MOLDEN

FUNCTION

Molden - is a package for displaying **Mol**ecular **Den**sity

DESCRIPTION

MOLDEN is a package for displaying molecular density. It is tuned to the Ab Initio packages `GAMESS`* and `GAUSSIAN`. It can read all the information it needs from a `GAMESS` or `GAUSSIAN` outputfile. In this form it has been running on a **Convex C-120**, an **Apollo DN10000**, an **IRIS 4d/70gt**, a **DECstation 5000** and a **SUN**, and in slightly adapted versions on a **VAX** and a **Cray-YMP**. It should run smoothly on most Unix machines. See section `INSTALLATION GUIDE` on how to install **MOLDEN**. (* The `GAMESS` version referred to here is the european version maintaned by `M.F. Guest et al` not to be confused with the american version maintained by `M.W. Schmidt et al`)

MOLDEN was written by:


```
G. Schaftenaar
CAOS/CAMM Center Nijmegen
Toernooiveld, Nijmegen
The Netherlands
(1991)

e-mail adress : schaft@caos.kun.nl
```


FILES

MOLDEN reads two files, these are:

`Inputfile`

File holding the title and keyword lines (FORTRAN unit 5)

`GAMESS/GAUSSIAN` Outputfile

A file produced by running `GAMESS/GAUSSIAN`. The name of this file must be defined with the keyword **file** . This must be the output of a single `GAMESS/GAUSSIAN` run. Care must be taken not to turn off printing of vectors and/or basis set in the case of `GAMESS`. In the case of `GAUSSIAN` printing of basis set information has to be turned on by the use of the keyword **GFINPUT**. Printing of MO coefficients has to be turned on by use of the keyword **IOP(6/7=1)**. (`FORTRAN` unit 30)

MOLDEN writes two files, these are:

`Outputfile`

Results file. (`FORTRAN` unit 6)

`plotfile`

The Plotter/Screen File called `plot` (`FORTRAN` unit 15). At the end of the **MOLDEN** run, a Screen file can be displayed by :

```
- simply typing 'cat plot' on Unix machines
                'type plot.dat' on VAX machines
```

Depending on the configuration of the Plotter, the Plot file can be displayed in the same way or has to be sent to a queue (usually by a `SUBMIT` command or an `lpr` command).

INPUTFILE

The layout of Inputfile for a **MOLDEN** job is as follows;

```
<Title line>                    ( 80 columns maximum )
<First line of key words>       ( 80 columns maximum )
<Second line of key words>      ( 80 columns maximum )
```

Both capitals and lowercase can be used. The keywords taking no parameters usually can be abbreviated to four characters.

SPECIFICATION OF TERMINAL/PLOTTER TYPE

The following commands specify which terminal or plotter driver is to be used.

tek4014

Syntax: **tek4014**

The `Tektronix4014` graphics language is considered to be something of a standard. A lot of graphical terminals can emulate tek4014 although you probably need to tell your terminal it is to do so. Some PC's having a VT100 terminal emulation can also emulate tek4014 (it runs at least on an ATARI and in an impaired way on a Macintosh) (some HP terminals with a tek4014 emulation run tek4014 faster than the HP language) **Xwindows** has an application `xterm` in which you can turn on a tek window. (Under Unix this command is usually located in the directory `/usr/bin/X11`)

hpgl

Syntax: **hpgl**

hpgl stands for **H**ewlett **P**ackard **G**raphics **L**anguage. The HP plotters all speak this language. Also HP laserwriters understand **hpgl**.

hp2392a

Syntax: **hp2392a**

Most HP terminals with graphics capability probably will be able to understand these escape codes.

postscript

Syntax: **postscript**

Most Laserwriters understand **postscript**. Although sometimes you have to tell them it is not getting plain text but **postscript**.

xwindows

Syntax: **xwindows**

Most workstations have **Xwindows** capability (See section `INSTALLATION GUIDE` how to install **MOLDEN** with an **Xwindows** driver). Before running **MOLDEN** you have to define the display you will be working on. On `Unix` machines just type; **setenv DISPLAY :0** when you want to use the display of the machine **MOLDEN** is executing on, and for instance **setenv DISPLAY camms3.caos.kun.nl:0** when you want to use the display of the remote machine with `internet address` **camms3.caos.kun.nl**. On **VAX** machine, the same is accomplished by typing **set display/create /node=camms3.caos.kun.nl/trans=tcpip**. A window will be created at run time, holding the picture. A rectangular part of this picture can be magnified by pressing the left mouse button and holding it down while dragging it untill the rectangle has the required size. Pressing the middle mouse button subsequently will undo the magnification. Pressing the right mouse button will end the **MOLDEN** session. When the cursor is in the **MOLDEN** window, the arrow keys can be used to rotate the 3-d picture and **i** and **d** keys will respectively **i**ncrease and **d**ecrease the vertical scale op the 3-d plot. The letter **m** is a toggle, which turns on/off the displaying of the molecule in the 3-d picture.

figure

Syntax: **figure**

`Figure` is a Graphics language developed by the Genetics Computer Group. With it goes a package that can convert `Figure` to a variety of graphics/plotter languages.

silly

Syntax: **silly**

The Silicon Graphics 3D version is requested through the use of the keyword **silly**. This is an experimental version. (See section `INSTALLATION GUIDE` on how to install this version.)

DEFINITION of DENSITY

The following keys define ;

whether the `total` electron density has to be plotted or the sum of the density owing to `some specified` molecular orbitals. (**occu**, **valence**)

whether the electron density or the value of an molecular orbital itself has to be plotted (**homo**, **lumo**, **psi**, **phase**)

whether the molecular density minus `spherically averaged` atomic density has to be plotted (**bonds**) or whether for O,F,S and Cl oriented ground state atomic densities have to be substracted. (**bonds** in combination with **orient**)

whether the interatomic overlap density has to be plotted (**overlap**) or the atomic part of difference density (**atomic**). These are the two components that make up the density matrix used by **bonds** (possibly in combination with **orient**)

atomic

Syntax: **atomic**

Specifies that the electron density of the free atoms is to be subtracted from the molecular electron density as with the use of the keyword **bonds**, only now the contribution from the interatomic overlap is set to zero. In fact the density matrices used by **atomic** and **overlap** together make up the density matrix used by **bonds**. The density map predominantly will have a negative value, since some of the atomic density has been transferred to interatomic overlap density. However lone pairs usually show up as positive contributions in the density map. (As with **bonds** it can be used in combination with the keyword **orient**)

See also **bonds**, **orient** and **overlap** !!

bonds

Syntax: **bonds**

Not to be used in conjunction with **homo**, **lumo**, **psi** or **occu**. **bonds** subtracts the spherically averaged atomic density from the molecular density . The result is a plot whose average value is zero, and shows where the electrons have come from and gone to when the bonds are formed. However a problem can arise. Most atomic ground states are not spherically symmetric. Oxygen for example has a **3P** ground state f.i. $p_x^2 p_y^1 p_z^1$. The electron density now is maximal along the x-axis, so that in fact when an atom in the molecular environment has retained a lot of its ground state character it will have a preferable orientation. Subtracting a spherically symmetric atom $p_x^{4/3} p_y^{4/3} p_z^{4/3}$ in this case can result in subtracting too much in the y and z directions and too little in the x direction. As a result the density difference plot may show a misleading decrease in electron density along for example a C-O axis. In this case the keyword **orient** must be used.(See also **orient**, **overlap** and **atomic**) This effect is usually most pronounced for the atoms O,F,S, and Cl, whereas for example Carbon in the molecular environment usually has a lot of spherically symmetric character.

bonds can only be used when one of the following basissets have been employed; **sto3g**, **3-21g**, **4-31g**, **6-31g**, optionally with polarisation functions (*, or **). They may differ per atom however and the atoms must be in the range Hydrogen up to Chlorine.

```
Hint : for contour plots cut=0.1 is recommended
       for 3d plots mult=20 is recommended
```

homo

Syntax: **homo**

For closed-shell systems with non-degenerate **H**ighest **O**ccupied **M**olecular **O**rbitals, the keyword **homo** can be used to produce an intensity map of the highest occupied molecular orbital. For other systems, the keyword **psi** should be used.

lumo

Syntax: **lumo**

For closed-shell systems with non-degenerate **L**owest **U**noccupied **M**olecular orbitals, the keyword **lumo** can be used to produce an intensity map of the lowest unoccupied molecular orbital. For other systems, the keyword **psi** should be used.

occu

Syntax: for example; **occu** = (1-22/0,9/1.0,10/2)

When the user wants to explicitly define an electronic configuration for a system, overriding the occupancies read in from the `GAMESS/ GAUSSIAN` output file, the keyword **occu** has to be used. Taking the occupancies read in from the outputfile as a starting point, the orbital occupancies in the example are modified in the following way, in sequential order;

Orbital	**Occupancy**
1-22	0
9	1
10	2

For the use with **U**nrestricted **H**artree **F**ock wavefunctions the keywords **occa** (for the `alpha` electrons) and **occb** (for the `beta` electrons) have to be used.

orient

Syntax: **orient** or **orient** = (n1,n2/n.nn/n.nn,...)

To be used in conjunction with the keyword **bonds** and not to be used in conjunction with **homo**, **lumo**, **psi** or **occu**. **bonds** subtracts the the `spherically averaged` atomic density from the molecular density . The result is a plot whose average value is zero, and shows where the electrons have come from and gone to when the bonds are formed. However a problem can arise. Most atomic ground states are **not** spherically symmetric. Oxygen for example has a **3P** ground state f.i. $p_x^2 p_y^1 p_z^1$ the electron density now is maximal along the x-axis ,so that in fact when an atom in the molecular environment has retained a lot of its ground state character it will have a preferable orientation. Subtracting a spherically symmetric atom $p_x^{4/3} p_y^{4/3} p_z^{4/3}$ in this case can result in subtracting too much in the y and z directions and too little in the x direction. As a result the density difference plot may show a misleading decrease in electron density along for example a C-O axis. This effect is usually most pronounced for the atoms `O`,F,S, and `Cl`, wheras for example Carbon in the molecular environment usually has a lot of spherically symmetric character.

When using **orient** without parameters, the atomic density (**Datom**) of `O`,F,S and `Cl` atoms will be oriented in such a way that the sum of **(Dmol**(i,j)-**Datom**(i,j)**)**^{2} (delta squared on the output file) is at a minimum. (**Dmol** being the atomic part of the molecular density matrix and i and j run over the p_x,p_y,p_z Atomic orbitals). This is done per atom.

When using **orient = (n1,n2,..)** only the specified atoms are oriented. For all others the spherically averaged atomic density is used. Checked is whether atom number **n1**,**n2** etc are `O`,F,S or `Cl` atoms.

When using **orient = (n1,n2/n.nn/n.nn,...)** the automatic orientation mechanism is overridden for atom **n2**. Instead it is oriented using the two angles supplied on the keyword after the slashes (alfa and beta on the output file). For example the atomic ground state density of Oxygen has an oval symmetry, having one direction in which 2 electrons participate and two perpendicular directions in which 1 electron each participates. The supplied angles define the direction of the unique axis. If the atom in the molecular environment has lost a lot of its ground state character the automatic orientation mechanism can give physically meaningless orientations. In the ultimate case when the oxygen has a pure spherical O^{2-} character, the use of oriented ground state densities is clearly erroneous. When the atomic part of the molecular density matrix shows one direction in which 1 electron participates and two directions in which 1.5 electrons each participate, the automatic orientation mechanism may provide a direction which results in a decrease in electron density at the middle of a bond axis. Here orientation by hand can result in a density-difference plot with the expected increase in electron density at the middle of a bond axis.

orient can only be used when one of the following basissets have been employed ; **sto3g**, **3-21g**, **4-31g**, **6-31g** optionally with polarisation functions (*, or **). They may differ per atom however and the atoms must be in the range Hydrogen up to Chlorine.

```
Hint : for contour plots cut = 0.1 is recommended
       for 3d plots mult = 20 is recommended
```

(see also `Chemical Deformation Densities`, W.H.E. Schwarz, K. Ruedenberg and L. Mensching, J. Am. Chem. Soc. 1989,**111**,6926-6933, where the orientation used here is termed '`naive`')

overlap

Syntax: **overlap**

The interatomic overlap density will be plotted. This is done by setting the atomic part of the Molecular density matrix to zero. The density map will predominantly have positive values with maxima roughly at midway the bond axes.

phase

Syntax: **phase**

phase is used to to reverse the sign of a Molecular Orbital. Used in conjuction with **psi**.

psi

Syntax: **psi = nn**

A specified molecular orbital is to be plotted (see also **phase**).

valence

Syntax: **valence**

Since the inclusion of non-valence electrons in ab initio calculations results in the predominance of the inner shell electron density on the total electron density, the interpretation of the chemically more interesting valence electron density is clouded. The use of **valence** results in non occupying those molecular orbitals which predominantly contain inner-shell electron density. It only works when your molecule exclusively consists of atoms from `H` to `Ar`. Otherwise you should use the keyword **occu**.

DEFINITION OF THE PLOTPLANE

The plane of the plot is essentially defined by :

The center of the plot (**(PX,PY,PZ)** on the **MOLDEN** outputfile)

The vector perpendicular to the plane of the plot (**(CX,CY,CZ)** on the **MOLDEN** outputfile)

The **edge** keyword specifying the size of the square plot

The center and vector perpendicular to the plane of the plot can either be specified ;

directly in cartesian coordinates by using the absolute form of the keywords **center** and **line**.

indirectly, by using the coordinates of the atoms on the **plane** and **rot** keywords (or the `atomic` form of the keywords **center** and **line**).

The use of **center/line** and **plane**(**/rot**) are mutually exclusive. Of these, the **plane**(**/rot**) keywords are the most flexible and easy to use. **plane** defines the plane of the plot by specifing three atoms and optionally this plane can then be rotated round the axis formed by two of these three atoms by use of the keyword **rot**. The remaining atom will then no longer lie in the plane of the plot.

Finally, when having defined the plane of the plot by either **center/line** or **plane**(**/rot**) the plane can be translated along the vector perpendicular to the plane by use of the keyword **lift**.

plane

Syntax: **plane = (n,n,n)**

This keyword is essential, it defines the plane of the plot. For example **plane=(1,2,3)** means the first three atoms in the coordinate list define the plane of the plot. The center of the triangle will be taken as center of the plot. The keywords **center/line** and **plane** are mutually exclusive.

rot

Syntax: **rot = (n,n,nnn.nn)**

This keyword is optional, it can only be use in conjunction with the keyword **plane**. For example, when previously having defined the plane of the plot by the first three atoms (**plane=(1,2,3)**), this plane can be rotated round the axis formed by atoms 2 and 3 by 45 degrees, by using **rot=(2,3,45.0)**, atom 1 will now no longer be part of the plane of the plot . The center of the plot has been shifted to the point midway the line connecting atoms 2 and 3.

center

Syntax: **center = nn** or **center = (n.nn,n.nn,n.nn)**

This keyword is essential, it defines the center of the plot Two formats are provided to define the center: (**a**) an atom number can be used, and (**b**) an absolute cartesian coordinate can be specified. Irrespective of which option is used, the center of the plot will be converted internally into absolute cartesian coordinates.

`Atomic`

Syntax: **center = n**. The location of atom **n** is defined as the center of the plot. Thus if atom n has cartesian coordinates (x=0.5, y=1.4, z=-0.8) then the center of the plot is (x=0.5, y=1.4, z=-0.8). Dummy atoms are not counted, so if any dummy atoms were used in the definition of the geometry, the atoms will have been renumbered (see the **MOLDEN** output file; coordinates section for the new numbering)

`Absolute`

Syntax: **center = (n.nn,n.nn,n.nn)** .The location of the center of the plot is defined as being (n.nn,n.nn,n.nn). Of course, before such a center can be defined, the user must know the cartesian coordinates of the atoms in the molecule.

line

Syntax: **line = nn** or **line = (n.nn,n.nn,n.nn)**

This keyword is essential, it defines a vector perpendicular to the plane of the plot. Two formats are provided to define the axis perpendicular to the plane of the plot. These formats use radically different concepts, so users are cautioned to verify that they understand both definitions, and the distinction between them. (**a**) an atom number can be used, and (**b**) an absolute cartesian coordinate can be specified. Irrespective of which option is used, the axis of the plot will be converted internally into a unit vector in cartesian coordinates.

`Atomic`

Syntax: **line = n** The axis of the plot is defined by the vector drawn from atom **n** to the defined center of the plot. Thus if atom n has cartesian coordinates (x=0.5, y=1.4, z=0.2) and the center of the plot is at point (x=0.5, y=1.4, z=-0.8) then the axis of the plot is (0.0, 0.0, 1.0). Dummy atoms are not counted, so if any dummy atoms were used in the definition of the geometry, the atoms will have been renumbered (see the **MOLDEN** output file, coordinates section for the new numbering)

`Absolute`

Syntax: **line = (n.nn,n.nn,n.nn)** The axis of a line perpendicular to the plane of the plot is (n.nn,n.nn,n.nn). This axis need not be normalized, but must be finite, that is the only axis not allowed is (0,0,0).

lift

Syntax: **lift = n.nn**

This keyword is optional. When having defined the plane of the plot by either **center/line** or **plane**(**/rot**) the plane can be translated along the vector perpendicular to the plane by use of the keyword **lift**. The length of this vector is specified as **n.nn** Atomic Units.

edge

Syntax: **edge = n.nn**

The length of the sides of the graph-plot is defined as being n.nn **A**tomic **U**nits (1 **AU** = 0.529177 Angstrom).

DEFINITION OF THE PLOTTYPE

Two kinds of plots are possible;

a contour plot (See **contour**)

a three dimensional plot (See **3d**)

contour

Syntax: **contour**

The use of **contour** requests a contour plot. This is the default, so that its specification is not required, except when the combination of a 3-D grid plot with a contour plot is desired. (See also **cut**, **fine** and **step** .)

cut

Syntax: **cut = n.nnn**

This keyword sets the maximum contour value at n.nnn times the maximum encountered density/intensity. **cut** should be within the range 0.0 - 1.0. This keyword was specially devised for use with gaussians since the inclusion of non-valence electrons in ab initio calculations results in the predominance of the inner shell electron density on the total electron density, which clouds the interpretation of the chemically more interesting valence electron density.

fine

Syntax: **fine**

Normally between 10 and 25 contours are plotted. In order to increase this number **fine** can be used, in which case 40 to 100 contours will be generated.

step

Syntax: **step = n.nn**

Normally between 10 and 25 contours are plotted. The program itself figures out which contour interval should be used. Sometimes the program comes up with different contour intervals for plots one would like to compare. In this case **step** should be used to force a contour interval used by previous runs.

3d

Syntax: **3d**

3d can be used to create a 3-D mesh of lines plot. **3d** can be used in conjuction with the keyword **axis** specifying the view direction. When **axis** is omitted a value of 0.5 is used. **3d** can also be used in combination with **contour**. (See also **axis** and **mult**)

axis

Syntax: **axis = n.nnn**

axis determines the view direction when a three dimensional plot is requested by the use of the keyword **3d**. The range of n.nn in **axis = n.nn** is 1.0 to 0.0. Where 1.0 would give a square plot, as if the user was viewing the plot from directly overhead looking straight down, 0.0 gives a view of the plot as if the user was looking at it from the horizon, looking horizontally. Clearly, **axis=1.0** would not show the relief. If **3d** was used only a perfectly square grid would be seen. A better choice would be **axis = 0.6**. Conversely, **axis = 0.0** would show the contours as straight lines (as they would be being viewed edge-on).

mult

Syntax: **mult = n.nn**

There is a default scale for the relief of a plot, when viewed as a 3-D structure. If this default is not suitable, say the plot is too flat, then **mult = n.nn** can be used to change the vertical scale. **mult=1.0** will do nothing, **mult = 2.0** will increase the vertical relief. Use in conjunction with the keyword **3d**. (See also **3d** and **axis**)

MISCELLANEOUS COMMANDS

before

Syntax: **before**

In case `GAMESS/GAUSSIAN` has performed a geometry optimise/saddle run, two sets of vectors are available from the outputfile, those before optimisation and those after. Specifying **before** results in using the vectors and geometry before optimisation.

debug

Syntax: **debug**

This keyword is in fact a way of getting a lot more output, such as: basis set information, vectors, density matrices and so on.

file

Syntax: **file** = `filename`

This keyword must be used to specify the `GAMESS/GAUSSIAN` outputfile from which is read. This must be the output of a single run and care must be taken not to turn off printing of vectors.

(When using the keyword **molpot** the output of a `DMA` analysis by `GAMESS` is expected, without the output of the corresponding startup job.)

generate

Syntax: **generate**

This keyword results in the writing of the density matrix in `FORTRAN` data statements. It is used to generate atomic data for use by the program internally. (of course a preceding atomic run by `GAMESS` has to be performed, do not use with a `GAUSSIAN` outputfile)

molpot

Syntax: **molpot**

This keyword can only be used when the output of a `DMA` analysis by `GAMESS` is supplied. In a **D**istributed **M**ultipole **A**nalysis the molecular electron density and atomic charges are fitted by an expansion of multipoles. With these multipoles the electrostatic potential of a point charge in the field of the molecule can be calculated and plotted. The `DEFINITION DENSITY` keywords are nolonger active when using **molpot**.

(see A.J. Stone and M. Alderton, `Molecular Physics`, 1985, **56**, 1047-1064)

EXAMPLES

BASIS SET EFFECTS

When using the minimal basis set **sto3g** together with the **bonds** option and comparing the electron density of the free atoms with that of the molecule, one usually finds an `increase` of electron density at the middle of the bonds axes and a `decrease` of electron density at the atoms themselves. Doing the same for the split-valence basis sets, such as **3-21g**, **4-31g** and **6-31g**, one also finds an `increase` of electron density at the middle of the bond axes, but also an `increase` of electron density close to the atom centres themselves is observed. A decrease of electron density is found in the outer regions of the molecule. This reflects the capability of the split-valence to yield a more contracted atom on going from the free atoms to the molecule. The minimal **sto3g** basis set clearly does not have this flexibility.

This effect is illustrated for the Hydrogen molecule with **sto3g** and **4-31g** basis sets. (See **Figures 1** and **2** respectively)

Figure 1

Inputfile:

```
1:  H2 sto3g Molecular Density minus atoms
2:  center=(0.0,0.0,0.0) line=(1.0,0.0,0.0) cut=0.1 edge=10.0
3:  bonds file=h2stoout postscript debug
```

Figure 2

Inputfile:

```
1:  H2 4-31g molecule minus atoms
2:  center=(0.0,0.0,0.0) line=(1.0,0.0,0.0) cut=0.1
3:  edge=10.0 post debug bonds file=h2gam431out
```

RING STRAIN

Figure 3 shows the electron density of the `benzene` molecule minus that of the free atoms. An increase of electron density at the `middle` of the C-C bond axes is observed. **Figure 4** shows the result of the same procedure followed for the `cyclopropane` molecule. Here the increase of electron density is clearly positioned `outside` the C-C bond axes. The overlap of atomic orbitals of the different C atoms is not optimal because of ring-strain. This effect is also known as the formation of banana bonds in cyclopropane.

Figure 3

Inputfile:

```
1:  benzene sto3g molecular minus atomic density
2:  plane=(1,3,5) edge=13.22 bonds cut=0.08
3:  file=benzene.out postscript
```

Figure 4

Inputfile:

```
1:  cyclopropane sto3g molecular density minus atomic
2:  plane=(1,2,3) edge=10.0 bonds cut=0.08
3:  file=cyclopropaan.out postscript
```

THE USE OF KEYWORD ORIENT

When using the keyword **bonds**, default the `spherically averaged` electron density of the atoms is subtracted form the molecular density. However some atoms have a groundstate that strongly deviates from the spherical symmetry. Oxygen for example has a 3P ground state, characterised by one direction in which 2 electrons participate and two perpendicular directions in which 1 electron each participates, this might be termed as an oval symmetry. If the atoms within the molecule have retained a lot of this oval symmetry, then subtracting spherically averaged atoms would result in subtracting too little (4/3 electrons) in the direction in which 2 electrons participate and too much(4/3 electrons) in the directions in which 1 electron each participate. The latter are the directions in which oxygen will form bonds and the overall result would be a seemingly decrease of electron density along the bond axes (See **Figure 5**, the planar conformation of Hydrogen Peroxide).

By using the keyword **orient**, the true atomic ground state density, after optimising its orientation within the molecule, is subtracted from the molecular density. This leads to the expected increase in electron density along the `O-O` axis in Hydrogen Peroxide (see **Figure 6** and **7**)

Figure 5

Inputfile:

```
1:  H2O2 sto3g Molecular Density minus Spherical atoms
2:  plane=(2,3,4) edge=7.0 bonds cut=0.1
3:  file=h2o2out postscript debug
```

Figure 6

Inputfile:

```
1:  H2O2 sto3g Molecular Density minus Oriented Oxygens
2:  plane=(2,3,4) edge=7.0 bonds orient cut=0.1
3:  file=h2o2out postscript debug
```

Figure 7

Inputfile:

```
1:  H2O2 sto3g Molecular Density minus Oriented Oxygens
2:  plane=(2,3,4) edge=7.0 post debug 3d
3:  file=h2o2out bonds orient mult=20
```

MORE ON THE USE OF THE KEYWORDS BONDS, ORIENT, ATOMIC AND OVERLAP

The use of the keywords **bonds**,**orient**,**atomic** and **overlap** is illustrated for the $NH_2COOCH_3^{+\cdot}$. molecule. Removing an electron from the neutral species results in an extra distortion of the atomic contribution to the molecular density. Before the geometry optimisation of this molecule, the automatic orientation mechanism of **orient** provides a chemically sensible orientation for both oxygens (See **Figure 8**). After the geometry optimisation however, the automatic orientation mechanism fails for the second oxygen, resulting in a decrease in electron density along the `C-O-C` axes (See **Figure 10**). Looking at the part of the density matrix involving p_x,p_y and p_z orbitals of the oxygens, it is clear to see why. For the first oxygen the p-part of the density matrix resembles that of the oxygen ground state. For the second oxygen we see a heavily distorted oxygen ground state (this is reflected in a larger value of delta squared). Where the oxygen ground state has one direction in which 2 electrons participate and two perpendicular directions where 1 electron each participate. The second oxygen has one direction in which roughly 1 electron participates (the y direction) and 2 perpendicular directions in which roughly 1.5 electrons each participate (the x and z directions). The automatic orientation mechanism now will try to align the unique atomic ground state axis with one of the 1.5 electrons axes in the molecule, after the geometry optimisation the balance topples over in favour of the x-direction. By forcing the unique axis to lie in the z-direction, a chemically sensible picture re-emerges in which an increase of electron density midway the C-O-C bond axes can be seen (See **Figure 11**).

Figure 8: Molecular density minus `oriented` groundstate atomic density

- Before geometry optimisation

Inputfile:

```
1:  NH2COOCH3+· sto3g Mol. Density - Oriented Atoms before
 Optimisation
2:  plane=(1,2,3) edge=12.0 bonds orient cut=0.1
3:  before file=carbso postscript
```

Excerpt from the **MOLDEN** outputfile:

```
   ----  O sto3g    -----------------------------------------

   p-part atomic density matrix before orientation

                2.0000   0.0000   0.0000
                0.0000   1.0000   0.0000
                0.0000   0.0000   1.0000


   delta squared                  =  8.092850549116692E-002
   alfa  optimised                =                 0.
   beta  optimised                =   90.0000000000000


   p-part molecular density matrix

                0.9728   0.0184   0.0000
                0.0184   0.9900   0.0000
                0.0000   0.0000   1.7182

   p-part oriented atomic density matrix

                1.0000   0.0000   0.0000
                0.0000   1.0000   0.0000
                0.0000   0.0000   2.0000

   ---- O sto3g   The C-O-C Oxygen --------------------------

   p-part atomic density matrix before orientation

                2.0000   0.0000   0.0000
                0.0000   1.0000   0.0000
                0.0000   0.0000   1.0000
```

```
delta squared                  =  0.486484885670038
alfa  optimised                =                 0.
beta  optimised                =   90.0000000000000


p-part molecular density matrix

             1.4122  -0.1816   0.0000
            -0.1816   0.8539   0.0000
             0.0000   0.0000   1.5212

p-part oriented atomic density matrix

             1.0000   0.0000   0.0000
             0.0000   1.0000   0.0000
             0.0000   0.0000   2.0000
```

Figure 9: Molecular density minus `spherically averaged` atomic density

- After geometry optimisation

Inputfile:

1: $NH_2COOCH_3^{+\cdot}$ sto3g Mol.Density - Spherical Atoms after Optimisation
2: **plane**=(1,2,3) **edge**=12.0 **bonds cut**=0.1
3: **file**=carbso **postscript**

Figure 10: Molecular density minus `oriented groundstate` atomic density

- After geometry optimisation

Inputfile:

1: $NH_2COOCH_3^{+\cdot}$ sto3g Mol. Density - Oriented Atoms after Optimisation
2: **plane**=(1,2,3) **edge**=12.0 **bonds orient cut**=0.1
3: **file**=carbso **postscript**

Excerpt from the **MOLDEN** outputfile:

```
---- O sto3g ---------------------------------------------

p-part atomic density matrix before orientation

             2.0000   0.0000   0.0000
             0.0000   1.0000   0.0000
             0.0000   0.0000   1.0000


delta squared                  =  7.890606938493698E-002
```

```
alfa  optimised                =                 0.
beta  optimised                =   90.0000000000000


p-part molecular density matrix

             0.9712   0.0198   0.0000
             0.0198   0.9809   0.0000
             0.0000   0.0000   1.7226

p-part oriented atomic density matrix

             1.0000   0.0000   0.0000
             0.0000   1.0000   0.0000
             0.0000   0.0000   2.0000

---- O sto3g  The C-O-C Oxygen ---------------------------

p-part atomic density matrix before orientation

             2.0000   0.0000   0.0000
             0.0000   1.0000   0.0000
             0.0000   0.0000   1.0000


delta squared                  =  0.517238577646048
alfa  optimised                =   18.3000000000000
beta  optimised                =   180.000000000000

p-part molecular density matrix

             1.4285  -0.2000   0.0000
            -0.2000   0.8900   0.0000
             0.0000   0.0000   1.4804

p-part oriented atomic density matrix

             1.9014  -0.2981   0.0000
            -0.2981   1.0986   0.0000
             0.0000   0.0000   1.0000
```

Figure 11: Forced orientation of the second Oxygen

- After geometry optimisation

Inputfile:

1: $NH_2COOCH_3^{+\cdot}$ sto3g Mol. - Atoms forced orientation C-O-C oxygen
2: **plane**=(1,2,3) **edge**=12.0 **orient**=(2,3/0/90) **bonds cut**=0.1
3: **file**=carbso **postscript debug**

Excerpt from the **MOLDEN** outputfile:

```
---- O sto3g   The C-O-C Oxygen -------------------------

p-part atomic density matrix before orientation

             2.0000   0.0000   0.0000
             0.0000   1.0000   0.0000
             0.0000   0.0000   1.0000


delta squared                  =  0.545626478874887
alfa  optimised                =                 0.
beta  optimised                =   90.0000000000000


p-part molecular density matrix

             1.4285  -0.2000   0.0000
            -0.2000   0.8900   0.0000
             0.0000   0.0000   1.4804

p-part oriented atomic density matrix

             1.0000   0.0000   0.0000
             0.0000   1.0000   0.0000
             0.0000   0.0000   2.0000
```

Interatomic overlap plots

Figure 12: Interatomic overlap contour plot

Inputfile:

1: sto3g $NH_2COOCH_3^{+\cdot}$ Interatomic Overlap Density
2: **plane**=(1,2,3) **edge**=12.0 **overlap**
3: **file**=carbso **postscript debug**

Figure 13: Interatomic overlap 3-d plot

Inputfile:

1: sto3g $NH_2COOCH_3^{+\cdot}$
2: **plane**=(1,2,3) **edge**=12.0 **3d postscript debug**
3: **file**=carbso **overlap**

ATOMIC deformation

Figure 14:

```
- Molecular density minus oriented groundstate Atomic density
- atomic deformation only
- Forced Orientation of second Oxygen
- After Geometry Optimisation
```

Inputfile:

```
1:  NH2COOCH3+· sto3g Mol.- Atoms forced orientation C-O-C Oxygen
 Atomic
2:  plane=(1,2,3) edge=12.0 bonds cut=0.1
3:  file=carbso postscript orient=(3,3/0/90) atomic
```

specification of **bonds** is not necessary when using keyword **atomic**.

INSTALLATION GUIDE

For some machines **Molden** contains some machine specific code (Particulary for the **VAX** and in lesser extent for the **Cray** and the 3-D version for the **Silicon Graphics**). Before compiling **Molden** these lines have to be made active, by removing the comment in front of these lines, like `cvax`, `ccray` or `csg`. For the **VAX** and the **Silicon Graphics**, these lines are exclusively located at the back of the program source. Depending on the local situation a compiler can run out of space, in this case you should chop **Molden** up in two pieces and try again.

When you want to use the **XWindows** driver of **Molden**, the last subroutine of **Molden**,`xwin` has to be deleted or commented out. In addition the file `xwin.c` has to be compiled with a **C** compiler. On Unix machines it is assumed that the following files are present:

file	**directory**
libX11.a	/usr/lib
Xlib.h	/usr/include/X11
Xutil.h	/usr/include/X11
Xos.h	/usr/include/X11
keysym.h	/usr/include/X11

If you do have these files, but they are not in the directories listed above, you can use the `-I`directory option on the **cc** command to tell the **C** compiler where the include files are. And you can specify the `-L`directory option in front of the `-l`X11 option to tell the loader where the **X** library file is to be found.

Following are the specific instructions per machine (If your machine is not one of the below and it is a Unix machine, try the **DEC/Ultrix** instructions).

VAX/VMS

Type: **rename molden.f molden.for** .Remove the comment cvax from lines starting with it.

Installation of **Molden** without the **XWindows** driver.

```
$ fortran/contin=90 molden
```

```
$ link molden
```

Installation of **Molden** with the **XWindows** driver; Create the file MAKE.COM containing the following lines.

```
$ fortran/contin=90 molden.for
$ cc/define="VMS" xwin.c
$ define lnk$library sys$library:vaxcrtl
$ link molden,xwin,sys$input/opt
sys$share:decw$xlibshr/share
```

Execute it by typing **@MAKE**.

To run **Molden**, create the file MOLDEN.COM containing the following lines;

```
$ assign 'p1' for005
$ assign 'p2' for006
$ run cc:[schaft.molden]molden.exe
$ deassign for005
$ deassign for006
```

Adapt the reference to the directory where **MOLDEN** is located. Execute it by typing **@molden inputfile outputfile**.

DEC/Ultrix

Installation of **Molden** without the **XWindows** driver.

```
f77 -o molden molden.f
```

Installation of **Molden** with the **XWindows** driver.

```
f77 -c molden.f
cc -c xwin.c
f77 -o molden molden.o xwin.o -lX11
```

To run **Molden**:

```
molden <inputfile >outputfile&
```

SUN

Installation of **Molden** without the **XWindows** driver.

```
f77 -Nl90 -o molden molden.f
```

Installation of **Molden** with the **XWindows** driver.

```
f77 -Nl90 -c molden.f
cc -c xwin.c
f77 -o molden molden.o xwin.o -lX11
```

To run **Molden**:

```
molden <inputfile >outputfile&
```

Apollo

Installation of **Molden** without the **XWindows** driver.

```
f77 -c molden.f
ld -o molden -A stacksize,FFFFF molden.o
```

Installation of **Molden** with the **XWindows** driver.

```
f77 -c molden.f
cc -c xwin.c
ld -o molden -A stacksize,FFFFF
      molden.o xwin.o -lX11
```

To run **Molden**:

```
molden <inputfile >outputfile&
```

Convex

Installation of **Molden** without the **XWindows** driver.

```
fc -o molden molden.f
```

Installation of **Molden** with the **XWindows** driver.

```
fc -c molden.f
cc -c xwin.c
fc -o molden molden.o xwin.o -lX11
```

To run **Molden**:

```
molden <inputfile >outputfile&
```

Cray under Unicos

The source lines specific for the **Cray** version can be made active by typing:

```
vi molden.f
:
1,$s/^ccray//
:
x
```

Installation of **Molden** without the **XWindows** driver.

```
cft77 molden.f
segldr -o molden molden.o
```

Installation of **Molden** with the **XWindows** driver.

```
cft77 molden.f
cc -c xwin.c
segldr -o molden molden.o xwin.o -lX11
```

To run **Molden**:

```
molden <inputfile >outputfile&
```

Silicon Graphics

The source lines for the 3-D real time rotation version can be made active by typing:

```
vi molden.f
:
1,$s/^csg//
:
x
```

Installation of **Molden** without the **XWindows** driver, and with the 3-D real time rotation capability, activated by the use of the keyword **silly**;

```
f77 -o molden molden.f -lfgl -lgls
```

Installation of **Molden** with the **XWindows** driver; and with the 3-D real time rotation capability;

```
f77 -c molden.f
cc -c xwin.c
f77 -o molden molden.o xwin.o -lfgl -lgls -lX11
```

To run **Molden**:

```
molden <inputfile >outputfile&
```