Reference Manual for mohelper

v1.0

Viktor Altergott

April 4, 2026

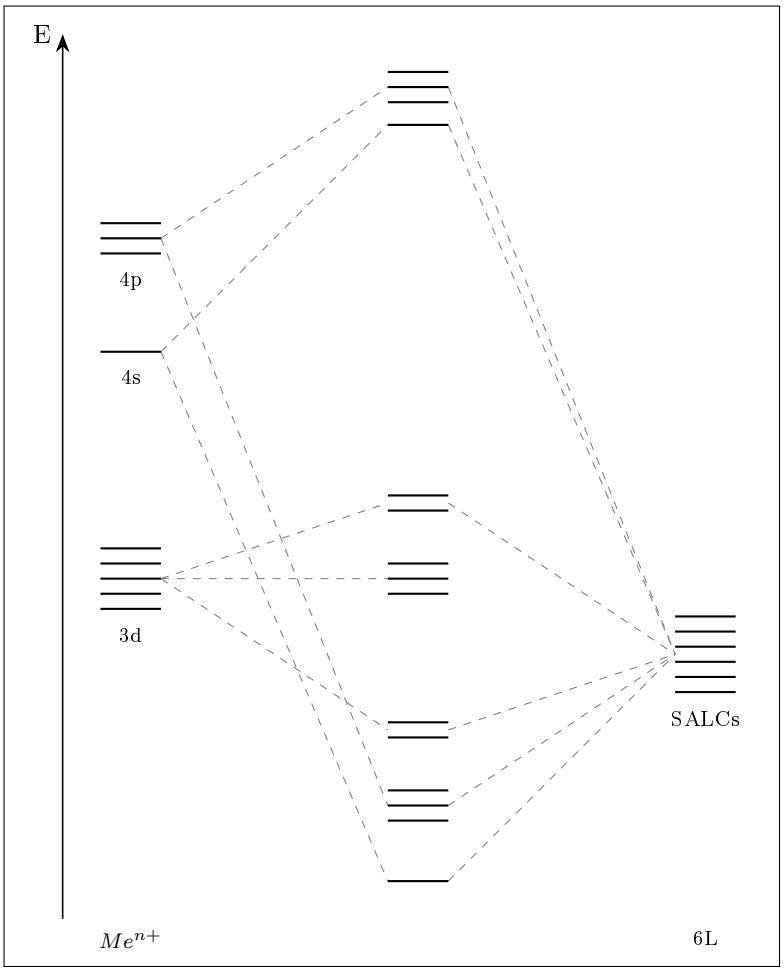


Figure 1: Octahedral Complex

Contents

1 Introduction

Stop fighting with pure TikZ. mohelper is a LaTeX package designed to create professional Molecular Orbital (MO) diagrams with minimal code. It handles the positioning, electron population, and dashed connection lines automatically, so you can focus on the chemistry, not the coordinates. Have a look:

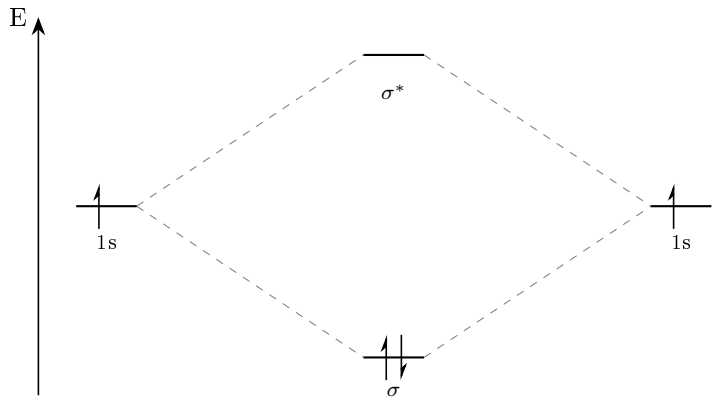


Figure 2: Hydrogen Molecule

```
\begin{mohelper}
    \fragment{
        \addOrbital{energy=0,label=1s,config=up}
    }
    \fragment{
        \addOrbital{energy=-2,label=$\sigma$,config=pair}
        \addOrbital{energy=2,label=$\sigma^*$}
    }
    \fragment{
        \addOrbital{energy=0,label=1s,config=up}
    }
    \addEnergyScale{0}
\end{mohelper}
```

2 Quick Start

To use the diagrams generated by this tool in your LaTeX project, follow these steps:

2.1 Include the Package

2.1.1 From CTAN

```
\usepackage{mohelper}
```

Include this into your preamble. Note that this might not be the newest version. In case you depend on using a newer version go to 2.1.2.

2.1.2 From Local

Download the package and include it into your project. To do so download the `mohelper.sty` file from

1. Releases (recommended, stable)
2. Repository (only if you know what you are doing)

and paste the file into your project's root directory. Ensure you have the mohelper package included in your preamble:

```
\usepackage{mohelper}
```

If you put `mohelper.sty` into a subdirectory, you must specify its path. Therefore, have this example shown:

```
\usepackage{./lib/mohelper}
```

Note that this method will throw a compiler warning because it is trying to find a package called `/lib/mohelper`. However, it still finds the package and it should work. Still, to avoid this warning put the package into your project's root directory, and input only through package name.

2.2 Configure your Diagram

Open the Web Helper and set up your orbitals, energies, and labels.

2.3 Copy and Paste

Copy the generated LaTeX code and paste it into your `figure` environment or document body.

3 Dependencies

`mohelper` uses some other pakets which are listed here:

- `tikz`, and some `tikzlibraries`
  - `arrows.meta`
  - `backgrounds`
- `xstring`
- `etoolbox`

4 Basic Usage

The heart of the package is the `mohelper` environment. It follows a multi-column logic:

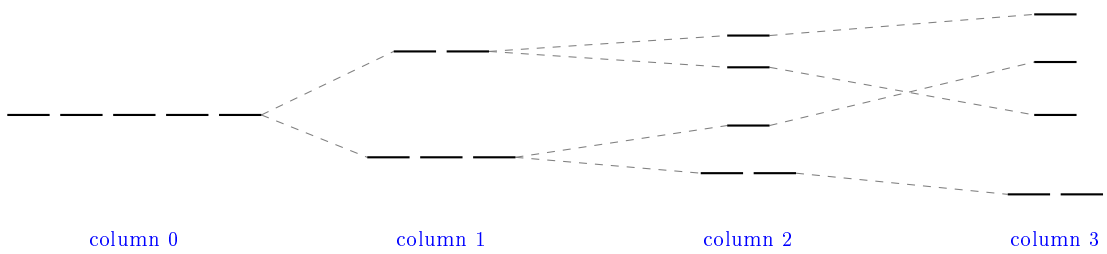


Figure 3: Ligand Field Splitting in Octahedral, Jahn-Teller, Square Planar

The package therefore extends above normal MO diagrams. `mohelper` allows having indefinite columns. They are not hard-coded to `left`, `middle` and `right`.

5 Macro Reference

5.1 The Environment

```
\begin{mohelper}[options]
    ...
\end{mohelper}
```

Wraps a `tikzpicture`. You can pass standard TikZ options like `[scale=0.8]`.

5.2 Defining Fragments

Each MO diagram consists of fragments. Each fragment is represented by a column. A standard MO diagram has 3 fragments.

- Left Atom
- Molecule
- Right Atom

Figure 3 has 4 fragments. It is quite easy when you forget the fragment part and just think about columns.

```
\fragment{
    ... % draw orbitals, etc.
}
```

The `\fragment` command builds a new column into your diagram. It acts as a shell in which you have to define orbitals etc. All important definitions are done within these scopes.

5.2.1 Orbitals

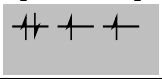
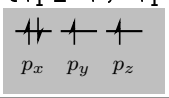
```
\addOrbital{ [attributes] }
```

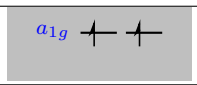
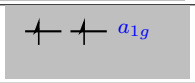
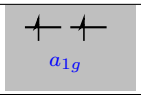
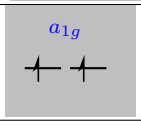
Adding orbitals is easier than you think. Between the scopes you need to set attributes. The syntax[1] looks like this:

```
\addOrbital{
    attribute1=value1,
    attribute2=value2,
    ...
}
```

The following table shows all attributes and their corresponding values.

[1] It is managed using an external library called `pgfkeys`.

Attribute	Description	Example
`energy`	**Y-Level** (Energy)	`-2.5`
`sym`	**Symmetry Group** (irreducible representation)	`t2g`, `a1`, `eg` (no special characters allowed!)
`config`	**Electron Configuration**, Array of spin values (Format: `{spin1, spin2, spin3}`)	`up`, `down` or `pair` (any other value will result in an empty orbital) example: `{pair, up, up}` 
`labels`	**Individual Labels Array**: label each orbital of the degenerated group individually	`{$p_x$, $p_y$, $p_z$}` 
`label`	**Single Label**: a single label for all degenerated orbitals	`$a_{1g}$`

`labelposition`	**Position of Label**: this affects only the single label, individual labels are untouched	`left`
		 
		`right`
		 
		`bottom`
		 
		`top`
		 

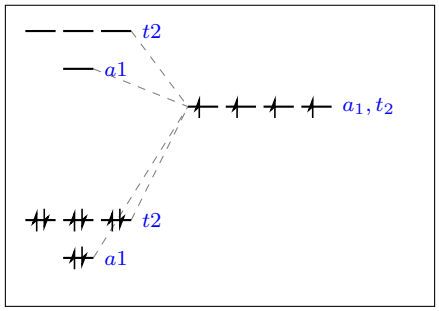
Note: It is possible to execute `\addOrbital` outside of `\fragment`. To do so, you must specify the additional attribute `column` with a corresponding number. However, drawing fragment labels (5.2.2) or energy difference arrows (5.3.2) is not possible outside of `\fragment`. It is not recommended to do that, as it will result in strange bugs!

Important Attributes

energy comes in pakets, quants. Therefore all states are quantized and $E = h \cdot \nu$. You know this theorem very well. This means for the diagram that the energy level you specify will mark the y-coordinate in the diagram. The diagram does not use any fancy scale like eV. Instead it uses paper units whereas 1unit $\approx 1cm$. It is as if you would draw the diagram on paper. If not specified the standard value is `0`.

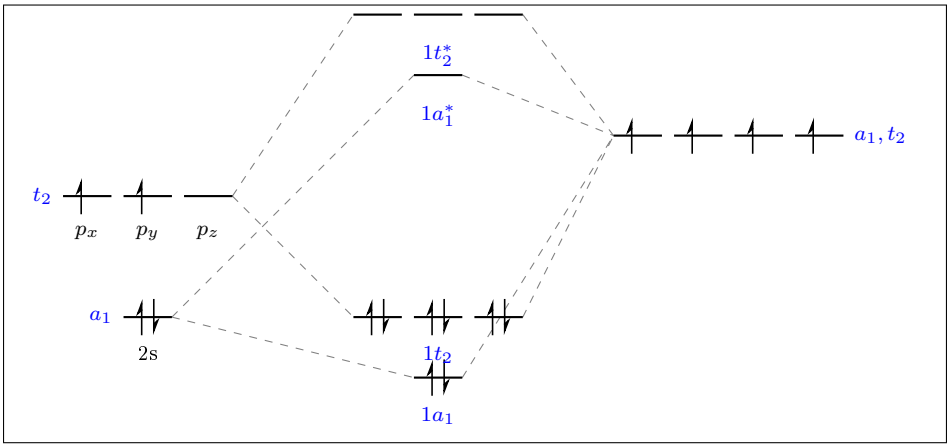
sym(metry) is quite intuitive when you understand it. This irreducible representation (Γ) is meant to help `mohelper` (yes, we help the helper) to choose which orbitals are actually overlapping. These orbitals are then connected with dotted lines. Only orbitals with the same symmetry ($\Gamma_{\psi_1} = \Gamma_{\psi_2}$) have a relevant constructive interference.
Note, that the `string` or `string array` you set for the `sym` attribute is only required for

internal calculations within the `mohelper` package. Therefore it is disallowed to use special characters. Styling or math formula is supposed to be done in labels. Instead of `$t_{2g}$` just type `t2g`. It is fast, no one will see it except you, and if you read the code, you still know it means t_{2g}.
And yes, you can specify more than one symmetry group. Put them in these brackets `{}` and separate them by comma.
This is very helpful when you initialize degenerated Ligand Group Orbitals (LGO) as Symmetry Adapted Linear Combinations (SALCs). If not specified the standard value is `a1`.



config(uration) attribute is what defines the degeneration and electron configuration of the orbital. In this `array` you specify the spin values. `up`, `down` or `pair` are the allowed values, every other string value will be treated as if there is no electron.
This excited electron configuration `{pair,up,}` will result in three degenerated orbitals. Empty spaces separated by comma are treated as empty orbitals.
If not specified the standard value is `{empty}` which will result in a non-degenerated empty orbital..

Example Have a look at this nice example ...



... and its code:

```
\begin{mohelper}
    \fragment{
        \addOrbital{
            energy=2, sym=t2, config={up,up,e},
            labels={$p_x$,$p_y$,$p_z$},
            label=$\color{blue}{t_2}$, labelposition=left
        }
```

```
        \addOrbital{
            energy=0, sym=a1, config={pair},
            labels={2s},
            label=$\color{blue}{a_1}$, labelposition=left
        }
    }
    \fragment{
        \addOrbital{
            energy=-1, sym=a1, config={pair},
            label=$\color{blue}{1a_1}$, labelposition=bottom
        }
        \addOrbital{
            energy=4, sym=a1,
            label=$\color{blue}{1a_1^*}$, labelposition=bottom
        }
        \addOrbital{
            energy=0, sym=t2, config={pair,pair,pair},
            label=$\color{blue}{1t_2}$, labelposition=bottom
        }
        \addOrbital{
            energy=5, sym=t2, config={e,e,e},
            label=$\color{blue}{1t_2^*}$, labelposition=bottom
        }
    }
    \fragment{
        \addOrbital{
            energy=3, sym={a1,t2}, config={up,up,up,up},
            label=$\color{blue}{a_1,t_2}$, labelposition=right
        }
    }
\end{mohelper}
```

You see, it's not so difficult when you get an example. By the way this is supposed to be methane (CH_4).

Manually Drawing Orbitals

Aside of `fragments` and `symmetry groups` you can manually declare orbitals. Note, they will not connect to other orbitals as they are not included into the auto-connection process.

```
\orbital{x}{y}{label}{spin}
```

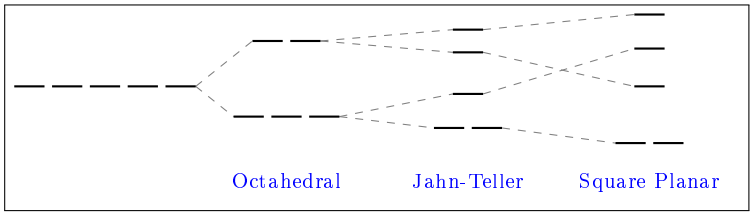
`\orbital` manually draws a single orbital at the specified coordinates. For standard orbital diagrams it should not be necessary to use this macro. Only use `\addOrbital` inside of `\fragment{}`.

Parameter	Description	Example
#1	**x** coordinate	`4.2`
#2	**y** coordinate (energy)	2.0
#3	Label	`1s`
#4	Spin	`up`, `down` or `pair` (any other value will result in an empty orbital)

5.2.2 Fragment Labels

```
\fragment{
    ... % draw orbitals
    \addLabel{text}{offset}
}
```

Labels a fragment. All fragment labels are horizontally aligned under the diagram.
`offset` (number) specifies the additional vertical offset from the diagram. As a default, just set it to `0`. `\addLabel` has to be executed inside `\fragment` or the anchor of the label will be set completely wrong.



> Tipp: To color the label you can use standard macros like `\color{color}{text}` in the `text` argument of `\addLabel`

5.3 Miscellaneous

5.3.1 Adding Energy Scale

```
\fragment{ ... }
% other fragments, other stuff
\addEnergyScale{offset}
```

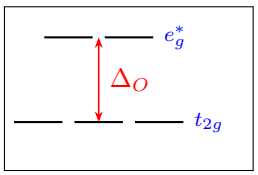
Adds an energy scale. `offset` (number) specifies the additional horizontal offset from the diagram. As a default, just set it to `0`. This macro has to be executed after the last fragment is defined. Otherwise the scale's size and position will be calculated incorrect.

5.3.2 Showing Energy Difference

```
\fragment{
    ... % draw orbitals & labels
    \showEnergyDifference{energy1, energy2}{offset}{text}
}
```

Draws a double-sided red arrow pointing out an energy difference. `offset` (number) specifies an additional horizontal offset. As a default, just set it to `0` which will center it. This macro has to be executed inside `\fragment` or the horizontal position will be completely wrong.

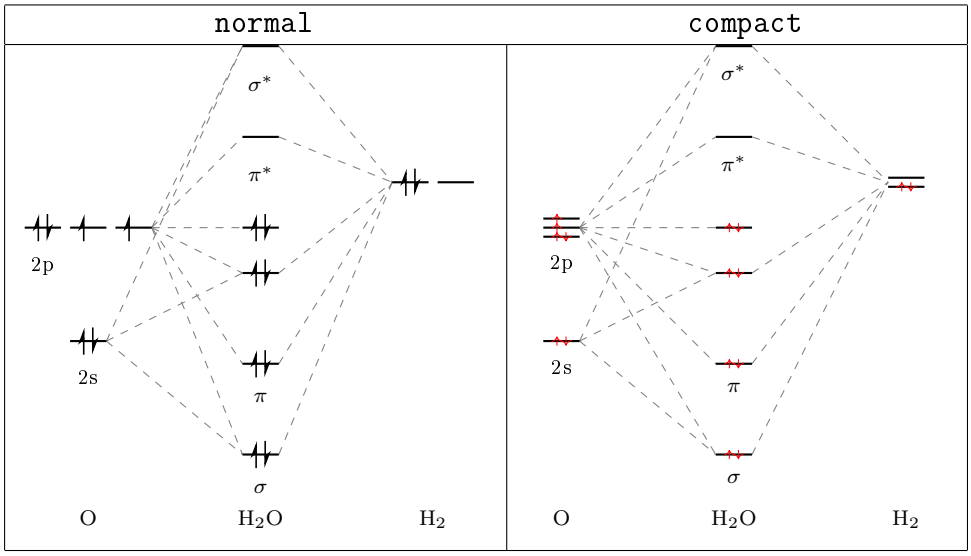
Parameter	Description	Example
#1	**Energy Array** (Format: `energy1, energy2`)	`2.0, 4.0`
#2	**Horizontal offset**, moves the arrow along x-axis	`0` (centered)
#3	**Label**	`$\Delta_O$` (Δ_O)



5.4 Customization

You can tweak the look for each diagram by redefining these values:

- `\def\columndistance{value}`: Set `value` to a number. It defines space between the AO und MO columns.
- `\def\diagrammode{compact}`: Switches to vertical stacking for degenerate orbitals (perfect for large systems).



- `\def\electroncolor{color}`: Set `color` to a valid color value. Changes the electron color to a specified one. The redefining has to stand before the relevant orbital is created using `\degenerate`.

6 Example

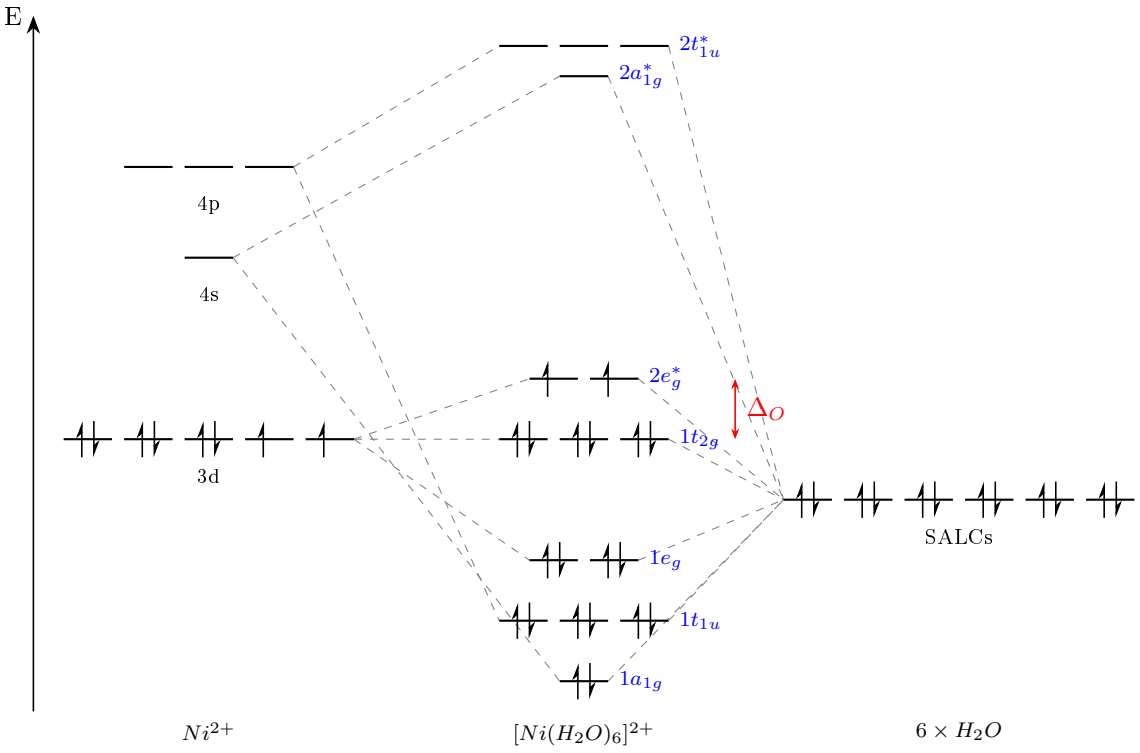


Figure 4: $[Ni(H_2O)_6]^{2+}$ complex

```
\begin{mohelper}[scale=0.8] % scale it to 0.8 of original size to fit on
    page
     \def\columndistance{5.4} % set column distance to 5.4
     \def\diagrammode{normal} % draw in normal mode
     \fragment{
          \addOrbital{energy=0,sym={t2g,eg},config={pair,pair,pair,up,up},
              label=3d}
          \addOrbital{energy=3,sym=a1g,label=4s}
          \addOrbital{energy=4.5,sym=t1u,config={,,},label=4p}
          \addLabel{$Ni^{2+}$}{0}
     }
     \fragment{
          \addOrbital{energy=-4,sym=a1g,config=pair,label=$\color{blue}{1a_{1g
              }}$,labelposition=right}
          \addOrbital{energy=-3,sym=t1u,config={pair,pair,pair},label=$\color{
              blue}{1t_{1u}}$,labelposition=right}
          \addOrbital{energy=-2,sym=eg,config={pair,pair},label=$\color{blue
              }{1e_g}$,labelposition=right}
          \addOrbital{energy=0,sym=t2g,config={pair,pair,pair},label=$\color{
              blue}{1t_{2g}}$,labelposition=right}
          \addOrbital{energy=1,sym=eg,config={up,up},label=$\color{blue}{2e_g
              ^*}$,labelposition=right}
          \addOrbital{energy=6,sym=a1g,label=$\color{blue}{2a_{1g}^*}$,
              labelposition=right}
          \addOrbital{energy=6.5,sym=t1u,config={,,},label=$\color{blue}{2t_{1
              u}^*}$,labelposition=right}
          \showEnergyDifference{0,1}{2.5}{$\Delta_O$} % show ligand field
              splitting
```

```
        \addLabel{$[Ni(H_2O)_6]^{2+}$}{0}
    }
    \fragment{
        \addOrbital{energy=-1,sym={a1g,t1u,t2g,eg},config={pair,pair,pair,
           pair,pair,pair},label=SALCs}
        \addLabel{$6 \times H_2O$}{0}
    }
    \addEnergyScale{0} % add the energy scale
\end{mohelper}
```

7 License

Distributed under the GPL-3.0 License. See `LICENSE` for more information.