CHEMFORMULA

v4.15a 2016/06/08

typeset chemical compounds and reactions

Clemens NIEDERBERGER

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

contact@mychemistry.eu

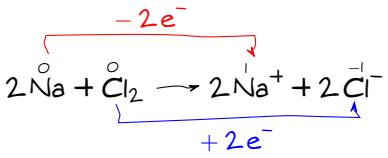

Table of Contents

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 1.3 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 [**bnd:l3kernel**], xparse, l3keys2e and xfrac (all three are part of the l3packages bundle [**bnd:l3packages**]), tikz[1] [**pkg:pgf**], amstext [**pkg:amstext**], nicefrac [**pkg:nicefrac**] and scrlfile (from the KOMA-Script[2] bundle [**bnd:koma-script**]).

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 noth 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}
```

H_2O
Sb_2O_3
H^+
CrO_4^{2-}
$AgCl_2^-$
$[AgCl_2]^-$
Y^{99+}
Y^{99+}
$H_{2(aq)}$
NO_3^-
$(NH_4)_2S$
$^{227}_{90}Th^+$
V_{H_2O}
Ce^{IV}
$KCr(SO_4)_2 \cdot 12\,H_2O$

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.

Another notable difference: CHEMFORMULA tries to avoid math mode whenever possible:

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

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

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}
```

$_{2}H_2O$
$2\,H_2O$

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 ~. 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 [**iupac:greenbook**]:

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

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`
Introduced in version 4.1
Allows you to choose which command is used with `frac-style` = `{math}`. This needs to be a command sequence that takes two arguments that are set in math mode.

`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-print` = {⟨*cs*⟩} Default: `\chstoich`
This option allows to redefine the macro that prints the stoichiometric factors. ⟨*cs*⟩ should be a macro that takes one mandatory argument. *Please note that using this option will disable* CHEMFORMULA*'s stoichiometric parsing as that is done by the default command* `\chstoich`*.*

```
\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}
```

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

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_nB_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}
```
$\mathbf{A_2}B_3$ 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.

Please also note that formulas are placed inside a group!

```
\ch{A2\color{red}B3 C4}
```
$A_2B_3C_4$

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-}
```

$\ce{A+B}$ $\ce{AB+}$
$\ce{A-B}$ $\ce{AB-}$

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^*}
```

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

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}
```

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

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

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

$\ce{SO4^2-}$ $\ce{Ca^2+ SO4^2-}$

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}
```

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

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

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 `$\oplus$` ⊕.

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}}
```

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

- 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: SO_4^{2-}, $(SO_4)^{2-}$. The staggered arrangement is now recommended. *IUPAC Green Book [***iupac:greenbook***]*

6.5. Bonds

6.5.1. Native Bonds

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

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

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

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.

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

Table 1: Bonds available with \bond.

name	appearance	aliases
single	−	normal, sb
double	=	db
triple	≡	tp
dotted	···	semisingle
deloc	···	semidouble
tdeloc	···	semitriple
co>	→	coordright
<co	←	coordleft

C−C=C≡C···C≡C→C←C

Own Bonds CHEMFORMULA offers commands to define own bond types:

\NewChemBond{⟨*name*⟩}{⟨*code*⟩}
Introduced in version 4.3
Define the new bond type ⟨*name*⟩. Issue an error if a bond ⟨*name*⟩ already exists.

\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*⟩}
Introduced in version 4.12a
Define the new bond type ⟨*name*⟩ only if it doesn't exist yet.

\NewChemBondAlias{⟨*new name*⟩}{⟨*old name*⟩}
Introduced in version 4.3
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}
```

C⦙⦙⦙⦙⦙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.

`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.

`charge-style` = `text|math` Default: `text`
Style that is used to typeset the superscripts.

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

`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
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)
Ti*k*Z options for the bonds.

`bond-penalty` = {⟨*num*⟩} Default: `10000`
Introduced in version 4.0a
The penalty that is inserted after a bond for (dis-)allowing line breaks.

`radical-style` = {⟨*TikZ*⟩} (initially empty)
Ti*k*Z 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+}
```

$\mathrm{SO_4^{2-}\ NH_4^{+}\ Na^{+}}$

They are shifted if they *follow* a subscript which follows IUPAC recommendations [**iupac:greenbook**]. 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+}
```

$\mathrm{SO_4^{2-}\ NH_4^{+}\ Na^{+}}$
$\mathrm{SO_4^{2-}\ NH_4^{+}\ Na^{+}}$
$\mathrm{SO_4^{2-}\ NH_4^{+}\ Na^{+}}$

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-}
```

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

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-}
```

M^{x+} SO_4^{2-}
M^{x+} 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}
```

H_2SO_4 Na_3PO_4
H_2SO_4 Na_3PO_4
H_2SO_4 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}
```

A_nB_m H_2SO_4
A_nB_m 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}
```

$Na_3PO_3 \cdot H_2O$
$Na_3PO_3 \cdot H_2O$

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: $CH≡CH$

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}
```

$H—H + N≡N + O=O$
$H—H + N≡N + O=O$

6.7. Standalone Formulae

Introduced in version 4.0

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

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{ - } −`
oduced in on 4.3a
Creates the minus sign between compounds with space around it:
`\ch{M - H}` $\mathrm{M} - \mathrm{H}$

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

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

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

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

oduced in on 4.9

A rubber length.

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

oduced in on 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`.

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

7.1.2. Symbols

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

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

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*⟩.

`\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.

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

$H_2O + H_2SO_4 \longrightarrow H_3O^+ + HSO_4^-$
$H_2O + H2SO4 \longrightarrow H_3O^+ + HSO_4^-$

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.

3. Is this the correct English term? Please correct me if it isn't.

```
\ch{H2O +}\textcolor{red}{\ch{H2SO4}}\ch{-> H3O+ + HSO4-} \par
\ch{H2O + @{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-}
```

$H_2O + H_2SO_4 \longrightarrow H_3O^+ + HSO_4^-$
$H_2O + H_2SO_4 \longrightarrow H_3O^+ + HSO_4^-$
or of course:
$H_2O + H_2SO_4 \longrightarrow H_3O^+ + HSO_4^-$

$H_2O + H2SO4 \longrightarrow H_3O^+ + HSO_4^-$
$H_2O + H2SO4 \longrightarrow H_3O^+ + HSO_4^-$

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'}
```

$Ca^{II}O$
… Na + … $Cl_2 \longrightarrow$ … 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:

`->`
standard right arrow

`<-`
standard left arrow

`-/>`
does not react (right)

`</-`
does not react (left)

`<->`
resonance arrow

`<>`
reaction in both directions

`==`
stoichiometric equation

`<=>`
equilibrium arrow

`>=<`
Introduced in version 4.5
reversed equilibrium arrow

`<=>>`
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
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

`<==>` $\ch{<==>}$

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}
```

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

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}
```

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

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}
```

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

10. Index