CHEMFORMULA

v4.17 2022/01/23

typeset chemical compounds and reactions

Clemens Niederberger

http://www.mychemistry.eu/forums/forum/chemformula/

contact@mychemistry.eu

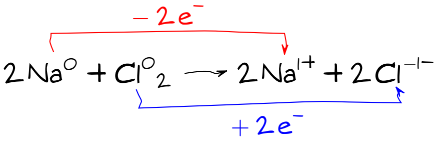


Table of Contents

1. Introduction

Probably every chemist using LaTeX 2ε is aware of the great mhchem package by Martin HENSEL. There have always been some difficulties intertwining it with the CHEMMACROS package, though. Also, some other minor points in mhchem always bothered me, but they hardly seemed enough for a new package. They weren't even enough for a feature request to the mhchem author. The challenge and the fun of creating a new package and the wish for a highly customizable alternative led to CHEMFORMULA after all.

CHEMFORMULA works very similar to mhchem but is more strict as to how compounds, stoichiometric factors and arrows are input. In the same time CHEMFORMULA offers *many* possibilities to customize the output.

2. Licence and Requirements

Permission is granted to copy, distribute and/or modify this software under the terms of the LaTeX Project Public License (LPPL), version 1.3c or later (`http://www.latex-project.org/lppl.txt`). The software has the status "maintained."

The CHEMFORMULA package needs and thus loads the packages l3kernel [L3Pa], xparse, l3keys2e and xfrac (all three are part of the l3packages bundle [L3Pb]), tikz[1] [Tan19], amstext [MS00], nicefrac [Rei98] and scrlfile (from the KOMA-Script[2] bundle [Koh19]).

3. Setup

If you're using CHEMFORMULA as a standalone package options are set up with the following command:

`\setchemformula{⟨options⟩}`
Set up CHEMFORMULA.

CHEMFORMULA is tightly intertwined with the CHEMMACROS package. If both packages are loaded together, CHEMFORMULA is integrated into the CHEMMACROS package. Then all of CHEMFORMULA's options belong to CHEMMACROS' module `chemformula`. This means if you load it via CHEMMACROS or in addition to CHEMMACROS they can be setup with

---

[1] on CTAN as pgf: `http://mirrors.ctan.org/graphics/pgf/`

[2] on CTAN as koma-script: `http://mirrors.ctan.org/macros/latex/contrib/koma-script/`

`\chemsetup[chemformula]{⟨options⟩}`
Set up options for CHEMFORMULA exclusively, or

`\chemsetup{chemformula/⟨option1⟩,chemformula/⟨option2⟩}`
Set up options for CHEMFORMULA together with others of CHEMMACROS' options.

4. The Basic Principle

CHEMFORMULA offers one main command.

`\ch[⟨options⟩]{⟨input⟩}`
CHEMFORMULA's main command.

The usage will seem very familiar to you if you're familiar with mhchem:

```
\ch{H2O} \par
\ch{Sb2O3} \par
\ch{H+} \par
\ch{CrO4^2-} \par
\ch{AgCl2-} \par
\ch{[AgCl2]-} \par
\ch{Y^{99}+} \par
\ch{Y^{99+}} \par
\ch{H2_{(aq)}} \par
\ch{NO3-} \par
\ch{(NH4)2S} \par
\ch{^{227}_{90}Th+} \par
$V_{\ch{H2O}}$ \par
\ch{Ce^{IV}} \par
\ch{KCr(SO4)2 * 12 H2O}
```

$\mathrm{H_2O}$

$\mathrm{Sb_2O_3}$

$\mathrm{H^+}$

$\mathrm{CrO_4{}^{2-}}$

$\mathrm{AgCl_2{}^-}$

$\mathrm{[AgCl_2]^-}$

$\mathrm{Y^{99+}}$

$\mathrm{Y^{99+}}$

$\mathrm{H_{2(aq)}}$

$\mathrm{NO_3{}^-}$

$\mathrm{(NH_4)_2S}$

$\mathrm{{}^{227}_{90}Th^+}$

$V_{\mathrm{H_2O}}$

$\mathrm{Ce^{IV}}$

$\mathrm{KCr(SO_4)_2 \cdot 12\,H_2O}$

However, there are differences. The most notable one: CHEMFORMULA distinguishes between different types of input. These different parts *have* to be separated with blanks:

```
\ch{part1 part2 part3 part4}
```

A blank in the input *never* is a blank in the output. This role of the blank strictly holds and disregarding it can have unexpected results and even lead to errors.

The most visible differences regard spacing and the shapes of the default arrows:

```
\ch{A + B ->[a] C} \par
\ce{A + B ->[a] C}
```

$\mathrm{A + B} \xrightarrow{\quad a \quad} \mathrm{C}$

$\mathrm{A + B} \xrightarrow{a} \mathrm{C}$

This means that `\ch{2H2O}` is recognized as a *single* part, which in this case is recognized as a compound.

```
\ch{2H2O} \par
\ch{2 H2O}
```

$_2H_2O$
$2\,H_2O$

This also means, that a part cannot contain a blank since this will automatically divide it into two parts. If you need an extra blank in the output you need to use `~` or `\ `. However, since commands in most cases gobble a space after them a input like `\ch{\command ABC}` will be treated as a single part. If you want or need to divide them you need to add an empty group: `\ch{\command{} ABC}`. The different input types are described in the following sections.

5. Stoichiometric Factors

A stoichiometric factor may only contain of numbers and the signs `.,_/()`

```
\ch{2} \par
\ch{12}

% decimals:
\ch{.5} \par
\ch{5,75}

% fractions:
\ch{3/2} \par
\ch{1_1/2}

% ``iupac'':
\ch{(1/2)}
```

2
12
0.5
5.75
$\frac{3}{2}$
$1\frac{1}{2}$
(1/2)

As you can see if you input decimal numbers a missing leading zero is added.

You have to be a little bit careful with the right syntax but I believe it is rather intuitive.

```
this won't work but will result in an error: \ch{1/1_1}
```

If stoichiometric factors are enclosed with parentheses the fractions are not recognized and missing leading zeros are not added. What's inside the parentheses is typeset as is.

```
\ch{(1/2) H2O} \ch{1/2 H2O} \ch{0.5 H2O}
```

(1/2) H_2O $\frac{1}{2}$ H_2O 0.5 H_2O

You can find many examples like the following for stoichiometric factors in parentheses in the IUPAC Green Book [Coh+08]:

$$(1/5)\,\mathrm{KMn^{VII}O_4} + (8/5)\,\mathrm{HCl} \quad = \quad (1/5)\,\mathrm{Mn^{II}Cl_2} + (1/2)\,\mathrm{Cl_2} + (1/5)\,\mathrm{KCl} + (4/5)\,\mathrm{H_2O}$$

There are a few possibilities to customize the output:

`decimal-marker = {⟨marker⟩}` Default: `.`

The symbol to indicate the decimal.

`frac-style = math|xfrac|nicefrac` Default: `math`

Determines how fractions are displayed.

`frac-math-cmd = {⟨command sequence⟩}` Default: `\frac{#1}{#2}`

Changed in version 4.17 (2022/01/23)

Allows you to choose which command is used with `frac-style = {math}`.

`stoich-space = {⟨skip⟩}` Default: `.1667em plus .0333em minus .0117em`

The space that is placed after the stoichiometric factor. A rubber length.

`stoich-paren-parse = true|false` Default: `false`

If set to true stoichiometric factors enclosed by parentheses also are parsed.

`stoich-format = {⟨code⟩}` (initially empty)

☆ New

Allows formatting of the stoichiometric factors. The last macro given may take an argument.

`stoich-print = {⟨code containing #1⟩}` Default: `#1`

Changed in version 4.17

This option allows to use specify a macro for printing the stoichiometric factors where `#1` represents the factor. *This option is depecated and will be removed some time in the future. Use* `stoich-format` *instead.*

```
\ch[decimal-marker={,}]{3.5} \ch[decimal-marker={$\cdot$}]{3,5}
```

3,5 3·5

The option `frac-style = {xfrac}` uses the `\sfrac` command of the xfrac package. The output strongly depends on the font you use.

```
\ch[frac-style=xfrac]{3/2} \ch[frac-style=xfrac]{1_1/2}
```

³⁄₂ 1½

CHEMFORMULA defines the instance `chemformula-text-frac` which you can redefine to your needs. See the xfrac documentation for further information. The default definition is this:

```
\DeclareInstance{xfrac}{chemformula-text-frac}{text}
 {
   slash-left-kern  = -.15em ,
   slash-right-kern = -.15em
 }
```

This document uses the font Linux Libertine O and the following definition:

```
\DeclareInstance{xfrac}{chemformula-text-frac}{text}
 {
   scale-factor        = 1 ,
   denominator-bot-sep = -.2ex ,
   denominator-format  = \scriptsize #1 ,
   numerator-top-sep   = -.2ex ,
   numerator-format    = \scriptsize #1 ,
   slash-right-kern    = .05em ,
   slash-left-kern     = .05em
 }
```

The option `frac-style` = `{nicefrac}` uses the `\nicefrac` command of the nicefrac package.

```
\ch[frac-style=nicefrac]{3/2} \ch[frac-style=nicefrac]{1_1/2}
```

$^{3}/_{2}$ $1^{1}/_{2}$

The option `stoich-space` allows you to customize the space between stoichiometric factor and the group following after it.

```
\ch{2 H2O} \par
\ch[stoich-space=.3em]{2 H2O}
```

$2\,H_2O$

$2\;H_2O$

6. Compounds

CHEMFORMULA determines compounds as the type that “doesn’t fit in anywhere else.” This point will become more clear when you know what the other types are.

```
\ch{H2SO4} \par
\ch{[Cu(NH3)4]^2+}
```

H_2SO_4

$[Cu(NH_3)_4]^{2+}$

6.1. Adducts

CHEMFORMULA has two identifiers which will create adducts.

```
\ch{A.B}
```

$A\cdot B$

```
\ch{A*B}
```

$A\cdot B$

```
\ch{CaSO4.H2O} \par
\ch{CaSO4*H2O}
```

$CaSO_4\cdot H_2O$

$CaSO_4\cdot H_2O$

Since numbers in a compound always are treated as subscripts (see section 6.2) you sometimes need to introduce stoichiometric factors for the right output:

```
\ch{Na3PO4*12H2O} \par
\ch{Na3PO4* 12 H2O} \par
\ch{Na3PO4 * 12 H2O}
```

$Na_3PO_4\cdot{}_{12}H_2O$

$Na_3PO_4\cdot 12\,H_2O$

$Na_3PO_4\cdot 12\,H_2O$

6.2. Subscripts

All numbers in a compound are treated as subscripts.

```
\ch{H2SO4}
```

H_2SO_4

If you want a letter to be a subscript you can use the math syntax:

```
\ch{A_nB_m}
```

$A_\mathrm{n}B_\mathrm{m}$

The subscript recognizes groups. You can also use math inside it.

```
\ch{A_{$n$}B_{$m$}} \par
\ch{NaCl_{(aq)}}
```

A_nB_m

$NaCl_{(aq)}$

6.3. Commands

Commands are allowed in a compound:

```
\ch{\textbf{A2}B3} \ch{A2\color{red}
  B3}
```

$\mathbf{A_2}\mathrm{B_3}$ $\mathrm{A_2B_3}$

However, if the commands demand numbers as argument, *e. g.*, space commands or CHEMMACROS’ `\ox` command the direct use will fail. This is because the numbers are treated as subscripts *before* the command expands.

```
\ch{A\hspace{2mm}B} will raise an error because \hspace sees something like
this: \hspace{$_2$mm}. Actually not at all like this but equally bad\ldots
```

See section 8.1 for a way around this.

6.4. Charges and Other Superscripts

Basics If a compound *ends* with a plus or minus sign it will be treated as charge sign and typeset as superscript. In other places a plus is treated as a triple bond and a dash will be used as a single bond, see section 6.5.

```
\ch{A+B} \ch{AB+} \par
\ch{A-B} \ch{AB-}
```

$\mathrm{A{\equiv}B}$ $\mathrm{AB^{+}}$

$\mathrm{A{-}B}$ $\mathrm{AB^{-}}$

For longer charge groups or other superscripts you can use the math syntax. It recognizes groups and you can use math inside them. Inside these groups neither `+` nor `-` are treated as bonds. If a dot `.` is inside a superscript it is treated as indicator for a radical. A `*` gives the excited state.

```
\ch{A^{x-}} \par
\ch{A^x-} \par
\ch{A^{x}-} \par
\ch{A^{$x-$}} \par
\ch{RNO2^{-.}} \par
\ch{^31H} \par
\ch{^{14}6C} \par
\ch{^{58}_{26}Fe} \par
\ch{NO^*}
```

$\mathrm{A^{x-}}$

$\mathrm{A^{x-}}$

$\mathrm{A^{x-}}$

A^{x-}

$\mathrm{RNO_2^{-\bullet}}$

$\mathrm{^{3}_{1}H}$

$\mathrm{^{14}_{6}C}$

$\mathrm{^{58}_{26}Fe}$

$\mathrm{NO^{*}}$

Changed in version 4.5a ()

Actually a dot `.` is not always treated as indicator for a radical: if the dot in the superscript is followed by a number it is interpreted as a decimal sign. It is typeset according to the option `decimal-marker`. This may be a good place to mention that a comma `,` in a superscript is also typeset according to `decimal-marker`.

```
\ch{^{22,98}_{11}Na}
\ch{^{22.98}_{11}Na}\par
\setchemformula{decimal-marker={,}}
\ch{^{22,98}_{11}Na}
\ch{^{22.98}_{11}Na}
```

${}^{22.98}_{11}\mathrm{Na}$ ${}^{22.98}_{11}\mathrm{Na}$
${}^{22,98}_{11}\mathrm{Na}$ ${}^{22,98}_{11}\mathrm{Na}$

Ions and ion composites with more than one charge can be typeset quite as easy:

```
\ch{SO4^2-} \ch{Ca^2+ SO4^2-}
```

$\mathrm{SO_4^{2-}}$ $\mathrm{Ca^{2+}SO_4^{2-}}$

Charge Commands You don't need to use `\mch` and related commands inside `\ch`. Indeed, you *shouldn't* use them as they might mess with the subscript and superscript alignment. The CHEMMACROS option `circled` is obeyed by `\ch`.

```
\chemsetup[charges]{circled=all}
\ch{H+ + OH- <=> H2O}
```

$\mathrm{H^{\oplus} + OH^{\ominus} \rightleftharpoons H_2O}$

CHEMFORMULA knows the options `circled` and `circletype` also on its own.

```
\setchemformula{circled=all}
\ch{H+ + OH- <=> H2O}
```

$\mathrm{H^{\oplus} + OH^{\ominus} \rightleftharpoons H_2O}$

These options are coupled with CHEMMACROS options, *i. e.*, setting CHEMMACROS' options will also set CHEMFORMULA's equivalents. The other way around the options act independently: setting CHEMFORMULA's options will *not* set CHEMMACROS' options.

`circled` = `formal|all|none` Default: `formal`

CHEMFORMULA uses two different kinds of charges which indicate the usage of real (+/−) and formal (⊕/⊖) charges. The choice `formal` distinguishes between them, choice `none` displays them all without circle, choice `all` circles all.

`circletype` = `chem|math` Default: `chem`

This option switches between two kinds of circled charge symbols: `\fplus` ⊕ and `$\oplus$` $\oplus$.

Behaviour The supercripts behave differently depending on their position in a compound, if there are super- and subscripts following each other directly.

```
\ch{^33B} \ch{{}^33B} \ch{3^3B} \ch{B^3} \ch{B3^3} \par
\ch{^{23}_{123}B} \ch{{}^{23}_{123}B} \ch{_{123}^{23}B}
\ch{B^{23}} \ch{B_{123}^{23}} \par
\ch{^{123}_{23}B} \ch{{}^{123}_{23}B} \ch{_{23}^{123}B}
\ch{B^{123}} \ch{B23^{123}}
```

${}^{3}_{3}\mathrm{B}$ ${}^{3}_{3}\mathrm{B}$ ${}^{3}_{3}\mathrm{B}$ B^{3} B^{3}_{3}
${}^{23}_{123}\mathrm{B}$ ${}^{23}_{123}\mathrm{B}$ ${}^{23}_{123}\mathrm{B}$ B^{23} B^{23}_{123}
${}^{123}_{23}\mathrm{B}$ ${}^{123}_{23}\mathrm{B}$ ${}^{123}_{23}\mathrm{B}$ B^{123} B^{123}_{23}

- If a compound *starts* with a sub- or superscript both sub- and superscript are aligned to the *right* else to the *left*.
- If a compound *does not start* with a sub- or superscript and there is both a sub- and a superscript, the superscript is shifted additionally by a length determined from the option `charge-hshift = {⟨dim⟩}`, also see page 13f.

The second point follows IUPAC's recommendations:

> In writing the formula for a complex ion, spacing for charge number can be added (staggered arrangement), as well as parentheses: $\mathrm{SO_4{}^{2-}}$, $\mathrm{(SO_4)^{2-}}$. The staggered arrangement is now recommended.
> *IUPAC Green Book [Coh+08, p. 51]*

6.5. Bonds

6.5.1. Native Bonds

There are three kinds of what I will call "native bonds":

```
single: \ch{CH3-CH3} \par
double: \ch{CH2=CH2} \par
triple: \ch{CH+CH}
```

single: $\mathrm{CH_3{-}CH_3}$
double: $\mathrm{CH_2{=}CH_2}$
triple: $\mathrm{CH{\equiv}CH}$

6.5.2. Flexible Bonds

Predefined Bonds In addition to the three native bonds there are a few more which can be called by

`\bond{⟨bond name⟩}`
Prints the bond type specified by ⟨*bond name*⟩.

The predefined bond types are shown in table 1 on the following page.

Table 1: Bonds available with \bond.

name	appearance	aliases
`single`	−	`normal`, `sb`
`double`	=	`db`
`triple`	≡	`tp`
`dotted`	⋯	`semisingle`
`deloc`	$\overset{\cdots}{-}$	`semidouble`
`tdeloc`	$\overset{\cdots}{=}$	`semitriple`
`co>`	→	`coordright`
`<co`	←	`coordleft`

```
\ch{C\bond{sb}C\bond{db}C\bond{tp}C\bond{deloc}C\bond{tdeloc}C\bond{co>}C\
  bond{<co}C}
```

C−C=C≡C$\overset{\cdots}{-}$C$\overset{\cdots}{=}$C→C←C

Own Bonds CHEMFORMULA offers commands to define own bond types:

`\NewChemBond{⟨name⟩}{⟨code⟩}`
Define the new bond type ⟨*name*⟩. Issue an error if a bond ⟨*name*⟩ already exists.

Introduced in version 4.3 (2014/01/24)

`\DeclareChemBond{⟨name⟩}{⟨code⟩}`
Define the new bond type ⟨*name*⟩ or overwrite it if it already exists.

`\RenewChemBond{⟨name⟩}{⟨code⟩}`
Redefine the existing bond type ⟨*name*⟩. Issue an error if a bond ⟨*name*⟩ doesn't exist.

`\ProvideChemBond{⟨name⟩}{⟨code⟩}`
Define the new bond type ⟨*name*⟩ only if it doesn't exist yet.

Introduced in version 4.12a (2015/08/29)

Introduced in version 4.3

`\NewChemBondAlias{⟨new name⟩}{⟨old name⟩}`
Declare the bond type ⟨*new name*⟩ to be an alias of ⟨*old name*⟩. Issue an error if a bond ⟨*new name*⟩ already exists.

`\DeclareChemBondAlias{⟨new name⟩}{⟨old name⟩}`
Declare the bond type ⟨*new name*⟩ to be an alias of ⟨*old name*⟩.

`\ShowChemBond{⟨name⟩}`
Print the definition of bond type ⟨*name*⟩.

The usage is best described with an example. So let's see how the `single` bond and the `co>` bond are defined:

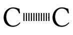
```
\NewChemBond{single}
  { \draw[chembond] (chemformula-bond-start) -- (chemformula-bond-end) ; }
\NewChemBond{coordright}
  {
    \draw[chembond,butt cap->]
      (chemformula-bond-start) -- (chemformula-bond-end) ;
  }
\NewChemBondAlias{co>}{coordright}
```

Two points are important:

- the names of the starting and the ending coordinates, `chemformula-bond-start` and `chemformula-bond-end`,
- and the Ti*k*Z style of the bonds `chembond`.

So, let's say you want to define a special kind of dashed bond. You could do this:

```
\usetikzlibrary{decorations.pathreplacing}
\makeatletter
\NewChemBond{dashed}
  {
    \draw[
      chembond,
      decorate,
      decoration={
        ticks,
        segment length=\chemformula@bondlength/10,amplitude=1.5pt
      }]
      (chemformula-bond-start) -- (chemformula-bond-end) ;
  }
\makeatother
\setchemformula{bond-length=2ex}
\ch{C\bond{dashed}C}
```



The last example showed you another macro: `\chemformula@bondlength`. It only exists so you can use it to access the bond length as set with `bond-length` directly.

6.6. Customization

These options allow you to customize the ouptut of the compounds:

`subscript-vshift` = {⟨*dim*⟩} Default: `0pt`

Extra vertical shift of the subscripts. This only works when `math-scripts` = `{false}` is in effect.

`subscript-style = text|math` Default: `text`

Style that is used to typeset the subscripts.

`charge-hshift = {⟨dim⟩}` Default: `.25em`

Shift of superscripts when following a subscript.

`charge-vshift = {⟨dim⟩}` Default: `0pt`

Extra vertical shift of the superscripts. This only works when `math-scripts = {false}` is in effect.

`charge-style = text|math` Default: `text`

Style that is used to typeset the superscripts.

`math-scripts = true|false` Default: `false`

Introduced in version 4.16 (2020/12/22)

Switches to TeX's native subscript and superscript mechanism which might be your option of choice for the sake of typographical consistency. *This option is experimental. Please report any problems you experience with this option to* CHEMFORMULA*'s bug tracker.*

`circled = formal|all|none` Default: `formal`

Introduced in version 4.6 (2014/06/30)

Like CHEMMACROS' package option but local to CHEMFORMULA's `\ch`. That is: since CHEMMACROS' macros use CHEMFORMULA's mechanism this is effectively an alias.

`circletype = chem|math` Default: `chem`

Introduced in version 4.6

Like CHEMMACROS' package option but local to CHEMFORMULA's `\ch`. That is: since CHEMMACROS' macros use CHEMFORMULA's mechanism this is effectively an alias.

`adduct-space = {⟨dim⟩}` Default: `.1333em`

Space to the left and the right of the adduct point.

`adduct-penalty = {⟨num⟩}` Default: `300`

Introduced in version 4.14 (2015/10/21)

The penalty inserted after the adduct point for (dis-)allowing line breaks.

`bond-length = {⟨dim⟩}` Default: `.5833em`

The length of the bonds.

`bond-offset = {⟨dim⟩}` Default: `.07em`

Space between bond and atoms.

`bond-style = {⟨TikZ⟩}` (initially empty)

TikZ options for the bonds.

`bond-penalty = {⟨num⟩}` Default: `10000`

Introduced in version 4.0a (2013/07/26)

The penalty that is inserted after a bond for (dis-)allowing line breaks.

`radical-style = {⟨TikZ⟩}` (initially empty)

TikZ options for the radical point.

`radical-radius = {⟨dim⟩}` Default: `.2ex`

The radius of the radical point.

`radical-hshift = {⟨dim⟩}` Default: `.15em`

Horizontal shift before the radical point is drawn.

`radical-vshift = {⟨dim⟩}` Default: `.5ex`

Vertical shift relative to the current baseline.

`radical-space = {⟨dim⟩}` Default: `.15em`

Horizontal shift after the radical point is drawn.

Maybe you have noticed that charges of certain ions are shifted to the right.

```
\ch{SO4^2-} \ch{NH4+} \ch{Na+}
```

SO_4^{2-} NH_4^+ Na^+

They are shifted if they *follow* a subscript which follows IUPAC recommendations [Coh+08, p. 51]. The amount of the shift can be set with the option `charge-hshift`.

```
\ch{SO4^2-} \ch{NH4+} \ch{Na+} \par
\setchemformula{charge-hshift=.5ex}
\ch{SO4^2-} \ch{NH4+} \ch{Na+} \par
\setchemformula{charge-hshift=.5pt}
\ch{SO4^2-} \ch{NH4+} \ch{Na+}
```

SO_4^{2-} NH_4^+ Na^+
SO_4^{2-} NH_4^+ Na^+
SO_4^{2-} NH_4^+ Na^+

Despite IUPAC's recommendation CHEMFORMULA does not make fully staggered arrangements in the default setting as I find it hard to read in some cases and ugly in others. Since this is a subjective decision CHEMFORMULA not only let's you define the absolute amount of the shift but also provides a possibility for full staggered arrangements. For this you have to use `charge-hshift = {full}`.

```
\ch[charge-hshift=0pt]{C5H11+} \ch[charge-hshift=0pt]{SO4^2-} \par
\ch{C5H11+} \ch{SO4^2-} \par
\ch[charge-hshift=1ex]{C5H11+} \ch[charge-hshift=1ex]{SO4^2-} \par
\ch[charge-hshift=full]{C5H11+} \ch[charge-hshift=full]{SO4^2-}
```

$C_5H_{11}^+$ SO_4^{2-}
$C_5H_{11}^+$ SO_4^{2-}
$C_5H_{11}^+$ SO_4^{2-}
$C_5H_{11}{}^+$ $SO_4{}^{2-}$

If you don't want the charges to be typeset in text mode you can switch to math mode:

```
\ch{M^x+} \ch{SO4^2-} \par
\setchemformula{charge-style = math}
\ch{M^x+} \ch{SO4^2-}
```

$\mathrm{M}^{\mathrm{x}+}\ \mathrm{SO}_4^{2-}$
$\mathrm{M}^{x+}\ \mathrm{SO}_4^{2-}$

The option `subscript-vshift` can be used to adjust the vertical shift of the subscripts:

```
\ch{H2SO4} \ch{Na3PO4} \par
\setchemformula{subscript-vshift=.5ex}
\ch{H2SO4} \ch{Na3PO4} \par
\setchemformula{subscript-vshift=-.2ex}
\ch{H2SO4} \ch{Na3PO4}
```

$\mathrm{H_2SO_4}\ \mathrm{Na_3PO_4}$
$\mathrm{H_2SO_4}\ \mathrm{Na_3PO_4}$
$\mathrm{H_2SO_4}\ \mathrm{Na_3PO_4}$

You can choose the mode subscripts are typeset in the same way as it is possible for the charges:

```
\ch{A_nB_m} \ch{H2SO4} \par
\setchemformula{subscript-style = math}
\ch{A_nB_m} \ch{H2SO4}
```

$\mathrm{A_nB_m}\ \mathrm{H_2SO_4}$
$\mathrm{A}_n\mathrm{B}_m\ \mathrm{H_2SO_4}$

The option `adduct-space` sets the space left and right to the adduct symbol ·.

```
\ch{Na3PO3*H2O} \par
\setchemformula{adduct-space=.2em}
\ch{Na3PO3*H2O}
```

$\mathrm{Na_3PO_3{\cdot}H_2O}$
$\mathrm{Na_3PO_3} \cdot \mathrm{H_2O}$

Changing the length of the bonds:

```
\setchemformula{bond-length=4mm}%
single: \ch{CH3-CH3} \par
double: \ch{CH2=CH2} \par
triple: \ch{CH+CH}
```

single: CH_3-CH_3
double: $CH_2=CH_2$
triple: $CH\equiv CH$

You can change the distance between bond and atom, too:

```
\ch{H-H + N+N + O=O} \par
\ch[bond-offset=1pt]{H-H + N+N + O=O}
```

$H-H + N\equiv N + O=O$
$H-H + N\equiv N + O=O$

6.7. Standalone Formulae

Introduced in version 4.0 (2013/07/06)

CHEMFORMULA offers a command that *only accepts* the “compound” input type:

\chcpd[⟨*options*⟩]{⟨*compound*⟩}
Typeset single compounds.

6.8. Extend Compound Properties

Introduced in version 4.10 (2015/03/16)

It is possible to extend the range of special input symbols within compounds. In the default setting those are `*.-=+'` and arabic numerals. Others can be added or the existing ones be changed with one of the following commmands:

\NewChemCompoundProperty{⟨*token*⟩}{⟨*replacement*⟩}
⟨*token*⟩ will be replaced by ⟨*replacement*⟩ within compounds. The property is only added if ⟨*token*⟩ is not yet part the compounds’ property list. Otherwise an error is issued.

\ProvideChemCompoundProperty{⟨*token*⟩}{⟨*replacement*⟩}

Introduced in version 4.12a

⟨*token*⟩ will be replaced by ⟨*replacement*⟩ within compounds. The property is only added if ⟨*token*⟩ is not yet part the compounds’ property list.

\RenewChemCompoundProperty{⟨*token*⟩}{⟨*replacement*⟩}
⟨*token*⟩ will be replaced by ⟨*replacement*⟩ within compounds. The property is only added if ⟨*token*⟩ is already part the compounds’ property list. Otherwise an error is issued.

\DeclareChemCompoundProperty{⟨*token*⟩}{⟨*replacement*⟩}
⟨*token*⟩ will be replaced by ⟨*replacement*⟩ within compounds. The property silently overwrites

any previously set ⟨*replacement*⟩ for ⟨*token*⟩ if ⟨*token*⟩ is already part the compounds' property list.

`\RemoveChemCompoundProperty{`⟨*token*⟩`}`
Removes ⟨*token*⟩ from the compounds' property list.

For example you can use

```
\NewChemCompoundProperty{\}{\slash}
```

to allow line breaks after slashes in compounds.

7. Special Input Types

There are some "special type" input groups.

7.1. Single Token Groups

The first kind are groups which consist of only one token. They are again divided into two groups, "addition symbols" and "symbols".

7.1.1. Addition Symbols

`\ch{ + }` +
Creates the plus sign between compounds with space around it:
`\ch{2 Na + Cl2}` $2\,\mathrm{Na} + \mathrm{Cl}_2$

`\ch{ - }` –
Introduced in version 4.3a ()
Creates the minus sign between compounds with space around it:
`\ch{M - H}` M – H

Addition symbols are surrounded with space which can be customized according to options explained in a bit. There is also some penalty prohibiting a line break after them which also can be customized with an option.

You can define/redefine your own addition symbols:

`\NewChemAdditionSymbol{`⟨*name*⟩`}{`⟨*input*⟩`}{`⟨*output*⟩`}`
Introduced in version 4.11 (2015/06/30)
Defines the addition symbol ⟨*name*⟩ with input symbol ⟨*input*⟩ and output ⟨*output*⟩.

`\ProvideChemAdditionSymbol{`⟨*name*⟩`}{`⟨*input*⟩`}{`⟨*output*⟩`}`
Introduced in version 4.12a
Defines the addition symbol ⟨*name*⟩ with input symbol ⟨*input*⟩ and output ⟨*output*⟩ only no addition symbol with then name ⟨*name*⟩ doesn't exist.

`\RenewChemAdditionSymbol{`⟨*name*⟩`}{`⟨*input*⟩`}{`⟨*output*⟩`}`
Introduced in version 4.11
Redefines the addition symbol ⟨*name*⟩ with input symbol ⟨*input*⟩ and output ⟨*output*⟩.

`\DeclareChemAdditionSymbol{⟨name⟩}{⟨input⟩}{⟨output⟩}`

Introduced in version 4.11

(Re-)Defines the addition symbol ⟨*name*⟩ with input symbol ⟨*input*⟩ and output ⟨*output*⟩ without checking if the symbol exists or not.

The space left and right of the plus and the minus sign and the signs themselves can be set with the following options:

`plus-space = {⟨skip⟩}` Default: `.3em plus .1em minus .1em`

A rubber length.

`plus-penalty = {⟨num⟩}` Default: `700`

Introduced in version 4.0a

The penalty that is inserted after the plus sign for (dis-)allowing line breaks.

`plus-output-symbol = {⟨code⟩}` Default: `+`

Introduced in version 4.9 (2014/10/07)

The ⟨*code*⟩ that is used for the plus sign.

`minus-space = {⟨skip⟩}` Default: `.3em plus .1em minus .1em`

Introduced in version 4.9

A rubber length.

`minus-penalty = {⟨num⟩}` Default: `700`

Introduced in version 4.9

The penalty that is inserted after the minus sign for (dis-)allowing line breaks.

`minus-output-symbol = {⟨code⟩}` Default: `$-$`

Introduced in version 4.9

The ⟨*code*⟩ that is used for the minus sign.

The corresponding three options are defined when `\NewChemAdditionSymbol` or one of the variants is used, ⟨*name*⟩`-space` and ⟨*name*⟩`-penalty` both with the same defaults as above, and ⟨*name*⟩`-output-symbol`.

```
\ch{A + B}\par
\ch[plus-space=4pt]{A + B}
```

A + B
A + B

7.1.2. Symbols

`\ch{ v }` ↓

Sign for precipitate: `\ch{BaSO4 v}` $BaSO_4$↓

`\ch{ ^ }` ↑

Sign for escaping gas[3]: `\ch{H2 ^}` H_2↑

You can define/redefine your own symbols:

`\NewChemSymbol{⟨input⟩}{⟨output⟩}`

Introduced in version 4.11

Defines the addition symbol with input ⟨*input*⟩ and output ⟨*output*⟩.

---

[3] Is this the correct English term? Please correct me if it isn’t.

`\ProvideChemSymbol{⟨input⟩}{⟨output⟩}`

Introduced in version 4.12a

Defines the addition symbol with input ⟨*input*⟩ and output ⟨*output*⟩ only if no symbol with input ⟨*input*⟩ exists.

`\RenewChemSymbol{⟨input⟩}{⟨output⟩}`

Introduced in version 4.11

Redefines the addition symbol with input ⟨*input*⟩ and output ⟨*output*⟩.

`\DeclareChemSymbol{⟨input⟩}{⟨output⟩}`

Introduced in version 4.11

(Re-)Defines the addition symbol with input ⟨*input*⟩ and output ⟨*output*⟩ without checking if the symbol exists or not.

7.2. Option Input

Sometimes you might want to apply an option only to a part of a, say, reaction. Of course you have the possibility to use `\ch` several times.

```
\ch{H2O +}\textcolor{red}{\ch{H2SO4}}\ch{-> H3O+ + HSO4-} \par
\ch{H2O +}\ch[subscript-vshift=2pt]{H2SO4}\ch{-> H3O+ + HSO4-}
```

$\mathrm{H_2O + H_2SO_4 \longrightarrow H_3O^+ + HSO_4^-}$

$\mathrm{H_2O + H2SO4 \longrightarrow H_3O^+ + HSO_4^-}$

This, however, interrupts the input in your source and *may* mess with the spacing. That's why there is an alternative:

`\ch{ @{⟨options⟩} }`

The options specified this way will be valid *only* until the next compound is set.

```
\ch{H2O +}\textcolor{red}{\ch{H2SO4}}\ch{-> H3O+ + HSO4-} \par
\ch{H2O + @{atom-format=\color{red}} H2SO4 -> H3O+ + HSO4-} \par
or of course:\par
\ch{H2O + \textcolor{red}{H2SO4} -> H3O+ + HSO4-}\par\bigskip
\ch{H2O +}\ch[subscript-vshift=2pt]{H2SO4}\ch{-> H3O+ + HSO4-} \par
\ch{H2O + @{subscript-vshift=2pt} H2SO4 -> H3O+ + HSO4-}
```

$\mathrm{H_2O + H_2SO_4 \longrightarrow H_3O^+ + HSO_4^-}$

$\mathrm{H_2O + H_2SO_4 \longrightarrow H_3O^+ + HSO_4^-}$

or of course:

$\mathrm{H_2O + H_2SO_4 \longrightarrow H_3O^+ + HSO_4^-}$

$\mathrm{H_2O + H2SO4 \longrightarrow H_3O^+ + HSO_4^-}$

$\mathrm{H_2O + H2SO4 \longrightarrow H_3O^+ + HSO_4^-}$

8. Escaped Input

In some cases it may be desirable to prevent CHEMFORMULA from parsing the input. This can be done in two ways.

8.1. Text

If you put something between `" "` or `' '` then the input will be treated as normal text, except that spaces are not allowed and have to be input with `~`.

`\ch{ "⟨escaped text⟩" }`
One of two possibilities to *escape* CHEMFORMULA's parsing.

`\ch{ '⟨escaped text⟩' }`
The second of two possibilities to *escape* CHEMFORMULA's parsing.

```
\ch{"\ox{2,Ca}" O} \par
\ch{"\ldots\," Na + "\ldots\," Cl2 -> "\ldots\," NaCl} \par
\ch{'A~->~B'}
```

Ca$^{\text{II}}$O
… Na + … $\mathrm{Cl_2} \longrightarrow$ … NaCl
A -> B

In many cases you won't need to escape the input. But when you get into trouble when using a command inside `\ch` try hiding it.

8.2. Math

If you especially want to input math you just enclose it with `$ $`. This output is different from the escaped text as it is followed by a space. The reasoning behind this is that I assume math will mostly be used to replace stoichiometric factors.

`\ch{ $⟨escaped math⟩$ }`
One of two possibilities to *escape* CHEMFORMULA's parsing into math mode.

`\ch{ \(⟨escaped math⟩\) }`
The second of two possibilities to *escape* CHEMFORMULA's parsing into math mode.

```
escaped text: \ch{"$x$" H2O} \par
escaped math: \ch{$x$ H2O} \par
also escaped math: \ch{\(x\) H2O} \par
\ch{$2n$ Na + $n$ Cl2 -> $2n$ NaCl}
```

escaped text: $x\mathrm{H_2O}$
escaped math: $x\ \mathrm{H_2O}$
also escaped math: $x\ \mathrm{H_2O}$
$2n\ \mathrm{Na} + n\ \mathrm{Cl_2} \longrightarrow 2n\ \mathrm{NaCl}$

The space that is inserted after a math group can be edited:

`math-space` = {⟨*skip*⟩} Default: `.1667em plus .0333em minus .0117em`

A rubber length.

```
\ch{$2n$ Na + $n$ Cl2 -> $2n$ NaCl} \par
\setchemformula{math-space=.25em}
\ch{$2n$ Na + $n$ Cl2 -> $2n$ NaCl} \par
\ch{$A->B$}
```

$2n\,\mathrm{Na} + n\,\mathrm{Cl}_2 \longrightarrow 2n\,\mathrm{NaCl}$

$2n\;\mathrm{Na} + n\;\mathrm{Cl}_2 \longrightarrow 2n\;\mathrm{NaCl}$

$A- > B$

9. Arrows

9.1. Arrow types

Arrows are input in the same intuitive way they are with mhchem. There are various different types:

`->` $\longrightarrow$
standard right arrow

`<-` $\longleftarrow$
standard left arrow

`-/>` $\not\longrightarrow$
does not react (right)

`</-` $\not\longleftarrow$
does not react (left)

`<->` $\longleftrightarrow$
resonance arrow

`<>` $\rightleftarrows$
reaction in both directions

`==` $=$
stoichiometric equation

`<=>` $\rightleftharpoons$
equilibrium arrow

`>=<` $\leftrightharpoons$
reversed equilibrium arrow

Introduced in version 4.5 (2014/04/07)

`<=>>`

unbalanced equilibrium arrow to the right

`>=<<`

Introduced in version 4.5

reversed unbalanced equilibrium arrow to the right

`<<=>`

unbalanced equilibrium arrow to the left

`>>=<`

Introduced in version 4.5

reversed unbalanced equilibrium arrow to the left

`<=o>`

Introduced in version 4.15 (2016/05/11)

quasi equilibrium arrow

`<=o>>`

Introduced in version 4.15

unbalanced quasi equilibrium arrow to the right

`<<=o>`

Introduced in version 4.15

unbalanced quasi equilibrium arrow to the left

`<o>`

isolobal arrow

`<==>`

Introduced in version 4.5

I've seen this one used. I'm not sure it actually has a meaning in chemical equations. If you have some official reference for this arrow type please feel free to contact me.

All these arrows are drawn with Ti*k*Z.

```
\ch{H2 + Cl2 -> 2 HCl} \par
\ch{H2O + CO3^2- <=> OH- + HCO3-} \par
\ch{A <- B} \par
\ch{\{[CH2=CH-CH2]- <-> {}[CH2-CH=CH2]- \}} \par
\ch{A <> B} \par
\ch{H+ + OH- <=>> H2O} \par
\ch{2 NO2 <<=> N2O4}
```

$H_2 + Cl_2 \longrightarrow 2\,HCl$

$H_2O + CO_3^{2-} \rightleftharpoons OH^- + HCO_3^-$

$A \longleftarrow B$

$\{[CH_2{=}CH{-}CH_2]^- \longleftrightarrow [CH_2{-}CH{=}CH_2]^-\}$

$A \rightleftarrows B$

$H^+ + OH^- \rightleftharpoons H_2O$

$2\,NO_2 \rightleftharpoons N_2O_4$

9.2. Labels

The arrows take two optional arguments to label them.

`->[⟨above⟩][⟨below⟩]`
Add text above or under an arrow.

```
\ch{A ->[a] B} \par
\ch{A ->[a][b] B} \par
\ch{A ->[\SI{100}{\celsius}] B}
```

$$A \xrightarrow{\text{a}} B$$
$$A \xrightarrow[\text{b}]{\text{a}} B$$
$$A \xrightarrow{100\,^{\circ}\mathrm{C}} B$$

The label text can be parsed seperately from the arrow. The recipe is easy: leave blanks.

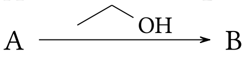
```
\ch{A ->[H2O] B} \par
\ch{A ->[ H2O ] B} \par
\ch{A ->[ "\ox{2,Ca}" F2 ] B} \par
\ch{A ->[ $\Delta$,~ [ H+ ]] B}
```

$$A \xrightarrow{\text{H2O}} B$$
$$A \xrightarrow{H_2O} B$$
$$A \xrightarrow{\overset{\mathrm{II}}{\mathrm{Ca}}F_2} B$$
$$A \xrightarrow{\Delta,\ [H^+]} B$$

If you leave the blanks CHEMFORMULA treats the groups inside the square brackets as seperated input types. The arrow reads its arguments *afterwards*. As you can see the arrows "grow" with the length of the labels. What stays constant is the part that protrudes the labels.

```
\ch{A ->[a] B} \par
\ch{A ->[ab] B} \par
\ch{A ->[abc] B} \par
\ch{A ->[abc~abc] B} \par
% needs the `chemfig' package:
\setchemfig{atom sep =15pt}
\ch{A ->[ "\chemfig{-[:30]-[:-30]OH}" ] B} \par
```

$$A \xrightarrow{\text{a}} B$$
$$A \xrightarrow{\text{ab}} B$$
$$A \xrightarrow{\text{abc}} B$$
$$A \xrightarrow{\text{abc abc}} B$$



☆ New

The labels are typeset inside of a `tabular{c}` environment. This means you can get several lines:

```
\ch{A ->[{a\\b}] B} \par
\ch{A ->[a \newline b] B}
```

$$A \xrightarrow{\substack{a\\b}} B$$

$$A \xrightarrow{\substack{a\\b}} B$$

The command `\\` must be hidden in order to work. `\tabularnewline` does work, of course. For convenience CHEMFORMULA also allows `\newline` here.

9.3. Customization

These are the options which enable you to customize the arrows:

`arrow-offset = {⟨dim⟩}` Default: `.75em`

This is the length that an arrow protrudes a label on both sides. This means an empty arrow's length is two times `arrow-offset`.

`arrow-min-length = {⟨dim⟩}` Default: `0pt`

Introduced in version 3.6b (2013/04/19)

The minimal length an error must have unless two times `arrow-offset` plus the width of the label is larger.

`arrow-yshift = {⟨dim⟩}` Default: `0pt`

Shifts an arrow up (positive value) or down (negative value).

`arrow-ratio = {⟨<factor>⟩}` Default: `.6`

The ratio of the arrow lengths of the unbalanced equilibrium. `.4` would mean that the length of the shorter arrow is 0.4× the length of the longer arrow.

`compound-sep = {⟨dim⟩}` Default: `.5em`

The space between compounds and the arrows.

`label-offset = {⟨dim⟩}` Default: `2pt`

The space between the labels and the arrows.

`label-style = {⟨font command⟩}` Default: `\footnotesize`

The relative font size of the labels.

`arrow-penalty = {⟨num⟩}` Default: `0`

Introduced in version 4.0a

The penalty that is inserted after an arrow for (dis-)allowing line breaks.

`arrow-style = {⟨TikZ⟩}` (initially empty)

Introduced in version 4.1a ()

Additonal Ti*k*Z keys for formatting the arrows.

The following code shows the effect of the different options on the <=>> arrow:

```
standard: \ch{A <=>>[x][y] B} \par
longer: \ch[arrow-offset=12pt]{A <=>>[x][y] B} \par
```

```
higher: \ch[arrow-yshift=2pt]{A <=>>[x][y] B} \par
more balanced: \ch[arrow-ratio=.8]{A <=>>[x][y] B} \par
labels further away: \ch[label-offset=4pt]{A <=>>[x][y] B} \par
larger distance to compounds: \ch[compound-sep=2ex]{A <=>>[x][y] B} \par
smaller labels: \ch[label-style=\tiny]{A <=>[x][y] B}
```

standard: A $\xrightleftharpoons[y]{x}$ B

longer: A $\xrightleftharpoons[y]{x}$ B

higher: A $\xrightleftharpoons[y]{x}$ B

more balanced: A $\xrightleftharpoons[y]{x}$ B

labels further away: A $\xrightleftharpoons[y]{x}$ B

larger distance to compounds: A $\xrightleftharpoons[y]{x}$ B

smaller labels: A $\xrightleftharpoons[y]{x}$ B

Introduced in version 4.7 (2014/08/04)

If you want to have different arrow tips there is an easy way to use existing arrow tips (as defined by TikZ). CHEMFORMULA uses three different arrow tips: `cf`, `left cf` and `right cf`. If you want them to match those of chemfig [Tel19] for example you could do:

```
\pgfkeys{
  cf /.tip = {CF@full} ,
  left cf /.tip = {CF@half}
}
```

chemfig has no equivalent of `right cf`. This mechanism relies on TikZ version 3.0.0 and the new `arrows.meta` library.

9.4. Modify Arrow Types

The arrows are defined with the commands

`\NewChemArrow{⟨type⟩}{⟨TikZ⟩}`

Define the new arrow type ⟨*type*⟩. Issue an error if an arrow type ⟨*type*⟩ already exists.

`\ProvideChemArrow{⟨type⟩}{⟨TikZ⟩}`

Introduced in version 4.12a

Define the new arrow type ⟨*type*⟩ only if it doesn't exist, yet.

`\DeclareChemArrow{⟨type⟩}{⟨TikZ⟩}`
Define the new arrow type ⟨*type*⟩ or overwrite it if it already exists.

`\RenewChemArrow{⟨type⟩}{⟨TikZ⟩}`
Redefine the arrow type ⟨*type*⟩. Issue an error if an arrow type ⟨*type*⟩ doesn't exist.

`\ShowChemArrow{⟨type⟩}`
Print out the current definition of the arrow type ⟨*type*⟩.

⟨*type*⟩ is the sequence of tokens that is replaced with the actual arrow code. For example the basic arrow is defined via

```
\NewChemArrow{->}{
  \draw[chemarrow,-cf] (cf_arrow_start) -- (cf_arrow_end) ;
}
```

In order to define arrows yourself you need to know the basics of Ti*k*Z.[4] The predefined arrows use the arrow tips `cf`, `left cf` and `right cf`. They also all except the net reaction arrow == use the Ti*k*Z-style `chemarrow` that you should use, too, if you want the option `arrow-style` to have an effect.

There are some predefined coordinates you can and should use. For completeness' sake the arrow tips and the Ti*k*Z-style are also listed:

`(cf_arrow_start)`
The beginning of the arrow.

`(cf_arrow_end)`
The end of the arrow.

`(cf_arrow_mid)`
The mid of the arrow.

`(cf_arrow_mid_start)`
The beginning of the shorter arrow in types like <=>>.

`(cf_arrow_mid_end)`
The end of the shorter arrow in types like <=>>.

`cf`
A double-sided arrow tip.

`left cf`
A left-sided arrow tip.

`right cf`
A right-sided arrow tip.

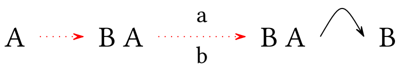
---

[4] Please see the pgfmanual for details.

`chemarrow`

CHEMFORMULA's Ti*k*Z-style that is applied to the arrows and set with `arrow-style`

```
\NewChemArrow{.>}{
  \draw[chemarrow,-cf,dotted,red] (cf_arrow_start) -- (cf_arrow_end);
}
\NewChemArrow{n>}{
  \draw[chemarrow,-cf]
    (cf_arrow_start)
      .. controls ([yshift=3ex]cf_arrow_mid) ..
    (cf_arrow_end);
}
\ch{A .> B} \ch{A .>[a][b] B} \ch{A n> B}
```



```
\texttt{\ShowChemArrow{->}} \par
\RenewChemArrow{->}{\draw[chemarrow,->,red] (cf_arrow_start) -- (cf_arrow_end
  ) ;}
\texttt{\ShowChemArrow{->}} \par
\ch{A -> B}
```

```
\draw [chemarrow,-cf](cf_arrow_start)--(cf_arrow_end);
\draw [chemarrow,->,red] (cf_arrow_start) -- (cf_arrow_end) ;
```

$A \longrightarrow B$

9.5. Standalone Arrows

Introduced in version 4.0

CHEMFORMULA offers a command that *only accepts* the "arrow" input type:

`\charrow{⟨type⟩}[⟨above⟩][⟨below⟩]`

Print the arrow type ⟨*type*⟩.

This command is internally used for the arrows, too, when `\ch` is parsed.

10. Names

10.1. Syntax

CHEMFORMULA has a built-in syntax to write text under a compound. In a way it works very similar to the arrows.

`\ch{ !(⟨text⟩)( ⟨formula⟩ ) }`
Writes ⟨*text*⟩ below ⟨*formula*⟩.

If an exclamation mark is followed by a pair of parentheses CHEMFORMULA will parse it this way:

```
\ch{!(ethanol)( CH3CH2OH )}
```

$\underset{\text{ethanol}}{\mathrm{CH_3CH_2OH}}$

The same what's true for the arrows arguments holds for these arguments: if you leave blanks the different parts will be treated according to their input type before the text is set below the formula.

```
\ch{!(water)(H2O)} \quad
\ch{!( "\textcolor{blue}{water}" )( H2O )} \quad
\ch{!( $2n-1$ )( H2O )} \quad
\ch{!( H2O )( H2O )} \quad
\ch{!(oxonium)( H3O+ )}
```

$\underset{\text{water}}{\mathrm{H2O}} \quad \underset{\text{water}}{\mathrm{H_2O}} \quad \underset{2n-1}{\mathrm{H_2O}} \quad \underset{\mathrm{H_2O}}{\mathrm{H_2O}} \quad \underset{\text{oxonium}}{\mathrm{H_3O^+}}$

If for some reason you want to insert an exclamation mark *without* it creating a name you only have to make sure it isn't followed by parentheses.

```
\ch{H2O~(!)} \par
\ch{A!{}()}
```

H2O (!)

A!()

10.2. Customization

CHEMFORMULA provides two options to customize the output of the names:

`name-format = {⟨commands⟩}` Default: `\scriptsize\centering`

Changed in version 4.15d (2017/02/02)

The format of the name. This can be arbitrary input and the last macro may receive the name as an argument.

`name-width = ⟨dim⟩|auto` Default: `auto`

The width of the box where the label is put into. `auto` will detect the width of the name and set the box to this width.

```
\ch{!(acid)( H2SO4 ) -> B} \par
\ch[name-format=\sffamily\small]{!(acid)( H2SO4 ) -> B} \par
\ch[name-format=\scriptsize N:~]{!(acid)( H2SO4 ) -> B} \par
\ch[name-width=3em,name-format=\scriptsize\raggedright]{!(acid)( H2SO4 ) -> B
 }
```

$\underset{\text{acid}}{H_2SO_4} \longrightarrow B$

$\underset{\textsf{acid}}{H_2SO_4} \longrightarrow B$

$\underset{\text{N: acid}}{H_2SO_4} \longrightarrow B$

$\underset{\text{acid}}{H_2SO_4} \longrightarrow B$

10.3. Standalone Names

Introduced in version 4.0

CHEMFORMULA offers a command that allows the usage of the “name” syntax in normal text. This is the command that a bang is replaced with in CHEMFORMULA’s formulas, actually. Both arguments are mandatory.

`\chname(⟨text 1⟩)(⟨text 2⟩)`
The command that is useed internally for placing ⟨*text 1*⟩ below of ⟨*text 2*⟩.

11. Format and Font

In the standard setting CHEMFORMULA doesn’t make any default changes to the font of the formula output. Let’s take a look at a nonsense input which shows all features:

```
\newcommand*\sample{%
  \ch{H2C-C+C-CH=CH+ + CrO4^2-
      <=>[x][y]
    2.5 Cl^{-.} + 3_1/2 Na*OH_{(aq)} + !(name)( A^n ) "\LaTeXe"}
}
\sample
```

$H_2C{-}C{\equiv}C{-}CH{=}CH^{+} + CrO_4^{2-} \xrightleftharpoons[y]{x} 2.5\,Cl^{-\bullet} + 3\tfrac{1}{2}\,Na\cdot OH_{(aq)} + \underset{\text{name}}{A^{n}}\ \text{\LaTeX}\,2_{\varepsilon}$

Now we’re going to change different aspects of the font a look what happens:

```
\sffamily Hallo \sample \\
\ttfamily Hallo \sample \normalfont \\
\bfseries Hallo \sample \normalfont \\
\itshape Hallo \sample
```

Hallo $H_2C{-}C{\equiv}C{-}CH{=}CH^+ + CrO_4^{2-} \xrightleftharpoons[y]{x} 2.5\,Cl^{-\bullet} + 3\tfrac{1}{2}\,Na\cdot OH_{(aq)} + \underset{name}{A^n}$ LaTeX 2_ε

Hallo $H_2C{-}C{\equiv}C{-}CH{=}CH^+ + CrO_4^{2-} \xrightleftharpoons[y]{x} 2.5\,Cl^{-\bullet} + 3\tfrac{1}{2}\,Na\cdot OH_{(aq)} + \underset{name}{A^n}$ LaTeX 2_ε

Hallo $\mathbf{H_2C{-}C{\equiv}C{-}CH{=}CH^+ + CrO_4^{2-} \xrightleftharpoons[y]{x} 2.5\,Cl^{-\bullet} + 3\tfrac{1}{2}\,Na\cdot OH_{(aq)} + \underset{name}{A^n}}$ **LaTeX** $\mathbf{2_\varepsilon}$

Hallo $H_2C{-}C{\equiv}C{-}CH{=}CH^+ + CrO_4^{2-} \xrightleftharpoons[y]{x} 2.5\,Cl^{-\bullet} + 3\tfrac{1}{2}\,Na\cdot OH_{(aq)} + \underset{name}{A^n}$ *LaTeX* 2_ε

As you can see most features adapt to the surrounding font.
If you want to change the default format you need to use this option:

`format` = {⟨*code*⟩} (initially empty)

Adds ⟨*code*⟩ before the output of `\ch`.

`atom-format` = {⟨*code*⟩} (initially empty)

Introduced in version 4.13 (2015/01/01)

This adds ⟨*code*⟩ before each formula. This allows to specify a format for the chemical formulas only and have a different format for the rest of the chemical equation.

```
\definecolor{newblue}{rgb}{.1,.1,.5}
\setchemformula{format=\color{newblue}\sffamily}
\sffamily Hallo \sample \\
\ttfamily Hallo \sample \normalfont \\
\bfseries Hallo \sample \normalfont \\
\itshape Hallo \sample
```

Hallo $H_2C{-}C{\equiv}C{-}CH{=}CH^+ + CrO_4^{2-} \xrightleftharpoons[y]{x} 2.5\,Cl^{-\bullet} + 3\tfrac{1}{2}\,Na\cdot OH_{(aq)} + \underset{name}{A^n}$ LaTeX 2_ε

Hallo $H_2C{-}C{\equiv}C{-}CH{=}CH^+ + CrO_4^{2-} \xrightleftharpoons[y]{x} 2.5\,Cl^{-\bullet} + 3\tfrac{1}{2}\,Na\cdot OH_{(aq)} + \underset{name}{A^n}$ LaTeX 2_ε

Hallo $\mathbf{H_2C{-}C{\equiv}C{-}CH{=}CH^+ + CrO_4^{2-} \xrightleftharpoons[y]{x} 2.5\,Cl^{-\bullet} + 3\tfrac{1}{2}\,Na\cdot OH_{(aq)} + \underset{name}{A^n}}$ **LaTeX** $\mathbf{2_\varepsilon}$

Hallo $H_2C{-}C{\equiv}C{-}CH{=}CH^{+} + CrO_4^{2-} \xrightleftharpoons[y]{x} 2.5\,Cl^{-\bullet} + 3\tfrac{1}{2}\,Na\cdot OH_{(aq)} + \underset{\text{name}}{A^{n}\ \text{\LaTeX}\,2_{\varepsilon}}$

You can also specifically change the fontfamily, fontseries and fontshape of the output.

`font-family = {⟨family⟩}` (initially empty)
Changes the fontfamily of the output with `\fontfamily{⟨family⟩}`.

`font-series = {⟨series⟩}` (initially empty)
Changes the fontseries of the output with `\fontseries{⟨series⟩}`.

`font-shape = {⟨shape⟩}` (initially empty)
Changes the fontshape of the output with `\fontshape{⟨shape⟩}`.

```
\setchemformula{font-series=bx}
Hallo \sample \par
\sffamily Hallo \sample \normalfont \par
\setchemformula{font-family=lmss,font-series=m} Hallo \sample
  \normalfont \par
\itshape Hallo \sample
```

Hallo $\mathbf{H_2C{-}C{\equiv}C{-}CH{=}CH^{+} + CrO_4^{2-} \xrightleftharpoons[y]{x} 2.5\,Cl^{-\bullet} + 3\tfrac{1}{2}\,Na\cdot OH_{(aq)} +}$ $\underset{\textbf{name}}{\mathbf{A^{n}\ \text{\LaTeX}\,2_{\varepsilon}}}$

Hallo $\mathbf{H_2C{-}C{\equiv}C{-}CH{=}CH^{+} + CrO_4^{2-} \xrightleftharpoons[y]{x} 2.5\,Cl^{-\bullet} + 3\tfrac{1}{2}\,Na\cdot OH_{(aq)} +}$ $\underset{\textbf{name}}{\mathbf{A^{n}\ \text{\LaTeX}\,2_{\varepsilon}}}$

Hallo $H_2C{-}C{\equiv}C{-}CH{=}CH^{+} + CrO_4^{2-} \xrightleftharpoons[y]{x} 2.5\,Cl^{-\bullet} + 3\tfrac{1}{2}\,Na\cdot OH_{(aq)} +$ $\underset{\text{name}}{A^{n}\ \text{\LaTeX}\,2_{\varepsilon}}$

Hallo $H_2C{-}C{\equiv}C{-}CH{=}CH^{+} + CrO_4^{2-} \xrightleftharpoons[y]{x} 2.5\,Cl^{-\bullet} + 3\tfrac{1}{2}\,Na\cdot OH_{(aq)} +$ $\underset{\text{name}}{A^{n}\ \text{\LaTeX}\,2_{\varepsilon}}$

If you're using XƎLaTeX or LuaLaTeX and have loaded fontspec you have the possibilty to set the font with it:

`font-spec = {⟨font⟩}` (initially empty)
Use font ⟨*font*⟩ for CHEMFORMULA's formulas.

or with options

`font-spec` = `{[`⟨*options*⟩`]`⟨*font*⟩`}`

Use font ⟨*font*⟩ with options ⟨*options*⟩ for CHEMFORMULA’s formulas.

Since this document is typeset with pdfLaTeX the option cannot be demonstrated here.

12. Usage In Math Equations

The `\ch` command can be used inside math equations. It recognizes `\\` and `&` and passes them on. However, you can’t use the optional arguments of `\\` inside `\ch`.

```
\begin{align}
 \ch{
   H2O & ->[a] H2SO4 \\
   Cl2 & ->[x][y] CH4
 }
\end{align}
\begin{align*}
\ch{
  RNO2      &<=>[ + e- ] RNO2^{-.} \\
  RNO2^{-.} &<=>[ + e- ] RNO2^2-
}
\end{align*}
```

$$\mathrm{H_2O} \xrightarrow{\mathrm{a}} \mathrm{H_2SO_4} \tag{1}$$

$$\mathrm{Cl_2} \xrightarrow[\mathrm{y}]{\mathrm{x}} \mathrm{CH_4} \tag{2}$$

$$\begin{aligned} \mathrm{RNO_2} &\xrightleftharpoons{+\ \mathrm{e}^-} \mathrm{RNO_2}^{-\bullet} \\ \mathrm{RNO_2}^{-\bullet} &\xrightleftharpoons{+\ \mathrm{e}^-} \mathrm{RNO_2}^{2-} \end{aligned}$$

13. Usage with Ti*k*Z or pgfplots and externalization

Introduced in version 4.1 (2013/08/24)

Since CHEMFORMULA uses Ti*k*Z to draw reaction arrows and bonds they would be externalized, too, if you use that facility with Ti*k*Z or pgfplots[5] [Feu18]. This may not be desirable since they are very small pictures maybe containing of a single line. This is why CHEMFORMULA’s default behaviour is to disable externalization for it’s bonds and arrows. This can be turned on and off through the following option:

---

[5] on CTAN as `pgfplots`: `http://mirrors.ctan.org/macros/latex/contrib/pgfplots/`

`tikz-external-disable` = `true|false` Default: `true`
dis- or enable Ti*k*Z' externalization mechanism for CHEMFORMULA's arrows and bonds.

If you should be using a formula that contains bonds or arrows inside of a `tikzpicture` that is externalized you should locally enable it for CHEMFORMULA, too:

```
\begin{tikzpicture}
  \setchemformula{tikz-external-disable=false}
  \begin{axis}[xlabel={\ch{2 H+ + 2 e- -> H2}}]
    \addplot ... ;
  \end{axis}
\end{tikzpicture}
```

14. Lewis Formulae

Introduced in version 4.2 (2013/10/24)

CHEMFORMULA offers a command to typeset Lewis formulae. This does not mean Lewis structures! Those can be achieved using the chemfig package [Tel19]. CHEMFORMULA provides the possibility to draw electrons as dots and pairs of dots or a line around an atom.

`\chlewis[`⟨*options*⟩`]{`⟨*electron spec*⟩`}{`⟨*atom*⟩`}`
Draws electrons around the ⟨*atom*⟩ according to ⟨*electron spec*⟩.

Electrons are specified by the angle to the horizontal in the couter-clockwise direction. The default appearance is a pair of electrons drawn as a pair of dots. Other specifications can be chosen. The specification follows the pattern ⟨*angle*⟩⟨*separator*⟩. ⟨*angle*⟩ is a positiv or negativ integer denoting the angle counter clockwise to the horizontal where the electrons should be drawn. ⟨*separator*⟩ is either a dot (`.`, single electron), a colon (`:`, electron pair), a vertical line (`|`, electron pair), an o (`o`, empty pair), or a comma (`,` default spec).

`\chlewis{`⟨*angle1*⟩⟨*type1*⟩⟨*angle2*⟩⟨*type2*⟩`}{`⟨*atom*⟩`}`
For example: `\chlewis{0,180}{O}` gives $:\mathrm{O}:$ and `\chlewis{0.90.180.270.}{C}` gives $\cdot\dot{\underset{\cdot}{\mathrm{C}}}\cdot$.

The appearance can be influenced by a number of options:

`lewis-default` = `.|:|||o|single|pair|pair (dotted)|pair (line)|empty` Default: `pair`
Sets the default type that is used when no type is given in ⟨*electron spec*⟩.

`lewis-distance` = `{`⟨*dim*⟩`}` Default: `1ex`
The distance of two electrons in a pair.

`lewis-line-length` = `{`⟨*dim*⟩`}` Default: `1.5ex`
The length of the line representing an electron pair.

`lewis-line-width` = `{`⟨*dim*⟩`}` Default: `1pt`
The thickness of a line representing an electron pair.

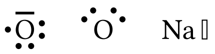
`lewis-offset = {⟨dim⟩}` Default: `.5ex`

The distance of the symbols from the atom.

The dots are drawn according to the `radical-radius` option mentioned in section 6.6.

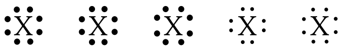
The basic usage should be more or less self-explaining:

```
\chlewis{0:90|180.270}{O}
\quad
\chlewis{45,135}{O}
\quad
\chlewis{0o}{Na}
```

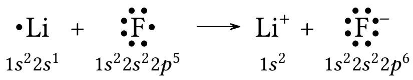


The next example shows the effect of some of the options:

```
\chlewis[lewis-default=.]{23,68,113,158,203,248,293,338}{X}
\quad
\chlewis{0,90,180,270}{X}
\quad
\chlewis[lewis-distance=1.25ex]{0,90,180,270}{X}
\quad
\chlewis[lewis-distance=.75ex,radical-radius=.5pt]{0,90,180,270}{X}
\quad
\chlewis[
  radical-radius=.5pt,
  lewis-default=.
]{23,68,113,158,203,248,293,338}{X}
```



```
\ch{
  !($1s^22s^1$)( "\chlewis{180.}{Li}" ) +
  !($1s^22s^22p^5$)( "\chlewis{0.90,180,270}{F}" )
   ->
  !($1s^2$)( Li+ ) + !($1s^22s^22p^6$)( "\chlewis{0,90,180,270}{F}" {}- )
}
```



15. Kröger-Vink Notation

Introduced in version 4.5

CHEMFORMULA also supports the Kröger-Vink notation.

`kroeger-vink` = `true|false` Default: `false`

Enable the Kröger-Vink notation. As most options this can be enabled globally via the setup command or locally as option to `\ch`.

With this option enabled several changes come into effect: `'` produces a prime, a `x` in a superscript produces $\times$, and both a `.` and a `*` produce a little filled circle. In the Kröger-Vink notation a prime denotes a negative relative charge, the circle a positive relative charge, and the cross denotes a neutral relative charge.

```
\setchemformula{kroeger-vink=true}
\ch{Al_{Al}^'}
\ch{Al_{Al}'}\par
\ch{Ni_{Cu}^{x}}\par
\ch{V_{Cl}^.}
\ch{V_{Cl}^*}\par
\ch{Ca_i^{..}}\par
\ch{e^'}\par
\ch{Cl_i^'}
\ch{Cl_i'}\par
\ch{O_i^{''}}
\ch{O_i''}
```

$\mathrm{Al}_{\mathrm{Al}}^{\prime}$ $\mathrm{Al}_{\mathrm{Al}}^{\prime}$

$\mathrm{Ni}_{\mathrm{Cu}}^{\times}$

$\mathrm{V}_{\mathrm{Cl}}^{\bullet}$ $\mathrm{V}_{\mathrm{Cl}}^{\bullet}$

$\mathrm{Ca}_{\mathrm{i}}^{\bullet\bullet}$

$\mathrm{e}^{\prime}$

$\mathrm{Cl}_{\mathrm{i}}^{\prime}$ $\mathrm{Cl}_{\mathrm{i}}^{\prime}$

$\mathrm{O}_{\mathrm{i}}^{\prime\prime}$ $\mathrm{O}_{\mathrm{i}}^{\prime\prime}$

There are a number of options for customizations:

`kv-positive-style` = `{`⟨*TikZ*⟩`}` (initially empty)

Ti*k*Z code for positive charge dot.

`kv-positive-radius` = `{`⟨*dim*⟩`}` Default: `.3ex`

Radius of positive charge dot

`kv-positive-hshift` = `{`⟨*dim*⟩`}` Default: `.15em`

Horizontal shift of positive charge dot

`kv-positive-vshift` = `{`⟨*dim*⟩`}` Default: `.5ex`

Vertical shift positive charge dot

`kv-positive-offset` = `{`⟨*dim*⟩`}` Default: `.4em`

The offset of two consecutive positive charge dots

`kv-neutral-symbol` = `{`⟨*TeX code*⟩`}` Default: `$\times$`

Symbol for neutral particles.

A. History Since Version 4.0

Version 4.0 (2013/07/06)

Introduced in version 4.0

- Since version 4.0, the CHEMFORMULA package is distributed independently from CHEMMACROS.

Version 4.1 (2013/08/24)

- New option `tikz-external-disable`.
- New option `frac-math-cmd`.

Version 4.2 (2013/10/24)

- New option `arrow-style`.
- New command `\chlewis` that allows to add Lewis electrons to an atom, see section 14.

Version 4.3 (2014/01/24)

- New option `stoich-print`.
- New command `\chstoich`.
- The commands `\DeclareChem`⟨...⟩ now don't give an error any more if the command already exists. This is more consistent with LaTeX's `\DeclareRobustCommand`. For all those commands a version `\NewChem`⟨...⟩ is introduced that *does* give an error if the new command is already defined.

Version 4.4 (2014/01/29)

- A single dash `-` in `\ch` is now treated as a minus sign. This is consistent with the behaviour of a `+`.

Version 4.5 (2014/04/07)

- New arrow types >=<, >=<<, >>=< and <==>.
- Internal changes to `\ch` allow usage of optional arguments of `\\` and `\label` in CHEMMACROS' `reactions` environment.

Version 4.6 (2014/06/30)

- New options `circled` and `circletype`. this allows to set the behaviour as described on CHEMMACROS' manual for a specific usage of `\ch`.

Version 4.7 (2014/08/04)

- Dependency change: CHEMFORMULA now requires the Ti*k*Z library `arrows.meta` instead of the library `arrows`. This requires Ti*k*Z version 3.0.0.

Version 4.8 (2014/08/08)

- The CHEMFORMULA package now is no longer part of the CHEMMACROS bundle but is distributed as a package of it's own.

Version 4.9 (2014/10/07)

- New options `minus-space` and `minus-penalty` equivalent to the existing `plus-space` and `plus-penalty`
- New options `plus-output-symbol` and `minus-output-symbol` for customizing the plus and minus signs in the output.

Version 4.10 (2015/03/16)

- New macro set `\NewChemCompoundProperty`, see section 6.8 for a description.

Version 4.11 (2015/06/30)

- New macro set `\NewChemAdditionSymbol`, see section 7.1.1.
- New macro set `\NewChemSymbol`, see section 7.1.2.

Version 4.12 (2015/08/12)

- Change package requirement: CHEMFORMULA now not loads complete amsmath but only amstext.

Version 4.13 (2015/01/01)

- Check for blank input parts and don't process them.
- Drop support for `\[` and `\]` as replacement for `[` and `]` inside arrow captions.
- New option `atom-format`.

Version 4.14 (2015/10/21)

- New option `adduct-penalty`.

Version 4.15 (2016/05/11)

- The order of arrow definitions doesn't matter any more.
- New quasi equilibria arrows.
- Require amsmath.
- Allow `name-format` to end with a macro that takes an argument.
- Various bug fixes.

Version 4.16 (2020/12/22)

- New option `math-scripts`
- Use LaTeX's new hooks and get rid of redundant package dependencies and code.

Version 4.17 (2022/01/23)

- Various bug fixes.
- Lazy boolean evaluation where possible.
- Place arrow's arguments in a tabular. This allows for multiple lines.

B. References

[Coh+08] E. Richard COHAN et al.
"Quantities, Symbols and Units in Physical Chemistry", IUPAC Green Book.
3rd Edition. 2nd Printing. IUPAC & RSC Publishing, Cambridge, 2008.

[Feu18] Christian FEUERSÄNGER. pgfplots. version 1.16, Mar. 28, 2018 (or newer).
URL: `https://www.ctan.org/pkg/pgfplots/`.

[Koh19] Markus KOHM. KOMA-Script. version 3.25, Jan. 14, 2019 (or newer).
URL: `https://www.ctan.org/pkg/koma-script/`.

[L3Pa] THE LATEX3 PROJECT TEAM. l3kernel. Oct. 27, 2020 (or newer).
URL: `https://www.ctan.org/pkg/l3kernel/`.

[L3Pb] THE LATEX3 PROJECT TEAM. l3packages. Oct. 27, 2020 (or newer).
URL: `https://www.ctan.org/pkg/l3packages/`.

[MS00] Frank MITTELBACH and Rainer SCHÖPF.
amstext. version 2.01, June 29, 2000 (or newer).
URL: `https://www.ctan.org/pkg/amstext/`.

[Rei98] Axel REICHERT. nicefrac. version 0.9b, Aug. 4, 1998 (or newer).
URL: `https://www.ctan.org/pkg/units/`.

[Tan19] Till TANTAU. TikZ/pgf. version 3.1.4b, Aug. 3, 2019 (or newer).
URL: `https://www.ctan.org/pkg/pgf/`.

[Tel19] Christian TELLECHEA. chemfig. version 1.41, May 21, 2019 (or newer).
URL: `https://www.ctan.org/pkg/chemfig/`.

C. Index