Chemistry in LaTeX 2ε — an overview of existing packages and possibilities

Clemens Niederberger

Abstract

This article provides an overview of the most useful and popular packages for writing chemistry-related material with LaTeX 2ε. It is based upon a series of blog posts written by the author on the (German-language) blog `http://texwelt.de/blog`.

1 Introduction

It was just about two years ago when I started a series of blog posts on chemistry-related LaTeX packages. Readers interested in the series and able to understand German will find the whole series on `http://texwelt.de/blog/tag/chemie`. This article more or less is a translation of those blog articles into English and follows the same structure:

- The basics using the chemmacros package, see section 2.
- Molecular formulas and reaction equations, see section 3.
- Structural formulas — the chemfig package, see section 4.
- Safety Data and GHS — the packages ghsystem, rsphrases, and hpstatement, see section 5.
- Numbering of compounds — the chemnum package, see section 6.
- Packages for special applications, see section 7.

I should add the disclaimer that I am the author of the majority of packages discussed. All packages mentioned in this article are on CTAN (`http://ctan.org/pkg/`⟨*pkgname*⟩), and they are also all included in the two major TeX distributions, MiKTeX and TeX Live. Anyone with an up-to-date distribution can find their manuals by typing `texdoc` ⟨*pkgname*⟩ on the command line.

2 The basics using the chemmacros package

The chemmacros package (disclaimer: I am the author) offers a variety of macros for different applications in chemistry: writing IUPAC names, supporting formal charges and oxidation numbers, displaying spectroscopy data, and much more. Recently (late August 2015), chemmacros was updated to v5.0 which came with huge changes [4]. This article assumes version 5.0 is available.

As of version 5.0, chemmacros is constructed in a modular way. Some of the modules are loaded by default, while others have to be loaded by the user. This article loads one additional module:

```
\usechemmodule{redox}
```

2.1 IUPAC names

IUPAC (International Union of Pure and Applied Chemistry) names can get rather long and hard to read. Moreover, line breaking often becomes problematic. Treating them as normal text is unlikely to have good results:

```
(4-(4,4'-Bis(dimethylaminophenyl)benz%
hydryliden)cyclohexa-2,5-dien-1-yliden)%
dimethylammoniumchlorid
```

(4-(4,4'-Bis(dimethylaminophenyl)benzhydryliden)cyclohexa-2,5-dien-1-yliden)dimethylammoniumchlorid

chemmacros offers the macro `\iupac` for easing both input and output. Inside of this command, '`-`' outputs a dash which also allows for hyphenation in the rest of the word, similar to `\babelhyphen`. '`|`' inserts a breakpoint along with a tiny amount of space, which can be customized.

```
\iupac{(4-(4,4'-Bis(di|methyl|amino|%
  phenyl)benz|hydryliden)cyclo|hexa%
  -2,5-dien-1-yliden)di|methyl|%
  ammonium|chlorid}
```

(4-(4,4'-Bis(dimethylaminophenyl)benzhydryliden)cyclohexa-2,5-dien-1-yliden)dimethylammoniumchlorid

If it were only about the line breaks an easier solution might be to use suitable babel shorthands. But `\iupac` does more: inside many macros for common typesetting tasks, IUPAC names are defined:

```
\iupac{\nitrogen-methyl|benz|amide} \\
\iupac{\cip{2S,3S}-Wein|s\"aure} \\
\iupac{\zusammen-2-Butene}
```

N-methylbenzamide
(*2S*,*3S*)-Weinsäure
(*Z*)-2-Butene

2.2 Phase descriptors

chemmacros tries hard to follow IUPAC's recommendations whenever possible. IUPAC's demands concerning phase descriptors are the following:

> The [...] symbols are used to represent the states of aggregation of chemical species. The letters are appended to the formula in parentheses and should be printed in Roman (upright) type without a full stop (period). [2, p. 54]

chemmacros offers ready-to-use macros for the most common phase descriptors which exactly follow this recommendation:

```
\ch{
  C\sld{} + 2 H2O\lqd{}
  ->
  CO2\gas{} + 2 H2\gas
}
```

$\mathrm{C(s) + 2\,H_2O(l) \longrightarrow CO_2(g) + 2\,H_2(g)}$

Table 1: Basic use cases for mhchem and chemformula.

mhchem		**chemformula**	
`\ce{H2O}`	H_2O	`\ch{H2O}`	H_2O
`\ce{Sb2O3}`	Sb_2O_3	`\ch{Sb2O3}`	Sb_2O_3
`\ce{H+}`	H^+	`\ch{H+}`	H^+
`\ce{H2O}`	H_2O	`\ch{H2O}`	H_2O
`\ce{[AgCl2]-}`	$[AgCl_2]^-$	`\ch{[AgCl2]-}`	$[AgCl_2]^-$
`\ce{^{227}_{90}Th+}`	${}^{227}_{90}Th^+$	`\ch{^{227}_{90}Th+}`	${}^{227}_{90}Th^+$
`\ce{SO4^2-}`	SO_4^{2-}	`\ch{SO4^2-}`	SO_4^{2-}
`\ce{KCr(SO4)2 * 12H2O}`	$KCr(SO_4)_2 \cdot 12\,H_2O$	`\ch{KCr(SO4)2 * 12 H2O}`	$KCr(SO_4)_2 \cdot 12\,H_2O$

However, for almost every setting chemmacros offers the possibility of customizing the output. The same input with

```
\chemsetup{phases/pos=sub}
```

gives another very common way of denoting phases:

$C_{(s)} + 2\,H_2O_{(l)} \longrightarrow CO_{2(g)} + 2\,H_{2(g)}$

2.3 A lot more

Since describing all of chemmacros' features would be rather pointless (they're all described in the manual) and would also exceed the limits of this article, I'll just give one more example before continuing with the next topic.

```
\ch{
  2 "\OX{o1,Na}" + "\OX{r1,Cl}" {}2
  ->
  2 "\OX{o2,Na}" {}+ + 2 "\OX{r2,Cl}" {}-
}
\redox(o1,o2)[->]{\small OX: \ch{- 2 e-}}
\redox(r1,r2)[->][-1]{\small RED: \ch{+ 2 e-}}
```

$$\overset{\text{OX: } -2\,e^-}{\overset{\frown}{2\,Na} + \underset{\smile}{Cl_2} \longrightarrow 2\,Na^+ + 2\,Cl^-}\underset{\text{RED: } +2\,e^-}{}$$

3 Molecular formulas and reaction equations — the mhchem and chemformula packages

Two packages need to be mentioned here: mhchem by Martin Hensel and chemformula by me. mhchem has been around for a longer time and thus is better known and probably has a larger user base.

chemformula started as part of the chemmacros package in January 2012 and was released as an independent package in July 2013. It is still closely connected with chemmacros in that chemmacros uses it extensively. Thus, people using chemmacros should use chemformula instead of mhchem in order to have consistent input and output.

Both packages are similar in input and output, but have important differences in both aspects.

Table 2: mhchem's and chemformula's arrows.

mhchem		**chemformula**	
`\ce{->}`	$\longrightarrow$	`\ch{->}`	$\longrightarrow$
`\ce{<-}`	$\longleftarrow$	`\ch{<-}`	$\longleftarrow$
`\ce{<->}`	$\longleftrightarrow$	`\ch{<->}`	$\longleftrightarrow$
		`\ch{<>}`	$\rightleftharpoons$
`\ce{<=>}`	$\rightleftharpoons$	`\ch{<=>}`	$\rightleftharpoons$
`\ce{<=>>}`	$\rightleftharpoons$	`\ch{<=>>}`	$\rightleftharpoons$
`\ce{<<=>}`	$\rightleftharpoons$	`\ch{<<=>}`	$\rightleftharpoons$
		`\ch{-/>}`	$\not\longrightarrow$
		`\ch{</-}`	$\not\longleftarrow$
`\ce{=}`	$=$	`\ch{==}`	$=$
		`\ch{<o>}`	$\overset{\circ}{\longleftrightarrow}$

Martin Hensel, the author of mhchem, once described the differences as follows [3]:

> [chemformula's] philosophy is *control to the user*. [...] mhchem's philosophy, on the other hand, is *ease of use*.

Both packages mainly provide one macro for the typesetting of chemical formulas and reactions:

- mhchem has `\ce{`⟨*input*⟩`}` and
- chemformula has `\ch{`⟨*input*⟩`}`.

Table 1 shows a brief comparison of basic use cases.

The differences are most visible when typesetting reactions:

```
\ce{4 Na + 2 H2SO4 -> 2 Na2SO4 + 2 H2 ^} \\
\ch{4 Na + 2 H2SO4 -> 2 Na2SO4 + 2 H2 ^}
```

$4\,Na + 2\,H_2SO_4 \longrightarrow 2\,Na_2SO_4 + 2\,H_2\uparrow$
$4\,Na + 2\,H_2SO_4 \longrightarrow 2\,Na_2SO_4 + 2\,H_2\uparrow$

You will notice differences in spacing (which is customizable in chemformula) and of course the different arrows. Table 2 shows the different kinds of arrows both packages provide.

There is much more to be said about both packages but again I'll leave it with this introduction and refer to the respective manuals for details.

4 Structural formulas — the chemfig package

Both mhchem and chemformula lack support for showing the complexities of organic compounds. For typesetting structural (skeletal) formulas, the most common way probably is to use an external program like ChemDraw, export the results as images and include those into the LaTeX document. However, there are ways to create such formulas and diagrams from within LaTeX. To my knowledge there are five packages for doing this. People interested in details can have a look at [6]. In this article I'll cover only one of them: chemfig by Christian Tellechea. Again this will be a rather brief introduction — the manual has more than 80 pages.

First of all: chemfig is a generic package (it's the only one such among the packages described in this article); it can be used in LaTeX, ConTeXt and plain TeX.

```
\input chemfig.tex   % plain TeX
\usepackage{chemfig} % LaTeX
\usemodule[chemfig]  % ConTeXt
```

chemfig uses PGF (Ti*k*Z) for drawing the formulas. The most important command is

\chemfig{⟨*input*⟩}.

The basic rules are simple: letters are atoms and bonds are input as `-`, `=`, etc. Atoms are typeset in math mode (more precisely with `\printatom` which expands to `\ensuremath{\mathrm{#1}}`) so subscripts can be input as in math:

```
\chemfig{H-CH_3}
```

H—CH_3

There are a number of different bond types, shown in table 3. The appearance can be customized with a number of parameters. All bonds have an optional argument taking a comma-separated list of five parameters, in this order:

1. the angle of the bond,
2. a scale factor,
3. the "departure" atom number,
4. the "arrival" atom number, and
5. Ti*k*Z options for customization of the bond.

The angle can be input in three different ways:

- ⟨*integer*⟩ — denotes an angle in multiples of 45°.
- `:`⟨*real*⟩ — denotes the angle of the bond counterclockwise (positive) or clockwise (negative) to the horizontal.
- `::`⟨*real*⟩ — denotes the angle of the bond counterclockwise (positive) or clockwise (negative) relative to the bond before.

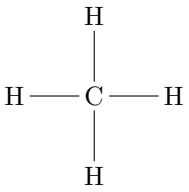
Thus, each of these input lines

Table 3: chemfig's bond types.

Input	Output
`\chemfig{A-B}`	A — B
`\chemfig{A=B}`	A = B
`\chemfig{A~B}`	A ≡ B
`\chemfig{A>B}`	A ▶ B
`\chemfig{A<B}`	A ◀ B
`\chemfig{A>:B}`	A ⫶ B
`\chemfig{A<:B}`	A ⫶ B
`\chemfig{A>|B}`	A ▷ B
`\chemfig{A<|B}`	A ◁ B

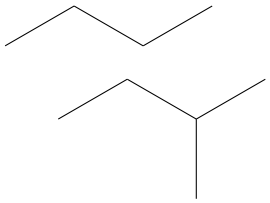
```
\chemfig{H-C(-[2]H)(-[6]H)-H}
\chemfig{H-C(-[:90]H)(-[:-90]H)-H}
\chemfig{H-C(-[::90]H)(-[::-90]H)-H}
```

gives the same output:



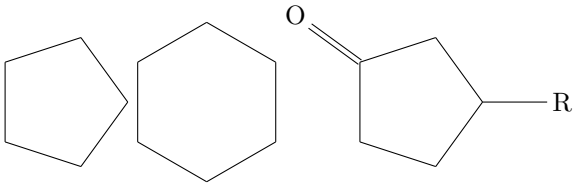
The last example also demonstrates how branching is done: surround the branch with parentheses.

```
\chemfig{-[:30]-[::-60]-[::60]}
\par\bigskip
\chemfig{-[:30]-[::-60](-[::-60])-[::60]}
```



Making rings is also quite easy:

```
\chemfig{*5(-----)}
\chemfig{*6(------)}
\chemfig{*5(--(-R)--(=O)-)}
```



Personally, I find this syntax very intuitive — indeed much more intuitive than some of the other packages mentioned in [6] — and it can be learned very fast.

I hope that the examples so far give a basic idea of chemfig's usage. To conclude this section, figure 1 shows an example of a more complicated scheme created with chemfig, to give you a larger impression of what is doable with the package. The code for the example can be seen in [5].

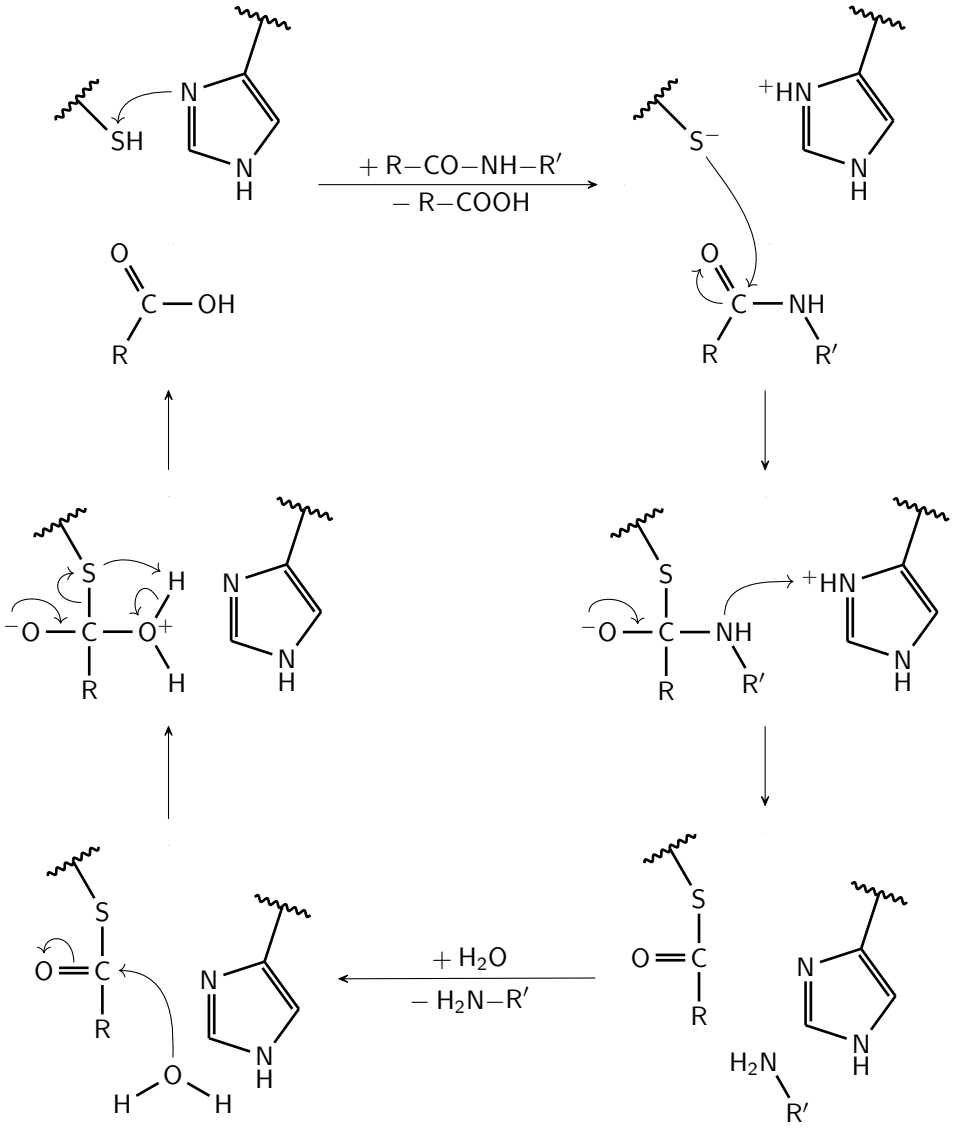


Figure 1: An example of using chemfig to create more complicated diagrams.

5 Safety data — the rsphrase, hpstatement, and ghsystem packages

Every student of chemistry knows the safety data phrases coordinated in the *Globally Harmonized System of Classification and Labelling of Chemicals*, in short, GHS. This system defines H (hazard) and P (precaution) statements. Before this became the standard there were R (risk) and S (safety) statements.

The R and S statements are available through the rsphrase package by Martin Hensel:

```
The statement \rsnumber{R34}
is ‘\rsphrase{R34}’
```

The statement R 34 is ‘Causes burns.’

Its usage is described in mhchem's manual.

The usage of the hpstatement by the same author also is described in mhchem's manual. The package provides support for the H and P statements of the GHS. It is a bit unfortunate that the package has

```
\RequirePackage{babel}
```

which forces the user to use the babel package and provide a language option.

```
The statement \hpnumber{H200}
is ‘‘\hpstatement{H200}’’
```

The statement H200 is “Unstable explosives.”

There is another package for GHS statements, ghsystem by me, which is a little bit older than hpstatement. It also provides an interface to get the statements per number.

```
The statement \ghs[hide]{h}{200}
```

```
is ``\ghs*{h}{200}'': \par
\ghs{h}{200}
```

The statement H200 is "Unstable explosives.":

H200: Unstable explosives.

In addition it provides the possibility of inserting the GHS pictograms.

```
\ghspic{skull}
\ghspic{flame}
\ghspic{aqpol}
```

Those pictograms are available as JPG, EPS, PNG, and PDF. There are plans to eventually provide the pictograms as Ti*k*Z pictures.

6 Numbering compounds — the chemnum package

It is a common task for chemists to refer to compounds with numbers. There are a number of packages for this task but the most flexible one is chemnum (again by me).

The basic usage is simple: add `\cmpd{`⟨*id*⟩`}` where you want the number:

```
This is about \ch{CH3-CH2-OH} (\cmpd{ethanol}).
\cmpd{ethanol} is used\ldots
```

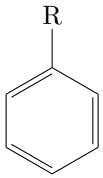
This is about CH_3-CH_2-OH (**1**). **1** is used...

Upon the first usage of ID ⟨*id*⟩ the command will create a new number for the compound. Later uses will then refer to this number. For usage in section titles or captions a version with optional `+` (`\cmpd+{`⟨*id*⟩`}`) can be used to refer to the compound without creating the number. There is also a starred version which will create the number but prints nothing. In this respect the syntax is the same as in the older package chemcompounds by Stephan Schenk.

chemnum also natively supports "sub-labelling": `\cmpd{`⟨*id*⟩`.`⟨*sub_id*⟩`}`.

Both ⟨*id*⟩ and ⟨*sub_id*⟩ may be comma-separated lists. If ⟨*sub_id*⟩ is a list it *must* be enclosed in braces.

```
\setatomsep{2em}
\chemname
  {\chemfig{*6(=-=-(-R)=-)}}
  {\cmpd{benzene.{H,Me,OH,NH2}}}
\begin{tabular}{lll}
    & \ch{-R} & Name \\
  \midrule
  \cmpd[sub-only]{benzene.H}   & \ch{-H}
    & Benzene \\
  \cmpd[sub-only]{benzene.Me}  & \ch{-CH3}
    & Toluene \\
  \cmpd[sub-only]{benzene.OH}  & \ch{-OH}
    & Phenol \\
  \cmpd[sub-only]{benzene.NH2} & \ch{-NH2}
    & Phenylamine (Aniline)
\end{tabular}
```



2a–d

	−R	Name
a	−H	Benzene
b	$-CH_3$	Toluene
c	−OH	Phenol
d	$-NH_2$	Phenylamine (Aniline)

This example shows that by default lists of sublabels are compressed. This can be turned off with an option.

```
\cmpd{benzene.{H,Me,OH,NH2}}
\setchemnum{compress=false}
\cmpd{benzene.{H,Me,OH,NH2}}
```

2a–d **2a,b,c,d**

When you have a list of labels which contains an ⟨*id*⟩ more than once but with different sublabels, each entry will be printed on its own:

```
\cmpd{a,b.{A},c,b.{B,C},a}
```

3, **4a**, **5** and **4b,c**

However, there is an option to merge those different entries into one:

```
\setchemnum{merge=true}
\cmpd{a,b.{A},c,b.{B,C},a}
```

3, **4a–c** and **5**

What we have not seen yet in the above examples is that lists are sorted as well:

```
\cmpd{a,b,c,b,a,ethanol,benzene} vs.\@
\cmpd{a}, \cmpd{b}, \cmpd{c}, \cmpd{b},
\cmpd{a}, \cmpd{ethanol} and \cmpd{benzene}
```

1, **2**, **3**, **4** and **5** vs. **3**, **4**, **5**, **4**, **3**, **1** and **2**

7 Packages for special applications

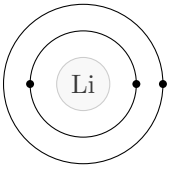
The number of packages available for support of chemistry-related topics is quite large. Both CTAN (`http://ctan.org/topic/chemistry`) and I, on my blog (`http://mychemistry.eu/known-packages/`), maintain a list of packages related to chemistry. The lists differ slightly, mostly because I also include fringe cases, e.g., packages which may belong more to biology than to chemistry. In this section I pick a few of my packages and will present them extremely briefly:

- bohr
- elements
- endiagram
- modiagram
- chemgreek

7.1 The bohr package

The package bohr was created as an answer to a question on http://tex.stackexchange.com [1]. It provides a simple way of drawing atomic orbitals according to the Bohr model:

```
\bohr{3}{Li}
```



7.2 The elements package

The elements package is rather new (released in June 2015). From its abstract:

> This package provides means for retrieving properties of chemical elements like atomic number, element symbol, element name, electron distribution or isotope number. Properties are defined for the elements up to the atomic number 112.

The following example gives a short impression of its capabilities:

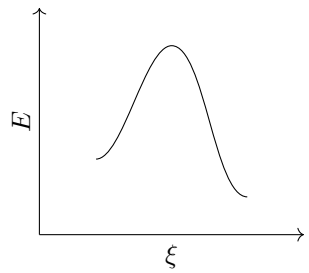
```
\elementname{Cu} \\
\elementname{11} \\
\atomicnumber{U} \\
\elconf{Cl} \\
\savemainelementisotope\foo{C}\foo
```

Copper
Sodium
92
$1s^2 2s^2 2p^6 3s^2 3p^5$
12

7.3 The endiagram package

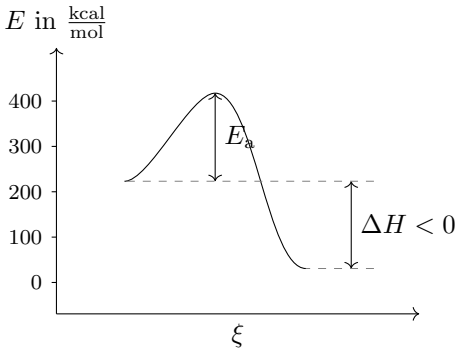
The endiagram package lets you create potential energy curve diagrams.

```
\begin{endiagram}
  \ENcurve{1,4,0}
\end{endiagram}
```



A more advanced example:

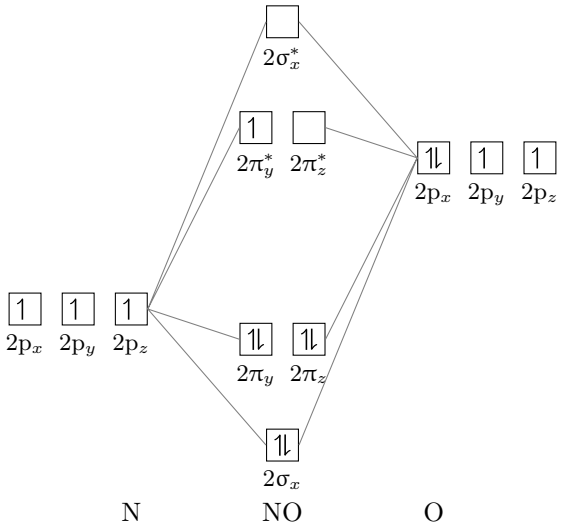
```
% preamble:
% uses 'siunitx' (loaded by 'endiagram')
\DeclareSIUnit{\calory}{cal}
\sisetup{per-mode = fraction}
% document:
\ENsetup{
  y-label = above ,
  energy-step = 100 ,
  energy-unit = \kilo\calory\per\mole ,
  energy-unit-separator = { in } ,
  calculate = false ,
  AddAxisLabel/font = \footnotesize
}
\begin{endiagram}[scale=1.2]
  \ENcurve{2.232,4.174,.308}
  \AddAxisLabel*{0;1;2;3;4}
  \ShowEa[label,connect={draw=none}]
  \ShowGain[label]
\end{endiagram}
```



7.4 The modiagram package

Like the bohr package, the modiagram package was created as an answer to a question on http://tex.stackexchange.com [7]. It offers an interface for drawing molecular orbital diagrams based on Ti*k*Z.

```
\begin{MOdiagram}[labels,names,style=square]
 \atom[N]{left}{
   2p = {0;up,up,up}
 }
 \atom[O]{right}{
   2p = {2;pair,up,up}
 }
 \molecule[NO]{
   2pMO  = {1.8,.4;pair,pair,pair,up}
 }
\end{MOdiagram}
```



7.5 The chemgreek package

The chemgreek package is a support package rather than one to be used by users directly. At the time of writing, the packages chemmacros, chemnum and mhchem make use of its features. The purpose of chemgreek is to provide a unified interface for upright Greek letters regardless of which math or font package provides upright letters and regardless of how those are accessed. This is important in chemistry since upright Greek letters are used in a variety of places:

```
\iupac{\a-\D-glucopyranose}
```

α-D-glucopyranose

The chemgreek package knows about a number of packages providing upright Greek letters. It is able to detect if one of those packages is loaded, and if a unique choice is possible it defines macros for each of the 24 lowercase and uppercase letters. If no unique choice is possible it falls back to a default mapping and users have to make a choice themselves. The chemmacros package has a user interface for this: `\chemsetup{greek =` ⟨*mapping*⟩`}`.

Other packages can now use those macros to define macros of their own.

If a document needs a font package with upright Greek letters which chemgreek doesn't know about, users have the capability to define a new ⟨*mapping*⟩ themselves, and (for example) activate it with chemmacros' interface, and then use chemmacros nomenclature commands using the new mapping.

8 Summary

This article briefly discussed the usage of the packages chemmacros, mhchem, chemformula, chemfig, rsphrase, hpstatement, ghsystem, chemnum, bohr, elements, endiagram, modiagram, and chemgreek. It shows that today there is considerable support for different typesetting tasks in chemistry and gives a short overview of some the existing possibilities.

References

[1] Andreas. Draw Bohr atomic model with electron shells in TEX? `http://tex.stackexchange.com/questions/73410/` [accessed 2015-08-29].

[2] E. Richard Cohan, Tomislav Cvitaš, Jeremy G. Frey, Bertil Holmström, Kozo Kuchitsu, Roberto Marquardt, Ian Mills, Franco Pavese, Martin Quack, Jürgen Stohner, Herbert L. Strauss, Michio Takami, and Anders J. Thor. *"Quantities, Symbols and Units in Physical Chemistry", IUPAC Green Book*. IUPAC & RSC Publishing, Cambridge, 3rd edition, 2nd printing, 2008.

[3] Martin Hensel. On mhchem vs. chemformula. `http://tex.stackexchange.com/q/237946#comment565361_238214` [accessed 2015-08-29].

[4] Clemens Niederberger. a new chemmacros — but how? `http://www.mychemistry.eu/2015/07/a-new-chemmacros-but-how/` [accessed 2015-08-29].

[5] Clemens Niederberger. A seemingly complex example for the usage of chemfig. `https://www.overleaf.com/read/nhbyvqvrqffj` [accessed 2015-08-29].

[6] rake. Can you make chemical structure diagrams in LATEX? `http://tex.stackexchange.com/questions/52722` [accessed 2015-08-29].

[7] Richard Terrett. Molecular orbital diagrams in LATEX? `http://tex.stackexchange.com/questions/13863` [accessed 2015-08-29].

◇ Clemens Niederberger
Am Burgrain 3
71083 Herrenberg
Germany
`contact (at) mychemistry dot eu`
`http://www.mychemistry.eu/`