Introduction to Molpro

by

Benjamin J. Lynch

March 21, 2006
Outline

• Background
• Capabilities
• SCF/DFT
  - MP2
  - FCI
  - MR-CI, MR-AQCC
  - CCSD(T)
  - DFT/HDFT
• Input
• Output
• Resource considerations
Background

• Maintained by H.-J. Werner and P.J. Knowles

• The package is focused on highly-accurate calculations.

• Costs $700-$31000 for a 5-year license
Overview

• Can perform CAS-SCF, MR-CI, MR-ACPF, MR-AQCC, FCI, various relativistic contributions, and fast CCSD(T)

• Runs in parallel
Not very user-friendly

- Spartan
- Gaussian
- GAMESS
- Molpro
- Columbus
Running Molpro

• Log into altix, balt, or regatta

module load molpro

molpro myinput.com
Temporary directories

• Temporary directories can be specified at runtime

```
molpro –W $HOME/tmp myinput.com
   - location of orbitals and CI vectors

molpro –d /scratch/$USER
   - location of large scratch files
```
Temporary directories

`molpro -I /scratch/$USER`

- permanent location of integral files. Integral files are stored in the directory specified with `--d`. When the job is complete, these files are copied to the permanent location specified with `-I`
Memory Usage

- Molpro will use a lot of memory (1-30 GB) for most calculations. Although the program is installed on the Netfinity and some SDVL machines, it is almost always run on the Altix, or Power4 (regatta) shared-memory machines.
More Molpro Run Options

• Specify the number of processors
  
molpro -n 8 myfile.com

• Specify the output filename
  
molpro -o myfile.output job1.com

• Overwrite output file if it exists
  
molpro -s myfile.com
.molprorc

• The default runtime options are in a file 
  /usr/local/molpro/molpro2002.6/bin/molpro.rc

• Default program settings can be changed by putting the file .molprorc in your home directory. These options will take precedence over the molpro.rc file.

• Options specified on the command line take precedence over both of these files.
Input Format

• Molpro input is a script that can specify any number of calculations. **If/then** and **do** loops can be put into the input file.

• You will need to be very careful about following the correct input format. Molpro is unforgiving and typically prints out cryptic error messages.
• At the start of an input file:

***, jobname

• This name must be the same if you want to restart a calculation
• At the end of an input file:

  --

• This is optional, anything after is ignored
• Commonly used files can be inserted into input files with:

```
Include, file
```

• This is commonly done to include basis sets, multi-step procedures, and geometries
• Instead of specifying the memory on the command line, it can be specified in the input file:

memory, 1000000
memory, 1000, K
memory, 1, M

• These all allocate 8 MB of memory
• Do loops can be used to perform energy scans or other tasks:

\[
\text{DR}=0.2 \\
\text{DO } \text{R}=1.0,6.0,\text{DR},\text{ANG} \\
\text{HF}; \\
\text{ENDDO}
\]

• Loop variables have a value and a unit (Angstroms)
• Do loops can loop over any set of values like this:

\[
\text{RVEC} = [1.0, 2.0, 4.0] \text{ ANG} \\
\text{DO I=1,} \#\text{RVEC} \\
\quad \text{R=RVEC(I)} \\
\quad \ldots \ldots \\
\text{ENDDO}
\]

• RVEC is an array, \#RVEC is the number of elements in the array.
• If/then statements can be used for flow control in your input file. This is especially common in user-defined procedures.

\[
\text{IF (logical expression) THEN statements ENDIF}
\]
• Several global variables can be set:

GTHRESH, ONEINT=1.d-12, ENERGY=1.d-6

• These control thresholds for including integrals, energy thresholds for convergence, density thresholds for convergence, etc.
• Information can be printed out anytime during the calculation using GPRINT.

GPRINT,BASIS,CIVECTOR
The geometry can be specified in several formats. A z-matrix is the default.

\[
\text{geometry} = \{o; h1, o, r; h2, o, r, h1, \theta}\]

INPUT
• A different geometry type can be specified by changing the geomtyp variable.

```plaintext
geomtyp=xyz
geometry={3
  water
  O,0.0,0.0,0.0
  H,1.0,0.0,0.0
  H,-0.7,0.7,0.0
}
```
• Basis sets that come with the program can be specified like:

```
basis=cc-pvdz
```
All Basis sets will use spherical $d$ and $f$ functions, even if the basis is defined otherwise.

You can override this behavior with the statement:

```plaintext
CARTESIAN
Basis=6-31G*
```
Alternative basis sets:

basis={...
– select MOLPRO in the Code box
– copy/paste resultant basis for selected atoms

ex basis=

! HYDROGEN (4s) -> [2s]
s,H0.187311370E+02,0.282539370E+01,0.640121700E+00,0.161277800E+00
c,1.3, 0.334946000E-01, 0.234726950E+00, 0.813757330E+00
c,4.4, 0.100000000E+01
}
• To run start the Hartree-Fock, mp2, and ccsd subprograms:

HF
mp2
ccsd
• For a BLYP dft calculation:

```
basis=vdz       !double-zeta basis set
df=[b,lyp]
uks
wf,16,4,2      !define wavefunction: 16 electrons,  
               !symmetry number, 2 unpaired  
               !electrons
```

• This name must be the same if you want to restart a calculation
• Be careful about your DFT input!

hf
dft

• This will calculate the Hartree-Fock energy, and then use the final HF density for the DFT calculation (instead of the KS density)
***, title
memory, 4, m
file, 1, name.int
file, 2, name.wfu
gprint, options
gdirect[, options]
basis=basisname
var1=value, var2=value, ...
geometry={...}
program
---
***,H2O test !title
memory,1,m         !memory, 1Mw = 8Mb
group={o;h1,o,r;h2,o,r,h1,theta} !Z-matrix
basis=cc-pvdz
r=1 ang            !bond length in angstroms
theta=104          !bond angle in degrees
runmp2             !perform HF and MP2 calculation
Pre-Defined Procedures

- `runmp2` performs HF and mp2 calculation
- `freqscf` performs SCF frequency calculation
- `freqdft` performs DFT frequency calculation
- `freqmp2` performs MP2 frequency calculation
Procedures:

Default procedures are contained in the file `molproi.rc`

A user modified version of the `molproi.rc` file can be generated by making a copy of the file in the users home directory

```bash
cp /usr/local/molpro2000.1/bin/molproi.rc ~/.molproirc
```

then appending the file with the newly defined procedure
• Example Procedure

• – Edit `~/.molproirc`

```plaintext
proc save_e
  if(#i.eq.0) i=0           !initialize variable if it does not exist
  i=i+1                        !increment i
  e(i)=energy              !save scf energy in variable e(i)
  method(i)=program !save the present method in variable m
endproc
```

!end of procedure
Using procedures:

```plaintext
***, H2O test
memory, 1, m
geometry={o;h1,o,r;h2,o,r,h1,theta} !Z-matrix
basis=cc-pvdz
r=1 ang, theta=104
hf
save_e
mp2
save_e
table, method, e
```

- `***, H2O test` — title
- `memory, 1, m` — memory, 1Mw = 8Mb
- `geometry={o;h1,o,r;h2,o,r,h1,theta}` — geometry variables
- `basis=cc-pvdz` — perform HF calculation
- `r=1 ang, theta=104` — call procedure, save results
- `hf` — perform MP2 calculation
- `save_e` — call procedure, save results
- `mp2` — print a table with results
- `save_e` —
- `table, method, e` —
• Pre-defined procedures:

• `optscf` performs SCF optimization
• `optdft` performs DFT optimization
• `optmp2` performs MP2 optimization
• look at the `~/.molproirc` file or the user manual for a full list of the pre-defined optimization procedures.
***, H2O test !title
memory, 1, m !memory, 1Mw = 8Mb
geomtyp=xyz !specify cartesian geometry
geometry={angstroms !coordinates in angstroms
3 !# of atoms in system
H2O Cartesian coordinates !title
O 0.0000 0.0000 0.0000 !coordinates
H 0.0000 0.0000 1.0000
H 0.9703 0.0000 -0.2419}
basis=cc-pvdz
runmp2 !HF and MP2 calculation
MOLPRO can use arithmetic, intrinsic, and logical expressions

arithmetic: +, −, *, /, ...

intrinsic: abs(), min(), sqrt(), cos(), acos(), ...

logical: .or., .and., .gt., .ge., ...

all trigonometric functions use or produce angles in degrees
Restarting a calculation

***,O2
file,2,O2.wfu,new
file,3,O2.aux,new
basis=vtz;
geometry={angstroms;o1;o2,o1,1.3}
optrscf;
Restarting a calculation

***,O2
file,2,O2.wfu
file,3,O2.aux
basis=vtz;
geometry={angstroms;o1;o2,o1,1.3}
optscf;
automatic geometry optimization

***, H2O test                !title
memory, 1, m                  !memory, 1Mw = 8Mb
geometry={o; h1, o, r; h2, o, r, h1, theta} !Z-matrix
basis=cc-pvdz
r=1 ang, theta=104            !geometry variables
hf                                !perform SCF calculation
optg                               !do SCF geometry optimization
FCI

hf

fci
FCI

- Only for tiny systems.
- For singlet system with $M$ orbitals and $N$ electrons

Number of CSFs = \[
\frac{M!(M + 1)!}{\left(\frac{N}{2}\right)!\left\{\frac{N}{2} + 1\right\}!\left(M - \frac{N}{2}\right)!\left(M - \frac{N}{2} + 1\right)!}
\]
Multi-Reference Configuration Interaction (MRCI)
casscf;
core,2,1,1,0,2,1,1,0;
closed,2,1,1,0,2,1,1,0;
occ,6,3,3,1,6,3,3,1
wf,36,5,2;
mrci
MR-ACPF and MR-AQCC

casscf;
core,2,1,1,0,2,1,1,0;
closed,2,1,1,0,2,1,1,0;
occ,6,3,3,1,6,3,3,1
wf,36,5,2;
acpf
Example PBS script on balt

#PBS -l nodes=1:ppn=2
#PBS -l arch=altix
#PBS -q altix
#PBS -l walltime=10:00:00
#PBS -l mem=23000mb
#PBS -e Ne2.e
#PBS -o Ne2.o
module add molpro
molpro -n 2 -d/scratch/blynch -m1000000 Ne2.com
Questions?

- blynch@msi.umn.edu
- help@msi.umn.edu