

*LaTeX typography guide*

*Benjamin J. Morgan*

*June 4, 2025*

This document is for ... One of the benefits of writing using L<sup>A</sup>T<sub>E</sub>X is the ease of producing attractive documents with good typography.

TODO: why worry about typography?

## Contents

<i>1 Use the <code>math</code> environment for mathematical text</i>	4
<i>2 Working with numbers and units using <code>siunitx</code></i>	5
<i>Basic physical quantities</i>	5
<i>Complex units and scientific notation</i>	6
<i>Ranges and lists of values</i>	6
<i>Number formatting</i>	6
<i>Uncertainties</i>	7
<i>Angles</i>	7
<i>Configuring <code>siunitx</code></i>	8
<i>Declaring custom units</i>	9
<i>Manual formatting: understanding the principles</i>	9
<i>3 The multiplication symbol</i>	9
<i>4 Numerical values in tables should be aligned on the decimal point.</i>	10
<i>5 Euler's number</i>	11
<i>6 The differential sign</i>	11
<i>7 Function names</i>	12
<i>8 Quotes</i>	13
<i>9 Automatic sizing of brackets in equations</i>	13
<i>10 Periods / full stops</i>	14
<i>11 Non-breaking spaces</i>	14
<i>12 Hyphens and dashes</i>	15
<i>13 Punctuating equations</i>	16
<i>14 Ellipses</i>	16
<i>15 Angle brackets</i>	17
<i>16 Text in math mode</i>	17

<i>17 Chemical symbols</i>	17
<i>18 Bibliography formatting: a systematic approach</i>	18
<i>Obtaining initial bibliography entries</i>	18
<i>Citation keys</i>	18
<i>Journal names</i>	19
<i>Title formatting</i>	19
<i>Example transformation</i>	20
<i>19 Vectors</i>	21
<i>20 Krämer-Vink notation</i>	21
<i>21 Further reading</i>	22
<i>General typography</i>	22
<i>L<sup>E</sup>T<sub>E</sub>X</i>	22

## 1 Use the math environment for mathematical text

L<sup>A</sup>T<sub>E</sub>X typesets text inside math environments differently to regular text. The two most common ways to include pieces of mathematical text and symbols are *inline* formulae, and *equations*.

Inline formulae are displayed as part of their surrounding text, such as  $y = mx + c$ . They are delimited using \$ ... \$ symbols. As an example, the first sentence in this paragraph is generated using

```
Inline formulae are displayed as part of their
surrounding text, such as $y=mx+c$.
```

Equations appear on their own lines, and are usually numbered (by default), e.g.

$$y = mx + c. \quad (1)$$

Equation environments are delimited using `\begin{equation}` and `\end{equation}` commands.

e.g. the equation above is generated using

```
\begin{equation}
y=mx+c.
\end{equation}
```

Formulae numbers can be removed by using `\begin{equation*}` and `\end{equation*}` commands to delimit the environment.

When referring to variables in regular text, surrounding these with \$ ... \$ symbols means they are shown consistently in an italicised math font.

Consider the following example, which does not use inline math environments when describing the variables  $m$ ,  $c$ , and  $y$ .

```
The equation for a straight line is
\begin{equation*}
y=mx+c,
\end{equation*}
where m is the slope and c is the intercept on the y axis.
```

The equation for a straight line is

$$y = mx + c,$$

where  $m$  is the slope and  $c$  is the intercept on the  $y$  axis.

Using inline math environments, this becomes

```
The equation for a straight line is
```

```
\begin{equation*}
y=mx+c,
\end{equation*}
where $m$ is the slope and $c$ is the intercept on
the $y$ axis.
```

The equation for a straight line is

$$y = mx + c,$$

where  $m$  is the slope and  $c$  is the intercept on the  $y$  axis.

The variables  $m$ ,  $c$ , and  $y$  are now correctly typeset using the italicised math font.

## 2 Working with numbers and units using *siunitx*

The *siunitx* package<sup>1</sup> provides a comprehensive solution for typesetting numbers, units, and physical quantities. This package should be your primary tool for all numerical formatting in scientific documents.

<sup>1</sup> <https://ctan.org/pkg/siunitx?lang=en>

### *Basic physical quantities*

Physical quantities are formally the product of a number and a unit, with the space regarded as a multiplication sign. The *siunitx* package handles this formatting automatically using the `\SI{number}{unit}` command:

```
The blue whale can grow up to \SI{30}{\meter} in length.
```

The blue whale can grow up to 30 m in length.

The package automatically provides appropriate spacing between the number and unit, and prevents them from being split across line breaks. Compare this with manual formatting approaches:

```
The blue whale can grow up to 30 m in length. % Wrong spacing
The blue whale can grow up to 30m in length. % No spacing
```

The blue whale can grow up to 30 m in length. (wrong spacing)

The blue whale can grow up to 30m in length. (no spacing)

### *Complex units and scientific notation*

The `siunitx` package excels at handling complex units and scientific notation:

```
$\Delta H = \SI{1.23e3}{\joule\per\mole\kelvin} \\ 
g = \SI{9.8}{\meter\per\second\squared} \\ 
\rho = \SI{2.7}{\gram\per\centi\meter\cubed}
```

$$\begin{aligned} \Delta H &= 1.23 \times 10^3 \text{ J mol}^{-1} \text{ K}^{-1} \\ g &= 9.8 \text{ m s}^{-2} \\ \rho &= 2.7 \text{ g cm}^{-3} \end{aligned}$$

### *Ranges and lists of values*

For ranges and lists of values, `siunitx` provides dedicated commands that ensure units appear with each value:

```
\SIrange{30}{40}{\meter} \\ 
\SIlist{30; 35; 40}{\meter}
```

$$\begin{aligned} &30 \text{ m to } 40 \text{ m} \\ &30 \text{ m, } 35 \text{ m and } 40 \text{ m} \end{aligned}$$

This is preferred over manually listing values:

```
$30$, $35$, and $40$\text{, }\text{\texttt{\textbackslash text{m}}}$ % Incorrect - missing units
```

$$| 30, 35, \text{ and } 40 \text{ m (incorrect - missing units)}$$

### *Number formatting*

The `\num{}` command formats numbers consistently, including appropriate use of thousand separators and scientific notation:

```
\num{1234567} \\ 
\num{1.23456789} \\ 
\num{1.23e-4} \\ 
\num{0.0001234}
```

$$\begin{aligned} &1\,234\,567 \\ &1.\,234\,567\,89 \\ &1.23 \times 10^{-4} \\ &0.000\,123\,4 \end{aligned}$$

## *Uncertainties*

Scientific measurements often include uncertainties, which `siunitx` handles automatically. By default, `siunitx` uses compact notation for uncertainties:

```
\SI{9.8(2)}{\meter\per\second\squared} \\
\SI{1.23(5)}{\meter} \\
\num{1.234(5)}
```

9.8(2) m s <sup>-2</sup>
1.23(5) m
1.234(5)

The compact notation `9.8(2)` represents  $9.8 \pm 0.2$ , and `1.23(5)` represents  $1.23 \pm 0.05$ .

To use explicit  $\pm$  notation instead, you can set this globally or locally:

**Global setting** (affects entire document):

```
\sisetup{uncertainty-mode = separate}
```

**Local setting** (affects individual commands):

```
\SI[uncertainty-mode = separate]{1.23 \pm 0.05}{\meter} \\
\SI[uncertainty-mode = separate]{9.8 \pm 0.2}{\meter\per\second\squared} \\
\num[uncertainty-mode = separate]{1.234 \pm 0.005}
```

(1.23 $\pm$ 0.05) m
(9.8 $\pm$ 0.2) m s <sup>-2</sup>
1.234 $\pm$ 0.005

Both input formats work regardless of the output mode setting:

```
% Compact input -> output depends on uncertainty-mode setting
\SI{9.8(2)}{\meter\per\second\squared}

% Explicit +/- input -> output depends on uncertainty-mode setting
\SI{9.8 \pm 0.2}{\meter\per\second\squared}
```

The `siunitx` package automatically converts between formats based on your `uncertainty-mode` setting.

## *Angles*

For simple angular measurements, use the `\degree` unit with the standard `\SI{}` command:

```
\SI{45}{\degree} \\
\SI{90.5}{\degree} \\
\SI{180}{\degree}
```

45°  
90.5°  
180°

For more complex angular notation involving degrees, arcminutes, and arcseconds, `siunitx` provides the dedicated `\ang{}` command:

```
\ang{45} \\
\ang{12;34;56} \\
\ang{90.5}
```

45°  
12°34'56"  
90.5°

The `\ang{}` command uses semicolon notation where the components represent degrees, arcminutes, and arcseconds respectively. For simple degree measurements, either approach works, but `\SI{45}{\degree}` is more consistent with the pattern of using `siunitx` for all physical quantities.

### *Configuring siunitx*

You can configure `siunitx` globally using `\sisetup{}`:

```
\sisetup{
    per-mode = reciprocal,
    bracket-unit-denominator = true,
    sticky-per,
    uncertainty-mode = separate,
}
```

These options control important formatting choices:

- `per-mode=reciprocal` formats units like  $5 \text{ m s}^{-1}$  as  $5 \text{ m s}^{-1}$  rather than  $\text{m/s}$
- `bracket-unit-denominator=true` adds brackets around complex denominators
- `sticky-per` ensures multiple units after `\per` are all treated as denominators
- `uncertainty-mode=separate` formats uncertainties as  $(1.23 \pm 0.05) \text{ m}$  rather than  $(1.23 \pm 0.05) \text{ m}$

## *Declaring custom units*

For units not predefined in `siunitx`, you can declare your own using `\DeclareSIUnit`. A common example is the Ångström:

```
\DeclareSIUnit{\angstrom}{\textup{\AA}}
```

This allows you to use `\SI{1.5}{\angstrom}` in your document:

```
The bond length is \SI{1.54}{\angstrom}.
```

| The bond length is 1.54 Å.

## *Manual formatting: understanding the principles*

While `siunitx` should be your primary approach, understanding the underlying typography principles is valuable. The manual approach places numbers and units in a math environment with a small space:

```
$30\,,\text{m}$ % Manual approach
```

| 30 m (manual formatting)

The `\,` command produces a thin space, and `\text{}` ensures the unit appears in the non-mathematical font. However, this manual approach lacks the automatic formatting, line-break protection, and extensive unit support that `siunitx` provides.

## *3 The multiplication symbol*

The correct symbol for multiplication (“times”) is  $\times$ . This can be generated using `\times` in a math environment.

```
$6\times6\times6$
```

|  $6 \times 6 \times 6$ .

If necessary, you can slightly reduce the spacing around the  $\times$  symbol using the “small negative space” command `\!`:

```
$6\!\times\!6\!\times\!6$
```

|  $6 \times 6 \times 6$ .

Using the letter “ $x$ ” gives the wrong symbol and incorrect spacing.  
e.g.

6x6x6

| 6x6x6

\$6x6x6\$

| 6x6x6

- 4 Numerical values in tables should be aligned on the decimal point.

Numerical data in tables containing decimal places should be vertically aligned with consistent decimal points.

```
\begin{table}
\centering
\begin{tabular}{ll} \hline
column 1 & column 2 \\
12.345 & 6.789 \\
123.4 & 12.4 \\ \hline
\end{tabular}
\end{table}
```

column 1	column 2
12.345	6.789
123.4	12.4

Table 1: Incorrect vertical alignment of numbers.

The `siunitx` package provides the `S` column type that automatically aligns numbers on their decimal points. Column headers need to be enclosed in braces {} to prevent them from being interpreted as numbers:

```
\begin{table}
\centering
\begin{tabular}{SS} \hline
{column 1} & {column 2} \\ \hline
12.345 & 6.789 \\
123.4 & 12.4 \\ \hline
\end{tabular}
\end{table}
```

The `S` column type can also handle units and uncertainties automatically:

column 1	column 2
12.345	6.789
123.4	12.4

Table 2: Using the `siunitx S` column type.

```
\begin{tabular}{S[table-format=3.2]} \hline
Temperature / \si{\kelvin} \\ \hline
298.15 \\
373.15 \\
77.35 \\ \hline
\end{tabular}
```

## 5 Euler's number

Euler's number  $e$  is a mathematical constant, and should not be italicised. Instead, use `\mathrm{e}`.

```
$\mathrm{e}^x$
```

versus

```
$e^x$
```

|  $e^x$

## 6 The differential sign

The “d” in integrals and derivatives is not a variable, and should not be italicised. Again, use `\mathrm{d}`.

```
\begin{equation}
y = \int f(x) \mathrm{d}x.
\end{equation}
```

$$y = \int f(x) \mathrm{d}x.$$

Note the small space between  $f(x)$  and  $\mathrm{d}x$ , produced using `\,`.

```
\begin{equation}
\frac{\mathrm{d}y}{\mathrm{d}x} = g(x).
\end{equation}
```

$$\frac{dy}{dx} = g(x).$$

## 7 Function names

In a formula like  $\exp(x + y)$ , the “`exp`”, which represents the exponential function, is a single word that is usually set in roman type. However, typing `exp` in a formula denotes the product of the three quantities  $e$ ,  $x$ , and  $y$ , which is printed as *exp*. To tell LATEX that you are referring to a named function, you would instead use the `\exp` command.

```
$\exp(x+y)$.
```

|  $\exp(x + y)$ .

A large number of standard mathematical functions are defined as LATEX commands, such as `sin`, `cos`, `tan`, `exp`, `lim`, `max`.

```
$\sin(x)$  
$\cos(\theta)$  
$\exp(ikx)$  
$\lim_{x \rightarrow 0}$  
$f(x)$
```

|  $\sin(x)$   
|  $\cos(\theta)$   
|  $\exp(ikx)$   
|  $\lim_{x \rightarrow 0}$   
|  $f(x)$

If you want to include a function name that does not have a corresponding LATEX command, you can define your own command using `amsmath` and `\DeclareMathOperator`:

```
\usepackage{amsmath}  
\DeclareMathOperator{\lambertW}{LambertW}
```

```
$\phi = \lambertW(x)$
```

|  $\phi = \text{LambertW}(x)$

## 8 Quotes

Printed text distinguishes between opening and closing quote marks “like these” or ‘these’. These are generated using the characters ‘ and ’.

```
' "This is a quote".'
```

```
| "This is a quote".
```

Note the result if you use the quote character ":

```
"This is a quote".
```

```
| "This is a quote".
```

## 9 Automatic sizing of brackets in equations

Use `\left(` and `\right)` for pairs of brackets in equations so that they are automatically resized to fit the enclosed expression.

```
\begin{equation}
\left(\frac{1}{2}\right)
\end{equation}
```

$$\left(\frac{1}{2}\right)$$

not

```
\begin{equation}
(\frac{1}{2})
\end{equation}
```

$$(\frac{1}{2})$$

This also works with other bracket types (note that braces {} need to be escaped using backslashes).

```
$\left[ x+y \right]$ \\
$\left\{ x+y \right\}$ \\
$\left| x+y \right|$ \\
$\left< x+y \right>$
```

[ $x + y$ ]
{ $x + y$ }
$x + y$
$\langle x + y \rangle$

LATEX will check that every `\left` command has a paired `\right` command. If you want unmatched brackets use e.g. `\right.` to complete the pair.

```
$\left(x+y\right.$
```

| ( $x + y$

## 10 Periods / full stops

To help readers distinguish the end of sentences, typesetters traditionally use slightly larger spaces after sentence-ending periods than between the words that make up each sentence. LATEX attempts to follow this convention, and assumes that a period (the character `.`) ends a sentence if it is followed by a space.

One consequence of this is that there are some cases where LATEX will insert an expanded space unnecessarily, for example, after abbreviations.

```
i.e. some other example.
```

| i.e. some other example.

We can tell LATEX to treat this as a normal inter-word space by adding a backslash

```
i.e.\ some other example.
```

| i.e. some other example.

## 11 Non-breaking spaces

Non-breaking spaces prevent text that forms a single object from being split across two lines. These can be indicated in your document using the tilde symbol `~`, e.g. for author initial lists for acknowledgements:

```
B.~J.~M. acknowledges support from $\ldots$
```

| B. J. M. acknowledges support from ...

## 12 Hyphens and dashes

Hyphens and dashes look similar, but they are not interchangeable.<sup>2</sup>

The hyphen: -. This is used in some multipart words, and compound adjectives:

```
Lattice-gas Monte Carlo; non-dilute concentrations;
single-particle description.
```

Lattice-gas Monte Carlo; non-dilute concentrations; single-particle description.

Dashes come in two sizes: the *en dash*: -, and the *em dash*: —.

The en dash: – This is used to indicate a range of values (pages 100–102; Eqns. 3–7.), and to denote a connection or contrast between pairs of words, e.g. lithium–lithium interactions. It can also be used instead of a hyphen in a compound adjective, when one of the component words is already hyphenated e.g. muon-spin-spectroscopy hopping rates.

An en dash is produced in L<sup>A</sup>T<sub>E</sub>X using either \textendash{} or --.

```
lithium--lithium interactions;
muon-spin--spectroscopy hopping rates.
```

lithium–lithium interactions; muon-spin–spectroscopy hopping rates.

The em dash: — This breaks up parts of a sentence. For example, it can be used in place of parentheses where these might break the flow of the text.

An em dash is produced in L<sup>A</sup>T<sub>E</sub>X using either \textemdash{} or ---.

```
The main thing---if you must know---is to use hyphens
properly.
```

The main thing—if you must know—is to use hyphens properly.

Finally, all three of these: the hyphen -, the en dash –, and the em dash —, are distinct from the minus sign –, which is produced inside a math environment using \$-\$.

<sup>2</sup> For a fuller discussion of the different uses of hyphens and dashes see Butterick's Practical Typography: <https://practicaltypography.com/hyphens-and-dashes.html>.

## 13 Punctuating equations

Grammatically you can think of an equation as a single noun. Equations form part of the surrounding text, and require appropriate punctuation; e.g.

```
The equation for a straight line is
\begin{equation}
y=mx+c.
\end{equation}
```

The equation for a straight line is

$$y = mx + c.$$

or

```
The equation for a straight line is
\begin{equation}
y=mx+c,
\end{equation}
where $m$ is the gradient,
and $c$ is the intercept on the $y$ axis.
```

The equation for a straight line is

$$y = mx + c,$$

where  $m$  is the gradient, and  $c$  is the intercept on the  $y$  axis.

## 14 Ellipses

Ellipses are produced using `\ldots`; not three periods.

```
We wrote \$\ldots\$ until we finished.
```

| We wrote ... until we finished. (correct)

```
We wrote ... until we finished.
```

| We wrote ... until we finished. (incorrect)

## 15 Angle brackets

For left and right angle brackets, use `\langle` and `\rangle`, or `\left<` and `\right>`. Don't use `<` and `>`.

```
$\langle x+y \rangle$ \\
$\left\langle x+y \right\rangle$ \\
$\langle x+y \rangle$
```

$\langle x + y \rangle$
$\langle x + y \rangle$
$\langle x + y \rangle$

## 16 Text in math mode

Text in equations that does not refer to variables should be in an upright typeface. This can be produced using `\mathrm{}`.

```
$\Delta E_{\mathrm{site}} = E_{\mathrm{oct}} - E_{\mathrm{tet}}$
```

$\Delta E_{\mathrm{site}} = E_{\mathrm{oct}} - E_{\mathrm{tet}}$
--

## 17 Chemical symbols

Chemical elements should be written in upright text, e.g. Cl, Mg, Li. Chemical formulae should contain elements in upright text, and correct subscripts and superscripts for stoichiometry, charge, etc. Subscripts and superscripts need to go inside math environments to render correctly.

```
H$_2$S0$_4$ and S0$_4^{2-}$.
```

$\text{H}_2\text{SO}_4$ and $\text{SO}_4^{2-}$ .
--

Variables in chemical formulae should be italicised:

```
Li$_x$Ti0$_2$.
```

$\text{Li}_x\text{TiO}_2$ .
-----------------------------

This can be simplified by using the `mhchem` package,<sup>3</sup> by adding `\usepackage{mhchem}` to your document header, and then using the `\ce{}` command.

```
$\ce{H2S04}$ and $\ce{S04^2-}$ \\
$\ce{Li_xTi02}$
```

<sup>3</sup> <https://ctan.org/pkg/mhchem?lang=en>

$\text{H}_2\text{SO}_4$	and	$\text{SO}_4^{2-}$
$\text{Li}_x\text{TiO}_2$		

Notice how the `\ce{}` command tries to intelligently interpret numbers corresponding to stoichiometries and charges.

## 18 Bibliography formatting: a systematic approach

Consistent bibliography formatting is essential for professional scientific documents, yet it is often where typography standards break down. A systematic approach to bibliography formatting provides several key benefits: bibliographies are well-formatted with consistent typography and formatting; .bib data can be reused across papers without reformatting; multiple authors can contribute to paper bibliographies without needing to check formatting conventions in each case; and standardised cite strings allow coauthors to have a better idea of which papers are being cited when they read the source .tex of a paper.

The conventions outlined below represent the preferred approach within our research group. These demonstrate how to handle the complex edge cases that commonly arise in scientific bibliographies, particularly those involving chemical notation.

### Obtaining initial bibliography entries

The starting point for any systematic bibliography formatting workflow is obtaining raw .bib entries. The most reliable source for initial entries is [doi2bib.org](https://doi2bib.org),<sup>4</sup> which converts DOIs directly to .bib format. Simply paste a DOI into the website and receive a properly structured bibliographic entry that can then be formatted according to your conventions.

<sup>4</sup> <https://doi2bib.org>

These raw entries typically require systematic formatting to achieve consistency, as demonstrated in the following sections.

### Citation keys

Citation keys should follow a systematic naming convention that makes entries easily identifiable and sortable. The format depends on the number of authors:

- **Single author:** AuthorJournalYear (e.g., MorganPhysChemChemPhys2024)
- **Two authors:** FirstAuthorAndSecondAuthorJournalYear (e.g., MorganAndWatsonJAmChemSoc2024)
- **Three or more authors:** FirstAuthorEtAlJournalYear (e.g., MorganEtAlPhysChemChemPhys2024)

The journal name component should have all spaces removed and use title case (e.g., `PhysChemChemPhys`, `JAmChemSoc`).

### *Journal names*

Journal names should use standard abbreviations following established conventions. Common examples include:

- Journal of Physical Chemistry → `J. Phys. Chem.`
- Journal of the American Chemical Society → `J. Am. Chem. Soc.`
- Physical Chemistry Chemical Physics → `Phys. Chem. Chem. Phys.`
- Nature Materials → `Nature Mater.`

### *Title formatting*

Titles require careful attention to capitalisation and chemical notation:

**Capitalisation protection:** Use braces {} to protect capitalisation that should be preserved regardless of bibliography style:

- The first word immediately after a colon or em dash
- Acronyms (e.g., `{NMR}`, `{DFT}`, `{XRD}`)
- Chemical element symbols (e.g., `{C}a`, `{S}i`, `{O}`)

**Chemical formulae:** Chemical formulae in titles need special handling to ensure correct typography:

```
CaSiO3 becomes {C}a{S}i{O}$\_3$
```

```
Al2O3 becomes {A}l$_2${O}$\_3$
```

Element symbols have their first letter protected by braces, and stoichiometric coefficients are correctly subscripted using math mode.

**Special characters:** Replace Unicode symbols with appropriate L<sup>A</sup>T<sub>E</sub>X commands:

- $\square \rightarrow \$\backslash square\$$
- $\leq \rightarrow \$\backslash leq\$$
- $\geq \rightarrow \$\backslash geq\$$
- $\rightarrow \rightarrow \$\backslash rightarrow\$$
- ${}^\circ \rightarrow \$^\backslash circ\$$

### *Example transformation*

The following example demonstrates these conventions in practice:

#### **Original entry:**

```
@article{Pedone2010,
    title = {Multinuclear NMR of CaSiO3 glass: simulation
              from first-principles},
    volume = {12},
    ISSN = {1463-9084},
    url = {http://dx.doi.org/10.1039/b924489a},
    DOI = {10.1039/b924489a},
    number = {23},
    journal = {Physical Chemistry Chemical Physics},
    publisher = {Royal Society of Chemistry (RSC)},
    author = {Pedone, Alfonso and Charpentier, Thibault
              and Menziani, Maria Cristina},
    year = {2010},
    pages = {6054}
}
```

#### **Formatted entry:**

```
@article{PedoneEtAl_PhysChemChemPhys2010,
    title = {Multinuclear {NMR} of {C}a{S}i{0}$_3$ Glass:
              {S}imulation from First-Principles},
    volume = {12},
    ISSN = {1463-9084},
    DOI = {10.1039/b924489a},
    number = {23},
    journal = {Phys. Chem. Chem. Phys.},
    publisher = {Royal Society of Chemistry (RSC)},
    author = {Pedone, Alfonso and Charpentier, Thibault
              and Menziani, Maria Cristina},
    year = {2010},
    pages = {6054}
}
```

Note the changes: systematic citation key, protected acronym ({NMR}), chemical formula with protected elements and subscripted stoichiometry ({C}a{S}i{0}\$\_3\$), protected capitalisation after the colon ({S}imulation), abbreviated journal name, and removal of the URL field.

## 19 Vectors

Vector quantities should be typeset in bold. This can be achieved using the `\mathbf{}` command:

```
$\mathbf{x}$
```

$\text{\LaTeX}$  includes a `\vec{}` command, that produces normal weight text with an arrow over variable names:

```
$\vec{x}$
```

The advantage of the command `\vec{}` is that it provides semantic information about a document. It is clear when reading the source file that `\vec{x}` refers to a vector quantity, while `\mathbf{x}` is ambiguous. To keep the semantic benefit of `\vec{}` you can redefine the command to give the alternate bold formatting:

```
\renewcommand{\vec}[1]{\mathbf{#1}}
$\vec{x}_i + \vec{x}_j$
```

$x_i + x_j$

## 20 Kröger-Vink notation

Kröger-Vink notation provides a way to describe defects in crystalline lattices.<sup>5</sup> Defects are denoted by a symbol with a subscript and superscript:  $A_S^c$ . The main symbol,  $A$ , describes the defect species. This is normally a chemical element, or a vacancy. The subscript,  $S$ , describes the site in the crystal lattice. This is normally the formula for the chemical element that would occupy that site in the perfect crystal, or  $i$  for an interstitial. The superscript is the defect charge, defined as the charge *relative* to the formal charge at this site in the perfect crystal. For example, a neutral oxygen vacancy has a charge of 0, which gives a Kröger-Vink charge of +2 relative to a formal  $O^{2-}$  ion. Positive relative charge is denoted by one or more dots,  $\cdot$ . Negative relative charge is denoted by one or more dashes,  $'$ . Zero relative charge is denoted by a cross,  $\times$ . Relative charges larger than  $\pm 1$  are indicated by repeating the appropriate symbol.

Kröger-Vink notation can be generated using the `kroger_vink` package,<sup>6</sup> by including `\usepackage{kroger_vink}` in your document header. This provides a `\kv{A}{S}{c}` command that takes three

<sup>5</sup> F. A. Kröger and H. J. Vink. *Relations Between Concentrations of Imperfections in Crystalline Solids*, volume 3, pages 307–435. Academic Press, New York, 1956; and F. A. Kröger. *The Chemistry of Imperfect Crystals*. North-Holland, Amsterdam, 1964

<sup>6</sup> <https://github.com/bjmorgan/kroger-vink>.

arguments. The first argument is the defect species, the second is the lattice site, and the third is the relative charge.

```
\usepackage{kroger_vink}
\kv{F}{i}{-1} \\
\kv{S}{0}{0} \\
\kv{\vac}{0}{+2}
```

$F'_i$
$S_O^\times$
$V_O^{\bullet\bullet}$

The default notation for a vacancy is an italic capital  $V$ . This can be changed to a lower case  $v$  or an empty square symbol  $\square$  by selecting the corresponding package options.

```
\usepackage[lowercasevac]{kroger_vink}
\kv{\vac}{0}{+2}
```

$v_O^{\bullet\bullet}$
------------------------

```
\usepackage[squarevac]{kroger_vink}
\kv{\vac}{0}{+2}
```

$\square_O^{\bullet\bullet}$
------------------------------

## 21 Further reading

### General typography

- Butterick's Practical Typography<sup>7</sup>

<sup>7</sup> <https://practicaltypography.com>

### LATEX

- LATEX A Document Preparation System
- The `siunitx` package documentation

### References

- [1] F. A. Kröger. *The Chemistry of Imperfect Crystals*. North-Holland, Amsterdam, 1964.
- [2] F. A. Kröger and H. J. Vink. *Relations Between Concentrations of Imperfections in Crystalline Solids*, volume 3, pages 307–435. Academic Press, New York, 1956.