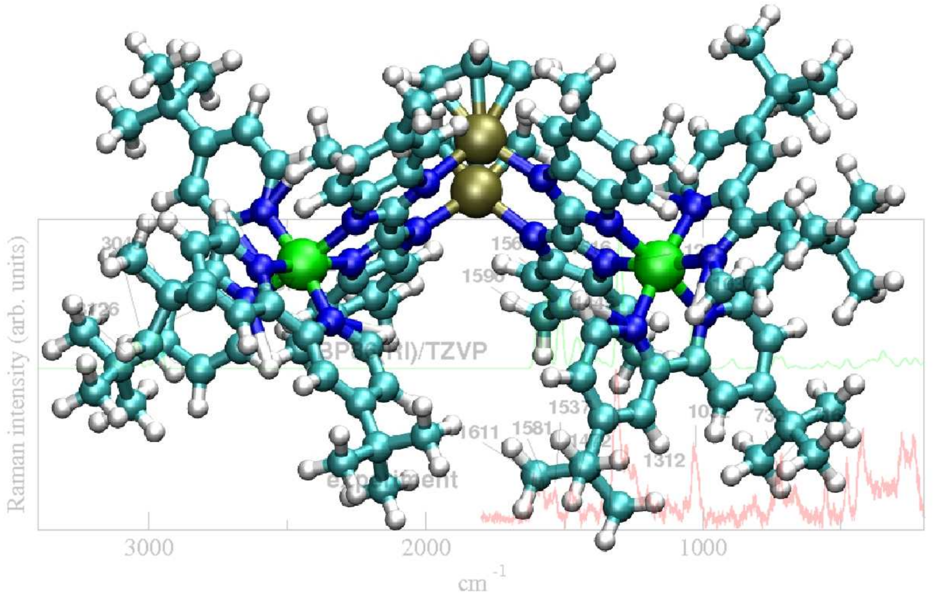
SNF

Quantum Chemical Calculation of Vibrational Spectra



written by

J. Neugebauer, C. Herrmann, and M. Reiher

Laboratorium für Physikalische Chemie[1], ETH Zurich,
Wolfgang-Pauli-Strasse 10, 8093 Zurich, Switzerland

Version 4.0.0
February 6th 2007

[1] email: markus.reiher@phys.chem.ethz.ch

Any use of this program that results in published material should **cite the following**:

J. Neugebauer, M. Reiher, C. Kind, B. A. Hess. *J. Comp. Chem.*, **23** (2002) 895 – 910.

The cover picture shows a tetranuclear [Ru_2Pd_2] complex, for which infrared, Raman, and resonance Raman spectra have been calculated with SNF. The calculation required 1632 single-point steps in total, which were carried out with the parallel raw data collector SNFDC.

Contents

1. Introduction

1.1 The program package Snf

The aim of this manual is to give a description of the installation, the usage, and the performance of the vibrational spectra calculations using the programs Snf, Snfdefine, and Snfdc. For the underlying theory we refer to the literature [1–5] This software is under constant development. Its present authors are: Johannes Neugebauer, Carmen Herrmann, Sandra Luber, and Markus Reiher.

The program package Snf is developed for the calculation of vibrational spectra. Infrared, Raman, and vibrational Raman optical activity (VROA) spectra can be obtained using the *harmonic* approximation for the frequencies and the *double harmonic* approximation for the intensities. Vibrational frequencies are determined using numerical differentiation of analytical gradients of the total electronic energy, while infrared, Raman and VROA intensities are obtained by numerical differentiation of dipole moments and (generalized) polarizability tensor components with respect to Cartesian nuclear coordinates.

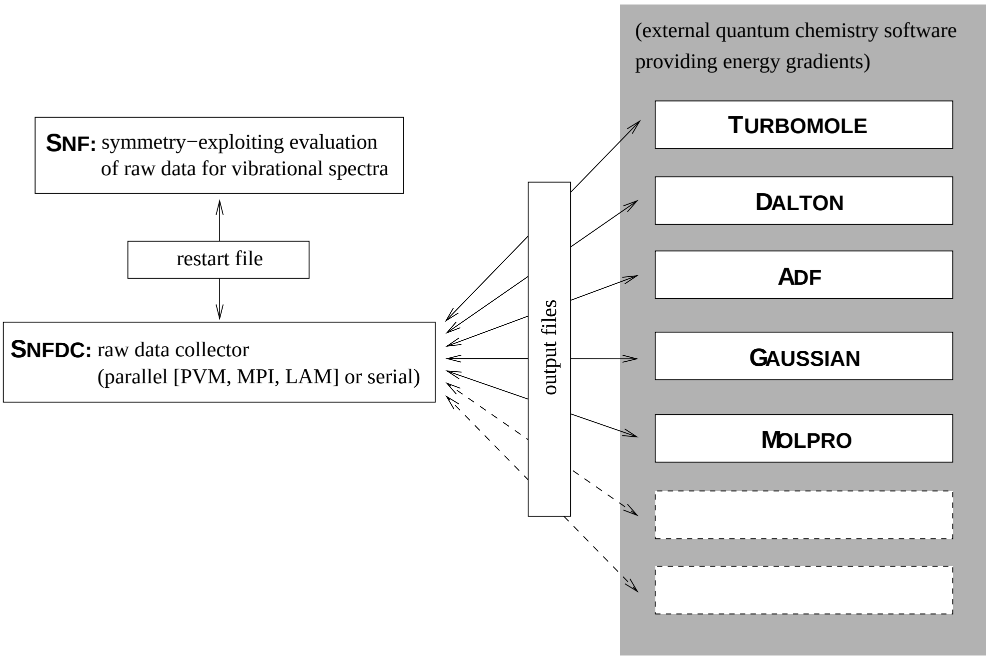
Snf takes maximum advantage of the molecular point group[1]. By using the data collector Snfdc, all required single point calculations can be performed using *coarse-grained parallelization* (Pvm [9] and different Mpi [10, 11] versions can be applied) with automatical *load-balancing* on almost any Unix-compatible operating system.

Many different quantum chemical methods can be applied in order to perform the single point calculations, because input and output files are used as interfaces between Snf and different quantum chemical programs. Only their output files are processed to obtain the data, i.e., gradients of the total electronic energy, dipole moments, and polarizabilities. Existing interfaces allow one to use the following electronic-structure programs and methods: [2]

[1] Strictly speaking, we are concerned with the molecular symmetry groups [6–8]. However, as mentioned in [8, p. 16] the symmetry group of a molecule is usually referred to as the *point group* and thus, we shall use this name in the following

[2] Note that VROA calculations are currently only possible for Dalton with HF/DFT.

Figure 1.1: Hierarchical structure of programs


(external quantum chemistry software providing energy gradients)
SNF: symmetry−exploiting evaluation of raw data for vibrational spectra
restart file
SNFDC: raw data collector (parallel [PVM, MPI, LAM] or serial)
output files
TURBOMOLE
DALTON
ADF
GAUSSIAN
MOLPRO

- Turbomole [12]: SCF (HF), (RI-)DFT, (RI-)MP2, RICC2
- Dalton [13]: SCF (HF), MCSCF, DFT, CCS, CCSD, CC2
- Gaussian [14]: DFT (other methods possible, but not tested)
- Adf [15]: DFT
- Molpro: test interface for MCSCF calculations

A scheme of the program structure is shown in Fig. 1.1. Due to the numerical differentiation it is possible to use static as well as dynamic (frequency-dependent) polarizabilities for the calculation of the Raman intensities. By increasing the number of grid points for the numerical derivatives, it is possible to achieve an accurate error control.

Snf diagonalizes the Hessian and determines the translational and rotational contributions to the normal modes. In a second evaluation of the data, non-vibrational contributions to the matrix containing the second

Raman intensities are not available for MP2 methods. However, SNF supports the calculation of MP2 gradients and dipole moments in combination with a SOPPA calculation of polarizabilities using Dalton.

derivatives are projected off. The calculation is completed by a detailed thermochemical analysis, which takes electronic contributions into account (spin-only values are considered).

The preparation of the input files and the selection of all program options can be done by using the interactive program SNFDEFINE. The results of the calculations can easily be displayed, since spectra plots are automatically created using GNUPLOT [16], and the normal modes are written to files in GAUSSIAN98 [17] and MOPAC [18] format, such that tools like MOLDEN [19] or JMOL [20] may be utilized to visualize them.

The manual is organized as follows: In Sect. 2, a description of SNF is given, and the requirements for the parallelization are explained. Sect. 3 gives a short introduction to the most relevant issues for practical applications of SNF. Sect. 4 deals with usage of the programs; all parts are explained in detail, starting at the preparation of the input files with SNFDEFINE and ending with the evaluation of the data collected by SNFDC with SNF. Sect. 5 contains information about all possible keywords for SNF and their meaning. In Sect. 6, the visualization of the results obtained with SNF is explained. Sect. 7 outlines details of the implementations of thermochemistry, and in Sect. 8, the influence of step size, basis sets, SCF convergence, and grid type onto the results of a SNF run is discussed. The different sections in the appendix concern the program history (Appendix A), the supported point groups (Appendix B), program modes (Appendix C, and compilers (Appendix D). Moreover, details of the elimination of vibrational and rotational modes can be found in Appendix E. Finally, Appendix F presents an overview of the tools which are provided with the SNF Package.

Any use of this program that results in published material should **cite the following**:

J. Neugebauer, M. Reiher, C. Kind, B. A. Hess. *J. Comp. Chem.*, **23** (2002) 895 – 910.

Note that this program represents experimental code that is under constant development. No guarantees of any kind are provided, and the authors do not accept any responsibilities for the performance of the code or the correctness of the results.

1.2 Further information and key references

Many details of the implementation of Snf and the underlying theory as well as applications are available in the literature. The following list gives an overview of references for selected topics:

1.2.1 General aspects

- **Theory of infrared and Raman spectroscopy, Snf's algorithmic structure**:
  J. Neugebauer, M. Reiher, C. Kind, B. A. Hess. *J. Comp. Chem.*, **23** (2002) 895 – 910.

- **Benchmark and methods (DFT, CC, MCSCF, . . .) comparison for Raman intensities from static and dynamic polarizabilities**:
  J. Neugebauer, M. Reiher, B. A. Hess, *J. Chem. Phys.*, **17** (2002) 8623 – 8633.

- **Methodological and benchmark study on vibrational Raman optical activity (VROA)**:
  M. Reiher, V. Liégeois, Kenneth Ruud, *J. Phys. Chem. A*, **109** (2005), 7567 – 7574.

- **Role of anharmonicity contributions**:
  J. Neugebauer, B. A. Hess, *J. Chem. Phys.*, **118** (2003), 7215 – 7225.

- **Theoretical Raman scattering cross sections compared to experiment**:
  M. Reiher, G. Brehm, S. Schneider, *J. Phys. Chem. A*, **108** (2004), 734 – 742.

1.2.2 Special topics

- **Intensities for selected vibrations of large systems**:
  M. Reiher, J. Neugebauer, B. A. Hess. *Z. Phys. Chem.*, **217** (2003), 91 – 103.

- **Resonance Raman calculations, excited-state frequencies**:
  J. Neugebauer, B. A. Hess, *J. Chem. Phys.*, **120** (2004) 11564 – 11577.

- **High-performance Raman applications for large molecules**:
  C. Herrmann, J. Neugebauer, M. Presselt, U. Uhlemann, M. Schmitt, S. Rau, J. Popp, M. Reiher, *submitted*, (2007).

- **Theoretical VROA applied to large systems**:
  C. Herrmann, K. Ruud, M. Reiher, *ChemPhysChem*, **7** (2006), 2189 – 2196.

- **Adsorbates on surfaces, harmonic approximation for partially relaxed structures**:
  C. Herrmann, M. Reiher, *Surf. Science*, **600** (2006), 1891 – 1900.

- **Selective calculation of molecular vibrations**:
  M. Reiher, J. Neugebauer, *J. Chem. Phys.*, **118** (2003), 1634 – 1641.

2. Installation

The package Snf was developed for large-scale numerical calculations of vibrational frequencies, which require hundreds of calculations for displaced structures. It is therefore intended to be used in a parallel environment, and the correct setup of such a parallel environment is therefore a prerequisite for installing Snf. This makes the installation process for Snf somewhat more involved, although much effort was made to simplify this step. The different parallelization standards which can be used with Snf are described in the following sections.

Since version 4.0.0, Snf also includes a serial version which allows new Snf users to get acquainted with the Snf machinery under "simplified" conditions, or to run Snf on single-processor machines.

2.1 Installing the serial version

The serial version will run both the master process as well as the slave processes, i.e., the single-point steps with the external electronic structure programs, on the local machine. The installation does, therefore, not depend on any parallel libraries like `mpich`, `pvm3` etc.

You can install the serial version of Snf by typing:

```
configure --with-serial [--with-FC=<compilername>
 --with-initial-scp]
```

In that case, the setup tool Snfdefine will still prepare the input files needed for a parallel run (see below), but they will not be used by Snfdc.

2.2 Parallelization standards

In order to use the parallelized data collector Snfdc, it is necessary to provide the software for one of the following parallelization standards.

2.2.1 Pvm

To run the Pvm version of Snfdc, it is necessary to install Pvm. It can be obtained from

`http://www.epm.ornl.gov/pvm/pvm_home.html`
It is necessary to set the following environment variables for PVM:

`PVM_ROOT`: PVM installation directory

`PVM_INCL`: parent directory to PVM include-files directory (not necessary if `PVM_ROOT/include` exists)

`PVM_ARCH`: architecture of the computer (LINUX or ALPHA)

SNFDC will look for the file `fpvm3.h` in `$PVM_ROOT/include` or `$PVM_INCL/include`, and for PVM libraries in `$PVM_ROOT/lib/$PVM_ARCH`

In order to avoid problems with incompatible objects, it is in general preferable to compile PVM using the same F90 compiler as used for compiling SNF. In our PVM installation (version 3.4.5), we noted problems when compiling PVM using `gcc` version 4.0 or higher. Switching to `gcc-3.3.6` solved these problems.

2.2.2 MPI, MPICH, LAM/MPI

MPI, MPICH, and LAM/MPI, resp., may be obtained from
`http://WWW.ERC.MsState.Edu/misc/mpi/`,
`http://www-unix.mcs.anl.gov/mpi/mpich`, or
`http://www.lam-mpi.org/`.
For the MPICH version it is necessary to set the variable `MPICH`, which is the path to the MPICH (or MPI) directory. SNFDC uses `include` files from `$MPICH/include` and libraries from `$MPICH/lib`. If you would like to use the LAM/MPI version, the variable `$LAMHOME` must be set. In case of the native MPI version no additional variables are required.

Note that MPICH provides a wrapper `mpif90` for the compiler that was used to build the MPICH version. This wrapper should be specified as the compiler for the SNF package with the configuration option `--with-FC=mpif90` in order to avoid problems with inconsistent objects.

The MPI version for which SNF was tested is `mpich-1.2.7p1`.

2.3 SNFDEFINE, SNFDC, and SNF

SNFDEFINE, SNFDC, and SNF can be installed using the automatic `Makefile` generation by `configure`. For this purpose, you should run

```
configure --with-<par>
```

in the subdirectory src of the SNF installation directory to create the makefiles and to prepare the compilation using the <par> parallelization standard. Valid arguments for <par> are:

- pvm: use the PVM version of SNFDC
- mpi: use the MPI version of SNFDC, which is native to the machine in question (e.g. on a SGI installation)
- mpich: use the MPICH version of SNFDC
- lam: use the LAM/MPI version of SNFDC
- serial: use the serial version of SNFDC (available since version 4.0.0)

Other configuration parameters can be displayed by running
configure --help
Of particular importance is the option --with-initial-scp. With this option, all inputfiles are copied from the master to the slave nodes using scp. Otherwise, the files will be copied using PVM or MPI, since scp might not be available on some computers, but this can be very slow if large files have to be distributed, such that scp is preferable, if it can be applied on your computers.

If you experience problems when linking to precompiled BLAS and LAPACK libraries, you can use the option --with-locallib.

If you would like to use SNF on a shared memory multiprocessor machine, you should use the option --with-SMP.

For non-standard compilers, it might be necessary to set path variables or to add/modify compiler flags[1]. The compiler information are stored (and read by configure) from the file PARM in the src/aux subdirectory of the SNF installation directory. To select another compiler, use the option --with-FC=<compilername>.

The configure script will also check if certain shell variables are set, which are necessary for the libraries linked to SNF. If these variables are not set, configure will give warnings or error messages.

Note that different compilers/settings can be used for the F90 programs SNF, SNFDEFINE and the F77 program SNFDC by running configure separately in the subdirectories src/snf and src/snfdc.

After this automatic generation of the makefiles, you may compile all packages by typing make (either in the src directory, or in the subdirectories snf or snfdc). The executable files can be found in the bin subdirectory of the SNF installation directory.

[1] See Appendix D for details

2.4 Further programs and scripts needed by SNF

In order to perform the single point calculations, one of the supported electronic-structure programs must be available on every node in the parallel machinery in `$PATH`.

Furthermore, the scripts from the `scripts` subdirectory of the SNF installation directory must be available in `$PATH`. The script `choose_nodes` also requires the file `ELIGIBLE` in the directory `$CIPROC`, which contains the names of computers available and their numbers of processors (see Sect. 4.5). An example file is provided with the SNF package. The variable $CIPROC must be set by the user. `choose_nodes` also requires the shell-command `ruptime` in an PVM environment in order to facilitate load-balancing and automatic addition of idle nodes on a cluster. See, however, Sections 4.5 and 4.5.2 for how to circumvent the usage of `ruptime` and `ELIGIBLE`. For the automatic generation of spectra plots, GNUPLOT must also be present in `$PATH`.

The variable `$SNF_PATH` should be set to the SNF home directory, so that executables can be taken from `$SNF_PATH/bin`.

2.5 Test suite

Starting from version 4.0.0, a test script, `$SNF_PATH/example/snftest`, is available which runs SNF test jobs and controls the results automatically. If invoked as `$SNF_PATH/example/snftest all`, this script will run a series of test jobs using all quantum chemistry programs for which interfaces are available (except for GAUSSIAN98, which may be tested separately). In can be run with one or several of the arguments `turbomole, dalton, adf, gaussian` or `molpro`, or also with names of test directories such as `turbomole/ridft/ir_raman`, in order to restrict the test to specific programs or test jobs only. It will run the jobs in the test subdirectories of the directory `$SNF_TESTDIR`, so this path should be set accordingly (usually to `$SNF_PATH/example/snftest`). `snftest` features an automatic check of the results of each test run, which is performed by calling the script `checkresults`. Information on whether (and which) test jobs failed is written to the screen and to the file `$SNF_TESTDIR/snftest.log`.

3. Quick start

This section provides a short summary of the most relevant practical aspects of Snf. For a detailed discussion of these issues, see the next chapter.

3.1 Checklist

Before running Snf, assure that the Snf environment is properly set, i.e., that

1. the environment variable `$SNF_PATH` is set to the Snf directory where the binaries you want to use are stored (in `$SNF_PATH/bin`),
2. your are able to log in on all nodes under their full name (e.g. `machname.ethz.ch`) and under any shortcut names you want to use (e.g. `machname`) without having to enter any passwords or conformations,
3. the executables of the quantum chemistry programs needed for the slave jobs, i.e., Turbomole, Dalton, Adf, Gaussian, or Molpro, are available on every slave node directly after opening a shell there. If a special path "`path/to/executable`" is to be used, it needs to be defined in the `DCINPUT` file of the working directory as "`PROGDIR path/to/executable`".

For Pvm users, it is furthermore necessary that

1. the script `choose_nodes` operates properly and chooses only from the list of slave nodes you want to employ (i.e., uses a suitable `ELIGIBLE` file); see, however, Sections 4.5 and 4.5.2 for how to circumvent the usage of `ruptime` and `ELIGIBLE`, which may be necessary when working with queuing systems.
2. no old Pvm demons are running on the slave nodes, and all old `pvmd.xxx` and `pvml.xxx` files are removed.

3.2 Essential steps

In order to calculate a vibrational spectrum with Snf, you should proceed as follows:

1. Assure that all necessary input files are present in your working directory. Which input files are required, depends on the quantum chemical program package to be used for the single-point calculations — it will usually be the same files as needed in order to start a molecular structure optimization.

2. Run `snfdefine` in order to set up the calculation. Snf settings are written to `snf_control`

3. Type `snfdc` in order to start the parallel evaluation of molecular gradients and properties for the distorted structures. These data will be written to the file `restart`.

4. Type `snf` in order to evaluate the raw data produced by `snfdc`. The output is contained in `snf.out`.

4. Frequency Analysis with Snf

The frequency analysis and the calculation of vibrational spectra using Snf consists of three steps, which are described in Sects. 4.4, 4.5, and 4.6. The following two sections give an overview of the whole procedure and the preparations necessary before starting the calculation.

4.1 Overview

The first step in the parallelized calculation of vibrational spectra is the preparation of the input files. Therefore, it is necessary to copy master input files for the single point calculations in the directory which serves as the working directory for the calculation. These input files are modified by the preparation tool Snfdefine, which also creates a restart file for the frequency analysis and an the additional input file `snf_control`. The latter contains general program and output control flags. The generic names of the input files needed for the different electronic structure programs are:

- Turbomole: `control, coord, basis, mos or (alpha, beta) [, auxbasis]`
- Dalton: `molecule.dal, molecule.mol`
- Adf: `adf.in`
- Gaussian: `gaussian.com`
- Molpro: `molpro.in`

A detailed description follows in Section 4.2.

The next step, the most time-consuming part, is the parallelized calculation of gradients of the total electronic energy, dipole moments, and polarizabilities for each distorted molecular structure. These calculations have to be performed in C_1 symmetry since the distortions destroy the molecular symmetry. The calculations are done by the data collector Snfdc, which distributes the single point calculations to all processors available. The data is stored in the restart file.

When the single point calculations are finished, the data is evaluated by Snf, which creates the output file `snf.out` containing the vibrational frequencies, the infrared and Raman intensities, the thermochemical data, the normal modes and other data depending on the output control flags. Additional output files can be generated for a graphical representation of the spectra and the normal modes.[1]

4.2 Preparations

The program Snf requires master input files, which are distributed to all nodes of the parallel machine and modified for each slave process. In general, the type of the input files provided by the user determines the quantum chemical program package which Snf will use in the single-point calculations on the slave nodes. An exception is possible in the case of Turbomole input files, which can be automatically converted into Dalton input files, so that Dalton can be used for the single-point calculations.

4.2.1 Turbomole

It is necessary to provide the following files in the working directory for the frequency analysis:

- `control`: general input file for the Turbomole calculation.
- `coord`: nuclear coordinates of the atoms. This must be the optimized structure of the molecule.
- `basis`: basis set information
- `mos`: molecular orbital coefficients. These MO vectors will be used as starting vectors for every single point calculation in the frequency analysis.
- `alpha, beta`: the files `alpha` and `beta` have to be supplied instead of the file `mos` for UHF/UKS calculations.

If a geometry optimization with Turbomole has been performed, all these files are already created and Snfdefine can be started.

The type of the calculation is in principle determined by the type of `control` file provided. If the geometry optimization performed was a Hartree-Fock `dscf` calculation, Snf will also perform HF single point calculations.

[1] If no isotope settings are specified Snf uses the masses of the most abundant isotopes for all evaluations (see Sect. 5.1.2).

If it was a `ridft` calculation, density functional calculations using the RI-approximation will be done. Note that no Raman intensities can be calculated using MP2/RI-MP2 methods, but it is possible to calculate Raman intensities from SOPPA polarizabilities in combination with frequencies and infrared intensities from MP2 calculations.

Starting from TURBOMOLE input, it is also possible to request the use of DALTON for the single-point calculations (which is necessary, for example, if VROA intensities shall be calculated after a TURBOMOLE geometry optimization). Appropriate input files in DALTON format will be generated by SNFDEFINE in this case, where the user has to specify the DALTON settings in an interactive menu.

4.2.2 Dalton

It is also possible to use DALTON input files. The following two files must be provided for this type of preparation:

- `molecule.dal`: general input file for the DALTON calculation.
- `molecule.mol`: nuclear coordinates of the atoms and basis set specification.

For DALTON applications, the type of the calculation is not determined automatically. Therefore, the user has to select the type of calculation in an interactive menu during the `snfdefine` run.

The user has to take care of the following formal constraints:

- the input files should be named `molecule.dal` (general input) and `molecule.mol` (geometry and basis set input)
- Z-matrix input is not supported; use Cartesian coordinates instead
- If the `INTGRL` basis set specification is applied, do not use the free format for exponents and coefficients; use either the default or the high precision format (`H` or `h`; see DALTON manual for details)
- Do not use symmetry generators; you may either use the automatic symmetry detection routines of DALTON (`SYMTXT` = ‘ ’) or a C_1 input (`SYMTXT`=‘ 0’). `snfdefine` determines the molecular symmetry through its own symmetry detection routines and sets the symmetry to C_1 in the single point calculations.

4.2.3 ADF

ADF users can use the runscript of an ADF geometry optimization directly. SNFDEFINE will recognize the `GEOMETRY` section and replace it by a section requiring a geometry optimization which is stopped after one cycle. Furthermore, some convergence options may be altered (see below). This runscript will be distributed onto the slave nodes, where the molecular coordinates will be changed. Apart from this, the settings in the ADF scripts are automatically those which will be used in the single-point calculations on the slaves. The runscript has to fulfill several conditions:

1. It must be named `adf.in`.
2. It must be executable.
3. It must contain exactly the settings used in the preceding geometry optimization.
4. The geometry must be specified in Cartesian coordinates in units of Å
5. The binary output file must be named `TAPE21`.
6. It should not contain any blocks requesting other calculations than a geometry optimization.

SNFDEFINE will check (and modify, if necessary) the following blocks in the `adf.in` file:

- `GEOMETRY` — delete it (if present) and write instead:

```
Geometry
  GO
  iterations 1
End
```

- `INTEGRATION` — if the grid settings are below `6.0 6.0`, replace them by `6.0 6.0`.
- `SCF` — if the convergence criteria are below `converge 1.0e-6 1.0e-6`, replace them by `converge 1.0e-6 1.0e-6`.
- Sort the list of Cartesian coordinates according to their nuclear charge. (This will facilitate reading in the data of the distorted structure-single point calculations in the SNFDC run.)

- OCCUPATIONS — the options KEEPORBITALS=1000000 SMEAR=0.0d0 will be set. If not present, this keyword will be added. This prevents ADF from using SCF convergence tools which may lead to results that are only acceptable in an geometric structure optimization, but not in a frequency run.

If SNFDEFINE modifies the integration or convergence parameters, it will print a warning message to the screen remembering the user to redo a geometry optimization with the new settings.

4.2.4 Gaussian

If GAUSSIAN shall be used for the single-point calculations, the user must provide a GAUSSIAN input file which must have the following characteristics:

- It must be named gaussian.com.
- It must contain the keyword #p force in addition to the keywords for the method and the basis set to be used.
- The equilibrium geometry must be given in Cartesian coordinates — z-matrix input is not accepted.

For an ethanol molecule, the gaussian.com file might look like this (with the last line being a blank line, of course):

```
#p force  bp86/TZVP

ethanol

0 1
C -1.225747   -0.225340    0.000002
H -1.296072   -0.865520   -0.891479
H -1.296085   -0.865499    0.891496
H -2.083095    0.464915   -0.000013
C  0.079992    0.551234   -0.000000
H  0.131474    1.203598    0.893871
H  0.131470    1.203596   -0.893875
O  1.161283   -0.399398   -0.000003
H  1.996579    0.098737    0.000020
```

Since this input file will be distributed onto the slave nodes, where nothing will be changed in it but the molecular coordinates, the settings defined there are automatically the settings which are used in the single-point calculations.

Optional additional sections (such as user-provided basis sets) are in general supported. However, sections related to the preceding structure optimization (such as specifications of frozen coordinates) should be deleted in the `gaussian.com` file.

4.2.5 Molpro

The MOLPRO interface in SNF is not yet fully developed, and using MOLPRO with SNF therefore requires special caution. Therefore, only MCSCF calculations can be carried out with MOLPRO in connection with SNF at the moment. Furthermore, SNF only supports calculations with MOLPRO version 2002.x; the new input/output format in version 2006.x is not yet supported in all cases. The MOLPRO interface is suited for calculating IR intensities, but not for Raman or VROA intensities.

The following requirements apply to the input file:

- It should be named 'molpro.in'.
- It has to start with "`***`" in the first data line.
- The coordinates of the equilibrium structure should be given in Å. Therefore, the keyword 'geomtyp=xyz'is necessary and the geometry block ('geometry={ }') should contain the Cartesian coordinates in XYZ format. This format consists of two header lines, the first of them contains the number of atoms, and the second one is either empty or it contains a title. Each of the remaining lines specifies the coordinates of one atom, with the chemical symbol in the first field, followed by the x, y, z coordinates. The coordinates and the chemical symbol have to be separated by a comma. A sequence number may be appended to the chemical symbol without using a comma. No blanks should be used.
- The interface is implemented for MCSCF calculations, therefore the keyword 'mcscf' must be included in the input file.
- For gradient calculation, 'gradtype=alaska' and 'force' must be added.
- The basis is chosen with 'BASIS='and the name of the basis set.

For further information see `www.molpro.net`.

4.3 Preparations for excited-state calculations

Excited-state calculations can be performed using the `egrad` module of the Turbomole V5.6 package. This module allows to calculate analytical excited state gradients, which can in turn be used for a geometry optimization of the excited state using SCF or DFT calculations. For the particular usage of the `egrad` program, we refer to the Turbomole manual. It should be kept in mind, however, that the frequency calculation is performed in C_1 symmetry, and therefore, the number of the excited state to be calculated must include all lower-lying excited states of the same spin-symmetry. In a calculation with a non-trivial point-group symmetry, the state-numbering refers always to excited states of a particular irrep. Note that no Raman spectra can be calculated for these excited states.

Excited-state calculations and optimizations can also be performed using Dalton's CASSCF module.

CAUTION: Snf assumes that you are calculating frequencies far away from any avoided crossing/conical intersection/near-degeneracy situation. If this is not the case, then there is a potential danger that the electronic character of a certain excited state changes from the equilibrium to the displaced structures. Additionally, the Born–Oppenheimer approximation will not be valid in such a context, upon which the frequency analysis in Snf is based.

4.4 Snfdefine

You can run Snfdefine simply by typing `snfdefine`.

First, the program checks whether the mandatory files for running single-point calculations with the desired quantum chemical program package are present in the working directory (see Section 4.5.2 to see which files these are). Then it tries to read the essential information from them. In case of Turbomole single-point calculations, after reading the Schoenflies symbol of the molecular point group from `control`, Snfdefine checks the symmetry of the molecule in the `coord` file and determines the distortions which are redundant because of symmetry. From version V2.2.1 on, Snfdefine can determine the correct symmetry of the molecule independent of its orientation in space, while former versions could only validate or falsify the specified symmetry[2]. The symmetry detection routines will only be applied

[2] The symmetry detection routines available in former versions of Snf require Turbomole's standard orientation for a successful determination of the molecular point group (i.e., the z-axis must be the principal axis of rotation etc.). This is still required for cubic

if the Schoenflies symbol in the `control` file is inconsistent with the geometrical data in the `coord` file or if their invocation is selected by the user. If DALTON, ADF, GAUSSIAN, or MOLPRO input files are used, SNFDEFINE will always determine the symmetry on its own.

For a n-point central differences formula, it is necessary to perform $(n-1)$ distortions in the direction of every Cartesian nuclear coordinate, i.e., $3 \cdot (n-1) \cdot N$ single point calculations are necessary for a molecule with N atoms if no symmetry information is available. However, the number of non-redundant distortions is by far smaller for highly symmetric molecules (compare benzene[3] (8 distortions instead of 72), SF_6 (4 instead of 42) or the Buckminsterfullerene C_{60} (5 instead of 360)). SNFDEFINE creates the file `restart` which contains entries for each of the $3 \cdot (n-1) \cdot N$ distorted geometries. As an example, the `restart` file for N_2 as created by SNFDEFINE is given below.

```
snf calculation
(4d20.10)
  170    1    1    1    6    8    2
    0.1000000000D-01
mydir
tempdir
  1  0  1  1  1  0  0
  ------  ------  ------  ------  ------  ------
  2  1  1  1  1  1  1
  ------  ------  ------  ------  ------  ------
 dipoles atom             1
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00
 dipoles atom             2
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00
 gradients atom             1
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
 gradients atom             2
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
 polarizabilities atom              1
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
```

and icosahedral groups.

[3] 3-point central differences formula assumed

```
   0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
   0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
   0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
polarizabilities atom            2
   0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
   0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
   0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
   0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
   0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
   0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
   0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
   0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
   0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
   0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
   0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
   0.0000000000D+00    0.0000000000D+00    0.0000000000D+00    0.0000000000D+00
```

The entries have the following meanings:

line No.	meaning
1	title
2	format string for data (dipole moments, gradients, and polarizabilities)
3	`version, mode, inputpol, iomeg, nvar, scfconv, naus`
4	`cstep`
5	working directory
6	scratch directory

with the variables

- `version`: version number of this restart file
- `mode`: characterizes the type of the calculation (see Appendix C)
- `inputpol`: flag for polarizabilities: 0 = no polarizabilities, 1 = static polarizabilities, 2 = dynamic polarizabilities
- `iomeg`: number of sets of dynamic polarizabilities (if `inputpol`=2); SNF allows to calculate Raman intensities based on frequency-dependent polarizabilities for up to 10 different frequencies.
- `nvar`: number of nuclear coordinates = number of atoms ×3
- `scfconv`: convergence criterion for the energy ($10^{-\texttt{scfconv}}$)
- `naus`: number of steps along each coordinate (for a n-point central differences formula for the numerical derivatives, `naus`= $(n-1)$; n must be odd).
- `cstep`: step size for the numerical derivatives, i.e., distortion of the Cartesian nuclear coordinates, in [*bohr*].

The names for the working and scratch directory will automatically be written to the `restart` file by SNFDC.

The next paragraph of the `restart` file contains two lines for every atom. The first line shows the number of the atom (in order of appearance in the file `coord`), and a step-flag for each distortion along one of the Cartesian coordinates of this atom, i.e.,

$$+x, -x, +y, -y, +z, -z$$

for a 3-point formula, and

$$+2x, +x, -x, -2x, +2y, +y, -y, -2y, +2z, +z, -z, -2z$$

for a 5-point formula, etc. (in units of `cstep`). Possible values for the step flags are:

0 single point calculation has not been done yet for this point

-1 calculation for this point is running or aborted

1 calculation for this point is finished (or redundant because of the molecular symmetry)

The second line for each atom contains sets of six dashes for each step, which are replaced by SNFDC by the name of the node that calculated the corresponding step. The control flags for the symmetry-redundant distortions are set equal to 1 by SNFDEFINE, which simulates a point on the PES that has already been calculated, such that during the run of SNFDC these points will not be calculated.

Components of the dipole moments, gradients, and polarizabilities are stored in the data fields, which represents the largest part of the `restart` file. SNFDEFINE fills all these fields with zero entries. If the calculation of Raman intensities is switched off, there are no data fields for the polarizabilities in the `restart` file.

In case of TURBOMOLE single-point calculations, SNFDEFINE will modify the `control` file for the TURBOMOLE single point calculations. Be aware that the original control file, which contained all entries with the correct point group symmetry will be overwritten by the C_1 settings. This is necessary to prepare the calculations of dipole moments, gradients, and polarizabilities. The first time you run SNFDEFINE in a particular directory, it will create a write-protected backup copy of the original `<inputfile>` (the copy `<inputfile>.bak` is only created if it does not already exist).

Moreover, SNFDEFINE allows to set a lot of options for the calculation and the final output of SNF (see Sect. 5). All SNF-specific settings are stored in the file `snf_control`, which is copied onto every slave node during the SNFDC run.

You can also use SNFDEFINE to modify options. If calculations have already been performed, SNFDEFINE will not overwrite the `restart` file without creating a backup-copy.

SNFDEFINE tries to generate the input files `USE_NODES` and `DCINPUT`, which are necessary to run SNFDC. Therefore, the scripts `choose_nodes` and `mkdcinput` must be available in the path, and the `ELIGIBLE` file must be present in the directory `$CIPROC`. While `DCINPUT` is a mandatory inputfile for SNFDC, `USE_NODES` is only necessary for the PVM version of SNFDC. SNFDEFINE furthermore creates the files MACHINES and `mpi.sub`, which may be used to run the MPI version of SNFDC. These scripts and files will be explained in Sect. 4.5.

Because all following single point calculations have to be performed in C_1 symmetry, the file `mos` (or `alpha` and `beta` in UHF/UKS cases) must contain MO starting vectors in C_1 symmetry. If the original symmetry of the molecule is higher, you can get the C_1 start-vector by using TURBOMOLE'S DEFINE.

Summary:

⇒ If you have performed a geometry optimization with TURBOMOLE, DALTON, ADF, GAUSSIAN, or MOLPRO just type `snfdefine` in your working directory.

⇒ Select the options in the menus (see Sect. 5 for details).

⇒ Note possible warnings in the output of SNFDEFINE.

⇒ If the symmetry mentioned in the original `control` file is higher than C_1, create a C_1 MO file by using TURBOMOLE's DEFINE when using TURBOMOLE for single-points.

⇒ For the generation of suitable input files from (DALTON) input see also Sect. 4.2.2.

- Mandatory files for SNFDEFINE:
  `control`, `coord` (TURBOMOLE), `molecule.mol` and `molecule.dal` (DALTON), `adf.in` (ADF), `gaussian.com` (GAUSSIAN), or `molpro.in` (MOLPRO)
- Mandatory files and scripts for automatical generation of SNFDC input files by SNFDEFINE:
  `$CIPROC/ELIGIBLE`, `choose_nodes`, `mkdcinput`
- Files created by SNFDEFINE:
  `restart`, `snf_control`, `<inputfile>.bak`, `DCINPUT`, `USE_NODES`, `MACHINES`, `mpi.sub`

4.5 Snfdc

Some further preparations are necessary to start the parallelized calculation with Snfdc besides the generation of the `restart` file and the modification of the `control` file.

4.5.1 General input: DCINPUT

First of all, it is necessary to supply the separate input file DCINPUT. This file has the following structure:

```
DCPATH /usr/bin/snfdc
MYDIR /home/<usrname>/calculations/example
PREFIX TMP<usrname>%
TMPDIR /tmp
LOGDIR /home/<usrname>/calculations/example/log
```

with the variables

- `DCPATH`: path to the Snfdc executable file (optional) (will be set to `$SNF_PATH/bin/snfdc` if you use `mkdcinput` to generate `DCINPUT`)
- `MYDIR`: your working directory (mandatory)
- `PREFIX`: prefix for logfiles and temporary files (optional)
- `TMPDIR`: directory for temporary files (including all temporary files of the single point calculations); must be available on every machine (optional)
- `LOGDIR`: directory for logfiles (optional)

If the optional variables are not specified in `DCINPUT`, Snfdc will try to use `$MYDIR/tmp` as `TMPDIR` and `$MYDIR/log` as `LOGDIR`. `PREFIX` will be set to "`TMP`".
The variables `DCPATH`,`TMPDIR`, and `LOGDIR` may be specified individually for some (or all) of the slave nodes. This is achieved by the entry
`DCPATH(<nodename>) <pathname>`
etc. Additional optional variables are

- `PROGDIR`: path to the Turbomole, Dalton, Gaussian, or Molpro executable files. Since Adf always requires to set the directory `ADFBIN`, Adf binaries will always be started as specified in the input file.

- CHOOSENOPT: options for the choose_nodes command when executed by SNFDC (or SNFDEFINE). Examples for options are:
  - -e <ELIGPATH>: sets path to the ELIGIBLE file which shall be used in a calculation to <ELIGPATH> (default: $CIPROC/ELIGIBLE)
  - -ignore: fake call that leaves the file USE_NODES unchanged; if USE_NODES does not exist, it creates an USE_NODES file without node entries (no reference to ruptime or ELIGIBLE).
  - -local: creates an USE_NODES file that just contains the local machine with one CPU (no reference to ruptime or ELIGIBLE).
  - -nlocal <n>: creates an USE_NODES file that just contains the local machine with <n> CPUs (no reference to ruptime or ELIGIBLE).

If you want to perform calculations with values for the variables cstep or scfconv which are out of the confidence interval (SNFDEFINE will inform you if this happens; see Appendix 5.1.3) the entry

```
CRAP_OK yes
```

must be added to DCINPUT.
To create a standard inputfile, you can use the script mkdcinput, which is automatically executed by SNFDEFINE. This file should be executed in the working directory. The settings for DCPATH and TMPDIR in this script must be adopted to the local settings before using it.

4.5.2 Preparations for the PVM version

The PVM version of SNFDC was originally intented to be applied in the following way: On a cluster on which several jobs may be running, SNFDC can be used to “fill up” idle nodes. That is done by means of the ruptime command, which reports the load status of all the machines of the cluster. SNFDC will then distribute the single-point calculations according to the load status of the nodes. The script choose_nodes calls ruptime in order to get the load information for all the machines, and it compares this information to a list of eligible nodes and their maximum acceptable load, which is per default taken from the file $CIPROC/ELIGIBLE. A typical ELIGIBLE file will look like:

```
tcl5   2
tcl6   2
tcl7   2
tcl8   2
tcl9   2
```

```
tcl10  2
tcl11  2
tcl12  2
tcat1 1
tcat2 1
tcat3 1
tcat4 1
tcat5 1
tcat6 1
tcat7 1
tcat8 1
tcat9 1
tcat10 1
```

By using the script choose_nodes, all processors which are idle will be added to the list in USE_NODES. The latter file is of the form

```
# 47 nodes with 69 processors are eligible.
# parallel environment is pvm
# Thu Oct  4 12:24:01 MEST 2001
node tcat1 1 1
node tcat10 1 1
node tcat2 1 1
node tcat4 1 1
node tcat5 1 1
node tcat8 1 1
node tcat9 1 1
node tcl5 1 2
node tcl6 1 2
node tcl7 1 2
node tcl8 1 2
node tcl9 1 2
# chosen 12 machines with 17 processors.
```

The file USE_NODES must be present in the directory where Snfdc is started. If you are using a queuing system, you must make sure that USE_NODES is copied to the scratch-directory where the actual Snfdc calculation is carried out. Usually USE_NODES is created by a call to choose_nodes that is automatically performed by Snfdefine. The program Snfdc itself also uses choose_nodes (Pvm version only; the Mpi version works without any reference to choose_nodes) in order to update the file USE_NODES during the calculation. If nodes have a heavy load, it will stop sending new single-point calculations to these machines. If, on the other hand, some nodes that had a high load before run idle, it will send new single-point calculations to them (provided they are listed in ELIGIBLE).

On clusters where a queuing system has to be used, it is not useful to perform these steps. The file USE_NODES (or the machine-file in case of the

MPI version) should be created once at the beginning of the calculation by the submit-script (see the examples in `scripts/submit_scripts`) and then remain unchanged. To avoid that the `choose_nodes` calls change the `USE_NODES` file, you can run it with the option `-ignore`. In order to forward this option to SNFDEFINE and SNFDC, you can use the option

```
CHOOSENOPT -ignore
```

in the file `DCINPUT`. This option can also be set via SNFDEFINE. See also Section 4.5.

Both `choose_nodes` and `mkdcinput` are executed by SNFDEFINE, such that no further preparations are necessary if the file `mkdcinput` is in line with the local settings (path containing the SNFDC executable, `TMPDIR`). After these preparations, SNFDC can be started using the command `snfdc`.

4.5.3 Preparations for the MPI versions

If one of the MPI versions shall be used, this can be done by the command

```
mpirun -machinefile MACHINES -np <number of CPU's> snfdc
```

The file `MACHINES` must contain the names of all computers available for the parallel calculation. You may create this file by running

```
hoose_nodes -mpi
```

and then renaming the file `USE_NODES`, which is created by `choose_nodes`, to `MACHINES`. This is done automatically by SNFDEFINE, which also creates the runscript `mpi.sub`. The runscript contains the `mpirun` command mentioned above; the number of CPU's is extracted from the files `USE_NODES`. Hence, after running SNFDEFINE you may immediately run the MPI version of SNFDC using the command `mpi.sub`. Note that it might be necessary to modify the `mpi.sub` runscript, since different MPI installations may use different options.

4.5.4 SNFDC calculations

SNFDC will first read the most essential input files specifying the settings for the program to be used in the single-point calculations (that is `control`and `coord` in case of TURBOMOLE calculations, for example), as well as the `restart` file to determine the points which have to be calculated. These points are then distributed to the processors chosen in the file `USE_NODES`. Temporary directories are created in the `TEMPDIR` directories of the slaves. A logfile is generated for every slave process. These logfiles — and also the file `fort.41`, which is created in the working directory (`MYDIR`) — can be used to get information about errors which occur during the run of Snfdc in the master process and the slave processes. To check for errors in the parallel machine, the Pvm files `pvml.<uid>` and `pvmd.<uid>` can be used.

Most important to observe the progress of the calculation is the file `TMPdcstat` [4], which contains information about the status of each slave process. It consists of 11 columns:

```
 2   1 tcat1     vbsy       0   90005   1 rdy     1    1 results sent
 3   2 tcat10               0   90005   1 rdy     2    2 results sent
 4   3 tcat11               0   90005   1 rdy     3    3 results sent
 5   4 tcat2     vbsy       0   90005   1 rdy     4    4 results sent
 6   5 tcat3     vbsy       0   90005   1 rdy     5    5 results sent
 7   6 tcat4                0   90005   1 rdy     6    6 results sent
 8   7 tcat5                0   90005   1 rdy    13   13 results sent
 9   8 tcat8                0   90005   1 rdy    14   14 results sent
10   9 tcl20                0   90005   1 rdy    15   15 results sent
11  10 tcl22                0     103   0 run    16   16  968:15 dscf
12  11 tcl23                0   90005   1 rdy    17   17 results sent
13  12 tcl24                0     102   0 run    18   18 1227:31 escf
```

The rows contain the following entries:

1 task ID
2 continuous numbering
3 node name
4 entry `vbsy` indicates a “very busy” machine
5 process ID (0 for finished processes)
6 cpu-load times 100, or
90005 for finished calculations, or
70005 for trouble in Turbomole programs, or
80005 for wrong results.
7 number of finished single point calculations on this node
8 status:
`rdy` = ready for next step,
`run` = running calculation,
`wait` = node waiting,
`pvme` = Pvm error
9 step number
10 step number (redundant)
11 cpu time and name of running program or message

As already indicated by the name of the data file, Snfdc has got a

[4] This file is not written if the `configure` option `--with-SMP` is used.

restart facility, such that any partly completed `restart` file can be supplied in a SNFDC run, if the program has been aborted.

Summary:

⇒ If the files `restart` and `control` have been prepared by SNFDEFINE, generate the input file `DCINPUT` for SNFDC using the script `mkdcinput` (this is done automatically if SNFDEFINE has been used before).

⇒ Create the file `USE_NODES` using the script `choose_nodes` (this is also done automatically by SNFDEFINE).

⇒ If necessary, stop all old PVM demons and remove PVM logfiles.

⇒ Type `snfdc` to start the parallelized calculation.

⇒ You may check the progress of the calculation via the status file `TMPdcstat`

- Mandatory files for SNFDC:
  `control, coord, basis, mos` [or `alpha, beta`] (TURBOMOLE),
  `molecule.mol` and `molecule.dal` (DALTON),
  `gaussian.com` (GAUSSIAN),
  `adf.in` (ADF), or
  `molpro.in` (MOLPRO)
  `restart, DCINPUT`;
  PVM version: `$CIPROC/ELIGIBLE, USE_NODES`
  MPI versions: `MACHINES` (for `mpirun`)

- Mandatory scripts for SNFDC:
  `choose_nodes, dc_killtask, ruptime`

- Files created by SNFDC in the working directory:
  `TMPdcstat, fort.41`

- Files created by SNFDC in the log directory:
  `PREFIXdclog.XXXXXXX`

- Files created by PVM during the parallelized calculation:
  `pvml.<uid>, pvmd.<uid>`

4.6 Snf

If all required data are available in the `restart` file, the evaluation can be performed by running `snf`. Snf will then create several output files:

- `snf.out`: main output-file (see description below)
- `line_spectrum.dat`: list of vibrational frequencies, IR and Raman intensities for a graphical representation (using, for instance, Gnuplot)
- `gauss/lorentz_spectrum.dat`: data for a model spectrum with peaks in shape of Gauss/Lorentz-functions (can also be plotted by Gnuplot).
- `ir_line.gpin, ir_line.gpin`: Gnuplot input files for the generation of EPS output of infrared spectra
- `raman_line.gpin, raman_line.gpin`: Gnuplot input files for the generation of EPS output of Raman spectra
- `ir_line.eps, ir_gauss/lorentz.eps, raman_line.eps, raman_gauss/lorentz.eps`: Spectra plots (EPS format)
- `g98.out`: Gaussian98 fake output for graphical representation of normal modes using, e.g., Molden (optional)
- `mopac.out`: Mopac fake output for graphical representation of normal modes using, e.g., Molden (optional)
- `xmol.XYZ.out`: Output readable (as XYZ) by Xmol [21] for graphical representation of normal modes (optional; it seems unclear if Xmol is still maintained)

Note that Gnuplot must be available in your path, otherwise the automatical generation of the EPS files by Snf will fail. However, any plotting program can be used to generate plots from the `.dat` files manually.

The main output file `snf.out` briefly summarizes details concerning the calculation, i.e., the options from the `snf_control` file and settings from the `restart` file, and the Cartesian nuclear coordinates of the molecule. The results of a symmetry check for the molecule and for the entries in the Cartesian Hessian matrix are shown. Moreover, a charge decomposition according to Cioslowski [22] is performed.

The next part of the `snf.out` file contains information about the vibrational modes. Wavenumbers as well as translational and rotational percentages of the modes are shown for each irrep. It follows a second evaluation, in which the translational and rotational contributions are eliminated before diagonalizing the Hessian matrix, such that this evaluation should yield more accurate results. Infrared and (if selected by the user) Raman intensities are given in this part of the analysis.

The next paragraph contains moments of inertia, the zero point kinetic energy (ZPE) and thermochemical data, followed by the mass-weighted normal coordinates and a list of all output files created. Further information can be obtained by choosing the corresponding output options in the options menu of SNFDEFINE (see Sect. 5).

Summary:

⇒ If all data are collected by SNFDC and written to the `restart` file, run `snf` in your working directory.

⇒ You may modify the output options by SNFDEFINE or by editing the file `snf_control` and re-run SNF.

⇒ Note possible warnings in the output of SNF.

- Mandatory files for SNF:
  `coord, control` (TURBOMOLE),
  `molecule.mol, molecule.dal` (DALTON),
  `gaussian.com` (GAUSSIAN),
  `adf.in` (ADF),
  `molpro.in` (MOLPRO),
  `restart, snf_control`

- Mandatory program for automatic generation of spectra plots: GNUPLOT.

- Files created by SNFDEFINE:
  `snf.out, line_spectrum.dat, gauss_spectrum.dat, ir_line.gpin, ir_gauss.gpin, raman_line.gpin, raman_gauss.gpin, ir_line.eps, ir_gauss.eps, raman_line.eps, raman_gauss.eps,g98.out`*`,`
  `mopac.out`*`, xmol.XYZ.out`*

4.7 Treatment of open shell cases in TURBOMOLE single-point calculations

SNF supports the treatment of open shell cases as far as TURBOMOLE does. This causes the problem, that Infrared intensities cannot be obtained directly, since `moloch`[5] cannot calculate dipole moments for open shell molecules, which are needed for the calculation of infrared intensities. However, SNFDC can provide dipole moments for open shell cases, which are calculated in the following manner: Two closed shell `moloch` calculations are performed; in the first calculation, all α spin orbitals are treated as doubly occupied molecular orbitals while the β spin orbitals are omitted. In the second calculation, all β spin orbitals are treated as doubly occupied orbitals, while the α spin orbitals are omitted. The mean value of the dipole moments of these calculations is then used as the dipole moment of the open shell molecule.

[5] See the TURBOMOLE manual for details.

5. Snf options

Most of the options for calculations with Snf can be selected using Snfdefine. Some additional options, which are less important or only for debugging purposes, can be chosen by special keywords in the file `snf_control`. Snfdefine is an interactive program, and short explanations are provided for all menus and options. It works in essentially the same way like Define of the Turbomole package.

The aim of this chapter is to give detailed information about the meaning of these options. The menus are explained in order of appearance in Snfdefine.

Snfdefine can, of course, be applied to change program options of existing input files. The data from existing `restart` files will be saved. If only program options which have no influence on the electronic data (gradients, dipole moments, and polarizabilities), like output options or non-standard isotopes, are modified, one can re-run Snf without re-running the data collection process.

5.1 Input Options

5.1.1 Selection of input files

When Snfdefine is started, it will check for old `snf_control` files. If such a file is found, you can either accept that file as input for Snf or modify it. In the next menu, you can enter the name of an alternative input file, which will be used as a template for the current calculation.

Snfdefine furthermore allows to select the program that should be used for the single point calculations. Depending on whether it finds a mandatory file for one of the supported quantum chemical program packages (see Section 4.5.4), Snfdefine will suggest the selection of this program package for the single-point calculations. However, when Dalton is selected for the single point calculations, it is possible to read input files either in Turbomole or in Dalton format (default: Dalton; cf. Sect. 4.2.2). In this case, it is also necessary to specify the type of the single point calculations in an interactive input menu in Snfdefine.

Furthermore, the program will read the molecular input files, check the point group given in `control` (if present), and print some information about the molecule.

The following main menu contains several submenus, which allow to set further input options. Moreover, it allows to invoke the automatic symmetry detection routines of SNF.

5.1.2 Selection of isotopes

The isotope menu can be utilized to select non-standard isotopes. By default, SNFDEFINE will choose the most abundant isotopes. However, the electronic data are — within the Born-Oppenheimer–approximation — independent of the nuclear masses, such that only one `restart` file and one SNFDC run are necessary for all evaluations with different isotopes.

The selection of isotopes changes the molecular point group symmetry. Thus, that part of the calculation which depends on the nuclear masses has to be performed using the symmetry of the molecule with modified isotopes, while the generation of the symmetry redundant electronic data is done using the point group symmetry of the original molecule.

The present version of SNF requires that the orientation of the molecule is the same in both parts of the calculation. This causes the problem that not the full symmetry of the molecule with modified isotopes can be applied. E.g., the standard orientation of benzene is in the xy plane, such that the z axis is the principal axis of rotation. If one ^{12}C atom is replaced by a ^{13}C isotope, the molecular point group is C_{2v}. But the principal axis of rotation is no longer the z-axis, but maybe the y-axis, such that SNF will detect the point group C_s instead of C_{2v}.

However, the effect of the symmetry for this part of the evaluation is of minor importance. It affects the assignment of the symmetry races to the peaks in the spectra and perhaps slightly increases the computational time for the evaluation (which is of the order of some seconds to some minutes). But the full symmetry can still be used for the most time-consuming part, the calculation of the gradients, dipole moments, and polarizabilities.

5.1.3 Options for numerical derivatives

The numerical derivative menu allows to select the number of grid points used for the differentiation, and to set the `cstep` value, which characterizes the step size in [bohr].

For the number of grid points, one may choose 3, 5, or 7. Note that the 5-point central differences formula will double the computational cost of the data collecting process compared to the 3-point formula. It is usually not necessary to use the 5- (or even 7-) point central differences formula, since the accuracy of the 3-point formula is in most cases sufficient. A higher number of points is neither recommended nor implemented in the program, since the gain in accuracy can be neglected in almost all cases. If higher accuracy should be achieved, it is more helpful to apply Richardson extrapolation [23].

As will be shown in Appendix 8, the `cstep` value should be $0.01 \leq$ `cstep` ≤ 0.05, since smaller values affect the numerical stability of the differentiation, while larger values lead to a failure of the harmonic approximation. If you would like to use values beyond this range, you have to set the `CRAP_OK yes` option in the SNFDC input file `DCINPUT` (see Sect. 4.5).

5.1.4 Spectrum settings

In the spectrum settings menu one can modify the upper and lower bound of the infrared and Raman spectra. Additionally, one can choose the half-width of the peaks in shape of Gauss- or Lorentz-type functions in the Gauss/Lorentz spectra, and the number of points for their graphical representation. It is also possible to specify a scaling factor for the frequencies (see, e.g., [24] and references cited therein for scaling factors).

The Gauss spectra are plotted using the normalized line-shape function

$$g_G(\tilde{\nu}) = \frac{1}{\sqrt{2\pi\gamma^2}} \cdot \exp\left(-\frac{(\tilde{\nu}_0 - \tilde{\nu})^2}{2\gamma^2}\right) \tag{5.1}$$

with the half-width δ_G,

$$\delta_G = 2\gamma\sqrt{2\ln 2}, \tag{5.2}$$

and resembles inhomogeneous line-broadening (e.g., Doppler broadening). For the Lorentz-spectra, the normalized line-shape function

$$g_L(\tilde{\nu}) = \frac{\delta_L/(2\pi)}{(\tilde{\nu}_0 - \tilde{\nu})^2 + (\delta_L/2)^2}, \tag{5.3}$$

with the half-width δ_L, and should be used if homogeneous line broadening (e.g., natural line broadening) is dominant.

5.1.5 Thermochemical settings

The thermochemistry menu can be used to select temperature and pressure for the thermochemical output.

5.1.6 Excited-state calculation menu

This section allows to change settings for the calculation of excited states employing the TURBOMOLE `dscf/ridft` and `egrad` modules or the Dalton CASSCF module. In case of Hartree–Fock calculations, either the RPA or the CIS method can be chosen for the calculation of excited state frequencies. Furthermore, the number of excited states to be treated in `egrad`, the excited state for which the frequencies shall be calculated, and the spin state of the excited state can be selected. Note, however, the warnings mentioned in Section 4.3.

5.1.7 Raman settings

There are several options concerning the Raman spectra, which can be found in the Raman menu. First of all, the calculation of Raman intensities must be switched on by the user. Since this is very time-consuming the option is switched off as default. If the calculation of Raman intensities has been selected, you may choose between the calculation using static polarizabilities (default), or dynamic polarizabilities. If dynamic polarizabilities are chosen, one can select up to 10 different frequencies for the calculation of the Raman intensities. These must be given either as wavelengths (in [nm]) or as angular frequencies (in a.u.,i.e., hartree/$\hbar$). Default is the wavelength of an Argon ion laser (514.5 nm).

It is also possible to change the default `rpaconv` parameter for the `escf` run, which sets the convergency criterion for the residual vector (see Turbomole manual). Further options concern the type of data to be plotted in the Raman spectrum output (Raman activities, differential scattering cross sections for different experimental setups) and the scattering angle for the calculation of scattering cross sections.

5.1.8 VROA settings

The VROA menu is analogous to the Raman menu. The only difference is that the output is written to `snf.out` in in Dalton format, i.e., intensity differences and CIDs are printed by default for scattering angles of 0° (forward scattering), 90°, and 180° (backscattering). For 90° scattering, intensity differences and CIDs are given for both parallel and perpendicular relative polarization of the incoming and the outgoing laser beam. In order to plot VROA spectra calculated with, you may use the program Specplot which is distributed along with the Snf package (see `$SNF_PATH/utilities/specplot`). It reads VROA (and also Raman) intensities from `snf.out` and generates Gnuplot inputfiles according to the settings chosen by the user

For a summary of the theoretical background of VROA, see Refs. [5] and [25], and for the first accounts on theoretical and experimental aspects, Refs. [26–30]

5.2 Output options

Many output options are available in Snfdefine. Some output options are hidden to keep the output options menu clear. They can be displayed by the command "`all`". Alternatively, these options can be (de-)activated by the corresponding keyword in the file `snf_control` (see Sect. 5.3).
Options available in the standard output menu are:

`iat <on/off>` turn on/off output of information about symmetry redundant atoms

`idt <on/off>` turn on/off output of information about symmetry redundant distortions of the non-redundant atoms.

`hess <on/off>` turn on/off output of the Cartesian Hessian

`sumdeg <on/off>` turn on/off summation of intensities over degenerate modes

`gau, mopac, xmol <on/off>` turn on/off output of normal modes in format of Gaussian98, Mopac, or as XYZ format readable by Xmol

`atcontrib <on/off>` turn on/off output of atomic contributions to vibrational intensities (as `.csv` files)

`tmpcl <on/off>` turn on/off removal of temporary directories for single point calculations

`logcl <on/off>` turn on/off removal of logfiles for slave processes

`elig <PATH>` specify an alternative `ELIGIBLE` file

`bak` create backup copy of the original `mos` file (To run Snfdc you need starting MO vectors in C_1 symmetry; if you use Turbomole's Define to get these MO vectors, the original `mos` file will be overwritten.)

5.3 Additional keywords

`xmolscl <val>` sets the scaling factor for normal modes in the XMOL output file equal to `<val>`

`total electronic energy <val>` sets the total electronic energy for the optimized structure equal to `<val>`. This can be done, if the automatical procedure to read this value from the files `job.last` (TURBOMOLE) or `molecule.out` (DALTON) by SNFDEFINE failed. It is only necessary for parts of the thermochemical analysis.

`total spin <val>` analogously for the total spin value

`theta <val>` sets the scattering angle θ [deg] for the calculation of Raman scattering cross sections, if their calculation has been selected in the Raman menu (default: 90 deg).

`maxsym <on/off>` turns on/off the determination of symmetry-redundant distortions of non-redundant atoms. If `maxsym off` is chosen, only the step flags of distortions of symmetry-redundant atoms will be set equal to 1 in the `restart` file (default: on).

`dtype [option]` selects the type of numerical differentiation to be used. CAUTION: this is a testoption, which works only for two-point formulae. Possible values of `[option]` are: `centr` (central differences, default), `fward` (forward differences), `bward` (backward differences). If forward or backward differences are selected, SNF tries to read the `gradient` file for the equilibrium structure. If the gradient can be read (only Turbomole format at the moment), the numerical precision of forward/backward differences can be expected to be at least one order of magnitude less accurate than central differences (numerical errors up to 20 - 30 cm^{-1}). If the gradient cannot be read, gradient components of `0.0d0` are assumed, which usually can give rise to numerical errors of > 100 cm^{-1} (depends on the optimization thresholds). NOTE: FORWARD AND BACKWARD DIFFERENCES SHOULD NEVER BE USED IN PRODUCTION CALCULATIONS. THEY CAN ONLY GIVE A FIRST IMPRESSION OF THE VIBRATIONAL FREQUENCIES OF A MOLECULE.

`print_symmetry <on/off>` turn on/off symmetry information (default: off)

`print_dip <on/off>` turn on/off output of dipole moment derivatives (default: off) `print_pol <on/off>` turn on/off output of polarizability derivatives (default: off)

`print_force <on/off>` turn on/off output of force constant matrix (default: off)

`print_cio <on/off>` turn on/off charge decomposition according to Cioslowski [22] (default: on)

`print_zpe <on/off>` turn on/off output of zero-point kinetic energy and moments of inertia (default: on)

`print_thermo <on/off>` turn on/off detailed thermochemical output (default: on)

`print_transrot <on/off>` turn on/off output of translational and vibrational contributions (default: on)

`dalmw <val>` sets the scratch memory size in Mwords in case of Dalton calculations. Minimum is `<val>` = 800000.

6. Visualization of results

There are several tools, most of which which are neither included in nor delivered with the SNF package, which can be applied to visualize the results of the SNF calculation.

6.1 Spectra plots

The spectra plots
`ir_line_spectrum.eps`,
`ir_gauss_spectrum.eps`,
`ir_lorentz_spectrum.eps`,
`raman_line_spectrum.eps`,
`raman_gauss_spectrum.eps`,
`raman_lorentz_spectrum.eps`
are generated by `gnuplot` in EPS format. Standard tools like `gv` can be used to display these plots, others — like `convert` — can convert the format. For Raman and backscattering VROA spectra, the tool
`$SNF_PATH/utilities/specplot/specplot`
can be employed in order to generate and visualize `gnuplot` files with the desired settings (see Section F.2 for further details).

6.2 Normal modes

The following list includes a few examples of programs which can read and display normal mode output by SNF:

`molden`: can read the output files `g98.out` or `mopac.out` created by SNF. Normal mode output can be generated as a series of `.gif` files, which can be animated by tools like `animate` or `xanim`. They can also be converted to animated GIF using programs like `whirlgif`. Another possible output are Postscript files, in which the normal modes are indicated by vectors.
`molden` can be obtained via:

`http://www.caos.kun.nl/~schaft/molden/molden.html` `xmol`: can read the `xmol.XYZ.out` files created by Snf. Normal modes can be displayed and exported (Postscript) as vectors.

`jmol`: can read the Snf output files `g98.out` and `xmol.XYZ.out`. Output can be generated using vectors (JPEG, PNG, etc.). Additionally, it is possible to create a series of pictures (e.g., PNG), which can then be animated using `animate` or `xanim`. `jmol` can be obtained via:

`http://jmol.sourceforge.net/`

7. Thermochemistry

Thermodynamical functions are calculated according to the statistical thermodynamics of a canonical ensemble of an ideal gas. For rotations and translations, contributions of 3/2 RT to the enthalpy are assumed. Rotational contributions to the entropy are treated classically. Electronic contributions are taken into account as spin-only values to the entropy in UHF/UKS cases.

The calculations are based on the following relationships[1]:

- **Enthalpy**

  o Translation: $H^t = 3/2RT$

  o Rotation: $H^r = 3/2RT$

  o Vibration:

$$H^v = RT \sum_i \frac{h\nu_i}{kT} \left\{ \frac{1}{2} + \left[\exp\left(\frac{h\nu_i}{kT}\right) - 1 \right]^{-1} \right\}$$

  NB: ZPVE is included.

  o Electronical contributions: none

- **Entropy and partition functions**

  o Translation

$$S^t = R(ln(z^t) + 5/2)\ ,\ z^t = (kT)^{5/2}/(ph^3)\,(2\pi m)^{3/2}$$

  m is the molecular mass.

  o Rotation

$$S^r = R(ln(z^r) + 3/2)\ ,\ z^r = \frac{\sqrt{\pi}}{\sigma}\sqrt{\frac{T^3}{T_1 T_2 T_3}}$$

[1] See [31] for details.

$$T_i = \frac{h^2}{8k\pi^2 J_i}$$

σ is the symmetry number and J_i is the moment of inertia.

o Vibration

$$S^{vib} = \sum_i \frac{h\nu_i}{kT}\left[\exp\left(\frac{h\nu_i}{kT}\right) - 1\right]^{-1} - \ln\left[1 - \exp\left(-\frac{h\nu_i}{kT}\right)\right]$$

Partition function with reference to the bottom of the potential well:

$$z^{vib,bot} = \prod_i \exp\left(-\frac{h\nu_i}{2kT}\right)\left[1 - \exp\left(-\frac{h\nu_i}{kT}\right)\right]^{-1}$$

Partition function with reference to the vibrational ground state energy:

$$z^{vib,v=0} = \prod_i \left[1 - \exp\left(-\frac{h\nu_i}{kT}\right)\right]^{-1}$$

o Electronical contributions

$$S^e = R\ln(g)\ ;\ g = \text{spin degeneracy of the ground state}$$

(assumed as spin-only values; excited electronic states must be sufficiently higher in energy)

- **Heat capacities**

o Translation: $c_v^t = 3/2R$

o Rotation:
non-linear molecules: $c_v^r = 3/2R$
linear molecules: $c_v^r = R$

o Vibration

$$c_v^{vib} = R\sum_i \left(\frac{h\nu_i}{kT}\right)^2 \exp\left(\frac{h\nu_i}{kT}\right)\left[\exp\left(h\nu_i kT\right) - 1\right]^{-2}$$

8. Parameter studies: step size, basis sets, SCF convergence, and grid type

This chapter presents investigations by C. Kind concerning the accuracy and numerical stability of the calculations for different values of the calculation parameters.

The step size for the numerical differentiation (`cstep`), the energy convergency criterion for the TURBOMOLE calculations (`scfconv`), the quality of the grid for the density functional calculation (`GRID`), the influence of the RI-approximation, and the basis set dependence were taken into account in these analyses.

The results presented here shall help to explain our choices of the default parameters for the calculations, and to provide a basis for error estimations in calculations with SNF.

8.1 Dependence on the step size (`cstep`)

As has already been mentioned in Sect. 5.1.3, it is dangerous to choose `cstep` values which are too small or too large. If the value is too small, there might occur instabilities in performing the central differences during the differentiation procedure. If it is too large, the harmonic approximation might fail. The following calculations have been carried out for *trans*-diazene using DFT/BP86 to analyse the effect of the `cstep` parameter.

- SVP/RI using TURBOMOLE default values (SVP default: SD)
- SVP/RI using `scfconv=8` and `GRID=5` (SVP accurate: SA)
- TZVP/RI using TURBOMOLE default values (TZVP default: TD)
- TZVP/RI using `scfconv=8` and `GRID=5` (TZVP accurate: TA)
- SVP calculation using GAUSSIAN98 [17] (SVP analyt.)
- TZVP calculation using GAUSSIAN98 (TZVP analyt.)

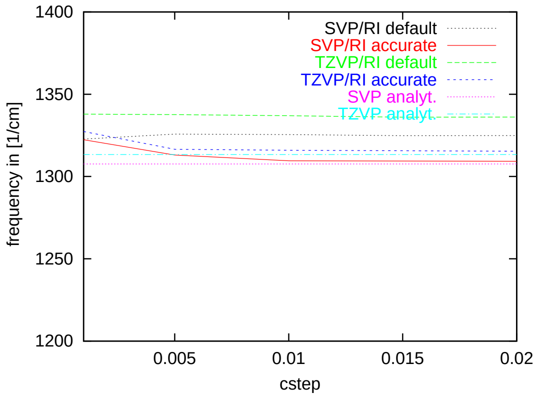
For each type calculations have been performed using the `cstep` values 0.001, 0.005, 0.01, 0,015, and 0.02.

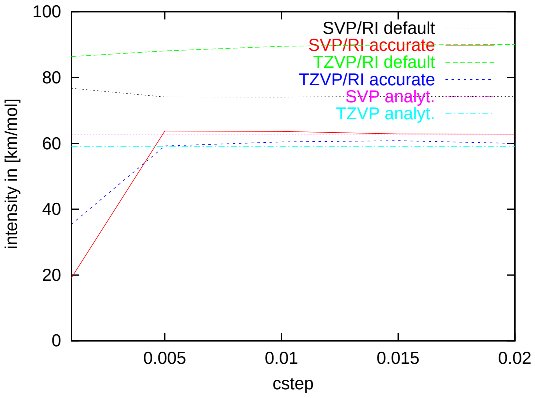
Gaussian98 calculations should help to compare our numerical results to analytically determined frequencies, which, of course, are independent of the step size.

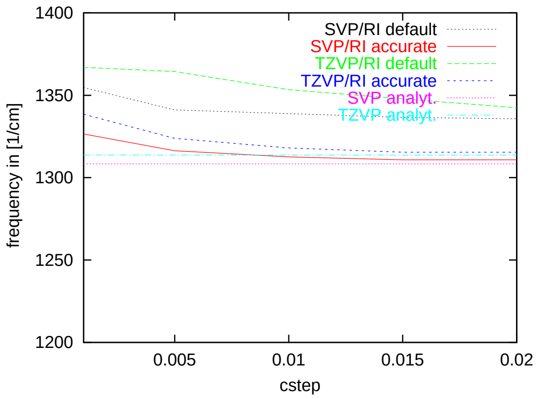
The following tables show the mean and maximum error for different `cstep` values compared to the analytical frequencies. All frequencies are given as wavenumbers in [1/cm], and the infrared intensities, which can also be found in the tables, are given as absorption coefficients in [km/mol].

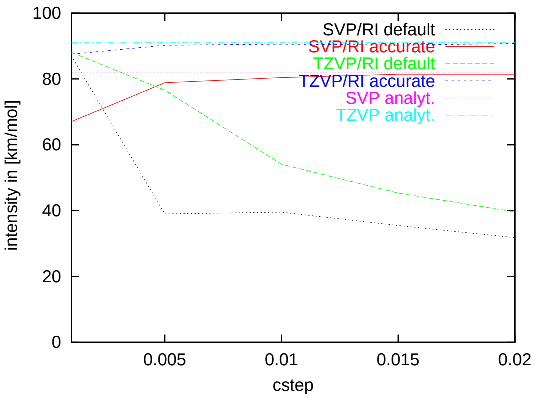
	mean error							
	IR intensities				frequencies			
cstep	SD	SA	TD	TA	SD	SA	TD	TA
0.001	13.5	33.1	12.6	5.9	53.0	25.7	60.5	22.1
0.005	23.7	1.0	13.7	0.3	42.0	7.0	52.9	5.7
0.010	25.7	0.6	28.4	0.4	35.8	4.6	42.7	5.0
0.015	24.1	0.3	29.2	0.5	35.4	4.0	40.7	4.1
0.020	26.0	0.3	32.1	0.3	31.3	4.1	34.3	3.9

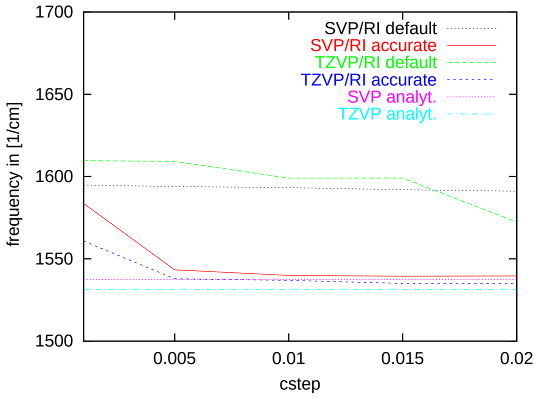
	maximum error							
	IR intensities				frequencies			
cstep	SD	SA	TD	TA	SD	SA	TD	TA
0.001	54.9	72.3	42.1	23.7	88.5	46.3	81.5	39.7
0.005	87.4	3.2	38.6	0.9	86.8	9.0	92.2	10.1
0.010	99.9	1.7	102.9	1.3	84.6	7.2	87.3	8.3
0.015	86.1	1.0	98.4	1.7	82.6	7.3	82.6	8.5
0.020	93.9	1.0	110.3	0.9	79.5	7.4	79.4	7.9

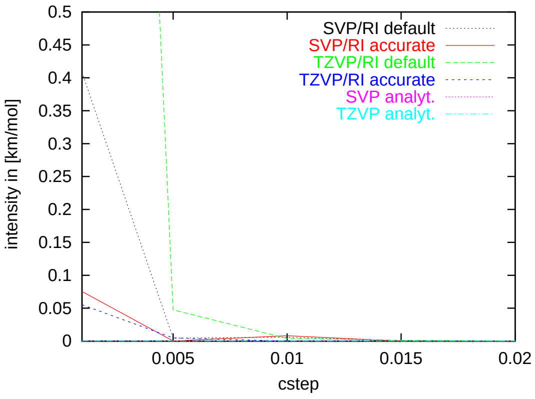

Mode 7: frequency

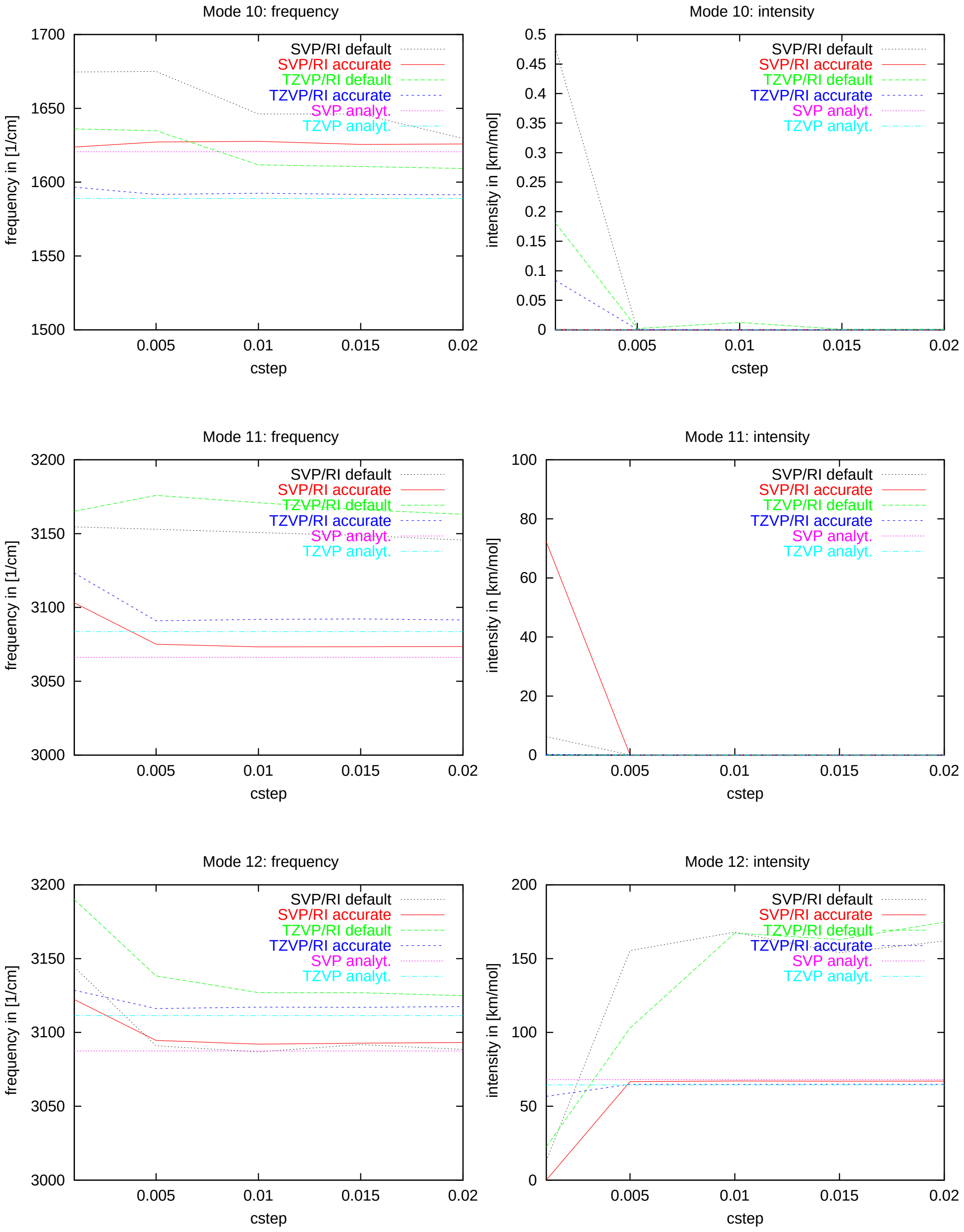

Mode 7: intensity


Mode 8: frequency


Mode 8: intensity


Mode 9: frequency


Mode 9: intensity



Since the numerical error of these calculations (3-point central differences formula) is proportional to (`cstep`)^2, the step size should in principle be chosen as small as possible. The above results demonstrate, that the smallest possible `cstep`-value which ensures numerical stability is `cstep=0.01`.

8.2 Influence of the SCF parameters scfonv and GRID

Of major importance for the numerical derivatives is the accuracy of the single point data. Two of the most important parameters for DFT Turbomole calculations have been investigated to check their influence on the frequencies: the SCF convergency criterion scfconv and the grid characterization parameter GRID[1].

Calculations have been carried out for *trans*-diazene using a TZVP basis set and cstep=0.01.

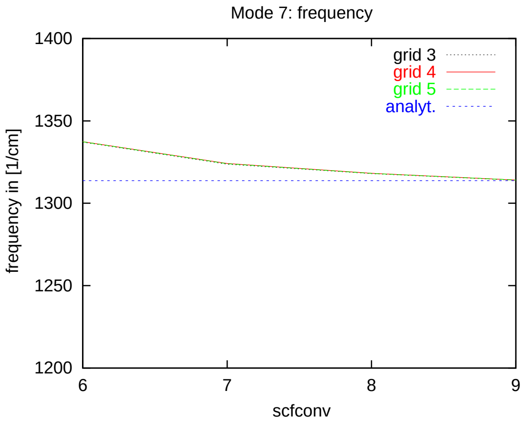
Mean and maximum errors of the numerical frequencies compared to the analytical ones are shown in the following two tables for different values of the parameters scfconv and GRID.

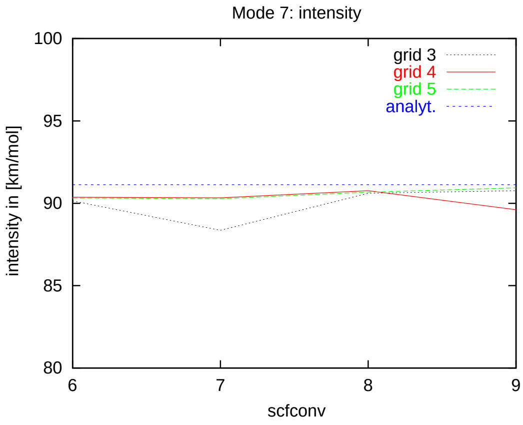
	mean error					
	IR intensities			frequencies		
scfconv	grid 3	grid 4	grid 5	grid 3	grid 4	grid 5
6	23.8	24.3	24.4	40.6	40.4	40.3
7	2.0	1.2	1.1	12.0	14.3	14.2
8	0.4	0.3	0.4	5.2	4.9	4.9
9	0.5	0.9	0.5	2.8	2.7	2.8

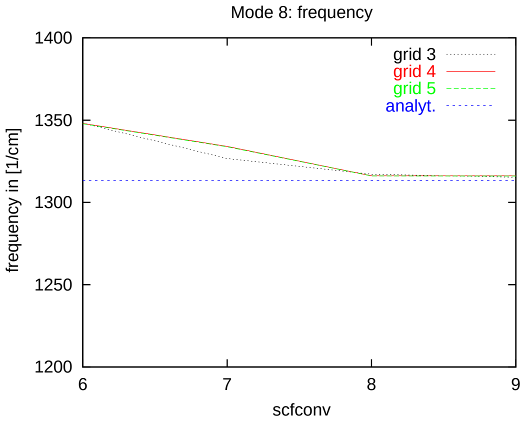
	maximum error					
	IR intensities			frequencies		
scfconv	grid 3	grid 4	grid 5	grid 3	grid 4	grid 5
6	128.6	131.2	131.5	85.4	85.2	85.3
7	5.3	4.6	4.5	16.6	20.7	20.4
8	1.4	1.4	1.3	7.2	7.9	7.7
9	1.6	2.6	1.4	4.9	4.3	4.5

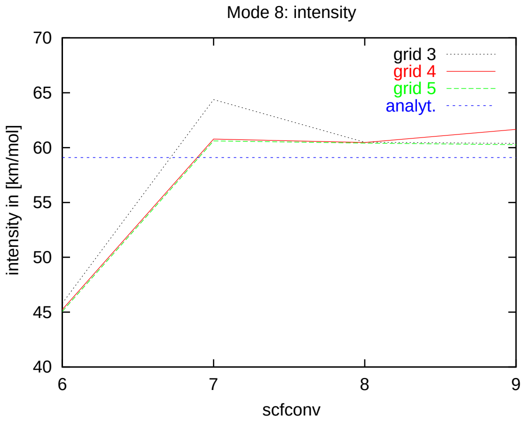
It can be concluded from these calculations, that only a simple grid (GRID = 3) is necessary. However, a scfconv value of 8 or higher is strongly recommended.

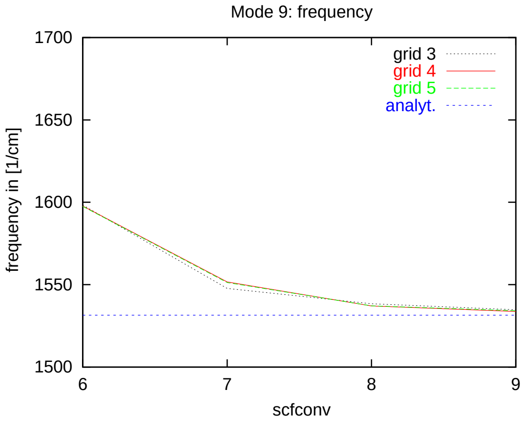
[1] For the meaning of the GRID parameter see the Turbomole manual.

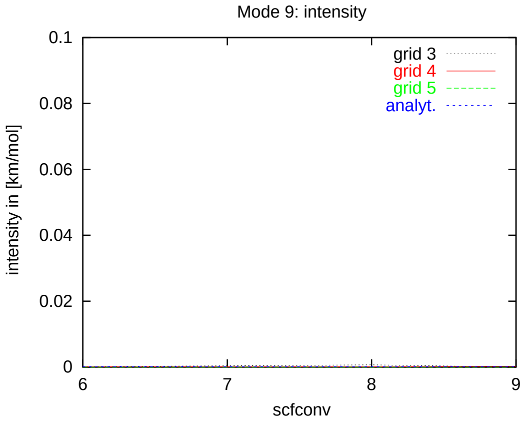

Mode 7: frequency

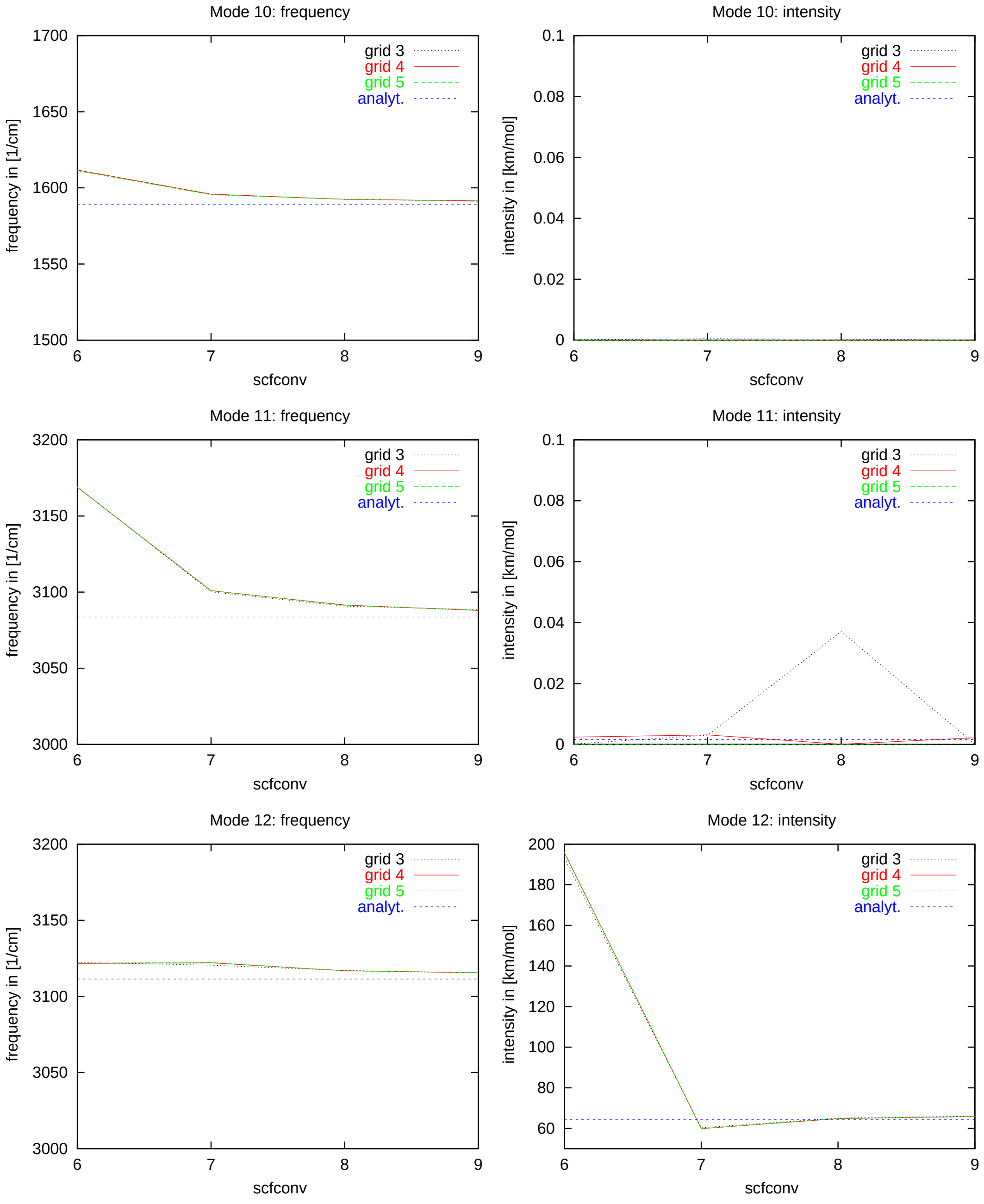

Mode 7: intensity


Mode 8: frequency


Mode 8: intensity


Mode 9: frequency


Mode 9: intensity



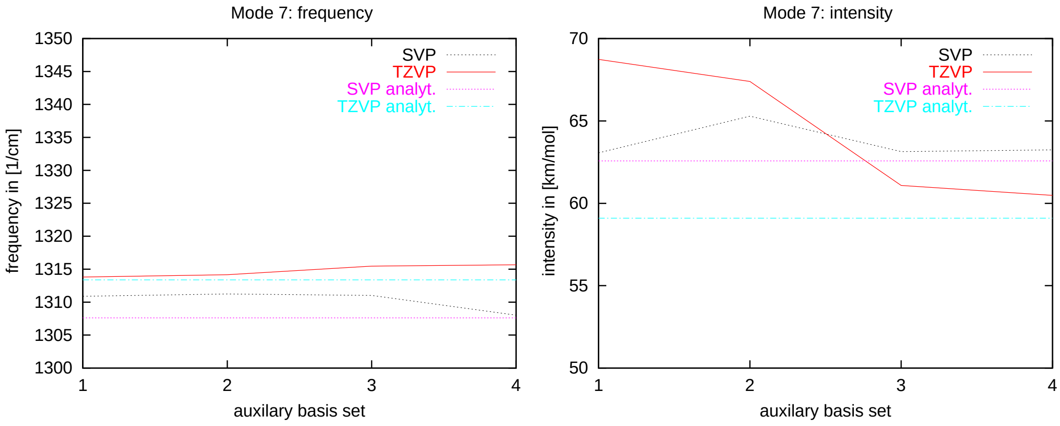
8.3 Influence of the RI-approximation

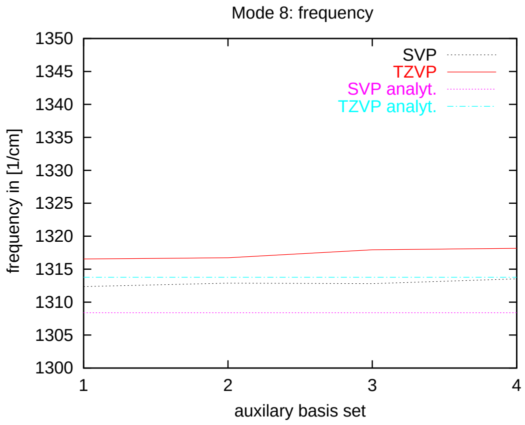
The influence of the RI-approximation [32,33] has been analyzed by varying the auxiliary basis set. Four different auxiliary basis sets have been investigated; however, for Nitrogen the auxiliary basis set SV is the same as SVP and TZVP is the same as TZVPP. For Hydrogen, all auxiliary basis sets are different.

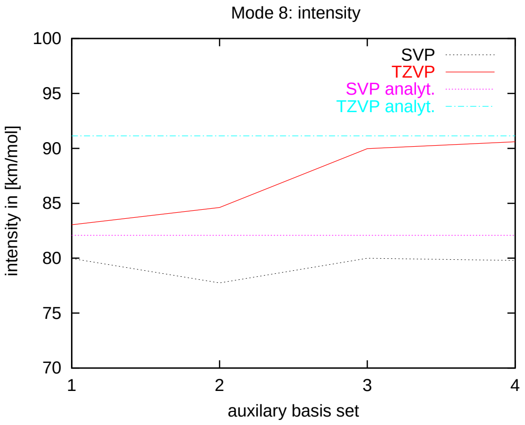
Mean and maximum errors (taking analytical frequencies without RI approximation as references) for the different auxiliary basis sets are listed in the tables below.

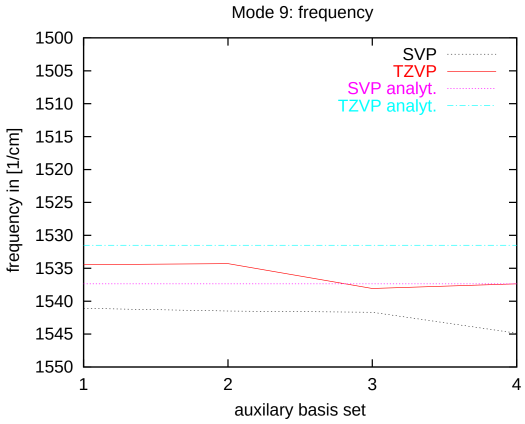
		mean error			
		IR intensities		frequencies	
No.	Auxbasis	SVP	TZVP	SVP	TZVP
1	SV	0.5	3.0	3.9	3.1
2	SVP	1.3	2.6	4.4	3.3
3	TZVP	0.5	0.6	4.5	4.3
4	TZVPP	0.6	0.4	4.8	4.9

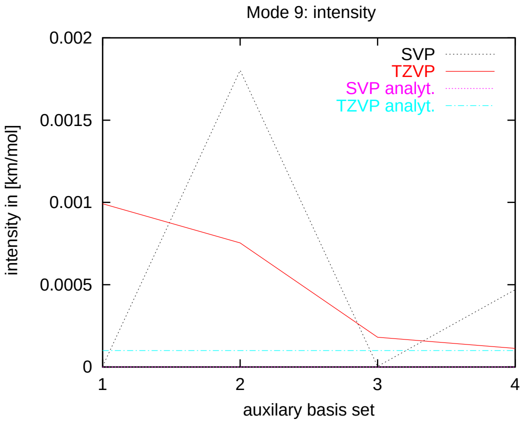
		maximum error			
		IR intensities		frequencies	
No.	Auxbasis	SVP	TZVP	SVP	TZVP
1	SV	2.1	9.6	6.9	10.9
2	SVP	4.3	8.3	8.0	11.6
3	TZVP	2.1	2.0	9.1	8.4
4	TZVPP	2.3	1.4	10.2	12.2

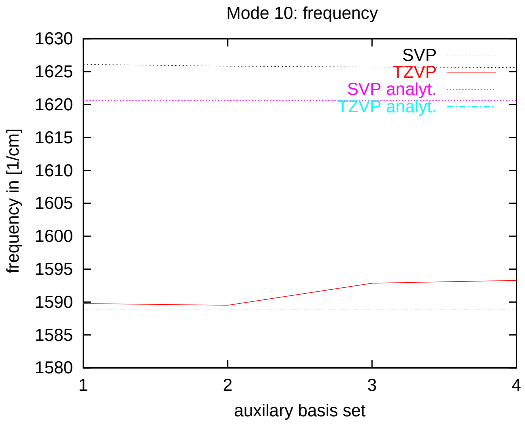


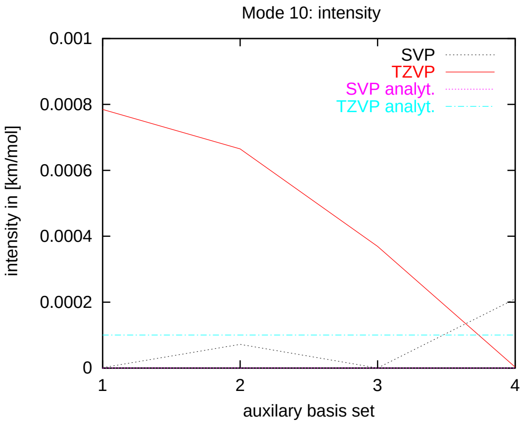

Mode 8: frequency

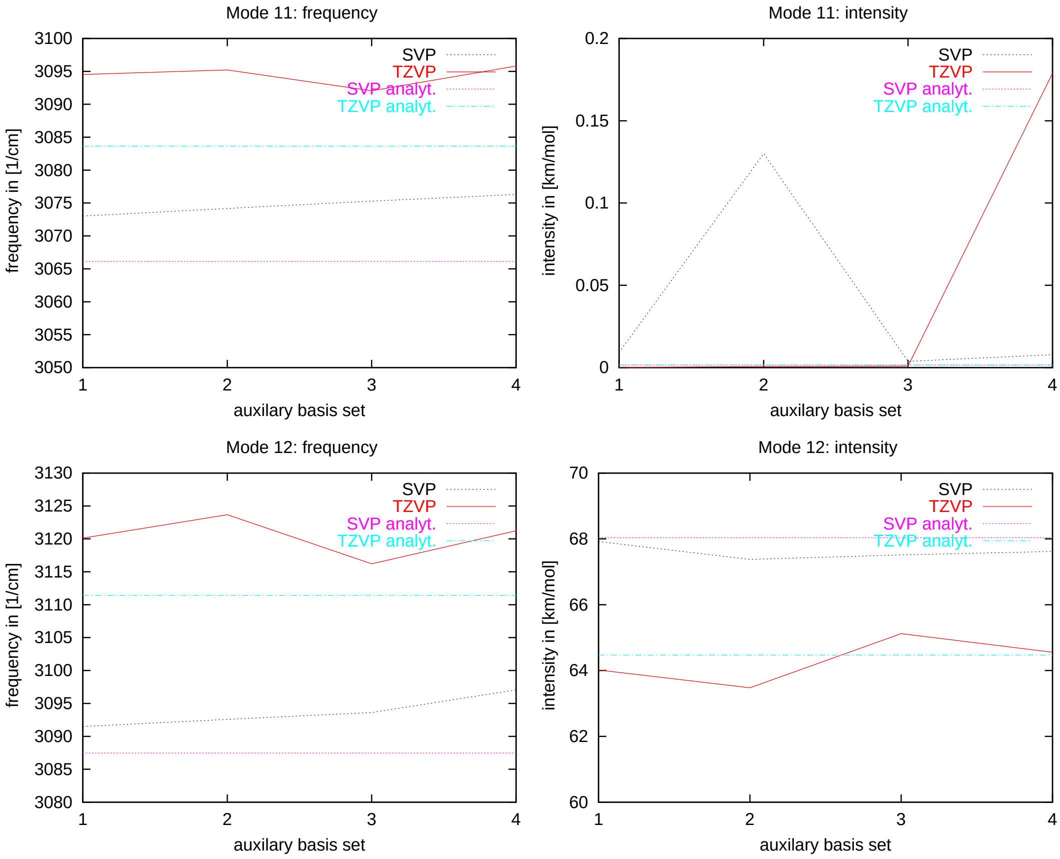

Mode 8: intensity


Mode 9: frequency


Mode 9: intensity


Mode 10: frequency


Mode 10: intensity



As can be seen from these results, the effects of the auxiliary basis sets are very small.

8.4 Basis set dependencies

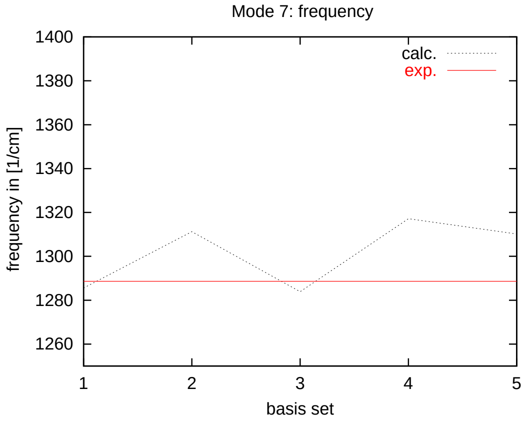
TURBOMOLE supplies 5 different standard split-valence basis sets. Contraction schemes for Hydrogen and Nitrogen are as follows:

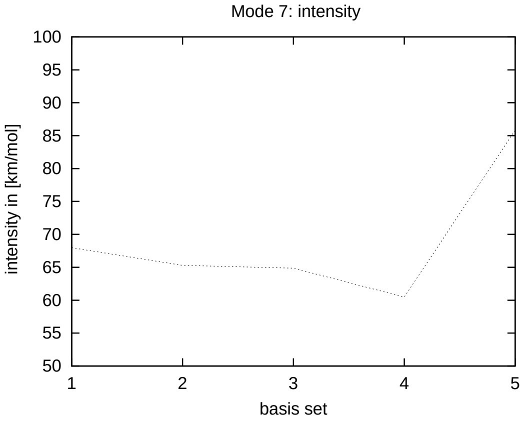
No.	Basis	H	N
1	SV	[4s,2s]	[7s4p,3s2p]
2	SVP	[4s1p,2s1p]	[7s4p1d,3s2p1d]
3	TZV	[5s,3s]	[11s6p,5s,3p]
4	TZVP	[5s1p,3s1p]	[11s6p1d,5s3p1d]
5	TZVPP	[5s3p,3s3p]	[11s6p2d1f,5s3p2d1f]

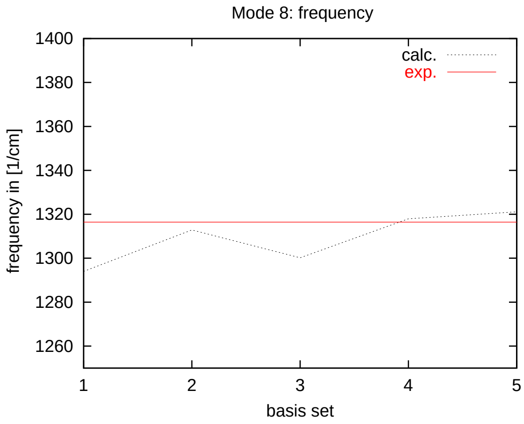
The following table contains maximum and mean errors compared to experimental frequencies (experimental IR intensities could not be obtained). Experimental frequencies for IR active modes have been taken from Hallin *et al.* 1981 [34] and from Hegelund and Burger 1994 [35] (gas phase). Frequencies of IR inactive modes have been taken from Raman experiments by Bondybey and Nibler 1973 [36] (N_2 matrix).

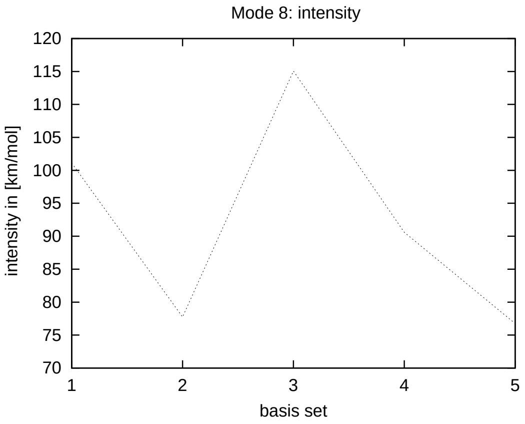
No.	Basis	max. error	mean error
1	SV	115.78	49.02
2	SVP	46.12	27.17
3	TZV	134.52	57.69
4	TZVP	29.47	14.89
5	TZVPP	21.52	10.73

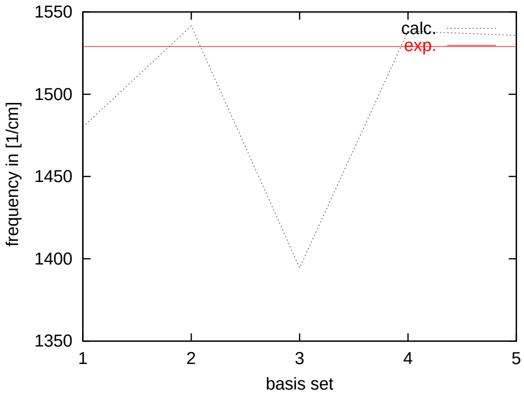
This indicates that polarization functions are mandatory in order to obtain satisfactory results; otherwise, errors up to 150 cm^{-1} may occur for the frequencies (and up to 20% for the intensities). The mean error for a TZVP basis set is smaller than 15 cm^{-1}, such that this basis set should be preferred to obtain accurate results.

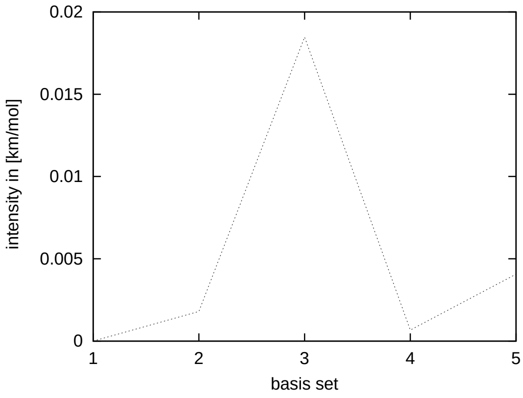


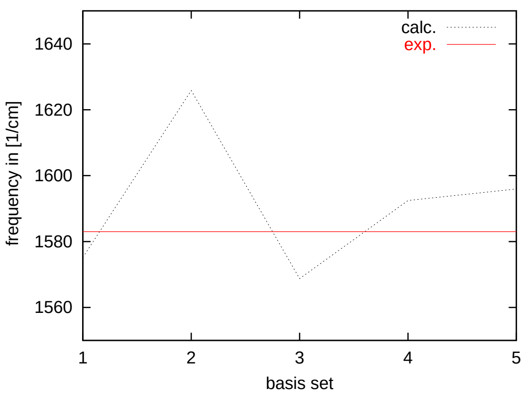


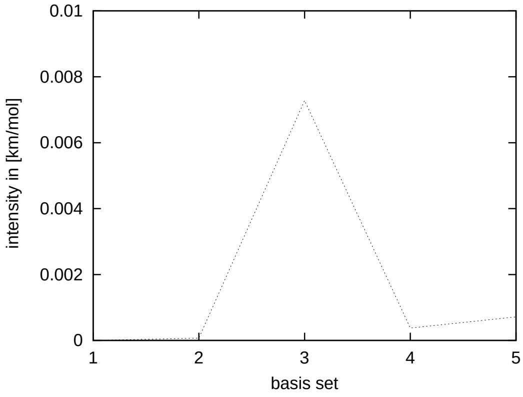


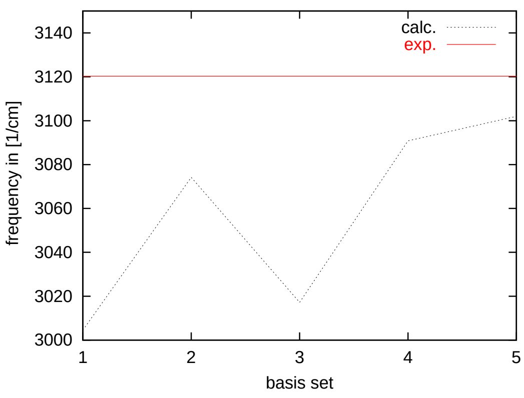


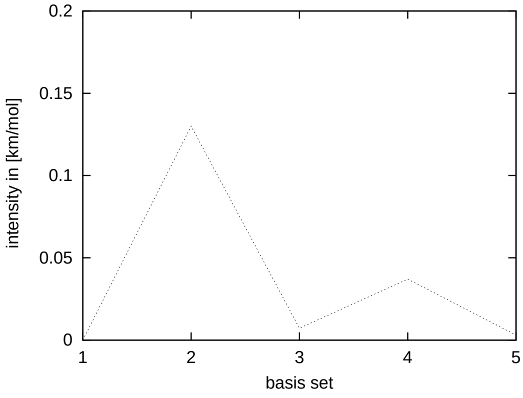

Mode 9: frequency

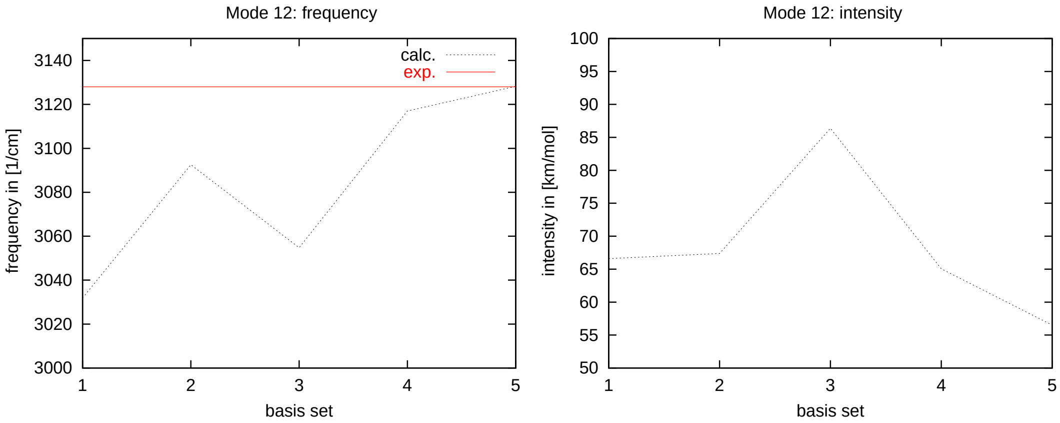

Mode 9: intensity


Mode 10: frequency


Mode 10: intensity


Mode 11: frequency


Mode 11: intensity



8.5 Raman intensities

For an analysis of the influence of the choice of basis set size and quantum chemical method on Raman intensities we refer to the extensive analyses given in [1] and [37].

A. Program history

The first version of the Fortran 90 program SNF was developed by C. Kind and M. Reiher at the University of Erlangen-Nuremberg in 1999. It used the program NUMFREQ [38] as the data collector.

The original Fortran 77 version of NUMFREQ was written by S. Grimme at the University of Bonn and contains contributions from C. Marian and M. Gastreich. This PVM parallelized version of the program has been largely extended in 1999 by B. A. Hess who also added the MPI, the LAM/MPI, and the MPICH versions of the program. This extensive rewriting of the code, which yielded a program that can now solely be used for the general parallelized collection of data (gradients, etc.), lead to the introduction of SNFDC (DC = data collector).

Substantial parts of the program SNF are based on subroutines for the utilization of projection operators and representation matrices, which have been written by C. Kind at the University of Cologne (1995/96). In winter 2000/2001 M. Reiher and C. Kind modified the program and included thermochemistry and Raman intensities based on static polarizabilities. In 2001 J. Neugebauer revised the existing code and included the cubic and icosahedral point groups as well as the calculation of Raman intensities based on dynamic polarizabilities. Further extensions by J. Neugebauer concern the treatment of dummy atoms and linear molecules, the calculation of isotope effects, and several output options.

Modifications after Oct. 2001 are listed below according to the version numbers.

V1.4.0 Implementation of the DALTON module. SNFDEFINE can now prepare DALTON calculations using the methods HF(SCF), MP2, CCS, CCSD, and CC2 if TURBOMOLE input files are available (JN, 11/08/2001).

V1.4.1 In open-shell cases, total spin expectation values are read to calculate electronic (spin-only) contributions to the entropy etc.; scattering cross sections can be calculated for any scattering angle (MR, JN, 11/22/2001).

V1.4.2 Symmetry redundant distortions of non-redundant atoms can be omitted (JN, 11/27/2001)

V1.4.3 New ESCF versions (binaries denoted as `escf.intel` and `escf.athlon`) can be applied for the calculation of polarizabilities. They allow to use the RI-approximation in DFT calculations (JN, 12/16/2001).

V1.4.4 The data for the 5-point central-differences formula can now be collected with SNFDC (JN, 12/17/2001).

V2.0.0 The TURBOMOLE version of SNF is available as a package containing the interactive input program SNFDEFINE, the data collector SNFDC, the evaluation program SNF, and all scripts which are necessary to run these programs (JN, 12/18/2001).

V2.0.1 The scripts `choose_nodes` and `mkdcinput` are executed by `snf_define` for easier preparation of SNFDC input files (JN, 01/04/2002).

V2.1.0 SNFDC now supports the DALTON program modes (JN, 01/13/2002).

V2.1.1 Minor error corrections in the MPICH version of SNFDC. SNFDC now prepares input files and runscripts for the MPI versions of SNFDC. SNF and SNFDEFINE fulfill Fortran 95 standard. Infrared intensities are available for TURBOMOLE MP2/RI-MP2-calculations. Logfiles for slave processes and temporary directories for single point calculations may be removed automatically (BAH, JN, CK, 01/24/2002).

V2.1.2 Configuration script makes use of TURBOMOLE configuration files; optional configuration using the `PARM` file (BAH, 01/28/2002).

V2.1.3 MCSCF Raman and infrared intensities are available using DALTON. Data from old restart files can now be used if calculations shall be restarted using a higher number of grid points for the numerical derivatives (JN, 02/06/2002).

V2.1.4 Minor corrections in the LAM/MPI version of SNFDC and in module `snf_control`. SNF is now distributed under the terms of the GNU General Public License; see file `LICENSE` in the SNF installation directory (JN, BAH, MR, 02/13/2002).

V2.1.5 Setting of `rpaconv` parameter possible in SNFDEFINE; 7-point central differences formula implemented (JN, 07/03/2002).

V2.1.6 Output of atomic contributions to vibrational intensities is possible via keyword `$atcontrib` (CK, 05/02/2002).

V2.2.0 SNF contains routines for the preparation of DALTON input files, which are invoked if BASED is not available. The molecular charge is determined via occupation numbers in `control` SNF supports the calculation of HF polarizabilities using DALTON's `RESPONSE` module instead of the `ABACUS` module. SNF supports the calculation of Raman intensities from SOPPA polarizabilities in combination with MP2 frequencies and infrared intensities. `plotcperu` keyword available, replaced by `rptype <str>` keyword in V2.2.2 (JN, 09/19/2002).

V2.2.1 Implementation of orientation independent symmetry detection routines. SNF now allows to use DALTON input files as well as TURBOMOLE input files for the preparation of DALTON single point calculations. BASED is no longer invoked, neither by SNFDEFINE, by SNF, nor by SNFDC (JN, 09/26/2002).

V2.2.2 Individual `ELIGIBLE` files can be specified for the parallel calculation. Improved error handling for SNFDC. New options for the Raman spectrum output are available: all kinds of differential cross sections calculated by the program can be plotted (keyword `rptype <str>`, can be set via SNFDEFINE). Empirical scaling factors can be applied for the evaluation of the data. (JN, CK, 11/12/2002). Forward and backward differences are implemented to check the precision of the numerical differentiation (JN, 12/10/2002).

V2.3.0 SNF can calculate vibrational frequencies and infrared spectra for excited states via the egrad module of the TURBOMOLE V5.6 package, using either RPA or CIS (only for HF) calculations for the excited state. SNFDEFINE was extended accordingly to allow an easy setup of excited state calculations. Gauss- and Lorentz-type spectra can be plotted (JN, 01/20/2003).

V3.0.0 The program has been re-structured in order to change its former TURBOMOLE-specific character to a general structure that allows for an easy interfacing of new quantum chemical programs (JN, spring 2003).

V3.1.0 The program was extended to allow for the calculation of Vibrational Raman Optical activity (VROA) spectra. These are now possible using the DALTON package. For the evaluation of the data (derivatives of London and non-London G tensor and of A tensors), parts of a subroutine from the ABACUS part of DALTON have been used with permission of K. Ruud (MR, June 2003).

3.3.0 Interfaces to the quantum chemical program packages Gaussian and Adf have been included (CH, 2005).

4.0.0 Snf now features a basic Molpro interface (for MCSCF calculations) and an RICC2 interface to Turbomole. A serial version of Snfdc is available. Several changes have been made to update the configuration machinery and to simplify the installation process. A test suite and a tool for plotting in particular VROA spectra (`specplot`) have been added (JN, CH, SL, 01/2007).

B. Supported point groups

Snf supports almost every molecular point group, with the exceptions of $C_{\infty v}$ and $D_{\infty h}$. In version 1.0 and all later versions, the following 45 point groups are implemented:

$C_1, C_i, C_s,$
$C_2, C_3, C_4, C_5, C_6, C_7, C_8$
$C_{2v}, C_{3v}, C_{4v}, C_{5v}, C_{6v},$
$C_{2h}, C_{3h}, C_{4h}, C_{5h}, C_{6h},$
$D_2, D_3, D_4, D_5, D_6,$
$D_{2h}, D_{3h}, D_{4h}, D_{5h}, D_{6h},$
$D_{2d}, D_{3d}, D_{4d}, D_{5d}, D_{6d},$
$S_4, S_6, S_8,$
$O, O_h, T, T_d, T_h, I, I_h$

C. Program modes

The program modes available in Snf are given in the table below, which also shows the corresponding program number (`progno`) and calculation number (`calcno`) which are stored in the `restart` file.

Table C.1: Program modes available in Snf.

`progno`	`calcno`	program package	type of calc.	note
1	1	Turbomole	SCF/DFT	
1	2	Turbomole	RI-DFT	
1	3	Turbomole	RI-MP2	no Raman
1	4	Turbomole	MP2	no Raman
1	5	Turbomole	EGRAD	no Raman
1	6	Turbomole	EGRAD/RI	no Raman
1	7	Turbomole	DSCF/RICC2	no Raman
2	1	Dalton	HF	
2	2	Dalton	MP2	no Raman
2	3	Dalton	CC2	
2	4	Dalton	CCS	
2	5	Dalton	CCSD	
2	6	Dalton	MCSCF	
2	7	Dalton	SCF/TDHF	
2	8	Dalton	MP2/SOPPA	
2	9	Dalton	DFT	
3	1	Gaussian	Gaussian98	no frequency-dependent Raman intensities
3	2	Gaussian	Gaussian03	
4	1	Adf	(all)	
5	1	Molpro	MCSCF	no Raman

D. Supported compilers

The program SNFDC is written in standard Fortran 77 and can be compiled with the GNU Fortran 77 compiler `g77` and compilers compatible with this standard (e.g., the PORTLAND GROUP compilers `pgf77, pgf90`, the INTEL compiler `ifc` etc.).

SNF is written in Fortran 90. Some default compilers that can be used to compile SNF are the following:

No.	Compiler	platform
1	INTEL Fortran Compiler for Linux (`ifc, ifort`)	x86, x86_64
2	PORTLAND GROUP High Performance Fortran	x86, x86_64
3	GNU gfortran	x86, x86_64

Some default paths are set in the file src/aux/PARM to the libraries of certain compilers, so that `$MKLLIB` should be the path to the `mkl` libraries for the INTEL compiler `ifc`, or `$PFDIR/../lib` should be the path to the libraries for the LAHEY/FUJITSU compiler `lf95` (`$PFDIR` should be the path to the `lf95` binary).

Note that you may select different compilers for SNF/SNFDEFINE and SNFDC by executing the `configure` and `make` commands separately in the directories `src/snf` and `src/snfdc`. For the compilation of SNFDC it is, however, necessary to compile the preprocessor `delrem` in `src/aux/delrem` previously.

In case of MPI environments, it may be necessary to use the wrapper `mpif90` provided by the MPICH package to compile SNFDC (by using the configure-option `--with-FC=mpif90`).

E. Elimination of translational and rotational modes

In this section, two methods will be explained to eliminate the contributions of translational and rotational modes to the Hessian matrix.

The first method determines the translational and rotational fractions of each mode after diagonalization of the Hessian, such that the 6 modes with the largest percentages of non-vibrational contributions are neglected. The second method explicitly removes the contributions of pure non-vibrational modes before diagonalizing the Hessian. For details of the notation utilized here, see [1].

E.1 Determination of translational and rotational fractions

Every pure vibrational mode must preserve the center of mass, i.e.,

$$\Delta M = \left| \sum_{i=1}^{N} \mathbf{R}_i^{(c)} m_i \right| = 0 \ , \tag{E.1}$$

where $\mathbf{R}_i^{(c)}$ is the displacement vector of atom i in Cartesian coordinates in this normal mode. The change ΔM_{trans} for a pure translational normal mode may be defined by a normalized motion of all atoms along a selected direction $\mathbf{R}_{\text{trans}}^{(m)}$, where the superscript indicates the mass-weighting. The normalization condition is

$$\sum_{i=1}^{N} |\mathbf{R}_{i,\text{trans}}^{(m)}|^2 = \sum_{i=1}^{N} |\mathbf{R}_{i,\text{trans}}^{(c)}|^2 m_i = |\mathbf{R}_{\text{trans}}^{(c)}|^2 M_{\text{total}} = 1 \ , \tag{E.2}$$

where we made use of the fact that all displacements $\mathbf{R}_{i,\text{trans}}^{(c)}$ are equal by definition. M_{total} is the total mass of the molecule. The translational amplitude is thus

$$|\mathbf{R}_{\text{trans}}^{(c)}| = \frac{1}{\sqrt{M_{\text{total}}}} \quad . \tag{E.3}$$

With the above equation, we obtain for ΔM_{trans}

$$\Delta M_{\text{trans}} = \left|\sum_{i=1}^{N} \mathbf{R}_{\text{trans}}^{(c)} m_i\right| = \sum_{i=1}^{N} \frac{m_i}{\sqrt{M_{\text{total}}}} = \sqrt{M_{\text{total}}} \ , \tag{E.4}$$

such that we get for the translational percentage

$$c_{trans} = \frac{\Delta M}{\Delta M_{trans}} = \frac{\Delta M}{\sqrt{M_{total}}} \ . \tag{E.5}$$

For the rotational percentage, an analogous procedure can be applied. In this case, the resulting angular momenta $\Delta\mathbf{L}$ and $\Delta\mathbf{L}_{\text{rot}}$ have to be considered,

$$\Delta\mathbf{L} = \sum_{i=1}^{N} m_i(\mathbf{r}_i \times \mathbf{R}_i^{(c)}) \ , \tag{E.6}$$

where $\mathbf{r}_i$ are the Cartesian coordinates of atom i in the center-of-mass coordinate system, as they may be obtained from the input file. The axes of rotation are per definition the x-, y-, and z-axis in this "laboratory coordinate system". Note that the above formula only describes angular momenta resulting from differential displacements of the atoms. But since this also holds for the determination of the normal modes, i.e., since rotational normal modes are only pure rotational for differential motions, it is legitimate to use the above definition in this framework.

The resulting angular momentum $\Delta\mathbf{L}_{\text{rot}}$ for a pure rotation about the axis specified by the vector $\Delta\mathbf{L}$ for the normal mode under consideration is normalized as follows:

$$\sum_{i=1}^{N} |\sqrt{m_i}(\mathbf{r}_i \times \vec{\omega}_{\text{rot}})|^2 = \sum_{i=1}^{N} m_i|\mathbf{r}_i^{\perp}| \cdot |\vec{\omega}_{\text{rot}}|^2 = 1 \ . \tag{E.7}$$

Here, $\vec{\omega}_{\text{rot}}$ is the angular velocity around the specified axis, which is identical for all atoms in this pure rotation, and $\mathbf{r}_i^{\perp}$ is the component of $\mathbf{r}_i$ perpendicular to this axis. The absolute value $|\vec{\omega}_{\text{rot}}|$ is then given as

$$|\vec{\omega}_{\text{rot}}| = \frac{1}{\sqrt{\sum_{i=1}^{N} |\mathbf{r}_i^{\perp}|^2 m_i}} \ . \tag{E.8}$$

The angular velocity is related to the angular momentum by the moment of inertia I,

$$\vec{\omega}_i = I^{-1}\Delta\mathbf{L}_i \ . \tag{E.9}$$

The final result for the rotational percentage to a normal mode is

$$c_{\text{rot}} = \frac{\Delta \mathbf{L}_i}{\Delta \mathbf{L}_{\text{rot}}} = \frac{|\vec{\omega}_i|}{|\vec{\omega}_{rot}|} \; . \tag{E.10}$$

The values c_{trans} and c_{rot} may then be used to eliminate the modes with the three largest percentages of rotational and translational contributions.

E.2 Elimination of non-vibrational contributions by means of projection operators

In the second method, pure translational and rotational modes, i.e., motions along the coordinates $\mathbf{R}^{(m)}_{\text{trans}}$ and $\mathbf{R}^{(m)}_{\text{rot}}$, are constructed and projected off the Hessian.

For this purpose, the projection operators

$$\mathbf{P}^{(m)}_{\text{trans}} = 1^{(m)} - \sum_{\text{trans}}^{3} \mathbf{R}^{(m)}_{\text{trans}} \mathbf{R}^{(m)\dagger}_{\text{trans}} \; , \tag{E.11}$$

$$\mathbf{P}^{(m)}_{\text{rot}} = 1^{(m)} - \sum_{\text{rot}}^{3} \mathbf{R}^{(m)}_{\text{rot}} \mathbf{R}^{(m)\dagger}_{\text{rot}} \; , \tag{E.12}$$

are applied on the matrices[1] $\mathbf{S}^{(m)}_{\mu}$ and the resulting matrix is orthonormalized with an appropriate matrix $\mathbf{A}$[2]. The matrix now obtained,

$$\mathbf{V}^{(m)}_{\mu} = \mathbf{A}\mathbf{P}^{(m)}_{\text{trans}} \mathbf{P}^{(m)}_{\text{rot}} \mathbf{S}^{(m)}_{\mu} \; , \tag{E.13}$$

is a transformation matrix in the basis of pure vibrational eigenvectors of the Hessian matrix. Applying this matrix on the mass-weighted Hessian yields

$$\mathbf{V}^{(m)\dagger}_{\mu} \mathbf{F}^{(m)} \mathbf{V}^{(m)}_{\mu} = \mathbf{F}^{(v)}_{\mu} \quad . \tag{E.14}$$

The dimensions of matrix $\mathbf{F}^{(v)}_{\mu}$ are reduced by the number of removed translational and rotational modes, which correspond to the irrep Γ_{μ} (compared to the matrix $\mathbf{F}^{(s)}_{\mu}$, which would be obtained without projecting off the non-vibrational contributions).

The pure translational and rotational modes are generated as follows:

[1] These matrices contain the eigenvectors of the projection operators that block-diagonalize the mass-weighted Hessian; see [1] for details

[2] This matrix is constructed via *Gram–Schmidt* orthogonalization, such that $\mathbf{V}^{(m)}_{\mu}$ is orthonormal and does not contain any zero vectors.

- As far as the translational modes are concerned, these are assumed as motions along the axes of the laboratory coordinate system. The absolute value of the normalized distortion is known from Eq. (E.3). After mass-weighting, we obtain for the entry corresponding to atom i and the rotational mode j

$$R^{(m)}_{\text{trans},ji} = \sqrt{m_i}\mathbf{R}^{(c)}_j = \frac{\sqrt{m_i}}{\sqrt{M_{\text{total}}}}\mathbf{e}_j \quad . \tag{E.15}$$

- The treatment of the rotational modes is more complicated. They are only orthogonal if the axis of rotation are the principal axis of the moment of inertia. This can be achieved by diagonalizing the matrix of the moment of inertia, $\mathbf{I}$,

$$\omega^{\dagger}_{\text{rot}}\mathbf{I}\omega_{\text{rot}} = \mathbf{I}^{(\text{diag})} \quad . \tag{E.16}$$

The absolute value of ω_{rot} is given by Eq. (E.8). After mass-weighting we obtain for the entry corresponding to atom i and the rotational mode j

$$\begin{aligned} R^{(m)}_{\text{rot},ji} &= \sqrt{m_i}R^{(c)}_{ji} && \text{(E.17)} \\ &= \sqrt{m_i}(\mathbf{x}^{\perp}_{ij} \times \omega_{\text{rot},j}) && \text{(E.18)} \\ \mathbf{x}^{\perp}_{ij} &= \mathbf{x}_i - \frac{\mathbf{x}_i \cdot \omega_{\text{rot},j}}{\omega^2_{\text{rot},j}}\omega_{\text{rot},j} \quad . && \text{(E.19)} \end{aligned}$$

The modes generated with this treatment correspond to circular motions, i.e., to exact rotations only in case of infinitesimal distortions. However, this is also the case for exact rotational normal modes, since they can also describe curvilinear motions only in the limit of infinitesimal steps.

F. Tools and scripts

F.1 Scripts helpful for running `snfdc`

In the subdirectory `scripts` of the installation directory, you can find several scripts, which are either mandatory or useful for running the programs of the SNF package. Therefore, these scripts should be available in your `$PATH`. The scripts have the following purposes:

`choose_nodes`: selects nodes from `$CIPROC/ELIGIBLE` file to be added to the virtual parallel machine; mandatory for the PVM version of SNFDC; see Secs. 4.4 and 4.5

`dc_killtask`: kills PVM tasks on remote machines; mandatory script for SNFDC; see Sec. 4.5

`mkdcinput`: prepares the `DCINPUT` file; see Secs. 4.4 and 4.5

`dcmore`: SNFDC status monitor; displays the file `TMPdcstat` in time intervals of 5 seconds. The update time can be set via `dcmore <val>`, `<val>` = update time in seconds.

F.2 Graphical tools

For the graphical visualization of the spectra obtained with SNF, the `Fortran77` program SPECPLOT is available under
`$SNF_PATH/src/utilities/specplot_1.4/specplot`.
Whereas IR and Raman spectra are plotted by `snf` by default, SPECPLOT is particularly designed for the following special purposes:

- plot VROA spectra (only backscattering intensities implemented so far),
- plot Raman spectra which are obtained as a “byproduct” of a VROA calculation (only backscattering intensities implemented so far),
- plot selected peaks of Raman and VROA spectra only,

- plot several IR spectra, which are read from g98.out files, into one figure.

The first three points mentioned above require a control file `vroaplot_control`. Examples may be found in `$SNF_PATH/examples/specplot`. Specplot is based on Gnuplot. A special feature of the program is that it provides the spectrum as a postscript file, which is automatically shown via Ghostview, and which is updated after every command you enter (provided the "watch file" option is selected in your Ghostview settings). Such commands might request, for example, adjusting axis ranges, selecting and deselecting individual peaks, or some other features, which can be chosen in the main Specplot menu:

```
Generate spectra plots
 ----------------------


  The current plot of your spectra is just being shown.
  The following settings can be changed:
  [d]ist      :  distance between baselines
  [xr]ange    :  minimum and maximum frequency
  [ymin] <num>:  minimum intensity = <num>
  [ymax] <num>:  maximum intensity = <num>
  [tit]le     :  title
  [xl]<string>:  label of x-axis = <string>
  [yl]<string>:  label of y-axis = <string>
  [l]egends   :  legends of individual plots
  [gnu]plot   :  enter gnuplot commands
  [th] <int>  :  set thickness of lines to <int>
  [pg/plo/pli]:  plot Gaussian / Lorentzian /line data
  [ali]       :  add line spectrum to
              :  Gaussian / Lorentzian plot
  [nali]      :  remove line spectrum
  [scli <num>]:  scale original line spectrum
                 by factor <num>
  [pw <num>]  :  use peakwidth = <num>
  [spw]       :  adjust peakwidth for selected region
  [rm] <a-b>  :  extinguish wavenumber range <a> to <b>
  [re] <fname>  <a-b> : reread peaks in wavenumber
                 range <a> to <b> from file <fname>
  [wi] <num>  :  width of plot  = <num>
```

```
[he] <num>   :  height of plot  = <num>
[gr] <a-b>   :  group peaks in wavenumber range
                <a> to <b>
[sqgr]       :  group peaks in wavenumber range
                <a> to <b> and plot them as rectangles

Further options:
! <command> :  execute shell command <command>
[pr] <int>  :  print list of freq.s and intensities
               for <int>th file (default: file 1)
[sa <file>] :  save spectrum as file.eps, file.gpin
               and file.dat
[su] <a>-<b>:  print sum of intensities of all peaks
               in wavenumber range <a> to <b>
<quit>     :  quit without changes
```

Please note that Specplot is work in progress - it should work for all the above-mentioned purposes, but in some cases, you may need to adjust some lines in the code to get exactly the result you want. The recommended compiler is Intel's `ifort`.

G. GNU Free Documentation License

GNU Free Documentation License
Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this

Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
D. Preserve all the copyright notices of the Document.
E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
H. Include an unaltered copy of this License.
I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions

it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the

Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate.

Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
      Copyright (c)  YEAR  YOUR NAME.
      Permission is granted to copy, distribute and/or modify this document
      under the terms of the GNU Free Documentation License, Version 1.1
      or any later version published by the Free Software Foundation;
      with the Invariant Sections being LIST THEIR TITLES, with the
      Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
      A copy of the license is included in the section entitled "GNU
      Free Documentation License".
```

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Bibliography

[1] J. Neugebauer, M. Reiher, C. Kind, B. A. Hess. Quantum Chemical Calculation of Vibrational Spectra of Large Molecules — Raman and IR Spectra for Buckminsterfullerene. *J. Comp. Chem.*, **23** (2002) 895–910.

[2] G. Herzberg. *Molecular Spectra and Molecular Structure II. Infrared and Raman Spectra of Polyatomic Molecules.* Van Nostrand Reinhold, New York, 1945.

[3] E. Bright Wilson, J. C. Decius, Paul C. Cross. *Molecular Vibrations.* McGraw-Hill, New York, 1955.

[4] Derek Albert Long. *The Raman Effect: A Unified Treatment of the Theory of Raman Scattering by Molecules.* John Wiley & Sons, New York, 2002.

[5] Laurence D. Barron. *Molecular Light Scattering and Raman Optical Activity.* Cambridge University Press, Cambridge, 2nd ed., 2004.

[6] P. R. Bunker, P. Jensen. *Molecular Symmetry and Spectroscopy.* NRC Research Press, Ottawa, Canada, 2 ed., 1998.

[7] Willard Miller. *Symmetry Groups and their Applications.* Academic Press, New York, 1972.

[8] David S. Schonland. *Molecular Symmetry.* Van Nostrand Reinhold Company, London, 1971.

[9] *PVM - Parallel Virtual Machine.* http://www.epm.ornl.gov/pvm/pvm_home.html.

[10] *MPI — Message Passing Interface* . http://www-unix.mcs.anl.gov/mpi/indexold.html, http://WWW.ERC.MsState.Edu/misc/mpi/, 1994.

[11] *MPICH - A Portable Implementation of MPI.* http://www-unix.mcs.anl.gov/mpi/mpich, 2001.

[12] Reinhart Ahlrichs, Michael Bär, Marco Häser, Hans Horn, Christoph Kölmel. Electronic structure calculations on workstation computers: The program system TURBOMOLE. *Chem. Phys. Lett.*, **162**(3) (1989) 165–169.

[13] C. Angeli, K. L. Bak, V. Bakken, O. Christiansen, R. Cimiraglia, S. Coriani, P. Dahle, E. K. Dalskov, T. Enevoldsen, B. Fernandez, C. Hättig, K. Hald, A. Halkier, H. Heiberg, T. Helgaker, H. Hettema, H. J. Aa. Jensen, D. Jonsson, P. Jørgensen, S. Kirpekar, W. Klopper, R. Kobayashi, H. Koch, A. Ligabue, O. B. Lutnæs, K. V. Mikkelsen, P. Norman, J. Olsen, M. J. Packer, T. B. Pedersen, Z. Rinkevicius, E. Rudberg, T. A. Ruden, K. Ruud, P. Salek, A. Sanchez de Meras, T. Saue, S. P. A. Sauer, B. Schimmelpfennig, K. O. Sylvester-Hvid, P. R. Taylor, O. Vahtras, D. J. Wilson, H. Ågren. DALTON, a molecular electronic structure program, Release 2.0 (2005), see http://www.kjemi.uio.no/software/dalton/dalton.html, 2005.

[14] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, J. A. Montgomery, Jr., T. Vreven, K. N. Kudin, J. C. Burant, J. M. Millam, S. S. Iyengar, J. Tomasi, V. Barone, B. Mennucci, M. Cossi, G. Scalmani, N. Rega, G. A. Petersson, H. Nakatsuji, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, M. Klene, X. Li, J. E. Knox, H. P. Hratchian, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, P. Y. Ayala, K. Morokuma, G. A. Voth, P. Salvador, J. J. Dannenberg, V. G. Zakrzewski, S. Dapprich, A. D. Daniels, M. C. Strain, O. Farkas, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. V. Ortiz, Q. Cui, A. G. Baboul, S. Clifford, J. Cioslowski, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham, C. Y. Peng, A. Nanayakkara, M. Challacombe, P. M. W. Gill, B. Johnson, W. Chen, M. W. Wong, C. Gonzalez, J. A. Pople. Gaussian 03, Revision C.02. Gaussian, Inc., Wallingford, CT, 2004.

[15] G. te Velde, F. M. Bickelhaupt, E. J. Baerends, C. Fonseca Guerra, S. J. A. van Gisbergen, J. G. Snijders, T. Ziegler. *J. Comp. Chem.*, **22**(9) (2001) 931–967.

[16] T. Williams, C. Kelley. gnuplot Version 3.7 — An interactive Plotting Program. Organized by D. Denholm, 1998.

[17] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, V. G. Zakrzewski, J. A. Montgomery, R. E. Stratmann, J. C. Burant, S. Dapprich, J. M. Millam, A. D. Daniels, K. N. Kudin, M. C. Strain, O. Farkas, J. Tomasi, V. Barone, M. Cossi, R. Cammi, B. Mennucci, C. Pomelli, C. Adamo, S. Clifford, J. Ochterski, G. A. Petersson, P. Y. Ayala, Q. Cui, K. Morokuma, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. Cioslowski, J. V. Ortiz, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. Gomperts, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham, C. Y. Peng, A. Nanayakkara, C. Gonzalez, M. Challacombe, P. M. W. Gill, B. G. Johnson, W. Chen, M. W. Wong, J. L. Andres, M. Head-Gordon, E. S. Replogle, J. A. Pople. Gaussian 98 (Revision A.7). Gaussian, Inc., Pittsburgh PA, 1998.

[18] J. J. P. Stewart, I. Rossi, W.-P. Hu, G. C. Lynch, Y.-P. Liu, Y.-Y. Chuang, J. Li, C. J. Cramer, P. L. Fast, D. G. Truhlar. Mopac 5.09mn. University of Minnesota, Minneapolis, 1999.

[19] Gijs Schaftenaar. Molden3.7 — a pre- and post processing program of molecular and electronic structure. CMBI, the Netherlands, 2001.

[20] Jmol — a free, open source molecule viewer. http://jmol.sourceforge.net, 2003.

[21] Xmol Version 1.3.1. Research Equipment Inc., dba Minnesota Supercomputer Center, Inc., 1993.

[22] Jerzy Cioslowski. *J. Am. Chem. Soc.*, **111** (1989) 8333–8336.

[23] L. F. Richardson. The Approximate Arithmetical Solution by Finite Differences of Physical Problems involving Differential Equations, with an Application to the Stresses in a Masonry Dam. *Phil. Trans. Roy. Soc. London A*, **210** (1910) 307–357.

[24] Guntram Rauhut, Peter Pulay. Transferable Scaling Factors for Density Functional Derived Vibrational Force Fields (Erratum: *ibid.* **99, 14572 (1995)**. *J. Phys. Chem.*, **99** (1995) 3093–3100.

[25] Laurence D. Barron, Lutz Hecht, Iain McColl, Ewan W. Blanch. Raman optical activity comes of age. *Mol. Phys.*, **102** (2004) 731–744.

[26] P. W. Atkins, L. D. Barron. Rayleigh scattering of polarized photons by molecules. *Mol. Phys.*, **16** (1969) 453–466.

[27] A. D. Buckingham, M. B. Dunn. Optical activity of oriented molecules. *J. Chem. Soc.*, **A** (1971) 1988–1991.

[28] L. D. Barron, A. D. Buckingham. Rayleigh and Raman scattering from optically active molecules. *Mol. Phys.*, **20**(6) (1971) 1111–1119.

[29] L. D. Barron, M. P. Boogard, A. D. Buckingham. Raman scattering of circularly polarized light by optically active molecules. *J. Am. Chem. Soc.*, **95** (1973) 603–605.

[30] W. Hug, S. Kint, G.G. Bailey, J. R. Scherer. Raman circular intensity differential spectroscopy. Spectra of $(-)$-α-pinene and $(+)$-α-phenylethylamine. *J. Am. Chem. Soc.*, **97** (1975) 5589.

[31] Donald A. McQuarrie, John D. Simon. *Molecular Thermodynamics.* University Science Books, Sausalito, 1999.

[32] Karin Eichkorn, Oliver Treutler, Holger Öhm, Marco Häser, Reinhart Ahlrichs. Auxiliary basis sets to approximate Coulomb potentials. *Chem. Phys. Lett.*, **240** (1995) 283–290.

[33] Karin Eichkorn, Oliver Treutler, Holger Öhm, Marco Häser, Reinhart Ahlrichs. Erratum: Auxiliary basis sets to approximate Coulomb potentials. *Chem. Phys. Lett.*, **242** (1995) 652–660.

[34] K.-E. J. Hallin, J. W. C. Johns, A. Trombetti. *Can. J. Phys.*, **59** (1981) 663.

[35] F. Hegelund, H. Burger, O. Polanz. *J. Mol. Spectrosc.*, **167** (1994) 1.

[36] V. E. Bondybey, J. W. Nibler. *J. Chem. Phys.*, **58** (1973) 2125.

[37] J. Neugebauer, M. Reiher, B. A. Hess. Coupled Cluster Raman Intensities: Assessment and Comparison with Multi-Configuration and DFT Methods. *J. Chem. Phys.*, **117**(19) (2002) 8623–8633.

[38] B. A. Hess. NUMFREQ. University of Erlangen-Nürnberg, 2001. [Based on work by S. Grimme, C. Marian and M. Gastreich, 1998, University of Bonn].