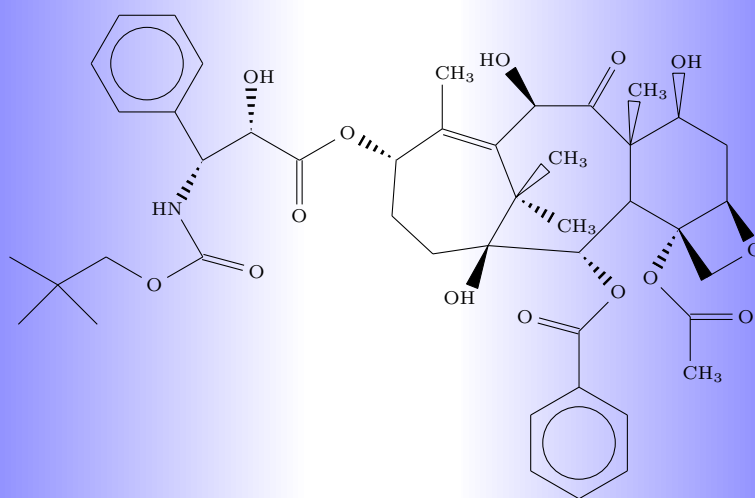


ChemFig

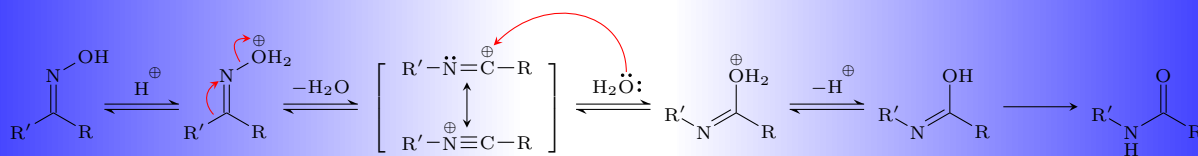
v0.4

March 7, 2011
Christian Tellechea

A \TeX package for drawing molecules



Taxotere



The Beckmann rearrangement

Contents

I	Introduction	3
1	New in version 0.3	3
2	Foreword	3
3	Presenting ChemFig	4
II	ChemFig for the impatient	5
1	Syntax	5
2	The different types of bonds	5
3	Different types of diagrams	6
3.1	Complete structural diagram	6
3.2	Condensed structural diagram	6
3.3	Cram representation	7
3.4	Skeleton diagram	7
3.5	Lewis diagrams	7
4	Branched molecules	7
5	Rings	8
6	Ions	9
7	Chemical equations	9
III	Operation of ChemFig	10
1	Groups of atoms	10
2	Different types of bonds	10
3	Bond angle	11
3.1	Predefined angles	12
3.2	Absolute angles	12
3.3	Relative angles	12
4	Length of a bond	13
5	Departure and arrival atoms	13
6	Customization of bonds	14
7	Default values	14
8	Branches	15
8.1	Principle	15
8.2	Nesting	16
8.3	Method	16
9	Connecting distant atoms	17

10 Rings	18
10.1 Syntax	18
10.2 Angular position	19
10.2.1 At the start	19
10.2.2 After a bond	20
10.3 Branches on a ring	20
10.4 Nested rings	21
10.5 Rings and groups of atoms	22
11 Representing electron movements	22
11.1 Mesomeric effects	23
11.2 Reaction mechanisms	24
12 Writing a name under a molecule	25
IV Advanced usage	27
1 Separating atoms	27
2 Displaying atoms	27
3 Optional arguments	28
4 Vertical alignment	29
5 Shifted double bonds	30
6 Delocalized double bonds	31
7 Saving a sub-molecule	31
8 Decorations	32
8.1 Lewis diagrams	32
8.2 Stacking characters	33
8.3 Chemical reactions	34
9 Using <code>\chemfig</code> in the <code>tikzpicture</code> environment	35
10 Beyond chemistry	35
11 Annotated examples	36
11.1 Ethanal	37
11.2 2-amino-4-oxohexanoic acid	37
11.2.1 Absolute angles	37
11.2.2 Relative angles	38
11.2.3 Ring	38
11.2.4 Nested rings	38
11.3 Glucose	39
11.3.1 Skeleton diagram	39
11.3.2 Fisher projection	39
11.3.3 "Chair" representation	40
11.3.4 Haworth projection	41
11.4 Adrenaline	41
11.4.1 Using one ring	41
11.4.2 Using two rings	42
11.5 Guanine	43

12 How to ...	44
12.1 Write a colored atom	44
12.2 Add a superscript without modifying a bond	45
12.3 Draw a curve bond	45
12.4 Modify the size of a molecule	46
12.5 Draw a ploymer element	46
12.6 Draw the symmetrical of a molecule	47
12.7 Add text above bonds and arc to angles	47
13 List of commands	48
V Gallery	50
Index	62

PART I

Introduction

1 New in version 0.3

Here's what's new in version 0.4:

1. The code is now written in plain $\epsilon\text{T}_{\text{E}}\text{X}$. Therefore, the package may be used with $\text{E}_{\text{T}}\text{E}_{\text{X}}$ or $\text{C}_{\text{O}}\text{N}_{\text{T}}\text{E}_{\text{X}}\text{T}$.
2. Bugfixes.

2 Foreword

This package has seen the light of day thanks to the assistance of Christophe CASSEAU, who had the idea after being confronted with the complexity of the syntax of the `ppctex` package.

Throughout the writing of the code, he helped me find interesting features. He always encouraged me to write more advanced features even though I was sometimes (nearly always?) reluctant; if `ChemFig` has the features it does, it is in large part thanks to him. I thank him as well for his testing of beta versions of this package, and for his contributions to the writing of this manual.

Experience shows that it is has been difficult to combine drawing of molecules with the typographic quality of a program like $\text{E}_{\text{T}}\text{E}_{\text{X}}$, leaving little choice for the user who wishes to have a vector format for these drawings. After having abandoned the `ppctex` package (developed for $\text{c}_{\text{O}}\text{n}_{\text{T}}\text{E}_{\text{X}}\text{T}$ and available under $\text{E}_{\text{T}}\text{E}_{\text{X}}$) because of the complexity of its syntax, I turned to the world of programs outside $\text{E}_{\text{T}}\text{E}_{\text{X}}$. The difficulty in this case is finding a compromise between quality and price. After many unsuccessful attempts I found the only option was a new package, and I would like to thank Christian TELLECHEA for bringing it to life. To meet my requirements, `ChemFig` needed to be easy to use but still have advanced features, something well nigh impossible. Yet he was able to put together a very flexible $\text{T}_{\text{E}}\text{X}$ code which

makes it a pleasure for me to write my molecules. I hope it will be the same for you readers looking for a package useful in the field of chemistry.

Christophe CASSEAU

Finally, I wish to warmly thank Theo HOPMAN for offering to translate this manual into English.

3 Presenting ChemFig

To use this package, start by adding the following code to the preamble:

- `\input chemfig.tex` with $\epsilon\text{T}_{\text{E}}\text{X}$;
- `\usepackage{chemfig}` with $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$;
- `\usemodule[chemfig]` with $\text{ConT}_{\text{E}}\text{Xt}$.

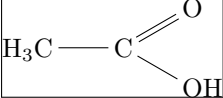
In all cases, the `tikz` package, if not loaded before, is loaded by `ChemFig`.

The most important command for drawing molecules is `\chemfig{<code>}`. The argument `code` is a set of characters describing the structure of the molecule according to the rules which are described in this manual.

Care has been taken to make it possible to draw the greatest possible number of molecular configurations, while maintaining a simple, flexible, and intuitive syntax. Despite this, the `<code>` which describes the 2D structure of the molecule increases in complexity in proportion to that of the molecule being drawn.

The command `\chemfig` draws a molecule using the commands provided by the `tikz` package, placed inside a `tikzpicture` environment. The choice of `tikz` implies that:

- the user has a choice of compilation method: $\text{pdf}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ can be used equally well in dvi mode ($\text{tex} \rightarrow \text{dvi} \rightarrow \text{ps} \rightarrow \text{pdf}$) or in pdf mode ($\text{tex} \rightarrow \text{pdf}$). In effect `tikz`, via the underlying `pgf`, gives identical graphical results in the two modes;
- the bounding box is automatically calculated by `tikz` and the user need not worry about any overlap with the text. However, care must be taken with alignment when the molecule is drawn in a paragraph. In the

following example, we have drawn the bounding box for the molecule: . `ChemFig` always places the first atom of the molecule on the baseline of the preceding code.

PART II

ChemFig for the impatient

This part is a non-exhaustive overview of the features of `ChemFig`. The goal is to introduce the basic ideas, allowing the user to get started drawing molecules as quickly as possible. This part does not go into detail; advanced use and a more formal approach to `ChemFig` commands will be discussed in the following parts.

1 Syntax

The command `\chemfig` is used in the following way:

```
\chemfig{<atom1><bond type>[<angle>,<coeff>,<n1>,<n2>,<tikz code>]<atom2>}
```

- `<angle>` is the bond angle between the two atoms;
- `<coeff>` is a coefficient multiplying the default bond length;
- `<n1>` and `<n2>` are the numbers of the departure and arrival atoms of the bond;
- `<tikz code>` is additional options concerning the colour or style of a bond.

Each bond takes optional arguments which are placed in square brackets. These arguments can adjust everything one needs for the bond. Each argument has a default value, so one can simply write:

<code>\chemfig{H-O-H}</code>	The water molecule	H — O — H
------------------------------	---------------------------	-----------

In all the examples, the grey line represents the baseline.

2 The different types of bonds

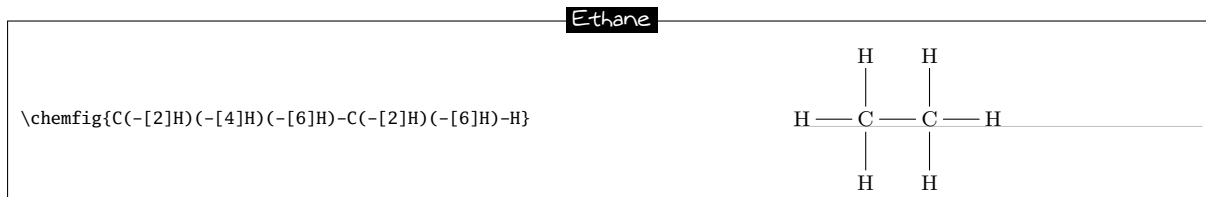
For `ChemFig`, bonds between two atoms are one of nine types, represented by the characters $\rule{0.5ex}{0.4pt}$, $\rule{0.5ex}{0.8pt}$, $\rule{0.5ex}{1.2pt}$, $\rule{0.5ex}{1.6pt}$, $\rule{0.5ex}{2pt}$, $\rule{0.5ex}{2.4pt}$, $\rule{0.5ex}{2.8pt}$, $\rule{0.5ex}{3.2pt}$ and $\rule{0.5ex}{3.6pt}$:

Bond #	Code	Result	Bond type
1	<code>\chemfig{A-B}</code>	A — B	Single
2	<code>\chemfig{A=B}</code>	A = B	Double
3	<code>\chemfig{A~B}</code>	A ≡ B	Triple
4	<code>\chemfig{A>B}</code>	A ► B	right Cram, plain
5	<code>\chemfig{A<B}</code>	A ◄ B	left Cram, plain
6	<code>\chemfig{A>:B}</code>	A ... B	right Cram, dashed
7	<code>\chemfig{A<:B}</code>	A ··· B	left Cram, dashed
8	<code>\chemfig{A> B}</code>	A ▷ B	right Cram, hollow
9	<code>\chemfig{A< B}</code>	A ◁ B	left Cram, hollow

The command `\setdoublesep{<dim>}` adjusts the spacing between the lines in double or triple bonds. This spacing is 2pt by default.

3 Different types of diagrams

3.1 Complete structural diagram

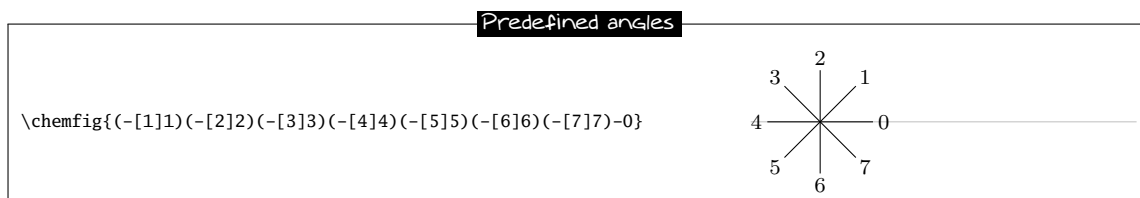


Each bond takes several optional arguments. The first optional argument defines the `<angle>` of the bond. Angles increase counterclockwise. If no angle is give then its default value is 0° .

Note: the parentheses allow multiple bonds from the same atom; see “Branched molecules”, page 7.

There are many ways of specifying the angle of a bond.

Predefined angles When the optional argument contains a whole number, this represents the angle that the bond makes with the horizontal, in multiples of 45° .



Absolute angle To give an angle in degrees relative to the horizontal, the optional argument must take this form: `[: <absolute angle>]`. The `<absolute angle>` may be positive or negative, and may have decimal places. It is reduced to the interval $[0, 360)$.

Relative angle It is often useful to give the bond angle relative to the preceding bond. In this case, the following syntax must be used: `[: : <relative angle>]`. The `<relative angle>` may be positive or negative, and may have decimal places.

3.2 Condensed structural diagram



It is sometimes useful to change the length of a bond. In the preceding example, the first two bonds need to be lengthened; they are too short compared to the double bond. To do this, the `<coeff>` is needed in the optional arguments.

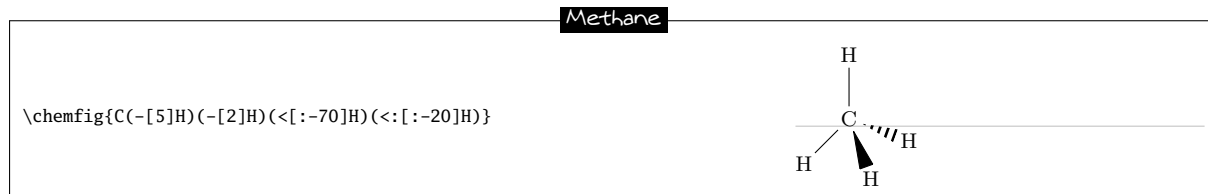


Notes:

- in the notation `[, 1.5]`, the comma indicates that the value placed in brackets corresponds to the second argument (`coeff`). To give a value for the fourth argument one would write `[, , , 2]`;
- characters between braces are not interpreted by `ChemFig`, which allows (for example) groupings of atoms to be written inside parentheses without having them treated as part of a branched molecule (see page 7).

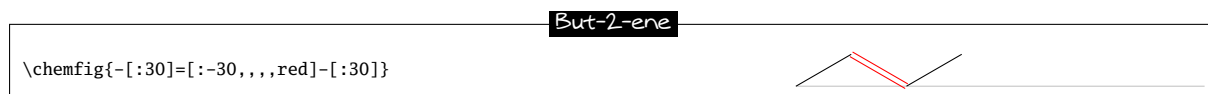
3.3 Cram representation

Another way of showing the bond angle between two atoms.

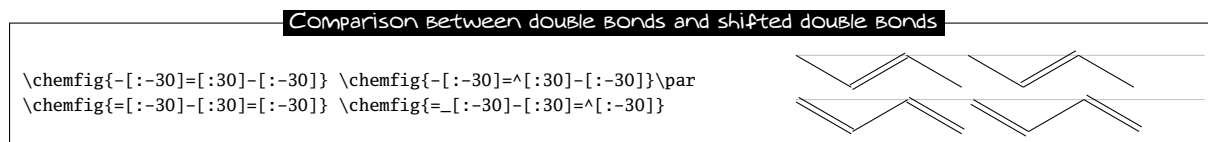


3.4 Skeleton diagram

The simplest input, only covalent bonds are listed with their possible settings.



The double bond lines are drawn on either side of where the single bond line would be. To keep graphical consistency between skeleton diagrams of various compounds it is useful to shift the second double bond line above or below the single bond line. Just follow the = symbol with ^ or _ , as shown below:

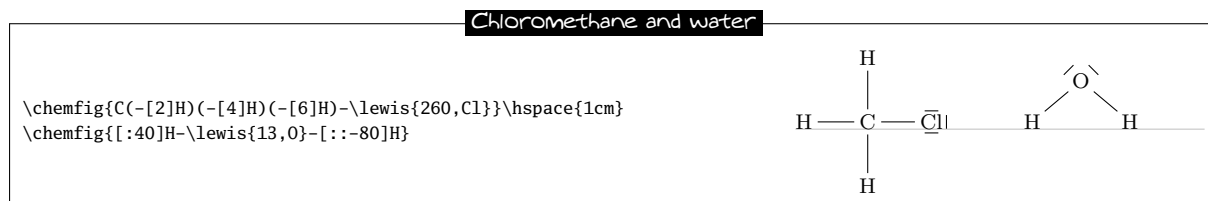


3.5 Lewis diagrams

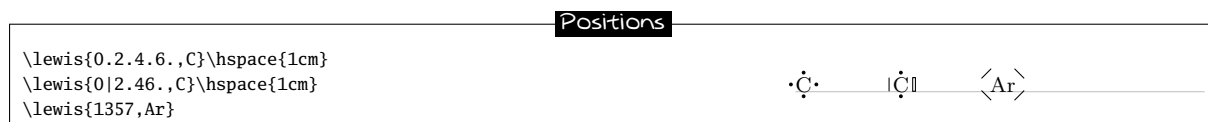
The syntax is as follows:

```
\lewis{<position index><electron state>,<atom>}
```

The electron states can be a lone pair, an empty electron slot, or a single electron. By default the electron state is a lone pair. A single electron is represented by the character \cdot and an empty slot by \square .



Note: the positions take whole number values between 0 and 7.

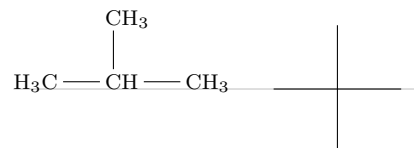


4 Branched molecules

To indicate a branch, simply follow the atom holding the branch with a <code> in parentheses. This <code> is the code of the submolecule which will be attached to the atom. Multiple branches can be attached to the same atom, and these can be nested.

Alkanes

```
\chemfig{H_3C-CH(-[2]CH_3)-CH_3}
\hspace{.5cm}\chemfig{-(-[2])(-[6])-}
```



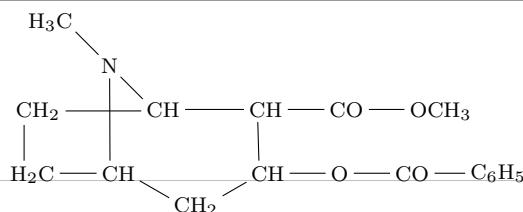
In this type of representation it is sometimes necessary to connect to distant atoms. To do this, the character ? is used, which creates a hook between two atoms. The function

```
?[<name>, <bond>, <tikz>]
```

takes three arguments: the name of the hook, the bond type, and tikz code. The following example shows how to make two different hooks.

Cocaine

```
\chemfig{H_2C(-C?[a]H-[:30]CH_2-[:30]C?[b]H-O-CO-C_6H_5)
-[2]CH_2-[,1.7]CH(-[3]N?[a]-[3]H_3C)(-[,1.35]C?[b]H-CO-OCH_3)}
```



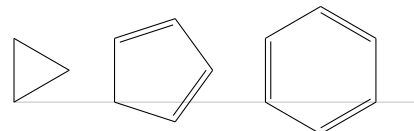
5 Rings

ChemFig can easily draw regular polygons. The syntax is the following:

```
\chemfig{*n(<code for the molecule>)}
```

Some rings

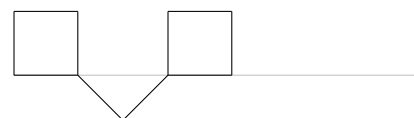
```
\chemfig{*3(---)}\hspace{.5cm}
\chemfig{*5(-----)}\hspace{.5cm}
\chemfig{*6(-----)}
```



Branches are used in the same way as before.

Rings and Branches

```
\chemfig{*4(---[1]*4(-----))---}
```



Note: a ring does not start or finish with the atom or group of atoms with which one wants to close the ring. The chemical entity on which the ring is based must be outside the ring definition.

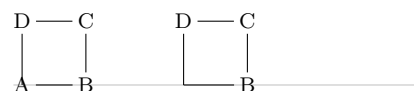
Proper coding

```
\chemfig{A*4(-B-C-D-)}
```



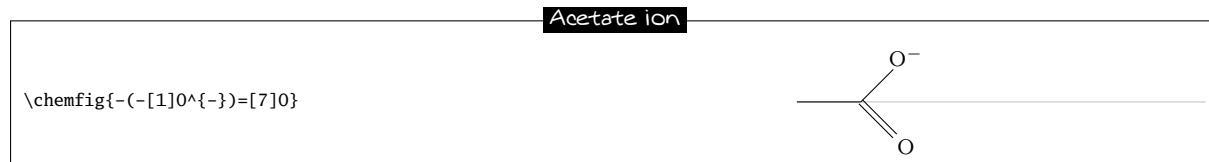
Bad coding

```
\chemfig{*4(A-B-C-D-)}\hspace{1cm}
\chemfig{*4(-B-C-D-A)}
```



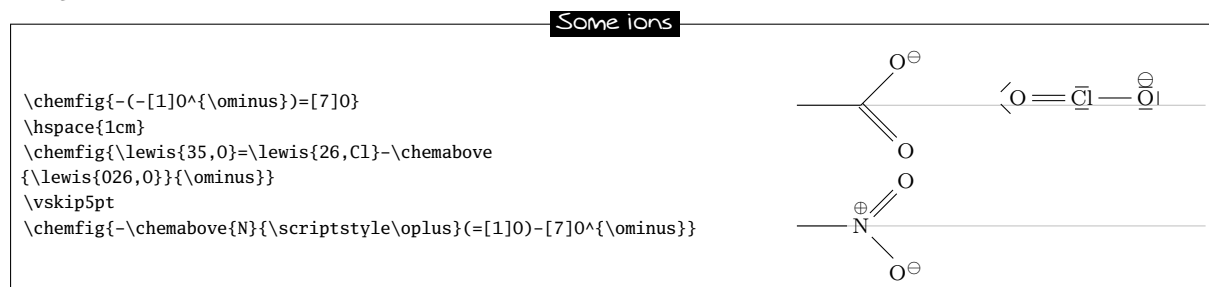
6 Ions

The chemfig commands enters the math mode¹ of T_EX, so it is very simple to write an ion. A negative charge (−) must always be enclosed in braces to avoid ChemFig confusing it with the symbol for a single bond.



For purists it is possible to circle the charge of an ion by using the commands `\ominus` and `\oplus`.

To meet all requirements, there are two other commands `\chemabove` and `\chembelow` which allow placement of charges above or below the current atom.

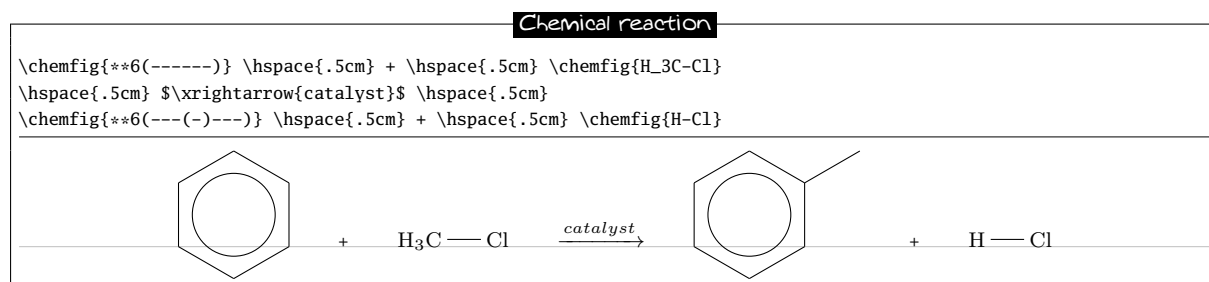


For those who are extremely picky, the commands `\chemabove` and `\chembelow` accept an optional argument which sets the distance between the charge and the atom.

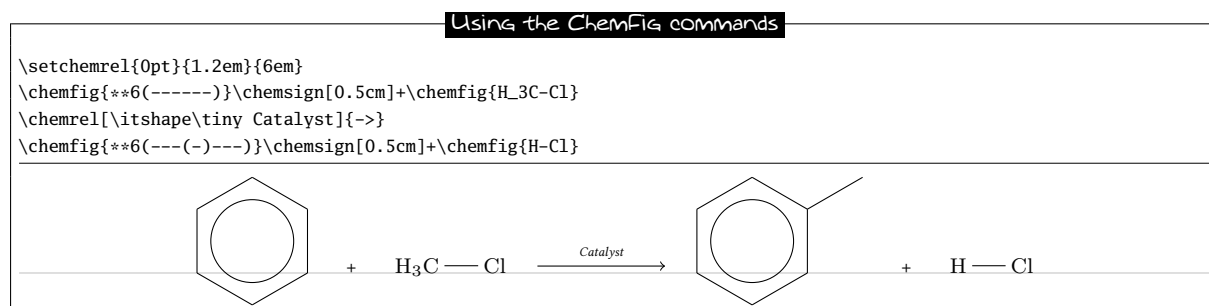


7 Chemical equations

Here is an example of a chemical reaction:



ChemFig adds two new commands `\chemsign` and `\chemrel` which slightly simplify the preceding syntax.



¹There is a problem with the placement of groups of atoms containing exponents or subscripts. See page 29.

The macro `\setatomsep{<dimension>}` adjusts the interatomic distance Δ . If the `<dimension>` is empty, it takes the default value of 3em. This command, like all other settings commands, affects all the following molecules.

Interatomic distance

```
\setatomsep{2em}\chemfig{A-B}\par
\setatomsep{50pt}\chemfig{A-B}
```

A — B
A ————— B

The command `\setbondoffset{<dimension>}` sets the spacing δ between the bond line and the atom. If the `<dimension>` is empty, δ takes the default value of 2pt.

Trimming Bonds

```
\setbondoffset{0pt}\chemfig{A-B}\par
\setbondoffset{5pt}\chemfig{A-B}
```

A — B
A — B

If one bond is followed immediately by another, then `ChemFig` inserts an empty group `{}`. Around this empty group the separation δ is zero:

Empty Groups

```
\chemfig{A-B==C}
```

A — B = C

The spacing δ for just one bond can be specified with the character `#`. This character must be placed *immediately* after the bond symbol and has one required argument between parentheses of the form “`#(<dim1>, <dim2>)`”, where `<dim1>` is the spacing δ at the beginning of the bond and `<dim2>` is the that at the end. If `<dim2>` is omitted, the spacing at the end of the bond takes the value of δ in effect at that time. One can see in the example how the shortening, set to 4pt to be more visible, is nullified for the bond arriving at “B”, then for the one leaving “B”, and finally for both:

Fine adjustment of bond shortening

```
\setbondoffset{4pt}
\chemfig{A-B-C}\par
\chemfig{A-#(,0pt)B-C}\par
\chemfig{A-B-#(0pt)C}\par
\chemfig{A-#(,0pt)B-#(0pt)C}
```

A — B — C
A — B — C
A — B — C
A — B — C

By default, all atoms within groups of atoms are typeset in math mode (spaces are ignored). They may therefore contain math mode specific commands such as subscripts or superscripts²:

Math mode

```
\chemfig{A_1B^2-C_3^4}
```

A₁B² — C₃⁴

There are settings specifically for Cram bonds. This syntax is used:

```
\setcrambond{<dim1>}{<dim2>}{<dim3>}
```

Any empty argument takes its default value. The three arguments are:

- `<dim1>` is the size of the base of the triangle, and is 1.5pt by default;
- `<dim2>` is the thickness of the dots, and is 1pt by default;
- `<dim3>` is the spacing between the dots, and is 2pt by default.

Here is an example where the three dimensions are changed:

Modified Cram Bonds

```
\setcrambond{10pt}{0.4pt}{1pt}
\chemfig{A>B>:C>|D}
```

A ► B ||||| C ▷ D

3 Bond angle

Each bond takes an optional argument in brackets. This optional argument can adjust every aspect of a bond, and consists of five optional fields separated by commas. The first of these fields defines the bond angle. Angles increase counterclockwise, and are relative to the horizontal. If the angle field is empty, the angle takes its default value of 0°. We will see later how to change this default.

There are several ways of specifying the bond angle.

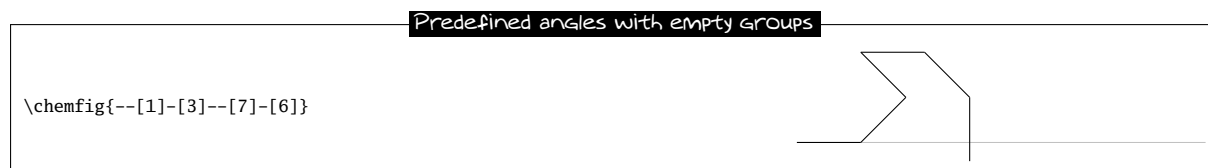
²There is a problem with the placement of groups of atoms containing exponents or subscripts. See page 29.

3.1 Predefined angles

When the angle field contains an integer, this represents the angle the bond makes relative to the horizontal, in multiples of 45° . For example, [0] specifies an angle of 0° , [1] is 45° , and so on up to [7] which specifies an angle of 315° . The integer may lie outside the interval [0, 7], in which case the angle is reduced to the interval [0, 360).

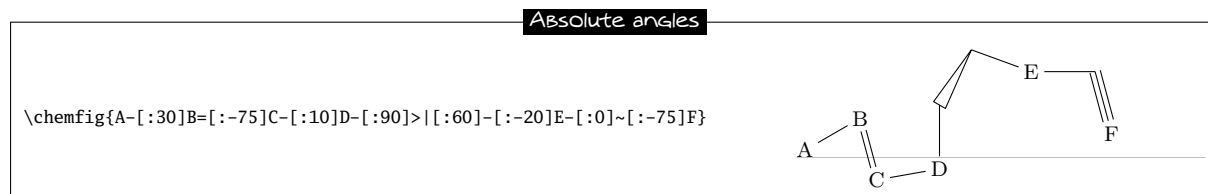


These angles remain valid if the atoms are empty, and this is the case for all the features we will see below:



3.2 Absolute angles

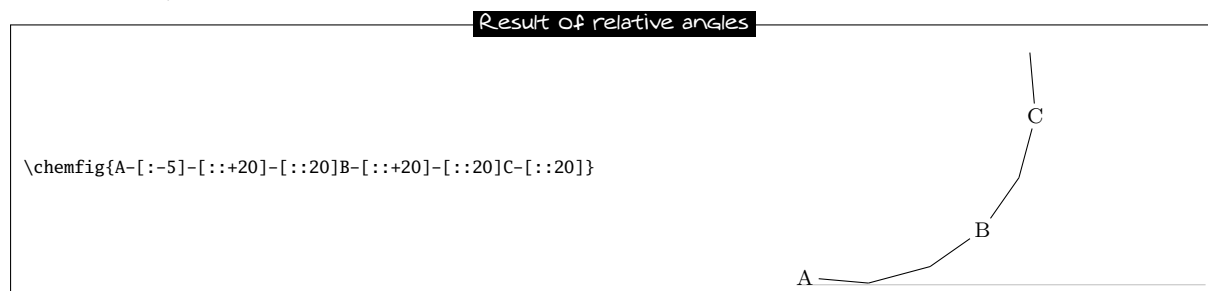
If one wishes to specify an angle in degrees relative to the horizontal, then the optional angle field must take this form: [:<absolute angle>]. If necessary, the <absolute angle> is reduced to the interval [0, 360):



3.3 Relative angles

It is often useful to specify a bond angle relative to the preceding bond. This syntax must be then be used: [::<relative angle>]. The sign of the <relative angle> can be omitted if it is a +.

Here is a molecule where the first bond has an absolute angle of -5° , and the rest of the bond angles are incremented by 20° :

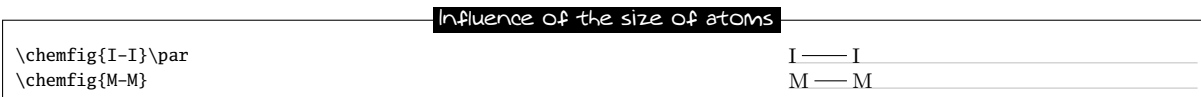


One can “break” a chain of relative angles by putting an absolute or predefined angle where desired. Here, atom “B” is followed by a bond at an absolute angle of 315° .



4 Length of a bond

Rather than speaking of length of a bond, we should use the term interatomic spacing. In effect, only the interatomic spacing is adjustable with `\setatomsep` as we have seen on page 10. Once this parameter is set, the length of a bond depends on the content of atoms and, to a lesser extent, the angle the bond makes with the horizontal. It should be obvious that two “slimmer” atoms will have larger edge separations than two which are larger. This can be seen easily in the following example where an “I” atom is narrower than an “M” atom, which means that the bond between the “I” atoms is longer than that between the “M” atoms:



This aspect of the size of atoms becomes particularly acute when the atom involves subscripts or superscripts. In this example, the bond is extremely short, to the point of confusion with a negative sign –:

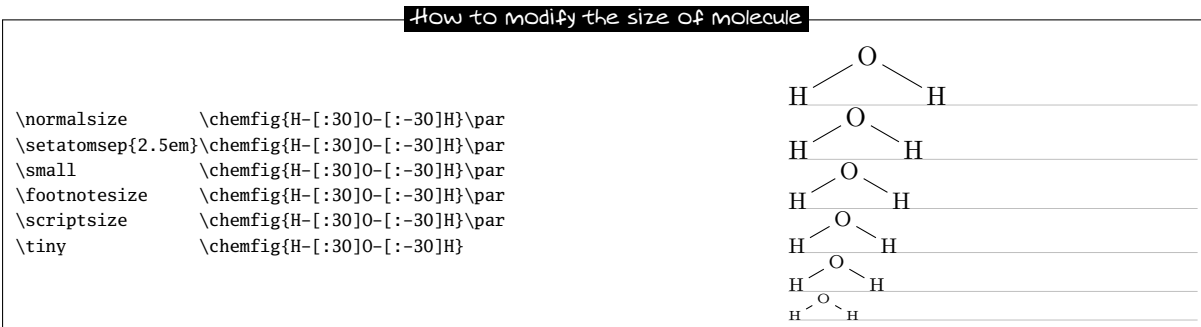


It is important to note that the exponent – is *put inside braces*. If this were not done, `ChemFig` would stop the atom on this character, which is a bond character. The atom would then be “B^”, which would lead to unexpected results.

We see in the example above that it is sometimes necessary to increase (or perhaps reduce) the interatomic distance associated with a bond. For this, the optional argument to bonds is actually made up of several comma-separated fields. As we have seen, the first field specifies the angle. The second field, if it is not empty, is a coefficient which multiplies the default interatomic distance Δ . Thus, writing `-[,2]` asks that this bond have the default angle (first field is empty) and that the atoms it connects be separated by twice the default distance.



We can change the size of molecules by altering the font size or the argument of `\setatomsep`, possibly on both³, being careful to confine these changes within a group if we want to limit the scope:

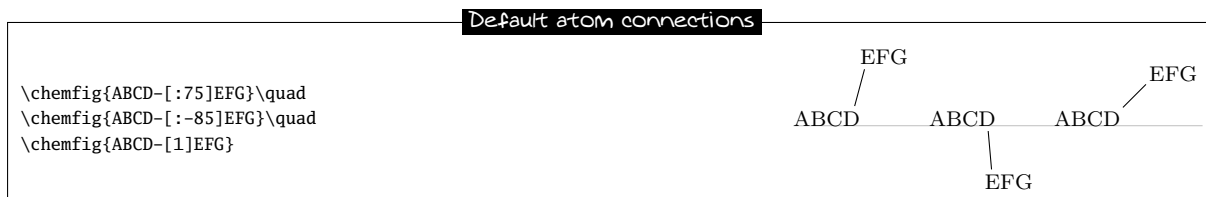


5 Departure and arrival atoms

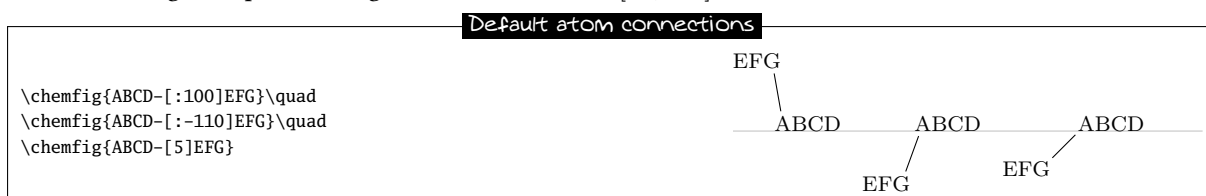
A group of atoms can contain several atoms. Suppose we want to connect the group “ABCD” to the group “EFG” with a bond. `ChemFig` calculates which atom of the first group and which of the second group are to be connected by looking at the angle of bond relative to the horizontal. If the angle is between (but not including) -90° and 90° (modulo 360°) then the bond is made between the last atom of the first group and the first atom of the second group. In all other cases, the bond is made between the first atom of the first group and the last atom of the second group.

³You can also use the second optional argument of `\chemfig`, see page 28.

Here are some examples where the bond is in the interval $(-90, 90)$, and where the bond is made between D and E:



In the following examples, the angles are in the interval $[90, 270]$ and so the bond is made between A and G:



One may sometimes want the bond partners to be atoms other than those determined by `ChemFig`. The departure and arrival atoms can be set with the optional bond argument by writing:

`[, , <integer 1> , <integer 2>]`

where `<integer 1>` and `<integer 2>` are the numbers of the desired departure and arrival atoms. These atoms must exist, otherwise an error message will be given.

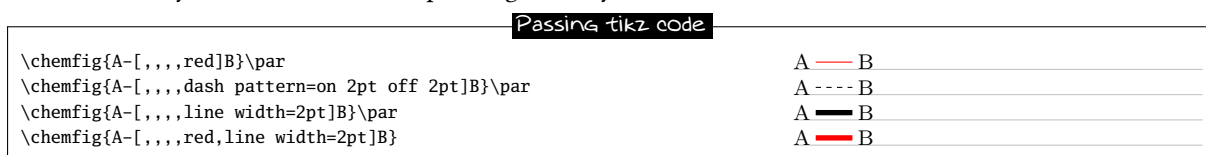


6 Customization of bonds

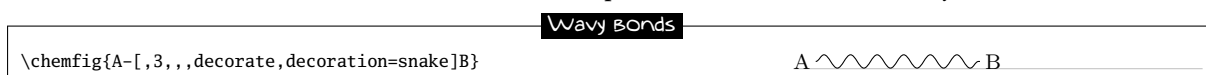
There is a fifth and last optional argument for bonds which is found after the fourth comma:

`[, , , , <tikz code>]`

This `<tikz code>` is passed directly to `tikz` when the bond is drawn. There one can put characteristics such as colour (red), dash type (`dash pattern=on 2pt off 2pt`), thickness (`line width=2pt`), or even decoration if the `tikz decoration` library has been loaded. A bond can be made invisible by writing `draw=none`. To set several attributes, the syntax of `tikz` is used, separating them by a comma:



Numerous `tikz` decoration libraries are available. For example, one can use the `pathmorphing` library by putting `\usetikzlibrary{decorations.pathmorphing}` in the preamble in order to draw wavy bonds:



Cram bonds ignore thickness and dash settings.

7 Default values

At the beginning of each molecule, the default values for the optional arguments are initialized. They are:

- 0° for the bond angle;
- 1 for the length multiplication coefficient;
- `<empty>` for the numbers of the departure and arrival atoms, which lets **ChemFig** calculate these based on the bond angle;
- `<empty>` for the parameters passed to `tikz`.

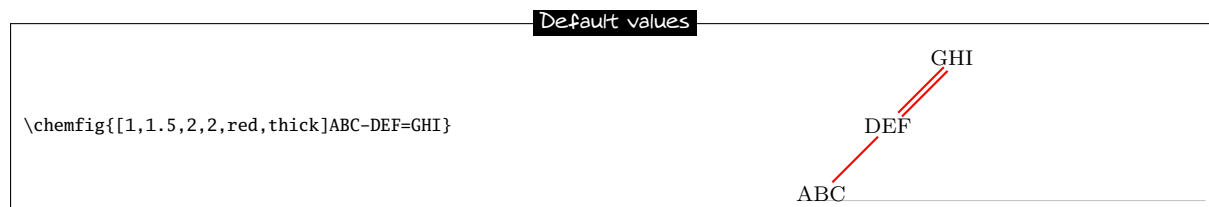
These default values can be changed for the whole molecule by beginning the molecule code with

```
[<angle>,<coeff>,<n1>,<n2>,<tikz code>
```

Thus, if the code of a molecule begins with `[:20,1.5]`, then all the bonds will be at angle of 20° by default, and the interatomic distances will have a length 1.5 times the default length. These default values can be overridden at any time by giving an optional argument, such as for the bond which follows atom “C” in this example:



If something odd like `[1,1.5,2,2,red,thick]` is written, then unless otherwise indicated all the bonds will have an angle of 45° , the interatomic distances will be 1.5 times the default distance, the bonds will begin and end on the second atom of each group, and the bonds will be red and thick:

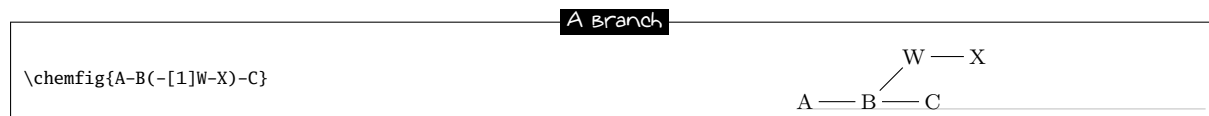


8 Branches

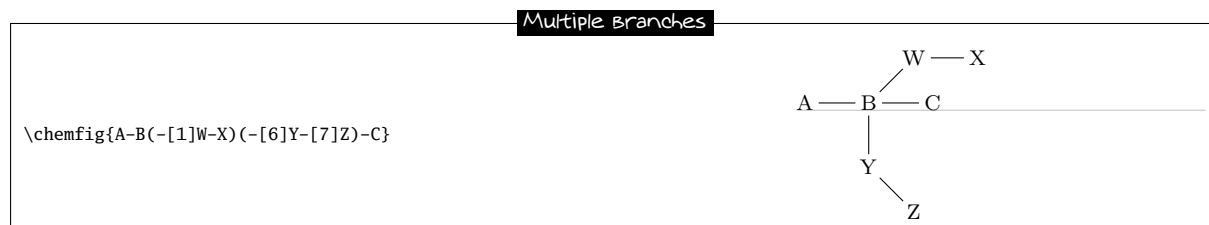
8.1 Principle

Up to now, all the molecules have been linear, which is rare. A sub-molecule can be attached to an atom by following the atom with `<code>` in parentheses. This `<code>` is the code of the submolecule which will be attached to the atom.

In this example, the sub-molecule “`-[1]W-X`” will be attached to atom “B”:



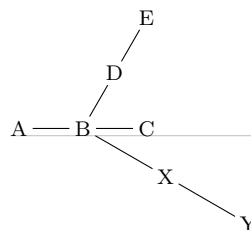
There can be several sub-molecules which are to be attached to the same atom. Just have several parentheses containing the code for each sub-molecule:



The code of each sub-molecule can define its own default values, which will be valid throughout the whole sub-molecule. Here a sub-molecule “`[:60]-D-E`” is attached to atom “B”, with a default angle of 60° absolute. A second sub-molecule “`[::-60,1.5]-X-Y`” is attached to “B” with a default bond angle 60° less than that of the preceding bond (which will be the one between “A” and “B”) and with an interatomic distance 1.5 times the default value:

Default values in branches

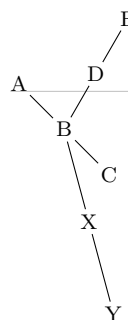
```
\chemfig{A-B([:60]-D-E)([:-30,1.5]-X-Y)-C}
```



Observe what happens if, at the beginning of the main molecule, one writes “[:-45]”:

Effect of the default bond angle

```
\chemfig{[:-45]A-B([:60]-D-E)([:-30,1.5]-X-Y)-C}
```



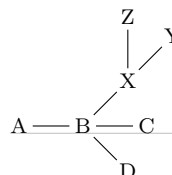
We see that the angle between the bond B-C and the bond B-X stays at 30° because it is a relative angle for the sub-molecule “-X-Y”. By contrast, the branch “-D-E” stays inclined at 60° to the horizontal, and does not follow the rotation given by the -45° angle at the beginning; this is expected because “-D-E” has an absolute angle. It is essential that all the angles be relative in order to rotate the whole molecule.

8.2 Nesting

Sub-molecules may be nested, and the rules seen in the preceding paragraphs stay in force:

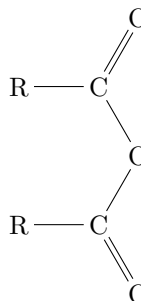
Nested branches

```
\chemfig{A-B([1]-X([2]-Z)-Y)(-[7]D)-C}
```



8.3 Method

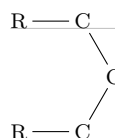
Suppose now that we want to draw an acid anhydride molecule: R—C(=O)—O—C(=O)—R



The best way to get this is to find the longest chain. Here, for example, we can draw the chain R-C-O-C-R taking into account angles and using only relative angles:

Acid anhydride structure

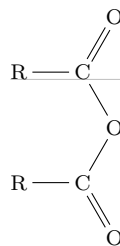
```
\chemfig{R-C-[:-60]O-[:-60]C-[:-60]R}
```



To this structure we just have to add two “=O” sub-molecules to each of the carbon atoms:

Acid anhydride

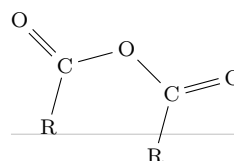
```
\chemfig{R-C(=[::+60]O)-[::-60]O-[::-60]C(=[::+60]O)-[::-60]R}
```



Because we used only relative angles, we can rotate this molecule by giving a default angle of e.g. 75°:

Rotation of a molecule

```
\chemfig{[:75]R-C(=[::+60]O)-[::-60]O-[::-60]C(=[::+60]O)-[::-60]R}
```



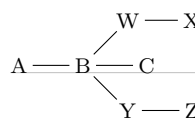
9 Connecting distant atoms

We have seen how to connect atoms *which are adjacent in the code*. It is often necessary to connect atoms which are not next to each other in the code. Let’s call these particular bonds “distant bonds”.

Let’s take this molecule:

Branched structure

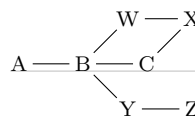
```
\chemfig{A-B(-[1]W-X)(-[7]Y-Z)-C}
```



and suppose that we want to connect the atoms X and C. In this case, **ChemFig** allows a “hook” to be placed *immediately* after the atom of interest. The character used for a hook is “?” because of its similarity to a hook. So, if one writes X? then the atom X will have a hook. Later in the code, all atoms followed by a ? will be connected to X:

Distant bond

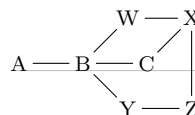
```
\chemfig{A-B(-[1]W-X?)(-[7]Y-Z)-C?}
```



We could connect other atoms to X by following them with ?. Here it’s the atoms C and Z:

Several distant bonds

```
\chemfig{A-B(-[1]W-X?)(-[7]Y-Z?)C?}
```



Now imagine if we were to leave the distant bonds X-C and X-Z while adding another: A-W. We must therefore ask for two *different* hooks, one on A and the other on X. Fortunately the character ? has an optional argument:

$$?[\langle \text{name} \rangle, \langle \text{bond} \rangle, \langle \text{tikz} \rangle]$$

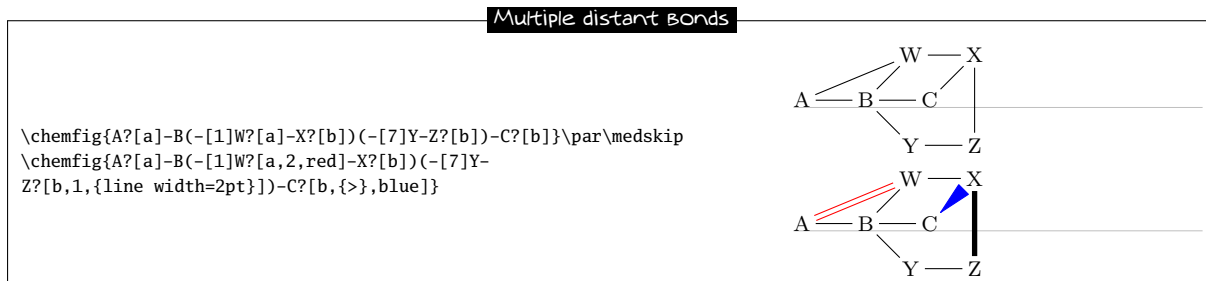
where each field takes its default value if it is empty:

- The $\langle \text{name} \rangle$ is the name of the hook: all alphanumeric characters (a...z, A...Z, 0...9) are allowed⁴. The name is a by default. In the first occurrence of the hook with this name, only this field is used.

⁴This is not exactly right. Actually all the characters that can be put between $\langle \text{csname} \dots \rangle \langle \text{endsname} \rangle$ are allowed.

- `<bond>` specifies how the atom with the current occurrence of the named hook is to be bonded to the atom with the first occurrence of the hook. There are two ways this can be done. First, this field can be an integer representing the desired bond type: 1=single bond, 2=double bond, etc. (See the table on page 5 for the bond codes.)
Second, the field can be one of the bond character codes, provided that this character is *between braces*.
- `<tikz>` will be passed directly to `tikz` as we have seen with regular bonds.

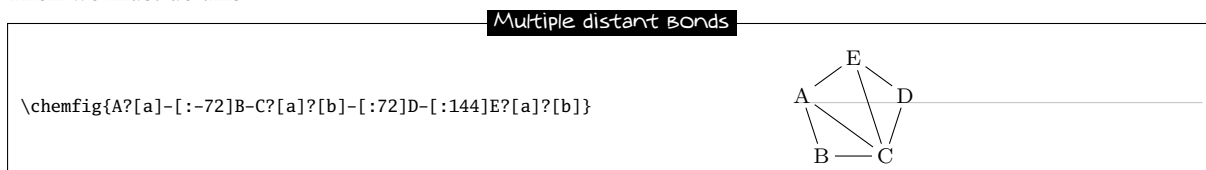
Here is our molecule with the required distant bonds, then with the bond A-W and X-C customized:



Several different hooks can be written after an atom. Suppose that in this unfinished pentagon, we wish to connect A-E, A-C and E-C:



Then we must do this:



10 Rings

The preceding example shows how to draw a regular polygon, but the method used is tedious because the angles depend on the number of sides of the polygon.

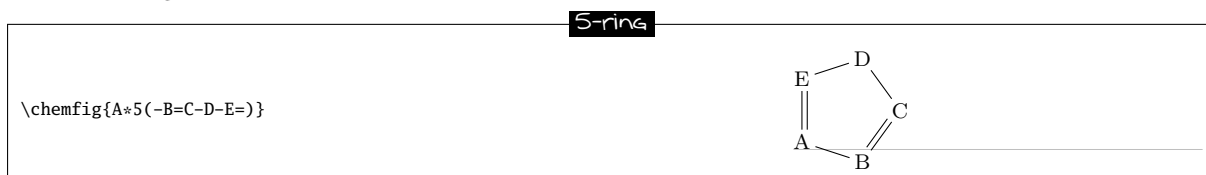
10.1 Syntax

`ChemFig` can easily draw regular polygons. The idea is to attach a ring to an `<atom>` outside the ring with this syntax:

$$\langle \text{atom} \rangle * \langle n \rangle (\langle \text{code} \rangle)$$

`<n>` is the number of sides of the polygon and the `<code>` describes the bonds and groups of atoms which make up its edges and vertices. This code *must* begin with a bond because the atom is outside the ring.

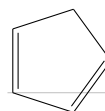
Here is a 5-ring, attached to the atom “A”:



A ring can also be drawn with one, several, or all the groups of atoms empty, as is the case for diagrams outside rings:

5-rings with empty groups

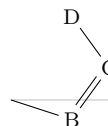
```
\chemfig{*5(-----)}
```



A ring can be incomplete:

Incomplete 5-ring

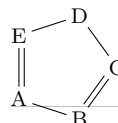
```
\chemfig{*5(-B=C-D)}
```



If a ring has a code which contains too many bonds and atom groups for the given number of vertices, all the bonds and groups over the maximum allowed are ignored:

Truncated 5-ring

```
\chemfig{A*5(-B=C-D-E=F-G=H-I)}
```



It is possible to draw a circle or an arc in the inside of a ring. To do so, the following syntax is used:

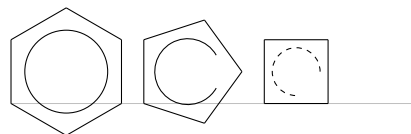
```
<atom>**[<angle 1>,<angle 2>,<tikz>]<n>(<code>)
```

where each field of the optional argument takes its default value if it is empty:

- `<angle 1>` and `<angle 2>` are the absolute angles of the start and finish of the arc. These default to 0° and 360° respectively so that a circle is drawn by default;
- `<tikz>` is the code that will be passed to `tikz` for drawing the arc.

Rings and arcs

```
\chemfig{*6(-----)}\quad
\chemfig{**[30,330]5(-----)}\quad
\chemfig{**[0,270,dash pattern=on 2pt off 2pt]4(-----)}
```



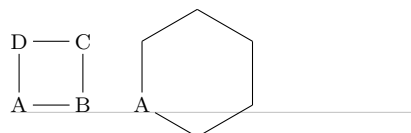
10.2 Angular position

10.2.1 At the start

As can be seen in the examples above, the rule is that the attachment atom “A” is always at the south-west of the ring. Furthermore, the ring is always constructed counterclockwise, and the last bond descends vertically onto the attachment atom:

Angular position of rings

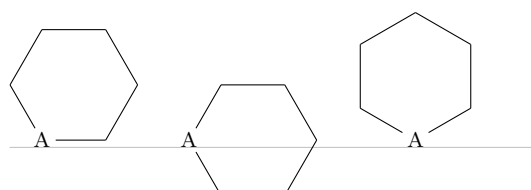
```
\chemfig{A*4(-B-C-D-)}\quad\chemfig{A*6(-----)}
```



If this angular position is not convenient, it is possible to specify another angle using the optional argument at the beginning of the molecule. Here is a 6-cycle which has been rotated by $+30^\circ$, by -30° , and lastly by $+60^\circ$:

Rotation of rings

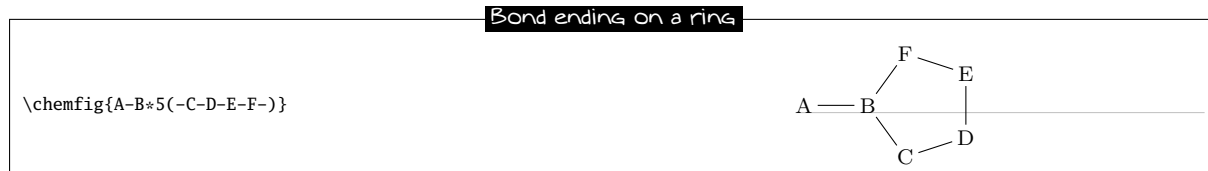
```
\chemfig{[:30]A*6(-----)}\quad
\chemfig{[:-30]A*6(-----)}\quad
\chemfig{[:60]A*6(-----)}
```



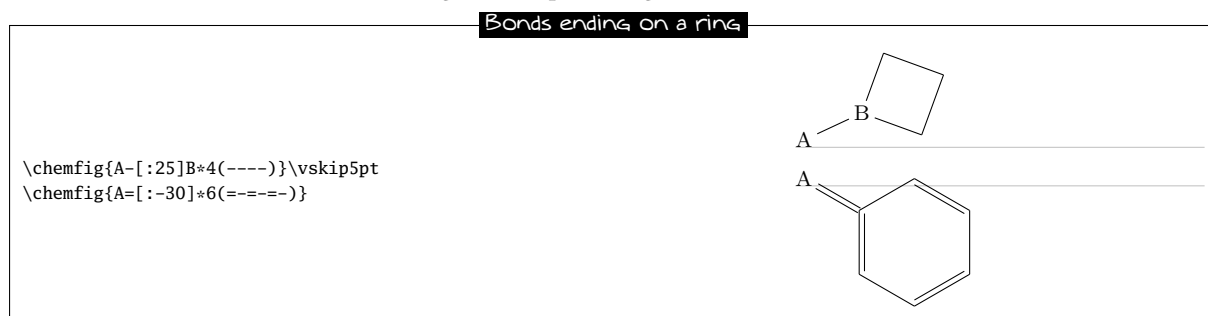
10.2.2 After a bond

When a ring does not begin a molecule and one or more bonds have already been drawn, the default angular position changes: the ring is drawn in such a way that the bond ending on the attachment atom bisects the angle formed by the first and last sides of the ring.

Here is a simple case:



The rule remains valid, whatever the angle of the preceding bond:

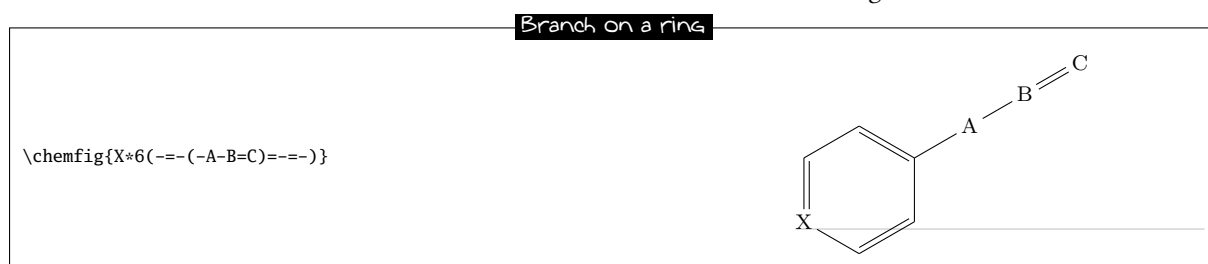


10.3 Branches on a ring

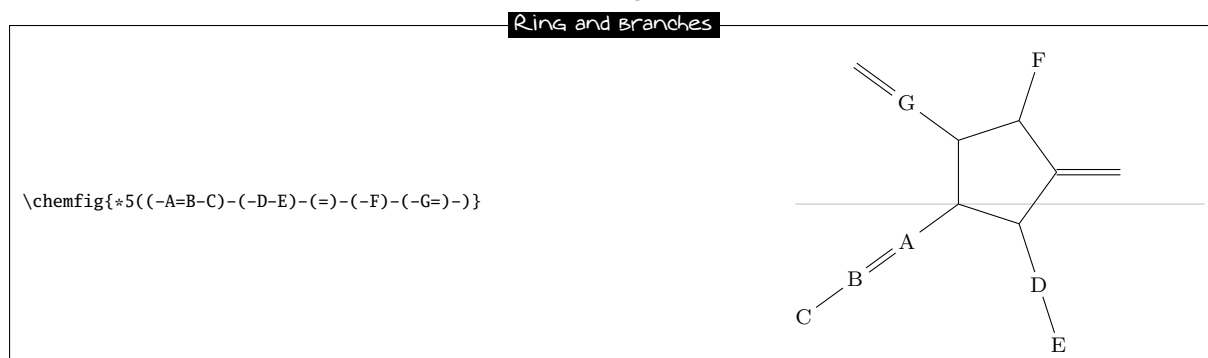
To have branches attached to the vertices of a ring, we use the syntax we have already seen:

`<atom>(<code>)`

where the `<code>` is that of the sub-molecule and the `<atom>` is at the vertex. Unique to rings, the default angle of the sub-molecule is not 0° but is calculated so that it will bisect the sides leaving the vertex:



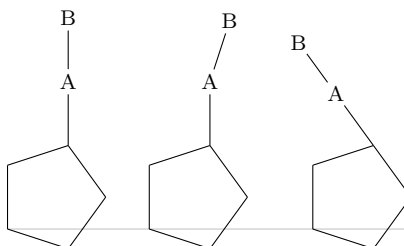
A sub-molecule can be attached to the first vertex of a ring, just like the other vertices:



If one wants the bond leaving a vertex not to be the bisector of its sides, one can tinker with the optional global parameter or the optional bond parameter:

Branches at specified angles

```
\chemfig{*5(---[:90]-A-B---)}\quad
\chemfig{*5(---[:-90]A-B---)}\quad
\chemfig{*5(---[:+0]-A-B---)}
```

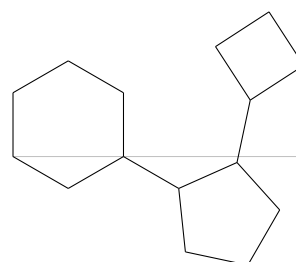


It is worth noting that in the third example, where a relative angle of 0° was given, the bonds of the branch are drawn in line with the preceding bond in the ring. This is the rule on page 12 which specified that the reference angle was that of the bond last drawn.

We can now connect together rings with bonds:

Connected rings

```
\chemfig{*6(--(*5(----(*4(----))--))----)}
```

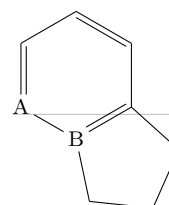


10.4 Nested rings

To “glue” two rings together, the syntax is only slightly different: the vertex is specified where the other ring is going to start. Simply follow this vertex by the usual syntax for a ring. Here for example is a 5-ring which is attached to the second vertex of a 6-ring:

Nested rings

```
\chemfig{A*6(-B*5(----)=---)}
```

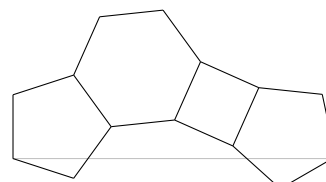


Note that the ring which is going to be attached to the main ring has an angular position such that two of the rings' sides coincide. In addition, the 5-ring has only four bonds “----”. In effect, the fifth will be useless because it is the second side of the 6-ring, which has already been drawn.

It is quite possible to glue multiple rings together:

Multiple nested rings

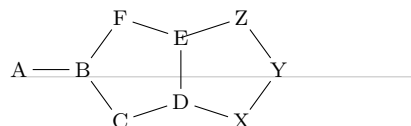
```
\chemfig{*5(--*6(--*4(--5(----)--))----)}
```



There is a case where a trick must be used. It can be seen in this example that the fourth side of the second 5-ring just passes through the centre of atom “E”.

Flawed drawing

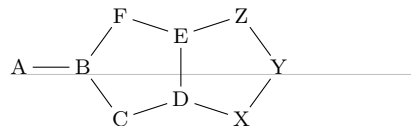
```
\chemfig{A-B*5(-C-D*5(-X-Y-Z)-E-F-)}
```



This is normal because the second 5-ring (which is attached to atom “D”) is drawn *before* ChemFig knows about atom “E”. In this case, it is necessary to use two hooks to draw the bond Z-E:

Distant bond and ring

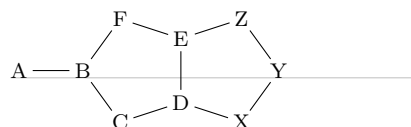
```
\chemfig{A-B*5(-C-D*5(-X-Y-Z?)-E?-F-)}
```



We could also use a `` at the last vertex of the 5-ring:

Using \phantom

```
\chemfig{A-B*5(-C-D*5(-X-Y-Z-\phantom{E})-E-F-)}
```

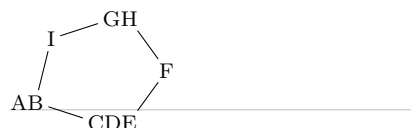


10.5 Rings and groups of atoms

Some care must be taken with rings when one or more vertices are made up of groups of atoms:

Ring and groups of atoms

```
\chemfig{AB*5(-CDE-F-GH-I-)}
```



In order for the ring to have a regular shape, it is necessary to override the ChemFig mechanism which automatically calculates the departure and arrival atoms of bonds. Here, C-F and F-G must be connected by using the optional argument of these bonds:

Forced departure and arrival atoms

```
\chemfig{AB*5(-CDE-[,,,1]F-[,,,1]GH-I-)}
```



11 Representing electron movements

Starting with ChemFig version 0.3, we can represent the movement of electrons in mesomeric effects or reaction mechanisms. This is done by marking the departure and arrival points of the electron movement arrow using the syntax “@{<argument>}”. This syntax allows a tikz node to be placed and makes this node accessible outside the argument of the `\chemfig` command thanks to the “remember picture” option which is passed to all the “tikzpicture” environments. It is assumed that the viewer supports “picture remembering” and that the compilation is done twice.

Two types of diagrams can arise, so we can ask for:

- a zero size node on a bond using the syntax “@{<name>, <coeff>}” placed at the beginning of the optional argument of the relevant bond, without being followed by a comma if there is a first optional argument. In this case, the node takes the name “<name>” and the <coeff>, which must be between 0 and 1, determines where the node is located on the bond. If “@{<name>}” is used, the <coeff> is set to 0.5 by default, which means that the node is placed halfway along the bond;

- a node on an atom using the syntax “@{<name>}” immediately before the relevant atom. In this case, the node has exactly the same footprint as the atom, but may be empty and therefore have zero dimensions.

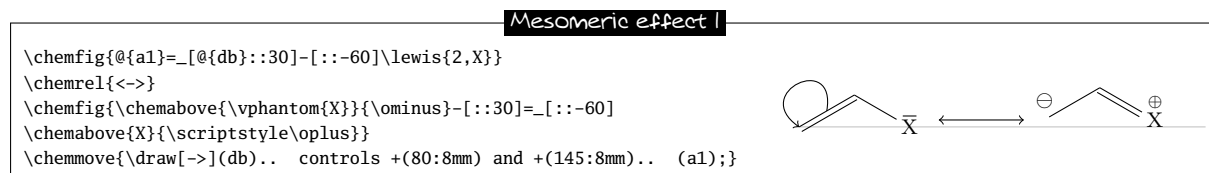
Once the `\chemfig` command has drawn the molecule(s) and has placed the nodes with the syntax described above, we can connect these nodes to each other with `tikz` instructions. These instructions are placed in the argument of the command `\chemmove`⁵ and has the following syntax if (for example) we need to connect a node named “<name1>” to the node named “<name2>”:

```
\chemmove[<opt>]{\draw[<tikz opt>](<name1><tikz link>(<name2>);}
```

The optional argument <opt> of the `\chemmove` command will be added to the argument of the `tikzpicture` environment in which the links between the nodes will be drawn. The <tikz opt> and <tikz link> instructions are describe in detail in the documentation of the `tikz` package.

11.1 Mesomeric effects

To make these concepts concrete, let’s take the example of a mesomeric effect involving a double bond and non-bonding lone pair conjugate. Let’s begin with the possible delocalization of electrons from the double bond. We will place a node named “db” (double bond) in the middle of the double bond and a node named “a1” on the end of the double bond.

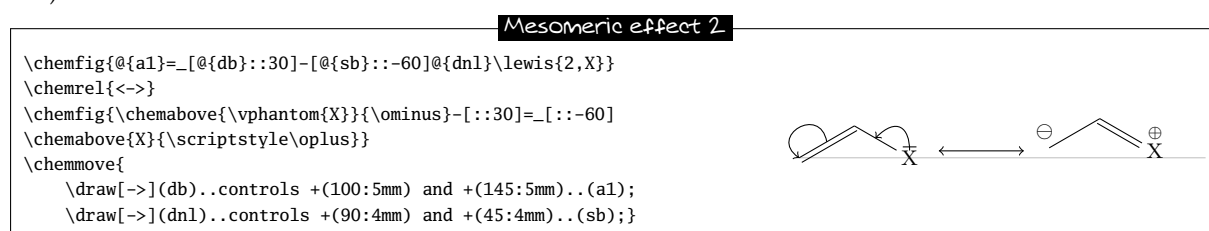


As noted above, there is no comma after the node placed in the optional arguments of a bond; we write “=_[@{db}::30]” and not “=_[@{db}, ::30]” as one might be tempted to do.

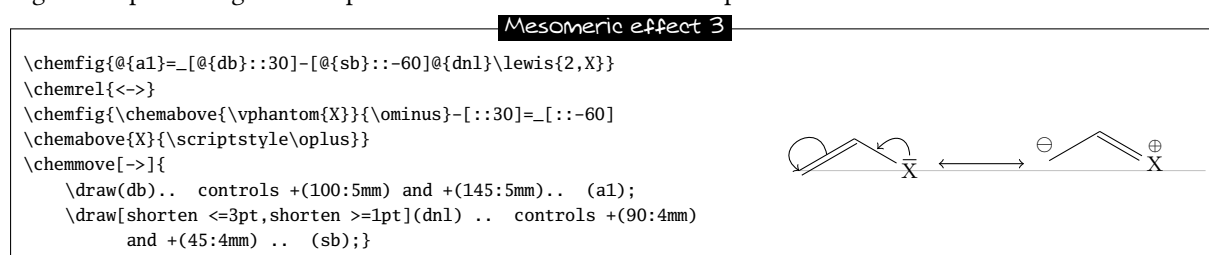
To link the nodes “db” and “a1” we have used the following syntax:

```
\chemmove{\draw[->](db)..controls +(80:8mm) and +(145:8mm)..(a1);}
```

In this example we ask for an arrow (`[->]`) and we use two control points⁶. These will be located using polar coordinates at 80° and 8 mm from “db” for the first and at 145° and 8 mm from “a1” for the second. Though this syntax may seem complicated at first reading, one need not be alarmed because its use will usually be a matter of copying and pasting. Only the names and coordinates of the control points need be changed, as can be verified from the example below, where an arrow has been added from the lone pair (node “dnl” to the single bond (node “sb”).



For our new arrow we have set the control points as follows: 4 mm at an angle of 90° from “dnl” and 4 mm at an angle of 45° from “sb”. But we are not completely satisfied, since we would like the arrow not to touch the line segment representing the lone pair. To do this we will add some options to our arrow.

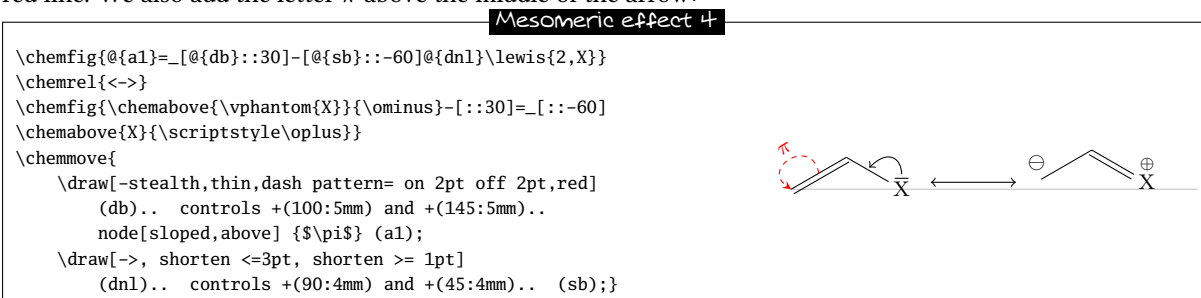


⁵Actually, the `\chemmove` command puts its argument in a “tikzpicture” environment with the options “remember picture, overlay”.

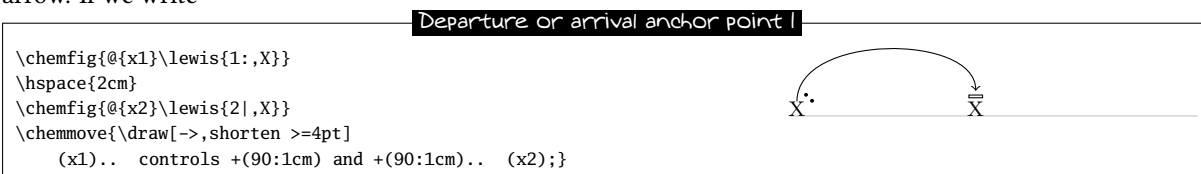
⁶To find all the ways of connecting two nodes with `tikz`, read the documentation for that package.

The option “shorten <=3pt” indicates that the tail of the arrow is to be shortened by 3 pt just as “shorten >=2pt” means that the head of the arrow is shortened by 2 pt.

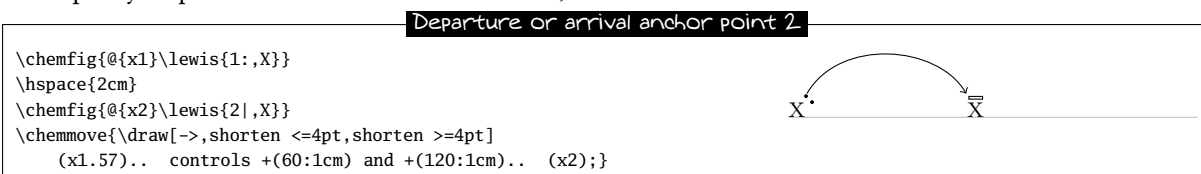
We can use all the power of `tikz` instructions to modify the style of the arrow. Here we change the head of the arrow leaving the double bond, writing “-stealth” instead of “->”, and we draw the arrow with a fine dashed red line. We also add the letter π above the middle of the arrow:



In the following example, we'll see how to indicate the position of the departure or arrival anchor points of the arrow. If we write



Note that the tail of the arrow does not leave correctly from our electrons; it leaves from the middle of the upper edge of the node. Indeed, we chose a departure angle of 90° and so `tikz` makes the arrow leave from the anchor “x1.90” which corresponds to the intersection of the ray leaving from the centre of node “x1” at a 90° angle relative to the horizontal and of the edge of the rectangular node. To get the arrow departure angle that we want, we must specify its position. After some trial and error, it is “x1.57”:



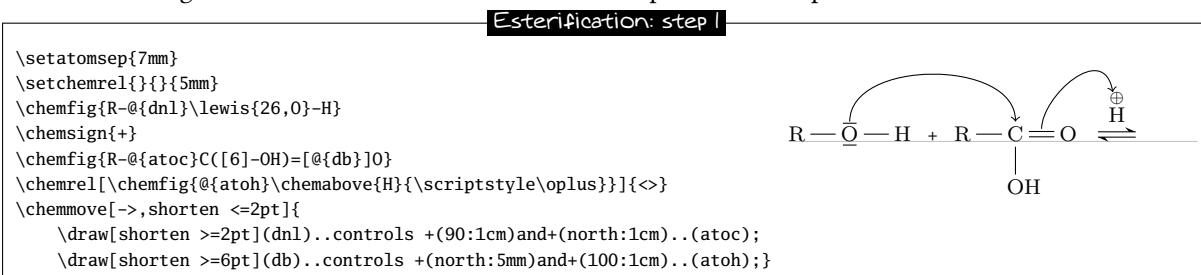
In some cases it will be easier to use Cartesian coordinates for the control points. Here we use just one control point placed 1 cm to the right of and 1.5 cm above “x1”:



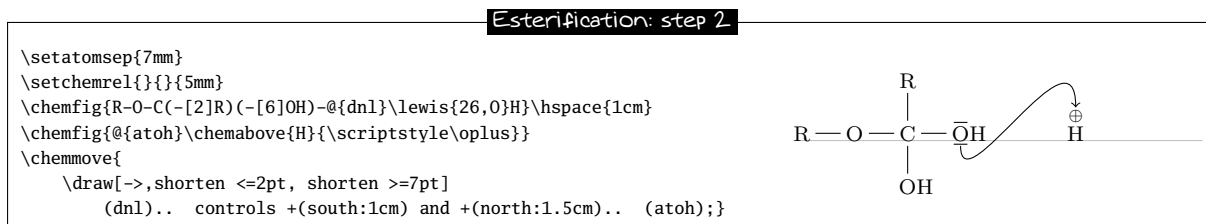
All the graphics drawn by means of the command `\chemmove` are superimposed and will not be included in the bounding boxes. We can see this in the preceding example.

11.2 Reaction mechanisms

Thanks to the option `remember picture` which is passed to all the “tikzpicture” environments we can easily draw arrows indicating reaction mechanisms. Let's take for example the first step of the esterification reaction.



The use of the `\chemabove{<code>}{<material>}` command does not change the dimensions of the bounding box of `<code>`. For this reason we can run into some difficulty in pointing to the symbol representing the charge carried (\oplus or \ominus). In the example above the solution is to create a control point with an angle of 110° at 1 cm from “atoh” and to shorten the arrow by 6pt. In the following example, the second step of the esterification reaction, we can see that the arrow can take more complicated forms without complicating the code.



The rest is left as an exercise to the reader...

12 Writing a name under a molecule

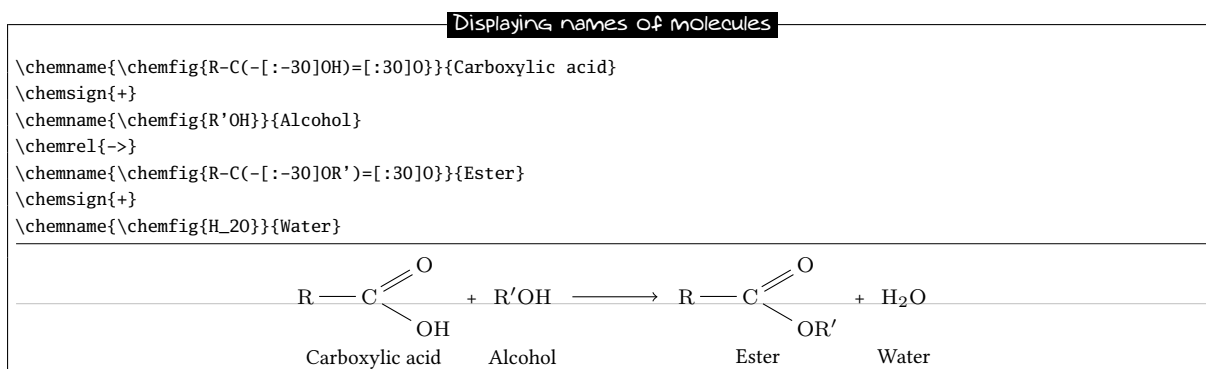
For convenience, **ChemFig** can write the name of a molecule underneath it with the command

```
\chemname[<dim>]{\chemfig{<code of the molecule>}}{<name>}
```

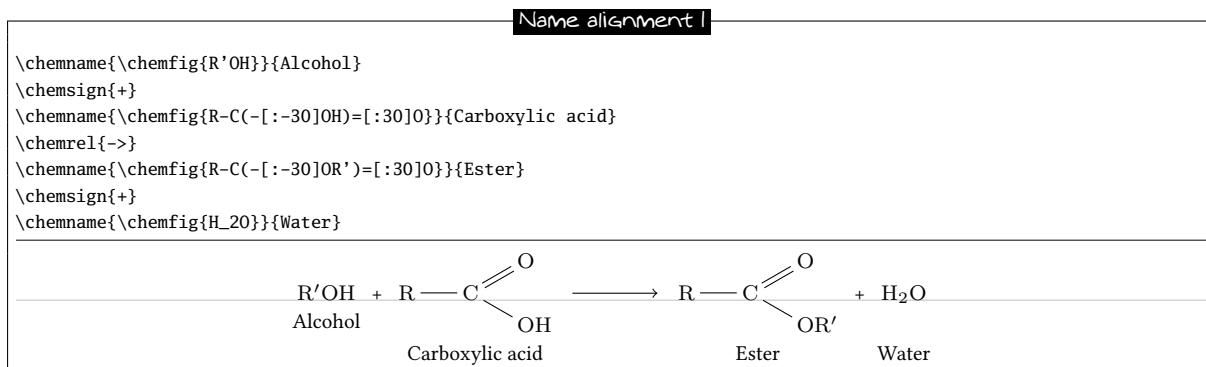
The `<dim>`, which is 1.5ex by default, will be inserted between the baseline of the molecule and the top of the letters of the `<name>`. The `<name>` will be centred relative to the molecule, but the `<name>` may not contain multiple paragraphs. As we see in this example: $\text{H} - \text{O} - \text{H}$, the `<name>` which is displayed under the molecule is

taken into account only for the vertical size of the bounding box. The horizontal size of `<name>` is always zero.

Here is a reaction with the names under the molecules:



There are some limitations to this command. Suppose we switch the acid and the alcohol on the left side:



In fact, to draw the `<name>` the command `\chemname` inserts $1.5\text{ex} + \textit{the largest of the depths}^7$ of the molecules *thus far* below the baseline of each molecule (light grey for the examples in this manual). The command `\chemnameinit{<stuff>}` initializes this largest depth with the `<stuff>`. Therefore one should:

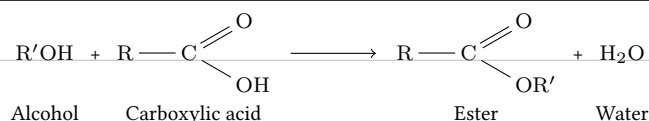
⁷In T_EX terms, the depth is the dimension which extends vertically below the baseline.

- write `\chemnameinit{<deepest molecule>}` before using the `\chemname` command in a reaction, unless the reaction begins with the deepest molecule;
- write `\chemnameinit{}` after having written all the names in a chemical reaction lest the greatest depth in this reaction interfere with a future reaction.

Thus the correct code uses `\chemnameinit` before and after the reaction:

Name alignment 2

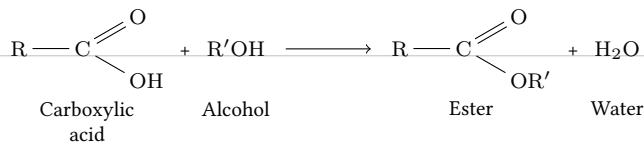
```
\chemnameinit{\chemfig{R-C(-[: -30]OH)=[ :30]O}}
\chemname{\chemfig{R'OH}}{Alcohol}
\chemsign{+}
\chemname{\chemfig{R-C(-[: -30]OH)=[ :30]O}}{Carboxylic acid}
\chemrel{->}
\chemname{\chemfig{R-C(-[: -30]OR')=[ :30]O}}{Ester}
\chemsign{+}
\chemname{\chemfig{H_2O}}{Water}
\chemnameinit{}
```



Finally, to write a name on multiple lines, the command `\par` encountered in a `<name>` causes a line break⁸:

Name on 2 lines

```
\chemname{\chemfig{R-C(-[: -30]OH)=[ :30]O}}{Carboxylic\par acid}
\chemsign{+}
\chemname{\chemfig{R'OH}}{Alcohol}
\chemrel{->}
\chemname{\chemfig{R-C(-[: -30]OR')=[ :30]O}}{Ester}
\chemsign{+}
\chemname{\chemfig{H_2O}}{Water}
\chemnameinit{}
```



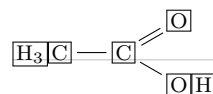
⁸Conversely, the command `\par` is forbidden and causes a compilation error.

- `\def\printatom#1{\ifmmode\rm#1\else$\rm#1$\fi}` when compiling with $\epsilon\text{T}\text{E}\text{X}$ ou $\text{C}\text{o}\text{n}\text{T}\text{E}\text{x}\text{T}\text{X}$.

One can modify the code of this macro to customize how atoms are displayed. In the following example, we redefine `\printatom` so that each atom will be enclosed in a rectangular box:

Redefinition of `\printatom`

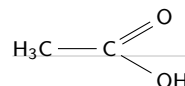
```
\fboxsep=1pt
\renewcommand*\printatom[1]{\fbox{\ensuremath{\mathrm{#1}}}}
\chemfig{H_3C-C(=[:30]O)(-[:-30]OH)}
```



Here is how to redefine it to use the “sf” font family of math mode:

Atoms displayed with “sf” font family

```
\renewcommand*\printatom[1]{\ensuremath{\mathsf{#1}}}}
\chemfig{H_3C-C(=[:30]O)(-[:-30]OH)}
```



3 Optional arguments

The `\chemfig` command takes two optional arguments; their syntax is as follows:

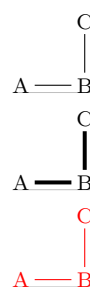
$$\backslash\text{chemfig}[\langle\text{opt1}\rangle][\langle\text{opt2}\rangle]\{\langle\text{molecule code}\rangle\}$$

The first optional argument `<opt1>` contains `tikz` instructions which will be passed to the `tikzpicture` environment in which the molecule is drawn. The second optional argument `<opt2>` contains `tikz` instructions which will be executed when each node⁹ is drawn.

With the use of the first optional argument one can, for example, choose the global colour or thickness of lines:

Style choice

```
\chemfig{A-B-[2]C}\par\medskip
\chemfig[line width=1.5pt]{A-B-[2]C}\par\medskip
\chemfig[red]{A-B-[2]C}
```



With the second optional argument, one can choose the colour of nodes drawn by `tikz`, change the angle of the drawing or its scale:

Style choices

```
\chemfig{A-B-[2]C}\par\medskip
\chemfig[red]{A-B-[2]C}\par\medskip
\chemfig[dash pattern=on 1pt off 2pt][red]{A-B-[2]C}\par\medskip
\chemfig[rotate=20]{A-B-[2]C}\par\medskip
\chemfig[scale=0.5]{A-B-[2]C}
```

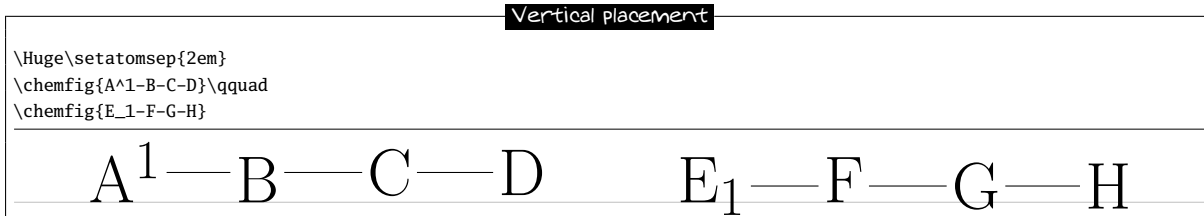


⁹These instructions are added to the end of the argument of every `node/.style{<argument>}`. This argument contains by default the following instructions: “`anchor=base,inner sep=0pt,outer sep=0pt,minimum size=0pt`”.

4 Vertical alignment

In some cases with condensed structural diagram of molecules having horizontal bonds, the placement of groups of atoms is incorrect.

Careful study of the following example shows that the groups of atoms are not correctly aligned on the baseline:

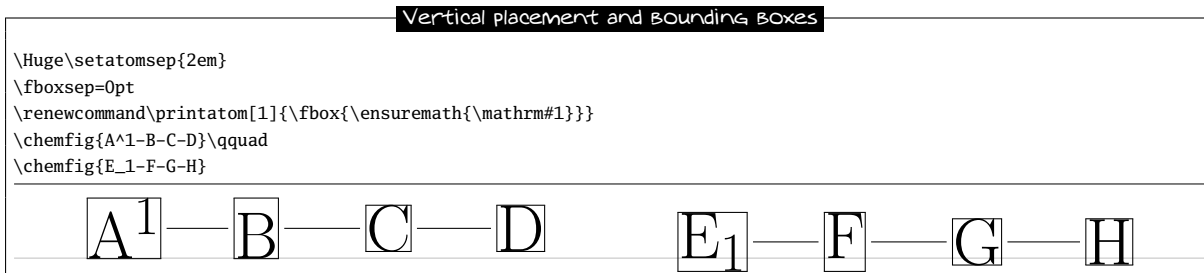


Surprisingly, the second atom is correctly aligned while the last two undergo a vertical shift which seems to be the results of the different height of the bounding box of the atoms “A¹” and “E₁”.

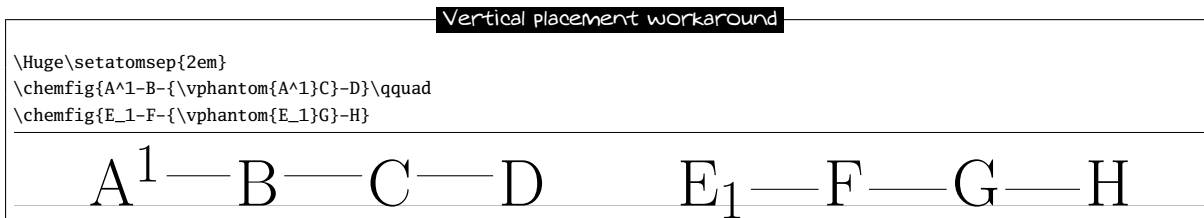
In order to understand this phenomenon, we need to consider how `ChemFig` places groups of atoms relative to each other. Let us limit ourselves to the case of horizontal bonds in order to simplify terminology, although the algorithm is the same for other bonds. A horizontal bond leaves from the middle of the right side of the bounding box of the departure atom of this bond. The arrival atom is positioned in such a way that the middle of the left side of its bounding box is at the end of the bond. It follows that the vertical placement of the arrival atom depends on the height of the departure atom. To limit this phenomenon, `ChemFig` adds to each arrival atom the `\vphantom` of the departure atom, but does not include it in the contents of the arrival atom; this `\vphantom` is not intended to affect the following atoms. The atoms remaining in each group are aligned so that their baseline coincides with the baseline of the preceding atom.

The defective alignment can thus be explained. The atoms “B” and “F” are aligned correctly as they reflect the height of the atoms before them because of their `\vphantom`. For the atoms “C” and “G”, the heights of the immediately preceding atoms are taken into account, but those of the atoms “A¹” and “E₁” are ignored! It follows that these atoms are a little too high or too low, depending on the height of these bonds.

We can show this by making visible the bounding boxes of the atoms; one sees clearly that the atoms “B” and “F” have bounding boxes that reflect the heights of the immediately preceding atoms:



Since there is no satisfactory manual solution, this problem can be worked around manually by putting *inside* the third atom a `\vphantom` having the same height as the first, so that the height affects the following atoms:



For any group of atoms it is possible to temporarily deactivate the alignment adjustment mechanism and thus neutralize the `\vphantom`. Simply place the `\chemskipalign` command in the group of atoms; the alignment will resume in the following group of atoms as if the group of atoms containing `\chemskipalign` had never existed. The following example shows the effects of this instruction: the reference point of the box containing the first atom is placed at the level of the bond which arrives from the left. The bounding boxes of the atoms are drawn in the second line.

Deactivation of the alignment mechanism

```

\large
\chemfig{A-. -B}\quad
\chemfig{A-\chemskipalign.-B}\par\bigskip
\fbboxsep=0pt
\renewcommand\printatom[1]{\fbox{\ensuremath{\mathrm{#1}}}}
\chemfig{A-. -B}\quad
\chemfig{A-\chemskipalign.-B}

```

This command is to be used with caution lest the alignment of atoms in the next group be disrupted. In general, all will be well if the group of atoms featuring `\chemskipalign` contains a *single atom* whose height and depth are *less* than those of the preceding and following atoms, and if the preceding and following atoms have identical heights and depths. Here is an example of the mess that results when the group of atoms contains two atoms, here “`\chemskipalign.`” and “`B`”:

Consequence of the `\chemskipalign` command

```

\large
\fbboxsep=0pt
\renewcommand\printatom[1]{\fbox{\ensuremath{\mathrm{#1}}}}
\chemfig{A-\chemskipalign.B-C}

```

This feature can sometimes be useful. Suppose we want to draw the following molecule



We can define commands which will draw the empty and full disks with `tikz`. To ensure that these disks are at the right height, namely the height of the bond arriving at them, we will use the command `\chemskipalign`. In the second line below the bonds are “stuck” to the disks by using the ability to change the bond shortening with the “`#`” character, a feature seen on page 11.

Use of `\chemskipalign` and `#`

```

\begingroup
\def\emptydisk{\chemskipalign\tikz\draw(0,0)circle(2pt);}
\def\fulldisk{\chemskipalign\tikz\fill(0,0)circle(2pt);}
\chemfig{A-\emptydisk-\fulldisk-B}\par
\chemfig{A-#(,0pt)\emptydisk-#(0pt,0pt)\fulldisk-#(0pt)B}
\endgroup

```

5 Shifted double bonds

All double bonds are made up of two line segments, and these segments are drawn on either side of the imaginary line along which a single bond would be drawn. It is possible to shift a double bond so that one of the line segments lies on the imaginary line. The other segment is then shifted above or below the bond. Actually, it is more correct to say “left” or “right” of the imaginary line, as the bond is traversed in the direction of drawing.

To shift the bond to the left, write “`=^`” and to shift it to the right, write “`=_`”:

Shifted double bonds

```

\chemfig{A==B}\par
\chemfig{A=^B}\par
\chemfig{A=_B}

```

In rings, double bonds are automatically shifted to the left. However, they can be shifted to the right by specifying it with “`=_`”:

Shifted double bonds and rings

```

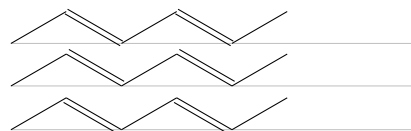
\chemfig{*6(==)}\quad
\chemfig{*6(=_=_)}

```

Shifted bonds are particularly useful in drawing skeleton diagrams of molecules consisting of carbon chains with double bonds. They give a continuous zig-zag path, whereas the path will be broken with regular double bonds:

Shifted bonds and skeleton diagrams

```
\chemfig{-[:30]=[: -30]-[:30]=[: -30]-[:30]}\par
\chemfig{-[:30]=^[: -30]-[:30]=^[: -30]-[:30]}\par
\chemfig{-[:30]=_[: -30]-[:30]=_[: -30]-[:30]}
```



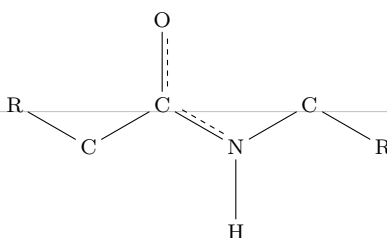
6 Delocalized double bonds

It is sometimes necessary to draw a double bond so that one line segment is dashed while the other is solid¹⁰. This feature is not hard-coded into **ChemFig**; instead *tikz*, its “decorations” library and its programmable styles make it possible.

First of all, after having loaded the “decorations” library by putting `\usetikzlibrary{decorations}` in the preamble, a decoration named “*ddbond*” (for Dashed Double Bond) is defined (lines 1 to 14) and then two *tikz* styles called “*lddbond*” and “*rddbond*” (lines 15 and 16). For the former, the dashed line segment is on the left of the solid middle segment; for the latter it is on the right. These styles can be called with the fifth optional argument of bonds:

Custom decorations

```
\pgfdeclaredecoration{ddbond}{initial}
{
  \state{initial}[width=4pt]
  {
    \pgfpathlineto{\pgfpoint{4pt}{0pt}}
    \pgfpathmoveto{\pgfpoint{2pt}{2pt}}
    \pgfpathlineto{\pgfpoint{4pt}{2pt}}
    \pgfpathmoveto{\pgfpoint{4pt}{0pt}}
  }
  \state{final}
  {
    \pgfpathlineto{\pgfpointdecoratedpathlast}
  }
}
\tikzset{lddbond/.style={decorate,decoration=ddbond}}
\tikzset{rddbond/.style={decorate,decoration={ddbond,mirror}}}
\setatomsep{4em}
\chemfig{[: -30]R-C-[: :60]C(-[: :60,,,rddbond]O)-[,,,lddbond]N(-[: -60]H)-[: :60]C-R}
```



7 Saving a sub-molecule

ChemFig is capable of saving a `<code>` as an alias for reuse in a more compact form in the code of a molecule. This is particularly useful when the `<code>` appears several times.

To do this, one gives the command

```
\definesubmol{<name>}{<code>}
```

which saves the `<code>` for recall in the code of the molecule via the shortcut “`!{<name>}`”. This `<name>` can be:

- a sequence of characters: all the alphanumeric characters able to be between `\csname` and `\endcsname` are accepted;

¹⁰Thanks to Andreas BRÖERMANN who suggested this feature and who gave me the solution to this problem.

- a control sequence.

In all cases, if the alias is already defined you should not overwrite it with a new definition using `\definesubmol`. A warning will be issued to the user that the old alias will be overwritten by the new one. To override the definition of an alias made previously, use:

$$\backslash\text{redefinesubmol}\{\langle\text{name}\rangle\}\{\langle\text{code}\rangle\}$$

Here is a code which draws the pentane molecule. An alias “xy” was defined beforehand for the code CH₂:

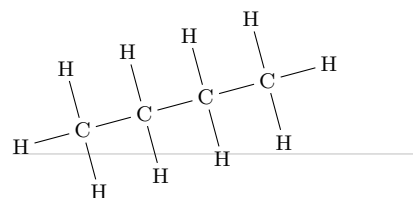
```
\definesubmol{xy}{CH_2}
\chemfig{H_3C-!{xy}-!{xy}-!{xy}-CH_3}
```



In this case the technique is not very interesting because “!{xy}” is just as long to type as the code it replaces.

But in certain cases, this feature saves a lot of space in the code of the molecule and increases readability. In the following example, we draw the complete structural diagram of butane. We will define an alias with the control sequence “\xx” for the sub-molecule CH₂. If we use only relative angles, it is possible to rotate the entire molecule to any given angle by using the optional global angle parameter which specifies the default bond angle of the main molecule. It is set to 15° here:

```
\definesubmol\xx{C(-[:+90]H)(-[:-90]H)}
\chemfig{[:15]H-!\xx-!\xx-!\xx-!\xx-H}
```



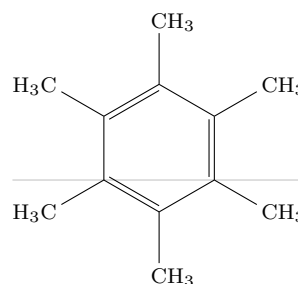
The `\definesubmol` command takes an optional argument; its syntax is as follows:

$$\backslash\text{definesubmol}\{\langle\text{name}\rangle\}[\langle\text{code1}\rangle]\{\langle\text{code2}\rangle\}$$

When the optional argument is present, the alias “!<name>” will be replaced by <code1> if the bond which arrives at the alias comes from the right, i.e., if the angle which the arriving bond makes is between but is not equal to 90° and 270°. For all the other cases where the bond arrives from the left of vertically, the alias will be replaced by <code2>.

We will define a control sequence `\Me` pour “methyl” so that the alias “!\Me” will be replaced by “H₃C” when the bond arrives from the right and by “CH₃” when it arrives from the left. We can observe in the example that with this alias we need no longer worry about the angle:

```
\definesubmol\Me[H_3C]{CH_3}
\chemfig{*6((-!\Me)=(-!\Me)-(-!\Me)=(-!\Me)-(-!\Me)-)}
```



8 Decorations

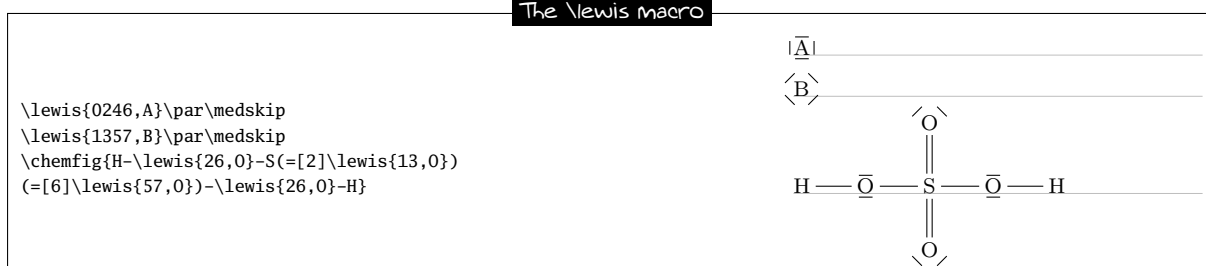
8.1 Lewis diagrams

The macro `\lewis` allows placement of pairs of electrons, of single electrons, or of empty slots. This syntax is used:

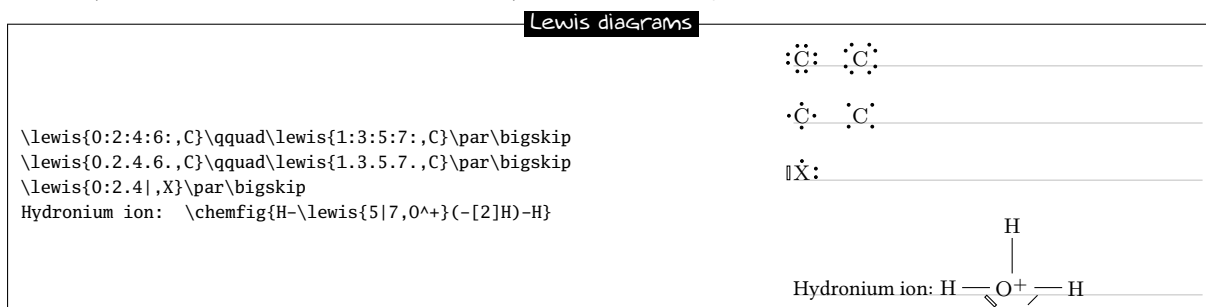
$$\backslash\text{lewis}\{\langle\text{n1}\rangle\langle\text{n2}\rangle\dots\langle\text{ni}\rangle,\langle\text{atom}\rangle\}$$

where the `<n1>...<ni>` represent the desired positions (in multiples of 45°) around the `<atom>`. These whole numbers must be between 0 and 7.

This command can also be used inside the argument of `\chemfig`:

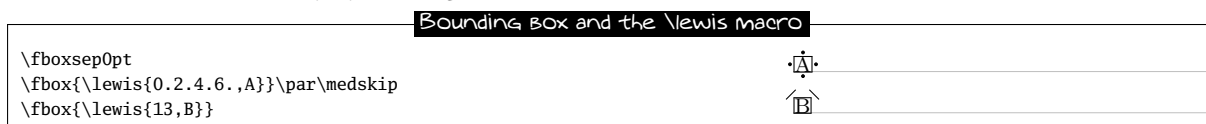


If one wishes to draw two electrons instead of a line, follow the integer with a “:”. If one wishes to draw a single electron, follow it with a “.”. To draw a lacuna, follow it with a “|”:



All the decorations drawn by `\lewis` are not included in the bounding box of the atom; they are drawn afterwards. A consequence of this is seen in the two examples above, where the frame does not appear to be properly fitted to the drawing of the molecule, which extends downward slightly. This will be seen more often in this the “Decorations” chapter, which presents commands which do not change the bounding box.

This can be seen more clearly by drawing an `\fbox` around decorated atoms:



Several parameters can be set with the help of the macro

$$\setlewis[\dim1]{\dim2}{\dim3}{\tikz code}$$

If an argument is empty, it takes its default value.

- `<dim1>` is the width of the rectangle which represents the empty slot obtained with the character “|”;
- `<dim2>` is the distance between the bounding box and the decoration. It is $0.2ex$ by default;
- `<dim3>` is the length of the line segment representing a pair of electrons. It is $1.5ex$ by default;
- `<code tikz>` is code which is passed directly to `tikz`. This code is empty by default.



8.2 Stacking characters

The macros

$$\chemabove[\dim]{\code}{\stuff}$$

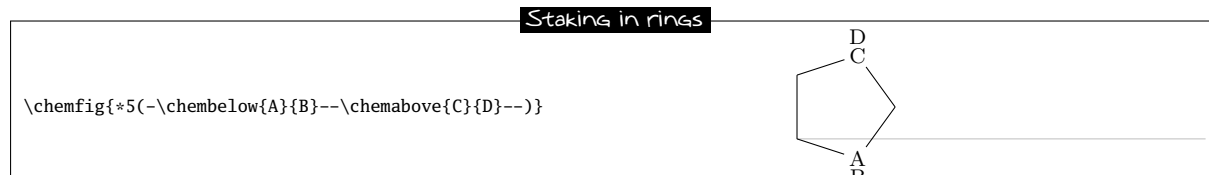
and

$$\chembelow[\dim]{\code}{\stuff}$$

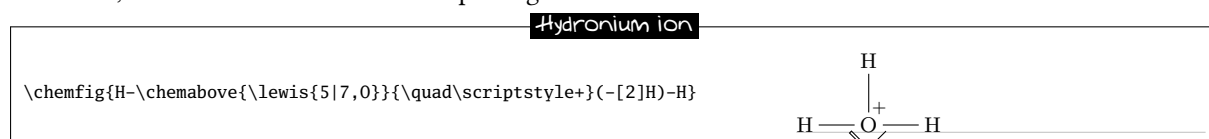
place the `<stuff>` above and below the `<code>` respectively at a vertical distance `<dim>`, without changing the bounding box of `<code>`. The length `<dim>` is 1.5pt by default.

These commands are independent of the macro `\chemfig` and can be used either inside or outside its argument.

They are especially useful in rings, if care is taken to put braces around the letters A, B, C and D in order to prevent `ChemFig` from starting a new atom on these letters:



They are sometimes useful for placing pseudo-exponentsring!charge which do not change the bounding box of the atoms, so that the bonds do not end up being too short:



8.3 Chemical reactions

To write chemical reactions, `ChemFig` provides a command for the signs and a command for the arrows.

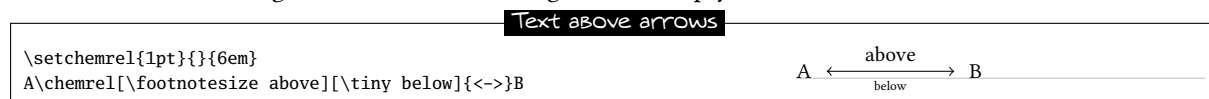
The command `\chemsign[<dim>]<sign>` typesets the `<sign>`, which is surrounded on both sides by an unbreakable horizontal space `<dim>` defaulting to 0.5em.

The command `\chemrel[<arg1>][<arg2>]{<arrow code>}` draws an arrow where the optional arguments `<arg1>` and `<arg2>` are placed above and below the arrow respectively, without modifying its bounding box. The `<arrow code>` is passed directly to `tikz` except when it involves “<”, which allows two arrows to be drawn one above the other.

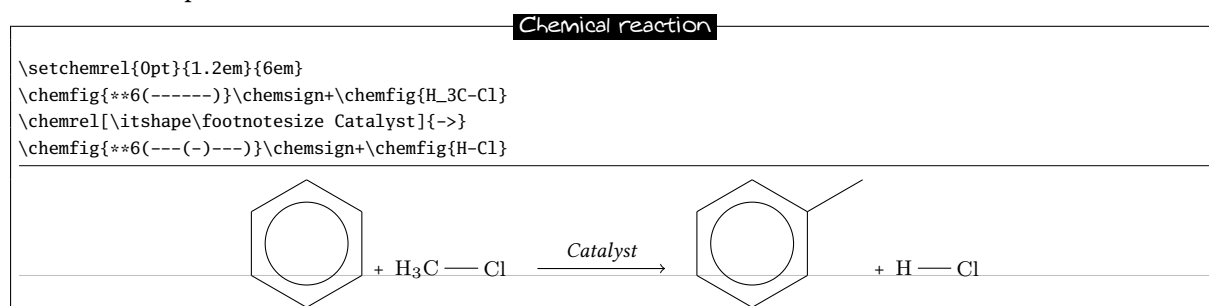


The command `\setchemrel{<dim1>}{<dim2>}{<dim3>}` allows settings of the dimensions used in drawing the arrow:

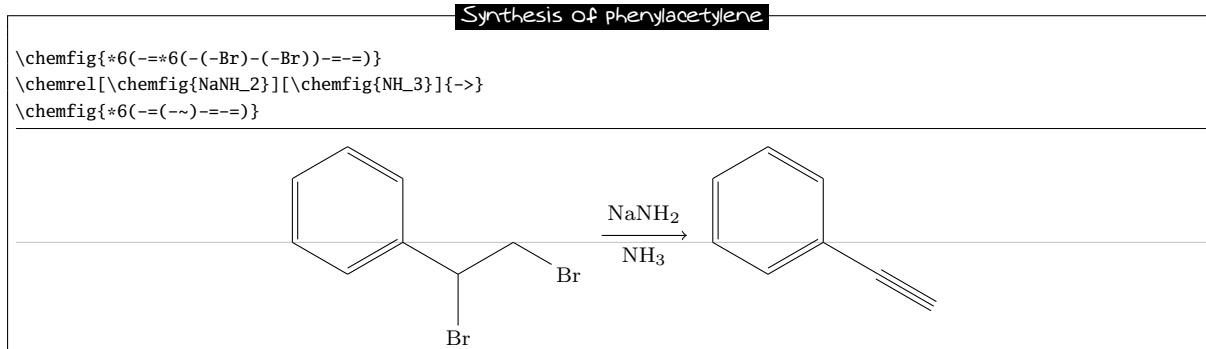
- `<dim1>` is the vertical spacing between the arrow and the optional text above and/or below it. If the argument is empty, it defaults to 2pt;
- `<dim2>` is the unbreakable horizontal space inserted before and after the arrow. If the argument is empty, it defaults to 0.7em;
- `<dim3>` is the length of the arrow. If the argument is empty, it defaults to 4em.



Here is an example of a chemical reaction:

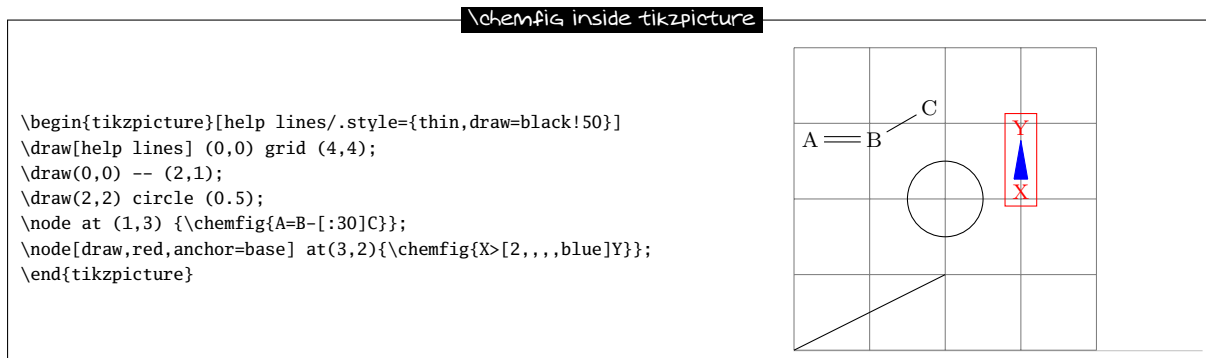


And another:



9 Using `\chemfig` in the `tikzpicture` environment

It is possible to call the `\chemfig` inside a `tikzpicture` environment:

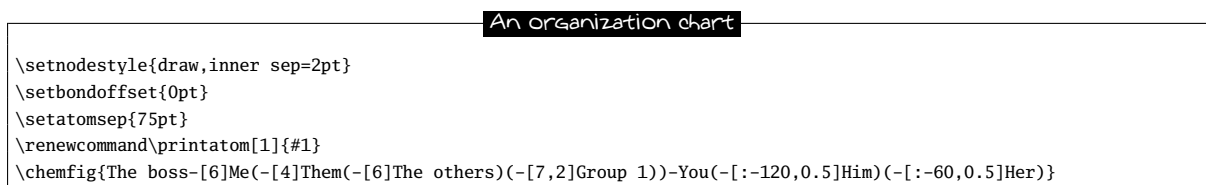


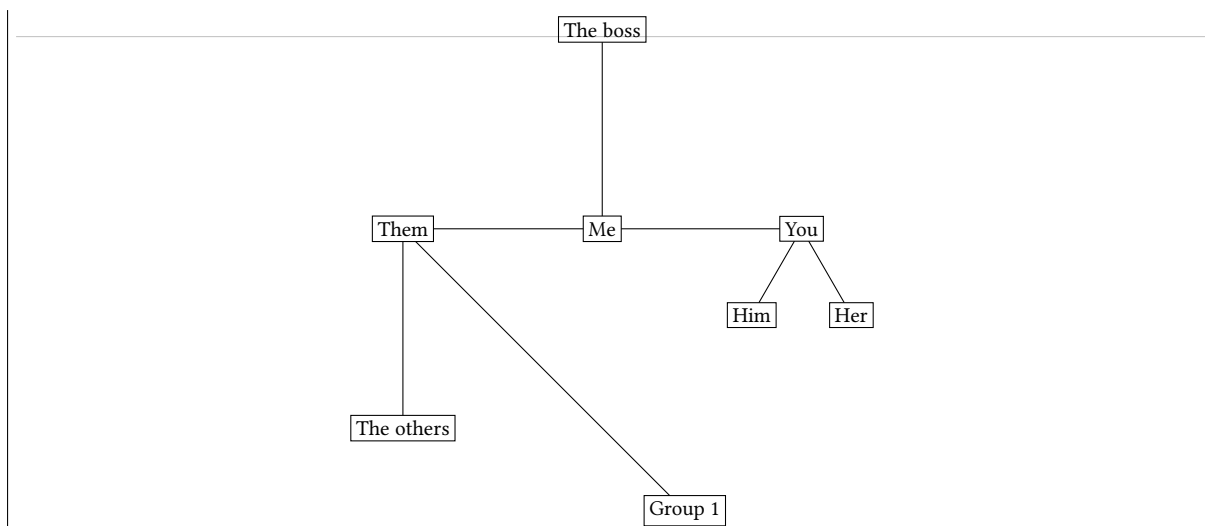
10 Beyond chemistry

At heart `ChemFig` is a tool for drawing graphs, and this tool has been programmed to adapt it for chemistry. In some ways it is possible to return `ChemFig` to its roots to draw organization charts or other diagrams represented by graphs.

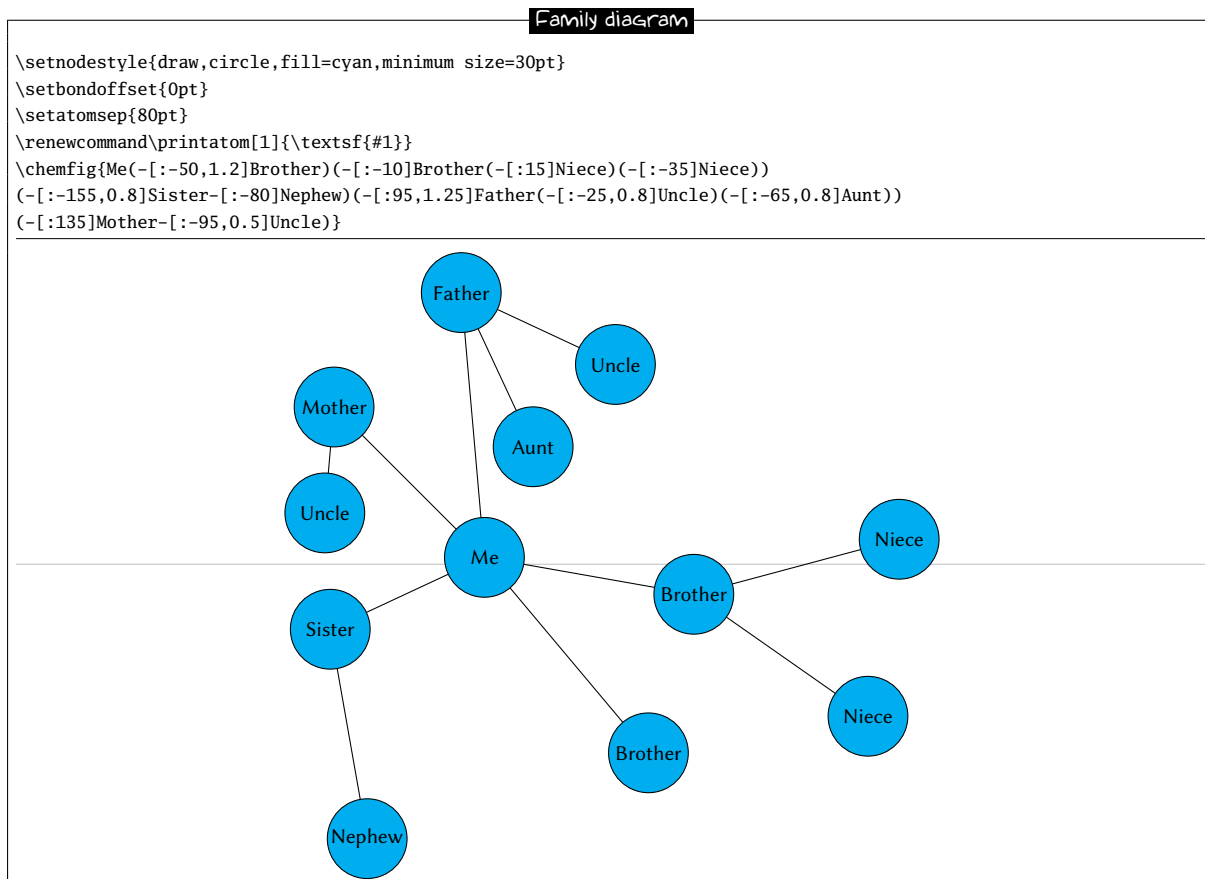
Each atom is contained in a `tikz` node. By default these nodes have an “inner sep” and an “outer sep” equal to 0pt. They are rectangular as seen on page 10. These defaults can be overwritten with the macro `\setnodestyle`, the argument of which is passed to `tikz` and specifies the style of the nodes containing the atoms.

In this example we specify only “draw,inner sep=2pt”, which has the effect of drawing the outline of the nodes and separating the outline and node contents by 2pt. We also specify `\setbondoffset{0pt}\setbondoffset` so that the bonds touch the edges of the nodes. The interatomic spacing is increased to 75pt. Finally, the command `\printatom` is made as simple as possible so that math mode is no longer used and spaces are thus preserved.





Here is another organization chart where the nodes are circular and coloured cyan:

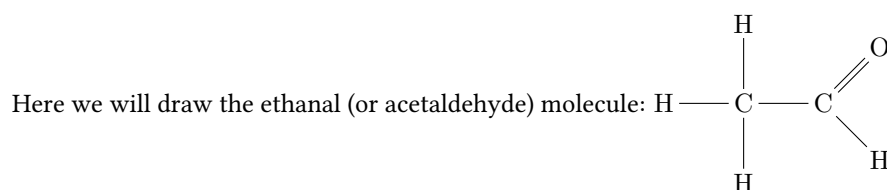


11 Annotated examples

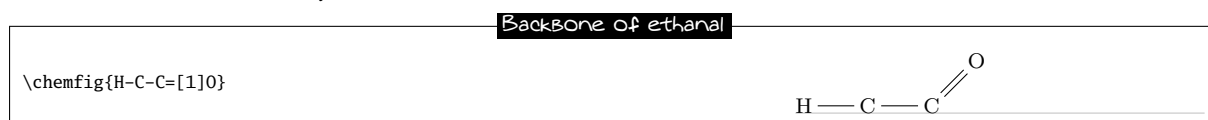
In this chapter, several molecules will be drawn, putting into use the methods previously described. The aim here is to show a logical order for putting together a molecule so that the user unfamiliar with **ChemFig** will learn how to construct complex molecules. The construction steps will be shown to help with this learning process.

In addition, several possibilities – some intuitive and others less so – will be shown which give the same graphical results, with the objective being to show that **ChemFig** allows some flexibility in encoding molecules. One can see how each is put together and adopt the methods with which one is most comfortable.

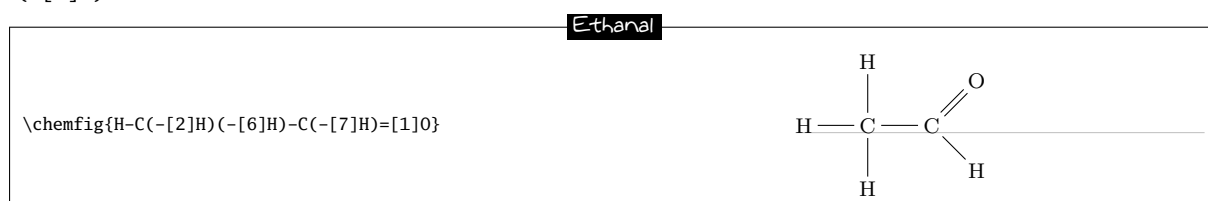
11.1 Ethanal



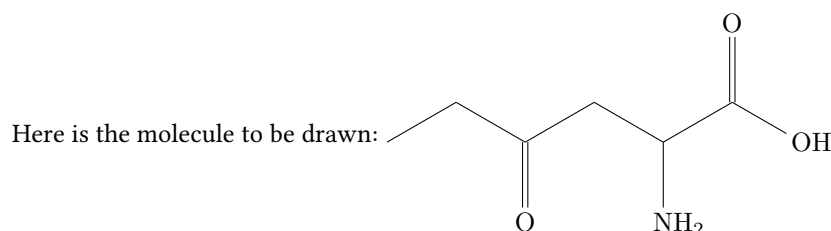
The best method for non-cyclic molecules is to select the longest chain. Here one could take “H-C-C=O” for example. The bond C=O is tilted to 45° by using the predefined angle “[1]”. This give a “backbone” of the molecule to which the branches merely have to be added:



The three hydrogen atoms still have to be placed at the correct orientation with the help of predefined angles. The first is at 90° with the branch “(-[2]H)”, the second at 270° with “(-[6H])”, and the one on the right at 315° with “(-[7]H)”:



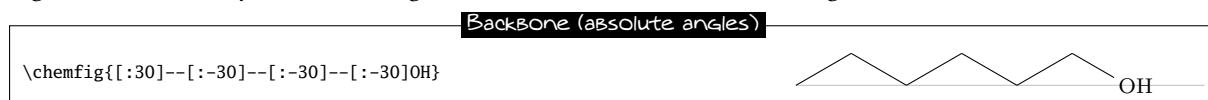
11.2 2-amino-4-oxohexanoic acid



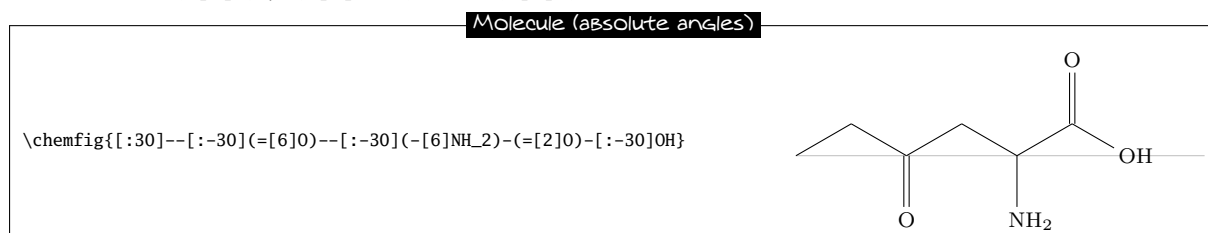
As is often the case for most molecules, there are several methods and for each several different ways of getting the result. Here we will look at four different methods.

11.2.1 Absolute angles

We will first of all draw the central chain with absolute angles. We set the default angle to +30° with the optional argument, and so only the descending bonds need to have their absolute angle set to -30°:

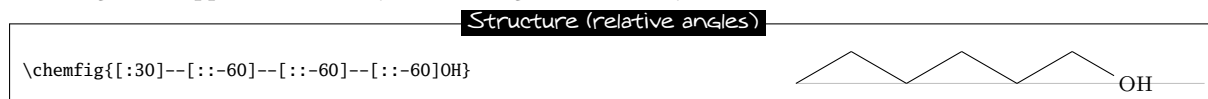


The branches “(=[6]O)”, “(-[6]NH₂)” and “(=[2]O)” still have to be added to the correct vertices:

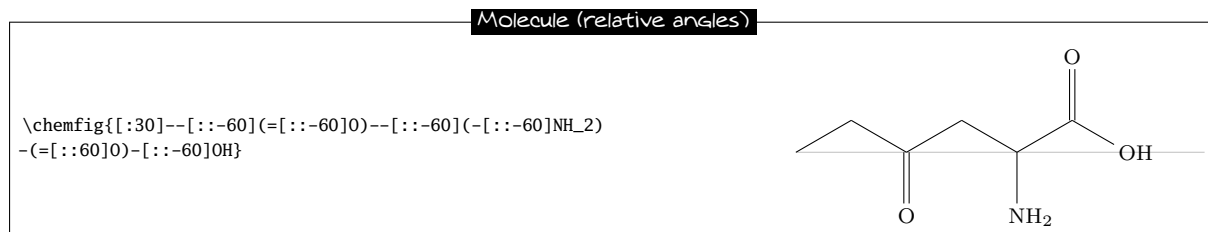


11.2.2 Relative angles

A more general approach uses only relative angles, in this way:



then

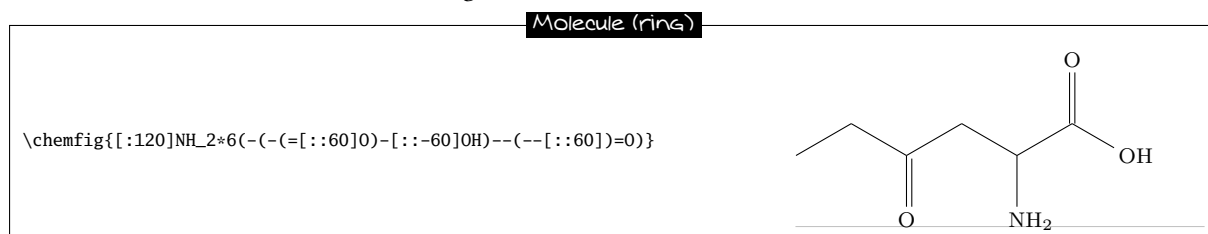


11.2.3 Ring

Since the angles between the bonds are 120° , it is possible to use a 6-ring, although this method is less natural. Here we take advantage of the fact that a ring can be left unfinished. The ring must be rotated 120° so that the first vertex is to the south-east of the ring:



Now the branches must be added to the right vertices:



11.2.4 Nested rings

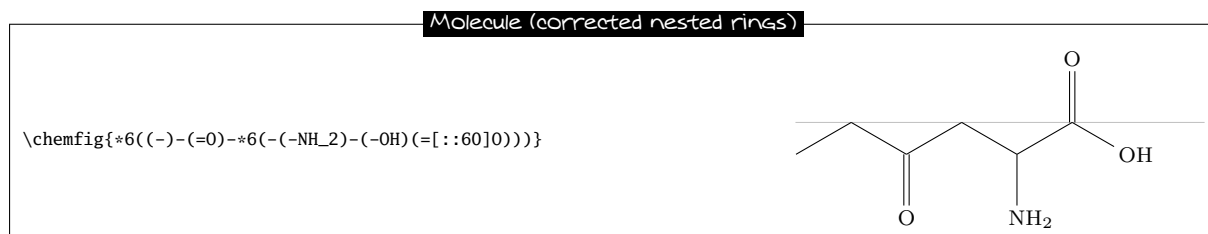
Delving deeper into the ring method, we can also consider nesting incomplete 6-rings. We could start with this backbone:



And then add the bonds which leave the vertices of these rings. There are no angles to worry about because the bonds leaving the rings are the bisectors of the sides of the ring, exactly what we want here:



A close look shows that the second line segment of the double bond to the oxygen atom is *inside* the incomplete 6-ring¹¹ Despite its brevity, this code does not give a perfect drawing. This can of course be corrected by adding a little to the code:



11.3 Glucose

The goal here is to represent the glucose molecule according to several different conventions.

11.3.1 Skeleton diagram

The code here looks like that of 2-amino-4-oxohexanoic acid. This gives almost the same structure with absolute angles, except here the default angle is -30° :

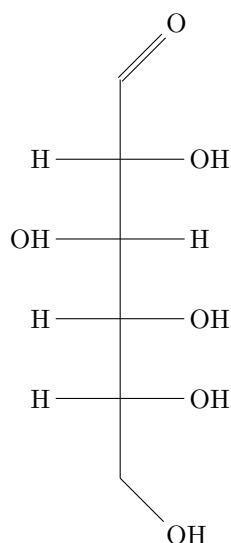


Adding the branches is no problem. We use predefined absolute angles:



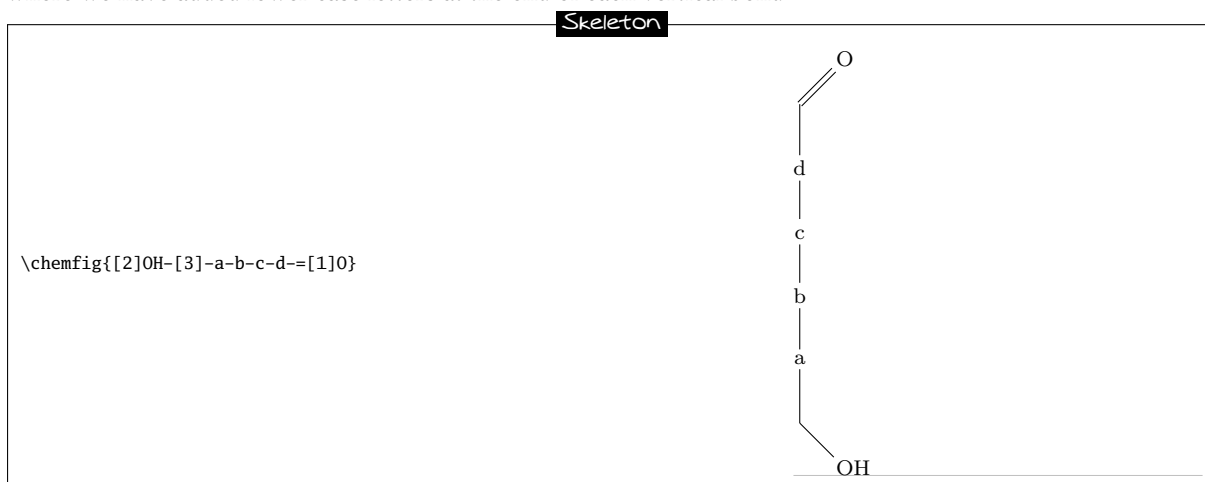
11.3.2 Fisher projection

The goal is to get the molecule below:

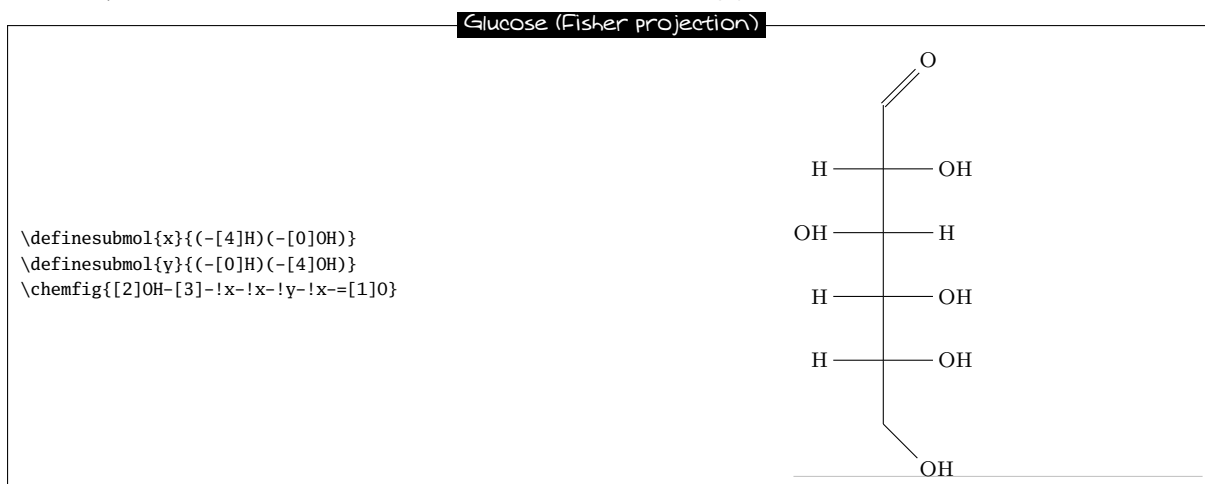


¹¹This was also true for the preceding method with one ring.

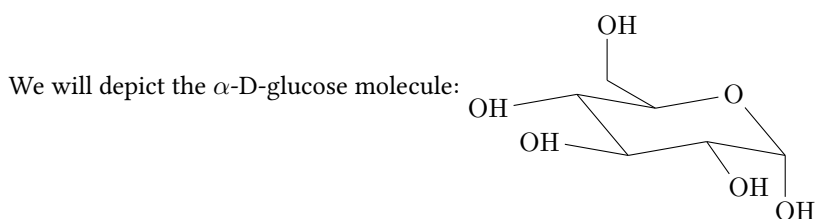
The idea is to begin to draw the longest chain vertically by giving a default angle of “[2]”. Here is the skeleton, where we have added lower case letters at the end of each vertical bond:



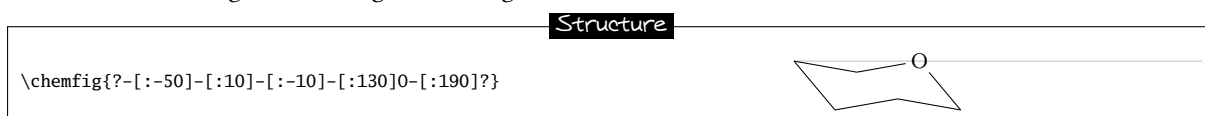
Next we define two aliases for the horizontal bonds and the atoms at their ends. Lets choose “x” which we will put in place of the lower case a, c and d, and “y” which will replace the letter c. Since these aliases are just one character, we do not need braces and can write “!x” instead of “!{x}”:



11.3.3 “Chair” representation



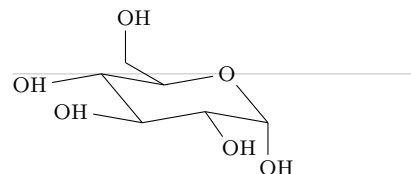
To do this, we will first of all draw five sides of the chair and link the first vertex to the last with a hook “?”. We will use the following absolute angles, running counterclockwise: -50° , 10° , -10° , 130° , 190° .



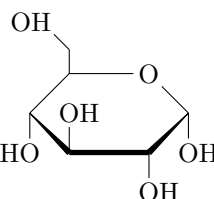
Now we simply add the branches inside parentheses. The angles are chosen to give the best impression of perspective, and some bonds are shortened by a factor of 0.7:

Chair representation

```
\chemfig{?(-[:190]OH)-[:50](-[:170]OH)-[:10](-[:55,0.7]OH)
-[:10](-[:6,0.7]OH)-[:130]O-[:190]?(-[:150,0.7]-[:2,0.7]OH)}
```



11.3.4 Haworth projection



The goal is to depict this D-glucopyranose molecule:

First of all we will choose the longest chain, which starts at the “HO” group on the left and continues through five sides of the ring. The ring will be closed with a hook. For the vertical bond which leaves from the first “HO” group, we need to specify that it will leave from the second atom using the optional argument. Furthermore, it will be shortened with a coefficient of 0.5. Its optional argument will thus be “[2,0.5,2]”.

Next, to give the impression of perspective to the ring, the diagonal bonds will be shortened by a coefficient of 0.7. For the bold diagonal lines we will use Cram bonds, having redefined the base of the triangles to be 2pt. The bold horizontal bond needs to be drawn with a thickness of 2pt, and so its optional argument will be “[0,,,line width=2pt]”. Here is the skeleton of the molecule:

Structure

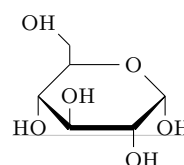
```
\setcrambond{2pt}{}{}
\chemfig{HO-[2,0.5,2]?<[7,0.7]-[,,,
line width=2pt]>[1,0.7]-[3,0.7]O-[4]?}
```



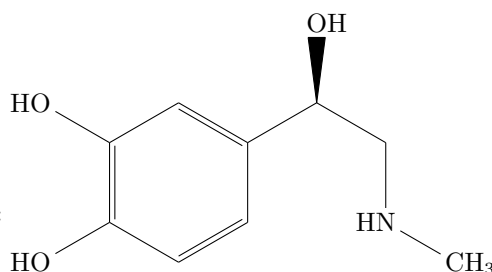
All that needs to be done now is to add the branches at the correct places, giving the right absolute angles and sometimes reducing the length to better give the illusion of perspective:

Haworth projection

```
\setcrambond{2pt}{}{}
\chemfig{HO-[2,0.5,2]?<[7,0.7](-[2,0.5]OH)-[,,,
line width=2pt](-[6,0.5]OH)>[1,0.7](-[6,0.5]OH)-[3,0.7]
O-[4]?(-[2,0.3]-[3,0.5]OH)}
```



11.4 Adrenaline



We want to draw the adrenaline molecule:

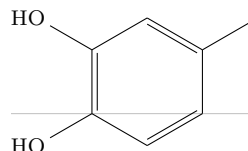
We are going to use two different methods.

11.4.1 Using one ring

First of all, we start with a 6-ring and we draw the start of the branches which leave it:

Skeleton of adrenaline

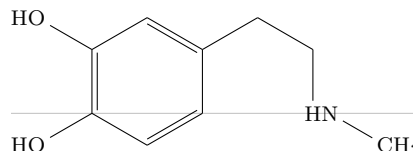
```
\chemfig{*6((-HO)--(-)--(-HO)=)}
```



The branch on the right still needs to be completed using, for example, relative angles:

Adrenaline, step two

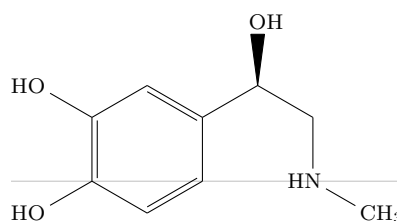
```
\chemfig{*6((-HO)--(--[: -60]-[: -60]
HN-[: +60]CH_3)--(-HO)=)}
```



Then we need to add a Cram-bonded OH and indicate that the bond which arrives at “HN” does so on the second atom, i.e., “N”. We use the fourth optional argument of the bond:

Adrenaline

```
\chemfig{*6((-HO)--(-(<[: :60]OH)-[: -60]-[: -60, , 2]
HN-[: +60]CH_3)--(-HO)=)}
```



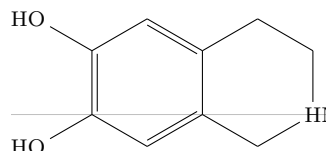
11.4.2 Using two rings

This method is less natural, but the goal is to show here how to make a bond invisible.

We could improve this code by considering that the drawing of the adrenaline molecule is made of two 6-rings adjacent to each other:

Adrenaline, two-ring skeleton

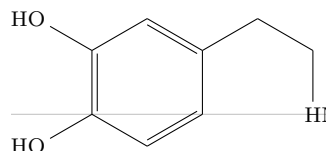
```
\chemfig{*6((-HO)--*6(--HN---)--(-HO)=)}
```



Now the first two bonds of the ring on the right need to be made invisible. To do this we use the argument that is passed to `tikz`, specifying “draw=none”. These bonds will thus have this code: “-[,,,draw=none]”. To keep the code readable, we define an alias named “&” for these bonds:

Adrenaline, step two

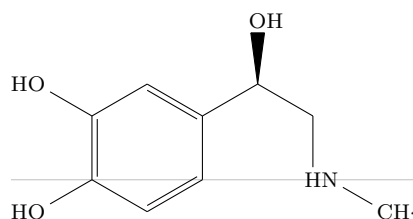
```
\definesubmol{&}{-[,,,draw=none]}
\chemfig{*6((-HO)--*6(!&!&HN---)--(-HO)=)}
```



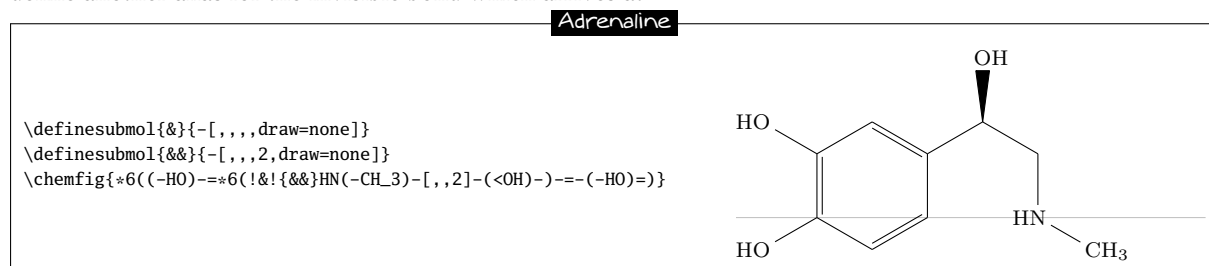
The rest becomes easy; just add the branches to the right vertices:

Adrenaline, step three

```
\definesubmol{&}{-[,,,draw=none]}
\chemfig{*6((-HO)--*6(!&!&HN(-CH_3)--(<OH)-)--(-HO)=)}
```

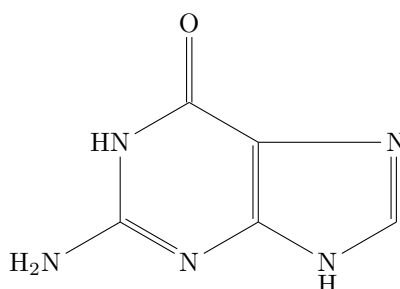


To finish, we specify that the bonds that *arrive at and leave from* “HN” must do so at the second atom. We therefore define another alias for the invisible bond which arrives at “HN”:



11.5 Guanine

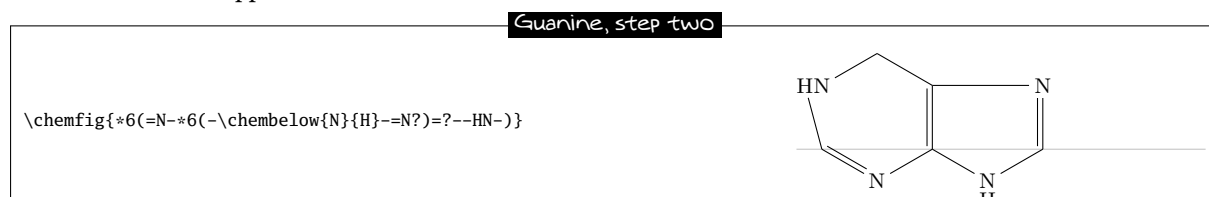
We will draw the guanine molecule:



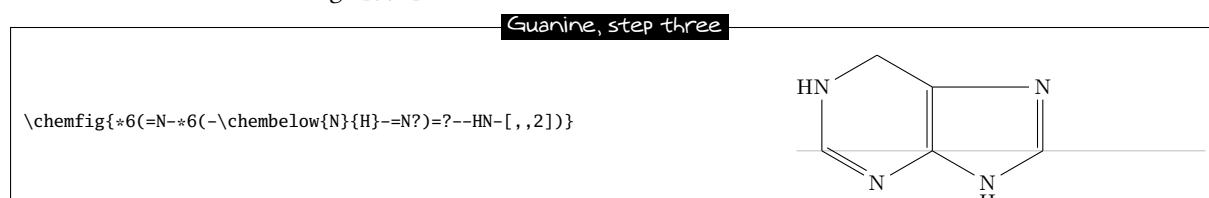
First of all, let's begin by drawing the nested rings, putting just the nitrogen atoms at the vertices:



Then we can draw the horizontal bond in the right ring with a hook. We will also place a hydrogen atom under the nitrogen atom of the 5-ring with the command `\chembelow{N}{H}`. We also need to write “HN” instead of “N” at the vertex at the upper left of the molecule:



We note that one bond leaves from the wrong atom¹²! The automatic calculation mechanism must be corrected so that the bond leaves from the second atom “N” instead of the first. To do this we give an optional argument for the last bond of the first 6-ring “[, , 2]”:

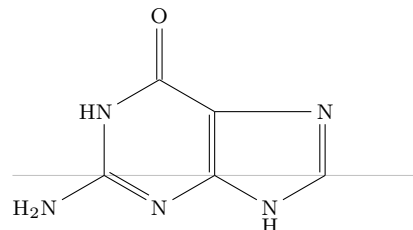


Simply add the branches to the right vertices. Note especially the branch leaving the first vertex of the first 6-ring “(-N_2N)”:

¹²This seems illogical because the angle of the bond from the HN group toward the first vertex lies between -90° and 90° ; **ChemFig** should therefore leave from the second atom. To explain this contradiction, one must know that in rings, the last bond always links the last vertex to the first, ignoring the *calculated theoretical* angle of this bond (which here is -90°). **ChemFig** uses this theoretical angle to determine the departure and arrival atoms, but does not use it to draw the bond because the two ends are already defined. The departure atom for the last bond is thus the first atom.

Guanine

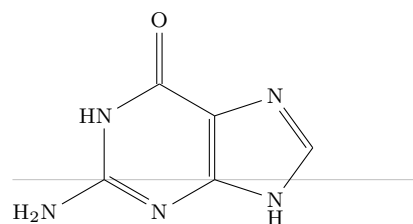
```
\chemfig{*6((-H_2N)=N-*6(-\chembelow{N}{H}-=N?)=?-(=O)-HN-[ , , 2])}
```



We could also draw the same molecule with a regular 5-ring, as is sometimes done:

Guanine with 5-ring

```
\chemfig{*6((-H_2N)=N-*5(-\chembelow{N}{H}-=N)-=(=O)-HN-[ , , 2])}
```



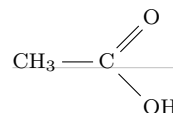
12 How to ...

12.1 Write a colored atom

Since the package `xcolor` is loaded by `tikz`, itself loaded by `ChemFig`, we can write color commands in the code of a molecule, mainly `\color` and `\textcolor`. The atoms are displayed in `tikz` nodes which behaves like boxes of \TeX and it is as if these atoms were put in a group. Therefore, the color change remains local to the atom.

Colors

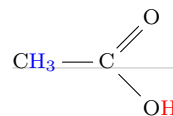
```
\chemfig{C\color{blue}H_3-C(=[1]O)-[7]O\color{red}H}
```



This code does not work, because of the rule used to separate atoms: here, the first atom starts at “C” and spreads to the next uppercase letter. Therefore, this atom is “`C\color{blue}`” and the color change occurs at the end of atom and has no effect. We need to force `ChemFig` to cut the first atom just after “C” with the character “|” and then include `\color{blue}H_3` between braces so that `ChemFig` does not stop the atom 2 before the uppercase “H” which would leave the color change alone and therefore ineffective in an atom:

Colors

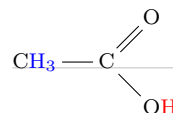
```
\chemfig{C|{\color{blue}H_3}-C(=[1]O)-[7]O|{\color{red}H}}
```



The same effect can be obtained with `\textcolor`:

Colors

```
\chemfig{C|\textcolor{blue}{H_3}-C(=[1]O)-[7]O|\textcolor{red}{H}}
```



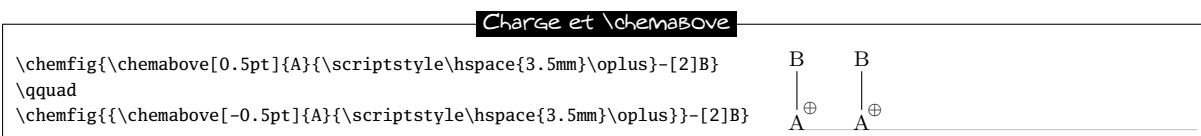
The main disadvantage is that we have to do the same for every atom that need to be colored, even if they are contiguous.

12.2 Add a superscript without modifying a bond

Adding a charge to an atom with a mathematical exponent implies that the box (and therefore the `tikz` node) containing the atom has its dimensions modified. It has no importance when the atom is trailing but the alignment may be compromised if a bond is attached to the atom. The first reaction is to put the charge in a box with no width and therefore use the command `\rlap13`, which often gives good results. We see here that with `\rlap`, the horizontal alignment of atoms is preserved:

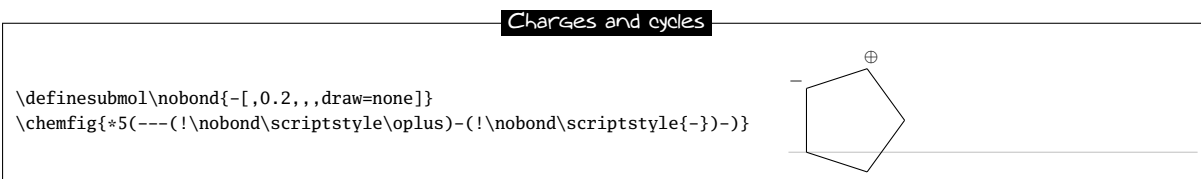


If you want to use the command `\oplus` which displays “ \oplus ”, some could find that the charge is too low: A^{\oplus} . In that case, why not use `\chemabove` to put as precisely as you will, both vertically and horizontally the charge:



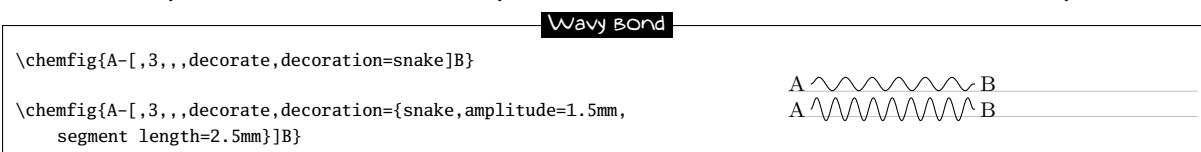
We notice an additional level of braces for the second molecule. Indeed, as we specify “-0.5pt” for the optional argument of `\chemabove` to lower the charge, it is necessary to prevent **ChemFig** to understand the sign “-” as a single bond.

To add a load near the vertex of a cycle, the best method is to attach an invisible bond to this vertex, which is done here with `\definesubmol` with a bond with a length coefficient equal to 0.2:

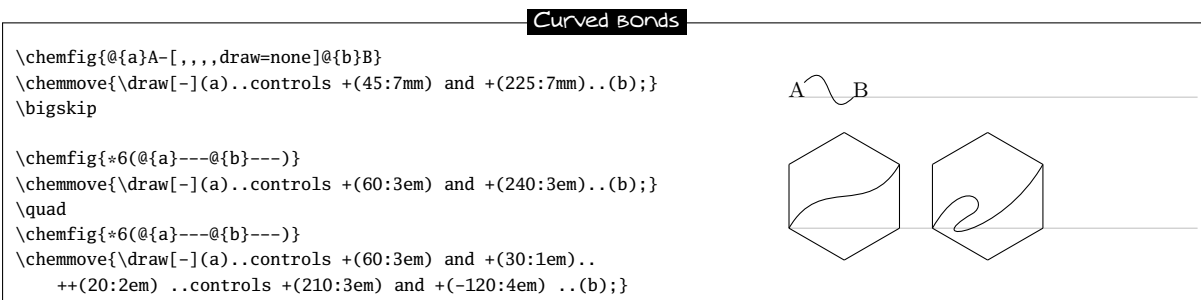


12.3 Draw a curve bond

We have already seen that with the `tikz` library “`decorations.pathmorphing`”, we can draw a wavy bond:



For more flexibility, you can also define nodes using the character “@” and reuse these nodes after the molecule has been drawn to connect them with a curved line using `\chemmove`:



¹³If you have to put the charge at the left of the atom, you must use the command `\llap`.

12.4 Modify the size of a molecule

size of molecules Two parameters determine the default size of a molecule: the font size and the parameter of `\setatomsep` which is 3em the default. We can modify independently these two parameters to change the size of a molecule:

Change the size	
<pre>\definesubmol\hho{H-[:30]O-[:-30]H} \chemfig{!\hho} \setatomsep{2.5em}\chemfig{!\hho} \scriptsize\chemfig{!\hho} \tiny\chemfig{!\hho}\setatomsep{5em}\chemfig{!\hho}</pre>	

You can use the optional second argument of `\chemfig` to *globally* enlarge or reduce a molecule, i.e. the text and links will be reduced by the same ratio:

Global change of the size	
<pre>\definesubmol\hho{H-[:30]O-[:-30]H} \chemfig{!\hho} \chemfig[][scale=0.75]{!\hho} \chemfig[][scale=0.5]{!\hho} \chemfig[][scale=0.33]{!\hho}</pre>	

12.5 Draw a polymer element

The difficulty lies in the display of delimiters (parentheses or brackets) on bond. For this, we will again use the character “@” to define global nodes that will be used later as anchors for delimiters. We will assume that the node for the location of the opening delimiter is called “opbr” and the one of the closing delimiter “clbr”.

We will write a simple macro `\setpolymerbracket`, followed by two characters which define the opening delimiter and closing delimiter.

Then, the macro `\makebraces` has an optional argument of the form “<dim up>, <dim down>”. These two dimensions are the height and depth of the delimiters from the baseline. The mandatory argument of `\makebraces` is the subscript located at the right bottom of the closing delimiter.

Polymers	
<pre>\newcommand\setpolymerdelim[2]{% \def\delimleft{#1}\def\delimright{#2}} \def\definedelimidim#1,#2\enddefinedelim{\def\delimht{#1}\def\delimdp{#2}} \newcommand\makebraces[2][Opt,Opt]{% \definedelimidim#1\enddefinedelim \chemmove{% \node[at=(opbr),yshift=(\delimht-\delimdp)/2] {\llap{\\$left\delimleft\strut\vrule height\delimht depth\delimdp widthOpt \right.\$}};% \node[at=(clbr),yshift=(\delimht-\delimdp)/2] {\rlap{\\$left.\strut\vrule height\delimht depth\delimdp widthOpt \right\delimright_{#2}\$}};% }% } \setpolymerdelim() Polyethylen: \chemfig{\vphantom{CH_2}-[@{opbr,1}]CH_2-CH_2-[@{clbr,0}]\makebraces{n} \bigskip Polyvinyl chloride: \chemfig{\vphantom{CH_2}-[@{opbr,1}]CH_2-CH(-[6]Cl)-[@{clbr,0}]} \makebraces[Opt,15pt]{!\!n} \bigskip</pre>	

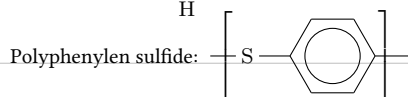
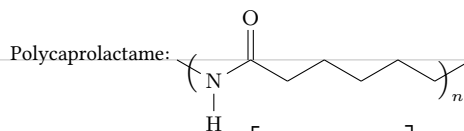
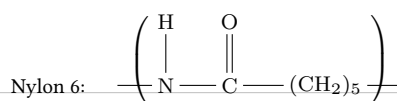
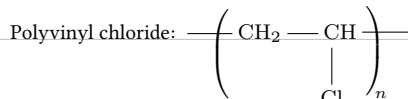
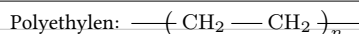
```

Nylon 6:
\chemfig{\phantom{N}-[@{opbr,1}]N(-[2]H)-C(=[2]O)-({CH_2}_5)-[@{clbr,0}]}
\makebraces[20pt,0pt]{ }
\bigskip

Polycaprolactame:\setatomsep{2em}
\chemfig{[:30]-[@{opbr,1}]N(-[6]H)-[:30](=[2]O)--[:30]--[:30]--[@{clbr,0}:30]}
\makebraces[5pt,5pt]{n}

\setpolymerdelim[]
Polyphenylen sulfide:
\chemfig{\vphantom{S}-[@{opbr,1}]S-(*6(---(-[@{clbr,0}]---))})}
\makebraces[15pt,15pt]{ }

```



12.6 Draw the symmetrical of a molecule

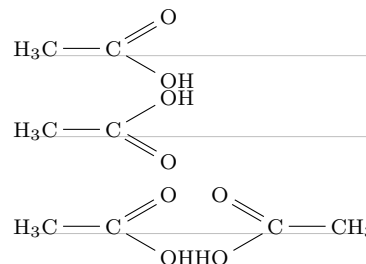
The two commands `\hflipnext` and `\vflipnext` allow to draw the symmetrical of the next molecule about a horizontal or vertical axis. If we want to draw more symmetrical molecules, we need to write these commands before each molecule involved.

Symmetry

```

\chemfig{H_3C-C(=[:30]O)-[:30]OH}% original
\vflipnext
\chemfig{H_3C-C(=[:30]O)-[:30]OH}\medskip
\chemfig{H_3C-C(=[:30]O)-[:30]OH}% original
\hflipnext
\chemfig{H_3C-C(=[:30]O)-[:30]OH}

```



12.7 Add text above bonds and arc to angles

Once we have understood that the character “@” can put a “global” tikz node, that is to say a node accessible after the molecule has been drawn, everything that tikz can do with nodes (that is to say a lot of things) becomes possible.

To write something above or below a bond, we can put two “global” nodes on the atoms at the ends of this bond and write midway of them a text, raised or lowered so that it falls to just above or below the bond. This is done by the macro `\bondname` in the code below ¹⁴.

To draw an arc between two bonds, three atoms are involved on which we have to put three “global” nodes. The macro `\arcbetweennodes` calculates the angle between two lines drawn from a node. Then `\arclabel` draws an

¹⁴The `\si` command is provided by the `siunitx` package.

arc between two bonds and writes a text next to the arc: the optional argument of this macro is the tikz code used to custom the arc. The second argument is the radius of the arc and the following three arguments are the names of global nodes between which the arc must be drawn, the middle name needs to be the vertex of the angle. The last argument is the text to write.

Arcs and text on bonds

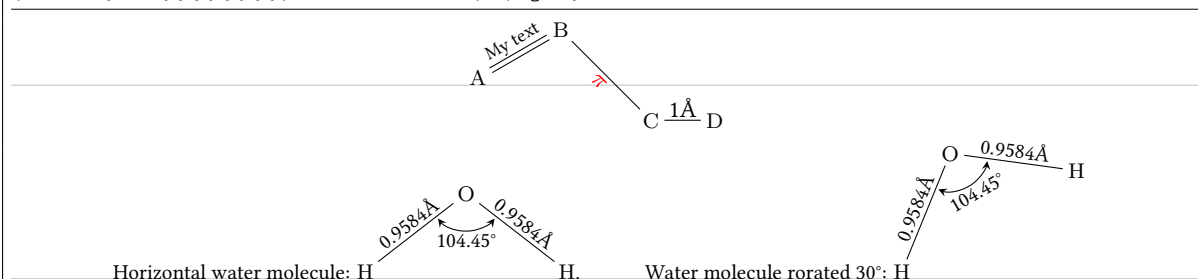
```
\newcommand\namebond[4][5pt]{\chemmove{\path(#2)--(#3)node[midway,sloped,yshift=#1]{#4};}}

\newcommand\arcbetweennodes[3]{%
  \pgfmathanglebetweenpoints{\pgfpointanchor{#1}{center}}{\pgfpointanchor{#2}{center}}%
  \let#3\pgfmathresult}

\newcommand\arclabel[6][stealth-stealth,shorten <=1pt,shorten >=1pt]{%
  \chemmove{%
    \arcbetweennodes{#4}{#3}\anglestart \arcbetweennodes{#4}{#5}\angleend
    \draw[#1]([shift=(\anglestart:#2)]#4)arc(\anglestart:\angleend:#2);
    \pgfmathparse{(\anglestart+\angleend)/2}\let\anglestart\pgfmathresult
    \node[shift=(\anglestart:#2+1pt)#4,anchor=\anglestart+180,rotate=\anglestart+90,inner sep=0pt,
      outer sep=0pt]at(#4){#6};}}
```

```
\chemfig{@{a}A=[:30,1.5]@{b}B-[7,2]@{c}C-@{d}D}
\namebond{a}{b}{\scriptsize My text}
\namebond[-3.5pt]{b}{c}{\small\color{red}$\pi$}
\namebond{c}{d}{\small\si\angstrom}
\medskip

Horizontal water molecule: \chemfig{@{1}H-[:37.775,2]@{2}O-[::-75.55,2]@{3}H}.
\namebond{1}{2}{\footnotesize0.9584\si\angstrom}
\namebond{2}{3}{\footnotesize0.9584\si\angstrom}
\arclabel{0.5cm}{1}{2}{3}{\footnotesize104.45\si\degree}
\quad
Water molecule rotated 30\si\degree: \chemfig{[:30]@1H-[:37.775,2]@2O-[::-75.55,2]@3H}
\namebond12{\footnotesize0.9584\si\angstrom}
\namebond23{\footnotesize0.9584\si\angstrom}
\arclabel{0.5cm}{1}{2}{3}{\footnotesize104.45\si\degree}
```



13 List of commands

The commands created by ChemFig are:

Commands	Description
<code>\chemfig<code></code>	draws the molecule whose design is described by the <code>
<code>\printatom</code>	displays the atoms within the molecules. It can be redefined to customize the output. See page 27
<code>\setnodestyle{<style tikz>}</code>	using tikz syntax, this macro defines the style of nodes containing the atoms. See page 35
<code>\hflipnext</code>	the next molecule will be horizontally flipped
<code>\vflipnext</code>	the next molecule will be vertically flipped
<code>\definesubmol{<name>}[code1]{<code2>}</code>	creates an alias !<nom> which can be put in the code of molecules to be drawn, and which will be replaced with <code1> or <code2> depending on the angle of the last bond. See page 31
<code>\chemskipalign</code>	tells the vertical alignment mechanism to ignore the current group of atoms. See page 29.

<code>\redefinesubmol{<name>}{<code>}</code>	replaces a preexisting alias !<name> with the new <code>. See page 32
<code>\setcrambond{<dim1>}{<dim2>}{<dim3>}</code>	sets the dimensions of the triangles representing Cram bonds: <dim1> is the size of the base, <dim2> is the spacing between the dashes and <dim3> is the side of the dashes. See page 11
<code>\setatomsep{<dim>}</code>	sets the interatomic distance. See page 11
<code>\setbondoffset{<dim>}</code>	sets the space between bonded atoms and the bond. See page 11
<code>\setdoublesep{<dim>}</code>	sets the spacing between the two lines of a double bond. See page 5
<code>\lewis{<codes>,<atom>}</code>	displays the <atom> and places Lewis dot decorations as specified in the <code>. The dots drawn do not change the bounding box. See page 32
<code>\setlewis{<dim1>}{<dim2>}{<tikz code>}</code>	sets the Lewis dot parameters; <dim1> is the distance between the atoms and the decoration, <dim2> is the length of the line representing the pair of electrons and <tikz code> is code which is passed directly to <i>tikz</i> . See page 33
<code>\chemmove[<tikz options>]<tikz code></code>	Makes a <i>tikzpicture</i> environment, adding to it the <tikz options>. Uses the <tikz code> to join the nodes specified in the molecules with the help of the "@" character. See page 22.
<code>\chemsign[<dim>]<sign></code>	draws the <sign>, placing before and after it an unbreakable space of length <dim>. See page 34
<code>\chemrel[<txt1>][<txt2>]{<arrow code>}</code>	draws an arrow described by its <arrow code> and places the optional text <txt1> and <txt2> above and below the arrow respectively. See page 34
<code>\setchemrel{<dim1>}{<dim2>}{<dim3>}</code>	sets the dimensions of the arrow drawn with the command <code>\chemrel</code> : <dim1> is the vertical space between the arrow and the optional text, <dim2> is the unbreakable horizontal space inserted before and after the arrow and <dim3> is the length of the arrow. See page 34
<code>\chemabove[<dim>]{<txt1>}{<txt2>}</code>	writes <txt1> and places <txt2> above, leaving <dim> of vertical space. This command does not change the bounding box of <txt1>. See page 33
<code>\chembelow[<dim>]{<txt1>}{<txt2>}</code>	writes <txt1> and places <txt2> below, leaving <dim> of vertical space. This command does not change the bounding box of <txt1>. See page 33
<code>\chemname[<dim>]{<molecule>}{<name>}</code>	Places <name> under the <molecule>
<code>\chemnameinit</code>	Initializes the greatest molecule depth to ensure correct alignment of the names of the following molecules.

PART V

Gallery

This manual concludes with drawings of molecules of varying complexity.

The curious user can look at the `<code>` of each molecule, though it does become less attractive the more complex the molecule gets. Indeed, beyond a certain level of complexity, though it is fairly easy to write `<code>`, it becomes much harder to read the `<code>` to analyze it afterwards. We quickly reached the limits of immediate readability of the code of a complex drawing.

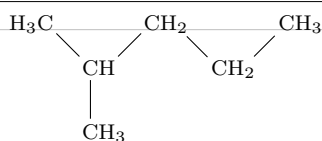
Anyway, I hope that this package will help all \LaTeX users wishing to draw molecules. Given the newness of this package, I hope that you will be forgiving with bugs you encounter and send me an [email](#) to let me know of any malfunctions or suggestions for improvement.

Christian TELLECHEA

*
* *

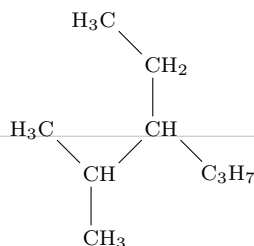
2-methylpentane

```
\chemfig{[7]H_3C-CH(-[6]CH_3)-[1]CH_2-CH_2-[1]CH_3}
```



3-ethyl-2-methylhexane

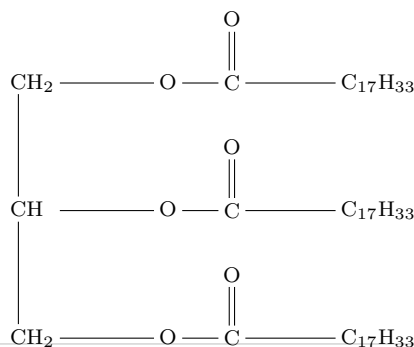
```
\chemfig{H_3C-[7]CH(-[6]CH_3)-[1]CH(-[7]C_3H_7)-[2]CH_2-[3]H_3C}
```



Stearine, condensed structural diagram

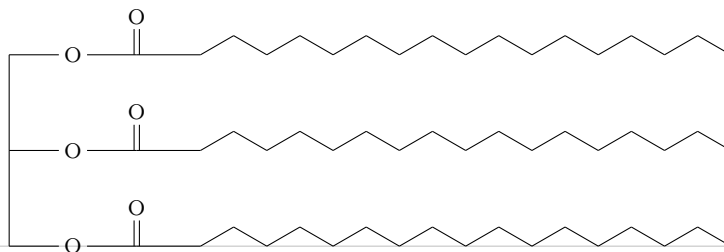
```
\definesubmol{@}{([0,2]-O-[0,1]C(=[2,1]O)-C_{17}H_{33})}
```

```
\chemfig{[2,2]CH_2!@-CH_{\phantom 2}!@-CH_2!@}
```



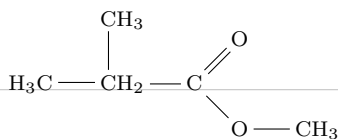
Stearine, skeleton diagram

```
\definesubmol{x}{-[:+30,.6]-[: -30,.6]}
\definesubmol{y}{-0-(=[2,.6]O)-!x!x!x!x!x!x!x}
\chemfig{[2](!y)-[,1.5](!y)-[,1.5](!y)}
```



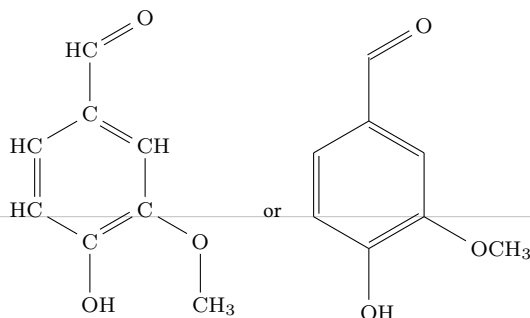
Methyl 2-methylpropanoate

```
\chemfig{H_3C-CH_2(-[2]CH_3)-C(=[1]O)-[7]O-CH_3}
```



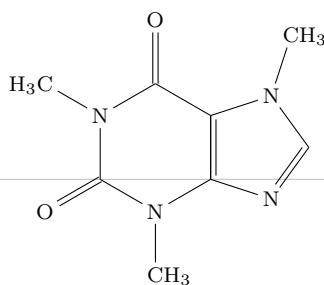
Vanillin

```
\chemfig{HC*6(-C(-OH)=C(-O-[: -60]CH_3)-CH=C(-[, ,2]HC=[: -60]O)-HC=[: ,2])} \quad or \quad
\chemfig{*6(-(-OH)=(-OCH_3)-(-[: -60]O)-=)}
```



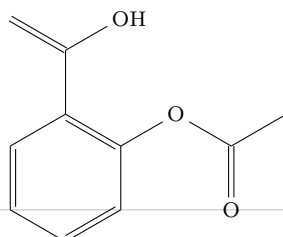
Caffeine

```
\chemfig{*6((=O)-N(-CH_3)-*5(-N=-N(-CH_3)-=)--(=O)-N(-H_3C)-)}
```



Aspirin

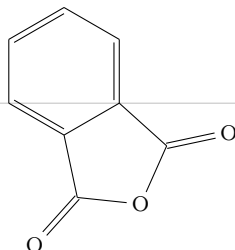
```
\chemfig{*6(=-(-O-[: -60])(=[: -60]O)-[: +60])=(-[: +60])-[: -60]OH)-=)}
```



Aspirin is a registered trademark of Bayer in many countries.

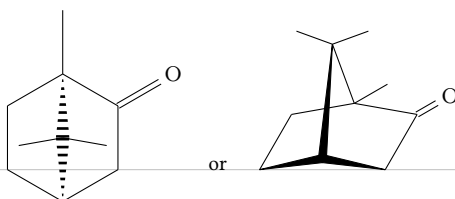
Phthalic anhydride

```
\chemfig{*6(=*5(- (=0)-0-(=0)-)---=)}
```



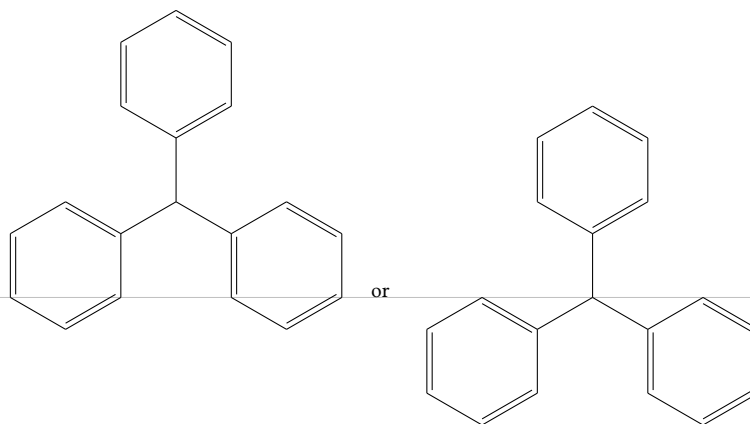
Camphor

```
\chemfig{*6(<[:120](-[:100,0.7])(-[:100,0.7]))--(=0)-(-[:120])--)}
\quad or \quad
\setcrambond{3pt}{}{}
\chemfig{<[:10](>[:85,1.8]?(-[:160,0.6])-[:20,0.6])
>[:10]-[:60](=[:30,0.6]0)-[:170]?(-[:30,0.6])-[:190]-[:240]}
```



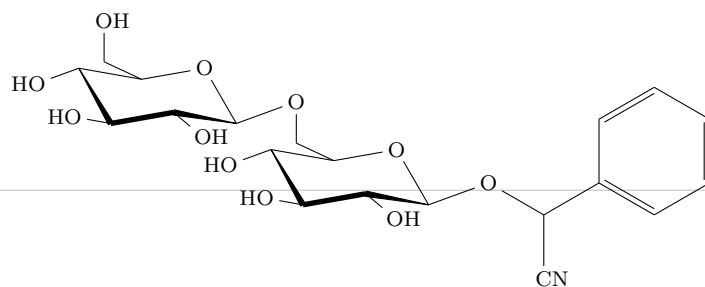
Triphenylmethane

```
\chemfig{*6(--*6(-(*6(-----))-*6(-----))=)}
\quad or \quad
\definesubmol{@}{*6(-----)}
\chemfig{(-[:30]!@)(-[:90]!@)(-[:210]!@)}
```



Amygdalin

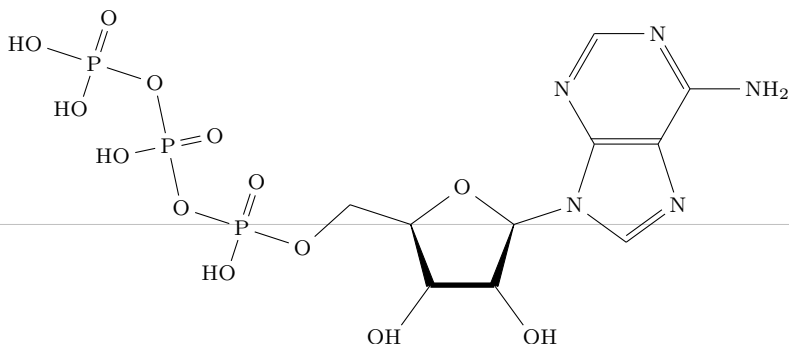
```
\setcrambond{2pt}{}{}
\definesubmol{c1}{-[:200]-[:120]0-[:190]}
\definesubmol{c2}{-[:170](-[:200,0.7]HO)<[:300](-[:170,0.6]HO)
-[:10,,,line width=2pt](-[:40,0.6]OH)>[:10]}
\definesubmol{csub}{-[:155,0.65]-[:90,0.65]}
\chemfig{0(!{c1}(!{csub}0(!{c1}(!{csub}OH)!{c2}))!{c2})-[:30](-[:90]CN)-[:30]*6(-----)}
```



Adenosine triphosphate

```
\setcrambond{3pt}{}
\definesubmol{a}{-P(=[: -90,0.75]O)(-[: :90,0.75]HO)-}
\chemfig{[: -54]*5((-[: :60]O([: -60]!aO([: -60]!aO([: :60]!aHO))))<(-OH)
-[,,, ,line width=2pt](-OH)>(-N*5(=N-*6(-(-NH_2)=N=N-)=_-)-O-)}

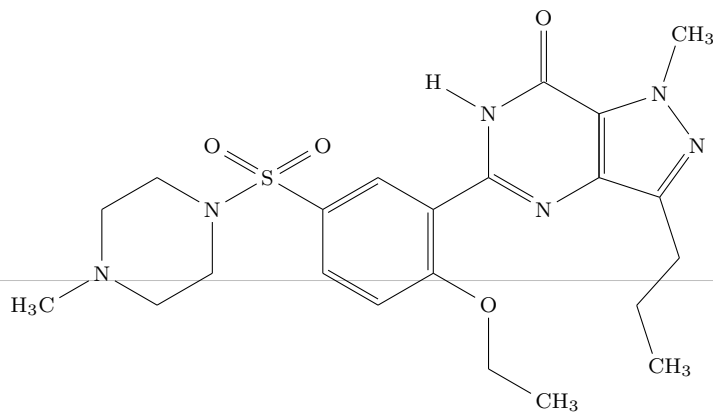
```



Viagra

```
\chemfig{N*6((-H_3C)---N(-S(=[: +120]O)(=[: +0]O)-[: -60]*6(=-(O-[: -60]-[: +60]CH_3)
=(-*6(=N-*5(-(-[: -60]-[: +60]CH_3)=N-N(-CH_3)-)-)--(=O)-N(-H)-)-)=)----)}

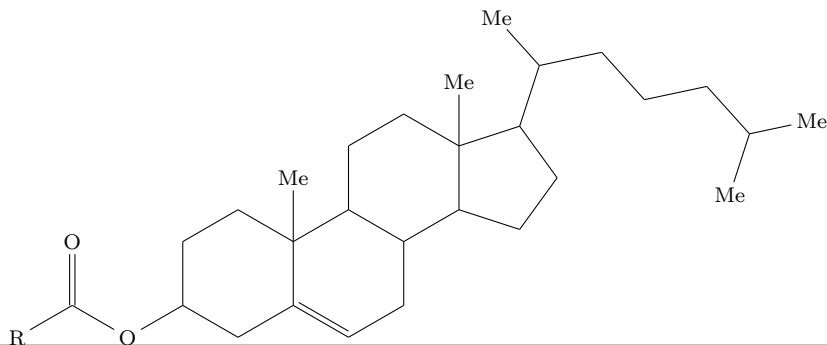
```



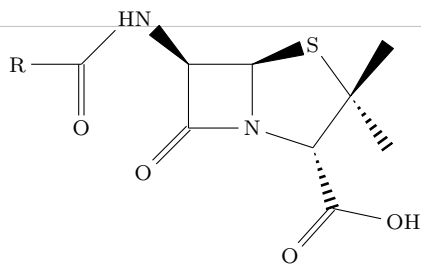
Cholesterol ester

```
\chemfig{[: 30]R-(=[: +60]O)-[: -60]O-*6(---*6(---*5(---(-[: +60]Me)
-[: -60]-[: -60]-[: +60]-[: -60](-[: -60]Me)-[: +60]Me)-(-[: +0]Me)---))--(-[: +0]Me)---)}

```

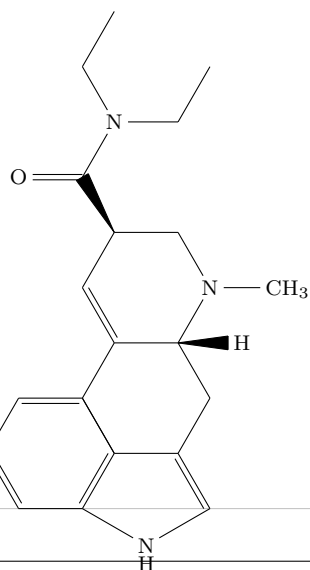



```
-[:+60]OH)-(<[:+0])(<[:-108]-S>--)}
```



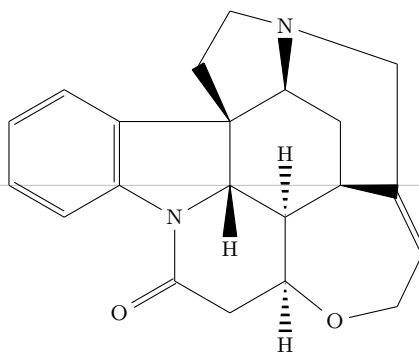
LSD

```
\chemfig{[:150]?*6(=*6(--*6(-N(-CH_3)--(<[:+60]O)-[: -60]N(-[:+60]-[: -60])
-[: -60]-[:+60])=)([: -120]<H)---)*6(-==(-[: -30,1.155]\chembelow{N}{H}?=))}
```



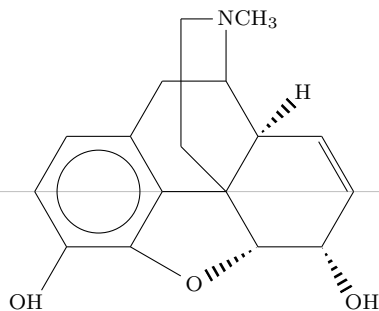
Strychnine

```
\chemfig{*6(=*6(-N*6(-=O)--([:-120]<:H)*7(-0---?0)([: -25.714]-[: ,2]?[1]))
-*6(-?[0,{>}]-(<N?[1]?[2])-([: -90]-[: -60]?[2]))(<[:+0]H)-([:+120]<H)--?)=?=-)}
```



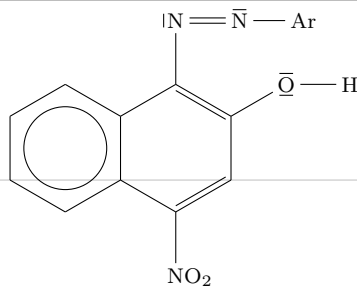
Codeine

```
\chemfig{[: -30]**6(-(-OH)-?*6(-(-[3]-[2,2]-[0,.5])*6(-<[: -150,1.155]0?)
-(<[:OH])--)-(<[:1]H)-(-[2]NCH_3)---)}
```

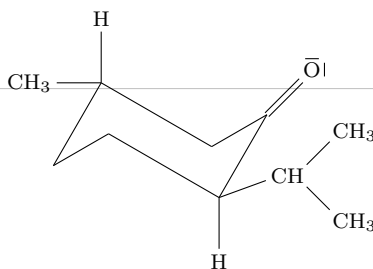
A dye (red)

```
\chemfig{*6(---6(-(-NO_2)--(-\lewis{26,0}-[0]H)=(-\lewis{4,N}=[0]\lewis{2,N}-[0]Ar)-)----)}
```



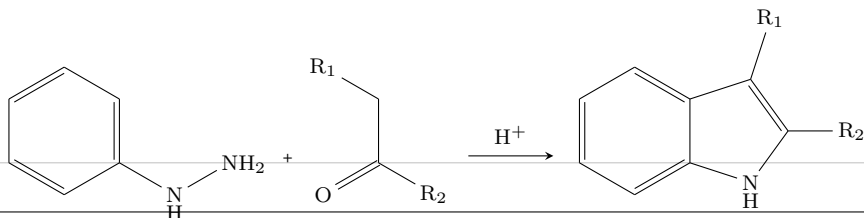
Menthone

```
\chemfig{CH_3-?(-[2]H)(-[::-30,2]-[:::+60](=[1]\lewis{20,0})
-[::-150,1.5](-[:20]CH(-[1]CH_3)(-[7]CH_3))(-[6]H)-[::-90,2]-[:::+60]?)}
```



Fischer indole synthesis

```
\chemfig{*6(==*6(-\chembelow{N}{H}-NH_2)===)}
\chemsign+
\chemfig{([:-150]O)(-[:-30]R_2)-[2]-[:150]R_1}
\chemrel[{\mathrm{H}^+}]{-stealth}
\chemfig{*6(==*5(-\chembelow{N}{H})-(R_2)=(R_1)-)===)}
```



Reaction mechanisms: carbonyl group

```
\chemfig{C([3-])([5-])=[@{db,.5}]@{atoo}\lewis{06,0}}
\chemrel{<>}
\chemfig{\chemabove{C}{\scriptstyle\oplus}([3-])([5-])-\chemabove
{\lewis{026,0}}{\hspace{5mm}\scriptstyle\ominus}}
\chemmove{\draw[->,shorten <=2pt, shorten >=2pt](db) ..controls +(up:5mm) and +(up:5mm)..(atoo);}
```

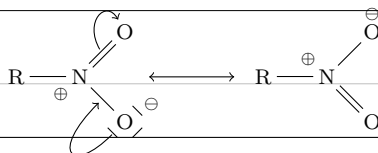


Reaction mechanisms: nitro group

```

\chemfig{R-\chembelow{N}{\hspace{-5mm}\scriptstyle\oplus}([1]=[@{db}]@{atoo1}O)([7]-[@{sb}]@{atoo2}
\chemabove{\lewis{157,0}}{\hspace{7mm}\scriptstyle\ominus})}
\chemrel{<->}
\chemfig{R-\chemabove{N}{\hspace{-5mm}\scriptstyle\oplus}([1]-\chemabove{O}{\scriptstyle\ominus})([7]=O)}
\chemmove{
\draw[->,shorten <=2pt, shorten >=2pt](db) ..controls +(120:5mm) and +(120:5mm)..(atoo1);
\draw[->,shorten <=3pt, shorten >=2pt](atoo2) ..controls +(225:10mm) and +(225:10mm)..(sb);
}

```

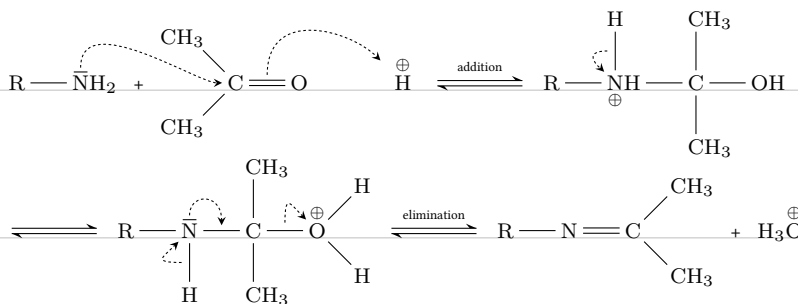


Nucleophilic addition. Primary amines

```

\chemfig{R-@{aton}\lewis{2,N}H_2}
\chemsign{+}
\chemfig{@{atoc}C([3]-CH_3)([5]-CH_3)=[@{atoo1}]O}
\hspace{1cm}
\chemfig{@{atoo2}\chemabove{H}{\scriptstyle\oplus}}
\chemmove[-stealth,shorten <=3pt,dash pattern= on 1pt off 1pt,thin]{
\draw[shorten >=2pt](aton) ..controls +(up:10mm) and +(left:5mm)..(atoc);
\draw[shorten >=8pt](atoo1) ..controls +(up:10mm) and +(north west:10mm)..(atoo2);}
\chemrel[\tiny addition]{<>}
\chemfig{R-@{aton}\chembelow{N}{\scriptstyle\oplus}H([2]-[@{sb}]H)-C(-[2]CH_3)(-[6]CH_3)-OH}
\par
\chemmove{
\draw[-stealth,dash pattern= on 1pt off 1pt,shorten <=3pt, shorten >=2pt]
(sb) ..controls +(left:5mm) and +(135:2mm)..(aton);}
\chemrel{<>}
\chemfig{R-@{aton}\lewis{2,N}([6]-[@{sbh}]H)-[@{sb}]C(-[2]CH_3)(-[6]CH_3)-[@{sbo}]@{atoo}
\chemabove{O}{\scriptstyle\oplus})(-[1]H)(-[7]H)}
\chemmove[-stealth,shorten <=3pt,shorten >=2pt,dash pattern= on 1pt off 1pt,thin]{
\draw(aton) ..controls +(up:5mm) and +(up:5mm)..(sb);
\draw(sbh) ..controls +(left:5mm) and +(south west:5mm)..(aton);
\draw(sbo) ..controls +(up:5mm) and +(north west:5mm)..(atoo);}
\chemrel[\tiny elimination]{<>}
\chemfig{R-N=C(-[1]CH_3)(-[7]CH_3)}
\chemsign{+}
\chemfig{H_3\chemabove{O}{\scriptstyle\oplus}}

```



Reaction mechanism of chlorination

```

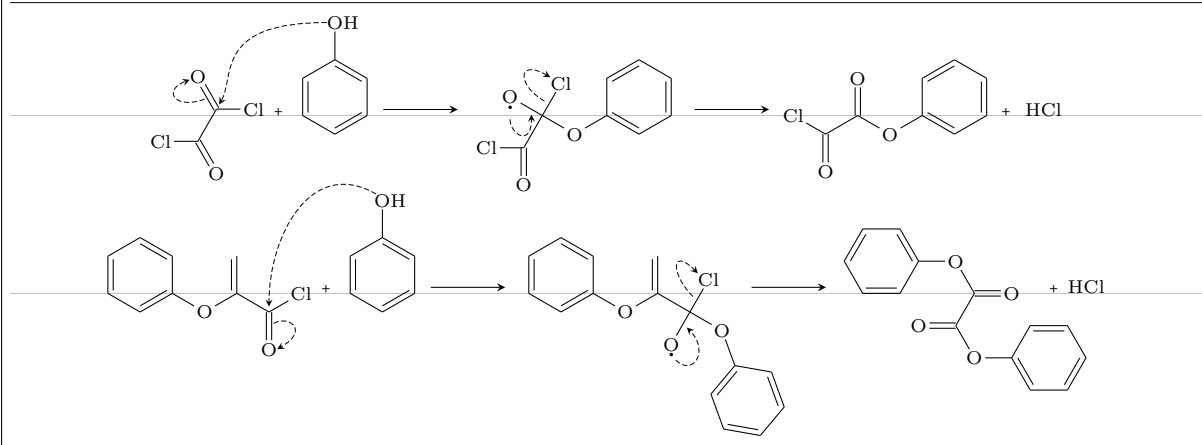
\scriptsize
\setbondoffset{1pt}\setatomsep{2em}
\chemfig{Cl-[4]@{a0}(=[@{a1}:120]@{a2}O)-[: -120](=[-60]O)-[4]Cl}\chemsign{+}
\chemfig{*6(---(-@{oh1}OH)-)}\chemrel{-stealth}
\chemfig{*6((-[:150](-[:0]@{o1}:150)@{o1}\lewis{6,.0})(-[:c10]:60)@{c11}Cl)-[:240](-[4]Cl)=[6]O)====)}
\chemrel{-stealth}\chemfig{*6((-[:150]([2]O)-[: -150]([6]O)-[:150]Cl)====)}\chemsign{+}
\chemfig{HCl}\par
\chemfig{*6(==6(-O-*6(-@{o2})(=[@{o3}]@{o4}O)-Cl)=-)}\chemsign{+}
\chemfig{*6(==(-@{oh2}OH)-)}\chemrel{-stealth}
\chemfig{*6(==6(-O-(-[:c12]:60)@{c13}Cl)(-[:o5]:-120)@{o6}\lewis{6,.0})-O-[: -40]*6(====))=)}
\kern-3em \chemrel{-stealth}\chemfig{[:30]*6(=-O-[: -60](=O)-[: -120]([4]O)-[: -60]O-*6(====))=)}
\kern-3em \chemsign{+}\chemfig{HCl}
\chemmove[line width=0.2pt,-stealth,dash pattern= on 2pt off 1pt]{
\draw[shorten <=2pt](a1) ..controls+(200:5mm)and+(200:5mm)..(a2);
}

```

```

\draw[shorten >=2pt](oh1.west)..controls+(180:15mm)and+(60:5mm)..(a0);
\draw[shorten <=6pt,shorten >=2pt](o1)..controls+(270:5mm)and+(270:5mm)..(o0);
\draw[shorten <=2pt](c10)..controls+(150:5mm)and+(150:5mm)..(c13.150);
\draw[shorten <=2pt](o3)..controls+(30:3mm)and+(30:5mm)..(o4.east);
\draw[shorten >=2pt](oh2.135)..controls+(150:10mm)and+(90:10mm)..(o2);
\draw[shorten >=2pt,shorten <=5pt]([xshift=-1.5mm]o6.315)..controls+(315:5mm)and+(315:5mm)..(o5);
\draw[shorten <=2pt](c12)..controls+(135:5mm)and+(135:5mm)..(c13.north west);}

```

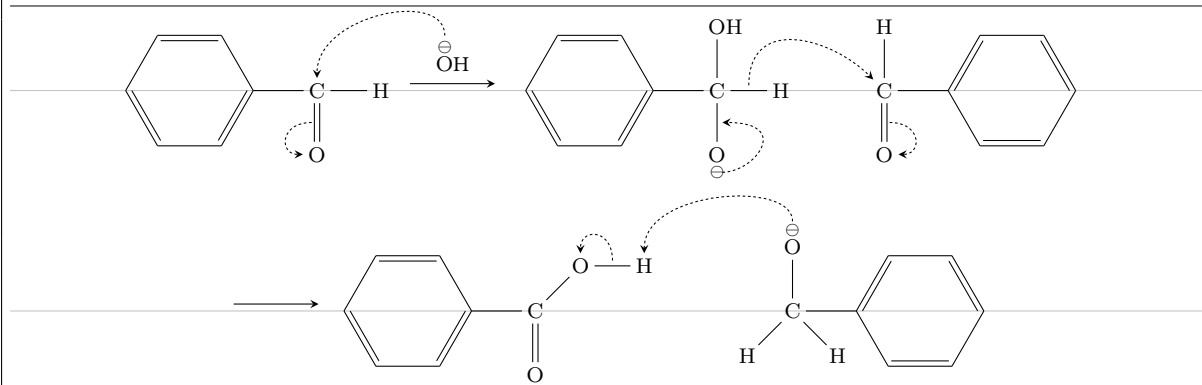


Cannizzaro reaction

```

\chemfig{[: -30]*6(==(-@{atoc}C([6]=[@{db}]@{atoo1}0)-H)---)}
\chemrel[\chemfig{@{atoo2}\chemabove{0}{\scriptstyle\ominus}}H]{-stealth}
\chemmove[-stealth,shorten >=2pt,dash pattern=on 1pt off 1pt,thin]{
  \draw[shorten <=8pt](atoo2)..controls+(up:10mm)and+(up:10mm)..(atoc);
  \draw[shorten <=2pt](db)..controls+(left:5mm)and+(west:5mm)..(atoo1);}
\chemfig{[: -30]*6(==(-C([6]-[@{sb1}]@{atoo1}\chembelow{0}{\scriptstyle\ominus})}([2]-OH)-[@{sb2}]H)---)}
\hspace{1cm}
\chemfig{[: -30]*6((-@{atoc}C([6]=[@{db}]@{atoo2}0)-[2]H)-----)}
\chemmove[-stealth,shorten <=2pt,shorten >=2pt,dash pattern=on 1pt off 1pt,thin]{
  \draw([yshift=-4pt]atoo1.270)..controls+(0:5mm)and+(right:10mm)..(sb1);
  \draw(sb2)..controls+(up:10mm)and+(north west:10mm)..(atoc);
  \draw(db)..controls+(right:5mm)and+(east:5mm)..(atoo2);}
\vspace{1cm}
\par
\chemrel{-stealth}
\chemfig{[: -30]*6(==(-C([1]-@{atoo2}0-[@{sb}0]@{atoh}H)([6]=0)---)} \hspace{1cm}
\chemfig{[: -30]*6((-C(-[5]H)(-[7]H)-[2]@{atoo1}\chemabove{0}{\scriptstyle\ominus})}-----)}
\chemmove[-stealth,shorten >=2pt,dash pattern=on 1pt off 1pt,thin]{
  \draw[shorten <=7pt](atoo1.90)..controls+(+90:8mm)and+(up:10mm)..(atoh);
  \draw[shorten <=2pt](sb)..controls+(up:5mm)and+(up:5mm)..(atoo2);}

```



Beckmann rearrangement

```

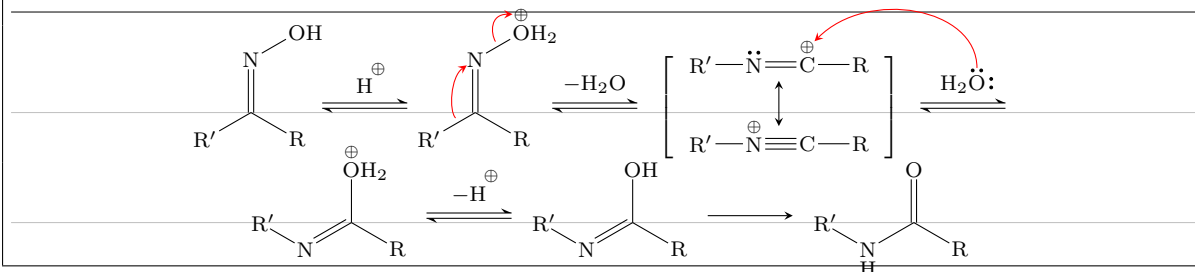
\setbondoffset{1pt}
\setatomsep{2.5em}
\chemfig{(-[: -150]R')(-[: -30]R)= [2]N-[: 30]OH}
\kern-1em
\chemrel[\chemfig{\chemabove{H}{\kern10pt\scriptstyle\oplus}}]{<>}
\kern-0.5em

```

```

\chemfig{(-[a0]:-150]R')(-[:-30]R)=[2]@[a1]N-[b0]:30]@[b1]\chemabove{0}{\scriptstyle\oplus}H_2}
\chemmove[red,-stealth,red,shorten <=2pt]{
  \draw(a0)..controls +(135:2mm) and +(215:4mm).. (a1);
  \draw(b0)..controls +(120:2mm) and +(180:3mm).. ([yshift=7pt]b1.180);
}
\kern-1.5em
\chemrel[\chemfig{{-}H_20}]{<>}
$\left[\begin{tabular}{c}
  \chemfig{R'-\lewis{2:,N}=@{a1}\chemabove{C}{\scriptstyle\oplus}-R}\backslash
  \tikz\draw[stealth-stealth](0,3mm)--(0,-3mm);\backslash
  \chemfig{R'-\chemabove{N}{\scriptstyle\oplus}~C-R}
\end{tabular}\right]$
\chemrel[\chemfig{H_2@[a0]\lewis{0:2:,0}}]{<>}\par
\chemmove[red,-stealth,red,shorten <=3pt]{
  \draw(a0)..controls+(90:10mm)and+(45:10mm)..([yshift=6pt]a1.45);
}
\chemfig{*6(R\rlap{$'$}-N(-R)-\chemabove{0}{\scriptstyle\oplus} H_2)}
\chemrel[\chemfig{{-}\chemabove{H}{\kern10pt\scriptstyle\oplus}}]{<>}
\chemfig{*6(R\rlap{$'$}-N(-R)-OH)}
\chemrel{-stealth}
\chemfig{*6(R\rlap{$'$}-\chembelow{N}{H})-(-R)(=[2]O)}

```

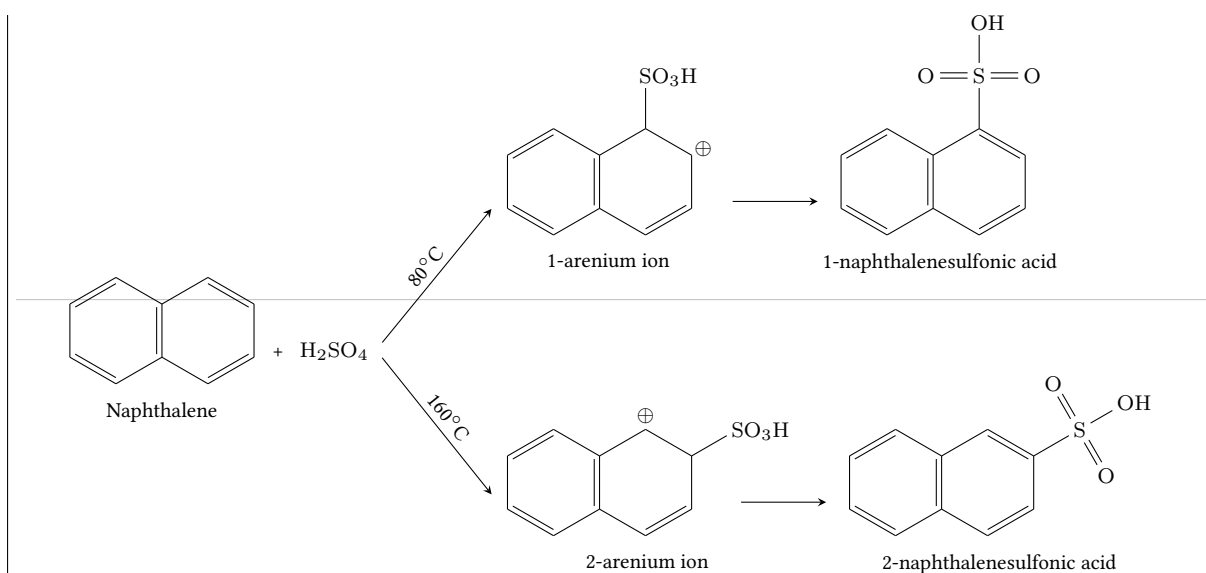


Sulfonation of naphthalene

```

\newcommand\putchemnode[1]{\begin{tikzpicture}[remember picture]\node(#1){};\end{tikzpicture}}
\definesubmol\cycleoplus{-[,0.25,, ,draw=none]\oplus}
\definesubmol{so2oh}{S(=[:90]O)(=[::-90]O)-OH}
\footnotesize
\setatomsep{2.5em}
\begin{tabular}{ll}
& \&\putchemnode{11}\chemname{\chemfig{*6(=-*6(=-(\cycleoplus)-(-SO_3H)-=-)}}{1-arenium ion}} \\
& \chemrel{-stealth} \\
& \chemname{\chemfig{*6(=-*6(=-(\so2oh)-=-)}}{1-naphthalenesulfonic acid}} \\
& \chemname{\chemfig{*6(=-*6(=---)}}{Naphthalene}} \\
& \chemsign{+} \\
& \chemfig{H_2SO_4}\putchemnode{a}\kern1cm\backslash[-5ex] \\
& \&\putchemnode{12}\chemname{\chemfig{*6(=-*6(=-(SO_3H)-(\cycleoplus)-=-)}}{2-arenium ion}}\kern-3em \\
& \chemrel{-stealth} \\
& \chemname{\chemfig{*6(=-*6(=-(\so2oh)-=-)}}{2-naphthalenesulfonic acid}} \\
\end{tabular}
\chemmove[-stealth,yshift=1cm]{
  \draw(a)--node[sloped,above]{80\degrees C}(11);\draw(a)--node[sloped,above]{160\degrees C}(12);}

```

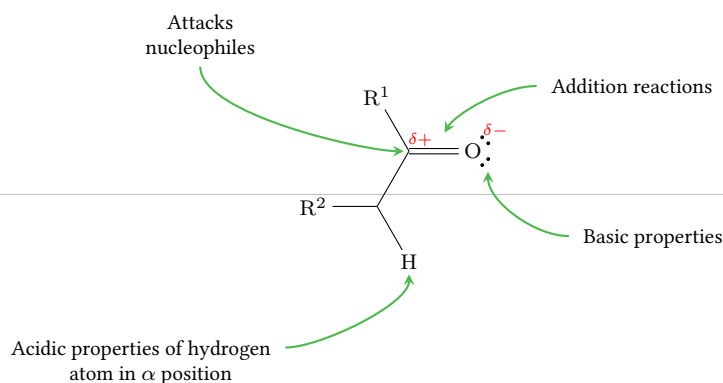


Explanatory diagram

```

\parbox{0.8\linewidth}{
\hspace{10em}
\tikz[remember picture]\node(n0){\chemname{}{Attacks\nucleophiles}};\par
\vspace{2ex}\hspace{15em}
\chemfig{R^2-(-[:60]@{a0}H)-[:60]@{a1}(-[:120]R^1)(-[:1,0.25,,,draw=none]\scriptstyle\color{red}\delta+)=
[:a2]@{a3}\lewis{1:7:,0}-[:1,0.5,,,draw=none]\scriptstyle\color{red}\delta-}}
\hspace{5em}
\chemname[-15ex]{}{\tikz[remember picture]\node(n1){};Addition reactions}\kern1em
\chemname{}{\tikz[remember picture]\node(n2){};Basic properties}\par
\vspace{6ex}\hspace{8em}
\chemname{}{Acidic properties of hydrogen\tikz[remember picture]\node(n3){};
\atom in $\alpha$ position}
\chemmove[-stealth,line width=0.8pt,green!60!black!70]{
\draw[shorten >=2pt](n0)..controls+(270:4em)and+(180:2em)..(a1);
\draw[shorten >=8pt](n1)..controls+(180:2em)and+(60:2em)..(a2);
\draw[shorten >=5pt](n2)..controls+(180:2em)and+(270:2em)..([xshift=3pt]a3.315);
\draw[shorten >=2pt](n3)..controls+(0:2em)and+(270:2em)..(a0);
}}

```



Crystallography

```

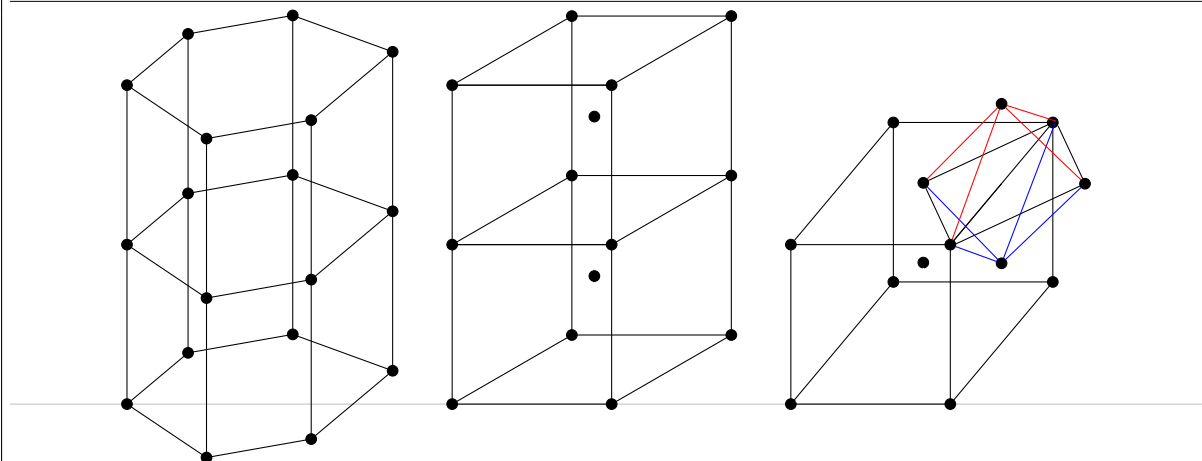
\newcommand\disk{\tikz\draw[fill=black,overlay](0,0)circle(2pt);}
\setatomsep{20pt}
\renewcommand\printatom[1]{#1}
\setbondoffset{2pt}
\definesubmol{hat}{-[:40,1.5]\disk-[:-30,2]\disk-[:-30,2]\disk-[:-120,2]\disk-[:-30,2]\disk}
\definesubmol{motif}{-[:40,1.5]\disk(-[:2,3])-[:-30,2]\disk
(-[:2,3])-[:-30,2]\disk(-[:2,3])-[:-120,2]\disk(-[:2,3])-[:-30,2]\disk}
\chemfig{\disk?[a](-[:2,3]\disk?[b](-[:2,3]\disk?[c]!\hat)?[c]!\motif?[b](-[:2,3]))!\motif?[a](-[:2,3])}
\qqad
\redefinesubmol{motif}{\disk(-[:2,3])(-[:42,3.6,,,draw=none]\disk)-[:30,2.6]\disk}

```

```

(-[2,3])-[0,3]\disk(-[2,3])-[: -150,2.6]\disk(-[4,3])-[2,3]-[4,3]}
\redefinesubmol{hat}{\disk-[:30,2.6]\disk-[0,3]\disk-[: -150,2.6]\disk-[4,3]}
\chemfig{!{motif}!{motif}!{hat}}
\qqquad
\redefinesubmol{motif}{(-[2,3])(-[:25,2.75,,white]-[2,1.5,,white]\disk)-[:50,3]\disk
(-[2,3])-[: -50,3]\disk(-[2,3])-[: -130,3]\disk-[2,3]-[4,3]\disk}
\redefinesubmol{hat}{-[:50,3]\disk-[: -50,3]\disk-[: -130,3]\disk}
\chemfig{-[4,3]\disk!{motif}(-[:25,2.75,,draw=none]\disk?[uat]?[dat](-[:0,2.75]?[uat1]?[dat1]
-[: -90,1.3]\disk?[uat2]?[dat2])-[: -90,1.3]-[:25,2.75])!{hat}?[uat3]?[dat3]-[:50,1.5]
(-[6,1.5,,draw=none]\disk?[dat,,blue]?[dat1,,blue]?[dat2,,blue]?[dat3,,blue])
-[2,1.5,,draw=none]\disk?[uat,,red]?[uat1,,red]?[uat2,,red]?[uat3,,red]}

```

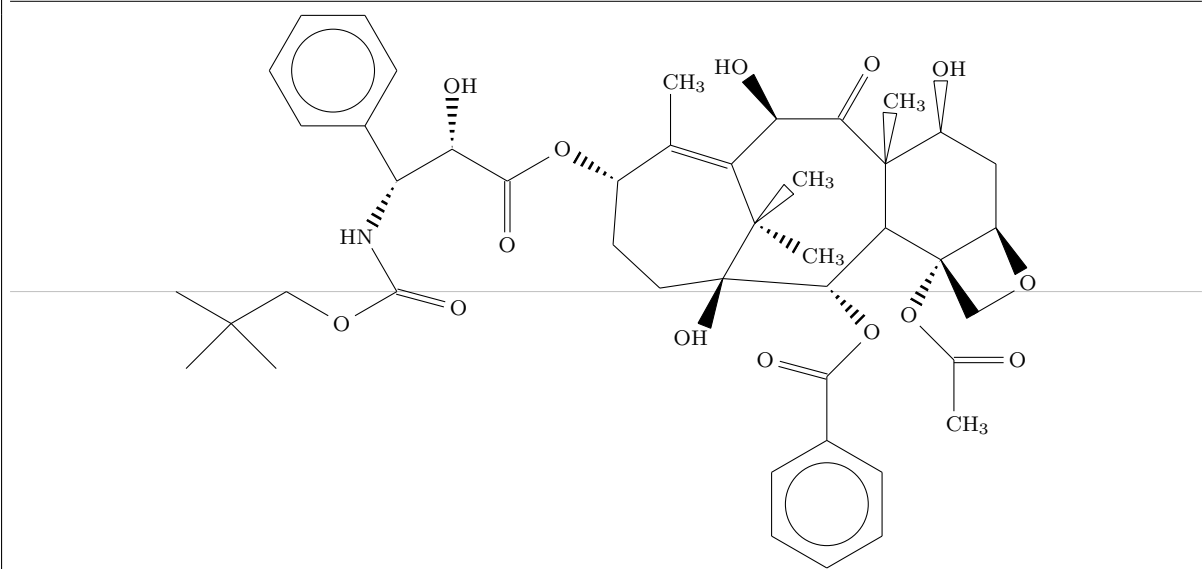


Taxotere

```

\chemfig{-[: -30](-[5])(-[7])-[: +60]-[: -60]0-[: +60](=[: -45]0)-[: +90]HN>[: -60](-[: +60]**6(-----))
-[: -30](<[:2]OH)-[: -60](=[:6]0)-[: +60]0>[: -60]*7(---?(<[: -120]OH)-<[:1]CH_3)<[: -90]CH_3)
-(-[:1](<[: +80]HO)-[:0](=[: +60]0)-[:7](<[: +130]CH_3)-[: +75](<[:2]OH)-[: -60]-[: -60](<[: +30]0-[: -90])
-[: -60](<[: +90])<[: +30]0-[:7](-[:6]CH_3)=[:0]0)-[: -60])-[6]-[:5,1.3]?(<[:7]0-[:5](-[: -60]0)
-[:6]**6(-----)))=(-[:2]CH_3)-}

```



Index

- Symboles**
- ! *see* save a molecule
 (..... *see* branched molecule
 * *see* ring
 ** *see* ring/inside arc
 -, =, ~ *see* bond/types
 =_, =^ *see* bond/shifted
 >, <, >|, <|, >:, <: *see* Cram diagram
 ? *see* bond/distant atoms
 @ 22, 45–47, 56–58, 60
 | *see* separating atom mechanism
 # 11, 30
 \\ 26
- B**
- baseline 4, 5, 25, 29
 bond
 angle 5, 6, 11
 absolute 6, 12, 37
 arc 47
 predefined 6, 12, 37
 relative 6, 12, 32, 38
 branch 37
 customization 5, 14
 delocalized 31
 departure and arrival atoms 5, 13, 22
 distant atoms 8, 17
 invisible 14, 42, 45
 length 5, 6, 10, 13
 optional argument 5, 6, 11–15, 22
 shifted 7, 30
 types 5, 10
 wavy 45
 with text 47
 bounding box 4, 25, 33, 34
 branched molecule 6, 7, 15
- C**
- chair representation 40
 charge 45
 \backslash delta 60
 \backslash ominus 9, 56, 58
 \backslash oplus 9, 24, 45, 56–59
 \backslash chemabove 9, 25, 33, 45, 49, 56–58
 \backslash chembelow 9, 33, 43, 49, 55–58
 \backslash chemfig 4, 5, 14, 23, 28, 35, 48
 optional arguments 28, 37, 41, 42, 46
 chemical reaction 9, 25, 34
 \backslash chemmove 23–25, 45, 49, 56–60
 \backslash chemname 25, 49, 59, 60
 \backslash chemnameinit 26, 49
 \backslash chemrel 9, 34, 49, 56–59
 \backslash chemsign 9, 34, 49, 56, 57, 59
 \backslash chemskipalign 29, 30, 48
 \backslash chemnameinit 25
- \backslash color 44, 60
 color (atom) 44
 compilation method 4
 ConTeXt 3, 4, 28
 control points 23–25
 Cram representation 7, 11
- D**
- \backslash definesubmol 31, 32, 40, 42, 43, 45, 48, 50–54, 59, 60
 dvi mode 4
- E**
- electron movements 22–25
 ϵ TeX 3, 28
- F**
- Fisher projection 39
- H**
- Haworth projection 41
 \backslash hflipnext 47, 48
 horizontal alignment 45
- I**
- \backslash input 4
 interatom length *see* bond/length
- L**
- L^ATeX 3, 4, 27
 \backslash Lewis 9, 32, 33, 49, 56–58, 60
 Lewis diagram 7, 32
- M**
- math mode 9, 11, 27
- P**
- pdf mode 4
 pgf 4
 \backslash phantom 50
 picture remembering 22
 polymer 46
 \backslash printatom 27, 28, 35, 36, 48, 60
- R**
- \backslash redefinesubmol 32, 49, 60
 ring
 angular position 19
 branch 8, 20, 38, 41
 charge 34
 inside arc 19
 nested 21, 38, 42, 43
 unfinished 19, 38
 \backslash rlap 45
- S**
- save a molecule 31, 32
 separating atom mechanism 10, 27, 44

`\setatomsep` 11, 13, 46, 49, 57–60
`\setbondoffset` 11, 35, 49, 57, 58, 60
`\setchemrel` 34, 49
`\setcrambond` 11, 41, 49, 52, 53
`\setdoublesep` 5, 49
`\setlewis` 33, 49
`\setnodestyle` 35, 48
size of molecules 13, 28, 46
skeleton diagram 7, 39
symmetrical of a molecule 47

T

`\textcolor` 44
`tikz` 4, 14
`tikzpicture` 4, 22, 23, 28, 35

U

`\usemodule` 4
`\usepackage` 4

V

vertical alignment 29
`\vflipnext` 47, 48
`\vphantom` 29