chemmacros v2.0a

2011/11/03

Clemens Niederberger

http://www.mychemistry.eu/
contact@mychemistry.eu

chemmacros is a collection of macros and commands which are intended to make typesetting chemistry documents with LaTeX 2ε faster and more convenient. Coverage includes some nomenclature commands, oxidation numbers, thermodynamic data, newman projections, etc.

Contents

1 What's New?

With the update to v2.0 lot's has changed behind the scenes. The settings now are done completely with a key/value system. This way there are now much more possibilities to customize commands. This also means that the syntax of a number of commands has changed. In order to provide compatibility with documents set with v1.*, there is now the package option `version=1`, which restores the old definitions. If a command has been changed, you can see this through a note in the margin. Some commands now are deprecated and are not longer provided unless the option `version=1` is used.

The command \command has a new syntax with v2.0. The package option `version=1` restores the former syntax.

\command is not provided any more. Use package option `version=1` to reactivate it.

2 Licence, Requirements

chemmacros v2.0 underlies the The LaTeX project public license version 1.3 or later. (http://www.latex-project.org/lppl.txt)

chemmacros uses the packages expl3, xparse[1], l3keys2e[2] and xfrac[3], which are part of the bundles l3kernel[4] and l3packages[5]. expl3[6] is part of l3kernel, and xparse, l3keys2e and xfrac are part of l3packages.

chemmacros also uses the packages siunitx[7], mhchem[8], mathtools[9] and environ[10] as well as TikZ (pgf[11]) and the TikZ libraries `calc` and `arrows`.

Package option `bpchem` (section 3) needs the package bpchem[12] and package option `xspace` needs the package xspace[13].

[1] CTAN: xparse
[2] CTAN: l3keys2e
[3] CTAN: xfrac
[4] CTAN: l3kernel
[5] CTAN: l3packages
[6] CTAN: expl3
[7] CTAN: siunitx
[8] CTAN: mhchem
[9] CTAN: mathtools
[10] CTAN: environ
[11] CTAN: pgf
[12] CTAN: bpchem
[13] CTAN: xspace

3 Package Options

chemmacros has several package options. They all are used as key/-value pairs like `\usepackage[option1 = <value1>, option2 = <value2>]{chemmacros}`. Some also can be used without value (`\usepackage[option1, option2]{chemmacros}`), which means that the underlined value is used.

This document is set with default behaviour.

`bpchem = true/false>` This option loads the package bpchem and adjusts the layout of the \NMR command to the bpchem commands \HNMR and \CNMR. (default: `false`)

`bpchem = false` ^{1}H-NMR: δ ;
`bpchem = true` ^{1}H-NMR: δ ;

`circled = formal/all/none` chemmacros uses two different kinds of charges[14], which indicate the usage of real ($+/-$) and formal ($\oplus/\ominus$) charges. The option `formal` distinguishes between them, option `none` displays them all without circle, option `all` circles all (default: `formal`).

`circled = none` $- + - +$
`circled = formal, circletype = chem` $- + \ominus \oplus$
`circled = all, circletype = chem` $\ominus \oplus \ominus \oplus$
`circled = formal, circletype = math` $- + \ominus \oplus$
`circled = all, circletype = math` $\ominus \oplus \ominus \oplus$

[14] Thanks to Christoph Schäfer, who pointed out to me, that v1.1 handled the charges too undifferentiated!

`circletype = chem/math` This option switches between two kinds of circled charge symbols: \fplus $\oplus$ and $\oplus$ $\oplus$. (Default: `chem`)

`EZ = chemmacros/cool` The command \E is defined by the package cool[15] as well as by chemmacros. With this option you can choose, which definition is used, see page 6. (default: `chemmacros`)

[15] CTAN: cool

`german = true/false` This option changes the commands \pKa, \sld and \lqd (default: `false`)

`german = false` pK_A, (s), (l)
`german = true` pK_A, (s), (l)

`version = 1/2` This option restores the old definitions of some commands, so documents set with v1.* will still compile correctly. You'll find notes in the margin for every changed command. (default: 2)

`xspace = true/false` With this option most commands are defined with a \xspace. (default: `true`)

4 Setup

Various of chemmacros's commands have key/value pairs with which they are customized. Most times they can be used as (optional) argument of the copmmands themselves. They also can most times be used with the \chemsetup command.

```
\chemsetup[<module>]{<key> = <value>} or
\chemsetup{<module>/<key> = <value>}
```

The keys each belong to a module, which defines for which commands they are intended for. If a key is presented, you'll see a box in the margin that gives you information to that key. You have two ways to use keys with the \chemsetup, as you can see in the box above.

<key> = <value>
default: <default>
module: <module>

The package options can also be seen as keys belonging to the module `option`. This means they can also be used with the \chemsetup command (except for the option `version=1/2`).

```
\chemsetup[option]{circled=none}\mch\ \pch\ \fmch\ \
    fpch\ \el\ \prt \\
\chemsetup[option]{circled=formal}\mch\ \pch\ \fmch\ \
    fpch\ \el\ \prt \\
\chemsetup[option]{circletype=math}\mch\ \pch\ \fmch\ \
    fpch\ \el\ \prt \\
\chemsetup{option/circletype=chem,option/circled=all}\
    mch\ \pch\ \fmch\ \fpch\ \el\ \prt \\
\chemsetup{option/circletype=math}\mch\ \pch\ \fmch\ \
    fpch\ \el\ \prt
```

$^{-}$ $^{+}$ $^{-}$ $^{+}$ e^{-} p^{+}
$^{-}$ $^{+}$ $^{\ominus}$ $^{\oplus}$ e^{-} p^{+}
$^{-}$ $^{+}$ $^{\circ\ominus}$ $^{\circ\oplus}$ e^{-} p^{+}
$^{\ominus}$ $^{\oplus}$ $^{\ominus}$ $^{\oplus}$ $e^{\ominus}$ $p^{\oplus}$
$^{\circ\ominus}$ $^{\circ\oplus}$ $^{\circ\ominus}$ $^{\circ\oplus}$ $e^{\ominus}$ $p^{\oplus}$

Keys *not* belonging to a module *cannot* be used with \chemsetup!

5 Particles, Ions and Symbols

Some simple macros for displaying often needed particles and a symbol. Please note, that they're displayed differently depending on the package options used, see section 3.

These commands can be used in text as well as in math mode.

- \Hpl H^+ (proton)
- \Hyd OH^- (hydroxide)
- \HtO H_3O^+ (oxonium ion) (**H** three **O**)
- \water H_2O
- \el e^- (electron)
- \prt p^+ (proton)
- \ntr n^0 (neutron)
- \Nu Nu^- (nucleophile)
- \El E^+ (electrophile)
- \ba ba^- (base)
- \fplus $\oplus$
- \fminus $\ominus$
- \transitionstatesymbol $\pm$ (uses Ti*k*Z)
- \standardstate $\ominus$. This symbol is only provided by chemmacros, if the package chemstyle[16] is not loaded; the idea is borrowed from there[17].

[16] CTAN: chemstyle

[17] thanks to the package author Joseph Wright.

There is another command which allows to typeset radicals with charges and subscripts.

```
\R[<sign>]{<subscript>}
```

R^+_{tert} R^-_{sek} R_{prim}

```
\R[+]{tert} \R[-]{sek} \R{prim}
```

The two particles \Nu and \ba can be modified. To do that you use the key `elpair`. It only has any effect, if the package chemfig[18] is loaded, since it uses it's command \Lewis.

```
elpair = false/dots/dash
default: false
module: particle
```

[18] CTAN: chemfig

ba:$^-$ Nu|$^-$

```
\documentclass{article}
\usepackage{chemmacros,
    chemfig}
\begin{document}
 \ba[elpair] \Nu[elpair=
    dash]
\end{document}
```

ba:$^-$ Nu:$^-$

```
\documentclass{article}
\usepackage{chemmacros,
    chemfig}
\begin{document}
 \chemsetup[particle]{
    elpair}
 \ba \Nu
\end{document}
```

6 Stereo Descriptors, Nomenclature, Latin Phrases

6.1 Stereo Descriptors and Nomenclature

The following macros are intended to make the writing of IUPAC names more convenient:

- Cahn-Ingold-Prelog:

```
\Rcip (R)
\Scip (S)
\cip{<conf>} e. g.: \cip{R,S} (R,S)
```

- Fischer:

```
\Dfi D
\Lfi L
```

- cis/trans and zusammen/entgegen:

```
\Z (Z)
\E (E) (\E is also defined by the package cool. By using the package option EZ = cool instead of \E and \Z chemmacros defines \Ent and \Zus.)
\cis cis
\trans trans
```

Please notice, that the commands \cis and \trans are defined by the bpchem package as well. If you load that package, they are redefined by chemmacros. With bpchem they *always* get a \xspace, with chemmacros only , when option xspace is used.

- ortho/meta/para:

```
\ortho o
\meta m
\para p
```

absolute configuration (uses Ti*k*Z):

```
\Rconf[<letter>] \Rconf: (R)  \Rconf[]: ( )
\Sconf[<letter>] \Sconf: (S)  \Sconf[]: ( )
```

Examples:

```
\Dfi-Weins\"aure = \cip{2S,3S}-Weins\"aure \\
\Dfi-($-$)-Threose = \cip{2S,3R}-($-$)-2,3,4-
    Trihydroxybutanal \\
\cis-2-Buten = \Z-2-Butene, \cip{2E,4Z}-Hexadiene \\
\meta-Xylol = 1,3-Dimethylbenzene \\
% with bpchem's command \IUPAC:
\IUPAC{\Dfi\-Wein\|s\"aure} = \IUPAC{\cip{2S,3S}\-Wein
    \|s\"aure}, \IUPAC{\Dfi\-($-$)\-Threose} = \IUPAC{\
    cip{2S,3R}\-($-$)\-2,3,4\-Tri\|hydroxy\|butanal}
```

D-Weinsäure = *(2S,3S)*-Weinsäure
D-(−)-Threose = *(2S,3R)*-(−)-2,3,4-Trihydroxybutanal
cis-2-Buten = *(Z)*-2-Butene, *(2E,4Z)*-Hexadiene
m-Xylol = 1,3-Dimethylbenzene
D-Weinsäure = *(2S,3S)*-Weinsäure, D-(−)-Threose = *(2S,3R)*-(−)-2,3,4-Trihydroxybutanal

Of course the appearance depends on the font you chose:
*(2S,3R) (E)(Z)*DL
*(2S,3R) (E)(Z)*DL
*(2S,3R) (E)(Z)*DL

6.2 Latin Phrases

At last there are two commands for common latin phrases.

```
\insitu in situ
\abinitio ab initio
```

If the package chemstyle has been loaded, too[19], they are defined using chemstyle's `\latin` command. This means that then the appearance depends on chemstyle's option `abbremph`:

in situ, *ab initio*
in situ, ab initio

```
\insitu, \abinitio\\
\cstsetup{abbremph=false}
\insitu, \abinitio
```

If chemstyle hasn't been loaded, they're always in *italics*.

[19] chemstyle defines other similar commands like *et al.*, *in vacuo*.

7 Units with siunitx

In chemistry some non-SI units are very common. siunitx provides the command `\DeclareSIUnit{<command>}{<unit>}` to add arbitrary units. chemmacros uses that command to provide some units. Like all siunitx units they're only valid inside `\SI{<num>}{<unit>}` and `\si{<unit>}`.

- `\atmosphere` atm
- `\atm` atm
- `\calory` cal
- `\cal` cal
- `\cmc` cm^3

The units `\cmc`, `\molar`, and `\Molar` are defined by the package chemstyle as well. chemmacros only defines them, if chemstyle is not loaded.

- \molar mol dm^{-3}
- \moLar mol L^{-1}
- \Molar M
- \MolMass g mol^{-1}
- \normal N
- \torr torr

By the way: \mmHg mmHg already is defined by siunitx and chemstyle.

8 Acid/Base

Easy representation of pH, pK_A ...

These commands can be used both in text and in math mode as well as inside the \ce command of the mhchem package. The command \pKa depends on the package option german, see section 3.

\pH pH
\pOH pOH
\pKa[<num>] \pKa pK_A, \pKa[1] pK_{A1}
\pKb[<num>] \pKb pK_B, \pKb[1] pK_{B1}
\p{<anything>} e. g. \p{K_w} pK_w

pK_A pK_{A1} pK_B pK_{B1}
pK_S pK_{S1} pK_B pK_{B1}

```
\pKa \pKa[1] \pKb \pKb[1]\\
\chemsetup[option]{german=true}
\pKa \pKa[1] \pKb \pKb[1]
```

9 Oxidation Numbers, real and formal Charges

chemmacros distinguishes between real ($+/-$) and formal ($\oplus/\ominus$) charge symbols, also see section 3. All commands using formal charge symbols start with a `f`.

9.1 Ion Charges

Simple displaying of (real) charges:

\pch[<number>] positive charge (**p**lus + **ch**arge)
\mch[<number>] negative charge (**m**inus + **ch**arge)

$^{+}$, Na^{+}, Ca^{2+}
$^{-}$, F^{-}, S^{2-}

```
\pch, Na\pch, Ca\pch[2]\\
\mch, F\mch, S\mch[2]
```

The same for formal charges:

\fpch[<number>] positive charge
\fmch[<number>] negative charge

$\oplus$ $\ominus$ $3\oplus$ $3\ominus$

```
\fpch\ \fmch\ \fpch[3] \
fmch[3]
```

There is a key which influences the behaviour of the charges.

- `append = true/false` if set `true`, the charge is appended together with an empty group.

`append = true/false`
default: `false`
module: `charges`

This is, how the key influences the behaviour:

```
\chemsetup[charges]{
    append=false}
\ce{H\pch\aq} \ce{H\aq\
    pch}

\chemsetup[charges]{
    append=true}
\ce{H\pch\aq} \ce{H\aq\
    pch}
```

In most cases this behaviour will be unwanted. However, in some cases it might be useful, for example together with the `\ox` (see next section):

```
\chemsetup[charges]{
    append=false}
\ce{\ox{1,H}\pch\aq}

\chemsetup[charges]{
    append=true}
\ce{\ox{1,H}\pch\aq}
```

9.2 Oxidation Numbers

Typesetting oxidation numbers:

`\ox[<keyval>]{<number>,<atom>}` places `<number>` above `<atom>`
`<number>` has to be a (rational) number!

The command `\ox` has a new syntax with v2.0. The package option `version=1` restores the former syntax.

```
\ox{+1,Na}, \ox{2,Ca}, \
    ox{-2,S}, \ox{-1,F}
```

There are a number of keys, that can be used to modify the `\ox` command.

- `parse = true/false` when `false` an arbitrary entry can be used for `<number>`.

`parse = true/false`
default: `true`
module: `ox`

- `roman = true/false` switches from roman to arabic numbers.

`roman = true/false`
default: `true`
module: `ox`

- `pos = top/super/side`; `top` places `<number>` above `<atom>`, `super` to the upper right and `side` to the right and inside brackets.
- `explicit-sign = true/false` shows the + for positiv numbers and the ± for 0.

- `decimal-marker = comma/point` choice for the decimal marker for formal oxidation numbers like $\overset{1.2}{\mathrm{X}}$.

$\overset{2}{\mathrm{Ca}}\ \overset{\mathrm{II}}{\mathrm{Ca}}$
$\mathrm{Fe}^{\mathrm{III}}$-Oxide
Fe(III)-Oxide
$\overset{?}{\mathrm{Mn}}$

```
\ox[roman=false]{2,Ca} \
    ox{2,Ca} \\
\ox[pos=super]{3,Fe}-
    Oxide \\
\ox[pos=side]{3,Fe}-Oxide
     \\
\ox[parse=false]{?,Mn}
```

The `pos=super` variant also can be set with the shortcut `\ox*`:

$\overset{\mathrm{III}}{\mathrm{Fe}}\ \mathrm{Fe}^{\mathrm{III}}$

```
\ox{3,Fe} \ox*{3,Fe}
```

Using the `explicit-sign` key will always show the sign of the oxidation number:

$\overset{+\mathrm{I}}{\mathrm{Na}},\ \overset{+\mathrm{II}}{\mathrm{Ca}},\ \overset{-\mathrm{II}}{\mathrm{S}},\ \overset{\pm 0}{\mathrm{F}_2}$

```
\chemsetup[ox]{explicit-
    sign = true}
\ox{+1,Na}, \ox{2,Ca}, \
    ox{-2,S}, \ox{0,F2}
```

As you could see in the last example, `<atom>` is placed within mhchem's `\ce` command. However, using this fact does not necessarily give good results.

```
Compare \ox{-1,O2^{2-}} to \ce{\ox{-1,O}{}_2 \mch[2]}
    or \ce{\ox{-1,O}{}_2^{2-}}.
```

Compare $\overset{-\mathrm{I}}{\mathrm{O}_2^{2-}}$ to $\overset{-\mathrm{I}}{\mathrm{O}_2^{2-}}$ or $\overset{-\mathrm{I}}{\mathrm{O}_2^{2-}}$.

Sometimes one might want to use formal oxidation numbers like 0.5 or $\frac{1}{3}$. This is possible:

$\overset{0.5}{\mathrm{Br}_2^{+}}\ \overset{1/3}{\mathrm{I}_3^{+}}$

```
\ox{.5,Br2\pch} \ox{1/3,I
    3+}
```

9.3 Partial Charges and similar Stuff

The next ones probably are seldomly needed but nevertheless useful:

`\delp` $\delta+$ (**del**ta + **p**lus)
`\delm` $\delta-$ (**del**ta + **m**inus)
`\fdelp` $\delta\oplus$
`\fdelm` $\delta\ominus$

These macros for example can be used with the `\ox` command or with the chemfig package:

`pos` = top/super/side
default: `top`
module: `ox`

`explicit-sign` = true/false
default: `false`
module: `ox`

`decimal-marker` = comma/point
default: `point`
module: `ox`

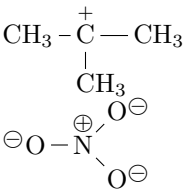
```
\chemsetup{
  option/circled = all,
  ox/parse      = false
}
\ox{\delp,H}\ox{\delm,Cl} \hspace*{1cm}
\chemfig{\chemabove[3pt]{\lewis{246,Br}}{\delm}-\
    chemabove[3pt]{H}{\delp}}
```

The following macros are useful together with chemfig, too.

```
\scrp +(scriptstyle + plus)
\scrm –(scriptstyle + minus)
\fscrp ⊕
\fscrm ⊖
\fsscrp ⊕(using \scriptscriptstyle)
\fsscrm ⊖
```

```
\setatomsep{1.8em}\chemfig{CH_3-\chemabove{C}{\scrp
    }(-[6]C|H_3)-\vphantom{H_3}CH_3}

\chemfig{\fmch{}|O-\chemabove{N}{\fscrp}(-[1]O|\fmch)
    -[7]O|\fmch}
```



10 Reaction Mechanisms

With the command

```
\mech[<type>]
```

This command can also be used in math mode and inside the \ce command of the mhchem package.

one can specify the most common reaction mechanisms. <type> can have one of the following values:

- \mech (empty, no opt. argument) nucleophilic substitution S_N
- \mech[1] unimolecular nucleophilic substitution S_N1
- \mech[2] bimolecular nucleophilic substitution S_N2
- \mech[se] electrophilic substitution S_E
- \mech[1e] unimolecular electrophilic substitution S_E1
- \mech[2e] bimolecular electrophilic substitution S_E2

- `\mech[ar]` electrophilic aromatic substitution Ar-S_E
- `\mech[e]` elimination E
- `\mech[e1]` unimolecular elimination E1
- `\mech[e2]` bimolecular elimination E2
- `\mech[cb]` unimolecular elimination "conjugated base", i. e. via carbanion $E1_{cb}$

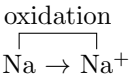
11 Redox Reactions

chemmacros provides two commands[20], to visualize the transfer of electrons in redox reactions. Both commands are using Ti*k*Z.

[20] Thanks to Peter Cao who suggested this feature.

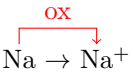
```
\OX{<name>,<atom>}
\redox(<name1>,<name2>)[<tikz>][<num>]{<text>}
```

`\OX` places `<atom>` into a node, which is named with `<name>`. If you have set two `\OX`, they can be connected with a line using `\redox`. To do so the names of the two nodes that are to be connected are written in the round braces. Since `\redox` draws a tikzpicture with options `remember picture,overlay`, the document needs to be *compiled at least two times*.



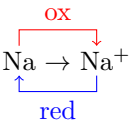

```
\OX{a,Na} $\rightarrow$ \
    OX{b,Na}\pch\redox(a,
    b){oxidation}
```

This line can be customized using Ti*k*Z keys in `[<tikz>]`:



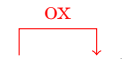

```
\OX{a,Na} $\rightarrow$ \
    OX{b,Na}\pch\redox(a,
    b)[->,red]{ox}
```

With the argument `[<num>]` the length of the vertical parts of the line can be adjusted. The default length is `.6em`. This length is multiplied with `<num>`. If you use a negative value the line is placed *below* the text.




```
\OX{a,Na} $\rightarrow$ \
    OX{b,Na}\pch
\redox(a,b)[->,red]{ox}
\redox(a,b)[<-,blue][-1]{
    red}
```

The default length of the vertical lines can be customized with the key `dist`:

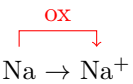



```
\chemsetup{redox/dist=1em
    }
\OX{a,Na} $\rightarrow$ \
    OX{b,Na}\pch\redox(a,
    b)[->,red]{ox}
```

```
dist = <dim>
default: .6em
module: redox
```

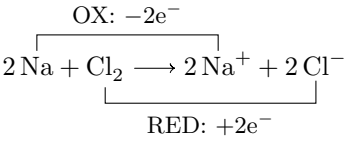
Additionally the key `sep` can be used to change the distance between the atom and the beginning of the line.

`sep` = <dim>
default: `.2em`
module: `redox`

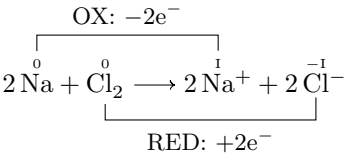



```
\chemsetup{redox/sep=.5em}
\OX{a,Na} $\rightarrow$ \OX{b,Na}\pch\redox(a,b)[->,red]{ox}
```

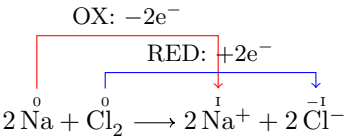
Examples:



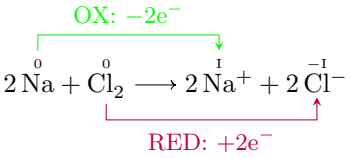

```
\ce{ 2 \OX{o1,Na} + \OX{r1,Cl2} -> 2 \OX{o2,Na}\pch + 2 \OX{r2,Cl}\mch }
\redox(o1,o2){\small OX: $- 2\el$}
\redox(r1,r2)[][-1]{\small RED: $+ 2\el$}
```




```
\chemsetup[charges]{append}
\ce{ 2 \OX{o1,\ox{0,Na}} + \OX{r1,\ox{0,Cl2}} -> 2 \OX{o2,\ox{+1,Na}}\pch + 2 \OX{r2,\ox{-1,Cl}}\mch }
\redox(o1,o2){\small OX: $- 2\el$}
\redox(r1,r2)[][-1]{\small RED: $+ 2\el$}
```




```
\chemsetup[charges]{append}
\ce{ 2 \OX{o1,\ox{0,Na}} + \OX{r1,\ox{0,Cl2}} -> 2 \OX{o2,\ox{+1,Na}}\pch + 2 \OX{r2,\ox{-1,Cl}}\mch }
\redox(o1,o2)[draw=red,->][3.33]{\small OX: $- 2\el$}
\redox(r1,r2)[draw=blue,->]{\small RED: $+ 2\el$}
```




```
\chemsetup[charges]{
    append}
\ce{ 2 \OX{o1,\ox{0,Na}}
    + \OX{r1,\ox{0,Cl2}}
    -> 2 \OX{o2,\ox{+1,Na
    }}\pch + 2 \OX{r2,\ox
    {-1,Cl}}\mch }
\redox(o1,o2)[green,-
    stealth]{\small OX:
    $- 2\el$}
\redox(r1,r2)[purple,-
    stealth][-1]{\small
    RED: $+ 2\el$}
```

In v1.1 there was the command `\setredoxdist{<dim>}`. By using the package option `version=1` it is provided again.

`\setredoxdist{<dim>}` is not provided any more. Use package option `version=1` to reactivate it.

12 (Standard) State, Thermodynamics

12.1 Thermodynamic Variables

The following commands use siunitx:

```
\Enthalpy[<keyval>](<subscript>){<value>}
\Entropy[<keyval>](<subscript>){<value>}
\Gibbs[<keyval>](<subscript>){<value>}
```

The command `\Enthalpy` has a new syntax with v2.0. The package option `version=1` restores the former syntax.

The command `\Entropy` has a new syntax with v2.0. The package option `version=1` restores the former syntax.

The command `\Gibbs` has a new syntax with v2.0. The package option `version=1` restores the former syntax.

Their usage is pretty much self-explaining:

$\Delta H^\ominus = 123\,\mathrm{kJ\,mol^{-1}}$
$S^\ominus = 123\,\mathrm{J\,K^{-1}\,mol^{-1}}$
$\Delta G^\ominus = 123\,\mathrm{kJ\,mol^{-1}}$

```
\Enthalpy{123} \\
\Entropy{123} \\
\Gibbs{123}
```

The argument `(<subscript>)` adds a subscript for specification: `\Enthalpy(r){123}` $\Delta_\mathrm{r} H^\ominus = 123\,\mathrm{kJ\,mol^{-1}}$.

There are several keys to customize the commands.

- `exponent = <anything>`
- `delta = <anything>/false`
- `subscript = left/right`
- `unit = <unit>`

`exponent = <anything>`
default: `\standardstate`
no module

`delta = <anything>/false`
no module

`subscript = left/right`
no module

`unit = <unit>`
no module

The default values depend on the command.

$\Delta H^\ominus = -285\,\mathrm{kJ}$
$G^\ominus = 0\,\mathrm{kJ\,mol^{-1}}$
$\Delta S = 56.7\,\mathrm{J\,K^{-1}\,mol^{-1}}$

```
\Enthalpy[unit=\kilo\
    joule]{-285} \\
\Gibbs[delta=false]{0} \\
\Entropy[delta=\Delta,
    exponent=]{56.7}
```

The unit is set corresponding to the rules of siunitx and depends on its settings:

```
\Enthalpy{-1234.56e3} \\
\sisetup{per-mode=symbol,exponent-product=\cdot,output-
    decimal-marker={,},group-four-digits=true}
\Enthalpy{-1234.56e3}
```

$\Delta H^\ominus = -1234.56 \times 10^3\,\mathrm{kJ\,mol^{-1}}$
$\Delta H^\ominus = -1\,234{,}56 \cdot 10^3\,\mathrm{kJ/mol}$

Create New Variables

You can use the command

`\setnewstate[<keyval>]{<name>}{<symbol>}{<unit>}`

The command \setnewstate has a new syntax with v2.0. The package option version=1 restores the former syntax.

to create new corresponding commands:

```
\setnewstate{Helmholtz}{A
    }{\kilo\joule\per\
    mole}
\setnewstate[subscript-
    left=false,exponent
    =]{ElPot}{E}{\volt}
\Helmholtz{123.4} \\
\ElPot{-1.1} \\
\ElPot[exponent=0]($\ce{
    Sn}|\ce{Sn \pch
    [2]}||\ce{Pb \pch
    [2]}|\ce{Pb}$){0.01}
```

$\Delta A^\ominus = 123.4\,\mathrm{kJ\,mol^{-1}}$
$\Delta E = -1.1\,\mathrm{V}$
$\Delta E^0_{\mathrm{Sn|Sn^{2+}||Pb^{2+}|Pb}} = 0.01\,\mathrm{V}$

The command has some keys with which the default behaviour of the new command can be set.

- `exponent = <anything>`
- `delta = <anything>/false`
- `subscript-left = true/false`
- `subscript = <anything>`

`exponent = <anything>`
default: \standardstate
no module

`delta = <anything>/false`
default: \Delta
no module

`subscript-left = true/false`
default: true
no module

`subscript = <anything>`
no module

Redefine Variables

With

`\renewstate[<keyval>]{<name>}{<symbol>}{<unit>}`

you can redefine the already existing commands:

```
\renewstate{Enthalpy}{h
    }{\joule}
\Enthalpy(f){12.5}
```

$\Delta_\mathrm{f} h^\ominus = 12.5\,\mathrm{J}$

The command \renewstate has a new syntax with v2.0. The package option version=1 restores the former syntax.

The command is analogous to \setnewstate, i. e. it has the same keys.

So – for following thermodynamic conventions – one could define a molar and an absolute variable:

$\Delta h = -12.3\,\mathrm{kJ\,mol^{-1}}$ $\Delta H = -12.3\,\mathrm{kJ}$

```
\setnewstate[exponent=]{
    enthalpy}{h}{\kilo\
    joule\per\mole}%
    molar
\renewstate[exponent=]{
    Enthalpy}{H}{\kilo\
    joule}% absolut
\enthalpy{-12.3} \
    Enthalpy{-12.3}
```

12.2 State

The commands presented in section 12.1 internally all use the command

```
\State[<keyval>]{<symbol>}{<subscript>}
```

Please note that {<subscript>} is an *optional* argument.

The command \State has a new syntax with v2.0. The package option version=1 restores the former syntax.

It can be used to write the thermodynamic variables without value and unit.

Examples:

$\Delta A^{\ominus}$, $\Delta_\mathrm{f} G^{\ominus}$, $\Delta E^{\ominus}_\mathrm{Na}$, $\Delta H^{1000\,^\circ\mathrm{C}}$

```
\State{A}, \State{G}{f},
    \State[subscript-left
    =false]{E}{\ce{Na}},
    \State[exponent=\SI
    {1000}{\celsius}]{H}
```

Again there are some keys to customize the command:

- exponent = <anything>
- subscript-left = true/false
- delta = <anything>/false

exponent = <anything>
default: \standardstate
module: state

subscript-left = true/false
default: true
module: state

delta = <anything>/false
default: \Delta
module: state

In v1.1 there was the command \setstatesubscript{<subscript pos>}. By using the package option version=1 it is provided again.

\setstatesubscript{<subscript pos>} is not provided any more. Use package option version=1 to reactivate it.

13 Spectroscopy

If substances are examined wether they are what they're supposed to, one often needs NMR spectroscopy. Measured results then are written in a way like: ^{1}H-NMR (400 MHz, $CDCl_3$): $\delta = 1.59\ldots$ chemmacros provides a command which simplifies writing this down (uses siunitx).

If you want the command to look like the corresponding bpchem commands (bpchem command \HNMR ^{1}H-NMR: δ and chemmacros command \NMR ^{1}H-NMR: δ), then you use the package option bpchem (see section 3).

```
\NMR{<num>,<elem>}(<num>,<unit>)[<solvent>]
\NMR*{<num>,<elem>}(<num>,<unit>)[<solvent>]
```

All arguments are optional! Without arguments we get:

^{1}H-NMR: δ ^{1}H-NMR	1 \NMR \\ 2 \NMR*

All arguments can be combined freely, the command can also be used in math mode.

With the first argument you can specify the kind of NMR:

^{13}C-NMR: δ	1 \NMR{13,C}

With the second argument the frequency (in MHz) can be specified:

^{1}H-NMR (400 MHz): δ	1 \NMR(400)

You also can change the unit:

^{1}H-NMR (4×10^{8} Hz): δ	1 \NMR(4e8,\hertz)

Please note that the setup of siunitx also has an impact on this command:

^{1}H-NMR ($4 \cdot 10^{8}$ Hz): δ	1 \sisetup{exponent-product =\cdot}\NMR(4e8,\ hertz)

And finally with the third argument the solvent can be specified:

^{1}H-NMR ($CDCl_3$): δ	1 \NMR[CDCl3]

With the keys `unit` and `nucleus` the default unit and the default nucleus can be changed.

Examples:

^{13}C-NMR (100 MHz): δ ^{13}C-NMR (100 MHz) ^{19}F-NMR ($CFCl_3$) ^{19}F-NMR (285 MHz, $CFCl_3$) ^{1}H-NMR (400 MHz, $CDCl_3$): δ = 1.59 (q, 1H, J = 11.6 Hz, H-4)	1 {\chemsetup[nmr]{nucleus ={13,C}}\NMR(100) \ NMR*(100) } \\ 2 \NMR*{19,F}[CFCl3] \NMR *{19,F}(285)[CFCl3] \\ 3 \NMR(400)[CDCl3] = \num {1.59} (q, 1H, \ textit{J} = \SI {11.6}{\hertz}, H-4)

`unit = <unit>`
default: \mega\hertz
module: nmr

`nucleus = {<num>,<elem>}`
default: {1,H}
module: nmr

14 Commands for mhchem

From v2.0 chemmacros loads the package mhchem.

Before the commands are described some words on using commands inside the \ce and \cee commands. Probably due to the way these commands are processed there can be difficulties especially when using commands with arguments.

Often you have to leave blank spaces:

```
\ce{Na\pch}\\    % no problem
\ce{Ca\pch[2]}\\ % displayed wrong
\ce{Ca \pch[2]}\\ % displayed right
\ce{Ca$\pch[2]$} % displayed right
```

Na^+
$Ca^+[_2]$
Ca^{2+}
Ca^{2+}

You also need to put curly braces at the end of commands:

```
\ce{\mch OMe}\\ % displayed wrong
\ce{\mch{} OMe}\\ % displayed right
\ce{$\mch$OMe}  % displayed right
```

^-OMe
$^-$OMe
$^-$OMe

This is *not* only true for chemmacros commands!

```
\ce{A \quad B} \ce{Na2\textbf{O}}\\ % displayed wrong
\ce{A \quad{} B} \ce{Na2 \textbf{O}}\\ % displayed right
\ce{A $\quad$ B} \ce{Na2 \textbf{O}} % displayed right
```

A B Na_2O
A B $Na_2\mathbf{O}$
A B $Na_2\mathbf{O}$

As you can see in most cases instead of using blank spaces or curly braces you can also put the according command between `$ $`.

14.1 Reaction Environments

Defined by chemmacros

You can use these environments for numbered...

```
\begin{reaction}
<mhchem code>
\end{reaction}
\begin{reactions}
<mhchem code>
\end{reactions}
```

...and their starred versions for unnumbered reactions.

```
\begin{reaction*}
<mhchem code>
\end{reaction*}
\begin{reactions*}
<mhchem code>
\end{reactions*}
```

With them you can create (un)numbered reaction equations similar to mathematical equations.

The environments `reaction`/`reaction*` use the `equation`/`equation*` environments and the environments `reactions`/ `reactions*` use the `align`/`align*` environments to display the reactions.

```
Reaction with counter:
\begin{reaction}
 A -> B
\end{reaction}
```

Reaction with counter:

$$\mathrm{A} \longrightarrow \mathrm{B} \tag{1}$$

```
Reaction without counter:
\begin{reaction*}
 C -> D
\end{reaction*}
```

Reaction without counter:

$$\mathrm{C} \longrightarrow \mathrm{D}$$

```
Several aligned reactions with counter:
\begin{reactions}
 A     &-> B + C \\
 D + E &-> F
\end{reactions}
```

Several aligned reactions with counter:

$$\begin{aligned} \mathrm{A} &\longrightarrow \mathrm{B} + \mathrm{C} && \{2\} \\ \mathrm{D} + \mathrm{E} &\longrightarrow \mathrm{F} && \{3\} \end{aligned}$$

```
Several aligned reactions without counter:
\begin{reactions*}
 G     &-> H + I \\
 J + K &-> L
\end{reactions*}
```

Several aligned reactions without counter:

$$\begin{aligned} \mathrm{G} &\longrightarrow \mathrm{H} + \mathrm{I} \\ \mathrm{J} + \mathrm{K} &\longrightarrow \mathrm{L} \end{aligned}$$

If you want to change the layout of the counter tags, you can use `\renewtagform{<tagname>}[<format>]{<right delim>}{<left delim>}`[21].

[21] Provided by the mathtools package

```
\renewtagform{reaction}[R \textbf]{[}{]}
\begin{reaction}
 H2O + CO2 <<=> H2CO3
\end{reaction}
```

$$H_2O + CO_2 \rightleftharpoons H_2CO_3 \qquad [R\ \mathbf{4}]$$

Note, that the name of the tagform has changed with v2.0. It was called `CMreaction` in v1.*. Option `version=1` restores this name.

Own Reactions

You can create new types of reactions with the command:

```
\newreaction[<keyval>]{<name>}{<math name>}
```

The command \newreaction has a new syntax with v2.0. The package option `version=1` restores the former syntax.

`<name>` will be the name of the new environment. `<math name>` is the used math environment.

The command has two keys. There is `star`, which will also define a starred version of the new environment, if the starred math environment exists. If it doesn't exist, this will cause an error.

`star = true/false`
default: `false`
no module

Then there is `arg`, which is used to define an environment with a mandatory argument. Of course this only works, if the used math environment has a mandatory argument.

`arg = true/false`
default: `false`
no module

The predefined environments are defined via
`\newreaction[star]{reaction}{equation}` and
`\reaction[star]{reactions}align`.

Let's suppose, you'd like to have the alignment behaviour of the `alignat` environment for mhchem reactions. You could do the following:

`\newreaction[star,arg]{reactionsat}{alignat}`

With this the `reactionsat` environment is defined.

```
\newreaction[star,arg]{reactionsat}{alignat}
\begin{reactionsat}{3}
 A     &-> B     &&-> C     &&-> D \\
 aaaaa &-> bbbbb &&-> ccccc &&-> ddddd
\end{reactionsat}
\begin{reactionsat*}{2}
 A     &-> B     & C           &-> D \\
 aaaaa &-> bbbbb &\quad{} ccccc &-> ddddd
\end{reactionsat*}
```

$$\begin{aligned} A &\longrightarrow B &&\longrightarrow C &&\longrightarrow D && \{5\} \\ aaaaa &\longrightarrow bbbbb &&\longrightarrow ccccc &&\longrightarrow ddddd && \{6\} \end{aligned}$$

$$\begin{aligned} A &\longrightarrow B & C &\longrightarrow D \\ aaaaa &\longrightarrow bbbbb & ccccc &\longrightarrow ddddd \end{aligned}$$

14.2 Phases

These commands are intended to indicate the phase of a compound. Although these commands were intended for the use with mhchem they can be used without it as well.

- `\sld[<anything>]` (s)
- `\lqd[<anything>]` (l)
- `\gas` (g)
- `\aq` (aq)

`\solid` is not provided any more. Use package option `version=1` to reactivate it.
`\liquid` is not provided any more. Use package option `version=1` to reactivate it.

Please notice, that the commands `\solid` and `\liquid` are now called `\sld` and `\lqd`, respectively.

```
\ce{C\sld{} + 2 H2O\lqd{} -> CO2\gas{} + 2 H2\gas}\\
To make it complete: NaCl\aq.
```

$\mathrm{C_{(s)} + 2\,H_2O_{(l)} \longrightarrow CO_{2(g)} + 2\,H_{2(g)}}$
To make it complete: $\mathrm{NaCl_{(aq)}}$.

With the package option `german` (see section 3) or by using the optional arguments you get the german versions:

```
{\chemsetup[option]{german=true}
\ce{C\sld{} + 2 H2O\lqd{} -> CO2\gas{} + 2 H2\gas} }\\
\ce{C \sld[f] + 2 H2O \lqd[fl] -> CO2\gas{} + 2 H2\gas}
```

$\mathrm{C_{(f)} + 2\,H_2O_{(fl)} \longrightarrow CO_{2(g)} + 2\,H_{2(g)}}$
$\mathrm{C_{(f)} + 2\,H_2O_{(fl)} \longrightarrow CO_{2(g)} + 2\,H_{2(g)}}$

If you looked closely, you have probably noticed that the german `\lqd` isn't identical to `\lqd[fl]` but to `\lqd[f\/l]`. This makes the subscript more readable.

Of course it doesn't matter which command with optional argument you use. Both of them just write a subscript with braces. `\sld[f]` is identical to `\lqd[f]`.

One can think of other uses, too:

$\mathrm{C_{(graphite)}}$

```
C\sld[graphite]
```

14.3 Text Under Compounds

chemmacros provides a command, with which you can place text below of compounds.

```
\mhName[<keyval>]{<formula>}{<text>}
```

The command \mhName has a new syntax with v2.0. The package option version=1 restores the former syntax.

For example:

```
1 \ce{4 C2H5Cl + Pb / Na -> \mhName{Pb(C2H5)4}{former
    antiknock additive} + NaCl}
```

$$4\,C_2H_5Cl + Pb/Na \longrightarrow \underset{\substack{\text{former}\\ \text{antiknock}\\ \text{additive}}}{Pb(C_2H_5)_4} + NaCl$$

align = <alignment>
default: \centering
module: mhName

There are several keys to customize \mhName.

- align
- fontattr
- fontsize
- width

fontattr = <commands>
module: mhName

fontsize = <fontsize>
default: \tiny
module: mhName

width = <dim>
module: mhName

```
1 \ce{4 C2H5Cl + Pb / Na -> \mhName[fontsize=\
    footnotesize]{Pb(C2H5)4}{former antiknock additive}
    + NaCl}\\
2 \chemsetup[mhName]{align=\raggedright,fontsize=\small,
    fontattr=\bfseries\color{red},width=3cm}
3 \ce{4 C2H5Cl + Pb / Na -> \mhName{Pb(C2H5)4}{former
    antiknock additive} + NaCl}
```

$$4\,C_2H_5Cl + Pb/Na \longrightarrow \underset{\substack{\text{former}\\ \text{antiknock}\\ \text{additive}}}{Pb(C_2H_5)_4} + NaCl$$

$$4\,C_2H_5Cl + Pb/Na \longrightarrow \underset{\textbf{former antiknock additive}}{Pb(C_2H_5)_4} + NaCl$$

In v1.1 there was the command \setmhName{<textattr>}. By using the package option version=1 it is provided again.

\setmhName{<textattr>} is not provided any more. Use package option version=1 to reactivate it.

15 Newman Projections

chemmacros provides the command

```
\newman[<keyval>](<angle>){<1>,<2>,<3>,<4>,<5>,<6>}
```

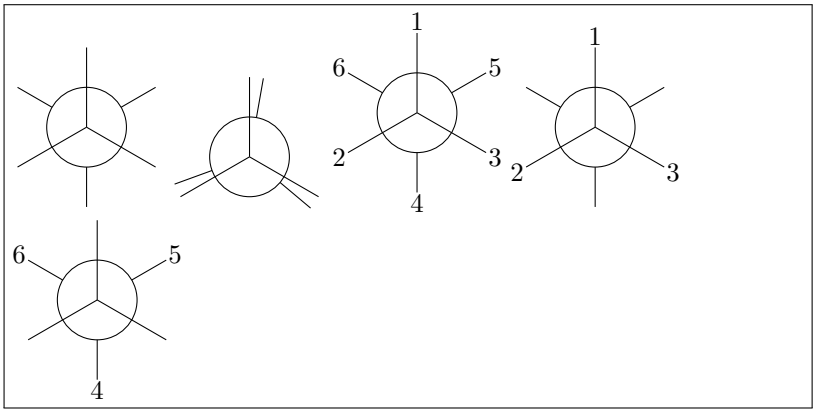
The command \newman has a new syntax with v2.0. The package option version=1 restores the former syntax.

that allows you to create newman projections (uses Ti*k*Z). With <angle> the back atoms are rotated counter clockwise with respect to the front atoms.

angle = <angle>
default: 0
module: newman

scale = <factor>

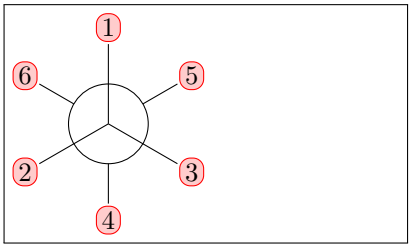
```
\newman{} \newman(170){}
\newman{1,2,3,4,5,6} \newman{1,2,3} \newman{,,,4,5,6}
```



Several keys allow customization:

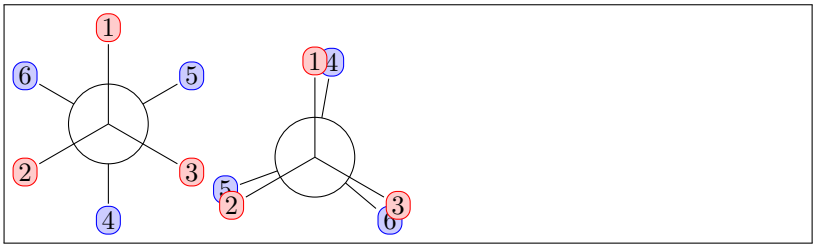
- `angle = <angle>` default angle
- `scale = <factor>` scale the whole projection
- `ring = <tikz>` customize the ring with Ti*k*Z keys
- `atoms = <tikz>` customize the nodes within which the atoms are set
- `back-atoms = <tikz>` explicitly customize the back atoms

```
\chemsetup[newman]{angle=45} \newman{}
\newman[scale=.75,ring={draw=blue,fill=blue!20}]{}
```




```
\chemsetup[newman]{atoms
    ={draw=red,fill=red
    !20,inner sep=2pt,
    rounded corners}}
\newman{1,2,3,4,5,6}
```

```
\chemsetup[newman]{
  atoms = {draw=red,fill=red!20,inner sep=2pt,rounded
      corners},
  back-atoms = {draw=blue,fill=blue!20,inner sep=2pt,
      rounded corners}
}
\newman{1,2,3,4,5,6} \newman(170){1,2,3,4,5,6}
```



16 s, p, and Hybrid Orbitals

In v1.1 there have been the commands \porb[<options>], \phorb[<options>], \pxorb, \pyorb und \pzorb. By using the package option version=1 they are provided again

chemmacros provides the following command to create orbitals:

\porb[<options>] is not provided any more. Use package option version=1 to reactivate it.
\phorb[<options>] is not provided any more. Use package option version=1 to reactivate it.
\pxorb is not provided any more. Use package option version=1 to reactivate it.
\pyorb is not provided any more. Use package option version=1 to reactivate it.
\pzorb is not provided any more. Use package option version=1 to reactivate it.

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

There are the following types available for <type>:

- s
- p
- sp
- sp2
- sp3

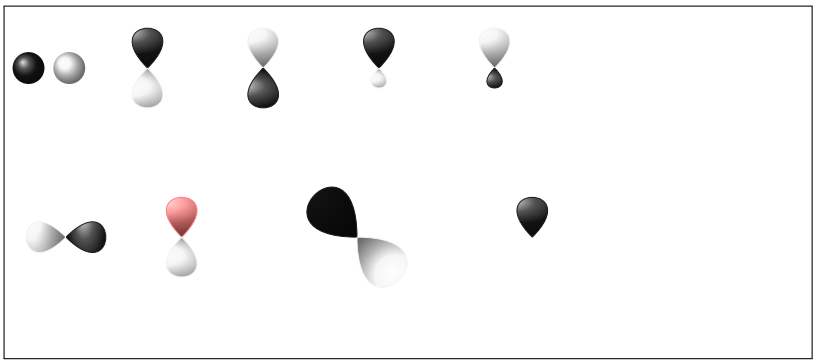
```
\orbital{s} \orbital{p} \orbital{sp} \orbital{sp2} \
    orbital{sp3}
```

Depending on the type you have different keys to modify the orbitals:

- `phase = +/-` changes the phase of the orbital (all types)
- `scale = <factor>` changes the size of the orbital (all types)
- `color = <color>` changes the color of the orbital (all types)
- `angle = <angle>` rotates the orbitals with a p contribution counter clockwise (all types except `s`)
- `half = true/false` displays only half an orbital (only `p`)

```
\orbital{s} \orbital[phase=-]{s}
\orbital{p} \orbital[phase=-]{p}
\orbital{sp3} \orbital[phase=-]{sp3}

\orbital[angle=0]{p} \orbital[color=red!50]{p} \orbital
    [angle=135,scale=1.5]{p} \orbital[half]{p}
```



Additionally there are two keys, with which the Ti*k*Z behaviour can be changed.

- `overlay = true/false` the orbital "doesn't need space"; it is displayed with the Ti*k*Z option `overlay`.
- `opacity = <num>` the orbital becomes transparent; `<value>` can have values between `1` (fully opaque) to `0` (invisible).

`phase = +/-`
default: `+`
module: `orbital/<type>`

`scale = <factor>`
default: `1`
module: `orbital/<type>`

`color = <color>`
default: `black`
module: `orbital/<type>`

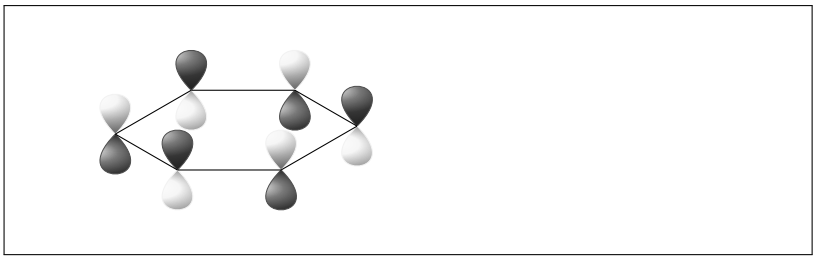
`angle = <angle>`
default: `90`
module: `orbital/<type>`

`half = true/false`
default: `false`
module: `orbital/<type>`

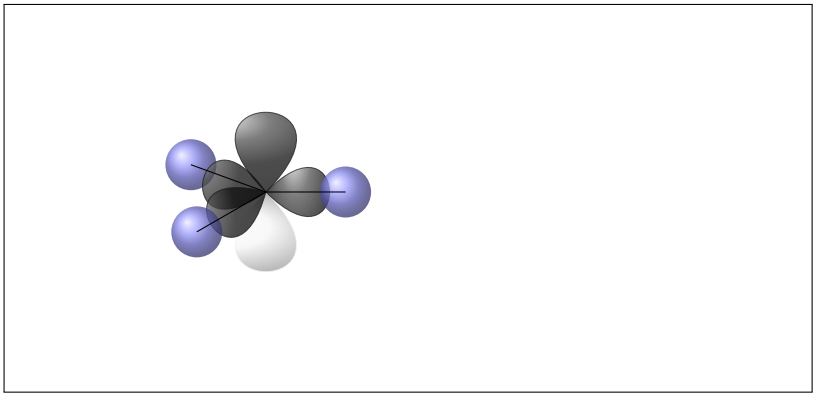
`overlay = true/false`
default: `false`
module: `orbital`

`opacity = <num>`
default: `1`
module: `orbital`

```
\vspace{1cm}\hspace{1cm}
\chemsetup[orbital]{
  overlay,
  p/color = black!70
}
\setbondoffset{0pt}
\chemfig{?\orbital{p}-[,1.3]{\orbital[phase=-]{p
    }}-[:30,1.1]\orbital{p}-[:150,.9]{\orbital[phase=-]{
    p}}-[4,1.3]\orbital{p}-[:-150,1.1]{\orbital[phase
    =-]{p}}?}
\vspace{1cm}
```



```
\vspace{2cm}\hspace{2cm}
\setbondoffset{0pt}
\chemsetup[orbital]{
  overlay ,
  opacity = .75 ,
  p/scale = 1.6 ,
  s/color = blue!50 ,
  s/scale = 1.6
}
\chemfig{\orbital{s}-[:-20]{\orbital[scale=2]{p}}{\
    orbital[half,angle=0]{p}}{\orbital[angle=170,half]{p
    }}{\orbital[angle=-150,half]{p}}(-[:-150]\orbital{s
    })-\orbital{s}}
\vspace{2cm}
```



17 Key Overview

In the table below all keys provided by chemmacros for customization are listed. All keys that belong to a module; can be set with `\chemsetup[<module>]{<keyval>}` or `\chemsetup{<module>/<keyval>}`.

Some keys can be set without value. Then the underlined value is used.

key	module	values	default	
Paket-Optionen:				
bpchem	option	true/false	false	page 3
circled	option	formal/all/none	formal	page 3
circletype	option	chem/math	chem	page 3
EZ	option	chemmacros/cool	chemmacros	page 3
german	option	true/false	false	page 3
version	option	1/2	2	page 4
xspace	option	true/false	true	page 4
\ba, \Nu:				
elpair	base	dots/dash/false	false	page 5
\pch, \mch, \fpch, \fmch:				
append	charges	true/false	false	Seite 9
\ox:				
parse	ox	true/false	true	page 9
roman	ox	true/false	true	page 9
pos	ox	top/super/side	top	page 10
explicit-sign	ox	true/false	false	page 10
decimal-marker	ox	comma/point	point	page 10
\OX, \redox:				
dist	redox	<dim>	.6em	page 12
sep	redox	<dim>	.2em	page 13
\Enthalpy, \Entropy, \Gibbs:				
exponent		<anything>	\standardstate	page 14
delta		<anything>/false		page 14
subscript		left/right		page 14
unit		<unit>		page 14
\setnewstate, \renewstate:				
exponent		<anything>	\standardstate	page 15
delta		<anything>/false		page 15
subscript		<anything>		page 15
subscript-left		true/false		page 15
\State:				
exponent		<anything>	\standardstate	page 16
delta		<anything>/false		page 16
subscript-left		true/false		page 16
\NMR:				
unit	nmr	<unit>	\mega\hertz	page 17
nucleus	nmr	{<num>,<atom symbol>}	{1,H}	page 17
\newreaction:				

key	module	values	default	
star		true/false	false	page 20
arg		true/false	false	page 20
\mhName:				
align	mhName	<alignment>	\centering	page 22
fontattr	mhName	<commands>		page 22
fontsize	mhName	<fontsize>	\tiny	page 22
width	mhName	<dim>		page 22
\newman:				
angle	newman	<angle>	0	page 22
scale	newman	<factor>	1	page 22
ring	newman	<tikz>		page 22
atoms	newman	<tikz>		page 22
back-atoms	newman	<tikz>		page 22
\orbital <type> = s/p/sp/sp2/sp3:				
phase	orbital/<type>	+/-	+	page 25
scale	orbital/<type>	<factor>	1	page 25
color	orbital/<type>	<color>	black	page 25
angle	orbital/<type>	<angle>	90	page 25
half	orbital/p	true/false	false	page 25
overlay	orbital	true/false	false	page 25
opacity	ornital	<num>	1	page 25

18 Suggestions and Bug Reports

Feedback on chemmacros is highly appreciated and welcome! If you have suggestions for macros, missing features etc., please don't hesitate to cantact me. If you recognize any errors, be it chemical ones, wrong documentation and the like, I'd be grateful about a short email to contact@mychemistry.eu. If you find any bugs, it would be best, if you'd send me a minimal example, with which I can reproduce the bug.

Many thanks to all the people, who already provided me with feedback!