case several bands will form a single physical band. These could be core, valence and conduction bands. In the following example there are 6 physical bands for a system containing 20 states $n$ altogether:

```
 _______> Structure of ‘‘physical’’ bands <________
Band 1 spans energy interval: [-19.8216,-17.9913] and states { 1, 1}
Band 2 spans energy interval: [ -6.4785, -2.6852] and states { 2, 4}
Band 3 spans energy interval: [ -1.2610, -1.0436] and states { 5, 5}
Band 4 spans energy interval: [ -0.1859, 15.1450] and states { 6, 8}
Band 5 spans energy interval: [ 15.2456, 37.8405] and states { 9,18}
Band 6 spans energy interval: [ 37.9769, 41.0874] and states {19,20}
```

The bands 1 and 2 are core and valence bands, the band 3 may (presumably) correspond to a local gap state while the remaining 3 bands are parts of the conduction band. This can be judged from studying energy intervals they span (energies are given in eV).

You have also numbers of states (as they appear in the file `band.out`) in this table. This is important as you can immediately see, for example, that only one state (the 5-th) is responsible for the third band, and thus this particular state may be a local state induced by a defect or irregularity in the lattice. Then, you can go and see the partial charge density, Section 4.6.3, associated with this state to find out, e.g. where in the cell it is localised.

#### 4.6.2.2 0 - Choose dispersion for the DOS smearing

Here ***lev00*** allows you to choose a reasonable dispersion $\sigma$ for your DOS smearing and the parameter $\gamma$ which controls the bands broadening, see Eq. (4.9). A menu which opens here is this:

```
.............MENU DOS - check SMEARING ..........
.....  Change these parameters if necessary:...
______________> DOS + Smearing <_______________
0.  THE CURRENT PICTURE IS: 1
1.  Maximum energy step for the plotting:  0.10000
2.  Smeared DOS/LDOS: ENABLED fast method
3.  Broaden band boundaries by 1.500 dispersions
4.  Number of dispersions to be examined:  ...  undefined ...
7.  Adopt present setting and calculate DOS
10.  Preview the current DOS
11.  PostScript file for the current DOS
12.  Upper limit to the energy-range of the plot:  DISABLED
13.  Quit:  do not proceed.
------> Choose the item and press ENTER:
```

You are first asked to choose a set of dispersions $\sigma$ of Eq. (4.9) in option **4**. The option will then change. For instance, if three dispersions 0.3, 1.0 and 1.5 are chosen, the option **4** will show as:

```
4.  Number of dispersions to be examined:  3
> Dispersions are:
0.30000 1.0000 1.50000
> Files for the plot:  dos.dsp1_1 , ...  , dos.dsp3_1
> Files with smearing plot:  dos.dsp1_1_ , ...  , dos.dsp3_1_
> Names of the PostScript files:  dos.dsp1_1.ps , ...  , dos.dsp3_1.ps
```

It now shows all the file names to be produced during the calculation. Note that the first number in any name changes from 1 to 3 (the number of dispersions), while the second one corresponds to the number of the current picture shown under option **0**. You can use this latter option to change the picture number in order to produce new files and not to overwrite the existing ones.

The quantity $\gamma$ of Eq. (4.9) is chosen in **3**.

When the dispersions are chosen, new options **8** and **9** appear that allow you to preview and save as a postcript file, respectively, each individual DOS as they are being calculated. The corresponding file names are gievn under option **4** as explained above.

As was mentioned in Section 4.4, there are two methods that can be used to calculate the DOS: *fast* and *slow*. The fast method for smearing is the default. It is possible to choose between the two methods in option **2**.

The calculation of the DOS curves is run by option **7**. After running the calculation, you can preview all curves simultaneously together with the raw DOS (no smearing) using **10** as shown in Fig. 4.1. This should allow you to choose the best possible parameters $\sigma$ and $\gamma$ which do not change significantly the structure of the raw DOS curve, and then use them in other DOS options of the main menu of ***lev00***. The appropriate postcript file is produced by **11**. Finally, the upper limit for the DOS is controlled by option **12**.

Finally, we remark on the energy step of option **1**. Normally, the default value of 0.1 eV is fine. The energy step here is needed only as a guidance as the actual step is

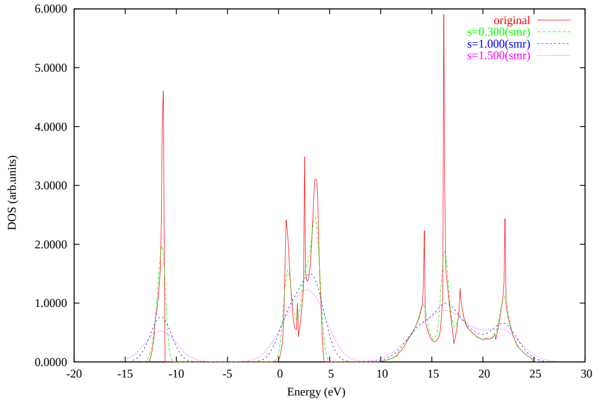


Figure 4.1: Total DOS for the MgO bulk for three values of the dispersion $\sigma$.

chosen individually for each physical band to ensure that there are at least 10 points even for very narrow bands.

#### 4.6.2.3 3 - Total DOS: you need only -> band.out, brill.dat

This option works similarly to the previous one. It calculates the total DOS (both raw and the smeared one). To run this option, two additional files are necessary: `band.out` containing energies and produced after the DFT run, and `brill.dat` produced by ***tetr*** during the generation of the **k**-points for the DOS option, see Section 3.6.6.2.

The new functionality here is an ability to calculate and plot the DOS curves for individual groups of bands (states). In other words, the index $n$ in Eqs. (4.4) and (4.5) runs over the chosen bands (states) only.

The menu:

```
..............MENU DOS/LDOS .......................
.....  Change these parameters if necessary:...
______________> Only total DOS <_______________
0.  THE CURRENT PICTURE IS: 1
2.  Maximum energy step for the plotting:  0.10000
   > Number of points for the DOS plot:  220
   > Steps for each "physical" band:
     0.10000 0.10000 0.10000
3.  Smeared DOS/LDOS: DISABLED
7.  Number of groups of bands:  ...  undefined ...
8.  Adopt present setting and calculate DOS/LDOS
11.  Preview the current DOS/LDOS
12.  PostScript file for the current DOS/LDOS
13.  Upper limit to the energy-range of the plot:  DISABLED
14.  Quit:  do not proceed.
------> Choose the item and press ENTER:
```

The number of groups of bands is chosen in **7**. In the calculated plots the DOS curves associated with each group will be shown by a different colour on the same plot. This is useful if the states responsible for certain features in the DOS need to be known.

If, say, out of all 8 bands three groups are chosen, namely 1-4, 5-6 and 7-8, then ***lev00*** will first show (just before updating the menu) how the energy is split between the groups:

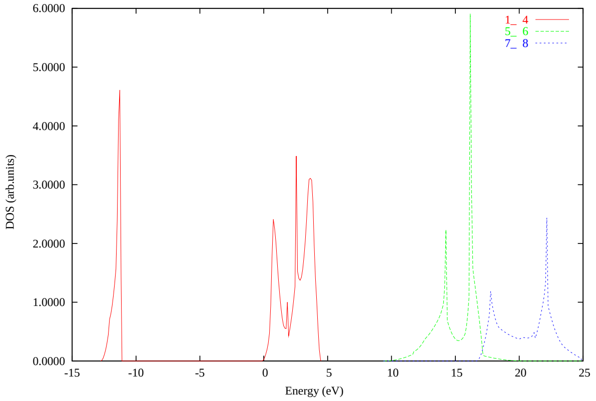


Figure 4.2: Total DOS of the MgO bulk: three groups of bands are shown by different colours.

```
...  The group 1 covers bands from 1 - 4
...  The group 2 covers bands from 5 - 6
...  The group 3 covers bands from 7 - 8
..........< Group 1 >...........
Energies E are between -.127706E+02 and 0.449350E+01 eV
..........< Group 2 >...........
Energies E are between 0.936580E+01 and 0.203830E+02 eV
..........< Group 3 >...........
Energies E are between 0.166957E+02 and 0.248751E+02 eV
```

The menu will then be updated with the option **7** changing into:

```
7.  Number of groups of bands:  3
> The following groups of bands are defined:
{1} 1_ 4; {2} 5_ 6; {3} 7_ 8;
> Files for the plot:  dos.dat1_1 , ...  , dos.dat3_1
> Names of the PostScript files:  dos.dat1_1.ps , ...  , dos.dat3_1.ps
```

It now shows explicitly all the file names to be produced. After calculating the DOS (option **8**), the DOS curves for each group can be seen in different colours (option **11**) as shown in Fig. 4.2.

Note that if the smearing is enabled (either slow or fast, option **3**, see Section 4.4), all calculations will be done using a single values for $\sigma$ and $\gamma$. Either original (raw) or smeared DOS can be previewed by controlling the option **6**. The calculation is performed using option **88**. If the smearing is on *before* option **8** is used, the calculation with smearing will also be performed automatically. Otherwise, it is necessary to apply option **88** directly.

Other options in the menu are similar or identical to those in the dispersion menu of Section 4.6.2.2, so that we will not consider them here.

#### 4.6.2.4 1,2,11 - Projected (local) DOS calculations

The calculation of the LDOS are accomplished by the options **11**, **1** and **2** of the main menu of ***lev00***. If the corresponding files with weights (see Section 4.3) are found and read in successfully, you will be prompted to the same sort of menu as described above for the total DOS:

```
.............MENU DOS/LDOS ......................
.....  Change these parameters if necessary:...
_____________> LDOS + total DOS <______________
0.  THE CURRENT PICTURE IS: 1
1.  Current LDOS task to be studied:  ...  undefined ...
2.  Maximal energy step for the plotting:  0.10000
   > Number of points for the DOS plot:  220
   > Steps for each "physical" band:
     0.10000 0.10000 0.10000
3.  Smeared DOS/LDOS: ENABLED fast method
4.  Broaden band boundaries by 1.500 dispersions
5.  Dispersion for the smearing:  0.30000
6.  For previewing:  original DOS/LDOS
7.  Number of plots in the group:  ...  undefined ...
8.  Adopt present setting and calculate DOS/LDOS
11.  Preview the current DOS/LDOS
12.  PostScript file for the current DOS/LDOS
13.  Upper limit to the energy-range of the plot:  DISABLED
14.  Quit:  do not proceed.
------> Choose the item and press ENTER:
```

Options (**0**, **2-14**) are exactly the same as in the previous Section 4.6.2.3. In particular, options **4**, **5** and **6** are shown only if the smearing (option **3**) is enabled.

The only new option that appear here is **1**. This requires some explanation. In the LDOS calculations you can specify several spheres (or layers when option **2** of the main menu, Section 4.6.1 is chosen). In addition, there is also a possibility to calculate $s$-, $p$- and $d$-projected LDOS for every sphere as well as the total local DOS. For each such a case we have a different factor $A_{n\mathbf{k}}$ calculated either by ***lev1/lev2*** (routines working with ***CASTEP*** - not anymore supported) or by ***VASP***.

Every such type of the LDOS is referred to as a "LDOS task" and is to be considered separately. Tasks are numbered and you have to specify a task number using a table which appears as you choose the menu item, e.g.:

```
_____________> LDOS + total DOS <______________
0.  THE CURRENT PICTURE IS: 1
1.  Current LDOS task to be studied:  ...  undefined ...
2.  Maximal energy step for the plotting:  0.10000
> Number of points for the DOS plot:  220
> Steps for each "physical" band:
0.10000 0.10000 0.10000
3.  Smeared DOS/LDOS: ENABLED fast method
4.  Broaden band boundaries by 1.500 dispersions
5.  Dispersion for the smearing:  0.30000
6.  For previewing:  original DOS/LDOS
7.  Number of plots in the group:  ...  undefined ...
8.  Adopt present setting and calculate DOS/LDOS
11.  Preview the current DOS/LDOS
12.  PostScript file for the current DOS/LDOS
13.  Upper limit to the energy-range of the plot:  DISABLED
14.  Quit:  do not proceed.
------> Choose the item and press ENTER:
1
......  You have the following tasks:  ......
1 task {t} with Radius= 1.00 and center in (0.000 0.000 0.000 ) from_vasp
2 task {s} with Radius= 1.00 and center in (0.000 0.000 0.000 ) from_vasp
3 task {p} with Radius= 1.00 and center in (0.000 0.000 0.000 ) from_vasp
4 task {d} with Radius= 1.00 and center in (0.000 0.000 0.000 ) from_vasp
5 task {t} with Radius= 1.00 and center in (2.100 2.100 2.100 ) from_vasp
6 task {s} with Radius= 1.00 and center in (2.100 2.100 2.100 ) from_vasp
7 task {p} with Radius= 1.00 and center in (2.100 2.100 2.100 ) from_vasp
8 task {d} with Radius= 1.00 and center in (2.100 2.100 2.100 ) from_vasp
```

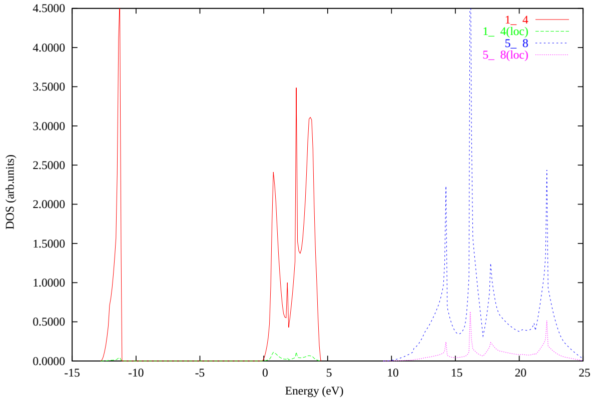


Figure 4.3: Total and local DOS for the MgO bulk (raw curves).Different groups of bands are shown separately.

```
    Choose one between 1 and 8 :
```

In this table, some useful additional information is shown, e.g. radius and position of the sphere and (in curly brackets) whether this is a total local DOS (`{t}`) or the *s*-, *p*-, *d*-projections (`{s}`, `{p}`, `{d}`). This information is provided to help you to choose the task correctly.

Now, choose the LDOS task, e.g. number 4; the option **1** will then transform into:

```
    1.  Current LDOS task to be studied:  4
       > Type of the LDOS: d
       > Radius of the sphere:  1.000
       > Center of the sphere:  ( 0.000, 0.000, 0.000)
       > Method of projection:  from_vasp
```

The data files produced here (options **8** and **88**, the latter one is on if the smearing is enabled) contain both the total DOS and LDOS columns. If the smearing is enabled, then the data files contain both smeared and raw total DOS and LDOS.

When previewed (option **11**), both total and local/projected DOS for every curve in the group (from option **7**) will be shown, as shown in Fig. 4.3. If smearing is on (check option **3**), smeared curves will be shown instead if these are chosen in option **6**, see Fig. 4.4.

Use option **13** to make the local/projected DOS visible since the total DOS usually appears too high on the graph.

## 4.6.3 Studying the partial or total charge/spin density

### 4.6.3.1 General remarks

Options **D** and (in the case of a spin-polarised calculation) **SD** work similarly. The difference is only in physical meaning of the objects ***lev00*** is dealing with:

- **D** - total (all occupied bands) or partial electron density or any other file (e.g. an electrostatic potential) of the same format

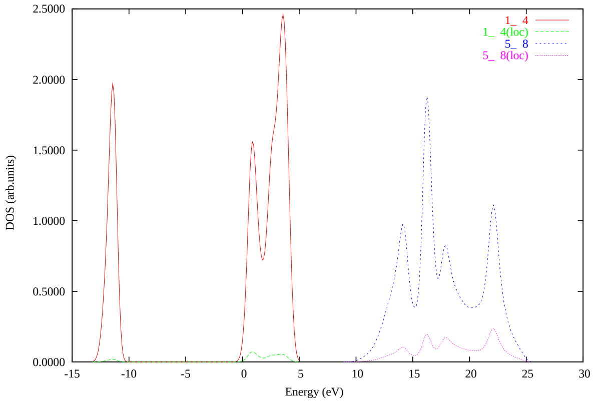


Figure 4.4: Total and local DOS for the MgO bulk shown in Fig. 4.3, that were smeared by the Gaussian with $\sigma$ =0.3 eV.

- **SD** - total spin density; equivalent to **D**, but reads in specifically the spin density (which is usually contained at the end of the full density file, at least, in the cases of ***VASP*** and ***SIESTA***).

In fact, in all these cases the structure of the files is the same or similar (for the given DFT code). Moreover, files of identical structure (if named appropriately) can also be read in and analysed, see Sections 5 and 6. For convenience, in the forthcoming Sections we shall be talking about density.

We shall first consider option **D** in detail. After that, percularities of options **4** (Section **??**) and **SD** (Section 4.6.3.14) will also be mentioned.

When activating option **D**, the file name to be read in is implicit for ***SIESTA***: job name appended by `.RHO`. In the case of ***VASP***, three possibilities exist: `CHGCAR` (total density), `PARCHG` (partial density), `LOCPOT` (electrostatic potential). The formats of these three files are similar. Then the file is read in:

```
Reading in charge density, please wait ...
it is done 10 %
it is done 20 %
it is done 30 %
it is done 40 %
it is done 50 %
it is done 60 %
it is done 70 %
it is done 80 %
it is done 90 %
it is done 100 %
Done!
The charge is = 0.800338E+01
.....> Total charge = 8.00337610269848 <.....
Done!
```

It shows the total number of electrons in the system[4] (8 in the example above). Then, the main menu of this option is opened up:

```
......  Choose between line or plane ......
NUMBER OF THE CURRENT PLOT: 1
```

[4] Of course, this only makes sense for the electron density.

```
1.  Plot density along a line
2.  Plot density in a plane
3.  Amount of charge inside a sphere
4.  Exploration of the density
5.  Dipole moment
6.  Quadrupole moment
7.  Get density via point charges:  match moments & potential
8.  Cutting atoms out of the density
9.  Write non-zero density as CHGCAR.new
10.  Write density in gOpenMol cube format
11.  Get density via point charges:  match density
12.  Transform the charge density for a shifted system
12.  Return to the previous menu
```

The options of this menu are explained in some detail in the Sections below. At this point, note the *plot number* in the first line. This is automatically updated every time you return to this menu. This number is used in file names created in various options and is necessary to avoid creating files with identical names.

#### 4.6.3.2 1 - Plot density along a line

This option is used to produce a 1D plot of the density along some direction is space. The line menu that is opened is this:

```
.............  LINE MENU ......................
......  Change these parameters if necessary:....
1.  Starting point (Angstroms):  ( 0.00000, 0.00000, 0.00000)
   Starting point (fractional):  ( 0.00000, 0.00000, 0.00000)
2.  Vector along the line:  .......  undefined .......
3.  Lenght along the line (in Angstroms):  0.00000
4.  Parameters for the plotting
5.  Perform calculation for the plotting:  file out.dat_1
6.  Preview the density
7.  Create a postscript file out.dat_1.ps for the plot
------- G e n e r a l   s e t t i n g s ---------
An.  Coordinates are specified in:  <Angstroms>
Co.  Show current atomic positions in fractional/Cartesian
------ L e a v e   t h e   m e n u ------------
Q. Return to the previous menu
------> Choose the item and press ENTER:
```

You will have to specify the starting point (option **1**), direction (option **2**) and the length along this direction (option **3**). Calculation along the line is performed in optin **5** (the file name produced is shown explicitly), while the results of the calculation can be previewed using option **6** and the Postscript file (with the name shown) is produced in option **7**. Note that the density is available on a 3D grid, therefore, an interpolation by 8 points is used when performing the calculation.

In "General settings" you can specify how the coordinates are to be given (option **An**), and also you can see all atomic positions (option **Co**). These two are analogous to those in ***tetr*** (Section 3.3). If **An** is set to `<AtomNumber>`, then the length (option **3**) is specified automatically when option **2** is done.

There is also an additional set of options available in **4**:

```
.............CUSTOMISATION .....................
.....  Change these parameters if necessary:...
1.  Resolution for the plot = 160
2.  Low chop = 0.00 {disabled}
3.  High chop = 0.00 {disabled}
4.  Multiplication factor = 1.00
```

```
5.  Return to the previous menu.
------> Choose the item and press ENTER:
```

Here it is possible to choose the number of points along the line (option **1**), low and high chops (**2** and **3**) and the multiplication factor (option **4**). The chops, if specified, will fix the density at the chop value if the density is higher ("High chop") or lower ("Low chop") than the chop value. The high chop is useful if it is necessary to see the details of the density in the regions of small density (e.g. between atoms), while the low chop may be useful for the spin density.

The file names created in this menu are `out.dat_M` and `out.dat_M.ps`, where `M` is the plot number of the parent (density) menu, see the bottom of Section 4.6.3.1.

#### 4.6.3.3 2 - Plot density in a plane

It is used to produce a 2D density plot. In this case a plane menu opens:

```
.............MENU for PLANE ......................
........  Change these parameters if necessary:......
/ 1.  Normal vector to the plane:  ( 0.00000, 0.00000, 1.00000)
| X1 vector in the plane:  ( 0.00000, -1.00000, 0.00000)
| Y1 vector in the plane:  ( 1.00000, 0.00000, 0.00000)
\\ 2.  The plane has been specified by 3 points:  NO
4.  Central point on the plane:
   in Angstroms => ( 0.00000, 0.00000, 0.00000)
   in fractional => ( 0.00000, 0.00000, 0.00000)
44.  Move the plane parallel to the existing one
5.  Width along X1 axis (Angstroms):  5.00000
6.  Width along Y1 axis (Angstroms):  5.00000
7.  Parameters for the plotting
8.  Preview the density
9.  Perform calculation for the plotting:  file out.dat_2
------- G e n e r a l   s e t t i n g s ---------
An.  Coordinates are specified in:  <AtomNumber>
Co.  Show current atomic positions in fractional/Cartesian
Ap.  Show atomic positions with respect to the plane
------ L e a v e   t h e   m e n u ------------
Q. Return to the previous menu
------> Choose the item and press ENTER:
```

To start, first specify the plane. This means: the plane position (its normal), the central point and the two widths (along the two axes within the plane). It can be done in two ways:

- by specifying the normal vector to the plane in **1** (this will fix the two axes within the plane as well[5]), the central point in **4** (this will fix the particular position of the plane in space, i.e. the points which it will go through) and the two widths in **5** and **6**;

- by specifying three points in **2** (works only if the number of atoms in the cell is more than 2); in this case, a new option **3** appears that suggests an additional method of choosing the central point as the centre of the triangle formed by the points as the plane centre; note, that option **4** disappears if option **3** was used;

[5] No adjastment is possible here.

to enable option **4** (e.g. if you want to use this method), use **3** again and the option **4** will reappear[6]; the widths are specified by **5** and **6** as before.

The second method is very convenient as it also shows positions of the three atoms (used to specify the plane) using coordinates along axes `X1` and `Y1` in the plane, e.g.

```
3.  Central point => the center of the triangle:  YES
   Central point on the plane:
   in Angstroms  => ( 0.00000, 0.00000, 0.66667)
   in fractional => ( 0.00000, 0.00000, 0.13333)
   The reference points A,B,C in (X1,Y1) are given as:
      A = ( 0.00000,-0.66667)
      B = (-1.00000, 0.33333)
      C = ( 1.00000, 0.33333)
```

However, even if the first method was chosen, there is still a way to find where your atoms are: the option **Ap** shows atomic positions with respect to the chosen plane in the in-plane coordinates; if an atom is out of the plane, its distance to the plane is also shown. For instance, in the example below the plane was chosen parallel to the water molecule plane, 0.5 Å away from it:

```
....  Species < O> ......> 1
Tot # |---------- Cartesian ----------| X-plane Y-plane out-of-plane
1   1     0.00000 0.00000 -0.06633      0.76138 -0.59276 -0.50000
....  Species < H> ......> 2
Tot # |---------- Cartesian ----------| X-plane Y-plane out-of-plane
2   1     0.76138 0.00000 0.52643      0.00000 0.00000 -0.50000
3   2    -0.76138 0.00000 0.52643      1.52276 0.00000 -0.50000
Hit ENTER when done ...
```

You can also see atomic positions with respect to the Cartesian system in **Co**, while the way coordinates are specified can be changed in **An**.

In addition, option **44** allows you to move the plane parallel to the existing one. This might be useful if parallel slices through the system are to be taken previewing density every time.

Once the plane has been chosen, you can preview the density (option **8**) and print out the density within the plane in **9** using a fine grid (given in option **7** submenu, see below). In the latter case, the file produced (its name is shown under that menu option) can be used in a plotting software to produce high quality graphs.

Similar to the line menu of Section 4.6.3.2, additional functionality is provided by option **7** that opens a submenu:

```
..............MENU ONE .......................
.....  Change these parameters if necessary:...
0.  Number of contour levels = 30
1.  Resolution for the plot = 160
2.  Resolution for the previewing = 30
3.  Type of the preview picture is:  contour
4.  Low chop = 0.00 {disabled}
5.  High chop = 0.00 {disabled}
6.  Multiplication factor 1.00
7.  Return to the previous menu.
```

[6] If option **4** is used here, there is always a risk to give a point which is out of the plane. In this case, a projection to the plane will be accepted as the central point.

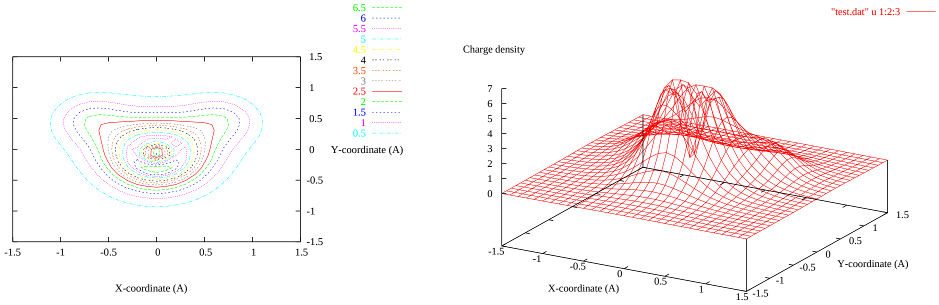


Figure 4.5: 2D (left) and 3D (right) electron density plots of a water molecule. The plane is 3 by 3 Å across.

```
------> Choose the item and press ENTER:
```

Here the number of contour levels for either of the plots (the preview and the high quality ones) is provided in option **0**; the grid (the resolution) for the preview plot is specified in **2**, while for the high quality file - in **1**. The option **3** allows you to choose between 2D and 3D type of the plot, while options **4** and **5** are used to specify the chops and **7** the multiplication factor as in Section 4.6.3.2.

The 2D and 3D density plots for a water molecule are shown in Fig. 4.5 as an example.

Names of files produced within this option are the same as in the case of 1D plots, see Section 4.6.3.2.

#### 4.6.3.4 3 - Amount of charge inside a sphere

Here it is possible to integrate the density over one or several spheres of different radii. The point is that the notion of *atomic charge* is no well defined. To get around it, e.g. in order to analyse charge transfer during atomic adsorption, one can calculate the charge of the atom in spheres of *several* different radii for both situations (before and after adsorption) and then compare the curves.

Here it is possible to plot one curve at a time. The comparison can be made separately. You should keel track of the data files produced, may be rename them for convenience, and then plot them together using other means. Note that, if you intend to use ***GNUPLOT***, you may benefit from the `test.gnu` file, produced by ***tetr*** for the plotting, and edit it appropriately.

The charge menu looks like this:

```
..............MENU for CHARGE ........................
.........  Change these parameters if necessary:.......
2.  The number of (possibly) overlaping spheres:  1
3.  The center(s) of your sphere(s):
   [1] A=> ( 0.000, 0.000, 0.000), fr=> ( 0.000, 0.000, 0.000)
4.  The smallest radius (Angstroms):  0.00000
5.  The largest radius (Angstroms):  0.00000
6.  The number of points between these radii:  ...  undefined ...
```

```
7.  Algorithm for the charge integration:  <conserving>
8.  X,Y,Z integration grid inside the sphere:  IRRELEVANT
9.  Perform calculation for the plotting:  file out.dat_2
10.  Preview the dependence of charge versus Radius
11.  Create a PostScript file out.dat_2.ps
------- G e n e r a l   s e t t i n g s ---------
An.  Coordinates are specified in:  <Angstroms>
Ne.  Representation of results:  through number of electrons
Co.  Show current atomic positions in fractional/Cartesian
------ L e a v e   t h e   m e n u -------------
Q. Return to the previous menu
------> Choose the item and press ENTER:
```

The number of spheres (up to five[7]) is specified in **2**, their centres in **3**, the range of radii in **4** and **5**, and the number of points between them - in **6**. The integration alsgorithm (see Section 4.5) is chosen in **7** and **8** (the latter is about the fine grid to be used and is only needed for the *nonconserving* method). The calculation is performed in **9** with the data file name as shown, and can be previewed in **10**; the Postscript file (as indicated) is produced in **11**. The three setting options allow you to choose how points in space are specified (**An**) and in which units (via the number of electrons or as a % of the total electron charge in the cell) the integrated charge is shown (**Ne**); in addition, you can preview atomic positions using **Co**.

If there are more than one sphere, the menu changes. In the example below three spheres were chosen:

```
..............MENU for CHARGE ........................
.........  Change these parameters if necessary:.......
2.  The number of (possibly) overlaping spheres:  3
3.  The center(s) of your sphere(s):
  [1] A=> ( 0.000, 0.000, -0.066), fr=> ( 0.000, 0.000, -0.013)
  [2] A=> ( 0.761, 0.000, 0.526), fr=> ( 0.152, 0.000, 0.105)
  [3] A=> ( -0.761, 0.000, 0.526), fr=> ( -0.152, 0.000, 0.105)
4.  The smallest radius (Angstroms):  0.00000
5.  The largest radius (Angstroms):  2.00000
55.  Shrinking factors:  1.000 1.100 0.900
6.  The number of points between these radii:  10
7.  Algorithm for the charge integration:  <nonconserv>
8.  X,Y,Z integration grid inside the sphere:  30
9.  Perform calculation for the plotting:  file out.dat_2
10.  Preview the dependence of charge versus Radius
11.  Create a PostScript file out.dat_2.ps
------- G e n e r a l s e t t i n g s ---------
An.  Coordinates are specified in:  <AtomNumber>
Ne.  Representation of results:  through number of electrons
Co.  Show current atomic positions in fractional/Cartesian
------ L e a v e t h e m e n u -------------
Q. Return to the previous menu
------> Choose the item and press ENTER:
```

To control the radii of different spheres in this case, you can specify "`Shrinking factors`" for every sphere in option **55** that opens if the number of spheres is larger than 1. To be precise, the radius of the sphere $i$ is calculated as $R_i = s_i R$, where $R$ changes from the minimum (in **4**) to the maximum (in **5**) values with the step calculated from **6**, whereas $s_i$ is the shrinking factor for this sphere. At first, spheres may not overlap, but, as the radius $R$ grows, they may. In this case the charge is counted only

[7] Can be changed in `menu.inc`, the variable `Num_Sph`.

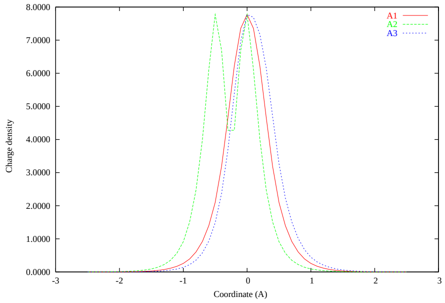


Figure 4.6: The 1D plots of the electron densities of a water molecule shot through the grid point of the highest density along the basic lattice vectors directions.

once in the overlapping regions. Note that the plot is shown as the charge versus the common radius $R$.

There is also one more facility that might be sometimes useful. When the calculation has been performed by **9**, two new options appear:

```
  9.  Perform calculation for the plotting:  file out.dat_2 <= DONE!
     98.  Number of electrons to fit the sphere in:  1.00000
     99.  Obtain the radius for the given # of electrons:
```

It is a "*charge calculator*": you can choose the number of electrons in **98** and then obtain (by **99**) the corresponding radius of the sphere that fits exactly this amount. This is done by linearly interpolating the calculated curve. In the case of many spheres, the common radius $R$ will be given.

The file names produced within this option are the same as in the case of 1D, Section 4.6.3.2, and 2D, Section 4.6.3.3, plots. The number in the files names correspond to the "`NUMBER OF THE CURRENT PLOT:`" in the parent menu, Section 4.6.3.1.

#### 4.6.3.5 4 - Exploration of the density

This is an interactive (no menus) option that allows one to find where the density is mostly concentrated in the cell and in what extent. When you run it, a special "exploration" routine is retrieved. It scans the density inside the cell, finds the grid point with the highest density and shows its Cartesian and fractional coordinates. Then, three 1D plots of the density along the cell basic lattice vectors $\mathbf{a}_i$ and passing through the grid point are calculated and plotted as shown in Fig. 4.6.

These peak-shaped curves allow you to estimate the width of each peak. This is needed as the next thing you will be asked to do is to specify the integration radius: the charge inside the sphere with that grid point as the centre will be calculated and shown.

Then, the charge inside the sphere will be "cut off"[8], and the procedure can be repeated for the rest of the cell as many times as you wish. This way the whole cell

[8] This will not affect the actual density read in by ***lev00*** and is only used within this option of the parent menu. After leaving this option, the same density as read in from the density file, will be available to other routines.

can be represented by lumps of densities in spheres. The result will be dumped in a file `explor.dat`. An example of the workout of the routine:

```
.......> Final distribution of the density:
Peak Hight _________(x,y,z)_________  Charge(e) Radius
1 7.77606 ( 0.00000, 0.25000, 4.83333) 5.88374 1.00000
2 2.09367 ( 0.66667, 4.91667, 0.50000) 0.24454 0.50000
```

#### 4.6.3.6 5 - Dipole moment

Calculates the dipole moment of the density in the cell. In principle, this is only meaningful in the cases of *molecules* contained inside big cells. The following menu opens:

```
.............MENU for DIPOLE .......................
.........  Change these parameters if necessary:.......
2.  Charges of nucleii by species:
   6 1
3.  The center of your sphere:
   A=> ( 0.000, 0.000, 0.000), fr=> ( 0.000, 0.000, 0.000)
4.  The smallest radius (Angstroms):  5.00000
5.  The largest radius (Angstroms):  10.00000
6.  The number of points between these radii:  10
9.  Calculate dipole moment; the file for plotting is out.dat_1
10.  Preview the dependence of dipole moment versus Radius
11.  Create a PostScript file out.dat_1.ps
------- G e n e r a l   s e t t i n g s ---------
0.  Coordinates are specified in:  <AtomNumber>
7.  Algorithm for the charge integration is FIXED to:  <nonconserv>
8.  X,Y,Z integration grid inside the sphere:  30
1.  Units:  electrons*Angstrem
------ L e a v e   t h e   m e n u -------------
12.  Return to the previous menu
------> Choose the item and press ENTER:
```

The menu is in a way similar to that of the charge integration considered in Section 4.6.3.4. Firstly, charges on nuclei are given in **2**. Then, a sphere is chosen which has the whole molecule inside it with the centre (option **3**) somewhere nearby the molecule centre. Note that both the starting (option **4**) and the ending (option **5**) radii of the sphere must include the *whole* molecule. The dipole moment in units of option **1** is calculated in **9** and previewed in **10** as a function of the sphere radius. A reasonable estimate can be obtained from a plato region in the graph.

Note that the integration method is fixed to *nonconservative* here.

#### 4.6.3.7 6 - Quadrupole moment

Calculates the second multipole moment of the density. This is also meant for molecules inside a big cell. The first menu looks like this:

```
.............MENU for QUADRUPOLE ...................
.........  Change these parameters if necessary:.......
1.  Charges of nucleii by species:
   6 1
2.  The center of your sphere:
   A=> ( 0.000, 0.000, 0.000), fr=> ( 0.000, 0.000, 0.000)
3.  The sphere radius (Angstroms):  5.00000
6.  Calculate quadrupole tensor.
------- G e n e r a l   s e t t i n g s ---------
0.  Coordinates are specified in:  <Angstroms>
4.  Algorithm for charge integration is FIXED to:  <nonconserv>
```

```
5.  X,Y,Z integration grid inside the sphere:  40
------ L e a v e   t h e   m e n u ------------
8.  Return to the previous menu
------> Choose the item and press ENTER:
```

As in the dipole menu (Section 4.6.3.6), the nuclei charges and the sphere centre are specified. Only one radius is chosen, this can be selected e.g. from the dipole calculation. Once the calculation is done (option **6**), it is possible to obtain principal axes of the tensor in a new option **7** as shown below:

```
.............MENU for QUADRUPOLE ...................
.........  Change these parameters if necessary:.......
1.  Charges of nucleii by species:
   6 1
2.  The center of your sphere:
   A=> ( 0.000, 0.000, 0.000), fr=> ( 0.000, 0.000, 0.000)
3.  The sphere radius (Angstroms):  5.00000
6.  Calculate quadrupole tensor.  <= DONE!
   Quadrupole tensor is [in Electron*Angstrem^2]:
      14.55748 0.00000 3.52430
      0.00000 -31.66499 0.00000
      3.52430 0.00000 17.10751
7.  Calculate principal axes of the quadrupole tensor
   Eigenvalues:  -31.66499 12.08465 19.58034
   Eigenvectors:  0.00000 0.81860 0.57437
                 1.00000 0.00000 0.00000
                 0.00000 -0.57437 0.81860
------- G e n e r a l   s e t t i n g s ---------
0.  Coordinates are specified in:  <Angstroms>
4.  Algorithm for charge integration is FIXED to:  <nonconserv>
5.  X,Y,Z integration grid inside the sphere:  40
------ L e a v e   t h e   m e n u ------------
8.  Return to the previous menu
------> Choose the item and press ENTER:
```

#### 4.6.3.8 7 - Get density via point charges: match moments & potential

This option works only for ***CASTEP***[9] and will only appear if both dipole and quadropule calculations have been performed. The total electronic density is represented by a collection of point charges by matching the exact electrostatic potential (from the PW run) and the calculated (from the previous menu option) dipole and quadrupole moments.

Then, the following menu appears:

```
.............MENU for SIMULATE .....................
.........  Change these parameters if necessary:.......
>>>> Dipole moment (e*A): -40.00000 -40.00000 -41.12812
>>>> Quadrupole moment tensor (e*A*A):
     14.55748 0.00000 3.52430
     0.00000 -31.66499 0.00000
     3.52430 0.00000 17.10751
1.  Precision of the summations:  0.100000E-09
2.  The center of your box:
   A=> ( 0.000, 0.000, 0.000), fr=> ( 0.000, 0.000, 0.000)
3.  Directions of the box sides are:  undefined
4.  Lengthes of the sides of the box are:  undefined
5.  Grid for the simulating charge:  5 5 5
   Total number of simulating charges = 27
```

[9] In fact, it should work for any code if the file with the potential is written in the correct format, see file `TOOLS/simulate.f`

```
6.  Read the exact potential on the grid <=== undefined!
8.  We fit to every 5-th grid point in the potential
9.  Calculate simulating charges and put them in the file out.dat_1
------- G e n e r a l    s e t t i n g s ---------
An.  Coordinates are specified in:  <Angstroms>
Co.  Show current atomic positions in fractional/Cartesian
------ L e a v e    t h e    m e n u -------------
Q. Return to the previous menu
------> Choose the item and press ENTER:
```

At the top, the calculated dipole and quadrupole moments are shown. What is done is this: the charge density in the cell is simulated by a collection of point charges which are positioned at the fine grid to be chosen in the menu (options **5**) within a box also to be specified (options **2**-**4**). The box should contain the whole molecule. The charges are chosen to match the exact potential in the cell region which is *outside* the molecule (i.e. outside the box). The exact potential is be read in separately in option **6**. Then, the positions and the values of charges that match both the potential and the dipole and quadrupole moments of the molecule are witten to the output file shown in **9**.

#### 4.6.3.9 8 - Cutting atoms out of the density

Here, it is possible to modify the electron density on the grid[10] by taking away (setting to zero) the density at all grid points that are inside spheres build around specified atoms. This might be useful if, for instance, one is interested in some local region in the cell, e.g. a defect. In this case the density associated with the atoms around that region can be removed to isolate the density in the region of interest. Sometimes, some of the density inside that region (e.g. around specific atoms) can also be removed. This way it is easy, for example, to get the "pure" F-centre density localised in the vacancy which can be then (see Section ) simulated by point charges.

The spheres radii are calculated inside the code from the charges that to be cut off around each atom; the charges are specified by the user. The initial menu:

```
.............MENU for CUT ..........................
.........  Change these parameters if necessary:.......
> > > > > Representation of results:  through number of electrons
> > > > > Algorithm for the charge integration:  <nonconserving>
1.  Specify atoms/electrons to be cut out of the density:
2.  The smallest radius (Angstroms):  5.00000
3.  The largest radius (Angstroms):  10.00000
4.  The number of points between these radii:  ...  undefined ...
5.  X,Y,Z integration grid inside the sphere:  30
6.  Scan atoms to obtain the radii using charge to cut
------- A t o m i c    p o s i t i o n s ---------
Co.  Show current atomic positions in fractional/Cartesian
------ L e a v e    t h e    m e n u -------------
Q. Return to the previous menu
------> Choose the item and press ENTER:
```

In option **1** you can specify a range of atoms (e.g. 5-10) that carry the same charges and which should be removed. At the same time, you will be asked about the (positive) charge to be associated with the atoms. It is not possible to specify several lists of atoms. However, when using option **1** several times, it is possible to update and

[10] I.e. the density in the memory will be destroyed!

complement the existing list of atoms, and add other atoms with different charges. For example, in the case of a MgO cell, you can first enter a list of Mg atoms with nearly zero electron charge, and then (the second time) - the list of O atoms with the charge of, say, 7.8.

The next step is to specify the radii (options **2** and **3**) and the number of points (option **4**). This information will be used for interpolation of the charge around the atoms listed in option **1**. Then, perform the scan of all specified atoms in **6**, and the radii for each set of atoms of equal charge will be calculated. The menu grows:

```
.............MENU for CUT ..........................
.........  Change these parameters if necessary:.......
>>>>> Representation of results:  through number of electrons
>>>>> Algorithm for the charge integration:  <nonconserving>
1.  Atoms to be cut out of the density:
To be cut 2 atoms with numbers(electrons):
1( 0.10) 2( 0.50)
2.  The smallest radius (Angstroms):  0.10000
3.  The largest radius (Angstroms):  2.00000
4.  The number of points between these radii:  10
5.  X,Y,Z integration grid inside the sphere:  30
6.  Scan atoms to obtain the radii using charge to cut <= DONE!
7.  Show the list of atoms and their radii + exact charge to be cut out
8.  The threshhold:  the smallest density allowed:  0.000000E+00
9.  Cut atoms out (current density (in memory) is destroyed!)
------- A t o m i c p o s i t i o n s ---------
Co.  Show current atomic positions in fractional/Cartesian
------ L e a v e t h e m e n u -------------
Q. Return to the previous menu
------> Choose the item and press ENTER:
```

The calculated information can be previewed in **7**, e.g.

```
Please, wait, scanning the charge density again ...
>>>>>>>> Radii + charges for chosen atoms <<<<<<<<
# atom radius   charge(asked) charge(estimate) charge(actual)
1   1  0.76386 0.10000000      0.09044981         0.08987854
2   2  0.23717 0.50000000      0.30739785         0.30747416
Press ENTER when ready ...
```

Here one can see for each atom the sphere radius found by interpolation from the target charge, an estimate of the charge calculated using the nonconserving algorithm (Section 4.5) for the same radius, and, finally, the actual charge inside the sphere calculated using the conserving algorithm.

It is also possible to remove the density from the grid points where the density is smaller than some specified threshold in **8**.

Finally, option **9** sets all grid points of the density that are found inside either of the specified spheres to zero. After the calculation, the following information is displayed:

```
9.  Cut atoms out (current density (in memory) is destroyed!)
   Out of the total original density = 8.00338
   ...  grid points removed due to radii = 4870
   ...  with the density cut out = 0.39735
   ...  add.  grid points removed due to threshold = 0
   ...  with the density cut out = 0.000000E+00
   ...  so that the TOTAL density left is = 7.60602
10.  Write charges cut out + core charges into a file
```

Some information about the charges on removed atoms can be obtained and then saved in a file `charges.cut` in a new option **10**.

#### 4.6.3.10 9 - Write non-zero density as NAME.new

This very simple option will write the current density as either `CHGCAR.new` (***VASP***) or `job.RHO.new` (***SIESTA***) file. The format in the files of the corresponding codes is kept.

Note that if the charge was cut off from the density (Section 4.6.3.9), the new density (with zeros at grid points that were inside the chosen spheres) will be written.

#### 4.6.3.11 10 - Write density in gOpenMol cube format

This option allows one to save a density in a so-called `Cube` format that can be then read in by ***gOpenMol*** to do nice pictures of isodensity surfaces. For extra convenience, the density can only be saved that fits inside a box. The latter may run across the cells. In addition, atomic positions inside the same or bigger box can also be saved (in the `xyz` format), so that ***gOpenMol*** would be able to show atoms alongside the density surfaces.

The menu that opens is this:

```
...........MENU for gOpenMol writer ................
.........  Change these parameters if necessary:.......
1.  The box center is at:  ( 0.00000, 0.00000, 0.00000)
    Directions of the box sides are fixed along:
         1 < 1.00000, 0.00000, 0.00000 >
         2 < 0.00000, 1.00000, 0.00000 >
         3 < 0.00000, 0.00000, 1.00000 >
2.  Lengths of the box sides (in A) are:  1.00000 1.00000 1.00000
   >>>> the starting point (corner) of the box is at -0.50 -0.50 -0.50
3.  Box extension for xyz-file:  [ 0.00000 0.00000 0.00000]
4.  The grid in the box in 3 directions:  1 1 1
   >>>> total number of grid points = 1
   >>>> steps along 3 directions are 1.00000 1.00000 1.00000
5.  Scan the box and integrate the charge (reference only):
W. Write the cube-format file for gOpenMol
------- G e n e r a l   s e t t i n g s ---------
An.  Coordinates are specified in:  <Angstroms>
Co.  Show current atomic positions in fractional/Cartesian
cT. Centre of a triangle of points/atoms
mD. Middle distance between two points/atoms
XY. Atoms in the XYZ box:  constrained by the density box
------ L e a v e   t h e   m e n u -------------
Q. Return to the previous menu
------> Choose the item and press ENTER:
```

The box for the density is specified in options **1** and **2**, and its possible extension for the atomic positions - in **3** (this option can be disabled by **XY**). The grid inside the box for the density interpolation is given in **4**. Note that the directions of the box faces are fixed along the Cartesian axes.

To check how much electron charge is inside the box, run **5**. The actual calculation is performed in W, after which the two files are saved:

- `gOpenMol.cube` - the `Cube` file with the density on the grid inside the box

- `gOpenMol.xyz` - atomic poisitons inside the same or larger box

Under the settings there are usual options **An** (how coordinates are specified) and **Co** (show atomic positions). In addition to those, two options **cT** and **mD** serve as a mini-calculator that may be sometimes useful to find the best position for the box centre in **1**.

### A short note on how to use *gOpenMol*

What you need to do with gOpenMol:

1. read the `gOpenMol.xyz` file by going to: `File -> Import -> Coord -> Browse -> Apply`; you should see your system in the main graphical window as connections between atoms
2. read in the `gOpenMol.cube` file by going to: `Run -> gCube2plt/g94cub2pl (cube) -> Browse -> Apply` ; it will open a window and will produce an output in it finished by "`Job done ...`"; a file `gOpenMol.plt` will be produced
3. then plot the density by reading the `gOpenMol.plt` file: `Plot -> Contour -> Browse -> Import File`. In a new window openned, you will find the minimum and the maximum values of the density; choose the density value in the boxes underneath the min/max info and click `Apply`: the density will be shown alongside the atomic positions in the graphical window where you will be able to rotate it
4. To make the density transparent, click on `Details` opposite the chosen density level and change the `Opacity` level, then click `Apply`. It is possible to plot several density levels in the same way by filling in several density levels and then changing their opacities.
5. To see atoms better, use: `View -> Atom Type` and then the option `Licorice` with e.g. `Cyl.rad` = 0.1 and `Sph. rad.` = 0.3. To have different radii for different species, use `CPK` as well. See the ***gOpenMol*** manual for more instructions. ***gOpenMol*** is freely available at: `http://www.csc.fi/gopenmol/`
6. To save a file with the picture in the graphical window, use: File -> Hardcopy -> Apply

An example of the ***gOpenMol*** image is shown in Fig. 4.7. Two transparent surfaces are shown.

#### 4.6.3.12 11 - Get density via point charges: match density

This may be useful if a point charge representation of the density in some region is required. For instance, in the case of a charged cell (e.g. a charged molecule or defect), the Coulomb correction is needed to be subtracted from the energy to avoid the cell-size dependence, this can most easily be done using point charges.

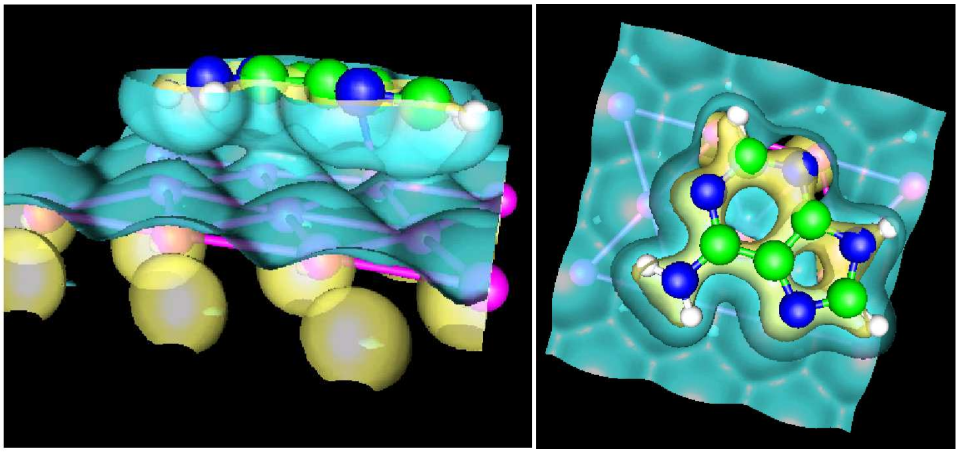


Figure 4.7: Surfaces of constant density for an adenine molecule on gold: side (left) and top (right) views.

Given the density, a box is chosen with a fine grid in it. Then, point charges are calculated by integrating the density in each parallelepiped of the grid. These charges give a reasonable representation of the density inside the box.

The menu when opens is this:

```
...........MENU for SIMULATE in the BOX ............
.........  Change these parameters if necessary:.......
>>>>> Representation of results:  through number of electrons
>>>>> Algorithm for the charge integration:  <nonconserving>
1.  The box center is at:  ( 0.00000, 0.00000, 0.00000)
2.  Directions of the box sides are along:
      1 1.00000, 0.00000, 0.00000)
      2 0.00000, 1.00000, 0.00000)
      3 0.00000, 0.00000, 1.00000)
3.  Lengths of the box sides (in A) are:  1.00000 1.00000 1.00000
>>>> corner of the box is at -0.50000 -0.50000 -0.50000
4.  The number of charges in each direction:  1 1 1
>>>> total number of charges = 1
>>>> the cell volume = 1.00000
5.  The integration grid in each cell of the box:  10 10 10
6.  Scan the box and integrate the charge (reference only):
7.  Get point charges using the grid specified
------- G e n e r a l   s e t t i n g s ---------
An.  Coordinates are specified in:  <Angstroms>
Co.  Show current atomic positions in fractional/Cartesian
------ L e a v e   t h e   m e n u -------------
Q. Return to the previous menu
------> Choose the item and press ENTER:
```

The box is psecified in options **1-3**. The grid inside the box (that will eventuallly give the total number of point charges) is specified in **4**. Each charge is positioned in the "centre of charge" in each parallellepiped of the grid. To calculate its position and the total charge, a fine grid is used inside each parallellepiped, this is given in **5**. To check if the two grids are reasonable, you can integrate the charge in **6**. This gives some information that might be useful to decide:

```
6.  Scan the box and integrate the charge (reference only):
>>>> total charge in the box = 8.00000
>>>> total grid points in the box = 216000
```

When this all done, apply **7** to calculate the point charges. A file `charges.sim` is written with all charges, and the menu changes into this:

```
7.  Get point charges using the grid specified <= DONE!
>>>> total number of charges = 110
>>>> total charge = 7.98368
>>>> charges between 0.00010 and 3.46360
8.  Show point charges
88.  Visualise point charges
9.  Number of unit cells to test the potential:  1
>>>> the test box defined as :  1x 1x 1
10.  Calculate the potential at a set of points
11.  Calculate the dipole moment
```

The view a table with all charges, use **8**. A better way to preview charges is to use **88** instead. In this case a file `charges.xyz` is written in the ***Xmol*** format with all charges, in which the species field of each charge is filled with one of the symbols `LV1`,...,`LV11`, the larger number correspond to a larger charge. If ***Xmol*** is properly configured, then small charges will be shown by smaller balls, while larger ones by larger balls.

The calculated point charges can now be used in simple claculations, for those new options **9-11** can be used. Option **9** gives the box $N \times N \times N$ of cells in which the potential is calculated by using **10**. As this is done, the menu option **10** changes into:

```
10.  Compare potential with the previous one
```

This means the following: you can now go and generate a new set of charges, e.g. by choosing a different number of them (a new grid). Then, by using **10**, you can find if the potential in the cells around the central one changed significantly. This can be used as a criteria for the proper representation of the charge density by the point charges generated. Finally, the dipole moment of the charge distribution in the box (with respect to the box centre) is calculated in **11**.

The settings options here are **Co** and **An** with their usual meaning of showing atoms and choosing the format for the coordinates.

#### 4.6.3.13 12 - Transform the charge density for a shifted system

It is possible that the ground state DFT calculation has been done with a position of the coordinate system corresponding to low symmetry. This is not good as the band-structure calculation would require more **k**-points. To get around this without repeating the ground state calculation, one should recalculate the density with respect to the "shifted" coordinate system.

This is done here. You will be asked about the position of the new origin of the Cartesian coordinate system (in Å). Then the density will be recalculated. Note that the old density will be destoryed. You can save the new density using option **9** of the density menu, Section 4.6.3.10 .

#### 4.6.3.14 Option SD of the density menu

Option **SD** only works with ***VASP*** and ***SIESTA***. The only difference between options **SD** and **D** of the main menu (Section 4.6.1), considered above, is that the density file is implied only[11], and the second part of it (containing the spin density) is read in, the first part (containig the total density) is skipped. After the spin density is read in, the same menu as in the case of option 5 is opened, so that you can refer to the previous Sections completely.

### 4.6.4 General linear combination of charge densities

For some applications it is useful to have a tool which could, for example, subtract one density from the other. This possibility is provided by option **M** of the main menu (both ***VASP*** and ***SIESTA*** are supported!). In fact, if two densities, $\rho_a$ and $\rho_b$ are given, it is possible to calculate a new density on the grid, $\rho_{ab}$, that is their linear combination:

$$\rho_c(i_1, i_2, i_3) = c_1\rho_a(i_1, i_2, i_3) + c_1\rho_b(i_1, i_2, i_3)$$

where $c_1$ and $c_2$ are arbitrary constants. It is necessary that the two files containing densities $\rho_a$ and $\rho_b$ be compatible with each other (the same lattice vectors and grid, atomic positions can be different).

The following menu opens:

```
.......  DENSITIES MANIPULATION MENU ................
...  CALCULATE the density from A,B-densities as:  ....
     final(grid) = 1.000 * A(grid) -1.000 * B(grid)
      <<< active code = VASP >>>
1.  file name of the A-density = CHGCAR_a
2.  file name of the B-density = CHGCAR_b
3.  file name of the final density = CHGCAR_ab
4.  coefficient c1= 1.00000000000000
5.  coefficient c1= -1.00000000000000
6.  calculate and write the final density
7.  quit
------------>
```

The file names in options **1-3** are initially suggested by the current DFT code, however, they can be changed in those options. The coefficients $c_1$and $c_2$ are specified in 4 and 5, while the actual calculation and saving of the density $\rho_{ab}$ to the file is performed in **6**. The sum of the density over all grid points (in units of electrons) is shown after the calculation for information, e.g.:

```
6.  calculate and write the final density <= DONE
.....> Total charge in CHGCAR = 8.00000219907003 <.....
.....> Total charge in 1 = 8.00000219907003 <.....
.....> Final charge in 2 = 0.00000000000000D+000 <.....
```

In this example two identical densities were used (both corresponding to 8 electrons) and $c_1 = -c_2 = 1$, so that their linear combination gives zero density.

[11] CHGCAR for ***VASP*** and the job name appended by .RHO for ***SIESTA***.

# Chapter 5

# Working with ***VASP***

In this Chapter we shall consider several issues not yet covered above which are concerned with running ***tetr*** and ***lev00*** with ***VASP***.

## 5.1 ***tetr*** and ***VASP***

First of all, the main usage of ***tetr*** is to produce the ***VASP*** input files like `CONTCAR` and `KPOINTS` as described in Section 3.6.

If the DOS option is used while generating **k**-points, the `brill.dat` file will be also produced that contains an important information needed for ***lev00*** to calculate the DOS (Section 3.6.6.2). Note that the `DOSCAR` file written by VASP does give the total DOS. However, using the route of ***lev00*** it is possible to have more functionality as described in Section 4.6.2. See Section 5.2.1.

Finally, the calculation of phonon frequencies with ***VASP*** is completely supported. ***tetr*** will take care of all input and output files as described in Section 3.6.8. No need for ***lev00*** in this case at all!

## 5.2 ***lev00*** and ***VASP***

Basically, the following functionality is supported at the moment:

- Total DOS
- Projected DOS and $s, p, d$-projected DOS (only on atoms of the system)
- Total electronic density
- Spin density
- Partial density
- Electrostatic potential

Actually, it covers nearly all of the ***lev00*** functionality.

### 5.2.1 What to do for the proper DOS/LDOS calculation ?

Here is a summary of what you must do:

1. First of all, if you want to get all the benefits of ***lev00*** for the DOS option, run ***tetr*** and create **k**-points for the DOS in the KP-menu, Section 3.6.6.2; save changes in the `KPOINTS` file (option **S**), the file `brill.dat` will be created automatically
2. Then run ***VASP*** with ICHARG=11 (to fix the density) in the `INCAR` file (the band-structure option)
3. Run ***lev00*** as explained in Section 4.6.2

In the case of the DOS calculation, you need only `OUTCAR` file after the ***VASP*** run. In the case of the $s, p, d$-projected DOS calculation (LDOS), you will also need the `PROCAR` file to be produced by ***VASP*** in the band-structure run if `RWIGS` are specified in the `INCAR` file. These parameters define the radii of the spheres for the LDOS calculation, see the ***VASP*** manual.

### 5.2.2 What to do for processing the electronic/spin/partial density or electrostatic potential?

For this calculation you can use the **k**-points of your ground state run. Therefore, follow this sequence:

1. Run ***VASP*** first:
   (a) Total/spin densities are produced (the file `CHGCAR`) provided that the LCHARG flag in the `INCAR` file is set to TRUE
   (b) To produce a partial charge density (the file `PARCHG`), specify `LPARD=.TRUE.` and `IBAND` with the range of bands to be included (e.g. `IBAND= 15 16 17 20`) in the `INCAR` file; see the ***VASP*** manual for more details
   (c) To work on the local electrostatic potential, the LOCPOT file should be produced by VASP; use `LVTOT = .TRUE.` in the `INCAR` file for this
2. Run ***lev00*** as explained in Section 4.6.3

Thus, you will need the following ***VASP*** output files:

- `CHGCAR` for the total density or the total spin density calculation
- `PARCHG` for the partial charge density
- `LOCPOT` for the electrostatic potential calculation

# Chapter 6

# Working with SIESTA

In this Chapter we shall consider several issues which are concerned with running ***tetr*** and ***lev00*** with ***SIESTA***.

## 6.1 *tetr* and *SIESTA*

First of all, the main usage of ***tetr*** is to produce the ***SIESTA*** input `<job>.fdf` file as described in Section 3.6. Here `<job>` is the name of the ***SIESTA*** job. ***tetr*** can process both `<job>.fdf` and `<job>.XV` files.

The DOS option is now supported for ***SIESTA*** as well, however, a patch to ***SIESTA*** 2.0 is required which can be obtained upon request by licence holders. To specify **k**-points directly, it is necessary to set up the block `BandPoints`, e.g.

```
%block BandPoints
0.000 0.000 0.000 # This is a comment.  eg this is gamma
1.000 0.000 0.000
0.500 0.500 0.500
%endblock BandPoints
```

The **k**-points are specified in the units corresponding to `BandLinesScale` option. Should be used only with option `SolutionMethod = diagon`. The band energies for each **k**-point are saved in the file `<job>.bands`. In the example above, **k**-points are specified via reciprocal lattice vectors (i.e. in fractional coordfinates). This is achieved by setting `BandLinesScale=ReciprocalLatticeVectors`. This is the option used by ***tetr*** to create **k**-points for the DOS option.

Finally, the calculation of phonon frequencies with ***SIESTA*** is completely supported. ***tetr*** will take care of all input and output files as described in Section 3.6.8. No need for ***lev00*** in this case at all!

## 6.2 *lev00* and *SIESTA*

Basically, the following functionality is supported at the moment:

- Total DOS and testing smearing for DOS

- Total electronic density
- Spin density

Actually, it covers nearly all of the ***lev00*** functionality related to densities.

### 6.2.1 What to do for the proper DOS/LDOS calculation ?

Here is a summary of what you must do:

1. First of all, if you want to get all the benefits of ***lev00*** for the DOS option, run ***tetr*** and create **k**-points for the DOS in the KP-menu, Section 3.6.6.2; save changes in the `<job>.fdf` file (option **S**), the file `brill.dat` will be created automatically.
2. Then run ***SIESTA*** with `SolutionMethod = diagon` in the `<job>.fdf` file
3. Run ***lev00*** as explained in Section 4.6.2

Projected DOS calculation is not yet supported.

### 6.2.2 What to do for processing the electronic/spin density?

For this calculation you can use the files of your ground state run. Therefore, follow this sequence:

1. Run ***SIESTA*** first; the total/spin densities are produced (the file `<job>.RHO`) provided that the `SaveRho` flag in the `<job>.fdf` file is set to TRUE
2. Run ***lev00*** as explained in Section 4.6.3

Thus, you will need the following ***SIESTA*** output files:

- `<job>.XV` file with the final geometry
- `<job>.RHO` file for the electron/spin density

# Chapter 7

# Miscellaneous

## 7.1 Limitations and parameters of the package

The package of computer programs considered in the previous sections has some preset parameters which are scattered over some files with the extension `.inc` (for ***tetr***) and in the corresponding modules (for ***lev00***). However, it is very unlikely that these will require any change. The meaning of all parameters is described in the corresponding files and do not need to be repeated here. If any of the codes gives an error message stating that some of the parameters are exceeded, go the corresponding `*.inc` file, change the parameter and recompile the code.

## 7.2 Bugs and comments

The author will be grateful for any comments concerning this manual and the package itself. Any bugs found should be reported to the author:
lev.kantorovitch@kcl.ac.uk
www.cmmp.ucl.ac.uk/˜lev

## 7.3 Enjoyed?

If you enjoyed using this package or, at least, found it useful, why don't you spread the word?

Also, if the usage of this package helped you to do your work, then, please refer to it in your publications as:

L. N. Kantorovich, *User-friendly visualisation program for ab initio DFT codes VASP* and *SIESTA*, 1996-2007, unpublished.

# Bibliography

[1] J. Lazewski, P. T. Jochym, K. Parlinski, and P. Piekarz, J. Mol. Structure **596**, 3 (2001).

[2] K. Parlinski, Z. Q. Li, and Y. Kawazoe, Phys. Rev. Lett. **78**, 4063 (1997).

[3] L. Kantorovich, *Quantum theory of the solid state: an introduction.*Fundamental Theories of Physics (Kluwer Academic Publishers, Dordrecht, 2004).