AtoState User's Guide

Quick Start

Most of the time you will be using the `atostate` function and occasionally the `atoterm` function, install the package and import them with:

```
#import "@preview/atostate:1.0.0": atostate, atoterm
```

If you're going to be use the functions a lot, you probably want shorter names. You can do this by aliasing the functions on import:

```
#import "@preview/atostate:1.0.0": atostate as ats, atoterm as att
```

You can then use them while in math mode:

```
The lowest-energy doubly-excited state of neon is
(Schulz _et al_, Phys. Rev. A, 1996)
$
  #atostate("[He] 2s2.2p4.(3P).3s.(2P).3p:1P*")
  thin,
$
but for readability purposes we shall simply refer
to it as the $#atostate("2p4.3s.3p")$ state.
```

The lowest-energy doubly-excited state of neon is (Schulz et al, Phys. Rev. A, 1996)

$$[\mathrm{He}]\,2s^2\,2p^4\,(^3P)\,3s\,(^2P)\,3p\;{}^1P^\circ,$$

but for readability purposes we shall simply refer to it as the $2p^4\,3s\,3p$ state.

Warning

The functions will work outside of math mode, but the typesetting won't work properly. For example, the n in $2s\,2p^6\,np$ does not render as a variable like it does in $2s\,2p^6\,np$.

Contents

Motivation

Lets imagine that we want to typeset a doubly-excited Rydberg series of neon. You might try something like this:

```
$
  ["He"]
  2s^2 2p^(-2)
  (attach("P", tl: 3))
  3s n p
  attach(P, tl: 1, tr: #sym.circle.tiny)
$
```

$$[\mathrm{He}]2s^2 2p^{-2} (^3\mathrm{P}) 3snp\, ^1P^\circ$$

Already this is horrific to type out. But also, it just looks terrible… At the very least, we need a bit of space:

```
$
  ["He"]
  thick
  2s^2 thin 2p^(-2)
  thin
  (attach(P, tl: 3))
  thin
  3s thin n p
  thick
  attach(P, tl: 1, tr: #sym.circle.tiny)
$
```

$$[\mathrm{He}]\ 2s^2\ 2p^{-2}\ (^3P)\ 3s\ np\ ^1P^\circ$$

Ok, we're getting somewhere. But still, that *np* does *not* look good. We probably want to take a leaf out of calculus' book and typeset azimuths and characters upright:

```
$
  ["He"]
  thick
  2"s"^2 thin 2"p"^(-2)
  thin
  (attach("P", tl: 3))
  thin
  3"s" thin n "p"
  thick
  attach("P", tl: 1, tr: #sym.circle.tiny)
$
```

$$[\mathrm{He}]\ 2\mathrm{s}^2\ 2\mathrm{p}^{-2}\ (^3\mathrm{P})\ 3\mathrm{s}\ n\mathrm{p}\ ^1\mathrm{P}^\circ$$

Right! That looks good. All done then.

What's that? Now you need to typeset the first excited state of xenon? Will this day never end?! *Fine…*

```
$
  ["Kr"]
  thick
  4"d"^10 thin 5"s"^2 thin 5"p"^5
  thin
  (
    attach(
      "P",
      tl: 2,
      tr: #sym.circle.tiny,
      br: sscript(3/2),
    )
  )
  thin
  6"s"
  thick
  attach(
    [3/2],
    tl: 2,
    tr: #sym.circle.tiny,
  )
$
```

$$[\mathrm{Kr}]\ 4\mathrm{d}^{10}\ 5\mathrm{s}^2\ 5\mathrm{p}^5\ \left(^2\mathrm{P}^\circ_{\frac{3}{2}}\right)\ 6\mathrm{s}\ ^2\left[\frac{3}{2}\right]^\circ$$

If you're just typesetting a single state then you can spend a bit of time doing this once and save it in a variable, but if you need to do this a lot then it quickly becomes tedious and error-prone. Fortunately, atostate allows you to do this in a far more convenient and rigorous way!

Usage

Lets first take a look at our previous two examples, to give you a taste for the syntax. The neon example can be typeset like so:

```
$ #atostate("[He] 2s2.2p-2.(3P).3s.np:1P*") $
```

$$[\mathrm{He}]\ 2\mathrm{s}^2\ 2\mathrm{p}^{-2}\ (^3\mathrm{P})\ 3\mathrm{s}\,n\mathrm{p}\ ^1\mathrm{P}^\circ$$

And the xenon example with:

```
$
  #atostate("[Kr] 4d10.5s2.(2P<3/2>*).6s:2[3/2]*")
$
```

$$[\mathrm{Kr}]\ 4\mathrm{d}^{10}\ 5\mathrm{s}^2\ \left(^2\mathrm{P}^\circ_{\frac{3}{2}}\right)\ 6\mathrm{s}\ ^2\left[\frac{3}{2}\right]^\circ$$

These examples should cover most of the syntax you are likely to need day-to-day. Both examples start with an *optional* element indicated by square brackets (`[]`), those are the `[He]` and `[Kr]` parts. Then there's a configuration list, separated by periods (`.`), which can be interspersed with term symbols by using round brackets (`()`). Finally there's an *optional* term symbol, the presence of which is indicated by a colon (`:`).

> **Note**
> The periods and colons are there to make parsing easier. It's *possible* that a future version with a better parser might make them obsolete, though that would probably require an updated syntax.

In general whitespace is ignored, as long as the whitespace is not in the middle of what should conceptually be a single token. So you can have this:

```
$ #atostate("[ He ] 2 s -2 . 3 p") $
```

$$[\mathrm{He}]\ 2\mathrm{s}^{-2}\ 3\mathrm{p}$$

But you can't have whitespace in the middle of the element:

```
$ #atostate("[H e] 2s-2.3p") $
```

Or the middle of a number:

```
$ #atostate("4d1 0") $
$ #atostate("2s- 2") $
```

`atostate` can be used inline and will still look pretty good—$2\mathrm{p}^4\ 3\mathrm{s}\ 3\mathrm{p}$—though particularly complex examples—$[\mathrm{Kr}]\ 4\mathrm{d}^{10}\ 5\mathrm{s}^2\ \left({}^2\mathrm{P}^\circ_{\frac{3}{2}}\right)\ 6\mathrm{s}\ {}^2\left[\frac{3}{2}\right]^\circ$—can look a bit funky when embedded in lots of text. In these cases I suggest you display mode instead.

Configuration Lists

Configuration lists consist of a list of orbitals and terms, separated by periods (`.`).

Each orbital **must** have a principal number and an azimuth. The principal number can be a multi-digit positive number or a single latin letter—which will be typeset as a variable[1]:

```
$ #atostate("10s.np") $
```

$$10\mathrm{s}\ n\mathrm{p}$$

And the azimuth can be a single letter—which will be typeset as text, like the 'd' in $\mathrm{d}x$—or a number or fraction surrounded by square brackets (`[]`):

```
$ #atostate("1s.2[11].3[3/2]") $
```

$$1\mathrm{s}\ 2[11]\ 3[3/2]$$

Note

Fractional azimuths are typeset horizontally to distinguish them from (the vastly more common) fractional characters in the term symbol.

An orbital can also *optionally* have an occupation number and/or a a subscript. The occupation number can be any integer number, positive or negative, and does not need any special markup:

```
$ #atostate("3d10.4s-2.4p6") $
```

$$3\mathrm{d}^{10}\ 4\mathrm{s}^{-2}\ 4\mathrm{p}^6$$

The subscript, a number or fraction, is surrounded by angle brackets (`<>`) and **must** come before the occupation number if it is present:

```
$ #atostate("2s<1>.2p<3/2>4") $
```

$$2\mathrm{s}_1\ 2\mathrm{p}^4_{\frac{3}{2}}$$

The occupation number will not be displayed if it is 1, regardless of whether you enter it or not:

```
$ #atostate("2s1.2p6.3p1") $
```

[1] Symbols, such as greek letters, are planned but require an update to the parser.

$$2s\,2p^6\,3p$$

However, you can force it to always be displayed by passing `display-single-occupation: true` to the `atostate` function:

```
$
  #atostate(
    "2s.2p6.3p",
    display-single-occupation: true,
  )
$
```

$$2s^1\,2p^6\,3p^1$$

See the section about configuring the package on page 6 for an easier way of configuring this.

Term symbols within a configuration list **must** be marked with round brackets (`()`) and must still be separated from surrounding orbitals by periods (`.`):

```
$ #atostate("2p4.(3P).3p1") $
```

$$2p^4\,(^3P)\,3p$$

Term Symbols

Trailing term symbols and term symbols within configuration lists both use the same syntax, with the only difference being that trailing term symbols must be preceded by a colon (`:`) whereas a term symbol in a configuration list must be surrounded by round brackets (`()`).

Term symbols **must** have a multiplicity and a character. The multiplicity is a multi-digit positive integer. The character can either be a single letter—which is typeset as text—or a number or fraction surrounded by square brackets:

```
$ #atostate("1s2.(1S*).2s2.(2[10]).2p6:10[3/2]") $
```

$$1s^2\,(^1S^\circ)\,2s^2\,(^2[10])\,2p^6\;{}^{10}\left[\frac{3}{2}\right]$$

A term symbol can *optionally* indicate odd parity with a trailing asterisk (`*`):

```
$ #atostate("2s.2p6.3p:1P*") $
```

$$2s\,2p^6\,3p\;{}^1P^\circ$$

Term symbols can also, *optionally*, indicate a J-value with a subscript. Like for orbitals, subscripts for term symbols are surrounded by angle brackets (`<>`) and can either be positive integers or fractions:

```
$ #atostate("4d10.(2[3]<1>).5p:2[3/2]<3/2>*") $
```

$$4d^{10}\,(^{2}[3]_{1})\,5p\;{}^{2}\left[\frac{3}{2}\right]^{\circ}_{\frac{3}{2}}$$

If both a parity marker and a J-value are present, the subscript **must** come first.

You can use a different symbol as a parity marker using the `parity-marker` argument:

```
$ #atostate("2s.2p6.3p:1P*", parity-marker: "#") $
```

$$2s\,2p^{6}\,3p\;{}^{1}P^{\#}$$

atoterm

For parsing reasons, the `atostate` function can't typeset just a term symbol[2]. If you need this you can instead use the `atoterm` function:

```
$ #atoterm("1P*") $
```

$$^{1}P^{\circ}$$

The syntax for this function is identical to the syntax for term symbols in the `atostate` function. You can't precede the term symbol by a colon (`:`) or surround it by round brackets (`()`), however.

Package Configuration

If you want to use different default configuration values, I suggest using the `with` method to override them:

```
#import "@preview/atostate:1.0.0": atostate

#let ats = atostate.with(parity-marker: "#")
#let ats-single = atostate.with(display-single-occupation: true)
```

API

AtoState

- atostate()
- atoterm()

atostate

Parse and render an atomic state.

```
$ #atostate("[He] 2s2.2p4.(3P).3s.(2P).np:1P*") $
```

$$[He]\;2s^{2}\,2p^{4}\,(^{3}P)\,3s\,(^{2}P)\,np\;{}^{1}P^{\circ}$$

```
$
  #atostate(
    "[Kr] 4d10.5s2.(2P<3/2>*).6s:2[3/2]*",
    parity-marker: circle(
      radius: 0.2em,
      fill: red,
    ),
    display-single-occupation: true,
  )
$
```

$$[Kr]\;4d^{10}\,5s^{2}\left(^{2}P^{\bullet}_{\frac{3}{2}}\right)\,6s^{1}\;{}^{2}\left[\frac{3}{2}\right]^{\bullet}$$

[2] This restriction may be lifted in the future.

Parameters

```
atostate(
  state: str,
  parity-marker: str content,
  display-single-occupation: bool
) -> content
```

state `str`

The atomic state to render.

parity-marker `str` or `content`

The symbol to use to indicate that a term has odd parity.

Default: `sym.circle.tiny`

display-single-occupation `bool`

Whether to display the occupation number of an orbital that only has one electron.

Default: `false`

atoterm

Parse and render a single term symbol.

```
$ #atoterm("2P<3/2>*") $
```

$ {}^{2}\mathrm{P}^{\circ}_{\frac{3}{2}} $

```
$
  #atoterm(
    "2[3/2]*",
    parity-marker: "#",
  )
$
```

$ {}^{2}\left[\frac{3}{2}\right]^{\#} $

Parameters

```
atoterm(
  term: str,
  parity-marker: str content
) -> content
```

term `str`

The term symbol to render.

parity-marker `str` or `content`

The symbol to use to indicate that the term has odd parity.

Default: `sym.circle.tiny`