BOHR

v0.4 2013/11/10

simple atom representation according to the Bohr model

Clemens Niederberger

https://bitbucket.org/cgnieder/bohr/

contact@mychemistry.eu

This package provides means for the creation of simple Bohr models of atoms up to the atomic number 112. Additionally commands are provided to convert atomic numbers to element symbols or element names and vice versa.

The package is inspired by a question on http://tex.stackexchange.com/: Draw Bohr atomic model with electron shells in TEX?

Table of Contents

1 Licence and Requirements

Permission is granted to copy, distribute and/or modify this software under the terms of the LATEX Project Public License (LPPL), version 1.3 or later (http://www.latex-project.org/lppl.txt). The software has the status "maintained."

The BOHR package loads and needs the packages pgf[1] [Tan10] and cnltx-base[2] [Nie13].

2 Options

Every option described in the manual can also be used as package option. Options are indicated as option and are all key/value like options. Some options can be set without value, too. Then

1. on CTAN as pgf: http://mirrors.ctan.org/graphics/pgf/
2. on CTAN as cnltx: http://mirrors.ctan.org/macros/latex/contrib/cnltx/

the underlined value is used.

3 Usage

BOHR is used like any other LaTeX 2_ε package:

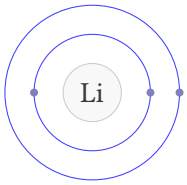
```
\usepackage{bohr}
```

The main command, \bohr, creates the models:

\bohr[⟨*num of shells*⟩]{⟨*num of electrons*⟩}{⟨*atom name*⟩}
The main command. The mandatory arguments take the number of electrons to be printed and the atom symbol that is printed in the center.

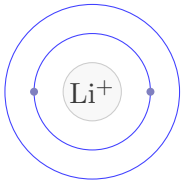
This is described best by an example:

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



There is not much more to it. Another example using the optional argument:

```
\bohr[2]{2}{$\mathrm{Li^+}$}
```



4 Customization

BOHR provides a handful of options to customize the appearance:

\setbohr{⟨*options*⟩}
Options are set in a key/value syntax using this command.

insert-symbol = true|false Default: false
If set to true BOHR will insert the atomic symbol suiting to the given electron number if *no* third argument is given.

insert-number = true|false Default: false
If set to true BOHR will use the appropriate number of electrons for the given element symbol in the third argument if *no* second argument is given. This of course only works if the third argument is one of the 112 element symbols.

`insert-missing` = `true`|`false` Default: `false`
Sets both `insert-symbol` and `insert-number`.

`atom-style` = {⟨*code*⟩} (initially empty)
This code will be placed immediatly before the third argument of `\bohr`. The last macro in it may need one argument.

`name-options-set` = {⟨*tikz option*⟩} (initially empty)
This value is passed to the options of the `\node` the third argument of `\bohr` is placed in.

`name-options-add` = {⟨*tikz options*⟩} (initially empty)
This value will be added to options set with `name-options-set`.

`nucleus-options-set` = {⟨*tikz options*⟩}
Default: `draw=black!80,fill=black!10,opacity=.25`
This value is passed to the options of the `\draw` command that draws the circle around the name-node.

`nucleus-options-add` = {⟨*tikz options*⟩} (initially empty)
This value will be added to options set with `nucleus-options-set`.

`nucleus-radius` = {⟨*dimension*⟩} Default: `1em`
The radius of the circle around the name-node.

`electron-options-set` = {⟨*tikz options*⟩} Default: `blue!50!black!50`
This value is passed to the options of the `\fill` command that draws the electrons.

`electron-options-add` = {⟨*tikz options*⟩} (initially empty)
This value will be added to options set with `electron-options-set`.

`electron-radius` = {⟨*dimension*⟩} Default: `1.5pt`
The radius of the circles that represent the electrons.

`shell-options-set` = {⟨*tikz options*⟩} Default: `draw=blue!75,thin`
This value is passed to the options of the `\draw` command that draws the circles that represent the shells.

`shell-options-add` = {⟨*tikz options*⟩} (initially empty)
This value will be added to options set with `shell-options-set`.

`shell-dist` = {⟨*dimension*⟩} Default: `1em`
The distance between the nucleus and the first shell and between subsequent shells.

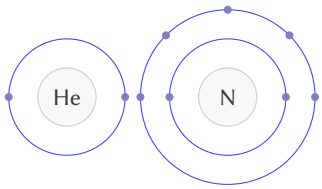
`language` = {⟨*language*⟩} (initially empty)
Introduced in version 0.2d
Select the language used for the element names manually rather than letting BOHR select it automatically. This option must be used as package option or in the preamble in order to have an effect. Currently provided languages are `English`, `French` and `German`. If this option is not used the document language at the end of the preamble (as set by babel or polyglossia) is used.

`distribution-method` = `periodic|quantum` Default: `quantum`

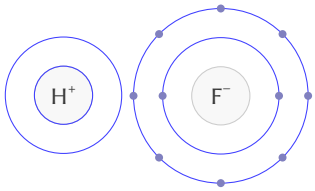
Introduced in version 0.3

Determines how the electrons are distributed on the shells. `periodic` distributes the electrons 2-8-8-18-18-32-32, *i. e.*, according to the place of the corresponding atom in the periodic table of elements. `quantum` distributes the electrons according the the electron configuration of the corresponding atom where each shell represents the main quantum number. Pd for example has the configuration $1s^2 2s^2 2p^6 3s^2 3p^6 3d^{10} 4s^2 4p^6 4d^{10}$ and would get two electrons on the first shell, 8 electrons on the second, and 18 electrons each on the third and fourth.

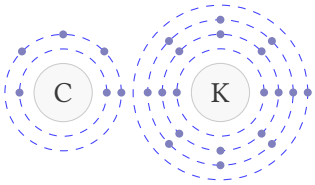
```
\setbohr{name-options-set={font=\footnotesize\sffamily}}
\bohr{2}{He} \bohr{7}{N}
```




```
% uses package 'chemmacros'
\setbohr{atom-style={\footnotesize\sffamily\ch}}
\bohr{0}{H+} \bohr{10}{F-}
```




```
\setbohr{
  shell-options-add = dashed,
  shell-dist        = .5em,
  insert-missing
}
\bohr{6}{} \bohr{}{K}
```



5 Additional Commands

BOHR provides some additional commands that return the element symbol or the element name to a given atomic number and vice versa.

`\elementsymbol{`⟨*atomic number*⟩`|`⟨*element name*⟩`}`
Returns the element symbol for a given atomic number or element name.
`\elementsymbol{80}`: Hg;
`\elementsymbol{rhenium}`: Rh.
The symbols are printed according to `atom-style`.

`\elementname{`⟨*atomic number*⟩`|`⟨*element symbol*⟩`}`
Returns the element name for a given atomic number or element symbol.
`\elementname{80}`: Mercury;
`\elementname{Rh}`: Rhenium.

`\atomicnumber{`⟨*element name*⟩`|`⟨*element symbol*⟩`}`
Returns the atomic number for a given element name or element symbol.
`\atomicnumber{Hg}`: 80;
`\atomicnumber{rhenium}` 75.

`\Z`
If this command isn't defined at begin document it is available as an alias of `\atomicnumber`.

`\elconf{`⟨*atomic number*⟩`|`⟨*element symbol*⟩`|`⟨*element name*⟩`}`
Introduced in version 0.4
Prints the electron configuration of the given element.
`\elconf{Cu}`: $1s^2 2s^2 2p^6 3s^2 3p^6 3d^{10} 4s^1$;
`\elconf{copper}`: $1s^2 2s^2 2p^6 3s^2 3p^6 3d^{10} 4s^1$;
`\elconf{29}`: $1s^2 2s^2 2p^6 3s^2 3p^6 3d^{10} 4s^1$.
Currently there is no way to get the shortened version (*e. g.* $[\mathrm{Ar}] 3d^{10} 4s^1$ for Copper).

`\writeelconf{`⟨*electron distribution*⟩`}`
Introduced in version 0.4
Writes the electron distribution ⟨*electron distribution*⟩. The input is the same as described below for `\setelectrondistribution`.
`\writeelconf{2,2+6,2+6+7}`: $1s^2 2s^2 2p^6 3s^2 3p^6 3d^7$.

`\setatomsymbol{`⟨*atomic number*⟩`}{`⟨*atom symbol*⟩`}`
Introduced in version 0.4
This sets the symbol associated with the atom number ⟨*atomic number*⟩.

`\setatomname{`⟨*atomic number*⟩`}{`⟨*element name*⟩`}`
Introduced in version 0.4
This sets the element name associated with the atom number ⟨*atomic number*⟩.

`\setelectrondistribution{`⟨*atomic number*⟩`}{`⟨*electron distribution*⟩`}`
Introduced in version 0.4
This set the electron distribution associated with the atom number ⟨*atomic number*⟩. ⟨*electron distribution*⟩ is a comma-separated list of the number of electrons placed on each shell from inner to outer shell. For example `\setelectrondistribution{3}{2,0,1}` would be an excited Lithium. The distribution thus declared is only used for drawing when `distribution-method = quantum` but is also used by `\elconf`. The number of electrons with the same principal quantum number but a different angular quantum number are separated with a `+` ordered by the angular quantum number, *i. e.*, first s, then p, then d, and then f. Copper's distribution would be declared like this: `\setelectrondistribution{29}{2,2+6,2+6+10,1}`.

A declaration with `\setelectrondistribution{29}{2,8,18,1}` would work but then `\elconf{29}` would give the wrong results.

```
1 The elements \elementname{F}, \elementname{Cl}, \elementname{Br},
2 \elementname{I} and \elementname{At} are called \emph{halogens}.
```

The elements Fluorine, Chlorine, Bromine, Iodine and Astatine are called *halogens*.

6 Internal Commands

The element properties used by BOHR are defined through the following commands:

`\DeclareAtomSymbol{`⟨*atomic number*⟩`}{`⟨*atom symbol*⟩`}`
The package file contains 112 lines like the following: `\DeclareAtomSymbol{29}{Cu}`. They associate atomic number with the corresponding atom symbol. This is the internal equivalent to `\setatomsymbol`.

`\DeclareAtomName{`⟨*atomic number*⟩`}{`⟨*element name*⟩`}`
BOHR comes with a few language files named `bohr_elements_`⟨*language*⟩`.def`. They contain of 112 entries like `\DeclareAtomName{29}{Copper}` which associate atomic number and element name. This is the internal equivalent to `\setatomname`.

`\DeclareElectronDistribution{`⟨*atomic number*⟩`}{`⟨*electron distribution*⟩`}`
The ⟨*electron distribution*⟩ is a comma separated list of integers which determines how the electrons are distributed on the shells when `distribution-method = quantum`. The package file contains 112 entries like `\DeclareElectronDistribution{29}{2,2+6,2+6+10,1}`. This is the internal equivalent to `\setelectrondistribution`.

References

[Nie13] Clemens Niederberger. cnltx. version 0.7a, Oct. 4, 2013.
URL: http://mirror.ctan.org/macros/latex/contrib/cnltx/.

[Tan10] Till Tantau. TikZ/pgf. version 2.10, Oct. 25, 2010.
URL: http://mirror.ctan.org/graphics/pgf/.

Index