Crash Course in using Turbomole at CSC, 25.4.2002

Nino Runeberg, CSC Ltd. runeberg@csc.fi, phone +358 (0)9 457 2733

1 Introduction

These notes are meant as an introduction to the usage of the Turbomole progam package as installed on the p690b (`p690b.csc.fi`) at CSC. The tutorial covers the following topics:

- Generating an input structure
- Using the interactive input generator **Define**
- Preparing batch jobs
- Single point calculations
- Geometry optimization
- Calculation of properties
- Visualization of results

2 Building the input structure

Small molecules can be fed into the TURBOMOLE program via the interactive input generator, but for larger systems it is usually much more convenient to prepare the structure in a 'molecular editor'.

CSC provides several different suructure editors — the one you are most familiar with being usually the best.

`Cerius`[2] is a program package for molecular modelling, which can be used for building structures. It is installed on `cedar.csc.fi`, and started with `cerius2`. Linux users, please note that for some reason the builder doesn't make bonds when the `Sawfish` window manager is used. The problem is fixed if the window manager is changed to e.g. `twm`, `fvwm2`, or `enlightenment`.

After logging into the `p690b.csc.fi`, enter a suitable working directory and copy your structure *filename.xxx* into it. After giving the command `use turbo`, convert the 'XXX' structure into Turbomole format by the command `tmbabel -ixxx filename.xxx -oturbo coord`. The `tmbabel` is a modified version of the popular `babel` molecular structure format interconversion program(`http://www.eyesopen.com/babel.html`), which is able to convert a wide variety of different structure formats into TURBOMOLE `coord` format, and vice versa. Enter `tmbabel` without arguments for a full list of supported input and output formats.

The file *coord* will now contain your structure in Turbomole format. If the initial structure is very rough (depending on the original source), it can be further improved by performing a geometry optimization at the Universal Force Field (UFF) level. First, run a single-point UFF calculation by giving the `uff` command. This will create a *control* file in your working directory.

```
$symmetry c1
$uff
        50         1          1 ! maxcycle,modus,nqeq
    111111                      ! iterm
  0.10D-07  0.10D-04            ! econv,gconv
      0.00  1.15                ! qtot,dfac
  0.10D+03  0.10D-04       0.30 ! epssteep,epssearch,dqmax
        25      0.10       0.00 ! mxls,dhls,ahls
      1.00      0.00       0.00 ! alpha,beta,gamma
         F         F          F ! transform,lnumhess,lmd
$grad file=uffgradient
$uffhessian file=uffhessian0-0
$ufftopology file=ufftopology
$coord file=coord
$forceinit on
  carthess
$hessian file=uffhessian0-0
$statistics dscf
$end
```

Change the `maxcycle` parameter in the *control* file from 1 to 50 and restart `uff`. This will optimize the structure at the UFF level. Now you have a decent starting structure, and you are ready to start the interactive TURBOMOLE input program `define`.

Using Define

By giving the command `define`, the following menu is introduced

```
*********************************************************
*                                                       *
*                    D E F I N E                        *
*                                                       *
*        TURBOMOLE'S  INTERACTIVE  INPUT  PROGRAM       *
*                                                       *
*  Quantum Chemistry Group       University of Karlsruhe  *
*                                                       *
*********************************************************

Please note the following restrictions in the current version:
* Internal coordinates menu: <iaut> automatic assignment
  may fail for complicated molecular topologies
* MO-menu: <use> command can not deal with change of basis set


DATA WILL BE WRITTEN TO THE NEW FILE control

IF YOU WANT TO READ DEFAULT-DATA FROM ANOTHER control-TYPE FILE,
THEN ENTER ITS LOCATION/NAME OR OTHERWISE HIT >return<.
```

Hit `<enter>` and it will ask for the input title

```
INPUT TITLE OR
ENTER & TO REPEAT DEFINITION OF DEFAULT INPUT FILE
```

After giving the title, hit `<enter>` and the geometry menu appears

```
SPECIFICATION OF MOLECULAR GEOMETRY ( #ATOMS=0     SYMMETRY=c1  )
YOU MAY USE ONE OF THE FOLLOWING COMMANDS :
sy <group> <eps> : DEFINE MOLECULAR SYMMETRY (default for eps=3d-1)
desy <eps>       : DETERMINE MOLECULAR SYMMETRY AND ADJUST
                   COORDINATES (default for eps=1d-6)
susy             : ADJUST COORDINATES FOR SUBGROUPS
ai               : ADD ATOMIC COORDINATES INTERACTIVELY
a <file>         : ADD ATOMIC COORDINATES FROM FILE <file>
aa <file>        : ADD ATOMIC COORDINATES IN ANGSTROEM UNITS FROM FILE <file>
sub              : SUBSTITUTE AN ATOM BY A GROUP OF ATOMS
i                : INTERNAL COORDINATE MENU
ired             : REDUNDANT INTERNAL COORDINATES
red_info         : DISPLAY REDUNDANT INTERNAL COORDINATES
m                : MANIPULATE GEOMETRY
w <file>         : WRITE MOLECULAR COORDINATES TO FILE <file>
r <file>         : RELOAD ATOMIC AND INTERNAL COORDINATES FROM FILE <file>
name             : CHANGE ATOMIC IDENTIFIERS
del              : DELETE ATOMS
dis              : DISPLAY MOLECULAR GEOMETRY
banal            : CARRY OUT BOND ANALYSIS
*                : TERMINATE MOLECULAR GEOMETRY SPECIFICATION
                   AND WRITE GEOMETRY DATA TO CONTROL FILE

IF YOU APPEND A QUESTION MARK TO ANY COMMAND AN EXPLANATION
OF THAT COMMAND MAY BE GIVEN
```

Read the coordinates you previously saved in the file *coord* by typing `a coord`. Give the command `desy` to automatically determine the symmetry. If you want to do a geometry optimization, issue the command `ired` in order to generate a set of redundant internal coordinates. Exit the geometry menu by typing `*`. In the next menu the basis sets are defined.

```
ATOMIC ATTRIBUTE DEFINITION MENU  ( #atoms=21    #bas=21    #ecp=0     )

b    : ASSIGN ATOMIC BASIS SETS
bb   : b RESTRICTED TO BASIS SET LIBRARY
bl   : LIST ATOMIC BASIS SETS ASSIGNED
bm   : MODIFY DEFINITION OF ATOMIC BASIS SET
bp   : SWITCH BETWEEN 5d/7f AND 6d/10f
lib  : SELECT BASIS SET LIBRARY
ecp  : ASSIGN EFFECTIVE CORE POTENTIALS
ecpb : ecp RESTRICTED TO BASIS SET LIBRARY
ecpi : GENERAL INFORMATION ABOUT EFFECTIVE CORE POTENTIALS
ecpl : LIST EFFECTIVE CORE POTENTIALS ASSIGNED
ecprm: REMOVE EFFECTIVE CORE POTENTIAL(S)
c    : ASSIGN NUCLEAR CHARGES (IF DIFFERENT FROM DEFAULTS)
cem  : ASSIGN NUCLEAR CHARGES FOR EMBEDDING
m    : ASSIGN ATOMIC MASSES (IF DIFFERENT FROM DEFAULTS)
dis  : DISPLAY MOLECULAR GEOMETRY
dat  : DISPLAY ATOMIC ATTRIBUTES YET ESTABLISHED
h    : EXPLANATION OF ATTRIBUTE DEFINITION SYNTAX
*    : TERMINATE THIS SECTION AND WRITE DATA OR DATA REFERENCES TO control
GOBACK=& (TO GEOMETRY MENU !)
```

If you are satisfied with the default SV(P) basis set just hit `*`.

In the next menu the occupation numbers and the initial guess are given.

```
OCCUPATION NUMBER & MOLECULAR ORBITAL DEFINITION MENU

CHOOSE COMMAND
infsao     : OUTPUT SAO INFORMATION
eht        : PROVIDE MOS && OCCUPATION NUMBERS FROM EXTENDED HUECKEL GUESS
use <file> : SUPPLY MO INFORMATION USING DATA FROM <file>
man        : MANUAL SPECIFICATION OF OCCUPATION NUMBERS
hcore      : HAMILTON CORE GUESS FOR MOS
&          : MOVE BACK TO THE ATOMIC ATTRIBUTES MENU
THE COMMANDS  use  OR  eht  OR  *  OR q(uit) TERMINATE THIS MENU !!!
FOR EXPLANATIONS APPEND A QUESTION MARK (?) TO ANY COMMAND

eht
```

Just type `eht` to select an extended Hückel guess. Answer `y` to the next question in order to use the default EHT-parameters.

```
DO YOU WANT THE DEFAULT PARAMETERS FOR THE EXTENDED HUECKEL CALCULATION ?
DEFAULT=y   HELP=?

y
```

Enter the total charge of the system.

```
ENTER THE MOLECULAR CHARGE  (DEFAULT=0)

0
```

The next menu presents the suggested occupation numbers. Answer `y` in order to accept the suggested occupation.

```
 AUTOMATIC OCCUPATION NUMBER ASSIGNMENT ESTABLISHED !
 FOUND CLOSED SHELL SYSTEM !
 HOMO/LUMO-SEPARATION :  0.149655
 ORBITAL  SYMMETRY      ENERGY     DEFAULT
 (SHELL)    TYPE                  OCCUPATION
    30    30a          -0.47282      2
    31    31a          -0.44996      2
    32    32a          -0.42558      2
    33    33a          -0.41026      2
    34    34a          -0.26060      0
    35    35a          -0.24558      0
    36    36a          -0.07592      0

 DO YOU ACCEPT THIS OCCUPATION ?  DEFAULT=y

y
```

Finally the 'general menu' is presented.

```
 GENERAL MENU : SELECT YOUR TOPIC
 scf    : SELECT NON-DEFAULT SCF PARAMETER
 mp2    : FROZEN CORE OPTION FOR MP2 CALCULATIONS
          (ACTIVE ONLY FOR CLOSED SHELL MP2 ENERGIES
          AND FOR RI-MP2 ENERGIES/GRADIENTS.)
 prop   : SELECT TOOLS FOR SCF-ORBITAL ANALYSIS
 drv    : SELECT NON-DEFAULT INPUT PARAMETER FOR EVALUATION
          OF ANALYTICAL ENERGY DERIVATIVES
          (GRADIENTS, FORCE CONSTANTS)
 rex    : SELECT NON-DEFAULT OPTIONS FOR GEOMETRY UPDATES
 e      : DEFINE EXTERNAL ELECTROSTATIC FIELD
 dft    : DFT Parameters
 ri     : RI Parameters
 trunc  : USE TRUNCATED AUXBASIS DURING ITERATIONS
 dis    : DISPLAY MOLECULAR GEOMETRY
 list   : LIST OF CONTROL FILE
 &      : GO BACK TO OCCUPATION/ORBITAL ASSIGNMENT MENU
 * or q : END OF DEFINE SESSION
```

If you just want to do an SCF calculation you can now exit the program by typing `*`.

If you want to perform a DFT calculation type `dft` and a menu for the DFT-options is presented.

```
STATUS OF DFT_OPTIONS:
 DFT is NOT used
   functional b-p
     gridsize m3

 ENTER DFT-OPTION TO BE MODIFIED

 func: TO CHANGE TYPE OF FUNCTIONAL
 grid: TO CHANGE GRIDSIZE
  on:   TO SWITCH ON  DFT
 Just <ENTER>, q or '*' terminate this menu.
```

Type `on` to switch DFT on. By typing `func` you will get a list of available functionals.

```
func

 Enter the name of the functional to be used.
 Possible Values are:
  becke-exchange (Becke88 exchange, no corr.)
  b-lyp          (Becke88 exch. + Lee/Yang/Parr corr. func.)
  b-vwn          (Becke88 exchange + VWN corr. func.)
  b-p            (Becke88 exchange + VWN corr. func. +
                  Perdew86 corr. correction)
  lyp            (Lee/Yang/Parr corr. func.
                  without Exchange       !)
  cam            (Cambridge functional d.h. Handy:
                  scaled Dirac exchange
                  reparametrised Becke correction (beta = 0.008)
                  Lee/Yang/Parr correlation functional)
  b-lypa         (Becke exchange reparametrised
                  + Lee/Yang/Parr corr. func)
  bh-lyp         (Becke88 exch. + Lee/Yang/Parr corr. func.
                  mit halbem HF-Austausch)
  b3-lyp         (Beckes 3-Parameterformel)
  slater-dirac-exchange (slater-dirac-exchange, no corr.)
  s-vwn        (Dirac exchange with VWN corr. functional)
  vwn          (VWN correlation functional without Exch.)
```

Select a suitable functional and return to the general menu by typing `*`. Exit `define` by `*`.

In addition to the coordinate files *coord* and the original structure *filename.xxx* you should now have the files *control, basis* and `mos` in the present directory. The *control* file contains all the parameters defining your calculation. The *basis* file contains the chosen basis sets, and *mos* contains the initial guess for the molecular orbitals.

If you activated the 'RI' approximation in the last 'general' menu, you will also have the *auxbasis* file containing the auxiliary basis sets.

3 Preparing batch jobs

Parallel jobs

The TURBOMOLE program package consists of a series of modules. Some of these modules have been parallelized and run very efficiently on the IBM. The parallelized modules include

dscf For semi-direct SCF and DFT calculations. Dscf supports restricted closed shell (RHF), spin restricted open shell (ROHF) as well as spin unrestricted (UHF) runs. Dscf includes an in-core version (two electron integrals stored in memory) for small molecules.

grad requires a successful `dscf` run and performs gradient calculations for all cases treated by `dscf`.

ridft and rdgrad Perform DFT calculations - as `dscf` and `grad` - with the additional approximation that the total density is given by a sum of atom centered s, p, d ... functions - the RI-J auxiliary basis. This allows a very efficient treatment of Coulomb interactions. The following functionals are supported: S-VWN, B-P, B-LYP. While the resulting errors are negligible in almost all applications, computation time typically reduces by a factor 4-20. Since the nonlocal exchange cannot be treated this way, the RI-technique has no advantage for so-called DFT hybrid functionals such as B3LYP or Becke half-and-half.

mpgrad Requires a well converged SCF run - by `dscf`- and performs RHF closed shell or UHF calculations yielding single point MP2 energies and, if desired, the corresponding gradient. Requires **mp2prep** and **turbo_preproc** to adjust the control file.

relax requires a gradient run – by GRAD, RDGRAD or MPGRAD– and proposes a new structure based on the gradient and the approximated force constants. The approximated force constants will be updated.

In order to perform geometry optimizations TURBOMOLE contains the `jobex` script, which performs calculation sequencies 'energy→gradient→force relaxation→new structure'. A standard SCF or DFT geometry optimization is started by simply calling `jobex`. An optimization at RI-DFT level is launched by `jobex -ri`.

In order to submit a parallel job on the IBM, one has to prepare a suitable batch-job file for parallel jobs. An example is given below:

```
#!/bin/ksh
# @ job_type = parallel
# @ environment = COPY_ALL
# @ output = jobfile.out
# @ error = jobfile.err
# @ class = parallel
# @ notification = complete
# @ checkpoint = no
# @ restart = no
# @ node = 1
# @ tasks_per_node = 16
# @ queue
#
#
export MP_EUILIB=ip
#
export PARA_ARCH=MPI
export PATH=$TURBODIR/bin/`sysname`:$PATH
#
export PARNODES=15
#
project=turbo_opt

# substitute 'jobex' below with any of the desired
# parallel modules. e.g.
# ridft > test.out 2> test.err
#
jobex  > $project.out 2> $project.err
#
```

The current IBM configuration contains 3 independent nodes with 32 processors on each node. In the "parallel" class one can allocate a maximum of 16 processors. Depending on the number of processors needed in the actual run, a few parameters have to be changed in the example above. The parameter `node`, allocates the number of nodes reserved for the job (at the moment only 1), and the parameter `tasks_per_node` allocates the number of processors per each allocated node (maximum 16). In addition, the TURBOMOLE variable `PARNODES` has to be set to `nodes * tasks_per_node - 1`, because of the additional server task that is automatically started by each parallel TURBOMOLE run. Depending on the type of parallel job, the actual executable can be, besides **jobex**, either **dscf, grad, mpgrad, rdgrad,** or **ridft**. For additional information on the batch-job keywords, please read the *IBM SP User's Guide* (`http://www.csc.fi/oppaat/ibmsp/cscibmsp.pdf`).

Some hints for parallel runs

- Large SCF and conventional DFT calculations, which are dominated by computation of ERIs and numerical integration, can effectively utilize up to 32 processors.
- Even for comparably large RIDFT calculations it is seldom worth to use more than 16 processors, and often already 8 processors is the optimal choise.
- Note also that on one CPU the serial version is usually faster than the parallel version.

- In order to use the parallel version of **mpgrad,** please call the preprocessor `mp2prep`. Generate a hostfile by `echo 'p690b' >host.list`, start the `turbo_preproc`, follow the instructions and then copy the resulting *control_par* to *control.* Change the `$parallel_platform SP2` to `$parallel_platform SP3` in *control*, and remember to remove the *host.list* before submitting the job.

Please, don't waste resources by allocating more processors than the optimal number. If you find it difficult to decide the optimal configuration, please contact Nino Runeberg

Serial jobs

In addition to the parallelized modules, TURBOMOLE contains a series of modules that, though efficient, can only be run as serial jobs. The serial modules include

rimp2 calculates MP2 energies and gradients for RHF and UHF wavefunctions, using the RI technique. Start with a well converged SCF run (`$denconv .1d-6`, can be run in parallel). Then prepare the MP2 calculation with `rimp2prep`.

ricc2 calculates electronic excitation energies at the CIS, CIS(D) and CC2 level using either a closed shell RHF or a UHF SCF reference function. Employs the RI technique to approximate two-electron integrals. Done similarly to **rimp2**, but requires some additional parameters about the level of theory and a list of excitations in *control* (sem manual for details).

aoforce and uhfforce require a well converged SCF run – by DSCF, see keywords – and perform analytic RHF closed shell (AOFORCE) or UHF (UHFFORCE) calculations of force constants, vibrational frequencies, IR intensities. (Raman cross sections only for SCF closed shells, not for DFT). DFT is supported for closed shell calculations, the RI-J approximation is not implemented. Raman cross sections are only available for closed shell Hartree-Fock. The numerical calculation of force constants is also possible (see tool NUMFORCE in the manual, section 1.4).

escf requires a well converged SCF or DFT run and calculates time dependent and dielectric properties: - static and frequency-dependent polarizabilities within the SCF approximation, RHF and UHF - static and frequency-dependent polarizabilities within the closed shell DFT formalism, including hybrid functionals such as B3LYP - electronic excitations within the RHF and UHF CI(S) restricted CI method - electronic excitations within the so-called SCF-RPA approximation (poles of the frequency dependent polarizibility) - electronic excitations within the time dependent DFT formalism (adiabatic approximation), for closed shells only. It can be very efficient to use the RI approximation here, provided that the functional is of non-hybrid type.

mpshift requires a converged SCF or DFT run for closed shells. **mpshift** computes NMR chemical shieldings for all atoms of the molecule at the SCF, DFT or MP2 level within the GIAO ansatz and the (CPHF) SCF approximation. From this one gets the NMR chemical shifts by comparison with the shieldings for the standard compound usually employed for this purpose, e. g. TMS for carbon shifts. Note that NMR shielding typically requires better basis sets than necessary for geometries or energies. ECP's are not supported in MPSHIFT. Furthermore, NMR shieldings are very sensitive to correlation effects. A large "paramagnetic contribution", e. g. of the order of "diamagnetic contribution" is a serious reason to worry about the accuracy.

moloch computes a variety of first-order properties and analyses of the wavefunction as can be seen from the keywords. Also atomic point charges can be fitted to the electrostatic potential of a a molecule.

In fact, all the parallelized modules are also available as serial versions. The names of the programs are the same, but the serial versions of the executables are automatically chosen, if the TURBOMOLE variable `PARA_ARCH=MPI` is not set.

An example of a batch-job file for serial jobs is given below:

```
#!/bin/ksh
# @ job_type = serial
# @ environment = COPY_ALL
# @ output = jobfile.out
# @ error = jobfile.err
# @ class = test
# @ notification = complete
# @ checkpoint = no
# @ restart = no
# @ queue
#
#
#
export PATH=$TURBODIR/bin/`sysname`:$PATH
#
#
project=h2o_rimp2_sp
# substitute 'jobex' below with any of the desired
# parallel modules. e.g.
# ridft > test.out 2> test.err
#
rimp2 > $project.out 2> $project.err
#
```

4 Single-point energy calculations

All single-point calculations are performed by simply modifying the batch-job file to execute the appropriate module, e.g.:

```
dscf > dscf.log
```

The procedure for performing energy calculations only, at various levels are summarized in the table below.

Table 1: Energy only calculations using TURBOMOLE. The 'Type' indicates wether the module runs in parallel (P) or only as serial (S).

Level	Type	Command	Comments
SCF	P	`dscf`	The SCF energy can be calculated directly after a `define` run, without any further keywords or previous runs.
DFT	P	`dscf`	The DFT energy calculations can be calculated directly after a `define` run, in which the DFT option is activated in the 'general menu' (this vill add the keyword `$dft` into the *control* file.)
RIDFT	P	`ridft`	As a normal DFT calculation but the 'RI' option has to be activated in the 'general menu' of `define`. Observe: If you run an RIDFT calculation using `dscf` the 'RI' flags are ignored and a normal DFT calculation is done.
MP2	P	`mpgrad`	MP2 calculations need to be preceded by a well converged `dscf` run. Insert `$denconv .1d-6` into the *control* file and rerun `dscf`. Continue with the preparations by running the `mp2prep -e` script. Prepare a hostfile by `echo 'p690b' >host.list`, run `turbo_preproc`, copy *control_par* to *control*, change `SP2` to `SP3` in *control*. Finally, remove the *host.list* before starting the actual `mpgrad` run.
RIMP2	S	`rimp2`	RIMP2 calculations are done as MP2 calculations, exept that the 'RI' option is activated in `define`, and the preparation is done with the `rimp2prep` tool. No further preparation needed.
RICC2	S	`ricc2`	RICC2 calculations are performed as RIMP2 calculations, by preparing the calculation with the `rimp2prep` tool. A few additional parameters has to be added manually into *control*

5 Geometry optimization

Geometry optimization is most conveniently done in TURBOMOLE by running the `jobex` tool, which will automatically cycle through the sequence 'energy→gradient→force relaxation→new structure', until the maximum number of cycles, or the desired convergence criteria are reached.

Before starting a geometry optimization a set of internal coordinates has to be generated. The simplest way is to use `ired` in the 'geometry menu' of `define`, which automatically generates a set of generalized internal coordinates. If define fails to generate a full set of internal coordinates the optimization can still be done either by constructing internal coordinates by hand (and then checking with `define` that they are linearly independent) or by using cartesian coordinates (which will make the optimization considerably slower) by setting

```
$optimize
   internal   off
   redundant  off
   cartesian  on
```

in the control file.

The commands for performing geometry optimzation at various levels are summarized in the table below.

Table 2: Geometry optimization using the `jobex` script. The 'Type' indicates wether the job runs in parallel (P) or only as serial (S).

Level	Type	Command	Comments
SCF	P	`jobex`	
DFT	P	`jobex`	Remember to activate 'DFT' in define
RIDFT	P	`jobex -ri`	Remember to activate 'DFT' and 'RI' in define
MP2	P	`jobex -level mp2`	Before starting `jobex` you should run `dscf` with `$denconv .1d-6` included in the *control* file, followed by the script `mp2prep -g $WRKDIR` and the `turbo_preproc` script.
RIMP2	S	`jobex -ri -level mp2`	Before starting `jobex` you should run `dscf` with `$denconv .1d-6` included in the *control* file, followed by the script `rimp2prep`.

For a list of available options to use `jobex -h`. When the `jobex` script is started it will create the file *job.start* which contains information about the current options, the file *not.converged*, and some other files. The file *job.last* contains all the output from the last completed geometry cycle, while the output of the running cycle is collected into the file *job.<cyclenumber>*. The file *gradient* contains the energy, structure, and gradient for all completed cycles. The convergence of a geometry optimization can be conveniently monitored by the command `grep dx gradient`, which lists all completed cycles, together with the corresponding energy and gradient.

6 Calculation of properties and solvation effects

6.1 COSMO

COSMO is the continuum solvation 'Conductor-Like Screening Model' developed by A. Klamt and G. Schuurmann (JCS Perkin Trans. 2, 1993, 799) to describe dielectric screening effects in solvents. The method is currently implemented in TURBOMOLE for SCF, DFT, and RI-DFT energies and gradients.*Observe that the COSMO calculation has to be done in* **`c1`** *symmetry.* First you set up a normal calculation using `define`(c1 symmetry). Then you prepare the COSMO calculation by running the `cosmoprep` script. In `cosmoprep` you typically only change the dielectric constant, and if you don't have hydrogens you have to set *rsolv* to the radius of the smallest atom in the system (default H = 1.4 Å). In the 'radius definition' menu just enter `r all o`. Now you can run an energy calculation or a geometry optimisation in the normal way.

An energy calculation starts with the construction of the cavity surface grid. Within the SCF procedure, the screening charges are calculated in every cycle and the potential generated by these charges is included into the Hamiltonian. This ensures a variational optimization of both the molecular orbitals and the screening charges, which then also allows the evaluation of analytic gradients (programs `grad` and `rdgrad`). Additional information on COSMO can be found in the TURBOMOLE manual (Appendix D, p 114).

6.2 MPSHIFT

The program `mpshift` computes NMR chemical shieldings for all atoms of the molecule at the SCF, DFT or MP2 levels within the GIAO ansatz and the (CPHF) SCF approximation. From this one gets the chemical shifts (NMR) by comparison with the shieldings for the standard compound usually employed for this purpose, e. g. TMS for carbon shifts.

For SCF or DFT calculations, no specifications have to be made in the *control* file, but `mpshift` needs converged MO vectors from a SCF or DFT run (`dscf` or `ridft`). It is not possible to run the `mpshift` program in the fully direct mode when doing an SCF, MP2 or a DFT (using hybrid functionals) run, so you'll have to do a statistics run by calling `mp2prep -c` before calling `mpshift.`

Please note that NMR shielding typically requires better basis sets than necessary for geometries or energies. Furthermore, NMR shieldings are very sensitive to correlation effects. A large 'paramagnetic contribution', e.g. of the order of the ‘diamagnetic contribution’ is a serious reason to worry about the accuracy. ECPs are currently not supported in `mpshift`.

6.3 ESCF

The program `escf` can be used in order to compute directly SCF linear response properties as excitation energies, oscillator strengths, optical rotatory strength, and static and frequency dependent polarizabilities. Furthermore, it allows to check the reliability of the SCF picture, by doing a stability analysis of single-determinant wave functions (second derivative of energy with respect to orbital rotations).

The capabilities of the present version of `escf` are summarized in the table below.

Table 3: Methods implemented in `escf` for calculating excitation energies and polarizabilities.

Level	Spin	$scfinstab	Comments
HF	open-shell	ucis, urpa	CIS or RPA excitation energies
DFT	”	urpa	TDDFT excitation energies
RIDFT	”	”	”
HF	closed-shell	ciss,cist,rpas,rpat	CIS or RPA excitation energies(singlet or triplet)
DFT	”	rpas,rpat	TDDFT excitation energies (singlet or triplet)
RIDFT	”	”	”

6.4 MOLOCH

The `moloch` program computes a variety of first-order properties and analyses of the wavefunction. Also atomic point charges can be fitted to the electrostatic potential of a a molecule. In order to prepare the input for `moloch` enter `define` and choose `props` from the last ’general menu’ (pass the other menus by simply hitting `<enter>`).

```
data group \$properties has not yet been specified



FOR INITIALIZING <moloch> KEYWORDS ENTER
  [return] : WRITE TO CONTROL FILE control (DEFAULT),   OR
  filename : WRITE TO ANOTHER FILE
```

Hit `<enter>` and the following menu is opened:

```
switch on one or more of the following options <i>
<i> = 1,..., 9
for switching off option <i>, specify -<i>

( 1) trace                                  off
( 2) moments                                off
( 3) potential                              off
( 4) cowan-griffin                          off
( 5) localization                           off
( 6) population analyses                    off
( 7) plot                                   off
( 8) firstorder                             off
( 9) fit                                    off

selecting an already active option indicates that
suboptions shall be modified
* or q(uit) = quit | for help, type help <integer>
```

The keyword `trace` compute the number of electrons N resulting from density matrix D and overlap matrix S according to N = trace (D * S) By selecting `moments` the electrostatic moments up to 3rd order can be calculated at a given point. The electrostatic potential, field, or field gradient can be computed by activating `potential`. The first order relativistic correction given as the expectation value of the cowan griffin operator (p4 term and Darwin term) is computed by selecting `cowan-griffin`. The `localization` option perform a localization of molecular orbitals following the procedure of Boys (separation of charge centers). If `population analyses` is activated a population analysis based on either Mulliken or occupation numbers. The option `firstorder` provide orbitals for a first order scf calculation by Löwdin orthogonalization of orbitals from subsystems. The keyword `fit` allows atomic point charges to be fitted to the electrostatic potential of a a molecule. By activating `plot` the molecular orbitals (delocalized or localized) or modified atomic orbital amplitudes or densities can be calculated on a 2-, or 3-dimensional grid. The resulting data can be used for graphical representation.

6.5 Visualizing amplitudes and densities

Next we will describe how to visualize orbitals or total densities using the `gOpenMol` program (`http://www.csc.fi/~laaksone/gopenmol/gopenmol.html`). Use the script `mkprop` (provided with gOpenMol) which extracts maximum and minimum coordinate values, and produces nice default dimensions for the grid

```
mkprop < coord >>control

$properties
    trace                          off
    moments                        off
    potential                      off
    cowan-griffin                  off
    localization                   off
    population analyses            off
    plot                           active
    firstorder                     off
    fit                            off
$grid
    mo density
    origin        .0000   .0000   .0000
    vector1      1.0000   .0000   .0000
    vector2       .0000  1.0000   .0000
    vector3       .0000   .0000  1.0000
    grid1 points 37 range -12.603491 12.562998
    grid2 points 23 range -8.090182 7.806338
    grid3 points 20 range -6.570322 6.873696
    infile=mos
    outfile=grid.dat
$end
```

Remove the redundant `$end` further up in the *control* file. Run `moloch` (if the cube is not very large, this can be done interactively), which produces the file `grid.dat` containing the total density on a grid. is used).

Although gOpenMol can be started on the `p690b` by the command `gopenmol`, it will be much faster and much more comfortable to install it on your own computer. (The installation is easy, visit
`http://www.csc.fi/~laaksone/gopenmol/gopenmol.html`
for details).

In order to display the calculated density, copy the structure and the corresponding data (e.g. the `grid.dat` file) into the directory where you start gOpenMol. After starting `gopenmol`, load the structure from the `File->Import->Coords` menu. Convert the density by using the `Run->TMole2plt` menu. Load the converted density from the `Plot-> Contour` menu. Import the density, and set the contour levels and colours.

If you want to display single orbitals change `mo density` to e.g. `mo 47a` (andi change the name of the output) in *control*. Then rerun `moloch`. For a orbitals, the "phase" information can be visualized by selecting two opposite contour levels (e.g. -0.01 and 0.01) with different colours.