TIKZORBILTAL Package

Germain SALVATO-VALLVERDU
<germain.vallverdu@univ-pau.fr>
http://gvallver.perso.univ-pau.fr/

April 8, 2015

Abstract

This package provides several LaTeX macros in order to draw easily molecular orbital diagrams and atomic orbital of type s, p and d inside tikzpicture environment. This documentation gives the syntax of the commands and complete examples of their utilization. The source code is give at the end.

Contents

1 Requirements and licence

TIKZORBILTAL underlies the LaTeX project public license (lppl) version 1.3 or later (http://www.latex-project.org/lppl.txt). It requires the following packages which all are usually included in standard texlive or MikTex distribution :

- tikz, pgfkeys (part of tikz) and tikz library shapes
- ifthen

Feel, free to send comments, contribution or suggestion by email.

2 Installation and utilization

Download the tarball which contains the files `tikzorbital.sty` and `tikzorbital.pdf` which are respectively the package file and its documentation (`http://gvallver.perso.univ-pau.fr/`). After extracting the files, copy them to directories where LaTeX will be able to find them. For example, under Linux operating systems :

```
% for the .sty file
/home/you/texmf/tex/latex/tikzorbital

% for the pdf documentation file
/home/you/texmf/doc/latex/tikzorbital
```

Under a macOS operating system the `texmf` directory is not in your home directory but in the `Library` directory.

Package TIKZORBILTAL do not have any options. After you have copied it in a directory where LaTeX will find it, simply load it as other LaTeX package :

```
\usepackage{tikzorbital}
```

All macros provided by TIKZORBILTAL must be uses inside a `tikzpicture` environment. The *[<pos>]* option of all the macros is a position which can be given in the tikz syntax and between braces.

Options of macros provided by TIKZORBILTAL must be given in a `key = value` syntax. For example :

```
\command[option 1=<value1>, option2=<value2> ...]{argument}
```

3 Molecular orbital diagrams

The package TIKZORBILTAL provides the following command in order to draw easily molecular orbital diagrams :

`\drawLevel` *[<options>]* { *<name>* }

This command draw a thick line with zero, one or two electrons which represent the occupation of the orbital and are drew as vertical arrows which indicate the spin of each electron. *<name>* is the name of the orbital and is used in order to define anchors which could be used in order to draw correlation lines or to add labels (see example 3.1). The following anchor are created :

- `left name` : at the left of the orbital level
- `right name` : at the right of the orbital level
- `middle name` : at the middle of the orbital level

The following *[<options>]* are available :

[<elec>] : Number of electrons, `<value>` must be one of {`up, down, updown, pair`} where `pair` and `updown` have got the same effect.
`default = no`

[<pos>] : left position of the level
`default = {(0,0)}`

`[<width>]` : level width

`default = 2`

`[<style>]` : a set of tikz commands which define the way levels have to be drawn

`default = {line width = 2pt, color = black!80, line cap = round}`

`[<spinstyle>]` : a set of tikz commands which define the way the arrows which represents the electrons have to be drawn

`default = {very thick, color = red!80, -stealth}`

`[<spinlength>]` : length of spin arrows

`default = 1`

Note that if the value of an option contains comma or parenthesis, you should enclose its in curly brace. The following macro allows you to change the length of levels :

`\levelWidth{<length>}` : this macro set the length of level.

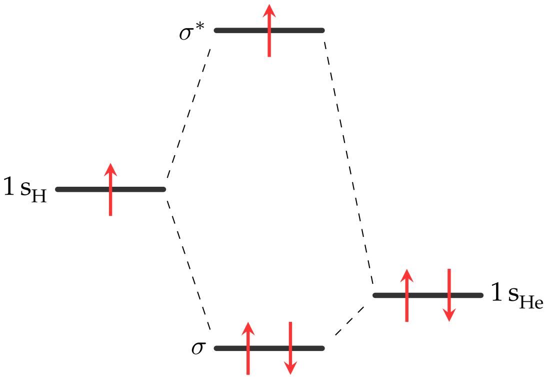
Example 3.1 shows how to use `\drawLevel` for molecule HHe.

```
\begin{tikzpicture}
   \drawLevel[elec=up, pos={(0,0)}]{1sH}
   \drawLevel[elec=updown, pos={(6,-2)}]{1sHe}

   \drawLevel[elec=pair, pos={(3,-3)}]{sigma}
   \drawLevel[elec=up, pos={(3,3)}]{sigmastar}

   \draw[dashed] (right 1sH) -- (left sigma)
         (right 1sH) -- (left sigmastar)
         (left 1sHe) -- (right sigmastar)
         (left 1sHe) -- (right sigma) ;

   \node[left] at (left 1sH) {\ce{1s_H}} ;
   \node[right] at (right 1sHe) {\ce{1s_{He}}} ;
   \node[left] at (left sigma) {$\sigma$};
   \node[left] at (left sigmastar) {$\sigma^*$};
\end{tikzpicture}
```



Example 3.1: Example of the molecular diagram of the molecule HHe$^+$ drew with the `\drawLevel` command.

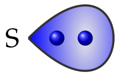
4 Molecular orbitals drawings

The package TIKZORBILTAL, provides the command `\orbital` in order to draw atomic orbital s, p or d. The general syntax is :

`\orbital[<options>]{<type>}`

where `<type>` is the type of the atomic orbital : `lobe, ±s, ±px, ±py, ±pz, dxy, dxz, dyz, dx2y2` or `dz2`. See example 4.1 to see all atomic orbital types. The `lobe` type simply draw one lobe of a p or d atomic orbital and may be used, for example, to draw hybrid orbital. With the `lobe` type you can also represent electrons as small spheres inside the orbital.

```
\begin{tikzpicture}
    \node (s) {S};
    \orbital[nelec=2, scale=1.5, pos=(s.east)]{lobe}
\end{tikzpicture}
```



General options of the `orbital` command

The following options, allow to change the position, the aspect and the size the atomic orbital. They are available for all type of atomic orbital. :

[<pos>] : position of the center of the atomic orbital
`default = {(0,0)}`

[<scale>] : scaling factor
`default = 1`

[<opacity>] : opacity of the atomic orbital. Useful if you wish to superimpose atomic orbital
`default = 1`

Color options

The color of atomic orbitals can be selected with options : *[<pcolor>]*, *[<ncolor>]* or *[<color>]*. The options *[<pcolor>]* and *[<ncolor>]* stand for the positive and the negative lobes of p or d-type atomic orbitals. The *[<color>]* option define the color of s-type or `lobe`-type orbital. For these types of atomic orbital, if no color is given the *[<pcolor>]* is used.

[<color>] : color of the atomic orbital for s-type or `lobe`-type orbital
[<pcolor>] `is used`

[<pcolor>] : color of the positive lobe (or color for s and $lobe$-type orbital if *[<color>]* is not given)
`default = blue`

[<ncolor>] : color of the negative lobe (for p and d-type orbital only)
`default = black!30`

The default values of *[<pcolor>]* and *[<ncolor>]* options can be set using the following macros :

`\setpcolor{<color>}` : this macro defines the color or positive lobes of orbitals.

`\setncolor{<color>}` : this macro defines the color or negative lobes of orbitals.

`lobe`-type specific options

The following options will have an effect only for the `lobe` type :

[<rotate>] : rotation of the atomic orbital

`default = 0`

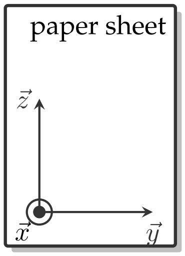
[<nelec>] : number of electron to draw inside the lobe

`default = 0`

Shapes of the atomic orbitals

Example 4.1 shows all atomic orbital types available. In order to decide the type of the atomic orbital you need, look at the axes definition below.

Cartesian axes definition :



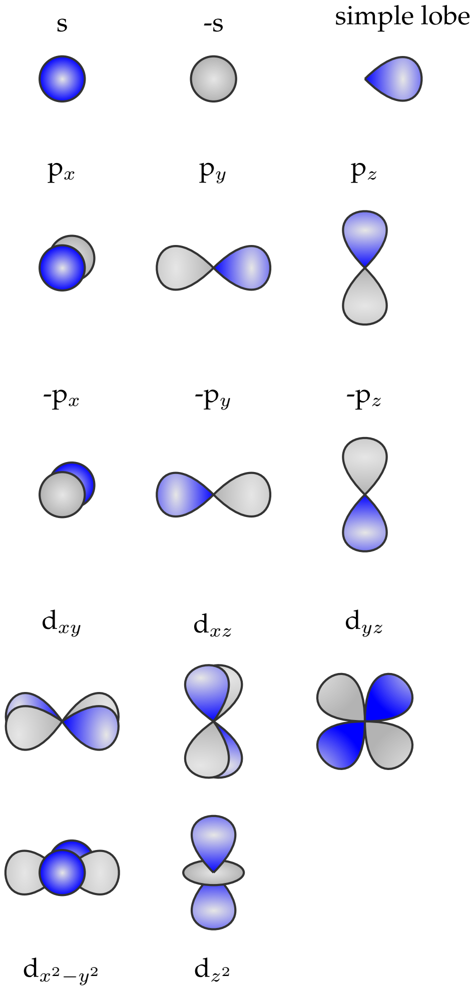
```
\begin{tikzpicture}
    \orbital[pos = {(0,5.5)}]{s}
    \node[above] at (0,6) {s};
    \orbital[pos = {(2,5.5)}]{-s}
    \node[above] at (2,6) {-s};
    \orbital[pos = {(4,5.5)}]{lobe}
    \node[above] at (4.5,6) {simple lobe};

    \orbital[pos = {(0,3)}]{px}
    \node[above] at (0,4) {p$_x$};
    \orbital[pos = {(2,3)}]{py}
    \node[above] at (2,4) {p$_y$};
    \orbital[pos = {(4,3)}]{pz}
    \node[above] at (4,4) {p$_z$};

    \orbital[pos = {(0,0)}]{-px}
    \node[above] at (0,1) {-p$_x$};
    \orbital[pos = {(2,0)}]{-py}
    \node[above] at (2,1) {-p$_y$};
    \orbital[pos = {(4,0)}]{-pz}
    \node[above] at (4,1) {-p$_z$};

    \orbital[pos = {(0,-3)}]{dxy}
    \node[above] at (0,-2) {d$_{xy}$};
    \orbital[pos = {(2,-3)}]{dxz}
    \node[above] at (2,-2) {d$_{xz}$};
    \orbital[pos = {(4,-3)}]{dyz}
    \node[above] at (4,-2) {d$_{yz}$};

    \orbital[pos = {(0,-5)}]{dx2y2}
    \node[below] at (0,-6) {d$_{x^2-y^2}$};
    \orbital[pos = {(2,-5)}]{dz2}
    \node[below] at (2,-6) {d$_{z^2}$};
\end{tikzpicture}
```



Example 4.1: All the atomic orbitals available from the command `\orbital`.

5 Atom and hybrid orbitals

The package TIKZORBILTAL provides the command `\satom` in order to quickly draw an atom with several orbital lobes around it. The general syntax of the command is :

`\satom[<options>]{<lobes>}`

The `<lobes>` argument is a comma separated list of lobe definition with the syntax

`color/rotation-angle/anchor/number of electrons/scale`

For each element of the list, the command `\satom` draw a lobe at the given anchor, with the given color, rotation, number of electrons and applies the scaling factor.

The following options are available in order to customize the drawing :

`[<pos>]` : position of the atom.
    `default = {(0,0)}`

`[<name>]` : name of the atom. Give also the name to the node where the atom is drawn.
    `default = X`

`[<color>]` : color of the atom.
    `default = green`

`[<opacity>]` : opacity of the lobe drawn around the atom.
    `default = 0.8`

`[<scale>]` : A global scaling factor of the whole atom and lobes.
    `default = 1.`

For backward compatibility the `\atom` command is still available. It works in the same way but without the possibility of applying a scaling factor individually on each lobe.

Example 5.1 show several applications of the command `\satom`.

6 More customization

Orbital borders and inner color

It is possible to change the inner color of orbital and the color of orbital borders. These two colors are defined as follow in TIKZORBILTAL package :

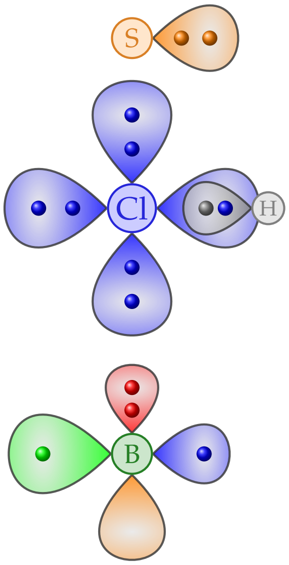
```
% inner color for orbital filling
\colorlet{innerColor}{black!10}
% color for orbital drawing
\colorlet{drawColor}{black!80}
```

Thus if you change the definition of these colors you will change the desired color on the drawing of the atomic orbitals.

```
\begin{tikzpicture}
    \satom[color=orange, name=S]{orange/0/east/2/1.}
\end{tikzpicture}
```

```
\begin{tikzpicture}
    \atom[name=Cl, color=blue, scale=1.2]{
        blue/90/north/2,
        blue/0/east/1,
        blue/270/south/2,
        blue/180/west/2}
    \atom[name=H, color=gray, pos={(1.8,0)}, scale=.8]{gray/180/west/1}
\end{tikzpicture}
```

```
\begin{tikzpicture}
    \satom[name=B, color=green!50!black]{
        red/90/north/2/.8,
        blue/0/east/1/.9,
        orange/270/south/0/1,
        green/180/west/1/1.2}
\end{tikzpicture}
```



Example 5.1: Utilization example of the \satom command.

Orbital customization

You can give a set of tikz options to the command \setOrbitalDrawing. This command acts as a tikz style which is applied every time an atomic orbital is drawn. All options give in this command will overwrite default style of atomic orbital. For example, if you want to draw atomic orbital in red with very thick line thickness :

```
\setOrbitalDrawing{{very thick, color = red}}
```

Change default value globally with pgfkeys

If you want to change the default value of the *[<width>]* option of the \drawLevel command or whatever other option for a whole tikzpicture, you can do this using the **\pgfkeys** command. You simply have to give to this command one or several options you want to set globally.

All options of a TIKZORBILTAL's command follow the tree : /tikzorbital/command/option. For example, if you want to change the *[<width>]* option of the \drawLevel command, you have to write :

```
\pgfkeys{tikzorbital/drawLevel/width = 1}
% or
\pgfkeys{tikzorbital/drawLevel/.cd, width = 1}
```

7 Inner macro \@alobe

In order to draw atomic orbital, TIKZORBILTAL use the inner macro \@alobe.

\@alobe {*<pos>*}{*<rotation>*}{*<scale>*}{*<color>*}{*<nelec>*}{*<opacity>*}

\@alobe macro draw one lobe of p or d orbital and corresponds to the lobe type of \orbital (see above). \@alobe accepts six arguments :

#1 the position

#2 angle of rotation

#3 scaling factor

#4 the color

#5 the number of electron, namely 0, 1 or 2

#6 the opacity of the lobe

no default are given. For example, the d_{yz} atomic orbital is defined as follow

```
\@alobe{\orbital@pos}{45}{\orbital@scale}{\orbital@pcolor}{0}{\orbital@opacity}
\@alobe{\orbital@pos}{135}{\orbital@scale}{\orbital@ncolor}{0}{\orbital@opacity}
\@alobe{\orbital@pos}{225}{\orbital@scale}{\orbital@pcolor}{0}{\orbital@opacity}
\@alobe{\orbital@pos}{315}{\orbital@scale}{\orbital@ncolor}{0}{\orbital@opacity}
```

8 Source code

```
% -------------------
% Package tikzorbital
% -------------------
%
% This package provides several commands in order to draw atomic orbitals and
% molecular diagrams.
%
% Germain Vallverdu <germain.vallverdu@univ-pau.fr>
% 05 decembre 2012
% http://gvallver.perso.univ-pau.fr/
%
% Licence : LaTeX Project Public Licence
% http://www.latex-project.org/lppl.txt
%
% Feel free to contact me if you have any ideas, suggestions or bugs report !
%
% Change
% ------
% 27/02/2013 : add -px, -py, -pz orbital type
% 05/03/2015 : add satom macro, with scaling options for each lobe
% 08/04/2015 : add macros to customize orbitals, add -s type orbital
%
% ------------------------------------------------------------------------------
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{tikzorbital}[2012/12/05 draw atomic orbitals and molecular diagrams with tikz]
% ------------------------------------------------------------------------------

\RequirePackage{tikz}
\usetikzlibrary{shapes}
```

```
\RequirePackage{ifthen}

\pgfdeclarelayer{background}
\pgfdeclarelayer{main}
\pgfdeclarelayer{foreground}
\pgfsetlayers{background,main,foreground}

% -------------------------------------------------------------------------------
% keys in order to submit tikz command to macro
% -------------------------------------------------------------------------------
\pgfkeys{/tikz/.cd,
    execute style/.style = {#1},
    execute macro/.style = {execute style/.expand once=#1}
}

% -------------------------------------------------------------------------------
% commande \drawLevel[key = value]{name}
% -------------------------------------------------------------------------------
% draw a level with a given name in order to draw molecular diagrams
%
%  argument
%     name       : base name of level anchor.
%
%  options
%     elec       : Number of electrons : up, down, updown or pair
%     pos        : left position of the level
%     width      : level widht
%     style      : level style (a tikzstyle)
%     spinstyle  : style of arrows which represents electrons (a tikzstyle)
%     spinlength : length of spin arrows
% -------------------------------------------------------------------------------

% macro \leveWidth allow to set up a default length of levels
\newcommand{\levelWidth}[1]{\def\@levelWidth{#1}}
\levelWidth{2}

\pgfkeys{/tikzorbital/drawLevel/.cd,
% number of electron in the level : up, down, updown or pair
    elec/.store in = \drawLevel@elec,
    elec/.default = no,
% position of the left anchor of the level
    pos/.store in = \drawLevel@pos,
    pos/.default = {(0,0)},
% width of levels
    width/.store in = \drawLevel@width,
    width/.default = \@levelWidth,
% style of levels
    style/.store in = \drawLevel@style,
    style/.default = {line width = 2pt, color = black!80, line cap = round},
% style of arrows
    spinstyle/.store in = \drawLevel@spinstyle,
    spinstyle/.default = {very thick, color = red!80, -stealth},
% length of spin arrows
    spinlength/.store in = \drawLevel@spinlength,
    spinlength/.default = 1,
% execute options
    elec, pos, width, style, spinstyle, spinlength
}

% the drawLevel command
\newcommand{\drawLevel}[2][]{%
    \begingroup
    \pgfkeys{/tikzorbital/drawLevel/.cd, #1}
    \def\drawLevel@name{#2}

    \draw[execute macro = \drawLevel@style]
        \drawLevel@pos
        node (left \drawLevel@name) {}
        -- ++ (\drawLevel@width, 0)
        node (right \drawLevel@name) {}
        node[pos=0.5] (middle \drawLevel@name) {}
        node[pos=0.3] (pos1) {}
        node[pos=0.7] (pos2) {};

    \ifthenelse{\equal{\drawLevel@elec}{updown} \or \equal{\drawLevel@elec}{pair}}{
        \draw[execute macro = \drawLevel@spinstyle]
            (pos1.center) ++ (0,-\drawLevel@spinlength/2) --
                          ++ (0,\drawLevel@spinlength);
        \draw[execute macro = \drawLevel@spinstyle]
            (pos2.center) ++ (0, \drawLevel@spinlength/2) --
                          ++ (0,-\drawLevel@spinlength);
    }{
        \ifthenelse{\equal{\drawLevel@elec}{up}}{
            \draw[execute macro = \drawLevel@spinstyle]
                (middle #2.center) ++ (0,-\drawLevel@spinlength/2) --
                                   ++ (0,\drawLevel@spinlength);
        }{
            \ifthenelse{\equal{\drawLevel@elec}{down}}{
                \draw[execute macro = \drawLevel@spinstyle]
                    (middle #2.center) ++(0,\drawLevel@spinlength/2) --
                                       ++ (0,-\drawLevel@spinlength);
            }{
            }
        }
    }
    \endgroup
}

```

```
% -----------------------------------------------------------------------------
% some customization of orbital
% -----------------------------------------------------------------------------

% inner color for orbital filling
\colorlet{innerColor}{black!10}

% color for orbital drawing
\colorlet{drawColor}{black!80}

% more style for lobe orbital drawing
\newcommand{\setOrbitalDrawing}[1]{\def\orbitalDrawing{#1}}
\setOrbitalDrawing{thick}

% color of positive coefficiant
\newcommand{\setpcolor}[1]{\colorlet{pcolor}{#1}}
\setpcolor{blue}

% color of negative coefficiant
\newcommand{\setncolor}[1]{\colorlet{ncolor}{#1}}
\setncolor{black!30}

% -----------------------------------------------------------------------------
% inner \@alobe command
% -----------------------------------------------------------------------------
% Draw one lobe of a p or d atomic orbital, at a given position with a given scale,
% color, rotation and opacity. Draw zero, one or two balls which represent electrons.
%
% arguments
%     #1 : position
%     #2 : rotation
%     #3 : scale
%     #4 : color
%     #5 : number of electron
%     #6 : opacity
% -----------------------------------------------------------------------------
\newcommand{\@alobe}[6]{
    \begin{scope}[rotate around = {#2:#1}]
        % draw orbital lobe
        \begin{pgfonlayer}{background}
            \draw[draw = drawColor, outer color = #4, inner color = innerColor,
                  opacity = #6, execute macro = \orbitalDrawing]
                #1 .. controls ++ (#3, #3) and ++ (#3, - #3) .. #1;
        \end{pgfonlayer}

        %Coordinates of the electrons
        \path #1 ++ (0.50 * #3, 0) node (e1) {};
        \path #1 ++ (0.25 * #3, 0) node (e2) {};
    \end{scope}

    % Draw the electrons
    \ifnum#5>0
        \foreach \n in {1,...,#5} {
            \shade[ball color = #4] (e\n) circle (1mm);
        }
    \fi
}

% -----------------------------------------------------------------------------
% commande \orbital[key = value]{type}
% -----------------------------------------------------------------------------
% draw an atomic orbital of a given type.
%
%  argument
%     type    : lobe, s, -s, px, py, pz, -px, -py, -pz, dxz, dyz, dxy, dz2, dx2y2
%
%  options
%     pos     : left position of the level
%     pcolor  : color of the positive lobe
%     ncolor  : color of the negative lobe
%     scale   : scaling factor
%     opacity : opacity of the orbital
%     rotate  : rotate of the AO (lobe type only)
%     nelec   : number of electron (lobe type only)
% -----------------------------------------------------------------------------

% define options
\pgfkeys{/tikzorbital/orbital/.cd,
% position of the orbital
    pos/.store in = \orbital@pos,
    pos/.default = {(0,0)},
% color of the positive lobe
    pcolor/.store in = \orbital@pcolor,
    pcolor/.default = pcolor,
% color of the negative lobe
    ncolor/.store in = \orbital@ncolor,
    ncolor/.default = ncolor,
% color for s type
    color/.store in = \orbital@color,
    color/.default = empty,
% scale factor
    scale/.store in = \orbital@scale,
    scale/.default = 1,
% opacity of the orbital
    opacity/.store in = \orbital@opacity,
    opacity/.default = 1.,
% lobe type options
% rotation of the orbital
```

```
    rotate/.store in = \orbital@rotate,
    rotate/.default = 0,
% number of electrons
    nelec/.store in = \orbital@nelec,
    nelec/.default = 0,
% execute options
    pos, pcolor, ncolor, scale, opacity, rotate, nelec, color
}

% orbital command
\newcommand{\orbital}[2][]{
    \begingroup
    \pgfkeys{/tikzorbital/orbital/.cd, #1}

    % orbital type
    \def\orbital@type{#2}

    % general style
    \tikzstyle{base} = [draw = drawColor, thick, inner color = innerColor,
                        circle, opacity = \orbital@opacity,
                        execute macro = \orbitalDrawing]

    % check if color was setted
    \ifthenelse{\equal{\orbital@color}{empty}}{
        \pgfkeys{/tikzorbital/orbital/.cd, color = \orbital@pcolor}
    }{}

    % draw the whished orbital
    \ifthenelse{\equal{\orbital@type}{lobe}}{
        \@alobe{\orbital@pos}{\orbital@rotate}{\orbital@scale}{\orbital@color}{\orbital@nelec}{\orbital
    }{
    \ifthenelse{\equal{\orbital@type}{py}}{
        \@alobe{\orbital@pos}{0}{\orbital@scale}{\orbital@pcolor}{0}{\orbital@opacity}
        \@alobe{\orbital@pos}{180}{\orbital@scale}{\orbital@ncolor}{0}{\orbital@opacity}
    }{
    \ifthenelse{\equal{\orbital@type}{-py}}{
        \@alobe{\orbital@pos}{180}{\orbital@scale}{\orbital@pcolor}{0}{\orbital@opacity}
        \@alobe{\orbital@pos}{0}{\orbital@scale}{\orbital@ncolor}{0}{\orbital@opacity}
    }{
    \ifthenelse{\equal{\orbital@type}{pz}}{
        \@alobe{\orbital@pos}{90}{\orbital@scale}{\orbital@pcolor}{0}{\orbital@opacity}
        \@alobe{\orbital@pos}{270}{\orbital@scale}{\orbital@ncolor}{0}{\orbital@opacity}
    }{
    \ifthenelse{\equal{\orbital@type}{-pz}}{
        \@alobe{\orbital@pos}{270}{\orbital@scale}{\orbital@pcolor}{0}{\orbital@opacity}
        \@alobe{\orbital@pos}{90}{\orbital@scale}{\orbital@ncolor}{0}{\orbital@opacity}
    }{
    \ifthenelse{\equal{\orbital@type}{px}}{
        \node[base, outer color = \orbital@ncolor, scale = \orbital@scale * 1.8,
              xshift = 2pt, yshift = 2pt] at \orbital@pos {};
        \node[base, outer color = \orbital@pcolor, scale = \orbital@scale * 1.8]
            at \orbital@pos {};
    }{
    \ifthenelse{\equal{\orbital@type}{-px}}{
        \node[base, outer color = \orbital@pcolor, scale = \orbital@scale * 1.8,
              xshift = 2pt, yshift = 2pt] at \orbital@pos {};
        \node[base, outer color = \orbital@ncolor, scale = \orbital@scale * 1.8]
            at \orbital@pos {};
    }{
    \ifthenelse{\equal{\orbital@type}{dyz}}{
        \@alobe{\orbital@pos}{45}{\orbital@scale}{\orbital@pcolor}{0}{\orbital@opacity}
        \@alobe{\orbital@pos}{135}{\orbital@scale}{\orbital@ncolor}{0}{\orbital@opacity}
        \@alobe{\orbital@pos}{225}{\orbital@scale}{\orbital@pcolor}{0}{\orbital@opacity}
        \@alobe{\orbital@pos}{315}{\orbital@scale}{\orbital@ncolor}{0}{\orbital@opacity}
    }{
    \ifthenelse{\equal{\orbital@type}{dxz}}{
        \@alobe{\orbital@pos}{80}{\orbital@scale}{\orbital@ncolor}{0}{\orbital@opacity}
        \@alobe{\orbital@pos}{280}{\orbital@scale}{\orbital@pcolor}{0}{\orbital@opacity}
        \@alobe{\orbital@pos}{100}{\orbital@scale}{\orbital@pcolor}{0}{\orbital@opacity}
        \@alobe{\orbital@pos}{260}{\orbital@scale}{\orbital@ncolor}{0}{\orbital@opacity}
    }{
    \ifthenelse{\equal{\orbital@type}{dxy}}{
        \@alobe{\orbital@pos}{10}{\orbital@scale}{\orbital@ncolor}{0}{\orbital@opacity}
        \@alobe{\orbital@pos}{170}{\orbital@scale}{\orbital@pcolor}{0}{\orbital@opacity}
        \@alobe{\orbital@pos}{350}{\orbital@scale}{\orbital@pcolor}{0}{\orbital@opacity}
        \@alobe{\orbital@pos}{190}{\orbital@scale}{\orbital@ncolor}{0}{\orbital@opacity}
    }{
    \ifthenelse{\equal{\orbital@type}{dx2y2}}{
        \begin{pgfonlayer}{background}
        \node[base, outer color = \orbital@pcolor, scale = \orbital@scale * 1.8,
              xshift = 2pt, yshift = 2pt] at \orbital@pos {};
        \end{pgfonlayer}
        \@alobe{\orbital@pos}{0}{\orbital@scale}{\orbital@ncolor}{0}{\orbital@opacity}
        \@alobe{\orbital@pos}{180}{\orbital@scale}{\orbital@ncolor}{0}{\orbital@opacity}
        \node[base, outer color = \orbital@pcolor, scale = \orbital@scale * 1.8]
            at \orbital@pos {};
    }{
    \ifthenelse{\equal{\orbital@type}{dz2}}{
        \@alobe{\orbital@pos}{270}{\orbital@scale}{\orbital@pcolor}{0}{\orbital@opacity}
        \begin{pgfonlayer}{background}
        \node[ellipse, minimum width = \orbital@scale * .8cm,
              minimum height = \orbital@scale * .3cm, draw = drawColor,
              inner color = innerColor, outer color = \orbital@ncolor,
              execute macro = \orbitalDrawing]
            at \orbital@pos {};
        \end{pgfonlayer}
        \@alobe{\orbital@pos}{90}{\orbital@scale}{\orbital@pcolor}{0}{\orbital@opacity}
    }{
```

```
    \ifthenelse{\equal{\orbital@type}{s}}{
        \node[base, outer color = \orbital@pcolor, scale = \orbital@scale * 1.8]
            at \orbital@pos {};
    }{
    \ifthenelse{\equal{\orbital@type}{-s}}{
        \node[base, outer color = \orbital@ncolor, scale = \orbital@scale * 1.8]
            at \orbital@pos {};
    }{
        \node[red] at \orbital@pos {orbital type unknown};
    }}}}}}}}}}}}}}
    \endgroup
}

%
% other possibility for dxy and dxz atomic orbital
% ------------------------------------------------
%
% dxz
%    \begin{scope}[xshift = 2.2pt, yshift = 2pt]
%        \@alobe{\orbital@pos}{90}{\orbital@scale}{\orbital@ncolor}{0}{\orbital@opacity}
%        \@alobe{\orbital@pos}{270}{\orbital@scale}{\orbital@pcolor}{0}{\orbital@opacity}
%    \end{scope}
%    \@alobe{\orbital@pos}{90}{\orbital@scale}{\orbital@pcolor}{0}{\orbital@opacity}
%    \@alobe{\orbital@pos}{270}{\orbital@scale}{\orbital@ncolor}{0}{\orbital@opacity}
%
% dxy
%    \begin{scope}[xshift = 2.2pt, yshift = 2pt]
%        \@alobe{\orbital@pos}{10}{\orbital@scale}{\orbital@ncolor}{0}{\orbital@opacity}
%        \@alobe{\orbital@pos}{170}{\orbital@scale}{\orbital@pcolor}{0}{\orbital@opacity}
%    \end{scope}
%    \@alobe{\orbital@pos}{350}{\orbital@scale}{\orbital@pcolor}{0}{\orbital@opacity}
%    \@alobe{\orbital@pos}{190}{\orbital@scale}{\orbital@ncolor}{0}{\orbital@opacity}
%
% -------------------------------------------------

% ------------------------------------------------------------------------------
% commande \atom[options]{lobes}
% ------------------------------------------------------------------------------
% quickly draw an atom with several orbital lobes around it.
% DEPRECATED, use satom insteed.
%
%  argument
%     lobes   : A comma separated list lobe definition with
%                  color/rotation-angle/anchor/number of electrons
%
%  options
%     pos     : position of the atom
%     name    : name of the atom, also used to label the node
%     color   : color of the atom
%     opacity : opacity of the orbital
%     scale   : scaling factor
% ------------------------------------------------------------------------------

% define options
\pgfkeys{/tikzorbital/atom/.cd,
% position of the atom
    pos/.store in = \atom@pos,
    pos/.default = {(0,0)},
% atom name
    name/.store in = \atom@name,
    name/.default = X,
% color of the atom
    color/.store in = \atom@color,
    color/.default = green,
% opacity of the orbitals
    opacity/.store in = \atom@opacity,
    opacity/.default = .8,
% scaling factor
    scale/.store in = \atom@scale,
    scale/.default = 1.,
% execute options
    pos, name, color, opacity, scale
}

% atom definition
\newcommand{\atom}[2][]{
    \begingroup
    \pgfkeys{/tikzorbital/atom/.cd, #1}
    \colorlet{atomColor}{\atom@color}
    \node[shape = circle, thick, inner sep = 0pt, minimum size = 1.5em,
          draw = atomColor!40, color = atomColor!70!gray, fill = atomColor!20,
          scale = \atom@scale]
          at \atom@pos (\atom@name) {\atom@name};
    \def\s{1.}
    \foreach \acolor/\rot/\anchor/\Ne in {#2} {
        \@alobe{(\atom@name.\anchor)}{\rot}{1.5*\atom@scale}{\acolor}{\Ne}{\atom@opacity}
    }
    \endgroup
}

% ------------------------------------------------------------------------------
% commande \satom[options]{lobes}
% ------------------------------------------------------------------------------
% quickly draw an atom with several orbital lobes around it
%
%  argument
%     lobes   : A comma separated list lobe definition with
%                  color/rotation-angle/anchor/number of electrons/scale
```

```
%
%  options
%     pos     : position of the atom
%     name    : name of the atom, also used to label the node
%     color   : color of the atom
%     opacity : opacity of the orbital
%     scale   : global scaling factor
% -----------------------------------------------------------------------------

% define options
\pgfkeys{/tikzorbital/satom/.cd,
% position of the atom
    pos/.store in = \satom@pos,
    pos/.default = {(0,0)},
% atom name
    name/.store in = \satom@name,
    name/.default = X,
% color of the atom
    color/.store in = \satom@color,
    color/.default = green,
% opacity of the orbitals
    opacity/.store in = \satom@opacity,
    opacity/.default = .8,
% scaling factor
    scale/.store in = \satom@scale,
    scale/.default = 1.,
% execute options
    pos, name, color, opacity, scale
}

% atom definition
\newcommand{\satom}[2][]{
    \begingroup
    \pgfkeys{/tikzorbital/satom/.cd, #1}
    \colorlet{atomColor}{\satom@color}
    \node[shape = circle, thick, inner sep = 0pt, minimum size = 1.5em,
          draw = atomColor!40, color = atomColor!70!gray, fill = atomColor!20,
          scale = \satom@scale]
          at \satom@pos (\satom@name) {\satom@name};
    \foreach \acolor/\rot/\anchor/\Ne/\s in {#2} {
        \@alobe{(\satom@name.\anchor)}{\rot}{1.5*\s*\atom@scale}{\acolor}{\Ne}{\satom@opacity}
    }
    \endgroup
}

%% end of file %%
```