

USEFUL PACKAGES

1 Scientific Typesetting

1.1 SIUNITX

`siunitx` provides many useful commands for typesetting units and numbers both inside and outside math environments. For example:

- Input: `\SI{2E-5}{\eV\per\nm\squared}`
- Output: $2 \times 10^{-5} \text{ eV nm}^{-2}$
- Input: `\num{10.0E-3}`
- Output: 10.0×10^{-3}
- Input: `\ang{30}`
- Output: 30°
- Input: `\SI{200}{\degreeCelsius}`
- Output: 200°C

If a unit is not SI, but you would like to use it with the package you can define custom units by declaring them in the document preamble, as mentioned in the package documentation:

```
\DeclareSIUnit[quantity-product = \,]  
\rpm{\text{rpm}}
```

This can now be used like any other SI unit:

- Input: `\SI{2000}{\rpm}`
- Output: 2000 rpm

1.2 Chemformula

`chemformula` provides simple macros for typesetting chemical formulae:

- Input: `\ch{H2SO4}`
- Output: H_2SO_4

1.3 Miller

`milller` provides excellent macros for typesetting crystal indices:

- Input: `\hkl<10-12>`
- Output: $\langle 10\bar{1}2 \rangle$

All standard closures are supported ($\langle \rangle$, $\{ \}$, $[]$, $()$)

1.4 Listings

`listings` provides support for blocks of syntax-highlighted code within LaTeX documents:

```
1 a = 1  
2 b = 'two'  
3 for i in range:  
4     print(i)
```

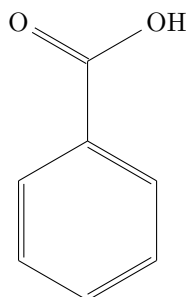
The syntax highlighting is defined in the `codestyling.tex` file, and can be modified as desired. The `codestyling.tex` file must be `input` into the document before it can be used.

2 Graphics and Drawing

2.1 Chemfig

`chemfig` provides rich support for creating nicely programmatically drawn chemical structures in LaTeX documents. The necessary syntax is somewhat confusing, but looking at the documentation is quite helpful.

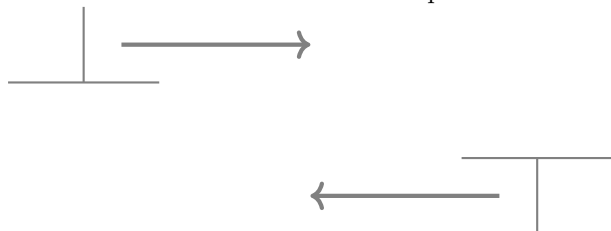
- Input: `\chemfig{*6(==(-([::60]=O)-[::-60]OH)-=)}`



- Output:

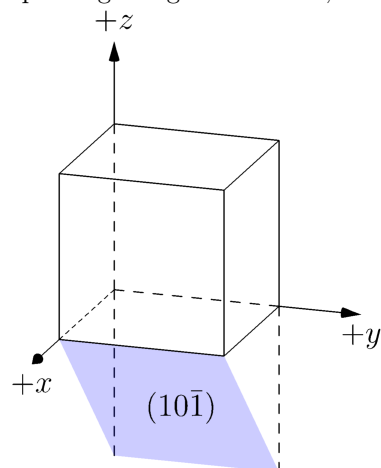
2.2 TikZ

`tikz` is the most commonly used 2D graphics package in the LaTeX world, and can do most things that you would want. It's quite useful for drawing diagrams, and small figures, but it can even be used to plot data from a `csv` file with the correct setup.



2.3 Asymptote

`asymptote` provides both 2D and 3D graphics primitives. It has its own graphics language, so it can be somewhat challenging to get what you want out of it. The build system can also have difficulties with updating things on overleaf, but when you get it working, it's quite capable.



2.4 Subcaption

`subcaption` allows for the creation of subfigures within a document. It's quite useful for prototyping figures and moving them around prior to committing to a final arrangement. Labels can be placed on subcaptions which can then be referenced in the text.

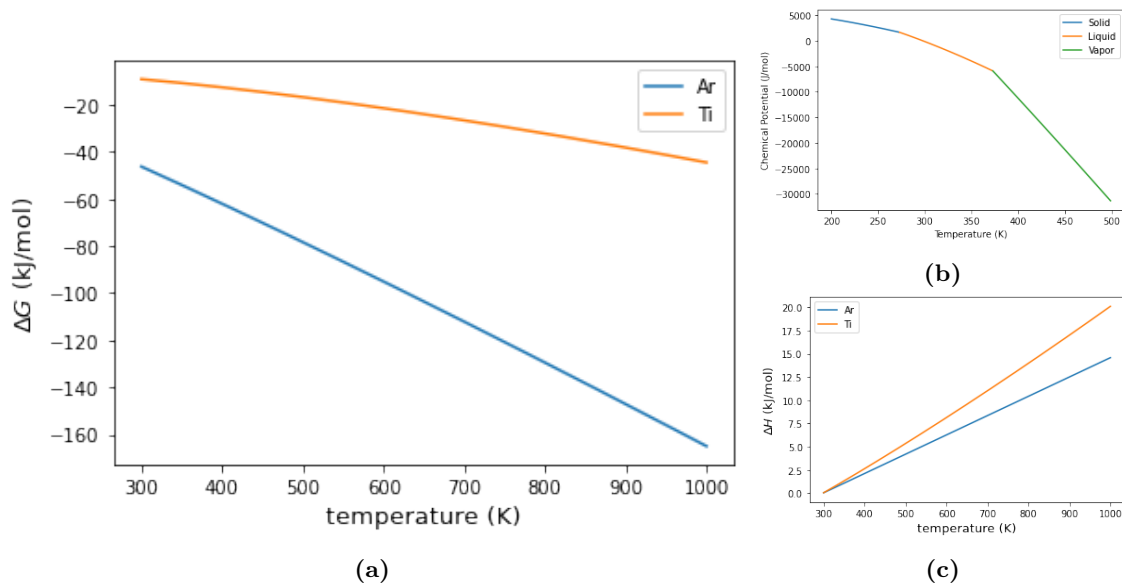


Figure 1: (a) basic demonstration of the `subref` command for a subfigure. (b) The other subfigure. (c) the last subfigure.

This is a somewhat more complicated example, but it shows that you can nest subfigures to obtain quite complicated layouts. The resulting figures can be referenced as usual, e.g. Figure 1b. Using the `ref` command will give a full reference, and using the `subref` command will give only the subreference, e.g. Figure 1 part (b).

2.5 PGFplots and CSVsimple

The combination of `pgfplots` and `csvsimple` allows for the reading of data, and plotting all within the LaTeX environment. For example:

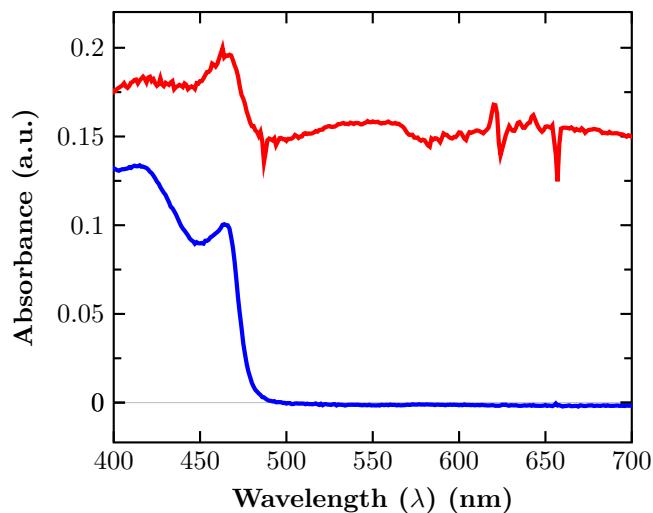


Figure 2: This is a test figure to demonstrate plotting a UV-Vis spectrum from raw data inside the LaTeX environment

Much of the boilerplate can be eliminated by defining a custom style:

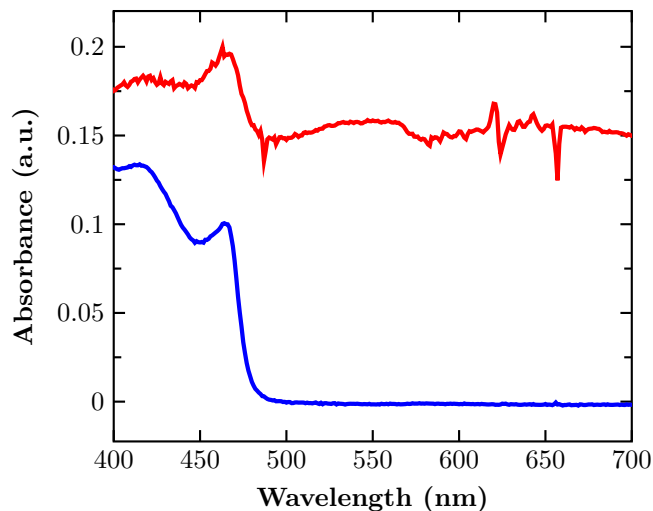


Figure 3: This is another test figure showing the use of a predefined style template to cut down on boilerplate code.

In principle, this approach can also be used to auto-populate tables, but I haven't been able to get it to work within Overleaf.