Useful Packages

William Beason and Evan Ott

March 27, 2014

Table of Contents

Table of Contents

Why siunitx?

Typesetting units in normal LaTeXis annoying.
Consider this equation:

$$E_n = \frac{n^2 4.38 \times 10^{-67}\ \mathrm{J}^2\,\mathrm{s}^2}{8 \times 9.11 \times 10^{-31}\,\mathrm{kg}\,\mathrm{fm}^2} = 6.02 \times 10^{-8}\,\mathrm{kg}\,\mathrm{m}^2\mathrm{s}^{-2} \qquad (1)$$

Why siunitx?

Typesetting units in normal LaTeXis annoying.
Consider this equation:

$$E_n = \frac{n^2 4.38 \times 10^{-67}\ \mathrm{J}^2\,\mathrm{s}^2}{8 \times 9.11 \times 10^{-31}\,\mathrm{kg}\,\mathrm{fm}^2} = 6.02 \times 10^{-8}\,\mathrm{kg}\,\mathrm{m}^2\mathrm{s}^{-2} \qquad (1)$$

This is what typesetting it looks like:

```
\begin{equation}
E_n=\frac{n^2 4.38\times 10^{-67}\;
\mathrm{J}^2\,\mathrm{s}^2}{8\times
9.11\times10^{-31}\,\mathrm{kg\,fm}^2}
=6.02\times10^{-8}\,\mathrm{kg\,m}^2
\mathrm{s}^{-2}
\end{equation}
```

The Difference

Meanwhile, the same equation in **siunitx**:

$$E_n = \frac{n^2 4.38 \times 10^{-67}\,\mathrm{J}^2\,\mathrm{s}^2}{8 \times 9.11 \times 10^{-31}\,\mathrm{kg\,fm}^2} = 6.02 \times 10^{-8}\,\mathrm{kg\,m}^2\,\mathrm{s}^{-2} \qquad (2)$$

```
\begin{equation}
E_n=\frac{n^2\SI{4.38e-67}{\J\squared\s\squared}}
{8\times\SI{9.11e-31}{\kg\femto\m\squared}}
=\SI{6.02e-8}{\kg\m\squared\per\s\squared}
\end{equation}
```

Most Common Commands

- `\si` - units
- `\SI` - a number and units
- `\numlist` - a list of numbers
- `\numrange` - a range of numbers
- `\SIlist` - a list of numbers with unts
- `\SIrange` - a range of numbers with units
- `\ang` - an angle
- `\num` - a number

\si and \SI

These are the bread and butter of siunitx. The command \SI takes two required arguments, a number and a unit.

```
\SI{3.4e5}{\centi\m\cubed\per\s}
```

$$3.4 \times 10^{5}\,\mathrm{cm}^{3}\,\mathrm{s}^{-1} \tag{3}$$

\si and \SI

These are the bread and butter of **siunitx**. The command \SI takes two required arguments, a number and a unit.

```
\SI{3.4e5}{\centi\m\cubed\per\s}
```

$$3.4 \times 10^{5}\,\mathrm{cm}^{3}\,\mathrm{s}^{-1} \tag{3}$$

```
\SI{3.4e5}{cm^3.s^{-1}}
```

$$3.4 \times 10^{5}\,\mathrm{cm}^{3}\,\mathrm{s}^{-1} \tag{4}$$

siunitx can interpret units directly or as macros.

Units in siunitx

Most units are native to siunitx they all have macros in their singular form, and many have abbreviations.

Unit	Macro(s)	Symbol
ampere	\ampere	A
kelvin	\K,\Kelvin	K
Celsius	\degreeCelsius	°C
joule	\J,\Joule	J
meter	\m,\meter,\metre	m
kilogram	\kg,\kilogram	kg
second	\s,\second	s
hertz	\hertz	Hz
degree	\degree	°

All SI prefixes are included, so you can write out anything from yocto to yotta.

For most cases, you can type out units as you would say them. Just make every standard unit a macro and make macro words singular.

$\mathrm{g\,cm^{-3}}$
$\mathrm{kg\,C/m^{3}}$
$\mathrm{planets/ua^{3}}$
$1 \times 10^{-23}\,\mathrm{mol\,\AA^{-3}}$
$2\,\mathrm{g\,m\,Hz} \times 3\,\mathrm{g\,m\,Hz}$

```
\si{\gram\per\cubic\centi\meter}\\
\si{\kilogram.\coulomb\per\meter\cubed}\\
\si{planets\per\astronomicalunit\cubed}\\
\SI{1e-23}{\mol\per\cubic\angstrom}
\SI{2 x 3}{\gram.\meter.\hertz}
```

Note the . between multiplied units. This lets **siunitx** know that these are separate units. All x's are replaced with $\times$.

Missing Units

`siunitx` does not include some common units such as \parsec, \lightyear, and \micron. To create macros for them, use `\DeclareSIUnit` in the preamble.

```
\DeclareSIUnit\parsec{pc}
\DeclareSIUnit{\lightyear}{ly}
\DeclareSIUnit\micron{\micro\m}
```

Missing Units

`siunitx` does not include some common units such as \parsec, \lightyear, and \micron. To create macros for them, use `\DeclareSIUnit` in the preamble.

```
\DeclareSIUnit\parsec{pc}
\DeclareSIUnit{\lightyear}{ly}
\DeclareSIUnit\micron{\micro\m}
```

Now we can use them as macros.
$\mathrm{pc}^{2}\,\mathrm{ly}^{-1}\,\mu\mathrm{m}^{-1}$

```
\si{\parsec\squared\per\lightyear\per\micron}
```

There are similar commands for prefix and postfix powers.

Lists and Ranges

Sometimes we want to display lists or ranges of numbers and units. `siunitx` has built-in typesetting for those.

$1\,\mathrm{m\,s^{-2}}$, $10\,\mathrm{m\,s^{-2}}$ and $100\,\mathrm{m\,s^{-2}}$

(1, 10 and 100) $\mathrm{m\,s^{-1}}$

1, 10 and $100\,\mathrm{m\,s^{-1}}$

$80\,\mathrm{K}$ to $100\,\mathrm{K}$

(80 to 100) K

```
\SIlist{1;10;100}{\m\per\s\squared}\\
\SIlist[list-units=brackets]{1;10;100}{\m\per\s}\\
\SIlist[list-units=single]{1;10;100}{\m\per\s}\\
\SIrange{80}{100}{\K}\\
\SIrange[range-units=brackets]{80}{100}{\K}
```

Numlist and numbrange work identically - the just don't take units.

Uncertainty

We are almost never exactly sure of a measurement. As usual, `situnitx` has features for this.

$(100 \pm 5)\,\mathrm{K}$

$100\,\mathrm{K} \pm 5\,\mathrm{K}$

$100 \pm 5\,\mathrm{K}$

$(100 \pm 5) \times 10^{5}\,\mathrm{m}$

Requires a setup command, preferrably in the preamble.

```
\sisetup{separate-uncertainty}
\SI{100(5)}{\K}
\SI[multi-part-units=repeat]{100(5)}{\K}
\SI[multi-part-units=single]{100(5)}{\K}\\
\SI{100(5)e5}{\m}
```

Angles

1°
2°
1.234°
$12^\circ 34'$
$1^\circ 23' 45''$

```
\ang{1}\\
\ang{2;;}\\
\ang{1.234}\\
\ang{12;34;}\\
\ang{1;23;45}
```

Table of Contents

Introduction

Typesets chemistry formulae intuitively.

$\mathrm{NaCl_{(aq)} + AgNO_{3(aq)} \longrightarrow NaNO_{3(aq)} + AgCl_{(s)}}$

```
\ce{NaCl_{(aq)} + AgNO3_{(aq)} ->
NaNO3_{(aq)} + AgCl_{(s)}}
```

Molecules

Use \ce{} to typeset chemistry.

Code	Result
H20	H_2O
C3H8	C_3H_8
H+	H^+
OH-	OH^-
O^2-	O^{2-}
SO4^2-	SO_4^{2-}
[SO4]^2-	$[SO_4]^{2-}$
Fe^{26+}	Fe^{26+}
Pb(NO3)2	$Pb(NO_3)_2$
KrO3	KrO_3

Amounts

To make amounts, just put them before chemicals.

Code	Result
2H2O	$2\,H_2O$
1/2C2H6	$\frac{1}{2}\,C_2H_4$
0.4C5H10	$0.4\,C_5H_{10}$

To do the last line, you may need to update `mhchem`.

To make isotopes, put superscripts and subscripts in front of elements.

Code	Result
^227_90Th	$^{227}_{90}Th$
^1_1H	$^{1}_{1}H$
^2_1H	$^{2}_{1}H$

`mhchem` includes an extensive list of arrows with options.

Code	Result
`H2 + 1/2O2 -> H2O`	$\mathrm{H_2 + \frac{1}{2} O_2 \longrightarrow H_2O}$
`H2 + 1/2O2 <- H2O`	$\mathrm{H_2 + \frac{1}{2} O_2 \longleftarrow H_2O}$
`H2O <=> H+ + OH-`	$\mathrm{H_2O \rightleftharpoons H^+ + OH^-}$
`H2O <=>[\alpha][\beta] H+ + OH-`	$\mathrm{H_2O \underset{\beta}{\overset{\alpha}{\rightleftharpoons}} H^+ + OH^-}$
`N2H2 <-> N2H2'`	$\mathrm{N_2H_2 \longleftrightarrow N_2H_2'}$
`A ->[\ce{+H2O}] B`	$\mathrm{A \xrightarrow{+H_2O} B}$
`A ->C[+H2O] B`	$\mathrm{A \xrightarrow{+H_2O} B}$
`A ->[\text{heat}] B`	$\mathrm{A \xrightarrow{heat} B}$

Combined with siunitx

A macro in siunitx which becomes much more useful with mhchem is \of:

$$1\,\mathrm{g_{H_2O}} = 1\,\mathrm{cm^3_{H_2O}} = 1\,\mathrm{ml_{H_2O}} \tag{5}$$

```
\SI{1}{\g\of{\ce{H2O}}}
=\SI{1}{\cubic\cm\of{\ce{H2O}}}
```

Table of Contents

\bordermatrix is a command natively included in LaTeXallowing each column and row to be labelled on the top and left of a matrix.

$$M = \begin{array}{c} \\ A \\ B \end{array} \begin{array}{c} \begin{array}{cc} x & y \end{array} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{array} \tag{6}$$

```
\begin{equation}
M=\bordermatrix{&x&y\cr A&1&0\cr B&0&1\cr}
\end{equation}
```

Typeset \bordermatrix as you would a matrix, except replace the \\to break lines with \cr and remember it is a command, not an environment. The first row and column are typeset outside the matrix.

$$H = \bordermatrix{ & H & \langle 211| & \langle 121| & \langle 112| \cr |211\rangle & E_{211} & 0 & 0 \cr |121\rangle & 0 & E_{121} & \gamma \cr |112\rangle & 0 & -\gamma & E_{112} \cr } \tag{7}$$

Table of Contents

`verbatim` and `lstlistings` are packages including environments allowing code to be typeset.

```
\begin{verbatim}
Anything can be typed here! \[\end{document} 2^55
\end{verbatim}
```

```
\begin{lstlisting}
Anything can be typed here! \[\end{document} 2^55
\end{lstlisting}
```

`verbatim` doesn't need setup to be used effectively. `lstlistings` has an important option which should be set, language. This enables syntax formatting.

```
\lstset{language={C++}}
\lstset{language={[LaTeX]TeX}}
```

```
#include <iostream>

int main()
{
    std::cout << "Hello, world!\n";
}
```